1. The structure of the supply model in CAPRI

The supply model

- Structure of GAMS file capmod.gms
  - General structure
  - Important sections and key files
  - Two modus for capmod with and without market model
- The suppy_model.gms file
  - Equations, Variable , Parameters, Sets

Set definition

```
*  --- include settings generated by JAVA GUI
*
$INCLUDE 'fortran.gms'

*  --- read sets used generally in CAPRI project
*
$include 'sets.gms'

$include 'capmod\define_regional_sets.gms'

$ifi %generateSimIni%==yes $include 'capmod\create_sim_ini_gdx.gms'
$ifi not %generateSimIni%==yes $include 'capmod\load_sim_ini_gdx.gms'

LOOP(countries,
   --- set dynamically the gdx file name to load
   put_utilities batch 'gdxin' / '%results_in%\capreg\res_%BAS%' countries.tl:2 '%addfile%.gdx';
   execute_loadpoint DATA2;

...ends_%BAS%%SIM%%addFile%%VersionCapTrdFarm%%inputDataId%.gdx

...in%\baseline\%trendFileName%" DATA2;

$include 'pol_input\%subfolder%\%result_type%.gms'
$          include 'supply\supply_model.gms';

$          include 'capmod\def_fert_and_requirements.gms'
```

- Settings from the GUI
- Set definition for common, regions, market, policy and report
- Load data from base year (capreg by country) and trend from baseline folder
- Reads in trend and base year data and store in simini.gdx
- Read ll data from simini.gdx (fast way to do it)
- Incudes the policy file
- Definition of the supply model
- Sets the feed module related parameters, Also translates the changes of our scenarios into the parameters (p_animReq, p_animProdDays)

Iteration loop

```
LOOP ( STEP $ ( (step.pos eq CARD(STEP)) or (p_iterChg("TOT","MAX","CUR") GT p_minIterChg)),

         putclose caplog // "*** CAPRI enters simulation loop Nr. ", step.pos, " of " p_runInfo("NSTEPS")  /;

*            --- define weights for past iterations
*
$           include 'capmod\define_stepWeights.gms'

$           ifthenI.supp %BASELINE% == on
$                    INCLUDE 'supply\calibrate_supply.gms'
*
$           else.supp

*
*                    Solve supplly models
*
*
                     option kill= debugregions;
*                    debugregions("RO030000")=yes;

$                    iftheni.thread1 %threads% == off
$                       include 'supply\simu_supply.gms'
$                    else.thread1
$                       include 'supply\simu_supply_parallel.gms'
$                    endif.thread1

*
*                    Reporting for price sensitivity experiments
*
*
$                    include 'pricescen\reporting.gms'

                     p_timElap(SIMY,STEP,"SupplyModels")   = p_timElap(SIMY,STEP,"SupplyModels") + TimeElapsed - p_timElapLst;
                     p_timElapLst = TimeElapsed;

$           endif.supp

*            --- update premiums based on results from regional models during 10 first iterations
*                (cut premiums if base areas overshot, and build a weighted average of premiums
*                over iterations)
$            batinclude 'capmod\calculate_premiums.gms' STEP

*            --- Solve global spatial market model with behavioural functions for supply and feed
$            ifI %MARKET_M% == on      $INCLUDE 'arm\simu_market.gms'
*
*            --- call dummy model with solvelink=1 to clean up memory
*
*            --- Define current prices, and set activity levells for current iteration
*                calcualte weighted prices to be used in next iteration, and calculate relative changes
*                in prices, production and feed use quantities, to check for convergence
$            include 'capmod\price_forecast.gms'
*
*               --- adjust yields to current prices, and update IO-coefficients
$            include 'supply\endog_yields.gms'
$            batinclude 'capmod\step_report.gms'
*            --- end of iteration loop
)
$ifi %REPORTS% == on $include 'capmod\reports.gms'
```

- Set PMP parameter under calibration
- Reports elasticities
- Runs supply models in a loop
- Runs supply models as separate parallel threads
- Adjust “PREM” in case new production program exceeds ceilings
- Run for new supply at EU the market model to get the price feedback
- Prepare for new supply run

```
$include 'supply\def_supply_model_par.gms';

  @purpose  : Define sets, parameters, variables etc. used in supply model
--------------------------------------------------------------------------------
    PARAMETER

--------------------------------------------------------------------------------
    p_linObjeCont(RALL,PACT,A)                "Costs and revenues for activities not covered by constraints"
    p_nitrBalance(*,*)                        "Nitrogen balance parameter"
    p_nitrFact                                "Nitrogen balance parameter"
    p_NVZshare(Rall,*)                        "Shares of NVZ area in total and implementation shares for balanced fertilisation"
*
    p_maxFeedShare(RALL,PACT,A,FEED %addtimedim_ast%)        "Maximum shares dry matter intake for each feedingstuff"
    p_minFeedShare(RALL,PACT,A,FEED %addtimedim_ast%)        "Minimum shares dry matter intake for each feedingstuff"
    p_animProdDays                            "Days per year in production system for animal activities"
*
    p_minShareMinFert(RALL,PACT,*,FNUT)       "Minimum share of mineral on total fertilizer input"
    p_maxShareMinFert(RALL,PACT,*,FNUT)       "Maximum share of mineral on total fertilizer input"
    p_nutContCropOutput(O,FNUT)               "Nutrient retention from harvested material"
*
    p_feedQuant(*,* %addtimedim_ast%)                        "Ammount of feed use in current aggregate to trim"
*
*TJ TREND
    p_PMPStep1                                "If set to 1, allow LEVL.up = LEVL.lo in first PMP step"
*
*
    p_pmpCnst(RALL,COLS,A)                    "PMP parameter for linear own area cost effect"
    p_pmpQuadPact(RALL,COLS,COLS)             "PMP parameter for cross-crop-groups quadratic PMP effects"
    p_pmpQuadLandTypes(RALL,COLS,COLS)        "PMP parameter for land markets"
    p_pmpQuadTechn(RALL,COLS,A,A)             "PMP parameter for own area v_sumOfPmpTermsLevlstic cost effect"
--------------------------------------------------------------------------------
*                     model variables
--------------------------------------------------------------------------------
    VARIABLES
*
       v_obje                                 "Objective value"
       v_actLevl(RALL,COLS,*)                 "Level of production activities in 1000 ha or 1000 heads"
--------------------------------------------------------------------------------
       v_youngAnimUse(RALL,OM)                "Intrasectoral use of young animals in 1000 heads"
       v_feedQuantReg(RALL,* %addtimedim_ast%)              "Regional feed use in 1000 t per year and herd"
       v_feedInpCoeff(RALL,MAACT,A,* %addtimedim_ast%)      "Feeding per head and year in kg"
       v_pmpCostFeedPerAnim(RALL,MAACT,A)     "Per unit PMP feed cost"
       v_netPutQuant(RALL,*)                  "Selling and buying activities in 1000 t"

       v_lossQuant(RALL,ROWS %addtimedim_ast%)              "Losses of straw and organic fertiliser in 1000 t"
       v_nutAvailFactExcr(RALL,FOUT,A)        "Nutrient availability factor in manure"
       v_nutAvailFactCRes(RALL,FOUT,A)        "Nutrient availability factor for crop residues"
       v_cropNutNeedMultFact(RALL,FNUT,*)     "Multiplative Nutrient need factor for crops, per region and technology"
       v_cropNutNeedAddFact(RALL,FNUT)        "Constant nutrient need factor for crops, per region"

       v_animReq (RALL,*,A,* %addtimedim_ast%)              "Requirements per head and day"
       v_linObjePart(RALL)                    "Linear part of objective"
       v_sumOfPmpTermsLevls                   "Objective contribution of PMP terms activities"
       v_sumOfPmpTermsFeed                    "Objective contribution of PMP terms feeding"
       v_pmpCostLandMarket                    "Objective contribution of land market"
       v_landSupCost(RALL)                    "Cost for supplying land to agriculture"
       v_labCap(RALL)
*
       v_CO2EquEmis(RALL,*)                   "Global warming emissions"
       v_nutSurPlus(Rall,Fnut)                "Nutrient surpluses in 1000 tons"
       v_minShareMinFertCorr(RALL,NGRP,FNUT)  "Correction of minimum application rates of mineral fertilizer"
*
       v_fertDist(RALL,*,FNUT,*)              "Distribution of organic and mineral N to groups of crops"
       v_ManureNPK(RALL,*)                    "Total N,P,K at tail net of gaseous losses"
*
       v_watUse(RALL,*)                       "Regional water use in 1000 m3"
       v_watCos(RALL,*)                       "Regional water cost in 1000 euros"
*
       v_SIGMSugb(RALL,A)                     "Sales multiplied with VCOEF (??)"
       v_cdfSugb(RALL,A,Qut_A_AB)             "Cummulative probability for the production to be lower then A or A+B quotas"
       v_pdfSugb(RALL,A,Qut_A_AB)             "point probability for production being equal to A res. A+B quota"
       v_sugbRev(RALL,A)                      "Revenues from sugar beet A,B,C sales"
       v_salesSugb(RALL,A)                    "Sugar beet sales per technology"
       v_fixCosts(RALL)                       "Fix costs and premiums to generate compensated supply response"

       v_nonfSlack(RALL,A)                    "Slack which allows to turn non-food into a unequality"
       v_corfSetr(RALL)                       "Correction factor to render set-aside binding"
```

```
$           include 'supply\supply_model.gms';

    MODEL m_capMod/LINEAR_,OBJEQF_,
                   SUPBAL_,INPANI_,
*
*    --- feeding block
*
                   REQSE_,REQSN_,
                   MAXSHR_,MINSHR_,
                   FEDUSE_,
*
*    --- fertilization block
*
                   NUTNED_,NUTMIN_,
                   fertDistExcr_,
                   fertDistMine_,
                   fertDistCres_,
                   ManureNPK_,
*
*    --- cost function
*
                   GRPLEVL_,QUADRA_
--------------------------------------------------------------------------------
*    --- set-aside
*
                   SETA_
                   SETAN_
                   MXSETA_
                   NONF_
                   sumEntl_
                   overShotEntl_
                   greenOverShot_
--------------------------------------------------------------------------------
                   fixTechfShares_
--------------------------------------------------------------------------------
                   nGrpLevl_,
                   nMax_,
                   lsDensMax_,
--------------------------------------------------------------------------------
                   SalesSugb_,SIGMSugb_,cdfSugb_,pdfSugb_,SugbRev_,
                   netPutQuantSugb_
                   winterCover_
                   ecoSetAside_
                   cropDivGreening_
                   /;
*

* ---------- same model as m_capMod, with pmp s and land constraint included ----------------------
*
*

    MODEL m_capModQ/m_capMod,
*                --- add the cost function terms for feeding
                 QUADRF_
                 QUADRF1_
*                --- add the obligatory set-aside constraints, missing in calibration to obtain expert dual
*                --- add land balances, missing in calibration to obtain expert dual
                 LandMarket_
                 LandBal_
                 labCap_
                 e_uaar
                 e_asym
                 e_landSupCost
                 permGrasGreening_
--------------------------------------------------------------------------------------------------------
                  /;
*
```

2. Important modules in the supply model exemplified by different scenarios

```
$          include 'supply\supply_model.gms';
```

Supply balances for final outputs

Goods output and inputs with balances in the supply model

Sales (positive) and purchases (negative)

YCOW, YBUL, YHEI, YCAM, YCAF, YPIG, YLAM, YCHI

"Non tradeable feeding stuffs" MAIF, ROOF, OFAR, GRAS, STRA, COMF, SGMF

"tradeable feeding stuffs" FCER, FPRO, FENE, FMIL, FOTH

```
     SUPBAL_(RUNR_OMS_IN_SUPBAL(RUNR,OMS) )   ..
*
*                                   --- sales or purchases
                                        (exemption: fodder which is not tradeable)
          v_netPutQuant(RUNR,OMS)    $ ( (NOT FODDO(OMS)) or ((p_PMPStep1 eq 1) $ (NOT SAMEAS(OMS,"STRA")))
                                         or (p_exoDemand(RUNR,"UVAG",OMS) $ (p_PMPStep1 eq 1)))
*
*
*                                   --- young animals needed in regions
*                                       (defined in equation INPANIE_)
          + v_youngAnimUse(RUNR,OMS)   $ OMYANI(OMS)
*
*                                   --- non-tradable feed use (FGRA,FOFA ...)
*                                       (defined in equation FEDUSE_)
          + SUM((FEED_TO_O(FODDI,OMS)), v_feedQuantReg(RUNR,FODDI   ))
*
*                                   --- feed use of tradables bulks (FCER,FPRO ...)
*                                       (defined in equation FEDUSE_)
          +  v_feedQuantReg(RUNR,OMS ) $ I(OMS)

          + p_exoDemand(RUNR,"DEMD",OMS)
*
                         =E=
*
*                                   --- total output: activity level * output coefficient
*                                   ---                * technology factor for alternative
*
         SUM( (MPACT,A) $ (PACT_TO_O(MPACT,OMS) $ p_technFact(RUNR,MPACT,"LEVL",A)),
            v_actLevl(RUNR,MPACT,A) * (DATA(RUNR,MPACT,OMS,"Y") * (p_technFact(RUNR,MPACT,OMS,A)+1.) * 0.001)

           )
*                                   --- keep seed and v_lossQuant in
*                                   --- constant proportion
*
*                                              * (1. - p_onFarmShare(RUNR,OMS));
*
*-------------------------------------------------------------------------------------------------------
*
```

```
$           include 'supply\supply_model.gms';
```

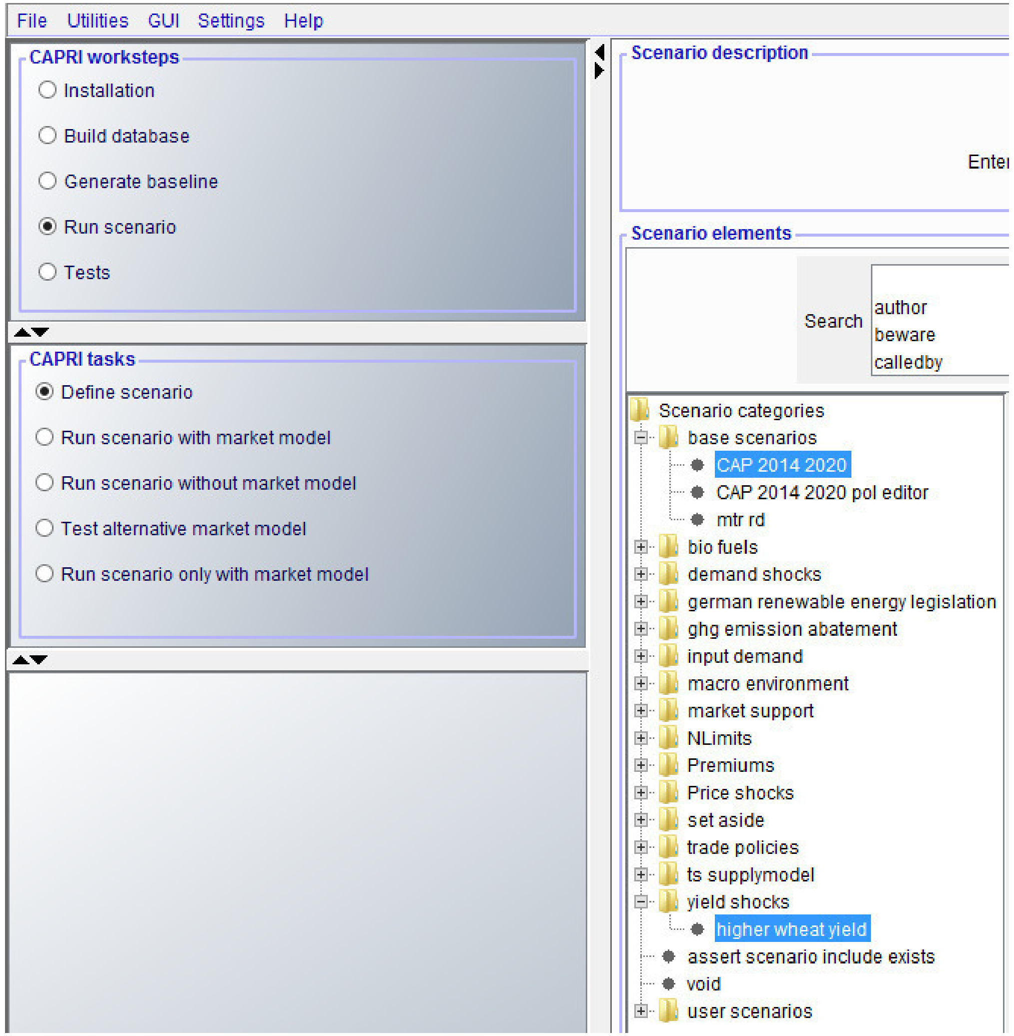
Young animal balance

```
* ---------- adding up use of young animals -------------------------------------------------
*
   INPANI_(RUNR,IYANI) $ SUM ( O_TO_YANI(OMYANI,IYANI) $ (v_youngAnimUse.lo(RUNR,OMYANI) ne v_youngAnimUse.up(RUNR,OMYANI)),1) ..
*
*                                          --- young animals needed by
*                                          --- region RUNR
*
     SUM ( O_TO_YANI(OMYANI,IYANI), v_youngAnimUse(RUNR,OMYANI) )
*
           =E=
*                                          --- total need added over activities
*                                          --- and alternatives
*
     0.001 * SUM( MAACT $ (p_technFact(RUNR,MAACT,"LEVL","T") $ PACT_TO_I(MAACT,IYANI)),
*
                   v_actLevl(RUNR,MAACT,"T") * DATA(RUNR,MAACT,IYANI,"Y") * (p_technFact(RUNR,MAACT,IYANI,"T")+1.)
                   );
*
```

```
$           include 'supply\supply_model.gms';
```

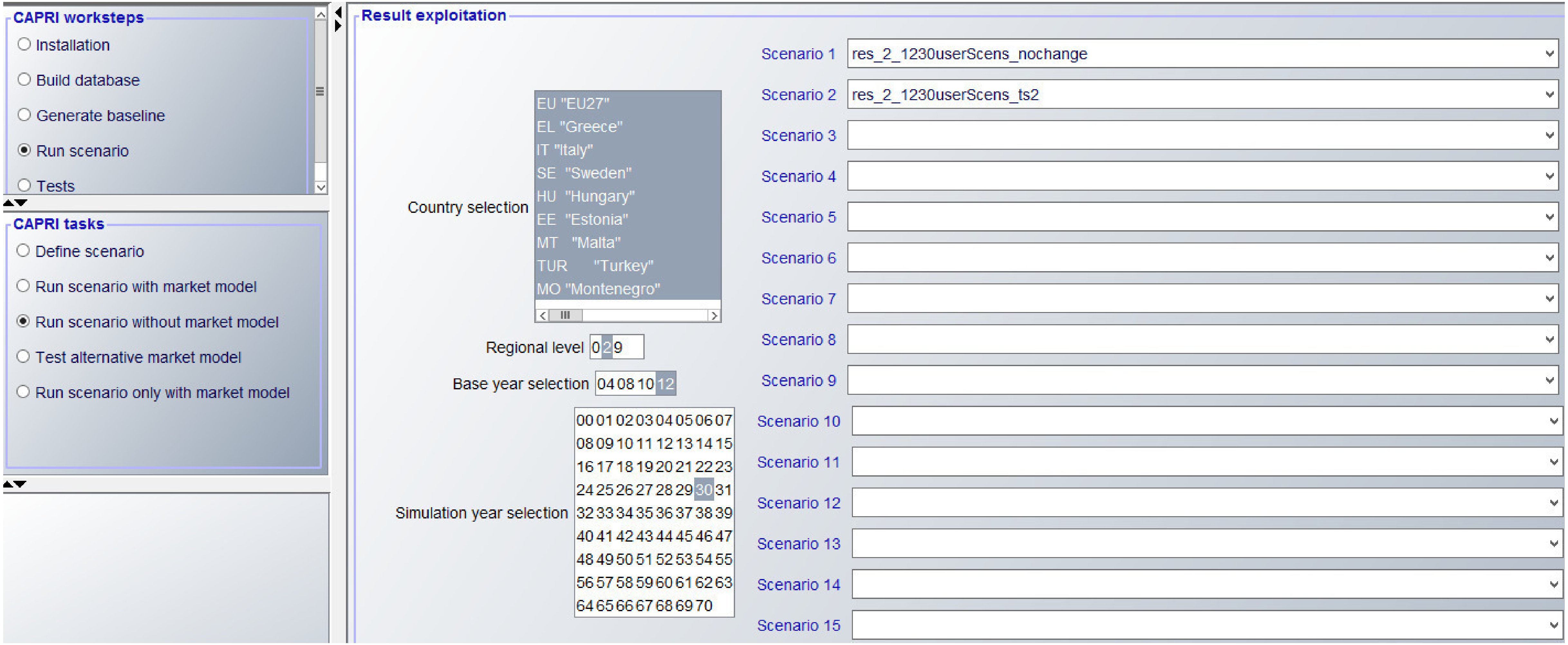
Area	Equations	Variables	Parameters or scalars	Exercise from the policy editor
Balance of products and young animals	---- EQU SUPBAL_ Supply balances for final outputs ---- EQU INPANI_ Input balances for young animals regional	v_netputQunt v_YoungAnimUse V_feedQuantReg V_actLevl	Yields DATA(RUNR,MPACT,OMS,"Y") young animal requirements DATA(RUNR,MAACT,IYANI,"Y") p_exoDemand	Yield, Young animal input p_exoDemand Exercise ts_scenario

Scenario: Wheat yield increase by 20%




```
********************************************************************************
*
* User supplied description :
*
* ts2: higher wheat yield
********************************************************************************
$setglobal SCENDES ts2: higher wheat yield
*
*  Category : base scenarios
*
$include ..\gams\scen\base_scenarios\CAP_2014_2020_pol_editor.gm
*
*  Category : yield shocks
*
* -------------------------------------------------------------------------------
*
*    Example file for exogenous yield changes
*    @Beware: input demand will be increased accordingly)
*
* -------------------------------------------------------------------------------
*
 DATA(RU,"SWHE","Yild","ChangeFactor") = 1.20;
```

Selection of baseline scenario (nochange) and yield increase scenario (ts2)



Results: Yield and income changes for wheat yield increase (+20%)

Supply details [0]

Region: Denmark — Year: 2030

	nochange		ts2	
	Income [Euro/ha or head]	Yield [kg, Const EU or 1/1000 head/ha]	Income [Euro/ha or head]	Yield [kg, Const EU or 1/1000 head/ha]
Cereals	257.58	1179.91	309.19 20.04%	1303.72 10.49%
Oilseeds	474.62	1593.05	473.88 -0.15%	1593.45 0.03%
Other arable crops	-76.58	4396.95	-70.58 7.83%	4400.93 0.09%
Vegetables and Permanent crops	41484.92	51814.03	41474.30 -0.03%	51807.56 -0.01%
Fodder activities	203.79	1022.21	206.67 1.41%	1020.90 -0.13%
Set aside and fallow land	332.83		332.22 -0.18%	
All cattle activities	2270.96	2730.27	2261.16 -0.43%	2730.94 0.02%
Beef meat activities	696.97	886.47	692.00 -0.71%	886.56 0.01%
All Dairy	2482.60	2978.20	2472.08 -0.42%	2978.86 0.02%
Other animals	407.62	1797.02	407.88 0.06%	1796.98 -0.00%

Results: Yield and income changes for wheat yield increase (+20%)

Supply details [0]

Region: Denmark

Year: 2030

	nochange		ts2	
	Income [Euro/ha or head]	Yield [kg, Const EU or 1/1000 head/ha]	Income [Euro/ha or head]	Yield [kg, Const EU or 1/1000 head/ha]
Cereals	257.58	1179.91	309.19 20.04%	1303.72 10.49%
Soft wheat	272.07	7725.17	388.20 42.69%	9228.73 19.46%
Rye and Meslin	302.34	6786.33	301.45 -0.29%	6787.89 0.02%
Barley	223.47	6234.14	222.66 -0.36%	6234.37 0.00%
Oats	229.86	5354.56	229.36 -0.22%	5351.06 -0.07%
Grain Maize	609.41	7648.96	608.62 -0.13%	7649.12 0.00%
Other cereals	475.64	6026.49	474.84 -0.17%	6026.65 0.00%

```
$          include 'supply\supply_model.gms';
```

Land Balance

- Two modes:
  - endowment fixed for grass and arable land (p_landIsFixed eq 1)
  - Via land supply function NONUAA in UAA -> in ARAB and GRAS (default setting)

```
changes
    02.11.09    T Jansson    New land supply function with nested decisions
                             introduced. There are now in total three tiers
                             in which the maximally available area (ASYM) is
                             transformed to different uses. The top level is
                             still governed by the old familiar PMP system.

                CROP1 .. CROPN  GRAE     GRAI    slack
                    \         /     \       /       /
                     \       /       \     /       /
                      \     /         \   /       /
                       ARAB            GRAS      /
                          \            /        /
                           \          /        /
                            \        /        /
                              UAAR           /
                                 \          /
                                  \        /
                                    ASYM
```

```
$          include 'supply\supply_model.gms';
```

Fixed land endowment

- P_landIsFixed

```
        LandBal_         "Land balances - e[illegible]r fixed endowments or market clearing"
---------------------------------------------------------------------------------------------
    LandBal_(RUNR,landTypesBal)  ..
*
*   --- sum of crops using that land type = total area of that type
*
        SUM( (landTypes_to_pact(landTypesBal,MPACT),A)  $ p_technFact(RUNR,MPACT,"LEVL",A),
                   v_actLevl(RUNR,MPACT,A)) =L=
            DATA(RUNR,landTypesBal,"LEVL","Y") $ (p_landIsFixed eq 1)
```

Grassland (GRAS), arable (ARAB), annual crops (ANNC)

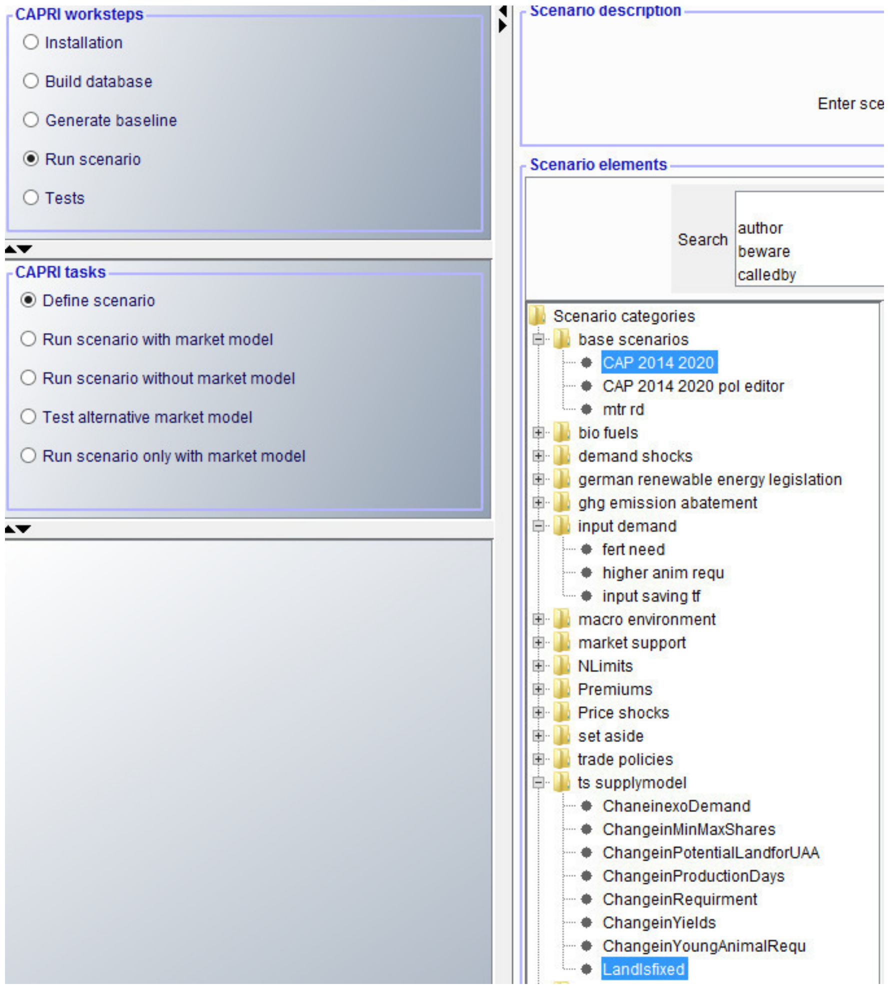
Check what activities enter which land type

Scenario with fixed land endowment

```
$       include 'supply\supply_model.gms';
```

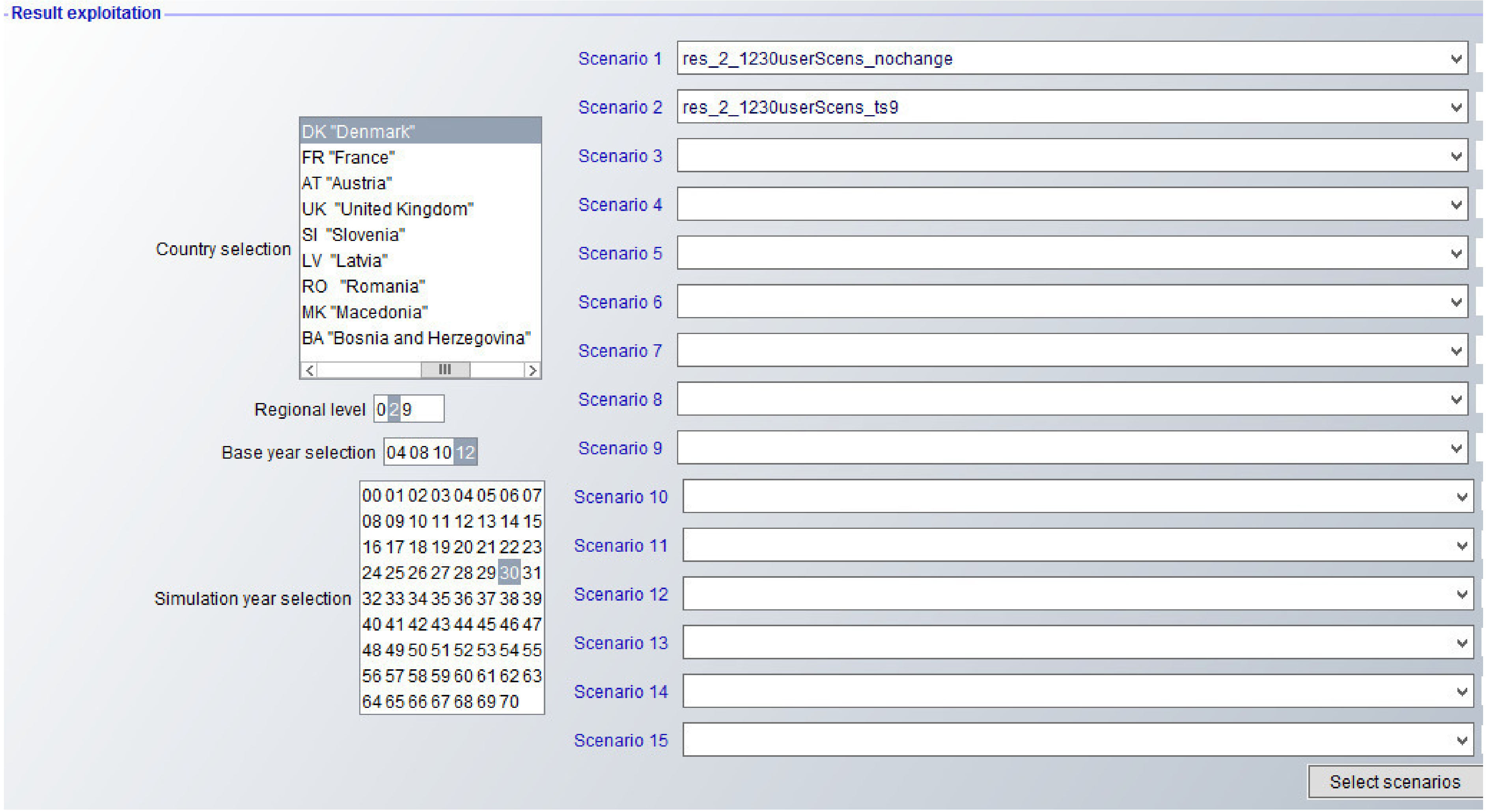
Area	Equations	Variables	Parameters or scalars	Exercise from the policy editor
Land Balance	LandBal_ Fixed endowment of market clearing	v_actLevl(Landtypes,"LEVL")	p_landisfixed DATA (RU, landtypes, "Levl","Y")	Exercise

Scenario: Reduction of available arable land by 10%




```
* User supplied description :
*
* ts9: land is fixed
********************************************************************************
$setglobal SCENDES ts9: land is fixed
*
*  Baseline scenario
*
$include ..\gams\scen\base_scenarios\CAP_2014_2020_pol_editor.gms
*
*  Category : ts supplymodel
*
********************************************************************************
$ontext

  CAPRI project

  GAMS file : LANDISFIXED.GMS

  @purpose  : fix land endowment for training session
  @author   : A. Gocht
  @date     : 11.06.18
  @refDoc   :
  @seeAlso  :
  @calledBy : capmod.gms

$offtext
********************************************************************************
scalar p_landIsFixedinScenario /1/;

DATA(RU,"ARAB","LEVL","ChangeFactor") = 0.90;
display "scenario setting:",DATA;
```

Selection of baseline scenario (nochange) and arable land reduction scenario (ts9)



Results: Reduction of available arable land by 10%

Supply details [0]

Region: Denmark

	nochange	ts9
	Hectares or herd size [1000 ha or hds]	Hectares or herd size [1000 ha or hds]
Utilized agricultural area	2640.71	2403.73 -8.97%
Pasture	270.90	270.90 0.00%
Arable land	2369.81	2132.83 -10.00%
All agricultural activities	6772.36	6551.00 -3.27%

Supply details [0]

Region: Denmark Year: 2030

	nochange		ts9	
	Income [Euro/ha or head]	Supply [1000 t, 1000 ha or Mio Const EU]	Income [Euro/ha or head]	Supply [1000 t, 1000 ha or Mio Const EU]
Cereals	257.58	1644.16	269.93 4.80%	1426.48 -13.24%
Oilseeds	474.62	246.22	483.04 1.78%	203.48 -17.36%
Other arable crops	-76.58	394.41	-103.69 -35.40%	410.10 3.98%
Vegetables and Permanent crops	41484.92	1190.00	41677.12 0.46%	1188.44 -0.13%
Fodder activities	203.79	872.09	117.83 -42.18%	911.06 4.47%

```
$          include 'supply\supply_model.gms';
```

Market clearing

- P_landIsFixed

Grassland (GRAS), arable (ARAB), annual crops (ANNC)

```
          LandBal_         "Land balances - either fixed endowments or market clearing"
------------------------------------------------------------------------------------------
     LandBal_(RUNR,landTypesBal)  ..
 *
 *   --- sum of crops using that land type = total area of that type
 *
         SUM( (landTypes_to_pact(landTypesBal,MPACT),A)  $ p_technFact(RUNR,MPACT,"LEVL",A),
                             v_actLevl(RUNR,MPACT,A)) =L=
              DATA(RUNR,landTypesBal,"LEVL","Y") $ (p_landIsFixed eq 1)
            + v_actLevl(RUNR,landTypesBal,"T")     $ (p_landIsFixed eq 0);
```

Check what activities enter which land type

Scenario with fixed land endowment

```
$            include 'supply\supply_model.gms';
```

Market clearing

- Land supply, see also in doc
- potentially be used by agriculture is allocated between agriculture and other land uses
- At the second level the regional representative farm decides how to allocate total land supplied to: arable and grass land {arab, gras}.
- Heterogeneity of land is reflected in a concave cost curve for increasing the allocation to the two land types.
- The representative farm optimizes the land allocation by maximizing the total land rent across land types minus the cost of allocating it to each land type.

```
$         include 'supply\supply_model.gms';
```

Constraints to potentially used land for agri.

Define UAA: Used agricultural areas

```
*tj -- Sum up total area used by agriculture (UAAR) by adding the land types

    e_uaar(RUNR) ..
        SUM(landTypes, v_actLevl(runr,landTypes,"t")) - v_actLevl(runr,"uaar","t") =E= 0;
```

Land supply function UAA less than potentially available land

```
*tj -- Total land potentially available for agriculture (the "asymptote") is
*      used plus unused land. We write this as an inequality, that should never
*      be binding due to the asymptotic curvature of the cost functi

    e_asym(RUNR) $ (p_landIsFixed eq 0)..
        v_actLevl(runr,"uaar","t") =L= %data%(runr,"asym","levl","y")*0.999;
```

```
$           include 'supply\supply_model.gms';
```

Costs for changes in the land allocation

- Nest is calibrated for the observed trend
- Two costs the farmer (land owner) has:
  - Supply UAA
  - Transform land GRAS and ARAB

Costs for supply land

A (parameter also used to calibrate the activities)

Enter the objective function

```
*tj -- Land supply for agriculture. The bottom level landUse nest.

    e_landSupCost(RUNR) $ (p_landIsFixed eq 0) ..

        v_landSupCost(RUNR) =E= p_pmpCnst(runr,"uaar","t")*v_actLevl(runr,"uaar","t")
                              + p_pmpQuadLandTypes(runr,"uaar","uaar")/p_pmpLandSupplyTail(runr,"uaar")
                              * [v_actLevl(runr,"uaar","t")/DATA(runr,"asym","levl","y")]**p_pmpLandSupplyTail(runr,"uaar")
                              * DATA(runr,"asym","levl","y");
```

b

$$\max\ p_{ua}x_{ua} - \left(a_{ua}x_{ua} + \frac{b_{ua}L}{c_{ua}}\left(\frac{x_{ua}}{L}\right)^{c_{ua}} \right) \qquad (1)$$

$$\text{s.t. } x_{ua} < L$$

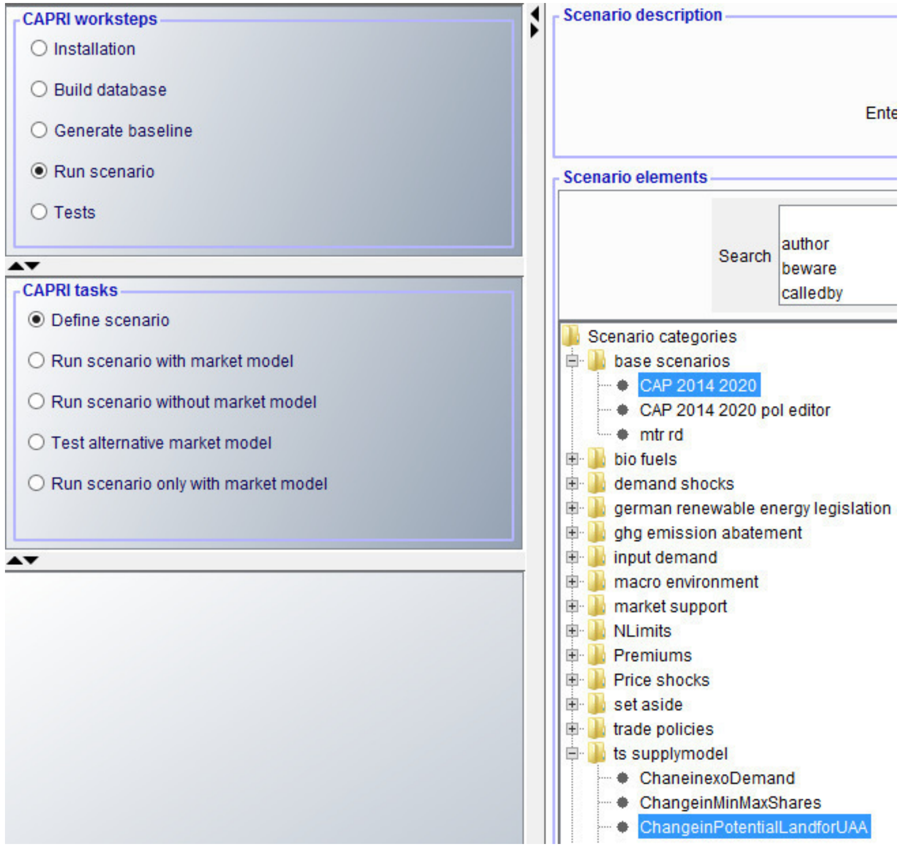
c

where the subscript *ua* denotes utilized agricultural area (uaar), *p* denotes the land rent, *x* the quantity of land used, *a*, *b* and *c* denote parameters to calibrate, and *L* is the maximally available agricultural area, MAA.

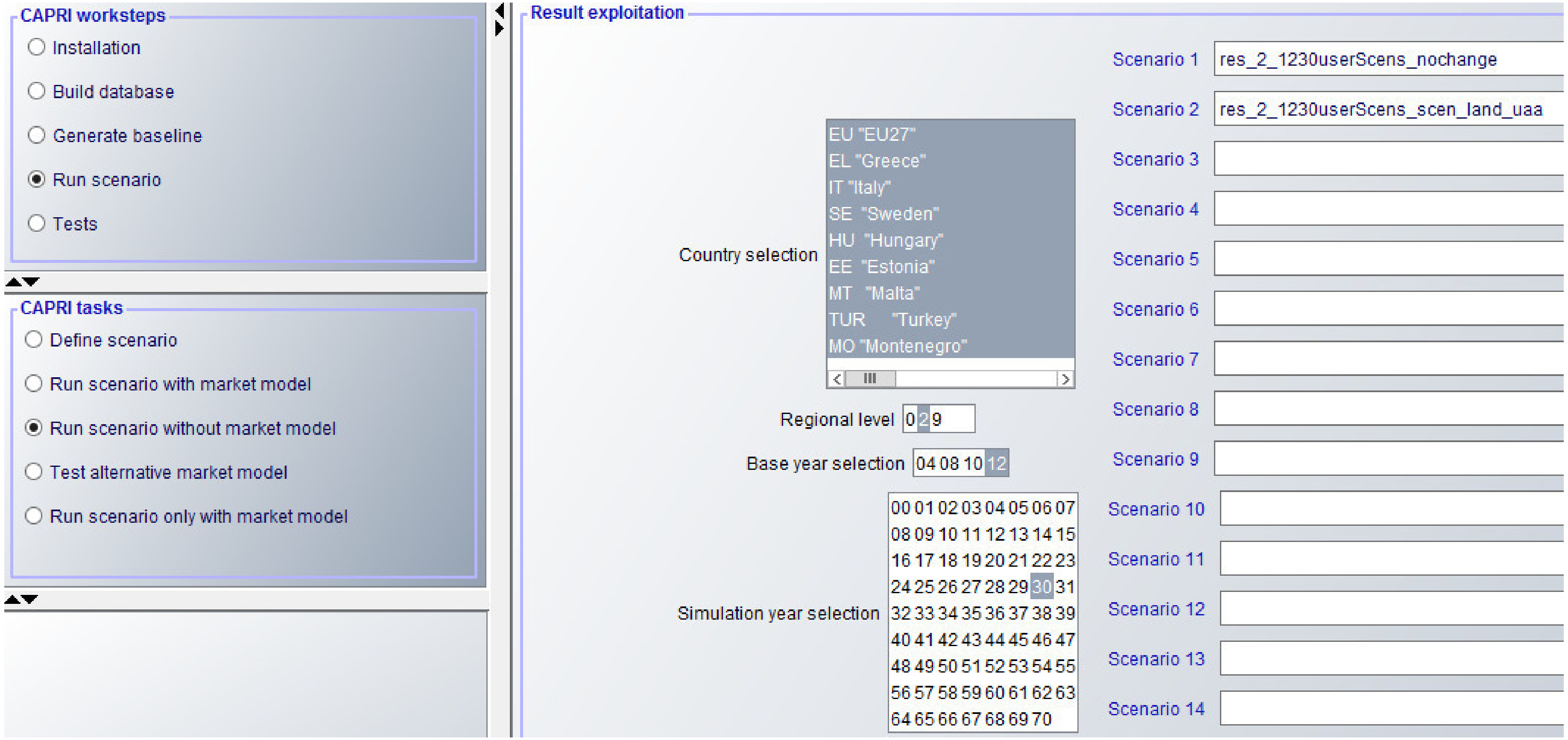
```
$            include 'supply\supply_model.gms';
```

Area	Equations	Variables	Parameters or scalars	Exercise from the policy editor
Land Balance	E_landSupCost See also OBJEQF_	v_actLevl(landtypesBal) v_landSupCost	p_landisfixed=1 DATA (RU, “Asym”, “Levl”,”Y”) p_pmpQuadLandTypes(UAAR) p_pmpLandSupplyTail	Exercise: ChangeofpotentialLandforUAA

Scenario: Change in available land for UAA (+10%)



Selection of baseline scenario (nochange) and increase of available land for UAA (scen_land_uaa)



Results: Change in available land for UAA (+10%)

Supply details [0]

Region: Denmark — Year: 2030

	nochange		scen_land_uaa	
	Income [Euro/ha or head]	Hectares or herd size [1000 ha or hds]	Income [Euro/ha or head]	Hectares or herd size [1000 ha or hds]
Cereals	257.58	1393.46	236.01 -8.37%	1522.13 9.23%
Oilseeds	474.62	154.56	455.11 -4.11%	170.37 10.23%
Other arable crops	-76.58	89.70	-69.15 9.70%	86.85 -3.17%
Vegetables and Permanent crops	41484.92	22.97	41353.67 -0.32%	23.03 0.28%
Fodder activities	203.79	853.15	197.60 -3.03%	837.26 -1.86%
Set aside and fallow land	332.83	126.87	311.06 -6.54%	188.69 48.72%
All cattle activities	2270.96	994.91	2240.03 -1.36%	993.26 -0.17%
Beef meat activities	696.97	117.92	687.38 -1.38%	117.80 -0.10%
All Dairy	2482.60	876.99	2448.94 -1.36%	875.46 -0.17%
Other animals	407.62	3136.75	401.22 -1.57%	3135.61 -0.04%

Supply details [0]

Region: Denmark

	nochange		scen_land_uaa	
	Income [Euro/ha or head]	Hectares or herd size [1000 ha or hds]	Income [Euro/ha or head]	Hectares or herd size [1000 ha or hds]
Cereals	257.58	1393.46	236.01 -8.37%	1522.13 9.23%
Soft wheat	272.07	565.76	250.93 -7.77%	610.32 7.88%
Rye and Meslin	302.34	116.00	282.44 -6.58%	128.10 10.43%
Barley	223.47	623.51	202.33 -9.46%	687.04 10.19%
Oats	229.86	54.64	210.32 -8.50%	61.90 13.28%
Grain Maize	609.41	15.52	588.35 -3.46%	15.52 0.00%
Other cereals	475.64	18.02	454.55 -4.43%	19.26 6.83%

```
$          include 'supply\supply_model.gms';
```

Cost for transforming land types

Enter the objective function

Quadratic Cost Function

Normalized by shares

```
*
    Land arket_(RUNR) $ (p_land  Fixed eq 0)..

        v_pmpCostLandMarket(RUNR) =E= SUM[ LandTypes $ DATA(runr,landTypes,"levl","y"),
                                    p_pmpCnst(runr,LandTypes,"T")*v_actLevl(runr,LandTypes,"T")
                                  + 0.5*p_pmpQuadLandTypes(runr,LandTypes,LandTypes)/v_actLevl(runr,"uaar","t")
                                       *SQR(v_actLevl(runr,LandTypes,"T"))];
```

```
$          include 'supply\supply_model.gms';
```

Cost for Transforming land types (Arab-Gras)

Area	Equations	Variables	Parameters or scalars	Exercise from the policy editor
Land Balance	LandMarket_ see also OBJEQF_	V_actLevel(landtypes) V_actLevel(“UAA”)	p_landisfixed=1 DATA (RU, “Asym”, “Levl”,”Y”) p_pmpCnst(LandTypes) p_pmpQuadLandTypes(landtypes)	Option: Change costs for transforming Land Types

1. Objective function in the supply model of CAPRI

```
$            include 'supply\supply_model.gms';
```

Objective function

- Linear function (LINEAR_)
  - sales/purchases valued by "unit value" price EAA
  - variable costs
  - Premiums
  - Minus Endogenous cut of premiums in case of overshot of entitlements (comes later)
- Total Objective function

  OBJEQF_ = summing linear and quadratic costs
- Target Variable

  v_obje in OBJEQF_ (solver print out)

```
$            include 'supply\supply_model.gms';

    LINEAR_(RUNR)..     v_linObjePart(RUNR)   =E=
*
*       --- sales/purchases valued by "unit value" price
*           from gross Economic Accounts for Agriculture

*

        SUM(   RUNR_OMOBJE(RUNR,OM_OBJE),
               v_netPutQuant(RUNR,OM_OBJE) * ( SUM(R_RAGG(RUNR,MSACT), (DATA(MSACT,"UVAG",OM_OBJE,"Y") $ (not p_useUvagScen) + (p_uvagScen(MSACT,OM_OBJE) $ p_useUvagScen)
                                                                                + (p_fnutuvag(RUNR,OM_OBJE) $ (p_useInputPScen $ sum(FNUT$sameas(OM_OBJE,FNUT),1)))     ))

      + SUM(   OMS $ (p_exoDemand(RUNR,"UVAG",OMS) $ (p_PMPStep1 eq 1)),
               v_netPutQuant(RUNR,OMS) * p_exoDemand(RUNR,"UVAG",OMS))

      + SUM(   OMS $ p_exoDemand(RUNR,"DEMD",OMS),
               p_exoDemand(RUNR,"DEMD",OMS) * (    p_exoDemand(RUNR,"UVAG",OMS) $ p_exoDemand(RUNR,"UVAG",OMS)
                               + DATA(RUNR,"UVAG",OMS,"Y") $ (not p_exoDemand(RUNR,"UVAG",OMS))))
*
*       --- in order to define the marginal value of the fodd supply balance,
*           introduce selling/buying activity in calibration step
*

      + SUM(   RUNR_OMS_IN_SUPBAL(RUNR,OMS) $ (FODDO(OMS) and  (NOT SAMEAS(OMS,"STRA"))),
               v_netPutQuant(RUNR,OMS) *  DATA(RUNR,"UVAG",OMS,"Y")) $ (p_PMPStep1 eq 1)
*
      + SUM(   A $ (p_technFact(RUNR,"SUGB","LEVL",A) $ p_technFact(RUNR,"SUGB","VCOF",A)), v_sug   ev(RUNR,A))
--------------------------------------------------------------------------------------------- 8 line(s) not displaye
*      --- activity levels multiplied with
*          variable costs (excluding feed and animals), negative values
*          premimums for specific alternatives
*          premimums defined in policy data set
*

     +  SUM(   (MPACT,A)   $ p_technFact(RUNR,MPACT,"LEVL",A),
              v_actLevl(RUNR,MPACT,A)   * (    p_linObjeCont(RUNR,MPACT,A) +   (p_inputPScen(RUNR,MPACT,A) $ p_useInputP
                                         +DATA(RUNR,MPACT,"PRME","Y")*(1+p_technFact(RUNR,MPACT,"PRME",A))
                                         ))
*
*      --- endogenous cut of premiums in case of overshot
*

     -  SUM(   (PSDPAY_cutEndog,DDTarget) $ p_entlLimit(RUNR,PSDPAY_cutEndog,   rget,"Limit"),
              v_overshotEntl(RUNR,PSDPAY_cutEndog,DDTarget)   * p_entlLimit(RUNR,P      Y_cutEndog,DDTarge

*
*   --- total objective (Income of agriculture + PQP-terms + PQP-feed-terms + mitig
*
    OBJEQF_..   v_obje =E= SUM(RUNR,
*
*           --- revenues - variable costs
*
            v_linObjePart(RUNR) - v_sumOfPmpTermsLevls(RUNR)*v_labCap.scale(runr) $ ( v_labCap.lo(RUNR) ne v_labCap.up(RUNR))
*
*           --- contribution of PMP terms for feeding
*
          + v_sumOfPmpTermsFeed(RUNR)*v_labCap.scale(runr) $ ((p_PMPStep1 ne 1) $ ( v_labCap.lo(RUNR) ne v_labCap.up(RUNR)))
*
*         --- cost of GHG mitigation options
*
*
*
*           --- land market
*

          - v_pmpCostLandMarket(RUNR)*v_labCap.scale(runr) $ ((p_landIsFixed eq 0) $ (p_PMPStep1 ne 1) $ (v_labCap.lo(RUNR) ne v_labCap.up(RUNR))
          - v_landSupCost(RUNR)          $ ((p_landIsFixed eq 0) $ (p_PMPStep1 ne 1))
          );
*
```

Scenario Solver

Only during calibration, calibrates to given prices for exogenous Demand

Only during calibration, calibration to given prices

Variable costs see define_obje.gms

Premium per activities

OM_OBJE(ROWS): Goods in objecti	
BARL	
OATS	
MAIZ	
OCER	
RAPE	
SUNF	
SOYA	
PARI	
OLIV	
PULS	
POTA	
SUGB	
TEXT	
TOBA	
TOMA	
OVEG	
APPL	
OFRU	
CITR	
TAGR	
TABO	
TWIN	
FCER	
FPRO	
FENE	
FMIL	
FOTH	
NITF	"Nitrogen in fertiliser"
PHOF	"Phospate in fertiliser [P2O5]
POTF	"Potassium in fertiliser [K2O]
YCOW	"Young cow"
YBUL	"Young bull"
YHEI	"Young heifer"
YCAM	"Young male calf"
YCAF	"Young female calf"
YPIG	"Young piglet"
YLAM	"Young lamb"
YCHI	"Young chicken"
COMI	
BEEF	
PORK	
SGMI	
SGMT	

```
$          include 'supply\supply_model.gms';
```

Objective function

Area	Equations	Variables	Parameters or scalars	Exercise from the policy editor
Objective Function	LINEAR_,	v_linObjePart v_overshotEntl v_actLevl v_netPutQuant	Premium: DATA(RUNR,MPACT,"PRME","Y") Premium loss when overshot: p_entlLimit(RUNR,PSDPAY_cutEndog, DDTarget,"Cut") Price: DATA(MSACT,"UVAG",OM_OBJE,"Y") Switch for scenario solver: p_useUvagScen:	
	OBJEQF_	v_linObjePart v_sumOfPmpTermsLevls v_labCap v_sumOfPmpTermsFeed v_pmpCostLandMarket v_landSupCost		

```
$          include 'supply\supply_model.gms';
```

Area	Equations	Variables	Parameters or scalars	Exercise from the policy editor
Feeding block	FEDUSE_ Balance for feedingstuff regional add up feed use over animals and alternatives (endogenous input coefficient in kg times herd size in 1000 animals), scaling to define feed use in 1000 t	v_feedInpCoeff v_actLevl(MAACT) v_lossQuant v_feedQuantReg	p_maxFeedShare	See exercise
	MI(A)NSHR_ Mi(a)nimum feed shares	v_feedInpCoeff	p_animReq(RUNR,MAACT,A,"DRMA") p_maxFeedShare p_animProdDays DATA(MSACT,"DRMA",FEED,"Y")	See exercise
	REQSE(N)_ Requirements of animals written as (in)-equality	v_feedInpCoeff	p_animReq p_animProdDays	See exercise

2. Definition and calculation of different scenarios

Link to Supply balance

Feed Mix per animal

Based on input all non-tradable and tradable feed

```
--------------------------------------------------------------------------------
FEDUSE_(RUNR,FEED ) $ ( v_feedQuantR g.LO(RUNR,FEED ) ne v_fee NR,FEED ) ) ..

   ---  add up feed use over animals and alternatives (endogen ient in kg times
        herd size in 1000 animals), scaling to define feed use

   SUM( (MAACT,A) $ p_maxFeedShare(RUNR,MAACT,A,FEED ),

       v_feedInpCoeff(RUNR,MAACT,A,FEED )*v_actLevl(RUNR ACT,A ) ) / 1000.

    --- permit some variation of straw v_lossQuant

                        + v_lossQuant(RUNR,"STR  ) $ SAMEAS(FEED,"FSTR")
                        + v_lossQuant(RUNR,"COMF" ) $ SAMEAS(FEED,"FCOM")

                    =E= v_feedQuantReg(RUNR,FEED );
```

Requirements of animal

As requirement are per day multiply with no of production cycle of animals

In supply model p_trimFeed = false

```
      REQSE_          "Requirements of animals written as equality"
--------------------------------------------------------------------------------------------------------
    REQSE_(RUNR,MAACT,A,REQMSE ) $ (  p_animReq(RUNR,MAACT,A,REQMSE ) $ p_technFact(RUNR,MAACT,"LEVL",A) ) ..
*
*                                     --- animal requirements per activity level and day
         (v_animReq(RUNR,MAACT,A,REQMSE )  $ p_trimFeed + p_animReq(RUNR,MAACT,A,REQMSE ) $ (Not p_trimFeed))

             * p_animProdDays(RUNR,MAACT,A )
*                                     --- are covered by feeding of feedingstuff
           -  SUM(FEED $ p_maxFeedShare(RUNR,MAACT,A,FEED ),
                   v_feedInpCoeff(RUNR,MAACT,A,FEED )
                        * SUM(R_RAGG(RUNR,MSACT),DATA(MSACT,REQMSE,FEED,"Y")))
*
             =E= 0.0;

     REQSN_          "Requirements of animals written as in-equality"
--------------------------------------------------------------------------------------------------------
  REQSN_(RUNR,MAACT,A,REQMSN ) $ (  p_animReq(RUNR,MAACT,A,REQMSN  ) $ p_technFact(RUNR,MAACT,"LEVL",A) ) ..

                                      --- animal requirements per activity level and day
      [(v_animReq(RUNR,MAACT,A,REQMSN ) $ p_trimFeed + p_animReq(RUNR,MAACT,A,REQMSN ) $ (Not p_trimFeed))


          * p_animProdDays(RUNR,MAACT,A )
                                      --- are covered by feeding of feedingstuff
        -  SUM(FEED $ p_maxFeedShare(RUNR,MAACT,A,FEED ),
                v_feedInpCoeff(RUNR,MAACT,A,FEED )
                     * SUM(R_RAGG(RUNR,MSACT),DATA(MSACT,REQMSN,FEED,"Y"))) ]


            =L= 0.0;
```

ENNE and CPRO content of feed at MS

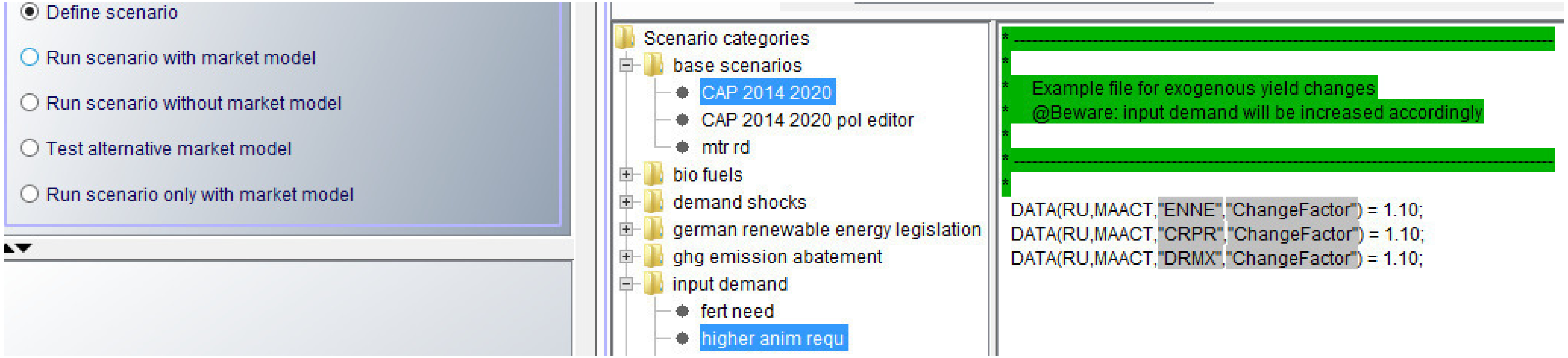
```
--- for maximum shares

MAXSHR_(RUNR,MAACT,A,FEED ) $ ( (p_maxFeedShare(RUNR,MAACT    D ) NE       re(RUNR,MAA

                          --- current tot   ry matter intake * maximum share
                          --- = maxi    dry matter intake of feedingstuff FEED
   p_animReq(RUNR,MAACT,A,"DRMA" )

                            * p_maxFeedShare(RUNR,MAACT,A,FEED )
                          --- corrector factor for maximum shares
                            * ((1+v_animReq(RUNR,MAACT,A,FEED )) $ p_trimFeed + 1. $

                            * p_animProdDays(RUNR,MAACT,A )

                          --- dry matter intake of feedingstuff FEE

             - v_feedInpCoeff(RUNR,MAACT,A,FEED )
               * SUM(R_RAGG(RUNR,MSACT),DATA(MSACT,"DRMA",FEED,"Y"))

                 =G= 0.0;


   --- for minimum shares

MINSHR_(RUNR,MAACT,A,FEED ) $ ( p_minFeedShare(RUNR,MAACT,A,FEED ) $ p_maxFeedShare(RUNR,MAACT,A,FEED )) ..

                          --- current total dry matter intake * minimum share
                          --- = minimum dry matter intake of feedingstuff FEED

   p_animReq(RUNR,MAACT,A,"DRMA" )

                            * p_minFeedShare(RUNR,MAACT,A,FEED )
                          --- corrector factor for minimum shares
                            * ((1-v_animReq(RUNR,MAACT,A,FEED ))  $ p_trimFeed + 1. $ (Not p_trimFeed))

                            * p_animProdDays(RUNR,MAACT,A )

                          --- dry matter intake of feedingstuff FEED

             - v_feedInpCoeff(RUNR,MAACT,A,FEED ) * SUM(R_RAGG(RUNR,MSACT),DATA(MSACT,"DRMA",FEED,"Y"))

                 =L= 0.0;
```

Convert into dry matter

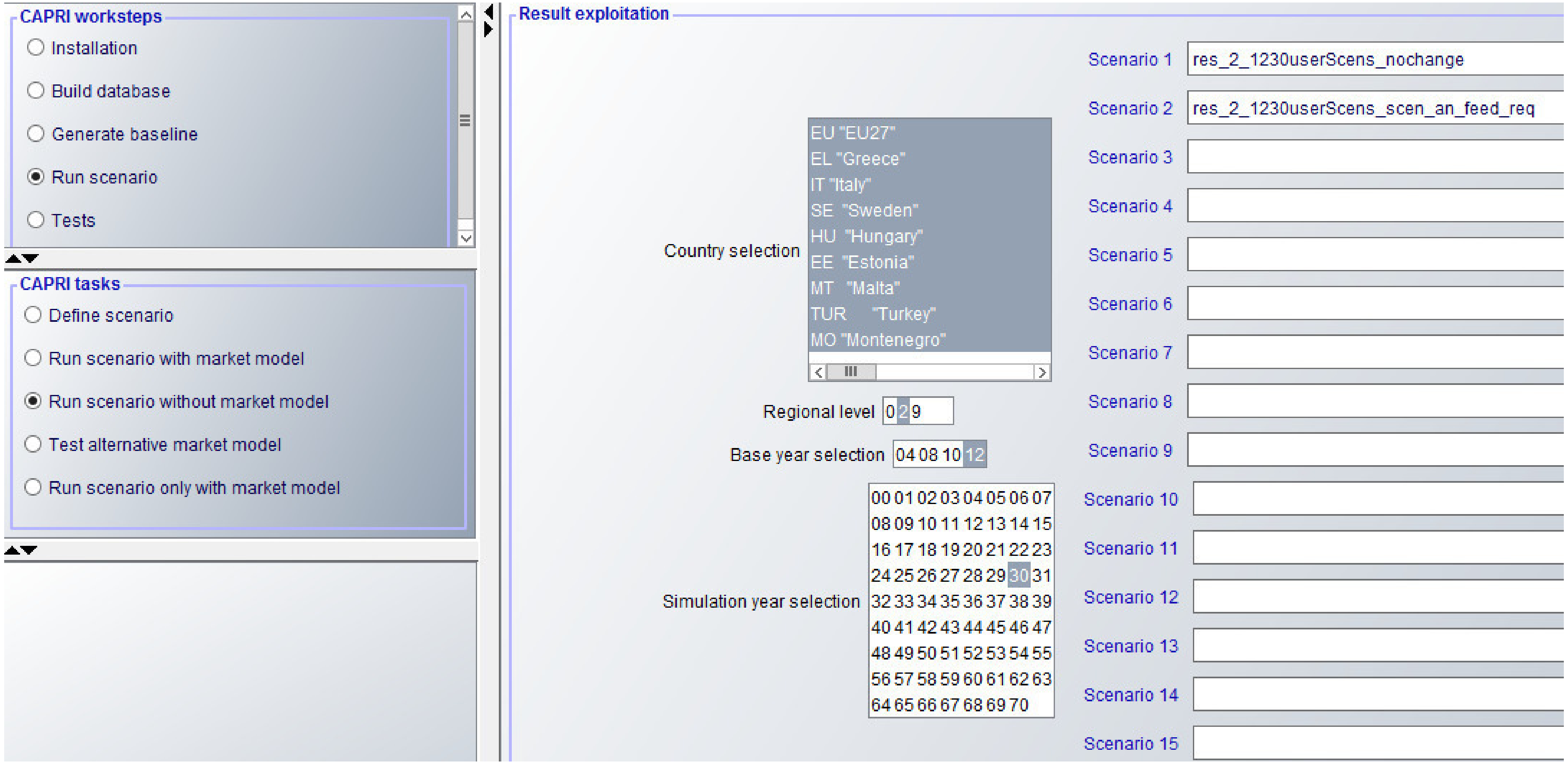
Multiply share

Feed mix is in fresh matter hence multiply with dry-matter content of the feed

Scenario: Higher animal feed requirements +10%
(for net energy lactation ("ENNE"), crude protein ("CRPR"), dry matter max ("DRMX")



Selection of baseline scenario (nochange) and increase of animal feed requirements (scen_an_feed_req)



Results: Higher animal feed requirements (+10%)

Feed Distribution - multiplied with herd size [0]

Region: Denmark | Year: 2030

	nochange				scen_an_feed_req			
	Feed cereals [kg/head]	Feed rich protein [kg/head]	Feed rich energy [kg/head]	Fodder maize [kg/head]	Feed cereals [kg/head]	Feed rich protein [kg/head]	Feed rich energy [kg/head]	Fodder maize [kg/head]
All cattle activities	344.84	708.10	4.22	6775.05	575.82 66.98%	816.61 15.32%	4.22 0.14%	7263.82 7.21%
All Dairy	339.67	703.27	3.85	6537.03	555.22 63.46%	807.44 14.81%	3.88 0.85%	7005.62 7.17%
Other animals	6339.53	1842.28	8.87	5706.02	6773.98 6.85%	1960.74 6.43%	8.69 -2.09%	5989.44 4.97%

Results: Higher animal feed requirements (+10%)

Supply details [0]

Region: Denmark

	nochange		scen_an_feed_req	
	Income [Euro/ha or head]	Hectares or herd size [1000 ha or hds]	Income [Euro/ha or head]	Hectares or herd size [1000 ha or hds]
Beef meat activities	696.97	117.92	651.19 -6.57%	106.83 -9.41%
All cattle activities	2270.96	994.91	2269.05 -0.08%	959.98 -3.51%
Other animals	407.62	3136.75	385.42 -5.45%	3040.95 -3.05%
All Dairy	2482.60	876.99	2471.64 -0.44%	853.15 -2.72%
Other arable crops	-76.58	89.70	-45.86 40.11%	87.84 -2.07%
Fodder activities	203.79	853.15	264.09 29.59%	849.32 -0.45%
Vegetables and Permanent crops	41484.92	22.97	41498.07 0.03%	22.98 0.04%
Oilseeds	474.62	154.56	494.29 4.15%	154.80 0.16%
Cereals	257.58	1393.46	300.61 16.71%	1417.43 1.72%

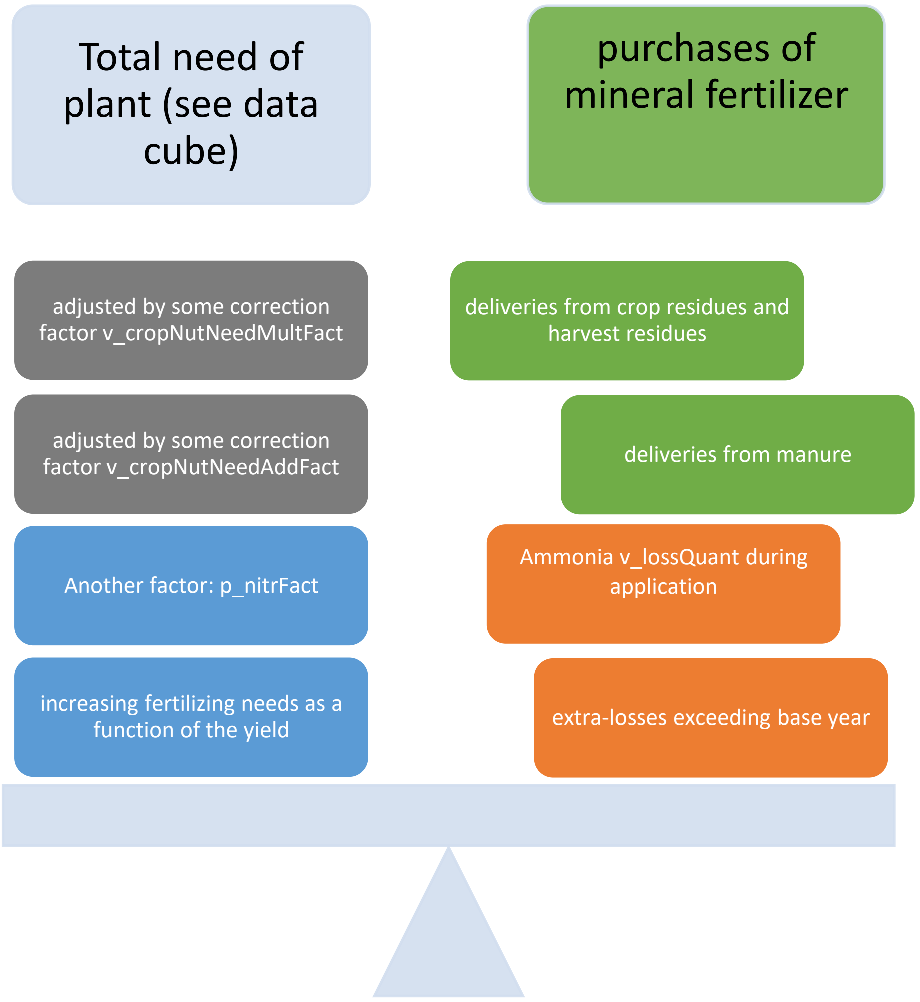
Fertilizer balance

- Currently under revision for Star2
- Balance is done not at activity level but on crop group level (NGRP)
- FNUT->NITF, "Nitrogen in fertiliser“, PHOF "Phospate in fertiliser [P2O5]“ and POTF "Potassium in fertiliser [K2O]“
- Mineral fertilizer is delivered from Supply balance

Balance

- Total need of plant (adjusted by some correction factor (v_cropNutNeedMultFact, v_cropNutNeedAddFact, p_nitrFact))

```
---------------------------------------------------------- 351 line(s) not displayed ------------------------------
    NUTNED_(RUNR,NGRP,FNUT) $ SUM( (CACT_TO_NGRP(MCACT,NGRP),A) $ p_technFact(RUNR,MCACT,"LEVL",A), 1) ..

      SUM( (CACT_TO_NGRP(MCACT,NGRP),A) $ p_technFact(RUNR,MCACT,"LEVL",A), v_actLevl(RUNR,MCACT,A)
*
*       --- total nutrient need of crops (retention - biolog. fixation for certain crops) * nutrient factor
*
         * (DATA(RUNR,MCACT,FNUT,"Y")
            * (1-p_nitrFact(RUNR,MCACT,"BioFix") $ SAMEAS(FNUT,"NITF")
            )
             * ^(RUNR,FNUT,A)
*
*       --- plus constant term of nutrient factor
*
             + v_cropNutNeedAddFact(RUNR,FNUT))
*
*       --- soil property effect
*
            * (p_nitrFact(RUNR,"ALL","DEFR") $ SAMEAS(FNUT,"NITF") + 1 $ (NOT SAMEAS(FNUT,"NITF")))
*
*       --- increasing fertilizing needs as a function of the yield
*
                                * SQRT(
                                    [1.+p_technFact(RUNR,MCACT,"Yield",A)
                                      + 0.2 $ SAMEAS(MCACT,"GRAI") - 0.2 $ SAMEAS(MCACT,"GRAE")
                                    ] $ DATA(RUNR,MCACT,FNUT,"Y")
                                   )
*
*       --- factor describing the changes in removal of nutrients
*           resulting from yield difference stored in TECHF
*
                               *(p_technFact(RUNR,MCACT,FNUT,A)+1.)
*
*
        )  * 0.001 =E=
*
*    --- purchases of anorganic fertiliser minus Ammonia v_lossQuant during application
*
       + v_fertDist(RUNR,NGRP,FNUT,"Mine") * (1.-p_emiLoss(RUNR,"NETF","N","GasRunTot") $ SAMEAS(FNUT,"NITF") )
*
*
*    --- extra-losses exceeding base year
*
       - v_fertDist(RUNR,NGRP,FNUT,"Loss")
*
*    --- deliveries from manure
*        corrected for further v_lossQuant (NOx, leaching)
*        (attention: a part of these v_lossQuant also hides in v_cropNutNeedMultFact)
*
       + v_fertDist(RUNR,NGRP,FNUT,"Excr") * SUM( FOUT_T_N(FOUT,FNUT),v_nutAvailFactExcr(RUNR,FOUT,"T"))
*
*    --- deliveries from crop residues (distribution across arable crops)
*
       + (v_fertDist(RUNR,NGRP,FNUT,"Cres")
             * SUM( FOUT_T_N(FOUT,FNUT),v_nutAvailFactCRes(RUNR,FOUT,"T"))
               * (1.-p_emiLoss(RUNR,"NETF","N","GasRunTot") $ SAMEAS(FNUT,"NITF") )) $ (NOT PERM_NGRP(NGRP))
*
*    --- delivery from harvest residues and atmospheric deposition (only for grass land, where it remains)
*
       + SUM( (CACT_TO_NGRP(MCACT,NGRP),FOUT_T_N(FOUT,FNUT),A)
              $ (p_technFact(RUNR,MCACT,"LEVL",A) and PERM_NGRP(NGRP)),
                 v_actLevl(RUNR,MCACT,A)
                  * DATA(RUNR,MCACT,FOUT,"Y")
                  * (p_technFact(RUNR,MCACT,FOUT,A)+1.)
                  * (1.-p_emiLoss(RUNR,"NETF","N","GasRunTot") $ SAMEAS(FNUT,"NITF") )
                  * v_nutAvailFactCRes(RUNR,FOUT,"T")) * 0.001;
*
```



```
$          include 'supply\supply_model.gms';
```

Area	Equations	Variables	Parameters or scalars	Exercise
Distribution of mineral and organic fertilizer to crops	fertDistMine_ ineral fertilizer distributed to group of crops must exhaust total mineral N sales	v_fertDist(RUNR,NGRP,FNUT,"Mine") v_netPutQuant(RUNR,FNUT)	No	
	NUTNED_ nutrient need balance for group of crops		Nutrient factors: p_nitrFact v_cropNutNeedAddFact v_cropNutNeedMultFact Nutrient need of crop: DATA(RUNR,MCACT,FNUT,"Y")	Exercise
	NUTMIN_ Minimum use of mineral fertilizer	v_actLevl(MCACT) v_fertDist(RUNR,NGRP,FNUT,"Mine")	p_minShareMinFert(MSACT,MCACT,A,FNUT) v_minShareMinFertCorr DATA(RUNR,MCACT,FNUT,"Y") p_nitrFact	Exercise

```
$          include 'supply\supply_model.gms';
```

Area	Equations	Variables	Parameters or scalars	Exercise from the policy editor
Feeding block	FEDUSE_ Balance for feedingstuff regional add up feed use over animals and alternatives (endogenous input coefficient in kg times herd size in 1000 animals), scaling to define feed use in 1000 t	v_feedInpCoeff v_actLevl(MAACT) v_lossQuant v_feedQuantReg	p_maxFeedShare	See exercise
	MI(A)NSHR_ Mi(a)nimum feed shares	v_feedInpCoeff	p_animReq(RUNR,MAACT,A,"DRMA") p_maxFeedShare p_animProdDays DATA(MSACT,"DRMA", FEED,"Y")	See exercise
	REQSE(N)_ Requirements of animals written as (in)-equality	v_feedInpCoeff	p_animReq p_animProdDays	See exercise

2. Definition and calculation of different scenarios

Link to Supply balance

Feed Mix per animal

```
------------------------------------------------------------------------------------------------
FEDUSE_(RUNR,FEED ) $ ( v_feedQuantR  g.LO(RUNR,FEED ) ne v_fee                NR,FEED ) ) ..

   ---   add up feed use over animals and alternatives (endogeno               ient in kg times
         herd size in 1000 animals), scaling to define feed use

   SUM( (MAACT,A) $ p_maxFeedShare(RUNR,MAACT,A,FEED ),

       v_feedInpCoeff(RUNR,MAACT,A,FEED )*v_actLevl(RUNR   ACT,A ) ) / 1000.

    --- permit some variation of straw v_lossQuant

                  + v_lossQuant(RUNR,"STR   ) $ SAMEAS(FEED,"FSTR")
                  + v_lossQuant(RUNR, COMF" ) $ SAMEAS(FEED,"FCOM")

              =E= v_feedQuantReg(RUNR,FEED );
```

Based on input all non-tradable and tradable feed

Requirements of animal

As requirement are per day multiply with no of production cycle of animals

In supply model p_trimFeed = false

```
REQSE_          "Requirements of animals written as equality"
----------------------------------------------------------------------------------------------------
    REQSE_(RUNR,MAACT,A,REQMSE ) $ (  p_animReq(RUNR,MAACT,A,REQMSE ) $ p_technFact(RUNR,MAACT,"LEVL",A) ) ..
*
*                                  --- animal requirements per activity level and day
       (v_animReq(RUNR,MAACT,A,REQMSE )  $ p_trimFeed + p_animReq(RUNR,MAACT,A,REQMSE ) $ (Not p_trimFeed))

              * p_animProdDays(RUNR,MAACT,A )
*                                  --- are covered by feeding of feedingstuff
            -  SUM(FEED $ p_maxFeedShare(RUNR,MAACT,A,FEED ),
                    v_feedInpCoeff(RUNR,MAACT,A,FEED )
                         * SUM(R_RAGG(RUNR,MSACT),DATA(MSACT,REQMSE,FEED,"Y")))
*
               =E= 0.0;
```

ENNE and CPRO content of feed at MS

```
REQSN_          "Requirements of animals written as in-equality"
----------------------------------------------------------------------------------------------------
  REQSN_(RUNR,MAACT,A,REQMSN ) $ (  p_animReq(RUNR,MAACT,A,REQMSN  ) $ p_technFact(RUNR,MAACT,"LEVL",A) ) ..

                                   --- animal requirements per activity level and day
     [(v_animReq(RUNR,MAACT,A,REQMSN ) $ p_trimFeed + p_animReq(RUNR,MAACT,A,REQMSN ) $ (Not p_trimFeed))


            * p_animProdDays(RUNR,MAACT,A )
                                   --- are covered by feeding of feedingstuff
          -  SUM(FEED $ p_maxFeedShare(RUNR,MAACT,A,FEED ),
                 v_feedInpCoeff(RUNR,MAACT,A,FEED )
                     * SUM(R_RAGG(RUNR,MSACT),DATA(MSACT,REQMSN,FEED,"Y"))) ]


              =L= 0.0;
```

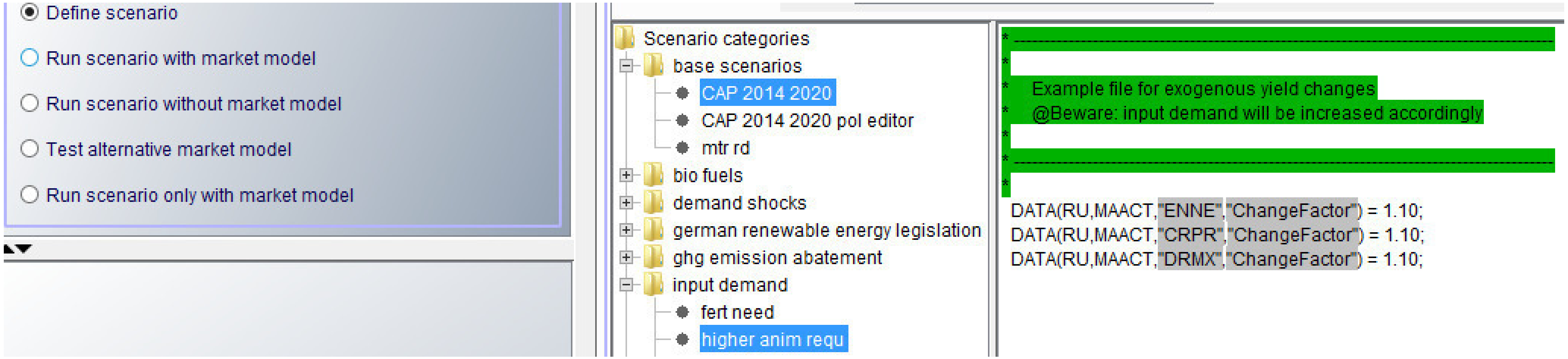
Convert into dry matter

Multiply share

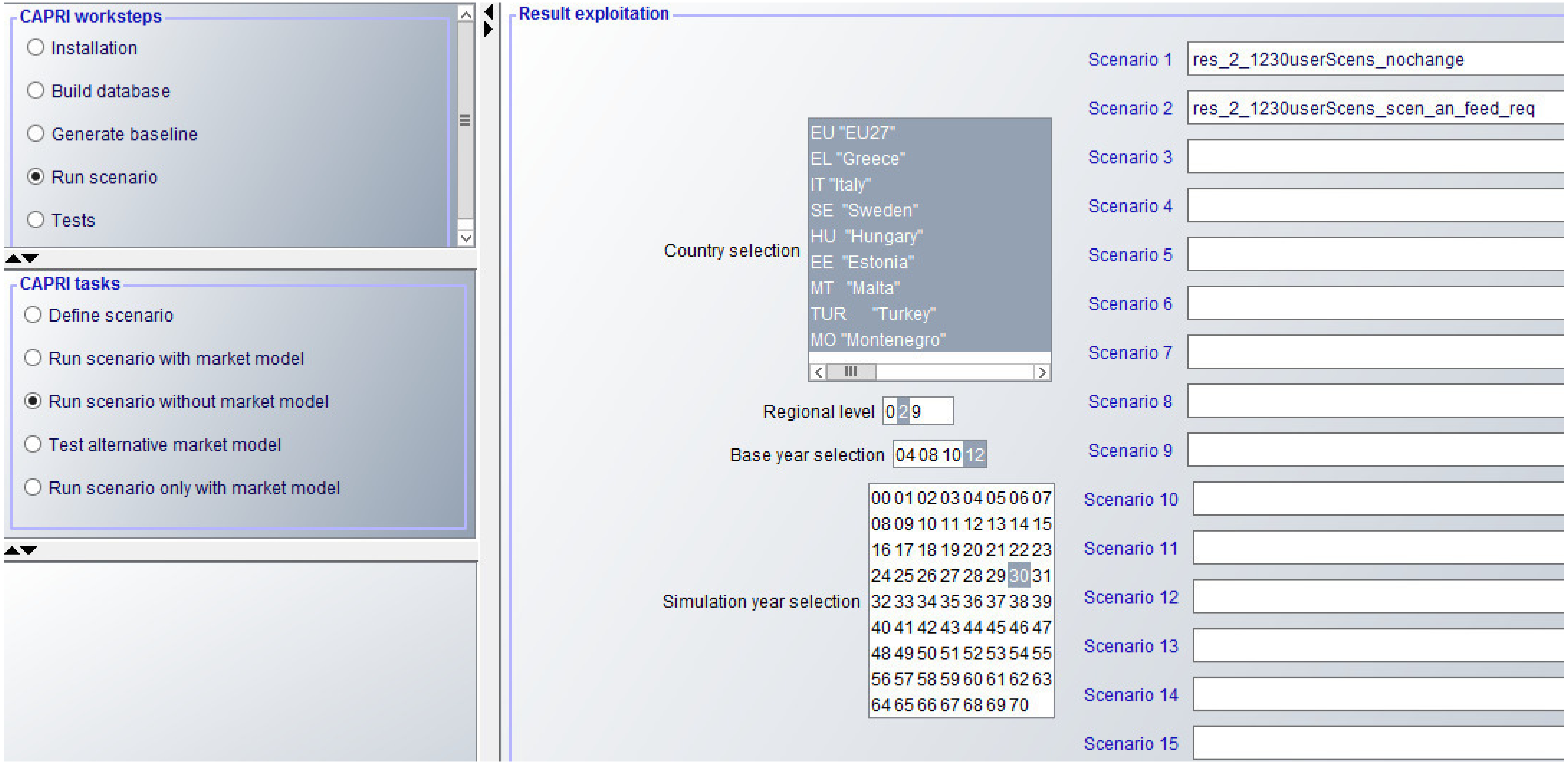
Feed mix is in fresh matter hence multiply with dry-matter content of the feed

```
    --- for maximum shares

 MAXSHR_(RUNR,MAACT,A,FEED ) $ ( (p_maxFeedShare(RUNR,MAACT        ) NE        re(RUNR,MAA

                                       --- current tot     ry matter intake * maximum share
                                       --- = maxi      dry matter intake of feedingstuff FEED
    p_animReq(RUNR,MAACT,A,"DRMA" )

                                         * p_maxFeedShare(RUNR,MAACT,A,FEED )
                                       --- corrector factor for maximum shares
                                         * ((1+v_animReq(RUNR,MAACT,A,FEED )) $ p_trimFeed + 1. $

                                         * p_animProdDays(RUNR,MAACT,A )

                                       --- dry matter intake of feedingstuff FEE

                            - v_feedInpCoeff(RUNR,MAACT,A,FEED )
                               * SUM(R_RAGG(RUNR,MSACT),DATA(MSACT,"DRMA",FEED,"Y"))

                                =G= 0.0;

    --- for minimum shares

 MINSHR_(RUNR,MAACT,A,FEED ) $ ( p_minFeedShare(RUNR,MAACT,A,FEED ) $ p_maxFeedShare(RUNR,MAACT,A,FEED )) ..

                                       --- current total dry matter intake * minimum share
                                       --- = minimum dry matter intake of feedingstuff FEED

    p_animReq(RUNR,MAACT,A,"DRMA" )

                                         * p_minFeedShare(RUNR,MAACT,A,FEED )
                                       --- corrector factor for minimum shares
                                         * ((1-v_animReq(RUNR,MAACT,A,FEED ))  $ p_trimFeed + 1. $ (Not p_trimFeed))

                                         * p_animProdDays(RUNR,MAACT,A )

                                       --- dry matter intake of feedingstuff FEED

                            - v_feedInpCoeff(RUNR,MAACT,A,FEED ) * SUM(R_RAGG(RUNR,MSACT),DATA(MSACT,"DRMA",FEED,"Y"))

                                =L= 0.0;
```

Scenario: Higher animal feed requirements +10%
(for net energy lactation ("ENNE"), crude protein ("CRPR"), dry matter max ("DRMX")



Selection of baseline scenario (nochange) and increase of animal feed requirements (scen_an_feed_req)



Results: Higher animal feed requirements (+10%)

Feed Distribution - multiplied with herd size [0]

Region: Denmark | Year: 2030

	nochange				scen_an_feed_req			
	Feed cereals [kg/head]	Feed rich protein [kg/head]	Feed rich energy [kg/head]	Fodder maize [kg/head]	Feed cereals [kg/head]	Feed rich protein [kg/head]	Feed rich energy [kg/head]	Fodder maize [kg/head]
All cattle activities	344.84	708.10	4.22	6775.05	575.82 66.98%	816.61 15.32%	4.22 0.14%	7263.82 7.21%
All Dairy	339.67	703.27	3.85	6537.03	555.22 63.46%	807.44 14.81%	3.88 0.85%	7005.62 7.17%
Other animals	6339.53	1842.28	8.87	5706.02	6773.98 6.85%	1960.74 6.43%	8.69 -2.09%	5989.44 4.97%

Results: Higher animal feed requirements (+10%)

Supply details [0]

Region: Denmark

	nochange		scen_an_feed_req	
	Income [Euro/ha or head]	Hectares or herd size [1000 ha or hds]	Income [Euro/ha or head]	Hectares or herd size [1000 ha or hds]
Beef meat activities	696.97	117.92	651.19 -6.57%	106.83 -9.41%
All cattle activities	2270.96	994.91	2269.05 -0.08%	959.98 -3.51%
Other animals	407.62	3136.75	385.42 -5.45%	3040.95 -3.05%
All Dairy	2482.60	876.99	2471.64 -0.44%	853.15 -2.72%
Other arable crops	-76.58	89.70	-45.86 40.11%	87.84 -2.07%
Fodder activities	203.79	853.15	264.09 29.59%	849.32 -0.45%
Vegetables and Permanent crops	41484.92	22.97	41498.07 0.03%	22.98 0.04%
Oilseeds	474.62	154.56	494.29 4.15%	154.80 0.16%
Cereals	257.58	1393.46	300.61 16.71%	1417.43 1.72%

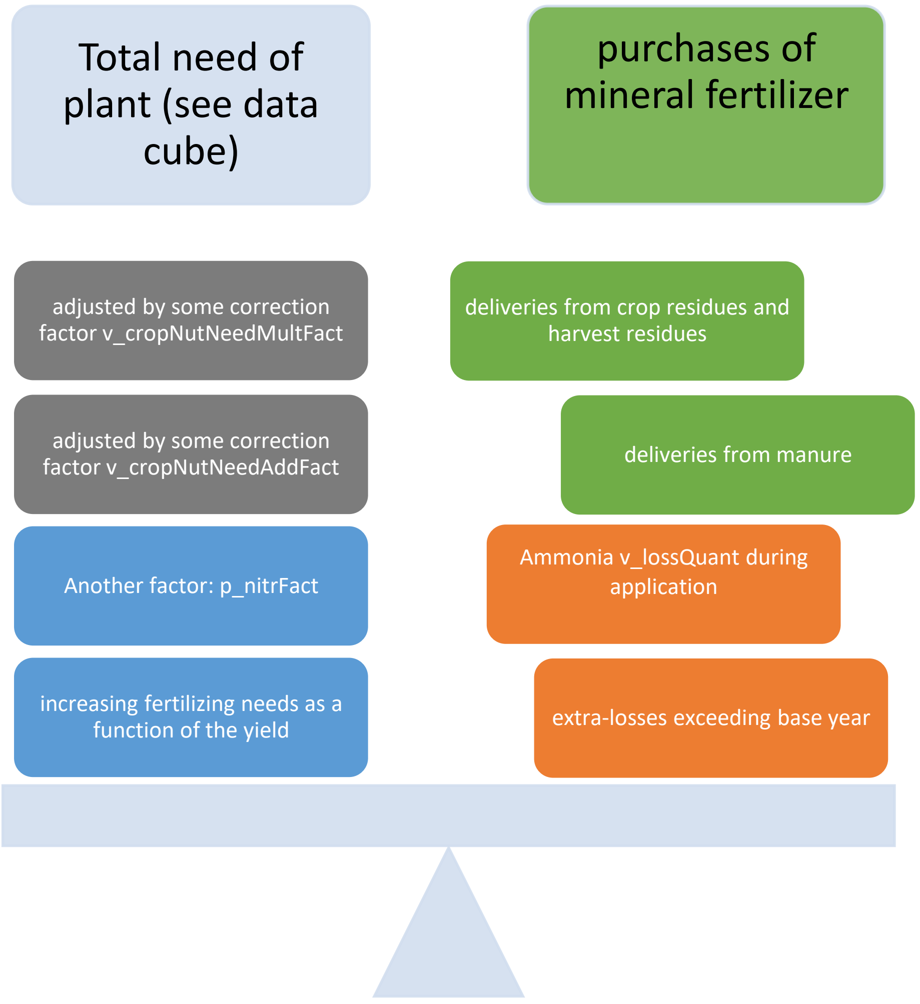
Fertilizer balance

- Currently under revision for Star2
- Balance is done not at activity level but on crop group level (NGRP)
- FNUT->NITF, "Nitrogen in fertiliser“, PHOF "Phospate in fertiliser [P2O5]“ and POTF "Potassium in fertiliser [K2O]“
- Mineral fertilizer is delivered from Supply balance

Balance

- Total need of plant (adjusted by some correction factor (v_cropNutNeedMultFact, v_cropNutNeedAddFact, p_nitrFact))

```
---------------------------------------------------------- 351 line(s) not displayed -----------------------------
   NUTNED_(RUNR,NGRP,FNUT) $ SUM( (CACT_TO_NGRP(MCACT,NGRP),A) $ p_technFact(RUNR,MCACT,"LEVL",A), 1) ..

     SUM( (CACT_TO_NGRP(MCACT,NGRP),A) $ p_technFact(RUNR,MCACT,"LEVL",A), v_actLevl(RUNR,MCACT,A)
*
*        --- total nutrient need of crops (retention - biolog. fixation for certain crops) * nutrient factor
*
          * (DATA(RUNR,MCACT,FNUT,"Y")
             * (1-p_nitrFact(RUNR,MCACT,"BioFix") $ SAMEAS(FNUT,"NITF")
             )
              * ^(RUNR,FNUT,A)
*
*        --- plus constant term of nutrient factor
*
              + v_cropNutNeedAddFact(RUNR,FNUT))
*
*        --- soil property effect
*
             * (p_nitrFact(RUNR,"ALL","DEFR") $ SAMEAS(FNUT,"NITF") + 1 $ (NOT SAMEAS(FNUT,"NITF")))
*
*        --- increasing fertilizing needs as a function of the yield
*
                                   * SQRT(
                                        [1.+p_technFact(RUNR,MCACT,"Yield",A)
                                           + 0.2 $ SAMEAS(MCACT,"GRAI") - 0.2 $ SAMEAS(MCACT,"GRAE")
                                        ] $ DATA(RUNR,MCACT,FNUT,"Y")
                                       )
*
*        --- factor describing the changes in removal of nutrients
*            resulting from yield difference stored in TECHF
*
                                  *(p_technFact(RUNR,MCACT,FNUT,A)+1.)
*
*
         )  * 0.001 =E=
*
*    --- purchases of anorganic fertiliser minus Ammonia v_lossQuant during application
*
       + v_fertDist(RUNR,NGRP,FNUT,"Mine") * (1.-p_emiLoss(RUNR,"NETF","N","GasRunTot") $ SAMEAS(FNUT,"NITF") )
*
*
*    --- extra-losses exceeding base year
*
       - v_fertDist(RUNR,NGRP,FNUT,"Loss")
*
*    --- deliveries from manure
*        corrected for further v_lossQuant (NOx, leaching)
*        (attention: a part of these v_lossQuant also hides in v_cropNutNeedMultFact)
*
       + v_fertDist(RUNR,NGRP,FNUT,"Excr") * SUM( FOUT_T_N(FOUT,FNUT),v_nutAvailFactExcr(RUNR,FOUT,"T"))
*
*    --- deliveries from crop residues (distribution across arable crops)
*
       + (v_fertDist(RUNR,NGRP,FNUT,"Cres")
             * SUM( FOUT_T_N(FOUT,FNUT),v_nutAvailFactCRes(RUNR,FOUT,"T"))
               * (1.-p_emiLoss(RUNR,"NETF","N","GasRunTot") $ SAMEAS(FNUT,"NITF") )) $ (NOT PERM_NGRP(NGRP))
*
*    --- delivery from harvest residues and atmospheric deposition (only for grass land, where it remains)
*
       + SUM( (CACT_TO_NGRP(MCACT,NGRP),FOUT_T_N(FOUT,FNUT),A)
              $ (p_technFact(RUNR,MCACT,"LEVL",A) and PERM_NGRP(NGRP)),
                  v_actLevl(RUNR,MCACT,A)
                  * DATA(RUNR,MCACT,FOUT,"Y")
                  * (p_technFact(RUNR,MCACT,FOUT,A)+1.)
                  * (1.-p_emiLoss(RUNR,"NETF","N","GasRunTot") $ SAMEAS(FNUT,"NITF") )
                  * v_nutAvailFactCRes(RUNR,FOUT,"T")) * 0.001;
*
```



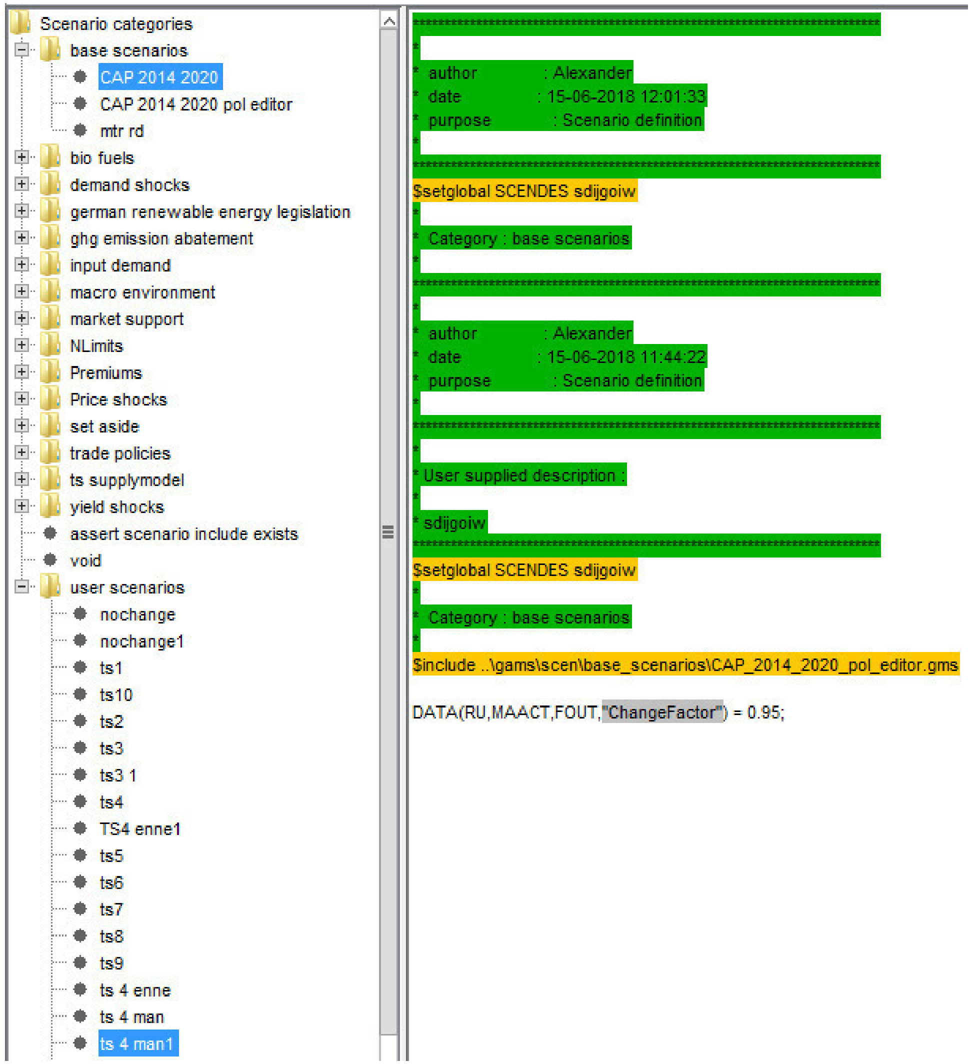
```
$           include 'supply\supply_model.gms';
```

Area	Equations	Variables	Parameters or scalars	Exercise
Distribution of mineral and organic fertilizer to crops	fertDistMine_ ineral fertilizer distributed to group of crops must exhaust total mineral N sales	v_fertDist(RUNR,NGRP, FNUT,"Mine") v_netPutQuant(RUNR, FNUT)	No	
	NUTNED_ nutrient need balance for group of crops		Nutrient factors: p_nitrFact v_cropNutNeedAddFact v_cropNutNeedMultFact Nutrient need of crop: DATA(RUNR,MCACT,FNUT,"Y")	Exercise
	NUTMIN_ Minimum use of mineral fertilizer	v_actLevl(MCACT) v_fertDist(RUNR,NGRP, FNUT,"Mine")	p_minShareMinFert(MSACT,MCACT,A, FNUT) v_minShareMinFertCorr DATA(RUNR,MCACT,FNUT,"Y") p_nitrFact	Exercise

```
$           include 'supply\supply_model.gms';
```

Area	Equations	Variables	Parameters or scalars	Exercise
mapping	fertDistExcr_ Total crop available nutrients from manure must be distributed to different crop groups	excrements distributed summed up over crop groups		
mapping	fertDistCres_ Total nutrients from crop resiudes and atmosperic deposition must be distributed to different crop groups	Only non permanent v_actLevl(nonPermAct)	DATA(RUNR,noPermCact,FOUT,"Y")	Option
mapping	ManureNPK_ Definition of total manure output of animals	v_ManureNPK v_actLevl(RUNR,MAACT	v_ManureNPKintraTrade DATA(RUNR,MAACT,FOUT,"Y") p_emiLoss	Manure from animals

Scenario: Reduction of manure output from animals by 5%



Results: Reduction of manure output from animals by 5%

Manure output per animal [0]

Region: Denmark

Year: 2030

Percentage diff. to Scen nochange

View: Tab

	Nitrogen [kg N / head or 1000 heads]	Phospate [kg P2O5 / head or 1000 heads]	Potassium [kg K2O / head or 1000 heads]	Nitrogen [kg N / head or 1000 heads]	Phospate [kg P2O5 / head or 1000 heads]	Potassium [kg K2O / head or 1000 heads]
All cattle activities	110	66	143	104 (-5%)	63 (-5%)	136 (-5%)
All Dairy	114	69	145	109 (-5%)	65 (-5%)	138 (-5%)
Other animals	57	23	23	54 (-5%)	22 (-5%)	22 (-5%)

Results: Reduction of manure output from animals by 5%

Environmental indicators [0]

Region: Denmark — Year: 2030

	nochange		ts_4_man1	
	Total [in 1000t]	Amount per ha [in 1000t/ha]	Total [in 1000t]	Amount per ha [in 1000t/ha]
GHG emissions from agricultural input industries in CO2 equivalents	540.58	204.71	572.50 5.90%	216.78 5.90%
Ammonium output	63.75		60.83 -4.58%	
CH4 Total emissions	235.54	89.20	234.84 -0.30%	88.92 -0.30%
N2O Total emissions	18.24	6.91	17.79 -2.43%	6.74 -2.44%
CO2 Total emissions	-5441.74	-2060.71	-5406.96 0.64%	-2047.39 0.65%
GHG emissions from agricultural input industries in CO2 equivalents	540.58	204.71	572.50 5.90%	216.78 5.90%

Economic Accounts for Agriculture [0]

Region: Denmark — Year: 2030

	nochange		ts_4_man1	
	Gross EAA value [Mio Euro]	Quantity [1000 t]	Gross EAA value [Mio Euro]	Quantity [1000 t]
Manure phospate	548.39	209.71	529.00 -3.54%	202.30 -3.54%
Manure output	1622.07	1151.78	1571.69 -3.11%	1117.43 -2.98%
Manure nitrate	648.98	506.81	629.15 -3.05%	491.33 -3.05%
Manure potassium	424.70	435.25	413.53 -2.63%	423.80 -2.63%

PMP term for activities

Constant

quadratic term

By group of activities defined in GRPLEVL

```
------------------------------------------------------------------------------------------------ 71 line(s) not displ
     QUADRA_           "Quadratic and linear PMP terms activities"
------------------------------------------------------------------------------------------------ 2020 line(s) not disp
QUADRA_(RUNR) ..  v_sumOfPmpTermsLevls(RUNR) =E= (

            Linear part
            (specified in tems of mPact to cover the case of waterm      e==on where qAct
            sum((mpact,A) $ (p_technFact(RUNR,mpact,"LEVL",A) $        MPStep1 ne 1)),
                v_actLevl(RUNR,mpact,A)*p_pmpCnst(RUNR,mpact

            plus the v_sumOfPmpTermsLevlstic part multi    ed by 0.5 for easy      vatives
            + 0.5 * (

            multiplied by the diagonal effects (direct own area cost     rect)
                sum((qact,A,A1) $ (p_technFact(RUNR,qact,"LEVL"        p_technFact(RUNR,qact,"LEVL",A1)),
                   v_actLevl(RUNR,qact,A)*v_actLevl(RUNR,qact
                   * (      p_pmpQuadTechn(RUNR,qact,A,A1)
                          + (SUM(EPRD_TO_GRP(qact,GRP),p    mpQuadPact(RUNR,GRP,GRP))/p_technFact(RUNR,qact,"LEVL",A)) $ (SAMEAS(A,A1) $ (p_useCropSpecificQMatrix eq 0))
                         + (p_pmpQuadPact(RUNR,qact,qact)/p_technFact(RUNR,qact,"LEVL",A))               $ (SAMEAS(A,A1) $ (p_useCropSpecificQMatrix eq 1))))

            plus the crop group effects
              + sum((GRP,GRP1) $ (p_useCropSpecificQMatrix eq 0),
                      v_actLevl(RUNR,GRP,"T")*v_actLevl(RUNR,GRP1,"T")*
                          (p_pmpQuadPact(RUNR,GRP,GRP1) $ (p_ordGrp(GRP) le p_ordGrp(GRP1))
                         + p_pmpQuadPact(RUNR,GRP1,GRP) $ (p_ordGrp(GRP) gt p_ordGrp(GRP1))))

            wipes out the mapping of the diagonal element
             - SUM(EPRD_TO_GRP(qact,GRP) $ (p_useCropSpecificQMatrix eq 0),
                  SQR(SUM(A $ p_technFact(RUNR,qact,"LEVL",A), v_actLevl(RUNR,qact,A)))
                                     * p_pmpQuadPact(RUNR,GRP,GRP))

            plus the full Q effect
              + sum((qact,qact1) $ ( (NOT SAMEAS(qact,qact1)) $ (p_useCropSpecificQMatrix eq 1)),

                      SUM(A $ p_technFact(RUNR,qact,"LEVL",A),  v_actLevl(RUNR,qact,A))
                    * SUM(A $ p_technFact(RUNR,qact1,"LEVL",A), v_actLevl(RUNR,qact1,A))

                      *  (p_pmpQuadPact(RUNR,qact,qact1) $ (qact.pos le qact1.pos)
                        + p_pmpQuadPact(RUNR,qact1,qact) $ (qact.pos gt qact1.pos)))


            plus the effect of changing irrigation / rainfed shares

------------------------------------------------------------------------------------------------ 30 line(s) not displ
            );
----------------------------------------------------------------------------------------------
        GRPLevl_          "Definition of crop groups for variable cost function"
----------------------------------------------------------------------------------------------


  GRPLEVL_(RUNR,GRP) $ (SUM( (qact,A) $ (EPRD_TO_GRP(qact,GRP) $ p_technFact(RUNR,qact,"LEVL",A)), 1)
                            $ p_pmpQuadPact(RUNR,GRP,GRP) $ (not p_useCropSpecificQMatrix) ) ..

      v_actLevl(RUNR,GRP,"T") =E= SUM( (qact,A) $ (EPRD_TO_GRP(qact,GRP) $ p_technFact(RUNR,qact,"LEVL",A)), v_actLevl(RUNR,qact,A));
```

PMP for feed

```
*
*  --- PMP for feed use per per region, activity and technology
*
   QUADRF1_(RUNR,MAACT %addtimedim%) $ p_technFact(RUNR,MAACT,"LEVL","T") ..
*
     v_pmpCostFeedPerAnim(RUNR,MAACT,"T") =E=

        SUM(  (FEED) $ p_maxFeedShare(RUNR,MAACT,"T",FEED %addtimedim%),
             v_feedInpCoeff(RUNR,MAACT,"T",FEED %addtimedim%)
                *     (p_pmpFeedInpCoeff(RUNR,MAACT,"T",FEED,"CNST")
                     + 0.5 * v_feedInpCoeff(RUNR,MAACT,"T",FEED %addtimedim%)*p_pmpFeedInpCoeff(RUNR,MAACT,"T",FEED,"SLOP")) );



------------------------------------------------------------------------- 421 line(s) not displayed -----------------------------------------------------------------------
         QUADRF_            "Quadratic and linear PMP terms feed"
------------------------------------------------------------------------ 1732 line(s) not displayed -----------------------------------------------------------------------
    QUADRF_(RUNR) ..
*
        v_sumOfPmpTermsFeed(RUNR) =E= SUM(  (MAACT) $ p_technFact(RUNR,MAACT,"LEVL","T"), v_actLevl(RUNR,MAACT,"T") * v_pmpCostFeedPerAnim(RUNR,MAACT,"T") );
*
```

PMP for feed

Area	Equations	Variables	Parameters or scalars	Exercise
Activity and feed mix calibration	QUADRA_ "Quadratic and linear PMP terms activities“ QUADRF_ , QUADRF1 "Quadratic and linear PMP terms feed"	v_sumOfPmpTermsFeed v_actLevl v_pmpCostFeedPerAnim	p_pmpCnst p_pmpQuadTechn p_pmpQuadPact p_pmpFeedInpCoeff (linear and constant)	Option change slope of feed costs

Questions

- How can we steer the mineral fertilizer consumption
- In which parameter are the variable input costs accounted
- Is land fixed or variable in CAPRI supply model
- What is the difference between non tradable fodder and tradable fodder in CAPRI
- Can we use more young animal in a region as we produced

…

Questions and answers

- How can we steer the mineral fertilizer consumption
  - v_netPutQuant, add constraint, change the price (“UVAG”) for NPK at farm level
- In which parameter are the variable input costs accounted
  - p_linObjecont
- Is land fixed or variable in CAPRI supply model
  - depends on settings eq 1(eq 0) for p_landsFixed
- What is the difference between non tradable fodder and tradable fodder in CAPRI
  - depends on settings eq 1(eq 0) for p_landsFixed
- Can we use more young animal in a region as we produced
  - yes as they are tradable due to an endogenous market for young animals

…