

# CGEBox – a flexible and modular toolkit for CGE modelling with a GUI

---

- Wolfgang Britz, Institute for Food Resource Economics –  
- October 2022 -  
(matches rev. 4850 of <https://svn1.agp.uni-bonn.de/svn/cgebox>)

## Content

1.	Introduction .....	9
1.1	How the documentation is organized .....	9
1.2	Why an implementation in GAMS? .....	9
1.3	Scalability, Modularity and Extensions as basic features of CGEBOX .....	10
2.	Basic model equations .....	16
2.1	Core sets .....	16
2.2	Reminder on CES and CET equations .....	17
2.3	Overview on the supply side .....	19
2.4	Production block .....	20
2.5	Factor markets .....	29
2.6	Income generation and distribution - overview .....	33
2.7	Income generation .....	33
2.8	Income distribution .....	36
2.9	Household consumption .....	37
2.9.1	CDE case .....	38
2.9.2	LES or CD case .....	39
2.10	Government consumption .....	41
2.11	Investments and savings .....	42
2.11.1	Global bank .....	44
2.11.2	Fixed allocation of savings .....	47
2.11.3	Capital account and balance of payments .....	47
2.12	Demand sub-nests .....	48
2.12.1	Motivation .....	48
2.12.2	Implementation in model .....	49
2.12.3	Benchmarking of demand sub-nests .....	51
2.12.4	References .....	55
2.13	International trade and domestic sales, and related prices .....	56

2.13.1	Overview .....	56
2.13.2	Individual equations .....	57
2.13.3	Melitz model.....	65
2.14	Price indices .....	70
2.15	List of main prices in model.....	71
2.16	Flexible nesting in the production function.....	72
3.	Modules .....	78
3.1	myGTAP module .....	78
3.1.1	Summary .....	78
3.1.2	Motivation .....	78
3.1.3	Implementation.....	79
3.1.4	Government debt dynamics.....	86
3.1.5	Methodology: Household demand.....	90
3.1.6	Graphical User Interface.....	96
3.1.7	Code implementation.....	99
3.1.8	Integration FAO household data into CGEBOX.....	102
3.1.9	Post-model micro-simulation .....	115
3.2	A GMIG module for CGEBox .....	123
3.2.1	Background and motivation .....	123
3.2.2	Equations.....	123
3.2.3	Technical aspects.....	125
3.2.4	Result exploitation.....	126
3.3	Sub-national dis-aggregation of production and factor markets in CGEBox.....	126
3.3.1	Background and motivation .....	126
3.3.2	Methodology .....	126
3.3.3	Integration into the modeling framework.....	127
3.3.4	Implementation for European countries at NUTS2 level .....	129
3.3.5	Result exploitation.....	132
3.4	Armington with commitment terms .....	133
3.4.1	Background .....	133
3.4.2	Benchmarking and GUI setting .....	133
3.4.3	Equations.....	135
3.5	Spatial Price Equilibrium instead of Armington .....	138
3.5.1	Background .....	138



3.5.2	Equations and technical implementation.....	139
3.5.3	Interface.....	140
3.6	MRIO extension .....	142
3.6.1	Background .....	142
3.6.2	Definition of consistent split factors.....	143
3.6.3	Model equations .....	147
3.6.4	Graphical User Interface.....	149
3.7	Partial equilibrium closure .....	151
3.8	Tariff Line Module.....	152
3.8.1	Summary .....	152
3.8.2	Background .....	152
3.8.3	Approach .....	153
3.8.4	Model equations .....	153
3.8.5	Entering the data.....	156
3.8.6	Calibration .....	157
3.8.7	Post model processing .....	158
3.8.8	Integration in the GUI .....	158
3.9	CO <sub>2</sub> , Non-CO <sub>2</sub> Accounting and air pollution emissions, energy use accounting .....	160
3.9.1	GHG emissions in CO <sub>2</sub> equivalents.....	160
3.9.2	Marginal abatement curves for GHG emissions .....	162
3.9.3	Air pollution emissions.....	177
3.9.4	Reporting of energy use in Million tons of oil equivalents (MTOE) .....	178
3.10	GTAP-E.....	178
3.11	GTAP-AGR.....	178
3.12	G-RDEM: Long-term baseline generation and analysis.....	180
3.12.1	Summary .....	180
3.12.2	Background and introduction .....	180
3.12.3	Overview .....	181
3.12.4	An AIDADS demand system with detail for food consumption.....	183
3.12.5	Differentiated productivity growth.....	194
3.12.6	Endogenous saving rates .....	198
3.12.7	Debt accumulation from foreign savings.....	202
3.12.8	Cost-share adjustment .....	205
3.12.9	Endogenous expenditure shares for government and investment demand .....	213

3.12.10	Recovering crop yield projections.....	215
3.12.11	Recovering crop land expansion .....	217
3.12.12	Steering of G-RDEM model runs.....	221
3.12.13	Population and GDP growth from the IIASA SSPs .....	222
3.12.14	Using the G-RDEM generated baseline for counterfactual analysis .....	224
3.12.15	Specific nestings in G-RDEM .....	226
3.12.16	Endogenous non-Hicks neutral technical progress and primary factors supply .....	227
3.12.17	Analysis of G-RDEM simulation output (exploitation) .....	229
3.12.18	Computing and data base considerations for G-RDEM runs .....	230
3.13	GTAP-Melitz: Heterogenous firm module.....	232
3.13.1	Summary .....	232
3.13.2	Introduction .....	232
3.13.3	Brief Literature overview .....	234
3.13.4	Implementation of Melitz model into CGEBox .....	236
3.13.5	Numerical stability and domestic industry demand for its own output .....	249
3.13.6	Technical implementation and an example application.....	249
3.13.7	Conclusion.....	254
3.13.8	Acknowledgments .....	255
3.14	GTAP-AEZ .....	256
3.14.1	Summary .....	256
3.14.2	Model equations .....	256
3.14.3	Parameterization.....	264
3.14.4	Visualization of GTAP-AEZ results.....	266
3.15	Climate Change Damages .....	269
3.15.1	Data input .....	269
3.15.2	Aggregation .....	269
3.15.3	Graphical User Interface.....	271
3.15.4	Shock definition .....	272
3.15.5	References .....	275
4.	Data base generation .....	276
4.1	Direct input from GDX version of GTAP Data Base.....	276
4.1.1	Background .....	276
4.1	Background .....	277
4.2	Importing the data into CGEBox.....	277

4.3	GTAP Data base import using zip containers for FLEXAGG .....	280
4.3.1	Set-up and installation of base data .....	280
4.3.2	Adding GMIG, AEZ and MRIO split factor .....	282
4.4	Deprecated: Input from GTAPagg .....	282
4.5	GTAPAGG: The data base generation utility of CGEBox.....	283
4.5.1	Defining your aggregation: The aggregator component.....	284
4.5.2	Aggregation in GAMS .....	288
4.5.3	Backward compatibility.....	288
4.5.4	Data base treatment after aggregation: Filtering or not.....	289
4.5.5	Simple deletion.....	289
4.5.6	Filtering and Rebalancing.....	290
4.5.7	Gender Differentiated labor data .....	300
4.5.8	Split utility .....	300
4.5.9	A complex example for a split.....	302
4.5.10	Balancing equations and objective function in the split problem.....	304
4.5.11	Proportionality assumptions for cost structures during splitting .....	310
4.5.12	Preparation of the split factors.....	311
4.5.13	Split solution process.....	315
4.5.14	Processing the solution from the split problem .....	316
4.5.15	MRIO split.....	318
4.5.16	Post-aggregation of GTAPAgg to yield non-diagonal make matrices .....	318
4.5.17	Storing results from GTAPAgg as a new Data base for further processing .....	319
4.6	Model size and solution behavior.....	321
4.7	Inspecting the resulting data base.....	324
4.8	Pre- or post-model aggregation .....	325
5.	Scenario runs .....	327
5.1	Defining the scenario.....	327
5.1.1	Implementation of shock files in GAMS.....	327
5.1.2	Scenario editor.....	330
5.2	Running the scenario .....	334
5.3	Output options .....	337
5.4	Model structure, parameterization and factor markets .....	339
5.4.1	Modules and parameters.....	339
5.4.2	Demand system .....	340

5.5	Factors .....	340
5.6	Parameters and indices .....	341
5.6.1	Indices .....	341
5.6.2	Scenario drivers .....	341
5.6.1	Parameters .....	342
5.7	Closures .....	343
5.8	Imperfect competition model .....	345
5.9	Pre-configurations .....	345
5.9.1	Standard GTAP model .....	346
5.9.2	GTAPinGAMS .....	346
5.9.3	GLOBE .....	346
5.9.4	MIRAGE .....	347
5.9.5	ENVISAGE .....	347
5.9.6	CGEBOX .....	347
5.9.7	CGEBOX+ .....	347
5.10	Options for solving the model: Algorithm .....	347
5.10.1	First-order Taylor approximation to define starting points .....	349
5.10.2	Variable initialization before solving .....	350
5.10.3	Presolves .....	350
5.11	GAMS .....	351
5.12	Pre-solves .....	353
6.	Post-model processing .....	358
6.1	Decomposing welfare changes in CGEBox .....	358
6.1.1	Background .....	358
6.1.2	Decomposition with regard to overall price indices and income components .....	358
6.1.3	Decomposition with a focus on prices in changes .....	360
6.1.4	Demand nests .....	360
6.1.5	Private household .....	360
6.1.6	Government and investments .....	362
6.1.7	Utility change for the regional household and EV .....	362
6.1.8	Decomposition following Huff and Hertel, 2000 .....	364
6.2	Trade in VA indicators .....	367
6.2.1	Background .....	367
6.2.2	Post-Model processing .....	368

6.2.3	GUI.....	369
6.3	Altax.....	372
6.3.1	Background .....	372
6.3.2	Technical realization .....	372
6.3.3	Pre-configuration.....	373
6.3.4	Post-Model processing .....	373
6.3.5	Use of the updated benchmark .....	374
6.4	Linking the physical MRIO for agricultural and food products FABIO to CGEBox .....	375
6.4.1	Introduction .....	375
6.4.2	Overview on FABIO .....	375
6.4.3	Methodology .....	376
6.4.4	Introduction .....	376
6.4.5	Split .....	377
6.4.6	Post-model dis-aggregation.....	381
6.4.7	Nutrition indicators.....	385
7.	GUI and installation .....	388
7.1	Software requirements and access to the code .....	388
7.2	Some comments for runGTAP and GEMPACK users.....	388
7.3	Background .....	389
7.4	SVN checkout.....	390
7.5	Starting the GUI .....	393
7.6	First steps.....	395
7.7	Making the interface working with GAMS.....	397
7.8	Inspecting the results .....	401
7.8.1	Notes for GEMPACK users or users familiar with other CGE models .....	403
7.8.2	Relative difference for all variables as in runGTAP .....	404
7.8.3	Trouble shooting with the viewer.....	406
7.8.4	Overview on existing views for exploitation.....	407
8.	Sensitivity analysis .....	417
8.1	Controls for sensitivity experiments.....	418
8.2	Parallel solves and analyzing the results .....	423
9.	Getting an overview on the code .....	428
9.1	HTML documentation .....	428
9.2	Equation and variable viewer (similar AnalyzeGE).....	430

9.3	Technical documentation.....	433
9.3.1	Overview on directory structure.....	433
9.3.2	GUI overview .....	433
9.3.3	Processes .....	434
9.3.4	Post-model processing.....	435
9.3.5	Substitution of variables based on macros .....	437
9.3.6	Exploitation and flexible aggregation.....	439
9.3.7	Main programs when solving a shock and integration of modules .....	442

## 1. Introduction

### 1.1 How the documentation is organized

This document (1) describes the methodology of CGEBox – including an equation by equation description of the GTAP standard model as implemented in here and of the different optional modules –, (2) provides its technical documentation and (3) details how to use the Graphical User Interface (GUI) and of to introduce specific data or shocks. The GTAP standard model and each module each have their own sections, these sections start with an introduction to the methodology, next detail the equations and the implementation in GAMS, followed (where applicable) by sub-sections on how to introduce additional data and on GUI choices and result exploitation.

Users who want to start using the basic model without having a look at details are invited to read the following two sections “Why an implementation in GAMS?” and “Scalability, Modularity and Extensions as basic features of CGEBOX” up to page 16 as an introduction, skip the following detailed sections first and continue with the section “GUI and installation”, page 388 ff.

### 1.2 Why an implementation in GAMS?

[GTAP](#) is certainly the far most widespread international data base for CGE (Computable General Equilibrium) modeling. The data base was for years only distributed with a default CGE model template in [GEMPACK](#), called the [standard GTAP model](#), as well as with many useful GEMPACK based utilities. GEMPACK as a special package for CGE modelling has certainly distinct advantages compared to less specialized Algebraic Modeling Languages<sup>1</sup>. Some users might however prefer [GAMS](#) when working with GTAP. Some features of GAMS not available with GEMPACK are used in CGEBox, such as being able to solve (several) constrained non-linear equation systems not written in their FOC in one program call, for instance to filter small entries from the global SAM and rebalancing it afterwards, see section “Filtering and Rebalancing”, or when applying the split utility, see section “Split utility”.

Indeed, outside the CGE community, GAMS seems much more common than GEMPACK as an Algebraic Modelling Language (AML), e.g. in Agricultural Economics (Britz and Kallrath 2012<sup>2</sup>). In 2015, the Center for Global Trade Analysis (GTAP) decided to release in parallel to the standard GTAP model in GEMPACK a version coded in GAMS, now available for GTAP Standard Version 7 (van der Mensbrugghe 2018<sup>3</sup>). Intensive testing showed that both versions produce the same results. The GAMS equations of that core model plus some extensions are discussed below in the section “Basic model equations”. These equations are mostly structurally identical to van der Mensbrugghe 2018 to which this documentation refers where applicable, while mnemonics and coding style are to a large degree harmonized as well. Further CGE models realized in GAMS are widely available. They share basic elements of the standard GTAP model while adding extensions in several directions, for

---

<sup>1</sup> Horridge, M., Meeraus, A., Pearson, K., & Rutherford, T. F. (2013). “Solution software for computable general equilibrium modelling”. In: Handbook of computable general equilibrium modeling (Vol. 1, pp. 1331-1381). Elsevier

<sup>2</sup> Britz, W., Kallrath, J. (2012): Economic Simulation Models in Agricultural Economics: The Current and Possible Future Role of Algebraic Modeling Languages, in: Kallrath, J. (eds.): Algebraic Modelling Systems: Modeling and Solving Real World Optimization Problems: 199-212, Springer, Heidelberg, Germany

<sup>3</sup> Van der Mensbrugghe, D. (2018). “The Standard GTAP Model in GAMS”, Version 7. *Journal of Global Economic Analysis*, 3(1), 1-83

instance [GTAPinGAMS](#) by Tom Rutherford<sup>4</sup>, [GLOBE](#) by Scott McDonald and Karen Thierfelder or [ENVISAGE](#) by Dominique van der Mensbrugghe. However, these models do not provide an exact replication of the standard GTAP Model.

The newly released replica of the standard GTAP model version 7 in GAMS comprised in CGEBox is not only interesting for research, but also provides together with the GTAP Data Base an excellent starting point either for teaching or own projects of students, especially in the context of course programs already using GAMS. CGEBox offers additionally some extensions compared to the standard GTAP model as discussed next, inspired by features of existing CGEs, and adds modules such as myGTAP, GMIG, GTAP-AEZ, GTAP-AGR, GTAP-E or GTAP-HET detailed in dedicated sections of this documentation. Furthermore, CGEBox offers a choice for certain structural elements such as with regard to the functional form in final demand or the aggregation of the Armington agents. The combination of the extensions and modules along with such flexibility allow adjusting the model's structure to a specific application or to replicate main layout features of well-known models (see Britz and van der Mensbrugghe 2018<sup>5</sup>).

The model documentation is structured as follows. The following section “Scalability, Modularity and Extensions as basic features of CGEBOX” discusses the main features of the modeling platform including a brief overview on the different modules. The structure of the core model is discussed equation by equation in the section “Basic model equations” (page 16ff). The following section “Modules” (pages 78ff) discusses the methodological and technical details of the individual modules. It is followed by a block dealing with generating the data base of the model (“Data base generation”, pages 269ff) with issues such as filtering the global SAM or the split utility before a section on running the model (“Scenario runs”, pages 327 ff). Next, post-model processing is detailed: with the sub-section on “Decomposing welfare changes in CGEBox”, “Trade in VA indicators”, the discussion of the “Altertax” utility and “Linking the physical MRIO for agricultural and food products FABIO to CGEBox”. Afterwards, the use of the Graphical User Interface is presented in the section “GUI overview” (pages 388ff). It discusses step-by-step how to install the model, construct a data base, define the shock, run the model and analyze results. A further section details the use of an extension on sensitivity analysis (“Sensitivity analysis”, pages 417ff), before section “Getting an overview on the code” (pages 428 ff) discusses some technical aspects. Some training videos are available at [http://www.ilr.uni-bonn.de/em/rsrch/cgebox/cgebox\\_e.htm#vid](http://www.ilr.uni-bonn.de/em/rsrch/cgebox/cgebox_e.htm#vid) – they might differ however in selected details as the Graphical User Interface was improved since their recording.

### **1.3 Scalability, Modularity and Extensions as basic features of CGEBOX**

As the standard GTAP model and many of its variants, CGEBox is highly scalable in the sense that it can work with differently sized, but identically structured data bases which differ in the lists of regions, commodities, sectors and factors. Increasing the detail of the data base can quickly lead to models of a size which is hard or even impossible to solve. The modeling platform offers several options to work with large-scale data bases. Firstly, a filter algorithm (see section “Filtering and Rebalancing”, page 290ff) can systematically remove tiny elements from the global SAM and afterwards rebalances it. Second, the GAMS code comprises a pre-solve algorithm which should speed

---

<sup>4</sup> Rutherford T. and Harbor A. (2005). GTAP6inGAMS: The Dataset and Static Model. Prepared for the Workshop: “Applied General Equilibrium Modeling for Trade Policy Analysis in Russia and the CIS”, The World Bank Resident Mission, Moscow, December 1-9, 2005, <http://www.mpsge.org/gtap6/gtap6gams.pdf>

<sup>5</sup> Britz, W., van der Mensbrugghe, D. (2018): CGEBox: A Flexible, Modular and Extendable Framework for CGE Analysis in GAMS, Journal of Global Economic Analysis 3(2): 106-176



up solution of larger models with more than 300.000 non-empty transactions in the global SAM (see section “Pre-solves”, pages 353ff). The code was intensively tested for numerical stability and acceptable solution time when applied to more dis-aggregated versions of the GTAP data base (Britz et al. 2015, Britz and Van der Mensbrugghe 2016<sup>6</sup>). Third, certain extensions such as the possibility to use as in GLOBE the same import and domestic shares for all Armington agents can reduce the number of variables and equations and, finally, the model allows to substitute out many variables from the system which otherwise drive up model size.

CGEBox includes and supports extensions beyond the GTAP standard model. Using these extensions is partly driven by the underlying SAM structure, such as allowing for a multi-product make matrix to support non-diagonal relations between production activities and commodities outputted. Other extensions are switched on depending on the parameterization of the system, such as allowing for non-infinite CET between domestic sales and exports, and between export flows. Similarly, non-diagonal make matrices can be applied in conjunction with a CET nest to steer output composition of multi-output sectors and a CES nest to determine consumption shares in case several sectors produce the same commodity. Many of these extensions are inherited from ENVISAGE, however not all ENVISAGE features are found in the code. The code also supports already a range of different closures, for instances, for foreign savings, private and government consumption which can be complemented by closures files provided by the user. Equally, different regional numeraires can be used.

Furthermore, the code structure has a modular design which facilitates introduction of new model features to the GAMS code<sup>7</sup>. So far, implementations of GTAP-AGR (see section “GTAP-AGR”, page 178ff), GTAP-E (see section “160”, page 160), GTAP-AEZ (see section “GTAP-AEZ”, page 256ff), myGTAP (see section “myGTAP module”, page 78ff) and CO2 emissions accounting and taxation (see section “CO2, Non-CO2 Accounting”, page 160ff, this also covers air pollution emissions and energy use accounting oil equivalents) are available as additional modules. Different impacts of climate change can be captured by the Climate Change Damages Module (see section “Climate Change Damages”, page 269ff).

The options for international trade are especially evolved with an implementation of a Melitz model (GTAP-HET, see section “GTAP-Melitz: Heterogenous firm module”, page 180ff), a MRIO split up of bi-lateral import demand by total intermediate consumption and the final demand agents, (see section “MRIO extension”, page 142), the possibility to use an Armington specification with commitment term at the bilateral level (see section “Armington with commitment terms” page 133ff) and an implementation of a Spatial Price Equilibrium approach (see section “Spatial Price Equilibrium instead of Armington”, page 138ff).

---

<sup>6</sup> Britz, W., van der Mensbrugghe, D. . (2016): Reducing unwanted consequences of aggregation in large-scale economic models - A systematic empirical evaluation with the GTAP model, *Economic Modelling* 59: 462-473

<sup>7</sup> The code makes a differentiation between extensions which change the nesting of the production function or introduce nested CET structures for factor supply. These extensions only require set-definitions and substitution respectively transformation elasticities, but no changes in the equation structure of the model or is post-processing. In opposite to that, the implementation of what is called a module consists typically of two files: one file which declares the additional symbols (parameters, variables, equations), these includes are found in “model.gms”, and a file which calibrates the related parameters for the benchmark, these includes are found in “cal.gms”. Potentially, there are also additional statements in the post-model reporting part.

The options for international trade can be combined with different agent aggregations in the Armington nests and with a module which dis-aggregates selected bi-lateral trade flows to the tariff line (see “Tariff Line Module”, page 152ff). The G-RDEM (GTAP based Recursive-Dynamics Model<sup>8</sup>) is available as a module in CGEBox to generate (long-term) baselines (see section “G-RDEM: Long-term baseline generation and analysis”, page 180 ff.). Additionally, the production functions and factor markets for EU countries can be broken down to sub-regional regions, for instance, to 280 so-called administrative NUTS2 regions for the EU (see section “Sub-national disaggregation of production and factor markets in CGEBox”, page 123ff).

Extensions derived from GTAP-variants are mostly not set-up as a perfect replica of the GEMPACK implementations and not tested for replication of results obtained with the corresponding GEMPACK code. Examples of how these extensions and modules can be used to generate from the same code models with quite different features offer the pre-defined configurations; see section “Pre-configurations”, pages 345ff. The code supports currently GTAP6 to GTAP10 data bases with matching land use data for 2007, 2011 and 2014 and allows using the GTAP9-Water data set. The “Split utility” can be used in conjunction with the FABIO MRIO (see section “Linking the physical MRIO for agricultural and food products FABIO to CGEBox”, pages 375ff) to add agri-food detail to the GTAP SAMs. There are some specific extensions for post-model analysis: Trade in Value Added indicators (see section “Trade in VA indicators”, pages 367ff), decomposition of welfare changes (see section “Decomposing welfare changes in CGEBox”, pages 358ff) and lastly, providing detail for agriculture and food including nutritional indicators based on the FABIO MRIO (see section “Linking the physical MRIO for agricultural and food products FABIO to CGEBox”, pages 375ff). The model can also, similar to the Altermat utility (see section “Altermat”, pages 372ff), derive a global SAM and other parameters from a simulated shock which can be used as starting point for another run.

An important feature of the model’s structure are sets of equations which allow the introduction of CES-nests comprised of factors, intermediates demand and sub-nests such that even complex nestings in the production function can be introduced via cross-set definitions without additional programming work in the model’s code. A similar generic implementation is provided for factor supply based on nested CET-functions and for CES sub-nests under final demand. In all three cases, the post-model processing and reporting part reflects these nesting as well. These features are discussed below in more detail in the section “Flexible nesting”, pages 72ff.

The basic application mode of the model, according to the GTAP standard model, is that of a comparative-static global model of a barter economy, i.e. with fixed exchange rates. Alternatively, the model can also be run in recursive-dynamic fashion and in that case combined with the features of “G-RDEM: Long-term baseline generation and analysis”, pages 180ff.. Equally, a single regional model can be directly derived from the equation system, and the single region to run can be selected by the GUI. The code and GUI also support deriving a partial equilibrium model where prices for some commodities and related intermediate and factor demand of the activities procuring these commodities are fixed along with total income and the model is solved for the remaining endogenous commodity markets.

---

<sup>8</sup> Roson, R., Britz, W. (2021): Simulating long run structural change with a dynamic general equilibrium model, *Int. J. Computational Economics and Econometrics* 11(4): 368-404; Britz, W., Roson, R. (2019): G-RDEM: A GTAP-Based Recursive Dynamic CGE Model for Long-Term Baseline Generation and Analysis, *Journal of Global Economic Analysis* 4(1): 50-96

Use of the options, extensions and modules discussed above does not require additional coding efforts; they can be directly activated from the interface or by using non-default parameter files. The set-up of the code should render it relatively straightforward to implement additional modules in a modular fashion. It should be noted here that the modules are to a large extent inter-operable, which allows setting up a model with captures simultaneously the features of GTAP-AGR, GTAP-AEZ and GTAP-E, treats some sectors a la Melitz, uses different household types etc..

Data input for the model is based on downloading zip-archives from the GTAP-center with the GTAP Data Base and additional data (land use, migration, MRIO etc.) with full detail (regions, products, sectors, factors) for a specific version and year. The content is then re-organized into a set of GDX containers into one folder (see section “Direct input from GDX version of GTAP Data Base”, page 276ff.), such that multiple data base (different release version, years, standard or GTAP-Power) can be used in parallel. The data base driver of CGEBox (see section “GTAPAGG: The data base generation utility of CGEBox”, page 283ff.) allows to define aggregation and to steer further details, such performing a split. The outcome is one single file used by the simulation model for benchmarking and simulation which can also be exchanged between CGEBox users. The modeling platform is supported by a Graphical User Interface, see section “GUI”, 388ff. which also details how to install the system.

Technically, the model is set-up as a constraint system of equations in levels. The equations and variables are to a largest extent paired in MCP style which also helps debugging the model especially during further developments. Due to the pairing, the model can be solved alternatively as a Mixed Complementary System (MCP) which allows exploiting features as such endogenous quota rents. An explorative version uses the so-called “Extended Mathematical Programming” (EMP) package of GAMS to optimize various taxes / subsidies such as to maximize a (global) welfare function, taking the equations of the CGE as constraints.

**Table 1:** Modules and extensions

Module	Status	Remarks
Data filter	Tested, not used with GEMPACK based model	Optionally removes small transactions from SAMS / trade matrices while maintaining closely important totals. Thought to support model applications with highly dis-aggregated data bases. Draws on code by T. Rutherford
GTAP-Standard	Tested for exact replication	With extensions from ENVISAGE such as non-diagonal make matrix, CET on export side
Completely flexible nesting of production functions	Tested, based on set-definitions	Should allow to quickly generate variants of the standard GTAP model currently available which differ in nesting of factors / intermediate demands
Completely flexible nesting for factor supply	Tested, based on set-definitions	Should allow to quickly generate variants of the standard GTAP model currently available which use nested CET structures to describe factor supply
Completely flexible of sub-nests under final demand	Tested, based on set-definitions, not available for the trade margins	Should allow to quickly generate variants of the standard GTAP model currently available which use CES-subnests under the top-level final demand equation
LES/CD/MAIDADS functions for final	Partly found in myGTAP	Parameters derived from CDE parameterization, respectively econometrically estimated (MAIDADS)

Module	Status	Remarks
demand		
GTAP-AEZ	Operational, not tested for exact replication	With land supply elasticities for natural land cover, differentiated by land cover, additive CET, optionally additional CET nests, carbon stock accounting
GTAP-AGR	Operational, not tested for exact replication	Applicable also to regional dis-aggregation different from original GEMPACK implementation, uses the flexible nesting approach, adjusts to sectoral detail.
GTAP-E	Operational, not tested for exact replications	Based on flexible nesting approach, adjusts to sectoral detail, uses the additional nesting for power generation from GTAP-Power if the detail is available in the data base
GTAP-Melitz: Heterogenous firm module	Operational, similar to GTAP-HET	Includes a fixed cost nest based on the flexible nesting approach, sector coverage can be flexibly chosen. Can also be turned in a Krugman specification
MRIO extension	Operational	Differentiation of bi-lateral import demands and tariffs by total intermediate demand and each final demand agent
myGTAP module	Operational, not tested for exact replication	Removes the regional household, supports multiple private households which manage their own factor stocks, can be linked to household surveys for 19 countries and post-model micro-simulation, subsistence module where households at sub-regional level consume (larger part) of own production of selected activities
GMIG	Operational	Bi-lateral migration of labour, remittances
Aggregate Armington aggregator for intermediate demand	Operational, not part of standard GTAP model or variants thereof available from GTAP	Domestic and import shares for intermediate demand and related tax rates are not sector specific, removes a large share of equations
Aggregate Armington aggregator for all agents	Operational, not part of standard GTAP model or variants thereof available from GTAP, found in GLOBE	Domestic and import shares for intermediate demand and related tax rates are not agent specific, removes a large share of equations
Third level nest for Armington / CET	Small shares can be treated as a Leontief under the second nest, found in GLOBE	Might avoid numerical problems with tiny shares
Armington with commitment terms	Under development	Allows to model emerging trade flows
Spatial Price Equilibrium instead	Under development	Allows to model emerging trade flows

Module	Status	Remarks
of Armington		
Tariff Line Module	Operational	Allows a CET/CES dis-aggregation of selected bi-lateral trade links including an explicit TRQ mechanism
Capital vintages		Draws on similar mechanism used in recursive-dynamic CGEs which differentiate vintage from new capital  Differentiation between depreciated capital stock (not mobile) and gross investment (mobile), capital stock can be rendered endogenous in comparative static mode
Sub-regions	Tested, not available in standard GTAP model	Breaks down production decisions and factor markets to sub-regional level, currently data available for NUTS2 administrative regions for Europe
Post-model reporting	Operational, EV decomposition as in GEMPACK	Generates SAM like structure, calculates world totals, regional and sectoral totals based on additional GTAP agg file, welfare decomposition etc., feeds into GUI exploitation tools, Trade in VA indicators, welfare decomposition, nutritional indicators, CO2 and Non-CO2 emissions relevant for climate change, air pollution emissions, energy accounting in oil equivalents, micro-simulation module using FAO households surveys for current 19 countries
Single region mode	Operational, not available in standard GTAP model	Fixes import prices and let export demand react to lower Armington nest at export destinations.
Recursive dynamic mode, G-RDEM: Long-term baseline generation and analysis	Operational, not tested for compliance with GDyn	Econometrically estimated MAIDADS demand system, endogenous saving rates, differentiated productivity growth, debt accumulation from foreign savings, income dependent industry cost shares and expenditure shares for government and investors, possibility to fix calorie intakes to an empirical relation to per-capita demands, crop yields and crop land expansion can be fixed to exogenous projections
Partial Equilibrium closure	Operational	Solves only one or some commodity markets and all factor markets, regional or household income exogenous
CO2, Non-CO2 Accounting	Operational	Can be combined with taxation or CO2 trading permits, considers non-CO2 emissions and carbon stock changes; difference between reported and taxed Co2 equivalent emisisions possible
Air pollution emissions	Operational	
Energy use accounting	Operational	Only post model, not model equations

Module	Status	Remarks
Climate Change Damages	Operational	Impacts on labour productivity, health, tourism, energy demands, crop yields and available land from sea level, based on data from Sartori & Roson 2016

## 2. Basic model equations

The basic model equations are to a large extent identical to Van der Mensbrugghe, D. (2018). “The Standard GTAP Model in GAMS”, Version 7. Journal of Global Economic Analysis, 3(1), 1-83. We refer below to the equations in that complete and excellent documentation (equation (x) in VDM 2018) and report differences where applicable. We refrain from repeating all equations in mathematical notation – the reader is invited to refer to VDM 2018. Instead, we insert screen shots of the actual code which we comment to ease understanding the technical implementation of the model.

### 2.1 Core sets

The equations of the basic model are comprised in the file “model.gms” and discussed in the following. The following general sets are used:

**Table 2: Core sets used in model equations**

Set name	Description
r, rp	Regions
rnat, natl	nations (to differentiate from sub-regions in the case the NUTS2 level is active)
disr	Nations which are dis-aggregated to sub-regions
subr	sub-regions (only populated in the case the NUTS2 level is active)
aa	Armington agents (sectors, private household, government, savings, transport modes)
a	production activities
i, j, k	products
t	time
m	mode of transport
f	factors
fm	mobile factor (fully mobile or sluggish)
fnm	immobile factor, i.e. sector specific
h	households
gov	government (single item)
Inv	investment (single item)
fd	final demand groups, used in demand nests
dNest	demand nests
tNest	technology nests
fNest	factor supply nests

Note that the lists of regions, activities, products and factors depend on the version of the GTAP Data Base used and the chosen aggregation. The list of demand, technology and factor supply nests is equally dynamic, depending on which modules are active and/or on additional nests introduced by user provided files. The myGTAP extensions might introduce several private households in the set  $h$  which is otherwise a singleton. In the standard layout, i.e. without using the NUTS2 extensions, all regions are defined as nations and the list of dis-aggregated regions  $disr$  is empty.

As the model might run as a single region or as a partial equilibrium model or recursive dynamically, dynamic sets are used to indicate for which regions, product, activities and time points the equations in the current model instance should be generated:

$rs$         regions in current solve  
 $ts$         time point in current solve  
 $aIn$        activities in current model  
 $iIn$        products in current model

Furthermore, to support sparsity, i.e. to avoid that equations and variables are only generated for non-empty items, a larger set of parameters are in use which serves as flags. The most important ones are listed here:

$vaFlag(r,a)$     Indicates if value added for region  $r$  and activity  $a$  is non-empty  
 $ndFlag(r,a)$     Indicates if intermediate demand composite for region  $r$  and activity  $a$  is non-empty  
 $xpFlag(r,a)$     Indicates if activity  $a$  for region  $r$  is non-empty  
 $xfFlag(r,f,a)$    Indicates if primary factor  $f$  is used by activity  $a$  in region  $r$   
 $xaFlag(r,i,aa)$  Indicates if Armington agent  $aa$  demands product  $i$  in region  $r$   
 $xwFlag(r,i,rr)$  Bilateral trade flag

Given these examples, the names of the other flags should be hopefully self-explanatory

## 2.2 Reminder on CES and CET equations

Many relations in the model are based on Constant Elasticity of Substitution (CES) and Constant Elasticity of Transformation (CET) functions, using constant returns to scale, i.e. the homothetic case. That implies that average costs (respectively returns) are equal to marginal costs (respectively returns). As a consequence, the functional form can only be used to find per unit minimal cost (respectively per unit maximal revenue), but not to define the total amount to demand or to transform. For each CES or CET nest comprising  $n$  inputs or outputs, there are  $n+1$  equations comprised in the model: one equation which defines the average price and  $n$  equations which define the  $n$  demand or output quantities under the nest.

The  $n$  quantity equations have always the same structure, shown below for the CES case which defines the demand for  $x_i$  as follows. A given share parameter  $\alpha_i$  defined at the benchmark is multiplied with the total quantity  $y$  to distribute. This total, as explained above, must be defined by another mechanism in the model. The resulting share on the total is then updated by multiplying it with the price relation

between the average or composite price (index)  $\tilde{p}$  and the price  $p_i$ , exponent the substitution elasticity  $\sigma$ . Note that settings the substitution elasticity to zero yields the Leontief case where the original physical shares stay constant. A shifter variable  $\lambda$  can be used to update preferences or the cost structure to reflect non-Hicks neutral technical progress. The reader should note again that the demands are homothetic: increasing the total  $y$  by 1% will increase also all individual demands  $x_i$  by 1% if prices and shifters stay constant.

$$x_i = \alpha_i y \left( \frac{\tilde{p}}{p_i} \right)^\sigma \lambda^{1-\sigma}$$

One might assume that the easiest way to define the average or composite price (index)  $\tilde{p}$  is to use value exhaustion, i.e. by introducing an equation  $\sum_i x_i p_i = y \tilde{p}$ . But this exhaustion equation requires the individual demands  $x_i$  being known. This is only possible if the average price  $\tilde{p}$  is known as well as it is used in the demand equation for  $x_i$ , while this average price in turn is implicitly defined by the revenue exhaustion equation. That circular relation makes it hard for the solver to find a simultaneous solution for the  $n+1$  equations.

Therefore, perhaps astonishingly, the far more non-linear dual price aggregator equation is used in most CGE models and also in CGEBox:

$$\tilde{p} = \left[ \sum_i \alpha_i \left( \frac{p_i}{\lambda_i} \right)^{1-\sigma} \right]^{\frac{1}{1-\sigma}}$$

The advantage is that the average price  $\tilde{p}$  can be defined from the individual prices, only, without requiring the total  $y$  or the individual demands  $x_i$ . That reflects the homothetic nature of the function where the total does not affect the optimal shares. Once the average price is solved for, the other  $n$  equations defined the quantities can be solved independently from each other.

The reader might note that the dual price aggregator is undefined for the CD case where the substitution elasticity is unity as the rightmost exponent will become undefined. The CD case leads to the following dual price aggregator:

$$\tilde{p} = \prod_i \left( \frac{p_i}{\alpha_i \lambda_i} \right)^{\alpha_i}$$

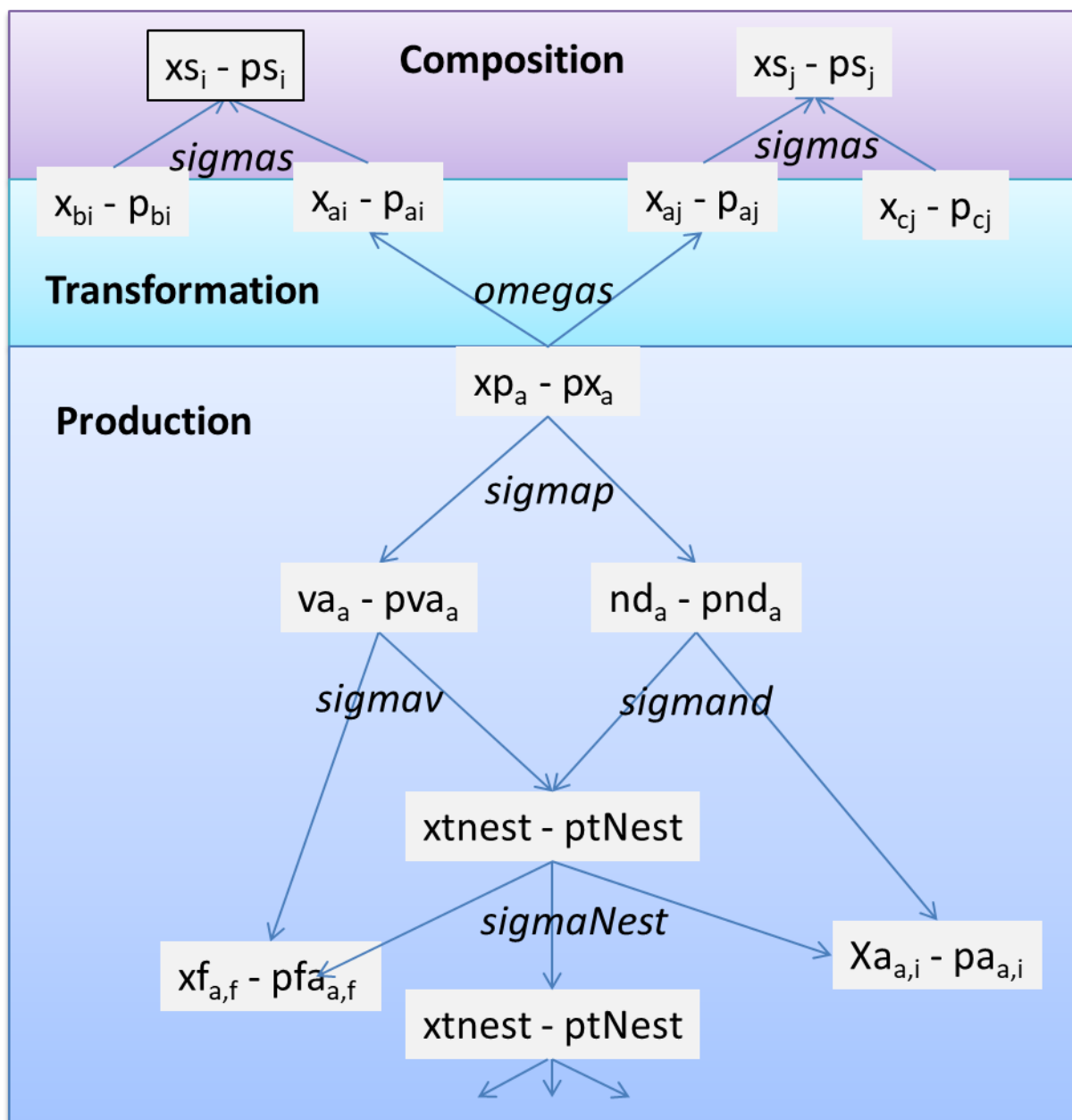
As most CES/CET nests in CGEBox allow for the CD case, both dual price aggregators are found in the model and switched on/off depending on the substitution respectively transformation elasticity. Finally, the case of infinite substitution or transformation cannot be handled by the equations above. They require instead a simply linear aggregator to add the components to the aggregate and the law of one price, i.e. the average price is equal to each component price.

The CET case looks structurally identical, but is not shown here.



## 2.3 Overview on the supply side

The following Figure 1 depicts the quantity and price variables as well as the substitution and transformation elasticities used on the supply side of the CGE model. The bottom part is defined for the production activities  $a$  with total output denoted with  $x_p$  and related price  $p_x$ . It is composed of a value added composite  $va$  and an intermediate demand composite  $nd$ . The value added composite  $va$  combines primary factors  $f$  and potentially technology sub-nests  $tNest$ . The intermediate demand composite  $nd$  combines intermediates defined as Armington demands  $x_a$  of the activities and potentially technology sub-nests  $tNest$ . Technology sub-nests can combine other sub-nests, primary factors and intermediates in a nested fashion. Note firstly that primary factors or intermediates can be present in different shares in sub-nests and, secondly, that if the Melitz / Krugmann specification is used for a sector, fix costs are present in a separate sub-nest which does not contribute to  $x_p$ .



**Figure 1: Overview on production function nesting**

The output of the activities  $x_p$  can be transformed for the non-diagonal make case to different commodities  $x$  as shown in the middle box. If several activities produce the same commodity, the different  $x$  can be combined in supply  $xs$  based on a CES aggregator as shown in the top box.

For the upper two boxes, the code supports the case of finite and infinite transformation respectively substitution where the infinite case implies a linear aggregation and the case of one price. For the production nests, only finite transformation is supported including the CD case.

## 2.4 Production block

The model is set up to work with non-diagonal make matrices where one activity might produce several outputs and one output might be produced by several activities. The production block therefore is defined for activities  $a$  and not for the outputs  $i$ . Furthermore, in case regions are dis-aggregated to sub-regions, the production function is defined for these dis-aggregated regions. Accordingly, nations

which are dis-aggregated to sub-regions *disr* are excluded from these equations. The production reflects the “Flexible nesting” approach which allows introducing CES-subnests under the value and the intermediate composite nests, or under other CES-subnests.

The nested production function for each activity *a* comprises a **top nest** which combines a value added *va* and intermediate demand *nd* composite with a substitution elasticity of  $p\_sigmap^9$ . The production frontier can be shifted with the variable *xp*. The top nest is represented by its dual price aggregator in the equation *e\_px*. This equation considers three cases which are shown below: (1)  $p\_sigmap$  is non-zero and different from unity with leads to the usual dual price aggregator for the CES case, (2) the CD case where  $p\_sigmap$  is unity with a different dual price aggregator and (3) the Leontief case with  $p\_sigmap$  equal to zero. The latter is introduced for technical reasons, only, as it will generate a more compact Jacobian in GAMS.

The price for the intermediate composite is called *pnd* and that for the valued added one *pva*. The related technology shifters are the variables *lambdand* and *lambdava* while the share parameters are called *p\_and* and *p\_ava*.

Note that the unit cost price *px* might be substituted out from the model in the diagonal make case based on the macro *mm\_px*. The equation is identical to VDM 2018, equation (3).

```
*
* --- Unit cost definition (also zero profit condition) equation (3) in VDM 2018
* (note: that equation is used with a non-diagonal make matrix)
*
e_px(rs,aIn(a),ts) $ (xpFlag(rs,a)
    $iftheni.notSubs "%subsOutVars%"=="on"
    $ (not diag(a))
    $endif.notSubs
    $ (not disr(rs))..
    xp(rs,a,ts)*mm_px(rs,a,ts) ==
*
* -- Non Leontief case with substitution between VA and ND
*
    ( ( p_and(rs,a,ts)*(pnd(rs,a,ts)/lambdand(rs,a,ts))**(1-p_sigmap(rs,a))
      + p_ava(rs,a,ts)*(pva(rs,a,ts)/lambdava(rs,a,ts))**(1-p_sigmap(rs,a)) ) ** (1/(1-p_sigmap(rs,a)))
    ) $ (p_sigmap(rs,a) $ (p_sigmaP(rs,a) ne 1))
*
* -- CD case with substitution between VA and ND, p_sigmap=1
*
+ ( p_axpCD(rs,a,ts)
    * ((pnd(rs,a,ts)/(p_and(rs,a,ts)*lambdand(rs,a,ts))**p_and(rs,a,ts)) $ p_and(rs,a,ts) + 1 $ (not p_and(rs,a,ts)))
    * ((pva(rs,a,ts)/(p_ava(rs,a,ts)*lambdava(rs,a,ts))**p_ava(rs,a,ts)) $ p_ava(rs,a,ts) + 1 $ (not p_ava(rs,a,ts)))
    ) $ (p_sigmap(rs,a) eq 1)
*
* -- Leontief case (only introduced to ease work for solver), as in standard GTAP model
*
+ (
    ( p_and(rs,a,ts)*(pnd(rs,a,ts)/lambdand(rs,a,ts))
      + p_ava(rs,a,ts)*(pva(rs,a,ts)/lambdava(rs,a,ts)) ) ) $ (not p_sigmap(rs,a))
;
*
```

The *mm\_px* macro is in the usual case equal to the *m\_xp* macro which is shown below. It directly uses the product specific supply price *ps* corrected for production taxes *prdtx* in case of a diagonal make relation for that activity, as depicted by the flag *diag(a)*. If the activity produces several outputs, its unit cost price *px* is used instead. Note that the *xFlag* indicates which outputs *k* are produced by activity *a* in region *r*:

```
$ifc %nDiag%<0 $macro m_px(r,a,t) ((px(r,a,t) $ (not diag(a)) + sum(k $ xFlag(r,a,k), ps(r,k,t)/(1 + prdtx(r,a,t))) $ diag(a)) \
$ifc %nDiag%=0 $macro m_px(r,a,t) (sum(k $ xFlag(r,a,k), ps(r,k,t)/(1 + prdtx(r,a,t))) \
$iftheni.subregs "%modulesGTAP_NUTS2%"=="on"
    $ifc %ModulesCO2_Emissions%==on - sum(rr_rr(r,rsnat), emis(rsnat,t) * emiaCo2eq(rsnat,a,t)) \
$else.subregs
    $ifc %ModulesCO2_Emissions%==on - emis(r,t) * emiaCo2eq(r,a,t) \
$endif.subregs
```

<sup>9</sup> Generally, all GAMS parameters carry a *p\_* prefix to distinguish them from variables. Parameters cannot become subject to a closure swap in GAMS.

The demand for the **value added** composite  $va$  is defined in the equation  $e\_va$ , VDM 2018, Equation (1). It comprises the same symbols as shown above in the top level unit cost definition. Note that the equation treats the Leontief case where  $p\_sigmap$  is zero differently by removing the prices from the equation which can speed up solution.

```
*
* --- Demand for aggregate value added (equation (1) in VDM 2018)
*
e_va(rs,aIn(a),ts) $ vaFlag(rs,a) ..
  va(rs,a,ts)/va.scale(rs,a,ts) =e= p_ava(rs,a,ts) * m_xp(rs,a,ts)/va.scale(rs,a,ts)
  * [ { (mm_px(rs,a,ts)/pva(rs,a,ts))**p_sigmap(rs,a) } $ p_sigmap(rs,a) + 1 $ (not p_sigmap(rs,a)) ]
  * (axp(rs,a,ts)*lambdava(rs,a,ts))**(p_sigmap(rs,a)-1) ;
```

Note that the  $e\_va$  equation is scaled with the scale field of the value added demand  $va.scale$ . Scaling factors are present basically in all equations relating to quantities or volumes to ease automated scaling by the solver and provide a more useful interpretation of the relative and absolute tolerances used by the solver.

The relevant activity output quantity is defined in the macro  $m\_xp$ . It uses directly the commodity supply  $xs$  in case of a diagonal make matrix for that activity, i.e.  $diag(a)$  is not zero, otherwise, it introduces the activity output  $xp$  in the equation. Note the symmetry with the  $m\_px$  macro shown above for the output price.

```
$evalGlobal nDiag card(a)-card(i)
$$ife %nDiag%<>0 $macro m_xp(r,a,t) ( xp(r,a,t) $ (not diag(a)) + sum(k $ xFlag(r,a,k), xs(r,k,t)) $ diag(a) )
$$ife %nDiag%=0 $macro m_xp(r,a,t) sum(k $ xFlag(r,a,k), xs(r,k,t))
```

The equation  $e\_nd$ , VDM 2018, Equation (2), identically structured as the  $e\_va$  equation above, drives the demand for the **intermediate demand** composite:

```
*
* --- Demand for intermediate demand bundle (equation (2) in VDM 2018)
*
e_nd(rs,aIn(a),ts) $ ndFlag(rs,a) ..
  nd(rs,a,ts)/nd.scale(rs,a,ts) =e= p_and(rs,a,ts) * m_xp(rs,a,ts)/nd.scale(rs,a,ts)
  * [ { (mm_px(rs,a,ts)/pnd(rs,a,ts))**p_sigmap(rs,a) } $ p_sigmap(rs,a) + 1 $ (not p_sigmap(rs,a)) ]
  * (axp(rs,a,ts)*lambdaNd(rs,a,ts))**(p_sigmap(rs,a)-1) ;
```

The demand for **primary factors**  $xf$  by each activity depends on a shifter variable  $lambdaf$ , the share parameter  $p\_af$ , total value added demand  $va$  and the price relation between the price of the value added bundle  $pva$  and the sector specific factor price  $pfa$ , defined via the macro  $m\_pfa$ , exponent the substitution elasticity  $p\_sigmav$ :

```
*
* --- Demand for factors, equation (4) in VDM 2018, with extension for commodity nests
*
e_xf(rs,f,aIn(a),ts) $ (xfFlag(rs,f,a) $ ( p_af(rs,f,a,ts)
  or sum(tNest_f_a(tNest,f,a), p_afNest(rs,f,a,tNest,ts)) ) $ (not disr(rs)) ) ..
  xf(rs,f,a,ts)/xf.scale(rs,f,a,ts)
  =e=
*
* --- demand for factors in VA nest
*
  p_af(rs,f,a,ts)*va(rs,a,ts)/xf.scale(rs,f,a,ts)*(pva(rs,a,ts)/m_pfa(rs,f,a,ts))**p_sigmav(rs,a)
  * lambdaf(rs,f,a,ts)**(p_sigmav(rs,a)-1)
*
* --- demand for factor inside technology nest
*
+ sum(tNest_f_a(tNest,f,a) $ p_afNest(rs,f,a,tNest,ts),
*
* --- factor cost share times demand for technology nest
*
  p_afNest(rs,f,a,tNest,ts)*xtNest(rs,tNest,a,ts)/xf.scale(rs,f,a,ts)
*
* --- relative price impact (in case sigmndNest is not 0)
*
* [ { (pTNest(rs,tNest,a,ts)/m_pfa(rs,f,a,ts))**p_sigmaNest(rs,tNest,a) } $ p_sigmaNest(rs,tNest,a)
  + 1 $ (not p_sigmaNest(rs,tNest,a)) ]
  * lambdaf(rs,f,a,ts)**(p_sigmaNest(rs,tNest,a)-1)) ;
```

That equation differs from VDM, equation (4) by the inclusion of the technology nests:

$$\begin{aligned}
 XF_{r,f,a}^d = & \alpha_{r,f,a}^f VA_{r,a} \left( \frac{PVA_{ra}}{PF_{r,f,a}^a} \right)^{\sigma_{r,a}^v} \left( \lambda_{r,f,a}^f \right)^{\sigma_{r,a}^v - 1} \\
 & + \alpha_{r,f,a}^{tNest} XTNEST_{r,tNest,a} \left( \frac{PNEST_{r,tNest,a}}{PF_{r,f,a}^a} \right)^{\sigma_{r,a}^{tNest}} \left( \lambda_{r,f,a}^f \right)^{\sigma_{r,a}^{tNest} - 1}
 \end{aligned} \quad (4^*)$$

The second part of the equation is not part of the standard model and only active if technology nests are used; it is described in the section “Flexible nesting”. It comprises the same elements: share parameters inside the nests `p_atNest`, the composite demand for the nest `xtNest`, the price relation which now uses the average price of the nest `pTNest` and the substitution elasticity `p_sigmaNest`. Note that the demand from technology nests is added, i.e. the model supports a layout where several technology nests and the value added nest can demand the same factor (or intermediate composite, see below) in different shares.

The dollar conditions might warrant some comments. The first one is the flag `xfFlag` indicating that the activity `a` is using factor `f`, while the second dollar condition ensures that also share parameters are present, either in the value added nests and/or some technology nests. That double security might catch against cases where due to numerical thresholds, share parameters are set to zero despite the fact that there some tiny quantity is reported in the SAM.

The **value added composite price** `pva` is defined in the `e_pva` equation. It differentiates the cases where the substitution elasticity `p_sigmay` between primary factors is (1) not unity, i.e. CES or Leontief, (2) unity, i.e. the CD case:

```

*
e_pva(rs,aIn(a),ts) $ vaFlag(rs,a) ..
    pva(rs,a,ts) =e=
*
*   --- CES case (or Leontief)
*
*   [ (
*
*       --- contribution of factors to top VA nests
*
*       sum(f $ p_af(rs,f,a,ts), (p_af(rs,f,a,ts)*(m_pfa(rs,f,a,ts)/lambdaf(rs,f,a,ts))**(1-p_sigmay(rs,a)))
*
*       --- contribution of technology nests to top VA nests (not part of standard GTAP model)
*
*       + sum(tNest_n_a("VA",tNest,a) $ p_atNest(rs,tNest,a,ts),
*           p_atNest(rs,tNest,a,ts)*(pTNest(rs,tNest,a,ts)/lambdan(rs,tNest,a,ts))**(1-p_sigmay(rs,a)))
*
*       )** (1/(1-p_sigmay(rs,a))) ] $ (p_sigmay(rs,a) ne 1)
*
*   --- CD case with p_sigmay == 1
*
*   + [ p_axVACD(rs,a,ts)
*
*       --- contribution of factors to top VA nests
*
*       * prod(f $ p_af(rs,f,a,ts), (m_pfa(rs,f,a,ts)/(lambdaf(rs,f,a,ts)*p_af(rs,f,a,ts))**p_af(rs,f,a,ts))
*
*       --- contribution of technology nests to top VA nests (not part of standard GTAP model)
*
*       * prod(tNest_n_a("VA",tNest,a) $ p_atNest(rs,tNest,a,ts),
*           (pTNest(rs,tNest,a,ts)/(p_atNest(rs,tNest,a,ts)*lambdan(rs,tNest,a,ts))**p_atNest(rs,tNest,a,ts))
*
*       ] $ (p_sigmay(rs,a) eq 1);

```

That equation differs from VDM 2018, equation (5) by the inclusion of the technology nests:

$$PVA_{r,a} = \left[ \sum_f \alpha_{r,f,a}^f \left( \frac{PF_{r,f,a}^a}{\lambda_{r,f,a}^f} \right)^{1-\sigma_{r,a}^v} + \sum_{tNest \in VA} \alpha_{r,f,a}^{tNest} \left( \frac{PTNEST_{r,tNest,a}}{\lambda_{r,f,a}^{tNest}} \right)^{1-\sigma_{r,a}^v} \right]^{\frac{1}{1-\sigma_{r,a}^v}} \quad (5^*)$$

Note that the flexible nesting approach allows to link nests into the value added composite such that both sums and products of the individual factors and over nests are introduced in the equation.

A similarly structured equation `e_pnd` defines the **intermediate composite price** `pnd`. It is driven by the input coefficients `p_io` and their activity specific price defined via a macro `m_pa` and product and activity specific technology shifters `lambdaio`, again captured by a macro. Note that the coefficients `p_io` describe shares inside the intermediate nest, and not relative to total output. As the standard GTAP model uses a Leontief representation for intermediate demand, the case where the substitution elasticity `p_sigmand` is zero is separated out here as well, such that we find three blocks (CES, CD and Leontief). Separating out the Leontief case reduces model complexity as the solver will define a linear instead of a non-linear price aggregator.

Note that we consider again the cases where the intermediate demand is driven by the intermediate composite (standard model) and/or by technology nests.

```

*
* e_pnd(rs,aIn(a),ts) $ ndFlag(rs,a) ..
*
pnd(rs,a,ts) =e=
*
* --- CES case
*
sum(r_r(rs,rsNat),
[ (
*
* --- aggregate over all intermediate inputs not assigned to a technology tests
*
sum(i $ p_io(rs,i,a,ts),
p_io(rs,i,a,ts) * (m_pa(rsNat,i,a,ts) / lambdaio(rs,i,a,ts)) ** (1-p_sigmand(rs,a)))
*
* --- add demand for technology nests linked into the top ND undie
*
+ sum(tNest_n_a("ND",tNest,a) $ p_atNest(rs,tNest,a,ts),
p_atNest(rs,tNest,a,ts) * (ptNest(rs,tNest,a,ts) / lambdaN(rs,tNest,a,ts)) ** (1-p_sigmand(rs,a)))
** (1/(1-p_sigmand(rs,a)))
] $ (p_sigmand(rs,a) $ (p_sigmand(rs,a) ne 1))

*
* --- CD case with p_sigmand == 1
*
+ [ p_axNdCD(rs,a,ts) *
*
* --- aggregate over all intermediate inputs not assigned to a technology tests
*
{
prod(i $ p_io(rs,i,a,ts),
(m_pa(rsNat,i,a,ts) / (p_io(rs,i,a,ts) * lambdaio(rs,i,a,ts))) ** p_io(rs,i,a,ts))
$ sum(i $ p_io(rs,i,a,ts), 1) + 1 $ (not sum(i $ p_io(rs,i,a,ts), 1)) }
*
* --- add demand for technology nests linked into the top ND undie
*
* {prod(tNest_n_a("ND",tNest,a) $ p_atNest(rs,tNest,a,ts),
(ptNest(rs,tNest,a,ts) / (p_atNest(rs,tNest,a,ts) * lambdaN(rs,tNest,a,ts))) ** p_atNest(rs,tNest,a,ts))
$ sum(tNest_n_a("ND",tNest,a), p_atNest(rs,tNest,a,ts))
+ 1 $ (not sum(tNest_n_a("ND",tNest,a), p_atNest(rs,tNest,a,ts))) }
] $ (p_sigmand(rs,a) eq 1)

```

```

*
*      --- Leontief case, as in standard GTAP model
*
*      + [
*
*      --- aggregate over all intermediate inputs not assigned to a io-group
*
*          sum(i $ p_io(rs,i,a,ts),
*              p_io(rs,i,a,ts)*(m_pa(rsNat,i,a,ts)/lambdaio(rs,i,a,ts)))
*
*      --- add aggregate intermediate group demands (not part of standard GTAP Model)
*
*      + sum(tNest_n_a("ND",tNest,a) $ p_atNest(rs,tNest,a,ts),
*          p_atNest(rs,tNest,a,ts) * (ptNest(rs,tNest,a,ts)/lambdaN(rs,tNest,a,ts))) $ card(tNest)
*
*      ] $ (not p_sigmand(rs,a))
*  ];

```

This equation thus differs again from VDM 2018 equation (7) due to the inclusion of the technology nests:

$$PND_{r,a} = \left[ \sum_i \alpha_{r,i,a}^{io} \left( \frac{PA_{r,i,a}}{\lambda_{r,f,a}^{io}} \right)^{1-\sigma_{r,a}^v} + \sum_{tNest \in ND} \alpha_{r,i,a}^{tNest} \left( \frac{PTNEST_{r,tNest,a}}{\lambda_{r,i,a}^{tNest}} \right)^{1-\sigma_{r,a}^{nd}} \right]^{1/(1-\sigma_{r,a}^{nd})} \quad (7^*)$$

The dual price aggregator equation e\_ptNest for technology nests is depicted below. It considers the possible components: intermediates with the related share parameter p\_ioNest, primary factors with their share parameter p\_afNest and finally sub-nests with share parameters atNest. The price and shifters used for intermediates and primary factors are identical to those described above for the value added and intermediate composite nests. Due to the different dual price aggregator necessary for the CD case, the equation comprises two blocks.

```

e_ptNest(rs,tNest,aIn(a),ts) $ xtFlag(rs,tNest,a) ..
ptNest(rs,tNest,a,ts) =e=
(
    sum(r_r(rs,rsNat),
    { [
*
*      --- aggregate over intermediate inputs assigned to technology nest
*
*          sum(tNest_i_a(tNest,i,a) $ p_ioNest(rs,i,a,tNest,ts),
*              p_ioNest(rs,i,a,tNest,ts)
*              * (m_pa(rsNat,i,a,ts)/lambdaio(rs,i,a,ts))**(1-p_sigmaNest(rs,tNest,a)))
*
*      --- aggregate over factors assigned to technology nests
*
*          + sum(tNest_f_a(tNest,f,a) $ p_afNest(rs,f,a,tNest,ts),
*              p_afNest(rs,f,a,tNest,ts)
*              * (m_pfa(rs,f,a,ts)/lambdaf(rs,f,a,ts))**(1-p_sigmaNest(rs,tNest,a)))
*
*      --- aggregate over sub technology nests assigned to that technology nests
*
*          + sum(tNest_n_a(tNest,tNest1,a) $ p_atNest(rs,tNest1,a,ts),
*              p_atNest(rs,tNest1,a,ts)
*              * (ptNest(rs,tNest1,a,ts)/lambdaN(rs,tNest1,a,ts))**(1-p_sigmaNest(rs,tNest,a)))
*              ]**(1/(1-p_sigmaNest(rs,tNest,a)))
    )
) $ (p_sigmaNest(rs,tNest,a) ne 1)

```



```

*
* ---- CES case with p_sigmaNest == 1 (shifter is missing ...)
*
* + [
*
* --- aggregate over intermediate inputs assigned to technology nest
*
*     prod(tNest_i_a(tNest,i,a) $ p_ioNest(rs,i,a,tNest,ts),
*          (m_pa(rsNat,i,a,ts)/(p_ioNest(rs,i,a,tNest,ts)*lambdaio(rs,i,a,ts)))*p_ioNest(rs,i,a,tNest,ts))
*
* --- aggregate over factors assigned to technology nests
*
*     * prod(tNest_f_a(tNest,f,a) $ p_afNest(rs,f,a,tNest,ts),
*          (m_pfa(rs,f,a,ts)/(p_afNest(rs,f,a,tNest,ts)*lambdaf(rs,f,a,ts)))*p_afNest(rs,f,a,tNest,ts))
*
* --- aggregate over sub technology nests assigned to that technology nests
*
*     * prod(tNest_n_a(tNest,tNest1,a) $ p_atNest(rs,tNest1,a,ts),
*          (ptNest(rs,tNest1,a,ts)/(p_atNest(rs,tNest1,a,ts)*lambdan(rs,tNest1,a,ts)))*p_atNest(rs,tNest1,a,ts)/p_axtNest(rs,tNest,a,ts)
*
* ] $ (p_sigmaNest(rs,tNest,a) eq 1)
*
* --- these more complex relations are only tested / introduced if the upper nests into which the nest is linked has not a cost share of unity
*
* )} $ (not sum(tNest_n_a(tNest,tNest1,a) $ (p_atNest(rs,tNest1,a,ts) eq 1),1))
*
*
*
* -- special nest where upper nest consist of only a lower nest
*
* + sum(tNest_n_a(tNest,tNest1,a) $ p_atNest(rs,tNest1,a,ts)*ptNest(rs,tNest1,a,ts))
*   $ sum(tNest_n_a(tNest,tNest1,a) $ (p_atNest(rs,tNest1,a,ts) eq 1),1)
*
*

```

Note that his equation is not part of the GTAP Standard as documented in VDM 2018, it reads in mathematical notation:

$$PNEST_{r,tNest,a} = \left[ \begin{aligned} & \sum_{i \in tNest} \alpha_{r,i,a}^{tNest} \left( \frac{PA_{r,i,a}}{\lambda_{r,i,a}^{tNest}} \right)^{1-\sigma_{r,a}^{tNest}} \\ & + \sum_f \alpha_{r,f,a}^{tNest} \left( \frac{PF_{r,f,a}^a}{\lambda_{r,f,a}^{tNest}} \right)^{1-\sigma_{r,a}^{tNest}} \\ & + \sum_{tNest1 \in tNest} \alpha_{r,tNest1,a}^{tNest} \left( PTNEST_{r,tNest1,a} \right)^{1-\sigma_{r,a}^{tNest}} \end{aligned} \right]^{1/(1-\sigma_{r,a}^{tNest})}$$

The demand for technology nests xtNest is depicted by the equation e\_xtNest shown below. There are three identically structured cases: (1) the nest is linked into the intermediate composite ND, (2) into the value added composite VA or (3) into another technology nests. The three cases differ in the aggregate price used (pnd, pva or ptNest) and the substitution elasticity (p\_sigmand, p\_sigmax or p\_sigmaNest). In all cases, the share parameter is denoted with p\_atNest and the related price with ptNest. Equally, in all cases, in order to reduce complexity for the solver, the price relation is taken out when the substitution elasticity is zero, i.e. the Leontief case.

Note that the third case where the technology nest tNest is part of another nest requires the alias tNest1 which depicts the nest which is higher up in the technology tree.



```

*
* --- Demand for technology nests (not part of standard GTAP model)
*
e_xtNest(rs,tNest,aIn(a),ts) $ (xtFlag(rs,tNest,a) $ (not tNest_n_a("Top",tNest,a))) ..
    xtNest(rs,tNest,a,ts)/xtNest.scale(rs,tNest,a,ts) =e=
*
* --- nest is part of top level intermediate demand
*
+ [ ( p_atNest(rs,tNest,a,ts) * nd(rs,a,ts)/xtNest.scale(rs,tNest,a,ts)
    * [ (pnd(rs,a,ts)/ptNest(rs,tNest,a,ts)) ** p_sigmaNd(rs,a)
        ] $ p_sigmaNd(rs,a) + 1 $ (not p_sigmaNd(rs,a)) ]
    * lambdaN(rs,tNest,a,ts) ** (p_sigmaNd(rs,a) - 1)
    ) $ tNest_n_a("ND",tNest,a)
*
* --- nest is part of top level VA demand
*
+ [ ( p_atNest(rs,tNest,a,ts) * va(rs,a,ts)/xtNest.scale(rs,tNest,a,ts)
    * [ (pva(rs,a,ts)/ptNest(rs,tNest,a,ts)) ** p_sigmax(rs,a) ] $ p_sigmax(rs,a) + 1 $ (not p_sigmax(rs,a)) ]
    * lambdaN(rs,tNest,a,ts) ** (p_sigmax(rs,a) - 1)
    ) $ tNest_n_a("VA",tNest,a)
*
*
* --- nest is part of other nests
*
+ sum(tNest_n_a(tNest1,tNest,a),
    p_atNest(rs,tNest,a,ts) * xtNest(rs,tNest1,a,ts)/xtNest.scale(rs,tNest,a,ts)
    * [ (ptNest(rs,tNest1,a,ts)/ptNest(rs,tNest,a,ts)) ** p_sigmaNest(rs,tNest1,a)
        ] $ p_sigmaNest(rs,tNest1,a) + 1 $ (not p_sigmaNest(rs,tNest1,a))
    * lambdaN(rs,tNest,a,ts) ** (p_sigmaNest(rs,tNest1,a) - 1)
    )
] $ (p_atNest(rs,tNest,a,ts) ne 1)
*
* -- special case where upper nest only comprises the lower one
*
+ [ (nd(rs,a,ts)/xtNest.scale(rs,tNest,a,ts)*lambdaN(rs,tNest,a,ts)**(p_sigmaNd(rs,a)-1)) $ tNest_n_a("ND",tNest,a)
    + (va(rs,a,ts)/xtNest.scale(rs,tNest,a,ts)*lambdaN(rs,tNest,a,ts)**(p_sigmax(rs,a)-1)) $ tNest_n_a("VA",tNest,a)
    + sum(tNest_n_a(tNest1,tNest,a) $ p_atNest(rs,tNest,a,ts),
        xtNest(rs,tNest1,a,ts)/xtNest.scale(rs,tNest,a,ts)*lambdaN(rs,tNest,a,ts)**(p_sigmaNest(rs,tNest1,a)-1))
    ] $ (p_atNest(rs,tNest,a,ts) eq 1);

```

Again, that equation is not part of the GTAP standard model version 7 as documented in VDM 2018. In mathematical notation it reads:

$$\begin{aligned}
 XTNEST_{r,tNest,a} &= \alpha_{r,tNest,a}^{va} VA_{r,a} \left( \frac{PVA_{r,a}}{PNEST_{r,tNest,a}} \right)^{\sigma_{r,a}^{va}} \\
 &+ \alpha_{r,tNest,a}^{nd} ND_{r,a} \left( \frac{ND_{r,a}}{PNEST_{r,tNest,a}} \right)^{\sigma_{r,a}^{nd}} \\
 &+ \sum_{tNest1} \alpha_{r,tNest,a}^{tNest1} XTNEST_{r,tNest1,a} \left( \frac{PNEST_{r,tNest1,a}}{PNEST_{r,tNest,a}} \right)^{\sigma_{r,a}^{tNest1}}
 \end{aligned}$$

More information on the nesting approach can be found above in the section “Flexible nesting”.

The **case of multiple outputs from one activity** is depicted in the equation e\_xq (Equation (8) in VDM 2018). That case is relevant when the flag diag(a) is not unity, i.e. a not diagonal activity. In that case, the p\_omegas transformation elasticity distributes the total output xp to activity specific output x of each product i based on the share parameter gx and the activity specific prices for each product i termed p in relation to average per activity prices defined by the macro m\_pp. In case of infinite transformation, the prices have to be identical:

---

```

*
* --- Convert output into commodities (non-diagonal make matrix with CET for activities, not part of standard GTAP model)
* equation (8) in VDM 2018
*
e_x(rsNat,aIn(a),i,ts) $ (xFlag(rsNat,a,i) $ (not diag(a)) )..
0 =e= (
    x(rsNat,a,i,ts)/xp.scale(rsNat,a,ts)
    - p_gx(rsNat,a,i,ts)*xp(rsNat,a,ts)/xp.scale(rsNat,a,ts) * (p(rsNat,a,i,ts)/m_pp(rsNat,a,ts))**p_omegas(rsNat,a)
) $ (p_omegas(rsNat,a) ne inf)
+ (
    p(rsNat,a,i,ts) $ (p_sigmas(rsNat,i) ne inf)
    + ps(rsNat,i,ts) $ (p_sigmas(rsNat,i) eq inf)
    - (1 + prdtx(rsNat,a,ts))*m_px(rsNat,a,ts)
) $ (p_omegas(rsNat,a) eq inf)
;

```

The related equation e\_xp, equation (9) in VDM 2018, considers these two cases accordingly: with infinite transformation, total output xp is equal to the sum of the commodity outputs, either xs in the diagonal case or x otherwise. With finite transformation, the producer price as defined in the macro m\_pp is derived from the dual CET price aggregator which uses the share parameters gx, the prices p or ps and the transformation elasticities p\_omegas. The choice of p or ps depends on whether consumers differentiate between the same commodities being produced by different activities as depicted by the substitution elasticity p\_sigmas.

```

*
* --- Matching price definition, equation (9) in VDM 2018
*
e_xp(rsNat,aIn(a),ts) $ (xpFlag(rsNat,a)
    $iftheni.notSubs "%subsOutVars%"=="on"
    $ (not diag(a))
    $endif.notSubs
) ..
0 =e=
*
* --- physical balance (inifinite transformation)
*
+ (xp(rsNat,a,ts)/xp.scale(rsNat,a,ts)
    - (
        sum(i$xFlag(rsNat,a,i), x(rsNat,a,i,ts)) $ (not diag(a))
        + sum(i$xFlag(rsNat,a,i), xs(rsNat,i,ts)) $ diag(a)
    ) /xp.scale(rsNat,a,ts)
    ) $ (p_omegas(rsNat,a) eq inf)
*
* --- dual price aggregatorsNat, transformation elasticity is not infinite
*
+ (m_pp(rsNat,a,ts) - (sum(i $ xFlag(rsNat,a,i),
    p_gx(rsNat,a,i,ts) * (
        p(rsNat,a,i,ts) $ (p_sigmas(rsNat,i) ne inf)
        + ps(rsNat,i,ts) $ (p_sigmas(rsNat,i) eq inf)
    ) ** (1+p_omegas(rsNat,a)) ** (1/(1+p_omegas(rsNat,a)))
    )
    ) $ (p_omegas(rsNat,a) ne inf)
;

```

The marco m\_pp which fined the activity specific producer price charges the production taxes prdtx on the unit costs m\_px:

---

```

$macro m_pp(r,a,t) {m_px(r,a,t) * m_prdtx(r,a,t)}

```

The related equations e\_p and e\_ps which aggregate the output of the same commodity from different activities are depicted next. Both are only active in the non-diagonal case (not diag(a) and not diag(i)). The first case depicts the relation between the price of the commodity i outputted by activity a termed p and the average supply price for the commodity ps. They are equal (second line) in case of infinite substitution, otherwise, the second equation defines the average supply price as non-linear weighted average. The first line in e\_p defines for the finite case the share of total supply xs demanded from activity a depicted by x based on the share parameter p\_ax, the price relation and the substitution elasticity p\_sigmas:

```

*
* --- Aggregate commodities from activities (non-diagonal make matrix, demand side, CES, not part of standard GTAP model)
* Equation (11) in VDM 2018
*
e_p(rsNat,aIn(a),i,ts) $ (xFlag(rsNat,a,i) $ (not diag(a)) )..
    0 =e= (
        x(rsNat,a,i,ts)/xp.scale(rsNat,a,ts)
        - p_ax(rsNat,a,i,ts)*xs(rsNat,i,ts)/xp.scale(rsNat,a,ts) * (ps(rsNat,i,ts)/p(rsNat,a,i,ts))**p_sigmas(rsNat,i)
        ) $ (p_sigmas(rsNat,i) ne inf)
    + (
        p(rsNat,a,i,ts) - ps(rsNat,i,ts)
        ) $ (p_sigmas(rsNat,i) eq inf)
    ;

*
* --- Matching price aggregator, quation (11) in VDM 2018
*
e_ps(rsNat,i,ts) $ ( (xs.range(rsNat,i,ts) ne 0) $ xsFlag(rsNat,i) $ (not diag(i)) )..
    0 =e=
    + (xs(rsNat,i,ts)/xs.scale(rsNat,i,ts) - sum(a $ xFlag(rsNat,a,i), x(rsNat,a,i,ts)/xs.scale(rsNat,i,ts)
    ) $ (p_sigmas(rsNat,i) eq inf)
    + (ps(rsNat,i,ts) - sum(a $ xFlag(rsNat,a,i), p_ax(rsNat,a,i,ts)
    ) $ (p_sigmas(rsNat,i) ne inf)
    *p(rsNat,a,i,ts)**(1-p_sigmas(rsNat,i))**(1/(1-p_sigmas(rsNat,i)))
    ;

```

The distribution of output from nations to different sub-regions is described in the section “Integration into the modeling framework” of the chapter “Sub-national dis-aggregation of production and factor markets in CGEBox”.

## 2.5 Factor markets

The **supply of fully mobile or sluggish factors** xft at national level rsNat is depicted by the equation e\_xft in case where the factor supply is not fixed (.range eq 0), equation 69 in VDM 2018. It is driven by the factor price pft relative to the global factor price index pwfact and the factor supply elasticity p\_etaf. If p\_etaf is zero, the price dependent part becomes a constant of unity and xft is fixed to the constant p\_af. Note that endogenous factor supply is not part of the standard GTAP model. Demand for new capital as a new factor is part of the capital vintage module and depicted differently (not shown here).

```

* Aggregate supply of factors (not part of standard GTAP model), equation (69) in VDM 2018
* Iso-elastic factor supply as a function of factor prices pft relative to
* global facto price index, times the factor supply elasticity etaf

e_xft(rsNat,fm,ts) $ (xftEqFlag(rsNat,fm) $ (xft.range(rsNat,fm,ts) ne 0) $ (Not sameas(fm,"newCap"))) ..
    xft(rsNat,fm,ts)/xft.scale(rsNat,fm,ts)
    =e= (p_af(rsNat,fm,ts) * (pft(rsNat,fm,ts)/pwfact(ts))**p_etaf(rsNat,fm))/xft.scale(rsNat,fm,ts);

```

The related **economy wide average factor price** pft is defined by the equation e\_pft, equation (71) in VDM 2018, expanded by potential factor supply nests, which distinguish the sluggish case with a dual price aggregator (first block) and the fully mobile case where the equation ensures market clearing (second block):

```

e_pft(rs, fm, ts) $ (xftEqFlag(rs, fm) and (pft.range(rs, fm, ts) ne 0) and (not discr(rs))) ..
0 =e=
*
*   --- sluggish factor mobility, dual price aggregator
*
[ pft(rs, fm, ts)
  - [
*
*       --- factors directly linked into top CET nest
*
sum(a $ (xftFlag(rs, fm, a) $ ((not sum fNest_a_f(fNest, a, f), 1) or fNest_a_f("xft", a, fm))),
  p_gf(rs, fm, a, ts) * pf(rs, fm, a, ts) ** (1 + p_omegaf(rs, fm)))
*
*       --- subnests linked into top CET nest
*
+ sum(fNest_n_f("xft", fNest, fm) $ p_gfNest(rs, fm, fNest, ts),
  p_gfNest(rs, fm, fNest, ts) * pfNest(rs, fm, fNest, ts) ** (1 + p_omegaf(rs, fm)))
  ] ** (1 / (1 + p_omegaf(rs, fm)))
] $ (p_omegaf(rs, fm) ne inf)
*
*   --- fully mobile factors, uniform prices => physiscal aggregation
*
+ [ xft(rs, fm, ts) / xft.scale(rs, fm, ts)
  - sum(a $ (xftFlag(rs, fm, a) $ ((not sum fNest_a_f(fNest, a, fm), 1) or fNest_a_f("xft", a, fm))), xft(rs, fm, a, ts) / xft.scale(rs, fm, ts)
  - sum(fNest_n_f("xft", fNest, fm) $ p_gfNest(rs, fm, fNest, ts), xftNest(rs, fm, fNest, ts) / xft.scale(rs, fm, ts)
  ] $ (p_omegaf(rs, fm) eq inf)
;

```

In case of sluggish factor supply, the agent specific factor prices net of taxes  $pf$  are aggregated using the dual price aggregator ( $\$ (\text{omegaf}(r, fm) \text{ ne inf})$ ) in the first block, considering factor demand captured by the value added composite and by technology nests. In case of fully mobile factors (the second block), the price is not directly defined in the equation, but rather indirectly via market clearing.

**Sector specific factor prices net of taxes**  $pf$  are directly or indirectly defined in the equation  $e\_pf$  shown below, equation (70) in VDM 2018. It considers five different cases. The first case considers sluggish factor supply where the sector specific factor demand is part of the value added nest. The usual CET distribution logic applies: the supply depends on the share parameter  $p\_gf$  and the total supply  $xft$  as well as on the relation between the price paid in the sector  $pf$  relative to the average one  $pft$ , exponent the transformation elasticity  $p\_omegaf$ . The second case where the factor is part of factor supply nest under sluggish supply is identically structured, with the difference that the average price is now nest specific, i.e.  $pfNest$  as is the transformation elasticity  $p\_omegafNest$ .

Next we have the two cases with fully mobile supply: either in case of economy wide full mobility or full mobility inside in a nest. In both cases, the sector specific price is equal to the average one. The last case depicts immobile factors: here, the default case is that the immobile factor supply elasticity  $p\_etaff$  is zero such that the factor demand  $xf$  must be equal to the given parameter  $p\_gf$ .

```

e_pf(rs, f, aIn(A), ts) $ (xftEqFlag(rs, f, a) $ (pf.range(rs, f, a, ts) ne 0) $ (not discr(rs))) ..
0 =e=
*
*   --- "sluggish" factor supply based on CET function, factor is part of top level nest
*
[ xf(rs, f, a, ts) / xf.scale(rs, f, a, ts)
  - p_gf(rs, f, a, ts) * xft(rs, f, ts) / xf.scale(rs, f, a, ts) * (pf(rs, f, a, ts) / pft(rs, f, ts)) ** p_omegaf(rs, f)
  ] $ (fm(f) $ (p_omegaf(rs, f) ne inf) $ ((not sum fNest_a_f(fNest, a, f), 1) or fNest_a_f("xft", a, f)))
*
*   --- "sluggish" factor supply based on CET function, factor is part of sub-nest
*   (not part of standard GTAP)
*
+ [ xf(rs, f, a, ts) / xf.scale(rs, f, a, ts)
  - sum(fNest_a_f(fNest, a, f),
    p_gf(rs, f, a, ts) * xfNest(rs, f, fNest, ts) / xf.scale(rs, f, a, ts)
    * (pf(rs, f, a, ts) / pfNest(rs, f, fNest, ts)) ** p_omegafNest(rs, fNest, f))
  ] $ (fm(f) $ sum(fNest_a_f(fNest, a, f) $ (p_omegafNest(rs, fNest, f) ne inf), 1))

```

```

*
*      --- fully mobile factor: prices are equal across sectors
*
*      ... factor is linked into top level nest
+ [
    pf(rs,f,a,ts) - pft(rs,f,ts)

    ] $ ((fm(f) $ (p_omegaf(rs,f) eq inf)) $ ( (not sum(fNest_a_f(fNest,a,f),1)) or fNest_a_f("xft",a,f))

*
*      ... factor is linked into sub-nest
+ [
    pf(rs,f,a,ts) - sum(fNest_a_f(fNest,a,f), pfNest(rs,f,fNest,ts))

    ] $ ((fm(f) $ sum(fNest_a_f(fNest,a,f) $ (p_omegafNest(rs,fNest,f) eq inf),1)))

*
*      --- immobile factors fnm, with factor supply elasticity
*
+ [
    xf(rs,f,a,ts)/xf.scale(rs,f,a,ts) - p_gf(rs,f,a,ts)/xf.scale(rs,f,a,ts)

*
    $$iftheni.noExpansionXF "%noExpansionXF%"=="on"

    * [ { (pf(rs,f,a,ts)/pwfact(ts))*p_etaff(rs,f,a) + 1
      - Sqrt( Sqr( (pf(rs,f,a,ts)/pwfact(ts))*p_etaff(rs,f,a) - 1 + Sqr(1.E-4) )) ) /2 }
      $ p_etaff(rs,f,a) + 1 $ (not p_etaff(rs,f,a))]
    $$else.noExpansionXF
    * [ { (pf(rs,f,a,ts)/pwfact(ts))*p_etaff(rs,f,a) } $ p_etaff(rs,f,a) + 1 $ (not p_etaff(rs,f,a))]
    $$endif.noExpansionXF
  ] $ (fnm(f)) ;

```

The **agent specific factor prices tax inclusive** are defined via the equation `e_pfa` and the macro `m_pfa`, equation (73) in VDM 2018:

```

e_pfa(rs,f,aIn(a),ts) $ xfFlag(rs,f,a) ..

pfa(rs,f,a,ts) =E= m_pfa(rs,f,a,ts);

```

The macro `m_pfa` distinguishes the case of finite transformation of factor supply and the case of full factor mobility, i.e. infinite transformation and adds the national tax rates, i.e. subsidy rates `fctts`, tax rates `fcttx` and an economy factor tax shifter `fcttxShift`:

```

$$macro m_pfa(r,f,a,t) { \
*
*      --- factors are immobile or sluggish: price before taxes differs by sector
*
    (pf(r,f,a,t) $ (fnm(f) or (p_omegaf(r,f) ne inf)) \
*
*      --- fully mobile factor: all sector face a uniform price before taxes
*
    + pft(r,f,t) $ (fm(f) $ (p_omegaf(r,f) eq inf))) \
*
*      --- add ad-valorem factor taxes
*
    * m_fctTaxes(r,f,a,t) \
}

```

The factor tax macro `m_fctTaxes` considers the case where sub-national regions are present, which requires an additional inner loop:

```

$$iftheni.subregs "%modulesGTAP_NUTS2%"=="on"

$$macro m_fctTaxes(r,f,a,t) sum(rr_rr(r,rsNat), ( 1 + fctts(rsNat,f,a,t) + fcttx(rsNat,f,a,t) + fcttxShift(rsNat,t) $ fTaxShift(f,a))\
    $$iftheni.CO2 %ModulesCO2_Emissions%==on
        + emisip(rsNat,t) * emifCo2eq(rsNat,f,a,t) \
        $$ifi "%modulesGTAP_AEZ%"=="on" + emisipLnd(rsNat,a,t) $ lnd(f) \
    $$endif.CO2
    )
$$else.subregs

$$macro m_fctTaxes(r,f,a,t) ( 1 + fctts(r,f,a,t) + fcttx(r,f,a,t) + fcttxShift(r,t) $ fTaxShift(f,a))\
    $$iftheni.CO2 %ModulesCO2_Emissions%==on
        + emisip(r,t) * emifCo2eq(r,f,a,t) \
        $$ifi "%modulesGTAP_AEZ%"=="on" + emisipLnd(r,a,t) $ lnd(f)
    $$endif.CO2
$$endif.subregs

```



The “Flexible nesting” approach allows introducing factor supply nests which can also be linked into other factor supply nests. The equation `e_pfNest` defines the average factor price of such a nest. It distinguishes the cases of finite factor transformation in the first block and infinite one in the second. In the first block, the average price `pfNest` is defined via a dual price aggregator, taking the share parameters (`p_gf` for factors and `p_gfNest` for sub-nest), the prices (`pf` for factors and `pfNest` for sub-nests) and the transformation elasticity `p_omegaFNest` into account. The case of infinite transformation is handled by the second block. Here, the price is indirectly defined from the adding up-condition of the factor quantities. This equation is not part of the GTAP standard model.

```

e_pfNest(rs, fm, fNest, ts) $ p_gfNest(rs, fm, fNest, ts) ..
0 =e=
*
*   --- sluggish factor mobility, dual price aggregator for sub-nests
*
*   (pfNest(rs, fm, fNest, ts)
*
*   --- factors linked into sub nest
*
*   - [ sum(fNest_a_f(fNest, a, fm) $ p_gf(rs, fm, a, ts),
*         p_gf(rs, fm, a, ts)*pf(rs, fm, a, ts)**(1+p_omegafNest(rs, fNest, fm)))
*
*   --- subnests linked into sub nest
*
*   + sum(fNest_n_f(fNest, fNest1, fm) $ p_gfNest(rs, fm, fNest1, ts),
*         p_gfNest(rs, fm, fNest1, ts)*pfNest(rs, fm, fNest1, ts)**(1+p_omegafNest(rs, fNest, fm)))
*   ]*(1/(1+p_omegafNest(rs, fNest, fm)))
*
*   ) $ (p_omegafNest(rs, fNest, fm) ne inf)
*
*   --- fully mobile factors, uniform prices => physical aggregation
*
+ (xfNest(rs, fm, fNest, ts)/xft.scale(rs, fm, ts)
  - sum(fNest_a_f(fNest, a, fm) $ p_gf(rs, fm, a, ts), xf(rs, fm, a, ts)/xft.scale(rs, fm, ts)
  - sum(fNest_n_f(fNest, fNest1, fm) $ p_gfNest(rs, fm, fNest1, ts), xfNest(rs, fm, fNest1, ts)/xft.scale(rs, fm, ts)
  ) $ (p_omegafNest(rs, fNest, fm) eq inf)
;

```

The total factor supply to such a nest `xfNest` as defined in the `e_xfNest` again considers these two cases. In case of finite transformation in the first block, total supply either depends on the sector wide supply of that factor of `xft` and the price relations or on the amount supplied to the nest `fNest1` to which the sub-nest belongs. In case of infinite transformation, the sub-nest price is either equal to the sector-wide factor price `pft` or to the price of the upper nest `pfNest` indexed with `fNest1`.

```

e_xfNest(rs, fm, fNest, ts) $ p_gfNest(rs, fm, fNest, ts) ..
0 =e=
*
*   --- "sluggish" factor supply from top level nest (xft, pft) to current nest
*
*   ( xfNest(rs, fm, fNest, ts)/xft.scale(rs, fm, ts)
*   - p_gfNest(rs, fm, fNest, ts) * xft(rs, fm, ts)/xft.scale(rs, fm, ts)
*     * (pfNest(rs, fm, fNest, ts)/pft(rs, fm, ts))**p_omegaf(rs, fm)
*   ) $ ( (p_omegaf(rs, fm) ne inf) $ fNest_n_f("xft", fNest, fm)
*
*   --- "sluggish" factor supply from another nest fNest1 to the current nest
*
+ sum(fNest_n_f(fNest1, fNest, fm), xfNest(rs, fm, fNest, ts)/xft.scale(rs, fm, ts)
  - p_gfNest(rs, fm, fNest, ts) * xfNest(rs, fm, fNest1, ts)/xft.scale(rs, fm, ts)
  * (pfNest(rs, fm, fNest, ts)/pfNest(rs, fm, fNest1, ts))**p_omegafNest(rs, fNest1, fm)
  ) $ ( sum(fNest_n_f(fNest1, fNest, fm) $ (p_omegafNest(rs, fNest1, fm) ne inf), 1)
*
*   --- non "sluggish" factor supply from top level nest (xft, pft) to current nest
*
+ ( pfNest(rs, fm, fNest, ts) - pft(rs, fm, ts) ) $ ( (p_omegaf(rs, fm) eq inf) $ fNest_n_f("xft", fNest, fm)
*
*   --- non "sluggish" factor supply from another nest to current nest
*
+ sum(fNest_n_f(fNest1, fNest, fm) $ (p_omegafNest(rs, fNest1, fm) eq inf),
  pfNest(rs, fm, fNest, ts) - pfNest(rs, fm, fNest1, ts) )
;

```

The factor supply from nation to sub-regions is described in the section “Integration into the modeling framework”, pages 127ff of the chapter “Sub-national dis-aggregation of production and factor markets in CGEBox”. Note also that the “GTAP-AEZ” module (pages 256ff) will introduce land

transformation at the level of Agro-Ecological Zones and replace some of the equations detailed above.

## 2.6 Income generation and distribution - overview

An overview on income generation and distribution under the regional household approach is depicted in Figure 2 below. Regional income is sourced (1) by factor income  $facty$  (factor remuneration including direct taxes) minus depreciation ( $valDep$ ) and (2) by indirect taxes  $yTaxInd$ , i.e. all tax flows  $yTaxTot$  minus direct taxes  $yTax_{dt}$  which are already comprised in the factor income.

Regional household income  $regy$  is distributed to final demand expenditures of private households  $yc$ , government  $yg$  and regional net savings  $rsav$ . Adding the value of depreciation  $valDep$  and of foreign savings  $valSavf$  to regional net savings  $rsav$  yields investment demand expenditures  $yi$ . The distribution of the final demand expenditures to the Amington demands for each product  $xa_i$  is based on CES demand systems for investments and the government which hence encompass the CD or Leontief case, whereas a CDE, LES or (M)AIDADS demand system can be used to distribute private household expenditure  $yc$ .

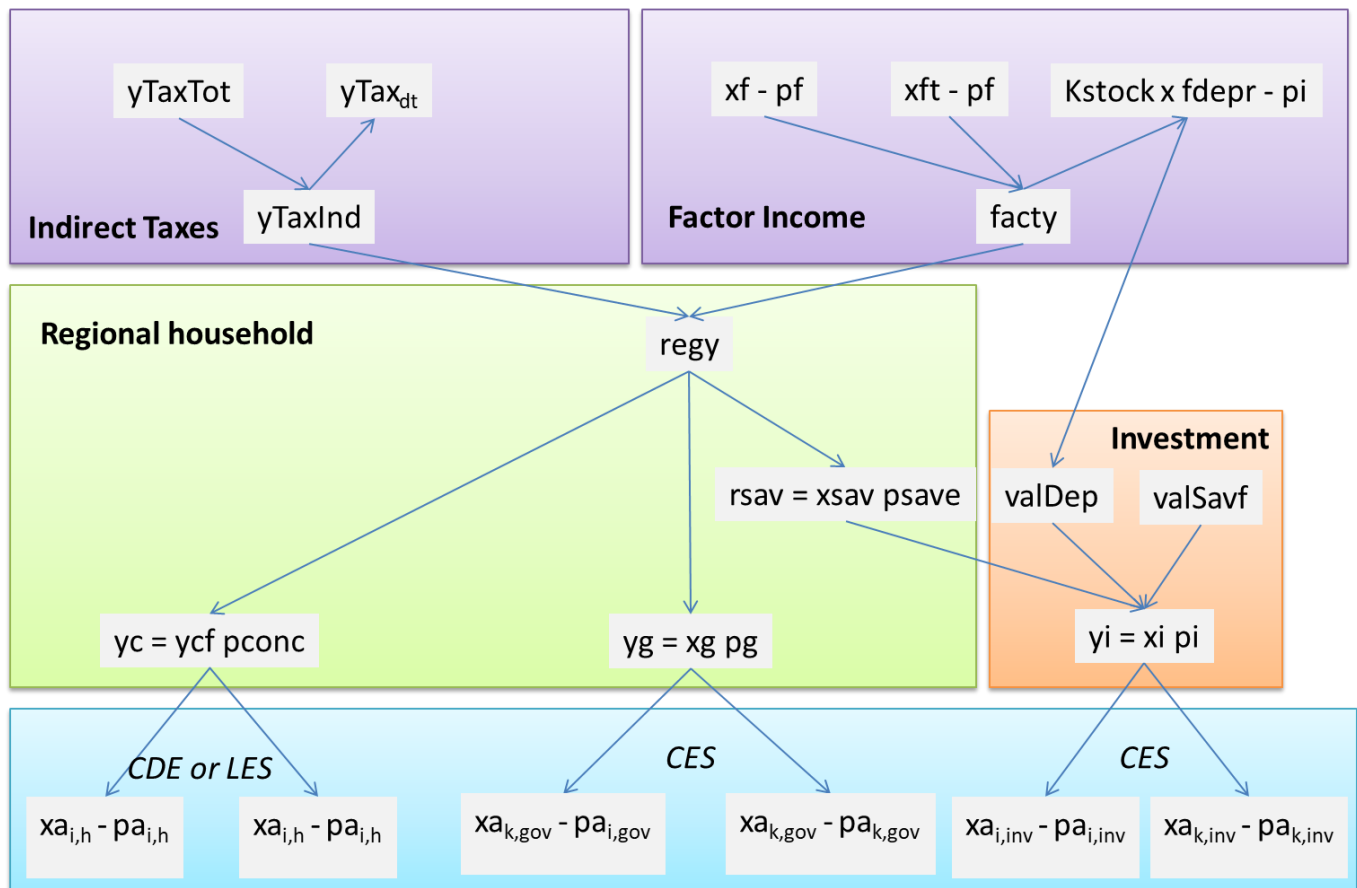


Figure 2: Overview in income generation and distribution

## 2.7 Income generation

**Regional income**  $regy$ , i.e. economy wide income which can be spent on net savings and final consumption by government and private households, is generated from factor income including direct taxes  $facty$  and indirect taxes  $yTaxInd$  as defined in the  $e\_regy$ , equation (26) in VDM 2018:

```

*
* --- regional income, equation (26) in VDM 2018
*
e_regY(rsNat,ts) $ (regy.range(rsNat,ts) ne 0) ..

    regY(rsNat,ts)/regy.scale(rsNat,ts) =e= (factY(rsNat,ts)
        + yTaxInd(rsNat,ts)
        - invFromCapInc(rsNat,ts)
        + finc(rsNat,ts)
        - p_capTrans(rsNat,ts)*pnum(ts))/regy.scale(rsNat,ts);

```

The model allows setting a parameter `p_invFromCapIncShr` which directly convert part of returns to capital into savings (e.g. to cover depreciation):

```

e_invFromCapInc(rsNat,ts) $ (invFromCapInc.range(rsNat,ts) ne 0) ..

    invFromCapInc(rsNat,ts)/invFromCapInc.scale(rsNat,ts)
    =e= [ sum(fm $ (xftFlag(rsNat,fm) $ cap(fm)), pft(rsNat,fm,ts)*xft(rsNat,fm,ts))
        + sum((fnn,a) $ (xfFlag(rsNat,fnn,a) $ cap(fnn)), pf(rsNat,fnn,a,ts)*xf(rsNat,fnn,a,ts)) ]
        * p_invFromCapIncShr(rsNat,ts)/invFromCapInc.scale(rsNat,ts);

```

The foreign income exchange variable `finc` is fixed to zero in the standard model, but can be activated with the `myGTAP` model. The parameter `p_capTrans` capture exogenous income exchanges with the rest-of-the-world as part of the debt accumulation mechanism of G-RDEM.

**Factor income** including direct taxes `factY` is defined by the `e_factY`, equivalent to equation (25) in VDM 2018. It considers returns to primary factors, i.e. economy wide factor prices `pft` multiplied with economy wide factor use `xft` for mobile factors and sector specific factor use `xf` and related prices `pf` for immobile factors. Note that factor income comprises direct taxes. As returns to capital also cover depreciation, the value of depreciation is deducted, considering the depreciation rate `p_fdepr`, the average price of investments `pi` and the capital stock `kstock`:

```

e_factY(rsNat,ts) $ (factY.range(rsNat,ts) ne 0) ..

    factY(rsNat,ts)/factY.scale(rsNat,ts)
    =e= [ sum((f,a) $ xfFlag(rsNat,f,a), pf(rsNat,f,a,ts)*xf(rsNat,f,a,ts))
        - p_fdepr(rsNat,ts)*pi(rsNat,ts)*kstock(rsNat,ts) ]/factY.scale(rsNat,ts);

```

**Indirect tax income** `yTaxInd` is calculated by the equation `e_ytaxInd` from total tax revenues `ytaxTot`, corrected for direct taxes (index “dt”) which are comprised in factor income `factY` as defined above, see equation (24) in VDM 2018:

```

e_ytaxInd(rsNat,ts) ..

    yTaxInd(rsNat,ts)/yTaxInd.scale(rsNat,ts) =e= (ytaxTot(rsNat,ts) - ytax(rsNat,"dt",ts))/yTaxInd.scale(rsNat,ts);

```

**Total tax income** `yTaxTot` considers all tax flow `gy` depicted in the model (see next equation) and is defined by the equation `e_ytaxTot`, equivalent to equation (23) in VDM 2018:

```

e_ytaxTot(rsNat,ts) ..

    ytaxTot(rsNat,ts)/yTaxTot.scale(rsNat,ts) =e= sum(gy, ytax(rsNat,gy,ts))/yTaxTot.scale(rsNat,ts);

```

**Different tax flows** `yTax` for the different types of tax flow `gy` are defined by the equation `e_ytax`, defined in equation (13)-(22) in VDM 2018. It considers the following blocks:

1. **Production taxes** `pt`, levied with the relative tax rate `prdtX` on sectoral revenues, i.e. output `m_px` times the related producer price `m_xp`. Note that fix cost might be present in the model if the Melitz/Krugmann extension is used, depicted by the technology nest “Top” on which also production taxes are charged.



```

e_ytax(rsNat,gy,ts) $ ((not sameas(gy,"fc")) $ (yTax.range(rsNat,gy,ts) ne 0)) ..
ytax(rsNat,gy,ts)/ytax.scale(rsNat,gy,ts) =e=
[
*
* --- Tax revenues from production tax, equation (13) in VDM 2018
[ sum(a $ (prdtx.l(rsNat,a,ts) $ xpFlag(rsNat,a)), prdtx(rsNat,a,ts)
* ( (m_px(rsNat,a,ts) * m_xp(rsNat,a,ts)) $ (not disr(rsNat))
+ sum(r_r(subr,rsNat) $ xpFlag(subr,a), m_px(subr,a,ts) * m_xp(subr,a,ts)) $ disr(rsNat)
+ sum(tNest $ tNest_n_a("Top",tNest,a),xtNest(rsNat,tNest,a,ts)*ptNest(rsNat,tNest,a,ts))))] $ sameas(gy,"pt")
$Sifi "%ModulesGTAT_NUTS2%"=="ON"

```

2. **Indirect taxes on private consumption** pc, levied with rate dintx on domestic consumption expenditure of private household m\_xd times the related domestic price m\_pd, and with rate mintx on imports by private households m\_xm times the average price of imports pmt:

```

$Siftheni.hsubr %hSubr%==on
+ [sum((r_r(rsNat,i,h,ts) + itxshft(rs,i,ts)) * m_domPrice(rs,i,h,ts)*xd(rs,i,h,ts) ] $ (m_alphad(rs,i,h,ts) gt eps)
+ [ (dintx(rs,i,h,ts) + itxshft(rs,i,ts)) * m_domPrice(rs,i,h,ts)*xn(rs,i,h,ts) ] $ (m_alphan(rs,i,h,ts) gt eps)
+ [ (mintx(rs,i,h,ts) + itxshft(rs,i,ts)) * m_pmt(rsNat,i,h,ts)*xm(rs,i,h,ts) ] $ (m_alpham(rs,i,h,ts) gt eps)
] $ sameas(gy,"pc")
$Selse.hsubr
+ [sum((i,h),
[ (dintx(rsNat,i,h,ts) + itxshft(rsNat,i,ts)) * m_domPrice(rsNat,i,h,ts)*xd(rsNat,i,h,ts) ] $ (m_alphad(rsNat,i,h,ts) gt eps)
+ [ (mintx(rsNat,i,h,ts) + itxshft(rsNat,i,ts)) * m_pmt(rsNat,i,h,ts)*xm(rsNat,i,h,ts) ] $ (m_alpham(rsNat,i,h,ts) gt eps)
] $ sameas(gy,"pc")
$Sendif.hsubr

```

Note the product specific general tax shifter itxShft which can be applied to multiple products which is normally not active and allows for specific closure swaps.

3. Two other blocks apply the very same logic to **government and investment consumption**, resulting in tax flows gc and ic:

```

* --- Indirect tax revenues from government consumption, equation (16) in VDM 2018
+ [sum((i,gov), [ (dintx(rsNat,i,gov,ts) + itxshft(rsNat,i,ts)) * m_domPrice(rsNat,i,gov,ts)*xd(rsNat,i,gov,ts) ] $ (m_alphad(rsNat,i,gov,ts) gt eps)
+ [ (mintx(rsNat,i,gov,ts) + itxshft(rsNat,i,ts)) * m_pmt(rsNat,i,gov,ts)*xm(rsNat,i,gov,ts) ] $ (m_alpham(rsNat,i,gov,ts) gt eps)
] $ sameas(gy,"gc")
* --- Indirect tax revenues from investment consumption, equation (17) in VDM 2018
+ [sum((i,inv), [ (dintx(rsNat,i,inv,ts) + itxshft(rsNat,i,ts)) * m_domPrice(rsNat,i,inv,ts)*xd(rsNat,i,inv,ts) ] $ (m_alphad(rsNat,i,inv,ts) gt eps)
+ [ (mintx(rsNat,i,inv,ts) + itxshft(rsNat,i,ts)) * m_pmt(rsNat,i,inv,ts)*xm(rsNat,i,inv,ts) ] $ (m_alpham(rsNat,i,inv,ts) gt eps)
] $ sameas(gy,"ic")

```

4. **Direct taxes** dt are levied with factor specific rates kappaF on factor income, i.e. factor prices pf respectively pft times factor use xf respectively xft. An endogenous or exogenous direct tax shifter kappaf can be added:

```

* --- Direct tax revenues, equation (22) in VDM 2018
+ [ sum(fm $ xftFlag(rsNat,fm), (kappaf(rsNat,fm,ts)+kappaShft(rsNat,ts)) * pft(rsNat,fm,ts)*xft(rsNat,fm,ts))
+ sum((fnn,a) $ xfFlag(rsNat,fnn,a), (kappaf(rsNat,fnn,ts)+kappaShft(rsNat,ts)) * pf(rsNat,fnn,a,ts)*xf(rsNat,fnn,a,ts))
] $ sameas(gy,"pdt")

```

5. **Export tax revenues** et are based on bi-lateral export tax rates exptx and potentially commodity specific export tax shifter etax. Note that depending on how exports are depicted (infinite transformation or not), different prices are used: (a) the bi-lateral export price pe if there is finite transformation between destination, (b) the average export price pet if there is infinite transformation between destinations, but not between exports and domestic sales and (c) the supply price ps if all transformations are infinite. Finally, if the Melitz module is active, the firm price defined in the macro m\_pFirm is used. The related quantity is defined in the macro m\_xws.

```

*      --- Export tax revenues, equation (21) in VDM 2018
+      [sum((i,rNat1) $ ( (exptx.1(rsNat,i,rNat1,ts) + etax.1(rsNat,i,ts)) $ m_xwFlag(rsNat,i,rNat1) ),
      (exptx(rsNat,i,rNat1,ts) + etax(rsNat,i,ts))
      * [ (
        pe(rsNat,i,rNat1,ts) $ psFlag(rsNat,i,"pe")
        + pet(rsNat,i,ts)      $ psFlag(rsNat,i,"pet")
        + ps(rsNat,i,ts)       $ psFlag(rsNat,i,"ps")
      )
      $$ifthen.iMel "%melitz%=="on"
      $ (not iMel(i))
      + (m_pFirm(rsNat,i,rNat1,ts) $ iMel(i))
      $$endif.iMel
      ]
      *m_xws(rsNat,i,rNat1,ts))] $sameas(gy,"et")

```

6. **Import taxes** “mt” are defined from bi-lateral import taxes imptx, a commodity specific import tax shifter mtax, the bi-lateral c.i.f. prices defined by the global %pmcif% and the bi-lateral flows xw.

```

*      --- Import tax revenues, equation (20) in VDM 2018
+      [sum((i,rNat1) $ m_xwFlag(rNat1,i,rsNat), (imptx(rNat1,i,rsNat,ts) + mtax(rsNat,i,ts))
      *%pmcif%(rNat1,i,rsNat,ts)*xw(rNat1,i,rsNat,ts))] $sameas(gy,"mt")

```

7. **Factor taxes** ft paid by each activity are levied on the activity specific factor price pf and use xf with the rates fcttx and a factor tax shifter fcttxShift:

```

*      ---- Tax revenues from factor taxes, equation (18) in VDM 2018
+      [sum((f,a) $ xfFlag(rsNat,f,a), (fcttx(rsNat,f,a,ts) + fcttxShift(rsNat,ts) $ fTaxShift(f,a))
      * pf(rsNat,f,a,ts) * xf(rsNat,f,a,ts))] $sameas(gy,"ft")

```

8. The same logic (less the shifter) applies for factor subsidies fs:

```

*      ---- Revenue costs from factor subsidies, equation (19) in VDM 2018
+      [sum((f,a) $ xfFlag(rsNat,f,a), fctts(rsNat,f,a,ts) * pf(rsNat,f,a,ts) * xf(rsNat,f,a,ts))] $sameas(gy,"fs")

```

9. Finally, **emission taxes** emis can be introduced, levied on emissions emis with the potentially endogenous price emisP. These can comprise CO2, non-Co2 and carbon stock changes.

```

*      ---- Revenues from emission tax (not part of Standard GTAP)
*      $$ifi %ModulesCO2_Emissions%==on + [emisTaxed(rsNat,ts) * emisP(rsNat,ts) * emisPShr(rsNat,ts) ] $ sameas(gy,"emis")

```

## 2.8 Income distribution

If the regional household approach is used, savings, government and household demand are distributed based on a modified CD utility function where the private household demand share is driven by the utility of total private expenditure with regard to utility phiP and the original private demand share betaP, part of equation (28) of VDM 2018:

```

*      --- intermediate terms to help solver, part of equation (28) in VDM 2018
*      e_betaPhi(rsNat,ts) ..
*      betaPhi(rsNat,ts) =E= betaP(rsNat,ts)/phiP(rsNat,ts);

```

The updated share betaP termed betaPhi implies that the shares as defined in the benchmark do not add to unity any longer. Therefore, an intermediate variable phiRegY is defined which scales regional income to reflect the updated sum of the shares. Assume that betaPhi is increased compared to the benchmark. That implies that the second term on the LHS exceed unity. The correction factor phiRegY

is accordingly proportionally decreased such that the three expenditure shares still sum up to unity, part of equation (28) in VDM 2018:

```
e_phiRegy(rsNat,ts) ..
    phiRegy(rsNat,ts)*(betaPhi(rsNat,ts) + betaG(rsNat,ts) + betaS(rsNat,ts))
    =E= 1;
```

Beside the CD case depicted in the GTAP Standard model, CGEBox can consider the general CES case where a price index pu for top-level utility is defined as follows:

```
*
*   --- price index related to CES utility function
*
e_pu(rsNat,ts) $ (p_sigmaU(rsNat) ne 1) ..
*
    pu(rsNat,ts) =E= ( betaPhi(rsNat,ts)*phiRegy(rsNat,ts) * pCons(rsNat,"hhsld",ts)**(1-p_sigmaU(rsNat))
    + betas(rsNat,ts) * phiRegy(rsNat,ts) * pSave(rsNat,ts) ** (1-p_sigmaU(rsNat))
    + betaG(rsNat,ts) * phiRegy(rsNat,ts) * pg(rsNat,ts) ** (1-p_sigmaU(rsNat))) ;
```

The amount spent for private consumption yc is defined in the equation e\_yc, (equation (29) in VDM 2018 for the first CD related part):

```
e_yc(rsNat,h,ts) ..
    yc(rsNat,h,ts)/yc.scale(rsNat,h,ts)
    =E= [ betaPhi(rsNat,ts)*phiRegy(rsNat,ts) * regy(rsNat,ts)/yc.scale(rsNat,h,ts) ] $ (p_sigmaU(rsNat) eq 1)
    + [ betaPhi(rsNat,ts)*phiRegy(rsNat,ts) * (regy(rsNat,ts)/pu(rsNat,ts))/yc.scale(rsNat,h,ts)
    * pCons(rsNat,"hhsld",ts)**(1-p_sigmaU(rsNat))] $ (p_sigmaU(rsNat) ne 1) ;
```

The amount spent for government consumption yg is defined in the equation e\_yg (equation (30) in VDM 2018 for the first CD related part):

```
e_yg(rsNat,ts) ..
    [ xg(rsNat,ts) $ (p_sigmaU(rsNat) ne 1) + yg(rsNat,ts) $ (p_sigmaU(rsNat) eq 1)]
    /yg.scale(rsNat,ts)
    =E= [ betaG(rsNat,ts)*phiRegy(rsNat,ts) * regy(rsNat,ts)/yg.scale(rsNat,ts) ] $ (p_sigmaU(rsNat) eq 1)
    + [ betaG(rsNat,ts)*phiRegy(rsNat,ts) * (regy(rsNat,ts)/pu(rsNat,ts))/yg.scale(rsNat,ts)
    * pg(rsNat,ts)**(-p_sigmaU(rsNat))] $ (p_sigmaU(rsNat) ne 1) ;
```

And finally, the amount of regional savings rsav is depicted in the equation e\_rsav (equation (31) in VDM 2018 for the first CD related part):

```
e_rsav(rsNat,ts) ..
    [ xsav(rsNat,ts) $ (p_sigmaU(rsNat) ne 1) + rsav(rsNat,ts) $ (p_sigmaU(rsNat) eq 1)]
    /xsav.scale(rsNat,ts)
    =E= [ betaS(rsNat,ts) * phiRegy(rsNat,ts) * regy(rsNat,ts)/xsav.scale(rsNat,ts) ] $ (p_sigmaU(rsNat) eq 1)
    + [ betaS(rsNat,ts) * phiRegy(rsNat,ts) * (regy(rsNat,ts)/pu(rsNat,ts))/xsav.scale(rsNat,ts)
    * pSave(rsNat,ts)**(-p_sigmaU(rsNat))] $ (p_sigmaU(rsNat) ne 1) ;
```

Note that the regional household approach can be replaced by separate accounts; see the sub-section “Model equations” in the section “myGTAP module”.

## 2.9 Household consumption

The **consumer price index** pcons is defined from the budget share xcshr and the Armington prices defined in the macro m\_pa, equation (38) in VDM 2018:

```
* --- Consumer expenditure deflator (approx.) -- PHHLINDEX (1005)
*
e_pcons(rs,h,ts) $ (sum(i, xaFlag(rs,i,h)) and ((pcons.range(rs,h,ts) ne 0) or (not rres(rs)) or (lcu.range(rs,ts) ne 0))) ..
    pcons(rs,h,ts) =E= sum(i $ xaFlag(rs,i,h), xcshr(rs,i,h,ts)*m_pa(rs,i,h,ts));
```

The Armington demands for household consumption can be defined either by a CDE demand system as used in the GTAP Standard model, a MAIDADS or a LES demand system as found in many other

CGEs. Note that the MAIDADS system can be turned into LES system if the marginal budget shares and the commitment terms are not rendered a function of utility, but fixed. Equally, the LES system collapses to a CD system if the commitments are removed, such the model can host four different demand systems for household consumption, for which also benchmarking code is provided.

### 2.9.1 CDE case

In the standard GTAP model, a constant difference in elasticity (CDE) indirect demand system is used. The equations can be found “model\dem\_cde.gms”. The **final demand quantity**  $xa$  (the Armington demand) for each household  $h$  and product  $i$  are defined from the budget shares  $xcshr$  and the private consumption expenditures  $yc$ , see equation  $e\_xac$ , equation (37) in VDM 2018:

```
e_xac(rs(r),iIn(i),h,ts(t)) $ xaFlag(r,i,h) ..
  xa(r,i,h,t) /xa.scale(r,i,h,t)*m_pa(r,i,h,t) =e= xcshr(r,i,h,t)/xa.scale(r,i,h,t)*yc(r,h,t);
```

The **budget shares**  $xcshr$  as defined in the equation  $e\_xcshr$  are derived from unscaled budget shares  $zcons$ , scaled again by unity based on their sum  $zConsSum$ , equation (36) in VDM 2018:

```
e_xcshr(rs(r),iIn(i),h,ts(t)) $ (xaFlag(r,i,h) $ (not sum(dNest_i_fd(dNest,i,h),1))) ..
  xcshr(r,i,h,t)/xcshr.scale(r,i,h,t) =e= zcons(r,i,h,t)/zConsSum(r,h,t) * 1/xcshr.scale(r,i,h,t);
```

The unscaled shares  $zcons$  as defined in the equation  $e\_zcons$  depend on utility  $uh$  (i.e. indirectly on expenditures) and the product prices defined in the macro  $m\_pa$  relative to income  $yc$  per capita, defined from population  $pop$  for that household and three given parameter vectors  $alphaa$ ,  $bh$  and  $eh$ . That is an unnumbered equation in VDM 2018, noted before equation (36):

```
*
* --- auxiliary variables in CDE demand system, not numbered equation in VDM 2018
* (above equation (36) in VDM 2018)
*
e_zcons(rs(r),iIn(i),h,ts(t)) $ (xaFlag(r,i,h) $ (not sum(dNest_i_fd(dNest,i,h),1))) ..
  zcons(r,i,h,t)/zcons.scale(r,i,h,t) =e= alphaa(r,i,h,t)*bh(r,i,t)

*
* --- bh measures substitution effect, driven by price change
* relative to income change
*
* m_pa(r,i,h,t)**bh(r,i,t)
* ((yc(r,h,t)/pop(r,h,t)*1000)**(-bh(r,i,t)))
*
* --- expansion effect, driven by utility (= income)
*
* (uh(r,h,t)**(eh(r,i,t)*bh(r,i,t)))/zcons.scale(r,i,h,t);
```

The sum of these unscaled shares must also consider the case of sub-nests in demand, defined in the equation  $e\_zConsSum$ , part of equation (36) in VDM 2018:

```
*
* ---- definition of sum of Z, part of equation (36) in VDM 2018
*
e_zConsSum(rs(r),h,ts(t)) $ sum(j,xaFlag(r,j,h)) ..
  zConsSum(r,h,t)/sum(j $xaFlag(r,j,h), zcons.scale(r,j,h,t))
  =e= [ sum(j $ (xaFlag(r,j,h) $ (not sum(dNest_i_fd(dNest,j,h),1))), zcons(r,j,h,t))
    + sum(dNest_n_fd("top",dNest,h) $ p_alphaDN(r,dNest,h,t), zconsDN(r,dNest,h,t))
    ] /sum(j $xaFlag(r,j,h), zcons.scale(r,j,h,t)) ;
```

The **utility level**  $u$  is indirectly defined by the following equation in the equation  $e\_uh$ , equation (32) in VDM 2018:

```

*
* --- Household utility (per capita), equation (32) in 2018
*
e_uh(rs(r),h,ts(t)) $ sum(i,xaFlag(r,i,h)) ..
    1 =e= sum(i $ (xaFlag(r,i,h) $ (not sum(dNest_i_fd(dNest,i,h),1))), zcons(r,i,h,t)/bh(r,i,t))
*
* --- add sub-nests under final demand function, not part of standard GTAP model
*
    + sum(dNest_n_fd("top",dNest,h) $ p_alphaDN(r,dNest,h,t), zconsDN(r,dNest,h,t)/bhDN(r,dNest,t));

```

Finally, the **elasticity of private expenditure versus private utility** `phip` is defined in the equation `e_phiP`. It updates the private consumption share in the regional household income distribution:

```

*
* --- Elast. of priv. exp. wrt to priv. utility, equation (27) in VDM 2018
*
e_phiP(rsNat(rNat),ts(t)) $ ((phip.range(rNat,t) ne 0)) ..
    phip(rNat,t) =e= sum( (r_r(rs,rNat),i,h) $ xaFlag(rs,i,h), xcshr(rs,i,h,t)*eh(rs,i,t)*pop(rs,h,t))
        /sum((r_r(rs,rNat),h) $ hFlag(rs,h),pop(rs,h,t));

```

## 2.9.2 LES or CD case

The equations for the LES or CD case are found in the file “model\dem\_les.gms”. The LES case is not part of GTAP Standard model.

The **Armington demands** `xa` in the LES case reflect the constant term `gammaLES`, often termed commitment, and a share `alphaLES` on no-committed income `yCNonCom` divided by the Armington price defined in the macro `m_pa`:

```

*
* --- demand equations for individual products (if not not under sub-nest, this is checked by flag p_hasXA)
*
e_xacLes(rs,i,h,ts) $ p_hasXA(rs,i,h) ..
    xa(rs,i,h,ts) /xa.scale(rs,i,h,ts)
    =e= (
        $iftheni.baseline "%fixCalories%"=="on"
            gammaLES(rs,i,h,ts)
        $else.baseline
            gammaLESM(rs,i,h,ts)
        $endif.baseline
        /1000*pop(rs,h,ts) + alphaLES(rs,i,h,ts)/m_pa(rs,i,h,ts) * yCNonCom(rs,h,ts)/xa.scale(rs,i,h,ts);

```

The `gammaLes` is an additional variable used in the context of G-RDEM if calories are driven by given per capita-demand.

This equation replaces equation (37) in VDM 2018, CDE case:

$$XA_{r,i,h} = \gamma_{r,i,h}^{LES} pop_{r,h} + \alpha_{r,i,h}^{LES} ycNonCom_{r,h} \quad (37^*)$$

The same functional relation is also used to define sub-nests demands as defined in the equation `e_xdNestLes`:

```

*
* --- demand for sub-nests under top nest
*
e_xdNestLes(rs,dNest,h,ts) $ sum(dNest_n_fd("top",dNest,h), p_hasXA(rs,dNest,h)) ..
    xdNest(rs,dNest,h,ts) /xdNest.scale(rs,dNest,h,ts)
    =e= (
        $iftheni.baseline "%fixCalories%"=="on"
            gammaLES(rs,dNest,h,ts)
        $else.baseline
            gammaLESM(rs,dNest,h,ts)
        $endif.baseline
        /1000*pop(rs,h,ts) + alphaLES(rs,dNest,h,ts)/pdNest(rs,dNest,h,ts) * yCNonCom(rs,h,ts)/xdNest.scale(rs,dNest,h,ts);

```



**Non-committed income**  $yCNonCom$  as defined in the equation  $e\_yCNonCom$  reflect total private consumption expenditure  $yc$  minus the value of the commitments, i.e. the gamma parameters multiplied with the Armington prices defined in the macro  $m\_pa$ :

```
*
* --- non committed income: total expenditure for private consumption minus
* value of commitments. Note that top-level nests (only) of the nested
* demand system are taken into account. Lower nests are all CES-functions
*
e_yCNonCom(rs,h,ts) $ sum(i,xaFlag(rs,i,h)) ..
ycNonCom(rs,h,ts)/yc.scale(rs,h,ts)
=E= (yc(rs,h,ts)
*
* --- note: commitment terms are defined in a per capita basis
*
$iftheni.baseline "%fixCalories%"=="on"
- sum(i $ p_hasXa(rs,i,h), gammaLES(rs,i,h,ts) *pop(rs,h,ts)/1000 * m_pa(rs,i,h,ts))
- sum(dNest_n_fd("top",dNest,h) $ p_hasXa(rs,dNest,h), gammaLES(rs,dNest,h,ts) *pop(rs,h,ts)/1000 * pdNest(rs,dNest,h,ts))/yc.scale(rs,h,ts);
$else.baseline
- sum(i $ p_hasXa(rs,i,h), gammaLES(rs,i,h,ts) *pop(rs,h,ts)/1000 * m_pa(rs,i,h,ts))
- sum(dNest_n_fd("top",dNest,h) $ p_hasXa(rs,dNest,h), gammaLES(rs,dNest,h,ts) *pop(rs,h,ts)/1000 * pdNest(rs,dNest,h,ts))/yc.scale(rs,h,ts);
$endif.baseline
```

In mathematical notation:

$$yCNonCom_{r,h} = \left( yc_{r,h} - \sum_i \gamma_{r,i,h}^{LES} pop_{r,h} PA_{r,i,h} - \sum_{dNest \in top} \gamma_{r,dNest,h}^{LES} pop_{r,h} PDNest_{r,i,h} \right)$$

The **budget shares**  $xcshr$  are defined from the Armington demands, prices and expenditures in the equation  $e\_xcshrLESeq$ :

```
*
* --- definition of budget shares
*
e_xcshrLES(rs,i,h,ts(t)) $ xaFlag(rs,i,h) ..
xcshr(rs,i,h,ts)*yc(rs,h,ts)/xa.scale(rs,i,h,ts) =e= xa(rs,i,h,ts)*m_pa(rs,i,h,ts)/xa.scale(rs,i,h,ts);
```

These budget share equation replace equation (36) in VDM 2018 for the CDE case:

$$s_{r,i,h}^p yC_{r,h} = XA_{r,i,h} PA_{r,i,h} \quad (36^*)$$

Finally, the utility for the private households  $uh$  is defined in the equation  $uhLESeq$ :

```
*
* --- definition of utility from private household, LES
*
e_uhLES(rs,h,ts) $ (sum(i,xaFlag(rs,i,h)) $ (p_demSystem eq LES)) ..
uh(rs,h,ts)
=E= ycNonCom(rs,h,ts)/pop(rs,h,ts) * 1000
* ( prod(i $ (xaFlag(rs,i,h) $ alphaLES.1(rs,i,h,ts)),
(alphaLES(rs,i,h,ts)/m_pa(rs,i,h,ts)) ** alphaLES(rs,i,h,ts))
* prod(dNest_n_fd("top",dNest,h) $ alphaLES.1(rs,dNest,h,ts),
(alphaLES(rs,dNest,h,ts)/pdNest(rs,dNest,h,ts)) ** alphaLES(rs,dNest,h,ts)))
* 1/uhLES(rs,h,ts);
```

Note that the CD case is comprised in the equations above if the commitment terms are set to zero. The equation above replaces equation (32) from VDM 2018 for the CDE case:

$$U_r^h = \frac{1}{UBas_r^h} \prod_i \left( XA_{r,i,h} - \gamma_{r,i,h}^{LES} pop_{r,h} \right)^{\alpha_{r,i,h}^{LES}} \prod_{tNest \in top} \left( XdNest_{r,tNest,h} - \gamma_{r,tNest,h}^{LES} pop_{r,h} \right)^{\alpha_{r,tNest,h}^{LES}} \quad (32^*)$$

The econometrically estimated AIDADS system as part of G-RDEM modules is detailed in section “An AIDADS **demand** system with detail for food consumption”, pages 183 ff.

## 2.10 Government consumption

First the reader is reminded that there is no separate government account under the regional household approach and hence no direct link between tax income and government spent. That can be changed by using the “myGTAP module”.

Government consumption expenditure  $y_g$  under the regional household approach is a share  $\beta_{tag}$  of regional income  $regy$ , corrected for the endogenous share of private spent captured by  $\phi_{Regy}$  (see above for savings), equation (39) in VDM 2018:

```
*
* --- Total government consumption expenditure, equation (30) in VDM 2018
*
e_yg(rsNat,ts) ..
    [ xg(rsNat,ts) $ (p_sigmaU(rsNat) ne 1) + yg(rsNat,ts) $ (p_sigmaU(rsNat) eq 1)]
    /yg.scale(rsNat,ts)
    =e= [betaG(rsNat,ts)*phiRegy(rsNat,ts) * regy(rsNat,ts)/yg.scale(rsNat,ts) ] $ (p_sigmaU(rsNat) eq 1)
        + [betaG(rsNat,ts)*phiRegy(rsNat,ts) * (regy(rsNat,ts)/pu(rsNat,ts))/yg.scale(rsNat,ts)
            * pg(rsNat,ts)**(-p_sigmaU(rsNat))] $ (p_sigmaU(rsNat) ne 1) ;
```

The physical demand aggregate  $x_g$  define in the equation  $e_{xg}$  is derived from the government price index  $pg$ .

```
*
* --- Nominal government expenditures, equation (41) in VDM 2018
*
e_xg(rsNat,ts) $ p_axg(rsNat,ts) ..

pg(rsNat,ts)*xg(rsNat,ts)/xg.scale(rsNat,ts) =e= yg(rsNat,ts)/xg.scale(rsNat,ts) ;
```

The equation  $e_{pg}$ , equation (40) in VDM 2018, is expanded in CGEBox to account for demand nests. The related price index  $pg$  is defined under the assumption of a CES / CD / Leontief demand system for the final demand of government using the typical dual price aggregator based on the government specific Armington prices defined by the marco  $m_{pa}$ , a preference shifter variable  $\lambda_{tag}$  and the share parameters  $\alpha_{tag}$  for government. Which case is used is defined by the related substitution elasticity  $\sigma_{tag}$ . Note that the equation also considers the case that government demand uses CES sub-nests which aggregate products or other product groups to product groups:

```
e_pg(rsNat,ts) $ p_axg(rsNat,ts) ..

pg(rsNat,ts) =E=
*
* --- CD case with p_sigmag==1
*
( [ prod((i,gov) $ ((not sum(dNest_i_fd(dnest,i,gov),1)) $ alphaa.1(rsNat,i,gov,ts)),
    (m_pa(rsNat,i,gov,ts)/alphaa(rsNat,i,gov,ts))**alphaa(rsNat,i,gov,ts))
*
* -- CES-subnests, not part of Standard GTAP model
*
* prod(dNest_n_fd("top",dnest,gov) $ p_alphaDN(rsNat,dNest,gov,ts),
    (pdNest(rsNat,dNest,gov,ts)/p_alphaDN(rsNat,dNest,gov,ts))**p_alphaDN(rsNat,dNest,gov,ts))
    ]/p_axg(rsNat,ts) ) $ (p_sigmag(rsNat) eq 1)

*
* --- CES or Leontief case with p_sigmag<>1
*
+
( [ sum((i,gov) $ ((not sum(dNest_i_fd(dnest,i,gov),1)) $ alphaa.1(rsNat,i,gov,ts)),
    alphaa(rsNat,i,gov,ts)*m_pa(rsNat,i,gov,ts)**(1-p_sigmag(rsNat)))
*
* -- CES-subnests, not part of Standard GTAP model
*
+ sum(dNest_n_fd("top",dnest,gov) $ p_alphaDN(rsNat,dNest,gov,ts),
    p_alphaDN(rsNat,dNest,gov,ts)*pdNest(rsNat,dNest,gov,ts)**(1-p_sigmag(rsNat)))
    ]**((1/(1-p_sigmag(rsNat)))/p_axg(rsNat,ts)
    ) $ (p_sigmag(rsNat) ne 1) ;
```

The macro  $m_{pa}$  is usually defined as follows. If both a share parameters for imports  $\alpha_{ham}$  and for domestic sales  $\alpha_{had}$  is given, it uses the definition of the Armington price  $m_{pade}$  given below.

Otherwise, it uses directly either the macro for the domestic price `m_pdp` or for the import prices `m_pmp`. Finally, if neither of the two share parameters is given, the Armington price is used, a case relevant when the Melitz module is active.

```

$macro m_pa(r,i,aa,t) {
    m_padef(r,i,aa,t) $ (alpham(r,i,aa,t) $ alphad(r,i,aa,t) ) \
    + (m_pdp(r,i,aa,t)*alphad(r,i,aa,t)) $ ((not alpham(r,i,aa,t)) $ alphad(r,i,aa,t))\
    + (m_pmp(r,i,aa,t)*alpham(r,i,aa,t)) $ ((not alphad(r,i,aa,t)) $ alpham(r,i,aa,t))\
    $if1 "%modulesGTAP_MELITZ%"=="on" + pa(r,i,aa,t) $ ( not (alpham(r,i,aa,t)+alphad(r,i,aa,t)) ) \
}

```

The macro `m_padef` can either introduce the Armington price `pa` or can replace it with the dual price aggregator, equation (47) in VDM 2018:

```

$macro m_padef(r,i,aa,t) { ( (alphad(r,i,aa,t)*m_pdp(r,i,aa,t)**(1-sigmam(r,i))) \
    + (alpham(r,i,aa,t)*m_pmp(r,i,aa,t)**(1-sigmam(r,i))) )**(1/(1-sigmam(r,i))) )

```

The total physical government demand `xg` is distributed in the equation `e_xag`, equation (39) in VDM 2018, to demands for individual products based on given share parameters `alphaa`, Armington prices captured by the `m_pa` macro and the average price `pg` based on the substitution elasticity `sigmag`:

```

*
* --- CES expenditure function for government, equation (39) in VDM 2018
*
e_xag(rsNat,iIn(i),gov,ts) $ (xaFlag(rsNat,i,gov) $ (not sum(dNest_i_fd(dnest,i,gov),1)) $ alphaa.1(rsNat,i,gov,ts)) ..
    xa(rsNat,i,gov,ts)/xa.scale(rsNat,i,gov,ts)
    =e= alphaa(rsNat,i,gov,ts)*xg(rsNat,ts)/xa.scale(rsNat,i,gov,ts)*(pg(rsNat,ts)/m_pa(rsNat,i,gov,ts))**p_sigmag(rsNat);

```

Note that that in the case of demand nests for government consumption (not part of the GTAP standard model), additional equations are used as described in the section “Demand sub-nests”. The demand nest equation for the government is defined as:

$$\begin{aligned}
 XdNest_{r,dNest,gov} &= \alpha_{r,dNest,gov}^{top} XG_r \left( \frac{PG_r}{PdNest_{r,dNest,gov}} \right)^{\sigma^g} \\
 &+ \alpha_{r,dNest,gov}^{dNest1} XdNest_{r,gov} \left( \frac{PdNest_{r,dNest1,gov}}{PdNest_{r,dNest,gov}} \right)^{\sigma^g}
 \end{aligned}$$

## 2.11 Investments and savings

**Gross investment expenditures** `yi` are composed of the value of depreciation `valDep` and of regional `rsav` and foreign savings `valSavf` as defined in the equation `e_yi`, equation (85) in VDM 2018:

```

e_yi(rsNat,ts)
$iftheni.emp not "%modelType%"=="EMP"
    $ ( ((not rres(rsNat)) or singleCountry or (card(rsNat) eq 1) or (pnum.range(ts) gt 0)) )
$endif.emp
..
yi(rsNat,ts)/yi.scale(rsNat,ts) =e= (
    ---- value of depreciation
    valDep(rsNat,ts)
    ---- savings of the regional household
    + rsav(rsNat,ts)
    + walras(ts) $ rres(rsNat)
    ---- direct investments of firm
    + invFromCapInc(rsNat,ts)
    ---- Value of foreign savings in local currency
    + valsavf(rsNat,ts))/yi.scale(rsNat,ts);

```



The price index of investments  $\pi_i$  is defined by the equation  $e_{\pi_i}$ , equation (43) in VDM 2018, expanded for demand nests. For a detailed explanation of this equation, refer to the explanation for government above:

```
*
* --- Investment expenditure price deflator, equation (43) in VDM 2018
*
e_pi(rsNat,ts) ..
    pi(rsNat,ts) =e=
*
*      --- CD case with p_sigmai==1
*
*      ( [ prod((i,inv) $ ((not sum(dNest_i_fd(dnest,i,inv),1)) $ alphaa.l(rsNat,i,inv,ts)),
*              (m_pa(rsNat,i,inv,ts)/(lambdai(rsNat,i,ts)*alphaa(rsNat,i,inv,ts)))**alphaa(rsNat,i,inv,ts))
*
*      -- CES-subnests, not part of Standard GTAP model
*
*      * prod(dNest_n_fd("top",dnest,inv) $ p_alphaDN(rsNat,dNest,inv,ts),
*              (pdNest(rsNat,dNest,inv,ts)/p_alphaDN(rsNat,dNest,inv,ts))**p_alphaDN(rsNat,dNest,inv,ts))
*              ]/p_axi(rsNat,ts) ) $ (p_sigmai(rsNat) eq 1)
*
*      --- CES or Leontief case with p_sigmai<>1
*
+      ([ sum((i,inv) $ ((not sum(dNest_i_fd(dnest,i,inv),1)) $ alphaa.l(rsNat,i,inv,ts)),
*              alphaa(rsNat,i,inv,ts)*(m_pa(rsNat,i,inv,ts)/lambdai(rsNat,i,ts))**((1-p_sigmai(rsNat))))
*
*      -- CES-subnests, not part of Standard GTAP model
*
+      sum(dNest_n_fd("top",dnest,inv) $ p_alphaDN(rsNat,dNest,inv,ts),
*          p_alphaDN(rsNat,dNest,inv,ts)*pdNest(rsNat,dNest,inv,ts)**((1-p_sigmai(rsNat))))
*
*      ]*(1/(1-p_sigmai(rsNat)))/p_axi(rsNat,ts)
*      ) $ (p_sigmai(rsNat) ne 1)
*
;
```

The **total physical investment demand**  $\xi_i$  is derived from the price index  $\pi_i$  defined above and total investment expenditure  $y_i$ , as defined in equation  $e_{\xi_i}$ , equation (44) in VDM 2018:

```
* --- Investment volume, equation (44) in VDM 2018
*
e_xi(rsNat,ts) ..
    yi(rsNat,ts)/xi.scale(rsNat,ts) =E= pi(rsNat,ts)*xi(rsNat,ts)/xi.scale(rsNat,ts);
```

**Product specific investment demand**  $x_{ai}$  as depicted in the equation  $e_{x_{ai}}$ , equation (42) in VDM 2018. It reflects the share parameters  $\alpha_{ai}$ , the substitution elasticity  $\sigma_{ai}$  and the shifter variable  $\lambda_{ai}$ :

```
*
* --- CES expenditure function, equation (42) in VDM 2018
*
e_xai(rsNat,iIn(i),inv,ts) $ (xaFlag(rsNat,i,inv) $ (not sum(dNest_i_fd(dnest,i,inv),1)) $ alphaa.l(rsNat,i,inv,ts)) ..
*
    xa(rsNat,i,inv,ts)/xa.scale(rsNat,i,inv,ts) =e=
        alphaa(rsNat,i,inv,ts)*xi(rsNat,ts)/xa.scale(rsNat,i,inv,ts)
        * [ ((lambdai(rsNat,i,ts)*pi(rsNat,ts)/m_pa(rsNat,i,inv,ts))**p_sigmai(rsNat)/lambdai(rsNat,i,ts) ) $ (p_sigmai(rsNat) ne 0)
        + lambdai(rsNat,i,ts) $ (p_sigmai(rsNat) eq 0)];
```

The **value of depreciation**  $\text{valDep}$  is a given share  $p_{\text{depr}}$  on the capital stock  $k_{\text{stock}}$  which jointly define physical depreciation, multiplied with the average price of savings  $\pi_i$ , equation (85) in VDM 2018:

```
*
* --- Nominal depreciation, part of equation (85) in VDM 2018
*
e_valDep(rsNat,ts) ..
*
    valDep(rsNat,ts)/valDep.scale(rsNat,ts) =E= pi(rsNat,ts)*p_depr(rsNat,ts)*kstock(rsNat,ts)/valDep.scale(rsNat,ts);
```

**Regional savings**  $\text{rsav}$  are a given share  $\beta_{\text{reg}}$  of regional income, corrected for expansion effects  $\phi_{\text{Regy}}$ , equation (31) in VDM 2018:

```
*
*   --- Total nominal saving, equation (31) in VDM 2018
*
e_rsav(rsNat,ts) ..
    [ xsav(rsNat,ts) $ (p_sigmaU(rsNat) ne 1) + rsav(rsNat,ts) $ (p_sigmaU(rsNat) eq 1)]
    /xsav.scale(rsNat,ts)
    =e= [ betaS(rsNat,ts) * phiRegY(rsNat,ts) * regy(rsNat,ts)/xsav.scale(rsNat,ts) ] $ (p_sigmaU(rsNat) eq 1)
        + [ betaS(rsNat,ts) * phiRegY(rsNat,ts) * (regy(rsNat,ts)/pu(rsNat,ts))/xsav.scale(rsNat,ts)
            * pSave(rsNat,ts)**(-p_sigmaU(rsNat))] $ (p_sigmaU(rsNat) ne 1) ;
```

The **physical amount of regional savings** xsav is based on the savings expenditures rsav and the average price of savings psave, equation (34) in VDM 2018:

```
*
*   --- Physical amount of savings, equation (34) in VDM 2018
*
e_xsav(rsNat,ts) ..
    xsav(rsNat,ts)*psave(rsNat,ts)/xsav.scale(rsNat,ts) =e= rsav(rsNat,ts)/xsav.scale(rsNat,ts);
```

The **regional value of foreign savings** valSavf is defined from the value in foreign currency savf and the exchange rate lcu, part of equation (85) in VDM 2018:

```
*
*   --- Value of foreign savings in local currency, part of equation (85) in VDM 2018
*
e_valSavf(rsNat,ts) ..
    valSavf(rsNat,ts)/savf.scale(rsNat,ts) =E= savf(rsNat,ts)*lcu(rsNat,ts)/savf.scale(rsNat,ts);
```

The **foreign savings** in foreign currency savf can be driven by different mechanism. We start by discussing the so-called global bank mechanism which uses expected returns to foreign savings to distribute global net investments.

### 2.11.1 Global bank

The global bank mechanism distributes total global savings such that expected returns to net investments are equal across regions. The different steps in the allocation procedure are described by the following variables and equations.

**Regional physical net investments** netInv are the difference between gross investment demand xi and physical depreciation derived from the capital stock kStock and the depreciation rate p\_depr, part of equation (86) in VDM 2018:

```
*
*   --- Physical net investment in each region, part of equation (86) in VDM 2018
*
e_netInv(rsNat,ts) ..
    netInv(rsNat,ts)/netinv.scale(rsNat,ts) =E= [xi(rsNat,ts) - p_depr(rsNat,ts)*kstock(rsNat,ts)]/netinv.scale(rsNat,ts);
```

The **beginning of period capital stock** kStock is defined in the equation e\_kStock, equation (75) in VDM 2018. It converts with the factor p\_krat the use of capital xft into the aggregate capital stock, where different capital types are defined by the set cap:

```
*
*   --- Aggregate capital stock, equation (75) in VDM 2018
*
e_kstock(rsNat,ts) $ ( (kstock.range(rsNat,ts) ne 0) ) ..
    kstock(rsNat,ts)/kapend.scale(rsNat,ts) =e= sum(cap, xft(rsNat, cap, ts))/(p_krat(rsNat,ts)*kapend.scale(rsNat,ts));
```

The **end of period capital stock** kapEnd is derived by deducting the depreciating rate p\_depr times the number of depreciation year p\_nDeprYears and adding physical gross investment xi times the number of depreciations years, equation (76) in VDM 2018:

```

*
* --- End of period stock of capital, equation (76) in VDM 2018
*   Note: The equation accounts for the possible that the simulation period covers several years,
*   Such that investments and deprecitation appear multiple times
*
e_kapEnd(rsNat,ts) ..
    kapEnd(rsNat,ts)/kapend.scale(rsNat,ts)
    =e= [
        (1 - p_depr(rsNat,ts)*p_nDeprYears(ts))*kstock(rsNat,ts) + xi(rsNat,ts)*p_nDeprYears(ts)
    ]/kapend.scale(rsNat,ts);

```

The **average returns of returns to capital after taxes** arent is defined in the equation e\_arent, equation (77) in VDM 2018, from capital prices pft, considering direct taxes kappaf and their potential tax shifter kappashft, and the factor which convert yearly capital use into stock values p\_krat:

```

*
* --- Aggregate capital rate of return after tax, equation (77) in VDM 2018
*   (Note: here, we use always the average across sectors as tax rates are uniform)
*
e_arent(rsNat,ts) $ (rorFlag ne fixedForeignSavings) ..
    arent(rsNat,ts)/arent.1(rsNat,"%t0%")
    =e= [sum((cap) $ xftFlag(rsNat, cap),
        (1-kappaf(rsNat, cap, ts))*pft(rsNat, cap, ts)*xft(rsNat, cap, ts))
        / kstock(rsNat,ts)] /arent.1(rsNat,"%t0%");

```

The **net rate of return to capital** rorc as defined in the equation e\_rorc, equation (77) in VDM 2018, corrects these gross returns factors arent for the depreciation rate p\_fdepr:

```

*
* --- Rate of return to capital, equation (78) in VDM 2018
*
e_rorc(rsNat,ts) $ (rorFlag ne fixedForeignSavings) ..
    rorc(rsNat,ts)/rorc.1(rsNat,"%t0%") =e= (arent(rsNat,ts)/pi(rsNat,ts) - p_fdepr(rsNat,ts))/rorc.1(rsNat,"%t0%");

```

The **expected net returns to capital** rore take the change in end of period capital stock kapEnd to beginning of period stock kStock exponent the elasticity – p\_RorFlex into account, considering the number of deprecitaions years p\_NDeprYears multiplied with the net rate of return to capital rorc, defined in the two equations e\_rorePart1 and e\_ror, equation (79) in VDM 2018,

```

*
* --- Intermediate term in equation (79) of VDM 2018, considering multiple years of simulation
*
e_rorePart1(rsNat,ts) $ (RorFlag ne fixedForeignSavings) ..
    rorePart1(rsNat,ts)/rorePart1.scale(rsNat,ts)
    =E= (kapEnd(rsNat,ts)/kstock(rsNat,ts))**(-p_RorFlex(rsNat,ts)+p_nDeprYears(ts)-1)/rorePart1.scale(rsNat,ts);
*
* --- Expected rate of return, equation (79) in VDM 2018
*
e_rore(rsNat,ts) $ (RorFlag ne fixedForeignSavings) ..
    rore(rsNat,ts)/rore.1(rsNat,"%t0%") =e= rorc(rsNat,ts)/rore.1(rsNat,"%t0%") * rorePart1(rsNat,ts);

```

As shown above, two equations are used to define this relation to avoid numerical problems in the solver.

Specificially, the global bank mechanism aims at equalizing the expected net returns rore to savings across regions by changing the distribution of foreign savings fsav to the different regions. The mechanism, as detaited above, considers returns to the (future) capital stock, corrected for taxation of capital returns and depreciation. Equally, investors dampen their expectation according to p\_rorFlex for future increases of the capital relative to the current one. This reflects that an increase in the capital stock, all things equal, will decrease capital returns as other factors will get relatively more scarce.

In the benchmark, a risk parameter  $p\_risk$  ensures that the average global return  $rorg$  are lined up with the expected returns in each region  $rore$ . Assume now that the price of mobile capital in a region after direct taxes increases, e.g. by tax reform. That will increase the expected returns in this region and thus attract foreign saving. Increasing the foreign savings in a region will in turn increase total savings and thus investments  $xi$ . That will change the end of period capital  $kapEnd$  which will affect the relation between end and beginning stock and thus decrease the expected rate  $rore$ .

The **value of global net investment**  $gblValNetInv$  as defined in the equation  $ge\_gblValNetInv$  is derived from the regional net investments  $netInv$ , their regional prices  $pi$  and the exchange rate  $lcu$ , part of equation (82) in VDM 2018:

```
*
* --- value of global net investments in current prices, part of equation (82) in VDM 2018
*
e_gblValNetInv(ts) ..
    gblValNetInv(ts)/sum(rNat,netInv.scale(rNat,ts))
    =E= sum(rNat, (pi(rNat,ts)/lcu(rNat,ts) *netInv(rNat,ts)) $ rsNat(rNat)
        + (pi.l(rNat,ts)/lcu.l(rNat,ts) *netInv.l(rNat,ts)) $ (not rsNat(rNat)))
        /sum(rNat,netInv.scale(rNat,ts));
```

Note the distinction between regions in the current solve  $rs$  or not. For the latter, investment prices and quantities are fixed. This reflects the case where the model is run in single country mode or during pre-processing where a sub-model is solved for one region, only.

**Regional net investments**  $netInv$  as defined in the  $e\_netInv$ , part of equation (86) in VDM 2018, are equal to aggregated gross investment demand  $xi$  minus depreciation, i.e. the depreciation rates  $p\_fdepr$  times the beginning of year capital stocks  $kStock$ :

```
*
* --- Physical net investment in each region, part of equation (86) in VDM 2018
*
e_netInv(rsNat,ts) ..
    netInv(rsNat,ts)/netInv.scale(rsNat,ts) =E= [xi(rsNat,ts) - p_depr(rsNat,ts)*kstock(rsNat,ts)]/netInv.scale(rsNat,ts);
```

The **total global net investments**  $xigbl$  are defined in the equation  $e\_xigbl$  by up the regional net investments  $netInv$ , equation (86) in VDM 2018:

```
*
* --- Global net investment, equation (86) in VDM 2018
*
e_xigbl(ts) ..
    xigbl(ts)/sum(rNat, netInv.scale(rNat,ts))
    =E= sum(rNat,
        netInv(rNat,ts) $ rsNat(rNat)
        ---add values of countries not in current solve, but in the global framework
        + netInv.l(rNat,ts) $ (not rsNat(rNat))
    )/sum(rNat,netInv.scale(rNat,ts));
```

The **average global expected returns to capital**  $rorg$  is the value – net investment  $netInv$  times saving prices  $pi$  – weighted average of the regional expected returns  $rore$  as defined in the two equations  $e\_rorg$  and  $e\_gblValNetInv1$ :

```
*
* --- average average expected return, equation (81) in VDM 2018
*   in case of fixed allocation of investments: average of regional expected rates
*   rorg and rorg should with fixed allocation not influence the solution behavior
*
e_rorg(ts) $ (RoRFlag eq fixedAllocationOfInv) ..
    rorg(ts)/rorg.scale(ts) =E= gblValNetInv1(ts) / gblValNetInv(ts) * 1/rorg.scale(ts);
```

```

*
* --- value of global net investments in current prices, part of equation (82) in VDM 2018
*
e_gblValNetInv1(ts) $ (RoRFlag eq fixedAllocationOfInv) ..

    gblValNetInv1(ts)/sum(rNat,netInv.scale(rNat,ts))
    =E= sum(rNat, (rore(rNat,ts) * pi(rNat,ts)/lcu(rNat,ts) * netInv(rNat,ts)) $ rsNat(rNat)
        + (rore.l(rNat,ts) * pi.l(rNat,ts)/lcu.l(rNat,ts) * netInv.l(rNat,ts)) $ (not rs(rNat)))
        /sum(rNat,netInv.scale(rNat,ts)) ;

```

The **distribution of the net investments in case of the global bank mechanism** (RoRFlag eq equalReturnToInv) is steered by the first part of the following e\_savf, equation (80) in VDM 2018, which requires for each region that risk adjusted expected returns rore are equal to the global average rorg:

```

*
* --- Determination of foreign savings, equation (80) in VDM 2019
*
e_savf(rsNat,ts) $ ( (RoRFlag eq equalReturnToInv)
    or ((RoRFlag eq fixedAllocationOfInv) $ (not rres(rsNat))))
    $ (savf.range(rsNat,ts) ne 0) ..

0 =e=

*
* --- equal relative changes in expected returns to investments
* across regions, rorg in the global currency
*
    ([rore(rsNat,ts)*p_risk(rsNat,ts) - rorg(ts)]/rorg.l("%t0%")) $ (RoRFlag eq equalReturnToInv)
*
* --- fixed allocation of global investment based on parameter chiInv
*
    + [(netInv(rsNat,ts) - chiInv(rsNat,ts)*xigbl(ts))/netInv.scale(rsNat,ts)] $ (RoRFlag eq fixedAllocationOfInv);

```

### 2.11.2 Fixed allocation of savings

The case of fixed allocation shares of savings is depicted in the second block of the equation e\_savf if RoRFlag is set equal to “fixedAllocationOfInv” and is based on parameters chiInv which reflect the benchmark distribution. Note that the mechanism refers to “capShrFix” in VDM 2018:

```

*
* --- Determination of foreign savings, equation (80) in VDM 2019
*
e_savf(rsNat,ts) $ ( (RoRFlag eq equalReturnToInv)
    or ((RoRFlag eq fixedAllocationOfInv) $ (not rres(rsNat))))
    $ (savf.range(rsNat,ts) ne 0) ..

0 =e=

*
* --- equal relative changes in expected returns to investments
* across regions, rorg in the global currency
*
    ([rore(rsNat,ts)*p_risk(rsNat,ts) - rorg(ts)]/rorg.l("%t0%")) $ (RoRFlag eq equalReturnToInv)
*
* --- fixed allocation of global investment based on parameter chiInv
*
    + [(netInv(rsNat,ts) - chiInv(rsNat,ts)*xigbl(ts))/netInv.scale(rsNat,ts)] $ (RoRFlag eq fixedAllocationOfInv);

```

Note that the residual region is excluded from the mechanism. Its foreign saving is defined by the constraint that the global capital account balance must be equal to zero.

### 2.11.3 Capital account and balance of payments

The global capital account balance e\_capAcct, equation (84) in VDM 2018, ensures that the global sum of the foreign savings savf is zero:

```

*
* --- Capital account balance: sum of foreign savings over regions is zero,
* defines implicitly regional income of residual regions, equation (84) in VDM 2018
*
e_capAcct(ts) $ ( (sum(rsNat $ (savf.range(rsNat,ts) ne 0),1) gt 1)
    or ( (RoRFlag eq fixedForeignSavings) and sum(rres(rsNat),1)
        and (pnun.range(ts) gt 0) or (not singleCountry) ))) ..

0 =e= sum(rNat, savf(rNat,ts) $ rsNat(rNat) + savf.l(rNat,ts) $ (not rsNat(rNat))
    - p_capTrans(rNat,ts)*pnun(ts) + finc(rNat,ts))
    /sum(rres,savf.scale(rres,ts));

```



The equation is only active if (1) there are at least two countries in the current solve where foreign savings are not fixed or (2) the residual region is in the current solve and the foreign savings for the other regions are fixed and the global model is used (not singleCountry) or the numeraire is not fixed.

The balance of payments equation `e_bop` is only a check for the correct setup of the model, i.e. the `bopSlack` should be equal to zero given the accuracies of the solver and original tiny numerical imbalances in the SAM:

```
*
* --- Test for BOP, not part of GTAP standard
*
e_bop(rsNat(rNat),ts) ..
    (valsavf(rNat,ts) + bopSlack(rNat,ts)*regy(rNat,ts) - p_capTrans(rNat,ts)*pnun(ts) + finc(rNat,ts))
    /regy.scale(rNat,ts)
    =E= [ - valFobCif(rNat,ts)
          - sum( (m,tmg) $ xaFlag(rNat,m,tmg), xa(rNat,m,tmg,ts)*m_pa(rNat,m,tmg,ts)*lcu(rNat,ts))
        ]
    /regy.scale(rNat,ts)
;
```

The `bopslack` variable should be zero due to B.O.T. == B.O.P.. In the model, this is reflected by two definitions. First, total final demand consists of regional income `regy` (household and government demand plus the share of gross savings financed from regional income `rsav`) and foreign savings `fsav` (= B.O.P in the standard GTAP model), where the foreign savings plus depreciation provide the residual income (positive or negative) to exhaust total final investment demand. Second, `regy` income comprises additional (or missing) indirect tax and factor income from a trade surplus (or deficite), i.e. the B.O.T.

## 2.12 Demand sub-nests

### 2.12.1 Motivation

Computable General Equilibrium (CGE) model use regular demand systems to depict final demand for goods and services, typically differentiated by demand from (multiple) household(s), government and investors. For government and investors, often a Cobb-Douglas or Leontief functional form is employed, while for households, Cobb-Douglas (CD) preferences or Linear Expenditure Systems (LES) are frequently used. Due to often relatively small constant (or so-called commitment) terms in the LES systems, the behavior of the CD and LES systems under price and income changes is often not much different. When the constant terms approach zero, a LES demand system exhibits income and own-price elasticities close to unity and cross-price elasticities close to zero, see also Ho et al. 2020. The resulting linear Engel curves, start at or close to the origin, don't capture structural changes on the demand side, such as regarding Engel's law. Here, either ad-hoc changes in parameters during simulation are introduced or a change in the functional form is needed.

The GTAP Standard model uses a CDE (Constant Difference in Elasticity) demand system (Hanoch 1975) of which the parameters are derived from an AIDADS demand system estimation by Hertel and Reimer 2004. Its three vectors of parameters allow, considering that one vector is needed for benchmarking, to calibrate to both own and income elasticities. However, the fixed expansion elasticities of the CDE demand system are also not well suited to depict long-term dynamics in budget shares.

CGEBox (Britz and Van der Mensbrugghe 2016) comprises code for benchmarking and simulation for the demand system mentioned above. Introduced in the context of its long-run component G-RDEM (Britz and Roson 2019, Roson and Britz 2021), CGEBox makes a step further and comprises as an option an empirically estimated MAIDADS demand system (Britz 2021) which exhibits exponential

Engel Curves. The MAIDADS system is the recommended option for long-run analysis in CGEBox and can be combined with an option ensuring plausible development of calorie intakes (Britz 2020).

None of the demand system above is flexible enough to differentiate cross-price elasticities. Using second-order flexible demand systems such as the Generalized Leontief Demand System (cf. Diewert and Wales 1989) employed in the Partial Equilibrium model CAPRI (Britz and Witzke 2014) is unusual in CGE analysis. Instead, following the structure of the production functions, CGE models employ nested demand systems where the lower nests are of the CES (Constant-Elasticity of Substitution) type. This implies expenditures elasticity of unity for commodity or services bundled in one nest, such that their income elasticities depend on the aggregate commodity they are linked to in the top-level demand system.

### 2.12.2 Implementation in model

Demand sub-nests aggregate individual Armington demands in final demand by private household, government or savings to aggregates based on CES utility function. They are not part of the GTAP Standard Model. The resulting nests can be either part of the top level demand function or linked into other demand nests. That mechanism increases the flexibility of depicting substitution relations between individual products in final demand beyond what the top-level final demand system offers, similar to the flexible nesting in the production function.

The **average price for a demand nest**  $pdNest$  for a such a nest  $dNest$  and the final demand agent  $fdn$  is defined by dual price aggregators which distinguish the CD from the CES/Leontief case depending on the substitution elasticity  $p\_sigmaFDNest$ . In both cases, the price index reflects the contribution of individual products  $i$  based on their share parameter  $alphaa$  and the contribution of sub-nests based on their share parameter  $p\_alphaDN$ .

```

p_dNest(rs,dnest,fdn,ts) $ sum(allDNests,p_alphaDN(rs,dNest,fdn,allDNests,ts)) ..

pdNest(rs,dNest,fdn,ts)
==
    --- CD case with p_sigmaFDNest==1
    ( [
        ---- contribution of individual commodity demands to price aggregator
        prod(dNest_i_fd(dNest,i,fdn) $ alphaa.l(rs,i,fdn,dNest,ts),
            (m_pa(rs,i,fdn,ts)/(alphaa(rs,i,fdn,dNest,ts)*(1+(lambdai(rs,i,ts)-1)$ sameas(fdn,"inv")))**alphaa(rs,i,fdn,dNest,ts))

        ---- contribution of sub-nests under the current nest to price aggregator
        * prod(dNest_n_fd(dNest,dNest1,fdn) $ p_alphaDN(rs,dNest1,fdn,dNest,ts),
            (pdNest(rs,dNest1,fdn,ts)/p_alphaDN(rs,dNest1,fdn,dNest,ts))**p_alphaDN(rs,dNest1,fdn,dNest,ts))

        * p_axdNestCD(rs,dNest,fdn,ts)

    ] $ (p_sigmaFDNest(rs,dNest,fdn) eq 1)

    --- CES or Leontief case with p_sigmaFDNest<>1
    + [
        { sum(dNest_i_fd(dNest,i,fdn) $ alphaa.l(rs,i,fdn,dNest,ts),
            alphaa(rs,i,fdn,dNest,ts) * (m_pa(rs,i,fdn,ts)/(1+(lambdai(rs,i,ts)-1)$ sameas(fdn,"inv")))**(1-p_sigmaFDNest(rs,dNest,fdn))
        )
        + sum(dNest_n_fd(dNest,dNest1,fdn) $ p_alphaDN(rs,dNest1,fdn,dNest,ts),
            p_alphaDN(rs,dNest1,fdn,dNest,ts) * pdNest(rs,dNest1,fdn,ts)**(1-p_sigmaFDNest(rs,dNest,fdn))
        )
        } ** (1/(1-p_sigmaFDNest(rs,dNest,fdn)))

    ] $ (p_sigmaFDNest(rs,dNest,fdn) ne 1)

] $ ( not singleDNest(rs,dNest,fdn))

--- case where only one product or nest is left
+ [
    sum(dNest_i_fd(dNest,i,fdn) $ alphaa.l(rs,i,fdn,dNest,ts), m_pa(rs,i,fdn,ts)/(1+(lambdai(rs,i,ts)-1)$ sameas(fdn,"inv")))
    + sum(dNest_n_fd(dNest,dNest1,fdn) $ p_alphaDN(rs,dNest1,fdn,dNest,ts),pdNest(rs,dNest1,fdn,ts))] $ singleDNest(rs,dNest,fdn);

```

```

*
*      --- CES or Leontief case with p_sigmaFDNest<=1
*
+ [
  { sum(dNest_i_fd(dNest,i,fdn) $ alphaa.1(rs,i,fdn,ts),
    alphaa(rs,i,fdn,ts) * (m_pa(rs,i,fdn,ts)/(1+(lambdai(rs,i,ts)-1) $ sameas(fdn,"inv")))**(1-p_sigmaFDNest(rs,dNest,fdn))
  )
  + sum(dNest_n_fd(dNest,dNest1,fdn) $ p_alphaDN(rs,dNest1,fdn,ts),
    p_alphaDN(rs,dNest1,fdn,ts) * pdNest(rs,dNest1,fdn,ts)**(1-p_sigmaFDNest(rs,dNest,fdn))
  )
  } ** (1/(1-p_sigmaFDNest(rs,dNest,fdn)))
] $ (p_sigmaFDNest(rs,dNest,fdn) ne 1) } $ ( ( sum(dNest_i_fd(dNest,i,fdn) $ alphaa.1(rs,i,fdn,ts),1)
+sum(dNest_n_fd(dNest,dNest1,fdn) $ p_alphaDN(rs,dNest1,fdn,ts),1)) gt 1)
*
*      --- case where only one product or nest is left
*
+ [
  sum(dNest_i_fd(dNest,i,fdn) $ alphaa.1(rs,i,fdn,ts), m_pa(rs,i,fdn,ts)/(1+(lambdai(rs,i,ts)-1) $ sameas(fdn,"inv")))
  + sum(dNest_n_fd(dNest,dNest1,fdn) $ p_alphaDN(rs,dNest1,fdn,ts),pdNest(rs,dNest1,fdn,ts))]
  $ ( ( sum(dNest_i_fd(dNest,i,fdn) $ alphaa.1(rs,i,fdn,ts),1)
+sum(dNest_n_fd(dNest,dNest1,fdn) $ p_alphaDN(rs,dNest1,fdn,ts),1)) le 1);

```

In mathematical notation:

$$PdNest_{r,dNest,fdn} = \left[ \sum_{i \in dNest} \alpha_{r,i,fdn}^{dNest} \left( \frac{PA_{r,i,fdn}}{\lambda_{r,i,fdn}} \right)^{\sigma_{fdn}^{dNest}} + \sum_{dNest1 \in dNest} \alpha_{r,dNest1,fdn}^{denst} \left( PdNest_{r,dNest1,fdn} \right)^{\sigma_{fdn}^{dNest}} \right]^{1/(1-\sigma_{fdn}^{dNest})}$$

$fdn = \{gov, inv, h\}$

With the list of final demands

The notation already underlines that such nests can comprise other nests.

The **demand for a sub-nest**  $xdNest$  depends on the total demand top (inv or gov) or is driven by the demand of a sub-nest  $dNest1$ :

```

e_xdNest(rs,dnest,fdn,ts) $ (sum(alldNestS,p_alphaDN(rs,dNest,fdn,alldNests,ts)) and (not sum(dNest_n_fd("top",dnest,h) $ sameas(fdn,h),1))) ..
*
*      --- total demand for sub-nest
*
xdNest(rs,dNest,fdn,ts) * m_var4scale(xdNest,rs,dNest,fdn,ts)/sum(i, xa.scale(rs,i,fdn,ts))
*=
[
*
*      --- driven by top nest demand in case of gov/inv (CES or CD)
*      Final household demand has separate equation to reflect the more complex CDE case
*
+ sum(dNest_n_fd("top",dnest,"gov") $ p_alphaDN(rs,dNest,"gov","top",ts),
  p_alphaDN(rs,dNest,"gov","top",ts) * xg(rs,ts) * m_var2scale(xg,rs,ts) * (pg(rs,ts)/pdNest(rs,dNest,"gov",ts)) ** p_sigmag(rs) $ sameas(fdn,"gov"))
+ sum(dNest_n_fd("top",dnest,"inv") $ p_alphaDN(rs,dNest,"inv","top",ts),
  p_alphaDN(rs,dNest,"inv","top",ts) * xi(rs,ts) * m_var2scale(xi,rs,ts) * (pi(rs,ts)/pdNest(rs,dNest,"inv",ts)) ** p_sigmai(rs) $ sameas(fdn,"inv"))
*
*      -- driven by another nest
*
+ sum(dNest_n_fd(dNest1,dNest,fdn) $ p_alphaDN(rs,dNest,fdn,dNest1,ts),
  p_alphaDN(rs,dNest,fdn,dNest1,ts) * xdNest(rs,dNest1,fdn,ts) * m_var4scale(xdNest,rs,dNest1,fdn,ts)
  * (pdNest(rs,dNest1,fdn,ts)/pdNest(rs,dNest,fdn,ts)) ** p_sigmaFDNest(rs,dNest1,fdn))
]/sum(i, xa.scale(rs,i,fdn,ts));

```

For the final household case with a CDE demand function (see dem\_cde.gms), the following equation is used:

```

e_zconsDn(%rsh%,dNest,h,ts) $ sum(dNest_n_fd("top",dNest,h), p_alphaDN(%rsh%,dNest,h,"top",ts)) ..
zconsDn(%rsh%,dNest,h,ts)/m_var4Scale1(zConsDn,%rsh%,dNest,h,ts) =e= p_alphaDN(%rsh%,dNest,h,"top",ts) * bhDN(%rsh%,dNest,ts)
* (pdNest(%rsh%,dNest,h,ts) ** bhDN(%rsh%,dNest,ts))
* ((yc(%rsh%,h,ts) * m_var3Scale(yc,%rsh%,h,ts)/pop(%rsh%,h,ts) * 1000) ** (-bhDN(%rsh%,dNest,ts)))
* (uh(%rsh%,h,ts) ** (ehDN(%rsh%,dNest,ts) * bhDN(%rsh%,dNest,ts))) / zconsDn.scale(%rsh%,dNest,h,ts);

```

The **Armington demands**  $xa$  **driven by a sub-nest** are defined in the equation  $e_xdNest$  and use the usual CES-structure, i.e. the share parameter  $\alpha$ , the sub-nest total demand  $xdNest$  and the price relation exponent the substitution elasticity  $p\_sigmaFDNest$  as well a preference shifter  $\lambda$  in case of savings demand:



```

*
* --- individual agent s Armington demands under sub-nests
*
* xaldNest(rs,i,fdn,ts) $ (xaFlag(rs,i,fdn) $ sum(dNest_i_fd(dNest,i,fdn),1) $ iIn(i)) ..
*
* xa(rs,i,fdn,ts)/m_var4scale1(xa,rs,i,fdn,ts) =e=
*   sum(dNest_i_fd(dNest,i,fdn),
*     alphaa(rs,i,fdn,dNest,ts)*xdNest(rs,dnest,fdn,ts)*m_var4scale(xdNest,rs,dNest,fdn,ts)/xa.scale(rs,i,fdn,ts)
*
*   --- not the preference shifter lambdai for investment demand
*
*   * [
*
*   +- CES case with price effect
*
*   {
*     ((1+(lambdai(rs,i,ts)-1) $ sameas(fdn,"inv"))*pdNest(rs,dNest,fdn,ts)/m_pa(rs,i,fdn,ts))
*       **p_sigmaFDNest(rs,dNest,fdn)
*       /((1+(lambdai(rs,i,ts)-1) $ sameas(fdn,"inv")))
*   } $ (p_sigmaFDNest(rs,dNest,fdn) ne 0)
*
*   --- Leontief case with fixed composition, no price effect
*
*   + {
*     (1+(lambdai(rs,i,ts)-1) $ sameas(fdn,"inv"))
*   } $ (p_sigmaFDNest(rs,dNest,fdn) eq 0)
*   ];

```

### 2.12.3 Benchmarking of demand sub-nests

In order to benchmark the structure above, the analyst has to define the underlying relations. These definitions of the nesting structure are based on set-definitions, such as in:

```

dNest("non-electric") = YES;
dNest_i_fd("non-electric",coal,fdn) = YES;
dNest_n_fd("non-electric","non-coal",fdn) = YES;
p_sigmaFDNest(r,"non-electric",fdn) = 0.50;

dNest("non-coal") = YES;
dNest_i_fd("non-coal",nonCoal,fdn) = YES;
p_sigmaFDNest(r,"non-coal",fdn) = 1.00;

```

as part of the nestings borrowed from GTAP-Power (Peters 2016)

The demand nesting allows to distribute individual products or nests to multiple nest in shares, such as in:

```

dNest("housing") = YES;
dNest_i_fd("housing","dwe-c",h) = YES;
dNest_n_fd("housing","energy",h) = YES;
dNest_n_fd("top","housing",h) = YES;
dNestShare(r,"housing","energy",h) = 0.5;
dNest("housing") = YES;

```

The benchmarking (*cal/cal\_dnest.gms*) consists to three major steps:

1. Checking that the nesting structures introduced by the analyst is consistent
2. Deriving the share parameters for the sub-nests
3. Modification of the top-level demand system parameters

First some symbols are defined. Nests for the international trade margin demand agent *trdmg* refused which are not supported by the code.

```

alias(*,uDom);
set wrongName;

set dNesting(*,*,fd);
parameter dNestingErrors(r,*,*,fd);

allDNests(dNest) = YES; allDNests("top") = YES;

set nestsToDo,curNest(*);
alias(nestsToDo,nestsToDo1);

set topdNest(*,i,fd);
*
* --- these nests are not supported
*
if(sum(dnest_i_fd(dNest,i,"trdmg"),1),
  abort "Demand sub-nests for 'trdmg' are not supported",dnest_i_fd);

if(sum(dnest_n_fd(dNest,dNest1,"trdmg"),1),
  abort "Demand sub-nests for 'trdmg' are not supported",dnest_n_fd);

if(sum(dnest_n_fd("top",dNest1,"trdmg"),1),
  abort "Demand sub-nests for 'trdmg' are not supported",dnest_n_fd);

```

The reminder of the code is executed only if nests are present:

```

if (card(dNest),

```

Otherwise, all demand is assigned to the top nest with shares of unity:

```

dNestShare(rs,"top",i,fdn) $ xa.1(rs,i,fdn,"%t0%") = 1;
dNest_i_fd("top",i,fdn) = yes;

```

### Checks during benchmarking

The first part of the checks uses the universal domain to make sure that only element names found in the dynamic set *allDNests* are used. The use of a dynamic set allows to add flexibly nests to the model at compile time. Using compile time statement is important to reflect the detail of different data bases. For instance, different electricity types are only present if the GTAP-Power Data Base is used in appropriate differentiation.

The tests are coded as follows:

```

:
: *
: * --- check for typos, as dest is not domain checked to allow for flexible includes
: *
:
: wrongName(uDom) $ sum(dNest_i_fd(uDom,i,fdn) $ (not allDNests(uDom)),1) = YES;
: if ( card(wrongName),
:   abort "Typo in dNest_i_fd on first position, not a final demand sub-nest",dNest_i_fd,dNest,wrongName);
:
: wrongName(uDom) $ sum(dNest_n_fd(dNest1,uDom,fdn) $ (not allDNests(uDom)),1) = YES;
: if ( card(wrongName),
:   abort "Typo in dNest_n_fd on second position, not a final demand sub-nest",dNest_n_fd,wrongName);
:
: wrongName(uDom) $ sum(dNest_n_fd(uDom,dNest,fdn) $ (not allDNests(uDom)),1) = YES;
: if ( card(wrongName),
:   abort "Typo in dNest_n_fd on first position, not a final demand sub-nest",dNest_n_fd,wrongName);
: *

```

Next, we define a parameter which shows the nesting structure:

```

*
* --- build tNesting structure
*
dNesting("top",i,fdn) $ (not sum(dNest_i_fd(dNest,i,fdn),1)) = YES;
dNesting(allDNests,dNest,fdn) $ dNest_n_fd(allDNests,dNest,fdn) = YES;
dNesting(allDNests,i,fdn) $ dNest_i_fd(allDNests,i,fdn) = YES;

```

And assign shares of unity if not defined in the code:

```
*
* --- set nest shares to unity if not set by user
*
dNest_i_fd("top",i,fdn) $ (not sum(dNest_i_fd(dNest,i,fdn),1)) = YES;
dNestShare(rs,alldNests,i,fdn) $ (dNest_i_fd(alldNests,i,fdn) $ (not dNestShare(rs,alldNests,i,fdn) $ (xaFlag(rs,i,fdn) or xaFlag(rs,fdn,i))) = 1;
dNestShare(rs,alldNests,dNest,fdn) $ (dNest_n_fd(alldNests,dNest,fdn) $ (not dNestShare(rs,alldNests,dNest,fdn))) = 1;
```

The following two checks ensure that the distribution shares add up to unity. They will hence also capture cases where nests are not correctly integrated in the overall structure and are dangling:

```
*
* --- Check that each commodity is fully distributed (shares add up to unity)
*
dNestingErrors(rs,dNest,i,fdn) = sum(dNesting(alldNests,i,fdn),dNestShare(rs,alldNests,i,fdn));
dNestingErrors(rs,dNest,i,fdn) $ (dNestingErrors(rs,dNest,i,fdn) eq 1) = 0;
abort $ card(dNestingErrors)
"Error: Final demand for a commodity in final demand sub-nests is not correctly distributed (not at all or shares don't add up to unity)",
dNestingErrors,dNesting;

*
* --- Check that each nest is fully distributed (shares add up to unity)
*
dNestingErrors(rs,dNest,dNest1,fdn) = sum(dNesting(alldNests,dNest1,fdn),dNestShare(rs,alldNests,dNest1,fdn));
dNestingErrors(rs,dNest,dNest1,fdn) $ (dNestingErrors(rs,dNest,dNest1,fdn) eq 1) = 0;
abort $ card(dNestingErrors)
"Error: Final demand for a sub-nests in final demand sub-nests is not correctly distributed (not at all or shares don't add up to unity)",
dNestingErrors,dNesting;
```

### *Sequential assignment of share parameters during benchmarking*

The share parameters in the nest must be defined bottom-up, from the lowest nests which can comprise single products, only, to the top-level demand system. In the beginning, no nest has already properly defined price and quantity indices:

```
nestsToDo(dNest) = YES;
option kill=xdNest.1;
```

We iterate until the set of not yet defined nests is empty and define in each iteration first the current nests to work on:

```
-
While( card(nestsToDo),

    --- we can calculate all nests which do not require information on nests
    not yet calculated

    option kill=curNest;
    curNest(nestsToDo) $ (not sum(dNest_n_fd(nestsToDo,nestsToDo1,fdn),1)) = YES;
```

At the end of this while statement, the now properly defined nests are taken from the set of nests to work on:

```
*
* --- and delete from lists of nests to process
*
nestsToDo(curNest) = NO;
```

As Armington prices are defined as unity at the benchmark, the resulting CES price indices for the demand nests are unity as well:

```
*
*
*
pdNest.1(rs,curNest,fdn,t0) = 1;
```

The quantity indices for the nests consider the demand quantities  $xa$  of single products  $i$  mapped to them and demand quantities of demand sub-nests  $xdNest$ , in both cases considering the distribution shares  $dNestShare$ :

```

--- calculate quantity index

xdNest.l(rs,curNest,fdn,t0)
= [    sum(dNest_i_fd(curNest,i,fdn),
          xa.l(rs,i,fdn,t0)*pa.l(rs,i,fdn,t0)*dNestShare(rs,curNest,i,fdn))

    + sum(dNest_n_fd(curNest,dNest1,fdn),
          xdNest.l(rs,dNest1,fdn,t0)*pdNest.l(rs,dNest1,fdn,t0)*dNestShare(rs,curNest,dNest1,fdn))

    ] / pdNest.l(rs,curNest,fdn,t0);

pdNest.l(rs,curNest,fdn,t0) $ (not xdNest.l(rs,curNest,fdn,t0)) = 0;

```

The share parameters for single products are defined as usual for the CES functional form as follows:

```

--- calculate cost share parameters for products in current nest

alphaa.fx(rs,i,fdn,curNest,t0) $ sum(dNest_i_fd(curNest,i,fdn), xa.l(rs,i,fdn,t0))
= sum(dNest_i_fd(curNest,i,fdn), xa.l(rs,i,fdn,t0)*dNestShare(rs,curNest,i,fdn)/xdNest.l(rs,curNest,fdn,t0)
    * (pa.l(rs,i,fdn,t0)/pdNest.l(rs,curNest,fdn,t0))**p_sigmaFDNest(rs,curNest,fdn)) ;

```

Similar, for sub-nests:

```

--- calculate cost share parameters for sub-nests in current nest

p_alphaDN(rs,dNest,fdn,curNest,t0) $ sum(dNest_n_fd(curNest,dNest,fdn) $ (xdNest.l(rs,curNest,fdn,t0) and xdNest.l(rs,dNest,fdn,t0)),1)
= sum(dNest_n_fd(curNest,dNest,fdn), xdNest.l(rs,dNest,fdn,t0)*dNestShare(rs,curNest,dNest,fdn)/xdNest.l(rs,curNest,fdn,t0)
    * (pdNest(rs,dNest,fdn,t0)/pdNest.l(rs,curNest,fdn,t0))**p_sigmaFDNest(rs,curNest,fdn)) ;

```

In case of CD relation (substitution elasticity is unity), a multiplier is needed as seen in the model equation above:

```

p_axdNestCD(rs,curNest,fdn,t0) $ (xdNest.l(rs,curNest,fdn,t0) $ (p_sigmaFDNest(rs,curNest,fdn) eq 1))
= pdNest.l(rs,curNest,fdn,t0) /
  [    prod(dNest_i_fd(curNest,i,fdn) $ alphaa(rs,i,fdn,curNest,t0),
          (pa.l(rs,i,fdn,t0)/alphaa(rs,i,fdn,curNest,t0))**alphaa(rs,i,fdn,curNest,t0))

    * prod(dNest_n_fd(curNest,dNest1,fdn) $ p_alphaDN(rs,dNest1,fdn,curNest,t0),
          (pdNest(rs,dNest1,fdn,t0)/p_alphaDN(rs,dNest1,fdn,curNest,t0))**p_alphaDN(rs,dNest1,fdn,curNest,t0))

    ];

```

### *Modification of the top-level demand system parameters and other final steps*

In order to reduce the computational burden, cases where a nest in a region comprises one product or sub-nest, only are treated differently in the code (see above). This is captured by the set “singleDNest” (see also model equation above):

```

singleDNest(rs,dNest,fdn) $ ( ([    sum(dNest_n_fd(dNest,dNest1,fdn) $ xdNest.l(rs,dNest1,fdn,"%t0%"),1)
    + sum(dNest_i_fd(dNest,i,fdn) $ xa.l(rs,i,fdn,"%t0%"),1) ] eq 1) $ xdNest.l(rs,dNest,fdn,"%t0%")) = YES;

p_alphaDN(rs,dNest1,fdn,dNest,"%t0%") $ sum(dNest_n_fd(dNest,dNest1,fdn) $ (singleDNest(rs,dNest,fdn) $ xdNest.l(rs,dNest1,fdn,"%t0%")),1) = 1;
alphaa.fx(rs,i,fdn,dNest,"%t0%") $ sum(dNest_i_fd(dNest,i,fdn) $ (singleDNest(rs,dNest,fdn) $ xa.l(rs,i,fdn,"%t0%")),1) = 1;

p_sigmaFDNest(rs,dNest,fdn) $ singleDNest(rs,dNest,fdn) = 0;

```

For the CDE demand system which is the default case, the expansion and substitution parameters for sub-nests are defined as follows:

```

option kill=bhdn.l,kill=ehdn.l;

nestsToDo(dNest) = YES;
While( card(nestsToDo),
    option kill=curNest;
    curNest(nestsToDo) $ (not sum(dNest_n_fd(nestsToDo,nestsToDo),h),1) = YES;
*
*   --- aggregate parameters, using expenditures as weights
*
    bhdn.fx(rs,curNest,t0)
        $ ( sum(dNest_i_fd(curNest,i,h), xcshr.l(rs,i,h,t0)*yc.l(rs,h,t0)*dNestShare(rs,curNest,i,h))
            + sum(dNest_n_fd(curNest,dNest1,h), xdNest.l(rs,dNest1,h,t0)*dNestShare(rs,curNest,dNest1,h)) )
        = ( sum(dNest_i_fd(curNest,i,h), xcshr.l(rs,i,h,t0)*yc.l(rs,h,t0)*dNestShare(rs,curNest,i,h)*bh.l(rs,i,t0))
            + sum(dNest_n_fd(curNest,dNest1,h), bhdn.l(rs,dNest1,t0)*xdNest.l(rs,dNest1,h,t0)*dNestShare(rs,curNest,dNest1,h)) )
        / ( sum(dNest_i_fd(curNest,i,h), xcshr.l(rs,i,h,t0)*yc.l(rs,h,t0)*dNestShare(rs,curNest,i,h))
            + sum(dNest_n_fd(curNest,dNest1,h), xdNest.l(rs,dNest1,h,t0)*dNestShare(rs,curNest,dNest1,h)) );

    ehdn.fx(rs,curNest,t0)
        $ ( sum(dNest_i_fd(curNest,i,h), xcshr.l(rs,i,h,t0)*yc.l(rs,h,t0)*dNestShare(rs,curNest,i,h))
            + sum(dNest_n_fd(curNest,dNest1,h), xdNest.l(rs,dNest1,h,t0)*dNestShare(rs,curNest,dNest1,h)) )
        = ( sum(dNest_i_fd(curNest,i,h), xcshr.l(rs,i,h,t0)*yc.l(rs,h,t0)*eh.l(rs,i,t0)*dNestShare(rs,curNest,i,h))
            + sum(dNest_n_fd(curNest,dNest1,h), ehdn.l(rs,dNest1,t0)*xdNest.l(rs,dNest1,h,t0)*dNestShare(rs,curNest,dNest1,h)) )
        / ( sum(dNest_i_fd(curNest,i,h), xcshr.l(rs,i,h,t0)*yc.l(rs,h,t0)*dNestShare(rs,curNest,i,h))
            + sum(dNest_n_fd(curNest,dNest1,h), xdNest.l(rs,dNest1,h,t0)*dNestShare(rs,curNest,dNest1,h)) );

    nestsToDo(curNest) = NO;
);

```

Based on this and the given parameters for the single products, the share parameters of the CDE demands system are defined as:

```

xcshrDN.l(rs,dnest,h,t0) $ sum(dNest_n_fd("top",dNest,h), xdNest.l(rs,dNest,h,t0)*dNestShare(rs,"top",dNest,h))
= xdNest(rs,dNest,h,t0)*dNestShare(rs,"top",dNest,h)*pdNest(rs,dNest,h,t0)/yc.l(rs,h,t0);

alphaa.fx(rs,i,h,"top",t0) $ (xaFlag(rs,i,h) and sum(dNest_i_fd("top",i,h),1))
= ((xcshr.l(rs,i,h,t0)*dNestShare(rs,"top",i,h)/bh.l(rs,i,t0))
* ((yc.l(rs,h,t0)/pop.l(rs,h,t0)*1000)/pa.l(rs,i,h,t0))*bh.l(rs,i,t0))
* (uh.l(rs,h,t0)*(-eh.l(rs,i,t0)*bh.l(rs,i,t0)))
/ ( sum(dNest_i_fd("top",j,h), xcshr.l(rs,j,h,t0)*dNestShare(rs,"top",j,h)/bh.l(rs,j,t0))
+ sum(dNest1 $ bhdn.l(rs,dnest1,t0), xcshrDN.l(rs,dNest1,h,t0)/bhdn.l(rs,dnest1,t0)) );

p_alphaDN(rs,dNest,h,"top",t0) $ sum(dNest_n_fd("top",dNest,h), xdNest.l(rs,dNest,h,t0)*dNestShare(rs,"top",dNest,h))
= ((xcshrDN.l(rs,dnest,h,t0)/bhdn.l(rs,dnest,t0))
* ((yc.l(rs,h,t0)/pop.l(rs,h,t0)*1000)/pdnest.l(rs,dnest,h,t0))*bhdn.l(rs,dnest,t0))
* (uh.l(rs,h,t0)*(-ehDN.l(rs,dnest,t0)*bhdn.l(rs,dnest,t0)))
/ ( sum(dNest_i_fd("top",j,h), xcshr.l(rs,j,h,t0)*dNestShare(rs,"top",j,h)/bh.l(rs,j,t0))
+ sum(dNest1 $ bhdn.l(rs,dnest1,t0), xcshrDN.l(rs,dNest1,h,t0)/bhdn.l(rs,dnest1,t0)) );

```

## 2.12.4 References

Britz, W. (2016). CGEBox model documentation. *University Bonn, Institute for Food and Resource Economics*. [http://www.ilr.uni-bonn.de/em/rsrch/cgebox/cgebox\\_GUI.pdf](http://www.ilr.uni-bonn.de/em/rsrch/cgebox/cgebox_GUI.pdf).

Britz, W. (2020): Maintaining plausible calorie intakes, crop yields and crop land expansion in long-run simulations with Computable General Equilibrium Models. Discussion Paper 2020:2, University Bonn, Institute for Food and Resource Economics, <https://ageconsearch.umn.edu/record/302922>.

Britz, W. (2021): Estimating a global MAIDADS demand system considering demography, climate and norms, *Bio-based and Applied Economics* 10(3): 219-238

Britz, W., & Witze, P. (2014). CAPRI model documentation, version 2014. University Bonn.

Britz, W., Roson, R. (2019): G-RDEM: A GTAP-Based Recursive Dynamic CGE Model for Long-Term Baseline Generation and Analysis, *Journal of Global Economic Analysis* 4(1): 50-96

Diewert, W. E., & Wales, T. J. (1989). Flexible functional forms and global curvature conditions, NBER Technical Paper Series No. 40, [https://www.nber.org/system/files/working\\_papers/t0040/t0040.pdf](https://www.nber.org/system/files/working_papers/t0040/t0040.pdf)

Hanoch, G. (1975). Production and demand models with direct or indirect implicit additivity. *Econometrica: Journal of the Econometric Society*, 395-419

Hertel, Thomas & Reimer, Jeffrey. (2004). Estimation of International Demand Behavior for Use with Input–Output Based Data. *Economic Systems Research*. 16. 347-366. 10.1080/0953531042000304245.

Ho, M., Britz, W., Delzeit, R., Leblanc, F., Roson, R., Schuenemann, F., Weitzel, M. (2020): Modelling Consumption and Constructing Long-Term Baselines in Final Demand, *Journal of Global Economic Analysis* 5(1): 63-80

Peters, J. C. (2016). GTAP-E-Power: an electricity-detailed economy-wide model. *Journal of Global Economic Analysis*, 1(2), 156-187

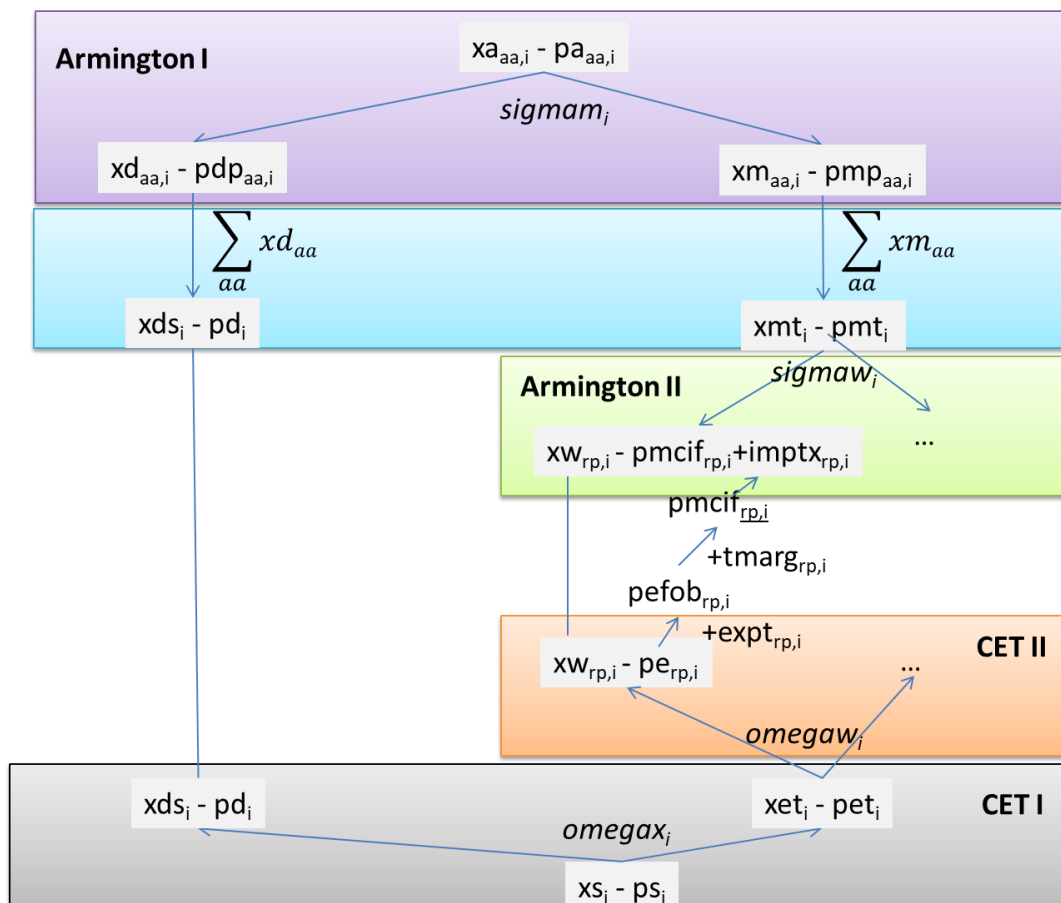
Roson, R., Britz, W. (2021): Simulating long run structural change with a dynamic general equilibrium model, *Int. J. Computational Economics and Econometrics* 11(4): 368-404

## **2.13 International trade and domestic sales, and related prices**

### **2.13.1 Overview**

The graphic below depicts these main relations in international trade and the distribution of supply. The top level Armington nest in the uppermost box distributes the Armington demand for each agent to demand for the domestic and imported origin. Next, the demands for the imported and domestic origin are aggregated over the agents. The total import demand is then split up into demands of the different exporters  $xw$ , driven by the c.i.f. price plus import taxes. The difference between the c.i.f. and f.o.b. price are the endogenous transport margins. Taking export taxation or subsidization into account, the export prices  $pe$  in each exporter region are derived.

Distribution of supply  $xs$  to total exports  $xet$  and domestic sales  $xds$  is driven by a transformation nest depicted in the lowest box. The transformation can also be infinite as the default case. Distribution of the total exports  $xet$  to different destination  $xw$  is handled by a second transformation nest, again, with infinite transformation as the default case.



**Figure 3: Overview on distribution of supply and sourcing of demand**

Note that the set of Armington agents depends on the chosen structure of the model:

1. In the default layout, it comprises the list of sector, one aggregate private household, government and investment demand.
2. Alternative, that differentiation can be completely removed, i.e. the shares in the upper nests are identical for each agent, or can be defined identical across sectors.

The MRIO extension introduces an own set of equations (see sub-section “Model equations”) which allows to dis-aggregate also the lower Armington nests, i.e. the bi-lateral demands, by agent. Furthermore, note that the Melitz and Krugmann extensions use one nest only considering love-of-variety.

### 2.13.2 Individual equations

The **agent specific prices for imports**  $pmp$  as defined in the equation  $e\_pmp$ , equation (46) in VDM 2018, reflects the average import price  $pmt$  and agent specific import taxes  $mintx$  plus tax shifts  $itxshft$  and emission taxes:

```
*
* --- Price for the imported origin, equation (46) in VDM 2018
*
e_pmp(rs,iIn(i),aa,ts) $ (p_alpham(rs,i,aa,ts) $ (not iMrio(i)) $ (not a(aa) or sum(sameas(aa,aIn),1))) ..
    pmp(rs,i,aa,ts) =e= m_pmpDef(rs,i,aa,ts);
```

According to the macro  $m\_pmpDef$ :



```

*
* --- the macro m_pmp substitutes out the imports' sales prices
*
*$macro m_pmpDef(r,i,aa,t) { \
*
*     ---- average import price ( pmt or pmtMrio )
*
*     m_pmt(r,i,aa,t) *(1 + mintx(r,i,aa,t) + itxshft(r,i,t)) \
*
*     --- add emission taxes for imported use
*     $$ifi %ModulesCO2_Emissions%==on          + emisp(r,t)* emiCo2eq(r,i,aa,t) \
*                                           }

```

Note that the equation is not active if the MRIO extension is switched on for that product (not iMrio(i)) as in that case, the basis to derive the agent specific is specific to group of agents and not equal to pmt.

Equally, the equation requires a non-zero share parameters for imports p\_alpham. The Melitz extension sets this share parameter to zero for the products handled in the Melitz model and accordingly removes the equation for these products. Finally, the model will for the default case substitute out the pmp prices for intermediate demand (not a(aa)).

Similar, the **agent specific prices for domestic origin** pdp reflect price in the domestic markets, defined in the equation e\_pd, equation (45) in VDM 2018. These are equal to sectoral prices ps under infinite transformation or equal to domestic sales prices pd in case of non-infinite transformation. Taxes are added as in the case imports above:

```

*
* --- Price for the domestic origin, equation (45) in VDM 2018
*
e_pdp(rs,iIn(i),aa,ts) $ (p_alphad(rs,i,aa,ts) $ (not a(aa) or sum(sameas(aa,aIn),1))) ..
pdp(rs,i,aa,ts) =e= m_pdpDef(rs,i,aa,ts);

*
* --- the macro m_pdp substitutes out the domestic sales prices: with an infinite CET for exports, use the supply
*     price ps, otherwise, use the domestic price
*
*$macro m_pdpDef(r,i,aa,t) { m_domPrice(r,i,aa,t) \
*                           *(1 + dintx(r,i,aa,t) + itxshft(r,i,t)) \
*
*     --- add emission taxes for domestic use
*     $$ifi %ModulesCO2_Emissions%==on          + emisp(r,t) * emidCo2eq(r,i,aa,t) \
*                                           }

```

Which in turn uses the macro m\_domPrice:

```

$$iftheni.mel "%melitz%"=="on"

*$macro m_domPrice(r,i,aa,t) [ ( pd(r,i,t) $ (p_omegax(r,i) ne inf) + ps(r,i,t) $ (p_omegax(r,i) eq inf)) $ (not iMel(i)) \
*                             + ( pd(r,i,t) $ p_melDomUse(r,aa,i) + m_pfirm(r,i,"domes",t) $ (not p_melDomUse(r,aa,i))) $ iMel(i)

$else.mel

*$macro m_domPrice(r,i,aa,t) ( pd(r,i,t) $ (p_omegax(r,i) ne inf) + ps(r,i,t) $ (p_omegax(r,i) eq inf))

$endif.mel

```

Note again that the Melitz extension will delete the alphad parameters for the products handled by imperfect competition to replace the equation e\_pdp as it uses a different pricing system.

The **Armington price** of the different agents pa is defined in the e\_pa, equation (47) in VDM 2018. That dual price aggregator as usually reflects the given shares for domestic p\_alphad and imported p\_alpham origin and the related prices as defined above as well as the substitution elasticity between imports and domestic origin p\_sigmam which is not agent specific:



```

*
* --- Armington price, equation (47) in VDM 2018
*
* Note WB: equation is only used if both domestic and import demand are
* present (check for share parameters), otherwise PA is substituted out with m_pa macro
*
e pa(rs,iIn(i),aa,ts) $ ( (p_alphad(rs,i,aa,ts) or p_alpham(rs,i,aa,ts)) $ (not a(aa) or sum(sameas(aa,aIn),1))
*
* WB: If additional prices are substituted out (switched on in interface)
* the Armington price aggregator for all agents aa
* is calculated in the macro m_pa and the equation is not active
*
$iftheni.subsPrices "%subsArmPrices%"=="on"
$ (1 eq 2)
$endif.subsPrices
) ..
pa(rs,i,aa,ts) =e mm_paDef(rs,i,aa,ts);

```

As defined in mm\_padeF:

```

$macro mm_padeF(r,i,aa,t) { ( [p_alphad(r,i,aa,t) * pdp(r,i,aa,t) ** (1-p_sigam(r,i))] $ p_alphad(r,i,aa,t) \
+ [p_alpham(r,i,aa,t) * (pmp(r,i,aa,t)/lambdaxm(r,i,aa,t)) ** (1-p_sigam(r,i))] $ p_alpham(r,i,aa,t) \
) ** (1/(1-p_sigam(r,i))) }

```

Note here that the equation is normally substituted out if only domestic or import demand is present. That is not the case if the standard GTAP layout is used or the substitution is explicitly switched off on the interface.

**Domestic demand** xd by the different agents is driven by the share parameter p\_alphad and times the total Armington demand defined in the macro m\_xa, times the price relation exponent the substitution elasticity, as defined in the equation e\_xd, equation (48) in VDM 2018:

```

*
* --- Domestic demand , equation (48) in VDM 2018
*
$iftheni.hsubr "%hSubr%"=="on"
e xd(rs,iIn(i),aa,ts) $ ( (p_alphad(rs,i,aa,ts) gt eps) $ (xd.range(rs,i,aa,ts) ne 0) ) ..
xd(rs,i,aa,ts)/xd.scale(rs,i,aa,ts) =e p_alphad(rs,i,aa,ts)*mm_xa(rs,i,aa,ts)/xd.scale(rs,i,aa,ts)
* (m_pa(rs,i,aa,ts)/m_pdp(rs,i,aa,ts)) ** p_sigam(rs,i);
e xn(subr,iIn(i),aa,ts) $ (p_alphan(subr,i,aa,ts) gt eps) ..
xn(subr,i,aa,ts)/xd.scale(subr,i,aa,ts) =e p_alphan(subr,i,aa,ts)*mm_xa(subr,i,aa,ts)/xd.scale(subr,i,aa,ts)
* (m_pa(subr,i,aa,ts)/sum(r_r(subr,dier),m_pdp(dier,i,aa,ts))) ** p_sigam(subr,i);
$else.hsubr
e xd(rsNat,iIn(i),aa,ts) $ ( (p_alphad(rsNat,i,aa,ts) gt eps) $ (xd.range(rsNat,i,aa,ts) ne 0) ) ..
xd(rsNat,i,aa,ts)/xd.scale(rsNat,i,aa,ts) =e p_alphad(rsNat,i,aa,ts)*mm_xa(rsNat,i,aa,ts)/xd.scale(rsNat,i,aa,ts)
* (m_pa(rsNat,i,aa,ts)/m_pdp(rsNat,i,aa,ts)) ** p_sigam(rsNat,i);
$endif.hsubr

```

Note that it is possible to define households at the level of sub-regional units (first block) where an additional demand for the national (as opposed to the regional) origin is introduced.

It is linearly aggregated over agents to **total domestic sales** xds in the equation e\_pd, equation (67) in VDM 2018, potentially taking the demand for the national origin from sub-regional households into account:

```

*
* --- Domestic goods market equilibrium, equation (67) in VDM 2018
*
e pd(rsNat,iIn(i),ts) $ xdFlag(rsNat,i) ..
xds(rsNat,i,ts)/xds.scale(rsNat,i,ts)
=e sum(aa $ (m_alphad(rsNat,i,aa,ts) gt eps), xd(rsNat,i,aa,ts))/xds.scale(rsNat,i,ts)
$iftheni.hsubr "%hSubr%"=="on"
+ sum(rs,h) $ (r_r(rs,rsNat) $ (p_alphan(rs,i,h,ts) gt eps)), xn(rs,i,h,ts))/xds.scale(rsNat,i,ts)
$endif.hsubr
;

```

**Imported demand** by each agent  $xm$  is defined accordingly in the equation  $e\_xm$ , equation (49) in VDM 2018, potentially considering the case of demand of household at sub-regional level:

```
*
* --- Import demand, equation (49) in VDM 2018
*
$$iftheni.hsubr "%hSubr%"=="on"
  e_xm(rs,iIn(i),aa,ts) $ ( (p_alpham(rs,i,aa,ts) gt eps) $ (xm.range(rs,i,aa,ts) ne 0) ) ..
  xm(rs,i,aa,ts)/xm.scale(rs,i,aa,ts) =e= p_alpham(rs,i,aa,ts)
  * mm_xa(rs,i,aa,ts)/xm.scale(rs,i,aa,ts)
  * (m_pa(rs,i,aa,ts)/sum(r_r(rs,rsNat),m_pmp(rsNat,i,aa,ts)))*p_sigam(rs,i)
  * lambdaXm(rs,i,aa,ts)**(p_sigam(rs,i)-1)
  ;
$$else.hsubr
  e_xm(rsNat,iIn(i),aa,ts) $ ( (p_alpham(rsNat,i,aa,ts) gt eps) $ (xm.range(rsNat,i,aa,ts) ne 0) ) ..
  xm(rsNat,i,aa,ts)/xm.scale(rsNat,i,aa,ts) =e= p_alpham(rsNat,i,aa,ts)
  * mm_xa(rsNat,i,aa,ts)/xm.scale(rsNat,i,aa,ts)
  * (m_pa(rsNat,i,aa,ts)/m_pmp(rsNat,i,aa,ts))*p_sigam(rsNat,i)
  * lambdaXm(rsNat,i,aa,ts)**(p_sigam(rsNat,i)-1)
  ;
$$endif.hsubr
```

**Total import demand**  $xmt$  is in the equation  $e\_xmt$  is defined as an adding up over the demand of the individual Armington agents, equation (50) in VDM 2018, potentially considering the case of demand of household at sub-regional level:

```
*
* --- Total import demand, equation (50) in VDM 2018
*
e_xmt(rsNat,iIn(i),ts) $ xmtFlag(rsNat,i) ..
xmt(rsNat,i,ts)/xmt.scale(rsNat,i,ts) =e= sum(aa $ (p_alpham(rsNat,i,aa,ts) gt eps), xm(rsNat,i,aa,ts))/xmt.scale(rsNat,i,ts)
  $$ifi "%ModulesGTAP_NUTS2%"=="ON" +sum( (subr,h) $ (r_r(subr,rsNat) $ (p_alpham(subr,i,h,ts) gt eps)),xm(subr,i,h,ts))/xmt.scale(rsNat,i,ts)
  ;
```

The bilateral **cost, insurance and freight prices**  $p_{mcif}$  are defined by the equation  $e\_p_{mcif}$  and the macro  $m\_p_{mcif}$ . As the default, these prices are substituted out from the model. The  $rrComb$  set is used in case of only one region being solved to depict the bi-lateral trade links to include.

```
*
* --- c.i.f. price, normally substituted out, macro defined above, equation (65) in VDM 2018
*
e_pmcif(rNat1,iIn(i),rNat,ts) $ (m_xwFlag(rNat1,i,rNat) $ rrComb(rNat1,rNat)) ..
p_{mcif}(rNat1,i,rNat,ts) =E= m_p_{mcif}(rNat1,i,rNat,ts);
```

The macro  $m\_p_{mcif}$  is defined as follows, equation (65) in VDM 2018:

```
*
*
* --- costs, insurance and freight price (c.i.f): add transport margins to f.o.b. price,
* equation (65) in VDM 2018, plus equation (60) substituted out
*
$$macro m_pmcif(r,i,rp,t) { [ m_pecob(r,i,rp,t)/lcu(r,t) \
  + sum(m $ p_amgm(m,r,i,rp), p_amgm(m,r,i,rp)*ptmg(m,t)\
  /m_lambdamg(m,r,i,rp,t)) * tmarg(r,i,rp,t) \
  ]*lcu(rp,t) }
```

It converts the bilateral f.o.b. (free on board) price defined in the macro  $m\_pecob$  to international currency (division by  $lcu$ ) and adds the per unit transport margin cost in international currency. These costs are defined as the transport mode  $m$  (see, air ...) specific shares  $p\_amgm$  on the given transport margin  $tmarg$ , updated with the mode specific average global price for that mode  $ptmg$ . The mode specific costs can be shifted by  $m\_lambdamg$ . The resulting costs - fob plus transport margin – in international currency and finally converted in local currency again by the multiplication by  $lcu$ .

Note that with a dense bi-lateral trade matrix, the number of variables relating to bi-lateral trade relations increases quadratic in the number of regions and linear in the number of sectors. Under full density, using 50 regions and 50 sectors implies hence  $50 \times 50 \times 50 = 125.000$  non-zero elements for

each variable defined bi-laterally. That explains why substitutions of the e.g. the f.o.b. and c.i.f. prices and the bi-lateral trade margins can dramatically reduce model size.

Bilateral **free on board prices** pefob are defined by the macro mm\_pefob, equation (64) in VDM 2018. They reflect: bilateral export taxes exptx and a product specific export tax shifter etax levied on the relevant price which depends if and how a transformation of output is used as seen below. If no CET approach is used, the supply price ps is the basis for fob calculation. In case there is only a CET between total exports and domestic sales, but none between bi-lateral export flows, the average price of exports pet is used, otherwise, bi-lateral export prices pe define the basis for fob prices.

```
*
* --- free on board price (f.o.b.), equation (64), in VDM 2018
*
*$macro mm_pefob(r,i,rp,t) { \
*
*     --- export taxes
*
*     (1 + exptx(r,i,rp,t) + etax(r,i,t)) \
*
*     --- the following might substitutes out pe depending on the CETs
*     between domestic sales and export resp. between exports
*
*     * ( \
*       --- use bilateral import prices if a CET between export destination is active
*       pe(r,i,rp,t) $ psFlag(r,i,"pe") \
*       --- use the export prices if there is only a CET between total exports and domestic use
*       + pet(r,i,t)   $ psFlag(r,i,"pet") \
*       --- otherwise, use supply price
*       + ps(r,i,t)    $ psFlag(r,i,"ps")    )}
*
}
```

Similar to the case of c.i.f. prices, the fob prices pefob are defined from that macro in the equation e\_pefob. Again, the default case is that these equations are substituted out from the model.

```
*
* --- f.o.b. price, normally substituted out, macro defined above, equation (64) in VDM 2018
*
e_pefob(rNat1,iIn(i),rNat,ts) $ (m_xwFlag(rNat1,i,rNat) $ rrComb(rNat1,rNat)) ..
    pefob(rNat1,i,rNat,ts) =E= m_pefob(rNat1,i,rNat,ts);
```

The user can define on the interface if these prices are substituted from the model.

The **average price of imports** pmt is defined in the equation e\_pmt, equation (52) in VDM 2018, and considers three cases. The first case applies for substitution elasticity different from unity and for shares not considered small. It uses the standard dual price aggregator using the bilateral demand share parameters p\_amw, the cif price defined by %pmcif% plus bilateral import taxes imptx plus a product specific import tax shifter mtax and reflects a preference shifter defined in the macro m\_lambdam. The second case uses the dual price aggregator for the CD case where the substitution elasticity is unity. The third case reflects small shares which are treated a la Leontief.

```

e_pmt(rNat,iIn(i),ts) $ ( (pmt.range(rNat,i,ts) ne 0) $ (not p_pmtElas(rNat,i)) $ xmtFlag(rNat,i) $ (not iMrio(i)) $ (not isPe(i)) ) ..
*
pmt(rNat,i,ts) =e=
*
--- CES case with p_sigmax < 1
*
[sum(rNat1 $ (xwFlag(rNat1,i,rNat)
  $ $iftheni.hasSmallShare=="on" $ (not isSmallImpShare(rNat1,i,rNat))
  ),
  p_amw(rNat1,i,rNat,ts)
  * ((1 + imptx(rNat1,i,rNat,ts) + mtax(rNat,i,ts)) * %pmcif(rNat1,i,rNat,ts) ]
  )**((1/(1-p_sigmax(rNat,i))) * p_axPmtCD(rNat,i,ts)
  ] $ ((p_sigmax(rNat,i) ne 1) $ p_axPmtCD(rNat,i,ts))
*
--- CD case with p_sigmax == 1
*
+ [prod(rNat1 $ (xwFlag(rNat1,i,rNat)
  $ $iftheni.hasSmallShare=="on" $ (not isSmallImpShare(rNat1,i,rNat))
  ),
  ((1 + imptx(rNat1,i,rNat,ts) + mtax(rNat,i,ts)) * %pmcif(rNat1,i,rNat,ts) ]
  ) / (m_lambdam(rNat1,i,rNat,ts) * p_amw(rNat1,i,rNat,ts)) ** p_amw(rNat1,i,rNat,ts)
  ) * p_axPmtCD(rNat,i,ts)
  ] $ ((p_sigmax(rNat,i) eq 1) $ p_axPmtCD(rNat,i,ts))
*
$ $iftheni.hasSmallShare "hasSmallShare=="on"
+ sum(rNat1 $ isSmallImpShare(rNat1,i,rNat), p_amw(rNat1,i,rNat,ts)
  * ((1 + imptx(rNat1,i,rNat,ts) + mtax(rNat,i,ts))
  * %pmcif(rNat1,i,rNat,ts) ] / m_lambdam(rNat1,i,rNat,ts))
$ $endif.hasSmallShare
;

```

Note that the single country case can either use the lower level Armington / CET equations of the trading partner of the country solved, or use import and export elasticities. To host that case, the `e_pmt` equation will not be introduced in the model if import prices are elasticity driven.

The allocation of total imports `xmt` to the **bi-lateral imports** `xw` is defined in the equation `e_xw`, equation (52) in VDM, and is based on the share parameters `p_amw`, the substitution between origins `p_sigmax` and the relevant price relation, i.e. the average import price `pmt` divided by the cif price `%pmcif%` plus bi-lateral import taxes `imptx` plus a potential import tax shifter `mtax`. Preference shifters as defined in the macro `m_lambdam` can be used as well.

```

*
* --- Bilateral import demand, equation (51) in VDM 2018
* Note: the price PM in VDM is always substituted out as pmCIF * (1+imptx+mtax), replaces equation (66)
*
e_xw(rNat1,iIn(i),rNat,ts) $ (xwFlag(rNat1,i,rNat) $ (not iMrio(i)) $ (not isPe(i)) $ rrComb(rNat1,rNat)) ..
*
xw(rNat1,i,rNat,ts)/xw.scale(rNat1,i,rNat,ts) =e= p_amw(rNat1,i,rNat,ts)*xmt(rNat,i,ts)/xw.scale(rNat1,i,rNat,ts)
*
* [ { (pmt(rNat,i,ts)/((1 + imptx(rNat1,i,rNat,ts) + mtax(rNat,i,ts))
*
* --- Note WB: cif price is defined via macro %pmcif%, normally not a variable in model
*
* %pmcif(rNat1,i,rNat,ts) ])**p_sigmax(rNat,i)
* m_lambdam(rNat1,i,rNat,ts)**(p_sigmax(rNat,i) - 1) }
*
$ $iftheni.hasSmallShare "hasSmallShare=="on"
$ (not isSmallImpShare(rNat1,i,rNat))
+ m_lambdam(rNat1,i,rNat,ts) $ isSmallImpShare(rNat1,i,rNat)
$ $endif.hasSmallShare
];
*

```

Note that in opposite to VDM 2018, the import price defined in equation (66) is always substituted out.

That macro `m_lambdam` is introduced to avoid that for each bi-lateral trade link in the model, a variable must be fixed to unity if no shifter is present. It thereof introduces unity in the equation – see first line below - if the shifter is not initialized, i.e. equal to zero and not fixed. The shifter variable is used if it either fixed, i.e. the range is zero, or its starting value is not zero.

```

*
* --- same, for bi-lateral Armington preferences shifters
*
$macro m_lambdam(r,i,rp,t) { [ 1 $ ( (lambdam.range(r,i,rp,t) ne 0) and (lambdam.l(r,i,rp,t) eq 0) ) \
+ lambdam(r,i,rp,t) $ ( (lambdam.range(r,i,rp,t) eq 0) or (lambdam.l(r,i,rp,t) ne 0) ) ] $ card(lambdam)\
+ 1 $ (card(lambdam) eq 0) }

```

Similar macros are used for other shifter variables as well.

As the model allows non-infinite transformation of outputs, the following equation `e_xds` (implicitly), equation (53) in VDM 2018, defines **domestic sales** `xds`. The first case is that of infinite transformation as found in the GTAP standard model where by definition the price of domestic sales `pd` is equal to the average supply price `ps`. The second case distributes total supply `xs` of a product to domestic sales based on the share parameter `p_gd` and the relation between the domestic sales price and average supply times exponent the transformation elasticity `p_omegax`:

```

-
e_xds(rsNat,iIn(i),ts) $ (xdFlag(rsNat,i) $ (p_omegax(rsNat,i) ne inf) ) ..
0 =e=
*
* --- infinite CET <=> prices are equal, GTAP standard
*
* (pd(rsNat,i,ts) - ps(rsNat,i,ts)) $ (p_omegax(rsNat,i) eq inf)
*
* --- CET mechanism otherwise
*
+ (xds(rsNat,i,ts)/xds.scale(rsNat,i,ts)
- p_gd(rsNat,i,ts)*xs(rsNat,i,ts)/xds.scale(rsNat,i,ts)
* (pd(rsNat,i,ts)/(ps(rsNat,i,ts) $ (not iACET(i)) + pds(rsNat,i,ts) $ iACET(i)))**p_omegax(rsNat,i)) $ (p_omegax(rsNat,i) ne inf);
-

```

A similar equation defines **total exports** `xet` in the equation `e_xet`, equation (54) in VDM 2018. The relevant share parameter is depicted by `p_ge` while the average price of exports is called `pet`:

```

*
* --- Total export demand, equation (54) in VDM 2018
*
e_xet(rsNat,iIn(i),ts) $ (xetFlag(rsNat,i) $ (xet.range(rsNat,i,ts) ne 0) ) ..
xet(rsNat,i,ts)/xet.scale(rsNat,i,ts)
=e=
*
* --- case with CET
*
* ((p_ge(rsNat,i,ts)*xs(rsNat,i,ts))/xet.scale(rsNat,i,ts)
* * (pet(rsNat,i,ts)/(ps(rsNat,i,ts) $ (not iACET(i)) + pds(rsNat,i,ts) $ iACET(i)))**p_omegax(rsNat,i)) $ (p_omegax(rsNat,i) ne inf)
*
* --- infinite CET: physical aggregation (global model) or elasticity driven (single country option)
*
+ ( sum( rpNat $ m_xwFlag(rsNat,i,rpNat), m_xws(rsNat,i,rpNat,ts)/xet.scale(rsNat,i,ts) ) $ (not p_petElas(rsNat,i))
+ (p_axPet(rsNat,i,ts)/xet.scale(rsNat,i,ts) * pet(rsNat,i,ts)**p_petElas(rsNat,i)) $ p_petElas(rsNat,i)
) $ (p_omegax(rsNat,i) eq inf);

```

Note the special case for the single country model where exports are driven by an export elasticity.

The average **supply price** `ps` is defined in the equation `e_xs`, equation (55) in VDM 2018. In case of infinite transformation and thus a linear aggregator – the first block – the sum of domestic sales `xds` and exports `xet` must be equal to physical output `xs`. In case of not-infinite transformation, a dual price aggregator is used based on the share parameters `p_gd` and `p_ge`, related prices `pd` and `pet` and the transformation elasticity `p_omegax`:

```

*
* --- Market clearing for domestic supply, equation (55) in VDM 2018
*
e_xs(rsNat,iIn(i),ts) $ (xsFlag(rsNat,i) $ (ps.range(rsNat,i,ts) ne 0) ) ..
0 =e=
*
* --- infinite CET <=> physical balancing
*
+ (xs(rsNat,i,ts)/xs.scale(rsNat,i,ts) - (xds(rsNat,i,ts)/xs.scale(rsNat,i,ts)) $ xdFlag(rsNat,i)
- (xet(rsNat,i,ts)/xs.scale(rsNat,i,ts)) $ xetFlag(rsNat,i) ) $ (p_omegax(rsNat,i) eq inf)
*
* --- dual price aggregator
*
+ ( ps(rsNat,i,ts) $ (not iACET(i))
+ pds(rsNat,i,ts) $ iACET(i)
- ( p_gd(rsNat,i,ts)*pd(rsNat,i,ts)**(1 $ (not iACET(i)) + p_omegax(rsNat,i))
+ p_ge(rsNat,i,ts)*pet(rsNat,i,ts)**(1 $ (not iACET(i)) + p_omegax(rsNat,i))
)**(1/(1 $ (not iACET(i)) + p_omegax(rsNat,i))) ) ) $ (p_omegax(rsNat,i) ne inf)
;

```

**Bilateral export supply** is by definition equal to bilateral export demand `xw`, that equality is used to define indirectly the bilateral export price `pe` in case of non-infinite transformation in the first block, as defined in the equation `e_pe`, equation (56) in VDM 2018. Otherwise, bilateral export prices `pe` and the average export prices `pet` are by definition equal:



```

*
* --- This equation substitutes out the equilibrium condition for bilateral trade, equation (56) in VDM 2018
* (makes it possible to drop equation (68) in VDM 2018)
*
e_pe(rNat,iIn(i),rNat1,ts) $ (m_xwFlag(rNat,i,rNat1) $ psFlag(rNat,i,"pe") $ rrComb(rNat,rNat1)) ..
0 =e= ( m_xws(rNat,i,rNat1,ts)/xw.scale(rNat,i,rNat1,ts)
      - (p_gw(rNat,i,rNat1,ts)*xet(rNat,i,ts))/xw.scale(rNat,i,rNat1,ts)
      * [ 1
        + { (pe(rNat,i,rNat1,ts)/(pdet(rNat,i,ts) $ iACET(i) + pet(rNat,i,ts) $ (not iACET(i))) ) **p_omegaw(rNat,i)
          } $ (not isSmallExpShare(rNat,i,rNat1)) ]
      ) $ (p_omegaw(rNat,i) ne inf)
*
* --- case with law of one price (infinite transformation)
*
+ ( pe(rNat,i,rNat1,ts) - (pet(rNat,i,ts) $ (not iACET(i)) + pdet(rNat,i,ts) $ iACET(i)) $ (p_omegax(rNat,i) ne inf)
  - ps(rNat,i,ts) $ (p_omegax(rNat,i) eq inf) )
  $ (p_omegaw(rNat,i) eq inf) ;

```

The set iACET depicts cases where a volume preserving CET is used.

The **aggregate price of export** pet defined in the equation e\_pet, equation (57) in VDM 2018, is either defined from a dual price aggregator in case of non-infinite transformation or equal to the supply price ps in case of infinite transformation. Note the inclusion of the special case of small export shares handled via Leontief:

```

*
* --- Aggregate price for exports, equation (57) in VDM 2018
*
e_pet(rNat,iIn(i),ts) $ (xetFlag(rNat,i) $ sum(rsNat,rrComb(rNat,rsNat)) $ (pet.range(rNat,i,ts) ne 0)
  $ (not psFlag(rNat,i,"ps"))) ..

0 =e= (
  pdet(rNat,i,ts) $ iACET(i)
  + pet(rNat,i,ts) $ (not iACET(i))
  - p_axPet(rNat,i,ts)
    * sum(rNat1 $ (m_xwFlag(rNat,i,rNat1) $ (not isSmallExpShare(rNat,i,rNat1))),
      p_gw(rNat,i,rNat1,ts) * pe(rNat,i,rNat1,ts) ** (1 $ (not iACET(i)) + p_omegaw(rNat,i))
      ) ** (1/(1 $ (not iACET(i)) + p_omegaw(rNat,i)))
    )
  $$ifthen1.smallShare "%hasSmallShare%"=="on"
*
* --- Leontief treatment of small export shares
*
  + sum(rNat1 $ (m_xwFlag(rNat,i,rNat1) $ isSmallExpShare(rNat,i,rNat1)),
    p_gw(rNat,i,rNat1,ts) * pe(rNat,i,rNat1,ts)) $ card(isSmallExpShare)
  $$endif.smallShare
  ) $ psFlag(rNat,i,"pe")

+ (xet(rNat,i,ts)/xet.scale(rNat,i,ts)
  - sum( rpNat $ m_xwFlag(rNat,i,rpNat), m_xws(rNat,i,rpNat,ts)/xet.scale(rNat,i,ts) ) $ (not p_petElas(rNat,i))
  + (p_axPet(rNat,i,ts)/xet.scale(rNat,i,ts) * pet(rNat,i,ts) ** p_petElas(rNat,i)) $ p_petElas(rNat,i)
  ) $ psFlag(rNat,i,"pet");

```

The **global demand for transport services** xtmg of mode m is based on a Leontief approach and defined in the equation e\_xtmg, summarizing equations (58), (59) and (61) in VDM 2018. The given bi-lateral transport margin demand tmarg are distributed to the different transport modes m based on the share parameter p\_amgm multiplied with the bilateral transport flows defined in the macro m\_xws, reflecting a potential demand shifter m\_lambdamg. Note that substitution is between regions providing shares on international transport by transport mode, and not between different modes:

```

*
* --- the following equation summarizes equations (58), (59), (61) in VDM 2018
* and thus save a considerable number of variables and equations in the overall model
*
e_xtmg(m,ts) $ (xtmg.range(m,ts) ne 0) ..

xtmg(m,ts)/xtmg.scale(m,ts)
=e= sum((rNat,i,rNat1) $ p_amgm(m,rNat,i,rNat1),

*
*          ---- equation (58) - bilateral import flows
*          times per unit demand for transport services
p_amgm(m,rNat,i,rNat1)
* ( m_xws(rNat,i,rNat1,ts) $ rrComb(rNat,rNat1)
  $sonexpand
  + m_xwsl(rNat,i,rNat1,ts) $ (not rrComb(rNat,rNat1))
  $soffexpand
)

*
*          --- equation (59) multiplied with trade margins
*
* tmarg(rNat,i,rNat1,ts)
/m_lambdamg(m,rNat,i,rNat1,ts)
)/xtmg.scale(m,ts);

```

The **region specific demand for each transport mode**  $x_a$ , defined in equation  $e\_xatmg$ , equation (62) in VDM 2018, is based on a CES demand system which reflects the average global price for each transport mode  $ptmg$  and the regional specific price defined in the macro  $m\_pa$  and the substitution elasticity  $p\_sigmamg$ :

```

*
* --- Armington demand for transport services by mode, distributes global demand
* xtmg to region, equation (62) in VDM 2018
*
e_xatmg(rsNat,m,tmg,ts) $ (xaFlag(rsNat,m,tmg) ) ..

xa(rsNat,m,tmg,ts)/xa.scale(rsNat,m,tmg,ts)
=e= alphaa(rsNat,m,tmg,ts)*xtmg(m,ts)/xa.scale(rsNat,m,tmg,ts)
* ( ptmg(m,ts)/(m_pa(rsNat,m,tmg,ts)/lcu(rsNat,ts)))**p_sigmamg(m) ;

```

The global average price for each transport mode  $ptmg$  is defined in the equation  $e\_ptm$ , equation (63) in VDM 2018, via a dual price aggregator which distinguishes the CD and CES/Leontief case:

```

*
* --- Average global price of transport mode m, equation (63) in VDM 2018
*
e_ptmg(m,ts) $ (ptmg.range(m,ts) ne 0) ..

ptmg(m,ts) =e=

*
* --- case with elasticity <> 1 (CES)
*
( sum( (rNat,tmg) $ alphaa.l(rNat,m,tmg,ts),
  alphaa(rNat,m,tmg,ts)
  * ( m_pa(rNat,m,tmg,ts)/lcu(rNat,ts)) $ rsNat(rNat)
  + (pa.l(rNat,m,tmg,ts)/lcu.l(rNat,ts)) $ (not rsNat(rNat)))** (1-p_sigmamg(m)) )** (1/(1-p_sigmamg(m))) )/p_axmg(m,ts)
) $ (p_sigmamg(m) ne 1)

*
* --- Cobb-double case with elasticity == 1
*
+ ( sum(tmg, prod(rNat $ alphaa.l(rNat,m,tmg,ts),
  ( (m_pa(rNat,m,tmg,ts)/lcu(rNat,ts)) $ rsNat(rNat)
  + (pa.l(rNat,m,tmg,ts)/lcu.l(rNat,ts)) $ (not rsNat(rNat)))
  /alphaa(rNat,m,tmg,ts)** (alphaa(rNat,m,tmg,ts))) )/p_axmg(m,ts)
) $ (p_sigmamg(m) eq 1)
;

```

### 2.13.3 Melitz model

The methodological details of the Melitz extension are described in a separate document. In here, only the equations are briefly discussed (see GtapMelitz\GtapMelitz\_model.gms).

The Armington price of each agent for a product  $iMel$  included in the Melitz model as shown in equation  $e\_pamel$  below uses the usual CES dual price aggregator, however, the share parameters

amel are updated which changes of the number of firms operating on the trade link to yield a different type of price aggregator. The number of firms is defined via the macro `m_nFirmOp`, while the numbers of firms operating under the benchmark is stored in the parameter `p_nFormsOp0`:

```
e_pamel(rsNat(rpNat),iMel(i),aa,ts(t)) $ ((m_alpham(rpNat,i,aa,t) or m_alphad(rpNat,i,aa,t))
$ (sum(sameas(aIn,aa),1) or (not a(aa)) )) ..
*
* rpNat is the import destination (= Armington agents)
* the r -> rr are the exports, "domes" are domestic sales
*
pa(rpNat,i,aa,t) =E=
[sum(rNat1 $ amel(rNat1,iMel,aa,rpNat,t),
amel(rNat1,i,aa,rpNat,t)*m_nFirmsOp(rNat1,i,rpNat,t)/p_nFirmsOp0(rNat1,i,rpNat,t)
*m_pAgentm(rNat1,i,aa,rpNat,t)**(1-p_sigmaMel(rpNat,i,rpNat)))
+[amel(rpNat,i,aa,"domes")*( (m_nFirmsOp(rpNat,i,"domes",t)/p_nFirmsOp0(rpNat,i,"domes",t)) $ (not mFlag(rpNat,aa,i))
+ 1 $ mFlag(rpNat,aa,i))
*m_pAgentd(rpNat,i,aa,t)**(1-p_sigmaMel(rpNat,i,rpNat)))] $ amel(rpNat,i,aa,"domes",t)
]**(1/(1-p_sigmaMel(rpNat,i,rpNat)))
```

The Melitz extensions allows to treat domestic sales of intermediates to same industry, i.e. the diagonal I/O element, different from the other agents by removing the love of variety effect and thus also the price markup. That has shown to help in many cases where the solver ran into infeasibilities. If that mechanism is active for a specific region, agent and product is depicted by the `mFlag`. In the equation above, in that case, a constant of unity ( $+ 1 \ \$ \ mFlag(rpNat,aa,i)$ ) is introduced in the equation instead of updating the share parameters by the number of farms of operating.

The relevant prices for the agents are depicted in the two macros `m_pAgentm` for bi-lateral import links and `m_pAgentd` for the domestic sales case.

The bi-lateral import case uses the c.i.f. price plus import and emission taxes:

```
$$macro m_pAgentm(r,i,aa,rrl,t) \
[ %pmcif$(r,i,rrl,t)*(1 + imptx(r,i,rrl,t) + mtax(rrl,i,t)) \
* (1 + mintx(rrl,i,aa,t) + itxshft(rrl,aa,t) $ iTaxShift(i)) \
$ $ifi %ModulesCO2_Emissions%=-on + emis(r,rrl,t) * (emi0(rrl,i,aa)/xma0(rrl,i,aa)) $ xma0(rrl,i,aa) \
]
```

Note that the basis for the f.o.b. price from which the c.i.f. price is derived are the prices charged by the average firm on the trade link defined by the macro `m_pFirm` for Melitz products `iMel(i)` as shown in the macro below:

```
$$macro m_pegob(r,i,rp,t) [ m_pFirm(r,i,rp,t) * (1 + exptx(r,i,rp,t) + etax(r,i,t)) $ iMel(i) \
+ mm_pegob(r,i,rp,t) $ (not iMel(i))]

$$macro m_pFirm(r,i,rp,t) pFirm(r,i,rp,t)
```

Total import demand `xm` as defined in the equation `e_xmMel` for each Armington agent `aa` is the sum over the agent specific bi-lateral Armington demands, which are driven by relation of average Armington price `pa` as defined in the `e_pamel` equation above and the bi-lateral price defined in the macro `m_pAgentm`, exponent the substitution elasticity `p_sigmaMel`. Note again that the share parameter is not fix, but the product of the share parameter at the benchmark `amel` and the change in the number of firms operating on that trade link:

```
e_xmMel(rsNat(rpNat),iMel(i),aa,ts(t)) $ (m_alpham(rpNat,i,aa,t)$ (sum(sameas(aIn,aa),1) or (not a(aa)) )) ..
xm(rpNat,i,aa,t)/xm.scale(rpNat,i,aa,t)
=E= sum(rNat1 $ amel(rNat1,i,aa,rpNat,t), amel(rNat1,i,aa,rpNat,t)
* m_nFirmsOp(rNat1,i,rpNat,t)/p_nFirmsOp0(rNat1,i,rpNat,t)*mm_xa(rpNat,i,aa,t)
* (pa(rpNat,i,aa,t)/m_pAgentm(rNat1,i,aa,rpNat,t))**p_sigmaMel(rpNat,i,rpNat))/xm.scale(rpNat,i,aa,t);
```

The domestic demand is defined symmetrically in the `e_xdMel` equation:



```
e_xdMel(rsNat(rpNat),iMel(i),aa,ts(t)) $ (m_alphad(rpNat,i,aa,t) $ (not mFlag(rpNat,aa,i))
$ (sum(sameas(aIn,aa),1) or (not a(aa)) )) ..

xd(rpNat,i,aa,t)/xd.scale(rpNat,i,aa,t) =E= amel(rpNat,i,aa,"domes",t)
*m_NFirmsOp(rpNat,i,"domes",t)/p_nFirmsOp0(rpNat,i,"domes",t)*mm_xa(rpNat,i,aa,t)
*(pa(rpNat,i,aa,t)/m_pAgentd(rpNat,i,aa,t))*p_sigmaMel(rpNat,i,rpNat)/xd.scale(rpNat,i,aa,t);
```

As mentioned already above, the Melitz extensions allows to treat domestic sales of intermediates to same industry, i.e. the diagonal I/O element, different from the other agents by removing the love of variety effect and thus also the price markup. That has shown to help in many cases where the solver ran into infeasibilities. In that case, the following simpler equation e\_xddMel defines the domestic demand by the same industry for its own output:

```
e_xddMel(rsNat(rpNat),iMel(i),aa,ts(t)) $ (m_alphad(rpNat,i,aa,t) $ mFlag(rpNat,aa,i)
$ (sum(sameas(aIn,aa),1) or (not a(aa)) )) ..

xd(rpNat,i,aa,t)/xd.scale(rpNat,i,aa,t) =E= amel(rpNat,i,aa,"domes",t)*mm_xa(rpNat,i,aa,t)
*(pa(rpNat,i,aa,t)/m_pAgentd(rpNat,i,aa,t))*p_sigmaMel(rpNat,i,rpNat)/xd.scale(rpNat,i,aa,t);
```

The total demand for domestic produce xds subject to price markups is defined by adding up over the domestic demands in the e\_xdMelf equation, however, without the diagonal intermediate demand case if the mFlag is active:

```
e_xdMelf(rsNat(rpNat),iMel(i),"domes",ts(t)) $ (sum(aa $ (not mFlag(rpNat,aa,i)), amel(rpNat,i,aa,"domes",t))) ..
*
xds(rpNat,i,t)/xds.scale(rpNat,i,t)
=E= sum(aa $ (amel(rpNat,i,aa,"domes",t) $ (not mFlag(rpNat,aa,i))), xd(rpNat,i,aa,t)/xds.scale(rpNat,i,t);
```

The total bi-lateral demand is defined accordingly as an adding up over the agents' bi-lateral demands in the equation e\_xwmelf. In the Melitz case, it defines the average output of the firms being active on that link. Alternatively, the equation structure can be used to use one Armington nest, only, which allows quantifying the impact on model solution compared to the usual two-stage Armington system. In that case, the equation drives the bi-lateral demands xw.

```
e_xwMelf(rpNat,iMel(i),rNat1,ts(t)) $ (xwFlagMel(rpNat,i,rNat1) $ rrComb(rpNat,rNat1)) ..
:
:   -- export demand, including exports to the same block
:
:   $$iftheni.dem not "%HETSetup%"=="Only aggregated demand (no heterogenous firms)"
:   qAvFirm(rpNat,i,rNat1,t)/xw.scale(rpNat,i,rNat1,t)
:   $$else.dem
:   xw(rpNat,i,rNat1,t)/xw.scale(rpNat,i,rNat1,t)
:   $$endif.dem
:
:   =E=
:
:   --- armington duals: NMRelibute total quantity according to price relations
:   (prices as in dual price aggregatorabove)
:
:   sum(aa $ amel(rpNat,i,aa,rNat1,t), amel(rpNat,i,aa,rNat1,t)/p_nFirmsOp0(rpNat,i,rNat1,t)*mm_xa(rNat1,i,aa,t)
:   *(pa(rNat1,i,aa,t)/m_pAgentm(rpNat,i,aa,rNat1,t))*p_sigmaMel(rpNat,i,rNat1)/xw.scale(rpNat,i,rNat1,t);
```

The fix cost price is equal to the price of the fix cost nest “fCost” if a separate fix cost nest is active. Otherwise, the fix cost price is defined in the e\_FCostP equation as equal to unit costs of production plus production taxes:

```
e_FcostP(rs(r),"fCost",a,ts(t)) $ (sum(iMel $ xFlag(r,a,iMel),p_MFirmsEnt0(r,iMel,t))
$ $iftheni "%fcost_nest%" == "on"
$ (1 eq 2)
$endif
) ..
pTNest(r,"fCost",a,t) =E= mm_px(r,a,t)*(1 + prdtx(r,a,t));
```

The price markup on each trade link is indirectly defined in e\_MKUP equation by the endogenous average firm productivity phiFirm and the price charged by the average firm pFirm, reflecting changes in supply price ps:

```
e_MKUP(rpNat,iMel(i),rrd,ts(t)) $ ((xwFlagMel(rpNat,i,rrd) $ (sum(sameas(rNat1,rrd),rrComb(rpNat,rNat1)) or rsNat(rpNat)))
$ (pFirm.range(rpNat,i,rrd,t) ne 0) ) ..

p_tau(rpNat,i,rrd,t)*ps(rpNat,i,t)*1/(1 - 1/p_sigmaMel(rpNat,i,rrd))
=E= phiFirm(rpNat,i,rrd,t)*pFirm(rpNat,i,rrd,t);
```

The endogenous number of firms entering the industry mFirmsEnt is defined as follows:

```
e_firmsEnt(rsNat(rNat),iMel(i),ts(t)) $ (p_mFirmsEnt0(rNat,iMel,t) ) ..

p_delt_fs(rNat,i,t)*MFirmsEnt(rNat,i,t)*sum(a $ xFlag(rNat,a,i), pTnest(rNat,"fCost",a,t))/p_mFirmsEnt0(rNat,i,t)

--- nFirmsOp * qAv is replaced by the resulting quantities xw and xds to simplify the equation

=E= sum(rrd $ xwFlagMel(rNat,i,rrd),
(sum(sameas(rp,rrd), xw(rNat,i,rrd,t)) + xds(rNat,i,t) $ sameas(rrd,"domes")))
$Iftheni "%HETSetup%"=="Melitz"
* pFirm(rNat,i,rrd,t) * (p_sigmaMel(rNat,i,rrd)-1)/(p_a*p_sigmaMel(rNat,i,rrd))
$Else
* (pFirm(rNat,i,rrd,t)-p_tau(rNat,iMel,rrd,t)*ps(rNat,i,t))
$Endif

)/p_mFirmsEnt0(rNat,i,t);
```

Note the difference between the Melitz set-up where fix costs and mark-ups are trade link specific and the Krugman model where only industry wide fix costs are present.

In the Krugman model, only the total industry size MFirmsEnt changes and updates also the number of firms being active on each trade link nFirmsOp as defined by e\_krug equation:

```
e_krug(rsNat(rNat),iMel(i),rrd,ts(t)) $ xwFlagMel(rNat,i,rrd) ..

nFirmsOp(rNat,i,rrd,t)/p_nFirmsOp0(rNat,i,rrd,t) =E= MFirmsEnt(rNat,i,t)/p_mFirmsEnt0(rNat,i,t);
```

The number of firms operating on a trade link in the Melitz case is defined by the zero-cutoff profit equation e\_ZCP. The LHS defines the fixed cost on that trade link, i.e. the given total benchmark fix cost p\_fc updated with changes in the fix cost price ptNest("fCost"). The RHS ensures that these fix costs are exhausted by the number of firms being active, reflecting the productivity distribution:

```
*
* --- zero cutoff profit, determines number of firms operating nFirmsOP on link r->rrd
*
e_ZCP(rNat,iMel(i),rrd,ts(t)) $ (xwFlagMel(rNat,i,rrd) $ (nFirmsOp.range(rNat,i,rrd,t))
$ (sum(sameas(rNat1,rrd),rrComb(rNat,rNat1)) or rsNat(rNat)) ) ..
*
* --- fixed costs (= unit costs times fix cost share)
*
sum(a $ xFlag(rNat,a,i), pTnest(rNat,"fCost",a,t))*p_fc(rNat,i,rrd,t)/QAvFirm.scale(rNat,iMel,rrd,t)
=E= QAvFirm(rNat,i,rrd,t)*pFirm(rNat,i,rrd,t)/QAvFirm.scale(rNat,iMel,rrd,t)*(p_a+1-p_sigmaMel(rNat,i,rrd))
/ (p_a*p_sigmaMel(rNat,i,rrd));
```

The pareto productivity phiFirm which enters the price markup equation is defined in the equation e\_PAR.

```
e_PAR(rNat,iMel(i),rrd,ts(t)) $ (xwFlagMel(rNat,i,rrd) $ (sum(sameas(rNat1,rrd),rrComb(rNat,rNat1)) or rsNat(rNat))) ..
phiFirm(rNat,i,rrd,t)/p_phiFirm0(rNat,iMel,rrd,t)
=E= p_b/p_phiFirm0(rNat,iMel,rrd,t) * (p_a/(p_a+1-p_sigmaMel(rNat,i,rrd)))*(1/(p_sigmaMel(rNat,i,rrd)-1))
* (nFirmsOp(rNat,i,rrd,t)/MFirmsEnt(rNat,i,t))**(-1/p_a);
```

Variable cost exhaustion is ensured by the e\_MKT equation:

```

e_MKT(rsNat(rNat),iMel(i),ts(t)) $ p_MFirmsEnt0(rNat,iMel,t) ..
sum(a $ xFlag(rNat,a,i), m_xp(rNat,a,t)/xp.scale(rNat,a,t))
=E=
*
* icebeg cost of trade p_tau times output per firm QAvFirm times # of operating firm,
* corrected for average productivity
*
* { sum(rrd $ xwFlagMel(rNat,i,rrd),
*   [sum(sameas(rrl,rrd), xw(rNat,i,rrl,t))+xds(rNat,i,t) $ sameas(rrd,"domes")]
*   *p_tau(rNat,i,rrd,t)
*   $Sifi "%HETSetup%""Melitz" /phiFirm(rNat,i,rrd,t)
*   )
*   -- costs of intermediate input use by same industry
*   + sum(aa $ (mFlag(rNat,aa,i) $ m_alphad(rNat,i,aa,t)), xd(rNat,i,aa,t))
*   $Sifi "%fcost_nest%" == "off" + sum(a $ xFlag(rNat,a,i), xtNest(rNat,"fCost",a,t)*(1+prdtx(rNat,a,t)))
*   )
*   /sum(a $ xFlag(rNat,a,i),xp.scale(rNat,a,t));

```

Note that the total output comprises fix cost input demand if no separate fix costs nest is active. Therefore, these costs are added for that case, see the line starting with \$Sifi.

Fix costs are defined as follows in the e\_fCost equation. They consist of industry wide fix costs, i.e. the endogenous number of firms in the industry MFirmEnt times the given fix costs per firm p\_delt\_fs, plus the given fix cost on each trade link p\_fc times endogenous number of the firms operating on the link defined by the m\_NFirmsOp macro:

```

e_fCost(rsNat(rNat),"fCost",a,ts(t)) $ (sum(iMel $ xFlag(rNat,a,iMel),p_MFirmsEnt0(rNat,iMel,t)) $ aIn(a)) ..
xtNest(rNat,"fCost",a,t)/xtNest.scale(rNat,"fCost",a,t) * (1+prdtx(rNat,a,t)) =E=
t
t --- fix costs times # of existing firms
t
t sum(i $ xFlag(rNat,a,i), p_delt_fs(rNat,i,t)*MFirmsEnt(rNat,i,t) +
t
t --- # of firms operating on the different markets times
t
t (1) bilateral fix cost p_fc
t (2) icebeg cost of trade p_tau times output per firm QAvFirm, corrected for average productivity
t
t sum(rrd $ xwFlagMel(rNat,i,rrd), m_NFirmsOp(rNat,i,rrd,t)*p_fc(rNat,i,rrd,t))
t )
t /xtNest.scale(rNat,"fCost",a,t);

```

There is a block of equations which defines variables used elsewhere in the model, but not defined by the equation in the Melitz model:

```

*
* --- definition of total exports from average output per firm and # of firms
* operating on the trade link rp->r
*
e_xwMel(rNat1,iMel(i),rNat,ts(t)) $ ( xwFlagMel(rNat1,i,rNat) $ rrComb(rNat1,rNat)) ..
xw(rNat1,i,rNat,t)/xw.scale(rNat1,i,rNat,t) =E= m_xw(rNat1,i,rNat,t)/xw.scale(rNat1,i,rNat,t);
e_xdsMel(rsNat(rNat1),iMel(i),ts(t)) $ xwFlagMel(rNat1,i,"domes") ..
xds(rNat1,i,t)/xds.scale(rNat1,i,t) =E= m_xds(rNat1,i,t)/xds.scale(rNat1,i,t);
*
* --- total export demand, aggregated from bi-lateral export quantities
*
e_xetMel(rsNat(rNat),iMel(i),ts(t)) $ sum(rNat1, xwFlagMel(rNat,i,rNat1)) ..
xet(rNat,i,t)/xet.scale(rNat,i,t) =E= sum(rNat1 $ xwFlagMel(rNat,i,rNat1), xw(rNat,i,rNat1,t))/xet.scale(rNat,i,t);
*
* --- average domestic and import prices for melitz products
* (replace Armington second nest price index in standard GTAP model)
*
e_pdMel(rsNat(rNat),iMel(i),ts(t)) $ sum(aa $ (not mFlag(rNat,aa,i)), amel(rNat,i,aa,"domes",t)) ..
pd(rNat,i,t) =E= m_pFirm(rNat,i,"domes",t);
*
* --- import aggregate
*
e_xmtMel(rsNat(rNat),iMel(i),ts(t)) $ sum(aa, m_alpham(rNat,i,aa,t)) ..
xmt(rNat,i,t)/xmt.scale(rNat,i,t) =E= sum(aa $ m_alpham(rNat,i,aa,t), xm(rNat,i,aa,t))/xmt.scale(rNat,i,t);

```

Note that the average price of imports is defined from the total value wmt in the e\_pmtMel equation:

```

e_pmtMel(rsNat(rNat),iMel(i),ts(t)) $ sum(rNat1,xwFlagMel(rNat1,i,rNat)) ..
*
* --- average import prices times total import quantities (Armington aggretege, not physically)
* (not that PMT in Standard GTAP model is unity, but not in here)
*
wmt(rNat,i,t)/xmt.scale(rNat,i,t)
*
=E=
*
* --- value of total imports, defined from value of each import flow
* (xm comprises the transport margins and the original border protection,
*
*
sum(rNat1 $ xwFlagMel(rNat1,i,rNat), xw(rNat1,i,rNat,t)*%pmCif%(rNat1,i,rNat,t)
    *(1 + imptx(rNat1,i,rNat,t) + mtax(rNat,i,t))
)/xmt.scale(rNat,i,t);

```

And the following equation e\_wmt:

```

e_wmt(rsNat(rNat),iMel(i),ts(t)) $ sum(rNat1,xwFlagMel(rNat1,i,rNat)) ..
wmt(rNat,i,t)/xmt.scale(rNat,i,t) =E= pmt(rNat,i,t)*(xmt(rNat,i,t)/xmt.scale(rNat,i,t));

```

In order to check for the correct functioning of model, a revenue check equation e\_revCheck is introduced:

```

e_revCheck(rsNat(rNat),iMel(i),ts(t)) $ p_MFirmsEnt0(rNat,iMel,t) ..
revCheck(rNat,i,t)
/sum(a $ xFlag(rNat,a,i),xp.scale(rNat,a,t))
=E=
[sum(rrd $ xwFlagMel(rNat,i,rrd),
    (sum(sameas(rNat1,rrd), xw(rNat,i,rNat1,t)) + xds(rNat,i,t) $ sameas(rrd,"domes"))*pFirm(rNat,i,rrd,t))
    -sum(a $ xFlag(rNat,a,i), m_xp(rNat,a,t)*ps(rNat,i,t))
    +sum(aa $ mFlag(rNat,aa,i), xd(rNat,i,aa,t))*ps(rNat,i,t)
    -sum(a $ xFlag(rNat,a,i), xtNest(rNat,"fCost",a,t)*pTNest(rNat,"fCost",a,t) * (1+prdtx(rNat,a,t)))
    ]
/sum(a $ xFlag(rNat,a,i),xp.scale(rNat,a,t));

```

The variable revCheck on the LHS should yield a zero in the benchmark and under a simulation. It adds what the agents pay for domestic and import purchases while paying the prices charged by the average firm, minus variable production costs, i.e. production output  $m\_xp$  times the supply price  $ps$ , adds remuneration for own intermediate consumption if the  $mFlag$  is present, and finally subtracts fix cost.

## 2.14 Price indices

The model defines different price indices which can be used as regional (or global) numeraires and/or for reporting purposes.

Average factor prices pft and total stock xft for non-non-mobile factors, are defined in the e\_pftFnm:

```

*
* --- calculate average price of non-mobile factors
*
E= pftFnm(rs,fnm,ts) $ ( (xftEqFlag(rs,fnm) or dis(rs)) $ sum(aIn $ xFlag(rs,fnm,aIn), xf.scale(rs,fnm,aIn,ts)) ) ..
pft(rs,fnm,ts)*xft(rs,fnm,ts)/sum(a $ xFlag(rs,fnm,a), xf.scale(rs,fnm,a,ts))
=E= vft(rs,fnm,ts)/sum(a $ xFlag(rs,fnm,a), xf.scale(rs,fnm,a,ts));
e_xftFnm(rs,fnm,ts) $ (xftEqFlag(rs,fnm) and (subr(rs) or (not sameas(fnm,"newCap")))) $ sum(aIn $ xFlag(rs,fnm,aIn), xf.scale(rs,fnm,aIn,ts)) ..
xft(rs,fnm,ts)/sum(a $ xFlag(rs,fnm,a), xf.scale(rs,fnm,a,ts))
=E= sum(a $ xFlag(rs,fnm,a), xft(rs,fnm,a,ts))/sum(a $ xFlag(rs,fnm,a), xf.scale(rs,fnm,a,ts));
e_vftFnm(rs,fnm,ts) $ ((xftEqFlag(rs,fnm) or dis(rs)) $ sum(aIn $ xFlag(rs,fnm,aIn), xf.scale(rs,fnm,aIn,ts)) ) ..
vft(rs,fnm,ts)/sum(a $ xFlag(rs,fnm,a), xf.scale(rs,fnm,a,ts))
=E= sum(a $ xFlag(rs,fnm,a), xft(rs,fnm,a,ts)*pf(rs,fnm,a,ts))/sum(a $ xFlag(rs,fnm,a), xf.scale(rs,fnm,a,ts));

```

**Regional factor price indices** pfact are defined in the equation e\_pfact based on factor prices pft and weights phif and are used to define them in the benchmark, whereas in shock or follow up years, equations (93) and (94) in VDM 2018:

```

*
* --- Regional index of factor prices, equations (93) and (94) in VDM 2018
*
$macro m_qfactr(tp,tq) sum (f $ xft.l(rNat,f,tp), \
    ( pft(rNat,f,tp)/lcu(rNat,tp)) $ (ts(tp) and rsNat(rNat)) \
    + (pft.l(rNat,f,tp)/lcu.l(rNat,tp)) $ ( (not ts(tp)) or (not rsNat(rNat))) ) \
    * ( xft(rNat,f,tq) $ (ts(tq) and rsNat(rNat)) \
    + xft.l(rNat,f,tq) $ ( (not ts(tq)) or (not rsNat(rNat))) ) )

*
e_pfact(rsNat(rNat),ts(tRun)) $ ( (pfact.range(rNat,tRun) ne 0) or (not rres(rNat)) or (lcu.range(rNat,tRun) ne 0) ) ..
pfact(rNat,tRun) =e= sum(f, p_phif(rNat,f,tRun)*pft(rNat,f,tRun)) $ sameas(tRun,"%t0%")
+ [ pfact.l(rNat,tRun-1) * sqrt( m_qfactr(tRun,tRun-1)/m_qfactr(tRun-1,tRun-1)
    * m_qfactr(tRun,tRun) /m_qfactr(tRun-1,tRun) ) ] $ (not sameas(tRun,"%t0%"));

```

The **average world price of factors** pwfact as defined in the equation e\_pwfact uses weights phifw and reflects the exchange rates lcu to aggregate the regional factor prices pft. It is global numeraire price in the model, equations (95) and (96) in VDM 2018.

```

*
* --- average world price of factors, equation (95) and (96) in VDM 2018
* fixed to numeraire with the following equation
*
$macro m_qfactw(tp,tq) sum ( rNat, m_qfactr(tp,tq))
e_pwfact(ts(tRun)) $ (pwfact.range(tRun) ne 0) ..

pwfact(tRun) =e= sum((rNat,f), p_phifw(rNat,f,tRun)*( pft(rNat,f,tRun) $ rsNat(rNat)
    + pft.l(rNat,f,tRun) $ (not rsNat(rNat)) )/lcu(rNat,tRun)) $ sameas(tRun,"%t0%")

*
+ [ pwfact.l(tRun-1) * sqrt( m_qfactw(tRun,tRun-1)/m_qfactw(tRun-1,tRun-1)
    * m_qfactw(tRun,tRun) /m_qfactw(tRun-1,tRun) ) ] $ (not sameas(tRun,"%t0%"));

```

**Regional producer price indices** pprod as defined in the equation e\_pprod are based on weights p\_phii:

```

*
* --- Regional producer price index,
*
e_pprod(rsNat(rNat),ts(t)) $ ( ((pprod.range(rNat,t) ne 0) or (not rres(rNat)) or (lcu.range(rNat,t) ne 0) ) ) ..
pprod(rNat,t) =e= sum(i, p_phii(rNat,i,t)*ps(rNat,i,t)) ;

```

**Average domestic consumption prices** pabs are an average of the Armington prices for the different types of final demand fd (final demand prices for households, government, investment and domestic supply of trade margins) and weights p\_phia. That is an approximate version of equation (90) in VDM 2018. That price index is not used elsewhere in the model.

```

*
* --- price index of domestic consumption, equation (90) in VDM 2018 (only approximate, not used in model equations)
*
e_pabs(rsNat,ts) ..
pabs(rsNat,ts) =e= sum((i,fd) $ p_phia(rsNat,i,fd,ts), p_phia(rsNat,i,fd,ts)*m_pa(rsNat,i,fd,ts)) ;

```

## 2.15 List of main prices in model

Table 3: Prices in model

Variable	Content	Indices
px	Unit costs of production	r,a,t
pp	Producer price	r,a,t
pva	Price of value added composite	r,a,t
pnd	Price of intermediate bundle	r,a,t
pf	Activity specific factor price, tax exclusive	r,f,a,t
pfa	Activity specific factor price, tax inclusive	r,f,a,t
ptnest	Price of technology nest	r,tNest,a,t
pft	Aggregate price of factors	r,f,t

Variable	Content	Indices
p	Price of output	r,a,i,t
ps	Price of domestic supply	r,i,t
pe	Prices for bilateral export supply	r,i,rp,t
pet	Average price of export supply	r,i,t
pefob	Border price of exports (free on board)	r,i,rp,t
pmcif	Border price of imports (cost, insurance, freight)	r,i,rp,t
pm	Bilateral price of imports, tax inclusive	r,i,rp,t
pmt	Average price of imports	r,i,t
pmtMrio	Price of aggregate imports, by mrio agent	r,i,mrioA,t
pd	Price of domestically produced good	r,i,t
pdp	Purchaser price of domestic good	r,i,aa,t
pdNest	Price aggregator for sub-nest below final demand equations	r,dNest,fd,t
pa	Armington prices	r,i,aa,t
pm	Bilateral price of imports, tax inclusive	r,i,rp,t
pmp	Public expenditure price deflator	r,i,aa,t
pcons	Consumer price deflator	r,h,t
pi	Investment expenditure price deflator	r,t
pg	Public expenditure price deflator	r,t
pfact	Public expenditure price deflator	r,t
pprod	World factor price index	r,t
pwfact	World factor price index	T
ptmg	Global price index of transport services by mode	m,t

## 2.16 Flexible nesting in the production function

While most CGE models apply the CES functional form to depict the production function, quite some differences exist how input composites are defined. Usually, each nest is represented in the model's programming code by its own quantity and price aggregator equation. Adding or changing nests thus requires coding efforts – new variables, equations and parameters need to be defined and properly assigned. Typically, also the equations relating to the top-level value-added aggregator and/or the intermediate demand nest need to be adjusted if the nesting is changed.

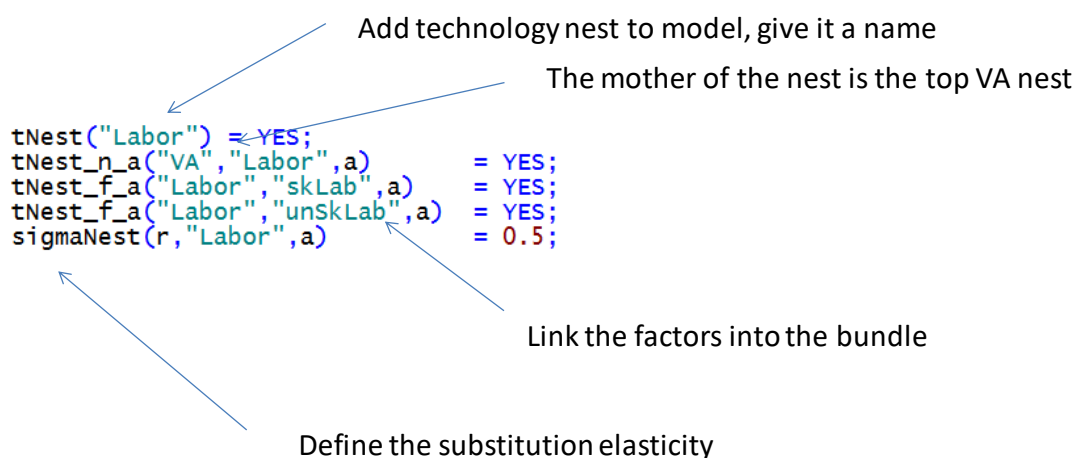
The GAMS code underlying CGEBox applies a different strategy. Here, nested CES structures in the production function are represented by a generic approach where a small number of equations and matching variables handle basically all possible nesting structures. The equations for the top-level VA and ND nests as discussed above are already set-up to host sub-nests along with equations describing sub-nests. In the standard GTAP-implementation, such sub-nests are not present and these equations empty.

That flexible and generic nesting approach is based on sets and cross-sets in GAMS which define the lists of factors, intermediates and sub-nests comprised in a CES composite nest, along with the upper-

level nest it belongs too. These nesting definitions enter the equations in the core model mentioned above and matching code dealing with parameter calibration. Hence, the user does not need to introduce additional equations in the code to use that feature – it is sufficient to provide the structure of the nesting to be applied via set definitions and the related substitution elasticities. The code also tests for potential errors such as duplicate assignments or sub-nests not linked into another nests.

The following examples should be sufficient to show the application of that feature and demonstrate its flexibility. Please note that the elasticity parameter in the current model version carry a “p\_” in front, in the model code, please use p\_sigmaNest where the screen shots still show sigmaNest.

1. An example of a sub-nest under the top VA nest which aggregates the two labor categories found in the GTAP 8 data base into an aggregate:



Note: The code will automatically remove the factors linked into nests from the top VA nest

The second example shows how to introduce a CES-composite of intermediates linked into the top ND-nest, as e.g. applied in the GTAP-E nesting, or to allow for high substitution of feed intermediates in animal production:



Add technology nest to model, give it a name

```
tNest("primEne") = YES;
```

Define a helper set for the intermediates  
(not used elsewhere)

```
Set primEne(i) /
```

```
coa-c
oil-c
gas-c
```

The mother of the nest is the top ND-bundle

```
/;
alias(primEne,primEne1);
```

```
tNest_n_a("ND","primEne",a)
tNest_i_a("primEne",primEne,a)
sigmaNest(r,"primEne",a)
```

```
$ sum( (r,primEne1,t), io(r,primEne1,a,t)) = YES;
$ sum( (r,primEne1,t), io(r,primEne1,a,t)) = YES;
$ sum( (primEne1,t), io(r,primEne1,a,t)) = 0.5;
```

Link the intermediates into the bundle

Define the substitution elasticity

Note: The code will automatically remove intermediates linked into nests from the top ND bundle

- The third example shows how to combine intermediates and factors into a CES composite, which in the example is a sub-nest of the top VA nest:

Add technology nest to model, give it a name

```
tNest("energy") = YES;
```

Define a helper set for the intermediates  
(not used elsewhere)

```
Set scndEne(i) /
```

```
p_c-c
ely-c
gdt-c
```

The mother of the nest is the top VA-bundle

```
/;
tNest_n_a("VA","energy",a)
tNest_f_a("energy","capital",a)
tNest_i_a("energy",scndEne,a)
sigmaNest(r,"energy",a)
```

```
= YES;
= YES;
= YES;
= 0.5;
```

Link the factors into the bundle

Link the intermediates into the bundle

Define the substitution elasticity

- The last example reproduces the nesting of the GTAP-E model (the composite of skilled and unskilled labor is already shown above):

```

*
* --- top level nest, aggregate of capital and energy
*
tNest("CAP+ENE") = YES;
tNest_n_a("VA","CAP+ENE",a) = YES;
tNest_f_a("CAP+ENE","capital",a) = YES;
tNest_n_a("CAP+ENE","energy",a) = YES;
sigmaNest(r,"CAP+ENE",a) = 0.25;

tNest("energy") = YES;
tNest_i_a("energy","ele-c",a) = YES;
tNest_n_a("energy","non-electric",a) = YES;
sigmaNest(r,"energy",a) = 1.00;

tNest("non-electric") = YES;
tNest_i_a("non-electric","coa-c",a) = YES;
tNest_n_a("non-electric","non-coal",a) = YES;
sigmaNest(r,"non-electric",a) = 0.50;

tNest("non-coal") = YES;
tNest_i_a("non-coal","gas-c",a) = YES;
tNest_i_a("non-coal","oil-c",a) = YES;
tNest_i_a("non-coal","p_c-c",a) = YES;
sigmaNest(r,"non-coal",a) = 1.00;

```

The actual code in use for the GTAP-E is more complex as it reflects a potential aggregation of the detailed products, with statements such as:

```

set nonCoal0 / gas,oil,p_c/;
set nonCoal(i); nonCoal(i) $ sum(mapi(nonCoal0,i),1) = yes;

```

Depending on the resulting sets, sub-nests might be skipped if they collapse into one commodity.

The nesting used in a specific model run along with the substitution elasticities are reported in a table in the exploitation tools. Equally, the code reports the quantity and price aggregators for nests and the resulting values and aggregates them up over individual sectors and over regions.

It is important to note that the post-model reporting redefines the top-level VA and ND nests such that they match the usual definition, i.e. an aggregation over primary factors and intermediates, respectively. Information about these nests in the definition used in the model can be retrieved with the separate tables showing all technology nests.

A similar generic approach is implemented for factor supply: nested CET functions can be used to supply primary factors to the production sectors. The top level nest is labeled with “xft”. The following example used in the GTAP-AGR implementation shows that approach:

```

*
* --- build nesting structure for factor supply to agr / non-agr
*
fNest("agr") = YES;
fNest_a_f("agr",agr,fm) = YES;
fNest_n_f("xft","agr",fm) = YES;
omegaFnest(r,"agr",fm) = omegaF(r,fm);

fNest("nonAgr") = YES;
fNest_a_f("nonAgr",nonAgr,fm) = YES;
fNest_n_f("xft","nonAgr",fm) = YES;
omegaFnest(r,"nonAgr",fm) = omegaF(r,fm);

```

Again, the definition of what activities are agricultural ones is endogenously determined based on the mapping from the GTAP9 sectors to the sectors used in the benchmark SAM:

```

set agr0(i0) / pdr    "Paddy rice"
               wht    "Wheat"
               gro    "Cereal grains nec"
               v_f    "Vegetables, fruit, nuts"
               osd    "Oil seeds"
               c_b    "Sugar cane, sugar beet"
               pfb    "Plant-based fibers"
               ocr    "Crops nec"
               ctl    "Cattle,sheep,goats,horses"
               oap    "Animal products nec"
               rmk    "Raw milk"
               wol    "Wool, silk-worm cocoons"
/;

```

```

set agr(a); agr(a) $ sum(mapa(agr0,A),1) = YES;

```

```

set nonAgr(a); nonAgr(a) $ (not agr(a)) = YES;

```

Similar to the CES production nests, the post-model processing reports the structure, parameterization and simulation results in the exploitation tools.

Finally, these flexible nestings are also applicable for final demand (government, investments and households), as shown here again with the nesting used by GTAP-E:

```

dNest("energy") = YES;
dNest_i_fd("energy","ele-c",fdn) = YES;
dNest_n_fd("energy","non-electric",fdn) = YES;
dNest_n_fd("top","energy",fdn) = YES;
sigmaFDNest(r,"energy",fdn) = 1.00;

dNest("non-electric") = YES;
dNest_i_fd("non-electric","coa-c",fdn) = YES;
dNest_n_fd("non-electric","non-coal",fdn) = YES;
sigmaFDNest(r,"non-electric",fdn) = 0.50;

dNest("non-coal") = YES;
dNest_i_fd("non-coal","gas-c",fdn) = YES;
dNest_i_fd("non-coal","oil-c",fdn) = YES;
dNest_i_fd("non-coal","p_c-c",fdn) = YES;
sigmaFDNest(r,"non-coal",fdn) = 1.00;

```

The code allows for factors or intermediate inputs to be linked to several nests. That is currently only used for the GTAP-Melitz extension to distinguish variable and fixed costs, but no parameters are introduced to allow a more general application.

As mentioned, the flexible nesting requires additional equations in the model. We discuss these here briefly for the production function, the approach for factor supply and final demand is similar. The input composite for a nest consists potentially of factors, intermediate inputs and other nests. The resulting price index for a nest  $ptNest$  is defined as usually in the dual formulation:

```

e ptNest(rs,tNest,aIn(a),ts) $ xtFlag(rs,tNest,a) ..

ptNest(rs,tNest,a,ts) =e=
  sum(r_r(rs,rsNat),
  (
  { [
* --- aggregate over intermediate inputs assigned to technology nest
*
    sum(tNest_i_a(tNest,i,a) $ p_ioNest(rs,i,a,tNest,ts),
        p_ioNest(rs,i,a,tNest,ts)
        * (m_pa(rsNat,i,a,ts)/lambdaio(rs,i,a,ts))**(1-p_sigmaNest(rs,tNest,a)))
*
* --- aggregate over factors assigned to technology nests
*
    + sum(tNest_f_a(tNest,f,a) $ p_afNest(rs,f,a,tNest,ts),
        p_afNest(rs,f,a,tNest,ts)
        * (m_pfa(rs,f,a,ts)/lambdaf(rs,f,a,ts))**(1-p_sigmaNest(rs,tNest,a)))
*
* --- aggregate over sub technology nests assigned to that technology nests
*
    + sum(tNest_n_a(tNest,tNest1,a) $ p_atNest(rs,tNest1,a,ts),
        p_atNest(rs,tNest1,a,ts)
        * (ptNest(rs,tNest1,a,ts)/lambdaN(rs,tNest1,a,ts))**(1-p_sigmaNest(rs,tNest,a)))
    ]** (1/(1-p_sigmaNest(rs,tNest,a)))
  )
  $ (p_sigmaNest(rs,tNest,a) ne 1)

```

The first summation aggregates the intermediate inputs assigned to the nest, based on the nest-specific input coefficient  $ioNest$ , Hicks non-neutral productivity shifters  $lambdaio$  and the sectoral specific tax inclusive price for the intermediate via the macro  $m\_pa$ . In a similar fashion, the second summation blocks aggregate factor use, while the last block considers sub-nests linked into the current nests. That implies that a multi-level nested structure does not require additional equations in the model.

The Armington intermediate demand  $xa$  for commodity  $i$  by sector  $a$  is shown below. It is driven by the relation between the Armington price  $m\_pa$  relative to the intermediate composite price  $pnd$ , reflecting its benchmark cost share on the intermediate composite  $io$  and the current total intermediate composite demand  $nd$ , plus potential demand from nests, consider total demand for the nest  $xtNest$ , the benchmark cost shares  $ioNest$  and the price for the nest defined above:

```

*
* --- Demand for intermediates, equation (6) in VDM 2018, with extension for commodity nests
*
e xap(rs,iIn(i),aIn(a),ts) $ (xapFlag(rs,i,a) $ (not discr(rs)) ) ..

xa(rs,i,a,ts)/xa.scale(rs,i,a,ts) =e=
*
* --- input coefficient times total intermediate demand
*
  (p_io(rs,i,a,ts)*nd(rs,a,ts)/xa.scale(rs,i,a,ts)
*
* --- relative price impact (in case sigmnd is not 0), not part of standard GTAP model
*
    * [ { (pnd(rs,a,ts)/m_paint(rs,i,a,ts))*p_sigmand(rs,a) } $ p_sigmand(rs,a) + 1 $ (not p_sigmand(rs,a)) ]
    * lambdaio(rs,i,a,ts)**(p_sigmand(rs,a)-1)) $ p_io(rs,i,a,ts)
*
  + sum(tNest_i_a(tNest,i,a) $ p_ioNest(rs,i,a,tNest,ts),
*
* --- input coefficient times demand for technology nest
*
    p_ioNest(rs,i,a,tNest,ts) * xtNest(rs,tNest,a,ts)/xa.scale(rs,i,a,ts)
*
* --- relative price impact (in case sigmndtNest is not 0)
*
    * [ { (ptNest(rs,tNest,a,ts)/m_paint(rs,i,a,ts))*p_sigmaNest(rs,tNest,a) } $ p_sigmaNest(rs,tNest,a)
    * lambdaio(rs,i,a,ts)**(p_sigmaNest(rs,tNest,a)-1)) $ tNest_i_a("some",i,a);

```

The demand is additive, as one commodity  $i$  can be demanded by different nests. A similar equation drives the demand for individual factors and the demand for nests:

```

*
* --- Demand for technology nests (not part of standard GTAP model)
*
* xtNest(rs,tNest,aIn(a),ts) $ (xtFlag(rs,tNest,a) $ (not tNest_n_a("Top",tNest,a))) ..
*
* xtNest(rs,tNest,a,ts)/xtNest.scale(rs,tNest,a,ts) =e=
*
* --- nest is part of top level intermediate demand
*
* + [ ( p_atNest(rs,tNest,a,ts) * nd(rs,a,ts)/xtNest.scale(rs,tNest,a,ts)
*      * [ {(pnd(rs,a,ts)/ptNest(rs,tNest,a,ts))*p_sigmaNd(rs,a)
*
*          } $ p_sigmaNd(rs,a) + 1 $ (not p_sigmaNd(rs,a)) ]
*      * lambdaN(rs,tNest,a,ts)**(p_sigmaNd(rs,a)-1)
*      ) $ tNest_n_a("ND",tNest,a)
*
* --- nest is part of top level VA demand
*
* + ( p_atNest(rs,tNest,a,ts) * va(rs,a,ts)/xtNest.scale(rs,tNest,a,ts)
*      * [ {(pva(rs,a,ts)/ptNest(rs,tNest,a,ts))*p_sigmax(rs,a)} $ p_sigmax(rs,a) + 1 $ (not p_sigmax(rs,a)) ]
*      * lambdaN(rs,tNest,a,ts)**(p_sigmax(rs,a)-1)
*      ) $ tNest_n_a("VA",tNest,a)
*
* --- nest is part of other nests
*
* + sum(tNest_n_a(tNest1,tNest,a),
*      p_atNest(rs,tNest,a,ts) * xtNest(rs,tNest1,a,ts)/xtNest.scale(rs,tNest,a,ts)
*      * [ {(ptNest(rs,tNest1,a,ts)/ptNest(rs,tNest,a,ts))*p_sigmaNest(rs,tNest1,a)} $ p_sigmaNest(rs,tNest1,a) + 1 $ (not p_sigmaNest(rs,tNest1,a)) ]
*      * lambdaN(rs,tNest,a,ts)**(p_sigmaNest(rs,tNest1,a)-1)
*      )
*
* ] $ (p_atNest(rs,tNest,a,ts) ne 1)

```

Note that it would be possible to completely remove the ND and VA nest definitions from the equation structure and only rely on the nesting structure. In order to increase readability, especially in case the standard GTAP configuration is used, the separate equations for ND and VA are kept.

### 3. Modules

The following chapters describe the different modules which are currently integrated into CGEBox. In most cases, the code of a module consists of three blocks of code:

1. Code declaring and defining additional data and parameters needed for the module
2. Code declaring and defining the necessary equations
3. Code used for post-model processing of the results

A Module introduces additional or alternative equations into the overall model. Some modules are mutually exclusive, such as different option to define aggregates of Armington agents. For others, settings might need to be adjusted for combined use. For instance, the Armington with commitment terms, the Melitz specification and the spatial arbitrage formulation are mutually exclusive for one international product market, but it is possible to define for each non-overlapping lists of products to introduce these options in parallel into one model configuration.

#### 3.1 myGTAP module

##### 3.1.1 Summary

The myGTAP module allows representing factor earnings, income taxes paid, consumption and saving decisions and additional variables such as inter-household transfers for one or multiple private households in the model, replacing the regional household approach of the GTAP standard model. This also implies a separate government account. The user supplies the necessary data such the list of households and their specific factor stock shares via an additional file in a specific format as detailed below, they are not part of the standard GTAP data base. The GUI allows selecting such a file along with closures rules related to different variables found in the module.

##### 3.1.2 Motivation

Most CGEs application at single country level will make use of a household survey to provide distributional impact analyses beyond economic totals. The regional household approach of the

standard GTAP model is not suitable for that type of analysis as it allocates the economy-wide sum of factor and tax income based on modified CD function to the different final demand categories. This decouples earned and spent of the accounts for the individual agents by introducing a central utility maximizer which decides about how much income is available to these agents. Accordingly, the myGTAP extension (Minor and Walmsley 2013<sup>10</sup>) replaces the regional household of the standard GTAP model by separated accounts. We provide in here a module which features of myGTAP to ease the integration of key data from household surveys in the CGEBox framework. The module also covers features such as remittances, foreign aids and transfers.

As general with the CGEBox framework, the myGTAP extension can be combined in a modular fashion with the different other extensions of CGEBox such as flexible nestings or the Melitz model implementation. Furthermore, the framework can be applied as a single country model or a global one, in comparative –static or recursive-dynamic fashion. One key aspect of the myGTAP implementation is the possibility to provide detailed household data for one or some countries only while using an aggregate private household for the remaining ones, an approach also chosen for the GEMPACK implementation.

### **3.1.3 Implementation**

#### **Model equations**

The implementation proposed in here is rather basic. The government account as well the potentially multiple private household use each a CD function to decide about consumption and savings share in their income. Alternatively, utility can be fixed and saving rates adjusted – a closure often found for the government account. The income for the government account encompasses all tax income, inclusive direct taxes, plus foreign aid exchanges minus transfers to households. The income of the different private households is defined as factor income minus direct taxes, remittance and foreign capital income exchanges, net transfers between households and from government. All international transfers are in international currency.

In the myGTAP module, each private household  $h$  supplies its own factor stocks  $x_{fth}$  to generate factor income  $factyH$  net of depreciation. Total household income  $hinc$ , see equation (1), may include beside the factor income  $factyH$  additionally transfers from government  $transg$ , remittances  $remih$ , foreign aid to households  $fyiH$ , from other households  $transh$  in the same region, and considers paying direct taxes  $dirTax$ , remittances  $remoH$ , foreign aid  $fyoH$  and transfers to other households  $trnh$ , as well as a share  $p\_shrDep$  of debt servings  $p\_captrans$ :

---

<sup>10</sup> Minor, Peter and Walmsley, Terrie (2013): MyGTAP: A Program for Customizing and Extending the GTAP Database for Multiple Households, Split Factors, Remittances, Foreign Aid and Transfers, GTAP Working Paper No. 79, [https://www.gtap.agecon.purdue.edu/resources/res\\_display.asp?RecordID=4321](https://www.gtap.agecon.purdue.edu/resources/res_display.asp?RecordID=4321)

```

*
* --- household income definition
*
e_hInc(rs(r),h,ts(t)) $ (hFlag(r,h) $ (hinc.range(r,h,t) ne 0)) ..
    hInc(r,h,t)/hinc.scale(r,h,t) =E[
*
*         --- factor income (defined above)
*
        factYH(r,h,t) $ (factYH.range(r,h,t) ne 0)
*
*         --- debt servings in G-RDEM linked to global bank, distirbuted with fixed share
*
        - sum(r_r(r,Rnat),p_capTrans(rNat,t)*pnum(t))*p_shrDep(rs,h,t)
*
*         --- government transfers (normally fixed in real terms), multiplid with gov cons. index
*
        + trng(r,h,t)*sum(r_r(r,Rnat),pg(rNat,t))
*
*         --- remittances received, converted from international to local currency
*
        + remiH(r,h,t)*sum(r_r(r,Rnat),lcu(rNat,t))
*
*         --- share on foreign capitial income (currently not use in default model)
*
        + fyiH(r,h,t)*fyihShrCor(t)*sum(r_r(r,Rnat),lcu(rNat,t))
*
*         --- private transfers received (normally fixed in real terms)
*         update with consumer price index of sending household
*
        + sum(h0, trnh(r,h0,h,t)*pcons(r,h0,t))
*
*         --- direct taxes paid (as defined above)
*
        $$iftheni.hSubr %hSubr%==on
        -sum(f $ (sum( a, p_gfh(r,h,f,a,t)) + p_gfh(r,h,f,"natPool",t)),dirTax(r,f,h,t))
        $$else.hSubr
        -sum(f $ sum( r_r(rp,r), sum(a, p_gfh(rp,h,f,a,t)) + p_gfh(rp,h,f,"natPool",t)),dirTax(r,
        $$endif.hSubr
*
*         --- remittances sent, converted from international currency
*
        - remoH(r,h,t)*sum(r_r(r,Rnat),lcu(rNat,t))
*
*         --- foreign capitial income sent, converted from international currency
*
        - fyoH(r,h,t)*sum(r_r(r,Rnat),lcu(rNat,t))
*
        - sum(r_r(r,Rnat) $ (factYH.range(r,h,t) ne 0),factYCorr(rNat,t)
          * sum(sameas(t,tRun),factyh.1(r,h,tRun-1) + factyh.1(r,h,t) $ sameas("%t0%
        /sum( (rp,h0) $ (r_r(rp,rNat) $ hFlag(rp,h0) $ (factYH.range(r,h0,t) ne 0)),
          sum(sameas(t,tRun),factyh.1(r,h0,tRun-1)+ factyh.1(r,h0,t) $ sameas("%t0%
*
*         --- private transfer sent (normally fixed in real terms), at own consumer price inde
*
        - sum(h0, trnh(r,h,h0,t)*pcons(r,h,t))/hinc.scale(r,h,t);

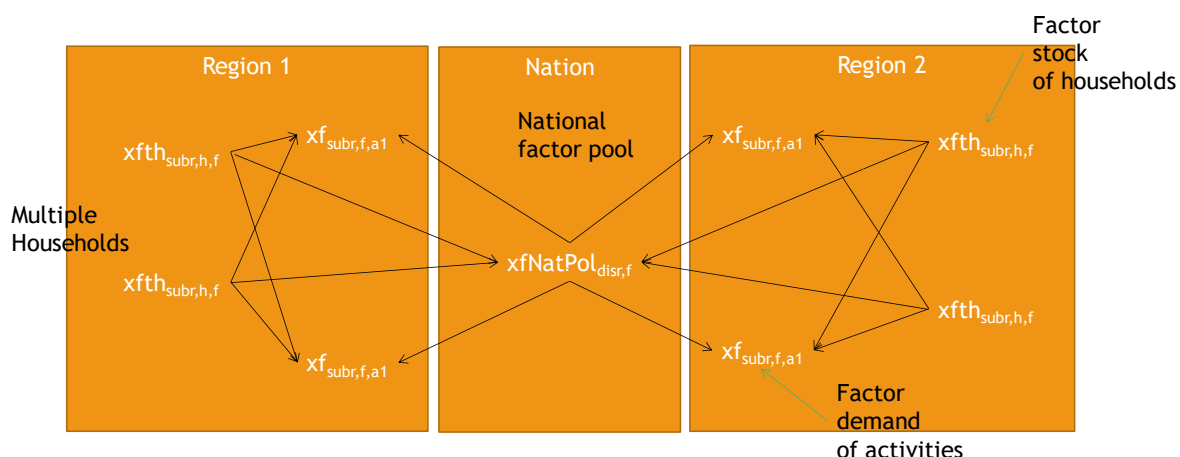
```

(1)

The elements trng, trnh, remih, remoh, fyoH, fyiH require additional data not available in the GTAP Data Base. They are therefore set to zero if not explicitly introduced by the user. However, entries for trng might be generated during benchmarking if data processing leads to negative household incomes or other undesired outcomes. Otherwise, they also default to zero.

In order to ensure homogeneity, some of these flows are multiplied with price indices: pg for transfers from government and pcons for household transfers. To consider the case of endogenous exchange rates lcu, all exchanges with the rest-of-the-world are converted into the international currency.





**Figure 4: Overview on factor supply in myGTAP module with sub-national households**

The myGTAP module also allows to link households to subsistence production in a sub-region as depicted in the graphic about. In that case, besides the direct allocation of household resources to production activities in the same region, it can allocate resources to a national pool from which they are distributed to the activities in the different regions. Specifically, the factor stock  $xf_{th}$  of a household  $h$  in sub-region  $subr$  is distributed based on a CET function to production activities in the same region (first line in equation (2), the relation is depicted by the left and right block in Figure 4 and to a national pool (second line in equation (2) and middle part of in Figure 4). That leads to the following composite price index  $pftH$  for the average returns to a factor  $f$  supplied by household  $h$ :

```

*
* --- average price received by household for supplying a factor to regional activities and the pool
*
e_pftH(rs(r),h,f,ts(t)) $ ((sum(a, p_gfh(r,h,f,a,t)) + p_gfh(r,h,f,"natPool",t))) $ pftH.range(r,h,f,t) ..
pftH(r,h,f,t)
= E= (
*
*      --- case with zero transformation elasticity, i.e. fixed shares
*
sum(a $ p_gfh(r,h,f,a,t), p_gfh(r,h,f,a,t) * pf(r,f,a,t))
+ p_gfh(r,h,f,"natPool",t) * sum(r_r(r,rNat), pftNatPool(rNat,f,t))) $ ( (p_omegafh(r,f) eq eps) $ p_omegafH(r,f))
*
*      --- sluggish supply via CET
*
+ [ (sum(a $ p_gfh(r,h,f,a,t), p_gfh(r,h,f,a,t) * pf(r,f,a,t) ** (1+p_omegafh(r,f)))
+ p_gfh(r,h,f,"natPool",t)
* sum(r_r(r,rNat), pftNatPool(rNat,f,t) ** (1+p_omegafh(r,f))) ** (1/(1+p_omegafh(r,f))) ] $ (p_omegafh(r,f) gt eps)
*
* --- Land under AEZ
*
+ pft(r,f,t) $ (not p_omegafH(r,f));
    
```

Where  $p\_gfh$  are the CET distribution parameters,  $pf$  are factor prices received for factor  $f$  employed in activity  $a$ , and  $pftNatPool$  is the price of the national factor pool. The cross-set  $r\_r$  depicts the relation between region  $r$  and the nation  $rNat$  it belongs to. If no sub-regions are presents, it comprises a diagonal relation between the nations themselves.

Note: The “\$ not  $p\_omegafH$ ” condition is used in conjunction with GTAP-AEZ. In that case, the average price of land in that region  $pft$  is paid to the households and fixed shares are used.

The factor demand of the activities  $xf$  is hence sourced by households in the same region  $r$  and a national pool of nation  $rNat$ :

```

*
* --- Regional use factor use of each activity
*
e_xfReg(rs(r),f,a,ts(t)) $ (xFlag(r,f,a) $ (p_omegafH(r,f) $ (pf.range(r,f,a,t) ne 0)
                                $ ( p_af(r,f,a,t)
                                or sum(tNest_f_a(tNest,f,a), p_afNest(r,f,a,tNest,t)))) ..
    xf(r,f,a,t) / Xf.scale(r,f,a,t)
=E=
*
* ---- households supplying factors to the regional activities in which it is located
* (the Leontief case with fixed shares is written separately)
*
    sum(h $ p_gfh(r,h,f,a,t),
        xftH(r,h,f,t)/Xf.scale(r,f,a,t)*pop(r,h,t)/1000 * p_gfh(r,h,f,a,t)) $ (p_omegafH(r,f) eq eps)
+   sum(h $ p_gfh(r,h,f,a,t),
        xftH(r,h,f,t)/Xf.scale(r,f,a,t)*pop(r,h,t)/1000 * p_gfh(r,h,f,a,t)
        * (pf(r,f,a,t)/pftH(r,h,f,t))*p_omegafH(r,f)) $ (p_omegafH(r,f) ne eps)
*
* ---- the national pool supply to the regional activities
*
+ sum(r $ r(r,rNat) $ ( p_gfNatPool(r,f,a,t) $ (p_omegafNatPool(rNat,f) eq eps)),
    xftNatPool(rNat,f,t)/Xf.scale(r,f,a,t) * p_gfNatPool(r,f,a,t))
+ sum(r $ r(r,rNat) $ p_gfNatPool(r,f,a,t),
    xftNatPool(rNat,f,t)/Xf.scale(r,f,a,t) * p_gfNatPool(r,f,a,t)
    * (pf(r,f,a,t)/pftNatPool(rNat,f,t))*p_omegafNatPool(rNat,f)) $ sum(r $ r(r,rNat) $ (p_omegafNatPool(rNat,f) ne eps),1);

```

(3)

The first expression in (3) above distributes a share of the per-capita factor stock  $xftH$  of the different households  $h$  in the same region to the activity  $a$ , depending on the relation between factor returns of that activity  $pf$  and the average returns at household level  $pftH$ , as defined above in equation (2). The second expression distributes a share of the per-capita factor stock to the endogenous national pool  $xftNatPool$ , using the relation to the average pool price  $pftNatPool$ .

As households supply a part of their factor stocks to a national pool (think about capital or migrant labor), the national pool  $xftNatPool$  is sourced by all households in this nation. That is captured by summing over both  $r_r(r,rNat)$  to cover all regions  $r$  belonging to the nation  $rNat$  and over households  $h$  in equation (4) below. The share of the factor stocks supplied to the national factor pool depends on the relation between the pool price  $pftNatPool$  and the average returns received by each household  $pftH$ :

```

*
* --- National pool (e.g. migrant labour): sourced by stocks of the different households (in different regions)
*
* Note: the factor stock of the household xftH are expressed at per capita basis (therefore the multiplication with pop)
*
e_xftNatPool(rsNat(rNat),f,ts(t)) $ (sum( (r_r(r,rNat),h) $ p_omegafH(r,f), p_gfh(r,h,f,"natPool",t))) ..
    xftNatPool(rNat,f,t) / XftNatPool.scale(rNat,f,t)
=E= sum( (r_r(r,rNat),h) $ (p_gfh(r,h,f,"natPool",t) $ p_omegafH(r,f)),
    [ xftH(r,h,f,t)/XftNatPool.scale(rNat,f,t)*pop(r,h,t)/1000 * p_gfh(r,h,f,"natPool",t) ] $ (p_omegafH(r,f) eq eps)
    + [ xftH(r,h,f,t)/XftNatPool.scale(rNat,f,t)*pop(r,h,t)/1000
        * p_gfh(r,h,f,"natPool",t) * (pftNatPool(rNat,f,t)/pftH(r,h,f,t))*p_omegafH(r,f) ] $ (p_omegafH(r,f) ne eps));

```

(4)

The average pool price  $pftNatPool$  is defined from the factor returns  $pf$  of the activities and the share parameters  $p\_gfNatPool$ :

```

* --- average price received for supplying primary factor from the national pool to the activities
* in the subnational regions
*
e_pftNatPool(rsNat(rNat),f,ts(t)) $ (sum( (r_r(r,rNat),h) $ p_omegafH(r,f), p_gfh(r,h,f,"natPool",t)) ) ..
    pftNatPool(rNat,f,t)
=E= sum( (r_r(r,rNat),a) $ p_gfNatPool(r,f,a,t), p_gfNatPool(r,f,a,t)*pf(r,f,a,t)) $ (p_omegafNatPool(rNat,f) eq eps)
    + [sum( (r_r(r,rNat),a) $ p_gfNatPool(r,f,a,t),
        p_gfNatPool(r,f,a,t)
        * pf(r,f,a,t)*(1 + p_omegafNatPool(rNat,f)) ) ** (1/(1+p_omegafNatPool(rNat,f))) ] $ (p_omegafNatPool(rNat,f) ne eps);

```

(5)

Note the different share parameters used:  $p\_gfh$  refers to the shares supplied by households,  $p\_gfNatPool$  to shares on the national pool.

The factor income reflects stocks  $xftH$  and related return  $pftH$ , and subtracts household specific shares  $p\_shrDep$  on depreciation:

```

*
* --- Factor income by households, from given (or elasticity/AEZ driven) stock and price determined above
*
e factYH(rs(r),h,ts(t)) $ (hFlag(r,h) $ (factYH.range(r,h,t) ne 0)) ..
    factYH(r,h,t)/factYH.scale(r,h,t)
    =e= [
        $$iftheni.hSubr "%hSubr%"=="on"
        *
        * --- household at sub-national level
        *
        sum( f $ (sum( a, p_gfh(r,h,f,a,t)) + p_gfh(r,h,f,"natPool",t)), xftH(r,h,f,t)*pop(r,h,t)/1000 *pftH(r,h,f,t))
        $$else.hSubr
        *
        * --- household at national level
        *
        sum( (r_r(rp,r),f) $ (sum( a, p_gfh(rp,h,f,a,t)) + p_gfh(rp,h,f,"natPool",t)), xftH(rp,h,f,t)*pop(rp,h,t)/1000 *pftH(rp,h,f,t))
        $$endif.hSubr
        *
        * ---- distribution depreciation based on fixed shares shrDep
        *
        - sum(r_r(r,rNat),p_fdepr(rNat,t)*pi(rNat,t)*kstock(rNat,t))*p_shrDep(r,h,t) ] /factYH.scale(r,h,t);

```

(6)

Income distribution is based on household specific saving shares betasH which are defined as variables to allow for closure swaps:

```

*
* --- saving of private households, share of household income
* (betasH might be endogenous of G-RDEM is active)
*
e rsavC(rs(r),h,ts(t)) $ hFlag(r,h) ..
    rsavC(r,h,t)/hinc.scale(r,h,t) =e= betaSH(r,h,t) * hInc(r,h,t)/hinc.scale(r,h,t);
*
e xsavC(rs(r),h,ts(t)) $ hFlag(r,h) ..
    xsavC(r,h,t)*sum(r_r(r,rNat),psave(rNat,t))/hinc.scale(r,h,t) =e= rsavC(r,h,t)/hinc.scale(r,h,t);

```

(7)

And:

```

*
* --- consumption by private households, share of household income
*
e yc(rs(r),h,ts(t)) $ hFlag(r,h) ..
    yc(r,h,t)/yc.scale(r,h,t) =e= (1-betaSH(r,h,t)) * hInc(r,h,t)/yc.scale(r,h,t);

```

(8)

The link to the remaining equations of the standard GTAP model is provided firstly by defining average direct tax rates:

```

*
* --- Weighted national average of household specific direct tax rates
*
e kappaD(rsNat(rNat),f,ts(t)) $ sum(a, xfflag(rNat,f,a)) ..
    (kappaD(rNat,f,t)+kappaShft(rNat,t)) * sum(a $ xfflag(rNat,f,a), pf(rNat,f,a,t)*xf(rNat,f,a,t)) /regy.scale(rNat,t)
    =E= [
        *
        *
        $$iftheni.hSubr "%hSubr%"=="on"
        sum( (r_r(rs,rNat),h) $ (hFlag(rs,h) $ (sum( a, p_gfh(rs,h,f,a,t)) + p_gfh(rs,h,f,"natPool",t))),dirTax(rs,f,h,t))
        $$else.hSubr
        sum(h $ (hFlag(rNat,h) $ sum( r_r(rp,rNat), sum(a, p_gfh(rp,h,f,a,t)) + p_gfh(rp,h,f,"natPool",t))),dirTax(rNat,f,h,t))
        $$endif.hSubr
    ] / regy.scale(rNat,t);

```

(9)

These enter the usual tax income equations which hence need not to reflect the different households and related factor ownership shares. Secondly, adding up from savings of the households and governments defines the regional savings used in the standard GTAP model:

```

*
* --- regional savings, sum of savings by households and saving by government
*
e rsav(rsNat(rNat),ts(t)) ..
    rsav(rNat,t)/rsav.scale(rNat,t) =e= (sum( (r_r(rs,rNat),h) $ hFlag(rs,h),rsavC(rs,h,t))+rsavG(rNat,t))/rsav.scale(rNat,t);

```

(10)

Equally, total factor income is defined:

```

*
* --- aggregated factor income
*
e factY(rsNat(rNat),ts(t)) ..
    factY(rNat,t)/factY.scale(rNat,t) =e= (sum( (r_r(rs,rNat),h) $ hFlag(rs,h), factYH(rs,h,t)))/factY.scale(rNat,t);

```

(11)

Remittance outflows from the households are defined as a share  $p\_remiitShr$  of labor income; the shares are calculated from GTAP-MIG data base if that is available:

```
*
* --- remittance outflows in local currency (global value*lcu) as a share of labor income
*
* e_remoh(rs(r),h,ts(t)) $ (hFlag(r,h) $ ((remohShr.range(r,h,t) ne 0) or (remohShr.up(r,h,t) ne 0)) $ (remoh.range(r,h,t) ne 0)) ..
*
* remoh(r,h,t)*sum(r_r(r,rNat),lcu(rNat,t))/remoh.scale(r,h,t)
*
*=E=
*
* ---- given share of the household of all regional remittances sent
*
* remohShr(r,h,t)/remoh.scale(r,h,t)
*
* $iftheni.gmig "%modulesGMIG%"=="on"
*
* --- the GMIG module determines bo-lateral remittances v_remit
*
* * sum((r_r(r,rNat),rNat1) $ sum(l,p_remit(rNat1,rNat,1)), v_remit(rNat1,rNat,t))
* $else.gmig
*
* --- GIG is not active: we distribute a fixed share of the regional labour income
*
* * sum((r_r(r,rNat),1,a) $ xfFlag(r,1,a), pf(r,1,a,t)*xf(r,1,a,t) * p_remitShr(rNat,"shr",1))
* $endif.gmig
*
```

(12)

Remittance inflows use the same data base to define a bi-lateral link:

```
*
* --- remittance inflows in global currency are a share of the global inflows
*
* e_remih(rs(r),h,ts(t)) $ (hFlag(r,h) $ (remih.range(r,h,t) ne 0)) ..
*
* remih(r,h,t)/remih.scale(r,h,t)
*
*=E=
*
* ---- given share of the household of all regional remittances received
*
* remihShr(r,h,t)/remih.scale(r,h,t)
* $iftheni.gmig "%modulesGMIG%"=="on"
*
* --- the GMIG module determines bo-lateral remittances v_remit
*
* * sum((r_r(r,rNat),rNat1) $ sum(l,p_remit(rNat,rNat1,1)), v_remit(rNat,rNat1,t))
* $else.gmig
*
* --- GIG is not active: we distribute a fixed share of all nations' rnat labour income to the country rNat1
* in which the household is located
*
* * sum((rNat,1,a) $ xfFlag(rNat,1,a), pf(rNat,1,a,t)*xf(rNat,1,a,t)*sum(r_r(r,rNat1),p_remitShr(rNat1,rNat,1)))
* $endif.gmig
*
```

(13)

Similarly, foreign capital income transfers are defined as share of capital income and a share of total global flows:

```
*
* --- foreign global capital inflows in global currency are a share of the global inflows
*
* e_fyih(rs(r),h,ts(t)) $ (hFlag(r,h) $ (fyih.range(r,h,t) ne 0)) ..
*
* fyih(r,h,t)/fyih.scale(r,h,t)
*=E= fyihShr(r,h,t) * sum((rp,h0) $ hFlag(r,h0), fyoh(rp,h0,t))/fyih.scale(r,h,t);
```

(14)

And

```
*
* --- foreign global capital outflows in local currency (fyoh*lcu) as share of capital income
*
* e_fyoh(rs(r),h,ts(t)) $ (hFlag(r,h) $ (fyoh.range(r,h,t) ne 0)) ..
*
* fyoh(r,h,t)*sum(r_r(r,rNat),lcu(rNat,t))/fyoh.scale(r,h,t)
*=E= fyohShr(r,h,t)
* * sum((sameas(f,cap),a) $ xfFlag(r,f,a), pf(r,f,a,t)*xf(r,f,a,t)*p_shrXf(r,h,f,a,t))/fyoh.scale(r,h,t);
```

(15)

Again, in case that these flows are fixed, shares adjust and a correction factor is introduced to main the global balance:

```

*
* --- ensure global balance, only active of fyih is fixed
*
e_fyihbal(ts(t)) ..
    sum( (r,h) $ hFlag(r,h), fyih(r,h,t))*fyihShrCor(t)/sum(r, regy.scale(r,t))
    =E= sum( (r,h) $ hFlag(r,h), fyoH(r,h,t))/sum(r, regy.scale(r,t));

```

(16)

Foreign aid paid by the government is defined as a share of factor income:

```

*
* --- foreign transfer in local currency as share of factor income
*
e_aido(rsNat,ts(t)) $ (aido.range(rsNat,t) ne 0) ..
    aido(rsNat,t)*lcu(rsNat,t)/regy.scale(rsNat,t) =E= aidoShr(rsNat,t) * facty(rsNat,t)/regy.scale(rsNat,t);

```

(17)

Inflowing aid is defined as a share of global outflows:

```

*
* --- share of incoming foreign aid in total global aid
*
e_aidi(rsNat,ts(t)) $ (aidi.range(rsNat,t) ne 0) ..
    aidid(rsNat,t)/regy.scale(rsNat,t) =E= aididShr(rsNat,t) * sum(rNat, aido(rNat,t))/regy.scale(rsNat,t);

```

(18)

In case that income aid is fixed, shares are endogenous and a correction factor ensures global balancing:

```

*
* --- ensure that sum of aidid = sum of aido, only active if aidid is fixed
*
e_aidiBal(ts(t)) ..
    sum(rNat, aidid(rNat,t))*aididShrCor(t)/sum(rNat, regy.scale(rNat,t))
    =E= sum(rNat, aido(rNat,t))/sum(rNat, regy.scale(rNat,t));

```

(19)

That leads to the following balance equation for government income:

```

*
* --- government account
*
e_yg(rsNat(rNat),ts(t)) $ (yg.range(rNat,t) ne 0) ..
    yg(rNat,t)/yg.scale(rNat,t) =E= [ytaxTot(rNat,t) - rsavG(rNat,t)
    + aidid(rNat,t)*aididShrCor(t)*lcu(rNat,t) - aido(rNat,t)*lcu(rNat,t)
    - sum((r_r(rs,rNat),h) $ hFlag(rs,h), trng(rs,h,t)*pg(rNat,t))]/yg.scale(rNat,t);

```

(20)

As both the government and the private households have their own saving variables, we have to add up to link up to the equation structure of GTAP standard to arrive a physical savings xsavG at national level:

```

*
* --- Savings by government, quantity index
*
e_xsavG(rsNat(rNat),ts(t)) ..
    xsavG(rNat,t)/yg.scale(rNat,t) * psave(rNat,t) =E= rsavG(rNat,t)/yg.scale(rNat,t);

```

(21)

The value of regional savings rsav is defined as:

```

*
* --- regional savings, sum of savings by households and saving by government
*
e_rsav(rsNat(rNat),ts(t)) ..
    rsav(rNat,t)/rsav.scale(rNat,t) =E= (sum((r_r(rs,rNat),h) $ hFlag(rs,h), rsavC(rs,h,t))+rsavG(rNat,t))/rsav.scale(rNat,t);

```

(22)

Equally, we need to introduce the net income transfer with the rest-of-the-world finc which enters the regional income equation, defined as:



```

*
* --- Net income transfers with ROW in local currency, enters regional income equation
* (the equation does not comprise the debt servings of G-RDEM; that is a separate entry
* in the household's income definition)
*
e_finc(rNat,ts(t)) $ rs(rNat) ..
finc(rNat,t) =E= + [aidi(rNat,t) + sum( (r_r(rs,rNat),h) $ hFlag(rs,h), remiH(rs,h,t)) + sum(h $ hFlag(rNat,h), fyiH(rNat,h,t))
- aido(rNat,t) - sum( (r_r(rs,rNat),h) $ hFlag(rs,h), remoH(rs,h,t)) - sum(h $ hFlag(rNat,h), fyoH(rNat,h,t))] *lcu(rNat,t);

```

(23)

Again, shares are defined as variables to allow for closure swaps. All other equations are untouched.

### 3.1.4 Government debt dynamics

In the standard layout discussed above, the government's consumptions and saving are directly driven by its tax income. An alternative mechanism drives these variables by GDP and introduces government bonds to close the account. This is described in this section.

With this layout, it is assumed that the saving rates of household is sensitive to changes in the relation between the consumer price index  $pCons$  and (expected) returns to savings  $pFinh$ . Higher consumer prices as well as higher returns to savings increase the saving rates based on a CES utility function with substitution elasticity  $p\_sigmaExph$ . The average price for the household's utility is accordingly a weighted average of these two prices. Let  $p\_aExph$  depict the expenditure share parameter attached to current consumption, the composite price index for region  $r$  and household  $h$  at time  $t$  for savings and consumption is defined as:

```

*
* --- average expenditure of households, CES weithed average of private consumption and investments
*
E pExpH(rs(r),h,ts(t)) $ (hFlag(r,h) $ (betasH.l(r,h,"%t0%") gt 0)) ..
pExpH(r,h,t) =E= [ p_aExph(r,h,t) * pCons(r,h,t)**(1-p_sigmaExph(r,h))
*
* --- if expected returns to savings go up, invest more
*
+ (1-p_aExph(r,h,t)) * (1/pFinh(r,h,t))**(1-p_sigmaExph(r,h)) ]**(1/(1-p_sigmaExph(r,h)));

```

The projected saving rates  $betash$  of the household stored on `betash.scale`, normalized with the benchmark rate, is accordingly sensitive to changes in returns in financial markets (physical investments, government bonds) relative to the price index defined above.

```

*
* --- the exogenously projected saving share reacts to changes in the returns in financial markets
*
e_betash(rs(r),h,ts(t)) $ (hFlag(r,h) $ (betasH.l(r,h,"%t0%") gt 0)) ..
betasH(r,h,t) =E= betasH.scale(r,h,t)/betasH.l(r,h,"%t0%")
* (1- p_aExph(r,h,t)) * ( pExpH(r,h,t)**pFinh(r,h,t))**p_sigmaExph(r,h);

```

Households are assumed to maximize expected revenues from their savings based on a CET function. Total savings are distributed between investments which increase the future capital stock and thus generate capital returns and government bonds which draw a fixed yield. Accordingly, the expected returns to household savings are a weighted average of the expected rate of return to capital,  $rore^{11}$ , and yields of newly issued government bonds,  $govBondYld$ . Both are normalized with their benchmark values to arrive at a unity price  $pFinH$  at the benchmark:

<sup>11</sup> The expected revenues are defined based on the variable *arent* as the returns to capital minus the ad-valorem direct tax rate on capital. This variable is then divided by the price index of investments and corrected for the depreciation rate to yield the variable *rorc* which is called the net rate of return to capital. From there, the expected rate of return to capital is defined by a dampening factor which reflect the end of year capital over the beginning of year capital stock.

```

*
* --- price index of expected return to capital and gov bond yields
* (CET revenue max)
*
e_pFinH(rs(r),h,ts(t)) $ (hFlag(r,h) $ (betasH.1(r,h,"%t0%") gt 0)) ..
pFinH(r,h,t) =E=
[ p_aFinRore(r,h,t) * sum(r_r(r,rnat),rore(rnat,t)/rore.1(rnat,"%t0%"))**(1+p_omegaFinH(rs,h))
+ (1-p_aFinRore(r,h,t)) * sum(r_r(r,rnat),govBondYld(rnat,t)/govBondYld.1(rnat,"%t0%"))**(1+p_omegaFinH(rs,h))
]**(1/(1+p_omegaFinH(rs,h)));

```

The amount of government bonds bought *newGovBond* is hence determined by the savings of the households *rSavc*, benchmark shares into investment *p\_aFinRore* and the interest paid by the government relative to the price index above:

```

*
* --- new government bonds are a share of the households savings, multiplied with the price relations
* between yield and the average returns to savings
*
e_newGovBond(rnat,ts(t)) ..
newGovBond(rnat,t) =E= sum((r_r(rs,rnat),h) $ (hFlag(rs,h) $ p_aFinRore(rs,h,t)),
rSavc(rs,h,t) * (1-p_aFinRore(rs,h,t)) * (govBondYld(rnat,t)/pFinH(rs,h,t)*govBondYld.1(rnat,"%t0%"))**p_omegaFinH(rs,h));

```

Government consumption *yg* is driven partly by nominal GDP *gdmp* and partly total tax income *yTaxTot*, minus the costs of MAC *valMaccs* and minus debt servings *p\_payPastGovBonds*. The share which is driven by the corrected tax income is called *spendShrByGovInc*:

```

*
* --- government account: gov consumption as a share of GDP (benchmark relation)
*
e_yg(rsNat(rNat),ts(t)) $ (yg.range(rNat,t) ne 0) ..
yg(rNat,t)/m_var2Scale1(yg,rNat,t) =E= yg.1(rnat,"%t0%")*m_var2Scale(yg,rnat,"%t0%")/yg.scale(rNat,t)
* [ gdmp(rnat,t)*m_var2Scale(gdmp,rnat,t)/(gdmp.1(rnat,"%t0%")*m_var2Scale(gdmp,rnat,"%t0%")) * (1 - spendShrByGovInc(rnat,t)
*
* --- let government consumption increase if it would otherwise run a surplus ...
*
+ ([0.005-ncpVUSin(newGovBond(rnat,t)/(yTaxTot(rNat,t)*m_var2Scale(yTaxTot,rnat,t)),0.005,1.E-6)]*500) $ (newGovBond.range(rnat,t) ne 0)
)
+ (yTaxTot(rnat,t)*m_var2Scale(yTaxTot,rnat,t)
$$if1 "%MACCsProcessEmis"=="on" - valMaccs(rnat,t)
- sum((r_r(rNat,t),h) $ hFlag(r,h), p_payPastGovBonds(r,h,t))/(yTaxTot.1(rnat,"%t0%")*m_var2Scale(yTaxTot,rnat,"%t0%"))
* spendShrByGovInc(rnat,t));

```

A similar equation drives government savings *rsavg* (regional savings by government):

```

*
* --- government account: gov savings as a share of GDP (benchmark relation)
*
e_rsavg(rsNat(rNat),ts(t)) $ ((rsavg.range(rNat,t) ne 0) $ (newGovBond.range(rnat,t) ne 0)) ..
rsavg(rNat,t)/rsavg.scale(rNat,t) =E= rsavg.1(rnat,"%t0%")/rsavg.scale(rNat,t)
* [ gdmp(rnat,t)*m_var2Scale(gdmp,rnat,t)/(gdmp.1(rnat,"%t0%")*m_var2Scale(gdmp,rnat,"%t0%")) * (1 - spendShrByGovInc(rnat,t)
+ (yTaxTot(rnat,t)*m_var2Scale(yTaxTot,rnat,t)
$$if1 "%MACCsProcessEmis"=="on" - valMaccs(rnat,t)
- sum((r_r(rNat,t),h) $ hFlag(r,h), p_payPastGovBonds(r,h,t))/(yTaxTot.1(rnat,"%t0%")*m_var2Scale(yTaxTot,rnat,"%t0%"))
* spendShrByGovInc(rnat,t)]
*
* --- demand for MACC
*
$$if1 "%MACCsProcessEmis"=="on" + valMaccs(rnat,t)/rsavg.scale(rNat,t)
;

```

The share *spendShrByGovInc* which is driven by the corrected government income depends on the debt servings relative to total government income minus debt servings:

```

*
* --- Share of government consumptions and saving driven by government income and not by GDP
*
e_spendShrByGovInc(rsNat(rNat),ts(t)) $ ((yg.range(rNat,t) ne 0) $ (newGovBond.range(rnat,t) ne 0)) ..
spendShrByGovInc(rnat,t) =E= [sum((r_r(rNat,t),h) $ hFlag(r,h), p_payPastGovBonds(r,h,t))
/[-ncpVUSin(-sum((r_r(rNat,t),h) $ hFlag(r,h), p_payPastGovBonds(r,h,t)),
- [yTaxTot(rnat,t)*m_var2Scale(yTaxTot,rnat,t)
$$if1 "%MACCsProcessEmis"=="on" - valMaccs(rnat,t)
- sum((r_r(rNat,t),h) $ hFlag(r,h), p_payPastGovBonds(r,h,t))],1.E-3)]]**(1/4);

```

The *ncpVUSin* is a smooth approximation and ensure that the shares stay below unity even if the debt servings would exceed the total tax income minus the debt serving. Due to the functional form, if 50% of the governments tax income is spend on serving debt, consumption and savings are solely driven by the tax income minus the debt servings and decoupled from GDP growth. This would imply that no new bonds are issued and the debt servings would start to fall. The exponent (1/4) will imply a rather sensitive reaction of this fiscal stability mechanism if debt serving relative to tax income increases.



The yield of government bonds is implicitly defined by closing the government account with the newly issued bonds:

```
*
* --- government account: demand for new bounds, implicitly defines their price
*
e_govAccountYld(rNat,rNat),ts(t)) $ (newGovBond.range(rNat,t) ne 0) ..

0 =e= [ytaxTot(rNat,t)*m_var2Scale(yTaxTot,rNat,t)
      + aidi(rNat,t)*aidiShrCor(t)*lcu(rNat,t) - aido(rNat,t)*lcu(rNat,t)
      - sum((r_r(rs,rNat),h) $ hFlag(rs,h), trng(rs,h,t)*pg(rNat,t))
      - yg(rNat,t)*m_var2Scale(yg,rNat,t)
      - rsavg(rNat,t)
      - sum((r_r(rNat,r),h) $ hFlag(r,h), p_payPastGovBonds(r,h,t))
      + newGovBond(rNat,t)]/yTaxTot.scale(rNat,t);
```

Other elements in the government account are net transfers to households *trng* and aid received from other governments *aidi* and paid out to other governments *aido*. These elements can also be zero depending on the underlying data (or assumptions).

Regional savings *rsav* as one components of savings besides depreciation and foreign savings in (I=S) are defined as the sum of household and government savings minus the value of newly issued government bonds:

```
*
* --- regional savings, sum of savings by households and saving by government
*
e_rsav(rsNat(rNat),ts(t)) ..

rsav(rNat,t)/rsav.scale(rNat,t) =e= (sum((r_r(rs,rNat),h) $ hFlag(rs,h),rsavC(rs,h,t))-newGovBond(rNat,t)+rsavg(rNat,t))/rsav.scale(rNat,t);
```

The GTAP Data Base does not comprise data on government savings which would allow to determine the deficit or surplus in its account. Available are data on total tax income and government final consumption. The rest are a contribution to the savings accounts. Re-bookings between different types of final consumption are a complex exercise, as expenditure shares of products, indirect tax rates and shares of the domestic and imported agent differ across the final consumption categories (private, government, investments).

Without touching the final demands and keeping I=S implies that a government deficit increases the government contribution of savings for investments and reduction in the contribution of final household. The newly issued bonds close the resulting gap in the budgets.

Benchmark of the equation structure above requires data on (1) the government deficit in the current year and on (2) yields of government bonds. For simplicity, we assume no repayment of past bonds.

In a first step, we increase the government savings by the assuming deficit (see *myGtap/myGtap\_cal.gms*):

```
*
* --- newly issued government bonds at benchmark, assume 5% gov deficit
*
newGovBond(rNat,t0) = p_budDeficit(rNat) * (yg.l(rNat,t0) + rsavg(rNat,t0));
*
* --- increase gov savings accordingly
*
rSavg(rNat,t0) = rsavg(rNat,t0) + newGovBond(rNat,t0);
xSavg(rNat,t0) = rsavg(rNat,t0);
```

Next, as we don't have information on the portfolio composition of individual households, we calculate total private savings as the sum of the households and derived from there the average expected returns:

```

*
* --- assume yield equal to half of expected return to capital
*
govBondYld(rnat,t0) = p_bondYield(Rnat);
*
* --- calculate average return to private savings
*
parameter p_privSav(rNat,t0);
p_privSav(rNat,t0) = sum( (r_r(r,rnat),h) $ hFlag(r,h), rsavc(r,h,t0));

pFinh(r,h,t0) $ hFlag(r,h) = sum(r_r(r,rNat), [ newGovBond(rNat,t0) * govBondYld(rnat,t0)/govBondYld(rnat,t0)
+ (p_privSav(rNat,t0)-newGovBond(rNat,t0) )*rore(rnat,t0)/rore(rnat,t0) ]
/p_privSav(rNat,t0));

```

Note that this calculation serves as a test only. Both prices are normalized with their benchmark levels and the spending on bonds and investment exhaust the total private savings. Accordingly, the resulting average revenue index is unity by definition.

Next, we set the transformation elasticity  $p_{\omega FinH}$  for the households and calculate the share parameter for the investments for each private household  $p_{aFinRore}$ :

```

*
* --- transformation elasticity in private savigns between investments and gov bonds
*
p_omegaFinH(r,h) = 2;
*
* --- assume same share of gov bonds in total household savings for all households
*
p_aFinRore(r,h,t0) $ hFlag(r,h) = sum(r_r(r,rNat), (p_privSav(rNat,t0)-newGovBond(rNat,t0))/p_privSav(rnat,t0)
* (rore(rnat,t0)/[pFinh(r,h,t0)*rore(rnat,t0)])**p_omegaFinH(r,h));

```

Finally, we set the substitution elasticity  $p_{\sigma Exph}$  for the distribution of household expenditures between final consumptions and savings and calculate its price index  $pExph$  and the share parameter for the expenditure on final demand on total household income  $p_{aExph}$ . The latter must be equal to one minus the given saving rates of the household.

```

*
* --- substitution elasticity between savings and final consumption
*
p_sigmaExph(r,h) = 0.2;
*
* --- average price of houshold expenditure (weighted average of consumer price index and returns to savings)
*
pExph(r,h,t0) $ hFlag(r,h) = [ (1-betash(r,h,t0))*pcons(r,h,t0) + betash(r,h,t0)*pFinh(r,h,t0) ];
*
* --- share parameter of private expenditures
*
p_aExph(r,h,t0) $ hFlag(r,h) = (1-betash(r,h,t0));

```

In order to determine the revenue stream from past bonds to each household, the model needs to keep track of yield levels and bought bonds by each household. This achieved by an additional parameter  $p_{bondReport}$ <sup>12</sup> which keeps track of how much bonds are bought at which yields in each simulated period, and calculates from there the revenues and add it up over past simulation points before the model is solved for the current period:

---

<sup>12</sup> The report can be added to the solve report output which is updated in between simulation points in recursive-dynamic runs to track the model behavior.

```

*
*   --- calculate payments from yields of old gov bonds
*
p_bondReport(r,h,"newBonds",tSim-1) $ (hFlag(r,h) $ (p_aFinRore(r,h,tSim-1) ne 1))
= sum( r_r(r,rnat) $ p_aFinRore(r,h,tSim-1),
      rsavC(r,h,tSim-1) * (1-p_aFinRore(r,h,tSim-1))
      * (govBondYld(rnat,tSim-1)/[pFinH(r,h,tSim-1)*govBondYld.1(rnat,"%t0%")]**p_omegaFinH(r,h));

p_bondReport(r,h,"newBondsToSavings",tSim-1) $ hFlag(r,h)
= p_bondReport(r,h,"newBonds",tSim-1)/rsavC(r,h,tSim-1);

p_bondReport(rNat,"gov","spendShrByGovInc",tSim-1) = spendShrByGovInc(rnat,tSim-1);
p_bondReport(rNat,"gov","newBonds",tSim-1) = newGovBond(rnat,tSim-1);
p_bondReport(rNat,"gov","NewBondsToTax",tSim-1) = newGovBond(rnat,tSim-1)/yTaxTot(rnat,tSim-1);

p_bondReport(r,h,"bondYld",tSim-1) $ hFlag(r,h) = sum(r_r(r,rnat), govBondYld(rnat,tSim-1));

p_bondReport(r,h,"rore",tSim-1) $ hFlag(r,h) = sum(r_r(r,rnat), rore(rnat,tSim-1));

p_bondReport(r,h,"bondRev",tSim-1) $ hFlag(r,h) = p_bondReport(r,h,"newBonds",tSim-1)*p_bondReport(r,h,"bondYld",tSim-1);

p_payPastGovBonds(r,h,tSim)
= sum(tRun $ (tRun.pos lt tSim.pos), p_bondReport(r,h,"bondRev",tRun)
      $ $ifi defined p_timeStep * p_timeStep(tRun)
      );

```

The report comprises some other positions to track the behavior of the mechanism

```

p_bondReport(rNat,"gov","payPast",tSim-1) = sum( (r_r(r,rnat),h),p_payPastGovBonds(r,h,tSim-1));
p_bondReport(rNat,"gov","payPastRel",tSim-1) = p_bondReport(rNat,"gov","payPast",tSim-1)/yTaxTot(rnat,tSim-1);
p_bondReport(rNat,"gov","GDPmP",tSim-1) = gdpmp(rnat,tSim-1);
p_bondReport(rNat,"gov","yTaxTot",tSim-1) = yTaxTot(rnat,tSim-1);
p_bondReport(rNat,"gov","yTaxTotAfterPayBond",tSim-1) = yTaxTot(rnat,tSim-1)-p_bondReport(rNat,"gov","payPast",tSim-1);
p_bondReport(rNat,"gov","yg",tSim-1) = yg(rnat,tSim-1);
p_bondReport(rNat,"gov","rsavg",tSim-1) = rsavg(rnat,tSim-1);
p_bondReport(rNat,"gov","grest",tSim-1) = -aidi(rNat,tSim-1)*aidiShrCor(tSim-1)*lcu(rNat,tSim-1) + aido(rNat,tSim-1)*lcu(rNat,tSim-1)
+ sum((r_r(r,rnat),h) $ hFlag(rs,h), trng(rs,h,tSim-1)*pg(rNat,tSim-1));

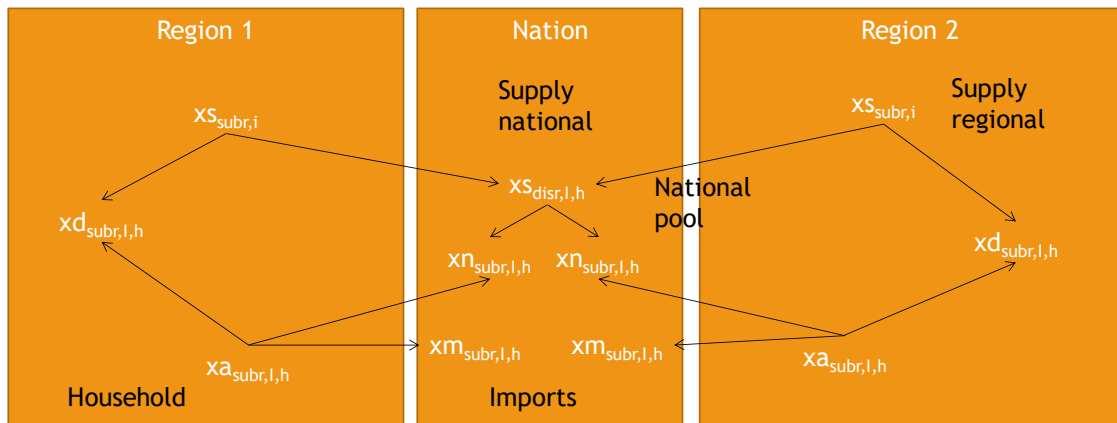
p_bondReport(rNat,"gov","SpendToTax",tSim-1) = ( p_bondReport(rNat,"gov","yg",tSim-1)
+ p_bondReport(rNat,"gov","rsavg",tSim-1)
+ p_bondReport(rNat,"gov","grest",tSim-1))/p_bondReport(rNat,"gov","yTaxTot",tSim-1);

p_bondReport(rNat,"gov","SpendToTaxNet",tSim-1) = ( p_bondReport(rNat,"gov","yg",tSim-1)
+ p_bondReport(rNat,"gov","rsavg",tSim-1)
+ p_bondReport(rNat,"gov","grest",tSim-1))/p_bondReport(rNat,"gov","yTaxTotAfterPayBond",tSim-1);

```

### 3.1.5 Methodology: Household demand

In opposite to GTAP standard, the module considers potentially three different origins in the top Armington nest: (1) regional demand, (2) national pool demand and (3) import demand. For all Armington agents which are not private households, the national pool demand layer is not present and defaults to what it called “regional” demand in the model, i.e. domestic demand. That is somewhat confusing (xd and pd might refer to the nation or the sub-region) but keeps the changes in the code structure minimal.



**Figure 5: Structure of demand system considering subsistence production**

The macro `m_pade` defining the Armington composite prices shown in equation (24) below hence comprises potentially three share parameters `p_alphan` (= national pool), `p_alphad` (domestic = region for private household, domestic for all other agents) and imports `alpham`. The macro `p_pdp` defines tax inclusive prices faced by the agents for the regional and the national origin. In case of the national pool, national prices are introduced as captured by the cross-set `r_r`. The domestic (or regional) demand uses the price of that region directly. The macro `p_pmp` define the import price plus agent specific taxes.

```

$macro m_pade(r,i,aa,t) { ( \
+ [alphan(r,i,aa,t)*sum(rr_rr(r,rNat),m_pdp(rNat,i,aa,t))**(1-sigman(r,i))] $ (alphan(r,i,aa,t) gt eps) \
+ [alphad(r,i,aa,t)*m_pdp(r,i,aa,t)**(1-sigman(r,i))] $ (alphad(r,i,aa,t) gt eps) \
+ [alpham(r,i,aa,t)*m_pmp(r,i,aa,t)**(1-sigman(r,i))] $ (alpham(r,i,aa,t) gt eps) \
) ** (1/(1-sigman(r,i))) }

```

(24)

The version of the macro shown above is only used if the global `hSubr` (= Household at sub-regional level) is set to “on” by a switch on the interface. Otherwise, the simpler default version in equation (25) is used which only considers the domestic and imported origin:

```

$macro mm_pade(r,i,aa,t) { ( [p_alphad(r,i,aa,t) * pdp(r,i,aa,t) ** (1-p_sigman(r,i))] $ p_alphad(r,i,aa,t) \
+ [p_alphan(r,i,aa,t) * (pmp(r,i,aa,t)/lambdaxm(r,i,aa,t)) ** (1-p_sigman(r,i))] $ p_alphan(r,i,aa,t) \
) ** (1/(1-p_sigman(r,i))) }

```

(25)

The demand equations are standard ones for a CES demand system; the national pool demand `xn` is defined as shown in equation (26) below. The left hand `xn` depends on the total Armington demand defined in the macro `mm_xa`, pre-multiplied with the share parameter for the national pool `p_alphan`. The resulting quantity is updated with the price relation between the Armington price `m_pa` and the national price faced by agent, considering the substitution elasticity between the imported and domestic origin `p_sigman` as supplied by the GTAP data base:

```

e_xn(subr,iIn(i),aa,ts) $ (p_alphan(subr,i,aa,ts) gt eps) ..
xn(subr,i,aa,ts)/xd.scale(subr,i,aa,ts) =e= p_alphan(subr,i,aa,ts)*mm_xa(subr,i,aa,ts)/xd.scale(subr,i,aa,ts)
* (m_pa(subr,i,aa,ts)/sum(r_r(subr,dier),m_pdp(dier,i,aa,ts)))**p_sigman(subr,i);

```

(26)

Note again the use of the `r_r` cross-set which puts the national price into the price relation.

The equation for the regional (respectively domestic) demand `xd` in equation (27) is unchanged from the existing CGEBox code, it comprises the same structural elements as above:

```

e_xd(rs,iIn(i),aa,ts) $ (p_alphad(rs,i,aa,ts) gt eps) $ (xd.range(rs,i,aa,ts) ne 0) ..
xd(rs,i,aa,ts)/xd.scale(rs,i,aa,ts) =e= p_alphad(rs,i,aa,ts)*mm_xa(rs,i,aa,ts)/xd.scale(rs,i,aa,ts)
* (m_pa(rs,i,aa,ts)/m_pdp(rs,i,aa,ts))**p_sigman(rs,i);

```

(27)

The total import demand equation `e_xmt` (28) now comprises an additional line which adds the import demand of sub-national households to the import demand of agents defined at national level (first line, the default from the GTAP Standard):

```

* --- Total import demand, equation (50) in VDM 2018
*
e_xmt(rsNat,iIn(i),ts) $ xmtFlag(rsNat,i) ..
xmt(rsNat,i,ts)/xmt.scale(rsNat,i,ts) =e= sum(aa $ (p_alphan(rsNat,i,aa,ts) gt eps), xmt(rsNat,i,aa,ts))/xmt.scale(rsNat,i,ts)
$if %ModuleGTAP_NUTS2%="ON" +sum( (subr,h) $ (r_r(subr,rsNat) $ (p_alphan(subr,i,h,ts) gt eps)),xmt(subr,i,h,ts))/xmt.scale(rsNat,i,ts)
;

```

(28)

The equilibrium for the national market depicted in the equation `e_pd` (29) is equally expanded by a second expression which considers demand of the private households from the national pool. The

preprocessor `$$ifthen-$$endif` commands allow to introduce the expression only in case the sub-national household layer is active:

```
*
* --- Domestic goods market equilibrium, equation (67) in VDM 2018
*
s pd(rsNat,iIn(i),ts) $ xdFlag(rsNat,i) ..
    xds(rsNat,i,ts)/xds.scale(rsNat,i,ts)
    =e= sum(aa $ (m_alphad(rsNat,i,aa,ts) gt eps), xd(rsNat,i,aa,ts))/xds.scale(rsNat,i,ts)
    $$iftheni.hsubr %hSubr%==on
    + sum( (rs,h) $ (r_r(rs,rsNat) $ (p_alphan(rs,i,h,ts) gt eps)),xn(rs,i,h,ts))/xds.scale(rsNat,i,ts)
    $$endif.hsubr
;
```

(29)

The equation `psSubreg` (30) which defines the share of domestic supply `xs` for dis-aggregated nations `disr` demanded from their sub-regions `subr` now considers the possibility that part of the demand stems directly from households in the same regions. That demand is subtracted in the first line from regional supply:

```
s psSubr(subr,iIn(i),ts) $ ( sum(a,xFlag(subr,a,i)) $ (xs.range(subr,i,ts) ne 0) $ diag(i)) ..
    (xs(subr,i,ts)
    $$iftheni.hsubr %hSubr%==on
    - sum(h $ (p_alphad(subr,i,h,ts) gt eps), xd(subr,i,h,ts))
    $$endif.hsubr
    )/xs.scale(subr,i,ts)
    =E= sum(r_r(subr,disr) $ p_gfi(subr,i,disr,ts),
    xs(disr,i,ts) * p_gfi(subr,i,disr,ts) * (ps(disr,i,ts)/ps(subr,i,ts))*p_sigmaSubr(disr,i))/xs.scale(subr,i,ts);
```

(30)

Thus, the output quantity of the sub-region `xs` is reduced by the demand of households. Only the remaining part feeds into the national output quantity `xs` of the dis-aggregated nation `disr`.

## Closures

The code currently supports two types of closures: either the driving equations which use a fixed share on a specific total as depicted above or fixing the transactions and rendering the shares endogenous:

```
$$iftheni "%closure_foreign_capital_income_sent%"=="Driven by capital income"
    fyoheq.fyoh
$$else
    fyoheq.fyohShr
$$endif
$$iftheni "%closure_foreign_capital_income_received%"=="Share on global total"
    fyiheq.fyih
$$else
    fyiheq.fyihShr
$$endif
$$iftheni "%closure_remittance_sent%"=="Driven by labor income"
    remoheq.remoh
$$else
    remoheq.remohShr
$$endif
$$iftheni "%closure_remittance_received%"=="Share on global total"
    remiheq.remih
$$else
    remiheq.remihShr
$$endif
$$iftheni "%closure_foreign_aid_sent%"=="Driven by factor income"
    aidoeq.aido
$$else
    aidoeq.aidoShr
$$endif
$$iftheni "%closure_foreign_aid_received%"=="Share on global total"
    aidieq.aidi
$$else
    aidieq.aidiShr
$$endif
```

The closures can be defined via the GUI as discussed below. As seen, that solution works currently globally. It is possible to fix also transactions for single countries and render shares endogenous, however, in that case, only the CNS and not the MCP solution algorithm can be used.

### Integrating data from household surveys

The module requires only very basic information assumed to be available from any household survey, namely factor income and population shares. Data on differences in direct tax rates or shares on total private savings can be provided additionally; otherwise identical savings and direct tax rates for all households as derived from the GTAP SAMs are used. International transfers, between households and from government to household are left out from the model if no data are entered.

Data are inputted as follows:

1. Factor income, savings and population are inputted as shares of each private household type on economic totals
2. Direct tax rates are relative changes against those of the aggregate average household
3. Transfers (foreign, between households, from government) are relative to factor income minus direct taxes
4. Foreign aid (in/out) is defined relative to total factor income

```

table shrHhsls(r,hhsls,*,t)
*
*   --- note: factor shares, savings and populationn as shares on aggregate total
*
*
*      land.(set.t)  unSkLab.(set.t)  skLab.(set.t)  capital.(set.t)  natRes.(set.t)
(set.r).poor      0.2          0.8          0.4          0.2          0.1
(set.r).rich      0.8          0.1          0.3          0.8          0.9
*
*   --- note: for direct taxes, these are relative differences in tax rates!
*
*
+      dirTax.(set.t)  savings.(set.t)  pop.(set.t)
(set.r).poor         0.5          0.3          0.8
(set.r).rich         1.5          0.7          0.2
*
*   --- note: for remiH, remoH,fyiH,fyoH,trng
*           these are shares on each household factor income minus direct taxes
*
+      remiH.(set.t)  remoH.(set.t)  fyiH.(set.t)  fyoH.(set.t)  trng.(set.t)
(set.r).poor        0.01          0.00          0.01          0.00          0.25
(set.r).rich        0.00          0.02          0.00          0.02          0.05
*
*   --- note: for household transfers (hsls (rows) -> hhslds (columns))
*           these are shares on each household factor income minus direct taxes
*
+      poor.(set.t)  rich.(set.t)
(set.r).poor        0.00          0.01
(set.r).rich        0.05          0.00
;
*
*   --- shares of foreign income to governments (in/ou), relative to factor income
*
*
table shrAid(r,*,t)
      aidI.(set.t)  aidO.(set.t)
(set.r)           0.01          0.02
.
;

```

The shares on factor income, savings and population are scaled to unity. That allows to either introducing them as shares (as shown below) or as totals. The scaling will in both cases take care of potential rounding or other editing errors. Note that for the other entries, absolute numbers must be translated to the respective shares. That can be done by data transformations in the very same file

using the SAM entries already available at the point in the code where the data information is included in the general calibration routine “cal.gms”.

The number of household types is comprised in the same file. Note the necessary “iftheni” clause which ensures that the information is made available to the code when the general set definitions are set-up in “build\loadGTAPagg.gms”:

```
$iftheni %1==decl
  set set_hhslsds(*) / poor,rich/;
$else
```

The code is set-up such that the household split up is introduced for one, some or all regions, solely driven by the data provided. Currently, the parameters of the CDE demand system are identical. The code allows for two options with regard to final demand by the disaggregated households:

1. The consumption shares between all households are identical.
2. The expansion parameters of the CDE system are used to update the consumption shares, and afterwards, the Armington demands for the different households are adjusted to simultaneously exhaust the given SAM entries and the final consumption total of the disaggregated households.

The user could provide household specific substitution elasticities between domestic and imports based on an extended parameter file, an option generally available for CGEBox.

An alternative is to assign factor income based on activities:

```
*
* --- assign income from agricultural activities to agricultural household
*
  shrHHslsds(r,"agr",agr,t) = 1;
..
```

Which can be combined with other shares. If data for activities are given, these have priorities.

## SAM rebalancing

The integration of new transactions between countries requires that global flows add up to zero. That condition is achieved by applying a uniform relative correction factors to the flows which distributes any original imbalances, here shown for the case of remittance flows:

```
*
* --- balance global flow of remittances
*
parameter p_surPlus(*,*,t);
p_surPlus("wor","rem",t) = sum( (r,hhslsds), remoH.l(r,hhslsds,t) - remiH.l(r,hhslsds,t));
p_surPlus("wor","wgt",t) = sum( (r,hhslsds), remoH.l(r,hhslsds,t) + remiH.l(r,hhslsds,t));
remoH.l(r,hhslsds,t) $ p_surPlus("wor","wgt",t) = remoH.l(r,hhslsds,t) * ( 1 - p_surPlus("wor","rem",t)/p_surPlus("wor","wgt",t));
remiH.l(r,hhslsds,t) $ p_surPlus("wor","wgt",t) = remiH.l(r,hhslsds,t) * ( 1 + p_surPlus("wor","rem",t)/p_surPlus("wor","wgt",t));
```

The first line calculates the imbalance in the original data (“rem”), the second the weights (“wgt”) to derive a uniform correction factor. The two remaining statements correct all remittance flows such that a zero global balance is resulting. The same logic is applied to balance foreign capital income inflows and outflows and foreign aid flows to governments, and also transfers between households inside of one country.

The introduction of new global transfer exchanges will generally imply that BOP and BOT are no longer in balance as it is the case in the original SAMS. In order to close the balance again, we first define the new additional net inflows:



```
*
* --- foreign contribution to income (can be negative)
*
*   finc.l(r,t) = + aidi.l(r,t) + sum(h,remiH.l(r,h,t)) + sum(h, fyiH.l(r,h,t))
*               - aido.l(r,t) - sum(h,remoH.l(r,h,t)) - sum(h, fyoH.l(r,h,t));
```

In order to maintain that BOT=BOP, foreign savings are downward corrected accordingly:

```
*
* --- correct foreign savings by contribution to income to maintain BOT
*
*   savf.l(r,t) = savf.l(r,t) - finc.l(r,t);
```

Which in turn requires, according to I=S, that the regional saving are re-balanced as well, using the same absolute correction:

```
*
* --- and correct regional savings in opposite direction to main I=S
*
*   rsav.l(r,t) = rsav.l(r,t) + finc.l(r,t);
```

That ensures that aggregate private consumption, government consumption and savings for each commodity need not to be rebalanced. The reader should note that the re-balancing approach was not checked for compliance with the myGTAP approach.

### Estimating final demand shares

In case the CDE expansion parameters are used to define final demand, the following procedure is used. As a first step, the Armington demands are derived from the given per capita income differences:

```
xa.l(r,i,hhslds,t) $ xa.l(r,i,"hhsld",t) = xa.l(r,i,"hhsld",t)
* yc.l(r,hhslds,t)/yc.l(r,"hhsld",t)
* [(yc(r,hhslds,t)/pop(r,hhslds,t))/(yc(r,"hhsld",t)/pop(r,"hhsld",t))] ** (eh(r,i,t));
```

The first line reflects differences in total final household consumption which depends on the data on factor income and direct tax shares of each household. The second line reflects differences in per capita income and the expansion effect based on the eh parameter of the CDE demand system.

In order to render the updated Armington demands for the different households consistent both with the given total spent for private consumption yc for each household and with the aggregate SAM entries for private consumption for each commodity, the following two consistency equations are used:

```
*
* --- sum of Armington demands by households must exhaust SAM entry
*
*   e_xaCons(rs,i,t) $ xa.l(rs,i,"hhsld",t) ..
*
*       sum(hhslds, xa(rs,i,hhslds,t)) =E= xa.l(rs,i,"hhsld",t);
*
*   e_ycCons(rs,hhslds,t) ..
*
* --- sum of Armington demands by households must exhaust given
*   private consumption
*
*       sum(i, xa(rs,i,hhslds,t)) =E= yc.l(rs,hhslds,t);
```

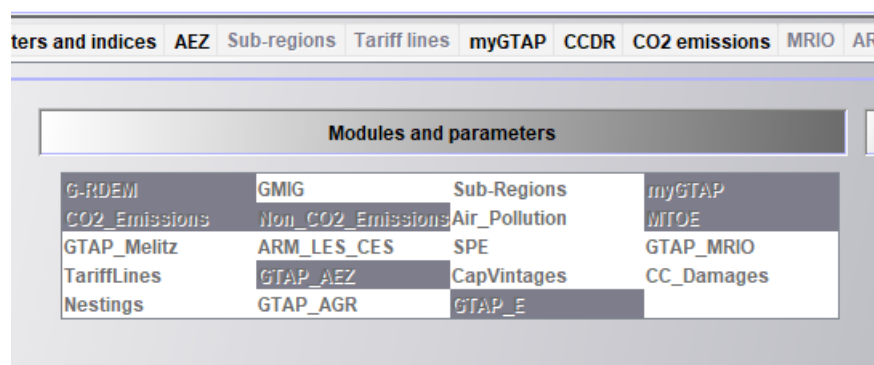
These are combined into a simple model which minimizes squared relative differences from the estimates resulting from applying the expansion elasticities:

```
e_hdpXA ..
v_hdp * sum( (rs,i,hhslds,t) $ xa.1(rs,i,hhslds,t),1)
=E= sum( (rs,i,hhslds,t) $ xa.1(rs,i,hhslds,t),
sqr( (xa(rs,i,hhslds,t)-xa.1(rs,i,hhslds,t))/xa.1(rs,i,hhslds,t))) ;
```

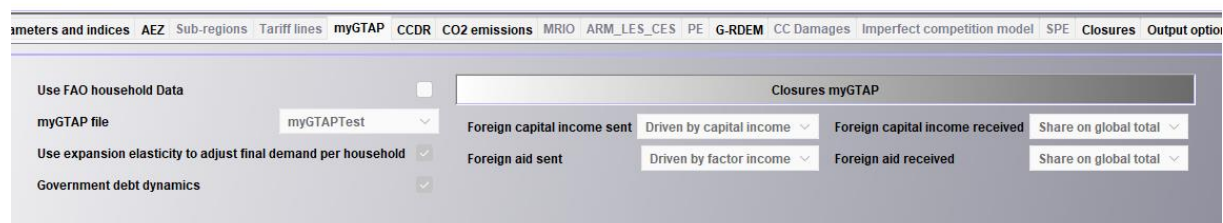
### 3.1.6 Graphical User Interface

#### Options




In order to use the extension, the “myGTAP” module must be switched on the interface:



In that case a tab becomes visible which allows detailing options for the myGTAP module as shown below. The closures have been discussed above.



The selection box of files under “myGTAP file” allows selecting a file with data on the household disaggregation and international transactions as discussed above. These files are stored in “gams\scen\myGTAP”, currently three test files are provided in the repository:

-  myGTAP\_oneHHSld.gms
-  myGTAPTest.gms
-  regHousehold.gms

The “myGTAPTest.gms” comprises the code shown above with the data for several households and can be used as an example to introduce own data.

The “regHousehold.gms” file removes only the regional household from the GTAP Model and introduces separate accounts for the agents without any household differentiation:

```

$iftheni %1==decl
  $$onempty
    set set_hhsls(*) ;;
  $$offempty
  $endif

```

The “myGTAP\_oneHHSLD.gms” shows how the extension can be used to introduce international transactions without dis-aggregating to different household types, i.e. all shares and relations for that single aggregate private household are set to unity, but remittance, foreign capital income and foreign information is provided:

```

$iftheni %1==decl
  set set_hhsls(*) / agg/;

  set set_myGTAP / dirTax,savings,pop,remiH,remoH,fyiH,fyoH,trng,aidi,aido /;
$else

  table shrHhsls(r,hhsls*,t)
  *
  *   --- note: for remiH, remoH,fyiH,fyoH
  *           these are shares on each household factor income minus direct taxes
  *
  *           land.(set.t)  unSkLab.(set.t)  skLab.(set.t)  capital.(set.t)  natRes.(set.t)
  (set.r).agg      1.0          1.0          1.0          1.0          1.0
  *
  *   --- note: for direct taxes, these are relative differences in tax rates!
  *
  *           dirTax.(set.t)  savings.(set.t)  pop.(set.t)
  (set.r).agg      1.0          1.0          1.0
  *
  *   --- note: for remiH, remoH,fyiH,fyoH,trng
  *           these are shares on each household factor income minus direct taxes
  *
  *           remiH.(set.t)  remoH.(set.t)  fyiH.(set.t)  fyoH.(set.t)  trng.(set.t)
  (set.r).agg      0.01          0.00          0.01          0.00          0.10
  ;

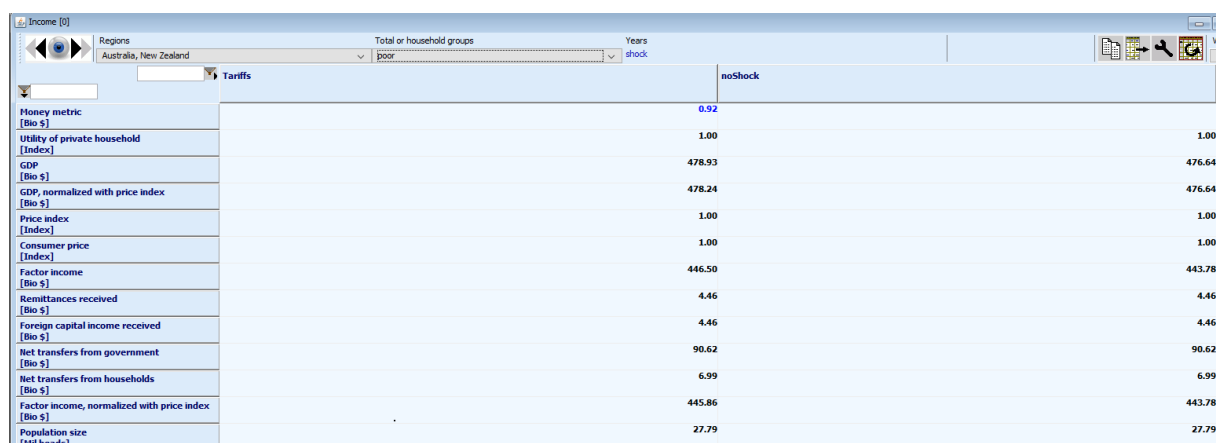
  *
  *   --- shares of foreign income to goverments (in/ou), relative to factor income
  *

  table shrAid(r,*,t)
  *           aidI.(set.t)  aido.(set.t)
  (set.r)      0.01          0.02
  ;
$endif

```

## Exploitation of myGTAP results

The post-processing code aggregates the information from single household to a regional private household and allows analyzing results at the level of the dis-aggregated household types:



	noShock	shock
Money metric [Bio \$]	0.92	
Utility of private household [Index]	1.00	1.00
GDP [Bio \$]	478.93	476.64
GDP, normalized with price index [Bio \$]	478.24	476.64
Price index [Index]	1.00	1.00
Consumer price [Index]	1.00	1.00
Factor income [Bio \$]	446.50	443.78
Remittances received [Bio \$]	4.46	4.46
Foreign capital income received [Bio \$]	4.46	4.46
Net transfers from government [Bio \$]	90.62	90.62
Net transfers from households [Bio \$]	6.99	6.99
Factor income, normalized with price index [Bio \$]	445.86	443.78
Population size [Mil heads]	27.79	27.79

On top, tables with demand and factor income information at that level are provided:

Household demand, by product [0]

Regions: Australia, New Zealand Oceania | Items: Quantity Q | Origins: Total tot | Years: shock shock | Percentage diff. to Scenarios: noShock

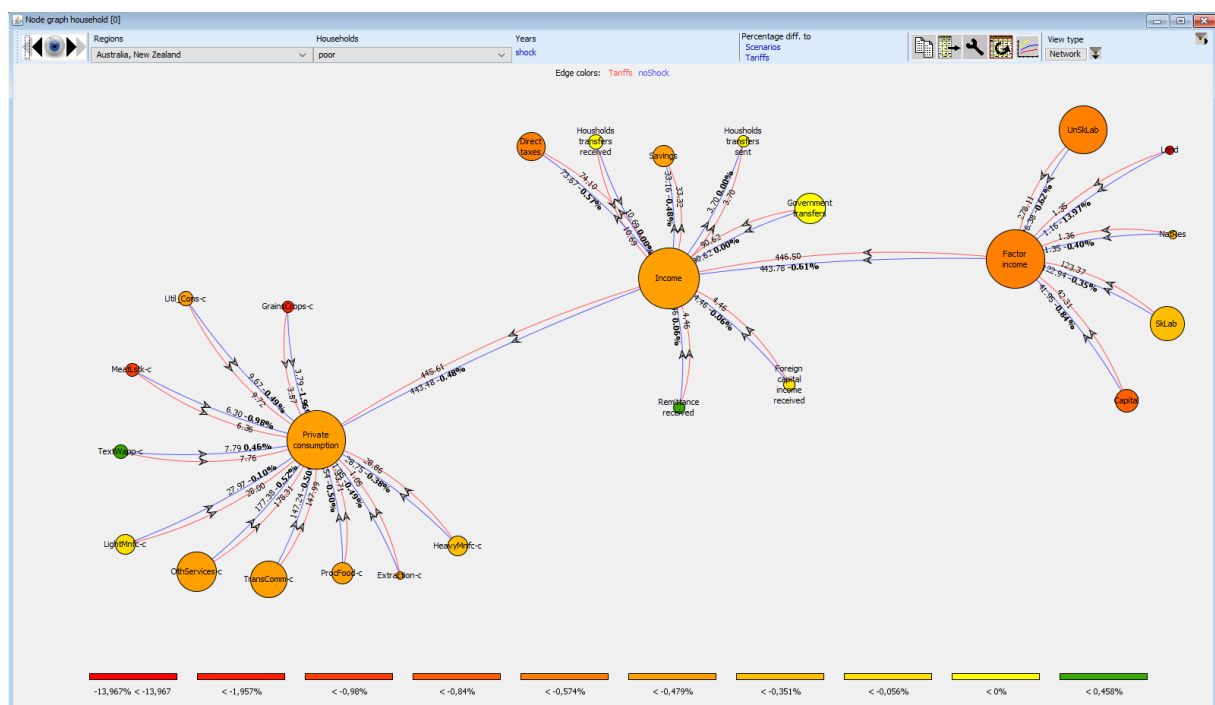
	Tariffs Tariffs			noShock noShock		
	Private consumption hhsld	poor poor	rich rich	Private consumption hhsld	poor poor	rich rich
Total totP	529.07	337.15	191.92	525.76	335.49	190.28
Total totP	529.07	337.15	191.92	525.76	335.49	190.28
Grains and Crops GrainsCrops-c	4.72	3.01	1.71	4.72	3.01	1.71
Livestock and Meat Products MeatLstk-c	8.06	5.14	2.92	8.10	5.17	2.93
Mining and Extraction Extraction-c	1.38	0.88	0.50	1.37	0.88	0.50
Processed Food ProcFood-c	33.61	21.43	12.19	33.47	21.36	12.11
Textiles and Clothing TextWapp-c	8.55	5.45	3.10	8.32	5.31	3.01
Light Manufacturing LightMnfc-c	33.59	21.41	12.19	32.97	21.04	11.93
Heavy Manufacturing HeavyMnfc-c	29.38	18.72	10.66	29.05	18.54	10.51
Utilities and Construction Util_Conc-c	12.04	7.67	4.37	11.97	7.64	4.33
Transport and Communication TransComm-c	180.82	115.22	65.60	179.85	114.76	65.09
Other Services OthServices-c	216.92	138.23	78.69	215.94	137.79	78.15

Factor income [0]

Regions: Australia, New Zealand | Items: Value (net of taxes) | Origins: dom | Years: shock

	Total	Land	UnSkLab	SkLab	Capital	NatRes
Aggregate Household	762.02	6.73	313.46	216.38	212.01	
	0.91%	15.85%	0.82%	0.58%	1.08%	
poor	447.37	1.35	278.63	123.65	42.40	
	0.81%	15.85%	0.82%	0.58%	1.08%	
rich	314.65	5.39	34.83	92.74	169.61	
	1.06%	15.85%	0.82%	0.58%	1.08%	

Node graphs (in value, in value normalized with price index, and per capita) depict graphical income generation and use for each household type, including transfer transactions:



### 3.1.7 Code implementation

#### Data entry and calibration

The information on the list of dis-aggregated household types is provided by the include file chosen on the user interface. It defines the household types which technically co-exist with the aggregate private household found in the standard option termed “hhsld”:

```
set h(fd) Households /
    hhsld          "Aggregate Household"
    set.set_hhslds
/ ;
```

That aggregate “hhsld” is only used when the data are loaded, in the calibration code “cal.gms” and are again introduced post-model for reporting. The household split-up is introduced in “cal.gms”. In case that only the regional household mechanism is replaced, but one aggregate private household agent is used, the split-up parameters are simply set to all zeros:

```
$$ifthen not defined ShrHhslds
*
*   --- no data on share of households, assume
*       top level regional household, i.e. one aggregate household own all factors
*
    parameter shrHhslds(r,hhslds,*,t);
    option kill=shrHhslds;
    option kill=shrXF;
    set dummyHslds/dirTax,Savings/;
$$else
    shrXF(r,hhslds,f,t) = shrHhslds(r,hhslds,f,t);
$$endif
```

In that case or if for some country, no distributional data is provided., the ownership share to factors are all unity:

```
*
*   --- if factor shares are missing for all households, use the standard aggregate
*       (allows to use dis-aggregated data for one country in full global model)
*
loop(r $ (not sum( (hhslds,f,t), shrXF(r,hhslds,f,t))),
    shrXF(r,"hhsld",f,t) = 1;
);
```

Next, the code checks if the factor share information is consistent, i.e. if for one region, information for each household type and factor is edited:

```
loop( (r,f) $ (card(hhslds) $ (not sum( (hhslds,t), shrXF(r,hhslds,f,t)))),
    shrXF(r,hhslds,f,t) = eps;
    abort "Factor shares for some households reported, but for other missings ",shrXF;
);
```

If the user wants to introduce zero shares, it should edit an “eps” value in the table shown above. The code will automatically scale the data to unity which allows to provide absolute data in the table instead of shares:

```

: *
: * --- scale factor shares to unity
: *
:   shrXF(r,h,f,t) = shrXF(r,h,f,t)/sum(h0,shrXF(r,h0,f,t));
:

```

Next, differences in direct tax rates are introduced. A similar test as above is implemented. If no information is given, uniform rate are used:

```

*
* --- shift direct taxes by given difference in direct taxation
*
loop( r $ ( sum( (hhslds,t), shrHhslds(r,hhslds,"dirTax",t))
              $ (sum( (hhslds,t) $ shrHhslds(r,hhslds,"dirTax",t),1) ne card(t)+card(hhslds)) ),
      shrHhslds(r,hhslds,"dirTax",t) $ (not shrHhslds(r,hhslds,"dirTax",t)) = eps;
      abort "Direct tax shares for some households reported, but for other missings, look out for eps ",shrHhslds;
);

kappafH.1(r,f,"hhsld",t) = kappaf.1(r,f,t);
kappafH.1(r,f,hhslds,t) $ (not shrHhslds(r,hhslds,"dirTax",t)) = kappaf.1(r,f,t);
kappafH.1(r,f,hhslds,t) $ shrHhslds(r,hhslds,"dirTax",t) = kappaf.1(r,f,t) * shrHhslds(r,hhslds,"dirTax",t);

```

From that information, household specific factor income and direct taxes are derived:

```

*
* -- factor income for each household, based on share on factor ownership
*
factYH.1(r,h,t) = sum(f, pft.1(r,f,t)*xft.1(r,f,t)*shrXf(r,h,f,t))
                 - fdepr(r,t)*pi.1(r,t)*kstock.1(r,t)*sum(sameas(f, cap), shrXf(r,h, cap,t))
                 + sum((fnm,a), pf.1(r,fnm,a,t)*xf.1(r,fnm,a,t)*shrXf(r,h,fnm,t));

factY.1(r,t) = sum(f, pft.1(r,f,t)*xft.1(r,f,t)) - fdepr(r,t)*pi.1(r,t)*kstock.1(r,t)
              + sum((fnm,a), pf.1(r,fnm,a,t)*xf.1(r,fnm,a,t));
factYH.1(r,"hhsld",t) = factY.1(r,t);

*
* --- direct taxes by household
*
dirTax.1(r,f,h,t) =
+ ( sum(sameas(fm,f), kappafh(r,fnm,h,t) * pft(r,fnm,t)*xft(r,fnm,t)*shrXf(r,h,fnm,t))
+ sum((sameas(f,fnm),a), kappafh(r,fnm,h,t) * pf(r,fnm,a,t)*xf(r,fnm,a,t)*shrXf(r,h,fnm,t)) );

dirTax.1(r,f,"hhsld",t) =
+ ( sum(sameas(fm,f), kappaf(r,fnm,t) * pft(r,fnm,t)*xft(r,fnm,t))
+ sum((sameas(f,fnm),a), kappaf(r,fnm,t) * pf(r,fnm,a,t)*xf(r,fnm,a,t)) );

```

Which allows defining direct tax rates which recovers economic wide totals. Next direct taxes paid by the household types are updated:

```

kappaFh.1(r,f,hhslds,t) $ dirTax.1(r,f,hhslds,t)
= kappaFh.1(r,f,hhslds,t)/[sum(hhslds0,dirTax.1(r,f,hhslds0,t))/dirTax.1(r,f,"hhsld",t)];

dirTax.1(r,f,hhslds,t) =
+ ( sum(sameas(fm,f), kappafh(r,fnm,hhslds,t) * pft(r,fnm,t)*xft(r,fnm,t)*shrXf(r,hhslds,fnm,t))
+ sum((sameas(f,fnm),a), kappafh(r,fnm,hhslds,t) * pf(r,fnm,a,t)*xf(r,fnm,a,t)*shrXf(r,hhslds,fnm,t)) );

```

A similar approach is used for saving rates, i.e. if no information is given, the same shares as for the aggregate private household are used:

```

*
* --- determine savings
*
loop( r $ ( sum( (hhslds,t), shrHhslds(r,hhslds,"savings",t))
              $ (sum( (hhslds,t) $ shrHhslds(r,hhslds,"savings",t),1) ne card(t)+card(hhslds)) ),
      shrHhslds(r,hhslds,"savings",t) $ (not shrHhslds(r,hhslds,"savings",t)) = eps;
      abort "Savings shares for some households reported, but for other missings, look out for eps ",shrHhslds;
);

*
* --- scale saving shares to unity
*
shrHhslds(r,hhslds,"savings",t) = shrHhslds(r,hhslds,"savings",t)/sum(hhslds0,shrHhslds(r,hhslds0,"savings",t));

```

In order to distribute regional savings to the different accounts, first the government savings are calculated residually:

```

*
* --- savings of government: by definition difference between total tax income and government consumption
*
rsavg.l(r,t) = ytaxTot(r,t) - yg(r,t);

```

In absence, of any information of foreign contributions to the regional government, it is assumed that the government saving comes from regional savings:

```

*
* --- regional savings of the households are defined as the difference between total regional savings and
*      government savings
*
rsavc.l(r,"hhsld",t) = rsav(r,t) - rsavg(r,t);

```

These savings are distributed to the household types either by the information provided or using uniform rates:

```

*
* --- if information on households's share on total saving are given, apply them
*
rsavc.l(r,hhslds,t) $ shrHhslds(r,hhslds,"savings",t) = rsavc.l(r,"hhsld",t) * shrHhslds(r,hhslds,"savings",t);
*
* --- otherwise, use average regional saving rate
*
rsavc.l(r,hhslds,t) $ (not shrHhslds(r,hhslds,"savings",t))
= rsavc.l(r,"hhsld",t)*(factYh(r,hhslds,t) - sum(f,dirTax(r,f,hhslds,t)))/(factY(r,t)-yTax(r,"dt",t));

```

From there, saving rates per household type are calculated, and the private consumption spending is calculated as well as the related value share:

```

*
* --- calculate savings rates for the households
*
betaSh.l(r,h,t) $ factYh(r,h,t) = rsavc(r,h,t)/(factYh(r,h,t) - sum(f,dirTax(r,f,h,t)));
*
* --- calculate final consumption and final consumption share residually
*
yc.l(r,h,t) = factYh(r,h,t) - sum(f,dirTax(r,f,h,t)) - rsavc.l(r,h,t);
betaPh.l(r,h,t) $ factYh(r,h,t) = yc(r,h,t)/(factYh(r,h,t) - sum(f,dirTax(r,f,h,t)));

```

The household type use the same consumption tax rates and split-up of domestic and import demand. Note that the information is only introduced if different household types are introduced:

```

:
: loop(r $ sum( (hhslds,t) , yc.l(r,hhslds,t)) ,
: *
: * --- same prices and tax rates as for aggregate, same consumption composition
: *
: pa.l(r,i,h,t) = pa.l(r,i,"hhsld",t);
: pdp.l(r,i,h,t) = pdp.l(r,i,"hhsld",t);
: pmp.l(r,i,h,t) = pmp.l(r,i,"hhsld",t);
: dintx.l(r,i,h,t) = dintx.l(r,i,"hhsld",t);
: mintx.l(r,i,h,t) = mintx.l(r,i,"hhsld",t);
: xa.l(r,i,h,t) = xa.l(r,i,"hhsld",t) * yc.l(r,h,t)/yc.l(r,"hhsld",t);
: xm.l(r,i,h,t) = xm.l(r,i,"hhsld",t) * yc.l(r,h,t)/yc.l(r,"hhsld",t);
: xd.l(r,i,h,t) = xd.l(r,i,"hhsld",t) * yc.l(r,h,t)/yc.l(r,"hhsld",t);
:

```

In that case, information on the aggregate household will be deleted:



```

: *
: *      --- delete data for aggregate
: *
:      yc.l(r,"hhsld",t) = 0;
:      xa.l(r,i,"hhsld",t) = 0;
:
:      betaSh.l(r,"hhsld",t) = 0;
:      betaPh.l(r,"hhsld",t) = 0;
:      factYh.l(r,"hhsld",t) = 0;
:      rsavc.l(r,"hhsld",t) = 0;
:      dirTax.l(r,f,"hhsld",t) = 0;
:
:      xaFlag(r,i,h) = xaFlag(r,i,"hhsld");
:      xaFlag(r,i,"hhsld") = 0;

```

## Post processing code

The post processing code basically reverts the code the code from the calibration, i.e. it aggregates from the individual household types to the aggregate private household “hhsld”:

```

if ( sum( (rs,hhslds), pcons.l(rs,hhslds,tSim) ),

pdNest.l(rs,"top",h,tsim) = pcons.l(rs,h,tsim);
xdNest.l(rs,"top",h,tsim) $ pcons.l(rs,h,tsim) = yc.l(rs,h,tsim)/pcons.l(rs,h,tsim);

yc.l(rs,"hhsld",t) = sum(hhslds $ pcons.l(rs,hhslds,t), yc(rs,hhslds,t));
xcshr.l(rs,i,"hhsld",t) = sum(hhslds $ xaFlag(rs,i,hhslds), xa(rs,i,hhslds,t)*pa(rs,i,hhslds,t))/yc(rs,"hhsld",t);
xa.l(rs,i,"hhsld",t) = sum(hhslds $ xaFlag(rs,i,hhslds), xa(rs,i,hhslds,t));
xm.l(rs,i,"hhsld",t) = sum(hhslds $ xaFlag(rs,i,hhslds), xm(rs,i,hhslds,t));
xd.l(rs,i,"hhsld",t) = sum(hhslds $ xaFlag(rs,i,hhslds), xd(rs,i,hhslds,t));
factYh.l(rs,"hhsld",t) = sum(hhslds $ pcons.l(rs,hhslds,t), factYh.l(rs,hhslds,t));

pa.l(rs,i,"hhsld",t) $ xa.l(rs,i,"hhsld",t)
= sum(hhslds $ xaFlag(rs,i,hhslds), xa(rs,i,hhslds,t)*pa(rs,i,hhslds,t))/xa(rs,i,"hhsld",t);

pdp.l(rs,i,"hhsld",t) $ xd.l(rs,i,"hhsld",t)
= sum(hhslds $ xaFlag(rs,i,hhslds), xd(rs,i,hhslds,t)*pdp(rs,i,hhslds,t))/xd(rs,i,"hhsld",t);

pmp.l(rs,i,"hhsld",t) $ xm.l(rs,i,"hhsld",t)
= sum(hhslds $ xaFlag(rs,i,hhslds), xm(rs,i,hhslds,t)*pmp(rs,i,hhslds,t))/xm(rs,i,"hhsld",t);

uh.l(rs,"hhsld",t) = sum(hhslds, yc(rs,hhslds,"check")*uh(rs,hhslds,t))/sum(hhslds, yc(rs,hhslds,"check"));
pcons(rs,"hhsld",t) = sum(i,xcshr(rs,i,"hhsld",t)*pa(rs,i,"hhsld",t));
xaFlag(r,i,"hhsld") $ sum(h, xaFlag(r,i,h)) = 1;

betaP(rs,tSim) = yc.l(rs,"hhsld",tSim) / regy.l(rs,tSim);
betaS(rs,tSim) = rsav.l(rs,tSim) / regy.l(rs,tSim);
betaG(rs,tSim) = yg.l(rs,tSim) / regy.l(rs,tSim);

);

```

## 3.1.8 Integration FAO household data into CGEBOX

Wolfgang Britz, October 2019

### Motivation

The following section provides documentation on the integration of FAO household survey data into CGEBox. The use of the data is twofold: first, to define flexibly aggregate household types for use with the myGTAP module and, second, to develop a post-model micro-simulation tool based on the data set. Both are fully operational and encompass the two usual ways household survey data sets are used in CGE work.

Post-model simulation does not require a household differentiation in the CGE model itself. Thus, it has the advantage of keeping the structure of the CGE model separated from the specifics of the household data set. Indeed, the micro-simulation itself can be conducted completely independent from the software code of the CGE model. This argument is not of relevance here as we integrate the post-model micro-simulation transparently into the code of CGEBox itself. Introducing (aggregate) households in the model structure which manage their own factor stocks to maximize revenues can add depth to the analysis which might be missed in post-model simulation. Equally, household

demand behaviour might be more appropriate depicted if aggregate household are introduced in the CGE itself. CGEBox allows now both, also in combination.

### Description of the data set

The FAO household data set<sup>13</sup> covers 19 countries (Kenya, Ethiopia, Malawi, Tanzania, Niger, Uganda, Nigeria, Ghana, Bangladesh, Vietnam, Nepal, Cambodia, Indonesia, Nicaragua, Guatemala, Bolivia, Panama, Albania, Tajikistan), in many cases, also the region inside the country in which the household is located is available. In total, the data set covers more than 193td households. The data had been collected between 1992 and 2013 (see Table 4). While there are in total more than 500 items, many of the items are comprised only in a specific sample such as e.g. household possession of certain equipment. The common set covers 115 variables, mostly related to income composition and some classification variables (rural / urban, male or female head, household size etc.).

**Table 4: Data coverage**

Country	years	# of households	# of regions
Kenya	2005	12547	8
Ethiopia	2012	3789	10
Malawi	2004, 2011	23431	3
Tanzania	2009,2011,2013	11923	26
Niger	2011	3910	8
Uganda	2005,2012	8259	4
Nigeria	2020,2013	4936	6
Ghana	2005,2013	24626	10
Bangladesh	2005	10058	6
Vietnam	1992,2002,2008	43406	8
Nepal	2003	3908	5
Cambodia	2004	14978	5
Indonesia	2000	10063	19
Nicaragua	1998,2005	10900	4
Guatemala	2006	13640	8
Bolivia	2005	4021	9
Panama	2003	6314	6
Albania	2005	3818	4
Tajikistan	2003,2007	7697	5

<sup>13</sup>

[http://www.fao.org/fileadmin/user\\_upload/smallholders\\_dataportrait/docs/Data\\_portrait\\_variables\\_description\\_new2.pdf](http://www.fao.org/fileadmin/user_upload/smallholders_dataportrait/docs/Data_portrait_variables_description_new2.pdf). Thanks to the FAO for making this highly valuable data set available for research.

### Income generation

We focus here mostly on income generation – how the income of the household is composed – at it is one main link of household survey data to CGE model results. As the data set is tailored to FAO needs and covers developing countries, it takes care to reflect the informal character of the agricultural sector in developing countries with subsistence features. This explains partly the top-level income categories found in the data set:

**Table 5: Top level categories in income generation**

Category label	Explanation
Selfemp	Income (partly imputed) from being self-employed outside the agriculture sector
Livstincome1gross	Income from livestock (mostly imputed, comprises the value of own-consumed crops)
Cropinome1gross	Income from crops (mostly imputed, comprises the value of own-consumed crops)
Transfer gross	Income from private and government transfers
agr_wge	Wage income from agriculture
Nonagr_wge	Wage income non-agriculture
Other	Mainly farm and non-farm rents

Alternatively, these items are expressed as shares on total income which might ease data processing. Checks have shown that the shares generally add up to unity while adding the absolute might show some imbalances. For the work with CGEBox, using shares is hence the implemented approach.

**Table 6 Top level share categories**

Category label	Explanation
Sh_agin	Income from agriculture (crops, livestock, farm rents)
shagwge	Wages from agriculture
shoffarm	Wages and self-employed non-agriculture
Shtrans	Income from private and government transfers
shother	Mainly non-farm rents

The shares of the top-level categories in Table 6 are broken down more detailed ones as shown in Table 7. The details allow linking the different wage and self-employed income categories of the households to some aggregate sectors in the economy, and in case of wages, also to two aggregate labor skill categories. That clearly adds depth to the link between the micro-data set and the macro-simulation in the CGE model.

**Table 7: Second level categories shares**

Category label	Explanation	Part of
Selfemp	Income (partly imputed) from being self employed outside the agriculture sector	

Livstincome1gross	Income from livestock including wages	sh_agin
Cropincome1gross	Income from crops including wages	sh_agin
Sh1privtransfer	Income from private transfers	shtrans
Sh1pubtransfer	Income from public transfers	shtrans
sh1wge2_1..3 .. shewge10_1..3	Wage income, different sectors (mining, manufacturing, electricity/utilities, constructions, commerce, transport/storage/communication, finance/insurance/real estate, other services, other); _1: skilled, _2: unskilled, _3: other	Shoffarm
Sh1selfemp (she2Self ... self1self)	Non-ag self-employed income, sectoral break as above	Shoffarm

### Data processing

The data in STATA format were converted by Olexandr Nekhay into CSV format. In order to import the data into GAMS, the CSV2GDX conversion routine was used:

```
$call csv2gdx "fao_hhsld.csv" useHeader=y id=p_dataFao index=2,4,7,6 values=8..lastCol acceptBadUels=Y
```

The outcome is a large GDX container with about 18 Mio non-zeros. It comprises a five-dimensional parameter with the household observations (country, region, household identifier, year, item). In order to speed up processing, first three cross sets were generated which show the relation between the country and the year, the country and the household id and the country and region for which data are available:

```
set country_hhid(country,hhid);
country_hhid(country,hhid) $ sum( (region,year), p_dataFao(country,region,hhid,year,"totIncome")) = YES;

set country_year(country,year);
country_year(country,year) $ sum( (region,hhid), p_dataFao(country,region,hhid,year,"totIncome")) = YES;

set country_region(country,region);
country_region(country,region) $ sum( (hhid,year), p_dataFao(country,region,hhid,year,"totIncome")) = YES;
```

Next, we compact the data by removing the year and region dimensions from the data set which are not needed for further applications and store the resulting parameter and the cross-sets to disk for later reuse:

```
p_compact(country_hhid(country,hhid),datVars)
= sum( (country_region(country,region),country_year(country,year)),
      p_dataFao(country,region,hhid,year,datVars));

execute_unload "compact.gdx" p_compact,country_hhid,country,hhid,region,year;
```

It should be noted that the household ids are unique for each country. The same id might however be found in several countries. Equally, one household might be found in multiple years.

### Linking the data set to the GTAP data

The income categories reported in the FAO data set do not match directly the sector resolution of GTAP. Similarly, the income definition comprises elements such as private and government transfers for which no data are available in the standard GTAP data set. The following paragraph briefly

documents how we derive from the income shares data to be used with the myGTAP module in CGEBox.

In order to link wage income to the GTAP data, we link the shares of wages as reported in the FAO household survey to GTAP labour skill categories and GTAP sectors:

```
set datvars_1_a(datVars,1,a) /

shlwge2_1.unSkLab .(coa-a,oil-a,gas-a,omn-a)
shlwge2_2.skLab .(coa-a,oil-a,gas-a,omn-a)
shlwge2_3.unskLab .(coa-a,oil-a,gas-a,omn-a)

shlwge3_1.unSkLab .set.manuf
shlwge3_2.skLab .set.manuf
shlwge3_3.unskLab .set.manuf

shlwge4_1.unSkLab .set.eleDist
shlwge4_2.skLab .set.eleDist
shlwge4_3.unskLab .set.eleDist

shlwge5_1.unSkLab .cns-a
shlwge5_2.skLab .cns-a
shlwge5_3.unskLab .cns-a

shlwge6_1.unSkLab .trd-a
shlwge6_2.skLab .trd-a
shlwge6_3.unskLab .trd-a

shlwge7_1.unSkLab .set.trsStoCom
shlwge7_2.skLab .set.trsStoCom
shlwge7_3.unskLab .set.trsStoCom

shlwge8_1.unSkLab .set.FinInsReal
shlwge8_2.skLab .set.FinInsReal
shlwge8_3.unskLab .set.FinInsReal

shlwge9_1.unSkLab .set.services
shlwge9_2.skLab .set.services
shlwge9_3.unskLab .set.services

/;
```

The sets used such as "set.manuf" list simply the GTAP sectors. A similar cross sets is used to link sector to self-employed earning shares:

```
set datvars_self(datVars,a) /
shlSelf2.(coa-a,oil-a,gas-a,omn-a)
shlSelf3.set.manuf
shlSelf4.set.eleDist
shlSelf5.cns-a
shlSelf6.trd-a
shlSelf7.set.trsStoCom
shlSelf8.set.FinInsReal
shlSelf9.set.services
/;
datVars_self("shlSelf10",nonAgr) = YES;
```

These relations are needed both for the application of the myGTAP module and for the post-model micro-simulation.

### Defining aggregate household types for the myGTAP module of CGEBox

The definition of household types can be managed via the interface where up to three classification

criteria can be defined:

Use FAO household Data <input checked="" type="checkbox"/>			
Classification criterion 1	incPerCa	q	25,50,75,100
Classification criterion 2	urban	v	rural 0, urban 1
Classification criterion 3	sh_Agin	v	"0-0.5','0.5-0.75','0.75-1.00"

The first field captures the name of the up to three items from the FAO household surveys of which the numerical values are used to assign households to classes. As shown in the second example, that can include flags such as rural/urban, male/female household head etc.

The second field defines if the limits entered should be interpreted directly (v=value) or defines quantiles (q). If quantiles are used, the household data are sorted (unweighted) by the item's numerical values and percentiles of these values define country specific class limits. The third field defines the numerical limits for the classes. Here, the following points need to be noted:

1. Classes must be ordered from low to high. (Attention here for flags, an input such as “female 1, male 0” will not produce the correct classification).
2. The last text field defines the elements of a set. Hence, the inputs must be valid GAMS syntax for the text between “SET /” and the closing “/;”. That especially means that limits defined as floats with a decimal point must be surrounded by quotation marks as shown in the example for “sh\_agin” (share of agricultural income). Otherwise, GAMS will assume that an element of a two-dimensional cross-set will be defined. Due to the interaction of the Java and GAMS code in the GUI, if quotation marks are used for one of the set elements, the whole text field must be surrounded by a second type of quotation marks as well.
3. If both a label and an explanatory text are given, the second element is interpreted as the numerical threshold. That is useful for cases which indicate flags as in the second field in the example above the label of the household will be comprise texts instead of “0” or “1”.
4. If a ‘-’ is found in a label, the limit is automatically derive from the number after the ‘-’, as shown in the “sh\_Agin” example above.
5. If only numbers are given, the name of the item is used as a prefix to ease reading the household names.

The example above will define a set with  $4 \times 2 \times 3 = 24$  household types. In order to reduce manual coding efforts and allow for a GUI driven definition of such classifications, embedded Python code defines the necessary cross-sets and defines sets such as shown in the example below.

The Python code firstly constructs easier to read set-definitions from the user input, for the example above, they look like:

---

```
set incPerCap/  
"incPerCap 25"  
"incPerCap 50"  
"incPerCap 75"  
"incPerCap 100"  
/;  
set urban/  
"rural"  
"urban"  
/;  
set sh_Agin/  
"incPerCap 0-0.5"  
"incPerCap 0.5-0.75"  
"incPerCap 0.75-1.00"  
/;
```

Secondly, it defines the names of the aggregate household types:

```
SET set_hhslsds /
"incPerCap 25_rural_sh_Agin 0-0.5"
"incPerCap 25_rural_sh_Agin 0.5-0.75"
"incPerCap 25_rural_sh_Agin 0.75-1.00"
"incPerCap 25_urban_sh_Agin 0-0.5"
"incPerCap 25_urban_sh_Agin 0.5-0.75"
"incPerCap 25_urban_sh_Agin 0.75-1.00"
"incPerCap 50_rural_sh_Agin 0-0.5"
"incPerCap 50_rural_sh_Agin 0.5-0.75"
"incPerCap 50_rural_sh_Agin 0.75-1.00"
"incPerCap 50_urban_sh_Agin 0-0.5"
"incPerCap 50_urban_sh_Agin 0.5-0.75"
"incPerCap 50_urban_sh_Agin 0.75-1.00"
"incPerCap 75_rural_sh_Agin 0-0.5"
"incPerCap 75_rural_sh_Agin 0.5-0.75"
"incPerCap 75_rural_sh_Agin 0.75-1.00"
"incPerCap 75_urban_sh_Agin 0-0.5"
"incPerCap 75_urban_sh_Agin 0.5-0.75"
"incPerCap 75_urban_sh_Agin 0.75-1.00"
"incPerCap 100_rural_sh_Agin 0-0.5"
"incPerCap 100_rural_sh_Agin 0.5-0.75"
"incPerCap 100_rural_sh_Agin 0.75-1.00"
"incPerCap 100_urban_sh_Agin 0-0.5"
"incPerCap 100_urban_sh_Agin 0.5-0.75"
"incPerCap 100_urban_sh_Agin 0.75-1.00"
/;
```

Third, it defines the cross-set which links the attribute values of each item to the household type:

```
..
$setglobal hhAggregSet hhlslsds_incPerCap_urban_sh_Agin(hhAgg,incPerCap,urban,sh_Agin)
SET hhlslsds_incPerCap_urban_sh_Agin(*,incPerCap,urban,sh_Agin) /
"incPerCap 25_rural_sh_Agin 0-0.5"."incPerCap 25"."rural"."sh_Agin 0-0.5"
"incPerCap 25_rural_sh_Agin 0.5-0.75"."incPerCap 25"."rural"."sh_Agin 0.5-0.75"
"incPerCap 25_rural_sh_Agin 0.75-1.00"."incPerCap 25"."rural"."sh_Agin 0.75-1.00"
"incPerCap 25_urban_sh_Agin 0-0.5"."incPerCap 25"."urban"."sh_Agin 0-0.5"
"incPerCap 25_urban_sh_Agin 0.5-0.75"."incPerCap 25"."urban"."sh_Agin 0.5-0.75"
"incPerCap 25_urban_sh_Agin 0.75-1.00"."incPerCap 25"."urban"."sh_Agin 0.75-1.00"
"incPerCap 50_rural_sh_Agin 0-0.5"."incPerCap 50"."rural"."sh_Agin 0-0.5"
"incPerCap 50_rural_sh_Agin 0.5-0.75"."incPerCap 50"."rural"."sh_Agin 0.5-0.75"
"incPerCap 50_rural_sh_Agin 0.75-1.00"."incPerCap 50"."rural"."sh_Agin 0.75-1.00"
"incPerCap 50_urban_sh_Agin 0-0.5"."incPerCap 50"."urban"."sh_Agin 0-0.5"
"incPerCap 50_urban_sh_Agin 0.5-0.75"."incPerCap 50"."urban"."sh_Agin 0.5-0.75"
"incPerCap 50_urban_sh_Agin 0.75-1.00"."incPerCap 50"."urban"."sh_Agin 0.75-1.00"
"incPerCap 75_rural_sh_Agin 0-0.5"."incPerCap 75"."rural"."sh_Agin 0-0.5"
"incPerCap 75_rural_sh_Agin 0.5-0.75"."incPerCap 75"."rural"."sh_Agin 0.5-0.75"
"incPerCap 75_rural_sh_Agin 0.75-1.00"."incPerCap 75"."rural"."sh_Agin 0.75-1.00"
"incPerCap 75_urban_sh_Agin 0-0.5"."incPerCap 75"."urban"."sh_Agin 0-0.5"
"incPerCap 75_urban_sh_Agin 0.5-0.75"."incPerCap 75"."urban"."sh_Agin 0.5-0.75"
"incPerCap 75_urban_sh_Agin 0.75-1.00"."incPerCap 75"."urban"."sh_Agin 0.75-1.00"
"incPerCap 100_rural_sh_Agin 0-0.5"."incPerCap 100"."rural"."sh_Agin 0-0.5"
"incPerCap 100_rural_sh_Agin 0.5-0.75"."incPerCap 100"."rural"."sh_Agin 0.5-0.75"
"incPerCap 100_rural_sh_Agin 0.75-1.00"."incPerCap 100"."rural"."sh_Agin 0.75-1.00"
"incPerCap 100_urban_sh_Agin 0-0.5"."incPerCap 100"."urban"."sh_Agin 0-0.5"
"incPerCap 100_urban_sh_Agin 0.5-0.75"."incPerCap 100"."urban"."sh_Agin 0.5-0.75"
"incPerCap 100_urban_sh_Agin 0.75-1.00"."incPerCap 100"."urban"."sh_Agin 0.75-1.00"
/;
```

The Python code also automatically defines the thresholds for the active classification criteria:



```

--f --
PARAMETER p_incPerCap(microR, incPerCap) /
(set.microR). "incPerCap 25" 25
(set.microR). "incPerCap 50" 50
(set.microR). "incPerCap 75" 75
(set.microR). "incPerCap 100" 100
/;
PARAMETER p_urban(microR, urban) /
(set.microR). "rural" 0
(set.microR). "urban" 1
/;
PARAMETER p_sh_Agin(microR, sh_Agin) /
(set.microR). "sh_Agin 0-0.5" 0.5
(set.microR). "sh_Agin 0.5-0.75" 0.75
(set.microR). "sh_Agin 0.75-1.00" 1.00
/;

```

In case of percentiles, such as for the item “incPerCap”, the values are replaced based on the statistical analysis. Again, to speed up processing, we define mapping sets for each classification criteria which indicates to which class a certain country household combination belongs. The code shown below uses globals (not shown) as the names of the fields are defined via the GUI and not hard-coded in the GAMS code itself:

```

set country_hhid_incPerCap(microR, hhid, incPerCap);
country_hhid_incPerCap(microR, hhid, incPerCap)
= sum( country_hhid(microR, hhid) $ (
    ((p_dataCompact(microR, hhid, "incPerCap") gt p_incPerCap(incPerCap-1)) or (incPerCap.pos eq 1))
    and (p_dataCompact(microR, hhid, "incPerCap") le p_incPerCap(incPerCap) ), 1);

set country_hhid_urban(microR, hhid, urban);
country_hhid_urban(microR, hhid, urban)
= sum( country_hhid(microR, hhid) $ (
    ((p_dataCompact(microR, hhid, "urban") gt p_urban(urban-1)) or (urban.pos eq 1))
    and (p_dataCompact(microR, hhid, "urban") le p_urban(urban) ), 1);

set country_hhid_femHead(microR, hhid, femHead);
country_hhid_femHead(microR, hhid, femHead)
= sum( country_hhid(microR, hhid) $ (
    ((p_dataCompact(microR, hhid, "femHead") gt p_femHead(femHead-1)) or (femHead.pos eq 1))
    and (p_dataCompact(microR, hhid, "femHead") le p_femHead(femHead) ), 1);

```

Next, we aggregate the household data using the aggregation weights. That requires linking each single household entry hhid to the aggregate category hhagg.:

```

parameter p_HHDataFao(microR, *, datVars);

p_HHDataFao(microR, hhagg, datVars) $ (not sameas(datVars, "weight"))
= sum( (hhlds incPerCap urban femHead(hhagg, incPerCap, urban, femHead),
country_hhid_urban(microR, hhid, urban)
country_hhid_femHead(microR, hhid, femHead)
),
p_dataCompact(microR, hhid, datVars) * p_dataCompact(microR, hhid, "weight"));

```

We need a separate statement for the weights:

```

p_HHDataFao(microR, hhagg, "weight")
= sum( (hhlds incPerCap urban femHead(hhagg, incPerCap, urban, femHead),
country_hhid_urban(microR, hhid, urban)
country_hhid_femHead(microR, hhid, femHead)
),
p_dataCompact(microR, hhid, "weight"));

```

After these statements, the data for the aggregate household comprise totals, i.e. the sum of the single observations multiplied with the weights. In order to express the items on a per household basis, we have to divide the sum of the weights. Equally, in order to ease inspecting the data, we express the weights for each aggregate household as percentages on the total:

```

p_agg(country,"all","weight") = sum(incLevl, p_agg(country,incLevl,"weight"));

p_agg(country,incLevl,datVars) $ (p_agg(country,incLevl,"weight") and (not sameas(datVars,"weight")))
= p_agg(country,incLevl,datVars) / p_agg(country,incLevl,"weight");

p_agg(country,incLevl,"weight")
= p_agg(country,incLevl,"weight") / p_agg(country,"all","weight")*100;

```

The resulting data set was checked for the exhaustion conditions discussed above, i.e. total income adding up and adding up of various share positions. The resulting data look like (only a selection):

	shincprod	sh_agin	shagwge	shoffarm	sl
incPerCap 75_rural_sh_Agin 0.75-1.00	0.84	0.93	0.03	0.01	
incPerCap 75_urban_sh_Agin 0-0.5	0.06	0.07	0.13	0.68	
incPerCap 75_urban_sh_Agin 0.5-0.75	0.58	0.59	0.07	0.14	
incPerCap 75_urban_sh_Agin 0.75-1.00	0.81	0.90	0.06	0.01	
incPerCap 100_rural_sh_Agin 0-0.5	0.12	0.13	0.11	0.72	
incPerCap 100_rural_sh_Agin 0.5-0.75	0.55	0.62	0.12	0.20	
incPerCap 100_rural_sh_Agin 0.75-1.00	0.77	0.94	0.02	0.01	
incPerCap 100_urban_sh_Agin 0-0.5	0.03	0.03	0.03	0.86	
incPerCap 100_urban_sh_Agin 0.5-0.75	0.59	0.62	0.08	0.19	
incPerCap 100_urban_sh_Agin 0.75-1.00	0.63	0.94	0.04	0.00	
all					

That approach offers quite some flexibility to define household types from the FAO data set without additional coding efforts. For more specific tasks, the code can be used as the basis to develop own aggregation routines which can be called from a file stored under “\scen\myGtap”.

### Integrating the data into the myGTAP accounting framework

Household surveys are a rich source to introduce household specific transactions into a CGE model, but their accounting logic often poorly matches the representation of transactions in a SAM. We present in the following an approach which tries to carry a larger share of the data reported on the earning side from the FAO’s household surveys into the myGTAP module of CGEBox.

The myGTAP module links primary factor earnings from the different production activities to the household by letting them manage their own primary factor stocks. The management is depicted in a stylized way by a revenue maximization problem represented by Constant Elasticity of Transformation (CET) function which captures the different real-world phenomena which render factor mobility sluggish. For instance, that means that households will not (and cannot) simply switch from being employed in one sector to another if wage rates change. Furthermore, it allows for household specific saving rates, direct tax rates and can depict government transfers as well as transfers between household types.

In order to determine spending behaviour, we use the AIDADS demand system built into CGEBox to estimate income dependent changes in the demand shares given the national averages found in the GTAP data base.

### Factor income and link to activities

The household survey data prepared as discussed above report shares of revenue categories (wages, public and private transfers, income from agriculture etc.) on total household income, total household income, household size and its relative weight.

#### 1. Agricultural income

Agricultural income is reported for crops and livestock activities. Using the household total income and the reported share for crop or livestock activities should match conceptually the total primary factor income from these activities in the SAM. We hence assign shares of these factor incomes based on the given weight (total income times shares from crop or livestock activities).

In order to produce a better match to the SAM, we first calculate the absolute income for each aggregate household and scale the result to yield shares. To do so, we multiply the per-household income with its population weight which should give the economic weight on total income. The total for each aggregate household is then divided by the sum over all household to yield shares:

```
*
* --- income, converted to shares
*
p_hhDataFao(faoR,hhsls,"incWeight") = p_hhDataFao(faoR,hhsls,"weight") * p_HHDataFao(faoR,hhsls,"totIncome");
p_hhDataFao(faoR,hhsls,"incWeight")
= p_hhDataFao(faoR,hhsls,"incWeight")/sum(hhsls0,p_hhDataFao(faoR,hhsls0,"incWeight"));
```

Based on these shares, we allocate distribution shares for the agricultural primary factors for crop and livestock activities to the households:

```
*
* --- assign income from crop and livestock production
*
shrXf(faoR,hhsls,f,crop_a,t0) $ xfFlag(faoR,f,crop_a)
= p_HHDataFao(faoR,hhsls,"shlCrop1") * p_HHDataFao(faoR,hhsls,"incWeight");

shrXf(faoR,hhsls,f,lstk,t0) $ xfFlag(faoR,f,lstk)
= p_HHDataFao(faoR,hhsls,"shlLivestock") * p_HHDataFao(faoR,hhsls,"incWeight");
```

## 2. Wages

The reported income from wages by skill category in different sectors (total factor income times reported share from wages) is used as a distribution key for wage income from the different sectors:

```
*
* --- assign wages from skilled and unskilled labour
*
shrXF(faoR,hhsls,l,a,t0) $ xfFlag(faoR,l,a)
= shrXF(faoR,hhsls,l,a,t0)
+ sum(datvars_l_a(wages,l,a), p_HHDataFao(faoR,hhsls,wages)) * p_HHDataFao(faoR,hhsls,"incWeight");
```

## 3. Income from self-employed:

The same logic as for wages is used:

```
*
* --- assign factor income for self employed
*
shrXF(faoR,hhsls,f,a,t0) $ xfFlag(faoR,f,a)
= shrXF(faoR,hhsls,f,a,t0)
+ sum(datVars_self(self,a), p_HHDataFao(faoR,hhsls,self)) * p_HHDataFao(faoR,hhsls,"incWeight");
```

If there are still missing shares, we use total non-farm income:

```
*
* --- if shares are still missing, use off farm income
*
shrXF(faoR,hhsls,f,a,t0) $ (xfFlag(faoR,f,a) and (not sum(hhsls0, shrXF(faoR,hhsls0,f,a,t0))))
= p_HHDataFao(faoR,hhsls,"incWeight") * p_HHDataFao(faoR,hhsls,"shOfFarm");
```

We also opted to introduce some assumptions on income dependent factor specific direct tax rates with the following parameter which depicts differences in tax rates. The first assumption is that households which receive a higher share of public transfer than the average across all households will pay lower direct tax rates:

```

parameter p_meanTransfer(r);

p_meanTransfer(faoR) = sum(hhslds, p_HHDataFao(faoR,hhslds,"shlpubTransfer")*p_HHDataFao(faoR,hhslds,"totIncome"));

shrHhslds(faoR,hhslds,"dirTax",t0) = eps;

shrHhslds(faoR,hhslds,"dirTax",t0) $ p_HHDataFao(faoR,hhslds,"popWeight")
= ( 1 - (p_HHDataFao(faoR,hhslds,"shlpubTransfer")-p_meanTransfer(faoR))) * p_HHDataFao(faoR,hhslds,"popWeight");

```

We estimate differences in saving rates based on the regression for macro-saving rates as part of G-RDEM, making sure that we don't yield negative saving rates and start from a minimum level of 5%. As the maximum rate found in the sample use for the regression where below 40%, we restrict the initialization to 50%:

```

shrHhslds(faoR,hhslds,"savings",t0) $ p_HHDataFao(faoR,hhslds,"popWeight")
= p_HHDataFao(faoR,hhslds,"incWeight") *
  min(0.5,max(1.E-3,max(0.05,betas.l(faoR,t0)) + (p_HHDataFao(faoR,hhslds,"incPerCap") - p_avgIncome(faoR))*0.0010213/1000))
  /max(0.05,betas.l(faoR,t0));

```

## Transfers

The data set reports the shares of private and public transfers received on income. The data cannot be easily used to assign some numbers. Instead a constrained optimization problem is defined. Its aim is not only to define the transfers, but also to generate income levels which in relative terms come close to those reported in the FAO data set.

It comprises the following balancing equations:

1. The reported income share on total income should be recovered, where rs are the regions, hhslds the household and t0 the benchmark period. Household income from the differences sources is defined as hinc and v\_corr are endogenous correction factors to minimize:

```

*
* --- recover given relative incomes
*
e_totInc(rs,t0) ..
    v_totInc(rs,t0)
    =E= sum(hhslds0 $ p_hhDataFao(rs,hhslds0,"income") , hinc(rs,hhslds0,t0));
e_hincRel(rs,hhslds,t0) $ p_hhDataFao(rs,hhslds,"income") ..
    hinc(rs,hhslds,t0)
    =E= v_totInc(rs,t0) * p_hhDataFao(rs,hhslds,"income") * v_corr(rs,hhslds,"income");

```

2. Total household income is defined from the given allocated factor income factyH and direct taxes dirTax on primary factors f, plus the endogenously determined transfer from government trng and transfer between households trnh. The variable trnh carries the delivering household on the second index and the receiving household on the third. We also consider remittances received remiH and a share on wages remoh which is sent abroad<sup>14</sup>:

---

<sup>14</sup> The data on remittances are available at bi-lateral level from the GTAP-MIG data base and define the total of remittances received and sent for each country resp. aggregate of countries.

```

*
* --- factor income plus government plus private transfers
*
e_hinc(rs,hhslds,t0) $ p_hhDataFao(rs,hhslds,"income") ..

    hinc(rs,hhslds,t0) =E= (factYH.1(rs,hhslds,t0)-sum(f,dirTax.1(rs,f,hhslds,t0)))
        + trng(rs,hhslds,t0)
        + remih(rs,hhslds,t0)
        - remoh.1(rs,hhslds,t0)
        + sum(hhslds0, trnh(rs,hhslds0,hhslds,t0))
        - sum(hhslds0, trnh(rs,hhslds,hhslds0,t0))
        ;
*

```

3. Income shares from public transfer should be recovered; a correction factor allows for deviations. Note that we don't have these data from the SAM. Instead, they are indirectly constructed by the balancing program.

```

*
* --- relative government transfers
*
e_trngShr(rs,hhslds,t0) $ p_hhDataFao(rs,hhslds,"income") ..

    trng(rs,hhslds,t0)
    =E= [p_hhDataFao(rs,hhslds,"shipubtransfer")+v_corr(rs,hhslds,"trng")]*(hinc(rs,hhslds,t0)+sum(hhslds0, trnh(rs,hhslds,hhslds0,t0)));

```

4. Transfer income from other households should be recovered, again allowing for deviations based on a correction factor:

```

*
* --- relative private transfers
*
e_trnhShr(rs,hhslds,t0) $ p_hhDataFao(rs,hhslds,"income") ..

    sum(hhslds0, trnh(rs,hhslds0,hhslds,t0))
    =E= [p_hhDataFao(rs,hhslds,"shiprivtransfer")+v_corr(rs,hhslds,"trnh")] * (hinc(rs,hhslds,t0)+sum(hhslds0, trnh(rs,hhslds,hhslds0,t0)));

```

5. Two equations ensure first that the given remittances are exhausted, and second, that estimated shares come close to given ones:

```

*
* --- remittances
*
e_remitBal(rs,t0) ..

    sum(hhslds $ p_hhDataFao(rs,hhslds,"income"),remih(rs,hhslds,t0)) =E= remih.1(rs,"hhsld",t0);

e_remitShr(rs,hhslds,t0) $ p_hhDataFao(rs,hhslds,"income") ..

    remih(rs,hhslds,t0)
    =E= [p_hhDataFao(rs,hhslds,"remittances")+v_corr(rs,hhslds,"remit")]*(hinc(rs,hhslds,t0)+sum(hhslds0, trnh(rs,hhslds,hhslds0,t0)));

```

The above equations define the constraints for a penalty function which minimizes squared differences for the four correction factors, given a high weight for recovering the relative income levels. We add terms which penalize transfer from poorer to richer households.

```

e_objeHHsld ..

v_objeHHsld * sum( (rs,hhslds,t0) $ p_hhDataFao(rs,hhslds,"income"),1)

=E= sum( (rs,hhslds,t0) $ p_hhDataFao(rs,hhslds,"income"),

    sqr( v_corr(rs,hhslds,"trnh"))
  + sqr( v_corr(rs,hhslds,"trng"))
  + sqr( v_corr(rs,hhslds,"remit"))
  + sqr( v_corr(rs,hhslds,"income")-1)

    --- transfers from poorer to richer households

  + sum(hhslds0
    $ ( ( p_hhDataFao(rs,hhslds,"income")/p_HHDataFao(rs,hhslds,"population"))
      < ( p_hhDataFao(rs,hhslds0,"income")/p_HHDataFao(rs,hhslds0,"population"))
      $ p_hhDataFao(rs,hhslds0,"income")) ,
    trnh(rs,hhslds,hhslds0,t0)/hinc(rs,hhslds,t0) )

  );

```

### Saving rates and foreign income contributions

The household surveys do not comprise data on saving rates. Unfortunately, due to the regional household concept, the GTAP data set reports only the total regional savings as the sum of savings (or additional debts) by private households, firms and public institutions. What is available from the SAM are total tax income  $yTaxTot$ , factor income after depreciation  $yFacty$ , direct taxes  $dirtax$  and the private  $yc$  and public consumption  $yg$  expenditures. Equally missing are data of transfers from the government to households  $trng$ . They are not needed for the regional household approach found in the GTAP Standard model as they net out if we add income sources of public and private agents. (The same holds for transfers between different households in the same country).

That means that we know for the government the usual main income source (tax income) and know what it spent for consumption ( $yg$ ), but miss data on savings or transfers to households:

$$\overline{yg} + trng + savg = \overline{yTaxTot} \quad (31)$$

Similar for the private aggregate household, we miss data on its savings (and again on transfers received from government):

$$\overline{yc} + savc = \overline{yFacty} - \overline{dirtax} + trng \quad (32)$$

What is known are total regional savings  $rsav$ :

$$\overline{rsav} = savc + savg \quad (33)$$

From the given balanced data found in the SAM, reflecting the equivalence of regional household income earned (RHS) and spent (LHS), it also holds that:

$$\overline{yg} + \overline{yc} + \overline{rsav} = \overline{yFacty} + \overline{yTaxTot} - \overline{dirtax} \quad (34)$$

These four accounting identities are linear dependent as seen easily from the fact that the government transfers  $trng$  can be set to zero, and  $savg$  and  $savc$  chosen to close the government (1) and private household budgets (2). Adding next (1) and (2) leads over (3) to the identity found in (4).

That implies that we need either additional data or use assumptions to define government transfers  $trng$  and the two savings positions. As governments seldom achieve a budget surplus, a rather simple rule sets  $savg$  to zero in case that total regional saving  $rsav$  is positive. That defines government transfers to households  $trng$  from (1). i.e. tax income is exactly exhausted by public consumption and transfers to households. Identity (32) then reflects that the households performs all regional savings as  $savg$  is zero by assumption. In case of overall negative savings, we set instead  $savc$  to zero (in opposite to government, it is not likely that the aggregate household has a structural deficit), and calculate from there the necessary transfers  $trng$  to close the household's budget. That implies that the total and under that case negative regional savings become in full additional debt of the government.

The balancing becomes only slightly more complex if we take remittances or other income exchanges with the rest-of-the world  $finc$  (for foreign income) into account:

$$\overline{yc} + savc = \overline{yFacty} - \overline{dirtax} + trng + finc \quad (35)$$

These income flows are part of the balance of payment, if introduced in the existing data base, the foreign savings  $fsav$  needs to be corrected by the same amount from the B.O.T. equals B.O.P. condition:

$$\bar{M} - \bar{X} = \text{savf} + \text{finc} \quad (36)$$

As the additional income  $\text{finc}$  in (35) on the RHS will increase savings  $\text{savc}$  of the household on the LHS by the same amount, we also maintain the identity of investments  $I$  and total saving  $S$ , the latter equal to the sum of foreign and regional savings:

$$\bar{I} \equiv \bar{S} = \text{savf} + \text{savc} + \text{savg} \quad (37)$$

Alternative assumptions on the distribution of total regional savings to government and private households are clearly possible as long as the accounting identities are maintained.

### 3.1.9 Post-model micro-simulation

Post-model micro-simulation became fashionable in the first decade of the 2000 (cf. Cockburn 2006<sup>15</sup>). It reflects the growing insights in the differentiated impacts of policy reform such as trade liberalization or structural adjustment programs on different household types. There are two major pathways how data of individual households are updated in that type of analysis: from the earning and from the spent side. From the earning side, policy reforms will lead to differentiated impacts on factor returns depending on the factor and the sector. Especially with sluggish factor markets, households can see their factor income develop quite differently depending in which sectors they are engaged, and, from which factors they draw income. Policy reforms can also directly affect factor returns e.g. by changing direct tax rates or factor taxes respectively subsidies faced by the different sectors. Many households depend also on private and public transfers. In the latter case, changes in tax income or how tax income is distributed can affect them. For private transfers, income changes of their donors typically also impact the size of transfers. Impacts from the spent side result from changes in consumer prices, from changes in domestic and foreign production costs, trade margins and the various taxes and subsidies which define consumer prices from production costs. These general considerations lead to a rather straightforward approach described next which reflects the state-of-the-art in that field.

The post-model micro-simulation builds technically on three steps which reflect the considerations above. First, changes in factor prices and factor quantities as well as tax and regional income from the CGE simulation are used to update the income position of each household in the FAO household data set. Second, we use the AIDADS demand system to estimate budget shares; results from the G-RDEM estimation of the saving rates allow to estimate saving rates specific to each household, and finally, we assign to each household a share of government consumption. That allows calculating the money metric for each household. The final step summarizes the results based on a regression of the money metric on income composition and by calculating percentiles of key results and household features, sorted by the money metric.

#### Updating household income

The FAO data set reports shares on total household income for different wage categories, for self-employed, crop and livestock production, public and private transfers and an additional position. We detail next how these different elements are used to update the reported benchmark income for each household.

---

<sup>15</sup> Cockburn, John (2016): "Trade liberalisation and poverty in Nepal: A computable general equilibrium micro-simulation analysis." *Globalisation and Poverty*. Routledge, 2006. 189-212



For wage income, we use changes in factor returns to the specific skill category and the individual GTAP sectors linked to the more aggregate categories in the FAO data set to estimate the change in wage income:

```
*
* --- income from different wage categories: link to change in labour remuneration
*
p_microRes(country_hhid(microR,hhid),wages,tSim)
= p_compact(microR,hhid,"totIncome") * p_compact(microR,hhid,wages)
  * sum( datvars_1_a(wages,l,a) $ xFlag(microR,l,a), xf.l(microR,l,a,tSim) *pf.l(microR,l,a,tSim) * (1-kappaf.l(microR,l,tSim)))
  / sum( datvars_1_a(wages,l,a) $ xFlag(microR,l,a), xf.l(microR,l,a,"%t0%") *pf.l(microR,l,a,"%t0%") * (1-kappaf.l(microR,l,"%t0%")));
```

For instance, for the second wage category linked to extraction, the following sectors are used:

```
set datvars_1_a(datVars,l,a) /

shlwge2_1.unskLab .(coa-a,oil-a,gas-a,omn-a)
shlwge2_2.skLab .(coa-a,oil-a,gas-a,omn-a)
shlwge2_3.unskLab .(coa-a,oil-a,gas-a,omn-a)
```

For self-employed and agriculture income from crop respectively livestock production, all factor returns are used (wages, interest, returns to natural resources and land) as it is probably rather unclear how especially wage income accruing to the household from a farm it manages is accounted for in the SAM and the household survey:

```
*
* --- income from different self-employed categories: link to change in factor remuneration
*
p_microRes(country_hhid(microR,hhid),self,tSim)
= p_compact(microR,hhid,"totIncome") * p_compact(microR,hhid,self)
  * sum( (datvars_self(self,a),f) $ xFlag(microR,f,a), xf.l(microR,f,a,tSim) *pf.l(microR,f,a,tSim) * (1-kappaf.l(microR,f,tSim)))
  / sum( (datvars_self(self,a),f) $ xFlag(microR,f,a), xf.l(microR,f,a,"%t0%") *pf.l(microR,f,a,"%t0%") * (1-kappaf.l(microR,f,"%t0%")));
```

The different aggregates (2-10) for which income from self-employment is reported in the FAO data (such as manufacturing) are mapped to the GTAP sectors:

```
set datvars_self(datVars,a) /
shlSelf2.(coa-a,oil-a,gas-a,omn-a)
shlSelf3.set.manuf
shlSelf4.set.eleDist
shlSelf5.cns-a
shlSelf6.trd-a
shlSelf7.set.trsStoCom
shlSelf8.set.FinInsReal
shlSelf9.set.services
//
datVars_self("shlSelf10",nonAgr) = YES;
```

Public transfers are updated with changes in total tax income. This assumes that the value share of tax income spent on transfers to each household type is unchanged:

```
*
* --- income from public transfer: tax income
*
p_microRes(country_hhid(microR,hhid),"pubTransfer",tSim)
= p_compact(microR,hhid,"totIncome") * p_compact(microR,hhid,"shlPubTransfer")
  * yTaxTot.l(microR,tSim)/yTaxTot.l(microR,"%t0%");
```

For private transfers, we assume that payments follow the aggregate private consumption changes:

```
*
* --- income from private transfer: private consumption
*
p_microRes(country_hhid(microR,hhid),"privTransfer",tSim)
= p_compact(microR,hhid,"totIncome") * p_compact(microR,hhid,"shlPrivTransfer")
  * yc.l(microR,"hhsld",tSim)/yc.l(microR,"hhsld","%t0%");
```

For the remaining positions under “other” which refer, for instance, to non-farm rental income and remittances (where the data set often fails to provide a detailed composition), we use changes in regional income:

```

*
* --- other income: regional income
*
p_microRes(country_hhid(microR,hhid),"other",tSim)
= p_compact(microR,hhid,"totIncome") * p_compact(microR,hhid,"shlOther")
  * regv.l(microR,tSim)/regv.l(microR,"%t0%");

```

After these changes, all income categories found in the FAO data set have been updated based on the CGE model results such that changes in income in real terms from the income generation side are simulated. The notion of real terms reflects that there is not inflation in the CGE model as only relative prices matter for simulated changes in quantities. The usual case in the GTAP model is to use a world factor price index as the numeraire.

### Estimating budget shares, their changes and the money metric

Income changes at inflation free factor prices alone are not sufficient to assess how policy reforms or other changes impact the economic situation of individual household. We need also to reflect which prices they face for the goods they are consuming.

The next step is therefore to estimate the impact of changes in consumer prices. The code is only active for the MAIDADS demand system which is especially suitable to capture impacts of changes in income levels on consumption pattern due to is rather non-linear Engels curves which have been econometrically estimated (Britz 2021<sup>16</sup>). In a first step, we calculate cardinal utility for each household from an econometrically estimated relation to per capita income:

```

p_microRes(country_hhid(microR,hhid),"yc",calcS) $ p_microRes(microR,hhid,"incPerCap",calcS)
= p_microRes(microR,hhid,"incPerCap",calcS) * (p_microRes(microR,hhid,"betap",calcS)+p_microRes(microR,hhid,"betag",calcS));

p_microRes(country_hhid(microR,hhid),"uh",calcS) $ p_microRes(microR,hhid,"yc",calcS)
= (p_estU("cnst") + p_estU("log") * log(p_microRes(microR,hhid,"yc",calcS)/pgdpmp.l(micror,calcS))
  + p_estU("logSqr") * sqrt(log(p_microRes(microR,hhid,"yc",calcS)/pgdpmp.l(micror,calcS)))
  + p_estU("logSqrt") * sqrt(log(p_microRes(microR,hhid,"yc",calcS)/pgdpmp.l(micror,calcS)))
  + sum(deltaDrivers, p_estu(deltaDrivers) * p_deltaShifters(micror,deltaDrivers));

```

From there, we define the driver of the exponential functions for the marginal budget shares and commitment terms:

```

p_microRes(country_hhid(microR,hhid),"uhExpD",calcS) $ p_microRes(microR,hhid,"incPerCap",calcS)
= p_parAIDADS("UD","cnst") + v_kappa(microR,"hhsld","UDC")
  + (v_kappa(microR,"hhsld","UDL") + p_parAIDADS("UD","lin")) * p_microRes(microR,hhid,"uh",calcS);

p_microRes(country_hhid(microR,hhid),"uhExpG",calcS) $ p_microRes(microR,hhid,"incPerCap",calcS)
= p_parAIDADS("UG","cnst") + v_kappa(microR,"hhsld","UGC")
  + (v_kappa(microR,"hhsld","UGL") + p_parAIDADS("UG","lin")) * p_microRes(microR,hhid,"uh",calcS);

```

From there, we define the marginal budget share as a linear combination of the marginal budget shares at very low income (= utility) level alphaAIDADS and the ones at very high income (= utility) level betaAIDADS;

```

p_microAlpha(country_hhid(microR,hhid),i,%2) $ p_microRes(microR,hhid,"yc",%2)
= [alphaAIDADS(microR,i,"hhsld")+betaAIDADS(microR,i,"hhsld") * sqExp(p_microRes(microR,hhid,"uhExpD",%2),10) ]
  / [ 1 + sqExp(p_microRes(microR,hhid,"uhExpD",%2),10) ];

p_microAlpha(country_hhid(microR,hhid),dNest,%2) $ p_microRes(microR,hhid,"yc",%2)
= [alphaAIDADS(microR,dNest,"hhsld")+betaAIDADS(microR,dNest,"hhsld")*sqExp(p_microRes(microR,hhid,"uhExpD",%2),10) ]
  / [ 1 + sqExp(p_microRes(microR,hhid,"uhExpD",%2),10) ];

```

<sup>16</sup> Britz, W. (2021): Estimating a global MAIDADS demand system considering demography, climate and norms, *Bio-based and Applied Economics*, in: online available 19 July 2021

And similar for the commitment terms:

```
p_microGamma(country_hhid(microR,hhid),i,%2) $ p_microRes(microR,hhid,"yc",%2)
= [gammaLo(microR,i,"hhsld")+gammaHi(microR,i,"hhsld") * sqExp(p_microRes(microR,hhid,"uhExpG",%2),10) ]
/[ 1 + sqExp(p_microRes(microR,hhid,"uhExpG",%2),10) ];

p_microGamma(country_hhid(microR,hhid),dNest,%2) $ p_microRes(microR,hhid,"yc",%2)
= [gammaLo(microR,dNest,"hhsld")+gammaHi(microR,dNest,"hhsld")*sqExp(p_microRes(microR,hhid,"uhExpG",%2),10)]
/[ 1 + sqExp(p_microRes(microR,hhid,"uhExpG",%2),10) ];
```

Given the absolute term of the AIDADS demand system gammaLES, we can define the non-committed income, the i.e. the income distributed by the marginal budget shares defined above<sup>17</sup>:

```
p_microRes(country_hhid(microR,hhid),"nonCommit",%2) $ p_microRes(microR,hhid,"yc",%2)
= max(p_microRes(microR,hhid,"yc",%2)*0.25,p_microRes(microR,hhid,"yc",%2)
- sum(i, p_microGamma(microR,hhid,i,%2)*pa.1(microR,i,"hhsld",%2) )
- sum(dNest, p_microGamma(microR,hhid,dNest,%2)*pdNest.1(microR,dNest,"hhsld",%2)));
```

The non-committed income in combination with the marginal budget shares allow to define the consumer price index for each household:

```
*
* --- consumer price for individual households
*
p_microRes(country_hhid(microR,hhid),"pCons",%2) $ p_microRes(microR,hhid,"incPerCap",%2)
*
* --- Armington demand times price, divided by income => summed defines the hhsld specific consumer price
*
= [ sum(i,
pa.1(microR,i,"hhsld",%2) * (p_microGamma(microR,hhid,i,%2)
+ p_microAlpha(microR,hhid,i,%2)/pa.1(microR,i,"hhsld",%2)*p_microRes(microR,hhid,"nonCommit",%2))
+ sum(dNest,
pdNest.1(microR,dNest,"hhsld",%2) * (p_microGamma(microR,hhid,dNest,%2)
+ p_microAlpha(microR,hhid,dNest,%2)/pdNest.1(microR,dNest,"hhsld",%2)*p_microRes(microR,hhid,"nonCommit",%2))
)
]/(p_microRes(microR,hhid,"incPerCap",%2)*p_microRes(microR,hhid,"betap",%2));
```

The contribution to utility from government consumption is based on the simple assumption that each citizen consumes the same share on the total amount:

```
*
* --- distribute government consumption on a per capita basis
*
p_microRes(country_hhid(microR,hhid),"yg",tSim) $ p_microRes(microR,hhid,"incPerCap",tSim)
= yg.1(microR,tSim)*%scale%*1000. / pop.1(microR,"hhsld",tSim);
```

From this assumption follows that the utility of poorer household depends to a larger share on the development of government consumption compared to richer ones:

```
*
* --- calculate implicate share of government consumption on regional income per capita
*
p_microRes(country_hhid(microR,hhid),"betag",tSim) $ p_microRes(microR,hhid,"incPerCap",tSim)
= p_microRes(microR,hhid,"yg",tSim)*(1-p_microRes(microR,hhid,"betas",tSim))
/ (p_microRes(microR,hhid,"incPerCap",tSim)+p_microRes(microR,hhid,"yg",tSim));
```

For savings, we first estimate saving rates depending on per-capita income, using the regression coefficient from the G-RDEM model (Britz and Roson 2019<sup>18</sup>). We cut-off estimates above 50% as the largest macro-saving rate found in any country in the underlying estimation was around 40% while per capita income levels at household level found in the samples easily exceed even the highest mean per capita income found for any country in the GTAP data set used in the econometric analysis. Note also that the CD-utility function does not allow for negative cost shares. We hence exclude cases where the household spends more than he earns (it is not clear if credits fall under other income):

<sup>17</sup> The actual code also considers demand nests.

<sup>18</sup> Britz, W., Roson, R. (2019): "G-RDEM: A GTAP-Based Recursive Dynamic CGE Model for Long-Term Baseline Generation and Analysis", *Journal of Global Economic Analysis* 4(1): 50-96

```

parameter p_avgInc(microR,t);
p_avgInc(microR,tSim) = sum(country_hhid(microR,hhid), p_compact(microR,hhid,"weight"));
p_avgInc(microR,tSim)
= sum(country_hhid(microR,hhid), p_compact(microR,hhid,"weight")*p_microRes(microR,hhid,"incPerCap",tSim))/p_avgInc(microR,tSim);

p_microRes(country_hhid(microR,hhid),"betas",tSim) $ p_microRes(microR,hhid,"incPerCap",tSim)
= min(0.5,max(1.E-3,betas.l(microR,tSim)) + (p_microRes(microR,hhid,"incPerCap",tSim) - p_avgInc(microR,tSim))*0.0010213);

```

With shares of (implicit) income spent to private (betap) and public (betag) consumption and savings (betas) defined, the utility level for each household can be calculated from the indirect utility function of the assumed top-level CD demand system:

```

*
* --- use indirect utility function to derive utility from income and arguments of CD-utility function
*
p_microRes(country_hhid(microR,hhid),"u",tSim) $ p_microRes(microR,hhid,"incPerCap",tSim)
= (p_microRes(microR,hhid,"incPerCap",tSim)+p_microRes(microR,hhid,"yg",tSim))/(1-p_microRes(microR,hhid,"betas",tSim))
* [
  (1/p_microRes(microR,hhid,"betas",tSim))*p_microRes(microR,hhid,"betas",tSim)
  * (1/p_microRes(microR,hhid,"betag",tSim))*p_microRes(microR,hhid,"betag",tSim)
  * (1/p_microRes(microR,hhid,"betap",tSim))*p_microRes(microR,hhid,"betap",tSim)
]
* ( 1 / p_microRes(microR,hhid,"pCons",tSim) ) ** p_microRes(microR,hhid,"betap",tSim)
* ( 1 / pg.l(microR,tSim) ) ** p_microRes(microR,hhid,"betag",tSim)
* ( 1 / pi.l(microR,tSim) ) ** p_microRes(microR,hhid,"betas",tSim);

```

Given the utility levels from the indirect CD utility function, we can calculate the money metric as the ultimate measure of changes in the households' purchasing power. The money metric reflects both changes in income (factor income, transfers, other) and the different price indices, considering differences in marginal budget shares across per capita income levels, saving rates and in the importance of government consumption in total household consumption:

```

*
* --- money metric from betas / saving prices and (1-betas) / pcons
*
$batinclude 'util/ttitle.gms' '"Calculate money metric for post-model micro-simulation %dataset%, %shock_withoutPath%"'

p_microRes(country_hhid(microR,hhid),"au",tSim)
= p_microRes(microR,hhid,"incPerCap",tSim) / [ p_microRes(microR,hhid,"u",tSim)
  * ( 1 / (1-max(1.E-3,betas.l(microR,tSim))) ) ** (1-max(1.E-3,betas.l(microR,tSim)))
  * ( 1 / max(1.E-3,betas.l(microR,tSim)) ) ** max(1.E-3,betas.l(microR,tSim)) ];

p_microRes(country_hhid(microR,hhid),"mm",tSim)
= ( p_microRes(microR,hhid,"u",tSim) - p_microRes(microR,hhid,"u",tSim) ) * p_microRes(microR,hhid,"au",tSim)
  * ( 1 / p_microRes(microR,hhid,"pCons",tSim) ) ** (1-max(1.E-3,betas.l(microR,tSim)))
  * (pi.l(microR,tSim)/pi.l(microR,tSim)) ** max(1.E-3,betas.l(microR,tSim));

```

We consider the money metric as the most suitable indicator to assess the well-being of household which can be calculated from the data set.

## Estimating individual demands and nutrient intakes

Given household specific marginal budget shares and commitment terms, the Marshallian demands can be defined, taking potential CES-subnests into account:

```

*
* (1) Directly driven by top-level demand system
*
p_microDem(country_hhid(microR,hhid),i,%2) $ p_hasXa(microR,i,"hhld")
= p_microGamma(microR,hhid,i,%2) + p_microAlpha(microR,hhid,i,%2)/pa.l(microR,i,"hhld",%2)
  * p_microRes(microR,hhid,"nonCommit",%2);

*
* (2) Demand nests driven by top-level demand system
*
p_microDem(country_hhid(microR,hhid),dNest,%2) $ sum(dNest_n_fd("top",dNest,h), p_hasXA(microR,dNest,"hhld"))
= p_microGamma(microR,hhid,dNest,%2) + p_microAlpha(microR,hhid,dNest,%2)/pdNest.l(microR,dNest,"hhld",%2)
  * p_microRes(microR,hhid,"nonCommit",%2);

*
* (2) In other cases, the homothecity of the CES sub-nesting implies that the demand (at equal prices across households!)
* depend solely on the demand for the nests in the top level demand function
*
p_microDem(country_hhid(microR,hhid),i,%2) $ (xaFlag(microR,i,"hhld") and (not p_hasXa(microR,i,"hhld")))
= xa.l(microR,i,"hhld",%2)
  * sum(topNest(dNest,i,"hhld"), p_microDem(microR,hhid,dNest,%2)/xdNest.l(microR,dNest,"hhld",%2));

```

Given the estimated nutrient contents for the products, nutrient intake per capita and day for calories, protein and fat can be estimated:

```
*
* --- calculate nutrient uptake
*
p_microRes(country_hhid(microR,hhid),nut,%2) $ p_microRes(microR,hhid,"incPerCap",%2)
= sum(i $ p_nutContFabio(microR,i,nut) , [ p_microDem(microR,hhid,i,%2)*0.001
+ ( sum(gov $ xaFlag(microR,i,gov),xa.1(microR,i,gov,%2))
+ sum(inv $ xaFlag(microR,i,inv),xa.1(microR,i,inv,%2))) /pop(microR,"hhsId",%2)]
* p_nutContFabio(microR,i,nut));
```

## Analysing the micro-simulation results

The samples for the different countries cover each several thousand observations such that it makes limited sense to analyse outcomes for each household separately. Instead, we first summarize findings in percentiles. To do so, we sort the households by their money metric, i.e. their changes in per capita purchasing power, and report for each quantile average earning shares and other results. We need to use the weights reported for each household. As GAMS is not designed for statistical analysis, especially reflecting the weights leads to somewhat clumsy code. We still consider it advantageous to stick to GAMS instead of moving data between different software packages.

As a first step, we use the GDXRank utility to define a sort index. As GDXRank works with a vector, we have to copy the money metric results for each household from the multi-dimensional parameter p\_microRes to a temporary one-dimensional parameter p\_temp. The permutation index returned p\_index is used to define lags and leads in order to generate a sorted copy of the results of interest, as represented by the set res.

```
p_temp(curHhid) = p_microRes(microR,curhhid,"mm",tSim);
execute_unload "rank_in.gdx", p_temp;
execute 'gdxrank rank_in rank_out.gdx > %system.nullfile%';
execute_load "rank_out.gdx", p_index = p_temp;
p_index(curHhid) = p_index(curHhid)-ord(curhhid);
p_Sorted(curhhid + p_index(curhhid),res) = p_microRes(microR,curhhid,res,tSim);
```

Next, we sum up the weights, and set the threshold for the first percentile as 1% of the total weights:

```
sumOfWeights = sum(curHhid, p_compact(microR,curHhid,"weight"));
p_thresHold = sumOfWeights/100;
count = 1;
```

In a loop over all households in the country, we add up the weights over all households p\_curSumW as well as the weights p\_curSumWQ and variables of interest multiplied with their weights p\_curSumQ for the next quantile to calculate. If the sum of all weights exceeds the next threshold, we calculate the averages for the result variables by dividing by the sum of weights in the current quantile. The results are collected in the p\_quantiles parameter. In order to prepare for the next quantile calculation, we reset the sum for the next quantile and update the threshold.

```
p_curSumW = 0;
p_curSumWQ = 0;
p_curSumQ(res) = 0;
loop(curHhid,
  p_curSumW = p_curSumW + p_sorted(curHhid,"weight");
  p_curSumWQ = p_curSumWQ + p_sorted(curHhid,"weight");
  p_curSumQ(res) = p_curSumQ(res) + p_sorted(curHhid,res)*p_sorted(curHhid,"weight");
  if ( p_curSumW ge p_thresHold,
    p_quantiles(microR,perc,res) $ (perc.pos eq count) = p_curSumQ(res) / p_curSumWQ;
    p_thresHold = p_thresHold + sumOfWeights/100;
    count = count + 1;
    p_curSumQ(res) = 0;
    p_curSumWQ = 0;
  );
);
```

One might now guess from the income shares etc. reported for the different quantiles where differences stem from. In order to improve here, we estimate an OLS model which use per capita income and the different shares as explanatory variables. The regression coefficients with high p-values can help to interpret the results. To do so, we use the LS solver from GAMS (or embedded Python for GAMS version 34 or later, not shown here):

```
set expV(datVars);expV(addInc) = YES; expV("incPerCap") = YES;
variable v_ssqr,v_cnst,v_par;
equation e_sumsqr,e_fit;
*
* --- simple linear model which estimates the money metric
*      from the income shares and total income
*
e_sumsqr..  v_ssqr =n= 0;
e_fit(country_hhid(curMicroR,curHhid),ts)..

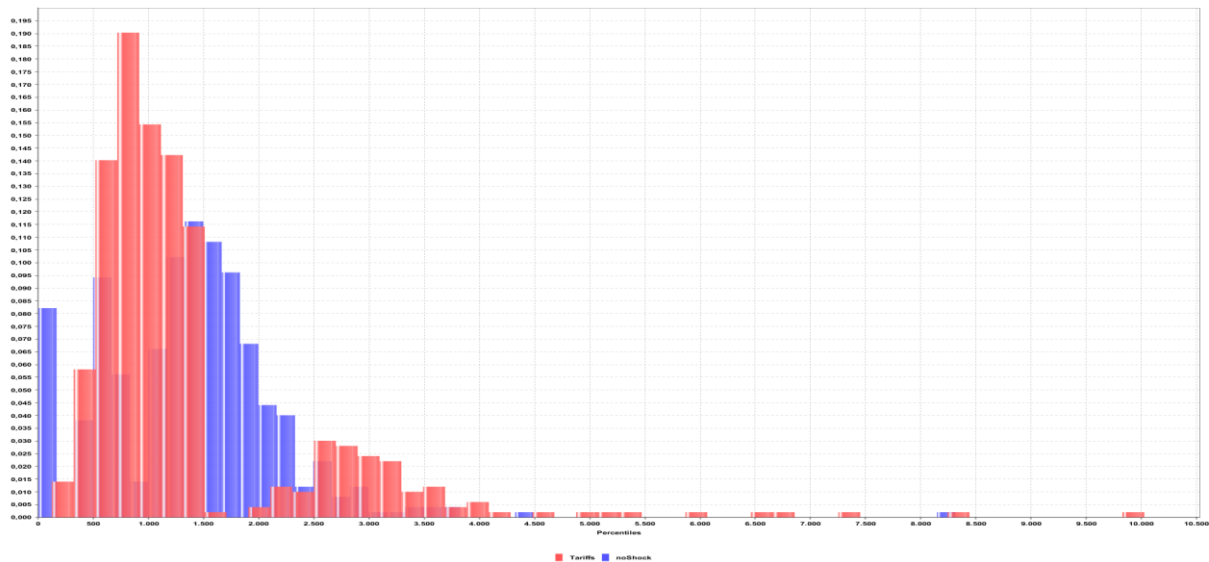
p_microRes(curMicroR,curHhid,"mm",ts) =e= v_cnst + sum(curExpV, v_par(curExpV) * p_compact(curMicroR,curHhid,curExpV));
```

Processing the results is somewhat cumbersome as the parameter and test statistics are not correctly labelled:

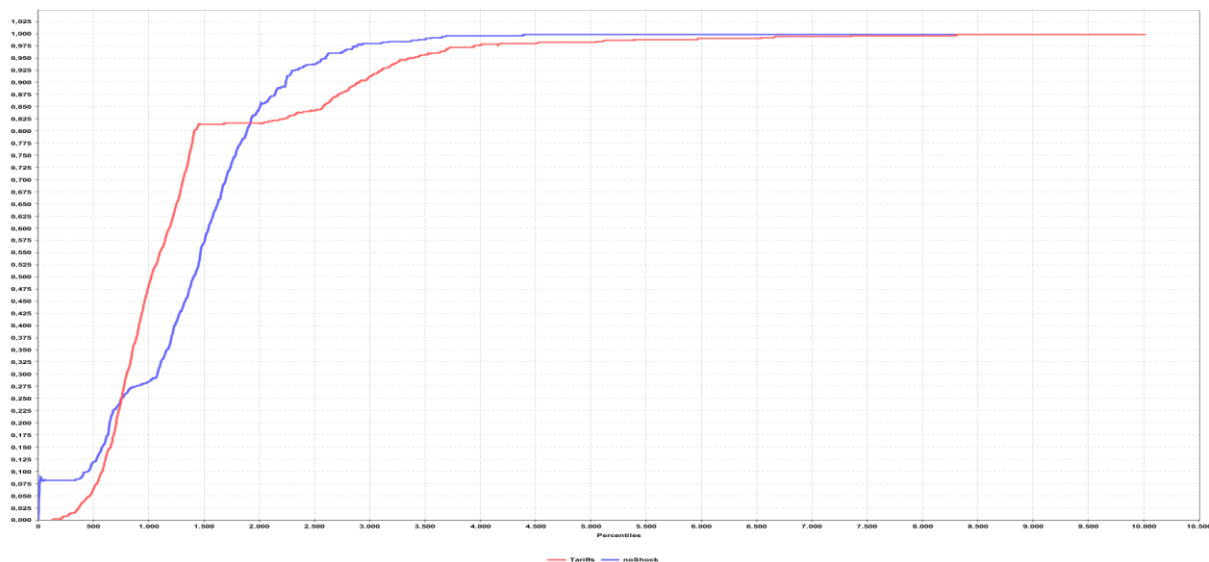
```
solve m_leastsq using lp minimizing v_ssqr;
execute_load 'ls', se, sigma, r2, df, tval, pval;
if ( card(tVal) eq card(curExpV)+1,
    p_quantiles(microR,"par",curExpV) = v_par(curExpV);
    p_quantiles(microR,"p-Val",curExpV) = sum(xls $ (xls.pos+1 eq curExpV.pos),pval(xls));
    p_quantiles(microR,"t-Val",curExpV) = sum(xls $ (xls.pos+1 eq curExpV.pos),tval(xls));
    p_quantiles(microR,"se",curExpV) = se(curExpV);
    p_quantiles(microR,"R2","") = r2;
    p_quantiles(microR,"df","") = df;
);
```

The results can be inspected with the interface, we offer:

1. Histograms which show the distribution of the various items (either for aggregated groups or rather details), for instance:



2. The same information, in cumulative format:



Both can also be shown as tables.

3. Finally, we show results for the OLS estimation:

Microsimulation results, OLS [0]				
Regions		Years		
Vietnam		shock		
Tariffs				
par		R2	df	pa
Money metric [cnst \$ per capita]			0.98	43364.00
Crop production [cnst \$ per capita]	0.12			
Livestock production [cnst \$ per capita]	0.11			
Public transfers [cnst \$ per capita]	-0.03			
Private transfers [cnst \$ per capita]	0.10			
Wages, agriculture, unskilled [cnst \$ per capita]	0.14			
Wages, agriculture, skilled [cnst \$ per capita]	0.15			
Wages, agriculture, other [cnst \$ per capita]	0.11			
Wages, extraction, unskilled [cnst \$ per capita]	0.11			



### 3.2 A GMIG module for CGEBox

Wolfgang Britz, November 2019

#### 3.2.1 Background and motivation

The GMIG data base (cf. Walsmley and et al. 2011<sup>19</sup>) reports international bi-lateral labour and population migration along with wage differences. Specifically, it extends the GTAP standard data set with data on persons employed of a certain nationality working in other countries and the wages they receive at market, buyers' and sellers' prices. It also adds data on bi-lateral remittances. One of the interesting features of the GMIG data base is that reports persons such that wages per capita can be defined. The data base was developed to source the GMIG model which simulates changes in these variables. The GMIG model otherwise is a rather straightforward extension of the GTAP standard model. Extending CGEBox with that module, especially for long-run analysis, seems therefore inviting.

GMIG considers that the labour stock and population of a country consists of nationals and migrants. The GMIG module in CGEBox assumes that an aggregate agent in each home country manages the labour stocks by skill category of nationals and distributes them to the home country and abroad according to a Constant-Elasticity-of-Transformation function such that revenues from wages after direct taxes and corrected for the consumer price index are maximized. Migrant labour leaves its home country along with some dependents and updates the population at home and abroad. Migrants send a share of the labour income as remittances back to the country they originate. Tax rates paid by firms and direct taxes paid by employees differ by country of origin of the migrants (respectively the home country).

#### 3.2.2 Equations

The key equation in the GMIG model links the number of workers of skill category  $l$  from region  $rNat$  who work in another region  $rNat1$  to differences in wages after direct taxes  $kappafMIG$  based on a CET function, i.e. it assumes that labour migration reflects a revenue maximization problem. To do so, we define purchasing power corrected wages  $v\_wagePP$  of workers from  $rNat$  received in  $rNat1$ , using the consumer price index  $pcons$  to reflect changes in purchasing power:

$$v\_wagePP_{rnat,rnat1,l} = pft_{rnat1,l}(1 - v\_kappafMig_{rnat,rnat1,l})/pcons_{rnat1} \quad (38)$$

```
e_wagePP(rNat,rnat1,l,ts(t)) $ p_xftMIG(rNat,rNat1,l,t) ..
v_wagePP(rnat,rnat1,l,t) =E= pft(rnat1,l,t) * (1 - v_kappafMig(rnat,rnat1,l,t))/pcons(rnat1,"hhsld",t);
```

For all equations, the diagonal elements  $rNat,rNat$  are defined as well, they denote, for instance, the workers who are employed in their home country. The average wage  $v\_wagePPA$  received for workers from  $rNat$  is defined from the dual price index at a transformation elasticity of  $\omega_{XftMig}$  and share parameters  $g_{XftMig}$ :

$$v\_wagePPA_{rnat,l} = \left[ \sum_{rnat} g_{Mig_{rnat,rnat1,l}} v\_wagePP_{rnat,rnat1,l}^{(1+\omega_{XftMig})} \right]^{1/(1+\omega_{XftMig})} \quad (39)$$

<sup>19</sup> Walmsley, Terrie L., Alan Winters, and Amer Ahmed. "The impact of the movement of labour: results from a model of bilateral migration flows." *Global Economy Journal* 11.4 (2011): 1850240

```
e_wagePPA(rNat,l,ts(t)) ..
v_wagePPA(rnat,l,t) =E= sum(rNat1 $ p_xftMIG(rNat,rNat1,l,t),
p_gMig(rnat,rnat1,l,t) * v_wagePP(rnat,rnat1,l,t)**(1+p_omegaMig(rnat,l))) ** (1/(1+p_omegaMig(rnat,l))));
```

The CET distribution determines the number of workers  $v\_xMig$ , measured in constant dollars at market prices, from  $rNat$  working in  $rNat1$  by distributing the given labour stock  $p\_xftMig$  of the originator country:

$$v\_XFmig_{rnat,rnat1,l} = g_{XFmig_{rnat,rnat1,l}} p\_xftMig_{rnat1,l} \left( \frac{v\_wagePP_{rnat,rnat1,l}}{v\_wagePPA_{rnat,l}} \right)^{\omega_{XFmig}} \quad (40)$$

```
e_labDist(rNat,rnat1,l,ts(t)) $ p_xftMIG(rNat,rNat1,l,t) ..
v_xfMig(rnat,rnat1,l,t)/v_xfMig.scale(rnat,rnat1,l,t)
=E= p_xftMIG(rNat,"tot",l,t)/v_xfMig.scale(rnat,rnat1,l,t)
* p_gMig(rnat,rnat1,l,t) * (v_wagePP(rnat,rnat1,l,t)/v_wagePPA(rnat,l,t))**p_omegaMig(rnat,l);
```

Adding up over the workers from different regions  $rnat1$  currently working in  $rnat$  defines the labour stock in  $rnat$  and drives the prices for labour  $pft$ :

$$XFT_{rnat,l} = \sum_{rnat1} v\_xfMig_{rnat,rnat1,l} \quad (41)$$

```
e_xftMig(rNat,l,ts(t)) $ (rsNat(rnat) $ (xft.range(rnat,l,t) ne 0)) ..
xft(rnat,l,t)/xft.scale(rnat,l,t) =E= sum(rnat1 $ p_xftMIG(rNat1,rNat,l,t), v_xfMig(rnat1,rnat,l,t))/xft.scale(rnat,l,t);
```

As the data base reports bi-lateral tax rates from a buyers'  $fcttx$  and sellers'  $kappaf$  perspective, we need to derive average rates applied in a country to the labour stock by skill category  $l$ :

$$XFT_{rnat,l} pft_{rnat,l} kappaf_{rnat,l} = \sum_{rnat1} v\_xfMig_{rnat,rnat1,l} pft_{rnat1,l} kappafMig_{rnat,rnat1,l} \quad (42)$$

```
e_kappaMig(rNat,l,ts(t)) $ rsNat(rnat) ..
kappaf(rnat,l,t) * xft(rnat,l,t) /xft.scale(rnat,l,t)
=E= sum(rnat1 $ p_xftMIG(rNat1,rNat,l,t), v_xfMig(rnat1,rnat,l,t) * v_kappaFMig(rnat1,rnat,l,t))/xft.scale(rnat,l,t) ;
```

In case of factor taxes, the update process is somewhat more complex as factor taxes are defined specific per activity in the GTAP standard model whereas the GMIG data base seems to report only average differences across activities:

$$XFT_{rnat,l,a} pft_{rnat,l} fcttx_{rnat,l,a} = \sum_{rnat1} v\_xfMig_{rnat,rnat1,l} pft_{rnat1,l} (fcttx_{rnat1,l,a}^0 + v\_fcttxMig_{rnat,rnat1,l} - v\_fcttxMig_{rnat,rnat1,l}^0) \quad (43)$$

```
e_fcttxMig(rNat,l,a,ts(t)) $ rsNat(rnat) ..
fcttx(rnat,l,a,t) * xft(rnat,l,t)/xft.scale(rnat,l,t)
=E= sum(rnat1 $ p_xftMIG(rNat1,rNat,l,t), v_xfMig(rnat1,rnat,l,t)
* (fcttx.l(rnat,l,a,"%t0%")+v_fcttxMig(rnat,rnat1,l,t)-v_fcttxMig.l(rnat,rnat1,l,"%t0%")))/xft.scale(rnat,l,t) ;
```

In both cases, the model equations drop the price weights  $pft$  which is found on both sides. They are shown in the equations above to underline that we use a value weighted average which fits the ad-valorem tax rate definitions used in the model.

Finally, we assume that the average relation between migrated people  $popMig$  and workers stays identical, using the head count  $p\_labMig$  by skill category reported in the data:

$$\frac{\text{popMig}_{\text{rnat},\text{rnat1}}}{\text{popMig}_{\text{rnat},\text{rnat1}}^0} = \frac{\sum_l \text{p\_labMig}_{\text{rnat},\text{rnat1},l} \frac{\text{v\_xfMig}_{\text{rnat},\text{rnat1},l}}{\text{v\_xfMig}_{\text{rnat},\text{rnat1},l}^0}}{\sum_l \text{p\_labMig}_{\text{rnat},\text{rnat1},l}} \quad (44)$$

```

e_popMig(rNat,rnat1,ts(t)) $ p_popMIG(rNat,rNat1) ..
v_popMig(rnat,rnat1,t)/v_popMig.scale(rnat,rnat1,t)
=E= sum(l $ p_xftMIG0(rNat,rNat1,l),
*
--- change in labour stock [measured in cnst dollar at market prices], per skill category
*
v_xfMig(rnat,rnat1,l,t)/p_xftMig0(rnat,rnat1,l)
*
--- multiplied with # of workers = change in labour force in #
*
* p_labMig(rnat,rnat1,l))
/
---- divided by # of workers at benchmark, multiplied with population in #
sum(l, p_labMig(rnat,rnat1,l)) * p_popMig(rnat,rnat1)/v_popMig.scale(rnat,rnat1,t);

```

From there, the current population in a country pop is defined

$$\text{pop}_{\text{rnat},\text{hhsld}} = \sum_{\text{rnat1}} \text{popMig}_{\text{rnat1},\text{rnat}} \quad (45)$$

```

e_popTot(rs(r),h,ts(t)) $ hFlag(r,h) ..
pop(r,h,t)/p_pop(r,h,t) /pop.scale(r,h,t)
=E= sum((r_r(r,rnat),rnat1) $ p_popMig(rnat1,rnat), v_popMig(rnat1,rNat,t)/p_pop(rnat,"hhsld",t))/pop.scale(r,h,t);

```

The remittances  $v\_remit$  are defined relative to base year level:

$$\text{v\_remit}_{\text{rnat},\text{rnat1}} = \sum_l \frac{\text{v\_xfMig}_{\text{rnat},\text{rnat1},l}}{\text{v\_xfMig}_{\text{rnat},\text{rnat1},l}^0} \text{p\_remit}_{\text{rnat},\text{rnat1},l} \quad (46)$$

```

e_remit(rnat,rnat1,ts(t)) $ p_popMIG(rNat,rNat1) ..
v_remit(rnat,rnat1,t)/v_remit.scale(rnat,rnat1,t)
=E= sum(l $ p_xftMig(rnat,rnat1,l,t),v_xfMig(rnat,rnat1,l,t)*pft(rnat1,l,t)/lcu(rNat1,t)
*p_remit(rnat,rnat1,l)/p_xftMig0(rnat,rnat1,l))/v_remit.scale(rnat,rnat1,t);

```

As the remittances are part of the original B.O.P. which is equal to B.O.T., we would need some re-bookings to correct regional income and foreign savings to introduce them already at the benchmark where they must be part of the reported data set. We therefore introduce an absolute change in the remittances balance  $\text{finc}$  (= foreign income), a position which is already part of the myGTAP module:

$$\begin{aligned} & V_{\text{finc},\text{rnat}} \\ &= \sum_{\text{rnat1}} V_{\text{remit},\text{rnat},\text{rnat1}} - V_{\text{remit},\text{rnat1},\text{rnat}} - v_{\text{remit},\text{rnat},\text{rnat1}}^0 + v_{\text{remit},\text{rnat1},\text{rnat}}^0 \end{aligned} \quad (47)$$

```

e_fincGMIG(rnat,ts(t)) $ (finc.range(rnat,t) ne 0) ..
finc(rnat,t)/regy.scale(rnat,t)
=E= [ sum(rnat1, v_remit(rnat,rnat1,t) $ p_popMIG(rNat,rNat1) - v_remit(rnat1,rnat,t) $ p_popMIG(rNat1,rNat))
-sum(rnat1, v_remit.l(rnat,rnat1,"%t0%") $ p_popMIG(rNat,rNat1) - v_remit.l(rnat1,rnat,"%t0%") $ p_popMIG(rNat1,rNat))
]/regy.scale(rnat,t);

```

The change enters the B.O.P. = B.O.T. identity and the regional income definitions.

### 3.2.3 Technical aspects

There are number of points to note with regard to the interaction with other modules in CGEBox

1. The GTAP standard model core has a supply equation for total factor supply  $e\_xft$ . This equation is switched off for labour categories if the GMIG module is active as (41) above defines the labour stock.
2. The myGTAP module has a simplified equation to determine remittances from total labour use in the countries which is replaced by equation (46) above.
3. Some standard closures are changed. So are population and direct taxes on labour fixed by default in the GTAP standard model and become endogenous if the GMIG module is switched on.

The GMIG module is fully compatible with the other modules of CGEBox and had been tested in various combinations.

### 3.2.4 Result exploitation

The post-model processing adds a table which shows the bi-lateral population and labour force distribution and report the wages receives, at market prices and after direct taxation.

Bil-lateral migration [0]

Items

Quantity

▼

Products

pop

▼

Years

shock

Tariffs

Oceania

EastAsia

SEAsia

SouthAsia

NAmerica

LatinAmerI

EU\_28I

MENAI

SSAI

RestofWorldI

Oceania

World

32.530.020.030.010.330.010.310.000.020.04

Australia, New Zealand

0.411579.231.010.043.740.200.890.010.070.08

East Asia

0.600.42600.110.103.730.021.050.040.030.90

Southeast Asia

0.220.060.431628.231.980.021.930.150.155.56

South Asia

0.130.120.070.02435.520.500.870.060.080.28

North America

0.090.330.040.019.32480.142.130.010.250.22

Latin America

2.310.100.120.047.081.35486.270.220.814.86

European Union 28

0.160.010.030.010.710.072.60249.820.163.86

Middle East and North Africa

0.190.030.030.070.930.072.230.06877.210.61

Sub-Saharan Africa

0.200.040.080.052.010.119.431.480.09505.09

Rest of World

## 3.3 Sub-national dis-aggregation of production and factor markets in CGEBox

### 3.3.1 Background and motivation

Especially questions around the use of immobile natural resources such as land, minerals or water often ask for an analysis at sub-national level, especially if environmental consequences are to be evaluated and externalities have a regional character. It is therefore not uncommon to find sub-national detail in CGE analysis at single country level. We present in here a rather basic approach to add such detail to the GTAP model. For Europe, ready-to-use data at the level of NUTS2 regions are automatically added during the data preparation step if single European countries are present in the data set. For other countries or model regions, the user must supply these data to employ the module. The interface allows selecting during simulation for which countries the module is active.

### 3.3.2 Methodology

We implement a rather basic dis-aggregation as a starting point for further work:

1. No intra-national trade margins are introduced. There is one uniform price for the domestic origin faced by intermediate demand in all sub-regions of the model region.
2. Commodity demand is accordingly modeled at national level.

3. Factor markets however are depicted by a CET structure which hence allows for sluggish or even no factor mobility across regions.
4. The tax system is not regionally differentiated.

As a consequence, the data requirements are relatively low. Basically, data on output quantities and cost shares net of taxes are required for the framework.

The myGTAP module can change the assumption above and introduce sub-national final demand with preferences for products produced in the same sub-national unit.

### 3.3.3 Integration into the modeling framework

There were limited changes necessary to introduce regional detail for the supply side into the model and there were mostly realized via macros. In order to introduce the uniform output prices in the framework the following macro is used:

```
$iftheni.subregs declared subregs
  $$macro mm_px(r,a,t) sum(r_r(r,rp),m_px(rp,a,t))
$else.subregs
  $$macro mm_px(r,a,t) m_px(r,a,t)
$endif.subregs
```

Where the cross-set  $r\_r(r, rp)$  depicts the relation between sub-region  $r$  and nation  $rp$ . In case of countries without regional detail, the cross-set depicts a diagonal relation, i.e.  $r\_r(rNat, rNat) = \text{yes}$ .

In order to avoid problems with a simply linear aggregation from regions to nations, a CES-aggregator is used in the code:

```

* --- Equations active for sub-regional units, not part of GTAP V7 model
*
* (a) CES price aggregator from sub-regional offer prices to national average
*
e_xsNat(disr,iIn(i),ts) $ ( sum(a,xFlag(disr,a,i)) $ diag(i) $ (xs.range(disr,i,ts) ne 0)) ..

    ps(disr,i,ts) =E= sum(r_r(subr,disr) $ (p_gfi(subr,i,disr,ts) gt eps),
        p_gfi(subr,i,disr,ts)*ps(subr,i,ts)**(1-p_sigmaSubr(disr,i))**(1/(1-p_sigmaSubr(disr,i))));

*
* (b) distribution of supply
*
e_psSubr(subr,iIn(i),ts) $ ( sum(a,xFlag(subr,a,i)) $ (xs.range(subr,i,ts) ne 0) $ diag(i)) ..

    (xs(subr,i,ts)
        $$iftheni.hsabr %hsabr%==on
            sum(h $ (p_alphad(subr,i,h,ts) gt eps), xd(subr,i,h,ts))
        $$endif.hsabr
    )/xs.scale(subr,i,ts)
    =E= sum(r_r(subr,disr) $ p_gfi(subr,i,disr,ts),
        xs(disr,i,ts) * p_gfi(subr,i,disr,ts) * (ps(disr,i,ts)/ps(subr,i,ts))**p_sigmaSubr(disr,i))/xs.scale(subr,i,ts);

```

In case of a non-diagonal make, the following equations are used:

```

*
*      (c) same as (a), above in case of non-diagonal activity: average costs of the activity
*
e_xpNat(disr,aIn(a),ts) $ ( xpFlag(disr,a) $ (not diag(a)) ) ..

px(disr,a,ts) =E= sum(r_r(subr,disr) $ p_gfa(subr,a,disr,ts),
  p_gfa(subr,a,disr,ts)*px(subr,a,ts)**(1-p_sigmaASubr(disr,a)))*(1/(1-p_sigmaASubr(disr,a)));
*
*      (d) same as (a), above in case of non-diagonal activity: distribution of activity level
*
e_pxSubr(subr,aIn(a),ts) $ ( xpFlag(subr,a) $ (not diag(a)) ) ..

xp(subr,a,ts)/xp.scale(subr,a,ts)
  =E= sum(r_r(subr,disr),
    xp(disr,a,ts) * p_gfa(subr,a,disr,ts) * (px(disr,a,ts)/xp(subr,a,ts))**p_sigmaASubr(disr,a))/xp.scale(subr,a,ts);

```

Where  $p\_sigmaASubr$  is set to 5. It defines the substitution elasticities when aggregating production output from sub-region to nation. This implies that the national composition of output from the regions can change rather flexibly.

There are various equations which ensure that firm demand is aggregated over regions to country level, while it is assumed that all regions face the same prices for intermediate inputs. Factor prices clearly differ reflecting the assumption with regard to factor mobility across regions and sectors.

In order to ease reading the code and speeding up execution, the set `disr` depicts those nations which feature regional detail while the set `subr` depicts sub-regions. A second equation adds up intermediate in each sector over regions to national level:

```
e_xapNat(rsNat,iIn(i),aIn(a),ts) $ ((xaFlag(rsNat,i,a) or xaFlag(rsBat,a,i)) $ disr(rsNat)) ..
xa(rsNat,i,a,ts)/xa.scale(rsNat,i,a,ts)
=e= sum(r_r(subr,rsNat) $ (xaFlag(subr,a,i) or xaFlag(subr,i,a)), m_xa(subr,i,a,ts))/xa.scale(rsNat,i,a,ts);
```

An additional equation was added which distributes the total national factor supply `xft(disr)` to the regions:

```
*
* --- Factor market clearing from sub-regions to nation, not part of standard GTAP
*
e_xftReg(subr,fm,ts) $ (xftEqFlag(subr,fm) $ (pft.range(subr,fm,ts) ne 0)) ..
0 =e=
*
* --- "sluggish" factor supply based on CET function, factor is part of top level nest
*
(xft(subr,fm,ts)/xft.scale(subr,fm,ts)
- sum(r_r(subr,disr),p_gfr(subr,fm,disr,ts)*xft(disr,fm,ts)/xft.scale(subr,fm,ts)
* (pft(subr,fm,ts)/pft(disr,fm,ts))*p_omegafr(disr,fm))
) $ (sum(r_r(subr,disr),p_omegafr(disr,fm)) ne inf)
*
* --- fully mobile factor: prices are equal across regions
*
+ (pft(subr,fm,ts) - sum(r_r(subr,disr),pft(disr,fm,ts))) $ (sum(r_r(subr,disr),p_omegafr(disr,fm)) eq inf);
```

Clearly, if the transformation elasticity across regions  $p\_omegafr$  is set to zero, the regional factor stock is fixed. The last line in the equation depicts the case of infinite transformation which leads to uniform prices. A matching dual price aggregator defines the price for mobile factors at national level:

```
*
* --- factor supply from nation to sub-regions, not part of standard GTAP
*
e_pftNat(disr,fm,ts) $ (xftEqFlag(disr,fm) and (pft.range(disr,fm,ts) ne 0)) ..
0 =E=
*
* --- sluggish factor mobility across regions, dual price aggregator
*
[ pft(disr,fm,ts) - sum(r_r(subr,disr) $ xftEqFlag(subr,fm),
p_gfr(subr,fm,disr,ts)*pft(subr,fm,ts)**(1+p_omegafr(disr,fm))**(1/(1+p_omegafr(disr,fm))))
] $ (p_omegafr(disr,fm) ne inf)
*
* --- fully mobile factors across regions, uniform prices => physiscal aggregation
*
+ [
xft(disr,fm,ts)/xft.scale(disr,fm,ts) - sum(r_r(subr,disr), xft(subr,fm,ts))/xft.scale(disr,fm,ts)
] $ (p_omegafr(disr,fm) eq inf);
```

Not that the last line provides the physical linear aggregation in case that the transformation elasticity is infinite.

In order to avoid that the tax income equations are changed in order to reflect sub-national detail, factor use for each activity and matching prices are defined in two additional equations at national level:



```

*
* --- Average factor returns at national level, aggregated from sub-regions, exhaustion
* (Not part of standard GTAP)
*
e_pfNat(disr,f,aIn(a),ts) $ (xfFlag(disr,f,a) $ (pf.range(disr,f,a,ts) ne 0)) ..
    pf(disr,f,a,ts) * xf(disr,f,a,ts)/xf.scale(disr,f,a,ts)
    =E= sum(r_r(subr,disr) $ xfFlag(subr,f,a), pf(subr,f,a,ts)*xf(subr,f,a,ts))/xf.scale(disr,f,a,ts);
*
* --- Total national factor use, added up over sub-regions
* (Not part of standard GTAP)
*
e_xfNat(disr,f,aIn(a),ts) $ xfFlag(disr,f,a) ..
    xf(disr,f,a,ts)/xf.scale(disr,f,a,ts)
    =E= sum(r_r(subr,disr) $ xfFlag(subr,f,a), xf(subr,f,a,ts))/xf.scale(disr,f,a,ts);

```

The code was updated at various places if \$ rNat conditions to avoid that data transformation for sub-regional data are executed which do not relate to production or factor markets. During variable initialization and calibration, it is assumed that the regional data on intermediate input and primary factor use match the national totals (see next section for an implementation). Factor, production and immediate taxes rates are taken from the national SAMs.

```

fctts.l(r,f,a,t0) $ fcttsV0(r,f,a) = fcttsV0(r,f,a)/sam0(r,f,a) ;
fcttx.l(r,f,a,t0) $ fcttxV0(r,f,a) = fcttxV0(r,f,a)/sam0(r,f,a) ;

fctts.l(subR,f,a,t0) = sum(r_r(subR,r),fctts.l(r,f,a,t0));
fcttx.l(subR,f,a,t0) = sum(r_r(subR,r),fcttx.l(r,f,a,t0));

sam0(subR,"fctts",a) = sum(f, sam0(subR,f,a)*fctts.l(subR,f,a,"%t0%"));
sam0(subR,"fcttx",a) = sum(f, sam0(subR,f,a)*fcttx.l(subR,f,a,"%t0%"));

```

The intermediate taxes are implicitly taken over by using the relation between the national Armington intermediate demands and the national SAM entries net of taxes:

```

*
* --- Armington aggregator quantity in benchmark
* (calculated from value exhaustion)
*
xa.l(r,i,aa,t0) $ pa.l(r,i,aa,t0)
    = (pdp.l(r,i,aa,t0)*xd.l(r,i,aa,t0)
    + pmp.l(r,i,aa,t0)*xm.l(r,i,aa,t0))/pa.l(r,i,aa,t0) ;

xa.l(subR,i,a,t0) $ sam0(subR,i,a)
    = sam0(subR,i,a) * sum(r_r(subR,r), xa.l(r,i,a,t0)/sam0(r,i,a));

```

There were otherwise very limited changes to the calibration code.

### 3.3.4 Implementation for European countries at NUTS2 level

The NUTS (Nomenclature des unités territoriales statistiques) system provides a classification of administrative units for the European Union and some further regions. After NUTS0 (= nation) and NUTS1 (=federal state or similar), NUTS2 provides already relatively small regional units for economy wide assessment and beyond. In the context of the EU funded research project CAPRI-RD ([http://www.ilr.uni-bonn.de/agpo/rsrch/capri-rd/capri\\_rd\\_e.htm](http://www.ilr.uni-bonn.de/agpo/rsrch/capri-rd/capri_rd_e.htm)), regional SAMs at NUTS2 level and matching national ones for the EU member and candidates countries were compiled (Ferrari et al. 2012<sup>20</sup>). The SAMs feature somewhat limited sector detail (Agriculture, Forestry, Other primary, Food

<sup>20</sup> Ferrari, E., Himics, M. and Mueller M. (2010): WP2.2 Databases – Regional Social Accounting Matrices Deliverable: D2.2.4, Procedure for the compilation of regional SAMs based on national SAMs and available regional datasets: dataset and documentation; CAPRI-RD deliverable D2.2.4, <http://www.ilr.uni-bonn.de/agpo/rsrch/capri-rd/docs/d2.2.4.pdf>



processing, Other Manufacturing, Energy, Construction, Trade and transport, Hotels and restaurants, Education, Other Services) which reflects both data availability and the aim to model rural development policies under the Common Agricultural Policy (CAP). A specific feature of the SAMs is a dis-aggregation of intermediate demand to regional, national and imported origin. A matching single country CGE was developed (Törmä et al. 2010<sup>21</sup>, 2010, Britz 2012<sup>22</sup>) with a rather detailed driver to map individual rural development measures from the CAP into shocks for the model and an interface to the regional supply models and the market model of the partial equilibrium CAPRI modeling system. So far, the application of that model is rather limited (Schröder et al. 2015<sup>23</sup>, Britz et al. 2015<sup>24</sup>). We use here only the production data of these regionals SAMs which also breaks down demand to the regional level and provides also details on the national and regional government accounts.

The introduction of the list of sub-regions into the static set r (= all regions) which is used as the domain for various variables, equations and parameters requires that the list is available when the data from GTAPAGG are read. Accordingly, the data set program is partly read rather early in the program sequence:

```
$if %modulesGTAP_NUTS2==on          $batinclude 'gtapNUTS2\gtapNUTS2.gms' decl
$batinclude 'util\title.gms' '''Load data base for %dataSet%, %shock_withoutPath%'''
$include "%datdir%\gtap8Set.gms"
$include "build\load_gtapAgg.gms"
```

To load the following list with the NUTS2 regions:

```
$iftheni.decl %1==decl
set s_nuts2SAM "NUTS2 labels used in SAM"/
BE100000 "Region de Bruxelles-Capitale"
BE210000 "Antwerpen (Arrondissement)"
BE220000 "Limburg (B)"
BE230000 "Oost-Vlaanderen"
BE240000 "Vlaams Brabant"
BE250000 "West-Vlaanderen"
BE310000 "Brabant Wallon"
BE320000 "Hainaut"
BE330000 "Verviers"
BE340000 "Luxembourg (B)"
BE350000 "Namur"
BG310000 "Severozapaden"
BG320000 "Severozentralen"
BG330000 "Severozisten"
```

---

<sup>21</sup> Törmä, H., Zawalinska, K., Blanco-Fonseco M., Ferrari, E., Jansspon T. (2010): WP3.2 Model development and adaptation – Regional CGEs Deliverable: D3.2.1 Regional CGE model layout with a focus on integration with the partial equilibrium models and modelling of RD measures; CAPRI-RD deliverable D3.2.1, <http://www.ilr.uni-bonn.de/agpo/rsrch/capri-rd/docs/d3.2.1.pdf>

<sup>22</sup> Britz W. (2012) WP3.2 Model development and adaptation – Regional CGEs Additional Deliverable: D3.2.4 RegCgeEU+ in GAMS, documentation including the Graphical User Interface, CAPRI-RD deliverable D3.2.4, <http://www.ilr.uni-bonn.de/agpo/rsrch/capri-rd/docs/d3.2.4.pdf>

<sup>23</sup> Schroeder, L. A., Gocht, A., Britz, W. (2015): The Impact of Pillar II Funding: Validation from a Modelling and Evaluation Perspective, Journal of Agricultural Economics 66(2): 415–441

<sup>24</sup> Britz, W., Dudu, H., Ferrari, E. (2015): Economy-wide Impacts of Food Waste Reduction: A General Equilibrium Approach, selected paper presented at the International Conference of Agricultural Economists (ICAE 2015), 8-14 August 2015, Milan (Italy).

The actual data processing consists of the following steps:

1. Defining the regional hierarchy. Currently, the program only introduces sub-regions if the related country is a separate region in the GTAPAGG data set:

```

set rSam(r0); rSam(r0) $ sum(r0_msSam(r0,s_msSam),1) = yes;

check(s_msSam) = sum((mapr(r0,r),r0_msSam(r0,s_msSam)),1)-1;
set s_msSam_r(s_msSam,r);
*
* --- take out regions where countries with NUTS2 and without NUTS2 information are mixed
*
s_msSam_r(s_msSam,r) $ sum((mapr(r0,r),r0_msSam(r0,s_msSam)),1) = YES;
*
* --- take out regions where countries with NUTS2 and without NUTS2 information are mixed
*
s_msSam_r(s_msSam,r) $ (s_msSam_r(s_msSam,r) $ sum(mapr(r0,r) $ (not sum(r0_msSam(r0,s_msSam),1)),1)) = NO;

```

From there, the regional hierarchy is defined:

```

set subReg_r(r2,r); subReg_r(r2,r) $ sum( (subReg_nuts0(r2,s_msSam),s_msSam_r(s_msSam,r)),1) = YES;

r_r(r2,r) $ sum(subreg_r(r2,r),1) = yes;
r_r(r,r) $ sum(subreg_r(r2,r),1) = no;

```

And the list of regions which are dis-aggregated:

```

disR(rNat) $ sum(subreg_r(r2,rNat),1) = yes;

```

2. As the regional SAMs will typically feature a differently detail sector list compared to what is used elsewhere in the model, a mapping and consistency check is required. First, the link between the original 57 GTAP sectors and the sectors in the regional SAMs is set-up:

```

*
* --- define mapping between current sectors in GTAP models and given
* sectors in REGCGE
*
set sInd_i0(s_ind,iori) / "AGR".(pdr,wht,gro,v_f,osd,c_b,pfb,ocr,ctl,oap,rmk,wol)
    "FOR".frs
    "OPP".(fsh,coa,oil,gas,omn)
    "FOP".(cmt,ont,vol,mil,pcr,sgr,ofd,b_t,tex,wap,lea)
    "MAN".(lum,ppp,p_c,crp,nmm,i_s,nfm,fmp,mvh,otn,ele,ome,omf)
    "ENE".(ely,gdt,wtr)
    "CNS".(cns,dwe)
    "TTR".(trd,otp,wtp,atp)
    "HOT".ros
    "EDU".osg
    "OSE".(cmn,ofi,isr,obs,cgds)
    /;

```

First, SAM entries for intermediate input use are set up which assign the (sum of the) original regional SAM to the new commodities and sectors:

```

*
* --- intermediate input use: aggregate regional sam entry
*
p_rSam(R2Cur,i,a) = sum( (i_in2(i,in2),a_in2(a,in2)), p_nuts2(R2Cur,in2,in22)*gblScale);
p_rSam(r2Cur,"int",a) = sum(i,p_rSam(r2Cur,i,a));

```

Next, a correction factor is defined and applied to ensure that the regional entries match the national ones:

```

p_corFacR(rs,balRows,A) = sum( subReg_r(r2Cur,rs),p_rSam(R2Cur,balRows,a));
p_corFacR(rs,balRows,A) $ p_corFacR(rs,balRows,A) = sam0(rs,balRows,a)/p_corFacR(rs,balRows,A);

p_rSam(R2Cur,balRows,a) $ p_rSam(R2Cur,balRows,a)
= p_rSam(R2Cur,balRows,a) * sum( subReg_r(r2Cur,rs),p_corFacR(rs,balRows,a));

```

The program is not fully shown here, it also removes tiny regional entries. Finally, the behavioral parameters are defined:

```

sigmau(subReg,a) = sum(subReg_r(subReg,rs), sigmau(rs,a));
sigmand(subReg,a) = sum(subReg_r(subReg,rs), sigmand(rs,a));

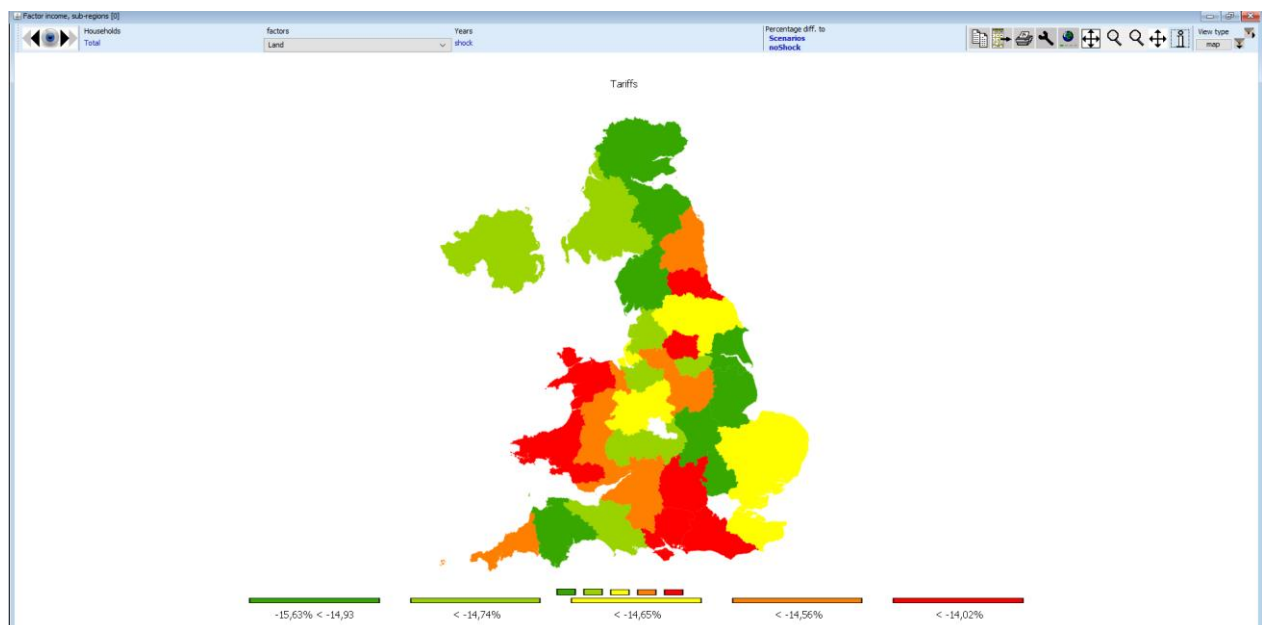
omegafr(rs,fn)      = 2.0;
omegafr(rs,"capital") = inf;
omegafr(rs,"land")   = 0.00;
    
```

i.e.:

3. The substitution elasticities at national level are also used for the regions, and
4. Land cannot move between regions, capital is assumed as fully mobile and labor as sluggish.  
Note that natural resources are immobile at sector level.

### 3.3.5 Result exploitation

So far, there is only one specific view which uses a map with regional detail:



### 3.4 Armington with commitment terms

- Wolfgang Britz, 2021 –

#### 3.4.1 Background

This module allows introducing constant terms similar to a LES function in the bi-lateral import demand equations driven otherwise by a CES function. This allows dampening or increases the price responsiveness of selected bi-lateral trade flows, for instance, to counteract the “small shares stay small” problem of the standard CES approach.

As seen from the following graphic with equations in mathematical notation, main differences to the usual CES aggregate are that (1) the Marshallians comprise constant terms  $c_i$ , (2) the exhaustion price is different from dual price CES index, (3) non-committed expenditure need to be calculated, which defines utility via exhaustion, and (4) the average import price used in the remainder of the model needs to be defined from expenditure exhaustion:

Utility  
(import demand  
quantity index)

Marshallians

$$x_i = u \delta_i \left( \frac{\tilde{p}}{p_i} \right)^\sigma + c_i$$

CES price index

$$\tilde{p} = \sum_i \delta_i p_i^{1-\sigma}$$

Non-committed expenditure

$$nce = e - \sum_i c_i p_i$$

u from exhaustion

$$u = nce / \tilde{p}$$

Definition of exhaustion price

$$e = u \hat{p}$$

A negative commitment  
increases the price elasticity

Note:  $e$  defined in other parts of model at given  $\hat{p}$

#### 3.4.2 Benchmarking and GUI setting

In opposite to the standard CES case, the LES-CES solution has an additional vector of product specific parameters. One set of import demand observations at a given substitution elasticity does hence not allow to identify both the share parameters and these additional constant terms. In order to proceed, a second set of “artificial” observations is introduced which described the desired changes in import shares relative to the standard model at a 10% price drop on the bi-lateral trade link.

The benchmarking process uses a separate NLP model, defined solely for one nation, product combination, which comprises the variables and equations from the LES-CES system, namely, for two set of observations of import prices and quantities called calPoints. This small model can also be used as a didactic one to experiment with expanded CES Armington model outside of the CGE model itself.

During calibration the share parameter  $v_s$  and the commitment terms  $v_c$  are to be defined such that the differences between the given second Marshallians observations  $p_x$  and the endogenous one  $v_x$  become minimal.

The benchmarking model comprises the following equations:

- (1) Definition of the CES price index  $v\_pindCes$  based on the usual dual CES price aggregator, at given prices  $p\_p$ :

```
*
* --- CES price index
*
e_imppCES(calPoints) ..
    v_pindCes(calPoints) =E= sum(rnat1 $ v_s.l(rnat1), v_s(rnat1) * p_p(rnat1,calPoints) ** (1-p_p_sigmaW))**(1/(1-p_p_sigmaW));
```

- (2) Definition of non-committed expenditure  $v\_nonComE$ . Note that the given budget for imports  $v\_bud$  is fixed during benachmarking.

```
*
* --- Non-Committed expenditure
*
e_nonComE(calPoints) ..
    v_nonComE(calPoints) =E= v_bud(calPoints) - sum(rnat1 $ v_s.l(rnat1), v_c(rnat1) * p_p(rnat1,calPoints));
```

- (3) Indirect definition of utility  $v\_u$  (which differs from total import quantity by the commitment terms) from non-committed expenditure and CES price index based on expenditure exhaustion:

```
*
* --- Definition of u
*
e_uCesLes(calPoints) ..
    v_u(calPoints) * v_pindCes(calPoints) =E= v_nonComE(calPoints);
```

- (4) Marshallian demands  $v\_x$ , the difference to the standard CES being the constant terms  $v\_c$ :

```
*
* --- Marshallian demands
*
e_marshall(rnat1,calPoints) $ v_s.l(rnat1) ..
    v_x(rnat1,calPoints) =E= (v_u(calPoints) * v_s(rnat1) * (v_pindCes(calPoints)/p_p(rnat1,calPoints))**(p_p_sigmaW + v_c(rnat1)));
```

- (5) Exhaustion price index  $v\_impp$

```
*
* --- Definition of price index (u is implicate in simulation model where budget is exogenous)
*
e_imppP(calPoints) ..
    v_bud(calPoints) =E= v_u(calPoints) * v_impp(calPoints);
```

- (6) Adding up of endogenous share parameters  $v_s$ :

```
*
* --- Share parameters add up to unity
*
e_addUpShare .. sum(rnat1 $ v_s.l(rnat1), v_s(rnat1)) =E= p_sumShare;
```

- (7) Inequality to make sure that Marshallian demand exceed commitment terms (to avoid that  $v_x$  is driven into the negative domain during simulation):

```
*
* --- Bounds on commitment terms to prevent problems in later sims
*
e_xc(rnat1,calPoints) $ v_s.l(rnat1) .. v_x(rnat1,calPoints)*0.50 =G= v_c(rnat1);
```

- (8) The objection function is defined as follows:

```

*
* --- fit for expert point (benchmark is fixed)
*
e_objFit .. v_opt * card(rnat1)
           =E= sum(rnat1, sqr[ (v_x(rnat1,"expert") - p_x(rnat1,"expert"))/(p_x(rnat1,"expert")+1.E-8) ]);

```

An additional equation is needed if the MRIO extension is switched on, where tariffs and import shares are given for each MRIO agent:

```

$iftheni.MRIO "%modulesGTAP_Mrio%"=="on"
*
* --- depending on the differences in c.i.f. prices across importers, the MRIO extension can imply quite different
* shares of the value of the commitment terms on total import value by agent. This equation ensures that the
* implicate expenditure can not exceed spendings by more than 25%
*
e_nonComEN(calPoints,mrioA) $ (card(curMioI) $ sum( (rsNat,curMioI,rnat1), mrioSplitFactors(rsNat,curMioI,rNat1,mrioA,"cif"))) ..
    sum(rnat1 $ (v_s.l(rnat1) $ sum( (rsNat,curMioI), mrioSplitFactors(rsNat,curMioI,rNat1,mrioA,"cif"))),
        v_c(rnat1) * p_p(rnat1,calPoints)
            * sum( (rsNat,curMioI), mrioSplitFactors(rsNat,curMioI,rNat1,mrioA,"cif")
                * (1+mrioSplitFactors(rsNat,curMioI,rNat1,mrioA,"m")/mrioSplitFactors(rsNat,curMioI,rNat1,mrioA,"cif"))))
            =G= -sum(rNat1, v_x(rNat1,"%t0%") * p_p(rNat1,calPoints)
                * sum( (rsNat,curMioI), mrioSplitFactors(rsNat,curMioI,rNat1,mrioA,"cif"),
                    mrioSplitFactors(rsNat,curMioI,rNat1,mrioA,"cif")
                        * (1+mrioSplitFactors(rsNat,curMioI,rNat1,mrioA,"m")/mrioSplitFactors(rsNat,curMioI,rNat1,mrioA,"cif"))
                    ) * 0.25;
$endif.MRIO

```

For the benchmarking, the user can (1) define a multiplicative factor which defines the change in import demand relative to the standard Armington when the c.i.f. price drops by 10% (which can be also understood as a drop in the tariff), (2) define a thresholds for the flows to boost.

Accordingly, the benchmarking code first calculates the Marshallians under the standard response:

```

*
* --- simulate standard response (i.e. without commitment terms) at given total demand and updated prices
*
v_x(rnat1,calPoints) $ v_s(rnat1) = v_u(calPoints) * v_s(rNat1) * (v_pindCes(calPoints)/p_p(rnat1,calPoints))**p_p_sigmaW;

```

and defines from there “expert” points for the two cases, i.e. expected unchanged response and boosted response:

```

*
* --- for shares above the critical limit, we calibrated against the standard response
*
p_x(rnat1,"Expert") $ ((v_s(rNat1) gt 1.E-%arm_les_ces_boostShare%*p_sumShare) $ v_s(rNat1)) = v_x.l(rnat1,"expert");
*
* --- otherwise, we simulate the response at a share which is multiplied with the user chosen boostFactor,
* but does not exceed the critical share limit
*
p_x(rnat1,"Expert") $ ( (v_s(rNat1) le 1.E-%arm_les_ces_boostShare%*p_sumShare) $ v_s(rNat1) $ arm_les_ces_exp_sel(rnat1))
= min(1.E-%arm_les_ces_boostShare%*v_u("expert") * (v_pindCes("expert")/p_p(rnat1,"expert"))**p_p_sigmaW,
    v_x(rnat1,"expert") * %arm_les_ces_boostFactor%);

```

The remainder code can be found in “armLesCes\armLesCES\_cal.gms”.

The relevant settings on the interface are shown below:

General settings   Model structure   Parameters and indices   AEZ   NUTS2   Tariff lines   myGTAP   MRIO   ARM_LES_CES   PE   G-RDEM   Imperfect competition model   SPE   Closures   Output options   Algorithm settings   GAMS																																											
ARM_LES_CES																																											
Products													Imports				Exporters																										
fat-c	bio-c	whi-c	bari-c	malz-c	sorg-c	oat-c	rye-c	ocor-c	soy-c	palm-c	rape-c	plb-c	oso-c	phi	Vietnam	phi	Vietnam																										
pot-c	rtb-c	leg-c	ovag-c	tom-c	appi-c	grap-c	cltr-c	banp-c	ofur-c	v_fo-c	c_b-c	ptb-c	coco-c	Bangladesh	ind	usa	usa																										
teas-c	coff-c	ocro-c	cattle-c	orum-c	plig-c	poul-c	osap-c	rmk-c	woic-c	frs-c	coo-c	oil-c	gas-c	can	Nicaragua	Bolivia	can	Nicaragua																									
ort-c	ctim-c	orumMeat-c	Pigmeat-c	othMeat-c	olivOil-c	palmOil-c	soyCake-c	soyOil-c	rapCake-c	rapOil-c	voic-c	mil-c	pcr-c	bra	arg	deu	bra	arg																									
sgir-c	ofdAnim-c	ofdOther-c	b_l-c	tex-c	wap-c	lea-c	lum-c	ppp-c	p_c-c	chm-c	bph-c	rrp-c	nmm-c	EU28	FormerUSSR	Ethiopia	EU28	FormerUSSR																									
l_s-c	ntm-c	fmp-c	ele-c	eeq-c	ome-c	mvh-c	oth-c	omf-c	gdl-c	wtr-c	chs-c	trd-c	afs-c	OceAsia	Malawi	Kenya	OceAsia	Malawi																									
otg-c	wip-c	atp-c	whs-c	cmr-c	ofc-c	ins-c	rsa-c	obs-c	ros-c	osg-c	edu-c	hnt-c	dwe-c	Ghana	Nigeria	RestOfAm	Ghana	Nigeria																									
TrD-c	NuclearBL-c	CoalBL-c	GasBL-c	WindBL-c	HydroBL-c	OilBL-c	OtherBL-c	GasP-c	HydroP-c	OilP-c	SolarP-c																																
Threshold share for flows to boost, 1.E-																																											
Boost factor at 10% c.i.f. reduction																																											
4%																																											
10%																																											

i.e. the user can determine for which products, importers and exports the “boost” mechanism shall be activated.

### 3.4.3 Equations

The  $e_{xw}$  equation for bi-lateral import demand in the GTAP Standard model is replaced by the following one called  $e_{xwArmLesCes}$  which considers simultaneously the standard case without and

case with commitment terms. The cases where the commitment terms are considered as defined by a cross of importing countries and products `arm_LES_CES(rnat,i)`:

```
e_xwArmLesCes(rNat1,iIn(i),rNat,ts) $ (xwFlag(rNat1,i,rNat) $ rrComb(rNat1,rNat) $ (not iMrio(i)) $ (not iSpe(i)) ) ..

xw(rNat1,i,rNat,ts)/xw.scale(rNat1,i,rNat,ts)
=e=
[ ( p_amw(rNat1,i,rNat,ts)
*
    * ( xmt(rNat,i,ts) $ (not arm_LES_CES(rnat,i))
      + xmtU(rNat,i,ts) $ arm_LES_CES(rnat,i))
    * [( ( pmt(rNat,i,ts) $ (not arm_LES_CES(rnat,i))
      + pmtU(rNat,i,ts) $ arm_LES_CES(rnat,i))
      /[(1 + imptx(rNat1,i,rNat,ts) + mtax(rNat,i,ts))
*
    --- Note WB: cif price is defined via macro %pmcif%, normally not a variable in model
*
    * %pmcif%(rNat1,i,rNat,ts) ])**p_sigmaw(rNat,i)
    * m_lambdam(rNat1,i,rNat,ts)**(p_sigmaw(rNat,i) - 1)}
    $$iftheni.smallShare "%hasSmallShare%"=="on"
      $ (not isSmallImpShare(rNat1,i,rNat))
      + m_lambdam(rNat1,i,rNat,ts) $ isSmallImpShare(rNat1,i,rNat)
    $$endif.smallShare
  ]
  + p_amwCom(rNat1,i,rNat,ts)
)/xw.scale(rNat1,i,rNat,ts) ]
```

The sole difference in here is that the utility aggregate from the import flows, aka total import demand by the agent, is captured by the variable `xmt` for the standard case and `xmtU` for the case with commitments. A problem might provide negative commitment terms which can provoke a situation where the adding the commitments to the usual share driven demands from a CES function implies a negative demand quantity. Therefore, the user can switch on a “fudging” mechanism which adds the following code to the equation:

```

    ,-----,
    $$iftheni.fudg "%armLesCesFudg%"=="on"
      $ (p_amwCom(rNat1,i,rNat,ts) ge 0)
    -{hnpVUSin(-xw.lo(rNat1,i,rNat,ts),-[
      ( p_amw(rNat1,i,rNat,ts)
*
    * ( xmt(rNat,i,ts) $ (not arm_LES_CES(rnat,i))
      + xmtU(rNat,i,ts) $ arm_LES_CES(rnat,i))
    * [( ( pmt(rNat,i,ts) $ (not arm_LES_CES(rnat,i))
      + pmtU(rNat,i,ts) $ arm_LES_CES(rnat,i))
      /[(1 + imptx(rNat1,i,rNat,ts) + mtax(rNat,i,ts))
*
    --- Note WB: cif price is defined via macro %pmcif%, normally not a variable in model
*
    * %pmcif%(rNat1,i,rNat,ts) ])**p_sigmaw(rNat,i)
    * m_lambdam(rNat1,i,rNat,ts)**(p_sigmaw(rNat,i) - 1)}
    $$iftheni.smallShare "%hasSmallShare%"=="on"
      $ (not isSmallImpShare(rNat1,i,rNat))
      + m_lambdam(rNat1,i,rNat,ts) $ isSmallImpShare(rNat1,i,rNat)
    $$endif.smallShare
  ]
  + p_amwCom(rNat1,i,rNat,ts)
  )
  ],1.E-7)/xw.scale(rNat1,i,rNat,ts)} $ (p_amwCom(rNat1,i,rNat,ts) lt 0)
    $$endif.fudg
    ;
```

It applies a Veelken-Ulbrich smoothing function which replaces a `min()` operator to ensure that `xw` is strictly positive. The average price of imports differs between the case with and without commitments:



as also seen from the mathematical notation above. The usual case without commitments defined the average import prices (= exhaustion) price from the CES dual price index aggregator directly. If commitments are present, the price index  $\text{pmtU}$  is different from this exhaustion price.

```
*
* ---- define non-commit expenditure on imports
*
e_nonComImpV(rNat,i,in(i),ts) $ ( (pmt.range(rNat,i,ts) ne 0) $ (not p_pmtElas(rNat,i)) $ xmtFlag(rNat,i) $ (not iMrio(i)) $ arm_LES_CES(rNat,i) ) .
nonComImpV(rnat,i,ts)/xmt.scale(rnat,i,ts)
=E=
*
*   --- total import expenditures
*
xmt(rNat,i,ts)/xmt.scale(rnat,i,ts) * pmt(rNat,i,ts)
*
*   --- minues values of commitment at c.i.f. prices
*
- sum(rnat1 $ p_ammCom(rNat1,i,rnat,ts),
      p_ammCom(rNat1,i,rnat,ts)*(1 + imptx(rNat1,i,rNat,ts) + mtax(rNat,i,ts)) * %pmcif(rNat1,i,rNat,ts) )/xmt.scale(rnat,i,ts);
```

```
e_xmtUArmLesCes(rNat,iIn(i),ts) $ ( (pmt.range(rNat,i,ts) ne 0) $ (not p_pmtElas(rNat,i)) $ xmtFlag(rNat,i)
                                     $ (not iMrio(i)) $ arm_LES_CES(rNat,i) ) ..

xmtU(rNat,i,ts)/xmt.scale(rnat,i,ts) * pmtU(rnat,i,ts)
=E= nonComImpV(rnat,i,ts)/xmt.scale(rnat,i,ts);
```

```
e_pmtUArmLesCes(rNat,iIn(i),ts) $ ( (pmt.range(rNat,i,ts) ne 0) $ (not p_pmtElas(rNat,i)) $ xmtFlag(rNat,i)
$ (not iMrio(i)) $ arm_LES_CES(rNat,i) ) ..

(pmt(rnat,i,ts)-pmtU(rnat,i,ts)) * xmtU(rnat,i,ts)/xmt.scale(rnat,i,ts) =E=
sum(rnat1 $ p_ammCom(rNat1,i,rnat,ts),
p_ammCom(rNat1,i,rnat,ts)*(1 + imptx(rNat1,i,rnat,ts) + mtax(rNat,i,ts)) $ %pmcif%(rNat1,i,rnat,ts) )/xmt.scale(rnat,i,ts)
```

### 3.5 Spatial Price Equilibrium instead of Armington

- Wolfgang Britz, 2021 -

#### 3.5.1 Background

The Armington assumption implies that quality differences are present according to where a product originates from. In many cases, this assumption makes sense already due to the fact that each product in the GTAP Data Base consists of many different tariff lines and with differences in the composition across regions. However, there are cases, such as for raw sugar or when working with data base with more detail, where the assumption of a homogenous quality and the law of one price might be more appropriate. This assumption can be employed at the level of bi-lateral flows with the spatial arbitrage solution drawing the Spatial Price Equilibrium (SPE) framework. It allows for emerging and vanishing trade flows and overcomes the “small shares stay small” problem. Furthermore, it does not require any behavioral parameters to steer substitution between the different origins. The SPE solution was used in a range of partial equilibrium models, for instance:

- Anania, G., 2010. EU Economic Partnership Agreements and WTO negotiations. A quantitative assessment of trade preference granting and erosion in the banana market. *Food Policy*, 35(2), pp.140-153
- Nolte, S., Buysse, J. and Van Huylenbroeck, G., 2012. Modelling the effects of an abolition of the EU sugar quota on internal prices, production and imports. *European review of agricultural economics*, 39(1), pp.75-94
- Wieck, C., Schlüter, S.W. and Britz, W., 2012. Assessment of the impact of avian influenza–related regulatory policies on poultry meat trade and welfare. *The World Economy*, 35(8), pp.1037-1052
- Jansson, T., Heckeley, T. (2009): A new estimator for trade costs and its small sample properties, *Economic Modelling* 26: 489-498

The assumption of the SPE framework implies first the law of one price, i.e. prices at c.i.f. plus tariff across origins for realized trade flows must be equal, and second, a physical aggregation from bi-lateral flows to total imports. Unobserved trade flows provide a challenge. According to the underlying assumptions, a zero flow occurs if f.o.b. plus transaction costs (= trade margins in the GTAP Data Base) plus tariff must exceed the c.i.f. price of the lowest cost importer. Unfortunately, the size of this difference is unknown as trade data bases (TASTE, MacMaps) underlying most global CGE analysis do not report tariffs for non-observed flows and the GTAP Data Base does not comprise transport margins for such zero observation. This requires assumptions, such as using the largest reported applied tariffs, or some empirical relation between transport margins and transport distance.

Moreover, the application of SPE comes along with numerical challenges. Tests indicate that a (stronger) non-linear response in CIF prices to changes in bi-laterally traded quantities is required then found in the standard GTAP model layout required for framework to function. The underlying reasons are as follows. The FOC require that a c.i.f. price plus tariff is equal to minimum c.i.f. plus tariff or trade is zero. Accordingly, a drop in a single c.i.f. price/tariff would imply that all other trade flows disappear. To keep some non-zero flows as a response to such a shock, the related production costs (defines f.o.b.) would need to fall to the point where the minimum bi-lateral price is reached, but this defines in turn new minimum prices for other export destination of the supplying region where production cost drop. This mechanism also does not work well as production costs hardly react to a moderate change in a specific bi-lateral trade flow. This means that the production cost adjustment response is too low to bring all SPE conditions into equilibrium for an import product market (at least

for smaller trade flows). A further reason for changes in import prices relates to the changes in trade cost, but the per unit transport margins and AVEs are independent of traded quantity on a flow (if there is no TRQ mechanisms). The transport margins are endogenous, but their changes depend on price feedback to changes in total trade. Further, the SPE mechanism typically applies to many bi-lateral flows for a product as GTAP tends to have dense bi-lateral trade matrices.

The proposed implementation renders the per unit bi-lateral transport margin depending on the traded quantities. Potential reasons why this might occur in reality is that longer domestic transport distances are needed to add the supply of more firms to expand trade, or other impact on marginal costs of transport if more is traded.

### 3.5.2 Equations and technical implementation

The first equation renders the per unit bi-lateral transport cost a function of traded quantities:

```
*
* --- per unit trade margins increase/decrease if trade flow increases/decreases relative to benchmark. Constant p_largestXW
* prevents (a) a division by zero for non-observed flows allowed to emerge, (b) that emerging flow face extremely
* large increasing in transport cost.
*
e_lambdamg(m,rnat,iIn(ispe),rpNat,ts) $ (p_amgm(m,rnat,ispe,rpNat) $ (lambdamg.range(m,rnat,ispe,rpNat,ts) ne 0) $ rrComb(rNat,rpNat)) ..
    lambdamg(m,rnat,ispe,rpNat,ts) =e=
    1/((xw(rNat,iSpe,rpNat,ts)+p_largestXW(rNat,iSpe))/(xw.1(rNat,iSpe,rpNat,"%t0%")+p_largestXW(rNat,iSpe))-1)*10+1);
```

In opposite to the usual way to exogenous the c.i.f. price, it is explicitly introduced in here:

```
*
* --- as per unit transport margins on each flows are endogenous, the definition of the c.i.f. price is more non-linear
* compared to the standard case. Combined with the (almost) convex wx x pmCIF =E= 0 relation, it has turned out that
* introducing the cif price as a variable helps the solver.
*
e_pmCIFspe(rNat,iIn(ispe),rpNat,ts) $ (xwFlag(rNat,iSpe,rpNat) $ rrComb(rNat,rpNat) $ (xw.range(rNat,iSpe,rpNat,ts) ne 0)) ..
    pmCIF(rNat,iSpe,rpNat,ts) =E= %pmCIF(rNat,iSpe,rpNat,ts);
```

The condition that the flow must be zero if price differences are found and no price differences are allowed if the flow is non-zero is captured by the following relation:

```
*
* --- spatial arbitrage condition: trade flows can only be non-zero if cif price is equal to import price.
* If the "manual" MCP solution method is used (speMultRef), an explicite slack is introduced
* and c.i.f. price not substituted out
*
e_xwSpe(rNat,iIn(ispe),rpNat,ts) $ (xwFlag(rNat,iSpe,rpNat) $ rrComb(rNat,rpNat) $ (xw.range(rNat,iSpe,rpNat,ts) ne 0)) ..
    (
    $$iftheni.multRef "%speMultRef%"=="on"
        pmCIF(rNat,iSpe,rpNat,ts)
    $$else.multRef
        %pmCIF(rNat,iSpe,rpNat,ts)
    $$endif.multRef

    * (1 + imptx(rNat,iSpe,rpNat,ts)) + mtax(rpNat,iSpe,ts) * p_speCal(rNat,iSpe,rpNat,ts)

    $$iftheni.multRef "%speMultRef%"=="on"
        - xwSpeSlack(rNat,iSpe,rpNat,ts) $ (solveType ne MCP)
    $$endif.multRef

    =E= pmt(rpNat,iSpe,ts);
```

Next, we have the market clearing condition, i.e. adding up total imports from the non-zero flows:

```
*
* --- market clearance for imports, defines average import price
*
e_pmtSpe(rsNat,iIn(ispe),ts) $ ( (pmt.range(rsNat,iSpe,ts) ne 0) $ (not p_pmtEla(rsNat,iSpe)) $ xmtFlag(rsNat,iSpe) ) ..
    xmt(rsNat,iSpe,ts)*xmt0(rsNat,iSpe)/xmt.scale(rsNat,iSpe,ts)
    =E= sum(rpNat $ xwFlag(rpNat,iSpe,rsNat), xw(rpNat,iSpe,rsNat,ts))/xmt.scale(rsNat,iSpe,ts);
```

In case a NLP solver instead of a MCP solver is used, the following implementation of a multiplicative fudging function is implemented:

```

*
* --- FOC from spatial arbitrage: xw * xwSpeSlack = 0. To improve numerical stability, this is slightly
* differently implemented.
*
e_xwSpeMultRef(rNat,iIn(iSpe),rpNat,ts) $ (xwFlag(rNat,iSpe,rpNat) $ rrComb(rNat,rpNat) $ (xw.range(rNat,iSpe,rpNat,ts) ne 0)
$ ((card(rsNat) eq 1) or (solveType eq CNS) or (muSpe le 1.E-10)) ) ..

(xwSpeSlack(rNat,iSpe,rpNat,ts)-1.E-16)*(xw(rNat,iSpe,rpNat,ts)/xw.scale(rNat,iSpe,rpNat,ts)-1.E-16)
=E= (xwSpeSlack.1(rNat,iSpe,rpNat,"%t0%")-1.E-16)*(xw.1(rNat,iSpe,rpNat,"%t0%")/xw.scale(rNat,iSpe,rpNat,ts)-1.E-16);

e_xwSpeMultRefMU(rNat,iIn(iSpe),rpNat,ts) $ (xwFlag(rNat,iSpe,rpNat) $ rrComb(rNat,rpNat) $ (xw.range(rNat,iSpe,rpNat,ts) ne 0)
$ ((card(rsNat) gt 1) and (solveType eq NLP) and (muSpe gt 1.E-10)) ) ..

(xwSpeSlack(rNat,iSpe,rpNat,ts)-1.E-16)*(xw(rNat,iSpe,rpNat,ts)/xw.scale(rNat,iSpe,rpNat,ts)-1.E-16)
=L= (xwSpeSlack.1(rNat,iSpe,rpNat,"%t0%")-1.E-16)*(xw.1(rNat,iSpe,rpNat,"%t0%")/xw.scale(rNat,iSpe,rpNat,ts)-1.E-16)
+ muSpe;

```

The resulting sub-model is defined as follows:

```

model m_spe /
  $$iftheni.ACET not "%SPENonLinear%"=="Additive CET"
    e_lambda.mg.lambdamg
    $$iftheni.multRef "%speMultRef%"=="on"
      e_pmcifSpe.pmcif
      e_xwSpeMultRef.xwSpeSlack
      e_xwSpeMultRefMu.xwSpeSlack
    $$endif.multRef
  $$endif.ACET
  e_xwSpe.xw
  e_pmtSpe.pmt
/;

```

The additive CET solution is not discussed in here.

Solving these equations on the context of the overall has proven quite challenging. Therefore, an own additional pre-solve mechanism was implemented. Where as standard pre-solve considers one model region, only, the pre-solves in here solve a global PE model for one product, only. In a loop, they take the products treated with the SPE mechanism (iSpe) into the model and restrict price changes to +/- 20% (changes of this size can usually only occur if some part of the model becomes infeasible).

```

*
* --- reset
*
muSpeOri = muSpe;
muSpe = 0;

*
* --- Note don't write iSpe(kp) as otherwise, implicate loop in fix_for_pe will not work!
*
loop(kp $ iSpe(kp),

  option kill=iIn; iIn(kp) = yes;
  option kill=aIn; aIn(a) $ sum ( (rs,iIn), xFlag(rs,a,iIn) ) = YES;
  $$include "model/fix_for_pe.gms"

  ps.lo(rs,iIn,tSim) = ps.1(rs,iIn,tSim) * 0.8;
  ps.up(rs,iIn,tSim) = ps.1(rs,iIn,tSim) * 1.2;

  gtap.optfile = standardOptfile;
  $$iftheni.modelType "%modelType%"=="MCP"
    solveType = MCP;
    solve gtap using MCP;
  $$else.modelType
    solveType = NLP;
    solve gtap using NLP minimizing dummy;
  $$endif.modelType
  $$batinclude "model/closures.gms" noIniVars

  if ((gtap.solverlink eq 6) and (not execError), handles(kp) = gtap.handle);
);

```

The code to collect and check the solutions is not shown in here. Afterwards, prices and quantities in all markets updated, before the full model is solved.

### 3.5.3 Interface

The interface allow, as shown below:

- To select which products are subject to the SPE implementation (this exclude to use other trade options, such as Melitz, Armington with commitments or the MRIO extension, for this product)
- If the non-linearities stem from updating the transport margins or from using a additive CET (not discussed in here)
- If pre-solves for the SPE markets should be used
- And the Mu parameter used during pre-solves

**Commodities**

SPE commodities

fsh-c	pdr-c	wht-c	barl-c	maiz-c	sorg-c	oat-c	rye-c
ocer-c	soy-c	palm-c	rape-c	olv-c	oso-c	pota-c	rttb-c
leg-c	oveg-c	toma-c	appl-c	grap-c	citr-c	banp-c	ofru-c
v_fo-c	c_b-c	pfb-c	coco-c	teas-c	coff-c	ocro-c	cattle-c
orum-c	plg-c	poul-c	oapo-c	rmk-c	wol-c	frs-c	coa-c
oil-c	gas-c	oxt-c	cttMeat-c	orumMeat-c	Pigmeat-c	othMeat-c	olivOil-c
palmOil-c	soyCake-c	soyOil-c	rapCake-c	rapOil-c	volc-c	mil-c	pcr-c
sgr-c	ofdAnim-c	ofdOther-c	b_t-c	tex-c	wap-c	lea-c	lum-c
ppp-c	p_c-c	chm-c	bph-c	rpp-c	nmm-c	i_s-c	nfm-c
fmp-c	ele-c	eeq-c	ome-c	mvh-c	otn-c	omf-c	gdt-c
wtr-c	cns-c	trd-c	afs-c	otp-c	wtp-c	atp-c	whs-c
cmn-c	ofi-c	ins-c	rsa-c	obs-c	ros-c	osg-c	edu-c
hht-c	dwe-c	TnD-c	NuclearBL-c	CoalBL-c	GasBL-c	WindBL-c	HydroBL-c
OilBL-c	OtherBL-c	GasP-c	HydroP-c	OilP-c	SolarP-c		

Non-linearities Transport margins ▾

Additional presolve for SPE markets ☒

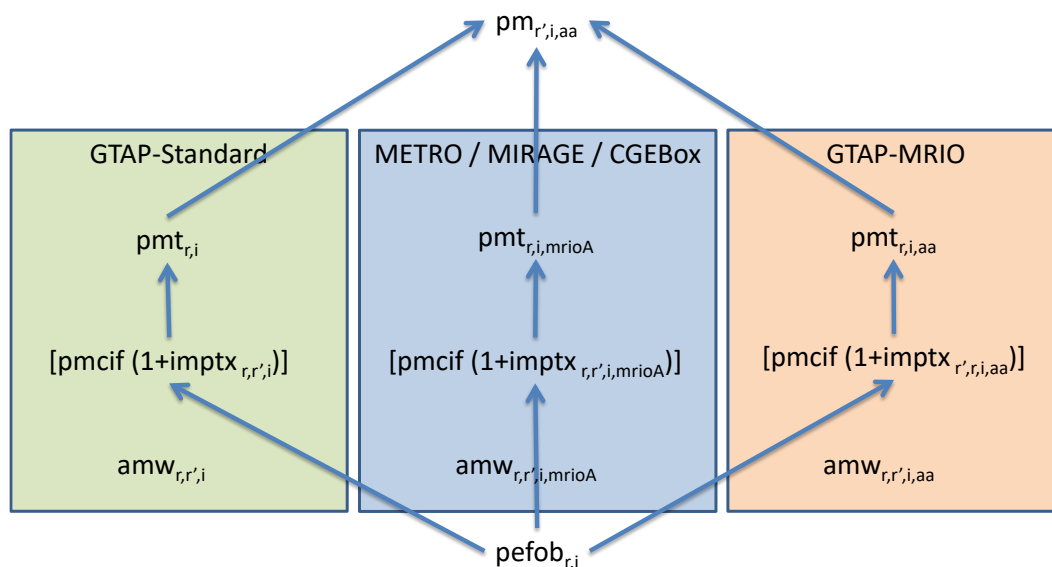
Mu, 1.E- 10 ▴ ▾

### 3.6 MRIO extension

- Wolfgang Britz, 2021 –

#### 3.6.1 Background

There is growing interest with regard to global value chains and life-cycle assessment along such chains which both require information on the bi-lateral sourcing of intermediate demand by industries and final demand agents. We discuss in here a module for CGEBox that is able to depict import demand differentiated by aggregate Armington agents, i.e. for aggregate intermediate use, household, government and investments. The module is set-up such that it can be used for specific sectors only, while the remaining ones use the standard Armington model or another trade specification; this selection is done via the GUI. The necessary data stem now from the GTAP MRIO data base (Carrico et al. 2020<sup>25</sup>) and can be added during the data preparation step by simply switching an option to active, whereas an older version was using MRIO split shares from the OECD's METRO model.



The graph above compares the chosen implementation (in the middle) with the GTAP-Standard approach and a fully fledged MRIO implementation as in GTAP-MRIO (Corong 2020<sup>26</sup>). In the GTAP-Standard model, all Armington agents face the same import prices as tariffs and bi-lateral trade shares are not differentiated by agent. In the full MRIO implementation, tariffs and bi-lateral trade shares can differ for each Armington agent, whereas in the chosen implementation, they are identical across production sectors, but can differ between intermediate, household, government and final demand, according to the so-called BEC classification at the tariff line. The chosen implementation considers that the GTAP-MRIO Data Base uses proportionality assumption to split the total (i.e. aggregated over production activities) bi-lateral intermediate input demand of a product to production

<sup>25</sup> Carrico, C., Corong, E., & van der Mensbrugghe, D. (2020). The GTAP version 10A Multi-Region Input Output (MRIO) Data Base. GTAP Research Memorandum No. 34, [https://www.gtap.agecon.purdue.edu/resources/res\\_display.asp?RecordID=6164](https://www.gtap.agecon.purdue.edu/resources/res_display.asp?RecordID=6164)

<sup>26</sup> Corong, E. (2020). The GTAP Multi-Region Input Output (MRIO) Model. GTAP Research Memorandum No. 37.



activities according to their total import demand (i.e. not differentiated by region of origin) to and thus aligns the model layout with the informational content of the underlying data base.

### 3.6.2 Definition of consistent split factors

In order to define a consistent benchmark for the model, the total import demands of the MRIO agents need to be exhausted by their bi-lateral demands for imports while at the same time the given bi-lateral export flows must be exhausted. We apply a Linear Loss framework to derive consistent shares (see `gams/build/mrio.gms`) instead of simply copying data derived from the GTAP-MRIO Data Base. This is necessary first as the filter program might change the value of bi-lateral import demands, and second, the GTAP-MRIO Data Base is not fully aligned with the standard version. A specific problem arises from the fact the filter process is not aware of the GTAP-MRIO information. Assume for instance that the GTAP-MRIO reports on a bi-lateral trade flow for a product demand of households, only, but that the overall import demand of households for this product is quite small in relative terms. The filter algorithm might remove the import demand by the household, but not the bi-lateral import flow of this product. As a consequence, there will a non-zero import flow in the SAM, but no import demand for any MRIO agent (here households) for which the GTAP-MRIO Data Base reports a bi-lateral import demand. This requires that the code is able to “invent” new MRIO flows as discussed below. Such cases are typical few and should relate to tiny flows anyhow. As same product detail is found in both data bases, a 1:1 mapping can be used. In case of additional product splits, currently, the same split factors are used.

As a first step, we aggregate the GTAP-MRIO data Base into a parameter which is termed “`mrioSplitFactors`” and stored in GDX container. It comprises in its raw format information on the value of bi-lateral trade for each of the four MRIO agents (total intermediate demand `int`, households `hhsld`, government `gov` and investment demand `inv`) at c.i.f. and c.i.f. plus tariffs. Note that also tariffs can differ across MRIO agents according to tariff dispersion at HS6 level.

As a first step, we aggregate the value at c.i.f. and c.i.f. plus tariffs, as captured by the set `priceLevel` from GTAP region `rDat` to model region `rNat` according to the mapping set `rr`:

```
mrioSplitFactors(rNat1,i,rNat,mrioA,priceLevel)
$ ( (sam0(rNat1,rNat,i)+imptxY0(rNat,rNat1,i)) and sum(mrioA_aa(mrioA,aa), xmm0(rnat1,i,aa)))
= sum( (rr(rDat,rNat),rr1(rDat1,rNat1),mm(prod_comm,i)), p_splitFactorsG(prod_comm,mrioA,rDat,rDat1,priceLevel));
```

In case of a split, bi-lateral values at c.i.f. are used as the allocation key:

```
*
* --- scale MRIO factors for split up products according to share on bi-lateral imports
*
mrioSplitFactors(rNat1,spliti,rNat,mrioA,priceLevel) $ mrioSplitFactors(rNat1,spliti,rNat,mrioA,priceLevel)
= mrioSplitFactors(rNat1,spliti,rNat,mrioA,priceLevel)
* sam0(rnat1,rNat,spliti)/sum(is_i(spliti1,i) $ is_i(spliti,i),sam0(rnat1,rNat,spliti1));
```

The balancing of this information against the given SAM and the information on the import demand at market prices for the production activities and final demand categories is based on LP framework with a linear loss objective. The reason is that performant LP solvers can be used in this case (see Britz 2021<sup>27</sup> for a discussion).

A first constraint `e_xmMrio` ensures that total import demand at market prices for each MRIO agent (i.e. at c.i.f. plus tariffs), when aggregated over the SAM agents linked to a MRIO agent (a 1:1 relation

---

<sup>27</sup> Britz, W. (2021): Comparing Penalty Functions in Balancing and Dis-aggregating Social Accounting Matrices, *Journal of Global Economic Analysis* 6(1): 34-81



for the three final demand categories, otherwise int is mapped to all production activities) as defined on the LHS must be exhausted by the MRIO transactions on the RHS:

```
*
* --- Exhaust total import demand by MRIO agent by the bi-lateral flows at cif+tariff
*
e_xmMrio(rnat,curI(i),mrioA) $ sum(mrioA_aa(mrioA,aa), xmm0(rnat,i,aa)) ..
    sum(mrioA_AA(mrioA,aa),xmm0(rnat,i,aa))/max(1,sum(mrioA_AA(mrioA,aa),xmm0(rnat,i,aa)))
    =E= [sum(rnat1 $ mrioSplitFactors(rNat,i,rNat1,mrioA,"m"),
        mrioSplitFactors(rNat,i,rNat1,mrioA,"m")) * (1+v_corP(rNat,i,rNat1,mrioA,"m")-v_corN(rNat,i,rNat1,mrioA,"m"))
        -sum(rnat1,(mrioSplitFactors(rNat,i,rNat1,mrioA,"m")+mrioSplitFactors(rNat,i,rNat1,mrioA,"m1")) * v_slack1(rNat,i,rNat1,mrioA))
    *
    * --- these are originally not observed entries. Their multiplier is pulled towards 0 => expected value is zero
    *
    + sum(rnat1 $ mrioSplitFactors(rNat,i,rNat1,mrioA,"m1"),
        mrioSplitFactors(rNat,i,rNat1,mrioA,"m1")) * v_mult(rNat,i,rNat1,mrioA,"m1"]]/max(1,sum(mrioA_AA(mrioA,aa),xmm0(rnat,i,aa)))
    .
```

The RHS comprises three terms:

- The originally reported transactions from the GTAP-MRIO Data Base as indicated with “m” for at market prices on the parameter mrioSplitFactors, multiplied with a correction factor which is split into a positive (v\_corP) and negative (v\_corN) relative correction.
- A slack which allows that values at c.i.f. and c.i.f. plus tariffs are not fully consistent, to capture potential small inaccuracies in the SAM.
- A term which allows introducing new MRIO flows, as motivated above.

Two equations exhaust bi-lateral trade as found in the SAM aggregated from the GTAP Data Base at c.i.f. and c.i.f. plus tariffs by the bi-lateral import demands of the MRIO agents.

```
*
* --- exhaustion of exports at c.i.f. value by the MRIO agents
*
e_cif(considerMrio(rnat,curI(i),rnat1)) $ sam0(rNat,rNat1,i) ..
    sam0(rnat,rnat1,i)/max(1,sam0(rnat,rnat1,i))
    =E= [ sum(mrioA $ mrioSplitFactors(rNat,i,rNat1,mrioA,"cif"),
        mrioSplitFactors(rNat,i,rNat1,mrioA,"cif")) * (1+v_corP(rNat,i,rNat1,mrioA,"cif")-v_corN(rNat,i,rNat1,mrioA,"cif"))
    *
    * --- these are originally not observed entries. Their negative correction factor is pulled toward -1 => expected value is zero
    *
    + sum(mrioA $ mrioSplitFactors(rNat,i,rNat1,mrioA,"cif1"),
        mrioSplitFactors(rNat,i,rNat1,mrioA,"cif1")) * v_mult(rNat,i,rNat1,mrioA,"cif1"]]/max(1,sam0(rnat,rnat1,i));
    .
```

For the case of c.i.f. plus tariffs, again at slack is introduced to reflect potential tiny imbalances in the SAM:

```
*
* --- exhaustion of exports at c.i.f. value + tariffs by the MRIO agents
*
e_cifM(considerMrio(rnat,curI(i),rnat1)) $ (sam0(rNat,rNat1,i)+imptxY0(rNat1,rNat,i)) ..
    [sam0(rnat,rnat1,i)+imptxY0(rNat1,rNat,i)]/max(sam0(rnat,rnat1,i)+imptxY0(rNat1,rNat,i),1)
    =E= [ sum(mrioA $ mrioSplitFactors(rNat,i,rNat1,mrioA,"m"),
        mrioSplitFactors(rNat,i,rNat1,mrioA,"m")) * (1+v_corP(rNat,i,rNat1,mrioA,"m")-v_corN(rNat,i,rNat1,mrioA,"m"))
        + sum(mrioA $ (mrioSplitFactors(rNat,i,rNat1,mrioA,"m1")+mrioSplitFactors(rNat,i,rNat1,mrioA,"m")),
            (mrioSplitFactors(rNat,i,rNat1,mrioA,"m1")+mrioSplitFactors(rNat,i,rNat1,mrioA,"m")) * v_slack2(rNat,i,rNat1))
        + sum(mrioA $ mrioSplitFactors(rNat,i,rNat1,mrioA,"m1"),
            mrioSplitFactors(rNat,i,rNat1,mrioA,"m1")) * v_mult(rNat,i,rNat1,mrioA,"m1"))
    ]/max(1,sam0(rnat,rnat1,i)+imptxY0(rNat1,rNat,i));
    .
```

There is an additional inequality constraint which ensures that all import tariffs assigned to a MRIO tariff, i.e. it ensure the value at c.i.f. plus tariffs on the position “m” exceed the c.i.f. value on “cif”:

```
*
* --- exports at c.i.f. value + tariffs must exceed exports at c.i.f. value for each agent to ensure positive import tariffs
* (slack not considered such it can contribute to feasibility)
*
e_cifLeM(considerMrio(rNat,curI(i),rnat1),mrioA) $ (mrioSplitFactors(rNat,i,rNat1,mrioA,"m") or mrioSplitFactors(rNat,i,rNat1,mrioA,"m1")) ..
    [ mrioSplitFactors(rNat,i,rNat1,mrioA,"m") * (1+v_corP(rNat,i,rNat1,mrioA,"m")-v_corN(rNat,i,rNat1,mrioA,"m"))
    + mrioSplitFactors(rNat,i,rNat1,mrioA,"m1") * v_mult(rNat,i,rNat1,mrioA,"m1"))
    /max(1, mrioSplitFactors(rNat,i,rNat1,mrioA,"m")+ mrioSplitFactors(rNat,i,rNat1,mrioA,"m1"))
    =G= [ mrioSplitFactors(rNat,i,rNat1,mrioA,"cif") * (1+v_corP(rNat,i,rNat1,mrioA,"cif")-v_corN(rNat,i,rNat1,mrioA,"cif"))
        + mrioSplitFactors(rNat,i,rNat1,mrioA,"cif1") * v_mult(rNat,i,rNat1,mrioA,"cif1")
        - (mrioSplitFactors(rNat,i,rNat1,mrioA,"cif")+mrioSplitFactors(rNat,i,rNat1,mrioA,"cif1")) * v_slack(rNat,i,rNat1,mrioA)
    ]/max(1, mrioSplitFactors(rNat,i,rNat1,mrioA,"m")+ mrioSplitFactors(rNat,i,rNat1,mrioA,"m1"));
    .
```

An additional inequality ensures that the corrections do not lead to implausible increases in tariffs by large differences between the correction on the “m” (= c.i.f. plus tariffs) and “cif” positions. The maximum allowed relative change is defined by the scalar `p_tarMaxDiff`:

```
*
* --- plausible values for newly introduce tariffs
*
e_cifGeT(considerMrio(rNat,curI(i),rNat1),mrioA) $ (mrioSplitFactors(rNat,i,rNat1,mrioA,"m") or mrioSplitFactors(rNat,i,rNat1,mrioA,"m1")) ..
[ mrioSplitFactors(rNat,i,rNat1,mrioA,"m") * (1+v_corP(rNat,i,rNat1,mrioA,"m")-v_corN(rNat,i,rNat1,mrioA,"m"))
+ mrioSplitFactors(rNat,i,rNat1,mrioA,"m1") * v_mult(rNat,i,rNat1,mrioA,"m1")]
/max(1, mrioSplitFactors(rNat,i,rNat1,mrioA,"m")+ mrioSplitFactors(rNat,i,rNat1,mrioA,"m1"))

=L=
[ mrioSplitFactors(rNat,i,rNat1,mrioA,"cif") * (1+v_corP(rNat,i,rNat1,mrioA,"cif")-v_corN(rNat,i,rNat1,mrioA,"cif"))
+ mrioSplitFactors(rNat,i,rNat1,mrioA,"cif1") * v_mult(rNat,i,rNat1,mrioA,"cif1")]

*
* --- less than maximum found in data
*
* [ smax(mrioA1 $ mrioSplitFactors(rNat,i,rNat1,mrioA1,"cif"),
* mrioSplitFactors(rNat,i,rNat1,mrioA1,"m")/mrioSplitFactors(rNat,i,rNat1,mrioA1,"cif"))
* sum(mrioA1, mrioSplitFactors(rNat,i,rNat1,mrioA1,"cif"))
* + ((sam0(rnat,rnat1,i)+imptxY0(rNat1,rNat,i))/sam0(rnat,rnat1,i))
* $ (not sum(mrioA1, mrioSplitFactors(rNat,i,rNat1,mrioA1,"cif"))))
*
*
* -- plus some correction factors
*
* + max(p_tarMaxDiff,p_tarMaxDiff * [(sam0(rnat,rnat1,i)+imptxY0(rNat1,rNat,i))/sam0(rnat,rnat1,i)-1])
*
* ]
*
* 1/max(1, mrioSplitFactors(rNat,i,rNat1,mrioA,"m")+mrioSplitFactors(rNat,i,rNat1,mrioA,"m1"));
```

The objective function minimizes the linear loss:

```
*
* --- minimize linear loss
*
e_objMrio ..
v_objeMrio * sum( (considerMrio(rNat,curI(i),rNat1),mrioA,priceLevel1) $ mrioSplitFactors(rNat,i,rNat1,mrioA,priceLevel1),1)
=E=
*
* --- sum up pos/neg deviation relative to given MRIO trade flows
*
sum( (considerMrio(rNat,curI(i),rNat1),mrioA,priceLevel) $ mrioSplitFactors(rNat,i,rNat1,mrioA,priceLevel),
v_corP(rNat,i,rNat1,mrioA,priceLevel)+v_corN(rNat,i,rNat1,mrioA,priceLevel))
*
* --- pull newly introduce ones to zero (1 is added to the neg. dev = pulled against -1)
*
+ sum( (considerMrio(rNat,curI(i),rNat1),mrioA,priceLevelN) $ mrioSplitFactors(rNat,i,rNat1,mrioA,priceLevelN),
v_mult(rNat,i,rNat1,mrioA,priceLevelN))*p_slackWeight**(1/4)
*
* --- sum up pos/neg slacks against import totals (this should not happen!), high penalty
*
+ sum( (considerMrio(rNat,curI(i),rNat1),mrioA)
$ (mrioSplitFactors(rNat,i,rNat1,mrioA,"m") or mrioSplitFactors(rNat,i,rNat1,mrioA,"m1")),
(mrioSplitFactors(rNat,i,rNat1,mrioA,"cif")+mrioSplitFactors(rNat,i,rNat1,mrioA,"cif1"))
* (v_slack(rNat,i,rNat1,mrioA)+v_slack1(rNat,i,rNat1,mrioA)*p_slackWeight)*p_slackWeight)
+ sum( (considerMrio(rNat,curI(i),rNat1),v_slack2(rNat,i,rNat1))*p_slackWeight);
```

i.e. the positive and negative corrections are pulled towards zero while the two slacks receive a quite high value:

```
p_slackWeight=1E+6;
```

The problem is solved product by product in a loop:

```
loop(j $ sum( (considerMrio(rNat,j,rNat1)),1),
option kill=curI;curI(j) = yes;
m_mrio.optfile = 5;
m_mrio.limcol = %limcol%;
m_mrio.limrow = %limrow%;
$$iftheni.solprint "%solprint%"=="silent"
m_mrio.solprint = 2;
$$elseifi.solprint "%solprint%"=="off"
m_mrio.solprint = 0;
$$else.solprint
m_mrio.solprint = 1;
$$endif.solprint

if ( execerror,execError = 0);
p_mMrioStat(j,"nTries") = 1;

p_tarMaxDiff = 0.1;

solve m_mrio using LP minimizing v_objeMrio;
```

In case of infeasibilities:

(1) Missing split factors “cif1” are added assuming uniform tariffs:

```
mrioSplitFactors(rNat,j,rNat1,mrioA,"cif1") $ ((not mrioSplitFactors(rNat,j,rNat1,mrioA,"cif1")) $ mrioSplitFactors(rNat,j,rNat1,mrioA,"m1")
= mrioSplitFactors(rNat,j,rNat1,mrioA,"m1") * sam0(rnat,rnat1,j)/(sam0(rnat,rnat1,j)+imptxY0(rNat1,rNat,j));
mrioSplitFactors(rNat,j,rNat1,mrioA,"m1") $ ((not mrioSplitFactors(rNat,j,rNat1,mrioA,"m1")) $ mrioSplitFactors(rNat,j,rNat1,mrioA,"cif1")
= mrioSplitFactors(rNat,j,rNat1,mrioA,"cif1")*(sam0(rnat,rnat1,j)+imptxY0(rNat1,rNat,j))/sam0(rnat,rnat1,j);
```

(2) The maximal tariff differences for newly introduced flows is increased:

```
p_tarMaxDiff = 0.25;
```

If the problem is still infeasible, this maximal difference is increased to 50% and then 100%, in the latter case, also the upper limit on negative corrections is removed. As expected, normally the problem will introduce almost no changes to the original MRIO data.

The code collects information on the final successful solve of each problem, i.e. product:

```

',
p_mMrioStat(j,"maxTariff") = p_tarMaxDiff;
p_mMrioStat(j,"sumInfes") = m_mrio.suminfes;
p_mMrioStat(j,"numInfes") = m_mrio.numinfes;
p_mMrioStat(j,"obj") = v_objeMrio.1;
```

It also reports the cases where MRIO flows had to be invented in order to derive a consistent set of MRIO data matching the filter GTAP Data Base:

```
parameter p_new;
p_new(considerMrio(rNat,i,rNat1),mrioA,"m") $ mrioSplitFactors(rNat,i,rNat1,mrioA,"m1")
= mrioSplitFactors(rNat,i,rNat1,mrioA,"m1") * v_mult.1(rNat,i,rNat1,mrioA,"m1");

p_new(considerMrio(rNat,i,rNat1),mrioA,"cif") $ mrioSplitFactors(rNat,i,rNat1,mrioA,"cif1")
= mrioSplitFactors(rNat,i,rNat1,mrioA,"cif1") * v_mult.1(rNat,i,rNat1,mrioA,"cif1");

p_new(considerMrio(rNat,i,rNat1),mrioA,"tar") $ p_new(rNat,i,rNat1,mrioA,"cif")
= p_new(rNat,i,rNat1,mrioA,"m")/p_new(rNat,i,rNat1,mrioA,"cif");

p_new(considerMrio(rNat,i,rNat1),mrioA,"tarAvg") $ p_new(rNat,i,rNat1,mrioA,"cif")
= (sam0(rnat,rnat1,i)+imptxY0(rNat1,rNat,i))/sam0(rnat,rnat1,i);
option p_new:2:4:1;
display p_new;
```

The maximum of these cases is also reported:

```
parameter p_maxNew;
p_maxNew("abs") = smax( (considerMrio(rnat,i,rnat1),mrioA,priceLeveln), v_mult.1(rnat,i,rnat1,mrioA,priceLeveln));
p_maxNew("rel") = smax( (considerMrio(rnat,i,rnat1),mrioA) $ (v_mult.1(rnat,i,rnat1,mrioA,"cif1") $ mrioSplitFactors(rNat,i,rNat1,mrioA,"cif")),
v_mult.1(rnat,i,rnat1,mrioA,"cif1")/mrioSplitFactors(rNat,i,rNat1,mrioA,"cif"));
p_maxNew("rel") = max( p_maxNew("rel"),
smax( (considerMrio(rnat,i,rnat1),mrioA) $ (v_mult.1(rnat,i,rnat1,mrioA,"m1") $ mrioSplitFactors(rNat,i,rNat1,mrioA,"m")),
v_mult.1(rnat,i,rnat1,mrioA,"m1")/mrioSplitFactors(rNat,i,rNat1,mrioA,"m")));
display p_maxNew;
```

After the solution, transactions are defined based on the equation structure and very tiny ones deleted:

```
mrioSplitFactors(considerMrio(rNat,i,rNat1),mrioA,"m") $ (mrioSplitFactors(rNat,i,rNat1,mrioA,"m") or mrioSplitFactors(rNat,i,rNat1,mrioA,"m1"))
= mrioSplitFactors(rNat,i,rNat1,mrioA,"m") * (1+v_corP.1(rNat,i,rNat1,mrioA,"m") - v_corN.1(rNat,i,rNat1,mrioA,"m"))
+ mrioSplitFactors(rNat,i,rNat1,mrioA,"m1") * v_mult.1(rNat,i,rNat1,mrioA,"m1");
mrioSplitFactors(considerMrio(rNat,i,rNat1),mrioA,"m") $ (mrioSplitFactors(rNat,i,rNat1,mrioA,"m") le 1.E-10) = 0;
```

The same for the value at c.i.f.:

```
mrioSplitFactors(considerMrio(rNat,i,rNat1),mrioA,"cif") $ (mrioSplitFactors(rNat,i,rNat1,mrioA,"cif") or mrioSplitFactors(rNat,i,rNat1,mrioA,"cif1"))
= mrioSplitFactors(rNat,i,rNat1,mrioA,"cif") * (1+v_corP.1(rNat,i,rNat1,mrioA,"cif") - v_corN.1(rNat,i,rNat1,mrioA,"cif"))
+ mrioSplitFactors(rNat,i,rNat1,mrioA,"cif1") * v_mult.1(rNat,i,rNat1,mrioA,"cif1");
mrioSplitFactors(considerMrio(rNat,i,rNat1),mrioA,"cif") $ (mrioSplitFactors(rNat,i,rNat1,mrioA,"cif") le 1.E-10) = 0;
mrioSplitFactors(considerMrio(rNat,i,rNat1),mrioA,"m") $ (mrioSplitFactors(rNat,i,rNat1,mrioA,"cif") le 1.E-10) = 0;
```

Afterwards, potential cases with negative tariffs are reported:

```
* --- recalculate slack variables necessary to ensure that m (=c.i.f. + tariffs) >= cif
*
option kill=v_slack.1;
v_slack.1(considerMrio(rnat,i,rnat1,mrioA),mrioA) $ (mrioSplitFactors(rNat,i,rnat1,mrioA,"cif") gt mrioSplitFactors(rNat,i,rnat1,mrioA,"m"))
= mrioSplitFactors(rNat,i,rnat1,mrioA,"cif") - mrioSplitFactors(rNat,i,rnat1,mrioA,"m");

option v_slack:7:3:1;
v_slack.1(considerMrio(rnat,i,rnat1,mrioA),mrioA) $ (v_slack.1(rnat,i,rnat1,mrioA) le 1.E-8) = 0;
display v_slack.1;

parameter p_maxSlack;
p_maxSlack("abs") = smax( (considerMrio(rnat,i,rnat1,mrioA), v_slack.1(rnat,i,rnat1,mrioA));
p_maxSlack("rel") = smax( (considerMrio(rnat,i,rnat1,mrioA) $ (v_slack.1(rnat,i,rnat1,mrioA) $ mrioSplitFactors(rNat,i,rnat1,mrioA,"m")),
v_slack.1(rnat,i,rnat1,mrioA)/mrioSplitFactors(rNat,i,rnat1,mrioA,"m"))*100;
display p_maxSlack;
```

Finally, the information so far in the total value of the transaction is mapped into shares:

```
*
mrioSplitFactors(rNat,i,rnat1,mrioA,priceLevel) $ { mrioSplitFactors(rNat,i,rnat1,mrioA,priceLevel) $ sam0(rNat,rnat1,i))
= mrioSplitFactors(rNat,i,rnat1,mrioA,priceLevel)/sum(mrioA1, mrioSplitFactors(rNat,i,rnat1,mrioA1,priceLevel));
```

### 3.6.3 Model equations

The model equations in the MRIO extension of CGEBox are equivalent to the second level Armington equations in the standard model, but are now differentiated by the MRIO agents (total intermediate demand int, households hhsld, government gov and investment demand inv).

Bi-lateral import demand is expressed via the usual share equations on the RHS, and aggregated over the set mrio on the LHS:

```
e_xwAggMrio(rNat1,iIn(iMrio),rNat,ts) $ (xwFlag(rNat1,iMrio,rNat) $ rrComb(rNat1,rNat)) ..
*
(xw(rNat1,iMrio,rNat,ts)
  $$iftheni.fudg "%modulesArm_Les_Ces%"=="on"
  $ (p_ammwCom(rNat1,iMrio,rNat,ts) ge 0)
  + xwToFudge(rNat1,iMrio,rNat,ts) $ (p_ammwCom(rNat1,iMrio,rNat,ts) lt 0)
  $$endif.fudg
)/xw.scale(rNat1,iMrio,rNat,ts)

=e=

sum(mrioA $ p_ammwMrio(rNat1,iMrio,mrioA,rNat,ts),
  p_ammwMrio(rNat1,iMrio,mrioA,rNat,ts)
    * ( xmtMrio(rNat,iMrio,mrioA,ts)
      $$iftheni.armLesCes "%ModulesArm_Les_Ces%"=="on"
      $ (not arm_LES_CES(rnat,iMrio))
      + xmtUMrio(rNat,iMrio,mrioA,ts) $ arm_LES_CES(rnat,iMrio)
      $$endif.armLesCes
    )/xw.scale(rNat1,iMrio,rNat,ts)

  * [ ( ( pmtMrio(rNat,iMrio,mrioA,ts)
    $$iftheni.armLesCes "%ModulesArm_Les_Ces%"=="on"
    $ (not arm_LES_CES(rnat,iMrio))
    + pmtUMrio(rNat,iMrio,mrioA,ts) $ arm_LES_CES(rnat,iMrio)
    $$endif.armLesCes
  )

*
* -- not here the agent specific tariff rate
*
* /[(1 + imptxMrio(rNat1,iMrio,mrioA,rNat,ts) + mtax(rNat,iMrio,ts))

* Note WB: cif price is defined via macro %pmcif%, not a variable in model
*
* %pmcif%(rNat1,iMrio,rNat,ts) ]]**p_sigmax(rNat,iMrio)
* m_lambdam(rNat1,iMrio,rNat,ts)**(p_sigmax(rNat,iMrio) - 1)

$$iftheni.smallShare "%hasSmallShare%"=="on"
$ (not isSmallImpShare(rNat1,iMrio,rNat))
+ m_lambdam(rNat1,iMrio,rNat,ts) $ isSmallImpShare(rNat1,iMrio,rNat)
$$endif.smallShare
])

$$iftheni.fudg "%modulesArm_Les_Ces%"=="on"
+ [p_ammwCom(rNat1,iMrio,rNat,ts)/xw.scale(rNat1,iMrio,rNat,ts)] $ (p_ammwCom(rNat1,iMrio,rNat,ts) ge 0)
$$endif.fudg
;
```

In mathematical notation without considering the special cases (commitment terms, small shares):



$$xw_{r,r',i} = \sum_{mrIOA} amw_{r,r',i,mrIOA} xmt_{r,i,mrIOA} \left( \frac{pmt_{r,i,mrIOA}}{pmcif_{r,r',i}(1 + impt_{r,r',i,mrIOA})} \right)^{\sigma_{w,r,i}} \quad (48)$$

In order to consider the case of the CES with commitment terms, the import totals are either xmtMrio (standard case) or xmtMrio (if commitment terms are active). This additive demand terms are added with the last term in the equation. The main differences to the equation in GTAP standard are share parameters p\_amwMrio and tariffs imptxMrio differentiated by MRIO agent.

The total import demand which is distributed by the share equation above is aggregated over the individual agents linked to the MRIO agent, which is for intermediate input demand, only, different from a 1:1 relation:

```
*
* --- Aggregate over MRIO agents (= identity for hhshld, gov, sav)
*
e_xmtMrio(rNat, iMrio, mrIOA, ts) $ (xmtFlag(rNat, iMrio) $ sum(mrio_a(mrIOA, aa), p_alpham(rNat, iMrio, aa, ts)
    $ $if1 "%hsubr%"=="on" + sum(r_r(subr, rNat), p_alpham(subr, iMrio, aa, ts))
    ) $ iIn(iMrio)) ..

xmtMrio(rNat, iMrio, mrIOA, ts)/xmtMrio.scale(rNat, iMrio, mrIOA, ts)

=e= sum(mrio_a(mrIOA, aa), xm(rNat, iMrio, aa, ts) $ (p_alpham(rNat, iMrio, aa, ts) gt eps)
    $ $if1 "%hsubr%"=="on" + sum(r_r(subr, rNat) $ (p_alpham(subr, iMrio, aa, ts) gt eps), xm(subr, iMrio, aa, ts))
    )/xmtMrio.scale(rNat, iMrio, mrIOA, ts);
```

In mathematical notation:

$$xmt_{r,i,mrIOA} = \sum_{aa \in mrIOA} xm_{r,i,aa} \quad (49)$$

The average import price of each MRIO agent is defined via the dual price aggregator and needs to differentiate between the case with (pmtUMrio) and without commitment terms (pmtMrio):

```
*
* --- dual price aggregator (considers case with commitment terms)
*
e_pmtMrio(rNat, iIn(iMrio), MrIOA, ts) $ ( (pmtMrio.range(rNat, iMrio, mrIOA, ts) ne 0) $ pmtMrio.lo(rNat, iMrio, mrIOA, ts)
    $ xmtFlag(rNat, iMrio) $ (not p_pmtElas(rNat, iMrio))) ..

*
pmtMrio(rNat, iMrio, MrIOA, ts)
$ $iftheni.armLesCes "%ModulesArm_Les_Ces%"=="on"
$ (not arm_LES_CES(rNat, iMrio))
+ pmtUMrio(rNat, iMrio, MrIOA, ts) $ arm_LES_CES(rNat, iMrio)
$ $endif.armLesCes

=e=

*
* --- CES case with p_sigmax <> 1
*
{ [sum(rNat1 $ (p_amwMrio(rNat1, iMrio, mrIOA, rNat, ts)
$ $iftheni.smallShare "%hasSmallShare%"=="on"
$ (not isSmallImpShare(rNat1, iMrio, rNat))
$ $endif.smallShare
) , p_amwMrio(rNat1, iMrio, mrIOA, rNat, ts)
* ((1 + imptxMrio(rNat1, iMrio, mrIOA, rNat, ts) + mtax(rNat, iMrio, ts))
* pmcif(rNat1, iMrio, rNat, ts) ]/m_lambdam(rNat1, iMrio, rNat, ts))**(1-p_sigmax(rNat, iMrio))
** (1/(1-p_sigmax(rNat, iMrio))) * p_axPmtCDMrio(rNat, iMrio, mrIOA, ts) ]
$ $iftheni.smallShare "%hasSmallShare%"=="on"
$ sum(rNat1 $ (xwFlag(rNat1, iMrio, rNat) $ (not isSmallImpShare(rNat1, iMrio, rNat))), 1)
$ $endif.smallShare
} $ (p_sigmax(rNat, iMrio) ne 1)
```

In mathematical notation, without considering the special cases:

(50)

$$pmt_{r,i,mrioA} = \left( \sum_r amw_{r,r',i,mrioA} [pmcif_{r,r',i} (1 + imptx_{r,r',i,mrioA})]^{1-\sigma_{w,r,i}} \right)^{1/\sigma_{w,r,i}}$$

The agents' import prices are defined in the equation e\_pmpMrio:

```
*
* --- normally not in model, substituted out by macro
*
e_pmpMrio(rs(r), iMrio(iMrio), aa, ts) $ (p_alpham(r, iMrio, aa, ts) $ iIn(iMrio)) ..
  pmp(r, iMrio, aa, ts) =e= sum(mrio_A(mrioA, aa), pmtMrio(r, iMrio, mrioA, ts)) * (1 + mintx(r, iMrio, aa, ts) + itxshft(r, iMrio, ts));
```

The average tariff is defined by:

```
*
* --- MRIO extension: average import tariff definition one ach trade flow
*
e_imptxMrio(rNat1, iIn(iMrio), rsnat(rNat), ts) $ (xwFlag(rNat1, iMrio, rNat) $ (imptx.range(rNat1, iMrio, rNat, ts) ne 0) $ rrComb(rNat1, rNat) ) ..
(
  $iftheni.fudg "%modulesArm_Les_Ces%"=="on"
    (xw(rNat1, iMrio, rNat, ts) - p_amwCom(rNat1, iMrio, rNat, ts)) $ (p_amwCom(rNat1, iMrio, rNat, ts) ge 0)
    + xwToFudge(rNat1, iMrio, rNat, ts) $ (p_amwCom(rNat1, iMrio, rNat, ts) lt 0)
  $else.fudg
    xw(rNat1, iMrio, rNat, ts)
  $endif.fudg
) * (imptx(rNat1, iMrio, rNat, ts) + mtax(rNat, iMrio, ts)) / xw.scale(rNat1, iMrio, rNat, ts)

=e=
sum(mrioA $ imptxMrio(rNat1, iMrio, mrioA, rNat, ts),

  p_amwMrio(rNat1, iMrio, mrioA, rNat, ts) * (
    xmtMrio(rNat, iMrio, mrioA, ts)
    $iftheni.armLesCes "%modulesArm_Les_Ces%"=="on"
      $ (not arm_LES_CES(rnat, iMrio))
      + xmtUMrio(rNat, iMrio, mrioA, ts) $ arm_LES_CES(rnat, iMrio)
    $endif.armLesCes
  ) / xw.scale(rNat1, iMrio, rNat, ts) * (imptxMrio(rNat1, iMrio, mrioA, rNat, ts) + mtax(rNat, iMrio, ts))

  * [ ( (
    pmtMrio(rNat, iMrio, mrioA, ts)
    $iftheni.armLesCes "%modulesArm_Les_Ces%"=="on"
      $ (not arm_LES_CES(rnat, iMrio))
      + pmtUMrio(rnat, iMrio, mrioA, ts) $ arm_LES_CES(rnat, iMrio)
    $endif.armLesCes
  ) ) / ((1 + imptxMrio(rNat1, iMrio, mrioA, rNat, ts) + mtax(rNat, iMrio, ts))

  )

  Note WB: cif price is defined via macro %pmcif%, not a variable in model

  * %pmcif%(rNat1, iMrio, rNat, ts) ]] **p_sigmaw(rNat, iMrio)
  * m_lambdam(rNat1, iMrio, rNat, ts) ** (p_sigmaw(rNat, iMrio) - 1) }

  $iftheni.smallShare "%hasSmallShare%"=="on"
    $ (not isSmallImpShare(rNat1, iMrio, rNat))
    + m_lambdam(rNat1, iMrio, rNat, ts) $ isSmallImpShare(rNat1, iMrio, rNat)
  $endif.smallShare

  );
```

In mathematical notation:

(51)

$$xw_{r,r',i} imptx_{r,r',i} =$$

$$\sum_{mrioA} imptx_{r,r',i,mrioA} amw_{r,r',i,mrioA} xmt_{r,i,mrioA} \left( \frac{pmt_{r,i,mrioA}}{pmcif_{r,r',i} (1 + imptx_{r,r',i,mrioA})} \right)^{\sigma_{w,r,i}}$$

### 3.6.4 Graphical User Interface

#### Data base generation

In order to generate the split factors when the data base is set-up, the following checkbox must be activated:

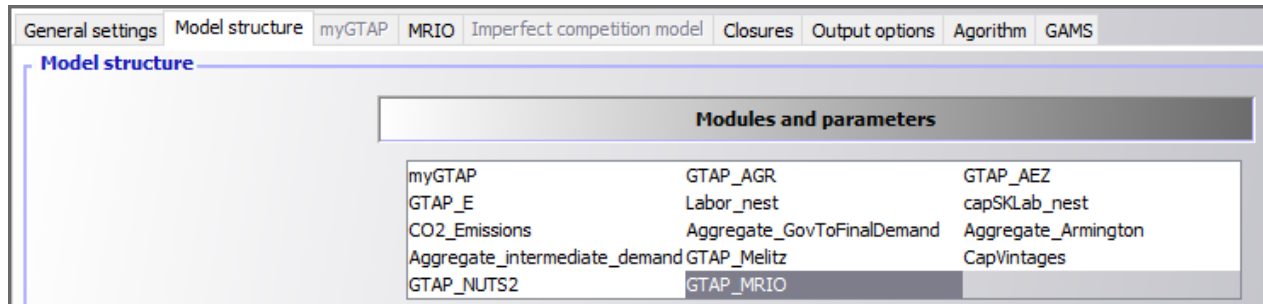
Generate MRIO split factors ☒

Fix import demand during split ☐

Furthermore, the user might activate the second check box to fix the import demands during the generation of the MRIO split factors.

### Simulation

When running the model, the MRIO module must be switched on:



In which case a tab allows selecting the products where the split is introduced:





### 3.7 Partial equilibrium closure

The partial equilibrium closure fixed prices and quantities of products not in the current solve. It also treats income as exogenous. Technically, that is implemented by three major approaches:

1. In “model\closures.gms”, the list of including products `iIn` and activities `aIn` which produce them is defined:

```
iIn(i) $ sum(sameas(i,pe_sel),1) = YES;
aIn(a) $ sum( (r,iIn), xFlag(r,a,iIn)) = YES;
```

Where `pe_sel` are the user-selected products in the partial equilibrium version of the model.

2. Variables referring to products or activities not in the model are fixed, for instance:

```
xTNest.fx(r,tNest,a,tSim) $ (xTNest.l(r,tNest,a,tSim) $ (not aIn(a))) = xTNest.l(r,tNest,a,tSim);
pTNest.fx(r,tNest,a,tSim) $ (xTNest.l(r,tNest,a,tSim) $ (not aIn(a))) = pTNest.l(r,tNest,a,tSim);
va.fx(r,a,tSim) $ (va.l(r,a,tSim) $ (not aIn(a))) = va.l(r,a,tSim);
nd.fx(r,a,tSim) $ (nd.l(r,a,tSim) $ (not aIn(a))) = nd.l(r,a,tSim);
```

3. Equations defining these variables are not included in the model, for instance:

```
vaeq(rs(r),aIn(a),ts(t)) $ vaFlag(r,a) ..
```

These fixing need also to be reflected during the pre-solve (`solve\presolve.gms`) and during calibration and initialization of the different modules.

The user should note that shock files should not update tax rates etc. belonging to products not in the current model as some initialization statements might update for instance prices based on tax rates under the shock.

### 3.8 Tariff Line Module

#### 3.8.1 Summary

The tariff line module allows to dis-aggregate selected bi-lateral trade flows to a lower detail and to explicitly model Tariff Rate Quotas. The user needs to provide the necessary data in a GAMS file in a specific format as detailed below, the file is selected from the GUI.

#### 3.8.2 Background

The GTAP data base offers bi-lateral protection and matching trade flow data based on MacMaps at the level of 57 respectively 65 GTAP sectors which are often further aggregated in model applications. That implies that information on protection measures at tariff line, which might also comprise complex instruments such as Tariff Rate Quotas (TRQs), needs to be aggregated to GTAP sector level and beyond. That might provoke aggregation bias (cf. Himics and Britz 2016<sup>28</sup>). Analysis especially of (potential) Free Trade Agreements requires often detail beyond the sectors of GTAP. The module for CGEBox described in here allows detail at tariff line for selected bi-lateral trade links and commodities while maintaining otherwise the chosen default trade structure for that sector (Armington or Armington plus CET, a combination with the Melitz module is currently not supported). If a MCP solver is available, multiple TRQs at bi-lateral tariff line can be introduced such that applied tariff rates become endogenous. At the same time, the structure provides a fully consistent aggregation from tariff line to commodity level.

The demand representation is based on a three-tier representation: (1) top level demand function such as the CDE in the GTAP standard, (2) First Armington nest between domestic sales and aggregate imports, (3) Bi-lateral Armington nest). A similar structure is found for supply transformation when the CET extensions is used: (1) first CET nest between domestic sales and aggregate exports, (2) CET nest between bi-lateral exports). Splitting up to tariff line level at the bi-lateral trade flow level as the lowest level of the demand system and supply transformation tree might seem odd at first look. One would perhaps rather like to split demand, for instance of dairy, first to tariff lines such as butter, cheese etc., next determine how much butter etc. is stemming from the home market and from imports, and finally how butter imports are composed from different importers. However, introducing the dis-aggregation and thus the new nest with tariff line information directly below the top level nests requires firstly introducing the tariff line split up for all bi-lateral trade flows of a region, and if a CET application is used, also for all its exporters and thus most probably for all regions in the model. That quickly renders the model quite large. Secondly, data on domestic sales at tariff line level must be available, data which are not generally available.

The proposed approach which rather split below the lowest level thus dramatically reduces data needs as it requires tariff line information on trade flows and protection data only for some bi-lateral links, namely those in the focus of the analysis. Data on domestic sales at tariff lines are not needed at all. Furthermore, it keeps the model at a manageable size even if some bi-lateral trade flows are depicted with rich dis-aggregated tariff line information. That is especially important if TRQs are modelled which requires a MCP solver which typically is slower compared to solving a simple non-linear equation system.

---

<sup>28</sup> Himics, M., Britz, W. (2016): Flexible and welfare-consistent tariff aggregation over exporter regions, *Economic Modelling* 53: 375-387

### 3.8.3 Approach

The user has to apply the following information:

- The set of tariff lines considered and their mapping to the GTAP sectors in the data base used by for model run
- Information on trade for selected bi-lateral trade flows at tariff line level. A bi-lateral trade flow for a commodity will only be depicted at tariff line level in the model when at least one non-zero flows for one tariff line is found. Otherwise, the bi-lateral commodity flow will continue to use the chosen default trade representation for that commodity without a disaggregation to tariff line.
- Matching applied tariff or TRQs with in- and out-of-quota rates.

Based on that information, the code will add the necessary equations and variables for the selected trade links, fit the data to the given benchmark and introduce matching parameters.

Note again that Melitz extension cannot be combined with the tariff line module for a product.

### 3.8.4 Model equations

The code for the tariff line sub-module can be found in the sub-directory “gams\tariffLines”. The equations and core definition are found in “tariffLines\_model.gms”.

The code of the model comprises four equations:

- Two equations which define the bi-lateral exports and imports at tariff line level
- And two matching dual price aggregators which define the bi-lateral export and import prices at the level of GTAP commodities at bi-lateral level.

The bilateral export supply equation is shown below:

```
e_xwstl(rNat,tl,rNat1,ts(t)) $ p_gwtl(rNat,tl,rNat1,t) ..
    xwstl(rNat,tl,rNat1,t) =e= sum[tl_i(tl,i), p_gwtl(rNat,tl,rNat1,t)* xws(rNat,i,rNat1,t)
    * (ptls(rNat,tl,rNat1,t)/pe(rNat,i,rNat1,t))**p_omegawtl(rNat,i) ];
```

In mathematical notation:

$$\begin{cases} XWTL_{r,tl,d}^s = \gamma_{r,tl,d}^w XW_{r,i,d}^s \left( \frac{PETL_{r,tl,d}}{PE_{r,i,d}} \right)^{\omega_{r,tl}^w} & \text{if } \omega_{r,tl}^w \neq \infty \\ PETL_{r,tl,d} = PE_{r,i,d} & \text{if } \omega_{r,tl}^w = \infty \end{cases} \quad (1)$$

The parameter `p_wgtl` depicts the share of the tariff line `tl` in total bi-lateral trade of commodity `i` shipped from exporter `rNat` to importer `rNat1`. The equation is hence only introduced if a share parameter was defined during calibration, i.e. for such bi-lateral trade flows where information on that tariff line was provided.

Note that the variable `pe`, the bi-lateral export price, is part of the standard model when non-infinite CET is used while the price at the tariff line level `ptls` is not. Note equally that in the standard layout, bi-lateral export and import quantities at commodity level `xw` are by definition equal as they are no

quality differences considered at that level without tariff line detail. That is no longer the case when the additional CES and CET nests at tariff line level are active such that the bi-lateral export quantity is now depicted by `xws` as a CET aggregate over tariff line. It is only identically to `xw` in the benchmark, in a simulation, the (implicit) non-linear aggregation based on the CES and CET will let the aggregate supply and demand quantities deviate.

The average import price is defined as follows:

```
*
* -- Related dual price aggregator
*
e_pe(rNat,i,rNat1,ts(t)) $ (gw(rNat,i,rNat1,t) $ sum[t1_i(t1,i),p_gwtl(rNat,t1,rNat1,t)]) ..
pe(rNat,i,rNat1,t) =e= sum[t1_i(t1,i), p_gwtl(rNat,t1,rNat1,t) * ptls(rNat,t1,rNat1,t)**(1+p_omegawtl(rNat,i)))]
** (1/(1+p_omegawtl(rNat,i))) ;
```

Where `gw` is the share parameter in the second level CET nest.

In mathematical notation:

$$\left\{ \begin{array}{l} PE_{r,i,d} = \left[ \sum_{t1 \in i} \gamma_{r,t1,d}^w PETL_{r,t1,d}^{1+\omega_{r,t1}^w} \right]^{1/(1+\omega_{r,t1}^w)} \text{ if } \omega_{r,t1}^w \neq \infty \\ XW_{r,i,d}^s = \sum_{t1} XWTL_{r,t1,d}^s \text{ if } \omega_{r,t1}^w = \infty \end{array} \right. \quad (2)$$

In the original layout, the export price `pe` is implicitly defined by the equation `e_pe` which defines `xw` from total export supply. To keep that equation untouched, it is replaced here by a variant which is used if NO tariff line equation is given. Beside the related \$ condition, it is identical to the equation in the standard model:

```
*
* -- That is the original pe equation from ENVISAGE, only active if there no tariff line detail for that product
*
e_peMod(rNat,iIn(i),rNat1,ts(t)) $ (m_xwFlag(rNat,i,rNat1) $ psFlag(rNat,i,"pe") $ rrComb(rNat,rNat1) $ (not sum[t1_i(t1,i),p_gwtl(rNat,t1,rNat1,t)]) ) ..
0 =e= ( xw(rNat,i,rNat1,t)/xw.scale(rNat,i,rNat1,t) - gw(rNat,i,rNat1,t)*xet(rNat,i,t)/xw.scale(rNat,i,rNat1,t)
* [1 $ isSmallExpShare(rNat,i,rNat1) + { (pe(rNat,i,rNat1,t)/pet(rNat,i,t))*omegaw(rNat,i) } $ (not isSmallExpShare(rNat,i,rNat1)) ] ) $ (omegaw(rNat,i)
$ (omegaw(rNat,i) ne inf)
+ (pe(rNat,i,rNat1,t) - pet(rNat,i,t) $ (omegaw(rNat,i) ne inf) - ps(rNat,i,t) $ (omegaw(rNat,i) eq inf)) $ (omegaw(rNat,i) eq inf) ;
```

The bi-lateral export supply at commodity level `xws` is defined identically, however neglects the case of small import shares or infinite transformation:

```
*
* -- same as above (infinite transformation case is however skipped, xw replaced by xws)
*
e_xws(rNat,iIn(i),rNat1,ts(t)) $ (m_xwFlag(rNat,i,rNat1) $ psFlag(rNat,i,"pe") $ rrComb(rNat,rNat1) $ sum[t1_i(t1,i),p_gwtl(rNat,t1,rNat1,t)]) ..
xws(rNat,i,rNat1,t)/xw.scale(rNat,i,rNat1,t)
=e= gw(rNat,i,rNat1,t)*xet(rNat,i,t)/xw.scale(rNat,i,rNat1,t)
* [1 $ isSmallExpShare(rNat,i,rNat1)
+ { (pe(rNat,i,rNat1,t)/pet(rNat,i,t))*omegaw(rNat,i) } $ (not isSmallExpShare(rNat,i,rNat1)) ] ;
```

The demand side comprises similar equations, first to define bi-lateral import demand at tariff line level `xwtl` (which by definition is equal to export supply):

```
*
* -- Bilateral import demand at tariff line level (= equal to bilateral export supply at tariff line level)
*
e_xwtl(rNat,t1,rNat1,ts(t)) $ p_amwtl(rNat,t1,rNat1,t) ..
xwtl(rNat,t1,rNat1,t) =e= sum[t1_i(t1,i),
p_amwtl(rNat,t1,rNat1,t)*xw(rNat,i,rNat1,t)
* (pm(rNat,i,rNat1,t)/(m_pmCifTl(rNat,t1,rNat1,t)
* (1 + imptXTl(rNat,t1,rNat1,t) + mtax(rNat1,i,t))/chipm(rNat,i,rNat1)) )**p_sigmaxwtl(rNat1,i))];
```

In mathematical equation:

$$XWTL_{s,tl,r}^d = \alpha_{s,tl,r}^w XW_{s,i,r}^d \left( \frac{PM_{s,i,r}}{PMTL_{s,tl,r}} \right)^{\sigma_{r,tl}^w} \quad (3)$$

The macro `m_pmcifTL` define the cif price, to which tariff line specific tariff `imptxTL` are added as well as potentially a general shifter for import taxation of product `i`. Note that `imptxTL` is defined as a variable to render it endogenous under a TRQ. The macro is based itself on a macro to define the f.o.b. price:

```
$macro m_pefobTL(r,tl,rp,t) [ ptls(r,tl,rp,t) * (1 + exptx(r,i,rp,t) + etax(r,i,t)) ]

$macro m_pmcifTL(r,tl,rp,t) { [ m_pefobTL(r,tl,rp,t)/lcu(r,t) \
+ (sum(m $ amgm(m,r,i,rp), amgm(m,r,i,rp)*ptmg(m,t)/lambdamg(m,r,i,rp,t))*tmarg(r,i,rp,t))\
$tmarg.l(r,i,rp,t)\
]*lcu(rp,t) }
```

Both macros are identically structured as the macros in the standard model. Note that export taxes are currently not differentiated at tariff line level and that the trade margins are also assumed to be identical for all tariff lines belonging to the same commodity.

The average bi-lateral import price `pm` for a commodity `i`, normally substituted out in the standard model with the `m_pmcif` macro, is in here explicitly defined as an aggregation over the tariff-line prices:

```
* -- Related dual CES price aggregator (average bi-lateral import price)

e_pm(rNat,i,rNat1,ts(t)) $ sum[tl_i(tl,i) $ p_amwtl(rNat,tl,rNat1,t),1] ..
pm(rNat,i,rNat1,t) =e=
sum[tl_i(tl,i) $ p_amwtl(rNat,tl,rNat1,t), p_amwtl(rNat,tl,rNat1,t)
* (m_pmcifTL(rNat,tl,rNat1,t)
* (1 + imptxTL(rNat,tl,rNat1,t) + mtax(rNat1,i,t)) )**(1-p_sigwawtl(rNat1,i))]
** (1/(1-p_sigwawtl(rNat1,i))) ;
```

In mathematical notation:

$$PM_{s,i,r} = \left[ \sum_{tl \in i} \alpha_{s,tl,r}^w PMTL_{s,tl,r}^{1-\sigma_{r,tl}^w} \right]^{1/(1-\sigma_{r,tl}^w)} \quad (4)$$

To keep the standard model simple, the modified xw equation `e_xwMod` which distinguishes between the case of dis-aggregation to tariff line or not is only comprised in the sub-module model:

```
*
* --- bilateral import demand at commodity level, note that the CEGBox default equation
* is the first block (not tariff line information) and uses pmcif plus tariffs,
* whereas the second block (tariff line information) uses the pm variable define by the dual
* price aggregator above
*
e_xwMod(rNat1,iIn(i),rNat,ts(t)) $ (xwFlag(rNat1,i,rNat) $ rrComb(rNat1,rNat) $ (not iMrio(i))) ..
xw(rNat1,i,rNat,t)/xw.scale(rNat1,i,rNat,t) =e= amw(rNat1,i,rNat,t)*xmt(rNat,i,t)/xw.scale(rNat1,i,rNat,t)

* ( [ (pmt(rNat,i,t)/[(1 + imptx(rNat1,i,rNat,t) + mtax(rNat,i,t))
*
* Note WB: cif price is defined via macro %pmcif%, not a variable in model
*
* %pmcif%(rNat1,i,rNat,t)/chipm(rNat1,i,rNat) ])**sigmaw(rNat,i)
* lambdam(rNat1,i,rNat,t)**(sigmaw(rNat,i) - 1) $ (not isSmallImpShare(rNat1,i,rNat))
+ lambdam(rNat1,i,rNat,t) $ isSmallImpShare(rNat1,i,rNat)
] $ (not sum[tl_i(tl,i) $ p_amwtl(rNat1,tl,rNat,t),1])
+ [ (pmt(rNat,i,t)*chipm(rNat1,i,rNat1)/pm(rNat1,i,rNat,t))**sigmaw(rNat,i)
* lambdam(rNat1,i,rNat,t)**(sigmaw(rNat,i) - 1) $ (not isSmallImpShare(rNat1,i,rNat))
+ lambdam(rNat1,i,rNat,t) $ isSmallImpShare(rNat1,i,rNat)
] $ sum[tl_i(tl,i) $ p_amwtl(rNat1,tl,rNat,t),1]
);
```

Finally, there is an equation for the TRQ mechanism:

```

*
* --- TRQ bound equation, paired with tariff
*
e_trq(rNat,tl,rNat1,ts(t)) $ (p_trq(rNat,tl,rNat1,t) $ rrComb(rNat1,rNat)) ..
    p_trq(rNat,tl,rNat1,t) =E= xwtl(rNat,tl,rNat1,t);

```

The TRQ equation clearly requires that the related tariff at tariff line level `imptxTL` is endogenous and not fixed. With a MCP solver, bounds on that tariff variable model can depict the regime switch from under-filled quota (= tariff at lower bound) to binding quota (= tariff between lower and upper bound, the tariff variable is equal to the per unit quota rent plus the in-quota tariff) and over-quota imports (= tariff at upper bound). Note that introducing TRQs with a CNS solver will only work for the cases where the quota is binding in the benchmark and the simulation, i.e. the implicate tariff is between its lower and upper bounds in the final solution.

For a more expanded discussion, see Jafari, Y., Himics, M., Britz, W., Beckman, J. (2021): It is all in the details: A bilateral approach for modelling trade agreements at the tariff line, *Canadian Journal of Agricultural Economics* 69(3): 415-442 and Jafari, Y., Britz, W., Guimbard, H., Beckman, J. (2021): Properly capturing tariff rate quotas for trade policy analysis in computable general equilibrium models, *Economic Modelling* 104(105620).

### 3.8.5 Entering the data

The user provided data are stored in files in the directory “gams\scen\tariffLine”. A trial implementation to model EU->CAN exports at tariff line level can be found there as “ceta.gms”. It fits any data base with at least EU28, USA and ROW as labels for regions and “mil” and “wht” as commodities.

Note that the information provided by such a file to module needs to be broken up in two blocks:

1. The **declarations**, i.e. the set of considered tariff lines and how they are mapped to the GTAP commodities in the current aggregation:

```

$iftheni.decl "%1"=="decl"

    set tl "Tariff line active in model"
        / cheese, butter,SMP,WMP,othDairy,
          baking_wht,feed_wht /;
    set tl_i(tl,i) "Link between tariff lines and GTAP commodities"
        / (cheese,butter,SMP,WMP,othDairy).mil-c
          (baking_wht,feed_wht).wht-c
        /;

```

Note that the set of tariff lines `tl` and there link to the commodity `tl_i` must be defined in block in a \$ifthen-else-endif structure as shown above.

2. The actual **data and parameter** to use

```

$else.decl
TABLE p_importTL (rp,tl,r) "Import Export Matrix of Cheese, importer first"
----- 19 line(s) not displayed
Can      EU_28      ROW
CAN.Baking_wht      0      5      47.97
EU_28.Baking_wht      0      0      24.46
ROW.Baking_wht      0.04      61.42      0
CAN.Feed_Wht      0      2      47.97
EU_28.Feed_Wht      0      0      24.46
ROW.Feed_Wht      0.04      61.42      0
;

*
* -- copy data to bi-lateral trade flow variable at tariff line,
*    will be used to calibrate parameters in tariff line model
*    and initialize further variables
*
xwtl.l(rNat,tl,rNat1,t0) = p_importTL(rNat1,tl,rNat);
*
* --- bi-lateral protection data
*
option kill=p_TRQ;
imptxTL.l(rNat,tl,rNat1,t0) $ p_importTL (rNat1,tl,rNat)
    = sum(tl_i(tl,i),impTx.l(rNat,i,rNat1,t0)) * uniform(0.5,1.5);
*
* --- set transformation and substitution elasticity at tariff line level
*
p_omegawtl(rNat,i) $ sum((tl_i(tl,i),rNat1), xwtl.l(rNat,tl,rNat1,"%t0%")) = 5;
p_sigmawtl(rNat1,i) $ sum((tl_i(tl,i),rNat), xwtl.l(rNat,tl,rNat1,"%t0%")) = 5;

$endif.decl

```

Note that the data above are invented and the tariff randomly drawn to provide a test framework for the module. The data to use need to be stored in “gams\scen\tariffline” and can from there selected via the GUI (see below).

### 3.8.6 Calibration

The bi-lateral trade flow and protection data are scaled to match the information in the global SAM in the file “gams\tariffLines\tariffLines\_cal.gms”

```

xwtl.l(rNat,tl,rNat1,t0) $ xwtl.l(rNat,tl,rNat1,t0)
    = xwtl.l(rNat,tl,rNat1,t0) * sum(tl_i(tl,i), xw.l(rNat,i,rNat1,t0) /sum(tl_ii(tl1,i), xwtl.l(rNat,tl1,rNat1,t0)));

imptxTL.l(rNat,tl,rNat1,t0) $ xwtl.l(rNat,tl,rNat1,t0)
    = imptxTL.l(rNat,tl,rNat1,t0) * sum(tl_i(tl,i), xw.l(rNat,i,rNat1,t0)*imptx.l(rNat,i,rNat1,t0)
        /sum(tl_ii(tl1,i), xwtl.l(rNat,tl1,rNat1,t0)*imptxTL.l(rNat,tl1,rNat1,t0)));

```

As export subsidies are assumed to be identical across tariff lines, the tariff line price at the benchmark is equal to the average export price:

```

ptls.l(rNat,tl,rNat1,t0) $ xwtl.l(rNat,tl,rNat1,t0)
    = sum(tl_i(tl,i),pe.l(rNat,i,rNat1,t0));

```

Such that the share parameters can be derived directly from the quantities:

```

p_gwtl(rNat,tl,rNat1,t0) $ xwtl.l(rNat,tl,rNat1,t0)
    = xwtl.l(rNat,tl,rNat1,t0) /sum(tl_i(tl,i),xws.l(rnat,i,rNat1,t0));

```

That is not the case for the importer side where tariff differentiation is assumed at tariff line level:



```

p_amwtl(rNat,t1,rNat1,t0) $ xwtl.1(rNat,t1,rNat1,t0)
= sum(tl_i(tl,i), xwtl.1(rNat,t1,rNat1,t0)/xw.1(rNat,i,rNat1,t0)

* (m_pmCifTl(rNat,t1,rNat1,t0)
    *(1 + imptxTl(rNat,t1,rNat1,t0) + mtax(rNat1,i,t0))
    /pm.1(rNat,i,rNat1,t0))*p_sigmaxTl(rNat1,i));

```

### 3.8.7 Post model processing

The code comprising the model equations also comprises a section for post-model processing:

```

p_results(rNat,"Q",tl,rNat1,"dom",%arg1%,"%version%") $ xwtl.1(rNat,t1,rNat1,%arg2%) = xwtl.1(rNat,t1,rNat1,%arg2%) * %scale%;
p_results(rNat,"P",tl,rNat1,"dom",%arg1%,"%version%") $ xwtl.1(rNat,t1,rNat1,%arg2%) = ptls.1(rNat,t1,rNat1,%arg2%);
p_results(rNat,"T",tl,rNat1,"dom",%arg1%,"%version%") $ xwtl.1(rNat,t1,rNat1,%arg2%) = sum(tl_i(tl,i),exptx.1(rNat,i,rNat1,%arg2%) + etax.1(rNat,i,%arg2%));

p_results(rNat,"TRQ",tl,rNat1,"imp",%arg1%,"%version%") $ p_trq(rNat1,t1,rNat,%arg2%) = p_trq(rNat1,t1,rNat,%arg2%);
p_results(rNat,"LowTar",tl,rNat1,"imp",%arg1%,"%version%") $ p_trq(rNat1,t1,rNat,%arg2%) = imptxTl.io(rNat1,t1,rNat,%arg2%) + eps;
p_results(rNat,"UppTar",tl,rNat1,"imp",%arg1%,"%version%") $ p_trq(rNat1,t1,rNat,%arg2%) = imptxTl.up(rNat1,t1,rNat,%arg2%);

p_results(rNat,"Q",tl,rNat1,"imp",%arg1%,"%version%") $ p_amwtl(rNat1,t1,rNat,%arg2%) = xwtl.1(rNat1,t1,rNat,%arg2%) * %scale%;
p_results(rNat,"P",tl,rNat1,"imp",%arg1%,"%version%") $ p_amwtl(rNat1,t1,rNat,%arg2%)
= sum(tl_i(tl,i),m_pmCifTl(rNat1,t1,rNat,%arg2%)*(1 + imptxTl(rNat1,t1,rNat,%arg2%) + mtax(rNat,i,%arg2%)));

p_results(rNat,"T",tl,rNat1,"imp",%arg1%,"%version%") $ xwtl.1(rNat1,t1,rNat,%arg2%)
= imptxTl(rNat1,t1,rNat,%arg2%) + sum(tl_i(tl,i),mtax(rNat,i,%arg2%));

p_results(rNat,"V",tl,rNat1,domImp,%arg1%,"%version%") $ p_results(rNat,"Q",tl,rNat1,"dom",%arg1%,"%version%")
= p_results(rNat,"Q",tl,rNat1,domImp,%arg1%,"%version%") * p_results(rNat,"P",tl,rNat1,"dom",%arg1%,"%version%");

p_results(rNat,"G",tl,rNat1,domImp,%arg1%,"%version%") $ p_results(rNat,"Q",tl,rNat1,domImp,%arg1%,"%version%")
= p_results(rNat,"Q",tl,rNat1,domImp,%arg1%,"%version%") * p_results(rNat,"T",tl,rNat1,domImp,%arg1%,"%version%");

p_results(rNat,"VG",tl,rNat1,domImp,%arg1%,"%version%") $ p_results(rNat,"Q",tl,rNat1,domImp,%arg1%,"%version%")
= p_results(rNat,"V",tl,rNat1,domImp,%arg1%,"%version%") + p_results(rNat,"G",tl,rNat1,domImp,%arg1%,"%version%");

p_results(rNat,"PT",tl,rNat1,domImp,%arg1%,"%version%") $ p_results(rNat,"VG",tl,rNat1,domImp,%arg1%,"%version%")
= p_results(rNat,"VG",tl,rNat1,domImp,%arg1%,"%version%") / p_results(rNat,"Q",tl,rNat1,domImp,%arg1%,"%version%");

```

### 3.8.8 Integration in the GUI

#### Model configuration

The module can as usually be switched on under the “Model structure” tab:

Note again that a non-default parameter file must be chosen which introduces transformation elasticities. It is not possible to use the model with an infinite transformation elasticity between export destinations.

The file with the set of considered tariff lines and related data can be chosen on the “Tariff lines” tab:

## Result exploitation

The results at tariff line level are comprised in the two tables “Exports, trade matrix, Tariff lines” and “Imports, trade matrix, Tariff lines”:

Trade	>	Exports, trade matrix
CO2	>	Exports, trade matrix, Tariff lines
No table		Exports, trade matrix flow map, detailed
Quantity		Exports, by product and destination
Price (net of taxes)		Imports, trade matrix
Value (net of taxes)		Imports, trade matrix, Tariff lines

A screen shot of a part of such a table is shown below:

Imports, trade matrix, Tariff lines [0]		Row1
Importer: Canada		
ceta		
EU_28		
cheese	Quantity	0.19
	Price (net of taxes)	1.05
	Value (net of taxes)	0.21
	Tax rate	0.27
	Tax income	0.05
	Price, tax inclusive	1.39
	Value, tax inclusive	0.27
	TRQ	0.19
	LowTar	0.00
	UppTar	1.44
butter	Quantity	0.01
	Price (net of taxes)	0.89
	Value (net of taxes)	
	Tax rate	1.91
	Tax income	0.02
	Price, tax inclusive	1.91
	Value, tax inclusive	0.02

### 3.9 CO2, Non-CO2 Accounting and air pollution emissions, energy use accounting

The CO2, Non-Co2, air pollution and energy use data are by now a default part of newer GTAP data base releases. They provide first additional results which can be exploited with the GUI. Second, the CO2 and Non-CO2 emissions can be used to define total CO2 equivalent emissions which can be taxed or become subject to an emission ceiling.

Parameters and indices AEZ NUTS2 Tariff lines myGTAP CO2 emissions MRIO ARM\_LES\_CES PE G-RDEM Imperfect competition

Modules and parameters			
myGTAP	GTAP_AGR	GTAP_AEZ	GTAP_E
CO2_Emissions	Non_CO2_Emissions	Air_Pollution	MTOE
Aggregate_GovToFinalDemand	Aggregate_Armington	Aggregate_firm_demand	GTAP_Melitz
ARM_LES_CES	SPE	GTAP_MRIO	TariffLines
CapVintages	GTAP_NUTS2	Parameters	Nestings
G-RDEM	GMIG		

#### 3.9.1 GHG emissions in CO2 equivalents

The CO2 accounting defines reported emissions in the model based on the following equations (see gams/co2eq/ceo2eq.gms):

```
*
* --- emissions at regional level
*
e_emisWOCarbon(rsNat,ts(tRun)) ..

emisWOCarbon(rsNat,tRun)/emis.scale(rsNat,tRun)
=E= [ sum( (i,aa) $ (m_alphad(rsNat,i,aa,tRun) $ emidCo2eq(rsNat,i,aa,tRun)), emidCo2eq(rsNat,i,aa,tRun) * xd(rsNat,i,aa,tRun))
+ sum( (i,aa) $ (m_alpham(rsNat,i,aa,tRun) $ emiCo2eq(rsNat,i,aa,tRun)), emiCo2eq(rsNat,i,aa,tRun) * xm(rsNat,i,aa,tRun))
+ sum( (a) $ (xpFlag(rsNat,a) $ emiaCo2eq(rsNat,a,tRun)), emiaCo2eq(rsNat,a,tRun) * m_xp(rsNat,a,tRun))
+ sum( (f,a) $ (xfFlag(rsNat,f,a) $ emifCo2eq(rsNat,f,a,tRun)), emifCo2eq(rsNat,f,a,tRun) * xf(rsNat,f,a,tRun))
]
/emis.scale(rsNat,tRun);
```

The emission factors emidCo2eq, defined per unit of USD and not a physical basis, are derived from the CO2 data base, and if, present, comprise also the ones from the non-CO2 emission data base. Only the latter also comprise factors per unit of factor use and per unit of output.

To these emissions, CO2 changes from carbon stock changes can be added as calculated by the AEZ module:

```
*
* --- emissions at regional level
*
e_emis(rsNat,ts(tRun)) ..

emis(rsNat,tRun)/emis.scale(rsNat,tRun) =E= [emisWOCarbon(rsNat,tRun)
$if1 "%modulesGTAP_AEZ%"=="on" - dCarbon(rsNat,tRun)*3.67 $ ( (not sameas(tRun,"%t0%"))
$ sum((r_r(rsNat,rNat),aez,land_Cover) $ aezFlag(rsNat,aez),p_carbon(rsNat,aez,land_cover,"carbon")))
]
/emis.scale(rsNat,tRun);
```

The reported emission can differ from the taxed ones:

```
*
* --- emissions at regional level
*
e_emisTaxed(rsNat,ts(tRun)) ..

emisTaxed(rsNat,tRun)/emis.scale(rsNat,tRun)
=E= [ sum( (i,aa) $ (m_alphad(rsNat,i,aa,tRun) $ emidCo2eqTaxed(rsNat,i,aa,tRun)), emidCo2eqTaxed(rsNat,i,aa,tRun) * xd(rsNat,i,aa,tRun))
+ sum( (i,aa) $ (m_alpham(rsNat,i,aa,tRun) $ emiCo2eqTaxed(rsNat,i,aa,tRun)), emiCo2eqTaxed(rsNat,i,aa,tRun) * xm(rsNat,i,aa,tRun))
+ sum( (a) $ (xpFlag(rsNat,a) $ emiaCo2eqTaxed(rsNat,a,tRun)), emiaCo2eqTaxed(rsNat,a,tRun) * m_xp(rsNat,a,tRun))
+ sum( (f,a) $ (xfFlag(rsNat,f,a) $ emifCo2eqTaxed(rsNat,f,a,tRun)), emifCo2eqTaxed(rsNat,f,a,tRun) * xf(rsNat,f,a,tRun))
]
$if1 "%modulesGTAP_AEZ%"=="on" - dCarbonTaxed(rsNat,tRun)* 3.67 $ ((not sameas(tRun,"%t0%")) $ sum(a $ (emispLnd.range(rsNat,a,tRun) ne 0),1))
]/emis.scale(rsNat,tRun);
```

Note that carbon stock changes can be taxed, only, if the GTAZ-AEZ module is active and the attached emission price is not fixed to zero.

The definition which emissions are taxed is determined by the selection of the user on the tab “CO2 emissions”:

An additional equation adds all global taxed emissions:

```
*
* --- adding up regional taxedemissions to global level
*
e_emisGblTaxed(ts) ..
    emisGblTaxed(ts)/sum(rNat,emis.scale(rNat,ts))
    =E= sum(rNat,emisTaxed(rNat,ts) $ rsNat(rNat) + emisTaxed.1(rNat,ts) $ (not rsNat(rNat)))/sum(rNat,emis.scale(rNat,ts));
```

The definition of the emission price under a cap can consider either nation-wide p\_permitCo2eq or global caps:

```
*
* --- regional emission ceiling equations, paired to emission price (=tax)
*
e_emisP(rsNat,ts) $ (emisp.range(rsNat,ts) ne 0) ..
    0 =E=
    *
    * --- case with regional ceiling
    *
    + ( (emisTaxed(rsNat,ts) - p_permitCo2eq(rsNat,ts))/p_permitCo2eq(rsNat,ts) ) $ ( p_permitCo2eq(rsNat,ts) $ (solveType eq MCP))
    + [ ( (emisTaxed(rsNat,ts) - p_permitCo2eq(rsNat,ts))/p_permitCo2eq(rsNat,ts)
    *   * emisp(rsNat,ts) ) ] $ (p_permitCo2eq(rsNat,ts) $ (solveType ne MCP))
    *
    * --- case with global ceiling: local and global emission price are equal
    *
    + [ emisp(rsNat,ts) - emispGblTaxed(ts) $ (card(rsNat) gt 1)
    *   - emispGbl.1(ts) $ (card(rsNat) eq 1)
    * ] $ p_permitCo2eqGbl(ts)
    *
    * --- neither, price is zero
    *
    + emisp(rsNat,ts) $ ( (not (p_permitCo2eq(rsNat,ts) or p_permitCo2eqGbl(ts) ) ));
```

The global cap is defined by the parameter p\_permitCo2eqGbl and defines the emission price:

```

*
* --- global emission ceiling equation, paired to global emission price (=tax)
*
e_emisPGbl(ts) $ (p_permitCo2eqGbl(ts) $ (card(rsNat) gt 1)) ..
    p_permitCo2eqGbl(ts)/sum(rNat,emis.scale(rNat,ts)) =E= emisGblTaxed(ts)/sum(rNat,emis.scale(rNat,ts));

```

Examples to define emission taxes or caps are found in the directory “scen/gams/scen/GHG”.

The changes in taxed carbon changes are calculated in the following equation:

```

e_dCarbonTaxed(rsNat(rNat),ts(tRun)) $ (not sameas(ts,"%t0%")) ..

dCarbonTaxed(rNat,tRun)
    /max(1,sum( (landCat_a(landCat,a),lnd) $ (p_carbon(rNat,"",landCat,"carbon") $ xfFlag(rNat,lnd,a)),
        0.01*p_landUse(rNat,"",a,"ha")*1.E-9
        *(p_carbon(rNat,"",unmanagedForest,"carbon")-p_carbon(rNat,"",landCat,"carbon"))))
    =E=
    sum( (landCat_a(landCat,a),lnd) $ (p_carbon(rNat,"",landCat,"carbon") $ xfFlag(rNat,lnd,a)),
        (xf(rNat,lnd,a,tRun)-xf.1(rNat,lnd,a,tRun-1))/xf.1(rNat,lnd,a,"%t0%"))
        *p_landUse(rNat,"",a,"ha")*1.E-6*gblScale
        *(-p_carbon(rNat,"",unmanagedForest,"carbon")+p_carbon(rNat,"",landCat,"carbon")))
    /max(1,sum( (landCat_a(landCat,a),lnd) $ (p_carbon(rNat,"",landCat,"carbon") $ xfFlag(rNat,lnd,a)),
        0.01*p_landUse(rNat,"",a,"ha")*1.E-9
        *(p_carbon(rNat,"",unmanagedForest,"carbon")-p_carbon(rNat,"",landCat,"carbon"))));

```

The calculation of carbon stock changes uses currently differences in the factor use of land  $xf(rNat,land,a,tRun)$  in the current simulation period to the previous one  $tRun-1$ , relative to benchmark factor use, multiplied with the related landuse in hectares at the benchmark, multiplied with the difference of carbon stock of the related land cover per ha to the carbon stock per ha of unmanaged forest. This assumes that the total changes in managed land stem from unmanaged forests. If total managed land in hectares does not change, i.e. if the changes in the  $xf$  converted in to hectares, then solely the differences in carbon stock content between the different land cover types define the carbon stock change.

In order to map this into a tax per unit of land used, the following equation is added:

```

e_emispLnd(rsNat(rNat),a,ts(tRun)) $ ( (not sameas(tRun,"%t0%")) $ sum(lnd,xfFlag(rNat,lnd,a)) $ (emispLnd.range(rNat,a,tRun) ne 0)) ..

emispLnd(rNat,a,tRun) * sum(lnd,xf(rNat,lnd,a,tRun)) * dCarbonTaxed(rNat,tRun)
/sum(lnd,xf.scale(rNat,lnd,a,tRun))
=E= sum(lnd,
    emisp(rNat,tRun)*3.67*((xf(rNat,lnd,a,tRun)-xf.1(rNat,lnd,a,tRun-1))/xf.1(rNat,lnd,a,"%t0%"))
    *p_landUse(rNat,"",a,"ha")*1.E-6*gblScale*(p_carbon(rNat,"",unmanagedForest,"carbon")
    -sum(landCat_a(landCat,a),p_carbon(rNat,"",landCat,"carbon"))))
    * (dCarbonTaxed(rNat,tRun) + dCarbon.1(rNat,tRun-1)-dCarbonTaxed.1(rNat,tRun-1))
/sum(lnd,xf.scale(rNat,lnd,a,tRun));

```

### 3.9.2 Marginal abatement curves for GHG emissions

Emission abatement based on the equations shown above is depicted in CGEBox by changes in the input and primary factor mix as emission factors are linked to these inputs and can be taxed. As discussed above, the model makes a distinction between reported emissions and taxed emissions which allows to exclude certain emission sources fully or partially from taxation or pricing. Substitution possibilities in the technologies, as for instance depicted by the structures and parameters taken from GTAP-E and GTAP-Power, allow firms to change their input mix according to the price signals which stem from the emission per unit of input use and carbon prices/taxes. Similarly, final demanders can have CES-sub-nests under the final demand system which allow adjustments in response to price changes stemming from carbon prices or taxes, again following GTAP-E respectively GTAP-Power, on top of the price responsiveness from the demand system.

Accordingly, depending on substitution possibilities in product and final demand, the different emission intensities of factors and products combined with carbon taxing or pricing let to structural change towards decarbonization, including changes in imports and exports might change. Generally, the differentiated updates of use taxes due to carbon taxing or pricing imply allocative losses by

forcing firms away from the cost minimal cost mix at the benchmark and consumers away from their preferred consumption bundles. In recursive-dynamic analysis, there are also potentially effects from lower capital accumulation arising from reduced income or changes in saving rates. The multi-regional model needs also to ensure BOP=BOT which is based in the standard layout of the GTAP model flexible trade balances interacting with the so-called global bank agent. Carbon taxes in this setup have an impact on foreign savings by changing implicitly expected returns to capital.

Abatement options currently not covered are those which reduce process emissions per unit of output, factor or intermediate input use from non-fossil fuels. These matter clearly mostly in case of production activities with considerably emissions per unit of output or input use (for instance, cement, paddy rice, ruminants). The taxation of these emission sources can imply massive changes in their per unit costs. Not considering cost effective abatement options in these cases will overexaggerate the impact of taxation. Equally, if demand for products stemming from activities with high process emission is price inelastic, emission levels might be overestimated.

Kiula et al. 2019 describe an approach where they define one economy wide additional production activity which serves as input into all others, based on Rutherford and Kiula 2013. Rutherford and Kiula 2013 describe also a so-called sector-specific approach where capital can substitute against abatable emissions. We follow their ideas in the following.

### ***Basic concept***

The basic concept of the proposed implementation of a MAC module is summarized as follows:

1. It is assumed that (points on) Marginal Abatement Cost Curves (MACC)s from bottom-up studies provide the backbone of the module, used to define the abated emissions per unit of input or factor use or per unit of output, and related cost at distinct points defined by carbon tax levels.
2. Total emissions are calculated from process emissions minus abated emissions.
3. The costs of abatement are depicted by a Leontief bundle of updated intermediate input demands per unit of abated emissions, financed from the savings account. If a separate government account is available, it is assumed that the government spending on savings is expanded to cover (a larger part of the abatement costs).
4. The integration in the overall model is based on an additional Armington agent which demands products to realize the abatement technologies.

The layout above will imply that the realization of abatement technologies leads to a new source of final demand which is linked to the emission sources of the activities, but not added to their production function nesting. This helps to keep impacts on the overall modeling structure clean and also avoids constructions such as “unproductive capital”. Instead, abatement technologies require specific inputs (for instance, construction, machinery or educational services) which need to be produced or imported, and thus compete with the given intermediate and final demand in the economy. This also allows to track in a transparent manner the impacts of realizing the abatement options.

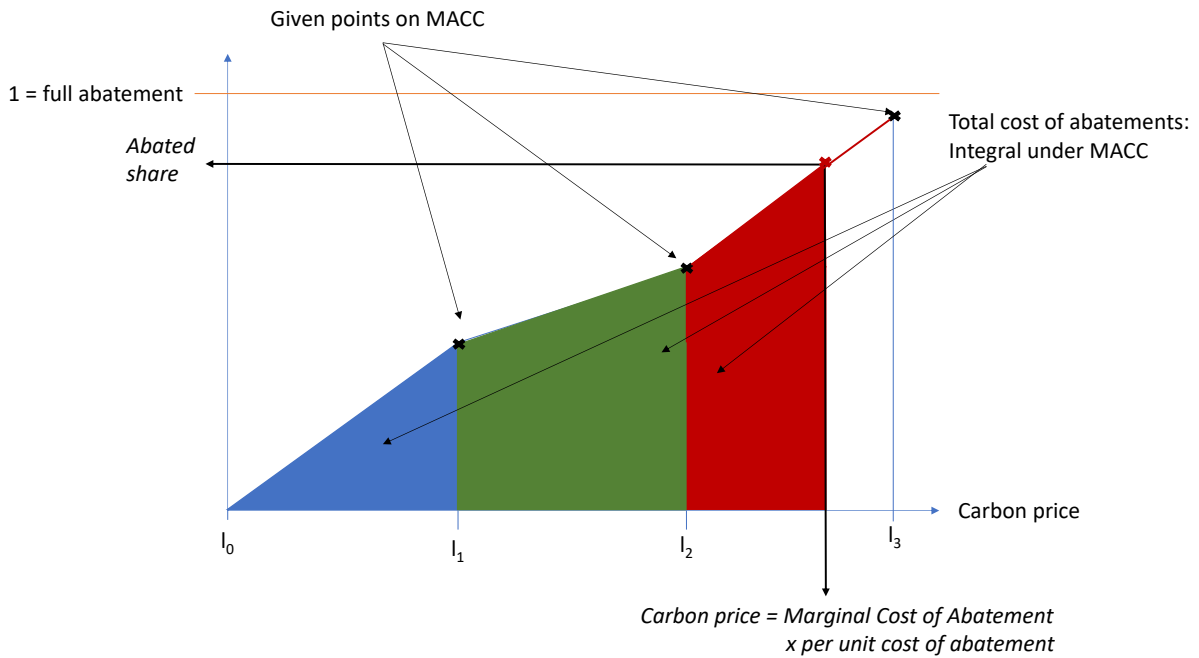
The implicit assumption here are that the abatement efforts need to be repeated in each period. This implies that, for instance, a one-time capital demand for an abatement technology reported in literature needs to be mapped into the annualized (investment) demand for specific products by the abatement agent.

**Equation structure**

The key variable is  $v\_redEmiaCo2eq$  which expresses the relative reduction in process emissions for a production activity  $a$  and region  $r$ . It can therefore take on values between 0 (= technology not used) and 1 (= technology removes emissions completely). It is driven by the carbon price  $emisp$ , corrected for the price index of abatement cost  $pindMaccs$ :

$$v\_redEmiaCo2eq_{r,a} = f\left(\frac{emisp_r}{pindMaccs_r}\right) \quad (52)$$

In the current code realization, no differences between different emitting sources (factors, intermediate, output levels) or emitted gases (NH4, NOx, FGAS) exists. The function in (52) is a step-wise linearization of points on MACC curves, see Figure 6. It is important to understand that the assumed abatement technology is not returns-to-scale, such that average costs of abatement differ from marginal ones.



**Figure 6: Step-wise linearization of points on a MACC**

Let  $emisRed$  define the set of points on the marginal abatement curve for activity  $a$  and region  $r$ , assuming that the same cost points  $l$  are used for all activities and regions while related abatement levels can differ. These points are ordered by increased level of abatement. In the example of Figure 6, three such points  $l_1$  until  $l_3$  are depicted by black crosses.

The related equations is shown next:

$$v\_redEmiaCo2eq_{r,a} = \sum_l \max[0, \min(cost_l, \frac{emisp_r}{pindMaccs_r}) - cost_{l-1}] \frac{emisRed_{r,a,l} - emisRed_{r,a,l-1}}{cost_l - cost_{l-1}} \quad (53)$$

In the expression above, the emission price is corrected for the price index of abatement services.



In order to calculate the level of abated emissions as a share of the total ones at a given carbon price  $emisp$ , we need to up add over the given points on the MACC. For the example above, up to point  $l_2$ , the summation will add up the total differences in emission abatement as the term in the max operator will be equivalent to the denominator  $cost_l - cost_{l-1}$ . As the carbon price is below point  $l_3$ , only a share of the term  $\frac{emisRed_{r,a,l} - emisRed_{r,a,l-1}}{cost_l - cost_{l-1}}$  will be applied for the additional emission reduction achieved between  $l_2$  and  $l_3$  as it is multiplied with  $\max\left[0, \min\left(cost_l, \frac{emisp_r}{pindMaccs_r}\right) - cost_{l-1}\right]$ . Accordingly, the expression will work as an interpolation which selects the point with the red cross in figure above, by a share of the additional emission abatement achieved from  $l_2$  to  $l_3$  according to given carbon price.

The total cost of abatement services as the integral under the MACC curve are calculated accordingly as shown in the formula below. The first term multiplied  $\frac{1}{2}$  would be equivalent to the blue triangle for the first step. For follow up steps, the reached emission reduction from the previous step must be taken into account as well:

$$qAbat_{r,a} = \sum_l \max\left[0, \min\left(cost_l, \frac{emisp_r}{pindMaccs_r}\right) - cost_{l-1}\right] \left(\frac{1}{2} \frac{emisRed_{r,a,l} - emisRed_{r,a,l-1}}{cost_l - cost_{l-1}} + emisRed_{r,a,l-1}\right) emis_{r,a} \quad (54)$$

In (54),  $emis$  are the total Co2 equivalent process emissions linked to activity  $a$  in region  $r$ , i.e. emissions per unit of factor use times factor use, emissions per unit of intermediate use times intermediate input use – differentiated by domestic and imported use - and emissions per process level times production output:

$$emis_{r,a} = \sum_{emits} \left( \begin{aligned} &xf_{r,f,a} emifCo2EqTaxed_{r,f,a} \forall f = emits \\ &+ xd_{r,i,a} emidCo2EqTaxed_{r,i,a} \forall i = emits \\ &+ xm_{r,i,a} emiiCo2EqTaxed_{r,i,a} \forall i = emits \\ &+ xp_{r,a} emiaCo2EqTaxed_{r,a} \forall o = emits \end{aligned} \right) \quad (55)$$

The total cost of abatement must be exhausted in a CGE by demand for products and services. To do so, a new final demand agent  $maccs$  is defined which minimizes abatement costs under a Leontief technology. Its demand for products and services  $i$  is defined by summing up over the total abatement services demand of the production activities  $a$  times a Leontief coefficient  $p\_maccsShare_{r,i,a}$ :

$$xa_{r,i,"maccs"} = \sum_a qAbat_{r,a} p\_maccsShare_{r,i,a} \quad (56)$$

The total value of these demand is:

$$valMaccs_r = \sum_a xa_{r,i,"maccs"} pa_{r,i,"maccs"} \quad (57)$$

This allows to define a price index for the  $maccs$  agent:

$$pindMaccs_r = \frac{\sum_a x a_{r,i,"maccs"} p a_{r,i,"maccs"}}{\sum_a x a_{r,i,"maccs"}} \quad (58)$$

This price index is used in two equations above.

### Abatement curves

The functional forms of MACCs (52) are not defined above. **Fehler! Verweisquelle konnte nicht gefunden werden.** depicts a typical case. A technology which only remove a part of the specific emissions and reach this abatement potential at a certain price. As the min and max operator used above are not differentiable, the Veelken-Ulbrich smoothed min function available from GAMS is used instead.

The layout defined above would support multiple abatement technologies which differentiated input demands as shown above.

Finding the data on changes in which inputs are underlying specific MACCs can probably prove quite demanding or even impossible. Possible fallbacks options:

- Using the economic wide composition of investment demands instead.
- Defining some broad types (such as 80% based on new structures and 20% training) etc.

### Technical test

We set up a test framework where we define the input by the emission reduction achieved by a set of emission prices or CO2 tax rates per unit of CO2 equivalent:

```
*
set emisPLlevels "Potential steps on abatement cost curve"
/ ep0,ep1,ep5,ep10,ep50,ep100,ep500,ep1000 /;
alias(emisPLlevels,emisPLlevels1);

parameter p_emisPLlevels "CO2 tax levels linked to the steps "
/ ep0 1.E-6,ep1 1,ep5 5,ep10 10,ep50 50,ep100 100,ep500 500,ep1000 1000/;
```

The prices are deemed to cover the relevant range up to a tax of 1000 USD per unit. The emission reduction potential of the different technologies must be defined at these prices:

```
table p_emisRed(rWor,aE,emisPLlevels) "Relative reduction in process emissions reduction"

               ep1      ep5      ep10      ep50      ep100      ep500      ep1000
wor.nfm-a      1.E-6    1.E-4                0.5      0.75
wor.ctl-a      1.E-6    1.E-4                0.10      0.5
;
```

In the next step, we linearly interpolate between given tax points to define relative abatement levels for missing ones:

```

*
* --- Set (almost) ( zero reduction at tax of zero
*
p_emisRed(rWor,ae,"ep0") $ sum(emisPLevels,p_emisRed(rWor,ae,emisPLevels))
= 1.E-10;

*
* --- make sure that last step is always set
*
p_emisRed(rWor,ae,"ep1000") $ (sum(emisPLevels,p_emisRed(rWor,ae,emisPLevels))
$ (not p_emisRed(rWor,ae,"ep1000")))
= smax(emisPLevels,p_emisRed(rWor,ae,emisPLevels));

*
* --- interpolate missing steps
*
loop(emisPLevels,
    p_emisRed(rWor,ae,emisPLevels) $ (not p_emisRed(rWor,ae,emisPLevels))
    = [smin(emisPLevels1 $ ((emisPLevels1.pos gt emisPLevels.pos) $ p_emisRed(rWor,ae,emisPLevels1)), p_emisRed(rWor,ae,emisPLevels1))
      - p_emisRed(rWor,ae,emisPLevels-1)]
      / [smin(emisPLevels1 $ ((emisPLevels1.pos gt emisPLevels.pos) $ p_emisRed(rWor,ae,emisPLevels1)),
        p_emisPLevels(emisPLevels1)) - p_emisPLevels(emisPLevels-1)]
      * [p_emisPLevels(emisPLevels) - p_emisPLevels(emisPLevels-1)] + p_emisRed(rWor,ae,emisPLevels-1);

);

```

The outcome of these process is a completely filled table:

```

-----      80 PARAMETER p_emisRed Relative reduction in process emissions reduction

                ep0          ep1          ep5          ep10          ep50          ep100          ep500          ep1000
wor.nfm-a 1.00000E-10 1.000000E-6 1.000000E-4          0.026          0.237          0.500          0.750          0.750
wor.ctl-a 1.00000E-10 1.000000E-6 1.000000E-4          0.005          0.047          0.100          0.278          0.500

```

Next, we set up a dummy partial equilibrium model for one region and the two activities shown above. The model has downward sloping demand curve  $e\_dem$  which defines the level of production output (supply=demand)  $v\_xp$ . Demand is driven by the per unit production cost  $v\_cost$  which are equal to unity if no CO<sub>2</sub> tax is active. The cost function  $e\_cost$  subtract the base process emissions per unit  $p\_emisCoeff$  times the CO<sub>2</sub>eq price  $emisP$ , taking into account that per unit process emission drop with increasing carbon prices. The change in per unit emissions is also reflected in the adding up of emissions to regional totals in the equation  $e\_emis$ .

The most demanding part from a numerical perspective is the step-wise function for the emission reduction per unit of output in  $e\_emisRed$ . The term

```

* ( p_emisRed(rWor,ae,emisPLevels)-p_emisRed(rWor,ae,emisPLevels-1))
/ ( p_emisPLevels(emisPLevels)-p_emisPLevels(emisPLevels-1))

```

Defines the changes in relative emission reductions per unit of change in the emission price between two neighboring point on the MACC curve. This term must be multiplied with

```

max(0,min(v_emisP(rWor),p_emisPLevels(emisPLevels)) - p_emisPLevels(emisPLevels-1))

```

If the current carbon tax  $v\_emisP$  exceeds the carbon tax level  $p\_emisPLevels$  at point  $emisPLevels$ , the full change will be applied. The minimum operator will ensure that not more than the differences to the next lower point will be applied. For cases where the current carbon tax does not reach level  $p\_emisPLevels(emisPLevels-1)$ , the max operator is needed.

The reduction level is hence defined as:

```

sum[emisPLevels $ p_emisRed(rWor,ae,emisPLevels-1),
{max(0,
  min(p_check(rWor,aE,draws,"tax"),p_emisPLevels(emisPLevels)) - p_emisPLevels(emisPLevels-1))}

  * ( p_emisRed(rWor,ae,emisPLevels)-p_emisRed(rWor,ae,emisPLevels-1))
  / ( p_emisPLevels(emisPLevels)-p_emisPLevels(emisPLevels-1))];

```

The min and max operators are not differentiable. We therefore use the ncpVUSin function from GAMS as smooth approximation of min:

ncpVUSin(r,s[,MU])	<p><b>NCP Veelen-Ulbrich: smoothed min(r,s)</b></p> $\text{result} = \begin{cases} \frac{(r+s- t )}{2} & \text{if }  t  \geq \mu \geq 0 \\ \frac{(r+s-(\frac{2\mu}{\pi} \sin(\frac{\pi}{2\mu} + \frac{3\pi}{2}) + \mu))}{2} & \text{otherwise} \end{cases}$ <p>where <math>t = r - s</math>. Default <math>MU = 0</math>.</p>
--------------------	---

Which leads to the term

```

sum[emisPLevels $ p_emisRed(rWor,ae,emisPLevels-1),
{-ncpVUSIN(0,
  -ncpVUSIN(p_check(rWor,aE,draws,"tax"),p_emisPLevels(emisPLevels),1.E-3) + p_emisPLevels(emisPLevels-1),1.E-3)}

  * ( p_emisRed(rWor,ae,emisPLevels)-p_emisRed(rWor,ae,emisPLevels-1))
  / ( p_emisPLevels(emisPLevels)-p_emisPLevels(emisPLevels-1))];

```

This term is introduced in the model equation *e\_emisRed* below. At MU=1.E-3, no measurable error in the approximation occurs of the relevance range of carbon taxes [0,1000].

```

variable v_emisRed(rWor,aE) "Relative CO2 price driven reduction in process emissions"
v_emis(rWor) "Total regional emissions"
v_emisp(rWor) "CO2 price"
v_xp(rWor,aE) "Supply (= demand in single region test model"
v_cost(rWor,aE) "Per unit production cost";

equation e_emisRed(rWor,aE),e_emis(rWor),e_dem(rWor,aE),e_cost(rWor,aE);

parameter p_emisCoeff(rWor,aE);

p_emisCoeff(rWor,AE) = 0.001;

e_dem(rWor,aE) ..

  v_xp(rWor,aE) =E= 100 - 50 * v_cost(rWor,aE);

e_cost(rWor,aE) ..

  v_cost(rWor,aE) =E= 1 + p_emisCoeff(rWor,aE) * (1 - v_emisRed(rWor,aE)) * v_emisp(rWor);

e_emis(rWor) ..

  v_emis(rWor) =E= sum(aE, v_xp(rWor,aE) * p_emisCoeff(rWor,aE) * (1 - v_emisRed(rWor,aE)));

e_emisRed(rWor,aE) $ (v_emisRed.range(rWor,aE) ne 0) ..

  v_emisRed(rWor,aE) =E=

    sum[emisPLevels $ p_emisRed(rWor,ae,emisPLevels-1),
    {-ncpVUSIN(0,
      -ncpVUSIN(v_emisp(rWor),p_emisPLevels(emisPLevels),1.E-3) + p_emisPLevels(emisPLevels-1),1.E-3)}

      * ( p_emisRed(rWor,ae,emisPLevels)-p_emisRed(rWor,ae,emisPLevels-1))
      / ( p_emisPLevels(emisPLevels)-p_emisPLevels(emisPLevels-1))];

model m_emisRed / e_emisRed,e_emis,e_dem,e_cost /;

```

We test the approach by first solving the model with a fixed carbon prices of 1 and emission coefficients of 1.E-3:

```

model m_emisRed / e_emisRed,e_emis,e_dem,e_cost /;
m_emisRed.solveLink = 5;
m_emisRed.limCol = 0;
m_emisRed.limRow = 0;
m_emisRed.holdfixed = 1;

p_emisCoeff(rWor,AE) = 0.001;
v_emisRed.lo(rWor,AE) = 0;
v_emisRed.up(rWor,AE) = 1;
v_emisP.fx(rWor) = 0;
option CNS = CONOPT4;
solve m_emisRed using CNS;

```

Afterwards, we fixing the emissions to a ceiling (from experiment *r0* onwards) which is equal to 90% of the emissions under the previous iteration and render the carbon tax endogenous.

```

set redStep /r0*r10 /;
parameter p_res;

loop(redStep,
  v_emisP.up(rWor) = +inf;
  v_emisP.lo(rWor) = 0;
  v_emis.fx(rWor) = v_emis.l(rWor);
  solve m_emisRed using CNS;
  p_res(rWor,ae,"xp",redStep) = v_xp.l(rwor,ae);
  p_res(rWor,ae,"emis",redStep) = v_xp.l(rWor,ae) * p_emisCoeff(rWor,ae) * (1 - v_emisRed.l(rWor,ae));
  p_res(rWor,ae,"emisRed",redStep) = v_emisRed.l(rWor,ae);
  p_res(rWor,"","emis",redStep) = v_emis.l(rWor);
  p_res(rWor,"","emisP",redStep) = v_emisP.l(rWor);
  p_res(rWor,ae,"cost",redStep) = 1 + p_emisCoeff(rWor,ae) * (1 - v_emisRed.l(rWor,ae)) * v_emisP.l(rWor);
  p_res(rWor,"","solve",redStep) = m_emisRed.modelStat;
  v_emis.l(rWor) = v_emis.l(rWor) * 0.90;
  v_emisp.l(rWor) = v_emisp.l(rWor) + 10;
);

```

The first solve was however infeasible if not starting value for the emission prices is set as shown above.

This leads to the following results:

	r0	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
nfm-a.xp	50.000	48.740	48.022	47.622	47.498	46.314	45.229	44.438	43.931	43.695	43.711
nfm-a.emis	0.050	0.043	0.036	0.030	0.024	0.021	0.019	0.017	0.015	0.013	0.011
nfm-a.emisRed		0.125	0.252	0.373	0.486	0.537	0.579	0.621	0.662	0.702	0.741
nfm-a.cost	1.000	1.025	1.040	1.048	1.050	1.074	1.095	1.111	1.121	1.126	1.126
ctl-a.xp	50.000	48.595	47.488	46.492	45.609	43.044	40.434	38.058	35.914	33.999	32.311
ctl-a.emis	0.050	0.047	0.045	0.043	0.041	0.038	0.034	0.031	0.028	0.026	0.024
ctl-a.emisRed		0.025	0.050	0.075	0.097	0.126	0.156	0.186	0.215	0.244	0.271
ctl-a.cost	1.000	1.028	1.050	1.070	1.088	1.139	1.191	1.239	1.282	1.320	1.354
.emis	0.100	0.090	0.081	0.073	0.066	0.059	0.053	0.048	0.043	0.039	0.035
.emisP		28.817	52.913	75.806	97.271	159.232	226.764	293.385	358.909	423.069	485.508
.solve	16.000	16.000	16.000	16.000	16.000	16.000	16.000	16.000	16.000	16.000	16.000

As expected, with stricter emission ceilings, production quantities have to adjust. In order to reduce emissions (*emis*) to 35% of their benchmark levels, a tax of almost 500 USD is required (*emisp*).

The reduction of the cattle activity “ctl-a” is higher despite an equal base emission coefficient of 0.001. This reflects that abatement possibilities in cattle are more limited. Whereas the non-ferrous metal activity “nfm-a” is assumed to abate up to 75% of its per unit emissions, reached at an emission price of 500 USD per ton CO<sub>2</sub>eq, the maximal abatement level for cattle is 50% and requires a tax of 1000 USD. At a 500 USD tax rate, the cattle sector will reduce its per unit emissions by around 28%, as shown above. Accordingly, the per unit cost of cattle production including the emission taxes reach up to 1.35 at the lowest ceiling, compared to 1.12 for non-ferrous metals. As output levels at the benchmark and the slopes of the demand curves are assumed identical, the cattle sector shrinks more in absolute and relative terms, from 50 to around 33 units compared to non-ferrous metals where output is reduced from 50 to about 43 units.

The test model underlines that the approach is feasible. It was hence chosen the implementation in CGEbox as described next.

### Integration into CGEBox

The integration into the CGE model CGEBox requires to map the equations above into the existing structure of the CGE. The code is added to the CO2 and Non-Co2 emission module (*co2eq/coe2eq.gms*). It comprises four main components:

1. New equations which define the emission reduction, the related demand for intermediate and their value and the link into the overall CGE.
2. An initialization block where the necessary parameters and start values for variables are defined.
3. Controls on the GUI
4. And the integration into the reporting part.

### Equations

Based on concept, there are three new equations in the systems. The first defines the relative reduction in process emissions  $v\_redEmiaCo2Eq$  as a smoothed linear step-function of the emission price  $emisP$  as discussed above, depicting the abatement level according to the MACC curve:

```
e_redEmiaCo2eq(rsNat(rNat),a,ts) $ ((v_redEmiaCo2eq.range(rsNat,a,ts) ne 0) $ p_emisRed(rsNat,a,"ep5000")) ..
v_redEmiaCo2eq(rsNat,a,ts) =E=
sum[emisPLevels $ p_emisRed(rsNat,a,emisPLevels-1),

{-ncpVUSIN(0,
-ncpVUSIN(emisP(rsNat,ts)/pindMaccs(rsNat,ts),p_emisPLevels(emisPLevels),1.E-3) + p_emisPLevels(emisPLevels-1),1.E-3)}

* ( p_emisRed(rsNat,a,emisPLevels)-p_emisRed(rsNat,a,emisPLevels-1))
/ ( p_emisPLevels(emisPLevels)-p_emisPLevels(emisPLevels-1))
];
```

The equation is introduced for the combination of region  $rsNat$ , activity  $a$  and solution period  $ts$  where a process emission coefficient  $emiaCo2eqTaxed$  is given and an emission reduction coefficient  $p\_emisRed$  is defined. The initialization code ensures that all points on the MACC curve are present. It is therefore sufficient to test for the highest price level, only.

The RHS comprises the step-wise linear function.

A second equation defines the demand for emission services for each production activity  $qAbat$  which consider (1) emission per unit of output, (2) emissions per unit of factor use and (3) emission from the use of domestic or imported intermediates. As these emission coefficients can also comprise emissions from the burning of fossil fuels where emission coefficients are seen as physical constants and per unit emissions as not abatable by technical measures, the parameter  $p\_nCo2ShareD$  and  $p\_nCo2ShareI$  capture the share of non-Co2 emissions at the benchmark which are subject to abatement:

```

*
* --- total demand for emission services
*
e_qAbat(rsNat(rnat),a,ts(tRun)) $ ((v_redEmiaCo2eq.range(rsNat,a,ts) ne 0) $ p_emisRed(rsNat,a,"ep5000")) ..
    qAbat(rsNat,a,ts)/max(1,qAbat.1(rsNat,a,tRun-1)) =E=
    sum[emisPLevels $ p_emisRed(rsNat,a,emisPLevels-1),
        {~ncpVUSIN(0,
            ~ncpVUSIN(emisP(rsNat,ts)/pindMaccs(rsNat,ts),p_emisPLevels(emisPLevels),1.E-3) + p_emisPLevels(emisPLevels-1),1.E-3)}
            * [0.5*(p_emisRed(rsNat,a,emisPLevels)-p_emisRed(rsNat,a,emisPLevels-1))/(p_emisPLevels(emisPLevels)-p_emisPLevels(emisPLevels-1))
              + p_emisRed(rsNat,a,emisPLevels-1)] *
        {
*
* --- quantity abated process emissions per unit of output
*
        emiaCo2eqTaxed(rsNat,a,ts)*m_xp(rsNat,a,ts)
*
* --- quantity of abated process emissions per unit of factor use
*
+ sum(f $ xFlag(rsNat,f,a),
    emifCo2eqTaxed(rsNat,f,a,ts)*xf(rsNat,f,a,ts))
*
* --- quantity of abated process emissions per unit of intermediate product use
*
+ sum( (j) $ (m_alphad(rsNat,j,a,ts) $ emidCo2eqTaxed(rsNat,j,a,ts)),
    emidCo2eqTaxed(rsNat,j,a,ts) * xd(rsNat,j,a,ts)*m_var4Scale(xd,rsNat,i,a,ts)*p_nCo2ShareD(rsNat,j,a))
+ sum( (j) $ (m_alphad(rsNat,j,a,ts) $ emiiCo2eqTaxed(rsNat,j,a,ts)),
    emiiCo2eqTaxed(rsNat,j,a,ts) * xm(rsNat,j,a,ts)*m_var4Scale(xm,rsNat,i,a,ts)*p_nCo2ShareI(rsNat,j,a))
    ]]/max(1,qAbat.1(rsNat,a,tRun-1))
;

```

Note here that we calculate the costs of process emission abatement not based on the last unit abated which defines the carbon tax according to marginal costs of abatement per unit being equal to the tax rate. Rather, we integrate over the step-wise linear function and thus consider that the first units are cheaper to abate, such as average costs of abatement are below marginal ones.

The third equation introduces a new demand agent called “maccs” in the model. Its demand is financed from the saving account. It is hence treated as an investment activity, but can have its own demand shares, defined by the parameter  $p\_macShare$ . As discussed below, the investment activity is not increasing the physical capital stock in follow-up period and competes with government demand. More on this later.

```

*
* --- demand for intermediates for process emission reduction
*
e_xaMaccs(rsNat(rnat),i,"maccs",ts) $ (xaFlag(rsNat,i,"maccs") or xaFlag(rsNat,"maccs",i)) ..
    xa(rsNat,i,"maccs",ts)*m_pa(rsNat,i,"maccs",ts)
    /xa.scale(rsNat,i,"maccs",ts)
    =E= sum(a $ ((v_redEmiaCo2eq.range(rsNat,a,ts) ne 0) $ p_emisRed(rsNat,a,"ep5000")),
        valAbat(rsNat,a,ts)*p_macShare(rsNat,i,a)
        + 1.E-10]
    /xa.scale(rsNat,i,"maccs",ts);

```

The right-hand side distributes the total value of abated process emissions  $valAbat$  to product demand based on the shares  $p\_macShare$ . The left-hand side defines the demand as the product of quantity and prices. The values on the parameter  $p\_macShare$  has to add up to unity when summed up over the products  $i$  for any activity  $a$  to ensure that the total abatement costs are exhausted by demand for services and products.

The last equation defines the total value of these demands over all products:

```

*
* --- total value of intermediate demand linked to process emission reduction
*
e_valMaccs(rsNat(rnat),ts) ..
    valMaccs(rsNat,ts)
    /max(1,sum(sameas(ts,tRun),valMaccs.1(rsNat,tRun-1)))
    =E= sum(i $ (xaFlag(rsNat,i,"maccs") or xaFlag(rsNat,"maccs",i)), xa(rsNat,i,"maccs",ts)*m_pa(rsNat,i,"maccs",ts))
    /max(1,sum(sameas(ts,tRun),valMaccs.1(rsNat,tRun-1)));

```

To reflect that changes in the prices of abatement services will impact their level, a prices index is defined:

```

e_pindMaccs(rsNat(rnat),ts(tRun)) ..
    pindMaccs(rsNat,ts)*sum(i $ (xaFlag(rsNat,i,"maccs") or xaFlag(rsNat,"maccs",i)), max(1.E-10,xa.1(rsNat,i,"maccs",tRun-1)))
    /max(1,sum(sameas(ts,tRun),valMaccs.1(rsNat,tRun-1)))
    =E= sum(i $ (xaFlag(rsNat,i,"maccs") or xaFlag(rsNat,"maccs",i)), max(1.E-10,xa.1(rsNat,i,"maccs",tRun-1))*m_pa(rsNat,i,"maccs",ts))
    /max(1,sum(sameas(ts,tRun),valMaccs.1(rsNat,tRun-1)));

```



This new variable is the one case where updates in the core model (*gams/model.gms*) are required. As it is assumed that savings finance the abatement technologies due to their investment character, we need to reduce the value of gross investments accordingly by the value of the related demand, *valMaccs*:

```

e_yi(rsNat,ts)
$iftheni.emp not "%modelType%"=="EMP"
    $ ( ((not rres(rsNat)) or singleCountry or (card(rsNat) eq 1) or (pnun.range(ts) gt 0)) )
$endif.emp
..

yi(rsNat,ts)/m_var2Scale1(yi,rsNat,ts) =e= (
*
*      ---- value of depreciation
*      valDep(rsNat,ts)*m_var2Scale(valDep,rsNat,ts)
*      ---- savings of the regional household
*      + rsav(rsNat,ts)*m_var2Scale(rsav,rsNat,ts)
$if "%modelType%"=="EMP"
*      + walras(ts) $ rres(rsNat)
*      ---- direct investments of firm
*      + invFromCapInc(rsNat,ts)
$if "%MACCsProcessEmis%"=="on"
*      - valMaccs(rsNat,ts)
*      ---- Value of foreign savings in local currency
*      + valsavf(rsNat,ts)*m_var2Scale(valSavf,rsNat,ts))/yi.scale(rsNat,ts);
.

```

As gross investment are reduced, the MACC demand is not increasing capital accumulation. The term *valMaccs* is introduced only when the MACC mechanism is switched on, i.e. the global *MACCsProcessEmis* is set to “on”. This is done via a checkbox on the GUI as shown below.

The same term *valMaccs* must be also reflected in the WALRAS check equation which calculates the above relation for the residual region *rres* for which is equation *e\_yi* is not active:

```

w_lras(ts) ..
{
  --- equivalent to savfeq (investments = savings), for residual region
  walras(ts) =e= sum(rres(rNat) $ rsNat(rNat),
    (yi(rNat,ts)*m_var2Scale(yi,rNat,ts)
    - (valDep(rNat,ts)*m_var2Scale(valDep,rNat,ts)
    + rsav(rNat,ts)*m_var2Scale(rsav,rNat,ts)
    + invFromCapInc(rNat,ts)
    + sum(ets_rnat(ets,rNat), emisEtsTrd.l(rNat,ts)*emisP(rNat,ts) $ (card(rNat) eq 1) + emisEtsTrd(rNat,ts)*emisP(rNat,ts) $ (card(rNat) gt 1))
    - valMaccs(rNat,ts)
    + valSavf(rNat,ts)*m_var2Scale(valSavf,rNat,ts))/yi.scale(rNat,ts) );
$if %ModulesCO2_Emissions%==on
$if %MACCsProcessEmis%==on

```

During simulation, in order to avoid that process emission abatement cost crowd out productive investments, for instance, the saving share of the government can be adjusted in a recursive-dynamic fashion taken last period's government spend and the value of the government subsidized cost of abatement into account:

```

*
*      --- shift government consumptions towards more investments
*
betasG.fx(rsNat,ts) = min(0.8,betasG.l(rsNat,"%t0%") + valMaccs.l(rsNat,ts)/(rsavg.l(rsNat,ts)+yg.l(rsNat,ts)));

```

Furthermore, the macro which adds to the production taxes *prdtx* emission taxes linked to process emissions had to be expanded in order into reflect that emission coefficients now can be endogenously reduced based on the variable *v\_redEmiaCo2eq*:

```

$iftheni.subregs "%modulesGTAP_NUTS2%"=="on"
$macro m_prdtx(r,a,t) (1 + prdtx(r,a,t)) \
  $if "%ModulesCO2_Emissions%"=="on" + sum(rr_rr(r,rsNat) $ v_redEmiaCo2eqTaxed(rsNat,a,t),emisP(rsNat,t) * (emiacCo2eqTaxed(rsNat,a,t) * (1 - v_redEmiaCo2eq(rsNat,a,t)))
$else.subregs
$macro m_prdtx(r,a,t) (1 + prdtx(r,a,t)) \
  $if "%ModulesCO2_Emissions%"=="on" + emisP(r,t) * (emiacCo2eqTaxed(r,a,t) * (1 - v_redEmiaCo2eq(r,a,t))) $ emiacCo2eqTaxed(r,a,t)
$endif.subregs

```

Similarly, the macro for factor taxes has to consider now the share of abated process emissions per unit of factor use:

```

$$macro m_fctTaxes(r,f,a,t) ( 1 + fctts(r,f,a,t) + fcttx(r,f,a,t) + fcttxShift(r,t) $ fTaxShift(f,a))\
  $$iftheni.CO2 %ModulesCO2_Emissions%==on
    + emispl(r,t) * emifCo2eqTaxed(r,f,a,t) * (1-v_redEmiaCo2Eq(r,a,t)) \
    $$ifi.AEZ "%modulesGTAP_AEZ%"=="on" + emisplnd(r,a,t) $ lnd(f)
  $$endif.CO2

```

And finally, the definitions of imported and domestic prices need to take the abated shares into account:

```

*
* --- the macro m_pdp substitutes out the domestic sales prices: with an infinite CET for exports, use the supply
* price ps, otherwise, use the domestic price
*
$$macro m_pdpDef(r,i,aa,t) ( m_domPrice(r,i,aa,t) \
  * (1 + dntx(r,i,aa,t) + itxshft(r,i,t)) \
  *
  --- add emission taxes for domestic use
  $$ifi %ModulesCO2_Emissions%==on + emispl(r,t) * emidCo2eqTaxed(r,i,aa,t) * (1-v_redEmiaCo2Eq(r,aa,t) * p_nCo2ShareD(r,i,aa) $ a(aa)) \
  )
*
* --- the macro m_pmp substitutes out the imports' sales prices
*
$$macro m_pmpDef(r,i,aa,t) ( \
  *
  ---- average import price ( pmt or pmtMrio )
  *
  m_pmt(r,i,aa,t) * (1 + dntx(r,i,aa,t) + itxshft(r,i,t)) \
  *
  --- add emission taxes for imported use
  $$ifi %ModulesCO2_Emissions%==on + emispl(r,t) * emiiCo2eqTaxed(r,i,aa,t) * (1-v_redEmiaCo2Eq(r,aa,t) * p_nCo2ShareI(r,i,aa) $ a(aa)) \
  )

```

This required to set fix this variable to zero if the MACC extension is not in use, in “co2Eq/coe2Eq.gms”:

```

$$iftheni.maccs not "%MACCsProcessEmis%"=="on"
  v_redEmiaCo2eq.fx(rsNat,a,tSim) $ emiaCo2eqTaxed(rsNat,a,tSim) = 0;

```

A share *abatCostSharePrdtx* of the MAC can be allocated to production taxes according to the following equation:

```

e_gdpAbat(rsNat(rnat),a,ts) $ ((prdtx.range(rnat,a,ts) ne 0) $ (v_redEmiaCo2eq.range(rsNat,a,ts) ne 0) $ p_emisRed(rsNat,a,"ep5000")) ..
  [prdtx(rnat,a,ts)-prdtx.l(rnat,a,"%t0%")]*m_xp(rnat,a,ts)/xp.scale(rnat,a,ts)
  =E= qAbat(rnat,a,ts)*pindMaccs(rnat,ts) * %AbatCostSharePrdtx%/xp.scale(rnat,a,ts);

```

The left-hand side defined the value of the changed tax income (difference in tax rates time production quantity as the tax base), the right hand side defines the total activity specific costs of abatement times the share which is charged to the activity itself. The remainder of the cost need to be subsidized by government.

An extension for of the myGTAP model with endogenous government borrowing considers that the MAC demand is financed by a mix of (increased) production taxes and government subsidies:

```

*
* --- government account: gov savings as a share of GDP (benchmark relation)
*
e_rsavg(rsNat(rnat),ts(t)) $ (rsavg.range(rnat,t) ne 0) ..
  rsavg(rnat,t)/rsavg.scale(rnat,t) =E= rsavg.l(rnat,"%t0%")/rsavg.scale(rnat,t)
  + (yTaxTot(rnat,t)
    $ifi "%MACCsProcessEmis%"=="on" - valMaccs(rnat,t)*(1-%AbatCostSharePrdtx%)
    -sum((r_r(rnat,r),h) $ hFlag(r,h), p_payPastGovBonds(r,h,t)))/yTaxTot.l(rnat,"%t0%") * spendShrByGovInc(rnat,t))
  )
*
* --- demand for MACC
*
$ifi "%MACCsProcessEmis%"=="on" + valMaccs(rnat,t)/rsavg.scale(rnat,t)
;

```

The last term will increase government saving by the value of the MACCs. The first term drives productive savings demand of the government partly by GDP and partly a tax income, corrected by the share of MAC which is paid via production taxes by the activities.

## Initialization

The initialization consists of two blocks:

### 1. Input of the MACCs

Currently, a dummy table is available (see *gams/scen/ghg/maccs\_dummy.gms*) for testing purposes:

```
table p_emisRed(*,*,emisPLevels) "Relative reduction in process emissions reduction"

      ep1      ep5      ep10      ep50      ep100      ep500      ep1000      ep5000
wor.nmm-a    1.E-6    1.E-4                0.75      0.95      1.00
wor.bph-a    1.E-6    1.E-4                0.75      0.95      1.00
wor.chm-a    1.E-6    1.E-4                0.75      0.95      1.00
wor.ele-a    1.E-6    1.E-4                0.50      0.95      1.00
wor.wtr-a    1.E-6    1.E-4                0.50      0.75      0.99      1.00
wor.ctl-a    1.E-6    1.E-4                0.10      0.50      0.90      1.00
wor.rmk-a    1.E-6    1.E-4                0.10      0.50      0.90      1.00
wor.rest     1.E-6    0.05      0.10      0.20      0.75      1.00
;
```

It comprises made-up default values for the world; for some sectors, explicit assumptions are invented for some sectors (non-minerals nec, chemicals, electric equipment, water, cattle and row milk) and for all others, one MACC curve is defined in the row “rest”. This data serves to test the functioning of the extension.

In a first step, we copy the defaults stored on “rest” to any activity with process emissions and no explicit MACC information:

```
set aProcEmis(a) "Activities with process emissions";
aProcEmis(a) $ sum(rNat, emiaCo2EqTaxed(rnat,a,"%t0%")) = YES;

*
*
* --- if no information on specific activity is given use information for "rest"
*
p_emisRed("wor",aProcEmis,emisPLevels) $ (not sum(emisPLevels1, p_emisRed("wor",aProcEmis,emisPLevels1)))
= p_emisRed("wor","rest",emisPLevels);
*
```

The next statements have been discussed above. They ensure that all points on the MACCs curves are present, based on linear interpolation:

```
*
* --- Set (almost) ( zero reduction at tax of zero
*
p_emisRed(rWor,aProcEmis,"ep0") $ sum(emisPLevels,p_emisRed(rWor,aProcEmis,emisPLevels))
= 1.E-10;
*
* --- make sure that last step is always set
*
p_emisRed(rWor,aProcEmis,"ep1000") $ (sum(emisPLevels,p_emisRed(rWor,aProcEmis,emisPLevels))
$ (not p_emisRed(rWor,aProcEmis,"ep1000")))
= smax(emisPLevels,p_emisRed(rWor,aProcEmis,emisPLevels));
*
* --- linearly interpolate missing steps
*
loop(emisPLevels,

  p_emisRed(rWor,aProcEmis,emisPLevels) $ ((not p_emisRed(rWor,aProcEmis,emisPLevels)) $ p_emisRed(rWor,aProcEmis,"ep0"))
  = [smin(emisPLevels1 $ ((emisPLevels1.pos gt emisPLevels.pos) $ p_emisRed(rWor,aProcEmis,emisPLevels1)),
    p_emisRed(rWor,aProcEmis,emisPLevels1))
    - p_emisRed(rWor,aProcEmis,emisPLevels-1)]
    / [smin(emisPLevels1 $ ((emisPLevels1.pos gt emisPLevels.pos) $ p_emisRed(rWor,aProcEmis,emisPLevels1)),
    p_emisPLevels(emisPLevels1)) - p_emisPLevels(emisPLevels-1)]

    * [p_emisPLevels(emisPLevels) - p_emisPLevels(emisPLevels-1)] + p_emisRed(rWor,aProcEmis,emisPLevels-1);
);
```

In case where no model region specific information is given (which is the case for the test data set for all regions), the world MACC is used:

```
*
*
* --- use world average if no information for country is given
*
p_emisRed(rNat,aProcEmis,emisPLevels) $ ((not p_emisRed(rNat,aProcEmis,emisPLevels)) $ emiaCo2EqTaxed(rnat,aProcEmis,"%t0%"))
= p_emisRed("wor",aProcEmis,emisPLevels);
```

## 2. Introduction of the new Armington agent

In order to introduce the demand for abatement into all relevant equations (domestic and import demand balances, tax revenues, definition of GDP, etc.), the set of SAM columns *is* and Armington agents *aa* in the model is extended (in *build/load\_gtapAgg.gms*):

```
$if not "%modulesCo2_emissions%"=="on" $setGlobal MACCsProcessEmis off

set is /
    set.is1
    set.set_hhslds
    $if "%MACCsProcessEmis%"=="on" maccs
/;

set aa(is) /
    set.aa_ori
    set.set_hhslds
    $if "%MACCsProcessEmis%"=="on" maccs
/;
```

To avoid that the global for the MACCs is on and variables such as *valMaccs* are introduced in the model despite the co2 emission accounting being switched off, the global is automatically set to off in this case as shown above.

For our test implementation, we assume that the abatement requires a mix of services, construction and machinery:

```
*
*   ---- assumed shares: 36% is better management, linked to demand for education, Business services nec
*                           and linked to services to implement (communication, other financial mediation)
*
p_maccsShare(rNat,i,aProcEmis) $ sameas(i,"obs-c") = 0.15;
p_maccsShare(rNat,i,aProcEmis) $ sameas(i,"ofi-c") = 0.03;
p_maccsShare(rNat,i,aProcEmis) $ sameas(i,"edu-c") = 0.15;
p_maccsShare(rNat,i,aProcEmis) $ sameas(i,"cmn-c") = 0.03;
*
*   ---- rest goes partly to construction and partly to other machiney and equipment (filters, ...)
*   assume 2% insurance cost for new installations
*
p_maccsShare(rNat,i,aProcEmis) $ sameas(i,"cns-c") = 0.64 * .21;
p_maccsShare(rNat,i,aProcEmis) $ sameas(i,"lum-c") = 0.64 * .03;
p_maccsShare(rNat,i,aProcEmis) $ sameas(i,"ome-c") = 0.64 * .39;
p_maccsShare(rNat,i,aProcEmis) $ sameas(i,"eeq-c") = 0.64 * .10;
p_maccsShare(rNat,i,aProcEmis) $ sameas(i,"omf-c") = 0.64 * .10;
p_maccsShare(rNat,i,aProcEmis) $ sameas(i,"trd-c") = 0.64 * .05;
p_maccsShare(rNat,i,aProcEmis) $ sameas(i,"ins-c") = 0.64 * .02;
```

Note that the demand for construction services “cns-c” is relatively small to avoid that indirectly a higher demand for cement is generated.

As we need information in consumption taxes and domestic and imported demand shares for the new agent, we borrow those from investments. This implies that we have to delete all MACC demand shares where no investment demand is found:

```
*
*   --- MACC demand is derived from investment demand, delete shares where not inv demand is observed
*
p_maccsShare(rNat,i,aProcEmis) $ (not xaFlag(rnat,i,"inv")) = 0;
```

We defined from the remaining shares all products which show MACC demand and scale the demand shares to unity to ensure the FOC condition for cost-minimal marginal abatement:

```

set iMaccs(i) "Products with MACCS demand";
iMaccs(i) $ sum((r,aProcEmis),p_maccsShare(r,i,aProcEmis)) = YES;
*
* --- scale sajres to unity
*
p_maccsShare(rNat,i,aProcEmis) $ p_maccsShare(rNat,i,aProcEmis)
= p_maccsShare(rNat,i,aProcEmis) * 1/sum(iMaccs,p_maccsShare(rNat,iMaccs,aProcEmis));

```

Next, we introduce the Armington demand into the model based on the flag *xaFlag* and borrow the domestic and imported demand shares and related taxes from investment demand:

```

*
* --- Set armington flag and tax over shares and taxes from investment
*
xaFlag(rNat,iMaccs,"maccs") $ xaFlag(rNat,iMaccs,"inv") = YES;
xaFlag(rNat,"maccs",iMaccs) $ xaFlag(rNat,"inv",iMaccs) = YES;

p_alphaD(rnat,iMaccs,"maccs",t0) = p_alphaD(rnat,iMaccs,"inv",t0);
p_alpham(rnat,iMaccs,"maccs",t0) = p_alpham(rnat,iMaccs,"inv",t0);

mintx(rnat,iMaccs,"maccs",t0) = mintx(rnat,iMaccs,"inv",t0);
dintx(rnat,iMaccs,"maccs",t0) = dintx(rnat,iMaccs,"inv",t0);

```

We finally assign a quite tiny starting value and from there import and domestic demands:

```

*
* --- almost zero starting values
*
xa(rnat,iMaccs,"maccs",t0) $ xa(rnat,iMaccs,"inv",t0) = 1.E-10;
pa(rnat,iMaccs,"maccs",t0) $ xa(rnat,iMaccs,"inv",t0) = 1;
xm(rnat,iMaccs,"maccs",t0) $ xa(rnat,iMaccs,"inv",t0) = xa(rnat,iMaccs,"maccs",t0) * xm(rnat,iMaccs,"inv",t0)/xa(rnat,iMaccs,"inv",t0);
xd(rnat,iMaccs,"maccs",t0) $ xa(rnat,iMaccs,"inv",t0) = xa(rnat,iMaccs,"maccs",t0) * xd(rnat,iMaccs,"inv",t0)/xa(rnat,iMaccs,"inv",t0);

valMaccs(rnat,t0) = 0;
valAbat(rnat,a,t0) = 0;

```

## Interface

On the interface, in the CO2 emission panel, there is now a checkbox to switch the MACC mechanism on and to select the file the MACC information (stored under *gams/scen/ghg/*; the file name must start with *maccs\_*):

The “Maximum abatement option to consider” allows to restrict the considered points on the MACCs. If emission prices are endogenous under a ceiling, removing more expensive options can imply only that other abatement options, such as movement on the production function towards less emitting inputs, must be used instead which should normally imply that total cost will be higher.

## Reporting

The reductions in emission coefficients are reflected when the process emissions are assigned to the *p\_results* array used by the exploitation tool, both for taxed and individual emissions:

```

p_results(rsNat,"CO2eqTaxed","xp",a,"dom",%arg1%,"%version%") $ xp.1(rsNat,a,%arg2%)
= emiaCo2EqTaxed(rsNat,a,%arg2%)*(1-v_redEmiaCo2Eq.1(rsNat,a,%arg2%))/gblScale * (xp.1(rsNat,a,%arg2%)+xTNest.1(rsNat,"fcost",a,%arg2%));

*
* --- linked to firm output
*
p_results(rsNat,nCO2,"xp",a,"dom",%arg1%,"%version%") $ xp.1(rsNat,a,%arg2%)
= emiIN(rsNat,nCO2,"",a,%arg2%)*(1-v_redEmiaCo2Eq.1(rsNat,a,%arg2%))/gblScale * (xp.1(rsNat,a,%arg2%)+xTNest.1(rsNat,"fcost",a,%arg2%));

```

Finally, the table with the market demands are expanded to comprise the new agent:

Demand, by product [1]

Regions: Romania Items: Quantity Origins: Total Years: 136

nnn\_GDP\_SSP2\_POP\_SSP2\_CC\_None\_CS\_ON

	Output	Total	Household demand	Government demand	Intermediate demand	Investment demand	MACCs demand	Exp
All commodities	1369.20	1635.43	323.40	71.64	717.43	182.08	14.53	
Grains and Crops	18.00	24.67	4.85	0.11	15.59	0.15		
Livestock and Meat Products	5.84	8.15	3.95	0.01	3.91	0.07		
Mining and Extraction	7.50	18.90	1.56	0.12	15.69	0.45		
Processed Food	34.03	43.58	25.10	0.09	11.81	0.00		
Textiles and Clothing	20.91	36.87	12.03	0.00	15.32	0.01		
Light Manufacturing	135.60	186.62	21.45	0.98	94.78	15.76		
Heavy Manufacturing	303.10	443.89	26.19	3.07	176.08	47.68		
Utilities and Construction	227.46	230.41	31.00	0.14	87.31	90.10	14.53	
Transport and Communication	308.30	319.34	100.03	7.71	167.56	15.37		
Other Services	308.46	323.00	97.25	59.40	129.38	12.49		
All commodities	1369.20	1635.43	323.40	71.64	717.43	182.08	14.53	
Grains and Crops	18.00	24.67	4.85	0.11	15.59	0.15		
Residuals	0.01	0.01			0.01			

## References

Britz, W., Ciaian, P., Gocht, A., Kanellopoulos, A., Kremmydas, D., Müller, M., Petsakos, A., Reidsma, P. (2021): A design for a generic and modular bio-economic farm model, *Agricultural Systems* 191(June 2021)

Kiuiila, O., Markandya, A., & Ščasný, M. (2019). Taxing air pollutants and carbon individually or jointly: results from a CGE model enriched by an emission abatement sector. *Economic Systems Research*, 31(1), 21-43.

Kiuiila, O., & Rutherford, T. F. (2013). The cost of reducing CO2 emissions: Integrating abatement technologies into economic modeling. *Ecological Economics*, 87, 62-71.

### 3.9.3 Air pollution emissions

There is one equation and related variable to determine air pollution emissions<sup>29</sup> in the core model:

```

$iftheni.air_pol %ModulesAir_Pollution%==on
variable airPol(rsNat,emisBst,t) "Air pollution emissions";
equation e_airPol(rsNat,emisBst,t) "Air pollution emissions";

set aaAir(aa);

e_airPol(rsNat,emisBst,ts) $ (airPol.range(rsNat,emisBst,ts) ne 0) ..

    airPol(rsNat,emisBst,ts) / airPol.scale(rsNat,emisBst,ts)
    =E=
    [
*      --- link to final consumption
    sum((aaAir,i) $ ((xaFlag(rsNat,aaAir,i) or xaFlag(rsNat,i,aaAir))) $ emiiA(rsNat,emisBst,i,aaAir)),
    nm_xa(rsNat,i,aaAir,ts) * emiiA(rsNat,emisBst,i,aaAir))
*      --- link to firm output
    + sum(a $ (xpFlag(rsNat,a) $ emiiA(rsNat,emisBst,"",a)),
    (m_xp(rsNat,a,ts) + xtNest(rsNat,"fCost",a,ts) $ xtNest.1(rsNat,"fCost",a,ts))
    * emiiA(rsNat,emisBst,"",a))
*      --- link to factor use
    + sum((a,f) $ (xfFlag(rsNat,f,a) $ emiiA(rsNat,emisBst,f,a)),
    xf(rsNat,f,a,ts) * emiiA(rsNat,emisBst,f,a))
    ] / airPol.scale(rsNat,emisBst,ts);

$endif.air_pol

```

In opposite to the CO2 and non-CO2 emissions, air pollution emission factors are not differentiated between the domestic and imported origin. Similar to non-CO2 emissions, there are also pollution factor linked to factor use and production output.

<sup>29</sup> Chepeliev, M. (2020). *Development of the Air Pollution Database for the GTAP Data Base Version 10A* (No. 6163). Center for Global Trade Analysis, Department of Agricultural Economics, Purdue University, GTAP research memorandum 33. <https://www.gtap.agecon.purdue.edu/resources/download/10042.pdf>

The set aaAir makes sure that only the individual Armington agents are used in the equations, and not the potentially aggregated positions “armA” and “int”, see cal\cal.gms:

```
$$iftheni.air pol %ModulesAir_Polution%==on
```

```
aaAir(aaArm) = YES;  
aaAir(aa)    = YES;  
aaAir("armA") = NO;  
aaAir("int")  = NO;
```

### 3.9.4 Reporting of energy use in Million tons of oil equivalents (MTOE)

The energy use accounting in Million tons of oil equivalents (mtoe) is solely done for reporting purposes and part of the post-model reporting code.

## 3.10 GTAP-E

GTAP-E is entirely based on the flexible nesting approach and is detailed there, see section “Flexible nesting.”

## 3.11 GTAP-AGR

GTAP-AGR is based on several changes in the GTAP in the standard layout and implemented in CGEBox by using the flexible nesting approach for the production function and the factor supply (see “gams\gtapAgr\gtapAgr-Model.gms”).

For livestock activities lstk, it introduces a sub-nest under the ND nests which allows for substitution between feedstock feed:

```
if ( card(feed),  
    tNest("feed") = yes;  
    tNest_i_a("feed", feed, lstk) = yes;  
    tNest_n_a("ND", "feed", lstk) = YES;  
);
```

The two sets are defined in “gams\gtapAgr\gtapAgr\_def.gms”:

```
: set lstk0(i0) /  
:         ctl      "Cattle,sheep,goats,horses"  
:         oap      "Animal products nec"  
:         rmk      "Raw milk"  
:         wol      "Wool, silk-worm cocoons"  
: /;  
: set lstk(a); lstk(a) $ sum(mapa(lstk0,a),1) = YES;  
:  
:  
: set feed0(i0) /  
:         set.icrops  
:         cmt  
:         omt  
:         vol  
:         mil  
:         pcr  
:         sgr  
:         ofd  
: /;  
:  
: set feed(i); feed(i) $ sum(mapi(feed0,i),1) = YES;
```

Intermediate demand substitution is also introduced in the food processing industries:



```
set foodPrc0(i0) / cmt,omt,vol,mil,pcr,sgr,ofd,b_t      /;
set foodPrc(a); foodPrc(a) $ sum(mapa(foodPrc0,a),1) = YES;

if ( card(agr_c),
    tNest("foodPrc") = yes;
    tNest_i_a("foodPrc",agr_c,foodPrc) = yes;
    tNest_n_a("ND","foodPrc",foodPrc) = YES;
);
```

Factor supply to mobile factors is differentiated between agricultural and non-agricultural activities:

```
if ( card(agr),

    fNest("agr") = YES;
    fNest_a_f("agr",agr,ffNest) = YES;
    fNest_n_f("xft","agr",ffNest) = YES;
    omega_fNest(r,"agr",ffNest) = omega_f(r,ffNest);

    fNest("nonAgr") = YES;
    fNest_a_f("nonAgr",nonAgr,ffNest) = YES;
    fNest_n_f("xft","nonAgr",ffNest) = YES;
    omega_fNest(r,"nonAgr",ffNest) = omega_f(r,ffNest);
);
```

Note that the set ffNest will cover all mobile factors with the exemption of land if the GTAP-AEZ module is active.

The parameterization is based on parameters by Keeney and Hertel, details do not matter here, the code in “gams\gtapAgr\gtapAgr\_def.gms” should be rather self-explanatory.

### 3.12 G-RDEM: Long-term baseline generation and analysis

This section is prepared by Roberto and Wolfgang Britz, drawing on: G-RDEM: A GTAP - Based Recursive Dynamic CGE model for Long Term Baseline Generation and Analysis, in: Bocconi Working Paper Series No. 105, April 2018. The working paper was in revised form published as Britz, W., Roson, R. (2019): G-RDEM: A GTAP-Based Recursive Dynamic CGE Model for Long-Term Baseline Generation and Analysis, *Journal of Global Economic Analysis* 4(1): 50-96. Second then, G-RDEM had been expended by additional elements: expenditure shares of the government and of investors can be rendered a function of income per capita changes, drawing on econometric work, crop yields and crop land expansions can be driven by exogenous projections, and calorie intakes per calories controlled based on a empirical relation to income per capita (Britz, W. (2020): Maintaining plausible calorie intakes, crop yields and crop land expansion in long-run simulations with Computable General Equilibrium Models. Discussion Paper 2020:2, Institute for Food and Resource Economics, University Bonn). Finally, the AIDADS demand system was expanded and to MAIDADS system and re-estimated (Britz, W. (2021): Estimating a global MAIDADS demand system considering demography, climate and norms, Bio-based and Applied Economics, pre-Journal print available online).

#### 3.12.1 Summary

The G-RDEM module allows to user to construct long-run baselines drawing on the so-called Socio-Economic Pathways (SSPs) developed for the International Panel on Climate Change (IPCC), adding model mechanisms important for the long-term. These baselines can be subsequently used for counterfactual analysis. The necessary data for the SSPs are distributed with the model code and need not to be supplied by the user. The different mechanisms to include in the model as well as the period to cover and the time resolution can be steered via the interface.

#### 3.12.2 Background and introduction

Due to issues such as climate change and depletion of global resources, there is an increasing demand for long-term quantitative analyses. Computable General Equilibrium models can contribute in that direction as they consistently consider the manifold interrelations occurring in the economy, while providing the often needed sectoral detail. They therefore complement approaches working at the more aggregate level (e.g. Dellink et al. 2017<sup>30</sup>) or focusing in detail on specific sectors (e.g. Alexandratos and Bruinsma 2012<sup>31</sup>). On the other hand, it should be noted that CGE models were not originally designed to this purpose, but rather for short-term policy assessment, like simulating the effects of a fiscal reform, or the implementation of a trade agreement. Accordingly, most parameters are usually “calibrated” to a relatively recent Social Accounting Matrix or Input Output Table, such that the observed structure of an economic system is taken as a benchmark, from which counterfactual experiments are conducted

Of course, when the economy is analysed at a horizon of 20, 30 years, or even more, the economic structure as emerging from some past national accounts, which may refer to five years back, is no more a valid starting point. One should consider trends in structural adjustment, driven by changing preferences, demographic composition, new technologies, variations in the endowments of primary

---

<sup>30</sup> Dellink, R., Chateau, J., Lanzi, E., & Magné, B. (2017). Long-term economic growth projections in the Shared Socioeconomic Pathways. *Global Environmental Change*, 42, 200-214

<sup>31</sup> Alexandratos, N., & Bruinsma, J. (2012). World agriculture towards 2030/2050: the 2012 revision (Vol. 12, No. 3). FAO, Rome: ESA Working paper

resources (including human capital), etc. The whole issue is not about forecasting: nobody actually knows which “breakthrough” technologies could emerge, or which unexpected phenomena could shape the economic structure in the future. What we do know from past observation is that a number of “slow” adjustment processes are active and therefore they should be taken into account in the generation of a credible and internally consistent future baseline.

The study of time evolution of the economic structure (“structural change”) is a rather active research field in theoretical and applied economics (Matsuyama, 2008<sup>32</sup>). Most of the studies in the literature, however, look at the past. Typical research questions are: the contribution of the changing industrial mix to aggregate productivity (e.g., Duarte and Restuccia 2010<sup>33</sup>); the declining share of the agricultural sector in developing economies (e.g., Üngör 2013<sup>34</sup>), etc., where some specific transition processes are identified. Here, rather than studying the past, we aim at drawing from some empirical findings and methodologies in this literature to infer, inside a CGE modelling framework, a possible future evolution of the economy.

To this end, a number of “unconventional” features have to be introduced into the standard CGE formulation, to create a model specifically designed for the generation and assessment of long-term economic scenarios. We present therefore in here a newly develop CGE model of this kind, termed G-RDEM (GTAP based Recursive Dynamic Economic Model). This model considers drivers of long run structural change, which we regard as especially relevant, namely: (1) non-linear Engel curves in household consumption, (2) productivity growth differentiated by sector, (3) debt accumulation from foreign savings and trade imbalances, (4) aggregate saving rates linked to population and income dynamics, and (5) time-varying and income dependent industrial cost shares.

G-RDEM extends the flexible and modular CGE modelling platform CGEBox, from which it inherits some important features. Firstly, the code is open source, to ensure transparency and invite the community of modellers to use the tool and contribute to its further development. Secondly, it maintains full flexibility in sectoral and regional aggregation. Thirdly, G-RDEM as a seamless integrated module in CGEBox offers the possibility to combine it with other modules such as CO2 and Non-CO2 emissions, GTAP-Water, GTAP-AEZ etc.. All new features are based on econometrically estimated parameters, thereby making the implementation fully documented and transparent. G-RDEM is encoded in the GAMS modelling language and, as a module of CGEBox, shares its graphical user interface

### 3.12.3 Overview

The construction of a long-term baseline in CGE models typically draws on population and GDP projections from other studies. Indeed, a recursive dynamic CGE only considers capital accumulation as an endogenous mechanism driving growth, while productivity changes and other drivers of structural change are usually kept exogenous. In order to let a CGE model replicate a given growth path, a total factor productivity shifter is endogenously determined during the construction of the reference baseline, by fixing GDP at each time period. In subsequent model runs and counterfactual

---

<sup>32</sup> Matsuyama, K. (2008). Structural change. In S. Durlauf, & L. E. Blume (Eds.), *The New Palgrave Dictionary of Economics* (2 ed.). Palgrave-Macmillan

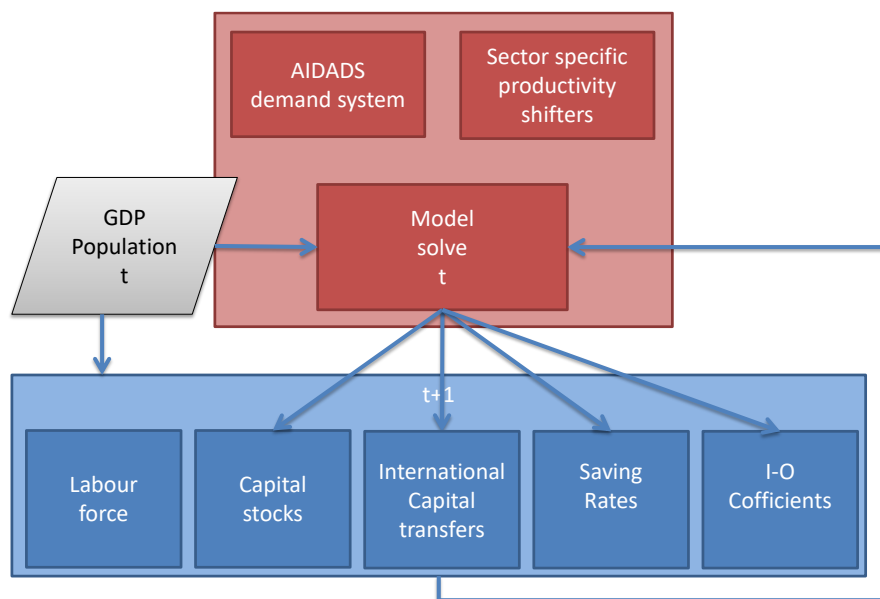
<sup>33</sup> Duarte, M., Restuccia, D., (2010). The role of the structural transformation in aggregate productivity. *The Quarterly Journal of Economics* 125, 129-173

<sup>34</sup> Üngör, M., (2013). De-agriculturalization as a result of productivity growth in agriculture. *Economics Letters* 119, 141-145

simulations, productivity parameters are then maintained at those estimated levels, while national income is endogenously computed.

This simple methodology aligns the output of the CGE model to a pre-determined aggregate growth path, but of course does not capture some fundamental structural changes which may take place in the economy, i.e. in the composition of output and demand. Instead, we aim in here to address the key elements driving such compositional change (Figure 7).

To this end, we introduce an AIDADS demand system to consider how budget shares in household consumption adjust to the changing levels of per capita income, to capture “non-linear Engel curves”, which are a salient feature of economic development. Secondly, the economy wide total factor productivity (TFP) shifter, aligning the model to the target GDP in any period, is here differentiated by sector. These two features are introduced through specific equations directly into the CGE framework itself (red boxes in Figure 1). Other elements are activated in between the solution points (blue boxes). Therefore, the intra-periodal equilibrium computed by the model, in combination with exogenous projections from the current period  $t$ , updates some parameters for the following period  $t+1$ . The labour force (by skill category) is adjusted to population and work force projections. Next year’s capital stocks reflect last year’s ending stocks and gross investments. International capital transfers reflect past foreign savings. Saving rates adjust to population and GDP growth, and I-O coefficients (factor shares in production processes) are updated on the basis of national income.



**Figure 7: Overview of the recursive-dynamic modelling framework G-RDEM**

The process thus requires some exogenous projections for GDP and population. G-RDEM offers the possibility to draw on a set of projections for the so-called SSPs (Shared Socio Economic Pathways) (Riahi et al. 2016<sup>35</sup>), available online from the IIASA Shared Socio Economic Pathways Database.<sup>36</sup> These SSPs were developed in the context of the IPCC scientific assessment on Climate Change. For each of these five SSPs, a single population and urbanization scenario, jointly developed by IIASA

<sup>35</sup> Riahi, K., Van Vuuren, D. P., Kriegler, E., Edmonds, J., O’neill, B. C., Fujimori, S., & Lutz, W. (2017). The shared socioeconomic pathways and their energy, land use, and greenhouse gas emissions implications: an overview. *Global Environmental Change*, 42, 153-168

<sup>36</sup> <https://tntcat.iiasa.ac.at/SspDb/dsd?Action=htmlpage&page=about>.

and NCAR (National Center for Atmospheric Research), can be combined with GDP projections from either the OECD or IIASA. These GDP and population projections are available in 5-year steps up to 2100, at a single country basis. They are aggregated in G-RDEM to the desired regional aggregation level and interpolated to yield yearly time series. They can also be complemented by Climate Change impacts on yields for a set of RCPs (Representative Concentration Pathways) and various combinations of GCMs (Global Circulation Models) and global gridded growth models provided by the AgCLIM50 project (van Meijl et al. 2017<sup>37</sup>).

A user might also add its own scenario assumptions during the construction of the baseline, such as about trade policies or alternative GDP projections. After the definition of the baseline, the software saves the resulting productivity shifters and other variables, which can subsequently be loaded as exogenous parameters for counterfactual analysis. The set of results from the baseline can also be directly employed, to get a much disaggregated definition of the economic scenario.

### **3.12.4 An AIDADS demand system with detail for food consumption**

#### **Background**

It is universally acknowledged that the relationship between consumption level and income (also known as Engel curve) can be complex and non-linear. Yet, many CGE models still adopt demand systems such as Cobb-Douglas (CD) or Linear Expenditure System (LES), having linear Engel curves. Those simplifying assumptions make the model easier to handle, but are defensible only if the model is used for simulations involving limited changes in income levels. Of course, this is not the case for long-term analyses. Keeping constant marginal budget shares would lead to an overestimation of the demand for necessities, such as food, while demand in other sectors will hence be underestimated. The consequences are implausible long-run structural changes in production, demand, and trade patterns. Some models employed for long-term analysis therefore use different demand systems and/or re-parameterize along the dynamic path. For instance, MAGNET (Woltjer et al. 2014, p. 84<sup>38</sup>) incorporates a module for re-calibrating the parameters of a CDE (Constant Differences in Elasticity) demand system to given income elasticities. Nonetheless, the authors admit: “All of these parameters and functional forms are very much ad hoc, and should be improved.”

Following Roson and van der Mensbrugghe 2018<sup>39</sup>, we rather implement an empirically estimated AIDADS demand system into the G-RDEM model, for broad product groups. The AIDADS is An Implicit, Directly Additive Demand System (Rimmer and Powell 1996<sup>40</sup>). It can be understood as a generalization of a LES demand system, where marginal budget shares are not fixed, but are a linear combination of two vectors, depicting the marginal budget structure at very low and very high utility (income) levels. Given that the marginal budget shares in the two vectors fulfil the adding up

---

<sup>37</sup> van Meijl, H., Havlik, P., Lotze-Campen, H., Stehfest, E., Witzke, P., Dominguez, I.P., Bodirsky, B., van Dijk, M., Doelman, J., Fellmann, T. and Humpenoeder, F., 2017. Challenges of Global Agriculture in a Climate Change Context by 2050 (AgCLIM50). JRC Science for Policy Report, EUR 28649 EN, doi:10.2760/7724

<sup>38</sup> Woltjer, G. B., Kuiper, M., Kavallari, A., van Meijl, H., Powell, J. P., Rutten, M. M., ... & Tabeau, A. A. (2014). The MAGNET model: Module description (No. 14-57). LEI Wageningen UR

<sup>39</sup> Roson R, & van der Mensbrugghe D. (2018), Demand-Driven Structural Change in Applied General Equilibrium Models, in F. Perali and P. L. Scandizzo (eds.), The New Generation of Computable General Equilibrium Models, Springer International, doi: /10.1007/978-3-319-58533-8\_2

<sup>40</sup> Rimmer, M. T., & Powell, A. A. (1996). An implicitly additive demand system. *Applied Economics*, 28(12), 1613-1622

condition to unity, any linear combination of the two vectors also leads to regular budget shares. In order to improve the detail inside the agri-food sector, we also took econometric estimates of income dependent marginal expenditure shares for food categories from Muhammad et al. 2011<sup>41</sup>, and incorporate them in the AIDADS framework.

Cranfield et al. 2000<sup>42</sup> improve on the original Rimmer and Powell 1996 approach, by developing an estimation method that does not rely on an approximation of utility. We follow their notation in the following. The demand system is defined below. Equation (1) determines the Marshallian demand, which is similar to that of a LES. Here, however, the marginal budget shares  $\delta_i$  are endogenous variables, defined by (2), expressed as a linear combination of two vectors  $\alpha$  and  $\beta$ , function of the utility level  $u$ , implicitly defined by (3).

$$x_i = \bar{\gamma}_i + \frac{\delta_i}{p_i} \left[ Y - \sum_j \bar{\gamma}_j p_j \right] \quad (1)$$

$$\delta_i = \frac{\bar{\alpha}_i + \bar{\beta}_i u}{1 + u} \quad (2)$$

$$\ln(u) = \sum_i \delta_i \ln(x_i - \bar{\gamma}_i) + \bar{\kappa} \quad (3)$$

### Estimation and integration into modelling framework

We first econometrically estimated  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $u$  using data from the International Comparison Program ICP 2015<sup>43</sup>, for ten broader expenditure categories (food, beverages and tobacco, clothing, housing, furniture, transportation, recreation, communication, health, education) plus details for food expenditures. The integration in the CGE model requires mapping the parameter estimates to the commodity resolution of the model.

The demand system is calibrated against the benchmark data of regional household consumption, from the GTAP v.9 data set. To this purpose, we regressed the utility levels  $u$  from our findings to total per capita consumption expenditure  $Y$  in each region. That allows us to estimate (from (2)) the marginal budget shares  $\delta_i$  in the calibration point. We then discarded the previously estimated  $\gamma$  and instead solve (1) for  $\gamma$  at given  $x$ ,  $Y$ ,  $p$  and the calibrated marginal budget shares. In the case that this implies a negative  $\gamma$ , we use a penalty minimization approach, which minimizes the difference between the estimated  $\alpha$ ,  $\beta$  and the “corrected” ones, such that all  $\gamma$  turn out to be positive.

<sup>41</sup> Muhammad, A., Seale, J. L., Meade, B., & Regmi, A. (2011). International evidence on food consumption patterns: an update using 2005 international comparison program data. (March 1, 2011). USDA-ERS Technical Bulletin No. 1929. Available at SSRN: <https://ssrn.com/abstract=2114337> or <http://dx.doi.org/10.2139/ssrn.2114337>

<sup>42</sup> Cranfield, J. A., Preckel, P. V., Eales, J. S., & Hertel, T. W. (2000). On the estimation of an implicitly additive demand system'. *Applied Economics*, 32(15), 1907-1915

<sup>43</sup> ICP (2015). Purchasing Power Parities and Real Expenditures of World Economics: A Comprehensive Report of the 2011 International Comparison Program. World Bank, Washington, D.C.

## Technical implementation of the MAIDADS demand system

Note that the equations as discussed in here refer to the updated estimation by Britz 2021<sup>44</sup> who uses an extended data set from IPC with more agro-food detail and introduces additional controls.

### Equations

Due to its resemblance with the LES system, which was already available in CGEBox, the code is a direct extension of the existing code both for the calibration to the benchmark and implementation of the demand system in the model's equation (see model\cal\_les.gms and model\dem\_les.gms).

The demand equations for both systems are indeed identical:

```
*
* --- demand equations for individual products (if not not under sub-nest, this is checked by flag p_hasXA)
*
e xacLes(rs,i,h,ts) $ p_hasXA(rs,i,h) ..
    xa(rs,i,h,ts) /xa.scale(rs,i,h,ts)
    =e= (
        $$iftheni.baseline "%fixCalories%"=="on"
            gammaLES(rs,i,h,ts)
        $$else.baseline
            gammaLESM(rs,i,h,ts)
        $$endif.baseline
        /1000*pop(rs,h,ts) + alphaLES(rs,i,h,ts)/m_pa(rs,i,h,ts) * yCNonCom(rs,h,ts))/xa.scale(rs,i,h,ts);
```

The demand is equal to a constant term gammaLES respectively gammaLESM per head multiplied with population in the household type h (which might be just one) plus the marginal budget share alphaLES times the non-committed income yCNonCom, divided by the Armington price index for the private household aggregate, defined in the macro m\_pa.

The non-committed income yNonCom is the difference between the given expenditures for final consumption yc and the value of the commitments, i.e. the constant terms multiplied with the Armington price:

```
*
* --- non committed income: total expenditure for private consumption minus
* value of commitments. Note that top-level nests (only) of the nested
* demand system are taken into account. Lower nests are all CES-functions
*
e yCNonCom(rs,h,ts) $ sum(i,xaFlag(rs,i,h)) ..
    ycNonCom(rs,h,ts)/yc.scale(rs,h,ts)
    =E= (yc(rs,h,ts)
    *
    * --- note: commitment terms are defined in a per capita basis
    *
    $$iftheni.baseline "%fixCalories%"=="on"
        - sum(i $ p_hasXA(rs,i,h), gammaLES(rs,i,h,ts) *pop(rs,h,ts)/1000 * m_pa(rs,i,h,ts))
        - sum(dNest_n_fd("top",dNest,h) $ p_hasXA(rs,dNest,h), gammaLES(rs,dNest,h,ts) *pop(rs,h,ts)/1000 * pdNest(rs,dNest,h,ts))/yc.scale(rs,h,ts);
    $$else.baseline
        - sum(i $ p_hasXA(rs,i,h), gammaLESM(rs,i,h,ts) *pop(rs,h,ts)/1000 * m_pa(rs,i,h,ts))
        - sum(dNest_n_fd("top",dNest,h) $ p_hasXA(rs,dNest,h), gammaLESM(rs,dNest,h,ts) *pop(rs,h,ts)/1000 * pdNest(rs,dNest,h,ts))/yc.scale(rs,h,ts);
    $$endif.baseline
```

Note that we allow for demand nests dNest which are CES aggregates of other such nests or individual commodity demands.

In the case of AIDADS, the marginal budget shares alphaLES are not fixed (as in the LES), but a function of utility uh:

<sup>44</sup> Britz, W. (2021): Estimating a global MAIDADS demand system considering demography, climate and norms, Bio-based and Applied Economics, in: online available 19 July 2021



```

*
* --- AIDADS: marginal budget shares are a function of utility (individual products or nests)
*
e_alphaAIDADS(rs,i,h,ts(tRun)) $ (xaFlag(rs,i,h) $ (alphaLES.range(rs,i,h,tRun) ne 0) $ p_hasXA(rs,i,h)
                                $ (alphaAIDADS.l(rs,i,h) + betaAIDADS.l(rs,i,h))
                                $ (p_demSystem eq AIDADS) $ hFlag(rs,h) $ hCur(h))..

alphaLES(rs,i,h,tRun)/(alphaAIDADS.l(rs,i,h) + betaAIDADS.l(rs,i,h) + 1.E-6)
=E= { [ alphaAIDADS(rs,i,h) + betaAIDADS(rs,i,h) * sqExp(uhExpD(rs,h,tRun),10) ]
      / [ 1 + sqExp(uhExpD(rs,h,tRun),10) ] }

*
* --- shifter terms to meet calorie intake
*
+ [alphaLES.l(rs,i,h,tRun-1) $ (alphaLES.l(rs,i,h,tRun-1) gt 1.E-8)]
  * ( alphaShiftF(rs,h,tRun) $ foodCal(i)
      + alphaShiftNF(rs,h,tRun) $ (not foodCal(i))) $ (not sameas(tRun,"%t0%"))
} / (alphaAIDADS.l(rs,i,h) + betaAIDADS.l(rs,i,h) + 1.E-6);

```

The two vectors `alphaAIDADS` and `betaAIDADS` are the limiting cases for the marginal budget shares at very low and very high utility levels. The utility of the (representative) private household is set to unity in the benchmark. In order to line up with the empirical implementation (see next section), the parameter `p_uh0` is introduced.

All `alphaAIDADS` might change endogenously according to a uniform shifter `alphaShiftF` for food product and `alphaShiftNF` for non-food items, which have the opposite sign. These shifters are driven by fixing calorie intakes as discussed below.

Cardinal utility is thus defined as follows:

```

*
* --- indirect (and implicate) utility definition in MAIDADS
*
e_uhAIDADS(rs,h,ts) $ (sum(i,xaFlag(rs,i,h)) $ (p_demSystem eq AIDADS)) ..

uh(rs,h,ts) =E= sqexp(uhl(rs,h,ts),10)/p_uh0(rs,h);

e_uh1AIDADS(rs,h,ts) $ (sum(i,xaFlag(rs,i,h)) $ (p_demSystem eq AIDADS)) ..

uhl(rs,h,ts) =E= log(ycNonCom(rs,h,ts)/pop(rs,h,ts) * 1000)
+ sum(i $ (p_hasXA(rs,i,h) $ (alphaAIDADS.l(rs,i,h) + betaAIDADS.l(rs,i,h))),
  alphaLES(rs,i,h,ts) * log(alphaLES(rs,i,h,ts)/m_pa(rs,i,h,ts)))
+ sum(dNest $ (p_hasXA(rs,dNest,h) $ (alphaAIDADS.l(rs,dNest,h) + betaAIDADS.l(rs,dNest,h))),
  alphaLES(rs,dNest,h,ts) * log(alphaLES(rs,dNest,h,ts)/pdNest(rs,dNest,h,ts)))
- v_kappa.l(rs,h,"U");

```

where `p_uh0` is an additional parameter in the demand system (corresponding to the parameter `A` in Cranfield et al. 2000, see their equation (10)). Note the equivalence of the utility definition to the one in the LES system: only consumption quantities exceeding the commitments `gammaLES` generate utility.

The cardinal utility is mapped into a driver of the logistic functions of the marginal budget shares and the commitment terms:

```

*
* --- utility level is transformed of a driver of the logistic functions for the marginal
* budget shares
*
e_uhExpG(rs,h,ts) $ (sum(i,xaFlag(rs,i,h)) $ (p_demSystem eq AIDADS) $ hFlag(rs,h) $ hCur(h)) ..

(uhExpG(rs,h,ts)) =E= (p_parAIDADS("UG","cnst") + v_kappa(rs,h,"UGC") + (v_kappa(rs,h,"UGL") + p_parAIDADS("UG","lin")) * uhl(rs,h,ts));

*
* --- utility level is transformed of a driver of the logistic functions of the commitment terms
*
e_uhExpD(rs,h,ts) $ (sum(i,xaFlag(rs,i,h)) $ (p_demSystem eq AIDADS) $ hFlag(rs,h) $ hCur(h)) ..

(uhExpD(rs,h,ts)) =E= (p_parAIDADS("UD","cnst") + v_kappa(rs,h,"UDC") + (v_kappa(rs,h,"UDL") + p_parAIDADS("UD","lin")) * uhl(rs,h,ts));

```

We also estimate the elasticity of private expenditure with regard to private utility:

```

e_phiPLES(rsNat,ts) $ (phip.range(rsNat,ts) ne 0) ..
    phip(rsNat,ts) =e=
        sum( r_r(rs,rsNat) $ sum(h $ hFlag(rs,h),1),
*
*   --- population weighted 1/[ du/dm]  from indirect utility function
*
        sum(h $ hFlag(rs,h), pop(rs,h,ts) * prod(i $ (p_hasXa(rs,i,h) $ alphaLES.1(rs,i,h,"%t0%")),
            (m_pa(rs,i,h,ts)/alphaLES(rs,i,h,ts))**alphaLES(rs,i,h,ts))
        * prod(dNest_n_fd("top",dNest,h) $ (p_hasXa(rs,dNest,h) $ alphaLES.1(rs,dNest,h,"%t0%")),
            (pdNest(rs,dNest,h,ts)/alphaLES(rs,dNest,h,ts))**alphaLES(rs,dNest,h,ts)))
        /sum(h $ hFlag(rs,h), pop.1(rs,h,"%t0%") * prod(i $ (p_hasXa(rs,i,h) $ alphaLES.1(rs,i,h,"%t0%")),
            (pa.1(rs,i,h,"%t0%")/alphaLES.1(rs,i,h,"%t0%"))**alphaLES.1(rs,i,h,"%t0%"))
        * prod(dNest_n_fd("top",dNest,h) $ (p_hasXa(rs,dNest,h) $ alphaLES.1(rs,dNest,h,"%t0%")),
            (pdNest.1(rs,dNest,h,"%t0%")/alphaLES.1(rs,dNest,h,"%t0%"))**alphaLES.1(rs,dNest,h,"%t0%")))
*
*   --- conversion to elasticity
*
        * sum(h $ hFlag(rs,h), pop(rs,h,ts) * uh(rs,h,ts))/sum(h $ hFlag(rs,h), pop(rs,h,ts) * yc(rs,h,ts))
*
*   --- and correct for benchmark relation and weight with population
*
        * sum(h $ hFlag(rs,h), pop.1(rs,h,"%t0%") * yc.1(rs,h,"%t0%"))
        /sum(h $ hFlag(rs,h), pop.1(rs,h,"%t0%") * uh.1(rs,h,"%t0%"))
        )
        /sum((r_r(rs,rsNat)) $ sum(h,hFlag(rs,h),1);

```

## Controlling for calorie intakes as a function of per capita income

In order to arrive at a plausible development of calorie intakes, an empirical estimated relation between calories and per-capita income is used,  $Y$  is per capita in income in USD and  $T$  is mean temperature:

$$\begin{aligned}
 calTot^* = & -947,275 - 16,257 T \\
 & + 738,910 \ln(Y) - 30,083 \ln(Y)^2
 \end{aligned} \tag{59}$$

The integration into the modelling framework needs to consider the error at the benchmark. Therefore, only changes in calories equation (59) from changes in income are considered in the second part of equation. There are two additional elements in the equations: (1) The first block defines a weighted average between the equation (59) and its first differences in the second block, such that after two hundredth years, the simulation would be identical to the empirically estimated relations. (2) the actual intake and the estimate at the benchmark is multiplied with the first differneces, such that countries which an above average intake will also tend to expand their calorie stronger.

```

e_caloriesShift(rs,h,ts(tRun)) $ (hFlag(rs,h) $ (not sameas(ts,"%t0%")) $ (alphaShiftF.range(rs,h,tRun) ne 0) ) ..
    v_nutrients(rs,h,"cal",ts)/v_nutrients.scale(rs,h,"cal",ts) =E=
    [
        $$iftheni.dyn not "%dynMode%"=="Comparative static"
*
*   --- draw the calorie intakes in year t0+200 to the estimates
*
        [1-( 201-sum(sameas(t,tRun),t.pos))/200]*
            {p_calCnst(rs) + 738.910 * log( yc(rs,h,tRun)/(pop(rs,h,tRun)*pCons(rs,h,tRun))*1000)
              - 30.083 * sqr(log( yc(rs,h,tRun)/(pop(rs,h,tRun)*pCons(rs,h,tRun))*1000)) }
*
*   --- second term updates estimates depending on difference between benchmark results
*   and estimates
*
        + [(201-sum(sameas(t,tRun),t.pos))/200]*
    $$endif.dyn

```

```

*
*
*   --- last year's intake
*
*   v_nutrients.l(rs,h,"cal",tRun-1)
*
*   --- add to last year the change implied by the income increase
*   (income dependent betas from regression)
*
*   + ( 738.910 * [ log( yc(rs,h,tRun) / (pop(rs,h,tRun) * pCons(rs,h,tRun)) * 1000)
*     - log( yc.l(rs,h,tRun-1) / (pop.l(rs,h,tRun-1) * pCons.l(rs,h,tRun-1)) * 1000) ]
*     - 30.083 * [ sqrt(log( yc(rs,h,tRun) / (pop(rs,h,tRun) * pCons(rs,h,tRun)) * 1000))
*       - sqrt(log( yc.l(rs,h,tRun-1) / (pop.l(rs,h,tRun-1) * pCons.l(rs,h,tRun-1)) * 1000)) ]
*   )
*
*   --- in case of benchmark intake below estimate, dampen by the relation between the actual intake
*   and estimates
*
*   * min(10.0, (v_nutrients.l(rs,h,"cal", "%t0%") / {p_calCnst(rs) + 738.910 * log( yc.l(rs,h, "%t0%") / pop.l(rs,h, "%t0%") * 1000)
*     - 30.083 * sqrt(log( yc.l(rs,h, "%t0%") / pop.l(rs,h, "%t0%") * 1000)) })))
*
*   ] / v_nutrients.scale(rs,h,"cal",ts);

```

Which can be depicted in mathematical notation as:

$$\begin{aligned}
 calTot_t = & \left(1 - \frac{201-t}{200}\right) calTot_t^* \\
 & + \frac{201-t}{200} \left( calTot_{t-1}^* + \frac{calTot_0}{calTot_0^*} [calTot_t^* - calTot_{t-1}^*] \right)
 \end{aligned} \tag{60}$$

As the RHS of this equation is exogenous if income is fixed during baseline construction, an endogenous variable needs to adjust. This is the shifter  $\alpha ShiftF$  for food products in the  $e\_alphaAidAds$  equation. It will be added to the marginal budget share as driven by the cardinal utility. As the budget shares need to add up to unity, a positive/negative shifter for food products (which is equivalent of increasing/decreasing the calorie intake) requires a shift of the marginal budget shares for non-food products in the opposite direction.

The correction factors for the marginal budget shares are integrated as follows where the endogenous correction factor  $cor$  differs for food  $f$  and non-food items  $nf$  and are defined relative to last year's marginal budget share, i.e. the corrections are defined in relative terms:

$$\delta_{i,t}^* = \delta_{i,t} + \delta_{i,t-1}^* \left( cor_f \wedge i \in f + cor_{nf} \wedge i \in nf \right) \tag{61}$$

For food items,  $cor_f$  it is implicitly determined by the calorie constraints (60). For non-food items,  $cor_{nf}$  is driven by the adding up conditions for marginal budget shares:

$$\sum_i \delta_{i,t-1}^* \left( cor_f \wedge i \in f + cor_{nf} \wedge i \in nf \right) = 0 \tag{62}$$

The correction for the commitment terms uses last year's per capita demands as weights. Using large year's marginal budget shares in (61) and last year's per capita demand in (62) should ensure that the provoked changes do not fundamentally change the composition inside of the food and inside of the non-food commodity bundle from what the benchmark step suggests.

The correction is also applied to the commitment terms:

```

*
* --- MAIDADS: commitments are a function of utility (individual products or nests)
*
e_gammaAIDADS(rs,i,h,ts(tRun)) $ ( (p_demSystem eq AIDADS) $ (gammaLESM.range(rs,i,h,tRun) ne 0)
$ p_hasXa(rs,i,h) $ hFlag(rs,h) $ hCur(h) )..

gammaLESM(rs,i,h,tRun)/(abs(gammaLo.scale(rs,i,h) + gammaHi.scale(rs,i,h)) + 1)
=E= { ([ gammaLo.scale(rs,i,h) + gammaHi.scale(rs,i,h) * sqExp(uhExpG(rs,h,tRun),10)]
/ [ 1 + sqExp(uhExpG(rs,h,tRun),10)] ) $ abs(gammaLo.scale(rs,i,h) + gammaHi.scale(rs,i,h))

*
* --- G-RDEM might introduce shifters to the commitment terms to let total calorie
* demand follow the calorie supply as a function of the household s per capita income
+ [ (alphaShiftF(rs,h,tRun) $ foodCal(i) + alphaShiftNF(rs,h,tRun) $ (not foodCal(i)))
* max(xa.1(rs,i,h,tRun-1)/pop.1(rs,h,ts)*1000,gammaLESM.1(rs,i,h,tRun-1))
] $ ((not sameas(tRun,"%t0%")) $ ( (alphaShiftF.range(rs,h,tRun) ne 0) or alphaShiftF.1(rs,h,tRun) ne 0) )

)/(abs(gammaLo.scale(rs,i,h) + gammaHi.scale(rs,i,h)) + 1);

e_gammaDAIDADS(rs,dNest,h,ts(tRun)) $ ( (p_demSystem eq AIDADS) $ (gammaLESM.range(rs,dNest,h,tRun) ne 0)
$ p_hasXa(rs,dNest,h) )..

gammaLESM(rs,dNest,h,tRun)/(abs(gammaLo.scale(rs,dNest,h) + gammaHi.scale(rs,dNest,h)) + 1)
=E= { ([ gammaLo.scale(rs,dNest,h) + gammaHi.scale(rs,dNest,h) * sqExp(uhExpG(rs,h,tRun),10)]
/ [ 1 + sqExp(uhExpG(rs,h,tRun),10)] ) $ abs(gammaLo.scale(rs,dNest,h) + gammaHi.scale(rs,dNest,h))

*
* --- G-RDEM might introduce shifters to the commitment terms to let total calorie
* demand follow the calorie supply as a function of the household s per capita income
+ [ (alphaShiftF(rs,h,tRun) $ foodCal(dNest) + alphaShiftNF(rs,h,tRun) $ (not foodCal(dNest)))
* max(xdNest.1(rs,dNest,h,tRun-1)/pop.1(rs,h,ts)*1000,gammaLESM.1(rs,dNest,h,tRun-1))
] $ ((not sameas(tRun,"%t0%")) $ ( (alphaShiftF.range(rs,h,tRun) ne 0) or (alphaShiftF.1(rs,h,tRun) ne 0) ) )

)/(abs(gammaLo.scale(rs,dNest,h) + gammaHi.scale(rs,dNest,h)) + 1)
;
    
```

The correction for the commitment terms uses last year's per capita demands  $xa$  respectively  $xdNest$  as weights. Using large year's marginal budget shares in (61) and last year's per capita demand in (62) should ensure that the provoked changes do not fundamentally change the composition inside of the food and inside of the non-food commodity bundle from what the benchmark step suggests.

Large commitment terms relative to the demand lead to a very low price elasticity and can provoke problems with solving the model. We fudge therefore the updated commitments with a Veelken-Ulbrich smooth min operator at 90% of last year's simulated per capita demands, see (63). This operator is an inbuilt function by GAMS such that it provides derivative information to the solver which might be better compared to, for instance, programming directly the otherwise somewhat simpler Huber max approximation. While we don't allow for negative commitment terms during benchmarking, the correction process could actually introduce negative ones which is another argument to control for calorie intakes at higher income level already during benchmarking to keep the changes from the correction small.

$$\gamma_{it}^* = -VU \left[ \begin{array}{l} -\gamma_{i,t} - \frac{xa_{i,t-1}}{pop_{t-1}} \left( cor_f \wedge i \in f + cor_{nf} \wedge i \in nf \right), \\ -\frac{9}{10} \frac{xa_{i,t-1}}{pop_{t-1}} \end{array} \right] \quad (63)$$

The code implementation looks as follows:

```

*
* --- Veelken-Ulbrich smoothed min to prevent that the gamma shifts provoke negative demands
*
e_gammaAIDADS(rs,dNest,h,ts(tRun)) $ ( (p_demSystem eq AIDADS) $ (gammaLES.range(rs,dNest,h,tRun) ne 0)
$ p_hasXa(rs,dNest,h) ) ..

gammaLES(rs,dNest,h,tRun) / (abs(gammaLo.scale(rs,dNest,h) + gammaHi.scale(rs,dNest,h)) + 1)
=E= [ [ -ncpVUSin(-gammaLESM(rs,dNest,h,tRun),
-(-0.90*max(0,xdNest.1(rs,dNest,h,tRun-1))/pop.1(rs,h,tRun-1)*1000),abs(xdNest.1(rs,dNest,h,tRun-1))/pop.1(rs,h,tRun-1)))
$ ( (not sameas(tRun,"%t0%")) $ ( (alphaShiftF.range(rs,h,tRun) ne 0) or (alphaShiftF.1(rs,h,tRun) ne 0)))
+ gammaLESM(rs,dNest,h,tRun)
$ ( sameas(tRun,"%t0%")) or ( (alphaShiftF.range(rs,h,tRun) eq 0) and (alphaShiftF.1(rs,h,tRun) eq 0)))
]
/ (abs(gammaLo.scale(rs,dNest,h) + gammaHi.scale(rs,dNest,h)) + 1);

e_gammaAIDADS(rs,i,h,ts(tRun)) $ ( (p_demSystem eq AIDADS) $ (gammaLES.range(rs,i,h,tRun) ne 0)
$ p_hasXa(rs,i,h) $ hFlag(rs,h) $ hCur(h) ) ..

gammaLES(rs,i,h,tRun) / (abs(gammaLo.scale(rs,i,h) + gammaHi.scale(rs,i,h)) + 1)
=E= [ [ -ncpVUSin(-gammaLESM(rs,i,h,tRun),
-(-0.90*max(0,xa.1(rs,i,h,tRun-1))/pop.1(rs,h,tRun-1)*1000),abs(xa.1(rs,i,h,tRun-1))/pop.1(rs,h,tRun-1)))
$ ( (not sameas(tRun,"%t0%")) $ ( (alphaShiftF.range(rs,h,tRun) ne 0) or (alphaShiftF.1(rs,h,tRun) ne 0)))
+ gammaLESM(rs,i,h,tRun)
$ ( sameas(tRun,"%t0%")) or ( (alphaShiftF.range(rs,h,tRun) eq 0) and (alphaShiftF.1(rs,h,tRun) eq 0)))
] / (abs(gammaLo.scale(rs,i,h) + gammaHi.scale(rs,i,h)) + 1);

```

## Calibration

The calibration code can be found in “model/cal\_les.gms”. The econometrically estimated parameters are read from a GDX file:

```

parameter p_AIDADEst(*,*);
set dummyAidAdsEst / "islam+demo+temp", "gammaUp", "lo", "hi", "kappaUD", "kappaUG", "lin", "cnst", "u", "log", "logSqr", "logSqrt" /;
execute_load "%datdir%/maidads.gdx" p_AIDADEst;

```

Integration into a GTAP model with flexible sectoral aggregation requires a mapping from these categories to the current sectors in the model. That is achieved in two steps:

1. The commodity groups from the AIDADS estimation are mapped to the 57 respectively 65 GTAP sectors (non-aggregated):
2. Next, the mapping mapi from the 57 respectively 65 GTAP sectors to the current aggregation is used to assign parameters to the commodities, as defined in the current run:

```

AIDADS_to_GTAP("Miscellaneous goods and services",idat) $ (not sum(AIDADS_to_GTAP(AIDADS_grps,idat),1)) = yes;

set AIDADS_to_i(AIDADS_grps,i) "Link to current sectoral aggregation";
AIDADS_to_i(AIDADS_grps,i) $ sum(AIDADS_to_GTAP(AIDADS_grps,idat),mapi(idat,i)) = YES;

```

3. The estimated parameters are assigned to the products, using budget shares as weights:

```

alphaAIDADS(r,i,h) $ xcshr.1(r,i,h,"%t0%")
= max(1.E-6,sum(AIDADS_to_i(AIDADS_grps,i),
(p_parAIDADS(AIDADS_grps,"alpha")
+sum((r_r(r,rp),deltaDrivers),p_deltaShifters(rp,deltaDrivers)*p_deltaShift(AIDADS_grps,deltaDrivers,"alpha"))
*xcshr.1(r,i,h,"%t0%"))
/sum(AIDADS_to_i(AIDADS_grps,j) $ AIDADS_to_i(AIDADS_grps,i),xcshr.1(r,j,h,"%t0%"))));

betaAIDADS(r,i,h) $ xcshr.1(r,i,h,"%t0%")
= max(1.E-6,sum(AIDADS_to_i(AIDADS_grps,i),
(p_parAIDADS(AIDADS_grps,"beta")
+sum((r_r(r,rp),deltaDrivers),p_deltaShifters(rp,deltaDrivers)*p_deltaShift(AIDADS_grps,deltaDrivers,"beta"))
*xcshr.1(r,i,h,"%t0%"))
/sum(AIDADS_to_i(AIDADS_grps,j) $ AIDADS_to_i(AIDADS_grps,i),xcshr.1(r,j,h,"%t0%"))));

gammaLo(r,i,h) $ xcshr.1(r,i,h,"%t0%")
= sum(AIDADS_to_i(AIDADS_grps,i), p_parAIDADS(AIDADS_grps,"gammaLo")*xcshr.1(r,i,h,"%t0%"))
/sum(AIDADS_to_i(AIDADS_grps,j) $ AIDADS_to_i(AIDADS_grps,i),xcshr.1(r,j,h,"%t0%")));

gammaHi(r,i,h) $ xcshr.1(r,i,h,"%t0%")
= sum(AIDADS_to_i(AIDADS_grps,i), p_parAIDADS(AIDADS_grps,"gammaHi")*xcshr.1(r,i,h,"%t0%"))
/sum(AIDADS_to_i(AIDADS_grps,j) $ AIDADS_to_i(AIDADS_grps,i),xcshr.1(r,j,h,"%t0%")));

```

In order to ensure that both vectors fulfil the adding up condition exactly, the elements of in the vector alphaAIDADS and betaAIDADS are scaled such their sum is equal to unity:

```

alphaAIDADS.fx(r,i,h,t0) $ xcshr.1(r,i,h,t0) = alphaAIDADS(r,i,h,t0)/sum(j,alphaAIDADS(r,j,h,t0));
betaAIDADS.fx(r,i,h,t0) $ xcshr.1(r,i,h,t0) = betaAIDADS(r,i,h,t0)/sum(j,betaAIDADS(r,j,h,t0));

```

The cardinal utility level, which drives the marginal budget composition, is an estimated parameter. To this end, Roson and van der Mensbrugghe 2018 regress the estimated utility levels on real GDP per



capita. The similar regression results from Britz 2021 are here employed to derive the utility level at the benchmark and thus the marginal budget shares:

```
Uh1.l(r,h,t0) $ p_perCapInc(r,h,"%t0%")
= (p_estU("cns") + p_estU("log") * log(p_perCapInc(r,h,"%t0%"))
+ p_estU("logSqr") * sqrt(log(p_perCapInc(r,h,"%t0%")))
+ p_estU("logSqrt") * sqrt(log(p_perCapInc(r,h,"%t0%")))
+ sum( (r_r(r,rNat),deltaDrivers), p_estu(deltaDrivers) * p_deltaShifters(rnat,deltaDrivers)
);
```

The obtained marginal budget shares  $\alpha$ LES at benchmark income and prices are then used to define the commitment terms. The code is not shown here as it was already comprised in CGEBox code and it is illustrated in the associated documentation.

Whenever that calibration procedure becomes infeasible, some deviations from the given  $\alpha$ s and  $\beta$ s are allowed:

```
if ( (fitLes.suminfes ne 0) and ( p_demSystem eq AIDADS) ,
*
*   --- AIDADS failed, GAMMA bounds became binding, release bounds on alpha and beta
*
alphaAIDADS.lo(rNat,i,h,t0) = min(alphaAIDADS.l(rNat,i,h,t0) * 0.00001,alphaAIDADS.l(rNat,i,h,t0) * 1000);
alphaAIDADS.up(rNat,i,h,t0) = max(0.1,alphaAIDADS.l(rNat,i,h,t0) * 0.00001,alphaAIDADS.l(rNat,i,h,t0) * 1000);

betaAIDADS.lo(rNat,i,h,t0) = min(betaAIDADS.l(rNat,i,h,t0) * 0.00001,betaAIDADS.l(rNat,i,h,t0) * 1000);
betaAIDADS.up(rNat,i,h,t0) = max(0.1,betaAIDADS.l(rNat,i,h,t0) * 0.00001,betaAIDADS.l(rNat,i,h,t0) * 1000);
solve fitLES using NLP minimizing fit;
);
```

In that case, squared relative differences between the corrected and a priori marginal budget shares enter the objective function:

```
e_fitLes ..
fit * sum ( (rs(r),i,h,t0) $ xaFlag(r,i,h),1)
=E= sum ( (rs(r),i,h,t0) $ xaFlag(r,i,h),
sqr( pElas(r,i,h,t0)-pElas.l(r,i,h,t0))*0.01
+ sqr( incElas(r,i,h,t0)-incElas.l(r,i,h,t0)) )
*
*   --- squared relative deviations from given parameters in case of AIDADS
*   if gamma bounds become binding
*
+ sum ( (rs(r),i,h,t0) $ xaFlag(r,i,h), sqr(v_scaleAlpha(r,i,h)-1)) $ (p_demSystem eq AIDADS);
```

Where  $v\_scaleAlpha$  enters the definition of  $\alpha$ LES and  $\beta$ LES:

```
e_alphaAIDADS1(rs(r),i,h,t0) $ (alphaAIDADS.l(r,i,h,t0) + betaAIDADS.l(r,i,h,t0) $ (p_demSystem eq AIDADS))..
alphaLES(r,i,h,t0) =E= [ alphaAIDADS(r,i,h,t0)*v_scaleAlpha(r,i,h)
+ betaAIDADS(r,i,h,t0)*v_scaleAlpha(r,i,h) * ( uh(r,h,t0)*p_uh0(r,h)) ]
/ [ 1 + uh(r,h,t0)*p_uh0(r,h) ];
```

Such as:

```
--- marginal budget shares must add to unity

e_sumAlpha(rs(r),h,t0) $ ( (p_demSystem eq AIDADS) $ sum(i,xaFlag(r,i,h))) ..
sum(i $ xaFlag(r,i,h), alphaAIDADS(r,i,h,t0)*v_scaleAlpha(r,i,h)) =E= 1;

e_sumBeta(rs(r),h,t0) $ ( (p_demSystem eq AIDADS) $ sum(i,xaFlag(r,i,h))) ..
sum(i $ xaFlag(r,i,h), betaAIDADS(r,i,h,t0)*v_scaleAlpha(r,i,h)) =E= 1;
```

Finally, a multiplier  $p\_uh1$  (termed A in Cranfield et al. 2000, eq 10) is calculated:

```

$iftheni.AIDADS "%demSystem%" == "AIDADS"

Uh.lo(r,h,t0) = 0.001;
Uh.up(r,h,t0) = inf;

p_uhl(r,h) = { sum[i $ alphaLES(r,i,h,"%t0%"),
                    alphaLES(r,i,h,"%t0%") * log( xa(r,i,h,"%t0%") - gammaLES(r,i,h,"%t0%")*pop.l(r,h,"%t0%"))]]
+sum[dNest $ alphaLES(r,dNest,h,"%t0%"),
          alphaLES(r,dNest,h,"%t0%") * log( xdNest(r,dNest,h,"%t0%") - gammaLES(r,dNest,h,"%t0%")*pop.l(r,h,"%t0%"))]]
}
- (log(uh.l(r,h,"%t0%"))+p_uh0(r,h));

$endif.AIDADS

```

The code allows introducing aggregates of individual commodities under which CES-sub-nests can be defined, which comprise these individual commodities.

**Table 8: Mapping between AIDADS categories and GTAP sectors**

AIDADS category	GTAP sector	Detailed description
Cereals	pdr wht gro pcr	Paddy Rice: rice, husked and unhusked Wheat: wheat and meslin Other Grains: maize (corn), barley, rye, oats, other cereals Processed Rice: rice, semi- or wholly milled
Ruminant meat	Ctl  cmt	Cattle: cattle, sheep, goats, horses, asses, mules, and hinnies; and semen thereof Cattle Meat: fresh or chilled meat and edible offal of cattle, sheep, goats, horses, asses, mules, and hinnies. raw fats or grease from any animal or bird
Other animal products	Oap  omt	Other Animal Products: swine, poultry and other live animals; eggs, in shell (fresh or cooked), natural honey, snails (fresh or preserved) except sea snails; frogs' legs, edible products of animal origin n.e.c., hides, skins and furskins, raw , insect waxes and spermaceti, whether or not refined or coloured Other Meat: pig meat and offal. preserves and preparations of meat, meat offal or blood, flours, meals and pellets of meat or inedible meat offal; greaves
Fish	fsH	Fishing: hunting, trapping and game propagation including related service activities, fishing, fish farms; service activities incidental to fishing
Dairy	mil	Milk: dairy products
Milk	rmk	Raw milk
Oils_Fats	Osd vol	Oil Seeds: oil seeds and oleaginous fruit; soy beans, copra Vegetable Oils: crude and refined oils of soya-bean, maize (corn),olive, sesame, ground-nut, olive, sunflower-seed, safflower, cotton-seed, rape, colza and canola, mustard, coconut palm, palm kernel, castor, tung jojoba, babassu and linseed, perhaps partly or wholly hydrogenated,inter-esterified, re-esterified or elaidinised. Also margarine and similar preparations, animal or vegetable waxes, fats and oils and their fractions, cotton linters, oil-cake and other solid residues resulting from the extraction of vegetable fats or oils; flours and meals of oil seeds or oleaginous fruits, except those of mustard; degreas and other residues resulting from the treatment of fatty substances or animal or vegetable waxes.
Fruits_vegs	v_f  ocr	Veg & Fruit: vegetables, fruitvegetables, fruit and nuts, potatoes, cassava, truffles Other Crops: live plants; cut flowers and flower buds; flower seeds and fruit seeds; vegetable seeds, beverage and spice crops, unmanufactured tobacco, cereal straw and husks, unprepared, whether or not chopped, ground, pressed or in the form of pellets; swedes, mangolds, fodder roots, hay, lucerne (alfalfa), clover, sainfoin, forage kale, lupines, vetches and similar forage



AIDADS category	GTAP sector	Detailed description
	c_b	products, whether or not in the form of pellets, plants and parts of plants used primarily in perfumery, in pharmacy, or for insecticidal, fungicidal or similar purposes, sugar beet seed and seeds of forage plants, other raw vegetable materials Cane & Beet: sugar cane and sugar beet
Food_other	ofd	Other Food: prepared and preserved fish or vegetables, fruit juices and vegetable juices, prepared and preserved fruit and nuts, all cereal flours, groats, meal and pellets of wheat, cereal groats, meal and pellets n.e.c., other cereal grain products (including corn flakes), other vegetable flours and meals, mixes and doughs for the preparation of bakers' wares, starches and starch products; sugars and sugar syrups n.e.c., preparations used in animal feeding, bakery products, cocoa, chocolate and sugar confectionery, macaroni, noodles, couscous and similar farinaceous products, food products n.e.c.
Sugar	sgf	sugar
Beverages and tobacco	b_t	Beverages and Tobacco products
Cothing	tex wap lea  wol pfb	Textiles: textiles and man-made fibres Wearing Apparel: Clothing, dressing and dyeing of fur Leather: tanning and dressing of leather; luggage, handbags, saddlery, harness and footwear Wool: wool, silk, and other raw animal materials used in textile Plant Fibres: cotton, flax, hemp, sisal and other raw vegetable materials used in textiles
Housing	dwe  obs wtr gdt  ely coa oil  gas  isr	Dwellings: ownership of dwellings (imputed rents of houses occupied by owners) Other Business Services: real estate, renting and business activities Water: collection, purification and distribution Gas Distribution: distribution of gaseous fuels through mains; steam and hot water supply Electricity: production, collection and distribution Coal: mining and agglomeration of hard coal, lignite and peat Oil: extraction of crude petroleum and natural gas (part), service activities incidental to oil and gas extraction excluding surveying (part) Gas: extraction of crude petroleum and natural gas (part), service activities incidental to oil and gas extraction excluding surveying (part) Insurance: includes pension funding, except compulsory social security
Furniture, Maintenance, appliances	p_c  frs omn  lum  crp  nmm i_s nfm  fmp	Petroleum & Coke: coke oven products, refined petroleum products, processing of nuclear fuel Forestry: forestry, logging and related service activities Other Mining: mining of metal ores, uranium, gems. other mining and quarrying  Lumber: wood and products of wood and cork, except furniture; articles of straw and plaiting materials Chemical Rubber Products: basic chemicals, other chemical products, rubber and plastics products Non-Metallic Minerals: cement, plaster, lime, gravel, concrete Iron & Steel: basic production and casting Non-Ferrous Metals: production and casting of copper, aluminium, zinc, lead, gold, and silver  Fabricated Metal Products: Sheet metal products, but not machinery and

AIDADS category	GTAP sector	Detailed description
	cns ele  ome omf	equipment Construction: building houses factories offices and roads Electronic Equipment: office, accounting and computing machinery, radio, television and communication equipment and apparatus Other Machinery & Equipment: electrical machinery and apparatus n.e.c., medical, precision and optical instruments, watches and clocks Other Manufacturing: includes recycling
Transport	Otp  wtp atp mvh  otn  Isr  p_c	Other Transport: road, rail ; pipelines, auxiliary transport activities; travel agencies Water transport Air transport  Motor Motor vehicles and parts: cars, lorries, trailers and semi-trailers Other Transport Equipment: Manufacture of other transport equipment nsurance: includes pension funding, except compulsory social security Petroleum & Coke: coke oven products, refined petroleum products, processing of nuclear fuel
Communication	cmn	Communications: post and telecommunications
Recreation	ros  ppp	Recreation & Other Services: recreational, cultural and sporting activities, other service activities; private households with employed persons (servants) Paper & Paper Products: includes publishing, printing and reproduction of recorded media
Education	osg	Other Services (Government): public administration and defense; compulsory social security, education, health and social work, sewage and refuse disposal, sanitation and similar activities, activities of membership organizations n.e.c., extra-territorial organizations and bodies
Health	Osg	See above
Restaurants and hotels	Trd	Trade: all retail sales; wholesale trade and commission trade; hotels and restaurants; repairs of motor vehicles and personal and household goods; retail sale of automotive fuel
Rest		Other Financial Intermediation: includes auxiliary activities but not insurance and pension funding (see next)

### 3.12.5 Differentiated productivity growth

#### Background and literature review

Productivity does not vary uniformly among industries and sectors. Harberger 1998<sup>45</sup> points out that the whole dynamics of economic progress actually resembles the growth process of “mushrooms”, rather than the steady rise of “yeast”. Indeed, differential productivity growth is one key factor of structural change in the economic systems, and probably the most important one (Swiecki 2017<sup>46</sup>). Several implications of different growth rates have been investigated in the literature, e.g.: relevance and empirics of the so-called “Baumol's disease” (Baumol 1986<sup>47</sup>, Triplett and Bosworth 2003<sup>48</sup>,

<sup>45</sup> Harberger, A. C., (1998). A Vision of the Growth Process. *The American Economic Review* 88 (1), 1-32

<sup>46</sup> Swiecki, T. (2017) Determinants of structural change. *Review of Economic Dynamics* 24, 95-131

<sup>47</sup> Baumol, W. J. (1986). Productivity Growth, Convergence and Welfare: What the Long-Run Data Show. *The American Economic Review* 76 (5), 1072-1085

Young 2014<sup>49</sup>); specialization and international trade (McMillan and Rodrik 2011<sup>50</sup>, Caron and Markusen 2014<sup>51</sup>); “premature deindustrialization” (Rodrik 2016<sup>52</sup>).

To introduce differentiated productivity growth in the G-RDEM model, we build on Roson 2018, who estimated trends in labour productivity, using the Groeningen GGDC 10-Sector Database (de Vries et al. 2015<sup>53</sup>). In that study, some trends and country specific dummies for labor productivity (VA/employment) are estimated. Results are subsequently employed in a cluster analysis, where three groups of countries with similar characteristics are identified. Table 9 below shows some of the findings used to obtain parameters for G-RDEM:

**Table 9: Average labour productivity growth rates**

Cluster	AGR	MAN	SER	TOT
Rising	6.23	11.43	5.65	8
Steady	7	7.88	5	5.93
Lagging	5.17	5.32	2.34	3.16

Source: Roson 2018

The last column in Table 1 (TOT) displays the average (yearly) growth rate in labor productivity in each group. It refers to value added per worker or hour, so it accounts for capital deepening and similar effects. Interestingly, the differences among industries depends on how fast an economy is growing.

### Estimation of differentiated productivity growth

In the development of the G-RDEM model we are not concerned about labour productivity in itself, but rather on the relative differences among the three broad sectors of Agriculture, Manufacturing and Services. To this end, a correspondence between the three clusters and the annual GDP growth rates used in the SSPs was established. The distribution of IIASA SSP data (OECD) on GDP was considered, and it was assumed that the average GDP growth in the Lagging group of countries corresponds to the 20% percentile of the SSP distribution, 50% for Steady, 80% for Rising. This means 1.2%, 2.5%, and 4.9%, respectively.

Second, the ratio of each sector productivity rate, relative the slowest growing sector, which is Services, was computed. Third, for each industry a quadratic interpolation between the three multipliers and the references GDP growth rates was undertaken, thereby getting three parameters of a quadratic polynomial relationship between a sectoral productivity shifter (ratio between industry

<sup>48</sup> Triplett, J. E., Bosworth, B. P., (2003). Productivity Measurement Issues in Services Industries: Baumol's Disease Has Been Cured. *Economic Policy* (September), 23-33

<sup>49</sup> Young, A., (2014). Structural transformation, the mismeasurement of productivity growth, and the cost disease of services. *American Economic Review* 104 (11), 3635-3667

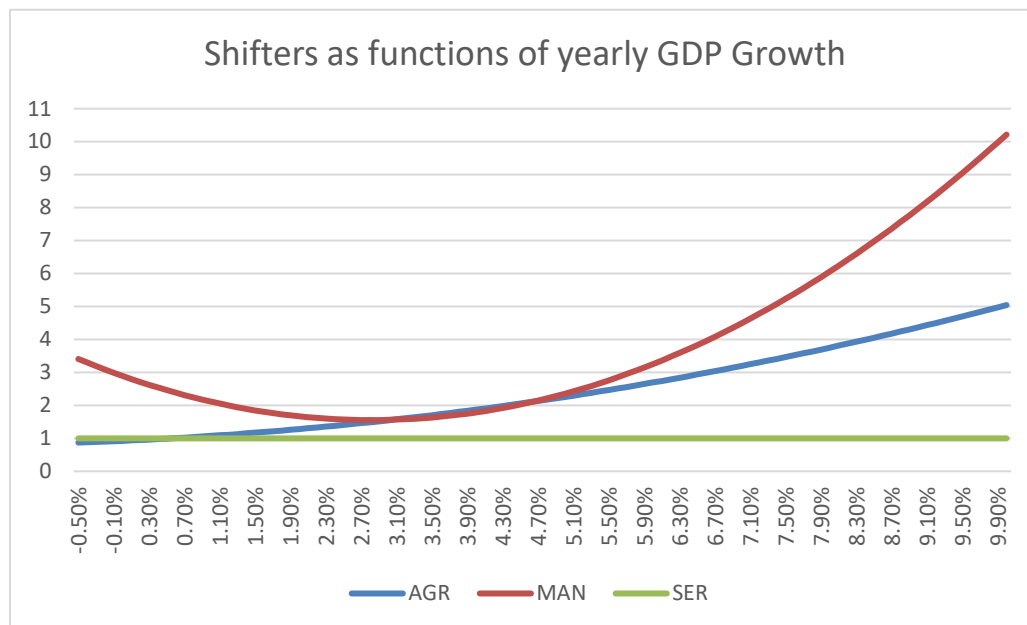
<sup>50</sup> McMillan, M., Rodrik, D., (2011). Globalization, Structural Change and Productivity Growth. NBER Working Paper w17143

<sup>51</sup> Caron, J., Markusen, J. R., (2014). International Trade Puzzles: a Solution Linking Production and Preferences. *The Quarterly Journal of Economics*, 1501-1552

<sup>52</sup> Rodrik, D., (2016). Premature deindustrialization. *Journal of Economic Growth* 21, 1-33

<sup>53</sup> de Vries, G., Timmer, M., de Vries, K., (2015). Patterns of structural change in develop-ing countries. In: Weiss, J., Tribe, M. (Eds.), *Routledge Handbook of Industry and Development*. Routledge, pp. 65-83

growth rate and the corresponding one in the Services) and GDP annual growth. This gives raise to the functions displayed in Figure 8.



**Figure 8 : Productivity growth relative to GDP growth**

The key finding is that productivity differentials are minimized (although still significant) at a moderate GDP growth of around 2%. For higher or lower rates, we can see that differences amplify, with manufacturing becoming the key sector. Notice that the shifter is a multiplier: if aggregate growth is negative, it will likely become negative for the reference slow sector as well. When the shifter for Manufacturing is high and positive, this means that productivity is decreasing there more than in the rest of the economy. In other words, productivity growth in Manufacturing appears as strongly correlated with the aggregate productivity growth, which suggests the existence of inter-sectoral externalities.

Implementation in G-RDEM is straightforward. Total factor productivity in the Services  $tfp(r)$  becomes endogenous during the construction of the baseline and is kept then fixed during counterfactual simulations. Total factor productivity for other sectors (indexed by  $i$ ) in region  $r$  at time  $t+1$  are defined as  $tfp(r)*sh(i,r)$ , where the latter is determined by equations like:

$$Sh(i,r) = a + b \frac{gdp(r, t+1) - gdp(r, t)}{gdp(r, t)} + c \left( \frac{gdp(r, t+1) - gdp(r, t)}{gdp(r, t)} \right)^2 \quad (4)$$

Here are the estimated values for the three parameters  $a$ ,  $b$  and  $c$ :

Table 10: Estimated parameter for sector specific productivity growth

	AGR	MAN
a	0.925391	2.893917
b	11.99205	-94.8599
c	291.8147	1680.554

### Implementation of the sector specific productivity shifters

The production shifter equation derives from the economy wide, regional specific tfp shifter `axpNat` sector specific shifters `lambdaVa` based on the quadratic function as detailed above (see `model.gms`):

```
*
prodShifteq(rs(r),aIn(a),ts(tRun)) $ (xpFlag(r,a) $ (lambdaVa.range(r,a,tRun) ne 0) $ sum(r_r(r,rNat),rgdmpmp.1(rNat,tRun-1))
lambdaVa(r,a,tRun) =E= lambdaVa(r,a,tRun-1) + sum(sameas(subr,r),axpReg(rs,tRun)) + sum(r_r(r,rNat), axpNat(rNat,tRun)
* {
  p_prodShft("a",a)
+ p_prodShft("b",a) * [ (rgdmpmp(rNat,tRun)-rgdmpmp(rNat,tRun-1))/rgdmpmp(rNat,tRun-1)]
+ p_prodShft("c",a) * sqr[ (rgdmpmp(rNat,tRun)-rgdmpmp(rNat,tRun-1))/rgdmpmp(rNat,tRun-1)] };
```

The parameters are derived from estimates in “GTAPRDEM\diffProdGrowth.gms”:

```
set tfpPars / a,b,c/
Table p_diffProdShifters(tfpPars,globSec)
a      AGR      MAN      SER
a      0.925391 2.893917      1
b      11.99205 -94.8599
c      291.8147 1680.554
;
p_prodShft(tfpPars,a) = sum( (globSec_iOri(globSec,iOri),mapa(iOri,a)),p_diffProdShifters(tfpPars,globSec))
/sum( (globSec_iOri(globSec,iOri),mapa(iOri,a)),1);
```

Which requires a mapping from the original GTAP sectors `iOri` to the currently active sector aggregation defined in the same program, of which here only an extracted part is shown:

```
set globSec / AGR,MAN,SER /;
set globSec_iOri(globSec,iOri) /
AGR.    pdr    "Paddy rice"
MAN.    coa    "Coal"
SER.    trd    "Trade"
/;
```

----- 13 line(s) not displayed  
----- 31 line(s) not displayed  
----- 10 line(s) not displayed

Note that using the factor productivity shift also for land can lead to highly implausible changes in crop yields. We therefore introduce an additional equations which downward corrects the land productivity:

```
lambdafLnDeq(rs(r),lnd,aIn(a),ts(tRun)) $ (xfFlag(r,lnd,a) $ (lambdaf.range(r,lnd,a,tRun) ne 0) $ lambdaf.1(r,lnd,a,tRun) $ (not disr(r)) .
lambdaf(r,lnd,a,tRun) =E= 1 / (1 + (lambdaf(r,a,tRun)-0.9999999)*0.5) + eps;
```

Assume a shifter of 1.5 for the value added, in that case, the function will yield  $1 / (1 + (1.5 - 1) * 0.5) = 1 / (1 + 0.25) = 0.8$  as the shifter for land productivity.

### 3.12.6 Endogenous saving rates

#### Background and literature review

We aim at developing a simple but robust mechanism to render aggregate saving rates in G-RDEM endogenous. One strand of literature, relying on cross-country differences of saving rates (e.g. Kirsanova and Sefton 2007<sup>54</sup>), works with micro-economic survey data. It explicitly accounts for factors such as demography, welfare state, retirement behaviour, borrowing constraints, income distribution over a lifetime and its uncertainty, as well as capital gains. The focus here is on the life-cycle hypothesis, which considers the change in available income over a lifetime. While these papers give robust evidence that the factors indeed explain the saving behaviour of individuals or households, they typically offer results only for one or a smaller group of countries.

Rather, we draw here on studies which employ cross-sectional analyses over countries to evaluate the factors affecting the economy-wide aggregate saving rates. Most of these works also take the lifecycle hypotheses into account (although indirectly) and find that even in cross-country analyses larger proportions of the young and the elderly compared to persons in working age (dependency ratios) generally decrease the saving rate (Doshi 1994<sup>55</sup>, Masson et al. 1998<sup>56</sup>, Laoayza et al. 2000<sup>57</sup>).

#### Estimation

Instead of directly using parameter values from the literature, we carry out our own cross-section estimation, using GTAP 9 and other data used in our modelling framework, to overcome any potential divergence in definitions, measurement units etc.. The reader might note that we face a potential endogeneity issue: higher rates of GDP growth require increased capital accumulation, thus larger net investments and consequently higher saving rates. The saving rate and GDP growth are hence structurally dependent. However, this is not an issue of major concern in this context, since we are not integrating the estimated equation into the model, but only updating saving rates, given GDP projections. Hence, our aim is solely to ensure that correlation, not causation, is properly accounted for. Notice also that we obtained our estimates from a sectional data base, which would make it impossible the introduction of lagged variables as instruments.

The distribution of the national aggregate saving shares in the GTAP 9 data set reveals a large spread, as shown in the Figure 9. We regressed those saving rates with OLS against the following explanatory variables:

1. Population composition by age group from the IIASA repository for 2010 (Lutz et al. 2017<sup>58</sup>)
2. GDP growth per capita from 2010 to 2011, in PPPs, from the OECD Env. Growth Model data base as found in the IIASA repository

---

<sup>54</sup> Kirsanova, T., & Sefton, J. (2007). A comparison of national saving rates in the UK , US and Italy. *European Economic Review*, 51(8), 1998-2028

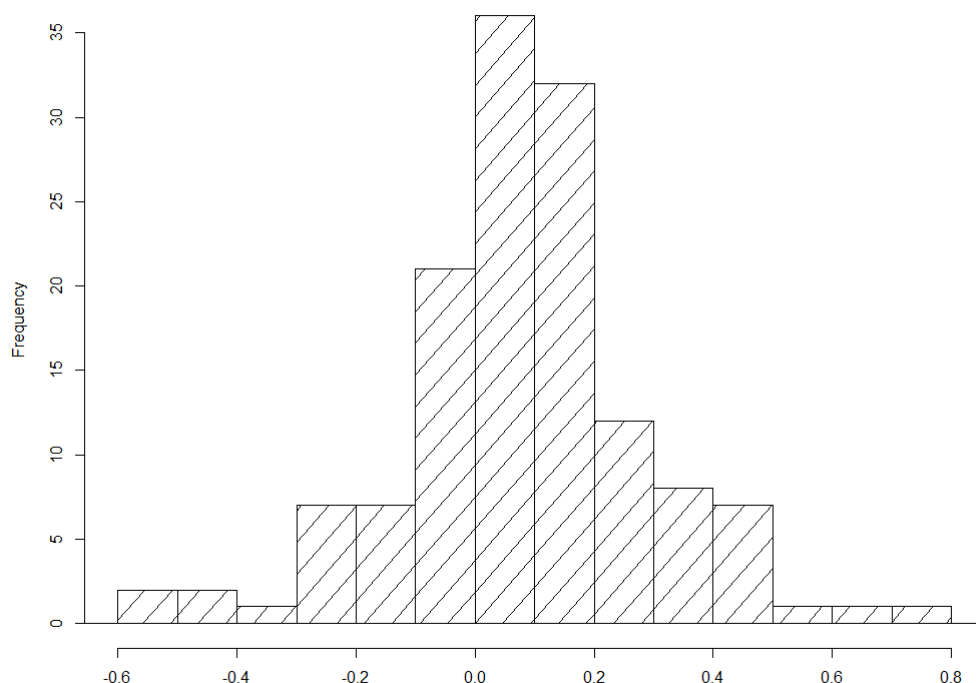
<sup>55</sup> Doshi, K. (1994). Determinants of the saving rate: an international comparison. *Contemporary Economic Policy*, 12(1), 37-45

<sup>56</sup> Masson, P. R., Bayoumi, T., & Samiei, H. (1998). International evidence on the determinants of private saving. *The World Bank Economic Review*, 12(3), 483-501

<sup>57</sup> Loayza, N., Schmidt-Hebbel, K., & Servén, L. (2000). What drives private saving across the world?. *Review of Economics and Statistics*, 82(2), 165-181

<sup>58</sup> Lutz, W., Butz, W. P., & Samir, K. E. (Eds.). (2017). *World Population & Human Capital in the Twenty-First Century: An Overview*. Oxford University Press

3. Foreign savings (trade balance) relative to regional income, from the GTAP 9 data base
4. We also tested, as a potential explanatory variable, the share of government consumption on regional income, but did not find a statistically significant relation.



**Figure 9 : Distribution of aggregated savings rates in GTAP 9**

We found a very good fit for our sectional analysis, with a  $R^2$  at 92% and all variables (with the exemption of the young dependency rate) statistically significant at  $<0.1\%$ . The young DR is nonetheless significant at the 5% level. All variables have the expected sign: dependency ratios decrease the saving rates, as postulated by the life cycle hypothesis, while a higher income per capita and a higher growth rate increase the saving rate. A positive trade surplus (i.e. negative foreign savings) also tends to increase the saving rates.



**Table 11: Regression output for saving rate estimation**

Coefficient	Estimate	Std. error	t-value	Pr(>  t )
Intercept	0.258	0.058	4.401	2.22e-05 ***
Yearly GDP growth %	2.208	0.390	5.659	9.18e-08 ***
GDP per capita	0.001	0.0003	2.899	0.00439 **
Dependency ratio elderly, square root	-0.364	0.079	-4.590	1.02e-05 ***
Dependency ratio young persons, square root	-0.145	0.056	-2.583	0.01089 *
Trade balance, relative to regional income	-0.967	0.047	-20.433	< 2e-16 ***
Trade balance, relative to regional income, squared	0.334	0.067	4.944	2.30e-06 ***

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05

Residual standard error: 0.05893 on 131 degrees of freedom

Multiple R-squared: 0.9213, Adjusted R-squared: 0.9177

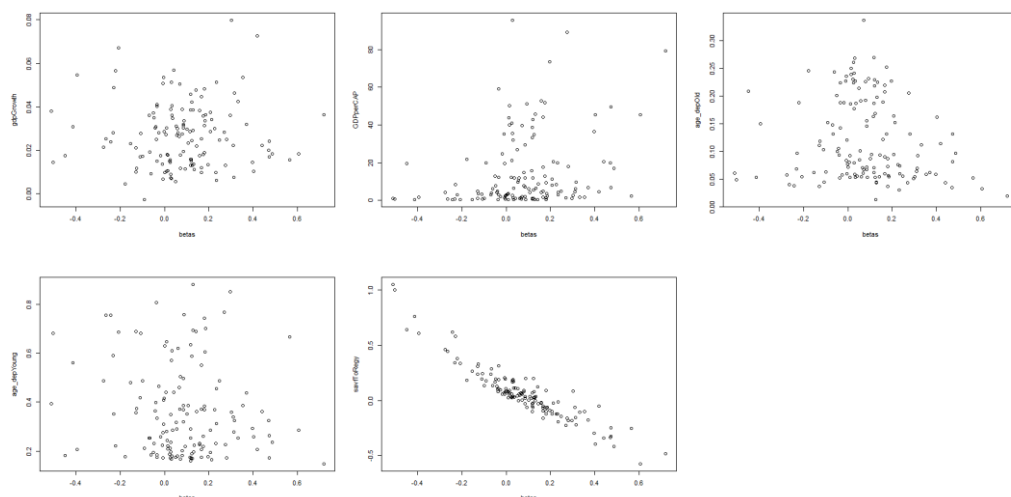
F-statistic: 255.5 on 6 and 131 DF, p-value: < 2.2e-16

The good fit of the regression stems to a large degree from the inclusion of foreign savings relative to regional income, i.e. a trade surplus indicator, (see Table 4 below), while the contributions of the dependency ratios and GDPperCap are in a similar range, with GDP growth trailing.

**Table 12: Analysis of Variance for saving rate estimation**

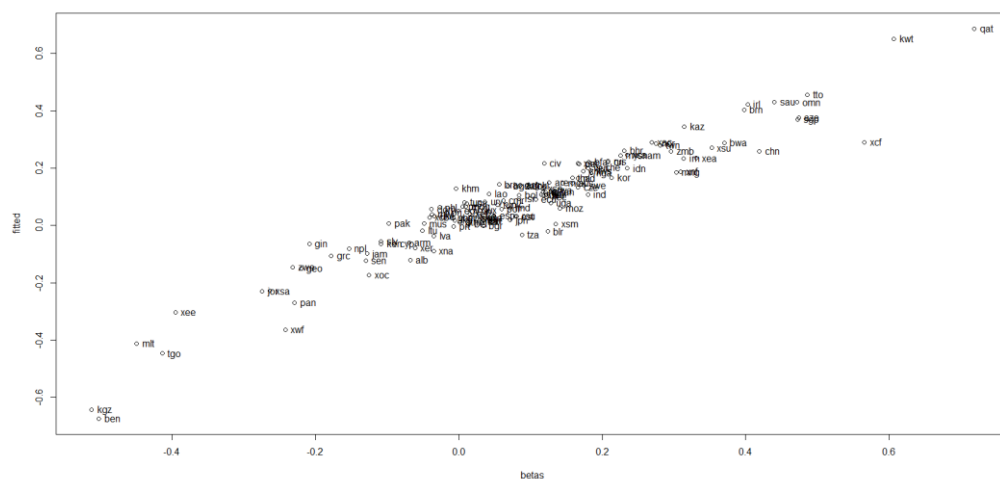
Coefficient	DF	Sum Sq	F value	Pr(>  t )
Yearly GDP growth %	1	0.0001	0.0287	0.8658
GDP per capita	1	0.5271	151	< 2.2e-16 ***
Dependency ratio elderly, square root	1	0.5092	146	< 2.2e-16 ***
Dependency ratio young persons, square root	1	0.3712	106	< 2.2e-16 ****
Trade balance, relative to regional income	1	3.8306	1103	< 2.2e-16 *****
Trade balance, relative to regional income, squared	1	0.0849	24	2.297e-06 ***
Error term	131	0.4549		

Some scatter plots (visualizing the ANOVA results above) between the explanatory variables and the saving rate are shown below in Figure 10.



**Figure 10 : Scatter plots of explanatory variables against the saving rate**

The high contribution of the relation between foreign savings and regional income is not astonishing, because it controls for cases such as oil exporting countries (high saving rates) as well as some other countries, often developing ones, with very low saving rates. The foreign saving indicators can be hence rather understood as a control variable for country specific unobserved features (large receiver of development aid in a group a country with otherwise similar macro-economic indicators, rich oil and gas reserves, tax havens ...). Accordingly, we do not use changes in the foreign savings during the process of baseline construction, to update the saving rates.



**Figure 11 : Fitted against observed saving rates, GTAP region codes as labels**

Note that the fitted values cannot be used as such, since we would then neglect any unexplained additional factors, which could imply large changes in the saving rates from the benchmark to subsequent simulation periods in some countries. Thus, we use relative changes in the estimates – neglecting foreign saving – to update the saving rates used in the model.

### Implementation of the endogenous saving rates

The GAMS code does not use the estimated intercept, but rather updates over the simulation horizon the “betas” for the individual countries in terms of the difference between two consecutive time

periods. Note that we leave the impact of the saving rate out – the estimated coefficient is multiplied in the formula below. The names of the RHS terms should be hopefully self-explaining.

```
p_betaTrack(rNat,tSim,"full") $ (tSim.pos gt 1)
=
( 0.2584670
+ sqrt(p_driver(rNat,"age0_14",tSim)/p_driver(rNat,"age15_64",tSim)) * (-0.1454039)
+ sqrt(p_driver(rNat,"age_gt65",tSim)/p_driver(rNat,"age15_64",tSim)) * (-0.3640043 )
+ p_perCapIncome(rNat,tSim) * (1+p_growthRate(rNat,"GDPPerCAP",tSim)/(100*p_timeStep(tSim))) * 0.0010213
+ (p_growthRate(rNat,"GDPPerCAP",tSim)/(100*p_timeStep(tSim))) * 2.2080586
+ (savf.l(rnat,tSim-1)/regy.l(rnat,tSim-1)) * (-0.9674676)*0
+ sign(savf.l(rnat,tSim-1)/regy.l(rnat,tSim-1)) * sqrt(savf.l(rnat,tSim-1)/regy.l(rnat,tSim-1)) * 0.3347001*0
);
```

Note that using dependency ratios restrict the application to simulation exercises where a projection of the population by age group is available. Given that the IIASA repository features these projections for SSP1-SSP5, we suggest to use one of these SSPs population scenarios directly, or at least the projected age composition.

The parameter `p_betaTrack` gives the estimated saving rates at a zero trade balance which is by definition is average in any simulation. It is calculated for  $t-1$  and  $t-2$ , and its relative change measured against the current saving rate or the simulated is used to update the saving rates in the model:

```
betas.l(rNat,tSim) $ (tSim.pos gt 2)
= betas.l(rNat,tSim) * (1 + sign(betas.l(rNat,tSim))
* (p_betaTrack(rNat,tSim,"full") - p_betaTrack(rNat,tSim-1,"full"))
/max(abs(betas.l(rNat,tSim)),abs(p_betaTrack(rNat,tSim-1,"full"))));
```

In order to maintain consistency, in the top-level expenditure distribution system of the regional household, the expenditure shares of savings, private and government consumption are scaled to unity:

```
p_betaTrack(rNat,tSim,"sum") = ( betas.l(rNat,tSim) + betag.l(rNat,tSim) + betaphi.l(rNat,tSim));

betas.fx(rNat,tSim) = betas.l(rNat,tSim) / p_betaTrack(rNat,tSim,"sum");
betag.fx(rNat,tSim) = betag.l(rNat,tSim) / p_betaTrack(rNat,tSim,"sum");
betap.fx(rNat,tSim) = betap.l(rNat,tSim) / p_betaTrack(rNat,tSim,"sum");
betaphi.l(rNat,tSim) = betaphi.l(rNat,tSim) / p_betaTrack(rNat,tSim,"sum");
```

### 3.12.7 Debt accumulation from foreign savings

Accounting identities in the model ensure (for each time period) that the sum of regional and foreign savings in each region equals gross investments, while foreign savings are equal to the foreign trade deficit. The latter is determined, in the GTAP model (Hertel and Tsigas 1997<sup>59</sup>, Corong et al. 2017<sup>60</sup>), which defines the intra-periodal equilibrium in G-RDEM) by the mechanism of regional allocation of investments. It turn, this is based on a distribution of global savings, driven by relative expected returns on capital, as it is briefly illustrated in the following.

Let denote the price of a homogeneous capital factor (services) as  $p_c$  and  $p_i$  as the price of investments (the cost of producing one unit of new capital good),  $\kappa$  the tax rate on capital earnings,  $fdepr$  the depreciation rate. The net rate of return in a region  $r$  ( $rorc$ ) is defined in the GTAP model as:

<sup>59</sup> Hertel, T.W. & Tsigas, M.E. (1997) "Structure of GTAP", in: T.W. Hertel (ed.), *Global Trade Analysis: Modeling and Applications*, Cambridge University Press

<sup>60</sup> Corong, E., Hertel, T., McDougall, R., Tsigas, M., & van der Mensbrugghe, D. (2017). The Standard GTAP Model, Version 7. *Journal of Global Economic Analysis*, 2(1), 1-119. doi:http://dx.doi.org/10.21642/JGEA.020101AF

$$rorc_r = \frac{p_{c,r} [1 - \kappa_r]}{p_{i,r}} - \overline{fdepr} \quad (5)$$

The expected rate of return  $rorc$  takes into account the difference between start and end of period capital stocks,  $k_s$  and  $k_e$ . The logic is that investors should become more cautious when aggregate investments lead to large changes in capital stocks:

$$rorc_r = rorc_r \left( \frac{k_{s,r}}{k_{e,r}} \right)^{\overline{rorFlex}} \quad (6)$$

The parameter  $rorFlex$  (whose default value is 10 for all for regions) dampens the relative differences in expected returns, thereby avoiding the generation of unrealistically large flows of (real) capital in international markets. In addition, a regional risk factor is introduced, to ensure that an arbitrage condition for the international investor holds in the calibration data set, meaning that a single global, risk-adjusted return  $rorg$  is identified:

$$rorc_r \overline{risk_r} = rorg \quad (7)$$

The condition (7) holds in all periods in G-RDEM, where  $rorg$  and  $rorc$  are endogenous variables. Therefore, the relationships above drive the distribution of foreign savings  $fsav$  or, equivalently, the amount of investments in each region (which do not generally match with regional savings).

The global investor hence expects equal returns of  $rorg$  on his savings in any region. Accordingly, the returns in year  $t$  from foreign savings add up to zero as, by construction, the global economy is closed, and total investments equal total savings (equivalently, the global trade balance is zero):

$$\sum_r fsav_r rorg \equiv 0 \quad (8)$$

The model considers four elements to define the BOP: (1) remittances, (2) development aid, (3) foreign savings and (4) payments to or from the global bank. The first two elements are only found if the myGTAP module is used. The debt accumulation mechanism determines the payments to or from the global bank and is based on the following core elements. Firstly, foreign savings are loans paid back in equal instalments over ten years, with a fixed interest rate equal to the expected rate of return in the year  $t$  and country  $r$  where the loan is granted. This expected rate  $rorc$  reflects returns to capital after depreciation and direct taxation and a country specific risk parameter  $risk$ . Note that a lower risk parameter implies a higher risk assumed by the global bank, according to the relation between  $rorc$  and the global average expected return  $rorg$ :

$$rorc_{r,t} risk_{r,t} = rorg_{r,t} \quad (64)$$

The total amount of outstanding debt  $debt$  for a country in a year is the sum of past loans granted by the global bank, i.e. in years with a positive foreign saving, minus repayments on past loans. We consider further that countries might default on part of their debt, expressed by the parameter  $accum$  which is the share of the debt honoured:

$$\text{debt}_{r,t} = \sum_{tt < t} \max(0, \text{savf}_{r,tt}) \text{accum}_{r,t} \max(0 - 1/10 [t - tt]) \quad (65)$$

The payments related to old debt comprise the repayments, i.e. 10% on the stock, and the interest rate  $\text{rorc}$ :

$$\text{paym}_{r,t} = \sum_{tt < t} \max(0, \text{savf}_{r,tt}) \text{accum}_{r,t} \max(0 - 1/10 [t - tt]) (1/10 + \text{rorc}_{r,tt}) \quad (66)$$

The payments which the global bank receives in the current year  $t$  from each country  $r$  and for each past year  $tt$  are:

$$\text{returns}_{tt,t} = \sum_{r, tt < t} \max(0, \text{savf}_{r,tt}) \text{accum}_{r,t} \max(0 - 1/10 [t - tt]) (1/10 + \text{rorc}_{r,tt}) \quad (67)$$

The returns for a country  $r$  in each year  $t$  are a share on the returns on the total global loans granted in previous years  $tt$ . Note that returns from past years are only occurring when the country acted as a lender, i.e. when its foreign savings were negative:

$$\text{returns}_{r,t} = \sum_{tt < t} \frac{\max(0, -\text{savf}_{r,tt})}{\sum_{rr} \max(0, -\text{savf}_{rr,tt})} \text{returns}_{tt,t} \quad (68)$$

The regional income equation of the country in each year reflects the different between the returns and the payments, i.e.

$$\text{capTrans}_{r,t} = \text{paym}_{r,t} - \text{returns}_{r,t} \quad (69)$$

That means that payments to the bank decrease the share of the GDP which can spent on final consumption and otherwise.

The share of the debt accum which is assumed to be honoured, i.e. not defaulted, is assumed to depend on the debt stock relative to GDP. Note that we take only foreign debt into account, i.e. the net position with the global bank. Specifically, we assume, that defaulting on part of the debt starts if the foreign debt to gdp level exceeds 30%. For each percentage point above that thresholds, one percentage of defaulted past loans are assumed. That mechanism avoids that country can accumulate very large debt stocks:

$$\text{accum}_{r,t} = \min \left( \max \left( 1 - \left\lfloor \frac{\text{debt}_{r,t}}{\text{gdp}_{r,t}} - 1 \right\rfloor - 0.3 \right), 0 \right) \quad (70)$$

That correction alone would however assume that the global bank does not respond to the fact that countries are not fully paying back past loans and serving interest. We therefore let the global bank adjust its country specific risk parameter depending on the current year's return received from payments to the bank relative to GDP. The global bank reacts before critical debt levels occur and part of the loans are not honoured, i.e. the risk parameter is already updated earlier. Similar, receiving payments from the bank decreases the risk. As the global bank mechanism is quite sensitive to very low or high risk parameters, we need some security bounds:

$$\text{risk}_{r,t} = \min \left( 0.2, \text{risk}_{r,t0}, \min \left( 1.25, \max \left( 0.4, \left( \left[ 1 - \frac{\text{debt}_{r,t}}{\text{gdp}_{r,t}} \right]^8 \right) \right) \right) \right) \quad (71)$$

Note that lenders to the bank also receive an update of their risk parameter, i.e. they become more attractive for foreign investments. It is obvious that the thresholds and the elasticity of eight are not the outcome of econometric work but of a trial-and-error approach which various aggregations.

A practical issue emerged when the mechanism above was applied to some special cases, where foreign savings account for a large share of investments or total final consumption. Examples are some developing countries, receiving large amounts of development aid or remittances, but also “tax havens” such as Malta. In such cases, we noticed that the mechanism above can lead, after some periods, to a situation where regional income gets unrealistically small. To avoid such extreme cases, while allowing for the existence of capital inflows or outflows determined by factors other than expected returns, we introduced a regional share parameter, such that only part of the debt may actually be served (see the Technical Annex for more details).

On the position “debt” the cumulated debt is defined: previous year debt level (which can be positive or negative) plus foreign saving in the current year (again positive or negative). The debt servings in the current year are defined in the “paym” column: the foreign savings on all previous years times expected returns – assuming hence a return fixed when the investment is made – times the size of the time step. If the model runs in 5 year steps, the payment in the

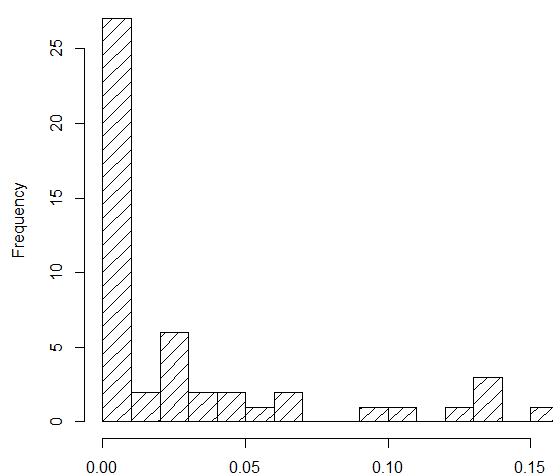
Note the scaling of p\_capTrans to ensure the global expenditures to serve the debt and related revenues add up to zero.

### **3.12.8 Cost-share adjustment**

The question if and to what degree the cost share of total intermediate demand factors changes in the course of economic development is an empirical question, whose answer depends on many factors, such as the industrial composition, the price structure, etc.. Analogously, cost shares might be subject to change for individual sectors. We therefore use again the GTAP 9/10 Data base for a statistical analysis, taking from there data on cost shares for the intermediate composite in the 57/65 GTAP industries and regional GDP per capita.

We exclude values which were 2.5 times the inter-quartile range away from the median. The estimations were performed on the log of the intermediate costs share relative to the log of income per capita and the square of log income per capita to capture potential saturation effects or turning points. We used a model selection procedure to remove insignificant coefficients. Population acted as regression weights to avoid that smaller countries with similar income levels, such as e.g. in case of the EU27 or partly for Africa, carry too much weight.

We find that the link between GDP per capita and the intermediate cost share can be generally confirmed (see Figure 12). However, for nine industries, the model selection process dropped the relation altogether while the remaining coefficients are mostly highly significant (see Table 13).



**Figure 12 : Significance level of regression coefficients of per capita income for cost share of intermediate composite**

There is however no linear trend connecting economic growth to the share of intermediate costs. Our regression reveals that up to around 5.000 USD per capita, the share of intermediate costs in the economy normally increases (first line, “tot”, in Table 6). A possible explanation could be the growing share of manufacturing in the economy, where the share of intermediates is higher. At higher income levels, the share drops, possibly reflecting the generally lower cost shares of intermediates in service sectors.



**Table 13: Intermediate costs shares in total and for the 57 GTAP sectors, mean, max and min estimates at increasing per capita GDP level in cst USD**

	mean	max	min	500	1000	5000	10000	50000	100000	
tot	0,515	0,669	0,291	0,495	0,524	<b>0,568</b>	0,558	0,475	0,421	
pdr	0,460	0,992	0,043	0,264	0,269	0,296	0,328	0,478	<b>0,596</b>	++
wht	0,571	1,000	0,001	<b>0,637</b>	0,592	0,499	0,463	0,390	0,363	--
gro	0,425	0,957	0,001	0,275	0,303	0,377	0,414	0,515	<b>0,566</b>	++
v.f	0,361	0,847	0,002	0,177	0,200	0,263	0,296	0,391	<b>0,440</b>	++
osd	0,451	0,994	0,011	0,249	0,255	0,288	0,328	0,529	<b>0,701</b>	++
c.b	0,442	0,989	0,029	0,300	0,303	0,317	0,333	0,398	<b>0,441</b>	++
pfb	0,535	0,998	0,044	0,286	0,341	0,472	0,497	0,458	<b>0,405</b>	**
ocr	0,325	0,869	0,008	0,227	0,230	0,245	0,263	0,337	<b>0,391</b>	**
ctl	0,520	0,989	0,046	0,378	0,386	0,430	0,483	0,737	<b>0,946</b>	++
oap	0,546	0,984	0,066	0,362	0,402	0,513	0,570	0,727	<b>0,807</b>	++
rmk	0,546	0,999	0,106	0,276	0,315	0,427	0,488	0,662	<b>0,756</b>	++
wol	0,765	0,996	0,173	0,816	0,719	0,589	0,598	0,788	<b>0,982</b>	++
frs	0,399	0,986	0,002	0,142	0,171	0,259	0,311	0,472	<b>0,566</b>	++
fsh	0,481	0,952	0,020	0,248	0,277	0,359	0,402	0,522	<b>0,584</b>	++
coa	0,561	1,000	0,037	0,480	0,419	0,348	0,372	0,608	<b>0,869</b>	+
oil	0,433	1,179	0,027	<b>0,514</b>	0,399	0,255	0,246	0,325	0,429	-
gas	0,471	1,265	0,018	0,320	0,320	0,320	0,320	0,320	0,320	
omn	0,528	1,000	0,071	0,411	0,427	0,466	0,484	0,528	<b>0,548</b>	++
cmt	0,729	0,995	0,307	0,736	0,736	0,736	0,736	0,736	0,736	
omt	0,726	1,008	0,223	0,763	0,763	0,763	0,763	0,763	0,763	
vol	0,784	0,999	0,358	0,813	0,813	0,813	0,813	0,813	0,813	
mil	0,752	0,947	0,440	0,726	0,747	0,784	0,787	0,761	0,737	+
pcr	0,771	1,000	0,224	0,728	0,754	<b>0,780</b>	0,753	0,615	0,536	
sgr	0,662	1,006	0,141	0,684	0,722	<b>0,760</b>	0,720	0,530	0,430	-
ofd	0,690	0,968	0,417	0,738	0,751	<b>0,756</b>	0,732	0,625	0,563	-
b.t	0,603	0,897	0,146	0,565	0,599	<b>0,656</b>	0,649	0,565	0,506	-
tex	0,664	0,952	0,272	0,693	0,710	<b>0,724</b>	0,704	0,605	0,546	-
wap	0,632	0,974	0,237	<b>0,726</b>	0,710	0,673	0,658	0,624	0,610	--
lea	0,669	1,000	0,275	<b>0,713</b>	0,710	0,693	0,677	0,619	0,587	--
lum	0,680	0,979	0,295	0,625	0,646	<b>0,683</b>	0,681	0,637	0,603	-
ppp	0,650	0,950	0,339	0,692	0,708	<b>0,714</b>	0,685	0,557	0,486	-
p.c	0,860	1,374	0,418	<b>0,939</b>	0,935	0,915	0,894	0,820	0,780	--
crp	0,732	1,000	0,395	<b>0,772</b>	0,769	0,750	0,730	0,663	0,626	-
nmm	0,657	1,000	0,235	<b>0,729</b>	0,705	0,653	0,631	0,584	0,565	--
i.s	0,778	0,998	0,411	<b>0,795</b>	0,792	0,774	0,757	0,694	0,660	
nfm	0,764	0,999	0,457	<b>0,792</b>	0,785	0,769	0,762	0,746	0,740	--
fmp	0,680	0,998	0,294	<b>0,739</b>	0,733	0,703	0,672	0,570	0,518	--
mvh	0,723	1,000	0,270	0,750	0,750	0,750	0,750	0,750	0,750	

	mean	max	min	500	1000	5000	10000	50000	100000	
otn	0,679	1,070	0,236	<b>0,734</b>	0,716	0,677	0,661	0,625	0,610	--
ele	0,694	0,999	0,197	<b>0,857</b>	0,809	0,724	0,709	0,716	0,739	-
ome	0,686	0,999	0,240	<b>0,801</b>	0,764	0,686	0,655	0,588	0,561	--
omf	0,679	0,999	0,167	<b>0,752</b>	0,711	0,635	0,615	0,596	0,599	--
ely	0,672	1,004	0,119	0,638	0,674	<b>0,711</b>	0,672	0,491	0,396	-
gdt	0,529	1,194	0,101	0,343	0,404	0,547	<b>0,572</b>	0,522	0,461	+
wtr	0,463	0,951	0,002	0,422	0,422	0,422	0,422	0,422	0,422	
cns	0,601	0,858	0,243	0,594	0,609	<b>0,621</b>	0,605	0,521	0,471	-
trd	0,429	0,842	0,083	0,296	0,316	0,367	0,391	0,455	<b>0,485</b>	++
otp	0,603	0,986	0,255	0,520	0,533	0,564	0,579	0,613	<b>0,628</b>	++
wtp	0,710	0,988	0,291	0,509	0,532	0,591	0,618	0,686	<b>0,718</b>	++
atp	0,733	1,051	0,324	0,583	0,610	0,678	0,710	0,789	<b>0,826</b>	++
cmn	0,393	0,702	0,113	0,271	0,295	0,361	0,393	0,481	<b>0,524</b>	++
ofi	0,383	0,912	0,000	0,232	0,236	0,260	0,287	0,417	<b>0,519</b>	++
isr	0,461	0,911	0,000	0,253	0,308	0,450	<b>0,487</b>	0,482	0,441	+
obs	0,381	0,740	0,079	0,383	0,383	0,383	0,383	0,383	0,383	
ros	0,447	0,964	0,025	0,405	0,407	0,419	0,432	0,483	<b>0,516</b>	++
osg	0,332	0,659	0,025	0,232	0,245	0,280	0,296	0,338	<b>0,358</b>	++
dwe	0,142	0,483	0,000	0,090	0,090	0,090	0,090	0,090	0,090	

Note: +/- depicts the cases the estimate cost share at the highest income is higher/lowest than at the lowest  
 ++/-- depicts cases where the share is estimated to increase/decrease over the whole income range

On the other hand, our estimates provide rather clear results for agricultural activities: with the exception of wheat (wht), all agro-industries show increasing cost shares of intermediates at higher levels of per capita income. For animal processes, the differences in the shares are more pronounced compared to crops. Similarly, there are strong increases in intermediate shares for forestry (frs), fisheries (fsh) and coal mining (coa), and moderate ones for “other minerals” (omn), whereas the shares are dropping for oil extraction and no significant relation was found for the gas extraction.

Contrary to primary sectors, almost no manufacturing industry displays increasing shares. Food and meat processing (cmt, omt) as well as oil seed crushing (vol) show no clear relation to income. For the remaining food processing sectors, the intermediate cost shares peak at around 5.000 USD. For the rest of manufacturing, the strongest are typically found at the low income levels, although differences in the shares are often not very large.

There is also no uniform tendency for electricity, gas and water distribution activities (ely, gas, wtr) and construction (cns), where intermediate shares peak at medium income levels. Interestingly, results for all service industries highlight a clear direction: intermediate shares increase at higher per capita income levels. Especially high shares are found in the transportation services (otp, wtp, atp), which are around 80%, while the remaining services typically reach values around 50%.

If preferences are a function of income per capita, reflected in non-linear Engel curves, then the portfolio of products offered by the economy clearly changes. As Chenery et al. 1986 put it “On the demand side, a rise in income can only be sustained if the goods and services made available correspond to the proportions in which consumers wish to spend their income”. We already addressed this issue for the final demand through the introduction of an AIDADS demand system, but further

adjustments are in order on the production side, to account for income-dependent variations in intermediate demand. Indeed, an often neglected aspect in CGE and input-output models is that industries internally include many diverse production processes, characterized by different technologies. Variations in demand patterns therefore occur not only between the macro-industries, but also inside them: aggregate industrial cost structures should be better interpreted as reflecting the internal composition of a sector, rather than describing the production function of a representative firm. Consequently, input-output coefficients can well evolve over time, following changes in income, prices, foreign trade, demography, etc., in a way not too different from the one affecting household final consumption. In parallel, processes such as capital accumulation and the related shift to more capital based production technologies can systematically affect the cost shares in certain industries.

Already Arrow and Hoeffenberg, 1959<sup>61</sup> decomposed changes in input-output coefficients into variations due to real disposable income and variations due to technology and tastes. Skolka, 1989 further provided a structural decomposition analysis for Austria along these lines, thereby explicitly considering that I-O coefficients are not static, but actually change along the process of economic development. This contrasts with the approach followed in most dynamic CGE models, where changes in the industrial cost shares are only attributed to two causes: non-Hicksian technological progress and changes in relative prices.

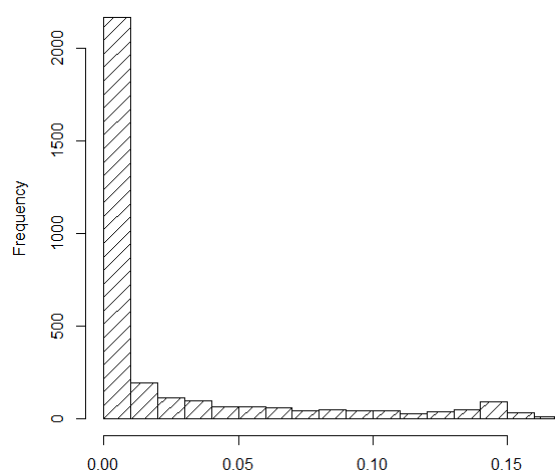
As an example, consider the “Other food” sector in GTAP, comprising a wide range of processing activities of crop and fish based products. Here the ICP data set, which was used to estimate the AIDADS demand system, includes expenditure data of subcategories such as “bread”, “Other bakery products”, “Pasta products”, “Jams, marmalades and honey”, “Confectionery, chocolate and ice cream” etc.. A simple regression on the shares of more disaggregated data relative to totals reveals that they typically are income dependent. As some of these categories require different inputs in production and are based on different technological processes, also the I-O composition of the “Other food” industry should hence be income dependent.

Therefore, I-O coefficients should be not considered as constant in the long-term, where income varies significantly. Since the model already accounts for price-induced compositional changes in intermediate demand, and possibly Hicksian non-neutral technical progress, we include in G-RDEM the modelling of income-related variations.

Our basic hypothesis is that I-O coefficients are income dependent, likewise final consumption shares. Since time series consistent with the GTAP industrial classification are not available, we test our hypothesis using a sectional approach, using once more the GTAP 9 data base, as we did for the intermediate composite. From the 3.249 I-O coefficients (57x57) in the data base, the model selection process filtered a significant relation to GDP per capita in as many as 3.213 of them. The histogram in Figure 9, showing the significance levels, highlights that for more than two thirds of elements in the sample they are significant at 1% level or below.

---

<sup>61</sup> Arrow, K. J. (1959). A time series analysis of interindustry demands (Vol. 17). North-Holland



**Figure 13 : Significance level of regression coefficients of per capita income for individual input coefficients**

Consider the agricultural “coarse grains” industry as an example, and its six largest cost shares (see Table 14). For two of them, corresponding to seeds and transport, there is no clear income dependence. Services tend to grow with higher per capita income, whereas shares of chemical and petroleum products decrease. Coarse grains could hence be seen as an example where economic development, through changes in technology – intensification in crop production by using more fertilizers, machinery etc. – and outsourcing of activities (contract work) drive differences in costs shares.

**Table 14: Main cost shares in coarse grains as a function of per capita income**

	mean	max	min	500	1000	5000	10000	50000	100000
gro.a_crp.c	0,173	0,666	0,000	0,113	0,138	0,199	0,210	0,187	0,161
gro.a_trd.c	0,048	0,611	0,000	0,041	0,041	0,047	0,053	0,083	0,109
gro.a_gro.c	0,043	0,315	0,000	0,043	0,043	0,043	0,043	0,043	0,043
gro.a_p.c.c	0,030	0,400	0,000	0,008	0,014	0,044	0,056	0,051	0,037
gro.a_otp.c	0,027	0,118	0,000	0,027	0,027	0,027	0,027	0,027	0,027
gro.a_obs.c	0,012	0,217	0,000	0,002	0,005	0,018	0,028	0,049	0,052
gro.a_ofi.c	0,008	0,071	0,000	0,001	0,003	0,014	0,018	0,012	0,007
Sum				0,233	0,267	0,378	0,416	0,439	0,428

Integration of the regression results in the model is not a trivial task, as shares found at the base year may deviate considerably from the estimates. One option is to realign estimates and data by adding an error term to the intercept in the regressions, while considering also the effect of changes.

This is necessary because, for example, absolute changes may lead to negative shares when the shares are small. On the other hand, relative changes would keep small shares forever small, much like as it happens for trade shares in the Armington formulation.

We therefore opt here for an approach where the error terms  $\text{err}$  are adjusted on the basis of the following equation, where  $\text{inc}$  stands for GDP per capita in the benchmark 0 and in the current simulation year  $t$ , whereas  $\alpha$  refers to the share parameter for the intermediate composite:

$$and_{corr} = and_{est} + err \left( \max \left( 0, 1 - \frac{1}{2} abs \left[ \frac{inc_t - inc_0}{inc_0} \right] \right) \right) \quad (11)$$

The idea behind the correction is simple: the farther away the given GDP per capita from the benchmark, the less structural information is provided by the initial SAM. Specifically, equation (11) defines a relation where at an income growth of 50%, 75% of the error term will be added. If real income doubles, 50% is added and if income triples, the estimates are directly used.

A further complication is due to the fact that the cost share is not a given parameter but reflects price changes for the output and the intermediate input composite. The price variations for the intermediate composite (pnd) and the unit cost of output (px) need to be taken into account in the model. The cost share parameter is therefore defined as:

$$and_{fin} = and_{corr} \frac{px}{pnd} \quad (12)$$

Table 8 illustrates the approach. The estimated cost share at benchmark mark income is about 27% (row Econometric estimate), while the share found in the SAM is about 33.8% (row AND parameter). That results in an error term of around 6%. In the first iteration period (t04), the estimate increases only slightly, as also GDP per capita is not changing much. Accordingly, around 97% of the error term is kept, and the cost share only drops slightly towards the lower econometric estimate. The cost share parameter is further corrected stronger as the per unit price decreases to 0.93, lower than the intermediate composite price index (0.97). In year 28 of the simulation, GDP has tripled and the correction factor drops to zero. Accordingly, from there onwards, the cost share of the intermediate composite (row ND cost share) is identical to the econometric estimate at each iteration round.

**Table 15: Example of cost share parameter adjustment in simulation**

	t00	t01	t04	t08	t12	t16	t20	t24	t28	t32	t36	t40
Econometric estimate		0,27	0,27	0,27	0,27	0,28	0,28	0,28	0,29	0,30	0,31	0,31
pnd		1,00	0,98	0,81	0,73	0,69	0,66	0,68	0,67	0,70	0,75	0,82
px		1,00	0,93	0,80	0,76	0,77	0,78	0,88	0,93	1,03	1,14	1,29
Error term		0,06	0,06	0,06	0,06	0,06	0,06	0,06	0,06	0,06	0,06	0,06
Correction factor		1,00	0,97	0,89	0,74	0,55	0,31	0,02				
GDP		1,42	1,42	1,49	1,73	2,14	2,69	3,38	4,19	5,19	6,30	7,49
AND parameter	0,33	0,33	0,31	0,32	0,33	0,34	0,35	0,37	0,40	0,44	0,46	0,49
ND cost share		0,33	0,33	0,32	0,31	0,31	0,30	0,29	0,29	0,30	0,31	0,31

The estimated coefficients are stored in the GDX file “gams/gtapRdem/io\_regr\_res.gdx” on the parameter p\_estres. As a first step, we remove any non-availables (NA):

```
p_estRes(iDat,jDat,regCoef) $ (mapVal(p_estRes(iDat,jDat,regCoef)) eq 5) = 0;
p_estRes(iDat,"nd.share",regCoef) $ (mapVal(p_estRes(iDat,"nd.share",regCoef)) eq 5) = 0;
```

Next, we deduct the estimate at benchmark per capita income and store it temporarily on the parameter p\_corr from the given io-coefficient

```
p_corr(r,a,i) $ (io(r,i,a,"%t0%")
  $ sum((mapa(iDat,a),mapi(jDat,i)),p_estRes(iDat,jDat,"log_GDP") or p_estRes(iDat,jDat,"sqr_log_GDP")))
= io(r,i,a,"%t0%")
- exp[sum((mapa(iDat,a),mapi(jDat,i)),p_estRes(iDat,jDat,"int")
  + sum(r_r(r,rNat),p_estRes(iDat,jDat,"log_GDP") * log(p_perCapIncome(rNat,"%t0%"))
  + p_estRes(iDat,jDat,"sqr_log_GDP") * sqrt(log(p_perCapIncome(rNat,"%t0%"))))
  * sign(log(p_perCapIncome(rNat,"%t0%")))))];
```

Similar statements perform the same operation for io-coefficients in technology nests and for the intermediate composite cost share. During simulation, we first calculate the error correction factor  $p\_errCorr$  as a function of per capita income in the current simulation point  $tSim$  and the benchmark  $t0$ , as discussed above:

```
p_errCorr(rNat) = max(0,1 - 0.5*abs((p_perCapIncome(rNat,tSim)-p_perCapIncome(rNat,"%t0%"))
  /p_perCapIncome(rNat,"%t0%")));
```

Next, the calculate the estimate at the current income

```
and(r,a,tSim) $ (and(r,a,tSim-1) $ p_corr(r,a,"nd"))
= exp[sum((mapa(iDat,a),p_estRes(iDat,"nd.share","int")
  + sum(r_r(r,rNat),p_estRes(iDat,"nd.share","log_GDP") * log(p_perCapIncome(rNat,"%t0%"))
  + p_estRes(iDat,"nd.share","sqr_log_GDP") * sqrt(log(p_perCapIncome(rNat,"%t0%"))))
  * sign(log(p_perCapIncome(rNat,"%t0%")))))];
```

Adding subsequently a part of the error term  $p\_corr$  at the benchmark according to the factor  $p\_errCorr$  and introduce security bounds:

```
and(r,a,tSim) $ (and(r,a,tSim-1) $ p_corr(r,a,"nd"))
= min((and(r,a,"%t0%")*2,1-ava(r,a,"%t0%")*0.5,and(r,a,tSim-1)*(1+0.1*p_timeStep(tSim))),
  max((and(r,a,"%t0%")*0.5,and(r,a,tSim-1)*(1-0.1*p_timeStep(tSim))),
  {and(r,a,tSim) + p_corr(r,a,"nd")*p_errCorr(r)}
  *px(r,a,tSim)/px(r,a,"%t0%")*1/pnd(r,a,tSim)
));
```

The value added share is calculated residually:

```
ava(r,a,tSim) = 1 - and(r,a,tSim);
```

Note here:

1. The application of the estimation is not allowed to more than double or half the benchmark coefficient.
2. The coefficients are not allowed to change by more then 0.1 time the number of simulated years. For individual IO-cost shares, the changes are limited to 0.025 per year.
3. The estimated error terms are introduced in the update process as well such that the for the first year  $tSim=1$ , per capita income for the benchmark will enter, such that the last term will yield exactly the observed IO coefficient. The process hence only adds the changes from year to year in the estimated to the benchmark coefficient.

Similar statements estimate and apply shifters for the other elements. The IO-cost shares are scaled to maintain the original sum:

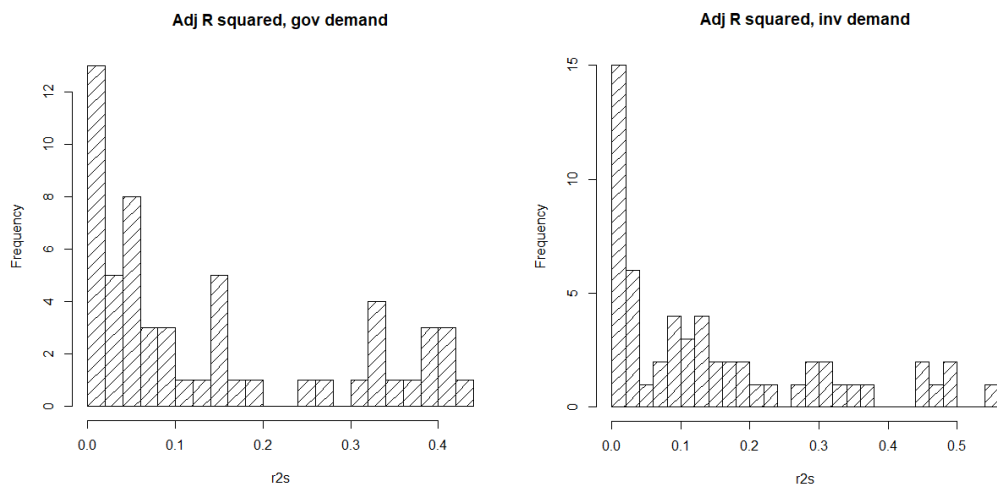
```
p_scale(r,a,"") $ sum(j,io(r,j,a,tSim))
= sum(j,io(r,j,a,"%t0%"))/sum(j,io(r,j,a,tSim));

io(r,i,a,tSim) $ io(r,i,a,tSim-1) = io(r,i,a,tSim) * p_scale(r,a,"");
```

### 3.12.9 Endogenous expenditure shares for government and investment demand

Similar to the discussion around industry cost shares depending on economic development, the composition of investment demand might change in the process of economic growth reflecting both a different sectoral composition of the economy and different capital stock composition in each sector. For instance, the share of investments in patents or similar might increase with higher income levels. In the same vein, the composition of government demand might change.

In both cases, we follow an empirical approach by employing regression analysis to the cross-sectional data provided by the GTAP SAM. We use expenditure shares for individual commodities on total investment respectively government demand as dependent variables, and the logarithm of income and its squares as explanatory ones. In case of government expenditure, we also use its total share on regional income, linear and in squares, as independents. A model selection process removes insignificant regressors from the estimation equation to avoid spurious results. We use total population as weights.



The integration into the simulation framework follows the approach used for industry cost shares. We use a weighted average of the expenditure share at the benchmark and the estimates, defining weights based on the relative difference of the projected per capita income from the benchmark. The closer the projection to the benchmark, the closer the expenditure share is to the one found in the SAM while large deviations will drive the shares towards their estimates.

Technically, in order to prepare for simulation, we need therefore the error terms for the current regional and sectoral aggregation. We therefore first define the estimates, temporarily on the parameter `p_corrInv`, using the per capita income `p_perCapIncome` at the benchmark “%t0%” on the RHS together with the parameters found on `p_estResInv`. The estimations are performed on the 57 sector resolution of the data base as depicted by the set `iDat`. In case of an aggregated data base being used for simulation, the cross set `mapi` depicts which of these 57 sectors are aggregated to the sector list in use `i` such that estimated shares are aggregated. The regression results are only used if income dependencies were found, i.e. at least one of the regression coefficients is non-zero:

```
p_corrInv(rNat,i) $ (alphaa.1(rNat,i,"inv","%t0%")
    $ sum((mapi(iDat,i)),p_estResInv(iDat,"log_GDP") or p_estResInv(iDat,"sqr_log_GDP")))
= sum((mapi(iDat,i)),p_estResInv(iDat,"int")
    + p_estResInv(iDat,"log_GDP")*log(p_perCapIncome(rNat,"%t0%"))
    + p_estResInv(iDat,"sqr_log_GDP")*sqr(log(p_perCapIncome(rNat,"%t0%")))
    * sign(log(p_perCapIncome(rNat,"%t0%")))*p_scaleInvToSplit(rNat,i);
```



From there, we define the error term at the benchmark as the difference to the benchmark share `p_alphaa.l`.

```
p_corrInv(rNat,i) $ (alphaa.l(rNat,i,"inv","%t0%") $ p_corrInv(rNat,i))
= alphaa.l(rNat,i,"inv","%t0%") - p_corrInv(rNat,i);
```

The amount of the error applied in year `tSim` is defined as follows (see industry cost shares for a discussion):

```
p_errCorr(rNat) = max(0,1 - 0.5*abs((p_perCapIncome(rNat,tSim)-p_perCapIncome(rNat,"%t0%"))
/p_perCapIncome(rNat,"%t0%")));
```

During simulation, we first define the estimated shared at the projected per capita income, the expression is identical to the one discussed above with the exemption that the current simulation year `tSim` is used instead of the benchmark point “%t0%”:

```
alphaa.l(rNat,i,"inv",tSim) $ (alphaa.l(rNat,i,"inv","%t0%")
$ sum((mapi(iDat,i)),p_estResInv(iDat,"log_GDP") or p_estResInv(iDat,"sqr_log_GDP")))
= sum((mapi(iDat,i)),p_estResInv(iDat,"int")
+ p_estResInv(iDat,"log_GDP") * log(p_perCapIncome(rNat,tSim))
+ p_estResInv(iDat,"sqr_log_GDP") * sqrt(log(p_perCapIncome(rNat,tSim)))
* sign(log(p_perCapIncome(rNat,tSim)))) * p_scaleInvToSplit(rNat,i);
```

Shares comprised in final demand nests `dNest` are aggregated to yield the share of the nest on expenditures:

```
alphaDN(rNat,dNest,"inv",tSim) = sum(dNest_i_fd(dNest,i,"inv"), alphaa.l(rNat,i,"inv",tSim));
```

Next, we add the share part `p_errCorr` of the error term `p_corrInv` at the benchmark to the estimate, that is the expression in the last line in curly brackets. If `p_errCorr` is equal to one, the full error term is added the the last line is equal to the benchmark share. As `p_errCor` decreases to zero, the expression in curly brackets approaches the regression the results. The surrounding min and max operators define security bounds preventing that expenditure shares more than double or half or increase by more than 5% from year to year:

```
alphaa.l(rNat,i,"inv",tSim) $ p_corrInv(rNat,i)
= min(alphaa.l(rNat,i,"inv","%t0%")*2,alphaa.l(rNat,i,"inv",tSim-1)*(1+0.05*p_timeStep(tSim)),
max(alphaa.l(rNat,i,"inv","%t0%")*0.5,alphaa.l(rNat,i,"inv",tSim-1)*(1-0.05*p_timeStep(tSim)),
{alphaa.l(rNat,i,"inv",tSim) + p_corrInv(rNat,i)*p_errCorr(rNat)}
));
```

Similar, expenditure share of demand nest `dNest` are defined:

```
alphaDn(rNat,dNest,"inv",tSim) $ p_corrInv(rNat,dNest)
= min(alphaDn(rNat,dNest,"inv","%t0%")*2,alphaDn(rNat,dNest,"inv",tSim-1)*(1+0.05*p_timeStep(tSim)),
max(alphaDn(rNat,dNest,"inv","%t0%")*0.5,alphaDn(rNat,dNest,"inv",tSim-1)*(1-0.05*p_timeStep(tSim)),
{alphaDn(rNat,dNest,"inv",tSim) + p_corrInv(rNat,dNest)*p_errCorr(rNat)}
));
```

Finally, we scale the shares to maintain the original sum:

```

--- scale to maintain original sum of investment cost shares

p_scaleInv(rNat) = [ sum(notInvInNest(i), alphaa.l(rNat,i,"inv","%t0%"))
+ sum(dNest, alphaDn(rNat,dNest,"inv","%t0%")) ]
/[ sum(notInvInNest(i), alphaa.l(rNat,i,"inv",tSim))
+ sum(dNest, alphaDn(rNat,dNest,"inv",tSim)) ];

alphaa.fx(rNat,i,"inv",tSim) $ (notInvInNest(i)) = alphaa.l(rNat,i,"inv",tSim) * p_scaleInv(rNat);
alphaDn(rNat,dNest,"inv",tSim) = alphaDn(rNat,dNest,"inv",tSim) * p_scaleInv(rNat);

```

The code for the government share is structurally identical.

The impact on the simulated results seems of lower importance compared to other elements in G-RDEM. This firstly reflects that investment demand is generally a minor component of overall demand. Furthermore, construction (“cns”) dominates with shares typically around 55% investment expenditure shares and is found to be little dependent on income, while many other products have very low shares. For government, a very large share is for government services, the share of government demand for other products on total demand is generally quite low.

### 3.12.10 Revovering crop yield projections

The sector specific productivity growth in G-RDEM would imply the also land productivity would follow the estimated relations, potentially leading to quite unrealistic partial productivities, i.e. crop yields.

During baseline construction, the partial land productivity for GTAP crop activity  $a$  in model region  $r$  and simulation year  $t$  is fixed to the yield development of the FAO projections time the benchmark productivity:

$$\frac{xp_{r,a,t}}{xf_{r,lnd,a,t}} = \frac{faoProj_{r,a,t}^{yld}}{faoProj_{r,a,t0}^{yld}} \frac{xp_{r,a,t}}{xf_{r,lnd,a,t}} \quad (72)$$

Where  $xp$  is output and  $xf$  factor use,  $lnd$  refers to land, and  $faoYldProj$  to the yield projection by the FAO in  $t \text{ ha}^{-1}$ . Fixing the yield is made possible by endogenously adjusting the Hoicks-non neutral land productivity shifter for  $r,a,t$  in each simulated point.

The model equations uses the projections  $p\_faoProjections$ , considering the partial productivity  $p\_agrYield$  derived from the benchmark SAM:

```

*
* --- fix yields
*
e_agrYield(rs,lnd,a,ts(t)) $ (p_agrYield(rs,a) $ (not disr(rs)) $ (lambdafShift.range(rs,lnd,a,t) ne 0) ) ..
*
* --- define output as exogenous yields (from FAO projection) times land used in the activity,
*      lambdaf for land will endogenously adjust
*
m_xp(rs,a,t)/xp.scale(rs,a,t)
=E= sum((r_r(rs,rsNat),i,faoYears) $ ((faoYears.pos eq t.pos) $ xFlag(rs,a,i)),
(p_faoProjection(rsNat,i,"yield",faoYears)/p_faoProjection(rsNat,i,"yield","2012") ) * p_agrYield(rs,a))
* xf(rs,lnd,a,t)/xp.scale(rs,a,t) ;

```

Note that in the equation, the definition of the partial productivity  $xp/xf$  is changed into a linear one by moving  $xf$  on the LHS.

We map the change in land productivity also to water in case that a differentiation between rainfed and irrigated crops is found in the data base:

```
e_lambdaShiftWater(rs,f,a,ts(tRun)) $ (p_agrYield(rs,a) $ xfflag(rs,f,a) $ (lambdaFShift.range(rs,f,a,tRun) ne 0) $ sameas(f,"water")) ..
    lambdafShift(rs,f,a,tRun) =E= sum(lnd, lambdafShift(rs,lnd,a,tRun));
```

The crop yield projections from FAO 2018<sup>62</sup> are mapped to GTAP based on a mapping between the GTAP sectors and the ones in the FAO:

```
set iOri_ac(%iOri%,faoCrops) "Link between GTAP sectors/products and crops covered by FAO projection" /

pdr.A_PARI
wht.A_WHEA
gro.A_MAIZ
gro.A_BARL
gro.A_MILL
gro.A_SORG
gro.A_XCER
v_f.A_POTA
v_f.A_SWPY
v_f.A_CASS
v_f.A_XRNT
v_f.A_PULS
v_f.A_XVEG
v_f.A_PLAN
v_f.A_BANA
v_f.A_CITR
v_f.A_XFRU
c_b.A_SUGB
c_b.A_SUGC
osd.A_RAPS
osd.A_SOYB
osd.A_GRND
osd.A_SUNF
osd.A_SESA
osd.A_CCNT
osd.A_XOLT
osd.A_PALM
osd.A_OLIV
ocr.A_COCO
ocr.A_COFF
ocr.A_TEAS
ocr.A_TOBA
pfb.A_COTT
pfb.A_XFIB
ocr.A_RUBB
ocr.A_XCRO
/;
```

In case, a link to the FABIO data base is used, a second mapping defines the link between FABIO and the FAO crops, not shown in here. Jointly, the define the mapping between the current products in the data base i and the FAO crops:

```
set i_faoCrops(i,faoCrops) "Link between current commodities and FAO crops";
i_faoCrops(i,faoCrops) $ sum(mapi(%iOri%,i) $ iOri_ac(%iOri%,faoCrops),1) = YES;
i_faoCrops(i,faoCrops) $ sum((fabc_i(fabc,i),fabc_ac(fabc,faoCrops1)),1) = no;
i_faoCrops(i,faoCrops) $ sum(fabc_i(fabc,i) $ fabc_ac(fabc,faoCrops),1) = yes;
```

From this, projections at the level of the GTAP products are defined, aggregation of the system fs={rainfed, irrigated} found in the FAO data base:

```
p_faoProjection(rNat,i,"1000ha",faoProj)
= sum( (r_faoReg(rNat,faoReg),i_faoCrops(i,faoCrops),fs), arable_land("%faoScen%",faoReg,faoCrops,fs,faoProj)*0.001);
*
* --- attention: output is harvested-area (not arable_land) times yield.
*
p_faoProjection(rNat,i,"1000t",faoProj)
= sum( (r_faoReg(rNat,faoReg),i_faoCrops(i,faoCrops),fs) $ yield_shift_crop_clim("%faoScen%",faoReg,faoCrops,fs,faoProj),
    area_harvested("%faoScen%",faoReg,faoCrops,fs,faoProj)*(yield_crop("%faoScen%",faoReg,faoCrops,fs,faoProj)
        $if(CCIImpact "%faoCCIImpact%"="off" /yield_shift_crop_clim("%faoScen%",faoReg,faoCrops,fs,faoProj)
    ));
```

<sup>62</sup> The future of food and agriculture – Alternative pathways to 2050. Rome. 224 pp. Licence: CC BY-NC-SA 3.0 IGO

The FAO projections are not available as a full time series, we therefore first interpolate linearly between neighboring years land use and crop output projections where projections are available:

```
*
*      sum(faoProj,p_wgts,faoYears,faoProj, "start" ,,,
*
* --- linearly interpolate between given FAO projection points
*
p_faoProjection(rnat,i,faoItems,faoYears) $ ((not p_faoProjection(rnat,i,faoItems,faoYears))
$ sum(faoProj,p_faoProjection(rnat,i,faoItems,faoProj)))
= sum(faoProj $ p_wgts(faoYears,faoProj,"start"),p_faoProjection(rnat,i,faoItems,faoProj))
*
* --- add difference in projection divided by # of years
*
+ ( sum(faoProj $ p_wgts(faoYears,faoProj,"end"),p_faoProjection(rnat,i,faoItems,faoProj))
- sum(faoProj $ p_wgts(faoYears,faoProj,"start"),p_faoProjection(rnat,i,faoItems,faoProj)))
* p_wgts(faoYears,"","mult");
```

And next smooth the results to avoid kinks:

```
*
* -- rolling 5-year average to smooth over jumps in delta resulting from linear interpolation
*
p_faoProjection(rNat,i,faoItems,faoYears) $ (p_faoProjection(rNat,i,faoItems,faoYears) $ (faoYears.pos gt 2))
= sum(faoYears1 $ (abs(faoYears.pos - faoYears1.pos) le 2), p_faoProjection(rNat,i,faoItems,faoYears1))
/ sum(faoYears1 $ ((abs(faoYears.pos - faoYears1.pos) le 2) $ p_faoProjection(rNat,i,faoItems,faoYears1)), 1);
```

And derive from there the crop yields:

```
p_faoProjection(rNat,i,"yield",faoYears) $ p_faoProjection(rNat,i,"1000ha",faoYears)
= p_faoProjection(rNat,i,"1000t",faoYears) / p_faoProjection(rNat,i,"1000ha",faoYears);
```

### 3.12.11 Revovering crop land expansion

In order to replicate closely the crop land expansion of the FAO, we endogenize a shifter of the productivity of all other primary factors, i.e. besides land, of the crop production processes. While the land productivity of each crop is fixed to the FAO yield projections according to (72), cropland expansion in total over the AEZ in a model region is restricted by the following equation.

$$\sum_{aez} x_{AezNest}^{cropLand}_{r,aez,t} = \frac{faoProj_{r,tot,t}^{yld}}{faoProj_{r,tot,t0}^{yld}} \sum_{aez} x_{AezNest}^{cropLand}_{r,aez,t} + \left( \frac{cropPrice_{r,t}}{cropPrice_{r,t0} pGdpMp_{r,t}} \right)^{0.15} - 1 \quad (73)$$

We add hence a driver depending on the change in average crop prices  $v\_cropPrice$  relative the GDP price deflator  $pGdpMp$  with an elasticity of 0.15 to the FAO projections which can provoke in some cases quite strong land rent changes over time. The related equations are as follows:

```
*
* --- crop land land expansion according to FAO (uniformly applied to sub-regions and aezs)
*
e_cropLand(rs,ts(t)) $ ((v_lambdaFCrop.range(rs,t) ne 0) $ v_cropPrice.l(rs,"%t0%")) ..
*
* --- regional crop land change follows FAO projection
*
[ sum( (r_r(rs,rsNat),faoYears) $ (faoYears.pos eq t.pos),
p_faoProjection(rsNat,"tot","1000ha",faoYears)/p_faoProjection(rsNat,"tot","1000ha","2012"))
*
* --- effect of crop price changes relative to GDP price deflator, elasticity 0.15
*
+ ( v_cropPrice(rs,t)/(v_cropPrice.l(rs,"%t0%") * sum(r_r(rs,rsNat),pgdpmp(rsNat,t))))**0.15 -1
]
=E= v_cropLandChg(rs,t);
```

The price index for crops  $v\_cropPrice$  is defined as:

```
e_cropPrice(rs,ts(t)) $ ( (v_lambdaFCrop.range(rs,t) ne 0) $ v_cropPrice.l(rs,"%t0%")) ..
v_cropPrice(rs,ts)*sum(a $ p_agrYield(rs,a), m_xp(rs,a,ts)) =E= sum(a $ p_agrYield(rs,a), m_xp(rs,a,ts)*m_px(rs,a,ts));
```

And the relative crop land change  $v\_cropLandChg$  over the AEZ in a region  $rs$  as:

```
*  
* --- define crop land change, aggregation over (potentially) sub-regions and AEZ  
*  
e_cropLandChg(rs,ts) $ ((v_lambdaFCrop.range(rs,ts) ne 0) $ sum( aezFlag(rs,aez), xaezNest.l(rs,aez,"cropLand","%t0%")) ..  
  v_cropLandChg(rs,ts) -E=  
  sum( aezFlag(rs,aez) $ xaezNest.l(rs,aez,"cropLand","%t0%"),xaezNest(rs,aez,"cropLand",ts))  
  / sum( aezFlag(rs,aez) $ xaezNest.l(rs,aez,"cropLand","%t0%"),xaezNest.l(rs,aez,"cropLand","%t0%"));
```

No further land use and management information besides the harvested and sown areas for the explicitly covered crops is reported by FAO. We hence miss information on the development of pasture lands or areas devoted to fodder crops such as alfalfa and their productivity developments. This requires further assumptions. The AEZ model assumes some transformation between arable and pasture lands which suggest that returns-to-land to these two categories cannot deviate too much from each other. When letting the model endogenously determine how much pastureland versus forestry is used at given total managed land expansion, we found in some case substantially different developments in cropland and pastureland returns. This impacts cost relations between concentrate feed and feed from pastureland, and between livestock and crop production in a hard to control manner. We therefore opted to render the land productivity in livestock production endogenous to ensure that the prices of crop and pastureland develop in parallel.

In order to project changes in total managed land driven by cropland projections, we processed the land cover statistics from FAO (FAO 2020) and run regressions per country which relate the yearly change in 1000 ha of “Primary forest”, “Managed forest” and “Meadows and pastures” on a constant term (= linear trend), a trend variable (= quadratic trend) and the yearly change in 1000 ha of “Crop land”. The sample for a country comprises all years, potentially starting in 1961, for which all variables are observed. The table below reports as an example the results for Brazil for which data are available from 1990 to 2015. The regression coefficients indicate, that on average of this period, a 1 ha change in cropland provokes a reduction of 0.13 ha of primary forest, of 0.29 ha of managed forest, of 0.54 ha of meadows and pastures and of 0.02 ha for other lands. The coefficients for the quadratic trend suggest a speeding up of the overall land conversion processes. In the last year 2017, the quadratic trend for crop land would imply  $(2017 - 1990 = 27)$  a yearly expansion of  $(-263,60 + 27 * 37 \sim 740.000$  ha in cropland. Considering these changes in crop lands and the effects of the trends, primary forest would be reduced by  $\sim 140.000$  ha, managed ones by  $-608.000$  ha, while meadows and pastures would expand by  $335.000$  ha and other land shrink by  $321.000$  ha. This suggests a strong relation between crop land change and deforestation in this example.

**Table 16: Regression results for land use changes from FAO land cover data set for Brazil**

		par	pVal	tVal	R2	df
Cropland	const	-263,59989	0,3941049	-0,8671538		
Cropland	t	37,1091453	0,06175627	1,95576466		
Cropland					0,13269777	25
Other land	Cropland	-0,0265623	0,83592004	-0,20938		
Other land	const	4350,69979	1,60E-17	22,2318278		
Other land	t	-172,32039	1,37E-12	-13,333918		
Other land					0,89627975	24
Primary Forest	Cropland	-0,1377859	0,01822203	-2,5338963		
Primary Forest	const	-1077,3289	3,03E-12	-12,843341		
Primary Forest	t	38,3359513	3,71E-07	6,92055666		
Primary Forest					0,66617658	24
Managed forest	Cropland	-0,2939631	0,00345819	-3,2432559		
Managed forest	const	-2316,0513	1,23E-14	-16,564657		
Managed forest	t	71,2875334	5,89E-08	7,72063332		
Managed forest					0,71367956	24
Meadows and pastures	Cropland	-0,5416888	6,07E-10	-9,8927345		
Meadows and pastures	const	-957,31958	4,04E-11	-11,333636		
Meadows and pastures	t	62,6969057	4,78E-11	11,2399276		
Meadows and pastures					0,87308303	24

If, for instance, crop land expansion was in the past accompanied by reductions of meadows and pastures, as in the example, above, a 1 ha crop land expansion will lead to a less than 1 ha expansion of natural vegetation. But the opposite can be clearly observed as well if crop and pasture land were both expanding. In order to not overly restrain the model, we only exogenize the overall change in natural cover driven by total FAO's crop land expansion estimate and the regression coefficients, and let the CET and CES nests in the GTAP-AEZ module determine changes in managed forest and pasture lands.

In order to estimate the effect of exogenous crop land change on the change in natural vegetation (primary forest and other land) for a model region and a year during baseline construction, we first aggregate the regression coefficients LUCCRegrCoeff from single country to model region, using the current cropland cover (latest data reported by FAO, at least 2015) as weights.

Another issue is the distribution of the nation-wide projected changes in crop land expansion to the different AEZ. We use as a distribution key distKey which considers the minimum of current cropland, pastureland and managed forest, the latter two divided by minus their regression coefficient to reflect expansion possibilities. If, for instance, 1 ha of cropland expansion is accompanied by 0.5 ha reduction of pastureland and only 10% pastureland per ha cropland exist, existing pastureland limits the expansion and reduces the distribution key. We add to this our estimates of the crop land buffer, divided again by the regression coefficient. Here, we add 25% of pastureland to croplands to define a minimum. In most cases, the distribution key is equal to current crop land cover.

In order to correct for cases where the interaction of the regression coefficients, the distribution keys and the exogenous projection of total crop land and crop yields leads to implausible price developments across AEZs and in total, we add two terms. The first term considers the relation between the changes in the returns to land at AEZ level relative to returns to land at the level of the model region, with an elasticity of 0.5. That means that land expansion will be stronger in AEZ with above average land rent increases. Second, we add an elasticity to changes in land rents of 0.15. These two elasticities have two important impacts. First, they imply an endogenous correction of the distribution keys, and second, they correct potentially unwanted consequences of the empirically estimated relation between crop and total land expansion on the development of land prices:



$$x_{AezNest_{r,aez,t}}^{top} = x_{AezNest_{r,aez,t_0}}^{top} \left( \text{distkey}_{r,aez} \text{LUCCRegrCoeff}_{r,natVeg} \frac{\text{faoProj}_{r,tot,t}^{ha}}{\text{faoProj}_{r,tot,t_0}^{ha}} + \left[ \frac{p_{AezNest_{r,aez,t}}^{top}}{p_{ft_{r,lnd,t}}} \right]^{0.5} - 1 + p_{ft_{r,lnd,t}} ** 0.15 - 1 \right) \quad (74)$$

The equation in the code looks as follows:

```
*
* --- changed in total managed land, derived from exogenous change in crop land and regression results
*
e_topLand(aezFlag(rs,aez),ts(t)) $ (xaezNest.l(rs,aez,"top", "%t0%") gt eps) ..
*
* --- absolute change in managed land
*
v_topLandChange(rs,aez,t)/xaezNest.scale(rs,aez,"top", "%t0%")
=E=
*
* --- distribution key of changes in total managed land cover from model regions to AEZ
*
p_trackLand(rs,aez,"top_DistKey2", "%t0%")/xaezNest.scale(rs,aez,"top", "%t0%")
*
* * sum(r_r(rs,rsNat),
*
* --- statistically estimated relation between change in total managed land cover and in crop land
* times the crop land cover (strictly positive)
*
p_topLandChangeShare(rsNat)*p_trackLand(rsNat,"","cropLand", "%t0%")
*
* --- crop land change from FAO relative to bechmark year
*
* sum((faoYears) $ (faoYears.pos eq t.pos),
*   p_faoProjection(rsNat,"tot","1000ha",faoYears)/p_faoProjection(rsNat,"tot","1000ha","2012")-1)
*
*
*
* + xaezNest.l(rs,aez,"top", "%t0%")/xaezNest.scale(rs,aez,"top", "%t0%")
*
*
* --- stronger expansion in AEZ which above average price increases, elasticity 2
*
* { (paezNest(rs,aez,"top",ts)/paezNest.l(rs,aez,"top", "%t0%"))/ sum(lnd,pft(rs,lnd,ts)) } ** 0.5 - 1
*
* --- all AEZ expand according to change in land price (elasticity 0.15)
*
* + sum(lnd,pft(rs,lnd,ts)) ** 0.15 - 1]
*
;
```

Setting the crop land expansion exogenous requires an additional endogenous variable, here chosen is a shift in the productivity of all primary factors with the exemption of land and water. The latter two are driven by fixed crop yields. The related equations looks as follow:

```
*
* --- map uniform shift in non-land productivity for all crops to individual ones, considers how strong the land productivity is changed
* to recover the exogenous yield projection
*
e_lambdaShiftCrop(rs,f,a,ts(tRun)) $ (p_agrYield(rs,a) $ xfFlag(rs,f,a) $ (lambdaFShift.range(rs,f,a,tRun) ne 0)
*   $ (not lnd(f)) $ (not sameas(f,"water"))) ..
*
lambdaFShift(rs,f,a,tRun) =E= v_lambdaFCrop(rs,tRun)
*   + sum(lnd,lambdaF(rs,lnd,a,tRun)*sum(al $ p_agrYield(rs,al),xf.l(rs,lnd,al,tRun-1))
*   /sum(al $ p_agrYield(rs,al),lambdaF(rs,lnd,al,tRun)*xf.l(rs,lnd,al,tRun-1))) ** 0.001;
```

As seen, we introduce a correction for very strong changes in land productivity (the elasticity of 0.001 will only correct the shifter if the land productivity changes are very strong).

A further point relates to pasture land where the FAO does not project productivity changes. We opt to assume that in the long run, the land rents in the same regional units for pasture and crop lands should not develop differently:



```

*
* --- average changes over AEZs in pasture land prices equal to change in crop land price
*
e_pastLandPrice(rs,ts) $ ( (v_lambdaLndLstk.range(rs,ts) ne 0) $ sum(aez,xAezNest.l(rs,aez,"pastureLand","%t0%"))
                        $ sum(aez,xAezNest.l(rs,aez,"cropLand","%t0%"))) ..

sum(aez $ xAezNest.l(rs,aez,"pastureLand","%t0%"),
xAezNest.l(rs,aez,"pastureLand","%t0%"))*pAezNest(rs,aez,"pastureLand",ts)/pAezNest.l(rs,aez,"pastureLand","%t0%"))
/sum(aez, xAezNest.l(rs,aez,"pastureLand","%t0%"))
=E=
sum(aez $ xAezNest.l(rs,aez,"cropLand","%t0%"),
xAezNest.l(rs,aez,"cropLand","%t0%"))*pAezNest(rs,aez,"cropLand",ts)/pAezNest.l(rs,aez,"cropLand","%t0%"))
/sum(aez, xAezNest.l(rs,aez,"cropLand","%t0%"));

```

And drive this exogenous relation by a uniform shifter of the land productivity in livestock processes:

```

*
* --- map uniform shift in land productivity for all animals to individual ones
*
e_lambdaShiftLstk(rs,lnd,lstk,ts) $ (xFlag(rs,lnd,lstk) $ (lambdaFShift.range(rs,lnd,lstk,ts) ne 0)) ..

lambdaFShift(rs,lnd,lstk,ts) =E= v_lambdaLndLstk(rs,ts);

```

Without this mechanism, land productivity in animals would follow the general productivity increases in the primary sectors, which could imply quite unrealistic developments relative to crops for which crop yields developments are given from the exogenous projections.

We tested the framework with differently detailed data base with regard to the country aggregation and found that it solved smoothly in most cases. We cannot exclude that especially for projections up to 2050, the overall framework including the G-RDEM mechanism might provoke infeasibilities. In such cases, one might try to increase the price elasticity effect in (3) as implausible high or low land prices can lead to frictions in other parts of model.

In this context it is worth noting that understanding the causes of infeasibilities in baseline construction can be challenging. A (too) low supply elasticity of natural resources, to give an example, can shift global income to a few countries with substantial production of related primary commodities and at the same time provoke quite unbalanced price developments across sectors. These reactions interact with the implicit correction via TFP changes to recover the given real GDP developments.

### 3.12.12 Steering of G-RDEM model runs

The screenshot presented in Figure 14 below shows how the user can select which data base aggregation to use, decide the time horizon of the baseline as well as the report frequency. For instance, retaining the flexible aggregation approach from GTAP in G-RDEM allows the user to develop a long-term baseline with a focus on only one country, while aggregating the rest of the world into a single aggregate. So far, the G-RDEM module has been tested with resolution with up to 80 countries.

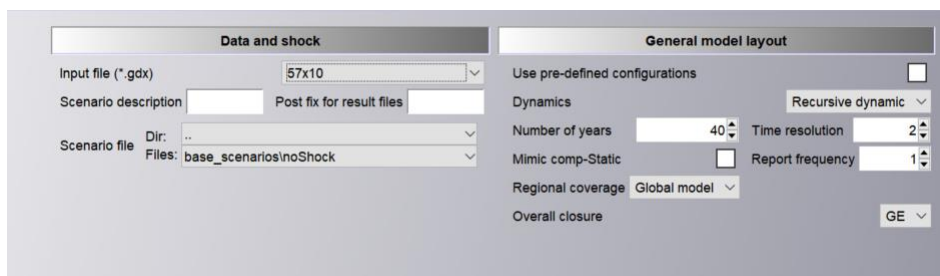


Figure 14 : Main steering panel of CGEBox

The set of years to run will always include the benchmark (t0), the last year (eq card(t)) and the first year of the simulation period, which can be used e.g. for estimations from the initialized variables.

```

$iftheni "%dynMode%"=="Recursive dynamic"

    trun(t0) = YES;
    trun(t) $ (t.pos eq card(t)) = YES;
    trun(t) $ (t.pos eq 2) = YES;
    trun(t) $ (mod(t.pos-1,%timeStep%) eq 0) = YES;

```

For the above input, the resulting years to run are:

```

---- 1406 SET trun

t00,    t01,    t05,    t10,    t15,    t20,    t25,    t30,    t35,    t40

```

In order to construct the baseline, the user selects (Figure 15) the desired SSP scenario for population and GDP, as well some procedural modules to use.

Figure 15 : Main steering panel of the G-RDEM module in CGEBox

### 3.12.13 Population and GDP growth from the IIASA SSPs

IIASA host a web application<sup>63</sup> which gives access to model results relating to the IPCC model activities comprising GDP growth rate projection by country from a range of models and for different Shared Socio-Economic Pathways (SSPs). Some if GDP projections were downloaded in XLSX format and converted into a GDX container. That container is read by the driver program:

```

*
* --- load GDP projections from 2005 to 2100, in 10 year steps
*
execute_load "%datDir%\worldPopProj.gdx" wpp15_Popt;

set modP /"OECD Env-Growth","IIASA GDP"/;
set scen / SSP1,SSP2 /;
set var /"GDP|PPP" /;
set unit /"billion US$2005/yr" /;
set gYears / 2010,2020,2030,2040,2050,2060,2070,2080,2090,2100/;
parameter p_gdp(modP,scen,popreg,var,unit,gYears);
execute_load "%datDir%\GDP_SSP-IIASA.GDX" p_gdp;
option p_gdp:2:5:1;
parameter p_gdpC(popreg,*);
p_gdpC(popreg,gYears) = p_gdp("OECD Env-Growth","SSP2",popreg,"GDP|PPP","billion US$2005/yr",gYears);

```

The labels for the country follow the same codes as for the UN population projections. As seen from above, the projections are not yearly, but by decade. In order to use them flexible with different time steps, the code first interpolates linearly between given time points:

<sup>63</sup> [http://www.iiasa.ac.at/web/home/research/researchPrograms/Energy/SSP\\_Scenario\\_Database.html](http://www.iiasa.ac.at/web/home/research/researchPrograms/Energy/SSP_Scenario_Database.html)

```

*
* --- interpolate linearly between simulation points
*
set pYears_gYears(allYears,gYears,gYears) / (2010*2019).2010.2020
                                              (2020*2029).2020.2030
                                              (2030*2039).2030.2040
                                              (2040*2049).2040.2050
                                              (2050*2059).2050.2060
                                              (2060*2069).2060.2070
                                              (2070*2079).2070.2080
                                              (2080*2089).2080.2090
                                              (2090*2100).2090.2100 /;

alias(gYears,gYears1);

p_gdpC(popReg,allYears) = sum(pYears_gYears(allYears,gYears,gYears1),
                             p_gdpC(popreg,gYears)
                             + (p_gdpC(popreg,gYears1)-p_gdpC(popreg,gYears))/10 * mod(allYears.pos-1,10));

```

The mapping to the regional aggregate used in CGEBox is structural identical to assigning the population projections:

```

*
* --- groth rate between simulation points
*
p_popGrowth(r,tRun) $ (sum(r_popr(r,popreg),1) $ (tRun.pos > 1))
= {[ sum( (r_popr(r,popreg),t_yp(tRun,pYears)), wpp15_popt("MEDP",popreg,pYears))
    /sum( (r_popr(r,popreg),t_yp(tRun-1,pYears)), wpp15_popt("MEDP",popreg,pYears)) ] -1} * 100;
..

```

For some countries, notably small islands, no projections are available. In these cases, global unweighted averages are used instead.

```

*
* --- use average of counties with data for countries with missing ones
*
p_gdpGrowth(r,tRun) $ (sum(sameas(r,emptyGdp),1) $ (tRun.pos > 1))
= sum(rr, p_gdpGrowth(rr,tRun))/sum(rr $ p_gdpGrowth(rr,tRun),1);
display p_gdpGrowth;

```

For use in the model, per capita growth is used:

```

*
* --- GPD growth per capita is the economy wide growth rate
*      minus population growth rate
*
p_gdpPerCapGrowth(r,tRun) $ (tRun.pos > 1)
= p_gdpGrowth(r,tRun) - p_popGrowth(r,tRun);
..

```

The interface allows to overwrite these projections based on external input. In order to do so, the “Use additional macro projection” checkbox needs to be switched on. The user can then select a scenario file from the folder “*gams/scen/macro*”:

A dummy implementation of overwriting the data from the IIASA portal is shown below (*see gams/scen/macro/macroproj\_dummy.gms*):

```
*
* --- example format to map from data in 1980 ... format to p_driver
*
set myProjItems / GDP /;
set myProjYears(allYears) / 2014*2050 /;
parameter p_myProj(rNat,myProjItems,myProjYears);
option kill=p_myProj;

*
* --- example format to set GDP|PPP
*
p_drvc(rNat,"GDP|PPP",tRun) = p_myProj(rNat,"GDP",myProjYears);
```

### 3.12.14 Using the G-RDEM generated baseline for counterfactual analysis

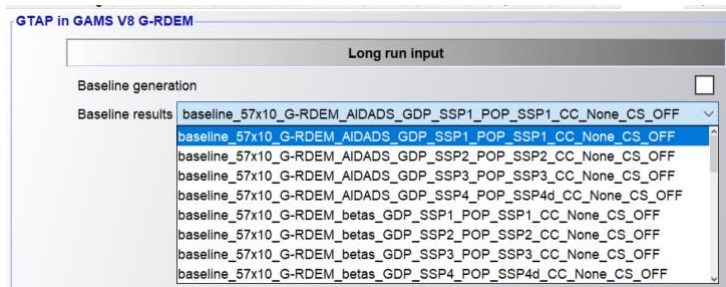
Besides analysing the baseline itself or comparing different baselines against each other, they also might serve as a starting point for counterfactual analysis. Two types of simulations are currently supported in the CGEBox platform. The first one uses the shifters generated during the baseline construction (changes in saving rates, productivity shifters, updated I-O coefficients), as well as population projections, as given parameters into a recursive-dynamic simulation. That makes only sense if the structural set-up of the model is unchanged (production function nesting, bi-lateral trade, etc.). During such a run, GDP would no longer be fixed, but endogenous and, as usual, shocks can be introduced, such as changes in policies or productivity, like climate change impacts on agricultural yields. Parameter estimates for some impacts (like those of climate) are already available in CGEBox and can be readily inserted during the simulation runs, by means of the graphical interface. For an application, see for instance Wilts et al. 2021<sup>64</sup>.

The second type of possible utilization of G-RDEM is in the provision of a benchmark data set data for comparative-static analysis. In this case, the global SAM and parameters for one of the simulated time points are loaded as a replacement for the benchmark data from GTAP. The full modular

<sup>64</sup> Wilts, R., Latka, C., Britz, W. (2021): Who is Most Vulnerable to Climate Change Induced Yield Changes? A Dynamic Long Run Household Analysis in Lower Income Countries, *Climate Risk Management* 33(100330)

flexibility of CGEBox could then be exploited, allowing to modify several characteristics of the model. That approach might be also interesting to produce not a long-run, but a rather a medium-term baseline as an ex-ante benchmark, which is for instance the usual practice in partial equilibrium modelling of agri-food markets.

The first option is to use the G-RDEM module in non-baseline generation mode:



In the case, the user can select the baseline results to employ. That will load the necessary shifters and exogenous parameters into the GAMS code:

```
variable axpPrev(r,a,t),betasPrev(r,t),betagPrev(r,t),betapPrev(r,t);
parameter ioPrev(r,i,a,t),ioNestPrev(r,i,a,t),avaPrev(r,a,t),andPrev(r,a,t);
$$GDXIN "%resdir%/run/%baselineRes%"
$$LOAD axpPrev = axp
$$LOAD betasPrev = betas
$$LOAD betagPrev = betag
$$LOAD betapPrev = betap
$$LOAD ioPrev = io
$$LOAD ioNestPrev = ioNest
$$LOAD andPrev = and
$$LOAD avaPrev = ava
$$LOAD p_timeStep
$$LOAD p_growthRate
$$GDXIN
```

We have checked that without additional shocks, the existing baseline will be recovered without that GDP is fixed.

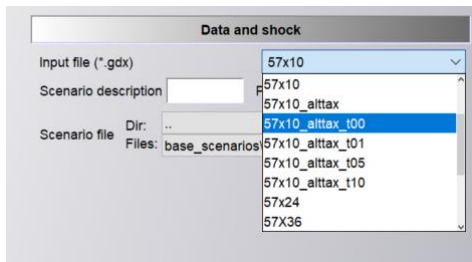
The second option requires that during baseline generation, altertax output is activated:



That will output the SAM and other information required for a benchmark calibration of the model are stored in “build” directory:

```
IDE 57x10_alttax_t01.gdx
IDE 57x10_alttax_t05.gdx
IDE 57x10_alttax_t10.gdx
IDE 57x10_alttax_t00.gdx
```

These benchmark data can be used similarly to the output from the data preparation step. The data can therefore be chosen as a starting point for comparative (or even recursive-dynamic) analysis:



That makes the layout of the model fully adjustable (e.g., by adding myGTAP for several household types, a different production nesting, changes in how bi-lateral trade is modelled).

### 3.12.15 Specific nestings in G-RDEM

We add some specific nestings for the production function (see `gams\gtaprdem\gtaprdem.gms`):

```
set natResA(a); natResA(a) $ sum(r, SAM0(r,"natRes",a)) = YES;

sigma(r,natResA) = 0.20;
tNest("CAP+NATRES") = YES;
tNest_n_a("UA","CAP+NATRES",natResA) = YES;
tNest_f_a("CAP+NATRES","capital",natResA) = YES;
tNest_f_a("CAP+NATRES","natRes",natResA) = YES;
sigmaNest(r,"CAP+NATRES",natResA) = 0.20;

tNest_n_a(tNest,"CAP+NATRES",a) $ ((not sameas(tNest,"CAP+NATRES")) $ tNest_f_a(tNest,"capital",a)) = YES;
tNest_f_a(tNest,"capital",a) $ (not sameas(tNest,"CAP+NATRES")) = no;
tNest_n_a("UA","CAP+NATRES",a) $ sum(tNest_n_a(tNest,"CAP+NATRES",a),1) = NO;

$$include 'extensions\labor_nest.gms'

set landA(a); landA(a) $ sum(r, SAM0(r,"land",a)) = YES;
tNest("labLand") = YES;
tNest_n_a("UA","LabLand",landA) = YES;
tNest_f_a("LabLand","Land",landA) = YES;
tNest_n_a("LabLand","Labor",landA) = YES;
tNest_n_a("UA","Labor",landA) = NO;
sigmaNest(r,"LabLand",landA) = 0.5;
```

GTAP-AGR has proven to work badly with some long-term projections, because of limited factor mobility between agricultural and non-agricultural sectors. Therefore, only the technology nests from GTAP-AGR are taken over:

```
$if not defined roecd $include 'gtapAgr\GtapAgr_def.gms'

*
* --- define feed demand sub-nest in livestock production
*
if ( card(feed),
    tNest("feed") = yes;
    tNest_i_a("feed",feed,lstk) = yes;
    tNest_n_a("ND","feed",lstk) = YES;
    sigmaNest(r,"Feed",lstk) = 2;
);

if ( card(agr_c),
    tNest("FoodPrc") = yes;
    tNest_i_a("FoodPrc",agr_c,foodPrc) = yes;
    tNest_n_a("ND","FoodPrc",foodPrc) = YES;
    sigmaNest(r,"FoodPrc",foodPrc) = 0.5;
);
```

In order to reflect stronger substitutional relations between animal and crop based products, two final demand nests are constructed:



```

set cropD(i); cropD(i) $ sum(mapi(iCrops,i),1) = YES;
cropD(i) $ mapi("vol",i) = YES;
cropD(i) $ mapi("pcr",i) = YES;
cropD(i) $ mapi("sgr",i) = YES;
cropD(i) $ mapi("ofd",i) = YES;
cropD(i) $ mapi("b_t",i) = YES;

dNest("cropD") = YES;
dNest_i_fd("cropD",cropD,fdn) = YES;
dNest_n_fd("top","cropD",fdn) = YES;
sigmaFDNest(r,"cropD",fdn) = .50;

set animD(i); animD(i) $ sum(mapi(1stk0,i),1) = YES;

animD(i) $ mapi("cmt",i) = YES;
animD(i) $ mapi("omt",i) = YES;
animD(i) $ mapi("mil",i) = YES;

dNest("animD") = YES;
dNest_i_fd("animD",animD,fdn) = YES;
dNest_n_fd("top","animD",fdn) = YES;
sigmaFDNest(r,"animD",fdn) = .50;

```

### 3.12.16 Endogenous non-Hicks neutral technical progress and primary factors supply

In addition to price driven changes in endowments, the marginal productivity can also be rendered price dependent. In the default setting, the labour force is reacting to population changes fully, while land, irrigation water and natural resources follow to a limited extend (positive changes) in population. The other element are by default switched off.

Elasticities to population growth		Hicks non-neutral TP to factor prices		Factor supply to factor prices	
Land	0.05	Land TP	0.00	Land XF	0.00
Irrigation water	0.05	Irrigation water TP	0.00	Irrigation water XF	0.00
Natural resources	0.20	Natural resources TP	0.00	Natural resources XF	0.00
Labour force	1.00	Labour force TP	0.00	Labour force XF	0.00
		Capital TP	0.00		

The input from the GUI is mapped to the following symbols in the GAMS code:

```

*
* --- link changes in factor endowments to changes in population,
*
p_popShift(r,"land") = %landToPopElas%;

$$ifthen.GTAPWATER %nFactors%>8
p_popShift(r,"water") = %WaterToPopElas%;
$$endif.GTAPWATER
p_popShift(r,"natres") = %natResToPopElas%;
*
* --- labour shares follow population growth
*
p_popShift(r,1) = %labToPopElas%;

```



```

* --- factor expansion depending on factor price developments:
*      labor participation rates react to wage rate changes

set dummyPS / xf,tp /;
option kill=p_priceShift;

*
* --- endogenous non-Hicks neutral progress to depending on development of factor prices
*
p_priceShift(r,"tp","land")          = %landTpToPfElas%;
$$ifthere.GTAPWATER %nFactors%>8
  p_priceShift(r,"tp","water")       = %waterTpToPfElas%;
$$endif.GTAPWATER
p_priceShift(r,"tp","natres")        = %natResTpToPfElas%;
p_priceShift(r,"tp",1)               = %labTpToPfElas%;
p_priceShift(r,"tp",cap)             = %capTpToPfElas%;
*
* --- factor supply to factor price
*
p_priceShift(r,"xf","land")          = %landXfToPfElas%;
$$ifthere.GTAPWATER %nFactors%>8
  p_priceShift(r,"xf","water")       = %waterXfToPfElas%;
$$endif.GTAPWATER
p_priceShift(r,"xf","natres")        = %natResXfToPfElas%;
p_priceShift(r,"xf",1)               = %labXfToPfElas%;

```

We also take changes in education levels into account:

```

*
$$ifthere.GTAPVersion %GTAPNO%<9
  p_popShift(rnat,1) $ ff("skLab",1)
    = (1+p_growthRate(rnat,"tertiary",tSim)/100) / (1+p_growthRate(rnat,"population",tSim)/100);
  p_popShift(rnat,1) $ ff("unskLab",1)
    = (1+p_growthRate(rnat,"primary",tSim)/100) / (1+p_growthRate(rnat,"population",tSim)/100);
$$else.GTAPVersion
  p_popShift(rnat,1) $ ff("off_mgr_pros",1)
    = (1+p_growthRate(rnat,"tertiary",tSim)/100) / (1+p_growthRate(rnat,"population",tSim)/100);
  p_popShift(rnat,1) $ ff("ag_othlowsk",1)
    = (1+p_growthRate(rnat,"primary",tSim)/100) / (1+p_growthRate(rnat,"population",tSim)/100);
$$endif.GTAPVersion

```

The population and factor price shifters are implemented as follows:

```

*
* ---- shift factor supply depending on population and factor prices
*
aft(r,fm,tSim) $ aft(r,fm,tSim)
  = aft(r,fm,tSim-1)
    * (1 + sum(r_r(r,rNat) $ (p_popt(rNat,tSim) gt p_popt(rNat,tSim-1)),
      p_popShift(rNat,fm) * (p_popt(rNat,tSim)/p_popt(rNat,tSim-1)-1))
    + sum(r_r(r,rNat),p_priceShift(rNat,"xf",fm)) * min(10.0,max(-0.90+0.9 $(not 1(fm)), (pft.1(r,fm,tSim-1)-1))) );

gf(r,fnm,a,tSim) $ xf.1(r,fnm,a,tSim-1)
  = gf(r,fnm,a,tSim-1)
    * (1 + sum(r_r(r,rNat) $ (p_popt(rNat,tSim) gt p_popt(rNat,tSim-1)),
      p_popShift(rNat,fnm) * (p_popt(rNat,tSim)/p_popt(rNat,tSim-1)-1))
    + sum(r_r(r,rNat),p_priceShift(rNat,"xf",fnm)) * min(10.0,max(-0.90+0.9 $(not 1(fnm)), (pf.1(r,fnm,a,tSim-1)-1))) );

```

The resulting technical progress changes are calculated as follows:

```

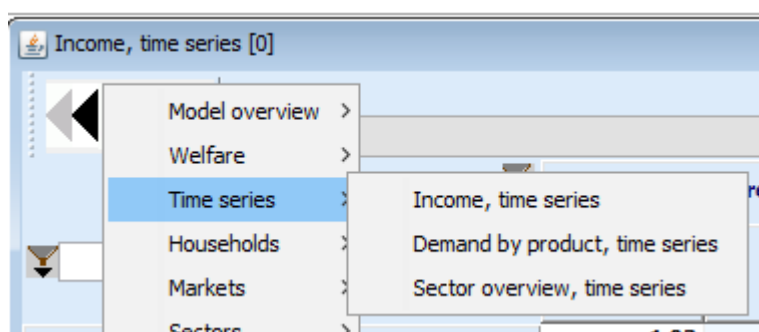
: *
: * --- productivity of factors is updated depending on factor price changes and GDP growth
: *
:
: lambda_daf.fx(r,f,a,tSim) $ xf.1(r,f,a,tSim-1)
:   = m_lambda_daf(r,f,a,tSim-1) * (1.0 + sum(r_r(r,rNat),p_priceShift(rNat,"tp",f)
:     * min(10,max(-0.90,(
:       pf.1(r,f,a,tSim) $ (fnm(f) or (omegaf(r,f) ne inf))
:       + pft.1(r,f,tSim) $ (fm(f) $ (omegaf(r,f) eq inf)) -1)))
:   + p_growthRate(rNat,"GDPPerCAP",tSim)/100* 0.5*0
:   + p_GrowthRate(rNat,"Population",tSim)/100*(1-p_popShift(rNat,f))*0.5)*0);
:

```

### 3.12.17 Analysis of G-RDEM simulation output (exploitation)

Loading all results from long time recursive dynamic with detailed global SAMs might require amounts of memory exceeding most personal computer configurations. Therefore, the exploitation selection allows to pick up only a few years for inspecting the results. Therefore, while all output variables are stored in the GDX repository output file, only the selected ones will be loaded into the memory.

Generally, any view existing in the exploitation tool can be changed to a time series using the pivot facility. In order to ease the exploitation, three views have been pre-configured, under the heading “Time series”:



The first one reports changes in GDP, price indices etc:

Regions		Total or household groups							
World		Total							
10x10_example_recDyn									
	t1	t2	t3	t4	t5	t6	t7	t8	t9
Total utility [Index]	1,02	1,04	1,06	1,09	1,11	1,13	1,16	1,18	
		2,03%	4,13%	6,31%	8,57%	10,92%	13,35%	15,86%	
Utility of private household [Index]	1,02	1,04	1,06	1,09	1,11	1,14	1,17	1,20	
		2,25%	4,64%	7,09%	9,64%	12,26%	14,98%	17,78%	
Utility of government [Index]	1,03	1,06	1,08	1,11	1,13	1,16	1,19	1,22	
		2,50%	4,81%	7,24%	9,77%	12,40%	15,11%	17,91%	
Utility of saving [Index]	1,04	1,05	1,06	1,07	1,09	1,10	1,12	1,13	
		1,08%	2,21%	3,46%	4,80%	6,21%	7,72%	9,31%	
GDP [Bio \$]	50969,86	52256,51	53722,55	55362,24	57135,19	59017,39	60990,06	63039,17	
		2,52%	5,40%	8,62%	12,10%	15,79%	19,66%	23,68%	
GDP, normalized with price index [Bio \$]	50444,38	51762,22	52973,64	54201,20	55445,69	56709,14	57994,27	59302,57	
		2,61%	5,01%	7,45%	9,91%	12,42%	14,97%	17,56%	
GDP, normalized with price index, per cap. [Bio \$]	7538,05	7653,68	7752,19	7852,07	7953,58	8057,09	8163,00	8271,47	
		1,53%	2,84%	4,17%	5,51%	6,89%	8,29%	9,73%	
Price index [Index]	1,01	1,01	1,01	1,02	1,03	1,04	1,05	1,06	
		-0,09%	0,37%	1,09%	1,96%	3,00%	4,08%	5,20%	
Consumer price [Index]	1,02	1,02	1,02	1,03	1,04	1,05	1,06	1,07	
		-0,00%	0,49%	1,24%	2,17%	3,20%	4,31%	5,44%	
Government price [Index]	1,00	1,00	1,01	1,02	1,03	1,04	1,05	1,06	
		0,45%	0,98%	1,77%	2,69%	3,71%	4,78%	5,89%	
Savings price [Index]	0,99	1,00	1,01	1,02	1,04	1,05	1,07	1,09	
		0,54%	1,54%	2,82%	4,28%	5,87%	7,54%	9,24%	
Factor income [Bio \$]	40419,89	41420,74	42554,94	43820,08	45185,25	46632,30	48146,78	49717,98	
		2,48%	5,28%	8,41%	11,79%	15,37%	19,12%	23,00%	
Factor income, normalized with price index [Bio \$]	40003,18	41028,95	41961,71	42901,09	43849,11	44808,44	45781,84	46770,98	
		2,56%	4,90%	7,24%	9,61%	12,01%	14,45%	16,92%	
Current account balance [Bio \$]	-0,00		-0,00	-0,00	-0,00	0,00	-0,00		
			0,00%	0,00%	-100,00%	300,00%	-300,00%		
Tax income [Bio \$]	8862,84	9096,80	9366,76	9670,67	10000,95	10352,84	10722,88	11108,39	
		2,64%	5,69%	9,11%	12,84%	16,81%	20,99%	25,34%	
Population size [Mil heads]	6691,97	6763,05	6833,37	6902,79	6971,16	7038,42	7104,53	7169,54	
		1,06%	2,11%	3,15%	4,17%	5,18%	6,16%	7,14%	

The second one focuses on the demand side:

## G-RDEM: Long-term baseline generation and analysis

Demand by product, time series [0]																
	Regions			Items			Sectors and institutions			Origins			Percentage diff. to			
	World			Quantity			Total			Total			Years			
	t1	t2	t3				t7	t8	t9	t10	t11	t12	t13	t14	t15	t1
Total	142485,78	146008,50	149704,95	153565,70	157563,80	161686,59	165926,36	170277,59	174730,84	179282,19	183929,30	188668,58	193499,03	198420,80	203434,33	42,78%
Total	142485,78	146008,50	149704,95	153565,70	157563,80	161686,59	165926,36	170277,59	174730,84	179282,19	183929,30	188668,58	193499,03	198420,80	203434,33	42,78%
Grains and Crops	2916,05	3018,27	3111,50	3206,81	3303,73	3401,67	3501,18	3602,60	3706,14	3811,86	3919,82	4030,00	4142,37	4256,90	4373,56	49,98%
Livestock and Meat Products	2238,93	2310,26	2379,05	2447,52	2515,89	2584,10	2652,63	2721,80	2791,83	2862,86	2934,95	3008,14	3082,46	3157,93	3234,56	44,74%
Mining and Extraction	6488,89	6680,69	6887,33	7108,58	7345,69	7599,84	7871,96	8162,67	8467,89	8786,67	9118,78	9463,23	9819,83	10188,75	10570,14	62,90%
Processed Food	4741,70	4858,49	4977,09	5095,79	5214,73	5334,18	5454,69	5576,63	5700,29	5825,87	5953,49	6083,19	6215,02	6349,05	6485,32	36,77%
Textiles and Clothing	2908,33	2982,27	3056,45	3132,91	3210,85	3290,30	3371,33	3454,00	3538,37	3624,46	3712,33	3802,02	3893,54	3986,90	4082,13	40,36%
Light Manufacturing	15828,16	16196,59	16584,91	16988,13	17402,62	17827,67	18262,65	18707,15	19160,68	19623,06	20094,20	20573,87	21062,04	21558,85	22064,41	39,40%
Heavy Manufacturing	34507,74	35399,53	36334,95	37309,64	38317,01	39355,67	40424,27	41521,92	42646,20	43796,34	44972,09	46172,49	47397,37	48646,94	49921,36	44,67%
Utilities and Construction	11842,57	12158,73	12489,41	12830,10	13179,25	13536,65	13902,15	14275,62	14656,89	15045,89	15442,59	15846,90	16258,82	16678,44	17105,85	44,44%
Transport and Communication	19758,69	20250,46	20774,90	21322,58	21889,05	22472,11	23070,19	23682,21	24307,76	24946,49	25598,09	26262,30	26939,04	27628,33	28330,25	38,81%
Other Services	41254,72	42153,21	43109,35	44123,64	45185,03	46284,40	47415,33	48572,99	49754,80	50958,69	52182,95	53426,46	54688,54	55968,72	57266,74	38,81%

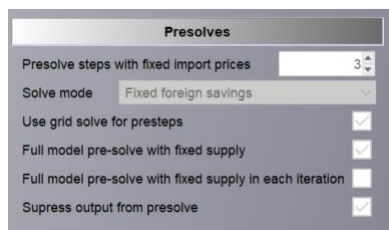
The third one reports production quantities and related input demands:

Sector overview, time series [0]																
	Regions			Items			Sectors			Origins			Percentage diff. to			
	World			Quantity			Total			Total			Years			
	t1	t2	t3	t4	t5		t10	t11	t12	t13	t14					
Total output	111673,89	114430,69	117323,98	120347,07	123478,20	Grains and Crops	9908,86	140466,48	144097,81	147800,14	151572,48	15541,7				
Output taxes	2089,11	2146,66	2214,24	2290,95	2374,69	Livestock and Meat Products	22,60%	2863,70	2971,65	3082,21	3195,17	3311,7				
Total intermediate demand	56359,58	57845,86	59441,21	61141,64	62927,28	Mining and Extraction	32,05%	2863,70	2971,65	3082,21	3195,17	3311,7				
Total intermediate input taxes	1688,87	1739,42	1800,32	1870,54	1948,00	Processed Food	25,54%	2863,70	2971,65	3082,21	3195,17	3311,7				
Total factor demand	46523,88	47852,70	49355,00	51017,93	52805,29	Textiles and Clothing	30,80%	60853,46	63038,09	65279,70	67574,77	69920,84	72311,7			
Total intermediate demand	56359,58	57845,86	59441,21	61141,64	62927,28	Light Manufacturing	26,24%	60853,46	63038,09	65279,70	67574,77	69920,84	72311,7			
Grains and Crops	1444,20	1507,18	1564,27	1623,01	1683,04	Heavy Manufacturing	21,91%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
Livestock and Meat Products	1029,96	1066,93	1102,40	1138,45	1174,97		25,54%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
Mining and Extraction	3407,47	3512,93	3624,66	3744,48	3873,07		29,26%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
Processed Food	1351,47	1389,92	1429,51	1470,40	1512,28		33,07%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
Textiles and Clothing	929,78	956,88	984,46	1013,42	1043,35		36,79%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
Light Manufacturing	6248,99	6403,78	6569,08	6744,22	6926,68		40,40%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
Heavy Manufacturing	15394,40	15805,70	16240,06	16697,97	17174,96		44,21%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
Utilities and Construction	3804,34	3900,58	4003,55	4113,45	4228,74		48,89%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
Transport and Communication	8912,65	9155,35	9417,58	9698,51	9994,11		52,22%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
Other Services	13836,31	14146,60	14505,64	14897,72	15316,08		55,22%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
Total factor demand	46523,88	47852,70	49355,00	51017,93	52805,29		58,29%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
Land	524,20	524,65	525,09	525,50	525,89		60,85%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
UnSkLab	15120,76	15406,20	15736,19	16159,44	16646,41		63,08%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
SkLab	9495,30	9553,90	9727,62	9953,62	10225,02		65,27%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
Capital	20739,04	21723,07	22720,96	23733,95	24762,29		67,57%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			
NatRes	644,58	644,87	645,15	645,42	645,68		69,92%	70752,24	72850,90	74999,56	77195,45	79437,23	81721,7			

### 3.12.18 Computing and data base considerations for G-RDEM runs

Compared to a comparative-static run with the model, constructing a baseline over several decades requires many solves. As indicated above, the time resolution can be chosen by the user. While larger time steps such as five year intervals decrease the number of necessary solves, the resulting shock in each period becomes larger, which implies more time spent by the solver to find a fixed point. For a dataset larger than around 57x24, it is therefore recommended to solve at least in bi-yearly steps, whereas less detailed data sets can still solve quite fast in five year steps. Using the built-in pre-solve mechanism with at least three steps is also recommended for more detailed datasets. Furthermore, we recommend aggregating over the Armington agents, which removes the agent specific differentiation

between domestic and imported shares and reduces the model complexity and rigidities in the demand composition.



Some relatively large data sets have been tested, including 33 from all 57 sectors and 80 countries from the 140 available in GTAP 9. One run over 40 years in 2 year steps with such detailed data base might require about two hours. Note that especially debt accumulation from foreign savings might lead to infeasibilities, such that it might be required to manually reduce the accounted share of related debt servings for specific regions. In the same vein, we typically used an ALTERNATE run to construct a benchmark where very high negative macro-economic saving rates are reduced (see `gams\scen\user_scenarios\corr_betas.gms`). These odd saving rates might point a specific problem in GTAP 9 data, as in the 2011 the global economy was still affected by an economic crisis.

More generally, it should be understood that baseline construction is only partially a process which can be delegated to a ready-to-use code. A close look at some key results, such as the evolution of foreign savings, saving rates and sectoral output per capita is recommended to rule out implausible findings. An ALTERNATE adjustment may be needed to vary the initial benchmark or it might be necessary to add specific shocks on top of the baseline mechanism. We also remind the reader that the SSP1 to 5 storylines comprise elements relating to governance such as policies to abate emissions or control land use change which are not integrated in the model structure discussed here.

### 3.13 GTAP-Melitz: Heterogenous firm module

This section is prepared by Yaghoob Jafari and Wolfgang Britz

#### 3.13.1 Summary

The GTAP-Melitz module replaces the Armington assumption to depict bi-lateral trade by a heterogenous firm module where each sector is assumed to consist of firms of differing productivity which each produce quality differentiated products. Firms face fix costs to enter the industry and to become active on specific trade links. The module does not need additional data, the user can select the parameters for the module on the GUI along which the sector to which the module is applied.

#### 3.13.2 Introduction

Since Armington (1969)<sup>65</sup> proposed to treat imported and domestic varieties of goods in the same classification as imperfect substitutes depicted by a CES-utility function, that approach dominated applied Computable General Equilibrium (CGE) analysis. It provides a powerful, but relatively simple framework for studying international trade policy, not at least as it can accommodate any observed pattern of trade flows and related prices, i.e. the intensive margin of trade. However, preferences for each origin in the Armington model are fixed such that changes in trade cannot impact average imported qualities on a trade link. It hence neglects potential variations at the extensive margin of trade such as trade flows in new products which are found as important in empirical analysis (Hummels and Klenow, 2005<sup>66</sup>; Chaneny, 2008<sup>67</sup>, among others).

The pioneer paper by Melitz (2003<sup>68</sup>) introduced firm productivity heterogeneity drawing from Hopenhayn (1992<sup>69</sup>) into the monopolistic competition framework by Krugman (1980)<sup>70</sup>. The Melitz model combines changes at the intensive and extensive margins of trade by allowing firms to self-select new export markets based on their productivity level. Subsequently, many papers applying the model (Bernard et al., 2003, 2006, 2007<sup>71</sup>; Eaton et al., 2004<sup>72</sup>) supported its empirical evidence by

---

<sup>65</sup> Armington, P.S. 1969. "Theory of demand for products distinguished by place of production." *International Monetary Fund Staff Papers*, 16(1): 159–178. doi:10.2307/3866403

<sup>66</sup> Hummels, D., and P.J. Klenow. 2005. "The variety and quality of a nation's exports." *American Economic Review*, 95(3): 704–723. doi:10.1257/0002828054201396

<sup>67</sup> Chaney, T. 2008. "Distorted gravity: the intensive and extensive margins of international trade." *American Economic Review*, 98(4): 1707–1721. doi:10.1257/aer.98.4.1707

<sup>68</sup> Melitz, M.J. 2003. "The impact of trade on intra-industry reallocations and aggregate industry productivity." *Econometrica*, 71(6): 1695–1725. doi:10.3386/w8881

<sup>69</sup> Hopenhayn, H. 1992. "Entry, exit, and firm dynamics in long run equilibrium." *Econometrica*, 60(5): 1127–1150. doi:10.2307/2951541

<sup>70</sup> Krugman, P. (1980), "Scale Economies, Product Differentiation, and the Pattern of Trade", *American Economic Review* 70, 950-959

<sup>71</sup> Bernard, A. B., and J. B. Jensen. (1995). Exporters, jobs, and wages in U.S. manufacturing, 1975-1987. In *Brookings Papers on Economic Activity: Microeconomics 1995*, pp. 67-112.; Bernard, A.B., J. Eaton, J.B. Jensen, and S. Kortum. 2003. "Plants and productivity in international trade." *American Economic Review*, 93(4): 1268–1290. doi:10.3386/w7688. ; Bernard, A.B., J.B. Jensen, and P.K. Schott. 2006. "Trade costs, firms and productivity." *Journal of Monetary Economics*, 53(5): 917–937. doi:10.1016/j.jmoneco.2006.05.010.; Bernard, A.B., Redding, S., Schott, P.K., 2007. Comparative advantage and heterogeneous firms. *Review of Economic Studies* 74, 31–66

reproducing salient trade patterns observed in recent micro level studies. As the Melitz model provides a more general framework to depict bi-lateral trade which has proven as empirically superior, there have been a number of efforts to introduce firm heterogeneity into CGE models (Zhai 2008<sup>73</sup>; Balistreri et al., 2011<sup>74</sup>; Oyamada, 2013<sup>75</sup>; Akgul et al., 2016<sup>76</sup>; and Dixon et al., 2016<sup>77</sup>).

The first paper which introduced firm heterogeneity following Melitz (2003) into a CGE framework is Zhai (2008). The Zhai-Melitz approach captures variations in the extensive margin of trade flows in contrast to traditional CGE models based on Armington's (1969) assumption. Zhai (2008) provided the theoretically well-grounded empirical model based on the assumption of no free entry and exit. However, the Zhai implementation allows for adjustments in the extensive margin of trade only as a result of changed export share of firms engaging in a specific trade link, while limiting the variety gains brought by new entrants. Consequently, this assumption results in overestimation of the extensive margin of trade and in turn productivity effect. Balistreri et al. (2011) overcomes that restriction by allowing also for new entrants on each trade link and accounts for a certain exogenous share of firms leaving in each period the industry. Akgul, Vitoria and Hertel (2016) introduced the Melitz framework into the standard GTAP model, abstracting from exogenous firm exit.

Besides, Balistreri and Rutherford (2013)<sup>78</sup> set out stylized versions of the Armington, Krugman, and Melitz under a unified treatment, and then compare the outcome of different approaches. Inspired by Balistreri and Rutherford (2013), Dixon et al. (2016) draws the connections between the three models by developing them sequentially as special cases of an encompassing model. Dixon et al. (2016) interestingly show that the Armington, Krugman, and Melitz models are progressively less restrictive special cases of a more generalized model, derived from a cost minimization problem of a worldwide planner. While Dixon et al. (2016) provide an illustrative numerical general equilibrium model with Melitz sectors, using the earlier version of Dixon et al. (2016), Oyamada (2014) develops a simple AGE framework within which user can switch between Armington, Krugman, and Melitz models, and shows how the Dixon et al. (2016) framework can be parameterized<sup>79</sup>.

---

<sup>72</sup> Eaton, J., S. Kortum, and F. Kramarz. 2004. "Dissecting trade: Firms, industries, and export destinations." *American Economic Review*, 94(2): 150–154. doi:10.3386/w10344

<sup>73</sup> Zhai, F. 2008. "Armington meets Melitz: introducing firm heterogeneity in a global CGE model of trade." *Journal of Economic Integration*, 23(3): 575–604. doi:10.11130/jei.2008.23.3.575

<sup>74</sup> Balistreri, E.J., R.H. Hillberry, and T.F. Rutherford. 2011. "Structural estimation and solution of international trade models with heterogeneous firms." *Journal of International Economics*, 83(2): 95–108. doi:10.2139/ssrn.1153944

<sup>75</sup> Oyamada, K. 2014. "Behavioral characteristics of applied general equilibrium models with an Armington, Krugman, and Melitz encompassing module." 17th Annual Conference of Global Economic Analysis, Dakar, Senegal. [https://www.gtap.agecon.purdue.edu/resources/res\\_display.asp?RecordID=4459](https://www.gtap.agecon.purdue.edu/resources/res_display.asp?RecordID=4459)

<sup>76</sup> Akgul, Z., N.B. Vitoria, and T.W. Hertel. 2016. "GTAP-HET: Introducing firm heterogeneity into the GTAP model." *Journal of Global Economic Analysis*, 1(1): 111–180

<sup>77</sup> Dixon, P.B., M. Jerie, and M.T. Rimmer. 2016. "Modern trade theory for CGE modelling: the Armington, Krugman and Melitz models." *Journal of Global Economic Analysis*, 1(1): 111–180

<sup>78</sup> Balistreri, E.J., and T.F. Rutherford. 2013. "Computing general equilibrium theories of monopolistic competition and heterogeneous firms." In *Handbook of Computable General Equilibrium Modeling*, edited by P. B. Dixon and D. W. Jorgenson

<sup>79</sup> See also Itakura, Ken and Kazuhiko Oyamada. 2013. "Incorporating Firm Heterogeneity into the GTAP Model." Paper presented at the 16th Annual conference on Global Economic Analysis, Shanghai, China for technical aspects and Roson and Oyamada (2014) Roson, R., and Kazuhiko Oyamada. 2014. "Introducing



We discuss in here the introduction of the Melitz model in CGEBOX. Besides realized in GAMS (GAMS Development Corporation, 2013<sup>80</sup>) and not in GEMPACK (Harrison and Pearson, 1996<sup>81</sup>), it differs from the standard GTAP model (Hertel, 1997) in several aspects. Inter alia, its equations are all written in levels, whereas the GEMACK realization uses mostly equations depicting relative changes in linearized form. Furthermore, it aims at a modular and easily extendable framework. The implementation of the Melitz model discussed in here provides an example of such a modular extension.

The remainder of this paper is structured as follows: the next section (2) presents a brief review of the trade theories literature; section (3) describe the implementation of firm heterogeneity in the CGEBox model; section (4) presents the technical implementation of the CGEBox model with heterogeneous firms along with results of an example application; and finally section (5) concludes.

### 3.13.3 Brief Literature overview

With regard to international trade, traditional applied general equilibrium models fail in two important aspects (see critique by Keohe 1994<sup>82</sup>): they do neither allow (trade) policies to impact on aggregate productivity nor on trade along the extensive margin. In order to overcome these shortcomings, a new type of international trade models has emerged drawing on the Melitz (2003) model. It considers heterogeneous firms under monopolistic competition which can self-select into export market.

In traditional trade theories, countries specialize in production and export of those commodities in which they have comparative advantages, in the Ricardian framework based on differences in technology and in the Heckscher-Ohlin one in endowments. These models assume perfect competition and constant returns to scale, often implying that firm size is indeterminate. Due to the resulting specialization, countries either import or export a certain product. Krugman (1980), Helpman (1981)<sup>83</sup> and Ethier (1982)<sup>84</sup> established the so-called “new” trade theories drawing on variety-based models. Here, firms specialize in distinct horizontally-differentiated varieties of a product which provides an explanation why countries simultaneously export and import within the same broader industry. From the demand side, that provides a specific interpretation of the Armington assumption where quality differences reflect these distinct varieties offered by firms. A key simplification in this strand of literature is the assumption of a representative firm within each country. It is also used by Helpman and Krugman (1985)<sup>85</sup> who combine the traditional and new trade theory within an integrated

---

Melitz- Style Firm Heterogeniety in CGE models: Technical aspects and implementations, [University Ca' Foscari of Venice, Dept. of Economics Research Paper Series No. 04/WP/2014](#).

<sup>80</sup> GAMS Development Corporation. General Algebraic Modeling System (GAMS) Release 24.2.1. Washington, DC, USA, 2013

<sup>81</sup> Harrison W.J. and K.R. Pearson, 1996 "Computing Solutions for Large General Equilibrium Models Using GEMPACK", *Computational Economics*, Vol. 9 (1996), pp.83-127

<sup>82</sup> Kehoe, Timothy J., Clemente Polo, and Ferran Sancho, “An Evaluation of the Performance of an Applied General Equilibrium Model of the Spanish Economy,” Federal Reserve Bank of Minneapolis Research Department Working Papers, October 1994, (480)

<sup>83</sup> Helpman, E. (1981), “International Trade in the Presence of Product Differentiation, Economies of Scale, and Monopolistic Competition”, *Journal of International Economics* 11, 305-340

<sup>84</sup> Ethier, W.J. (1982), “National and International returns to Scale in the Modern Theory of International Trade”, *American Economic Review* 72, 225-238

<sup>85</sup> Helpman, E. and P. Krugman (1985), *Market Structure and Foreign Trade: Increasing Returns, Imperfect Competition, and the International Economy*, Cambridge, MA: MIT Press



equilibrium approach to provide an explanation for the pattern of trade where both inter-industry trade and “cross-hauling” can take place.

Increased availability of micro data on plant and firm level since the late 1980s generated empirically evidence challenging these theories of international trade. It became clear that firms are heterogeneous in terms of productivity, export behavior response to trade shocks, and other economic characteristics even within narrowly-defined industries (“one variety”). Specifically, analysis of micro datasets on firms and plants showed that i) only a small minority of firms actually engage in export (Bernard and Jensen 1995, Bernard et al. 2007) while there is considerable variation in export market participation rates across industries (Bernard et al. 2007); ii) exporters are more productive and larger than non-exporters (Bernard and Jensen 1995, Bernard et al. 2003,2007); iii) productivity dispersion exists across coexisting firms within any sector (Bartelsman and Dom 2000<sup>86</sup>) and among exporters in the number of markets (Eaton et.al 2004); iv) there are substantial reallocation effects within an industry following trade liberalization episodes where more productive firm replaces less productive ones (Foster et.al 2001<sup>87</sup>, Aw et al. 2001<sup>88</sup>) increasing aggregate productivity; and v) and endogenous changes in firm productivity via shifting market shares influences within-industry resource allocation (Bernard et al. 2006,2010) and therefore the aggregate productivity.

None of the above-mentioned observations found in micro datasets can be explained with the simplifying assumption of a representative firm within countries or industries made both in the traditional and “new” trade theories. This led to the development of international trade models with firm heterogeneity like that of Melitz (2003). He introduced firm heterogeneity into the model of Krugman (1980) which considers horizontal differentiation into varieties and increasing returns to scale.

Further, the role of the extensive margin, i.e. export of goods not exported before, in observed international trade patterns recently received increased attention. Several studies highlighted the importance of new varieties in export markets and the related welfare implications (see Romer, 1994<sup>89</sup>; Feenstra, 1994<sup>90</sup>; Broda and Weinstein, 2006<sup>91</sup>; Balistreri et al, 2010<sup>92</sup>; Keohe and Ruhl, 2013<sup>93</sup>; among others). That strand of literature is complemented by more theoretical works which discuss

---

<sup>86</sup> Bartelsman, E.J., Doms, M., 2000. Understanding productivity: lessons from longitudinal microdata. *Journal of Economic Literature* 38, 569–594

<sup>87</sup> Foster, L., Haltiwanger, J., Krizan, C., 2001. Aggregate productivity growth: lessons from the microeconomic evidence. In: Dean, E., Harper, M., Hulten, C. (Eds.), *New Directions in Productivity Analysis*. University of Chicago Press, pp. 303–363

<sup>88</sup> Aw, B.Y., Chen, X., Roberts, M.J., 2001. Firm-level evidence on productivity differentials and turnover in Taiwanese manufacturing. *Journal of Development Economics* 66, 51–86

<sup>89</sup> Romer, P.M., 1994. New goods, old theory, and the welfare costs of trade restrictions. *Journal of Development Economics* 43, 5–38

<sup>90</sup> Feenstra, R.C., 1994. New product varieties and the measurement of international prices. *The American Economic Review* 84, 157–177

<sup>91</sup> Broda, C., Weinstein, D.E., 2006. Globalization and the gains from variety. *Quarterly Journal of Economics* 121, 541–585

<sup>92</sup> Balistreri, E.J., Hillberry, R.H., Rutherford, T.F., 2010. Trade and welfare: does industrial organization matter? *Economics Letters* 109, 85–87

<sup>93</sup> Kehoe, Timothy J. and Kim J. Ruhl. 2013 . How Important Is the New Goods Margin in International Trade? (Revised version) Staff Report 324, Federal Reserve Bank of Minneapolis;

impacts of trade liberalization on the extensive margin (Hummel and Klenow, 2005; Chaney, 2008, among others). Traditional applied general-equilibrium models based on the Armington assumption cannot depict changes at the extensive margin of trade<sup>94</sup> as an important source of new trade (Kehoe, 2005). Heterogeneous firm based models like that of Melitz (2003) overcome that shortcoming by depicting changes simultaneously at the intensive and extensive margins which motivates the implementation into CGEBox discussed in here. The logic of the Melitz model is that actions that facilitate trade will raise both export variety and average productivity.

### **3.13.4 Implementation of Melitz model into CGEBox**

The standard GTAP model (Hertel 1997), developed by the Centre for Global Trade Analysis is a global, multi-regional, comparative static CGE based on neoclassical assumptions and equilibrium conditions that follow Walras' law. Policies are depicted by fixed relative price wedges. The model and variants thereof are the most widely used tools for the ex-ante analysis of economy-wide trade effects of multilateral or bilateral trade agreements. Based on a modified Cobb-Douglas Utility function, national income in each region is allocated among three types of final aggregate demand agents, namely government, private households, and savings. Each aggregate agent features its own Armington composite of domestic produce and aggregate imports for each product category, while the aggregate import composition for each product category is determined by a shared second Armington nest which also encompasses intermediate demand. Final demand expenditures on the aggregated Armington commodities reflect utility maximization, in the case of the representative private household drawing on a non-homothetic constant difference of elasticity expenditure (CDE) function; in case of the government and investment agents based on constant elasticity of substitution (CES) utility function. Markets are assumed competitive.

Production in each country and all sectors assumes a constant return to scale technology drawing on nested CES functions. In the standard GTAP model, the top level nest is a Leontief aggregate of value added and intermediate input use; the composition of the latter is based on fixed physical input coefficients. The value added nest allows for substitution between primary factors. As for the final demand agents, each sector features its own Armington nest to determine the composition of intermediate input demand for each commodity from domestic product and imports. The import composition is however identical across sectors and final demand, as mentioned above. Primary factors can either be assumed to be perfectly mobile across sectors such that the law of one price holds, or can be treated as “sluggish” based on Constant elasticity of transformation (CET) specification such that return to factors can differ between sectors.

In the standard GTAP framework, saving and capital is determined endogenously through a fictitious Global Bank. The Global Bank allocates investment across regions such that it equates the changes in the expected returns across countries. In the model, ad-valorem wedges can depict policy induces impacts on product price at the level of production, export, import and final consumption. The FOB (free on board) prices are differentiated by exporter and hence reflect bi-lateral export taxes or subsidies, adding international transport margins defines the CIF (cost, insurance and freight price) for each importer to which import taxes or subsidies are added. That allows for a rather detailed analysis of trade policy.

---

<sup>94</sup> Eaton et al. (2004) and Hillberry and Hummels (2008) defines the role of extensive margin in terms of firms serving a market while Hummels and Klenow (2005) and Broda and Weinstein (2006) identify the extensive margin in terms of the role of change in the number of products a firm trades or in the number of its trade partners (countries).

We now turn to the implementation of the Melitz (2003) model into the CGEBox version developed by Van der Mensbrugghe and Britz (2015). The model's equations are written in levels, and not as a mix of equations in levels and in linearized relative differences as found in GEMPACK based CGE implementations. That CGEBox<sup>95</sup> version allows for an exact replica of the standard GEMPACK version as developed by Hertel (1997). However, based on a flexible and modular code structure, it can also accommodate different assumptions of which we mention only some important ones. On the production side, non-diagonal make matrices, potentially combined with a CET approach are possible, while a flexible nesting approach allows more complex CES nests such as e.g. found in GTAP-AGR and GTAP-E. The Armington nests can be combined with a two-stage CET approach which allows for price dependent supply changes in with regard to the shares of domestic sales and total exports, respectively bi-lateral exports. The model can also be used in recursive-dynamic fashion. In the following, we only refer to the standard GTAP replica.

The actual implementation of the Melitz model into CGEBox draws to a large extent on the empirical method by Balistreri and Rutherford (hereafter BR) (2013) to introduce the Melitz (2003) model into an applied equilibrium model. Differences are detailed below. Further, we show how the Melitz structure compares to the Armington one. In our equation structure, sectors either are based on the standard representation, i.e. a two stage Armington structure on the demand side combined with constant return to scale industries, or follow the Melitz model with monopolistic composition and a different demand representation as detailed next.

The monopolistic competition sectors draw strictly on the framework of Melitz (2003): each firm produces one single unique variety over a continuum of varieties under conditions of monopolistic competition arising from imperfect substitution in demand for these varieties. Accordingly, the number of varieties produced in a regional industry is equal to the number of firms operating. New firms can freely enter an industry by paying a fixed entry cost which is thereafter sunk. However, before a firm enters, it is uncertain about its productivity level which becomes known once the sunk entry cost is paid. That productivity level is determined by a draw from a given productivity distribution. Once a firm knows its productivity level, it will only operate on those trade links, i.e. serving the domestic market or a specific export destination, where its profits are positive. The latter are defined as total sales revenues on a trade link minus the bilateral fixed cost of trade plus per unit variable cost times sales quantity, the per unit variable cost are assumed to be independent from the output level and the trade link. All firms face the same bilateral fixed cost on each link; however these costs differ across the trade links. Accordingly, the individual firms' decision to operate on a specific link depends on its productivity level which determines per unit variable cost. At given fixed costs of bi-lateral trade, given potential demand for an additional variety on a trade link and the price received for it, there is hence a cut-off productivity ( $\tilde{\varphi}_{rs}$ ) level beyond which profits become negative.

Those firms which draw a productivity level higher than that zero-profit cutoff productivity will operate on the trade link and those below the cutoff level will not. The firm with the productivity level equal to the cut off level is called the marginal firm and faces zero profits on that link; all other firms operating on that link make positive ones. A reduction in bi-lateral fixed cost of trade or higher

---

<sup>95</sup> Tom Rutherford provides since a long time a GAMS based implementation of a CGE called GTAPinGAMS which however differs somewhat to the standard GTAP model. The M-B implementation is coded in "Dervis KJ, de Melo J, Robinson S (1982) General equilibrium models for development policy. Cambridge University Press, New York tradition" and departs significantly from the nomenclature used by Rutherford and that used by in the GEMPACK version of the model.

demand, for instance from trade liberalization, opens hence a window of opportunity for less productive firms to serve new trade links which benefits consumer by providing more diversity in the product bundle imported from a specific origin.

There is no restriction on the number of markets that each firm can serve. The same firm is typically not active on all trade links as they differ in fixed bilateral cost of trade. Since serving the domestic markets requires lower bilateral (here country to same country) fixed cost, the more productive firms participate in export markets. Considering that intra-industry differentiation allows to depict impacts of trade policy changes on captured in conventional trade models where a trade policy leads only to re-allocation of resources between industries. In firm heterogeneity models however, a trade policy induces additionally re-allocation of resources within each industry. Firms can expand their market shares by absorbing resources of less productive ones forced to exit.

In these models, a policy that reduces the worldwide barriers to trade thus increases profits that existing exporters can earn in foreign markets and reduces the export productivity cutoff above which firms export. Input demand within the industry rises, due both to expansion by existing exporters and to new firms beginning to export. The increase in input demand bids up factor prices and reduces the profits of non-exporters. Reduction in profits in the domestic market induces some low productivity firms who were previously marginal to exit the industry. As low productivity firms exit and output and production factors are reallocated towards higher-productivity firms, average industry productivity rises.

Comparing the constant returns-to-scale sectors in the standard GTAP model to the monopolistic competition sector in Melitz reveals three main differences. First, the standard GTAP model is an aggregate industry level framework capturing the behavior of a representative firm in a perfectly competitive industry. In opposite to that, firms face fixed cost of entry in the monopolistic competition framework which leads to increasing return to scale. Second, the standard GTAP model with one representative firm in each industry cannot reflect productivity differences as depicted in the Melitz model. Third, in the standard GTAP model, consumer's utility is defined by an Armington composite of goods from different origins, while in monopolistic completion models it is defined over the Dixit-Stiglitz (1977)<sup>96</sup> composite of varieties differentiated by origin which allows for monopolistic competition between firms operating on the same trade link. The combination of these differences allows depicting the extensive margin of trade in CGE models with firm heterogeneity.

### **Algebraic representation of the firm heterogeneity into the GTAP model**

This section presents an algebraic representation of the implementation of the Melitz (2003) model as implemented in CGEBox. Note that the equations in GAMS code are documented above in section "Melitz model". The Melitz framework focuses on intra-industry differentiation where each firm produces a single unique variety. However, data at the firm level are limited and applied equilibrium models work at rather aggregate industry levels. Fortunately, Melitz offers a numerical framework build around (marginal changes of) the average firm operating on a trade link. That average firm's productivity comprises all necessary information on the distribution of productivity levels of firms active on that link. That vastly eases the model's implementation by effectively eliminating any data needs at individual firm level as detailed below. Against the background of that definition of an

---

<sup>96</sup> Dixit, A., and J. Stiglitz. 1977. "Monopolistic competition and optimum product diversity." *American Economic Review*, 67(3): 297–308. doi:10.1017/cbo9780511492273.004

average firm on each trade link, we now focus on the formulation of an empirically computable version of Melitz model and its linkages with the GTAP model.

Assume that a representative agent  $a$  (private households, government, investors, intermediate inputs by the different firms) in region  $s$  obtains utility  $U_{aais}$  from consumption of the range of differentiated varieties of product  $i$ . Considering the constant elasticity of substitution (CES) utility function as proposed by Dixit and Stiglitz (1997), the aggregate demand by each agent  $a$  for commodity  $i$  in region  $s$  ( $Q_{ais}$ ) which is equivalent to utility ( $Q_{ais} \equiv U_{aais}$ ) can be represented as

$$Q_{ais} = \left( \sum_r \int_{\omega \in \Omega_{irs}} \lambda_{airs}^{\frac{1}{\sigma_{is}}} Q_{airs}(\omega)^{\frac{\sigma_{is}-1}{\sigma_{is}}} d\omega \right)^{\frac{\sigma_{is}}{\sigma_{is}-1}} \quad (5)$$

where  $\Omega_{irs}$  represents the set of products  $i$  sourced from region  $r$  to  $s$  and  $\omega \in \Omega_{irs}$  index the varieties in the set  $\Omega_{irs}$ . In this context,  $Q_{airs}(\omega)$  represents the demand quantity of commodity  $i$  for variety  $\omega$  in region  $s$  by agent  $a$  which is sourced from region  $r$ ,  $\sigma_i$  represents the constant elasticity of substitution for each commodity, and  $\lambda_{airs}$  are preference weights (share parameters)<sup>97</sup> which reflect differences between origins not linked to diversity in varieties. Note that substitution elasticities might be differentiated by destination region  $s$ , but are uniform across agents in each region in our implementation.

The resulting CES unit expenditure function which is the dual price index on Dixit-Stiglitz composite demand in region  $s$  ( $P_{ais}$ ) is given by

$$P_{ais} = \left( \sum_s \int_{\omega \in \Omega_{irs}} \lambda_{airs} P_{airs}(\omega)^{1-\sigma_{is}} d\omega \right)^{\frac{1}{1-\sigma_{is}}} \quad (6)$$

where  $P_{airs}(\omega)$  is agent's  $a$  (purchase) price of product  $i$  for variety  $\omega$  in region  $s$  sourced from region  $r$ . Using the aggregate price index in Melitz (2003) based on his definition of the average firm and considering that varieties do not differ in their marginal utility for the first unit, one can define the price index as equivalent to the dual price defined in (6)

$$P_{ais} = \left( \sum_r \lambda_{airs} N_{irs} \widetilde{P}_{airs}^{1-\sigma_{is}} \right)^{\frac{1}{1-\sigma_{is}}} \quad (7)$$

where  $\widetilde{P}_{airs}$  denotes the agent price inclusive of export, import and consumption taxes for the average firm, and  $N_{irs}$  refers to the number of firms operating on the trade link  $r$ - $s$ . Note that consistent with Melitz (2003), there is a one to one mapping between firms and varieties such that the number of firms is equal to the number of varieties on each trade link. Comparing to (2), which is based on the individual varieties, (3) summarizes the compositional change, i.e. change in the number of varieties which go along with an update of the average price. Note again that we assume the same substitution elasticities across agents

---

<sup>97</sup> The reader should note that the share parameters are absent in the original Melitz paper. We hence allow here a differentiation between products from different origins as in the Armington model in addition to the love of variety effect.

The total ( $Q_{airs}$ ) and average per firm ( $\tilde{Q}_{airs}$ ) demand for the average variety by an agent to be shipped from  $r$  to  $s$  ( $\tilde{Q}_{airs}$ ) can be obtained by applying Shephard's Lema on the expenditure function:

$$Q_{airs} = \tilde{Q}_{airs} N_{irs} = \lambda_{airs} N_{irs} Q_{ais} \left( \frac{P_{ais}}{\bar{P}A_{airs}} \right)^{\sigma_{is}} \quad (8)$$

Which reveals the main differences to a standard Armington composite: the share parameters vary with the number of operating firms, i.e. the number of varieties comprised in the bilateral trade bundles. As the agent demand for the average firm's output in region  $r$  in each industry  $i$  in region  $s$  ( $Q_{airs}$ ) depends on the aggregate regional demand for that industry  $Q_{ais}$ , we need to determine this in equilibrium for each agent. In other word, we need to determine the demand for use of  $i$  as an intermediate input and as final demand for household consumption, government consumption, investment, and for international transport margins. In the standard GTAP model, each agent has a specific preference function which determines the demand for her Armington commodity; the government and saving sector based CES preferences while households used a CDE indirect demand function. The Armington demand for each agent and commodity is then decomposed into a domestic and import component in a first Armington nest. The second one decomposes import demand by each region by origin, independent of the agent.

The implementation of the Melitz model thus simplifies the demand structure present in the standard GTAP model by aggregating the two Armington nests into a single one, however, note that the GTAP data base so far does not differentiate in the SAM bi-lateral flows by agent. We hence used the same shares by origin to split up import demand for the different agents.

Assume a small profit maximizing firm facing the constant elasticity of demand according to (8) for its variety. Based on the assumption of the large group monopolistic competition, firms will not consider its impact on the average price index and therefore follow the usual markup rule to translate their marginal cost of production to the optimal price.

Firms in Melitz (2003) face different types of cost: sunk fixed cost of entry  $f^e$ , fixed cost of operating on a trade link  $f_{irs}$  and marginal cost  $c_{ir}$ . Let  $\varphi_{irs}$  indicate the firm's specific productivity which measures the amount of "variable composite unit" needed per unit of output  $Q_{irs}$ . Accordingly, the marginal cost per unit is the amount of "composite input" required per  $(\frac{1}{\varphi_{irs}})$  times the unit cost of the "variable composite input" ( $c_{ir}$ ) in industry  $i$  of region  $r$ . Therefore, a firm wishing to supply  $Q_{irs}$  units from region  $r$  to  $s$  employs  $(f_{irs} + \frac{Q_{irs}}{\varphi_{irs}})$  units of "variable composite input". Let,  $\tau_{irs}$  denote the fixed iceberg cost of trade which represent domestic production costs, and not the international trade margins present in GTAP. Focusing on the average firm with a productivity  $\tilde{\varphi}_{irs}$  operating on a trade link, and solving the firm's profit maximization problem, the price charged by the average firm in region  $r$  to supply region  $s$   $\tilde{P}F_{irs}$  (inclusive of domestic transport margin) is

$$\tilde{P}F_{irs} = \frac{\sigma_{is}}{\sigma_{is} - 1} \frac{\tau_{irs} c_{ir}}{\tilde{\varphi}_{irs}} \quad (9)$$

where  $\frac{\sigma_{is}}{\sigma_{is}-1}$  represents the constant markup ratio in industry  $i$  which reflects market power due to product differentiation into varieties. The linkages between the firm and agent prices are provided in Appendix 1.

The average price in (5) depends hence on the price of variable composite input  $c_{ir}$ , which is a function of the price of intermediates and primary factors. Given the assumption of constant return to scale and the way technology is presented in the standard GTAP model, the unit cost function for sector  $i$  in region  $r$   $c_{is}$  in GTAP is given by the Leontief composite of the value added bundle (CES aggregate of factors of production) and the aggregate of intermediate demand (Leontief aggregate of intermediate demands). In the CGEBox,  $m\_px$  is a macro defined as producer price which constitute per unit costs corrected for production taxes. To be consistent with our Melitz formulation, the unit cost inclusive of production tax is directly introduced in the markup equation (5). It should be emphasized that the presence of fixed cost in the Melitz model is the source of increasing returns to scale in a monopolistically competitive industry: if firms expand production, the fixed cost can be distributed over a larger outputs such that per unit cost decrease.

While observed data on quantities traded and related prices allow identifying the necessary attributes on the average firm, additional information is needed to gain information about the marginal firm, i.e. the firm which earns zero profit. Obviously, the distance in productivity between the average and marginal firm reflects properties of the underlying distribution. We rely here on a Pareto Productivity distribution which has analytical tractability (Cheney, 2008) and was shown as empirically relevant (Axtell, 2001<sup>98</sup>; Luttmer, 2002<sup>99</sup>; Gabaix, 2008<sup>100</sup>; Eaton et al., 2001<sup>101</sup>).

Let  $M_r$  indicate the number of firms choosing to incur the fixed entry cost, i.e. total industry size, each individual firm receives its productivity  $\varphi$  draws from a Pareto distribution with Probability Density Function (PDF)

$$g(\varphi) = a \left( \frac{b^a}{\varphi^{a+1}} \right) = \frac{a}{\varphi} \left( \frac{b}{\varphi} \right)^a \quad (10)$$

and Cumulative Distribution Function (CDF)

$$G(\varphi) = 1 - \left( \frac{b}{\varphi} \right)^a \quad (11)$$

where  $b$  is the minimum productivity and  $a$  is a shape parameter. Lower values of the shape parameter imply higher productivity dispersion among firms. As discussed in Melitz (2003),  $a > \sigma_i - 1$  should be applied in order to ensure a finite average productivity level in the industry.

---

<sup>98</sup> Axtell, Robert L. 2001. “Zipf Distribution of U. S. Firm Sizes.” *Science*, 293(5536): 1818–20. Crozet, Matthieu, and Pamina Koenig. 2007. “Structural Gravity Equation with Extensive and Intensive Margins.” Unpublished

<sup>99</sup> Luttmer, Erzo G. J. 2007. “Selection, Growth, and the Size Distribution of Firms.” *Quarterly Journal of Economics*, 122(3): 1103–44

<sup>100</sup> Gabaix, Xavier. 2008. “Power Laws.” In *The New Palgrave Dictionary of Economics*. 2nd edition, ed. Steven N. Durlauf and Lawrence E. Blume. New York: Palgrave Macmillan

<sup>101</sup> Eaton, J., S. Kortum, and F.Kramarz. 2001. “An anatomy of international trade: evidence from French firms.” *Econometrica*, 79(5): 1453–1498. doi:10.3386/w14610



On each bilateral trade link, the given the fixed bilateral trade cost, variable costs and demand define jointly a certain cut off productivity level ( $\varphi_{rs}^*$ ) at which firms will receive zero profit. A firm which has drawn the productivity equal to that threshold level ( $\varphi_{rs} = \varphi_{rs}^*$ ) will hence face zero profits and act as the marginal firm from region  $r$  supplying  $s$ . Those firm whose productivity is above the threshold level ( $\varphi_{rs} > \varphi_{rs}^*$ ) will receive a positive profit and will operate on the  $r - s$  link and those firm whose productivity is below the threshold level ( $\varphi_{rs} < \varphi_{rs}^*$ ) will not operate on the  $r - s$  link. Focusing on the fixed operating cost  $f_{irs}$  in composite input units, the marginal firm on  $r$ - $s$  link receives zero profit at

$$c_{ir}f_{irs} = \frac{r(\varphi_{irs}^*)}{\sigma_i} \quad (12)$$

where  $r(\varphi_{irs}^*) = p(\varphi_{irs}^*)q(\varphi_{irs}^*)$  denotes the revenue of marginal firm at the productivity equal to the cut off level ( $\varphi_{irs} = \varphi_{irs}^*$ ).

The zero cut off productivity level in each bilateral market  $\varphi_{irs}^*$  can be obtained by solving (12). However, it is numerically easier to define this condition in terms of the average rather than the marginal firm. To do this, we define the productivity and revenue of the average firm relative to that of the marginal firm.

The average productivity in a trade link is determined by the productivity level of the operating firms on that link which by definition are at or above the cutoff productivity level. Following Melitz (2003), that average productivity is defined as the CES aggregation of productivities of all firms operating on a given trade link

$$\tilde{\varphi}_{irs} = \left[ \frac{1}{1 - G(\varphi_{irs}^*)} \int_{\varphi_{irs}^*}^{\infty} \varphi_{irs}^{\sigma_{is}-1} g(\varphi) d\varphi \right]^{\frac{1}{1-\sigma_{is}}} \quad (13)$$

If these productivities are Pareto distributed, we can write<sup>102</sup>

$$\tilde{\varphi}_{irs} = \left[ \frac{a}{(a + 1 - \sigma_{is})} \right]^{\frac{1}{1-\sigma_{is}}} * \varphi_{irs}^* \quad (14)$$

Eq. (10) provides the relationship between the productivities of the average and marginal firm (for further details see Allen and Arkolakis (2016)<sup>103</sup>.

<sup>102</sup> One could use industry specific shape parameter ( $a_i$ ) given the availability of data at sectoral level. In this study we assume that all firms entering in different industries draw their productivity from the Pareto distribution function with same characteristics (i.e. same scale and shape parameter).

<sup>103</sup> Allen, T., and C. Arkolakis (2016): “Elements of Advanced International Trade,” Unpublished: available at: <http://www.econ.yale.edu/~ka265/teaching/GradTrade/notes/ClassNotes.pdf>.

Using optimal firm pricing according to (9) and given the input technology, the ratio of revenues of the firms with marginal productivity  $r_{irs}(\varphi^*)$  in relation to the revenue of the firm with the average productivity  $r_{irs}(\tilde{\varphi})$  is defined as

$$\frac{r_{irs}(\varphi^*)}{r_{irs}(\tilde{\varphi})} = \left(\frac{\varphi^*}{\tilde{\varphi}}\right)^{\sigma_i} \quad (15)$$

Solving (14) for  $\frac{\varphi^*}{\tilde{\varphi}}$ , substituting it into (15), and then solving the resulting equation for  $r_{irs}(\varphi^*)$  and replacing its value in (12), defines a relation between the bilateral fixed cost at current composite input price (the left hand side of (12) below), the average firms revenue ( $\tilde{P}F_{irs} \tilde{Q}_{irs}$ ), the shape parameter of the Pareto distribution of the productivities and the substitution elasticity of demand:

$$c_{ir}f_{irs} = \frac{(a + 1 - \sigma_{is})}{a\sigma_{is}} \tilde{P}F_{irs} \tilde{Q}_{irs} \quad (16)$$

Note that average firm's sale in region  $r$  in each industry  $i$  to region  $s$  ( $\tilde{Q}_{irs}$ ) at the equilibrium is composed of the demand for use of  $i$  by different agents<sup>104</sup>.

The optimal pricing in the markup equation (5) requires information on the average productivity on each bilateral trade link. In Melitz (2003), the probability that a firm will operate is  $1 - G(\varphi_{irs}^*)$  which is equal to the fraction of operating firms over total number of firms choosing to draw their productivity ( $\frac{N_{ir}}{M_{ir}}$ ). Using the Pareto cumulative distribution function (11) and inverting it we have

$$\varphi_{irs}^* = \frac{b}{\left(\frac{N_{irs}}{M_{ir}}\right)^{\frac{1}{a}}} \quad (17)$$

substituting (17) into (14) results in

$$\tilde{\varphi}_{irs} = b \left[ \frac{a}{(a + 1 - \sigma_{is})} \right]^{\frac{1}{1 - \sigma_{is}}} * \left(\frac{M_{ir}}{N_{irs}}\right)^{-\frac{1}{a}} \quad (18)$$

Next, the number of firms selecting to enter the market  $M_{ir}$  is determined. Based on the free entry condition, the last firm which enters has expected profits over its life time which just offset the sunk cost of entry. Industry entry of a firm requires a one-time payment of  $f^{ie}$ . An entered firm faces a probability of  $\delta$  in each future period to suffer a shock which forces its exit. Therefore,  $\delta M_{ir}$  firms are lost in each period. Based on Melitz (2003), in a stationary equilibrium, the number of aggregate variables must remain constant over time, including industry size. This requires that the number of new entrants in every period is equal to the number of firms lost  $\delta M_{ir}$ . Therefore, total entry cost is equal to  $c_{ir} \delta M_{ir} f^{ie}$ . Each firm faces the same expected share on that cost, i.e.  $c_{ir} \delta f^{ie}$  if risk neutral behavior and no time discounting is assumed. The firm's expected share on entry costs must be equal to the flow of expected profit on the condition that firm will operate.

---

<sup>104</sup>  $\tilde{Q}_{irs} = \sum_a \tilde{Q}_{airs}$

$$\tilde{\pi}_{irs} = \frac{\widetilde{PF}_{irs} \tilde{Q}_{irs}}{\sigma_{is}} - c_{ir} f_{irs} \quad (19)$$

The probability that a firm will operate on the r-s trade link is given by the ratio of  $\frac{M_{ir}}{N_{ir}}$ <sup>105</sup>. Thus, the free entry condition ensures that the expected industry profits, i.e. the profits summed up over all potential bilateral trade links, is equal to the annualized flow of the fixed costs of entry

$$c_{ir} \delta f^{ie} M_r = \sum_s N_{irs} \widetilde{PF}_{irs} \tilde{Q}_{irs} \frac{\sigma_{is} - 1}{a\sigma_{is}} \quad (20)$$

where zero profit condition (16) is used to replace the fixed operating cost  $c_{ir} f_{irs}$ .

With the number of entered firm established in (20), we now turn to total composite input demand of the industry Y which consists of three components: sunk entry costs of all entrants ( $\delta M_{ir} f^{ie}$ ), operating fixed cost ( $\sum_s N_{irs} f_{irs}$ ) on each trade link and variable costs ( $\sum_s N_{irs} \frac{\tau_{irs} \tilde{Q}_{irs}}{\tilde{\varphi}_{irs}}$ ). Therefore, composite input demand is defined as

$$Y_{ir} = \delta M_{ir} f^{ie} + \sum_s N_{irs} (f_{irs} + \frac{\tau_{irs} \tilde{Q}_{irs}}{\tilde{\varphi}_{irs}}) \quad (21)$$

This equation provides the link to the equations in the GTAP model describing the technology and related costs. Table 17 summarizes full set of Melitz equations which are introduced into the GTAP model.

Table 17: Equilibrium conditions in the Heterogenous Firm model

Equation	Equilibrium condition	Associated variable
(7)	Sectoral Aggregation	$P_{is}$ : Sectoral price index
(8)	Firm-level demand	$\widetilde{PF}_{irs}$ : Average firm price
(9)	Firm-level Pricing	$\tilde{Q}_{irs}$ : Average firm quantity
(16)	Zero cut off condition	$M_{ir}$ : Number of operating firms
(18)	CES Weighted Average productivity	$N_{ir}$ : Average firm productivity
(20)	Free entry condition	$N_{irs}$ : Number of entered firm
(21)	Factor market clearing condition	$c_{ir}$ : Sectoral composite input price

<sup>105</sup> The probability that the firm will operate is  $1 - G(\varphi_{irs}^*) = \frac{N_{ir}}{M_{ir}}$

Variables through which Melitz model is linked to the GTAP
$Q_{ais}$ : Sectoral Demand
$c_{ir}$ : Price of composite input
$Y_{ir}$ : Sectoral composite input demand

### Calibration of the model

In order to apply the above equation structure, the different parameters must be chosen such as to recover an observed benchmark. That benchmark consists of the global SAM provided by the GTAP data base<sup>106</sup> against which the remaining equations of the GTAP model are calibrated as well. It comprises values on domestic sales and on bilateral international trade expressed in USD million for each sector and region which are key data in the context of the Melitz module discussed above. Global detailed SAMs comprise many small entries both in relative and absolute terms which can affect the numerical stability during solution of a CGE. We therefore recommend filtering out in a systematic way small transactions in relative terms from the database when using the model with many sectors and regions, following by rebalancing the global SAM.

We focus here mainly on the calibration of the firm heterogeneity module while the calibration of Armington sectors are similar to the standard GTAP model and are not further discussed. In the following, a superscript “0” denotes a benchmark value, for instance,  $Q_{is}^0$  denotes the benchmark value of demand for commodity  $i$  in region  $s$ . In order to line up the variables in the Melitz module with the SAM, the following identities must hold:

$$c_{ir}^0 Y_{ir}^0 \equiv vom_{ir} \quad (22)$$

$$p_{is}^0 Q_{ais}^0 \equiv xafm_{ais} \quad (23)$$

where  $vom_{ir}$ , represent the production value of commodity  $i$  in region  $r$  at producer tax inclusive prices and  $xafm_{ais}$  represent the value of each agent's demand for commodity  $i$  in region  $s$ , consumer tax inclusive. The first identity indicates that the cost of input supply must equal the value of output, and the second identity ensures that demand for all goods and factors is equal to supply at the benchmark. Accordingly, using the conventional choice of unity prices, i.e.  $p_{irs}^0 = 1$ , and  $c_{irs}^0 \equiv 1$ , total quantity demanded  $Q_{is}^0$  and total input supplied  $Y_{ir}^0$  is locked down.

It should be noted here again that the bilateral import demand in the SAM is aggregated over agents. The RHS entries in equation (19) above are hence constructed by splitting up that total by each agent's share on total imports.

Given the agents' demand for each commodity  $Q_{ais}^0$  and using the definition of the firm average in Melitz, we have

---

<sup>106</sup> We currently use the GTAP9 Data Base which carries a snapshot of the 2011 world economy, covering 140 regions (aggregate of 226 countries) and 57 sectors. But all GTAP Data Bases share the basic structure such that the code can also be used with other releases.

$$Q_{ais}^0 = \sum_r Q_{airs}^0 = \sum_r \tilde{Q}_{airs}^0 N_{irs}^0 \quad (24)$$

We now briefly compare the demand side based on the Armington assumption with the Melitz model to demonstrate that the choice of  $N_{irs}^0$  does not matter for the model's simulation behavior. Under the Armington assumption, a CES utility function is used to differentiate between origins. That implies that the (average) quality on the trade links can differ along with the price. As the resulting demand function is homothetic, expenditure shares are independent of the income level and solely depend on relative prices for these qualities. Conveniently, calibration is performed at given substitution elasticities by choosing share parameters such that given expenditure shares are recovered at given relative prices.

The Ditz-Stieglitz price index used in the Melitz (2003) is also based on a CES utility function; however, here we have a continuum of varieties. As share parameters are absent in the original model, the marginal utility of the first unit for each variety is the same. Each firm is assumed to produce its own variety and multiple firms are allowed to operate on a trade link. Thus, more firms imply more varieties and a higher utility per unit of traded output on that link. Total demand on a link is hence defined as the product of average output per operating firm and the number of firms operating on that link. Recovering expenditures shares at given prices for each trade link can hence be based either by deriving the number of firms on a link at given average firm output on that link or by pre-selecting the number of firms for each link and deriving average outputs. Share parameters are not needed; however, as a consequence, the resulting price index cannot be controlled. As only relative prices matter, that only affects readability and not simulation behavior.

In order to improve readability and provide a combined interpretation of the two models, we introduce share parameters in the CES-demand function used in the Melitz model. That allows a convenient interpretation of the extension introduced by the Melitz model: changes in the number of firms lead to preference shifts in the Armington model as they update the share parameters. The resulting price index  $P_{ais}$  and matching demands  $\tilde{Q}_{airs}$  for product  $i$  in region  $s$  from origins  $r$  by agent  $a$  can be depicted as

$$P_{ais}^0 = \left( \sum_r \lambda_{airs} \frac{N_{irs}}{N_{irs}^0} \tilde{P}A_{airs}^0 \right)^{\frac{1}{1-\sigma_{is}}} \quad (25)$$

$$Q_{airs}^0 = \lambda_{airs} \frac{N_{irs}}{N_{irs}^0} Q_{ais}^0 \left( \frac{P_{ais}^0}{\tilde{P}A_{airs}^0} \right)^{\sigma_{is}} \quad (26)$$

where  $N_{irs}$  depicts the number of operating firms on a trade link. Note that, as  $N_{irs}$  and  $\lambda_{airs}$  (the share parameter) are identical in the calibration point, we scale without any impact on the simulation behavior the lambdas  $\lambda_{irs}$  such that the price index  $P_{is} \equiv 1$  (or something else) at the benchmark. That allows to recover any given quantity index  $Q_{ais}$  and matching  $P_{ais}$ . Equations 25 and 26 show the link between the Armington and Melitz models. A given Armington model can hence be simply extended by adding changes in the number of operating firms. With the choice of  $\tilde{P}A_{airs}^0 = 1$ ,  $P_{ais}^0 = 1$ , since we now have the observed value of  $Q_{airs}^0$ ,  $Q_{ais}^0$ , and estimated value  $\sigma_{is}$ <sup>107</sup>,

<sup>107</sup> In GTAP the first level Armington demand for each agent  $XA_{i,h}$ ,  $XA_{i,gov}$ ,  $XA_{i,inv}$  is a CES composite of domestic and aggregate imported good with the substitution elasticity  $\sigma_m$ ; and the second level Armington

the value of  $\lambda_{airs}$  is calibrated in a way that we do not need the information on the benchmark number of firms. Accordingly, the values of  $\lambda_{airs}$  are recovered by inverting the demand functions.

$$\lambda_{airs} = \frac{Q_{airs}^0}{Q_{ais}^0} \left( \frac{\widetilde{PA}_{airs}^0}{P_{ais}^0} \right)^{\sigma_{is}} \quad (27)$$

Let  $\widetilde{PF}_{irs}$  denote the average firm's offer price on a specific trade link, from which  $\widetilde{PF}_{airs}$  is derived by considering export taxes (to arrive at cif), international transport margins (to arrive at fob), and import and consumption taxes.

The composite input demand linked to bilateral fixed cost  $f_{irs}$  is derived from the zero-profit cut off condition, where the values of  $c_{ir}^0$  and  $\widetilde{PF}_{irs}^0$  are set to unity at the benchmark,  $\widetilde{Q}_{irs}$  is estimated from (23), and the value of shape parameter  $a$  is taken from literature<sup>108</sup>:

$$f_{irs} = \frac{\widetilde{PF}_{irs}^0 \widetilde{Q}_{irs}^0 (a + 1 - \sigma_i)}{c_{ir}^0 a \sigma_i} \quad (28)$$

Please note that  $Q_{irs}^0 = \widetilde{Q}_{irs}^0 N_{irs}^0$  implies that changing the number of firms at the benchmark updates  $f_{irs}$ , but not the total industry cost linked to bi-lateral fixed cost.

Similarly, the free entry condition allows deriving the composite input demand of sunk entry cost  $c_{ir} \delta f^{ie}$  at given price and quantity of the average firm:

$$c_{ir} \delta f^{ie} = \sum_s \frac{N_{irs}^0}{M_r^0} \widetilde{PF}_{irs}^0 \widetilde{Q}_{irs}^0 \frac{\sigma_{is} - 1}{a \sigma_{is}} = \sum_s \frac{1}{M_r^0} \widetilde{PF}_{irs}^0 Q_{irs}^0 \frac{\sigma_{is} - 1}{a \sigma_{is}} \quad (29)$$

demand (aggregated import) is a CE's composite of import demand by each region of origin with the substitution elasticity of  $\sigma_w$ . However, in Melitz structure represented consume has Dixit–Stiglitz preferences over the varieties including domestically produced commodities and imported commodities by source of origin with the substitution elasticity of  $\sigma_i$ . To be consistent with Dixit–Stiglitz framework,  $\sigma_i$  is derived based on the weighted average of the substitution elasticities of first and second level Armington nests where the weights given to  $\sigma_m$  and  $\sigma_w$  are domestic and import purchases at agent price, respectively. We also insured that derived substitution elasticity is smaller than  $a + 1$  to ensure a finite average productivity level in the industry.

<sup>108</sup> Estimates of the value shape parameter vary and are conditional on a choice of elasticity of substitution. The choices of shape parameter and substitution elasticity are important as these key parameters have significant implications (for example on welfare). The importance of these variables is well discussed in Akgul et al. (2015, 2016) and Dixit (2016). Bernard et al. (2007) choose a shape parameter equal to 3.4, and estimates of Eaton et al. (2004) show  $a = 4.2$ , while Balistreri et al. (2011) find shape parameter of 5.17, 4.58 and 3.92 depending on different trade cost-distance elasticities but under the maintained assumption that  $\sigma = 3.8$ . Akgul, Z., N.B. Villoria, and T.W.Hertel.2016. “Theoretically-Consistent Parameterization of a Multi-sector Global Model with Heterogeneous Firms” Forthcoming calibrated shape parameter of 2.89 for manufacturing sector consistent with a shape parameter obtained Spearot, A. 2016. “Unpacking the long run effects of tariff shocks: New structural implications from firm heterogeneity models.” AEJ Microeconomics, 8(2):128–67. doi:10.1257/mic.20140015. In this study we used the weighted average of the substitution elasticities of first and second level Armington nests which is not necessary consistent with the shape parameter (4.6) which is taken from Balistreri et al.(2011). Once decided which sectors are treated as sectors with heterogeneous firms, the theoretically consistent sectoral elasticity of substitution and sectoral shape parameter should be obtained. The minimum productivity parameter is chosen as  $b = 0.2$  following Balistreri et al. (2011) from Bernard et al. (2007).

Note again that the given values on the trade links  $Q_{irs}^0$  determine together with the chosen industry size the sunk entry costs such that, as above, either the number of operating firms or the average quantities can be chosen without affecting the calibration. Furthermore, the condition reveals that the choice of total industry size  $M_r^0$  does not matter either, as the sunk entry costs will adjust proportionally such that the industry's total annualized costs of industry entry solely depend on the shape parameter of the Pareto distribution, the substitution elasticity and the given SAM values.

Next, given a minimum productivity parameter of the Pareto distribution  $b$ , the average firm productivity on trade link can be initialized as:

$$\tilde{\varphi}_{irs} = b \left[ \frac{a}{(a + 1 - \sigma_{is})} \right]^{\frac{1}{1 - \sigma_{is}}} * \left( \frac{M_r^0}{N_{irs}^0} \right)^{-\frac{1}{a}} \quad (30)$$

Having determined the value of average productivity on each r-s link, and setting  $c_i \equiv 1$ ,  $\widetilde{PF}_{irs}^0 \equiv 1$ , the domestic input demand related to the transport margin cost can be computed by inverting the mark up equation (9) as follows:

$$\tau_{irs} = \frac{\sigma_{is}-1}{\sigma_{is}} * \tilde{\varphi}_{irs}^0 \quad (31)$$

Here, it is less obvious why  $M_r^0$  and  $N_{irs}^0$  can be freely chosen without affecting simulation behavior.  $\tilde{\varphi}_{irs}$  and  $\tau_{irs}$  enter the total industry cost and the price markup equations. Total industry costs on a trade link r-s are given as:  $N_{irs} \left( f_{irs} + \frac{\tau_{irs} \tilde{Q}_{irs}}{\tilde{\varphi}_{irs}} \right)$  which shows that changes in  $\tilde{\varphi}_{irs}^0$  and  $\tau_{irs}$  resulting from different benchmark values of  $N_{ir}$  and  $M_r$  do not matter. The same holds for the price markup equation:  $\widetilde{PF}_{irs} = \frac{\sigma_{is}}{\sigma_{is}-1} \frac{\tau_{irs} c_{ir}}{\tilde{\varphi}_{irs}}$ .

Accordingly, sensitivity analysis with different values for  $M_r^0$  and  $N_{irs}^0$  showed no differences in simulated results. However, the choice might affect numerical stability by affecting the overall scaling of the model. In order to improve readability, we have generally chosen  $N_{irs}^0 \equiv 1$  which gives average firm outputs on each trade link equal to given bi-lateral respectively domestic sales values at benchmark prices.

### Technology nesting

Similar the GTAP-HET model, we apply a different nesting for variable costs of trade and the fixed costs related to industry entry and operating on a link. The variable costs maintain whatever nesting is chosen originally by the analyst. The fix cost only use fixed value added, applying the same substitution elasticity as used for the value added nest in the original model. Note that this is a first implementation which should be improved as it implies for industries using land or natural resources that they also become part of the fixed costs. Technically, the flexible nesting structure by CGEBox is applied by defining a new technology nest termed “fcost”. At the same time, the implementation of the nesting was extended to allow for the case that several nests demand the same intermediate input or factor. Alternatively, the model can be simplified to use the same composite input for variable and fixed costs. The implementation allows combining the default Melitz model with its differentiation



between variable and fixed costs composites with other nesting structures on the production side such as the implementation of GTAE-E or GTAP-AGR realized as modules of CGEBox.

### 3.13.5 Numerical stability and domestic industry demand for its own output

Test runs with the model in different configurations with regard to sectoral and regional detail as well as differently deep multi-lateral trade liberalization shocks have consistently shown non-stable behavior resulting in infeasibilities in cases where an industry's cost share of domestic intermediate input is relatively high, say around 30%. Under these conditions, the model easily ends up in a vicious circle where a cost increase on the domestic link amplifies itself such that corner solutions with zero industry sizes provoke infeasibilities. Changing the parameterization seems to help only in some selected cases.

A firm should not apply mark-up pricing for own produced and used intermediates if that decreases its own competitiveness, i.e. provokes real costs and is not a tax evasion tactic. However, the average firm model does not differentiate domestic intermediate demand inside a sector between a firm's demand for its own produce and demand by other firms. In order to allow for a numerical stable implementation, we therefore allow excluding domestic intermediate demand for the diagonal I/O element from the love of variety effect, i.e. applying Armington preferences. Accordingly, there is also no markup pricing involved if that option is chosen. That has proven to effectively prevent the down-spiraling solution behavior in most cases. However, it can still happen if a bundle of sectors shows quite high costs share of domestic demand for the same bundle of sectors. In that case, the defense by arguing over the single firm's behavior is not longer valid.

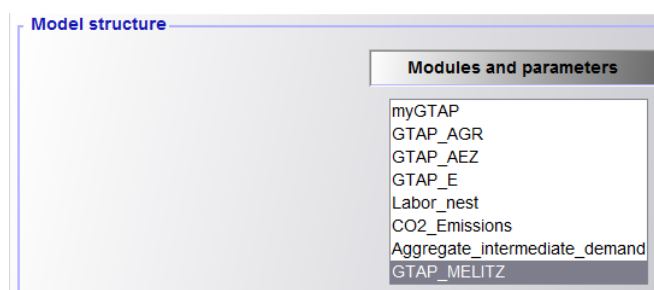
Potential improvements in that regard could encompass a split up of the diagonal domestic intermediate demand into a share subject to the love of variety assumption and a remainder treated as competitive. But before introducing further complexity into the framework, more testing is required.

### 3.13.6 Technical implementation and an example application

#### Technical implementation

The necessary GAMS code for the calibration of the model can be found in "gams\GTAPMelitz\GTAPMelitz\_cal.gms". Besides the calibration steps discussed above, it is important to mention that certain flags are deleted to ensure that equations in the standard model are inactive for Melitz commodities.

The user can switch the module one on the Graphical User Interface under "Model Structure":



which generates a new tab with the following input possibilities.

Melitz model Closures Output options Algorithm GAMS

Melitz commodities

pdr-c	wht-c	gro-c	v_f-c	osd-c	c_b-c	pfb-c	ocr-c
cti-c	oap-c	rmk-c	wol-c	frs-c	fsh-c	coa-c	oil-c
gas-c	omn-c	cmt-c	omt-c	vol-c	mil-c	per-c	sgr-c
ofd-c	b_t-c	tex-c	wap-c	lea-c	lum-c	ppp-c	p_c-c
crp-c	nmm-c	i_s-c	nfm-c	fmp-c	mvh-c	otn-c	ele-c
ome-c	omf-c	ely-c	gdt-c	wtr-c	cns-c	trd-c	otp-c
wtp-c	atp-c	cmn-c	ofi-c	isr-c	obs-c	ros-c	osg-c
dwe-c							

Only aggregated demand, no heterogenous firms ☐

Fix cost nest ☐

Pareto distribution lower limit 0.20 Shape parameter 4.60

Max cost share for HET own domestic interm. demand 0.10

Under Melitz commodities, the commodities / sectors can be chosen the parameter of the Pareto-distribution of the productivities defined.

The checkbox “Only aggregated demand, not heterogenous firms” changes the demand structure of the standard GTAP such that only one Armington nest shared by all agents is present. The substitution elasticity is defined as in the case when the full Melitz model is used. The remaining equations of the Melitz model are absent and replaced by the equations of the standard GTAP model. That allows disentangling impacts of the full Melitz model from the structural changes on the demand side.

The checkbox “Fix cost nest” introduces a differentiation between variable and fix costs input composites where the latter only comprises primary factors, at least as long as there is some minimal primary factor cost share left in the variable input composite. The spinner termed “Max cost share of HET domestic interm. demand” allows switching off love of variety for domestic intermediate demand by same industry depending on the costs share. Only sector-regional combinations with a share above the threshold will be excluded. Setting the threshold to unity will hence leave the love of variety effect switch on for all domestic intermediate demand, while zero will switch it off on all cases.

Detail on simulated value of variables found in the Melitz module can be found in two tables under the “Trade group” as shown below:



## An example application

We use a 50% reduction of all imports tariff and export subsidies globally with a 10x10 aggregation of the GTAP9 data base as an example application<sup>109</sup>, focusing on welfare changes and highlighting the newly available information from the model. We compare the standard GTAP model against two variants, one where only the demand structure of the Melitz model is used and a second one with its full implementation.

<sup>109</sup> Appendix 2 provides the sectoral and regional aggregation of GTAP sectors into the new mapping.

Table 18 below reports the Money metric for the full Melitz model (Tariffs\_m), the model where only the demand side of the Melitz model is used (Tariffs\_d) and the Standard GTAP model (Tariffs). First, it can be seen that aggregate welfare is almost not affected by choosing the more simple demand side with only one Armington nest shared by all agents. However, results for individual regions show some sensitivity.

As expected, adding the full Melitz model increases welfare, here, it more than quadruples the global welfare impact of multi-lateral trade liberalization. Interestingly, that impact is not uniform across regions: whereas North America suffers a small welfare loss in the standard configuration, it gains under the Melitz model. However, the welfare in Middle East and North Africa is the same under two structures. Generally the results are in line with finding of Balistreri et al. (2011) which also found welfare increases around factor four.

Note that, for our simulation exercise, we compare the welfare impact of the policy shock under different structures where more or less the same value of Armington elasticity is assigned in each structure. Specifically, our calibration code restricts the substitution elasticity to an interval of  $\pm 1/2$  around the share parameter which for many sectors will yield more or less the same consumption quantity weighted elasticity as found in the two-level Armington system of GTAP.

Dixon et al. (2016) argues that in order to compare the welfare impacts of a policy shock under Armington structure and Melitz structure one should assign the substitution elasticity ( $\sigma$ ) in the Armington framework such that it simulated trade flows comparable to a Melitz framework. In the two commodity and two country model of Dixon et al. (2016); a substitution elasticity of 8.45 yielded trade flows similar to a Melitz model with a substitution elasticity of 3.8 when tariff in one of the countries increases by 7.18 percent. Dixon et al. (2016) obtained this equivalent Armington value of substitution using a trial and error approach. The substitution elasticity in an Armington model which yields similar results depends not only on the parameterization of the Melitz model, but also on the type and magnitude of the shock and structure of the model as a whole. Using trial and error to an “equivalent” substitution elasticity to replicate the trade pattern after a given shock can only be used if the model is rather small. However, with a medium sized model, we found that once  $\sigma$  is adjusted in Armington framework to replicate one of the trade flows, the error in other bilateral trade flows might even become larger. Such finding is in line with that of Balistreri et al. (2011). Akgul (2015, 2016), who introduced a theoretically consistent framework from which one can calibrate the sector specific Armington elasticity given the obtained sector specific value of the shape parameter and econometrically estimated results. Given data limitations, we adjust the given Armington elasticities around a uniform Pareto shape parameter across regions and sectors. This assumption remains open to critic and the choices of theoretically consistent substitution elasticities across regions and sectors is

**Table 18:** Money metric in comparison of Melitz and Standard GTAP model

Money metric decomposition [0]			
Sectors and institutions		Total	Years shock
Total [Bio \$]		Total	
Tariffs		Tariffs_d	Tariffs_m
World	44.88	50.89	177.64
Australia, New Zealand	1.63	1.77	4.23
East Asia	30.14	31.72	94.15
Southeast Asia	2.93	4.13	13.64
South Asia	3.08	1.34	10.19
North America	-3.76	-0.62	10.96
Latin America	0.80	0.23	5.01
European Union 25	4.50	9.21	23.92
Middle East and North Africa	0.53	-0.02	0.53
Sub-Saharan Africa	0.68	0.35	1.41
Rest of World	4.34	2.77	13.60

The next Table 19 shows the simulated overall price index and seems to indicate, as expected, that the Melitz model tends to amplify the impacts found under Armington model.

**Table 19:** Aggregate price index of the Armington agents

Income, compare across regions [0]				
Total or household groups		Sectors and institutions	Years shock	Percentage diff. to Scenarios no Shock
Total		Price index [Index]		
Tariffs		Tariffs_d	Tariffs_m	
World	0.99 -0.62%	0.99 -0.89%	0.99 -1.01%	
Australia, New Zealand	1.00 0.10%	1.00 0.12%	1.00 0.02%	
East Asia	1.01 0.68%	1.01 0.54%	1.01 0.61%	
Southeast Asia	1.00 -0.01%	1.00 0.03%	1.00 0.18%	
South Asia	0.99 -1.36%	0.98 -2.17%	0.98 -2.27%	
North America	1.00 -0.48%	1.00 -0.38%	0.99 -0.67%	
Latin America	0.99 -0.79%	0.99 -0.86%	0.99 -1.12%	
European Union 25	1.00 -0.45%	1.00 -0.28%	0.99 -0.50%	
Middle East and North Africa	0.98 -1.80%	0.98 -2.08%	0.98 -2.35%	
Sub-Saharan Africa	0.98 -1.74%	0.98 -2.07%	0.98 -2.41%	
Rest of World	0.99 -0.93%	0.99 -1.07%	0.99 -1.29%	

The impacts on factor income corrected for changes in the overall price index are less clear with more diversified changes across the regions. In an environment with multiple factors, effects on factor prices differ which seems significant in our application.

**Table 20:** Factor income, corrected with price index aggregate Armington agent

Income, compare across regions [0]				
Total or household groups		Sectors and institutions		Percentage diff. to
Total		Factor income, normalized with price index [Bio \$]		Scenarios
		Years shock		no Shock
		Tariffs	Tariffs_d	Tariffs_m
World	39756.41 0.69%	39861.78 0.95%	39922.23 1.11%	
Australia, New Zealand	759.26 0.55%	759.35 0.56%	760.90 0.76%	
East Asia	7226.88 0.69%	7223.42 0.64%	7268.05 1.26%	
Southeast Asia	1081.11 1.23%	1081.91 1.30%	1089.57 2.02%	
South Asia	1242.61 1.52%	1239.76 1.29%	1242.17 1.48%	
North America	12335.02 0.10%	12337.88 0.12%	12345.46 0.18%	
Latin America	2051.05 0.63%	2050.54 0.60%	2054.21 0.78%	
European Union 25	10260.98 0.11%	10266.71 0.17%	10276.61 0.27%	
Middle East and North Africa	448.36 1.67%	447.76 1.53%	447.79 1.54%	
Sub-Saharan Africa	682.86 1.39%	682.22 1.30%	682.14 1.29%	
Rest of World	3561.89 0.76%	3559.80 0.70%	3566.61 0.89%	

Table 21 below summarizes main variables found in the Melitz module. As indicated in the table, as expected, the total number of firms entered (number of domestic verities) decrease as result of liberalization while the number of operating firm indicating the sum of verities consumed increases.

**Table 21:** Example summary information from Melitz model

Melitz, overview [0]				
Exporter		Products		Percentage diff. to
Australia, New Zealand		Heavy Manufacturing		Scenarios
		Years shock		no Shock
		Tariffs_m	no Shock	
All firms entered	0.98 -2.12%	1.00		
All firms operating	0.05 0.65%	0.05		
Variable cost price	1.00 0.09%	1.00		
Fix cost price	1.00 0.09%	1.00		
General fixed cost	27.51 -2.03%	28.08		
Bi-lateral fixed cost	4.75 -0.57%	4.77		
Variable cost	126.57 -2.03%	129.19		
General fixed cost per unit	0.13 0.37%	0.13		
Bi-lateral fixed cost per unit	0.02 1.86%	0.02		
Variable cost per unit	0.58 0.37%	0.58		

The last Table 22 below details information on each trade link. Note first the impact on the domestic sales: as the firm's price in the domestic market drops, the number of firms operating on the domestic link is reduced and average productivity and output per firm increases. In opposite to that, the firms prices of selling to "South Asia" increase by almost 1.3 % which increases the number of operating firms on that link by 1.12% and let average quantities and productivity drop by about -2.87%. Still, as a result of these effects, export increase by 8.7%.

**Table 22: Example information from Melitz model by trade link**

Melitz, trade matrix [0]											
Exporter		Products						Years		Percentage diff. to	
Australia, New Zealand		Heavy Manufacturing						shock		Scenarios	
										no Shock	
	Domestic sales	Australia, New Zealand	East Asia	Southeast Asia	South Asia	North America	Latin America	European Union 25	Middle East and North Africa	Sub-Saharan Africa	Rest of World
Firm price	1.00	1.00	1.01	1.01	1.03	0.99	1.01	1.00	0.99	0.98	1.00
	-0.15%	-0.09%	1.38%	0.51%	3.05%	-0.67%	1.30%	-0.37%	-0.58%	-2.35%	-0.45%
Number of operating firms	0.97	0.97	1.04	1.00	1.12	0.95	1.03	0.96	0.95	0.87	0.95
	-3.22%	-2.96%	3.78%	-0.24%	11.92%	-5.49%	3.41%	-4.17%	-5.12%	-12.66%	-4.55%
Avg. output per firm	102.25	7.52	18.54	6.78	4.54	6.60	0.60	9.33	0.15	1.53	4.20
	0.25%	0.19%	-1.26%	-0.41%	-2.87%	0.76%	-1.19%	0.46%	0.68%	2.51%	0.55%
Avg. productivity per firm	0.34	0.34	0.29	0.34	0.29	0.30	0.30	0.35	0.35	0.35	0.35
	0.25%	0.19%	-1.26%	-0.41%	-2.87%	0.76%	-1.19%	0.46%	0.68%	2.51%	0.55%
Total output sold	98.97	7.30	19.24	6.76	5.08	6.24	0.62	8.94	0.14	1.34	4.01
	-2.96%	-2.77%	2.47%	-0.65%	8.71%	-4.77%	2.18%	-3.73%	-4.48%	-10.47%	-4.03%

### 3.13.7 Conclusion

While the Armington specification based on regionally differentiated goods provides a popular and robust specification for numerical simulations of trade policy, it fails to explain empirical observations at firm-level in newer international trade literature. Recent models of international trade with heterogeneous firms overcome the limitations of the Armington specification and can at the same time be relatively easy integrated into aggregated equilibrium analysis. That has opened up the opportunity for CGE models to better depict and analyze mechanisms through which productivity and number of varieties impact the extensive margin of trade. This paper discusses an operational implementation of the firm heterogeneity theory of Melitz (2003) into the CGEBox model which addresses the shortcoming of the Armington specification while being relatively simple. It has proven numerically stable at least for medium sized aggregations of the GTAP data base.

### Appendix 1: Price linkages

The following section presents the price linkages between firms and agents. Firm's price  $PF_{irs}$  (inclusive of domestic transport margin) is defined as the price received by producers in region  $r$  for commodity  $i$  to be shipped to the sink region. If the commodity is shipped to the domestic market, the agent (purchase) price is

$$PA_{airs} = PF_{irs} (1 + \tau_{air}^{cd} + \varsigma_{air}^{cd}) \quad (32)$$

where  $\tau_{air}^c$  denote the consumption (sale) tax on each specific commodity for each agent while  $\varsigma_{air}^c$  is a uniform consumption tax across commodities and/or agents.

If the commodity is shipped to the other region/country, a bilateral export subsidy or tax ( $\tau_{irs}^e$ ) is applied to the firm offer price and determines the free on board (fob) price. An additional tax  $\varsigma_{irs}^e$  is also introduced into (33) representing the uniform export tax across destinations.

$$PE_{irs}^{FOB} = PM_{irs} (1 + \tau_{irs}^e + \varsigma_{irs}^e) \quad (33)$$

The FOB price  $PE_{irs}^{FOB}$  is augmented by the international transport margin  $t_{irs}^{tmg}$  (observed from GTAP and endogenous to the model) to establish the cif (cost –insurance and fricht) price:

$$PM_{irs}^{CIF} = PE_{irs}^{FOB} + t_{irs}^{tmg} \quad (34)$$

The bilateral import tax ( $\tau_{irs}^m$ ) converts the cif price into the bilateral import price, and  $\zeta_{irs}^m$  reflect a uniform tax shift across source countries

$$PM_{irs} = P_{irs}^{CIF} (1 + \tau_{irs}^m + \zeta_{irs}^m) \quad (35)$$

and finally the resulting bilateral import prices are converted to the agent prices by adding a consumption tax on imported commodities:

$$PA_{airs} = P_{irs}^{CIF} (1 + \tau_{air}^{cim} + \zeta_{air}^{cim}) \quad (36)$$

While uniform shift parameters are initially set to zero, the value of all other parameters are observed from GTAP 8 database. Potential users are strongly urged to consult Hertel (1997) and Mensbrugghe (2015).

### 3.13.8 Acknowledgments

Authors would like to thank Thomas W. Hertel, Edward J. Balistreri, and Dominique van der Mensbrugghe for their helpful comments.



### 3.14 GTAP-AEZ

#### 3.14.1 Summary

The GTAP-AEZ module allows to break land-use in a region to different agro-ecological zones using a multi-tier land-supply approach. The necessary data are available from GTAP and covered by a GTAP data base license and need to be added during the data preparation step. The GUI gives limited methodological choice. The exploitation part allows generating global maps which depict land use (changes).

The module draws on Huey-Lin, Incorporating Agro-Ecologically Zoned Land Use Data and Land-based Greenhouse Gases Emissions into the GTAP Framework (<https://www.gtap.agecon.purdue.edu/resources/download/2245.pdf>) and uses the following logic. Each region (nation, group of nation or NUTS2) in the model is split into Agro-Ecological Zones (AEZs, set aez). The total land use of an activity at regional level is a CES-aggregate of the land use in the different AEZs, i.e. all other costs shares do not differ across AEZs. In each AEZ, individual land use activities compete for available total managed land which is not fully mobile among individual activities, described by a four-tier or six-tier CET factor supply system.

The relation between the regions and the AEZs depicted by the cross-set aezFlag:

```
set aezFlag(r,aez);
```

In each AEZ, the total endogenous land use is dis-aggregated to the following three top-level land categories and an agricultural aggregate:

```
set landCat(aezNest) "Land categories" / forest,PastureLand,CropLand,Graz_Agr /;
```

The different activities in the GTAP Data Base are mapped to these to land categories, i.e. forestry comprises managed forest (frs), the grazing categories comprises all land used by animal production activities while agriculture covers all crops.

```
set landCat_a0(landCat,i0) "Map between land categories and activities"/
    forest.frs,
    PastureLand.(ctl,oap,rmk,wol)
    CropLand.(set.icrops)
/;
```

The code does not require that the SAM comprises the full detail of activities as defined above, but assigns potentially higher aggregated activities a to the land categories based on the mapping defined above. It however will throw an error if the sectoral aggregation leads to the case where one activity in the SAM would be assigned to multiple land categories as, for instance, in the case where all agricultural activities in the GTAP data base are aggregated.

```
set landCat_a(landCat,a);
landCat_a(landCat,a) $ sum(mapa(i0,a) $ landCat_a0(landCat,i0),1) = YES;

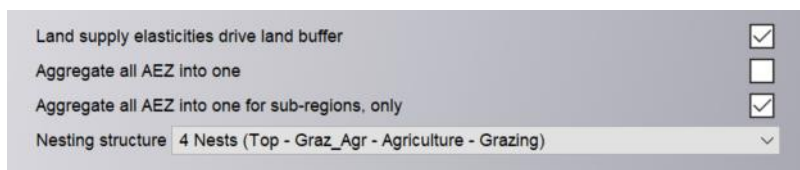
abort $ (sum(landCat_a(landCat,a),1) ne sum(a $ sum(landCat_a(landCat,a),1),1))
    "Not enough detail for agriculture / forestry, several sector assigned to same land category",
    landCat_a;
```

#### 3.14.2 Model equations

The model equations are defined in “gams\GtapAEZ\gatpAez\_model.gms”. The implementation differs in five important aspects from the AEZ model as coded in GEMPACK:

1. It comprises a four or six instead of three-tier CET distribution of total managed land: the upper nest distributes total managed land to forestry and all agricultural uses, the middle nest from all agricultural uses to pasture, i.e. land used in animal, and cropland, i.e. crop activities, and the lowest tier distributes pasture to the different animal activities to cropland to the individual crop activities. The middle tier is not present in the original implementation. The introduction of the additional nest allows making managed forest cover more “sticky” compared to switching between aggregate animal and crop use.
2. Additionally, the crop nests can be broken down to two nests with typical arable crops and horticulture in the other.
3. The two upper tiers use a volume preserving form of the CET (van der Mensbrugghe, D., & Peters, J. C. (2016, April). Volume preserving CES and CET formulations. In 2016 GTAP Conference Paper. GTAP Resource (Vol. 5070)).
4. Land supply elasticities can either relate to the benchmark total land stock or to a land buffer.
5. If NUTS2 regions for European countries as sub-national regions are present, they are also broken down to individual AEZs.
6. It is possible to aggregate all AEZ into one, either for all regions or for sub-national regions, only.

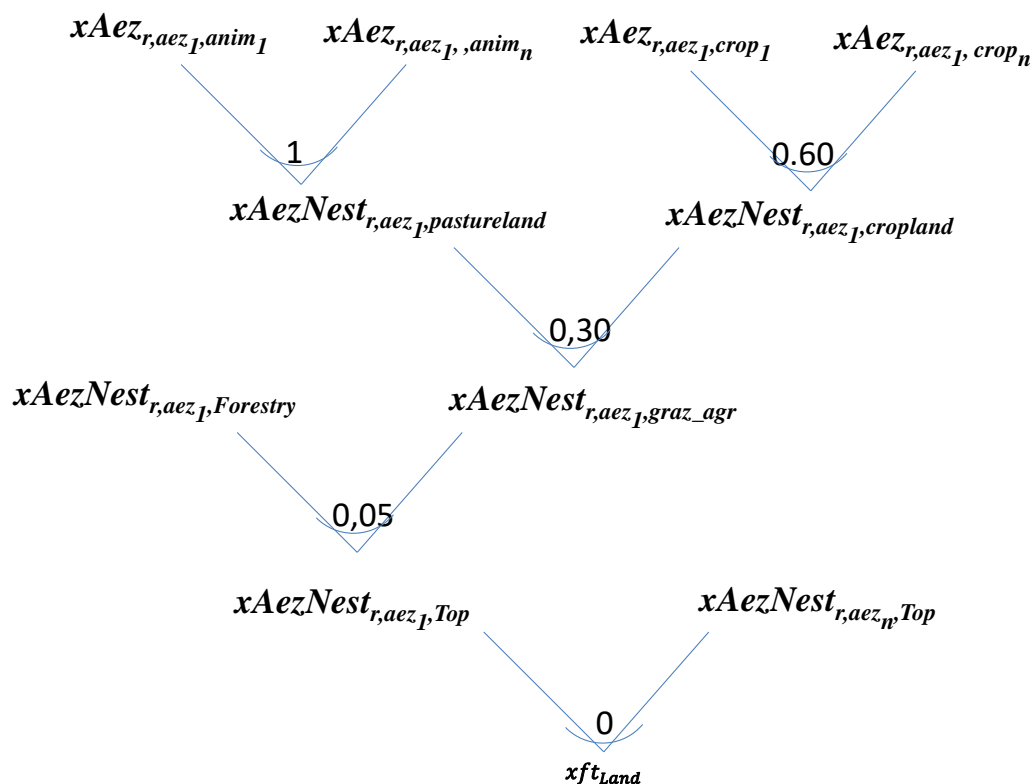
It is hence recommended switching the following options on via GUI controls. Switching the land supply off gives a model closer to the GEMPACK implementation:



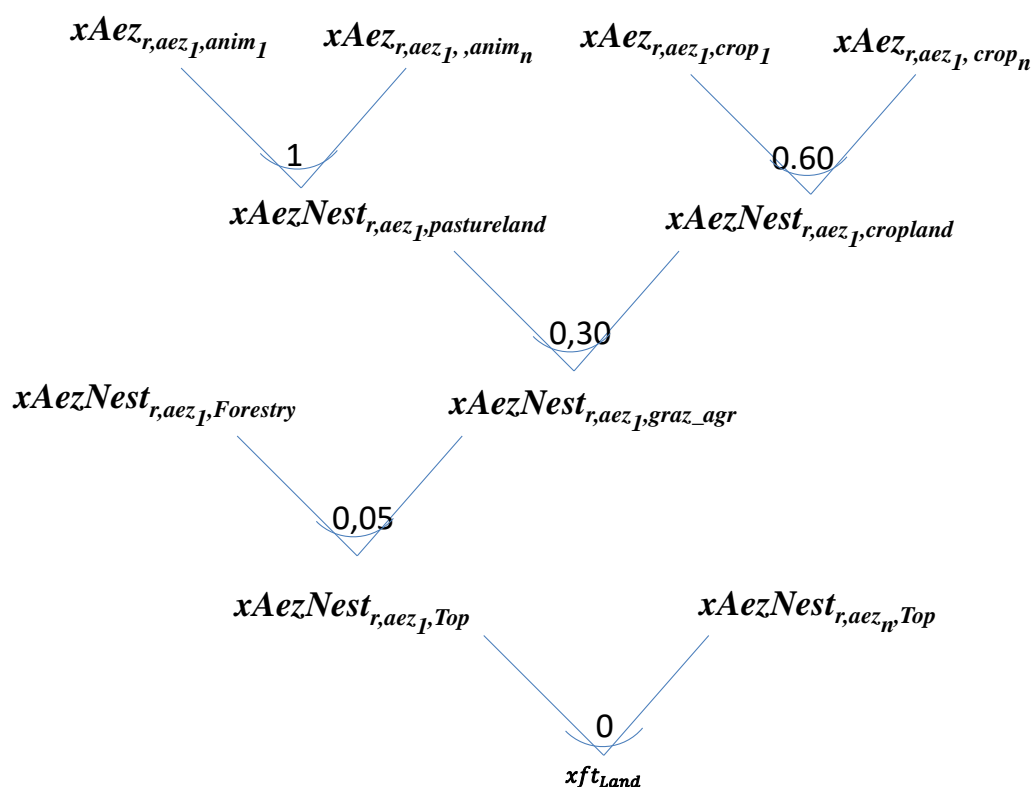
The screenshot shows a GUI panel with four options, each with a checkbox on the right, and a dropdown menu at the bottom. The options are:

- Land supply elasticities drive land buffer: ☒
- Aggregate all AEZ into one: ☐
- Aggregate all AEZ into one for sub-regions, only: ☒
- Nesting structure: 4 Nests (Top - Graz\_Agr - Agriculture - Grazing) (dropdown menu)

The figure below gives an overview on the CET based transformation of the total land stock in a region to the land stock in the different AEZs and to the four layers.



The version with the six layers looks as follows:



### Total economically used land stock

The model can be used in two variants to depict the total land stock in economic use in each AEZ:

1. The usual factor stock equation with a supply elasticity  $\text{etaf}$  which is equivalent – leaving the dis-aggregation to AEZs aside – to the factor stock equation in the GTAP standard model. The land stock at the benchmark  $\text{xAezNestBase}$  is updated with the factor supply elasticity  $\text{etaf}$  applied to the numeraire normalized average returns to land in the region  $\text{pft}$ . In that version, total land use in all AEZs of a region will hence expand or shrink by the same percentage as it driven by a uniform land prices at model region level and not an AEZ specific land price. If  $\text{p\_etaf}$  is zero, the equation defines a fixed stock.

```

*
*
* --- land supply to agriculture from stock (that is the standard
* equation in the GTAP model)
*
e_xftTop(rs,aez,"top",ts) $ (p_xAezNestBase(rs,aez,"top",ts) $ (pdAezNest.range(rs,aez,"top",ts) ne 0)) ..
    xAezNest(rs,aez,"top",ts)/xAezNest.scale(rs,aez,"top",ts)

=E= (p_xAezNestBase(rs,aez,"top", "%t0%")
    * sum((r_r(rs,rNat),lnd), (pft(rs,lnd,ts)/ pfact(rNat,ts))**p_etaf(rNat,lnd)))/xAezNest.scale(rs,aez,"top",ts);

```

2. A version where land “buffers” are defined in 1000ha, which drive the amount of unmanaged land, i.e. higher land rent move specific types of unmanaged land into economic use.

```

e_unManagedLnd(rs,aez,unManaged,ts(tRun)) $ ( (p_xAezNestBase(rs,aez,"top",ts) gt eps) $ (p_bufferShare(rs,aez,unManaged) gt eps)) ..
    unManagedLnd(rs,aez,unManaged,ts)/unManagedLnd.scale(rs,aez,unManaged,ts)
=E=
    (p_landUse(rs,aez,unManaged,"ha") * 1.E-6
    --- exogenously determined changes in total managed land in 1000ha plus very modest effect of price changes
    - (v_topLandChange(rs,aez,ts)
        + ( xAezNest.l(rs,aez,"top", "%t0%") $ useAezTopForEtaf
            + p_landUse(rs,aez,"buffer", "ha") * 1.E-6 $ (not useAezTopForEtaf))
        * [
            --- this increases land abandonment if price start to drop below 30% of the benchmark value,
            but never abandons more than the given starting area
            m_smoothMax( [(ncpVUSin(0.30,paezNest(rs,aez,"top",ts)/paezNest.l(rs,aez,"top", "%t0%"),1.E-3)-0.30)*20], (-1))
            $ (paezNest.l(rs,aez,"top",tRun-1) le 0.5)
            + [(paezNest(rs,aez,"top",ts)/p_paezTop(rs,aez,ts))**sum(lnd,p_etaf(rs,lnd)) - 1 ] $ (v_topLandChange.range(rs,aez,ts) eq 0)
        ]
    )
    --- distributed according to shares on total buffer of unmanaged land (= different types such as nat.forest, savannah ..)
    * (p_landUse(rs,aez,unManaged,"ha")*p_bufferShare(rs,aez,unManaged))

    /sum(unManagedl,p_landUse(rs,aez,unManagedl,"ha")*p_bufferShare(rs,aez,unManagedl))

    /unManagedLnd.scale(rs,aez,unManaged,ts)
;

```

In this second case, the land supply elasticity  $\text{p\_etaf}$  is applied to a land bugger. This share is AEZ specific and derived from national data on total available cropland (in the file `data/landBuffer.gms`). The national data are distributed to the AEZs in each aggregated region during benchmark using data on savanna, shrubland and other lands, the latter only with 1%.

```

parameter p_bufferDistr(land_cover) / savnGrassLnd 1
                                     shrubLand 1
                                     unmanagedForest 1
                                     otherLand 0.01
                                     /;

p_distrKey(rNat,Aez) $ aezFlag(rNat,aez)
= sum(land_cover,p_bufferDistr(land_cover)*landUse(rNat,aez,land_cover,"ha"));

landUse(rNat,aez,"buffer","ha") $ p_distrKey(rNat,aez)
= Max(0, [p_croplandBuffer(rNat) - sum(aezFlag(rNat,aez1), landUse(rNat,aez1,"cropland","ha"))]
    * p_distrKey(rNat,aez)/sum(aezFlag(rNat,aez1), p_distrKey(rNat,aez1)));

```

For this case, the total stock land in an AEZ is defined as follows. We add to the benchmark stock of total managed land  $\text{xAezNestBase}_{\text{top},t0}$  the endogenously determined changes in the extend of the different categories unmanaged of natural land  $\text{unManagedLnd}$  compared to the base ( $\text{landUse}$ ). The resulting difference in hectares is then converted in constant USD by multiplying with the base year stock of land in economic use in the AEZ.

The `v_topLandChange` variable used in the equation `e_unManagedLnd` is generated by G-RDEM if exogenous crop land expansion projections are used. If it is fixed to zero – indicating that this mechanism is not active – the land supply elasticity `p_etaf` is used instead. To these effects comes as a safeguard mechanism in case that land rents drop below 30% of the benchmark. It is active only if the land rents dropped below 50% in the previous period simulation.

### Distribution of total land demand from regional to AEZ level

The land demand `xAez` for each activity `a` in each Agro-Ecological Zone `aez` is defined in the equation `e_pAez` based on a CES function. Factor returns at market level `pf` are used as agent specific ad-valorem tax rates on land which determine the differences between returns at market prices `pf` and at agent prices `pfa` are not differentiated by AEZ:

```
*
* --- Land Demand for each activity in each aez, drives distribution in space
* (relation between average firm's price for land for the activity
* and the firm's price at AEZ level drives the distribution)
*
pAezEq(rs(r),aez,aIn(a),ts(t)) $ (aezFlag(r,aez) $ andAez(r,aez,a,t) ) ..
*
xAez(r,aez,a,t)/xAez.scale(r,aez,a,t)
=e= andAez(r,aez,a,t)
* sum(lnd, xf(r,lnd,a,t)/xAez.scale(r,aez,a,t)
* (pf(r,lnd,a,t)/pAez(r,aez,a,t))**sigmaAez(r,a));
*
```

The distribution uses hence a CES approach, where the relation between the average return for land in the activity at regional level as defined by the macro `m_pfa` relative to its return in a specific AEZ drive the allocation of the activity in space. This can be understood as a cost minimization problem to allocate crops such as to minize the land rents to pay, taking however differences in land qualities into account. The total to distribute is equal to the total land use of that activity at regional level `xf`.

The average return `pf` uses the usual dual price aggregator defined in the equation `e_pfaAez`. Note again that ad-valorem factor taxes are not present as they appear as a multiplicative term on both sides of the equation. The equation considers the simpler case of a Leontief relation with fixed allocation shares separately.

```
*
* --- average factor price per activity aggregated over AEZs in each region
*
e_pfaAez(rs(r),lnd,aIn(a),ts(t)) $ ( sum(aezFlag(r,aez), p_andAez(r,aez,a,t) ) $ (pf.range(r,lnd,a,t) ne 0) ) ..
pf(r,lnd,a,t)
=e= [sum(aezFlag(r,aez) $ p_andAez(r,aez,a,t),
p_andAez(r,aez,a,t) * pAez(r,aez,a,t)**(1-p_sigmaAez(r,a)))*(1/(1-p_sigmaAez(r,a)))] $ p_sigmaAez(r,a)
+ sum(aezFlag(r,aez) $ p_andAez(r,aez,a,t),p_andAez(r,aez,a,t) * pAez(r,aez,a,t) $ (not p_sigmaAez(r,a));
```

The volume preserving formulation used for the top level land use nests maximizes the revenue of a CET aggregator to distribute a physical land stock  $V$  among competing uses  $X$ :

$$\max R = A \left[ \sum_i g_i (\lambda_i P_i X_i)^v \right]^{1/v} \quad (3)$$

Where  $g_i$  are share parameters,  $P_i$  are given prices,  $A$  is a scaling factor and  $X_i$  are the land use quantities to be optimized and  $v$  refers to the transformation elasticity. The A-CET introduces the condition that the sum of  $X$  is fixed to the given volume  $V$ , in this case, a land stock measured in physical units:

$$V = \sum_i X_i \quad (4)$$

With a given transformation elasticity  $\varpi$ , this generates the following solutions for the competing land uses  $X_i$  and the dual price aggregator  $P^c$

$$X_i = \gamma_i (A \lambda_i)^\varpi \left( \frac{P_i}{P^c} \right)^\varpi \quad (5)$$

$$P^c = A \left[ \sum_i \gamma_i (\lambda_i P_i)^\varpi \right]^{1/\varpi} \quad (6)$$

The advantage of using the modified A-CET formulation is twofold. Firstly, the parameters provided by the GTAP-AEZ model – or by alternative models based on the standard CET – can still be used for benchmarking. Secondly, structural changes to the equations in the GTAP-AEZ model are relatively minor. Specifically, equation (5) is indeed identical to the original CET formulation while equation (6) only changes the exponent. However, the price index in (6) is no longer equal to the average revenue exhaustion price  $P^e$  which has to be defined in a separate equation:

$$P^e = \sum X_i P_i / V \quad (7)$$

Accordingly, the dual prices adggregator from individual land use activities and land use sub-nest to a land use nests are defined as follows in equation `e_pDAezNest`, where the differences between the volume preserving and standard CET are solely in the exponents:

```
*
* ----- Dual price aggregator for CET, either default or for volume preserving
*
e_pDAezNest(rs,aez,aezNest,ts) $ ( sum(aezNest_a(aezNest,a), p_gAez(rs,aez,a,ts))
    or sum(aezNest_n(aezNest,aezNest1), p_gAez(rs,aez,aezNest1,ts))
    $ (xAezNest.range(rs,aez,aezNest,ts) ne 0)
    ) ..
0 = 0 =
(
*
*   --- sluggish land mobility in aez nest, dual price aggregator for nests
*   (CET is not infinite)
*
pDAezNest(rs,aez,aezNest,ts)/pDAezNest.scale(rs,aez,aezNest,ts)
- [
*
*   --- activities linked into nest (not the missing 1 if a volume preserving CET is used)
*
sum(aezNest_a(aezNest,a) $ p_gAez(rs,aez,a,ts),
    p_gAez(rs,aez,a,ts) * pAez(rs,aez,a,ts)**(1 $ p_aezCETType(aezNest,"CET") + p_omegaAez(rs,aez,aezNest)))
*
*   --- subnests linked into sub nest
*
+ sum(aezNest_n(aezNest,aezNest1) $ p_gAez(rs,aez,aezNest1,ts),
    p_gAez(rs,aez,aezNest1,ts) * pAez(rs,aez,aezNest1,ts)**(1 $ p_aezCETType(aezNest,"CET") + p_omegaAez(rs,aez,aezNest)))
]**(1/(1 $ p_aezCETType(aezNest,"CET") + p_omegaAez(rs,aez,aezNest))) * 1/pDAezNest.scale(rs,aez,aezNest,ts)
) $ (p_omegaAez(rs,aez,aezNest) ne inf)
```

The CET dual price index `pDAezNest` for the case of A-CET is defined by two equations. The first defines total land rent revenues `vAezNest`:

```

*
* --- Value of land rent in aezNests (exhaustion), needed for volume preserving nests
*
e_vAezNest(rs(r), aez, aezNest, ts(t)) $ ( [ sum(aezNest_a(aezNest, a), p_gAez(rs, aez, a, ts))
or sum(aezNest_n(aezNest, aezNest1), p_gAez(rs, aez, aezNest1, ts)) ]
$ p_aezCETType(aezNest, "volPreserv") $ (p_omegaAez(rs, aez, aezNest) ne inf) ) ..
*
* --- This equation is not needed for the default CET where the dual price aggregator
*
vAezNest(rs, aez, aezNest, ts)/vAezNest.scale(rs, aez, aezNest, ts)

=e= [ sum(aezNest_a(aezNest, a) $ p_gAez(rs, aez, a, ts),
pAez(rs, aez, a, ts) * xAez(rs, aez, a, ts))

+ sum(aezNest_n(aezNest, aezNest1) $ p_gAez(rs, aez, aezNest1, ts),
pAezNest(rs, aez, aezNest1, ts) * xAezNest(rs, aez, aezNest1, ts))
] /vAezNest.scale(rs, aez, aezNest, ts) ;

```

And from there the average exhaustion price pAezNest:

```

*
* --- Average land rent in aezNest (exhaustion), needed for volume preserving nests
*
e_pAezNest(rs(r), aez, aezNest, ts(t)) $ ( sum(aezNest_a(aezNest, a), p_gAez(rs, aez, a, ts))
or sum(aezNest_n(aezNest, aezNest1), p_gAez(rs, aez, aezNest1, ts)) ) ..
*
* --- for the default CET formulation, this is a just a copy of the dual aggregator
*
[pAezNest(rs, aez, aezNest, ts)*xAezNest(rs, aez, aezNest, ts)/vAezNest.scale(rs, aez, aezNest, ts)]
$ (p_aezCETType(aezNest, "volPreserv") $ (p_omegaAez(rs, aez, aezNest) ne inf) )

+ (pAezNest(rs, aez, aezNest, ts)/pAezNest.scale(rs, aez, aezNest, ts)) $ (p_aezCETType(aezNest, "CET") or (p_omegaAez(rs, aez, aezNest) eq inf))

=e= [vAezNest(rs, aez, aezNest, ts)/vAezNest.scale(rs, aez, aezNest, ts)] $ (p_aezCETType(aezNest, "volPreserv") $ (p_omegaAez(rs, aez, aezNest) ne inf)

*
* --- for the default CET formulation, this is a just a copy of the dual aggregator
*
+ (pDAezNest(rs, aez, aezNest, ts)/pAezNest.scale(rs, aez, aezNest, ts)) $ (p_aezCETType(aezNest, "CET") or (p_omegaAez(rs, aez, aezNest) eq inf));

```

If the standard CET formulation used, the e\_vAezNest equation is not needed, and the second equation simply copies the dual price aggregator results pDAezNest to the variable pAezNest, as marginal and average returns are identical.

The land supply from a land use nest xAezNest to the individual land use activities xAez is defined by the equation e\_xAez. No difference between the two CET formulations is found. As in the equations above, the case with fully mobile allocation needs to be treated differently:

```

*
* --- land supply to activity from land supply nest
*
e_xAez(rs, aez, a, ts) $ (aezFlag(rs, aez) $ p_andAez(rs, aez, a, ts) $ (pAez.range(rs, aez, a, ts) ne 0)) ..
*
0 =e= [xAez(rs, aez, a, ts)/xAez.scale(rs, aez, a, ts)
- sum(aezNest_a(aezNest, a),
(p_gAez(rs, aez, a, ts) * xAezNest(rs, aez, aezNest, ts))/xAez.scale(rs, aez, a, ts)
* (pAez(rs, aez, a, ts) / pDAezNest(rs, aez, aezNest, ts))**p_omegaAez(rs, aez, aezNest)) ]
$ (sum(aezNest_a(aezNest, a), p_omegaAez(rs, aez, aezNest)) ne inf)

+ [pAez(rs, aez, a, ts)/p_gAez(rs, aez, a, ts) - sum(aezNest_a(aezNest, a), pDAezNest(rs, aez, aezNest, ts))]
$ (sum(aezNest_a(aezNest, a), p_omegaAez(rs, aez, aezNest)) eq inf);

```

The same relation holds for sub-nests:

```

*
* --- land supply to sub-nest from higher level land-supply nest
*
e_xAezNest(rs, aez, aezNest, ts) $ (sum(aezNest_n(aezNest1, aezNest), p_gAez(rs, aez, aezNest, ts)) $ (pDAezNest.range(rs, aez, aezNest, ts) ne 0)) ..
*
0 =e=
*
* --- "sluggish" factor supply from another nest AezNest1 to the current nest
*
+ sum(aezNest_n(aezNest1, aezNest),
xAezNest(rs, aez, aezNest, ts)/xAezNest.scale(rs, aez, aezNest, ts)
- (p_gAez(rs, aez, aezNest, ts) * xAezNest(rs, aez, aezNest1, ts))/xAezNest.scale(rs, aez, aezNest, ts)
* (pAezNest(rs, aez, aezNest, ts)/pDAezNest(rs, aez, aezNest1, ts))**p_omegaAez(rs, aez, aezNest1))
$ (sum(aezNest_n(aezNest1, aezNest) $ (p_omegaAez(rs, aez, aezNest1) ne inf), 1))
*
* --- non "sluggish" factor supply from another nest to current nest
*
+ sum(aezNest_n(aezNest1, aezNest) $ (p_omegaAez(rs, aez, aezNest1) eq inf),
pAezNest(rs, aez, aezNest, ts)
- pAezNest(rs, aez, aezNest1, ts) )
;

```



## Link into global model

As land is immobile across the AEZ, total factor use of land `xft` and the related price `pft` are defined by the following three equations. The first equation `e_xftLand` sums the total endogenous land `xftTop` in each `aez` to the national or sub-national total `xft`. The second equation `e_vftLand` sums up total land rents and the third one `e_pftLand` defines the average factor price for land, `pft`:

```
*
* --- aggregate land use over aez and calculate average price,
* feeds into regional income / tax equations in standard GTAP model
*
xftLandEq(rs,lnd,ts) $ ((Xft.range(rs,lnd,ts) ne 0) $ sum(aezFlag(rs,aez),1) $ xftFlag(rs,lnd)) ..
  xft(rs,lnd,ts)/xft.scale(rs,lnd,ts)
  =E= sum(aezFlag(rs,aez), xAezNest(rs,aez,"top",ts))
    /sum(aezFlag(rs,aez), xAezNestBase(rs,aez,"top",ts)) * xftBase(rs,lnd)/xft.scale(rs,lnd,ts);

vftLandEq(rs,ts) $ (sum(lnd,xftFlag(rs,lnd) $ sum(aezFlag(rs,aez),1))) ..
  vftLand(rs,ts)/sum(lnd,xft.scale(rs,lnd,ts))
  =E= sum(aezFlag(rs,aez), xAezNest(rs,aez,"top",ts)*pAezNest(rs,aez,"top",ts)) /sum(lnd,xft.scale(rs,lnd,ts));

pftLandEq(rs,lnd,ts) $ ((pft.range(rs,lnd,ts) ne 0) $ sum(aezFlag(rs,aez),1) $ xftFlag(rs,lnd)) ..
  pft(rs,lnd,ts) * xft(rs,lnd,ts)/xft.scale(rs,lnd,ts)
  =E= vftLand(rs,ts)/xft.scale(rs,lnd,ts) * p.scaleVft(rs);
```

## Carbon stock accounting

The carbon stock accounting equation `e_dcarbon` (shown in full below) comprises in a first block changes between the three economic use aggregates forest, cropland and pastureland for which carbon stock data are available. It multiplies the absolute change in each aggregate in current simulation year (or against the benchmark in a comparative static run), normalized with the stock in the benchmark to define a relative change, with the stock in ha stored on `p_landUse` times the carbon content `p_carbon` in metric tons per ha, converted to millions metric tons, and multiplied with the overall scaler used in the model:

```
--- change in ha managed top categories to which carbon stocks are attached

sum((r_r(rs,rNat),aez,aezNest) $ (xAezNestBase(rs,aez,aezNest,"%t0%") $ p_carbon(rNat,aez,aezNest,"carbon")),
  (xAezNest(rs,aez,aezNest,tRun) - xAezNest.l(rs,aez,aezNest,tRun-1))
  /xAezNestBase(rs,aez,aezNest,"%t0%")

--- map into hectare changes times carbon stock per ha

  * landUse(rs,aez,aezNest,"ha")*p_carbon(rNat,aez,aezNest,"carbon"))* 1.E-6 * gblScale
```

The second block is only used if land supply elasticities drive the different categories of unmanaged land reflects equation `e_unManagedLnd` discussed above. We assign carbon stock `p_carbon` to the changes in the unmanaged land extent `unManagedLnd` against last year's result or the benchmark:

```
.. + sum((r_r(rs,rNat),aez,unManaged) $ (landUse(rs,aez,unManaged,"ha") $ landSupplyElas(rs,aez,unManaged)
*                                     $ p_bufferShare(rs,aez,unManaged))),
*
* --- price induce released of un-managed land categories in ha, times carbon stock per ha
*
+ (unManagedLnd(rs,aez,unManaged,tRun)-unManagedLnd.l(rs,aez,unManaged,tRun-1))
  * p_carbon(rNat,aez,unManaged,"carbon")) * gblScale
```

Combined, the equation looks as follows:

```

e_dcarbon(rsNat(rNat),ts(tRun)) $ ( (not sameas(tRun,"%t0%")) $ sum((r_r(rs,rNat),aez,land_Cover)
$ aezFlag(rs,aez),p_carbon(rNat,aez,land_cover,"carbon"))) ..

dCarbon(rNat,tRun)
/sum((r_r(rs,rNat),aez,land_Cover) $ aezFlag(rs,aez),p_carbon(rNat,aez,land_cover,"carbon"))
=E= [
*
*      --- change in ha managed top categories to wich carbon stocks are attached
*
*      sum((r_r(rs,rNat),aez,aezNest) $ ( (p_xAezNestBase(rs,aez,aezNest,"%t0%") gt eps) $ p_carbon(rNat,aez,aezNest,"carbon")),
*      (xAezNest(rs,aez,aezNest,tRun) - xAezNest.l(rs,aez,aezNest,tRun-1))
*      /p_xAezNestBase(rs,aez,aezNest,"%t0%")
*
*      --- map into hectare changes times carbon stock per ha
*
*      * p_landUse(rs,aez,aezNest,"ha")*p_carbon(rNat,aez,aezNest,"carbon")* 1.E-6 * gblScale
*
*      $$iftheni.landSupply "%landSupplyElas%"=="on"
*      + sum((r_r(rs,rNat),aez,unManaged) $ ( (p_landUse(rs,aez,unManaged,"ha") gt eps)
*      $ (p_xAezNestBase(rs,aez,"top",ts) gt eps)
*      $ (p_bufferShare(rs,aez,unManaged) gt eps) $ aezFlag(rs,aez)),
*
*      --- price induce released of un-managed land categories in ha, times carbon stock per ha
*
*      + (unManagedLnd(rs,aez,unManaged,tRun)-unManagedLnd.l(rs,aez,unManaged,tRun-1))
*      * p_carbon(rNat,aez,unManaged,"carbon") * gblScale
*      $endif.landSupply
*
*      ]
/sum((r_r(rs,rNat),aez,land_cover) $ aezFlag(rs,aez),p_carbon(rNat,aez,land_cover,"carbon"));

```

We add an equation which defines the global change in CO2 equivalents:

```

e_dCarbonGblCo2(ts) $ ((card(rsNat) gt 0) $ (not sameas(ts,"%t0%"))) ..

dCarbonGblCo2(ts) =E= sum(rsNat $ sum((r_r(rs,rNat),aez,land_Cover) $ aezFlag(rs,aez),p_carbon(rsNat,aez,land_cover,"carbon")),
dCarbon(rsNat,ts)) * 3.67;

```

## MCP parings

The resulting pairings for all model equations in MCP format are shown below:

```

model GTAP_AEZ
/

$$iftheni.landSupply "%landSupplyElas%"=="on"
pftTopPriceEffect1Eq.pftTopPriceEffect1
pftTopPriceEffectEq.pftTopPriceEffect
unManagedLndEq.unManagedLnd
$endif.landSupply

dCarbonEq.dCarbon
dCarbonGblCo2Eq.dCarbonGblCo2

xftTopEq.pdAezNest
vAezNestEq.vAezNest
xAezNestEq.pdAezNest
pAezNestEq.pAezNest
pAezNestEq.xAezNest

xAezEq.pAez
pAezEq.xAez
pAezEq.pf

pftLandEq.pft
xftLandEq.xft
vftLandEq.vftLand
/;

```

### 3.14.3 Parameterization

The necessary data for the AEZ module must be loaded during the data preparation step. The aezFlags are set depending on finding any land use activities in the aez data base (see gams/gtapAez/gtapAez\_cal.gms):

```

aezFlag(rNat,aez) $ (sum(a, landUse(rNat,aez,a,"vfm"))
and landuse(rNat,aez,"land","vom")) = Yes;

```

The land use by activity is initialized reflecting the global scaling factor as long as it is above a minimum threshold:

```

*
* --- land use of individual activity in measured in constant USD
* at market prices (same factor use at regional level)
*
xAez.l(r,aez,a,t0) $ aezFlag(r,aez) = landUse(r,aez,a,"vfm") * gblScale;
xAez.l(r,aez,a,t0) $ (xAez.l(r,aez,a,t0) le threshold) = 0;

```

If a land use activity is zero in all AEZs with data for a region, total land use shares are used:

```

*
* --- missing land use for an activity in all aezs of a region:
* use total land use times share on land factor income at market prices
*
p_landUse(aezFlag(r,aez),"land","vfm") = sum(a, p_landUse(r,aez,a,"vfm"));

xAez.l(r,aez,a,t0) $ (sum(lnd,xf.l(r,lnd,a,t0)) and aezFlag(r,aez)
and (not sum(aezFlag(r,aez1),xAez.l(r,aez1,a,t0))))

*
* --- share on total land return
*
= sum(lnd,xf.l(r,lnd,a,t0)/xft.l(r,lnd,t0))

*
* --- share on total land return of these AEZ on all AEZ
*
* p_landUse(r,aez,"land","vfm")/sum(aezFlag(r,aez1),p_landUse(r,aez1,"land","vfm"));

```

Afterwards, scaling ensures that adding up over the AEZs recovers the given total land use in constant USD. This accounts for (1) tiny numerical inaccuracies in the original data, (2) impacts of filtering out small transactions from the SAM and below from the AEZ Data base

```

*
* -- remove tiny land and assign again
*
xAez.l(aezFlag(r,aez),a,t0) $ (xAez.l(r,aez,a,t0) le threshold) = 0;
xAez.l(r,aez,a,t0) $ (sum(lnd,xf.l(r,lnd,a,t0)) and aezFlag(r,aez)
and (not sum(aezFlag(r,aez1),xAez.l(r,aez1,a,t0))))
= sum(lnd,xf.l(r,lnd,a,t0)/xft.l(r,lnd,t0))
* p_landUse(r,aez,"land","vfm")/sum(aezFlag(r,aez1),p_landUse(r,aez1,"land","vfm"));

*
* --- scale land income at market prices at the level of AEZ to regional one
*
xf.scale(r,lnd,a,t0) = sum(aezFlag(r,aez), xAez.l(r,aez,a,t0));
xAez.l(aezFlag(r,aez),a,t0) $ xAez.l(r,aez,a,t0)
= xAez.l(r,aez,a,t0) * sum(lnd,xf.l(r,lnd,a,t0)/xf.scale(r,lnd,a,t0));
xf.scale(r,lnd,a,t0) = 1;

p_landUse(aezFlag(r,aez),a,"vfm") = xAez.l(r,aez,a,"%t0%");
p_landUse(aezFlag(r,aez),"land","vfm") = sum(a, p_landUse(r,aez,a,"vfm"));

```

In order for the A-CET to work properly, the quantities must be defined in physical units. We therefore redefine xAez in Mio ha and define the prices such that they exhaust the reported value from the data base:

```

*
* ---- redefine xAez to be defined in Mio ha
*
xAez.scale(r,aez,aeza,"%t0%") = xAez.l(r,aez,aeza,"%t0%");
xAez.l(r,aez,aeza,"%t0%") $ xAez.l(r,aez,aeza,"%t0%") = landUse(r,aez,aeza,"ha")*1.E-6;
pAez.l(r,aez,aeza,"%t0%") $ xAez.l(r,aez,aeza,"%t0%") = pAez.l(r,aez,aeza,"%t0%") * xAez.scale(r,aez,aeza,"%t0%")/xAez.l(r,aez,aeza,"%t0%");

```

From there, we aggregate the nests:

```

*
* --- aggregate quantities and prices over single activities to nests
*
set aezNestToDo(aezNest), curAezNestToDo(aezNest);
alias(aezNestToDo, aezNestToDo1);
aezNestToDo(aezNest) = YES;

option kill=pAezNest.1;

While( card(aezNestToDo),
*
* --- we can calculate all nests which do not require information on nests
*   not yet calculated
*
option kill=curAezNestToDo;
curAezNestToDo(aezNestToDo) $ (not sum(aezNest_n(aezNestToDo, aezNestToDo1), 1)) = YES;

xAezNest.1(r, aez, curAezNestToDo, t0) $ ( (not xAezNest.1(r, aez, curAezNestToDo, t0)) $ aezFlag(r, aez) )
= sum(aezNest_n(curAezNestToDo, aezNest1), xAezNest.1(r, aez, aezNest1, t0))
+ sum(aezNest_a(curAezNestToDo, a), xAez.1(r, aez, a, t0));

pAezNest.1(r, aez, curAezNestToDo, t0) $ (xAezNest.1(r, aez, curAezNestToDo, t0) $ aezFlag(r, aez) )
= [ sum(aezNest_n(curAezNestToDo, aezNest1), xAezNest.1(r, aez, aezNest1, t0) * pAezNest.1(r, aez, aezNest1, t0))
+ sum(aezNest_a(curAezNestToDo, a), xAez.1(r, aez, a, t0) * pAez.1(r, aez, a, t0))
] / xAezNest.1(r, aez, curAezNestToDo, t0);

aezNestToDo(curAezNestToDo) = NO;
);
*
pDAezNest.1(r, aez, aezNest, t0) $ aezFlag(r, aez) = pAezNest.1(r, aez, aezNest, t0);

xAezNestBase(r, aez, aezNest, t0) $ aezFlag(r, aez) = xAezNest.1(r, aez, aezNest, t0);
xAezNests.1(r, aez, aezNest, t0) $ aezFlag(r, aez) = xAezNest.1(r, aez, aezNest, t0);

```

We leave some additional code out which estimates missing data in hectares from the given volume data in hectares and given economic data.

If the user decides to aggregate over the AEZ (either for all model regions or for sub-national units, only), a new AEZ type called aezAgg is defined to which all data are aggregated:

```

$$onMulti
set aez / aezAgg /;
$$offMulti

p_landUse(rs, "aezaggg", a, "vfm") = sum(aez, p_landUse(rs, aez, a, "vfm"));
p_landUse(rs, "aezaggg", "land", "vom") = sum(aez, p_landUse(rs, aez, "land", "vom"));
p_landUse(rs, "aezaggg", lt, "ha") = sum(aez, p_landUse(rs, aez, lt, "ha"));
p_landUse(rs, "aezaggg", a, "ha") = sum(aez, p_landUse(rs, aez, a, "ha"));
p_landUse(rs, "aezaggg", a, "evfa") = sum(aez, p_landUse(rs, aez, a, "evfa"));

p_carbon(rs, "aezaggg", lt, "carbon") $ p_landUse(rs, "aezaggg", lt, "ha")
= sum(aez, p_carbon(rs, aez, lt, "carbon") * p_landUse(rs, aez, lt, "ha")) / p_landUse(rs, "aezaggg", lt, "ha");

p_carbon(rs, "aezaggg", lt, "foreGoneSeqPerYear") $ p_landUse(rs, "aezaggg", lt, "ha")
= sum(aez, p_carbon(rs, aez, lt, "foreGoneSeqPerYear") * p_landUse(rs, aez, lt, "ha")) / p_landUse(rs, "aezaggg", lt, "ha");

```

For post-model reporting purposes, original flags are stored, and then removed:

```

aezFlagOri(rs, aez) = aezFlag(rs, Aez);
aezFlag(rs, aez) = no;

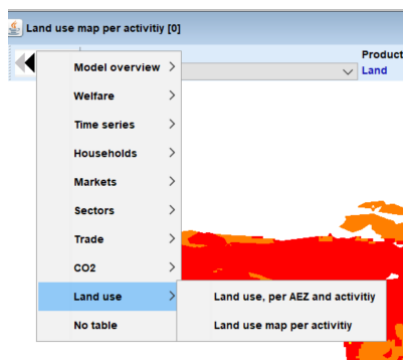
aezFlag(rs, "aezaggg") $ (sum(a, p_landUse(rs, "aezaggg", a, "vfm"))
and p_landuse(rs, "aezaggg", "land", "vom")) = Yes;

```

If this aggregation mechanism is active, proportionality assumptions are used to generate aez-specific results for reporting.

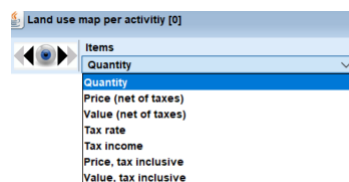
### 3.14.4 Visualization of GTAP-AEZ results

The views relating to results of GTAP-AEZ can be found under



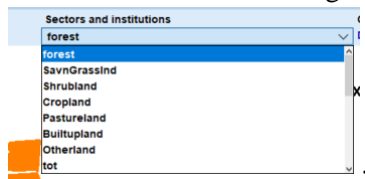
The first view provides a tabular overview at global level, i.e. results are aggregated over regions to the global 18 AEZs:

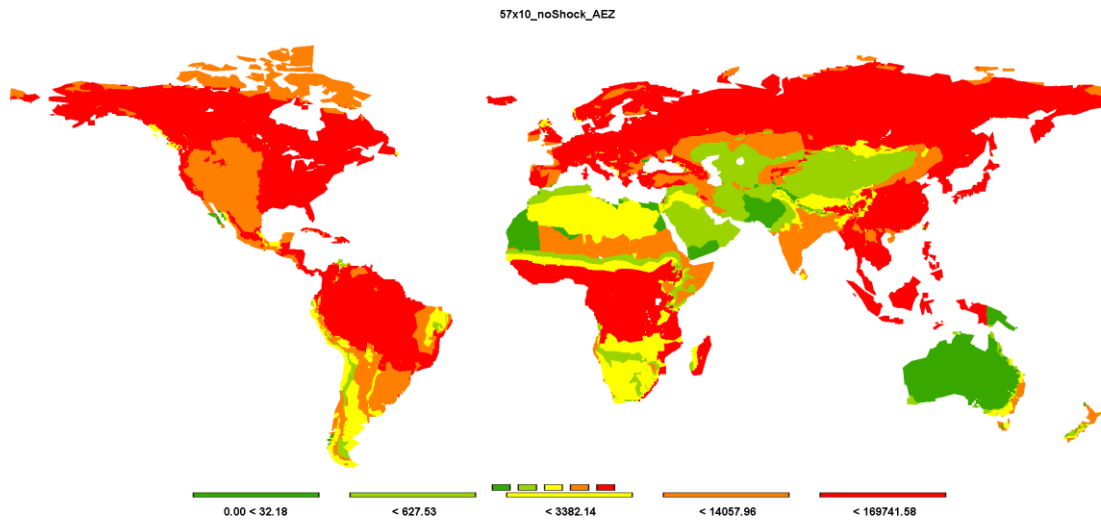
Land use, per AEZ and activity [0]										
Regions	Items		Origins		Years					
wor	Quantity		Domestic		shock					
87x10_noShock_AEZ										
forest	SavnGrassland	Shrubland	Cropland	Pastureland	Builtupland	Otherland	tot	Paddy rice	Wheat	
AEZ1	5368.78	35709.01	149810.88	27488.91	202805.64	1110.16	199946.16	622239.56	1414.23	
AEZ2	2200.48	74089.58	48676.56	59038.25	114788.45	527.87	97412.90	396734.09	1606.34	
AEZ3	13716.38	164351.48	42867.55	122297.73	133454.83	1824.37	190384.25	668896.56	16434.10	
AEZ4	64931.06	188750.91	26263.92	125355.00	184610.78	1972.18	236235.67	828119.50	30520.65	
AEZ5	152295.12	165305.42	9033.02	122840.49	190275.94	2996.64	259542.78	902289.44	26842.60	
AEZ6	339716.09	54816.60	4113.99	136598.34	98362.40	3871.14	371389.38	1008827.94	30220.06	
AEZ7	8699.00	186182.55	483962.00	103135.23	759016.78	4380.51	429032.88	1974408.88	2856.97	
AEZ8	25573.48	130645.14	72681.83	182671.56	273203.09	5679.49	294637.41	965092.00	3781.65	
AEZ9	95863.79	87963.52	47098.09	173573.83	117137.34	5902.01	226083.73	763642.31	2612.37	
AEZ10	231230.83	58136.07	32001.07	228214.50	117277.01	14034.57	245123.62	926017.69	3331.23	
AEZ11	132749.08	36259.89	13443.31	107017.04	72203.89	9985.93	122888.27	494547.41	8821.41	
AEZ12	148136.67	49226.12	2903.87	95872.81	113211.05	7391.83	168611.97	583354.12	15362.91	
AEZ13	13291.09	43104.68	15982.81	29135.98	145489.50	278.95	53618.25	304891.25	110.68	
AEZ14	200944.56	89155.51	3094.43	16597.48	97490.17	494.72	244674.50	632451.38	109.58	
AEZ15	242743.47	57460.41	1673.36	31692.38	62462.44	1146.59	218235.50	616414.12	452.41	
AEZ16	13472.57	12304.34	2363.53	2380.94	16928.22	95.36	3030.18	50575.04	107.92	
AEZ17	1893.54	1672.16		32.22	861.61	1.35	248.43	4709.30	1.38	
AEZ18				1.26	158.97	0.21	594.89	755.32		
tot	1690826.00	1415133.25	955970.00	1562904.00	2703738.00	61693.88	3361690.75	11752966.00	142587.47	



The second view over global maps on

and the land categories





### 3.15 Climate Change Damages

The implementation here follows Roson & Sartori 2016 which develop a data base with Climate Change (CC) impacts for the GTAP V9 Data Base. An application to version 10 seems straightforward as the sole relevant difference seems the addition of Tajikistan in version 10 for which a proxy region from Version 9 can be defined. The implementation follows rather strictly what is proposed by Roson & Sartori 2016, exemptions are motivated below.

#### 3.15.1 Data input

##### Raw data processing

The basis for the current implementation is the supplementary material from Roson & Sartori 2016. In order to allow for a smoother integration into GAMS, the data were copy-and-pasted in a different presentation to an EXCEL work book and afterwards stored in GDX container called “ccDamage.gdx”.

The following screenshot shows how the data are organized. For each country, six different impact categories are present, which in case of labour productivity (prodLab) and changes in the energy consumption of household (Energy\_HSLD) are broken down further to sub-categories. In case of sea-level rise impacts at the same temperature increase, a differentiation between 2050 and 2100 is found in the raw data. To ease processing, these two years have been introduced for all impact categories. The last dimension depicts the temperature deltas.

Plane index (empty)								
				+1°C	+2°C	+3°C	+4°C	+5°C
AUS	prodLab	AGR	2050			-0.0109142280147764	-0.0285779350094674	-0.0554673197099693
			2100			-0.0109142280147764	-0.0285779350094674	-0.0554673197099693
		MAN	2050				-0.000211522238644735	-0.00867679339769511
			2100				-0.000211522238644735	-0.00867679339769511
	landStock	<empty>	2050	-0.00069213547798508	-0.00114231521645835	-0.00159249495493162	-0.00204267469340489	-0.00249285443187817
			2100	-0.00138427095597016	-0.0022846304329167	-0.00318498990986325	-0.00408534938680979	-0.00498570886375633
	Energy_HHSLD	Electr.	2050	-2.00960622378688E-5	-3.15989420083961E-5	-3.51893139802473E-5	-3.14931270413276E-5	-2.10864874671552E-5
			2100	-2.00960622378688E-5	-3.15989420083961E-5	-3.51893139802473E-5	-3.14931270413276E-5	-2.10864874671552E-5
		Gas	2050	-0.00026309919323963	-0.000524470808453779	-0.000784470779228347	-0.00104339298101878	-0.00130147806289453
			2100	-0.00026309919323963	-0.000524470808453779	-0.000784470779228347	-0.00104339298101878	-0.00130147806289453
		Oil.P.	2050	-0.0403849240953697	-0.0796601317361653	-0.117871475953865	-0.15506227794245	-0.191273502086637
			2100	-0.0403849240953697	-0.0796601317361653	-0.117871475953865	-0.15506227794245	-0.191273502086637
	Yield	<empty>	2050	-0.019051744	-0.040090688	-0.063116832	-0.088130176	-0.11513072
			2100	-0.019051744	-0.040090688	-0.063116832	-0.088130176	-0.11513072
	Tourism		2050	-0.0013958180893753	-0.0030919127591279	-0.00502925661768942	-0.00714971781765354	-0.00940100988336589
			2100	-0.0013958180893753	-0.0030919127591279	-0.00502925661768942	-0.00714971781765354	-0.00940100988336589
	Health		2050	-0.00148875971692301	-0.00148875971692301	-0.00148875971692301	-0.00148875971692301	-0.00148875971692301
			2100	-0.00148875971692301	-0.00148875971692301	-0.00148875971692301	-0.00148875971692301	-0.00148875971692301

#### 3.15.2 Aggregation

These raw data are read in “gtapAgg.gms” which prepares the data base for benchmarking of CGEBox. If the file with the damages is found on disk, sets are defined from the labels in the parameter above and the parameter is inputted:



```

$iftheni.ccDamages exist 'cc_Damag/cc_damages.gdx'

*
* --- read data from Roson-Sartori from GDX container
*
set ccShocks;
set ccSubShocks;
set temp;
set ccShockYears;

$$GDXIN 'cc_damag/cc_damages.gdx'
  $$load ccShocks      < p_ccDamages.dim2
  $$load ccSubShocks   < p_ccDamages.dim3
  $$load ccShockYears  < p_ccDamages.dim4
  $$load temp          < p_ccDamages.dim5

  parameter p_ccDamages(*,ccShocks,ccSubShocks,ccShockYears,temp);
  $$load p_ccDamages
$$GDXIN

```

Afterwards, the raw impact data are aggregated from single GTAP region (V9, 140 regions) to the model regions defined by the user. The following weights are used:

1. For labour productivity and health impacts, population:

```

set popWeighted(ccShocks) / prodLab,Health /;

*
* --- labour productivity and health impacts weighted with population
*
p_ccDamages(rNat,ccShocks,ccSubShocks,ccShockYears,temp) $ popWeighted(ccShocks)
= sum( rr(rdat,rnat), p_ccDamages(rDat,ccShocks,ccSubShocks,ccShockYears,temp) * popGtap(rDat))
/ pop(rnat);

```

2. Yield impacts are aggregated using crop land acreages:

```

*
* --- yield impacts weighted with crop land acreage
*
p_ccDamages(rNat,"yield",ccSubShocks,ccShockYears,temp) $ sum(aez,landUse(rNat,aez,"cropLand","ha"))
= sum( rr(rdat,rnat), p_ccDamages(rDat,"yield",ccSubShocks,ccShockYears,temp)
* sum(aez,landUse(rDat,aez,"cropLand","ha")) / sum(aez,landUse(rNat,aez,"cropLand","ha")) );

```

3. Land stock losses due to sea level rise are weighted with the sum of land-covers reported in the AEZ data base:

```

*
* --- landstock losses weighted with total area (including unmanaged land)
*
p_ccDamages(rNat,"landStock",ccSubShocks,ccShockYears,temp) $ sum(aez,land_cover),landUse(rNat,aez,land_cover,"ha"))
= sum( rr(rdat,rnat), p_ccDamages(rDat,"landStock",ccSubShocks,ccShockYears,temp)
* sum( (aez,land_cover),landUse(rDat,aez,land_cover,"ha"))
/ sum( (aez,land_cover),landUse(rNat,aez,land_cover,"ha")) );

```

4. Tourism impacts are weighted with the production value of the “hotels and restaurants” sector in GTAP:

```

*
* --- tourism (gdp% effect) weighted with total production value of hotel and restaurants
*
p_ccDamages(rNat,ccShocks,ccSubShocks,ccShockYears,temp) $ sum(rr(rDat,rNat),tVom("hht",rDat))
= sum( rr(rdat,rnat), p_ccDamages(rDat,ccShocks,ccSubShocks,ccShockYears,temp)
* tVom("hht",rDat))
/ sum(rr(rDat,rNat),tVom("hht",rDat));

```

5. The energy demand changes for the private household are weighted with the household demand at agent prices for the related GTAP products:

```

*
* --- energy effects weighed with energy demands at agent prices
*
p_ccDamages(rNat,"energy_hhsld","gas",ccShockYears,temp) $ sum(rr(rDat,rNat), vdpaGtap("gas",rDat)+vipaGtap("gas",rDat))
= sum( rr(rdat,rnat), p_ccDamages(rDat,"Energy_hhsld","gas",ccShockYears,temp)
      * (vdpaGtap("gas",rDat)+vipaGtap("gas",rDat)))
/sum(rr(rDat,rNat), vdpaGtap("gas",rDat)+vipaGtap("gas",rDat));

p_ccDamages(rNat,"energy_hhsld","oil.P.",ccShockYears,temp)
$ sum(rr(rDat,rNat), vdpaGtap("oil",rDat)+vipaGtap("oil",rDat)+vdpaGtap("p_c",rDat)+vipaGtap("p_c",rDat))
= sum( rr(rdat,rnat), p_ccDamages(rDat,"Energy_hhsld","oil.P.",ccShockYears,temp)
      * (vdpaGtap("oil",rDat)+vipaGtap("oil",rDat)+vdpaGtap("p_c",rDat)+vipaGtap("p_c",rDat)))
/sum(rr(rDat,rNat), vdpaGtap("oil",rDat)+vipaGtap("oil",rDat)+vdpaGtap("p_c",rDat)+vipaGtap("p_c",rDat));

p_ccDamages(rNat,"energy_hhsld","Electr.",ccShockYears,temp) $ sum(rr(rDat,rNat), vdpaGtap("oil",rDat)+vipaGtap("oil",rDat)
= sum( rr(rdat,rnat), p_ccDamages(rDat,"Energy_hhsld","Electr.",ccShockYears,temp)
      * (vdpaGtap("ely",rDat)+vipaGtap("ely",rDat)))
/sum(rr(rDat,rNat), vdpaGtap("ely",rDat)+vipaGtap("ely",rDat));

$endif.ccDamages

```

The resulting aggregated (and raw) CC impacts are then stored in the GDX container along with other data used for benchmark in a GDX container.

### 3.15.3 Graphical User Interface

The CC damage modules can be added to other modules via the following control:

Modules and parameters			
myGTAP	GTAP_AGR	GTAP_AEZ	GTAP_E
CO2_Emissions	Non_CO2_Emissions	Air_Pollution	MTOE
Aggregate_GovToFinalDemand	Aggregate_Armington	Aggregate_firm_demand	GTAP_Melitz
ARM_LES_CES	SPE	GTAP_MRIO	TariffLines
CapVintages	GTAP_NUTS2	Parameters	Nestings
G-RDEM	GMIG	CC_Damages	

The interface allows to selected which of the six impact pathways are used

**Impacts**

Labour productivity ☒

Human health ☒

Available land ☒

Crop yields ☒

Household energy demand ☒

Tourism ☒

If the model is run in comparative-static mode, the temperature impact is defined by setting an increase for single shock:

**Temperate increase**

Shock [delta Degree Celsius]

For recursive-dynamic runs, three temp delta are given, the (already existing) base year increase, and the increase until 2050 and until 2101. There is difference in subtracting for the 2050 and 2100

number the base year effect. The chosen implementation should make it easier to input data from other sources:

Temperate increase	
Base year [delta Degree Celsius]	0.6
2050 [delta Degree Celsius]	1.0
2100 [delta Degree Celsius]	1.5

### 3.15.4 Shock definition

Currently, the code implementation does not yet support a combined use with G-RDEM. For baseline generation with G-RDEM, introducing the CC shocks makes limited sense as G-RDEM takes real GDP changes as given. Combining the CC-Damages with the information from the IAASA repository of GDP changes would assume that these GDP changes reflect these CC-Damages. It is therefore potentially more useful to first develop a baseline without CC damages in G-RDEM and next simulate the impact of CC-Damage against this baseline. This is the next step in code development. Currently, the code throws an error if G-RDEM is switched on.

#### Defining impacts for a specific year and temperature change

Identical to the data base processing step, first, the aggregated CC damages are inputted:

```
parameter p_ccDamages(*,*,*,*,*)
          p_ccDamagesApplied(*,*,*,*);

set ccShocks;
set ccSubShocks;
set temp;
set ccShockYears;
$$GDXIN "%resdir%/build/%dataset%.gdx"
  $$load ccShocks      < p_ccDamages.dim2
  $$load ccSubShocks   < p_ccDamages.dim3
  $$load ccShockYears  < p_ccDamages.dim4
  $$load temp          < p_ccDamages.dim5
  $$load p_ccDamages
$$GDXIN
```

Based on the information on the interface, we define the temperature increase in each year:

```
$$iftheni.recDyn "%dynMode%"=="Recursive dynamic"

  p_tempIncrease(t) $ (t.pos le 2050-%baseYear%)
    = %tempIncreaseBase% + (%tempIncrease2050%-%tempIncreaseBase%)/(2050-%baseYear%-1)*(t.pos-1);

  p_tempIncrease(t) $ (t.pos gt 2050-%baseYear%)
    = %tempIncrease2050%
    + (%tempIncrease2100%-%tempIncrease2050%)/50*(t.pos+%baseYear%-2050);

  t_ccShockYears(t,"2050") $ (t.pos le 2050-%baseYear%) = yes;
  t_ccShockYears(t,"2100") $ (t.pos gt 2050-%baseYear%) = yes;

$$else.recDyn

  p_tempIncrease(tRun) $ (not sameas(tRun,"%t0%"))
    = %tempIncreaseShock%;

  t_ccShockYears(t,"2050") $ (%tempIncreaseBase% le 2050) = yes;
  t_ccShockYears(t,"2100") $ (%tempIncreaseBase% ge 2050) = yes;

$$endif.recDyn
```

For the recursive-dynamic case, we consider the years between the base year and the year 2050 and the years between 2050 and 2100 differently, reflecting that the user can define different temperature changes on the interface for these two years. For the years up to 2050, the difference between the delta in 2050 and the already present temperature increase in the base year is distributed in equal steps over the simulation period up to 2050. Any remaining simulation years beyond 2050 receive a 1/50 of the difference in the increase in 2100 and 2050.

In case of a comparative static run, the inputted temperature delta is stored for the shock.

The above simple mapping for all years about 2050 to the 2100 sea level rise could be improved further with an additional interpolation.

As damages are reported for 1 degree deltas in temperatures, we interpolate linearly between the lower and upper deltas for a user inputted temperature delta:

```
*
* --- interpolate between two temperatures where shocks are given (e.g. 0.4 for 2.4, and the range 2+ and 3+)
*
p_interPolateTemp(tRun) = p_tempIncrease(tRun) - floor(p_tempIncrease(tRun));
*
* --- lower and higher temperature set element (e.g. 2+ and 3+ for a 2.4 increase)
*
tempIncrease(tRun,"lo",temp) $ ((p_tempIncrease(tRun) ge p_dtemp(temp)) $ (p_tempIncrease(tRun) lt p_dtemp(temp+1))) = Yes;
tempIncrease(tRun,"hi",temp) $ ((p_tempIncrease(tRun) lt p_dtemp(temp)) $ (p_tempIncrease(tRun) gt p_dtemp(temp-1))) = Yes;
```

Based on this information, the damages for the different years are mapped to a four dimensional parameter which is subsequently used to define the shocks:

```
*
* --- calculate damages for different years
*
p_ccDamagesApplied(rNat,ccShocks,ccSubShocks,tRun)
= sum(t_ccShockYears(tRun,ccShockYears),
      sum( tempIncrease(tRun,"lo",temp), p_ccDamages(rNat,ccShocks,ccSubShocks,ccShockYears,temp))
      + p_interPolateTemp(tRun)
      * [ sum( tempIncrease(tRun,"hi",temp), p_ccDamages(rNat,ccShocks,ccSubShocks,ccShockYears,temp))
          - sum( tempIncrease(tRun,"lo",temp), p_ccDamages(rNat,ccShocks,ccSubShocks,ccShockYears,temp)) ]
      );
```

## Mapping the impacts into shocks

### *Labour productivity*

For labour productivity, impacts are available for three broad sectors. These are linked to the detailed sectors in the model based on a cross-set (partly shown below):

```
set globSecD_iOri(globSecD,iOri) "Mapping of three broad sectors to original sectors in data base" /

AGR.    (set.icrops)
AGR.    ctl        "Cattle,sheep,goats,horses"
AGR.    oap        "Animal products nec"
AGR.    rmk        "Raw milk"
AGR.    wol        "Wool, silk-worm cocoons"
AGR.    frs        "Forestry"
AGR.    fsh        "Fishing"
MAN.    coa        "Coal"
MAN.    oil        "Oil"
MAN.    gas        "Gas"
$$ifthene.gtap10 %GtapNO%>9
MAN.    oxt        "Minerals nec"
$$else.gtap10
MAN.    omn        "Minerals nec"
$$endif.gtap10
MAN.    cmt        "Meat: cattle,sheep,goats,horse"
MAN.    omt        "Meat products nec"
MAN.    vol        "Vegetable oils and fats"
```

The shock changes the labour productivity shifter *lambdaf*:

```

$iftheni.labProd "%CC_damage_labProd%"=="on"

    --- labour productivity shifts mapped from three broad sectors
        to individual ones, all labour categories

    lambdaf.fx(rs,l,a,tSim) = 1 + %rr%sum( (globSecD_iOri(globSecD,iOri),mapa(iOri,a)),
                                            p_ccDamagesApplied(%rPrice%,"prodLab",globSecD,tSim))
                                            /sum( (globSecD_iOri(globSecD,iOri),mapa(iOri,a)),1);

```

The set *l* relates to all labour categories in the benchmarking data set, *rs* for the (current) regions (nations and sub-regions thereof), *a* are the production activities and *tSim* is the current simulation year. For a comparative-static experiment, this refers to the label “*shock*”. The %rr% macro either is empty or depicts rr(rs,rsNat) in case that sub-national units are present in the model. The set mapa links original GTAP sectors to the activities in the model according to the make matrix

### Health impacts

Health impacts (if active) are added on top of these labour productivity shocks:

```

    lambdaf.fx(rs,l,a,tSim) = 1;
$endif.labProd
$iftheni.health "%CC_damage_health%"=="on"
*
*    --- add health effect on labour productivity
*
    lambdaf.fx(rs,l,a,tSim) = lambdaf.l(rs,l,a,tSim)
        + %rr% p_ccDamagesApplied(%rPrice%,"Health","",tSim)

$endif.health

```

### Tourism

Roson and Sartori (2016) develop the tourism impact based on estimates in changes of people travelling to specific destination which suggests in change in foreign income contributions. We therefore map there estimates in %GDP changes in changes in international transfers which enter the BOT==BOP balance of CGEBox. A loss of tourism revenues therefore implies that the net imports need to be reduced (or net exports to be expanded), interacting with the global bank mechanism of the GAP model:

```

$iftheni.tourism "%CC_damage_tourism%"=="on"
t
t    -- change in income inflow from abroad
t
    p_capTrans(rsNat,tSim) = rGdpmp(rsNat,tSim) * p_ccDamagesApplied(rsNat,"Tourism","",tSim);

$endif.tourism

```

### Energy demand changes for households

These are implemented as shifts of the CES share parameters in the energy nests which requires the nesting of the GTAP-E module. The code therefore aborts if these are not active.



```

$iftheni.hed "%CC_damage_hed%"=="on"
*
* --- change in household energy demand
*
$ifnot "%modulesGTAP_E%"=="on" $abort "CC impact on energy demand of household require nestings from GTAP-E, please switch on"

alphaa.fx(rsNat,"gas-c",h,tSim) = alphaa(rsNat,"gas-c",h,tSim) * (1+p_ccDamagesApplied(rsNat,"energy_hhsltd","gas",tSim));
alphaa.fx(rsNat,"gdt-c",h,tSim) = alphaa(rsNat,"gdt-c",h,tSim) * (1+p_ccDamagesApplied(rsNat,"energy_hhsltd","gas",tSim));
alphaa.fx(rsNat,"oil-c",h,tSim) = alphaa(rsNat,"oil-c",h,tSim) * (1+p_ccDamagesApplied(rsNat,"energy_hhsltd","oil.p.",tSim));
alphaa.fx(rsNat,"p_c-c",h,tSim) = alphaa(rsNat,"p_c-c",h,tSim) * (1+p_ccDamagesApplied(rsNat,"energy_hhsltd","oil.p.",tSim));
alphaa.fx(rsNat,ely,h,tSim) = alphaa(rsNat,ely,h,tSim) * (1+p_ccDamagesApplied(rsNat,"energy_hhsltd","electr.",tSim));

$endif.hed

```

In the above,  $h$  are different households and  $\alpha_{haa}$  the cost shares in the CES-subnests in the demand system.  $Ely$  is the set of electricity products, which comprises either one element in case of the standard GTAP Data Base or multiple electricity carriers if the GTAP-Power Data Base is used.

### Crop yield changes

map crop yield changes into a change in tfp:

```

$iftheni.yields "%CC_damage_yields%"=="on"
*
* -- change in tfp of crops
*
axp.fx(rs,crop_a,tSim) = (1+ %rr%p_ccDamagesApplied(%rPrice%,"yield","",tSim));

$endif.yields

```

Where  $axp$  is the production shifter at the top of the production structure.

### Land loss due to sea level rise

This is implemented as a shifter of total land stock in economic use. Land in GTAP is only used in agriculture and forestry, the shock does not hence not affect the production possibility set of manufacturing or sectors directly.

The actual implementation differs depending on whether the AEZ module is switched on or not, and on detail of how the AEZ module is used:

```

*
* --- change in economic land stock
*
$iftheni.aez "%modulesGTAP_AEZ%"=="on"

p_xAezNestBase(rs,aez,"top",tSim)
= p_xAezNestBase(rs,aez,"top",t0) * (1+ %rr%p_ccDamagesApplied(%rPrice%,"landstock","",tSim));

$iftheni.landSupply "%landSupplyElas%"=="on"

v_topLandChange.fx(rs,aez,tSim) = xAezNest(rs,aez,"top",tSim)
* %rr%p_ccDamagesApplied(%rPrice%,"landstock","",tSim);

$endif.landSupply
$else.aez

p_aft(rs,lnd,tSim) = p_aft(rs,lnd,tSim) * (1+ %rr%p_ccDamagesApplied(%rprice%,"landstock","",tSim));
xft.fx(rs,lnd,tSim) $ (xft.range(rs,lnd,tSim) eq 0) = p_aft(rs,lnd,tSim);

$endif.aez
$endif.landstock

```

## 3.15.5 References

Roson, R., & Sartori, M. (2016). Estimation of climate change damage functions for 140 regions in the GTAP9 database. *Policy Research Working Paper 7728, World Bank, Available at:*

<https://openknowledge.worldbank.org/bitstream/handle/10986/24643/Estimation0of00n0the0GTAP90database.pdf?sequence=1>

## 4. Data base generation

During the data base generation, the user decides:

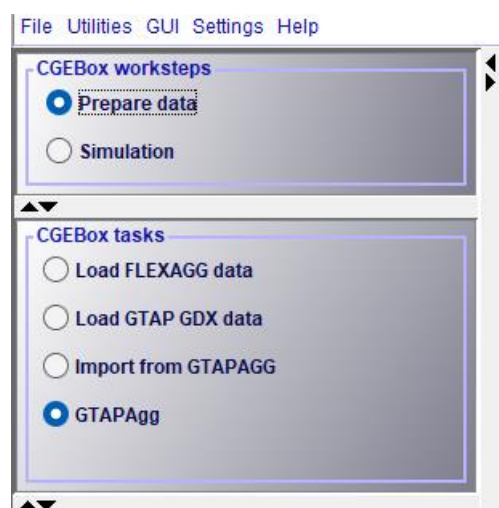
- which data base to start from
- which regional and sector resolution to use, i.e. how to aggregate the data base
- if tiny costs shares are to be removed (recommended);
- which additional data to introduce (sub-regional data for non-EU countries, MRIO split factors);
- if specific sectors and products are subject to a further split, using user information
- If the resulting data should be stored again as a data base which can be subject to the same steps

Generating benchmark data sets requires selecting the workstep” Prepare data.” First, data from the GTAP website must be downloaded and imported by the task “Load GTAP data”. Thanks to harmonization of the GTAP Data since Version 10, this streamlined process to integrate various data sources into a CGEBox installation is now available as documented next. It feeds into the task “GTAPagg” which is run afterwards. The task “Import from GTAPAGG” is available for legacy reasons, but not recommended for production runs any longer as not all functionality is fully supported.

The process consists of two steps:

- 1) Installing the data sets at full regional and sector/product differentiation from the GTAP center
- 2) Deriving from there a model data base with the desired content (see above)

The first version of CGEBox were based on import from GTAPAGG. While this option is still available, it is not longer recommended and will be dropped as it is cannot handle additional sets (AEZ, Non-Co2 emissions, etc.),



### 4.1 Direct input from GDX version of GTAP Data Base

#### 4.1.1 Background

The GTAP Centre releases since 2022 its data bases directly in GDX format and the GTAPAGG utility in CGEBox is designed to work with the data structures used. This version is currently maintained and subject to bug fixes and improvements.



## 4.1 Background

So far, the GTAP Data Base was distributed as zipped archives comprising HAR files, a GEMPACK specific binary data format. Using the data in CGEBox and other GAMS based models required after unzipping the archive first a data conversion from HAR to GDX before further processing steps. Since 2022, the GTAP center also distribute almost all data also as GDX containers (access to this page requires a GTAP license):

https://www.gtap.agecon.purdue.edu/databases/download.asp?

<b><u>Bilateral TS Trade Data</u></b>	<a href="#">Download</a>	N/A	N/A	N/A	N/A
---------------------------------------	--------------------------	-----	-----	-----	-----

**GDX files**  
 These files contain the fully disaggregated GTAP Data Base. The format is consistent with GTAP v7 Model. The downloads consist of the following folder structure, 'flavour\year\'', for example: 'GTAP 10a Data Base\2014' contains the GTAP Data Base for reference year 2014. Air pollution and NCO2 files are contained in 'gsdtpolut.har' within the GTAP 10a Data Base.

	All Reference Years	2014	2011	2007	2004
<b>GTAP 10a Data Base</b>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>
<b><u>GTAP Energy Subsidies</u></b>	N/A	<a href="#">Download</a>	N/A	N/A	N/A
<b><u>GTAP Land Use and Cover Data Base</u></b>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>
<b><u>GTAP-MRIO</u></b>	N/A	<a href="#">Download</a>	N/A	N/A	N/A
<b><u>GTAP-POWER</u></b>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>
<b><u>GTAP-E</u></b>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>
<b><u>GTAP-APT Data Base</u></b>	N/A	<a href="#">Download</a>	N/A	N/A	N/A

CGEBox supports now the direct imports of the core data base (first line), of the Land use (AEZ) and the MRIO data. Hopefully, the GMIG data will also become available as GDX container soon. The core data also comprises the CO2, non-Co2, air pollution and energy accounting data.

## 4.2 Importing the data into CGEBox

In order to import the data into CGEBox, first generate a new folder under the data directory with a senseful name, such “GTAPV10\_2014” with, for instance, the Windows explorer. If, for instance, your cgebox installation is hosted on “D:/CGEBOX”, generate a new folder in “D:/CGEBOX/DATA”.

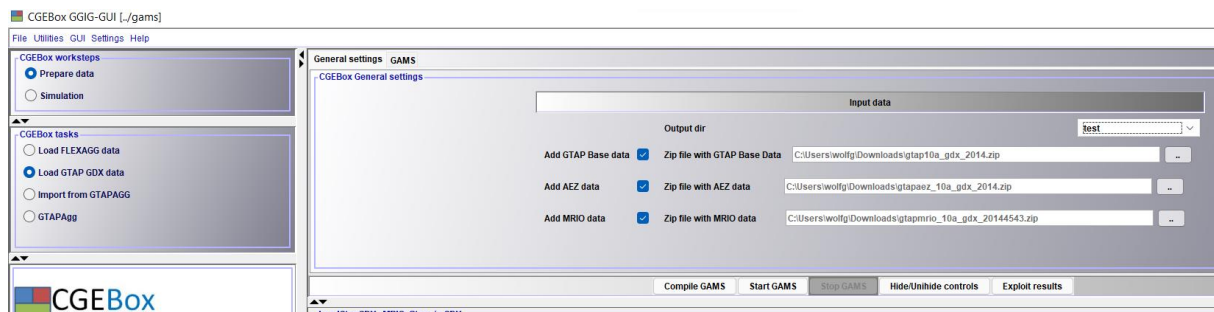
Next, download the core and AEZ Data from the GTAP website which are always processed by the data driver of CGEBox, choosing the single year edition:

## Importing the data into CGEBox

Downloads					
	<input type="checkbox"/> Name	Änderungsdatum	Typ	Größe	
	Heute (3)				
	gtap10a_gdx_2014.zip	08.03.2022 05:24	ZIP-komprimierter...	127.678 KB	
	gtapaez_10a_gdx_2014.zip	08.03.2022 14:39	ZIP-komprimierter...	116.499 KB	

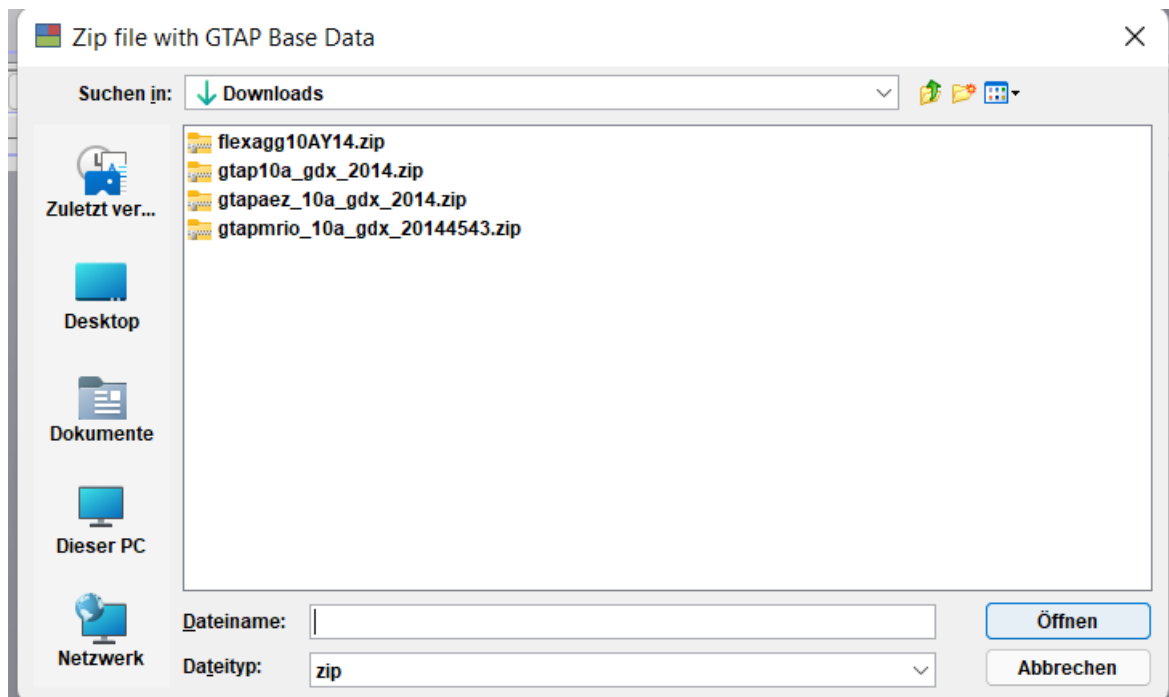
Remember into which directory you have downloaded the data, this location is needed for the next step, the import into CGEBox. You don't need to unzip the downloaded containers or move them to another location for the import step.

Next, open the CGEBox GUI, select the work step "Prepare data" and the task "Load GTAP GDX data":



As seen from the interface, you need to select in the first line the sub-directory in the data folder into which you import, here "test". The folder should have been generated by you.

You can choose which data parts (base data, AEZ, MIO) to add based on the three check boxes. Pressing one of the button "." to the right of the three text fields opens a file dialog based on which you can select the matching download zip container:

















Afterwards, press “Start GAMS” button. The GAMS script will unzip the selected downloaded ZIP archives into the scratch directory, for instance:

```
--- call gmsunzip -j -o "C:\Users\wolfg\Downloads\gtapaez_10a_gdx_2014.zip" -d "C:\smdir\aez"
Archive:  C:/Users/wolfg/Downloads/gtapaez_10a_gdx_2014.zip
  inflating: C:/smdir/aez/gsdffat.gdx
  inflating: C:/smdir/aez/gsdfemiss.gdx
  inflating: C:/smdir/aez/gsdfpar.gdx
  inflating: C:/smdir/aez/gsdfSAM.gdx
  inflating: C:/smdir/aez/gsdfset.gdx
  inflating: C:/smdir/aez/gsdfshock.gdx
  inflating: C:/smdir/aez/gsdftax.gdx
  inflating: C:/smdir/aez/gsdfview.gdx
  inflating: C:/smdir/aez/gsdfvole.gdx
  inflating: C:/smdir/aez/readme.txt
```

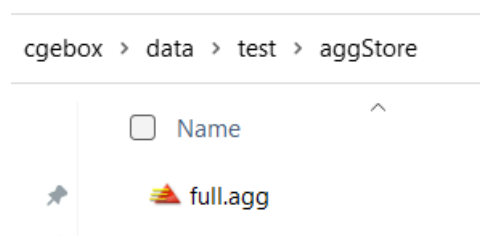
Please make sure that in you have enough disk space in the scratch directory. If no scratch directory is explicitly set, the scratch directory of GAMS (225...) will be used. Note that the MRIO data require >1 GB of disk space for processing.

The process for all three data sources currently supported requires on fast Laptop less than a minute.

You can check if the stepp worked by checking with the Windows explorer or a similar tool the directory chosen for output:

<input type="checkbox"/> Name	Anforderungsdatum	Typ	Größe
<input checked="" type="checkbox"/>  aggStore	08.03.2022 17:11	Dateiordner	
 aez_d.gdx	08.03.2022 17:11	GAMS file	267 KB
 gsdfdat.gdx	18.01.2022 17:34	GAMS file	106.325 KB
 gsdfemiss.gdx	18.01.2022 17:34	GAMS file	867 KB
 gsdfpar.gdx	18.01.2022 17:34	GAMS file	547 KB
 gsdfpolut.gdx	18.01.2022 17:34	GAMS file	3.730 KB
 gsdfSAM.gdx	18.01.2022 17:34	GAMS file	61.325 KB
 gsdfset.gdx	18.01.2022 17:34	GAMS file	9 KB
 gsdfshock.gdx	18.01.2022 17:34	GAMS file	12.147 KB
 gsdftax.gdx	18.01.2022 17:34	GAMS file	12.146 KB
 gsdfview.gdx	18.01.2022 17:34	GAMS file	84.675 KB
 gsdfvole.gdx	18.01.2022 17:34	GAMS file	2.266 KB
 mrio_splitfactor_v10_new.gdx	08.03.2022 17:11	GAMS file	57.149 KB
<input type="checkbox"/>  readme.txt	13.12.2021 20:36	TXT-Datei	1 KB

The AggStore directory should comprise a file with a 1:1 aggregation called “full.agg”:



In the current release of the GDX containers by the center, no long-text for regions, products and factors are available, a point hopefully corrected soon.

### 4.3 GTAP Data base import using zip containers for FLEXAGG

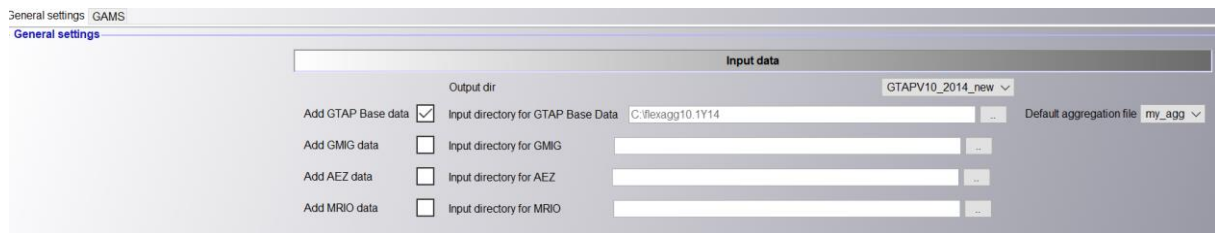
This option will be dropped once all data are released by the GTAP center in GDX format. It delivers the same output as the download of the ZIP archives with the GDX containers.

#### 4.3.1 Set-up and installation of base data

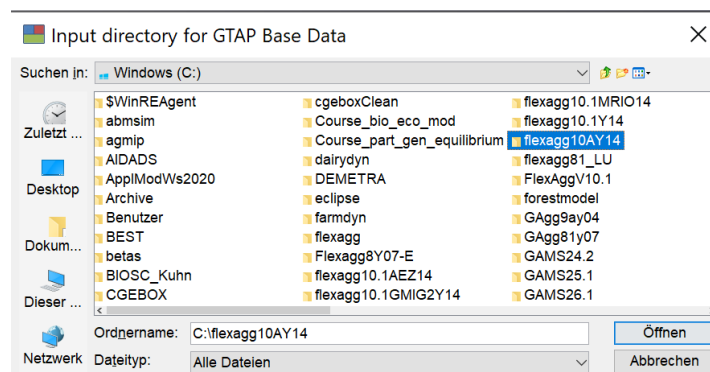
Please perform the following steps:

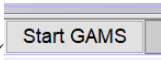
1. Download the desired base version of the GTAP Data Base (GTAP, GTAP-E, GTAP-Power) in FlexAgg format from the GTAP website, single year version. The data are provided as a zip archive. GTAP-AEZ, GMIG and MRIO data are added later and should not be used as the base data version to install.

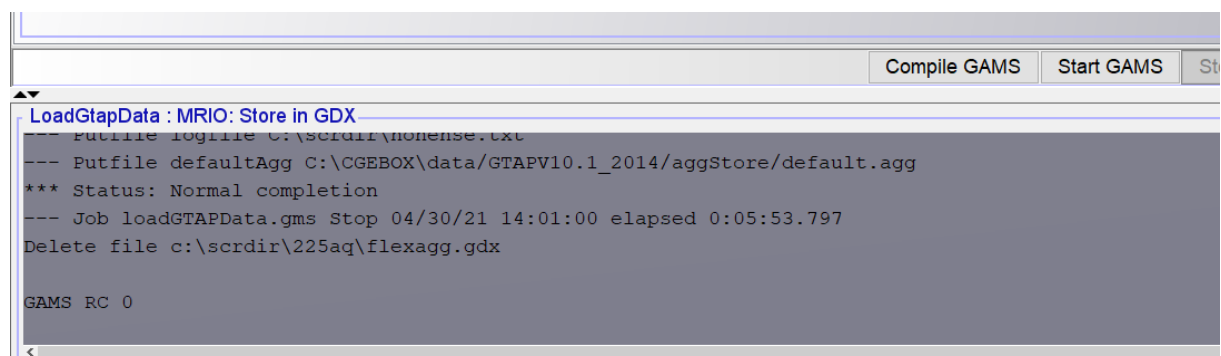
2. Unzip the archive into a folder outside of the CGEBox directory.
3. Generate a sub-folder in the folder “CGEBox/data” with the desired name of the data base to install, for instance, “CGEBox/data/GTAP\_10.1\_POWER\_2014”
4. Open the CGEBox GUI, select as workstep “Prepare data” and as Task “Load GTAP data”



5. Make sure that the “output dir” points to the newly created folder into which you want to install, in the example above this is “GTAP\_10.1\_2014”.
6. Tick the checkbox “Add GTAP Base data”. Use the “..” button to open a file dialog to select the folder where the content of the zip can be found. Access to the folder is only needed during installation.



7. Add the base data by starting the data driver (  ).
8. Check if the process did run without problem. The last or second last line in the output window should be “GAMS RC 0”, as seen below.



The new release will automatically add Non-Co2 and air emission data. If you don't need the data from FLEXAGG in HAR format for other purposes, you might delete the folder generated by unzipping the archive. You might also want to delete the original zip folder.

The driver will use the agg-file “full\_agg.txt” to define a default aggregation file. Another probably formatted aggregation file with long-text information for the original products, sector and factors can be used instead and can be selected with the file dialog

#### 4.3.2 Adding GMIG, AEZ and MRIO split factor

The process is identical to the one described above with the sole extension that the directory with the base data should already be present:

1. Download the additional data in FLEXAGG format for the matching year and data release from the GTAP website.
2. Unzip into a folder outside of CGEBox.
3. Tick the checkbox for the data to add and choose the folder in which you unzipped the archive.
4. Run the data driver.

Note that CGEBox stores selected parts of these three data bases, only. In case of the AEZ data, this refers to the break-down of land rents from regional to the level of the AEZ as well as to land use and land cover data at AEZ level. For GMIG, this relates to transactions depicting bi-lateral labour force use and remittances. The MRIO data at c.i.f., c.i.f. plus tariffs and at agent prices are aggregated from firm-by-firm-region-by-region to intermediate demand-by-region-region data, such that only four MRIO agents result (see Britz and Corong 2021 for detail). All other data provided by the AEZ, GMIG and MRIO versions of the GTAP data base are taken from the chosen base data instead.

You can add multiple sources at once, including the data base. If you don't need the data from FLEXAGG in HAR format for other purposes, you might delete the folders generated by unzipping the archives.

#### 4.4 Deprecated: Input from GTAPagg

The data driver supports two types of input data sets. First, the input derived from the downloaded ZIP containers as described above, and second, output from the GTAPagg utility which is linked to GEMPACK. The GTAPagg utility allows you to build a GTAP data base with a sector and regional aggregation chosen by you (see e.g. <https://www.gtap.agecon.purdue.edu/products/packages.asp>, a training video on you tube <https://www.youtube.com/watch?v=QDBR0KqNuzE&feature=youtu.be> or the first pages of [http://economia.unipv.it/pagp/pagine\\_personali/msassi/QPA/materialQPA/Introduction%20to%20GTAP.pdf](http://economia.unipv.it/pagp/pagine_personali/msassi/QPA/materialQPA/Introduction%20to%20GTAP.pdf)). That data base comprises different data sets stored in HAR (header array) files, a proprietary data format for GEMPACK. Mark Horridge has developed a program called HAR2GDX which converts a HAR file into the proprietary data format GDX used by GAMS. HAR2GDX is part of a GAMS installation and used by us to load output from GTAPagg.

If you generated a data base with GTAPagg intended for use with CGEBox, please copy the zip file generated by GTAPagg to the “data” directory (or store the file directly there from GTAPagg). In order to make the data available to the model, choose the workstep “Prepare data” and the task “Import from GTAPAGG”. The file you copied in the data directory should be available in the dropbox under “Input file from GTAPAGG (\*.zip)” and selected by you.

Important:

1. **Do not rename a zip file** generated by GTAPAGG as the program will not find the correct agg file in that case and data base generation for CGEBOX will fail.
2. For the same reason, do not use the “default.agg” from GTAPGG. If you want to use these aggregation definitions, store them first under a different name before you generate the aggregated data set.

The intermediate GDX and HAR files can be deleted. Keeping them eases debugging in case something went wrong during data processing.

Please note that importing data from GTAPAgg will not deliver additional data such as Non-CO2 emissions or land use data, solely the data comprised in standard GTAP data set.

#### 4.5 GTAPAGG: The data base generation utility of CGEBox

The screenshot below gives an overview on the core options available with data base generation utility of CGEBox. It aggregates the GTAP Data Base (Standard or Power) and additional data (CO2, Non-Co2, air pollution, migration, land use ...) according to user defined aggregation rules, filters out small numbers and allow further steps, such as applying a split or adding sub-national data.

The outcome of the data base utility is one single GDX container stored in the ‘results/build’ directory, under the name aggregation file plus a potential postfix entered by the user. This GDX container along with code and data under version control comprises all necessary data and parameters to perform simulation runs with CGEBox, including its extensions.

Such GDX containers can hence also be exchanged between CGEBox users. For instance, one team member can generate a suitable data base with the GTAPAgg utility and share the container with all others. Installing the GTAP Data Base and other data sets from the GTAP center (AEZ, non-CO2 etc.) is necessary only to build these GDX containers, not for simulation runs.

Please note that the GDX containers comprise the copy-right protected data from the GTAP center for which a license is required. The license for Version 10 states (April 2022): “Licensees are free to distribute small aggregations or RunGTAP applications for joint work or teaching purposes. However, the licensee must obtain permission from the Center for the distribution of large aggregations. A large aggregation is defined as one in which there are more than 10 regions or more than 10



commodities.”<sup>110</sup> Adhering to these requirements, GDX containers up to a 10x10 aggregation might hence be shared with co-authors or students using CGEBox not having the required GTAP license.

#### 4.5.1 Defining your aggregation: The aggregator component

The first step usually consists of defining an aggregation definition file. This is done with the aggregator component found when the tab “Aggregation” is active. The aggregator component is used both when you downloaded zip containers comprising directly GDX containers or HAR files for FLEXAGG. It replaces the GTAPAgg utility form the GTAP center working with HAR-Files. It allows to generate non-diagonal aggregations where the sector and product aggregations differ. It outputs a file in a format similar to the one used for the GTAPAgg utility. The user can also generate such files with a text editor at its own risk as certain consistency checks cannot be easily performed on the aggregation files.

The component is integrated on a tabbed pane in the GUI controls of the data driver under the tab “Aggregation” and comprises itself a tabbed pane with the three tabs “Products“, “Regions”, “Factors” and “Sectors”:

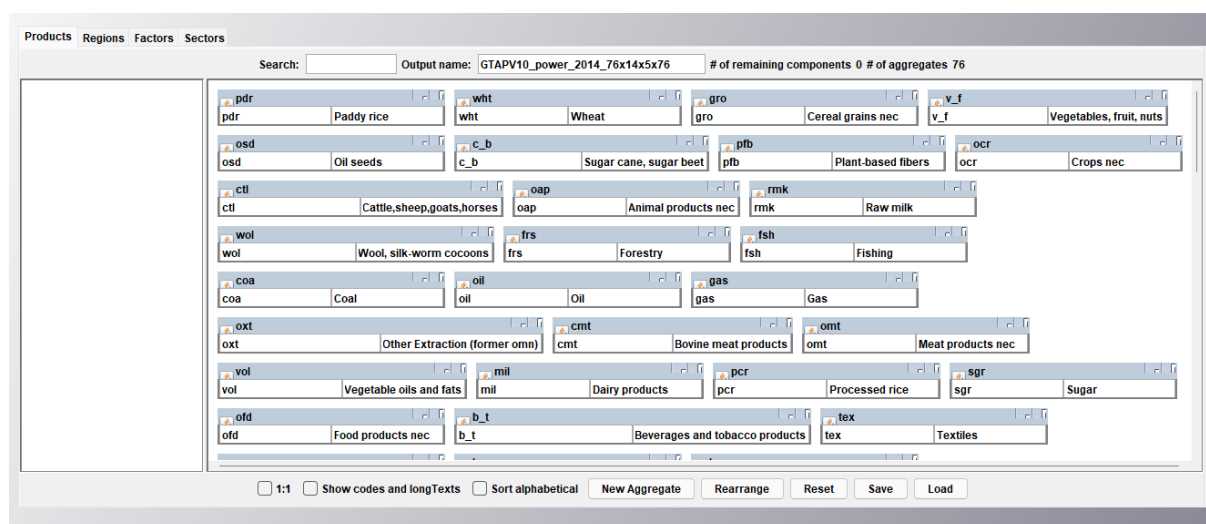


Figure 16: Aggregator component on a tab of the GUI

The aggregation control as the central control shown above in Figure 16 comprises on the LHS a field with the list of unassigned components (here empty) and on the RHS an area where new aggregates can be defined and existing ones are shown. This area is empty in Figure 16 above as the default aggregation rules, loaded at start from “default.agg”, correctly assigns already all components to an aggregate. While the functionality is similar to the user interface of GTAPAgg, the GUI component uses intensively “drag & drop” as a faster option with a more modern Look and Feel.

There are two ways to define aggregates:

1. Using the button “**New aggregate**” (in the control bar at the bottom) will generate a new window on the RHS to host components with a default label - “New” plus a number –as shown in Figure 17 below. The user will overwrite this label to give the aggregate its desired name and potentially also provide a long text in the text field to the right of the label field. That is similar to the GTAPAgg solution. The field below the labels will hold

<sup>110</sup> [https://www.gtap.agecon.purdue.edu/databases/documents/LicenseAgreement\\_v10.pdf](https://www.gtap.agecon.purdue.edu/databases/documents/LicenseAgreement_v10.pdf), visited 30.04.2022

the components, also similar to GTAPAgg. What differs is that each aggregate is visualized as a small window which can be dragged, defining the order of the aggregate, or can be minimized to leave more room to define larger aggregation. In order to populate an aggregate, components are dragged from the LHS into the area below the two text fields (which is empty in the screenshot below).

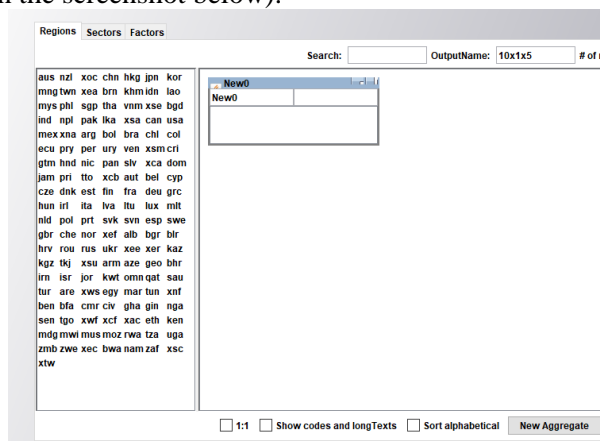


Figure 17: A new empty aggregate generated from “New Aggregate”

- The second option **defines aggregates directly by drag and drop from the LHS or from existing aggregates**. In both cases, a list of components - a single one, one or more sequences etc. – is selected with the mouse (keep the CTRL key of the keyboard pressed to select multiple one) and next jointly dragged to the RHS. If the drop ends in a window of an existing aggregate, the components are added to it. Otherwise, if the drop ends somewhere in the RHS outside of an aggregate window, a new aggregate is defined comprising the selected components:

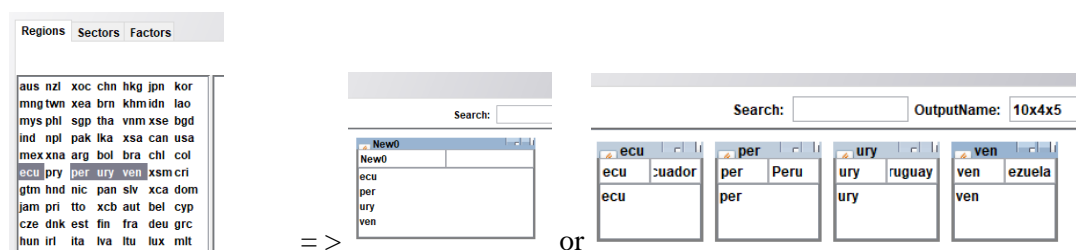


Figure 18: Aggregates defines by drag and drop examples

That behaviour changes if the user has activated the “1:1” option ☒ **1:1**. In that case, (multiple) aggregates are defined from the dragged selection, each comprising a single component, and the label and long-text are automatically taken over from the component, as shown above on the right. The drag & drop approach to generate new aggregates is probably the more frequently used one compared to first generating a new aggregate and next dragging components into it.

In order to generate a diagonal aggregation with the same aggregate definition for products and sectors, only defined product aggregate and keep all single sectors in the RHS window (or press “RESET” to delete all sector aggregate).

Further functionalities embedded in the aggregation tool cover:

- Aggregates can be deleted with the “Delete” button found in their pop-up menu, their components are moved back into to LHS panel. The “Fit” button from the same pop-up menu will adjust the size of the aggregate window such that all components are visible.

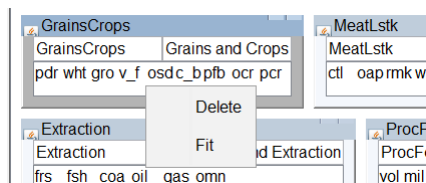


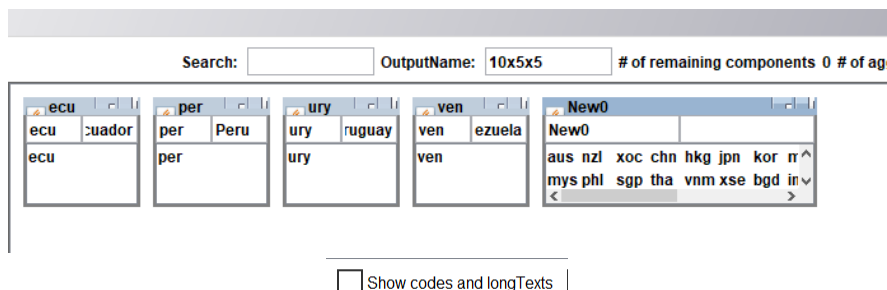
Figure 19: Pop-up menu for aggregate windows

- The “Reset” button deletes all aggregates; consequently, all components are moved back into the LHS. A typical application is to delete all sector aggregates as loaded from “default.agg”, to select next the ☒ option and finally drag in one go all original data base sectors to the RHS to generate an aggregation definition with full sectoral detail. Subsequently, one could still drag some components into others to define aggregates. Note however that empty aggregates are not automatically deleted. That might be unwanted if one has edited the labels and long texts of aggregates already and next moves components around, generating temporary states where aggregates are empty.
- The “Rearrange” button resizes all aggregates again and orders them (from top to down and left to right), that is also the order in which they are outputted and e.g. shown in model listings or later, for instance, in tables of the exploitation tool. One can hence simply drag an aggregate up or more towards the left to move it up in the ordered list.
- The number of aggregates and not allocated components are shown in a pane on top. It also comprises a text field with a generated default name of the output file – it can be overwritten by the user - and offers a search field (see further explanation to the search field below).

Search:  OutputName: 10x10x5 # of remaining components 0 # of aggregates 10

Figure 20: Text fields are search, name of output file and information label

- The window of an aggregate can also be temporarily minimized. The title bar shows the current aggregate name in that case.
- The user can switch between the wrapped lists with labels and lines each with a label – long text pair, as shown in Figure 21 below. 1:1 elements will hide their component area.



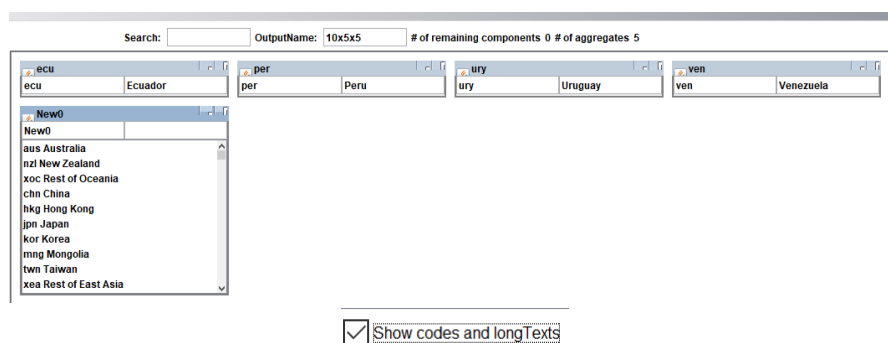


Figure 21: Aggregates shown with code or code and long texts

9. The “search” field allows finding a component, it becomes selected. If more than three characters are entered, the search will switch from a “Start with” to a “contains” type search in the long texts if these are shown.

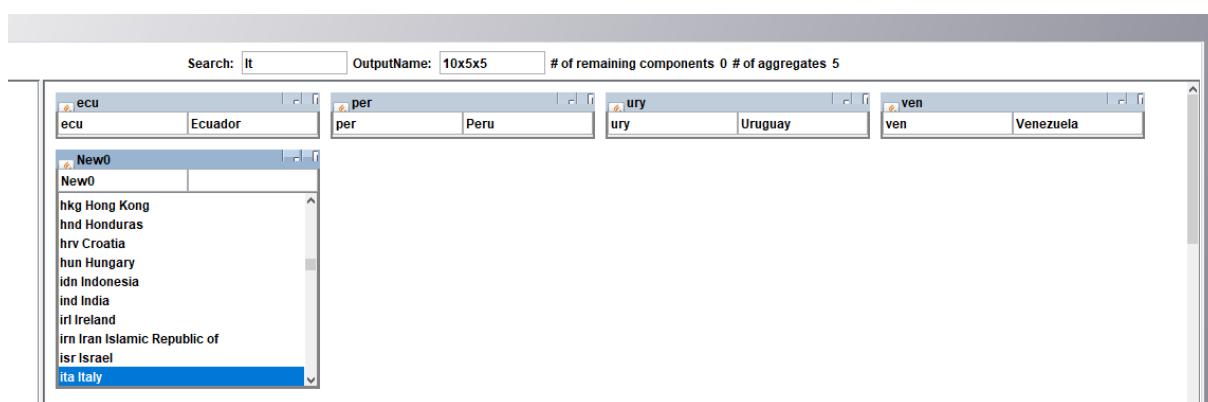


Figure 22: Example of search field

10. The user has the choice to show components either in the order as read from “default.agg” or sorted alphabetically:

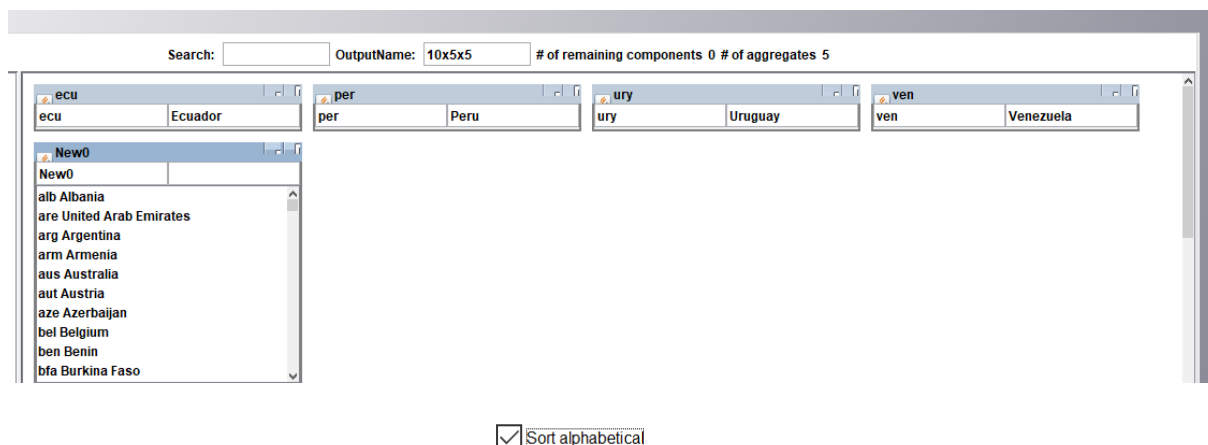


Figure 23: Alphabetical sorting

The “save” button  will save the new definition as an .agg file in the “aggstore” directory.

For each year and data base version, there is a separate folder in the data directory. That folder comprises an “aggStore” folder for matching aggregation definitions.

For data base versions successfully tested with CGEBox, an empty folder with a sub-folder “aggStore” comprising a matching a default aggregation will be committed. The user has to copy the GDX files with the licensed data in the empty folder from the zip downloaded from the GTAP web site.

#### 4.5.2 Aggregation in GAMS

Once you have generated a suitable aggregation definition file and chosen it and you press start, the GAMS script will aggregate the data from the GTAP center. If you are not interested on how this process works, you can skip this section.

The code of this scrip follows the set definitions and builds directly on the code of VDM. In order to ease inspection and debugging, the code does not assign aggregated GTAP symbols directly to the SAM or other matrices or vectors used by CGEBox simulation codes. Instead, it defines first the GTAP symbols at the level of regional / sector / factor aggregates, similar to the output from GTAPAgg, e.g.

```
vdga(i,rNat) = sum( (ii(prod_comm,i),rr(rDat,rNat)) $ vdgaGtap(prod_comm,rDat), vdgaGtap(prod_comm,rDat));  
vdgm(i,rNat) = sum( (ii(prod_comm,i),rr(rDat,rNat)) $ vdgmGtap(prod_comm,rDat), vdgmGtap(prod_comm,rDat));  
vdpa(i,rNat) = sum( (ii(prod_comm,i),rr(rDat,rNat)) $ vdpaGtap(prod_comm,rDat), vdpaGtap(prod_comm,rDat));  
vdpm(i,rNat) = sum( (ii(prod_comm,i),rr(rDat,rNat)) $ vdpmGtap(prod_comm,rDat), vdpmGtap(prod_comm,rDat));
```

Figure 24: Example of aggregation code

This eases debugging potential problem during aggregation.

Before assigning the aggregated base data entries to the SAM and other matrices / vectors, the filter program can be used to remove small entries as discussed in the model documentation. It is based on QP programming and uses a final LP step to ensure quite high feasibility accuracy. Accordingly, besides filtering, small numerical rounding errors as typically found in the original data are further reduced. Additionally, a Generalized RAS is run independent from filtering or not. The code will throw an error if numerical differences between column and rows sum greater 1E5 USD are found and the error in relative terms to the sum is higher than 0.01%.

The code handles in a similar fashion:

- The data from the GTAP data base and related parameters
- The CO2 emission data
- The NON-CO2 emission data
- The air pollution data
- The AEZ related data (here, only data at AEZ level related to land use are stored)

Open is currently how to deal with carbon stock factors and the GMIG data base. All data necessary for a simulation run including the sets are stored into a GDX container in the build directory. The user can also for debugging purposes store all output in a GDX but it is not guaranteed that such a container will work in simulation mode as the code renames some symbols if not all symbols are put into a GDX container. That option requires a user level of “debugger”.

#### 4.5.3 Backward compatibility

Old versions of the GTAP used partially different naming conventions etc. compared to the test data set produced by the center. The GTAP V9 release as available from the GTAP web site for users with a valid license was therefore as a test converted to the new structure:

1. For the parameters, that required some renames, which can be simply done with a GAMS script:

```
*
* --- get the parameters from default.prm and rename to new naming convention
*
parameter esubd, esubm, esubt, esubva, etrae, incpar, subpar, rorflex;
$GDXIN "default_prm.gdx"
  $$load esubd esubm esubt esubva etrae incpar subpar rorflex
$GDXIN
execute_unload "gtap_p.gdx" esubd=esbd, esubm=esbm, esubt=esbt, esubva=esbv, etrae=etre, incpar=incp, subpar=subp, rorflex=rflx;
```

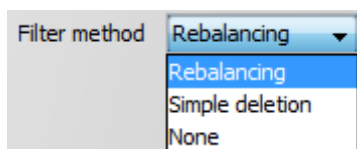
2. The core GTAP data format and names have not changed such that is sufficient to do a HAR2GDX with the new name “GTAP\_d.gdx”.
3. The new structure puts the sets in a separate container. GAMS code can easily accomplish that:

```
*
* --- the sets are in Angel's solution in a separate container:
*
set prod_comm, endw_comm, trad_comm, marg_comm, reg, tartype;
$GDXIN "gtap_d.gdx"
  $$load prod_comm endw_comm trad_comm marg_comm reg tartype
$GDXIN
execute_unload "gtap_s.gdx" prod_comm, endw_comm, trad_comm, marg_comm reg, tartype;
```

4. The structure of the aez data is different. The data had been distributed in two HAR files before, with different names and partly a different order of the indices. The GAMS code to map the old data into the new format looks like:

#### 4.5.4 Data base treatment after aggregation: Filtering or not

During data base loading, you can use a facility to additionally filter out small values, three options are available:



Choosing the option “None” will simply load the data base as it is, i.e., without filtering, which is the way the GEMPACK version operates. The following short section discusses the two other options. The recommended option is “Rebalancing” which filters out small values from the data base.

#### 4.5.5 Simple deletion

With simple deletion, transactions loaded from the HAR file are removed from the data base if they are in absolute terms below the threshold entered on the interface under “Absolute tolerance”. With a value of 1.E-10, that deletion step is skipped. Afterwards, the transactions are formatted into a SAM structure. The resulting SAM is then cleansed with the chosen absolute tolerance. No attempt is made with “simple deletion” to maintain the resulting SAM balanced. That option is mostly maintained as a fall back, in case the more refined rebalancing step normally recommended and discussed next should not work. Please note that there is no guarantee that a global SAM “cleansed” by simple delete will work with the model as such cleansing can lead not only to numerically, but also logically inconsistent data. Imagine the case where a subsidy rate is above 100%: cleansing might delete the tax base such as primary factor use in a sector, but keep the tax flow.

### 4.5.6 Filtering and Rebalancing

#### Background

The “Rebalancing” option uses more advanced tactics to select transactions to delete, and perhaps more importantly, rebalances the resulting SAMs. As with simple deletion, first transactions loaded from the GDX/HAR file which in absolute terms are below the chosen absolute threshold are removed from the data base. With a value of 1.E-10, this preliminary deletion step is skipped. It is generally not recommended to use absolute deletion thresholds above 1.E-6 in combination with rebalancing as the subsequent relative thresholds will anyhow apply more refined rules. If available, it is recommended to use dedicated QCP/LP solvers such as CPLEX for the rebalancing step. For a discussion of the solver choice, see Britz, W. (2021): Comparing Penalty Functions in Balancing and Dis-aggregating Social Accounting Matrices, *Journal of Global Economic Analysis* 6(1): 34-81.

The consecutively applied filtering and rebalancing approach is an extension of the method and code developed by Tom Rutherford for “GTAPinGAMS” (see e.g. <http://www.mpsge.org/gtap6/>). It deletes component of the SAM depending on their shares on specific totals, according to the “Relative tolerance” entered on the interface:

- Domestic and imported intermediate demand of a commodity are dropped relative to its total output
- Private / government / investment domestic respectively import demand of a commodity are dropped relative to total private / government / investment domestic respectively import demand
- Trade flows of a product are dropped if both shares on total exports of that product and its exporter and on imports of that product and its importer are below the relative threshold
- Production is dropped if net production of a commodity, i.e. after intermediate use of that commodity in its own production is deducted, is below the relative threshold with regard to total net production

The filtering process imposes restrictions which should maintain the regional SAMs balanced. Additional constraints ensure that production activities require added value and intermediate inputs, if not already otherwise found in the data base.

#### User options during filtering

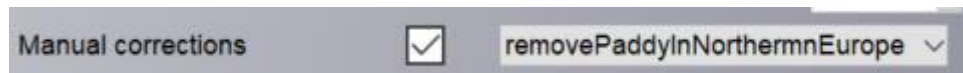
As filtering systematically removes elements from the SAM and the trade matrices, the process implies without further corrections shrinking the economies. During rebalancing, the algorithm can therefore add penalties for deviations from the following aggregate transactions:

Penalty for deviation from	Private consumption	Government consumption	Investments
	Intermediate consumption	Factor income plus bop	GDP

By adding these penalties terms, the non-deleted entries (and thus most important transactions) tend to be scaled upwards. It is generally recommended to use these penalties terms. The code will also scale all non-deleted trade flows to approximately maintain the total volume of international trade and related international transport margins.

Equally, the user can add files with corrections to raw data:

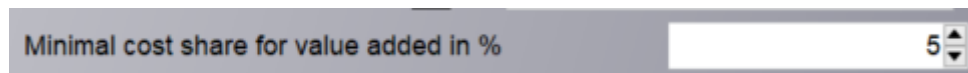




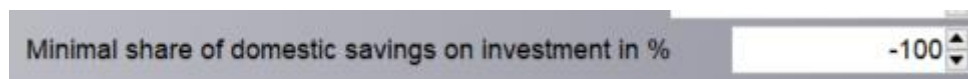
A further option allows removing non-diagonal intermediate crop use (with the exemption of the other crop product ocr which comprises seed) and to control for the maximal seed cost share:



Finally, a minimum cost share for the value added for each activity can be imposed:



And, a minimal share of domestic savings on investments can be imputed:



The absolute and relative thresholds are stepwise enforced. For the first few steps, exponential increases are used, starting with minus half the number of steps. For six steps, to give an example, the first thresholds applied will be the 1.E-3 of the final one, next 1.E-2 and finally 10%. The remaining steps will use equal linear increases between 10% and the desired final ones. Once the final thresholds are active, filtering is still applied several times until no small values are found any longer. The code should ensure that the resulting transactions are still fully consistent with each other, i.e. both the resulting trade matrices and the SAMs are balanced. The changes imposed by filtering and subsequent balancing are stored on the “itrlog” symbol in the GDX container with the final results. Inspecting how the stepwise enforcement of the thresholds impacts on the number of non-zero items can inform on an appropriate level for tolerances to be used.

The SAMs used during filtering are – as in GTAPAGG – defined in Million constant dollars. An absolute threshold of 1.E-6 will hence delete any economic transactions worth a single dollar or less. In SAMs with high regional and sectoral detail, even such tiny transactions might make up to 10% of the non-zero entries. Increasing the threshold for 1000 \$ might remove ¼ or more of all non-empty transactions. Similar results are found from using relative tolerances of 0.001 %.

Thanks to balancing, also rather dis-aggregated versions of the model with large number of sectors and regions can be used. The biggest impact of the filtering is typically on transactions related to bi-lateral trade flows. Here, often 50% or more of the flows account for only 1% of the total value of these transactions. Thus, tiny changes in the relative tolerance can have a considerable impact on the number of deleted transaction, and one might need to experiment with settings in the range around 1.E-1 to 1.E-4 to find a compromise between sparsity and the implied changes on structure of the economy. For very large data sets (e.g. a 1:1 version) filtering thresholds above 1% might be needed to yield reasonable model sizes. The user can additionally define a minimum number of transactions to be kept, which reduces the need to experiment with different thresholds as the filtering process will stop once less than the desired number of transactions is reached. Tests with the model have shown that the model in full resolution of the GTAP 8.1 data base without filtering, i.e. 57 sectors and 134 regions, can be solved in partial trade liberalization scenarios, solution has failed with other shocks on models with more than 400.000 transactions, especially if the global bank mechanism active. A close look at the filtering statistics is recommended, to avoid sharp impacts on the structure of the economy. A more

detailed discussion on the relation between model dis-aggregation, filtering, solution behavior and simulated welfare impacts provide Britz et al. 2015.

### Equations in the re-balancing program

The balancing program is found in “build\filter.gms” draws heavily on the code of Tom Rutherford. For each sector, it guarantees revenue exhaustion for the region currently subject to filtering. The RHS shows the revenues corrected for the output tax rate  $rto$ , where  $vdm\_$  are domestic sales which can adjust during balancing and  $vxm$  are fixed export revenues. The RHS shows the firms demand for imported and domestic intermediates  $vifm\_$  and  $vdfm\_$  as adjusting variables with related given tax rates  $rtfi$  and  $rtfd$  and factor use  $vfm\_$  with related taxes  $rtf$ . The sets are  $td$  for traded products (aliased with  $td1$ ) and  $fe$  for factors.

Two equations ensure that the domestic market use  $vdm\_$  plus fixed exports are produced by the related production activities (according to  $make$  set), and that the production output  $makeM\_$ , considering output taxes  $txo$ , is equal to production cost. The production costs consist of imported  $vifm\_$  and domestic  $vdfm\_$  intermediate demand, plus use taxes  $rtfi$  respectively  $rtfd$ , and factor demand  $vfm\_$  plus factor taxes  $rtf$ :

```

*
* --- revenue exhaustion (output value = value of inputs)
*
e_make(td,rb(rNat)) $ sum(make(a,td),vom(a,rNat)) ..
    sum(make(a,td),makeM_(a,td,rNat)) =E= vdm_(td,rNat) + vxm(td,rNat);

e_profit(a,rb(rNat)) $ vom(a,rNat) ..
    sum(make(a,td),makeM_(a,td,rNat)) * (1-rto(a,rNat))
    =E=
    sum(td1,      vifm_(td1,a,rNat)*(1+rtfi(td1,a,rNat)) $ vifm0(td1,a,rNat)
      + vdfm_(td1,a,rNat)*(1+rtfd(td1,a,rNat)) $ vdfm0(td1,a,rNat))
    + sum(f,      vfm_(f,a,rNat)*(1+rtf(f,a,rNat)) $ vfm0(f,a,rNat));
```

The following three equations ensure that factor use and intermediate input use cannot drop to zero as long as there is production, using the “BigM” approach found in integer programming. And additional equation ensures a minimal cost shares for value added, the threshold `%minCostShareVA%` can be set by the user on the interface:

```

*
* --- "BigM" equations for factor use and intermediate input use
* (i.e. if production -> va,nd )
*
e_fe(f,a,rb(rNat)) $ (      (vfm0(f,a,rNat)*bigM>vom0(a,rNat))
                             and vom(a,rNat) and bigM)..

    sum(make(a,td),makeM_(a,td,rNat)) * (1-rto(a,rNat))
    =L= vfm_(f,a,rNat) * bigM;

e_va(a,rb(rNat)) $ (      (sum(f,vfm0(f,a,rNat)*(1+rtf(f,a,rNat)))*bigM>vom0(a,rNat))
                             and vom(a,rNat) and %minCostShareVA%)..

    sum(f $ vfm0(f,a,rNat), vfm_(f,a,rNat)*(1+rtf(f,a,rNat))) =G=
    sum(make(a,td),makeM_(a,td,rNat)) * (1-rto(a,rNat)) * (%minCostShareVA%/100);

e_nd(a,rb(rNat)) $ (      (sum(tdl, vifm0(tdl,a,rNat)+ vdfm0(tdl,a,rNat))*bigM>vom0(a,rNat))
                             and vom(a,rNat) and bigM)..

    sum(make(a,td),makeM_(a,td,rNat)) * (1-rto(a,rNat))
    =L= sum(tdl,      vifm_(tdl,a,rNat)*(1+rtfi(tdl,a,rNat)) $ vifm0(tdl,a,rNat)
              + vdfm_(tdl,a,rNat)*(1+rtfd(tdl,a,rNat)) $ vdfm0(tdl,a,rNat)) * bigM;

```

The maximal seed cost share %maxSeedCostShare%, inputted via the interfaced, is controlled by the following equation:

```

e_seed(a,rb(rNat)) $ (vom(a,rNat) $ sum(make(a,cropsI),1)) ..

    sum(make(a,cropsI),vdm_(cropsI,rNat) + vxm(cropsI,rNat)) * %maxSeedCostShare%
    =G= sum(make(a,cropsI),      vifm_(cropsI,a,rNat)*(1+rtfi(cropsI,a,rNat)) $ vifm0(cropsI,a,rNat)
              + vdfm_(cropsI,a,rNat)*(1+rtfd(cropsI,a,rNat)) $ vdfm0(cropsI,a,rNat));

```

The domestic market balance ensures that domestic sales vdm\_ are exhausted by government demand vdgm\_, private demand vdpm\_, investment demand vdim\_ and the intermediate demand of the sectors vdfm\_:

```

e_dommkt(td,rb(rNat)) $ vdm(td,rNat) ..

    vdm_(td,rNat) =e= vdgm_(td,rNat) $ vdgm0(td,rNat)
                    + vdpm_(td,rNat) $ vdpm0(td,rNat)
                    + vdim_(td,rNat) $ vdim0(td,rNat)
                    + sum(a $ vdfm0(td,a,rNat), vdfm_(td,a,rNat));

```

A similar equation ensures that the given total imports are exhausted:

```

e_impmt(td,rb(rNat)) $ vim(td,rNat) ..

    vim(td,rNat) =e= vigm_(td,rNat) $ vigm0(td,rNat)
                    + vipm_(td,rNat) $ vipm0(td,rNat)
                    + viim_(td,rNat) $ viim0(td,rNat)
                    + sum(a $ vifm0(td,a,rNat), vifm_(td,a,rNat));

```

Thinning out the SAM without further correction would imply that the overall size of the regional economy shrinks. In order to avoid that, the program defines scaling factor “keepCor” for total intermediate consumption:

```

e_intKeep(rb(rNat)) ..

    sum( (tdl,a) $ vom(a,rNat), vifm_(tdl,a,rNat) $ vifm0(tdl,a,rNat)
          + vdfm_(tdl,a,rNat) $ vdfm0(tdl,a,rNat) )
    / sum( (tdl,a), vifm0(tdl,a,rNat)+vdfm0(tdl,a,rNat))
    =E= ( 1 + keepCor("int",rNat)/100);

```

For GDP:

```
e_gdpKeep(rb(rNat)) ..
  (1 + keepCor("gdp",rNat)/100)
  =E= sum(td,
    vdim_(td,rNat) $ vdim0(td,rNat)
  + viim_(td,rNat) $ viim0(td,rNat)
  + vdpm_(td,rNat) $ vdpm0(td,rNat)
  + vdgm_(td,rNat) $ vdgm0(td,rNat)
  + vigm_(td,rNat) $ vigm0(td,rNat)
  + vipm_(td,rNat) $ vipm0(td,rNat))/gdp(rNat);
```

And for factor income plus B.O.P.:

```
e_factinc(rb(rNat)) ..
*
  (1 + keepCor("factyBop",rNat)/100)
  =E= [sum( (f,a) $ vfm0(f,a,rNat) , vfm_(f,a,rNat))
    - sum( (td,rNat1) $ vxmd0(td,rNat1,rNat1) , vxmd(td,rNat,rNat1)*(1-rtxs(td,rNat,rNat1)))
    + sum( (td,rNat1) $ vxmd0(td,rNat1,rNat1) , viws(td,rNat1,rNat) * vxmd(td,rNat1,rNat)/vxmd0(td,rNat1,rNat))
    - sum(td, vst(td,rNat))]
  / (facty(rNat) + bop0(rNat) - sum(td,vst0(td,rNat)));
```

And for factor income, for each factor and in total:

```
*
* --- try to maintain benchmark factor income plus BOP
*
e_vfmKeep(f,rb(rNat)) $ sum( (a) , vfm0(f,a,rNat)*(1+rtf(f,a,rNat))) ..
*
  (1 + sum(sameas(keepItem,f) , keepCor(keepItem,rNat)/100))
  =E= sum( (a) $ vfm0(f,a,rNat) , vfm_(f,a,rNat) *(1+rtf(f,a,rNat)) )/sum( (a) , vfm0(f,a,rNat)*(1+rtf(f,a,rNat)));
e_vfmTotKeep(rb(rNat)) ..
*
  (1 + keepCor("vfm",rNat)/100)
  =E= sum( (f,a) $ vfm0(f,a,rNat) , vfm_(f,a,rNat) *(1+rtf(f,a,rNat)) )/sum( (f,a) , vfm0(f,a,rNat)*(1+rtf(f,a,rNat)));
```

Similar statements define correction factors for government demand, investments, private consumption. These elements can be switched on the interface.

## Overall solution logic

The program stepwise increases the threshold applied,

```
curAbsTol      = absTol * tolMultiplier(itr+1);
curRelTol(td,rNat) = relTolSpec(td,rNat) * TolMultiplier(itr+1);
curRelTol(td,rNat) = round(curRelTol(td,rNat),8);
```

At the beginning of each iteration, it re-defines the aggregate values of imports, transport demand, exports, domestic sales and aggregate production, e.g.

```
*
* --- Aggregate value of imports (from bi-lateral exports, plus transport marings, plus import tariffs))
vim(td,rNat) = sum(rNat1,(vxmd(td,rNat1,rNat)*(1-rtxs(td,rNat1,rNat))+sum(tdl,vtwr(tdl,td,rNat1,rNat)))*(1+rtms(td,rNat1,rNat)));
```

Next, it deletes positions which are below the thresholds, for instance:

```
--- delete import and domestic gov demand for products
if below tolerance times total gov demand
vigm(td,rNat) $ ((vigm(td,rNat) < curRelTol(td,rNat)/100 * tvgm(rNat) * tvgmi(rNat)/tvgm(rNat)) $ vigm(td,rNat)) = 0;
vdgm(td,rNat) $ ((vdgm(td,rNat) < curRelTol(td,rNat)/100 * tvgm(rNat) * (1-tvgmi(rNat)/tvgm(rNat))) $ vdgm(td,rNat)) = 0;
```

After dropping these items, total import and domestic demand are re-aggregated:

```
--- aggregate total import and domestic demand per product
```

```
vim(td,rNat) = sum(td1,vifm(td,td1,rNat)) + vipm(td,rNat) + vigm(td,rNat) + viim(td,rNat);
vdm(td,rNat) = sum(td1,vdfm(td,td1,rNat)) + vdpm(td,rNat) + vdgm(td,rNat) + vdim(td,rNat);
```

Exports of a product by a region are dropped if (1) the product's exports fall below the threshold of the exporters' total exports, and (2) its import value fall below the threshold of the total import of the importer of that product:

```
if ( relTol > 1.E-10,
    dropexports(td,rNat) $ (vxm(td,rNat) < curRelTol(td,rNat)/100*sum(td1,vxm(td1,rNat)) and
        (sum(rNat1 $
            ( (vxmd(td,rNat,rNat1)*(1-rtxs(td,rNat,rNat1))+sum(td1,vtwr(td1,td,rNat,rNat1))*(1+rtms(td,rNat,rNat1)) /
              vim(td,rNat1)) < curRelTol(td,rNat1)/100)
            $ vim(td,rNat1),1) eq sum(rNat1 $ vim(td,rNat1),1) and vxm(td,rNat)) = YES;
);
```

Related bi-lateral entries are removed:

```
--- delete bilateral entries if exports are dropped
```

```
vxmd(td,rNat,rNat1) $ dropexports(td,rNat) = 0;
vtwr(td1,td,rNat,rNat1) $ dropexports(td,rNat) = 0;
rtxs(td,rNat,rNat1) $ dropexports(td,rNat) = 0;
rtms(td,rNat,rNat1) $ dropexports(td,rNat) = 0;
```

Imports are dropped if there is no import demand left:

```
--- drop imports if no import demand
```

```
dropimports(td,rNat) $ (vigm(td,rNat)+vipm(td,rNat)+viim(td,rNat)+sum(td1,vifm(td,td1,rNat)) eq 0) = yes;
dropimports(td,rNat) $ (not vdm_.range(td,rNat)) = no;

vim(td,rNat) $ dropimports(td,rNat) = 0;
```

Equally, production is dropped if its share – corrected for own intermediate consumption such as seed use – is below the share on total production value:

```
dropprod(td,rNat) $ ( (vom(td,rNat)-vdfm(td,td,rNat)<curRelTol(td,rNat)/100*sum(td1,vom(td1,rNat)-vdfm(td1,td1,rNat)))
    $ vom(td,rNat)) = yes;
```

Equally, production is dropped if no domestic and export demand is left:

```
*
* --- drop production if no domestic and export demand
*
dropprod(a,rNat) $ ( sum(make(a,td), (vxm(td,rNat)+vdgm(td,rNat)+vdim(td,rNat)+vdpm(td,rNat)
    +sum(aCur $ (not sameas(aCur,a)),vdfm(td,aCur,rNat)))) eq 0) = yes;
```

Some of these statements are repeated in a loop to ensure that all removals are consistent, not shown here.

Afterwards, bi-lateral flows are subject to a drop if there value on total imports is below the threshold:

```
droptrade(td,rNat,rNat1) $ vxmd(td,rNat,rNat1)
    = yes $ (vxmd(td,rNat,rNat1)<min(curRelTol(td,rNat1),curRelTol(td,rNat))/100 * min( vxm(td,rNat), vim(td,rNat1)));
```

In order to make sure that the bi-lateral trade balances are approximately maintained, the remaining trade flows are scaled:

```

--- rescale vxmd to match approx. old exports of exporter and imports of importer

bilatTradeScale(rNat,rNat1) = ( [sum(tdl,vxm0(tdl,rNat))/sum(tdl,vxm(tdl,rNat))] $ sum(tdl,vxm(tdl,rNat))
+ [sum(tdl,vim0(tdl,rNat1))/sum(tdl,vim(tdl,rNat1))] $ sum(tdl,vim(tdl,rNat1)) )/2;

vxmd(td,rNat,rNat1) $ vxmd(td,rNat,rNat1) = vxmd(td,rNat,rNat1) * bilatTradeScale(rNat,rNat1);
vtwr(tdl,td,rNat,rNat1) $ vtwr(tdl,td,rNat,rNat1) = vtwr(tdl,td,rNat,rNat1) * bilatTradeScale(rNat,rNat1);

```

After these corrections, a global scaler is applied to ensure that global trade in a product does not change too much:

```

--- rescale global trade (up to 20%)

vxt(td) = sum( rNat,rNat1, vxmd(td,rNat,rNat1));
vxt(td) $ vxt(td) = min(1.2,vxt0(td)/vxt(td));
vxmd(td,rNat,rNat1) $ vxmd(td,rNat,rNat1) = vxmd(td,rNat,rNat1) * vxt(td);

--- apply same scaler to trade margins

vtwr(tdl,td,rNat,rNat1) $ vtwr(tdl,td,rNat,rNat1) = vtwr(tdl,td,rNat,rNat1) * vxt(td);

```

Similarly, not shown here, also import and export tax rates are scaled to maintain total tariff income and export revenues for each country, not shown here. The resulting entries are used as starting points for the next iteration:

```

vdm_.1(td,rNat) = vdm(td,rNat);
vfm_.1(fe,td,rNat) = vfm(fe,td,rNat);
vifm_.1(td,tdl,rNat) = vifm(td,tdl,rNat);
vdfm_.1(td,tdl,rNat) = vdfm(td,tdl,rNat);
vdgm_.1(td,rNat) = vdgm(td,rNat);
vdpm_.1(td,rNat) = vdpm(td,rNat);
vigmm_.1(td,rNat) = vigmm(td,rNat);
vipm_.1(td,rNat) = vipm(td,rNat);
vdim_.1(td,rNat) = vdim(td,rNat);
viim_.1(td,rNat) = viim(td,rNat);

```

Afterwards, each country is solved individually to rebalance the SAM. That is possible as bilateral transactions are fixed in the equation system. The solution process consists of two steps: first a linear approximation is used to provide a good start point, before a Highest Posterior Density estimator minimizes relative squared differences.

The linear estimator pulls tiny entries to zero, i.e. such entries where the a-priori values is zero. Take the following entry as an example which relates to domestic private consumption:

```
vdpm_(td,rNat) $ ((vdpm(td,rNat) eq 0) $ vdpm0(td,rNat))
```

It entered the penalty to minimize if (1) it desired level is zero – first term, and (2) there was an entry in the data base from GTAPAgg – second term.



```

vzdef..      vz =e= sum((rNat,td)$rb(rNat),
                        1/gdp(rNat) * [
                            vdpm_(td,rNat) $ ((vdpm(td,rNat) eq 0) $ vdpm0(td,rNat))
                        + vipm_(td,rNat) $ ((vipm(td,rNat) eq 0) $ vipm0(td,rNat))
                        + vdgm_(td,rNat) $ ((vdgm(td,rNat) eq 0) $ vdgm0(td,rNat))
                        + vigm_(td,rNat) $ ((vigm(td,rNat) eq 0) $ vigm0(td,rNat))
                        + vdim_(td,rNat) $ ((vdim(td,rNat) eq 0) $ vdim0(td,rNat))
                        + viim_(td,rNat) $ ((viim(td,rNat) eq 0) $ viim0(td,rNat))
                        + sum(fe $ vfm0(fe,td,rNat), vfm_(fe,td,rNat) $ (vfm(fe,td,rNat) eq 0))
                        + sum(tdl, vifm_(td,tdl,rNat) $ ((vifm(td,tdl,rNat) eq 0) $ vifm0(td,tdl,rNat))
                          + vdfm_(td,tdl,rNat) $ ((vdfm(td,tdl,rNat) eq 0) $ vdfm0(td,tdl,rNat)) )
                        ]);

```

That linear sparsity measure is added to the relative deviations, e.g.:

```

(sqr(vdpm_(td,rNat)-vdpm0(td,rNat))/(vdpm0(td,rNat)+1.E-8))$(vdpm(td,rNat) ne 0)

```

In order to speed up processing, the non-linear problem can be solved on a grid. There are more technical details not discussed in here. The rebalancing step produces output on screen which might be interesting to check. The following screen-shots show results from filtering out from a 57x20 data base with a maximal threshold of 0.001%:

The first block reports the global totals, these deviations should be small, in here, the sum, maximal around 974 out of 35035323 are lost:

```

-- loadGTAPAgg.gms (3289) 84 Mb
  item          start      end      abs diff  rel diff
-- loadGTAPAgg.gms (3290) 84 Mb
  vfm           58750721   58749829      -892      0.00 %
  GDP           68168524   68168548        24      0.00 %
  vom           141872873  141872304     -569      0.00 %
  vxmd          19129902   19129902         0      0.00 %
  vtwr           776031    776031         0      0.00 %
  vifm          13747932   13747675     -257      0.00 %
  vdfm          60683808   60683471     -337      0.00 %
  vipm          4313228    4313725       497      0.01 %
  vdpm          35035323   35034349     -974      0.00 %
  vigm          222442     222426       -16     -0.01 %
  vdgm          12306107   12306312       205      0.00 %
  viim          2349721    2349497     -224     -0.01 %
  vdim          13941703   13942239       536      0.00 %
-- loadGTAPAgg.gms (3298) 84 Mb

```

The second block shows the reduction in the non-zero SAM entries. For the given example, despite the rather small threshold, the non-zero SAM entries as defined in the program could be reduced by almost a quarter (-23.27% in total), clearly with some differences across countries:



loadGTAPAgg.gms (3302) 84 Mb				
item	start	end	abs diff	rel diff
loadGTAPAgg.gms (3303) 84 Mb				
total	216819	166359	-50460	-23.27 %
Oceania	11915	9940	-1975	-16.58 %
Chn	11540	9398	-2142	-18.56 %
Jpn	11528	8172	-3356	-29.11 %
Kor	11429	7859	-3570	-31.24 %
Tw	11438	6877	-4561	-39.88 %
HKG	11453	7363	-4090	-35.71 %
EastAsia	11120	8754	-2366	-21.28 %
SEAsia	11954	9936	-2018	-16.88 %
SouthAsia	11600	9349	-2251	-19.41 %
Usa	11612	8732	-2880	-24.80 %
Can	11528	9682	-1846	-16.01 %
Mexx	11345	8834	-2511	-22.13 %
Arg	11171	7241	-3930	-35.18 %
Bra	11312	8417	-2895	-25.59 %
LatinAmer	11729	10638	-1091	-9.30 %
Deu	11513	9044	-2469	-21.45 %
EU_28	11786	10482	-1304	-11.06 %
MENA	11951	10839	-1112	-9.30 %
Ken	10958	7389	-3569	-32.57 %
SSA	12002	10598	-1404	-11.70 %
RestofWorld	11849	10876	-973	-8.21 %

A second list reports the changes for each product:

pdr	3735	1855	-1880	-50.33 %
wht	3816	2227	-1589	-41.64 %
gro	4011	2722	-1289	-32.14 %
v_f	4260	3208	-1052	-24.69 %
osd	4134	2626	-1508	-36.48 %
c_b	3168	1074	-2094	-66.10 %
pfb	3921	2026	-1895	-48.33 %
ocr	4311	3366	-945	-21.92 %
ctl	3897	2476	-1421	-36.46 %
oap	4236	3109	-1127	-26.61 %
rmk	3087	1430	-1657	-53.68 %

(...) and a last one for the different items, see below. Interestingly, perhaps different from expected, bi-lateral trade entries vtwr are hardly thinned out, only about 1.6% of the entries are removed. That reflects the rather conservative way how the threshold is applied for trade flows as discussed above.

vfm	3969	3844	-125	-3.15 %
vom	1197	1165	-32	-2.67 %
vxmd	24717	22896	-1821	-7.37 %
vtwr	43296	42606	-690	-1.59 %
vifm	68229	40849	-27380	-40.13 %
vd fm	68229	50520	-17709	-25.96 %
vipm	1197	1033	-164	-13.70 %
vdpm	1197	1076	-121	-10.11 %
vigm	1197	658	-539	-45.03 %
vdgm	1197	452	-745	-62.24 %
viim	1197	559	-638	-53.30 %
vd im	1197	701	-496	-41.44 %

Overall, the balancing has proven as quite useful when solving models with both high regional and sector detail.

### Special treatment for specific regions and sectors

When building a data base for a project, it might be desirable to apply less aggressive filtering thresholds for specific regions and / or sectors in the focus of the application. The algorithm therefore allows defining lists of regions / sectors with accompanying specific thresholds. The codes for regions / sectors needs to be inputted in the two text fields. “Reduced thresholds only in combination” will apply the different threshold only to the intersection of the inputted regions and sectors, otherwise, all regions and sectors inputted will be receive different thresholds. Take an example where you enter for regions “xoc” and for sectors “pd”. If “Reduced thresholds only in combination” is NOT switched on, all transactions of the region “xoc” and all transactions for the sector “pd” will be treated differently. If the “Reduced thresholds only in combination” is active, only the transaction relating both to “pd” and the region “xoc” are exemptions.

However, filtering for the remaining sectors / regions has still an impact on these exemptions. For example, if production of a sector in a region is dropped, the related export flows need to be dropped as well, affecting potentially transactions in regions and sectors where tighter thresholds are used. Tests have however indicated that very few transactions are lost in regions/sector where stringent thresholds are applied as long as the overall filtering thresholds are not too aggressive.

### Technical aspects

The global GTAP SAM is clearly not well scaled from a data balancing problem as very small and very large transaction must be hosted in the same matrix. That poses a problem for the solvers. The code uses a number of tactics to respond to that challenge. CPLEX as a specialized LP / QCP solver is extremely fast with solving large scale balancing problems with linear constraints. It is therefore the preferred choice to solve balancing problems. The quality of the QCP solution with regard to feasibility accuracy is however lower compared to a LP solution and generally also lower compared to

CONOPT4. Therefore, both for the filter and the split utility, after a solution to the balancing using squared relative differences is found, a LP is solved. Its objective function minimizes the difference between last solution and a solution fitting more accurate in the constraints. That will systematically driver the estimates up, therefore, tight bounds are introduced around the given estimates. CONOPT4 is usually able to improve the feasibility slightly over CPLEXD, such that first CPLEXD is used in that step – it is faster – and a second go with CONOPT4 starting with the solution of CPLEXD is appended. These two additional steps after the true balancing problems clearly require some extra time but prevent tiny imbalances in the SAM in later model runs. To improve the accuracy further, a generalized RAS is run over maximal 20 rounds if remaning differences in column and rows sum exceed 1.E-8. Generally, the imbalances in the resulting SAMs both in absolute and relative terms are quite low.

#### 4.5.7 Gender Differentiated labor data

Use WorldBank GDLG gender labour data



The World Bank provides labor data matching the region and sector definition of GTAP V10 (141 regions, 65 sector, base year 2014) which are available as GDX containe with CGEBox (data/gdld\_2014\_v10.gdx). The data are merge into the data set when the check-box above is operated. It can be used with Power Data base in which case proportionality assumptions are used to map the shares from the single electricity sector in the standard data set to the additional sector found in the GTAP Power Data Base.

Using the GDLG (Gender Differentiated Labour Data) will overwrite the aggregation rules from the 5 labour categories in the GTAP Data base in your aggregation file. Instead, four labour categories will be derive: female skilled, female unskilled, male skilled and male unskilled.

#### 4.5.8 Split utility

Apply SAM split



Split information file

water\_fab\_a

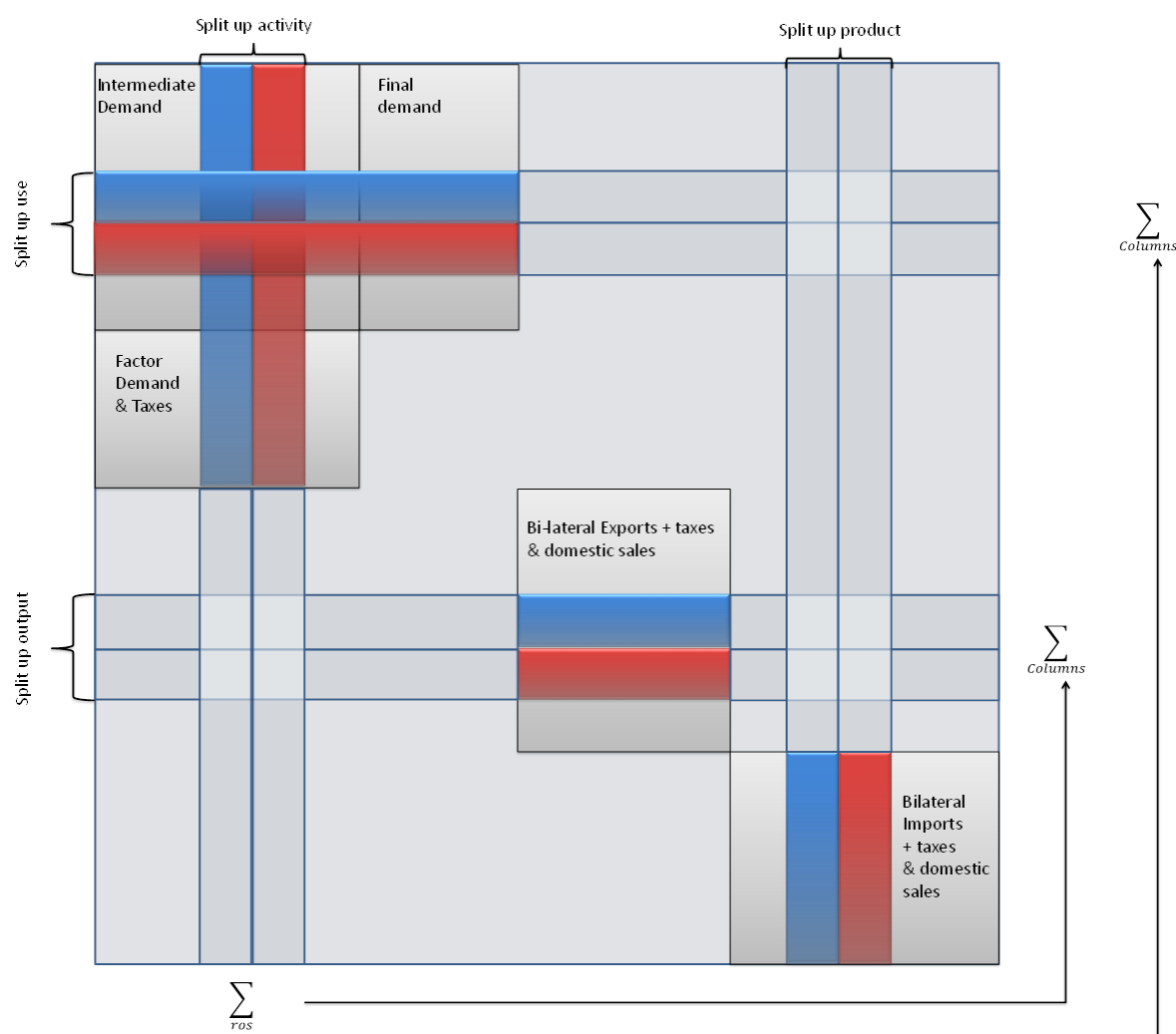


The split utility (split\split.gms), switch on with the check box shown above, allows to dis-aggregate the global SAM generated by the GTAPAgg data driver of CGEBox to further detail. It splits commodities and activities and rebalances the database using a mix of absolute and absolute normalized differences. It is discussed in Britz, W. (2021): Comparing Penalty Functions in Balancing and Dis-aggregating Social Accounting Matrices, Journal of Global Economic Analysis 6(1): 34-81.

It requires a file in a specific format which needs to be selected on the interface. These files are stored under “gams/build/split” where examples can be found.

To understand the working of the split utility, consider the case where we take one activity in each region and its related output products and split it into two activities each producing one new related output. Starting from the 57 or 65b GTAP sectors, we would add two new activities and remove one which is dis-aggregated in the new data base. The same holds for the products. SAM and other data base entries not referring either to the activity to split up or its output, i.e. product, remain untouched. For any SAM cell referring to the activity to split or its output, the new, more detailed SAM needs to provide a consistent dis-aggregation. At the same time, for the newly introduced columns and rows in the SAM, column and row sums must be identical. Additionally, auxiliary data not directly covered by the SAM such as direct and indirect tax rates differentiated by agent and by the imported and exported origin, bi-lateral import and export taxes etc. must be defined for the split-up activities and products

such as to exactly exhaust the original, more aggregated data. As detailed below, that is achieved by defining these conditions as constraints of a programming problem.



Graphic: The split problem

The user of the split utility does not need to understand in detail the set-up and solution of the programming problem as discussed below. He feeds a-priori information on split shares (or absolute values) and tax rates into the problem, minimally, it is sufficient to provide shares on output. This information is provided via GAMS code written by the user.

Examples for using the split utility can be found in the chapter relating to the FABIO MRIO, see "Linking the physical MRIO for agricultural and food products FABIO to CGEBox", pages 375ff. As detailed in that section, there is a ready-to-use link to that agri-food data base which only requires defining the mapping between the newly introduced activities and products to those found in FABIO. Another example of applying the split utility is found in Escobar, N., Haddad, S., Börner, J., Britz, W. (2018). "Land use mediated GHG emissions and spillovers from increased consumption of bioplastic", Environmental Research Letters, who split up the chemical sector of the GTAP data base into conventional plastics, bio-based ones and the rest of the chemical sector.

The balancing problem is set up as a Linear Programming (LP) problem which renders it possible to use performant LP solvers achieving a high accuracy. A former solution based on quadratic

programming proved numerically less stable. For global SAMs with many regions while introducing at the same time many splits, the resulting data balancing problems can comprise several hundredth thousands or even millions of variables and equations. It is recommended to use CPLEX as a specialized LP solver for problems of that size even if CONOPT4 might also do the job. As the data distributed by the GTAP Center have been balanced (or at least stored) with simple precision arithmetic, the split utility will typically fail if the input data are not rebalanced based on the filter routine before as detailed above.

#### 4.5.9 A complex example for a split

In the following, we discuss the input for a complex split example to help users developing their own split input. It introduces a difference between subsistence and commercial products in the SAM where only the commercial variant is internationally traded:

```
set s_spliti / wht_sub-c "Wheat subsistence"
              wht_com-c "Wheat commercial"
```

Along with related production activities:

```
set s_splita / wht_sub-a "Wheat subsistence"
              wht_com-a "Wheat commercial"
```

These set definitions must be in a first block:

```
$iftheni.mode %1==decl
↓
```

It is assumed that the subsistence activity variant uses little intermediate inputs. We first introduce assumptions on the output shares for the commercial variant and define the set of commercial activities as those where output shares are given:

```
table p_outPutShare(splita,rNat)
      set.rNat
wht_com-a 0.3
pdr_com-a 0.3
gro_com-a 0.3
v_f_com-a 0.4
osd_com-a 0.3
ocr_com-a 0.5
;

set comA(splita) "Commercial variants";
comA(splita) $ sum(rNat, p_outputShare(splita,rNat)) = YES;
```

The distinction is deemed to be only relevant for low-income countries. We therefore split the regions into two: those with a per-capita factor income above 2.500 \$ and below and set the output share for the high income countries to unity:

```
: *
: * --- Rest of the world and similar (check by counting the mapped original countries)
: * is not split up
: *
: set restOfWorld(rNat);
: restOfWorld(rNat) $ ( sum( (f,a), sam0(rNat,f,a)) / pop0(rNat) > 2500 ) = YES;
:
: set DC(rNat);
: DC(rNat) $ (not restOfWorld(rNat)) = yes;
:
: p_outPutShare(coma,restOfWorld) = 1;
↓
```

In order to exclude international trade of the subsistence product, we introduce “eps” for a very small number as a split factor, and set the factor for the commercial variant to unity:

```
*
* --- no imports and exports of subsistence product
*
p_splitFactor(rNat,splitI,rNat1) $ sum(i_a(splitI,subsA),1) = eps;
p_splitFactor(rNat,splitI,rNat1) $ sum(i_a(splitI,comA),1) = 1;
```

Based on the output share, we introduce assumptions on intermediate input use:

```
*
* --- almost non intermediate use for subsistence
*
p_splitFactor(DC,subsA,i) = p_splitFactor(DC,subsA,"prod") * 0.25 * sum(as_a(subsA,a),sam0(DC,i,a));
p_splitFactor(DC,comA,i) = max(sum(as_a(comA,a),sam0(DC,i,a))*1.E-4,
sum(as_a(comA,a),sam0(DC,i,a)) - sum(subsA_coma(subsA,comA),p_splitFactor(DC,subsA,i)));
```

Similar statements introduce assumption on factor cost shares. The definition of the split factors must be introduced in a second block:

```
$elseifi.mode %1==run
```

The last block allows introducing bounds and tax rates into the balancing problems after the split-factors had been processed and a-priori values for the new SAM entries derived. In our example, we remove the fix bounds on tax rates and replace them by tiny ones, and next remove tax rates for home production (= subsistence products):

```
*
* --- no taxes on consumption on home production
*
p_mintx(DC,splitI,aa) $ sum(i_a(splitI,subsA),1) = 0;
p_dintx(DC,splitI,aa) $ sum(i_a(splitI,subsA),1) = 0;

v_mintx.fx(DC,splitI,aa) $ sum(i_a(splitI,subsA),1) = 0;
v_dintx.fx(DC,splitI,aa) $ sum(i_a(splitI,subsA),1) = 0;
```

As the average tax for the subsistence and commercial has to exhaust the one found in the GTAP data base, the tax rates for the commercial one becomes an endogenous variables:

```
*
* --- consumption taxes on commercial production are endogenous
*
v_mintx.lo(DC,splitI,aa) $ (sum(i_a(splitI,comA),1) $ p_mintx(DC,splitI,aa)) = -1E+10;
v_mintx.up(DC,splitI,aa) $ (sum(i_a(splitI,comA),1) $ p_mintx(DC,splitI,aa)) = 1E+10;

v_dintx.lo(DC,splitI,aa) $ (sum(i_a(splitI,comA),1) $ p_dintx(DC,splitI,aa)) = -1E+10;
v_dintx.up(DC,splitI,aa) $ (sum(i_a(splitI,comA),1) $ p_dintx(DC,splitI,aa)) = 1E+10;
```

To avoid large deviations from the desired output shares, we introduce bounds on the sum of the row entries of the subsistence variant of each split-up activity

```
*
* --- fix desired share of subsistence production
*
v_rowSum.lo(DC,SubsA) $ (p_outputShare(SubsA,DC) gt eps)
= sum((as_a(SubsA,a),i), sam0(DC,a,i) * p_outputShare(SubsA,DC)) * 0.9;
*
v_rowSum.up(DC,SubsA) $ (p_outputShare(SubsA,DC) gt eps)
= sum((as_a(SubsA,a),i), sam0(DC,a,i) * p_outputShare(SubsA,DC)) * 1.1;
```

Deviations from these bounds receive a high penalty during balancing. The program from which the example is taken can be found under “gams\build\split\split\_subs\_fab.gms”. In that directory, more examples can be found which can serve as the basis to develop own splits.

#### 4.5.10 Balancing equations and objective function in the split problem

The four core equations of the split program shown below ensure that the given entries in the SAM for a product or activity to split are exhausted by the newly introduced more dis-aggregated commodities or products. Here, *spliti* are the newly introduced commodities and *splita* the new activities. The cross set *is\_i* shows the link between the new commodities and the commodity to split while *as\_a* catches the relations between the activities. *SAM0* is the global SAM before splitting.

The exhaustion condition of SAM cells referring to products to split found in the rows of the SAM, such as exports (excluding input demand of split-up activities, captured in the equation above), is defined as:

$$\overline{\text{sam0}}_{r, is} = \sum_{\text{spliti} \in i} v_{\text{sam}, r, \text{spliti}, is} \quad (75)$$

The exhaustion of SAM cells referring to products to split in columns, such as imports (excluding the ones referring to split-up activities), is given by the following constraint:

$$\overline{\text{sam0}}_{r, is, i} = \sum_{\text{spliti} \in i} v_{\text{sam}, r, is, \text{spliti}} \quad (76)$$

The related equations in GAMS are shown below. The first dollar conditions in each make sure that only non-zero entries in the global data base are subject to a split. The parameter *p\_sam0* comprise starting values for the new SAM entries, whereas *v\_sam* are the endogenous new SAM entries subject to balancing.

```
e_samIJ(rRun(rNat),i,js) $ (sum(is_i(spliti,i),sam0(rNat,i,js)) $ (not sum(as_a(splita,a) $ sameas(a,js),1))) ..
sam0(rNat,i,js)
=E= sum(is_i(spliti,i) $ p_sam0(rNat,spliti,js), v_sam(rNat,spliti,js));
e_samJI(rRun(rNat),js,i) $ (sum(is_i(spliti,i),sam0(rNat,js,i)) $ (not sum(as_a(splita,a) $ sameas(a,js),1))) ..
sam0(rNat,js,i)
=E= sum(is_i(spliti,i) $ p_sam0(rNat,js,spliti), v_sam(rNat,js,spliti));
```

Balancing for input and factor demands as well as taxes related to activities to split *a* is ensured by the following constraint:

$$\overline{\text{sam0}}_{r, is, a} = \sum_{\text{splita} \in a} v_{\text{sam}, r, is, \text{splita}} + \sum_{\text{splita} \in a, \text{spliti} \in is} v_{\text{sam}, r, \text{spliti}, \text{splita}} \quad (77)$$

The first term on the RHS depicts SAM rows which do not refer to products subject to splitting; these are added with the second expression. The set of new and original SAM rows is indexed with *is*; *splita* and *spliti* are the newly introduced activities and products, respectively. Variables which are zero by definition (such as demand for products to disaggregate in the matrix *v\_sam*) are fixed exogenously to zero.

The exhaustion condition for the SAM columns relating to activities in the rows (outputs, production taxes) is defined as follows:



$$\overline{\text{sam0}}_{r,a,is} = \sum_{\text{split} \in a} v_{\text{sam},r,\text{split},is} + \sum_{\text{split} \in a, \text{split} \in is} v_{\text{sam},r,\text{split},\text{split}} \quad (78)$$

In the implementation in GAMS, the first expression after the equal sign is the simpler case where the row is not itself subject to a split. The second line refers to products which are themselves split up.

```
e_samIA(rRun(rNat),is,a) $ sum(as_a(splita,a), sam0(rNat,is,a)) ..
sam0(rNat,is,a)
=E= [sum(as_a(splita,a) $ (p_sam0(rNat,is,splita) $ (not sum(is_i(splitI,i) $ sameas(is,i),1))), v_sam(rNat,is,splita))
+sum((as_a(splita,a),is_i(splitI,i)) $ (p_sam0(rNat,spliti,splita) $ sameas(is,i)), v_sam(rNat,spliti,splita))];
e_samAI(rRun(rNat),a,is) $ sum(as_a(splita,a), sam0(rNat,a,is)) ..
sam0(rNat,a,is)
=E= [sum(as_a(splita,a) $ p_sam0(rNat,splita,is), v_sam(rNat,splita,is))
+sum((as_a(splita,a),is_i(splitI,i)) $ (p_sam0(rNat,splita,spliti) $ sameas(is,i)), v_sam(rNat,splita,spliti))];
```

The split utility allows introducing a priori-information on bi-lateral import and export taxes, taxes on consumption for the split-up products as well as production taxes, factor taxes and subsidies for the split-up activities. If these are not provided, the rates for the products resp. sectors to split-up are used.

In order to balance indirect taxes by the different Armington agents, new parameters and variables are introduced: p\_dm0 and p\_mm0 are starting entries for the domestic and import demands to split, and v\_dm and v\_mm are the related variables subject to balancing. For case where neither case is relevant, the original data entries as read from GTAPAgg, i.e. xdm0 and xmm0, enter the RHS of the equation:

```
*
* --- indirect taxes from given tax rates and estimated domestic and import use
*
e_indtx(rRun(rNat),aa) $ (p_sam0(rNat,"indtx",aa) $ (not sum(as_a(splita,a) $ sameas(a,aa),1))) ..
[ v_sam(rNat,"indtx",aa) $ sum(sameas(aa,splita),1)
+ p_sam0(rNat,"indtx",aa) $ (not sum(sameas(aa,splita),1))]
*
* (that is solved perhaps somewhat curious to keep the model linear
* we multiply the endogenous quantity v_dm with a given tax rate dintx
*
* --- entries for split up products and/or split up activities
*
=E= [ sum( is_i(i,j) $ p_dm0(rNat,i,aa), p_dintx(rNat,i,aa) * v_dm(rNat,i,aa)
+ v_dintx(rNat,i,aa) * p_dm0(rNat,i,aa))
+ sum( is_i(i,j) $ p_mm0(rNat,i,aa), p_mintx(rNat,i,aa) * v_mm(rNat,i,aa)
+ v_mintx(rNat,i,aa) * p_mm0(rNat,i,aa))
*
* --- taxes linked to non-split up products for not split-up activities
*
+ sum( is_i(i,j) $ (not p_dm0(rNat,i,aa)), p_dintx(rNat,i,aa) * xdm0(rNat,i,aa))
+ sum( is_i(i,j) $ (not p_mm0(rNat,i,aa)), p_mintx(rNat,i,aa) * xmm0(rNat,i,aa));
```

That solution requires some commenting upon. In order to use CPLEXD as an efficient solver, the constraints are kept linear, i.e. we do not define a product of the split up transaction such as the demand for the domestic origin v\_dm multiplied with the related tax rate dintx. Instead, the a-priori chosen rate (either supplied by the GTAP SAM or introduced by the user) p\_dintx is multiplied with the endogenous transaction v\_dm and endogenous correction to the tax rate v\_dintx is multiplied with the a priori quantity p\_dm0. Exhaustion of the given tax income reported in the GTAP SAM is ensured by an additional equation which split up the domestic demand at agents' prices:

```

e_dintx(rRun(rNat),armSplit(i,aa)) $ xda0(rNat,i,aa) ..
    xda0(rNat,i,aa) =E=
*
*   sum( is_i(j,i),
*
*   --- these are agents not subject to a split
*
*   (1+p_dintx(rNat,j,aa)) * v_dm(rNat,j,aa) $ p_dm0(rNat,j,aa)
+   v_dintx(rNat,j,aa) * p_dm0(rNat,j,aa) $ p_dintx(rNat,j,aa)
*
*   --- add the split-up activities
*
+   sum(as_a(splita,a) $ (p_dm0(rNat,j,splita) $ sameas(a,aa)),
      (1+p_dintx(rNat,j,splita)) * v_dm(rNat,j,splita)
+   v_dintx(rNat,j,splita) * p_dm0(rNat,j,splita) $ p_dintx(rNat,j,splita)));
*

```

To avoid that tax rate become implausible high or low, two additional inequalities restrict their value relative to the tax base:

```

e_dintx0(rRun(rNat),i,aa) $ (p_dm0(rNat,i,aa) $ (v_dintx.range(rNat,i,aa) ne 0) $ p_dintx(rNat,i,aa)) ..
    v_dintx(rNat,i,aa) * p_dm0(rNat,i,aa) + v_dm(rNat,i,aa) * p_dintx(rNat,i,aa)
=L= v_dm(rNat,i,aa) * max(0.75,p_dintx(rNat,i,aa));

e_dintx1(rRun(rNat),i,aa) $ (p_dm0(rNat,i,aa) $ (v_dintx.range(rNat,i,aa) ne 0) $ p_dintx(rNat,i,aa)) ..
    v_dintx(rNat,i,aa) * p_dm0(rNat,i,aa) + v_dm(rNat,i,aa) * p_dintx(rNat,i,aa)
=G= v_dm(rNat,i,aa) * min(-0.75,max(-0.99,p_dintx(rNat,i,aa)));

```

Similar equations are present for factor taxes and subsidies, production taxes and import and export taxes.

The global SAM does not comprise information of the domestic versus imported use of the different Armington agents, see also above. Maintaining the balance here requires additional equations. The first one makes sure that the total use of product  $i$  by an Armington agent  $aa$  – new and old commodities and sectors are considered - is exhausted by domestic and imported consumption:

$$v_{sam_{r,i,aa}} = v_{dm_{r,i,aa}} + v_{mm_{r,i,aa}} \quad (79)$$

In GAMS:

```

e_arm(rRun(rNat),i,aa) $ p_sam0(rNat,i,aa) ..

[ v_sam(rNat,i,aa) $ consider(i,aa)
+ p_sam0(rNat,i,aa) $ (not consider(i,aa))]
=E= [ v_dm(rNat,i,aa) $ p_dm0(rNat,i,aa)
+ v_mm(rNat,i,aa) $ p_mm0(rNat,i,aa)
+ xdm0(rNat,i,aa) $ (not p_dm0(rNat,i,aa))
+ xmm0(rNat,i,aa) $ (not p_mm0(rNat,i,aa))];

```

The import and export tax equations are defined as follows, where the first equations defines all import taxes related to split-up products and the second exhaust for each bi-lateral flow the import taxes comprised in auxiliary matrices. Note that the definition during the solve of the parameter  $imptxY0$  for the products to split-up – i.e. the given values – refers to the total transaction, while for split-up products to the tax rates. As explained for the indirect taxes above the endogenous tax rate is defined implicitly from two terms which both multiply a constant with an endogenous variable:

```

*
* --- import taxes from for split-up product
*
e_imptx(rRun(rNat),spliti) $ p_sam0(rNat,"imptx",spliti) ..
    v_sam(rNat,"imptx",spliti)
    =E= sum((r0,is_i(spliti,i)) $ (p_sam0(rNat,r0,spliti) and sam0(rnat,r0,i)),
        v_sam(rNat,r0,spliti) * imptxy0(r0,rNat,spliti)
        + p_sam0(rNat,r0,spliti) * v_imptx(r0,rNat,spliti));
*
* --- absorption of bi-lateral import taxes
*
e_imptxB(rNat,rNat1,i) $ (sam0(rnat,rnat1,i) $ sum(is_i(spliti,i),1)) ..
    imptxy0(rNat1,rNat,i)
    =E= sum(is_i(spliti,i) $ p_sam0(rNat,rNat1,spliti),
        v_sam(rNat,rNat1,spliti) * imptxy0(rNat1,rNat,spliti)
        + p_sam0(rNat,rNat1,spliti) * v_imptx(rNat1,rNat,spliti));

```

Besides the exhaustion condition for the original cells, the column and row sums for the newly introduces commodities and activities must be equal by definition:

$$v\_rowSum_{r,spliti} = \sum_{is} v\_sam_{r,is,spliti} \quad (80)$$

$$v\_rowSum_{r,spliti} = \sum_{is} v\_sam_{r,spliti,is} \quad (81)$$

$$v\_rowSum_{r,splita} = \sum_{is} v\_sam_{r,is,splita} \quad (82)$$

$$v\_rowSum_{r,splita} = \sum_{is} v\_sam_{r,splita,is} \quad (83)$$

The row sum variables for split-up products are substituted out, while they are maintained for split-up activities as they are used in other equations. One could naturally delete the `v_rowSum` variable altogether which would save equations and variables in the overall split program. It turned however useful in certain cases to control the sums, e.g. if information for the production value of a split-up activities is to be closely maintained.

```

*
* --- sum of column entries for split products must be equal to related row sum
*
e_samBalI(rRun(rNat),spliti) $ sum(js,p_sam0(rNat,js,splitI)) ..
    sum(js $ p_sam0(rNat,js,splitI), v_sam(rNat,js,splitI)) =E= sum(js $ p_sam0(rNat,splitI,js), v_sam(rNat,splitI,js));
*
* --- same, for split activities
*
e_samBalA(rRun(rNat),splita) $ sum(js,p_sam0(rNat,js,splitA)) ..
    sum(js $ p_sam0(rNat,js,splitA), v_sam(rNat,js,splitA)) =E= sum(js $ p_sam0(rNat,splitA,js), v_sam(rNat,splitA,js));
*
* --- variables related to bounds on sum of SAM rows for new products / activities
*
e_rowSumA(rRun(rNat),splita(splita)) $ sum(jsj,p_sam0(rNat,jsj,splitA)) ..
    v_rowSum(rnat,splita) =E= sum(jsj $ p_sam0(rNat,jsj,splitA), v_sam(rNat,jsj,splitA));
e_rowSumI(rRun(rNat),spliti) $ (sum(jsj,p_sam0(rNat,jsj,splitI))
    $ sum(splita,p_sam0(rnat,splita,spliti))) ..
    v_rowSum(rnat,splitI) =E= sum(jsj $ p_sam0(rNat,jsj,splitI), v_sam(rNat,jsj,splitI));

```

Further equations ensure that f.o.b. price plus margins are equal to the c.i.f. price, assuming that the per unit margin  $xmarg0$  is the identical between the split up commodities. Note that the order of the regional indices is reverted: the LHS has the exporter region  $rNat$  on the first index to depict the value at f.o.b. prices, the RHS has the importer index  $rp$  first to depict the value at c.i.f.:

$$v\_sam_{r,spliti,r0} = v\_sam_{r0,r,spliti} \left( 1 + \sum_{spliti \in m} \frac{xmarg0_{m,r0,i,r}}{sam0_{r0,i,r}} \right) \quad (84)$$

In GAMS:

```
e_trade(rRun(rNat),splitI,rp) $ (p_sam0(rNat,spliti,rp) $ (card(rRun) gt 1) ) ..
v_sam(rNat,splitI,rp) * ( 1 + sum( (is_i(spliti,i),m) $ sam0(rNat,i,rp),xmarg0(m,rNat,i,rp)/sam0(rNat,i,rp))
=E= v_sam(rp,rNat,spliti);
```

Equally, the given imported  $xmm0$  and domestic  $xdm0$  use for each product at market prices in the original SAM for the Armington agents needs to be exhausted:

$$x\_mm0_{r,i,aa} = \sum_{spliti \in i} v\_mm_{r,spliti,aa} + \sum_{splita \in aa} v\_mm_{r,i,splita} \quad (85)$$

$$x\_dm0_{r,i,aa} = \sum_{spliti \in i} v\_dm_{r,spliti,aa} + \sum_{splita \in aa} v\_dm_{r,i,splita} \quad (86)$$

In GAMS:

```
e_xmmIAA(rRun(rNat),i,aa) $ (xmm0(rNat,i,aa) $ (not sum(sameas(splita,aa),1)) $ (card(rRun) gt 1)) ..
xmm0(rNat,i,aa)
=E= [ sum(is_i(j,i), v_mm(rNat,j,aa) $ p_mm0(rNat,j,aa) + xmm0(rNat,j,aa) $ ((not consider(j,aa)) and (not exclude(j,aa))
+ sum(as_a(splita,a) $ (p_mm0(rNat,j,splita) $ sameas(a,aa)), v_mm(rNat,j,splita)))]];

e_xdmIAA(rRun(rNat),i,aa) $ (xdm0(rNat,i,aa) $ (not sum(sameas(splita,aa),1)) ) ..
xdm0(rNat,i,aa)
=E= [ sum(is_i(j,i), v_dm(rNat,j,aa) $ p_dm0(rNat,j,aa) + xdm0(rNat,j,aa) $ ((not consider(j,aa)) and (not exclude(j,aa))
+ sum(as_a(splita,a) $ (p_dm0(rNat,j,splita) $ sameas(a,aa)), v_dm(rNat,j,splita)))]];
```

Similar conditions hold for the demand of the domestic and imported origina at market prices which take indirect taxes into account:

$$xma0_{r,i,aa} = \sum_{j \in i} v_{mm,r,j,aa} (1 + p_{mintx_{r,j,aa}}) + v_{mintx_{r,j,aa}} + \sum_{splita \in aa} v_{mm,r,j,splita} (1 + p_{mintx_{r,j,splita}}) + v_{mintx_{r,j,splita}} \quad (87)$$

$$xda0_{r,i,aa} = \sum_{j \in i} v_{dm,r,j,aa} (1 + p_{dintx_{r,j,aa}}) + v_{dintx_{r,j,aa}} + \sum_{splita \in aa} v_{dm,r,j,splita} (1 + p_{dintx_{r,j,splita}}) + v_{dintx_{r,j,splita}} \quad (88)$$

Based on this, indirect taxes revenues for each agent are defined:

$$sam0_{r,"indtx",aa} = \sum_{j \in i} v_{mm,r,j,aa} (1 + p_{mintx_{r,j,aa}}) + v_{mintx_{r,j,aa}} + \sum_{j \in i} v_{dm,r,j,aa} (1 + p_{dintx_{r,j,aa}}) + v_{dintx_{r,j,aa}} \quad (89)$$

The imported use for the newly introduced commodities, added up over the Armington agents, must be equal to the bi-lateral imports at c.i.f. prices plus import taxes:

$$\sum_{aa} v_{mm,r,spliti,aa} = \sum_{r0} v_{sam,r,r0,spliti} (1 + imptxyY0_{r0,r,spliti} + v_{imptx}_{r0,r,spliti}) \quad (90)$$

Note there that  $v_{imptx}$  is an endogenous correction of the tariff revenues. In GAMS:

```
e_imp(rRun(rNat),spliti) $ (card(rRun) gt 1) ..
  (RHS fixed in single country solve)

-- imported use

sum(aa $ p_mm0(rNat,spliti,aa), v_mm(rNat,splitI,aa))
=E=
sum(r0 $ p_sam0(rnat,r0,spliti), v_sam(rnat,r0,splitI) * (1+sum(is_i(spliti,i) $ sam0(rNat,r0,i),imptxy0(r0,rNat,i)/sam0(rNat,r0,i))));
```

Finally, the market balance for the new commodities must hold:

$$\sum_{aa} v_{dm,r,spliti,aa} + \sum_{r0} v_{sam,r,spliti,r0} - v_{sam,r,"exptx",spliti} = \sum_{splita \in aa} v_{sam,r,splita,spliti} \quad (91)$$

In GAMS:

```
e_mkt(rRun(rNat),spliti) ..
*
*   -- domestic use
*
*   [ sum(aa $ p_dm0(rNat,spliti,aa), v_dm(rNat,splitI,aa))
*
*   -- exports
*
*   + sum(r0 $ p_sam0(rNat,spliti,r0), v_sam(rNat,splitI,r0))
*
*   -- exports taxes
*
*   - v_sam(rNat,"exptx",spliti) $ p_sam0(rNat,"exptx",spliti)
* ]
*
*   -- output
*
*   =E= sum(splita $ p_sam0(rNat,splita,spliti), v_sam(rNat,splita,spliti));
```

It turned useful out to introduce bounds for factor use cost shares:

```
*
*   --- plausibility bounds on cost shares
*
e_shareLo(rRun(rnat),consider(is,splita)) $ (p_sam0(rNat,is,splita) $ (p_shareLo(rNat,is,splita) ne 0)) ..
  p_shareLo(rNat,is,splita) * v_rowSum(rnat,splitA) =L= v_sam(rNat,is,splita);

e_shareUp(rRun(rnat),consider(is,splita)) $ (p_sam0(rNat,is,splita) $ (p_shareUP(rNat,is,splita) ne 0)
  $ (p_shareUP(rNat,is,splita) ne 1)) ..
  p_shareUP(rNat,is,splita) * v_rowSum(rnat,splitA) =G= v_sam(rNat,is,splita);
```

The objective function minimizes absolute (normalized) deviations between the desired new SAM entries  $p\_sam0$  and the endogenous entries  $v\_sam$  which fit into the equation system defined above. In order to do so, we need to define separate variables for the positive and negative deviations:

```
e_samSlack(rRun(rNat),considerHPD(is,js)) $ (p_sam0(rNat,is,js) and (not sam0(rnat,is,js))) ..
  v_sam(rNat,is,js) + v_samSlackP(rNat,is,js) - v_samSlackN(rNat,is,js) =E= p_samCor0(rnat,is,js);
```

These slack variables enter the objective function:

```
+ sum( (rRun(rNat),considerHPD(is,js)) $ p_sam0(rNat,is,js),
  (v_samSlackP(rNat,is,js)+v_samSlackN(rNat,is,js)) / (abs(p_sam0(rNat,is,js))+1))
```

Along with similar entries for the tax rates. To avoid that tiny imbalances left in the global SAM to split up prevent finding a solution, tiny slacks for the production activity related SAM entries to split up are introduced.

As detailed below,  $p\_sam0$  and  $p\_samCor0$  are identical for larger starting values of the split-up SAM entries, but changed to favor sparsity for very small ones. The objective function minimizes squared relative differences, in order to prevent that deviations for quite tiny entries are too important, the normalization adds one dollar to the absolute value of the starting value  $p\_sam0$ . The second term in the LHS only scales the objective function to improve overall scaling, it counts the non-zero entries and divides the sum by factor ten.

#### 4.5.11 Proportionality assumptions for cost structures during splitting

For many dis-aggregation exercises of SAMs, a-priori information is available for selected intermediate input and factor demands, only. Split factors for the remaining cost items are then derived based on proportionality assumptions. These do not carry additional information. It would therefore make sense to set their weight in the objective function to zero. The split factors where a priori information is available should then via the balancing conditions adjust the non-controlled ones. However, this likely gives the balancing framework unwanted degrees of freedom. Many different combinations of only indirectly controlled input demands might balance the problem at unchanged penalties. Technically, the problem will have a degenerate solution.

Using small penalties for costs shares derived from proportionality assumptions might help to avoid degeneracy, but especially for small absolute estimates, might provoke numerical challenges. Additionally, penalizing deviation for costs share derived from proportionality assumptions is likely to lead to some violation of proportionality conditions. This reflects that is typically impossible to derive completely consistent a-priori information. If this is the case, the whole balancing exercise would not be necessary.

Therefore, proportionality assumptions are explicitly controlled in here by two constraints.

Let  $propAssmpt$  define a three-dimensional indicator set of region, input or factor demand and activity combinations derived from to proportionality assumptions. For these cases, the following conditions are enforced:

$$v\_sam_{r,is,splita} = v\_propAssmpt_{r,splita} \sum_{splita \in a} sam0_{r,is,a} \bigwedge propAssmpt_{rnat,is,splita} \quad (92)$$

$$v\_sam_{r,spliti,a} = v\_propAssmpt_{r,spliti} \sum_{spliti \in ai} sam0_{r,i,a} \bigwedge propAssmpt_{rnat,spliti,a} \quad (93)$$

Where  $v\_propAssmpt$  are endogenously determined distribution factors which ensure that all cost shares included in  $propAssmpt$  are receive the same share on the total. These factors are not entering the objective function. Equally, entries relating to input demands controlled by proportionality assumptions are removed from the objective.

In GAMS:

```

*
* --- Same splitfactor for pre-defined lists of input demands (there is slack for the activities with the largest
* a-priori demand)
*
e_propAssmptA(propAssmpt(rNat,isi,splita)) $ (p_sam0(rnat,isi,splita) $ (v_sam.range(rnat,isi,splita) ne 0)) ..
    v_sam(rNat,isi,splita) =E= v_propAssmpt(rnat,splita)

*
* --- Same splitfactor for pre-defined lists of split-up input demands of not-split up activities
*
e_propAssmptI(propAssmpt(rNat,spliti,a)) $ (p_sam0(rnat,spliti,a) $ (v_sam.range(rnat,spliti,a) ne 0) $ (not splita(a))) ..
    v_sam(rNat,spliti,a) =E= v_propAssmpt(rnat,spliti)
                          * sum(is_i(spliti,i),sam0(rnat,i,a));

```

The demand for the domestic and imported origin for the cases shown above are using the same distribution factors. In order to avoid linear dependencies, only the larger of the demand for the domestic or imported origin is handled by the equations. They are also active, only, if both demands are present, as otherwise, one of the two is identical to total demand:

```

*
* --- keep import / domestic shares for not-split up input
*
* (1) Only the equation for the larger of the domestic or imported origin will be active
* to avoid linear dependencies with the equation for the sum of the two
*
* (2) For the same reason, if only one of the two positions is reported, drop both equations
*
e_sameDomShare(rNat,i,splita) $ (p_sam0(rnat,i,splita) $ (p_dm0(rnat,i,splita) gt p_mm0(rnat,i,splita))
    $ p_mm0(rnat,i,splita) $ p_dm0(rnat,i,splita) $ (v_dm.range(rnat,i,splita) ne 0)
    $ (v_sam.range(rnat,i,splita) ne 0) $ (not Spliti(i))) ..
    v_dm(rNat,i,splita) =E= v_sam(rnat,i,splita) * sum(as_a(splita,a),xdm0(rnat,i,a)/sam0(rnat,i,a));

e_sameImpShare(rNat,i,splita) $ (p_sam0(rnat,i,splita) $ (p_mm0(rnat,i,splita) ge p_dm0(rnat,i,splita))
    $ p_mm0(rnat,i,splita) $ p_dm0(rnat,i,splita) $ (v_mm.range(rnat,i,splita) ne 0)
    $ (v_sam.range(rnat,i,splita) ne 0) $ (not spliti(i))) ..
    v_mm(rNat,i,splita) =E= v_sam(rnat,i,splita) * sum(as_a(splita,a),xmm0(rnat,i,a)/sam0(rnat,i,a));

```

For the demand of split up commodities by non split-up activities, the domestic demand is controlled by an equation if it is present:

```

e_propAssmptID(propAssmpt(rNat,spliti,a)) $ (p_sam0(rnat,spliti,a) $ p_dm0(rnat,spliti,a) $ p_mm0(rnat,spliti,a) $ (v_dm.range(rnat,spliti,a) ne 0)
    $ (v_sam.range(rnat,spliti,a) ne 0) $ (not splita(a))) ..
    v_dm(rNat,spliti,a) =E= v_propAssmpt(rnat,spliti)
                          * sum(is_i(spliti,i),xdm0(rnat,i,a));

```

As an example, consider the crop activities where FABIO delivers a-priori information on production, land demands and seed use. The difference for a disaggregated crop activity between its share on production and its share on land and seed use of the aggregate activity should be its share on all other costs. These other cost items shall be distributed on proportionality assumptions. The equation above will ensure that the relations between these other cost items do not change. For instance, while the relation between capital and land will change for a dis-aggregated crop – reflecting a different cost share of land based on yield information from FABIO –, the relation between capital and a labour category will be identical to the aggregate activity for an example), reflecting the constraint above. A lower yield, such as in case of rye in the example below, will imply a higher cost share of land and lower cost shares of other factors (and intermediate demand, not shown in the table).

#### 4.5.12 Preparation of the split factors

The minimum information required to perform a split are split factors for the production values. These factors do not need to add up to the unity, one could also enter totals as the GAMS code will scale the sum of the split factors for each SAM cell to unity.

If no split factors for the four following demand categories are given:



```
set dem(is) / hhsld,gov,inv,int /;
```

The program will use the related split factor for output instead:

```
*
* --- if split information of final demand category is missing, use output factor instead
*
p_splitFactor(rNat,spliti,dem) $ (not sum(is_i(spliti1,i) $ is_i(spliti,i), p_splitFactor(rNat,spliti1,dem)))
    = sum(i_a(splitI,splitA), p_splitFactor(rNat,splitA,"prod"));
*
* --- if splitfactor is still missing, assume equal shares
*
p_splitFactor(rNat,spliti,dem) $ (not sum(is_i(spliti1,i) $ is_i(spliti,i), p_splitFactor(rNat,spliti1,dem))) = 1;
```

Split factors below 1.E-5 will be deleted; however, “eps” entries which imply that the split factor should be zero in the final solution are maintained:

```
p_splitFactor(rNat,splitI,is) $ ((abs(p_splitFactor(rNat,splitI,is)) le 1.E-5)
    $ (p_splitFactor(rNat,splitI,is) ne eps)) = 0;
```

Afterwards, the non-zero split factors for the demand categories are scaled to unity:

```
p_splitFactor(rNat,splitI,is) $ ( (p_splitFactor(rNat,splitI,is) ne eps) $ p_splitFactor(rNat,splitI,is))
    = p_splitFactor(rNat,splitI,is) / sum(is_i(spliti1,i) $ is_i(spliti,i), p_splitFactor(rNat,spliti1,is));
```

These two statements are repeated to decrease the chance of very tiny split factors. That logic is also applied for split-factors on output “prod”, and for intermediate input use and output of commodities. Attention: the order of column and rows in the split factors are reverted compared to the SAM to ease their handling in calling programs, i.e. `p_splitFactor(rNat,splitA,is)` refers to the use of row is in the activity splitA.

Split factors are used as follows. Explicit split factors for a specific SAM cells (intermediate input or factor demand by an activity, output coefficients, demand positions, export flows) are given precedence. If no split-factor for an individual intermediate input demand is found, a split factor for general intermediate demand (“int”) for the newly introduced product is used. For intermediate input demands of split-up products by split-up activities, the product of the “int” and “prod” factors is used if no explicit factor is present. If split factor for intermediate input demands are missing, the split factor for production (“prod”) of the activity demanding the input is used instead. For individual outputs of a new activity, if no explicit split factor is given, the split factor for production “prod” is used as well. The same holds if factor demands of new activities are missing. During these assignments, split factors representing original zeros in the underlying database are not overwritten. As discussed below in next section, split-factors constructed for cost shares where no original a-priori information is available have no impact on the results.

After these assignments, a check tests for each SAM cell to disaggregate if related split factors are present. Missing split factors can result from removing small shares or from “true” zeros in the original data. If no split-factors are found for a SAM cell to be disaggregated, the aggregated SAM value is distributed in equal shares to the new SAM cells. Afterwards, split factors for any SAM cell to disaggregate are guaranteed to be present. A-priori estimates of new SAM cells to introduce are generated by multiplying their split factor with the related aggregate SAM cell. All new SAM cells are afterwards scaled to exhaust the related aggregate ones.

A challenge provides the decomposition of the individual demand positions to the domestic and imported origin. Using the shares for the related aggregate product for newly introduced demands can lead to inconsistent a-priori information which does not match the a-priori data on disaggregated bi-

lateral trade and domestic production. For instance, import demand of products in a region might be created where no imports are reported, or domestic demand where no output is produced. To address this, the domestic and import shares for all Armington demands for the aggregate products are calculated first. This step produces import and domestic demand shares in sum of all Armington agents  $aa$  for each national market for the aggregated products to split  $i$ :

$$\text{impShare}_{r,i} = \frac{\sum_{aa} \text{xm0}_{r,i,aa}}{\sum_{aa} \text{sam}_{r,i,aa}} \quad (94)$$

$$\text{domShare}_{r,i} = 1 - \text{impShare}_{r,i} \quad (95)$$

Symbols with a 0 at the end, such as  $\text{sam0}$ , refer to the benchmark or a priori values.  $\text{xm0}$  refers to import and  $\text{xdm0}$  to domestic demand. Endogenous variables under control of the split balancing framework start with a  $v\_$ . Accordingly,  $v\_ \text{sam}$  refers to the newly introduced, disaggregated SAM cells,.

Next, imports for disaggregated products into region  $r$  from the exporter regions  $r0$  are set in relation to their total demands to arrive at import share estimates for the disaggregated products  $\text{split}_i$  in this region:

$$\text{impShare}_{r,\text{split}_i} = \frac{\sum_{r0} v\_ \text{sam}_{r,r0,\text{split}_i} (1 + p\_ \text{imptx}_{r0,r,\text{split}_i})}{\sum_{aa} v\_ \text{sam}_{r,\text{split}_i,aa}} \quad (96)$$

where  $\text{split}_i$  are the newly, more disaggregated products and  $p\_ \text{imptx}$  the bilateral import tax rates applied on the import flow at c.i.f. value.

If domestic output is observed, the domestic share is determined as the residual, with a security threshold of 1%. This is important for cases where the a-priori estimates for imports exceed the reported demand:

$$\text{domShare}_{r,\text{split}_i} = \max(0 + 0,01 \exists \text{ output}_{r,\text{split}_i}, 1 - \text{impShare}_{r,\text{split}_i}) \quad (97)$$

Due to the maximum operator and the fact that domestic shares must become zero if no domestic output is reported, the import share has to be re-calculated afterwards:

$$\text{impShare}_{r,\text{split}_i} = 1 - \text{domShare}_{r,\text{split}_i} \quad (98)$$

Based on these shares, relations between the aggregate and disaggregated import and domestic use shares are defined for the disaggregated products:

$$\text{impRel}_{r,\text{split}_i} = \frac{\text{impShare}_{r,\text{split}_i}}{\sum_{i \in \text{split}_i} \text{impShare}_{r,i}} \quad (99)$$

$$\text{domRel}_{r,\text{split}_i} = \frac{\text{domShare}_{r,\text{split}_i}}{\sum_{i \in \text{split}_i} \text{domShare}_{r,i}} \quad (100)$$

This allows the calculation of a scaling factor  $scaleArm$  which ensures that the SAM entries to disaggregate will be exhausted when the distribution factors for the domestic and imported origin calculated above are applied:

$$scaleArm_{r,spliti,aa} = \sum_{i \in spliti} \frac{x_{dm0}(rnat, i, aa) domRel_{r,spliti} + x_{mm0}(rnat, i, aa) impRel_{r,spliti}}{sam0_{r,i,aa}} \quad (101)$$

Afterwards, the reported aggregated import and domestic demand shares for a product and an Armington agent are multiplied with the demand for the split up product, as derived from the split factors ( $v_{sam_{r,spliti,aa}}$ ), and with the relation between the disaggregated and aggregated import respectively domestic shares, as well as with the scaling factor:

$$v_{mm_{r,spliti,aa}} = v_{sam_{r,spliti,aa}} \sum_{i \in spliti} \frac{x_{mm0_{r,i,aa}}}{sam0_{r,i,aa}} impRel_{r,spliti} scaleArm_{r,spliti,aa} \quad (102)$$

$$v_{dm_{r,spliti,aa}} = v_{sam_{r,spliti,aa}} \sum_{i \in spliti} \frac{x_{dm0_{r,i,aa}}}{sam0_{r,i,aa}} domRel_{r,spliti} scaleArm_{r,spliti,aa} \quad (103)$$

$v_{mm}$  refers to the disaggregated import demands and  $v_{dm}$  to the disaggregated domestic ones, while  $x_{dm0}$  and  $x_{mm0}$  depict the Armington demands at market prices from the filtered GTAP Data Base.

Afterwards, further checks and potential changes to the a-priori data are introduced:

- If no domestic or import demand for an Armington demand to disaggregate is found, the given total is assigned and the so-constructed disaggregated values are scaled to exhaust the given total.
- In case where exports are reported, but no import demand (from aggregated import demand over all Armington agents), the export flow is deleted if there are other export flows of disaggregated products relating to the same GTAP product on the same trade link. If export flows cannot be deleted as they are the only ones reported, each Armington demander showing import demand in each import country where such missing export flows are observed gets assigned a very tiny import demand.
- In case of missing production, exports flows and domestic demands are deleted.

Afterwards, indirect, export and factor tax rates are defined from the aggregate transactions and assigned to the disaggregated ones. Bilateral import tariffs are taken from TASTE as long as the necessary scaling factor to align the detailed tariff rate to given tariff revenues is not larger than five or smaller than one fifth. Individual ad-valorem tariff rates which exceed the average rates for a product on a bilateral trade flow by a very large factor are deleted.

After such corrections, the disaggregated data are re-scaled to exhaust the given totals. The resulting data provide the a-priori information for the linear-loss estimator framework (for a technical discussion see Britz 2021 for details) discussed next. The main equations found in the framework are documented in the next section. During the solution of the split problem, solely weights for the “secure” data (TASTE, production, intermediate demand and land use from FABIO) are used, which higher weights for production, bi-lateral trade and land use. The framework also introduces penalties

for very small entries for new SAM cells to favour sparsity and comprises constraints to control for reasonable factor cost shares and tax rates.

#### 4.5.13 Split solution process

As seen from the equation structure above, the constraints are kept linear to ease solving the problem. As the objective function is linear, besides CONOPT as the default solver user in CGEBox, also highly performant LP solvers such as CPLEX can be used. That has shown to speed up dramatically splitting up SAMs with many sectors and regions where the rebalancing problem comprises several 10 thousands of variables and equations. The QP solver (CONOPT4 or CPLEX) can be chosen on the interface and will be used as the LP solver for the split. In order to speed up the solution process, the split factors are used to define starting values for the new SAM entries `v_sam`, for instance:

```
v_sam.l(rNat,splitI,a) = sum(is_i(splitI,i), sam0(rNat,i,a) * p_splitFactor(rNat,splitI,"int"));
```

The same holds for other elements such as domestic and imported use of the Armington agents, for instance:

```
v_mm.l(rNat,splitI,aa) $ sum(is_i(splitI,i), xmm0(rNat,i,aa))
= v_sam.l(rNat,splitI,aa) * sum(is_i(splitI,i), xmm0(rNat,i,aa)/sam0(rNat,i,aa));
v_dm.l(rNat,splitI,aa) $ sum(is_i(splitI,i), xdm0(rNat,i,aa))
= v_sam.l(rNat,splitI,aa) * sum(is_i(splitI,i), xdm0(rNat,i,aa)/sam0(rNat,i,aa));
```

In order to avoid that, for instance, tiny split shares in production lead to extremely small SAM entries in intermediate consumption, entries with a values < 1 USD cent are pulled to zero. That has shown to help with sparsity issues:

```
p_samCor0(rNat,is,js) = p_sam0(rNat,is,js);
*
* --- pull tiny SAM entries to zero (sparsity), units USD - all < 1 cent is pulled towards zero
*
p_samCor0(rNat,consider(is,js)) $ ( (p_sam0(rNat,is,js) gt 0) $ (p_sam0(rNat,is,js) lt 1.E-2)) = -1;
p_samCor0(rNat,consider(is,js)) $ ( (p_sam0(rNat,is,js) lt 0) $ p_sam0(rNat,is,js) $ (p_sam0(rNat,is,js) gt -1.E-2)) = +1;
*
p_samCor0(rNat,consider(is,js)) $ ( (p_sam0(rNat,is,js) gt 0) $ (p_sam0(rNat,is,js) lt 1.E-3)) = -10;
p_samCor0(rNat,consider(is,js)) $ ( (p_sam0(rNat,is,js) lt 0) $ p_sam0(rNat,is,js) $ (p_sam0(rNat,is,js) gt -1.E-3)) = +10;
*
p_samCor0(rNat,consider(is,js)) $ ( (p_sam0(rNat,is,js) gt 0) $ (p_sam0(rNat,is,js) lt 1.E-4)) = -100;
p_samCor0(rNat,consider(is,js)) $ ( (p_sam0(rNat,is,js) lt 0) $ p_sam0(rNat,is,js) $ (p_sam0(rNat,is,js) gt -1.E-4)) = +100;
```

The global SAM is solved several times, changing the option files used by the solver in case of numerical problems:

```
set tries / try1*try10/;
scalar nTries / 0/;
while ( (nTries < card(tries))
and ( ( (m_split.sumInfes > m_split.tolInfeas*m_split.numEqu/1000) and (m_split.sumInfes > 1.E-5))
or ( (m_split.modelstat > 2) and (m_split.modelstat ne 7))
) )
$batinclude 'util/title.gms' "'Search feasible solution for split, nTry '" nTries:0:0 "', sumInfes '" m_split.sumInfes
nTries = nTries + 1;
```

Several steps are taken to help with infeasibilities:

1. The option file is changed, option files with a higher number use less accurate feasibility requirement.

```

*
*   --- if the sum of infeasibilities is too high, relax accuracy of solver
*
IF ( (m_split.suminfes > m_split.tolinfeas*m_split.numEqu/1000),
    if ( nTries > 2, m_split.optFile = 6);
    if ( nTries > 4, m_split.optFile = 7);
    if ( nTries > 6, m_split.optFile = 8; m_split.tolinfeas = 1.E-7);
    if ( nTries > 7, m_split.optfile = 9; m_split.tolinfeas = 1.E-6);
    if ( nTries > 8, option LP=CONOPT4;m_split.optFile = 8);
);
*
*   --- same if the solver reports some error
*
if ( (nTries > 2) $ (m_split.modelStat > 2), m_split.optfile = 6);
if ( (nTries > 4) $ (m_split.modelStat > 2), m_split.optfile = 7);
if ( (nTries > 6) $ (m_split.modelStat > 2), m_split.optfile = 8);
if ( (nTries > 7) $ (m_split.modelStat > 2), m_split.optfile = 9);
if ( (nTries > 8) $ (m_split.modelStat > 2), m_split.optfile = 8);

```

## 2. Requirements that costs shares must fall in a certain range are step-wise relaxed:

```

--- relax lower /upper share constraints if binding
p_shareLo(rNat,is,splita) $ ( ( e_shareLo.l(rNat,is,splita) gt m_split.tolinfeas)
                             or (abs(e_shareLo.m(rNat,is,splita)) gt 1.E-12))
                             and (p_shareLo(rNat,is,splita) ne p_shareup(rNat,is,splita))
                             and (p_shareLo(rNat,is,splita) gt 0))
= p_shareLo(rNat,is,splita) * 0.75;

p_shareLo(rNat,is,splita) $ ( ( p_shareLo(rNat,is,splita) le 1.E-5) $ (p_shareLo(rNat,is,splita) gt 0) and (nTries > 2)) = 0;
p_shareLo(rNat,is,splita) $ ( ( p_shareLo(rNat,is,splita) le 1.E-4) $ (p_shareLo(rNat,is,splita) gt 0) and (nTries > 4)) = 0;
p_shareLo(rNat,is,splita) $ ( (p_shareLo(rNat,is,splita) gt 0) $ (nTries > 8)) = 0;

p_shareUp(rNat,is,splita) $ ( ( e_shareUp.l(rNat,is,splita) lt -m_split.tolinfeas)
                             or (abs(e_shareUp.m(rNat,is,splita)) gt 1.E-12))
                             and (p_shareLo(rNat,is,splita) ne p_shareup(rNat,is,splita))
                             and (p_shareUp(rNat,is,splita) gt 0))
= max(1,p_shareUp(rNat,is,splita) * 1.25);

p_shareUp(rNat,is,splita) $ ( nTries > 6 ) = 1;

```

## 3. Corrections to the desired tax rates are step-wise relaxed:

```

--- give some freedom in tax bounds (ar pulled towards zero anyhow in objective)

if ( nTries > 2,
    $$batinclude 'build\split_set_taxBounds.gms' 1.E-3
);
if ( nTries > 4,
    $$batinclude 'build\split_set_taxBounds.gms' 1.E-2
);
if ( nTries > 6,
    $$batinclude 'build\split_set_taxBounds.gms' 1.E-1
);
if ( nTries > 8,
    $$batinclude 'build\split_set_taxBounds.gms' 1
);

```

If the final solution has still a sum of infeasibilities above a certain threshold, the program will abort with an error message:

```

IF ( (m_split.suminfes > m_split.tolinfeas*m_split.numInfes*10),
    execute_unload "%scdir%\split_%dataset%.gdx" v_sam.l,v_dm.l,v_mm.l,v_fShare.l;
    option QCP=conopt;
    solve m_split using QCP minimizing v_hpd;
    abort "Split program could not balance SAM";
);

```

### 4.5.14 Processing the solution from the split problem

After successful balancing, the information is copied to the SAM:

```

--- assing results from dis-aggregated SAM entries to global SAM
SAM0(rNat,consider(is,js)) = v_sam.l(rNat,is,js);

```

The same is done with the domestic and import use vectors, afterwards, the information for split up commodities is deleted:

```

*
* --- assgin imports and domestic sales by agent
*
xnm0(rNat,consider(i,aa)) = v_mm.l(rNat,i,aa);
xdm0(rNat,consider(i,aa)) = v_dm.l(rNat,i,aa);

xnm0(rNat,i,aa) $ sum(is_i(spliti,i), 1) = 0;
xdm0(rNat,i,aa) $ sum(is_i(spliti,i), 1) = 0;

xnm0(rNat,i,a) $ sum(as_a(splita,a), 1) = 0;
xdm0(rNat,i,a) $ sum(as_a(splita,a), 1) = 0;

xma0(rNat,consider(i,aa)) = xnm0(rNat,i,aa) * (1+mintx(rNat,i,aa));
xda0(rNat,consider(i,aa)) = xdm0(rNat,i,aa) * (1+dintx(rNat,i,aa));

```

There are further statements not shown which refer to the CO2 and Non-Co2 emissions factors, assumed currently to be equal across the split-up commodities respectively activities.

Furthermore, behavioral parameters are copied over, as well as taxes, and entries for the aggregate ones deleted.

```

*
* --- assing behavioral parameters and tax rates (and other elements not comprised in SAM, but used by model)
*
sigmaw0(rNat,Spliti) = sum(is_i(spliti,i), sigmaw0(rNat,i));
sigmam0(rNat,Spliti) = sum(is_i(spliti,i), sigmam0(rNat,i));

eh0(rNat,Spliti) = sum(is_i(spliti,i), eh0(rNat,i));
bh0(rNat,Spliti) = sum(is_i(spliti,i), bh0(rNat,i));

sigmap0(rNat,Splita) = sum(as_a(splita,a), sigmap0(rNat,a));
sigmav0(rNat,Splita) = sum(as_a(splita,a), sigmav0(rNat,a));

*
* --- assign factor taxes
*
fcttxY0(rNat,f,splita) $ sam0(rNat,f,splita) = sam0(rNat,f,splita) * sum(as_a(splita,a), fcttxY0(rNat,f,a)/sam0(rNat,f,a));
fcttsY0(rNat,f,splita) $ sam0(rNat,f,splita) = sam0(rNat,f,splita) * sum(as_a(splita,a), fcttsY0(rNat,f,a)/sam0(rNat,f,a));

fcttxY0(rNat,f,A) $ sum(as_a(splita,a),1) = 0;
fcttsY0(rNat,f,A) $ sum(as_a(splita,a),1) = 0;

*
* --- assign import and export taxes
*
imptxY0(rp,rNat,SplitI) $ sam0(rNat,rp,splitI) = sam0(rNat,rp,splitI) * sum(is_i(splitI,i), imptxY0(rp,rNat,i)/sam0(rNat,rp,i));
exptxY0(rNat,rp,SplitI) $ sam0(rNat,splitI,rp) = sam0(rNat,splitI,rp) * sum(is_i(splitI,i), exptxY0(rNat,rp,i)/sam0(rNat,i,rp));

imptXY0(rp,rNat,i) $ sum(is_i(splitI,i),1) = 0;
exptXY0(rNat,rp,i) $ sum(is_i(splitI,i),1) = 0;

```

Equally, trade margins are copied (not shown). Afterwards, the SAM entries for the split-up commodities and activities are removed and some SAM entries re-calculated to avoid problems from rounding errors in the solution process.

```

*
* --- remove SAM entries referring to aggregated rows and columns
*
sam0(rNat,i,js) $ sum(is_i(spliti,i),1) = 0;
sam0(rNat,is,j) $ sum(is_i(spliti,j),1) = 0;

sam0(rNat,a,js) $ sum(as_a(splita,a),1) = 0;
sam0(rNat,is,a) $ sum(as_a(splita,a),1) = 0;

* --- Government account Taxes

sam0(rNat,gov,"indtx") = sum(js, sam0(rNat,"indtx", js)) ;
sam0(rNat,gov,"prdtx") = sum(a, sam0(rNat,"prdtx", a)) ;
sam0(rNat,gov,"imptx") = sum(i, sam0(rNat,"imptx", i)) ;
sam0(rNat,gov,"exptx") = sum(i, sam0(rNat,"exptx", i)) ;
sam0(rNat,gov,"fcttx") = sum(a, sam0(rNat,"fcttx",a)) ;
sam0(rNat,gov,"fctts") = sum(a, sam0(rNat,"fctts",a)) ;

* --- bilateral bops, bops and trade margins, and bop = foreign saving

sam0(rNat,rp,"bop") = sum(i, sam0(rNat,i,rp)) ;
sam0(rNat,"bop",rp) = sum(i, sam0(rNat,rp,i)) ;
sam0(rNat,"trdmg","bop") = sum(i, sam0(rNat,i,"trdmg")) ;
sam0(rNat,inv,"bop") = -(sum(rp, sam0(rNat,rp,"bop")) - sam0(rNat,"bop",rp)) + sam0(rNat,"trdmg","bop")) ;

```

There is a further section not detailed in here which handles the case of a non-diagonal make being introduced with the split.

#### 4.5.15 MRIO split

The introduction of MRIO split values and the possibility to dis-aggregate commodities in the global SAM is discussed in some detail in section at the end of that documentation.

#### 4.5.16 Post-aggregation of GTAPAgg to yield non-diagonal make matrices

☐ Apply SAM split
 ☒ Use manual re-aggregation

Manuel aggregation file

The code of CGEBox clearly separates activities from products and supports non-diagonal make matrices where one activity might produce several outputs and one output might be produced by several sectors. The current GTAPAgg facility does not support such a differentiation. The data driver therefore allows the user to adding aggregate definitions which define such relations. The necessary files must be stored under “gams\build\nonDiag”. An example is given to remove the differentiation between irrigated and non-irrigated crops in GTAP-Water:



```
*
* --- remove differentiation between irrigated and non-irrigated products
*      but keep differentiation for activities => non-diagonal make matrix
*
mapi("pdrn","pdri-c") = YES;
mapi("pdrn","pdrn-c") = NO;

mapi("whtn","whti-c") = YES;
mapi("whtn","whtn-c") = NO;

mapi("gron","groi-c") = YES;
mapi("gron","gron-c") = NO;

mapi("v_fn","v_fi-c") = YES;
mapi("v_fn","v_fn-c") = NO;

mapi("osdn","osdi-c") = YES;
mapi("osdn","osdn-c") = NO;

mapi("c_bn","c_bi-c") = YES;
mapi("c_bn","c_bn-c") = NO;

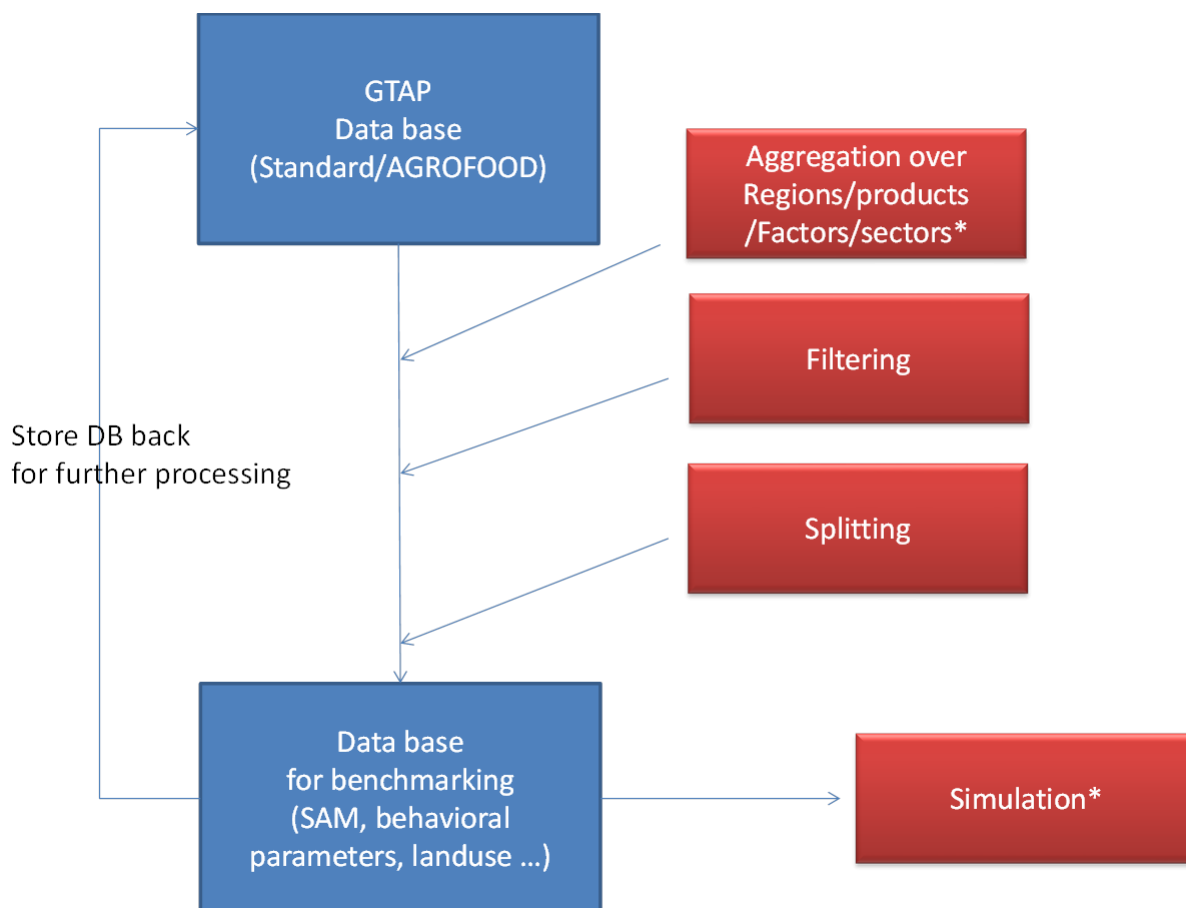
mapi("pfbn","pfbi-c") = YES;
mapi("pfbn","pfbn-c") = NO;

mapi("ocrn","ocri-c") = YES;
mapi("ocrn","ocrn-c") = NO;
```

Note here: the first element refers to the original product in the GTAP data base, while the second indicates the aggregated products. The example hence assumes a 1:1 mapping in GTAPAGG for all the crops, maintaining the differentiation between irrigated and non-irrigated. That distinction is now removed: the mapping to new products ending with “-c” are deleted (= NO) and are mapping to the irrigated ones instead. As the activity aggregation is maintained, no further changes are required to let the code generate the desired non-diagonal global SAM.

#### 4.5.17 Storing results from GTAPAgg as a new Data base for further processing

While the split utility has been successfully tested on extremely large split problem, such as introduced 50 additional sectors and products in the GTAP V10 Data base with full regional and sector detail, there are cases where it is useful to perform the split in steps. One might need, for instance, cost and expenditure shares of split-up products or activities to derive split factors. The data base driver of CGEBox therefore opens the possibility to store the outcome of the split process back as a data base in the same format as the data downloaded from the GTAP center. A typical processing flow is depicted below.



Graph: Data base generation for further processing

This option is activated by the selection box “Generate new DB” as shown below. The user has to fill in this case the name of the sub-folder under the data directory where the data base should be stored:

Input data	
Data base version	GTAPV10_2014
Data base variant (GTAP,POWER)	gtap_d
Aggregation file	full
Postfix for data set name	
Generate MRIO split factors	<input type="checkbox"/>
Apply SAM split	<input checked="" type="checkbox"/>
Split information file	fabio_agr
Split model	m_split
Regionalisation non-NUTS2	<input type="checkbox"/>
Use manual re-aggregation	<input type="checkbox"/>
Output all symbols to GDX	<input type="checkbox"/>
Generate new DB	<input checked="" type="checkbox"/>
Name for folder for new DB	FABIO

The outcome of the process is a folder which can comprise the following data files, where the presence of AEZ, air pollution and non-Co2 data base depends on their availability in the input data set.



#### 4.6 Model size and solution behavior

The model template including the GTAP-AEZ and GTAP-AGR modules was intensively tested under a suite of test shocks, applied simultaneously to all regions and sectors: 50 % reductions in tax rates (direct, consumption, factor, tariffs/export subsidies), 20% endowment changes and 10 % tfp shocks for all primary factors. The model was tested in three configurations: (1) the GTAP standard model with fixed allocation of foreign savings and (2) alternatively the global bank mechanism as a default in the standard GTAP model, and (3) the global bank mechanism in combination with GTAP-AGR and GTAP-AEZ. The tests used a full sector dis-aggregation (57 sectors) and varyingly sized regional aggregations (10,24,36,45,56,68). Additionally, a 10x10 model was solved for which the GTAP-AGR and GTAP-AEZ modules cannot be used due to missing sectoral detail. The parameterization was kept at defaults. Most of the tested models are large with regard to the number of sectors and countries compared to applications of GTAP reported in publications. All tests were also run solving the model as MCP in PATH and as a CNS in CONOPT.

GAMS version 25.5.4 was used, on a computer server, in combination with a beta-release of CONOPT4 which executes certain part of the NLP algorithm in parallel. We would like to acknowledge the continued support of Arne Drud, especially to let use CONOPT4 for the tests. Results with CONOPT3 are somewhat slower. The pre-solve algorithm with five pre-solve steps was applied. The aim of these tests was not only to ensure a stable numerical implementation of the model, but also to gain experiences with solution behavior on larger dis-aggregations.<sup>111</sup> The times reported in the table below are for a full model run without post-model processing switch on, but including the time needed to store all symbols in a GDX container, i.e. they cover loading the data base, model

---

<sup>111</sup> The aggregation definition files (\*.agg) used are found in the repository in the data directory. The test shocks can be found in the “scen” directory.

calibration, a benchmark solve and solving the shock.<sup>112</sup> The test show the expected more than linear increases in solution time if model size and complexity expands. Solution times with a fast multi-core laptop will be about double the times reported below.

**Table 2: Data bases used in the standard tests**

Model size	Filtering thresholds (relative minimum tolerance, # of transactions)	Resulting non-zeros in global SAM, including trade flows	Model size (GTAP Standard, maximal number of variables substituted out)
10x10	None	~4.700	~8.000
57x10	None	~35.000	~92.000
57x24	None	~157.000	~225.000
57x36	1%, 160.000	~120.000	~96.000
57x45	1%, 160.000	~130.000	~99.000
57x56	1%, 160.000	~133.000	~98.000
57x68	1%, 160.000	~135.000	~99.000

Note: The number of transactions accounted for during filtering is somewhat higher than the resulting non-zero SAM entries

Many of the test solves will actually run even somewhat faster without the pre-solves switched on. However, if the solver fails to solve the shock without any pre-solves quickly, it can often spent several minutes until all infeasibilities are removed. As the potential further gains in solution speed by switching off the pre-solves are limited, we opted to show in here results obtained with the default settings. For small models and special applications such as sensitivity analysis, it might however pay off to check if the overhead of using the pre-solves can be avoided.

---

<sup>112</sup> The distribution which includes the GUI also comprises the described above test suite for the model, realized as an input file for the batch utility (gui\testbatch.txt). It is recommended to run the test suite after changes to the model code.

**Table 2: Solution time with different data bases and model configurations, CNS**

Data base	GTAP standard, Fixed allocation of global savings	GTAP Standard, Global bank mechanism	GTAP Standard + GTAP-AGR+GTAP-AEZ Global bank mechanism
10x10	10 sec	10 sec	(not possible)
57x10	30 sec	30 sec	35 – 45 sec
57x24	1. min 30 sec	1 min 30 sec – 2 min	1 min 40 sec – 2 min 30 sec
57x36	1min	1 min – 1 min 30 sec	1 min – 1min 30 sec
57x45	1 min 10 sec	1 min 10 sec – 2 min	1 min 15 sec – 3 min
57x56	1.min 20 sec	1 min 20 sec – 3 min 20 sec	1 min 20 sec – 3 min 20 sec
57x68	1 min 30 sec	1 min 30 sec – 3 min	1 min 30 sec – 11 min <sup>113</sup>

Even if additional tests showed that rather large models such as a 57x82 variant can be solved even without any filtering at least on some larger shocks in around ten minutes, some of the shocks tested failed with a model using the full resolution of the data base (57x134), but also with certain larger shocks on the 57x82 variant in any reasonable time. The tests were only possible with the beta version of CONOPT4 which is distributed with newer GAMS versions, as CONOPT3 will exceeds its internal memory limit of 8GB on very large models (around 600.000 equations which implies a SAM with about the same number of entries). That implies that users might run into very long solution times under a combination of larger shocks and very detailed data bases or might even not be able to load the model into the solver.

Analysis of changes in simulated welfare changes under a multi-lateral trade liberalization scenario which dismantled 50% of all import tariffs and export subsidies indicate that filtering can influence substantially results obtained. Unfortunately, the same holds for changing the regional and sectoral resolution of the model. We can therefore generally only recommend running highly dis-aggregated models with limited filtering. For further detail on solution times and aggregation bias due to sectoral and regional aggregation or the use of filtering, refer to Britz et al. 2015 who provide a larger sensitivity analysis with the standard GTAP model in GAMS.

The tests have shown that in combination with the pre-solves, solving the model as a CNS with CONOPT is usually faster if the model is larger compared of using PATH and solving as a MCP. Very small models are typically faster solved with PATH. Having both solvers available for tests might

---

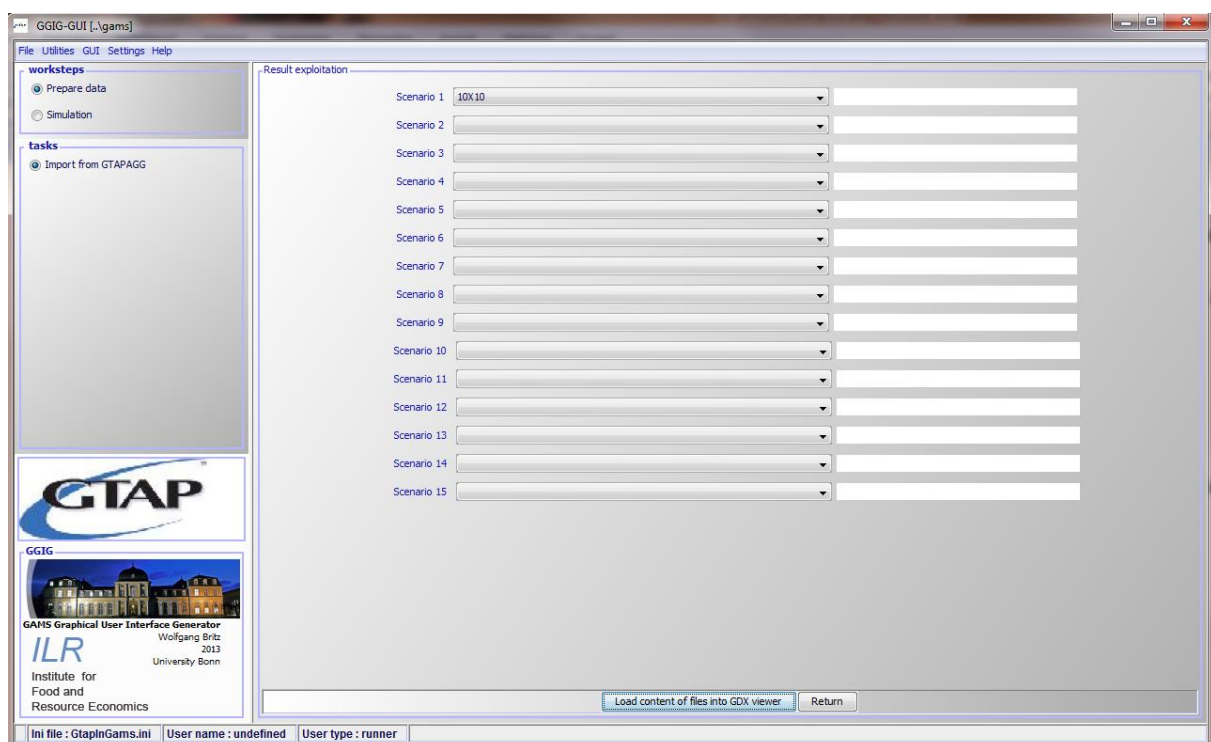
<sup>113</sup> The long solution times occurred under the direct tax cuts. As the standard GTAP model assumes that FDI is taxed at the same rate as domestic one, the direct tax cut impacts fully the expected returns to capital by foreign investors which drive the distribution of foreign savings under the global bank mechanism. A 50% reduction as simulated in the test shock can provoke very large differences in expected returns depending on the regional resolution of the data base. Thus, a new equilibrium can require massive changes in investments to decrease or increase the regional capital stock in order to drive marginal returns to capital up and down towards the global average. For some regions, that scenario might not feasible without fixing some variables at their bounds. Solving the model as a MCP should yield automatically that solution. However, solving as a CNS will yield infeasibilities in that case. Therefore, an algorithm is comprised in the GAMS code which tries to find, based on equations becoming infeasibility, which variables to fixed, which however requires to solve the model potentially several times. That algorithm can also fix similar problems which might not be related to the global bank mechanism.

hence pay off if solving large models is part of a project. And clearly, certain policy instruments such as production quotas are best captured by a KTT-condition embedded in a MCP model. Note that the GUI allows letting the model first be solved with CONOPT (as a CNS), and if that fails, to try a test with PATH (as MCP):

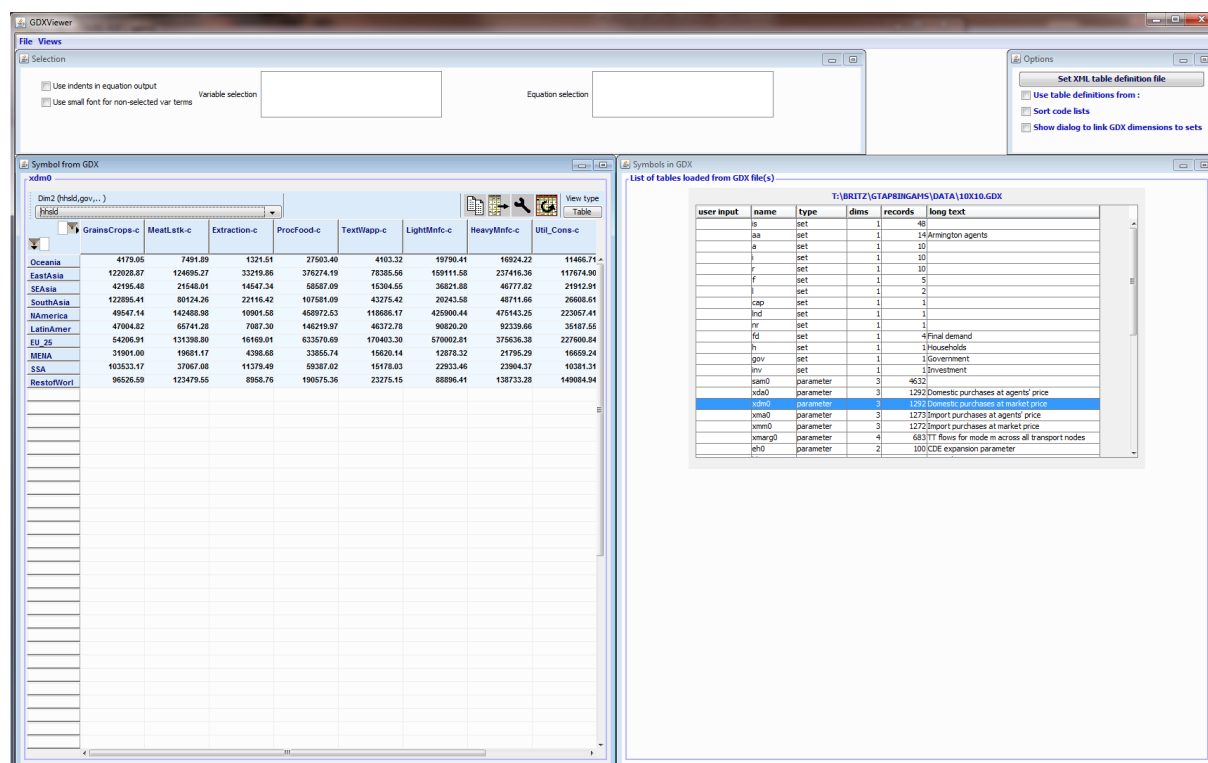
Checks for medium sized models have shown no differences in the results produced by CONOPT or PATH. That seems to indicate that once a model is declared feasible, the results can be used. Care should be clearly given when using PATH as a MCP solver after structural changes to equations: MCP solvers might e.g. accept a fixed variable paired to an equation which results in a non-square system, whereas solution as a CNS will throw an error in that case.

### 4.7 Inspecting the resulting data base

You can check if loading and filtering worked by pressing on “Exploit results”, selected as shown as output file generated, a GDX comprising different parameter and set definitions:



Once you press “Load content of files into GDX viewer”, you can inspect the individual symbols used in the model:



Double clicking on any on the symbols in the table on the RHS will open a new window as shown above.

The effect of using filtering and the rebalancing can be checked with the symbol “itrlog” which shows totals ,aggregated over commodities or sectors, for each rebalancing step (per country and “trace” at global level), the number of nonzero at global level and related changes. The example below shows the output from filtering a 32x34 case. Filtering removes about 76% of the original almost 135.000 non-zero items (see row “Total”). The line “curRelTol” shows the applied threshold in the iteration, which is increased stepwise to the desired maximum of 0.25%. Inspecting the relation between the dropped items and the cutoff used in that preliminary iteration can help to find a good compromise between model size and a too aggressive filtering where also more important transaction are wiped out.

	start	absFilt	itr0	itr1	itr2	itr3	itr4	itr5	itr6	itr7	itr8	itr9	itr10	itr11	itr12	done	delta	delta (%)
Total	134874.00	134874.00	134874.00	70372.00	61189.00	54179.00	51886.00	45676.00	42426.00	39275.00	36709.00	34697.00	32334.00	31778.00	31737.00	31732.00	-105142.00	-76.47
vftm	3774.00	3774.00	3774.00	3346.00	3163.00	2957.00	2957.00	2707.00	2584.00	2472.00	2256.00	2263.00	2179.00	2150.00	2147.00	2147.00	-1027.00	-43.11
vernd	35870.00	35870.00	35870.00	25333.00	22336.00	20032.00	19276.00	17081.00	15971.00	14861.00	14064.00	13296.00	12486.00	12267.00	12247.00	12247.00	-23623.00	-65.86
vtrcr	19970.00	19970.00	19970.00	16897.00	15246.00	13783.00	13345.00	11723.00	10934.00	10080.00	9421.00	8825.00	8215.00	8061.00	8045.00	8045.00	-11027.00	-57.82
vftm	35904.00	35904.00	35904.00	8474.00	6693.00	5588.00	5134.00	4359.00	3938.00	3572.00	3200.00	2920.00	2686.00	2659.00	2659.00	2659.00	-33247.00	-92.60
vftm	35904.00	35904.00	35904.00	14220.00	11022.00	9987.00	9428.00	8138.00	7388.00	6733.00	6164.00	5725.00	5338.00	5228.00	5219.00	5219.00	-30685.00	-85.46
vipm	1088.00	1088.00	1088.00	739.00	702.00	675.00	657.00	636.00	617.00	603.00	591.00	585.00	576.00	571.00	571.00	571.00	-517.00	-47.52
vdpm	1088.00	1088.00	1088.00	844.00	783.00	739.00	721.00	692.00	657.00	629.00	603.00	579.00	560.00	554.00	554.00	554.00	-534.00	-49.06
vipm	1088.00	1088.00	1088.00	391.00	260.00	238.00	214.00	205.00	199.00	193.00	185.00	184.00	176.00	176.00	176.00	176.00	-912.00	-83.32
vdpm	1088.00	1088.00	1088.00	219.00	185.00	169.00	157.00	145.00	138.00	131.00	125.00	120.00	118.00	118.00	118.00	118.00	-970.00	-89.15
curRelTol				0.03	0.05	0.08	0.10	0.12	0.15	0.17	0.20	0.22	0.25	0.25	0.25	0.25		
curAbsTo				0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		

## 4.8 Pre- or post-model aggregation

The standard case in GTAP applications is a project specific pre-model aggregation. Given the total size of the data base – there are already about 1 Million non-zero trade flows reported – pre-model aggregation can be hardly provided, it yields smaller models which are faster to solve, easier to debug and typically also numerically more stable with less small values being present. However, aggregation comes clearly at a price: not only might detail of interest for the application be lost, but also peaks of



policy instruments etc. wiped out. That can have important effects, Ko and Britz 2013<sup>114</sup> show in example applications that one might increase considerably simulated welfare from FTAs by simply increasing sectoral and regional detail.

CGEBox tries to soften the decision on what regional and sectoral aggregation level to use based on three features. Firstly, filtering and rebalancing of a data base can help to solve models with rather high number of sectors and regions (but clearly, the filtering process will also introduce aggregation bias). Secondly, the equations of the model had been carefully arranged; additionally scales and bounds for variables are introduced to stabilize the solution behavior of the solvers and to speed up solution time. And thirdly, post-model processing allows for a second aggregation step for reporting purposes. To do so, the user has to provide a second aggregation definition file (\*.agg, can be produced by GTAPagg). See also Britz and Van der Mensbrughe 2016<sup>115</sup> for discussion of solution times and aggregation bias.

---

<sup>114</sup> Ko, J.-H. and Britz, W. (2013) Does Regional and Sectoral Aggregation Matter? Sensitivity Analysis in the Context of an EU-South Korea FTA", <https://www.gtap.agecon.purdue.edu/resources/download/6313.pdf>

<sup>115</sup> Britz, W., van der Mensbrughe, D. (2016): Reducing unwanted consequences of aggregation in large-scale economic models - A systematic empirical evaluation with the GTAP model, *Economic Modelling* 59: 462-473

## 5. Scenario runs

### 5.1 Defining the scenario

#### 5.1.1 Implementation of shock files in GAMS

Running shocks with a CGE (or more generally an equilibrium model) is strongly related to the concept of a partition which defines what is endogenous and what exogenous, also called a closure. Different closures reflect that such a model comprises many variables or parameters, much more than equations. All these variables and parameters are possible candidates to become endogenous which implies that they will adjust to the shock while the exogenous ones will stay unchanged. Solving the model requires fixing exactly so many variables or parameters that the remaining number of endogenous variables is exactly equal to the number of equations. Fixing more than the number of equations will typically lead to infeasibilities, fixing less generally implies that the solution is not unique. Besides offering some default closures on the interface, the user can also provide its own closure files.

In both cases, shocks are only possible for non-endogenous elements of the model which can be technically either GAMS parameters or fixed variables where the lower is equal to upper bound. A symbol defined in GAMS as a parameter can however never become an endogenous variable. A shock file redefines the level of parameters in the model or fixes exogenous variables to a different value, away from the benchmark level. Consequently, the benchmark results of the endogenous variables do not longer provide a solution to the equations. The solver needs hence to search for a new combination of values for the endogenous variables which provide a fix-point for the new exogenous values.

The installation comprises a set of example shock files of which we discuss here two. We start with a shock which reduces all bi-lateral import tariffs and export subsidies, a partial multi-lateral trade liberalization (see `gams\scen\policy_shocks\tarifs.gms`). Such a shock file is always structured in two parts: a section which declare symbols, called before the solution loop, and a section, executing during the simulation loop, where parameters or fixed variable levels are updated.

The declaration section starts with the pre-processor command “\$ifthen “%includeMode%”==”decl”, that statement is necessary and must be found in all shock files. In our example below, we declare two parameters, one for shocking the import tariffs (`p_taxShockTariifs`) and one for shocking the export subsidies (`p_taxShockExpSubs`). As these parameters are only used in the shock file, their names can be freely chosen by the user as long as they do not clash with other symbol names. In our simple example below, we shock all exporters (first dimension), importers (third) dimension and products (second dimension), reducing the taxes and subsidies by 50%. In the declaration part, we only define the size of the shock and what to shock, it is not yet applied to parameters or variables in the model. The declaration code alone would hence not force the model away from the benchmark.

```
$iftheni "%inclMode%"=="decl"
*
*
*   --- define which bi-lateral combinations and commodities to shock
*       that can be achieved in multiple ways, we define here a multiplier
*
*       parameter p_taxShockTariffs(r,i,rp)
*                   p_taxShockExpSubs(r,i,rp);
*
*   --- here, we could e.g. also insert a table, read from
*       EXCEL or from a GDX container
*
*       0.9 would mean reduce by 90%
*
*       p_taxShockTariffs(r,i,rp) = 0.50;
*       p_taxShockExpSubs(r,i,rp) = 0.50;
*
$else
```

The second part after the \$else statement applies the defined levels of the shock to re-define new fixed levels of the two variables `imptx` and `exptx` on the LHS. That will lower as a first order effect the import prices faced by demanders and the tax income of the governments with subsequent second and third round effects towards a new equilibrium.

The RHS of the assignments shown below reduces the benchmark level (“%t0%” is either the year “t00” in a recursive-dynamic run or the benchmark point “bench” in a comparative static one) of the exogenous variables according to the desired shock, implementing it in equal steps over the simulation horizon. How that stepwise application of the shock is programmed in GAMS is discussed next. The set `tSim` in a comparative static simulation only comprises two elements, “bench” and “sim”. The shock file is only called for “sim” to leave the benchmark point unchanged; the position of the element “sim” in the set `tSim` is second. Hence, `tSim.pos-1` yields unity. The divisor is the number of elements in the set, i.e. `card(tSim)`, minus one, which is also equal to unity. The code will for a comparative-static run multiply the shock as declared in the parameters above with one, in other words, it is fully applied in the one and only simulation run with the model.

Now, we look at the case with a recursive-dynamic run over 10 years, the elements in the set `tSim` would be called `t00`, `t01`, ..., `t10`. If the driver calls the code for the element `t01`, which is the second element in the set, the numerator is  $(2-1) = 1$  as in the example above while the denominator is  $(11-1)=10$ , such that 10% of the shock will be applied. In each year, the numerator will increase by one, such that in the next year 20% of the shock will be present and so forth.

```

*
* r = importer regions, i=products, rp= exporters, tsim= simulation time point
* imptx: Bilateral import taxes

imptx.fx(r,i,rp,tsim) $ (imptx.l(r,i,rp,tsim) $ p_taxShockTariffs(r,i,rp))
= imptx.l(r,i,rp,"%T0%")
*
* --- this distributes the reduction in equal steps over the
* simulation horizon (or in step-wise shocks to ease solving
* if mimic comp-static is switched on)
*
* * (1-p_taxShockTariffs(r,i,rp) * (tsim.pos-1)/(card(tsim)-1));
*
* --- note that we only cut export subsidies, i.e. the cases where exptx < 0
*
exptx.fx(r,i,rp,tsim) $ ((exptx.l(r,i,rp,tsim) lt 0 ) $ p_taxShockExpSubs(r,i,rp))
= exptx.l(r,i,rp,"%t0%") * (1-p_taxShockExpSubs(r,i,rp) * (tsim.pos-1)/(card(tsim)-1));

```

The mechanism with the step-wise changes can also be used in a pseudo recursive-dynamic run without capital accumulation or other features relating to dynamics. In that case, the sole purpose is to apply not the full shock in one go to the model. That might ease solving strong shocks on complex models. Similar files are available which define shocks consumption, factor and direct taxes.

Another type of often applied shocks changes productivity levels. The related variables are mostly called lambda + postfix, where the postfix is the name of the variable to shock, as shown in the table below:

Shifter variable	Meaning
axp(r,a,t)	Production function frontier
Lambdand(r,a,t)	Demand for intermediate demand composite
lambdava(r,a,t)	Demand for value added composite
lamdaf(r,f,a,t)	Factor demand, separate for each factor
lambdaio(r,i,a,t)	Intermediate demand, separate for each commodity

The code below shows the content of the example file “gams\scen\end\_and\_tp\_shocks\factor\_specific\_tp.gms”. It increases the factor productivity for all regions, activities and factors by 10%.

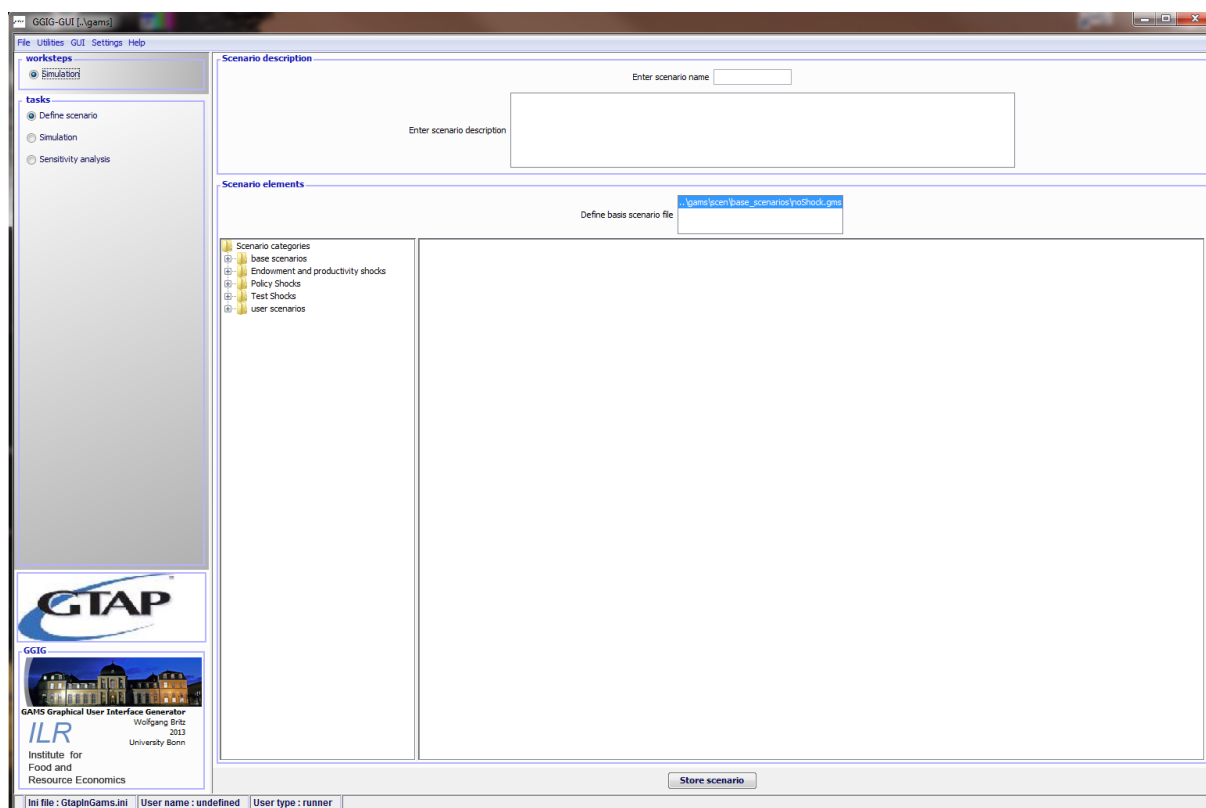
```
$iftheni "%inclMode%"=="decl"
*
*   parameter p_tpf(r,f) "Shifter for factor demand";
*
*   --- here, we could e.g. also insert a table, read from
*       EXCEL or from a GDX container
*
*   p_tpf(r,f) = 1.1;
*
$else
*
*   r= region, f=factors, a=activities, t=time
*   If Lambdaf is not equal across factors, implies Hicks non-neutral TP
*
*   lambdaf.fx(r,f,a,tsim) $ p_tpf(r,f)
*
*   --- this distributes the reduction in equal steps over the
*       simulation horizon (or in step-wise shocks to ease solving
*       if mimic comp-static is switched on)
*
*       = ( 1 + (p_tpf(r,f)-1) * (tsim.pos-1)/(card(tsim)-1)) ;
$endif
```

Note again the dollar operation “\$ p\_tpf(r,f) on the LHS. That ensures that only non-zero changes are used in the shock. If, for instance, values for p\_tpf are only found for one region, the productivity shifters for all regions will stay at their benchmark point. The usual way for the experienced coder is to define shock files in text editor such as GAMIDE, alternatively, the simpler build in scenario editor can be used.

The most important step after a shock was run is to make sure that the desired changes show up in the model. One way to to so is to let the GAMS code store all symbols into a GDX file (see also next section). Opening a symbol such as lambdaf in the example above should then show the desired change from the benchmark to simulated point. Productivity shifters and tax rates can also be inspected by pre-define views in the exploitation tools.

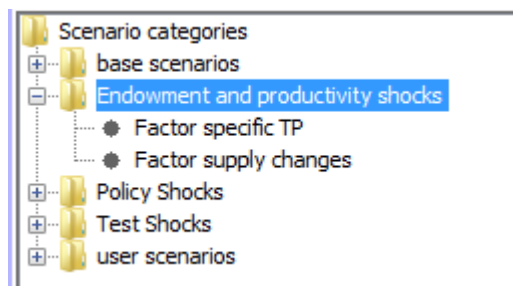
### 5.1.2 Scenario editor

The interface helps with the definition of scenarios – it has a set of example files with shocks which can be combined and edited directly in the GUI. Alternatively, you can use any text editor and define a shock in GAMS and store the file somewhere under “\gams\scen”. That is the recommend way to work in order to define complex scenarios and for users familiar with GAMS coding. In order to use the in-built scenario editor instead, click on “Define scenario” and the interface should look like below.

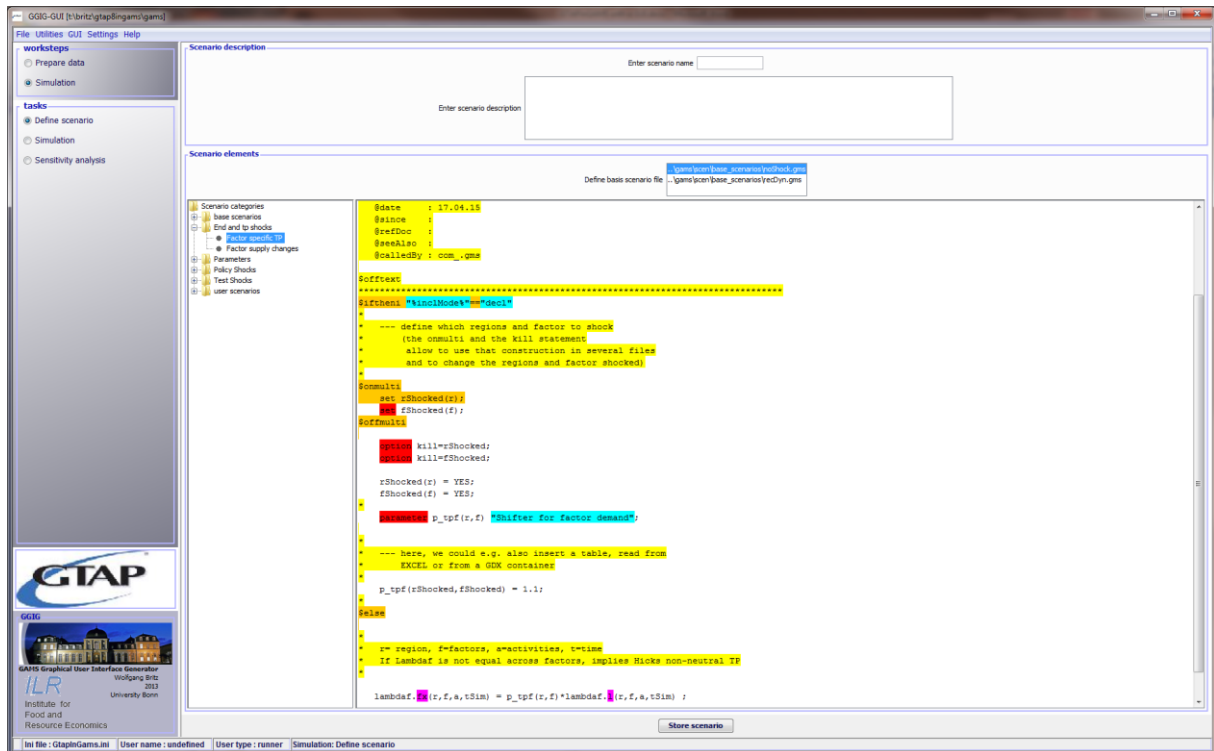


Currently, there are two base scenarios:

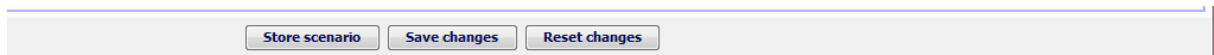
- **No shock** – that is simply an empty file
- **RecDyn** – a test implementation for a driver for a recursive-dynamic baseline.
- In order to add the content of pre-defined shock files to your base scenarios, click on a tree element, e.g. “Endowment and productivity shocks”. You should be able to see a list of files as below:



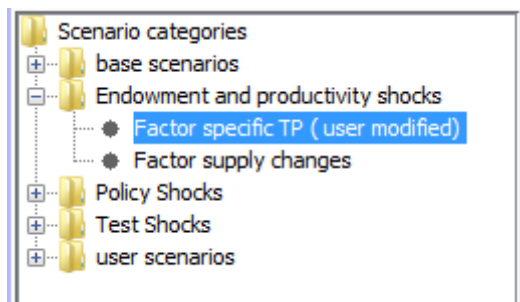
Double-click for example on “Factor specific TP”, and the content of that file is loaded in the editor on the right as seen below:



You can now modify the content with the editor by typing directly in the right hand window. Once you have edited the code, additional buttons will appear:

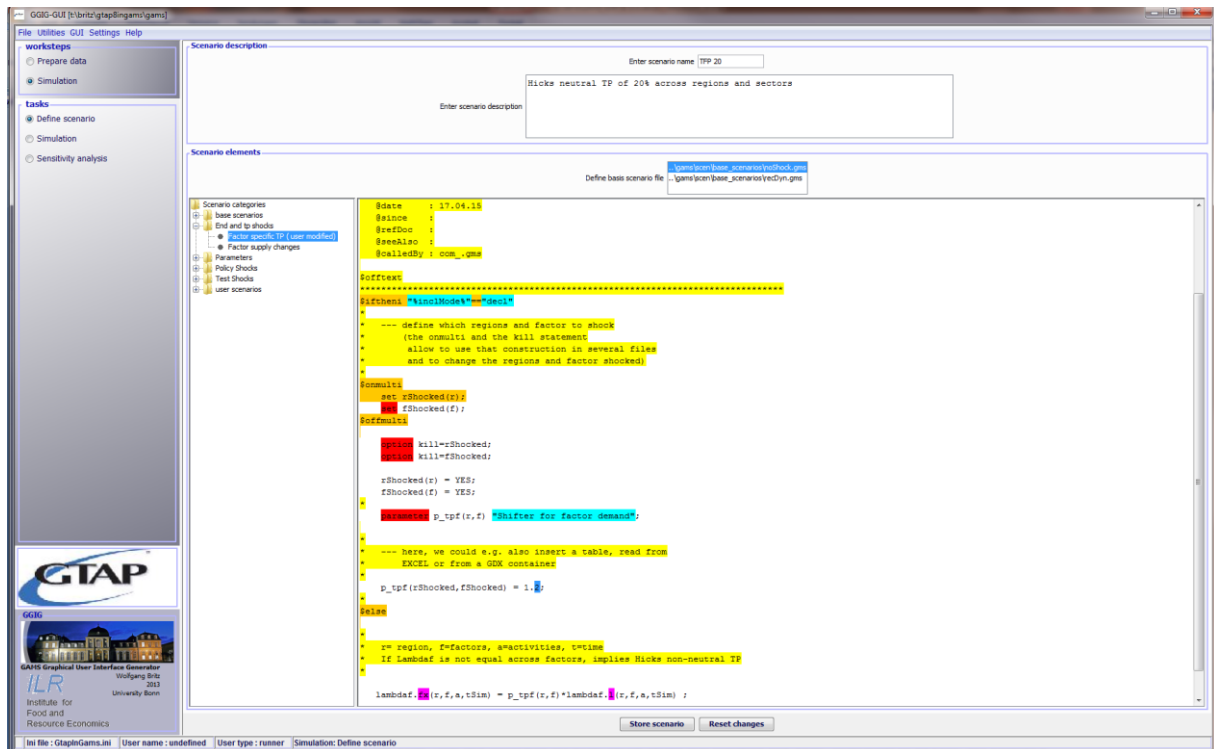


The list with the scenario groups and shock files will now indicate that you have made changes to the shock file.

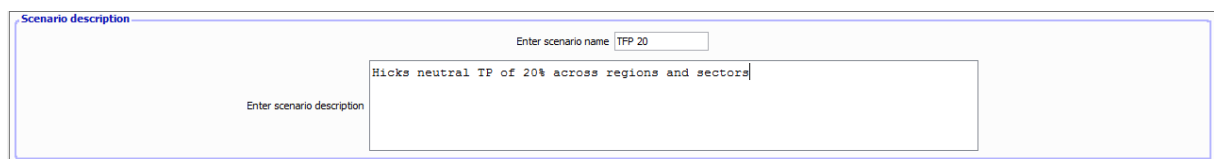


In order to save your edits, press “Save changes”. If you do not press “Save changes”, your edits won’t make it in final shock file generated. You can now double click on further files to add them to your shock file. The files you have currently combined are shown in blue; you can also deselect some of these files again. Files which were edited by you (only a temporarily copy is changed) have a “(user modified)” appended to their name. Edits in such files are shown in blue (in the example below, the factor 1.1 was changed to 1.2).

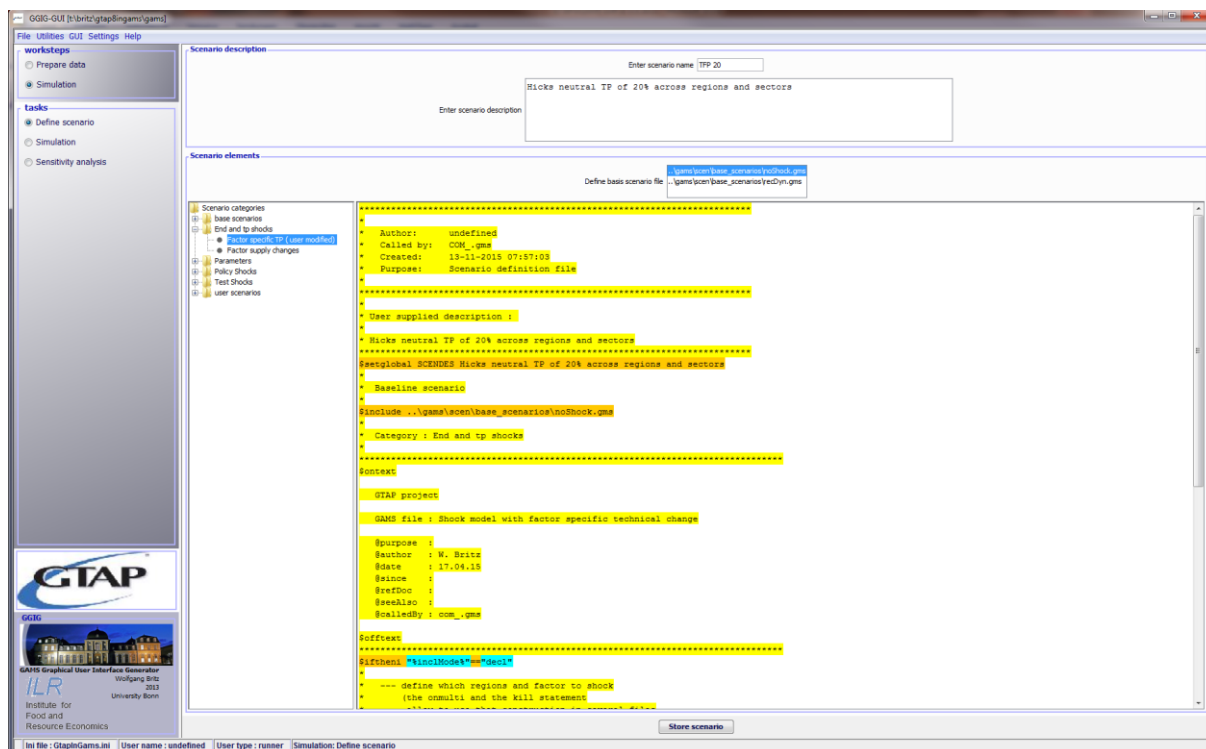




Once you have chosen the files you want to combine and potentially edited (and saved!) your changes, you should now enter name for the shock file to create, e.g. TFP20 and a description of the scenario as shown below:



Afterwards, press “Store scenario”:

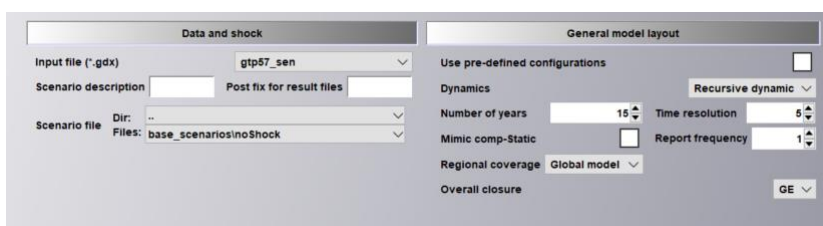


The window shows now the file as stored in the disk and included by the standard GTAP in GAMS model driver, when you use it to run a scenario.

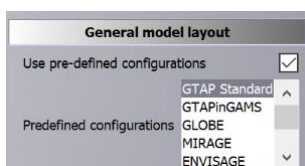
### Notes:

1. The scenario editor can also be used to edit parameters settings for the model. Several files which show the parameters used in different modules are provided.
2. Unchanged files are inserted in the new shock file with include statements, otherwise, the full content of the file with your edits is comprised in the new file.
3. The new file is stored in the directory “gms\\scen\\user\_scenarios”.

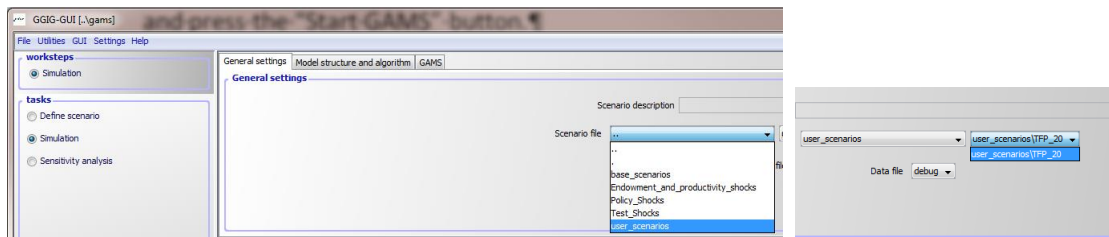
## 5.2 Running the scenario



In order to use the GTAP Standard configuration, check the box “Use pre-defined configurations” and chose “GTAP Standard”:



In order to run the scenario, click on “Simulation”, select first the directory with the user scenarios as shown below, and then pick the second box with scenario you just generated

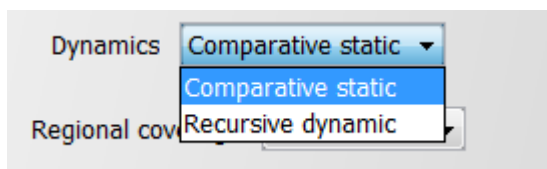


and press the “Start GAMS” button.

That should run your counterfactual against the reference and generate results in a GDX file.

The tab “General settings” gives you the following additional input possibilities:

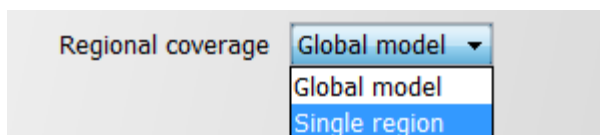
- Entering a scenario description: Normally, the output files generated are named after the scenario file used. This name can be overwritten with a non-empty content of this text field.
- Post fix for result files: The post fix is appended to the names of the output files. It can e.g. used to differentiate versions using the same shock, e.g. results generated based on different closures or by using different model configuration.
- Choice between the Comparative static or Recursive dynamic version of the model:



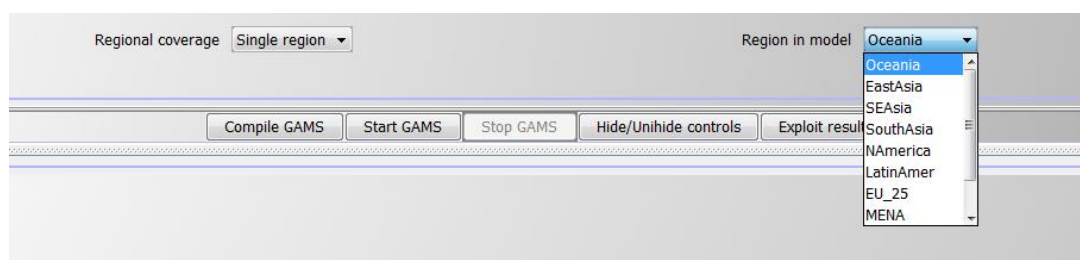
If the “Recursive dynamic” option is switched on, the following options are shown:



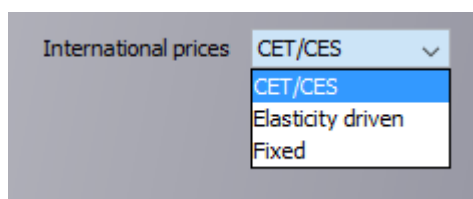
1. The number of years defines the simulation horizon. The actual years simulated depend on the benchmark year of the data set chosen (e.g. 2014 as the default with GTAP Version 10).
  2. The time resolution defines which years of the horizon are simulated (time steps). The first and last year are always included.
  3. The report frequency defines for which of the simulation years the results are saved. It must be multiple of the time resolution or equal to the time resolution.
  4. “Mimic comp-static” allows switching off capital accumulation such that a comparative-static model is solved in several steps. That is useful for step-wise introduction of shocks, also called “shock slicing”.
- Choice between the Global Model or Single Region version of the model:



If the single region mode is switched on, a drop down box appears where the region can be chosen:



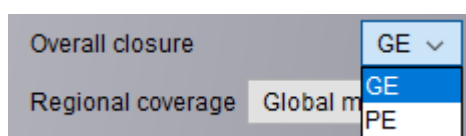
Furthermore, the way international prices and related to that, how import and export flows are treated can be chosen in case of a single region model:



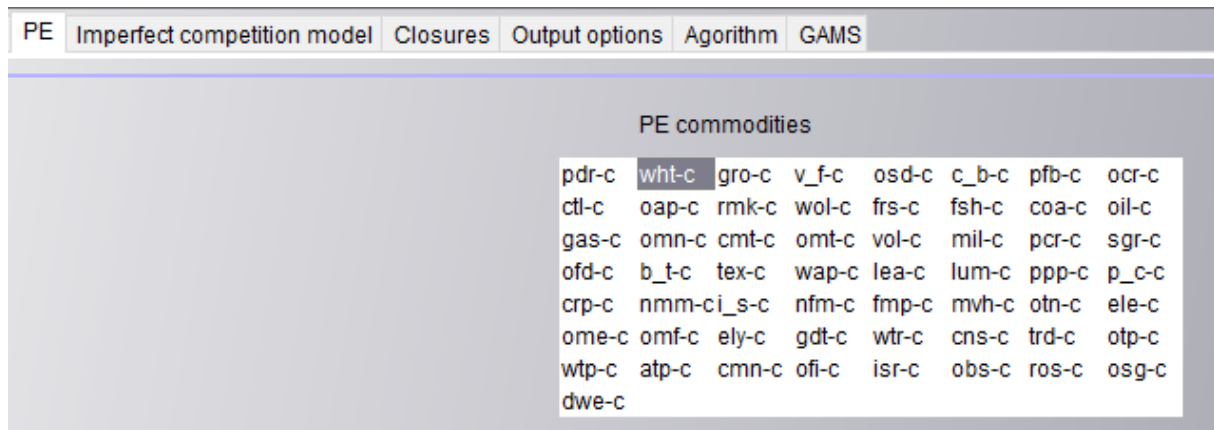
- CET/CES: Imports to other regions react to changes in the region's cif prices according to the share equations of the lower Armington nest at fixed total import quantities and aggregate import prices. If CET elasticities for exporters to the region are not infinite, the lower CET nest at the export destination are active and fob prices react accordingly. Otherwise, they are fixed.
- Elasticity driven: in that case, aggregate import and export prices of the region are driven by iso-elastic functions of the region's total import and export, respectively.
- Fixed: import and export prices of the region are fixed. That will typically only make sense if a CET between domestic and export supply is active.

The reader should note that the alternative to a single region model is a layout with a two-region aggregation of a GTAP Data Base: the single region against all other regions aggregated into one Rest-of-the-World model region. This layout will automatically reflect cases where the small country assumption makes limited sense, for instance, cases where the region singled out has a large export or import share in selected market.

- Choice between the General Equilibrium or Partial Equilibrium closure of the model:



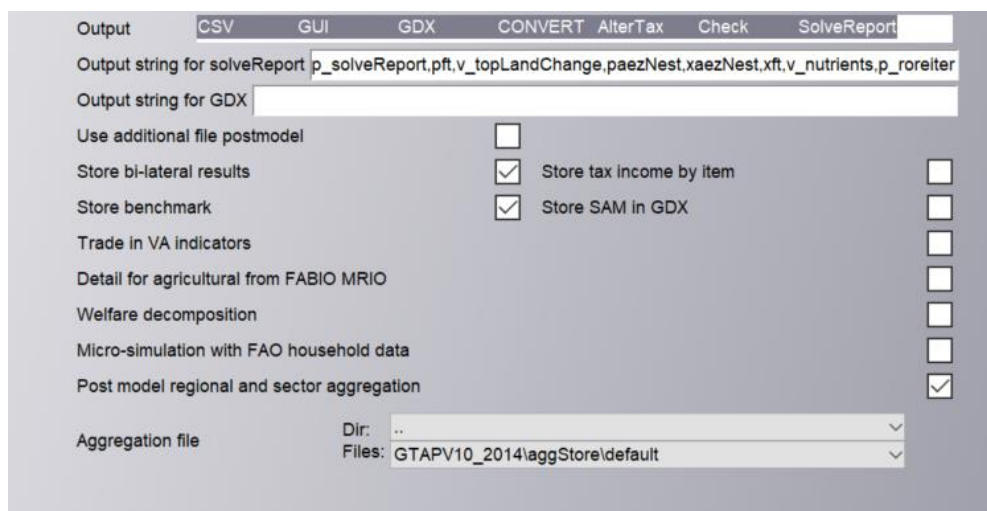
The GE closure is the usual set-up for a General Equilibrium model. The PE closure allows solving for only one; several or all commodity markets while clearing factor markets at fixed income. The product markets which clear can be chosen on the tab "PE":



Please note that the usual welfare interpretation of a CGE is no longer valid if income is exogenous and/or only some markets clear.

### 5.3 Output options

The output option give some flexibility how to inspect the final model solution, which typically will relate to the levels (and derived from there, changes) in endogenous variables. Related to this, it is useful to also check changes in exogenous elements to make sure that the shock was correctly implemented. The output options panel looks like this if only GUI (the default) is selected:



Also, make sure that output for the GUI is selected:



The different output options are sketched as:

- CSV will generate a SAM and some other core results and store it in CSV format under “results/run”
- GUI will generate a GDX container with a parameter for use with the exploitation tools (see below)
- GDX will store all other GAMS symbols (parameters, variables, equations, sets etc.) in a GDX container, including the output parameter for the GUI if GUI selected. The user can overwrite this behavior by inputting a list of symbol to export in the text filed “Output string for GDX”.

- CONVERT will generate output for the “Equation and Variable viewer” which is similar to AnalyseGE (see section below).
- AlterTax will store the generated SAM and some vectors back in a format which can be used as input for the model and thus allows constructing a new benchmark based on a model run.
- SolveReport will store during simulation a parameter `p_solveReport` in a GDX file in the scratch directory, for instance, to track the behavior of the model in recursive dynamic run.

Output string for solveReport `p_solveReport`

The user can add, separated by comma, further symbols to be stored in the GDX container.

The user can add its own file for post-model reporting, for instance, to calculate summary indicators by switching “Use additional file postmodel” on. In this case, a user provided file can be selected. Such files are stored in the directory “/gams/scen/post\_model”. This is only useful if also “GDX” output is switched on and the “Output string for GDX” is left empty or comprises the generated report parameter.

Use additional file postmodel	<input checked="" type="checkbox"/>	Postmodel file	empty
-------------------------------	-------------------------------------	----------------	-------

Most other options are relevant for GUI output only:

Store bi-lateral results	<input checked="" type="checkbox"/>	Store tax income by item	<input type="checkbox"/>
Store benchmark	<input checked="" type="checkbox"/>	Store SAM in GDX	<input type="checkbox"/>

The option “Store-bi-lateral results” will put bi-lateral trade flow information into the result set. For data sets with many regions and products, swichting this off can dramatically reduce the size of the data set generated. Similarly, “Store tax income by item” will reduce the number of items stored for each transaction. Normally, for each transaction, the quantity index, the price index, the value (quantity times price), the tax rate and also the value of quantity times (price plus tax) will be stored.

The same holds for the post-model aggregation. The screenshot below shows what post model aggregation contributes: in most views, results are not only available in the regional and sector dis-aggregation of the underlying data base, but additionally also aggregated results as shown below.

**Demand, by product [0]**

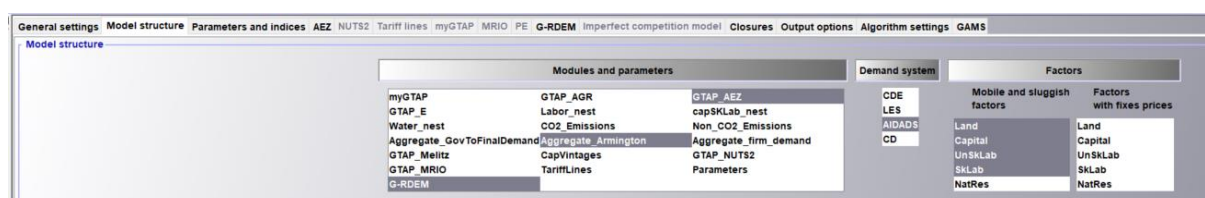
Regions: World | Items: Quantity | Origins: Total

**TARIFFS**

	Total	Household demand	Government demand	Intermediate demand	Investment demand	Export demand	International transport demand
<b>Total</b>	<b>137582,97</b> 0,41%	<b>31212,10</b> 0,30%	<b>9534,91</b> 0,02%	<b>54631,16</b> 0,01%	<b>12416,49</b> -0,69%	<b>23091,78</b> 2,03%	<b>591,84</b> 4,30%
<b>Agriculture</b>	<b>2058,32</b> 0,43%	<b>800,55</b> 0,29%	<b>1,87</b> 12,82%	<b>504,73</b> 0,58%	<b>0,57</b> -0,73%	<b>413,50</b> 0,78%	
<b>Mining and Extraction</b>	<b>5474,23</b> 2,91%	<b>58,66</b> -1,99%		<b>2484,97</b> 0,57%	<b>1,05</b> 0,20%	<b>1542,53</b> 6,36%	
<b>Light Manufacturing</b>	<b>17319,36</b> 1,02%	<b>3950,65</b> 1,51%	<b>18,43</b> 1,91%	<b>4854,85</b> 0,74%	<b>985,41</b> -0,50%	<b>5116,43</b> 1,23%	
<b>Heavy Manufacturing</b>	<b>32869,18</b> 0,68%	<b>2062,65</b> 0,66%	<b>53,34</b> 0,27%	<b>15055,72</b> -0,24%	<b>2527,85</b> -0,77%	<b>11201,62</b> 2,35%	
<b>Utilities and Construction</b>	<b>11765,99</b> -0,44%	<b>509,02</b> 0,28%	<b>9,54</b> -0,57%	<b>3485,60</b> -0,59%	<b>7415,18</b> -0,64%	<b>292,50</b> 5,26%	
<b>Transport and Communication</b>	<b>22034,38</b> 0,20%	<b>9171,68</b> 0,21%	<b>67,26</b> 0,88%	<b>10126,40</b> 0,09%	<b>593,31</b> -0,63%	<b>1725,47</b> 1,05%	<b>591,84</b> 4,30%
<b>Other Services</b>	<b>46061,50</b> 0,01%	<b>14658,88</b> -0,01%	<b>9384,46</b> 0,01%	<b>18118,90</b> 0,01%	<b>893,11</b> -1,15%	<b>2799,73</b> 0,40%	
<b>Total</b>	<b>137582,97</b> 0,41%	<b>31212,10</b> 0,30%	<b>9534,91</b> 0,02%	<b>54631,16</b> 0,01%	<b>12416,49</b> -0,69%	<b>23091,78</b> 2,03%	<b>591,84</b> 4,30%
<b>Agriculture</b>	<b>2058,32</b> 0,43%	<b>800,55</b> 0,29%	<b>1,87</b> 12,82%	<b>504,73</b> 0,58%	<b>0,57</b> -0,73%	<b>413,50</b> 0,78%	
<b>Paddy rice</b>	<b>21,36</b> -1,22%	<b>6,28</b> -0,03%	<b>0,19</b> -1,34%	<b>11,79</b> -1,76%		<b>1,55</b> -1,52%	
<b>Wheat</b>	<b>99,08</b> -0,12%	<b>12,88</b> 0,28%		<b>26,67</b> -0,05%		<b>29,76</b> -0,24%	
<b>Cereal grains nec</b>	<b>124,97</b> 0,45%	<b>48,10</b> 0,76%	<b>0,42</b> 22,49%	<b>17,66</b> -1,59%		<b>34,18</b> 0,75%	
<b>Vegetables, fruit, nuts</b>	<b>910,70</b> 0,70%	<b>477,79</b> 0,29%		<b>252,22</b> 1,14%		<b>114,20</b> 2,43%	
<b>Oil seeds</b>	<b>133,49</b>	<b>8,28</b>		<b>58,67</b>		<b>33,27</b>	

## 5.4 Model structure, parameterization and factor markets

You might also select from different model setups:



### 5.4.1 Modules and parameters

Currently, different modules and extensions as shown above can be added to the GTAP standard model. Please note that these modules are not a full replica of the GEMPACK implementations. The labor nest – which is also part of GTAP-E – depicts substitution between skilled and unskilled labor. The “CapSkLab nest” depicts substitution between capital and skilled labor.

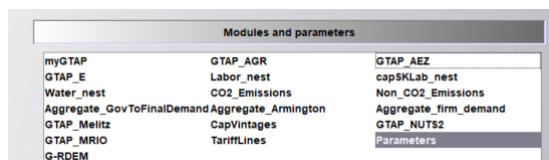
The “Aggregate intermediate demand” extension reduces the number of Armington agents to four: final, government and investment demand and aggregate intermediate demand. The latter replaces the sector specific nests in the GTAP standard model. “Aggregate Armington” makes a step further by using the same shares for domestic and import demand for all agents.

GTAP\_NUTS2 (see section “Sub-national dis-aggregation of production and factor markets in CGEBox”, page 123ff) provides a dis-aggregation of the production function and factor markets to



sub-regions, assuming homogenous output products from the regions in a nation. Given current data availability, that module works for European countries, only.

In order to introduce non-default parameters (such as a CET or non-zero substitution between ND and VA), switch “Parameters”

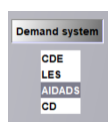


and selected the desired file on the “Parameter and

Indices” tab, see section 5.6.1, page 342.

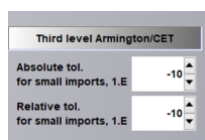
### 5.4.2 Demand system

The modular approach covers also the choice of the demand system for private households. Specifically, the standard CDE demand system found in GTAP can be replaced by a Linear Expenditure System or a Cobb-Douglas representation.



The LES system will determine the marginal budget shares based on income elasticities derived from the parameterization of the CDE demand system. The commitment terms are derived by assuming an 80% share of non-committed income, ensuring that commitments are non-negative and do not exceed 50% of observed demand quantities. The CD system uses the observed budget shares to determine the share parameters.

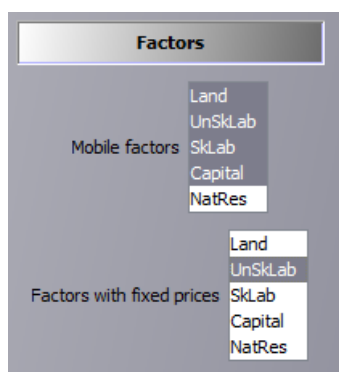
The tolerance for small imports steers the third level of the Armington system and additionally the CET, if switched on. The option can only be activated as developer and otherwise not visible.



That features found in GLOBE uses a Leontief relation between total imports respectively exports and small trade flows. That can overcome numerical stability issues with tiny trade flows and decreases overall model complexity. The user should note that it is not possible to use simultaneously a Leontief relation on the import and the export side for the same flow. Therefore, the code will not use the Leontief on the export side if already active on the import side.

## 5.5 Factors

The first selection possibility allows defining which factors are mobile – potentially combined with a CET approach to model sluggish factor mobility – and which are sector specific and thus immobile.

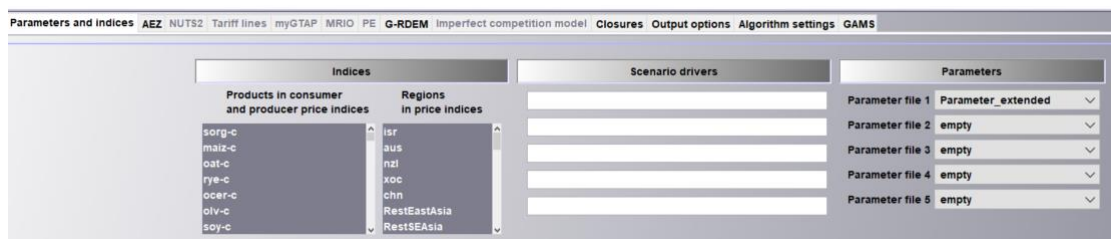


The second selection allows fixing factor prices - either economy wide or sector specific, depending on the choice above – such that their stock is endogenously adjusting. That is an often found solution to model unemployment based on a fixed wage rate. The reader should note that the model can also be parameterized with factor supply elasticities to model e.g. a wage curve approach or to introduce a land supply curve.

## 5.6 Parameters and indices

What is shown on the panel depends on whether the “Parameters” module is switched on and if the user is registered as a developer or not.

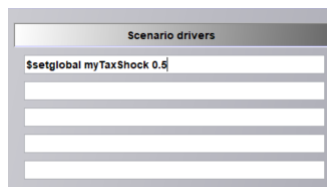
### 5.6.1 Indices



The indices are only visible for a developer; otherwise, all products and regions are used as a default. The selection of parameter files was discussed above.

### 5.6.2 Scenario drivers

The scenario drivers allow to write text directly in the include file. Consider the following example:



The input from the interface is directly inputted in the include file “com\_inc.gms”:

```
* ---- Scenario drivers
$setglobal myTaxShock 0.5
```

That allows to e.g. use that global variables in a shock file to change the size of the shock without the need to modify the GAMS code. That for instance the following extension of the example shock file “scen/policy/tariffs.gms”:

```

imptx.fx(r,i,rp,tsim) $ (imptx.l(r,i,rp,tsim) $ p_taxShockTariffs(r,i,rp))
= imptx.l(r,i,rp,"%T0%")
*
* --- this distributes the reduction in equal steps over the
* simulation horizon (or in step-wise shocks to ease solving
* if mimic comp-static is switched on)
*
* (1-%myTaxShock%*p_taxShockTariffs(r,i,rp) * (tsim.pos-1)/(card(tsim)-1));

```

The global variable is here introduced to change the size of the shock directly from the interface. That option is helpful for batch files to run multiple variants of the same basic shock as shown below. Note that the names after the \$setglobal can be chosen by the user as preferred.

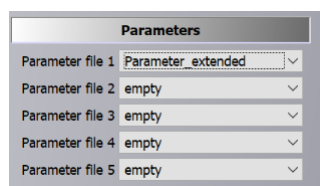
```

for trade = 0 \ 1
for migr = 0 \ 1
for savf = 0 \ 1
    ScenDriver1 = $setglobal scenDriver1 %trade%
    ScenDriver2 = $setglobal scenDriver2 %migr%
    ScenDriver3 = $setglobal scenDriver3 %savf%
    ScenDriver4 = $setglobal scenDriver4 %trade%
    batchName = brexit_t%trade%_m%migr%_s%savf%_n%trade%
    Scenario description = brexit_t%trade%_m%migr%_s%savf%_n%trade%
    execute=Gamsrun
endfor
endfor
endfor

```

### 5.6.1 Parameters

Non-Default parameter files can be used with the model if the module “Parameters” is switched on. In that case, the tab parameters and indices allows to selected up to 5 different files which comprises changes to the default parameterization:



These files are stored in the directory “gams\scen\Parameters”. The user can hence change existing files or add new ones which then can be selected from the interface.

The possibility to use non-default parameters is important as certain model extensions are driven by parameter settings. To give an example: in the standard GTAP model, the transformation elasticity between domestic output and exports is infinite, such that producer prices between the two destinations are equal and physical balancing is used. The GAMS version allows alternatively using a CET structure where prices differs and consequently, a non-linear quantity aggregator is used. That feature can be switched on by providing a parameter file in GAMS format where a CET elasticity different from infinity is set.

**Note:** Modules and extensions and the use of non-default parameters are disabled if pre-defined configurations are used.

## 5.7 Closures

The screenshot shows a configuration window with several dropdown menus and a checkbox. The options are as follows:

Label	Selected Option
Residual region	Senegal
Foreign savings	Global equal returns to capital
Government	Tax income
Final consumption	Spending
Regional reference price	Exchange rate
GDP price deflator	Fisher

There is also a checkbox labeled "Use additional closure file" which is currently unchecked.

As shown above, the code supports alternative closures different from the ones used in the standard GTAP model:

- For **foreign savings**, besides the default global bank mechanism which leads to identical expected returns to capital across all regions, the model can run with a fixed global allocation of investments and with fixed foreign savings. These closures are only available in the global model set-up. The balance of trade (BOP) solves for the exchange rate, it requires that the factor price index is used as the regional numeraire.

The screenshot shows the 'Foreign savings' dropdown menu with the following options:

- Global equal returns to capital (selected)
- Global equal returns to capital
- Fixed global allocation of investments
- Fixed foreign savings
- BOP

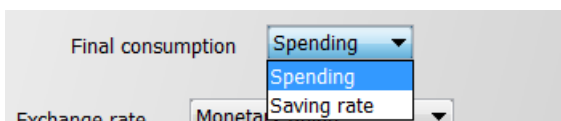
- The standard closure for the **government** account is to calculate tax income at given tax rates, and to let the top level utility function allocate a certain share of regional income to government consumption. The reader should note that there is no immediate relation between changes in tax income and government consumption. Changes in government consumption relative to private consumption and regional saving depend on the one hand on changes of the related price indices and on the other on the elasticity of the private consumption share to changes in overall utility. As such, there is no closed government account in the regional household approach, one could argue that implicitly government saving adjust. The alternative closures endogenize tax shifters either on all or selected products in consumption or for all or selected factors while keeping the real tax income for government consumption fixed, using the government price index to define real tax income. These two closures come closer to a representation of separate government account.

The screenshot shows the 'Government' dropdown menu with the following options:

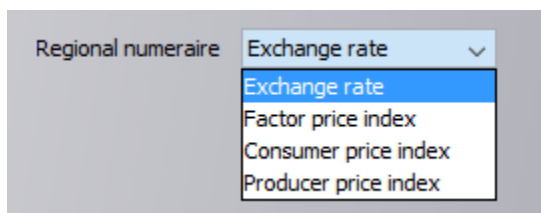
- Tax income (selected)
- Tax income
- Consumption taxes
- Factor taxes
- Direct taxes
- Savings
- Consumption

For the closures modifying taxes can be further detailed by applying the shifters. The closures where “savings” and “consumption” adjust are only available with the myGTAP extensions where a separate government account is present.

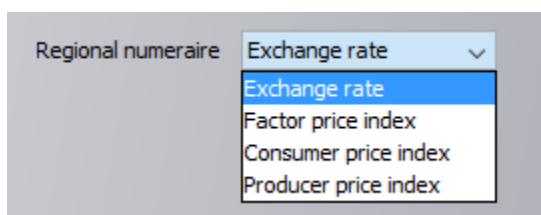
- The standard closure for **private consumptions and saving** is to have flexible budget shares for private and government consumption and regional savings. The alternative closures allow to fix private consumption spending and to let the saving rate adjust.



- The user can also use a regional reference price. That does not influence the results (quantities, welfare impacts), but simulated price levels will differ.



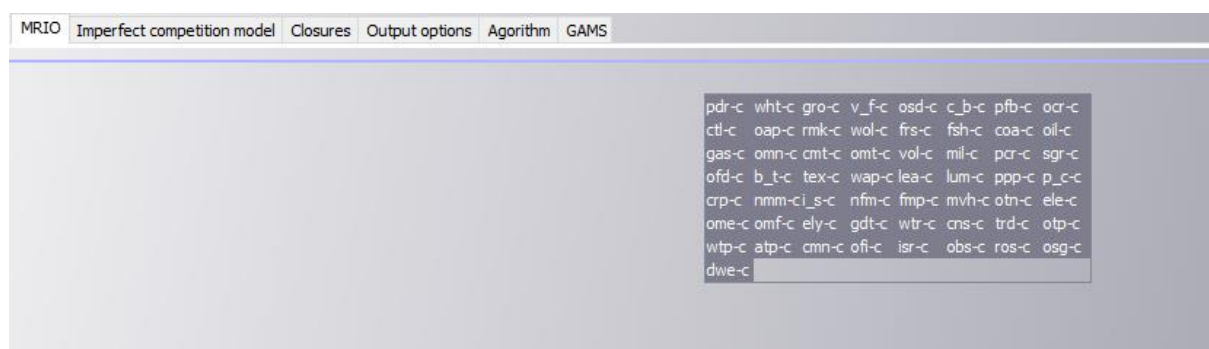
- For the single region layout, two closures for the current account balances are offered: either fixed foreign saving with a flexible exchange rate or the reverse combination.



**Note:** The pre-defined configurations also define specific closures.

The user can also provide its own closure file to introduce alternative and additional closures.

Currently, the MRIO extension lets the user solely depict the commodities for which bi-lateral import demand is dis-aggregated to total intermediate demand and each final demand agent:



The list of products shown adjusts dynamically with the data base chosen. Hence, check it when switching the data base.

Note that the MRIO extensions requires that MRIO split factors were generated during data base generation:

Generate MRIO split factors ☒

## 5.8 Imperfect competition model

Activating the Melitz module will generate a new tab with the following input possibilities:

Under Melitz commodities, the commodities / sectors can be chosen the parameter of the Pareto-distribution of the productivities defined.

The model allows for three different base configurations:

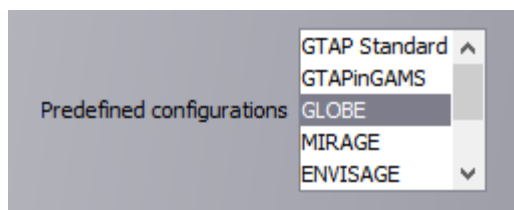
1. “Only aggregated demand, not heterogenous firms” changes the demand structure of the standard GTAP such that only one Armington nest shared by all agents is present. The substitution elasticity is defined as in the case when the full Melitz model is used. The remaining equations of the Melitz model are absent and replaced by the equations of the standard GTAP model. That allows dis-entangling impacts of the full Melitz model from the structural changes on the demand side.
2. “Krugman”: Monopolistic competition at industry level, the number of firms in each sector and region is endogenous and defines the number of varieties.
3. “Melitz”: Melitz model with fix costs and number of firms operating differentiated by trade link, additionally to the industry detail from Krugmann.

The checkbox “Fix cost nest” introduces a differentiation between variable and fix costs input composites where the latter only comprises primary factors, at least as long as there is some minimal primary factor cost share left in the variable input composite. The spinner termed “Max cost share of HET domestic interm. demand” allows switching off love of variety for domestic intermediate demand by same industry depending on the costs share. Only sector-regional combinations with a share above the threshold will be excluded. Setting the threshold to unity will hence leave the love of variety effect switch on for all domestic intermediate demand, while zero will switch it off on all cases.

Note that with the MRIO extension switch one, bi-lateral demand by agent is not based on equal shares, but reflects the shares used in the MRIO-extension.

## 5.9 Pre-configurations

In order to replicate as close as possible some existing well-known global CGEs, pre-configurations are defined which use the modular structure of CGEBox to mimic these model.



### 5.9.1 Standard GTAP model

The details of the implementation of standard GTAP model in GAMS are not discussed in here, see Van der Mensbrugghe 2015. It is however worthwhile to note that using the pre-configuration does not substitute prices out such that the full equation system can be seen. This option will however slow down solution for large models.

### 5.9.2 GTAPinGAMS

The GTAPinGAMS by Tom Rutherford is quite similar to the standard GTAP model with however the following differences which are mostly simplification which render that model ideal as a didactic model:

- There is no global bank mechanism, foreign savings are fixed
- Regional savings and investment demand are fixed as well

### 5.9.3 GLOBE

The GLOBE pre-configuration uses the following attributes. The page numbers below relate to the following GLOBE documentation: McDonald S., Thierfelder, K. (2014): Globe v2: A SAM Based Global CGE Model using GTAP Data.

- The agents share both Armington nests (see page 42), i.e. the Armington aggregation module is switched on
- The regional household is switched off by using a dummy implementation of myGTAP with one household. No remittances etc. are considered (see page 52).
- The government is closed by savings which implies that real consumption is fixed (see page 58). The distribution of government demand to commodities is based on Leontief coefficients.
- Household demand uses a fixed saving rate and adjusts its total spent on consumption to exhaust household income.
- Foreign savings are fixed
- The capital account balance for each country is closed based on flexible exchange rates
- A LES demand system is used
- The VA nest comprises a sub-nest for different labor qualities
- VA and the intermediate composite can be substituted
- There is a CET matching the Armington

The following specifics of GLOBE can currently not be captured by CGEBOX:

- The GLOBE region which differentiates bi-lateral margins trades and thus allows to define bi-lateral trade balances
- The rather flexible tax rate adjustment closures
- The Armington prices are gross of taxes, i.e. consumption taxes are equalized across agents (see page 53 for the definition in GLOBE).



The reader should note that alternative closures are possible, but then, the pre-configuration cannot be used and the different options must be set manually.

#### 5.9.4 MIRAGE

The MIRAGE configuration is defined as follows:

- The GTAP-AGR module is switched on
- There is sub-nest under the VA nest substituting between labor and capital
- All industry sectors are depicted a la Krugmann
- The agents share the Armington specification in the Krugmann model (which endogenous preference shifters depicting love for varieties)
- A LES demand system is used
- There is separate government account closed by government savings
- A specific parameter set is used

#### 5.9.5 ENVISAGE

- The GTAP-AGR and GTAP-E modules are switched on
- A LES demand system is used
- Capital vintages are differentiated in comparative static mode, considering 5 years of depreciation
- A separate government account is closed under fixed savings adjusting direct taxes
- A specific parameter files is used which e.g. introduced a CET on the supply side

#### 5.9.6 CGEBOX

- The GTAP-AGR, GTAP-AEZ and GTAP-E modules are switched on
- Capital vintages are differentiated in comparative static mode, considering 5 years of depreciation

#### 5.9.7 CGEBOX+

As above, but additional, the Melitz module is used for industry sectors

For further details on these pre-defined configurations, consult directly the GAMS code

### 5.10 Options for solving the model: Algorithm

The algorithm panel is available in two layouts. The default one is for normal user levels:

Option	Value
Maximal seconds for one solve	240
Number of repeated solves	3.0
Stop after benchmark solve	<input type="checkbox"/>
Zero iterations shock	<input type="checkbox"/>
Presolve steps with fixed import prices	1
Threshold for SAM imbalance tests, 1.E-	6

Please note that the default maximal time for model solution of 600 seconds might be too low for very large models. Allowing to solve the model multiple (number of repeated solve) is useful for the following reasons. The first solve is done as a CNS (Constrained System of Equations) which is usually faster than solving the problem as a NLP (Non-Linear Programming) problem. If the CNS is not solved to full feasibility, a second attempt as a NLP is made. This can help, for instance, in cases where the model comprises some highly non-linear equations. Equally, with each repeated solve, the feasibility tolerances are slightly relaxex.

If “Stop after benchmark” solve is switched on, solely the benchmark test solve is performed. This solve is always done with zero iterations to check if all model equations are feasible after the benchmarking process. If this option is chosen, the listing for the benchmark model is allows switched on. Similarly, the model after introducing the shock can be solved with zero iterations. This can be helpful to check if the shock is correctly set-up by inspecting which equations are infeasible. Note that the pre-solves are preformed with this option. It hence can also be used to check how successful the pre-solves are in reducing infeasibilities. For information on the pre-solves, please refer to the next section. The thresholds for SAM balance tests can be increased for badly defined input data set.

The steering panel is available in a second version for developers and debuggers which gives additional control on the solution and benchmarking process:

The model can be solved either as a constrained system of equations (CNS, formally solved with a dummy objective to yield a NLP which might help CONOPT or CONPT4) or as a MCP. The latter option is interesting if tax rates are to be endogenized e.g. under emissions ceilings, trade or production quotas, features not comprised in the standard model.

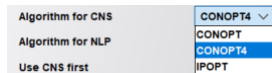
The settings shown above are recommended defaults.

Solving the model as a constrained system of equations CNS

- especially in connection with pre-solves, see below – has proven to be generally faster for larger models (> 250.000 transactions in SAM). The CNS solution will also automatically check if the model is square. MCP gives additionally flexibility such as introducing Tariff Rate or production quotas. Equally, the MCP version will automatically check for unpaired equations and variables. If during development of shock files, GAMS throws the error of a non-square model, switching to MCP might help to find unpaired, but not free variables. It is possible to first ty to solve the model as CNS, and if that fails, switch to MCP.

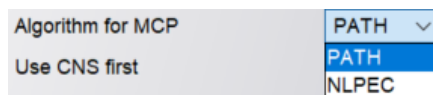
The standard option file is 1, which uses a lower number of minor iteration, shown to speed up solution with larger models. The option file 2 with relaxed convergence tolerance is automatically chosen if the model cannot be solved in a first attempt. Option file 3 relaxes the convergence tolerance further, and is not intended for production runs, but for debugging purpose. Both with MCP and CNS, the above mentioned combinations of solver settings have proven to work best. The user can also

choose which solver to use for NLP and CNS. CONOPT4 which parallelizes e.g. matrix factorization has shown considerable speedups for very large models and is now the recommended NLP solver. For older GAMS version, CONOPT4 was not available or only as a beta version, the user can turn back to



CONOPT3: IPOPT comes for free with a GAMS license and will generally not be able to solve larger models.

If one of the MCP options is chosen, the MCP algorithm can be chosen:



NLPEC is not a solver, but a model re-formulation tool. It will convert the MCP as defined from the pairing to a NLP which is then delegated for solution to CONOPT4. This has proven to solve also very large MCP problems. It allows users without a license for PATH as dedicated MCP solver to solve GE problem in CGEBox comprising complementary condition, such as Tariff-Rate Quotas.

If the model is solved as a NLP, it will on demand first solved as a CNS. CONOPT uses a somewhat simpler and faster algorithm with CNS which might however sometimes fail. It is generally recommended to try “Use CNS first”.

Unexperienced users are generally discouraged to switch to developer level and change the settings e.g. under “Options”, especially relating to variable substitutions and scaling.

### 5.10.1 First-order Taylor approximation to define starting points

In order to speed up processing in recursive-dynamic runs or when shock slicing is used, the code uses a first-order Taylor approximation to estimate a starting point for the next solve (see *gams/solve/forecast.gms*). Starting values for variables which drive the variable initialization (see next section) are defined from the previous solve plus the absolute changes from between the two proceeding solves. Tests have shown that using absolute changes is more stable compared to relative ones, it also accommodates cases such as the BOT which can switch their sign. Tests to use also a second-order approximation were not successful, they provoked infeasibilities in pre-solves in some cases by producing implausible starting points for specific regions and markets.

The GAMS code first defines macros for variables with different number of indices, as seen below.

```
$iftheni.def "%1"=="def"
*
* --- proportional : last period's value update with relative change from t-1 and t-2
*
$macro m_extrM1(var,t,transFormL,maxR) $ (&var(t-2) $ (var.range(t) ne 0)) \
= min( max(&var(t-1)*(1+maxR), &var(t-1)*(1-maxR)), max(min(&var(t-1)*(1+maxR), &var(t-1)*(1-maxR)), \
var.lo(t)*2, &var(t-1) * (1 + ((&var(t-1)/&var(t-2)) -1)*transFormL));

$macro m_extrM2(var,d1,t,transFormL,maxR) $ (&var(d1,t-2) $ (var.range(d1,t) ne 0)) \
= min( max(&var(d1,t-1)*(1+maxR), &var(d1,t-1)*(1-maxR)), max(min(&var(d1,t-1)*(1+maxR), &var(d1,t-1)*(1-maxR)), \
var.lo(d1,t)*2, &var(d1,t-1) * (1 + ((&var(d1,t-1)/&var(d1,t-2)) -1)*transFormL));
```

Afterwards, the variables to project are updated using these macros. As seen below, the first argument of the macro is the variable name, followed by the driving sets (indices). The RC argument defines the share of first order approximation to apply. It is set to 0.5 for the second solved period and to 1.0 afterwards. The last argument defines a maximal relative change. Generally, price changes are

restricted to  $\pm 50\%$  and quantity changes are (virtually) unrestricted. As seen below, for specific quantity variables such as total imports and exports, more cautious maximal relative changes are used.

```
*
* --- extrapolation from previous rec-dyn solve (or shock slicing) of variables which drive bi-lateral demand in iniVars.gms
* and provide a starting point for the pre-solves / global model solve
*
xet(rsNat,i,tSim) m_extrL3(xet,rsNat,i,tSim,rc,2.)
xmt(rsNat,i,tSim) m_extrL3(xmt,rsNat,i,tSim,rc,2.)
xs(rs,i,tSim) m_extrL3(xs,rs,i,tSim,rc,10000.)
xds(rs,i,tSim) m_extrL3(xds,rs,i,tSim,rc,2.)
ps(rs,i,tSim) m_extrL3(ps,rs,i,tSim,rc,0.50)
px(rs,a,tSim) m_extrL3(px,rs,a,tSim,rc,0.50)
```

### 5.10.2 Variable initialization before solving

In order to provide a starting point to the model where many equations are feasible, the projected variable values combined with exogenous changes from the shock (tech shifters, changes in tax rates, population etc.) are used to recursively define starting values for most variables in the model (see “gams/solve/iniVars.gms”). To give an example: composite prices for technology nests, the value-added nests and the intermediate demand nests are defined from given product and factor prices, from their input demands are defined. As a consequence, a basis which usually small infeasibilities in some equations, only, is passed on to the solver.

### 5.10.3 Presolves

For large models, pre-solves can be used during which single region CGEs for each region in the model are solved independently. That process can be repeated several times to inform the individual country CGEs about changes in others regions. The single region models are only introduced as an intermediate step towards the solution of the full model. They thus differ from regular single country CGEs which are usually solved either at fixed international prices or by rendering import and export prices depending on import and export quantities, not considering different trading partners. The pre-solves in here aim at providing a good starting point for a solve of the full global model. Therefore, their structure needs to reflect the bi-lateral trade relations and other linkages between regions in the final multi-regional model to solve. The layout of the single country models therefore differs somewhat from the usual single country CGE structure and solely uses equations and variables already comprised in the model at unchanged parameterization.

In the standard GTAP model, there is no CET on the supply side. We therefore simply assume that a change in demand for bilateral exports of a single country has a negligible impact on the supply price of each importer. Accordingly, we drive the single country models with fixed fob prices on the import side. On the export side, however, we exploit the Armington structure: we feed changes in the country’s supply prices in the Armington share equations of the importers at fixed total import and prices. The reader should note that solving the model also with fixed prices for exports would not help much in providing a good starting point for the model. Without a CET, the export price is by definition equal to the supply price in the domestic market. If one would hence fix the export prices, the domestic prices in the current solve would also be fixed. There would hence be no updated price information be passed along from solving the single country models to the next round of solves.

Further links in the full models between regions are based on the global factor price index which is fixed during a global model solve to reflect that there is no money solution, i.e. the behavioral equations are of degree zero in prices. However, during single model solves, fob prices are fixed such that it is impossible to fix at the same time the factor price index. In order to still drive the single models during iterations towards a global factor price index of unity, the fob prices are divided before

each solve by the currently calculated global factor price index as are the saving price index and the factor prices of all regions which enter fixed the single model solves.

Finally, we reflect the global bank mechanism by calculating iterations the average expected return at global level, based on weighting with regional net investments. A heuristic estimates a change in foreign savings which should drive the country's expected rate towards the global average.

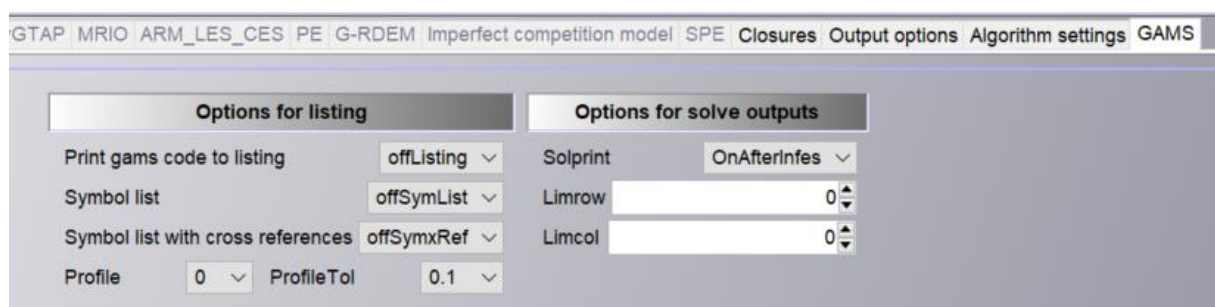
There are a number of options available for the pre-solves:

1. The number of pre-solves. For larger shocks and a higher number of regions, three iterations are recommended as the pre-solves are relatively cheap. In recursive dynamic runs, one round of pre-solves is often giving the fastest results.
2. Use of grid solves. That solves the single country models in parallel. The grid solve is the recommended option. Solving the single countries one after another leads to hard to control changes in prices depending on the order in which the region are defined. Equally, even cheap modern laptop are multi-core machines which benefit from the option to solve models in parallel.
3. Stopping the pre-solve loop if all single models are feasible. Tests have shown that especially in recursive-dynamic run or with shock slicing, the first-order approximation combined with variable initialization provide typically excellent starting points such that the sum of infeasibilities after a first round of solves when combining the solutions into the global model is quite small.

CNS models based with CONOPT have proven to benefit much more from the pre-solves than the MCP version. Tests have shown that the combination of presolves and CONOPT4 allow to solve 57x82 models without any filtering in about 10 minutes even under larger shocks.

Improvements in solvers, processor speed and the number of cores available even in laptops, as well as code improvements in CGEBox mean that the using pre-solves can slow the overall model solve for smaller shocks and medium sized models. For medium sized models (less than 20 regions at full product detail of the GTAP Data Base), usually, pre-solves might not gain much.

## 5.11 GAMS



The screenshot shows the GAMS software interface with the 'GAMS' tab selected. The interface is divided into two main panels: 'Options for listing' and 'Options for solve outputs'.

**Options for listing:**

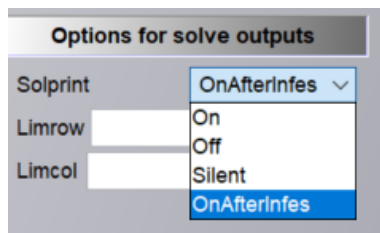
- Print gams code to listing: offListing (dropdown)
- Symbol list: offSymList (dropdown)
- Symbol list with cross references: offSymxRef (dropdown)
- Profile: 0 (dropdown)
- ProfileTol: 0.1 (dropdown)

**Options for solve outputs:**

- Solprint: OnAfterInfes (dropdown)
- Limrow: 0 (spin box)
- Limcol: 0 (spin box)

The left-hand side controls what is outputted to the listing. For details, refer to the GAMS documentation.

The recommend setting for solprint is “OnAfterInfes” as it makes mostly limited sense to inspect the typically quite long model listing if the model solves. For complex simulation, “OnAfterInfes” offers the possibility to generate a listing if the model could not be solved without infeasibilities in the last try in each solved period.

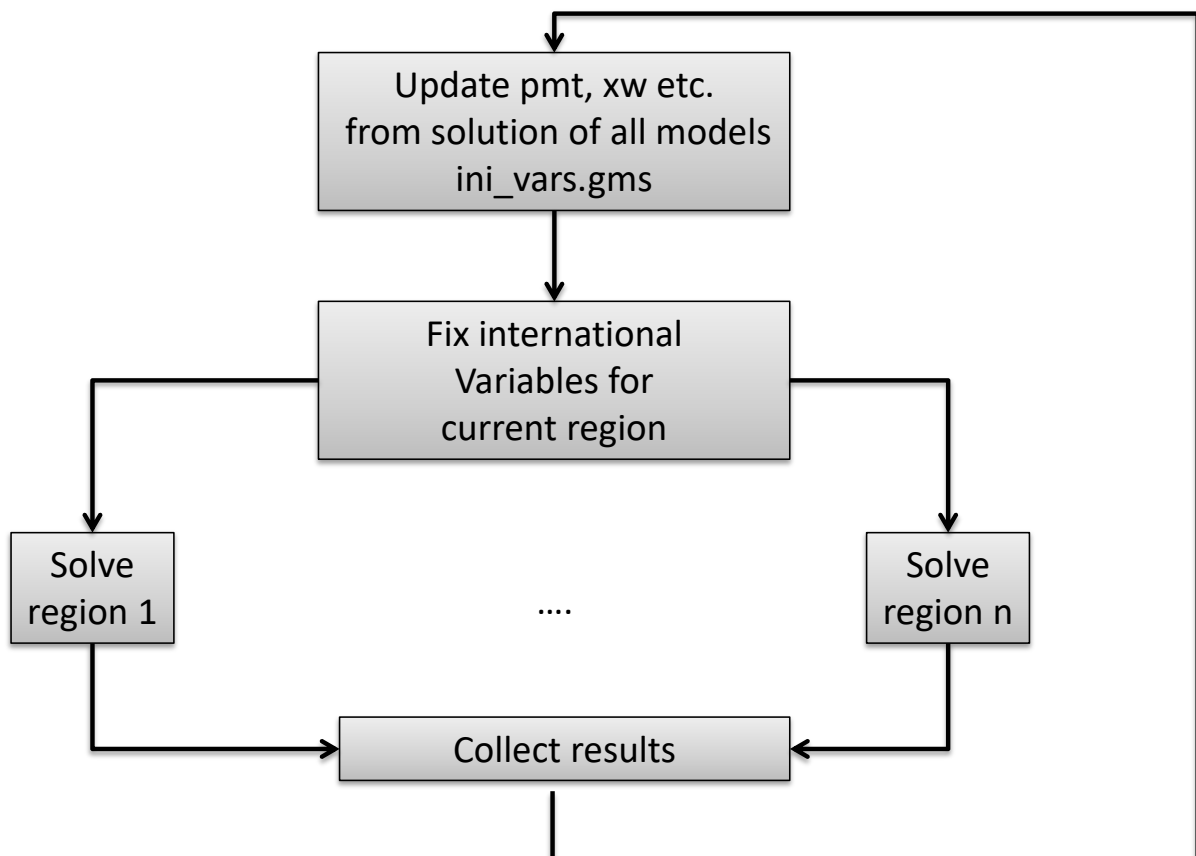


Not that all symbols (variables, equations) can be stored in a GDX container with GDX output switched on which is often more efficient to inspect individually variable.

A daunting exercise provides debugging infeasible model solutions provoked by bounds on variables. Non-zero lower bounds for many variables are introduced by default to prevent math trap errors such as from division by zero or logarithms of zero values during the solution process, which typically will mean that the solver will fail. Finding variables where the bounds might have chosen too tight in a large model listing is almost impossible. Therefore, if the last model solve fails with some larger infeasibilities, embedded python code will automatically generate log output on infeasible equations and variables which are at their bounds to ease the debugging process.

## 5.12 Pre-solves

The pre-solves are introduced to speed up solution of very large models. During the pre-solve iterations, each model region is solved separately as a single country model where only some variables in the second Armington nest of their export destinations or the second level CET nests of their import origins react. The basic process is depicted in graphic below. In between iterations, variables in each regional model which depend on other regions' endogenous variables such as their c.i.f. prices driven by f.o.b. exporter prices are updated. Next, variables not related to the current region but comprised in the single region model are fixed and the model is sent to the grid for solution. While the solve process for this model starts in the background, the next regional model is prepared and sent to grid. The code then waits until all model solutions are returned and collects their solutions to start the next iteration.



To solve the model for one region, only, all model equations therefore carry a dynamic set `rs`, indicating the region(s) in current solve. The presolve code can be found in “gams\solve\presolve.gms”. The number of pre-solves is defined via the interface and drives a loop:

```
loop( presolve,
```

In that loop, several steps are performed:

1. A heuristic determines the foreign savings for that region, later fixed during the solve, using an interpolation over previous solves:



```

p_reIter(rsNat,"savfIter",preSolve) $ p_reIter(rsNat,"distMin",preSolve)
= p_reIter(rsNat,"savfMin",preSolve)
+ [ ( p_reIter(rsNat,"savf",preSolve-1)-p_reIter(rsNat,"savf",preSolve))
/ (p_reIter(rsNat,"rorg",preSolve-1)-p_reIter(rsNat,"rorg",preSolve)) )
* (p_reIter("wor","rorg",preSolve)-p_reIter(rsNat,"rorg",preSolve)) ]
$ (p_reIter(rsNat,"rorg",preSolve-1)-p_reIter(rsNat,"rorg",preSolve));

savf.1(rsNat,tSim) $ p_reIter(rsNat,"savfIter",preSolve) = p_reIter(rsNat,"savfIter",preSolve);

```

2. A loop over the regions is performed and dynamic sets are updated which indicate the nations in the current model to generate rsNat and sub-regions therein rs, including the nation itself. The set rrComb comprises all bi-lateral combinations where the region is either an importer or an exporter.

```

loop( rr $ rNat(rr),

--- include only current region in model

option kill=rs;
option kill=rsNat;
rs(rr)      = YES;
rsNat(rr)   = YES;
rs(subr) $ r_r(subR,rr) = YES;

--- bilateral combinations of trade flows to be included:
    all import and export flows of current regions

option kill=rrComb;
rrComb(rNat,rNat1) $ (rsNat(rNat) or rsNat(rNat1)) = YES;

```

3. The variables for the current region are initialized from the given results so far in “solve\iniVars.gms” and negative variables reset in “solve\resetNegative.gms”. That step is only included in the loop itself if the models are solved sequentially. If the models are solved in parallel on a grid, that step is performed outside of the loop over the regions at the beginning of the iteration.
4. Nest, the trade related variables for all regions are fixed and next, bounds introduced for the region in the current solve, e.g.

```

xmt.fx(rNat,i,tSim) = xmt.l(rNat,i,tSim);

xmt.lo(rsNat,i,tSim) $ xmtFlag(rsNat,i) = 0;
xmt.up(rsNat,i,tSim) $ xmtFlag(rsNat,i) = +inf;

```

5. The model for that region is solved:

```

$$iftheni.mt "%firstModeltype%"=="MCP"

--- MCP solution
solve gtap using mcp;
$$else.mt

--- presolve in CNS mode, switch to NLP for current region if infeasible in last presolve
(NLP might come up with something useful)

if (useCnsFirst and (p_reIter(rr,"numinfes",presolve-1) eq 0),
    solve gtap using cns;
else
    solve gtap using nlp minimizing dummy;
);

$$endif.mt

```

The advantages of the pre-solve process compared to solving the full model directly on the shock depend on the size of the model and its complexity, the nature of the shock, as well as the hardware and solver used. For small to medium sized models under smaller shocks, using the pre-solves will typically drive up the overall computing time, especially if using CONOPT4 which itself parallelizes part of the solution process. There are also cases where the pre-solves might render it harder to solve the full model and can provoke infeasibilities. It is generally worth a try to use the pre-solves if the solution time of the full model exceeds 30 seconds and a multi-processor computer is used.

The rrComb set will ensure that only the bi-lateral flows in which the current region is involved are introduced in the single region model, e.g.:

```
xw(rNat1,i,rNat,t)/xw.scale(rNat1,i,rNat,t) =e= amw(rNat1,i,rNat,t)*xmt(rNat,i,t)/xw.scale(rNat1,i,rNat,t)

* [ { (pmt(rNat,i,t)/[(1 + imptx(rNat1,i,rNat,t) + mtax(rNat,i,t))
*
*   --- Note WB: cif price is defined via macro %pmcif%, normally not a variable in model
*
*   * %pmcif(rNat1,i,rNat,t)/chipm(rNat1,i,rNat) ])**sigmaw(rNat,i)
*   * m_lambdam(rNat1,i,rNat,t)**(sigmaw(rNat,i) - 1)}
*
*   $$iftheni.smallShare "%hasSmallShare%"=="on"
*   $ (not isSmallImpShare(rNat1,i,rNat))
*   + m_lambdam(rNat1,i,rNat,t) $ isSmallImpShare(rNat1,i,rNat)
*   $$endif.smallShare
*
* ];
```

In the above equation and the case of the current region acting as the exporter rNat1, the total price of imports pmt and total imports xmt of the trade partner rNat are fixed, but changes in the current regions f.o.b. price will change the c.i.f. price – along e.g. with shocks to the importers tariffs – and change the simulated export flows xw. Thus, at least as long as the exporter's share in the total imports of rNat is not very large, the major impact of the Armington structure is captured. Furthermore, as pmt and xmt will be updated after each pre-solve, the individually regional model solves will iteratively approximate the solution of the global model.

Similarly, if a CET on the export side is present, for the case where rnat1 acts as an importer, the impact of changing import demand on the f.o.b. price faced is approximated:

```
*
* --- This equation substitutes out the equilibrium condition for bilateral trade, equation (56) in VDM 2018
* (makes it possible to drop equation (68) in VDM 2018)
*
* peeq(rNat,iIn(i),rNat1,ts(t)) $ (m_xwFlag(rNat,i,rNat1) $ psFlag(rNat,i,"pe") $ rrComb(rNat,rNat1)) ..
*
* 0 =e= ( m_xws(rNat,i,rNat1,t)/xw.scale(rNat,i,rNat1,t)
*         - gw(rNat,i,rNat1,t)*xet(rNat,i,t)/xw.scale(rNat,i,rNat1,t)
*         + { 1
*             + { (pe(rNat,i,rNat1,t)/pet(rNat,i,t))**omegaw(rNat,i) } $ isSmallExpShare(rNat,i,rNat1)
*             ) $ (not isSmallExpShare(rNat,i,rNat1)) ]
*         ) $ (omegaw(rNat,i) ne inf)
```

The foreign savings are fixed during the pre-solves such at the related equation is not introduced:

```
*
* --- Determination of foreign savings, equation (80) in VDM 2019
*
* savfeq(rsNat(rNat),ts(t)) $ ( ( (RoRFlag eq equalReturnToInv)
*                               or ((RoRFlag eq fixedAllocationOfInv) $ (not rres(rNat))))
*                               $ (savf.range(rNat,t) ne 0))..
*
* 0 =e=
```

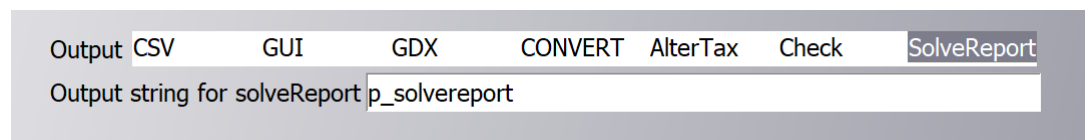
The same holds for the capital account equation which in the full model determines the foreign savings for the residual region, see the first dollar condition which requires that at least two regions with unfixed foreign savings need to be present in the model:

```
*
capAccteq(ts(t)) $ ( ( (sum(rsNat $ (savf.range(rsNat,t) ne 0),1) gt 1)
```

The parameter `p_solveReport` reports various statistics such as the time needed or potential infeasibilities:

```
p_solveReport(tSim,"Presolves, maxSumInfes") = sMax( (preSolves,rNat), p_roreIter(rnat,"sumInfes",preSolves));
p_solveReport(tSim,"Presolves, infes, step") $ p_solveReport(tSim,"Presolves, maxSumInfes")
= sMax( (preSolves,rNat) $ p_roreIter(rnat,"sumInfes",preSolves),preSolves.pos);
p_solveReport(tSim,"Presolves, maxNumInfes") = sMax( (preSolves,rNat), p_roreIter(rnat,"numInfes",preSolves));
p_solveReport(tSim,"Presolves, seconds") = timeElapsed - p_solveReport(tSim,"Presolves, seconds");
```

That allows also tracking the solution behaviour in recursive-dynamic solves:



It is stored after each year in the scratch directory. The GDX viewer from the GAMSIDE allows keeping a file open for read while other processes update the file in the background. The user can hence open that file in the viewer and check there the solution behaviour while the simulation continues:

Entry	Symbol	Type	Dim	Nr Elem
1	p_solveReport	Par	2	29

p_solveReport(*, *)				
	t00	t01	t02	
timeElapsed	21.959000	151.863000	185.751000	
prep,seconds	0.163000	98.649000	1.188000	
Presolves, seconds		16.959000	17.639000	
full Solve, seconds	8.610000	12.319000	12.644000	
Full solve, sumVar	133090.000000	134275.000000	134275.000000	
Full solve, modelStat	1.000000	16.000000	16.000000	
Full solve, solveStat	11.000000	1.000000	1.000000	
Full solve, nTries	1.000000	1.000000	1.000000	
Full solve, resUsd	0.359000	4.843000	5.125000	
seconds	21.219000	128.586000	32.165000	

The use of the pre-solves is switched off by settings the number of pre-solves to zero on the tabe “Algorithm settings”:

If a non-zero value is chosen, additional controls for the pre-solve process become available if the user level is set to developer:

Switching the grid solve off makes sense only to debug the process. Switching off the CNS in the first solve already might help with CONOPT4 in cases of (1) highly non-linear relations or (2) if CONOPT4 bounces against a bound of a variable which stops the search for a smaller sum of infeasibilities in CNS mode. If the single region models are solved as a NLP with a dummy objective instead, CONOPT4 might find a solution or at least a solution with a small sum of infeasibilities, but will require more time in the standard case of a feasible solve. The pre-solve script will automatically solve the next iteration in NLP mode with the previous one was infeasible.

If solution listings are to be checked for the pre-solves, the output can be activated. Finally, it is possible to leave the loop of pre-solves early if all single model region were feasible. This is especially useful if the solver might not find a feasible solution in the early pre-solves in recursive-dynamics runs. Allowing, for instance, for up to three presolve rounds and leaving the minimum pre-solves at one will in many cases mean that just one round of pre-solves is preformed.

## 6. Post-model processing

### 6.1 Decomposing welfare changes in CGEBox

- Wolfgang Britz, December 2018 -

#### 6.1.1 Background

When analyzing impacts of policy changes or other shocks, the simultaneous adjustments in various elements impacting welfare such as in factor and commodity prices or in tax income or foreign savings render it challenging to understand what drives the simulated welfare change. CGEBox offers two approaches to decompose welfare changes which stem from different “schools”. In the world of partial equilibrium modelers, income is fixed, but products are often close substitutes such that cross-price effects in demand matter. Therefore, that community often decomposes the welfare change with regard to individual product price changes. The first approach detailed next draws on that concept and adds also the perspective of changes in individual factor prices and quantities.

The general equilibrium community has another focus and typically decomposes welfare with regard to terms-of-trade effects and allocative effects from removing tax distortions. That approach is, for instance, implemented in the GEMPACK version of the GTAP Standard model and provided in CGEBox as well. Both approaches have some overlap, but clearly complement each other and are useful to understand where welfare changes result from.

#### 6.1.2 Decomposition with regard to overall price indices and income components

With regard to overall price indices, we measure the impact from updating the price indices from the benchmark to the simulation points, e.g. for savings:

```
u.1(rsNat,%2) = au.1(rsNat,"%t0%") * uh.1(rsNat,"hhsld","%t0%")**betap.1(rsNat,"%t0%")
               * ug.1(rsNat,"%t0%")**betag.1(rsNat,"%t0%")
               * [us.1(rsNat,"%t0%")*pSave.1(rsNat,"%t0%")/pSave.1(rsNat,%2)]**betas.1(rsNat,"%t0%");
```

Next, we use an EV calculation to express that utility changes in terms of required utility to reach the new income level at base year prices (in file postModel\calcExp.gms).

```
p_results(rsNat,"V",%4,%3,"tot",%1,"%version%")
$ (p_results(rsNat,"V","ra","u","tot",%1,"%version%") gt eps)
= (u.1(rsNat,%2)-u.1(rsNat,"%t0%")) * p_results(rsNat,"V","au","mm","tot",%1,"%version%") *
  ( p_pConsHlp(rsNat,%2)/betap.1(rsNat,"%t0%")**betap.1(rsNat,"%t0%")
  * [ { ( pg.1(rsNat,"%t0%")/betag.1(rsNat,"%t0%"))**betag.1(rsNat,"%t0%")
        }$ axg(rsNat,"%t0%") + 1 $ (not axg(rsNat,"%t0%")) ]
  * [ { ( pi.1(rsNat,"%t0%")/betas.1(rsNat,"%t0%"))**betas.1(rsNat,"%t0%") }
        $ (p_results(rsNat,"V","ra","betas","tot",%1,"%version%") ge 0)
        + 1 $ (p_results(rsNat,"V","ra","betas","tot",%1,"%version%") lt 0) ] );
```

The argument %4 is the code for the change measured, for the saving price index it is “ps”:

```
$batinclude 'postModel/calcExp.gms' '%2' '%2' 'mm' 'ps'
```

Similar statements are found for the private household and government price indices.

To measure impacts of changes in factor income, depreciation and indirect taxes, we only update these components from benchmark to shock, e.g. for factor income excluding changes in depreciation:

```

*
* --- update only factor income
*
      u.l(rsNat,%2) = u.l(rsNat,"%t0%") * (facty.l(rsNat,%2)
      + fdepr(rsNat,%2)*pi(rsNat,%2)*kstock(rsNat,%2)
      - fdepr(rsNat,"%t0%")*pi(rsNat,"%t0%")*kstock(rsNat,"%t0%")
      + yTaxInd.l(rsNat,"%t0%"))/regy.l(rsNat,"%t0%");
$batinclude 'postModel/calcExp.gms' '%2' '%2' 'mm' 'va'

```

For depreciation:

```

*
* --- update only depreciation
*
      u.l(rsNat,%2) = u.l(rsNat,"%t0%") * (facty.l(rsNat,"%t0%")
      - fdepr(rsNat,%2)*pi(rsNat,%2)*kstock(rsNat,%2)
      + fdepr(rsNat,"%t0%")*pi(rsNat,"%t0%")*kstock(rsNat,"%t0%")
      + yTaxInd.l(rsNat,"%t0%"))/regy.l(rsNat,"%t0%");
$batinclude 'postModel/calcExp.gms' '%2' '%2' 'mm' 'dep'

```

And for indirect taxes:

```

$batinclude 'postModel/calcExp.gms' '%2' '%2' 'mm' 'dep'
*
* --- update only indirect taxes
*
      u.l(rsNat,%2) = u.l(rsNat,"%t0%") * (facty.l(rsNat,"%t0%") + yTaxInd.l(rsNat,%2))/regy.l(rsNat,"%t0%");
$batinclude 'postModel/calcExp.gms' '%2' '%2' 'mm' 'xt'

```

These three components above are also found in the EV decomposition by Huff and Hertel 2000.

The total impact of regional income changes is defined as:

```

      u.l(rsNat,%2) = u.l(rsNat,"%t0%") * regy.l(rsNat,%2)/regy.l(rsNat,"%t0%");
$batinclude 'postModel/calcExp.gms' '%2' '%2' 'mm' 'x'

```

For the individual factors, we loop over the factors and update their price and quantities from benchmark and shock

```

| loop(f0,
*
*      --- update quantities and prices for current factor f0
*
      pft.l(rsNat,f0,%2) = pft.scale(rsNat,f0,%2);
      xft.l(rsNat,f0,%2) = xft.scale(rsNat,f0,%2);
      pf.l(rsNat,f0,a,%2) $ xftFlag(rsNat,f0,a) = pf.scale(rsNat,f0,a,%2);
      xf.l(rsNat,f0,a,%2) $ xftFlag(rsNat,f0,a) = xf.scale(rsNat,f0,a,%2);

```

From there, we recalculate the factor income:

```

factY.l(rsNat,%2) = [ sum(fm $ xftFlag(rsNat,fm), pft(rsNat,fm,%2)*xft(rsNat,fm,%2))
+ sum((f0,a) $ xftFlag(rsNat,f0,a), pf(rsNat,f0,a,%2)*xf(rsNat,f0,a,%2))
- fdepr(rsNat,"%t0%")*pi(rsNat,"%t0%")*kstock(rsNat,"%t0%")
];

```

And calculate as discussed for other income contributions, the change in utility and the related EV:

```

--- update u and calculate money metric (homothetic preferences ...)
u.l(rsNat,%2) = u.l(rsNat,"%t0%") * (facty.l(rsNat,%2) + yTaxInd.l(rsNat,"%t0%"))/regy.l(rsNat,"%t0%");
$batinclude 'postModel/calcExp.gms' '%2' '%2' 'mm' f0

```



Afterwards, we reset the prices and quantities

```
pft.1(rsNat,f0,%2) = pft.1(rsNat,f0,"%t0%");
xft.1(rsNat,f0,%2) = xft.1(rsNat,f0,"%t0%");
pf.1(rsNat,f0,a,%2) $ xFlag(rsNat,f0,a) = pf.1(rsNat,f0,a,"%t0%");
xf.1(rsNat,f0,a,%2) $ xFlag(rsNat,f0,a) = xf.1(rsNat,f0,a,"%t0%");
);
```

### 6.1.3 Decomposition with a focus on prices in changes

In here, we calculate the approximate impact from individual price changes for commodities and factors. To do so for the product prices, the following steps are performed:

1. We update each Armington price sequentially from benchmark to simulation level.
2. Next, we update the price indices (respectively utility in case of the LES demand system) for the private household, government and investments which enter the top-level utility function of the regional household.
3. Based on these updated price indices, we recalculate the change in utility of the regional household.
4. And calculate the EV equivalent to the simulated change in utility from updating only that single price.

### 6.1.4 Demand nests

The CES demand nest prices needs to be updated first to capture their impact on the different demand systems. That requires a recursion, starting with the lowest to the highest level to capture the case of nests nested in other nests:

```
nestsToDo(dNest) = YES;
while( card(nestsToDo),
option k11=curNest;
curNest(nestsToDo) $ (not sum(dNest_n_fd(nestsToDo,nestsToDo1,fdn),1)) = YES;
pdNest(rNat,dNest,fdn,%2)
= [ prod(dNest_i_fd(dNest,i,fdn) $ alphaa.1(rNat,i,fdn,%2),
(m_pa(rNat,i,fdn,%2)/(alphaa(rNat,i,fdn,%2)*(1+(lambdai(rNat,i,%2)-1)$ sameas(fdn,"inv"))))*alphaa(rNat,i,fdn,%2))
* prod(dNest_n_fd(dNest,dNest1,fdn) $ alphaDN(rNat,dNest1,fdn,%2),
(pdNest(rNat,dNest1,fdn,%2)/alphaDN(rNat,dNest1,fdn,%2))*alphaDN(rNat,dNest1,fdn,%2))
* axdNestCD(rNat,dNest,fdn,%2)
] $ (sigmaFDNest(rNat,dNest,fdn) eq 1)
+ [ { sum(dNest_i_fd(dNest,i,fdn) $ alphaa.1(rNat,i,fdn,%2),
alphaa(rNat,i,fdn,%2) * (m_pa(rNat,i,fdn,%2)/(1+(lambdai(rNat,i,%2)-1)$ sameas(fdn,"inv"))))*alphaa(rNat,i,fdn,%2))
+ sum(dNest_n_fd(dNest,dNest1,fdn) $ alphaDN(rNat,dNest1,fdn,%2),
alphaDN(rNat,dNest1,fdn,%2) * pdNest(rNat,dNest1,fdn,%2))*alphaDN(rNat,dNest1,fdn,%2))
} *(1/(1-sigmaFDNest(rNat,dNest,fdn)))
] $ (sigmaFDNest(rNat,dNest,fdn) ne 1);
nestsToDo(curNest) = no;
);
```

The calculation in the loop requires different dual price aggregator for the CD case where the substitution elasticity is unity – first block – or not, as captured in the second block. Once the demand nest prices are updated, the price indices respectively utility for the three elements of the regional household – private, government and investment consumption can be calculated.

### 6.1.5 Private household

For the private household, the approach is different for the LES and the CDE demand.

1. LES

The LES demand has an indirect utility function which can be calculated from income and prices as shown below



```

uh.1(rsNat,"hhsld",%2)
= (yc(rsNat,"hhsld",%2)
  - sum(i $ (xaFlag(rsNat,i,"hhsld") $ alphaLES.1(rsNat,i,"hhsld",%2)),
    gammaLES(rsNat,i,"hhsld",%2)*pop(rsNat,"hhsld",%2) * pa.1(rsNat,i,"hhsld",%2))
  - sum(dNest_n_fd("top",dNest,"hhsld"),
    gammaLES(rsNat,dNest,"hhsld",%2)*pop(rsNat,"hhsld",%2) * pdNest(rsNat,dNest,"hhsld",%2))
  )*( prod(i $ (xaFlag(rsNat,i,"hhsld") $ alphaLES.1(rsNat,i,"hhsld",%2)),
    (alphaLES.1(rsNat,i,"hhsld",%2)/pa.1(rsNat,i,"hhsld",%2))**alphaLES.1(rsNat,i,"hhsld",%2))
    * prod(dNest_n_fd("top",dNest,"hhsld") $ alphaLES.1(rsNat,dNest,"hhsld",%2),
    (alphaLES(rsNat,dNest,"hhsld",%2)/pdNest(rsNat,dNest,"hhsld",%2))**alphaLES(rsNat,dNest,"hhsld",%2)))
  * 1/uhLES(rsNat,"hhsld",%2)
/ p_welfTemp(rsNat,"", "uh.1",%2)
* uh.1(rsNat,"hhsld", "%t0%");

```

The calculation faces the challenge that the even the relative impact of the price depends on the income level of the private household due to non-homothetic demands. If one uses as shown above in the first block the simulate income at the benchmark (%2 index) and updates all prices, the utility of the private household after all price change and all adjustments in income flows would be simulated, i.e. the full utility change and not only the one attributed to price changes. We therefore correct for the total change in utility from benchmark to the simulation outcome with the last term, even if that is somewhat ad-hoc.

The last statement calculates the changes in the price index for the private household necessary to reach that utility.

## 2. CDE

The expenditure function for the CDE case is only implicitly defined. We therefore refrain from fully consistent approach. Instead, we update the Z parameters based on the price change:

```

zCons.1(rsNat,i,"hhsld",%2) $ pa.1(rsNat,i,"hhsld", "%t0%")
= zCons.1(rsNat,i,"hhsld", "%t0%") * (pa.1(rsNat,i,"hhsld",%2)**bh(rsNat,i,%2))
/ (pa.1(rsNat,i,"hhsld", "%t0%")**bh(rsNat,i,%2));

zConsDN.1(rsNat,dNest,"hhsld",%2) $ pdNest.1(rsNat,dNest,"hhsld", "%t0%")
= zConsDN.1(rsNat,dNest,"hhsld", "%t0%") * (pdNest.1(rsNat,dNest,"hhsld",%2)**bhdN(rsNat,dNest,%2))
/ (pdNest.1(rsNat,dNest,"hhsld", "%t0%")**bhdN(rsNat,dNest,%2));

```

These parameters must add up to unity, see the following equation comprised in the GAMS code of the CDE demand system:

```

uheq(rs(r),h,ts(t)) $ sum(i,xaFlag(r,i,h)) ..
1 == sum(i $ (xaFlag(r,i,h) $ (not sum(dNest_i_fd(dNest,i,h),1))), zcons(r,i,h,t)/bh(r,i,t))
*
* --- add sub-nests under final demand function, not part of standard GTAP model
*
+ sum(dNest_n_fd("top",dNest,h) $ alphaDN(r,dNest,h,t), zconsDN(r,dNest,h,t)/bhdN(r,dNest,t));

```

We therefore calculate and apply a scaling factor:

```

zConsSum(r,"hhsld",%2)
= sum(i $ (xaFlag(r,i,"hhsld") $ (not sum(dNest_i_fd(dNest,i,"hhsld"),1))), zcons(r,i,"hhsld",%2)/bh(r,i,%2))
+ sum(dNest_n_fd("top",dNest,"hhsld") $ alphaDN(r,dNest,"hhsld",%2), zconsDN(r,dNest,"hhsld",%2)/bhdN(r,dNest,%2));

zCons.1(rsNat,i,"hhsld",%2) = zCons.1(rsNat,i,"hhsld",%2)/zConsSum(rsNat,"hhsld",%2);
zConsDN.1(rsNat,dNest,"hhsld",%2) = zConsDN.1(rsNat,dNest,"hhsld",%2)/zConsSum(rsNat,"hhsld",%2);

```

From there, the budget shares are defined:

```

zConsSum(rsNat,"hhsld",%2)
= sum(i $ (xaFlag(rsNat,i,"hhsld") $ (not sum(dNest_i_fd(dNest,i,"hhsld"),1))), zcons(rsNat,i,"hhsld",%2))
+ sum(dNest_n_fd("top",dNest,"hhsld") $ alphaDN(rsNat,dNest,"hhsld",%2), zconsDN(rsNat,dNest,"hhsld",%2));

xcshr(rsNat,i,"hhsld",%2) = zcons(rsNat,i,"hhsld",%2)/zConsSum(rsNat,"hhsld",%2);

```

Which allows the calculation of the changes in the price index:

```
p_welfTemp(rsNat,j,"pcons",%2)
= sum(i $ xaFlag(rsNat,i,"hhs1d"), xcshr(rsNat,i,"hhs1d",%2)*pa.1(rsNat,i,"hhs1d",%2)) ;
```

### 6.1.6 Government and investments

For the government, the consistent price CD or CES index is used:

```
p_welfTemp(rsNat,iIn(j),"pg",%2) $ axg(rsNat,%2)
= (sum( gov,prod( i $ ((not sum(dNest_i_fd(dNest,i,gov),1)) $ alphaa.1(rsNat,i,gov,%2)),
(p_welfTemp(rsNat,j,i,gov)/alphaa(rsNat,i,gov,%2))**alphaa(rsNat,i,gov,%2))
* prod(dNest_n_fd("top",dNest,gov) $ alphaDN(rsNat,dNest,gov,%2),
(p_welfTemp(rsNat,j,dNest,gov)/alphaDN(rsNat,dNest,gov,%2))**alphaDN(rsNat,dNest,gov,%2))
)/axg(rsNat,%2) ) $ (sigmag(rsNat) eq 1)
+ (sum(gov,
sum(i $ ((not sum(dNest_i_fd(dNest,i,gov),1)) $ xaFlag(rsNat,i,gov)),
alphaa(rsNat,i,gov,%2)*p_welfTemp(rsNat,i,gov,%2)**(1-sigmat(rsNat)))
+ sum(dNest_n_fd("top",dNest,gov) $ alphaDN(rsNat,dNest,gov,%2),
alphaDN(rsNat,dNest,gov,%2)*p_welfTemp(rsNat,j,dNest,gov)**(1-sigmat(rsNat)))
)**(1/(1-sigmat(rsNat)))/axg(rsNat,%2) ) $ (sigmag(rsNat) ne 1);
```

The same holds for investments:

```
p_welfTemp(rsNat,iIn(j),"pSave",%2) $ axg(rsNat,%2)
= (sum( inv,prod( i $ ((not sum(dNest_i_fd(dNest,i,inv),1)) $ alphaa.1(rsNat,i,inv,%2)),
(p_welfTemp(rsNat,j,i,inv)/alphaa(rsNat,i,inv,%2))**alphaa(rsNat,i,inv,%2))
* prod(dNest_n_fd("top",dNest,inv) $ alphaDN(rsNat,dNest,inv,%2),
(p_welfTemp(rsNat,j,dNest,inv)/alphaDN(rsNat,dNest,inv,%2))**alphaDN(rsNat,dNest,inv,%2))
)/axg(rsNat,%2) ) $ (sigmag(rsNat) eq 1)
+ (sum(inv,
sum(i $ ((not sum(dNest_i_fd(dNest,i,inv),1)) $ xaFlag(rsNat,i,inv)),
alphaa(rsNat,i,inv,%2)*p_welfTemp(rsNat,i,inv,%2)**(1-sigmat(rsNat)))
+ sum(dNest_n_fd("top",dNest,inv) $ alphaDN(rsNat,dNest,inv,%2),
alphaDN(rsNat,dNest,inv,%2)*p_welfTemp(rsNat,j,dNest,inv)**(1-sigmat(rsNat)))
)**(1/(1-sigmat(rsNat)))/axg(rsNat,%2) ) $ (sigmag(rsNat) ne 1);
```

### 6.1.7 Utility change for the regional household and EV

Next, we update the utility of the regional household under its CD-utility function. Note that we are not updating the shares to reflect that the elasticity of private expenditure to an utility change might be affected by the price change – which reflects the simplified treatment of private demand:

```
*
* --- calculate new utility based on updated prices
*
p_welfTemp(rsNat,iIn(j),"u.1",%2)
= au.1(rsNat,"%t0%")
* [uh.1(rsNat,"hhs1d", "%t0%")*p_pconsH1p(rsNat,%2)/p_welfTemp(rsNat,j,"pcons",%2)]**betap.1(rsNat,%2)
* [ug.1(rsNat,"%t0%")*pg.1(rsNat,"%t0%")/p_welfTemp(rsNat,j,"pg",%2)]**betag.1(rsNat,%2)
* [us.1(rsNat,"%t0%")*pSave.1(rsNat,"%t0%")/p_welfTemp(rsNat,j,"pSave",%2)]**betas.1(rsNat,%2);
```

From there, the equivalent variation is defined:

```
*
* --- EV
*
p_results(rsNat,"v","mm",iIn(j),"tot",%1,"%version%")
$ (p_results(rsNat,"v","ra","u", "%t0%", "%version%") gt eps)
= ( p_welfTemp(rsNat,j,"u.1",%2)-u.1(rsNat,"%t0%") ) * p_results(rsNat,"v","au","mm","tot",%1,"%version%") *
( p_pconsH1p(rsNat,%2)/p_results(rsNat,"v","ra","betap","tot",%1,"%version%")
**p_results(rsNat,"v","ra","betap","tot",%1,"%version%")
* [ { ( pg.1(rsNat,"%t0%")/p_results(rsNat,"v","ra","betag","tot",%1,"%version%")
**p_results(rsNat,"v","ra","betag","tot",%1,"%version%") ) $ axg(rsNat,"%t0%") + 1 $ (not axg(rsNat,"%t0%")) ]
* [ { ( pi.1(rsNat,"%t0%")/p_results(rsNat,"v","ra","betas","tot",%1,"%version%")
**p_results(rsNat,"v","ra","betas","tot",%1,"%version%") )
$ ( p_results(rsNat,"v","ra","betas","tot",%1,"%version%") ge 0)
+ 1 $ (p_results(rsNat,"v","ra","betas","tot",%1,"%version%") lt 0) ];
```

As we sequentially updating over the product, the last product will measure the total EV of all price changes against the benchmark utility. We hence only consider differences between the sequential steps to the attribution to individual products:

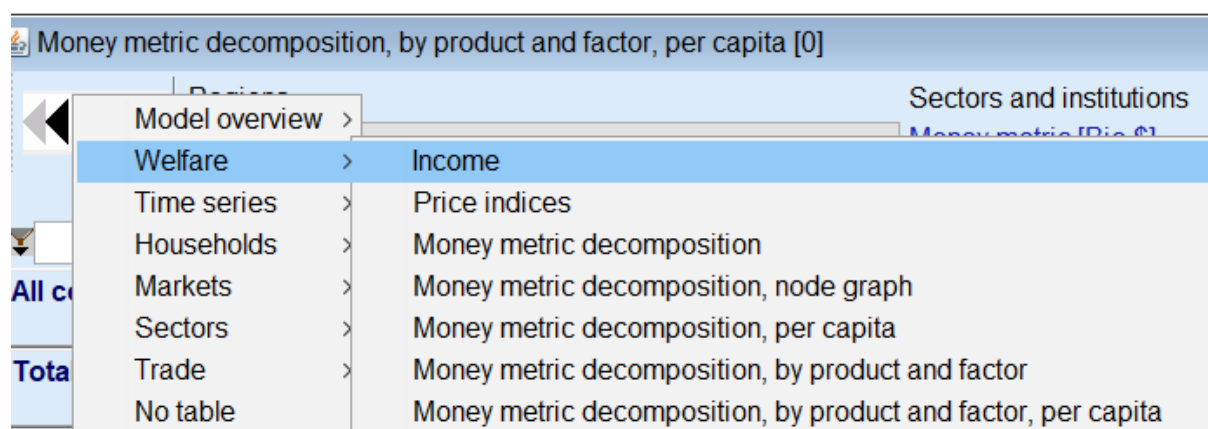
```
p_results(rsNat,"V",j,"mm","tot",%1,"%version%")
= p_results(rsNat,"V",j,"mm","tot",%1,"%version%")
- p_results(rsNat,"V",j-1,"mm","tot",%1,"%version%");
```

In order to capture differences between the in parts ad-hoc treatment and prices, the difference to the total is calculated:

```
p_results(rsNat,"V","diff","mm","tot",%1,"%version%")
= p_results(rsNat,"V","ra","mm","tot",%1,"%version%")
- sum(j, p_results(rsNat,"V",j,"mm","tot",%1,"%version%"))
- sum(f0, p_results(rsNat,"V",f0,"mm","tot",%1,"%version%"))
- p_results(rsNat,"V","dep","mm","tot",%1,"%version%")
- p_results(rsNat,"V","xt","mm","tot",%1,"%version%");
```

The results can be found in the following tables in the interface:

Note that due to the interaction between income and price changes, the total EV is not equal to the sum of the individual contributing elements. The results can be found in the following tables in the interface:



The first table reports the contribution of the overall changes in the prices indices, in factor income and indirect tax revenues:

Money metric decomposition [0]

Scenarios: Tariffs  
 Sectors and institutions: mm  
 Years shock

	World	Australia, New Zealand	East Asia	Southeast Asia	South Asia	North America	Latin America
Total [Bio \$]	55.17	2.48	54.39	4.56	-0.81	-11.63	
Overall price index [Bio \$]	644.96	3.80	-66.02	7.99	60.00	204.21	
Private price index [Bio \$]	477.05	2.16	-31.50	3.26	41.54	159.01	
Government price index [Bio \$]	69.10	-0.16	-29.90	-0.92	2.54	31.65	
Savings price index [Bio \$]	95.40	1.79	-4.76	5.64	15.36	13.06	
Income total [Bio \$]	-578.73	-1.42	120.19	-3.66	-58.97	-213.38	
Factor income [Bio \$]	0.00	4.45	118.65	15.88	-4.95	-44.63	
Depreciation [Bio \$]	81.90	1.17	-2.92	2.35	7.05	20.91	
Indirect tax income [Bio \$]	-660.63	-7.04	4.46	-21.89	-61.06	-189.66	

Note that the table was pivoted to show a comparison across countries. The global factor income change impact is by definition zero in the GTAP standard model as (1) factor stocks are fixed, (2) the global factor price index is used as the numeraire.

The decomposition discussed above is found in the “by product and factor” table, where “Sum” captures the total as shown above. It repeats the impacts of changes in nonal income (factor income, :

Money metric decomposition, by product and factor [0]

Regions: World  
 Sectors and institutions: Money metric [Bio \$]  
 Years shock

	World
All commodities	642.15
Total factor demand	0.00
Sum	55.17
Indirect taxes	-660.63
Depreciation	81.90
Interaction effects	-8.25
All commodities	642.15
Grains and Crops	21.37
Livestock and Meat Products	27.46
Mining and Extraction	2.50
Processed Food	76.85
Textiles and Clothing	51.84
Light Manufacturing	88.16
Heavy Manufacturing	94.03
Utilities and Construction	43.77
Transport and Communication	69.95
Other Services	166.23
Total factor demand	0.00
Land	-9.67
Capital	11.49
UnSkLab	-0.00
SkLab	-0.00
NatRes	-1.81

### 6.1.8 Decomposition following Huff and Hertel, 2000

The decomposition departs from the definition of regional income and allocates shares in total EV changes to changes in income contributions from changes in factor endowments at factor prices at the base year, from changes in quantities on which taxes are levied, from the trade balance etc.. The main differences to the approach above is that here the change in EV is attributed in shares to components of income.

For endowment changes, we use the absolute changes in endowment xft times factor prices pft at the benchmark:

```
p_deltaEV(rNat,"xft",%arg2%) = sum(f,pft(rNat,f,"%t0%") * (xft(rNat,f,%arg2%)-xft(rNat,f,"%t0%")));
```

Similar, for depreciation in case of changes in the capital stock:

```
p_deltaEV(rNat,"depr",%arg2%) = - fdepr(rNat,"%t0%")*pi(rNat,"%t0%")*(kstock(rNat,%arg2%)-kstock(rNat,"%t0%"));
```

For the taxes, the GAMS code copies the equation defining the different tax revenues into assignments, where the quantity is replaced by the quantity change, e.g.:

Tax revenues from production tax, equation (13) in VDM 2018

```
[ sum(a $ prdtx.1(rNat,a,"%t0%"), prdtx(rNat,a,"%t0%")*(m_px(rNat,a,"%t0%")*(m_xp(rNat,a,"%t0%")-m_xp(rNat,a,"%t0%"))
```

For the terms of trade effects, changes in f.o.b. and c.i.f. prices at benchmark trade flows are taken into:

```
p_deltaEV(rNat,"t.o.t",%arg2%) = sum( (i,rNat1), (%pefob(rNat,i,rNat1,%arg2%)-%pefob(rNat,i,rNat1,"%t0%")) *xw(rNat,i,rNat1,"%t0%"))
- sum( (i,rNat1), (%pefob(rNat1,i,rNat,%arg2%)-%pefob(rNat1,i,rNat,"%t0%")) *xw(rNat1,i,rNat,"%t0%"))
```

Plus the impact of changes in the trade margins for exports:

```
+ sum((m,i,rNat1) $ amgm(m,rNat,i,rNat1),
      amgm(m,rNat,i,rNat1)
      * ( m_xws(rNat,i,rNat1,"%t0%") $ rrComb(rNat,rNat1)
          $onexpand
          + m_xws1(rNat,i,rNat1,"%t0%") $ (not rrComb(rNat,rNat1))
          $offexpand
        )
      * tmarg(rNat,i,rNat1,"%t0%")/m_lambdamg(m,rNat,i,rNat1,"%t0%")
      --- margin prices in region r ???
      * (ptmg(m,%arg2%)-ptmg(m,"%t0%"))
    )
```

And imports:

```
--- value of transport margins for imports of rNat times changes in margins price
- sum((m,i,rNat1) $ amgm(m,rNat1,i,rNat),
      amgm(m,rNat1,i,rNat)
      * ( m_xws(rNat1,i,rNat,"%t0%") $ rrComb(rNat1,rNat)
          $onexpand
          + m_xws1(rNat1,i,rNat,"%t0%") $ (not rrComb(rNat1,rNat))
          $offexpand
        )
      * tmarg(rNat1,i,rNat,"%t0%")/m_lambdamg(m,rNat1,i,rNat,"%t0%")
      * (ptmg(m,%arg2%)-ptmg(m,"%t0%"))
    );
```

The final changes stems from updating prices of investments:

```
p_deltaEV(rNat,"fsav",%arg2%) = netInv(rnat,"%t0%") * (pi(rNat,%arg2%)-pi(rNat,"%t0%"))
- yi(rNat,"%t0%") * (psave(rNat,%arg2%)-psave(rNat,"%t0%"));
```

We assign the total EV in shares to these individual effects:

## Decomposing welfare changes in CGEBox

```
p_results(rnat,"v","ev",evEff,"tot",%arg1%, "%version%") $ p_deltaEV(rNat,"tot",%arg2%)
= p_results(rNat,"v","ra","mm","tot",%arg1%, "%version%")/p_deltaEV(rNat,"tot",%arg2%)*p_deltaEV(rNat,evEff,%arg2%);
```

And aggregate also to regional aggregates:

```
p_results(rAgg,"v","ev",evEff,"tot",%arg1%, "%version%")
= sum(ragg_r(ragg,rnat), p_results(rNat,"v","ev",evEff,"tot",%arg1%, "%version%"));
```

The results can be inspected in the inspected on a per capita basis:

EV decomposition, per capita [0]											
Years shock											
Tariffs											
	World	Australia, New Zealand	East Asia	Southeast Asia	South Asia	North America	Latin America	European Union 25	Middle East and North Africa	Sub-Saharan Africa	Rest of World
Total	8.33	71.36	35.07	8.03	-0.53	-26.44	-6.23	12.04	0.94	-1.03	11.18
Terms of trade	3.04	29.33	13.86	-1.86	0.16	-2.56	1.68	-0.18	-0.55	0.38	-3.82
Foreign savings	-0.86	16.93	-1.22	3.11	-0.16	-18.09	-4.76	6.39	0.81	-0.77	4.14
Depreciation	-0.00		-0.00		-0.00			-0.00		0.00	0.00
Factor stocks	-0.00		-0.00	0.00	0.00	-0.00	0.00	-0.00	-0.00	0.00	0.00
Import taxes	3.22	18.12	10.87	5.74	-0.34	-4.35	-3.16	4.82	0.60	-0.52	7.22
Export taxes	1.46	0.11	6.13	0.25	-0.11	-0.17	-0.15	-0.10	0.00	-0.05	1.31
Taxes on firm consumption	0.04	0.38	0.13	-0.02	-0.02	-0.12	-0.16	0.22	0.07	-0.01	0.34
Taxes on private consumption	0.66	5.49	2.29	0.70	-0.02	-1.20	0.08	0.60	-0.01	-0.07	1.62
Taxes on public consumption	0.00	-0.02	0.01	0.00	-0.00	-0.00	0.01	0.00	-0.00	0.01	-0.02
Taxes on investment consumption	0.51	1.74	2.08	0.10	-0.04	-0.14	0.14	0.04	0.01	0.01	0.17
Taxes on factors	-0.02	-0.02	-0.07	-0.01	-0.00	-0.00	-0.00	0.01	-0.00	-0.00	-0.00
Subsidies on factors	0.27	-0.69	0.98	-0.00		0.19	0.09	0.24	0.00	0.00	0.22

Or as totals:

EV decomposition, total [0]	
Years shock	
Tariffs	
World	
Total	55.17
Terms of trade	20.14
Foreign savings	-5.72
Depreciation	-0.00
Factor stocks	-0.00
Import taxes	21.34
Export taxes	9.67
Taxes on firm consumption	0.29
Taxes on private consumption	4.36
Taxes on public consumption	0.02
Taxes on investment consumption	3.37
Taxes on factors	-0.11
Subsidies on factors	1.80

## 6.2 Trade in VA indicators

### 6.2.1 Background

Trade has a double role by firstly allowing consumers to not only demand domestic produce, but also imports and secondly offering income opportunities by exports. The first role can be easily assessed in a CGE framework by reporting import shares on demand. The second one is harder to assess as export revenues comprise both direct value added from the exporting sector and indirect value added comprised in domestic intermediates. In the following, we present the implementation of a well-established framework drawing on Leontief Multiplier analysis into the post-processing of CGEBox to assess the income generate role of exports.

We start with an illustrative example of an open economy with 2 two sectors and no foreign savings. Imports serve in our example only as intermediates inputs are not detailed by commodity:

	Agr	NonAgr
Agr share	0,1	0,05
NonAgr share	0,4	0,5
Import share	0,2	0,1
VA share	0,3	0,35
Output	30	80
VA	9	28
Final demand incl. Exports	23	28
Exports	2	12
Final domestic demand	21	16

The example is balanced as the export revenues of  $14 = 2 + 12$  are equal to the value of imports  $(,2 * 30 + ,1 * 80) = 12$ . The question is about the contribution of the exports (or domestic sales) to value added. Based on usual Leontief approach, we first calculate  $(I-A)$ , where  $I$  is the identity matrix and  $A$  the matrix of domestic input coefficients:

In our example that yields the following 2x2 table:

0,9	-0,05
-0,4	0,5

Inverting that matrix delivers the Leontief multipliers  $(I-A)^{-1}$ :

1,1627907	0,11627907
0,93023256	2,09302326



Multiplying the value added shares of the sectors with the Leontief multipliers delivers the VA multipliers for each commodity:

0,6744186 0,76744186

If we now multiply these with final demand and export, we derive the VA contribution of the demand contributions:

Final domestic demand contribution	26,4418605
Export contribution	10,5581395
<b>Total Value added</b>	<b>37</b>

Which are equal to the value added as reported in our simple SAM and, as the SAM is balanced also equal to final domestic demand. The aim is now to implement the above steps into the CGEBOX framework.

### 6.2.2 Post-Model processing

As a first step, we define (I-A) as follows, where the p\_results parameter was populated before from the model results. Note that we are using the “V” field which captures value (= quantity x price):

```
--- invert (I-A)
execute_unload '%scrdir%\p_IA.gdx',i,p_IA;
execute '=invert.exe %scrdir%\p_IA.gdx i p_IA %scrdir%\p_IAInv.gdx p_IAInv';
execute_load '%scrdir%\p_IAInv.gdx',p_IAInv;
```

Next, we use the “INVERT” utility from GAMS to define  $(I-A)^{-1}$ :

```
--- invert (I-A)
execute_unload '%scrdir%\p_io.gdx',i,p_IA;
execute '=invert.exe %scrdir%\p_IA.gdx i p_IA %scrdir%\p_IAInv.gdx p_IAInv';
execute_load '%scrdir%\p_IAInv.gdx',p_IAInv;
```

From there, we calculate multipliers for each primary factor – commodity combination:

```
--- define multipliers for each commodity j and each primary factor f
p_results(rs,"Q",j,f,"tradeInVa",%1,"%version%")
= sum( (a,i) $ xFlag(rs,a,i),
      p_results(rs,"U",f,a,"dom",%1,"%version%")/p_results(rs,"U","out",a,"dom",%1,"%version%") * p_IAInv(i,j));
```

Which can be aggregated to yield value added multipliers

```
--- aggregate to VA multipliers
p_results(rs,"Q",j,"va","tradeInVa",%1,"%version%") = sum(f, p_results(rs,"Q",j,f,"tradeInVa",%1,"%version%"));
```

Using these multipliers and export volumes allows to calculate the VA contribution of exports:

```

*
*      --- calculate income by primary factor embodied in trade
*
p_results(rs,"U",j,f,"tradeInVa",%1,"%version%")
= p_results(rs,"Q",j,f,"tradeInVa",%1,"%version%") * p_results(rs,"Q",j,"exp","dom",%1,"%version%");
*
*      --- calculate value added embodied in trade
*
p_results(rs,"U",j,"va","tradeInVa",%1,"%version%") = sum(f, p_results(rs,"U",j,f,"tradeInVa",%1,"%version%"));

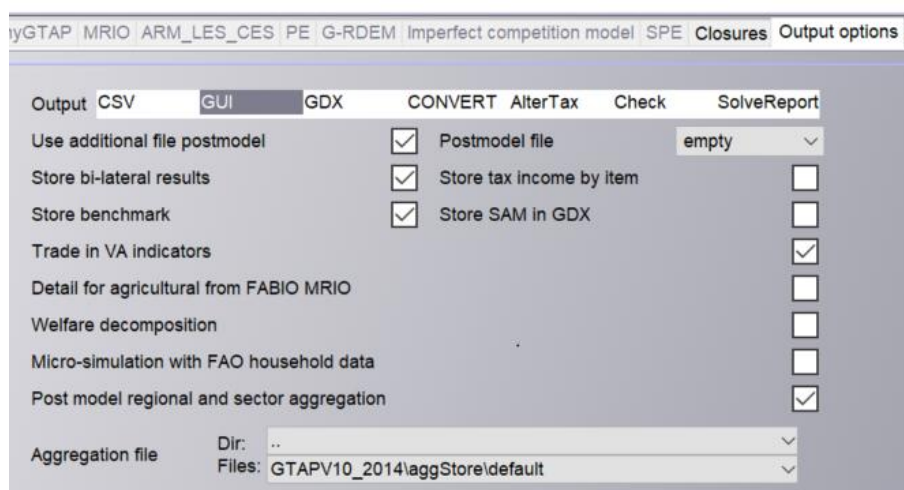
```

Further code lines not shown here aggregate the information over regions and commodities.

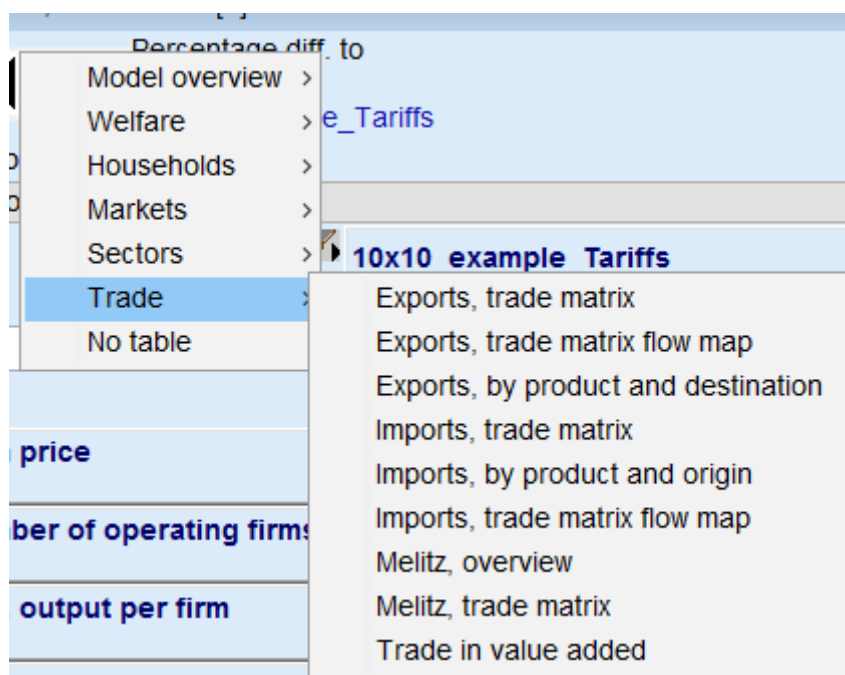
The TIVA approach also defines output multipliers. This allows assigning CO2 equivalent discharges (taxed and emitted) allocated to production activities (linked to intermediate or factor use, process emissions) in delivering countries to final demand in use countries. The resulting embodied emissions per constant unit dollar are stored in temporary file “embed.gdx” in the scratch directory and can be used, for instance, for simulation of a Carbon Border Adjustment mechanism.

### 6.2.3 GUI

In order to generate the information, the “GUI” output option must be switched on as well as the “Trade in VA indicators” checkbox:



That combination produces a table under the theme “trade”:



Which reports both the multipliers and the contribution of trade to value added. Additionally, the share from export related value added on total income is reported:

The screenshot shows a software interface with a table of trade in value added indicators. The table has four columns: 'Per unit of output', 'In total trade', 'In trade, relative to GDP', and 'Ir'. The rows are categorized by sectors and institutions, including Total, Grains and Crops, Livestock and Meat Products, Mining and Extraction, Processed Food, Textiles and Clothing, Light Manufacturing, Heavy Manufacturing, Utilities and Construction, Transport and Communication, and Other Services.

	Per unit of output	In total trade	In trade, relative to GDP	Ir
Total	0,7508	9889,7246	0,1986	
Total	0,7508	9889,7246	0,1986	
Grains and Crops	0,9726	257,4330	0,0052	
Livestock and Meat Products	0,8502	122,8059	0,0025	
Mining and Extraction	0,8705	1186,6638	0,0238	
Processed Food	0,7638	425,7714	0,0085	
Textiles and Clothing	0,6706	400,4073	0,0080	
Light Manufacturing	0,6488	1697,5337	0,0341	
Heavy Manufacturing	0,6005	3742,0999	0,0751	
Utilities and Construction	0,7302	113,5394	0,0023	
Transport and Communication	0,7690	747,9837	0,0150	
Other Services	0,8304	1195,4865	0,0240	

As seen below, the table also provides detail by factor:

The screenshot shows a software interface with a table of trade in value added indicators by factor. The table has four columns: 'Per unit of output', 'In total trade', 'In trade, relative to GDP', and 'Ir'. The rows are categorized by sectors and institutions, including Total, Grains and Crops, and Land. The 'Land' row is highlighted, showing a value of 118,6026 for 'In trade, relative to GDP'.

	Per unit of output	In total trade	In trade, relative to GDP	Ir
Total			118,6026	
Total	0,0092		118,6026	
Grains and Crops	0,2257		53,2937	



## 6.3 Altertax

### 6.3.1 Background

There are various use cases where it makes sense to construct a new benchmark from the GTAP data base before shocking the model instead of comparing different shocks against each other where all comprise the same changes against the benchmark. One option to construct a new benchmark changes first entries in the global SAM and next re-balances it. That approach is especially suitable if benchmark changes are motivated by new or updated data. The filtering approach implemented in CGEBox can be seen as a specific variant of that approach.

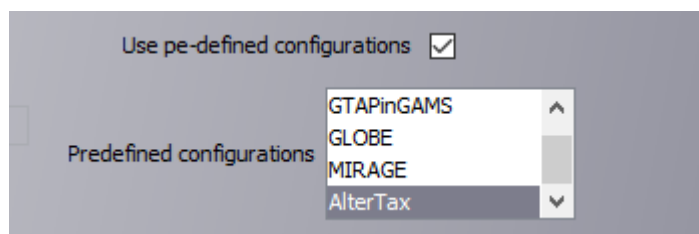
The option discussed in here uses the results of the simulation model itself. There are two possible motivations for it. First, this approach is especially appropriate for changes where the behavioral responses embodied in the equation structure provide a better way to re-adjust the benchmark compared to the more mechanical re-balancing the approaches. Second, one might want to conduct comparative-static analysis to the outcome of baseline process with the G-RDEM model.

Conceptually, re-using the simulation outcome as a benchmark data base set is straightforward as the accounting identities underlying a balanced SAM must also be fulfilled by a simulation with the CGE itself. One main challenge is to map the simulated results back in same format used on the input side of the simulation model. As CGEBox is a SAM based CGE approach, that step is not overly complex. Second, additional data sets, for instance, relating to land use or emissions, might not have included in the simulation process but shall still be made available for subsequent benchmarking. The brief following documentation will discuss the basic methodology and technical implementation

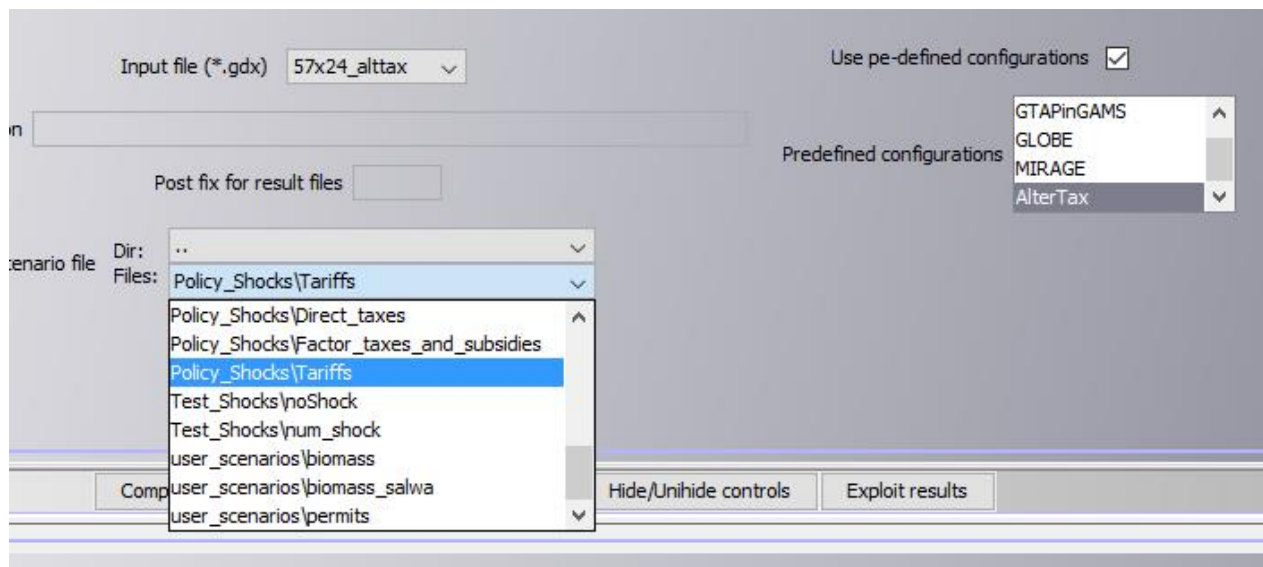
### 6.3.2 Technical realization

The technical realization re-uses existing elements of CGEBox to implement alterTax:

1. A predefined configuration which can be chosen from the interface set-ups a simulation model useful for alterTax, i.e. the parameterization of the model is changed such that most substitution elasticities are unity (or close to unity).



The user can select any shock file:



6. As part of post-processing, the relevant entries for a start of the model define a new benchmark, including the global SAM.

Note: The alterTax output can be also generated from any normal model run, in case the pre-configuration given next is not suitable.

### 6.3.3 Pre-configuration

The pre-configuration changes the following elements compared to the standard GTAP Model:

- The demand system for private households is changed to a CD.
- The production nesting using substitution elasticities of unity between VA and the intermediate composite, between primary factors and between intermediate.
- The Armington elasticities are set close to unity (a CD implementation is currently not supported).
- All factors are mobile

The use of the CD functional form allows to keep the value shares constant during the alterTax simulations which thus maintains basic relations between transactions in the SAM. If highly detailed global SAMs are subject to AlterTax, the per-solve mechanism can be used as well.

### 6.3.4 Post-Model processing

The SAM output option is switched on and the results for the shock are stored again as a SAM in a new data file. The same holds for various sub-matrices, for instance:

```
xda0(r,i,aa)    $ xda0(r,i,aa)    = xd.l(r,i,aa,"shock") * pdp.l(r,i,aa,"shock") / gblscale;
```

The update symbols are stored to a new data file:

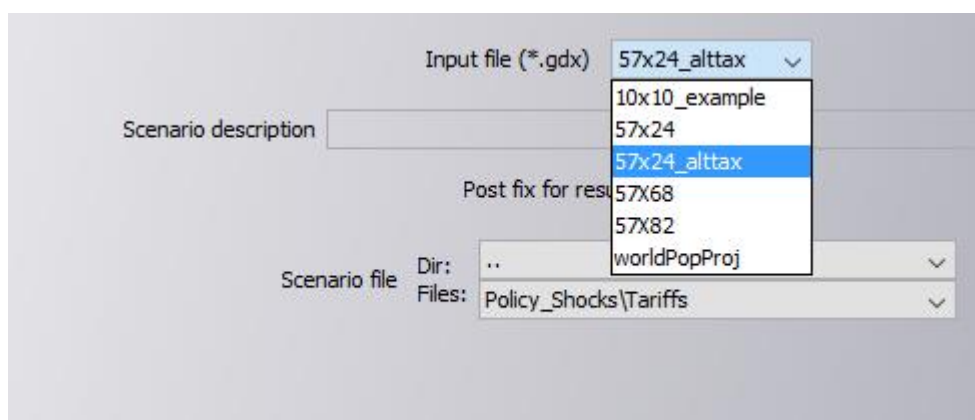
```

execute_unload "%datdir%\%dataset%\%postfix%\alttax.gdx", is, aa, a, i, r, f, l, cap, lnd, nr, fd, h, gov, inv,
sam0, xda0, xdn0, xna0, xmm0, xmarg0,
eh0, bh0, sigmap0, sigmav0, sigmai0, sigman0, sigmaw0, omegaf0, RoRflex0,
kstock0, depry0, pop0,
emid0, emii0,
exptxY0, imptxY0, fcttxY0, fcttsY0
mapi=ii, mapr=rr, m, h, mapa=aa1, ff,
$$ifthen defined rAez
    rAez, rAez_r_aez, landUse, AEZ, land_cover
$$endif
*$$ifi "%filterMethod%"=="Rebalancing" itrlog, keepCor.1
*$$ifi "%loadLandUse%"=="on" landUse, AEZ, land_cover
    $$ifthen defined s_meta
        s_meta
    $$endif
;

```

### 6.3.5 Use of the updated benchmark

The updated benchmark data set is stored in the data directory with the post-fix “alttax” appended and can be used as any other data set:





## 6.4 Linking the physical MRIO for agricultural and food products FABIO to CGEBox

- Wolfgang Britz, Martin Bruckner, Salwa Haddad –

### 6.4.1 Introduction

FAOSTATs offers a wealth of data on global agriculture including bi-lateral trade, herd sizes and harvested area for UN Member countries and many agri-food products. These data were in a systematic way integrated in the Multi-Regional Input-Output (MRIO) data base FABIO (ref). We present in here two approaches to link CGEBOX (Britz 2017) as a global CGE to FABIO. The first one uses FABIO in a generic way to split up the global GTAP SAM used by CGEBox to more detail on agricultural and related processing activities where the desired additional detail is user defined. The second approach uses FABIO to disaggregate post-model the more aggregated results simulated with CGEBOX, in order to generate detailed results at country and product level for individual agricultural and selected derived products and derive from there nutritional indicators.

### 6.4.2 Overview on FABIO

FABIO is a MRIO in physical and partly monetary terms for agri-food products. In opposite to typical national IO-tables, an MRIO describes inter-industrial relations on a country-by-country basis, i.e. the intermediate input sourcing of each industry in each region is dis-aggregated by imported origin. It is hence not only possible, to give an example, to quantify how much soy bean is used in China's soy bean crushing industry, but also from which countries the industry imports that intermediate input. That is different from the global GTAP SAMs underlying CGEBox which depict bi-lateral trade flows, but not differentiated by agent.

FABIO covers around 130 agricultural and related processing activities. Furthermore, it provides harvested areas and herd sizes in heads for the agricultural activities. The main data underlying FABIO are different statistical data sets from FAOSTAT. Building on FAOSTAT, it uses its concept of primary product equivalents to convert trade and consumption in derived products such as wheat flour and bread back to wheat. That is one major difference to a SAM used in a CGE. The second one is that FABIO is focusing on agri-food, i.e. it reports intermediate input between agri-food items, only.

The original FABIO data base is handled in R, using numerical linear indices to yield two dimensional matrices of bilateral final demand, intermediate demand etc. The version provided in GDX container comprises two four-dimensional GAMS parameters with the following structure and the necessary sets with the list of items (= products), ISO codes for the regions and processes.

Entry	Symbol	Type	Dim	Nr Elem
3	ISO	Set	1	192
4	items	Set	1	130
2	p_fablnv	Par	4	17,722,530
1	p_fabio	Par	4	3,533,855
5	processes	Set	1	121

The first parameter comprises the global MRIO with additional information the parameter is called “p\_fabio”:

1. First dimension: exporter (= supplier) region or “tot” for sum over all export regions. The regions are coded by 3-character ISO codes
2. Second dimension: the 130 FABIO items, i.e. products, in long text

3. Third dimensions: importer (= use region)
4. Fourth dimension: use categories (food, other, stock changes, balancing or the 121 FABIO processes, long texts)

The total supply is stored under “prod” on the third dimension, the fourth dimension is then empty “”. The same holds for the acreage, stored under “land” with the fourth dimension again shows an empty “”. In order to ease processing, bi-lateral exports respectively domestic sales are stored in the position “exp”.

The following shows an example “Potatoes and products” for “DNK” = Denmark. As the supplier is equal to the demander region, the data refer to domestic sales. The first position, potatoes used in its production refer to seed potatoes. The largest use position is sweetener production, the rest goes to food (including not specified food processing activities such as producing French Fries). Small quantities are reported for food use.

DNK	Potatoes and products	DNK	Potatoes production	318390
			Sweeteners production, Other	643413
			Cattle husbandry	404
			Sheep husbandry	1
			Pigs farming	8827
			Poultry Birds farming	577
			Horses husbandry	83
			Dairy cattle husbandry	663
			food	398517
			exp	1370874

At the end of the entries for the producer country are the two position with total production and related land use:

DNK	Potatoes and products	prod	<empty>	1640505
		land		41600

The second container comprises the symmetric Leontief inverse.

### 6.4.3 Methodology

#### 6.4.4 Introduction

A MRIO such as FABIO can be used without a CGE to answer stand-alone questions in the tradition of IO-analysis such as “what would be impact on production in the different countries and land use if final consumption of a specific product in a specific country would increase by x%”? That requires constructing the so-called Leontief-Inverse. However, there are major differences between such a multiplier analysis and running a shock in a CGE which motivate the links detailed below. Firstly, the Leontief-Multiplier analysis with a MRIO assumes that physical shares in bi-lateral input use are constant, i.e. if beef producers in country x in the observed period are assumed to use 100 kg soy cake from country y per ton of beef outputted, that relation cannot change in a simulation. CGE models use here the Armington mechanism or variants thereof which update bi-lateral trade shares based on relative price changes. The same holds also for land use, in case of beef production, applying the Leontief inverse implies that stocking rates in beef production can’t change; it thus also excludes that concentrate use and grazing act as (imperfect) substitutes. The consequence of keeping crop yields and stocking rates constant is that global land use expands proportionally to production. Here, CGEs

assume either that the economically used land stock is fixed or that land supply reacts to price changes, in both cases, the composition at least of value added (labour, capital, land, natural resources) can change in the production functions of the different sectors. That also implies that yields and stocking rates in a CGE analysis depend on price relations.

Thus, both an IO-multiplier analysis and a CGE use the given data to derive the relation between inputs and outputs at the benchmark and calibrate from there production functions. The IO-multiplier analysis keeps these relations fixed, i.e. it assumes a Leontief technology such that average and marginal technology are identical. That implies completely elastic supply of primary production factors. A CGE instead typically assumes that primary production factor stocks are fixed at the level of the total economy or even of individual sectors – the latter typically assumed in case for natural resources. Under that assumption, adjustments in the structural composition of the economy require some flexibility in the production function such that at least for part of the technology, marginal input-output relations are not constant.

A multiplier-analysis is not an economic model in the strict sense since there is no choice of the agents involved, such that also no behavioural assumptions are necessary: final demand is changed exogenously, and industry output adjusts according to fixed intra-industry relations in intermediate demand, updating primary factor use such that it matches the physical demand emanating from the updated industry composition. In opposite to that, a CGE makes assumptions about market power and product differentiation, behaviour of the final demanders and producers as well as factor suppliers, and solves the model under market clearing conditions in all good and factor markets as well as macro-economic accounting identities.

#### **6.4.5 Split**

##### **Approach**

The split approach combines to a large degree the advantages of using detailed physical data as provided by FABIO and the economic adjustment mechanisms captured by a CGE. Specifically, it reduces the aggregation bias stemming from aggregating agricultural and derived products to rather broad sectors.

The shortcomings of the split approach relate to the different content covered by FABIO as a physical IO for agriculture and food-processing and the global GTAP SAM. Firstly, FABIO does not offer input coefficients beyond agri-food, i.e. there are no data on intermediate input use of products such as diesel, fertilizers, plant protection products, insurance, contract work. Secondly, land use is the only reported primary factor. Thirdly, the GTAP SAM covers different tax (respectively subsidy) rates such as output, consumer, import and export taxes which might differ inside a GTAP sector, information also missing in FABIO such that the assumption of similar cost shares and equal tax rates is used as a default. For import taxes, i.e. tariffs, the TASTE data base is linked into the split process to provide data on bi-lateral trade and import tariffs at HS6 tariff line level which are consistent to the GTAP Data itself.

23 agri-food processes

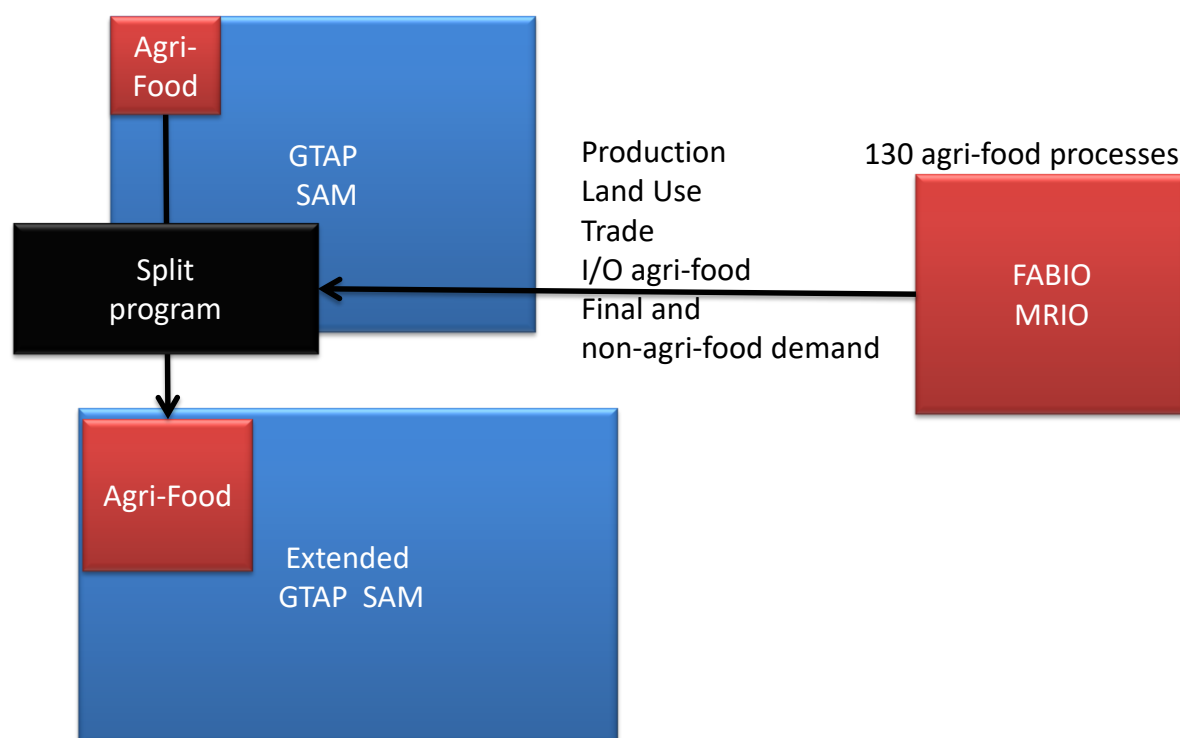


Figure 1: Schematic overview on split approach

### Technical integration

The split is performed based on the GAMS code of the generic split program comprised in CGEBox, see the dedicated section in this documentation and (Britz 2021<sup>116</sup>). It requires as a minimum split shares for production and the different final demand categories, assuming equal cost shares, tax rates etc. if no further detailed data are provided. FABIO does not only provide these minimum data, but offers dis-aggregated data on bi-lateral trade and parts of intermediate input use which contribute additional split factors.

In order to split up the SAM of CGEBox based on FABIO, the user needs first to provide lists with the new products and activities and their relations. The names `s_spliti`, `s_splita` and `ia` for these sets are fixed to allow for a proper integration in the overall data generation process of CGEBox:

<sup>116</sup> Britz, W. (2021): Comparing Penalty Functions in Balancing and Dis-aggregating Social Accounting Matrices, *Journal of Global Economic Analysis* 6(1): 34-81

```
set s_spliti / olv-c "Olives"
               soy-c "Soy bean"
               oso-c "Other oilseeds"

               olvo-c "Olive oil"
               volo-c "Other vegetable oils and cakes"
               /;

*
set s_splita / olv-a  "Olives"
               soy-a  "Soy beans"
               oso-a  "Other oilseeds"
               olvo-a "Olive oil"
               volo-a "Other vegetables oils and cakes"

               /;

set i_a(*,*) / olv-c.olv-a
               soy-c.soy-a
               oso-c.oso-a
               olvo-c.olvo-a
               volo-c.volo-a
               /;
```

Note that it will in most applications necessary to define a residual category when splitting up the GTAP SAM based on FABIO. For the example of oilseeds (primary agriculture) and vegetables cakes and oils (a food processing sector) shown above, FABIO offers a lot of detail of which only data on olives and soy are used in the example code. We hence need to assign the not explicitly assigned products and activities ones in a “rest-of” or “other” category to allow exhausting the data found in the SAM at aggregate level.

Furthermore, the link of the newly introduced products to the GTAP sectors needs to be provided:

```
set is_i(*,*) / (olv-c,soy-c,oso-c).osd-c
                (olvo-c,volo-c).vol-c
                /;
```

The set definitions show above must be in the block following:

```
$iftheni.decl "%1"=="decl"
```

That position in the code ensures that the newly defined sectors are properly integrated in the list of SAM rows and columns.

The main additionally information which needs to be provided by the user to make use of the FABIO data base is the link between the newly introduced products and activities (spliti and splita from above) and the products and activities of FABIO data base, as shown for the example below:

```
set fabC_spliti(fabC,spliti) /
    "Olives (including preserved)"      ".olv-c"
    "Soyabeans"                        ".soy-c"
    "Olive oil"                        ".olvo-c"
    /;

set fabA_splitA(fabA,splitA) /
    "Olives production"                ".olv-a"
    "Soyabeans production"             ".soy-a"
    "Olive Oil extraction"              ".olvo-a"
    /;
```

The remainder of program is more or less standardized:

```

fabC_i(fabc,spliti) $ fabc_spliti(fabc,spliti) = YES;
fabA_a(faba,splita) $ fabA_splitA(faba,splita) = YES;

fabC_i(fabc,"oso-c") $ (fabC_i(fabc,"osd-c") $ (not sum(fabc_spliti(fabc,spliti),1))) = YES;
fabC_i(fabc,"volo-c") $ (fabC_i(fabc,"vol-c") $ (not sum(fabc_spliti(fabc,spliti),1))) = YES;

fabA_a(faba,"oso-a") $ (fabA_a(faba,"osd-a") $ (not sum(faba_splita(faba,splita),1))) = YES;
fabA_a(faba,"volo-a") $ (fabA_a(faba,"vol-a") $ (not sum(faba_splita(faba,splita),1))) = YES;

$$include 'split\split_based_on_fabio.gms'
*
* --- input demand for oil seed processing
*
p_splitFactor(rNat,"volo-a","olv-c") = eps;
p_splitFactor(rNat,"olvo-a","olv-c") = 1;

p_splitFactor(rNat,"olvo-a","oso-c") = eps;
p_splitFactor(rNat,"olvo-a","soy-c") = eps;
p_splitFactor(rNat,"volo-a","oso-c") = 1;
p_splitFactor(rNat,"volo-a","soy-c") = 1;
    
```

The last block which excludes certain IO-relations or assigns existing flows fully to others is optional. Here, as seen, the “other oilseeds processing” activity “volo-a” is not allowed to use olives “olv-c”, they must be fully processed by the “olvo-a” activity. The same holds for soy beans and other oil seed: they cannot enter the olive processing activity and enter instead the residual one.

All follow operations are already coded in GAMS (see build\split\_based\_on\_fabio.gms). That program firstly calculates average f.o.b. prices to link the physical quantities in FABIO to the SAM in monetary terms:

```

parameter p_fobPrices(*,fabc);
execute_load "%datDir%\fob_fabio.gdx" p_fobPrices;
p_fobPrices("wor",fabc) = sum(fabr $ p_fobPrices(fabr,fabc),1);
p_fobPrices("wor",fabc) $ p_fobPrices("wor",fabc)
    = sum(fabr,p_fobPrices(fabr,fabc))/p_fobPrices("wor",fabc);
p_fobPrices(fabr,fabc) $ (Not p_fobPrices(fabr,fabc)) = p_fobPrices("wor",fabc);
p_fobPrices(fabr,fabc) = p_fobPrices("wor",fabc);
p_fobPrices(fabr,fabc) $ (Not p_fobPrices(fabr,fabc)) = 0.001;
    
```

Next, it assigns amounts of private and government demand based on the mappings from above between the new products and the FABIO ones fabc\_i. The assignment is conditional on such consumption being reported in the SAM, i.e. `sum(is_i(splitiF,i),sam0(rNat,i,dem))`. Furthermore, the demand categories “int” and “inv” are excluded. Last, the split product must have a link to FABIO which renders it possible to combine the split based on FABIO with splits based on other data sets. The summation considers the mapping fabr\_r between the FABIO regions fabr and the regional aggregates rNat in the GTAP SAM from GTAPagg, that mapping is automatically set-up elsewhere in the GAMS code and does not need to be provided by the user. The same holds for the link between the FABIO products fabc and the product in the SAM splitiF which are linked to the FABIO data base:

```

p_splitFactor(rNat,splitiF,dem) $ (sum(is_i(splitiF,i),sam0(rNat,i,dem))
    $ (not (sameas(dem,"int") or sameas(dem,"inv"))))
    = sum((fabr_r(fabr,rNat),fabc_i(fabc,splitiF)),
        p_fabio("tot",fabc,fabr,"food")*p_fobPrices(fabr,fabc)) + eps;
    
```

Similar mappings are defined for investment and intermediate demand, production and bi-lateral trade flows. The split program (build\split.gms) will convert the absolute quantities assigned to split factors above in shares.

Note that firstly missing information in FABIO on tariff and export subsidy rates and output taxes will imply that the average ones from the SAM will be used. There is currently also no GAMS code which could handle more specific information in that case.

Secondly, FABIO does only comprise physical input relations between products covered by FAOSTAT and land use. All other cost shares are taken from the SAM, but these can be overwritten by the user, while sector specific input tax rates are currently not supported.

#### **6.4.6 Post-model dis-aggregation**

As an alternative to introduce additional detail in the SAM which makes the CGE model larger and thus potentially harder to solve, a post-model dis-aggregation can be used instead or additionally. It is based on proportionality assumptions.

#### **Methodology**

The basic methodology is very simple: we proportionally shock demand in FABIO based on more aggregated CGE results and apply the Leontief inverse from FABIO to inform on changes in supply, harvested acreage and herd sizes. We hence neglect the manifold differences underlying global CGE modelling and multiplier analysis in MRIO, not aiming at consistency. These differences should however be understood in order to properly interpret the findings, and we will comment briefly on some. The data base of the global CGE is defined in constant US dollars such that price and quantity changes are expressed against that benchmark. GTAP differentiates strictly between primary agricultural products and processed ones, whereas the FAO uses the concept of primary product equivalents. Thus, whereas wheat, wheat consumption and trade – final or intermediate – in GTAP refers to wheat only, FAO uses conversion factors to convert trade and consumption of derived products such as flour, break etc. back into wheat equivalents. Whereas in some cases, there is very limited consumption and trade in derived form reported in FAOSTAT, such as in the case of soy beans, substantial shares might be found in others. It is however important to note that dairy (split up in butter and the rest), vegetable oil and cakes and meats are separate products in FAOSTAT and FABIO, covering thus some important processed food categories a. Besides production, also factor use in GTAP is reported in constant dollar. Thus, whereas FABIO reports land use in harvested hectares, GTAP reports economic returns to land.



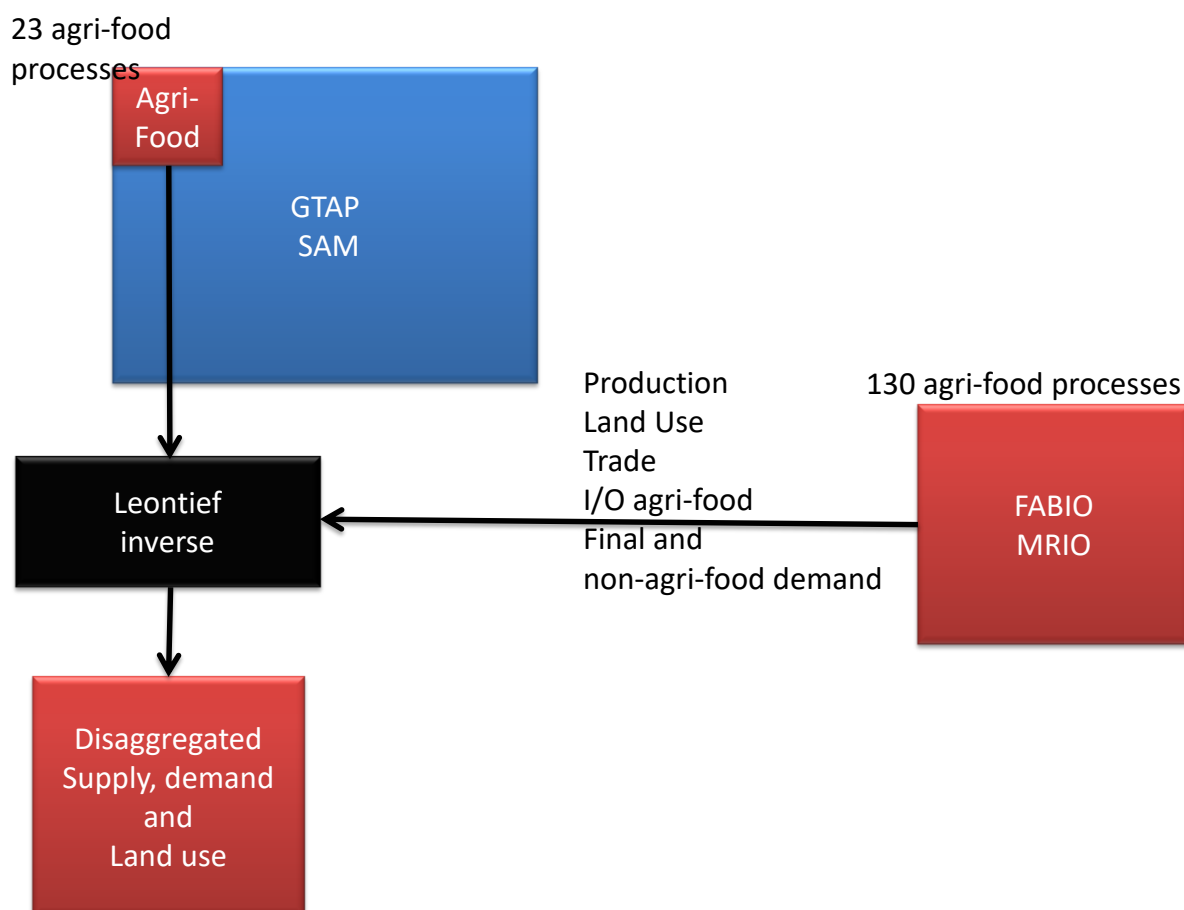


Figure 2: Overview on post-model dis-aggregation

Besides differences in data definition, the fundamental differences between a Leontief multiplier analysis and a CGE simulation should be kept in mind. A Leontief multiplier analysis assumes that the physical input / output relations remain constant, such as physical yields per hectares as the relevant case for FABIO. For a MRIO, it also implies that bi-lateral trade shares are fixed. A CGE uses CES production function where input / output relations might react to price changes; the Armington assumption (potentially in a combination with a Constant Elasticity of Transformation function to distribute supply) will also update trade shares. Even more importantly, the Leontief multiplier analysis implies that factor demand will fully adjust to changes in supply, where the CGE analysis typically assumes fixed economic factor stocks or some price responsiveness.

### Technical implementation

The main steps in the approach are detailed next, commenting directly on the GAMS code. Firstly, we need a mapping between the GTAP regions and sectors and those in FABIO. Both provide ISO country codes, such that we can directly use that information:

```
fabr_r(fabr,r) $ sum(rrmap(gisr,gtapr) $ (mapr(gtapr,r) $ fabr_ISO(fabr,gisr)),1) = YES;
```

The cross-set `fabr_r` defines the mapping between the individual regions in FABIO `fabr` and the current regions used in CGEBox (see `fabio\fabio.gms`). It is derived from links between the ISO country codes stored in the set `gisr` to both the FABIO regions (`fabr_ISO`) and the countries and

regions in the full GTAP data base rmap. Finally, we use the information on how the GTAP data is aggregated to the regions in the model, stored in mapr.

The mapping between the items in FABIO fabc and the original GTAP sectors iDat is defined manually in the cross-set fabc\_IDAT (see fabio/fabio.gms):

```
set fabc_IDAT(fabc,iDat) /  
    "Rice (Milled Equivalent)      ".pdr  
    "Wheat and products            ".wht  
    "Barley and products           ".gro  
    "Maize and products            ".gro  
    "Rye and products              ".gro
```

Taken the sectoral aggregation into account, the mapping between the current sectors in the model *i* and the FABIO items is defined as:

```
fabc_i(fabc,i) $ sum( (fabc_IDat(fabc,iDat),mapi(iDat,i)),1) = YES;
```

In order to define the food demand in the regional and item resolution in FABIO we need to update the FABIO data matrix *p\_y* whose rows and columns are not labelled. We hence need a mapping from the regional and use label to the index position which is provided by the three dimensional set *fabr\_dem\_j*. Similarly, the three dimensional set *fabr\_fabc\_i* shows the linear index *i* belonging to a country and product combination.

In order to take into account that most of the food demand by final consumers is not at the stage of primary agricultural products, but at the stage of processed products, we convert with a Leontief inverse derived from the GTAP IO-coefficients into primary product equivalents, not considering feed demand which is separately handled by FABIO. The set *excludeIJ(i,j)* captures these exemptions:

```
excludeIJ(i,j) $ sum( (rnat,cropBasedA,feedUseA) $ (xFlag(rNat,cropBasedA,i) $ xFlag(rNat,feedUseA,j)),1) = YES;
```

We define first the (1-A) matrix from the GTAP IO-coefficients, i.e. total use of GTAP commodity *i* in sector *a* producing output *j* divided by the output of sector *a*:

```
p_GTAPLinV(i,j) $ (sum(a $ xFlag(rNat,a,j), p_results(rNat,"Q","out",a,"tot",%arg1%,"%version%"))  
    $ (not excludeIJ(i,j)))  
    = 1 $ sameas(i,j)  
    - sum(a $ xFlag(rNat,a,j),  
        p_results(rNat,"Q",i,a,"tot",%arg1%,"%version%"))  
    /sum(a $ xFlag(rNat,a,j), p_results(rNat,"Q","out",a,"tot",%arg1%,"%version%"));
```

Which we invert with an in-built GAMS utility:

```
--- invert  
  
execute_unload '%scrdir%/p_IA.gdx',i,p_GTAPLinV;  
execute '=invert.exe %scrdir%/p_IA.gdx i p_GTAPLinV %scrdir%/p_IAGInv.gdx p_GTAPLinV';  
execute_load '%scrdir%/p_IAGInv.gdx',p_GTAPLinV;  
  
--- store for further use  
  
p_mult(rNat,i,j) = p_GTAPLinV(i,j);
```

First, we define the final demand for each GTAP product in the simulated point and at the benchmark on a vector which will be later converted into primary product equivalents:

```

*
*   --- define final demand change in GTAP for each product
*
p_primEquDem(rNat,i,%arg1%) = sum(aa $ (not a(aa)), xa.1(rNat,i,aa,%arg2%));
p_primEquDem(rNat,i,"%t0%") = sum(aa $ (not a(aa)), xa.1(rNat,i,aa,"%t0%"));
    
```

Next, we define the list of GTAP products foodI which comprises commodities with a link to agri-food sectors and thus are related to FABIO:

```

foodI(i) $ sum(fabc_i(fabc,i),1) = YES;
foodI(i) $ sum(mapi("cmt",i),1) = YES;
foodI(i) $ sum(mapi("omt",i),1) = YES;
foodI(i) $ sum(mapi("vol",i),1) = YES;
foodI(i) $ sum(mapi("mil",i),1) = YES;
foodI(i) $ sum(mapi("pcr",i),1) = YES;
foodI(i) $ sum(mapi("sgr",i),1) = YES;
foodI(i) $ sum(mapi("ofd",i),1) = YES;
foodI(i) $ sum(mapi("b_t",i),1) = YES;
foodI(i) $ sum(mapi("trd",i),1) = YES;
    
```

For these products, we calculate the implicit demand for primary agricultural products (= primary product equivalents) by multiplying total final demand with the Leontief inverse:

```

p_primEquDem(rNat,i,%arg1%) = sum(FoodI, p_primEquDem(rNat,foodI,%arg1%) * p_mult(rNat,i,foodI));
p_primEquDem(rNat,i,"%t0%") = sum(FoodI, p_primEquDem(rNat,foodI,"%t0%") * p_mult(rNat,i,foodI));
    
```

We now need to map from the aggregate regions rNat and product i in constant USD in to the individual countries and products in FABIO in physical units:

```

p_results(fabr,"Q",fabC,"food","fabio",%arg1%,"%version%")
= sum( (fabr_r(fabr,rNat),fabc_i(fabc,i)) $ p_primEquDem(rNat,i,"%t0%"),
    --- final demand change GTAP
    p_primEquDem(rNat,i,%arg1%)/p_primEquDem(rNat,i,"%t0%")
    --- original final demand in FABIO (across all importers)
    * p_fabio("tot",fabc,fabr,"food")
);
    
```

In order to derive from there a shock in supply, we use FABIO Leontief inverse p\_LInv:

```

p_results(fabr,"Q",fabC,"Supp","fabio",%arg1%,"%version%") $ p_fabio(fabr,fabc,"prod","")
= p_fabio(fabr,fabc,"prod","")
+ sum( (fabcl,fabr1) $ p_fabInv(fabr,fabc,fabr1,fabcl),
    (p_results(fabr1,"Q",fabC1,"food","fabio",%arg1%,"%version%")
    - p_fabio(fabr,fabc,fabr1,"food")) * p_fabInv(fabr,fabc,fabr1,fabcl));
    
```

As FABIO comprises a matching vector of herd size and harvested areas for the processes, we can use the updated output quantities to assign changes in land use as well:

```

p_results(fabr,"Q",fabC,"Land","fabio",%arg1%,"%version%") $ p_fabio(fabr,fabc,"prod","")
= p_fabio(fabr,fabc,"Land","")
* p_results(fabr,"Q",fabC,"Supp","fabio",%arg1%,"%version%")/p_fabio(fabr,fabc,"prod","");
    
```

Finally, for reporting purposes, we also aggregate the results back to the regional aggregation used in CGEBox:

```

p_results(r,"Q",fabC,"Land","fabio",%arg1%,"%version%")
= sum( fabr_r(fabr,r), p_results(fabr,"Q",fabC,"Land","fabio",%arg1%,"%version%"));
    
```

Note that different extensions could be possible:

1. One could adjust the resulting changes in detailed output quantities to match simulated output quantity changes in GTAP.
2. The same could be possible for intermediate input and land use.
3. Instead of the corrections 1. and 2., the I/O coefficients in FABIO could be updated and the Leontief-inverse calculated from the updated MRIO.

#### 6.4.7 Nutrition indicators

The detailed data on food demand also open the door to calculate food supply indicators for calories, protein and fat which are also provided by the FAO in their food balance sheets (FBS). That firstly requires defining nutrient content per product. The FAO seems to use country specific values, we instead calculate world average ones using the global FBS results. The data for 2011 have been downloaded in CSV format from the FAO web page and converted in a GAMS readable table, they are stored under “fbs\_nut\_cont.gms”, an example for the relevant data is shown below:

"Wheat and products	". "Food supply quantity (kg/capita/yr)	" 65.27
"Wheat and products	". "Food supply (kcal/capita/day)	" 526
"Wheat and products	". "Protein supply quantity (g/capita/day)	" 15.88
"Wheat and products	". "Fat supply quantity (g/capita/day)	" 2.41

From there, the contents per unit of use are calculated:

```
p_nutContent(fbsProd,"cal") $ p_fbs(fbsProd,"food","data")
= p_fbs(fbsProd,"Food supply (kcal/capita/day)","data")
/ [ p_fbs(fbsProd,"food","data")/p_fbs("Population","Total Population","data")] * 365 / 1000;

p_nutContent(fbsProd,"prot") $ p_fbs(fbsProd,"food","data")
= p_fbs(fbsProd,"Protein supply quantity (g/capita/day)","data")
/ [ p_fbs(fbsProd,"food","data")/p_fbs("Population","Total Population","data")] * 365 / 1000;

p_nutContent(fbsProd,"fat") $ p_fbs(fbsProd,"food","data")
= p_fbs(fbsProd,"Fat supply quantity (g/capita/day)","data")
/ [ p_fbs(fbsProd,"food","data")/p_fbs("Population","Total Population","data")] * 365 / 1000;
```

Note that we recalculate the daily per capita demands to avoid potential rounding errors in the original data. For two items, there are naming / definitional differences:

```
p_nutContent("Sugar, Refined Equiv",nut) = p_nutContent("Sugar (Raw Equivalent)",nut) * 169489353/155930208;
p_nutContent("Mutton and Goat Meat",nut) = p_nutContent("Mutton & Goat Meat",nut);
```

There are some cases where the FBS seems to use a different accounting scheme which can lead to double counting. Sugar cane, to give an example, shows “Food” use in FABIO, but FABIO also report food use from “Sugar (refined)”. We therefore delete entries for a range of products:

```
p_nutContent("Sugar cane",nut) = 0;
p_nutContent("Sugar beet",nut) = 0;
p_nutContent("Beer",nut) = 0;
p_nutContent("Beverages, alcoholic",nut) = 0;
```

Alcoholic beverages are deleted to avoid that barley used for beer production is accounted twice.

The contents are multiplied with the “food” results, calculated as shown above:

```

p_results(fabr,"Q",fabC,nut,"fabio",%arg1%,"%version%")
= p_results(fabr,"Q",fabC,"food","fabio",%arg1%,"%version%")
  * p_nutContent(fabC,nut);

p_results(fabr,"Q","tot",nut,"fabio",%arg1%,"%version%")
= sum(fabC, p_results(fabr,"Q",fabC,nut,"fabio",%arg1%,"%version%"));

p_results(fabr,"Q","tot","food","fabio",%arg1%,"%version%")
= sum(fabC, p_results(fabr,"Q",fabC,"food","fabio",%arg1%,"%version%"));

```

## Result exploitation

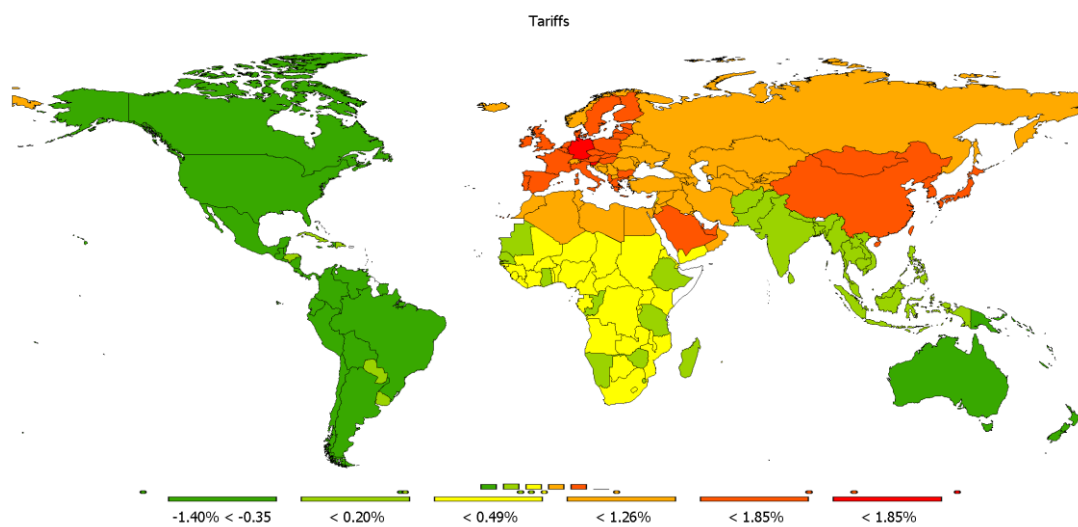
The three matrices with results on output, demand and land use / herds and the nutritional indicators

Land use	cts	32.61
FABIO	FABIO - Disaggregated results	
No table	FABIO - Disaggregated results, map	
essed Food	FABIO - Food and nutrient supply, per capita	
iles and Clothing	FABIO - Food and nutrient supply, per capita, map	
	FABIO - GTAP regional aggregates	

are accessible via the interface

FABIO - Disaggregated results [0]			
Items	Regions	Years	
Quantity	United States of America	shock	
	noShock		
	Harvested area or herd [ha or head]	Supply [ton]	Food demand [ton, primary equivalent]
tot			370489952.00
Rice (Milled Equivalent)	738042.00	5595325.00	2547301.00
Wheat and products	18760022.00	54418808.00	25049954.00
Barley and products	888857.19	3391683.00	3027635.00
Maize and products	34083712.00	313934784.00	7658345.50
Rye and products	96421.17	160681.00	85013.00

FABIO - Food and nutrient supply, per capita [0]				
Items	Regions	Years		
Quantity	United States of America	shock		
	Tariffs			
	Food demand [kg per caput and year]	KCal supply [per caput and day]	Protein supply [per caput and day]	Fat supply [ton]
tot	1192.12 -0.13%	4054.21 -0.07%	119.76 -0.26%	183.30 -0.09%
Rice (Milled Equivalent)	8.20 -0.04%	82.48 -0.04%	1.55 -0.04%	0.23 -0.04%
Wheat and products	80.67 -0.04%	650.18 -0.04%	19.63 -0.04%	2.98 -0.04%
Barley and products	9.75 -0.04%	69.10 -0.04%	1.97 -0.04%	0.30 -0.04%
Maize and products	24.66 -0.04%	203.39 -0.04%	4.92 -0.04%	1.74 -0.04%
Rye and products	0.27 -0.04%	1.95 -0.04%	0.06 -0.04%	0.01 -0.04%
Oats	4.38 -0.04%	22.11 -0.04%	0.96 -0.04%	0.44 -0.04%
Millet and products				
Sorghum and products	7.54 -0.04%	63.44 -0.04%	1.86 -0.04%	0.63 -0.04%



## 7. GUI and installation

### 7.1 Software requirements and access to the code

GAMS version 24.5 (released 2015) or higher is necessary to run the GAMS code, independently from using the code with or without the GUI. For GAMS versions before 24.8 (released 2016), it is not recommended to use the grid solve for the pre-solves or during data preparation. CONOPT4 is only available with newer GAMS version, if it is found, CONOPT should be used instead. The GUI requires a Java 8 run-time installation on the computer, for more detail see section 7.4 below, pages 390 ff.

The code and GUI are distributed in two ways. First, a zip archive can be downloaded from the GTAP web site (<https://www.dropbox.com/s/olcqbqprk6i9gth/CGEBOX.zip?dl=0>). The GAMS code reflects the status of the project in December 2018. In order to benefit from code improvements and bug-fixes, it is recommend to use the SVN option, see section 7.4 below, pages 390 ff.

### 7.2 Some comments for runGTAP and GEMPACK users

CGEBox is not thought or designed as a tool to easily switch from runGTAP or a model written in GEMPACK to a GAMS based CGE modelling platform. Even if CGEBox can replicate the structure of the standard GTAP model – however with equations written in levels and not in log-linearized form – the GAMS code uses different mnemonics and a different coding style, following Van der Mensbrugghe 2018<sup>117</sup>. That is not only due to differences between GAMS and GEMPACK, but also reflects a different style how to code a complex model and to allow for modularity. Please refer to the equation by equation documentation of the core of CGEBox in the next section “Basic model equations”. Equally, the GUI does not attempt to mimic the touch and feel of runGTAP or other GEMPACK utilities.

Users which so far have worked with runGTAP and GEMPACK can clearly expect to benefit from their knowledge about CGE modeling and the concept of GTAP, and their experience of having worked with a modelling language. With CGEBox, they will face however a new modeling platform which will most probably lead also to disappointments if things work differently than presumed, if usability is for some aspects assessed as lower and time must be spent with the documentation and code to get things going. It might be best to expect that things work differently and to have a look at the documentation first, instead of simply starting to work with the system along the lines used from runGTAP.

Note specifically that upper and lower case letters in symbol names have no meaning in the sense of indicating variables in levels or in percentage changes. Equally, note that GAMS makes a distinction between parameters which are always constants in a model and can never be changed by the solver and variables. While variables in GAMS can be technically turned into constants during a model solve by fixing them, parameters can never be turned in endogenous variables. This implies that symbols defined as parameters can never be subject to a closure swap.

Sparsity might warrant some further comments. In most GTAP data bases, all transactions are non-zero, but many are often very tiny. CGEBox offers a filter approach to remove such tiny transactions from the global SAM. Doing so reduces data base and model size, thus speeding up model generation

---

<sup>117</sup> Van der Mensbrugghe, D. (2018). The Standard GTAP Model in GAMS, Version 7. *Journal of Global Economic Analysis*, 3(1), 1-83



and solution as well as memory needs. However, allowing for sparsity in the data base requires that in most cases zero transactions are excluded in the GAMS code from data transformation and from entering the model. Otherwise, math trap errors such as divisions by zero are easily provoked. GAMS throws run time errors if that occurs and will as a default not solve any model after such an error occurs. That must be especially reflected when the user develops its own shock files.

### 7.3 Background

Working with a larger existing model such as GTAP can provoke serious entry barriers for newcomers, partly due to the need to learn specific model mnemonics underlying the code. That knowledge is necessary to analyze results if only model output directly produced by the modeling language is available, e.g. listings or proprietary binary result files. Equally, the sheer amount of symbols found in larger models (parameters, variables, equations, labels for regions, sectors, factors etc.) and the dimensionality of results can overwhelm beginners. This might shy them away from a fascinating research field, even if excellent model documentation is available, such as in the case of GTAP. Memorizing model mnemonics and partly model structure is rather independent from the modeling language used –beginners will always have to digest a larger amount of information to make the first steps with a larger economic simulation model.

It might therefore be useful to reduce entry barriers such that students might become enthusiastic about the model's capabilities and then take the more tedious steps to familiarize themselves with the model's code. Therefore, CGEBox comes with a Graphical User Interface (GUI) which complements the GAMS code. The reader should however note that the GUI linked to CGEBox follows a somewhat different approach compared to runGTAP, the GUI available for the GEMPACK version (see also Britz 2014<sup>118</sup> and Britz et al. 2015<sup>119</sup>) or, for instance, the GAMSIDE often used with economic models realized in GAMS. As a consequence, some functionalities found in runGTAP are not comprised in the GUI linked to the CGEBox and vice versa. The development of the GUI for CGEBox and its GAMS code for post-model processing feeding into the exploitation part of the GUI went partly in parallel with the development of the model codes to ensure seamless inter-operability. However, it is important to note that the GAMS code of CGEBox can be used completely independent from the GUI.

The GUI provides two major functionalities: firstly, it allows running the model while also liberating a user to carry out changes that are otherwise necessary in the GAMS code—such as to select a specific data base or shock file. Instead, controls on the GUI allow steering the model run, thereby avoiding code edits. That include configuration of the model by selecting modules and parameters files. Secondly, the GUI offers tools to view and analyze model results. Specifically, a post-model reporting part maps back the variables into a structure similar to a SAM, and additionally stores quantities, prices, values, tax rates and tax incomes. That structure can then be viewed in a flexible exploitation tool.

The post model code also provides some eventually useful aggregations, e.g. to total intermediate input, total factor input, total output, total demand and world totals and provides a welfare

---

<sup>118</sup> Britz, W. (2014): A New Graphical User Interface Generator for Economic Models and its Comparison to Existing Approaches, *German Journal for Agricultural Economics*, 63(4)

<sup>119</sup> Britz, W., Domínguez, I. P. and Gopalakrishnan, B. N. (2015): Analyzing results from agricultural large-scale Economic Simulation Model: Recent Progress and the Way Ahead, *German Journal for Agricultural Economics*, 107-119

decomposition. The user can also provide based on an aggregation definition file generated with GTAPAgg a second, more aggregated sectoral and/or regional aggregation which is used for post-model aggregation of these results.

The exploitation tools available with the GUI provide interesting features: they allow to view simultaneously results from different shocks without running additional programs, support maps and a wider range of graphics as well as basic statistics and thus might hence also be of interest to seasoned modelers. Furthermore, the GUI comprises some additional utilities: a tool similar to “EXAMINER” in GTAP which shows the linearized version of the model’s equations and allows inspecting parameters and variables found in these equations. Similar to “AnalyzeGE”, that tool also allows decomposing changes of the LHS of each equation to changes of the variables on the RHS. Another utility allows documenting the code in HTML pages. Finally, instead of using the GUI in interactive model, a batch mode allows to perform model runs. The batch reports these runs in a HTML page and allows parallel execution of different shocks or of the same shock in different model variants.

This section provides a hand-on guide on how to use the model in conjunction with the GUI. The latter is based on a GUI generator for GAMS projects ([http://www.ilr.uni-bonn.de/agpo/staff/britz/ggig\\_e.htm](http://www.ilr.uni-bonn.de/agpo/staff/britz/ggig_e.htm), Britz 2010<sup>120</sup>), which is also used for some larger projects such as CAPRI (<http://www.capri-model.org/dokuwiki/doku.php?id=start>), but also a range of other modeling systems such as the Policy Evaluation Model (PEM) by the OECD; a dynamic single farm model, an Agent Based Model of structural change in agriculture or a recursive-dynamic Hydro-economic river basin model. For those interested in a discussion about interfaces used with economic simulation models, see for example <https://www.gtap.agecon.purdue.edu/resources/download/5864.pdf> (Pérez Domínguez et al. 2012) or Britz et al. 2015. The GUI generator is flexible enough to also host extensions as shown by the implementations of GTAP-AEZ, GTAP-AGR and GTAP-E.

A first version of the GUI was developed for a class at Bonn University, Institute for Food and Resource Economics, in the winter term 2013/2014, building on Tom Rutherford’s GTAPinGAMS version. The students had already some knowledge of GAMS and CGEs, and wanted to use a global CGE in a project. A year later, again a class of students wanted to use a global CGE, which led to further improvements such as flexible aggregation of regions in the mapping viewer or the introduction of sensitivity analysis. In 2014, when the GTAP Center decided to release a full-fledged version of the standard GTAP model in GAMS, it was decided to adjust the interface to operate with that version. The further work on GTAP led to improvements to the GUI generator, also to the benefit of other modeling systems applying the GUI. The GUI itself is distributed for free. However, the underlying Java code is currently not open source.

## 7.4 SVN checkout

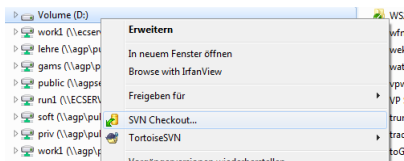
The preferred option to access code and GUI is to benefit from SVN, a software version system on which the model codes are hosted on the repository <https://svn1.agp.uni-bonn.de/svn/cgebox/>. The SVN option is especially recommended for users who both want to use the GUI and foresee also own changes to the code. In the SVN repository, the development of CGEBox continues. We expect that over the next years additional modules will be integrated into the code and clearly, bugs corrected and improvements implemented once the code is more widespread used. Using SVN allows updating more easily to bug fixed or new extensions.

---

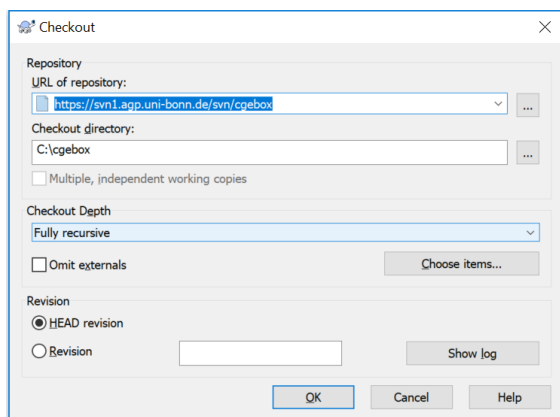
<sup>120</sup> Britz, W. (2010): GAMS Graphical Interface Generator Programming guide. Institute for Food and Resource Economics, University Bonn. In: [http://www.ilr.uni-bonn.de/agpo/staff/britz/GGIG\\_programming\\_guide.pdf](http://www.ilr.uni-bonn.de/agpo/staff/britz/GGIG_programming_guide.pdf)

Therefore, as a first step, we recommend installing a SVN client such as TortoiseSVN (<http://tortoisetsvn.net/index.de.html>, freeware) on your machine. Afterwards, make a checkout of the repository <https://svn1.agp.uni-bonn.de/svn/cgebox/> (i.e. download the code from the server to your local machine), using the user id “cgebox” and the pass word “cgebox”. Using SVN will allow you to keep your local copy automatically updated to code improvements, to check which changes you introduced in the code and to switch easily back to original versions of all or specific files if deemed useful, if necessary on a line by line basis.

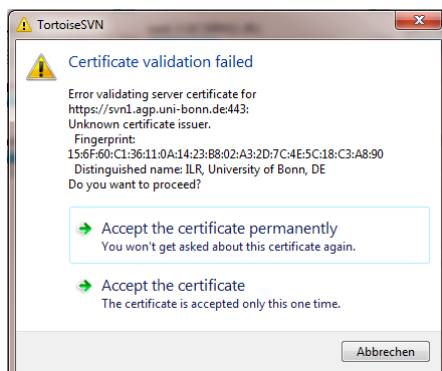
The basic steps for using SVN to install CGEBox based on TortoiseSVN are detailed below. First, navigate in the windows explorer to the disk / directory where you want to install GTAP, and open the context menu with a right mouse click:



Chose, “SVN Checkout ...” and copy the URL “<https://svn1.agp.uni-bonn.de/svn/cgebox/>” in the first field as shown below and press “OK”.

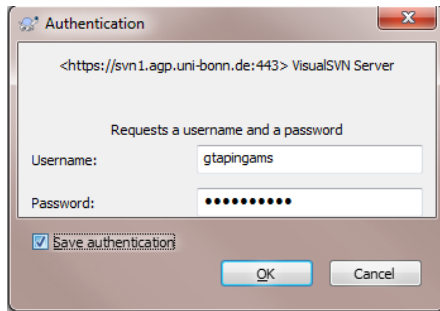


In case you get a warning such as:

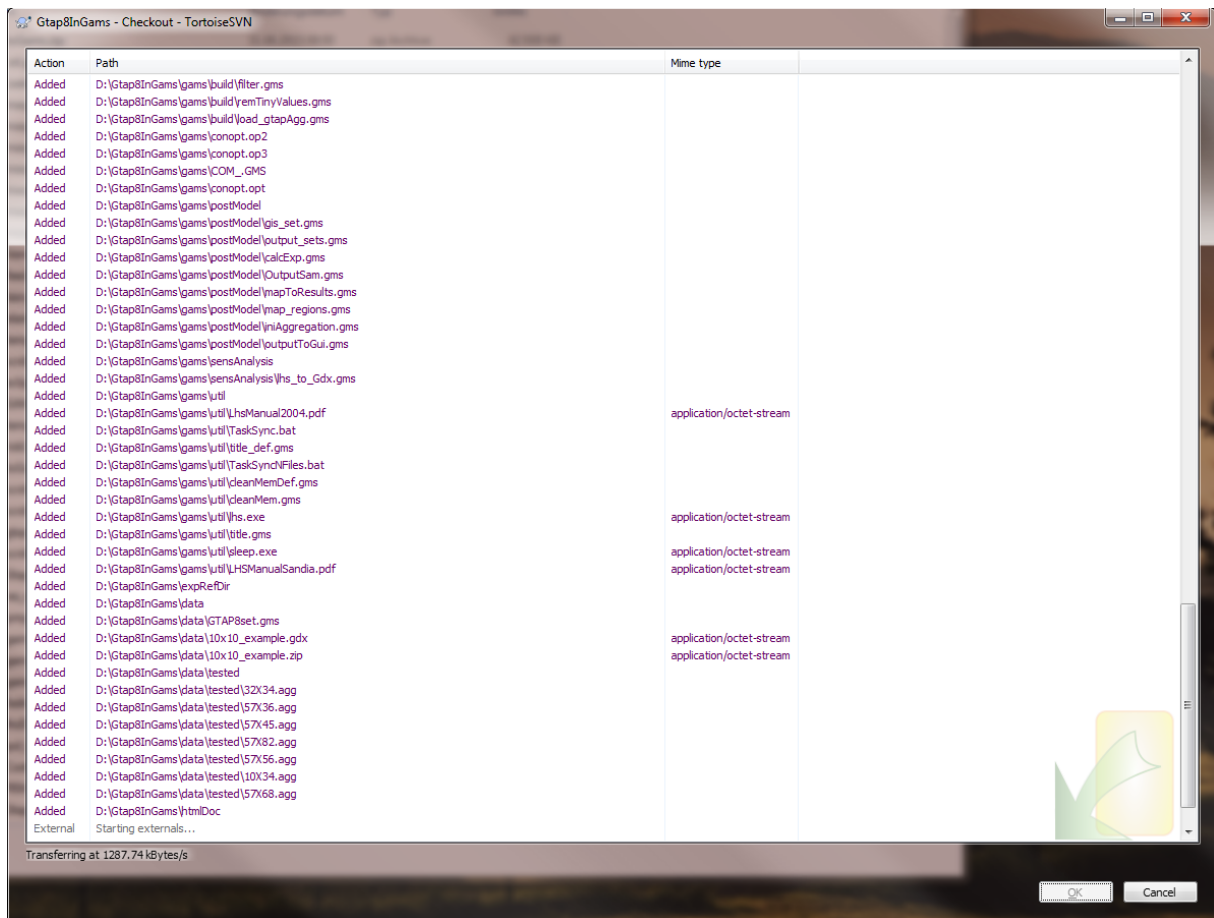


choose “Accept the certificate permanently”. In the next dialogue, enter the username and password “cgebox” ( not the one shown below), check “Save authentication” in the checkbox in the lower part of the dialog and press “OK”.

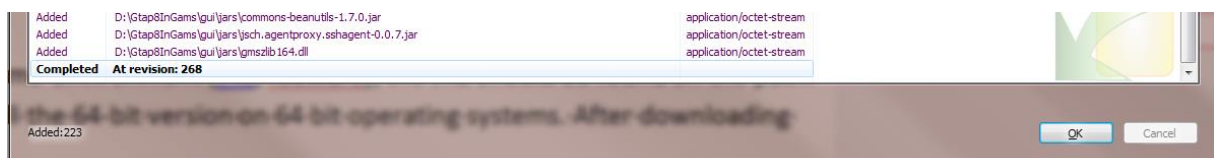
## SVN checkout



The SVN client will next download the newest release of all files to your computer which might take some time:



And should end with a “Completed At revision ..” message.



After you press “OK”, you should find the new folder on your computer:

✓ CGEBOX

12/12/2018 2:47 PM

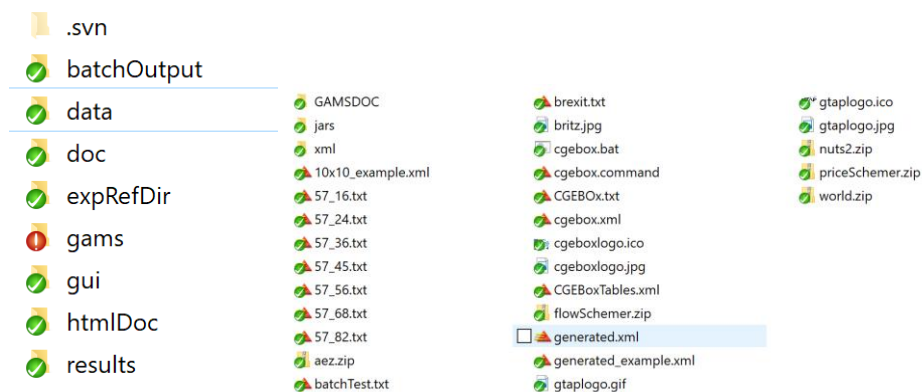
Dateiordner

The green mark indicates that there are no local modifications to files found in the folder, i.e. that the files on your disk are identical to those just downloaded from the repository. If you introduce changes in any file, a different symbol will be shown.

For further information on how to benefit from TortoiseSVN for your daily work with CGEBox for other projects, please refer to its documentation available on its web-site. We would like to mention here only that you have always two versions of each file on your computer: (1) the so-called local working copy – these are the files in your normal directory with which you can work as usual and (2) a second version in a hidden data base which reflects the latest download from the repository. That allows the system to e.g. find out which files changed and to highlight changes line by line. It is important to note that you will never lose edits on your local computing when you perform “updates”, i.e. download newer versions from the central repository.

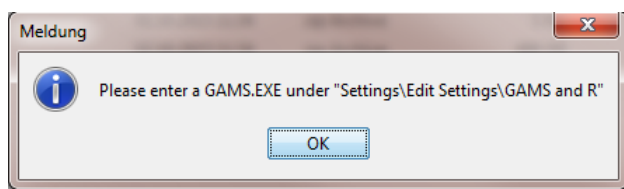
## 7.5 Starting the GUI

The GUI requires a Java 8 run time environment ([JRE](#), feeware); the JRE should be found on the path. Please make sure that you install the 64-bit version of the JRE on 64 bit operating systems. After downloading the GAMS sources and other files via SVN or having unzipped the downloaded folder, navigate to the GUI folder:



and double-click in the GUI folder on the file “cgebox.bat”; it should open the interface<sup>121</sup>. You might want to put a shortcut to that batch file on your desktop to easily startCGEBox.

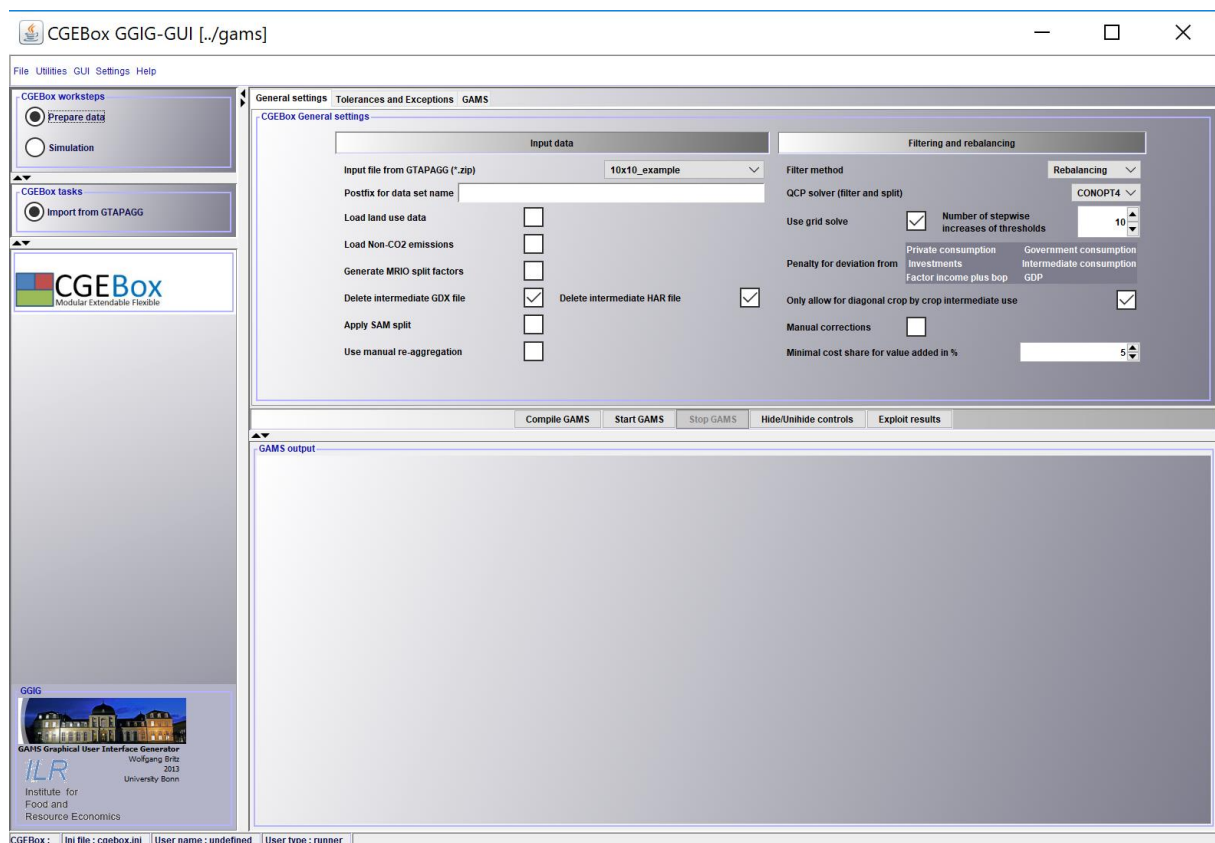
You might first see a message which tells you that a GAMS executable is not yet registered:



You can first ignore that warning for now. After pressing “OK”, you should see a program opened as below:

---

<sup>121</sup> If that does not work, open a command prompt, start from there the “cgebox.bat” batch file and analyze the error. Most probably, JAVA is not found. In that case, either put it on the path or change the batch file such that it calls Java from the directory where it is installed.



**Figure: Interface at first start**

**Note:** The layout shown in screen shots might not be completely identical to the distribution version

The left hand side allows selecting work steps and the tasks comprised in them; the right hand side carries controls to steer these tasks. The bar in the middle comprises buttons to start the GAMS code of a task and exploit the resulting output.

Typical reasons why the **GUI does not open** and possible remedies are described briefly below:

- **Java is not installed** or not registered on the PATH. In order to test that, open a DOS prompt and type Java. If you receive a message that the command is not found, (re-)install Java and make sure that Java also updates the PATH during the installation.
- A **wrong Java Runtime Environment** is installed. Please first check if Java version 8 is installed and not some older version (that check is not only useful for the GUI of CGEBox, but also generally for security reasons if one uses Java plugins in web pages). On 64 bit Windows operation systems, a 64 bit version of Java should be used. You can test that again in a DOS prompt by typing “java -version”. The response should look similar as shown below (note the string “64-Bit”). Unfortunately, many 64 bit Windows machines use a 32 bit browser, and if you install Java from such as browser, the 32 bit version is downloaded. The link [JRE](#) points both to a 32 and 64 bit version, make sure you select the 64 bit version on a computer with a 64 bit OS. Installing a 64 bit version on a machine where the 32 bit version is already installed does no harm and makes the 64 bit version the default.

```
java version "1.8.0_25"
Java(TM) SE Runtime Environment (build 1.8.0_25-b18)
Java HotSpot(TM) 64-Bit Server VM (build 25.25-b02, mixed mode)
```

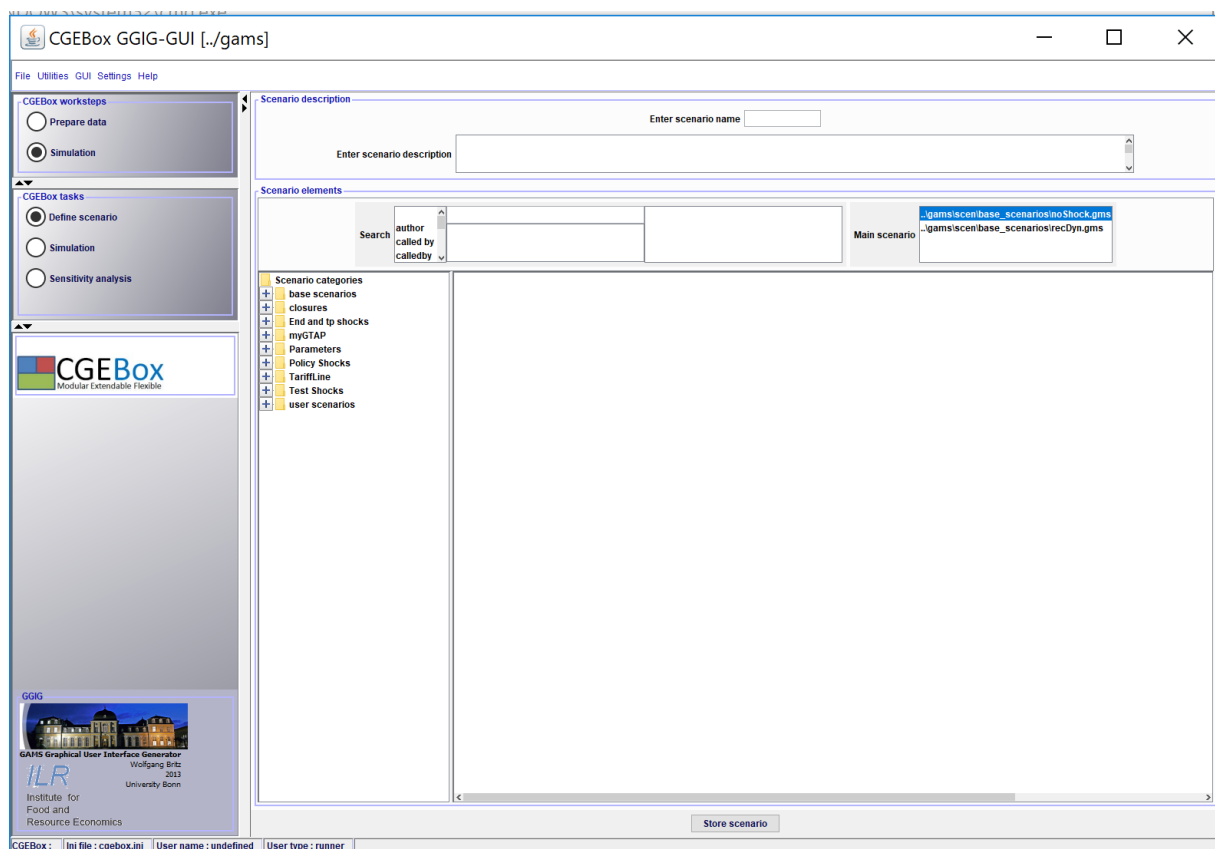
- If that still does not help, open a DOS prompt, navigate to the GUI folder and type “cgebox.bat” – possible errors will be shown in the window. In case you cannot find help in your team with regard to these error messages, contact the author.

General information on how to work with the GUI can be found in the GGIG user guide at [http://www.ilr.uni-bonn.de/agpo/staff/britz/GGIG\\_user\\_Guide.pdf](http://www.ilr.uni-bonn.de/agpo/staff/britz/GGIG_user_Guide.pdf). That pdf document can also found in the folder “GUI\jars” in your installation. Please note that the hints given below in the document on using the interface are only thought as a first introduction, refer to the GGIG user guide for any further details or use the inbuilt help system by pressing F1.

We would finally like to note that we did our best to test the combination of the CGEBox code and the GUI, but that improvements are certainly possible.

## 7.6 First steps

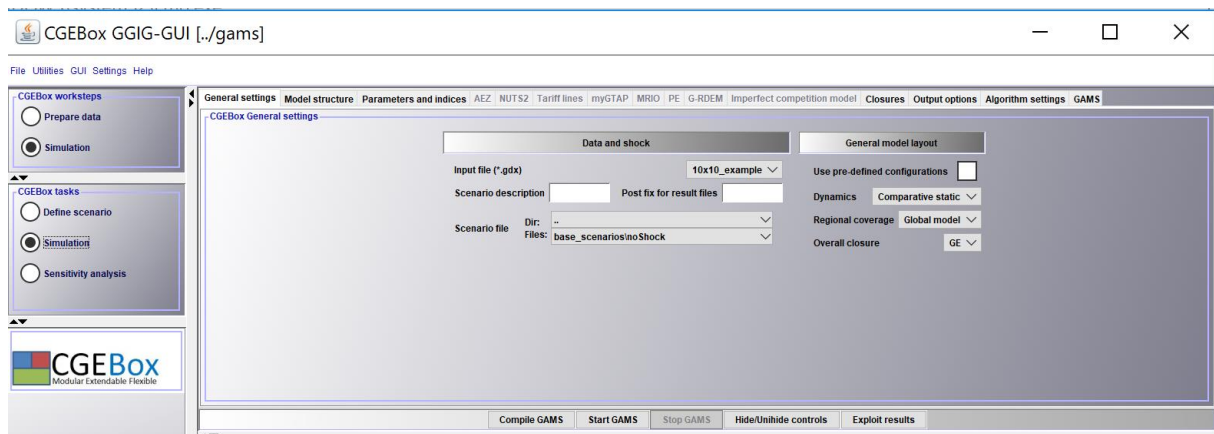
The installation comprises a set of results for testing the interface without the need of actually running GAMS code. You can now already look at these results by selecting on the left hand side the work step “Simulation”:



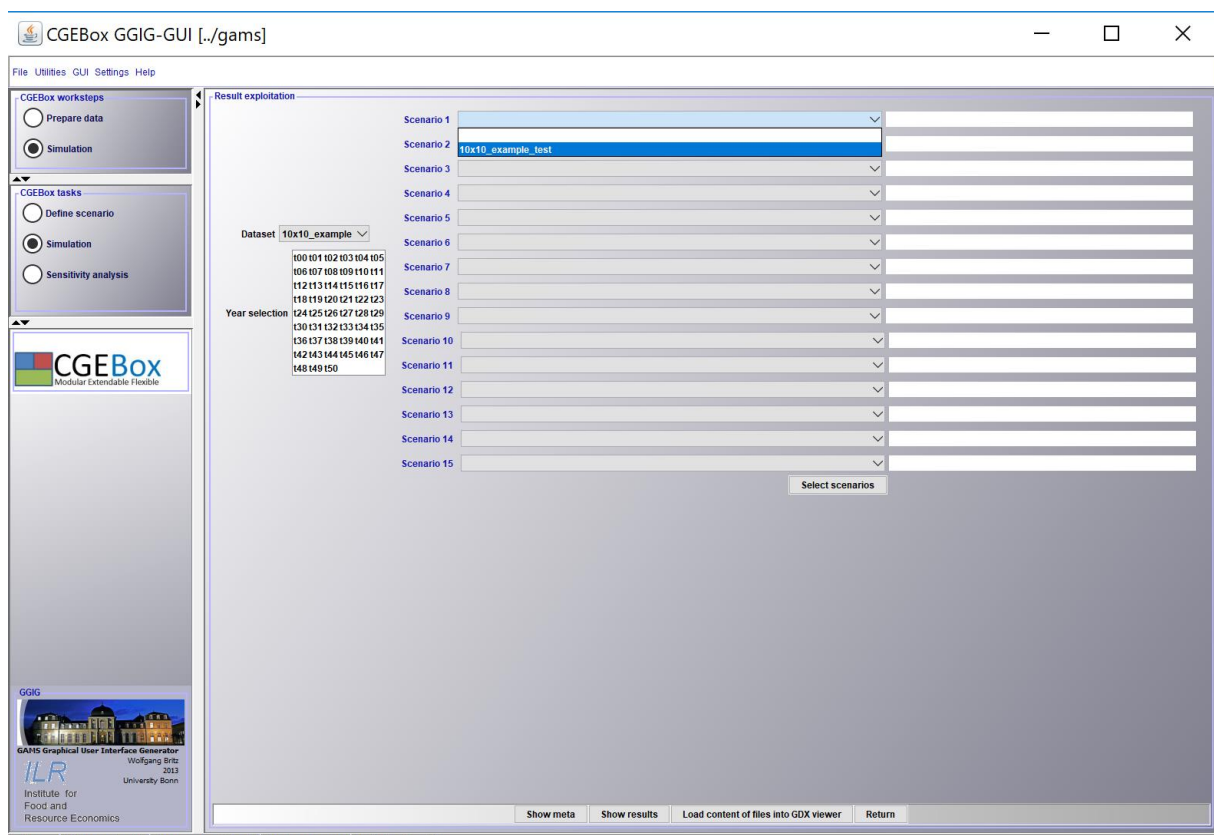
You will see the interface for the “Scenario editor” which is discussed later. We skip that functionality here and move directly to the task “Simulation”:



## First steps



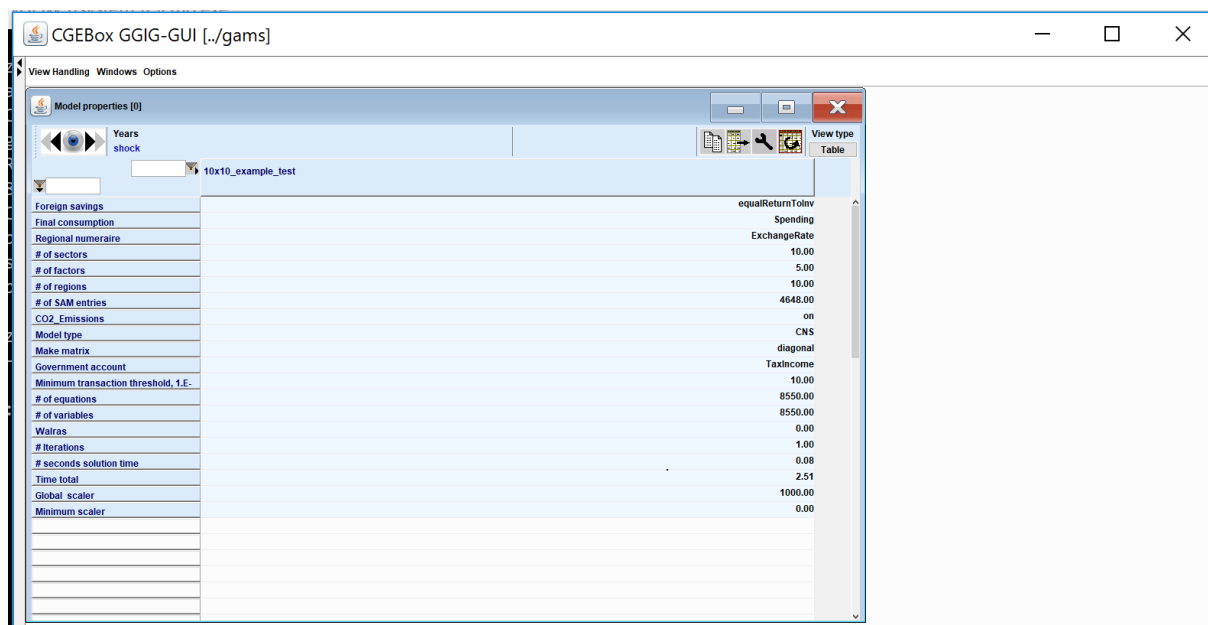
and next pressing the “Exploit results” button. The layout will change as seen below. Please click now once in the “Dataset” selection (otherwise you will get a warning message later):




**Figure: Scenario selection for exploitation**

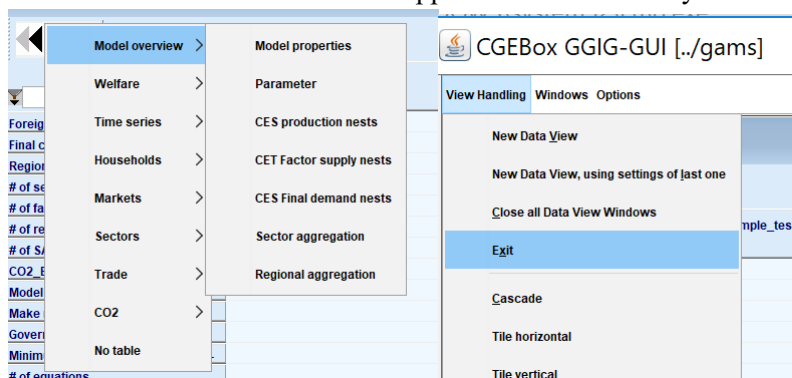
Select the scenario “10x\_10\_example\_test” and press “show results”, you should see a table similar as the one shown below:

Note: In the year selection panel, nothing should be selected for comparative-static runs.



(Once you have produced your own results, you can find the data bases you used and the scenarios generated in the drop-down boxes, and you might combine different scenarios for comparison.)

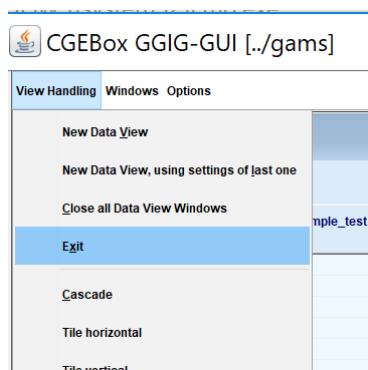
The “” view button in the upper left corner lets you select different views on your results:

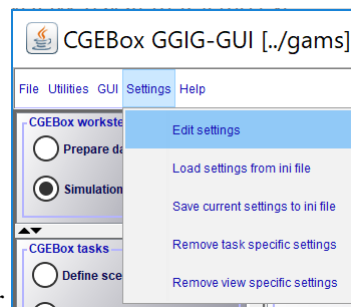


How to produce your own results is discussed in the following. Information on the different views is given in a section below.

## 7.7 Making the interface working with GAMS

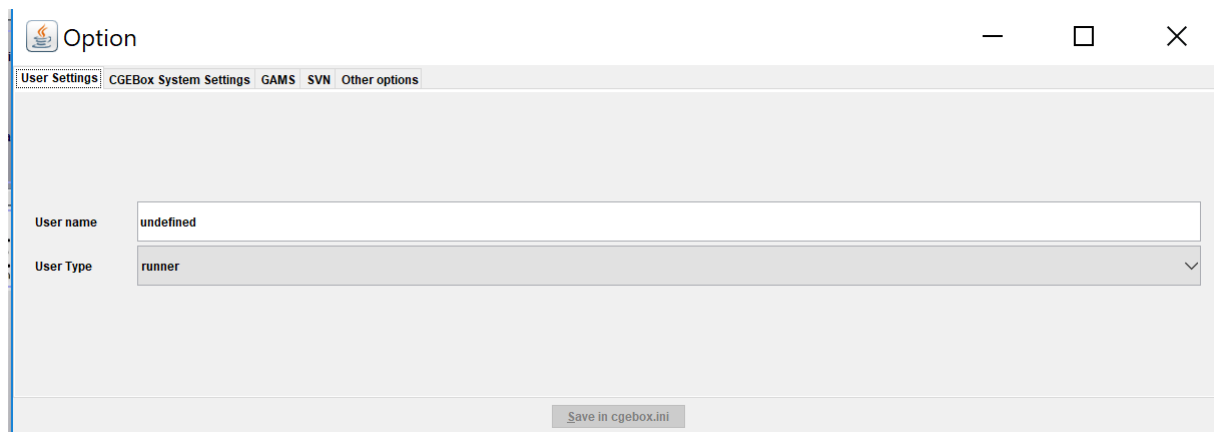
In order to use the GUI to generate data sets or for simulation runs, you have first to register a GAMS executable. In order to do so, first leave the result viewer as shown below



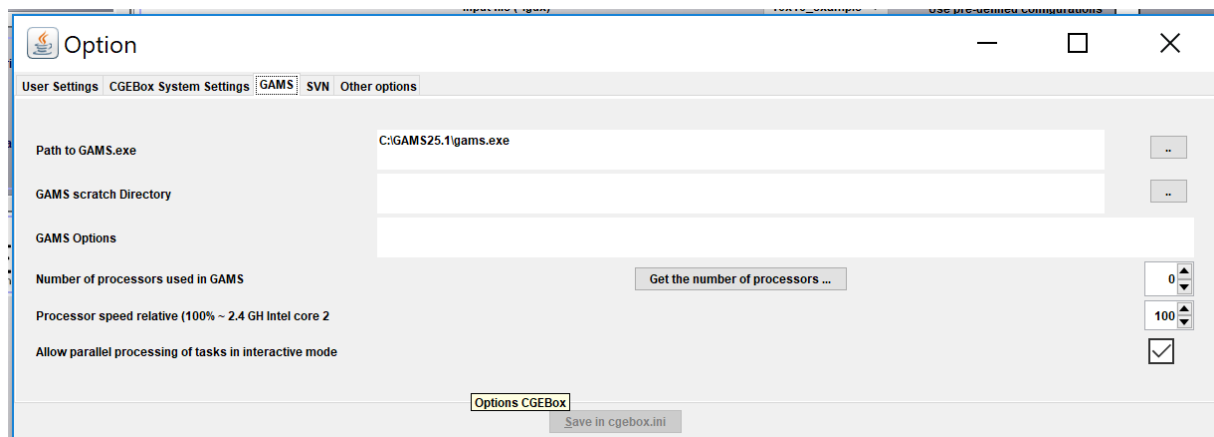


And open the setting dialogue from the menu bar

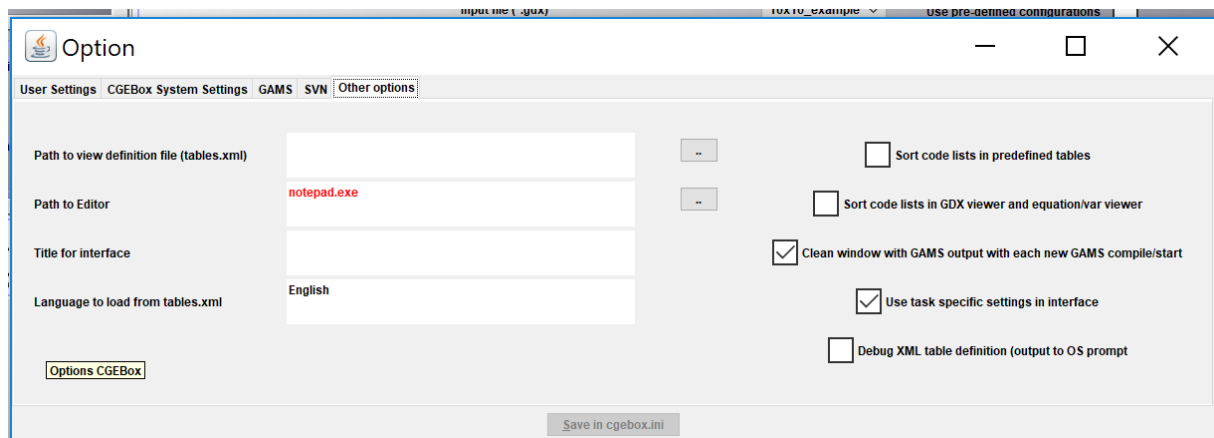
First, you should enter your name:



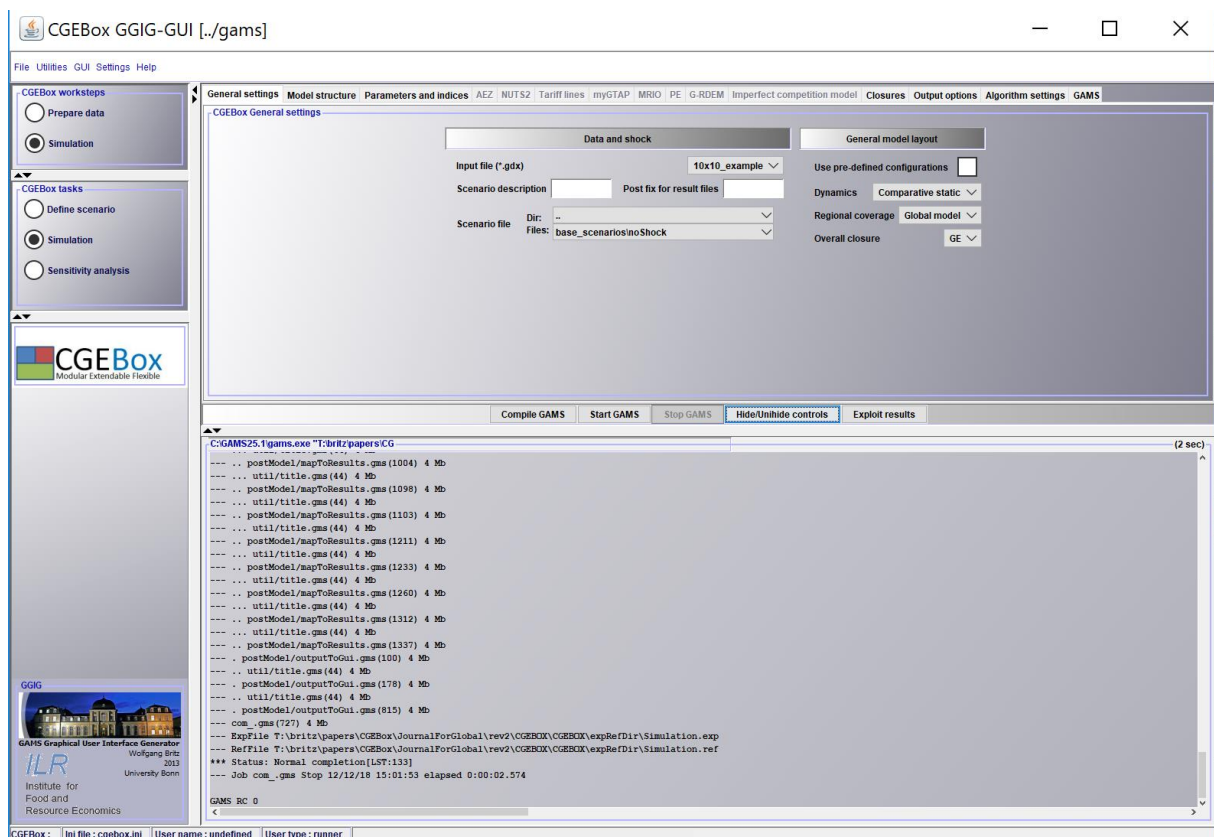
Next, click on the tab “GAMS” and either type in text field next to “Path to GAMS.EXE” which “GAMS.EXE” you would like to use. Alternatively, you can use the button to the right of the field to navigate to the directory where “GAMS.EXE” is found via a file selection dialogue. Please do not only enter a directory, but the full file name as shown below (and choose GAMS.EXE, not the user interface of GAMS, GAMSIDE.EXE). Depending how you install GAMS on your machine, the correct GAMS executable might already be entered by the Java code.



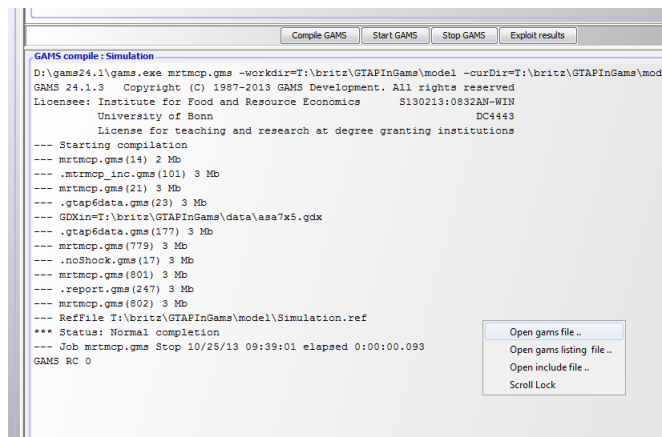
If you are using regularly a text editor, you can register it under “Other options”



In order to check if it worked, press the “Compile GAMS” button. You should see now the compile output from GAMS in the window below the button.



If you have registered an editor, you can also check its proper functioning: a right mouse click in the window with the output should open up a pop-up menu:



That pop-up menu offers you the options to open in your favorite editor the GAMS file, the listing produced by GAMS and the include file generated by the interface.

Notes:

- The use of the GTAP-AGR or GTAP-E module works best with a data base with a full disaggregation of primary agriculture respectively detail for the energy sector.
- Some pre-defined configuration such as “MIRAGE” comprise also list of sector which assume that you use a data base with full sector resolution.
- You cannot use the global bank mechanism (Global equal returns to capital) if capital is declared immobile. There are further combinations of options where warnings are raised.
- A license either for PATH or CONOPT is required to solve the model.

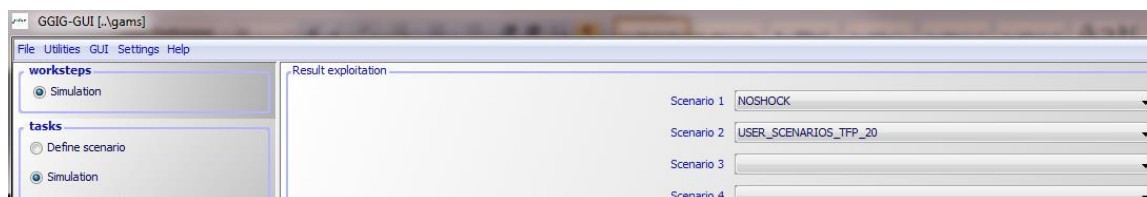
## 7.8 Inspecting the results

Before we turn to the exploitations tool of the GUI, the reader should be reminded that CGEBox allows for a wider range of possibilities to look at results:

1. As with any GAMS model, solprint=on will produce a listing of the model solution showing variables and equations, see the GAMS documentation for details. That option can be switched on from the interface as discussed above.
2. If additionally, LIMROW and LIMCOL are non-zero, the individual equations will be shown with their Jacobian entries at current variable levels. The numbers entered under LIMROW and LIMCOL will limit the output of individual instances shown and can be set by the GUI. For more information on how to interpret the resulting output, please refer to the GAMS manual.
3. If the output option GDX is switched on, all GAMS symbols (sets, parameters, variables) will be outputted to a GDX container and can be analyzed in a GDX viewer, for instance, the one delivered with the GAMS IDE or the GDX viewer built into the GUI.
4. The model's results can be mapped back into a SAM, and the SAM will be stored in the GDX.
5. The information from 1. -4. can also be inspected with the “Equation and Variable viewer” discussed in another section, which also provides a decomposition of each equation similar to “analyzeGE”.
6. Finally, the exploitation tools can be used, which also cover the SAM, as discussed in the following.

Which option works best depends on the use case and user preferences. Model listings with “limcol” and “limrow” switched on are extremely useful during model development and debugging, whereas the exploitation tools are probably the best approach to systematically analyze results from a shock. Individual equation decomposition can complement both approaches. We can only recommend trying out all approaches at least once to find out which one fits best one's preferences under specific use scenarios.

Inspecting results with GUI is nothing new: You have done that already if you followed the short introduction ... so press on the “Exploit results” button and next on “Show results” again. Use the “NoShock” and your own scenario:




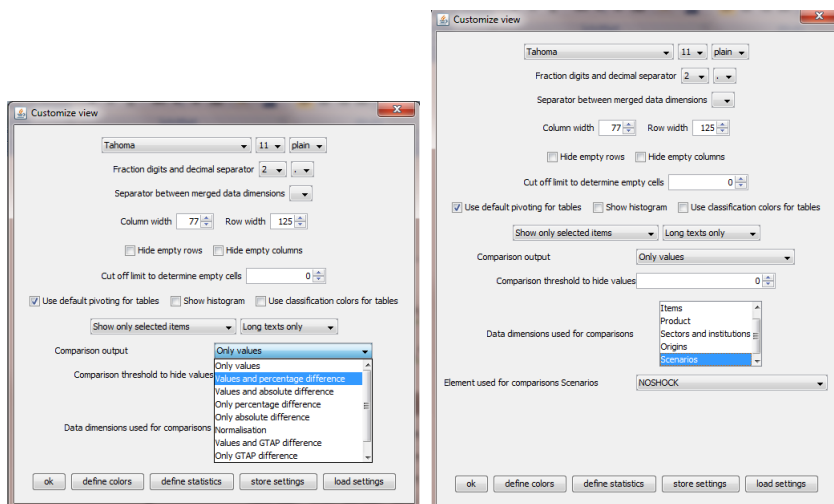
And now, you can see the counterfactual and the simulation side by side in the tables:

## Inspecting the results

The screenshot shows the GGIG-GUI interface with a table of economic indicators. The table has columns for 'Regions' (World), 'Total or household groups' (Total), and 'USER\_SCENARIOS\_TFP\_20'. The 'Customize Table' option is highlighted in the context menu.

Indicator	World	TFP_20
Total utility [Index]	1.000	1.218
Utility of private household [Index]	1.000	1.215
Utility of government [Index]	1.000	1.225
Utility of saving [Index]	1.000	1.220
Total income [Mill \$]	55518.188	56632.293
Savings price [Index]	7.000	5.801
Factor income [Mill \$]	39484.758	40464.602
Current account balance [Mill \$]		0.000
Tax income [Mill \$]	16033.430	16167.688
Population size [Mill heads]	6620.293	6620.293

You might now want to add relative differences: click with the mouse in the table or use the “Customize” (  ) button in order to customize your view (see screenshot below). Choose “Values and percentage difference” as the “Comparison output”, select “Scenarios” as the “Data dimensions used for comparisons” and make sure that the “NOSHOCK” scenario is used as the comparison element:



After closing the dialogue, you find now relative difference in small numbers beyond the simulated results in each cell of the table.

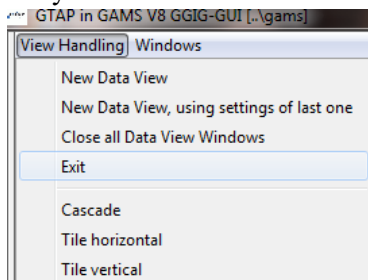


Indicator	Value	Percentage diff. to Scenarios NOSHOCK
Total utility [Index]	1.000	1.218
Utility of private household [Index]	1.000	21.753%
Utility of government [Index]	1.000	1.215
Utility of saving [Index]	1.000	21.467%
Total income [Mil \$]	55518.188	1.225
Consumer price [Index]		22.499%
Government price [Index]		1.220
Savings price [Index]	7.000	21.981%
Factor income [Mil \$]	39484.758	56632.293
Current account balance [Mil \$]		2.007%
Tax income [Mil \$]	16033.430	
Population size [Mil heads]	6620.293	0.000%

(Note: as expected, total utility increases, but prices drop ...).

There are many other functionalities found in the exploitation tools, the GGIG user manual discusses these options: [http://www.ilr.uni-bonn.de/agpo/staff/britz/GGIG\\_user\\_Guide.pdf](http://www.ilr.uni-bonn.de/agpo/staff/britz/GGIG_user_Guide.pdf).

If you want to leave the viewer again, use “Exit” from “View Handling” menu:



### 7.8.1 Notes for GEMPACK users or users familiar with other CGE models

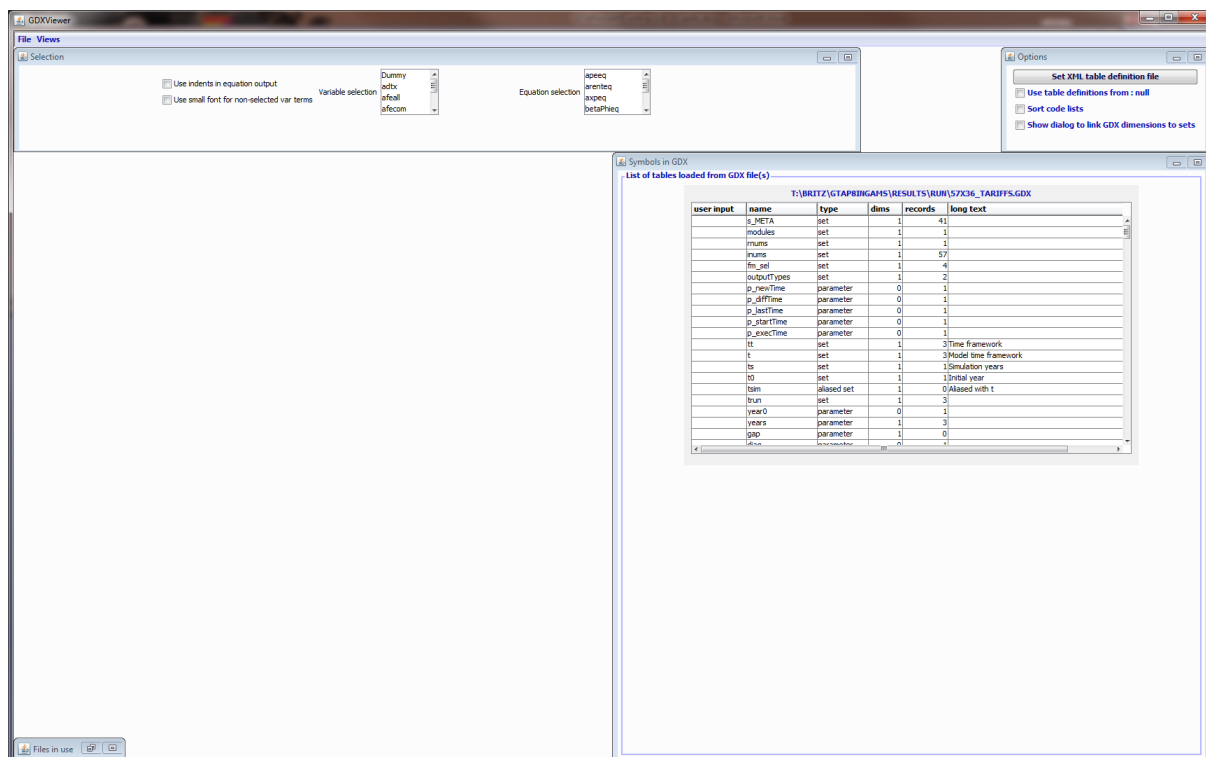
- The view “Model overview, SAM” shows a SAM generated from the model results (including row and column sums and potential imbalances) if the option “Store SAM in GDX” is switched on.
- Many tables comprise additional information on aggregates on top of the other rows. Only in some cases is what is called “total” a simple sum of the rows below. Thus, the information should generally not be confused with the “sum” information shown in “viewwhar”.
- There are only a few cases with a 1:1 relation between a variable in the model and numbers shown in table; typically the results shown in the different views stem from some post-model processing (e.g. multiplying a quantity index with a price).
- Instead of using the button, “Show results” which presents the different views discussed below, the “Load content of files in GDX viewer” allows to inspect all symbols used in the code, including all variables and the generated SAM (see next section). The GDX files are stored under “results\run” and can be naturally be opened with another GDX viewer such as the one comprised in the GAMSIDE.
- If the user want to inspect variables similar to “AnalyzeGE”, including decomposing equations, she can use the “Equation and variable viewer” (see below).
- Note again that all model equations are written in levels and not in relative changes.

## 7.8.2 Relative difference for all variables as in runGTAP

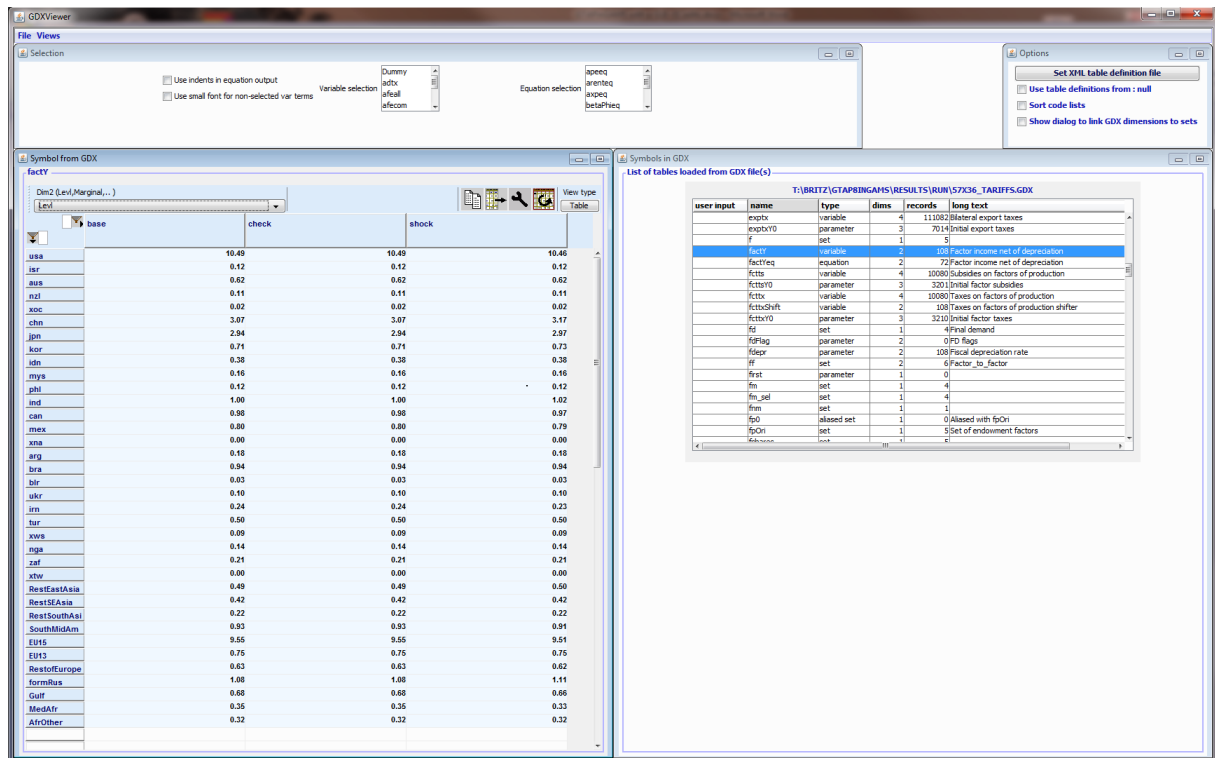
GEMPACK users typically compare simulated against benchmark levels for model variables. That is also possible with the interface. In order to do so, select the file with the result and press the “Load content of files into GDX viewer” button:




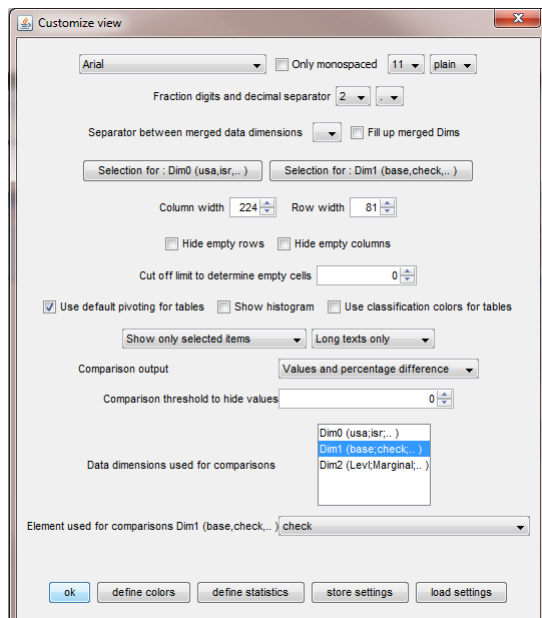
That will open a new window as shown below:



The window “List of table loaded from GDX file(s)” shows the status of all GAMS symbols (variable, equations, sets, parameters) after the solution of the shock. Double-click on any symbol you are interested in, e.g. the factor income “facy” as shown in the screen shot below. That will open a window where the results are shown, for the base case (= as initialized from the data base), the benchmark test (= check) and the shock.



In order to show percentage differences against the benchmark, open the option dialogue by pressing “”. Select e.g. “Values and percentage differences” under comparison output, chose the data dimension which end with “(check; ... ) and the item “check”:




The window with the result will now also comprises the information on the relative change as shown below:

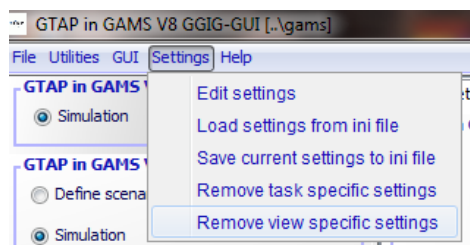
	base	check	shock
usa	10.49 0.00%	10.49	10.46 -0.28%
isr	0.12 0.00%	0.12	0.12 -0.58%
aus	0.62 0.00%	0.62	0.62 -0.75%
nzl	0.11 0.00%	0.11	0.11 1.04%
xoc	0.02 0.00%	0.02	0.02 -0.19%
chn	3.07 0.00%	3.07	3.17 3.18%
jpn	2.94 0.00%	2.94	2.97 1.29%
kor	0.71 0.00%	0.71	0.73 2.62%
idn	0.38 0.00%	0.38	0.38 0.10%
mys	0.16 0.00%	0.16	0.16 0.65%
phl	0.12 0.00%	0.12	0.12 0.20%

## 7.8.3 Trouble shooting with the viewer

If you are not sure what controls are for, try pressing the F1 button when the mouse is hovering over the control. In most cases, the PDF user guide will open at a page offering information.

Especially when working in the beginning with the viewer, one often ends up with a table showing no longer any data, an awkward selection, an unsuitable pivot etc.. The following strategies can be used to overcome such problems:


1.  use the “close” button to remove the view, and use “View Handling”, “New Data View” to open a fresh one.
2. Leave the viewer (“View Handling”, “Exit”), and click on “Remove view specific settings” under “Settings” in the menu bar. That will set the viewer back to the original defaults.



3. If that does also not help, try to close and re-open the application.
4. If the viewer still shows curious things, it is most probably some programming error ... contact the author.

### 7.8.4 Overview on existing views for exploitation

**Model overview/Model properties:** reports some basic attributes of the model instance as seen below, such as the modules switched one, closures used or the size of the model:



Property	Value
Foreign savings	equalReturnToInv
CO2_Emissions	on
# of sectors	10.00
# of factors	5.00
# of regions	10.00
# of SAM entries	4648.00
Model type	CNS
Make matrix	diagonal
Global scaler	1000.00
# of equations	12380.00
# of variables	12380.00
# Iterations	15.00
# seconds solution time	0.53
Time total	3.43

There are also two tables which report the aggregation of the data base under **Model overview/regional aggregation** and **Model overview/sectoral aggregation**:



Region	Aggregation
Australia, New Zealand (Australia, New Zealand, Rest of Oceania, )	
East Asia (China, Hong Kong, Japan, Korea, Mongolia, Taiwan, Rest of East Asia, )	
Southeast Asia (Cambodia, Indonesia, Lao People's Democratic Republ, Malaysia, Philippines, Singapore, Thailand, Viet Nam, Rest of Southeast Asia, )	
South Asia (Bangladesh, India, Nepal, Pakistan, Sri Lanka, Rest of South Asia, )	
North America (Canada, United States of America, Mexico, Rest of North America, )	
Latin America (Argentina, Bolivia, Brazil, Chile, Colombia, Ecuador, Paraguay, Peru, Uruguay, Venezuela, Rest of South America, Costa Rica, Guatemala, Honduras, Nicaragua, Panama, El Salvador, Rest of Central America, Caribbean, )	
European Union 25 (Austria, Belgium, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Poland, Portugal, Slovakia, Slovenia, Spain, Sweden, United Kingdom, )	
Middle East and North Africa (Rest of Western Asia, Egypt, Morocco, Tunisia, Rest of North Africa, )	
Sub-Saharan Africa (Benin, Burkina Faso, Cameroon, Cote d'Ivoire, Ghana, Guinea, Nigeria, Senegal, Togo, Rest of Western Africa, Central Africa, South Central Africa, Ethiopia, Kenya, Madagascar, Malawi, Mauritius, Mozambique, Rwanda, Tanzania, Uganda, Zambia, Zimbabwe, Rest of Eastern Africa, Botswana, Namibia, South Africa, Rest of South African Customs, )	
Rest of World (Oman, Israel, Switzerland, Norway, Rest of EFTA, Albania, Bulgaria, Belarus, Croatia, Romania, Russian Federation, Ukraine, Rest of Eastern Europe, Rest of Europe, Kazakhstan, Kyrgyzstan, Rest of Former Soviet Union, Armenia, Azerbaijan, Georgia, Bahrain, Iran Islamic Republic of, Kuwait, Qatar, Saudi Arabia, Turkey, United Arab Emirates, Rest of the World, )	

## Inspecting the results

Sector aggregation [0]

Scenarios  
TEST

Grains and Crops (Paddy rice, Wheat, Cereal grains nec, Vegetables, fruit, nuts, Oil seeds, Sugar cane, sugar beet, Plant-based fibers, Crops nec, Processed rice, )	
Livestock and Meat Products (Cattle,sheep,goats,horses, Animal products nec, Raw milk, Wool, silk-worm cocoons, Meat: cattle,sheep,goats,horse, Meat products nec, )	
Mining and Extraction (Minerals nec, Forestry, Fishing, Coal, Oil, Gas, )	
Processed Food (Vegetable oils and fats, Dairy products, Sugar, Food products nec, Beverages and tobacco products, )	
Textiles and Clothing (Textiles, Wearing apparel, )	
Light Manufacturing (Leather products, Wood products, Paper products, publishing, Metal products, Motor vehicles and parts, Transport equipment nec, Manufactures nec, )	
Heavy Manufacturing (Petroleum, coal products, Chemical,rubber,plastic prods, Mineral products nec, Ferrous metals, Metals nec, Electronic equipment, Machinery and equipment nec, )	
Utilities and Construction (Electricity, Gas manufacture, distribution, Water, Construction, )	
Transport and Communication (Trade, Transport nec, Sea transport, Air transport, Communication, )	
Other Services (Insurance, Financial services nec, Business services nec, Recreation and other services, PubAdmin/Defence/Health/Educat, Dwellings, )	

**Model Overview/SAM** reports the results in a SAM, including row and columns sum and potential imbalances. Due to rounding errors and feasibility tolerances of the solvers, small imbalances in relative terms are possible and not of concern:

SAM [1]

Regions: World  
Origins: Total  
Years: shock

10x10\_Tariffs

Grain	Oceania	EastAsia	SEAsia	SouthAsia	NAmerica	LatinAmer	EU_25	MEIA	SSA	RestofWorld	Grains and Crops	Liv Me
Australia, New Zealand											5470.86	
East Asia											13403.21	
Southeast Asia											19017.48	
South Asia											13406.56	
North America											78980.38	
Latin America											53563.66	
European Union 25											72707.31	
Middle East and North Africa											7597.30	
Sub-Saharan Africa											17189.90	
Rest of World											27491.80	
Grains and Crops	1628.41	45166.89	14725.96	10667.06	34534.85	13419.00	96181.22	16060.23	8774.68	33307.83		
Livestock and Meat Products	1257.37	27600.54	3567.70	710.45	17218.91	4356.69	63032.86	2422.86	2899.07	19383.95		
Mining and Extraction	9455.21	416827.66	68202.07	90717.98	297363.31	35377.76	423494.28	8742.51	15718.52	102812.74		
Processed Food	9131.17	65037.52	25261.48	12142.52	80825.78	22016.18	235287.66	15308.34	18758.60	61598.29		
Textiles and Clothing	7148.06	79334.26	22636.69	11149.33	125823.12	19808.73	218646.17	13868.54	12829.08	73258.90		
Light Manufacturing	50377.86	226457.47	79897.42	39464.47	648833.81	97419.51	1151365.38	38315.93	59522.32	333534.00		
Heavy Manufacturing	96718.02	1261821.50	387924.38	154015.48	1176961.00	240079.50	2372662.00	101033.31	119092.05	698601.50		
Utilities and Construction	1444.59	19565.64	6879.54	1591.53	9805.61	4257.45	75993.51	3427.62	6088.09	36543.52		
Transport and Communication	19522.31	173821.56	50788.73	18943.08	137347.89	34221.13	460401.50	11411.07	19856.88	100875.77		
Other Services	22045.32	165100.83	64577.36	40070.86	252669.31	46522.71	692118.50	21865.73	33369.14	158218.98		
Unskilled labor												
Skilled labor												
Capital												
Land												
Household												

Note: the SAM can only be shown if SAM output was activated on the interface when the simulation was run.

**Model Overview/CES products nests** reports the nesting structure used in the production module:

**CES production nests [0]**

Regions: Oceania  
Items: dNests  
sectors: Paddy rice  
Years: shock

	ND	VA	Labor	CAP+EME	energy	non-electric	non-coal
Paddy rice	1.00						
Wheat	1.00						
Cereal grains nec	1.00						
Vegetables, fruit, nuts	1.00						
Oil seeds	1.00						
Sugar cane, sugar beet	1.00						
Plant-based fibers	1.00						
Crops nec	1.00						
Cattle,sheep,goats,horses	1.00						
Animal products nec	1.00						
Raw milk	1.00						
Wool, silk-worm cocoons	1.00						
Forestry	1.00						
Fishing	1.00						
Coal						0.50	
Oil							1.00
Gas							1.00
Minerals nec	1.00						
Meat: cattle,sheep,goats,horse	1.00						
Meat products nec	1.00						
Vegetable oils and fats	1.00						
Dairy products	1.00						
Processed rice	1.00						
Sugar	1.00						

Similar views are available for the nesting used for factor supply and final demand:

**CES Factor supply nests [0]**

Regions: Oceania  
Items: fNests  
xft

Paddy rice	on
Wheat	on
Cereal grains nec	on
Vegetables, fruit, nuts	on
Oil seeds	on
Sugar cane, sugar beet	on
Plant-based fibers	on
Crops nec	on
Cattle,sheep,goats,horses	on
Animal products nec	on
Raw milk	on
Wool, silk-worm cocoons	on
Forestry	on
Fishing	on
Coal	on
Oil	on
Gas	on
Minerals nec	on
Meat: cattle,sheep,goats,horse	on
Meat products nec	on
Vegetable oils and fats	on
Dairy products	on
Processed rice	on
Sugar	on
Food products nec	on
Beverages and tobacco products	on
Textiles	on
Wearing apparel	on

**CES Final demand nests [0]**

Regions: Oceania  
Items: dNests  
Final demand: Private consumption  
Years: shock

	energy	non-electric	non-coal
Paddy rice	on		
Wheat	on		
Cereal grains nec	on		
Vegetables, fruit, nuts	on		
Oil seeds	on		
Sugar cane, sugar beet	on		
Plant-based fibers	on		
Crops nec	on		
Cattle,sheep,goats,horses	on		
Animal products nec	on		
Raw milk	on		
Wool, silk-worm cocoons	on		
Forestry	on		
Fishing	on		
Coal		on	
Oil			on
Gas			on
Minerals nec	on		
Meat: cattle,sheep,goats,horse	on		
Meat products nec	on		
Vegetable oils and fats	on		
Dairy products	on		
Processed rice	on		
Sugar	on		
Food products nec	on		
Beverages and tobacco products	on		
Textiles	on		
Wearing apparel	on		

The table **Welfare/Income** provides an overview in key results such as GDP and price indices:

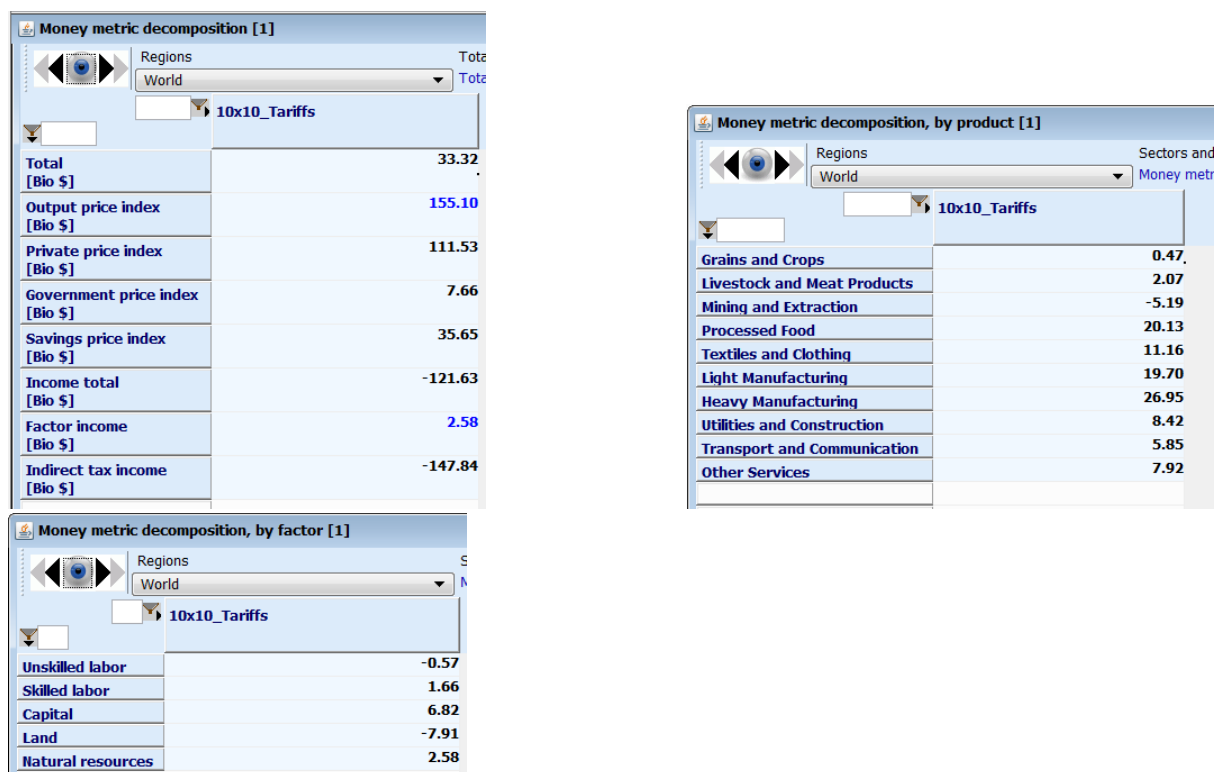
**Income [1]**

Regions: World  
Total or household: Total  
10x10\_Tariffs

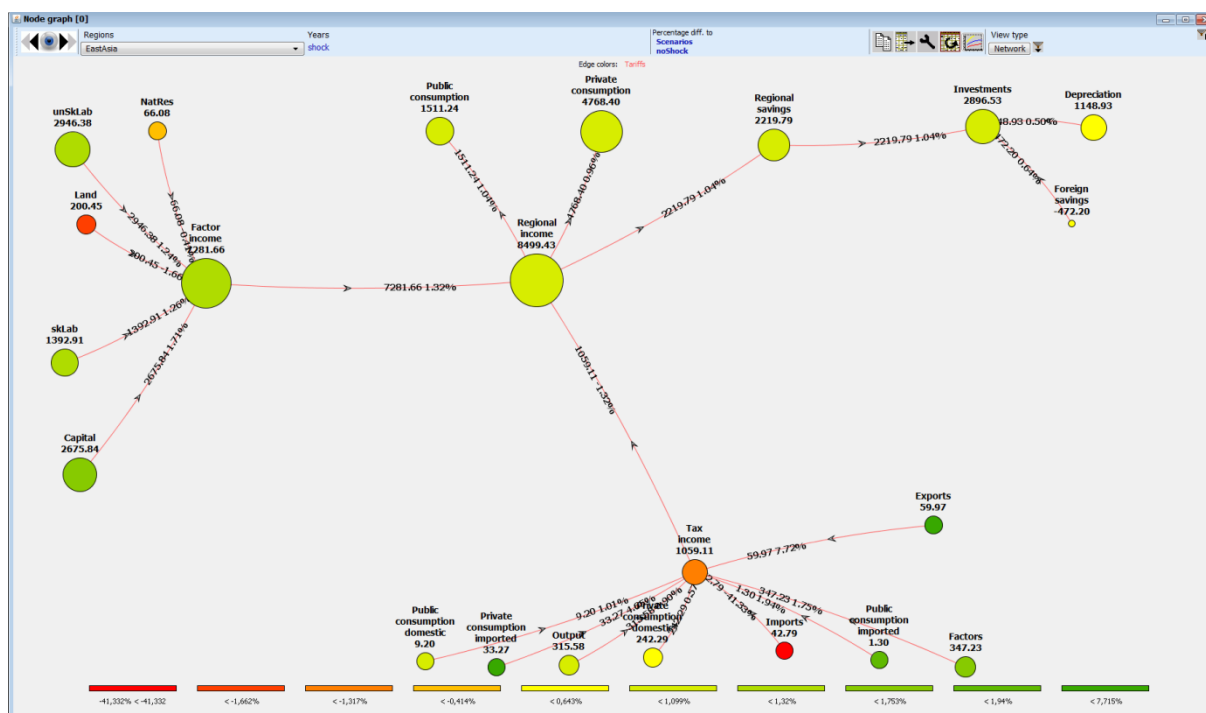
Money metric [Bio \$]	33.32
Total utility [Index]	1.00
Utility of private household [Index]	1.00
Utility of government [Index]	1.00
Utility of saving [Index]	1.00
GDP [Bio \$]	49780.15
Consumer price [Index]	0.99
Government price [Index]	1.00
Factor income [Bio \$]	39510.97
Current account balance [Bio \$]	-0.00
Tax income [Bio \$]	9547.97
Population size [Mil heads]	6620.29



The three tables **Welfare/Money metric decomposition**, **Welfare/Money metric decomposition, by product** and **Welfare/ Money metric decomposition, by factor** support an welfare analysis. “By products” reports welfare gains or losses related to output price changes, by factor linked to changes in factor prices or quantities:



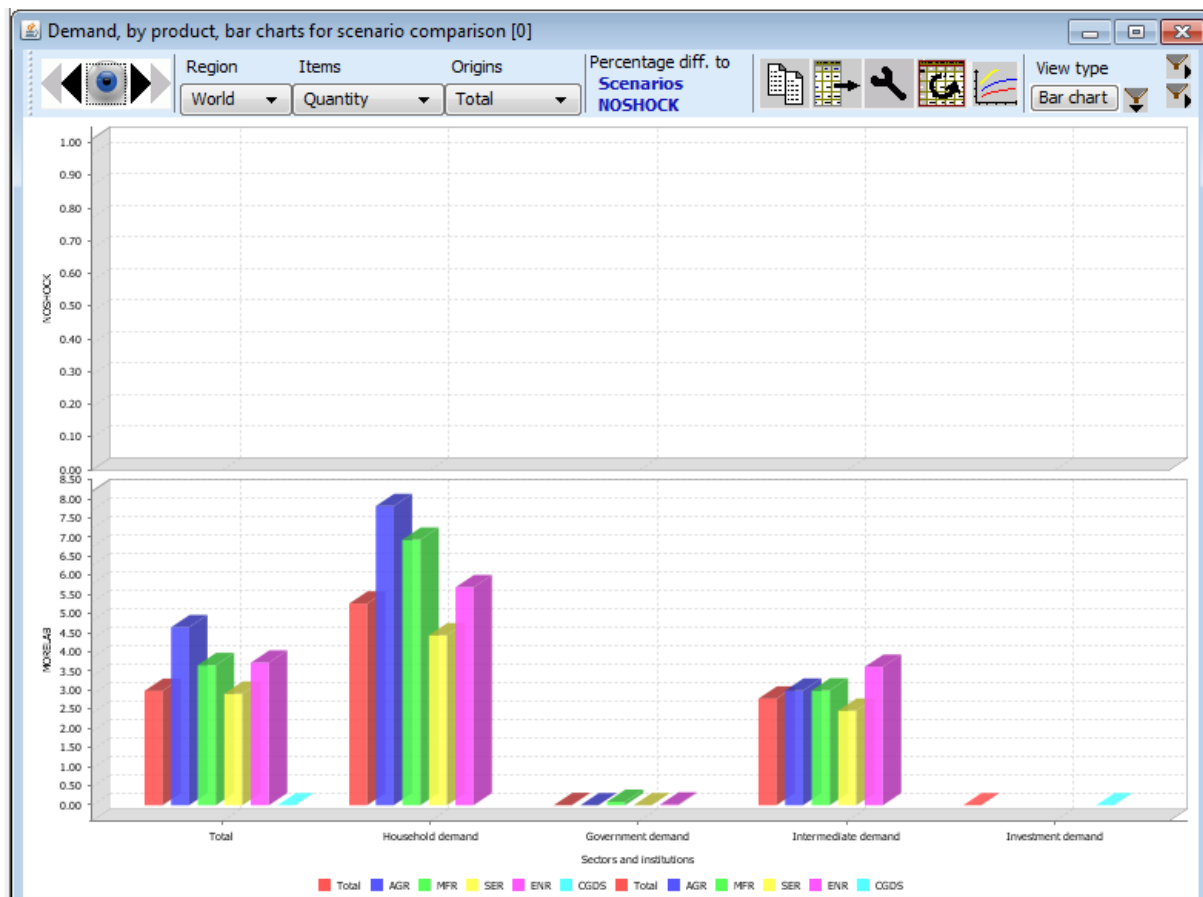
A graphical view on income generation and use provides **Welfare/Node graph**, the following example uses advanced options, such as coloring of the nodes by simulated change and sizing them according to simulated value:



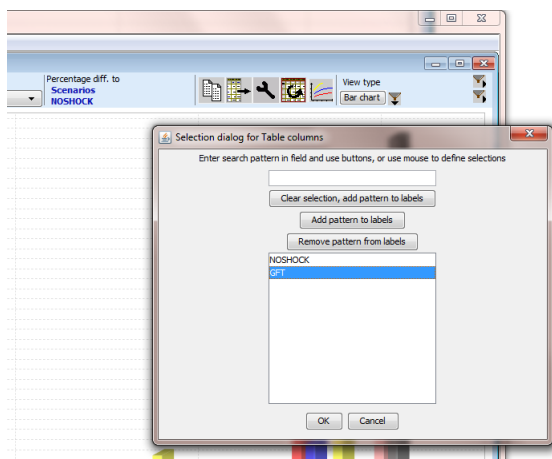
**Markets/Demand by product:** Shows a table with demand by institutions, products and totals shown in the row. The items box allows for looking at quantities, prices, tax rates, value and tax income, the origins by “domestic”, “imported” and “total”. The first box allows selecting the regions. Note: total is an aggregation over all products.

	Total	Household demand	Quantity	Price (net of taxes)	Value (net of taxes)	Tax rate	Tax income	Price, tax inclusive	Value, tax inclusive	Export demand	International transport demand	Total	Household demand	USER_SCENARIOS_TFP_20
<b>Total</b>	137954.328	31209.328								29255.543	541.999	165232.531		19.773%
<b>Crops</b>	2325.690	679.039	3.520	1135.439	12.553					495.141		2654.458		14.136%
<b>Livestock and Fishery</b>	1328.746	369.028	2.529	809.677	42.006					105.506		1574.340		18.483%
<b>Energy Extraction and Exploration</b>	4687.890	22.361	0.001	2336.248	0.013					2329.266		5644.262		20.401%
<b>Other Minerals and Mining</b>	1323.381	36.883	0.638	839.756	21.985					424.119		1587.975		19.594%
<b>Processed Food</b>	6239.726	3103.143	4.847	1874.002	8.685					1249.049		7399.113		18.581%
<b>Textiles and Apparel</b>	3567.711	1105.375	4.233	1124.771	13.450					1319.881		4343.022		21.731%
<b>Light Manufacturing</b>	17505.932	2301.049	59.600	9099.064	263.720					5782.497		21084.553		20.442%
<b>Heavy Manufacturing</b>	30399.992	2108.420	49.331	12138.285	3935.351					12168.604		35848.355		17.922%
<b>Utilities</b>	2871.889	773.766	3.358	1930.692	14.312					149.761		3467.724		20.747%
<b>Services</b>	67703.383	20710.266	9446.101	24099.453	8215.841					5231.719	541.999	81628.727		20.568%

A predefined view “**Demand, by product, bar charts for scenario comparison**” as seen below shows how the combination of a certain pivot, view type while using the comparisons can be used to produce bar charts showing relative differences. It might be useful to the filter out the comparison scenario from the view such that the empty upper graph is not shown.



Note: you use the second “filter” bottom next to view type to “view type” (double click on it) to open a dialogue which lets you remove the benchmark scenario:



That will produce a nicer graph.

**Markets/Demand by institution:** same content as before, but different pivot (could also be quickly produced by using the pivot possibility which can be applied to any view)

Demand, by institution [0]

Region: USA, Items: Quantity, Origins: Total

NOSHOCK

	Total	AGR	MFR	SER	ENR	ENR	ENR	ENR
Total	21263.64	242.58	5951.03	12460.10	619.30	1990.64	21931.60	261.5
Household demand	6925.93	48.30	1389.33	5352.55	135.75		6623.22	42.5
Government demand	1528.65	1.78	228.70	1298.10	0.06		1529.02	1.7
Intermediate demand	9910.88	161.00	3706.44	5571.60	471.84		10417.64	181.1
Investment demand	1990.64					1990.64	1990.64	
Export demand	888.88	31.49	626.56	219.18	11.64		1350.66	35.9

**Markets/Final demand nests:** shows the demand aggregates –quantities, prices and values – according to the active demand nesting:

Final demand nests [0]

Regions: Oceania, Items: Quantity, Origins: Demand

Private consumption, Government consumption, Investment

	Private consumption	Government consumption	Investment
top	576.57	184.14	271.59
energy	18.43	0.01	11.03
non-electric	13.66	0.00	0.00
non-coal	13.65	0.00	0.00

**Markets/Balance:** Shows total use and how it is sourced:

Markets, balance [0]

Region: World, Items: Quantity, Origins: Total

TEST

	Use	Domestic	Imports
Total	136815.61	121112.05	15703.55
Grains and Crops	2194.30	1864.09	330.21
Livestock and Meat Products	1931.71	1786.12	145.59
Mining and Extraction	4864.46	3282.53	1581.93
Processed Food	4500.96	3911.90	589.05
Textiles and Clothing	2405.43	1787.59	617.84
Light Manufacturing	12850.76	9985.44	2865.32
Heavy Manufacturing	26660.95	19783.81	6877.14
Utilities and Construction	11185.15	11018.84	166.31
Transport and Communication	19005.00	17974.03	1030.97
Other Services	38384.38	36885.19	1499.19
That must be the investment good?	12832.52	12832.52	

**Demand, per product, per capita in \$** is an example for a tabled definition which uses the in-built expression evaluator, it divides total demand by the population size:

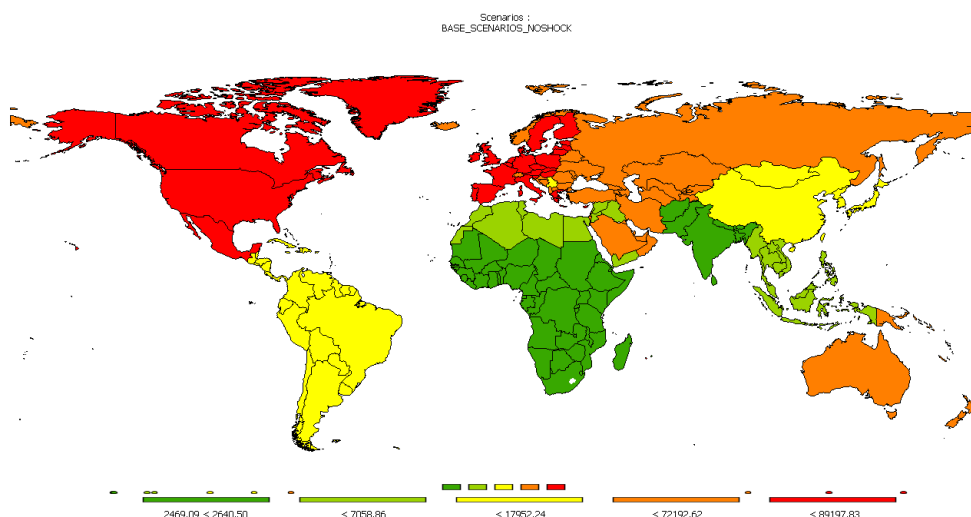
Demand, by product, per capita in \$ [0]

Region: World Items: Q Origins: Total View type: Table

BASE\_SCENARIOS\_NOSHOCK

	Total	Household demand	Government demand	Intermediate demand	Investment demand	Export demand
Total	19660.35	4602.31	1424.48	9700.63	1938.36	1924.02
Grains and Crops	217.49	89.83		111.79		15.87
Livestock and Meat Products	212.19	101.69		100.34		10.15
Mining and Extraction	638.36	7.88		444.66		185.82
Processed Food	635.65	372.83		191.11		71.71
Textiles and Clothing	244.44	103.20		89.15		52.09
Light Manufacturing	1901.86	352.23	4.66	1157.64		387.32
Heavy Manufacturing	3883.43	345.09	2.69	2614.66		920.99
Utilities and Construction	1635.37	125.83	1.69	1507.48		0.37
Transport and Communication	2765.58	1183.57	15.02	1371.13		125.32
Other Services	5587.62	1920.16	1400.42	2112.67		154.38
That must be the investment good?	1938.36				1938.36	

That view can alternatively be visualized as a map in **Demand, per product, per capita in \$, map**:



Maps can be made from any table by putting the regions in the rows and choosing “Map” under “View type”.

**Markets/Intermediate demand:** break of the intermediate demand by sector. Note: the total in the columns is an aggregation over all sectors.

Intermediate demand [0]

Region: USA, Items: Quantity, Origin: Total

NOSHOCK

World, USA, EUR, CHN, ROW

MORELAB

	SER	ENR	CGDS	Total	AGR	MFR	SE
Total	9910.88	129.12	2976.38	4501.76	312.98	1990.64	10417.64
AGR	161.00	18.24	125.81	15.86	0.03	1.05	181.18
MFR	3706.44	49.22	1818.59	1029.74	12.45	796.45	3989.68
SER	5571.60	58.84	922.14	3300.12	97.36	1193.13	5739.79
ENR	471.84	2.82	109.84	156.04	203.14	0.00	506.98

**Markets/Factor demand:** report primary factor use, price etc. by sector

Factor demand [0]

Region: USA, Items: Quantity, Origins: Factor demand

NOSHOCK

MORELAB

	AGR	MFR	SER	ENR	CGDS	Total	AGR
LND	42.10	42.10				42.10	4
LAB	3156.50	32.24	674.81	2414.43	35.02	3472.15	3
SKL	2230.39	2.46	381.57	1830.26	16.09	2230.39	
Capital	3644.02	38.89	694.84	2779.58	130.72	3644.02	4
RES	22.61	2.09	2.35	18.17		22.61	

**Sectors/Sector overview:** reports for each sector total output, output taxes, total intermediate and total factor demand, as well the individual intermediate and factor demands.

Sector overview [0]

Region: World, Items: Quantity, Origins: Total

NOSHOCK

MORE

	AGR	MFR	SER	ENR	CGDS	Total
Total output	65315.98	2009.74	19605.71	34325.31	2633.08	6742.14
Output taxes	1050.87	-5.75	320.25	645.96	90.41	
Total intermediate demand	35040.87	904.67	12931.68	13104.96	1517.95	6581.61
Total intermediate input taxes	491.28	-2.40	110.98	207.20	14.97	160.53
Total factor demand						
Total factor taxes	2813.62	-2.29	661.88	2106.04	47.99	
AGR	1329.38	282.08	889.01	115.06	13.98	29.24
MFR	14741.90	325.16	8106.09	3651.91	172.65	2486.09
SER	16812.19	249.35	3336.39	8756.92	403.27	4066.27
ENR	2157.39	48.08	600.19	581.06	928.05	0.01
LND	294.81	294.81				
LAB	8605.80	484.29	2134.70	5844.96	141.86	
SKL	5205.31	11.98	920.29	4201.65	71.39	
Capital	11613.01	283.11	2510.56	8214.55	604.79	
RES	200.41	41.32	15.38		143.71	

**Sectors/Sector cost shares:** a selection of the value item of the view above, normalized by total output:

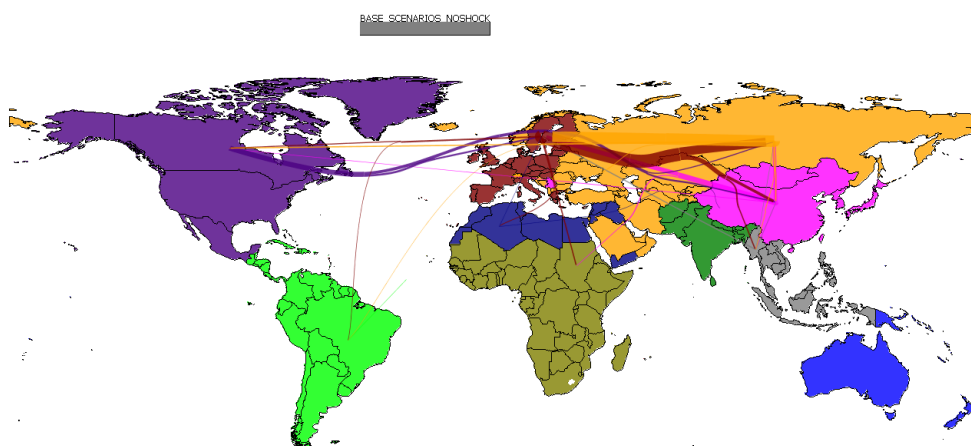
Cost shares [0]

Region: World | Items: Value | Origins: Domestic

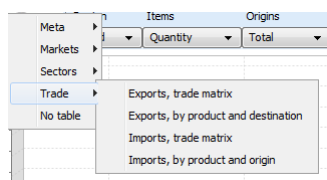
NOSHOCK

	Total	AGR	MFR	SER	ENR	CGDS
Total output	1.00	1.00	1.00	1.00	1.00	1.00
Total intermediate demand	0.45	0.40	0.52	0.34	0.42	0.82
Total intermediate input taxes	0.01	-0.00	0.00	0.01	0.00	0.02
Total factor demand						
Total factor taxes	0.04	-0.00	0.03	0.06	0.02	
AGR	0.02	0.13	0.04	0.00	0.01	0.00
MFR	0.16	0.13	0.30	0.09	0.04	0.22
SER	0.24	0.12	0.16	0.24	0.14	0.60
ENR	0.03	0.02	0.03	0.02	0.22	0.00
LND	0.00	0.15				
LAB	0.13	0.24	0.11	0.17	0.05	
SKL	0.08	0.01	0.05	0.12	0.03	
Capital	0.18	0.14	0.13	0.24	0.23	
RES	0.00	0.02	0.00		0.05	

There are further tables available to bi-lateral trade which can also be visualized as flow maps as shown in the screen shot below,



and more might be added in the future. But the examples above might be sufficient to judge if these views (in addition of using a GDX viewer to look out variables) are a useful addition to result analysis.





## 8. Sensitivity analysis

One of the daunting aspects of using larger simulation models is the high number of parameters needed, both behavioral ones and parameters linked to technology. Generally, most of these parameters are not known with certainty which raises the question how robust results from a shock are. The same often applies to the exact definition of a shock. The idea of sensitivity analysis is to repeatedly solve the same structural model under the same (structural) shock with different parameters to assess the distribution of results. That can help to pinpoint parameters which strongly drive results (which explains the notion of “sensitivity”) but also to learn if the model’s structure with the underlying behavioral and technology assumptions or its parameterization ultimately determine the direction and size of simulated impacts. If the latter are not changing much under different set of parameters, the results can be termed robust. Note here the difference to an econometric exercise where the distribution of the error terms gives immediate information on how robust results are; in most cases, it is even possible to calculate confidence intervals for simulated results. That is generally not possible with large-scale simulation models, not at least as we have not overall knowledge about the “true” multi-dimensional distributions of the parameter space of the model. Still, it is reassuring to find that core results do not differ much even if parameters considered sensitive are changed.

GEMPACK and thus using runGTAP with GEMPACK upports systematic sensitivity analysis (Channing 1996<sup>122</sup>). CGEBox also offers similar functionality drawing on experiences with other GAMS-based models. In order to keep the technical solution relatively easy, the following approach is chosen:

1. The GAMS program which is used to run a single shock is also used for each single sensitivity experiment, in order to use as much as possible existing code and to avoid duplicating code fragments. Note that some parameter or shock changes will requiring a new benchmarking of the model which is supported by the code automatically, even if that drives up somewhat the overall time needed.
2. A second GAMS program “sensitivity.gms” acts as the master program. In combination with a utility from SANDIA labs<sup>123</sup>, it defines the draws and executes in parallel GAMS children. It let each child solve the same (structural) shock. Each child uses its own parameter settings; changes in these settings are defined by the experiments.
3. That second GAMS mother program collects the results from the children and combines them into one GDX file which can be inspected later. It always reports the mean and standard deviations over the draws, but can also store the results for each draw which clearly requires from memory and disk space.

Currently, the sensitivity analysis allows changing any parameters with up to three driving sets. The user can define for each set on each parameter if each element is subject to a specific random shock or if the same shock is applied to all elements on the set.

Whereas Arndt 1996 uses Gaussian Quadratures, the approach in here is based on Latin Hypercube Sampling (LHS). The user chooses the number of draws, and the LHS uses stratified random sampling for space filling design. The LHS is chosen as it can be easily applied also in case of non-symmetric,

---

<sup>122</sup> Arndt, Channing. "An introduction to systematic sensitivity analysis via Gaussian quadrature." GTAP Technical Papers (1996): 2., <https://www.gtap.agecon.purdue.edu/resources/download/1209.pdf>

<sup>123</sup> <http://www.sandia.gov/>

truncated distributions. In the default case, the LHS as a stratified random sampling mechanism simply returns (more or less) uniform distributed draws over the chosen potentially truncated PDF. The mean of the draws is generally close to the mode. The user should keep in mind that non-symmetric bounds will lead to a systematic deviation from the mean parameter used without sensitivity analysis.

The character of an LHS should generally recover the basic character of the distribution with a limited number of draws such as ten. Ten draws will imply that for each parameter, the PDF of its distribution is sliced into ten bins, and one draw is made from each bin. If you draw from a uniform between 0.5 and 1.5, you will hence get one draw from the interval [0.5,0.6], one from [0.6,0.7] .... The GAMS code will scale the draws from the LHS such that the average draw for each factor is equal to its desired mean. You can either use random pairing where the draws for each factor are done independently from each other or use restricted pairing which will try to reduce the correlations between factor draws. The user has the choice between three different distributions, all truncated at the chosen lower and upper limit:

1. Uniform
2. Truncated Normal distribution
3. Triangular distribution

Equally, the user can determine with the draws are added to the base case or multiplied with it. The combination of user set truncation points and distributions allows for a rather flexible design of the experiments. If the lower is equal to the upper bound, the parameter will not be subject to sensitivity analysis. Clearly, at least for one parameter, the upper and lower bounds must differ. The following section details the options.

## 8.1 Controls for sensitivity experiments

Draws		Factors												
Number of experiments	Pairing	Symbol	1. set	Mode	2. Set	Mode	3. Set	Mode	Distribution	Apply	Lo	Up	mode	Std Dev
10	Random	sigmaW	rNat	same	i	same	same	same	Trunc. normal	Mult	0.50	1.50	1.00	0.10
				same		same	same	same	Trunc. normal	Add	-0.50	0.50	0	0.10
				same		same	same	same	Trunc. normal	Mult	0.50	1.50	1.00	0.10
				same		same	same	same	Trunc. normal	Mult	0.50	1.50	1.00	0.10
				same		same	same	same	Trunc. normal	Mult	0.50	1.50	1.00	0.10

The left-hand panel labelled “Draws” determines

- The **number of draws**. For each draw, a full model solve is performed, increasing the number of draws implies a higher sampling accuracy but also longer execution times. If the individual draws are not stored, more draws will not imply higher memory or disk space use for the final result.
- If draws for the factors are generated independent from each other (“**Pairing** random”) or if the LHS should try to reduce correlations between factors (“Pairing restricted”). The latter can drive up the time to construct the experiments.
- How many **GAMS child processes are used in parallel**. It is generally not recommended to use more parallel processes than cores on the computer. As CONOP4 solves in parallel, it might be even best to use only one or two parallel processes for models where the final solve of the full model takes a long time. For larger models, pre-solve can be used which in most cases will help to speed up the overall time especially in sensitivity analysis. Keep also in mind that modern processor layouts allow to temporarily increase the frequency of single cores which can speed up tremendously non-parallelized processing on multi-core machines.

The higher frequency will heat up that specific core, therefore, the BIOS will automatically re-schedule threads to another core once a critical temperature is reached which requires to reduce its frequency. Forcing your computer to run many threads in parallel might hence mean that none of your core can boost its frequency. Here, careful listening to your computer might help: if your computer permanently makes a lot of fan noise, reduce the number of parallel threads. The least wanted outcome from a sensitivity analysis is a destroyed motherboard ... (that would give the term “robustness” a new meaning ...).

- The user might also choose “**Only generate draws**” to prevent the actual model runs, in order to solely check the layout of the scenarios. The file “LHS.lst” found in the scratch directory will show you detailed statistics such as the individual draws, the correlation between the factors etc.. The resulting draws for the LHS are also reported in the GAMS listing file (gams\sensitivity.lst): A look here might help to get an idea if the distributions are sensibly chosen, keeping in mind that it will not report the parameter values themselves, but either the multiplicative change (as below) or additive ones.

```

----- 5745 PARAMETER p_report draws

                                sigmaW
                                Oceania
                                GrainsCrops-c

                                sigmaW
                                Oceania
                                MeatLstk-c

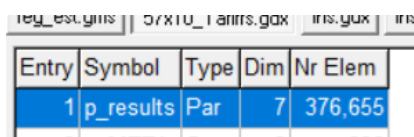
                                sigmaW
                                Oceania
                                Extraction-c

                                sigmaW
                                Oceania
                                ProcFood-c

draws001                        1.00                        1.06                        0.96                        0.91
draws002                        1.22                        0.96                        1.02                        1.04
draws003                        0.97                        1.04                        0.85                        0.94
draws004                        1.11                        0.99                        1.04                        0.98
draws005                        0.96                        1.10                        1.16                        0.86
draws006                        0.88                        0.93                        1.06                        1.23
draws007                        1.03                        1.20                        1.09                        1.05
draws008                        0.93                        1.01                        0.99                        1.00
draws009                        1.07                        0.81                        0.88                        0.91
draws010                        0.82                        0.90                        0.93                        1.08

```

If you only want information on the mean and standard deviation of the results, keep “**Store each draw**” off. Otherwise, all results from all draws are collected in the GDX container. That allows you to generate e.g. histograms or scatter plots as discussed below but can also lead to very large GDX containers. In doubt, run the base case with a normal simulation and open the resulting GDX container from the run under “results\run” with the GAMSIDE or similar. Check the number of non-empty elements in the parameter “p\_results” and its size on disk.



Entry	Symbol	Type	Dim	Nr Elem
1	p_results	Par	7	376,655

Multiply the numbers reported there by the number of draws to get an idea of the size of the resulting combined result set. Don't count on being able to load more than say 10-20 Mio non-zeros into a viewer without manually changing run-time options for the Java run time engine or even running out of memory on your machine.

You might also consider switching some part of the post-model processing off to reduce the size of the result set. For instance, if you are not interested in the distribution of bi-lateral trade flows, tick off “Store bi-lateral results”; you can also reduce the size by only reporting tax rates, but not tax income for each item or you might not need the distributions of the welfare decomposition. Reducing post-model reporting clearly also speeds up running the individual experiments.

Output CSV GUI GDX CONVERT AlterTax Check SolveReport

Store bi-lateral results ☒

Store tax income by item ☒

Trade in VA indicators ☐

Detail for agricultural from FABIO MRIO ☐

Store SAM in GDX ☐

Store benchmark ☐

Welfare decomposition ☒

Post model regional and sector aggregation ☐

The **right hand panel “Factors”** steers the the design of experiments for each parameter subject to sensitivity analysis. The standard example, delivered as the default with the GUI, changes the second level Armington elasticities sigmaW around their given levels, using multiplicative draws with a truncated normal.

Symbol	1. set	Mode	2. Set	Mode	3. Set	Mode	Distribution	Apply	Lo	Up	mode	Std.Dev
sigmaW	rNat	same	i	draws		same	Trunc. normal	Mult	0.50	1.50	1.00	0.10

The **“symbol”** field as the first text field defines the name of the GAMS parameter subject to sensitivity analysis, here “sigmaW”. The three following text fields (**1. Set**, **2. Set**, **3. Set**) are the names of the sets for which changes are introduced, in order they appear on the parameter. In our example, sigmaW is defined over the nations rNat and the commodities i such that the third field has to be left empty.

If we want to restrict the draw to only some regions or commodities, we need to define sub-sets in our shock file and use it in the table above. If a parameter has less than three dimension, leave the last field empty etc.. If a parameter has more than three dimensions, you need to use a helper parameter instead as discussed below.

For each set, we have the **mode** choice between “same” or “draws”, where “same” means that the same random shock being used for all elements in each experiment, and “draws” means that each set elements receives its own shock in each experiment. If you use a 57x10 data base and you chose “draws” both for regions and commodities, you will hence have 570 independent factors in our LHS.

Let’s start with “same”, “same” which implies that we have only one factor level for each draw:

Symbol	1. set	Mode	2. Set	Mode	3. S
sigmaW	rNat	same	i	same	

The include file for a specific experiment will comprise statements as shown below, i.e. all 10x10 = 100 elements in sigmaW will for one experiment (i.e. model solve) be changed by the same relative random change, i.e. the LHS will work with one factor only:

```

sigmaW('Oceania','GrainsCrops-c') = 1.0800361320*sigmaW('Oceania','GrainsCrops-c');
sigmaW('Oceania','MeatLstk-c') = 1.0800361320*sigmaW('Oceania','MeatLstk-c');
sigmaW('Oceania','Extraction-c') = 1.0800361320*sigmaW('Oceania','Extraction-c');
sigmaW('Oceania','ProcFood-c') = 1.0800361320*sigmaW('Oceania','ProcFood-c');
sigmaW('Oceania','TextWapp-c') = 1.0800361320*sigmaW('Oceania','TextWapp-c');
sigmaW('Oceania','LightMnfc-c') = 1.0800361320*sigmaW('Oceania','LightMnfc-c');
sigmaW('Oceania','HeavyMnfc-c') = 1.0800361320*sigmaW('Oceania','HeavyMnfc-c');
sigmaW('Oceania','Util_Cons-c') = 1.0800361320*sigmaW('Oceania','Util_Cons-c');
sigmaW('Oceania','TransComm-c') = 1.0800361320*sigmaW('Oceania','TransComm-c');
sigmaW('Oceania','OthServices-c') = 1.0800361320*sigmaW('Oceania','OthServices-c');
sigmaW('EastAsia','GrainsCrops-c') = 1.0800361320*sigmaW('EastAsia','GrainsCrops-c');
sigmaW('EastAsia','MeatLstk-c') = 1.0800361320*sigmaW('EastAsia','MeatLstk-c');
sigmaW('EastAsia','Extraction-c') = 1.0800361320*sigmaW('EastAsia','Extraction-c');
sigmaW('EastAsia','ProcFood-c') = 1.0800361320*sigmaW('EastAsia','ProcFood-c');
sigmaW('EastAsia','TextWapp-c') = 1.0800361320*sigmaW('EastAsia','TextWapp-c');
sigmaW('EastAsia','LightMnfc-c') = 1.0800361320*sigmaW('EastAsia','LightMnfc-c');
sigmaW('EastAsia','HeavyMnfc-c') = 1.0800361320*sigmaW('EastAsia','HeavyMnfc-c');
sigmaW('EastAsia','Util_Cons-c') = 1.0800361320*sigmaW('EastAsia','Util_Cons-c');

```

If we put instead “**draws**” on the region set rNat:

Symbol	1. set	Mode	2. Set	Mode	3. s
sigmaW	rNat	draws	i	same	

We get for each experiment a different random change for each region as the LHS will work with 10 factors in our 10x10 example data set:

```

sigmaW('Oceania','GrainsCrops-c') = 1.0022894030*sigmaW('Oceania','GrainsCrops-c');
sigmaW('Oceania','MeatLstk-c') = 1.0022894030*sigmaW('Oceania','MeatLstk-c');
sigmaW('Oceania','Extraction-c') = 1.0022894030*sigmaW('Oceania','Extraction-c');
sigmaW('Oceania','ProcFood-c') = 1.0022894030*sigmaW('Oceania','ProcFood-c');
sigmaW('Oceania','TextWapp-c') = 1.0022894030*sigmaW('Oceania','TextWapp-c');
sigmaW('Oceania','LightMnfc-c') = 1.0022894030*sigmaW('Oceania','LightMnfc-c');
sigmaW('Oceania','HeavyMnfc-c') = 1.0022894030*sigmaW('Oceania','HeavyMnfc-c');
sigmaW('Oceania','Util_Cons-c') = 1.0022894030*sigmaW('Oceania','Util_Cons-c');
sigmaW('Oceania','TransComm-c') = 1.0022894030*sigmaW('Oceania','TransComm-c');
sigmaW('Oceania','OthServices-c') = 1.0022894030*sigmaW('Oceania','OthServices-c');
sigmaW('EastAsia','GrainsCrops-c') = 1.0563722580*sigmaW('EastAsia','GrainsCrops-c');
sigmaW('EastAsia','MeatLstk-c') = 1.0563722580*sigmaW('EastAsia','MeatLstk-c');
sigmaW('EastAsia','Extraction-c') = 1.0563722580*sigmaW('EastAsia','Extraction-c');
sigmaW('EastAsia','ProcFood-c') = 1.0563722580*sigmaW('EastAsia','ProcFood-c');
sigmaW('EastAsia','TextWapp-c') = 1.0563722580*sigmaW('EastAsia','TextWapp-c');
sigmaW('EastAsia','LightMnfc-c') = 1.0563722580*sigmaW('EastAsia','LightMnfc-c');
sigmaW('EastAsia','HeavyMnfc-c') = 1.0563722580*sigmaW('EastAsia','HeavyMnfc-c');
sigmaW('EastAsia','Util_Cons-c') = 1.0563722580*sigmaW('EastAsia','Util_Cons-c');

```

And finally, choosing “**draws**” for both sets will generate 10 regions x10 commodities =100 factors

Symbol	1. set	Mode	2. Set	Mode	3
sigmaW	rNat	draws	i	draws	



```

sigmaW('Oceania','GrainsCrops-c') = 1.1184663500*sigmaW('Oceania','GrainsCrops-c');
sigmaW('Oceania','MeatLstk-c') = 0.9331679670*sigmaW('Oceania','MeatLstk-c');
sigmaW('Oceania','Extraction-c') = 0.9607824656*sigmaW('Oceania','Extraction-c');
sigmaW('Oceania','ProcFood-c') = 0.9866457350*sigmaW('Oceania','ProcFood-c');
sigmaW('Oceania','TextWapp-c') = 0.9877797127*sigmaW('Oceania','TextWapp-c');
sigmaW('Oceania','LightMnfc-c') = 1.0682751920*sigmaW('Oceania','LightMnfc-c');
sigmaW('Oceania','HeavyMnfc-c') = 1.0975363260*sigmaW('Oceania','HeavyMnfc-c');
sigmaW('Oceania','Util_Cons-c') = 0.9429919894*sigmaW('Oceania','Util_Cons-c');
sigmaW('Oceania','TransComm-c') = 0.9942992523*sigmaW('Oceania','TransComm-c');
sigmaW('Oceania','OthServices-c') = 0.9188807855*sigmaW('Oceania','OthServices-c');
sigmaW('EastAsia','GrainsCrops-c') = 0.9263090962*sigmaW('EastAsia','GrainsCrops-c');
sigmaW('EastAsia','MeatLstk-c') = 1.0423022530*sigmaW('EastAsia','MeatLstk-c');

```

If you want to work e.g. with a variable or parameter which has more then three dimensions, you can simply type in a new symbol name:

p_shock	rNat	same ▾	i	same ▾	rnat1	same ▾	Ti
---------	------	--------	---	--------	-------	--------	----

The program code will put random draws according to your desired layout in a run specific include file; values which you can pick up in your shock file.

```

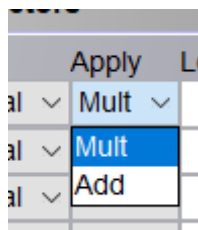
parameter p_shock;
p_shock('Oceania','GrainsCrops-c','Oceania') = 0.7841810194;
p_shock('Oceania','GrainsCrops-c','EastAsia') = 0.7841810194;
p_shock('Oceania','GrainsCrops-c','SEAsia') = 0.7841810194;
p_shock('Oceania','GrainsCrops-c','SouthAsia') = 0.7841810194;
p_shock('Oceania','GrainsCrops-c','NAmerica') = 0.7841810194;
p_shock('Oceania','GrainsCrops-c','LatinAmer') = 0.7841810194;
p_shock('Oceania','GrainsCrops-c','EU_25') = 0.7841810194;
p_shock('Oceania','GrainsCrops-c','MENA') = 0.7841810194;
p_shock('Oceania','GrainsCrops-c','SSA') = 0.7841810194;
p_shock('Oceania','GrainsCrops-c','RestofWorld') = 0.7841810194;
p_shock('Oceania','MeatLstk-c','Oceania') = 0.7841810194;
p_shock('Oceania','MeatLstk-c','EastAsia') = 0.7841810194;

```

Next, we detail how to define the individual draws. For each symbol subject to sensitivity analysis, we have the choice between three **distributions**:

Distribution	Ap
✓ Trunc. normal ▾	M
✓ Trunc. normal	A
✓ Uniform	M
✓ Triangular	M
✓ Trunc. normal ▾	M

Next, we can determine if the random draws are multiplied with the base case of the parameter or added to it (that choice is only relevant if the parameter is already declared such that are values to add something to or multiply with, otherwise, the draws are stored as shown for p\_shock above):



For a multiplicate draw, the range of the draws defined up “lo” and “up” should set around unity which the “mode” close to unity. For additive draws, the “lo” and “up” should be around zero and the mode close to zero. The “Std.Dev” field is only relevant for the truncated normal. The mode and std.dev. in case of a truncated normal refer to the underlying not-truncated normal distribution.

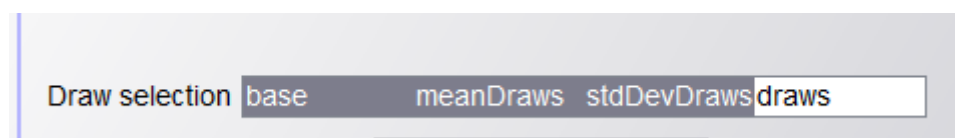
Apply	Lo	Up	mode	Std.Dev
▼ Mult ▼	0.50	1.50	1.00	0.10
▼ Add ▼	-0.50	0.50	0	0.10

Note that bounds which are not symmetric around a mode of unity for multiplicative or a mode of zero in case of additive draws will imply a systematic difference between the parameters of the base case and the mean of the draws for each parameter (besides the fact that the mode of the draws will differ from the parameter inputted).

## 8.2 Parallel solves and analyzing the results

The actual solve for each individual experiment starts from a savepoint generated after loading the data base, such that execution time is only spend on benchmarking, model generation, solving and post-model reporting. Fifty draws of the 10x10 model under a 50% multi-lateral trade simulation experiment took about 30 seconds to solve on my rather fast laptop, another 5 seconds are spend to collect and combine the results. 20 draws with a 57x10 model under the same shock took about two minutes, with the 57x24 model, time goes up to about 6 minutes.

When you inspect the results, the interface will allow showing either all draws and/or the standard case, the average of the draws and the standard deviation:



If only the mean over the draws is selected, the result tables look almost identical to a single simulation run. The tables will report two outcomes: (1) the mean of the draws labeled “meanDraws” and the (2) result at the original parameters “base” – what one would simulate without any sensitivity analysis and (3) the standard deviation over the draws:

Demand, by product [1]		Regions	Items
		World	Quantity
		10x10_example_Policy_Shocks_Tariffs	
		Total	Household demand
All commodities	base	124270.91	
	meanDraws	124270.88	
	stdDevDraws	0.65	

If the draws are selected, you get views such as below:



## Parallel solves and analyzing the results

[<] [v] [>] [x]		RES_BASE_SCENARIOS_FUTURE											
Product		Australia, New Zealand		East Asia		Southeast Asia		South Asia		North America		Latin America	
Private consumption index		draws001	797.37	7544.61	1229.13	1069.38	15272.12						
		draws002	778.09	7940.37	1314.19	481.31	14421.81						
		draws003	810.42	7292.86	1176.10	1168.03	15627.11						
		draws004	603.39	3556.26	1374.50	824.14	14800.16						
		draws005	556.75	3978.81	1475.00	64.01	13897.42						
		draws006	71.61	6402.1	1475.00	64.01	13897.42						
		draws007	849.09	6704.1	1475.00	64.01	13897.42						
		draws008	903.13	6063.1	1475.00	64.01	13897.42						
		draws009	795.44	8931.9	1475.00	64.01	13897.42						
		draws010	778.99	6363.1	1475.00	64.01	13897.42						
		draws011	886.35	5567.1	1475.00	64.01	13897.42						
		draws012	851.03	6647.1	1475.00	64.01	13897.42						
		draws013	800.89	6010.8	1475.00	64.01	13897.42						
		draws014	801.67	7473.1	1475.00	64.01	13897.42						
		draws015	851.37	6700.1	1475.00	64.01	13897.42						
		draws016	789.14	7731.1	1475.00	64.01	13897.42						
		draws017	806.36	7355.1	1475.00	64.01	13897.42						
		draws018	988.29	4842.1	1475.00	64.01	13897.42						
		draws019	866.05	6624.1	1475.00	64.01	13897.42						
		draws020	800.01	7511.1	1475.00	64.01	13897.42						
		draws021	858.21	6672.1	1475.00	64.01	13897.42						
		draws022	853.76	6673.1	1475.00	64.01	13897.42						
		draws023	953.02	6613.1	1475.00	64.01	13897.42						
		draws024	807.46	8504.1	1475.00	64.01	13897.42						
		draws025	871.74	6540.1	1475.00	64.01	13897.42						
		draws026	809.68	7359.1	1475.00	64.01	13897.42						
		draws027	859.39	6665.1	1475.00	64.01	13897.42						
		draws028	862.80	6656.1	1475.00	64.01	13897.42						

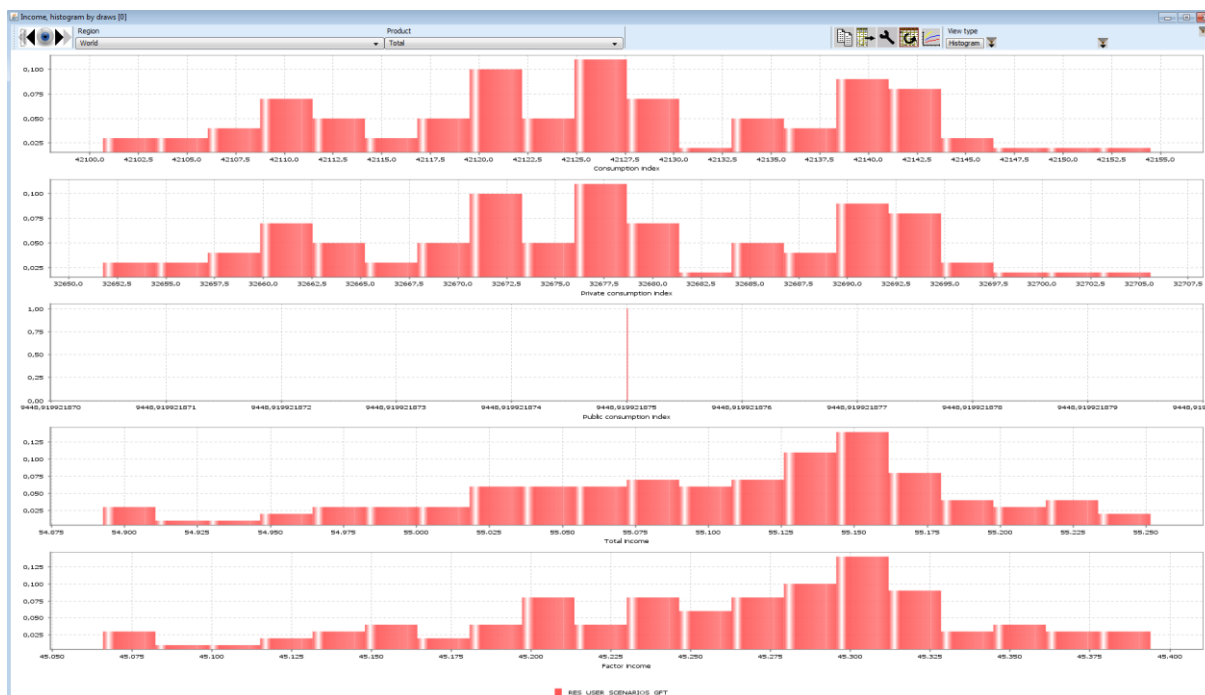
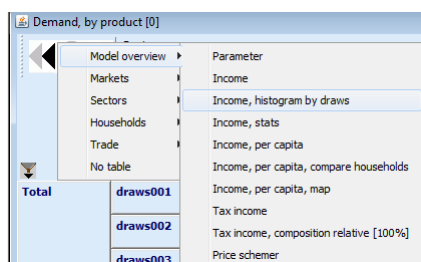
That allows you to analyze each draw, if wished. Alternatively, the statistics built in the interface can be used. In order to do this, open with a right click in the table, the pop-up menu and select statistics:

The 'Set statistics' dialog box is open, showing options to select statistics and check 'Only show outliers'. The 'Set maximum percentage of outliers' is set to 2. The 'Free evaluation field' is empty.

Select the statistics you want to be shown (such as mean and stdDev) and select “Only show outliers” to only show outliers and statistics:

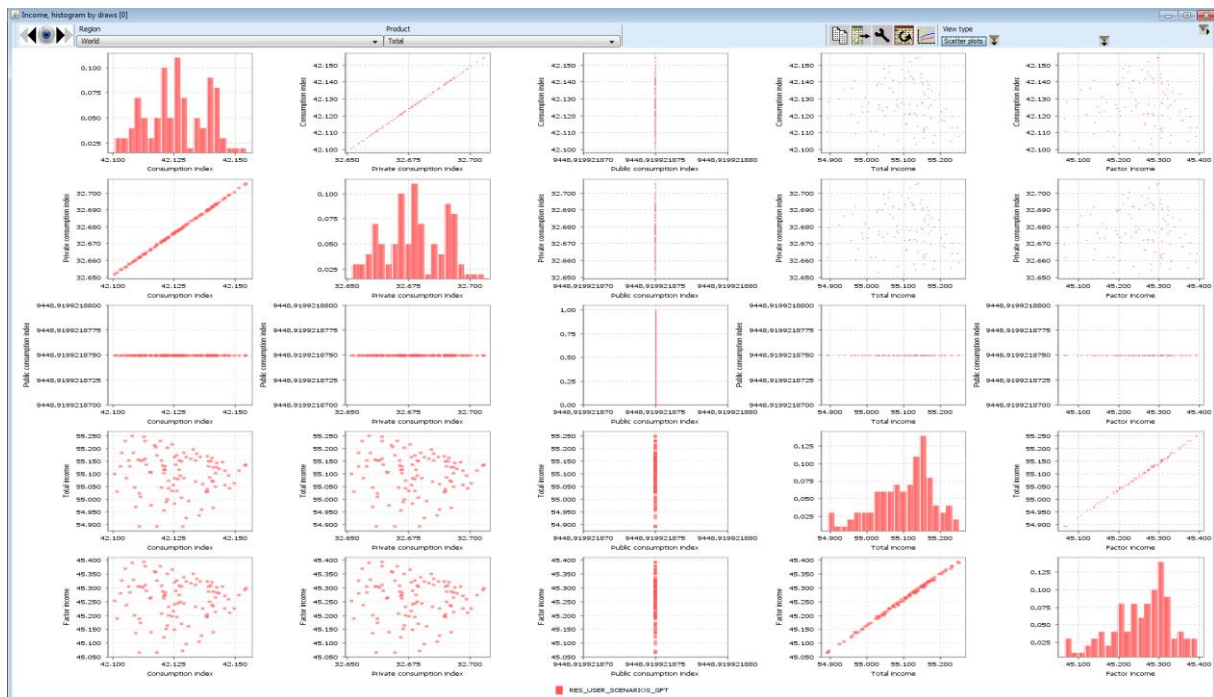
		RES_BASE_SCENARIOS_FUTURE			
		Australia, New Zealand	East Asia	Southeast Asia	South Asia
Consumption index	Mean	1064.58	8790.18	1267.71	
	StdDev	144.51	1437.99	340.46	
Private consumption	Mean	788.21	6588.45	1083.75	
	StdDev	144.51	1437.99	340.46	
Public consumption	Mean	276.37	2201.73	183.95	
	StdDev	Nan	2.07	Nan	
Total income	Mean	1011.97	9587.35	1547.67	
	StdDev	706.44	9601.15	1457.58	
Factor income	Mean	830.71	8670.79	1540.18	
	StdDev	600.08	8442.18	1356.41	
Current account	Mean	21.77	-351.98	-104.45	
	StdDev	2.74	44.34	13.16	
Tax income	Mean	159.50	1268.54	111.94	
	StdDev	106.74	1139.17	100.42	
Population size	Mean	52.11	2326.15	851.49	
	StdDev	0.04	Nan	Nan	

A view with the core income results is defined which generates histograms:

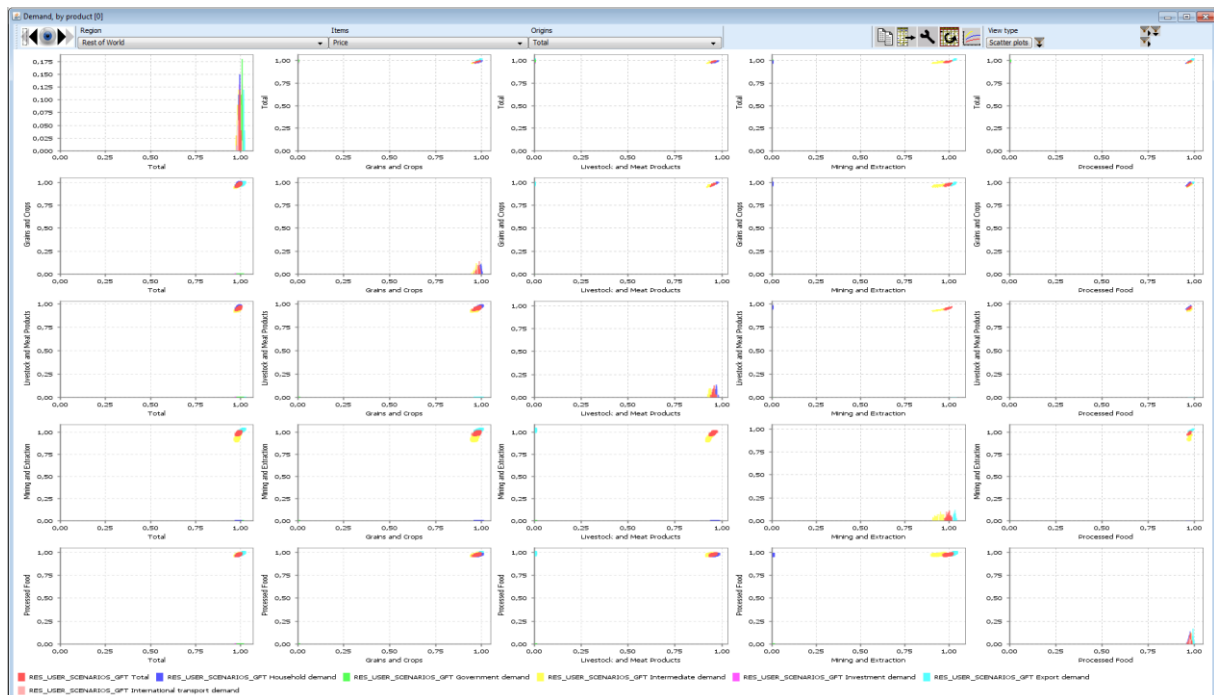


Note: If several scenarios are analyzed, the distributions for several scenarios will be shown together.

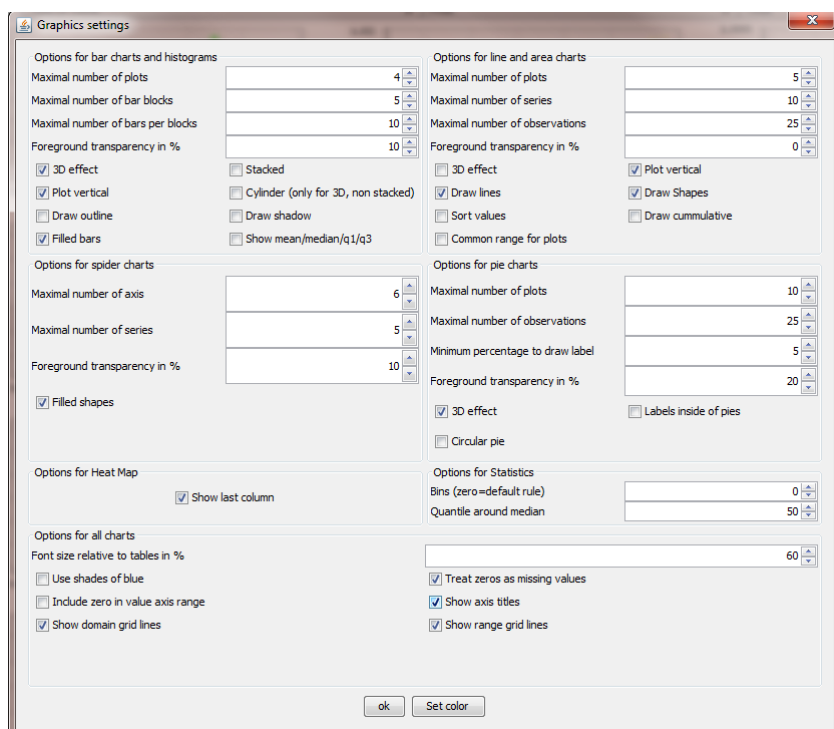
Alternatively, the “Scatter plot” type graphics can be used to plot additionally the correlation between the variables depicted:



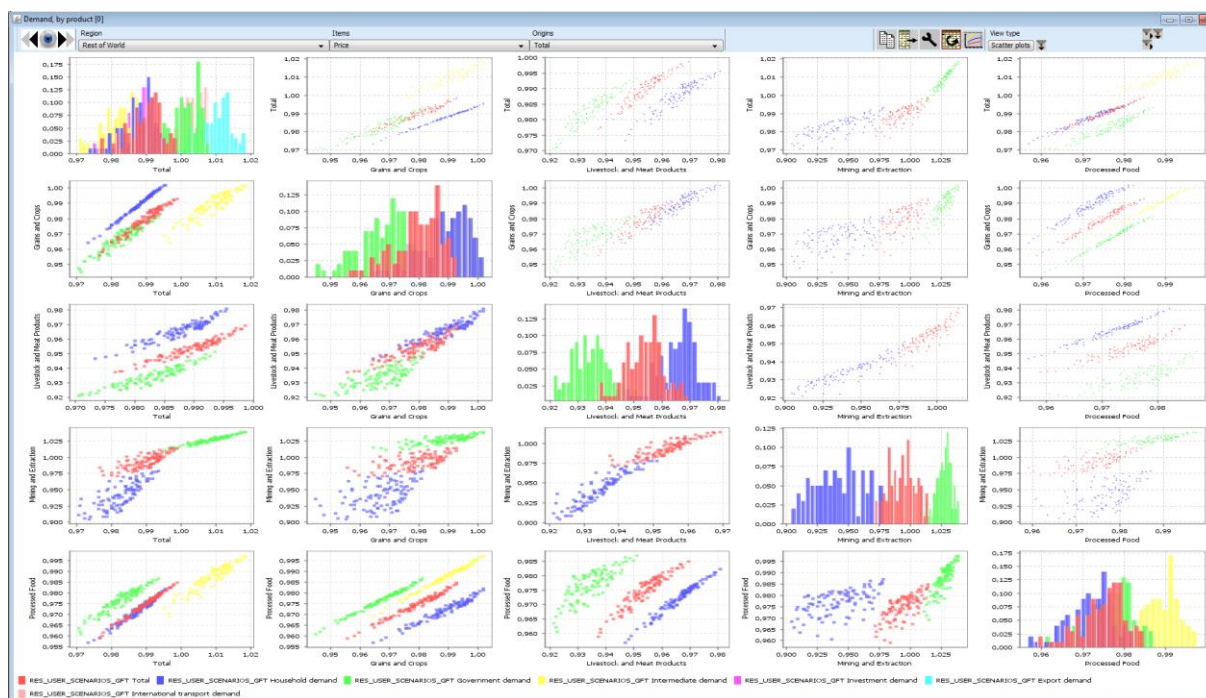
If the view type from existing tables is changed to “Scatter plots”, the result might look as below (the example shows different types of prices in each plot, and different sectors in the rows/columns):



The visualization of histograms and scatter plots can be improved, if the graphics settings are changed such that (1) zeros are treated as missing values, and (2) zero is not automatically included in the axis range:



The resulting scatter plots and histograms look as follows:



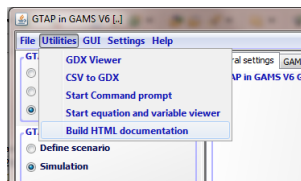
## 9. Getting an overview on the code

The GUI offers two possibilities to ease working with the code:

1. Generation of a set of static HTML pages
2. An equation, parameter and variable viewer linked to a specific model instance, which also allows looking at the GAMS code.

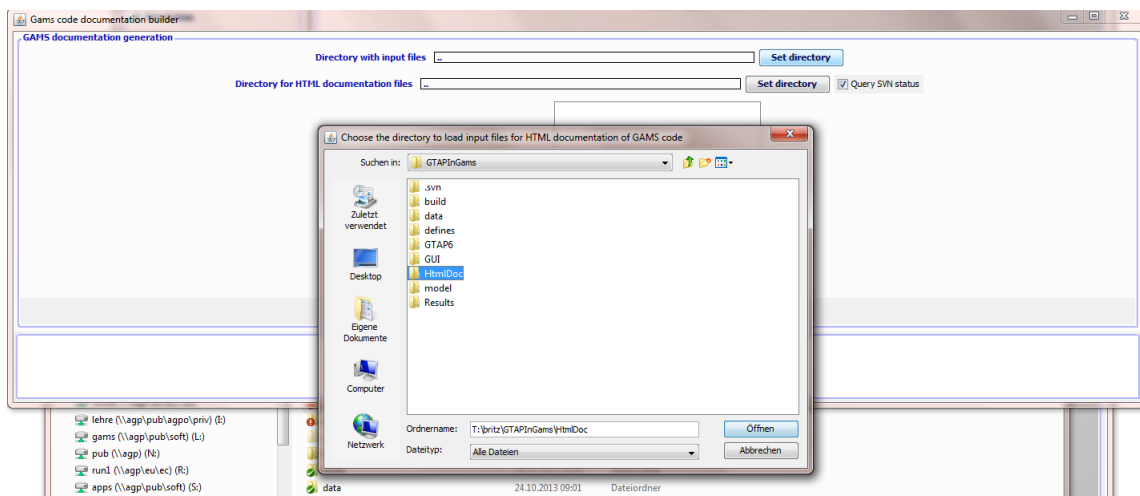
### 9.1 HTML documentation

A HTML documentation of the code can be generated via “**Utilities/Build HTML documentation**”:

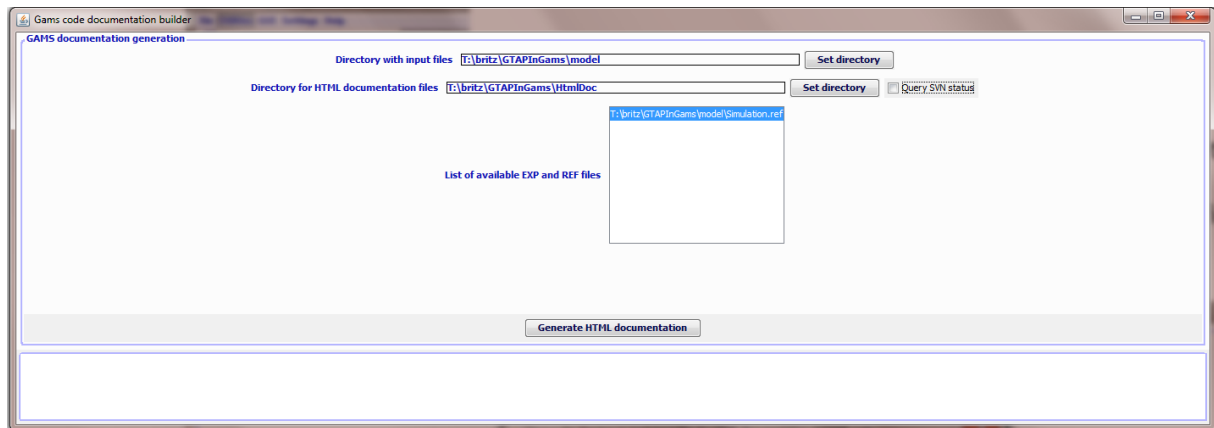


The documentation will include any changes you introduced to the project’s GAMS code.

First choose the directory where the files should be stored (best use the “HtmlDoc” directory as shown below; beware: the utility will generate many files, do not use a directory where other files are already stored). The input files are found in the folder where GAMS was run, in the case of standard GTAP model in GAMS, typically in the “model” directory.



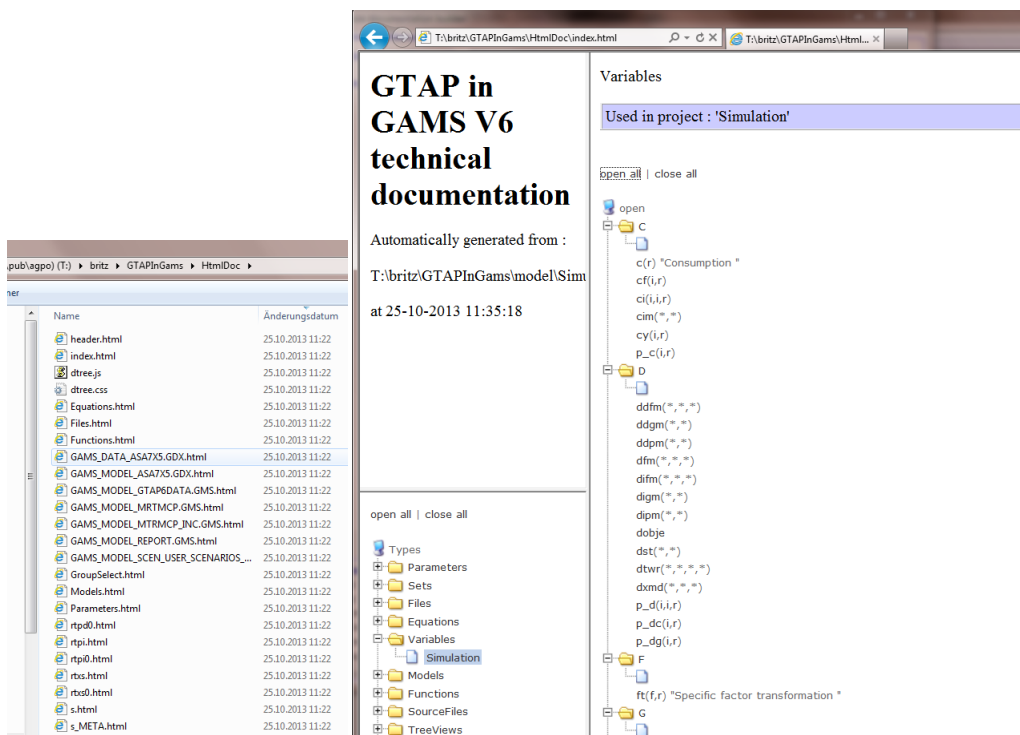
You should be able to find a file named “simulation.ref”. Make sure that “Query SVN Status” is switched off and next press the “Generate HTML documentation” button.



The program will work for a while and should end with “HTML documentation is ready” as shown below.

```
HTML documentation is ready
parsing T:\britz\GTAPInGams\model\mrtmcp.gms done
parsing T:\britz\GTAPInGams\model\report.gms done
parsing T:\britz\GTAPInGams\model\scen\user_scenarios\noshock.gm
Generate HTML page for Gams symbols
Generate HTML page for Gams source files
Generate HTML pages for project : Simulation
HTML pages are generated
```

Afterwards, you find in the chosen output directory a list of files, “index.html” is the starting point.

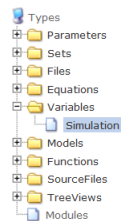


You can e.g. open all “Variables” used in the task “simulation”, clicking with the mouse opens a symbol page:

## GTAP in GAMS V6 technical documentation

Automatically generated from :  
T:\britz\GTAPInGams\model\Sim  
at 25-10-2013 11:35:18

open all | close all



### Variable ddfm

Name	ddfm(*,*,*)
Type	Variable
Number of dimensions	3
Used in project(s)	<a href="#">Simulation</a>

ddfm is declared in:

[Top](#) | [Declarations](#) | [Definitions](#) | [Assignments](#) | [References](#) | [Elements](#)

#### File

#### Projects

[GAMS\MODEL\MRTMCP.GMS](#) [Simulation](#) [Edit](#)

ddfm is assigned in:

[Top](#) | [Declarations](#) | [Definitions](#) | [Assignments](#) | [References](#) | [Elements](#)

#### File

#### Projects

[GAMS\MODEL\MRTMCP.GMS](#) [Simulation](#)

ddfm is referenced in:

[Top](#) | [Declarations](#) | [Definitions](#) | [Assignments](#) | [References](#) | [Elements](#)

#### File

#### Projects

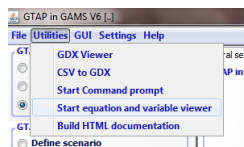
[GAMS\MODEL\MRTMCP.GMS](#) [Simulation](#)

[GAMS\MODEL\REPORT.GMS](#) [Simulation](#)

[Top](#) | [Declarations](#) | [Definitions](#) | [Assignments](#) | [References](#) | [Elements](#)

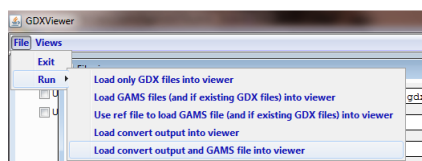
## 9.2 Equation and variable viewer (similar AnalyzeGE)

Whereas the HTML pages document the project, the equation and variable viewer helps you to analyze a specific model instance. The necessary input files are automatically generated with each run in comparative static mode if “CONVERT” output is chosen.



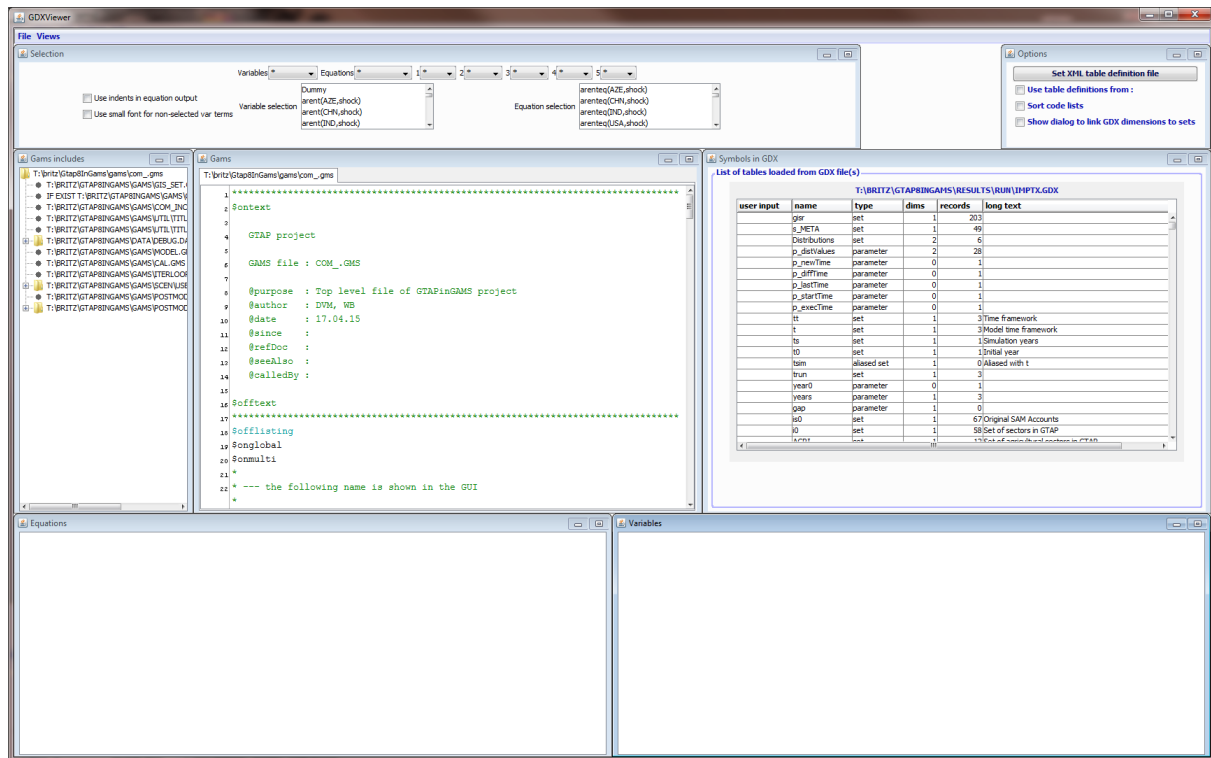
(note: one might want to change the first GDX files to the last experiment you have run)

Select “Load convert output and GAMS file into viewer” from the file menu:



You get now view like below:





The viewer allows you to see:

- Linearized views on the equations, which you can select via the selection button above. The information in green behind each equation provides, separated by commata:
  1. The absolute change in the variable
  2. The absolute change multiplied with the Jacobian, i.e. the approximate change in the equation due to the simulated change
  3. The contribution of the variable to the change in the LHS in percent, using information in the absolute changes in the LHS and the variable and the related Jacobian entries. The variable which the code analyzer has detected as the LHS is indicated with a \* - in the example below that is xd. For the LHS, the relative change is report. In example below, the change is almost entirely coming from the change in nd (100%), the contribution of the other changes is very small.

```
xdeg(SouthAsia,MeatLstk-c,ProcFood-a,shock) "Agents demand for domestic goods " ..
-0.00902671001153436*(0.9999999486802*(0.995354810037336*(1.0000000513198*ps(SouthAsia,MeatLstk-c,shock)
{-0.005,-7.2E-5,1.0%}))**(-2.13050842285156)
+0.00464540767085857*(1.00000000249911*pmt(SouthAsia,MeatLstk-c,shock){0.0033,-4.7E-5,0.8%})**(-2.130
50842285156))**(-0.469371530886303)/ps(SouthAsia,MeatLstk-c,shock){-0.005,-7.2E-5,1.0%})**3.130508422
85156*nd(SouthAsia,ProcFood-a,shock){0.67,-0.006,100.0%}
+0.183822553739227*xd(SouthAsia,MeatLstk-c,ProcFood-a,shock){0.034,0.0062,0.6%*})=E=0;
```

- A GDX viewer which allows merging of symbols (from one or several files) for combined analysis.
- A quick view on any symbol: click on any symbol in the GAMS code shown in blue, and it will be loaded in the “Symbol from GDX” view. You can directly compare your counterfactual against the base (or any scenarios against each other which you loaded as GDX file).

## Equation and variable viewer (similar AnalyzeGE)

Gams

T:\pritz\GTAPInGams\model\wrtmcp.gns

```

62: *
63: *   Profit function for firms:
64: *
65: *   Value shares for firm inputs of factors and goods:
66: *
67: parameter          thetaf(f,j,z)   Factor share of value added,
68:                      thetad(i,j,z) Domestic share of intermediate input,
69:                      thetai(i,j,z) Import share of intermediate input,
70:                      theta_g(j,z)  Value added share of sectoral output;
71:
72: alias (f,ff);
73: thetaf(f,j,z)$sum(ff,vfm(ff,j,z)*(1+rtf0(ff,j,z)))
74:   = vfm(f,j,z)*(1+rtf0(f,j,z)) / sum(ff,vfm(ff,j,z)*(1+rtf0(ff,j,z)));
75: thetad(i,j,z)$ (vdfm(i,j,z)*(1+rtfd0(i,j,z)) + vifm(i,j,z)*(1+rtfi0(i,j,z)))
76:   = vdfm(i,j,z)*(1+rtfd0(i,j,z)) /
77:     (vdfm(i,j,z)*(1+rtfd0(i,j,z)) + vifm(i,j,z)*(1+rtfi0(i,j,z)));
78:
79: thetai(i,j,z)$vom(j,z)
80:   = (vdfm(i,j,z)*(1+rtfd0(i,j,z)) + vifm(i,j,z)*(1+rtfi0(i,j,z))) / vom(j,
81: z);
82: theta_f(j,z)$vom(j,z) = sum(ff,vfm(ff,j,z)*(1+rtf0(ff,j,z))) / vom(j,z);

```

Symbol from GDx

Dim2 (noShock.gdx,GFT.gdx,...)

noShock.gdx

SAF

RSA

RSS

EUR

ROW

View type

Table

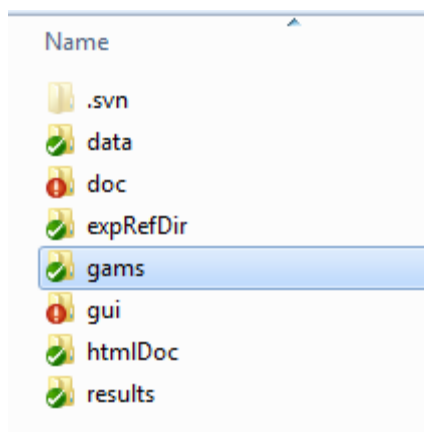
	SAF	RSA	RSS	EUR	ROW	
agr		0.37	0.55	0.68	0.45	0.51
foo		0.28	0.39	0.29	0.32	0.29
ext		0.32	0.56	0.55	0.31	0.34
lgt		0.38	0.38	0.36	0.34	0.34
hvy		0.29	0.25	0.22	0.21	0.24
tch		0.36	0.41	0.33	0.34	0.33
ser		0.48	0.46	0.41	0.49	0.50

## 9.3 Technical documentation

### 9.3.1 Overview on directory structure

The screenshot below shows the directory structure of a “CGEBox” installation. The information is only provided for those who have an interest into technical detail.

The GUI with the Java libraries is found in GUI. The results directory comprises the output from model runs. The data directory comprises also the GAMS readable versions of available data bases. All GAMS files can be found under “gams”, in the subdirectory “scen”, scenario templates and user defined scenarios are stored. Results of model runs are found under “results/run”. The directory “expRefDir” and “HtmlDoc” are used to generate a HTML based documentation of the GAMS project.

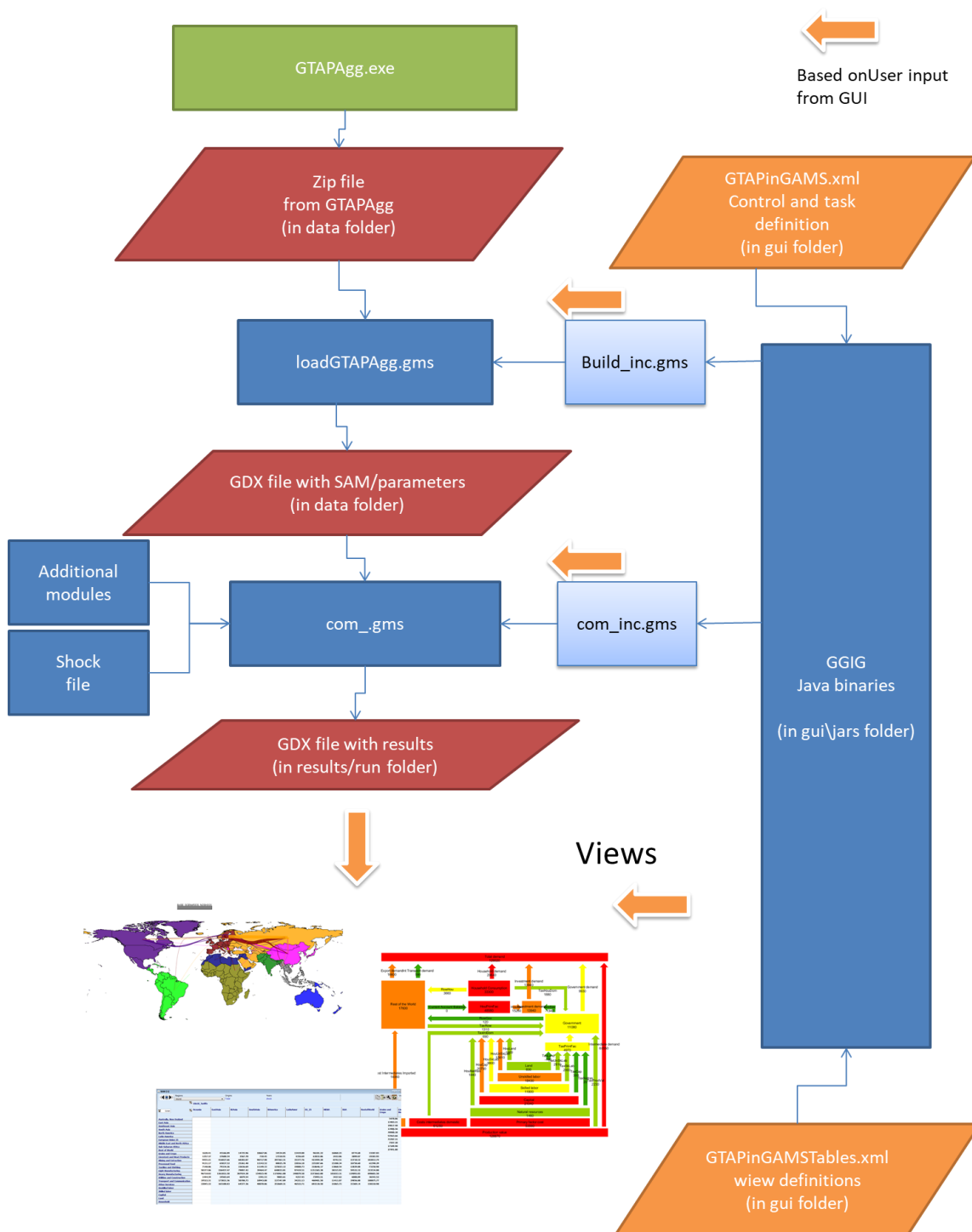


### 9.3.2 GUI overview

The GUI is programmed in Java and based on GGIG ([Gams Graphical Interface Generator](#)), which is also used for other projects, e.g. CAPRI ([www.capri-model.org](http://www.capri-model.org)). While binaries are distributed for free, the Java code itself is currently not open source. It uses third-party libraries provided under licenses which allow for further distribution, and often as well code changes. The main jar to start is “jars\ggig.jar”. In order to modify the GUI, no Java programming is necessary, as the Java code reads a XML file (“gui\GtapInGams.xml”) which defines tasks, work steps and the individual GUI controls. Equally, a XML file (“gui\GtapInGamsTables.xml”) defines the different views. The link to GDX files is based on the Java API distributed by GAMS.com as part of any GAMS installation, and uses dynamic link libraries. As both Java and GAMS are available also for non-Windows operation systems, it would be possible to also port the standard GTAP model in GAMS to other software platforms. Indeed, a CAPRI user had a few years back generated a native MAC version.

The interaction between the GUI and the GAMS processes is based on includes files which are generated anew for each run and capture the state of the user operated control.

### Graphic: Overview on interaction between user, GUI and GAMS



### 9.3.3 Processes

In order to work with the model, GTAPAGG (which is clearly not part of the installation) must first be used to produce a zip file which comprises the necessary GTAP data, parameters etc. at the user chosen aggregation with regard to sectors, regions and factors. The GTAPAGG output should be stored in the “data” directory. The zip file comprises data in proprietary GEMPACK formats and cannot be used directly with GAMS code.

Next, the “LoadGTAPAgg.gms” programs reads that output from GTAPAGG and converts it into GAMS symbols which are stored in a GDX container. It uses the “ConvertGtapAgg.jar” to read the content of the “agg” file with the aggregation information used by GTAPAgg to generate definitions with long texts which are temporarily outputted into “agg.gdx”..

On demand, the “filter.gms” program processes the inputted data to remove small values. The result from these step are stored in the “data” directory.

The GDX files generated from “loadGTAPAgg.gms” provide the input to actual model runs with “com\_.gms”. The GAMS code initializes all variables to benchmark values and starts the solver. That model start should basically need no iterations and lead to zero infeasibilities, it should prove that the model’s parameters are correctly set up to replicate the benchmark. Next, the user chosen shock file is read and changes to parameters introduced. Afterwards, the model is solved for the shock.

#### 9.3.4 Post-model processing

While the processes described above are more or less identical to the original code of Dominique and Van der Mensbrugghe and Tom Rutherford, the post-model processing was added to allow using the exploitation part of GGIG. The file “report.gms” stores the results back into a SAM like structure, e.g. consumer prices

```
p_results(r,"P",i,"hou","dom") = p_dc.l(i,r);
```

The “p\_results” parameter is stored in a gdx file and later read by the interface for exploitation. It is generally organized as follows:

- First dimension: regions, including a world aggregated stored under “wor”
- Second dimension: tables for values “V”, quantities “Q”, prices “P”, tax rate “T” and tax income “G”
- Third dimension: commodities, factors
- Fourth dimension: institutions, sectors
- Fifth dimension: origins / destinations
- Sixth dimension: base, check or shock

Based on the flexible aggregation used in GTAP, the sets used for regions, sectors and factors are run specific and depend on the data set loaded.

The information about how the output is logically structured is also inputted in the GGIG definition file “gtapInGams.xml”:

```
<task>
  <name>Simulation</name>
  <gamsFile>mrtmcp</gamsFile>
  <incFile>model\mtrmcp_inc</incFile>
  <curDir>model</curDir>

  <regionDim>8</regionDim>
  <dim5Dim>1,Items</dim5Dim>
  <productDim>2</productDim>
  <activityDim>3,Sectors and institutions</activityDim>
  <dim6Dim>4,Origins</dim6Dim>
  <scenDim>5,Scenarios</scenDim>

  <resdir>run</resdir>
  <gdxSymbol>p_results</gdxSymbol>
  <Filemask>.*gdx$</filemask>
</task>
```

The “report.gms” program also performs aggregations, e.g. to total use

```
*
* --- aggregate over origins (domestic and imported) to totals
*
  set agg(PQU) / Q,U,G /;

*
* --- aggregate to total use
*
  set useCols / set.i,hou,gov,intTrs,set.s,inv /;
  p_results(r,agg,i,"use",oris) = sum(useCols, p_results(r,agg,i,useCols,oris));
```

Average prices and tax rates are calculated afterwards:

```
p_results(r,"P",i,"use",oris) $ p_results(r,"Q",i,"use",oris)
= p_results(r,"U",i,"use",oris) / p_results(r,"Q",i,"use",oris);

p_results(r,"T",i,"use",oris) $ p_results(r,"U",i,"use",oris)
= p_results(r,"G",i,"use",oris) / p_results(r,"U",i,"use",oris);
```

The “report.gms” programs also stores meta-information on the run on the cube, such as the number of sectors, regions and factors; model size and solution status, and about how factor mobility is modeled:

```

* -----
*
*   Meta information on model structure and solve
*
* -----
*
p_results("Wor","Meta","# of sectors","tots","dom") = card(i);
p_results("Wor","Meta","# of factors","tots","dom") = card(f);
p_results("Wor","Meta","# of regions","tots","dom") = card(r);

acronym MCP,CNS;
$iftheni %modeltype% == MCP

p_results("Wor","Meta","Model type","tots","dom") = MCP;
$else

p_results("Wor","Meta","Model type","tots","dom") = CNS;
$endif

$iftheni NOT %modeltype% == NONE

p_results("Wor","Meta","# of equations","tots","dom") = %model%.numequ;
p_results("Wor","Meta","# of variables","tots","dom") = %model%.numvar;
p_results("Wor","Meta","SumInfes","tots","dom") = %model%.sumInfes $ (%model%.sumInfes ne inf);
p_results("Wor","Meta","NumInfes","tots","dom") = %model%.numInfes $ (%model%.numInfes ne inf);
p_results("Wor","Meta","# Iterations","tots","dom") = %model%.iterusd;
p_results("Wor","Meta","# seconds solution time","tots","dom") = %model%.resusd;

acronym sluggish, mobile, fixed priced;

p_results("Wor","Meta","Lab Market","tots","dom") = %labmarket% ;
p_results("Wor","Meta","skl Market","tots","dom") = %sklMarket% ;
p_results("Wor","Meta","capital Market","tots","dom") = %capitalMarket% ;
p_results("Wor","Meta","res Market","tots","dom") = %resMarket% ;
p_results("Wor","Meta","lnd Market","tots","dom") = %lndMarket% ;

$endif

```

That approach differs considerably from the way “runGTAP” allows to exploit results. For a formal discussion on these differences, see Britz 2014 and Britz et al. 2014. The “equation and variable” viewer, discussed above, allows views on the variables, equations and related parameters, more similar to the “runGTAP” exploitation tools.

### 9.3.5 Substitution of variables based on macros

With highly detailed SAMs, the number of variables and equations could reach several millions. Beside filtering, the possible to aggregate the Armington first nests, variable substitutions are used to reduce model size. We will in here briefly discuss some of these macros to ease reading the code. Generally, macros are named as the related variables with the prefix “m\_”. The macro m\_pa hence defines the variable p\_a in the model, i.e. the Armington prices for the agents. As these is a complex case, it will be used here as the example.

```

*
*   --- same as above, difference is each sector has its own Armington nest for intermediate demand
*
$macro m_pa(r,i,aa,t) { [ pa(r,i,aa,t) $ (not a(aa)) \
+ m_pade(r,i,aa,t) $ a(aa) ] $ (alphan(r,i,aa,t) $ alphad(r,i,aa,t)) \
+ (m_pdp(r,i,aa,t)*alphad(r,i,aa,t)) $ ((not alphan(r,i,aa,t)) $ alphad(r,i,aa,t)) \
+ (m_pmp(r,i,aa,t)*alphan(r,i,aa,t)) $ ((not alphad(r,i,aa,t)) $ alphan(r,i,aa,t)) \
+ pa(r,i,aa,t) $ ( not (alphan(r,i,aa,t)*alphad(r,i,aa,t)) ) \
}

```

The macro considers four different cases:

- (1) Both domestic and import demand is present (alphan and alphad is given). In that case, for agents which are not production sectors, the variable pa itself is used and defined in an equation (first line). Otherwise, another macro m\_pade is used.
- (2) Only domestic demand is present. In that case, the price is equal to the domestic price, defined by the macro m\_pdp.



- (3) Only import demand is present, in which case only the import price defined by the macro `m_pmp` is used.
- (4) No import and domestic shares are defined, a case found when the Melitz model is used.

The macro `m_pade` is the actual dual price aggregator:

```
*
* ---- definition of Armington dual price aggregator from domestic and import price
*
**macro m_pade(r,i,aa,t) { ( (alphad(r,i,aa,t)*m_pdp(r,i,aa,t)**(1-sigman(r,i))) $ alphad(r,i,aa,t) \
+ (alphan(r,i,aa,t)*m_pmp(r,i,aa,t)**(1-sigman(r,i))) $ alphan(r,i,aa,t)**(1/(1-sigman(r,i))) ) }
```

The domestic price macro `m_pdp` is defined as follows, i.e. it applies taxes to the domestic price, again defined via the macro `m_domPrice`. As taxes are usually fixed, it will in most cases simply remove a linear equation by substitution:

```
*
* --- the macro m_pdp substitutes out the domestic sales prices: with an infinite CET for exports, use the supply
* price ps, otherwise, use the domestic prices
*
**macro m_pdp(r,i,aa,t) { m_domPrice(r,i,aa,t) \
*(1 + dintx(r,i,aa,t) + itxshft(r,aa,t) $ iTaxShift(i)) \
*
* --- add emission taxes for domestic use
**$if1 %ModulesCO2_Emissions%==on + emisp(r,t) * (emid0(r,i,aa)/xda0(r,i,aa)) $ xda0(r,i,aa) \
}
```

The macro for domestic price reflects if a CET approach is used on the supply side. If the CET is infinite, the supply price `ps` is used directly, otherwise the domestic sales prices `pd` is used (we refrain from a discussion of the details of the Melitz model):

```
**macro m_domPrice(r,i,aa,t) [ ( pd(r,i,t) $ (omegax(r,i) ne inf) + ps(r,i,t) $ (omegax(r,i) eq inf)) $ (not iMel(i)) \
+ ( pd(r,i,t) $ (not mFlag(r,aa,i)) + ps(r,i,t) $ mFlag(r,aa,i)) $ iMel(i) ]
```

The macro for the import price `m_pmp` is defined as follows, as for the domestic prices, it considers taxes on the import price `pmt`:

```
*
* --- the macro m_pmp substitutes out the imports sales prices
*
**macro m_pmp(r,i,aa,t) {pmt(r,i,t)*(1 + mintx(r,i,aa,t) + itxshft(r,aa,t)$ iTaxShift(i)) \
*
* --- add emission taxes for domestic use
**$if1 %ModulesCO2_Emissions%==on + emisp(r,t) * (emi0(r,i,aa)/xma0(r,i,aa)) $ xma0(r,i,aa) \
}
```

A large set of equations is dropped by substituting out prices relating to bi-lateral trade flows. The basis free-on-board price for products not treated a la Melitz are defined via a macro `mm_pegob` as follows:

```

*
* --- free on board price (f.o.b.):
*
* $$macro mm_pefob(r,i,rp,t) { \
*
*     --- export taxes
*
*     [(1 + exptx(r,i,rp,t) + etax(r,i,t)) \
*
* --- the following might substitutes out pe depending on the CETs
* between domestic sales and export resp. between exports
*
*     * ( \
*
*         --- use bilateral import prices if a CET between export destination is active
*         pe(r,i,rp,t) $ psFlag(r,i,"pe") \
*         --- use the export prices if there is only a CET between total exports and domestic use
*         + pet(r,i,t) $ psFlag(r,i,"pet") \
*         --- otherwise, use supply price
*         + ps(r,i,t) $ psFlag(r,i,"ps") )] $ (pefob.range(r,i,rp,t) ne 0) \
*         --- case where export price is fixed
*         (used in pre-solves with only one country in model)
*         + pefob.l(r,i,rp,t) $ (pefob.range(r,i,rp,t) eq 0) }

```

As seen, beside considering export taxes, the macro reflects which offer price to use depending whether a CET is present or not.

The final macro used in the model considers if the product is treated a la Melitz or not:

```

*
* $$macro m_pefob(r,i,rp,t) [ m_pFirm(r,i,rp,t) * (1 + exptx(r,i,rp,t) + etax(r,i,t)) $ iMel(i) \
* + mm_pefob(r,i,rp,t) $ (not iMel(i))]

```

That macro is subsequently used to define based on the macro m\_pmcif the cif price, taking the transport margins and exchange rates into account:

```

*
* --- costs, insurance and freight price (c.i.f): add transport margins to f.o.b. price
*
* $$macro m_pmcif(r,i,rp,t) { [ m_pefob(r,i,rp,t)/lcu(r,t) \
* + sum(n $ angn(n,r,i,rp), angn(m,r,i,rp)*ptng(n,t)/lambdang(m,r,i,rp,t))*tnarg(r,i,rp,t)]*lcu(rp,t) }

```

### 9.3.6 Exploitation and flexible aggregation

The exploitation is based on views in the multi-dimensional cube defined in p\_results. Several such cubes representing different counter-factual runs (plus typically a benchmark) can be loaded simultaneously in the viewer. The views are defined in “CgeboxTables.xml”. As the set of sectors, regions and factors can differ from run to run, “report.gms” generates a XML file “generated.xml” which that information, e.g. for the products:

```

loop(i,
  put "<product>" /;
  put "  <itemName>", i.te(i),"</itemName>" /;
  put "  <key>", i.tl,"</key>" /;
  put "  <sel>all,products,sectorView</sel>" /;
  put "</product>" /;
);

```

That file is included into the view definitions at run time.

In order to allow the viewer to work with maps, a co-ordinate set of individual countries is stored in “GUI\world.zip”. The standard case of GGIG applications are fixed lists of regional code and matching coordinate set which link each regional id to a list of polygons. The flexible regional aggregation in GTAP required a more flexible approach. The XML-definition of a view can register for a region a list of components under the tag “<disagg>”:

```
<region>
  <itemName>Australia, New Zealand</itemName>
  <key>Oceania</key>
  <sel>all</sel>
  <disagg>[AUS,FJI,PYF,KIR,NCL,NZL,MNP,PNG,SLB,TON,UUT,WSM,]</disagg>
</region>
```

The individual code listed under “<disagg>” indicate the regional Ids used in the coordinate set. In the case of GTAP, these are codes for individual countries. The actual mapping between the aggregates used in the current model instance and the GTAP regions is read from an “agg”-file; the mapping between the GTAP regions code in the data base and the individual country ids is defined in “model\map\_regions.gms”, e.g.

```
rrmap("BRN","XSE") = YES;
rrmap("MMR","XSE") = YES;
```

That file should be currently set-up to work with GTAP version 8. Other versions require an update of the mappings, as the list of GTAP regions might differ.

A first view reports the meta-information on the model. It is here used as a first example for the XML based definition of views. Basically, each view can define filters in the different dimension. The filters end with “sel” and start with the logical name of a dimension. To give an example: where normally information on a sector or institution would be stored (fourth dimension, the “activityDim”), the label “tots” is used. It is chosen in the filter below based on a regex expression:

```
  <activitySel>tots_REGEX</activitySel>

<table>
  <theme>Model overview</theme>
  <name>Model properties</name>
  <regionSel>Wor_REGEX</regionSel>
  <regionText>hide</regionText>
  <productSel>.*_REGEX</productSel>
  <productText>hide</productText>
  <activitySel>tots_REGEX</activitySel>
  <activityText>hide</activityText>
  <dim5Sel>Meta</dim5Sel>
  <dim5Text>hide</dim5Text>
  <hideEmptyRows>YES</hideEmptyRows>

  <dim6Text>hide</dim6Text>

  <defview>Table</defview>
  <defpivot>ØP ØS</defpivot>

  <itemDim>dim6</itemDim>
    <item>
      <itemName>Dummy</itemName>
      <key>dom</key>
    </item>
</table>
```

Another example provides the table for intermediate demand. Here all products and sectors are selected; instead of a regex expression, the pre-defined lists found in “generated.xml” are used:

```

<table>
  <name>Intermediate demand</name>
  <theme>Markets</theme>
  <regionSel>all</regionSel>
  <productSel>products</productSel>
  <productText>Inputs</productText>
  <activitySel>sectors</activitySel>
  <activityText>sectors</activityText>
  <dim5Sel>all</dim5Sel>
  <dim6Text>Origin</dim6Text>
  <dim6Sel>origins</dim6Sel>

  <defview>Table</defview>
  <defpivot>0PSA</defpivot>
  <itemDim>dim6</itemDim>
  <item>
    <itemName>Total</itemName>
    <key>tot</key>
  </item>
  <item>
    <itemName>Imported</itemName>
    <key>Imp</key>
  </item>
  <item>
    <itemName>Domestic</itemName>
    <key>Dom</key>
  </item>
</table>

```

Details on how views can be defined can be found in GGIG programming guide, Britz 2010b.

Detail on simulated value of variables found in the Melitz module can be found in two tables under the “Trade group” as shown below:



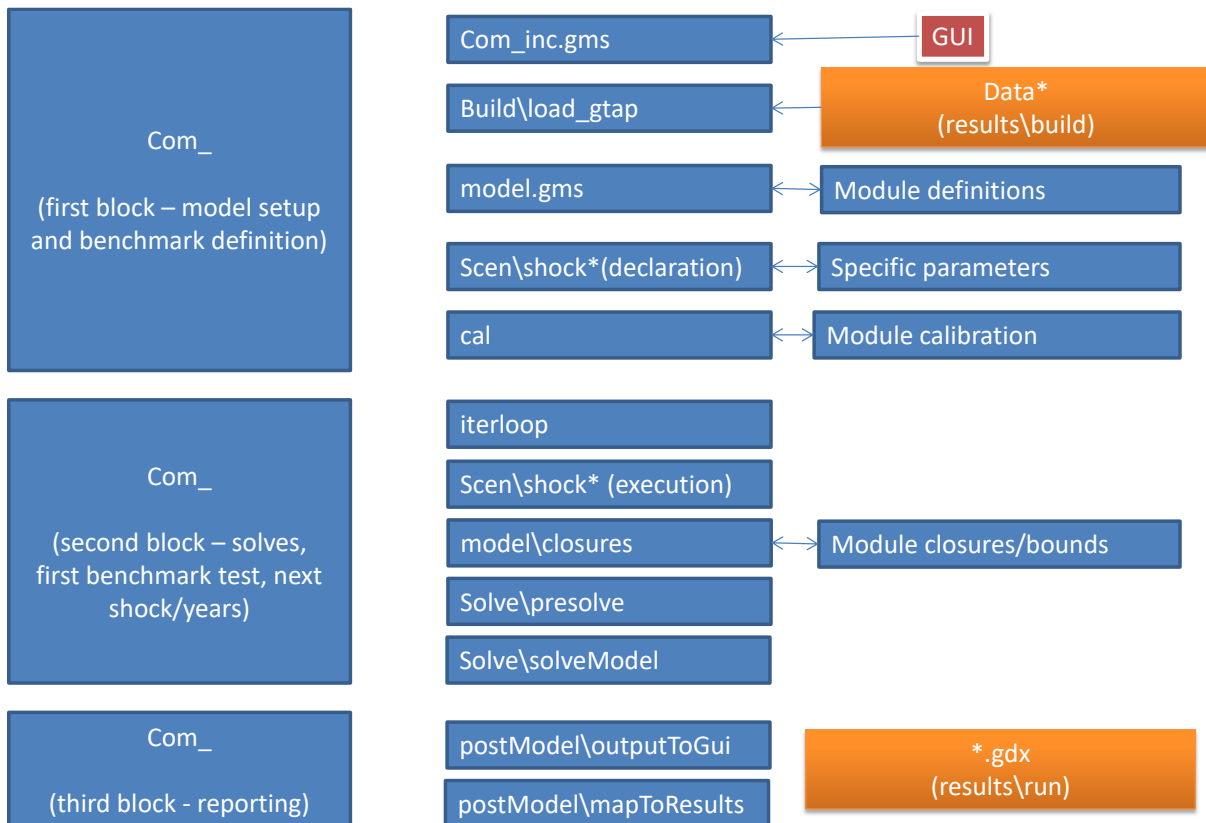
Melitz, overview [0]											
Products											Percenta
Total											Scenario
Years											noShock
ttp											
	All firms entered	All firms operating	Variable cost price	Fix cost price	General fixed cost	Bi-lateral fixed cost	Variable cost	General fixed cost per unit	Bi-lateral fixed cost per unit	Variable cost per unit	All firms entered
World			0,00	1,00	1,00	5217,96	1637,32	24002,60	0,14	0,04	0,66
			0,11%	-0,01%	-0,00%	0,00%	0,00%	-0,00%	0,00%	-0,00%	
European Union 28	1,00	0,00	1,00	1,00	1654,72	457,21	7611,71	0,15	0,04	0,69	1,00
	0,01%	0,41%	-0,02%	-0,01%	-0,01%	-0,01%	-0,01%	-0,02%	-0,02%	-0,02%	
Norway, Switzerland	1,00	0,03	1,00	1,00	63,72	15,01	293,13	0,16	0,04	0,72	1,00
	-0,02%	0,05%	-0,01%	-0,01%	-0,03%	-0,06%	-0,03%	-0,01%	-0,03%	-0,01%	
United States of America	1,00	0,00	1,00	1,00	929,68	315,31	4276,51	0,15	0,05	0,67	1,00
	0,01%	0,71%	0,00%	0,02%	0,03%	0,06%	0,03%	0,05%	0,07%	0,05%	
North America	1,00	0,02	1,00	1,00	105,15	35,58	483,70	0,15	0,05	0,68	1,00
	0,01%	-0,04%	0,00%	0,01%	0,03%	-0,01%	0,03%	0,02%	-0,02%	0,02%	
Mercosur	1,00	0,01	1,00	1,00	157,07	63,75	722,54	0,15	0,06	0,67	1,00
	0,00%	0,01%	-0,01%	-0,01%	-0,01%	-0,01%	-0,01%	-0,01%	-0,02%	-0,01%	
China	1,00	0,00	1,00	1,00	694,68	237,00	3195,53	0,13	0,04	0,59	1,00
	0,00%	-0,01%	-0,01%	-0,00%	-0,00%	-0,00%	-0,00%	-0,00%	-0,01%	-0,00%	
ASEAN 10	1,00	0,01	1,00	1,00	198,15	54,63	911,50	0,15	0,04	0,68	1,00
	0,01%	-0,01%	-0,01%	0,00%	0,01%	-0,03%	0,01%	-0,00%	-0,04%	-0,00%	
OtherOECD	1,00	0,00	1,00	1,00	694,27	231,10	3193,65	0,13	0,04	0,62	1,00
	0,00%	-0,01%	-0,00%	-0,00%	-0,00%	-0,01%	-0,00%	-0,00%	-0,01%	-0,01%	
EU Mediteranean Partners	1,00	0,01	1,00	1,00	123,51	38,28	568,13	0,15	0,05	0,70	1,00
	-0,00%	0,04%	-0,01%	-0,01%	-0,01%	-0,03%	-0,01%	-0,01%	-0,02%	-0,01%	
Low Income	1,00	0,10	1,00	1,00	19,82	5,97	91,17	0,16	0,05	0,72	1,00
	-0,00%	0,07%	-0,02%	-0,02%	-0,02%	-0,06%	-0,02%	-0,01%	-0,06%	-0,01%	
Rest of World	1,00	0,00	1,00	1,00	577,18	183,48	2655,02	0,15	0,05	0,69	1,00
	-0,00%	0,01%	-0,01%	-0,01%	-0,01%	-0,02%	-0,01%	-0,01%	-0,02%	-0,01%	
World			0,00	1,00	5217,96	1637,32	24002,60	0,14	0,04	0,66	
			0,11%	-0,01%	-0,00%	0,00%	0,00%	-0,00%	0,00%	-0,00%	
European Union 28	1,00	0,00	1,00	1,00	1654,72	457,21	7611,71	0,15	0,04	0,69	1,00
	0,01%	0,41%	-0,02%	-0,01%	-0,01%	-0,01%	-0,01%	-0,02%	-0,02%	-0,02%	

Meltz, trade matrix [0]

Items: Firm price, Products: Total, Years: shock, Percentage diff. to Scenarios: noShock

	Domestic sales	European Union 28	Norway, Switzerland	United States of America	North America	Mercosur	China	ASEAN 10	OtherOECD	EU Mediterranean Partners	Low Income	Rest of World	Domestic sales	Eu Un
World	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
European Union 28	-0,02%	0,02%	-0,01%	0,10%	0,01%	-0,01%	-0,01%	0,00%	-0,00%	-0,01%	-0,01%	-0,01%	-0,01%	1,00
Norway, Switzerland	-0,06%	-0,02%	1,00	0,49%	0,02%	-0,01%	-0,00%	0,00%	0,00%	-0,01%	-0,01%	-0,01%	-0,01%	1,00
United States of America	-0,01%	-0,03%	-0,00%	-0,00%	0,01%	-0,01%	-0,00%	0,01%	0,00%	-0,00%	-0,01%	-0,00%	-0,00%	1,00
North America	-0,04%	1,01	1,00	-0,03%	0,01%	-0,02%	-0,01%	-0,00%	-0,01%	-0,02%	-0,02%	-0,02%	-0,02%	1,00
Mercosur	0,01%	-0,03%	-0,03%	-0,01%	-0,02%	-0,02%	-0,01%	-0,00%	-0,01%	-0,02%	-0,02%	-0,02%	-0,01%	1,00
China	-0,01%	-0,06%	-0,02%	-0,02%	0,02%	-0,01%	-0,00%	-0,02%	-0,01%	-0,02%	0,02%	-0,01%	-0,01%	1,00
ASEAN 10	-0,01%	-0,02%	-0,02%	0,01%	0,01%	-0,01%	1,00	0,00%	-0,00%	-0,02%	-0,02%	-0,02%	-0,01%	1,00
OtherOECD	-0,00%	-0,03%	-0,02%	-0,02%	-0,00%	-0,02%	-0,01%	-0,00%	-0,01%	-0,02%	-0,02%	-0,02%	-0,01%	1,00
EU Mediterranean Partners	-0,00%	-0,03%	-0,02%	0,00%	0,02%	-0,01%	-0,01%	0,00%	-0,00%	-0,02%	-0,02%	-0,02%	-0,01%	1,00
Low Income	-0,01%	-0,02%	-0,02%	0,01%	0,01%	-0,01%	-0,00%	0,00%	-0,00%	-0,01%	-0,01%	-0,01%	-0,01%	1,00
Rest of World	-0,01%	-0,11%	-0,02%	0,00%	0,02%	-0,01%	1,00	0,00%	-0,00%	-0,00%	-0,00%	-0,00%	-0,00%	1,00
World	-0,01%	-0,03%	-0,02%	-0,00%	0,02%	-0,01%	-0,00%	0,00%	-0,00%	-0,01%	-0,01%	-0,01%	-0,01%	1,00
European Union 28	-0,03%	0,02%	-0,01%	0,10%	0,01%	-0,01%	-0,01%	0,00%	-0,00%	-0,01%	-0,01%	-0,01%	-0,01%	1,00
European Union 28	-0,06%	-0,02%	-0,01%	0,49%	0,02%	-0,01%	-0,00%	0,00%	0,00%	-0,01%	-0,01%	-0,01%	-0,01%	1,00
European Union 28	-0,06%	-0,02%	-0,01%	0,49%	0,02%	-0,01%	-0,00%	0,00%	0,00%	-0,01%	-0,01%	-0,01%	-0,01%	1,00

### 9.3.7 Main programs when solving a shock and integration of modules



The main driver program “com\_.gms” can be roughly broken into three major blocks:

- I. Benchmark set-up: That first block deals with loading the data and parameters and setting up the model such that it replicates the benchmark.
  - a. The information about what files, extensions and modules to use and further detail on the model run is read from “com\_inc.gms”, generated by the GUI. As a next step, the

“build\loadGTAP\_agg.gms” reads the global SAM and related parameters and initialized core sets (products, activities, agents, households etc.). These sets are either directly read from the data set or are defined by run specific file such as the household types present in the myGTAP module.

- b. Next, the equation system is set-up. The core equations for the standard GTAP model plus equations to host extensions are found in “model.gms” along with the necessary declarations of variables and parameters. According to the modules in use, additional equations are loaded in from module definitions which might replace equations from the standard GTAP model.
- c. Based on that information, the calibration step in “cal.gms” initializes all variables for the benchmark and calculates parameters such as the share parameters for CES functions. The different modules as the GTAP-Melitz extension have their own calibration programs which are included on demand.

## II. The second block comprises the model solve.

- a. First, all variables are copied from t-1 to t if not the benchmark test is run.
- b. Next, if not in benchmark check, the shock file is executed which hence can either use the results from t-1 (recursive dynamics) and/or benchmark results.
- c. Followed by that step is the definition of bounds for the variable which can include fixing depending on the closures. Again, module specific includes will initialize bounds e.g. for GTAP-AGR.
- d. Next, the current year is solved. If that is a benchmark solve, it will only build the equation system and does not allow the solver to improve the model (iterlim=0). That step should yield a total sum of infeasibilities  $< 1.E-4$ , otherwise, probably, some error in the benchmarking has occurred. In a more solve, pre-solve steps for the individual regions might be added. The model can attempt several solve and fix variables at their lower (tiny) bounds in CNS/NLP mode.

## III. Reporting

- a. The reporting block will in the simplest case only store all variable and parameters to a GDX.
- b. If output to the GUI is selected, the variables and parameters are mapped onto one multi-dimensional parameter p\_results which can be inspected by the GUI (see above), performed by “postmodel\mapToResults”. In that case, a XML file will be generated which comprises the list of products, activities etc. for use with the exploitation tools.