



Introduction to GAMS Modeling Language

Dr. Alex Gocht

Thuenen Institute of Farm Economics, Braunschweig

Content

- Fertilizer pricing
- Compile time conditions

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		

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:

- Top Window:** Displays the source code with line 46 highlighted. The code includes:

```
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
```
- Middle Window:** Shows the same source code, but with a red box highlighting the '</>' icon in the toolbar, indicating the compilation step.
- Bottom Window:** Shows the compiled output with numerical values for variables and a red box around the 'Run' button in the toolbar. The output is:

```
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
```