Using price sensitivity analysis with the supply models to parameterize the market model

- Wolfgang Britz, March 2013 –

Content

Inhalt

Background and motivation

The interaction between the market and the supply module(s) in CAPRI is based on sequential iteration where the market models are calibrated against the latest points in the price / quantity space simulated with the supply models. Afterwards, the market models are solved. The supply models simulate then again a solution, at prices which are an average from the last iterations. The process is repeated until differences between the last two iterations both in prices and quantities fall beyond a threshold.

The speed of the process, i.e. the number of iterations necessary and how fast the models solve, depends to some extent how well the supply and feed demand reactions in the market model mimic the behavior of the supply models. If we just look at an own price effects and supply, differences in reactions root in differences in the (unknown) slope of the implicit marginal cost curve in the programming models and the slope used for the supply function in the market model. So far, the slope terms were derived from the parameters of the programming models (see http://www.capri-model.org/docs/YANI.pdf).

Since a while, the supply models in conjunction with the so-called scenario solver from GAMS can be used for sensitivity analysis with regard to prices (see http://www.capri-model.org/docs/priceScens.pdf). That approach is now used to derive the parameters for supply and feed demand in the market model. At the same time, the separate market model for young animals is given up and the young animal prices are simulated in the global market model together will all other markets.

The aims of the changes are the following:

- Improve the representation of supply and demand of the countries with supply models in the market model to speed up convergence, also to generate a stand-alone version of the market model.
- Remove code doing similar calculations – the price sensitivity experiments deliver the same information then the parametric derivation of the Hessian.
- Remove the separate young animal market model.

Changes in the market model

The main changes in the market model are:

1. The young animals (xxOmYani) are now part of the products covered by the market model

```
set xxOmYani(XX) /
     YCOW   Young cow
     YBUL   Young bull
     YHEI   Young heifer
     YCAM   Young male calf
     YCAF   Young female calf
     YPIG   Young piglet
     YLAM   Young lamb
/;
alias(xxOmYani,xxOmYani1);
```

2. The market model comprises an equation to clear animal markets. As there is by definition only use in agriculture and no trade between trade blocks, a separate equation seemed more transparent compared to a solution where young animals were included in almost any equation of the market model:

```
*
 yaniMarket_(RM,xxOmYani) $ ( sum(RM_TO_RMS(RM,RMSSUP),1) $ XXX(xxOmYani)) ..
*
       sum(RM_TO_RMS(RM,RMSSUP),

            p_cnstNQSupp(RMSSUP,xxOmYani)

              + sum(YY1 $ (DATA(RMSSUP,"PPri",YY1,"CUR") $ (NOT SAMEAS(YY1,"INPE"))),
                           p_hessNQSupp(RMSSUP,xxOmYani,YY1,"CUR")
                             * v_prodPrice(RMSSUP,YY1)/v_prodPrice(RMSSUP,"Inpe"))

            + SUM( feed_trd $ p_hessNQSupp(RMSSUP,xxOmYani,feed_trd,"Cur"),
                           p_hessNQSupp(RMSSUP,xxOmYani,feed_trd,"CUR")
                             * v_feedBlkPrice(RMSSUP,feed_trd)/v_prodPrice(RMSSUP,"Inpe"))
       )
             / (sum(RM_TO_RMS(RM,RMSSUP),
                 DATA(RMSSUP,"GROF",xxOmYani,"CUR")
                  +SUM(O_TO_YANI(xxOmYani,IYANI), DATA(RMSSUP,"GROF",IYANI,"CUR")))*0.5 + 1)

          =E=  sum(RM_TO_RMS(RM,RMSSUP),
                DATA(RMSSUP,"GROF",xxOmYani,"CUR")
                 -SUM(O_TO_YANI(xxOmYani,IYANI), DATA(RMSSUP,"GROF",IYANI,"CUR")))
             / (sum(RM_TO_RMS(RM,RMSSUP),
                 DATA(RMSSUP,"GROF",xxOmYani,"CUR")
                  +SUM(O_TO_YANI(xxOmYani,IYANI), DATA(RMSSUP,"GROF",IYANI,"CUR")))*0.5 + 1);
*
```

3. The supply functions now comprise the prices of the bulk feed stuff (FCER …) as additional arguments. Please note that also cross-price effects with young animals and raw milk are now integrated (there is no longer a separate equation for raw milk supply):

```
*
 ProdNQ_(RMS,XX1) $ (  (     (NOT xx_no_prodNQ_equation(XX1,RMS))
                           $ (not sameas(XX1,"FEDE"))
                           $  (XXX1(XX1) or XXX1("Fede")
                        )
*
*                      --- production must be non-fixed
*
                       $ (v_prodQuant.lo(RMS,XX1) ne v_prodQuant.up(RMS,XX1))
*
*                      --- production must be given in calibration point
*
                       $ DATA(RMS,"Prod",XX1,"CUR"))   ..

      (v_prodQuantNeg(RMS,XX1) $ ( v_prodQuant.LO(RMS,XX1) LE 0)
 +     v_prodQuant(RMS,XX1)     $ ( V_prodQuant.LO(RMS,XX1) GT 0)) / (DATA(RMS,"Prod",XX1,"CUR")+0.1)

       =E= (p_cnstNQSupp(RMS,XX1)
            + SUM( YY1 $ (DATA(RMS,"PPri",YY1,"CUR") $ (NOT SAMEAS(YY1,"INPE"))),
                           p_hessNQSupp(RMS,XX1,YY1,"CUR")
                         * v_prodPrice(RMS,YY1)/v_prodPrice(RMS,"Inpe"))


            + SUM( feed_trd $ p_hessNQSupp(RMS,XX1,feed_trd,"Cur"),
                           p_hessNQSupp(RMS,XX1,feed_trd,"CUR")
                         * v_feedBlkPrice(RMS,feed_trd)/v_prodPrice(RMS,"Inpe"))
                         )
                         /(DATA(RMS,"Prod",XX1,"CUR")+0.1);
*
```

The changes must be necessarily reflected in different programs, e.g. in “arm\prep_market.gms” or “arm\simu_calibration_to_supply.gms” where the constant terms (p_cnstNQSupp) are defined.

Changes in baseline mode

In baseline mode, price sensitivity experiments are conducted (see “priceScen\runExperiments.gms”). First the experiments are defined for the country: each price present in the supply models is increased by 10%. Afterwards, the scenario solved is called, and finally, the model solved again at unchanged prices. The same code is also comprised in the separate threads.

```
   if ( p_isLastStep,

       option kill=p_uvagScens;
       p_uvagScens(OM_OBJE_SCEN,MSACT,OM_OBJE) $ RUNR_OMOBJE(RU,OM_OBJE) = DATA(MSACT,"UVAG",OM_OBJE,"Y");
       p_uvagScens(OM_OBJE_SCEN,MSACT,OM_OBJE) $ sameas(om_obje_Scen,om_obje) = p_uvagScens(OM_OBJE_SCEN,MSACT,OM_OBJE)  * 1.10;

 $batinclude 'util\title.gms' "'Price sensitivity analysis with supply model for '" RU_SMALL.tl
 *
          p_useUvagScen = 1;
 *
 *        --- use scenario solver (= hot model updates) to perform
 *            sensitivity analysis for prices

          SOLVE m_capmodQ USING NLP MAXIMIZING v_obje SCENARIO uvagDict;
 *
 *        --- collect results
 *
          execute_load 'scenrep.gdx', scenRep;
 *
 *        --- solve again at unchanged prices
 *
          p_useUvagScen = 0;
          SOLVE m_capmodQ USING NLP MAXIMIZING v_obje;
 *

   );
```

That naturally requires processing time, in tests on rather larger multi-core computers, the baseline calibration took about 3 minutes longer.

Afterwards, the results are mapped into definition of the market model (pricescen\reporting.gms):

```
p_hessNqSupp(RMS,XXX,YYY,"CAL") $ sum(sameas(RMS,MS),1)
     = sum( sameas(RMS,MS),
                              sum( (cap_to_trd(ROWS,XXX),OM_OBJE_SCEN) $ (cap_to_trd(OM_OBJE_SCEN,YYY) $ DATA(MS,"UVAG",OM_OBJE_SCEN,"Y")),

                                  [    ( p_netPutScens(OM_OBJE_SCEN,MS,ROWS) - v_netPutQuant.l(MS,ROWS) )
                                     / ( DATA(MS,"UVAG",OM_OBJE_SCEN,"Y")*0.1 )

                                       + sum(pact_to_y(mcact,rows), p_yieldElas(MS,mcact))  $ sameas(XXX,YYY)
                                        *  v_netPutQuant.l(MS,ROWS)/DATA(MS,"UVAG",OM_OBJE_SCEN,"Y") ] * p_inflationFactor(MS)));
```

In order to so, as seen above, we aggregate the simulated quantity changes (p_netPuctScens) against the base (v_netPutQuant) divided by the price change and add the effect of the yield elasticity to derive a Hessian matrix. The latter expresses the change (typically delta 1000 tons) in quantities if an own or cross price changes change (delta 1 Euro).

Some specific code is necessary to deal with sugar which is not shown here.

Finally, the resulting Hessian matrices are stored back to disk:

```
execute_unload "%results_out%\arm\elas%BAS%%SIM%.gdx" p_ElasFeed,p_elasDem,p_eLasSupp,p_elasProc,p_elasDair,
                                                 P_pdGl,p_pbGL,p_Lus,P_luf,P_luc,P_lum,
                                                 p_hessNQSupp,p_hessNQFeed,p_hessNQDairy,p_hessNQProc,
                                                 p_nInfesTrimPar,p_aveDiff;
```

They are then loaded by “arm\market1.gms”.

Effects on simulation behavior

The model was tested on a range of scenarios such as removal of Pillar I and WTO. It converged typically in about 10 iterations, showing rather limited changes after iteration 3. That seems to show that the basic idea was correct that a parameterization based on the price sensitivity experiments could improve convergence.

The report below shows the maximal change in quantities and prices for a simulation of a 50% reduction in pillar I premiums.

It might be good to discuss here how the changes in the first iterations can be understood. In order to understand that discussion of the iteration report, it must first be noted that activity levels and production as well as feed quantities are always reported as simulated with the supply models.

If the shock stem from the supply side, e.g. a change in premiums as in the experiment shown above, the reports can be interpreted as follows:

- Step1 reports changes in activity levels, production and feed quantities at unchanged prices, but with the shock implemented. The price changes reported are based on a re-calibrated market model which reflects these quantity changes. [The large change (143%) stems from reduction of the pulses areas in Latvia from 2000 ha to 300 ha].

- Step 2 then reports how the activity levels, production and feed quantities react to prices from Step 1, prices are simulated based on these updated quantities. The reader should note that also the premiums are updated, the reaction on the supply side is hence is combined effects of the price and premium changes.

Let's start with the case if the shock emanates from the market model.

- Activity levels, production and feed quantities are reported as unchanged in step 1, while the price changes stem from the market model (simulated with the model calibrated against the baseline).
- Step 2 will then report the first time changes from the supply side.
- Step 3 gives then an indication how well the market model anticipated the changes on the supply side.

Step 3 hence delivers the first time an indication how well the market model mimics the supply and feed demand behavior of the programming models.

Summary and conclusions

The code changes documented implement price sensitivity experiments with the supply models in baseline mode to parameterize the supply and feed demand function in the market model for countries with programming models.

The changes (1) removes code (young animal market model, estimation of supply and feed demand parameters from supply model parameterization), (2) integrates from cross-price effects for countries with supply models into the market model, (3) seems to speed up convergences.