Lecture 4.1

Introduction to GAMS

David Roland-Holst, Sam Heft-Neal, and Anaspree Chaiwan
UC Berkeley and Chiang Mai University

Training Workshop
Economywide Assessment of High Impact Animal Disease
14-18 January 2013
InterContinental Hotel, Phnom Penh, Cambodia
• Excel CGE models are easily accessible, but must be highly simplified to be tractable.
• Using a higher level programming language enables us to include more economic structure and behavior.
• The Generalized Algebraic Modeling System (GAMS) is the language of choice for this kind of work.
• **Description:**
  - automates the process of going from a mathematical statement of the problem to the solution.
  - GAMS transforms the mathematical representation to representations required by specific *Solver* engines like OSL, CPLEX,..
  - models and solves complex *linear, nonlinear* and *integer* programming problems.
  - lets you build your model in a natural, logical structure using compact algebraic statements.

• **Typical use:**
  - Optimization
As you move away from the centre of the diagram, the software is more advanced, but often less friendly.

Try the software in the second tier first if you are looking for a middle way.
GAMS for CGE Models: A Schematic View

Developement

Simulation

Analysis

Box Color Key to Software Implementation:
Green – Microsoft Excel
Yellow – GAMS

17 January 2013
Sample Transportation Problem

- Satisfy market demand, but with minimal costs of transporting the goods from producers to the markets.
- We are given the supplies at several plants and the demands at several markets for a single commodity, and we are given the unit costs of shipping the commodity from plants to markets. The economic question is: how much shipment should there be between each plant and each market so as to minimize total transport cost?
• Indices:
  - \(i\) = plants
  - \(j\) = markets

• Given Data:
  - \(a_i\) = supply of commodity of plant \(i\) (in cases)
  - \(b_j\) = demand for commodity at market \(j\) (cases)
  - \(c_{ij}\) = cost per unit shipment between plant \(i\) and market \(j\) ($/case)

• Variables:
  - costs
  - \(x_{ij}\) = amount of commodity to ship from plant \(i\) to market \(j\) (cases),
  - where \(x_{ij} \geq 0\), for all \(i, j\)
Transportation Problem (cont.)

Objective Function:

\[
\text{Minimize } \sum_{i} \sum_{j} c_{ij} x_{ij} \quad (\text{\$K})
\]

Constraints:

Observe supply limit at plant \( i \):

\[
\sum_{j} x_{ij} \leq a_i \quad \text{for all } i \quad (\text{cases})
\]

Satisfy demand at market \( j \):

\[
\sum_{i} x_{ij} \geq b_j \quad \text{for all } j \quad (\text{cases})
\]
• Model definition and *Solve* statement
• Model definition
  ▪ what is in the model (indices): *sets*
  ▪ data: *.scalars, parameters, tables*
  ▪ What you are looking for: *variables*
  ▪ relationships: *equations*
  ▪ *Model* statement
  ▪ *Solve* statement
• Declaration
  ▪ declaring the existence of something and giving it a *unique* name – “identifier”

• Definition/Assignment
  ▪ giving a specific value or form
  ▪ e.g., labels - set elements
Model Components: Sets

• Indices
• Group of elements with similar characteristics
• Define what you are considering in the model
  ▪ Producers, markets, time periods...
Set: declaration and definition

- key word: Sets
- identifiers:
  - i canning plants /seattle, san-diego/
  - j markets /market1 market in new-york
    /market2 market in chicago
    /market3 "market in topeka"
  - t time periods /1987*1990/
• Describe what you know
• Different presentation of data: dimensionality
• Scalars, parameters, and tables
### Scalars

A number

<table>
<thead>
<tr>
<th>Scalar</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>freight in dollars per case per thousand miles</td>
</tr>
<tr>
<td>Value</td>
<td>/90/  ;</td>
</tr>
</tbody>
</table>

**Identifier**: keyword

```
```
Parameters: declaration and definition

Characteristics of set elements

- **keyword**: Parameters
- **names**
  - a(i)
    - seattle: 350
    - san-diego: 600
- **domains**
  - b(j)
    - new-york: 325
    - chicago: 300
    - topeka: 275

```plaintext
Capacity of plant i in cases
seattle 350
san-diego 600
Demand at market j in cases
new-york 325
chicago 300
topeka 275
```
### Table d(i,j)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Distance in thousands of miles</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>identifier</strong></td>
<td><strong>domain</strong></td>
</tr>
<tr>
<td>Seattle</td>
<td>New-York: 2.5, Chicago: 1.7, Topeka: 1.8</td>
</tr>
<tr>
<td>San-Diego</td>
<td>New-York: 2.5, Chicago: 1.8, Topeka: 1.4</td>
</tr>
</tbody>
</table>

*Note:* Distances are approximate.
Direct assignment of data values

Example 1: declare parameter \( c \) and assign its value

\[
\text{Parameter } c(i,j) \quad \text{transport cost in thousands of dollars per case;}
\]

\[
c(i,j) = f \times d(i,j) / 1000;
\]

Valid only if the values of \( f \) and \( d(i,j) \) are previously assigned

Example 2: Assignment of a value to an element

\[
c('seattle', 'new-york') = 0.40;
\]
Exponent operator

- $x^{**n}$
  - $x$ should always have a positive value
  - $n$ can be any number
- `power(x,n)`
  - Positive or negative value of $x$
  - $n$ is integer
Index Operations

- **sum**
  summation over controlling index,
  \[ x_1 + x_2 + \ldots + x_{10} = \sum_{k=1}^{10} x_k \]
  \[ \text{sum}(k, x(k)) \]

- **prod**
  product over controlling index
  \[ x_1 \cdot x_2 \cdot \ldots \cdot x_{10} = \prod_{k=1}^{10} x_k \]
  \[ \text{prod}(k, x(k)) \]

- **smin, smax**
  minimum and maximum over controlling index
  \[ \text{smin}(k, x(k)) \]
Model Components: Variables

- What you are looking for
- Declaration, assignment of type, assignment of bounds and/or initial values

**Variables**

\[
\begin{align*}
    x(i,j) & \quad \text{shipment quantities in cases} \\
    z & \quad \text{total transportation costs in thousands of dollars} \\
\end{align*}
\]

Positive Variable \( x \);
### Variable Types

<table>
<thead>
<tr>
<th>Variable Type</th>
<th>Allowed Range of Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>free (default)</td>
<td>-∞ to +∞</td>
</tr>
<tr>
<td>positive</td>
<td>0 to +∞</td>
</tr>
<tr>
<td>negative</td>
<td>-∞ to 0</td>
</tr>
<tr>
<td>binary</td>
<td>0 or 1</td>
</tr>
<tr>
<td>integer</td>
<td>0, 1, ..., 100 (default)</td>
</tr>
</tbody>
</table>
### Variable attributes

<table>
<thead>
<tr>
<th>Variable attribute</th>
<th>Variable suffix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lower bound</td>
<td>.lo</td>
<td>The lower bound for the variable. Set by the user either explicitly or through default values.</td>
</tr>
<tr>
<td>upper bound</td>
<td>.up</td>
<td>The upper bound for the variable. Set by the user either explicitly or through default values.</td>
</tr>
<tr>
<td>fixed value</td>
<td>.fx</td>
<td>The fixed value for the variable.</td>
</tr>
<tr>
<td>activity level</td>
<td>.l</td>
<td>The activity level for the variable. This is also equivalent to the current value of the variable. Receives new values when a model is solved.</td>
</tr>
<tr>
<td>marginal or dual value</td>
<td>.m</td>
<td>The marginal value for the variable. Receives new values when a model is solved.</td>
</tr>
<tr>
<td>scale value</td>
<td>.scale</td>
<td>This is the scaling factor on the variable. This is normally an issue with nonlinear programming problems and is discussed in detail later.</td>
</tr>
<tr>
<td>branching priority value</td>
<td>.prior</td>
<td>This is the branching priority value of a variable. This parameter is used in mixed integer programming models only, and is discussed in detail later.</td>
</tr>
</tbody>
</table>

**Example:**

`y.fx = 1000;`
Model Components: Equations

- Relationships among variables and parameters
- Declaration, definition

```
Keyword | Equations
-------|----------
cost   | define objective function
supply(i) | observe supply limit at plant i
demand(j) | satisfy demand at market j

\[
\begin{align*}
\text{cost} & : z = \text{sum}(\text{i, j}, c(i,j)\times x(i,j)) ; \\
\text{supply}(i) & : \text{sum}(j, x(i,j)) = \text{a}(i) ; \\
\text{demand}(j) & : \text{sum}(i, x(i,j)) = \text{b}(j) ;
\end{align*}
\]
```
### Equation Types

<table>
<thead>
<tr>
<th>Equation type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>=e=</td>
<td>Equality: rhs must equal lhs</td>
</tr>
<tr>
<td>=g=</td>
<td>Greater than: lhs must be greater than or equal to rhs</td>
</tr>
<tr>
<td>=l=</td>
<td>Less than: lhs must be less than or equal to rhs</td>
</tr>
<tr>
<td>=n=</td>
<td>No relationships enforced between lhs and rhs. This equation type is rarely used.</td>
</tr>
</tbody>
</table>
### Equation Values

<table>
<thead>
<tr>
<th>Type</th>
<th>.lo</th>
<th>.up</th>
<th>.l</th>
</tr>
</thead>
<tbody>
<tr>
<td>=e=</td>
<td>rhs</td>
<td>rhs</td>
<td>rhs</td>
</tr>
<tr>
<td>=l=</td>
<td>-inf</td>
<td>rhs</td>
<td>rhs</td>
</tr>
<tr>
<td>=g=</td>
<td>rhs</td>
<td>inf</td>
<td>rhs</td>
</tr>
<tr>
<td>=n=</td>
<td>-inf</td>
<td>inf</td>
<td>any</td>
</tr>
</tbody>
</table>
Quick Note: ‘=‘ and ‘=e=‘

• ‘=‘
  ▪ used only in direct assignments
  ▪ gives a desired value to a parameter
  ▪ executed before solver is called
  ▪ must not involve variables

• ‘=e=‘
  ▪ used only in equation definitions
  ▪ executed after the solver is called
  ▪ must contain variables
Model Components:

- Model statement
  - Model – collection of equations
  - Declaration and definition

Examples:

```plaintext
Model transport /all/
```

```plaintext
Model nortonL linear version /cb, rc, dfl, bc, obj/
nortonN nonlinear version /cb, rc, dfn, bc, obj/
```
<table>
<thead>
<tr>
<th>Model Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LP</strong></td>
<td>Linear Programming. There are no nonlinear terms or discrete (binary or integer) variables in the model</td>
</tr>
<tr>
<td><strong>NLP</strong></td>
<td>Nonlinear programming. There are nonlinear terms involving only “smooth” functions in the model, but no discrete variables</td>
</tr>
<tr>
<td><strong>MIP, MINLP</strong></td>
<td>Mixed Integer Programming. The model can contain discrete (integer or binary) variables</td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td>DNLP, RMIP, MPEC, MCP, CNS</td>
</tr>
</tbody>
</table>
Model Components: \textbf{Solve} statement

- Objective variable: scalar and type free

\texttt{Solve_transport\ using\ lp\ minimizing\ z;}

\begin{itemize}
  \item \textbf{keyword:} \texttt{Solve}
  \item \textbf{solution procedure:} \texttt{transport}
  \item \textbf{name of variable to be optimized:} \texttt{lp}
  \item \textbf{name of the model:} \texttt{minimizing}
  \item \textbf{keyword:} \texttt{z ;}
\end{itemize}
Display Statement

**Display** \( x.l, x.m \);

- Only non-default values are displayed
- Default value is generally zero
- for the .lo and .up subtypes of variables and equations the default values can be zero, -INF or +INF
- Display control examples:
  
  ```
  option decimals = 1;  // number of digits after decimal point for all displayed variables
  option x : 5;         // number of digits after decimal point for variable x;
  ```
1. Model is translated into *Solver* language
2. Comprehension aid is written to output
3. Error check, if error – program termination
4. *Solver* solves the model
5. GAMS reports the status of the solution and load solution values from *Solver*
Reports results, facilitates debugging

- Echo Print
- Reference Map
- Equation Listing
- Column Listing
- Model Statistics
- Status Report Solver report
- Solution Report
- Report Summary
- Results
• Copy of input file with line numbers
• Dollar-print-control statements
  ▪ output control
  ▪ start in column 1
  ▪ examples:
    $Title TEXT print TEXT on top of each page
    $Ontext
    $Offtext comments
    $Offlisting no printing of the input file
• Summaries of the input file for debugging purposes
• Two parts of cross-reference map
  ▪ alphabetical list of all entities and a coded reference of each appearance
  ▪ List of all entries grouped by type
• Does GAMS generate the model you intended?
• Describe equations for specific values of set elements and parameters
• Nonlinear system – first order Taylor approximation (i.e. linear approximation)
• shows the coefficients of three specific variables for each generic variable
• control of equation and column listing:

    option limrow = r, limcol = c ;

r – desired number of equations

c – desired number of columns
Model Statistics

- BLOCK counts – number of generic equations and variables
- SINGLE counts refer to individual equations and variables
- NON ZERO ELEMENTS - number of non-zero coefficients in the problem matrix
- NONLINEAR N-Z – number of nonlinear entries in nonlinear models
• CODE LENGTH, DERIVATIVE POOL, CONSTANT POOL – type of nonlinearity in nonlinear models

• GENERATION TIME - time used since the syntax check finished
• solve summary
• Desired SOLVER STATUS: 1 NORMAL COMPLETION
• Desired MODEL STATUS:
  ▪ Linear model: 1 OPTIMAL
  ▪ Nonlinear model: 2 LOCALLY OPTIMAL
  ▪ Integer model: 8 INTEGER SOLUTION
<table>
<thead>
<tr>
<th>solvestat</th>
<th>solver status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>normal completion</td>
</tr>
<tr>
<td>2</td>
<td>iteration interrupt</td>
</tr>
<tr>
<td>3</td>
<td>resource interrupt</td>
</tr>
<tr>
<td>4</td>
<td>terminated by solver</td>
</tr>
<tr>
<td>5</td>
<td>evaluation error limit</td>
</tr>
<tr>
<td>6</td>
<td>unknown</td>
</tr>
<tr>
<td>7</td>
<td>(unused)</td>
</tr>
<tr>
<td>8</td>
<td>error preprocessor error</td>
</tr>
<tr>
<td>9</td>
<td>error setup failure</td>
</tr>
<tr>
<td>10</td>
<td>error solver failure</td>
</tr>
<tr>
<td>11</td>
<td>error internal solver error</td>
</tr>
<tr>
<td>12</td>
<td>error post-processor error</td>
</tr>
<tr>
<td>13</td>
<td>error system failure</td>
</tr>
<tr>
<td></td>
<td>Model Status</td>
</tr>
<tr>
<td>----</td>
<td>-------------------------</td>
</tr>
<tr>
<td>1</td>
<td>optimal</td>
</tr>
<tr>
<td>2</td>
<td>locally optimal</td>
</tr>
<tr>
<td>3</td>
<td>unbounded</td>
</tr>
<tr>
<td>4</td>
<td>infeasible</td>
</tr>
<tr>
<td>5</td>
<td>locally infeasible</td>
</tr>
<tr>
<td>6</td>
<td>intermediate infeasible</td>
</tr>
<tr>
<td>7</td>
<td>intermediate non-optimal</td>
</tr>
<tr>
<td>8</td>
<td>integer solution</td>
</tr>
<tr>
<td>9</td>
<td>intermediate non-integer</td>
</tr>
<tr>
<td>10</td>
<td>integer infeasible</td>
</tr>
<tr>
<td>11</td>
<td>(unused)</td>
</tr>
<tr>
<td>12</td>
<td>error unknown</td>
</tr>
<tr>
<td>13</td>
<td>error no solution</td>
</tr>
</tbody>
</table>
• message identifying the solver and its authors
• diagnostic messages if anything unusual was detected
• specific performance details
• Results of optimization
• Four levels of equations – low bound, level value, upper bound, and marginal
• Values
  “.” – zero
  EPS – close to zero
  INFES - infeasible
  NOPT - marginal values of the wrong sign
  UNBND – unbounded
• Turned off by line: \textit{option solprint = off}
Report Summary

- total number of non-optimal, infeasible, and unbounded rows and columns
- INFES – row/column is infeasible. The level value is not between upper and lower bounds
- NOPT – row/column is non-optimal. Marginal value is incorrect
- UNBND – row/column is unbounded
Error Report

- coded message following the line with error
- Look for ****
- contain a "$" directly below the point at which the compiler thinks the error occurred
- Always check carefully for the cause of the first error
- Look at the previous line (especially for missing semicolons) if nothing seems obvious
Remarks

• You are free to use either upper- or lower- case letters

• GAMS treats singular and plural synonymously
  ▪ E.g., Set and Sets

• Multi-word names are not allowed. Use hyphens
  ▪ E.g., ‘New-York’ instead of ‘New York’
Most users of GAMS can run the system in the Integrated Development Environment (IDE).

When GAMS-IDE is started, a window will appear with a menu bar along the top and a main Edit Window for GAMS applications. As with most such systems, input and output operations are controlled by the File pull down menu, with other menu items used in edit operations, and in running the GAMS system.

The IDE version provides for standard, mouse-driven editing of input files in the main GAMS Edit Window. If the appropriate file is not already displayed, use the New or Open commands on the File menu to activate one. Then create or correct the file with the mouse and tools provided on the Edit and Search menus. The Matching Parenthesis button helps with the many parentheses in GAMS by skipping the cursor to the parenthesis that corresponds to the one it is now positioned in front of. The Find in file is also a useful tool, if you work with a complex model.
• Users should begin each session by selecting a "project". A project is a system file you save but never have to touch. Still, its location is important because the folder (directory) of the current project file is where (.gms) input and (.lst) output files are saved by default.

• This allows you to easily keep all the input and output files for any task together in the same directory, and use different directories for different projects.

• The starting project file (if any) is shown at the top of the main GAMS window.
Initiating a Project

In the picture below, the starting project file is “W:\WRK\GAMS\my project.gpr”. To select another, or create a new one, use the Project item on the File menu.
## 8 A GAMS Tutorial

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets</td>
<td>Echo Print</td>
</tr>
<tr>
<td>Declaration</td>
<td>Reference Maps</td>
</tr>
<tr>
<td>Assignment of members</td>
<td>Equation Listings</td>
</tr>
<tr>
<td>Data</td>
<td>Status Reports</td>
</tr>
<tr>
<td>(Parameters, Tables, Scalars)</td>
<td>Results</td>
</tr>
<tr>
<td>Declaration</td>
<td></td>
</tr>
<tr>
<td>Assignment of values</td>
<td></td>
</tr>
<tr>
<td>Variables</td>
<td></td>
</tr>
<tr>
<td>Declaration</td>
<td></td>
</tr>
<tr>
<td>Assignment of type</td>
<td></td>
</tr>
<tr>
<td>Assignment of bounds and/or initial values (optional)</td>
<td></td>
</tr>
<tr>
<td>Equations</td>
<td></td>
</tr>
<tr>
<td>Declaration</td>
<td></td>
</tr>
<tr>
<td>Definition</td>
<td></td>
</tr>
<tr>
<td>Model and Solve statements</td>
<td></td>
</tr>
<tr>
<td>Display statement (optional)</td>
<td></td>
</tr>
</tbody>
</table>
3.3 DATA TYPES AND DEFINITIONS

**Style 1**

**DATA**
- set declarations and definitions
- parameter declarations and definitions
- assignments
- displays

**MODEL**
- variable declarations
- equation declarations
- equation definitions
- model definitions

**SOLUTION**
- solve
- display

**Style 2**

**MODEL**
- set declarations
- parameter declarations
- variable declarations
- equation declarations
- equation definitions
- model definition

**DATA**
- set definitions
- parameter definitions
- assignments
- displays

**SOLUTION**
- solve
- display
$TITLE APPLICATION OF THE CGE123 MODEL IN GAMS

$Context

This file presents a GAMS-based implementation of the 123 CGE model developed at the World Bank and involving papers by Jaime de Melo, Sherman Robinson, Jeff Lewis, Delfin Go, Pekka Sinko, and Shanta Devarajan (in various combinations of authors). The basic theory is spelled out in the paper:


This application is based on the description of the 123 model in Applied Methods for Trade Policy Analysis: A Handbook.


The Equation numbers match those in the chapter, as do variable definitions. The core data also follow from the example provided by Devarajan et al and circulated in spreadsheet form -- the macro-economic accounts for Sri Lanka in 1991. All values have been scaled relative to GDP. These data are listed below.

Rs Billion       Output=1

National Accounts

<table>
<thead>
<tr>
<th>Output (Value Added)</th>
<th>Wages</th>
</tr>
</thead>
<tbody>
<tr>
<td>324.6940</td>
<td>163.3200</td>
</tr>
<tr>
<td>1.0000</td>
<td>0.5030</td>
</tr>
</tbody>
</table>
Parameter declarations

Parameters

* Here we define a mix of policy and functional parameters

- $P_{Wm}$: world price of import good
- $P_{Wm}$: world price of export good
- $cm$: import tariff
- $te$: export subsidy rate
- $ts$: sales or excise of VAT tax rate
- $ty$: direct income tax rate
- $tr$: government transfers
- $ft$: foreign transfers to government
- $re$: foreign remittances to private sector
- $sr$: average savings rate
- $X$: aggregate output
- $G$: government demand
- $B$: balance of trade
- $at$: technical shift term for CET expression
- $OMEGA$: export transformation elasticity
- $ct$: the CET exponential term
- $aq$: technical shift term for CES expression
- $sigma$: import substitution elasticity
- $rq$: the CES exponential term
- $bq$: the CES weight term in the Armington function
- $bt$: the CET weight term in the national product function
- $values(vars,exp)$: a table to hold experiment values

*Initialization of parameters

- $P_{Wm} = 0.0860$
- $P_{Wm} = 1.0107$
- $cm = 0.1287$
- $te = 0.0107$
- $ts = 0.0839$
- $ty = 0.0350$
- $tr = 0.1237$
Equation definitions

Equations

* Real flow equations with numbering following Devadasan et al
  EQ621  definition of the product transformation surface
  EQ622  definition of the Armington aggregation function
  EQ623  expenditure definition of national income
  EQ624  supply ratios based on EQ621
  EQ625  supply ratios based on EQ622

* Nominal flows
  EQ626  government tax income definition
  EQ627  value added definition of national income
  EQ628  savings definition
  EQ629  consumption definition

* Price equations
  EQ630  internal import prices
  EQ631  internal export prices
  EQ632  consumer prices
  EQ633  national product price index
  EQ634  producer prices for the composite good
  EQ635  real exchange rate

* Market clearing Equations
  EQ636  domestic good excess demand
  EQ637  Armington composite excess demand
  EQ638  capital and current account relationship
  EQ639  savings investment linkage

* Objective function

OBJ  This is the fake objective function for GAMS
In this next section we implement a free trade experiment, along the lines of the Devarajan et al chapter. We set the import tariff and export tax to zero.
Running in GAMS IDE

model1.iterlim = 1000; model1.optfile=4;
Solve model1 maximizing GDP using nlp;

values(vars,exp) = endog.1(vars);

$OnText
In this next section we implement a free trade experiment,
along the lines of the Devarajan et al chapter. We set the import
tariff and export tax to zero

$OffText

tm = 0;
te = 0;

option nlp=minos5;
model1.iterlim = 1000; model1.optfile=4;
Solve model1 maximizing GDP using nlp;
The solution screen

GAMS Rev 120  Copyright (C) 1987-2001 GAMS Development. All rights reserved
Licenses: Economics Department

--- Starting compilation
--- CGE123.gms(133) 1 Mb
--- Starting execution
--- CGE123.gms(276) 1 Mb
--- Generating model model1
--- CGE123.gms(277) 2 Mb
--- 20 rows, 21 columns, and 55 non-zeroes.
--- CGE123.gms(277) 2 Mb
--- Executing MINOS5

MINOS5         Feb 14, 2001 WIN.MS.MS 19.0 108.043.039.WAT

The options file could not be found -- using defaults
Work space allocated            -- 0.04 Mb
Reading data...                
Reading nonlinear code...

<table>
<thead>
<tr>
<th>Major</th>
<th>Minor</th>
<th>Ninf</th>
<th>Ninf, Objective</th>
<th>Viol</th>
<th>RG</th>
<th>NSB</th>
<th>Mcon</th>
<th>Penalty</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.0000000000E+00</td>
<td>8.4E-02</td>
<td>0.0E+00</td>
<td>14</td>
<td>3</td>
<td>1.0E+01</td>
<td>1.0E+00</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>0</td>
<td>1.13158934E+00</td>
<td>5.9E-03</td>
<td>0.0E+00</td>
<td>0</td>
<td>4</td>
<td>1.0E+01</td>
<td>1.0E+00</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1.12988912E+00</td>
<td>8.9E-06</td>
<td>0.0E+00</td>
<td>0</td>
<td>5</td>
<td>1.0E+01</td>
<td>1.0E+00</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1.12988053E+00</td>
<td>1.5E-11</td>
<td>0.0E+00</td>
<td>0</td>
<td>6</td>
<td>1.0E+01</td>
<td>1.0E+00</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1.12986053E+00</td>
<td>2.2E-16</td>
<td>0.0E+00</td>
<td>0</td>
<td>7</td>
<td>0.0E+00</td>
<td>1.0E+00</td>
</tr>
</tbody>
</table>

EXIT -- OPTIMAL SOLUTION FOUND

Major, Minor itns     | 5 | 14
Objective function   | 1.1298605309416E+00

17 January 2013
Some results

---

333 PARAMETER values a table to hold experiment values

<table>
<thead>
<tr>
<th>benchmark</th>
<th>freetrade</th>
<th>revenue_r~</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>0.3276</td>
<td>0.3371</td>
</tr>
<tr>
<td>M</td>
<td>0.5030</td>
<td>0.5138</td>
</tr>
<tr>
<td>Dσ</td>
<td>0.6724</td>
<td>0.6626</td>
</tr>
<tr>
<td>Dd</td>
<td>0.6724</td>
<td>0.6626</td>
</tr>
<tr>
<td>Qσ</td>
<td>1.1754</td>
<td>1.1750</td>
</tr>
<tr>
<td>Qd</td>
<td>1.1754</td>
<td>1.1750</td>
</tr>
<tr>
<td>Pr</td>
<td>1.0000</td>
<td>1.0197</td>
</tr>
<tr>
<td>Pm</td>
<td>1.0000</td>
<td>0.8860</td>
</tr>
<tr>
<td>Pd</td>
<td>0.9999</td>
<td>0.9435</td>
</tr>
<tr>
<td>Pt</td>
<td>1.0839</td>
<td>0.9939</td>
</tr>
<tr>
<td>Px</td>
<td>1.0000</td>
<td>0.9639</td>
</tr>
<tr>
<td>Pq</td>
<td>1.0000</td>
<td>0.9169</td>
</tr>
<tr>
<td>T</td>
<td>0.1920</td>
<td>0.1284</td>
</tr>
<tr>
<td>R</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>Sg</td>
<td>-0.0100</td>
<td>-0.0612</td>
</tr>
<tr>
<td>Y</td>
<td>1.1299</td>
<td>1.0835</td>
</tr>
<tr>
<td>C</td>
<td>0.8288</td>
<td>0.8668</td>
</tr>
<tr>
<td>S</td>
<td>0.2660</td>
<td>0.2059</td>
</tr>
<tr>
<td>Z</td>
<td>0.2454</td>
<td>0.2082</td>
</tr>
<tr>
<td>ts</td>
<td>0.0639</td>
<td>0.0639</td>
</tr>
</tbody>
</table>

EXECUTION TIME = 0.050 SECONDS

1.4 Mo WIN98-120

USER: Economics Department
Erasmus Universiteit Rotterdam

Roland-Holst 58
• The homepage of the GAMS corporation ([www.gams.com](http://www.gams.com)) contains a lot of useful information.

• From the homepage, a full user guide can be downloaded at [www.gams.com/docs/document.htm](http://www.gams.com/docs/document.htm); the user guide contains the syntax for all GAMS commands and very helpful as a reference when writing GAMS models. Note that the user guide is also available via the Help function in GAMS-IDE ([http://www.gams.com/dd/docs/tools/gamside.pdf](http://www.gams.com/dd/docs/tools/gamside.pdf)).

• All readers are advised to study the introductory chapter of this manual when starting to learn the GAMS software.
Download and Installation

- There is a free version of GAMS available for installation on your own computer. This is a limited version of GAMS, which cannot solve large problems, but it can be used for the sample models in this course.

- A free copy of the restricted, student version is available for download at [http://www.gams.com/download](http://www.gams.com/download)
Questions?