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

- Wolfgang Britz, Institute for Food Resource Economics –
- September 2017 -

(matches rev. 950 of https://svn1.agp.uni-bonn.de/svn/Gtap8InGams)

Content

Why an implementation in GAMS?

GTAP is certainly the far most widespread international data base for CGE modeling. The data base was so far 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 CGEs has certainly distinct advantages compared to less specialized languages. Some users might however prefer GAMS when working with GTAP. Indeed, outside the CGE community, GAMS seems much more common than GEMPACK as an Algebraic Modelling Language, e.g. in Agricultural Economics (Britz and Kallrath 2012). 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 (van der Mensbrugghe 2015). Intensive testing shows 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".

CGE models realized in GAMS sharing basic elements of the standard GTAP model while adding extensions in several directions are widely available , for instance GTAPinGAMS by Tom Rutherford, 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 in GAMS 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. It 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, GTAP-AEZ, GTAP-AGR, GTAP-E or GTAP-HET. Furthermore, there is 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 flexibility allow adjusting the model's structure to a specific application or to replicate main layout features of well-known models, see "Pre-configurations."

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. "Some comments for runGTAP and GEMPACK users" follow next. The structure of the core model is discussed equation by equation in the section "Basic model equations" (page 11ff), before the use of the Graphical User Interface is presented in the section "GUI" (page 54ff). It discusses step-by-step how to install the model, construct a data base, define the shock, run the model and analyze results. The following section "Modules" (page 93ff) discusses the methodological and technical details of the individual modules. Four sections complement the documentation, they discuss on result analysis ("Inspecting the results", page 160ff), on sensitivity analysis ("Sensitivity analysis", page 178ff), automated code documentation and equation analysis ("Getting an overview on the code" page 186ff) and on technical detail ("Technical documentation", page 190ff). Please note also that some training videos are available at http://www.ilr.uni-bonn.de/em/rsrch/cgebox/cgebox_e.htm#vid – they might differ in selected details.

Scalability, Modularity and Extensions as basic features of CGEBOX

As the standard GTAP model and many of its variants, the GAMS version 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. However, increasing the detail of the data base can quickly lead to models which are 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 “Rebalancing”, page 68ff) can systematically filter out tiny elements from the global SAM and afterwards rebalances it. Second, the GAMS code comprises a pre-solve algorithm which should speed up solution of larger models with more than 300.000 non-empty transactions in the global SAM. 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). 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 consumed. Other extensions are switched one 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 e.g. for foreign savings, private and government consumption. 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[1]. So far, implementations of GTAP-AGR, GTAP-E, GTAP-AEZ, myGTAP (see section “myGTAP module”, Page 93ff) and CO2 emissions accounting and taxation are available as additional modules, along with an implementation of the Melitz model (GTAP-HET, see section “GTAP-Melitz: Heterogenous firm module”, 133) and a MRIO split up of bi-lateral import demand by total intermediate consumption and the final demand agents, (see “MRIO extension”, page 119ff). 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 124ff).

Additionally, the production functions and factor markets for EU countries can be broken down to about 280 sub-regional so-called administrative NUTS2 regions. It should however be noted that these extensions are partly not set-up as a perfect replica of the GEMPACK implementations and not tested for replication of results obtained with the corresponding GEMPACK code. The code supports currently GTAP6 to GTAP9 data bases with matching land use data for 2007 and 2011 and allows using the GTAP9-Water data set. Examples of how these extensions and modules can be used to

[1] 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.

generate from the same code models with quite different features offer the pre-defined configurations (see section ”Pre-configurations”, page 89).

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”.

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, again drawing on ENVISAGE. A formal comparison to the features of the dynamic GTAP model Gdyn is pending. 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 these commodities are fixed along with total income and the model is solved for the remaining endogenous commodity markets.

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 in a modular fashion additional modules. It should be noted here that the module 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.. Details on these modules can be found in the section “Modules”.

Data preparation (see section “Data base generation”) for the model is based on GTAPAGG output to which satellite accounts such as for land use and non-CO2 emissions can be added, along with user provided additional data and information to e.g. split sectors and commodities in the SAM or to aggregate commodities or sectors to yield non-diagonal make relation. The modeling platform is supported by a Graphical User Interface, see section “GUI” which also details how to install the system.

Technically, the model is set-up a constraints 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.

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

Module	Status	Remarks
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
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 functions for final demand	Found in myGTAP	Parameters derived from CDE parameterization
GTAP-AEZ	Operational, not tested for exact replication	With land supply elasticities for natural land cover, differentiated by land cover
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
GTAP-Melitz	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 Krugmann specification.
GTAP-MRIO	Operational	Differentiation of bi-lateral import demand by total intermediate demand and each final demand agent
myGTAP	Operational, not tested for exact replication	Removes the regional household, support multiple private households
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,	Might avoids numerical problems with tiny shares

Module	Status	Remarks
	found in GLOBE	
Tariff lines	Operational	Allows a CET/CES dis-aggregation of selected bi-lateral trade links
Capital vintages	Differentiation between depreciation capital stock (not mobile) and gross investment (mobile), capital stock can be rendered endogenous in comparative static mode	Draws on similar mechanism used in recursive-dynamic CGEs which differentiate vintage from new capital
NUTS2 break down for European countries	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	Could be done 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
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	Operational, not tested for compliance with GDyn	baseScenario\recDyn comprises a driver program with various features of interest
Partial Equilibrium closure	Operational	Solves only one or some commodity markets and all factor markets, regional or household income exogenous
CO2 emissions	Operational	Can be combined with taxation or CO2 trading permits
Non-CO2 emissions	Operational	Only post model reporting

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. That is not only due to differences between GAMS and GEMPACK, but also due to different approaches 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, the concept of GTAP and their experience of having worked with a modelling language. They will face 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 working with the system along the lines of 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 constants in a model and can never be changed by the solver and variables. While variables in GAMS can be technically turned into constant during a model solve by fixing them, parameters hence cannot turned in an endogenous variables. That 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 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 afterwards. That must be especially reflected when the user develops its own shock files.

Basic model equations

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,nat1	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 avtive)
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)

Set name	Description
fnm	non-mobile 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 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 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 which serves as flags are used. 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 composite added 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

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 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 α_i defined at the benchmark is multiplied with the total quantity *y* to distribute. That 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 price $\tilde{p}$ and the price p_i, exponent the substitution elasticity σ. Note that settings the substitution elasticity to zero yields the Leontief case where the original shares stay constant. A shifter variable λ can be used to update preferences or the cost structure to reflect non-Hisks neutral technical progress. The reader should not 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 price $\tilde{p}$ is to use value exhaustion, i.e. introducing an equation $\sum_i x_i p_i = y\tilde{p}$. But note that this exhaustion equation requires the individual demands x_i being known, which 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 CGEs 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. Once the average price is solved for, the other n equations 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 CEGBox 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.

Overview on the supply side

The following graphic depicts the quantity and price variables as well as the substitution and transformation elasticities used on the supply side. The bottom part is defined for the production activities *a* with total output denoted with *xp* and related price *px*. It is composed of a value added composite *va* and an intermediate demand composite *nd*. The value added composite combines primary factor *f* and potentially technology sub-nests *tNest*. The intermediate demand composite *nd* combines intermediates defined as Armington demands *xa* 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 *xp*.

The output of the activities *xp* 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.

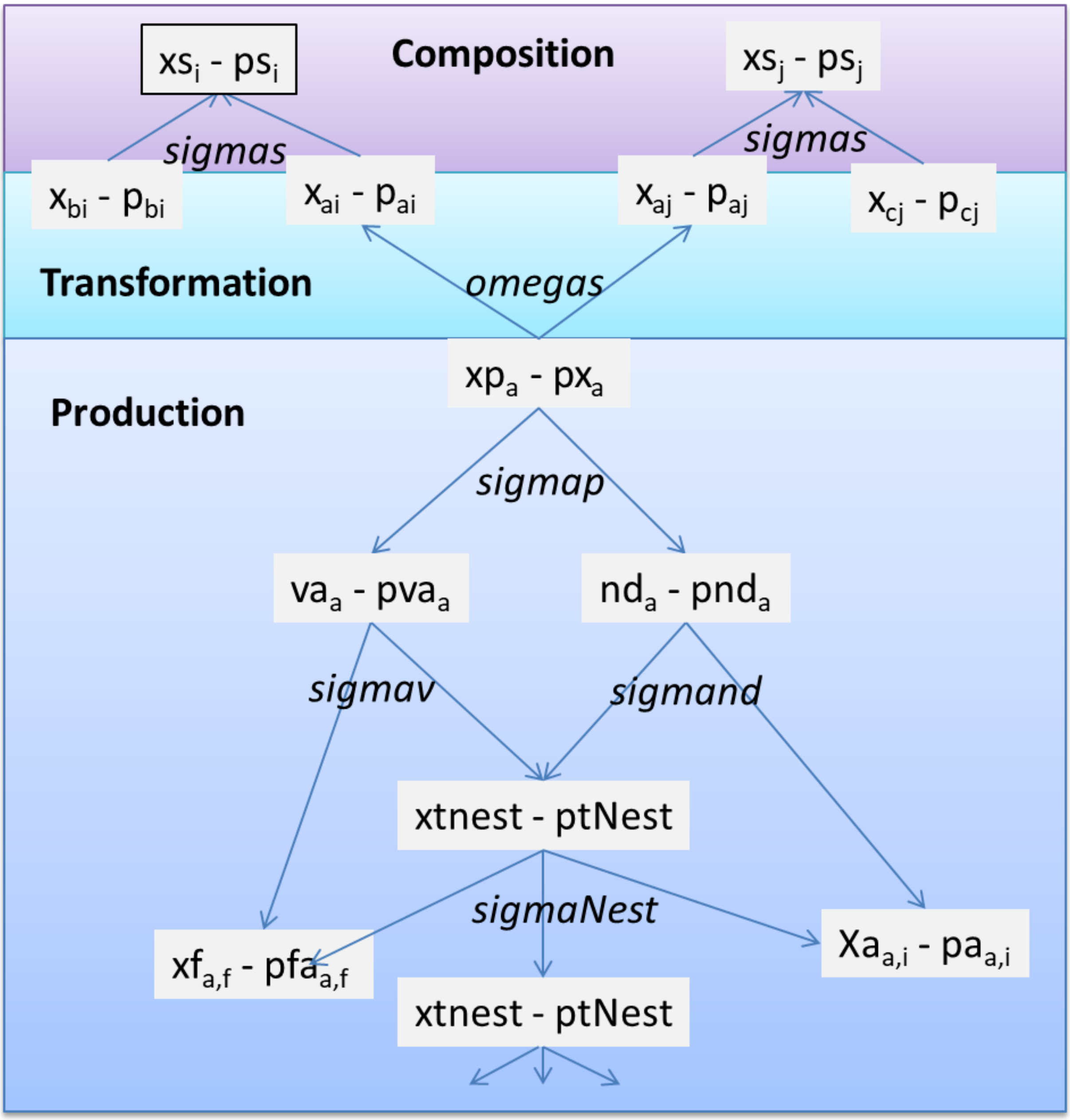


Figure 1: Overview on production function nesting

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 *sigmap*. The production frontier can be shifted with the variable *axp*. The top nest is represented by its dual price aggregator in the equation *pxeq*. That equation considers three cases which are shown below: (1) *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 *sigmap* is unity with a different dual price aggregator and (3) the Leontief case with *sigmap* equal to zero. 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 *and* and *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*.

```
pxeq(rs(r),aIn(a),ts(t)) $ (xpFlag(r,a) $ (not diag(a)) $ (not disr(r)))..
   axp(r,a,t)*mm_px(r,a,t) =e=
*
*        -- Non Leontief case with substitution between VA and ND
*
          (  (   and(r,a,t)*(pnd(r,a,t)/lambdand(r,a,t))**(1-sigmap(r,a))
               + ava(r,a,t)*(pva(r,a,t)/lambdava(r,a,t))**(1-sigmap(r,a))) ** (1/(1-sigmap(r,a)))  ) $ (sigmap(r,a) $ (sigmaP(r,a) ne 1))

*
*        --- CD case with substitution between VA and ND, sigmap=1
*
       +  (       axpCD(r,a,t)*   ([(pnd(r,a,t)/(and(r,a,t)*lambdand(r,a,t)))**and(r,a,t)] $ and(r,a,t) + 1 $ (not and(r,a,t)))
                                 *([(pva(r,a,t)/(ava(r,a,t)*lambdava(r,a,t)))**ava(r,a,t)] $ ava(r,a,t) + 1 $ (not ava(r,a,t))) ) $ (sigmap(r,a) eq 1)
*
*        -- Leontief case (only introduced to ease work for solver), as in standard GTAP model
*
      +  (
            (   and(r,a,t)*(pnd(r,a,t)/lambdand(r,a,t))
              + ava(r,a,t)*(pva(r,a,t)/lambdava(r,a,t)) ))  $ (not sigmap(r,a))
      ;
```

The *mm_px* macro is in the usual case equal to the *m_xp* macro which is shown 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*:

```
$$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))
```

The demand for the **value added** composite *va* is defined in the equation *vaeq*. It comprises the same symbols as shown above in the top level unit cost definition. Note that the equation treats the Leontief case where *sigmap* is zero differently by removing the prices from the equation which can speed up solution.

```
*  Demand for aggregate value added -- VADEMAND (1262)

vaeq(rs(r),aIn(a),ts(t)) $ vaFlag(r,a) ..
   va(r,a,t)/va.scale(r,a,t) =e= ava(r,a,t) * m_xp(r,a,t)/va.scale(r,a,t)
                       * [   {(mm_px(r,a,t)/pva(r,a,t))**sigmap(r,a)} $ sigmap(r,a) + 1 $ (not sigmap(r,a)) ]
                       *  (axp(r,a,t)*lambdava(r,a,t))**(sigmap(r,a)-1) ;
```

Note that the *vaeq* equation is scaled with the scale field of the value added demand *va.scale*. Scaling factors are present basically in all equation 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 driven the value added demand is defined in the macro *m_xp*. It uses directly the commodity supply *xs* in case of 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.

```
$$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) )
```

The equation *ndeq*, identically structured as the *vaeq* equation above, drives the demand for the **intermediate demand** composite:

```
ndeq(rs(r),aIn(a),ts(t)) $ ndFlag(r,a) ..
   nd(r,a,t)/nd.scale(r,a,t) =e= and(r,a,t) * m_xp(r,a,t)/nd.scale(r,a,t)
                         * [   {(mm_px(r,a,t)/pnd(r,a,t))**sigmap(r,a)} $ sigmap(r,a) + 1 $ (not sigmap(r,a)) ]
                            *   (axp(r,a,t)*lambdand(r,a,t))**(sigmap(r,a)-1) ;
```

The demand for **primary factors** *xf* by each activity depends on a shifter variable *lambdaf*, the share parameter *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 *sigmav*:

```
xfeq(rs(r),f,aIn(a),ts(t)) $ (xfFlag(r,f,a) $ (af(r,f,a,t) or sum(tNest_f_a(tNest,f,a), afNest(r,f,a,tNest,t))) ) ..
   xf(r,f,a,t)/xf.scale(r,f,a,t) =e= af(r,f,a,t)*va(r,a,t)/xf.scale(r,f,a,t)*(pva(r,a,t)/m_pfa(r,f,a,t))**sigmav(r,a)
                 * m_lambdaf(r,f,a,t)**(sigmav(r,a)-1)
*
* --- demand for factor inside technology nest
*
  + sum(tNest_f_a(tNest,f,a) $ afNest(r,f,a,tNest,t),
*
*    --- factor cost share times demand for technology nest
*
        afNest(r,f,a,tNest,t)*xtNest(r,tNest,a,t)/xf.scale(r,f,a,t)
*
*   --- relative price impact (in case sigmndNest is not 0)
*
     * [ {  (pTNest(r,tNest,a,t)/m_pfa(r,f,a,t))**sigmaNest(r,tNest,a)    } $ sigmaNest(r,tNest,a)
                                                                         +  1 $ (not sigmaNest(r,tNest,a)) ]
                  * m_lambdaf(r,f,a,t)**(sigmaNest(r,tNest,a)-1));
```

The second part of the equation is not part of the standard model and only active if technology nests are used and is described in the section “Flexible nesting”. It comprises the same elements: share parameters insides the nests *afNest*, the composite demand for the nest *xtNest*, the price relation which now uses the average price of the nest *pTNest* and the substitution elasticity *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 sahres.

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

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

```
pvaeq(rs(r),aIn(a),ts(t)) $ vaFlag(r,a) ..
   pva(r,a,t) =e=
*
*     --- CES case (or Leontief)
*
      [ (
*
*         --- contribution of factors to top VA nests
*
        sum(f $ af(r,f,a,t), af(r,f,a,t)*(m_pfa(r,f,a,t)/m_lambdaf(r,f,a,t))**(1-sigmav(r,a)))
*
*         --- contribution of technology nests to top VA nests (not part of standard GTAP model)
*
      + sum(tNest_n_a("VA",tNest,a), aTNest(r,tNest,a,t)*pTNest(r,tNest,a,t)**(1-sigmav(r,a)))

        )** (1/(1-sigmav(r,a))) ] $ (sigmav(r,a) ne 1)

*
*     --- CD case with sigmaV == 1
*
    + [  axVACD(r,a,t)
*
*         --- contribution of factors to top VA nests
*
     *   prod(f $ af(r,f,a,t),(m_pfa(r,f,a,t)/(m_lambdaf(r,f,a,t)*af(r,f,a,t)))**af(r,f,a,t))
*
*         --- contribution of technology nests to top VA nests (not part of standard GTAP model)
*
      * prod(tNest_n_a("VA",tNest,a) $ atNest(r,tNest,a,t), (pTNest(r,tNest,a,t)/aTNest(r,tNest,a,t))**aTNest(r,tNest,a,t))

         ] $ (sigmav(r,a) eq 1);
```

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 *pdneq* defines the **intermediate composite price** *pnd*. It is driven by the input coefficients *io* and their activity specific price *paint* defined via a macro and individual technology shifters l*ambdaio* again captured by a macro. Note that the coefficients *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 substation elasticity *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 here again we consider the cases where the intermediate demand is driven by the intermediate composite (standard model) and/or by technology nests.

```
pndEq(rs(r),aIn(a),ts(t)) $ ndFlag(r,a) ..

   pnd(r,a,t) =e=
*
*         --- CES case
*
          [ (
*
*          --- aggregate over all intermediate inputs not assigned to a technology tests
*
              sum(i $ io(r,i,a,t),
                         io(r,i,a,t)*(m_paint(r,i,a,t)/m_lambdaio(r,i,a,t))**(1-sigmand(r,a)))
*
*          --- add demand for technology nests linked into the top ND undle
*
           + sum(tNest_n_a("ND",tNest,a) $ atNest(r,tNest,a,t),
               atNest(r,tNest,a,t) * ptNest(r,tNest,a,t)**(1-sigmand(r,a)))

             )**(1/(1-sigmand(r,a)))

          ] $ (sigmand(r,a) $ (sigmand(r,a) ne 1))

*
*         --- CD case with sigmaND == 1
*
      +   [  axNdCD(r,a,t) *
*
*          --- aggregate over all intermediate inputs not assigned to a technology tests
*
             {prod(i $ io(r,i,a,t),
                         (m_paint(r,i,a,t)/(io(r,i,a,t)*m_lambdaio(r,i,a,t)))**io(r,i,a,t))
               $ sum(i $ io(r,i,a,t), 1) + 1 $ (not sum(i $ io(r,i,a,t), 1)) }
*
*          --- add demand for technology nests linked into the top ND undle
*
           *  {prod(tNest_n_a("ND",tNest,a),  (ptNest(r,tNest,a,t)/atNest(r,tNest,a,t))**atNest(r,tNest,a,t))
                 $ sum(tNest_n_a("ND",tNest,a), atNest(r,tNest,a,t))
               + 1 $ (not sum(tNest_n_a("ND",tNest,a), atNest(r,tNest,a,t))) }

          ] $ (sigmand(r,a) eq 1)
*

*
*         --- Leontief case, as in standard GTAP model
*
       +  [
*
*          --- aggregate over all intermediate inputs not assigned to a io-group
*
          sum(i $ io(r,i,a,t),
                     io(r,i,a,t)*(m_paint(r,i,a,t)/m_lambdaio(r,i,a,t)))
*
*          --- add aggregate intermediate group demands (not part of standard GTAP Model)
*
           + sum(tNest_n_a("ND",tNest,a), atNest(r,tNest,a,t) * ptNest(r,tNest,a,t)) $ card(tNest)

          ] $ (not sigmand(r,a))
      ;
```

The dual price aggregator equation *ptNestEq* for technology nests is depicted below. It considers the possible components: intermediates with the related share parameter *ioNest*, primary factors with their share parameter *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.

```
ptNestEq(rs(r),tNest,aIn(a),ts(t)) $ (xpFlag(r,a) $ ( (atNest(r,tNest,a,t) gt 0)
                                   or sum(tNest_f_a(tNest,f,a), afNest(r,f,a,tNest,t))
                                   or sum(tNest_i_a(tNest,i,a), ioNest(r,i,a,tNest,t)) )) ..

   ptNest(r,tNest,a,t) =e=
   {[
*
*  --- aggregate over intermediate inputs assigned to technology nest
*
       sum(tNest_i_a(tNest,i,a) $ ioNest(r,i,a,tNest,t),
                              ioNest(r,i,a,tNest,t)
                                * (m_paint(r,i,a,t)/m_lambdaio(r,i,a,t))**(1-sigmaNest(r,tNest,a)))
*
*  --- aggregate over factors  assigned to technology nests
*
     + sum(tNest_f_a(tNest,f,a) $ afNest(r,f,a,tNest,t),
                              afNest(r,f,a,tNest,t)
                                * (m_pfa(r,f,a,t)/m_lambdaf(r,f,a,t))**(1-sigmaNest(r,tNest,a)))
*
*  --- aggregate over sub technology nests assigned to that technology nests
*
     + sum(tNest_n_a(tNest,tNest1,a) $ atNest(r,tNest1,a,t),
                              atNest(r,tNest1,a,t)
                                * ptNest(r,tNest1,a,t)**(1-sigmaNest(r,tNest,a)))
                                                                     ]**(1/(1-sigmaNest(r,tNest,a)))
  }
   $ (sigmaNest(r,tNest,a) ne 1)
```

```
*
*  ---- CES case with sigmaNest == 1 (shifter is missing ...)
*
  + [
*
*  --- aggregate over intermediate inputs assigned to technology nest
*
       prod(tNest_i_a(tNest,i,a) $ ioNest(r,i,a,tNest,t),
                                  (m_paint(r,i,a,t)/(ioNest(r,i,a,tNest,t)*m_lambdaio(r,i,a,t)))**ioNest(r,i,a,tNest,t))
*
*  --- aggregate over factors  assigned to technology nests
*
     * prod(tNest_f_a(tNest,f,a) $ afNest(r,f,a,tNest,t),
                                  (m_pfa(r,f,a,t)/(afNest(r,f,a,tNest,t)*m_lambdaf(r,f,a,t)))**afNest(r,f,a,tNest,t))
*
*  --- aggregate over sub technology nests assigned to that technology nests
*
     * prod(tNest_n_a(tNest,tNest1,a) $ atNest(r,tNest1,a,t),
                                  (ptNest(r,tNest1,a,t)/atNest(r,tNest1,a,t))**atNest(r,tNest1,a,t))/axTNest(r,tNest,a,t)
   ] $ (sigmaNest(r,tNest,a) eq 1);

  ;
```

The demand for technology nests *xtNest* is depicted by *xtNesteq* 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 (*sigmaNd*, *sigmav* or *sigmaNest*). In all cases, the share parameter is denoted with *atNest* and the related price with *ptNest*. Euqally, 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.

```
xtNestEq(rs(r),tNest,aIn(a),ts(t)) $ (xpFlag(r,a) $ atNest(r,tNest,a,t) $ (not  tNest_n_a("Top",tNest,a))) ..

    xtNest(r,tNest,a,t)/xtNest.scale(r,tNest,a,t) =e=
*
*        --- nest is part of top level intermediate demand
*
      +  ( atNest(r,tNest,a,t) * nd(r,a,t)/xtNest.scale(r,tNest,a,t)
                          * [ {(pnd(r,a,t)/ptNest(r,tNest,a,t))**sigmaNd(r,a)} $ sigmaNd(r,a) + 1 $ (not sigmaNd(r,a)) ] ) $ tNest_n_a("ND",tNest,a)
*
*        --- nest is part of top level VA demand
*
       + ( atNest(r,tNest,a,t) * va(r,a,t)/xtNest.scale(r,tNest,a,t)
                          * [ {(pva(r,a,t)/ptNest(r,tNest,a,t))**sigmav(r,a)} $ sigmav(r,a) + 1 $ (not sigmav(r,a)) ] ) $ tNest_n_a("VA",tNest,a)
*
*        --- nest is part of other nests
*
       + sum(tNest_n_a(tNest1,tNest,a), atNest(r,tNest,a,t) * xtNest(r,tNest1,a,t)/xtNest.scale(r,tNest,a,t)
                          * [ {(ptNest(r,tNest1,a,t)/ptNest(r,tNest,a,t))**sigmaNest(r,tNest1,a)} $ sigmaNest(r,tNest1,a) + 1 $ (not sigmaNest(r,tNest1,a)) ]);
```

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 *xeq*. That case is shown when the flag *diag(a)* is not unity, i.e. a not diagonal activity. In that case, the *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 teach product *i* termed *p* in relation to average per activity prices found in the macro *m_pp*. In case of infinite transformation, the prices have to be identical:

```
xeq(rsNat(rNat),aIn(a),i,ts(t)) $ (xFlag(rNat,a,i) $ (not diag(a))  )..
   0 =e= (x(rNat,a,i,t)/xp.scale(rNat,a,t)  - gx(rNat,a,i,t)*xp(rNat,a,t)/xp.scale(rNat,a,t) *(p(rNat,a,i,t)/m_pp(rNat,a,t))**omegas(rNat,a))   $ (omegas(rNat,a) ne inf)
     +  (p(rNat,a,i,t) $ (sigmas(rNat,i) ne inf) + ps(rNat,i,t) $ (sigmas(rNat,i) eq inf) -  (1 + prdtx(rNat,a,t))*m_px(rNat,a,t)  )     $ (omegas(rNat,a) eq inf)
     ;
```

The related equation *xpeq* 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 *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 *sigmas*.

The *marco m_pp* which fined the activity specific producer price charges the production tax prdtx on the unit costs *m_px*:

```
$$macro m_pp(r,a,t)   ((1 + prdtx(r,a,t))*m_px(r,a,t))
```

The related equations *peq* and *pseq* to aggregate 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 *peq* defines for the finite case the share of total supply *xs* demanded from activity *a* depicted by *x* based on the share parameter ax, the price relation and the substitution elasticity *sigmas*:

```
peq(rsNat(rNat),aIn(a),i,ts(t)) $ (xFlag(rNat,a,i) $ (not diag(a)) )..
   0 =e= (x(rNat,a,i,t)/xp.scale(rNat,a,t)   - ax(rNat,a,i,t)*xs(rNat,i,t)/xp.scale(rNat,a,t) *(ps(rNat,i,t)/p(rNat,a,i,t))**sigmas(rNat,i)) $ (sigmas(rNat,i) ne inf)
      +  (p(rNat,a,i,t) - ps(rNat,i,t)) $ (sigmas(rNat,i) eq inf)
      ;

pseq(rsNat(rNat),i,ts(t)) $ ( (xs.range(rNat,i,t) ne 0) $ xsFlag(rNat,i) $ (not diag(i))  )..
   0 =e=
       + (xs(rNat,i,t)/xs.scale(rNat,i,t) - sum(a$xFlag(rNat,a,i), x(rNat,a,i,t))/xs.scale(rNat,i,t)) $  (sigmas(rNat,i) eq inf)
       + (ps(rNat,i,t)                  -  sum(a$xFlag(rNat,a,i), ax(rNat,a,i,t)*p(rNat,a,i,t)**(1-sigmas(rNat,i)))**(1/(1-sigmas(rNat,i)))) $  (sigmas(rNat,i) ne inf)
```

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

Factor markets

The **supply of fully mobile or sluggish factors** x*ft* at national level *rsNat* is depicted by the equation *xfteq* in case where the factor supply is not fixed (.range eq 0). It is driven by the factor price *pft* relative to the price of aggregate domestic absorption *pabs* and the factor supply elasticity *etaf*. If *etaf* is zero, the price dependent part becomes a constant of unity and *xft* is fixed to the constant *aft*. 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.

```
xfteq(rsNat(rNat),fm,ts(t) ) $ (xftEqFlag(rNat,fm) $ (xft.range(rNat,fm,t) ne 0) $ (Not sameas(fm,"newCap"))) ..
   xft(rNat,fm,t)/xft.scale(rNat,fm,t) =e= (aft(rNat,fm,t)*(pft(rNat,fm,t)/sum(r_r(rNat,rp),pabs(rp,t)))**etaf(rNat,fm))/xft.scale(rNat,fm,t);
```

The related **economy wide average factor price** *pft* is defined by the equation *pfteq* which distinguished the sluggish case with a dual price aggregator (first block) and the fully mobile case where the equation ensures market clearing (second block):

```
pfteq(rs(r),fm,ts(t)) $ (xftEqFlag(r,fm) and pft.range(r,fm,t) and (not disr(r))) ..

   0 =e=
*
*        --- sluggish factor mobility, dual price aggregator
*
        (pft(r,fm,t)
*
*            --- factors directly linked into top CET nest
*
            -   sum(a $ (xfFlag(r,fm,a) $ (not sum(fNest_a_f(fNest,a,f),1))),
                       gf(r,fm,a,t)*pf(r,fm,a,t)**(1+omegaf(r,fm)))**(1/(1+omegaf(r,fm)))
*
*            --- subnests linked into top CET nest
*
            -   sum(fNest_n_f("xft",fNest,fm) $ gfNest(r,fm,fNest,t),
                     gfNest(r,fm,fNest,t)*pfNest(r,fm,fNest,t)**(1+omegaf(r,fm)))**(1/(1+omegaf(r,fm)))

            ) $ (omegaf(r,fm) ne inf)
*
*        --- fully mobile factors, uniform prices => phyiscal aggregation
*
      +  (xft(r,fm,t)/xft.scale(r,fm,t)
                      - sum(a $ (xfFlag(r,fm,a) $ (not sum(fNest_a_f(fNest,a,fm),1))), xf(r,fm,a,t))/xft.scale(r,fm,t)
                      - sum(fNest_n_f("xft",fNest,fm) $ gfNest(r,fm,fNest,t),          xfNest(r,fm,fNest,t))/xft.scale(r,fm,t)

         ) $ (omegaf(r,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 (`$ (omegaf(r,fm) 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 *pfeq* shown below. It considers five different cases. The first case considers sluggish factor supply where the factor is part of the value added nest. The usual CET distribution logic applies: the supply depends on the share parameter *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 *omegaf*. The second case where the factor is part of factor supply nest under sluggish supply is identically is structured, with the difference that the average price is now nest specific, i.e. *pfNest* as is the transformation elasticity *omegafNest*.

Next we have the two cases with fully mobile supply: either in case of economy wide full mobility or fully 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 *etaff* is zero such that the factor demand *xf* must be equal to the given parameter *gf*.

```
pfeq(rs(r),f,aIn(A),ts(t)) $ (xfEqFlag(r,f,a) $ pf.range(r,f,a,t) $ (not disr(r)) )  ..
   0 =e=
*
*       --- "sluggish" factor supply based on CET function, factor is part of top level nest
*
          (xf(r,f,a,t)/xf.scale(r,f,a,t)
             -  gf(r,f,a,t)*xft(r,f,t)/xf.scale(r,f,a,t)  *  (pf(r,f,a,t)/pft(r,f,t))**omegaf(r,f)

             )  $ (fm(f) $ (omegaf(r,f) ne inf) $ (not sum(fNest_a_f(fNest,a,f),1)) )

*
*       --- "sluggish" factor supply based on CET function, factor is part of sub-nest
*
        + (xf(r,f,a,t)/xf.scale(r,f,a,t)
             - sum(fNest_a_f(fNest,a,f),
                gf(r,f,a,t)*xfNest(r,f,fNest,t)/xf.scale(r,f,a,t)  *  (pf(r,f,a,t)/pfNest(r,f,fNest,t))**omegafNest(r,fNest,f))

             )  $ (fm(f) $ sum(fNest_a_f(fNest,a,f) $ (omegafNest(r,fNest,f) ne inf),1))
*
*       --- fully mobile factor: prices are equal across sectors
*
*           ... factor is linked into top level nest
       +  (pf(r,f,a,t) - pft(r,f,t))                            $ ((fm(f) $ (omegaf(r,f) eq inf)) $ (not sum(fNest_a_f(fNest,a,f),1)))
*           ... factor is linked into sub-nest
       +  (pf(r,f,a,t) - sum(fNest_a_f(fNest,a,f), pfNest(r,f,fNest,t)))  $ ((fm(f) $ sum(fNest_a_f(fNest,a,f) $ (omegafNest(r,fNest,f) eq inf),1)))
*
*       --- immobile factors
*
       +  (xf(r,f,a,t)/xf.scale(r,f,a,t)
            - gf(r,f,a,t)/xf.scale(r,f,a,t)*[ {(pf(r,f,a,t)/sum(r_r(r,rNat),pabs(rNat,t)))**etaff(r,f,a)} $ etaff(r,f,a) + 1 $ (not etaff(r,f,a))] ) $(fnm(f)) ;
```

The **agent specific factor prices tax inclusive** are defined via the equation *pfaeq* and the macro m_pfa:

```
pfaeq(rs(r),f,aIn(a),ts(t)) $ xfFlag(r,f,a) ..

   pfa(r,f,a,t) =E= m_pfa(r,f,a,t);
```

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 *fcctxShift*:

```
*
* --- factor price of agents
*
  $$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 (omegaf(r,f) ne inf)) \
*
*                                 --- fully mobile factor: all sector face a unifrom price before taxes
*
                                  + pft(r,f,t)  $ (fm(f) $ (omegaf(r,f) eq inf))) \
*
*                                 --- add ad-valorem factor taxes
*
                                  * sum(r_r(r,rNat), ( 1 + fctts(rNat,f,a,t) + fcttx(rNat,f,a,t) + fcttxShift(rNat,t) $ fTaxShift(f))) )
```

The "Flexible nesting" approach allows introducing factor supply nests which can also be linked into other factor supply nests. The equation *pfNestEq* 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 dual price aggregator, taking the share parameters (*gf* for factors and *gfNest* for sub-nest), the prices (*pf* for factors and *pfNest* for sub-nests) and the transformation elasticity *omegaFNest* into account. In case of infinite transformation handled by the second block, the price is indirectly defined from the adding up-condition of the factor quantities.

```
pfNestEq(rs(r),fm,fNest,ts(t)) $ gfNest(r,fm,fNest,t) ..

   0 =e=
*
*        --- sluggish factor mobility, dual price aggregator for sub-nests
*
        (pfNest(r,fm,fNest,t)
*
*            --- factors linked into sub nest
*
            - [    sum(fNest_a_f(fNest,a,fm) $ gf(r,fm,a,t),
                       gf(r,fm,a,t)*pf(r,fm,a,t)**(1+omegafNest(r,fNest,fm))

*
*            --- subnests linked into sub nest
*
                +  sum(fNest_n_f(fNest,fNest1,fm) $ gfNest(r,fm,fNest1,t),
                       gfNest(r,fm,fNest1,t)*pfNest(r,fm,fNest1,t)**(1+omegafNest(r,fNest,fm)))
              ]**(1/(1+omegafNest(r,fNest,fm)))

            ) $ (omegafNest(r,fNest,fm) ne inf)
*
*        --- fully mobile factors, uniform prices => physical aggregation
*
      +  (xfNest(r,fm,fNest,t)/xft.scale(r,fm,t)
                    - sum(fNest_a_f(fNest,a,fm) $ xfFlag(r,fm,a),  xf(r,fm,a,t))/xft.scale(r,fm,t)
                    - sum(fNest_n_f(fNest,fNest1,f),               xfNest(r,fm,fNest1,t))/xft.scale(r,fm,t)

         ) $ (omegafNest(r,fNest,fm) eq inf)
      ;
```

The total factor supply to such a nest *xfNest* as defined in the *xfNestEq* 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*.

```
xfNestEq(rs(r),fm,fNest,ts(t)) $ gfNest(r,fm,fNest,t) ..

    0 =e=
*
*      --- "sluggish" factor supply from top level nest (xft,pft) to current nest
*
         ( xfNest(r,fm,fNest,t)/xft.scale(r,fm,t)
           - gfNest(r,fm,fNest,t) * xft(r,fm,t)/xft.scale(r,fm,t)
               * (pfNest(r,fm,fNest,t)/pft(r,fm,t))**omegaf(r,fm))
                 $ ( (omegaf(r,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(r,fm,fNest,t)/xft.scale(r,fm,t)
             - gfNest(r,fm,fNest,t) * xfNest(r,fm,fNest1,t)/xft.scale(r,fm,t)
                 * (pfNest(r,fm,fNest,t)/pfNest(r,fm,fNest1,t))**omegafNest(r,fNest1,fm))
                   $ ( sum(fNest_n_f(fNest1,fNest,fm) $ (omegafNest(r,fNest1,fm) ne inf),1))
*
*      --- non  "sluggish" factor supply from top level nest (xft,pft) to current nest
*
       + ( pfNest(r,fm,fNest,t) - pft(r,fm,t) ) $ ( (omegaf(r,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) $ (omegafNest(r,fNest1,fm) eq inf),
            pfNest(r,fm,fNest,t) - pfNest(r,fm,fNest1,t) )
   ;
```

The factor supply form nation to sub-regions is described in the section “Integration into the modeling framework” of the chapter “Sub-regional dis-aggregation of production and factor markets in CGEBox”. Note also that the “GTAP-AEZ” module (page 158ff) will introduce land transformation at the level of Agro-Ecological Zones.

Income generation and distribution

And overview on income generation and distribution under the regional household approach is depicted 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 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 or LES demand system can be used to distribute private household expenditure *yc*.

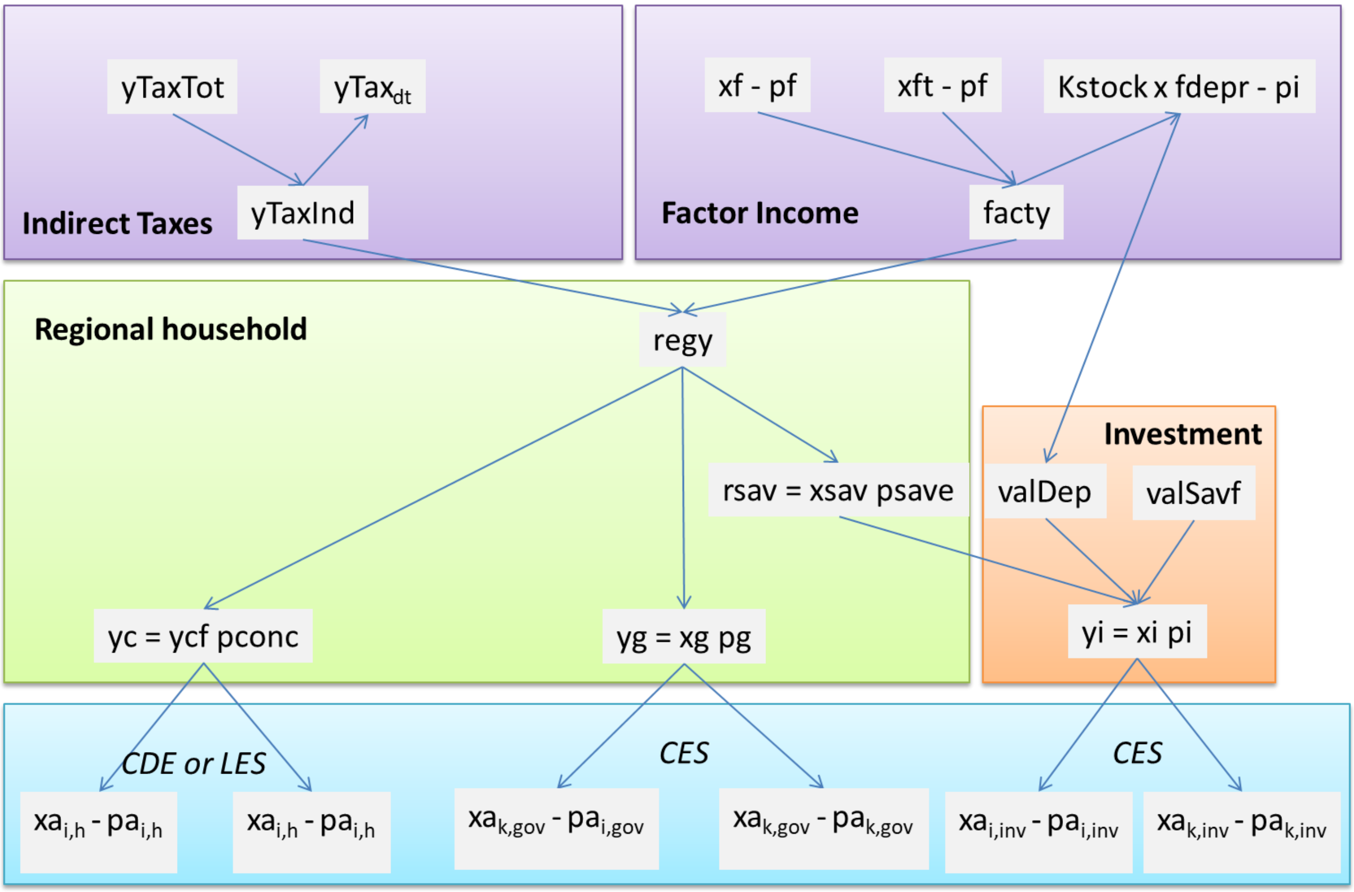


Figure 2: Overview in income generation and distribution

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 *regYeq*:

```
regYeq(rsNat(rNat),ts(t)) $ (regy.range(rNat,t) ne 0) ..
   regY(rNat,t)/regy.scale(rNat,t) =e= (factY(rNat,t) + yTaxInd(rNat,t))/regy.scale(rNat,t);
```

Factor income including direct taxes *factY* is defined by the *factYeq*. Is 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 *fdepr*, the average price of investments *pi* and the capital stock *kstock*:

```
 factYeq(rsNat(rNat),ts(t))  ..
   factY(rNat,t)/facty.scale(rNat,t)  =e= [    sum(fm $ xftFlag(rNat,fm), pft(rNat,fm,t)*xft(rNat,fm,t))
                                          + sum((fnm,a) $ xfFlag(rNat,fnm,a), pf(rNat,fnm,a,t)*xf(rNat,fnm,a,t))
                                          - fdepr(rNat,t)*pi(rNat,t)*kstock(rNat,t) ] / facty.scale(rNat,t);
```

Indirect tax income *yTaxInd* is calculated by the *ytaxIndeq* from total tax revenues *ytaxTot*, corrected for direct taxes (index "*dt*") comprised in factor income *factY* as defined above:

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

Total tax income *yTaxTot* considers all tax flow *gy* depicted in the model (see next equation) and is defined by the equation *ytaxTotEq*:

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

Tax flows *yTax* for the different types of tax flow *gy* are defined by the equation *ytaxeq*. 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.

```
[sum(a $ xpFlag(rNat,a), prdtx(rNat,a,t)*(m_px(rNat,a,t)*m_xp(rNat,a,t)
  +sum(tNest $ tNest_n_a("Top",tNest,a),xtNest(rNat,tNest,a,t)*ptNest(rNat,tNest,a,t))))] $sameas(gy,"pt")
```

2. **Indirect taxes on private consumption** *pc*, charged with rate *dintx* on domestic consumption 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*:

```
+   [sum((i,h),   [  (dintx(rNat,i,h,t) + itxshft(rNat,h,t) $ iTaxShift(i))*m_domPrice(rNat,i,h,t)*m_xd(rNat,i,h,t) ]  $ m_alphad(rNat,i,h,t)
              + [  (mintx(rNat,i,h,t) + itxshft(rNat,h,t) $ iTaxShift(i))*pmt(rNat,i,t)*m_xm(rNat,i,h,t)           ]  $ m_alpham(rNat,i,h,t))] $ sameas(gy,"pc")
```

Note the product specific general tax shifter *itxShft* which is normally not active.

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

3.

```
--- Indirect tax revenues from government consumption -- TGCRATIO (965)
+   [sum((i,gov),    [ (dintx(rNat,i,gov,t) + itxshft(rNat,gov,t) $ iTaxShift(i))*m_domPrice(rNat,i,gov,t)*m_xd(rNat,i,gov,t) ] $ m_alphad(rNat,i,gov,t)
                + [ (mintx(rNat,i,gov,t) + itxshft(rNat,gov,t) $ iTaxShift(i))*pmt(rNat,i,t)*m_xm(rNat,i,gov,t)            ] $ m_alpham(rNat,i,gov,t))] $sameas(gy,"gc")

--- Indirect tax revenues from investment consumption -- TIURATIO (1327)
+   [sum((i,inv),     [ (dintx(rNat,i,inv,t) + itxshft(rNat,inv,t) $ iTaxShift(i))*m_domPrice(rNat,i,inv,t)*m_xd(rNat,i,inv,t) ] $ m_alphad(rNat,i,inv,t)
                 + [ (mintx(rNat,i,inv,t) + itxshft(rNat,inv,t) $ iTaxShift(i))*pmt(rNat,i,t)*m_xm(rNat,i,inv,t)            ] $ m_alpham(rNat,i,inv,t))] $ sameas(gy,"ic")
```

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:

4.

```
+   [   sum(fm $ xftFlag(rNat,fm), (kappaf(rNat,fm,t)+kappaShft(rNat,t)) * pft(rNat,fm,t)*xft(rNat,fm,t))
      + sum((fnm,a) $ xfFlag(rNat,fnm,a), (kappaf(rNat,fnm,t)+kappaShft(rNat,t)) * pf(rNat,fnm,a,t)*xf(rNat,fnm,a,t)) ] $ sameas(gy,"dt")
```

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 destination, 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*.

5.

```
+   [sum((i,rNat1) $ m_xwFlag(rNat,i,rNat1), (exptx(rNat,i,rNat1,t) + etax(rNat,i,t))
      * [ ( pe(rNat,i,rNat1,t) $ (omegaw(rNat,i) ne inf)
          + pet(rNat,i,t)    $ ((omegaw(rNat,i) eq inf) $ (omegax(rNat,i) ne inf))
          + ps(rNat,i,t)     $ (omegax(rNat,i) eq inf)) $ (not iMel(i)) + (m_pFirm(rNat,i,rNat1,t) $ iMel(i)) ]
           *m_xws(rNat,i,rNat1,t))] $sameas(gy,"et")
```

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 via *%pmcif%* and the bi-lateral flows *xw*.

6.
```
+   [sum((i,rNat1) $ m_xwFlag(rNat1,i,rNat), (imptx(rNat1,i,rNat,t) + mtax(rNat,i,t))*%pmcif%(rNat1,i,rNat,t)*xw(rNat1,i,rNat,t))] $sameas(gy,"mt")
```

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*:

7.
```
+   [sum((f,a) $ xfFlag(rNat,f,a), (fcttx(rNat,f,a,t) + fcttxShift(rNat,t) $ fTaxShift(f))  * pf(rNat,f,a,t) * xf(rNat,f,a,t))] $sameas(gy,"ft")
```

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

8.
```
+   [sum((f,a) $ xfFlag(rNat,f,a), fctts(rNat,f,a,t)       * pf(rNat,f,a,t) * xf(rNat,f,a,t))] $sameas(gy,"fs")
```

Finally, emission taxes *emis* can be introduced, levied on emissions *emis* with the potentially endogenous price *emisP*. Currently, these only relate to CO2 emissions.

9.
```
+   [emis(rNat,t) * emisP(rNat,t) $ permit(rNat,t) ]  $ sameas(gy,"emis")
```

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*:

```
betaPhiEq(rsNat(rNat),ts(t))  ..
   betaPhi(rNat,t) =E= betaP(rNat,t)/phiP(rNat,t);
```

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 total expenditure *phiRegy* is accordingly proportionally decreased to yield still total regional income *regY*.

```
phiRegyEq(rsNat(rNat),ts(t))  ..
   phiRegy(rNat,t)*(betaPhi(rNat,t) + betaG(rNat,t) + betaS(rNat,t))/regy.scale(rNat,t)
        =E= regY(rNat,t)/regy.scale(rNat,t);
```

That corrected income then dries the private, government and savings expenditures. The amount spent for private consumption *yc* is defined in the equation *yceq*:

```
*  Total private consumption expenditure -- PRIVCONSEXP (2260)
 yceq(rsNat(rNat),h,ts(t))  ..
   yc(rNat,h,t)/yc.scale(rNat,h,t) =e= betaPHi(rNat,t)*phiregY(rNat,t)/yc.scale(rNat,h,t);
```

The amount spent for government consumption *yg* is defined in the equation *ygeq*:

```
*  Total government consumption expenditure -- GOVCONSEXP (2265)
 ygeq(rsNat(rNat),ts(t))  ..
   yg(rNat,t)/yg.scale(rNat,t) =e= betaG(rNat,t)*phiRegy(rNat,t)/yg.scale(rNat,t);
```

And finally, the amount of regional savings *rsav* is depicted in the equation *rsaveq:*

```
*  Total nominal saving -- SAVING (2270)
 rsaveq(rsNat(rNat),ts(t))  ..
   rsav(rNat,t)/xsav.scale(rNat,t) =e= betaS(rNat,t)*phiRegY(rNat,t)/xsav.scale(rNat,t);
```

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

Household consumption

The **consumer price index** *pcons* is defined from the budget share *xcshr* and the Armington prices defined in the macro *m_pa*:

```
* Consumer expenditure deflator (approx.) -- PHHLDINDEX (1005)

pconseq(rs(rNat),h,ts(t)) $ (sum( i, xaFlag(rNat,i,h))  and ( (pcons.range(rNat,h,t) ne 0) or (not rres(rNat)) or (lcu.range(rNat,t) ne 0)) )  ..
    pcons(rNat,h,t) =e= sum( i $ xaFlag(rNat,i,h), xcshr(rNat,i,h,t)*m_pa(rNat,i,h,t)) ;
```

The Armington demands for household consumption can be defined either by a CDE demand system as used in the GTAP Standard model or a LES demand system as found in my other CGEs. Note that the LES system collapses to a CD system if the commitments are removed, such the model can host three different demand system for household consumption.

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 *xaceq*:

```
xaceq(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 *xchsreq* are derived from unscaled shares *zcons*, scaled again by unity based on their sum *zConsSum*:

```
xcshreq(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) =e= zcons(r,i,h,t)/zConsSum(r,h,t);
```

The unscaled shares *zcons* as defined in the equation *zconseq* 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 parameter vectors *alphaa*, *bh* and *eh*.

```
zconseq(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 *zConsSumeq*:

```
 zConsSumeq(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) $ 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 *uheq*:

```
uheq(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) $ 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 *phiPEq*. It updates the private consumption share in the regional household income distribution:

```
phiPeq(rs(r),ts(t)) $ ((phip.range(r,t) ne 0) $ rNat(r)) ..
  phiP(r,t) =e= sum((i,h) $ xaFlag(r,i,h), xcshr(r,i,h,t)*eh(r,i,t))/sum(h $ hFlag(r,h),1);
```

LES or CD case

The equations for the LES or CD case are found in the file “model\dem_les.gms”.

The **Armginton demands** *xa* in the LES case reflect the constant term *gammaLES*, often termed commitment, and a share *alphaLES* on ono-committed income *yCNonCom* divided by the Armington price defined in the macro *m_pa*:

```
*
*  --- demand equations
*
   xacLesEq(rs(r),i,h,ts(t)) $ (xaFlag(r,i,h) $ alphaLES.l(r,i,h,t)) ..
    xa(r,i,h,t) /xa.scale(r,i,h,t)
      =e= (gammaLES(r,i,h,t) + alphaLES(r,i,h,t)/m_pa(r,i,h,t) * yCNonCom(r,h,t))/xa.scale(r,i,h,t);
*
```

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

```
*
*  --- demand for sub-nests under top nest
*
   xdNestLesEq(rs(r),dNest,h,ts(t)) $ sum(dNest_n_fd("top",dNest,h), alphaLES.l(r,dNest,h,t)) ..
    xdNest(r,dNest,h,t) /xdNEst.scale(r,dnest,h,t)
      =e= (gammaLES(r,dNest,h,t) + alphaLES(r,dnest,h,t)/pdNest(r,dNest,h,t) * yCNonCom(r,h,t))/xdNest.scale(r,dNest,h,t);
*
```

Non-committed income *yCNonCom* as defined in the equation *yCNonComEq* reflect total private consumption expenditure *yc* minus the value of the commitments, i.e. the gamma parameters multiplied with the Armginton prices defined in the macro *m_pa*:

```
  yCNonComEq(rs(r),h,ts(t)) $ sum(i,xaFlag(r,i,h))  ..
    yCNonCom(r,h,t)/yc.scale(r,h,t)
     =E= (yc(r,h,t) - sum(i $ (xaFlag(r,i,h) $ alphaLES.l(r,i,h,t)), gammaLES(r,i,h,t) * m_pa(r,i,h,t))
                    - sum(dNest_n_fd("top",dNest,h), gammaLES(r,dNest,h,t) * pdNest(r,dNest,h,t)))/yc.scale(r,h,t);
```

The **budget shares** *xcshr* are defined from the Armington demands, prices and expenditures in the equation *xcshrLESeq*:

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

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

```
   uhLESeq(rs(r),h,ts(t)) $ sum(i,xaFlag(r,i,h)) ..

     uh(r,h,t) =E= 1/uhLES(r,h,t)
                    *prod(i $ (xaFlag(r,i,h) $ alphaLES.l(r,i,h,t)),
                        (xa(r,i,h,t) - gammaLES(r,i,h,t))**alphaLES(r,i,h,t) )

                    *prod(dNest_n_fd("top",dNest,h) $ alphaLES.l(r,dNest,h,t),
                        (xdNest(r,dNest,h,t) - gammaLes(r,dNest,h,t))**alphaLES(r,dNest,h,t) );
```

Note that the CD case is comprised in the equations above if the commitment terms are set to zero.

Government consumption

First the reader is reminded that under the regional household approach, there is not separate household account and hence no direct link between tax income and government spent. That can be changed by using the “myGTAP module”.

Government consumption *yg* under the regional household approach is a share *betag* of regional income *regy*, corrected for the endogenous share of private spent captured by *phiRegy* (see above for savings):

```
ygeq(rsNat(rNat),ts(t))  ..
  yg(rNat,t)/yg.scale(rNat,t)  =e=  betaG(rNat,t)*phiRegy(rNat,t)/yg.scale(rNat,t);
```

The physical demand aggregate *xg* is derived from the price index *pg* defined by the equation *pgeq*. The price index *pg* is defined under the assumption of CES / CD / Leontief demand for governemtn using the typical dual price aggregator based on the investment specific Armington prices defined by the marco *m_pa*, a preference shifter variable *lambag* and the share parameters *alphaa* for government. Which case is used is defined by the related substitution elasticity *sigmag*. Note that the equation also considers the case that governemtn demand uses CES sub-nests which aggregates products to product groups:

```
pgeq(rs(rNat),ts(t)) $ axg(rNat,t) ..

    pg(rNat,t) =E=
*
*              --- CD case with sigmag==1
*
              ( [ prod((i,gov) $ ((not sum(dNest_i_fd(dnest,i,gov),1)) $ alphaa.l(rNat,i,gov,t)),
                      (m_pa(rNat,i,gov,t)/alphaa(rNat,i,gov,t))**alphaa(rNat,i,gov,t))
*
*                 -- CES-subnests, not part of Standard GTAP model
*
                 * prod(dNest_n_fd("top",dnest,gov) $ alphaDN(rNat,dNest,gov,t),
                       (pdNest(rNat,dNest,gov,t)/alphaDN(rNat,dNest,gov,t))**alphaDN(rNat,dNest,gov,t))
                 ]/axg(rNat,t) ) $ (sigmag(rNat) eq 1)

*
*              --- CES or Leontief case with sigmag<>1
*
      +          ([ sum((i,gov) $ ((not sum(dNest_i_fd(dnest,i,gov),1)) $ alphaa.l(rNat,i,gov,t)),
                    alphaa(rNat,i,gov,t)*m_pa(rNat,i,gov,t)**(1-sigmag(rNat)))
*
*                 -- CES-subnests, not part of Standard GTAP model
*
                 + sum(dNest_n_fd("top",dnest,gov) $ alphaDN(rNat,dNest,gov,t),
                    alphaDN(rNat,dNest,gov,t)*pdNest(rNat,dNest,gov,t)**(1-sigmag(rNat)))

                 ]**(1/(1-sigmag(rNat)))/axg(rNat,t)
                     ) $ (sigmag(rNat) ne 1);
```

The macro *m_pa* is usally defined as follows. If both a share parameters for imports *alpham* and for domestic sales *alphad* is given, it uses the definition of the Armington price *m_padef* 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 Armginton price is used, a case relevant when the Melitz module is active.

```
$$macro m_pa(r,i,aa,t) {           m_padef(r,i,aa,t)                   $ (alpham(r,i,aa,t) $ alphad(r,i,aa,t) ) \
                               + (m_pdp(r,i,aa,t)*alphad(r,i,aa,t)) $ ((not alpham(r,i,aa,t)) $ alphad(r,i,aa,t))\
                               + (m_pmp(r,i,aa,t)*alpham(r,i,aa,t)) $ ((not alphad(r,i,aa,t)) $ alpham(r,i,aa,t))\
$$ifi "%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:

```
$$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 physical demand *xg* is distributed in the equation *xageq* to demand 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*:

```
xageq(rs(rNat),iIn(i),gov,ts(t)) $ (xaFlag(rNat,i,gov) $ (not sum(dNest_i_fd(dnest,i,gov),1)) $ alphaa.l(rNat,i,gov,t)) ..
   xa(rNat,i,gov,t)/xa.scale(rNat,i,gov,t)  =e= alphaa(rNat,i,gov,t)*xg(rNat,t)/xa.scale(rNat,i,gov,t)*(pg(rNat,t)/m_pa(rNat,i,gov,t))**sigmag(rNat) ;
```

Note that that in the case of demand nests, additional equations are used as described in the section Demand sub-nests.

Investments and savings

Gross investment expenditures *yi* are composed of the value of depreciation *valDep* and of regional *rsav* and foreign savings *valSavf* by the equation *yieq*:

```
yieq(rsNat(rNat),ts(t)) $ ( ((not rres(rNat)) or singleCountry or (pnum.range(t) gt 0))  ) ..
   yi(rNat,t)/yi.scale(rNat,t)   =e= (
*                                   ---- value of depreciation
                                    valDep(rNat,t)
*                                   ---- savings of the regional household
                                  + rsav(rNat,t)
*                                   ---- Value of foreign savings
                                  + valsavf(rNat,t))/yi.scale(rNat,t);
```

The price index of investments *pi* is defined by the equation *pieq*. For a detailed explanation of the equation, refer to the explanation for government above.:

```
pieq(rsNat(rNat),ts(t))  ..

   pi(rNat,t)  =e=
*
*              --- CD case with sigmai==1
*
               ( [ prod((i,inv) $ ((not sum(dNest_i_fd(dnest,i,inv),1)) $ alphaa.l(rNat,i,inv,t)),
                      (m_pa(rNat,i,inv,t)/(lambdai(rNat,i,t)*alphaa(rNat,i,inv,t)))**alphaa(rNat,i,inv,t))
*
*                -- CES-subnests, not part of Standard GTAP model
*
                 * prod(dNest_n_fd("top",dnest,inv) $ alphaDN(rNat,dNest,inv,t),
                      (pdNest(rNat,dNest,inv,t)/alphaDN(rNat,dNest,inv,t))**alphaDN(rNat,dNest,inv,t))
                  ]/axi(rNat,t) ) $ (sigmai(rNat) eq 1)

*
*              --- CES or Leontief case with sigmai<>1
*
      +         ([ sum((i,inv) $ ((not sum(dNest_i_fd(dnest,i,inv),1))  $ alphaa.l(rNat,i,inv,t)),
                   alphaa(rNat,i,inv,t)*(m_pa(rNat,i,inv,t)/lambdai(rNat,i,t))**(1-sigmai(rNat)))
*
*                -- CES-subnests, not part of Standard GTAP model
*
                 + sum(dNest_n_fd("top",dnest,inv) $ alphaDN(rNat,dNest,inv,t),
                   alphaDN(rNat,dNest,inv,t)*pdNest(rNat,dNest,inv,t)**(1-sigmai(rNat)))

                 ]**(1/(1-sigmai(rNat)))/axi(rNat,t)
                    ) $ (sigmai(rNat) ne 1)
      ;
```

The **total physical investment demand** *xi* is derived from the price index *pi* defined above and total investment expenditure *yi*:

```
xieq(rsNat(rNat),ts(t))  ..
   pi(rNat,t)*xi(rNat,t)/yi.scale(rNat,t) =e= yi(rNat,t)/yi.scale(rNat,t);
```

Product specific investment demand *xa* depicted in the equation *xaieq* reflects the share parameters *alphaa*, the substitution elasticity *sigmai* and the shifter variable *lamdbai*:

```
xaieq(rs(rNat),iIn(i),inv,ts(t)) $ (xaFlag(rNat,i,inv) $ (not sum(dNest_i_fd(dnest,i,inv),1)) $ alphaa.l(rNat,i,inv,t)) ..

   xa(rNat,i,inv,t)/xa.scale(rNat,i,inv,t) =e=
      alphaa(rNat,i,inv,t)*xi(rNat,t)/xa.scale(rNat,i,inv,t)
         * [ ((lambdai(rNat,i,t)*pi(rNat,t)/m_pa(rNat,i,inv,t))**sigmai(rNat)/lambdai(rNat,i,t) ) $ (sigmai(rNat) ne 0)
             + lambdai(rNat,i,t)  $ (sigmai(rNat) eq 0)];
```

The **value of depreciation** *valdep* is a given share *depr* on the variably capital stock *kstock* which together define the physical depreciation, multiplied with the average price of savings *pi*:

```
valDepEq(rsNat(rNat),ts(t))  ..
   valDep(rNat,t)/valDep.scale(rNat,t) =E=  pi(rNat,t)*depr(rNat,t)*kstock(rNat,t)/valDep.scale(rNat,t);
```

Regional savings *rsav* are a given share *betas* of regional income, corrected for expansion effects:

```
*  Total nominal saving -- SAVING (2270)
 rsaveq(rsNat(rNat),ts(t))  ..
   rsav(rNat,t)/xsav.scale(rNat,t) =e= betaS(rNat,t)*phiRegY(rNat,t)/xsav.scale(rNat,t);
```

The correction implied by phiRegY ensures that the shares of savings *betaS*, government *betaG* and private consumption *betaPhi* add up to unity:

```
phiRegyEq(rsNat(rNat),ts(t))  ..
   phiRegy(rNat,t)*(betaPhi(rNat,t) + betaG(rNat,t) + betaS(rNat,t))/regy.scale(rNat,t)
        =E= regY(rNat,t)/regy.scale(rNat,t);
```

The **physical amount of regional savings** *xsav* is based on the savings expenditures *rsav* and the average price of savings *psave*:

```
xsaveq(rsNat(rNat),ts(t))  ..
   xsav(rNat,t)*psave(rNat,t)/xsav.scale(rNat,t) =e= rsav(rNat,t)/xsav.scale(rNat,t);
```

The **regional value of foreign savings** *valSavf* is defined from the value in foreign currency *savf* and the exchange rate *lcu*:

```
valSavfEq(rsNat(rNat),ts(t))  ..
   valSavf(rNat,t)/savf.scale(rNat,t) =E= savf(rNat,t)*lcu(rNat,t)/savf.scale(rNat,t);
```

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.

Global bank

The global bank mechanism distributes foreign savings across regions 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 *depr*:

```
netInvEq(rsNat(rNat),ts(t)) ..
   netInv(rNat,t)/netinv.scale(rNat,t) =E= [xi(rNat,t) - depr(rNat,t)*kstock(rNat,t)]/netinv.scale(rNat,t);
```

The **beginning of period capital stock** *kStock* is defined in the eqation *kStockEq*. It converts with the factor *krat* capital use *xft* (for mobile or sluggish capital) or non-mobile capital use *xf* into the aggregate capital stock, where capital (types) are define by the set *cap*:

```
kstockeq(rsNat(rNat),ts(t)) $ ( (kstock.range(rNat,t) ne 0) ) ..
   kstock(rNat,t)/kapend.scale(rNat,t) =e= [  sum(fm(cap), xft(rNat,cap,t))
                                    + sum((fnm(cap),a) $ xfFlag(rNat,cap,a), xf(rNat,cap,a,t))]/(krat(rNat,t)*kapend.scale(rNat,t));
```

The **end of period capital stock** *kapEnd* is derived by deducting the depreciating rate depr times the number of depreciation year *nDeprYears* and adding gross investment times the number of depreciations years:

```
kapEndeq(rsNat(rNat),ts(t))  ..
   kapEnd(rNat,t)/kapend.scale(rNat,t) =e= [(1 - depr(rNat,t)*nDeprYears)*kstock(rNat,t) + xi(rNat,t)*nDeprYears]/kapend.scale(rNat,t);
```

The **average returns of returns to capital after taxes** *arent* is defined in the equation *arenteq* from mobile capital prices *pft*, considering direct taxes *kappaf* and their potential shifter *kappaShft* as well as the factor which convert yearly capital use into stock values *krat*:

```
arenteq(rsNat(rNat),ts(t)) $ (rorFlag ne fixedForeignSavings) ..
*
*  --- note WB: (1) the set cap comprises only one element
*               (2) does not yet work with cap being immobile
*
   arent(rNat,t)/arent.scale(rNat,t) =e= sum(cap $ fm(cap), pft(rNat,cap,t)*(1-kappaf(rNat,cap,t)-kappaShft(rNat,t))/krat(rNat,t))/arent.scale(rNat,t) ;
```

The **net rate of return to capital** *rorc* as defined in the equation *rorceq* corrects these gross returns factors *arent* for the depreciation rate *fdepr*:

```
rorceq(rsNat(rNat),ts(t)) $ (rorFlag ne fixedForeignSavings) ..
   rorc(rNat,t)/rorc.scale(rNat,t) =e= (arent(rNat,t)/pi(rNat,t) - fdepr(rNat,t))/rorc.scale(rNat,t);
```

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 *–RorFlex* into account, multiplied with the net rate of return to capital *rorc*, defined in the two equations *rorePart1Eq* and *roreEq*:

```
rorePart1Eq(rsNat(rNat),ts(t)) $ (RorFlag ne fixedForeignSavings)..
   rorePart1(rNat,t)/rorePart1.scale(rNat,t) =E= [(kapEnd(rNat,t)/kStock(rNat,t))**(-RoRFlex(rNat,t))]/rorePart1.scale(rNat,t) ;

*  Expected rate of return -- ROREXPECTED (1620)

roreeq(rsNat(rNat),ts(t)) $ (RorFlag ne fixedForeignSavings)..
    rore(rNat,t)/rore.scale(rNat,t)  =e=     rorc(rNat,t)/rore.scale(rNat,t)  * rorePart1(rNat,t);
```

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

Specifically, the global bank mechanism aims at equalizing the expected net returns *rore* across regions by changing the distribution of foreign savings *fsav* to the different regions. In the benchmark, a risk parameter *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 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 expected rate *rore*.

The **value of global net investment** *gblValNetInv* as defined in the equation *gvlValNetInvEq* is derived from the regional net investments *netInv*, their regional prices *pi* and the exchange rate *lcu*:

```
gblValNetInvEq(ts(t)) ..
   gblValNetInv(t)/sum(rNat,netInv.scale(rNat,t)) =E= sum(rNat(r), (pi(r,t)/lcu(r,t)    *netInv(r,t))   $ rs(r)
                                                      +(pi.l(r,t)/lcu.l(r,t)*netInv.l(r,t)) $ (not rs(r)))
                                                    /sum(rNat,netInv.scale(rNat,t));
```

Note the distinction between regions in the current solve *rs*, in case the model is run in single country mode or the pre-processing is used, and those exogenous.

Regional net investments *netInv* as defined in the *netInvEq* are equal to aggregated gross investment demand *xi* minus depreciation, i.e. the depreciation rates fdepr times the beginning of year capital stocks *kStock*:

```
netInvEq(rsNat(rNat),ts(t)) ..

  netInv(rNat,t)/netinv.scale(rNat,t) =E=
   [xi(rNat,t) - depr(rNat,t)*kstock(rNat,t)]/netinv.scale(rNat,t);
```

The **total global net investments** *xigl* are defined in the equation *xigbleq* as the summing up of the regional net investments *netInv*:

```
xigbleq(ts(t)) ..
    xigbl(t)/sum(r, netInv.scale(r,t)) =e= sum(rNat(r), netInv(r,t)    $ rs(r)
*                                       ---add values of countries not in current solve,
*                                          but in the global framework
                                            + netInv.l(r,t) $ (not rs(r)))/sum(r,netInv.scale(r,t)) ;
```

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 equation *rorgeg* and *gblValNetInv1Eq*:

```
gblValNetInv1Eq(ts(t)) $ (RoRFlag eq fixedAllocationOfInv) ..
   gblValNetInv1(t)/sum(rNat,netInv.scale(rnat,t))
        =E= sum(rNAt(r),  (rore(r,t)    * pi(r,t)/lcu(r,t)      * netInv(r,t)) $ rs(r)
                        + (rore.l(r,t) * pi.l(r,t)/lcu.l(r,t) * netInv.l(r,t)) $ (not rs(r)))
                 /sum(rNat,netInv.scale(rNat,t)) ;

rorgeq(ts(t)) $ (RoRFlag eq fixedAllocationOfInv) ..
   rorg(t) =e= gblValNetInv1(t) / gblValNetInv(t);
```

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 *savfeq* which requires for each region that risk adjusted expected returns are equal to the global average:

```
savfeq(rsNat(rNat),ts(t)) $( (     (RoRFlag eq equalReturnToInv)
                              or ((RoRFlag eq fixedAllocationOfInv) $ (not rres(rNat))))
                          $   (savf.range(rNat,t) ne 0))..
    0    =e=
 *
 *   --- equal relative changes in expected returns to investments
 *       across regions, rorg in the global currency
 *
          ([rore(rNat,t)*risk(rNat,t) - rorg(t)]/rore.scale(rNat,t) )       $ (RoRFlag eq equalReturnToInv)
 *
 *    --- fixed allocation of global investment based on parameter chiInv
 *
       +  [(netInv(rNat,t) - chiInv(rNat,t)*xigbl(t))/netInv.scale(rNat,t)]  $ (RoRFlag eq fixedAllocationOfInv);
```

Fixed allocation of foreign savings

That case is depicted in the second block of the equation *savfeq* if the *RoRFlag* is set equal to "fixedAllocationOfInv" and is based on given parameters *chiInv* which reflect the benchmark distribution:

```
savfeq(rsNat(rNat),ts(t)) $( (     (RoRFlag eq equalReturnToInv)
                              or ((RoRFlag eq fixedAllocationOfInv) $ (not rres(rNat))))
                          $   (savf.range(rNat,t) ne 0))..
    0    =e=
 *
 *   --- equal relative changes in expected returns to investments
 *       across regions, rorg in the global currency
 *
          ([rore(rNat,t)*risk(rNat,t) - rorg(t)]/rore.scale(rNat,t) )       $ (RoRFlag eq equalReturnToInv)
 *
 *    --- fixed allocation of global investment based on parameter chiInv
 *
       +  [(netInv(rNat,t) - chiInv(rNat,t)*xigbl(t))/netInv.scale(rNat,t)]  $ (RoRFlag eq fixedAllocationOfInv);
```

Note that the residual region is excluded from the mechanism and defined via the capital account balance.

Capital account and balance of payments

The capital account balance *capAcctEq* ensures that the sum of the foreign savings *savf* is zero:

```
*
capAccteq(ts(t)) $ ( (    (sum(rNat(rs) $ (savf.range(rs,t) ne 0),1) gt 1)
                      or (       (RoRFlag eq fixedForeignSavings) and sum(rres(rs),1)
                            and ( (pnum.range(t) gt 0) or (not singleCountry) ))) ) ..

   0 =e= sum(rNat(r), savf(r,t) $ rs(r) + savf.l(r,t) $ (not rs(r)))/sum(r,savf.scale(r,t));
```

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 *bopEq* 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:

```
bopEq(rsNat(rNat),ts(t))  ..

         (valsavf(rNat,t) + bopSlack(rNat,t))
         /savf.scale(rNat,t)
          =E= [ - sum( (i,rNat1)  $ m_xwFlag(rNat,i,rNat1), m_xws(rNat,i,rNat1,t)*%pefob%(rNat,i,rNat1,t))
                - sum( (m,tmg)  $ xaFlag(rNat,m,tmg), mm_xa(rNat,m,tmg,t)*m_pa(rNat,m,tmg,t)*lcu(rNat,t))
                + sum( (rNat1,i)  $ m_xwFlag(rNat1,i,rNat), xw(rNat1,i,rNat,t)*%pmcif%(rNat1,i,rNat,t))]
        /savf.scale(rNat,t)
  ;
```

Demand sub-nests

Demand sub-nests aggregate individual Armington demands in final demand by private household, government or savings to aggregates based on CES utility function. The resulting nests can be either part of the top level demand function of the function or linked into other demand nests. That mechanism allows increasing the flexibility of depicting substitution relations between individual products.

The **average price for a demand nest** *pdNest* for the nest *dNest* and the demand agent *fdn* is defined by dual price aggregators which dinstiguish the CD from the CES/Leontief case depending on the substitution elasticity *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 *alphaDN*.

```
pdNest(rNat,dNest,fdn,t)
     =E=

         --- CD case with sigmaFDNest==1

        [

            ---- contribution of individual commodity demands to price aggregator

            prod(dNest_i_fd(dNest,i,fdn) $ alphaa.l(rNat,i,fdn,t),
                         (m_pa(rNat,i,fdn,t)/(alphaa(rNat,i,fdn,t)*(1+(lambdai(rNat,i,t)-1)$ sameas(fdn,"inv"))))**alphaa(rNat,i,fdn,t))

            ---- contribution of sub-nests under the current nest to price aggregator

         *  prod(dNest_n_fd(dNest,dNest1,fdn) $ alphaDN(rNat,dNest1,fdn,t),
                         (pdNest(rNat,dNest1,fdn,t)/alphaDN(rNat,dNest1,fdn,t))**alphaDN(rNat,dNest1,fdn,t))

         * axdNestCD(rNat,dNest,fdn,t)


        ] $ (sigmaFDNest(rNat,dNest,fdn) eq 1)

         --- CES or Leonteif case with sigmaFDNest<>1

       + [
         { sum(dNest_i_fd(dNest,i,fdn) $ alphaa.l(rNat,i,fdn,t),
                alphaa(rNat,i,fdn,t) * (m_pa(rNat,i,fdn,t)/(1+(lambdai(rNat,i,t)-1)$ sameas(fdn,"inv")))**(1-sigmaFDNest(rNat,dNest,fdn))
            )

         + sum(dNest_n_fd(dNest,dNest1,fdn) $ alphaDN(rNat,dNest1,fdn,t),
                alphaDN(rNat,dNest1,fdn,t) * pdNest(rNat,dNest1,fdn,t)**(1-sigmaFDNest(rNat,dNest,fdn))
            )

         } **(1/(1-sigmaFDNest(rNat,dNest,fdn)))

        ]  $ (sigmaFDNest(rNat,dNest,fdn) ne 1);
```

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

```
xdNestEq(rsNat(rNat),dnest,fdn,ts(t)) $ (alphaDN(rNat,dNest,fdn,t) and (not sum(dNest_n_fd("top",dnest,h) $ sameas(fdn,h),1))) ..
*
*  --- total demand for sub-nest
*
  xdNest(rNat,dNest,fdn,t)/sum(i, xa.scale(rNat,i,fdn,t))
      =E=

          [
*
*          --- driven by top nest demand in case of gov/inv (CES or CD)
*              Final household demand has separate equation to reflec the more complex CDE case
*
          + sum(dNest_n_fd("top",dnest,fdn) $ (alphaDN(rNat,dNest,fdn,t) and sameas(fdn,"gov")),
                   alphaDN(rNat,dNest,fdn,t) * xg(rNat,t)   * (pg(rNat,t)/pdNest(rNat,dNest,fdn,t))**sigmaG(rNat))

          + sum(dNest_n_fd("top",dnest,fdn) $ (alphaDN(rNat,dNest,fdn,t) and sameas(fdn,"inv")),
                   alphaDN(rNat,dNest,fdn,t) * xi(rNat,t)   * (pi(rNat,t)/pdNest(rNat,dNest,fdn,t))**sigmai(rNat))
*
*          -- driven by another nest
*
          + sum(dNest_n_fd(dNest1,dNest,fdn) $ alphaDN(rNat,dNest,fdn,t),
                   alphaDN(rNat,dNest,fdn,t) * xdNest(rNat,dNest1,fdn,t) * (pdNest(rNat,dNest1,fdn,t)/pdNest(rNat,dNest,fdn,t))**sigmaFDNest(rNat,dNest1,fdn))

         ]/sum(i, xa.scale(rNat,i,fdn,t));
```

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

```
zconsDnEq(rs(r),dNest,h,ts(t) ) $ sum(dNest_n_fd("top",dNest,h), alphaDN(r,dNest,h,t)) ..

   zconsDN(r,dNest,h,t)/zconsDN.scale(r,dNest,h,t) =e= alphaDN(r,dNest,h,t)*bhDN(r,dnest,t)
                      * (pdNest(r,dNest,h,t)**bhDN(r,dNest,t))
                      *  ((yc(r,h,t)/pop(r,h,t)*1000)**(-bhDN(r,dNest,t)))
                      *  (uh(r,h,t)**(ehDN(r,dNest,t)*bhDN(r,dNest,t)))/zconsDN.scale(r,dNest,h,t);
```

The **Armington demands** ***xa*** **driven by a sub-nest** are defined in the equation *xdDNesteq* and use the usual CES-structure, i.e. the share parameter *alphaa*, the sub-nest total demand *xdNest* and the price relation exponent the substitution elasticity sigmaFDNest as well a preference shifter *lambdai*:

```
  xaDNestEq(rsNat(rNat),i,fdn,ts(t)) $ (xaFlag(rNat,i,fdn) $ sum(dNest_i_fd(dNest,i,fdn),1) $ iIn(i)) ..

   xa(rNat,i,fdn,t)/xa.scale(rNat,i,fdn,t) =e=
      sum(dNest_i_fd(dNest,i,fdn),
            alphaa(rNat,i,fdn,t)*xdNest(rNat,dnest,fdn,t)/xa.scale(rNat,i,fdn,t)
*
*              --- not the preference shifter lambdai for investment demand
*
               * [
*
*                   --- CES case with price effect
*
                   (
                     ((1+(lambdai(rNat,i,t)-1)$ sameas(fdn,"inv"))*pdNest(rNat,dNest,fdn,t)/m_pa(rNat,i,fdn,t))
                            **sigmaFDNest(rNat,dNest,fdn)
                             /((1+(lambdai(rNat,i,t)-1)$ sameas(fdn,"inv")))
                    )   $ (sigmaFDNest(rNat,dNest,fdn) ne 0)
*
*                   --- Leontief case with fixed composition, no price effect
*
                  + (
                       (1+(lambdai(rNat,i,t)-1) $ sameas(fdn,"inv"))
                    )  $ (sigmaFDNest(rNat,dNest,fdn) eq 0)
                  ]);
```

International trade and domestic sales, and related prices

Overview

The model uses in its standard layout a two-stage Armington system where the shares of the lower nests representing bi-lateral imports are identical across the different Armington agents. The Armington approach can be complemented by a CET to distribute supply in each region based on finite transformation. Alternatives to the standard layout in demand are: a further aggregation in the Armington system where also the upper nested is shared across sectors or across all agents, the “MRIO extension” and the heterogenous firm extension, see the sub-section “Melitz model”.

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 from domestic origin and imports. Next, these demands are aggregated over the agents. The total import demand is then split up into demands of the different exporters *xw*, driven by the cif price plus import taxes. The difference between the cif and fob 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.

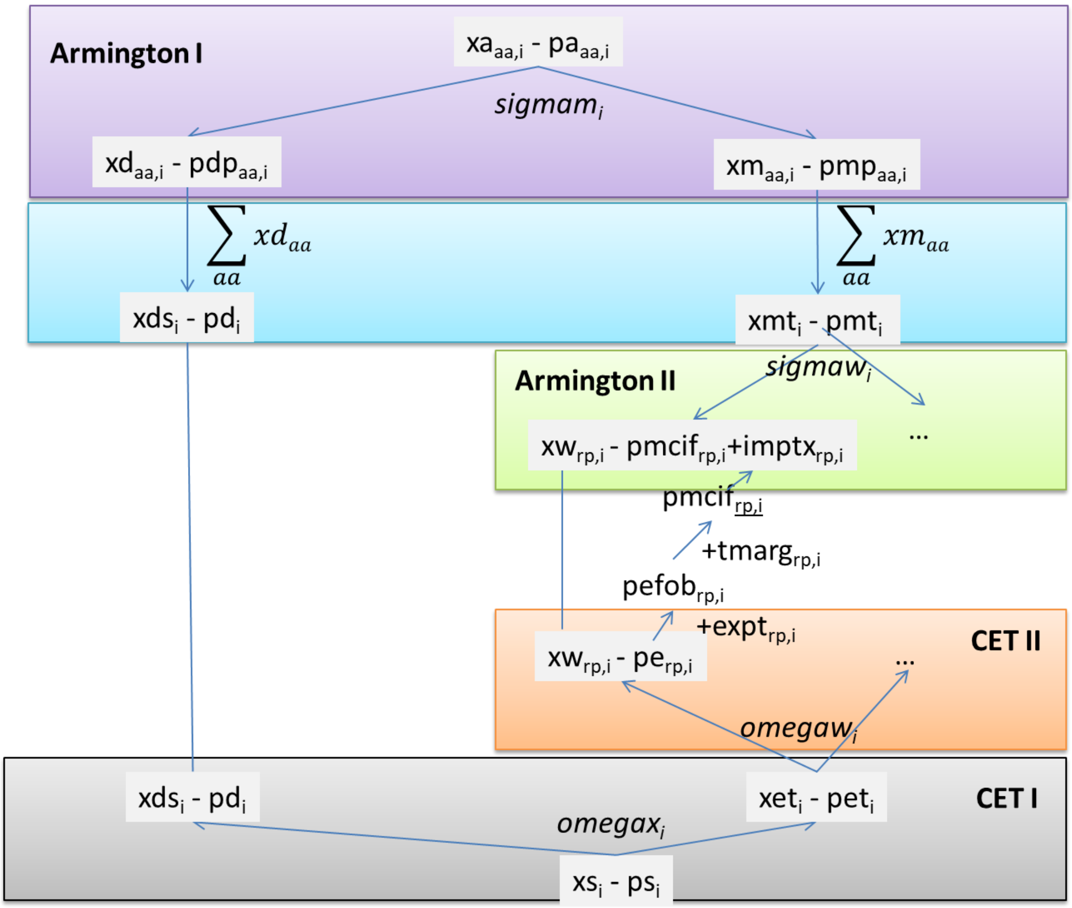


Figure 3: Overview on distribution of supply and sourcing of demand

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.

Individual equations

The **agent specific prices for imports** *pmp* as defined in the equation *pmpeq* reflects the average import price *pmt* and agent specific import taxes *mintx* plus tax shifts *itxshft* and emission taxes:

```
pmpeq(rsNat(rNat),iIn(i),aa,ts(t)) $ (alpham(rNat,i,aa,t) $ (not iMrio(i)) $ (not a(aa) or sum(sameas(aa,aIn),1))) ..
   pmp(rNat,i,aa,t) =e= pmt(rNat,i,t)*(1 + mintx(rNat,i,aa,t) + itxshft(rNat,aa,t)$ iTaxShift(i))
*
*       --- add per unit emission tax
*
       $$ifi %ModulesCO2_Emissions%==on  + emisp(rNat,t)* (emii0(rNat,i,aa)/xma0(rNat,i,aa)) $ xma0(rNat,i,aa)
  ;
```

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 that the share parameters for imports *alpham* is not zero. The Melitz extension sets the share parameter to zero for the production handled in the Melitz model and thus also the 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 *pdeq*. 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:

```
pdpeq(rsNat(rNat),iIn(i),aa,ts(t)) $ (alphad(rNat,i,aa,t) $ (not a(aa) or sum(sameas(aa,aIn),1))) ..
   pdp(rNat,i,aa,t) =e= (    pd(rNat,i,t) $ (omegax(rNat,i) ne inf)
                           + ps(rNat,i,t) $ (omegax(rNat,i) eq inf)) *(1 + dintx(rNat,i,aa,t) + itxshft(rNat,aa,t)$ iTaxShift(i))
*
*       --- add per unit emission tax
*
       $$ifi %ModulesCO2_Emissions%==on  + emisp(rNat,t) * (emid0(rNat,i,aa)/xda0(rNat,i,aa)) $ xda0(rNat,i,aa)
  ;
```

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

The **Armington price** of the different agents *pa* is defined in the *paeq*. That dual price aggregator as usually reflects the given shares for domestic *alphad* and imported *alpham* origin and the related prices as defined above as well as the substitution elasticity between imports and domestic origin *sigmam* which is not agent specific:

```
 paeq(rsNat(rNat),iIn(i),aa,ts(t)) $ (  [        ((alphad(rNat,i,aa,t) $ alpham(rNat,i,aa,t)))
                                      or (gtapStd and (alphad(rNat,i,aa,t) or alpham(rNat,i,aa,t)))   ]
                                       $ (not a(aa) or sum(sameas(aa,aIn),1))
*
*           WB: If additional prices are substitued out (switched on in interface)
*               the Armington price aggregator for firm demands (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(rNat,i,aa,t) =e=
                    (    [ (alphad(rNat,i,aa,t)*m_pdp(rNat,i,aa,t)**(1-sigmam(rNat,i))) ]  $ alphad(rNat,i,aa,t)
                       +[ (alpham(rNat,i,aa,t)*m_pmp(rNat,i,aa,t)**(1-sigmam(rNat,i))) ]  $ alpham(rNat,i,aa,t))**(1/(1-sigmam(rNat,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 *alpham* 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 *xdeq*:

```
xdeq(rsNat(rNat),iIn(i),aa,ts(t)) $ ( alphad(rNat,i,aa,t) $ (alpham(rNat,i,aa,t) or gtapStd) $ (not a(aa) or sum(sameas(aa,aIn),1))) ..
   xd(rNat,i,aa,t)/xd.scale(rNat,i,aa,t)   =e=     alphad(rNat,i,aa,t)*mm_xa(rNat,i,aa,t)/xd.scale(rNat,i,aa,t)
                                                 * (m_pa(rNat,i,aa,t)/m_pdp(rNat,i,aa,t))**sigmam(rNat,i);
```

It is linearly aggregated over agents to **total domestic sales** *xds* in the equation *pdeq*:

```
pdeq(rsNat(rNat),iIn(i),ts(t)) $ xdFlag(rNat,i) ..
   xds(rNat,i,t)/xds.scale(rNat,i,t)
          =e= sum(aa $ alphad(rNat,i,aa,t), m_xd(rNat,i,aa,t))/xds.scale(rNat,i,t);
```

Imported demand by each agent *xm* is defined accordingly in the equation *xmeq*:

```
xmeq(rsNat(rNat),iIn(i),aa,ts(t))  $ ( (alphad(rNat,i,aa,t) or gtapStd) $ alpham(rNat,i,aa,t) $ (not a(aa) or sum(sameas(aa,aIn),1)) )  ..
   xm(rNat,i,aa,t)/xm.scale(rNat,i,aa,t)   =e=    alpham(rNat,i,aa,t)*mm_xa(rNat,i,aa,t)/xm.scale(rNat,i,aa,t)
                                              * (m_pa(rNat,i,aa,t)/m_pmp(rNat,i,aa,t))**sigmam(rNat,i);
```

Total import demand *xmt* is in the equation *xmteq* is defined as an adding up over the demand of the individual Armington agents:

```
xmteq(rsNat(rNat),iIn(i),ts(t)) $ xmtFlag(rNat,i) ..
   xmt(rNat,i,t)/xmt.scale(rNat,i,t) =e= sum(aa $ alpham(rNat,i,aa,t), m_xm(rNat,i,aa,t))/xmt.scale(rNat,i,t);
```

The bilateral **cost, insurance and freight prices** *pmcif* are defined by the equation *pmcifeq* and the macro *m_pmcif*. 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.

```
pmcifeq(rNat1,iIn(i),rNat,ts(t)) $ (m_xwFlag(rNat1,i,rNat) $ rrComb(rNat1,rNat)) ..
   pmcif(rNat1,i,rNat,t) =E= m_pmcif(rNat1,i,rNat,t);
```

The macro *m_pmcif* is defined as follows:

```
 $$macro m_pmcif(r,i,rp,t) { [ m_pefob(r,i,rp,t)/lcu(r,t) \
                            + (sum(m $ amgm(m,r,i,rp), amgm(m,r,i,rp)*ptmg(m,t)/m_lambdamg(m,r,i,rp,t))*tmarg(r,i,rp,t))\
                                                                                     $tmarg.l(r,i,rp,t)\
                            ]*lcu(rp,t) }
```

It converts the bilateral fob (free on board) price defined in the macro *m_pefob* 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 *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 realtions 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 50x50x50 = 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*. 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.

```
 $$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) }
```

Similar to the case of cif prices, the fob prices *pefob* are defined from that macro in the equation *pefobeq*. Again, the default case is that these equations are substituted out from the model.

```
pefobeq(rNat1,iIn(i),rNat,ts(t)) $ (m_xwFlag(rNat1,i,rNat) $ rrComb(rNat1,rNat)) ..
   pefob(rNat1,i,rNat,t) =E= m_pefob(rNat1,i,rNat,t);
```

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 *pmteq* 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 *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.

```
pmteq(rNat,iIn(i),ts(t)) $ ( (pmt.range(rNat,i,t) ne 0) $ xmtFlag(rNat,i) $ (not pmtElas(rNat,i)) ) ..
*
   pmt(rNat,i,t) =e=
*
*         --- CES case with sigmaw <> 1
*
          [sum(rNat1 $ (xwFlag(rNat1,i,rNat) $ (not isSmallImpShare(rNat1,i,rNat))), amw(rNat1,i,rNat,t)
                    *([(1 + imptx(rNat1,i,rNat,t) + mtax(rNat,i,t)) * %pmcif%(rNat1,i,rNat,t)/chipm(rNat1,i,rNat) ]/m_lambdam(rNat1,i,rNat,t))**(1-sigmaw(rNat,i))
                  )**(1/(1-sigmaw(rNat,i))) * axPmtCD(rNat,i,t) ]
                     $ ((sigmaw(rNat,i) ne 1) $ axPmtCD(rNat,i,t))
*
*         --- CD case with sigmaw == 1
*
        + [prod(rNat1 $ (xwFlag(rNat1,i,rNat) $ (not isSmallImpShare(rNat1,i,rNat))),
                    ([(1 + imptx(rNat1,i,rNat,t) + mtax(rNat,i,t)) * %pmcif%(rNat1,i,rNat,t)/chipm(rNat1,i,rNat) ]
                     /(m_lambdam(rNat1,i,rNat,t)*amw(rNat1,i,rNat,t)))**amw(rNat1,i,rNat,t)
                  ) * axPmtCD(rNat,i,t) ]
                      $ ((sigmaw(rNat,i) eq 1) $ axPmtCD(rNat,i,t))
*
        + sum(rNat1 $ isSmallImpShare(rNat1,i,rNat), amw(rNat1,i,rNat,t)
                  *([(1 + imptx(rNat1,i,rNat,t) + mtax(rNat,i,t)) * %pmcif%(rNat1,i,rNat,t)/chipm(rNat1,i,rNat) ]/m_lambdam(rNat1,i,rNat,t)));
```

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 *pmteq* 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 *xweq* and is based on the share parameters *amw*, the substitution between origins *sigmaw* 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.

```
xweq(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)
                             *  m_lambdam(rNat1,i,rNat,t)**(sigmaw(rNat,i) - 1)} $ (not isSmallImpShare(rNat1,i,rNat))

                          + m_lambdam(rNat1,i,rNat,t) $ isSmallImpShare(rNat1,i,rNat)
                         ];
```

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.

```
   $$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 )) ]
```

Similar macros are used for other shifter variables as well.

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

```
xdseq(rsNat(rNat),iIn(i),ts(t)) $ (xdFlag(rNat,i) $ (omegax(rNat,i) ne inf) ) ..
   0 =e=
*
*     --- infinite CET <=> prices are equal
*
         (pd(rNat,i,t) - ps(rNat,i,t))                                                  $ (omegax(rNat,i) eq inf)
*
*     --- CET mechanism otherwise
*
      +  (xds(rNat,i,t)/xds.scale(rNat,i,t) - gd(rNat,i,t)*xs(rNat,i,t)/xds.scale(rNat,i,t)*(pd(rNat,i,t)/ps(rNat,i,t))**omegax(rNat,i))  $ (omegax(rNat,i) ne inf);
```

A similar equation defines **total exports** *xet* in the equation *xeteq*. The relevant share parameter is *ge* while the average price of exports is called *pet*:

```
xeteq(rsNat(rNat),iIn(i),ts(t)) $ (xetFlag(rNat,i) $ (xet.range(rNat,i,t) ne 0) $ (pet.range(rNat,i,t) ne 0)  )..
   xet(rNat,i,t)/xet.scale(rNat,i,t) =e=
*
                          (ge(rNat,i,t)*xs(rNat,i,t)/xet.scale(rNat,i,t)*(pet(rNat,i,t)/ps(rNat,i,t))**omegax(rNat,i))  $ (omegax(rNat,i) ne inf)
*
*                          --- infinite CET: physical aggregation
*
                          +  (  sum( rp $ xwFlag(rNat,i,rp), m_xws(rNat,i,rp,t)/xet.scale(rNat,i,t)) $ (not petElas(rNat,i))
                             + (axPet(rNat,i,t)  * pet(rNat,i,t)**petElas(rNat,i))               $      petElas(rNat,i)           )  $ (omegaw(rNat,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 *xseq*. 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 *gd* and *ge*, related prices *pd* and *pet* and the transformation elasticity *omegax*:

```
xseq(rsNat(rNat),iIn(i),ts(t)) $ (xsFlag(rNat,i) $ (ps.range(rNat,i,t) ne 0))..
   0 =e=
*
*     --- infinite CET <=> physical balancing
*
      +  (xs(rNat,i,t)/xs.scale(rNat,i,t) - (xds(rNat,i,t)/xs.scale(rNat,i,t)) $ m_xdFlag(rNat,i)
                                          - (xet(rNat,i,t)/xs.scale(rNat,i,t)) $ m_xetFlag(rNat,i) ) $ (omegax(rNat,i) eq inf)
*
*     --- dual price aggregator
*
      +  (ps(rNat,i,t) - (  gd(rNat,i,t)*pd(rNat,i,t)**(1+omegax(rNat,i))
                          + ge(rNat,i,t)*pet(rNat,i,t)**(1+omegax(rNat,i)))**(1/(1+omegax(rNat,i))) ) $ (omegax(rNat,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 *peeq*. Otherwise, bilateral export prices *pe* and the average export prices *pet* are by definition equal:

```
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 $ isSmallExpShare(rNat,i,rNat1) + { (pe(rNat,i,rNat1,t)/pet(rNat,i,t))**omegaw(rNat,i) } $ (not isSmallExpShare(rNat,i,rNat1)) ])
                   $ (omegaw(rNat,i) ne inf)

     +  (pe(rNat,i,rNat1,t) - pet(rNat,i,t) $ (omegax(rNat,i) ne inf) - ps(rNat,i,t) $ (omegax(rNat,i) eq inf)) $ (omegaw(rNat,i) eq inf) ;
```

The **aggregate price of export** *pet* defined in the equation *peteq* 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:

```
peteq(rNat,iIn(i),ts(t)) $ (m_xetFlag(rNat,i) $ sum(rsNat,rrComb(rNat,rsNat)) $ (pet.range(rNat,i,t) ne 0)) ..
   pet(rNat,i,t) =e= [  axPet(rNat,i,t) *sum(rNat1 $ (m_xwFlag(rNat,i,rNat1) $ (not isSmallExpShare(rNat,i,rNat1))),
                                    gw(rNat,i,rNat1,t)*pe(rNat,i,rNat1,t)**(1+omegaw(rNat,i)))**(1/(1+omegaw(rNat,i)))
                    + sum(rNat1 $ (m_xwFlag(rNat,i,rNat1) $ isSmallExpShare(rNat,i,rNat1)),      gw(rNat,i,rNat1,t)*pe(rNat,i,rNat1,t))
                  ]      $ (omegaw(rNat,i) ne inf)

                  +  ps(rNat,i,t) $ (omegaw(rNat,i) eq inf);
```

The **global demand for transport services** *xtmg* of mode *m* is based on a Leontief approach and defined in the equation *xtmgeq*. The given bi-lateral transport margin demand *tmarg* are distributed to the different transport modes *m* based on the share parameter *amgm* and 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:

```
xtmgeq(m,ts(t)) ..
   xtmg(m,t)/xtmg.scale(m,t) =e= sum((rNat,i,rNat1) $ amgm(m,rNat,i,rNat1), amgm(m,rNat,i,rNat1)*tmarg(rNat,i,rNat1,t)
                                           * (  m_xws(rNat,i,rNat1,t)    $ rrComb(rNat,rNat1)
$onexpand
                                             + m_xwsl(rNat,i,rNat1,t)  $ (not rrComb(rNat,rNat1)) )
$offexpand
                                             /m_lambdamg(m,rNat,i,rNat1,t))/xtmg.scale(m,t);
```

The **region specific demand for each transport mode** *xa* is based on a CES demand system which reflects the average global price for each transport mode *pgtm* and the regional specific price defined in the macro *m_pa* and the substitution elasticity *sigmamg*:

```
xatmgeq(rsNat(rNat),m,tmg,ts(t)) $ (xaFlag(rNat,m,tmg) and sum(sameas(m,iIn),1)) ..
   xa(rNat,m,tmg,t)/xa.scale(rNat,m,tmg,t)
    =e= alphaa(rNat,m,tmg,t)*xtmg(m,t)/xa.scale(rNat,m,tmg,t)  *(ptmg(m,t)/(m_pa(rNat,m,tmg,t)/lcu(rNat,t)))**sigmamg(m)  ;
```

The global average price for each transport mode *ptgm* is defined in the equation *ptmeq* via a dual price aggregator which distinguishes the CD and CES/Leontief case:

```
ptmgeq(m,ts(t)) ..

   ptmg(m,t) =e=
       (   (sum(  (rNat,tmg) $ alphaa.l(rNat,m,tmg,t),
                     alphaa(rNat,m,tmg,t)
                    *(  (m_pa(rNat,m,tmg,t)/lcu(rNat,t))     $ rsNat(rNat)
                     +  (pa.l(rNat,m,tmg,t)/lcu.l(rNat,t))  $ (not rsNat(rNat)))**(1-sigmamg(m))  )**(1/(1-sigmamg(m))))/axmg(m,t)  ) $ (sigmamg(m) ne 1)

     +  ( sum(tmg, prod(rNat $ alphaa.l(rNat,m,tmg,t), ((   (m_pa(rNat,m,tmg,t)/lcu(rNat,t))    $ rsNat(rNat)
                                                        + (pa.l(rNat,m,tmg,t)/lcu.l(rNat,t))  $ (not rsNat(rNat)))
                                                              /alphaa(rNat,m,tmg,t))**(alphaa(rNat,m,tmg,t))))/axmg(m,t)  ) $(sigmamg(m) eq 1)
     ;
```

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",t)*( (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:

```
 $$macro m_pAgentm(r,i,aa,rr1,t) \
                 [ %pmcif%(r,i,rr1,t)*(1 + imptx(r,i,rr1,t) + mtax(rr1,i,t))                \
                                     * (1 + mintx(rr1,i,aa,t) + itxshft(rr1,aa,t) $ iTaxShift(i))            \
                      $$ifi %ModulesCO2_Emissions%==on    + emisp(rr1,t) * (emii0(rr1,i,aa)/xma0(rr1,i,aa)) $ xma0(rr1,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_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))]


   $$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,rp,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 Krugmann model where only industry wide fix costs are present.

In the Krugmann 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(rr1,rrd), xw(rNat,i,rr1,t))+xds(rNat,i,t) $ sameas(rrd,"domes")]
                                                   *p_tau(rNat,i,rrd,t)
                                       $$ifi "%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))

  $$ifi "%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 $$ifi.

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=
*
*         --- fix costs times # of existing firms
*
        sum(i $ xFlag(rNat,a,i),  p_delt_fs(rNat,i,t)*MFirmsEnt(rNat,i,t) +
*
*         --- # of firms operating on the different markets times
*
*             (1) bilateral fix cost p_fc
*             (2) icebeg cost of trade p_tau times output per firm QAvFirm, corrected for average productivity

               sum(rrd $ xwFlagMel(rNat,i,rrd), m_NFirmsOp(rNat,i,rrd,t)*p_fc(rNat,i,rrd,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 aggretage, 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)
            $$ifi "%fcost_nest%" == "on"        -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.

Price indices

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

Regional factor price indices *pfact* are defined in the equation *pfacteq* based on factor prices *pft* and weights *phif*. In order to allow inclusion of non-mobile factors, their average price is fined in the *pftFnmEq*:

```
*
* --- calculate average price of non-mobile factors
*
pftFnmEq(rs(r),fnm,ts(t)) $ (xftEqFlag(r,fnm) or disr(r)) ..

   pft(r,fnm,t)*sum(a $ xfFlag(r,fnm,a), xf(r,fnm,a,t))/sum(a $ xfFlag(r,fnm,a), xf.scale(r,fnm,a,t))
    =E= sum(a $ xfFlag(r,fnm,a), xf(r,fnm,a,t)*pf(r,fnm,a,t))/sum(a $ xfFlag(r,fnm,a), xf.scale(r,fnm,a,t));

*  Regional index of factor prices
*
pfacteq(rsNat(rNat),ts(t)) $ ( ((pfact.range(rNat,t) ne 0) or (not rres(rNat)) or (lcu.range(rNat,t) ne 0))  ) ..
   pfact(rNat,t) =e= sum(f, phif(rNat,f,t)*pft(rNat,f,t)) ;
```

The **average world price of factors** *pwfact* as defined in the equation *pwfacteq* uses weights *phifw* and reflects the exchange rates *lcu* to aggregate the regional factor prices *pft*. It is global numeraire price in the model.

```
pwfacteq(ts(t)) ..
   pwfact(t) =e= sum((rNat,f), phifw(rNat,f,t)*(  pft(rNat,f,t) $ rsNat(rNat)
                                                + pft.l(rNat,f,t) $ (not rsNat(rNat)) )/lcu(rNat,t));
```

Regional producer price indices *pprod* as defined in the equation *pprodeq* are based on weights *phii*:

```
pprodeq(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, 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 *phia*:

```
pabseq(rsNat(rNat),ts(t)) ..
   pabs(rNat,t) =e= sum((i,fd), phia(rNat,i,fd,t)*m_pa(rNat,i,fd,t)) ;
```

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
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

Variable	Content	Indices
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

Flexible nesting

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 the GTAP in GAMS project 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 top nest it belongs too. These nesting definitions enter the above mentioned equations in the core model 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.

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:

Add technology nest to model, give it a name

The mother of the nest is the top VA nest

```
tNest("Labor") = YES;
tNest_n_a("VA","Labor",a)          = YES;
tNest_f_a("Labor","skLab",a)       = YES;
tNest_f_a("Labor","unSkLab",a)     = YES;
sigmaNest(r,"Labor",a)             = 0.5;
```

Link the factors into the bundle

Define the substitution elasticity

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 “GTAP-E” module.

2. ” to allow for high substitution of feed intermediates in animal production:

Add technology nest to model, give it a name

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

```
tNest("primEne") =YES;
Set primEne(i) /

  coa-c
  oil-c
  gas-c

/;
alias(primEne,primEne1);

tNest_n_a("ND","primEne",a)        $ sum( (r,primEne1,t), io(r,primEne1,a,t)) = YES;
tNest_i_a("primEne",primEne,a)     $ sum( (r,primEne1,t), io(r,primEne1,a,t)) = YES;
sigmaNest(r,"primEne",a)           $ sum( (primEne1,t), io(r,primEne1,a,t))   = 0.5;
```

The mother of the nest is the top ND-bundle

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

3. 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;

Set scndEne(i) /

  p_c-c
  ely-c
  gdt-c

/;

tNest_n_a("VA","energy",a)        = YES;
tNest_f_a("energy","capital",a)   = YES;
tNest_i_a("energy",scndEne,a)     = YES;
sigmaNest(r,"energy",a)           = 0.5;
```

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

The mother of the nest is the top VA-bundle

Link the factors into the bundle

Link the intermediates into the bundle

Define the substitution elasticity

4. 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 GTAP8 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 theses 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:

```
* Price for technology nests (not in standard GTAP model)

ptNestEq(rs(r),tNest,a,ts(t)) $ (xpFlag(r,a) $ (atNest(r,tNest,a,t)
                                    or sum(tNest_f_a(tNest,f,a), afNest(r,f,a,tNest,t))
                                    or sum(tNest_i_a(tNest,i,a), ioNest(r,i,a,tNest,t)) )) ..

   ptNest(r,tNest,a,t) =e=
   {[
*
*  --- aggregate over intermediate inputs assigned to technology nest
*
        sum(tNest_i_a(tNest,i,a) $ ioNest(r,i,a,tNest,t),
                                   ioNest(r,i,a,tNest,t)
                                     * (m_pa(r,i,a,t)/lambdaio(r,i,a,t))**(1-sigmaNest(r,tNest,a)))
*
*  --- aggregate over factors  assigned to technology nests
*
      + sum(tNest_f_a(tNest,f,a) $ afNest(r,f,a,tNest,t),
                                   afNest(r,f,a,tNest,t)
                                     * (m_pfa(r,f,a,t)/lambdaf(r,f,a,t))**(1-sigmaNest(r,tNest,a)))
*
*  --- aggregate over sub technology nests assigned to that technology nests
*
      + sum(tNest_n_a(tNest,tNest1,a) $ atNest(r,tNest1,a,t),
                                   atNest(r,tNest1,a,t)
                                     * ptNest(r,tNest1,a,t)**(1-sigmaNest(r,tNest,a)))
                                                                                     ]**(1/(1-sigmaNest(r,tNest,a)))
  }

   $ (sigmaNest(r,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 -- INTDEMAND (1286), with adjustments

xapeq(rs(r),i,a,ts(t)) $ (xapFlag(r,i,a) $ ndFlag(r,a) ) ..
   xa(r,i,a,t)/xp.scale(r,a,t)  =e=
*
*   --- input coefficient times total intermediate demand
*
    io(r,i,a,t)*nd(r,a,t)/xp.scale(r,a,t)
*
*   --- relative price impact (in case sigmnd is not 0), not part of standard GTAP model
*
                * [ {(pnd(r,a,t)/m_pa(r,i,a,t))**sigmand(r,a) } $ sigmand(r,a) + 1 $ (not sigmand(r,a)) ]
                   *  lambdaio(r,i,a,t)**(sigmand(r,a)-1)
*
     + sum(tNest_i_a(tNest,i,a),
*
*   --- input coefficient times demand for technology nest
*
    ioNest(r,i,a,tNest,t) * xtNest(r,tNest,a,t)/xp.scale(r,a,t)
*
*   --- relative price impact (in case sigmndtNest is not 0)
*
                    * [ {(ptNest(r,tNest,a,t)/m_pa(r,i,a,t))**sigmaNest(r,tNest,a) } $ sigmaNest(r,tNest,a)
                                                                  + 1 $ (not sigmaNest(r,tNest,a)) ]
                         *  lambdaio(r,i,a,t)**(sigmaNest(r,tNest,a)-1));
```

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)

xtNestEq(rs(r),tNest,a,ts(t)) $ (xpFlag(r,a) $ atNest(r,tNest,a,t)) ..

    xtNest(r,tNest,a,t)/xp.scale(r,a,t) =e=

*
*         --- nest is part of top level intermediate demand
*
          ( atNest(r,tNest,a,t) * nd(r,a,t)/xp.scale(r,a,t)
                  * [ {(pnd(r,a,t)/ptNest(r,tNest,a,t))**sigmaNd(r,a)} $ sigmaNd(r,a) + 1 $ (not sigmaNd(r,a)) ] ) $ tNest_n_a("ND",tNest,a)
*
*         --- nest is part of top level VA demand
*
        + ( atNest(r,tNest,a,t) * va(r,a,t)/xp.scale(r,a,t)
                  * [ {(pva(r,a,t)/ptNest(r,tNest,a,t))**sigmav(r,a)} $ sigmav(r,a) + 1 $ (not sigmav(r,a)) ] ) $ tNest_n_a("VA",tNest,a)
*
*         --- nest is part of other nests
*
        + sum(tNest_n_a(tNest1,tNest,a), atNest(r,tNest,a,t) * xtNest(r,tNest1,a,t)/xp.scale(r,a,t)
                  * [ {(ptNest(r,tNest1,a,t)/ptNest(r,tNest,a,t))**sigmaNest(r,tNest1,a)} $ sigmaNest(r,tNest1,a) + 1 $ (not sigmaNest(r,tNest1,a)) ]);
```

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 are kept.

GUI

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. That 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 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 and Britz et al. 2015) or, for instance, the GAMSIDE often used with economic models realized in GAMS. As a consequence, some functionalities found in runGTAP are not comprised the GUI linked to the GAMS version of the GTAP model and vice versa. The development of the GUI for the GAMS version and 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 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 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 allows 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 to using the GUI in interactive model, a batch mode allows to perform model runs. The batch reports these run in a HTML page and allows parallel execution of different shocks or of the same shock in different model variants.

The paper 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, Britz 2010b), 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 system 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.

Getting the model code and setting up the GUI

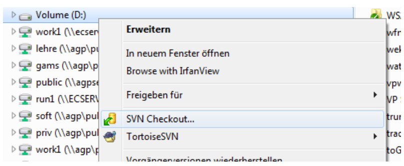
The code and GUI are distributed in three ways. First, a GAMS code only package can be downloaded as a zip archive from the GTAP web site. The GAMS code is identical to the one used in the GUI version, but the java binaries and other files necessary for the GUI are not included. That GAMS only package might be preferred by users wanting to use tools such as the GAMSIDE, develop their own GUI or may want to link the GTAP code into other GAMS projects. Second, a zipped archive “Gtap8InGams.zip” at ftp://gtapingams:gtapingams@ftp.agp.uni-bonn.de comprises both the GAMS code and the necessary files for the GUI.

SVN checkout

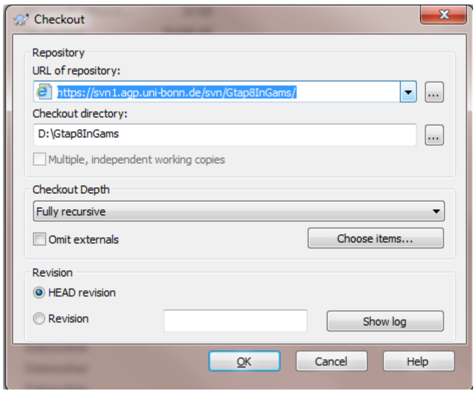
The third option is to benefit from SVN, a software version system on which the model codes are hosted on the repository https://svn1.agp.uni-bonn.de/svn/gtap8ingams/. The option is especially recommended for users who both want to use the GUI and foresee also own changes to the code. In the SVN repository, the development of CGEBox continues. We expect that over the next few months 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.

Therefore, as a first step, we recommend installing a SVN client such as TortoiseSVN (http://tortoisesvn.net/index.de.html, freeware) on your machine. Afterwards, make a checkout of the repository https://svn1.agp.uni-bonn.de/svn/Gtap8InGams/ (i.e. download the code from the server to your local machine), using the user id “gtapingams” and the pass word “gtapingams”. 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.

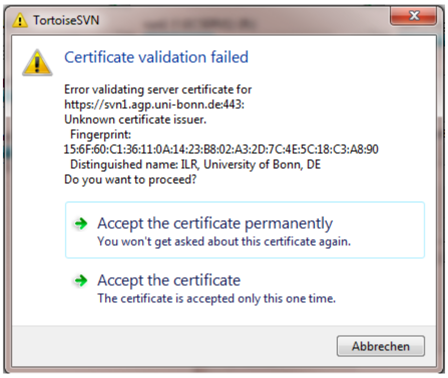
The basic steps for using SVN to install CGEBox based on TortoiseSVN are detailed below. First, navigate in the windows explorer to the disk / directory where you want to install GTAP, and open the context menu with a right mouse click:



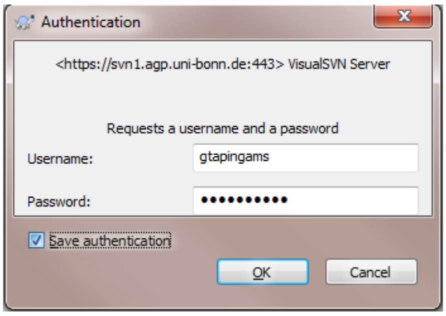
Chose, “SVN Checkout …” and copy the URL “https://svn1.agp.uni-bonn.de/svn/Gtap8InGams/“ in the first field as shown below and press “OK”.



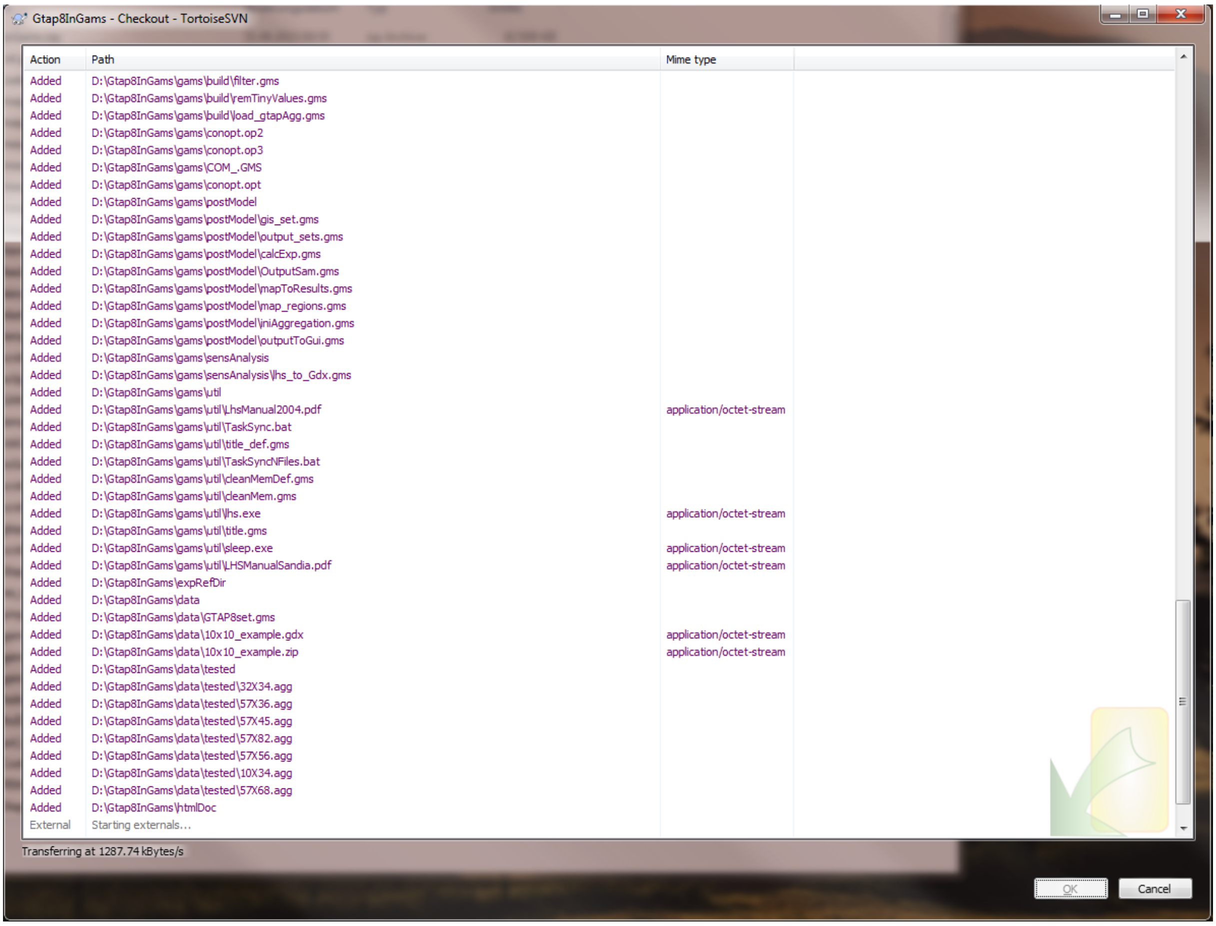
In case you get a warning such as:



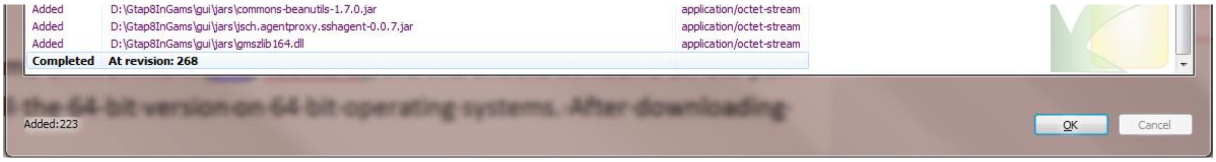
chose “Accept the certificate permanently”. In the next dialogue, enter the username and password, check “Save authentication” in the checkbox in the lower part of the dialog and press “OK”.



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.



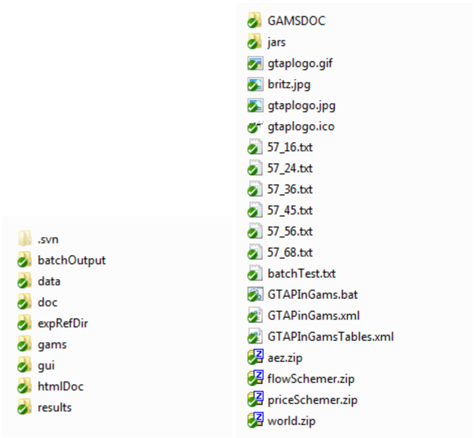
After you press "OK", you should find the new folder on your computer:

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 these files, a different symbol will be shown.

For further information on how to benefit from TortoiseSVN for your daily work with GTAP8InGAMS for other projects, please refer to the web. 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.

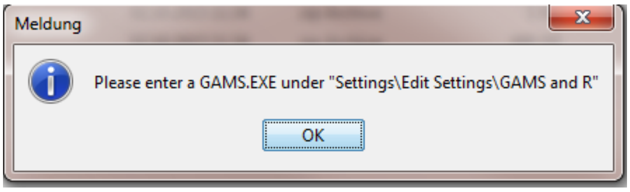
Starting the GUI

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 "*GTAPinGAMS.bat*"; it should open the interface[2]. 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:

[2] If that does not work, open a command prompt, start from there the "GTAPinGAMS.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.

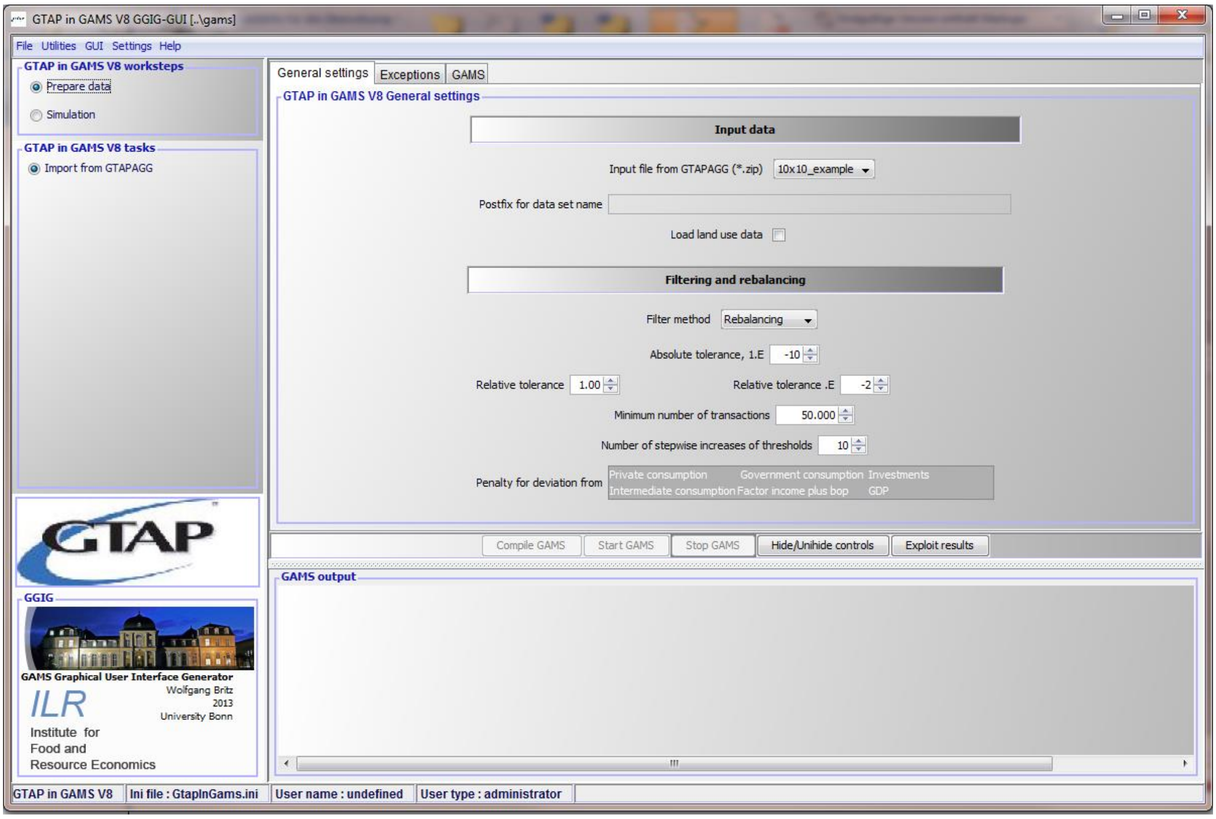


Figure: Interface at first start

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 GTAP8inGAMS, 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)
```

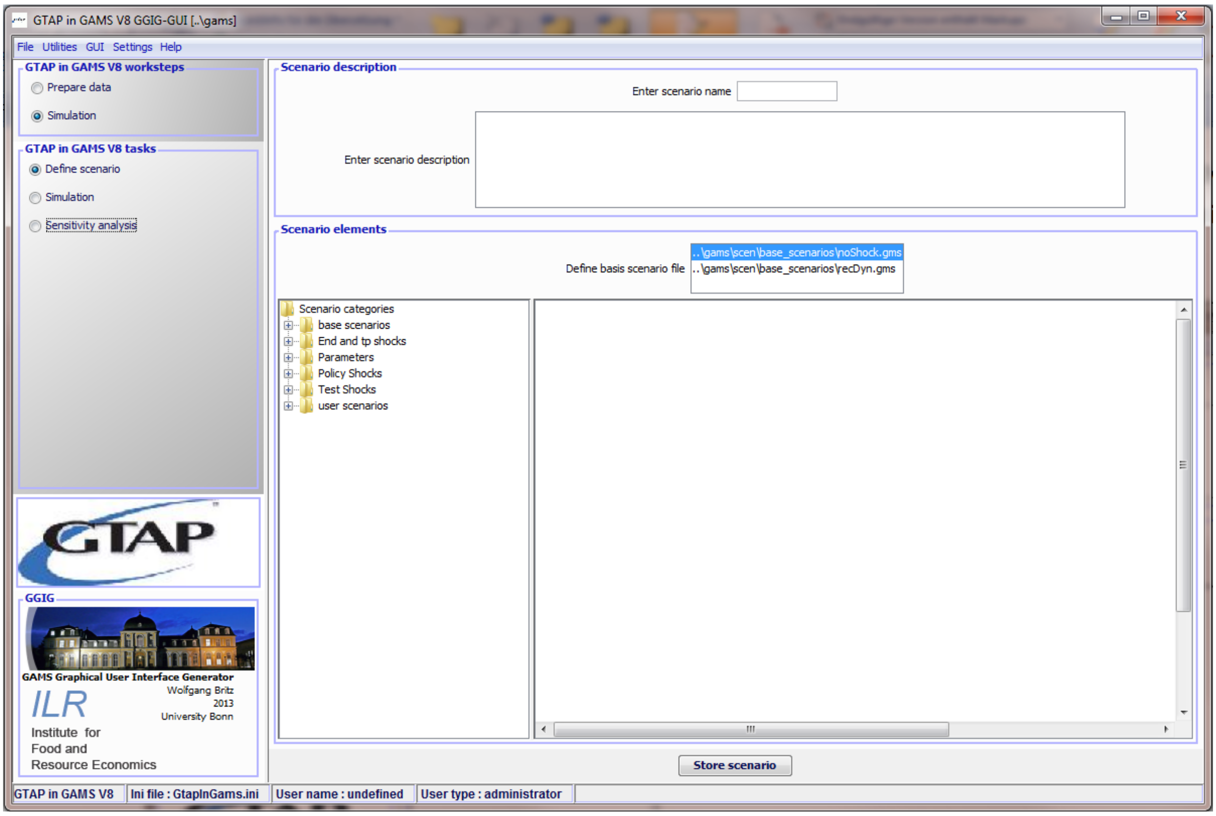
- If that still does not help, open a DOS prompt, navigate to the GUI folder and type "GtapInGams.bat" – possible errors will be shown in the window. In case you cannot find help in your team with regard to these error messages, contact the author.

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. That pfd 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.

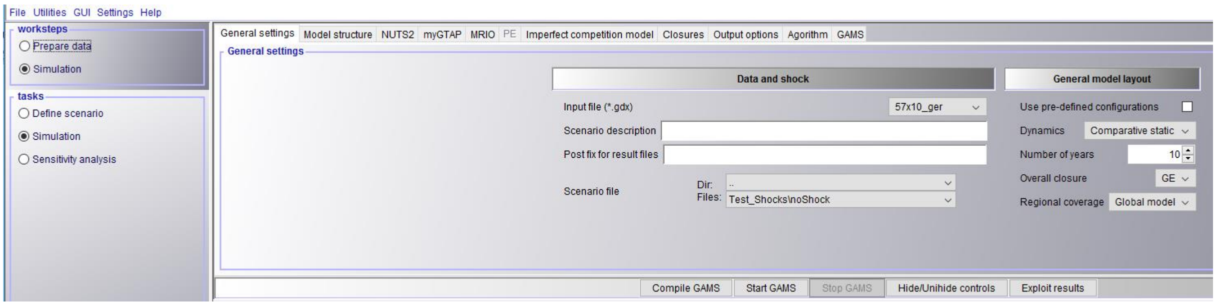
We would finally like to note that we did our best to test the combination of the CGEBox code and the GUI, but that improvements are certainly possible.

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":



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):

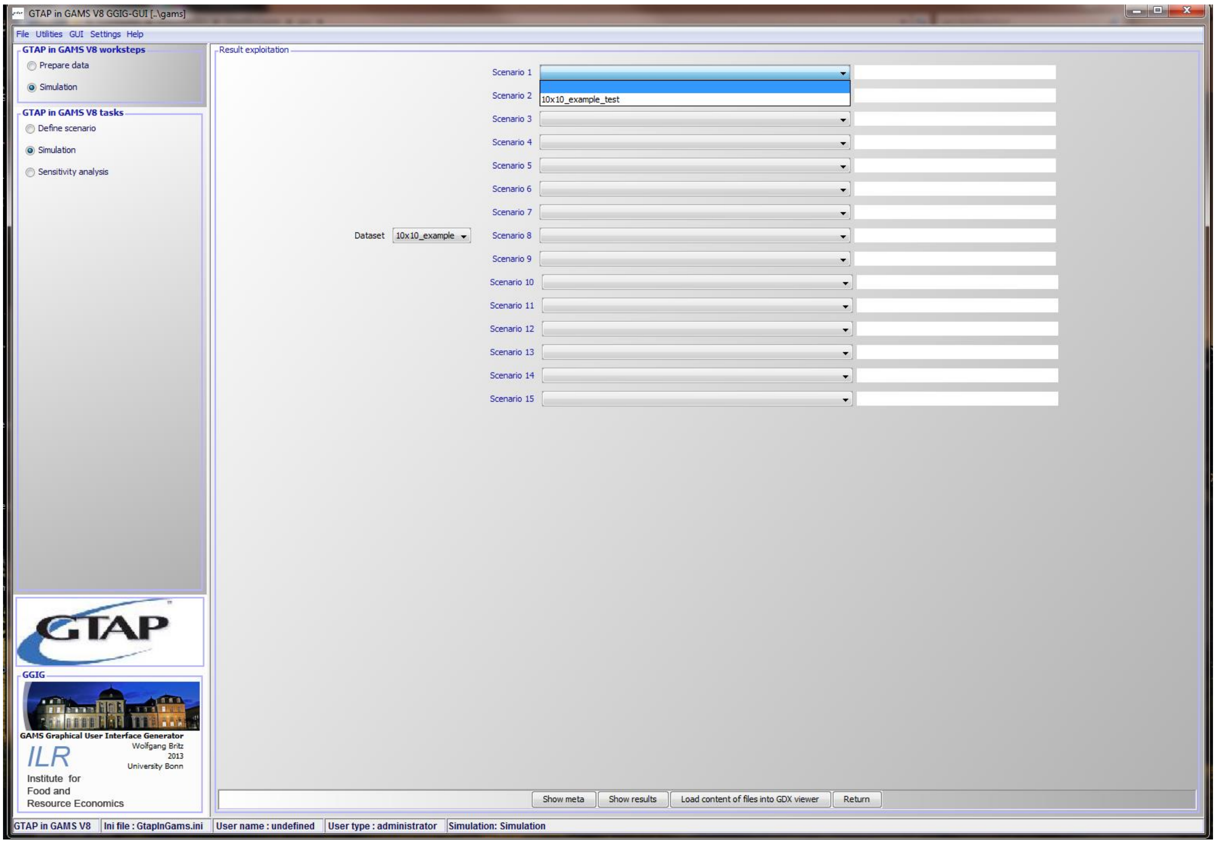
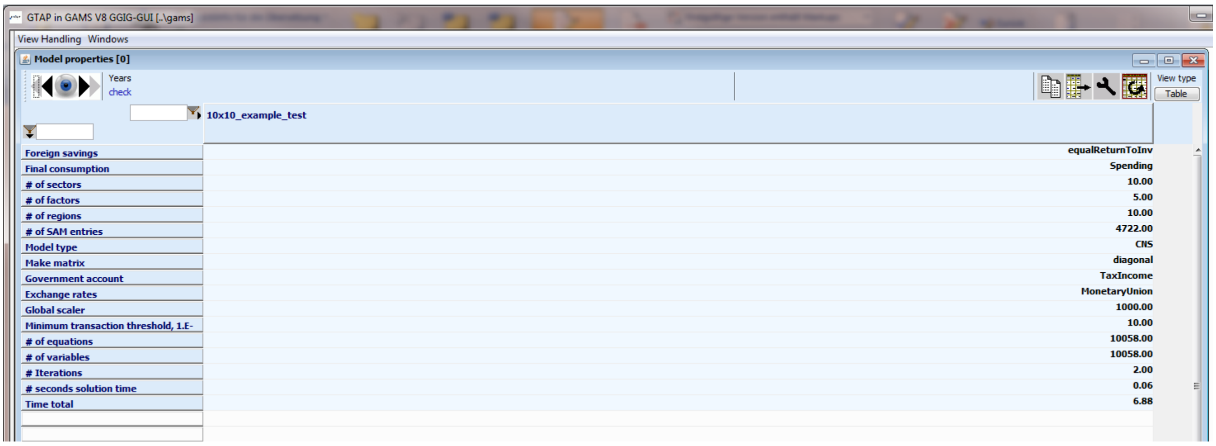


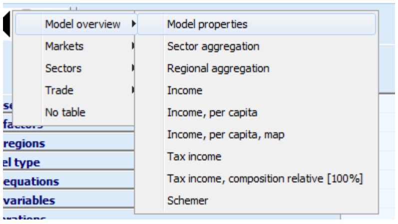
Figure: Scenario selection for exploitation

Select the scenario “10x_10_example_test” and press “show results”, you should see a table similar as the one shown below:



(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.)

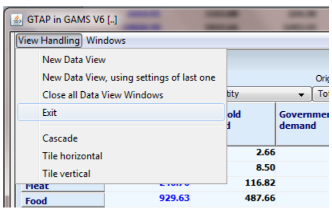
The "" view button in the upper left corner lets you select different views on your results:

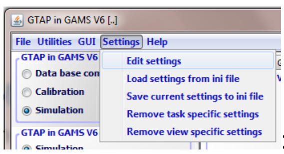


How to produce your own results is discussed in the following. Information on the different views is given in a section below.

Making the interface working with GAMS

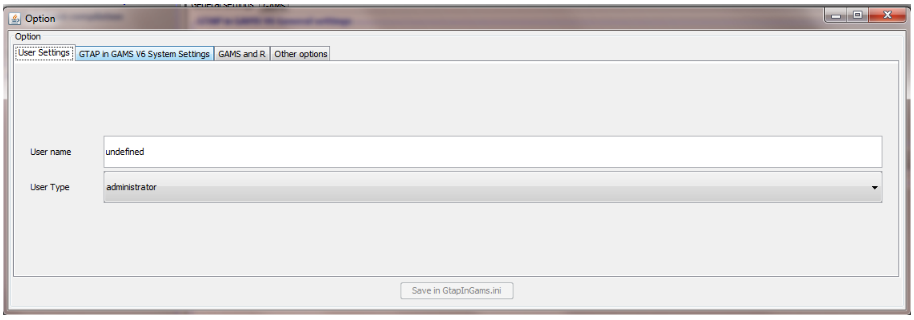
In order to use the GUI to generate data sets or for simulation runs, you have first to register a GAMS executable. In order to do so, first leave the result viewer as shown below



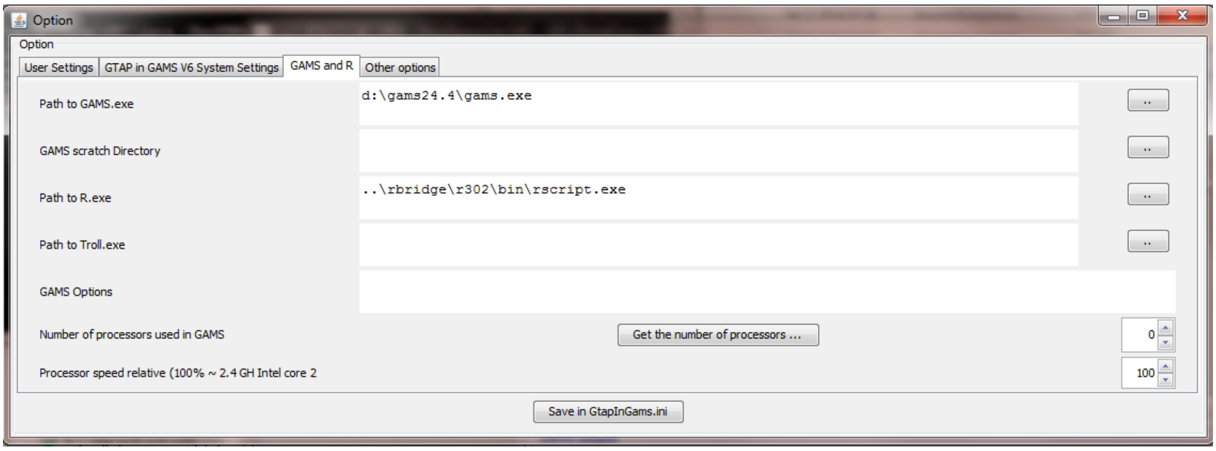
And open the setting dialogue from the menu bar 

:

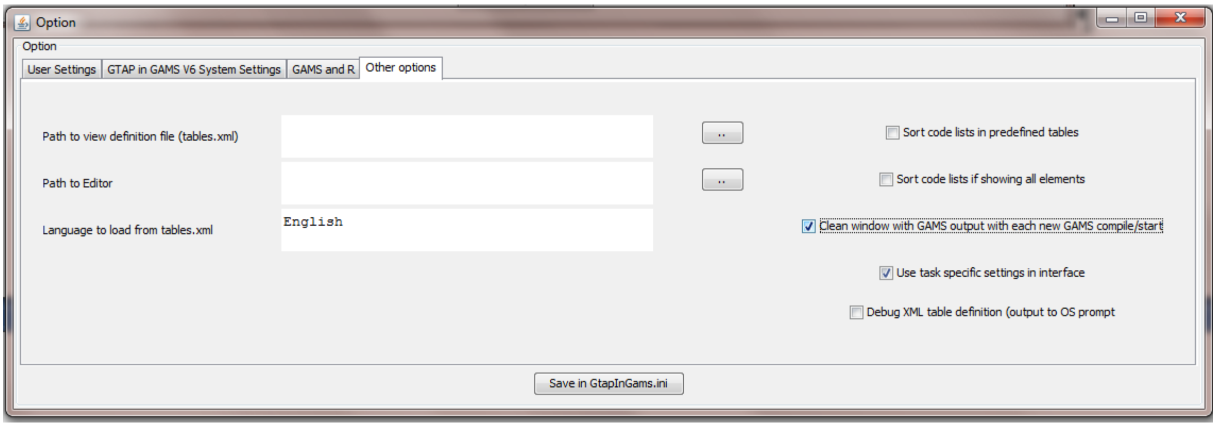
First, you should enter your name:



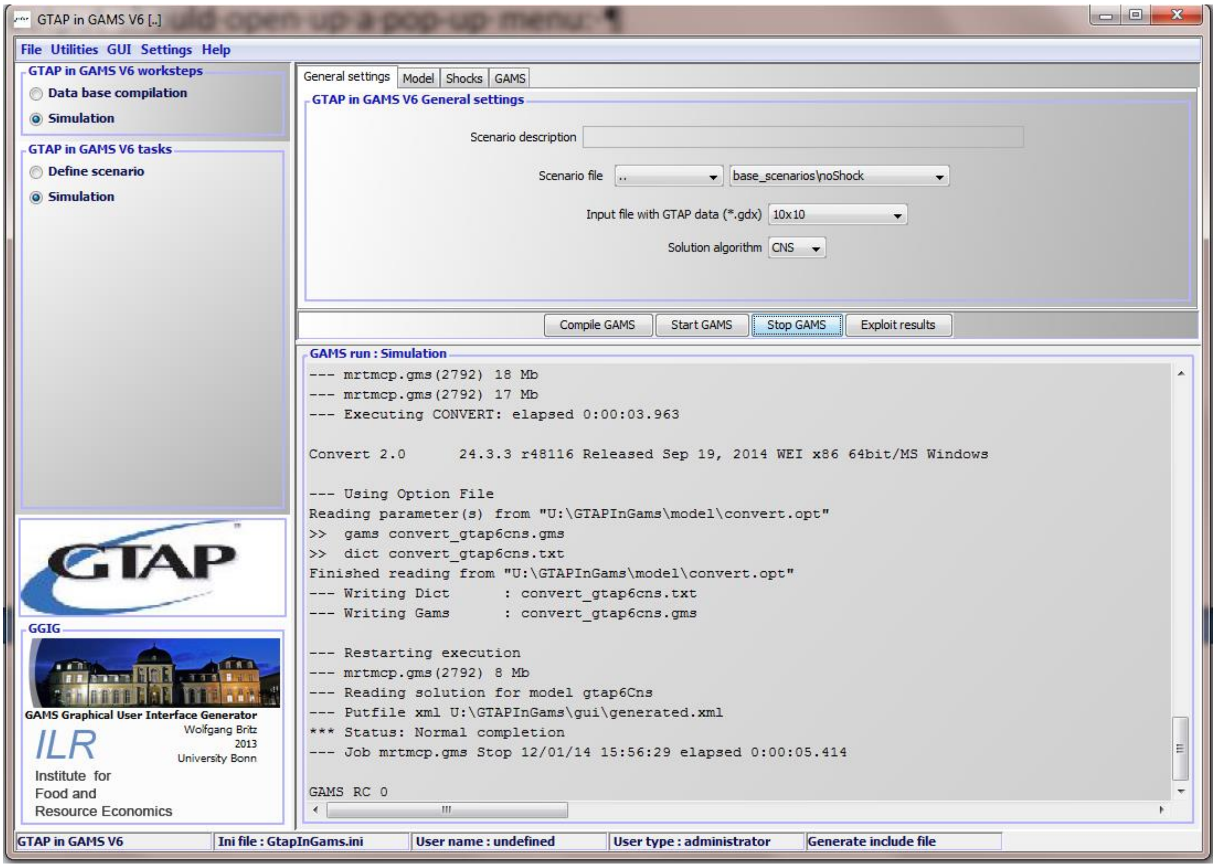
Next, click on the tab "GAMS and R" and either type in text field next to "Path to GAMS.EXE" which "GAMS.EXE" you would like to use. Alternatively, you can use the button to the right of the field to navigate to the directory where "GAMS.EXE" is found via a file selection dialogue. Please do not only 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 by entered by the Java code.



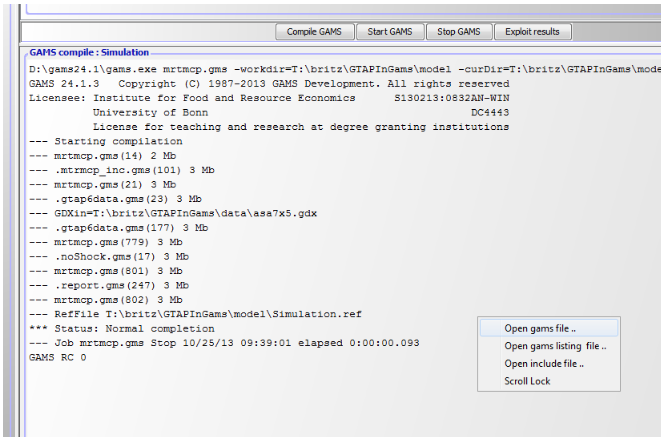
If you are using regularly a text editor, you can register it under "Other options". You might also want to check the box "Clean window with GAMS output with each new GAMS compile/start".



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 dis-aggregation 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 where warning are raised.
- A license either for PATH or CONOPT is required to solve the model. Using the pre-solve algorithm requires CONOPT, as does using the filtering approach when generating a data base.

Data base generation

The GTAPAgg utility, which you should have received together with your GTAP data base, 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). 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.

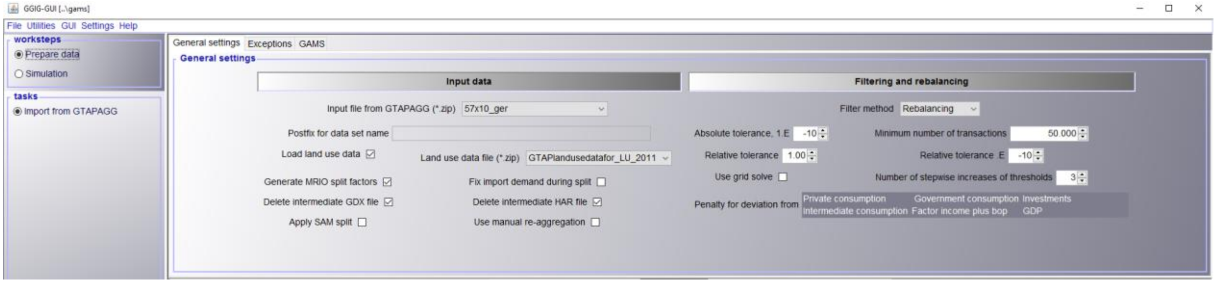
If you want to use the **GTAP-AEZ** extensions, copy the "2007Luv81.tzip" in the data directory and check "Load land use data". In order to work with the GTAP9 version, download the 2011 data set "LAND_USE_140_REG.ZIP from https://www.gtap.agecon.purdue.edu/resources/res_display.asp?RecordID=4844 [3] and copy it to the data directory under the name "2011Luv9.zip". Note that a license for GTAP9 is required. The data set can be combined with **GTAP-Water** data In order to benefit from introducing water and a distinction between irrigated and non-irrigated crops in the model, please use the diagonal version of the data base "GTAP-Water-V9-A.0.zip"[4]. The content of that zip file can be uncompressed data base can be copied into a (separate) GTAPAGG installation. The data might then be aggregated to the non-diagonal make matrix, see below.

[3] The authors would like to thank Farzad Taheripour for releasing an additional data set with Value Added at AEZ level for 2011.

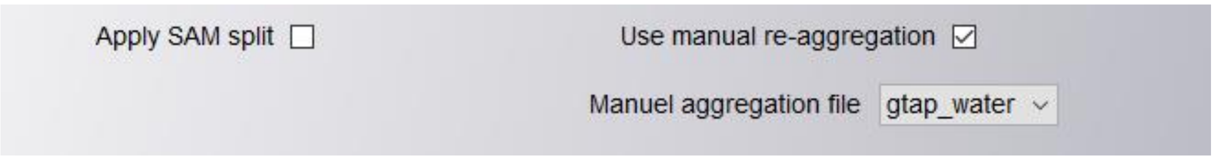
[4] The author would like to thank Iman Haqiqi for useful feedback on the content of the data base and releasing a new version which fixed issues preventing to use the data base with CGEBox.

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.



Post-aggregation of GTAPAgg to yield non-diagonal make matrices



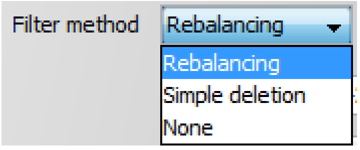
The code of CGEBox clearly separates activities from products and supports non-diagonal make matrices where one activity might produce several outputs and one output might be produced by several sectors. The current GTAPAgg facility does not support such a differentiation. The data driver therefore allows 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 distinctions is now removed: the mapping to new products ending with “n-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.

The intermediate GDX and HAR files can be deleted. Keeping them eases debugging in case something went wrong during data processing.

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.

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 absolution 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.

Rebalancing

The “Rebalancing” option uses more advances tactics to select transactions to delete, and perhaps more importantly, rebalances the resulting SAMs. As with simple deletion, first transactions loaded from the HAR file which in absolute term are below the chosen absolute threshold are removed from the data base. With a value of 1.E-10, that 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. Please note that rebalancing was only tested with CONOPT and that alternative NLP solver such as PATHNLP might not work satisfactory.

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.

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:

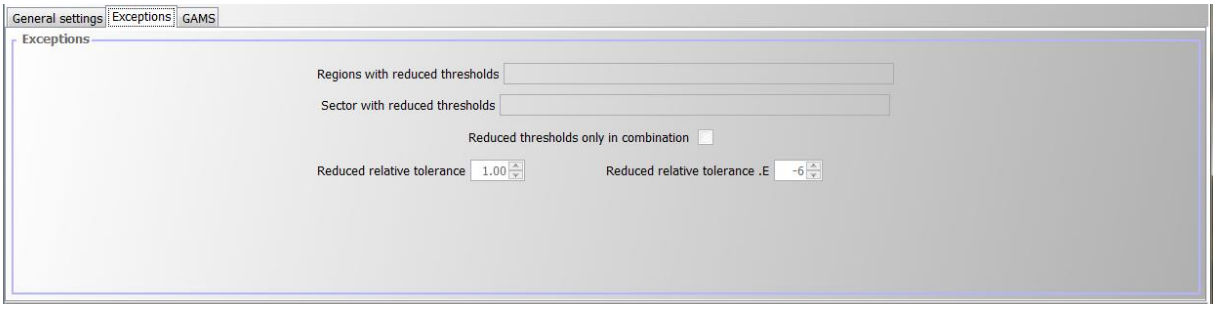
By adding these penalties terms, the non-deleted entries (and thus most important transactions) tend to be scaled upwards. It is generally recommended to use these penalties terms. The code will also scale all non-deleted trade flows to approximately maintain the total volume of international trade and related international transport margins.

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 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.

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.

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.[5] 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 calibration, a benchmark solve and solving the shock.[6] 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 tolerance, minimum # 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 that 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.

[5] 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.

[6] 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[7]

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

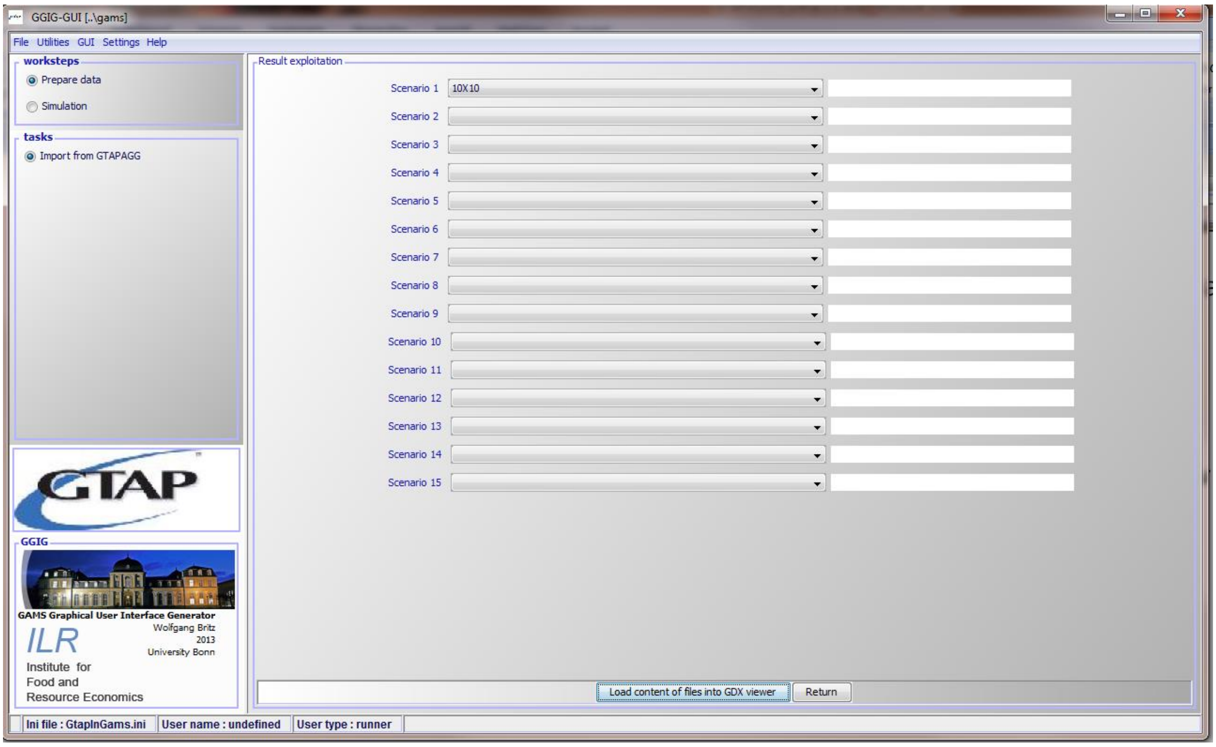
[7] The long solution times occurred under the direct tax cuts. As the standard GTAP model assumes that FDI is taxed at the same rate as domestic one, the direct tax cut impacts fully the expected returns to capital by foreign investors which drive the distribution of foreign savings under the global 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):

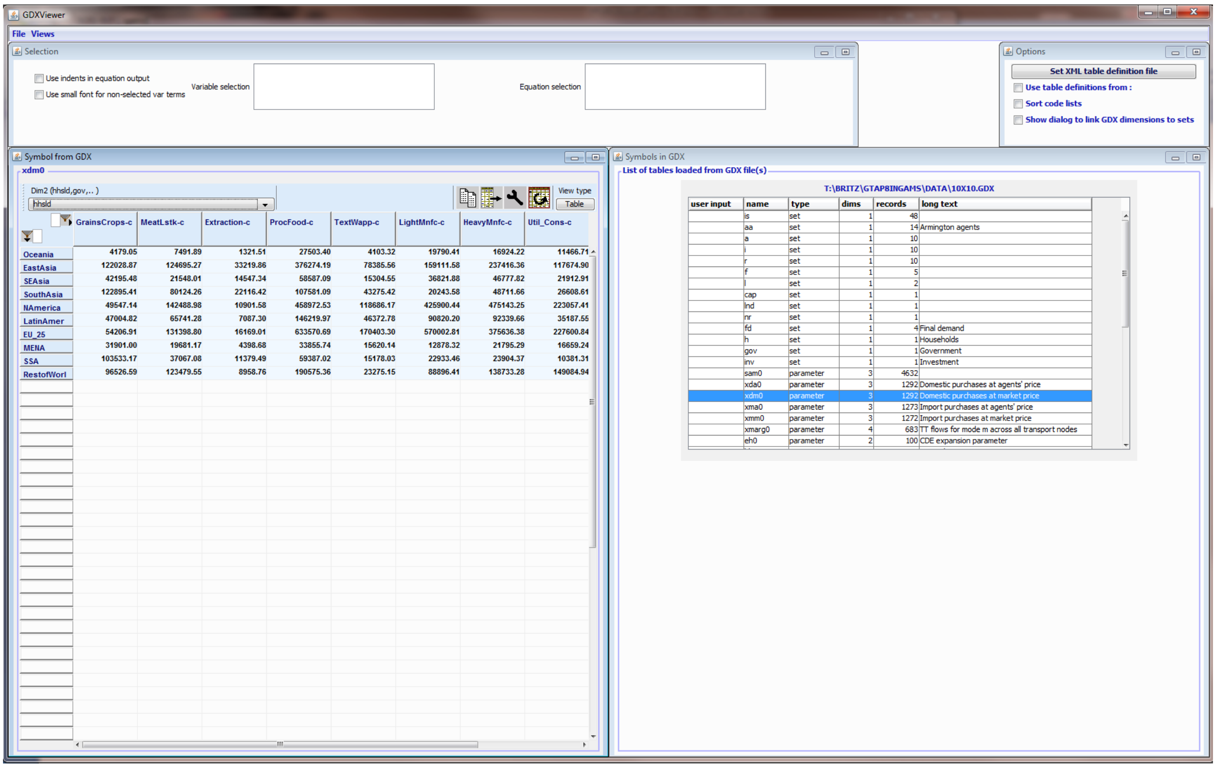
Checks for medium sized models have shown no differences in the results produced by CONOPT or PATH. That seems to indicate that once a model is declared feasible, the results can be used. Care should be clearly given when using PATH as a MCP solver after structural changes to equations: MCP solvers might 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.

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:

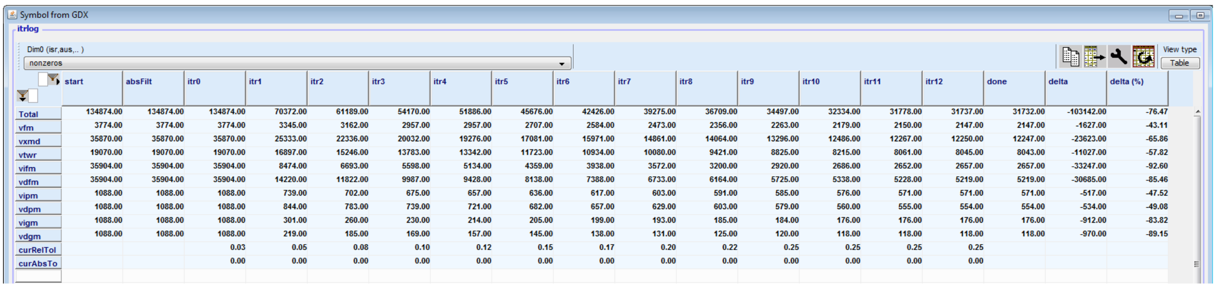


Once you press "Load content of files into GDX viewer", you can inspect the individual symbols used in the model:



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	54170.00	51886.00	45676.00	42426.00	39275.00	36709.00	34497.00	32334.00	31778.00	31737.00	31732.00	-103142.00	-76.47
vfm	3774.00	3774.00	3774.00	3345.00	3162.00	2957.00	2957.00	2707.00	2584.00	2473.00	2356.00	2263.00	2179.00	2150.00	2147.00	2147.00	-1627.00	-43.11
vxmd	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	12250.00	12247.00	-23623.00	-65.86
vtwr	19070.00	19070.00	19070.00	16897.00	15246.00	13783.00	13342.00	11723.00	10934.00	10080.00	9421.00	8825.00	8215.00	8061.00	8045.00	8043.00	-11027.00	-57.82
vifm	35904.00	35904.00	35904.00	8474.00	6693.00	5598.00	5134.00	4359.00	3938.00	3572.00	3200.00	2920.00	2686.00	2652.00	2657.00	2657.00	-33247.00	-92.60
vdfm	35904.00	35904.00	35904.00	14220.00	11822.00	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	682.00	657.00	629.00	603.00	579.00	560.00	555.00	554.00	554.00	-534.00	-49.08
vigm	1088.00	1088.00	1088.00	301.00	260.00	230.00	214.00	205.00	199.00	193.00	185.00	184.00	176.00	176.00	176.00	176.00	-912.00	-83.82
vdgm	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			

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 show in example applications that one might increase considerably simulated welfare from FTAs by simply increasing sectoral and regional detail.

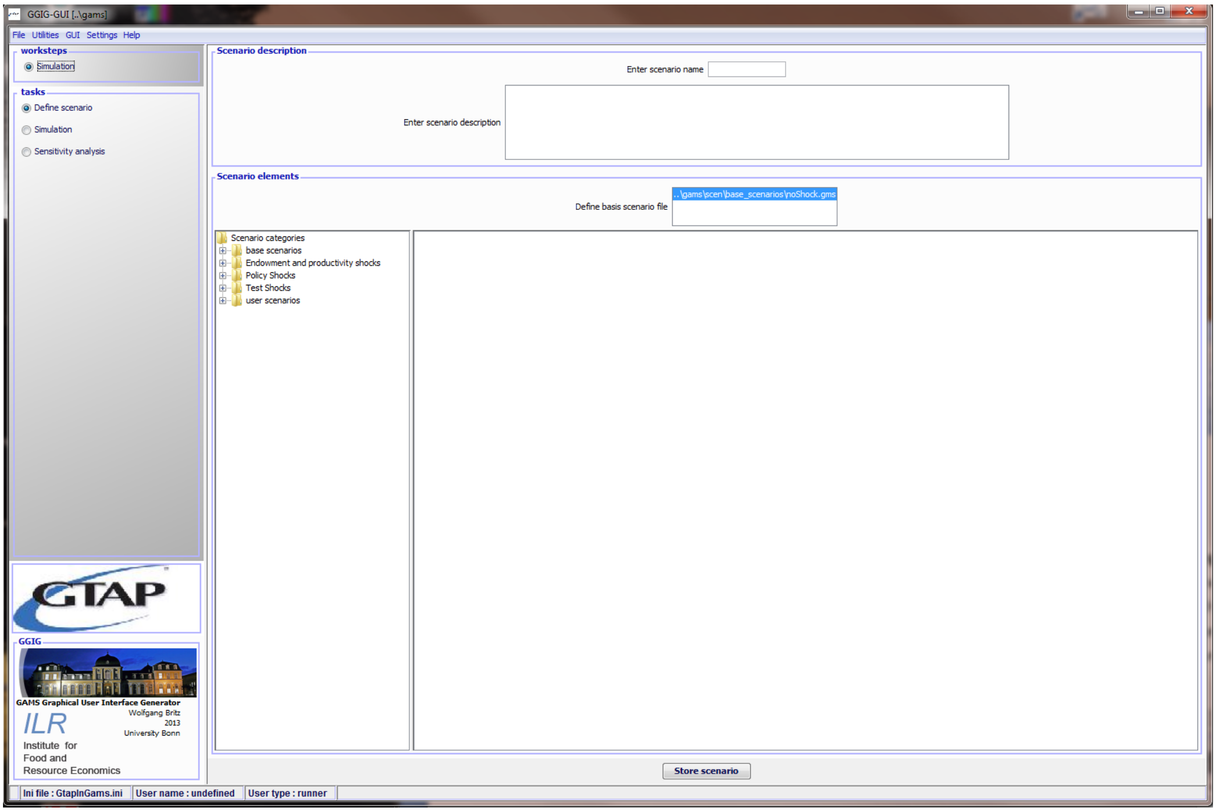
CGEBox tries to soften the decision on what regional and sectoral aggregation level to use based on three features. Firstly, filtering and rebalancing of a data base can help to solve models with rather high number of sectors and regions (but clearly, the filtering process will also introduce aggregation 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 et al. 2015 for discussion of solution times and aggregation bias.

Scenario runs

Defining the scenario

Scenario editor

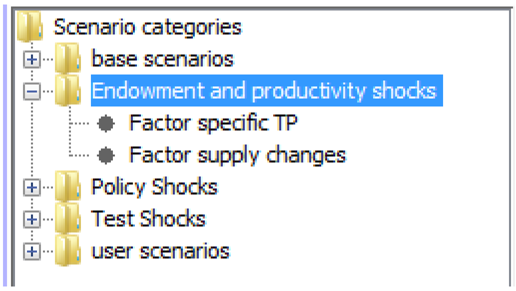
The interface helps with the definition of scenarios – it has a set of example files with shocks which can be combined and edited directly in the GUI. Alternatively, you can use any text editor and define a shock in GAMS and store the file somewhere under "\gams\scen". That is the recommend way to work 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.



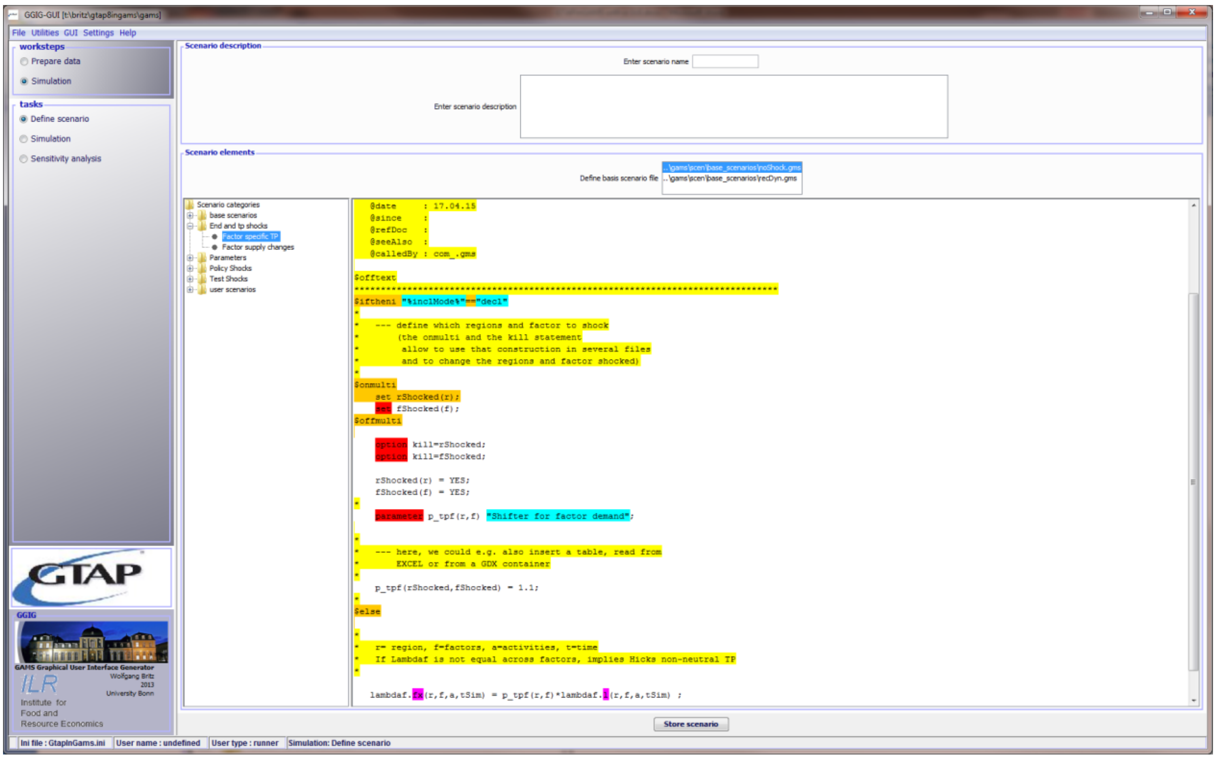
Currently, there are two base scenarios:

- **No shock** – that is simply an empty file
- **RecDyn** – a test implementation for a driver for a recursive-dynamic baseline.

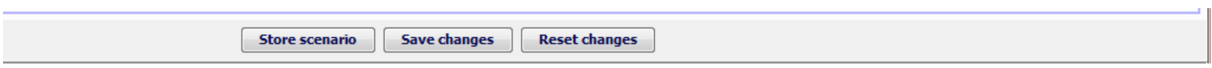
- In order to add the content of pre-defined shock files to your base scenarios, click on a tree element, e.g. "Endowment and productivity shocks". You should be able to see a list of files as below:



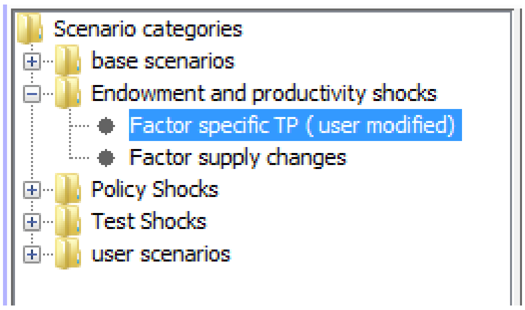
Double-click for example on "Factor specific TP", and the content of that file is loaded in the editor on the right as seen below:



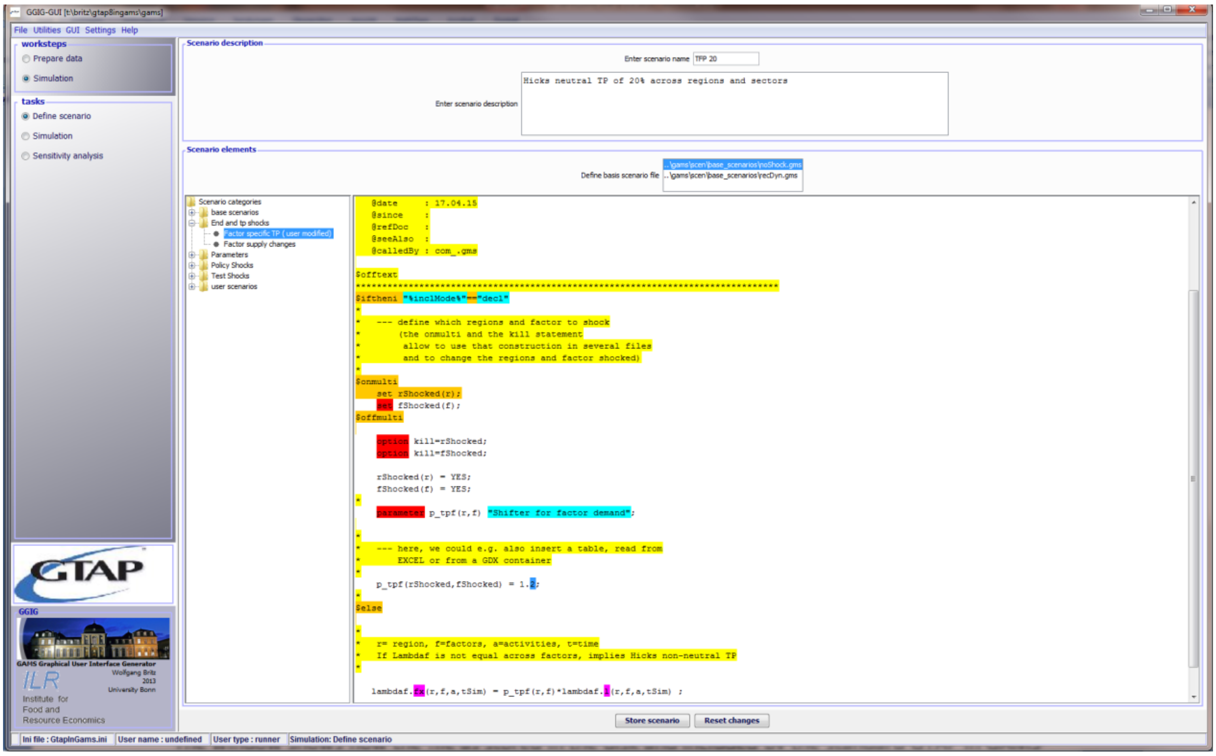
You can now modify the content with the editor by typing directly in the right hand window. Once you shave edited the code, additional buttons will appear:



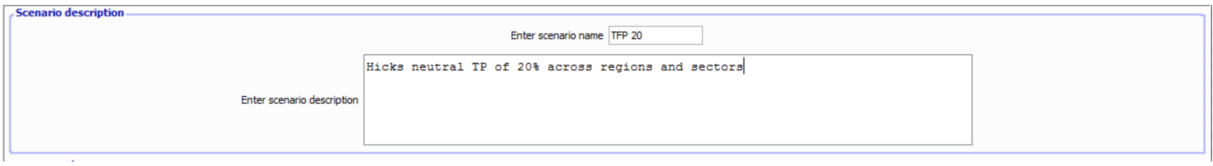
The list with the scenario groups and shock files will now indicate that you have made changes to the shock file.



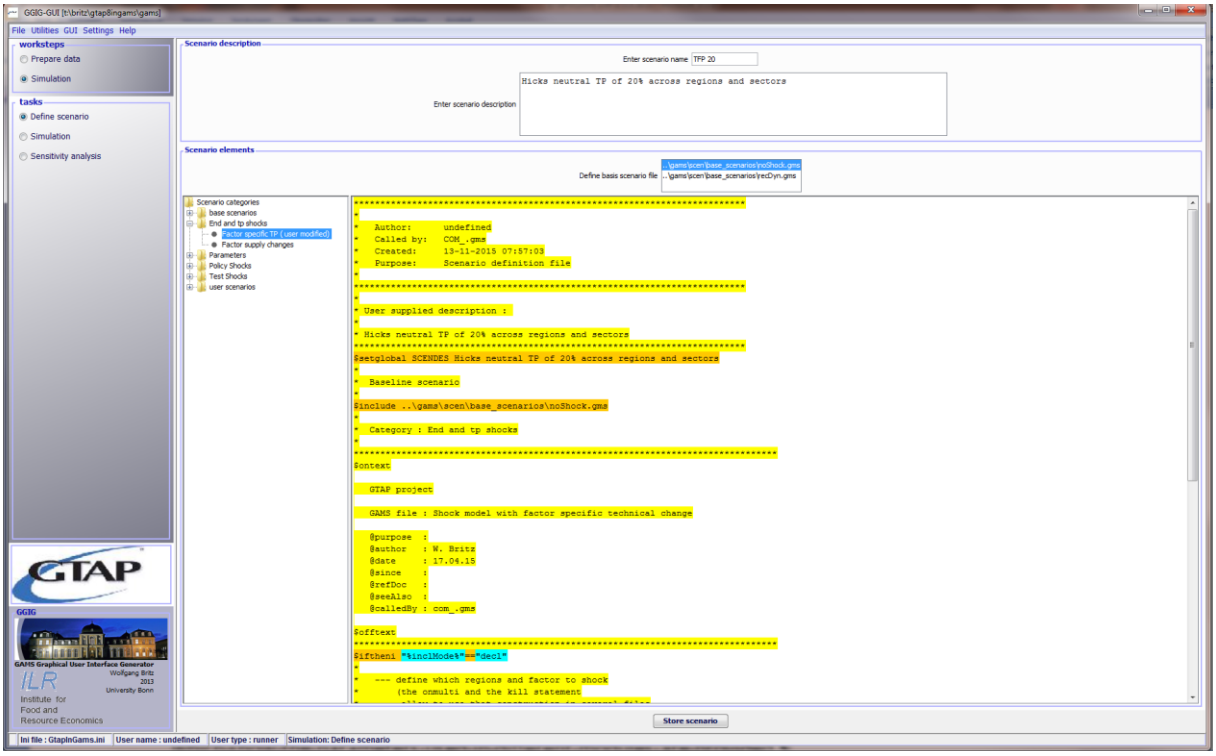
In order to save your edits, press "Save changes". If you do not press "Save changes", your edits won't make it in final shock file generated. You can now double click on further files to add them to your shock file. The files you have currently combined are shown in blue; you can also deselect some 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 "gams\scen\user_scenarios".

Implementation of shock files in GAMS

GAMS does not allow to declare symbols inside loops and similar program flow features. It is however often highly useful to include declarations in a shock file, e.g. to define the regions affected by a shock. The code therefore includes the shock file twice and sets the global parameter "inclMode" to allow the programmer to only compile and execute certain parts of the shock file depending on the mode. These calls are made:

1. Before the simulation loop, with "inclMode" set to "incl":

```
*
* ---- include potential declaration in shock file
*      (file needs a $ifftheni "%inclMode%=="decl" structure to work properly)
*
$setglobal inclMode decl
$include "scen\%shock%"
```

2. Inside the simulation loop for all years beside the "check" case (= benchmark), with "inclMode" set to "run":

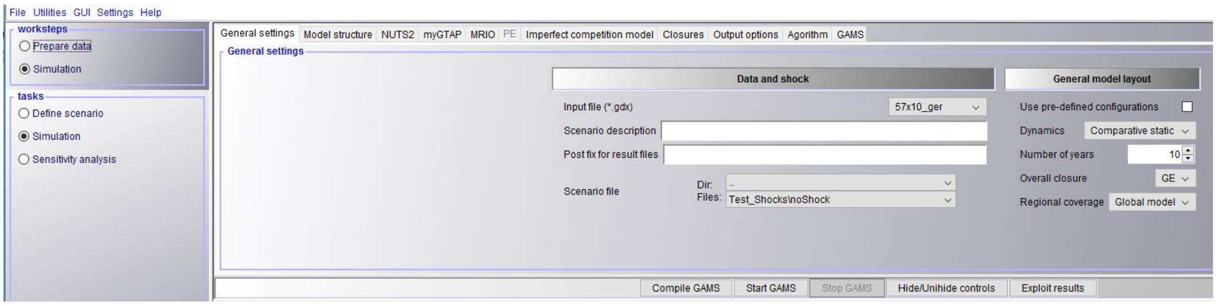
```
   if( not sameas(tsim,"check"),
*
* --- run shock file selected by user
*
$setglobal inclMode run
   $$include "scen\%shock%"
   ) ;
```

A shock file therefore **must** ensure that it properly reflects the two calls. The examples provided with the code in the diretories under “gams\scen” and used with the scenario editor are correctly implemented and can be used as a starting point for own shock files. The example below highlights the important points:

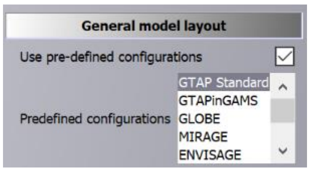
1. Enclose the declaration inside the first part of a $iftheni “%inclMode%”==”decl” - $else $endif structure such that the symbols are declared and potentially initialized with the first call of the file before the simulation loop. Declaration outside of such a structure will lead to a compilation error.
2. As the scenario editor allows to combine several files, make sure that you enclose potential multiple declarations of the same symbol with $onmulti - $offmulti. Use “option kill=” to clean its content to ensure that the shock is applied only to the combination of regions, factors, commodities, agents etc. intended.
3. Make sure that the actual shock is only applied inside the loop to the actual simulated year tSim.

```
$iftheni "%inclMode%"=="decl"
*
*   --- define which regions and factor to shock
*       (the onmulti and the kill statement
*        allow to use that construction in several files
*        and to change the regions and factor shocked)
*
$onmulti
    set rShocked(r);
    set fShocked(f);
$offmulti

    option kill=rShocked;
    option kill=fShocked;

    rShocked(r) = YES;
    fShocked(f) = YES;
*
    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(rShocked,fShocked) = 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)*lambdaf.l(r,f,a,tSim) ;

$endif
```

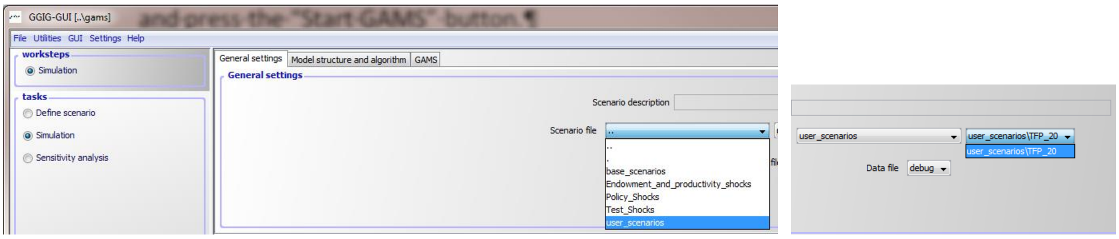
Running the scenario



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

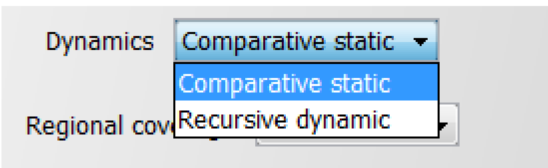


and press the “Start GAMS” button.

That should run your counterfactual against the reference and generate results in a GDX file.

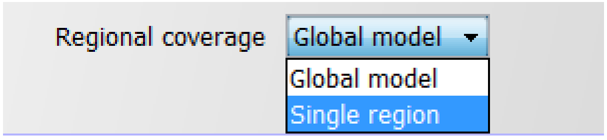
The tab “General settings” gives you the following additional input possibilities:

- Entering a *scenario description*: Normally, the output files generated are named after the scenario file used. That name can be overwritten with content of that 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 modules.
- Choice between the *Comparative static or Recursive dynamic version* of the model:

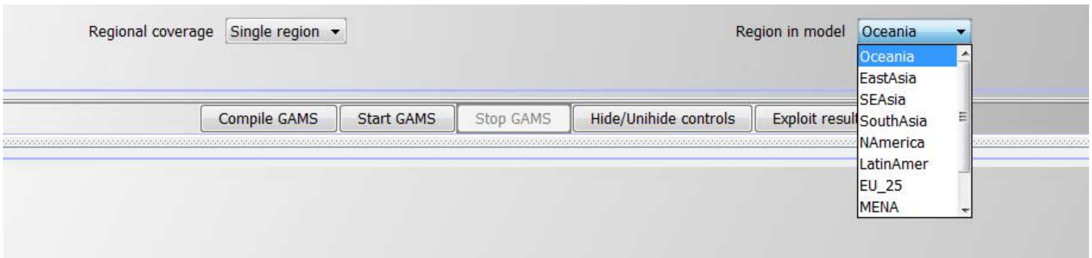


Note that a simple example scenario which shows how to introduce shocks dynamically can be found under “scen\base_cenarios\regDyn.gms”. If the “Recursive dynamic” option is switched on, a drop down box is added which allows defining the number of years:

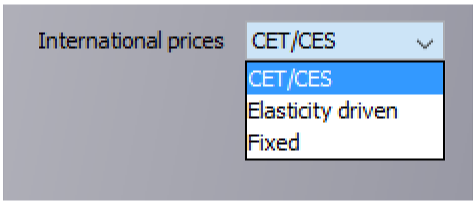
- Choice between the *Global Model or Single Region version* of the model:



If the single region mode is switched on, a drop down box appears where the region can be chosen:



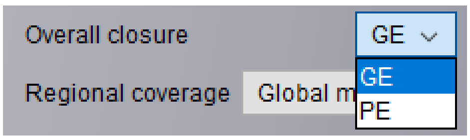
Furthermore, the way international prices and related to that, how import and export flows are treated can be chosen:



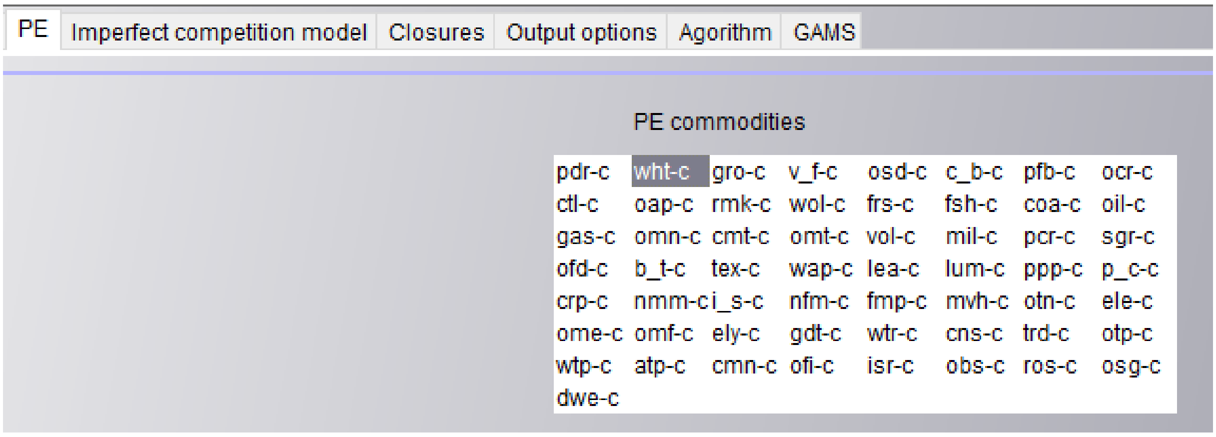
- CET/CES: Imports to other regions react to changes in the region's cif prices according to the share equations of the lower Armington nest at fixed total import quantities and aggregate import prices. If CET elasticities for exporters to the region are not infinite, the lower CET nest at the export 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: the single region against all other regions. That will automatically reflect cases where the small country assumption makes limited sense.

- Choice between the *General Equilibrium* or *Partial Equilibrium closure* of the model:

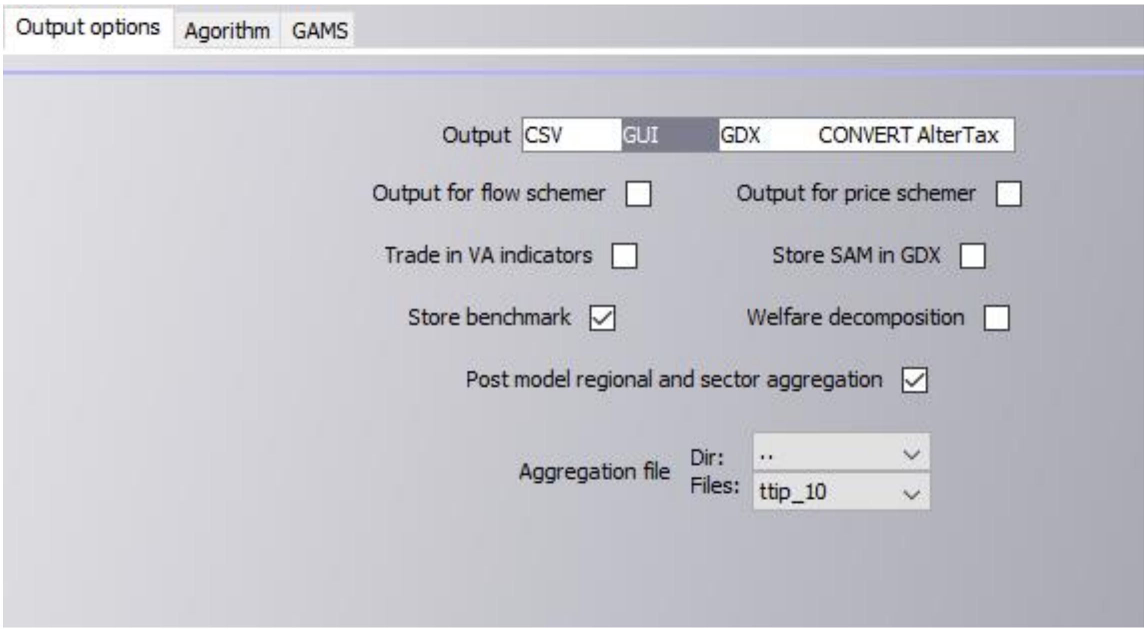


The GE closure is the usual set-up for a General Equilibrium model. The PE closure allows solving for only one; several or all commodity markets while clearing factor market at fixed income. The product markets which clear can be chosen on the tab “PE”:



Please note that the usual welfare interpretation of a CGE is no longer valid if income is exogenous and/or only some markets clear.

Output options



Also, make sure that output for the GUI is selected:

- 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 output for the GUI if selected

- 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.

Usually, “Store benchmark” should be switched on. If you are not interested in a graphical depiction of flows and price changes (see chapter “Overview on existing views for exploitation” below), you can deselect these options to save some memory and speed up post-model processing. The same holds for the decomposition of the welfare changes. For a description of the “Trade in VA indicators”, see the respective section.

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]

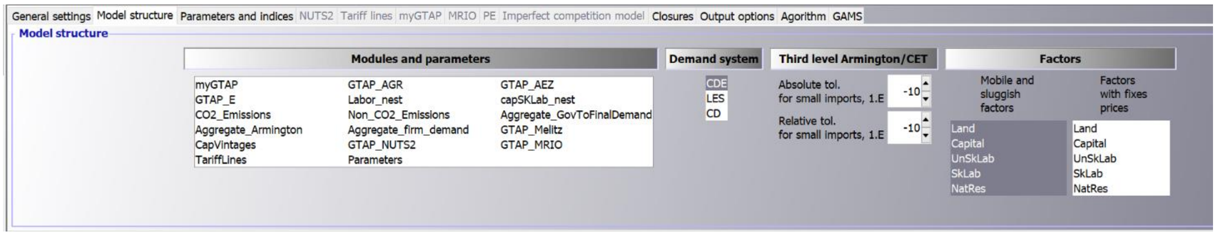
Regions: World | Items: Quantity | Origins: Total

TARIFFS

	Total	Household demand	Government demand	Intermediate demand	Investment demand	Export demand	International transport demand
Total	137582,97 0,41%	31212,10 0,30%	9534,91 0,02%	54631,16 0,01%	12416,49 -0,69%	23091,78 2,03%	591,84 4,30%
Agriculture	2058,32 0,43%	800,55 0,29%	1,87 12,82%	504,73 0,58%	0,57 -0,73%	413,50 0,78%	
Mining and Extraction	5474,23 2,91%	58,66 -1,99%		2484,97 0,57%	1,05 0,20%	1542,53 6,36%	
Light Manufacturing	17319,36 1,02%	3950,65 1,51%	18,43 1,91%	4854,85 0,74%	985,41 -0,50%	5116,43 1,23%	
Heavy Manufacturing	32869,18 0,68%	2062,65 0,66%	53,34 0,27%	15055,72 -0,24%	2527,85 -0,77%	11201,62 2,35%	
Utilities and Construction	11765,99 -0,44%	509,02 0,28%	9,54 -0,57%	3485,60 -0,59%	7415,18 -0,64%	292,50 5,26%	
Transport and Communication	22034,38 0,20%	9171,68 0,21%	67,26 0,88%	10126,40 0,09%	593,31 -0,63%	1725,47 1,05%	591,84 4,30%
Other Services	46061,50 0,01%	14658,88 -0,01%	9384,46 0,01%	18118,90 0,01%	893,11 -1,15%	2799,73 0,40%	
Total	137582,97 0,41%	31212,10 0,30%	9534,91 0,02%	54631,16 0,01%	12416,49 -0,69%	23091,78 2,03%	591,84 4,30%
Agriculture	2058,32 0,43%	800,55 0,29%	1,87 12,82%	504,73 0,58%	0,57 -0,73%	413,50 0,78%	
Paddy rice	21,36 -1,22%	6,28 -0,03%	0,19 -1,34%	11,79 -1,76%		1,55 -1,52%	
Wheat	99,08 -0,12%	12,88 0,28%		26,67 -0,05%		29,76 -0,24%	
Cereal grains nec	124,97 0,45%	48,10 0,76%	0,42 22,49%	17,66 -1,59%		34,18 0,75%	
Vegetables, fruit, nuts	910,70 0,70%	477,79 0,29%		252,22 1,14%		114,20 2,43%	
Oil seeds	133,49	8,28		58,67		33,27	

Model structure, parameterization and factor markets

You might also select from different model setups:



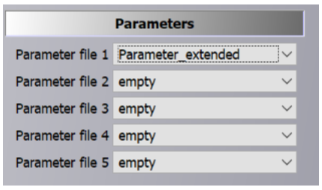
Modules and parameters

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-regional dis-aggregation of production and factor markets in CGEBox", page 113ff) 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.

Non-Default parameters 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:

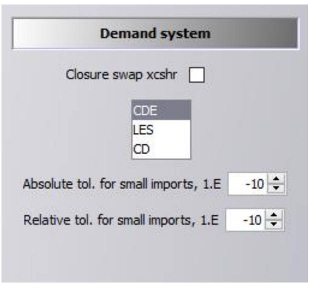


The possibility to use non-default parameters is important as certain model extensions are driven by parameter settings. To give an example: in the standard GTAP model, the transformation elasticity between domestic output and exports is infinite, such that producer prices between the two destinations 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.

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.

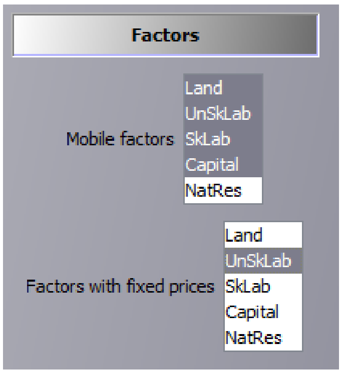


The LES system will determine the marginal budget shares based on income elasticities derived from the parameterization of the CDE demand system. The commitment terms are derived by assuming an 80% share of non-committed income, ensuring that commitments are non-negative and do not exceed 50% of observed 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. 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.

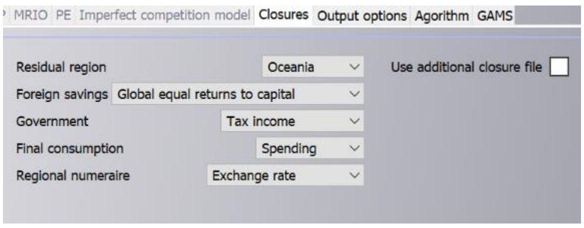
Factors

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.



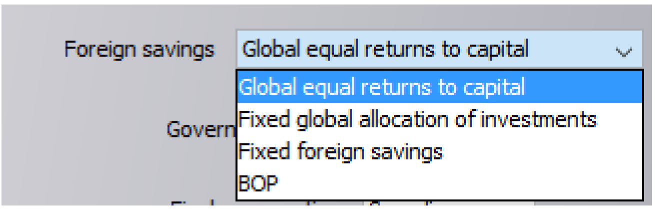
The second selection allows fixing factor prices - either economy wide or sector specific, depending on the choice above – such that their stock is endogenously adjusting. That is an often found solution to model unemployment based on a fixed wage rate. The reader should note that the model can also be parameterized with factor supply elasticities to model e.g. a wage curve approach or to introduce a land supply curve.

Closures

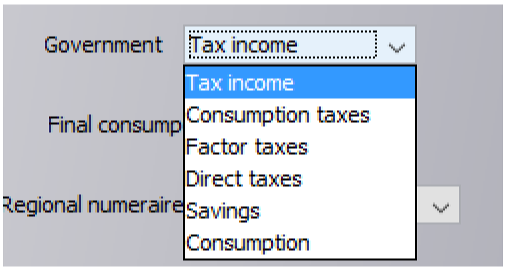


As shown above, the code supports alternative closures different from the ones used in the standard GTAP model:

- 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.

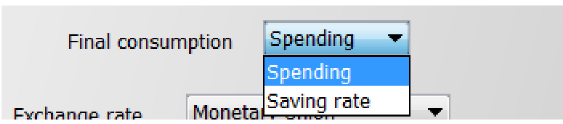


- The standard closure for the **government** account is to calculate tax income at given tax rates, and to let the top level utility function allocate a certain share of regional income to government consumption. The reader should note that there is no immediate relation between changes in tax income 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.

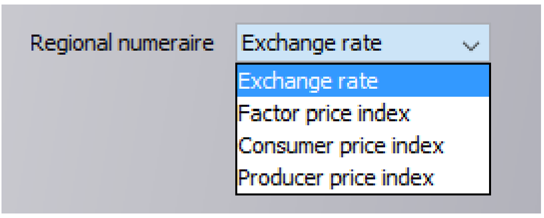


  For the closures modifying taxes can be further detailed by applying the shifters The closures where “savings” and “consumption” adjust are only available with the myGTAP extensions where a separate government account is present.

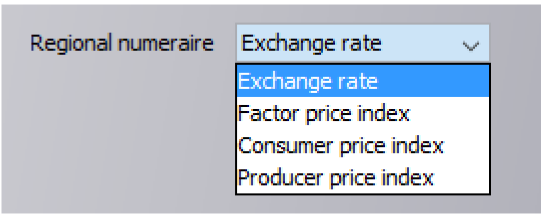
- The standard closure for **private consumptions and saving** is to a have flexible budget shares for private and government consumption and regional savings. The alternative closures allow to fix private consumption spending and to let the saving rate adjust.



- The final closure relates to the **regional numeraire**. The choice of the regional numeraire does not influence the results (quantities, welfare impacts), but simulated prices 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.

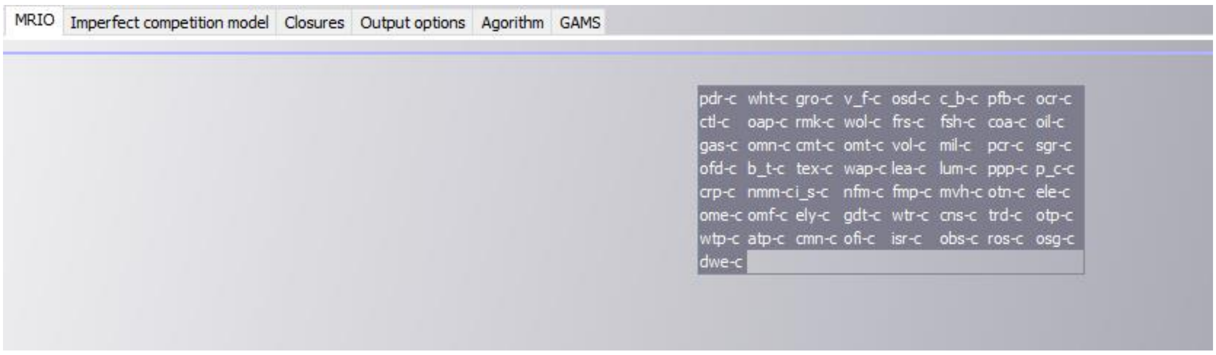
The user can also provide its own closure file to introduce alternative and additional closures.

Numeraires

- The residual region closes global savings (see equations walraseq, savfeq and yieq in "model.gms")
- The numeraire regions and products define a price index. For the model, only the choice of the numeraire regions matters which enter the global factor price index, whereas the numeraire products and regions together define producer and consumer price index which can be used as alternative reginal numeraires and are also reported post-model.
- The regional numeraire will clearly not change relative prices in the model, but price levels. The choice is hence a question of model result analysis.

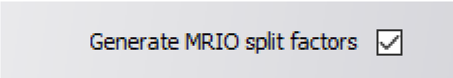
MRIO extension

Currently, the MRIO extension (for details see the section "MRIO extension", page 119ff) 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:



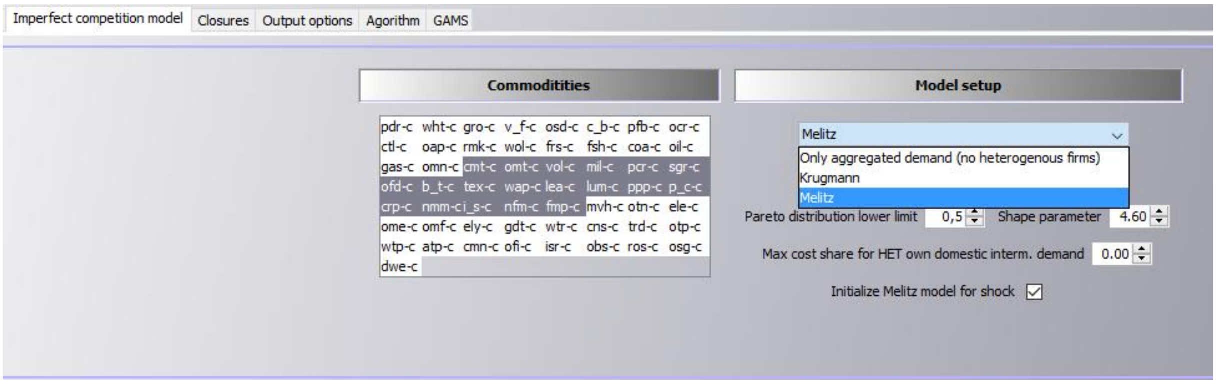
The list of products shown adjusts dynamically with the data base chosen. Hence, check it when switching the data base.

Not that the MRIO extensions requires that MRIO split factors were generated during data base generation:



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:

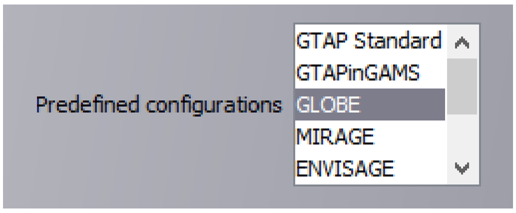
1. "Only aggregated demand, not heterogenous firms" changes the demand structure of the standard GTAP such that only one Armington nest shared by all agents is present. The substitution elasticity is defined as in the case when the full Melitz model is used. The remaining equations of the Melitz model 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. "Krugmann": Monopolisitic competition at industry level, the number of firms in each sector and region is endogenous and defines the number of varities.
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.

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 mimick these model.



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.

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

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.

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

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

CGEBOX

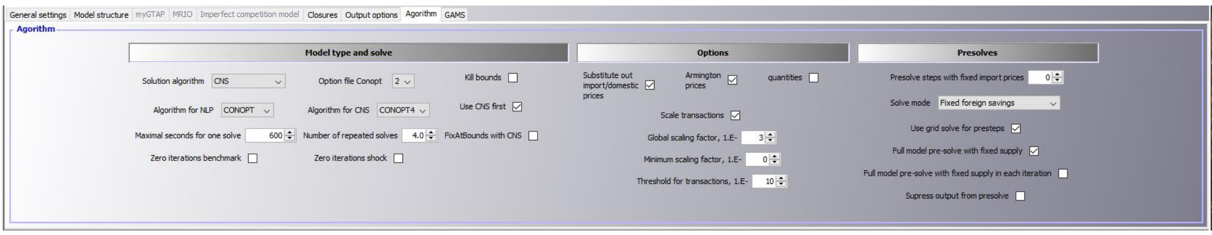
- The GTAP-AGR, GTAP-AEZ and GTAP-E modules are switched on
- Capital vintages are differentiated in comparative static mode, considering 5 years of depreciation

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

Algorithm



The model can be solved either as a constrained system of equations (formally solved with a dummy objective to yield a NLP which might help CONOPT) 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.

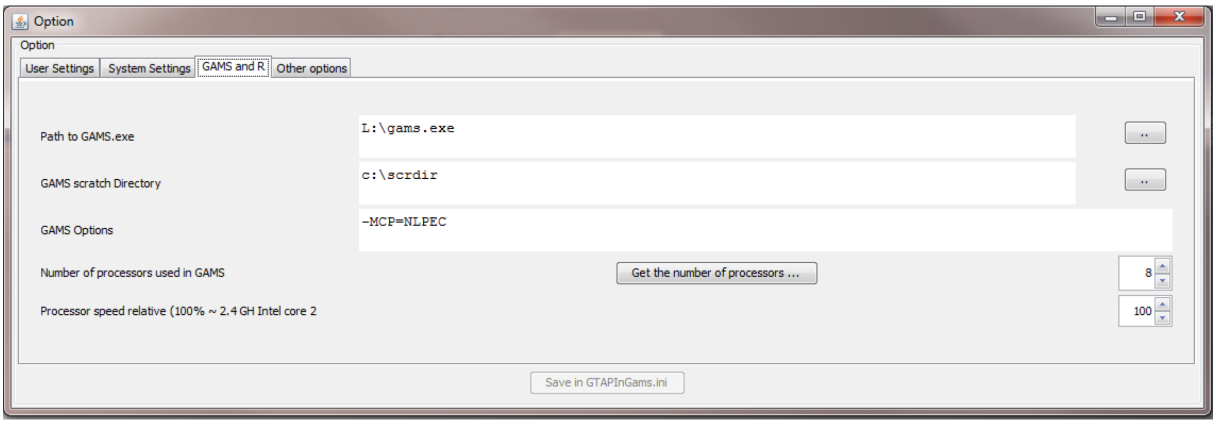
Please note that the default maximal time for model solution of 600 seconds might be too low for very large models.

The settings shown above are recommended defaults. Generally, solving the model as a constrained system of equations CNS - especially in connection with pres-solves, see below – has proven to be generally faster for larger models (> 250.000 transactions in SAM). The CNS solution will 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 a unpaired equations and variables.

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. Tests with a beta version of CONOPT4 which parallelizes e.g. matrix factorization has shown considerable speedups for very large models. The reader should however note the so far CONOPT4 is not officially released and that up to that point, only users highly familiar with the code of the model should use that beta version which shows great potential.

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”.

In order to solve larger models as a MCP, a PATH license is required. Users with a CONOPT license, but no PATH license, can try to solve smaller model as a MCP by using the GAMS provided solver NLPEC (no additional license required). The following settings should be entered in the file “GAMS option” under the “GAMS” tab in the “Settings” dialogue to make NLPEC the solver used for MCP:



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 is repeated several times to inform the individual country CGEs about changes in others regions. The single region models are only introduced as an intermediate steps 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 GTAP model. The layout of the single country models therefore differs somewhat from the usual single country CE 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 one single country model to the next.

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 *indx* 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 estimate 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, ten iterations are recommended as the pre-solves are relatively cheap.
2. *Use of grid solves*. That solves the single country models in parallel with the exemption of the last pre-solve, but exploits less information as results from other countries can only be exploited once all single countries are solved. The grid solve is the recommended option for modern multi-core machines.
3. *Solving the model with fixed supply*. Either only once after the pre-solves, or in each iteration.

CNS models based with CONOPT have proven to benefit much more from the pre-solves than the MCP version. Test have shown that the combination of presolves and CONOPT4 allow sto solve 57x82 models without any filtering in about 10 minutes even under larger shocks.

GAMS

As all symbols (variables, equations) are also stored in a GDX container with GUI output switched on, one might normally switch “solprint” to “Silent” to keep the listing at minimal size. For complex simulation, “OnAfterInfes” offers the possibility to generate a listing if the model could not be solved without infeasibilities in the first try.

Modules

myGTAP module

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 sum of factor and tax income based on modified CD function. That 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) replaces the regional household of the standard GTAP model by separated accounts. We provide in here a close to truthful replication of the myGTAP structure to ease the integration of key data from household surveys in the CGEBox framework which captures also features such as remittance, foreign aid and transfers.

As general with the CGEBox framework, the myGTAP extension can in a modular fashion combined 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.

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 – an often found closure 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.

The factor income reflects ownership shares *shrXf* on factor supply by activity *a* and factor *f*, while direct taxation can reflect differences in tax rates *kappaFH* between households:

```
*
* --- Factor income by households, depends on ownership shares shrXF on factor supply xft / xf
*
 factYHeq(rs(r),h,ts(t)) $ hFlag(r,h) ..

   factYH(r,h,t)/regy.scale(r,t) =e= [   sum( (f,a) $ xfFlag(r,f,a), pf(r,f,a,t)*xf(r,f,a,t)*shrXf(r,h,f,a,t))
                                        - fdepr(r,t)*pi(r,t)*kstock(r,t)*shrDep(r,h,t) ] / regy.scale(r,t);
*
* --- Direct taxation of household, depends on ownership shares shrXF on factor supply xft / xf
*     and household specific direct tax rates kappaFH
*
 dirTaxEq(rs(r),f,h,ts(t)) $ hFlag(r,h) ..

   dirTax(r,f,h,t)/regy.scale(r,t) =E=
             sum( a $ xfFlag(r,f,a),  kappaFH(r,f,h,t) * pf(r,f,a,t)*xf(r,f,a,t)*shrXf(r,h,f,a,t))/regy.scale(r,t);
```

Compared to the standard model, household income reflects additionally remittances (*remiH*,*remoH*), foreign capital income transactions (*fyiH*,*fyoH*) and transfers between households (t*rnh*) and from government to the household (*trng*)[8]:

```
*
* --- household income definition
*
 hInceq(rs(r),h,ts(t)) $  sum(f,shrXf(r,h,f,t)) ..

   hInc(r,h,t)/regy.scale(r,t) =E=[  factYH(r,h,t)
                     + trng(r,h,t)
                     + remiH(r,h,t)*remihShrCor(t)*lcu(r,t)
                     + fyiH(r,h,t)*fyihShrCor(t)*lcu(r,t)
                     + sum(h0, trnh(r,h0,h,t))
                     - sum(f, dirTax(r,f,h,t))
                     - remoH(r,h,t)*lcu(r,t)
                     - fyoH(r,h,t)*lcu(r,t)
                     - sum(h0, trnh(r,h,h0,t))]/regy.scale(r,t);
```

Note that all transaction to and from abroad are corrected from international to local currency. That relative exchange is unity in the standard closure of the model with a uniform global factor price index. An alternative closure allows to fix the factor price index in each country and close the BOP based on flexible exchange rates in which case the exchange rate are endogenous.

The correction for the income shares *remihShrCor* from abroad is discussed below. If the user wants to use for some all countries the standard approach with solely one private household, the code handles that case based on the “h” set of different households.

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

---

[8] The reader is reminded that CGEBox is completely written in level. Small caps in variable or parameter names do not indicate percentage changes.

```
*
* --- saving of private households, share of household income
*
 rsavCeq(rs(r),h,ts(t)) $ sum(f,shrXf(r,h,f,t)) ..
   rsavC(r,h,t)/regy.scale(r,t) =e= betaSH(r,h,t) * hInc(r,h,t)/regy.scale(r,t);
*
* --- consumption by private households, share of household income
*
 yceq(rs(r),h,ts(t)) $ sum(f,shrXf(r,h,f,t))  ..
   yc(r,h,t)/regy.scale(r,t)     =e= betaPH(r,h,t)  * hInc(r,h,t)/regy.scale(r,t);
```

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

```
*
* --- Weighted average of household specific direct tax rates
*
  kappafEq(rs(r),f,ts(t)) ..

    kappaf(r,f,t) * sum(a $ xfFlag(r,f,a), pf(r,f,a,t)*xf(r,f,a,t))
            =E= sum(h $ hFlag(r,h),dirTax(r,f,h,t));
```

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
*
 rsaveq(rs(r),ts(t)) ..
   rsav(r,t)/regy.scale(r,t) =e= (sum(h,rsavC(r,h,t))+rsavG(r,t))/regy.scale(r,t);
```

Remittance outflows from the households are defined as a share of labor income:

```
*
* --- remittance outflows in local currency (global value*lcu) as a share of labor income
*
 remohEq(rs(r),h,ts(t)) $ hFlag(r,h) ..

    remoh(r,h,t)*lcu(r,t)/regy.scale(r,t)
       =E= remohShr(r,h,t)

          *  sum((sameas(f,lab),a) $ xfFlag(r,f,a), pf(r,f,a,t)*xf(r,f,a,t)*shrXf(r,h,f,a,t))/regy.scale(r,t);
```

Remittance inflows are a share of global remittance or fixed in which case the shares are endogenous:

```
*
* --- remittance inflows in global currency are a share of the global inflows
*
 remihEq(rs(r),h,ts(t)) $ hFlag(r,h) ..

    remih(r,h,t)/regy.scale(r,t)
       =E= remihShr(r,h,t)  * sum( (rp,h0) $ hFlag(r,h0), remoH(rp,h0,t))/regy.scale(r,t);
```

In order to ensure the global balance in case of endogenous shares, the following equation is introduced:

```
*
*   --- ensure global balance, only active of remih is fixed
*
 remihbalEq(ts(t))  ..

    sum( (r,h) $ hFlag(r,h), remih(r,h,t))*remihShrCor(t)/sum(r, regy.scale(r,t))
     =E=  sum( (r,h) $ hFlag(r,h), remoH(r,h,t))/sum(r, regy.scale(r,t));
```

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

```
*
* --- foreign global capital outflows in local currency (fyoh*lcu) as share of capital income
*
 fyohEq(rs(r),h,ts(t)) $ hFlag(r,h) ..

    fyoh(r,h,t)*lcu(r,t)/regy.scale(r,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)*shrXf(r,h,f,a,t))/regy.scale(r,t);
*
* --- foreign global capital inflows in global currency are a share of the global inflows
*
 fyihEq(rs(r),h,ts(t)) $ hFlag(r,h) ..

    fyih(r,h,t)/regy.scale(r,t)
       =E= fyihShr(r,h,t) * sum( (rp,h0) $ hFlag(r,h), fyoH(rp,h0,t))/regy.scale(r,t);
```

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
*
 fyihbalEq(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));
```

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

```
*
* --- share of incoming foreign aid in total global aid
*
 aidiEq(rs(r),ts(t))  ..

    aidi(r,t)/regy.scale(r,t) =E= aidiShr(r,t) * sum(rp, aido(rp,t))/regy.scale(r,t);
*
* --- foreign transfer in local currency as share of factor income
*
 aidoEq(rs(r),ts(t))  ..

    aido(r,t)*lcu(r,t)/regy.scale(r,t) =E= aidoShr(r,t) * facty(r,t)/regy.scale(r,t);
```

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

```
*
* --- ensure that sum of aidi = sum of aido, only active if aidi is fixed
*
 aidiBalEq(ts(t))  ..

    sum(r, aidi(r,t))*aidiShrCor(t)/sum(r, regy.scale(r,t))
     =E= sum(r, aido(r,t))/sum(r, regy.scale(r,t));
```

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

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 shrHhslds(r,hhslds,*,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 (hslds (rows) - > hhslds (colums))
*               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 goverments (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_hhslds(*) / 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 Armginton demands for the different households are adjusted to simultaneously exhaust the given SAM entries and the final consumption total of the dis-aggregated 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
*
   shrHHslds(r,"agr",agr,t) = 1;
```

Which can be combined which 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,hhslds), remoH.l(r,hhslds,t) - remiH.l(r,hhslds,t));
  p_surPlus("wor","wgt",t) = sum( (r,hhslds), remoH.l(r,hhslds,t) + remiH.l(r,hhslds,t));

  remoH.l(r,hhslds,t)  $ p_surPlus("wor","wgt",t) =  remoH.l(r,hhslds,t) * ( 1 - p_surPlus("wor","rem",t)/p_surPlus("wor","wgt",t));
  remiH.l(r,hhslds,t)  $ p_surPlus("wor","wgt",t) =  remiH.l(r,hhslds,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 Armginton 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 Armginton demands by households must exhaust given
*        private consumption
*
     sum(i, xa(rs,i,hhslds,t)) =E= yc.l(rs,hhslds,t);
```

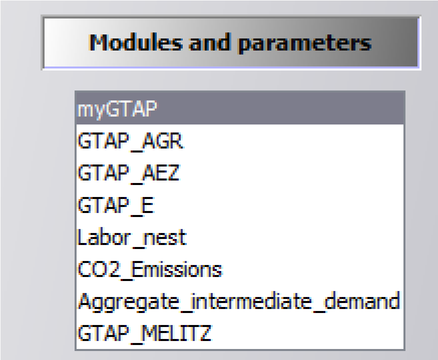
These are combined into a simple model which minimizes squared relative differences from the estimates resulting from applying the expansion elasticities:

```
 e_hdpXA ..

    v_hdp * sum( (rs,i,hhslds,t) $ xa.l(rs,i,hhslds,t),1)

          =E=  sum( (rs,i,hhslds,t) $ xa.l(rs,i,hhslds,t),
              sqr( (xa(rs,i,hhslds,t)-xa.l(rs,i,hhslds,t))/xa.l(rs,i,hhslds,t)));
```

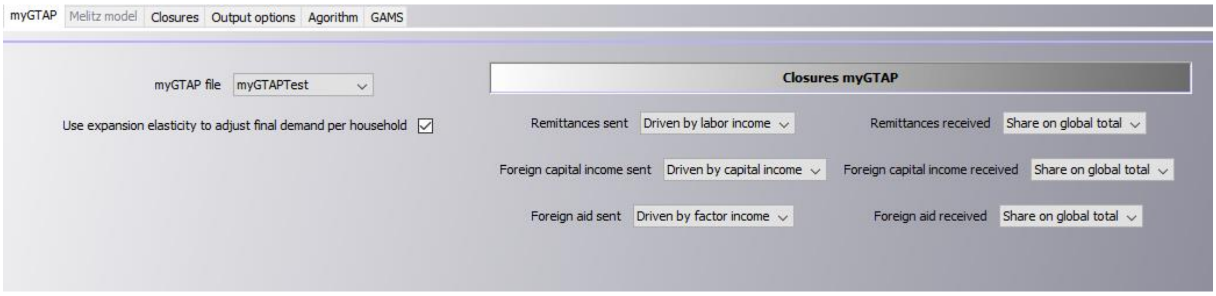
Graphical User Interface

Options

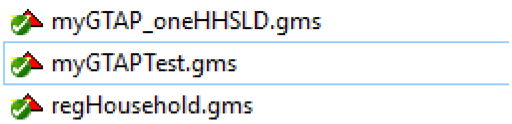
In order to use the extension, the “myGTAP” module must be switched on the interface:



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 dis-aggregation and international transactions as discussed above. These files are stored in “gams\scen\myGTAP”, currently three test files are provided in the repository:



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:

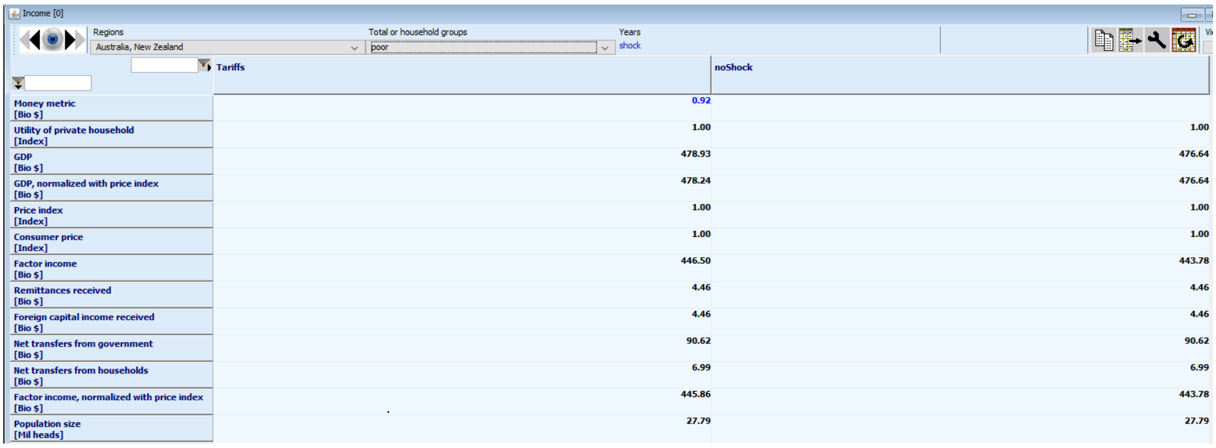
```
$iftheni %1==decl
  $$onempty
   set set_hhslds(*) //;
  $$offempty
$endif
```

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_hhslds(*) / agg/;

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

    table shrHhslds(r,hhslds,*,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:



Income [0] — Regions: Australia, New Zealand; Total or household groups: poor; Years: shock

	Tariffs	noShock
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 | View type: Table

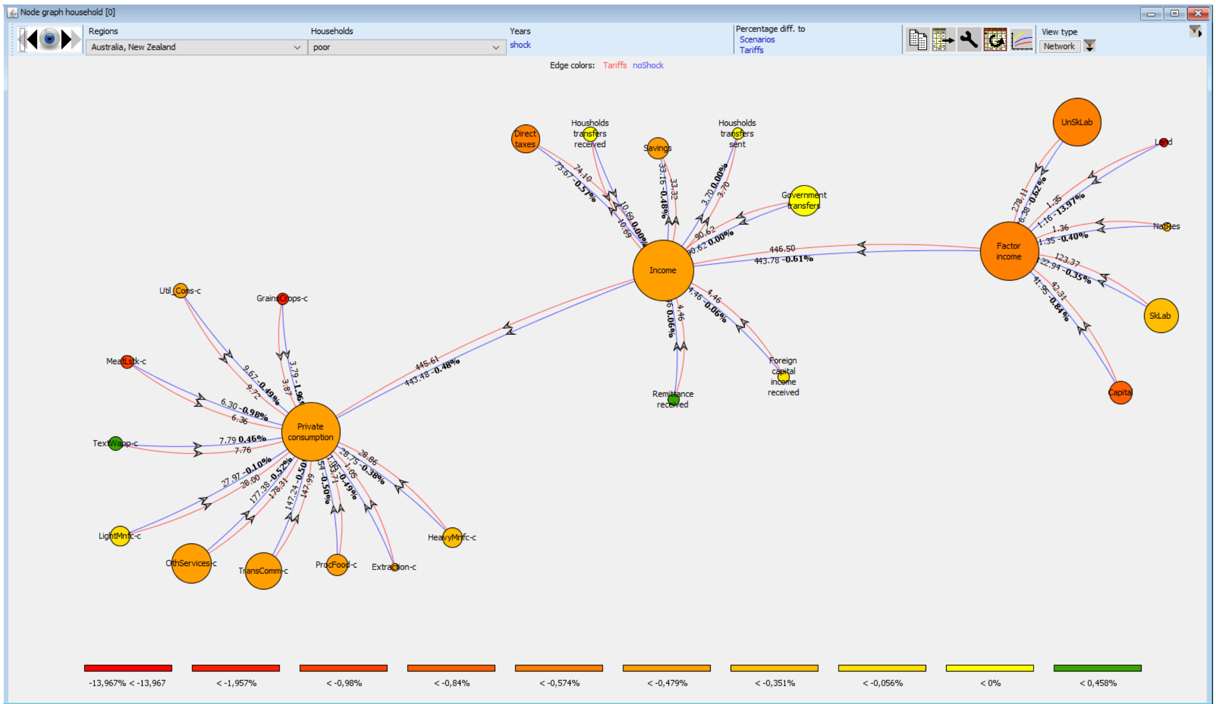
	Tariffs Tariffs			noShock noShock		
	Private consumption hhsld	poor poor	rich rich	Private consumption hhsld	poor poor	rich rich
Total totP	529.07 0.63%	337.15 0.50%	191.92 0.86%	525.76	335.49	190.28
Total totP	529.07 0.63%	337.15 0.50%	191.92 0.86%	525.76	335.49	190.28
Grains and Crops GrainsCrops-c	4.72 -0.02%	3.01 -0.03%	1.71 -0.01%	4.72	3.01	1.71
Livestock and Meat Products MeatLstk-c	8.06 -0.40%	5.14 -0.51%	2.92 -0.20%	8.10	5.17	2.93
Mining and Extraction Extraction-c	1.38 0.53%	0.88 0.41%	0.50 0.73%	1.37	0.88	0.50
Processed Food ProcFood-c	33.61 0.42%	21.43 0.31%	12.19 0.62%	33.47	21.36	12.11
Textiles and Clothing TextWapp-c	8.55 2.73%	5.45 2.60%	3.10 2.96%	8.32	5.31	3.01
Light Manufacturing LightMnfc-c	33.59 1.89%	21.41 1.75%	12.19 2.13%	32.97	21.04	11.93
Heavy Manufacturing HeavyMnfc-c	29.38 1.14%	18.72 1.01%	10.66 1.38%	29.05	18.54	10.51
Utilities and Construction Util_Cons-c	12.04 0.57%	7.67 0.44%	4.37 0.81%	11.97	7.64	4.33
Transport and Communication TransComm-c	180.82 0.54%	115.22 0.40%	65.60 0.78%	179.85	114.76	65.09
Other Services OthServices-c	216.92 0.45%	138.23 0.32%	78.69 0.69%	215.94	137.79	78.15

Factor income [0]

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

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



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.l(r,f,"hhsld",t) = kappaf.l(r,f,t);
    kappafH.l(r,f,hhslds,t) $ (not shrHhslds(r,hhslds,"dirTax",t)) = kappaf.l(r,f,t);
    kappafH.l(r,f,hhslds,t) $ shrHhslds(r,hhslds,"dirTax",t) = kappaf.l(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.l(r,h,t)   = sum(f, pft.l(r,f,t)*xft.l(r,f,t)*shrXf(r,h,f,t))
                          - fdepr(r,t)*pi.l(r,t)*kstock.l(r,t)*sum(sameas(f,cap), shrXf(r,h,cap,t))
                        + sum((fnm,a), pf.l(r,fnm,a,t)*xf.l(r,fnm,a,t)*shrXf(r,h,fnm,t));

    factY.l(r,t)   = sum(f, pft.l(r,f,t)*xft.l(r,f,t)) - fdepr(r,t)*pi.l(r,t)*kstock.l(r,t)
                    + sum((fnm,a), pf.l(r,fnm,a,t)*xf.l(r,fnm,a,t));
    factYH.l(r,"hhsld",t)   = factY.l(r,t);
 *
 *  --- direct taxes by household
 *
     dirTax.l(r,f,h,t) =
             +    (    sum(sameas(fm,f),    kappafh(r,fm,h,t)  * pft(r,fm,t)*xft(r,fm,t)*shrXf(r,h,fm,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.l(r,f,"hhsld",t) =
             +    (    sum(sameas(fm,f),    kappaf(r,fm,t)  * pft(r,fm,t)*xft(r,fm,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 piad by the household types are updated:

```
    kappaFh.l(r,f,hhslds,t) $ dirTax.l(r,f,hhslds,t)
       =  kappaFh.l(r,f,hhslds,t)/[sum(hhslds0,dirTax.l(r,f,hhslds0,t))/dirTax.l(r,f,"hhsld",t)];

    dirTax.l(r,f,hhslds,t) =
            +    (    sum(sameas(fm,f),    kappafh(r,fm,hhslds,t)  * pft(r,fm,t)*xft(r,fm,t)*shrXf(r,hhslds,fm,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 goverment: 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
*       goverment 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) $ xd.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);

);
```

References

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

Trade in VA indicators

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 Multiplicator 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 multiplies $(I-A)^{-1}$:

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

Total Value added	**37**

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.

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,"V",f,a,"dom",%1,"%version%")/p_results(rs,"V","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%"));
```

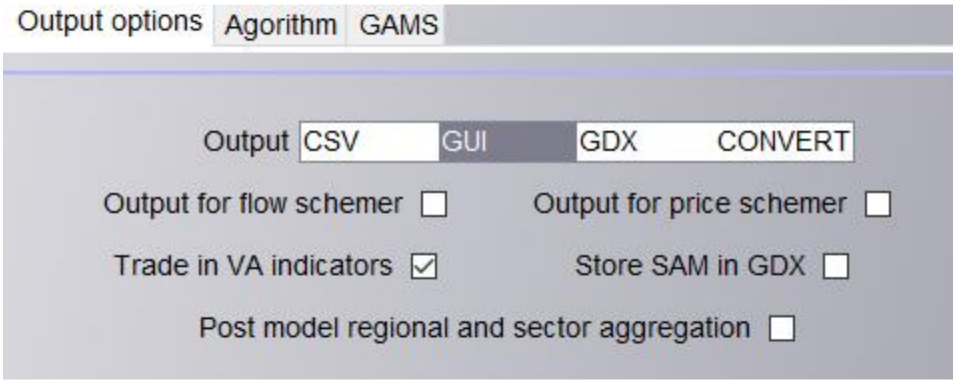
Using these multipliers and export volumes allows to calculate the VA contribution of exports:

```
*
*         --- calculate income by primary factor embodied in trade
*
          p_results(rs,"V",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,"V",j,"va","tradeInVa",%1,"%version%") = sum(f, p_results(rs,"V",j,f,"tradeInVa",%1,"%version%"));
```

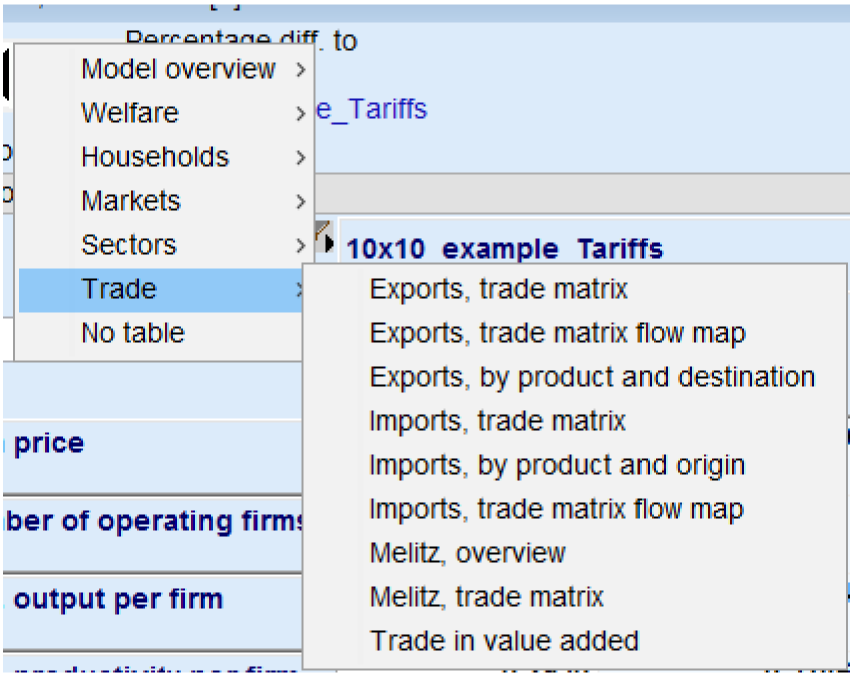
Further code lines not shown here aggregate the information over regions and commodities.

GUI

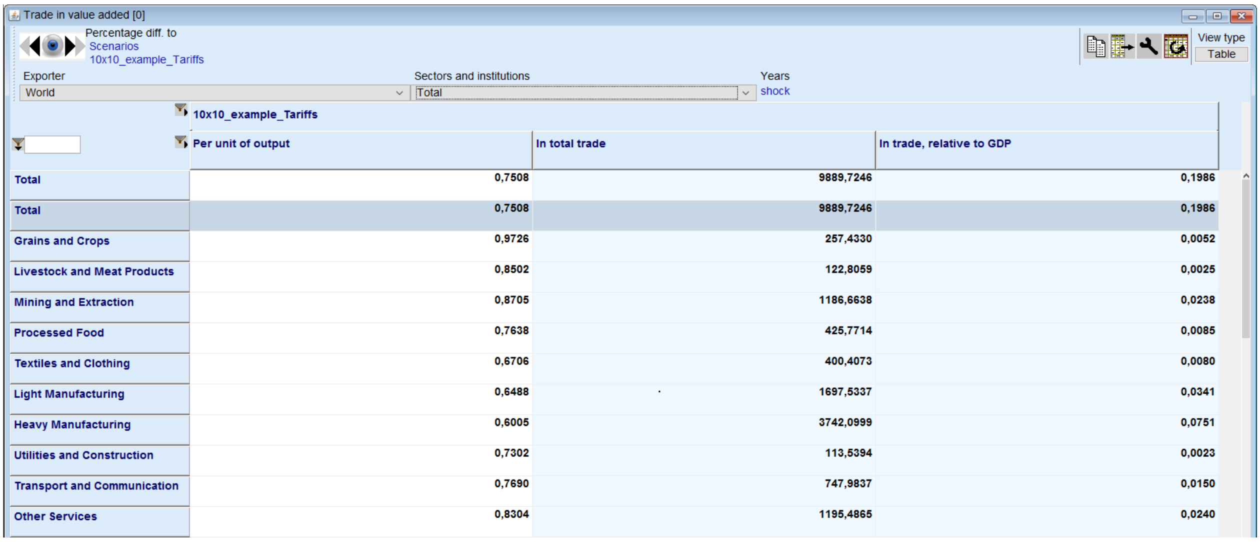
In order to generate the information, the "GUI" output option must be switched on as well as the "Trade in VA indicators" checkbox:



That combination produces a table under the theme “trade”:

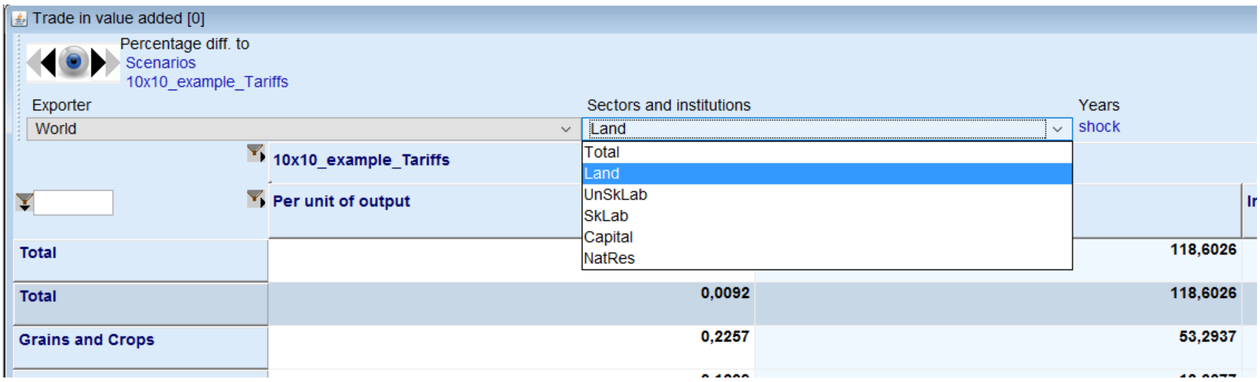


Which reports both the multipliers and the contribution of trade to value added. Additionally, the share from export related value added on total income is reported:



	10x10_example_Tariffs		
	Per unit of output	In total trade	In trade, relative to GDP
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:



Altertax

Background

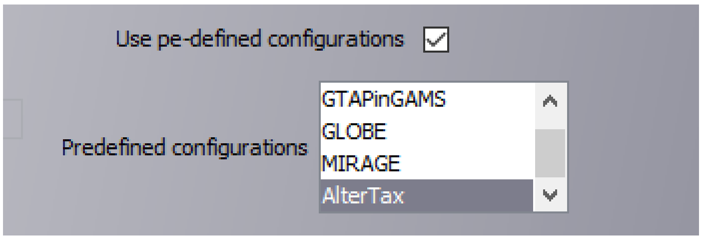
There are various settings where it makes sense to construct a new benchmark from the GTAP data base before shocking the model instead of comparing different shocks against each other where all comprise the same changes against the benchmark. One option to construct a new benchmark changes first entries 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 simulation model itself. That 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. Conceptually, that is straightforward as the accounting identities underlying a balanced SAM must also be fulfilled by a simulation with the CGE itself. The 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. That brief documentation will discuss the basic methodology and technical implementation of AlterTax in CGEBox.

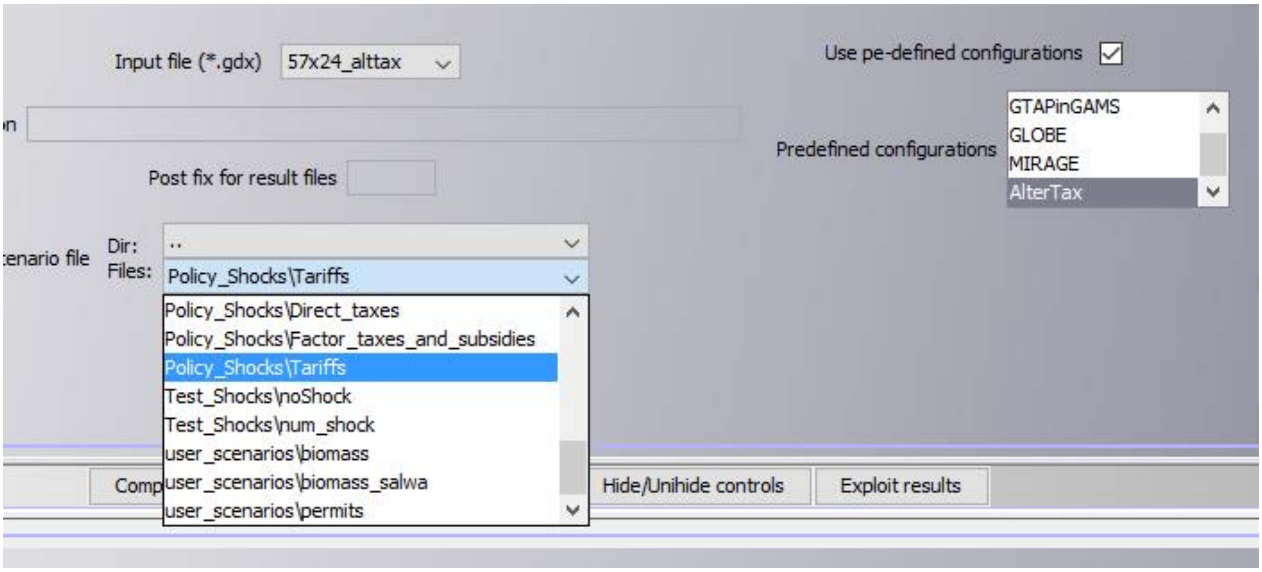
Technical realization

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:



2. 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.

Pre-configuration

The pre-configuration changes the following elements comparted 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.

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;
```

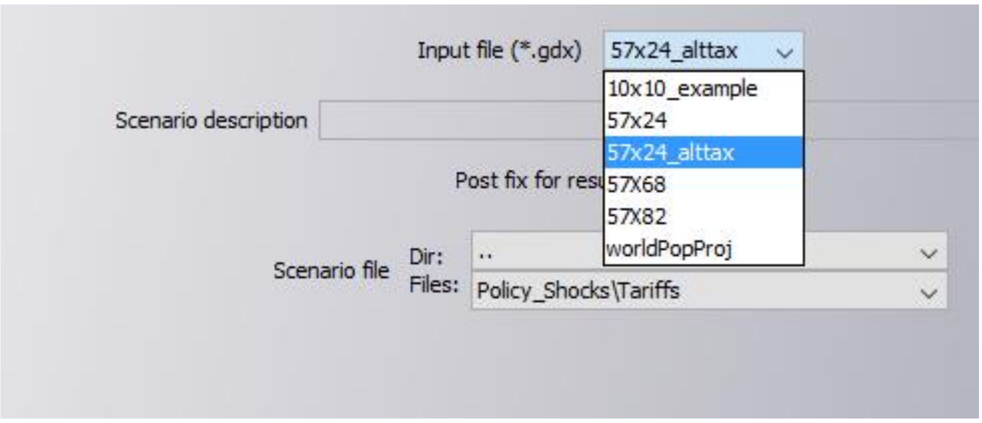
The update symbols are stored to a new data file:

```
execute_unload "%datdir%\%dataset%%postfix%_alttax.gdx", is, aa, a, i, r, f, l, cap, lnd, nr, fd, h, gov, inv,
   sam0, xda0, xdm0, xma0, xmm0, xmarg0,
   eh0, bh0, sigmap0, sigmav0, sigmai0, sigmam0, 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
*$ifi "%filterMethod%"=="Rebalancing" itrlog,keepCor.l
*$ifi "%loadLandUse%"=="on" landUse,AEZ,land_cover
   $$ifthen defined s_meta
     s_meta
   $$endif
;
```

Note that the AEZ implementation does not yet work properly.

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:



Sub-regional dis-aggregation of production and factor markets in CGEBox

Background and motivation

Especially question around use of immobile natural resources such as land, minerals or water often ask for an analysis at regional level, especially if environmental consequences are to evaluated and externalities have a regional character. It is therefore not uncommon to find sub-regional detail in CGE work at single country level. We present in here a rather simplistic approach to add such detail to the CGE modelling framework.

Methodology

We implement a rather basic dis-aggregation as a starting point for further work:

1. Outputs are assumed homogenous at national level and no intra-national trade margins are introduced. Consequently, there is one uniform output price faced by all regions in the same nation.
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.

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)* depict 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) = yes*.

In order to avoid problems with a simply linear aggregation from regions to nations, a CES-aggregator is used in the code:

```
xsNateq(rs(r),i,ts(t)) $ (sum(r_r(subr,r) $ sum(a,xFlag(subr,a,i)),1) $ diag(i) $ (xs.range(r,i,t) ne 0) $ disr(r) ) ..

   ps(r,i,t) =E= sum(r_r(subr,r) $ sum(a,xFlag(subr,a,i)),
       gfi(subr,i,r,t)*ps(subr,i,t)**(1-omegasi(r,i)))**(1/(1-omegasi(r,i)));

xpNateq(rs(r),a,ts(t))  $ (sum(r_r(subr,r) $ xpFlag(subr,a),1) $ (not diag(a)) $ disr(r) ) ..

  px(r,a,t) =E=  sum(r_r(subr,r),
       gfa(subr,a,r,t)*px(subr,a,t)**(1-omegasa(r,a)))**(1/(1-omegasa(r,a)));
```

Where *omegasi* and *omegasa* either aggregate product or activities from region to nation, both set to 10. That means 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 input use in each region and sector to national level:

```
xapNateq(rs(r),i,a,ts(t)) $ (xaFlag(r,i,a) $ disr(r)) ..

   xa(r,i,a,t)/xa.scale(r,i,a,t)  =e= sum(subr $ r_r(subr,r), m_xa(subr,i,a,t))/xa.scale(r,i,a,t);
```

An additional equation was added which distributes the total national factor xft(disr) to the regional factor xft(subr):

```
xftRegEq(rs(r),fm,ts(t)) $ (xftEqFlag(r,fm) $ pft.range(r,fm,t) $ subr(r) )  ..
   0 =e=
*
*      --- "sluggish" factor supply based on CET function, factor is part of top level nest
*
        (xft(r,fm,t)/xft.scale(r,fm,t)
            -  sum(r_r(r,disr),gfr(r,fm,disr,t)*xft(disr,fm,t)/xft.scale(r,fm,t) * (pft(r,fm,t)/pft(disr,fm,t))**omegafr(disr,fm))

            )  $ (sum(r_r(r,disr),omegafr(disr,fm)) ne inf)
*
*      --- fully mobile factor: prices are equal across regions
*
      +  (pft(r,fm,t) - sum(r_r(r,disr),pft(disr,fm,t))) $ (sum(r_r(r,disr),omegafr(disr,fm)) eq inf);

      ;
```

Clearly, if the transformation elasticity across regions *omegafr* is set to zero, the regional factor stock is fixed. The last line in the equation depicts the case of inifinite 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
*
pftNateq(rs(r),fm,ts(t)) $ (xftEqFlag(r,fm) and pft.range(r,fm,t) and disr(r)) ..

  0 =E=
*
*        --- sluggish factor mobility across regions, dual price aggregator
*
        (pft(r,fm,t) -   sum(r_r(subr,r) $ xftEqFlag(subr,fm),
                              gfr(subr,fm,r,t)*pft(subr,fm,t)**(1+omegafr(r,fm)))**(1/(1+omegafr(r,fm)))

             ) $ (omegafr(r,fm) ne inf)
*
*        --- fully mobile factors across regions, uniform prices => phyiscal aggregation
*
      +  (xft(r,fm,t)/xft.scale(r,fm,t) - sum(r_r(subr,r), xft(subr,fm,t))/xft.scale(r,fm,t)
            ) $ (omegafr(r,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 equation are changed to aggregate over sub-regions, factor use for each activities and matching prices are defined in two additional equations:

```
pfNatEq(rs(r),f,a,ts(t)) $ (xfFlag(r,f,a) $ disr(r) ) ..

   pf(r,f,a,t) * xf(r,f,a,t) =E= sum(r_r(subr,r) $ xfFlag(subr,f,a), pf(subr,f,a,t)*xf(subr,f,a,t));


xfNatEq(rs(r),f,a,ts(t)) $ (xfFlag(r,f,a) $ disR(R)) ..

   xf(r,f,a,t) =E= sum(r_r(subr,r) $ xfFlag(subr,f,a), xf(subr,f,a,t));
```

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) $ fcttsY0(r,f,a) = fcttsY0(r,f,a)/sam0(r,f,a) ;
 fcttx.l(r,f,a,t0) $ fcttxY0(r,f,a) = fcttxY0(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.

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/caprird_e.htm), regional SAMs at NUTS2 level and matching national ones for the EU member and candidates countries were compiled (Ferrari et al. 2012). The SAMs feature somewhat limited sector detail (Agriculture, Forestry, Other primary, Food 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. 2010, Britz 2012) 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, Britz et al. 2015). 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:

```
$ifi %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  "Severen tsentralen"
   BG330000  "Severoiztochen"
```

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,omt,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,in22)), 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 which ensures 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:

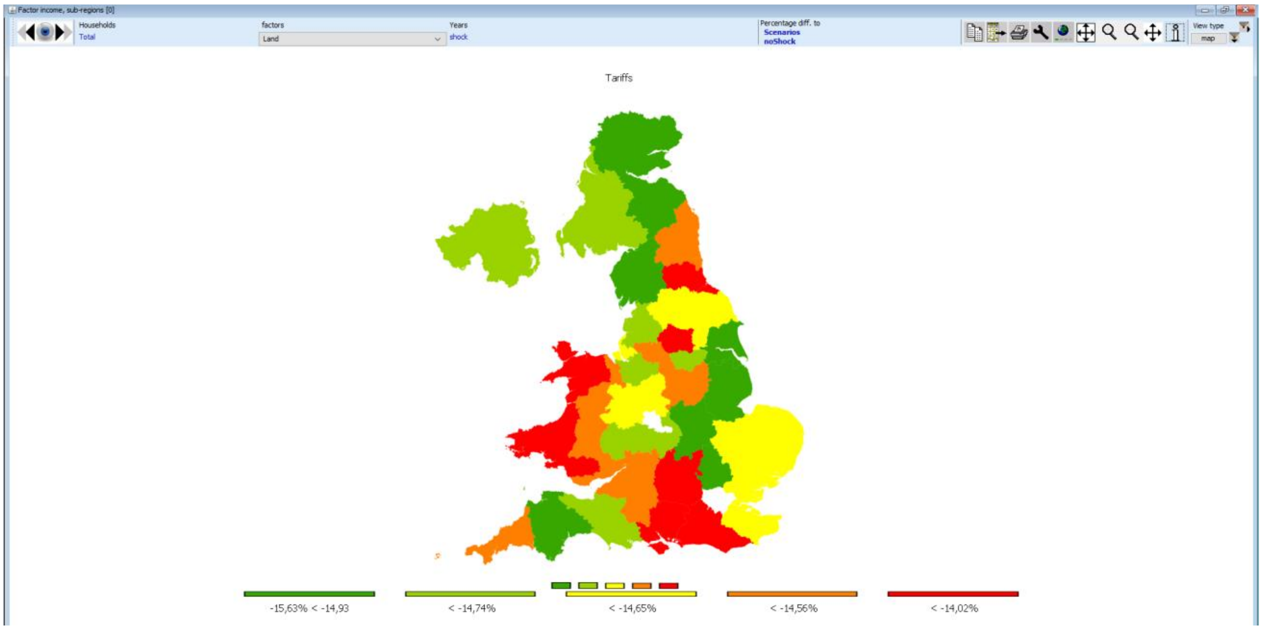
```
sigmav(subReg,a)  = sum(subReg_r(subReg,rs), sigmav(rs,a));
sigmand(subReg,a) = sum(subReg_r(subReg,rs), sigmand(rs,a));

omegafr(rs,fm)          = 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.

Result exploitation

So far, there is only one specific view which uses a map with regional detail:



References

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

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).

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

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

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

MRIO extension

- Wolfgang Britz and Salwa Haddad, 14.12.2016 –

Background

There is growing interest both with regard 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.

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 export flows must be exhausted. We apply a Highest Posterior Density (HPD) estimator to derive consistent shares based on the following equations.

We allow during re-balancing that the total import demand and the domestic demand change for the agents change for each product, while their sum is maintained. However, the user can also fix total import (and thus implicitly total domestic demand) during the consistency step.

If domestic demands by individual agents change, we need to ensure that the total domestic output sold is still exhausted by the agents' summed up demand, ensured by the following equation:

```
*
* --- sum of agent's domestic demand at market prices it not allow to change
*
 e_xd(rC(r),iC(i)) $ sum(aa, xdm0(r,i,aa))   ..

    sum(aa, xdm0(r,i,aa)) =E= sum(aa $ xdm0(r,i,aa), v_xdm(r,i,aa));
```

Secondly, each agent's summed up demand for domestic and import use, both a market and agent prices, need to be maintained:

```
*
* --- xmm and xdm for the sectors must be maintained
*
 e_xmd(rc(r),ic(i),aa) $ (xmm0(r,i,aa)+xdm0(r,i,aa)) ..

    xmm0(r,i,aa) + xdm0(r,i,aa) =E= v_xmm(r,i,aa) $ xmm0(r,i,aa)
                                  + v_xdm(r,i,aa) $ xdm0(r,i,aa);
*
* --- Armington demand must be maintained
*
 e_arma(rc(r),ic(i),aa) $ ((xma0(r,i,aa)+xda0(r,i,aa)) $ (not sameas(aa,"int"))) ..

    (xma0(r,i,aa) + xda0(r,i,aa)) =E=
       (v_xmm(r,i,aa)*(1+mintx(r,i,aa))) $ xmm0(r,i,aa)
     + (v_xdm(r,i,aa)*(1+dintx(r,i,aa))) $ xdm0(r,i,aa);
```

A further equation ensures that estimated total import demand at market prices *v_mm*, i.e. at cif values plus import taxes, by each MRIO agent is exhausted by a share of the bi-lateral imports at world market price (after trade margins) SAM0(r,rp,i) plus bi-lateral import tariffs (IMPTXY0(rp,r,i):

```
*
* --- the import demand of each agent must be exhausted by its bi-lateral demand
*
  e_xmm(rC(r),iC(i),mrioA) $ sum(mrioA_aa(mrioA,aa), xmm0(r,i,aa))   ..

     v_xmm(r,i,mrioA)

        =E= sum(rp, (sam0(r,rp,i)+imptxY0(rp,r,i)) * v_share(r,i,rp,mrioA));
```

Finally, these shares need add up to unity such total imports and import taxes are exhausted:

```
 e_addUp(rc(r),ic(i),rp) $ (sam0(r,rp,i)+imptxY0(rp,r,i)) ..

     sum(mrioA $ sum(mrioA_aa(mrioA,aa), xmm0(r,i,aa)), v_share(r,i,rp,mrioA)) =E= 1+v_slackMM(r,i,rp);
```

Note that a small slack, bounded by +/- 1.E-7 is added to the RHS to account for any numerical accuracies in the in-going global SAM.

As we solve for the shares for total intermediate demand only, we need to ensure that the total import demand over all sectors is equal to that sum:

```
*
* --- import demand of sector must add up to total intermediate demand
*     (as sum of mm + dm per sector is fxied, that also implies that the
*      the sum of md(a) = md("int"))
*
  e_xmmInt(rc(r),ic(i)) $ sum(a,xmm0(r,i,a)) ..

    v_xmm(r,i,"int") =E= sum(a $ xmm0(r,i,a), v_xmm(r,i,a));
```

The HPD estimator minimizes squared relative differences from the given split factors and from total import demand by agent:

```
*
* --- HPD estimator
*
  e_hpdmm .. v_hpdmm * (   sum((rC(r),iC(i),rp,mrioA_aa(mrioA,aa)) $ (xmm0(r,i,aa) and (sam0(r,rp,i)+imptxY0(rp,r,i))),1)
                         + sum((rC(r),iC(i),aa)    $ xmm0(r,i,aa),0.1))

                  =E=    sum((rC(r),iC(i),rp,mrioA_aa(mrioA,aa)) $ (xmm0(r,i,aa) and (sam0(r,rp,i)+imptxY0(rp,r,i))),
                             sqr(  (v_share(r,i,rp,mrioA) - v_share.l(r,i,rp,mrioA))/(v_share.l(r,i,rp,mrioA)+1.E-3)))

                     +   sum((rC(r),iC(i),aa) $ xmm0(r,i,aa),
                             sqr(  (v_xmm(r,i,aa) - xmm0(r,i,aa))/(xmm0(r,i,aa)+1.E-3))) * 0.1;
```

The problem can be solved for each country and product independently. Accordingly, we solve in a loop to reduce overall computing time by yielding smaller NLP problem. Furthermore, we use the grid solve mechanism to solve all sectors for one region in parallel:

```
  loop( (r0),

    $$ifthene.version  %system.gamsrelease%>24.5
         m_hpdMM.solvelink = 6;
    $$else.version
          m_hpdMM.solvelink = 3;
    $$endif.version

*
*   --- solve individual products on grid
*
    loop(iCur $ sum(mrioA_aa(mrioA,aa), xmm0(r0,iCur,aa)),

        $$batinclude 'util\title.gms' "'Balance MRIO shares for '" r.te(r0) "' and '" i.te(iCur)

        option kill=rc,kill=ic;
        rc(r0) = YES; ic(iCur) = YES;

        v_share.up(r0,iCur,rp,mrioA) = 1;
        m_hpdMM.solprint = 2;
        solve m_hpdMM using NLP minimizing v_hpdmm;
        handelsC(iCur) = m_hpdMM.handle;
    );
```

After collection of the solutions from the grid, we check if infeasibilities occur. In that case, the bounds on the slacks are somewhat relaxed and the balancing problem is resolved, however now not via the grid solve facility:

```
    $$ife.version %system.gamsrelease%>24.5 retCode = readyCollect(handelsC,120);

    $$batinclude 'util\title.gms' "'Balance MRIO share for '" r.te(r0) "', Collect model solutions '"

    repeat

       loop(iCur $ handleCollect(handelsC(iCur)),
         display $ handledelete(handelsC(iCur)) 'trouble deleting handelsC';

*
*        --- if balancing yielded infeasibilities, re-solve with increased bounds for slacks
*
         if ( m_hpdMM.numinfes > 0,
               m_hpdMM.solvelink = 5;
               option kill=rc,kill=ic;
               rc(r0) = YES; ic(iCur) = YES;
               v_slackMM.lo(rc,ic,rp) = -1.E-6;
               v_slackMM.up(rc,ic,rp) = +1.E-6;
               m_hpdMM.solprint = 1;
               solve m_hpdMM using NLP minimizing v_hpdmm;
               if ( m_hpdMM.numinfes > 0,
                  v_slackMM.lo(rc,ic,rp) = -1.E-3;
                  v_slackMM.up(rc,ic,rp) = +1.E-3;
                  solve m_hpdMM using NLP minimizing v_hpdmm;
               );
               if ( m_hpdMM.numinfes > 0, abort "Imports shares by MRIO agents could not be balanced",rc,ic);
           );
         handelsC(iCur) = 0;
       );
     retCode = sleep(card(handelsC)*0.1);

     until ( (card(handelsC) eq 0) or (timeElapsed > 3600) );
```

The process requires about a minute for 57x68 global SAM on a performant desktop.

Derivation of the split factors

We first map the split factors from the OECD METRO model which are based on the GTAP9 data base to the sector and regions of the dis-aggregate GTAP8 data base if that is used. That implies that we use for a couple of cases uniform split factors for service sectors. Furthermore, some countries are missing, here ROW shares are applied. The mapping of these shares to the actual sector and regional aggregation used by the model uses import at world market prices (cif) plus import taxes as weights, .i.e. bi-lateral imports at domestic market prices. These weights are stored in a GDX container along with the split factors. The aggregation of the split factors works as follows:

```
v_share.l(r0,i,rp,mrioA)
   = sum( (rg(r,rsplit),rg1(rp,rsplit1),ii(iOri,i)) $ p_splitFactorsG(rsplit1,mrioA,iOri,rsplit), p_weights(rSplit1,iOri,rSplit));

v_share.l(r0,i,rp,mrioA) $ v_share.l(r0,i,rp,mrioA)
   = sum( (rg(r,rsplit),rg1(rp,rsplit1),ii(iOri,i)),   p_splitFactorsG(rsplit1,mrioA,iOri,rsplit)*p_weights(rSplit1,iOri,rSplit))
                                          /v_share.l(r0,i,rp,mrioA);
```

i.e. we first assign the weights and next calculated the weighted average.

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 by each MRIO agent is expressed via the usual share equations:

```
xwMrioEq(rp,iMrio(i),mrioA,r,ts(t)) $ (amwMrio(rp,i,mrioa,r,t) $ rrComb(rp,r)) ..
   xwMrio(rp,i,mrioA,r,t)/xw.scale(rp,i,r,t) =e= amwMrio(rp,i,mrioA,r,t)*xmtMrio(r,i,mrioA,t)/xw.scale(rp,i,r,t)

                            * [ {(pmtMrio(r,i,mrioa,t)/[(1 + imptx(rp,i,r,t) + mtax(r,i,t))
*
*   Note WB: cif price is defined via macro %pmcif%, not a variable in model
*
                              * %pmcif%(rp,i,r,t)/chipm(rp,i,r) ])**sigmaw(r,i)
                                *  lambdam(rp,i,r,t)**(sigmaw(r,i) - 1)} $ (not isSmallImpShare(rp,i,r))

                               + lambdam(rp,i,r,t) $ isSmallImpShare(rp,i,r)
                              ];
```

The equation substitutes out the cif price with a macro to reduce model size and allows for the GLOBE solution of small import shares being linked in Leontief fashion to total import demand by the 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 only relevant for intermediate input demand:

```
xmtMrioeq(rs(r),iMrio(i),mrIOA,ts(t)) $ (xmtFlag(r,i) $ (sum(mrio_a(mrioA,aa), alpham(r,i,aa,t)))) ..
   xmtMrio(r,i,mrioA,t)/xmt.scale(r,i,t) =e= sum(mrio_A(mrioA,aa) $ alpham(r,i,aa,t), m_xm(r,i,aa,t))/xmt.scale(r,i,t);
```

The average import price of each MRIO agent is defined via the dual price aggregator:

```
pmtMrioEq(r,iMrio(i),MrIoA,ts(t)) $ ( (pmtMrio.range(r,i,mrioA,t) ne 0) $ xmtFlag(r,i) $ (not pmtElas(r,i)) $ pmtMrio.l(r,i,mrioA,t) ) ..
*
   pmtMrio(r,i,MrioA,t) =e=
*
*          --- CES case with sigmaw <> 1
*
           [sum(rp $ (amwMrio(rp,i,mrioA,r,t) $ (not isSmallImpShare(rp,i,r))), amwMrio(rp,i,mrioA,r,t)
                       *([(1 + imptx(rp,i,r,t) + mtax(r,i,t)) * %pmcif%(rp,i,r,t)/chipm(rp,i,r) ]/lambdam(rp,i,r,t))**(1-sigmaw(r,i))
                  )**(1/(1-sigmaw(r,i))) * axPmtCDMrio(r,i,mrioA,t) ]
                     $ ((sigmaw(r,i) ne 1) $ sum(rp $ (xwFlag(rp,i,r) $ (not isSmallImpShare(rp,i,r))), 1))
*
*          --- CD case with sigmaw == 1
*
         + [prod(rp $ (amwMrio(rp,i,mrioA,r,t)  $ (not isSmallImpShare(rp,i,r))),
                       ([(1 + imptx(rp,i,r,t) + mtax(r,i,t)) * %pmcif%(rp,i,r,t)/chipm(rp,i,r) ]
                        /(lambdam(rp,i,r,t)*amw(rp,i,r,t)))**amwMrio(rp,i,mrioA,r,t)
                  ) * axPmtCDMrio(r,i,mrioA,t) ]
                      $ ((sigmaw(r,i) eq 1) $ sum(rp $ (xwFlag(rp,i,r) $ (not isSmallImpShare(rp,i,r))), 1))
*
         + sum(rp $ (amwMrio(rp,i,mrioA,r,t)  $ isSmallImpShare(rp,i,r)), amwMrio(rp,i,mrioA,r,t)
                  *([(1 + imptx(rp,i,r,t) + mtax(r,i,t)) * %pmcif%(rp,i,r,t)/chipm(rp,i,r) ]/lambdam(rp,i,r,t)));
```

The dual price aggregator considers three cases (1) standard case with a non-unity substitution elasticity, (2) CD-case with a substitution elasticity of unity and (3) the additional contribution of the small shares handled in Leontief fashion.

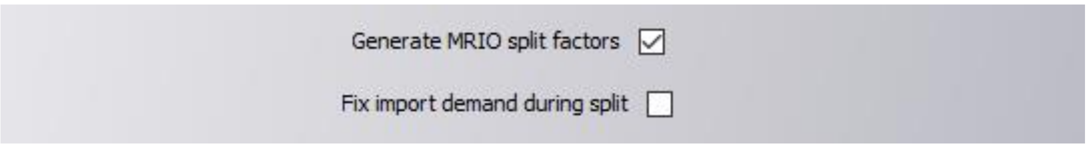
The link into the overall framework requires defining the aggregated bi-lateral import demands:

```
*
* --- MRIO extension: bi-lateral demand by agents
*
xwAggMrioEq(rp,iMrio(i),r,ts(t)) $ (xwFlag(rp,i,r) $ rrComb(rp,r)) ..
   xw(rp,i,r,t)/xw.scale(rp,i,r,t) =e= sum(mrioA $ amwMrio(rp,i,mrioa,r,t),xwMrio(rp,i,mrioA,r,t))/xw.scale(rp,i,r,t);
```

Graphical User Interface

Data base generation

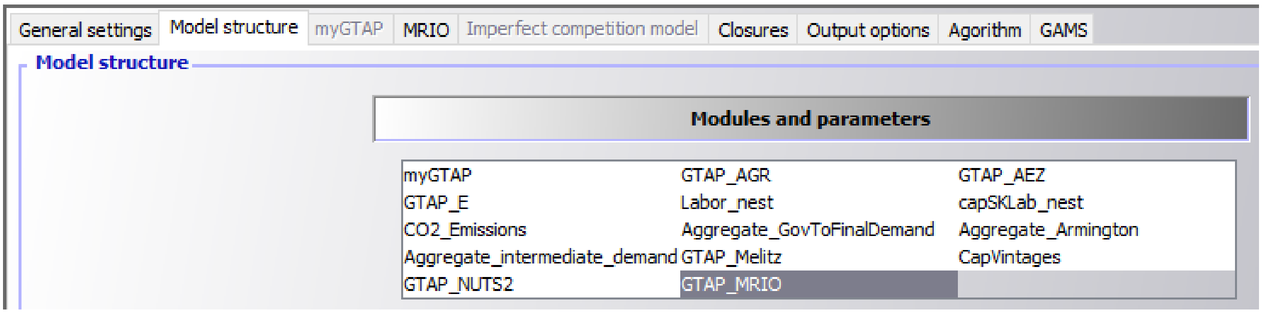
In order to generate the split factors when the data base is set-up, the following checkbox must be activated:



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:

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.

Tariff Line Module

Background

The GTAP data base offers bi-lateral protection and matching trade flow data based on MacMaps at the level of 57 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). Analysis especially of (potential) Free Trade Agreements requires often detail beyond the 57 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.

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 dis-aggregation 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.

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) ..

     xwtl(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) ];
```

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[tl_i(tl,i),p_gwtl(rNat,tl,rNat1,t)]) ..
     pe(rNat,i,rNat1,t) =e= sum[tl_i(tl,i), p_gwtl(rNat,tl,rNat1,t) * ptls(rNat,tl,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 the original layout, the export price pe is implicitly defined by the equation peeq 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[tl_i(tl,i),p_gwtl(rNat,tl,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
                    $ (omegaw(rNat,i) ne inf)

     + (pe(rNat,i,rNat1,t) - pet(rNat,i,t) $ (omegax(rNat,i) ne inf) - ps(rNat,i,t) $ (omegax(rNat,i) eq inf)) $ (omegaw(rNat,i) eq inf) ;
```

The bi-lateral export supply at commodity level xws is defined identically, however neglecta 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[tl_i(tl,i),p_gwtl(rNat,tl,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,tl,rNat1,ts(t)) $ p_amwtl(rNat,tl,rNat1,t) ..
   xwtl(rNat,tl,rNat1,t) =e= sum[tl_i(tl,i),
                             p_amwtl(rNat,tl,rNat1,t)*xw(rNat,i,rNat1,t)

                               * (pm(rNat,i,rNat1,t)/(m_pmCifTl(rNat,tl,rNat1,t)
                                    *(1 + imptxTl(rNat,tl,rNat1,t) + mtax(rNat1,i,t))/chipm(rNat,i,rNat1))    )**p_sigmawtl(rNat1,i)];
```

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_sigmawtl(rNat1,i))]
                                **(1/(1-p_sigmawtl(rNat1,i)));
```

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(rNat,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 implicite tariff is between its lower and upper bounds in the final solution.

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"
                              Can        EU_28          ROW
---------------------------------------------------------------------- 19 line(s) not displayed

   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 initiliaze further variablers
*
   xwtl.l(rNat,tl,rNat1,t0) = p_importTL(rNat1,tl,rNat);
*
*  --- bi-lteral 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).

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,tl,rNat1,t0) $ xwtl.l(rNat,tl,rNat1,t0)
   = sum(tl_i(tl,i), xwtl.l(rNat,tl,rNat1,t0)/xw.l(rnat,i,rNat1,t0)

    *  (m_pmCifTl(rNat,tl,rNat1,t0)
                             *(1 + imptxTl(rNat,tl,rNat1,t0) + mtax(rNat1,i,t0))
         /pm.l(rNat,i,rNat1,t0))**p_sigmawtl(rNat1,i));
```

Post model processing

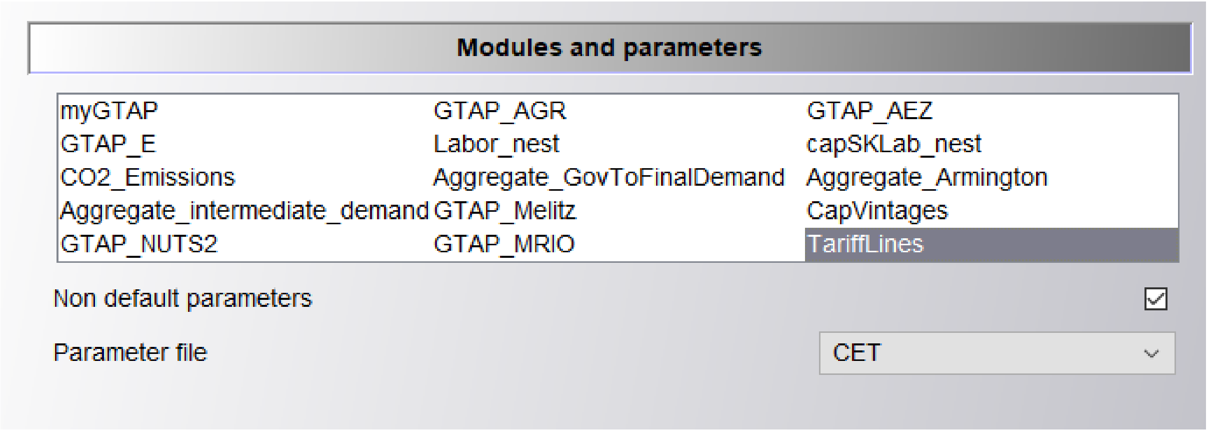
The code comprising the model equations also comprises a section for post-model processing:

```
p_results(rNat,"Q",tl,rNat1,"dom",%arg1%,"%version%") $ xwtl.l(rNat,tl,rNat1,%arg2%) = xwtl.l(rNat,tl,rNat1,%arg2%) * %scale%;
p_results(rNat,"P",tl,rNat1,"dom",%arg1%,"%version%") $ xwtl.l(rNat,tl,rNat1,%arg2%) = ptls.l(rNat,tl,rNat1,%arg2%);
p_results(rNat,"T",tl,rNat1,"dom",%arg1%,"%version%") $ xwtl.l(rNat,tl,rNat1,%arg2%) = sum(tl_i(tl,i),exptx.l(rNat,i,rNat1,%arg2%) + etax.l(rNat,i,%arg2%));

p_results(rNat,"TRQ",tl,rNat1,"imp",%arg1%,"%version%")    $ p_trq(rNat1,tl,rNat,%arg2%) = p_trq(rNat1,tl,rNat,%arg2%);
p_results(rNat,"LowTar",tl,rNat1,"imp",%arg1%,"%version%") $ p_trq(rNat1,tl,rNat,%arg2%) = imptxTl.lo(rNat1,tl,rNat,%arg2%) + eps;
p_results(rNat,"UppTar",tl,rNat1,"imp",%arg1%,"%version%") $ p_trq(rNat1,tl,rNat,%arg2%) = imptxTl.up(rNat1,tl,rNat,%arg2%);

p_results(rNat,"Q",tl,rNat1,"imp",%arg1%,"%version%") $ p_amwtl(rNat1,tl,rNat,%arg2%) = xwtl.l(rNat1,tl,rNat,%arg2%) * %scale%;
p_results(rNat,"P",tl,rNat1,"imp",%arg1%,"%version%") $ p_amwtl(rNat1,tl,rNat,%arg2%)
    = sum(tl_i(tl,i),m_pmCifTl(rNat1,tl,rNat,%arg2%)*(1 + imptxTl(rNat1,tl,rNat,%arg2%) + mtax(rNat,i,%arg2%)));

p_results(rNat,"T",tl,rNat1,"imp",%arg1%,"%version%") $ xwtl.l(rNat1,tl,rNat,%arg2%)
       = imptxTl(rNat1,tl,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%");
```

Integration in the GUI

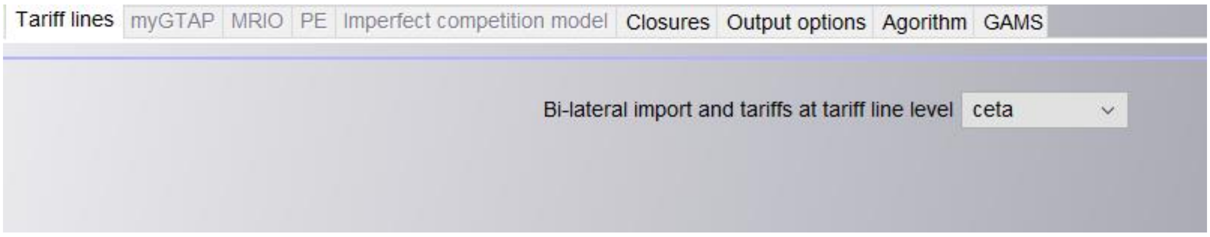
Model configuration

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.

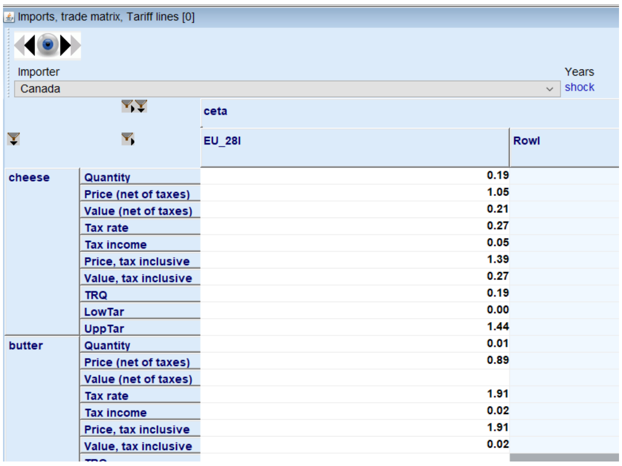
The file with the set of considered tariff lines and related data can be chosen on the "Tariff lines" tab:



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":

A screen shot of a part of such a table is shown below:



Imports, trade matrix, Tariff lines [0]

Importer: Canada — Years: shock

		ceta	
		EU_28I	RowI
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	

References

Himics, M., Britz, W. (2016): Flexible and welfare-consistent tariff aggregation over exporter regions, Economic Modelling 53: 375-387

CO2 and Non-CO2 Accounting

The CO2 accounting defined emissions in the model based on the follow three equations plus allows to use a MCP-mechanism to render the emission tax endogenous under an emission ceiling.

```
*
* --- emissions linked to consumption of domestic good
*
 emidEq(rs(r),iIn(i),aa,ts(t)) $ (m_alphad(r,i,aa,t) $ (xda0(r,i,aa)*emid0(r,i,aa))) ..
   emid(r,i,aa,t)/emid.scale(r,i,aa,t) =E= m_xd(r,i,aa,t)/xda0(r,i,aa)*emid0(r,i,aa)/emid.scale(r,i,aa,t);
*
* --- emissions linked to consumption of imports
*
 emiiEq(rs(r),iIn(i),aa,ts(t)) $ (m_alpham(r,i,aa,t) $ (xma0(r,i,aa)*emii0(r,i,aa))) ..
   emii(r,i,aa,t)/emii.scale(r,i,aa,t) =E= m_xm(r,i,aa,t)/xma0(r,i,aa)*emii0(r,i,aa)/emii.scale(r,i,aa,t);
*
* --- total emissions in each regions
*
 emisEq(rs(r),ts(t)) ..
   emis(r,t)/emis.scale(r,t) =E= [  sum( (i,aa) $ (m_alphad(r,i,aa,t) $ (xda0(r,i,aa)*emid0(r,i,aa))), emid(r,i,aa,t))
                                  + sum( (i,aa) $ (m_alpham(r,i,aa,t) $ (xma0(r,i,aa)*emii0(r,i,aa))), emii(r,i,aa,t))]
                                 /emis.scale(r,t);
*
* --- emission ceiling equations, paired to emission price (=tax)
*
 emisPEq(rs(r),ts(t)) $ permit(r,t) ..
    permit(r,t)/emis.scale(r,t) =E= emis(r,t)/emis.scale(r,t);
```

Note that the code does not use emissions factors as parameters, but the total benchmark emissions related to domestic and import demand by the Armington agents *emid0* and *emii0*. Emission are calculated by updating these totals based on the change in import demand *m_xm* and domestic demand *m_xd* relative to benchmark demands stored as xma0 and xda0.

Non-CO2 accounting is currently only done for reporting purposes. It uses emissions factors and not total.

GTAP-E

GTAP-E is entirely based on the flexible nesting approach and is detailed there, see section Flexible nesting.

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;
   omegafNest(r,"agr",ffNest)          = omegaf(r,ffNest);

   fNest("nonAgr")                     = YES;
   fNest_a_f("nonAgr",nonAgr,ffNest)   = YES;
   fNest_n_f("xft","nonAgr",ffNest)    = YES;
   omegafNest(r,"nonAgr",ffNest)       = omegaf(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.

GTAP-Melitz: Heterogenous firm module

This section is prepared by Yaghoob Jafari and Wolfgang Britz

Introduction

Since Armington (1969) 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 (Hummel and Klenow, 2005; Chenny, 2008, among others).

The pioneer paper by Melitz (2003) introduced firm productivity heterogeneity drawing from Hopenhayn (1992) into the monopolistic competition framework by Krugman (1980). 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; Eaton et al., 2004) supported its empirical evidence by 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; Baliatreri et al., 2011; Oyamada, 2013; Akgul et al., 2016; and Dixon et al., 2016).

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. Baletreri 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, Viloria and Hertel (2016) introduced the Melitz framework into the standard GTAP model, abstracting from exogenous firm exit.

Besides, Balistreri and Rutherford (2013) 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[9].

We discuss in here the introduction of the Melitz model in CGEBOX. Besides realized in GAMS (GAMS Development Corporation, 2013) and not in GEMPACK (Harrison and Pearson, 1996), 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.

Brief Literature overview

With regard to international trade, traditional applied general equilibrium models fail in two important aspects (see critique by Keohe 2005): 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) and Ethier (1982) 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) who combine the traditional and new trade theory within an integrated 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

[9] See also Itakura and Oyamada (2013) for technical aspects and Roson and Oyamada (2014) for a review of this emerging field.

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) 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, Aw et al. 2001) 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; Feenstra, 1994; Broda and Weinstein, 2006; Balistreri et al, 2010; Keohe and Ruhl, 2013; among others). That strand of literature is complemented by more theoretical works which discuss impacts of trade liberalization on the extensive margin (Hummel and Klenow, 2005; Chenny, 2008, among others). Traditional applied general-equilibrium models based on the Armington assumption cannot depict changes at the extensive margin of trade[10] 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.

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;

[10] 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).

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.

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[11] 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.

[11] 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 et al. (1982) tradition and departs significantly from the nomenclature used by Rutherford and that used by in the GEMPACK version of the model.

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 totals 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 ($\widetilde{\varphi}_{rs}$) 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 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) 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 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 ($Q_{ais} \equiv U_{ias}$) can be represented as

$$Q_{ais} = \left(\sum_r \int_{\omega \epsilon \Omega_{irs}} \lambda_{airs}^{\frac{1}{\sigma_{is}}} Q_{airs}(\omega)^{\frac{\sigma_{is}-1}{\sigma_{is}}} d\omega \right)^{\frac{\sigma_{is}}{\sigma_{is}-1}} \quad \textbf{(1)}$$

where Ω_{irs} represents the set of products i sourced from region r to s and $\omega \epsilon \Omega_{irs}$ index the varieties in the set Ω_{isr}. In this context, $Q_{airs}(\omega)$ represents the demand quantity of commodity i for variety ω in region s by agent a which is sourced from region r, σ_i represents the constant elasticity of substitution for each commodity, and λ_{airs} are preference weights (share parameters)[12] 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

$$P_{ais} = \left(\sum_s \int_{\omega \epsilon \Omega_{irs}} \lambda_{airs} \, PA_{airs}(\omega)^{1-\sigma_{is}} d\omega \right)^{\frac{1}{1-\sigma_{is}}} \quad \textbf{(2)}$$

[12] 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.

where $PA_{airs}(\omega)$ is agent's a (purchase) price of product i for variety ω 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 (**2**)

$$P_{ais} = \left(\sum_r \lambda_{airs} \; N_{irs} \; \widetilde{PA}_{airs}{}^{1-\sigma_{is}} \right)^{\frac{1}{1-\sigma_{is}}} \qquad \textbf{(3)}$$

where $\widetilde{PA}_{irs}$ 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

The total (Q_{airs}) and average per firm ($\widetilde{Q}_{airs}$) demand for the average variety by an agent to be shipped from r to s ($\widetilde{Q}_{airs}$) can be obtained by applying *Shephard's Lema* on the expenditure function:

$$Q_{airs} = \widetilde{Q}_{airs} \, N_{irs} \; = \lambda_{airs} \; N_{irs} \, Q_{ais} \left(\frac{P_{ais}}{\widetilde{PA}_{airs}} \right)^{\sigma_{is}} \qquad \textbf{(4)}$$

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 (**4**) 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^{ie}, fixed cost of operating on a trade link f_{irs} and marginal cost c_{ir}. Let φ_{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, τ_{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 $\widetilde{PF}_{irs}$ (inclusive of domestic transport margin) is

$$\widetilde{PF}_{irs} = \frac{\sigma_{is}}{\sigma_{is} - 1} \frac{\tau_{irs}\, c_{ir}}{\tilde{\varphi}_{irs}} \quad \textbf{(5)}$$

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 (Chaney, 2008) and was shown as empirically relevant (Axtell, 2001; Luttmer, 2002; Gabaix, 2008; Eaton et al., 2001).

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 φ draws from a Pareto distribution with Probability Density Function (PDF)

$$g(\varphi) = a\left(\frac{b^a}{\varphi^{a+1}}\right) = \frac{a}{\varphi}(\frac{b}{\varphi})^a \quad \textbf{(6)}$$

and Cumulative Distribution Function (CDF)

$$G(\varphi) = 1 - (\frac{b}{\varphi})^{a} \quad \textbf{(7)}$$

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.

On each bilateral trade link, the given the fixed bilateral trade cost, variable costs and demand define jointly a certain cut off productivity level (φ^*_{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

$$c_{ir} f_{irs} = \frac{r(\varphi^*_{irs})}{\sigma_i} \quad \textbf{(8)}$$

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 φ^*_{irs} can be obtained by solving (**8**). **888888**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

$$\widetilde{\varphi}_{irs} = \left[\frac{1}{1 - G(\varphi^*_{irs})} \int_{\varphi^*_{irs}}^{\infty} \varphi_{irs}^{\sigma_{is}-1} g(\varphi)\, d\varphi\right]^{\frac{1}{1-\sigma_{is}}} \quad \textbf{(9)}$$

If these productivities are Pareto distributed, we can write[13]

$$\widetilde{\varphi}_{irs} = \left[\frac{a}{(\mathrm{a} + 1 - \sigma_{is})}\right]^{\frac{1}{1-\sigma_{is}}} * \varphi^*_{irs} \quad \textbf{(10)}$$

[13] 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).

Eq. (10) provides the relationship between the productivities of the average and marginal firm (for further details see Allen and Arkolakis (2016).

Using optimal firm pricing according to (**5**) 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 $r_{irs}(\widetilde{\varphi})$ is defined as

$$\frac{r_{irs}(\varphi^*)}{r_{irs}(\widetilde{\varphi})} = (\frac{\varphi^*}{\widetilde{\varphi}})^{\sigma_i} \quad \textbf{(11)}$$

Solving (**10**) for $\frac{\varphi^*}{\widetilde{\varphi}}$, substituting it into (**11**), and then solving the resulting equation for $r_{irs}(\varphi^*)$ and replacing its value in (**8**), defines a relation between the bilateral fixed cost at current composite input price (the left hand side of (12) below), the average firms revenue ($\widetilde{PF}_{irs}\,\widetilde{Q}_{irs}$), the shape parameter of the Pareto distribution of the productivities and the substitution elasticity of demand:

$$c_{ir}f_{irs} = \frac{(a+1-\sigma_{is})}{a\sigma_{is}}\,\widetilde{PF}_{irs}\,\widetilde{Q}_{irs} \quad \textbf{(12)}$$

Note that average firm's sale in region r in each industry i to region s ($\widetilde{Q}_{irs}$) at the equilibrium is composed of the demand for use of i by different agents[14].

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 (**7**) and inverting it we have

$$\varphi^*_{irs} = \frac{b}{(\frac{N_{irs}}{M_{ir}})^{\frac{1}{a}}} \quad \textbf{(13)}$$

substituting (**13**) into (**10**) results in

$$\widetilde{\varphi}_{irs} = b\left[\frac{a}{(\mathrm{a}+1-\sigma_{is})}\right]^{\frac{1}{1-\sigma_{is}}} * (\frac{M_{ir}}{N_{irs}})^{-\frac{1}{a}} \quad \textbf{(14)}$$

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 δ in each future period to suffer a shock which forces its exit. Therefore, δM_{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 δM_{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

[14] $\tilde{Q}_{irs} = \sum_a \tilde{Q}_{airs}$

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.

$$\tilde{\pi}_{irs} = \frac{\widetilde{PF}_{irs}\,\tilde{Q}_{irs}}{\sigma_{is}} - c_{ir} f_{irs} \quad \textbf{(15)}$$

The probability that a firm will operate on the r-s trade link is given by the ratio of $\frac{M_{ir}}{N_{ir}}$[15]. 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

$$c_{ir}\,\delta f^{ie} M_r = \sum_s N_{irs}\,\widetilde{PF}_{irs}\,\tilde{Q}_{irs}\,\frac{\sigma_{is}-1}{a\sigma_{is}} \quad \textbf{(16)}$$

where zero profit condition (**12**) is used to replace the fixed operating cost $c_{ir} f_{irs}$.

With the number of entered firm established in (16), 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 ($\sum_s N_{irs}\,\frac{\tau_{hrs}\,\tilde{Q}_{irs}}{\tilde{\varphi}_{irs}}$). Therefore, composite input demand is defined as

$$Y_{ir} = \delta M_{ir}\, f^{ie} + \sum_s N_{irs}\,(f_{irs} + \frac{\tau_{irs}\,\tilde{Q}_{irs}}{\tilde{\varphi}_{irs}}) \quad \textbf{(17)}$$

This equation provides the link to the equations in the GTAP model describing the technology and related costs. **Table 4** summarizes full set of Melitz equations which are introduced into the GTAP model.

Table 4: Equilibrium conditions in the Heterogenous Firm model

Equation	Equilibrium condition	Associated variable
(**3**)	Sectoral Aggregation	P_{is}: Sectoral price index
(**4**)	Firm-level demand	$\widetilde{PF}_{irs}$Average firm price
(**5**)	Firm-level Pricing	$\tilde{Q}_{irs}$: Average firm quantity
(**12**)	Zero cut off condition	M_{ir}Number of operating firms
(**14**)	CES Weighted Average productivity	N_{ir}:Average firm productivity

[15] The probability that the firm will operate is 1- $G(\varphi^*_{irs}) = \frac{N_{ir}}{M_{ir}}$

(**16**)	Free entry condition	N_{irs}:Number of entered firm
(**17**)	Factor market clearing condition	c_{ir}: Sectoral composite input price
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[16] 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 (see Britz and Van der Mensbrugghe 2016).

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:

$$c_{ir}^{0} Y_{ir}^{0} \equiv vom_{ir} \tag{18}$$

$$P_{is}^0 Q_{ais}^0 \equiv xafm_{ais} \tag{19}$$

where vom_{ir}, represent the production value of commodity i in region r at producer tax inclusive prices and $xafm_{airs}$ represent the value of each agent's a 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.

[16] We currently use the GTAP8 database (Narayanan and McDougall 2012) which carries a snapshot of the 2007 world economy, covering 129 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.

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

$$Q_{ais}^{0} = \sum_{r} Q_{airs}^{0} = \sum_{r} \widetilde{Q}_{airs}^{0} N_{irs}^{0} \tag{20}$$

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 $\widetilde{Q}_{airs}$ for product i in region s from origins r by agent a can be depicted as

$$P_{ais}^{0} = \left(\sum_{r} \lambda_{airs} \frac{N_{irs}}{N_{irs}^{0}} \widetilde{PA}_{airs}^{0\;\;1-\sigma_{is}} \right)^{\frac{1}{1-\sigma_{is}}} \tag{21}$$

$$Q_{airs}^{0} = \lambda_{airs} \frac{N_{irs}}{N_{irs}^{0}} Q_{ais}^{0} \left(\frac{P_{ais}^{0}}{\widetilde{PA}_{airs}^{0}} \right)^{\sigma_{is}} \tag{22}$$

where N_{irs} depicts the number of operating firms on a trade link. Note that, as N_{irs} and λ_{airs} (the share parameter) are identical in the calibration point, we scale without any impact on the simulation behavior the lambdas λ_{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 21 and 22 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 $\widetilde{\mathrm{PA}}^0_{\mathrm{airs}} = 1$, $\mathrm{P}^0_{\mathrm{ais}} = 1$, since we now have the observed value of $\mathrm{Q}^0_{\mathrm{airs}}$, $\mathrm{Q}^0_{\mathrm{ais}}$, and estimated value σ_{is} [17], the value of λ_{airs} is calibrated in a way that we do not need the information on the benchmark number of firms. Accordingly, the values of λ_{airs} are recovered by inverting the demand functions.

$$\lambda_{airs} = \frac{Q^0_{airs}}{Q^0_{ais}} \left(\frac{\widetilde{PA}^0_{airs}}{P^0_{ais}} \right)^{\sigma_{is}} \tag{23}$$

Let $\widetilde{\mathrm{PF}}_{\mathrm{irs}}$ denote the average firm's offer price on a specific trade link, from which $\widetilde{\mathrm{PF}}_{\mathrm{airs}}$ 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 $\mathrm{c}^0_{\mathrm{ir}}$ and $\widetilde{\mathrm{PF}}^0_{\mathrm{irs}}$ are set to unity at the benchmark, $\widetilde{\mathrm{Q}}_{\mathrm{irs}}$ is estimated from (23), and the value of shape parameter a is taken from literature[18]:

$$f_{irs} = \frac{\widetilde{PF}^0_{irs}\,\widetilde{Q}^0_{irs}}{c^0_{ir}} \frac{(a + 1 - \sigma_i)}{a\sigma_i} \tag{24}$$

Please note that $Q^0_{irs} = \widetilde{Q}^0_{irs}\, N^0_{irs}$ 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.

[17] 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 σ_m; and the second level Armington demand (aggregated import) is a CEs composite of import demand by each region of origin with the substitution elasticity of σ_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 σ_i. To be consistent with Dixit –Stiglitz framework, σ_i is derived based on the weighted average of the substitution elasticities of first and second level Armington nests where the weights given to σ_m and σ_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.

[18] 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 (2016) calibrated shape parameter of 2.89 for manufacturing sector consistent with a shape parameter obtained Spearot (2016). 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).

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:

$$c_{ir}\ \delta f^{ie} = \sum_s \frac{N_{irs}^{0}}{M_r^0} \widetilde{PF}_{irs}^{0}\ \widetilde{Q}_{irs}^{0}\ \frac{\sigma_{is}-1}{a\sigma_{is}} = \sum_s \frac{1}{M_r^0} \widetilde{PF}_{irs}^{0}\ Q_{irs}^{0}\ \frac{\sigma_{is}-1}{a\sigma_{is}} \quad \textbf{(25)}$$

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:

$$\widetilde{\varphi}_{irs} = b\left[\frac{a}{(\mathrm{a}+1-\sigma_{is})}\right]^{\frac{1}{1-\sigma_{is}}} * \left(\frac{M_r^0}{N_{irs}^{0}}\right)^{-\frac{1}{a}} \quad (26)$$

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 (5) as follows:

$$\tau_{irs} = \frac{\sigma_{is}-1}{\sigma_{is}} * \widetilde{\varphi}_{irs}^{0} \quad \textbf{(27)}$$

Here, it is less obvious why M_r^0 and N_{irs}^{0} can be freely chosen without affecting simulation behavior. $\widetilde{\varphi}_{irs}$ and τ_{irs} enter the total industry cost and the price markup equations. Total industry costs on a trade link r-s are given as: $N_{irs}\left(f_{irs} + \frac{\tau_{hrs}\ \widetilde{Q}_{irs}}{\widetilde{\varphi}_{hrs}}\right)$ which shows that changes in $\widetilde{\varphi}_{irs}^{0}$ and τ_{irs} resulting from different benchmark values of N_{ir} and M_r do not matter. The same holds for the price markup equation: $\widetilde{PF}_{irs} = \frac{\sigma_{is}}{\sigma_{is}-1}\frac{\tau_{irs}\ c_{ir}}{\widetilde{\varphi}_{irs}}$.

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 to Zeynep et al. 2016 in their 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.

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.

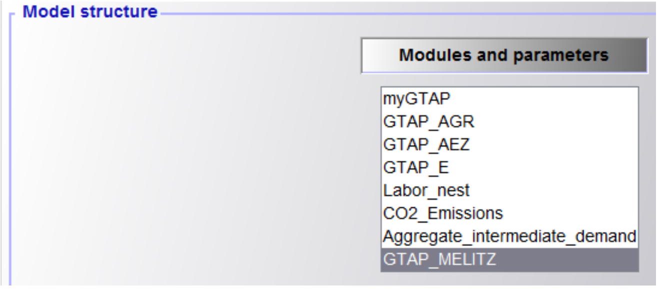
Potential improvements in that regard could encompass a split up of the diagonal domestic intermediate demand into a share subject to the love of variety assumption and a remainder treated as competitive. But before introducing further complexity into the framework, more testing is required.

Technical implementation and an example application

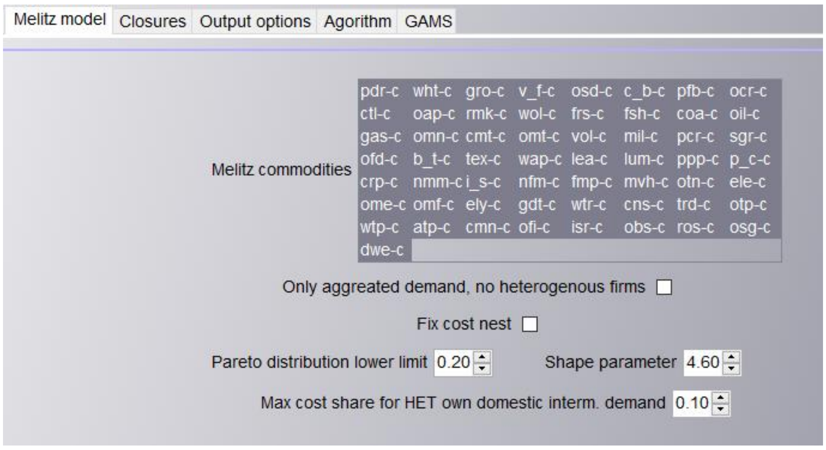
Technical implementation

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.



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 dis-entangling 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 GTAP8 data base as an example application[19], 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.

Table 5 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 +/- ½ 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 (σ) 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 σ 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 5: Money metric in comparison of Melitz and Standard GTAP model

[19] Appendix 2 provides the sectoral and regional aggregation of GTAP sectors into the new mapping.

Money metric decomposition [0]

Sectors and institutions: Total [Bio $] | Total: Total | Years: shock

	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 6 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 6: Aggregate price index of the Armington agents

Income, compare across regions [0]

Total or household groups: Total | Sectors and institutions: Price index [Index] | Years: shock | Percentage diff. to Scenarios: no Shock

	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 7: Factor income, corrected with price index aggregate Armington agent

Income, compare across regions [0]

Total or household groups: Total | Sectors and institutions: Factor income, normalized with price index [Bio $] | Years: shock | Percentage diff. to Scenarios: 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 8 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 8: Example summary information from Melitz model

Melitz, overview [0]

Exporter: Australia, New Zealand | Products: Heavy Manufacturing | Years: shock | Percentage diff. to Scenarios: 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 9 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 9: Example information from Melitz model by trade link

Melitz, trade matrix [0]

Exporter: Australia, New Zealand | Products: Heavy Manufacturing | Years: shock | Percentage diff. to 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 -0.15%	1.00 -0.09%	1.01 1.38%	1.01 0.51%	1.03 3.05%	0.99 -0.67%	1.01 1.30%	1.00 -0.37%	0.99 -0.58%	0.98 -2.35%	1.00 -0.45%
Number of operating firms	0.97 -3.22%	0.97 -2.96%	1.04 3.78%	1.00 -0.24%	1.12 11.92%	0.95 -5.49%	1.03 3.41%	0.96 -4.17%	0.95 -5.12%	0.87 -12.66%	0.95 -4.55%
Avg. output per firm	102.25 0.25%	7.52 0.19%	18.54 -1.26%	6.78 -0.41%	4.54 -2.87%	6.60 0.76%	0.60 -1.19%	9.33 0.46%	0.15 0.68%	1.53 2.51%	4.20 0.55%
Avg. productivity per firm	0.34 0.25%	0.34 0.19%	0.29 -1.26%	0.34 -0.41%	0.29 -2.87%	0.30 0.76%	0.30 -1.19%	0.35 0.46%	0.35 0.68%	0.35 2.51%	0.35 0.55%
Total output sold	98.97 -2.98%	7.30 -2.77%	19.24 2.47%	6.76 -0.65%	5.08 8.71%	6.24 -4.77%	0.62 2.18%	8.94 -3.73%	0.14 -4.48%	1.34 -10.47%	4.01 -4.03%

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

$$PA_{airs} = PF_{irs}\,(1 + \tau^{cd}_{air} + \varsigma^{cd}_{air}) \qquad \textbf{(28)}$$

where τ^{c}_{air} denote the consumption (sale) tax on each specific commodity for each agent while ς^{c}_{air} 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 (τ^{e}_{irs}) is applied to the firm offer price and determines the free on board (fob) price. An additional tax ς^{e}_{irs} is also introduced into (**29**) representing the uniform export tax across destinations.

$$PE^{FOB}_{irs} = PM_{irs}\,(1 + \tau^{e}_{irs} + \varsigma^{e}_{irs}) \qquad \textbf{(29)}$$

The FOB price P^{FOB}_{irs} is augmented by the international transport margin t^{tmg}_{irs} (observed form GTAP and endogenous to the model) to establish the cif (cost –insurance and fright) price:

$$PM^{CIF}_{irs} = P^{FOB}_{irs} + t^{tmg}_{irs} \qquad \textbf{(30)}$$

The bilateral import tax (τ^{m}_{irs}) converts the cif price into the bilateral import price, and ς^{m}_{irs} reflect a uniform tax shift across source countries

$$PM_{irs} = P_{irs}^{CIF}\ (1 + \tau_{irs}^{m} + \varsigma_{irs}^{m}) \qquad \textbf{(31)}$$

and finally the resulting bilateral import prices are converted to the agent prices by adding a consumption tax on imported commodities:

$$PA_{airs} = P_{irs}^{CIF}\ (1 + \tau_{air}^{cim} + \varsigma_{air}^{cim}) \qquad \textbf{(32)}$$

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).

Acknowledgments

Authors would like to thank Thomas W. Hertel, Edward J. Balistreri, and Dominique van der Mensbrugghe for their helpful comments.

References

Akgul, Z.,N.B.Villoria, andT.W.Hertel. 2016.“ GTAP-HET: Introducing firm heterogeneity into the GTAP model.” Journal of Global Economic Analysis, 1(1):111-180.

Akgul, Z.,N.B.Villoria,andT.W.Hertel.2016. .“ Theoretically-Consistent Parameterization of a Multi-sector Global Model with Heterogeneous Firms” Forthcoming.

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.

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.

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.

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.

Balistreri, E.J., and T.F. Rutherford. 2013. “Computing general equilibrium theoriesofmonopolisticcompetitionandheterogeneousfirms.”InHandbookofComputable General Equilibrium Modeling, edited by P. B. Dixon and D. W. Jorgenson.

Balistreri,E.J.,R.H.Hillberry,andT.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.

Balistreri, E.J., Hillberry, R.H., Rutherford, T.F., 2010. Trade and welfare: does industrial organization matter? Economics Letters 109, 85–87.

Bartelsman, E.J., Doms, M., 2000. Understanding productivity: lessons from longitudinal microdata. Journal of Economic Literature 38, 569–594.

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

Britz W. and van der Mensbrugghe, D. (2015). "Interactive interface for an extended version of the standard GTAP model in GAMS," GTAP Technical Paper, forthcoming

Broda, C., Weinstein, D.E., 2006. Globalization and the gains from variety. Quarterly Journal of Economics 121, 541–585

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.

Dervis KJ, de Melo J, Robinson S (1982) General equilibrium models for development policy. Cambridge University Press, New York

Dixit, A., and J. Stiglitz. 1977. “Monopolistic competition and optimum product diversity.” American Economic Review, 67(3): 297–308. doi:10.1017/cbo9780511492273.004.

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.

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.

Eaton,J.,S.Kortum,andF.Kramarz.2011.“An anatomy of international trade: evidence from French firms.”Econometrica,79(5): 1453–1498. doi:10.3386/w14610. Eaton, J., S.

Ethier, W.J. (1982), “National and International returns to Scale in the Modern Theory of International Trade”, American Economic Review 72, 225-238.

Feenstra, R.C., 1994. New product varieties and the measurement of international prices. The American Economic Review 84, 157–177.

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.

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.

Harrison W.J. and K.R. Pearson, "Computing Solutions for Large General Equilibrium Models Using GEMPACK", *Computational Economics*, Vol. 9 (1996), pp.83-127.

Helpman, E. and P. Krugman (1985), Market Structure and Foreign Trade: Increasing Returns, Imperfect Competition, and the International Economy, Cambridge, MA: MIT Press.

Helpman, E. (1981), “International Trade in the Presence of Product Differentiation, Economies of Scale, and Monopolistic Competition”, Journal of International Economics 11, 305-340.

Hopenhayn, H. 1992. “Entry, exit, and firm dynamics in long run equilibrium.” Econometrica, 60(5): 1127–1150. doi:10.2307/2951541

Horridge, M. (2006). GTAPAgg data aggregation program. Global Trade, Assistance, and Production: The GTAP 6 Data Base.

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.

GAMS Development Corporation. General Algebraic Modeling System (GAMS) Release 24.2.1. Washington, DC, USA, 2013.

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.

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).

Krugman, P. (1980), "Scale Economies, Product Differentiation, and the Pattern of Trade", American Economic Review 70, 950-959.

Luttmer, Erzo G. J. 2007. "Selection, Growth, and the Size Distribution of Firms." Quarterly Journal of Economics, 122(3): 1103–44.

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.

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.

Narayanan, G., Badri, Angel Aguiar and Robert McDougall, Eds. 2012. *Global Trade, Assistance, and Production: The GTAP 8 Data Base*, Center for Global Trade Analysis, Purdue University.

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.

Romer, P.M., 1994. New goods, old theory, and the welfare costs of trade restrictions. Journal of Development Economics 43, 5–38.

Rosen,R., and Kazuhiko Oyamada.2014. "Introducing 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*

S.J. Redding, and P.K. Schott. 2007. "Firms in international trade." Journal of Economic Perspectives, 21(3): 105–130. doi:10.3386/w13054.

Thomas W. Hertel, editor. Global Trade Analysis: Modeling and Applications. Cambridge University Press, New York, NY, 1997.

Van der Mensbrugghe D. and Britz W. (2015). The standard GTAP model in GAMS Version 6.2, GTAP Technical Paper, forthcoming

Spearot, A. 2016. "Unpacking the long run effects of tariff shocks: New structural implicationsfromfirmheterogeneitymodels."AEJMicroeconomics,8(2):128–67. doi:10.1257/mic.20140015

Wolfgang Britz, Arne Drud and Dominique van der Mensbrugghe (2015), Reducing unwanted consequences of aggregation in large-scale economic models- a systematic empirical evaluation with the GTAP model, Discussion Paper 2015:4, Institute for food and resource economics, university of Bonn.

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.

Appendix 2:

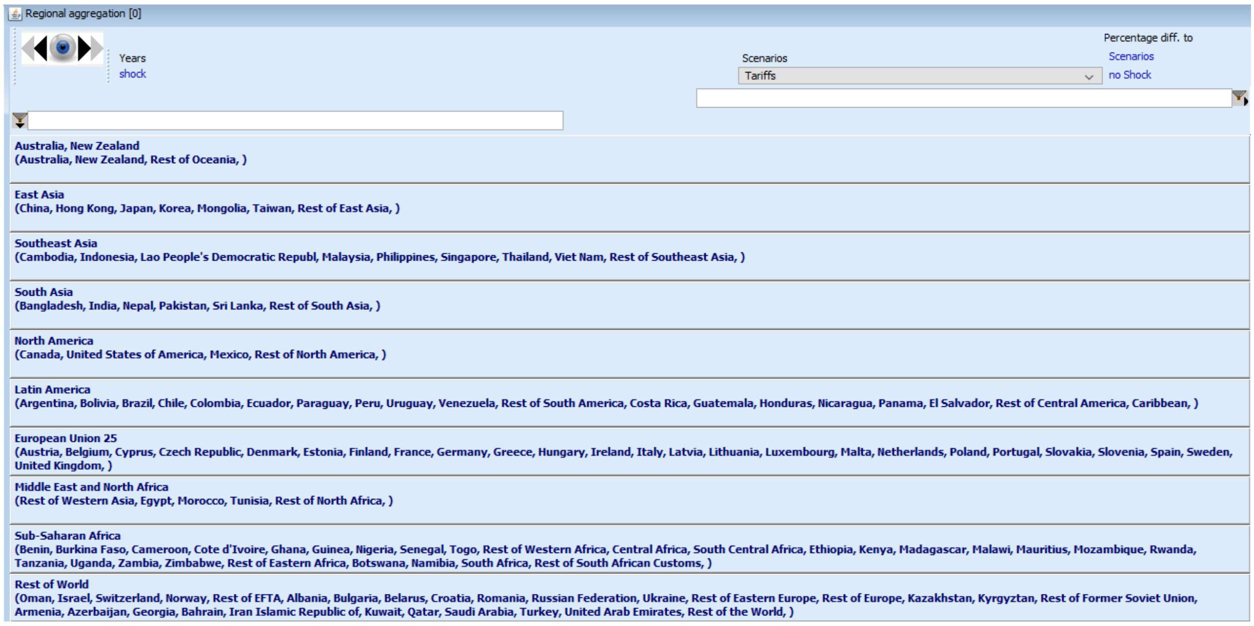
Table A.1: Sectoral aggregations

Sector aggregation [0]

Years: shock | Scenarios: Tariffs | Percentage diff. to: Scenarios: no Shock

Grains and Crops (Paddy rice, Wheat, Cereal grains, n.e.s., Vegetables and fruits, Oil seeds, Sugar cane and sugar beet, Plant-based fibers, Crops, n.e.s., Processed rice,)
Livestock and Meat Products (Bovine cattle, sheep and goats, horses, Animal products n.e.s., Raw milk, Wool, silk-worm cocoons, Bovine cattle, sheep and goat, horse meat products, Meat products n.e.s.,)
Mining and Extraction (Forestry, Fishing, Coal, Oil, Gas, Minerals n.e.s.,)
Processed Food (Vegetable oils and fats, Dairy products, Sugar, Food products n.e.s., 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 n.e.s., Manufactures n.e.s.,)
Heavy Manufacturing (Petroleum, coal products, Chemical, rubber, plastic products, Mineral products n.e.s., Ferrous metals, Metals n.e.s., Electronic equipment, Machinery and equipment n.e.s.,)
Utilities and Construction (Electricity, Gas manufacture, distribution, Water, Construction,)
Transport and Communication (Trade, Transport n.e.s., Sea transport, Air transport, Communication,)
Other Services (Financial services n.e.s., Insurance, Business services n.e.s., Recreation and other services, Public administration and defense, education, health services, Dwellings,)

Table A.2: Regional aggregations



GTAP-AEZ

The module uses the following logic. Each nation or sub-region is dis-aggregated to different agro-ecological zones (set *aez*). If a certain region comprises a certain aez is 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 land categories:

```
set landCat "Land categories" / forestry,Grazing,Agriculture /;
```

The different activities in the model are mapped to these to land categories:

```
set landCat_a0(landCat,i0) "Map between land categories and activities"/
                forestry.frs,
                Grazing.(ctl,oap,rmk,wol)
                Agriculture.(set.icrops)
/;
```

The code does not require that the SAM comprises the full detail of activities, but checks the mapping:

```
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;
```

Model equations

The model equations are defined in “gams\GtapAEZ\gatpAez_model.gms”.

The land demand *xftAez* for each activity in each Aez *aez* is defined in the equation pftAezEq:

```
  pftAezEq(rs(r),aez,aIn(a),ts(t)) $ (aezFlag(r,aez) $ andAez(r,aez,a,t) ) ..
*
   xftAez(r,aez,a,t)/xftAez.scale(r,aez,a,t) =e= andAez(r,aez,a,t) * xf(r,"land",a,t)/xftAez.scale(r,aez,a,t)
          * (m_pfa(r,"land",a,t)
                / (pftAez(r,aez,a,t)
                    *(1+sum(r_r(r,rnat),fctts(rNat,"land",a,t) + fcttx(rNat,"land",a,t) + fcttxShift(rNat,t))))))**sigmaAez(r,a);
```

It uses hence a CES approach, where the relation between the average returns for land in the activity defined by the macro m_pfa and the returns in the AEZ drives the allocation of the activity in space. The returns to land use of an activity in the AEZ reflect the specific price net of taxes *pftAez* plus the nation wide taxes for land use in that activity.

The average return *m_pfa* uses the usual dual price aggregator defined in the *pfaAezEQ*:

```
  pfaAezEq(rs(r),"land",aIn(a),ts(t)) $ sum(aezFlag(r,aez), andAez(r,aez,a,t)) ..


  m_pfa(r,"land",a,t)
    =e= sum(aezFlag(r,aez) $ andAez(r,aez,a,t),
        andAez(r,aez,a,t)
         *(pftAez(r,aez,a,t)
            *(1+sum(r_r(r,rNat),fctts(rNat,"land",a,t) + fcttx(rNat,"land",a,t) + fcttxShift(rNat,t))))**(1-sigmaAez(r,a)))**(1/(1-sigmaAez(r,a)));
```

The average return to land in the AEZ *pftTop* is a CET aggregation over the prices *pftLandCat* for the different land categories *landCat* based on the transformation elasticity *omegaAezTop*:

```
  pftTopEq(rs(r),aez,ts(t)) $ (aezFlag(r,aez) and sum(landCat,  gfTop(r,aez,landCat,t))) ..
*
   pftTop(r,aez,t) =e=
         sum(landCat $ gfTop(r,aez,landCat,t),
             gfTop(r,aez,landCat,t) * pftLandCat(r,aez,landCat,t)**(1+omegaAezTop(aez)))**(1/(1+omegaAezTop(aez)));
```

The distribution of the total land in the AEZ xftTop to these land categories uses the standard CET equation based on the share parameter gfTop, the price relation and the transformation elasticity, as depicted by the equation xftLandCatEq:

```
 xftLandCatEq(rs(r),aez,landCat,ts(t)) $ (aezFlag(r,aez) and gfTop(r,aez,landCat,t)) ..

   xftLandCat(r,aez,landCat,t)/xftLandCat.scale(r,aez,landCat,t) =e=
        gfTop(r,aez,landCat,t) * xftTop(r,aez,t)/xftLandCat.scale(r,aez,landCat,t)
            * (pftLandCat(r,aez,landCat,t)/pftTop(r,aez,t))**omegaAezTop(aez);
```

The average land price for each land category *pftLandCat* is defined in the equation *pftLandCatEq* using a dual price aggregator and the transformation elasticity *omegaAez*:

```
  pftLandCatEq(rs(r),aez,landCat,ts(t)) $ (aezFlag(r,aez) and  gfTop(r,aez,landCat,t)) ..


  pftLandCat(r,aez,landCat,t) =E=
         sum(landCat_a(landCat,a) $ gfAez(r,aez,a,t),
             gfAez(r,aez,a,t) * pftAez(r,aez,a,t)**(1+omegaAez(aez,landCat)))**(1/(1+omegaAez(aez,landCat)));
```

Accordingly, the distribution of the land in each land category to the different activities in the equation *xftAezEq* is based on the share parameter *gfAez*, the given total *xftLandCat*, the price relation and the transformation elasticity:

```
  xftAezEq(rs(r),aez,aIn(a),ts(t)) $ ( aezFlag(r,aez) and gfAez(r,aez,a,t)) ..
*
   xftAez(r,aez,a,t)/xftAez.scale(r,aez,a,t) =e=
            sum(landCat_a(landCat,a),
                gfAez(r,aez,a,t) * xftLandCat(r,aez,landCat,t)/xftAez.scale(r,aez,a,t)
                 * (pftAez(r,aez,a,t)/pftLandCat(r,aez,landCat,t))**omegaAezTop(aez));
```

As land is not mobile across the AEZ, total factor use of land *xft* and the related price *pft* are defined by the following tree equations. The first equations sums the total endogenous land sue in each aez xftTop to xft. The second equation sums up total factor remuneration and the third one defines the average price:

```
xftLandEq(rs(r),"land",ts(t)) $ ( (Xft.range(r,"land",t) ne 0) $ sum(aezFlag(r,aez),1) $  xftFlag(r,"land")) ..
   xft(r,"land",t)/xft.scale(r,"land",t) =E= sum(aezFlag(r,aez), xftTop(r,aez,t))/xft.scale(r,"land",t);

vftLandEq(rs(r),ts(t)) $ (xftFlag(r,"land") $ sum(aezFlag(r,aez),1)) ..
   vftLand(r,t)/xft.scale(r,"land",t) =E= sum(aezFlag(r,aez), xftTop(r,aez,t)*pftTop(r,aez,t)) /xft.scale(r,"land",t);

pftLandEq(rs(r),"land",ts(t)) $ ((pft.range(r,"land",t) ne 0) $ sum(aezFlag(r,aez),1) $ xftFlag(r,"land")) ..
   pft(r,"land",t)
       =E= vftLand(r,t)/xft(r,"land",t);
```

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:

```
aezFlag(rNat,aez) $ (sum(a, landUse(rNat,aez,a,"vfm"))) = Yes;
```

The land use by activity is initialized reflecting the global scaling factor as long as it is above a minimum threshold:

```
*
 xftAez.l(r,aez,a,t0) $ aezFlag(r,aez) = landUse(r,aez,a,"vfm") * gblScale;
 xftAez.l(r,aez,a,t0) $ (xftAez.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 share are used:

```
xftAez.l(r,aez,a,t0) $ (xf.l(r,"land",a,t0) and aezFlag(r,aez)
                         and (not sum(aezFlag(r,aez1),xftAez.l(r,aez1,a,t0))))
   = xf.l(r,"land",a,t0) * landUse(r,aez,"land","vom")/sum(aezFlag(r,aez1),landUse(r,aez1,"land","vom"));
```

Afterwards, scaling ensures that adding up over the aez recovers the given total:

```
xf.scale(r,"land",a,t0) = sum(aezFlag(r,aez), xftAez.l(r,aez,a,t0));
xftAez.l(r,aez,a,t0) $ xftAez.l(r,aez,a,t0)
   = xftAez.l(r,aez,a,t0) * xf.l(r,"land",a,t0)/xf.scale(r,"land",a,t0);
```

The prices are initialized to zero, and the aggregates defined from summing up, i.e.:

```
xftLandCat.l(r,aez,landCat,t0)   $ aezFlag(r,aez) = sum(landCat_a(landCat,a), xftAez.l(r,aez,a,t0));

xftTop.l(r,aez,t0)       = sum(landCat, xftLandCat.l(r,aez,landCat,t0));
```

Inspecting the results

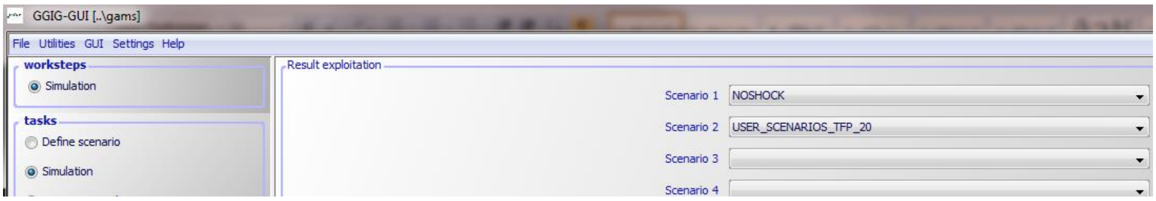
Before we turn to the exploitations tool of the GUI, the reader should be reminded that CGEBox allows for a wider range of possibilities to look at results:

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 the Jacobian entries. The numbers will limit the output of individual instances shown and can be set by the GUI. For more information on how to interpret that output, please look at the GAMS manual.

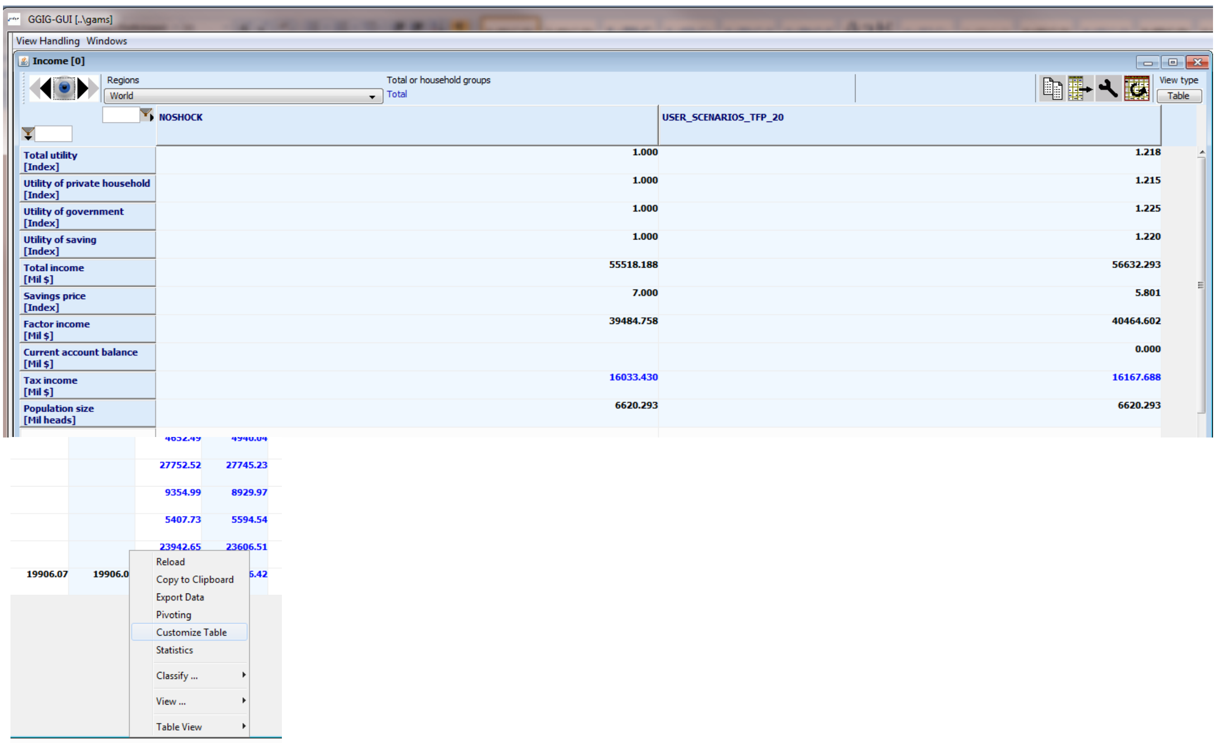
3. If the output option GDX is switched on, all GAMS symbols (sets, parameters, variables) will be outputted to a GDX container and can be analyzed in a GDX viewer, either the one delivered with the GAMS IDE or the GDX viewer built into the GUI.
4. The model's results can be mapped back into a SAM, and 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.

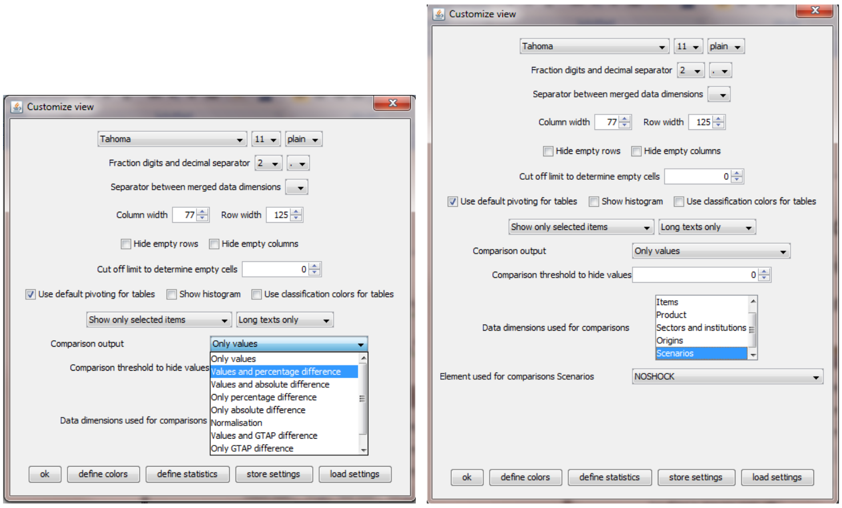
Inspecting results with GUI is nothing new: You have done that already if you followed the short introduction … so press on the "Exploit results" button and next on "Show results" again. Use the "NoShock" and your own scenario:



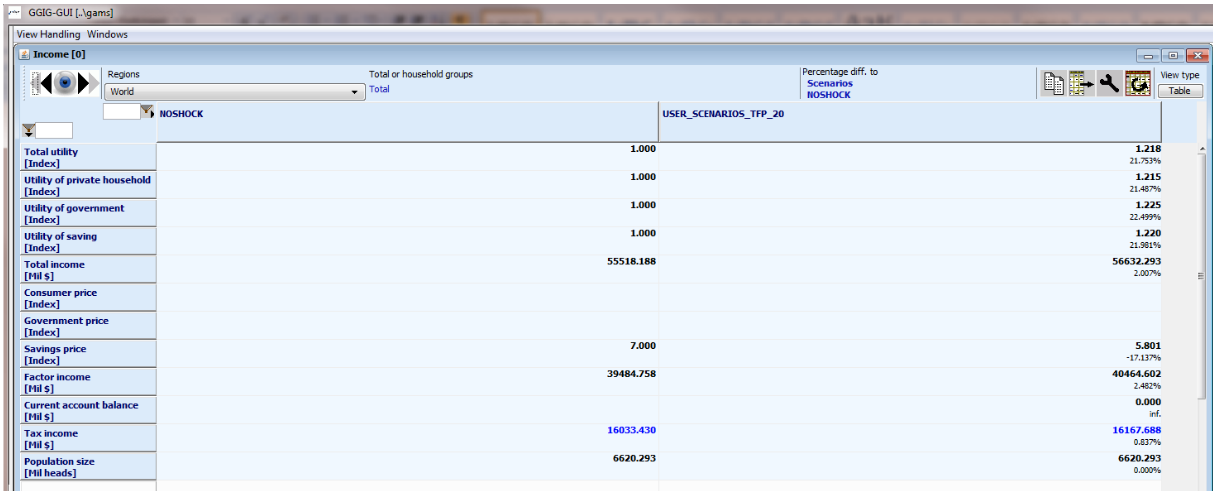
And now, you can see the counterfactual and the simulation side by side in the tables:



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:



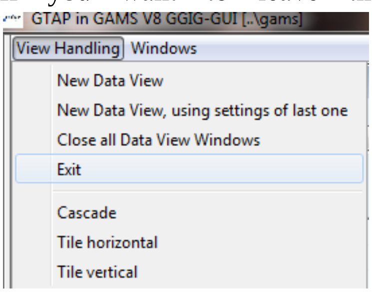
After closing the dialogue, you find now relative difference in small numbers beyond the simulated results in each cell of the table.



(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.

If you want to leave the viewer again, use "Exit" from "View Handling" menu:

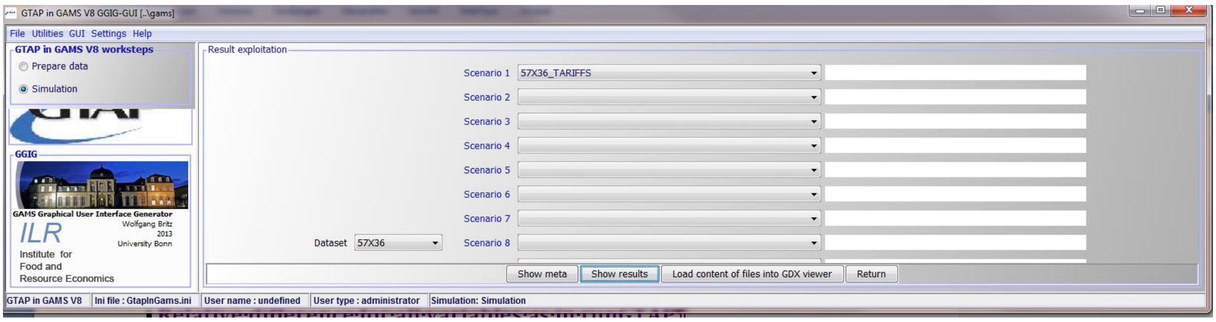


Notes for GEMPACK users or users familiar with other CGE models

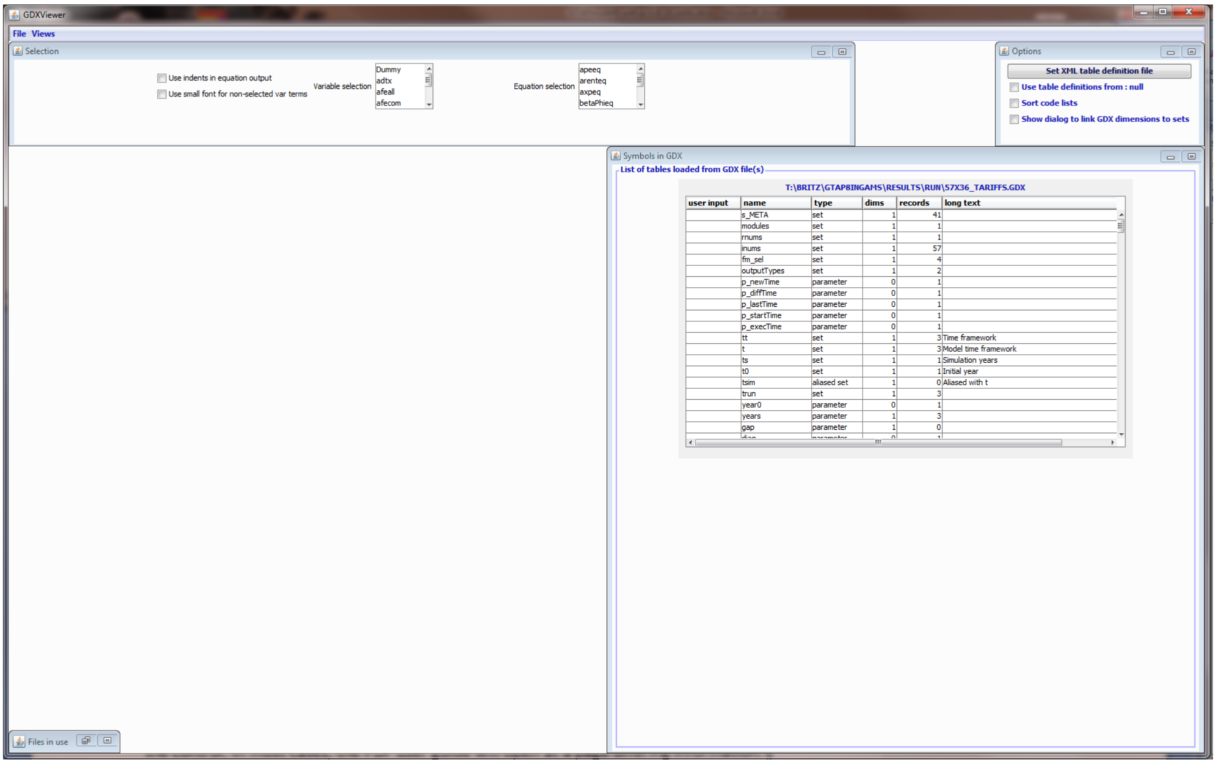
- The view “Model overview, SAM” shows a SAM generated from the model results (including row and column sums and potential imbalances) if the option “Store SAM in GDX” is switched on.
- Many tables comprise additional information on aggregates on top of the other rows. Only in some cases is what is called “total” a simple sum of the rows below. Thus, the information should generally not be confused with the “sum” information shown in “viewhar”.
- 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.

Relative difference for all variables as in runGTAP

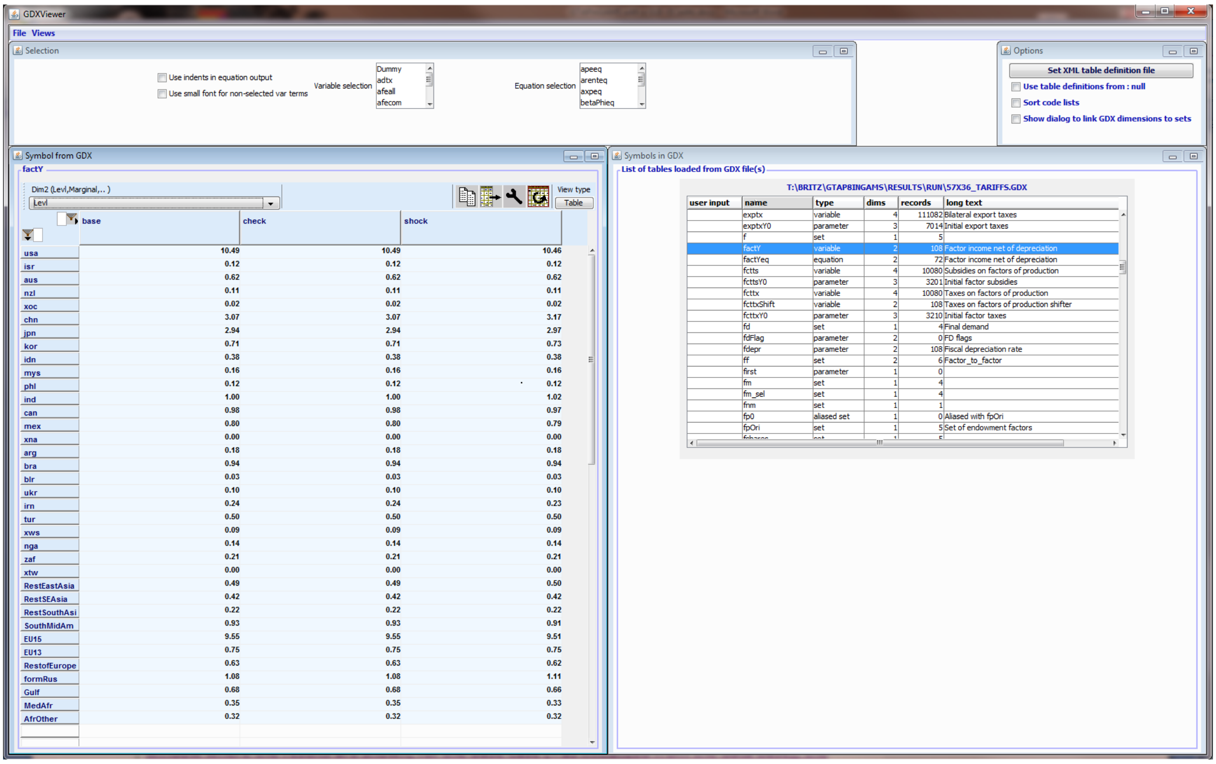
GEMPACK users typically compare simulated against benchmark levels for model variables. That is also possible with the interface. In order to do so, select the file with the result and press the “Load content of files into GDX viewer” button:



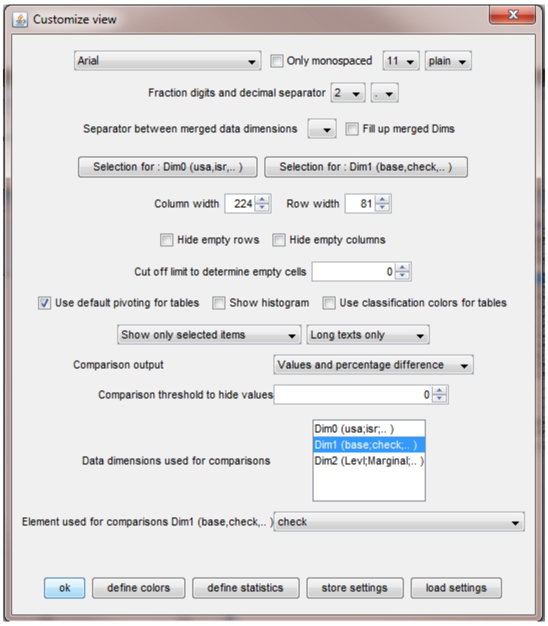
That will open a new window as shown below:



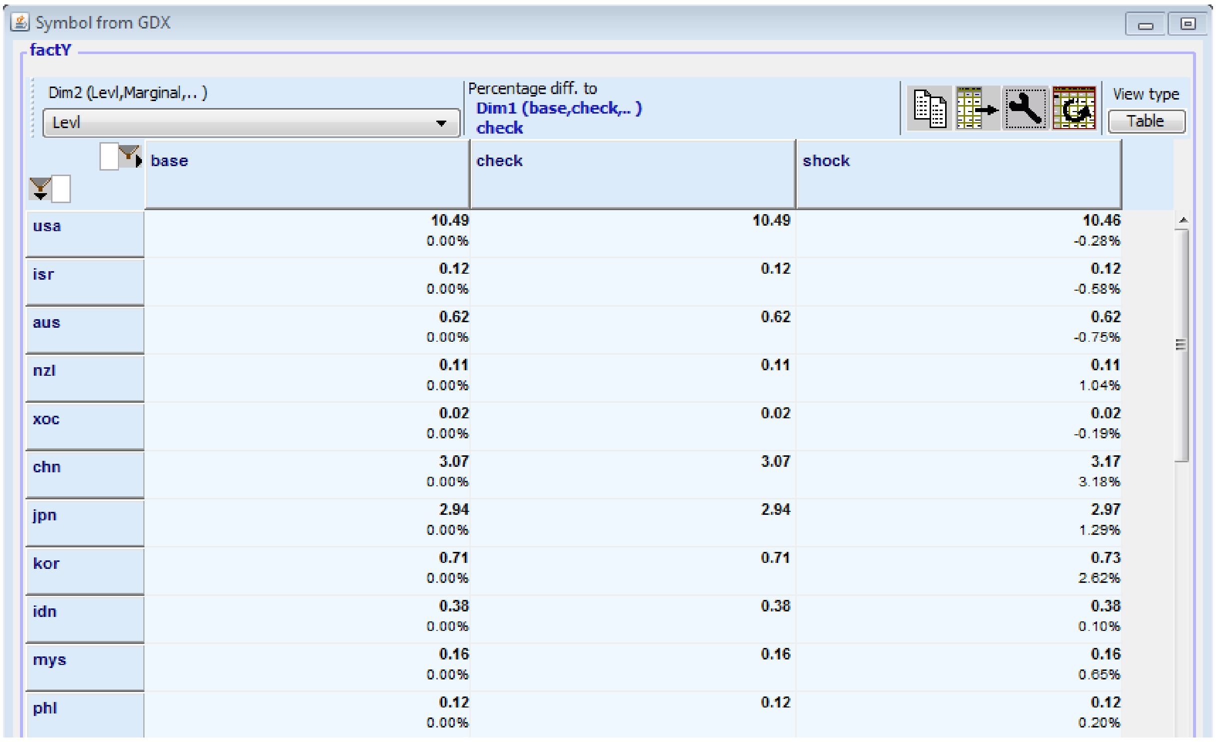
The window "List of table loaded from GDX file(s)" shows the status of all GAMS symbols (variable, equations, sets, parameters) after the solution of the shock. Double-click on any symbol you are interested in, e.g. the factor income "facty" as shown in the screen shot below. That will open a window 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:

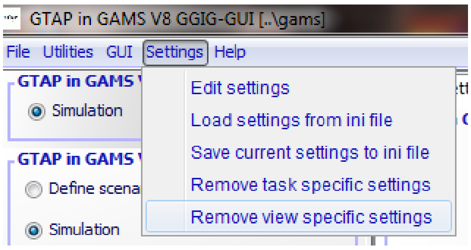


Trouble shooting with the viewer

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:

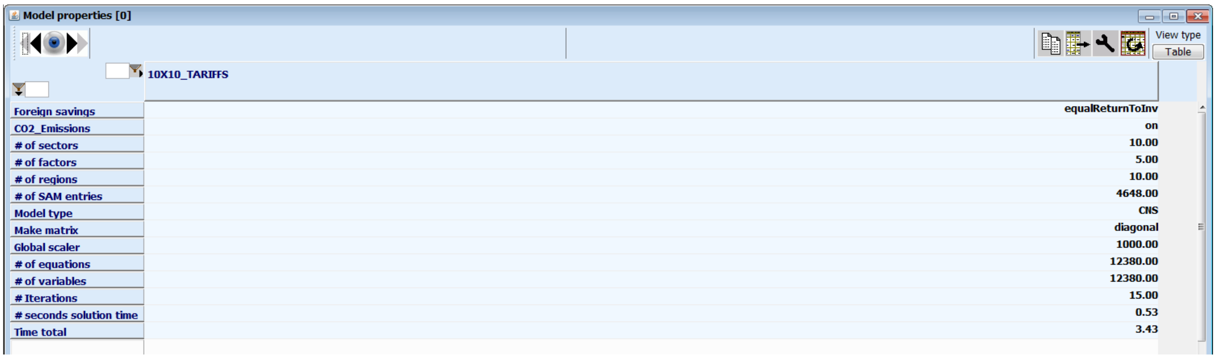
1.  use the “close” button to remove the view, and use “View Handling”, “New Data View” to open a fresh one.
2. Leave the viewer (“View Handling”, “Exit”), and click on “Remove view specific settings” under “Settings” in the menu bar. That will set the viewer back to the original defaults.



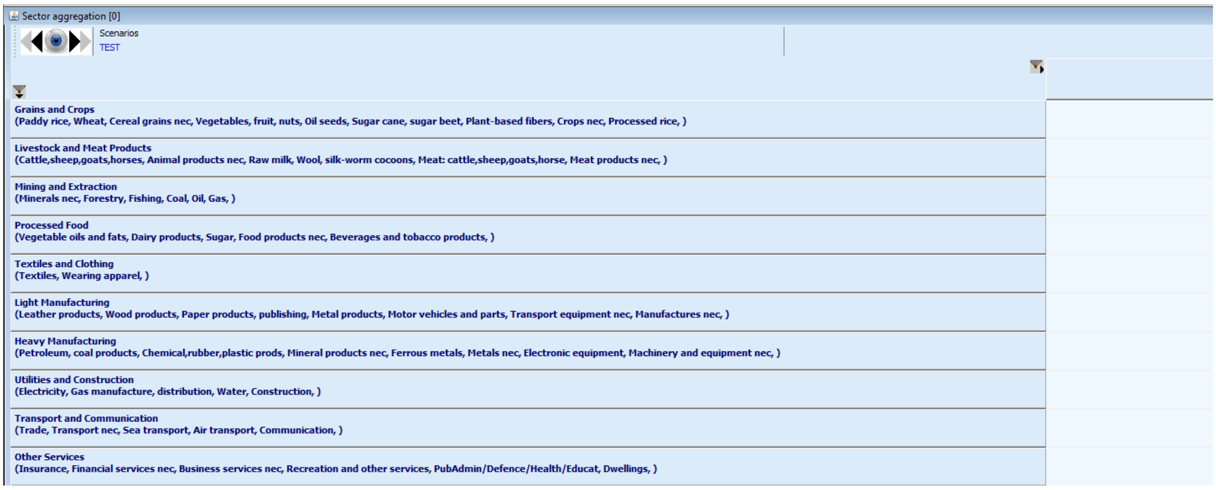
3. If that does also not help, try to close and re-open the application.
4. If the viewer still shows curious things, it is most probably some programming error … contact the author.

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:



There are also two tables which report the aggregation of the data base under **Model overview/regional aggregation** and **Model overview/sectoral aggregation**:



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	MENA	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.

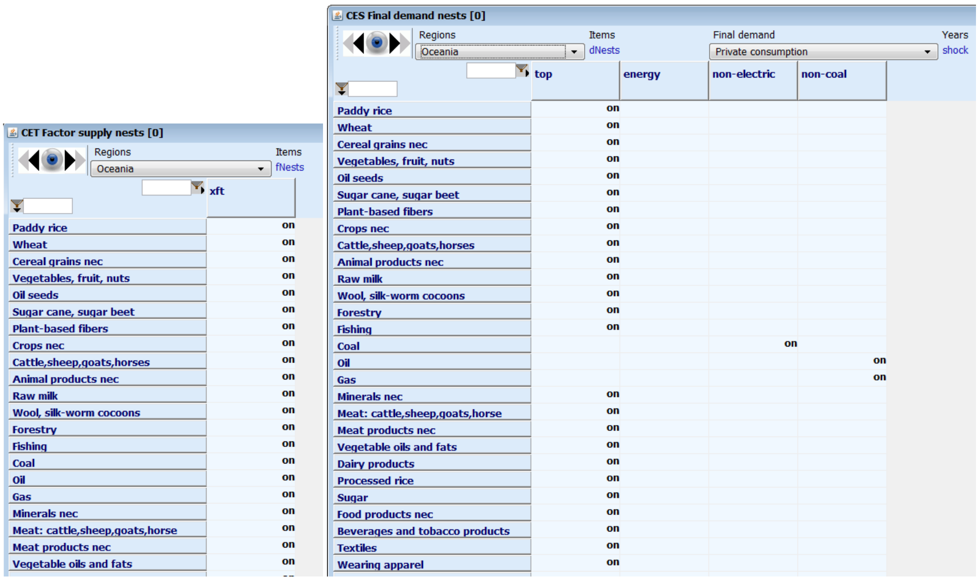
Model Overview/CES products nests reports the nesting structure used in the production module:

CES production nests [0]

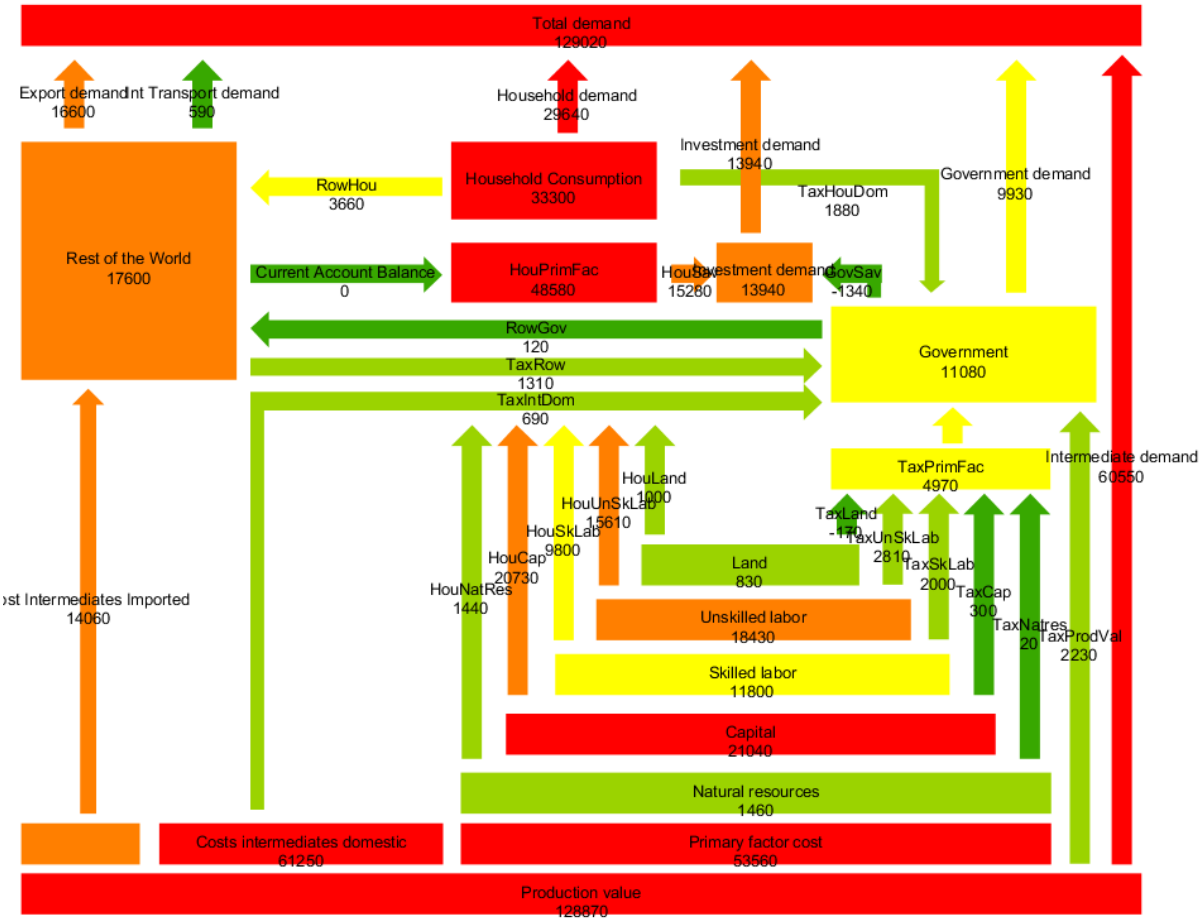
Regions: Oceania | Items: tNests | sectors: Paddy rice | Years: shock

	ND	VA	Labor	CAP+ENE	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						

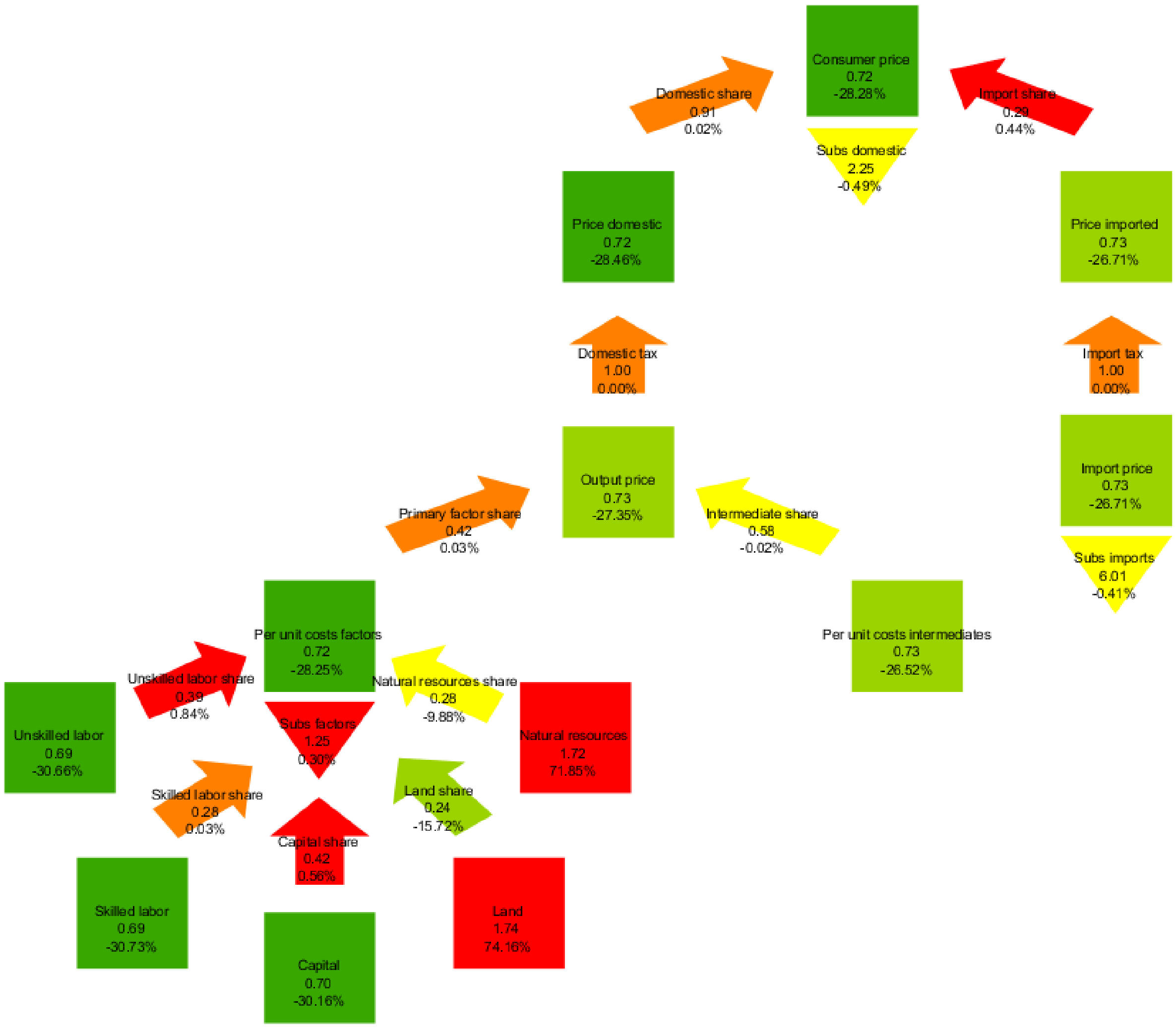
Similar views are available for the nesting used for factor supply and final demand:



Model Overview/Flow schemer uses the in-built mapping utility in a somewhat unusual way to draw a flow chart with all transactions in the economy, some aggregated. The lowest rectangle shows the total production value; the related cost are split into intermediate (imported and domestic) demand on the LHS rectangles above, and total primary factor costs on the RHS rectangle above. Total primary factor income remuneration for each factor is split up into tax (arrows on right) and household income (arrows to the left). The rectangle "Government" shows the government account; incoming arrows show tax income, outgoing ones government spending (for domestic, imports and saving; the latter is residually calculated). The rectangle "Household consumption" shows final consumption by the household, the arrow to its left shows spending for imports net of taxes. The rectangle on the top is final demand; it is sourced by export and international transport demand (left), household, saving and government demand, and domestic intermediate demand for that sector. The schemer view, as with any view, can also show differences between scenarios and provides an aggregation to total world. The schemer can also be used to visualize individual sectors and commodities, or an aggregation to total output/demand.


Total demand
129020
Export demand
16600
Int Transport demand
590
Household demand
29640
Investment demand
13940
Government demand
9930
Rest of the World
17600
RowHou
3660
Household Consumption
33300
TaxHouDom
1880
Current Account Balance
0
HouPrimFac
48580
HouSav
15260
Investment demand
13940
GovSav
-1340
RowGov
120
TaxRow
1310
TaxIntDom
690
Government
11080
TaxPrimFac
4970
Intermediate demand
60550
HouLand
1000
HouUnSkLab
15610
HouSkLab
9800
HouCap
20730
HouNatRes
1440
TaxLand
-170
TaxUnSkLab
2810
TaxSkLab
2000
TaxCap
300
TaxNatres
20
TaxProdVal
2230
Land
830
Unskilled labor
18430
Skilled labor
11800
Capital
21040
Natural resources
1460
Costs intermediates domestic
61250
Primary factor cost
53560
Production value
128870

Similarly, **Model Overview/Price schemer** visualizes the composition of per unit costs and prices, where the boxes represent price, the arrows show costs shares respectively tax base multipliers and the triangles substitution elasticities. Combined, as shown below, with a percentage difference to the baseline, it allows a rather straightforward interpretation of major price changes in the model. The results are available per region and sector/commodity, as well as aggregated to total output respectively private consumption and world level.



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:

Money metric decomposition [1]

Regions: World — Total

	10x10_Tariffs
Total [Bio $]	33.32
Output price index [Bio $]	155.10
Private price index [Bio $]	111.53
Government price index [Bio $]	7.66
Savings price index [Bio $]	35.65
Income total [Bio $]	-121.63
Factor income [Bio $]	2.58
Indirect tax income [Bio $]	-147.84

Money metric decomposition, by product [1]

Regions: World — Sectors and ...: Money metr...

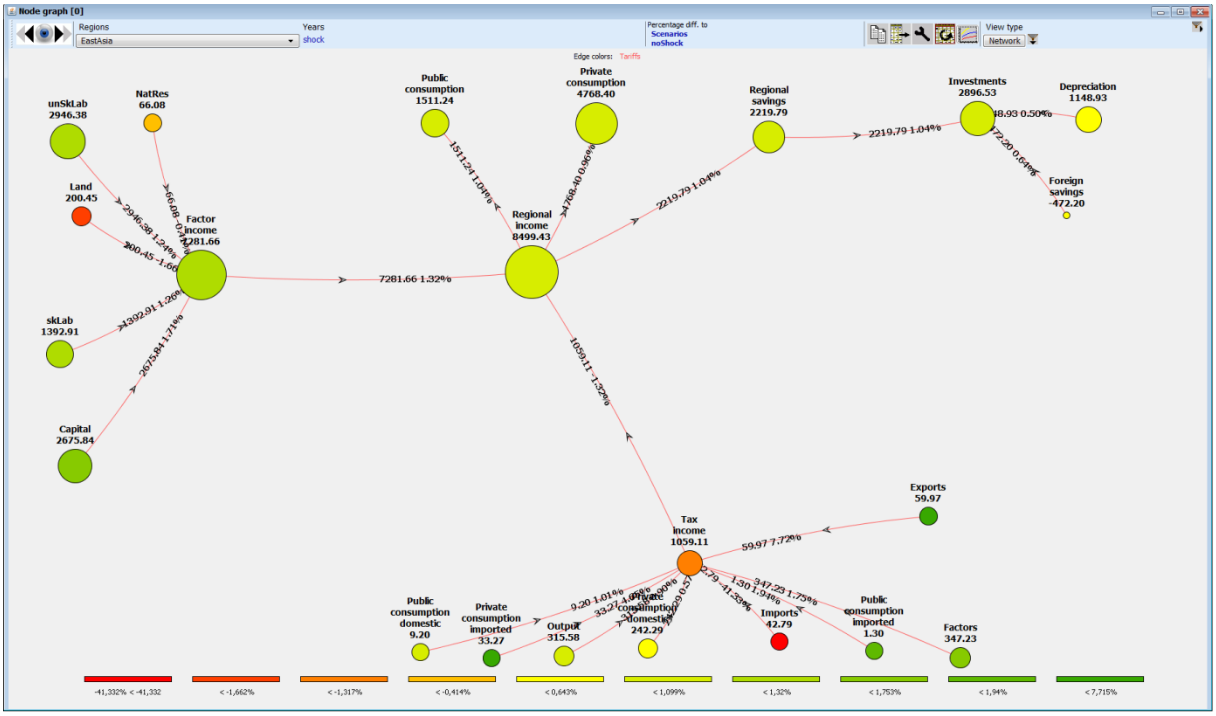
	10x10_Tariffs
Grains and Crops	0.47
Livestock and Meat Products	2.07
Mining and Extraction	-5.19
Processed Food	20.13
Textiles and Clothing	11.16
Light Manufacturing	19.70
Heavy Manufacturing	26.95
Utilities and Construction	8.42
Transport and Communication	5.85
Other Services	7.92

Money metric decomposition, by factor [1]

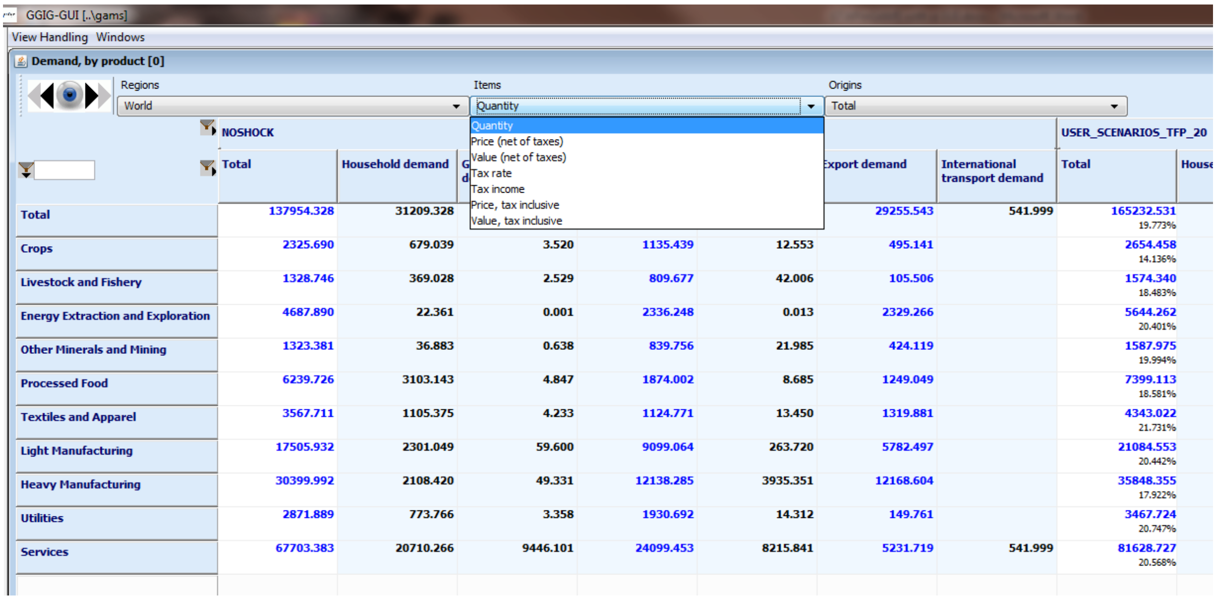
Regions: World

	10x10_Tariffs
Unskilled labor	-0.57
Skilled labor	1.66
Capital	6.82
Land	-7.91
Natural resources	2.58

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:

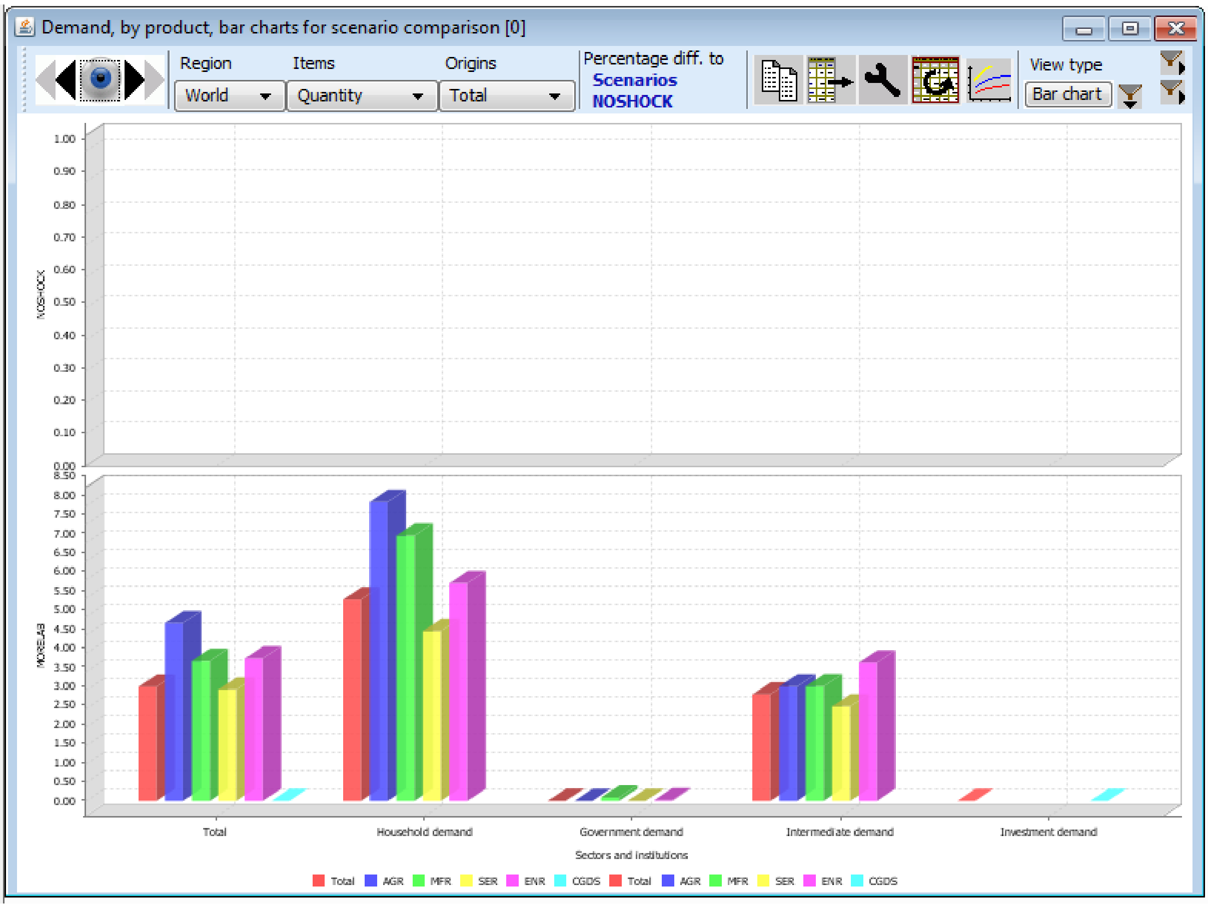


Markets/Demand by product: Shows a table with demand by institutions, products and totals shown in the row. The items box allows for looking at quantities, prices, tax rates, value and tax income, the origins by "domestic", "imported" and "total". The first box allows selecting the regions. Note: total is an aggregation over all products.

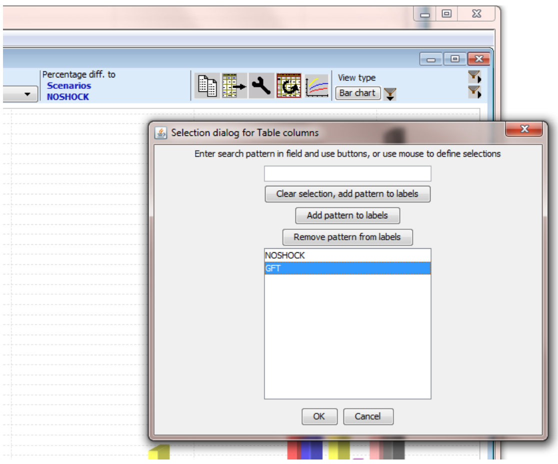


	NOSHOCK							USER_SCENARIOS_TFP_20	
	Total	Household demand	G d			Export demand	International transport demand	Total	House
Total	137954.328	31209.328				29255.543	541.999	165232.531 19.773%	
Crops	2325.690	679.039	3.520	1135.439	12.553	495.141		2654.458 14.136%	
Livestock and Fishery	1328.746	369.028	2.529	809.677	42.006	105.506		1574.340 18.483%	
Energy Extraction and Exploration	4687.890	22.361	0.001	2336.248	0.013	2329.266		5644.262 20.401%	
Other Minerals and Mining	1323.381	36.883	0.638	839.756	21.985	424.119		1587.975 19.994%	
Processed Food	6239.726	3103.143	4.847	1874.002	8.685	1249.049		7399.113 18.581%	
Textiles and Apparel	3567.711	1105.375	4.233	1124.771	13.450	1319.881		4343.022 21.731%	
Light Manufacturing	17505.932	2301.049	59.600	9099.064	263.720	5782.497		21084.553 20.442%	
Heavy Manufacturing	30399.992	2108.420	49.331	12138.285	3935.351	12168.604		35848.355 17.922%	
Utilities	2871.889	773.766	3.358	1930.692	14.312	149.761		3467.724 20.747%	
Services	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.

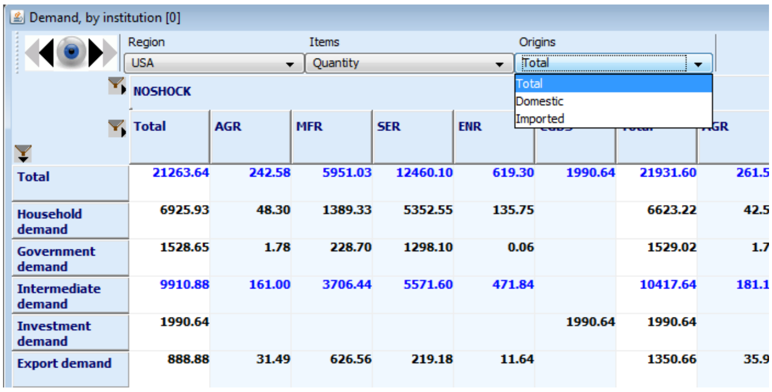


Note: you use the second "filter" bottom next to view type to "view type" (double click on it) to open a dialogue which lets you remove the benchmark scenario:



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 (Total / Domestic / Imported)

NOSHOCK

	Total	AGR	MFR	SER	ENR			GR
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

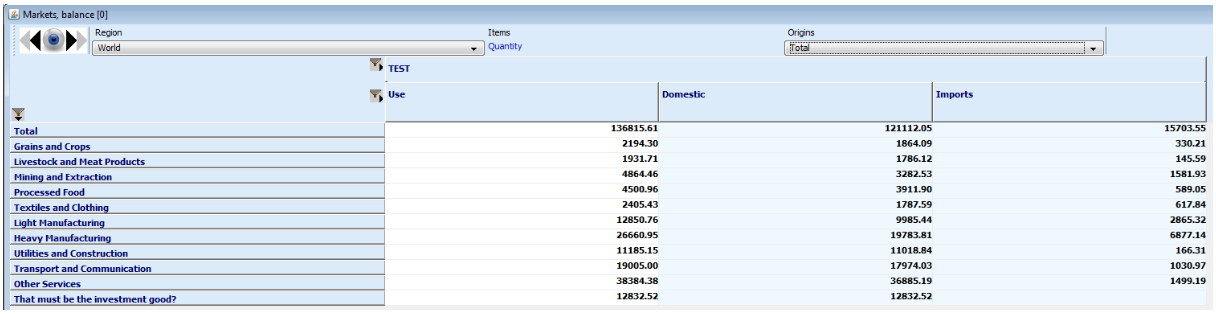
Markets/Final demand nests: shows the demand aggregates –quantities, prices and values – according to the active demand nesting:

Final demand nests [0]

Regions: Oceania | Items: Quantity | Origins: Demand

	Private consumpti	Governme nt	Investme nt
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:

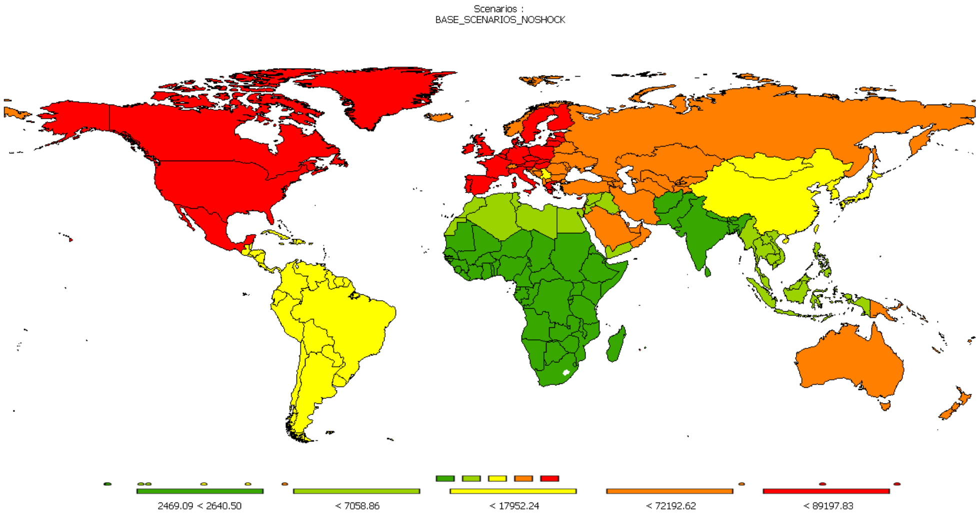
Demand, by product, per capita in $ [0]

Region: World | Items: Q | Origins: Total | View type: Table

BASE_SCENARIOS_NOSHOCK

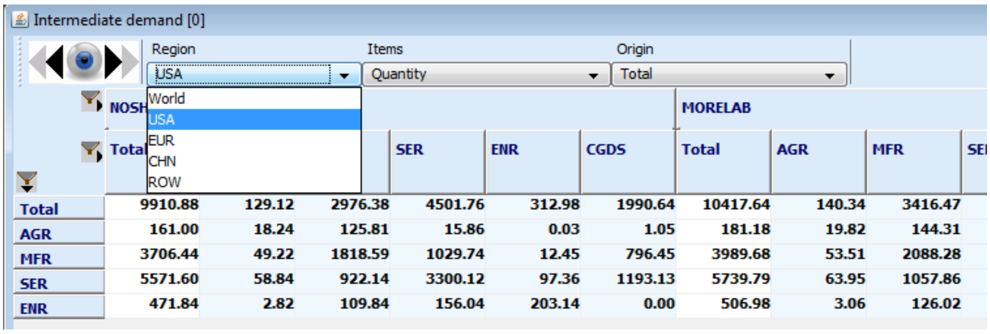
	Total	Household demand	Government demand	Intermediate demand	Investment demand	Export demand
Total	19660.35	4602.31	1424.48	9700.63	1938.36	1924.02
Grains and Crops	217.49	89.83		111.79		15.87
Livestock and Meat Products	212.19	101.69		100.34		10.15
Mining and Extraction	638.36	7.88		444.66		185.82
Processed 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:**



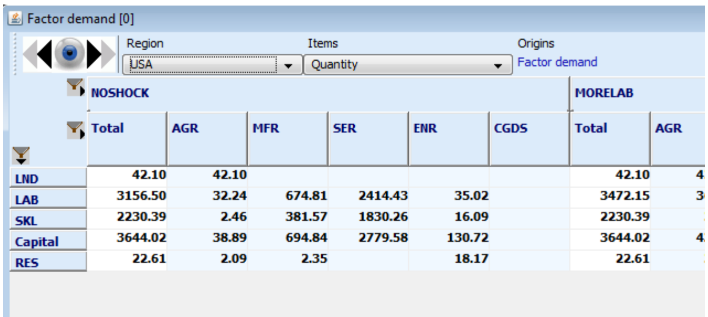
Maps can be made from any table by putting the regions in the rows and choosing “Map” under “View type”.

Markets/Intermediate demand: break of the intermediate demand by sector. Note: the total in the columns is an aggregation over all sectors.



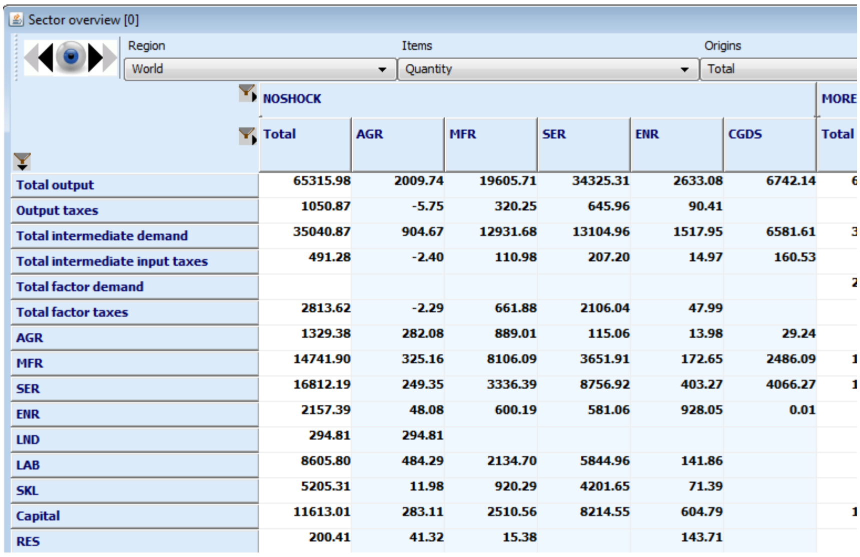
	NOSH						MORELAB			
	Total			SER	ENR	CGDS	Total	AGR	MFR	SEI
Total	9910.88	129.12	2976.38	4501.76	312.98	1990.64	10417.64	140.34	3416.47	
AGR	161.00	18.24	125.81	15.86	0.03	1.05	181.18	19.82	144.31	
MFR	3706.44	49.22	1818.59	1029.74	12.45	796.45	3989.68	53.51	2088.28	
SER	5571.60	58.84	922.14	3300.12	97.36	1193.13	5739.79	63.95	1057.86	
ENR	471.84	2.82	109.84	156.04	203.14	0.00	506.98	3.06	126.02	

Markets/Factor demand: report primary factor use, price etc. by sector



	NOSHOCK						MORELAB	
	Total	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.



	NOSHOCK						MORE
	Total	AGR	MFR	SER	ENR	CGDS	Total
Total output	65315.98	2009.74	19605.71	34325.31	2633.08	6742.14	6
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	3
Total intermediate input taxes	491.28	-2.40	110.98	207.20	14.97	160.53	
Total factor demand							2
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	1
SER	16812.19	249.35	3336.39	8756.92	403.27	4066.27	1
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		1
RES	200.41	41.32	15.38		143.71		

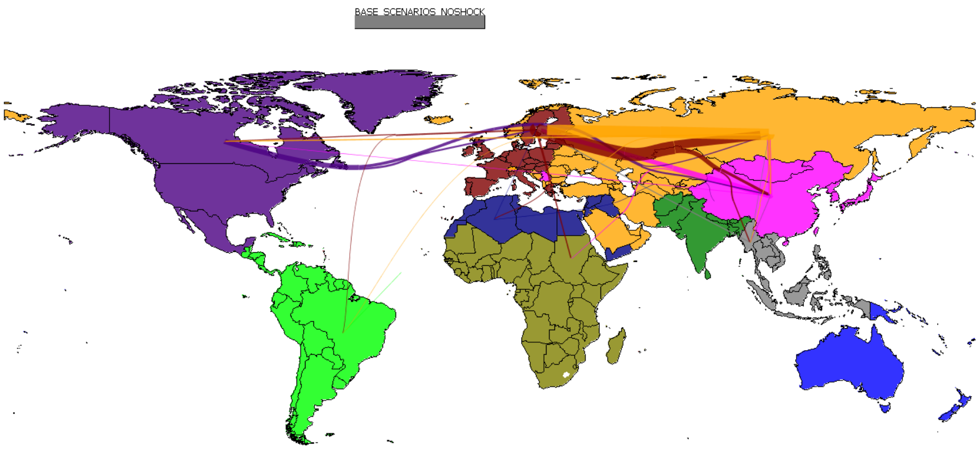
Sectors/Sector cost shares: a selection of the value item of the view above, normalized by total output:

Cost shares [0]

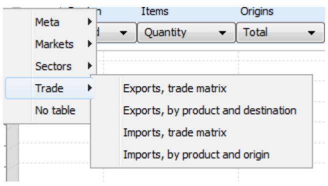
Region: World — Items: Value — Origins: Domestic

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.



Sensitivity analysis

GEMPACK supports systematic sensitivity analysis (Channing 1996). The standard GTAP in GAMS model 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 single sensitivity experiments, in order to use as much as possible existing code and to avoid duplicating code fragments.
2. A second GAMS program in combination with utility from SANDIA labs[20] defines the experiments and executes in parallel GAMS children, which solve the same scenario using each its own parameter settings.
3. That second GAMS mother program collects the results from the children and combines them into one GDX file.

Currently, the sensitivity analysis allows changing substitution and transformation elasticities by defining *relative* ranges around the given parameters, plus generates on demand additional draws for two “shocks”, which needs to be translated by changes in the GAMS code into something useful. These relative ranges need not be symmetric. The draws can be defined generally in two modes:

1. The same relative change is applied simultaneously for all sector/factors/shocks. To give an example: if the drawn relative change for the factor transformation is 0.5, all transformation elasticities are reduced by 50%.
2. Individual draws for each sector and factor are generated.

Whereas Arndt 1996 uses Gaussian Quadratures, the approach in here is based on a combination of Latin Hypercube Sampling (LHS) and an entropy estimator. 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, truncated distributions. In the default case, the LHS simply returns (more or less) uniform distributed draws across the factor range, i.e. between the relative lower and upper limit chosen by the user, with an average presenting the mean between the lower and upper limit.

If the user has chosen non-symmetric bounds around zero, the simulations would hence lead to a systematic deviation from the mean parameter used without sensitivity analysis. The entropy estimator can therefore be used to determine probability weights for the draws which ensure that the weighted average over the draws for each relative factor change is close to unity. That approach is hence quite similar to Gaussian Quadrature which also applies weights to the observations. The difference here is that a uniform distribution over the chosen range is the preferred distribution, while the Gaussian Qudratures used in GEMPACK tries to mimic normal or triangular distributions.

The user has the choice between *three difference distributions*, all truncated at the chosen lower and upper limit:

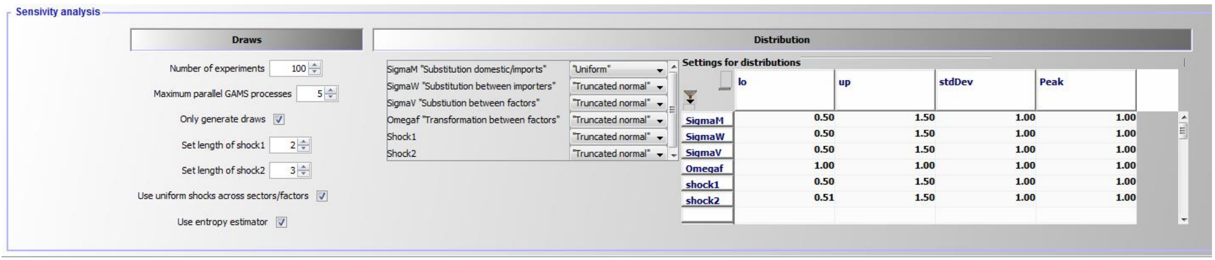
1. Uniform
2. Truncated Normal distribution
3. Triangular distribution

For the truncated Normal, the standard deviation must be set, for the triangular distribution the peak. The combination of user set truncation points and distributions allows for a rather flexible design of the experiments. The program can also easily be extended to also apply sensitivity analysis to shocks.

[20] http://www.sandia.gov/

The general set-up of the sensitivity analysis is defined on the left-most group of controls: the number of experiments (= parameter draws and related model solves), the maximum number of parallel solves.

Controls for sensitivity experiments



The left hand side determines the number of draws and how many GAMS processes are used in parallel to simulate the different experiments. The user might also chose “Only generate draws” to prevent the actual model runs, in order to check the layout of the scenarios.

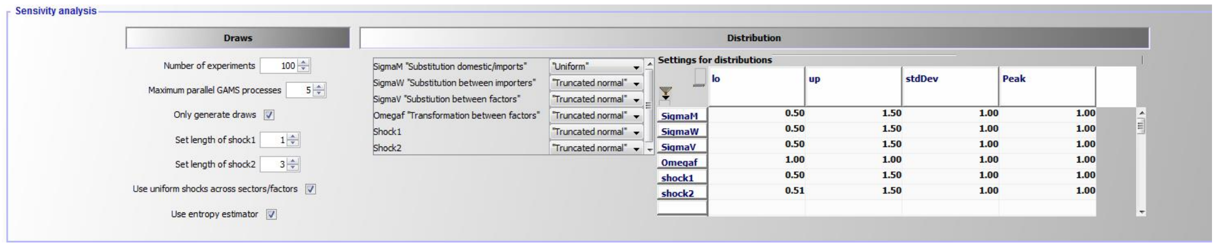
Using uniform shocks across sector/factors will apply the same relative change for the chosen parameters to all sectors and factors. Otherwise, each sector / factor will receive its own random shock in each experiment. The panel on the right hand side allows choosing the type of distribution for each parameter and the related settings. The “Peak” defines the mode for the triangular case.

Entropy estimator

In the case of non-symmetric truncation points and use of triangular or normal distributions, as in the last example above, the mean of the draws will deviate from unity. On demand, an entropy estimator will try to find probability weights for the draws which recover a mean of unity. If no solution is found, the estimator tries bounds around each mean of +/-1%, +/- 2.5% and +/-5%. If that fails, the program will abort with an error.

Without the entropy estimator, each draw receives an equal weight and the mean relative parameter draws might deviate from unity.

Drawing systematic experiments for other elements of the simulation



As seen from above, the interface allows defining two “shocks” with related set lengths. The simplest layout is shown above: one shock with a set length of unity. The automatically generated include files will each carry lines as follows:

```
parameter shock1(*);set s1/s1*s1/;
 shock1('s1') =1.4839095406;
```

The shock1 parameter can be used to e.g. change the population size in all regions by the same factor. If regional specific shocks are needed, the length of the set must be equal or higher than the number of regions. Assume e.g. a dis-aggregation with 5 regions, and the set length of shock1 accordingly increased to 5. The include file will comprise lines as follows:

```
parameter shock1(*);set s1/s1*s5/;
 shock1('s1') =0.6891764639;
 shock1('s2') =0.7597044235;
 shock1('s3') =0.6128377057;
 shock1('s4') =1.2482232299;
 shock1('s5') =1.3406050177;
```

With a cross set between r and s1, using the **pos** operator from GAMS, the numerical values on shock1 can be mapped e.g. on the population size:

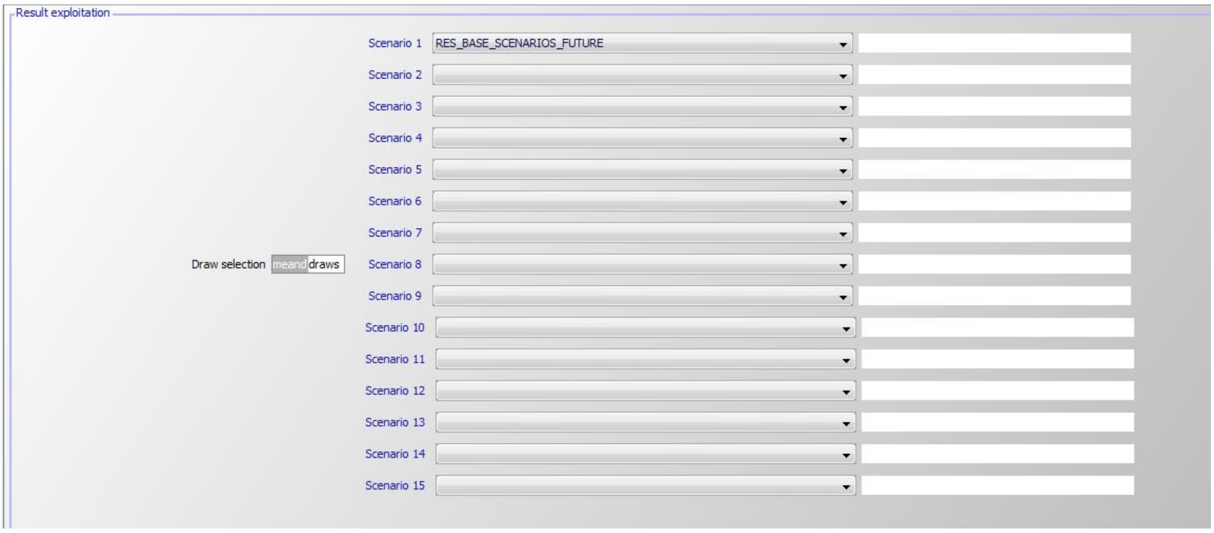
```
set r_to_s1(r,s1);

r_to_s1(r,s1) $ (r.pos eq s1.pos) = yes;
pop(r) = pop(r) * sum(r_to_s1(r,s1), shock1(s1));
```

The two shocks in combination with flexible set lengths should hence allow drawing from experiments for different types of shocks quite easily without the need to change the GAMS code for the sensitivity analysis. Note that these shocks can be combined with changes to the parameters.

Parallel solves and analyzing the results

The actual solves for each individual experiments start from a savepoint generated after solving the benchmark, such that execution time is only spend on model generation, solving and post-model reporting. The solve times reported above stem from a by now medium fast four-core desktop and underline that from a computational view point, sensitivity analysis is no longer a concern.

The interface will allow showing either all draws and/or the weighted average of the draws (meand):



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 “meand” and the (2) result at the original parameter “mean” – what one would simulate without any sensitivity analysis:

Demand, by product [0]

Region: World | Items: Value | Origins: Total | Percentage diff. to: Draws mean

RES_USER_SCENARIOS_GFT

		Total	Household demand	Government demand	Intermediate demand	Investment demand	Export demand	International transport demand
Total	meand	131296.89 -0.02%	30592.51 -0.02%	9509.66 -0.02%	64336.11 -0.03%	12834.53 -0.02%	13503.61 -0.03%	520.48 -0.04%
	mean	131327.41	30598.80	9511.28	64352.24	12837.40	13507.00	520.67
Grains and Crops	meand	1462.23 -0.13%	591.16 -0.12%		751.72 -0.13%		119.35 -0.10%	
	mean	1464.07	591.89		752.71		119.47	
Livestock and Meat Products	meand	1410.77 -0.05%	674.03 -0.08%		665.66 -0.04%		71.07 0.12%	
	mean	1411.47	674.56		665.91		70.99	
Mining and Extraction	meand	4261.06 -0.09%	51.20 -0.14%		2871.75 -0.10%		1338.11 -0.07%	
	mean	4264.97	51.27		2874.67		1339.03	
Processed Food	meand	4244.36 -0.02%	2477.29 -0.03%		1248.83 -0.02%		518.24 -0.01%	
	mean	4245.42	2478.05		1249.07		518.30	
Textiles and Clothing	meand	1744.34 -0.08%	683.58 -0.04%		642.55 -0.15%		418.20 -0.03%	
	mean	1745.66	683.84		643.49		418.33	
Light Manufacturing	meand	12733.19 -0.01%	2332.11 -0.01%	29.84 -0.04%	7643.46 -0.01%		2727.78 -0.01%	
	mean	12734.61	2332.41	29.86	7644.27		2728.07	
Heavy Manufacturing	meand	25936.77 -0.03%	2292.69 -0.02%	17.20 -0.04%	17176.41 -0.02%		6450.47 -0.03%	
	mean	25943.26	2293.26	17.21	17180.67		6452.12	
Utilities and Construction	meand	10878.77 -0.03%	834.30 -0.04%	11.21 -0.12%	10030.96 -0.02%		2.31 -0.38%	
	mean	10881.54	834.60	11.22	10033.40		2.32	
Transport and Communication	meand	18524.80 -0.02%	7878.17 -0.02%	100.56 -0.04%	9187.13 -0.02%		838.46 -0.01%	520.48 -0.04%
	mean	18528.80	7879.68	100.60	9189.25		838.58	520.67
Other Services	meand	37266.07 -0.01%	12777.97 -0.01%	9350.83 -0.02%	14117.64 -0.01%		1019.62 -0.02%	
	mean	37270.21	12779.25	9352.39	14118.79		1019.78	
That must be the investment good?	meand	12834.53 -0.02%				12834.53 -0.02%		
	mean	12837.40				12837.40		

Note: In the example above, relative differences to the case with standard parameters, "mean" where added.

If the draws are selected, an additional dimension is introduced in most tables:

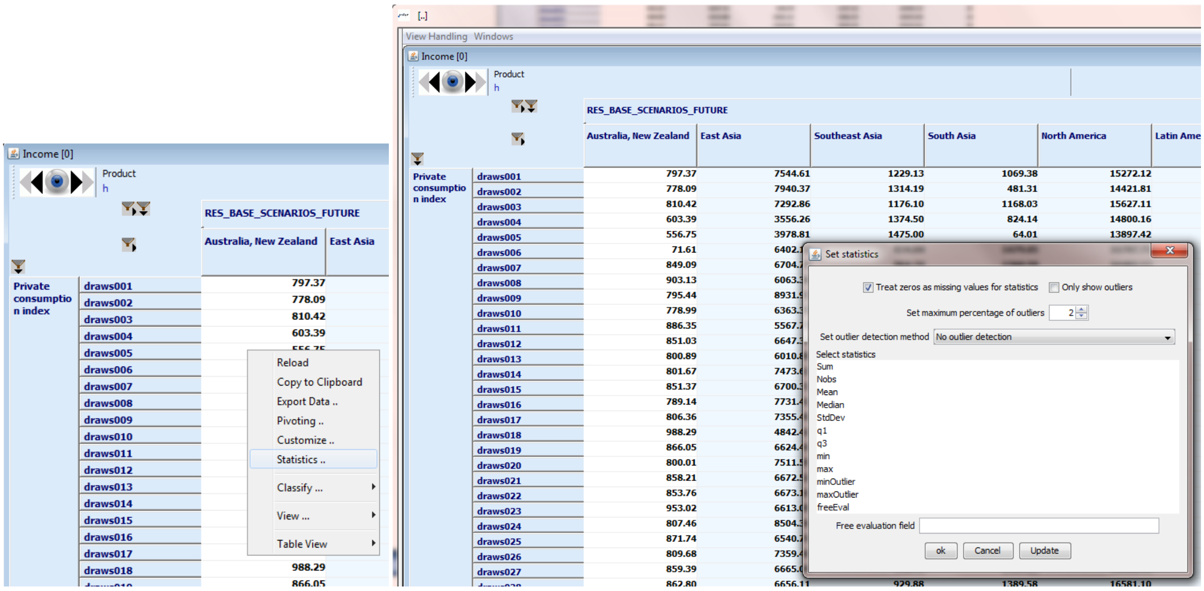
Income [0]

Product: h

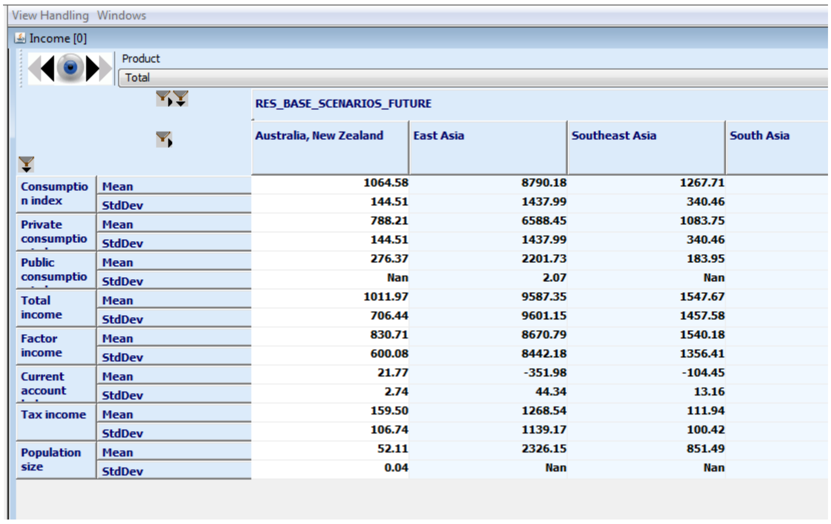
RES_BASE_SCENARIOS_FUTURE

		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	[illegible]
	draws002	778.09	7940.37	1314.19	481.31	14421.81	[illegible]
	draws003	810.42	7292.86	1176.10	1168.03	15627.11	[illegible]
	draws004	603.39	3556.26	1374.50	824.14	14800.16	[illegible]
	draws005	556.75	3978.81	1475.00	64.01	13897.42	[illegible]
	draws006	71.61	6402.12	834.04	1479.05	16787.71	[illegible]
	draws007	849.09	6704.72	964.74	1360.59	16482.53	[illegible]
	draws008	903.13	6063.38	1574.94	1249.33	17305.68	[illegible]
	draws009	795.44	8931.93	1509.94	526.53	13246.60	[illegible]
	draws010	778.99	6363.35	766.56	1240.33	11474.69	[illegible]
	draws011	886.35	5567.74	6.94	152.76	11201.23	[illegible]
	draws012	851.03	6647.34	939.79	1357.23	16505.33	[illegible]
	draws013	800.89	6010.88	1021.11	1002.45	16943.04	[illegible]
	draws014	801.67	7473.63	1219.41	1076.35	15317.61	[illegible]
	draws015	851.37	6700.31	955.51	1368.90	16510.20	[illegible]
	draws016	789.14	7731.41	1268.88	971.92	14990.37	[illegible]
	draws017	806.36	7355.40	1185.63	1159.15	15576.62	[illegible]
	draws018	988.29	4842.42	2090.11	429.43	15733.19	[illegible]
	draws019	866.05	6624.44	927.46	1393.34	16591.68	[illegible]
	[illegible]	800.01	7511.59	1216.66	1133.76	15436.62	[illegible]

That allows a user 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:

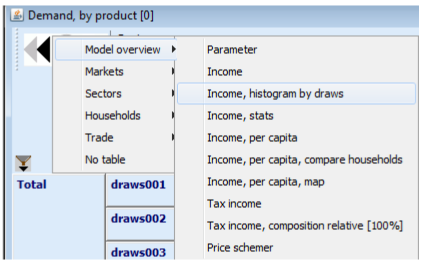


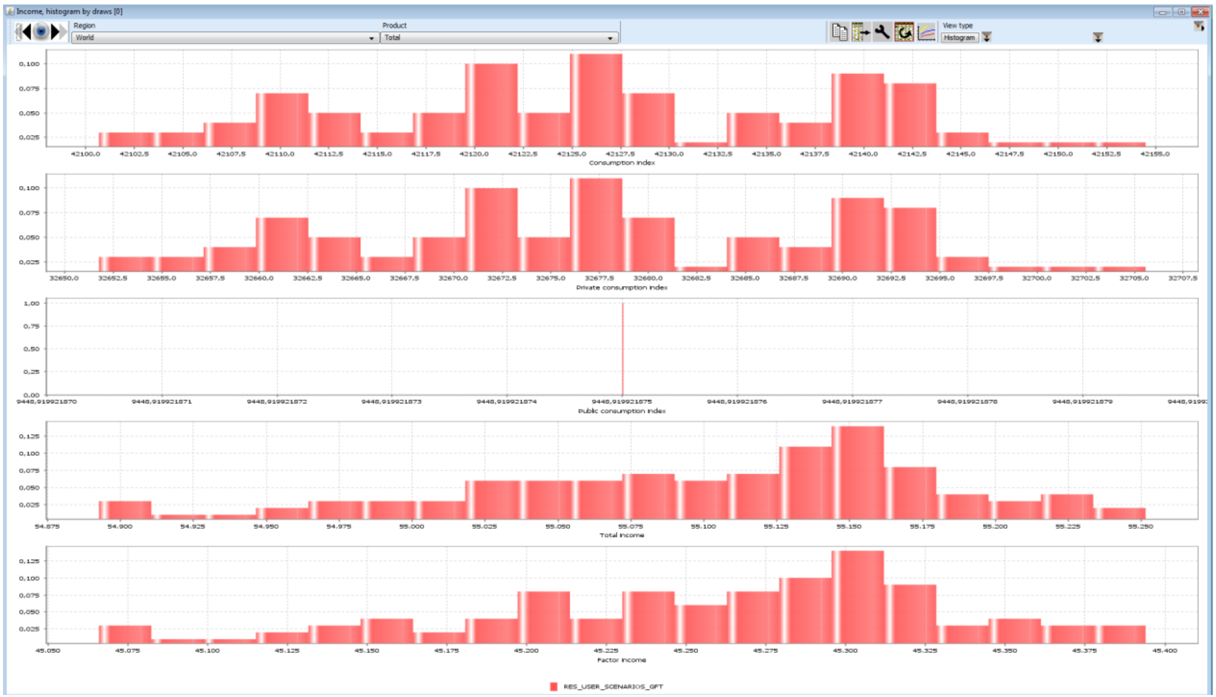
Select the statistics you want to be shown (such as mean and stdDev) and select "Only show outliers" to only show outliers and statistics:


Income [0]
Product: Total
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	

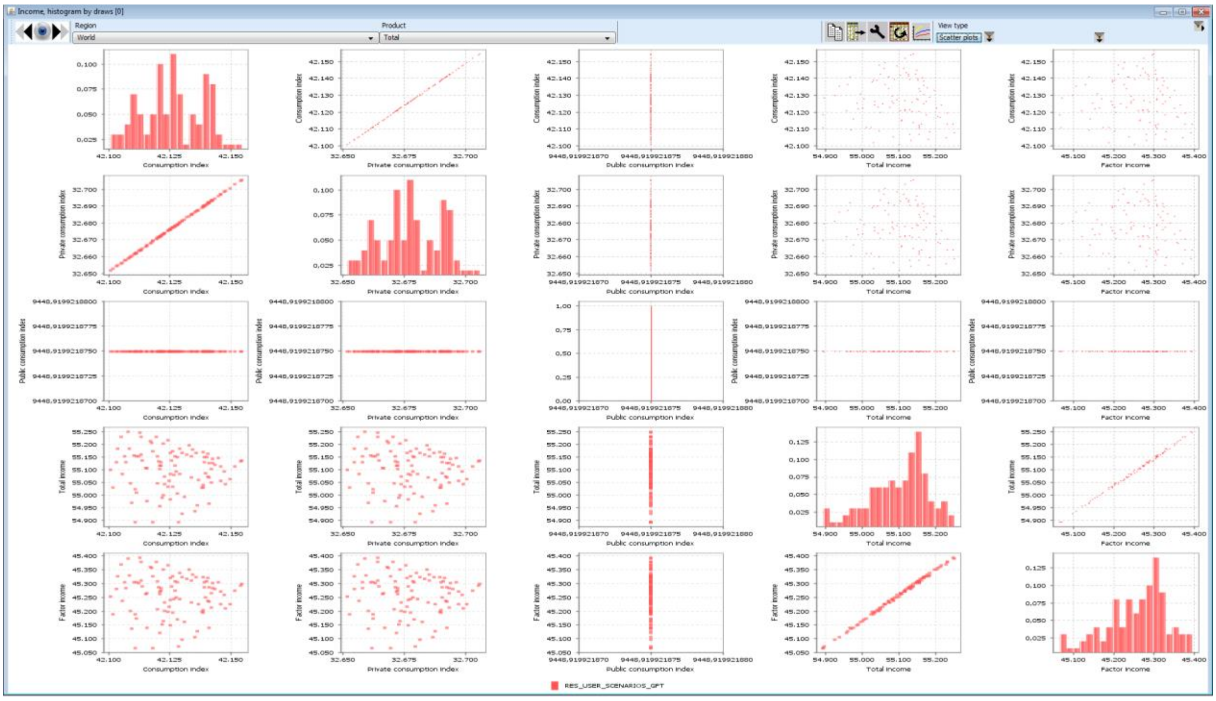
Further work is necessary to add views which automatically generated e.g. histograms. Currently, only one view with the core income results is defined:



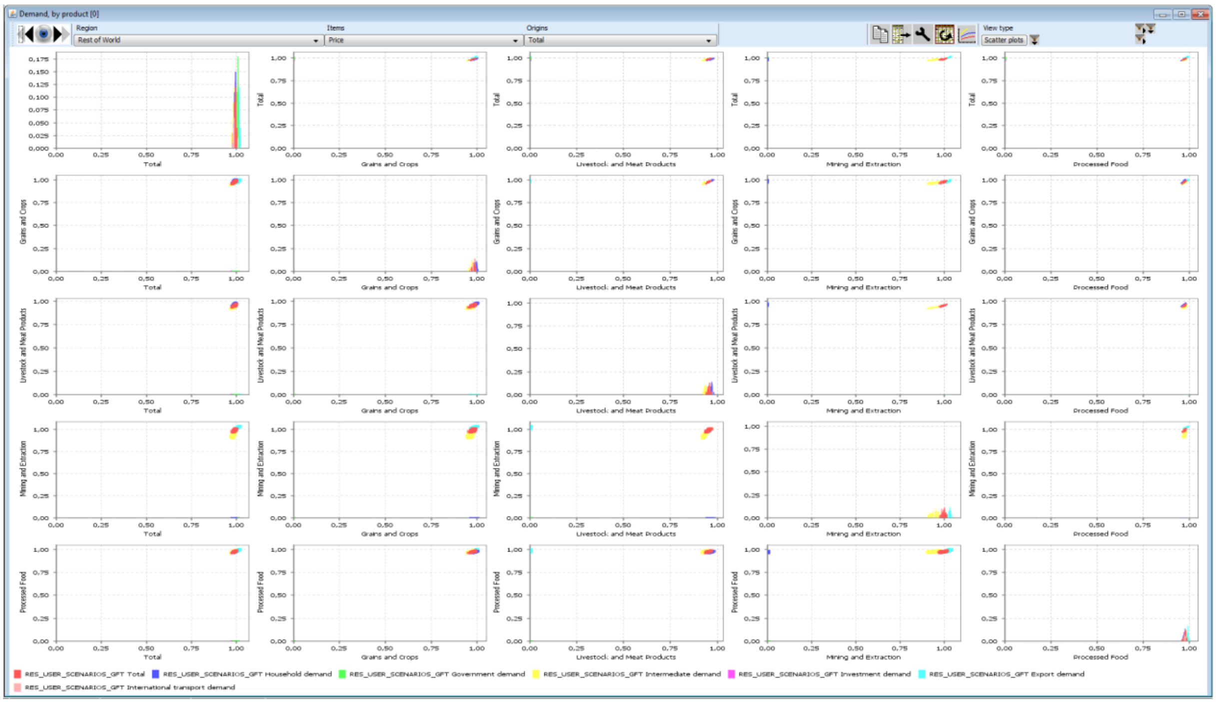


Note: If several scenarios are analyzed, the distributions for several scenarios will be shown together.

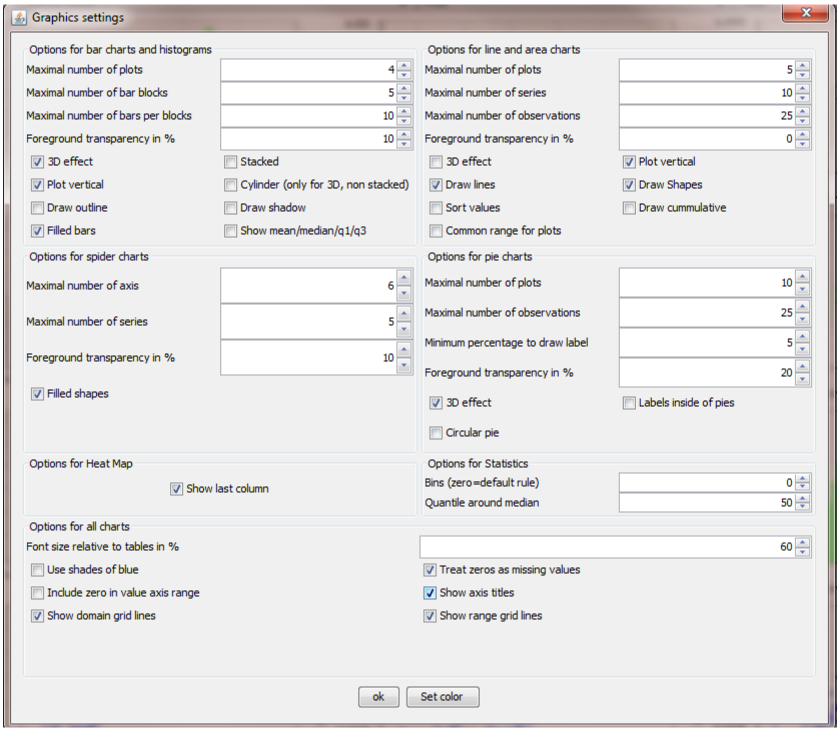
Alternatively, the "Scatter plot" type graphics can be used to plot additionally the correlation between the variables depicted:



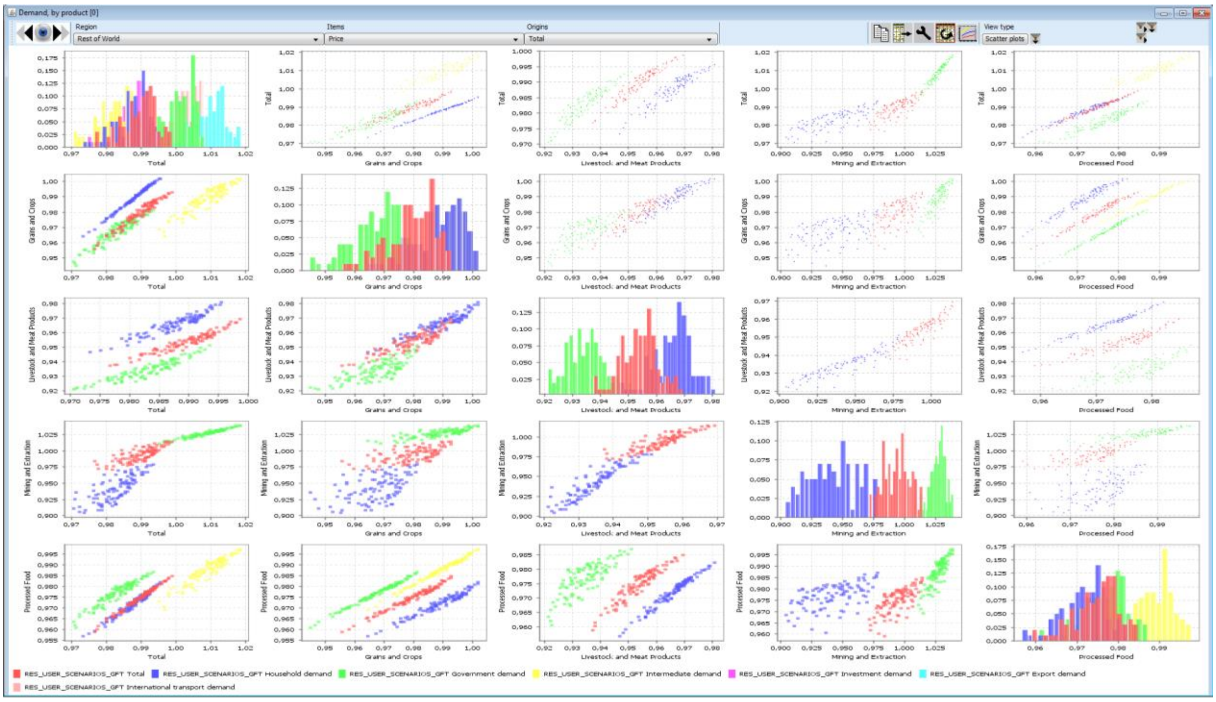
If the view type from existing tables is changed to "Scatter plots", the result might look as below (the example shows different types of prices in each plot, and different sectors in the rows/columns:



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:



Note: If the entropy estimator is used, the "mean" calculated in the GAMS code uses the probabilities as weights, but they are not reflected in the GUI when the in-built statistics are used.

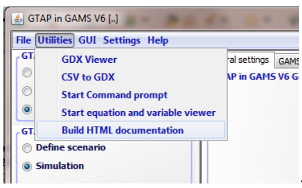
Getting an overview on the code

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.

HTML documentation

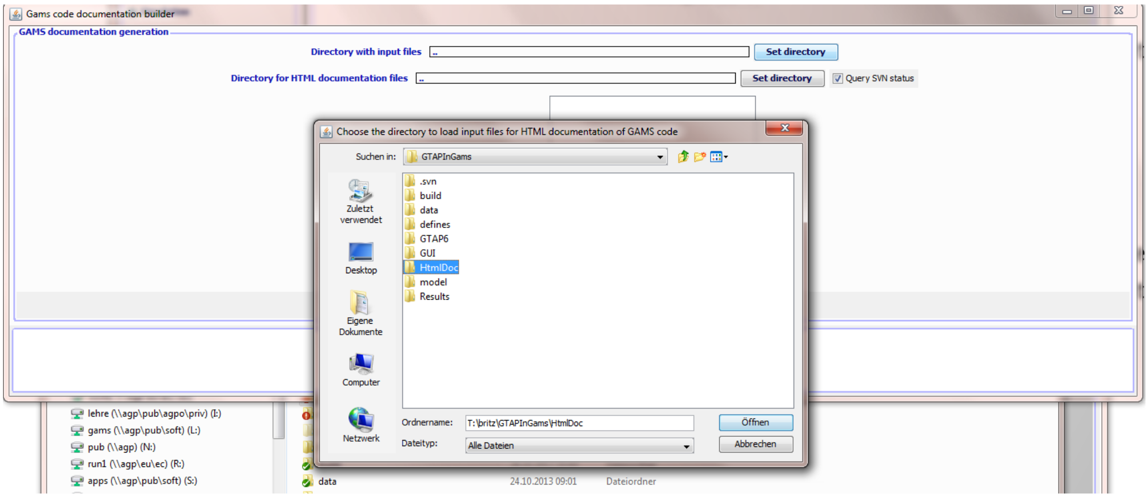
A HTML documentation of the code can be generated via "**Utilities/Build HTML documentation**":



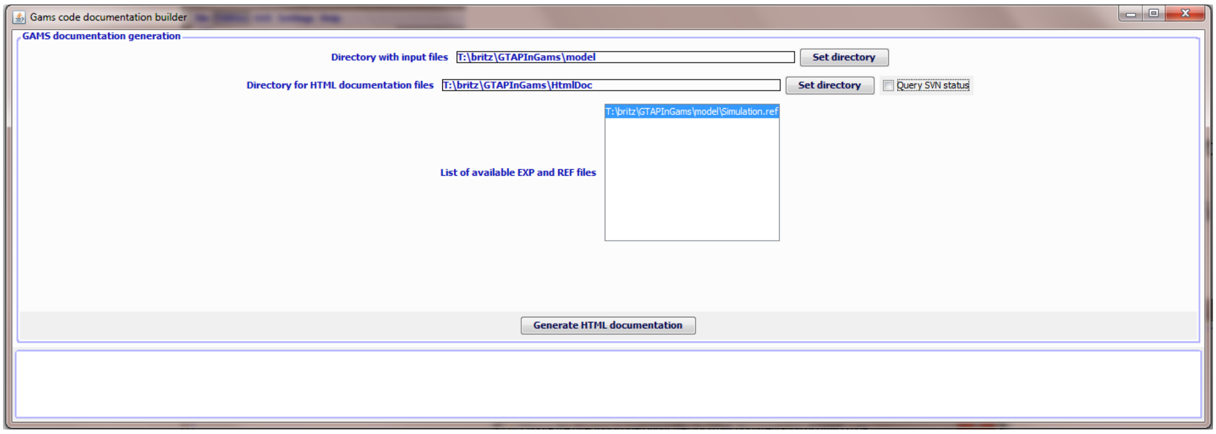
.

The documentation will include any changes you introduced to the project's GAMS code.

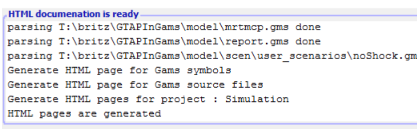
First choose the directory where the files should be stored (best use the "*HtmlDoc*" directory as shown below; beware: the utility will generate many files, do not use a directory where other files are already stored). The input files are found in the folder where GAMS was run, in the case of standard GTAP model in GAMS, typically in the "*model*" directory.



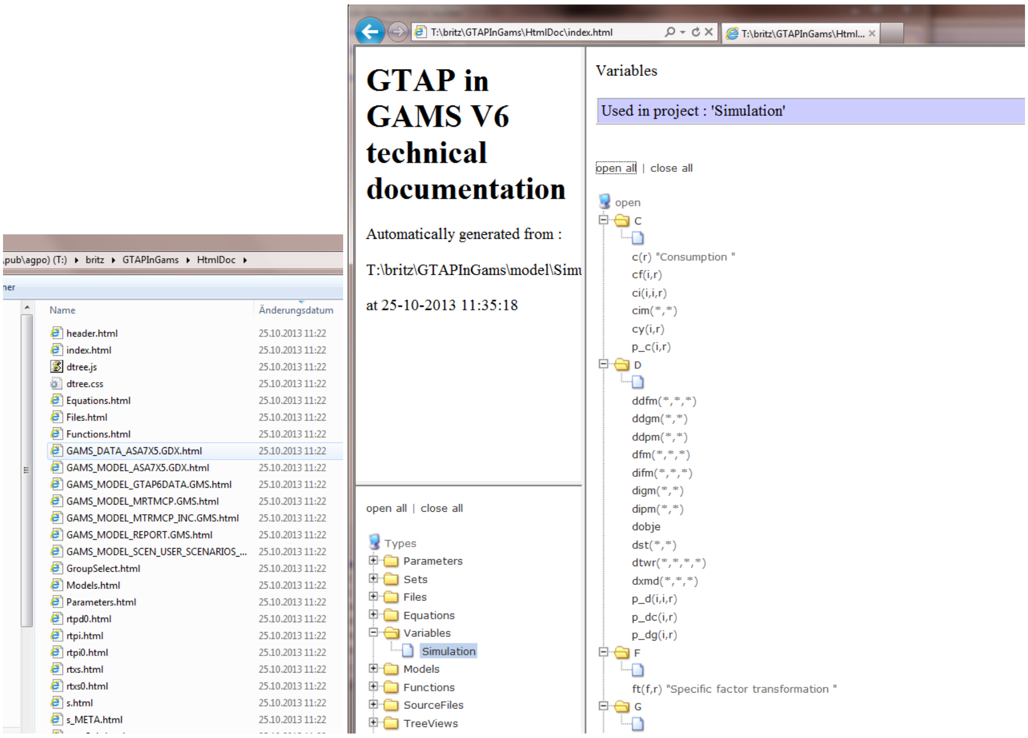
You should be able to find a file named “simulation.ref”. Make sure that “Query SVN Status” is switched off and next press the “Generate HTML documentation” button.



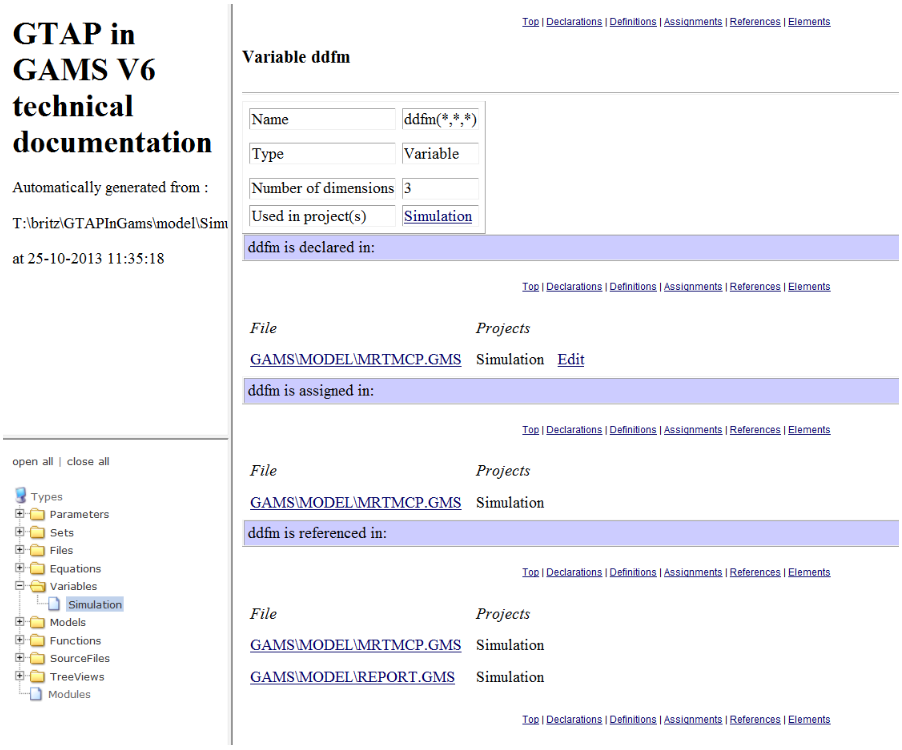
The program will work for a while and should end with “HTML documentation is ready” as shown below.



Afterwards, you find a in the chosen output directory a list of file, “*index.html*” is the starting point.

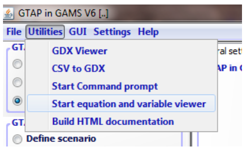


You can e.g. open all "Variables" used in the task "simulation", clicking with the mouse opens a symbol page:



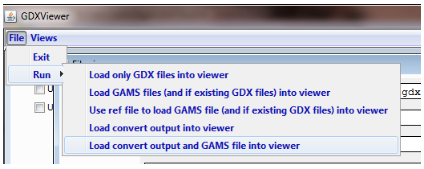
Equation and variable viewer (similar AnalyzeGE)

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.

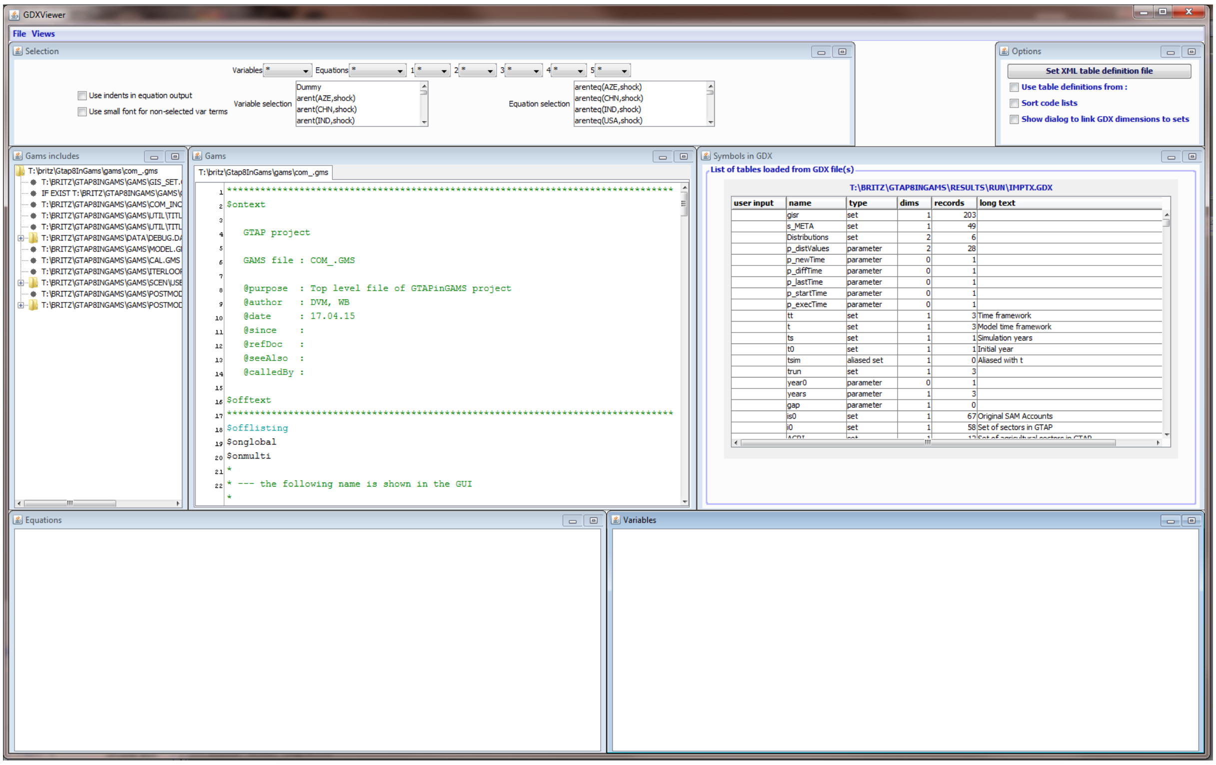


(note: one might want to change the first GDX files to the last experiment you have run)

Select “Load convert output and GAMS file into viewer” from the file menu:



You get now view like below:



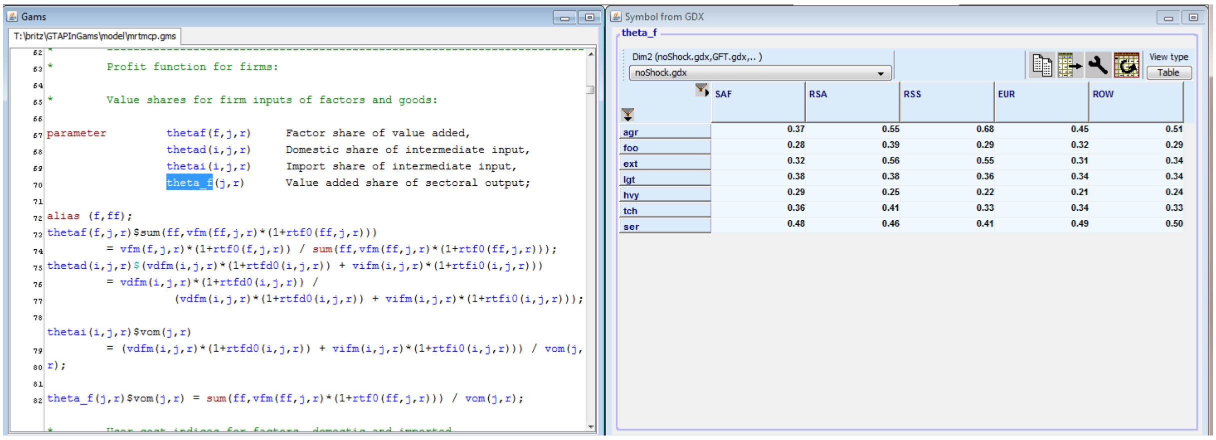
The viewer allows you to see:

- 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.

```
xdeq(SouthAsia,MeatLstk-c,ProcFood-a,shock) "Agents demand for domestic goods " ..

  -0.00902671001153436*(0.9999999486802*(0.995354810037336*(1.0000000513198*ps(SouthAsia,MeatLstk-c,shoc
  {-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).

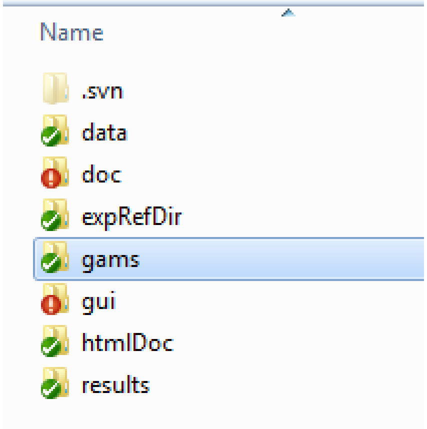


Technical documentation

Overview on directory structure

The screenshot below shows the directory structure of a "standard GTAP model in GAMS" 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.

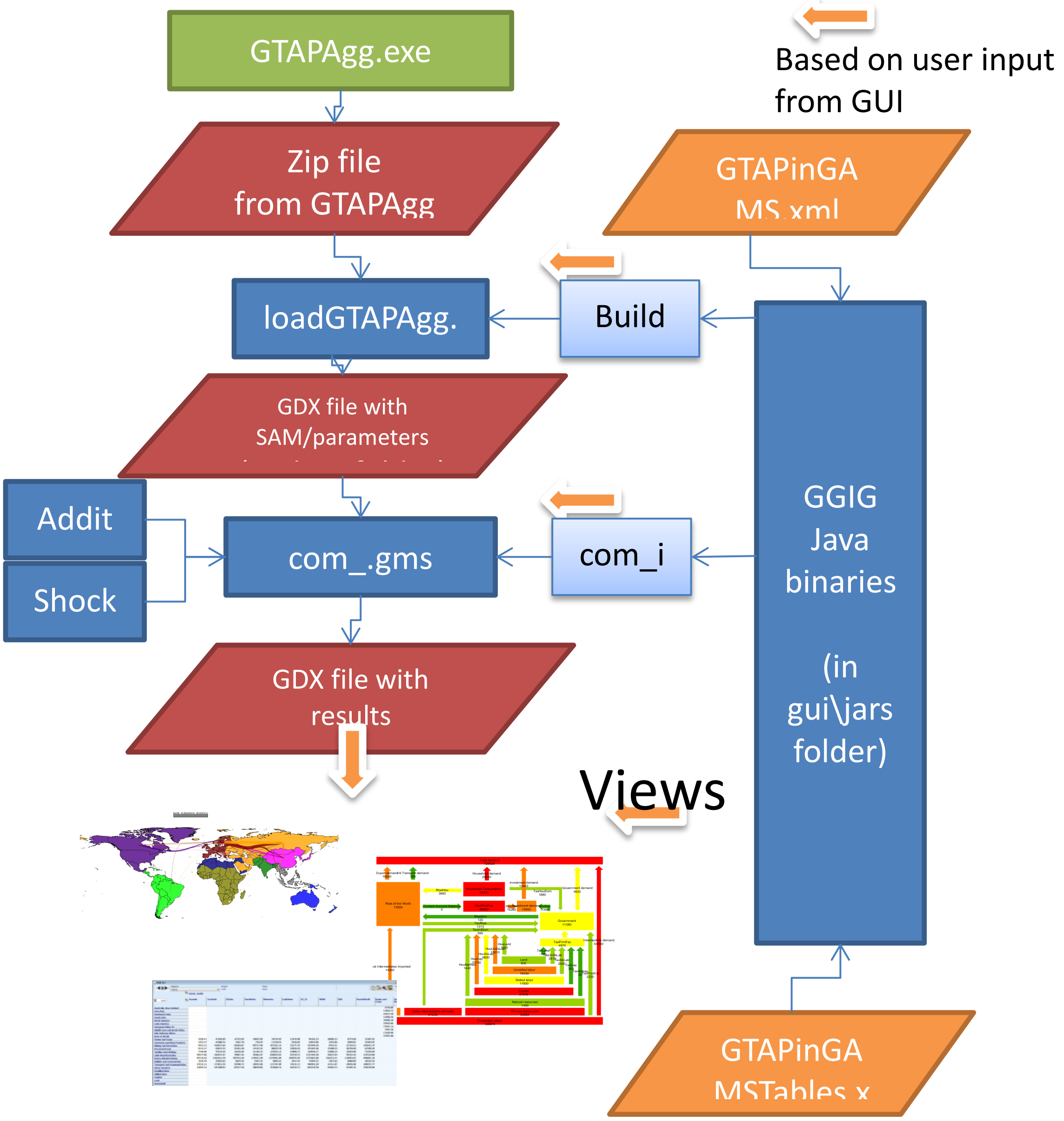


GUI overview

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). 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



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.

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>0</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(PQV) / Q,V,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,"V",i,"use",oris) / p_results(r,"Q",i,"use",oris);

   p_results(r,"T",i,"use",oris) $ p_results(r,"V",i,"use",oris)
      = p_results(r,"G",i,"use",oris) / p_results(r,"V",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 %modeltye% == 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.

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_padef(r,i,aa,t) $ a(aa)          ]  $ (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))\
                                     + pa(r,i,aa,t) $ ( not (alpham(r,i,aa,t)+alphad(r,i,aa,t)) )\
                                      }
```

The macro considers four different cases:

(1) Both domestic and import demand is present (*alpham* 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_padef* 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_padef* is the actual dual price aggregator:

```
*
* ---- definition of Armington dual price aggregator from domestic and import price
*

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

The domestic price macro *m_pdp* is defined as follows, i.e. is 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 inifinite CET for exports, use the supply
*       price ps, otherwise, used the domestic prices
*
    $$macro m_pdp(r,i,aa,t)  {  m_domPrice(r,i,aa,t) \
                           *(1 + dintx(r,i,aa,t) + itxshft(r,aa,t) $ iTaxShift(i))                  \
*
*             --- add emission taxes for domestic use
              $$ifi %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
*
    $$macro m_pmp(r,i,aa,t)  {pmt(r,i,t)*(1 + mintx(r,i,aa,t) + itxshft(r,aa,t)$ iTaxShift(i)) \
*
*             --- add emission taxes for domestic use
              $$ifi %ModulesCO2_Emissions%==on        + emisp(r,t)* (emii0(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_pefob* 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(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)]*lcu(rp,t) }
```

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 "*GtapInGamsTables.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,VUT,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>
```

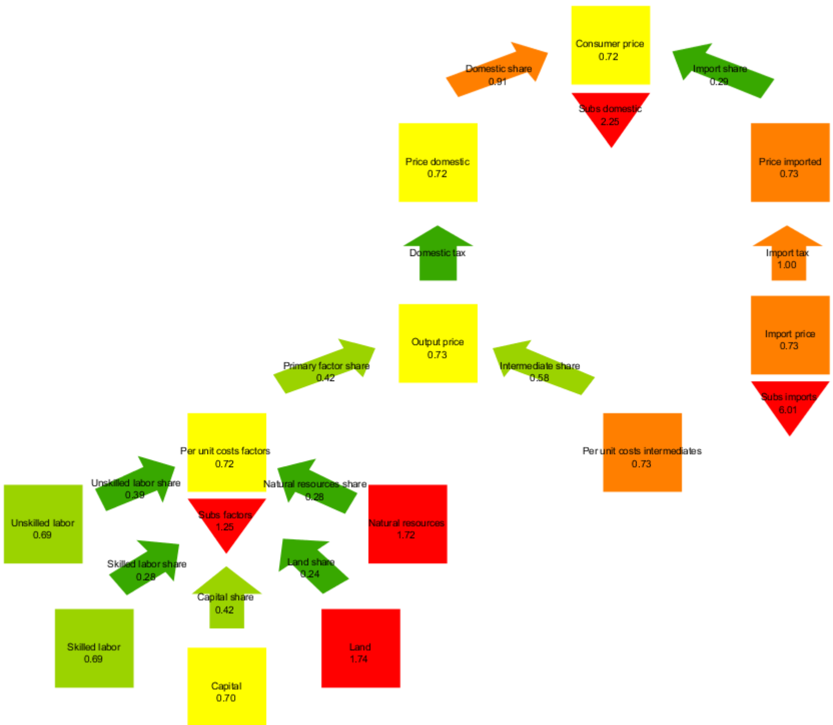
```
<table>
   <theme>Model overview</theme>
   <name>Model properties</name>
   <regionSel>Wor_REGEX</regionSel>
   <regionText>hide</regionText>
   <productSel>.*_REGEX</productSel>
   <productText>hide</productText>
   <activitySel>tots_REGEX</activitySel>
   <activityText>hide</activityText>
   <dim5Sel>Meta</dim5Sel>
   <dim5Text>hide</dim5Text>
   <hideEmptyRows>YES</hideEmptyRows>

   <dim6Text>hide</dim6Text>

   <defview>Table</defview>
   <defpivot>0P0S</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.

As part of the exploitation tool, the composition of the producer price (= per unit costs) and the consumer price is visualized as shown below:



That type of schemer is constructed based on registering hyperlinks to graphical symbols on a powerpoint slide and storing the presentation as html. A java programs reads the coordinates of the mouse regions which linked to the hyperlinks and stores them technically as if they would be coordinates for a geographic map. The hyperlinks registered in powerpoint must hence match the keys in a data dimension put in the row dimension of a tabular view, the following screenshots show a part of the view definition underlying the scheme visualize above and related GAMS statements:

```
<item>
    <key>pc</key>
    <itemName>Consumer price</itemName>
</item>
<item>
    <key>esubd</key>
    <itemName>Subs domestic</itemName>
</item>
<item>
    <key>pic</key>
    <itemName>Price imported</itemName>
</item>
<item>
    <key>pics</key>
    <itemName>Import share</itemName>
</item>
```

```
* --------------------------------------------------------------------------------
*
*   Store prices for price schemer
*
* --------------------------------------------------------------------------------

    p_results(r,"P",i,"if","tot")      = p_results(r,"P","int",i,"tot");
    p_results(r,"P",i,"cf","tot")      = cf.l(i,r);
    p_results(r,"P",i,"py","tot")      = p_results(r,"P","out",i,"dom");
    p_results(r,"P",i,"pc","tot")      = p_results(r,"P",i,"hou","tot");
```

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: Total — Years: shock — Percenta… Scenari… noShock

	ttip										noShock	
	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	All fir opera
World		0,00 0,11%	1,00 -0,01%	1,00 -0,00%	5217,96 0,00%	1637,32 0,00%	24002,60 0,00%	0,14 -0,00%	0,04 0,00%	0,66 -0,00%		
European Union 28	1,00 0,01%	0,00 0,41%	1,00 -0,02%	1,00 -0,01%	1654,72 -0,01%	457,21 -0,01%	7611,71 -0,01%	0,15 -0,02%	0,04 -0,02%	0,69 -0,02%	1,00	
Norway, Switzerland	1,00 -0,02%	0,03 0,05%	1,00 -0,01%	1,00 -0,01%	63,72 -0,03%	15,01 -0,06%	293,13 -0,03%	0,16 -0,01%	0,04 -0,03%	0,72 -0,01%	1,00	
United States of America	1,00 0,01%	0,00 0,71%	1,00 0,00%	1,00 0,02%	929,68 0,03%	315,31 0,06%	4276,51 0,03%	0,15 0,05%	0,05 0,07%	0,67 0,05%	1,00	
North America	1,00 0,01%	0,02 -0,04%	1,00 0,00%	1,00 0,01%	105,15 0,03%	35,58 -0,01%	483,70 0,03%	0,15 0,02%	0,05 -0,02%	0,68 0,02%	1,00	
Mercorsur	1,00 0,00%	0,01 0,01%	1,00 -0,01%	1,00 -0,01%	157,07 -0,01%	63,75 -0,01%	722,54 -0,01%	0,15 -0,01%	0,06 -0,02%	0,67 -0,01%	1,00	
China	1,00 0,00%	0,00 -0,01%	1,00 -0,01%	1,00 -0,00%	694,68 -0,00%	237,00 -0,00%	3195,53 -0,00%	0,13 -0,00%	0,04 -0,01%	0,59 -0,00%	1,00	
ASEAN 10	1,00 0,01%	0,01 -0,01%	1,00 -0,01%	1,00 0,00%	198,15 0,01%	54,63 -0,03%	911,50 0,01%	0,15 -0,00%	0,04 -0,04%	0,68 -0,00%	1,00	
OtherOECD	1,00 0,00%	0,00 -0,01%	1,00 -0,00%	1,00 -0,00%	694,27 -0,00%	231,10 -0,01%	3193,65 -0,00%	0,13 -0,01%	0,04 -0,01%	0,62 -0,01%	1,00	
EU Mediterrean Partners	1,00 -0,00%	0,01 0,04%	1,00 -0,01%	1,00 -0,01%	123,51 -0,01%	38,28 -0,03%	568,13 -0,01%	0,15 -0,01%	0,05 -0,02%	0,70 -0,01%	1,00	
Low Income	1,00 -0,00%	0,10 0,07%	1,00 -0,02%	1,00 -0,02%	19,82 -0,02%	5,97 -0,06%	91,17 -0,02%	0,16 -0,01%	0,05 -0,06%	0,72 -0,01%	1,00	
Rest of World	1,00 -0,00%	0,00 0,01%	1,00 -0,01%	1,00 -0,01%	577,18 -0,01%	183,48 -0,02%	2655,02 -0,01%	0,15 -0,01%	0,05 -0,02%	0,69 -0,01%	1,00	
World		0,00 0,11%	1,00 -0,01%	1,00 -0,00%	5217,96 0,00%	1637,32 0,00%	24002,60 0,00%	0,14 -0,00%	0,04 0,00%	0,66 -0,00%		
European Union 28	1,00 0,01%	0,00 0,41%	1,00 -0,02%	1,00 -0,01%	1654,72 -0,01%	457,21 -0,01%	7611,71 -0,01%	0,15 -0,02%	0,04 -0,02%	0,69 -0,02%	1,00	

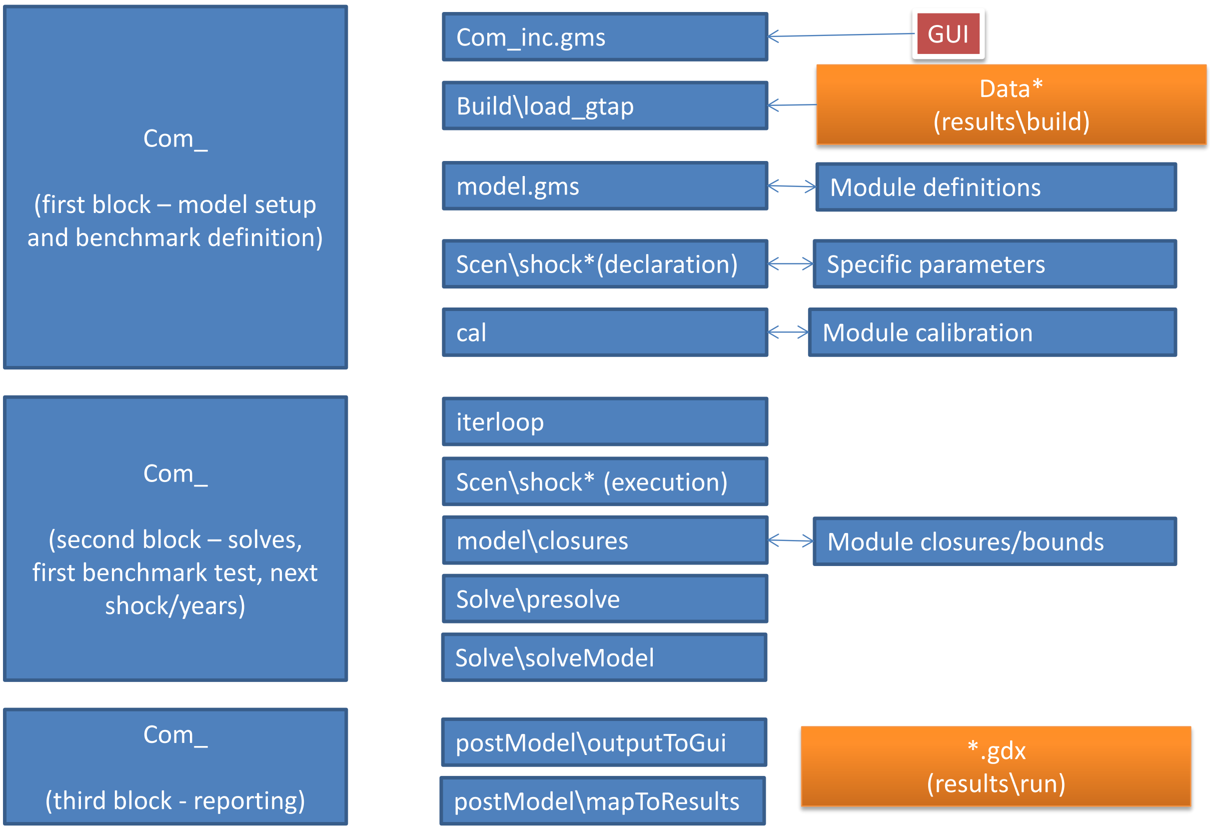
Melitz, trade matrix [0]

Items: Firm price | Products: Total | Years: shock | Percentage diff. to Scenarios noShock

	ttip Domestic sales	European Union 28	Norway, Switzerland	United States of America	North America	Mercorsur
World	1,00 -0,03%	1,00 0,02%	1,00 -0,01%	1,00 0,10%	1,00 0,01%	1,00 -0,01%
European Union 28	1,00 -0,06%	1,00 -0,02%	1,00 -0,01%	1,00 0,49%	1,00 0,02%	1,00 -0,01%
Norway, Switzerland	1,00 -0,01%	1,00 -0,03%	1,00 -0,00%	1,00 -0,00%	1,00 0,01%	1,00 -0,01%
United States of America	1,00 -0,04%	1,01 0,76%	1,00 -0,03%		1,00 0,01%	1,00 -0,02%
North America	1,00 0,01%	1,00 -0,03%	1,00 -0,03%	1,00 -0,01%		1,00 -0,02%
Mercorsur	1,00 -0,01%	1,00 -0,06%	1,00 -0,02%	1,00 -0,02%	1,00 0,02%	1,00 -0,01%
China	1,00 -0,01%	1,00 -0,02%	1,00 -0,02%	1,00 0,01%	1,00 0,01%	1,00 -0,01%
ASEAN 10	1,00 -0,00%	1,00 -0,03%	1,00 -0,02%	1,00 -0,02%	1,00 -0,00%	1,00 -0,02%
OtherOECD	1,00 -0,00%	1,00 -0,03%	1,00 -0,02%	1,00 0,00%	1,00 0,02%	1,00 -0,01%
EU Mediterrean Partners	1,00 -0,01%	1,00 -0,02%	1,00 -0,02%	1,00 0,01%	1,00 0,01%	1,00 -0,01%
Low Income	1,00 -0,01%	1,00 -0,11%	1,00 -0,02%	1,00 0,00%	1,00 0,02%	1,00 -0,01%
Rest of World	1,00 -0,01%	1,00 -0,03%	1,00 -0,02%	1,00 -0,00%	1,00 0,02%	1,00 -0,01%
World	1,00 -0,03%	1,00 0,02%	1,00 -0,01%	1,00 0,10%	1,00 0,01%	1,00 -0,01%
European Union 28	1,00 -0,06%	1,00 -0,02%	1,00 -0,01%	1,00 0,49%	1,00 0,02%	1,00 -0,01%
European Union 28	1,00 -0,06%	1,00 -0,02%	1,00 -0,01%	1,00 0,49%	1,00 0,02%	1,00 -0,01%

	China	ASEAN 10	OtherOECD	EU Mediterrean Partners	Low Income	Rest of World	noShock Domestic sales
World	1,00 -0,01%	1,00 0,00%	1,00 -0,00%	1,00 -0,01%	1,00 -0,01%	1,00 -0,01%	1,00
European Union 28	1,00 -0,00%	1,00 0,00%	1,00 0,00%	1,00 -0,01%	1,00 -0,01%	1,00 -0,01%	1,00
Norway, Switzerland	1,00 -0,00%	1,00 0,01%	1,00 0,00%	1,00 -0,00%	1,00 -0,01%	1,00 -0,00%	1,00
United States of America	1,00 -0,01%	1,00 -0,00%	1,00 -0,01%	1,00 -0,02%	1,00 -0,02%	1,00 -0,02%	1,00
North America	1,00 -0,01%	1,00 -0,00%	1,00 -0,01%	1,00 -0,02%	1,00 -0,02%	1,00 -0,01%	1,00
Mercorsur	1,00 -0,00%	1,00 -0,02%	1,00 -0,00%	1,00 -0,01%	1,00 0,02%	1,00 -0,01%	1,00
China		1,00 0,00%	1,00 -0,00%	1,00 -0,02%	1,00 -0,02%	1,00 -0,01%	1,00
ASEAN 10	1,00 -0,01%	1,00 -0,00%	1,00 -0,01%	1,00 -0,02%	1,00 -0,02%	1,00 -0,01%	1,00
OtherOECD	1,00 -0,01%	1,00 0,00%	1,00 -0,00%	1,00 -0,02%	1,00 -0,02%	1,00 -0,01%	1,00
EU Mediterrean Partners	1,00 -0,00%	1,00 0,00%	1,00 -0,00%	1,00 -0,01%	1,00 -0,01%	1,00 -0,01%	1,00
Low Income	1,00 0,00%	1,00 0,01%	1,00 -0,00%	1,00 -0,00%	1,00 -0,00%	1,00 -0,00%	1,00
Rest of World	1,00 -0,00%	1,00 0,00%	1,00 -0,00%	1,00 -0,01%	1,00 -0,01%	1,00 -0,01%	1,00
World	1,00 -0,01%	1,00 0,00%	1,00 -0,00%	1,00 -0,01%	1,00 -0,01%	1,00 -0,01%	1,00
European Union 28	1,00 -0,00%	1,00 0,00%	1,00 0,00%	1,00 -0,01%	1,00 -0,01%	1,00 -0,01%	1,00
European Union 28	1,00 -0,00%	1,00 0,00%	1,00 0,00%	1,00 -0,01%	1,00 -0,01%	1,00 -0,01%	1,00

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\loagGTAP_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 paremeter 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.

References

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

Britz, W. (2010a): GAMS Graphical Interface Generator User Guide. Institute for Food and Resource Economics, University Bonn. In: http://www.ilr.uni-bonn.de/agpo/staff/britz/GGIG_user_Guide.pdf.

Britz, W. (2010b): 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

Britz, W. (2014): A New Graphical User Interface Generator for Economic Models and its Comparison to Existing Approaches, German Journal for Agricultural Eonomics, 63(4), forthcoming

Britz, W., and 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

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

Britz, W., Drud A. and van der Mensbrugghe, D. (2015): Reducing unwanted consequences of aggregation in large-scale economic models - a systematic empirical evaluation with the GTAP model, Discussion Paper of the Institute for Food and Resource Economics 2015:4, in: http://www.ilr.uni-bonn.de/agpo/publ/dispap/download/dispap15_04.pdf

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 Soving Real World Optimization Problems: 199-212, Springer, Heidelberg, Germany

Domínguez, I. P., Britz, W., & Gopalakrishnan, B. N. Post-model Analysis in large-scale models: the examples of Aglink-Cosimo, CAPRI and GTAP, Paper presented at 2012 GTAP conference.

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

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