# WELCOME

## Session 5

Get used to the GAMS software

# INTRODUCTION TO GAMS

**SS**

# What is GAMS?

GAMS is a software package for designing and solving various types of models. GAMS stands for General Algebraic Modeling System. It was developed by economists at the World Bank for economic models, but it can be used to solve systems of equations from any field of study. GAMS takes a program file as input and sends the results to an output file.

There are two parts to GAMS

- First, the core of GAMS is just a language for defining the variables and equations in a model. This "language" has ways of describing any equation, even very complicated equations, in an ordinary text file (ASCII format).

- Second, there is a set of solvers, each of which is a complex program for finding the solution to a specific type of problem. Some solvers only do linear programming, some will do non-linear programming, and others will solve different types of equilibrium models

# Starting GAMS

To create and run a model, we need to learn how to use the language to describe our model, and we need to

know which solver will work with our model. In this manual, we will be using equilibrium models with both equalities and inequalities.

To open GAMS, double-click on the GAMS IDE icon. The opening screen is shown in

Box 1. The menu bar is relatively simple with just six options:

| **Menu headings** | **Function** |
|---|---|
| File | For opening and saving files, running programs, printing, and setting options. |
| Edit | For cutting and pasting text and for undoing a mistake. |
| Search | For searching and replacing text within GAMS files. |
| Windows | For arranging the windows in different ways on the screen. |
| Utilities | For making macros, importing and exporting spreadsheets, and other tasks. |
| Help | For getting assistance in how to use GAMS |

# Box 1. Screen after opening GAMS

When you first start on a new model, you will need to create a *project, which tells GAMS where to find the* files you will be working with. To create a project, click File/Project/New Project, select a folder, and type in the name of the project.

When you create and name a project, GAMS puts a file with that name and extension prj in the folder you have chosen. This makes it easier to go back to this folder and find your files in the future.

# Writing or editing a GAMS program

You can start writing a new GAMS program from scratch by clicking on File/New. A blank window will open and you can begin writing commands to compose a program. But you will probably never need to start writing a GAMS program from scratch. It is almost always easier to start with an existing program and modify it to meet your needs.

To open an existing GAMS program, click on File/Open and select the program file.

For example, if you select “SD1 simple.gms”, you will see a window appear with the GAMS program. Alternatively, from the Windows Explorer you can double-click on the file “SD1 simple.gms.” Because Windows knows that the .gms extension is for a GAMS program, it will open GAMS and then open the program. In either case, you will see the program on the left side of the screen.

This program, “SD1 simple.gms,” is the simplest possible economic model, with the supply and demand for one commodity in one region.

In spite of its simplicity, it has all the basic components of a GAMS program:

- First, the VARIABLE command gives names and labels to the two variables, quantity and price;

- Next, the EQUATIONS command names and labels the two equations (demand and supply) and then writes out the two equations (note that =E= is the equals sign);

- The MODEL command gives the model a name, “Market,” and lists the two equations that are in it;

- Finally, the SOLVE command tells GAMS which solver to use.

The model is so simple that we can solve it without GAMS. The two equations are:

$Q = 1800 - 2*P$

$Q = 1350 + P$

This implies that:

$1800 - 2*P = Q = 1350 + P$

and, by adding 2*P to both sides and subtracting 1350 from both sides, we get:

$450 = 3*P$

so $P = 150$.

Substituting this into the second equation above, we get $Q = 1500$.

In this case, we do not need GAMS to solve the problem. But economic models generally have dozen, hundreds, or even thousands of variables and equations. These problems clearly cannot be solved with a paper and pencil, but GAMS solves them in a fraction of a second.

It is important to note that all GAMS programs have the four components

1) A list of (endogenous) variables
2) A list of equations
3) A command to define the model
4) A command to tell GAMS how to solve it

# Running a program and reading the results

MS program, we are asking GAMS to read the program and solve the problem.

To run a GAMS program, we click on the button with the red arrow. Almost instantly, GAMS will

- Solve the model
- Create two files with output
- Open windows to view each of these files, as shown in Box 2.

# Box 2. Screen after running a GAMS program

The two windows are called the process window and the output window.

- The "process window" appears on the right side of the screen and shows you the .log file. This window provides information about the process of solving the model. Most importantly, it tells you if the program ran without errors or not.
- Also, the "output window" appears on the right side of the screen. It provides various results from running the program, including the solution. The output window is actually a tabbed window linked to the original program, so you can easily switch back and forth between the program and the output.

If the program ran successfully, the process window will contain a statement in blue print saying “Status: Completion normal”. And the output window will show a lot of information, including the following:

- The GAMS program with numbered lines
- An “Equation Listing”
- A “Column Listing” with information about the variables
- Some “Model Statistics” describing the size of the model, and
- The “Solve Summary” with the solution to the model

The most important section is in the Solve Summary where the variables are listed with four columns. The column labeled “Level” gives the solution values for each variable.

If, on the other hand, the program has a problem, the process window will contain a statement in blue print saying “Status: Compilation error(s)” or some other type of error.

In the output window, the place where GAMS found the errors in the program will be marked with “****” and an error code under the word in the command that may have caused the problem.

In the output window, you will find the entire program repeated, This is actually a new “tab” within the same window that showed the GAMS program. You can switch back and forth between the program and the output by clicking on the tabs at the top of the window.

# Syntax for key GAMS commands

Now, we turn to a few of the most important GAMS commands: PARAMETERS, VARIABLES, EQUATIONS, MODEL, AND SOLVE. In the description of the format, the words in brackets (< >) describe what type of words should go in that place. The format shows groups of three but there may be more or less than three. There is also an example from the SDP4 in Annex 1.

# PARAMETERS

This command defines the parameters used in the model. First, you list the names of the parameters with a short description. Then, you give the equations that define each parameter. The format is as follows:

PARAMETERS

<parameter name> <parameter description>
<parameter name> <parameter description>
<parameter name> <parameter description> ;
<parameter name> = <expression defining parameter> ;
<parameter name> = <expression defining parameter> ;
<parameter name> = <expression defining parameter> ;

An example from the SDP4.gms program (see Annex 1) is given below, along with the line numbers of each command (note that the line numbers were added to the text; they are not part of a GAMS program).

```
164     PARAMETERS
246     NER             Nominal exchange rate (1000 LC per US$)
247     POP(R )         Population in 1995 (inhabitants)
268     NER              = 11 ;
269     POP(R)           = POP94(R)*1.0225 ;
```

# VARIABLES

This command defines the variables used in the model. You list the variables with a short description of each.

VARIABLES

<variable name> <variable description>
<variable name> <variable description>
<variable name> <variable description> ;

An example from SDP4 is given below.

```
164     VARIABLES
165     P(C,R)          Equilibrium price (LC per kg)
166     D(C,R)          Quantity demanded (thousand tons)
167     S(C,R)          Quantity supplied (thousand tons) ;
```

# EQUATIONS

This command defines the equations that describe the relationships among the variables and parameters in the model. Every equation has a name. First, you list the names of the equations with a brief description of each one. Then, you list the equations themselves. Notice that the equations use =E= to represent the equals sign.

EQUATIONS

<equation name> <equation description>
<equation name> <equation description>
<equation name> <equation description> ;
<equation name>..
<variable name> =E= <expression> ;
<equation name>..
<variable name> =E= <expression> ;
<equation name>..
<variable name> =E= <expression> ;

Part of the EQUATIONS command from SDP4.gms (shown in Annex 1) is given below:

```
176   EQUATIONS
177   DEMAND          Demand equation
178   SUPPLY          Supply equation ;
186   DEMAND(C,R)..
187   D(C,R)  = E =  DA(C,R) + DB(C,R)*P(C,R) + DC(C,R)*Y0(R) ;
188
189   SUPPLY(C,R)..
190   S(C,R)  = E =  SA(C,R) + SB(C,R)*P(C,R) ;
```

# MODEL

This command names the model and the equations that are part of the model. Usually, all the equations listed are part of the model, but not always. The format is as follows:

MODEL <name of model> / <equation name>
<equation name>
<equation name> / ;

You can put two equation names on the same line, but they must be separated by a comma. The MODEL

command in the model SDP4 (see Annex 1) is:

```
213     MODEL MARKET / DEMAND
214     SUPPLY
215     IN_OUT
216     DOM_TRADE.TQ
217     EXPORTS.X
218     IMPORTS.M
219     XQUOTA.IXT
220     MQUOTA.IMT / ;
```

# SOLVE

This command tells GAMS the method to use to solve the model. Solving the model means finding the values of the variables that make all the equations true. The format is as follows:

SOLVE <name of model> USING <method> ;

The SOLVE command in the model SDP4 (see Annex 1) is given below.

221 SOLVE MARKET USING MCP ;

All the models in this course use the MCP method for solving models.