

Introduction to GAMS Modeling Language

Day III – 15.05.2025

PD Dr. Alexander Gocht

Dr. Ferike Thom

Thuenen Institute of Farm Economics, Braunschweig

What we covered so far

1) What is CAPRI?

2) Introduction to GAMS

- Main elements of a GAMS model
 - variables, parameters, equations, model, solver
- Improving the efficiency of the model
 - sets, subsets, alias, sum
 - prod, table, variable attributes, loop
- Data exchange with Excel
- Common income support schemes to MyFarm LP
- Production quota in MyFarm LP, GAMS conditionals, parameter dimensions

Content today

Extending the MyFarm LP model

- Ex 5: Crop nutrient need
- Ex 6: Fertilizer pricing
- Ex 7: Shocks to fertilizer prices

Compile time conditionals

Introduction to a Market Model (PE)

- Ex 8: MyMarketModel

Introduction to the CAPRI market model

- Ex 9: run the CAPRI market model (didactic version)

Extending the MyFarm Model

Ex 5: Crop nutrient need



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

Background/Approach: Nutrient need

- Crops need fertilization. The source of nutrient can come from different sources
- Animal manure, mineral fertilizer, biological fixation
- Many environmental indicators depend on the nutrient balance
- Aim of the exercise is to define:
 - N/P/K nutrient need of our crops
 - Adding three equation which sums the total N/P/K need during model solve
 - Write gams equation more compact

Ex 5: Crop Nutrient Need

- ✓ Define a new parameter, where the nutrient requirements are stored per crop and fertilizer type
- ✓ Define parameter always with **p_** at the beginning, similar **e_** and **v_** for equation and variables (improves readability)

```
Parameter p_nutrientNeedperCrop(*,cols) "kg per  
hectar crop need of nutrients";
```

```
p_nutrientNeedperCrop("NITF","wheat")      = 178;  
p_nutrientNeedperCrop("PHOF","wheat")      = 77;  
p_nutrientNeedperCrop("POTF","wheat")      = 151;  
p_nutrientNeedperCrop("NITF","barley")     = 122;  
p_nutrientNeedperCrop("PHOF","barley")     = 77;  
p_nutrientNeedperCrop("POTF","barley")     = 120;  
p_nutrientNeedperCrop("NITF","rapeseed")   = 218;  
p_nutrientNeedperCrop("PHOF","rapeseed")   = 119;  
p_nutrientNeedperCrop("POTF","rapeseed")   = 66;  
p_nutrientNeedperCrop("NITF","sugarbeet")  = 321;  
p_nutrientNeedperCrop("PHOF","sugarbeet")  = 178;  
p_nutrientNeedperCrop("POTF","sugarbeet")  = 280;
```

Ex 5: Crop Nutrient Need

- ✓ Define three new equations, one for each of the nutrients to multiply the hectare with the nutrient requirements
- ✓ Define a new variable that is responsible to sum up during optimisation the total crop need

equation

```
e_NUTNED_NITF "equation for N",  
e_NUTNED_PHOF "for P",  
e_NUTNED_POTF "for K";
```

alternative:

```
equation e_NUTNED_NITF "equation for N";  
equation e_NUTNED_PHOF "for P";  
equation e_NUTNED_POTF "for K";
```

Ex 5: Crop Nutrient Need

- ✓ The equation should sum over all crops take the optimized hectare and multiply it with the parameter which contains the crop nutrient need.

```
variable v_Nutrientneed(*) "total fertilzer required";
```

```
e_NUTNED_NITF.. sum( cols, v_actLevl(cols) *  
p_nutrientNeedperCrop("NITF",cols)) =E= v_Nutrientneed("NITF");
```

```
e_NUTNED_PHOF.. sum( cols, v_actLevl(cols) *  
p_nutrientNeedperCrop("PHOF",cols)) =E= v_Nutrientneed("PHOF");
```

```
e_NUTNED_POTF.. sum( cols, v_actLevl(cols) *  
p_nutrientNeedperCrop("POTF",cols)) =E= v_Nutrientneed("POTF");
```

Ex 5: Crop Nutrient Need

- ✓ Define a new model and solve it and report what is the production program and answer the questions

```
model myfarm_Nutrientneed /all/;  
solve myfarm_Nutrientneed using lp maximizing v_obje;
```

Q7.1: How much N/P/K in tones is used with the baseline production program?

Ex 5: Crop Nutrient Need

- ✓ We now show the real power of GAMS compared to other programs; We reduce the three similar equations by using additional dimensions for the equations, the dimension is defined over the nutrients using a set FNUT.
- ✓ First we define the nutrients as a set.

```
set FNUT / NITF          "Nitrogen in fertiliser"  
         PHOF          "Phospate in fertiliser [P2O5]"  
         POTF          "Potassium in fertiliser [K2O]" /;
```

- ✓ Define an equation but with one dimension for FNUT and a new model

```
equation e_NUTNED_(FNUT);
```

```
e_NUTNED_(FNUT)..    sum( cols, v_actLevl(cols) *  
p_nutrientNeedperCrop(FNUT,cols) ) =E= v_Nutrientneed(FNUT);
```

```
model myfarm_Nutrientneed_compact /all - e_NUTNED_NITF -  
e_NUTNED_PHOF - e_NUTNED_POTF/;
```

```
solve myfarm_Nutrientneed_compact using lp maximizing v_obje;
```

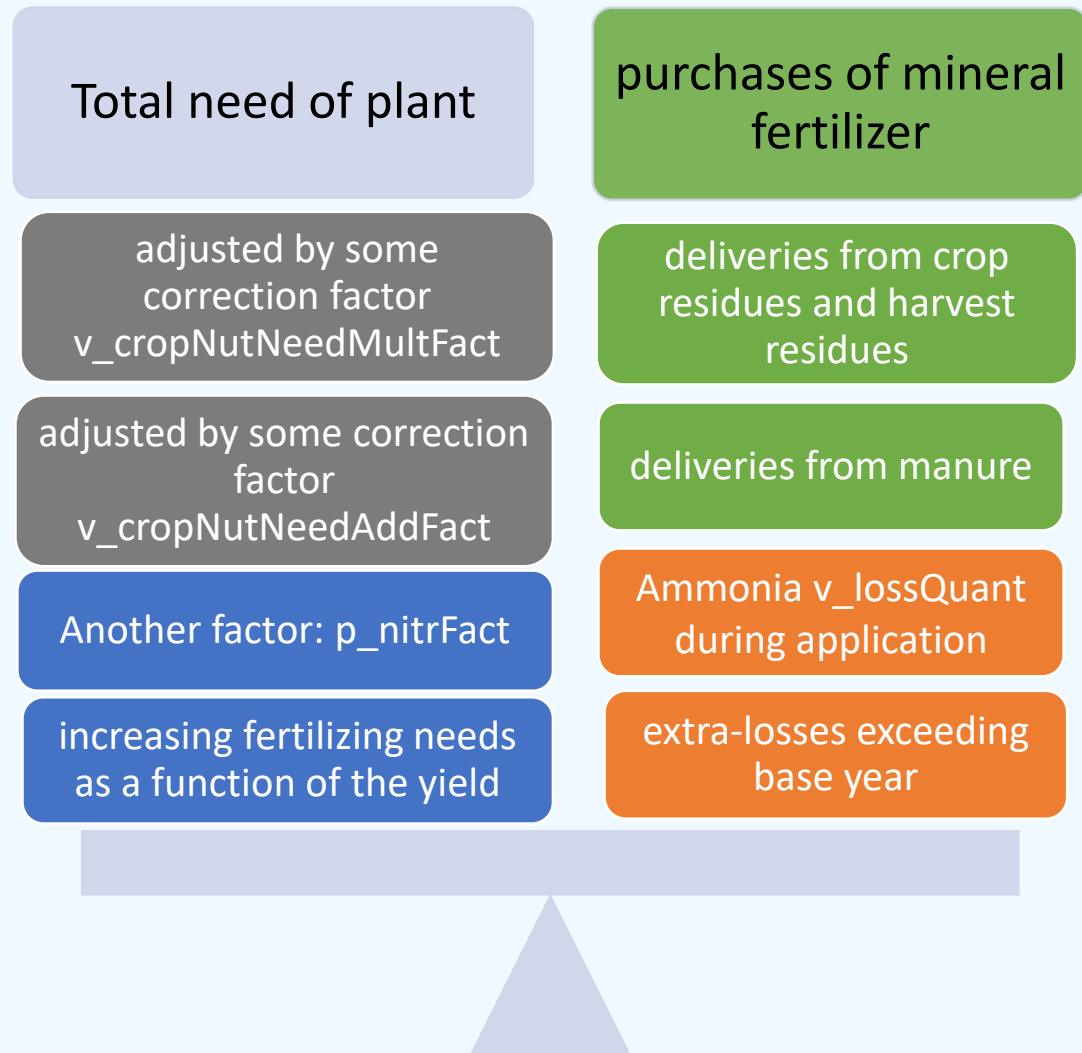
Ex 5: Crop Nutrient Need

- Q7.2: Did the N/P/K in tones is used with the baseline production program changed?
- Q7.3: Do a new simulation and restrict the nutrient requirements for NITF by 10%. How did the production program change? Hint: variables can be upper bounded during solution like

```
v_Nutrientneed.up("NITF") =  
v_Nutrientneed.l("NITF")*0.9;  
solve myfarm_Nutrientneed using lp maximizing v_obje;  
p_results(cols,"compact_10") = v_actLevl.l(cols);  
p_results(FNUT,"compact_10") = v_Nutrientneed.l(FNUT);  
*reset the upper bound to positive infitity  
v_Nutrientneed.up("NITF") = +inf;
```

Ex 5: Crop Nutrient Need

Nutrient balance in CAPRI



```

NUTNED_(RUNR,NGRP,FNUT) $ SUM(
(CACT_TO_NGRP(MCACT,NGRP),A) $
[p_techFact(RUNR,MCACT,"LEVL",A) and
%data%(RUNR,MCACT,"levl","Y")], 1) ..

SUM( (CACT_TO_NGRP(MCACT,NGRP),A) $
p_techFact(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")
)
* v_cropNutNeedMultFact(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_techFact(RUNR,MCACT,"Yield",A)
+ 0.2 $ SAMEAS(MCACT,"GRAI") - 0.2 $ SAMEAS(MCACT,"GRAE")
] $ %data%(RUNR,MCACT,FNUT,"Y")
)
)=E=

```

Continue>>>

```

*
* --- 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") )
*
* --- deliveries from manure
+ 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_techFact(RUNR,MCACT,"LEVL",A) and PERM_NGRP(NGRP)),
v_actLevl(RUNR,MCACT,A)
* %data%(RUNR,MCACT,FOUT,"Y")
* (p_techFact(RUNR,MCACT,FOUT,A)+1.)
* (1.-p_emiLoss(RUNR,"NETF","N","GasRunTot") $
SAMEAS(FNUT,"NITF") )
* v_nutAvailFactCRes(RUNR,FOUT,"T"));

```

Ex 5: Crop Nutrient Need

Results

	Nutrientneed	compact	compact_10
barley	145.098	145.098	162.85
sugarbeet	54.902	54.902	37.1505
NITF	35325.5	35325.5	31792.9
PHOF	20945.1	20945.1	19152.2
POTF	32784.3	32784.3	29944.1

Content today

Extending the MyFarm LP model

- Ex. 5: Crop nutrient need
- Ex. 6: Fertilizer pricing
- Ex. 7: Shocks to fertilizer prices

Compile time conditionals

Introduction to a Market Model (PE)

- Ex. 8: MyMarketModel

Introduction to the CAPRI market model

- Ex. 9: run the CAPRI market model (didactic version)

Extending the MyFarm Model

Ex 6: Fertilizer pricing



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

Ex 6: Fertilizer Pricing

- ✓ Crop nutrient need is assumed to be **covered by fertilizer purchased**
- ✓ This are **costs** which **can impact the decision of a farmer** as certain crops need more and others less fertilizer particular regarding nitrogen fertilizer
- ✓ Aim is to **introduce prices for N/P/K** and **include this as additional costs in the objective function** of the farmer

Ex 6: Fertilizer Pricing

- ✓ Let's assume that the requirements need to be fulfilled by purchasing fertilizer to a given price.
- ✓ Please define a price parameter for our nutrients

```
parameter p_fertilizer_price(FNUT);  
p_fertilizer_price("NITF") = 0.18;  
p_fertilizer_price("PHOF") = 0.25;  
p_fertilizer_price("POTF") = 0.11;
```

Then define a new objective function

```
equation e_obj_fertilizer "new objective for including costs of fertilizer";  
e_obj_fertilizer .. v_obje =E= sum (cols, v_actLevl(cols)* p_UVAG(cols)) -  
sum(FNUT, v_Nutrientneed(FNUT) * p_fertilizer_price(FNUT));  
model myfarm_fert /all - e_obj - e_NUTNED_NITF -e_NUTNED_PHOF -  
e_NUTNED_POTF/;  
solve myfarm_fert using lp maximizing v_obje;
```

Minus the costs for nutrients

Add some code line for reporting crops levels and nutrient need as we have learnt to see the effect in a parameter as scenarios

Ex 6: Fertilizer Pricing

- Q6.1: How does the cropping program changes when accounting for fertilizer costs? What could be the reason?

Ex 6: Fertilizer Pricing

Results

	Nutrientneed	compact	compact_10	fertcost
barley	145.098	145.098	162.85	200
sugarbeet	54.902	54.902	37.1505	
NITF	35325.5	35325.5	31792.9	24400
PHOF	20945.1	20945.1	19152.2	15400
POTF	32784.3	32784.3	29944.1	24000

Content today

Extending the MyFarm LP model

- Ex. 5: Crop nutrient need
- Ex. 6: Fertilizer pricing
- Ex. 7: Shocks to fertilizer prices

Compile time conditionals

Introduction to a Market Model (PE)

- Ex. 8: MyMarketModel

Introduction to the CAPRI market model

- Ex. 9: run the CAPRI market model (didactic version)

Extending the MyFarm Model

Ex 7: Fertilizer price shock



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

Ex 7: Fertilizer price shock

- ✓ We use a loop statement to simulate different price for fertilizer N and observe the cropping pattern, shadow value changes and total gross margin changes using a parameter and execute the results to GDX

Ex 7: Fertilizer price shock

✓ Please define a set and a parameter used to change the price

```
set steps/1*5/;  
parameter p_steps(steps);
```

Please define for each run in the loop a change factor

```
p_steps("1")= 100;  
p_steps("2")= 1.;  
p_steps("3")= 0.5;  
p_steps("4")= 0.1;  
p_steps("5")= 0.0;
```

Please define a reporting parameter with three dimensions

```
parameter p_reporting(steps,*,*);parameter p_fertilizer_price_ori(FNUT);  
*store value  
p_fertilizer_price_ori(FNUT)=p_fertilizer_price(FNUT);  
loop(steps,  
  p_fertilizer_price("NITF") = p_fertilizer_price_ori("NITF") * p_steps(steps);  
  Solve myfarm_fert using lp maximizing v_obje;  
  p_reporting(steps,cols,"prod") = v_actLevl.l(cols);  
  p_reporting(steps,"total","GM") = v_obje.l;  
  p_reporting(steps,cols,"Shadow") = v_actLevl.m(cols) ;  
  p_reporting(steps,"price NITF","shock") = p_fertilizer_price_ori("NITF") *  
p_steps(steps);  
);  
execute_unload "all.gdx";
```

Ex 7: Fertilizer price shock

- Q7.2: Simulate a fertilizer tax for nitrogen "NITF" by decreasing and increasing the fertilizer price stepwise
- Q7.2: Draw or show the relationship between total gross margin and the fertilizer tax
- Q7.3: When the fertilizer price for nitrogen increases drastically what happens with the production program

Ex 7: Fertilizer price shock

Results

	prod		GM	Shadow				shock
	barley	sugarbeet	total	wheat	barley	rapeseed	sugarbeet	price N
1				-2986.86	-1785.45	-3677.01	-5337.3	18
2	200		77718	-203.49		-180.84	-5.67	0.18
3	145.098	54.902	80586	-195.81		-170.04		0.09
4	145.098	54.902	83129.4	-188.688		-160.6		0.018
5	145.098	54.902	83765.3	-186.907		-158.239		

Content today

Extending the MyFarm LP model

- Ex. 5: Crop nutrient need
- Ex. 6: Fertilizer pricing
- Ex. 7: Shocks to fertilizer prices

Compile time conditionals

Introduction to a Market Model (PE)

- Ex. 8: MyMarketModel

Introduction to the CAPRI market model

- Ex. 9: run the CAPRI market model (didactic version)

Compile time conditions

- ✓ When GAMS executes the program, the code is first compiled and then executed
- ✓ During compilation it checks (and modify) the code but does not produce any results (does not run any logic) but produces a listing file
- ✓ All command with a \$ in the **first** column of a code-line are compile time statements and are interpreted during the compilation.
- ✓ Compile time statements cannot be seen in the .lst file
- ✓ Dollar control options are not part of the GAMS language. They instruct the compiler to perform some task. Therefore, dollar control options are not terminated with a semicolon as real GAMS language statements.
- ✓ See also https://gams.com/latest/docs/UG_DollarControlOptions.html#print=1#UG_DollarControl_Syntax

Compile time conditions

The screenshot illustrates the compilation process in GAMS Studio. It shows three overlapping windows of the software interface. The top window displays the initial GAMS code with line 46 highlighted. The middle window shows the same code with a red box around the '</>' icon in the toolbar. The bottom window shows the code after compilation, with a red box around the 'Run' (play) icon in the toolbar. The code in the bottom window shows numerical values assigned to variables like p_UVAG and p_lab.

```
39 p_UVAG("rapeseed") = 284;  
40 p_UVAG("sugarbeet") = 516;  
41  
42 p_lab("wheat") = 25;  
43 p_lab("barley") = 36;  
44 p_lab("rapeseed") = 27;  
45 p_lab("sugarbeet") = 87;  
46  
47  
48 Variables  
49 v_obje objective function value  
50 ;  
51
```

Compile time statements



Only Compile

```
$setglobal TRIGGER on  
    parameter p(*);  
    set s/1*10/;  
$iftheni %TRIGGER% == on  
    p(s)=uniform(1,2);  
$endif  
    display p;
```

Compilation

```
GAMS 40.1.1 23eb37fb Aug 16, 2022 DEX-DEG x86 64bit/Mac OS X -  
General Algebraic Modeling System  
Compilation  
  
2 parameter p(*);  
3 set s/1*10/;  
5 p(s)=uniform(1,2);  
7  
8 display p;  
  
COMPILATION TIME = 0.001 SECONDS 3 MB 40.1.1 23eb37fb I
```



Compile & Execute

```
$setglobal TRIGGER on  
    parameter p(*);  
    set s/1*10/;  
$iftheni %TRIGGER% == on  
    p(s)=uniform(1,2);  
$endif  
    display p;
```

Compilation

Execution
> Display

```
GAMS 40.1.1 23eb37fb Aug 16, 2022 DEX-DEG x86 64bit/Mac OS X - 09/01/22 21:11:51 I  
General Algebraic Modeling System  
Compilation  
  
2 parameter p(*);  
3 set s/1*10/;  
5 p(s)=uniform(1,2);  
7  
8 display p;  
  
COMPILATION TIME = 0.001 SECONDS 3 MB 40.1.1 23eb37fb DEX-DEG  
GAMS 40.1.1 23eb37fb Aug 16, 2022 DEX-DEG x86 64bit/Mac OS X - 09/01/22 21:11:51 I  
General Algebraic Modeling System  
Execution  
  
---- 8 PARAMETER p  
1 1.172, 2 1.843, 3 1.550, 4 1.301, 5 1.292, 6 1.224, 7 1.350,  
  
EXECUTION TIME = 0.000 SECONDS 4 MB 40.1.1 23eb37fb DEX-DEG
```

Compile time statements



Only Compile

```
$setglobal TRIGGER off  
parameter p(*);  
set s/1*10/;  
$iftheni %TRIGGER% == on  
p(s)=uniform(1,2);  
$endif  
display p;
```

```
C o m p i l a t i o n  
GAMS 40.1.1 23eb37fb Aug 16, 2022 DEX-DEG x86 64bit/Mac OS X - 09/01/22 :  
General Algebraic Modeling System  
C o m p i l a t i o n  
  
2 parameter p(*);  
3 set s/1*10/;  
5  
6 display p;  
**** $141  
**** LINE 8 INPUT /Users/gocht/Nextcloud/Applied data Analysis 2021/mod  
**** 141 Symbol declared but no values have been assigned. Check for missing  
**** data definition, assignment, data loading or implicit assignment  
**** via a solve statement.  
**** A wild shot: You may have spurious commas in the explanatory  
**** text of a declaration. Check symbol reference list.  
  
**** 1 ERROR(S) 0 WARNING(S)  
  
COMPILATION TIME = 0.002 SECONDS 3 MB 40.1.1 23eb37fb DEX-DEG
```



Compile & Execute

```
$setglobal TRIGGER off  
parameter p(*);  
set s/1*10/;  
$iftheni %TRIGGER% == on  
p(s)=uniform(1,2);  
$endif  
display p;
```

```
C o m p i l a t i o n  
GAMS 40.1.1 23eb37fb Aug 16, 2022 DEX-DEG x86 64bit/Mac OS X - 09/01/22 21:  
General Algebraic Modeling System  
C o m p i l a t i o n  
  
2 parameter p(*);  
3 set s/1*10/;  
5  
6 display p;  
**** $141  
**** LINE 8 INPUT /Users/gocht/Nextcloud/Applied data Analysis 2021/module  
**** 141 Symbol declared but no values have been assigned. Check for missing  
**** data definition, assignment, data loading or implicit assignment  
**** via a solve statement.  
**** A wild shot: You may have spurious commas in the explanatory  
**** text of a declaration. Check symbol reference list.  
  
**** 1 ERROR(S) 0 WARNING(S)  
  
COMPILATION TIME = 0.001 SECONDS 3 MB 40.1.1 23eb37fb DEX-DEG
```

CAPRI uses a lot of compile time conditions

- ✓ To copy the content of another file into the program

```
$include 'envind\def_gascoeff.gms'
```

- ✓ To define variable a simple logic which can be assessed later during compilation

```
$setglobal declAmmoSymbols NO
```

- ✓ To check of a file exist in a folder and to do simple character manipulation

```
$ifi exist '%results_in%\simini\sim_ini_%NTSLVL%%BAS%%SIM%%MODID%.gdx'
```

- ✓ They can be combined in one line

```
$ifi %BASELINE%==on $setglobal declAmmoSymbols NO
```

- ✓ To enforce to ignore parts of the code to be compiled

```
$goto after_simini  
code  
$label after_simini  
$ontext  
code  
$offtext
```

Content today

Extending the MyFarm LP model

- Ex. 5: Crop nutrient need
- Ex. 6: Fertilizer pricing
- Ex. 7: Shocks to fertilizer prices

Compile time conditionals

Introduction to a Market Model (PE)

- Ex. 8: MyMarketModel

Introduction to the CAPRI market model

- Ex. 9: run the CAPRI market model (didactic version)

Introduction to a Market Model

Ex 8: MyMarketModel



UFOP e.V., https://www.ufop.de/files/8216/7507/1560/Cover_EN_VB_2023.png, accessed on 29.08.2024

Based on the course “Introduction to Simulation Models in Market and Policy Analysis” by the group for “International Agricultural Trade and Development” at Humboldt-University Berlin

Introduction to a Market Model

MyFarm model



This Photo by Unknown Author is licensed under [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/)

- Linear Programming
- Optimal solution in underdetermined equation system

Market model



UFOP e.V.,
https://www.ufop.de/files/8216/7507/1560/Cover_EN_VB_2023.png,
accessed on 29.08.2024

- Equilibrium Model
- One unique solution in fully determined equation system

The mathematical model

$$DEMAND = par_dem \times PRICE^{elas_dem}$$

$$SUPPLY = par_supply \times PRICE^{elas_supply}$$

$$NX = SUPPLY - DEMAND$$

Parameters:

- Value assigned or calibrated at the beginning.
- Remain unchanged after that.
- Lower case letters.

Variables:

- Changed to find an equilibrium.
- Can be fixed, min/max values can be specified.
- Start values can be assigned.
- Upper case letters.

Start values for variables

- Assignment of values to variables
 - Makes solving easier and quicker. Necessary for calibration.
 - Use .L (=level) suffix

*Start values for variables

SUPPLY.L = 256;

DEMAND.L = 180;

NX.L = SUPPLY.L - DEMAND.L;

PRICE.L = 400;

Values for parameters

- Assign values to parameters

```
el_supply = 0.5;  
el_dem    = -0.1;
```

$DEMAND = par_dem \times PRICE^{elas_dem}$

$SUPPLY = par_supply \times PRICE^{elas_supply}$

- What about `par_dem` and `par_supply`?

Calibrate parameters

- Calibrate `par_dem` and `par_supply`

$$DEMAND = par_dem \times PRICE^{elas_dem}$$

$$SUPPLY = par_supply \times PRICE^{elas_supply}$$

$$par_supply = \frac{SUPPLY}{PRICE^{elas_supply}}$$

$$par_dem = \frac{DEMAND}{PRICE^{elas_dem}}$$

```
par_supply = SUPPLY.L/PRICE.L**el_supply;
```

```
par_dem = DEMAND.L/PRICE.L**el_dem;
```

Equations

$$DEMAND = par_dem \times PRICE^{elas_dem}$$

$$SUPPLY = par_supply \times PRICE^{elas_supply}$$

$$NX = SUPPLY - DEMAND$$

```
q_dem..    DEMAND =E= par_dem * PRICE**el_dem;
```

```
q_supply.. SUPPLY =E= par_supply * PRICE**el_supply;
```

```
q_nx..    NX      =E=  SUPPLY - DEMAND;
```

Matching equations & variables

- Fully determined system with one unique solution needs: no. of variables = no. of equations

$$DEMAND = par_dem \times PRICE^{elas_dem}$$

$$SUPPLY = par_supply \times PRICE^{elas_supply}$$

$$NX = SUPPLY - DEMAND$$

- 4 variables: SUPPLY, DEMAND, NX , PRICE
- 3 equations: q_dem, q_supply, q_nx
- One variable needs to be fixed using the .fx suffix
 - NX.fx=76; or PRICE.fx=400;

Solve the model

- Before:

Solve myfarm **using lp maximizing** v_obje;

LP (=Linear Programming) for underdetermined systems, optimization models

- Now:

Solve market **using MCP**;

MCP (=Mixed Complementarity Problem) for fully determined systems, equilibrium models

**SOLVE market maximizing price using nlp;*

**SOLVE market minimizing price using nlp;*

NLP (=nonlinear Programming) for underdetermined systems, optimization models with non-linear equations

Ex 8: MyMarket model

To Do:

- Go through the file
- Fill out the missing parts (**~~~~~ fill this ~~~~~*):
 - Start values for variables
 - Calibration of parameters
 - Definition of equations
 - Closures (to ensure a square model)
- Run the model
- Together: Try out the different solvers

Content today

Extending the MyFarm LP model

- Ex. 5: Crop nutrient need
- Ex. 6: Fertilizer pricing
- Ex. 7: Shocks to fertilizer prices

Compile time conditionals

Introduction to a Market Model (PE)

- Ex. 8: MyMarketModel

Introduction to the CAPRI market model

- Ex. 9: run the CAPRI market model (didactic version)

Introduction to the CAPRI market model

Ex. 9: run the CAPRI market model (didactic version)






















Based on the course “Didactic model prepared for the CAPRI training session 2019, Seville, Spain, by Mihaly Himics.”

Introduction to the CAPRI market model

- The didactic model has the same names of files, step variables, parameters etc. as the market model in CAPRI
- It should be used to read and understand GAMS program code, execute it and examine values in the parameters

Introduction to the CAPRI market model

- One product, three regions model
- Supply functions derived from Normalized Quadratic profit functions
- Generalized Leontief demand system
- Two-stage Armington (CES formulation) (bilateral Trade)
- No processing industry, no feed, no biofuels... => demand is only human consumption (What else do consume?).
- Tariffs are exogenous, only ad-valorem (see $p_{tarAdVal}$)
- Two equivalent formulations for the market model
 - NLP Non-linear maximization of a dummy objective
 - MCP (Mixed Complementarity Problem) with orthogonality constraints

- ▼  **market_bare_onec**
 - ▼  include
 - ▼  base
 -  calibrate_NQ_supply_model.gms
 -  data_prep.gms
 -  prep_market.gms
 -  calibration.gms
 -  save_results.gms
 -  test_calibration.gms
 -  money_metric.gms
 - ▼  onec
 -  market_model_one.gms
 -  calibrate_GL_demand_model.gms
 -  data_cal_one.gms
 -  calibrate_GL_demand_one.gms
 - ▼  results
 -  results_onec.gdx
 -  **market_bare_onec.gms**
 -  market_bare_onec.lst

Ex 9: Run the model

Run the didactic CAPRI market model: Main file
market_bare_onec.gms

1. Check the listing file to see if the model
m_GlobalMarket runs successfully.

```

C o m p i l a t i o n
I n c l u d e   F i l e   S u m m a r y
E x e c u t i o n
> Display
Equation Listing   SOLVE GL_demandSystem Using NLP From line 1069
> Equation
Column Listing     SOLVE GL_demandSystem Using NLP From line 1069
> Column
Range Statistics   SOLVE GL_demandSystem Using NLP From line 1069
Model Statistics   SOLVE GL_demandSystem Using NLP From line 1069
Solution Report    SOLVE GL_demandSystem Using NLP From line 1069
> SolEQU
> SolVAR
E x e c u t i o n
> Display
Range Statistics   SOLVE m_trimElas Using NLP From line 1274
Model Statistics   SOLVE m_trimElas Using NLP From line 1274
Solution Report    SOLVE m_trimElas Using NLP From line 1274
> SolEQU
> SolVAR
E x e c u t i o n
> Display
Equation Listing   SOLVE m_GlobalMarket Using MCP From line 1558
> Equation
Column Listing     SOLVE m_GlobalMarket Using MCP From line 1558
> Column
Range Statistics   SOLVE m_GlobalMarket Using MCP From line 1558
Model Statistics   SOLVE m_GlobalMarket Using MCP From line 1558
Solution Report    SOLVE m_GlobalMarket Using MCP From line 1558
E x e c u t i o n
v Display
PS_CAL
PS_Y
p_welfareRes

```

```

Model Statistics   SOLVE m_GlobalMarket Using MCP From line 1558

MODEL STATISTICS

BLOCKS OF EQUATIONS           16   SINGLE EQUATIONS           54
BLOCKS OF VARIABLES           17   SINGLE VARIABLES           63
NON ZERO ELEMENTS             183  NON LINEAR N-Z             105
CODE LENGTH                    423  CONSTANT POOL              62

GENERATION TIME = 0.063 SECONDS      4 MB 46.5.0 ae71108d WEX-WEI
GAMS 46.5.0 ae71108d May 8, 2024      WEX-WEI x86 64bit/MS Windows - 05/14/25 15:19:36 Page 18
shortcapri
Solution Report    SOLVE m_GlobalMarket Using MCP From line 1558

S O L V E   S U M M A R Y

MODEL   m_GlobalMarket
TYPE    MCP
SOLVER  PATH                      FROM LINE 1558

**** SOLVER STATUS   1 Normal Completion
**** MODEL STATUS   1 Optimal

RESOURCE USAGE, LIMIT      0.016 100000000000.000
ITERATION COUNT, LIMIT    0      2147483647
EVALUATION ERRORS         0          0
54 row/cols, 168 non-zeros, 5.76% dense.

Path 5.0.07 (Wed May 8 03:39:22 2024)
Written by Todd Munson, Steven Dirkse, Youngdae Kim, and Michael Ferris

INITIAL POINT STATISTICS
Maximum of X . . . . . 6.4232e+04 var: (v_GLDemandFS(R3))
Maximum of F . . . . . 5.8388e-09 eqn: (XiS_(R1,'X1'))
Maximum of Grad F . . . . . 4.1305e+03 eqn: (XiS_(R2,'X1'))
var: (v_GLDemandGiS(R2,'X1'))

INITIAL JACOBIAN NORM STATISTICS
Maximum Row Norm . . . . . 4.2830e+03 eqn: (XiS_(R2,'X1'))
Minimum Row Norm . . . . . 1.7669e-03 eqn: (SupBalm_(R1,'X1'))
Maximum Column Norm . . . . . 4.1315e+03 var: (v_GLDemandGiS(R2,'X1'))
Minimum Column Norm . . . . . 6.8586e-03 var: (v_expQuant(R2,'X1'))

FINAL STATISTICS
Inf-Norm of Complementarity . . 5.8388e-09 eqn: (XiS_(R1,'X1'))
Inf-Norm of Normal Map . . . . . 5.8388e-09 eqn: (XiS_(R1,'X1'))
Inf-Norm of Minimum Map . . . . . 5.8388e-09 eqn: (XiS_(R1,'X1'))
Inf-Norm of Fischer Function . . 5.8388e-09 eqn: (XiS_(R1,'X1'))
Inf-Norm of Grad Fischer Fcn . . 1.3758e-05 eqn: (GLDemandGiS_(R1,'X1'))
Two-Norm of Grad Fischer Fcn . . 1.5081e-05

FINAL POINT STATISTICS
Maximum of X . . . . . 6.4232e+04 var: (v_GLDemandFS(R3))
Maximum of F . . . . . 5.8388e-09 eqn: (XiS_(R1,'X1'))
Maximum of Grad F . . . . . 4.1305e+03 eqn: (XiS_(R2,'X1'))
var: (v_GLDemandGiS(R2,'X1'))

**** REPORT SUMMARY :      0   NONOPT
                          0   INFEASIBLE
                          0   UNBOUNDED
                          0   REDEFINED
                          0   ERRORS

GAMS 46.5.0 ae71108d May 8, 2024      WEX-WEI x86 64bit/MS Windows - 05/14/25 15:19:36 Page 19
shortcapri
E x e c u t i o n

---- 1780 PARAMETER PS_CAL price in calib. point

X1 inpe

```

R
n
1

Ex 9: Run the model

Run the didactic CAPRI market model: Main file
market_bare_onec.gms

1. Check the listing file to see if the model m_GlobalMarket runs successfully.
2. Analyze also the GDX file results_onec.gdx in the result folder and:
 1. Confirm that we have one agricultural product (see set XX). What is the second product?

Ex 9: Run the model

Run the didactic CAPRI market model: Main file `market_bare_onec.gms`

1. Check the listing file to see if the model `m_GlobalMarket` runs successfully.
2. Analyze also the GDX file `results_onec.gdx` in the result folder and:
 1. Confirm that we have one agricultural product (see set `XX`). What is the second product?
 2. Confirm that we have three regions defined in the model.

Ex 9: Run the model

Run the didactic CAPRI market model: Main file
market_bare_onec.gms

1.

2.

The screenshot shows the GAMS software interface with the 'model' window open. The table below lists the model's components:

Entry	Name	Type	Dim	Records	
1	R	Set	1	3	regions
4	basCalCur	Set	1	3	time
8	R1	Alias	1	0	Aliased with R
9	R2	Alias	1	0	Aliased with R
10	RUNR	Alias	1	0	Aliased with R
16	p_dpCESTrade	Parameter	3	12	
17	p_tradeFlows	Parameter	4	6	Physical import flows, destination first
18	p_doubleZero	Parameter	4	0	Double zero agreements
19	p_impPrice	Parameter	4	9	import price
21	p_tarAdVal	Parameter	3	6	Ad-valorem tariffs
26	p_rhoArm1	Parameter	2	3	Substitution parameter first stage
27	p_rhoArm2	Parameter	2	3	Substitution parameter second stage
28	p_results	Parameter	5	174	reporting parameter
33	ProdNQ_	Equation (=e=)	2	3	Supply function from NQ profit function

the model.

the
object
in

Ex 9: Run the model

Run the didactic CAPRI market model: Main file
market_bare_onec.gms

1. Check the listing file to see if the model m_GlobalMarket runs successfully.
2. Analyze also the GDX file results_onec.gdx in the result folder and:
 1. Confirm that we have one agricultural product (see set XX). What is the second product?
 2. Confirm that we have three regions defined in the model.
 3. How much does the R1 import from R2 and R2 from R1?

Ex 9: Run the model

R ¹	R ²	XX ³	Level	Marginal	Lower	Upper	Scale
R1	R1	X1	0	0	0	0	1
R1	R2	X1	95	0	0	+INF	1
R1	R3	X1	89	0	0	+INF	1
R2	R1	X1	72	-0	0	+INF	1
R2	R2	X1	0	0	0	0	1
R2	R3	X1	58	0	0	+INF	1
R3	R1	X1	70	-0	0	+INF	1
R3	R2	X1	64	-0	0	+INF	1
R3	R3	X1	0	0	0	0	1

1. Confirm that we have one agricultural product (see set XX). What is the second product?

2. C

3. H

```

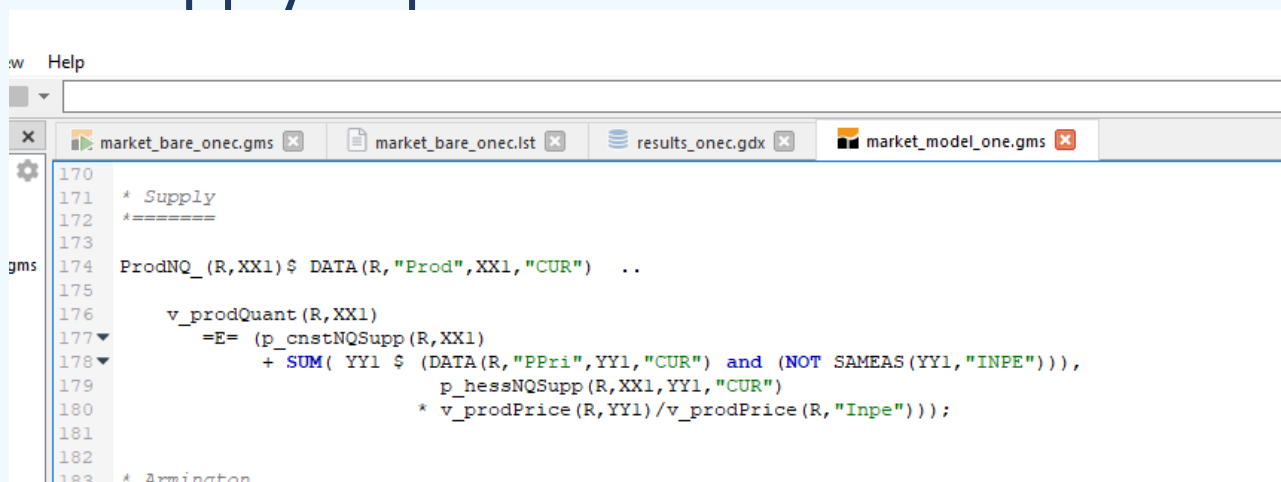
parameters
20     data(*,*,*,*)                "an equivalent of the capri data cube (stores m
21
22
23     p_dpCESTrade(R,*,XX)
24     p_tradeFlows(R,R,XX,basCalCur)  "Physical import flows, destination first"
25     p_doubleZero(R,R,XX,basCalCur)  "Double zero agreements"
26     p_impPrice(R,R,XX,basCalCur)    "import price"
27     p_trc(R,R,XX,basCalCur)        "transportation cost"
  
```

Ex 9: Normalized Quadratic profit

Exercise: check the file

“include/onec/market_model_one.gms” and answer the questions:

1. Find the supply function and check the parameters and variables in the function. Of what does supply depend on?



```
170
171 * Supply
172 *=====
173
174 ProdNQ_ (R,XX1)$ DATA(R,"Prod",XX1,"CUR") ..
175
176     v_prodQuant(R,XX1)
177     =E= (p_cnstNQSupp(R,XX1)
178         + SUM( YY1 $ (DATA(R,"PPri",YY1,"CUR") and (NOT SAMEAS(YY1,"INPE"))),
179             p_hessNQSupp(R,XX1,YY1,"CUR")
180             * v_prodPrice(R,YY1)/v_prodPrice(R,"Inpe")));
181
182
183 * Armington
```

Ex 9: Policy Simulation

Please review the file `market_bare_onec.gms` from line 99 to 114 for the following actions:

1. Activate the free-trade agreement (FTA) between R1 and R2, which is implemented as a double zero agreement, resulting in no tariff between R1 and R2.
2. What is the role of the parameter `p_doubleZero`? (look at `market_model_one.gms`)
3. Run the model again and inspect the GDX output, please report how trade has been changed?

Ex 9: Policy Simulation

3. Run the model again and inspect the GDX output, please report how trade has been changed?

no FTA between R1 and R2

market_bare_onec.gms | market_bare_onec.lst | results_onec.gdx

v_tra | Export | Table View

Entry	Name	Search ...
60	v_tradeFlows	Variab

R ¹	R ²	XX ³	Level	Margi
R1	R1	X1	0	
R1	R2	X1	95	
R1	R3	X1	89	
R2	R1	X1	72	
R2	R2	X1	0	
R2	R3	X1	58	
R3	R1	X1	70	
R3	R2	X1	64	
R3	R3	X1	0	

market_bare_onec.gms | market_bare_onec.lst | results_onec.gdx

v_imp | Export | Table View

Entry	Name	Search ...
52	v_impPrice	Variab

R ¹	R1 ²	XX ³	Level	Ma
R1	R1	X1	0	
R1	R2	X1	1064	
R1	R3	X1	867	
R2	R1	X1	1078	
R2	R2	X1	0	
R2	R3	X1	964	
R3	R1	X1	1078	
R3	R2	X1	1064	
R3	R3	X1	0	

FTA between R1 and R2

market_bare_onec.gms | market_bare_onec.lst | results_onec.gdx

v_tra | Export | Table View

Entry	Name	Search ...
60	v_tradeFlows	Variab

R ¹	R ²	XX ³	Level	Margi
R1	R1	X1	0	
R1	R2	X1	226	
R1	R3	X1	58	
R2	R1	X1	188	
R2	R2	X1	0	
R2	R3	X1	38	
R3	R1	X1	66	
R3	R2	X1	55	
R3	R3	X1	0	

market_bare_onec.gms | market_bare_onec.lst | results_onec.gdx

v_imp | Export | Table View

Entry	Name	Search ...
52	v_impPrice	Variab

R ¹	R1 ²	XX ³	Level	Margi
R1	R1	X1	0	
R1	R2	X1	862	
R1	R3	X1	845	
R2	R1	X1	863	
R2	R2	X1	0	
R2	R3	X1	939	
R3	R1	X1	1079	
R3	R2	X1	1078	
R3	R3	X1	0	

Content today

Extending the MyFarm LP model

- Ex. 5: Crop nutrient need
- Ex. 6: Fertilizer pricing
- Ex. 7: Shocks to fertilizer prices

Compile time conditionals

Introduction to a Market Model (PE)

- Ex. 8: MyMarketModel

Introduction to the CAPRI market model

- Ex. 9: run the CAPRI market model (didactic version)