Generalized Algebraic Modeling System (GAMS)

M. B. Abaee
m.b.abaee@gmail.com
M. S. Ghazizadeh
ghazizadeh@pwut.ac.ir

PWUT – Spring 2010

CONTENTS

CONTENTS

CONTENTS

CONTENTS

Email: m.b.abaee@gmail.com

CONTENTS

2.10. OUTPUT

CONTENTS

Email: m.b.abaee@gmail.com

CONTENTS

CONTENTS

CONTENTS

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

CONTENTS

CONTENTS

CONTENTS

1. INTRODUCTION

1.1. The Most Important Features of GAMS

- ✓ Capability to solve from small scale to large scale problems with small code by means of the use of index to write blocks of similar constraints from only one constraint.
- ✓ The model is independent from the solution method, and it can be solved by different solutions methods by only changing the solver.
- ✓ The translation from the mathematical model to GAMS is almost transparent since GAMS has been built to resemble mathematical programming models.
- ✓ GAMS also uses common English words, thus is easy to understand it's statements.

1. INTRODUCTION

1.2. Using GAMSIDE

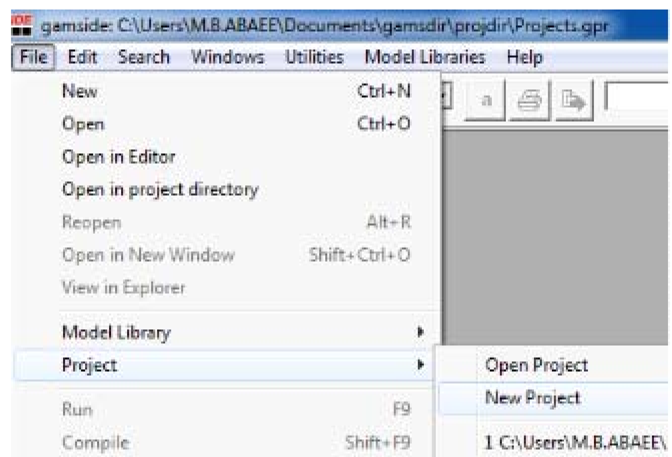
1.2.1. Install GAMS and IDE

The GAMSIDE is automatically installed when GAMS is installed.

1.2.2. Open the IDE through the icon

1.2.3. Create a Project

by going to the *File* menu.
Select to *New Project* (Later you will use your previous projects).



1. INTRODUCTION

1.2. Using GAMSIDE …

What is a Project?

The GAMSIDE employs a “Project” file for two purposes:

- First, the project location determines where all saved files are placed (to place files elsewhere use the save as dialogue) and where GAMS looks for files when executing.
- Second the project saves file names and program options associated with the effort.

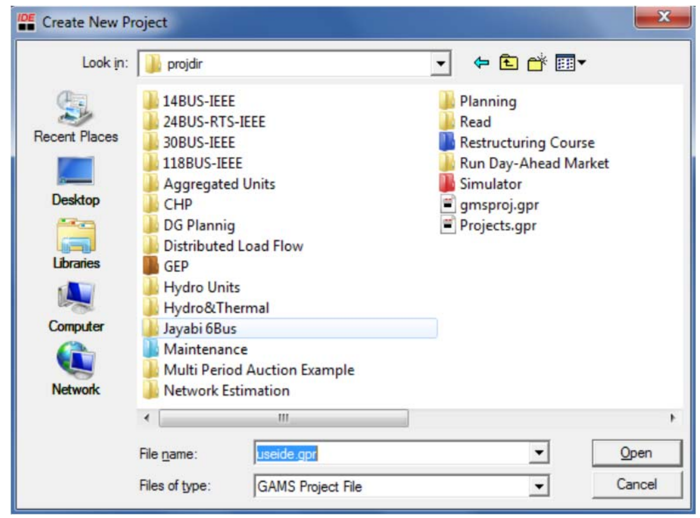
We recommend that you define a new project every time you wish to change the file storage directory.

1. INTRODUCTION

1.2. Using GAMSIDE …

1.2.4. Define a Project name and location

Put it in a directory you want to use. All files associated with this project will be saved in that directory.



In the “File name” area type in a name for the project file you wish to use. If I was doing this, I would go to a suitable subdirectory and create a subdirectory called Projdir and put in the name useide. In turn, your project name will be called useide.gpr where gpr stands for GAMS project.

1. INTRODUCTION

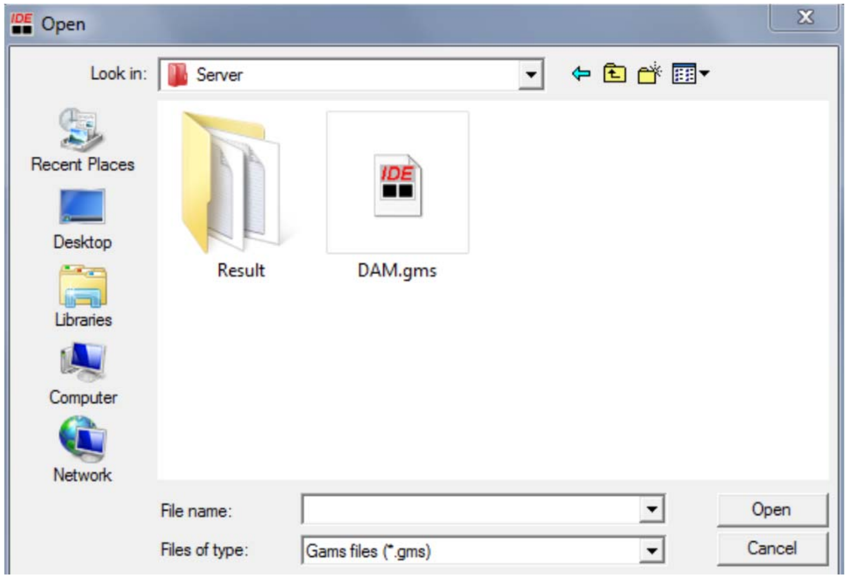
1.2. Using GAMSIDE …

1.2.5. Create or open an existing file of GAMS instructions

Several cases are possible:

a) Create a new file

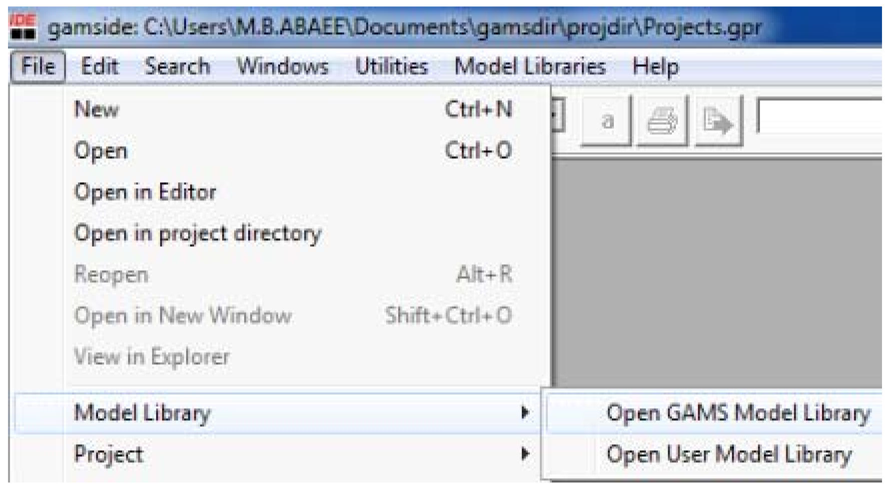
b) Open an existing file



1. INTRODUCTION

1.2. Using GAMSIDE …

c) Open a model library file (the simplest at this stage and the one we will use)



Select a model like transportation

SeqNr	Name +	Application Area	Type	Contributor	Description
314	FEASOPT1	Management Science and OR	LP	Dantzig, G B	An Infeasible Transportation Problem analyzed with Cplex option FeasOpt
122	FEEDTRAY	Chemical Engineering	MINLP	Viswanathan,	Optimum Feed Plate Location

It will be automatically saved in your project file

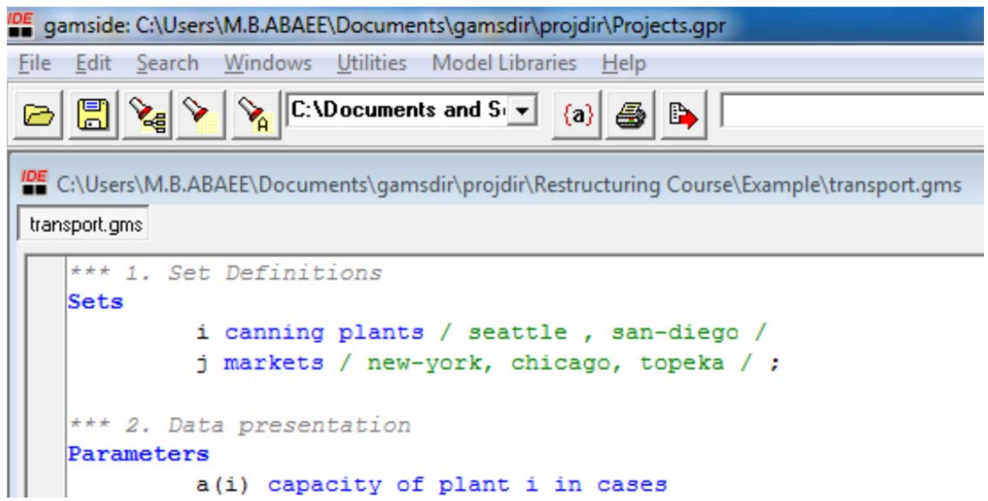
Email: m.b.abaee@gmail.com

1. INTRODUCTION

1.2. Using GAMSIDE …

1.2.6. Prepare the file so you think it is ready for execution

When using model library transport.gms should now appear as part of your IDE screen.



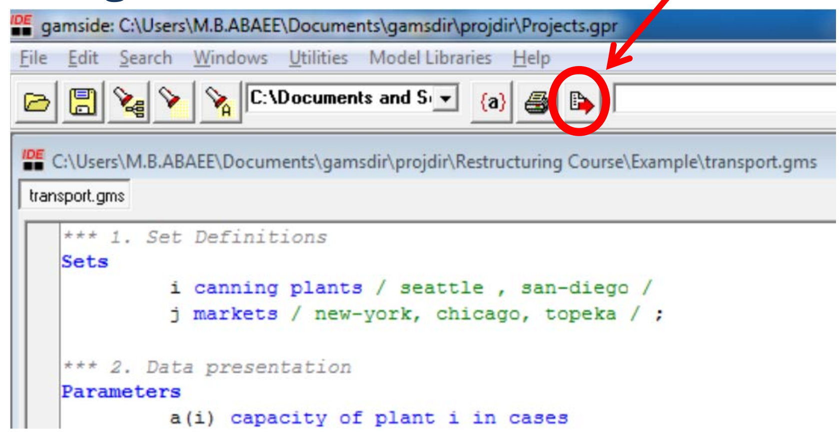
The IDE contains a full featured editor. Go through the file and change what you want.

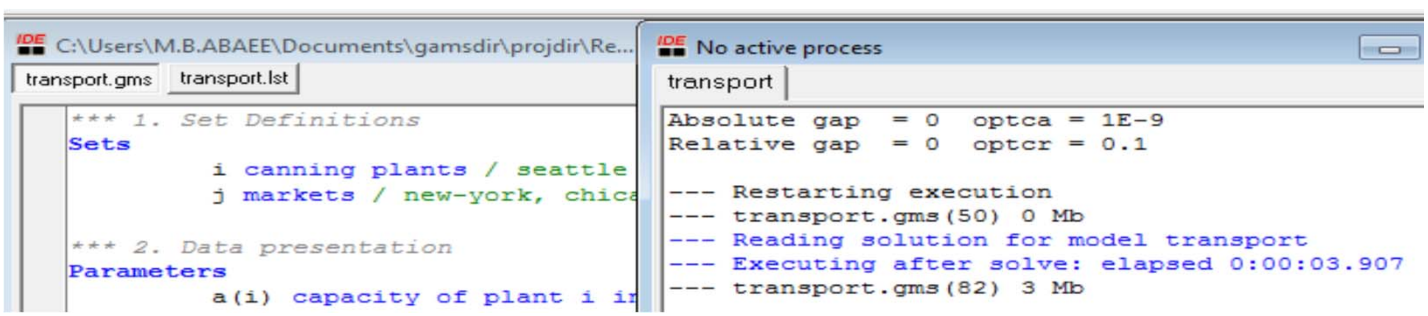
1. INTRODUCTION

1.2. Using GAMSIDE …

1.2.7. Run the file with GAMS by punching the run button

The so called process window will then appear which gives a log of the steps GAMS goes through in running the model and your model will run.



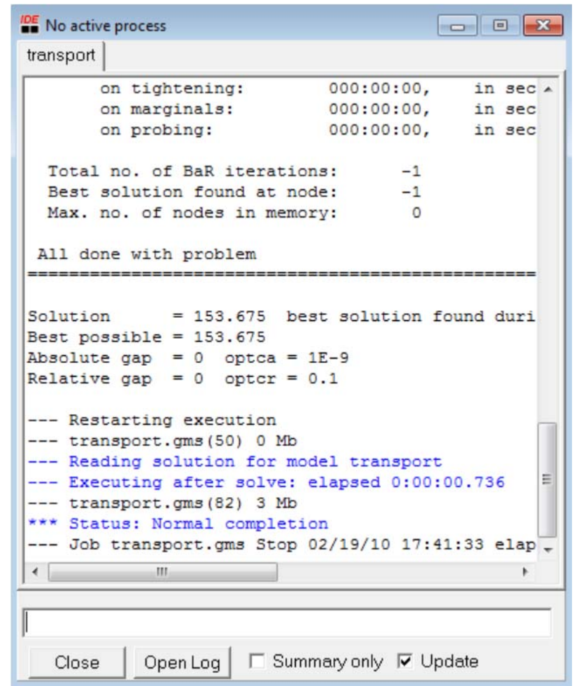


1. INTRODUCTION

1.2. Using GAMSIDE …

1.2.8. Open and navigate around the output

By double clicking on lines in the process window you can access program output both in general and at particular locations. The positioning of your access is determined by the color of the line you click on.



Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

1. INTRODUCTION

1.2. Using GAMSIDE …

Color of Line in Process Window	Function and Destination When Double Clicked
Blue line	Opens LST file and jumps to line in LST file corresponding to bolded line in Process file
Non bolded black line	Opens LST file and jumps to location of nearest Bolded Line
Red line	Identifies errors in source file. Cursor Jumps to Source (GMS) file location of error. Error description text in process window and in LST file which is not automatically opened.

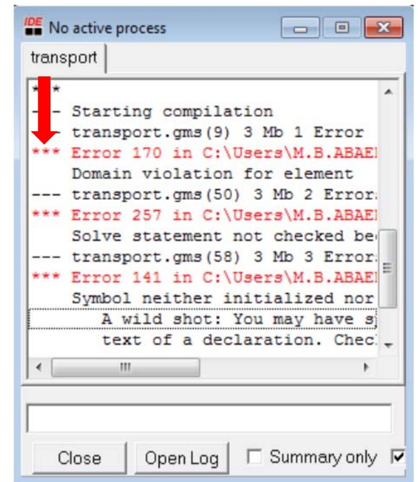
Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

1. INTRODUCTION

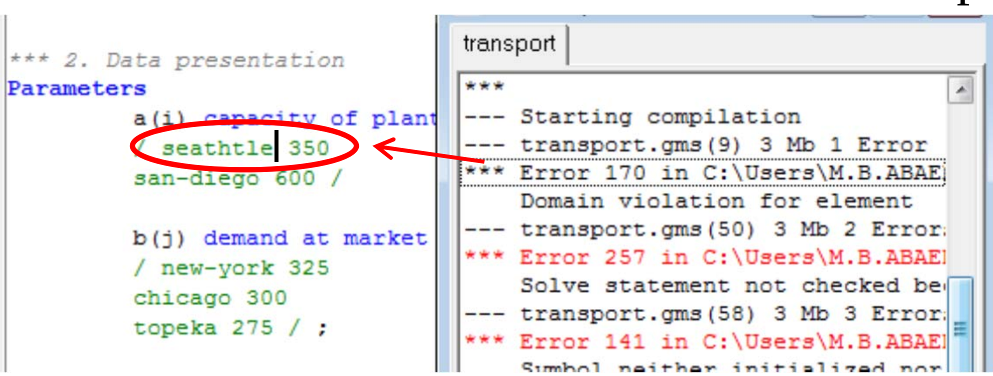
1.2. Using GAMSIDE …

1.2.9. Fixing Compilation Errors



No one is perfect, errors occur in everyone's GAMS coding. The IDE can help you in finding and fixing those errors.

The red lines mark errors. To see where the errors occurred double-click on the top one.



A double-click takes you to the place in the source where the error was made.

The tip here is always start at the top of the process file when doing this.

Courtesy of Mohammad Bagher Abaee , Spring 2010

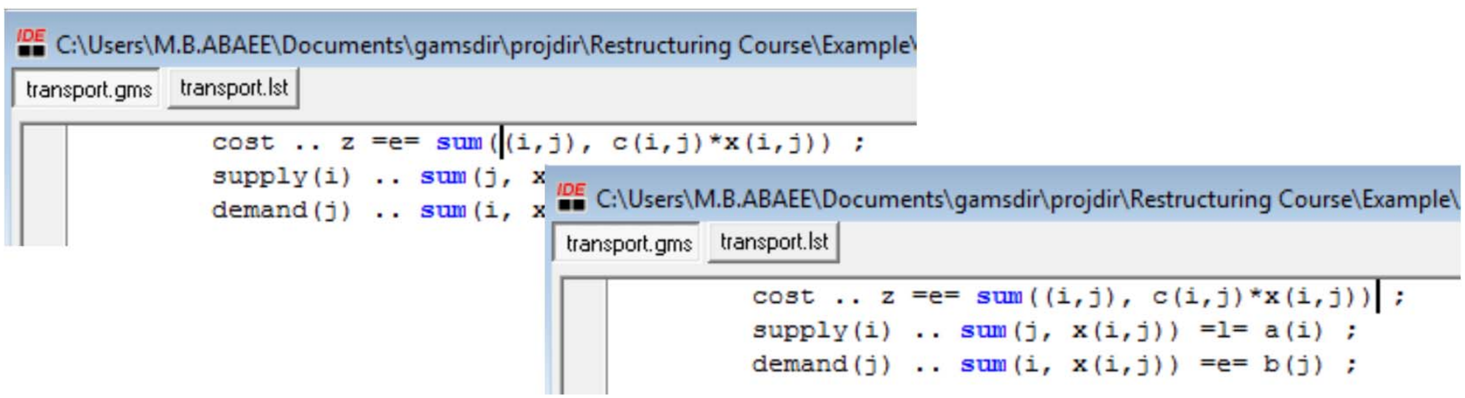
Email: m.b.abaee@gmail.com

1. INTRODUCTION

1.2. Using GAMSIDE …

1.2.10. Matching Parentheses

The IDE provides you with a way of checking on how the parentheses match up in your GAMS code. This involves usage of the symbol {a} from the menu bar coupled with appropriate cursor positioning. Suppose we have a line of GAMS code like:

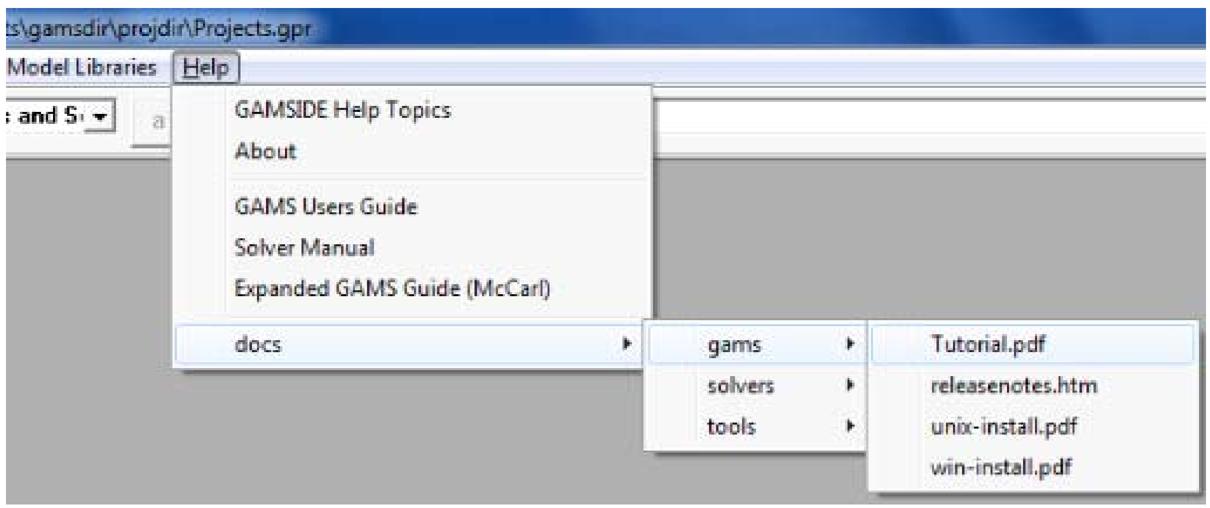


1. INTRODUCTION

1.2. Using GAMSIDE …

1.2.11. Accessing documentation on GAMS through the IDE

The GAMSIDE has a tie in to documentation. In particular suppose we wish to know about a particular item and there happens to be a file on that item.



2. INTRODUCTORY INFORMATION

2.1. Model Structure

The parts of a GAMS Model or Program are defined in the table below:

1. SETS	▪ Structures consisting of a complex of indices or names
2. DATA	▪ PARAMETERS, TABLES, SCALARS ▪ Determination of values of input parameters
3. VARIABLES	▪ Variables or arrays of variables ▪ Declaration with assigning a type of variable ▪ Declaration of limits for possible changes, initial level
4. EQUATIONS	▪ Equations or complexes and arrays of equations ▪ Declaration with assigning a name ▪ Recording of equations in the GAMS language
5. MODEL, SOLVE	▪ Model and methods of solution
6. OUTPUT	▪ Output of information into a separate file

Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.2. Comments in Models

There are two main methods of introducing comments and elucidation's in the body of a GAMS model.

a) If a line begins with the “ * ”symbol, the contents of the line are treated as a comment. For example:

```
*plants (plants)
```


Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.2. Comments in Models …

b) A program section beginning with the line $ONTEXT and ending with the line $OFFTEXT is a comment. The above functional words begin with the first symbol of the line.

```
$ONTEXT
– – – – – – – – – –
This guide is for a scientist who knows the
Persian language.
+ + + + + + + + + +
$OFFTEXT
```

2. INTRODUCTORY INFORMATION

2.3. Terms, Symbols and Reserved Words

2.3.1. Characters

A to Z	Alphabet	a to z	Alphabet	0 to 9	numerals
&	Ampersand	"	double quote	#	pound sign
*	multiply	=	equal	?	question mark
@	at	>	greater than	;	semicolon
\	back slash	<	less than	‘	single quote
:	colon	-	minus	/	slash
,	comma	( )	parenthesis		space
$	dollar	[ ]	square brackets	_	underscore
.	dot	{ }	braces	!	exclamation mark
+	plus	%	percent	^	Circumflex

2. INTRODUCTORY INFORMATION

2.3. Terms, Symbols and Reserved Words

2.3.2. Reserved Words

Abort	acronym	Acronyms	Alias	all	And
assign	binary	Card	Display	eps	Eq
equation	equations	Ge	Gt	Inf	Integer
Le	Loop	Lt	Maximizing	Minimizing	Model
models	Na	Ne		Negative	Not
options	Or	Ord	Parameter	Sets	Positive
Prod	Scalar	Scalars	Set	System	Smax
Smin	Sos1	Sos2	Sum	Yes	Table
Using	variable	Variables	Or	Else	Repeat
Until	While	If	Then	put	Semicont
semiint	File	Files	Putpage		free
No	option	Solve	for		

2. INTRODUCTORY INFORMATION

2.3. Terms, Symbols and Reserved Words

2.3.3. Operators

Premiership	Operator	Example	
(1)	**	x ** y ($x > 0$)	x^y
(2)	*	x * y	$x \times y$
(3)	/	x / y	$\frac{x}{y}$
(4)	- , +	$x \pm y$	$x \pm y$

Courtesy of Mohammad Bagher Abaee , Spring 2010
Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.4. Common mathematical functions

2.4.1. Abs

Expressions can contain a function that calculates the absolute value of an expression or term.

```
X=abs(t);
X=abs(y+2);
Eq1..  z=e=abs(yy);
```

$$X = |t|$$
$$X = |y + 2|$$
$$z = |yy|$$

It's use in .. equations in terms that involve variables requires the model type be NLP.

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.4. Common mathematical functions …

2.4.2. Exp

Expressions can contain a function that calculates the exponentiation e^q of an expression or term.

```
X=exp(t);
X=exp(y+2);
Eq2..  z=e=exp(yy);
```

$X = e^t$
$X = e^{y+2}$
$z = e^{yy}$

It's use in .. equations in terms that involve variables requires the model type be NLP.

2.4. Common mathematical functions …

2.4.3. Log, Log10

Expressions can contain a function that calculates the natural logarithm or logarithm base 10 of an expression or term.

```
X=log(t);
X=log(y+2);
Eq3..z=e=log(yy);
X=log10(tt);
X=log10(y+2);
Eq4..z=e=log10(yy);
```

$$X = Ln(t)$$
$$X = Ln(y+2)$$
$$z = Ln(yy)$$
$$X = \log_{10}(tt)$$
$$X = \log_{10}(y+2)$$
$$z = \log_{10}(yy)$$

It's use in .. equations in terms that involve variables requires the model type be NLP.

Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.4. Common mathematical functions …

2.4.4. Max , Min

Expressions can contain a function that calculates the maximum or minimum of a set of expressions or term.

```
X=min(y+2,t,r);
Eq..  z=e=max(yy,t);
```

$$X = min\{y + 2, t, r\}$$
$$z = min\{yy, t\}$$

It's use in .. equations in terms that involve variables requires the model type be DNLP.

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.4. Common mathematical functions …

2.4.5. Prod

Expressions can contain a function that calculates the product of set indexed expressions or terms.

```
X=prod(i,  a(i)*2);
```

$$X = \prod_{i \in I} \left(2 * a(i)\right)$$

It's use in .. equations in terms that involve variables requires the model type be NLP.

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.4. Common mathematical functions …

2.4.6. Round

Data calculation expressions can contain a function that rounds the numerical result of an expression or term. There are 2 variants of the rounding function.

The first, Rounds the result to the nearest integer value.

```
X=round(12.432);
```

$X = 12$

The second, rounds the result to the number of decimal points specified by the second argument.

```
X=round(12.432,2);
```

$X = 12.430$

This function may be used on data during GAMS calculations. It cannot be used in models

2. INTRODUCTORY INFORMATION

2.4. Common mathematical functions …

2.4.7. Smin , Smax

Expressions can contain a function that calculates the minimum or maximum of set indexed expression or term.

```
X=Smin(I,  a(i));
Eq..z=e=Smax((i,j),v(i,j));
```

$$X = min\{a(i)\}_{i=1}^{n}$$
$$z = max\{V(i,j)\}_{i,j=1}^{n}$$

It's use in .. equations in terms that involve variables requires the model type be DNLP.

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.4. Common mathematical functions …

2.4.8. Sqr

Expressions can contain a function that calculates the square of an expression or term.

```
X= sqr (y+2);
Eq10.. z=e= sqr (yy);
```

$$X = (y + 2)^2$$
$$z = (yy)^2$$

It's use in .. equations in terms that involve variables requires the model type be NLP.

Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.4. Common mathematical functions …

2.4.9. Sqrt

Expressions can contain a function that calculates the square root of an expression or term.

```
X= sqrt (y+2);
Eq10.. z=e= sqrt (yy);
```

$$X = \sqrt{y + 2}$$
$$z = \sqrt{yy}$$

It's use in .. equations in terms that involve variables requires the model type be NLP.

2. INTRODUCTORY INFORMATION

2.4. Common mathematical functions …

2.4.11. Other Mathematical functions

Function	Description	Use on Variables in Models
ArcCos (x)	Arc cosine of the argument x	Allowed but requires NLP model
ArcSin (x)	Arc sine of the argument x	Allowed but requires NLP model
ArcTan (x)	Arc tangent of the argument	Allowed but requires NLP model
Cos(x)-Sin(x)	Cosine-Sinose of the argument x where x must be in radians	Allowed but requires NLP model, can be discontinuous
Cosh (x)	Hyperbolic cosine of the argument x	Allowed but requires NLP model, can be discontinuous
Edist(x,y,z,…)	Sum of squared values calculation = $x^2 + y^2 + z^2 + \ldots$	Allowed but requires NLP model
Fact(x)	Factorial of x where x is an integer	Not allowed

Courtesy of Mohammad Bagher Abaee , Spring 2010 Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.4. Common mathematical functions …

2.4.10. Sum

Expressions can contain a function that calculates the sum of set indexed expressions or term.

```
X=Sum(i,  a(i));

Eq..z=e=Sum((i,j),B(i,j));
```

$$X = \sum_{i \in I} a(i)$$

$$Z = \sum_{i \in I} \sum_{j \in J} B(i,j)$$

This function is linear. It can be used on data during GAMS calculation or in models on variables or parameters.

2. INTRODUCTORY INFORMATION

2.4. Common mathematical functions …

2.4.11. Other Mathematical functions …

Function	Description	Use on Variables in Models
Floor (x)	Largest integer that is less than or equal to x	Not allowed
Normal(x,y)	Random number normally distributed with mean x and standard deviation y	Not allowed
Uniform(x,y)	Random number with uniform distribution between x and y	Not allowed
Uniformint(x,y)	Integer random number with uniform distribution between x and y	Not allowed
Pi	Value of Pi 3.141716…	Allowed
Power(x,y)	x raised to an integer power. x^y, where y must be an integer	Allowed but requires NLP model
Sign(x)	Sign of x. Returns 1 if x > 0, -1 if x < 0, and 0 if x = 0	Allowed but requires DNLP model

2. INTRODUCTORY INFORMATION

2.5. SET

2.5.1. Set Naming and Declaration

- SETS are the equivalent of indices in a typical programming language.
- In GAMS, indices have names, written through a combination of letters and digits, without spaces.
- A set name must begin with a letter, but the next symbol can be a letter, digit or the marks "+" and "-". For example:

Jlobest	1999	1972-138
GenCo	D6H83	Navruz-99

2. INTRODUCTORY INFORMATION

2.5. SET …

2.5.1. Set Naming and Declaration …

- The set declaration contains:
  - the set name
  - a list of elements in the set (up to 63 characters long spaces etc allowed in quotes)
  - optional labels describing the whole set
  - optional labels defining individual set elements

- The general format for a set statement is:

  SET setname optional defining text
  / firstsetelementname optional defining text
  secondsetelementname optional defining text
  ... /;

2. INTRODUCTORY INFORMATION

2.5. SET …

2.5.1. Set Naming and Declaration …

```
SET PROCESS PRODUCTION PROCESSES /X1,X2,X3/;
SET Consumers Cities /
            Tokyo   City1
            London  "$ City2"
            Moscow  /
    Year    /1990*2010/
    b       /a1b*a20b/
    a       /a1a*a20b/
 ;
```


Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.5. SET …

2.5.2. Subsets

$$Bus: I = \{1,2,3,\ldots,10\}$$
$$Generation\ Bus: G = \{2,5,9,10\}$$
$$\Rightarrow G \subset I$$

```
SET i     Bus /1*10/
    g(i) Generation Bus/2,5,9,10/;
```


Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.5. SET …

2.5.3. Multi-dimensional Sets

- GAMS allows sets with up to 20 dimensions.

```
SET     i       /a, b/
        j       /c, d, e/
    ij1(i,j) /a.c, a.d/ or /a.(c,d)/
    ij2(i,j) /a.c, b.c/
    ij3(i,j) /a.c, b.c, a.d, b.d/or/(a,b).(c,d)/
   ;
```

2. INTRODUCTORY INFORMATION

2.5. SET …

2.5.4. The Alias Statement: Multiple Names for a Set

There are occasions when one may wish to address a single set more than once in a statement. In GAMS this is done by giving the set another name through the ALIAS command as follows:

```
ALIAS(knownset,newset1,newset2,...);
```

```
SET      i  Bus /1*10/ ;
ALIAS(i, j, k);
```


Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.5. SET …

2.5.5. SET Operations

Operations can be performed on sets using the symbols:

```
+        -        *        Not
```

- Symbol " + " performs the set union operation

```
s3(n) = s1(n) + s2(n);
```

$$s3(n) = s1(n) \cup s2(n)$$

- Symbol " - " performs the operation of difference of sets. This set consists of elements, which belong to set A but not set B

```
s3(n) = s1(n) - s2(n);
```

$$s3(n) = s1(n) - (s1(n) \cap s2(n))$$

2. INTRODUCTORY INFORMATION

2.5. SET …

2.5.5. SET Operations …

- Operator " * " performs the set intersection operation; only the elements included in both set A and set B belong to the intersection of the sets A and B

```
s3(n) = s1(n) * s2(n);
s3(n) = s1(n) And s2(n);
```

$$s3(n) \;=\; s1(n) \cap s2(n)$$

- The symbol " **Not**" performs the set complement operation

```
s3(n) = Not s1(n);
```

$$s3(n) = N - \; s1(n)$$

Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.6. DATA

Digital data are contained in arrays (zero, scalar, or multidimensional matrices called parameters in GAMS). The SETs, described in section 2.5, can play the role of indices for these arrays. To declare an array to contain data values GAMS provides for three forms:

- SCALAR (zero-dimensional)
- PARAMETER (one-dimensional)
- TABLE (multidimensional)

Scalar entry is for scalars, Parameter generally for vectors and Table for matrices.

2. INTRODUCTORY INFORMATION

2.6. DATA …

2.6.1. SCALAR

- A scalar is regarded as a parameter that has no domain.
- Should define value for SCALAR.
- SCALAR format is used to enter items that are not defined with respect to SETs.
- The SCALAR declaration contains:
  - the set itemname
  - numerical value
  - optional labeling text
- The general format for a set statement is:

  SCALAR itemname optional labeling text / NumericalValue /**;**

2. INTRODUCTORY INFORMATION

2.6. DATA …

2.6.1. SCALAR …

```
SCALAR V / 0.005 /;
SCALAR Price Energy Price($/MWh)  /40/
       Load /350/;
```

$$V = 0.005$$
$$Price = 40$$
$$Load = 350$$

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.6. DATA …

2.6.2. PARAMETER

- Generally parameter format is used with data items that are one-dimensional (vectors) although multidimensional cases can be entered.
- Zero is the default value for all parameters. Therefore, you only need to include the nonzero entries in the element-value list, and these can be entered in any order.
- PARAMETER format is used to enter items defined with respect to SETs.

2. INTRODUCTORY INFORMATION

2.6. DATA …

2.6.2. PARAMETER …

- The PARAMETER declaration contains:
  - the set itemname
  - Set dependency
  - optional text
  - list of elements in the set
  - associated value
- The general format for a set statement is:

  PARAMETER itemname (setdependency) optional text
  / firstsetelementname associatedvalue
  secondsetelementname associatedvalue
  … /;

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.6. DATA …

2.6.2. PARAMETER …

```
SET t / H1*H3 /;
PARAMETER Tem(t) temperature
          /H1   3
           H2
           H3   -2.5
          /
;
```

$$Temperature(H1) = +3$$
$$Temperature(H2) = 0$$
$$Temperature(H3) = -2.5$$

2. INTRODUCTORY INFORMATION

2.6. DATA …

2.6.2. PARAMETER …

```
SET t        / H1*H3 /;
SET day      / day1,day2 /;
PARAMETER Tem1(t, day)
            /H1.day1   3
             H1.day2   3.5
             H3.day1   -1
             H3.day2   -2
            /
;
```

$$Temperature(H1, day1) = 3$$
$$Temperature(H1, day2) = 3.5$$
$$Temperature(H2, day1) = 0$$
$$Temperature(H2, day2) = 0$$
$$Temperature(H3, day1) = -1$$
$$Temperature(H3, day2) = -2$$

2. INTRODUCTORY INFORMATION

2.6. DATA …

2.6.3. TABLE

- Generally table format is used with data items that are multidimensional (matrix).
- Zero is the default value for all element of table. Therefore, you only need to include the nonzero entries in the element-value, and these can be entered in any order.
- Items in tables must be defined with respect to at least 2 sets and can be defined over up to 20 sets. When more than two dimensional items are entered, as in the equilibrium example, periods(.) set off the element names set1elementname.set2elementname.set3elementname etc.

2. INTRODUCTORY INFORMATION

2.6. DATA …

2.6.3. TABLE …

- The TABLE declaration contains:
  - the set itemname
  - Sets
  - descriptive text
  - associated value
- The general format for a set statement is:

```
TABLE  itemname (set1,set2,…)  descriptive text
                          | set_2_element_1  | set_2_element_2  …
       set_1_element_1    | value11          | value12
       set_1_element_2    | value21          | value22
       … ;
```

 Courtesy of Mohammad Bagher Abaee , Spring 2010 Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.6. DATA …

2.6.3. TABLE …

```
SET t       / H1*H3 /
    day     / day1,day2 /;
TABLE Tem1(t, day)
          /     day1  day2
           H1   3      3.5
           H2          0
           H3  -1      -2
          /
;
```

$$Temperature(H1, day1) = 3$$
$$Temperature(H1, day2) = 3.5$$
$$Temperature(H2, day1) = 0$$
$$Temperature(H2, day2) = 0$$
$$Temperature(H3, day1) = -1$$
$$Temperature(H3, day2) = -2$$

2. INTRODUCTORY INFORMATION

2.6. DATA …

2.6.3. TABLE …

```
SET t      / H1*H3 /
    day    / day1,day2 /
    Y      / Y1,Y2 /;
TABLE Tem2(t, day, Y) temperature
   day1.Y1   day2.Y1   day1.Y2   day2.Y2
H1  3         3.5       4         4.3
H2  1                             -2
H3  -1        -2        2
;
```

2. INTRODUCTORY INFORMATION

2.6. DATA …

2.6.3. TABLE …

```
SET t       / H1*H3 /
    day     / day1,day2 /
    Y       / Y1,Y2 /;
TABLE Tem2(t, day, Y) temperature
         Y1   Y2
H1.day1  3     4
H2.day1  1
H3.day1  -1   2
H1.day2  3.5 4.3
H2.day2      -2
H3.day2  -2
;
```

2. INTRODUCTORY INFORMATION

2.6. DATA …

2.6.4. Data Entry Through Computation

- After declaration, parameter values may be computed later in the model. Determination of data through direct computation or assignment following the initial definition is possible and sometimes useful.

2. INTRODUCTORY INFORMATION

2.6. DATA …

2.6.4. Data Entry Through Computation

```
PARAMETER   A
            B
            C(t)
            D(t,day)
            E(t,day,Y);

A = tem('H1');
C(t) = SUM(day,tem1(t,day));
D(t,day) = tem1(t,day) * abs(tem2(t,day,'Y1'));
E(t,day,Y) = 54 * tem2(t,day,Y);
```

2. INTRODUCTORY INFORMATION

2.7. VARIABLE

- The decision variables (or endogenous variables) of a GAMS-expressed model must be declared with a Variables statement.
- Named variables can be defined over from 0 up to 20 sets and thus one variable name may be associated with a single case or numerous individual variables, each associated with a specific simultaneous collection of set elements for each of the named sets.

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.7. VARIABLE

2.7.1. Variable Declaration

The general syntax for VARIABLE declaration is:

```
VARIABLE
        firstvariablename    optional text
                  /optional value for attribute/
        secondvarname (setdependency) optional text
                  /optional values for attributes/
                  ...
;
```


Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.7. VARIABLE …

2.7.2. Type of Defining Variables

Variable Type	Allowed Range of Variable
Free	$-\infty$ ***to*** $+\infty$
Positive	**0** ***to*** $+\infty$
Negative	$-\infty$ ***to*** **0**
Binary	**0** ***or*** **1**
Integer	0; 1; … ; 100 (default)

The variable that serves as the quantity to be optimized must be *a scalar* and must be of the *free type*.

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.7. VARIABLE …

2.7.3. Suffix of Variables

Suffix	Description
.L	Value (level)
.LO	Lower boundary
.UP	Upper boundary
.FX	Fixed value
.m	Dual value

Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.7. VARIABLE …

2.7.4. Example

```
VARIABLE
            obj         objective value
            P(i)        Product of Generator(i)
            Flow(i,j)   Line Flow
;
POSITIVE VARIABLE P(i);
BINARY   VARIABLE Rn(i,t);

Rn.fx(i,'H0') = 0;
P.up('2')     = 300;
```

2.8. EQUATION

2.8.1. Declaration of Equations

Declaration of equation names is similar to the declaration of SETs or PARAMETERs. The similarity is in the fact that a list and comments are allowed and recommended.

The general syntax for EQUATION declaration is:

```
EQUATION
        firstequationname                    optional text
        secondeqname (setdependency)    optional text
                  ... ;
```

 Courtesy of Mohammad Bagher Abaee , Spring 2010 Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.8. EQUATION …

2.8.1. Declaration of Equations …

- EQUATION is a keyword which must appear before the name of each equation if there is semicolon at the end of the line.
- Commas separate different names of equations in a row or one row is assigned for each name.
- Equations can be defined over from 0 up to 20 sets.
- The name can include two parts:
  - an *identifier* of the name of the equation, and
  - *indices* in **parentheses**.
- The identifier (name) of an equation may include no more than 10 symbols and must always begin with a letter.

2. INTRODUCTORY INFORMATION

2.8. EQUATION …

2.8.1. Declaration of Equations …

```
EQUATION Eq1(i), Eq2(i,j), Objective;

      OR

EQUATION
         Objective                 objective function
         Eq1(i)                    Equation1(i)
         Eq2(i,j)                  Equation2(i,j)
;
```

2. INTRODUCTORY INFORMATION

2.8. EQUATION …

2.8.2. Definition of Equations

The definition of equation structures is a mathematical peculiarity in the GAMS language.

The syntax for defining equations in GAMS is as follows :

equationname(setdependency)$optional logical condition ..
lhs_equation_terms equation_type rhs_equation_terms **;**

= **G** =	$\geq$	right part is less than or equal to the left hand side
= **E** =	$=$	right hand side is equal to the left hand side
= **L** =	$\leq$	right hand side is greater than or equal to the left hand side

Courtesy of Mohammad Bagher Abaee , Spring 2010 Email: m.b.abaee@gmail.com

2.8. EQUATION …

2.8.2. Definition of Equations …

```
EQUATION Eq1(i), Eq2(i,j), Objective
;

Objective.. obj =E= (k1-k2)*(k1-k2);
Eq1(i)    .. Y1(i) + SUM(j,Y2(i,j))=E=5*x(i)*x(i);
Eq2(i,j) ..  Y2(i,j) =G= -10*x(i)+100*q(i,j);
```

2. INTRODUCTORY INFORMATION

2.9. MODEL, SOLVE

2.9.1. Assembling a Model

- Models are objects that GAMS solves.
- They are collections of the specified equations and contain variables along with the upper and lower bound attributes of the variables.
- The basic form of the model statement is:

MODEL ModelName1 Optional Text /**Model Contents** /
ModelName2 Optional Text /**Model Contents** /**;**

```
MODEL   One   first model    /all/
        Two   second model  /obj,Eq1/
        Three third model   /Two,Eq2/;
```

 Email: m.b.abaee@gmail.com

2. INTRODUCTORY INFORMATION

2.9. MODEL, SOLVE …

2.9.2. Solving a Model

- Once a model has been defined, the next step is to solve it.
- This is ordered by including a SOLVE statement in the model.
- The format of the solve statement is as follows:

  1. The key word **SOLVE**
  2. The name of the **model** to be solved
  3. The key word **using**
  4. An available **solution procedure**.
  5. The keyword **'minimizing'** or **'maximizing'**
  6. The name of the **variable** to be optimized

SOLVE One using LP minimizing obj;

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

2.9. MODEL, SOLVE …

2.9.3. Solution Procedure

Solution Procedure	Description
LP	Linear programming. The model cannot contain nonlinear or discrete (binary and integer) variables.
NLP	Nonlinear programming. In the model, nonlinear forms must be continuous functions and the model may not contain discrete variables.
MIP	Mixed integer programming. Similar to RMIP, but the requirements of discreteness of variables and equations are stringent. Discrete variables must take discrete values within boundaries.
MINLP	Mixed integer nonlinear programming. The same characteristics as for RMINLP, but the requirements of discreteness are very stringent.
DNLP	for nonlinear programming with discontinuous derivatives

2. INTRODUCTORY INFORMATION

2.10. OUTPUT

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features

3.1.1. The Loop Statement

The Loop statement allows one to execute a group of statements for each element of a set.

The syntax of the Loop statement is:

```
LOOP ( (set1,set2,…) [$(condition)],
          Statement or Statements to Execute;
       );
```

One cannot place GAMS set, acronym, for, scalar, parameter, table, variable, equation or model statements inside the Loop statement.

Email: m.b.abaee@gmail.com

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features …

3.1.1. The Loop Statement …

```
SET     t       / 1985*1990 /;

PARAMETERS pop(t)    / 1985 3456 /
           growth(t) / 1985 25.3, 1986 27.3
                       1987 26.2, 1988 27.1
                       1989 26.6, 1990 26.6/;
```

```
LOOP(t,
      pop(t+1) = pop(t) + growth(t);
     );
```


Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features …

3.1.2. The If-Elseif-Else Statement

The IF-ELSE operator is useful for transferring from one operator to another. In some cases, it can be written down as a set of $ conditions.

The following syntax is for the "IF-THEN-ELSE" operator:

```
IF ( Condition,
          Statements;
   {ElseIf  Condition ,
          Statements; }
   [Else
          Statements;]
  );
```

One cannot place GAMS set, acronym, for, scalar, parameter, table, variable, equation or model statements inside the If statement.

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features …

3.1.2. The If-Elseif-Else Statement …

```
IF (f <= 0,
        p(i) = -1 ;
        q(j) = -1 ;
ElseIf ((f > 0) and (f < 1)),
        p(i) = p(i)**2 ;
        q(j) = q(j)**2 ;
Else
        p(i) = p(i)**3 ;
        q(j) = q(j)**3 ;
    );
```

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features …

3.1.3. The While Statement

The While statement allows one to repeatedly execute a block of statements until a logical condition is satisfied.

Ordinarily, the syntax of the While statement is:

```
WHILE (Condition,
        Statement or Statements to Execute;
      );
```

One cannot place GAMS set, acronym, for, scalar, parameter, table, variable, equation or model statements inside the While statement.

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features …

3.1.3. The While Statement …

```
WHILE (x<10,
       x=x+0.01;
);

            OR

$Onend
WHILE x<10 do
      x=x+0.01;
EndWhile;
```

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features …

3.1.4. The For Statement

The For statement allows one to repeatedly execute a block of statements over a successively varied values of a scalar.

The syntax of the For statement is:

FOR (Scalarq = Startval To (Downto) Endval by Increment,
Statement or Statements to Execute**;**
);

- Scalarq: is a scalar
- Startval: is the constant or scalar giving the value where scalarq will begin
- To: indicates that GAMS will add increment until scalarq gets equal to or larger than end

 Courtesy of Mohammad Bagher Abaee , Spring 2010 Email: m.b.abaee@gmail.com

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features …

3.1.4. The For Statement …

The syntax of the For statement is:

FOR (Scalarq = Startval To (Downto) Endval by Increment,
Statement or Statements to Execute**;**
);

- Downto: indicates that GAMS will subtract increment until scalarq gets equal to or smaller than end.
- Endval: is a constant or scalar giving the value that will result in statement termination when scalarq equals or passes it.
- Increment: is a positive constant or scalar which is optional and defaults to one.

One cannot place GAMS set, acronym, for, scalar, parameter, table, variable, equation or model statements inside the For statement.

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features …

3.1.4. The For Statement …

```
FOR (x=12 downto 1 by 2,
      data(i)=x;
);

           OR

$Onend
FOR x=12 downto 1 by 2 do
      data(i)=x;
EndFor;
```

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features …

3.1.5. The Repeat Statement

The Repeat statement causes one to execute a block of statements over and over until a logical condition is satisfied.

The syntax of the Repeat statement is:

```
REPEAT (
        Statement or Statements to Execute;
UNTIL logical condition is true );
```

One cannot place GAMS set, acronym, for, scalar, parameter, table, variable, equation or model statements inside the Repeat statement.

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features …

3.1.5. The Repeat Statement …

```
SCALAR x /5/;

REPEAT(
       x = x + 0.01;
      UNTIL  x > 10 );
```

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features …

3.1.6. Numerical Relationship Operators

Operator		Meaning
Lt	<	Strictly less than
Le	<=	Less than-or-equal to
Eq	=	Equal to
Ne	< >	Not equal to
Ge	>=	Greater than or equal to
Gt	>	Strictly greater than

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features …

3.1.7. Logical Operators

Operator	Meaning
Not	Not
And	And
Or	Inclusive or
Xor	Exclusive or $(A.\bar{B} + \bar{A}.B)$

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features …

3.1.7. Logical Operators …

Operands		*Results*			
a	***b***	***a and b***	***a or b***	***a xor b***	***not a***
0	**0**	**0**	**0**	**0**	**1**
0	**non-zero**	**0**	**1**	**1**	**1**
non-zero	**0**	**0**	**1**	**1**	**0**
non-zero	**non-zero**	**1**	**1**	**0**	**0**

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

3. SPECIAL ORDERS GAMS

3.1. Programming Flow Control Features …

3.1.7. Mixed Logical Conditions

Logical Condition	*Numerical Value*	*Logical Value*
$(1 < 2) + (3 < 4)$	**2**	***True***
$(2 < 1)\ and\ (3 < 4)$	**0**	***False***
$(4 * 5 - 3) + (10/8)$	**17.125**	***True***
$(4 * 5 - 3)\ or\ (10 - 8)$	**1**	***True***
$(4\ and\ 5) + (2 * 3 <= 6)$	**2**	***True***
$(4\ and\ 0) + (2 * 3 < 6)$	**0**	***False***

Email: m.b.abaee@gmail.com

3. SPECIAL ORDERS GAMS

3.2. Ordered Sets

3.2.1. ORD Operator

The ORD returns an ordinal number equal to the index position in a set.

```
SET          t    time periods / 2000*2010 /;
PARAMETER   val(t);
val(t) = ORD(t);
```

$$val(2000) = 1$$
$$val(2001) = 2$$
$$\dots$$
$$val(2010) = 11$$

Email: m.b.abaee@gmail.com

3. SPECIAL ORDERS GAMS

3.2. Ordered Sets …

3.2.2. CARD Operator

The CARD operator returns an ordinal number equal to the number of elements in a set.

```
SET            t    time periods / 2000*2010 /;
PARAMETER   S;
S = CARD(t);
```

$$\boxed{S = 11}$$

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

3. SPECIAL ORDERS GAMS

3.2. Ordered Sets …

3.2.3. SAMEAS Operator

One may wish to do conditional processing dependent upon the text defining a name of a set element matching the text for a particular text string or matching up with the text for a name of a set element inset.

SAMEAS(SetElement, OtherSetElement) or **SAMEAS(aSetElement, "text")** returns an indicator that is true if the text giving the name of **Etelement** is the same as the text for **OtherSetElement** and a false otherwise.

3. SPECIAL ORDERS GAMS

3.2. Ordered Sets …

3.2.3. SAMEAS Operator …

```
SET    cityI / "new york", Chicago, boston/;
SET    cityJ /boston/;

SCALAR A ;

A = SUM(SAMEAS(cityI, cityJ),1);

Display A;
```

$$A = 1$$

3. SPECIAL ORDERS GAMS

3.2. Ordered Sets …

3.2.4. DIAG Operator

DIAG(setelement, othersetelement) or **diag(asetelement, "text")** returns a number that is one if the text giving the name of **setelement** is the same as the text for **othersetelement** and a zero otherwise.

```
SET    cityI / "new york", Chicago, boston/;
SET    cityJ /boston/;
SCALAR A ;
A = SUM((cityI, cityJ), DIAG(cityI, cityJ));
Display A;
```

$$\boxed{A = 1}$$

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

3. SPECIAL ORDERS GAMS

3.2. Ordered Sets …

3.2.5. Lag and Lead Operator

Operators LAG and LEAD are used for correlating an element of a set with the next or preceding element of the set. GAMS has two variants of LAG and LEAD operators:

- Linear LAG and LEAD operators (+, -)
- Circular LAG and LEAD operators (++, --).

The difference between these two types of operators is in the method of processing at the initial and final points of the sequence. In circular operators, after the final element of the sequence, there comes the first, while in linear operators sets are broken.

Email: m.b.abaee@gmail.com

3. SPECIAL ORDERS GAMS

3.2. Ordered Sets …

3.2.5. Lag and Lead Operator …

```
SET   t /t1*t3/;

VARIABLE a1(t),a2(t) ;

EQUATION Eq1(t),Eq2(t),Eq3(t),Eq4(t) ;

Eq1(t) ..  a1(t)    =e=  a2(t+1);
Eq2(t) ..  a1(t-1)  =e=  a2(t);
Eq3(t) ..  a1(t)    =e=  a2(t++1);
Eq4(t) ..  a1(t--1) =e=  a2(t);
```

3. SPECIAL ORDERS GAMS

3.2. Ordered Sets …

3.2.5. Lag and Lead Operator …

$$
\begin{aligned}
&Eq1: \\
&\quad a1(t1) = a2(t2); \\
&\quad a1(t2) = a2(t3); \\
&\quad a1(t3) = 0;
\end{aligned}
$$

$$
\begin{aligned}
&Eq2: \\
&\quad 0 = a2(t1); \\
&\quad a1(t1) = a2(t2); \\
&\quad a1(t2) = a2(t3);
\end{aligned}
$$

$$
\begin{aligned}
&Eq3: \\
&\quad a1(t1) = a2(t2); \\
&\quad a1(t2) = a2(t3); \\
&\quad a1(t3) = a2(t1);
\end{aligned}
$$

$$
\begin{aligned}
&Eq4: \\
&\quad a1(t3) = a2(t1); \\
&\quad a1(t1) = a2(t2); \\
&\quad a1(t2) = a2(t3);
\end{aligned}
$$

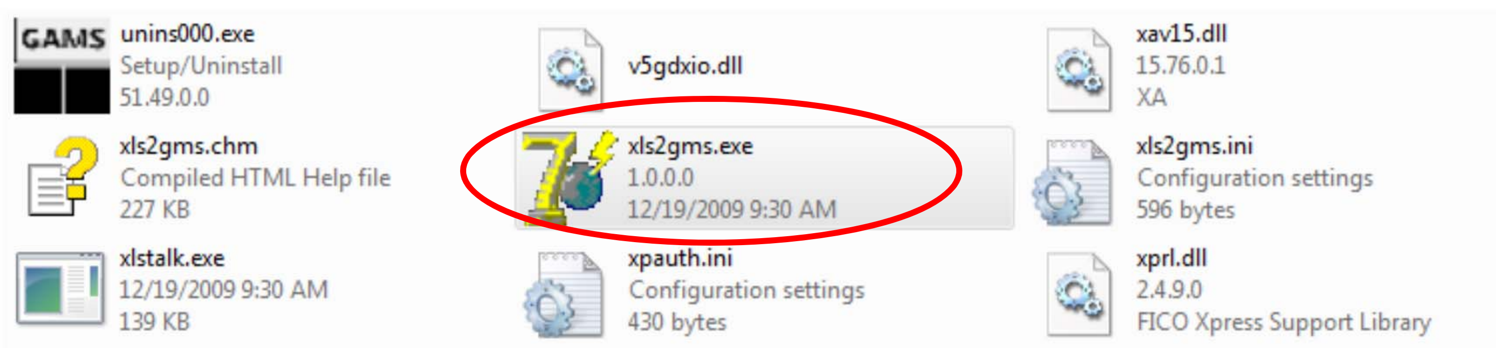
4. GAMS and Other Applications

4.1. GAMS/Excel

4.1.1. Converting Data to GAMS

From wherever your gams system directory is, run "XLS2GMS.exe". For GAMSIDE users, this may be:

C:\Program Files\GAMS

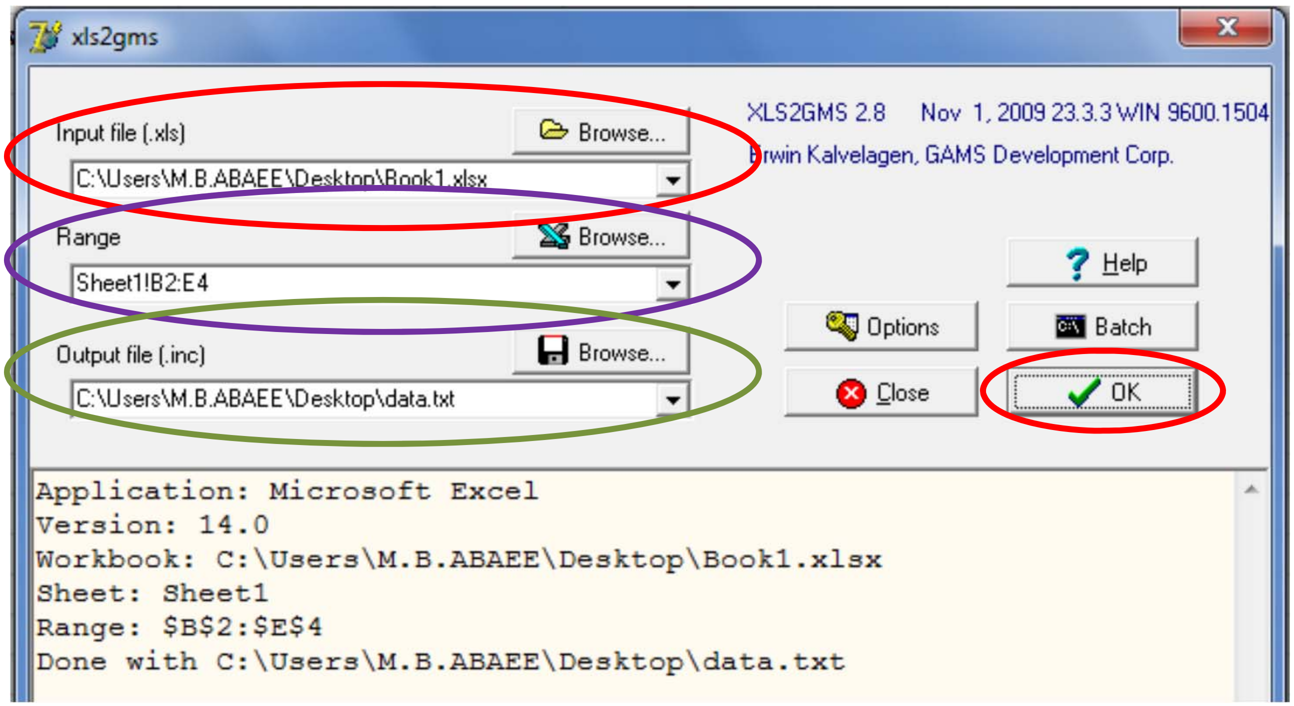


 Courtesy of Mohammad Bagher Abaee , Spring 2010 Email: m.b.abaee@gmail.com

4. GAMS and Other Applications

4.1. GAMS/Excel …

4.1.1. Converting Data to GAMS …



Input file (*.XLS)

Range

Output GAMS Include file (*.txt or *.inc)

Email: m.b.abaee@gmail.com

4. GAMS and Other Applications

4.1. GAMS/Excel …

4.1.1. Converting Data to GAMS …

```
* ------------------------------------------------------
* XLS2GMS 2.8      Nov  1, 2009 23.3.3 WIN 9600.15043 VIS x86/MS Windows
* Erwin Kalvelagen, GAMS Development Corp.
* ------------------------------------------------------
* Application: Microsoft Excel
* Version:     14.0
* Workbook:    C:\Users\M.B.ABAEE\Desktop\Book1.xlsx
* Sheet:       Sheet1
* Range:       $B$2:$E$4
* ------------------------------------------------------
          new-york chicago topeka
seattle   2.5      1.7     1.8
san-diego 2.5      1.8     1.4
* ------------------------------------------------------
```

```
TABLE d(i, j)
$include data.txt
;
```

4. GAMS and Other Applications

4.1. GAMS/Excel …

4.1.2. Import Data from Excel

When calling **XLS2GMS** directly from GAMS we want to specify all command and options directly from the command line or from a command file.

Command is:

$call = xls2gms I="C:\test.xls" O="C:\data.txt" R=Sheet1!B1:E10

- Command Statement
- I=inputfilename
- I=inputfilename
- Sheet of excel
- Range of sheet

4. GAMS and Other Applications

4.1. GAMS/Excel …

4.1.2. Import Data from Excel …

```
SET    i /
$call = xls2gms  I=Book1.xlsx O=setI.txt R=Sheet2!A1:A3
$include setI.txt
/
       j /
$call = xls2gms  I=Book1.xlsx O=setJ.txt R=Sheet2!B1:B3
$include setJ.txt
/;
TABLE  d(i, j)
$call = xls2gms  I=Book1.xlsx O=Data.txt R=Sheet1!B2:E4
$include data.txt
;
```

4. GAMS and Other Applications

4.2. GAMS/MATLAB

4.2.1. Installation

First of all, you need to install both MATLAB and GAMS on your machine. We will assume that the relevant system (installation) directories are :

C:\MATLAB and **C:\GAMS**

1. Copy **matout.gms** in **C:\GAMS\inclib\matout.gms**
2. Copy **gams.dll** in **C:\MATLAB\toolbox\local\gams.dll**
3. Set the environment variable PATH of GAMS

4. GAMS and Other Applications

4.2. GAMS/MATLAB …

4.2.1. Installation …

To test the installation, carry out the following steps:

1. Start up **MATLAB**
2. In the **MATLAB** command window, change directories to the examples directory provided as part of the distribution. (This directory contains at least two files, **testinst.m** and **testinst.gms** that are required for this test.)
3. Run the example "**testinst**" that is found in the examples directory of the distribution. At the **MATLAB** prompt you just type:

```
>> testinst
```

4. GAMS and Other Applications

4.2. GAMS/MATLAB …

4.2.1. Installation …

The resulting output will depend on the platform on which you run this from. It should include the output given below:

```
Q =
  name: 'Q'
   val: [3x3 double]
ans =
     1 0 0
     0 1 0
     0 0 1
Q =
    2 0 0
    0 2 0
    0 0 2
J =
   '1'
   '2'
   '3'
….
```

```
Q =
  name: 'Q'
   val: [3x3 double]
J =
  name: 'J'
   val: {3x1 cell}
Q =
  name: 'Q'
   val: [3x3 double]
J =
  name: 'J'
   val: {3x1 cell}
A =
  name: 'A'
   val: [2x3 double]
….
```

```
ans =
     3 0 0
     0 3 0
     0 0 3
ans =
     '1'
     '2'
     '3'
ans =
     0 2 -5
     2 0 2
```


Email: m.b.abaee@gmail.com

4. GAMS and Other Applications

4.2. GAMS/MATLAB …

4.2.2. Returning Values to MATLAB

In order to run the same model within **MATLAB** and return the solution vector variable of ***x(i)*** and parameter of ***d(i, j)*** back into the **MATLAB** workspace, one change is required to the GAMS file, namely to add the below line after the **SOLVE** statement:

```
$libinclude matout x.l i
$libinclude matout  d  i j
```

This just writes out the level values of the solution to a file that can be read back into **MATLAB**. In **MATLAB**, you just execute the following statement:

```
[x d] = gams('NameGamsFile');
```

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

7.1. Power Transit

Figure 7.1.1 shows a 6-Bus system. There are two Power Plants at Bus1 and Bus2 and three Steel Factory at Bus3, Bus4 and Bus5. Power Plants can contract bilateral.

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?

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

7.2. Economic Dispatch

To appreciate the advantages of dispatching a power system according to the solution of the ED problem, consider the case where a power plant supplies 10,000MW during 1 h at an average cost of \$0.05/kWh; if the consumers buy this energy at the rate of \$0.06/kWh, this arrangement results in a net profit to the supplier of \$100,000/h. In this case an improvement in supply efficiency of just 1% through the use of ED would result in a profit increment of \$5000/h or \$43.8 million in one year.

7.2. Economic Dispatch …

7.2.1. Basic Economic Dispatch

Consider two generating units supplying the system demand $\boldsymbol{P_D^{total}}$. The quadratic unit cost functions are characterized by the parameters provided in the table as follows:

Unit	$C_0(\$/h)$	$a(\$/MWh)$	$b[\$/(MW)^2h]$	$P_G^{min}(MW)$	$P_G^{max}(MW)$
G1	**100**	**20**	**0.05**	**0**	**400**
G2	**200**	**25**	**0.10**	**0**	**300**

$$\boldsymbol{C_i(P_{Gi}) = C_{0i} + a_i P_{Gi} + \frac{1}{2} b_i P_{Gi}^2}$$

7.2. Economic Dispatch …

7.2.1. Basic Economic Dispatch …

For the specific system demand levels of 40, 250, 300, and 600MW, calculate and analysis:

A. No generation limits

1) ED generation levels
2) Cost of generation each unit
3) Incremental cost

B. With generation limits

1) ED generation levels
2) Cost of generation each unit
3) Incremental cost

7.2. Economic Dispatch …

7.2.2. Economic Dispatch with Elastic Demand

The generating units of example 7.2.1 as well as the elastic demands characterized below are considered in this example:

Load	$a^D(\$/MWh)$	$b^D[\$/(MW)^2h]$	$P_{\boldsymbol{D}}^{min}(MW)$	$P_{\boldsymbol{D}}^{max}(MW)$
L1	**55**	**-0.2**	**0**	**300**
L2	**50**	**-0.1**	**0**	**350**

$$\boldsymbol{U_j(P_{Dj}) = a_j^D P_{Dj} + b_j^D P_{Dj}^2}$$

 Courtesy of Mohammad Bagher Abaee , Spring 2010 Email: m.b.abaee@gmail.com

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

7.2. Economic Dispatch …

7.2.2. Economic Dispatch with Elastic Demand …

Calculate and analysis the ED with elastic demand:

1) ED generation levels
2) ED load levels
3) Incremental cost
4) Social welfare
5) Cost of generation each unit
6) Utility of each load

7.2. Economic Dispatch …

7.2.3. Economic Dispatch with Losses

In the generating units of example 7.2.1 as well as consider sensitivity coefficients of the losses with respect to the generation levels. Sensitivity coefficients of the losses provided in the table as follows:

Unit	***l***
G1	**0.00047**
G2	**0.00023**

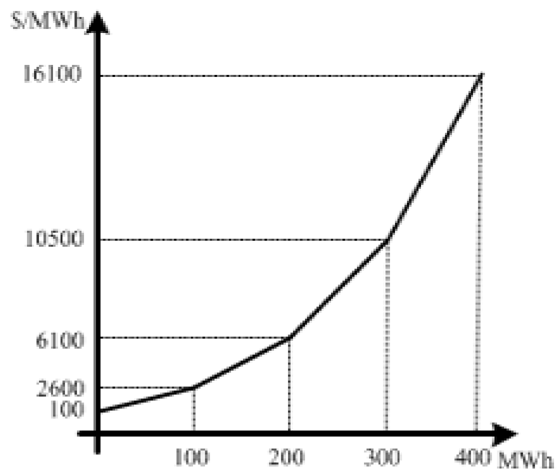
$$Loss = \sum_{i \in I} l_i P_{Gi}^2$$

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

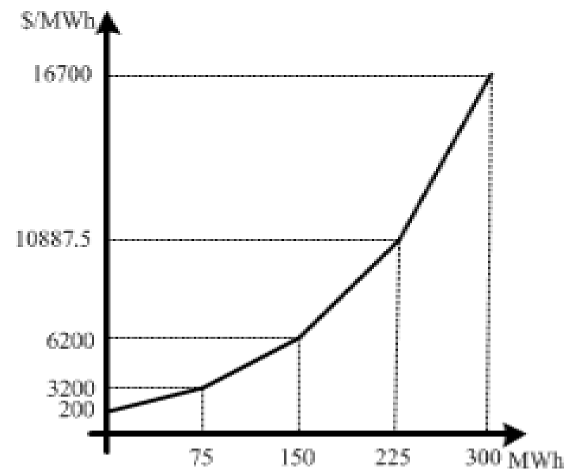
7.2. Economic Dispatch …

7.2.4. First Project …

In example 7.2.1, suppose that quadratic cost functions are piecewise linear cost functions. Piecewise linear functions are illustrated in below figures for both generators.



Cost(G1)



Cost(G2)

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

7.2. Economic Dispatch

7.2.4. First Project …

Now, calculate, analysis and compare with results of section 7.2.1 at the specific system demand levels of 40, 250, 300, and 600MW:

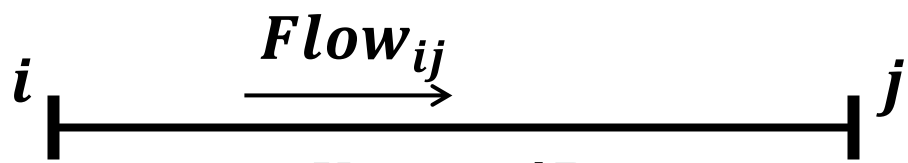
A. No generation limits
1) ED generation levels
2) Cost of generation each unit
3) Incremental cost

B. With generation limits
1) ED generation levels
2) Cost of generation each unit
3) Incremental cost

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

7.3. Optimal Power Flow

7.3.1. DC Load Power Formulation



$$Y_{ij} = -jB_{ij}$$

$$\begin{bmatrix} P_1 \\ \vdots \\ P_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{B_{11}} & \cdots & \boldsymbol{B_{1n}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{B_{n1}} & \cdots & \boldsymbol{B_{nn}} \end{bmatrix} \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_n \end{bmatrix}$$

$$\boldsymbol{Flow_{ij} = B_{ij}(\delta_i - \delta_j)}$$

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

7.3. Optimal Power Flow …

7.3.2. Optimal DC Power Flow Formulation

A)Quadratic Cost Function

$$Minimize_{(P_i,\delta_i)} \sum_{i=1}^{n} \left(C_{0i} + a_i P_{Gi} + \frac{1}{2} b_i P_{Gi}^2 \right)$$

Subject to:

$$(P_{Gi} - Load_i) = \sum_{j=1}^{n} B_{ij}\delta_j \qquad (i)$$

$$P_{Gi} \le P_{Gi}^{max}\ I_i \qquad (i)$$

$$P_{Gi} \ge P_{Gi}^{min}\ I_i \qquad (i)$$

$$Flow_{ij} = B_{ij}(\delta_i - \delta_j) \qquad (i,j)$$

$$-Flow_{ij}^{max} \le Flow_{ij} \le Flow_{ij}^{max} \qquad (i,j)$$

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

7.3. Optimal Power Flow …

7.3.2. Optimal DC Power Flow Formulation …

B)Step Cost Function

$$\boldsymbol{Minimize}_{(P_i,\delta_i)} \sum_{i=1}^{n}\sum_{k=1}^{m} P_{ik}\boldsymbol{C}_{ik}$$

Subject to:

$$\left(\sum_{k=1}^{m}(\boldsymbol{P}_{ik}) - \boldsymbol{Load}_i\right) = \sum_{j=1}^{n}\boldsymbol{B}_{ij}\boldsymbol{\delta}_j \qquad (\boldsymbol{i})$$

$$\boldsymbol{I}_i\boldsymbol{P}_i^{min} \le \sum_{k=1}^{m}\boldsymbol{P}_{ik} \le \boldsymbol{P}_i^{max}\boldsymbol{I}_i \qquad (\boldsymbol{i})$$

$$\boldsymbol{Flow}_{ij} = \boldsymbol{B}_{ij}(\boldsymbol{\delta}_i - \boldsymbol{\delta}_j) \qquad (\boldsymbol{i},\boldsymbol{j})$$

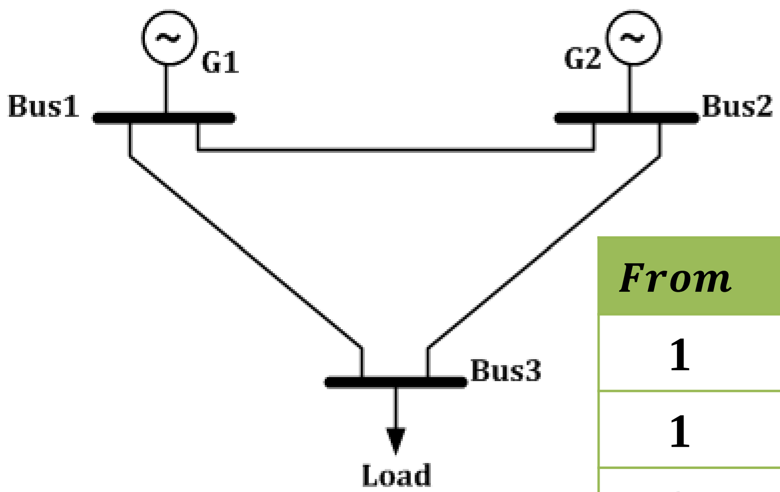
$$-\boldsymbol{Flow}_{ij}^{max} \le \boldsymbol{Flow}_{ij} \le \boldsymbol{Flow}_{ij}^{max} \qquad (\boldsymbol{i},\boldsymbol{j})$$

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

7.3. Optimal Power Flow …

7.3.3. Economic Dispatch Including a Line with Limited Transmission Capacity

Consider the three-bus, three-line network characterized in the table and depicted in below figure (shut susceptances neglected).



$From$	To	$Reactance\,(pu)$	$Flow^{max}\,(MW)$
1	**2**	**0.1**	**unlimited**
1	**3**	**0.1**	**140**
2	**3**	**0.1**	**unlimited**

7.3. Optimal Power Flow …

7.3.3. Economic Dispatch Including a Line with Limited Transmission …

Buses 1 and 2 include the two units characterized in Section 7.2.1. Voltage magnitudes are constant and equal to 1. A base of 100 kV and 200MVA is considered.

Now, calculate, analysis and compare with results of section 7.2.1 at the specific system demand levels of 250 and 300MW with and without transmission limit and generation limits in line 1–3:

1) ED generation levels
2) Cost of generation each unit
3) LMPs

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

7.3. Optimal Power Flow …

7.3.4. Economic Dispatch with Bilateral Contracts

Considering the units with the cost functions of Section 7.2.1, suppose that the generating unit 2 and the load engage in a bilateral contract of 50MW at $31.5/MWh.

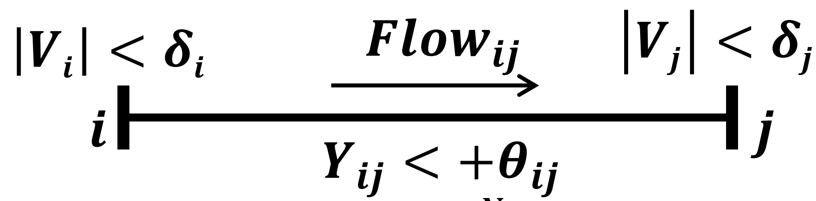
Now, for the specific system demand levels of 50, 250, 300, and 600MW, calculate and analysis:

1) ED generation levels
2) Cost of generation each unit
3) Profit of each unit
4) Incremental cost

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

7.3. Optimal Power Flow …

7.3.5. AC Load Power Formulation



A)Polar Equations

$$\begin{cases} V_i = |V_i|e^{j\delta_i} \\ V_j = |V_j|e^{j\delta_j} \\ Y_{ij} = |Y_{ij}|e^{j\theta_{ij}} \end{cases}$$

$$S_i = V_i I_i^* \overset{*}{\Rightarrow} S_i^* = V_i^* I_i \Rightarrow P_i - jQ_i = V_i^* \sum_{j=1}^{N} Y_{ij} V_j \Rightarrow$$

$$\Rightarrow P_i - jQ_i = \sum_{j=1}^{N} |V_i| e^{-j\delta_i} \times |V_j| e^{j\delta_j} \times |Y_{ij}| e^{j\theta_{ij}} \Rightarrow$$

$$\Rightarrow \begin{cases} P_i = \sum_{j=1}^{N} |V_i||V_j||Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j) \\ Q_i = -\sum_{j=1}^{N} |V_i||V_j||Y_{ij}| \sin(\theta_{ij} - \delta_i + \delta_j) \end{cases}$$

Courtesy of Mohammad Bagher Abaee , Spring 2010

Email: m.b.abaee@gmail.com

7.3. Optimal Power Flow …

7.3.5. AC Load Power Formulation …

$$|V_i| < \delta_i \qquad \xrightarrow{Flow_{ij}} \qquad |V_j| < \delta_j$$

$$i \;\underset{Y_{ij} < +\theta_{ij}}{\longleftrightarrow}\; j$$

B)Cartesian Equations

$$\begin{cases} V_i = V_{Ri} + jV_{Ii} \\ V_j = V_{Rj} + jV_{Ij} \\ Y_{ij} = G_{ij} - jB_{ij} \end{cases} \quad S_i = V_i I_i^* \overset{*}{\Rightarrow} S_i^* = V_i^* I_i \Rightarrow P_i - jQ_i = V_i^* \sum_{j=1}^{N} Y_{ij} V_j \Rightarrow$$

$$\Rightarrow P_i - jQ_i = (V_{Ri} - jV_{Ii}) \sum_{j=1}^{N} (G_{ij} - jB_{ij})(V_{Rj} + jV_{Ij})$$

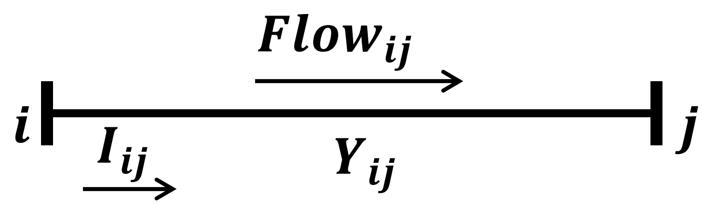
$$\Rightarrow \begin{cases} P_i = \sum_{j=1}^{N} \left[V_{Ri}\left(V_{Rj} G_{ij} + V_{Ij} B_{ij}\right) + V_{Ii}\left(V_{Ij} G_{ij} - V_{Rj} B_{ij}\right) \right] \\ Q_i = \sum_{j=1}^{N} \left[V_{Ii}\left(V_{Rj} G_{ij} + V_{Ij} B_{ij}\right) - V_{Ri}\left(V_{Ij} G_{ij} - V_{Rj} B_{ij}\right) \right] \end{cases}$$

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

7.3. Optimal Power Flow …

7.3.5. AC Load Power Formulation …

B)Cartesian Equations



$$\begin{cases} V_i = V_{Ri} + jV_{Ii} \\ V_j = V_{Rj} + jV_{Ij} \\ Y_{ij} = G_{ij} - jB_{ij} \end{cases}$$

$$I_{ij} = (V_i - V_j)Y_{ij} = I_{Rij} + jI_{Iij}$$

$$Flow_{ij} = V_i I_{ij}^* = P_{ij} + jQ_{ij}$$

$$\Rightarrow \begin{cases} I_{Rij} = G_{ij}[V_{Ri} - V_{Rj}] + B_{ij}[V_{Ii} - V_{Ij}] \\ I_{Iij} = G_{ij}[V_{Ii} - V_{Ij}] - B_{ij}[V_{Ri} - V_{Rj}] \end{cases}$$

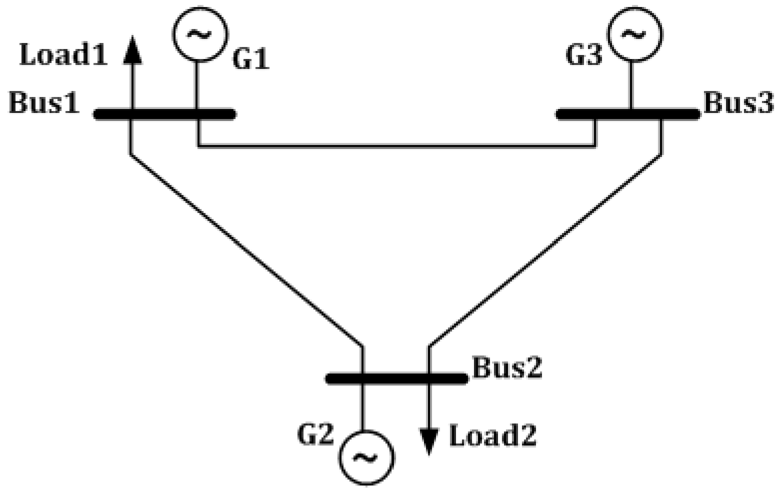
$$\Rightarrow \begin{cases} P_{ij} = V_{Ri}I_{Rij} + V_{Ii}I_{Iij} \\ Q_{ij} = V_{Ii}I_{Rij} - V_{Ri}I_{Iij} \end{cases}$$

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

7.3. Optimal Power Flow …

7.3.6. Optimal AC Power

Consider the three-bus, three-line network that is depicted in below figure.



7.3. Optimal Power Flow …

7.3.6. Optimal AC Power …

Base of 100MVA is considered.

Calculate:

1) Generations
2) magnitude and angle of Voltages
3) Active and reactive loss
4) LMPs
5) Current and power flow of lines

While objective is:

A. Minimum Cost
B. Minimum Network Loss

7.3. Optimal Power Flow …

7.3.6. Optimal AC Power …

Unit	Unit Cost Coefficient			Pmax (MW)	Pmin (MW)	Qmax (MVAr)	Qmin (MVAr)
	a (MBtu)	b (MBtu/MWh)	c (MBtu/MW2h)				
G1	176.9	13.5	0.1	250	0	150	-100
G2	129.9	32.6	0.1	150	0	80	-50
G3	137.4	17.6	0.1	200	0	100	-80

Line No.	From Bus	To Bus	R(pu)	X(pu)	Flow Limit (MVA)
1	1	2	0.02	0.04	150
2	1	3	0.01	0.03	100
3	2	3	0.0125	0.025	200

Bus No.	Load		Voltage-Max (pu)	Voltage-Min (pu)
	MW	MVAr		
1	100	0	1.05	0.95
2	400	250	1.05	0.95
3	0	0	1.05	0.95

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

7.4. UNIT COMMITMENT

- ✓ In formulating the ED, we consider that all generating units are online and ready to produce.
- ✓ The unit commitment problem is solved over a particular time period T; in the day-ahead market, the time period is usually 24 hours.
- ✓ UC is the problem of determining the schedule of generating units within a power system subject to device and operating constraints.
- ✓ The decision process selects units to be ON or OFF, the type of fuel, the power generation for each unit, the fuel mixture applicable, and the reserve margins.
- ✓ Mathematically, UC is a Non-convex, non-linear, large-scale, mixed-integer optimization problem with a great number of 0-1 variables, and a series of prevailing equality constraints.

7.4. UNIT COMMITMENT …

Objective Function:

$$Minimize\{ProductionCost + StartupCost + ShutdownCost\}$$

Subject to:

- **Area Constraints**
  - System Load Balance
  - System Spinning and Operating Reserve Constraints
- **Zonal Constraints**
  - System Spinning and Operating Reserve Constraints
- **Security Constraints**
- **Unit Constraints**
  - Minimum and Maximum Generation limits

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

7.4. UNIT COMMITMENT …

- System Spinning and Operating Reserve Constraints
- Reserve limits
- Minimum Up/Down times
- Hours up/down at start of study
- Must run schedules
- Pre-scheduled generation schedules
- Ramp Rates
- Hot, Intermediate, & Cold startup costs
- Maximum starts per day and per week
- Maximum Energy per day and per study length

7. BASIC MODELS FOR ECONOMICS OF ELECTRICITY

7.4. UNIT COMMITMENT …

7.4.1. Unit Commitment Formulation

Objective Function

$$Minimize\left\{\sum_{i=1}^{N}\sum_{t=1}^{T}\left[F_i\left(P_{i,t}\right)+SU_{i,t}+SD_{i,t}\right]\right\}$$

Subject to:

Power Balance: $\boldsymbol{Demand_i=\sum_{i=1}^{N}P_{i,t}}$ $\forall\ (\boldsymbol{t})$

Min Generation: $\boldsymbol{P_{i,t}\geq P_i^{Min}I_{i,t}}$ $\forall\ (\boldsymbol{i,t})$

Max Generation: $\boldsymbol{P_{i,t}\leq P_i^{Max}I_{i,t}}$ $\forall\ (\boldsymbol{i,t})$

Email: m.b.abaee@gmail.com

7.4. UNIT COMMITMENT …

7.4.1. Unit Commitment Formulation …

Ramp-up limit:	$P_{i,t} - P_{i,t-1} \leq RampUp_i$	$\forall\ (i,t)$
Ramp-down limit:	$P_{i,t-1} - P_{i,t} \leq RampDown_i$	$\forall\ (i,t)$
Minimum up time:	$\left[Xon_{i,t} - MUT_i\right]\left[I_{i,t-1} - I_{i,t}\right] \geq 0$	$\forall\ (i,t)$
Minimum down time:	$\left[Xoff_{i,t-1} - MDT_i\right]\left[I_{i,t} - I_{i,t-1}\right] \geq 0$	$\forall\ (i,t)$
Synchronal Reserve	$\sum_{i=1}^{N} P_i^{Max} I_{i,t} \geq SR_t$	$\forall\ (t)$