The regional CGEs of CAPRI – RegCgeEU+ Model documentation including Graphical User Interface

Wolfgang Britz, University Bonn

Version Feburary 2014

Inhalt

Introduction

Overview on the model

RegCegEU+ is a set of regionalized, single country, deterministic, comparative-static CGE models which cover the EU Member Countries, Turkey and Western Balkans. Each country model depicts an open economy, import and export prices are driven by behavioral functions.

Supply is represented by nested Constant Elasticity of Subsitution (CES) functions: substitution between value added and total intermediate input use; substitution between primary factors; Leontief composition of intermediate input use; substitution at the level of sector-by-sector intermediate input demands between regional, national and international origin.

Final household demand is based on Linear Expenditure System (LES), distribution between final consumption and saving is based on a Cobb-Douglas (CD) type utility function. Households own primary factor from which they draw revenues, receive subsidies from regional government and net borrow from the rest-of-the-world, while paying regional and national income tax on primary factor earning.

The *regional government* collects regional income taxes and receives a share of other national tax revenues. Their budget, after subsidies to households are paid, is distributed between savings and final demand according to a CD type utility function, final demand of the government is based on a LES.

The *national government* collects all other taxes (national income tax, sales taxes, investment taxes, production taxes, taxes on primary factor use by industries), which can also be negative and thus represent subsidies and net borrows with the rest-of-the world. The benchmark revenue is distributed to the regions according to given historical shares, any changes are distributed according to population size of the region.

Total *investment demand* in value is equal to regional savings. Investment demand for specific sectors is firstly based on fixed quantities which represent a share of investment demand in the benchmark situation; the remaining savings are then distributed by fixed shares.

Final demand is sourced from regional, national and international origins according to a CES function (*Armington* assumption), supply is distributed to these destinations based on a Constant Elasticity of Transformation (CET) function.

The model allows *different market structure for primary factors*: factors can be modeled completely immobile, sluggish based on a CET or fully mobile. Supply of labor can be represented by a wage curve, equally, labor prices can be fixed (sticky wages). The total capital stock in the economy can be either fixed, depreciated and update with new investment or the so-called DPSV-rule can be used.

Further model features include *net migration functions*, and different closures for the household, government and trade accounts.

The model comprises a component for projecting the SAM given a agricultural baseline and some macro drivers (GDP, population). The model is realized in GAMS and steered via a Graphical User Interface. The model can be used in stand-along modus or integrated in the CAPRI modeling system.

Background of the development of RegCgeEU+ and a GAMS version

In the CAPRI-RD project (2009-2013), the team of Hannu Törmä from Finland was responsible for the development of a CGE template suitable to model rural development (RD) measures, drawing on the REGFIN model for Finland. In the deliverables D3.2.3 and D.3.3.3 of the CAPRI-RD project, that template and technical details about its implementation in GEMPACK are discussed in detail. A running GEMPACK version was developed, and already early in the project a technical solution to link GAMS and GEMPACK based models was designed and successfully tested in a link between CAPRI and REGFIN. GEMPACK is an established algebraic modeling language (AML) specialized in CGE modeling with a matching graphical user interface which facilitates many standard tasks related to CGE modeling. Equally, GEMPACK is a package with a long history and a large user community.

AMLs such as GEMPACK and GAMS often use similar language concepts, in the specific case, both are based on a set driven concept which avoids to a larger extent using explicit loops. But differences between the languages are nevertheless many. The maintenance of a rather elaborated template such as regCgeEU+ for its application to almost 300 regions and a complex policy such as the second pillar of the EU Common Agricultural Policy asks clearly for an experienced GEMPACK modeler. More so, ensuring that the GEMPACK model interacts correctly with GAMS based modules in the rest of the CAPRI requires specialized technical expertise in both modeling languages, including knowledge about programs dealing with format conversion. Any serious changes to RegCgeEU+ require a GEMPACK license (the compiled standard template can however be used without a license).

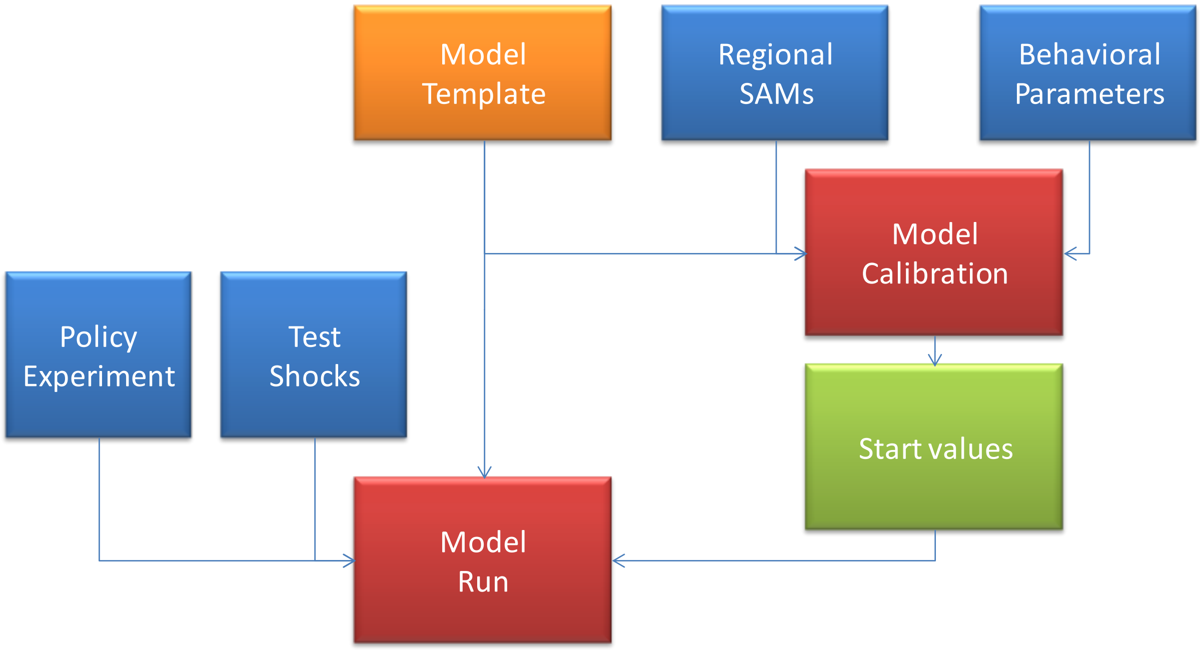
From a CAPRI system maintenance and development perspective, a GAMS based implementation of the regional CGEs clearly has a number of advantages. Most importantly, no format conversions and software changes are necessary to let the regional CGEs interact with the regional programming and other parts of the GAMS based CAPRI code. CAPRI developers with experience in GAMS stand a good chance to understand the GAMS code dealing with the regional CGEs. Changes in a GAMS based model template do not require a GEMPACK license or knowledge about GEMPACK. Existing features of the CAPRI GUI such as generating HTML based documentation of the GAMS code can be used also for the regional CGE layer. The coding standard developed in CAPRI-RD to support a modular structure and to decrease learning and maintenance costs can be directly used for the implementation of the template.

Using GAMS for the CGEs comes however also with clear disadvantages. Firstly, GEMPACK includes a highly specialized solver for CGEs, and GEMPACK is specifically suitable to deal with models which are written in percentage form, probably mixed with equations/variables in levels or differences. Implementing these features in GAMS is much more cumbersome, most CGE templates realized in GAMS prefer therefore equations written in levels. The percentage change presentation in GEMPACK eases e.g. also the decomposition of effects. Secondly, the team responsible for the CGE layer in CAPRI-RD is more experienced in GEMPACK than in GAMS and no funds had been specifically allocated in the work plan for a GAMS based version, so that the actual conversion to GAMS was tackled by the team in Bonn.

Despite obvious disadvantages of a GAMS based solution and the fact that the work program of the project did not foresee funds for coding in two languages, especially the project coordinator of CAPRI-RD promoted at least a parallel implementation of the CGEs in GAMS. The arguments above for a GAMS based implementation seemed to be serious enough to at least do first steps towards a GAMS based solution. The main aim was to get a more informed view about the costs of a GAMS conversion and the overall pro and cons of both software solutions. The decision to do so was based on the assumptions that firstly previous developments of CGE templates in GAMS should allow for a rather fast conversion to GAMS. Secondly, past work to link models implemented in GAMS to Java based interface generators and the Java based exploitation tools of CAPRI should speed up the interface development. The actual development process of a working template took indeed only a few working days.

Once the GAMS version was developed and operational, its concept and structure deviated from the GEMPACK template.

File structure



Graphic 1: General overview on structure

The file structure consists of the following elements and is thought to support a clear separation of declarations and code performing data transformations or model runs:

1. **regcge.gms** provides a stand-alone environment detached from the CAPRI simulation engine to calibrate the models, run test shocks or actual scenarios, both in single or multi-thread parallel execution mode.
2. **regcge\regce_templ.gms** represent the *model template*, i.e. it defines the equations, variables and parameters used in the model template; the details are documented below. The underlying sets etc. are defined in **regce\regcge_decl.gms**.
3. **regcge\regce_ini.gms** performs the *model calibration*, initializes the parameters used in the template such as tax rates, substitution elasticities, marginal budget shares etc. as far as possible from the given national and regional Social Accounting Matrices (SAMs). It defines from the SAMs starting values for the variables in the templates and test if the resulting variables and parameters provide together a calibrating model. It also comprises currently still code to rebalance the SAM.

4. **Regcge\regcge_testShocks.gms** defines a series of test shocks such as changes in total factor productivity or factor endowment to test the simulation behavior of the model.
5. **Regcge\regcge_runsim.gms** runs a single policy experiment.

The matching input data, currently only the SAMS, are stored in "dat\regcge". Missing are currently data on unemployment at regional level. There a few other smaller programs discussed below.

Initialization of parameters and starting values for variables

The philosophy in CAPRI is to separate the task of model calibration from counterfactual runs, and if possible, to also separate the code for doing so. The template supports that idea: counterfactual runs load behavioral both parameters and start values generated by a separate calibration step. In order to do so, two approaches are chosen:

1. *Behavioral parameters* such as shift and share parameters of CES/CET nests are stored in a GDX parameter from which they are loaded when a counterfactual run is started:

```
*
 execute_load "%resdir%\..\simini\ini.gdx"
                              p_cgeRes,
                              p_rhoS,p_RhoC,p_rhoT,
                              pv_commit,pv_margBudShare,pv_sigma,pv_delta,pv_gamma,
                              p_margBudShareGov,pv_savRateGov,
                              pv_shiftParArm,pv_shiftParCet,pv_shiftParCes
                              p_iCoeffR,p_iCoeffN,p_iCoeffI,
                              p_popSize,p_primFacEnd,p_invShare,p_capStock,
                              p_primFacTax,p_taxProd,p_salesTaxRate,
                              p_taxIncLocRate,p_taxIncStaRate,p_invTaxRate,
                              p_cnstNetMigr,p_slopeNetMigr,p_labPerPop,
                              p_savRateHou,p_deprRate,p_prudInvFactor,p_netRateReturnCapCorr
                              p_cnstWageCurve,p_elasWageCurve;
```

2. GAMS code stores all *endogenous variables* in a multi-dimensional GAMS parameter (regcge\regcge_reg.gms). That parameter *p_cgeRes* from the calibration step, loaded as seen above at run time, is used (regcge\set_start_point.gms) to initialize the variables for a counterfactual run. The counterfactual run adds its results to that parameter and saves at it the end of the run to disk:

```
*
 execute_unload '%gdxFileName%' p_cgeRes=p_res,meta;
```

These results can be loaded in the exploitation tools or in a GDX viewer (see also last chapter on the graphical user interface).

Parallel execution

The first possibility to solve the models is standard GAMS mode, i.e. the country models are solved one after another in the same GAMS process. Neither CONOPT nor PATH as the two recommended solvers currently support parallel execution, i.e. they only use one CPU. Accordingly, on most modern computers some cores will be idling if only a GAMS process is running on the machine. As in other parts of the CAPRI GAMS code, it looks hence inviting to allow as a second options to spawn several GAMS processes which execute simultaneously and solve the models in parallel.

In order to do so, the programs firstly determines in which mode it is running:

```
* ------------------------------------------------------------------------------
*
*   Determine Simulation mode
*
* ------------------------------------------------------------------------------
*
$iftheni.child_thread %threads%==true
*
$setglobal mode child
*
$else.child_thread
*
$iftheni %use_Threads%==true
*
$setglobal mode mother
*
$else
*
$setglobal mode single
*
$endif
*
$endif.child_thread
```

In the modes “child” or “single”, the program will run a simulation or a range of predetermined test shocks. Whereas as the “mother”, it will start the child processes. The following snapshot of the code shows how that is accomplished:

```
  LOOP(cgeMSIncl,
*
*    --- generate a scratch directory for each country
*        (if it exists, try to remove it so that we start if possible with an empty one)
*
     put_utility 'shell' / "if     exist %scrdir%\gen_"cgeMSIncl.tl:2" rmdir /S /Q %scrdir%\gen_"cgeMsIncl.tl:2;
     put_utility 'shell' / "if not exist %scrdir%\gen_"cgeMSIncl.tl:2" mkdir %scrdir%\gen_"cgeMsIncl.tl:2;
*
*    --- generate a flag file which works as an indicator that the child process
*        is running (exists) or has terminated (flag file is deleted at end of run)
*
     put_utility batch 'shell' / "echo test > %scrdir%\resCge_"cgeMsIncl.tl".flag";
*
*    --- spawn the chil process
*
     put_utility batch 'shell' / 'start /B /NORMAL %GAMSPATH%\GAMS.EXE' ' %CURDIR%\regcge.gms'
*
*        (1) Work as a child (threads=on)
       ' --threads=on',
*
*        (2) Use the current include file from the interface (scen=%scen%)
       ' --scen=%scen%',

*        (3) Work for current country
       ' --myMS='cgeMsIncl.tl,
*
*        (4) Generate a listing file under the name of the country in the scratch directory
       ' -o=%scrdir%\gen_'cgeMSIncl.tl:2'.lst'
*
*        (5) generate a result file carrying the countries label in the scratch directory
       ' --gdxFileName=%scrdir%\gen_'cgeMSIncl.tl:2'\res.gdx';
  );
```

After that point, if there are no compilation errors, each country will be executed in its own GAMS process which runs the test shocks respectively a policy experiment. The mother process needs to know when the last model is solved. As seen above, the code generates for each country a flag file. That flag file is deleted at finalization of the child process.

A mechanism is hence needed to let the mother process wait with further action until the last flag file is removed. That is achieved by a small Windows script which is also used in other parts of CAPRI:

```
*
*   --- wait for 5 minutes for the child processes
*       (should be sufficient for 25 shocks even with a large country)
*
$batinclude 'util\title.gms' "'Waiting for thread models'"
```

Finally, the results from the individual processes need to be loaded and merged together which is done by the following code:

```
*
* --- and collect the solutions
*
  loop(cgeMSIncl,
*
*   --- set dynamically the gdx file name to load
*
    put_utilities batch 'gdxin' / '%scrdir%\gen_' cgeMSIncl.tl:2'\res.gdx';
    execute_loadpoint p_cgeRes=p_res,meta,p_shockTest;

    if ( execerror > 0,
        option kill=cgeMsRun;cgeMsRun(cgeMsIncl) = YES;
        abort "%CURDIR%\regcge.gms did not write a result file for ",cgeMSRun,
          "check listing: %scrdir%\gen_",cgeMSRun,".lst";
    );
  );
```

After that point, *p_cgeReg* will comprise the results for all countries.

Template

The idea of the GAMS based implementation consists in translating as far as possible the existing GEMPACK template to a GAMS based one. That template should fulfill two major requirements. Firstly, CAPRI coding standards following D7.1 of CAPRI-RD, e.g. regarding naming conventions of GAMS symbols and file structure should be followed. As a consequence, the GAMS symbols do not carry the same name as in the GEMPACK template. Secondly, the template should be comparable to other existing CGE templates in GAMS to decrease learning costs. Specifically, to the extent possible, all equations and variable are written in levels as usually done in a GAMS template. Standard GAMS code is used and not MPSCGE.

Generally, all quantity variables are expressed in constant Mio Euro 2005 and thus reflect the original SAMs entries. If possible, prices in the starting point are equal to unity.

The template is written such it can be solved for individual countries separately, but also for several countries simultaneously opening up the possibility to link it to some trade presentation.

Production system

The underlying SAM comprises a symmetric I/O table where each industry produces one matching output. The production function chosen is rather standard:

- A **top level Leontief nes**t links the **value-added aggregate** and **intermediate demand**.

- **Intermediate demand is Leontief**, consisting of three CES linked sub-matrices (regional, rest-of-nation, rest-world).[1]
- The **value added nest is a CES aggregate of capital, labor and land**. Land is only present in agriculture and forestry.

Figure 1 illustrates the production side nests. Note that in the equations, the scale was chosen such that the quantity of value added v_quantVA exactly equals the quantity of output v_quantProd, so that the former could be substituted out of the system.

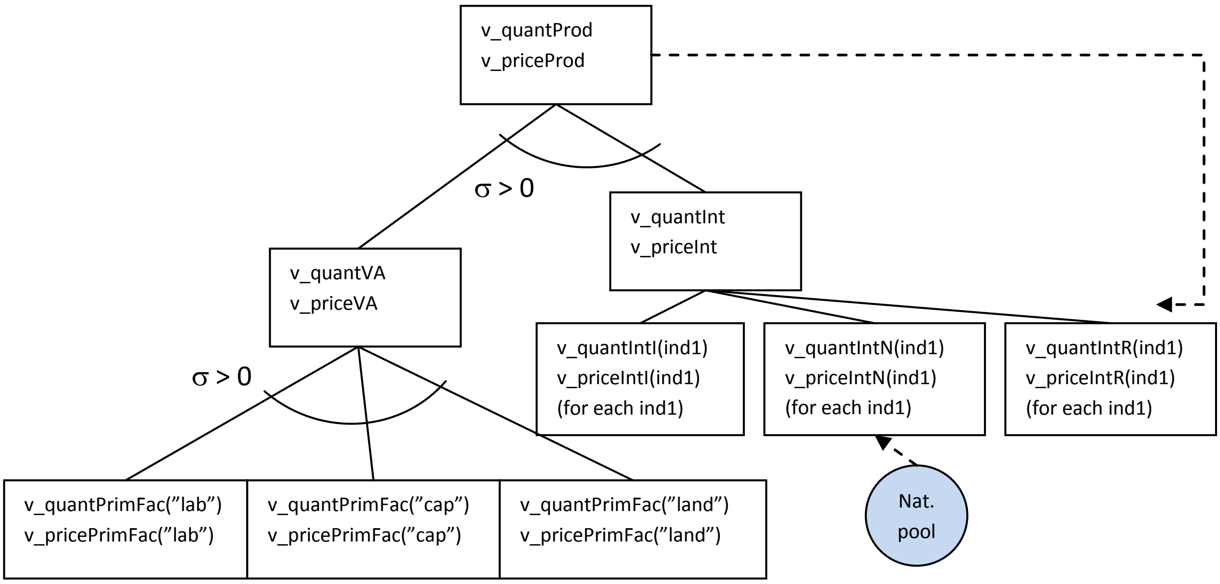


Figure 1: Production side nests and associated variable names

The primal cost minimization problem of producer for a given production level q and input prices w, using input quantities x is described by the following equations, assuming a CES technology:

$$\min \sum_i x_i w_i \quad s.t.\ \ \alpha\left(\sum_i \sigma_i^{\rho-1} x_i^{\rho}\right)^{\frac{1}{\rho}} \geq q$$

The Lagrangean function becomes

$$\mathcal{L}(x_1, \dots, x_i, \dots x_{n,}\pi) = \sum_i x_i w_i - \pi\left(\alpha\left(\sum_i \sigma_i^{\rho-1} x_i^{\rho}\right)^{\frac{1}{\rho}} - q\right)$$

The derivatives of the Lagrangean (first order conditions), rewritten for later convenience, and with π denoting the dual value of the constraint are:

$$x_i = \alpha\left(\sum_i \sigma_i^{\rho-1} x_i^{\rho}\right)^{\frac{1}{\rho}} \left(\alpha^{\rho} \sigma_i^{\rho-1} \frac{\pi}{w_i}\right)^{\frac{1}{1-\rho}}$$

[1] The reader should note that the solution to have explicit input coefficient for imports from rest-of-the-nation and rest-of-the-world differs from the original templates developed by the Finnish team and documented in the CAPRI RD- deliverables D3.2.3 and D3.2.3

$$q \le \alpha \left(\sum_i \sigma_i^{\rho-1} x_i^{\rho} \right)^{\frac{1}{\rho}}$$

Assuming that the restriction is always binding, we can insert the restriction into the first equation of the first order conditions. Then the dual value π still need to be defined. Observing that π is in fact the output price (thus renamed from π to p), we add the zero profit condition for a constant returns to scale industry, which is "value exhaustion", to obtain the following two behavioural model equations:

$$x_i = q\sigma_i \left(\alpha^{\rho} \frac{p}{w_i} \right)^{\frac{1}{1-\rho}} \quad \text{for each i = 1...n}$$

$$p = \sum_i \alpha {w_i}^{\rho}$$

Those are the equations, which in their percentage-change forms should be familiar to CGE-modelers, that we need to implement the CES-nest of value added in the code (note that it would be more convenient to substitute the substitution elasticity σ for ρ). The former represents the conditional factor demand equations, the latter are called value-exhaustion conditions.

Substitution between value added and intermediates

The *top nest* defines at the on hand the demand for the primary factor aggregate[2]:

```
*
 e_quantVa(cgeRegRun,ind) $ (v_quantVa.up(cgeRegRun,ind) ne 0) ..

    {
      v_quantProd(cgeRegRun,ind)
        /v_quantVa.scale(cgeRegRun,ind)

*   With this CES formulation, factor demand is: A share of output ...
    *  pv_shareParCES(cgeRegRun,ind,"va")

*   ... times a total productivity factor exponent substitution elasticity minus 1
    *   pv_shiftParCES(cgeRegRun,ind)**(p_rhos(cgeRegRun,ind)-1)

*   ... times ratio of output price to factor price plus tax, to the power of the substitution elasticity
    *   [  v_priceProd(cgeRegRun,ind)
         / (v_priceVa(cgeRegRun,ind)*(1 + p_prodTaxRate(cgeRegRun,ind)))
        ]**(p_rhos(cgeRegRun,ind))

    }

    =E= (v_quantVa(cgeRegRun,ind))/v_quantVa.scale(cgeRegRun,ind);
```

At the other hand, it drives the total demand for intermediates:

[2] The reader should note that almost any variable in the system is scaled (division by .scale) to ease scaling for the solving and interpretation of possible infeasibilities which are due to the scaling in relative terms.

```
*
 e_quantInt(cgeRegRun,ind) $ (v_quantInt.up(cgeRegRun,ind) ne 0) ..

    {
      v_quantProd(cgeRegRun,ind)
            /v_quantInt.scale(cgeRegRun,ind)

*   With this CES formulation, factor demand is: A share of output ...
    *  pv_shareParCES(cgeRegRun,ind,"int")

*   ... times a total productivity factor exponent substitution elasticity minus 1
    *   pv_shiftParCES(cgeRegRun,ind)**(p_rhos(cgeRegRun,ind)-1)

*   ... times ratio of output price to factor price plus tax, to the power of the substitution elasticity
    *   [  v_priceProd(cgeRegRun,ind)
         / (v_priceInt(cgeRegRun,ind)*(1 + p_prodTaxRate(cgeRegRun,ind)))
        ]**(p_rhos(cgeRegRun,ind))
    }

    =E= (v_quantInt(cgeRegRun,ind))/v_quantInt.scale(cgeRegRun,ind);
```

The related price index is defined as:

```
  e_unitCost(cgeRegRun,ind) $ (v_quantProd.up(cgeRegRun,ind) ne 0) ..
*
*    --- production value
*
     v_unitCost(cgeRegRun,ind) =E=

       1 /  pv_shiftParCES(cgeRegRun,ind) *

       [
*
*          --- value added price times CES aggregate of primary factors
*
          (pv_shareParCES(cgeRegRun,ind,"va")
              * (v_priceVa(cgeRegRun,ind)*(1+v_slackCge(cgeRegRun,ind,"va")))
                   ** (1-p_rhos(cgeRegRun,ind))) $ pv_shareParCES.l(cgeRegRun,ind,"va")
*
*          --- cost of intermediates times CES aggregate of intermediate composition
*              from regional / national / imported
*
        + (pv_shareParCES(cgeRegRun,ind,"int")
             * (v_priceInt(cgeRegRun,ind)*(1+v_slackCge(cgeRegRun,ind,"intUse")))
                   ** (1-p_rhos(cgeRegRun,ind)) ) $ pv_shareParCES.l(cgeRegRun,ind,"int")
*
       ] ** ( 1/(1-p_rhos(cgeRegRun,ind)));
```

The *v_slackCge* variable are present infeasibilities and should not exceed tiny thresholds (say 0.0001) in a final solution. They are only introduced in the model if the solver cannot find a feasible solution to the complete system, to ease analysis and interpretation of the solution.

Value added nest: substitution between primary factors

The demand for the individual primary factors capital, labor and land is described by the following equations:

```
 *
 *    --- nest inside the VA: substitution between primary factors
 *
 e_primFacUse(cgeRegRun,ind,primFac) $ [v_primFacUse.up(cgeRegRun,ind,primFac) GT 0] ..

*   With this CES formulation, factor demand is: A share of output ...

    v_quantVa(cgeRegRun,ind)/v_primFacUse.scale(cgeRegRun,ind,primFac)

    *  pv_shareParCES(cgeRegRun,ind,primFac)

*   ... times ratio of output price to factor price plus tax, to the power of the substitution elasticity
    *   [  v_priceVA(cgeRegRun,ind)
         /(v_pricePrimFacInd(cgeRegRun,ind,primFac)*(1+pv_primFacTaxRate(cgeRegRun,ind,primFac)))
        ]**(p_rhoVa(cgeRegRun,ind))

    =E= (v_primFacUse(cgeRegRun,ind,primFac))
                            /v_primFacUse.scale(cgeRegRun,ind,primFac);
 *
```

The related price index is defined as:

```
*
  e_vaExhaust(cgeRegRun,ind) $ (v_quantVa.up(cgeRegRun,ind) ne 0) ..

    v_priceVa(cgeRegRun,ind)
*
       =E=

       sum(primFac $ pv_shareParCES.l(cgeRegRun,ind,primFac),
*
                     pv_shareParCES(cgeRegRun,ind,primFac)
*
*                  --- price plus per unit tax
*                      (or minus per unit subsidy)
*
                   * [v_pricePrimFacInd(cgeRegRun,ind,primFac)
                       * ( 1+ pv_primFacTaxRate(cgeRegRun,ind,primFac))
                           *(1+v_slackCge(cgeRegRun,ind,primFac))
                     ]**(1-p_rhoVa(cgeRegRun,ind))

              ) ** ( 1 / (1-p_rhoVa(cgeRegRun,ind)) );
```

Primary factor taxes

The industries pay per unit taxes on primary factors and receive subsidies for their use (*p_primFacTax*) which are added to the primary factor prices applicable for each industry *v_pricePrimFacInd.* Accordingly, the first-order-conditions for factor use are defined in share form as follows:

```
* --- Conditional factor demand equations derived from CES aggregator with cost minimization

*
*     --- nest inside the VA: substitution between primary factors
*
 e_primFacUse(cgeRegRun,ind,primFac) $ [v_primFacUse.up(cgeRegRun,ind,primFac) GT 0] ..

*    With this CES formulation, factor demand is: A share of output ...

     v_quantVa(cgeRegRun,ind)/v_primFacUse.scale(cgeRegRun,ind,primFac)

     *  pv_shareParCES(cgeRegRun,ind,primFac)

*    ... times ratio of output price to factor price plus tax, to the power of the substitution elasticity
     *  [  v_priceVA(cgeRegRun,ind)
         /(v_pricePrimFacInd(cgeRegRun,ind,primFac)*(1+pv_primFacTaxRate(cgeRegRun,ind,primFac)))
        ]**(p_rhoVa(cgeRegRun,ind))

     =E= (v_primFacUse(cgeRegRun,ind,primFac))
                           /v_primFacUse.scale(cgeRegRun,ind,primFac);
```

The value of the primary taxes paid summed over all industries *v_taxPrimFac* in a region is thus defined as:

```
*
* --- primary factor tax income regional
*
  e_taxPrimFac(cgeRegRun,primFac) ..

    v_taxPrimFac(cgeRegRun,primFac)
        =E= sum(ind, v_primFacUse(cgeRegRun,ind,primFac)
                  * p_primFacTax(cgeRegRun,ind,primFac));
```

Intermediate demand

The intermediate demand is distributed to the individual sectors based on Leontief coefficients, and sourced from regional, national and international origin according to the Armington assumption:

```
*
* --- Nest below total intermediate demand: substitution between regional / national / imported
*
e_useInt(cgeRegRun,ind1,ind,dest) $ p_iCoeff(cgeRegRun,ind1,ind,dest) ..

*   With this CES formulation, factor demand is: A share of output ...
*   Time the leontief coefficient, scale to recover total intermediate demand

     [v_quantInt(cgeRegRun,ind)
       * sum(dest1, p_iCoeff(cgeRegRun,ind1,ind,dest1))/p_intShare(cgeRegRun,ind)]
       /v_useInt.scale(cgeRegRun,ind1,ind,dest)

    *   pv_shareParInt(cgeRegRun,ind1,ind,dest)

*   ... times ratio of aggregate intermediate price to price of intermediates from that destination
    *   [  v_priceIntCoeff(cgeRegRun,ind1,ind)
          /
          (
           v_priceSalesR(cgeRegRun,ind1) $ sameas(dest,"REG")

         + sum(r2_r0(cgeRegRun,cgeMsRun),  v_priceSalesN(cgeMsRun,ind1)) $ sameas(dest,"NAT")

         + sum(r2_r0(cgeRegRun,cgeMsRun),  v_priceImpI(cgeMsRun,ind1) * v_exChgRate(cgeMsRun))
                                                                    $ sameas(dest,"INT")
          )

        ]**(p_rhoInt(cgeRegRun,ind))

         =E= v_useInt(cgeRegRun,ind1,ind,dest)/v_useInt.scale(cgeRegRun,ind1,ind,dest);
```

As the input coefficient *p_iCoeff* are, as conventionally done, define in relation to total output, the *p_intShare* parameter captures the value of total intermediate demand in total output in the benchmark to relate the input coefficients to total intermediate demand.

The related prices for each input coefficient aggregated over the three origins are defined as:

```
*
*   --- average price of input coefficients from different destination
*
e_priceIntCoeff(cgeRegRun,ind1,ind) $ sum(dest, pv_shareParInt.l(cgeRegRun,ind1,ind,dest)) ..

   sum[dest $ pv_shareParInt.l(cgeRegRun,ind1,ind,dest),
             pv_shareParInt(cgeRegRun,ind1,ind,dest)
        *
         [
           (v_priceSalesR(cgeRegRun,ind1)     *  (1+v_slackCge(cgeRegRun,ind,"REG")))  $ sameas(dest,"REG")

         + sum(r2_r0(cgeRegRun,cgeMsRun),  v_priceSalesN(cgeMsRun,ind1)
                                            *(1+v_slackCge(cgeRegRun,ind,"NAT"))) $ sameas(dest,"NAT")

         + sum(r2_r0(cgeRegRun,cgeMsRun),  v_priceImpI(cgeMsRun,ind1) * v_exChgRate(cgeMsRun)
                                            *(1+v_slackCge(cgeRegRun,ind,"INT"))) $ sameas(dest,"INT")
         ] ** ( 1-p_rhoInt(cgeRegRun,ind) )
      ] ** ( 1 / ( 1-p_rhoInt(cgeRegRun,ind)))

     =E= v_priceIntCoeff(cgeRegRun,ind1,ind);
```

The average price per unit of intermediate demand (entering the top level nest which substitutes between the value added aggregate and intermediates) is defined based on the Leontief coefficients:

```
*
*   --- price index of intermediate demand
*
e_priceInt(cgeRegRun,ind) $ sum( (ind1,dest), p_iCoeff(cgeRegRun,ind1,ind,dest)) ..

   sum[ind1,
       v_priceIntCoeff(cgeRegRun,ind1,ind) * sum(dest,p_iCoeff(cgeRegRun,ind1,ind,dest))
     ]/sum( (ind1,dest), p_iCoeff(cgeRegRun,ind1,ind,dest))
         =E= v_priceInt(cgeRegRun,ind);
```

Further equations define aggregates of intermediate across sectors:

```
*
* --- Intermediate demand by origin
*
  e_quantIntN(cgeRegRun,ind,dest) $ (v_quantIntN.range(cgeRegRun,ind,dest) ne 0) ..

   v_quantIntN(cgeRegRun,ind,dest)/v_quantIntN.scale(cgeRegRun,ind,dest)

       =E=
         sum(ind1 $ p_iCoeff(cgeRegRun,ind,ind1,dest),
                 v_useInt(cgeRegRun,ind,ind1,dest))
           /v_quantIntN.scale(cgeRegRun,ind,dest);
```

Please note that the RHS and LHS position of the terms have specific meanings in an MCP model. If LHS > RHS, the paired variable is driven to its lower limit. The economic meaning would be that if marginal costs exceed marginal revenues, the output drops to zero.

As the technology implies constant return to scales, marginal and average production costs are equal so that the producer price *v_priceProd* is exhausted by the per unit costs for the value added aggregate (*v_vaPrice* as defined above) plus the costs for intermediate and per unit taxes on production *p_taxProd*:

```
  e_zeroProfit(cgeRegRun,ind) $ (v_quantProd.up(cgeRegRun,ind) ne 0) ..

         v_priceProd(cgeRegRun,ind) =E= v_unitCost(cgeRegRun,ind)
*
*                                            --- taxes on production
*
                                        * ( 1 + p_prodTaxRate(cgeRegRun,ind));
```

As the template could be applied to several countries simultaneously, the mapping sets r2_r0 and r0_r2 which regional units belong to a specific county *cgeMsRun*.

Household

The regional household draws first income from primary factors *v_incPrimFacHou*:

```
*
* --- primary factor income definition
*     (note: taxes on primary factor use are paid by industries)
*
  e_incPrimFacHou(cgeRegRun) ..

    v_incPrimFacHou(cgeRegRun)/v_incPrimFacHou.scale(cgeRegRun)

     =E=

        sum( primFac, v_incPrimFac(cgeRegRun,primFac))
                    /v_incPrimFacHou.scale(cgeRegRun);
*
* --- primary factor income definition
*     (note: taxes on primary factor use are paid by industries)
*
  e_incPrimFac(cgeRegRun,primFac) ..

    v_incPrimFac(cgeRegRun,primFac)/v_incPrimFac.scale(cgeRegRun,primFac)

     =E=

        sum( ind,
                      v_primFacUse(cgeRegRun,ind,primFac)
                    * v_pricePrimFacInd(cgeRegRun,ind,primFac))
                    /v_incPrimFac.scale(cgeRegRun,primFac);
```

Please note that firstly the primary factor endowment might not be fully used, i.e. the sum of the factor use over the industries might not be equal to the factor endowment and that secondly returns to factor use net of taxes in the industries need not be equal, depending on the factor market structure chosen. The reader should also note that formally taxes or subsidies on primary factor use (labor, land, capital) are paid/received by the production sectors.

The regional household pays local *v_taxcIncLoc* and national income tax *v_taxIncSta* on their primary factor revenues according to percentage tax rates *p_taxIncLocRate* and *p_taxIncStaRate*:

```
*
* --- local/state income tax based on percentage tax rates
*
  e_taxIncLoc(cgeRegRun) ..

    v_taxIncLoc(cgeRegRun) =E=

         sum( (ind,primFac),
                  v_primFacUse(cgeRegRun,ind,primFac))* p_incLocTaxRate(cgeRegRun);

  e_taxIncSta(cgeRegRun) ..

    v_taxIncSta(cgeRegRun) =E=

        sum( (ind,primFac),
                 v_primFacUse(cgeRegRun,ind,primFac))

            * (    p_incStaTaxRate(cgeRegRun)
*
*              --- alternative closure rule for national state budget:
*                  change tax rate in all regions by the same percentage points
*
               + sum(r2_r0(cgeRegRun,cgeMsRun), v_chgIncStaTaxRate(cgeMSRun)) );
```

The last term, the variable *v_chgIncStaTaxRate* is linked to a closure rule for the national state account which changes in the income tax rate charged by the central government.

Additionally to primary factor income after income tax, the households receive income subsidies from the local government *v_subsHouGovLoc*, and net-borrowing from rest-of-the-world *v_nebrHou* such that total household income *v_incHou* is defined as:

```
* --- local household income
*
  e_incHou(cgeRegRun) ..

    v_incHou(cgeRegRun)/v_incHou.scale(cgeRegRun) =E=
*
*                   --- subsidies from local government
*
               [      v_subsHouGovLoc(cgeRegRun)*v_cpi(cgeRegRun)
                    + p_shockSubsHouGovLoc(cgeRegRun)
*
*                   --- primary factor earnings
*
                    + v_incPrimFacHou(cgeRegRun)
*
*                   --- net borrowing
*
                    + v_nebrHou(cgeRegRun)*sum(r2_r0(cgeRegRun,cgeMsRun),v_exChgRate(cgeMsRun))
*
*                   --- income taxes to local and national goverment
*
                    - v_taxIncLoc(cgeRegRun) - v_taxIncSta(cgeRegRun) ]/v_incHou.scale(cgeRegRun);
```

That income net of taxes and subsidies is spent on saving *v_savHou* and final consumotion *v_cnsHous* according to the (depending on the closure endogenous) saving rate, i.e. a CD utility function:

```
*
*
  e_cnsHou(cgeRegRun) ..

     [v_cnsHou(cgeRegRun)*v_cpi(cgeRegRun)]/v_cnsHou.scale(cgeRegRun)
         =E= [v_incHou(cgeRegRun) * (1-pv_savRateHou(cgeRegRun))]
               /v_cnsHou.scale(cgeRegRun);


*
* --- Savings
*
  e_savHou(cgeRegRun) ..

    v_savHou(cgeRegRun)/v_incHou.scale(cgeRegRun)
        =E= v_incHou(cgeRegRun)/v_incHou.scale(cgeRegRun) * pv_savRateHou(cgeRegRun);
```

The reader should note that total final demand *v_cnsHou* is defined in constant terms (net of changes in the consumer price index *v_cpi*).

Final demand demand *v_quantPerCap* is based on Linear Expenditure System (Stone-Geary-Utility function) and defined on a capita basis to reflect that net migration and thus population size is an endogenous variable:

```
*
* --- Final demand per capita based on Linear expenditure system
*
  e_quantPerCap(cgeRegRun,ind) $ (v_quantHou.up(cgeRegRun,ind) ne 0)..

    v_quantPerCap(cgeRegRun,ind) =E=
*
*          --- absolute term = commitment
*
           pv_commit(cgeRegRun,ind)

*          --- marginal bugdet share divided by
*              product price (= Armington price plus sales tax)
         + pv_margBudShare(cgeRegRun,ind)
              / (v_priceArm(cgeRegRun,ind)+p_salesTaxRate(cgeRegRun))
*
*          --- times non-committed income:
*              net income minus savings minus
*              values of commitments
*
          * v_nonComIncPerCap(cgeRegRun);
```

It is driven by the Armington prices *v_priceArm* which is weighted average price over purchases from the regional, rest-of-nation and rest-of-world market plus per unit sales taxes *p_salesTaxRate.*

The LES demand system has two parameters per product: the *pv_commit* parameter describes the so-called commitments, i.e. quantities consumed independent of prices and income, and the marginal budget shares *pv_margBudShare*. The non committed income per capita *v_nonComIncPerCap* is defined as total regional house income minus savings divided by the population size minus the value of the commitments:

```
*
* --- Non committed income per capita
*
  e_nonComIncPerCap(cgeRegRun) ..

    v_nonComIncPerCap(cgeRegRun)/v_nonComIncPerCap.scale(cgeRegRun)

     =E=

      [      (v_cnsHou(cgeRegRun)*v_cpi(cgeRegRun))
                      /v_popSize(cgeRegRun)

            - sum(ind1,
                pv_commit(cgeRegRun,ind1)
                           * v_priceArm(cgeRegRun,ind1)*(1+p_salesTaxRate(cgeRegRun))
            )
      ]/v_nonComIncPerCap.scale(cgeRegRun)
 ;
```

The population size in the calibration point is equal to *p_popSize*, the endogenous population size is hence given by adding net migration *v_netMigr*:

```
 e_popSize(cgeRegRun) ..

   v_popSize(cgeRegRun)/p_popSize(cgeRegRun)
    =E= (p_popSize(cgeRegRun)+v_netMigr(cgeRegRun))/p_popSize(cgeRegRun);
```

The per capita consumption is aggregated to total by multiplication with the population size:

```
*
* --- Final demand total (from per capita to regional total)
*
  e_quantHou(cgeRegRun,ind) $ (v_quantHou.up(cgeRegRun,ind) ne 0)..
*
    v_quantHou(cgeRegRun,ind)/v_quantHou.scale(cgeRegRun,ind)

       =E=

       v_quantPerCap(cgeRegRun,ind)/v_quantHou.scale(cgeRegRun,ind)
                 * v_popSize(cgeRegRun);
```

Closure for the household

Currently, three different closure rules are integrated: (1) by adjusting the saving rate, (2) by adjusting subsidies received from the government or (3) by adjusting net borrowing.

```
$iftheni "%closure_hou%"=="Saving rate"
        e_balHou.pv_savRateHou
$elseifi "%closure_hou%"=="Government subsidies"
        e_balHou.v_subsHouGovLoc
$else
        e_balHou.v_nebrHou
$endif
```

Depending on the closure rule, certain variables are fixed or not (regcge\regcge_set_bounds.gms)

```
* ----------------------------------------------------------------------------
*
*   Alternative closures for reginal household
*
* ----------------------------------------------------------------------------

   display "closure hou =%closure_hou%";
*
* --- standard case: consumption and saving adjusts
*
  pv_savRateHou.fx(cgeRegRun)   =  pv_savRateHou.l(cgeRegRun);
  v_nebrHou.fx(cgeRegRun)       =  v_nebrHou.l(cgeRegRun);


$iftheni "%closure_hou%"=="Saving rate"
*
* --- final consumption fixed and net borrowing from abroad
*     are fixed, the household balances by more or less savings
*
  v_cnsHou.fx(cgeRegRun) = v_cnsHou.l(cgeRegRun);

  pv_savRateHou.lo(cgeRegRun) = -2;
  pv_savRateHou.up(cgeRegRun) = +1;


$elseifi "%closure_hou%"=="Foreign borrowing"
*
* --- final consumption is fixed and saving
*     depend on total income, household
*     balanced by adjusting net borrowing from abroad
*
  v_cnsHou.fx(cgeRegRun) = v_cnsHou.l(cgeRegRun);

  v_nebrHou.lo(cgeRegRun)   = -abs(v_nebrHou.l(cgeRegRun))*10 - 1000;
  v_nebrHou.up(cgeRegRun)   = +abs(v_nebrHou.l(cgeRegRun))*10 + 1000;


$endif
```

Migration

The template allows for migration *v_netMigr* between regions, driven by the relation between the per capita household income *v_incPerCap* at regional and national level *v_incPerCapN* and a similar term for the employment rate (*v_emplRate*, *v_emplRateN*):

```
*
* --- Net migration behaviorual equation
*
  e_netMigr(cgeRegRun) $ (v_netMigr.range(cgeRegRun) ne 0) ..

    v_netMigr(cgeRegRun) =E=

           [      p_cnstNetMigr(cgeRegRun)

*
*              --- term driven by regional in relation to national
*                  per capita income
*
              + p_slopeNetMigr(cgeRegRun,"incPerCap") * p_popSize(cgeRegRun)
                * (                               v_incPerCap(cgeRegRun)
                  /sum( r2_r0(cgeRegRun,cgeMsRun), v_incPerCapN(cgeMsRun))-1)
*
*              --- term driven by regional in relation to national
*                  employment rate
*
              + p_slopeNetMigr(cgeRegRun,"emplRate") * p_popSize(cgeRegRun)
                * (                                v_EmplRate(cgeRegRun)
                  /sum( r2_r0(cgeRegRun,cgeMSRun), v_EmplRateN(cgeMsRun))-1)
            ]
*
            + p_popSize(cgeRegRun)
*
*           ---. the following term esnures that net migration
*                balances
*
                          * sum(r2_r0(cgeRegRun,cgeMSRun),
                                  v_netMigrClosTerm(cgeMsRun));
```

The last term ensures, driven by *v_netMigrClosTerm,* ensures that the sum of the net migration over the regions in a country is equal to zero and is linked to the following balancing equation:

```
*
* --- sum of regional net migration must be equal zero, defines v_netMigrClosTerm
*
  e_netMigrMS(cgeMsRun) $ (sum( map_rr(cgeMsRun,cgeRegRun),
                              v_netMigr.range(cgeRegRun)) ne 0) ..

    sum(map_rr(cgeMsRun,cgeRegRun), v_netMigr(cgeRegRun)) =E= 0;
```

The net migration does not only update the population size, but also the work force and thus impacts labor markets and consequently also the employment rates in the regions:

```
*
* --- define regional labor force
*
  e_labForce(cgeRegRun) ..

     v_labForce(cgeRegRun) =E=    p_primFacEnd(cgeRegRun,"labor")
                                + v_netMigr(cgeRegRun) * p_labPerPop(cgeRegRun);
```

Per capita consumption and the employment rate as the variables driving net migration are defined in the four following equations:

```
*
* --- define income per head
*
  e_incPerCap(cgeRegRun) ..

    v_incPerCap(cgeRegRun)
       =E= v_incHou(cgeRegRun)/(p_popSize(cgeRegRun)+v_netMigr(cgeRegRun));
*
* --- define income per head national
*
  e_incPerCapN(cgeMsRun) $ (sum( map_rr(cgeMsRun,cgeRegRun),
                                   v_netMigr.range(cgeRegRun)) ne 0) ..

    v_incPerCapN(cgeMsRun)
      =E= sum( map_rr(cgeMsRun,cgeRegRun), v_incHou(cgeRegRun))
         /sum( map_rr(cgeMsRun,cgeRegRun), p_popSize(cgeRegRun)+v_netMigr(cgeRegRun));
*
* --- define employment rate, regional
*
  e_EmplRate(cgeRegRun) $ (v_netMigr.range(cgeRegRun) ne 0) ..

    v_EmplRate(cgeRegRun) =E=
                sum(ind1, v_primFacUse(cgeRegRun,ind1,"labor"))
             /    v_labForce(cgeRegRun);
*
* --- define employment rate, national
*
  e_EmplRateN(cgeMSRun) $ (sum( map_rr(cgeMsRun,cgeRegRun),
                                    v_netMigr.range(cgeRegRun)) ne 0) ..

    v_EmplRateN(cgeMsRun) =E=

      sum( (map_rr(cgeMsRun,cgeRegRun),ind1), v_primFacUse(cgeRegRun,ind1,"labor"))
  /   sum(  map_rr(cgeMsRun,cgeRegRun),        v_labForce(cgeRegRun));
```

Government

The government is split into two regional levels: the national one and the regional ones. The national government collects all taxes except local income tax which defines its income *v_incGovSta*:

```
*
* --- tax income government
*
  e_incGovSta(cgeMsRun) ..

    v_incGovSta(cgeMsRun) =E=

        sum(map_rr(cgeMsRun,cgeRegRun),

*
*       --- national income tax collected
*
           v_taxIncSta(cgeRegRun)
*
*       --- primary factor taxes collected
*
          + v_incTaxPrimFac(cgeRegRun)
*
*       --- production taxes collected from industry
*
          + v_incTaxProd(cgeRegRun)
*
*       --- sales tax from sale transaction of household
*
          +  v_incTaxSales(cgeRegRun)
*
*       --- investment tax
*
          + v_incTaxInv(cgeRegRun)

        );
```

The different parts of the tax revenue are defined as follows:

```
*
* --- Production taxes
*
  e_incTaxProd(cgeRegRun) ..

    v_incTaxProd(cgeRegRun) =E=
          sum(ind,  v_quantProd(cgeRegRun,ind) * p_prodTaxRate(cgeRegRun,ind));
*
* --- sales taxes
*
  e_incTaxSales(cgeRegRun) ..

    v_incTaxSales(cgeRegRun) =E=
          sum(ind,  v_quantHou(cgeRegRun,ind) * p_salesTaxRate(cgeRegRun));
*
* --- investment taxes
*
  e_incTaxInv(cgeRegRun) ..

    v_incTaxInv(cgeRegRun) =E=
          sum(ind,  v_quantInv(cgeRegRun,ind) * p_invTaxRate(cgeRegRun));
*
* --- primary factor taxes
*
  e_incTaxPrimFac(cgeRegRun) ..

    v_incTaxPrimFac(cgeRegRun) =E=
          sum(primFac, v_taxPrimFac(cgeRegRun,primFac));
```

Currently, all tax rates with the exemption of the income tax rate and the primary factor taxes are defined as parameters and hence cannot be used in a closure rule. That can be easily changed, though. Primary factor taxes can be negative in which case they are interpreted as per unit subsidies.

In order to reflect the fact that budgets for subsidies are often fixed, a budget envelope for a subsidy *p_facSubBudget* can be introduced:

```
*
* --- endogenous factor subsidy rate by given budget
*
  e_facSubBudget(cgeRegRun,ind,primFac) $ p_facSubBudget(cgeRegRun,ind,primFac) ..

    p_facSubBudget(cgeRegRun,ind,primFac) =E=
      -pv_primFacTaxRate(cgeRegRun,ind,primFac) * v_primFacUse(cgeRegRun,ind,primFac);
```

An increase in primary factor use as a response to the introduction of the subsidy will hence endogenously decrease the subsidy rate *pv_primFacTaxRate* to avoid implausible costs to the tax payer.

The national government does not consume or invest directly, but only acts as a distribution agency:

```
*
* --- state government balance
*     (income = spending)
*
  e_balGovSta(cgeMsRun) $ v_nebrGovSta.range(cgeMsRun)   ..

    v_incGovSta(cgeMsRun)/v_incGovSta.scale(cgeMsRun)

     =E=

    [
*
*    --- subsidies to local governments
*
       v_taxTransfers(cgeMsRun) * v_cpiGov(cgeMsRun)
*
*    --- national state borrowing
*
       - v_nebrGovSta(cgeMsRun) * v_exChgRate(cgeMsRun)

       ]/v_incGovSta.scale(cgeMsRun);
```

The *v_taxStateHelp* variable distributes national tax income plus net borrowing - the national government budget imbalance (*v_nebrGovSta*) - to the local governments:

```
  e_taxStateHelpDistKey(cgeMsRun) ..

      v_taxStateHelpDistKey(cgeMsRun)
       /v_taxTransfers.scale(cgeMsRun)

        =E=
*
*        --- difference in gov account
*
            [            v_incGovSta(cgeMsRun)
                        +v_nebrGovSta(cgeMsRun)*v_exChgRate(cgeMsRun)
                        -v_taxTransfers.scale(cgeMsRun)*v_cpiGov(cgeMsRun)/v_cpiGov.scale(cgeMsRun)
             ]

      /v_taxTransfers.scale(cgeMsRun);
```

The local government draws local income tax and receives its share of the central tax revenue as defined above plus transfer from the EU for RD measures:

```
*
  e_incGovLoc(cgeRegRun) ..

    v_incGovLoc(cgeRegRun)/v_incGovLoc.scale(cgeRegRun)

      =E=
       [
          v_taxIncLoc(cgeRegRun)
        + v_taxStateHelp.l(cgeRegRun) * sum(r0_r2(cgeMsRun,cgeRegRun), v_cpiGov(cgeMsRun)/v_cpiGov.scale(cgeMsRun))

        + sum(r2_r0(cgeRegRun,cgeMsRun), v_taxStateHelpDistKey(cgeMSRun)*p_popSize(cgeRegRun)
                   /sum(cgeRegRun1 $ r0_r2(cgeMSRun,cgeRegRun1), p_popSize(cgeRegRun1)))

        + p_EUBudTransfer(cgeRegRun)
       ]/v_incGovLoc.scale(cgeRegRun);
```

That local government income *v_incGovLoc* is distributed to subsidies paid to the local household *v_subsHouGovLoc*, and the sum of final consumption and savings *v_cnsSavGov*:

```
*
* --- Local government balance
*     (income = spending)
*
  e_cnsSavGov(cgeRegRun) ..

   v_cnsSavGov(cgeRegRun)/v_cnsSavGov.scale(cgeRegRun) =E=

    [
              v_incGovLoc(cgeRegRun)
*
*             --- subsidies paid to household
*
              -  v_subsHouGovLoc(cgeRegRun)*v_cpi(cgeRegRun)
*
              - p_shockSubsHouGovLoc(cgeRegRun)
    ]/v_cnsSavGov.scale(cgeRegRun);
```

The local government distributes the income not distributed to the households according to a CD function to consumption and savings based on possibly endogenous saving rate *p_savRateGov*, once exogenous savings (= investment activities of the governments) are taking into account:

```
*
* --- Local goverment savings (= fix budget shares)
*
  e_savGov(cgeRegRun) ..

    v_savGov(cgeRegRun)/v_savGov.scale(cgeRegRun)

     =E=

    [
      (    v_cnsSavGov(cgeRegRun)
         - sum(ind, p_shockInvGov(cgeRegRun,ind)))
*
*     --- saving rate from SAMs
*
      * (    p_savRateGov(cgeRegRun)
*
*     --- uniform change at country level to close account
*         (possible closure rule for national government account)
*
           + sum(r2_r0(cgeRegRun,cgeMsRun),v_chgSavRateGov(cgeMSRun))
        )
```

Total final demand by the government *v_cnsGov* is then the residual:

```
*
* --- Local government balance
*     (income = spending)
*
  e_cnsGov(cgeRegRun) ..

    v_cnsGov(cgeRegRun)/v_cnsGov.scale(cgeRegRun)

     =E=
           [
                v_cnsSavGov(cgeRegRun)
              - v_savGov(cgeRegRun)
            ]/v_cnsGov.scale(cgeRegRun);
```

The demand is based on a LES system, once exogenous demand shocks are considered:

```
*
  e_nonCommitIncGov(cgeRegRun) ..

    v_nonCommitIncGov(cgeRegRun)/v_nonCommitIncGov.scale(cgeRegRun)

         =E= [
                v_cnsGov(cgeRegRun)

             - sum(ind1, pv_commitGov(cgeRegRun,ind1)*v_priceArm(cgeRegRun,ind1)
                        +p_shockDemGov(cgeRegRun,ind1)
                  )
              ]/v_nonCommitIncGov.scale(cgeRegRun);

*
*     Note: exogenous shocks are deducted beforehand
*
  e_quantGov(cgeRegRun,ind) $ (v_quantGov.up(cgeRegRun,ind) ne 0) ..

    v_quantGov(cgeRegRun,ind)/v_quantGov.scale(cgeRegRun,ind)

         =E=
             [ v_nonCommitIncGov(cgeRegRun)
                 * sum(r2_r0(cgeRegRun,cgeMsRun),pv_margBudShareGov(cgeMsRun,ind))
                   /v_priceArm(cgeRegRun,ind)
*
*               --- commitment
*
                + pv_commitGov(cgeRegRun,ind)
*
*               --- additional exogenous demand of state
*
                + p_shockDemGov(cgeRegRun,ind)/v_priceArm(cgeRegRun,ind)

             ]/v_quantGov.scale(cgeRegRun,ind);
```

Closure rules for the government account

Three different closures are implemented: (1) adjustment of the saving rate, (2) adjustment of net borrowing or (3) adjustment of the state income tax rate.

```
$iftheni "%closure_gov%"=="Saving rate"
         e_balGovSta.v_chgSavRateGov
$elseifi "%closure_gov%"=="Borrowing"
         e_balGovSta.v_nebrGovSta
$else
         e_balGovSta.v_chgIncStaTaxRate
$endif
```

Depending on the closure rule, certain variables in the model are fixed or not (see regcge\regcge_closure.gms):

```
* ----------------------------------------------------------------------------
*
*   Alternative closures for national government account
*
* ----------------------------------------------------------------------------
*
  display "closure gov =%closure_gov%";
*
* --- standard case: spending of national gov adjusts
*
  v_chgIncStaTaxRate.fx(cgeMSRun) = 0;
  v_nebrGovSta.fx(cgeMsRun)       = v_nebrGovSta.l(cgeMsRun);
  v_chgSavRateGov.fx(cgeMsRun)    = 0;
  v_taxTransfers.fx(cgeMsRun) = v_taxTransfers.l(cgeMsRun);
*
$iftheni "%closure_gov%"=="Saving rate"
*
*  ---- saving rate adjusts while spending is fixed
*

  v_chgSavRateGov.lo(cgeMsRun) = -1;
  v_chgSavRateGov.up(cgeMsRun) = +1;

$elseifi "%closure_gov%"=="Foreign borrowing"
*
*  ---- foreign debt adjusts while spending is fixed
*
  v_nebrGovSta.lo(cgeMsRun)   = -abs(v_nebrGovSta.l(cgeMsRun))*10 - 100;
  v_nebrGovSta.up(cgeMsRun)   = +abs(v_nebrGovSta.l(cgeMsRun))*10 + 100;

$elseifi "%closure_gov%"=="Income tax"
*
*  ---- national income tax rate adjusts while spending is fixed
*
  v_chgIncStaTaxRate.lo(cgeMSRun) = -1;
  v_chgIncStaTaxRate.up(cgeMSRun) = +1;

$else

  v_taxTransfers.lo(cgeMsRun) = 0.1 * v_taxTransfers.l(cgeMsRun);
  v_taxTransfers.up(cgeMsRun) = 10. * v_taxTransfers.l(cgeMsRun);

$endif
```

Savings and investment

The total budget for non-exogenous investments *v_invBudget* defined as follows:

```
*
* --- "Free" investment buget
*
*            Beware saving of the government already comprise
*            exogenous shocks to capital stocks
*
 e_invBudget(cgeRegRun) ..

  v_invBudget(cgeRegRun)
        / [ (v_savHou.scale(cgeRegRun) + v_savGov.scale(cgeRegRun)) * 0.1]


       =e= [v_savHou(cgeRegRun) + v_savGov(cgeRegRun)
*
*            --- take into account value of exogenous investment shocks
*                and fixed investments

              - sum(ind1, pv_fixedInv(cgeRegRun,ind1)
                          *v_priceArm(cgeRegRun,ind1)*(1+p_invTaxRate(cgeRegRun))
                   )]
        / [(v_savHou.scale(cgeRegRun) + v_savGov.scale(cgeRegRun)) * 0.1];
*
```

The distribution of household and government savings to individual commodities *v_quantInv* is based on fixed investment quantities *p_fixedInv* which are identical to the share of the original SAM values and strictly positive shares *p_invShare*, however taking into account that investment shock *p_shockInvGov* might be introduced for the local government:

```
*
  e_quantInv(cgeRegRun,ind) $ (v_quantInv.up(cgeRegRun,ind) ne 0) ..

    v_quantInv(cgeRegRun,ind)/v_quantInv.scale(cgeRegRun,ind)

      =E=

         [
*
*           --- fixed investments from SAM plus government shocks
*               (quantity)
*
            pv_fixedInv(cgeRegRun,ind)
*
*           --- investment budget
*
          + v_invBudget(cgeRegRun)
                * pv_invShare(cgeRegRun,ind)
                   /(v_priceArm(cgeRegRun,ind) * (1+p_invTaxRate(cgeRegRun)))

         ]/v_quantInv.scale(cgeRegRun,ind);
```

The fixed part is introduced to avoid problems with negative investment entries in the SAM. In the latter case, an overall increase in investment demand will trigger further disinvestment in the sector with negative SAM entries. That might cause a rather implausible overall simulation behavior, as these sectors then become Giffen-goods if household and/or government income increases.

Endogenous capital stocks

There are two options to endogenize the capital stocks: the DPSV rule with sector specific investments into sector specific capital stocks or a simplified structure where total investments augment the regional capital stock.

DPSV

If the DPSV investment rule is used, expected returns to adjusted capital stocks of the industries drive the distribution of investment to the industry specific capital stocks. The basis of the distribution is the assumption that the aggregate investor will ensure that gross investments in all industries generate equal expected economic returns (or the investment drops to zero). That expected return corrects for the depreciation (difference between gross and net investment), but also a prudency factor which accounts for size of the planned gross investment *v_grossInv* in relation to the exogenous starting capital stock p_capStock. That prudency factor is applied to the change in the returns to capital:

```
e_expectedReturnToCap(cgeRegRun,ind) $ ( (v_capStock.range(cgeRegRun,ind) ne 0) and p_capStock(cgeRegRun,ind)) ..

  v_expectedReturnToCap(cgeRegRun,ind) =E=

*
    [
        v_pricePrimFacInd.scale(cgeRegRun,ind,"capital")*v_cpi(cgeRegRun)
*
*    --- change in gross returns to capital
*
        + (v_pricePrimFacInd(cgeRegRun,ind,"capital")-v_pricePrimFacInd.scale(cgeRegRun,ind,"capital")*v_cpi(cgeRegRun))
*
*    -- corrected by "prudency factor" (model solving period < depreciation
*       period => part of future returns is not clear). The larger
*       expansion of the capital stock, the higher a reduction of
*       expected return
*
       * [  1 -  p_prudInvFactor(cgeRegRun)
                * sqr( (v_capstock(cgeRegRun,ind) - p_capStock(cgeRegRun,ind))
                        /p_capStock(cgeRegRun,ind))
         ]

      ]

*
*    --- corrected for depreciation rate
*
       * (1. - p_deprRate(cgeRegRun,ind));
```

The per unit price of investment *v_priceInv* is defined as:

```
*
* --- calculate average price of investment
*
  e_priceInv(cgeRegRun) ..

          v_priceInv(cgeRegRun)

         =E=

          [ (v_savHou(cgeRegRun)  + v_savGov(cgeRegRun) )/v_quantInvTot(cgeRegRun)];
```

Total saving and investments are simple quantity aggregators and must be balanced by adjusting the expected returns to capital:

```
*
* --- add investment (= quantity index)
*
  e_quantInvTot(cgeRegRun) ..

   v_quantInvTot(cgeRegRun) =E=
        sum(ind, v_quantInv(cgeRegRun,ind));


*
* --- add capital stock changes
*
  e_grossInvTot(cgeRegRun) $ ( sum(ind, v_capStock.range(cgeRegRun,ind)) ne 0) ..

   v_grossInvTot(cgeRegRun) =E=
        sum(ind, v_grossInv(cgeRegRun,ind))
      + sum(ind, p_shockInvGov(cgeRegRun,ind));
```

The starting capital stock *p_capStock* minus depreciation plus gross investment *v_grossInv* and investment shocks defines the capital stocks *v_capStock* used by the industries:

```
*
*  --- Capital stock definition:
*      starting stock minus depreciation plus
*      net investments
*
 e_capStock(cgeRegRun,ind) $ ( v_capStock.range(cgeRegRun,ind) ne 0)   ..
*
       v_capStock(cgeRegRun,ind) / v_capStock.scale(cgeRegRun,ind)

       =E=

       [
*
*       --- gross investments by private sector
*
      + v_grossInv(cgeRegRun,ind)
*
*       --- gross investments from RD policy shock
*           (= expansion of capital stock)
*
      + p_shockInvGov(cgeRegRun,ind)/v_priceGrossInv(cgeRegRun,ind)
*
*       --- starting stock minus deprecitation
*
      + p_capStock(cgeRegRun,ind) * ( 1 - p_deprRate(cgeRegRun,ind))

      ] / v_capStock.scale(cgeRegRun,ind);
```

The price for investment in each industry is defined by:

```
 e_priceGrossInv(cgeRegRun,ind) $ p_shockInvGov(cgeRegRun,ind) ..

   v_priceGrossInv(cgeRegRun,ind) =E=
          sum(ind1 $ pv_invShare.l(cgeRegRun,ind1),
                      pv_invShare.l(cgeRegRun,ind1)*(v_priceArm(cgeRegRun,ind1) * (1+p_invTaxRate(cgeRegRun))));
```

Market clearing

Factor mobility

The model allows to use immobile factor (= primary factors are fixed at sectoral level), sluggish factor mobility according to a CET function and fully mobile factor:

```
*
* --- determine which factors are fixed, sluggish or mobile
*
  option kill=sf,kill=ff;


$ifi "%capMob%"  == "sluggish" sf("capital")  = YES;
$ifi "%labMob%"  == "sluggish" sf("labor")    = YES;
$ifi "%landMob%" == "sluggish" sf("land")     = YES;

$ifi "%capMob%"  == "fixed"  ff("capital")  = YES;
$ifi "%labMob%"  == "fixed"  ff("labor")    = YES;
$ifi "%landMob%" == "fixed"  ff("land")     = YES;

mf(primFac) $ (not [sf(primFac) or ff(primFac)]) = YES;
```

If a factor is defined fully mobile (member of set mf), the prices across the factors are identical:

```
*
* --- Equal returns for a sector as they use a common stock
*
  e_primFacInd(cgeRegRun,ind,mf(primFac)) $  ( ( (not sameas(primFac,"capital")) or (p_capSupply ne dpsv))
                                              $   v_primFacUse.up(cgeRegRun,ind,primFac)) ..

     v_pricePrimFacInd(cgeRegRun,ind,primFac)

  =E=
     v_pricePrimFac(cgeRegRun,primFac);
```

In case of a sluggish factor (set sf), the distribution of factors to sector is based on a CET function based on primary factor returns net of taxes:

```
*
* --- CET based sluggish factor markets
*
  e_primFacCET(cgeRegRun,sf(primFac),ind) $ v_primFacUse.up(cgeRegRun,ind,primFac) ..

   v_primFacUse(cgeRegRun,ind,primFac)
          /  v_primFacUse.scale(cgeRegRun,ind,primFac)

     =E=

       v_primFacUseTot(cgeRegRun,primFac)
          /  v_primFacUse.scale(cgeRegRun,ind,primFac)

         * pv_shareParFacCET(cgeRegRun,primFac,ind)

         * [  v_pricePrimFac(cgeRegRun,primFac)
            / v_pricePrimFacInd(cgeRegRun,ind,primFac)
           ]

         **(-p_rhotFac(cgeRegRun,primFac));
```

The related price aggregator is defined as:

```
*
* --- Calculate average return to factors across industries
*     (for sluggish factors)
*
  e_primFac(cgeRegRun,sf(primFac)) $ sum(ind $ pv_shareParFacCET.l(cgeRegRun,primFac,ind),1) ..

     v_pricePrimFac(cgeRegRun,primFac)

       =E=

          sum(ind $ pv_shareParFacCET.l(cgeRegRun,primFac,ind),
*
                      pv_shareParFacCET(cgeRegRun,primFac,ind)
*
*                   --- price plus per unit tax
*                       (or minus per unit subsidy)
*
                    * v_pricePrimFacInd(cgeRegRun,ind,primFac)**(1+p_rhotFac(cgeRegRun,primFac))

              ) ** ( 1 / (1+p_rhotFac(cgeRegRun,primFac)) );
```

The two equations for sluggish factors define implicitly a non-linear aggregator for factor use. In opposite so that, in case of fixed or fully mobile factors, simple linear aggregators are used:

```
*
* --- Simply aggregation of total primary factor use
*
 e_primFacUseTot(cgeRegRun,primFac) $ (ff(primFac) or mf(primFac)) ..

    v_primFacUseTot(cgeRegRun,primFac)
     /v_primFacUseTot.scale(cgeRegRun,primFac)

     =E=

      sum(ind,  v_primFacUse(cgeRegRun,ind,primFac))
              /v_primFacUseTot.scale(cgeRegRun,primFac);
```

Primary factor supply

The primary factors markets for land, capital and labor operate differently.

```
$ifi "%labSupply%" == "wage curve"            p_labSupply = wageCurve;
$ifi "%labSupply%" == "full employment"       p_labSupply = fullEmpl;
$ifi "%labSupply%" == "fixed employment rate" p_labSupply = fixedEmpl;
$ifi "%labSupply%" == "sticky wages"          p_labSupply = stickyWage;

$ifi "%capSupply%" == "Fixed stock"           p_capSupply = fixed;
$ifi "%capSupply%" == "DPSV rule"             p_capSupply = DPSV;
$ifi "%capSupply%" == "Capital stock updated by investments"   p_capSupply = updated;
```

Lower bounded factor prices

The *labor markets* work alternatively with the assumption of sticky wages, i.e. the wage rate is lower bounded. That will normally mean that the labor endowment is not fully used, driving the employment rate which impacts net migration. Note that in the MCP formulation of the model, the wage rate can increase to the point where the market operates as if under a market clearing price.

```
*
* --- sticky wages
*
  e_primFacG(cgeRegRun,ind,primFac) $ (     sameas(primFac,"labor")
$ifi      "%LAB%"=="Sticky wages"        and 1
$ifi not "%LAB%"=="Sticky wages"         and 0
                                         and v_primFacUse.up(cgeRegRun,ind,primFac)) ..

   v_labForce(cgeRegRun) =G=
     sum(ind1, v_primFacUse(cgeRegRun,ind1,primFac));
```

Wage curve

The wage curve describes the relation between the employment level and the relation between the real wage rate, i.e. the returns to labor divided by the consumer price index:

```
*
* --- Wage curve
*
  e_wageCurve(cgeRegRun,"Labor") $ ( p_labSupply eq wageCurve) ..

     log[  v_primFacUseTot(cgeRegRun,"labor") /v_labForce(cgeRegRun) ]

         =E=   p_cnstWageCurve(cgeRegRun)

             +  p_elasWageCurve(cgeRegRun)
              * log ( v_pricePrimFac(cgeRegRun,"labor")/v_cpi(cgeRegRun));
```

Where the consumer price index is defined as:

```
*
* -- consumer price index
*
  e_cpi(cgeRegRun) ..

    v_cpi(cgeRegRun) * sum(ind, v_quantHou(cgeRegRun,ind))

                  =E= sum(ind, v_quantHou(cgeRegRun,ind)
                                *(v_priceArm(cgeRegRun,ind)+p_salesTaxRate(cgeRegRun)));
```

Beware that the template drive the employment rate, so the elasticity must be negative and typically much large than one related to unemployment.

Sector specific endowments

The third alternative uses sector specific endowments so that also marginal returns to factors before taxes might deviates between sectors. That format can be used for all three factors and the standard way to operate the land market:

```
*
* --- Primary factors are allocated to sectors, i.e. fixed
*     returns might differ across industrires
*
  e_primFacAlloc(cgeRegRun,ind,primFac) $ ( (ff(primFac) or (sameas(primFac,"capital") and (p_capSupply eq dpsv))
                                          $ v_primFacUse.up(cgeRegRun,ind,primFac) ) ..

  [
   v_primFacUse.up(cgeRegRun,ind,primFac) $ sameas(primfac,"labor")
 + v_primFacUse.up(cgeRegRun,ind,primFac) $ sameas(primfac,"land")
 + v_capStock(cgeRegRun,ind)              $ sameas(primFac,"capital")

  ]/ v_primFacUse.scale(cgeRegRun,ind,primFac)
           =E=
     v_primFacUse(cgeRegRun,ind,primFac)/v_primFacUse.scale(cgeRegRun,ind,primFac);
```

Common factor pool (sluggish or fully mobile factors)

Total factor supply *v_primFacUseTot* depends on the specification for the different factors:

- Land supply is current always fixed (= p_primFacEnd).
- The labor force is either fixed, reacts according to the wage curve (see equation above, equation below not used) and can be updated by net migration. Further on, when fixed, either the employment rate can be kept at the benchmark level or full employment can be enforced.
- The capital supply is either fixed, i.e. equal to the sum of the sector specific stocks in the benchmark, the benchmark stock is depreciated and updated by total investment, or sector specific stocks are generated based on the DPSV rule (equation is not used):

```
*
* --- Common pool used by all industries
*
  e_primFacE(cgeRegRun,primFac) $ ( (mf(primFac) or sf(primFac))
                                    $ ( (not sameas(primFac,"labor"))   or (p_labsupply ne wageCurve))
                                    $ ( (not sameas(primFac,"capital")) or (p_capSupply ne dpsv))
                                    $ sum(ind, v_primFacUse.up(cgeRegRun,ind,primFac))) ..

  [
*
*    -- land: always equal to given endowment
*
     p_primFacEnd(cgeRegRun,"land") $ sameas(primFac,"land")

*
*    -- labor: use labor force
*
     +   v_labForce(cgeRegRun)      $ sameas(primFac,"labor")

     * ( v_emplRate.l(cgeRegRun) $ (p_labSupply  eq fixedEmpl)
         + 1                     $ (p_labSupply  ne fixedEmpl) )
*
*    -- capital
*
    + (
          sum(ind1, p_capStock(cgeRegRun,ind1))        $ (p_capSupply eq fixed)

        + { sum(ind1, p_capStock(cgeRegRun,ind1)
              * ( 1 - p_deprRate(cgeRegRun,ind1)))
              +     v_quantInvTot(cgeRegRun)       }   $ (p_capSupply eq updated)

     ) $ sameas(primFac,"capital")

  ]
     /  sum(ind1, v_primFacUse.scale(cgeRegRun,ind1,primFac))

     =E=

        v_primFacUseTot(cgeRegRun,primFac)
     /  sum(ind1, v_primFacUse.scale(cgeRegRun,ind1,primFac));
```

Commodities

Final demand and CET for supply

Final demand in any region is the sum of household, government and investment demand. The total final quantity demanded, *v_quantArm*, is sourced from regional, national and international markets following the Armington approach, using a CES function to determine the shares of each market depending on the relative prices. Similarly, on the production side, local output is transformed to regional, national and international sales using a CET function. The demand structure and the product transformation are illustrated in figure 2.

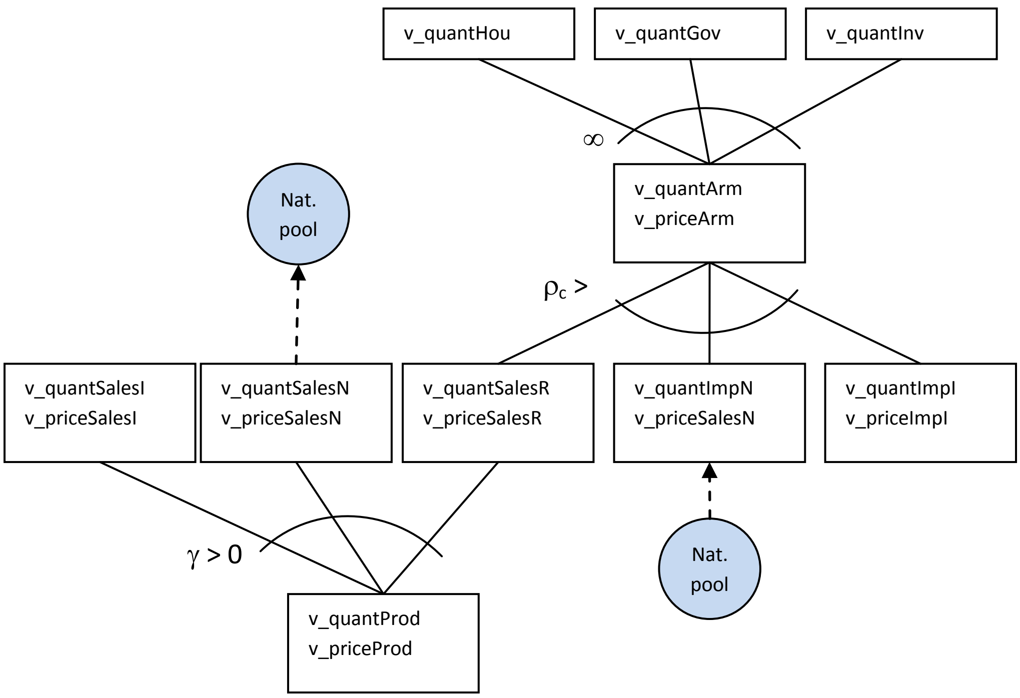


Figure 2: Demand structure and transformation of local output.

Final demand

Final demand *v_quantArm* consists of final demand by the local household *v_quantHou*, the local government *v_quantGov* and local investment demand *v_quantInv*:

```
*
* --- product markets = armington clearance
*
  e_armBal(cgeRegRun,ind) ..

     v_quantArm(cgeRegRun,ind) =E=
*
*       --- final household demand
*
        + v_quantHou(cgeRegRun,ind)
*
*       --- final government demand
*
        + v_quantGov(cgeRegRun,ind)
*
*       --- investment demand
*
        + v_quantInv(cgeRegRun,ind);
```

The average price *v_priceArm* is defined via the following equation:

```
*
* --- Armington value absorption
*
 e_priceArm(cgeRegRun,ind) ..

     [
*
*     --- regional sales times regional product price
*
      (pv_shareParArm(cgeRegRun,ind,"REG") * v_priceSalesR(cgeRegRun,ind)
           ** (1-p_rhoc(cgeRegRun,ind))) $ pv_shareParArm.l(cgeRegRun,ind,"REG")
*
*     --- imports from domestic market for final consumption
*
    + (pv_shareParArm(cgeRegRun,ind,"NAT")
        * sum(r2_r0(cgeRegRun,cgeMSRun), v_priceSalesN(cgeMsRun,ind))
            ** (1-p_rhoc(cgeRegRun,ind)) ) $ pv_shareParArm.l(cgeRegRun,ind,"NAT")
*
*     --- imports from international market for final consumption
*
    + (pv_shareParArm(cgeRegRun,ind,"INT")
           * sum(r2_r0(cgeRegRun,cgeMSRun), v_priceImpI(cgeMsRun,ind)*v_exChgRate(cgeMsRun))
                 ** (1-p_rhoc(cgeRegRun,ind)) ) $ pv_shareParArm.l(cgeRegRun,ind,"INT")

     ] ** ( 1 /(1-p_rhoc(cgeRegRun,ind)))

     =e=

     v_priceArm(cgeRegRun,ind);
```

Please note that the international price in the model in defined in international currency and driven by iso-elastic functions (see above). The exchange rate *v_exrChgRate* is defined in international currency (e.g. US$) per local currency (e.g. €). It converts the international US$ price this in the local currency € price. If the local currency devaluates, the exchange rate drops (less U$ per €) and import and export prices defined in local currency increase. That triggers less imports and exports and thus helps to drive the balance of trade towards zero.

The distribution of final demand to three origins (regional, national, international) is derived from the first order conditions from utility maximization expressed in share form:

```
*
* --- Armington share equations
*     (regional sales are used as denominator)
*
  e_armShareReg(cgeRegRun,ind) $ (v_quantSalesR.up(cgeRegRun,ind) ne 0) ..

*   With this CES formulation, factor demand is: A share of output ...

    (v_quantSalesR(cgeRegRun,ind))/v_quantSalesR.scale(cgeRegRun,ind) =E=

     v_QuantArm(cgeRegRun,ind)/v_quantSalesR.scale(cgeRegRun,ind)


    *  pv_shareParArm(cgeRegRun,ind,"REG")

*   ... times a total productivity factor exponent substituion elasticity minus 1
    *   pv_shiftParArm(cgeRegRun,ind)**(p_rhoc(cgeRegRun,ind)-1)

*   ... times ratio of output price to factor price plus tax, to the power of the substitution elasticity
    *   [  v_priceArm(cgeRegRun,ind)  / v_priceSalesR(cgeRegRun,ind) ]**(p_rhoc(cgeRegRun,ind));
```

```
e_armShareNat(cgeRegRun,ind) $ (v_quantImpN.up(cgeRegRun,ind) ne 0) ..

   v_quantImpN(cgeRegRun,ind)/v_quantImpN.scale(cgeRegRun,ind)

    =E=

     v_QuantArm(cgeRegRun,ind)/v_quantImpN.scale(cgeRegRun,ind)

   * pv_shareParArm(cgeRegRun,ind,"NAT")

*    ... times a total productivity factor exponent substituion elasticity minus 1
     *   pv_shiftParArm(cgeRegRun,ind)**(p_rhoc(cgeRegRun,ind)-1)

*    ... times ratio of output price to factor price plus tax, to the power of the substitution elasticity
     *   [  v_priceArm(cgeRegRun,ind)/
              sum(r2_r0(cgeRegRun,cgeMSRun), v_priceSalesN(cgeMsRun,ind))
         ]**p_rhos(cgeRegRun,ind);


e_armShareInt(cgeRegRun,ind) $ (v_quantImpI.up(cgeRegRun,ind) ne 0) ..

  (v_quantImpI(cgeRegRun,ind))/v_quantImpI.scale(cgeRegRun,ind) =E=

    v_QuantArm(cgeRegRun,ind)/v_quantImpI.scale(cgeRegRun,ind)

   * pv_shareParArm(cgeRegRun,ind,"INT")

*    ... times a total productivity factor exponent substituion elasticity minus 1
     *   pv_shiftParArm(cgeRegRun,ind)**(p_rhoc(cgeRegRun,ind)-1)

*    ... times ratio of output price to factor price plus tax, to the power of the substitution elasticity
     *   [  v_priceArm(cgeRegRun,ind)/
              sum(r2_r0(cgeRegRun,cgeMSRun), v_priceImpI(cgeMsRun,ind)*v_exChgRate(cgeMsRun))
         ]**p_rhos(cgeRegRun,ind);
```

National market balance

The national market must balance deliveries from the regions to the national market (**v_napExp**) and final and intermediate demand from the regions for national products:

```
*
* --- national market balance
*
  e_natBal(cgeMsRun,ind) ..

        sum(map_rr(cgeMsRun,cgeRegRun), v_natExp(cgeRegRun,ind))

    =E= sum(map_rr(cgeMsRun,cgeRegRun),
*
*            --- domestic imports for final consumption
*                (households, government, investments)
             v_natImp(cgeRegRun,ind)
*
*            --- intermediate demand
*
           + v_quantIntNN(cgeRegRun,ind));
```

The total intermediate demand for rest-of-the nation production *v_quantIntNN* of each is sector is defined as seen below:

```
*
* --- Intermediate demand for national products
*
  e_quantIntNN(cgeRegRun,ind) ..

   v_quantIntNN(cgeRegRun,ind)
       =E= sum(ind1 $ p_iCoeffN(cgeRegRun,ind,ind1),
           v_quantIntN(cgeRegRun,ind,ind1));
```

CET transformation of output

The related producer price *v_priceProd* must exhaust the revenues from intermediate regional use and selling of final demand on three regional level (domestic, national, international) according to the following price definition:

```
*
* --- CET value absorption of local production value
*
  e_priceProd(cgeRegRun,ind) $ (     (v_quantProd.up(cgeRegRun,ind) ne 0) ) ..
*
       v_priceProd(cgeRegRun,ind)*(1+v_slackCge(cgeRegRun,ind,"unitCost"))
          =E=

       [
*
*        --- regional sales and intermediate demand for local products
*
       +  [pv_shareParCET(cgeRegRun,ind,"REG") * v_priceSalesR(cgeRegRun,ind)
               ** ( 1 + p_rhot(cgeRegRun,ind))]
                       $ pv_shareParCET.l(cgeRegRun,ind,"REG")
*
*        --- export to national market
*
       + [pv_shareParCET(cgeRegRun,ind,"NAT") * sum(r2_r0(cgeRegRun,cgeMSRun), v_priceSalesN(cgeMsRun,ind))
               ** ( 1 + p_rhot(cgeRegRun,ind) ) ]
                       $ pv_shareParCET.l(cgeRegRun,ind,"NAT")
*
*        --- export to international market
*
       +  [pv_shareParCET(cgeRegRun,ind,"INT") * sum(r2_r0(cgeRegRun,cgeMSRun), v_priceIntExp(cgeMsRun,ind)*v_exChgRate(cgeMsRun))
               ** ( 1 + p_rhot(cgeRegRun,ind) ) ]
                       $ pv_shareParCET.l(cgeRegRun,ind,"INT")

      ] ** ( 1 / (1 +  p_rhot(cgeRegRun,ind)));
*
```

As above for the rest-of-the-nation, total intermediate demand from regional industries *v_quantIntRN* for each sector is defined as:

```
*
* --- Intermediate demand regional
*
  e_quantIntRN(cgeRegRun,ind) ..

   v_quantIntRN(cgeRegRun,ind)
       =E= sum(ind1 $ p_iCoeffR(cgeRegRun,ind,ind1),
           v_quantIntR(cgeRegRun,ind,ind1));
```

The first order conditions of profit maximization according to the CET transformation lead to the following share equations:

```
 e_cetShareReg(cgeRegRun,ind) $  [    (v_quantSalesR.up(cgeRegRun,ind) ne  0)
                                   or  (v_quantIntN.up(cgeRegRun,ind,"REG")  ne  0)
                                  ] ..

      (     v_quantSalesR(cgeRegRun,ind)       + v_quantIntN(cgeRegRun,ind,"REG")  )
     /(     v_quantSalesR.scale(cgeRegRun,ind) + v_quantIntN.scale(cgeRegRun,ind,"REG")  )

          =E=

           v_quantProd(cgeRegRun,ind)
     /(    v_quantSalesR.scale(cgeRegRun,ind) + v_quantIntN.scale(cgeRegRun,ind,"REG")  )

         * pv_shareParCET(cgeRegRun,ind,"REG")

         * pv_shiftParCet(cgeRegRun,ind)**(p_rhot(cgeRegRun,ind)+1)

         * [  v_priceProd(cgeRegRun,ind) / v_priceSalesR(cgeRegRun,ind) ]

     **(-p_rhot(cgeRegRun,ind));
```

```
e_cetShareNat(cgeRegRun,ind) $ (v_quantSalesN.up(cgeRegRun,ind) ne 0) ..

     v_quantSalesN(cgeRegRun,ind)/v_quantSalesN.scale(cgeRegRun,ind)

      =E=

          v_quantProd(cgeRegRun,ind)/v_quantSalesN.scale(cgeRegRun,ind)

        * pv_shareParCET(cgeRegRun,ind,"NAT")

       * pv_shiftParCet(cgeRegRun,ind)**(p_rhot(cgeRegRun,ind)+1)

       * [  v_priceProd(cgeRegRun,ind)
          / sum(r2_r0(cgeRegRun,cgeMSRun), v_priceSalesN(cgeMsRun,ind))
         ]

       **(-p_rhot(cgeRegRun,ind));


*
  e_cetShareInt(cgeRegRun,ind) $ (v_quantSalesI.up(cgeRegRun,ind) ne 0) ..

      v_quantSalesI(cgeRegRun,ind)/v_quantSalesI.scale(cgeRegRun,ind)

       =E=

           v_quantProd(cgeRegRun,ind)/v_quantSalesI.scale(cgeRegRun,ind)

        * pv_shareParCET(cgeRegRun,ind,"INT")

         * pv_shiftParCet(cgeRegRun,ind)**(p_rhot(cgeRegRun,ind)+1)

         * [  v_priceProd(cgeRegRun,ind)
            / sum(r2_r0(cgeRegRun,cgeMSRun), v_priceIntExp(cgeMsRun,ind)*v_exChgRate(cgeMsRun))
           ]

         **(-p_rhot(cgeRegRun,ind));
```

International exports and imports

The prices of the international exports and imports in foreign currency drive export demand and import supply based on an iso-elastic function:

```
 * --- Double-log export demand function
 *
   e_expDemand(cgeMsRun,ind) $ sum(r0_r2(cgeMsRun,cgeRegRun), v_quantSalesI.up(cgeRegRun,ind)) ..

      log(v_sumExports(cgeMsRun,ind)+sum(r0_r2(cgeMsRun,cgeRegRun),10))
    / log( sum(r0_r2(cgeMsRun,cgeRegRun), v_quantSalesI.scale(cgeRegRun,ind)+10))


       =E=

     [
       p_cnstIntExp(cgeMsRun,ind)
     + p_elasIntExp(cgeMsRun,ind) * log(v_priceIntExp(cgeMsRun,ind))

     ]/ log( sum(r0_r2(cgeMsRun,cgeRegRun), v_quantSalesI.scale(cgeRegRun,ind)+10));


*
* --- Double-log import supply function
*
  e_impSupply(cgeMsRun,ind) $ ( sum(r0_r2(cgeMsRun,cgeRegRun), v_quantImpI.up(cgeRegRun,ind)
                                                              +v_quantIntN.up(cgeRegRun,ind,"INT"))
                               and (v_priceImpI.range(cgeMSRun,ind) ne 0))  ..

     log(v_sumImports(cgeMsRun,ind)+sum(r0_r2(cgeMsRun,cgeRegRun),10))
      / log( sum(r0_r2(cgeMsRun,cgeRegRun), v_quantImpI.scale(cgeRegRun,ind)+10))

       =E=

   [
      p_cnstIntImp(cgeMsRun,ind)
    + p_elasIntImp(cgeMsRun,ind) * log(v_priceImpI(cgeMsRun,ind))
   ]
   / log( sum(r0_r2(cgeMsRun,cgeRegRun), v_quantImpI.scale(cgeRegRun,ind)+10));
```

The function use aggregates which are defined for the imports as follows:

```
*
* --- aggregate final demand for imports over regions
*
  e_sumImportsF(cgeMsRun,ind) $ sum(r0_r2(cgeMsRun,cgeRegRun), v_quantImpI.up(cgeRegRun,ind)) ..

       v_sumImportsF(cgeMsRun,ind)
   /   ( sum(r0_r2(cgeMsRun,cgeRegRun), v_quantImpI.scale(cgeRegRun,ind)+10))

     =E= sum(r0_r2(cgeMsRun,cgeRegRun), v_quantImpI(cgeRegRun,ind))
   /   ( sum(r0_r2(cgeMsRun,cgeRegRun), v_quantImpI.scale(cgeRegRun,ind)+10));


*
* --- aggregate imports from intermediate input demand for imports
*     and final import demand v_sumImportF
*
  e_sumImports(cgeMsRun,ind) $ (    sum(r0_r2(cgeMsRun,cgeRegRun), v_quantImpI.up(cgeRegRun,ind)
                                                                  +v_quantIntN.up(cgeRegRun,ind,"INT")) )..

       v_sumImports(cgeMsRun,ind)
   /   ( sum(r0_r2(cgeMsRun,cgeRegRun), v_quantImpI.scale(cgeRegRun,ind)+10))

     =E= [
              v_sumImportsF(cgeMsRun,ind)
            + sum(r0_r2(cgeMsRun,cgeRegRun),v_quantIntN(cgeRegRun,ind,"INT"))
         ]
   /   ( sum(r0_r2(cgeMsRun,cgeRegRun), v_quantImpI.scale(cgeRegRun,ind)+10));
```

Similar equations are introduced for exports:

```
*
* ---- aggregates exports to internatinal markets over regions
*
  e_sumExports(cgeMsRun,ind) $ sum(r0_r2(cgeMsRun,cgeRegRun), v_quantSalesI.up(cgeRegRun,ind)) ..

    v_sumExports(cgeMsRun,ind)
      / sum(r0_r2(cgeMsRun,cgeRegRun), v_quantSalesI.scale(cgeRegRun,ind)+10)
    =E= sum(r0_r2(cgeMsRun,cgeRegRun), v_quantSalesI(cgeRegRun,ind))
      / sum(r0_r2(cgeMsRun,cgeRegRun), v_quantSalesI.scale(cgeRegRun,ind)+10);
```

The elasticities can be set to zero to work with fixed import and export prices in foreign currency.

Balance of trade

The balance of trade is defined in the international currency, it shows the differences between international export revenues and the costs of international imports, both for each sector and in the aggregate:

```
*
* --- balance of trade definition in international currency ($)
*
*
  e_balOfTradeInd(cgeMsRun,ind) ..

    v_balOfTradeInd(cgeMSRun,ind)/v_balOfTradeInd.scale(cgeMsRun,ind)
       =E=

       [
           v_sumExports(cgeMsRun,ind) * v_priceIntExp(cgeMsRun,ind)
         - v_sumImports(cgeMsRun,ind) * v_priceImpI(cgeMsRun,ind)
        ] /v_balofTradeInd.scale(cgeMsRun,ind);
;
  e_balOfTrade(cgeMsRun) ..

    v_balOfTrade(cgeMSRun)/v_balOfTrade.scale(cgeMsRun)
       =E=

       sum(ind,v_balOfTradeInd(cgeMSRun,ind))
         /v_balOfTrade.scale(cgeMsRun);
```

Please note that the model distinguishes between imports for final demand *v_intImp* and imports for intermediate input use *v_equantIntIN*. Again, the latter is aggregated over individual industries:

```
*
* --- Intermediate demand international
*
  e_quantIntIN(cgeRegRun,ind) ..

   v_quantIntIN(cgeRegRun,ind)
       =E= sum(ind1 $ p_iCoeffI(cgeRegRun,ind,ind1),
           v_quantInti(cgeRegRun,ind,ind1));
*
```

The current account balance is defined in local currency and shows net borrowing of the regional households and the central government:

```
*
* --- current account balance in local currency
*
  e_curAccBal(cgeMsRun) ..

    v_curAccBal(cgeMsRun)
        /v_balOfTrade.scale(cgeMsRun)

        =E=

      [
*
*       --- net borrowing of state
*           (negative = surplus)
*
        v_nebrGovSta(cgeMsRun)
*
*       --- net borrowing of regional households
*           (negative = surplus)
*
      + sum( (r0_r2(cgeMSRun,cgeRegRun)),
                v_nebrHou(cgeRegRun))

      + sum(cgeRegRun, p_EuBudTransfer(cgeRegRun))

       ]  /v_balOfTrade.scale(cgeMsRun);
```

The current account balance, converted in international currency, depending on the closure, can be equal to the trade balance:

```
*
* --- balance of trade equals current account balance
*
  e_balTradAcc(cgeMsRun) ..

    v_balOfTrade(cgeMSRun)/v_balOfTrade.scale(cgeMsRun)
       =E= -v_curAccBal(cgeMsRun)/v_balOfTrade.scale(cgeMsRun);
```

The closing is achieved by adjusting the real exchange rate and, implicitly, also by adjustments in net borrowing. The exchange rate can be alternatively fixed so that the current account and trade balance might deviate.

Model calibration

The model calibration part comprises quite some code lines, a larger part is however related to rebalancing the SAM to overcome problems such as negative net returns to primary factors in the current release. It can be expected that these problem will not be present in the final release, so that these part can be commented out later.

The general idea is to use for calibration the very same equations as during simulation runs based on closure swaps. The commitment parameters found in the LES demand system, to give an example, are defined formally as variables in GAMS. By fixing the final consumption quantities, the behavioral equations are used to define the commitments in a separate model

```
*
* --- model to define commitments in LES demand system
*
  model m_calLES / e_quantHou,e_dummyCNS /;
```

As follows:

```
     pv_commit.lo(cgeRegRun,ind)  = -v_incHou.l(cgeRegRun)/p_popSize(cgeRegRun);
     pv_commit.up(cgeRegRun,ind)  = +v_incHou.l(cgeRegRun)/p_popSize(cgeRegRun);
     pv_commit.l (cgeRegRun,ind)  = p_margBudShare(ind) * v_incHou.l(cgeRegRun)/p_popSize(cgeRegRun) * 0.1;

     pv_commit.fx(cgeRegRun,ind)  $ (NOT v_quantHou.l(cgeRegRun,ind)) = 0;

*
*    --- use solver to calculate commitments
*
*       --- fix quantities / price
*
     v_incHou.fx(cgeRegRun)        = v_incHou.l(cgeRegRun);
     v_savHou.fx(cgeRegRun)        = v_savHou.l(cgeRegRun);
     v_priceArm.fx(cgeRegRun,ind) = v_priceArm.l(cgeRegRun,ind);
     v_quantHou.fx(cgeRegRun,ind) = v_quantHou.l(cgeRegRun,ind);

     m_calLES.holdfixed = 1;
     m_calLES.solprint  = 2;
     m_calLES.solvelink = 5;
     solve m_calLES using nlp minimizing v_obje;
```

Similar models are used to define the share and shift parameters.

Test shocks

In order to test the functioning of the template, a serious of standard test shocks is coded in GAMS in the program **regcge\testShocks.gms**. The shocks are tested for each country separately to ease location of possible problems with underlying data and parameters, such as very small value shares:

```
*
 LOOP(cgeMsIncl,
*
*    --- load current MS and related NUTS2 in template
*
    option kill=cgeMsRun;
    cgeMsRun(cgeMsIncl) = YES;

    option kill=cgeRegRun;
    cgeRegRun(cgeReg) $ sum( map_rr(cgeMsRun,cgeReg), 1) = YES;
```

There are currently four sets of shocks defined which **can** be switched on via the user interface:

```
set shocksTfp10 "Total factor productivity increase for the sector by 10%"/
    tfpIncrease10_AGR
    tfpIncrease10_FOR
    tfpIncrease10_OPP
    tfpIncrease10_FOP
    tfpIncrease10_MAN
    tfpIncrease10_ENE
    tfpIncrease10_CNS
    tfpIncrease10_TTR
    tfpIncrease10_HOT
    tfpIncrease10_EDU
    tfpIncrease10_OSE
/;

set shocksEnd10 "Endowment increase by 10%" /
      endIncrease10_Land
      endIncrease10_Capital
      endIncrease10_Labor
/;

set shocksInv50 "Investment increase by 50%" /
    InvIncrease50_AGR
    InvIncrease50_FOR
    InvIncrease50_OPP
    InvIncrease50_FOP
    InvIncrease50_MAN
    InvIncrease50_ENE
    InvIncrease50_CNS
    InvIncrease50_TTR
    InvIncrease50_HOT
    InvIncrease50_EDU
    InvIncrease50_OSE
/;
```

And finally, shocks based on rural development measures

The first set of shocks with change total factor productivity looks as follows:

```
$iftheni %shock_TFP_10% == true

*
*   --- First round of shocks:
*
*       Factor productivity increases for all sectors by 10%
*
    LOOP( shocksTfp10,

        option kill=actInd;
        actInd(ind) $ ( shocksTfp10.pos eq ind.pos ) = YES;

$batinclude "regcge\set_start_point.gms"  '"CAL"'
$batinclude "regcge\regcge_set_bounds.gms"

        pv_shiftParCes.fx(cgeRegRun1,actInd) = pv_shiftParCes.l(cgeRegRun1,actInd) * 1.1;

$batinclude "regcge\solve_model.gms"

$batinclude 'regcge\regcge_rep.gms' shocksTfp10

        pv_shiftParCes.fx(cgeRegRun1,actInd) = pv_shiftParCes.l(cgeRegRun1,actInd) / 1.1;

        p_shockTest("tfpIncrease10%",cgeMsRun,actInd,"numInfesN") = m_regCgeNlp.numinfes;
        p_shockTest("tfpIncrease10%",cgeMsRun,actInd,"sumInfesN") = m_regCgeNlp.suminfes;
        p_shockTest("tfpIncrease10%",cgeMsRun,actInd,"numInfes")  = m_regCge.numinfes;
        p_shockTest("tfpIncrease10%",cgeMsRun,actInd,"sumInfes")  = m_regCge.suminfes;
        p_shockTest("tfpIncrease10%",cgeMsRun,actInd,"secs")      = p_timeUsed;
        p_shockTest("tfpIncrease10%",cgeMsRun,actInd,"iter")      = m_regCge.iterusd;
    );

$endif
```

The shocks are implemented separately for each industry *actInd*. First, all model variables are rest to the calibration point (regcge\set_start_point.gms) and appropriate bounds are added (regcge\regcge_set_bounds.gms). Before the model is solved, the shift parameter of the production function in each region for the industry chosen *actInd* is increased by 10%. Afterwards, the model is solved (the helper program regcge\solve_model.gms) as discussed above. The results are stored in a parameter (regcge\regcge_rep.gms) for the specific shock so that they can be inspected later with the interface and with a GDX viewer.

Afterwards, the shift parameter is rest to its original value and the some basic information stored, with the main focus currently in feasibility.

In a similar fashion, the factor endowments are changed and investment shocks run. The fourth set of shocks looks differently: it switches in each experiment a specific type of RD-shock on and introduces a shock equal to 10% of the historic budget spent for the measures related to that shock. As the RD shocks change tax rates and I/O coefficients, these are re-initialized to the calibration point. Afterwards, the shocks defined (see next section) and the simulation performed.

```
*
    LOOP(rdShocks,

       option kill=actRdShocks;
       actRdShocks(rdShocks) = YES;

       option kill=p_rdSpend;
       p_rdSpend(nutsRD,rdMeasures,rdYears,bud) $ rdShocks_rdMeasures(rdShocks,rdMeasures)
         = p_rdSpendOri(nutsRD,rdMeasures,rdYears,bud)*0.1;

       execute_load "%resdir%\..\simini\ini.gdx"
                             p_iCoeffR,p_iCoeffN,p_iCoeffI,
                             pv_savRateHou;


      p_prodTaxRate(cgeRegRun,ind)                  = p_cgeRes(cgeRegRun,"",ind,"TaxProd","CAL");
      p_primFacTaxOld(cgeRegRun,ind,primFac)        = p_cgeRes(cgeRegRun,"taxRate",primFac,ind,"CAL");
      pv_primFacTaxRate.fx(cgeRegRun,ind,primFac) = p_cgeRes(cgeRegRun,"taxRate",primFac,ind,"CAL");
      pv_shiftParCES.fx(cgeRegRun,ind)              = pv_shiftParCES.scale(cgeRegRun,ind);

$batinclude "regcge\set_start_point.gms" '"CAL"'


$include 'policy\rd_shocks.gms'
$batinclude "regcge\regcge_set_bounds.gms"

$batinclude 'util\title.gms' "'Run shock : '" rdShocks.te(rdShocks) "'" for '" rall.te(cgeMSIncl)


$batinclude "regcge\solve_model.gms"

$batinclude "regcge\regcge_rep.gms" rdShocks

       p_shockTest("shock_rd_10",cgeMsRun,rdShocks,"numInfesN") = m_regCgeNlp.numinfes;
       p_shockTest("shock_rd_10",cgeMsRun,rdShocks,"sumInfesN") = m_regCgeNlp.suminfes;
       p_shockTest("shock_rd_10",cgeMsRun,rdShocks,"numInfes")  = m_regCge.numinfes;
       p_shockTest("shock_rd_10",cgeMsRun,rdShocks,"sumInfes")  = m_regCge.suminfes;
       p_shockTest("shock_rd_10",cgeMsRun,rdShocks,"secs")      = p_timeUsed;
       p_shockTest("shock_rd_10",cgeMsRun,rdShocks,"iter")      = m_regCge.iterusd;
   );
$endif


 );
```

As discussed below, the interface allows selecting the shocks and countries to run.

Policy simulations

For policy simulations, the model loads the behavioral parameters from a GDX container along with the results from the calibration run. From the results, starting points for the variables are derived and appropriate bounds introduced. Without a shock, the model should be in perfect balance after that initialization.

The reader should however note that the model is calibrated against a certain unemployment rate. If the labor market is set to full labor mobility and market clearing prices, the wage rate will drop and the unemployment rate driven to zero. With that setting, the model will hence not be able to recover the calibration point.

The assignment of RD related policy shocks follows the logic discussed in D3.2.3. In order to document the GAMS code and to avoid possible misinterpretation, for the different group of measures, the chosen solution is discussed below.

Processing the budget data

In a first step, the budget data provided by Janet Dwyer's team where transferred into a GAMS table, maintaining the original labels.

The different measures are documented with long texts in a set

```
set rdMeasures /
      "            "
*
*   --- investments in human capital in agriculture
*
       111  "Vocational training, information actions, including diffusion of scientific knowledge and innovative practices f
       114  "Use by farmers and forest holders of advisory services"
       115  "Setting up farm management, farm relief and farm advisory services, as well as forestry advisory services"
       132  "Supporting farmers who participate in food quality schemes"
       133  "Supporting producer groups for information and promotion activities for products under food quality schemes"
       142  "Setting up of producer groups"
       143  "Provision of farm advisory and extension services in Bulgaria and Romania"
       331  "Training and information for economic actors operating in the field covered by Axis 3"
```

Additional sets define the years with observations (rdYears), regional labels used (nutsRd) and the amount stemming from the EU or the national budget (bud).

```
TABLE RD_SPEND(nutsrd,rdMeasures,rdYears,bud)
                                                       EUSpend              PublicSpend
AT11              ."          ".       2001            1515787.83           2300675.51848166
AT11              ."          ".       2002            925100.44            1647686.17058467
AT11              ."          ".       2003            6773634.22           9629664.57507114
AT11              ."          ".       2004            10852916.34          15314451.8053846
AT11              ."          ".       2005            12368240.84          17050087.2544256
AT11              ."          ".       2006            13478500.52          19389168.1838849
AT11              ."111       ".       2000            11398.411            15197.8813333333
AT11              ."111       ".       2001            205476.835           273969.113333333
AT11              ."111       ".       2002            274804.796           366406.394666667
AT11              ."111       ".       2003            385122.263           513445.750666667
AT11              ."111       ".       2004            217965.587           290620.782666667
AT11              ."111       ".       2005            302408.706           403075.674666667
```

In order to assign the budget data to specific shocks, the list of shocks is defined (policy\rd_shocks.gms):

```
set rdShocks /

     humCapAgr      "Investment in human capital in agriculture"
     humCapRest     "Investments in human capital in other sectors"
     InvAgr         "Increase capital stock in agriculture"
     demGovCns      "Increase government demand for construction"
     capAgrFor      "Capital subsidies to agriculture and forestry"
     capFop         "Capital subsidies to food processing"
     incSub         "Income transfers to households"
     agriEnv        "Agri-environmental measures"
     landSubAgr     "Land subsidies to agriculture"
     landSubFor     "Land subdidies to forestry"
     subsServ       "Production subsidies to services"
     none
/;
```

The different measures present in the budget data base are mapped to these shocks based on a matching set definition:

```
set rdShocks_rdMeasures(rdShocks,rdMeasures) /

        humCapAgr.(111, 114, 115, 132, 133, 142, 143, 331)

        humCapRest.(341, 411,412,413, 421, 431, 511, "511?")

        invAgr.(121, 131, 141)

        demGovCns.(321, "322/323", 323)

        capAgrFor.("122/123/225",125,126,"321/125")

        capFop.(123)

        incSub.(112,113)

        agriEnv.(214,"215-6",216,220)

        landSubAgr.("211-213")

        landSubFor.(221,222,223,225,226,227)

        subsServ.(311,312,313)

        none."           "
/;
```

In order to avoid that measures are assigned twice or not assigned at all, a sequence of statements checks that each measure is exactly assigned once and abort program execution in a case of an error:

```
*
*  --- check that each measures is attributed exactly one time
*
   parameter rdShocksWrongAssigned(rdMeasures);

   rdShocksWrongAssigned(rdMeasures) = sum(rdShocks_rdMeasures(rdShocks,rdMeasures),1) + 1;

   rdShocksWrongAssigned(rdMeasures) $ ( rdShocksWrongAssigned(rdMeasures) eq 2) = 0;

   if (card(rdShocksWrongAssigned),
      abort "Missing or duplicate assignment of RD measures :",rdShocksWrongAssigned,
            "in %system.fn%, line %system.incline%, abort";
   );
```

The budget data are than aggregated over the base period of the SAM (2004-2006), assigning to the regional labels used in CAPRI:

```
*
* --- assign budget to shocks to CAPRI NUTS2 regions
*
 parameter p_budRdMeasures(cgeReg,rdShocks,bud) "Budget by shock type in Mio Euro, average 2004-2006";

 set avYears(rdYears) / 2004*2006 /;

 p_budRdMeasures(cgeRegRun,rdShocks,bud)
   =  sum( (r2_nutsRD(cgeRegRun,nutsRD),rdShocks_rdMeasures(rdShocks,rdMeasures),avYears),
                RD_SPEND(nutsRd,rdMeasures,avYears,bud))/card(avYears) * 1.E-6;
```

Based on that information, appropriate shocks are introduced in the CGEs.

Human capital investment in agriculture

The measures 111 "Vocational training, information actions, including diffusion of scientific knowledge and innovative practices for persons engaged in the agricultural, food and forestry sectors", 114 "Use by farmers and forest holders of advisory services", 115 "Setting up farm management, farm relief and farm advisory services, as well as forestry advisory services", 132 "Supporting farmers who participate in food quality schemes", 133 "Supporting producer groups for information and promotion activities for products under food quality schemes", 142 "Setting up of producer groups", 143 "Provision of farm advisory and extension services in Bulgaria and Romania", 331 "Training and information for economic actors operating in the field covered by Axis 3" are thought to help farmers to save production costs and are modeled as an increase in total factor productivity in agricultural in combination with an increased demand of state for education. The global variable *ftpFactor* which can be set via the interface determines by how € output increases per of € spend on the measure.

```
* ------------------------------------------------------------------------------
*
*   Human capital shocks agriculture: humCapAgr
*
* ------------------------------------------------------------------------------

*
*   --- set TFP such that subsidies save a certain % of the costs

   pv_shiftParCES.fx(cgeRegRun,"agr")
    = pv_shiftParCES.l(cgeRegRun,"agr")
        *
          p_cgeRes(cgeRegRun,"Quant","agr","prod","cal")
         /(   p_cgeRes(cgeRegRun,"Quant","agr","prod","cal")
           -
          (   p_budRdShocks(cgeRegRun,"humCapAgr","EuSpend")
           + p_budRdShocks(cgeRegRun,"humCapAgr","PublicSpend")) * %tfpFactor%);
*
*   --- the costs are provoked by higher demand for education
*
   p_shockDemGov(cgeRegRun,"edu")
        =  (   p_budRdShocks(cgeRegRun,"humCapAgr","EuSpend")
           + p_budRdShocks(cgeRegRun,"humCapAgr","PublicSpend"));
```

Human capital investment outside of agriculture

That group of measures aims to generating jobs in rural economy and is modeled by dividing the budget between a subsidy to capital use for non-agricultural sectors and an increase in government investments in those sector, assuming that the related output boosts which draws labor in these sectors is stronger than the substitution effects between factors. Equally, TFP for these sectors is increased by a certain share of the budget spend:

```
* ----------------------------------------------------------------------------
*
*   Human capital shocks outside of agriculture: humCapRest
*
* ----------------------------------------------------------------------------


   p_shockInvGov(cgeRegRun,nonAgr)
        =  p_shockInvGov(cgeRegRun,nonAgr)
      + (  p_budRdShocks(cgeRegRun,"humCapRest","EuSpend")
         + p_budRdShocks(cgeRegRun,"humCapRest","PublicSpend"))

        *            p_cgeRes(cgeRegRun,"Quant","capital",nonAgr,"cal")
         /sum(nonAgr1, p_cgeRes(cgeRegRun,"Quant","capital",nonAgr1,"cal")) * %invShare%;

    pv_primFacTaxRate.fx(cgeRegRun,nonAgr,"capital") $ p_cgeRes(cgeRegRun,"Quant","capital",nonAgr,"cal")
      =
            pv_primFacTaxRate.l(cgeRegRun,nonAgr,"capital")

         - (   p_budRdShocks(cgeRegRun,"humCapRest","EuSpend")
             + p_budRdShocks(cgeRegRun,"humCapRest","PublicSpend"))
          /sum(nonAgr1, p_cgeRes(cgeRegRun,"Quant","capital",nonAgr1,"cal")) * (1-%invShare%);

   pv_shiftParCES.fx(cgeRegRun,nonAgr)
    = pv_shiftParCES.l(cgeRegRun,nonAgr)
        *
            p_cgeRes(cgeRegRun,"Quant",nonAgr,"prod","cal")
          /(  p_cgeRes(cgeRegRun,"Quant",nonAgr,"prod","cal")
              -
            (  p_budRdShocks(cgeRegRun,"humCapRest","EuSpend")
             + p_budRdShocks(cgeRegRun,"humCapRest","PublicSpend"))
        *            p_cgeRes(cgeRegRun,"Quant",nonAgr,"prod","cal")
          /sum(nonAgr1, p_cgeRes(cgeRegRun,"Quant",nonAgr1,"prod","cal")) * %tfpFactor%);
```

Capital subsidies to food processing

That shock is related to the measures 123 “Adding value to agricultural and forestry products” and implemented in the same fashion:

```
* ----------------------------------------------------------------------------
*
*   Capital subsidies to food processing: capFop
*
* ----------------------------------------------------------------------------

*
*   --- boost capital stock in food processing
*
    p_shockInvGov(cgeRegRun,"fop")
        =  p_shockInvGov(cgeRegRun,"fop")
          +  (  p_budRdShocks(cgeRegRun,"capFop","EuSpend")
              + p_budRdShocks(cgeRegRun,"capFop","PublicSpend")) * %invShare%;

*
*   --- decrease capital tax by the given ammount for food processing
*
     pv_primFacTaxRate.fx(cgeRegRun,"fop","capital") $ p_cgeRes(cgeRegRun,"Quant","capital","fop","cal")
       = pv_primFacTaxRate.l(cgeRegRun,"fop","capital")

         -  (  p_budRdShocks(cgeRegRun,"capFop","EuSpend")
             + p_budRdShocks(cgeRegRun,"capFop","PublicSpend")) * (1 - %invShare%)
          /p_cgeRes(cgeRegRun,"Quant","capital","fop","cal");

*
*   --- set TFP such that subsidies save % of costs
*
   pv_shiftParCES.fx(cgeRegRun,"fop")
    = pv_shiftParCES.l(cgeRegRun,"fop")
        *
           p_cgeRes(cgeRegRun,"Quant","fop","prod","cal")
         /(  p_cgeRes(cgeRegRun,"Quant","fop","prod","cal")
             -
           (  p_budRdShocks(cgeRegRun,"capFop","EuSpend")
            + p_budRdShocks(cgeRegRun,"capFop","PublicSpend"))    * %tfpFactor%);
```

Investment in agriculture

The three measures 121 "Farm modernisation", 131 “Helping farmers to adapt to demanding standards based on Community legislation" and 141 "Supporting semi-subsistence farms undergoing restructuring" are modeled as direct increase of the capital stock of agriculture plus a increase in TFP:

```
* ------------------------------------------------------------------------------
*
*   Investment in agriculture: invAgr
*
* ------------------------------------------------------------------------------

*
*   --- boost capital stock in agriculture
*
    p_shockInvGov(cgeRegRun,"agr")
        = p_shockInvGov(cgeRegRun,"agr")
          + (   p_budRdShocks(cgeRegRun,"invAgr","EuSpend")
             + p_budRdShocks(cgeRegRun,"invAgr","PublicSpend"));
*
*   --- set TFP such that subsidies saves x% of costs

   pv_shiftParCES.fx(cgeRegRun,"agr")
    = pv_shiftParCES.l(cgeRegRun,"agr")
        *
           p_cgeRes(cgeRegRun,"Quant","agr","prod","cal")
          /(   p_cgeRes(cgeRegRun,"Quant","agr","prod","cal")
              -
           (   p_budRdShocks(cgeRegRun,"invAgr","EuSpend")
            + p_budRdShocks(cgeRegRun,"invAgr","PublicSpend")) * %tfpFactor%);
```

Increase in state demand for construction

The measures 321 "Basic services for the economy and rural population", "322/323" "Village renewal" and 323 "Conservation and upgrading of the rural heritage" are implemented as an increase of the government demand for construction:

```
* ------------------------------------------------------------------------------
*
*   Additional state investment in construction: demGov
*
* ------------------------------------------------------------------------------

    p_shockInvGov(cgeRegRun,"cns")
         =  (   p_budRdShocks(cgeRegRun,"demGovCns","EuSpend")
             + p_budRdShocks(cgeRegRun,"demGovCns","PublicSpend"));
```

Income subsidies

The measures 113 (*early retirement*) and 112 (*support for young farmers*) are simply implemented as a transfer of government money to the local household:

```
*
*   --- increase subsidy to local households
*
    v_subsHouGovLoc.fx(cgeRegRun) = v_subsHouGovLoc.l(cgeRegRun)
                                  + p_budRdMeasures(cgeRegRun,"incSub","EuSpend")
                                  + p_budRdMeasures(cgeRegRun,"incSub","PublicSpend");
```

Agri-environmental measures

The measures 214 (*Agri-Environmental Programs*),"215-6" (Animal welfare), 216 (*Support for non-productive investments*), 220 (???) and 224 (*Natura 2000 payments*) are treated as a subsidy to land in agriculture and combination with a TFP reduction (extensification), i.e. a reduction in the intermediate input coefficients:

```
* ----------------------------------------------------------------------------
*
*   Agri-env: subsidy to land use with TFP decrease
*
* ----------------------------------------------------------------------------
*
*   --- decrease land tax by the given ammount
*
    pv_primFacTaxRate.fx(cgeRegRun,"agr","land") $ p_cgeRes(cgeRegRun,"Quant","land","agr","cal")
      =  pv_primFacTaxRate.l(cgeRegRun,"agr","land")

        - (     p_budRdShocks(cgeRegRun,"agriEnv","EuSpend")
             +  p_budRdShocks(cgeRegRun,"agriEnv","PublicSpend"))
          /p_cgeRes(cgeRegRun,"Quant","land","agr","cal");


*
*  --- adjust intermediate input coefficients such that they save the ammount
*      of the agrienv-subsidies
*
  p_iCoeffMult(CgeRegRun,"agr") =
*
*     --- value of intermediate inputs in calibration point
*         minus the subsidy
*
     [( 1- p_cgeRes(cgeRegRun,"","agr","PriceVa","cal"))*p_cgeRes(cgeRegRun,"Quant","agr","prod","cal")
         - p_budRdShocks(cgeRegRun,"agriEnv","EuSpend")
         - p_budRdShocks(cgeRegRun,"agriEnv","PublicSpend")]
*
*     --- value of intermediate inputs in calibration point
*
    / [( 1- p_cgeRes(cgeRegRun,"","agr","PriceVa","cal"))*p_cgeRes(cgeRegRun,"Quant","agr","prod","cal")];

   p_iCoeffR(cgeRegRun,ind,"agr") = p_iCoeffR(cgeRegRun,ind,"agr") * p_iCoeffMult(CgeRegRun,"agr");
   p_iCoeffN(cgeRegRun,ind,"agr") = p_iCoeffN(cgeRegRun,ind,"agr") * p_iCoeffMult(CgeRegRun,"agr");
   p_iCoeffI(cgeRegRun,ind,"agr") = p_iCoeffI(cgeRegRun,ind,"agr") * p_iCoeffMult(CgeRegRun,"agr");


*
*  --- decrease output by value of subsidies
*

   pv_shiftParCES.fx(cgeRegRun,"agr")
      = ( v_quantProd.l(cgeRegRun,"agr")
                   - p_budRdShocks(cgeRegRun,"agriEnv","EuSpend")
                   - p_budRdShocks(cgeRegRun,"agriEnv","PublicSpend"))
        / sum[primFac $ (v_primFacUse.l(cgeRegRun,"agr",primFac) ne 0),
                pv_sigma.l(cgeRegRun,"agr",primFac)
              * v_primFacUse.l(cgeRegRun,"agr",primFac) ** (-p_rhos(cgeRegRun,"agr"))
                  ] ** (-1/p_rhos(cgeRegRun,"agr"));
```

Land subsidies to agriculture

The measures "211-213", (*Less Favored Areas*), are treated as land subsidies to agriculture:

```
* ----------------------------------------------------------------------------
*
*   Land subsidies to agriculture: landSubAgr
*
* ----------------------------------------------------------------------------
*
*   --- decrease land tax by the given ammount
*
    pv_primFacTaxRate.fx(cgeRegRun,"agr","land") $ p_cgeRes(cgeRegRun,"Quant","land","agr","cal")
      =  pv_primFacTaxRate.l(cgeRegRun,"agr","land")

        - (     p_budRdShocks(cgeRegRun,"landSubAgr","EuSpend")
            +   p_budRdShocks(cgeRegRun,"landSubAgr","PublicSpend"))
         /p_cgeRes(cgeRegRun,"Quant","land","agr","cal");

*
*   --- if the tax rate drops below -0.8,
*       increase to -0.8 and map budget to simple transfer
*
    p_budRdShocks(cgeRegRun,"incSub","EuSpend") $ ( pv_primFacTaxRate.l(cgeRegRun,"agr","land") le -0.8)
        =  p_budRdShocks(cgeRegRun,"incSub","EuSpend")
         + (-0.8 - pv_primFacTaxRate.l(cgeRegRun,"agr","land"))
              *p_cgeRes(cgeRegRun,"Quant","land","agr","cal");


    pv_primFacTaxRate.fx(cgeRegRun,"agr","land")
      = max(-0.8,pv_primFacTaxRate.l(cgeRegRun,"agr","land"));
*
*   --- now check if the change is above 0.5
*
    p_budRdShocks(cgeRegRun,"incSub","EuSpend")
       $ ( pv_primFacTaxRate.l(cgeRegRun,"agr","land") le (p_primFacTaxOld(cgeRegRun,"agr","land")-0.5))
        =  p_budRdShocks(cgeRegRun,"incSub","EuSpend")
         + ( p_primFacTaxOld(cgeRegRun,"agr","land")-0.5 - pv_primFacTaxRate.l(cgeRegRun,"agr","land")   )
              *p_cgeRes(cgeRegRun,"Quant","land","agr","cal");

    pv_primFacTaxRate.fx(cgeRegRun,"agr","land")
       $ ( pv_primFacTaxRate.l(cgeRegRun,"agr","land") le (p_primFacTaxOld(cgeRegRun,"agr","land")-0.5))
      =  p_primFacTaxOld(cgeRegRun,"agr","land")-0.5;
```

Land subsidies to forestry

Similarly, the measures 221 (*First afforestration of agricultural land*), 222 (*First establishment of agroforestry systems on agricultural land*), 223 (*First afforestration of non-agricultural land*), 225 (*Forest environment payments*) and 226 (*Restoring forestry potential and introducing prevention actions*) as well as 227 (*Support for non-productive investments*) are treated as subsidies to forestry land:

```
*
*   --- decrease land tax by the given ammount
*
    p_primFacTax(cgeRegRun,"for","land") $ p_cgeRes(cgeRegRun,"Quant","land","for","cal")
      = p_primFacTax(cgeRegRun,"for","land")

        - (     p_budRdMeasures(cgeRegRun,"landSubFor","EuSpend")
            +   p_budRdMeasures(cgeRegRun,"landSubFor","PublicSpend"))
         /p_cgeRes(cgeRegRun,"Quant","land","for","cal");
```

Similar statements as in the case of agriculture prevent implausible per unit subsidy rates.

Capital subsidies to agricultural and forestry

The measures "122/123/225" "Improving the economic value of the forest", 125 "Improving and developing infrastructure related to the development and adaptation of agriculture and forestry", 126 "Restoring agr. production potential damaged by natural disasters and introducing appropriate prevention actions" and "321/125" "Improving and developing infrastructure related to the development and adaptation of agriculture and forestry" are modeled as capital subsidy to agriculture and forestry:

```
    set agrFor(ind) / agr,"for" /;

*
*   --- decrase capital tax in agriculture and forrestry
*
    p_primFacTax(cgeRegRun,agrFor,"capital") $ p_cgeRes(cgeRegRun,"Quant","capital",agrFor,"cal")
       =    p_primFacTax(cgeRegRun,agrFor,"capital")

        -  (  p_budRdMeasures(cgeRegRun,"capAgrFor","EuSpend")
            + p_budRdMeasures(cgeRegRun,"capAgrFor","PublicSpend"))
          /(  p_cgeRes(cgeRegRun,"Quant","capital","agr","cal")
            + p_cgeRes(cgeRegRun,"Quant","capital","for","cal"));
```

Production subsidies to services

The subsidies are paid to the three service sectors in the model (trade and transport, hotel and restaurants, other services). The capture the measures 311 "Diversification into non-agricultural activities", 312 "Support for the creation and development of micro-enterprises" and 313 "Encouragement of tourism activities". By reducing production costs, the boost the output these sectors which consequently will also use more production factors (capital, labor).

```
* -----------------------------------------------------------------------------
*
*   Subsidies to services subsServ
*
* -----------------------------------------------------------------------------

    set serv(ind) / TTR,HOT,OSE /;
    alias(serv,serv1);
*
*   --- decrease production taxes to services
*
    p_taxProd(cgeRegRun,serv) $ p_cgeRes(cgeRegRun,"Quant",serv,"prod","cal")
     = p_taxProd(cgeRegRun,serv)

        - (    p_budRdMeasures(cgeRegRun,"subsServ","EuSpend")
           +   p_budRdMeasures(cgeRegRun,"subsServ","PublicSpend"))
          /sum(serv1,p_cgeRes(cgeRegRun,"Quant",serv1,"prod","cal"));
```

Budget transfer from EU to local government

The budget transfers from the EU are aggregated and added to the income of the regional government:

```
* -----------------------------------------------------------------------------
*
*   Introduce EU subsidies to local government
*
* -----------------------------------------------------------------------------
*
*   --- update EU transfers to local government
*
    p_EuBudTransfer(cgeRegRun) =  sum(rdShocks,p_budRdMeasures(cgeRegRun,rdShocks,"EUSpend"));
```

Post model processing

All endogenous variables from the model are stored on a multi-dimensional parameter in **regcge\regcge_rep.gms**:

```
*
* --- prices
*
 p_cgeRes(cgeRegRun,"",primFac,"price",%1) = v_pricePrimFac.l(cgeRegRun,primFac);
 p_cgeRes(cgeRegRun,"",ind,"priceReg",%1)  = v_priceReg.l(cgeRegRun,ind);
 p_cgeRes(cgeRegRun,"",ind,"priceProd",%1) = v_priceProd.l(cgeRegRun,ind);

 p_cgeRes(cgeRegRun,"",ind,"priceVa",%1)   = v_vaPrice.l(cgeRegRun,ind);
 p_cgeRes(cgeRegRun,"",ind,"priceArm",%1)  = v_priceArm.l(cgeRegRun,ind);

 p_cgeRes(cgeMSRun,"",ind,"priceNat",%1)     = v_priceNat.l(cgeMsRun,ind);
 p_cgeRes(cgeMSRun,"Price",ind,"impInt",%1) = v_priceIntImp.l(cgeMsRun,ind);
 p_cgeRes(cgeMSRun,"Price",ind,"expInt",%1) = v_priceIntExp.l(cgeMsRun,ind);
 p_cgeRes(cgeMSRun,"","Price","exchr",%1)    = v_exrChgRate.l(cgeMsRun);
```

Etc.

The program also aggregates all regional results to the national level, e.g.

```
*
* --- aggregate local government positions to Member State
*
 p_cgeRes(cgeMSRun,"Quant",ind,"gov",%1)     = sum(map_rr(cgeMSRun,cgeRegRun),
                                                p_cgeRes(cgeRegRun,"Quant",ind,"gov",%1));

 p_cgeRes(cgeMSRun,"",ind,"TaxProd",%1)      = sum(map_rr(cgeMSRun,cgeRegRun),
                                                p_cgeRes(cgeRegRun,"",ind,"TaxProd",%1));

 p_cgeRes(cgeMSRun,"",aggGovToMS,"gov",%1) = sum(map_rr(cgeMSRun,cgeRegRun),
                                                p_cgeRes(cgeRegRun,"",aggGovToMs,"gov",%1)
```

Welfare analysis and its decomposition

As usually done in CGEs, the welfare measurement used is the equivalent variation based on the final consumption of the household, derived from the LES demand system and taking into account changes in the consumer price index:

```
 p_cgeRes(cgeRegRun,"","EV","hou",%1)
     =
*
*    --- LES, only considering consumption of the final consumer (= assume that state consumption
*        either increases TFP or is a complement to final household consumption?)
*
      prod(ind $ pv_margBudShare.l(cgeRegRun,ind),
                (  [(p_cgeRes(cgeRegRun,"",ind,"priceArm","CAL") + p_cgeRes(cgeRegRun,"rate","Sales","hou","CAL"))
                      /p_cgeRes(cgeRegRun,"","cpi","hou","CAL")]
                  /[(p_cgeRes(cgeRegRun,"",ind,"priceArm",%1)     + p_cgeRes(cgeRegRun,"rate","Sales","hou",%1))
                      /p_cgeRes(cgeRegRun,"","cpi","hou",%1)]
               )**pv_margBudShare.l(cgeRegRun,ind) )

      *  [  ( p_cgeRes(cgeRegRun,"","Income","hou",%1)
             -p_cgeRes(cgeRegRun,"","sav","hou",%1) + p_cgeRes(cgeRegRun,"","nebrHou","hou",%1))

              / p_cgeRes(cgeRegRun,"","Pop","hou",%1)

            - sum(ind1,   pv_commit.l(cgeRegRun,ind1)
                          * (p_cgeRes(cgeRegRun,"",ind1,"priceArm",%1)+p_cgeRes(cgeRegRun,"rate","Sales","hou",%1))) ]
               /p_cgeRes(cgeRegRun,"","cpi","hou",%1)

           - [ (  p_cgeRes(cgeRegRun,"","Income","hou","CAL")
                 -p_cgeRes(cgeRegRun,"","sav","hou","CAL") + p_cgeRes(cgeRegRun,"","nebrHou","hou","CAL"))
                 / p_cgeRes(cgeRegRun,"","Pop","hou","CAL")

            - sum(ind1,  pv_commit.l(cgeRegRun,ind1)
                         * (p_cgeRes(cgeRegRun,"",ind1,"priceArm","CAL")+p_cgeRes(cgeRegRun,"rate","Sales","hou","CAL"))) ]

               /p_cgeRes(cgeRegRun,"","cpi","hou","CAL");
```

In order to analyze easier which price and quantity changes provoke the EV, a first round of an (approximate) decomposition looks at the effect of the consumer price changes. Before the loop, all prices are set to the calibration run. Afterwards, the prices are one after another increased to the simulation results, and the resulting EV at unchanged household income is calculated:

```
*
*  --- decomposition by effects, first, keep income but change process
*
 p_evPrices(cgeRegRun,ind) = (p_cgeRes(cgeRegRun,"",ind,"priceArm","CAL")+p_cgeRes(cgeRegRun,"rate","Sales","hou","CAL"))
                           / p_cgeRes(cgeRegRun,"","cpi","hou","CAL");
*
 LOOP(ind2,

*
*   --- deflated prices
*
    p_evPrices(cgeRegRun,ind2) = (p_cgeRes(cgeRegRun,"",ind2,"priceArm",%1)+p_cgeRes(cgeRegRun,"rate","Sales","hou",%1))
                               / p_cgeRes(cgeRegRun,"","cpi","hou",%1);

    p_cgeRes(cgeRegRun,"Price","EV",ind2,%1)
     =
*
*    --- LES, only considering consumption of the final consumer (= assume that state consumption
*        either increases TFP or is a complement to final household consumption?)
*
     + prod(ind $ pv_margBudShare.l(cgeRegRun,ind),

           ( [    (p_cgeRes(cgeRegRun,"",ind,"priceArm","CAL") + p_cgeRes(cgeRegRun,"rate","Sales","hou","CAL"))
                / p_cgeRes(cgeRegRun,"","cpi","hou","CAL")]
            /p_evPrices(cgeRegRun,ind))**pv_margBudShare.l(cgeRegRun,ind) )

        *  [ {(  p_cgeRes(cgeRegRun,"","Income","hou","CAL")
                -p_cgeRes(cgeRegRun,"","sav","hou","CAL") + p_cgeRes(cgeRegRun,"","nebrHou","hou","CAL"))
                / p_cgeRes(cgeRegRun,"","Pop","hou","CAL")
              } /p_cgeRes(cgeRegRun,"","cpi","hou","CAL")

              - sum(ind1, pv_commit.l(cgeRegRun,ind1)  * p_evPrices(cgeRegRun,ind1)) ]

             - [ (  p_cgeRes(cgeRegRun,"","Income","hou","CAL")
                  - p_cgeRes(cgeRegRun,"","sav","hou","CAL") + p_cgeRes(cgeRegRun,"","nebrHou","hou","CAL"))
                / p_cgeRes(cgeRegRun,"","Pop","hou","CAL")

              - sum(ind1, pv_commit.l(cgeRegRun,ind1)
                      * (p_cgeRes(cgeRegRun,"",ind1,"priceArm","CAL")+p_cgeRes(cgeRegRun,"rate","Sales","hou","CAL"))) ]
                /p_cgeRes(cgeRegRun,"","cpi","hou","CAL");

 );
```

In a similar fashion, the effect of changes in the factor price and factor use on the household is calculated:

```
*
*  --- decomposition by effects, now update household income by changing primary factor prices/endowment
*
 Loop( (evEffects,primFac),

     p_evIncome(cgeRegRun)
      = { p_cgeRes(cgeRegRun,"","Income","hou","CAL")/p_cgeRes(cgeRegRun,"","cpi","hou","CAL")

         +[  sum( (ind),
             (     p_cgeRes(cgeRegRun,"Quant",primFac,ind,%1)     $ (sameas(evEffects,"quant") or sameas(evEffects,"value"))
               +   p_cgeRes(cgeRegRun,"Quant",primFac,ind,"CAL") $  sameas(evEffects,"price")
             )

           * (     (p_cgeRes(cgeRegRun,"price",primFac,ind,%1)
                   /p_cgeRes(cgeRegRun,"","cpi","hou",%1)     )     $ (sameas(evEffects,"price") or sameas(evEffects,"value"))

               +   (p_cgeRes(cgeRegRun,"price",primFac,ind,"CAL")
                   /p_cgeRes(cgeRegRun,"","cpi","hou","CAL"))     $  sameas(evEffects,"quant")
             ))

            -sum( (ind),
                p_cgeRes(cgeRegRun,"Quant",primFac,ind,"CAL")
              * p_cgeRes(cgeRegRun,"price",primFac,ind,"CAL"))/p_cgeRes(cgeRegRun,"","cpi","hou","CAL")
          ]
        } * (1. - p_savRateHou(cgeRegRun));

     p_cgeRes(cgeRegRun,evEffects,"EV",primFac,%1)
*
*    --- LES, only considering consumption of the final consumer (= assume that state consumption
*        either increases TFP or is a complement to final household consumption?)
*

          =     (p_evInCome(cgeRegRun) / p_cgeRes(cgeRegRun,"","Pop","hou",%1))

             - [ (p_cgeRes(cgeRegRun,"","Income","hou","CAL")
                 -p_cgeRes(cgeRegRun,"","sav","hou","CAL")
                + p_cgeRes(cgeRegRun,"","nebrHou","hou","CAL"))
                   / p_cgeRes(cgeRegRun,"","Pop","hou","CAL")]/p_cgeRes(cgeRegRun,"","cpi","hou","CAL");

 );
```

The results can be inspected in the user interface, see below.

Helper programs

There are further small helper programs: **regcge\solve_model.gms** solved the model first with CONOPT as a NLP and then as a MCP with PATH, with several trials.

CONOPT seems to be generally faster in finding a solution and also more stable when used in NLP mode. The NLP solution works fine as long as variables do not hit logical bounds which represent a KKT-relation. The sticky wage representation is such a case: it is defined as inequality to allow solving the model as a NLP, by fixing the wage rate to the predetermined lower limits. However, the NLP might become infeasibility once the unemployment rates drops to zero.

In case of infeasibilities, a solution listing is produced:

```
*
* --- first solve as "pseudo"-CNS, i.e. as a NLP model
*     with a dummy objective
*
  v_pricePrimFacInd.fx(cgeRegRun,ind,"labor") = 1;
  m_regCgeNlp.solprint  = 2;
  solve m_regCgeNlp using NLP minimizing v_obje;

  p_timeUsed = m_regCgeNlp.resusd;
```

Etc.

These results can be used to define start values for the model in the program **regcge\set_start_point.gms**, which basically reverts the statements from the program above:

```
*
* --- prices
*
 v_pricePrimFac.l(cgeRegRun,primFac)      =        p_cgeRes(cgeRegRun,"",primFac,"price",%1) ;
 v_priceReg.l(cgeRegRun,ind)              =        p_cgeRes(cgeRegRun,"",ind,"priceReg",%1)  ;
 v_priceProd.l(cgeRegRun,ind)             =        p_cgeRes(cgeRegRun,"",ind,"priceProd",%1) ;
 v_vaPrice.l(cgeRegRun,ind)               =        p_cgeRes(cgeRegRun,"",ind,"priceVa",%1)   ;
 v_priceArm.l(cgeRegRun,ind)              =        p_cgeRes(cgeRegRun,"",ind,"priceArm",%1)  ;
                                                                                             ;
 v_priceNat.l(cgeMsRun,ind)               =        p_cgeRes(cgeMSRun,"",ind,"priceNat",%1)   ;
 v_priceIntImp.l(cgeMsRun,ind)            =        p_cgeRes(cgeMSRun,"price",ind,"impInt",%1);
 v_priceIntExp.l(cgeMsRun,ind)            =        p_cgeRes(cgeMSRun,"price",ind,"expInt",%1);
*
```

Etc.

If correctly set-up, the second file should generate a starting point free of infeasibility if no further shocks are introduced in the model.

The same file also introduces manual scaling for all variables and equations in the system to help solving and lead to a more sensible interpretation of feasibility and optimality related solved settings.

Graphical User Interface (GUI)

The GUI of the stand-alone version of RegCgeEU+ in GAMS is based on an interface building toolkit developed by Wolfgang Britz in Java and used also for other smaller GAMS based projects. It uses a small text file to define user operable controls. The states of these controls are converted into GAMS code stored in a file which can be included. The interface builder incorporates the CAPRI exploitation tools including the XML based view definition.

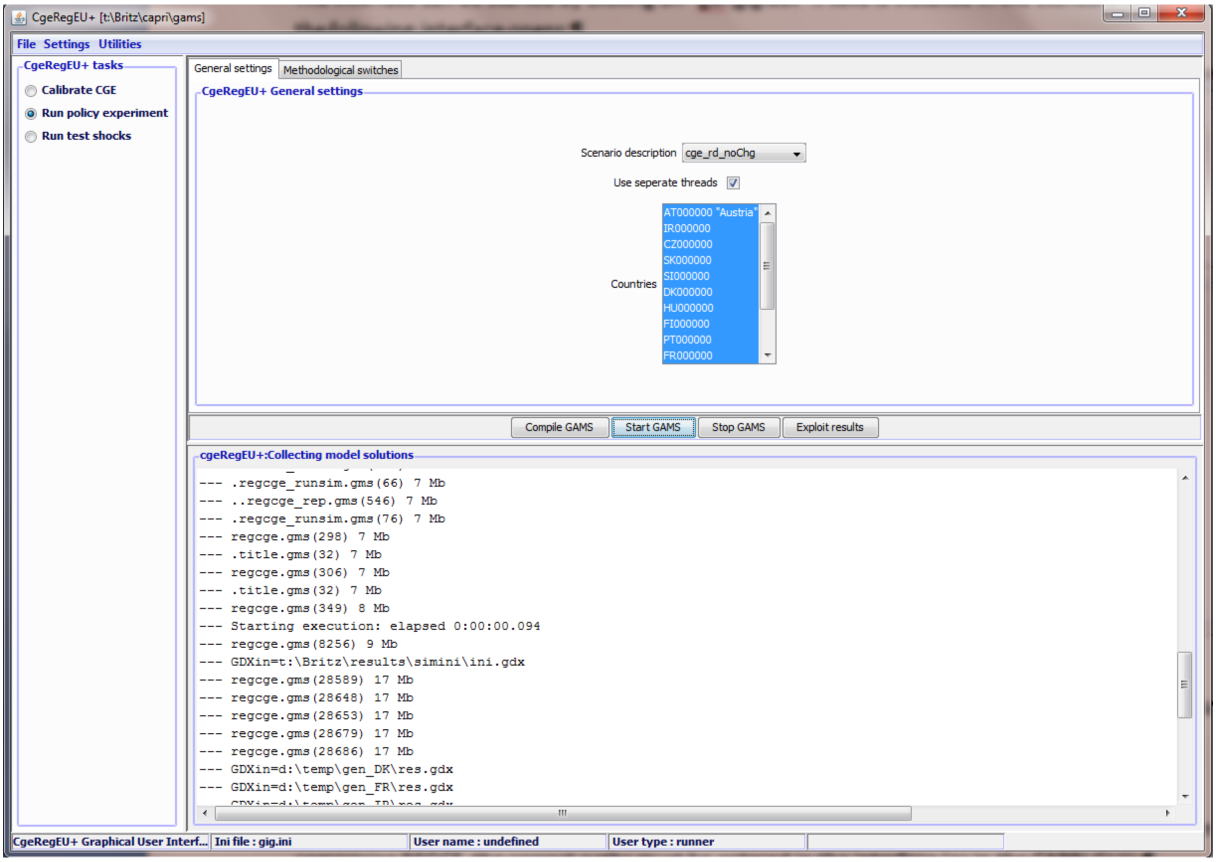
The current interface allows for three tasks:

1. Calibrating the model

2. Running a policy experiment
3. Running a series of pre-defined shocks

After the first scenario definition files are available, a third task to run these will be added.

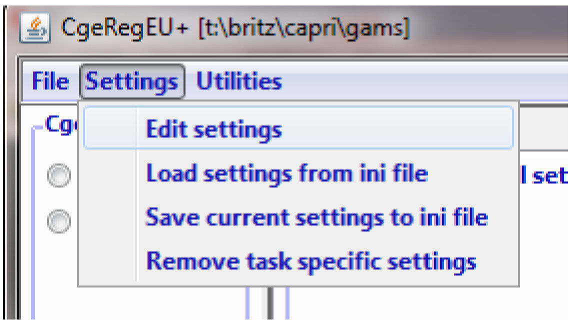
The interface can be started by clicking on “gui\regcge.bat” if Java is installed in the standard directory the following interface opens:

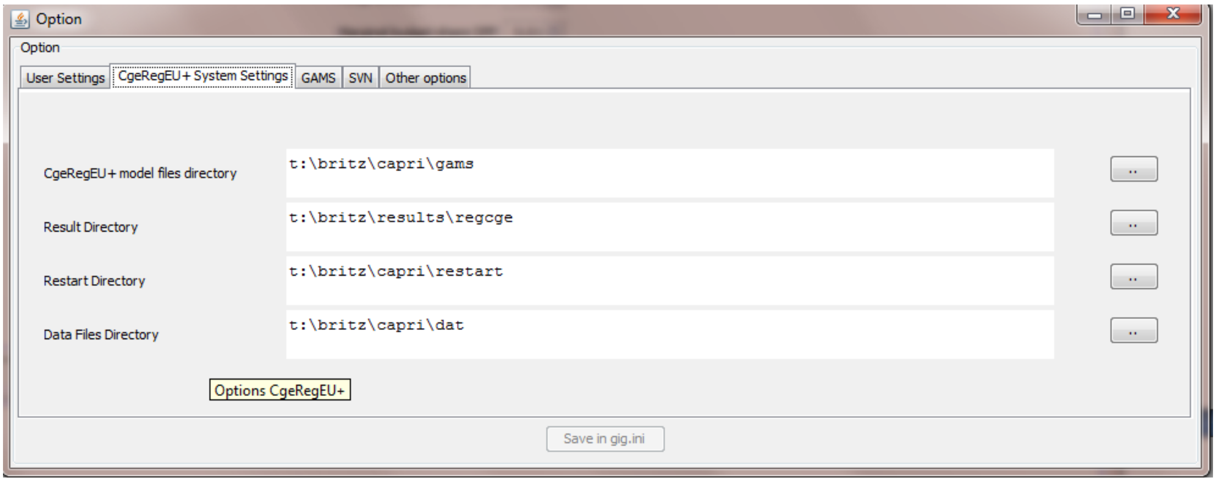


The left hand side allows selecting the task to run. The right hand side is split in three parts: the upper part comprises task specific controls, typically on several panes, the middle part a range of bottom to interact with GAMS or to load results, and the bottom part tracks the output from GAMS.

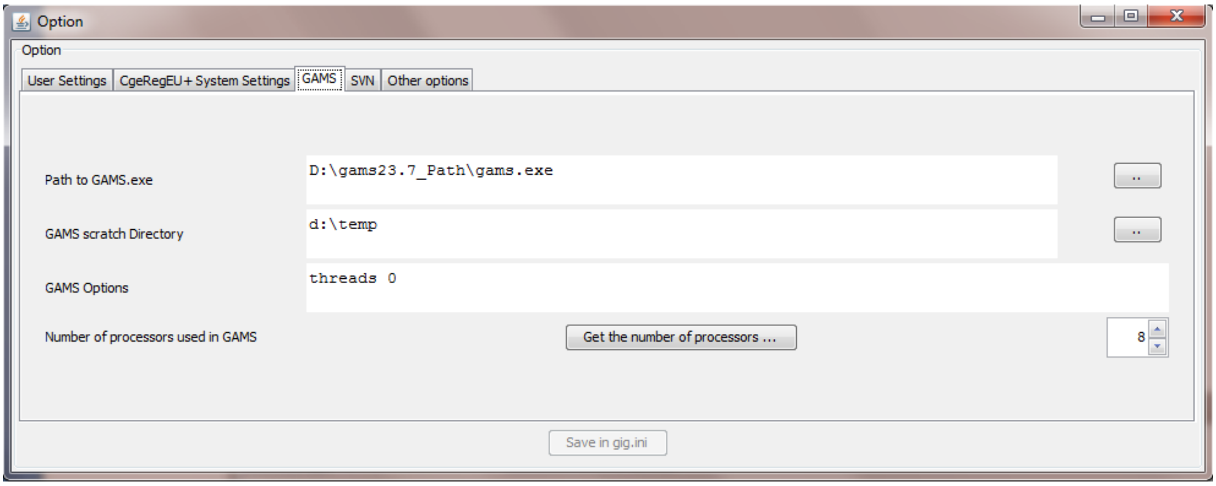
Entering directory information

Before the interface can be used after an update (respectively checkout) of the latest CAPRI version comprising REGCE, the correct paths must be entered in the interface (as in the CAPRI GUI):





Equally, GAMS related setting should be updated:



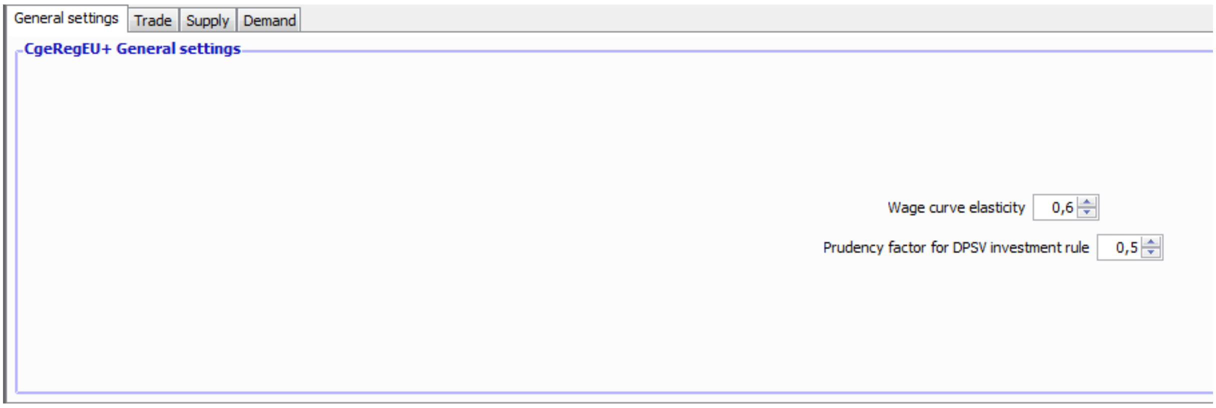
Please note that due to actual (sticky wages) or implicit inequalities (endogenous capital stocks) in the current equation structure, the model cannot be solved in all cases successfully with a NLP solver. If the model runs against these bounds, a MCP solver is necessary. It is possible that the MCP version can be dropped in most production runs where shocks should be generally small enough to avoid running against bounds. By using simpler market structures (see below), the chance to solve the model solely based on NLP is much higher.

Calibrating the model

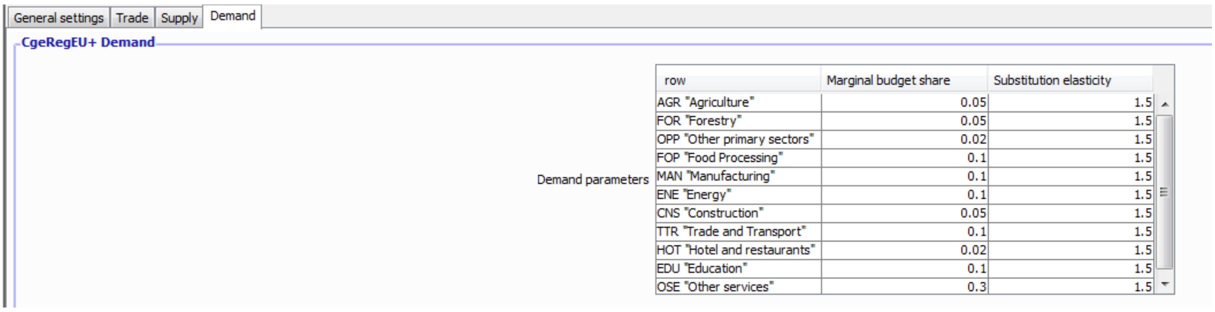
The user can, as seen from below, change the behavioral parameters of the model and, by pressing “start gams”, perform a calibration of the model against the given SAM based on the chosen parameters. The results of the calibration run will stored on disk and automatically loaded when a shock is started.

The following parameters can be set:

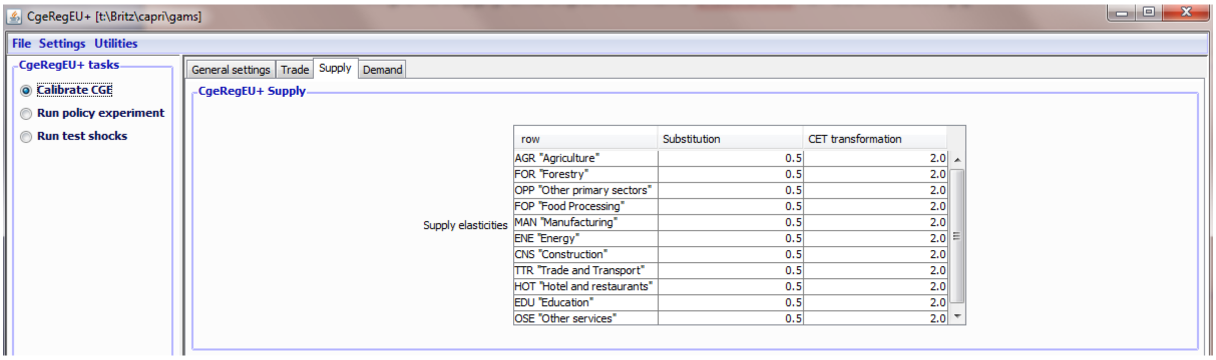
- The *prudency parameter for the DPSV investment rule* and the *wage curve elasticity*



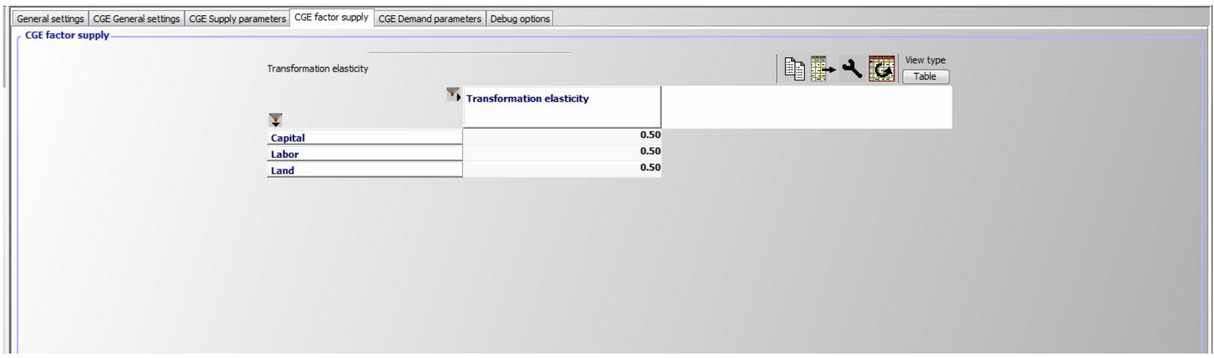
- **Armington substitution elasticities** and *marginal budget shares of final demand* for each industry



- *Primary factor substitution elasticities* and *CET transformation elasticities* for each industry



- *Transformation elasticities between factor (only used if sluggish factor market are used)*

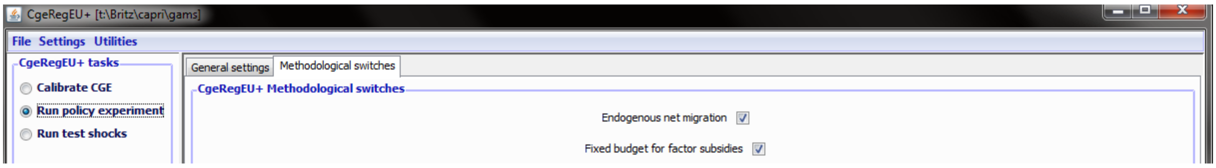


Model variants

Model variants can currently be generated regarding:

- *Net migration*: can be switched off

- *The subsidies paid for factor use can be modulated to exhaust the existing budgets so that total spending does not increase if factor use is expanding in a simulation.*
- *Different market structures for primary factor markets*
- *Different closure rules for the household and government account and the current account balance*

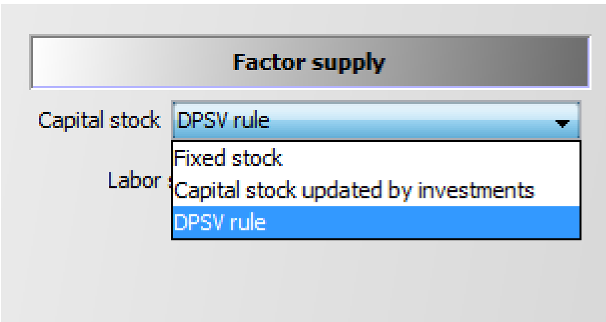


These settings are available both for test shocks and policy scenarios. The reader should note that switching the labor market to "Fully mobile labor with fully employment" will even without any further shock provoke a response of the model against the calibration run. The model is calibrated against a given rate of unemployment.

Options for factor supply

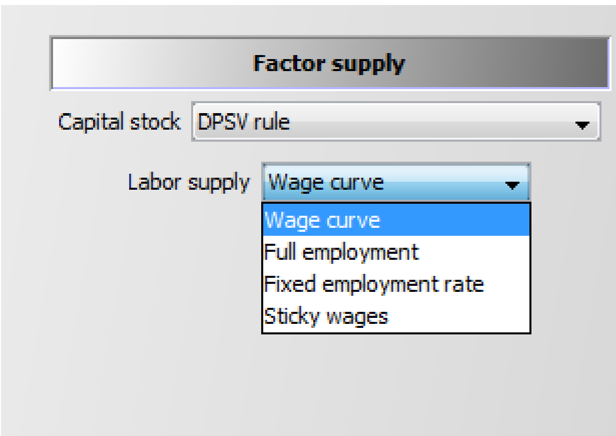
Capital supply / investments

The three options are shown below. They had been discussed above

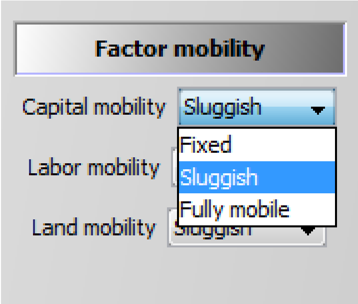


Labor markets

Four different market structures supported as seen below. Sticky wages prevent a drop of the wage rate below the calibration point, thus typically provoking unemployment. The wage curve links the labor force by an elasticity to the relation between the wage rate and the consumer price index, a mechanism argued to be provoked by the negotiation power of the labor unions.



Options for factor mobility

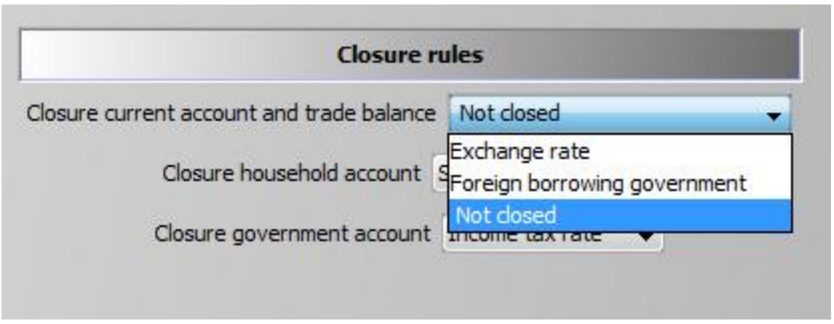


Note: If the DPSV rule is used, the sector specific capital stock is calculated based on that rule, and the choice of factor mobility for capital has no effect.

Model closures

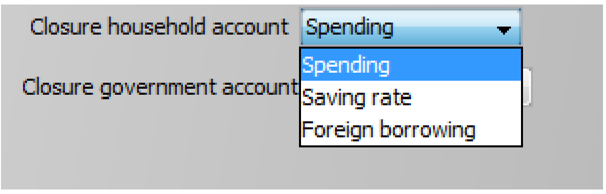
Current account resp. trade balance:

- The *exchange rate* can be used to balance current account and the trade balance.
- Alternatively, *foreign borrowing of the government* can be used
- Alternatively, the two balances can also be kept *detached*, i.e. both the exchange rate and foreign borrowing are fixed.



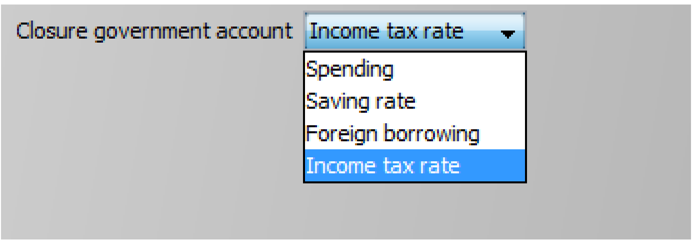
Household account

The household account can either be driven by adjusting spending (= final demand and saving), by adjusting the saving rate at fixed real consumption expenditures, or by adjusting foreign borrowing at fixed real consumption expenditures:



Government account

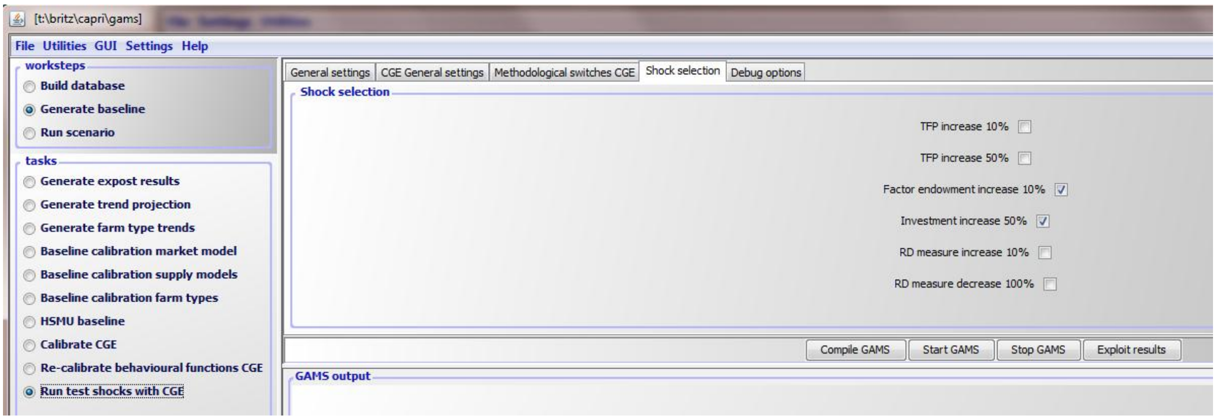
The closures for the government account are shown below. Please note that the foreign borrowing by government cannot be simultaneously used to close both the government and the trade balance. The model will throw an error in that case.



Running test shocks with the model

The GAMS code for a test shocks can be started from the GUI.

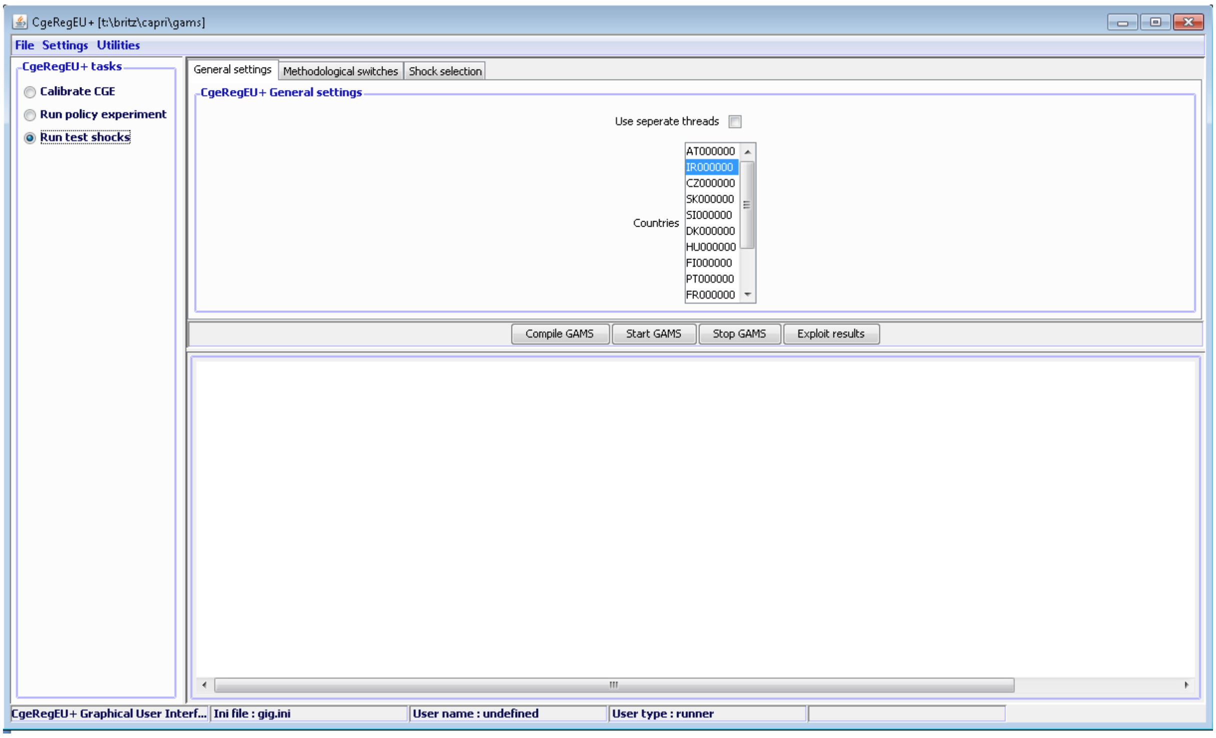
Selecting the chocks



The screenshot above shows the tab to select the shocks to run when the task “Run shocks” is selected. The user might select up to three types of shocks:

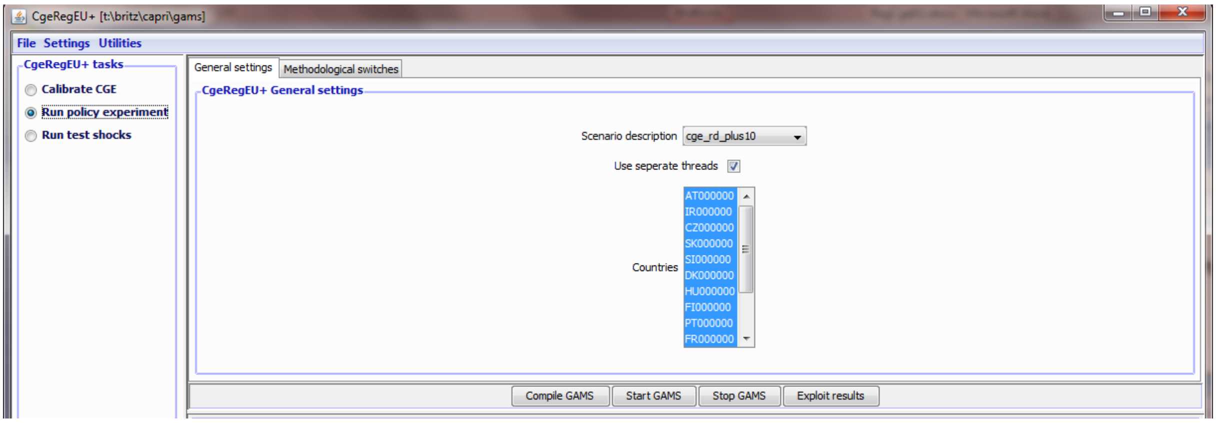
1. *Total factor productivity increases by 10%*, separately for each sector, but simultaneously for all regions
2. *Total factor productivity increases by 50%*, separately for each sector, but simultaneously for all regions
3. *Increasing the factor endowments by 10 %,* separately for each factor, but simultaneously for all regions
4. *An increase in the investments demand by 50%* (if positive) for each sector, but simultaneously for all regions
5. *An shock for each type of RD measure of 10%* of the historic budget spend.

Starting the shocks



The user can select the countries to test. Equally, he might run the model in single or multiple thread mode. The results are always stored as “TestShocks”.

Running a policy experiment



The “Scenario description” field shows the available scenario files from the “pol_input” directory starting with “cge_”. It thus defines

- Which GAMS file is included from the “pol_input” directory, similar to the way the general CAPRI system works. That GAMS file defines the shock to run.
- Equally, the GDX with the results will be stored under that name.

As above, the user can select for which countries the model should run. Equally, he can run the model in single or multiple thread mode.

The policy file is supposed to define a vector of shocks in the definition of the template. The discussion on the RD measures above shows one simple way to do so: one set-up budgets for the different measures (or directly for the shock) category for each region.

Two pre-defined examples are stored:

- One with zero changes to test replication of the base (cge_rdNoChg)
- One with a 10% increase in all RD measures (cge_rd_plus10)

In both cases, it is assumed that the effects of the ex-post measures found in the data on budget spent, including effects on government budgets, are already incorporated in the SAMs and thus the baseline. The zero change example hence looks as follows:

```
$include '%datdir%\policy\rd_spend_2000_2006.gms'
option kill=p_rdSpend;
option kill=actRdShocks;
```

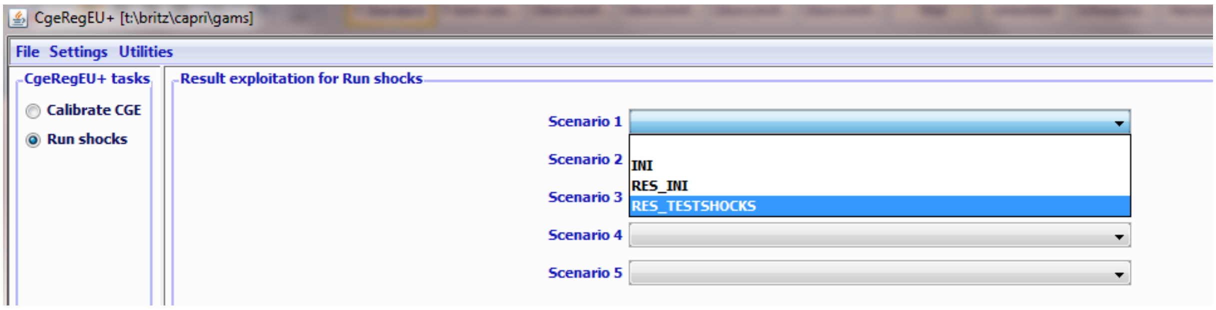
It can be used to check if the model is correctly calibrated. The 10% increase is defined as:

```
$include '%datdir%\policy\rd_spend_2000_2006.gms'

 option kill=actRdShocks;
 actRdShocks(rdShocks) = YES;

p_rdSpend(nutsRD,rdMeasures,rdYears,bud)
 = p_rdSpendOri(nutsRD,rdMeasures,rdYears,bud) * 0.1;

$setglobal tfpFactor 0.20
$setglobal invShare  0.50
```

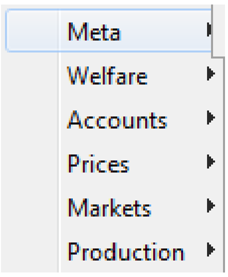
These policy files equally define the share of investment related shocks which decrease the capital tax rate or increase government demand, and the TFP multiplier for such shocks.

Inspecting results

In order to view the results, click on “Exploit results”:



The interface will load the results in the CAPRI exploitation tools, offering a set of tables, grouped as follows:



Generally, the user guide to the CAPRI GUI (http://www.capri-model.org/docs/Gui2011.pdf) can be used to work with the tables. Maps and further information can be added later quite easily. The structure is set up such that the reporting and exploitation part can be later easily integrated in the CAPRI system.

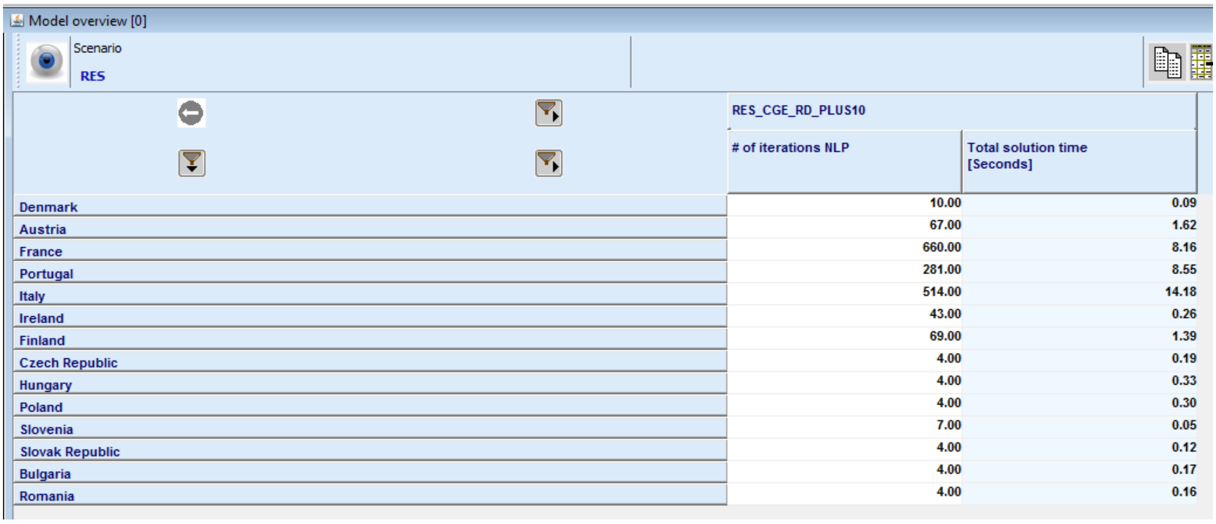
Current results shown in the interface

The reader can expect that the number of tables, their specific content and presentation form to change rapidly as serious application will reveal which information is necessary to analyze the results.

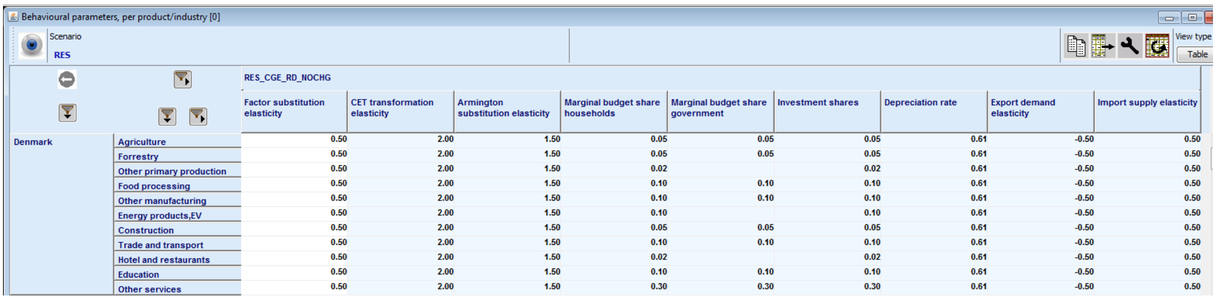
Meta

“**Meta**” shows three tables:

- The first one reports *technical details* (number of iteration, possible infeasibilities etc.) on the model run. A higher sum of infeasibilities in the MCP solution hints at serious problems so that results cannot be used.
- The second one reports most *behavioral parameters* which are sector/commodity specific. At the regional level, the parameters should be normally unchanged between runs which are compared. The only exceptions are sensitivity experiments with identical shocks to analyze how different parameters impact model behavior.
- The third one reports behavioral parameters which are non-sector/commodity specific.



	RES_CGE_RD_PLUS10	
	# of iterations NLP	Total solution time [Seconds]
Denmark	10.00	0.09
Austria	67.00	1.62
France	660.00	8.16
Portugal	281.00	8.55
Italy	514.00	14.18
Ireland	43.00	0.26
Finland	69.00	1.39
Czech Republic	4.00	0.19
Hungary	4.00	0.33
Poland	4.00	0.30
Slovenia	7.00	0.05
Slovak Republic	4.00	0.12
Bulgaria	4.00	0.17
Romania	4.00	0.16

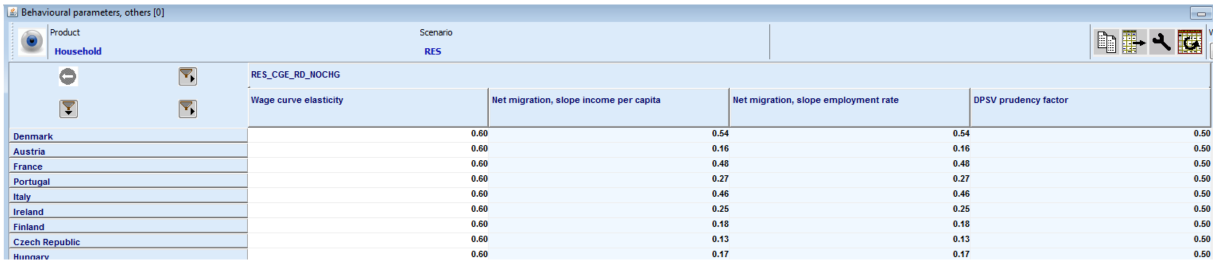


Behavioural parameters, per product/industry [0]

Scenario: RES

RES_CGE_RD_NOCHG

		Factor substitution elasticity	CET transformation elasticity	Armington substitution elasticity	Marginal budget share households	Marginal budget share government	Investment shares	Depreciation rate	Export demand elasticity	Import supply elasticity
Denmark	Agriculture	0.50	2.00	1.50	0.05	0.05	0.05	0.61	-0.50	0.50
	Forrestry	0.50	2.00	1.50	0.05	0.05	0.05	0.61	-0.50	0.50
	Other primary production	0.50	2.00	1.50	0.02		0.02	0.61	-0.50	0.50
	Food processing	0.50	2.00	1.50	0.10	0.10	0.10	0.61	-0.50	0.50
	Other manufacturing	0.50	2.00	1.50	0.10	0.10	0.10	0.61	-0.50	0.50
	Energy products,EV	0.50	2.00	1.50	0.10		0.10	0.61	-0.50	0.50
	Construction	0.50	2.00	1.50	0.05	0.05	0.05	0.61	-0.50	0.50
	Trade and transport	0.50	2.00	1.50	0.10	0.10	0.10	0.61	-0.50	0.50
	Hotel and restaurants	0.50	2.00	1.50	0.02		0.02	0.61	-0.50	0.50
	Education	0.50	2.00	1.50	0.10	0.10	0.10	0.61	-0.50	0.50
	Other services	0.50	2.00	1.50	0.30	0.30	0.30	0.61	-0.50	0.50



Behavioural parameters, others [0]

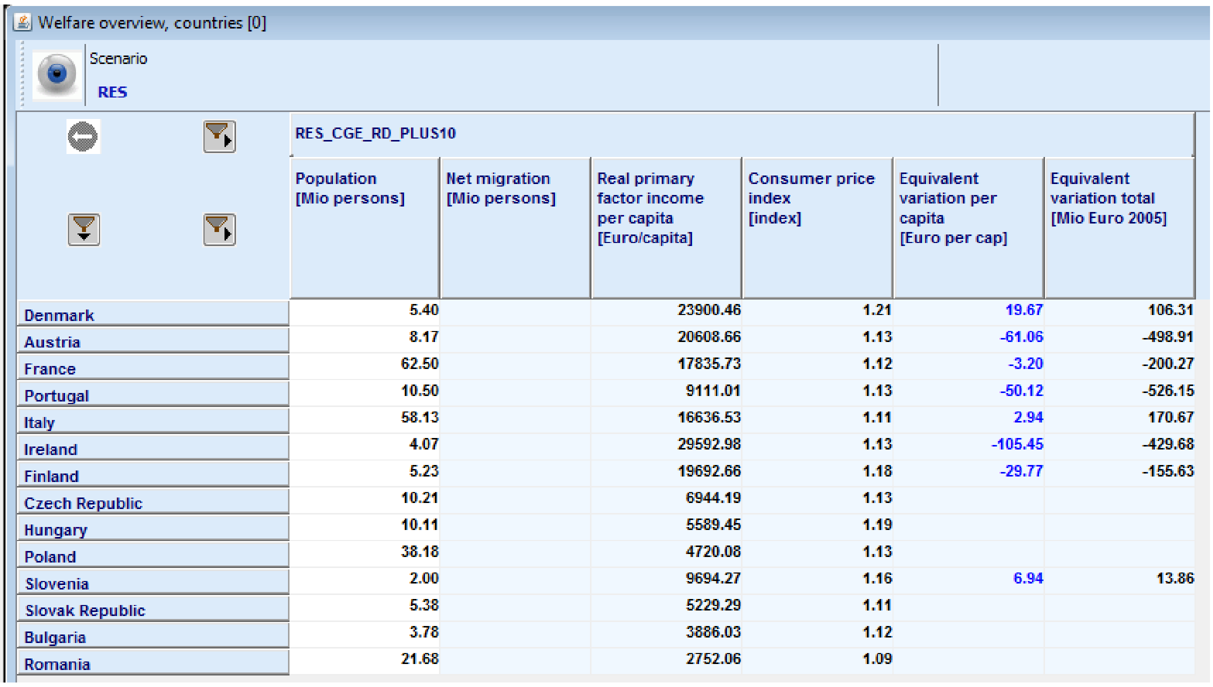
Product: Household — Scenario: RES

RES_CGE_RD_NOCHG

	Wage curve elasticity	Net migration, slope income per capita	Net migration, slope employment rate	DPSV prudency factor
Denmark	0.60	0.54	0.54	0.50
Austria	0.60	0.16	0.16	0.50
France	0.60	0.48	0.48	0.50
Portugal	0.60	0.27	0.27	0.50
Italy	0.60	0.46	0.46	0.50
Ireland	0.60	0.25	0.25	0.50
Finland	0.60	0.18	0.18	0.50
Czech Republic	0.60	0.13	0.13	0.50
Hungary	0.60	0.17	0.17	0.50

Welfare analysis

The welfare analysis reports the population, net migration, real factor income per capita, the consumer price index the EV in total or per capita by country or for all regions:



Welfare overview, countries [0]

Scenario: RES

RES_CGE_RD_PLUS10

	Population [Mio persons]	Net migration [Mio persons]	Real primary factor income per capita [Euro/capita]	Consumer price index [index]	Equivalent variation per capita [Euro per cap]	Equivalent variation total [Mio Euro 2005]
Denmark	5.40		23900.46	1.21	19.67	106.31
Austria	8.17		20608.66	1.13	-61.06	-498.91
France	62.50		17835.73	1.12	-3.20	-200.27
Portugal	10.50		9111.01	1.13	-50.12	-526.15
Italy	58.13		16636.53	1.11	2.94	170.67
Ireland	4.07		29592.98	1.13	-105.45	-429.68
Finland	5.23		19692.66	1.18	-29.77	-155.63
Czech Republic	10.21		6944.19	1.13		
Hungary	10.11		5589.45	1.19		
Poland	38.18		4720.08	1.13		
Slovenia	2.00		9694.27	1.16	6.94	13.86
Slovak Republic	5.38		5229.29	1.11		
Bulgaria	3.78		3886.03	1.12		
Romania	21.68		2752.06	1.09		

Welfare overview [0]

Scenario: RES

RES_CGE_RD_PLUS10

	Population [Mio persons]	Net migration [Mio persons]	Equivalent variation per capita [Euro per cap]	Equivalent variation total [Mio Euro 2005]
Denmark	5.40		1.30	7.05
Austria	8.17		-164.48	-1344.00
France	62.50		-365.22	-22824.49
Portugal	10.50		-560.28	-5881.50
Italy	58.13		-187.38	-10892.49
Ireland	4.07		29.80	121.41
Finland	5.23		-180.58	-944.15
Czech Republic	10.21			
Hungary	10.11			
Poland	38.18			
Slovenia	2.00		2.22	4.44
Slovak Republic	5.38			
Bulgaria	3.78			
Romania	21.68			
Ile de france	11.82	0.11	-388.75	-4596.76
Champagne-Ardenne	1.37	-0.01	-338.61	-463.61
Picardie	1.93	-0.00	-244.87	-473.31
Haute-Normandie	1.86	0.00	-177.97	-331.52
Centre	2.57	-0.00	-295.98	-759.33
Basse-Normandie	1.49	-0.00	-283.92	-422.54
Bourgogne	1.68	0.01	-256.71	-431.09
Nord-Pas-De-Calais	4.16	0.00	-244.79	-1017.72
Lorraine	2.41	0.00	-179.13	-431.03
Alsace	1.86	0.01	-82.39	-153.34
Franche-Comte	1.17	-0.01	-218.51	-254.62

and offers a decomposition by price and factor quantities / prices of the EV:

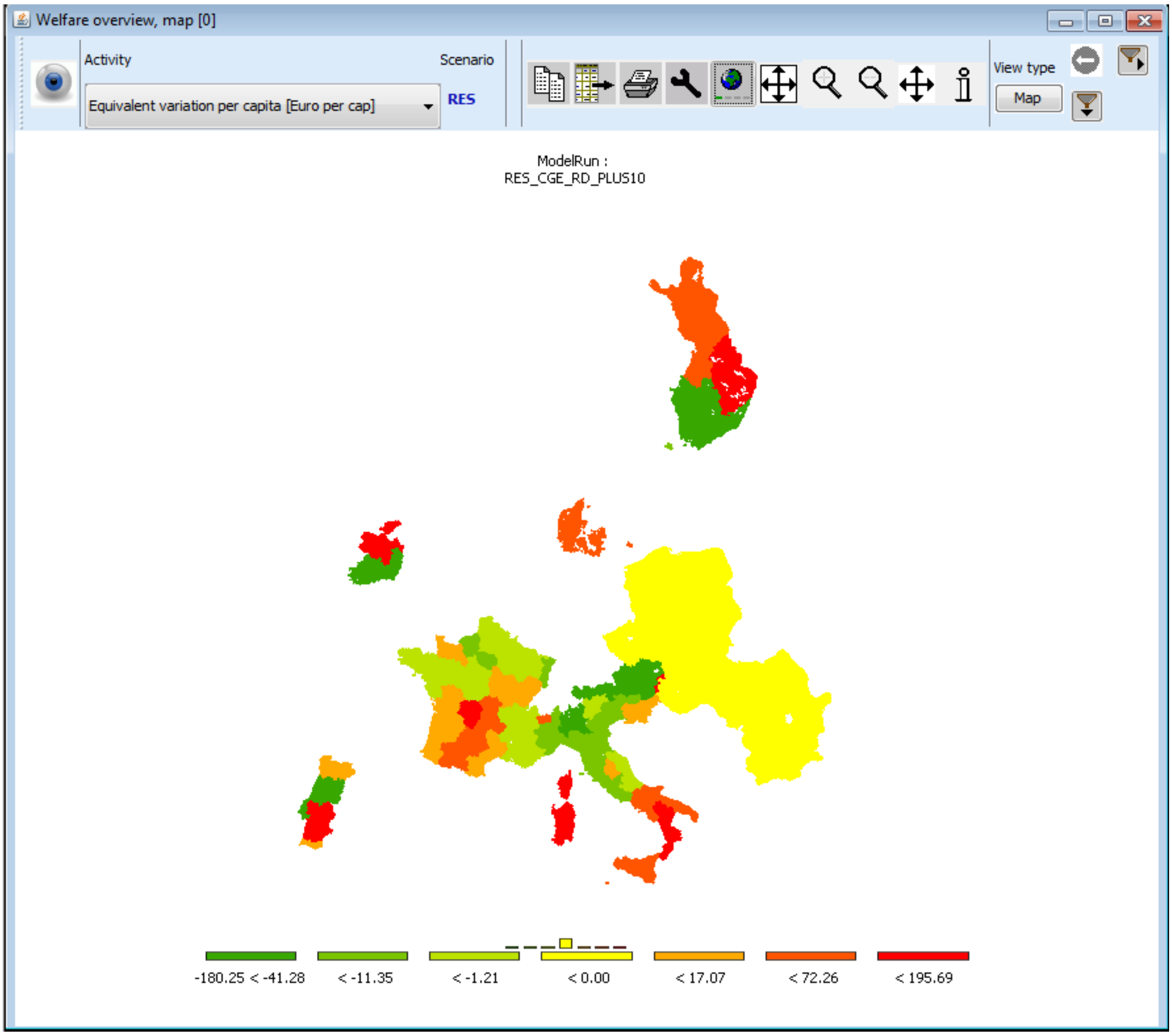
EV decomposition [0]

Region: Denmark | Activity: Equivalent variation per capita [Euro per cap] | Scenario: RES

RES_CGE_RD_PLUS10

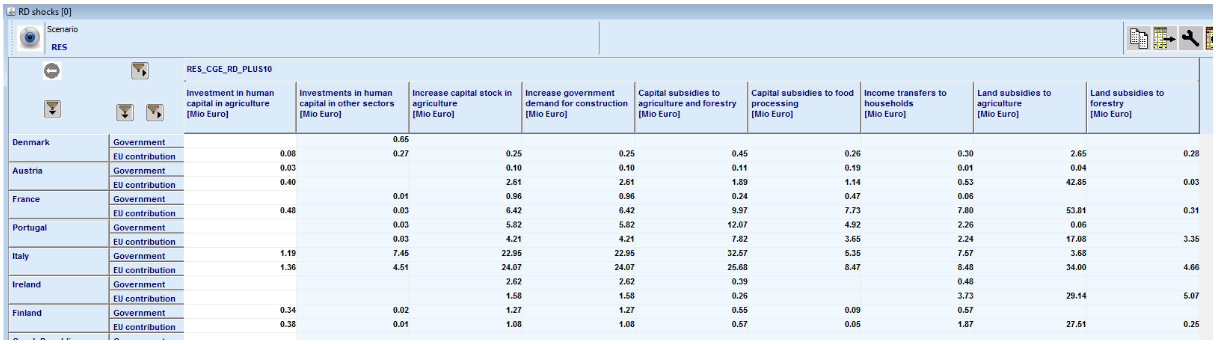
	price	quant	value
Agriculture	0.00		
Forrestry	0.70		
Other primary production	0.70		
Food processing	0.69		
Other manufacturing	0.68		
Energy products,EV	0.67		
Construction	0.65		
Trade and transport	0.60		
Hotel and restaurants	0.59		
Education	0.71		
Other services	0.61		
Capital	-0.09	0.90	0.81
Labor	-0.10	0.44	0.31
Land	0.00	0.00	0.00

Equally, maps for certain items can be drawn:



Policy and shocks

These groups reports the absolute changes introduced by the RD shocks:



RD shocks [0]

Scenario: RES

RES_CGE_RD_PLUS10

		Investment in human capital in agriculture [Mio Euro]	Investments in human capital in other sectors [Mio Euro]	Increase capital stock in agriculture [Mio Euro]	Increase government demand for construction [Mio Euro]	Capital subsidies to agriculture and forestry [Mio Euro]	Capital subsidies to food processing [Mio Euro]	Income transfers to households [Mio Euro]	Land subsidies to agriculture [Mio Euro]	Land subsidies to forestry [Mio Euro]
Denmark	Government		0.65							
	EU contribution	0.08	0.27	0.25	0.25	0.45	0.26	0.30	2.65	0.28
Austria	Government	0.03		0.10	0.10	0.11	0.19	0.01	0.04	
	EU contribution	0.40		2.61	2.61	1.89	1.14	0.53	42.85	0.03
France	Government		0.01	0.96	0.96	0.24	0.47	0.06		
	EU contribution	0.48	0.03	6.42	6.42	9.97	7.73	7.80	53.81	0.31
Portugal	Government		0.03	5.82	5.82	12.07	4.92	2.26	0.06	
	EU contribution		0.03	4.21	4.21	7.82	3.65	2.24	17.08	3.35
Italy	Government	1.19	7.45	22.95	22.95	32.57	5.35	7.57	3.68	
	EU contribution	1.36	4.51	24.07	24.07	25.68	8.47	8.48	34.00	4.66
Ireland	Government			2.62	2.62	0.39		0.48		
	EU contribution			1.58	1.58	0.26		3.73	29.14	5.07
Finland	Government	0.34	0.02	1.27	1.27	0.55	0.09	0.57		
	EU contribution	0.38	0.01	1.08	1.08	0.57	0.05	1.87	27.51	0.25

And they related RD measures:

RD measures [0]

Scenario: RES

RES_CGE_RD_PLUS10

		Vocational training, information actions [Mio Euro]	Setting up young farmers [Mio Euro]	Early retirement of farmers and farm workers [Mio Euro]	Setting up farm management, farm relief and farm/forestry advisory services [Mio Euro]	Farm modernisation [Mio Euro]	Adding value to agricultural and forestry products [Mio Euro]	Improving and developing infrastructure related to the development and adaptation of agriculture and forestry [Mio Euro]	Restoring agr. production potential damaged by natural disasters and introducing appropriate prevention actions [Mio Euro]	Helping farmers to adapt to demanding standards based on Community legislation [Mio Euro]	LFA, Natura 2000 [Mio Euro]	Agri-environment payments [Mio Euro]	Non-productive investments [Mio Euro]	First afforestation of agricultural land [Mio Euro]	Diversi non-agi activitie [Mio Eu
Denmark	Government														
	EU contributi	0.02	0.26	0.03		0.25	0.26	0.02			0.07	2.28	0.29	0.28	
Austria	Government	0.03	0.01			0.10	0.19	0.03					0.04		
	EU contributi	0.40	0.52			2.61	1.14	0.89			10.86	31.78	0.21	0.03	
France	Government		0.06			0.96	0.47	0.00							
	EU contributi	0.32	6.65	1.15	0.16	6.42	7.73	3.84	0.36		24.98	25.26	3.57	0.31	
Portugal	Government		2.26			5.82	4.92	4.06	0.94				0.06		
	EU contributi		1.64	0.60		4.21	3.65	2.60	0.70		7.84	9.20	0.04	3.35	
Italy	Government	0.93	7.57		0.25	22.95	5.35	23.56	0.10				3.68		
	EU contributi	1.03	8.21	0.27	0.32	24.07	8.47	17.58	0.16		7.28	23.35	3.37	4.66	

Accounts

The goup "**Accounts**" offers account for the balance of trade, households, the central and local government:

Household account [0]

Product: Household — ModelRun: RES_TESTSHOCKS

CAL

	Primary factor income [Mio EU 2005]	Taxes minus subsidies [Mio EU 2005]	Net borrowing [Mio EU 2005]	Income [Mio EU 2005]	Savings [Mio EU 2005]	Consumption [Mio EU 2005]
Ireland	138368.89	10259.28	-28534.93	120093.24	39732.68	80360.57
Border, Midlands and Western	25374.39	1881.86	-5225.65	22030.60	7288.16	14742.44
Southern and Eastern	112994.50	8377.42	-23309.28	98062.65	32444.51	65618.13

Balance of trade [0]

ModelRun: RES_TESTSHOCKS

CAL

	Balance of trade [Mio EU 2005 in int currency]	Exchange rate [Int currency/EU]	Current account balance [Mio EU 2005]	Net borrowing government [Mio EU 2005]	Net borrowing household [Mio EU 2005]	Ba tr [M in cu
Ireland	19608.35	1.00	-19608.35	8926.58	-28534.93	

National government account [0]

Product: Government — ModelRun: RES_TESTSHOCKS

CAL

	Capital taxes [Mio EU 2005]	Labor taxes [Mio EU 2005]	Land taxes [Mio EU 2005]	National income tax [Mio EU 2005]	Production taxes [Mio EU 2005]	Sales taxes [Mio EU 2005]	Investment taxes [Mio EU 2005]	Net borrowing [Mio EU 2005]	Subsidies to regional governments [Mio EU 2005]
Ireland		4899.71	-1638.69	14004.59				8926.58	47592.99

Local government account [0]

Product: Government — ModelRun: RES_TESTSHOCI

CAL

	Subsidies from national government [Mio EU 2005]	Subsidies to local households [Mio EU 2005]	Savings [Mio EU 2005]	Final consumption [Mio EU 2005]
Ireland	47592.99		1416.09	21913.03

The results for the household are also available on a per capita basis.

Prices

The group "**prices**" offers tables relating to the industries (product prices), factors in average of the region and factor prices by industry:

Product prices [0]

Region: Denmark — ModelRun: RES_TESTSHOCKS — CAL

	Producer price [index]	Value-Added price [index]	Per unit Tax [index]	Regional price [index]	National price [index]	Export price [index, in Euro]	Import price [index, ine Euro]	Armington price [index]
Agriculture	1.00	0.37	-0.07	1.00	1.00	1.00	1.00	1.00
Forrestry	1.00	0.48	-0.04	1.00	1.00	1.00	1.00	1.00
Other primary	1.00	0.83	0.04	1.00	1.00	1.00	1.00	1.00
Food processing	1.00	0.25	0.01	1.00	1.00	1.00	1.00	1.00
Other manufacturi	1.00	0.37	0.00	1.00	1.00	1.00	1.00	1.00
Energy products	1.00	0.36	0.00	1.00	1.00	1.00	1.00	1.00
Construction	1.00	0.38	0.01	1.00	1.00	1.00	1.00	1.00
Trade and transport	1.00	0.43	0.01	1.00	1.00	1.00	1.00	1.00
Hotel and restaurants	1.00	0.44	0.04	1.00	1.00	1.00	1.00	1.00
Education	1.00	0.74	0.03	1.00	1.00	1.00	1.00	1.00
Other services	1.00	0.62	0.04	1.00	1.00	1.00	1.00	1.00

Factor prices, averages [0]

Region: Denmark — CAL

	Price [index]
Capital	1.00
Labor	1.00
Land	1.00

Factor prices, per industry [0]

Region: Denmark — CAL

	Capital [index]	Labor [index]	Land [index]
Agriculture	1.00	1.00	1.00
Forrestry	1.00	1.00	1.00
Other primary	1.00	1.00	1.00
Food processing	1.00	1.00	1.00
Other manufacturi	1.00	1.00	1.00
Energy products	1.00	1.00	1.00
Construction	1.00	1.00	1.00
Trade and transport	1.00	1.00	1.00
Hotel and restaurants	1.00	1.00	1.00
Education	1.00	1.00	1.00
Other services	1.00	1.00	1.00

Factor taxes, per industry [0]

Region: Denmark — dim5: taxrate — Sce RE — RES_CGE_RD_PLUS10

	Capital [index]	Labor [index]	Land [index]
Agriculture	-0.00	0.08	
Forrestry	-0.00	0.11	
Other primary	-0.00	0.21	
Food processing	-0.00	0.38	
Other manufacturing	-0.00	0.37	
Energy products,EV	-0.00	0.37	
Construction	-0.00	0.09	
Trade and transport	-0.00	0.04	
Hotel and restaurants	-0.00	0.02	
Education	-0.00	0.11	
Other services	-0.00	0.10	

Markets

The tables grouped under “**market**” report under “**production**” production quantities and where they are sold.

Production [0]

Region: Denmark — dim5: quant — ModelRun: RES_INI — CAL

	Production [Mio Euro 2005]	Regional sales [Mio Euro 2005]	National sales [Mio Euro 2005]	Exports [Mio Euro 2005]
Agriculture	7331.72	275.56	2433.95	1289.94
Forrestry	416.16	113.73	113.84	87.01
Other primary production	6822.76	56.39	1098.03	4066.90
Food processing	14167.89	2460.69	2055.64	7682.14
Other manufacturing	43778.16	4390.88	8198.07	20358.93
Energy products	7947.73	1895.34	1901.79	2120.71
Construction	24320.07	14415.86	5950.99	160.84
Trade and transport	85253.63	21327.75	12906.27	35399.94
Hotel and restaurants	5917.26	2859.91	1031.56	934.77
Education	13685.81	11534.14	1618.64	15.82
Other services	129120.33	70204.56	27988.88	4914.90

The final demand composition is shown under “**final demand**”:

Final demand [0]

Region: Denmark | dim5: quant | ModelRun: RES_INI

CAL

	Final demand [Mio Euro 2005]	Household demand [Mio Euro 2005]	Government demand [Mio Euro 2005]	Investment demand [Mio Euro 2005]
Agriculture	433.83	306.74	79.54	47.54
Forrestry	189.98	73.74	22.07	94.17
Other primary production	88.55	33.55		55.00
Food processing	3651.67	3369.46	10.88	271.33
Other manufacturing	6749.75	2277.65	96.04	4376.06
Energy products	2817.64	2283.12		534.52
Construction	22128.63	596.10	884.89	20647.65
Trade and transport	29870.74	24655.86	386.85	4828.02
Hotel and restaurants	3986.07	3980.51		5.56
Education	13164.91	968.29	12144.49	52.13
Other services	89881.59	43292.10	41010.77	5578.73

Whereas its origins are reported under “**armington**”:

Armington [0]

Region: Denmark | dim5: quant | ModelRun: RES_INI

CAL

	Final demand [Mio Euro 2005, util]	Regional purchases [Mio Euro 2005]	National purchases [Mio Euro 2005]	Imports [Mio Euro 2005]
Agriculture	433.83	275.56	82.49	75.78
Forrestry	189.98	113.73	38.28	37.97
Other primary production	88.55	56.39	13.95	18.21
Food processing	3651.67	2460.69	506.58	684.40
Other manufacturing	6749.75	4390.88	957.71	1401.16
Energy products	2817.64	1895.34	383.57	538.73
Construction	22128.63	14415.86	3046.46	4666.31
Trade and transport	29870.74	21327.75	3306.00	5236.98
Hotel and restaurants	3986.07	2859.91	444.35	681.81
Education	13164.91	11534.14	1325.51	305.26
Other services	89881.59	70204.56	10298.30	9378.73

The last table in that group reports the “**primary factors**”:

Primary factors [0]

Region: Denmark | ModelRun: RES_INI

CAL

	Use [Mio Euro 2005]	Stock [Mio Euro 2005]	Utilization rate [%]
Capital	59736.41	59736.41	100.00
Labor	97953.30	108836.99	90.00
Land			

Production sectors

The first table report “**primary factor use by industry**”:

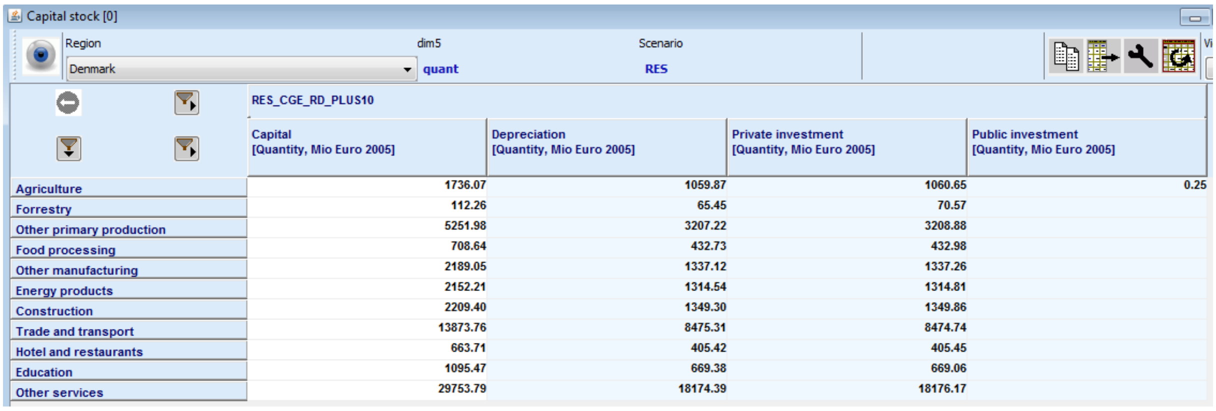
Pimary factor use by industry [0]

Region: Denmark | dim5: quant | ModelRun: RES_INI

CAL

	Capital [index]	Labor [index]	Land [index]
Agriculture	1735.04	885.67	
Forrestry	107.14	83.56	
Other primary production	5250.32	341.14	
Food processing	708.39	2019.13	
Other manufacturing	2188.91	10161.86	
Energy products	2151.95	516.76	
Construction	2208.84	6461.14	
Trade and transport	13874.34	22034.19	
Hotel and restaurants	663.68	1889.17	
Education	1095.79	8105.06	
Other services	29752.01	45455.62	

Followed by one on the **capital stocks**



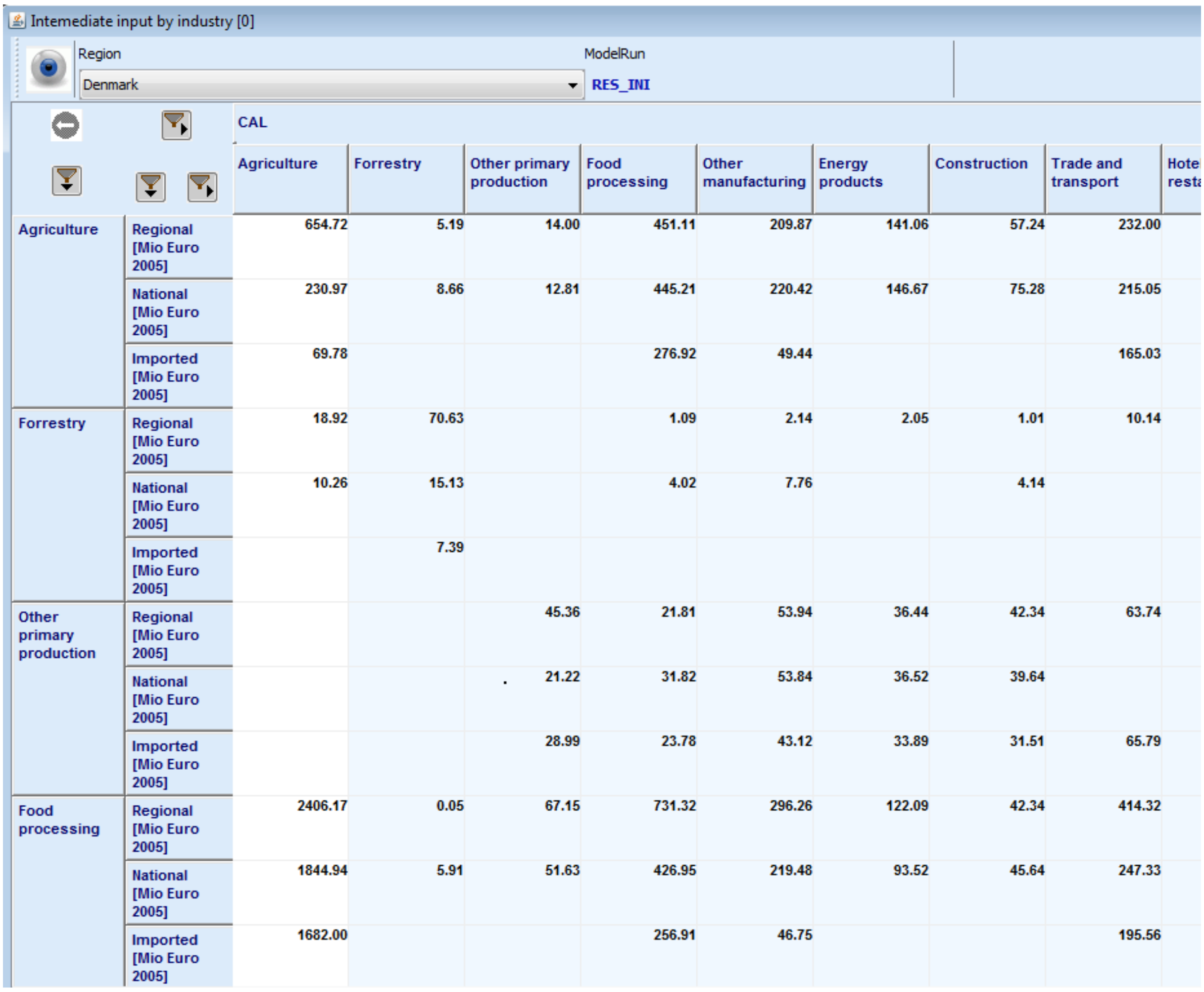
Capital stock [0]

Region: Denmark | dim5: quant | Scenario: RES

RES_CGE_RD_PLUS10

	Capital [Quantity, Mio Euro 2005]	Depreciation [Quantity, Mio Euro 2005]	Private investment [Quantity, Mio Euro 2005]	Public investment [Quantity, Mio Euro 2005]
Agriculture	1736.07	1059.87	1060.65	0.25
Forrestry	112.26	65.45	70.57	
Other primary production	5251.98	3207.22	3208.88	
Food processing	708.64	432.73	432.98	
Other manufacturing	2189.05	1337.12	1337.26	
Energy products	2152.21	1314.54	1314.81	
Construction	2209.40	1349.30	1349.86	
Trade and transport	13873.76	8475.31	8474.74	
Hotel and restaurants	663.71	405.42	405.45	
Education	1095.47	669.38	669.06	
Other services	29753.79	18174.39	18176.17	

The third one shows **intermediate input use**:



Intemediate input by industry [0]

Region: Denmark | ModelRun: RES_INI

CAL

		Agriculture	Forrestry	Other primary production	Food processing	Other manufacturing	Energy products	Construction	Trade and transport	Hote resta
Agriculture	Regional [Mio Euro 2005]	654.72	5.19	14.00	451.11	209.87	141.06	57.24	232.00	
	National [Mio Euro 2005]	230.97	8.66	12.81	445.21	220.42	146.67	75.28	215.05	
	Imported [Mio Euro 2005]	69.78			276.92	49.44			165.03	
Forrestry	Regional [Mio Euro 2005]	18.92	70.63		1.09	2.14	2.05	1.01	10.14	
	National [Mio Euro 2005]	10.26	15.13		4.02	7.76		4.14		
	Imported [Mio Euro 2005]		7.39							
Other primary production	Regional [Mio Euro 2005]			45.36	21.81	53.94	36.44	42.34	63.74	
	National [Mio Euro 2005]			21.22	31.82	53.84	36.52	39.64		
	Imported [Mio Euro 2005]			28.99	23.78	43.12	33.89	31.51	65.79	
Food processing	Regional [Mio Euro 2005]	2406.17	0.05	67.15	731.32	296.26	122.09	42.34	414.32	
	National [Mio Euro 2005]	1844.94	5.91	51.63	426.95	219.48	93.52	45.64	247.33	
	Imported [Mio Euro 2005]	1682.00			256.91	46.75			195.56	