INTRODUCTION TO THE GAMS MODELING PROGRAM

B. Essama Nssah

Cornell University
Food and Nutrition Policy Program
1400 16th Street, N.W., Suite 420
Washington, D.C. 20036

Document prepared under the framework of the
Cooperative Agreement No. AFR-000A-00-8045-00 between
the U.S. Agency for International Development and
Cornell University

November 1991
English Translation - March 1993

PREFACE

The Office of Economic Analysis and Forecasting (DAEP) of the Ministry of Planning of the Republic of Niger is conducting, in close collaboration with the Cornell University and under the aegis of the U.S. Agency for International Development, a study of the impact of structural adjustment in Niger.

This essentially quantitative analysis aims at an approach for modeling the macro-microeconomic interactions based on general equilibrium theory. In fact, a Computable General Equilibrium (CGE) model, calibrated for 1987, is in the process of being implemented under the project.

It goes without saying that the true success of this project is measured not by the degree of sophistication of the analytical tools used, but by the effective transfer of these tools to the appropriate Nigerian functionaries. Thus, Cornell University decided to organize, in collaboration with the UNDP project on planning, a series of seminars at the DAEP, targeted at interested officials.

This document represents one of the instructive contributions of the Cornell project. It is an introduction to the GAMS computer program which has been used for the numeric solution of the Niger model.

TABLE OF CONTENTS

1. INTRODUCTION

GAMS, the acronym for General Algebraic Modeling System, is a program that has several modules allowing the solution of problems of mathematical programming. At the same time, it enables an efficient use of the solutions obtained.

This modeling tool was introduced during the 1980s at the World Bank by a team of researchers lead by Alexander Meeraus. Later, the team left the Bank to found a private corporation for the development of the program: The GAMS Development Corporation, based in Washington, D.C.

Interaction with GAMS is done using files. The model is submitted to GAMS using a program input file, and GAMS writes the results to an output file. Both kinds of files are presented in ASCII (American Standard Code for Information Interchange).

The input file contains the program, that is, the collection of coded instructions for the solution of the model in question. One of the advantages of the GAMS language compared to others is that the code in GAMS retains a close resemblance to the original algebraic expressions. Thus, models constructed with this program are extremely compact and easily transferrable.

In fact, GAMS serves to mediate between the modeler and various solution algorithms for problems in mathematical programming. The GAMS programs consists of:

(1) BDM LP1.01, developed by Brooke, Drud and Meeraus for solving problems of linear programming;

(2) MINOS 5.2 (Modular In-core Non-linear Optimization System) developed by the Department of Operations Research at Stanford University in California;

(3) ZOOM, a sub-routine suited to problems where certain variables can only take whole numbers as values;

(4) HERCULES which is an approach for solving computable general equilibrium models.

As it was emphasized in the preface, this paper is an introduction to using GAMS for economic analysis. Therefore, the accent will be on writing the code for certain economic models, and using GAMS, conducting comparative static analyses basic to every study of economic policy.

The plan is the following. The second section of this paper presents a simplified example of the allocation of resources. Section 3 contains the elements of syntax. The Leontief model is presented in Section 4. Section 5 is an introduction to computable general equilibrium models. Section 6 provides conclusions.

2. AN EXAMPLE

2.1 THE PROBLEM

A production unit uses three types of resources or inputs (perhaps capital and two types of labor) to produce three types of output. The production of one unit of the first good requires three units of the first input and two units of the third input. The production of one unit of the second good requires one unit of each type of input. Two units of the second input and one unit of the third are needed to produce one unit of the third good. Given endowments of ten units of the first input, six units of the second input and eight units of the third, what is the optimal production of the three goods, selling on the market at three, two and five thousand francs, respectively?

2.2 THE MODEL CODE

The below program resolves the given problem.

```
SETS

     I    Category of Inputs
          / RES1*RES3/

     J    Category of Goods
          /GOOD1    First Product
           GOOD2    Second Product
           GOOD3    Third Product /

PARAMETERS

     B(I)      Availability of inputs
               /RES1 10, RES2 6, RES3 8/

     C(J)      Unit Prices by Product
               /GOOD1 3, GOOD2 2, GOOD3 5/
```

TABLE
A(I,J)   Matrix of Technical Coefficients

	GOOD1	GOOD2	GOOD3
RES 1	3	1	
RES 2		1	2
RES 3	2	1	1

```
VARIABLES

     R             TOTAL REVENUE
     X(J)          LEVEL OF PRODUCTION

POSITIVE VARIABLES

     X;

EQUATIONS

REVENUE OBJECTIVE FUNCTION
CONTR(I)  CONSTRAINTS;
REVENUE...     SUM(J, C(J)*X(J)) =E=R;
CONTR(I)...    SUM(J, A(I,J)*X(J)) =L=B(I);
MODEL     REV Enterprise Revenue  / ALL /;
SOLVE     REV  MAXIMIZING R USING LP;
DISPLAY   X.L, X.M, CONTR.L, CONTR.M;
```

Solving the problem requires that the above commands be translated to an ASCII file. In this case, we have named the input file EXEMPLE1.GMS. The following command submits the GAMS file for processing:

```
GAMS EXEMPLE 1
```

It should be noted at this point that GAMS looks for a file with the expansion ".GMS". If another extension is used instead, it must be specified the moment that GAMS is invoked.

Once the problem is solved, GAMS writes the results to an output file bearing the same name as the input file but with the extension ".LST". This file is present in the appendix. The optimal production plan indicates that 2.5 units of the first good and 3 units of the third should be produced. In terms of the GAMS code, x = (2.5 0 3).

2.3 THE STRUCTURE OF THE PROGRAM

To better understand the structure of the model presented above, it is useful to consider the mathematical representation of the problem to be solved. It is, in effect, a linear programming problem written in the following manner:

maximize $$r = \sum_j c_j n_j, \quad j = 1, 2, 3$$

such that $$\sum_j ai_j n_j \le b_j, \quad i = 1, 2, 3.$$

This problem implies at least six concepts or entities. First, there are two indices i and j indicating the ranges of other variables or parameters.

The initial data are represented by parameters c_j, b_i and a_{ij}. There are two types of variables: three instrumental variables n_1, n_2 and n_3, and one criterion variable, r, the level of which enables us to evaluate the optimal nature of the decision. The other three concepts of importance are the equations (or inequalities), the model and the solution method. The equations or inequalities represent the relationships between variables. The model is the combination of the relationships between variables. Here we distinguish an equation describing the objective function and three inequalities representing the constraints. Since we are concerned with an optimization problem where all relationships between variables are linear, the required solution method is that of linear programming.

The logic of programming in GAMS follows from the structure that we just presented. In general, all components (indices, data, variables, equations, model and solution algorithms) must be declared and defined. Eventually, instructions are added to facilitate reading the results.

A declaration in GAMS is a statement (using proper syntax) presenting the type and name of a component (eg., a variable or a parameter). Such a statement has three parts. It must begin with a key word followed by the name of the entity. The third element is optional. This is descriptive text of up to 80 characters. The key word determines the class of the element under consideration. The key words appearing in our program are:

1.	SET(S):	Indices
2.	PARAMETER(S):	Data in vector form
3.	TABLE(S):	Data in matrix form
4.	VARIABLE(S):	Variables
5.	EQUATION(S):	Equations
6.	MODEL:	Model
7.	SOLVE:	Solution

It should be remembered that all words are in English, although the programmer has the choice of language used in the cases of symbols representing the various components and the descriptive text. In general, though, if the descriptive text is in French, one must not use accents because they will cause errors during compilation.

A definition is a statement that attributes a particular value or form to a component provided that the latter had been previously declared. Except for the case of equations, one may combine a declaration with a definition in a single statement. The following statement declares the index i of resources.

```
SET       I    Index of resources;
(Structure:  keyword, symbol, descriptive text).
```

By contrast, the following combines the declaration with the definition of the same index.

```
SET       I    Index of resources      /RES1*RES3 /;
(Structure:  keyword, symbol, descriptive text and assignment of values).
```

Regarding punctuation, it should be remembered that statements in GAMS should be separated by a semicolon (;). However, if a statement begins with a key word, a semicolon is not required at the end of the preceding statement.

The general ideas that we have just presented through our example are fundamental to the GAMS modeling system. To better understand the GAMS system, we should consider in greater detail the various aspects of the programming language.

3. ELEMENTS OF SYNTAX

3.1 INDICES

We have seen above, indices such as *i* or *j*, serve to determine the domain defined with variables or parameters. Each index consists of a series of elements serving as labels to the parameters and variables to which they are assigned. GAMS treats indices as sets. This is elsewhere translated by the choice of the key word "SET."

A set may be defined, whether by listing every element or by specifying some common property. GAMS uses both methods. The following expression:

$$E = \{a,b,c\}$$

which states that set E consists of elements a, b, and c, is written in GAMS in the following way:

```
SET  E    / a, b, c /
```

The elements must be separated by a comma or be at the end of a line. As such, set E may also be defined as follows:

```
SET  E    /a   first element
            b   second element
            c   third element /
```

It should be noted that the list of elements is contained between two slashes (/ /). Accents are omitted (when the program is written in French) so as to avoid syntactical errors during compilation.

The name of the set as well as those of the elements must not exceed ten characters in length. The first character must be a letter of the alphabet. Descriptive text may accompany labels of elements provided that the two are separated by spaces. The set of goods in section 2 had been defined in this manner:

```
SET  J    Index of Goods
     /GOOD1    First good
      GOOD2    Second Good
      GOOD3    Third Good /
```

If the elements of a set follow a consecutive order, such a set may be defined in an abbreviated manner by placing an asterisk between the name of the first element and that of the last. For example:

```
SET  I    Index of Resources
```

```
        / RES1*RES3 /
```

The definition of this set is equivalent to:

```
        / RES1, RES2, RES3 /
```

In certain instances, it becomes necessary to use two different names for the same set. The instruction is given as follows:

```
    ALIAS(k,l);
```

This means that set *k* is equivalent to set *l*.

To conclude the discussion on sets, we illustrate, using the nomenclature of the national accounts system, the typical operations using sets.

We define the set of economic activities:

```
    SET       S    Activities by sector
    /AGRICULT           Agriculture
     ELEVAGE            Livestock
     PECHE-FOR          Fishing and Forestry
     MINES              Mining
     MANUFACT           Manufacturing
     ENERGIE            Electricity and Water
     BTP                Construction and Public Works
     COMMERCE           Commerce, hotels and restaurants
     TRANSPORT          Transportation and Telecommunications
     SERVICES           Private Services
     ADMINIST           Administration /
```

We can ask GAMS to count the total number of elements in set S or to give their ordering. We proceed in the following fashion:

```
    PARAMETER      N         Number of elements in S
                   O(S)      Order of elements of S
    N = CARD(S)    O(S) = ORD(S);
```

First we created two parameters *N* and *O(S)*. With the CARD( ) function, GAMS assigns to *N* the value of 11, which is equal to the total number of elements in set *S*. Later, we will discuss entry and processing of data using parameters. The ORD( ) function gives the relative position of an element in the set. Thus, ORD(S) assigns the value 1 to the element "AGRICULT", and 2 to "ELEVAGE", and so on. The result is stored in *O(S)*.

We create three sub-sets of set *S* with the following commands:

```
    SET       PR(S)     Primary Sector
              / AGRICULT, ELEVAGE, PECHE-FOR /
              SE(S)     Secondary Sector
                  / MINES, MANUFACT, ENERGIE, BTP /
              TE(S)     Tertiary Sector
```

```
/ COMMERCE, TRANSPORT, SERVICES, ADMINIST /;
```

The domain *S* must appear between parentheses in the specification of the names of the sub-sets to indicate their domain characteristics. Thus, PR(S) means that the elements of PR are also the elements of *S*.

Supposing that, for whatever reason, the mining sector is considered as part of the primary sector rather than the secondary sector, we can modify the composition of the sub-sectors with the aid of the following commands:

```
PR("MINES") = YES;
SE("MINES") = NO;
```

The first command adds the element to the primary sector while the second removes it from the secondary sector. If by mistake, one enters the following instructions:

```
PR(S) = YES;
SE(S) = NO;
```

The result would be that the primary sector includes all 11 sectors while the secondary sector would be reduced to an empty set.

The union, the difference and the intersection of sets are created using the symbols +, -, *. The complement of a set is obtained by using the operator NOT. To illustrate, we will create the following sets:

```
SET  UNI(S)    Union
     DIF(S)    Difference
     INTR(S)   Intersection
     COMP(S)   Complement;

UNI(S)    =    PR(S)  + SE(S);
DIF(S)    =    UNI(S) - SE(S);
INTR(S)   =    SE(S)  * TE(S);
COMP(S)   =    NOT TE(S);
```

The set *UNI*, defined over *S*, includes all the elements of *PR* and all the elements of *SE*; *DIF* is the set of all the elements of *UNI* excluding the those belonging to *SE*; *INTR* is the set of elements belong to both *SE* and *TE*; Finally, *COMP(S)* is the set of all elements of *S* not belonging to *TE*, that is, the set of non-tertiary sectors. The results of these operations on sets are presented in the appendix.

3.2 DATA

Data Entry. The data for a problem may be entered three ways: as scalars, vectors or as matrices. The structure of the commands is as follows:

```
Keyword        Symbol(domain)          Descriptive text
```

The key words used are:

```
SCALAR       for scalars
PARAMETER    for vectors and
TABLE        for matrices
```

The specification of parameter names is done according to the same conventions used for indices. The maximum length is 10 characters.

The program presented in section two, above, provided the following two examples.

```
    PARAMETER          B(I)          Availability of Resources

         / RES1  10, RES2  6, RES3  8 /

TABL  A(I,J)  Matrix of Technical Coefficients

                    GOOD1          GOOD2        GOOD3

    RES1             3               1
    RES2                             1            2
    RES3             2               1            1
```

Note that it is not necessary to enter zero. The vector providing the availability of resources could have been entered in the following manner:

```
    PARAMETER          B(I)          Availability of Resources

         / RES1   10
           RES2    6
           RES3    8 /
```

Thus each point is a pair represented by an element from the index domain and the value associated with the parameter.

Regarding matrices, it should be mentioned that the defined domain is determined by two indices in parentheses. The first defines the lines and the second defines the columns of the matrix. GAMS will indicate an error if there is any question about which column a number goes with because the number is not properly aligned within the column.

The following table illustrates the point:

```
Table   D(line, col)          Misalignment
                  Col 1                          Col 2        Col 3
 Line 1                                 35495      25          420
 Line 2             4950                           44          550
 Line 3            7500                            52          350
```

With correction, the table should look like this:

```
Table     D(line, col)
                        Col 1              Col 2        Col 3
 Line 1                 35495               25           420
 Line 2                 4950                44           550
 Line 3                 7500                52           350
```

Although the numbers in the first column are not exactly aligned with the column label, they do not overflow into the second column.

A matrix may be so large that all the columns do not fit on one line. In this case, one can divide the matrix as is shown below:

```
Table A(I,J) Technical Coefficients
              GOOD 1      GOOD 2      GOOD 3      GOOD 4      GOOD 5
 RES1            x           x           x           x           x
 RES2            x           x           x           x           x
 RES3            x           x           x           x           x
 +                 GOOD 6      GOOD 7      GOOD 8
 RES1            x           x           x
 RES2            x           x           x
 RES3            x           x           x
```

Here is an example of data entry using a scalar:

```
SCALAR:       r         Interest rate      /10.07/;
```

Processing of data. The database may be transformed using standard functions as well as various other operations. Table 3.1 describes several functions commonly used by GAMS.

The following symbols are used for mathematical functions:

**	exponent
*	multiplication
/	division
+	addition
-	subtraction
=	equal sign

Example:

```
    SCALAR          x          An arithmetic operation;

                x = 5 + (4*(3**2));
```

The instructions above tell GAMS to create a scalar *x* and to give it a value of three squared times four plus 5, that is x = (9 x 4) + 5 = 41. To facilitate reading the results in the GAMS output file, we use the command:

DISPLAY *x*;

It should be noted that the operation *x****n* (*x* raised to the power *n*) is not defined in GAMS if *x* is negative. In this case the POWER(*x*,*n*) function must be used provided that *n* is a whole number.

Concerning parameters with defined domains (represented by indices), certain calculations are carried out by index operations using the following operators:

SUM	sum
PROD	product
SMIN	minimum
SMAX	maximum

Table 3.1 — Several Common GAMS Functions

Name	**Description**
ABS	Absolute value of the argument
COS	Cosine of the argument where the angle is measured in radians
EXP	Exponential function
LOG	Natural Logarithm (base e)
LOG10	Logarithm (base 10)
MAX	Maximum of a series of arguments
MIN	Minimum of a series of arguments
POWER	First argument raised to the power of the second argument which must be a whole number
ROUND	Rounding of the first argument to the number of decimal places indicated by the second argument
SIGN	= +1 if the argument is positive -1 if it is negative 0 if it is equal to zero
SIN	Sin (angle in radians)
SQR	Argument squared
SQRT	Square root of the argument

Source: Brooke, Kendrick and Meeraus (1988) p. 69.

Let us now reconsider our example of the allocation of inputs from section 2. Total income corresponding to the production of a unit of each good is equal to:

$$r = \sum_{j} c_j,$$

In GAMS this is written as: *r = SUM(J, C(J));*

The smallest and the largest are found using the following commands:

```
SCALAR        pp        small price
              gp        large price

              pp  =     SMIN(J, C(J));
              gp  =     SMAX(J, C(J));
```

The structure of the operator PROD is better shown in the context of a production function. Let us consider the Cobb-Douglas production function, for example. If the level of production is represented by *x*, the factors of production by F(I) and the exponents by ALPH(I), we have the following relationship:

```
x = PROD(I, F(I) ** ALPH(I));
```

Other operators permit us to impose various conditions on calculations. These are:

LT	less than
LE	less than or equal to
EQ	equal to
NE	not equal to (different from)
GE	greater than or equal to
GT	greater than
NOT	not
AND	and
OR	or
XOR	or, or
$	Dollar

The table of authorities of the four logical operators is given in Table 3.2.

In general, the dollar sign operator has two functions in GAMS. It may instruct the program on the structure of the output file; in this case it appears in the first column of the line containing the relevant command. It can also be used in conditional statements. The meaning of the condition varies with the operator appearing to the right or left of the equals sign. Consider the following examples:

```
SCALAR    x, y
```

```
    y = 2;    x = 1;
    x = 2$(y GT 1.5);
```

The first two commands create two scalars *x*, and *y*, and assign to them the values of 1 and 2, respectively. The last instruction changes the value of *x* on the condition that *y* assumes a certain value. The fact that the dollar sign operator appears on the right hand side of the equation means the following: if *y* is greater than 1.5, *x* takes on the value of 2, if not, *x* is equal to zero.

By contrast, if we had written:

```
SCALAR    y, x
    y = 2;    x = 1;
    x$(y GT 1.5) = 2;
```

The interpretation would have been: if *y* had been greater than 1.5, then *x* equals 2; if not, the value of *x* would remain the same, that is , 1. This new interpretation is due to the fact that the operator appears on the left hand side of the equation.

Table 3.2 — Results of Logical Operators[1]

Variables		Operations			
a	b	a and b	a or b	a xor b	not a
0	0	0	0	0	1
0	non null	0	1	1	1
non null	0	0	1	1	0
non null	non null	1	1	0	0

Source: Brooke, Kendrick and Meeraus (1988) p. 72.

The expression *x = 1/y$y* is equivalent to *x = 1/y$(y NE 0)*. The condition imposed on *y* allows us to avoid division by zero (an undefined operation).

The following commands illustrate matrix algebra based on the two previously defined matrices *A(I,J)* and *B(I,J)*.

```
PARAMETERS    C(I,J)    sum of A and B
              AB(I,K)   product of A and B
              AT(J,I)   transpose of A;
```

[1] If a and b are considered as elementary propositions, then table 3.2 is a table of authorities where 0 means false and 1 true.

```
     C(I,J)    =    A(I,J) + B(I,J);
     AB(I,K)   =    SUM(J,A(I,J)*B(J,K));
     AT(J,I)   =    A(I,J);
```

Multiplication of two matrices is not possible if the number of columns of the first is not equal to the number of rows of the second. To subtract matrices, is suffices to replace the + sign by the - sign in the equation.

3.3 VARIABLES

Variables are the unknown entities determined with the solution of the model. There are four ranges of values associated with a variable and are signified by the following suffixes:

.LO	lower bound
.UP	upper bound
.L	level
.M	marginal or dual value

The first two bounds must be determined by the GAMS user whereas the level and the marginal value change after the solution of the model is found. If the lower and upper bounds are not preset, GAMS assigns values by default according to the types of variables. Table 3.3 shows the values in question. Thus a free variable varies from negative infinity (-INF) to positive infinity (+INF).

Variable declarations are done in the same manner as those of indices or parameters. It is also possible to combine the variable declaration with its specification. Specification of a variable requires the use of one of the key words in Table 3.3. We use the example from Section 2 (above) to illustrate the two ways that one can declare and specify variables in GAMS:

```
(1)  VARIABLES
          R         total revenue
          X(J)      level of production
     POSITIVE VARIABLES      X;

(2)  FREE VARIABLE
          R         total revenue
     POSITIVE VARIABLE
          X(J)      level of production;
```

In the first case, *R* is implicitly defined as a free variable. In the second case, this definition is explicit and associated with the declaration. In certain situations, especially under the rubric of non-linear programming, it is necessary to change the implicit bounds of certain variables or to set initial values to start the solution algorithm. For example, to avoid division by zero, the lower bound of a variable may be modified in the following manner:

```
X.LO(J) = 0.01;
```

Table 3.3 — Types of Variables and Associated Bounds

Keyword	Bounds	
	.LO	.UP
FREE	-INF	+INF
POSITIVE	0	+INF
NEGATIVE	-INF	0
BINARY	0	1
INTEGER	0	100

Source: Brooke, Kendrick and Meeraus (1988) p. 83.

If one wants the upper bound to be equal to the lower bound, the suffix .FX is used as in the following example:

```
X.FX(J) = 5;
```

In the context of the example presented in Section 2, let us suppose that we wish to impose the constraint that the level of production of Good 1 remain fixed at 1.5 units. We would write:

```
X.FX('GOOD1') = 1.5;   Or X.FX("GOOD1") = 1.5;
```

In general, the four ranges of values associated with a variable can be used in algebraic calculations such as those presented in the above subsection entitled "Processing of Data." For example, after solving the model of Section 2, we can create a parameter to save total revenue. The commands are:

```
SCALAR   RTOT          Total Revenue;
         RTOT  = SUM(J,C(J)*X.L(J));
```

3.4 EQUATIONS

The equations represent the mathematical structure of the problem. In general, they express the various algebraic relationships between variables.

The symbols used in these equations are the following:

Symbol	Meaning
=E=	Equality: the right hand side (RHS) must equal the left hand side (LHS)
=G=	Greater than: The LHS is greater than the RHS
=L=	Less than: The LHS is less than the RHS
=N=	No relationship exists between the LHS and the RHS

As we have already emphasized, the declaration of equations must be done separately from their definition, the two groups of propositions being separated by a semi-colon (;). The declaration of equations follows the same syntax rules as those already discussed in the context of indices and parameters.

In the example from Section 2, the two types of equations were declared in the following manner:

```
EQUATIONS
     REVENUE        Objective function
     CONTR(I)       Constraints;
```

The structure is: keyword, identification(domain), explanation.

The same equations were specified as follows:

```
REVENUE..           SUM(J,C(C)) =E= R;
CONTR(I)..          SUM(J,A(I,J)*X(J)) =L= B(I);
```

The two dots (..) which follow the equation name alert GAMS that an algebraic expression will follow. The semicolon (;) at the end of the expression is absolutely indispensable because the proposition that follows in the same block does not begin with a keyword.

In the context of the specification of equations in GAMS, it is appropriate to note that the expressions are adequately compact because of the use of indices representing the domains. Thus, the equation CONTR(I) represents in fact three equations, one for each resource. The expression would not have changed if set I had instead contained 150 elements!

Finally, it should also be noted that there are four ranges of values associated with equations in GAMS. The suffixes used to construct them are the same as those used with variables: .L, .LO, .M, and .UP. These values are not assigned until after the model has been solved and can be viewed using the DISPLAY command. In particular,the value (.M) associated with a constraint represents that of the Lagrange or Kuhn-Tucker multiplier. This value tells us how much the optimum value of the objective function will change if the constraint was relaxed a bit.

3.5 MODELS AND SOLUTION ALGORITHMS

As with the other entities in GAMS, models and solution algorithms must be declared and specified. This may be accomplished by one statement for each entity. Again, the case from Section 2 provides an example:

```
MODEL     REV       Enterprise revenue      /ALL/;
```

The structure of this statement breaks down into the following:

Keyword	MODEL
Identification	REV
Explanatory text	Enterprise revenue
Definition	/ALL/
End of statement	;

The word ALL between slashes is shorthand notation which informs GAMS that the model (or system of equations) that we have named REV, contains all the previously defined equations. In the same manner, we could have written:

```
MODEL     REV       Enterprise revenue
          /REVENUE,CONTR/
```

This feature assumes much more importance in the context of large models where several versions may be specified vis-à-vis sub-sets of defined equations.

The solution algorithm must include a keyword, the name of the model to be solved, the direction of optimization, the criterion variable which measures the level of the objective function, and the type of algorithm to be used.

Example:

```
SOLVE     REV  MAXIMIZING R USING LP;
```

SOLVE is the keyword, REV is the model name, MAXIMIZING gives the direction of optimization (for minimization one would write MINIMIZING), *R* is the criterion variable and USING LP means that linear programming algorithms are to be used to solve the problem. If our problem had been non-linear we would have written USING NLP.

It is worth noting that GAMS does not solve the problem by itself. It analyzes, interprets and submits the program to an associated sub-program. Once the problems has been solved, GAMS translates the results into an output file. For our simple example, we want to examine the levels of production and the marginal values, the levels of resource use and the multipliers associated with the constraints. The command is:

```
DISPLAY X.L, X.M, CONTR.L, CONTR.M;
```

3.6 THE OUTPUT FILE

It is useful to remember that from the beginning, our interaction with GAMS is done using files. We submit our problem in an input file and GAMS responds with an output file. In the context of our example in Section 2, GAMS wrote the results to a file named EXEMPLE1.LST. We present it in its entirety in the appendix and our discussion here will center on this file.

The information contained in the output file result from two types of procedures: the compilation, which can be seen as a syntactic verification of our work by GAMS, and the solution of the model. If the slightest error is detected during compilation, the program will not be executed. GAMS generates a list of all the errors encountered and makes suggestions regarding their source. The structure of the output file various then according to whether or not there are any errors. If there are errors, the information contained in the output file depends on the nature of the error, as it may be a case of an error of programming language or of calculation during the solution routines. For the moment our discussions will concentrate on the ideal case where the program runs without error.

The first information presented by GAMS is a copy of the program that we submitted to GAMS. This always appears at the beginning of the output file, whether or not there were any errors.

From here on, we present the blocks of information in the order that they appear in the file, using the same (english) headings as does GAMS.

3.6.1 Symbol Listing

At this stage, GAMS provides information on all the identifiers (symbols) contained in the program. This information is provided in two forms. First, an alphabetical listing of all the terms. The first column gives the name of the symbols. The second column indicates their types (SET, PARAMETER, etc.), and the last bloc of columns provides the line numbers where the symbols appear. The addresses are indicated by a word and a line number. Such a reference tells us what the identifier does at the particular line of the program. The words used are the following:

DECLARED — The identifier is declared as to type.

DEFINED — The identifier is defined.

ASSIGNED — The values if the symbols are modified by an arithmetic or logical operator.

IMPL-ASN — This is an implicit assignment. The range of values of an equation or a variable are updated as a result of being referred implicitly in a solve statement.

CONTROL — The index determines the domain of an equation, of an operator such as SUM, PROD, SMAX and SMIN, or any other arithmetic or logical expression.

REF — Reference to a symbol.

For symbol *I* of our example, we have the following characteristics:

```
I    SET  DECLARED 4     DEFINED 5      REF 12

                    17  33  2*36  CONTROL 36
```

This means that *I* is a set, declared on line 4 of the program, and defined on the fifth line; the symbol is referenced once on line 12, once on line 17, once on line 33 and twice on line 36. Finally, *I* serves as the control index on line 36.

After this table, GAMS again reproduces all the symbols, this time classified by type (SETS, PARAMETERS, VARIABLES, etc...). If the user does not want to see these two listings, the following commands are inserted at the very beginning of the program.

```
$OFFSYMXREF   OFFSYMLIST
```

The dollar sign must be placed in the first position (first column) of the line.

3.6.2 Equation Listing

This list constitutes the first result of the execution of the SOLVE command. Here, the equations are rearranged so that all the variables contained in an equation appear on the lefthand side of the expression and all the constants appear on the righthand side. This transfer of terms between the two sides of the equation is carried out according to the rule that every variable that changes sides must change sign as well. Thus, equation REVENU is displayed in the following manner:

```
REVENU..       -R + 3*X(GOOD1) + 2*X(GOOD2)

                    +5*X(GOOD3) =E= 0;
```

The block of equations CONTR(I) creates three individual equations following the definition of *I*. The first equation of the block appears as follows:

```
CONTR(RES1).. 3*X(GOOD1) + X(GOOD2) =L= 10;
```

In the context of non-linear models, if the coefficient of a variable is displayed in parentheses, it is because the term is nonlinear. To find the value in parentheses, one takes the partial derivative of the equation relative to the variable and evaluates it using the levels (.L) of all the variables contained in the equation.

By default, GAMS only uses the first three equations of each block. If the user wants to see more or less than three, the following command is inserted before the SOLVE statement:

```
OPTION LIMROW = n;
```

where *n* is a whole number indicating the number of equations that one wants to see.

3.6.3 Column Listing

GAMS provides by default for the first three cases of each variable (according to the domain definition) the values associated with each field (.LO, .L, .UP), a column of coefficients, as well as the names of the equations where this variable is assigned by the given coefficients. Thus, for variable *R* we have:

```
R
          (.LO, .L, .UP = -INF, 0, +INF)
-1  REVENU
```

This means that variable *R* may take on values between negative and positive infinity. Its initial value is zero and is assigned to the coefficient -1 in the equation REVENU.

If the user wants to view more or less than three cases for each variable, the following command is inserted before the SOLVE statement:

```
OPTION    LIMCOL = n;
```

3.6.4 Model Statistics

This concerns the number of equations and variables, the time taken by GAMS to generate the model and the execution time. This time is measured from the end of the verification of the model syntax. Occasionally this block contains information on the non-linearity of the model.

3.6.5 Solve Summary

At this level, we have a condensed report on the model. GAMS displays the name of the model, the criterion variable, the type of model, the direction of optimization and the name of the sub-program used for finding the solution. In three lines beginning with four asterisks (****) messages appear. These messages let us verify if the sub-routine for the solution has successfully resolved the problem, if the solution obtained is optimal, and what is the value of the criterion variable. The messages are as follows:

```
**** SOLVER STATUS 1    NORMAL COMPLETION

**** MODEL STATUS  1    OPTIMAL

**** OBJECTIVE VALUE         22.5000
```

The number 1 appearing before the messages is a code. For example, in the framework of non-linear problems, the solution method can only guarantee one local optima. In this case, the code indicates 2 and the message: LOCALLY OPTIMAL. Additional technical information is provided at this point. We leave them aside in this introductory framework.

Later, GAMS provides a more detailed solution report. For each equation and variable, the values associated with each field are displayed provided that they are calculated by the solution routine. The columns are entitled: LOWER, LEVEL, UPPER and MARGINAL. A period (.) in the table represents zero while the expression EPS means that the number is very small but not zero.

After this detailed report, GAMS displays another summary. In the context of our example, we have:

```
**** REPORT SUMMARY     0 NONOPT

                        0 INFEASIBLE

                        0 UNBOUNDED
```

The zeros represent the number of lines or columns in the solution indicating NONOPT (non-optimal), INFES (infeasible) or UNBD (unbounded). The summary above indicates a good solution.

If our programs finishes with the command DISPLAY, the results appear after the summary. The last information contained in the output file are the names of the input and output files.

After this brief presentation of basic syntax, we will consider two types of applications. One uses linear programming and the other use non-linear programming solution methods. One is a Leontief model and the other is a computable general equilibrium model.

4. THE LEONTIEF MODEL

4.1 STRUCTURE

Assume an economy of *n* sectors. Each sector is assumed to only produce one good using one technology. The production process is such that the sector uses goods and services from other sectors as factors of production in addition to primary factors of production. We assume as well that the quantity of good *i* used per unit of good *j* is directly proportional to the level of production of sector *j*. Algebraically we write:

$$x_{ij} = a_{ij}\, x_j \qquad \textbf{(1)}$$

where a_{ij} is a constant technical coefficient. An immediate implication of this proposition is that the technology is characterized by constant returns to scale.

Thus, the distribution of sectoral production between intermediate and final consumption may be described by a system of linear equations representing the equilibrium of availabilities and uses for each good.

$$x_i = \sum_j a_{ij}\, x_j + b_i,\ i = 1,2..,n \qquad \textbf{(2)}$$

or in matrix form

$$x = Ax + b \qquad \textbf{(3)}$$

A is a square matrix of dimension *n*, *x* and *b* are vectors of dimension *n*, and *b* represents the vector of final demands.

Equation 3 could be written in the following manner:

$$(I - A)\, x = b \qquad \textbf{(4)}$$

where *I* is a diagonal matrix of the same dimension as *A* and where each element of the diagonal is equal to 1 (identity matrix). If matrix (I-A) has an associated inverse matrix, which we will call U, then the system is equivalent to:

$$x = Ub \qquad (5)$$

or in other terms

$$x_i = \sum_j U_{ij} b_j \qquad (6)$$

That is to say that the level of production in each sector *i* is a linear combination of the level of final demand in all sectors of the economy. The coefficients are the Leontief inverse elements.

To better interpret these coefficients, note that:

$$Uij = \frac{\partial x_i}{\partial b_j} \qquad (7)$$

This is the partial derivative of the output of sector *i* relative to the final demand for sector *j*. This measures the multiplier effect of a small change in final demand of sector j for output from sector *i*. The total effect of the economy of such a change is measured by:

$$m_j = \sum_i u_{ij} \qquad (8)$$

By contrast:

$$m_i = \sum_j u_{ij} \qquad (9)$$

measures the stimulus of sector *i* if all final demand were augmented by one unit. In fact, (4.9) is obtained by replacing b_j by 1 in (4.6). The two measures may be normalized by dividing each by:

$$m = \frac{1}{n} \sum_i \sum_j u_{ij} \qquad (10)$$

We then obtain the Rasmussen indices m_j^* and m_i^*, which allow us to identify the key sectors of an economy. A sector is considered to be a key one if at least one of the indices is greater than one. If $m_j^* > 1$,, growth in final demand of sector *j* has a greater than average effect on the economy. If $m_i^* > 1$, the sector is likely to receive a greater than average stimulus if final demand increases in all sectors.

4.2 APPLICATION

We give here the structure of a GAMS program which allows us to calculate the Leontief inverse matrix and to derive the Rasmussen indices.

The data that we use here are taken from Jefferson and Boisvert (1989). the economy under consideration consists of four sectors: agriculture, manufacturing, transportation and services.

Intersectoral transactions are presented in the following table:

TABLE IO(*i*,*j*) Intersectoral Transactions

	AGRICULT	MANUF	TRANSPORT	SERVICES
AGRICULT	34	290	0	0
MANUF	25	1 134	5	201
TRANSPORT	6	304	54	105
SERVICES	48	962	71	877

The rest of the data is entered in the following manner:

TABLE SECTORS(*, *j*) Sectoral Data

	AGRICULT	MANUF	TRANSPORT	SERVICES
VA	356	11 622	480	57 140
CD	7	607	22	2 558
AD	38	12 340	119	2 381
X	469	14 312	610	6 897

It should be noted that the domain of variation of the rows in matrix SECTRES is given by the symbol *. In this case, GAMS may accept descriptions that have not already been defined[2].

The matrix of technical ccefficients is calculated using the following command:

$$A(i,j) = IO(i,j) \ / \ XO(j);$$

where $XO(j)$ represents the sectoral level of production. The index *i* of sectors of production is also called *j* or *k*. This was accomplished using the command:

[2] VA = Value added; CD = Household Final Demand; AD = other final demand and X = production.

```
    ALIAS (i,j,k);
```

placed after the definition of index *i*.

The 21 variables and 21 equations are defined in three blocks.

```
    VARIABLES

         GDP        GDP         (1 variable)

         INV(j,k)   Leontief inverse (16 variables)

         X(j)       Level of Production (4 variables)

    POSITIVE VARIABLE  X;

    EQUATIONS

         GDPEQ            Definition of GDP (1 equation);

         INVEQ(i,k)       Constraints to calculate from the inverse (16
                          equations);

         MBEQ(i)          Supply-demand equilibrium by product (4 equations);

         GDPEQ..          SUM(j,PVO(j) * X(j)) =E= GDP;

         INVEQ(i,k)..     SUM(j,(IDM(i,j)-A(i,j))*INV(j,k)) =E= IDM(i,k);

         MBEQ(i)..        SUM(j,(IDM(i,j)-A(i,j))*X(j)) =E= FDO(i);
```

The last block of equations is equivalent to the expression $(I-A)x = b$, insofar as the second block is of the form:

$$(I-A)B=I \qquad \textbf{(11)}$$

Which implies that:

$$B=(I-A)-1 \qquad \textbf{(12)}$$

After the definition of the Leontief model, we redefine the parameters that will enable us to calculate the Rasmussen indices and to present the results in tabular form. The entire program and details of the results are presented in the appendix. The main results are found in the following tables:

Table 4.1.: Technical Coefficients

	AGRICULT	MANUF	TRANSPORT	SERVICES
1	0.0725	0.0203	0.0000	0.0000
2	0.0533	0.0792	0.0082	0.0291
3	0.0128	0.0212	0.0885	0.0152
4	0.1023	0.0672	0.1164	0.1272

Table 4.2.: Léontif Inverse Matrix

	AGRICULT	MANUF	TRANSPORT	SERVICES
AGRICULT	1.0796	0.0238	0.0003	0.0008
MANUF	0.0669	1.0905	0.0145	0.0367
TRANSPORT	0.0190	0.0273	1.0999	0.0201
SERVICES	0.1343	0.0904	0.1478	1.1523

Table 4.3. Rasmussen Indices

	$m^{\bullet}_{j}$	$m^{\bullet}_{i}$
AGRICULT	1.0392	0.8831
MANUF	0.9850	0.9663
TRANSPORT	1.0094	0.9324
SERVICES	0.9665	1.2182

According to these results, the two key sub-sectors of the economy are: agriculture and services. The first is able to generate a stimulus above the economy-wide mean while the second is likely to be stimulated more than the other sectors by a general increase in demand emanating from other sectors.

5. COMPUTABLE GENERAL EQUILIBRIUM MODELS

5.1 THE CONCEPT OF GENERAL EQUILIBRIUM

Economics, as a science, is generally defined as the study of the allocation of scarce resources to satisfy unlimited wants. The basic elements constituting an economic system are: agents, goods and services and institutions. In a market economy, one distinguishes two fundamental types of agents: the consumer (household) and the producer (firm). We assume that there is one market for each good of service. As well, we may state that the allocation of resources, in this context, is determined by individual behavior and by interactions among groups of agents in the framework of markets.

The study of agent behavior and the interactions between them is based on the principles of optimization and equilibrium. We assume that every economic agent has an objective which guides his or her actions. The range of action is limited by a certain number of constraints facing each agent. For example, the consumer seeks the maximization of utility (preference) from goods and services the he/she consumer subject to a budget constraint. The producer, facing technical and market constraints, seeks to maximize profits, or, equivalently, to minimize costs of production.

The optimization hypothesis imposes restrictions on the behavior of economic agents. The restrictions are reflected in the structure of demand and supply functions. For a given good, for example rice, we may think of the quantities demanded and supplied as being dependent on price. Furthermore, we may represent the two functions by D(p), for demand, and S(p), for supply (p is the price of the good). The market is in equilibrium when supply is equal to demand. The equilibrium price is represented by p^*. For a given market, if supply is not equal to demand, we assume that the price will adjust to bring about equilibrium conditions.

The way in which we just described supply and demand functions represents only a partial look. It is evident that the supply and demand for a good such as rice depends not only on its price, but on the price of substitute or complementary goods (millet, maize, etc..). In the framework of partial equilibrium analysis, the prices of complements or substitutes are considered as exogenous variables and do not explicitly appear in supply and demand functions.

In the context of general equilibrium analysis, by contrast, the focus is on the simultaneous determination of several prices based on supply and demand conditions prevailing in the relevant markets. General equilibrium can be modeled using a system of equations where the main variables are prices and quantities of goods and services. A situation of general equilibrium is characterized by the fact that supply is equal to demand in each market. Walras' Law states that, for a system of n markets, if $(n-1)$ are in equilibrium, then the last market will automatically be in equilibrium. Thus, there are only *(n-1)* prices that can be determined independently. In other words, a system of equations describing a state of general equilibrium only determines relative prices. Some good can be chosen to be a numéraire with a reference price fixed at 1.

A fundamental question arises regarding the desirable characteristic of an equilibrium situation. In the framework of pure and perfect competition between all economic agents, it can be demonstrated that an equilibrium situation is Pareto optimal, that is to say that there is no alternative unanimously preferred by all agents. In other words, the allocation of resources associated with general equilibrium is such that one can not improve the situation of one agent without causing a deterioration in the situation of another.

The basic ideas that we have just presented are necessary for a good understanding of the structure of a CGE model and a correct interpretation of the results of analysis based on the results of such a model.

5.2 BASIC STRUCTURE OF A CGE MODEL FOR AN OPEN ECONOMY

For purely pedagogical reasons, we present here the most simple example of a CGE model. This version is similar to that studied by de Melo and Robinson (1989). The only difference is that we introduce the government explicitly.

We assume that the country is small in terms of international trade. This means that it can not influence the terms of trade with the rest of the world. We distinguish four categories of economic agents and three categories of goods. The three categories of goods are: a local good, (sold on the local market and abroad), a foreign good and a composite good.

Agents consist of a producer who seeks to maximize receipts from the sales of the local good on the two markets (domestic and foreign); a household that tries to maximize consumption of the composite good (that is, a combination of the local good and the imported good); government whose budgetary receipts derive from a tax on production and customs duties; and the rest of the world which buys the country's exports and sells it imports. To simplify things, we hypothesize that a portion of the fiscal receipts are paid to the producer in the form of an export subsidy and the rest is transferred to the consumer. In addition, the trade deficit has an offsetting inflow of foreign capital, and denominated in domestic currency, is paid to the household sector which spends it.

The basic structure of this economy is best shown using a social accounting matrix (SAM). A social accounting matrix is somewhat like a comprehensive economic table that generalizes the principles of national accounting which are at the foundation of the input-output table of inter-industry linkages. A social accounting matrix is a collection of economic accounts describing the totality of exchange flows between economic agents for a given year. The principles of optimization and equilibrium govern the behavior of agents and their interaction imposes an internal consistency to this matrix. Each account is represented by a line and a column with the same title. The receipts of an account are read off of the lines and the payments from the columns. Since all the accounts must maintain equilibrium between resources and uses, it follows that the SAM obeys Walras' Law. If *(n-1)* accounts are balanced, then the *nth* will be balanced as well.

Each account represents either a market or an agent. For our small economy, we distinguish five accounts: an "Activity" account for the producer, a "Goods" account representing the market for goods produced locally and abroad, a "Household" account for the consumer, and "Government" account, and a "World" account for the rest of the world which is exogenous to our system. The basic structure is presented in table 5.1.

TABLE 5.1. Structure of a Social Accounting Matrix

	Activity	Goods	Household	Government	World	Total
Activity		Supply of Domestic Goods		Export Subsidies	Exports	Total Receipts
Goods			Household Income			Total Demand
Household	GDP at Factor Cost			Fiscal Transfers	Trade Balance	Household Revenues
Government	Indirect Taxes	Custom Duties				Government Revenues
World		Imports				Imports
Total	Domestic Supply	Absorption	Household Expenditures	Public Expenditures	Foreign Savings	

It should be noted that this structure abstracts from intermediate consumption, markets for factors of production (labor, capital, land), direct taxes, public expenditures on products, and financial markets (savings and investment). But it suffices for purposes of illustrating the specification of a CGE in an open economy.

The specification begins with a symbolic representation of the basic structure, shown in table 5.2.

Table 5.2: Symbolic Representation of a Social Accounting Matrix

	ACTIVITY	GOODS	HOUSEHOLD	GOVERNMENT	WORLD
ACTIVITY		$P^t.D^D$		T^e	$R.pw^e.E$
GOODS			$P^q.Q^D$		
HOUSEHOLD	$P^x.X$			GR	$R.B$
GOVERNMENT	$t^a.P^d.D^s$	$t^m.R.pw^m.M$			
WORLD		$R.pw^m.M$			
TOTAL	$PIB+T^a$	$P^q.Q^s$	$P^q.Q^D$	$GR+T^e$	

where: $P^x.X = P^e.E+P^a.D^s$, $P^q.Q^s = P^m.M+P^t.D^D$ and $T^e = t^e.R.pw^e.E$

The meanings of the symbols are as follows:

X	:	Real GDP
E	:	Export volume
M	:	Import volume
D^s	:	Domestic good supply
D^D	:	Domestic good demand
Q^s	:	Composite good supply
Q^D	:	Composite good demand
P^e	:	Domestic price of exports
P^m	:	Domestic price of imports
P^d	:	Before tax price of domestic good
P^t	:	Post tax price of domestic good
P^x	:	GDP deflator
P^q	:	Consumer price index
R	:	Nominal exchange rate
GR	:	Fiscal receipts net of subsidies
T^e	:	Export subsidies
t^e	:	Export subsidy rate
t^m	:	Import tariff
t^a	:	Indirect tax rate
pw^e	:	World export price
pw^m	:	World import price
B	:	Trade balance

5.3 MATHEMATICAL FORMAT AND GAMS PROGRAM CODE

The general equilibrium model associated with table 5.2 is merely a system of equations presumed to explain the flows described above. The equations may be classified in at least five categories according to whether they describe the system of prices, quantities, incomes, expenditures and equilibrium conditions.

5.3.1 Prices

The system of prices is described by the following block of equations.

$$P^m = pw^m(1+t^m)\ R \tag{13}$$

$$P^e = pw^e(1+t^e)\ R \tag{14}$$

$$P^t = P^d(1+t^d) \tag{15}$$

$$P^q = (P^t.D^D+P^m.M)\ /\ Q \tag{16}$$

$$P^x = (P^d.D^s+P^e.E)\ /\ X \tag{17}$$

In GAMS, the same equations are written as follows:

```
PMD =E= PWM*(1+TM)*ER ;                (16)

PED =E= PWE*(1+TE)*ER ;                (17)

PDT =E= PDD*(1+TD)  ;                  (20)

PQ*Q =E= PDT*DD+PMD*M ;                (21)

PX =E= (PED*E+PDD*DS)/X ;              (22)
```

All the symbols are explained in the appendix. The first two equations describe the domestic prices of tradable goods (imports and exports) and reflect tariff policy (or trade policy in general) of the country. The third equation defines the price, tax inclusive, of the composite good. The last two define two general price indices, one for consumption and the other for production.

5.3.2 Quantities

The four following equations refer to the quantity variables in the model:

$$X = A(\alpha E^{h} + (1-\alpha) D_s^{h})^{\frac{1}{h}} \tag{23}$$

$$Q = B(\beta M^{-\rho} + (1-\beta) D_D^{-\rho})^{\frac{-1}{\rho}} \tag{24}$$

$$\frac{E}{D_s} = \left[\frac{P^{e}(1-\alpha)}{P^{d}\alpha}\right]^{\Omega} \tag{25}$$

$$\frac{M}{D_D} = \left[\frac{P^{c}\beta}{(1-\beta) P^{m}}\right]^{\sigma} \tag{26}$$

where $\Omega = \frac{1}{h-1}$ *and* $\sigma = \frac{1}{1+}$

In GAMS, the same equations are written as follows:

```
X =E= A*(((alpha*(E**h))+((1-alpha)*(DS**h)))**(1/h)) ; (27)
```

$$Q \;=E=\; B*(((beta*(M**(-rho)))+((1-beta)*DD**(-rho))))**(-\frac{1}{rho})) \;; \qquad (28)$$

$$E \,/\, DS \;=E=\; ((PED \,/\, PDD)*((1-alpha) \,/\, alpha))**omega \;; \qquad (29)$$

$$M \,/\, DD \;=E=\; ((PDT \,/\, PMD)*(beta \,/\, (1-beta)))**sigma \;; \qquad (30)$$

In fact, Equations 25 - 28 describe the behavior of the two private sector agents: the producer and the consumer. Equation 25 represents the first order conditions for producer income maximization subject to technological constraints described by the product transformation function (Equation 23) and given the incentive structure represented by the price block. The product transformation function is of the constant elasticity of transformation (CET) variety and is homogeneous to degree 1.

In the same manner, if the consumer maximizes the utility of his/her consumption of composite good Q, the optimal ratio of demand for the imported good relative to that of the local good is given by Equation 26. This ratio is a function of relative prices insofar as the composite good is a constant elasticity of substitution (CES) aggregation of the domestic and imported goods. This function is also homogeneous to degree 1.

5.3.3 Incomes

Government and household revenues are respectively defined as:

$$GR = t^{d}.P^{d}.D^{s}+t^{m}.R.pw^{m}.M-t^{e}.R.pw^{e}.E \qquad (29)$$

$$Y = P^{x}.X + R.B + GR \qquad (30)$$

Which are written in GAMS as:

$$GR \;=E=\; (TD*PDD*DS)+(TM*ER*PWM*M)-(TE*ER*PWE*E) \;; \qquad (31)$$

$$Y\ {=}E{=}\ (PX*X)\ +\ (ER*B)\ +\ GR\ ; \tag{32}$$

Household expenses are equal to $P^{q}.Q$. The last block of equations describe the general equilibrium conditions and the associated constraints.

5.3.4 Equilibrium Conditions

$$X = \overline{X} \tag{33}$$

$$D^{D} = D^{s} \tag{34}$$

$$pw^{m}.M - pw^{e}.E = B \tag{35}$$

The equivalent expressions are in GAMS are:

$$X\ {=}E{=}\ \overline{X}; \tag{36}$$

$$DD\ {=}E{=}\ DS\ ; \tag{37}$$

$$(PWM*M)\ -\ (PWE*E)\ {=}E{=}\ B\ ; \tag{38}$$

Equation 33 expresses a situation of full employment as X represents real maximum GDP achievable given resources and technological availabilities.

Equation 35 describes the trade balance: the difference between imports and exports valued at world prices. This equation may be considered as a description of the foreign exchange market. If B is fixed *a priori*, then equilibrium is established through movements in the nominal exchange rate which affects the relative prices of imports and exports.

The demand function for imports and the supply function for exports determines the levels of the two variables so that they yield the exogenous level of the trade deficit. In fact, the model determines a stable relationship between the exchange rate and the trade balance[3].

The system thus far described is homogeneous to degree zero relative to prices since economic agents base their decisions on relative prices. There are more variables than there are equations: 21 variables as against 14 equations. It is evident that not all the variables can be endogenous. We must therefore make 7 variables exogenous. This decision is made empirically, guided by the issues that are to be studied with the model and by the real functioning of the system under consideration.

In the framework of the application that we are going to study next, the "small country" assumption implies that world export and import prices (pw^e and pw^m) are exogenous. Moreover, we have imposed a constraint on the balance of payments, rendering B exogenous. Composite good Q was chosen as a numéraire and consequently its price is fixed as 1. Finally, the three instruments of fiscal policy are exogenous. Thus the fourteen remaining variables can be determined by the fourteen equations of the system. This choice of exogenous variables falls under the heading of the "closure" and can change the role of various equations. Thus for example, the fact that we chose good Q as a numéraire enables us to use Equation 16 to determine the price of the domestic good. Equation 17 can serve to determine the domestic price of exports since the growth rate is determined by Equation 14. The rest of the equations that contain the equilibrium conditions can determine the remaining unknown variables.

In other respects, the assumption of optimization used to explain economic behavior imposes a dual relationship between Equation 16 (P^q) and Equation 24 (Q) on the one hand, and Equation 17 (p^x) and Equation 23 (X), on the other hand. In certain instances, it may be easier to include the dual price equations and drop the CES and CET functions which describe the composite quantities. In such a situation, variable Q will be determined by Equation 16 while Equation 17 determines variable X. This demonstrates that the role of equations in the system is not rigid. It essentially revolves around the choice of exogenous variables or the model closure.

5.4 APPLICATION

We will continue our discussion based on the fictitious economy whose structure is described by the SAM presented above. The data correspond to the numerical example discussed in de Melo and Robinson (1989).

[3] Devarajan, Lewis and Robinson 1991b.

Table 5.3 - A Social Accounting Matrix for a Fictitious Economy: Base Year without Government Transactions

	ACTIVITY	GOODS	HOUSEHOLD	GOVERNMENT	WORLD	TOTAL
ACTIVITY		75			25	100
GOODS			100			100
HOUSEHOLD	100					100
GOVERNMENT						0
WORLD		25				25
TOTAL	100	100	100	0	25	

It should be noted that the government account is empty. This assumes that all tax rates are set at zero for the base year (after all, this is a fictitious economy).

We propose to study in the framework of this economy and with the aid of a GAMS computable general equilibrium model the effects of external shocks (inflow of foreign capital and a deterioration of the terms of trade), and economic policies. We examine, in turn, the role of calibration of the model, the case of capital inflows, the deterioration of the terms of trade and an increase in tariffs.

5.4.1 Calibration

A general equilibrium model becomes computable when all parameters and exogenous variables assume numerical values that are consistent with the base data set. The model is computable insofar as it is capable of generating numerical solutions. If the model is able to reproduce the initial SAM based on the data then the process of calibration has been accomplished. In other words, the numerical values assigned to the parameters are consistent with the initial structure of the economy as represented by the SAM.

In our case, the parameters of interest are the elasticities of substitution and transformation (σ *and* Ω, the exponents in the CES and CET functions (ρ *and* h) , the share parameters (α *and* β), and the scaling factors (A and B). In fact all these parameters are associated with the CES and CET functions and affect the behavior of economic agents.

As concerns the substitution and transformation elasticities, following from the assumption that locally sold goods are imperfect substitutes for goods sold in world markets, we have arbitrarily set these elasticities as follows:

$\sigma = 0.2$ *and* $\Omega = 0.2$

In GAMS we write:

SIGMA = 0.2; OMEGA = 0.2

By definition the exponents ρ *and* h are calculated as follows:

$$\rho = \frac{1}{\sigma} - 1 \qquad \text{or rho = (1/SIGMA) - 1;}$$

and $$h = \frac{1}{\Omega} + 1 \qquad \text{or} \quad \text{h = (1/OMEGA) + 1.}$$

The share parameter α, in the CET function is calculated from Equation 25, replacing all variables by their initial values (base year values). Raising both sides of the equation to the power of $\frac{1}{\Omega}$, we obtain the following expression:

$$(E_o \;/\; D_o)^{\frac{1}{\Omega}} = (1-\alpha)P_o^e \;/\; \alpha P_o^d \tag{41}$$

The index "O" represents the base year. Multiplying both sides by P^d_0/P^e_0 we obtain:

$$(1-\alpha) \;/\; \alpha = (P_o^d \;/\; P_o^e) \; (E_o \;/\; D_o)^{\frac{1}{\Omega}}$$

This implies that:

$$(1-\alpha) = \alpha(P_o^d \;/\; P_o^e) \; (E_o \;/\; D_o)^{\frac{1}{\Omega}}$$

where $$\alpha = \frac{1}{(\frac{P_o^d}{P_o^e}) \cdot (\frac{E_o}{D_o})^{\frac{1}{\Omega}} + 1}$$ and $$(1-\alpha) = \frac{(P_o^d \;/\; P_o^e)\,(E_o \;/\; D_o)^{\frac{1}{\Omega}}}{(P_o^d \;/\; P_o^e)\,(E_o \;/\; D_o)^{\frac{1}{\Omega}} + 1}$$

This is written in GAMS as:

ALPHA = 1/((PDDO/PE$_0$) * (E$_0$/D$_0$) ** (1/OMEGA) + 1;

using the same reasoning applied to Equation 26 leads us to the following expression for β:

$$\beta = \frac{1}{((\frac{P_o^m}{P_o^d}) . (\frac{M_o}{D_o})^{\frac{1}{\sigma}} + 1)}$$

In GAMS we write:

BETA = 1/((PMO/PDDO) * (M_0/D_0) ** (1/SIGMA) + 1;

Finally, the scaling factors A and B are determined from Equation 23 and 24, respectively. We leave the derivation to the reader.

Before continuing on to the numerical results, it is useful to emphasize that the calculation must occur in the following order for the CET and CES functions: elasticity, exponent, share parameter, and finally scaling factor. This follows from the fact that the exponents and share parameters depend directly on the elasticities while the scaling factors depend on both the share parameters and the exponents.

The input file that we used to reproduce the base solution is titled DMR89.GMS. The corresponding output file is in the appendix. The careful reader will notice that the following information appears at end of the output file.

```
*** FILE SUMMARY

     INPUT     C:\GAMS386\LAB\DMR89.GMS
     OUTPUT    C:\GAMS386\LAB\DMR89.LST
     SAVE      C:\GAMS386\LAB\DUTCH.G0?
```

This signifies that the input file is named DMR89.GMS; the output file is DMR89.LST and the work files are specially saved in a series of files with the root name of DUTCH (we provide this information to GAMS) and the extensions run from .G01 to G08. The command that produces these results is :

```
     GAMS      DMR89     S=DUTCH
```

This command is entered in DOS after saving the input file.

We asked GAMS to save the work files (scratch files) because comparative static analysis requires that the model be solved several times and that each new solution be compared to the base solution. The ability to save the scratch files means that GAMS can calculate the new solution more efficiently by referring to the saved solution file instead of having to recalculate it.

5.4.2 Dutch Disease

To study the structural effects of a foreign capital inflow in our small economy, we created a new input file DMR89DU.GMS. The structure is the following. The first three lines give the title of the simulation in question, change the level of the trade balance and solve the model again. They are:

```
$TITLE DUTCH DISEASE
BBAR = 10.00;
SOLVE GSS USING NLP MAXIMIZING Q;
```

In fact, the last two commands are sufficient for GAMS to simulate the syndrome of Dutch Disease provided that we had taken the precaution of asking GAMS to save the scratch files from the base solution.

In our case, we added commands which update various variables and parameters. Since we had previously arbitrarily fixed the trade elasticities, we have included some lines which enable us to conduct sensitivity analysis based on different values for the elasticities. The new input file is submitted to GAMS using the following command:

```
GAMS      DMR89DU   R=DUTCH
```

The specification "R = DUTCH" tells GAMS where it can find the base solution files required to solve the new simulation. The results are contained in the following table.

Table 5.4 - Effects of Foreign Capital Inflows

Elasticities												
σ	Ω	*Q*	P^d	*TCR*	*TCRE*	*TCRM*	*TCERQ*	*TCERX*	*TCN*	*E*	*DD*	*M*
0.2	0.2	106.97	1.20	0.38	0.38	0.38	0.46	0.45	0.46	21.28	77.39	31.28
0.5	0.5	108.66	1.09	0.68	0.68	0.68	0.75	0.74	0.75	21.46	77.95	31.46
2.0	2.0	109.64	1.02	0.91	0.91	0.91	0.93	0.93	0.93	21.57	78.28	31.57
5.0	5.0	109.86	1.01	0.96	0.96	0.96	0.97	0.97	0.97	21.59	78.35	31.59
15.0	0.2	109.92	1.01	0.98	0.98	0.98	0.98	0.98	0.98	24.92	75.08	34.92
0.2	15.0	109.89	1.01	0.97	0.97	0.97	0.98	0.98	0.98	17.60	82.30	27.6

TCR :Real exchange rate
TCRE :Real export exchange rate
TCRM :Real import exchange rate
TCERQ :Real effective exchange rate for consumption
TCERX :Real effective exchange rate for production
TCN :Nominal exchange rate = R = TCERQ by choice of numéraire.

The following observations shed some light on the results presented in table 5.4. In general, an influx of foreign capital leads to an appreciation of the real exchange rate, defined as the relative price of tradable and non-tradable goods. In this context, an appreciation means an increase in the price of the domestic good, P^d. Structural changes will cause changes in the real exchange rate, as the variable that encapsulates the structure of the incentive system. Thus, on the consumption side, we observe a growth in demand for the domestic good and for imports. By contrast, on the production side, relative prices are unfavorable for the export sector, resulting in a shift in resources towards the more profitable domestic sector. This is the Dutch Disease.

Now the question arises if this phenomenon repeats itself in the sensitivity analysis. The magnitude of the appreciation of the real exchange rate as well as the associated structural shifts depend, fundamentally, on the elasticities of substitution and transformation. From table 5.4, we may observe that the real exchange rate appreciation is much larger the smaller is the elasticity of substitution. When the elasticity is 0.2, a capital inflow of 10 units leads to a 20 percent growth in domestic prices. By contrast, when (σ) = 15, the same shock only results in a 1 percent growth in domestic prices. In this case, the production of exports only falls by 0.32 percent as compared to 15 percent in the first case.

In practice, one is confronted with the choice of the price index for estimating P^d. There is the choice between the consumer price index, represented here by P^q, and the GDP deflator, represented by P^x. By adopting one or the other, the exchange rate thus calculated is called the real effective exchange rate. The results above show that these two exchange rates are biassed and the bias disappears when the elasticities of substitution and transformation tend to infinity. Equation 17, equivalent to:

$$P^x = P^d . \frac{D^s}{X} + P^e . \frac{E}{X}$$

suggests a method for constructing an index for P^d from national accounts data[4]. In fact, given S^e, the share of exports in real GDP,

$$S^e = \frac{E}{X} \text{ and } (1-S^e) = \frac{D_s}{X}$$

where $P^d = \dfrac{P^x - S^e . P^e}{1-S^e}$.

[4] Devarajan, Lewis and Robinson (1991a).

5.4.3 Deterioration in the Terms of Trade

We created an input file similar to that created to the Dutch Disease file in order to simulate a deterioration in the terms of trade. The file is called DMR89WPM.GMS. We increased the world price of imports by 10 percent using the command: PWM = 1.10;

The results of the new solution are presented in table 5.5. In view of these results the following comments may be made.

The real export exchange rate (TCRE) is no longer equal to the real import exchange rate (TCRM) as it was in the preceding case. The observed equality in table 5.4 resulted from the equality in the two world prices. One also sees that a general depreciation in the real import exchange rate indicates that the domestic good has become cheaper relative to the imported good. One observes a fall in imports which is greater the larger is the elasticity of substitution (σ). In all the cases where this elasticity is less than unity, one also sees a depreciation in the real export exchange rate associated with an increase in exports. It is necessary to sell more to the reset of the world in order to obtain the means to pay for more expensive imports. By contrast, when the elasticity of substitution is greater than unity, we observe another type of adjustment. There is an appreciation in the exchange rate and the export sector contracts to the benefit of the domestic sector.

The fundamental mechanism explaining these results encompasses two effects as regards the deterioration in the terms of trade. It is useful to remember that a change in the terms of trade is in fact a change in relative prices.

With every change in prices there is an associated income effect and a substitution effect. The final outcome depends on which effect dominates. When the substitution elasticity is less than one, the income effect dominates. The imported and domestic goods are complementary. When imports become more costly, consumption of both imported and domestic goods fall. The contraction of the domestic sector frees up resources which are drawn to the export sector. By contrast, when the two goods are substitutes, a deterioration in the terms of trade provokes an appreciation of the real exchange rate, followed by a growth in the domestic sector at the expense of the export sector.

5.4.4 Customs Tariffs

Recall that the base solution was obtained with all taxes fixed at zero. To study the effects of an increase in the tariff by 10 percent, it is only necessary to copy the initial program into another input file, DMR89TAR.GMS for example. In this file, it is necessary to change the closure by changing the command: TM.FX = 0; by TM.FX = 0.10; the file can also be restructured to allow an examination of the role of elasticities. The procedure is analogous to that which we just followed. To minimize the costs associated with the production of this report, we leave it to the initiative of the reader to pursue the study of this case.

Table 5.5 Deterioration in the Terms of Trade

σ	Ω	*Q*	P^d	*TCR*	*TCRE*	*TCRM*	*TCERQ*	*TCERX*	*TCN*	*E*	*DD*	*M*
0.2	0.2	97.44	0.93	1.21	1.21	1.33	1.12	1.15	1.12	25.70	74.23	23.37
0.5	0.5	97.58	0.96	1.05	1.05	1.15	1.01	1.05	1.01	25.45	74.54	23.13
2.0	2.0	97.71	0.98	0.98	0.98	1.07	0.96	0.98	0.96	24.11	75.87	21.92
5.0	5.0	97.84	0.99	0.96	0.96	1.06	0.95	0.97	0.95	21.58	78.35	19.62
15.0	0.2	97.14	1.00	0.92	0.92	1.01	0.91	0.94	0.91	24.67	75.32	22.43
0.2	15.0	97.57	0.97	1.01	1.01	1.11	0.98	1.00	0.98	26.44	73.56	24.03

6. CONCLUSION

As we said in the beginning, this document represents an introductory course to the GAMS program for economic modeling.

This program facilitates the specification and solving of mathematical programs. In fact, the logic of GAMS programming is suited to optimization problems. A GAMS program is a collection of instructions. The distinction is made between declaration instructions and/or definition of execution commands. The first group serves to describe the class of symbols (indices, parameters, variables, equations, models, etc..). The second group, in contrast, executes tasks such as the processing of raw data, the solution of models, and the production of reports. The basic structure of the statements in the first group includes a key word followed by an identifier; the identifier is eventually followed by its definition. Commands in the second group also begin with a key word followed by an identifier and an instruction describing the task to be executed.

The program presents some advantages for economic modeling. It is compact, "transparent" and based on the fundamental principle of economics: optimization. It is transparent because the GAMS expressions are very similar to the actual algebraic expressions of the model.

Economic models are constructed to explain and predict socio-economic phenomena. The examples that we used to illustrate GAMS brought out a fundamental idea of modeling. The specification of a model is based partly on economic theory and partly on the stylized facts of the structure of the socio-economic structure of the country under consideration (if the model concerns and entire economy). From this, the consequence and thus the interpretation of the results from the model are determined by its structure. It is for this reason that it is very important that an economic model be as close as possible as the entity that it represents. Otherwise, the economic policy recommendations based on such a model will be of limited value.

BIBLIOGRAPHY

Bisschop, Johannes and Alexander Meeraus. 1982. On the Development of a General Algebraic Modeling System in a Strategic Planning Environment. World Bank Reprint Series: Number 308. Washington, D.C.: The World Bank.

Bousselmi, N., B. Decaluwé, A. Martens and M. Monette. 1989. Développement Socio-économique et Modèles Calculables d'Equilibre Général: Méthodes et Application à la Tunisie. Tunis: Ministère du Plan et des Finances and Université de Montréal.

Brooke, A., D. Kendrick and A. Meeraus. 1988. GAMS: A User's Guide. Redwood City, CA: The Scientific Press.

Bulmer-Thomas, V. 1982. Input-Output Analysis in Developing Countries. New York: John Wiley and Sons.

de Melo, Jaime and Sherman Robinson. 1989. Product Differentiation and the Treatment of Foreign Trade in Computable General Equilibrium Models of Small Economies. Journal of International Economics. vol. 27: 47-67.

Dervis, Kemal, Jaime de Melo and Sherman Robinson. 1979. Les Modèles d'Equilibre Général Calculable et le Commerce International, in Economie Appliquée Archives of the Institut de Sciences Mathématiques et Economiques Appliquées. Vol. XXXII, No. 4. DROZ - Geneva.

Dervis, Kemal, Jaime de Melo and Sherman Robinson. 1982. General Equilibrium Models for Development Policy. Cambridge, UK: Cambridge University Press.

Devarajan, Shantayanan. 1988. Lecture Notes on Computable General Equilibrium Models. Cambridge, MA: J.F.K. School of Government. Harvard University. Photocopy.

Devarajan, S., J.D.Lewis and S. Robinson. 1990. Policy Lessons from Trade Focused Two-sector Models. Journal of Policy Modeling. 12(4): 625-657.

Devarajan, S., J.D.Lewis and S. Robinson. 1991a. External Shocks, Purchasing Power Parity and the Equilibrium Real Exchange Rate. Berkeley, CA: Department of Agricultural and Resource Economics. Photocopy.

Devarajan, S., J.D.Lewis and S. Robinson. 1991b. From Stylized to Applied Models: Building Multisectoral CGE Models for Policy Analysis. Washington, D.C.: World Bank and Harvard University. Photocopy.

Hugon, Philippe. 1990. Aspect Méthodologiques et Comparatifs des Programmes d'Ajustement Structurel: Le Cas du Niger. Niamey, Niger: Ministère du Plan and UNDP. Project NER/88/014.

Jefferson, R.W. and R.N. Boisvert. **1989.** A Guide to Using the General Algebraic Modeling System (GAMS) for Applications in Agricultural Economics. Ithaca, New York: Department of Agricultural Economics, Cornell University.

Miller, R.E. and Peter D. Blair. **1985.** Input-Output Analysis: Foundations and Extensions. Englewood Cliffs, N.J.: Prentice-Hall.

Robinson, S., M. Kilkenny and K. Hanson. **1990.** The USDA/ERS Computable General Equilibrium (CGE) Model of the United States. Staff Report No. AGES 9049. Washington, D.C.: Economic Research Service, U.S. Department of Agriculture.

Suwa, Akiko. 1991. Les Modèles d'Equilibre Général Calculable. Economie et Prévision No. 97: 69-76.

Varian, Hal R. 1984. Microeconomic Analysis. New York: Norton.

ANNEXE

```
GAMS 2.20 DOS-386                              .    91/11/01 08:15:48   PAGE       1
AN EXAMPLE OF LINEAR PROGRAMMING IN GAMS



   2
   3   SETS
   4      I  INDICE DES RESSOURCES
   5          /RES1*RES3/
   6      J   INDICE DES PRODUITS
   7          /BIEN1  PREMIER PRODUIT
   8           BIEN2  DEUXIEME PRODUIT
   9           BIEN3  TROISIEME PRODUIT/
  10
  11   PARAMETERS
  12      B(I) DISPONIBILITES EN RESSOURCES
  13          /RES1 10, RES2 6, RES3 8/
  14      C(J) PRIX UNITAIRES PAR PRODUIT
  15          /BIEN1 3, BIEN2 2, BIEN3 5/
  16   TABLE
  17      A(I,J) MATRICE DES COEFFICIENTS TECHNIQUES
  18
  19               BIEN1       BIEN2       BIFN3
  20
  21     RES1       3            1
  22     RES2                    1           2
  23     RES3       2            1           1
  24
  25   VARIABLES
  26      R          REVENU TOTAL
  27      X(J)       NIVEAUX DE PRODUCTION
  28
  29   POSITIVE VARIABLES X ;
  30
  31   EQUATIONS
  32      REVENU   FONCTION OBJECTIVE
  33      CONTR(I) CONTRAINTES ;
  34
  35    REVENU ..   SUM(J, C(J)*X(J)) =E= R;
  36    CONTR(I) ..  SUM(J, A(I,J)*X(J)) =L= B(I);
  37
  38   MODEL REV Revenu d'Exploitation   / ALL /;
  39
  40   SOLVE REV MAXIMIZING R  USING LP;
  41
  42   DISPLAY  X.L , X.M,CONTR.L,CONTR.M ;

GAMS 2.20 DOS-386                                   91/11/01 08:15:48   PAGE       2
AN EXAMPLE OF LINEAR PROGAMMING IN GAMS
SYMBOL LISTING


SYMBOL       TYPE   REFERENCES

A            PARAM  DECLARED        17  DEFINED        17        REF        36
B            PARAM  DECLARED        12  DEFINED        13        REF        36
C            PARAM  DECLARED        14  DEFINED        15        REF        35
CONTR        EQU    DECLARED        33  DEFINED        36 IMPL-ASN        40
                         REF        38      2*42
I            SET    DECLARED         4  DEFINED         5        REF        12
                          17        33      2*36  CONTROL       36
J            SET    DECLARED         6  DEFINED         7        REF        14
                          17        27      2*35     2*36  CONTROL        35
                          36
R            VAR    DECLARED        26 IMPL-ASN        40        REF        35
                          40
REV          MODEL  DECLARED        38  DEFINED        38 IMPL-ASN        40
                         REF        40
REVENU       EQU    DECLARED        32  DEFINED        35 IMPL-ASN        40
                         REF        38
X            VAR    DECLARED        27 IMPL-ASN        40        REF        29
                          35        36      2*42
```

```
SETS

I           INDICE DES RESSOURCES
J           INDICE DES PRODUITS


PARAMETERS

A           MATRICE DES COEFFICIENTS TECHNIQUES
B           DISPONIBILITES EN RESSOURCES
C           PRIX UNITAIRES PAR PRODUIT


VARIABLES

R           REVENU TOTAL
X           NIVEAUX DE PRODUCTION



EQUATIONS

CONTR       CONTRAINTES
REVENU      FONCTION OBJECTIVE


SYMBOL LISTING

MODELS

REV

COMPILATION TIME     =       11.200 SECONDS        VER: 386-SP-005

EQUATION LISTING    SOLVE REV USING LP FROM LINE 40

---- REVENU       =E=  FONCTION OBJECTIVE

REVENU..  - R + 3*X(BIEN1) + 2*X(BIEN2) + 5*X(BIEN3) =E= 0 ;

---- CONTR        =L=  CONTRAINTES

CONTR(RES1)..  3*X(BIEN1) + X(BIEN2) =L= 10 ;

CONTR(RES2)..  X(BIEN2) + 2*X(BIEN3) =L= 6 ;

CONTR(RES3)..  2*X(BIEN1) + X(BIEN2) + X(BIEN3) =L= 8 ;

COLUMN LISTING      SOLVE REV USING LP FROM LINE 40

---- R           REVENU TOTAL

R
                (.LO, .L, .UP =  -INF, 0,  +INF)
       -1       REVENU

---- X           NIVEAUX DE PRODUCTION

X(BIEN1)
                (.LO, .L, .UP = 0, 0,  +INF)
        3       REVENU
        3       CONTR(RES1)
        2       CONTR(RES3)



X(BIEN2)
```

```
                (.LO, .L, .UP = 0, 0,  +INF)
        2       REVENU
        1       CONTR(RES1)
        1       CONTR(RES2)
        1       CONTR(RES3)

X(BIEN3)
                (.LO, .L, .UP = 0, 0,  +INF)
        5       REVENU
        2       CONTR(RES2)
        1       CONTR(RES3)

MODEL STATISTICS    SOLVE REV USING LP FROM LINE 40
MODEL STATISTICS

BLOCKS OF EQUATIONS     2     SINGLE EQUATIONS      4
BLOCKS OF VARIABLES     2     SINGLE VARIABLES      4
NON ZERO ELEMENTS      11
GENERATION TIME      =        2.470 SECONDS
EXECUTION TIME       =        5.870 SECONDS        VER: 386-SP-005

SOLUTION REPORT     SOLVE REV USING LP FROM LINE 40

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

     MODEL   REV                  OBJECTIVE  R
     TYPE    LP                   DIRECTION  MAXIMIZE
     SOLVER  BDMLP                FROM LINE  40

**** SOLVER STATUS      1 NORMAL COMPLETION
**** MODEL STATUS       1 OPTIMAL
**** OBJECTIVE VALUE                22.5000

 RESOURCE USAGE, LIMIT          0.000      1000.000
 ITERATION COUNT, LIMIT         3          1000

 BDM - LP  VERSION 1.01 386-005
 A. Brooke, A. Drud, and A. Meeraus,
 GAMS Development Corporation,
 2828 Albemarle Street NW,
 Washington, D.C. 20008, U.S.A.
 Work space needed (estimate)   --    3820 words.
 Work space available           --    3820 words.

 EXIT -- OPTIMAL SOLUTION FOUND.

                        LOWER     LEVEL     UPPER    MARGINAL

---- EQU REVENU           .         .         .        -1.000

  REVENU      FONCTION OBJECTIVE

---- EQU CONTR       CONTRAINTES

        LOWER     LEVEL     UPPER    MARGINAL

RES1     -INF     7.500    10.000       .
RES2     -INF     6.000     6.000     1.750
RES3     -INF     8.000     8.000     1.500

                        LOWER     LEVEL     UPPER    MARGINAL

---- VAR R               -INF     22.500     +INF        .

  R           REVENU TOTAL

SOLUTION REPORT     SOLVE REV USING LP FROM LINE 40
```

```
---- VAR X            NIVEAUX DE PRODUCTION

         LOWER      LEVEL      UPPER    MARGINAL

BIEN1      .        2.500       +INF       .
BIEN2      .          .         +INF     -1.250
BIEN3      .        3.000       +INF       .


**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED

E X E C U T I N G


----     42 VARIABLE  X.L           NIVEAUX DE PRODUCTION

BIEN1 2.500,    BIEN3 3.000


----     42 VARIABLE  X.M           NIVEAUX DE PRODUCTION

BIEN2 -1.250
----     42 EQUATION  CONTR.L       CONTRAINTES

RES1 7.500,    RES2 6.000,    RES3 8.000


----     42 EQUATION  CONTR.M       CONTRAINTES

RES2 1.750,    RES3 1.500


**** FILE SUMMARY

INPUT    C:\GAMS386\LAB\EXEMPLE1.GMS
OUTPUT   C:\GAMS386\LAB\EXEMPLE1.LST
SAVE     C:\GAMS386\LAB\EXEMPLE.G0?

EXECUTION TIME       =         4.780 SECONDS         VER: 386-SP-005
```

```
GAMS 2.20 DOS-386                                  91/11/02 23:06:48   PAGE
OPERATIONS ON SETS

    3
    4  SET   S Secteurs d'activites
    5
    6
    7        /Agricult            Agriculture
    8         Elevage             Elevage
    9         Peche-for           Peche et foret
   10         Mines               Industries extractives
   11         Manufact            Industries manufacturieres
   12         Energie             Electricite et Eau
   13         BTP                 Batiment et Travaux Publics
   14         Commerce            Commerce Hotels Restaurants
   15         Transport           Transports et Telecommunications
   16         Services            Services Prives
   17         Administ            Administrations/
   18
   19        PR(s)   Secteur Primaire
   20            /Agricult,Elevage,Peche-for/
   21        SE(s)   Secteur Secondaire
   22            /Mines,Manufact,Energie,BTP/
   23        TE(s)   Secteur Tertiaire
   24            /Commerce,Transport,Services,Administ/
   25        UNI(s)  Union
   26        DIF(s)  Difference
   27        INTR(s) Intersection
   28        COMP(s) Complement
   29        TOUT(s) Ensemble Univers
   30        INTER(s)       Une nouvelle Intersection;
   31
   32  DISPLAY PR,SE,TE;
   33
   34  PARAMETER  N  Nombre d'elements dans S
   35             O(s)  Ordre des elements de S;
   36
   37
   38             N  = CARD(S) ;
   39             O(s)      = ORD(S) ;
   40
   41  DISPLAY N,O;
   42
   43
   44  PR("MINES")   = YES;
   45  SE("MINES")   = NO;
   46
   47  DISPLAY PR,SE;
   48
   49
   50    UNI(s)      = PR(s) + SE(s);
   51    DIF(s)      = UNI(s) - SE(s);
   52    INTR(s)     = SE(s)*TE(s);
   53    COMP(s)     = NOT TE(s);
   54    TOUT(s)     = YES;
   55    INTER(s)    = TOUT(s)*TE(s);
   56  DISPLAY  UNI,DIF,INTR,COMP,TOUT,INTER;
   57
   58   PR(S) = YES;
   59   SE(S) = NO;
   60   DISPLAY PR,SE;

COMPILATION TIME     =        0.440 SECONDS        VER: 386-SP-005


E X E C U T I N G

----      32 SET       PR              SECTEUR PRIMAIRE
```

```
AGRICULT,     ELEVAGE,     PECHE-FOR

----      32 SET        SE              SECTEUR SECONDAIRE

MINES,    MANUFACT,    ENERGIE,    BTP

----      32 SET        TE              SECTEUR TERTIAIRE

COMMERCE,    TRANSPORT,    SERVICES,    ADMINIST

----      41 PARAMETER N           =       11.000 NOMBRE D'ELEMENTS  DANS S
----      41 PARAMETER O                ORDRE DES ELEMENTS DE S

AGRICULT        1.000,       ELEVAGE        2.000, PECHE-FOR 3.000,   MINES      4.000
MANUFACT        5.000,       ENERGIE        6.000, BTP        7.000,   COMMERCE   8.000
TRANSPORT       9.000,       SERVICES      10.000, ADMINIST 11.000

----      47 SET        PR              SECTEUR PRIMAIRE

AGRICULT ,    ELEVAGE  ,    PECHE-FOR,    MINES

----      47 SET        SE              SECTEUR SECONDAIRE

MANUFACT,    ENERGIE ,    BTP

----      56 SET        UNI             UNION

AGRICULT,    ELEVAGE,     PECHE-FOR,   MINES,       MANUFACT,    ENERGIE      BTP
----      56 SET        DIF             DIFFERENCE

AGRICULT ,    ELEVAGE  ,    PECHE-FOR,    MINES

----      56 SET        INTR            INTERSECTION

                        (EMPTY)

----      56 SET        COMP            COMPLEMENT

AGRICULT,    ELEVAGE,     PECHE-FOR,   MINES,       MANUFACT,    ENERGIE,     BTP

----      56 SET        TOUT            ENSEMBLE UNIVERS

AGRICULT,    ELEVAGE,     PECHE-FOR,   MINES,       MANUFACT,   ENERGIE
BTP,         COMMERCE,    TRANSPORT,   SERVICES,    ADMINIST

----      56 SET        INTER           UNE NOUVELLE INTERSECTION

COMMERCE ,    TRANSPORT,    SERVICES ,    ADMINIST

----      60 SET        PR              SECTEUR PRIMAIRE

AGRICULT,    ELEVAGE,     PECHE-FOR,   MINES,       MANUFACT,   ENERGIE
BTP,         COMMERCE,    TRANSPORT,   SERVICES,    ADMINIST

----      60 SET        SE              SECTEUR SECONDAIRE

                        (EMPTY)


**** FILE SUMMARY

INPUT   C:\GAMS386\LAB\ENSEMBLE.GMS
OUTPUT  C:\GAMS386\LAB\ENSEMBLE.LST

EXECUTION TIME     =       0.930 SECONDS       VER: 386-SP-005
```

```
GAMS 2.20 DOS-386                           91/10/20 15:48:26  PAGE
APPLICATION OF THE LEONTIEF MODEL

*Source: Jefferson and Boisvert (1989) A Guide to Using the General
*        Algebraic Modelling System (GAMS).Department of Agricultural
*        Economics. Cornell University, Ithaca ,New York 14853
*Modifications: Essama Nssah. Cornell Policy Reform Project
*        DAEP/Minitere du Plan,Niamey, Niger

*#################### DECLARATION DES INDICES #######################
SETS
    i  Secteurs    /AGRICULT        Agriculture
                    MANUF           Industries manufacturieres
                    TRANSPORT       Transports
                    SERVICES        Services/

ALIAS (i,j,k)

*##################### DECLARATION DES PARAMETRES ##################

PARAMETERS

 UNO(i)          Vecteur artificiel
 X0(i)           Production sectorielle initiale
 VA0(i)          Valeur ajoutee
 CD0(i)          Consommation des menages
 AD0(i)          Autre demande finale
 INT0(i)         Demande intermediaire
 PV0(i)          Prix de la valeur ajoutee
 FD0(i)          Demande finale totale
 IDM(i,k)        Matrice unite de dimension CARD(i)
 DIPUR(j)        Cout unitaire intermediaire
 A(i,j)          Coefficients techniques;

*####################### SAISIE DES DONNEES ######################

 TABLE   IO(i,j) Transactions intersectorielles en unites  monetaires

                         AGRICULT     MANUF  TRANSPORT    SERVICES

       AGRICULT              34      290          0        0
       MANUF                 25     1134          5      201
       TRANSPORT              6      304         54      105
       SERVICES              48      962         71      877

 TABLE   SECTRES(*,j)   Donnees sectorielles  en unites monetaires
          AGRICULT              MANUF    TRANSPORT      SERVICES
 VA            356              11622          480          57140
 CD              7                607           22           2558
 AD            138              12340          119           2381
 X             469              14312          610           6897
                 ;

       X0(j)  = SECTRES("X",j) ;
       VA0(j) = SECTRES("VA",j) ;
       CD0(j) = SECTRES("CD",j);
       AD0(j) = SECTRES("AD",j) ;
       FD0(i) = CD0(i)+AD0(i) ;

       IDM(i,k) = 1$(ORD(i) EQ ORD(k));
       A(i,j) = IO(i,j)/X0(j) ;
       DIPUR(j)    = SUM(i,A(i,j));
       UNO(j)     = 1.00 ;
       PV0(j)     = UNO(j) - DIPUR(j);
       INT0(i)= SUM(j,A(i,j)*X0(j)) ;

VARIABLES
        GDP       PIB
        INV(j,k)  Inverse de la matrice de Leontief
 73          X(j)      Niveaux de production
 74
 75  POSITIVE VARIABLES X;
 76
 77  EQUATIONS
 78
 79          GDPEQ        Definition du PIB
 80          INVEQ(i,k)  Contraintes pour calcul de l'inverse
 81          MBEQ(i)      ERE par produit;
 82
 83          GDPEQ.. SUM(j,PV0(j)*X(j))=E= GDP ;
 84          INVEQ(i,k).. SUM(j,(IDM(i,j) - A(i,j))*INV(j,k))=E= IDM(i,k);
 85          MBEQ(i)..  SUM(j,(IDM(i,j) - A(i,j))*X(j)) =E=FDO(i);
 86
 87  MODEL
 88        LEONTIEF   Calcul de l'inverse de Leontief
 89       /GDPEQ,INVEQ, MBEQ/;
 90
 91  OPTION  LIMROW=0 ,LIMCOL=0;
 92  OPTION  SOLPRINT=OFF;
 93  SOLVE LEONTIEF  USING LP MAXIMIZING GDP;
 94
 95  PARAMETERS
 96          XX(i)        Verification calcul production
 97          LEON(j,k)    Inverse de Leontief
 98          CSM(j)               Total colonne
 99          RSM(i)               Total ligne
100          DENO                 Denominateur des indices de Rasmussen
101          B(j)                 Impact de la demande de j sur l'economie
102          F(i)                 Impact chang demande dns tous les sect sur sect i
103          RAP(j,*)             Rapport final
104              ;

107          LEON(j,k) = INV.L(j,k);
108          XX(j) = SUM(k,LEON(j,k)*FD0(k));
109          CSM(j) = SUM(i,LEON(i,j));
110          RSM(i) = SUM(j,LEON(i,j));
111          DENO =(1/(CARD(i) )**2)*(SUM((i,j),LEON(i,j)));
112          B(J) =( 1/(CARD(i))*CSM(j))/DENO;
113          F(I) =((1/CARD(i))*RSM(i))/DENO;
114   *  Rapport
115           RAP(j,"Activites")  =  XX(j);
116           RAP(j,"TotalCol")   =  CSM(j);
117           RAP(j,"RasmuBj")    =  B(j);
118           RAP(j,"TotaLign")   =  RSM(j);
119           RAP(j,"RasmuFi")    =  F(j);
120           RAP(j,"DemandeInt") =  INTO(j);
121           RAP(j,"DemandeFin") =  FD0(j);
122           RAP(j,"Cout-u-Int") =  DIPUR(j);
123           RAP(j,"Prix-VA")    =  PV0(j);
124  OPTION DECIMALS = 4;
125
126  DISPLAY  DENO, A,LEON,RAP ;
127
128
129
130
COMPILATION TIME      =         1.100 SECONDS         VER: 386-SP-005

MODEL STATISTICS    SOLVE LEONTIEF USING LP FROM LINE 93
MODEL STATISTICS

BLOCKS OF EQUATIONS      3     SINGLE EQUATIONS      21
BLOCKS OF VARIABLES      3     SINGLE VARIABLES      21
NON ZERO ELEMENTS       75

GENERATION TIME      =         1.430 SECONDS
EXECUTION TIME       =         4.450 SECONDS         VER: 386-SP-005
SOLUTION REPORT     SOLVE LEONTIEF USING LP FROM LINE 93
```

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

     MODEL   LEONTIEF            OBJECTIVE  GDP
     TYPE    LP                  DIRECTION  MAXIMIZE
     SOLVER  BDMLP               FROM LINE  93

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE           18172.0000

 RESOURCE USAGE, LIMIT          1.000      1000.000
 ITERATION COUNT, LIMIT        11          1000


 BDM - LP  VERSION 1.01 386-005

 A. Brooke, A. Drud, and A. Meeraus,
 GAMS Development Corporation,
 2828 Albemarle Street NW,
 Washington, D.C. 20008, U.S.A.

 Work space needed (estimate)    --     4917 words.
 Work space available            --     4917 words.

 EXIT -- OPTIMAL SOLUTION FOUND.


**** REPORT SUMMARY :        0      NONOPT
                             0      INFEASIBLE
                             0      UNBOUNDED

E X E C U T I N G

----     126 PARAMETER DENO                =       0.3127 DENOMINATEUR DES
                                                          INDICES DE RASMUSSEN

----     126 PARAMETER A              COEFFICIENTS TECHNIQUES

                    AGRICULT            MANUF         TRANSPORT    SERVICES

AGRICULT            0.0725              0.0203
MANUF               0.0533              0.0792        0.0082       0.0291
TRANSPORT           0.0128              0.0212        0.0885       0.0152
SERVICES            0.1023              0.0672        0.1164       0.1272

----     126 PARAMETER LEON           INVERSE DE LEONTIEF

                    AGRICULT            MANUF         TRANSPORT    SERVICES

AGRICULT            1.0796              0.0238        0.0003              0.0008
MANUF               0.0669              1.0905        0.0145              0.0367
TRANSPORT           0.0190              0.0273        1.0999              0.0201
SERVICES            0.1343              0.0904        0.1478              1.1513

----     126 PARAMETER RAP            RAPPORT FINAL

                    ACTIVITES    TOTALCOL      RASMUBJ      TOTALIGN            RASMUFI

AGRICULT              469.0000   1.2998        1.0392       1.1046              0.8831
MANUF               14312.0000   1.2320        0.9850       1.2086              0.9663
TRANSPORT             610.0000   1.2626        1.0094       1.1662              0.9324
SERVICES             6897.0000   1.2088        0.9665       1.5238              1.2182

             +      DEMANDEINT   DEMANDEFIN    COUT-U-INT     PRIX-VA
AGRICULT             324.0000      145.0000    0.2409              0.7591
MANUF               1365.0000    12947.0000    0.1880              0.8120
TRANSPORT            469.0000      141.0000    0.2131              0.7869
SERVICES            1958.0000     4939.0000    0.1715              0.8285
```

```
**** FILE SUMMARY

INPUT   C:\GAMS386\NER\JEFF.GMS
OUTPUT  C:\GAMS386\NER\JEFF.LST

EXECUTION TIME       =       13.350 SECONDS        VER: 386-SP-005
```

```
A SIMPLE MODEL OF A COMPUTABLE GENERAL EQUILIBRIUM MODEL: BASE SOLUTION             91/11/01 09:42:59
PAGE      1
                                                                              GAMS 2.20 DOS-386     4
*Modification du Code Obtenu de Shantayanan Devarajan
 *Washington,D.C., Aout,1991

     SCALARS
              sigma  Elasticite de substitution
              omega  Elasticite de transformation
              EO     Exportations inititales
              MO     Importations initiales
              XBAR   Possibilites de production
              X0     Valeur initiale de XBAR
              QO     Possibilites de consommatio
              DO     Valeur initiale  demand domestique
              BBAR   Valeur exogene de la balance commerciale
              PO     Indice initial  des prix a la consommation
              PWEO   Prix mondial initial des exportations
              PWE    Prix mondial des exportations
              PWMO   Prix mondial initial des importations
              PWM    Prix mondial des importations
              PDDO   Prix domestique initial ;

              sigma     =      .2;
              omega     =      .2;
              EO        =      25;
              MO        =      25;
              XBAR      =     100;
              X0        =     XBAR;
              QO        =     100;
              DO        =      75;
              BBAR      =       0;
              PO        =       1;
              PWEO      =       1;
              PWE       =       1;
              PWMO      =       1;
              PWM       =       1;
              PDDO      =       1;
     PARAMETERS  rho   Exposant Fonction Armington
                 h     Exposant Fonction CET
                 alpha  Parametre de proportion fonction CET
                 beta        Parametre de proportion fonction Armington
                 A           Parametre de dimension fonction CET
                 B           Parametre de dimension fonction Armington;
 *Calibrage
      rho =    (1/sigma) - 1;
      h   =    (1/omega)+1;
      alpha =  1/( (PDDO/PWEO) * (EO/DO)**(1/omega) +1);

beta=((PWMO/PDDO)*(MO/DO)**(1/sigma))/((PWMO/PDDO)*(MO/DO)**(1/sigma)+1);
      A=X0*(alpha*EO**h + (1-alpha)*DO**h)**(-1/h);
      B=QO*(beta*MO**(-rho)+(1-beta)*DO**(-rho))**(1/rho);

  DISPLAY rho,h,alpha,beta,A,B;

  POSITIVE VARIABLES
     E     Exportations
     DD    Demand  en bien domestique
     DS    Offre en bien domestique
     M     Importations
     X     Offre du bien compose
     PX    Indice du prix au producteur
     PQ    Indice du prix au consommateur
     PED   Prix domestique des exportations incluant la subvention
     PMD   Prix domestique des importations incluant les droits de douane
     PDT   Prix du bien domestique incluant la taxe
     PDD   Prix hors taxe du bien domestique
     ER    Taux de change nominal
     Y    PNB

  FREE VARIABLES
     Q    Consommation du bien compose
     TM   Droits de douane(taux)
     TE   Subvention a l'exportation(taux)
     TD   Taxe sur le bien domestique(taux)
     GR   Revenu net du Gouvernement
*  WALRAS Balance entre PNB et Depenses totales

  EQUATIONS
   OUTPUT  Production du bien domestique
   CONS    Consommation du bien compose
   EXPRAT  Ratio  Exportations-Offre domestique
   IMPRAT  Ratio  Importations-Offre domestique
   EXCH    Taux de change
   PEXP    Prix des exportations
   PIMP    Prix des Importations
   PDOM    Prix du bien domestique
   PDTEQ   Prix toutes taxes comprises du bien domestique
   GREQ    Deficit budgetaire
   YEQ     Definition du PNB
   G       Contrainte de Production
   D       Equilibre sur le marche interieur
   BOP     Balance commerciale
*  REV     Equilibre Revenu-Depenses
   ;
  OUTPUT.. X =E= A *(((alpha*(E**h))+((1-alpha)*(DS**h)))**(1/h));
  CONS..   Q
  =E=B*(((beta*(M**(-rho)))+((1-beta)*(DD**(-rho))))**(-1/rho));
  EXPRAT.. E/DS =E= ((PED/PDD)*((1-alpha)/alpha))**omega;
  IMPRAT.. M =E= DD*(((PDT/PMD)*(beta/(1-beta)))**sigma);
  EXCH..   PED =E= ER*PWE*(1+TE);
  PEXP..   PX =E= (PED*E + PDD*DS)/X;
  PIMP..   PMD =E= PWM*ER*(1+TM);
  PDOM..   PDT*DD + PMD*M =E= PQ*Q;
  PDTEQ..  PDT =E= (1+TD)*PDD;
  GREQ..   GR =E= (TM*ER*PWM*M) + (TD*PDD*DD) - (TE*ER*PWE*E) ;
  YEQ..    Y =E= (PX*X) + (ER*BBAR) + GR;
  G..      X =E= XBAR;
  D..      DD =E= DS;
  BOP..    (PWM*M) - (PWE*E) =E= BBAR;
*  REV..    Y =E= PQ*Q + walras;

* Initial Conditions
  DD.L =      75;
  DS.L =      75;
  M.L  =      25;
  E.L  =      25;
  X.L  =     100;
  Q.L  =     100;
  PED.L =    1.00;
```