Impact
Project

Impact Centre
The University of Melbourne
153 Barry Street, Carlton
Vic. 3053 Australia
Phones: (03) 344 7417
Telex: AA 35185 UNIMEL
Telegrams: UNIMELB, Parkville

IMPACT is an economic and demographic research project conducted by Commonwealth Government agencies in association with the Faculty of Economics and Commerce at The University of Melbourne, the Faculty of Economics and Commerce and the Departments of Economics in the Research Schools at the Australian National University and the School of Economics at La Trobe University.

AUTOMATING THE COMPUTATION OF SOLUTIONS OF LARGE ECONOMIC MODELS

by

K. R. Pearson
La Trobe University, University of Melbourne,
and the IMPACT Research Centre

Preliminary Working Paper No. IP-27 Melbourne March 1986

The views expressed in this paper do not necessarily reflect the opinions of the participating agencies, nor of the Commonwealth government

ISSN 0813 - 7986
ISBN 0 642 10024 1

Contents

ABSTRACT

The computer implementation of any large economic model is usually a very expensive and time consuming task. This paper describes a software package, called GEMPACK, which is being developed specifically to reduce dramatically the research time, effort and cost required to set up one solution method (the Johansen method) on a computer. Existing features of GEMPACK are described in detail, as are developments planned for the near future. The software is model-independent, in that it works for a wide class of economic models, irrespective of the form of the equations or underlying theory. GEMPACK is portable to most mini and main-frame computers, because it has been written in ANSI standard FORTRAN 77, with such portability as a primary requirement.

AUTOMATING THE COMPUTATION OF SOLUTIONS OF LARGE ECONOMIC MODELS

by

K. R. PEARSON

1. INTRODUCTION

The use of large non-linear models which require substantial computing in order to obtain numerical results is essential in many areas of economics; and especially so in computable general equilibrium (CGE) analysis, which is probably the fastest growing field in applied economics. Yet the computer implementation of any one model is usually a very expensive and time consuming task.

This paper outlines proposals for dramatically reducing the research time and effort required to set up one solution method (the Johansen method) on a computer. Software for doing a substantial part of this has already been developed. This software works equally well for many different models and is readily portable to most mini and main-frame computers. The currently available features of this software package (called GEMPACK, for General Equilibrium Model PACK) are described in section 4 while section 5 discusses planned developments of GEMPACK. In section 2, desirable features of software are discussed. Section 3 summarizes the steps in the development and use of a large model solved by Johansen's method. The major aim of GEMPACK is to allow economists to develop or use models with a minimum of concern for the computing details (which they may or may not be familiar with or interested in). Economists wishing to obtain GEMPACK for their own use will find information in section 6.

2. PORTABLE, MODEL-INDEPENDENT SOFTWARE

A computable economic model requires a body of theory, the derivation of equations capturing this theory and the assembly of sufficient appropriate data (e.g., behavioural and technological coefficients, and initial values of variables) as a basis for the computations. But even after the equations are written down, and their coefficients are assigned specific numerical values, normally much work is required before a large model can actually be implemented on a computer because large amounts of non-trivial software must be written for this purpose. (For example, it is estimated that the software written to implement the ORANI model of the Australian economy (see Dixon, Parmenter, Sutton and Vincent (1982), hereafter referred to as DPSV) took about 48 man-months.) Often the software is model-dependent in that it is relevant only to the particular model and is of little or no use when a second model is developed. Equally often, the software is machine-specific in that it is tailored to a specific computer and cannot be moved (except by an investment comparable with the initial writing) to a different computer. Clearly it is desirable to have software that is model-independent (i.e., applies equally well to all, or at least a large range of, models) and is portable in the sense that it can be moved to many other machines with minimal effort (certainly at least one order of magnitude less than that required to write the code initially). GEMPACK is a suite of software with these properties. It can be used to solve most large economic models (more precisely, those whose underlying functions are sufficiently smooth - being continuously differentiable is sufficient - for the Johansen linearization procedure to be a good approximation). It is portable because it is written in ANSI standard FORTRAN 77 (see ANSI (1978)) which is readily available on most mini and main-frame computers.

The Johansen method was chosen for GEMPACK mainly for the following two reasons:

(1) the desire to harness the simplicity of the linear mathematics inherent in the Johansen method and thereby to exploit relatively recently developed techniques called sparse matrix methods. The use of Johansen's method means that these are widely applicable to virtually any model and can be used as the basis of a procedure powerful enough to solve much larger systems than has previously been possible.

(2) a desire to produce a more efficient computational means of deriving solutions for the ORANI model of the Australian economy (see DPSV), where users routinely use Johansen style solutions.

That the Johansen method does not produce the exact numerical solution, because it approximates nonlinear equations by linear equations in percentage changes of the original variables, is something that must be kept not only in mind, but also in perspective. Firstly, experience at the Impact Project has shown that the numerical results obtained are sufficiently accurate to serve as a basis for sensible economic interpretation, where the different orders of magnitude, rather than the third and fourth decimal places, are what is vital. Secondly, there is always the possibility of eliminating linearization errors by using the so-called "large change" method whereby a large shock is broken into several smaller shocks which are applied one at a time. (See section 31.4 of DPSV for more details.)

The first version of the program SAGEM (which is the heart of the actual solving part of GEMPACK) was produced in 1984. It was model-independent in that, although it was aimed at ORANI, it worked equally well with other models. The first test was a cliometric application in which Mahinda Siriwardana used SAGEM to solve his model of the economy of Victoria (now an Australian state) in 1880 (Siriwardana (1985)); it is conservatively estimated that the computer work required to solve his model was reduced by several man-months by the availability of SAGEM. Since then SAGEM has also been used to solve MO (a skeletal version of the ORANI model (see DPSV sections 3-7)) and various regional models (Parmenter, Pearson and Jagielski (1984)). However this first version was not portable because it used file structures specific to VAX11 computers. This was rectified with new portable code for GEMPACK written in 1985. This code has been ported to a PRIME computer (late 1985) and to Pyramid and NAS computers (early 1986).

Another software package, MPS/GE, aimed at automating the solution of general equilibrium models, has been developed recently by Rutherford (see Rutherford (1985a) and (1985b)). The broad aims of MPS/GE are very similar to those of GEMPACK, but the detailed implementation differs considerably. MPS/GE uses Mathiesen's Sequential Linear Complementarity algorithm (see Mathiesen (1985)) to obtain equilibrium prices and activities. There are restrictions on the allowed production and utility functions, but most of the commonly used functions are catered for.

3. DEVELOPMENT AND USE OF A LARGE MODEL

There are two important stages in the life of a computable economic model. The first is the development of the model (its theory, equations and data) and its implementation on a computer. The second is its use by economists for simulations, policy analysis, forecasting, etc. Below, the economists responsible for the first phase are referred to as model developers and those involved in the second as model users. Clearly all model developers are model users, but not vice-versa.

For developers, the main attraction of the package will be the elimination of considerable delay and much tedious work normally encountered after model formulation and data assembly. The frustrating requirement, so commonly encountered, for major code redevelopment to accommodate essentially minor changes in economic specification, will be largely eliminated. For users, the friendliness and flexibility of the package will be its major advantages. For developers and users alike, the minimal computer familiarity required will free intellectual energies for effort more closely related to the economics of the problem. Moreover, the model-independence of the code means that once an economist has used it with one model, he/she should have no further computational hurdles to jump before using it with a second model.

Below are listed various steps in the design, implementation and Johansen solution of any large model. Note that the model developer begins at step (1) and the model user enters about step (7).

(1) Development of the theory underlying the model.

(2) Formulation of the equations - nonlinear form.

(3) Formulation of the equations - linear form. (The system produced is called the original system of equations.)

(4) Condensation of the system of linear equations. (This is necessary because many models have far too many variables in them to be solved for, even on massive computers. For example, the full ORANI model has several million variables. Thus condensation involves the elimination of many variables and the absorption of others into condensed variables.)

(5) Assembly of the standard data base (including parameter values).

(6) Calculation (from the standard data base) of (default) coefficients for the condensed system.

(7) If the default data and/or parameters are not the ones wanted by the user, calculation of the modified coefficients (from modified data supplied by the user).

(8) Choice of closure and shocks.

(9) Solution of the condensed system.

(10) Calculations of any back-solutions required (that is, calculation of the values of endogenous variables eliminated from the condensed system in step (4)).

(11) Printing of results.

(12) Is the large change procedure wanted? If so, do the following steps once or several times, as required:

(a) recalculation of new data base from solution;

(b) calculation of new coefficients for condensed system;

(c) solution of the new condensed system.

Readers familiar with the Johansen solution of any particular model will be able to identify each step in their favourite model. For example, for the ORANI model, see DPSV sections 10-23 for (1), (2), (3), sections 24-26 and 28-29 for (5), section 32 for (4), section 27 and Table 32.1 for (6), sections 34 and 43-45 for (8), (9), (10) and (11), and sections 8 and 47 for (12).

While presumably steps (1), (2) and (5) cannot be automated (it is certainly not my idea to suggest that they can), I believe it is possible to make largely automatic (in the sense that the computing becomes relatively straightforward and not unduly time-consuming) many, if not all, of the rest; and to do so in a model-independent, portable and user friendly manner. Of course, it will be possible to automate some steps more fully than others.

In addition, the following three features are desirable and need automating:

(13) (For model developers) Modification of an existing model. When a new model has substantial overlap with an existing model, it should be possible to build easily the new model by modifying the old one (rather than by starting from scratch). In particular, it should be easy to build a new model by adding or deleting equations and/or variables from an existing model.

(14) (For model users) Choice of closure and shocks. Ideally it should be possible to choose any variable (whether or not it is in the condensed system) to be exogenous or endogenous, and to shock any variable. This means it should be possible to make

(a) variables eliminated (at step (4)) exogenous or endogenous,

(b) variables absorbed (at step (4)) exogenous or endogenous.

(15) (For model users) Choice of user-specified parameters. Ideally, it should be possible for the user to vary these in any (economically sensible) way allowed by the theory. (See 3. of Figure 44.1 of DPSV for the user-specified parameters in ORANI.)

The next two sections discuss the automation of steps (3), (4) and (6)-(15). Section 4 is concerned with those already covered by GEMPACK, and section 5 with those still to be included in GEMPACK.

4. EXISTING FEATURES OF GEMPACK

4.1 GEMPACK 1 - For Model Users (Steps (8), (9), (11) above)

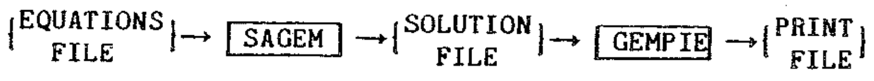

Figure 1: Summary of GEMPACK 1

The core of GEMPACK, which handles all of the basic tasks (i.e., (8), (9) and (11) above) that model users must do, is called GEMPACK 1, and is shown schematically in Figure 1. The starting point is the Equations file which holds all the linear equations for the condensed system of the particular model. More precisely, the condensed system of linear equations for the model can be written as

$$Cz = 0$$

where C is an n x m matrix (where n is the number of endogenous variables and m is the total number of variables in the condensed system) and z is an m x 1 vector of all the variables. The Equations file is essentially just a computer realization of the matrix C.

The program SAGEM (Solution Algorithm for General Equilibrium Models) is a major piece of software which allows the user to select the closure (i.e., choice of exogenous and endogenous variables), the exogenous variables to be shocked and their numerical shocks. Provided the closure is economically sensible, the output from SAGEM is a Solution file which contains the effect on selected endogenous variables of some or all of the shocks and/or the total (accumulated over all the shocks) effect on selected endogenous variables.

The Solution file produced by SAGEM is a binary file and so cannot be readily viewed or printed. The purpose of GEMPIE is to produce a Print file which can be displayed on the terminal or printed on a line printer for a hard copy of the results of the simulation done via SAGEM. GEMPIE allows users to control which parts of the Solution file are printed.

In order to discuss some of the features of SAGEM in more detail, we first need to describe the algorithm performed by SAGEM. As with any Johansen solution, once the exogenous/endogenous split has been chosen, the condensed system of equations $Cz = 0$ becomes

$$Az_1 = - Dz_2$$

where z_1 and z_2 are, respectively, the vectors of endogenous and exogenous variables in the condensed system, A is n x n and D is n x (m - n). The columns of A and D are those columns of C corresponding, respectively, to endogenous and exogenous variables. If k of the m - n exogenous variables are given nonzero shocks, the problem of finding the change in each of the n endogenous variables resulting from each of these k shocks reduces to solving the matrix equation

$$AX = B$$

where B is an n x k matrix, the jth column of which is obtained by multiplying the column of -D corresponding to the jth shocked variable by the shock given to this variable. The n x k matrix X is the "solution" of the simulation: its entry x_{ij} in row i and column j shows the percentage change in the ith endogenous variable as a result of the shock given to the jth shocked variable (assuming zero shock to all other variables). Because of the linearity of the method, the sum of all entries in any row i of X can be taken as the percentage change in the ith endogenous variable as a result of <u>all</u> of the k shocks given (assuming zero shock to all the other exogenous variables). The elasticity of the ith endogenous variable with respect to the jth shocked variable can be obtained by dividing x_{ij} by the shock given to the jth shocked variable.

The Solution file produced by SAGEM contains selected rows and columns of the solution matrix X, showing the effects on selected endogenous variables of selected shocks, and/or selected row totals across all columns of X, showing the total effect (accumulated over all the shocks) on selected endogenous variables.

A very important feature of SAGEM is that it allows users complete freedom over the exogenous/endogenous split in that any variables in the condensed system can be chosen as exogenous (or endogenous). Of course, variables which have been eliminated in the condensation process cannot be so handled (this is the point of (14)). However, because of the sparse matrix techniques used (see below), SAGEM can handle condensed systems having several thousand endogenous variables, and similar numbers of exogenous variables. For most current-generation models this provides great flexibility in that, if the elimination strategy (step (4)) is carefully chosen, most, if not all, of the variables of regular interest can be retained in the condensed system. As a consequence, most variables an economist wishes to shock can be shocked directly using SAGEM; and the effects of shocks on most of the endogenous variables of interest can be obtained directly using SAGEM. In particular, tasks (10) and (14) are now less important than previously, when the use of matrix inversion as the solution method limited to several hundred the number of endogenous variables in the system which could be directly solved. This erstwhile limitation often made difficulties for economists who wished to shock variables not retained in the (smaller) system solved and/or to find out the effects on endogenous variables eliminated from that system.

SAGEM can handle such a large number of variables because it takes advantage of the fact that the matrix A is sparse, meaning that only a small fraction of its entries are nonzero. (For ORANI, A has about 2600 rows and columns, but only about 2% of the over 6 million entries of A are nonzero.) SAGEM does not use the traditional

$$X = A^{-1} B$$

solution method since to store all of A^{-1} would limit n to a few hundred instead of a few thousand. The sparse matrix techniques used are outlined in Pearson and Rimmer (1983) and (1985). The important point, as far as the present account is concerned, is that use of these sparse matrix techniques enables SAGEM to work with a much larger condensed system than traditional methods have allowed. SAGEM has as its kernel the sparse matrix code MA28 developed by I.S. Duff at Harwell (Duff (1977)).

Sparse matrix methods have been used by others to solve economic models. Duchin and Szyld (1979), Szyld (1981) and Szyld (1983) report on their use in solving an input-output model of the world economy. Larry Cook has harnessed sparse methods for the Johansen solution of a model of Norway (Cook (1980)) and a simplified ORANI-type model (Cook (1982)). In all of these applications, as in ours, the sparse matrix code used is the Harwell routines MA28 (Duff (1977)).

Often different simulations use the same exogenous/ endogenous split but shock different sets of the exogenous variables. A valuable computer-resource-saving feature of SAGEM is that results from one such simulation can be saved and used to do the second one very quickly indeed. In the notation used above, the matrices A and D do not change. All that changes is the values for the vector z_2 and the matrix B. If the LU decomposition of A (produced by the Harwell sparse matrix code) has been saved from one simulation, the effect of different shocks can be readily calculated by recomputing B and solving the new system AX = B. For example, for the ORANI model on a VAX11/780 using the VMS operating system, the central processor unit (CPU) time taken to solve the 2635 equation condensed model initially is about 15 minutes (most of this time is taken calculating the LU decomposition of the matrix A), but subsequent shocks can be solved in about 5 CPU seconds per column (or variable shocked).

If no use is made of previous simulations, SAGEM is normally run non-interactively because it usually requires a very large amount of user input and takes many minutes of CPU time (for large models such as ORANI). However, if a previous simulation is used in the manner described in the previous paragraph, SAGEM may require only a few items of user input and then it can be run interactively with almost immediate turn around. GEMPIE can always be run interactively and takes only a few seconds CPU time.

4.2 GEMPACK 2 - Basic Package for Model Developers (Steps (6), (13), (10), (14) above)

This part of GEMPACK assists developers greatly with tasks (6) and (13) above. As will be explained later in this section, (10) and (14) may be viewed, from a computational viewpoint, as special cases of (13); thus GEMPACK 2 also assists model users with tasks (10) and (14). The major existing overhead in solving a model by Johansen's method is step (6), the calculation of the coefficient matrix C for the condensed system Cz = 0 of linear equations. Typically this requires calculation of a large number of submatrices corresponding to different equations of the model and their assembly in appropriate row and column positions in C. (For example, for the ORANI condensed system of about 2600 equations in about 6000 unknowns, each of the about 100 separate submatrices, namely A_1, A_2, ..., Z_2 in Table 32.1 of DPSV, must be calculated.) At this stage of the development of GEMPACK, model developers must provide model-specific code to generate the submatrices for the particular model and to put them in files called Submatrix Data files. (Automating the production of these Submatrix Data files is the next major extension of GEMPACK planned - see section 5.1.)

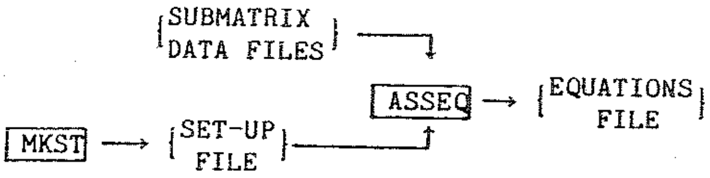

Figure 2: Assembling the Equations file for a new model

To form the Equations file for a new model, proceed as follows (see Figure 2). First produce (using user-provided, model-specific code) one or more Submatrix Data files which between them contain all the submatrices required in the condensed system. (For example, for condensed ORANI as in Table 32.1 of DPSV, we use 7 Submatrix Data files; each one of these is procduced by a separate piece of code which is specific to ORANI.) Then, using the program MKST (Make a SeT-up file) in GEMPACK 2, create a Set-up file for the model. This Set-up file is merely a computer realization of the tableau showing in which rows and columns of the coefficient matrix C the various submatrices are to be placed, and in which of the Submatrix Data files each submatrix is to be found. Finally, run the program ASSEQ which ASSembles the EQuations file by picking out from each of the Submatrix Data files those submatrices wanted (ASSEQ obtains this information from the Set-up file) and forms the coefficient matrix C by putting them into the appropriate rows and columns. This process is summarized in Figure 2.

The Set-up file is a vital ingredient in the way this process works. For example, the Submatrix Data files searched by ASSEQ can contain submatrices not needed for the particular model whose Equations file is being formed. A submatrix will be inserted in the matrix C only if the Set-up file indicates it is to be included. Another possibility is that two of the Submatrix Data files may contain different versions of the same submatrix; which one is included in C is dictated by the Set-up file. We explain below how these features make it possible to modify an existing model (step (13)).

Suppose, for example, that we wish to modify ORANI by replacing its investment theory, basically equation (32.11) in Table 32.1 of DPSV which says

$$y = C_1 \kappa(0) + C_2 r(0) + C_3 \omega + C_4 i_R + C_5 f^{(2)},$$

by some other theory that involves a new vector variable d_4, but which does not involve the scalar variable ω or the vector variable $f^{(2)}$. The new equations can be written symbolically as

$$y = C_6 \kappa(0) + C_7 r(0) + C_8 i_R + C_9 d_4. \qquad (32.11')$$

To implement this new model using GEMPACK 2, proceed as follows. First, write a program (regrettably, one-off code) to create a new Submatrix Data file containing just the new submatrices C_6, C_7, C_8, C_9. Then, using a suitable Set-up file, assemble the new Equations file via ASSEQ by picking out all the same submatrices as before from the old Submatrix Data files except for the superseded C_1, ..., C_5 relevant to the old investment equation (32.11), including instead the new C_6, ..., C_9 from the new Submatrix Data file. The new Set-up file required can be created easily from the existing Set-up file for condensed ORANI by using the program MODST (MODify a SeT-up file) in GEMPACK 2. This program accepts simple instructions to add or delete specific variables, equations, Submatrix Data files or submatrices. In the example above, the modifications required are:

(i) delete the old equation (32.11) - this automatically deletes the associate submatrices $C_1,\ldots,C_5$;

(ii) add a new variable d_4;

(iii) add a new equation block (32.11');

(iv) add a new Submatrix Data file;

(v) add new submatrices $C_6,\ldots,C_9$ to be found in the new Submatrix Data file.

This procedure for changing an existing model, involving the modification of an old Set-up file by MODST and the creation of one or more new Submatrix Data files, is summarized in Figure 3.

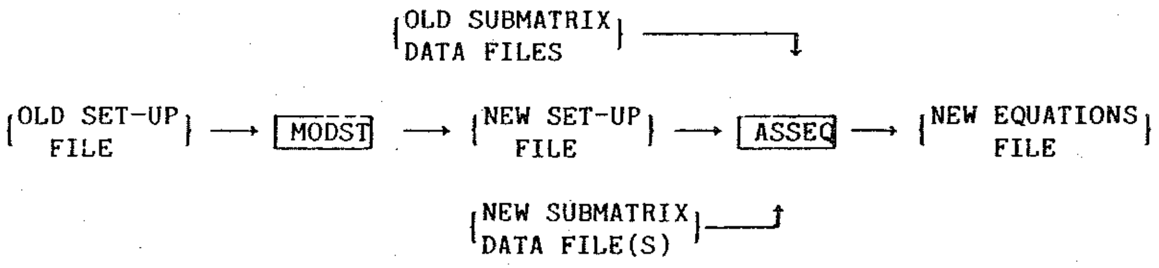

Figure 3: Modifying an existing model

The ability of GEMPACK 2 to handle task (13) also gives developers and sophisticated users tools for handling tasks (10) - calculation of the values of endogenous variables eliminated from the condensed system - and (14) - making eliminated and/or absorbed variables exogenous or endogenous. In fact, (10) and (14) are (computationally) just special cases of (13). If the values of some endogenous variables not in the condensed system are wanted (task (10)), simply enlarge the condensed system by adding the equations that express these endogenous variables in terms of variables in the existing condensed system, and solve the enlarged system. Task (14) can be handled similarly by adding to the existing condensed system, equations that express the connection between the desired variables and those in the existing condensed system, and then using the flexibility given by SAGEM for the choice of exogenous/endogenous split to solve the larger system. Thus (10) and (14) can also be handled by GEMPACK 2, provided the additional (one-off) code to generate the appropriate Submatrix Data files is written and provided the enlarged condensed system to be solved by SAGEM is not too large to be fitted into the computer memory available.

As an example, consider the vector variable x(g+1,1), denoting employment by industry, which has been eliminated from the condensed system for ORANI (Table 32.1 of DPSV) by using equation (12.64) on page 130 of DPSV. This variable is usually endogenous in ORANI simulations and frequently its values are required as a back-solution (task (10)). Occasionally an economist may wish to make it exogenous and to shock it (task (14)(a)). Either of these can be achieved by adding, to condensed ORANI, the appropriate equations which express x(g+1,1) in terms of variables in the usual condensed system. In this case, these equations can be written symbolically as

$$x(g+1,1) = A_{10}\ p(g+1,1) + A_{11}\ p(g+1,2) + A_{12}\ p(g+1,2) + A_{13}\ z + b_{14}$$

(where b_{14} is a new vector variable, similar to b_1, b_2, ..., b_{13} in Table 32.1 of DPSV, which absorbs the relevant technical change terms). The required new Equations file (for this enlarged model) can be assembled, as in Figure 3, by using MODST to produce the required new Set-up file and writing a new program to produce a new Submatrix Data file containing the submatrices A_{10},..., A_{13} in the new equation above.

5. FURTHER DEVELOPMENTS

5.1 Calculation of Submatrix Data Files (Steps (6) and (4))

While GEMPACK 2 assists developers a little with task (6), there is no doubt that this task is the major obstacle remaining in the quest of facilitating the speedy and easy computer implementation of a large economic model. Plans for rectifying this are given below.

Calculation of an individual submatrix to go in a Submatrix Data file usually involves simple algebraic manipulations (sums, products, shares) of items in the data base. At present this is done via specially written code - one piece of code for each different submatrix. Automating this procedure can be achieved by having a very flexible program, perhaps to be called WRSMD, standing for WRite SubMatrix Data files, which is driven by user input telling it which matrices from the data base are to be used, via which standard matrix-like operations they are to be combined, how the resulting submatrix is to be written on the Submatrix Data file and whereabouts in the whole condensed system matrix C it fits. This program could be run interactively or in batch mode.

The submatrices for the original system of linear equations are much more simply obtained from the data base than are those for the condensed system. The more elimination and absorption is done, the more messy are the calculations and algebra required. For this reason it is much easier to describe and form the submatrices for the original system than for a condensed system. Also the original system is easier for economists to check.

An alternative to producing the Submatrix Data files for the condensed system directly is the following, which also takes care of step (4), the condensation.

(i) Store on Submatrix Data files all submatrices for the original system of linear equations. (Use the yet-to-be-written program WRSMD.)

(ii) Automate the symbolic transition from the original system of equations, held in the computer in symbolic form, to the condensed system, by doing formal substitutions. Suppose, for example, that the original system can be written symbolically as

$$\begin{aligned} v &= A_1 \, p + A_2 \, t + w, \\ t &= B_1 \, z + B_2 \, p, \\ C_1 \, \phi &+ C_2 \, w + C_3 \, z = 0, \\ w &= D_1 \, t + \phi, \end{aligned}$$

where v, p, t, w, z and ϕ are (vector) variable names and A_1, A_2, B_1, B_2, B_3, C_1, C_2, C_3, D_1 are the names of the submatrices. We might proceed to the condensed system by using the second equation to substitute out variable t and the last equation to eliminate w. The resulting condensed system could be written symbolically as

$$\begin{aligned} v &= (A_1 + D_1 B_2) \, p + A_2 \, t + D_1 B_1 \, z + \phi \, , \\ (C_1 + C_2) \, \phi &+ (C_2 D_1 B_1 + C_3) \, z + C_2 D_1 B_2 \, p = 0. \end{aligned}$$

(iii) Automate the numerical calculation of the submatrices for the condensed system from the results of (i) and (ii) above. As can be seen from the example in (ii) above, where the submatrices of the condensed system are

$$A_1 + D_1B_2,\ A_2,\ D_1B_1,\ C_1 + C_2,\ C_2D_1B_1 + C_3,\ C_2D_1B_2\ ,$$

this means automating procedures for adding and multiplying submatrices held in the Submatrix Data files for the original system, and writing the resulting submatrices to the Submatrix Data files for the condensed system.

A preliminary version of code to do (ii) has been written. While step (iii) is easy in principle, it is somewhat intricate in detail because the submatrices are stored on Submatrix Data files in several different standard forms (for example, diagonal matrices, full matrices, sparse matrices, identity matrices) in order to economise on disk space. There is also a real possibility that the large amount of number-crunching and file accessing associated with (iii) will involve unacceptably high use of computer resources (such as CPU time, disk and/or memory space).

The route (i), (ii), (iii) above is very desirable. The instructions for forming the submatrices from the data base are simpler for the original system than for the condensed, and steps (ii) and (iii) would bypass all the very messy algebra and code associated with the complicated process of elimination and absorption of variables. It would also enable different simulations to use different condensations (depending on which variables are of major interest), and this would be another way of handling (14) and (10). Finally, the burden of checking the calculation of coefficients would be placed with the original system, where it is much easier than with the condensed system.

If however, (iii) uses unacceptably many computing resources, calculation of the Submatrix Data files for the condensed system (step (6)) will have to be done directly, from the data base, using more complicated user inputs for the program WRSMD than for (i).

5.2 Varying Data and/or User-Specified Parameters (Steps (7) and (15))

Users may wish to vary part of the data base or some of the parameters (e.g., elasticities) for the model (step (7)) or, if the model allows for this, to vary some of the user-specified parameters (task (15)). These can both be regarded as being automated once step (6) is automated. For then, only the affected submatrices need to be recalculated (using the new data or user-specified parameters), and the whole new Equations file assembled using ASSEQ.

5.3 The Large Change (Step (12))

The large change procedure is a method used to reduce errors introduced by the Johansen solution method in which non-linear equations are approximated by linear equations in the percentage changes of the original variables. Roughly speaking, a large shock (say 50%) is broken up into several smaller shocks (say 5 each of 10%) which are applied one at a time. After each smaller shock the resulting changes to all variables are taken into account in updating the data base before the next small shock is applied. Thus the large change is really an Euler style computation (see DPSV section 31.4).

A partial automation could be achieved by writing model-independent code that links model-specific procedures for

(i) calculating coefficients from the data base, and

(ii) recalculating the data base from a new solution

to the model-independent solving program SAGEM, and keeps cycling through these steps until the desired large change has been made; writing the model-independent linking code should not be difficult. To fully automate the large change would mean also automating (i) (which is really just step (6)) and (ii). The automation of (ii) could be achieved in an analogous way to that of (6) - namely providing a piece of software that accepts commands from a user which make appropriate matrix-like operations on parts of a Solution file and writes the results to the relevant parts of the updated data base. This will, of course, present considerable bookkeeping problems in models where values of endogenous variables not in the condensed system are required in the process of updating the data base.

5.4 Forming the Linear Equations (Step (3))

Finally, there is the possibility of automating the transition from the non-linear equations in the levels to the linear ones in percentage changes. This would involve dealing with equations symbolically (like in section 5.1). Where the non-linear equations involve elementary functions like addition, multiplication and powers, and when non-elementary functions like CRESH, CRETH are involved, this should be somewhat routine. But when, as is often the case (see for example, DPSV sections 5.1.1 or 12.1), the non-linear functions are defined only indirectly as solutions to optimization problems, automating the formation of the linearized equations will be less straightforward.

6. MACHINE REQUIREMENTS FOR, AND AVAILABILITY OF, GEMPACK

The only requirement for running GEMPACK is that a suitable version of the programming language FORTRAN 77 be available on the machine. More precisely, the version of FORTRAN 77 available must be one which includes all the language constructs of the full language (not just the subset) as laid down in the ANSI standard (ANSI (1978)). In practice this means that GEMPACK can be run on most, if not all, mini and main-frame computers, but only on some microcomputers.

The size of computer memory required for GEMPACK depends on the size of the economic models for which it is used. The programs in GEMPACK can all be modified readily to change the memory requirements. For this purpose, the basic quantity which gives a good guide as to whether or not a particular model fits into a given memory space is the number NZ of nonzero entries in the coefficient matrix C when the condensed linear equations of the model are written as Cz=0. Very roughly, to solve the model using GEMPACK it is necessary to be able to access about 4NZ integers or reals in one program. Thus the size requirements vary greatly from model to model. For example for condensed ORANI, NZ is about 130,000 while for the miniature model skeletal (see DPSV sections 3-7) NZ is about 170. It follows that the memory available to you on your machine will limit the size of models you can solve on it.

Because essentially all GEMPACK code has been written strictly within ANSI Standard FORTRAN 77 (the full language), it is portable from the VAX, where it was developed, to other machines relatively easily. However, a small number of subroutines (for GEMPACK 1 there are just 3, all dealing with file opening and closing) are, of necessity, somewhat machine-specific. Also, it may be necessary to change some of the parameters in some of the main programs for different machines. But making the changes required is usually a straightforward matter. Versions of GEMPACK are currently running on VAX (under VMS), PRIME (under Primos), Pyramid (under UNIX) and NAS (running IBM VS FORTRAN under VM) machines.

The GEMPACK software is being developed as a portable, model-independent and user friendly package precisely with the aim of making available a working tool to a large number of economists. As the development outlined above proceeds, more parts will be available for release to other sites. GEMPACK 1 is available now and GEMPACK 2 will be available soon. GEMPACK 1 gives users the ability to solve models already set up (principally standard ORANI and its skeletal version MO) while GEMPACK 2 will include all software and documentation required for users who wish to set up their own models for solution.

Enquiries about obtaining GEMPACK for your computer should be addressed to the author, IMPACT Project, the University of Melbourne, Parkville 3053, Australia. Please state the type of machine and type (and version) of operating system. Note also that, because the Harwell sparse matrix code (Duff (1977)) is called by SAGEM, a licence to use this code must be obtained from Harwell.

The author will welcome enquiries from, and suggestions by, economists wishing to implement other models for solution via GEMPACK, and on other aspects of GEMPACK. This kind of feedback is essential if GEMPACK is to be of maximum possible use to working economists.

ACKNOWLEDGMENTS

A number of people have made significant contributions to the evolution of GEMPACK. Without Russell Rimmer to span the apparent gulf between practising mathematician and economists, the work would not have started. I am especially grateful to Alan Powell, Brian Parmenter and Russell Rimmer for encouragement, advice and feedback, and to Rom Jagielski who has written some of the code. I am also grateful to John Madden whose request to move the software to the University of Tasmania's PRIME computer was the catalyst behind my belated realization that portability was a vital feature for GEMPACK.

The work reported in this paper has been assisted by support from the Australian Research Grants Scheme.

REFERENCES

ANSI (1978) American National Standard Programming Language FORTRAN X3.9-1978. New York: American National Standards Institute.

Cook, L.H. (1980) Validation of a Johansen Type Multisectoral Model: Norway, 1949-1961. Ph.D. Thesis, Department of Economics, Monash University.

Cook, L.H. (1982) A Simplified ORANI-Type Model. Centre of Policy Studies, Monash University (pre-print).

Dixon, Peter B., Parmenter, B.R., Sutton, John and Vincent, D.P. (1982) ORANI: A Multisectoral Model of the Australian Economy. Amsterdam: North-Holland, pp. 372.

Duchin, Faye and Szyld, Daniel B. (1979) Application of Sparse Matrix Techniques to Inter-Regional Input-Output Analysis, Economics of Planning 15: 142-167.

Duff, I.S. (1977) MA28 - a Set of FORTRAN Subroutines for Sparse Unsymmetric Systems of Linear Equations. Report R. 8730, A.E.R.E. Harwell, HMSO, London.

Mathiesen, L. (1985) Computation of Economic Equilibria by a Sequence of Linear Complementarity Problems, Mathematical Programming Study, 23.

Parmenter, B.R., Pearson, K.R. and Jagielski, R. (1984) Bottoms-up, Tops-down and Hybrid Methods for Regionalising Computable General Equilibrium Models: Some Comparisons. In Papers of the 9th Australian and New Zealand Regional Science Conference (R.E. Glass, Ed.), p. 144-187.

Pearson, K.R. and Rimmer, Russell J. (1983) Sparse Matrix Methods for Computable General Equilibrium Models of the Johansen Class. IMPACT Preliminary Working Paper No. OP-43, Melbourne, pp. 40.

Pearson, K.R. and Rimmer, Russell J. (1985) An Efficient Method for the Solution of Large Computable General Equilibrium Models, J. Int. Assoc. Math. Comp. Sim. 27:223-229.

Rutherford, Thomas Fox (1985a) MPS/GE User's Guide. Department of Operations Research, Stanford University, pp. 72.

Rutherford, Thomas Fox (1985b) Operating Instructions for CASGEN. Department of Operations Research, Stanford University, pp. 9.

Siriwardana, A.M. (1985) A Computable General Equilibrium Model of Tariff Protection in the Colony of Victoria Using Input-Output Data for 1880. Ph.D. Thesis, School of Economics, La Trobe University.

Syzld, D.B. (1981) Using Sparse Matrix Techniques to Solve a Model of the World Economy. in Sparse Matrices and their Uses (I.S. Duff, Ed.), Academic Press, p.357-365.

Syzld, Daniel B. (1983) Preliminary Results in Implementing a Model of the World Economy on the CYBER 205: A Case of Large Sparse Nonsymmetric Linear Equations. In CYBER 200 Applications Seminar: Proceedings of Seminar Held in Lanham, Maryland, October 1983 (J. Patrick Gary, Compiler), NASA.