

Getting Started with GAMS/MCP

James Markusen
Thomas F. Rutherford

October 18, 2004

Introduction to Computable General Equilibrium
Modeling with GAMS and MPSGE

University of Colorado, Boulder

Overview

- Installation of GAMS
- Editors for GAMS code
- Some basic GAMS syntax
- Corner solutions and the complementarity format

Installing GAMS

- Execute `D:/GAMS MPSGE/systems/win/setup.exe` on the GAMS 2.50 CD. This installs:
 - The GAMS system (with a 2-month evaluation license)
 - On-line documentation
 - A model library

Installing GAMS (cont.)

- File handling: model code
 - Project directories
 - **Do not work in the GAMS system directories**
 - Develop a backup system
- No license file? GAMS then operates in *demo mode*.

Text Editors for GAMS Programming

Model Development \Leftrightarrow Text Hacking

What is the best way to hack text?

Answer: *EMACS (or some other professional text editor)*

Related issue: *Hysterisis of programming expertise...*

Text Editors for GAMS Programming

- GAMS Integrated Development Environment (GAMS-IDE)
 - User-friendly
 - Helpful tips on “what comes next”
 - Places certain limits on long-term productivity

GAMS' Principles

- Model development and model solution are logically separate activities.
- Your GAMS program should provide a means of documenting your work.
- Focus first on the economics of your model, and think about the interface issues only after the model is running.
- The GAMS model library provides an excellent source of ideas for how to model various economic phenomena.

- Use the on-line documentation:

gams system directory/docs/bigdocs/GAMSUsersGuide.pdf

Model development in GAMS

1. Study issues and available data.
2. Program a simple pilot model
3. Repeat:
 - (i) Debug.
 - (ii) Create *ex-ante tables and graphs*.
 - (iii) Solve scenarios and create reports.

(iv) Look at the results and assess.

(v) Archive.

(vi) Elaborate or modify the model.

The Structure of a Prototypical GAMS Program

- **Inputs**

- Sets
- Data (Parameter and Table statements)
- Variables - Equations - Model statement
- Scenario definitions and Solve statements
- Display and other reporting statements

The Structure of a Prototypical GAMS Program (cont.)

- **Outputs**

- Echo prints of benchmark data
- Reference maps of where symbols are used in the program
- Equation listings
- Solver status reports
- Results, including display statements, text and Excel report files

GAMS Program Syntax: Key Ideas

- The input format is free form:
 - GAMS ignores blanks and case
 - Tabs are ignored *except in TABLES* where tab stops are assumed (by default) to be set every 8 characters.
 - Semicolons separate GAMS statements

GAMS Program Syntax: Key Ideas (cont)

- Good GAMS programmers insert the optional descriptive text wherever it is permitted:
 - *Explanatory text* for sets, set elements, parameters, variables, equations, models.
 - Comment lines, indicated by “*” in the first column, can be inserted to describe the logic underlying assignment statements.
 - Longer commentary can be introduced between `$ontext` and `$oftext` delimiters.

Two Types of GAMS Statements

Declarative statements: those which define sets, data and the logical structure of models (like Excel).

Procedural statements: those which instruct the computer to undertake a specific set of tasks in a particular sequence (like Visual Basic).

A Simple Example

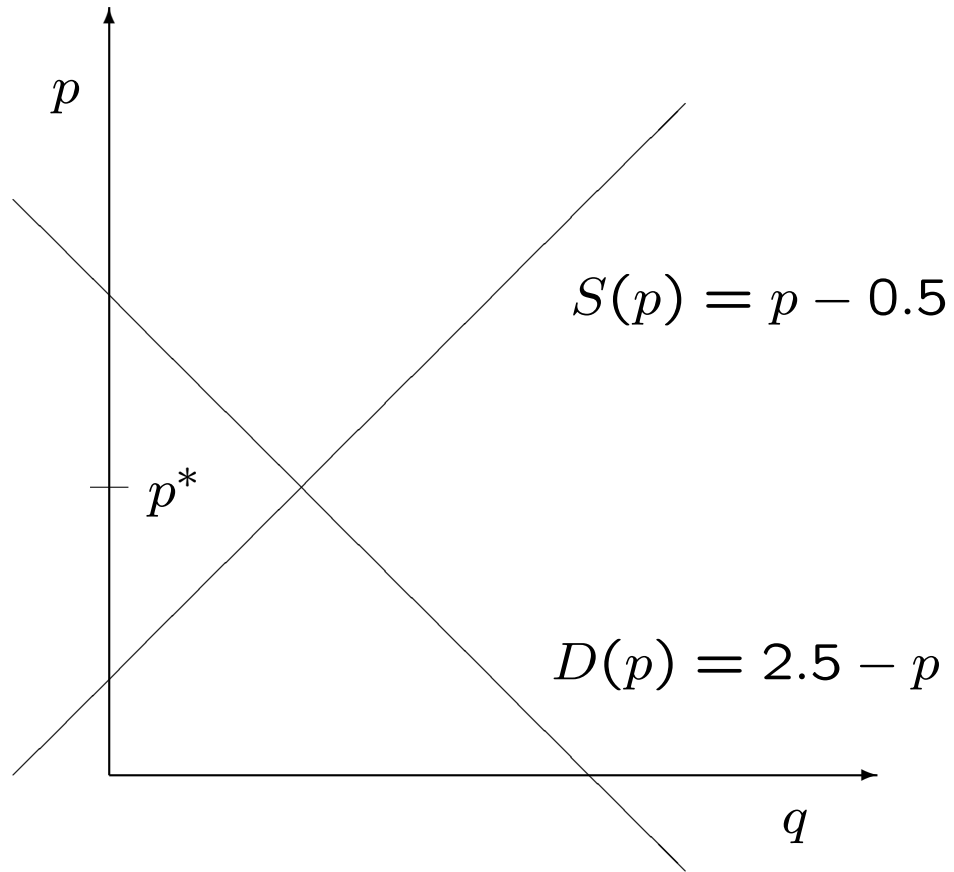
Use complementarity to solve a competitive market equilibrium model with linear supply and linear demand functions. Assume:

$$D(p) = a - bp$$

$$S(p) = c + dp$$

where a , b , c and d are given parameters.

An equilibrium price p^* solves: $S(p) = D(p)$?



GAMS Syntax for this problem:

```
$TITLE Single Commodity Market Equilibrium
```

```
VARIABLE          p          Equilibrium price;
```

```
EQUATION          mkt          Market clearance;
```

```
*          s(p)          =          d(p)
```

```
mkt..    p - 0.5 =e= 2.5 - p;
```

```
MODEL mkteql /mkt.p/;
```

```
SOLVE mkteql USING MCP;
```

GAMS Listing File (mkteq1.lst):

1. Source Listing

GAMS Rev 139 Intel /MS Window

10/18/04 02:01:38 Page 1

General Algebraic Modeling System

Compilation

```
1 VARIABLE          p          Equilibrium price;
2
3 EQUATION          mkt          Market clearance;
4
5 *          s(p)          =          d(p)
6
7 mkt..          p - 0.5 =e= 2.5 - p;
8
9
10 MODEL mkteq1 /mkt.p/;
11
12 SOLVE mkteq1 USING MCP;
```

2. Equation Listing

G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m

Equation Listing SOLVE mkteql Using MCP From line 12

---- mkt =E= Market clearance

mkt.. 2*p =E= 3 ; (LHS = 0, INFES = 3 ***)

3. Column Listing

General Algebraic Modeling System

Column Listing SOLVE mkteql Using MCP From line 12

---- p Equilibrium price

p
 (.LO, .L, .UP = -INF, 0, +INF)
 2 mkt

4. Model Statistics

GAMS Rev 139 Intel /MS Window

General Algebraic Modeling System

Model Statistics SOLVE mkteql Using MCP From line 12

MODEL STATISTICS

BLOCKS OF EQUATIONS	1	SINGLE EQUATIONS	1
BLOCKS OF VARIABLES	1	SINGLE VARIABLES	1
NON ZERO ELEMENTS	1	NON LINEAR N-Z	0
DERIVATIVE POOL	6	CONSTANT POOL	14
CODE LENGTH	1		

GENERATION TIME = 0.020 SECONDS 2.9 Mb WIN214-139 Sep 01, 2004

EXECUTION TIME = 0.020 SECONDS 2.9 Mb WIN214-139 Sep 01, 2004

5. Solution Report

GAMS Rev 139 Intel /MS Window 10/18/04 02:01:38 Page 5

General Algebraic Modeling System

Solution Report SOLVE mkteql Using MCP From line 12

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

MODEL mkteql
TYPE MCP
SOLVER PATH FROM LINE 12

**** SOLVER STATUS 1 NORMAL COMPLETION

**** MODEL STATUS 1 OPTIMAL

RESOURCE USAGE, LIMIT	0.020	1000.000
ITERATION COUNT, LIMIT	0	10000
EVALUATION ERRORS	0	0

Solution Report (cont.)

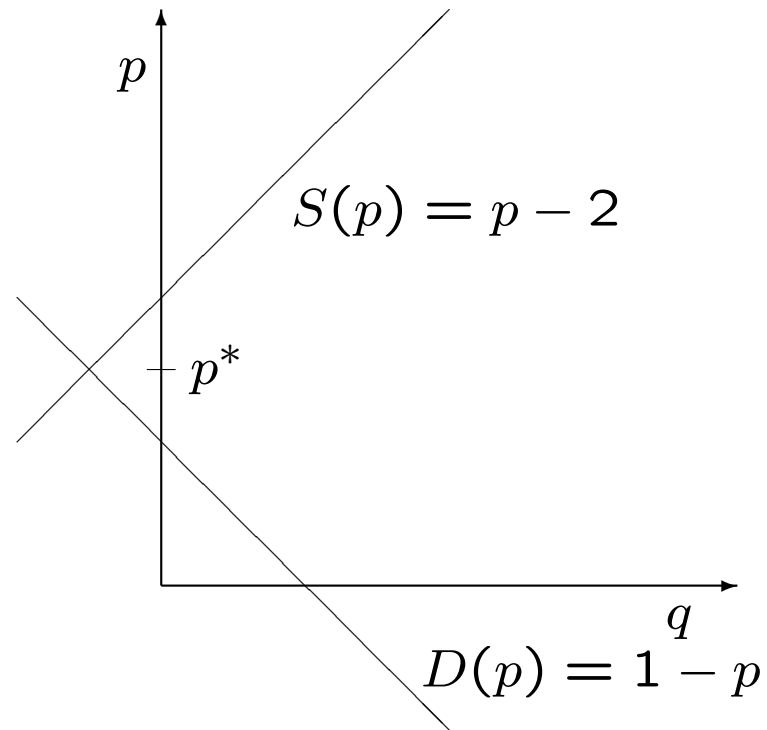
	LOWER	LEVEL	UPPER	MARGINAL
---- EQU mkt	3.0000	3.0000	3.0000	1.5000
mkt Market clearance				

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR p	-INF	1.5000	+INF	.
p Equilibrium price				

**** REPORT SUMMARY :

0	NONOPT
0	INFEASIBLE
0	UNBOUNDED
0	REDEFINED
0	ERRORS

The *complementarity formulation* can account for situations in which the equilibrium price or quantity is zero:



To represent this model in GAMS define two variables in addition to p :

x Equilibrium quantity demanded ($= D(p^*)$ if $D(p^*) > 0$)

y Equilibrium quantity supplied ($= S(p^*)$ if $S(p^*) > 0$)

We then have the *linear complementarity problem*:

$$y \geq x \quad p \geq 0 \quad p(y - x) = 0$$

$$y \geq S(p) \quad y \geq 0 \quad y(y - S(p)) = 0$$

$$x \geq D(p) \quad x \geq 0 \quad x(x - D(p)) = 0$$

These equations can also be written:

$$y \geq x \perp p \geq 0$$

$$y \geq S(p) \perp y \geq 0$$

$$x \geq D(p) \perp x \geq 0$$

GAMS Syntax for this problem:

```
$title Market Equilibrium with Corners

POSITIVE
VARIABLES      x      Equilibrium demand,
                y      Equilibrium supply,
                p      Equilibrium price;

EQUATIONS      mkt      Market clearance,
                supply  Defines y,
                demand  Defines x;

supply..       y =g= p - 2;

demand..       x =g= 1 - p;

mkt..         y =e= x;

MODEL mkteql /supply.y, demand.x, mkt.p/;

SOLVE mkteql USING MCP;
```

5. Solution Report

GAMS Rev 139 Intel /MS Window

10/18/04 02:21:15 Page 6

General Algebraic Modeling System

Solution Report SOLVE mkteql Using MCP From line 18

supply Defines y
demand Defines x
mkt Market clearance

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR x	.	.	+INF	.
---- VAR y	.	.	+INF	1.0000
---- VAR p	.	1.0000	+INF	.

x Equilibrium demand
y Equilibrium supply
p Equilibrium price

A Multimarket Example – Part I (Declarative)

Sets i canning plants / seattle, san-diego /
 j markets / new-york, chicago, topeka / ;

Parameters

a(i) Reference supplies /seattle 350, san-diego 600/,
b(j) Reference demand /new-york 325, chicago 300, topeka 275/,
esub(j) Demand elasticities / new-york 1.5, chicago 1.2, topeka 2.0 /,
f Freight in dollars per case per thousand miles /90/;

Table d(i,j) distance in thousands of miles

	new-york	chicago	topeka
seattle	2.5	1.7	1.8
san-diego	2.5	1.8	1.4;

Parameter

c(i,j) Transport cost in thousands of dollars per case,
pbar(j) Reference price at demand node j,
report(*,*,*) Summary report;

A Multimarket Example – Part II (Declarative)

Positive variables

w(i) shadow price at supply node i,
p(j) shadow price at demand node j,
x(i,j) shipment quantities in cases;

Equations

supply(i) supply limit at plant i,
fxdemand(j) fixed demand at market j,
prdemand(j) price-responsive demand at market j,
profit(i,j) zero profit conditions;

profit(i,j).. w(i) + c(i,j) =g= p(j);
supply(i).. a(i) =g= sum(j, x(i,j));
fxdemand(j).. sum(i, x(i,j)) =g= b(j);
prdemand(j).. sum(i, x(i,j)) =g= b(j) * (pbar(j)/p(j))**esub(j);

Model fixedqty LP model in MCP format / profit.x, supply.w, fxdemand.p/ ;
Model equilqty NLP model in MCP format / profit.x, supply.w, prdemand.p/;

A Multimarket Example – Part III (Procedural)

```
c(i,j) = f * d(i,j) / 1000 ;
```

```
*      Assing initial level values::
```

```
p.l(j) = 1;
```

```
w.l(i) = 1;
```

```
*      Solve the fixed demand model:
```

```
Solve fixedqty using mcp;
```

```
report("fixed",i,j) = x.l(i,j);
```

```
report("fixed","price",j) = p.l(j);
```

```
report("fixed",i,"price") = w.l(i);
```

```
*      Calibrate the demand functions to the
```

```
*      shadow prices in this equilibrium:
```

```
pbar(j) = p.l(j);
```

```
*      Replicate the fixed demand equilibrium:
```

```
Solve equilqty using mcp;
```

```
report("equil",i,j) = x.l(i,j);  
report("equil","price",j) = p.l(j);  
report("equil",i,"price") = w.l(i);
```

```
*      Compute a counter-factual equilibrium in  
*      which the cost of shipments from Seattle  
*      to Chicago are reduced by 50\%:
```

```
c("seattle","chicago") = 0.5 * c("seattle","chicago");
```

```
Solve fixedqty using mcp;  
report("fixed",i,j) = x.l(i,j);  
report("fixed","price",j) = p.l(j);  
report("fixed",i,"price") = w.l(i);
```

```
*      Compute the same scenario in the fixed demand
*      model:
```

```
Solve equilqty using mcp;
report("equil",i,j) = x.l(i,j);
report("equil","price",j) = p.l(j);
report("equil",i,"price") = w.l(i);
```

```
*      Display the report:
```

```
Display report;
```

Two Types of GAMS Errors

1. *Compilation errors* occur when GAMS is reading your program. An example: a misspelled keyword.
2. *Execution errors* occur when GAMS after your program has been loaded. An example: divide by zero, model fails to solve, your program attempts to evaluate the logarithm of a negative number, etc.

Dealing with Errors in GAMS Programs

- Standard mode of operation for any computer code in the development process is *dysfunction*.
- Two types of errors with GAMS programs: *compilation errors* and *execution errors*.
- Errors are identified by “***” in the listing file.
- Compilation errors often cascade – one error causes others.
- Typical causes of GAMS compilation errors are:
 - Missing semicolons
 - Spelling errors, particularly for keywords.
 - Misaligned numbers in tables.

Dealing with Errors in GAMS Programs (cont.)

- Execution errors are most challenging: “short-circuits between the headphones” .
- Use debugging output.
- Look at the error code (\$) and its explanation.