Eleventh Floor
Menzies Building
PO Box 11E, Monash University
Wellington Road
CLAYTON Vic 3800 AUSTRALIA

Telephone: from overseas:
(03) 9905 2398, (03) 9905 5112   61 3 9905 5112 or 61 3 9905 2398

Fax:
(03) 9905 2426   61 3 9905 2426
e-mail   impact@vaxc.cc.monash.edu.au
web site   http://www.monash.edu.au/policy/

*C*ENTRE *of*

*P*OLICY

*S*TUDIES*and*

the *I*MPACT *of*

*P*ROJECT

INTRODUCTION TO
GEMPACK FOR GAMS USERS

by

M. KOHLHAAS

Deutsches Institut für Wirtschaftsforschung
Berlin, Germany

and

K.R. PEARSON

Centre of Policy Studies, Monash University

Introduction to GEMPACK for GAMS Users

M. Kohlhaas[1] and K.R. Pearson[2]

April 2002

Abstract

This document introduces GEMPACK to readers who are familiar with using GAMS for general equilibrium modeling. The document points out similarities and differences between GAMS and GEMPACK. The many similarities make it relatively easy for a GAMS modeler to begin using GEMPACK productively.

A very basic general equilibrium model is used to illustrate the similarities and differences. Readers wishing to obtain hands-on experience with GEMPACK can find detailed instructions for running this model in GEMPACK using the free Demonstration Version. Both the model and the GEMPACK software can be downloaded from the web.

---

[1] Deutsches Institut für Wirtschaftsforschung, 14191 Berlin, Germany.
   mkohlhaas@diw.de

[2] Centre of Policy Studies, Monash University, Clayton Vic 3800, Australia.
   Ken.Pearson@BusEco.monash.edu.au

1   Introduction[3]

GEMPACK is a suite of general-purpose economic modeling software especially suitable for general and partial equilibrium models. It supports the modeler in the four basics steps of using models for the analysis of economic problems: to build or modify a model, to construct a database, to run simulations and to interpret and present the results.

This section describes briefly the suite of GEMPACK programs and the main similarities and differences between GAMS and GEMPACK. More details about these are given in section 2.

In section 3 we tell you how to download the model we use to illustrate the similarities and differences between GAMS and GEMPACK. Readers keen to get to the hands-on part should first skim the first two sections (and perhaps come back to them later).

We conclude the paper in sections 4 and 5 where we briefly describe some features of GEMPACK not covered earlier and suggest ways in which you can find out more about GEMPACK.

1.1   The Suite of GEMPACK programs

In GEMPACK, models are written down, using a notation similar to normal algebra, in a TABLO Input (or TAB) file. The GEMPACK program TABLO is used to process TAB files. In particular TABLO reports syntax and semantic errors.

Any text editor can be used to write down the model code. However, GEMPACK provides the TABmate editor which facilitates this task by highlighting different parts of the model code during editing.

Various GEMPACK programs support setting up or inspecting an existing database. The most important are ViewHAR and MODHAR.

Once a model and database have been established, you can carry out a simulation (that is, solve the model) using the GEMPACK program GEMSIM.

The results of a simulation are contained in various types of data files, as we explain in more detail below. In particular, the programs ViewSOL and GEMPIE can be used to look at the change or percentage-change results from the simulation.

The program AnalyseGE assists modelers to analyse and understand their simulation results by making the numerical values of the components of any equation or variable available with a mouseclick.

More details about all of these steps are given later in this paper.

1.2   Model Specification MODEL.GMS or MODEL.TAB

Consider a simple GAMS model which is specified by the file **MODEL.GMS**.[4] This file MODEL.GMS contains

- the theoretical specification of the model,
- the starting data,
- the specifications of one or more simulations , and

---

[3] We are grateful to Joaquim Ferreira for helpful comments on this paper.

[4] In this document we have in mind models which are expressed in basic GAMS not in MPSGE syntax (see Rutherford, 1999).

- formulas and DISPLAY statements for calculating and reporting other results (for example, percentage changes).

In GEMPACK, there are three different files corresponding to MODEL.GMS, namely

- **MODEL.TAB** (**TABLO Input file**) which contains the theoretical specification of the model,

- **MODEL.DAT** (**data file**) which contains the starting data,[5] and

- **MODEL.CMF** (**Command file**) which contains the specifications for a single simulation.

The file MODEL.GMS may contain details for several simulations (solves) with the model. Each different simulation (solve) requires a different Command file in GEMPACK. See section 1.12 for more details.

Similar syntax is used in the sets, variables, formulas and equations in the files MODEL.GMS and MODEL.TAB. For example,

Sets in GAMS:

```
SETS
    i Sectors /s1, s2/
    f Factors /labor, capital/ ;
```

The same Sets in GEMPACK:

```
Set SECT # Sectors # (s1-s2) ;
Set FAC # Factors  # (labor, capital) ;
```

Variables in GAMS:

```
Variables
    Y           Total nominal household expenditure
    PC(i)       Price of commodity i ;

Positive Variables  Y, PC;
```

The same Variables in GEMPACK:

```
Variable (GE 0) Y # Total nominal household expenditure # ;
Variable (GE 0) (all,i,SECT) PC(i) # Price of commodity i # ;
```

Formulas in GAMS:

```
* initial prices
    PC_Z(i) = 1 ;
* Factor input f in industry j
  XF_Z(f,i) = DVFACIN(f,i)/PF_Z(f) ;
```

The same Formulas in GEMPACK:

```
!   Formulas for initial prices   !
FORMULA (all,i,SECT) PC(i) = 1 ;
FORMULA # Factor input f in industry j #
  (all,f,FAC)(all,j,SECT)  XF(f,j) = DVFACIN(f,j)/PF(f) ;
```

---

[5] It is possible to read data from separate files in GAMS, too.

An **Equation** in GAMS:

```
Cons_dem(i)  Household expenditure functions  ;
  Cons_dem(i)..    XH(i) =E= ALPHAH(i)*Y/PC(i)  ;
```

The same Equation in GEMPACK:

```
EQUATION Consumer_demands # Household expenditure functions #
  (all,i,SECT)  XH(i) = ALPHAH(i)*Y/PC(i)  ;
```

1.3 Levels/Linearized Equations and Variables in GEMPACK

When GEMPACK solves for a new equilibrium, it starts from an initial equilibrium and calculates the effect of a shock by using information of the "curvature" of the underlying curves.[6] So rather than determining the new equilibrium values directly, GEMPACK calculates the changes in the endogenous variables from a previous equilibrium. The new equilibrium values can then be found by adding these changes to the initial values of those variables.[7]

When a model builder writes down the model, he has the choice to specify the conditions for an equilibrium such that they refer to the level of the variables or to changes between two equilibria. Thus GEMPACK makes the distinction between **levels** and **linearized** variables and equations. The respective variables have to be declared accordingly. A (percentage) change equation can be derived from a levels equation by differentiating it.

The variables and equations in MODEL.TAB may be the levels variables and equations but, more commonly, are linearized versions of these levels variables and equations. For example, consider the household expenditure equation in the last example in section 1.2 above.

```
Variable (Levels, GE 0) Y # Total nominal household expenditure # ;
Variable (Levels, GE 0) (all,i,SECT) PC(i) # Price of commodity i # ;

  EQUATION Consumer_demands # Household expenditure functions #
    (all,i,SECT)  XH(i) = ALPHAH(i)*Y/PC(i)  ;
```

Variables Y and PC(i) measure household expenditure (in dollars, say) and the price of commoditiy i in the same way as the corresponding variables in MODEL.GMS do, that is, as level variables.

In many GEMPACK models, the corresponding linearized variables and equations would appear instead. These would be as follows.

```
Variable (Linear) y
  # Percent change in nominal household expenditure # ;
Variable (Linear) (all,i,SECT) pc(i)
  # Percent change in price of commodity i # ;

Equation (Linear) Consumer_demands
  # Household expenditure functions #
    (all,i,SECT)  xh(i) = y - pc(i) ;
```

The last equation indicates that the percentage change xh(i) in household consumption of commodityi is equal to the percentage change in household expenditure y minus the percentage change in the price of commodity i, pc(i). Sometimes the notation p_PC(i) would be used instead of pc(i) in the

---

[6] It approximates changes by linearizing the system in the neighborhood of the initial solution. The effects of a shock can be calculated as accurately as necessary by breaking down shocks into small changes, which are evaluated in consecutive calculations. At each step, GEMPACK just remembers the percentage change (or change) from the start.

[7] Often the initial equilibrium values are not known explicitly (though they can be inferred from the starting data), and all that is reported are the percentage changes (or the changes).

linear case. In most TAB files the qualifier "(Linear)" and the words "percent change" would be omitted, since these are usually the defaults (see section 2.2).

These two equivalent ways of expressing the same variables and equations are distinguished in GEMPACK speak by referring either to **levels** variables and equations (as in the first example above) or by referring to **linear** (or **linearized**) variables and equations (as in the second example above).

It is possible to use only levels equations and levels variables in a TAB file, or to use a mixture of linear and levels equations in a TAB file. However, most models (including the well-known GTAP, ORANI and MONASH models) are implemented in GEMPACK via TAB files containing only (or mainly) linearized variables and equations.

One very important point is that you obtain exactly the same results from a model whether it is written down in levels or linearized form. This is well documented (see, for example, Hertel *et al* (1992) and Pearson (1991)). When you carry out the same simulation using both GAMS and GEMPACK (see section 3 below), you will find the same results. We indicate how it is possible to get the correct nonlinear results from linearized equations in section 2.9 below.

1.4 Starting Data

For a relatively simple GAMS model MODEL.GMS, the starting data (eg a SAM) is often contained inside the model code. In GEMPACK, there is usually a separate data file MODEL.DAT containing this data. MODEL.DAT can be a text file or a binary file called a **Header Array** file. There are **READ** statements, for example,

```
READ DVFAC FROM FILE basedata Header "abcd" ;
```

in MODEL.TAB for associating numbers on the data file with the variable DVFAC in the model.

In both GAMS and GEMPACK, the starting data must be balanced, i.e. they must represent a solution for all the (levels) variables. In GAMS a SOLVE is done as a replication check of the data before any simulations are carried out. In GEMPACK, a separate check that the data is balanced should be carried out, usually by a special piece of code which may not be part of MODEL.TAB.

1.5 Specifying Exogenous/Endogenous Variables and Shocks

Within MODEL.GMS, the exogenous variables are specified as **Parameters** or by indicating the desired levels values of the relevant variables via **.FX** statements. In either case the exogeneity and the desired value can be given at the same time. For example, the statements

```
X1.FX = 10 ;
Parameter   X2   /20/ ;
```

indicate both the exogeneity of X1 and X2 and their desired values.

In GEMPACK, the exogenous variables for the simulation are specified in a list in the Command file as in, for example,

```
exogenous X1, X2 ;
```

In GAMS a shock can be implemented by changing the value of a parameter or a fixed variable between two model runs. In GEMPACK the shock has to be applied to an exogenous variable, not a parameter. The change in the value of an exogenous variables is given via **shock** statements in the Command file. For example the statement

```
Shock x1 = 10 ;
```

means that the post-simulation value of the levels value associated with percentage change variable x1 is 10% higher than its pre-simulation value (the pre-simulation value inferred from the data base).

1.6 Carrying Out a Simulation

When the model is solved in GAMS, the program GAMS.EXE is run. This processes the theoretical specification of the model, then solves the model and reports the results.[8]

In GEMPACK it is necessary to run two programs, TABLO.EXE and GEMSIM.EXE. First TABLO process the theoretical specification of the model in MODEL.TAB and it writes a computer version of this.[9] Then GEMSIM is run to carry out the simulation.[10] GEMSIM reads

- the computer version of MODEL.TAB produced when TABLO was run,

- the data in MODEL.DAT, and

- the exogenous/endogenous split and shocks in the Command file MODEL.CMF. The split of the variables into exogenous and endogenous is referred to as the **closure** of the model.

GEMSIM writes the simulation results to a so-called **Solution file** (say MODEL.SL4). MODEL.SL4 is a binary file which contains the changes or percentage changes in each of the variables in the model. It also contains the pre-simulation and post-simulation values of some or all of the variables.

GEMSIM also writes the so-called **updated** data file MODEL.UPD. This file contains post-simulation values of all the data items which are contained in the file MODEL.DAT.[11]

1.7 Solution Methods

GAMS provides a wide array of different solution methods. The method to be used is specified in a SOLVE statement.

GEMPACK provides one basic method for solving models. This relies on a system of linearized equations of the model and the initial solution. A model is presented to the GEMPACK solver as a so-called **Initial Value** problem. GEMPACK provides the alternative methods **Gragg** and **Euler** for solving such problems.[12] Even though GEMPACK solves a linearized form of the model, it is able to produce accurate solutions of the underlying nonlinear levels equations.[13] For the same model and simulation, the numerical solutions produced by GAMS and GEMPACK are the same even though they use different solution methods and model representations.

---

[8] Actually several programs may be run automatically once GAMS.EXE is started (but users do not need to be concerned about that).

[9] There is an alternative in which TABLO writes a so-called **TABLO-generated FORTRAN program** say MODEL.FOR. This program must be compiled and linked to produce the executable image MODEL.EXE. This program MODEL.EXE can then be run to carry out the simulation in much the same way that GEMSIM is run. We say more about this alternative in section 4.1.

[10] In fact the program TABLO only needs to be run as part of a simulation if the theoretical specification of the model (in file MODEL.TAB) changes. If the model does not change (that is, MODEL.TAB does not change), several simulations with the same model can be carried out just be running GEMSIM with different Command files (and possibly different data files).

[11] For example, suppose that one item on MODEL.DAT is the pre-simulation value of household consumption of commodity "food". If the price of "food" increases by 5 percent and the quantity of "food" consumed by households increases by 3 percent, then the value of household consumption of "food" shown on MODEL.UPD will be 8.15 percent (100*1.05*1.03=108.15) higher than that on MODEL.DAT.

[12] These are well-known methods for solving initial value or ordinary differential equations numerically – see, for example, chapter 6 of Atkinson (1989) or chapter 15 of Press *et al* (1986).

[13] GEMPACK also makes Newton's method available, but only if all equations are written as levels equations. In this case, the initial values of the levels variables do not need to be a solution of the levels equations.

Sometimes models in GEMPACK are solved using **Johansen's** method. This is a 1-step Euler method and only produces an approximate solution of the underlying levels results. While a Johansen solution may be useful for debugging a model or identifying the main economic mechanisms in the model producing the results, these Johansen results are not usually sufficiently accurate for reporting results of the model.

1.8   Viewing the Results of the Simulation

In GAMS, the results are shown in the Listing file **MODEL.LST** produced by GAMS. This contains the post-simulation values of all the variables in the model. Additionally, other parameters may be included via DISPLAY statements in MODEL.GMS. For example, if you wish to see the percentage changes in a levels variable X1 in the model, you would include the statements

```
P_X1 = (X1/X1_BASE - 1) * 100 ;
DISPLAY P_X1 ;
```

(where X1_BASE has been set equal to the pre-simulation value of X1).

In GEMPACK there are two lots of results, namely the changes or percentage changes in the variables of the model in MODEL.SL4 and the updated data in MODEL.UPD. The changes or percentage changes are calculated automatically for all (endogenous and exogenous) variables of the model – no statements in MODEL.TAB are required. GEMPACK provides the windows program **ViewSOL** for viewing the results in MODEL.SL4. Recall that the original data file MODEL.DAT could be a text file or a binary (Header Array) file. The updated data file MODEL.UPD has the same format as the original data file MODEL.DAT. If this is a Header Array file, the updated data can be viewed using the windows program **ViewHAR**. If the updated data is a text file, the updated data can be viewed in your favourite text editor.

1.9   Writing Reports

Both GAMS and GEMPACK provide access to tools for writing reports. In particular they provide means of putting results into spreadsheets for producing graphs and tables.

In GEMPACK, both ViewHAR and ViewSOL provide simple ways of copying tables into spreadsheet programs and allow users to produce graphs on the screen. These graphs can be copied into word processors (eg Microsoft Word) or spreadsheet programs (eg Microsoft Excel). GEMPACK also provides various utility programs (notably SLTOHT) for converting parts of the Solution file to tables suitable for importing into word processors or spreadsheet programs.

1.10 Windows Tools for Model Development and Model Solution

GAMS provides GAMS IDE as a Windows development environment. This assists with processing mainly .GMS files and results.

GEMPACK provides

- **WinGEM** for model development and model solution. This assists modelers with the sequence (run TABLO, run GEMSIM, run ViewSOL) and assists with file names and editing TAB and Command files.

- **TABmate** for model development. This is an environment customised for building, modifying and debugging TAB files.

- **RunGEM** for model solution. This is an environment customised for running different simulations with the same model. It automates the specification of closures and shocks, and the

naming and viewing of post-simulation results, including those on the Solution file and on the updated data file(s).[14]

1.11 Analysing Simulation Results

After carrying out a simulation, modelers need to be able to analyse the results in order to identify the main mechanisms in the model (both theoretical and data) which are producing the results.

GEMPACK provides the Windows program **AnalyseGE** which assists in this task. AnalyseGE gives users "point and click" access to the equations of the model, the data and consequences, and to the simulation results. A modeler can click on any equation or formula and ask the software to group the terms into different natural parts and give the numerical values of each term.

Having access to the linearized equations of the model seems to be a distinct advantage when analysing simulation results in this way. [Carrying out such an analysis on the basis of the levels equations seems to be more difficult.]

1.12 Multiple Solves

In GAMS multiple solves are available directly in the MODEL.GMS file. There can be several SOLVE statements in the file. Alternatively, there is a LOOP statement which allows modelers to repeat the statements (including SOLVE statements) within a loop.

In GEMPACK, only a single solve is allowed each time GEMSIM is run. But several solves are easily carried out by several runs of GEMSIM.

For example, consider the 3 solves in GAMS specified as follows (for a simple model with just one exogenous variable X1).

```
X1.FX = 1.0 ;
* Replication check
SOLVE MODEL USING NLP ;
```

- The values of the variable after the first solve will be those from the initial solution, as reflected in the initial data base.

```
X1.FX = 1.2 ;
SOLVE MODEL USING NLP ;

* Code to report changes from the initial levels values. For example
P_X2 = (X2/X2_BASE - 1) * 100 ;
```

- The P_X2 results reported after the second solve will be the percentage change in X2 between its value when X1=1.2 and its initial value X2-BASE.

```
X1.FX = 1.3 ;
SOLVE MODEL USING NLP ;
P_X2 = (X2 – X2_BASE - 1)/100 ;
```

- The P_X2 results reported after the third solve will be the percentage change in X2 between its value when X1=1.3 and its initial value X2_BASE.

```
* Code to reset C2_BASE to be its current value.
X2_BASE = X2 ;
```

---

[14] RunGEM is a general-purpose version of the windows interface **RunGTAP** (see the GTAP web site http://www.gtap.agecon.purdue.edu/products/utilities/default.asp) used to automate carrying out simulations with the GTAP model.

```
* Code to specify new value for X1 and to solve the model again
X1.FX = 1.4 ;
SOLVE MODEL USING NLP ;
P_X2 = (X2/X2_BASE - 1) * 100 ;
```

- The P_X2 results reported after the fourth solve will be the percentage change in X2 between its value when X1=1.4 and its value when X1=1.3.

In GEMPACK nothing corresponds to the first solve (replication check).[15] The results from the other 3 solves would be produced as follows.

- In MODEL1.CMF (second GAMS solve), start from MODEL.DAT. Specify a shock of 20 percent for X1 (from 1.0 to 1.2). Specify the updated data as MODEL1.UPD. The relevant Command file statements are

  ```
  File BaseData = model.dat ;
  Shock x1 = 20 ;
  updated file BaseData = model1.upd ;
  ```

  The results reported in MODEL1.SL4 will be the percentage changes from the initial solution (when X1=1.0) to the solution in which X1=1.2.

- In MODEL2.CMF (third GAMS solve), start again from MODEL.DAT. Specify a shock of 30 percent for X1 (from 1.0 to 1.3). Specify the updated data as MODEL2.UPD. The relevant Command file statements are

  ```
  File BaseData = model.dat ;
  Shock x1 = 30 ;
  updated file BaseData = model2.upd ;
  ```

  The results reported in MODEL2.SL4 will be the percentage changes from the initial solution (when X1=1.0) to the solution in which X1=1.3.

In MODEL3.CMF (fourth GAMS solve), start from MODEL2.UPD, which represents the state of the economy when X1=1.3 (that is, after the third GAMS solve). Specify a shock of 7.6923 percent for X1 (from 1.3 to 1.4). Specify the updated data as MODEL3.UPD. The relevant Command file statements are

```
File BaseData = model2.upd ;
Shock x1 = 7.6923 ;
updated file BaseData = model3.upd ;
```

The results reported in MODEL3.SL4 will be the percentage changes from the solution when X1=1.3 to the solution in which X1=1.4.

---

[15] The balance of the data base should be checked separately. This check is logically equivalent to the replication check GAMS solve.

2 A Closer Look at a Program File

The following table shows the basic structure of a model in GEMPACK and GAMS respectively.

GEMPACK .TAB file	GAMS .GMS file
DEFAULT statements	
FILEs	
SETs	SETs
COEFFICIENTs	PARAMETERs (and SCALARs)
	TABLEs
VARIABLEs	VARIABLEs
READ	READ
FORMULAs	Calibration
EQUATIONs	EQUATIONs
UPDATEs	
	MODEL statement
(in Command file)	SOLVE
Additional calculations and reporting	Additional calculations and reporting

The table shows that the basic structure is quite similar. The following sections give some details about the different categories.

2.1 Comments and Case Insensitivity

Comments are allowed (and are important) in GAMS and TAB files. In GAMS files, comments are often on a separate line which begins with ***** which indicates a comment. In TAB files, comments begin with an exclamation mark **!** and end with another **!**. For example,

```
VARIABLE TAXRATE ;    ! can be used to simulate carbon tax changes !
```

in which only the second part of the line is a comment.[16]

In both GAMS and TAB files, case (upper or lower) is not important as far as the software is concerned. However modelers often use upper or lower case to make the files more readable.

2.2 Default Statements

In GEMPACK when some of the entities are declared a qualifier is used to specify of which sub-type the entity is. For convenience GEMPACK allows the qualifier to be omitted. In these cases the sub-type indicated in the default statement will be assigned. For example,

---

[16] Somewhat confusingly, comments in GEMPACK Command files and other text files follow a different rule. There an exclamation mark **!** begins a comment which ends at the end of the line as in, for example,
```
shock x1 = 10 ;  ! 10 percent increase
shock x2 = 20 ;
```
where the end of the first line is a comment and the second line is not a comment.

VARIABLE (DEFAULT = LEVELS) ;

Indicates that, unless specified otherwise, all VARIABLE statements define LEVELS (as distinct from Linearized) variables. The alternative default statement would be

VARIABLE (DEFAULT = LINEAR) ;

2.3 FILE and READ Statements

In GAMS data values can be specified by TABLEs or direct assignment to SCALARs or PARAMETERs. In GEMPACK data are usually stored in a data file and read from this file via a READ statement.[17] The FILE statement defines the logical name of the files, i.e. the name by which the data files are addressed in the TABLO file. When running a simulation, the actual name of the data file on the computer will be given in the Command file.[18] This allows the same model to be easily solved with different starting data files (for example, data representing different years or different aggregations).

2.4 SETs

The use of SETs is very similar in both environments. The name of the set is declared and the members are either listed or (in GEMPACK) can be read from a data file. There is one major difference. In GAMS the name of a SET will be used as an index whenever the members of this set are addressed. If a set is to be addressed by a different name, this name has to be introduced by an ALIAS statement. In GEMPACK the index is chosen independently of the name of the set. Whenever a coefficient, variable or equation is to be indexed, a **quantifier** of the type

(ALL, index, SET)   [for example, (All,i,SECT)]

defines the name of the index and the set over which it ranges. (ALL is the keyword for the quantifier).

2.5 SCALARs, PARAMETERs and COEFFICIENTs

In GAMS SCALARs and PARAMETERs are used as entities to store values which are assigned to them or the results of calculations. In GEMPACK the corresponding entities are called COEFFICIENTs.

Multi-dimensional COEFFICIENTs are declared by using quantifiers of the form (ALL, indexname, setname) as in, for example,

```
COEFFICIENT
 (all,f,FAC) (all,j,SECT)  ALPHAFAC(f,j) #Factor parameter#;
```

In GAMS, the value of a parameter (e.g. a tax rate) can be changed between two simulations to model the policy shock to the economy. In GEMPACK exogenous VARIABLEs are usually used for this purpose.

Example: Modeling tax rate changes.

GAMS:

```
     Parameter TaxRate ;
     TaxRate = 10 ;
     SOLVE …
```

---

[17] It is possible to read data from separate files in GAMS, too.

[18] In GAMS, these are referred to as internal and external file names.

```
      TaxRate = 20 ;
      SOLVE …
```

GEMPACK:[19]

```
[in TAB file]
Variable (Linear) taxrate # percent change in tax rate # ;

[in Command file]
Exogenous taxrate ;
Shock taxrate = 100 ; ! per cent increase – takes level from
                      ! 10 to 20
```

In GEMPACK, Coefficients can be declared as what GEMPACK refers to as PARAMETERs which implies that their values cannot be changed by assignments during a model run. [This is different from the use of PARAMETER in GAMS.] Without this qualifier the values of a Coefficient can vary through a simulation. For example, Armington substitution elasticities are Parameters in GEMPACK since they cannot change during a simulation (though they may change from simulation to simulation). These are declared using the PARAMETER qualifier as in

```
Coefficient (Parameter)  (all,i,SECT)
    ARMINGTON(i) # Armington parameter # ;
```

2.5.1   Coefficients in Linearized TAB Files

In linearized TAB files, Coefficients are often used to hold and indicate the current value of something which could be regarded as a levels variable (for example, total household expenditure or GDP). The values of such Coefficients vary throughout a simulation (that is, from step to step of a multi-step simulation). An UPDATE statement may be used to describe the changes. For example, consider DVHOUS in a linearized TAB version of the model in section 3 below.

```
Coefficient (All,i,SECT)
    DVHOUS(i) # household expenditure on i # ;
Update (All,i,SECT) DVHOUS(i) = p_PC(i) * p_XH(i) ;
Coefficient (All,i,SECT) DVCOM(i) # total demand for i # ;
Formula (All,i,SECT)
  DVCOM(i) = SUM(j,SECT, DVCOMIN(i,j)) + DVHOUS(i) ;
```

In the example above, DVHOUS and DVCOM are Coefficients whose values change from step to step of a multi-step calculation. Initially (pre-simulation), the values of DVHOUS and DVCOMIN are read from the database while those of DVCOM are calculated (via the Formula) from the values of DVHOUS and DVCOMIN. In subsequent steps,

- the values of DVHOUS are updated according to the change in the relevant prices and quantities from the previous step (via the UPDATE statement above). [For example, if the value of DVHOUS is 10 at the beginning of step 3, and if p_PC increases by 5% during step 3 and XH increases by 3% during step 3, then the value of DVHOUS will be increased by (5+3)% to become 10.8 at the end of step 3 (or the beginning of step 4).] [20]

---

[19] Usually in GEMPACK models, the power of the tax (1+t) rather than the tax rate t would be shocked, since the tax rate may be zero in the initial situation or change the sign due to a policy shock. This, however, would cause problems in a percentage change representation. In the following example, the tax rate is used to make the examples comparable.

[20] You may wonder why there is a * in update statement instead of the + in the corresponding linear equation. This is explained in section 3.10.1.

- the values of DVCOM in each step are calculated via the Formula above from the values of DVHOUS and DVCOMIN at the start of this step. Thus the DVCOM values change from step to step to reflect changes in DVHOUS and DVCOMIN values.

2.6 TABLEs

Tables are a convenient way to enter data into a GAMS program. In GEMPACK no equivalent to TABLE exists. Data are read from files and programs such as MODHAR and ViewHAR facilitate creating and modifying data files. Data files in GEMPACK can be either text files (which can be created using a text editor, or imported from a spreadsheet or other program) or binary (Header Array files) which can be created using MODHAR or ViewHAR.

2.7 VARIABLEs

In GAMS VARIABLE is the name for what are called endogenous variables by economists (Brooke *et al* 1998).[21]

In GEMPACK variables are used to model endogenous as well as exogenous variables. When a variable is declared, no decision has to be made if it is to be endogenous or exogenous. VARIABLEs are declared endogenous or exogenous in the Command file. This flexibility permits the modeler in GEMPACK to vary the closure (that is, which variables are exogenous and which are endogenous) without having to change the TAB file (and hence without having to rerun the program TABLO). Closure changes are indicated in the Command file.

In GEMPACK there are various types of variables which have to be specified by a qualifier. There is a distinction between LEVELS and LINEAR variables as well as between CHANGE and PERCENT_CHANGE linear variables. The difference between linear and levels variables has been explained in section 1.3 above.

Within GEMPACK, even when variables are declared as levels variables in the TAB file, all simulation results are reported via the corresponding linear variables. Results are reported either as percentage changes (the most common) or as absolute changes. Which result is reported is determined by how the variable is declared in the TAB file. Consider a price variable. If it is declared via

```
Variable (Percent_Change) (all,i,SECT) pcom(i) ;
```

then results will be given as percentage changes.[22] [For example, a result of 15 means that the price has increased by 15 percent – say from 1 to 1.15 .] If it is declared as

```
Variable (Change) (all,i,SECT) pcom(i) ;
```

then results will be given as absolute changes. [For example, a result of 0.08 means that the price has increased by 0.08 – say from 0.5 to 0.58 .]

Note that the percent change or change reported is measured from the pre-simulation value (that is, the solution inferred or made explicit in the TAB file from the pre-simulation data – see section 1.4 above).

In GEMPACK, the pre-simulation levels value of a variable may not be explicit, especially when the model is implemented via a linearized TAB file. For example, in the linearized TAB file for the example model in section 3 below, the pre-simulation levels values of prices and quantities are not made explicit. But percent change results are easily understood.

---

[21] It is however possible to fix a variable which is considered exogenous in a specific model by the statement "variable_name.fx = value.

[22] The qualifier (Percent_Change) is the default so is usually omitted.

For levels variables a range of acceptable values can be specified, similar to GAMS.

GAMS:

```
Positive variable    Y   household expenditure ;
* lower and upper bounds
Y.LO = 0.0001 ;  Y.UP = 20 ;
```

GEMPACK:

```
Variable (GE 0)  Y  # household expenditure # ;
Variable (GE 0.0001, LE 20)  PLAB # price of labor # ;
```

The above statements show the different GAMS and GEMPACK style for specifying **labels** for variables (and other entities). These labels are used in printouts and in results to describe the variable whose values are being reported. In GAMS the label "household expenditure" simply appears after the variable name Y. In GEMPACK the label is given inside hashes # # .

2.8   FORMULAs and EQUATIONs

The formulas in GAMS and GEMPACK look very similar as the examples below show.

GAMS:

```
* Intermediate input of commodity i in industry j
   XC_Z(i,j) = DVCOMIN(i,j) / PC_Z(i) ;
```

GEMPACK:

```
Formula # Intermediate input of commodity i in industry j #
 (all,i,SECT)(all,j,SECT)  XC(i,j) = DVCOMIN(i,j) / PC(i) ;
```

GEMPACK requires the keyword "Formula".   It may be followed by a qualifier INITIAL or ALWAYS. FORMULA(INITIAL) are only calculated during the first step in a multi-step simulation, while FORMULA(ALWAYS) are calculated at every step. The default is ALWAYS for formulas with a real coefficient on the left-hand side and is INITIAL in the case of an integer coefficient.

The Equations also look very similar as the examples below show.

GAMS:

```
EQUATIONS
  Com_clear(i) ;

* Commodity market clearing
 Com_clear(i)..   XCOM(i) =E= XH(i) + SUM(j, XC(i,j)) ;
```

GEMPACK:

```
Equation Com_clear # Commodity market clearing #
(all,i,SECT)   XCOM(i) = XH(i) + SUM(j,SECT, XC(i,j)) ;
```

In GAMS the group of equations Com_clear(i) is declared first and later the equation itself appears after "Com_clear(i)..". In GEMPACK the keyword Equation is followed by the equation name Com_clear (no indices after) and then the equation itself, in which the quantifiers are explicit. GAMS uses the operator "=E=" between the two sides of the equation whereas GEMPACK just uses "=". Both use the SUM operator, although the syntax after "SUM(" is slightly different – in GEMPACK the set SECT over which we are summing is explicit whereas this set is implicit by the use of the alias "j" in the GAMS case.

2.9 UPDATEs

These GEMPACK statements have no counterpart in a GAMS file. These are the statements which enable GEMPACK to produce accurate solutions of the underlying nonlinear levels equations from a linearized TAB file. In essence, these statements tell how something read from the data base (typically a dollar value) changes when the associated prices and quantities change. See section 2.5.1 above for an example which shows the effect of Update statements.

When you use GEMPACK to carry out a simulation, the software breaks the shocks (changes to the exogenous variables) into several small parts. The solution procedure consists of several steps. In each step, one part of the shocks are applied. [For example, there may be 10 steps. In each step, one-tenth of each of the shocks is applied.] In each step, the linearized equations of the model are solved. Then, before the next step, the UPDATEs are used to recalculate the dollar values in the database. This ensures that the values and shares which enter into the linearized equations as coefficients reflect the changes in the exogenous and endogenous variables which occurred in this step. Thus, although the linearized equations are solved at each step, the solution path taken by the software closely approximates the non-linear path implied by the underlying (usually nonlinear) equations of the model. In this sense, it is the UPDATE statements which allow GEMPACK to accurately solve the underlying levels equations of the model.[23]

---

[23] Section 2.11.3 of GEMPACK document GPD-1 (see section 6.1) explains in more detail how GEMPACK accurately solve nonlinear equations. That section also introduces the important idea of extrapolation on the basis of two or three different multi-step calculations.

3 A Simple Example Model

In this section we introduce a simple model and use it as the basis of detailed hands-on examples designed to introduce you to GEMPACK. We suggest that you work through this section on your computer, beginning by solving the model using GAMS and then solving the same model using GEMPACK. We include instructions for downloading, installing and using the Demonstration Version of GEMPACK, which assume that you are working on a Windows PC.

The model is written down in

- the GAMS file GAMSGP.GMS

- the GEMPACK file GAMSGP.TAB which is a direct "translation" of the GAMS file into the TABLO language used by GEMPACK.

Later in this section we also introduce alternative TAB files for the same model.

3.1 An Overview of the Model

The illustrative example used in this section is a conventional neoclassical general equilibrium model of a closed economy with two sectors and two primary factors. The two commodities can be used as an intermediate input or for final consumption. The firms maximize profits subject to the production function, which is Cobb-Douglas for all inputs. There is one household which owns the labour and capital endowment and consumes the two goods. It maximizes a Cobb-Douglas utility function subject to the budget constraint. There is no government. Walras' law is used to drop one equation, in this case the household budget constraint. (Alternatively, any of the market clearing conditions could be omitted.) The price of commodity 1 is chosen as the numeraire. (Alternatively, the price of any commodity of primary factor could have been chosen.)

The database consists of an input-output table which contains information about the values of the inputs of the primary factors, labour and capital, as well as the intermediate inputs. The database is balanced, i.e. all markets are cleared in the initial situation. For Cobb-Douglas functions all parameters can be inferred from this database. The model will be solved for the endogenous variables, i.e. the factor and commodity prices as well as the quantities supplied and demanded.

The simulation undertaken is an increase of labour endowment by 10 percent. This will result in a fall of wages relative to the rate of return to capital and a fall of the relative price of the labour-intensive commodity 2.

3.1.1 Downloading the GAMSGP Model

You need to download the GAMSGP model files from the GEMPACK web site. Go to the address

   http://www.monash.edu.au/policy/gpgams1.htm

and download GAMSGP.ZIP. We suggest that you put this file into the directory C:\GAMSGP on your PC. Unzip the files into this directory.

3.2 Solving the Model Using GAMS

Load GAMSGP.GMS into your GAMS software. Read the equations of the model and become familiar with the notation used.

There is just a single simulation in GAMSGP.GMS, in which the labor endowment is increased by 10 per cent (and the capital endowment is held fixed).[24]

Run GAMS to solve this simulation. Then examine the results and check that they are qualitatively as you would expect. For comparison with the GEMPACK results, note the percentage changes in the prices and outputs of the two commodities and in the prices of the primary factors, which are reported at the end of the gamsgp.lst file. In particular, check the following results:

```
VAR XCOM  Total demand supply of commodity i

        LOWER      LEVEL      UPPER     MARGINAL

s1 8.0000E-4     8.471       +INF

s2    0.001     12.828       +INF


PARAMETER XCOM100   output
      s1 5.885,    s2 6.899
PARAMETER DVHOUS_new  Dollar value of household use of commodity i
      s1 2.118,    s2 4.235
PARAMETER DVHOUS100
      s1 5.885,    s2 5.885
```

3.3   Downloading and Installing the Demonstration Version of GEMPACK

In this section we tell you how you can download and install the Demonstration Version of GEMPACK. If you already have a version of GEMPACK installed on your computer, you can skip this section and go on to the next section.

Download the Demonstration Version of GEMPACK from the GEMPACK web site
   http://www.monash.edu.au/policy/gpdemo.htm

Follow the instructions to download at least the GEMPACK Programs (TABLO, GEMSIM etc) and WinGEM (the Windows Version of GEMPACK) and associated programs (ViewHAR etc). For WinGEM, we suggest that you download the LF90 version. You may also wish to download the GEMPACK documentation and the AnalyseGE program.

Once you have downloaded these, install the GEMPACK programs (by running DM_CD.EXE) and the windows programs (by running WIN_CD90.EXE)

This installation process will put many files into your C:\GP directory. It will also put some icons on your desktop including one for WinGEM.

3.4   Starting WinGEM and Looking at GAMSGP.TAB

You should have a WinGEM icon on your desktop. Double-click on this icon to start *WinGEM* (the Windows version of GEMPACK) running.[25] This should give the main menu, as shown, across the top of the screen.

---

[24] A homogeneity test of the model could be undertaken first by activating the statement
```
*  PC.fx("s1")   = 2 * PC_Z("s1") ;
```
which doubles the price of the numeraire and which is commented out in GAMSGP.GMS.

[25] If you cannot find the WinGEM icon on your desktop, use **My Computer, Explorer** or a similar program to locate WINGEM.EXE, which is probably in C:\GP. Double-click on that to start WinGEM.

```
┌─────────────────────────────────────────────────────────────┐
│           WinGEM  -  GEMPACK for Windows                      │
├─────────────────────────────────────────────────────────────┤
│  File   Simulation   HAFiles   OtherTasks   Programs   Options   Window   Help │
└─────────────────────────────────────────────────────────────┘
```

You may need to look closely to see this since WinGEM is rather self-effacing and only occupies a small part of the top of you screen - the rest of the screen is as it was before.

3.4.1   Setting the Working Directory

The first thing to do is to tell WinGEM in which directory the files for input are stored and in which directory it should write output files. These directories are known as the **default** input and working directories respectively. Often (as will be the case below), these two are the same.

For the GAMSGP model examples here, the input and working directories need to be **C:\GAMSGP** in which you installed the files (see section 3.1.1 above). To set this, first click on

File

in the main WinGEM menu. This will produce a drop-down menu. In the drop-down menu, click on the menu item

Change both default directories...

The notation we use for this sequence is

File | Change both default directories...

In the file selection box that appears, choose drive   **C:** (or the drive containing your directory \GAMSGP if it is on a different drive). Then double-click on **C:** (this will be at the top of the list of directories shown) and then double-click on the subdirectory   **GAMSGP**. [Make sure that the directory name shown in blue above the selection box changes to C:\GAMSGP (or D:\GAMSGP etc if your \GAMSGP directory is on another drive).] Click on the *Ok* button.

3.4.2   Editing Text Files in WinGEM

We strongly suggest that you use **TABmate** as your test editor when working with GEMPACK TAB files.[26] To do this, select

Options | Change Editor | Use TABmate

from WinGEM's Options menu.

3.5   The GEMPACK Files for the Simulation (TAB, Data, Command files)

Before you run the simulation using GEMPACK, we suggest you look at the three GEMPACK files used to run the simulation. These are the

• TAB file GAMSGP.TAB (this contains the theory of the model, much like GAMSGP.GMS),

• the data file GAMSGP.DAT (this contains the input-output data for the model),

• the Command file GAMSGP.CMF (this contains the closure and shock).

---

[26] Similar to GAMSIDE, TABmate displays the model code in several colours which indicate the function of a piece of code.

3.5.1   Looking at the TAB File GAMSGP.TAB

Now look at the TAB file version of the GASMGP1 model in TABmate. To do this, select

File | Edit file…

from WinGEM's File menu and select file **GAMSGP.TAB**

TABmate uses different colors and fonts to distinguish the different parts of the TABLO syntax.

The first thing that will strike you is the section at the top with a blue background. This is the start of the GAMS file GAMSGP.GMS commented out using the so-called **strong comment markers** which are **![[!** at the start and **!]]!** at the end of the comment.

In GAMSGP.TAB, we have left all the statements and comments from GAMSGP.GMS and commented them out with strong comment markers. We have placed GEMPACK statements which achieve the same effect after these sections. We hope that, by doing this, you will be able to see readily how different parts of the GAMS file are translated into TABLO language.

We suggest that you look down the file, noting the GAMS parts commented out (they all have a blue background) and then looking briefly at the corresponding GEMPACK parts which follow them. You may find it useful to look again at the Table at the start of section 2.

Below we comment on some notable features.

- Notice that GEMPACK uses exclamation marks **!** to start and end comments (sometimes called "weak" comments to distinguish them from the "strong" comments described above).

- You can see the DEFAULT statements (see section above). These indicate that all Variables declared without a qualifier will be levels variables, similarly Equations will be levels equations and all Coefficients will be Parameters. Although there is not DEFAULT statement to indicate this, all associated linear variables will report percentage changes (rather than absolute changes).[27]

- Not far down from the start of the file you will see the GEMPACK versions of the two sets SECT and FAC. [GEMPACK does not need an ALIAS statement – see section 2.4 above.]

- Next the variables are declared in the GAMS version. You can see the corresponding GEMPACK declarations. Whenever a levels variables is declared in the TAB file, GEMPACK provides the associated linear variable whose name begins with "p_". For example, the linear variable p_Y represents the percentage change in the levels variable Y.

- Following down a little further, you can see where the data are included in the GAMS file via TABLEs. Note that there is no corresponding data in the TAB file. Instead there is a **File** statement followed by three **READ** statements. In GEMPACK, the data for the GAMSGP model is held on the data file GAMSGP.DAT, and not in the TAB file.

- Next come various formulas which set the initial values of various variables (including prices and quantities). The GEMPACK section on initial prices is very similar to the corresponding section in GAMS.

- A surprise may be the fact that many of what are formulas and closely related equations in the GAMS file are combined in the GEMPACK version as "Formula & Equation" statements. See for example the Comin equation.

- The formulas for the parameters (such as ALPHACOM) are very similar in the two versions.

---

[27] This is the case in all TAB files unless a contrary DEFAULT statement
DEFAULT (VARIABLE = CHANGE) ;
is present.

- Various levels equations (such as Consumer_demands) follow. These are very similar in the two versions. GAMS announces all equations first (in the EQUATIONS section) by declaring the names of the different equations and their indices (for example "Cons_dem(i)"). The actual equations come later. In GEMPACK the equation names, indices and the actual equation all come in the same Equation statement.

- The equation Numeraire in the GEMPACK case is a little unusual. Read the comments in GAMSGP.TAB about it.

- Now come various parts of the GAMS file which have no direct counterpart in the GEMPACK TAB file. These include the MODEL statement, starting values, lower bounds, fixed (that is, exogenous) variables, the shock, the solve statement and the section for reporting results. In GEMPACK the starting values are given by the Formulas, the exogenous variables and shocks are on the Command file GAMSGP.CMF, the instructions about how to solve (which method, how many steps) are in the Command file and there is no reporting section since all percentage changes are automatically reported.

3.5.2   Looking at the Data using ViewHAR

The input-output data used in the Stylized Johansen model is contained in the data file GAMSGP.DAT. This is a special GEMPACK *binary* file - called a **Header Array file**[28] - so you cannot just look at it in a text editor. Instead you have to use one of the programs for reading Header Array files, called ViewHAR. Select from the main WinGEM menu :

HA Files | View VIEWHAR

The ViewHAR window will appear. Click on   *File | Open...*   and select the file GAMSGP.DAT. This will open the file and show its contents on the contents screen.

Each row corresponds to a different array of data. Look at the column under the heading Name to see what these arrays are.

	Header	Type	Size	Name
1	**CINP**	**RE**	**SECTxSECT**	**Intermediate inputs of commodities to ind..**
2	FINP	RE	SECTxFACT	Intermediate inputs of primary factors - dollar
3	HCON	RE	SECT	Household use of commodities - dollar values

The first array is the "Intermediate inputs of commodities to industries - dollar values".
The Header **CINP** is just a label for this array. (Headers can have up to 4 characters.)
The array is of Type **RE.** The **R** means this is an array of real numbers. The **E** means that this array has set and element labelling (see chapter 5 of GEMPACK document GPD-4 – see section 6.1).

Double click on   **CINP**   to see the numbers in this array.

Compare these numbers with the input-output data for the model as shown in the TABLEs in GAMSGP.GMS.

---

[28] For this reason these files often have the extension .HAR.

Input-Output Data Base for GAMSGP

		Intermediate Inputs to			
		Industry 1	Industry 2	Households	Total Sales
Sectors	Commodity 1	4.0	2.0	2.0	8.0
	Commodity 2	2.0	6.0	4.0	12.0
Factors	Labor	1.0	3.0		4.0
	Capital	1.0	1.0		2.0
Total Costs		8.0	12.0	6.0	

To return to the Contents Screen, click on *Contents* in the ViewHAR menu.

Look at the other Header Arrays called **FINP** and **HCON** to see where their numbers fit in the input-output data base.

Close ViewHAR in the normal Windows way *File | Exit*.

3.5.3 Looking at the Command file (Closure, Shocks)

The Command file GAMSGP.CMF is where the closure (that is, which variables are exogenous and which are endogenous) and shocks are specified. To look at this file, select

File | Edit file…

from WinGEM's File menu and then open this file GAMSGP.CMF.

- Can you see where the closure is specified? Note that just one variable is specified as exogenous, namely the variable p_XFAC. This is not quite the name of the variable in the TAB file. There it is called simply XFAC. Here the "p_" at the front indicates that it is the associated linear variable (representing the percent change in XFAC). [When a levels variable (eg XFAC) is declared, TABLO automatically defines an associated linearized variable which has a "p_" at the front of the otherwise identical name (hence p_XFAC). Note that GEMPACK only allows you to refer to linearized variables (rather than the corresponding levels variables) in the Command file.[29]]

- Can you see where the shock is specified? Again the associated linear name is used. The "labor" in quotes indicates just the labor endowment part (but not the capital endowment) is changed.

- What Solution file will GEMSIM put the results (percentage changes) in? The convention is that, unless specified otherwise, the name of the Solution file is the same as the name of the Command file, but with the standard Solution file suffix **.SL4** . So here the Solution file will be called GAMSGP.SL4.

- What is the name of the updated data file? You can see the statement
  ```
  updated file iodate = <cmf>.upd ;
  ```

---

[29] You can find more details about the way TABLO linearizes levels equations in section 2.2 of GPD-2 (see the GEMPACK documents in section 6.1 at the end).

Here the **<cmf>** means the name of the Command file (except for the suffix .CMF). Hence the updated data file will be called GAMSGP.UPD.

- How does this Command file know to attach to the equations etc in GAMSGP.TAB (rather than to some other model in a different TAB file)? This is indicated by the statement
   **Auxiliary files = gamsgp ;**
    which refers to the GEMSIM Auxiliary files which are produced when TABLO is run (see section 3.6.1 below).

Now that you have had a quick look at the TAB file, the data and the Command file, we suggest that you proceed to the next section where you will use GEMPACK to carry out the same simulation as you carried out with GAMS in section 3.2 above.

3.6   The Simulation Using GEMPACK

Now it is time for you to carry out the same simulation using GEMPACK. You are going to do this in three steps using the following items from WinGEM's **Simulation** menu:

TABLO Implement
GEMSIM Solve
View Solution (ViewSOL)
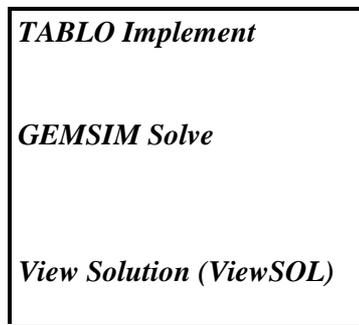
TABLO, GEMSIM and ViewSOL are the names of programs used to carry out the three steps of a simulation :

Step 1 - Implement the model

Step 2 - Solve the equations of the model

Step 3 - Look at the results

WinGEM will guide you through these steps and indicate what to do next.

3.6.1   Step 1 – Implement the Model using TABLO

To begin, go back to WinGEM and select

Simulation | TABLO Implement…

from WinGEM's Simulation menu.

A window for TABLO will appear. Click on the *Select* button to select the name of the TABLO Input file **GAMSGP.TAB** (which you have looked at earlier in TABmate). This is all TABLO needs to implement the model.

In the menu for the TABLO window, select the *Options* menu item. Then in this menu choose

TABLO Options...

A new TABLO Options window will appear giving many and varied special options for TABLO. Ignore all options for now except the two in the top left hand corner which read

```
O    PGS   Generate GEMSIM auxiliary files

O    WFP   Generate Fortran code
```

There is a black dot in one of the buttons. Click on the first option **PGS** because we want you to create the GEMSIM Auxiliary files. Then click on the *Ok* button to return to the TABLO window.

By "implement" we mean convert the TABLO Input file into binary computer files which are used in the simulation program GEMSIM in the next step. These files are referred to as Auxiliary files (or sometimes as the GEMSIM Statement and Table files) and in this case, are called GAMSGP.GSS and GAMSGP.GST.

Click on the *Run* button. The program TABLO runs in a DOS box and when complete returns you to the TABLO window with the names of files it has created: the Information file GAMSGP.INF and the Log file. Look at both of these files by clicking the *View* buttons beside them.

The Information file gives information about the TABLO Input file such as whether there are any syntax or semantic errors found by the program TABLO when it was checking the TABLO Input file. Search the file for **%%** to see if there are any errors. Search the file for "syntax error" to see how many syntax errors and semantic problems there are (hopefully none). Go to the end of the file to see what actions GEMSIM can carry out with the Auxiliary files produced in this TABLO run.

When you have looked at these two files, click on the *Go to GEMSIM* button at the bottom of the TABLO window to go on to the next step in running a simulation: Step 2 - Solve the equations of the model.

3.6.2   Step 2 - Solve the equations of the model using GEMSIM

The *Go To GEMSIM* button takes you to the GEMSIM window. (Alternatively you can start this window by choosing *Simulation | GEMSIM Solve* from WinGEM's main menu.)

First *Select* the Command file called GAMSGP.CMF. Since Command files are text files. Look at this Command file in the text editor by clicking the *Edit* button.

How is the closure specified? What shock is applied? What data file is used by this model? How many steps are used in the multi-step solution?[30]

(*File | Exit* to return to the GEMSIM window.)

Click on *Run* to run GEMSIM with the Command file GAMSGP.CMF.

[If there is an error, view the Log file, and try to rectify the error.]

There is no point in trying to look at the Solution file directly because it is a binary file, not a text file. There are two ways of looking at the Solution results. The most common is to view them with ViewSOL, the Windows solution viewer, which we suggest you do in the section below.[31]

---

[30] More details about GEMPACK Command files can be found in section 2.8 of GEMPACK document GPD-1 – see section 6.1.

[31] The other is to use program GEMPIE to convert the solutions into a Print file that can be edited or printed out. You may wish to try this alternative (via the "Go to GEMPIE" button) later on.

3.6.3   Step 3 - Look at the results with ViewSOL

Click on the button *Go to ViewSOL*   This will run ViewSOL and open the Solution file GAMSGP.SL4 created on the previous example.[32]

You will see the **Contents** page listing many of the variables of the model. ViewSOL has three slightly different formats for this Contents list. Select *Format...* from ViewSOL's main menu and there click on *Arrange vectors by name* (in the panel headed Vector options); then click *Ok* which will put you back to the Contents list.

To see the results of one of these variables listed by name, just **double-click** on the corresponding row in the Contents list. First double-click on the **p_XCOM** row to see the results for this variable (demand for the two commodities). Select 3 decimal places (see the third drop-down list box along the top row of the current ViewSOL window - the only one with a single figure in it). Then you should see something like the following:[33]

```
p_XCOM       gamsgp       Pre gamsgp  Post gamsgp Chng gamsgp
s1           5.885        8.000       8.471             0.471
s2           6.899        12.000      12.828            0.828
```

Across the **s1** row you see the percentage change result (5.885%), the pre-simulation levels value (8.000), the post-simulation levels value (8.471) and the change (0.471); these are the results for the total supply of commodity **s1**.

Then click on *Contents* to return to the Contents list.

To see the **p_XFAC** results, double-click on this row. You will see

```
p_XFAC       gamsgp       Pre gamsgp  Post gamsgp Chng gamsgp
labor        10.000       4.000       4.400             0.400
capital      0            2.000       2.000             0
```

This time all the numbers are in red which is used to remind you that, for this simulation, both components of this variable p_XFAC (percent change in factor endowments) are exogenous. You should easily be able to understand all of these results.

Then click on *Contents* to return to the Contents list.

To see the **p_XC(i,j)** results [intermediate inputs of commodity i into industry j], double-click on this row. Now you see

```
p_XC  s1           s2
s1    5.885        5.885
s2    6.899        6.899
```

and, in the second drop-down box you should see "**1 gamsgp**". This indicates that you are seeing the linearized simulation results (the percentage changes in the four components of this variable). You cannot see the pre- and post-simulation levels results at the same time since this variable p_XC is a matrix variable. To see the pre-simulation levels results, click on the second drop-down list box (the one showing "1 gamsgp") and select the second alternative ("2 Pre gamsgp"). Then you will see the pre-simulation levels results. You might also like to look at the post-simulation levels results and the changes. When you have finished looking at these results, click on *Contents* to return to the Contents list.

---

[32] An alternative way of running ViewSOL is to select from the main WinGEM menu *Other tasks* / *View Solution (ViewSOL)*   and open the Solution file you want to view.

[33] If you do not see all columns below, you probably need to select the appropriate options for ViewSOL. To do this, select *File* / *Options* from ViewSOL's main menu and then make sure that option **Show levels results (if present)** is checked. Then close ViewSOL and reopen it.

When you have finished looking at the results for other variables, exit from ViewSOL. This will return you to the screen for the simulation.

3.6.4 Looking at the Updated Data

You can look at the updated data (that is, the new version of the data base reflecting the new primary factor endowments) by clicking on the button labelled *View Input/Output Files* on the GEMSIM window (which should still be visible). Select the second option there (the one mentioning **updated iodata**). ViewHAR will run and load the updated data.

The line

```
    updated file iodata = <cmf>.upd ;
```

in the Command file GAMSGP.CMF (see section 3.5.3) means that, when you run the simulation, the software produces the so-called **updated data file GAMSGP.UPD**. This file contains the data as it would be after the shocks (in this case, the increase in labor supply) have worked their way through the model. This data represents the state of the economy after the shocks, which is why the updated data is sometimes referred to as the **post-simulation data**.

If you look at the various arrays of data, you should see the values as in the Table below (which are shown to 3 decimal places).

		Industry		Households	Total Sales
		1	2		
Commodity	1	4.235	2.118	2.118	8.471
Sectors					
Commodity	2	2.118	6.353	4.235	12.707
Labor	3	1.059	3.177		4.235
Factors					
Capital	4	1.059	1.059		2.118
Total Production		8.471	12.707	6.353	

Updated Input-output Data in File GAMSGP.UPD

You can check that these post-simulation values are consistent with the results of the simulation as seen in ViewSOL above. For example, the p_DVHOUS results show that the value of household expenditure on commodity s2 increased by 5.885 percent from its pre-simulation value of 4 to its post-simulation value of 4.235 (which agrees with the Commodity 2 Households value in the table above).

3.7 Comparing GAMS and GEMPACK Results

You have probably been comparing the GAMS and GEMPACK results as you have been going along. Here we take you through a few comparisons to reassure you that the results are indeed the same, and also to increase your familiarity with the different ways in which GEMPACK makes the results available.

XCOM results: As you saw above via ViewSOL, demand for the first commodity [p_XCOM("s1")] has increased by 5.885 percent respectively, namely from a pre-simulation value of 8 to a post-simulation value of 8.471. This agrees with the GAMS results in GAMSGP.LST which shows the value 5.885 for XCOM100("s1"). Also you can see from the "VAR XCOM" results in GAMSGP.LST that the post-simulation value of XCOM("s1") is 8.471. You can make similar checks for commodity s2.

DVHOUS results for commodity s1. You can see the results, namely

	p_DVHOUS	gamsgp	Pre gamsgp	Post gamsgp	Chng gamsgp
s1	5.885	2.000	2.118	0.118	

You can see the same from the GAMS report for DVHOUS_new which shows the same value of 2.118 for s1. You can see this a different way from the updated data file GAMSGP.UPD (see section 3.6.4) which shows (via ViewHAR) the same value of 2.118.

You might like to check other results in similar ways.

3.8 A Second Simulation (GAMSGPB.CMF)

Suppose that you wanted to do a different simulation, this time increasing the endowment of capital by 3% at the same time. In GAMS you would add a second solve at the end of GAMSGP.GMS. In GEMPACK the model itself has not changed so you make no change to the TAB file. Instead you need a new Command file to set out the different shocks. We have prepared GAMSGPB.CMF (which we did by first copying GAMSGP.CMF and then making appropriate changes). The only change from GAMSGP.CMF is to add the extra "shock" statement[34]

Shock p_XFAC("capital") = 3 ;

How do you carry out this simulation in GEMPACK? In the first example (GAMSGP.CMF) there were three steps. Here there is no need to repeat the first step since the TAB file has not changed. Thus it is just necessary to run GEMSIM (Step 2) and look at the results (Step 3).

Proceed as in section 3.6.2 above to run GEMSIM by selecting

Simulation | GEMSIM Solve

from WinGEM's Simulation menu. This time select Command file GAMSGPB.CMF instead of GAMSGP.CMF. Then look at the results via the **Go to ViewSOL** button. Check for example that demand for the first commodity increases by 7.145 per cent (the p_XCOM("s1") result).

This partly explains the separation made in GEMPACK between the TAB file (theory of the model) and the Command file (closure and shocks) and why GEMPACK has two separate programs, TABLO for processing TAB files and GEMSIM for carrying out simulations. It is only necessary to run TABLO when the TAB file is changed.[35]

3.9 A Second TAB File GAMSGPM.TAB (Mixed Levels/Linear Version)

GEMPACK allows either levels equations or their linearized versions in TAB files. For historical reasons, many models implemented in GEMPACK have mostly (or entirely) linearized equations.[36]

We believe that there are good reasons for

- always writing down accounting identities as levels equations,
- writing behavioural equations (such as CES) as linearized equations.

In a nutshell, the linearized equations are simpler and more useful than the levels version when interpreting results. We give a couple of examples later in this section. See Harrison *et al* (1994) for details.

---

[34] We also changed the "verbal description" to be an accurate description of the shocks.

[35] In contrast, if you add extra Solve statements to a GAMS file, the program GAMS.EXE checks the whole model again before it processes the extra Solve statements.

[36] GEMPACK only allowed linearized equations in TAB files before Release 5.1 (April 1994). The well-known ORANI, GTAP and MONASH models have all linearized equations in their TAB files.

If you wish to work with an existing GEMPACK model (perhaps you need to add some equations to it), you need to be familiar with the idea of linearizing equations. For this reason we have prepared a second version of the TAB file for the GAMSGP model. This second version is called **GAMSGPM.TAB** (where the **M** at the end serves as a reminder that this is what we often refer to as a **Mixed levels/linear TAB file**). The accounting equations in GAMSGPM.TAB are written down in the levels (exactly as in GAMSGP.TAB) but the behavioural equations (for example, the equation for factor demands) are written as linearized equations in the TAB file.

In this section we

- draw your attention to the differences between GAMSGP.TAB and GAMSGPM.TAB,

- tell you briefly how to carry out the same simulation as above with this second version of the model.

3.9.1 Differences Between GAMSGPM.TAB and GAMSGP.TAB

Start WinGEM running and select *File | Edit file* then open GAMSGPM.TAB. You will see that we have produced GAMSGPM.TAB from GAMSGP.TAB so that all the GAMSGP.TAB statements are present in GAMSGPM.TAB, although some may be commented out when they are replaced by a different statement in GAMSGPM.TAB. The main changes in GAMSGPM.TAB - the (linearized) behavioural equations - come in the section containing the equations. To find this, search for **Consumer_demands** . . We explain below how the linearized version of the **Consumer_demands** equation has been obtained from the levels version.

In GTAPGP1.TAB this equation is

```
EQUATION Consumer_demands
  (all,i,SECT)  XH(i) = ALPHAH(i)*Y/PC(i)  ;
```

The linearized version in GAMSGPM.TAB is

```
EQUATION(LINEAR)  Consumer_demands
  (all,i,SECT) p_XH(i) = p_Y - p_PC(i) ;
```

The linearization is based on the following two simple rules.

1. **If X = C*Y where C is a constant then p_X = p_Y.**

2. **If X = Y/Z then p_X = p_Y – p_Z. [Linearization of Ratio]**

The first says that if X is proportional to Y (where C is the constant of proportionality) then the percentage change in X equals the percentage change in Y. The second says that the percentage change in a ratio is the difference between the percentage changes. The first of these rules is exact in the sense that it is exactly true no matter how large the changes in Y and X are.

The second rule is only an approximation, indeed a better approximation for small changes than for large changes.[37] Nevertheless, GEMPACK is still able to obtain accurate solutions of the underlying (nonlinear) levels equations of the model by breaking down changes into a series of small changes as explained in section 2.9 above. You will see this when you carry out the same simulation using GAMSGPM.TAB.

---

[37] To see that, suppose that the initial values of Y and Z are 200 and 100 respectively so that the initial value of X is 2. If Y increases by 2 percent and Z increases by 1 percent then the new value of X is 204/101 = 2.0198. This is an increase of 0.099 percent in X whereas the rule 2 above suggests that there would be an increase of (2-1)=1%.
Suppose alternatively that Y increases by 60 percent and Z increases by 10 percent. Then the new value of X is 320/110 = 2.9091. This is an increase of 45.46 percent in X whereas the rule 2 above suggests that there would be an increase of (60-50)=50%.

The linearization of the Consumer_demands equation illustrates why the linearized equations are sometimes simpler to implement than the corresponding levels equations. In this case the parameter ALPHAH(i) which appears in the levels version disappears from the linearized version. This is why the calibration phase, which is so important for levels models, is somewhat less important for models written down via linearized versions of the equations.

To see more examples of this look below in GAMSGPM.TAB at the linearized equations which follow the Consumer_demands equations. You can see how each of these follows from the corresponding levels equation using the two rules above plus the following rule

3. If X = Y*Z then p_X = p_Y + p_Z. [Linearization of Product]

Another change between GAMSGP.TAB and GAMSGPM.TAB is that the scale parameters Q(j) are not needed in the linearized equations. [Although these are present in the levels versions of equations Price_formation, they are not needed in the linearized version of this equations as you can check.]

3.9.2   The Simulation using GAMSGPM.TAB

Here we describe briefly how to carry out the same simulation as above (section 3.6) using GAMSGPM.TAB and associated Command file GAMSGPM.CMF.

First you might like to look at the Command file GAMSGPM.CMF. You will see that the only change (apart from changes in the comments) from GAMSGP.CMF is in the "auxiliary files" statement where we now point to GAMSGPM.TAB instead of GAMSGP.TAB.

The procedure for running the simulation is very similar to that in section 3.6 above.

Step 1. Proceed as in section 3.6.1 above, using GAMSGPM.TAB instead of GAMSGP.TAB

Step 2. Proceed as in section 3.6.2 above, using GAMSGPM.CMF instead of GAMSGP.CMF.

To look at the results, proceed as in section 3.6.3 above. Compare the results with those from the GAMSGP.CMF simulation. You should find that they are the same.

3.9.3   Linearizing Other Equations

Models implemented via GEMPACK were traditionally written down using linearized equations. Accordingly there are readily available references which document how to linearize the standard GE model functions such as CES and the LES (Linear Expenditure System). Modelers do not have to work out the linearization each time. See for example, Horridge *et al* (1993) for details about linearizing the various standard GE functions which occur in the ORANI model.

One advantage of using the linearized equations is that some of the scale parameters which are required (for example in the levels versions of CES) do not appear in the corresponding linearized versions. [Parameters Q(j) and ALPHAH(j) in GAMSGP.TAB are examples – see section 3.9.1 above.] In standard GE models such as ORANI and GTAP, the main parameters required are various sorts of substitution elasticities.

If you want to understand in more detail how some linearized equations have been worked out, you may like to read Chapter 9 of GPD-2 (see section 6.1).

3.10 A Third TAB File GAMSGPLN.TAB (All Equations Linearized)

GEMPACK allows either levels equations or their linearized versions in TAB files.

As we indicated earlier, for historical reasons, many models implemented in GEMPACK have mostly (or entirely) linearized equations. This means that, if you wish to work with an existing GEMPACK model (perhaps you need to add some equations to it), you need to be familiar with TAB files containing only linearized equations.

For this reason we have prepared a third version of the TAB file for the GAMSGP model. This third version is called **GAMSGPLN.TAB** (where the **LN** at the end serves as a reminder that this is what we often refer to as a **Linearized TAB file**). The crucial difference is that both the accounting equations and the behavioural equations in GAMSGPLN.TAB are written down as linearized equations. Since accounting equations are usually sums, we need to linearize sums. Another consequence is the use of update statements, which will be explained below.

In this section we

- show you how to linearize equations which contain sums,

- explain the concept of update statements,

- draw your attention to the differences between the mixed version of the model in GAMSGPM.TAB and the linearized version in GAMSGPLN.TAB,

- tell you briefly how to carry out the same simulation as above with this third version of the model.

3.10.1 Linearizing Sums

There are two ways to linearize sums, which will be illustrated using the levels equation Factor_use from GAMSGPM.TAB

```
XFAC(f) = SUM(j,SECT, XF(f,j)).
```

Firstly differentiate to note that

```
c_XFAC(f) = SUM(j,SECT, c_XF(f,j))        (*)
```

where "c_" denotes "change in". [That is, the change in a SUM is just the SUM of the changes.] Then, note that, for any variable Z, the change c_Z and the percentage change p_Z in Z are related via the identity

```
p_Z = 100*(c_Z/Z)    or      c_Z = Z * p_Z /100
```

If you make this replacement on both sides of the equation (*) above and multiply both sides by 100 you see that

```
XFAC(f)*p_XFAC(f) = SUM(j,SECT, XF(f,j)*p_XF(f,j) )
```

Now multiply both sides by the common price PFAC(f) to see that

```
PFAC(f)*XFAC(f) *p_XFAC(f)  = SUM(j,SECT, PFAC(f)*XF(f,j)*p_XF(f,j))
```

However PFAC(f)*XFAC(f) just equals what we have called DVFAC(f) and PFAC(f)*XF(f,j) equals what we have called DVFACIN(f,j). Hence

```
DVFAC(f) * p_XFAC(f) = SUM(j,SECT, DVFACIN(f,j) * p_XF(f,j) )
```

which is the linearized version of the Factor_use equation in GAMSGPLN.TAB.[38]

Alternatively, in many linearized TAB files, accounting equations are linearized using shares. If you take the linearized version

```
DVFAC(f)*p_XFAC(f) = SUM(j,SECT, DVFACIN(f,j)*p_XF(f,j) )
```

(see above) and divide both sides by DVFAC(f) then this becomes

```
p_XFAC(f) = SUM(j,SECT, SHFACIN(f,j) * p_XF(f,j) )
```

---

[38] This equation has the following easy interpretation. The left-hand side is just 100 times the change in the quantity XFAC(f) at base period prices while the right-hand side is the sum of 100 times the changes in the quantitites XF(f,j) also at base period prices.

where SHFACIN(f,j) = DVFACIN(f,j)/DVFAC(f) is the **share** of industry j in the value of demand for factor f. In some linearized TAB files you will see the share form of equations while in other ones you will see the change form (as we have included for Factor_use in GAMSGPLN.TAB).

3.10.2 Update Statements

A notable feature of both forms of linearized equations above is that they directly or indirectly contain levels terms (in our example DVFACIN and DVFAC or SHFACIN, which is a quotient of the two). Since there are no levels variables in a fully linearized model, these need to be declared as **COEFFICIENTs**.

With every step of the simulation, the values of these coefficients must reflect changes of the underlying variables. Look for example at DVFACIN, which is defined by DVFACIN(f,j) = PF(f) * XF(f,j). Any change in PF(f) and XF(f,j) must be reflected in the value of DVFACIN when it is used in the next step of the simulation. Since DVFACIN is read from the database, we need to update the database in every step of the simulation.

The way to do this is by an UPDATE statement. Update statements indicate how something read from the database is affected by changes in the relevant variables of the model (usually the relevant price and quantity). For example

```
UPDATE (all,f,FAC)(all,j,SEC) DVFACIN(f,j) = p_PF(f) * p_XF(f,j) ;
```

translates changes in p_PF and P_XF into changes of DVFACIN in the database. The use of ***** (multiplication) on the right-hand side of the Update statement is because, in the levels, the price and quantity are multiplied. TABLO converts this multiplication to the appropriate addition of percentage changes used to update the value of DVFACIN after small changes in the relevant price and quantity.[39]

Understanding these two basic concepts, you are now ready to look at the differences between GAMSGPLN.TAB and GAMSGPM.TAB.

3.10.3 Differences Between GAMSGPLN.TAB and GAMSGPM.TAB

Start WinGEM running and select *File | Edit file* then open GAMSGPLN.TAB.

Again GAMSGPLN.TAB contains all statements from its precursors GAMSGP.GMS, GAMSGP.TAB and GAMSGPM.TAB, possibly commented out. This enables you to trace the development all the way from the GAMS file to the linearized file GAMSGPLN.TAB.

The first main difference is that the levels variables in GAMSGP.TAB and GAMSGPM.TAB are replaced by linearized versions. Search for **p_Y** and compare the declarations of the linearized versions with the declaration of the levels variables. Note the new "qualifier" ORIG_LEVEL= in these declarations. This enables GEMPACK to report the pre-simulation values (and hence the post-simulation values and the changes) when reporting simulation results. [You will see these when you look at simulation results from GAMSGPLN.TAB via ViewSOL – see section 3.10.4 below.]

Now search for **DVCOMIN**. You will see that what were levels variables in GAMSGPM.TAB (namely the dollar values DVCOMIN, DVFACIN and DVHOUS) are now declared as **Coefficient**s rather than Variables. This is typical of linearized TAB files. Percentage changes in prices (for example, p_PC(i)) and quantities (for example, p_XCOM(i)) are linear Variables while dollar values (read from the data base) are just Coefficients (not linearized variables).

---

[39] TABLO converts this statement to the fomula
DVFACIN_new(f,j) = DVFACIN_old(f,j)*[1 + p_PF(f)/100 + p_XF(f,j)/100] .

Follow down GAMSGPLN.TAB. You will see that the DVCOMIN, DVFACIN and DVHOUS values are read as before.

Just below the Read statements, search for **PC(i)**. You will see that the formulas in GAMSGP.TAB and GAMSGPM.TAB setting these prices to 1 are not needed in GAMSGPLN.TAB. Instead, the initial price level is indicated by the qualifier ORIG_LEVEL.

Now search for **UPDATE**. You will see three Update statements. These have been discussed in section 3.10.2 above.

Next come the linearized versions of the accounting equations from GAMSGPM.TAB. To see these, search for **Com_clear**. You will see that the Formula & Equations Com_clear and Factor_use from GAMSGPM.TAB have been replaced by

- firstly Formulas calculating DVCOM and DVFAC (values of total commodity output and factor use). These Formulas are carried out at every step of a multi-step calculation. After a small part of the shock is applied, the data base values (DVCOMIN, DVFACIN, DVHOUS) are updated using the Update statements earlier. Then these Formulas ensure that the values of DVCOM and DVFAC reflect these updated values. The updated values go into the linearized equations (for example Equation Com_clear).

- secondly linearized Equations involving these dollar values as coefficients and the percentage changes in the relevant quantities. Again the above is typical of what happens in a linearized TAB file. Dollar values deduced from the data base (for example, value of total demands DVCOM) are declared as Coefficients (rather than levels variables) and calculated at each step via the relevant Formulas.

If you go down a little in GAMSGPLN.TAB you will see a similar change in the status of **Y** (Total nominal household expenditure). In the linearized file it is a Coefficient and its values are calculated via a Formula.

The remaining equations in GAMSGPLN.TAB are the behavioural equations. These are the same as in GAMSGPM.TAB since they were already linearized in GAMSGPM.TAB.

This completes a brief tour of the differences between GAMSGPLN.TAB and GAMSGPM.TAB.

3.10.4 The Simulation using GAMSGPLN.TAB

Here we describe briefly how to carry out the same simulation as above (section 3.6) using GAMSGPLN.TAB and associated Command file GAMSGPLN.CMF. The procedure is very similar to that in section 3.6 above.

Step 1. Proceed as in section 3.6.1 above, using GAMSGPLN.TAB instead of GAMSGP.TAB

Step 2. Proceed as in section 3.6.2 above, using GAMSGPLN.CMF instead of GAMSGP.CMF.

To look at the results, proceed as in section 3.6.3 above. Compare the results with those from the GAMSGP.CMF simulation. You should find that they are the same.

3.10.5 How Update Statements Affect Linearized Equations – An Example

The following section shows how the linearized equations change from step to step of a multi-step calculation. It is quite technical and can be skipped without loss of understanding.

Consider the linearized Factor_use equation for factor labor, namely

```
DVFAC("labor")*p_XFAC("labor") = SUM(j,SECT,DVFACIN("labor",j)*p_XF("labor",j))
```

In the initial database, DVFAC("labor") = 4, DVFACIN("labor","s1") = 1 and DVFACIN("labor","s2") = 3. Thus, in the first step of the multi-step simulation, this equation says

4 * p_XFAC("labor") = 1 * p_DVFACIN("labor","s1") + 3 * p_DVFACIN("labor","s2")

Here the numbers 4,1,3 are the coefficients while p_XFAC("labor"), p_DVFACIN("labor","s1") and p_DVFACIN("labor","s2") are the variables.[40]

The values of DVFACIN("labor",j) occurring in this equation are affected by the Update statement

```
 Update (all,f,FAC)(all,j,SEC) DVFACIN(f,j) = p_PF(f) * p_XF(f,j) ;
```

During the first step of a multi-step simulation, the system of linear equations is solved to produce results for the linearized variables. Suppose that, as a result of solving these equations, p_PF("labor") = 1, p_XF("labor","s1") = 2 and p_XF("labor","s2") = 3. Then, as a result of the Update statement above, the values of DVFACIN("labor",j) at the start of the second step of the multi-step calculation are

DVFACIN("labor","s1") = 1.03 and DVFACIN("labor","s2") = 3.12

since the Update statement above converts to the fomula

```
DVFACIN_new(f,j) = DVFACIN_old(f,j)*[ 1 + p_PF(f)/100 + p_XF(f,j)/100] .
```

Then the value of DVFAC("labor") at the start of this second step is 4.15, as implied by the formula

```
 (all,f,FAC) DVFAC(f) = SUM(j,SECT, DVFACIN(f,j)) ;
```

Hence in the second step of the simulation, the equation above becomes

 4.15 * p_XFAC("labor") = 1.03 * p_DVFACIN("labor","s1") + 3.12 * p_DVFACIN("labor","s2")

As you can see the slope of this equation is slightly different in step 2 than it was in step 1. This is how the GEMPACK solution algorithm is able to follow a nonlinear solution path even though only linear equations are solved at each step.

3.11 Levels, Mixed or Linearized TAB Files – Which is Better?

You have seen 3 different TAB files each implementing the same model. There seems to be a consensus amongst GEMPACK users that the linearized form of the behavioural equations are the easiest to interpret. Some have argued [see Harrison *et al* (1994)] that the accounting equations are easiest to understand and implement in levels form.

In the end, it is a matter of personal preference which you decide to use.

---

[40] This would seem more familiar if we used x,y,z for the variables as in "4x = y + 3z". This example also helps to explain why quantities such as DVFAC(f) are called "Coefficients" in the TAB file.

4   Other Features of GEMPACK

4.1   TABLO-generated Programs and GEMSIM

This feature of GEMPACK is sometimes difficult for new users to understand. When TABLO is run to process the TAB file MODEL.TAB, the output files can be for GEMSIM or they can be a so-called TABLO-generated program.

GEMSIM (like GAMS.EXE) is a general-purpose program which can solve any GE model implemented in a TAB file. Why then is there a need for the alternative TABLO-generated program?

For each TAB file MODEL.TAB, the associated TABLO-generated program is a Fortran program MODEL.FOR which is written by TABLO.EXE. This program is not general-purpose, instead being specific to the model in MODEL.TAB. When MODEL.FOR is compiled and linked to produce the executable image MODEL.EXE, this program MODEL.EXE can solve the model in MODEL.TAB (but not other models).

The reason for going to all this trouble is that the TABLO-generated program can solve the model noticeably more quickly than GEMSIM. For moderately large models, the speed-up is significant. For example (see Chapter 4 of GPD-8 for more details),

- for the 25-sector MONASH model, GEMSIM takes about 6 times longer than the TABLO-generated program.

- for a 122-sector ORANIG model, GEMSIM takes about 8 times as long as the TABLO-generated program.

The more complex the model, the greater is the speed-up from using a TABLO-generated program.

Note that, as long as MODEL.TAB does not change, many different simulations can be carried out with MODEL.EXE. In such a case, the one-off cost of compiling and linking MODEL.EXE is often small compared to the time saved using MODEL.EXE rather than GEMSIM.EXE.

4.2   Condensation

This is a feature which allows GEMPACK to solve very large models. The basic idea is to remove some variables by substituting for them an expression involving other variables. This can be illustrated by the equation for consumer demands presented above.

```
EQUATION Consumer_demands
  (all,i,SECT)  XH(i) = ALPHAH(i)*Y/PC(i)  ;
```

In the model the variable XH(i) can be replaced in all instances by the RHS of this equation. The variable XH(i) will not appear in the model any more. Thus the number of variables will be reduced by i and the i equations XH(i) can be dropped from the model.

TABLO can carry out algebraic manipulations of the symbolic equations to reduce the number of actual equations solved. You can find an introduction in section 3.9 of GPD-1. You can find a hands-on example in section 2.1.17 of GPD-8.

4.3   Systematic Sensitivity Analysis (SSA)

Part of analysing model results can involve carrying out systematic sensitivity calculations to determine how sensitive results are to the values of key model parameters (such as substitution elasticities) or to the values of the shocks.

In GAMS this can be done via multiple solves (see section 1.12 above), where the relevant parameters or shocks are explicitly varied from solve to solve.

In GEMPACK SSA is automated via RunGEM (see section 1.11 above). This provides a Windows interface for

- selecting the way in which the parameters or shocks vary. Distributions can be triangular or uniform.

- sampling from the relevant probability space. This uses Gaussian quadratures (see, for example, DeVust and Preckel (1997)).

- presenting the results. Means and standard deviations for all variables in the model are available via ViewSOL.

4.4   Inequalities and Complementarities

GAMS handles inequalities explicitly whereas GEMPACK only allows equation (not inequalities). In particular, GAMS models can incorporate complementarities readily, often using MCP (mixed complementarity) methods such as PATH (see Dirkse *et al* (1995)) to solve them.

In Release 8.0 of GEMPACK (to be officially released in 2002), a COMPLEMENTARITY statement is allowed in TAB files. This makes it possible to implement and solve complementarities (including volume quotas and tariff-rate quotas) easily in GEMPACK. The algorithm used to solve the model in such cases is different from the algorithms used by GAMS. The GEMPACK algorithm (suggested by Mark Horridge) works well even if many of the equations of the model are written as linearized equations in the TAB file. In contrast, the algorithms (for example, PATH) used by GAMS to solve MCP problems require all equations to be known in levels form.

For more details about the GEMPACK procedures for handling complementarities, see Harrison *et al* (2002) and the draft GEMPACK Release 8.0 documentation.

5   Summary and Outlook

This paper gives an introduction to GEMPACK for those who have previously done CGE modeling using GAMS. It shows that are many similarities between the two programming languages and explained some of the most important differences. This should help economists using GAMS to understand and modify models developed in the GEMPACK environment and improve the exchange between modelers using these two software packages.

The paper also describes some of the tools that GEMPACK provides to handle data and to interpret simulation results. You can find more details about the design and development of GEMPACK in Harrison and Pearson (1996).

If you want to learn more about GEMPACK and improve your skills in modeling in this environment, you might want to use GEMPACK with other models and to read the "official" GEMPACK documentation which consists of documents referred to as GPD-1 to GPD-4 and GPD-8 (see section 6.1).

5.1   A Short Guide to the GEMPACK documentation

Document GPD-1 "Introduction to GEMPACK" is the normal starting place for new users. Chapters 2 and 3 of GPD-1 introduce new users to carrying out simulations (Chapter 2) and TAB files (Chapter 3) using the model known as "the Stylized Johansen model". In fact GAMSGP you have been working with is exactly the same model – we have even used the same notation for the variables in GAMSGP. The only difference is that the TAB files referred to in Chapter 3 of GPD-1 as Stylized Johansen are called

- SJ.TAB (this corresponds to the mixed levels/linear version GAMSGPM.TAB),

- SJLV.TAB (this corresponds to the levels version GAMSGP.TAB), and

- SJLN.TAB (this corresponds to the linear version in GAMSGPLN.TAB).

To find out more about GEMPACK, you might like to do all or some of the following:

- Read section 4 which introduces some other features of GEMPACK.

- Look at Chapter 1 of GPD-8 which describes briefly some of the different models distributed with GEMPACK. If you wish to work with one of these you may find "hands-on" suggestions for getting started with the model in the relevant section in Chapter 2 of GPD-8.

- Read Chapters 2 and 3 of GPD-1. In particular section 2.8 of GPD-1 describes Command file statements in more detail, section 2.11 describes the way GEMPACK calculates accurate solutions, section 3.4 introduces the way data files such as GAMSGP.DAT are constructed and section 3.5 goes through the linear TAB file SJLN.TAB (ie, GAMSGPLN.TAB) in more detail.

- Learn how to use the Windows program RunGEM which makes it easy for new users to carry out simulations with models – see Chapter 5 of GPD-8.

5.2   Building or Modifying Models

If you want to build a model in GEMPACK you will probably choose to write down the levels equations in your TAB file since a levels TAB file is the closest to a GAMS file for the same model. However, the linearized form especially of behavioural equations might be easier to write and interpret than the levels form.

If you want to work with a GEMPACK model produced by someone else (possibly to add some features to it), you may find that you have to work with a TAB file in which all the equations are written down in linearized form.[41] In this case, new equations can be added as levels equations or as linearized equations. In many cases we recommend that you use levels equations, especially accounting or market clearing equations which are easier expressed as levels. Adding levels equations at the bottom of a linearized TAB file is discussed in some detail in Harrison and Pearson (1999).

---

[41] The best-known models implemented via GEMPACK have all equations written in linearized form. This includes ORANI, GTAP and MONASH.

6 References

Atkinson, K.E. (1989), *An Introduction to Numerical Analysis*, 2nd edition, Wiley, New York.

Brooke, T., D. Kendrick and A. Meeraus (1988), *GAMS: A User's Guide*, The Scientific Press, Redwood City, California.

DeVuyst, Eric A. and Paul V. Preckel (1997) "Sensitivity Analysis Revisited: A Quadrature-Based Approach", *Journal of Policy Modelling*, vol. 19, pp. 175-185.

Dirkse, S.P. and M.C. Ferris (1995), 'The PATH Solver: A Non-Monotone Stabilization Scheme for Mixed Complementarity Problems', *Optimization Methods and Software*, vol. 5, pp. 123-156.

Harrison, W.J., J.M. Horridge, K.R. Pearson and G. Wittwer (2002), 'A Practical Method for Explicitly Modeling Quotas and Other Complementarities', to be presented at the GTAP Conference, Taiwan, June 2002.

Harrison, W.J. and K.R. Pearson (1996), 'Computing Solutions for Large General Equilibrium Models Using GEMPACK', *Computational Economics*, vol. 9, pp.83-127.
[A preliminary version was *Impact Preliminary Working Paper* No. IP-64, Monash University, Clayton (June 1994), pp.55.]

Harrison, W.J. and K.R. Pearson (1999), 'Adding Accounting-Related Behaviour to a Model Implemented Using GEMPACK', available from the FAQ (Frequently Asked Questions) part of the GEMPACK web site.

Harrison, W.J., K.R. Pearson and A.A. Powell (1996), 'Features of Multiregional and Intertemporal AGE Modelling with GEMPACK', *Computational Economics*, vol. 9, pp. 331-353.
[A preliminary version was "Multiregional and intertemporal AGE modelling via GEMPACK", *Impact Preliminary Working Paper* No. IP-66, September 1994, pp. 21.]

Harrison, W.J., K.R. Pearson, A.A. Powell and E.J. Small (1994), 'Solving Applied General Equilibrium Models Represented as a Mixture of Linearized and Levels Equations', *Computational Economics*, vol. 7, pp. 203-223.

Harrison, W.J. and E.J. Small (1993b), 'TABLO Input Files for the Stylized Johansen, Miniature. ORANI and ORANI-F Models', *Impact Computing Document* No. C11-01, Monash University, Clayton (September 1993), pp.29.

Hertel, T.W., J.M. Horridge and K.R. Pearson (1992), 'Mending the Family Tree: A Reconciliation of the Linearized and Levels Schools of AGE Modelling', *Economic Modelling*, vol.9, pp.385-407.

Horridge, J.M., B.R. Parmenter and K.R. Pearson (1993), 'ORANI-F: A General Equilibrium Model of the Australian Economy', *Economic and Financial Computing*, vol.3, pp.71-140.

Pearson, K.R. (1991), 'Solving Nonlinear Economic Models Accurately via a Linear Representation', *Impact Preliminary Working Paper* No. IP-55, Melbourne (July), pp.39.

Press, W.H., B.P. Flannery, S.A. Teukolsky and W.T. Vetterling (1986), *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge.

Rutherford, Thomas F. (1999), 'Applied General Equilibrium Modeling with MPSGE as a GAMS Subsystem: An Overview of the Modeling Framework and Syntax', *Computational Economics*, vol. 14, pp. 1-46.

6.1 GEMPACK Documents (Release 7.0, October 2000)

Harrison, W.J. and K.R. Pearson (2000), *An Introduction to GEMPACK,* GEMPACK Document No. 1 **[GPD-1],** Monash University, Clayton, Fifth edition, October 2000.

Harrison, W.J. and K.R.Pearson (2000), *TABLO Reference,* GEMPACK Document No. 2 **[GPD-2]** Monash University, Clayton, Third edition, October 2000.

Harrison, W.J. and K.R.Pearson (2000), *Simulation Reference: GEMSIM, TABLO-generated Programs and SAGEM,* GEMPACK Document No. 3 **[GPD-3]**, Monash University, Clayton, First edition, October 2000.

Harrison, W.J. and K.R.Pearson (2000), *Useful GEMPACK Programs,* GEMPACK Document No. 4 **[GPD-4]** Monash University, Clayton, First edition , October 2000.

Harrison, W.J. and K.R. Pearson (2000), *Installing and Using the Source-Code Version of GEMPACK on DOS/Windows PCs with Lahey Fortran*, GEMPACK Document No. 6 **[GPD-6],** Monash University, Clayton, Tenth edition, October 2000.

Harrison, W.J. and K.R. Pearson (2000), *Installing and Using the Executable-Image Version of GEMPACK on DOS/Windows PCs,* GEMPACK Document No. 7 **[GPD-7],** Monash University, Clayton, Seventh edition, October 2000.

Harrison, W.J. and K.R. Pearson (2000), *Getting Started with GEMPACK: Hands-on Examples,* GEMPACK Document No. 8 **[GPD-8],** Monash University, Clayton, Second edition, October 2000.