

Getting Started with GAMS

Short Course on CGE Modeling, University of the South Pacific

John Gilbert

Professor

Department of Economics and Finance
Jon M. Huntsman School of Business
Utah State University
jgilbert@usu.edu

July 23-27, 2012



- Computable general equilibrium models can be built using numerous different programs.
- The most widely used is GAMS, which is specifically designed for solving large-scale numerical programming problems.
- In this session we will introduce the main features of programming and running models in GAMS.

Session Outline

- 1 What is GAMS?
- 2 Getting and Installing GAMS
- 3 GAMS Primer
- 4 Example and the List File

What is GAMS?

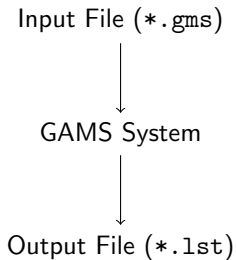
- GAMS is an acronym that stands for the General Algebraic Modeling System.
- It is a high level programming language designed for building and solving mathematical models numerically. GAMS provides a framework for model development that is independent of the platform on which the model is to be run, and distinct from the mathematical algorithms that are used to solve the model.
- GAMS can solve a wide variety of problems, and is capable of handling very large mathematical systems.
- It is in very widespread use in both the academic and business worlds, and is the most widely used development platform for computable general equilibrium models.

Getting and Installing GAMS

- GAMS Corporation provides a student/demonstration version free of charge, which is suitable for small models.
- The latest version can be downloaded for various platforms from <http://www.gams.com/download/>.
- Once the file has downloaded, double click on it to start the installation process. A prompt will appear asking if you wish to copy a license file. You can click no (without the license file GAMS will run in student/demonstration mode).
- Once it has installed, GAMS will ask if you want to launch the IDE, or integrated development environment. This is the main GAMS interface. Click yes and GAMS should appear.

A Quick GAMS Primer

- A GAMS program is a text file that describes the model structure in terms of its component variables, parameters and relationships. The text file is usually given the suffix `gms`.
- To run the program, the text file is submitted to the GAMS system. GAMS then checks for syntax errors, and then translates the model into a form usable by the solution algorithm. This process is invisible to the user.
- The solution algorithm attempts to solve the model, and then reports back to GAMS the result.
- A list file (with the suffix `lst`) is produced that contains information on the solution. If something went wrong in the process, the list file will contain information on where the problem lies.



Example

- To illustrate the basics, we'll use GAMS to solve a simple algebraic problem.
- Suppose that we want to know the simultaneous solution to $y = 10 + 2x$ and $y = 100 - x$.


```
VARIABLES  
Y, X;  
EQUATIONS  
EQ1, EQ2;  
EQ1..Y=E=10+2*X;  
EQ2..Y=E=100-X;  
MODEL EXAMPLE /ALL/;  
SOLVE EXAMPLE USING LP MAXIMIZING Y;
```

- **Echo Prints:** In this section of the list file GAMS will echo (repeat) the input to the list. Each input line is given a number by GAMS. GAMS will reference these line numbers if it finds an error in the model.
- **Equation Listing:** Here GAMS shows a specific instance of the model when the current values for the variables are substituted into the general algebraic form of the model.
- **Model Statistics:** Reports some statistics on the size of the model.
- **Status Reports:** GAMS will report back on the status of the solve (whether or not a solution has been found).
- **Solution Reports:** If a solution has been determined by GAMS, it can be found here.

Echo Prints

```
1 VARIABLES
2 Y, X;
3 EQUATIONS
4 EQ1, EQ2;
5 EQ1..Y=E=10+2*X;
6 EQ2..Y=E=100-X;
7 MODEL EXAMPLE /ALL/;
8 SOLVE EXAMPLE USING LP MAXIMIZING Y;
```

Equation Listing

```
EQ1..  Y - 2*X =E= 10 ; (LHS = 0, INFES = 10 ****)  
EQ2..  Y + X =E= 100 ; (LHS = 0, INFES = 100 ****)
```

MODEL STATISTICS

BLOCKS OF EQUATIONS 2

BLOCKS OF VARIABLES 2

NON ZERO ELEMENTS 4

SINGLE EQUATIONS 2

SINGLE VARIABLES 2

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

MODEL	EXAMPLE	OBJECTIVE Y
TYPE	LP	DIRECTION MAXIMIZE
SOLVER	CPLEX	FROM LINE 13

**** SOLVER STATUS	1 Normal Completion
**** MODEL STATUS	1 Optimal
**** OBJECTIVE VALUE	70.0000

Solution Reports

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR Y	-INF	70.000	+INF	.
---- VAR X	-INF	30.000	+INF	.

- This example is from Gilbert and Tower (2012), chapter 2.
- Further details on GAMS can be found in the GAMS User's Guide. The GAMS User's Guide contains a tutorial (chapter 2) that is a useful starting point for basic programming in GAMS.
- Zenios (1996) is another useful reference on the capabilities of GAMS.
- Bruce McCarl also has an online reference to GAMS that has very useful programming advice.