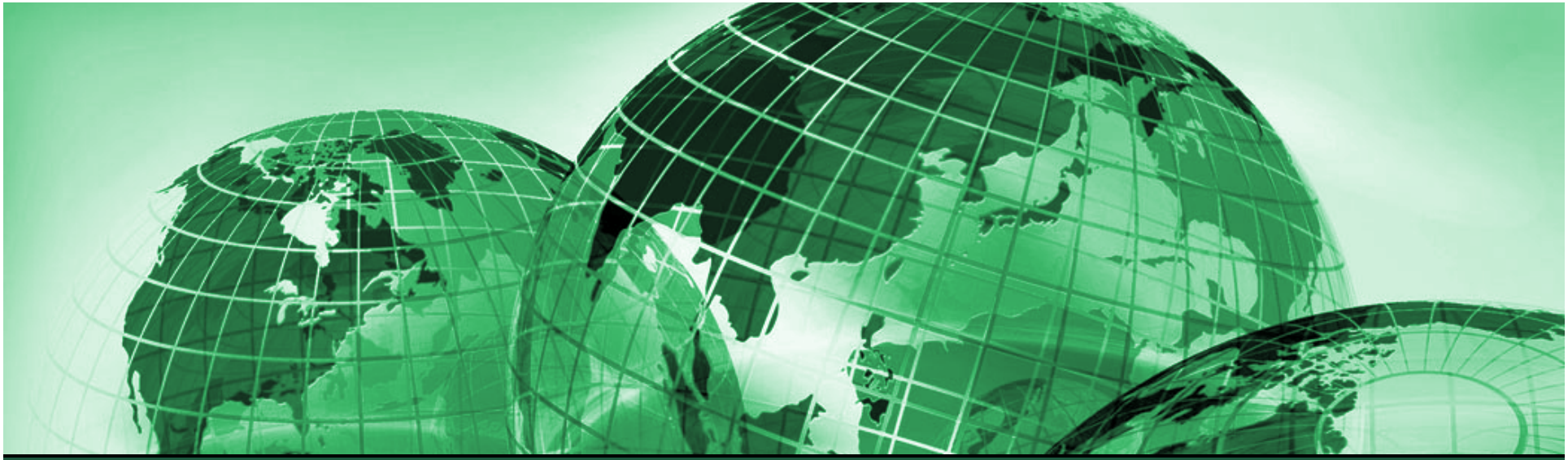


Lecture 4.1

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

Moving to GAMS

- 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.

Introduction to GAMS

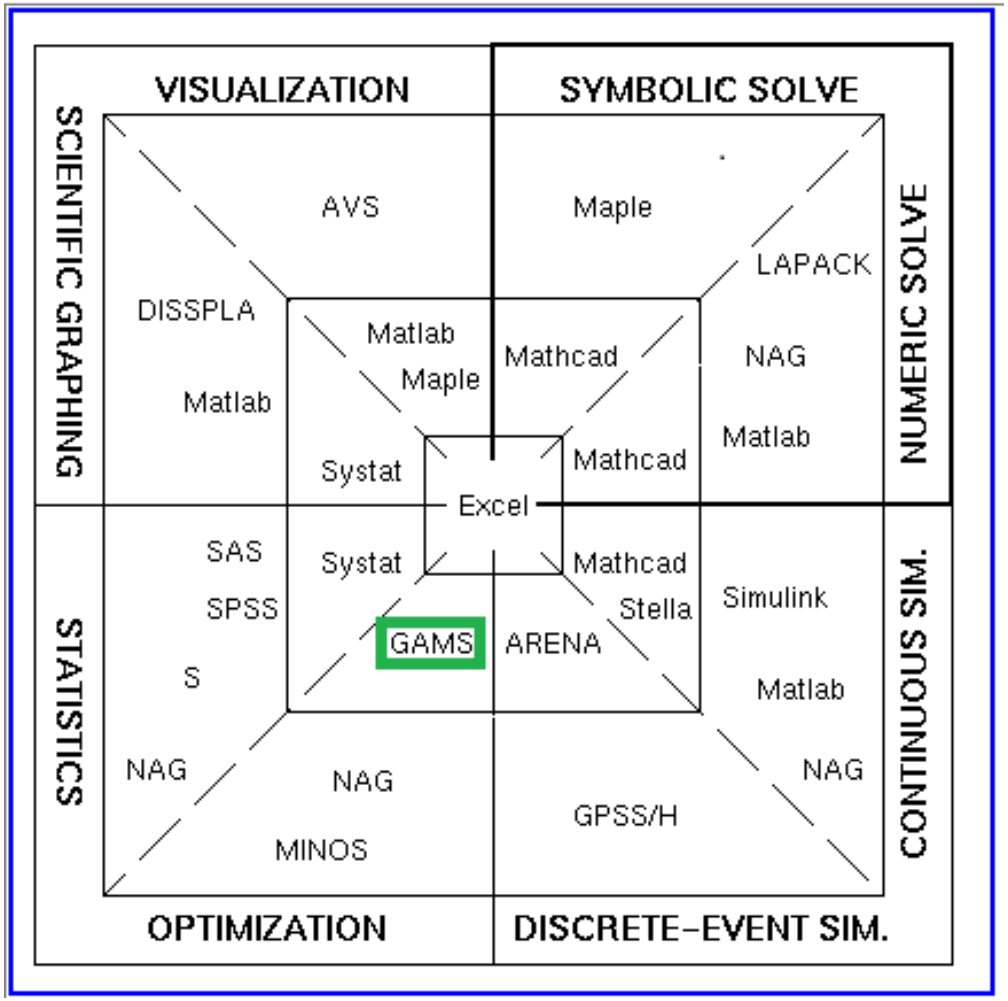
- **Description:**
  - automates the process of going from a mathematical statement of the problem to the solution.
  - GAMS transforms the mathematical representation to representations required by specific *Solver* engines like OSL,CPLEX,..
  - models and solves complex *linear*, *nonlinear* and *integer* programming problems.
  - lets you build your model in a natural, logical structure using compact algebraic statements.
- **Typical use:**
  - Optimization

Scientific Software Comparison Chart

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.

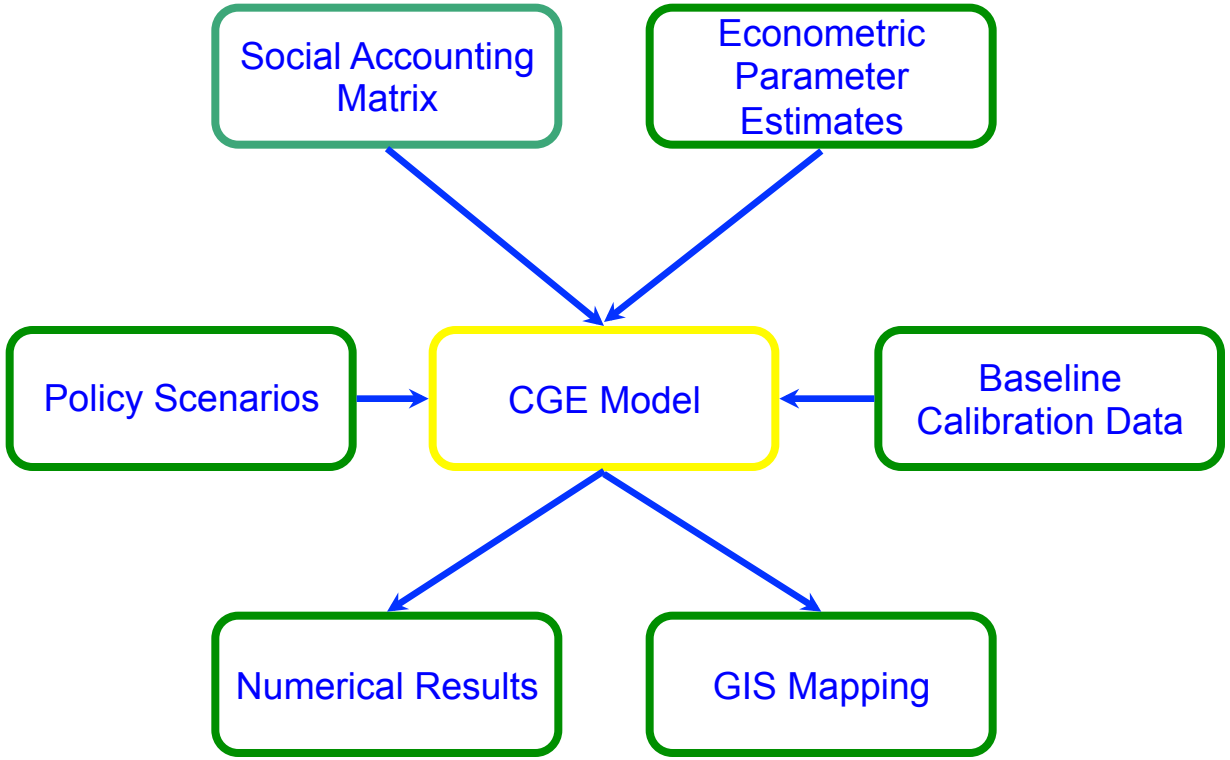


GAMS for CGE Models: A Schematic View

Development

Simulation

Analysis



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

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?

Transportation Problem (cont.)

- 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} >= 0, for all i, j

Transportation Problem (cont.)

Objective Function: $\text{Minimize} \sum_i \sum_j c_{ij} x_{ij}$ (\$K)

Constraints:

Observe supply limit at plant i: $\sum_j x_{ij} \le a_i$ for all i (cases)

Satisfy demand at market j: $\sum_i x_{ij} \ge b_j$, for all j (cases)

GAMS Program

- 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

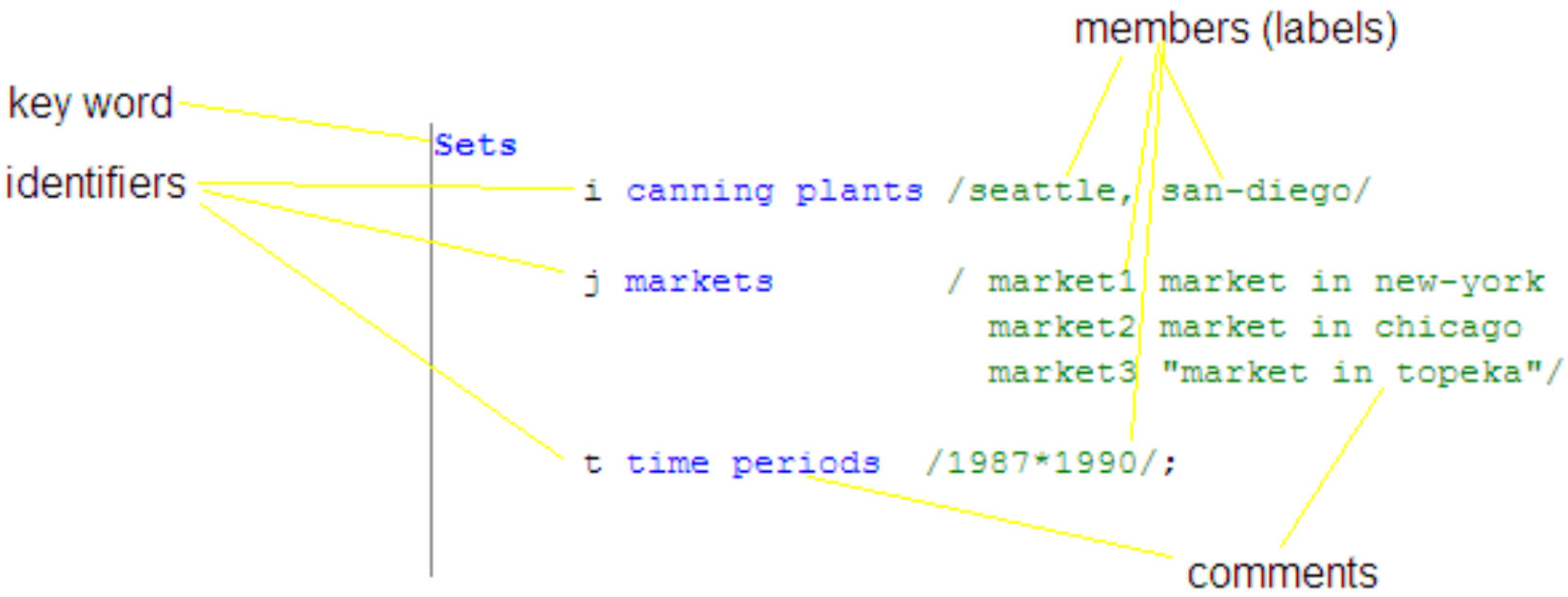
Defining Model Components

- 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



Data

- Describe what you know
- Different presentation of data: dimensionality
- Scalars, parameters, and tables

Scalars

A number

```
Scalar f  freight in dollars per case per thousand miles  /90/ ;
```

keyword

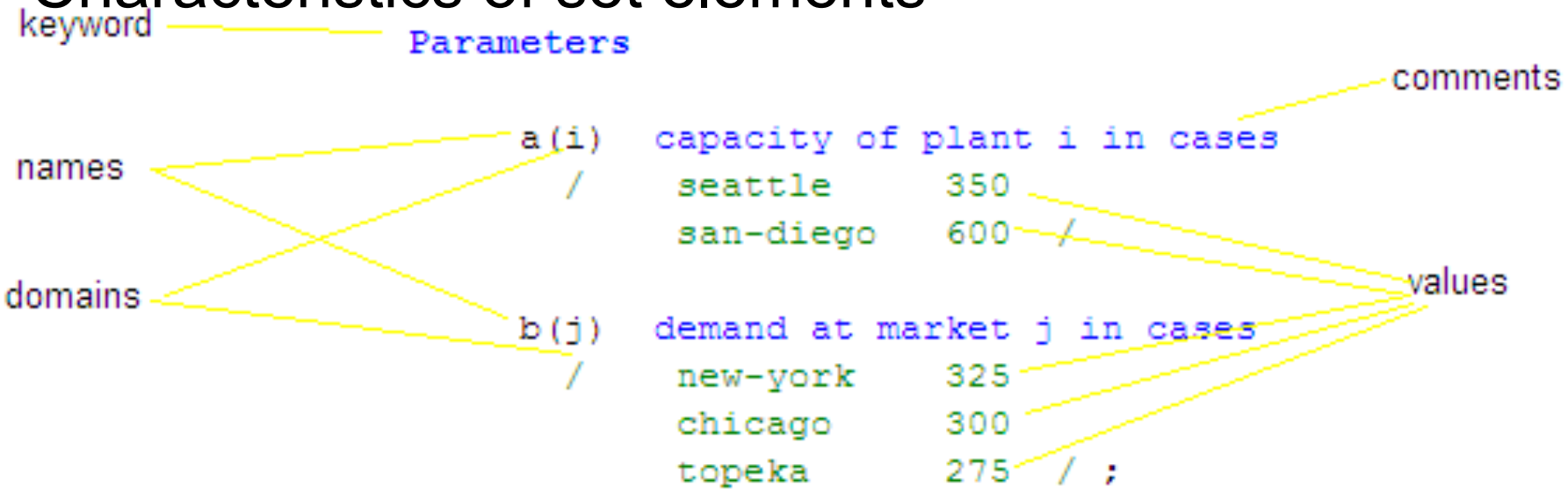
identifier

description

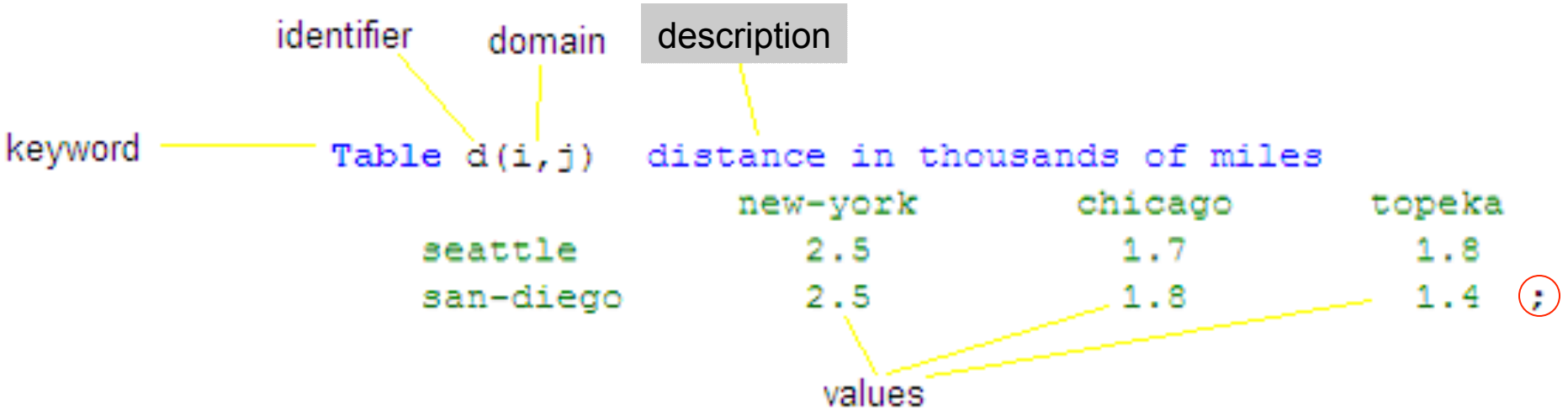
value

Parameters: declaration and definition

Characteristics of set elements



Tables


identifier
domain
description
keyword
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 ;
values

Direct assignment of data values

Example 1: declare parameter *c* and assign its value

```
Parameter c(i,j)  transport cost in thousands of dollars per case ;

          c(i,j) = f * 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,

$$x_1 + x_2 + \ldots + x_{10} = \sum_{k=1}^{10} x_k$$

```
sum(k, x(k))
```

- prod

  product over controlling index

$$x_1 \cdot x_2 \cdot \ldots \cdot x_{10} = \prod_{k=1}^{10} x_k$$

```
prod(k, x(k))
```

- smin, smax

  minimum and maximum over controlling index

```
smin(k, x(k))
```

Model Components: Variables

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

```
Variables
     x(i,j)   shipment quantities in cases
     z        total transportation costs in thousands of dollars ;

Positive Variable x ;
```

Variable Types

Variable Type	Allowed Range of Variable
`free(default)`	$-\infty$ to $+\infty$
`positive`	0 to $+\infty$
`negative`	$-\infty$ to 0
`binary`	0 or 1
`integer`	0,1,....., 100 (default)

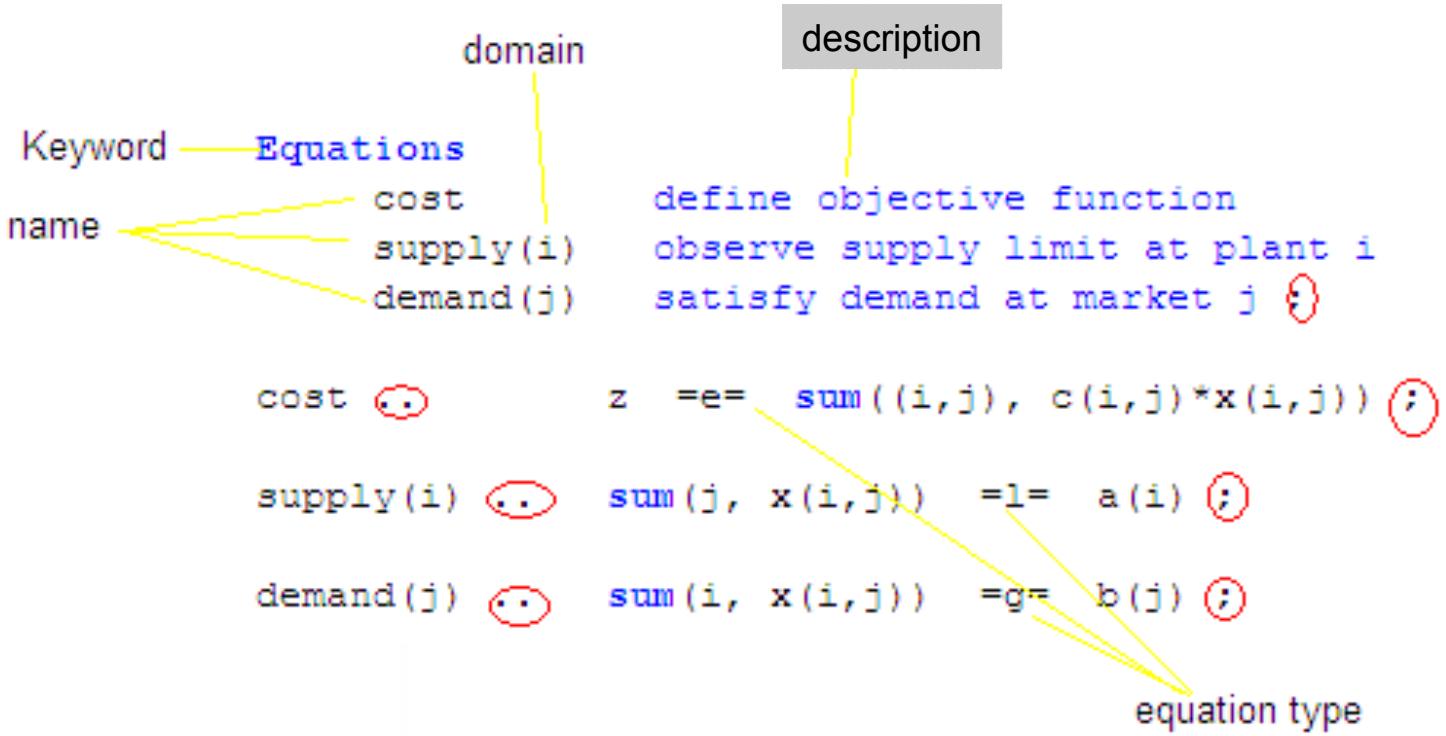
Variable attributes

Variable attribute	Variable suffix	Description
lower bound	.lo	The lower bound for the variable. Set by the user either explicitly or through default values.
upper bound	.up	The upper bound for the variable. Set by the user either explicitly or through default values.
fixed value	.fx	The fixed value for the variable.
activity level	.l	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.
marginal or dual value	.m	The marginal value for the variable. Receives new values when a model is solved.
scale value	.scale	This is the scaling factor on the variable. This is normally an issue with nonlinear programming problems and is discussed in detail later.
branching priority value	.prior	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.

Example: y.fx = 1000;

Model Components: Equations

- Relationships among variables and parameters
- Declaration, definition



Equation Types

Equation type	Description
=e=	Equality: rhs must equal lhs
=g=	Greater than: lhs must be greater than or equal to rhs
=l=	Less than: lhs must be less than or equal to rhs
=n=	No relationships enforced between lhs and rhs. This equation type is rarely used.

Equation Values

Type	.lo	.up	.l
=e=	rhs	rhs	rhs
=l=	-inf	rhs	rhs
=g=	rhs	inf	rhs
=n=	-inf	inf	any

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

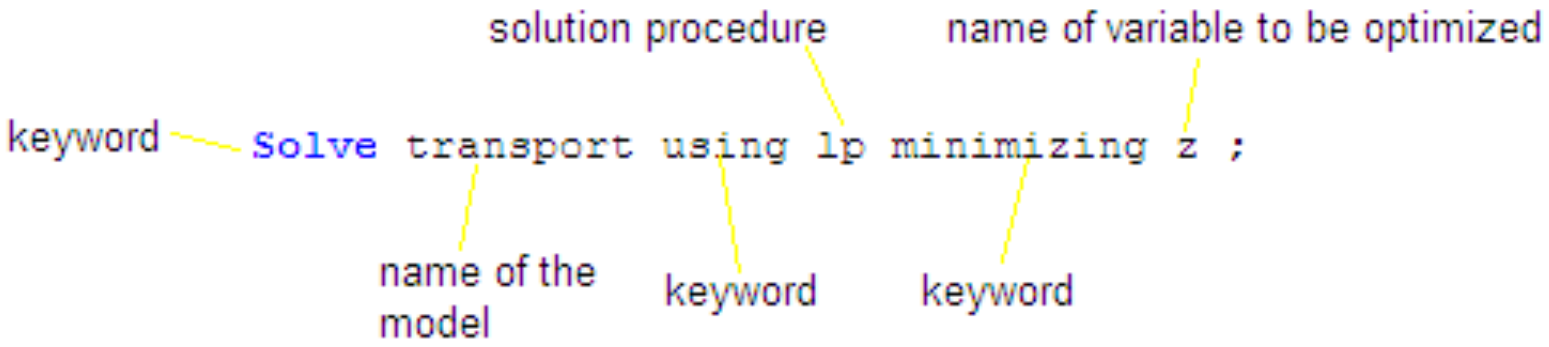
  Examples:

```
Model transport /all/ ;

Model nortonL   linear version      /cb, rc, dfl, bc, obj/
      nortonN   nonlinear version   /cb, rc, dfn, bc, obj/   ;
```

Types of Problems

Model Type	Description
LP	Linear Programming. There are no nonlinear terms or discrete (binary or integer) variables in the model
NLP	Nonlinear programming. There are nonlinear terms involving only “smooth” functions in the model, but no discrete variables
MIP, MINLP	Mixed Integer Programming. The model can contain discrete (integer of binary) variables
Other	DNLP, RMIP, MPEC, MCP, CNS

Model Components: *Solve* statement



- Objective variable: scalar and type free

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;

Solve execution

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*

GAMS Output

Reports results, facilitates debugging

- Echo Print
- Reference Map
- Equation Listing
- Column Listing
- Model Statistics
- Status Report Solver report
- Solution Report
- Report Summary
- Results

Echo Print

- 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

Reference Map

- 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

Equation Listing

- 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)

Column Listing

- 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

Model Statistics (cont.)

- CODE LENGTH, DERIVATIVE POOL, CONSTANT POOL – type of nonlinearity in nonlinear models
- GENERATION TIME - time used since the syntax check finished

Status Report

- 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

Solver Status

solvestat	solver status
1	normal completion
2	iteration interrupt
3	resource interrupt
4	terminated by solver
5	evaluation error limit
6	unknow
7	(unused)
8	error preprocessor error
9	error setup failure
10	error solver failure
11	error internal solver error
12	error post-processor error
13	error system failure

Model Status

modelstat	model status
1	optimal
2	locally optimal
3	unbounded
4	infeasible
5	locally infeasible
6	intermediate infeasible
7	intermediate non-optimal
8	integer solution
9	intermediate non-integer
10	integer infeasible
11	(unused)
12	error unknown
13	error no solution

Solver Report

- message identifying the solver and its authors
- diagnostic messages if anything unusual was detected
- specific performance details

Solution Report

- 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: *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’

GAMS Integrated Development Environment (IDE)

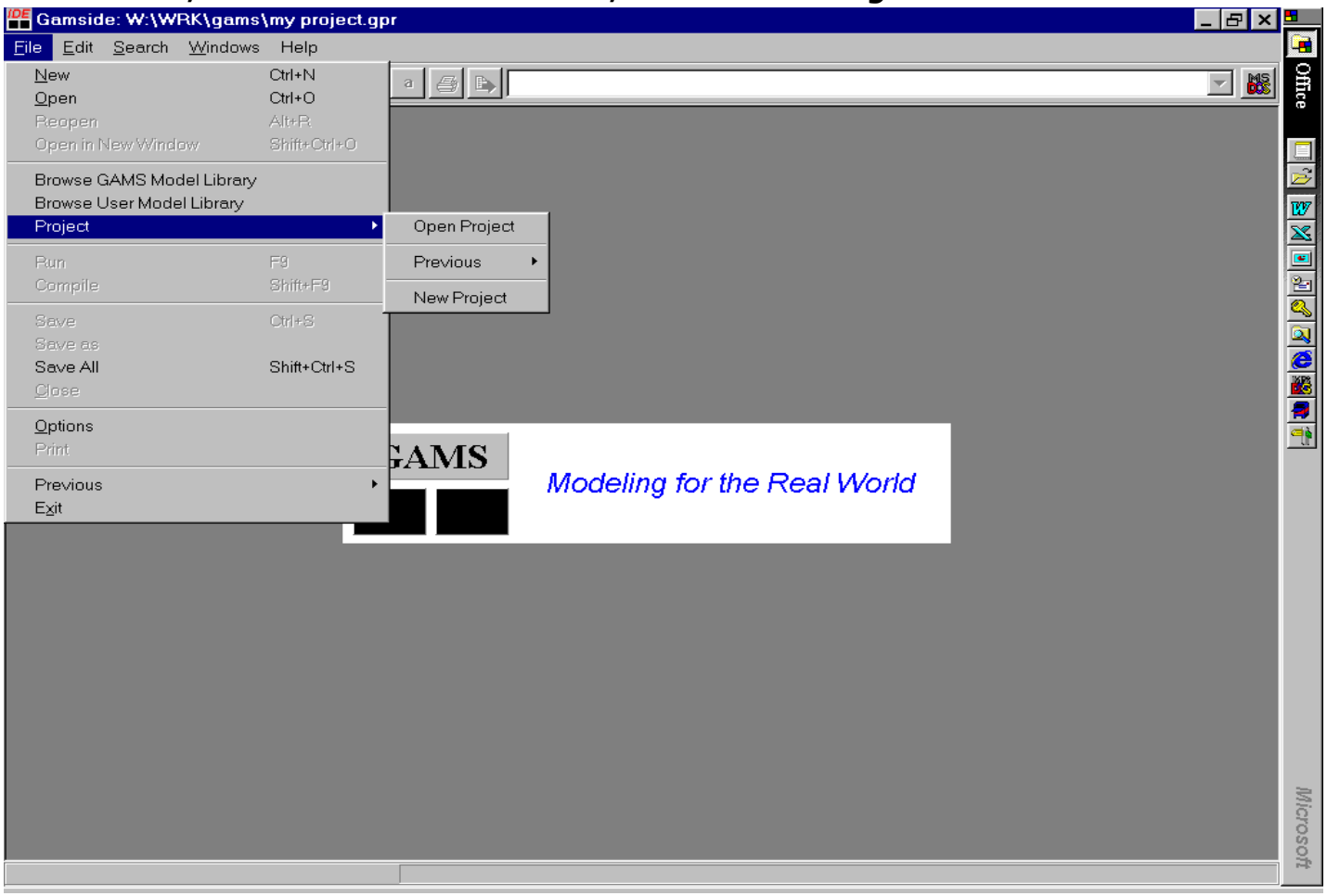
- Most users of GAMS can run the system in the Integrated Development Environment (IDE).
- When GAMS-IDE is started, a window will appear with a menu bar along the top and a main Edit Window for GAMS applications. As with most such systems, input and output operations are controlled by the **File** pull 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.

IDE - Continued

- 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.



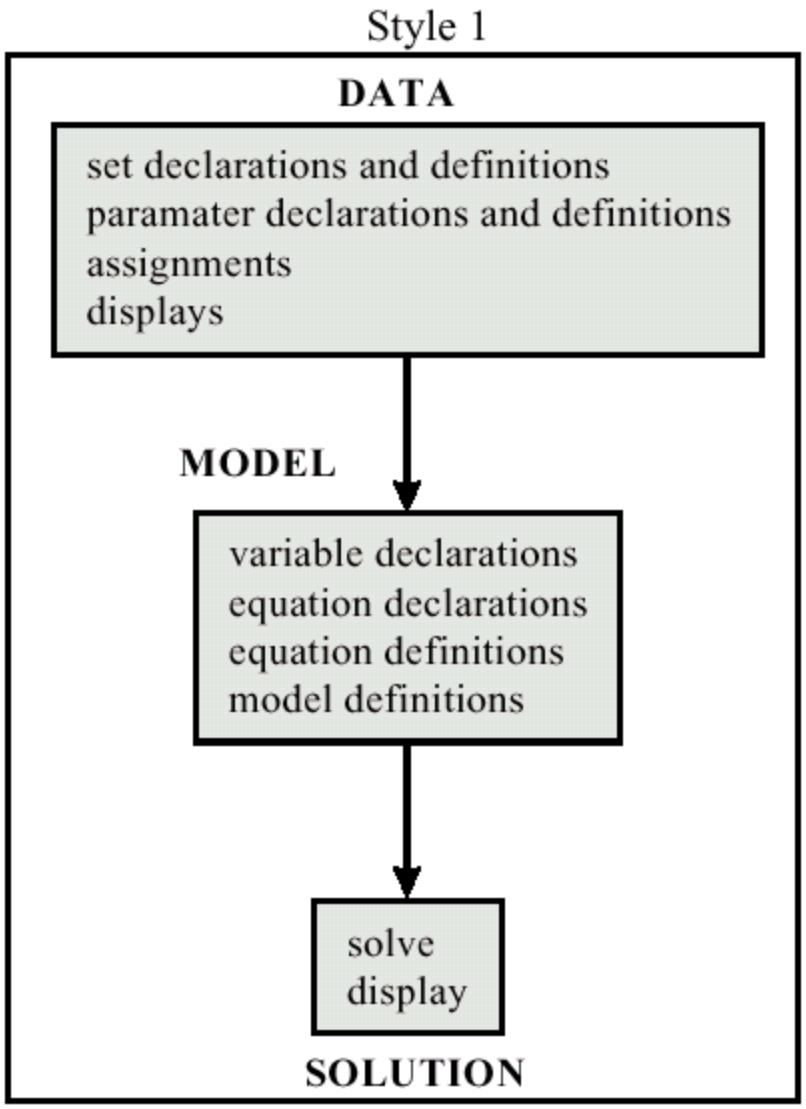
GAMS syntax: the manual

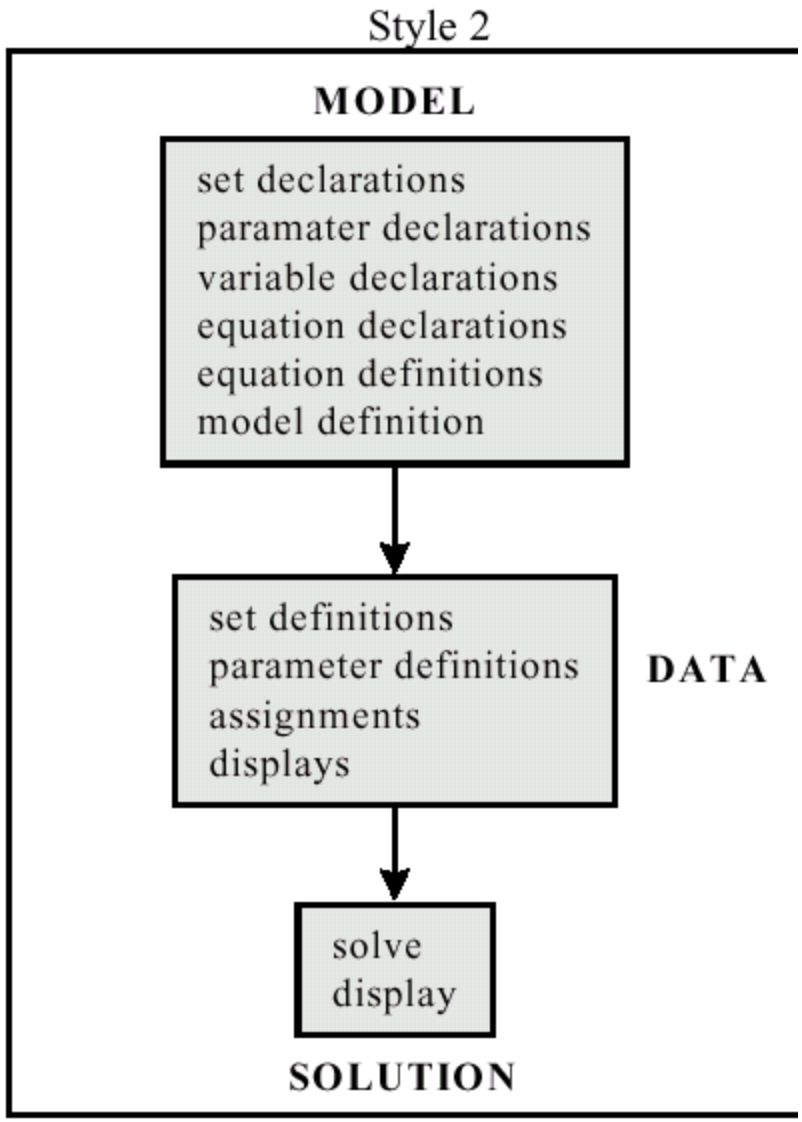
8 A GAMS TUTORIAL

Inputs	Outputs
• `Sets` Declaration Assignment of members • Data `(Parameters, Tables, Scalars)` Declaration Assignment of values • `Variables` Declaration Assignment of type • Assignment of bounds and/or initial values (optional) • `Equations` Declaration Definition • `Model` and `Solve` statements • `Display` statement (optional)	• Echo Print • Reference Maps • Equation Listings • Status Reports • Results

GAMS logic: the manual

3.3 DATA TYPES AND DEFINITIONS





GAMS-based 123 model

Gamside: D:\Models\Small\small.gpr - [D:\TEACHI~1\TINBER~1\CGEMOD~1\DISK\GAMSMO~1\CGE123\CGE123.GMS]

File Edit Searc

cge123.gms

```
$TITLE APPLICATION OF THE CGE123 MODEL IN GAMS

$Ontext
        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:

           de Melo, J. and S. Robinson, (1989).  "Product Differentiation
           and the Treatment of Foreign Trade in Computable General
           Equilibrium Models of Small Open Economies,"  Journal of
           International Economics 27: 47-67.

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

           Devarajan, S., D.S. Go, J.D. Lewis, S. Robinson, and P.
           Sinko (1997), "Simple General Equilibrium Modeling,"
           Chapter 6 in J.F. Francois and K.A. Reinert eds.,
           Applied Methods for Trade Policy Analysis:  A Handbook,
           Cambridge University Press: Cambridge UK:  156-188.

       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

        Output (Value Added)     324.6940        1.0000
          Wages                  163.3200        0.5030
```

10: 62 Insert

Parameter declarations

Gamside: D:\Models\

File Edit Search W

cge123.gms

```
Parameters
* Here we define a mix of policy and functional parameters
           PWm       world price of import good
           PWe       world price of export good
           tm        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
           rt        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

           PWm     = 0.8860;
           PWe     = 1.0107;
           tm      = 0.1287;
           te      = 0.0107;
           ts      = 0.0839;
           ty      = 0.0350;
           tr      = 0.1237;
```

142: 62 Insert

Equation definitions

Gamside: D:\Models\Small\small.gpr - [D:\TEACHI~1\TINBER~1\CGEMOD~1\DISK\GAMSMO~1\CGE123\CGE123.GMS]

File Edit Searc

cge123.gms

```
$Stitle equation definitions

Equations

*Real flow equations with numbering following Devarajan 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
           ;
```

241: 62 Insert

Model specification

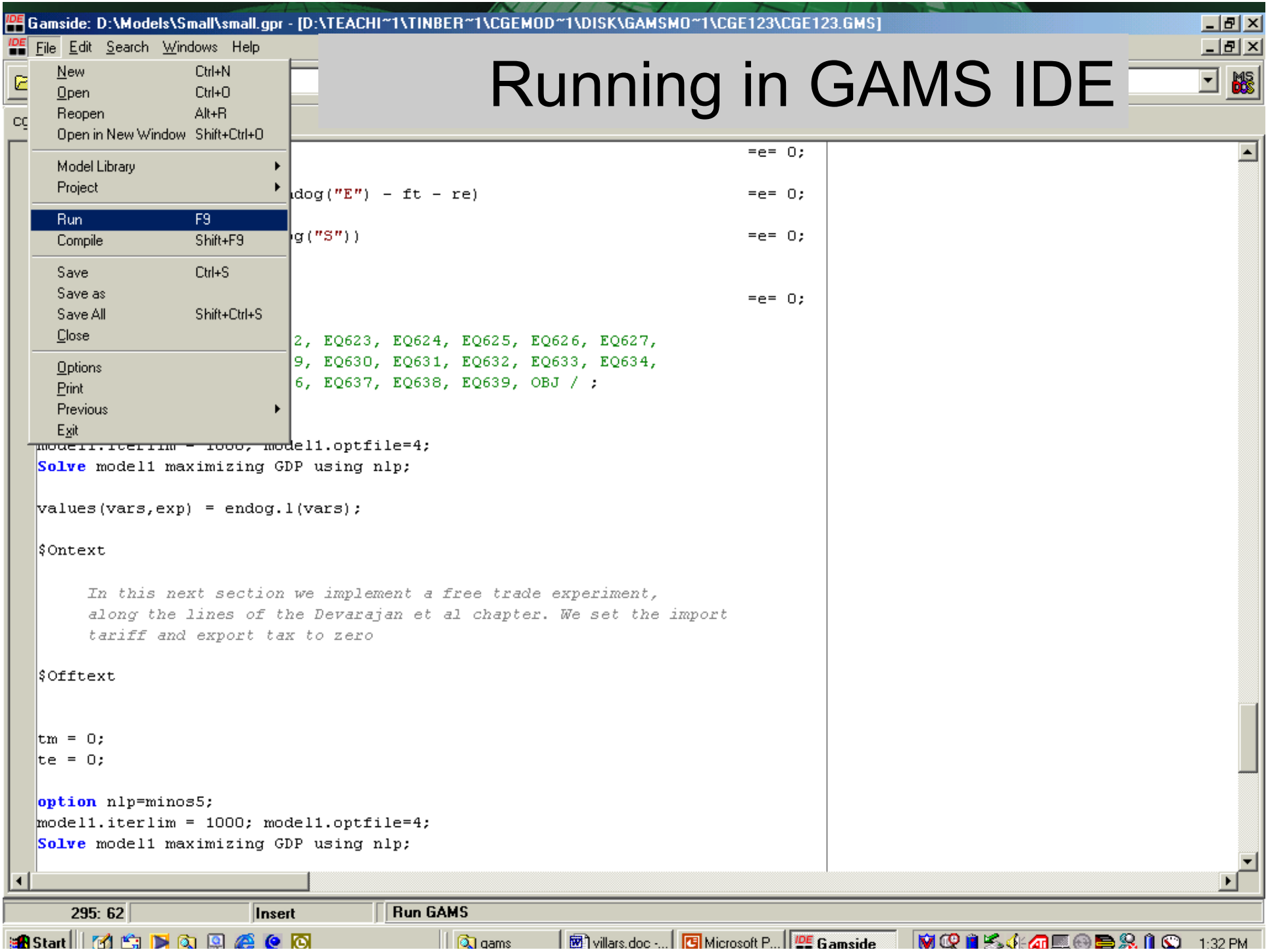
Gamside: D:\Models\Small\small.gpr - [D:\TEACHI~1\TINBER~1\CGEMOD~1\DISK\GAMSMO~1\CGE123\CGE123.GMS]

File Edit Search Winc

cge123.gms

```
 - endog("Qd")                                                    =e= 0;
 EQ638..      B
 - (PWm*endog("M") -PWe*endog("E") - ft - re)                     =e= 0;
 EQ639..      endog("Z")
 - ((1/endog("Pt")) * endog("S"))                                 =e= 0;
 OBJ..
              GDP
 - endog("Y")                                                     =e= 0;

Model model1 / EQ621, EQ622, EQ623, EQ624, EQ625, EQ626, EQ627,
               EQ628, EQ629, EQ630, EQ631, EQ632, EQ633, EQ634,
               EQ635, EQ636, EQ637, EQ638, EQ639, OBJ / ;

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

values(vars,exp) = endog.l(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;
```

295: 62 Insert



Running in GAMS IDE

The solution screen

Gamside: D:\Models\Small\small.gms [No active process]

File Edit Search Windows Help

cge123

```
GAMS Rev 120  Copyright (C) 1987-2001 GAMS Development. All rights reserved
Licensee: Economics Department                          A010221:1033CP-WIN
          Erasmus Universiteit Rotterdam                              DC418
--- Starting compilation
--- CGE123.GMS(333) 1 Mb
--- Starting execution
--- CGE123.GMS(276) 1 Mb
--- Generating model model1
--- CGE123.GMS(277) 2 Mb
---    20 rows, 21 columns, and 65 non-zeroes.
--- CGE123.GMS(277) 2 Mb
--- Executing MINOS5

 MINOS5        Feb 14, 2001 WIN.M5.M5 19.8 108.043.039.WAT

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

 Major Minor  Ninf  Sinf,Objective    Viol     RG    NSB  Ncon Penalty    Step
     1     0     1  0.00000000E+00 8.4E-02 0.0E+00   14     3 1.0E+01 1.0E+00
     2    14     0  1.13158934E+00 5.9E-03 0.0E+00    0     4 1.0E+01 1.0E+00
     3     0     0  1.12985912E+00 8.9E-06 0.0E+00    0     5 1.0E+01 1.0E+00
     4     0     0  1.12986053E+00 1.5E-11 0.0E+00    0     6 1.0E+01 1.0E+00
     5     0     0  1.12986053E+00 2.2E-16 0.0E+00    0     7 0.0E+00 1.0E+00

 EXIT -- OPTIMAL SOLUTION FOUND

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

Close | Open Log | Summary only | Update

295: 62 | Insert | Double-Click to Open File

Some results

Gamside: D:\Models\Small\small.gpr - [D:\Models\Small\CGE123.lst]

File Edit Search Windows Help

cge123.gms | cge123.lst

```
E x e c u t i o n


----    333 PARAMETER values  a table to hold experiment values

     benchmark   freetrade  revenue_r~

E       0.3276      0.3371      0.3363
M       0.5030      0.5138      0.5129
Ds      0.6724      0.6626      0.6634
Dd      0.6724      0.6626      0.6634
Qs      1.1754      1.1760      1.1760
Qd      1.1754      1.1760      1.1760
Pe      1.0000      1.0107      1.0000
Pm      1.0000      0.8860      0.8860
Pd      0.9999      0.9405      0.9360
Pt      1.0839      0.9939      1.0487
Px      1.0000      0.9639      0.9573
Pq      1.0000      0.9169      0.9144
T       0.1920      0.1284      0.1920
R       1.0000      1.0000      1.0000
Sg     -0.0100     -0.0612      0.0044
Y       1.1299      1.0835      1.0766
C       0.8288      0.8668      0.8163
S       0.2660      0.2069      0.2713
Z       0.2454      0.2082      0.2587
ts      0.0839      0.0839      0.1469


EXECUTION TIME       =        0.050 SECONDS     1.4 Mb       WIN198-120


USER: Economics Department                              A010221:1033CP-WIN
      Erasmus Universiteit Rotterdam                                DC418
```

1: 1 | Read Only | Insert

Accessing GAMS

- The homepage of the GAMS corporation (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; 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).
- 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

Questions?