GAMS-CAPRI Training

Sevilla, 9-11 April 2018

Multimarket model (small example)

Maria Blanco

Dep. Agricultural Economics

Technical University of Madrid

maria.blanco@upm.es

Multi-commodity models

Focus on several interlinked markets

- Supply side: links between commodities through resource competition (land, labour, water), cross price elasticities
- Demand side: links between commodities through income and substitution effects
- Interregional trade: links through commodity trade

Markets depicted by demand and supply functions

- They often depict not only primary products but also secondary products (processing activities)

Multi-commodity models

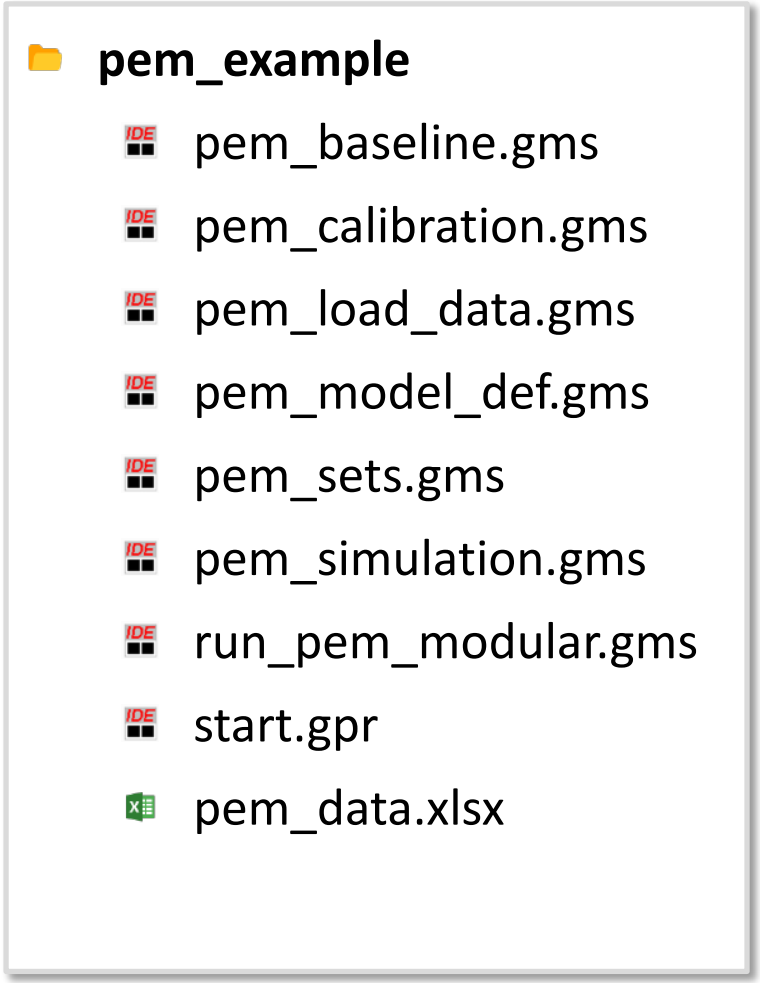
- Exercises in this section based on a small multi-commodity model
- Open the folder **pem_example** and have a look at the model structure. This is a small template of a multi-commodity model in GAMS, formulated as a NLP model
  - 3 regions and 6 commodities
  - trade between regions
  - known price elasticities
  - observed market data

Multi-commodity model: main features

- Partial equilibrium model
  - Isoelastic supply and demand functions => **system of non-linear equations**
  - Calibrated to observed data (base period = 2010)
  - Solved as optimization model (artificial objective function)

Multi-commodity model: model structure

- Modular approach
  - First, open the GAMSIDE by clicking on the project file **start.gpr**
  - Next, open the executable file of the model : **run_pem_modular.gms**



Multi-commodity model: model structure

- Open file **run_pem_modular.gms**
  - Look at how the model is written in pieces (modules) that are called from a master file (**run_**)

C:\gams_capri\models\pem_modular\run_pem_modular.gms

run_pem_modular.gms

```
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
*~~~~~~~~~~ PARTIAL EQUILIBRIUM MODEL: MODULAR FORMULATION  ~~~~~~~~~~~~~~~~~~~~*

$include "pem_sets.gms"

$include "pem_load_data.gms"

$include "pem_model_def.gms"

$include "pem_calibration.gms"

$include "pem_baseline.gms"

$include "pem_simulation.gms"
```

Multi-commodity model: SETS

- Open the first module: **pem_sets.gms**
  - Look at how the set elements are defined using subsets and a master set.
  - Look at how some sets have been duplicated using ALIAS

C:\gams_capri\models\pem_modular\pem_sets.gms

pem_sets.gms

```
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ INDICES  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
* Dimensions: regions (rall), products (pall), measures (m) and years (yall)

sets
*   ---- regional units

REG  'countries/regions'    /AME,EUR,ROW/

RALL 'all regions and aggregates'    /set.REG,WLD/

R(rall) 'regions'  /set.REG/

*   ---- commodities
VEG 'vegetal products'   /
   WHEA  'wheat'
   BARL  'barley'
   FVEG  'fruits and vegetables'
   /
```

GAMS features (ALIAS)

ALIAS statement

- Gives another name to a set defined previously
- Useful in market equilibrium problems to specify cross elasticities

Syntax

```
alias(knownSet,newSet);

* example
set c commodities ;
alias(c,cc);
```

Multi-commodity model: DATA

- Open the module: **load_pem_data.gms**
  - Data parameters ("master" set as declaration domain)
  - Ex-post data is imported from an Excel file (through GDX) and stored in a datacube

C:\gams_capri\models\pem_modular\pem_load_data.gms

pem_load_data.gms

```
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ PARAMETERS ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
parameter
*    ---- 'datacube (ex-post data)'
p_data(rall,pall,m,yall) 'parameter to store observed data'

*    ---- observed market balances
X0(rall,pall,*)     'observed initial values'

*    ---- parameters of supply and demand functions
selas(rall,pall,pall) 'price elasticity of supply'
delas(rall,pall,pall) 'price elasticity of demand'

*    ---- technical and policy parameters
p_mar(rall,pall,yall) marketing margin (from producer price to consumer price)
p_sub(rall,pall,yall) producer net subsidy (euro per ton)
p_tav(rall,pall,yall) ad valorem import tariff
;
```

Multi-commodity model: MODEL DEFINITION

- Open the module: **pem_model_def.gms**
  - Model definition: variables and equations
  - Naming convention for variables (**v_**varname) and equations (eqname**_**)

C:\gams_capri\models\pem_modular\pem_model_def.gms

pem_model_def.gms

```
*~~~~~~~~~~~~~~~~~~~~~~~~~ MODEL NLP  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
parameter
logy   'needed to formulate trends in log form'
;
logy(yall) = log(ord(yall)) ;

variables
   DUMMY                    'artificial objective variable'
   v_netT(rall,pall,yall)   'net trade (Mio t)'
;
positive variables
   v_prodQ(rall,pall,yall) 'domestic production (Mio t)'
   v_consQ(rall,pall,yall) 'domestic consumption (Mio t)'

   v_prodP(rall,pall,yall) 'producer price (euro/t)'
   v_consP(rall,pall,yall) 'consumer price (euro/t)'
   v_worldP(pall,yall)     'world price (euro/t)'
```

Multi-commodity model: MODEL DEFINITION

- Model definition: variables and equation.
  - System of non-linear equations solved as optimization under constraints (artificial objective function)
  - Isoelastic supply and demand functions (**double-log formulation**)
  - Base period = 2010
  - Time horizon = 2010-2020
  - Market equilibrium: total net exports equal zero

Isoelastic demand function

$$QD_{r,p} = ad_{r,p} * \prod_{p1} PD_{r,p1}^{\varepsilon_{r,p,p1}} * (1 + chd_{r,p})$$

In GAMS language

product summation

```
DEMAND(r,p)..   QD(r,p) =E= ad(r,p)
                * prod(p1,PD(r,p1)**delas(r,p,p1))
                * (1+chd(r,p));
```

Isoelastic demand function (double-log)

$$log(QD_{r,p}) = ad_{r,p} + \sum_{p1} \varepsilon_{r,p,p1} * log(P_{r,p1}) + log(1 + chd_{r,p})$$

In GAMS language

double-log format

```
DEMAND(r,p)..   log(QD(r,p)) =E= ad(r,p)
                + sum(p1, delas(r,p,p1)*log(PD(r,p1)))
                + log(1+chd(r,p));
```

REMAINDER: GAMS functions

Common mathematical functions

sum	sum of set indexed expressions
prod	product of set indexed expressions
sqr	square of an expression or term
sqrt	square root of an expression or term
log	natural logarithm
abs	absolute value
max, min	maximum or minimum of a set of expressions or terms
smax, smin	maximum or minimum of set indexed expressions or terms

REMAINDER: Working with non-linear models

Assigning initial values to variables

- They help GAMS to find the optimal solution and speed up the iteration process (default value is zero)
- They need to be entered before the solve statement

Syntax

```
var_name.L = ini_value ;
solve nlpModel ...
```

REMAINDER: Working with non-linear models

Providing lower and upper bounds

- They speed up the iteration process
- They are useful when working with variables that are undefined if another variable becomes zero
- They need to be entered before the solve statement

Syntax

```
var_name.LO = lower_value ;
var_name.UP = upper_value ;
solve nlpModel ...
```

Multi-commodity model: CALIBRATION

- Open the module: **pem_calibration.gms**
  - Calibration of the model to the observed market data in the base period (2010)
  - Estimation of the intercepts of the demand and supply functions
  - Assumptions on trends for supply (technical change) and demand (demand change).

Multi-commodity model: BASELINE

- Open the module: **pem_baseline.gms**
  - Generation of the baseline or reference scenario run
  - Creation of reporting parameters
  - Storage of the baseline results in a GDX file

Multi-commodity model: SIMULATION

- Open the module: **pem_simulation.gms**
  - Definition and running of a new scenario
  - Creation of reporting parameters
  - Storage of the scenario results in a GDX file
- Run the model (executable file **run_pem_modular.gms**) and check model results

GAMS-CAPRI Training

Sevilla, 9-11 April 2018

Multimarket model (Hands-on exercises)

Maria Blanco

Dep. Agricultural Economics

Technical University of Madrid

maria.blanco@upm.es

Overview

- Mappings
- Folder structure
- File with Global Settings
- Generic scenario
- Model and solver status
- Abort
- Model and display options
- Link to Excel files

To do the exercises, a copy of the model is available in folder **pem_handson**

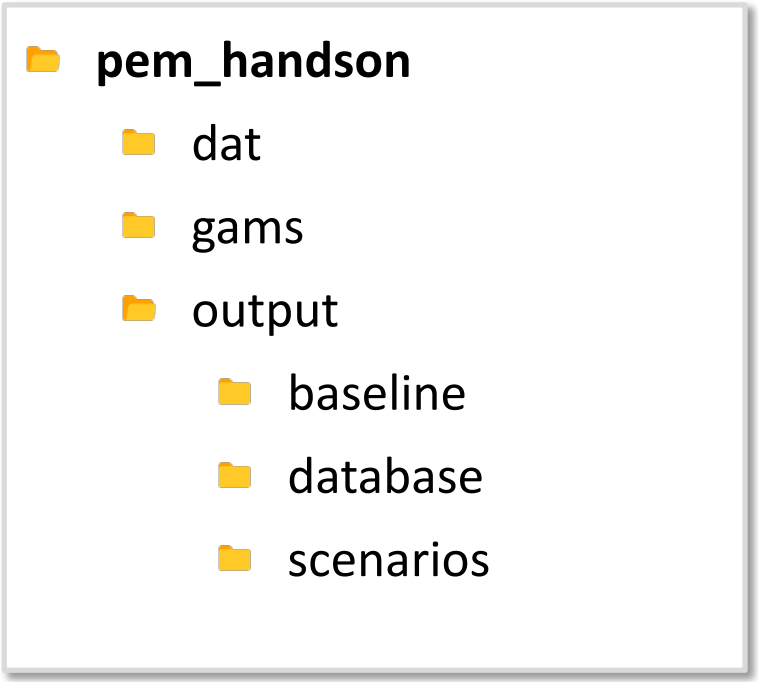
All the exercises are done in this folder pem_handson

Exercise 1: Using mappings

1. Practice how to use subsets and mappings
   a) Open the file ***pem_sets.gms***
   b) Create a set of groups of products (i.e., two elements: vegetal products and animal products)
   c) Create a 2-dimensional set to map the products to the groups
2. Once created, how can we modify the mapping (add or delete elements)?

Exercise 2: Folder structure

1. Change the folder structure, creating subfolders and including contain the different pieces of code



2. Check that the model is still running and generating results: what changes in the code are needed?

Exercise 3: Global settings

1. Define global settings
   a) Create a file **global_settings.gms** in folder **pem_handson\gams**
   b) Include in this file global variables to indicate the paths to: the gams directory, the data directory and the results directory.
2. Call this file from the executable **run_pem_modular.gms** and change the code accordingly (use relative paths)

Exercise 4: Generic scenario

1. Create a batinclude file (**scen_subsidy.gms**) to define a generic simulation scenario.

   a) The generic scenario allows to simulate a new subsidy for a particular region and product. The subsidy will be set at 50 eur/t for each region and product separately.

2. Check whether the parameter “p_sub" has changed in the desired way.

Exercise 5: Check model and solver status

1. Use model attributes to check
   a) Model status
   b) Solver status
2. Create a parameter p_check to save these model attributes in the results file.

Exercise 6: Abort

1. Add a condition to stop the model in case the base data is not available.
   a) Condition to abort the model in case pem_data.gdx does not exist. Add a message to show the error.
   b) Condition to abort the simulation of scenarios in case the baseline results are missing.

Exercise 7: Model and display options

1. Try several options controlling the content of the LST file
   a) $ options
   b) Option statements
2. Try several options controlling the solver
   a) Model options
   b) Model attributes

Model options (example)

example

```
$offlisting
option limrow=18,limcol=0,solprint=off;

option nlp = conopt;
```

GAMS features (OPTION)

Option display

- To modify the display formatting

Syntax

```
option parName:decimals:rowItems:colItems;

* example
option result:1:1:1;
```

GAMS features (OPTION)

Option decimals

- Specifies the default number of decimal places to be printed by all subsequent display statements
- The default value is 3 and the range is from 0 to 8

Syntax

```
option decimals=number;

* example
option decimals=1;
```

Exercise 8: Link to Excel files

1. Use GDX2XLS to dump all the content of the file **pem_sets.gms** to an Excel file
   a) Verify that every set gets its own sheet in the Excel file
   b) Verify that you can view the sets through the Table of Contents