MONASH
COPSIP
PWP-IP
66

CENTRE of

POLICY

STUDIES and

the IMPACT

PROJECT

Eleventh Floor
Menzies Building
Monash University Wellington Road
CLAYTON Vic 3168 AUSTRALIA

Telephone:
(03) 9905 2398, (03) 9905 5112

Fax numbers:
(03) 9905 2426, (03)9905 5486

e-mail

from overseas:
61 3 9905 2398
or
61 3 9905 5112
from overseas:
61 3 9905 2426 or 61 3 9905 5486

impact@vaxc.cc.monash.edu.au

MULTIREGIONAL AND INTERTEMPORAL AGE MODELLING VIA GEMPACK

by

W. Jill HARRISON
Monash University

K.R. PEARSON
Monash University and La Trobe University

and

Alan A. Powell
Monash University

Preliminary Working Paper No. IP-66 September 1994

ABSTRACT

The past few years have seen increasing use of multiregional and intertemporal (that is, forward-looking) applied general equilibrium (AGE) models. These models share the characteristic of rapidly expanding dimensions. Because multilateral trade models increase in size with the square of the number of regions and intertemporal models increase in proportion to the number of time periods, they can be very difficult to solve purely because of their size.

This paper considers three recent models of these kinds, namely, GTAP, the Global Trade Analysis Project's multiregional model of the world; MRES, a multiregional forecasting model of Australia; and ORANI-INT, an intertemporal model of the Australian economy. Each of these has been implemented using the GEMPACK suite of general-purpose economic modelling software. In this paper we discuss features of GEMPACK which are especially relevant for multiregional and/or intertemporal models. These include an automatic facility to condense models to a manageable size; separation of theory, data, and closure/shocks; automatic creation of updated (that is, post-simulation) data files; and a solver that is well suited to intertemporal models with substantial elements of forward-looking behaviour. Each feature is illustrated via references to one or more of the three models mentioned above.

J.E.L. classification nos: C68, C53, C88.

CONTENTS

LIST OF TABLES

FIGURE

MULTIREGIONAL AND INTERTEMPORAL AGE MODELLING

VIA GEMPACK*

by

W. Jill HARRISON   K. R. PEARSON   Alan A. POWELL
Monash University   *Monash University and La Trobe University*   *Monash University*

1. Introduction

The past few years have seen an increasing interest in the use of multiregional and also intertemporal (that is, forward-looking) applied general equilibrium (AGE) models. For example, multiregional models have been built in order to investigate trade policies (such as the GATT) and also to examine the regional effects within a single country of changes in government policy. Intertemporal models are essential for obtaining insights into issues, such as environmental ones, in which the timing of policy changes, and agents' reactions to them, are crucial.

Multiregional and intertemporal models share the characteristic of rapidly expanding dimensions. Multilateral trade models increase in size with the square of the number of regions while intertemporal models increase in proportion to the number of time periods. Even with only modest sectoral (and regional) disaggregation (and a modest number of time points in the intertemporal case), these models can be very difficult to solve purely because of their size.

This paper considers three recent models of these kinds. These are:

- GTAP, the Global Trade Analysis Project's multiregional model of the world, designed as a tool for use by the wide-ranging cross-section of policy advisers involved with trade policy (see Hertel and Tsigas (1994));

- MRES, a multiregional model of Australia, designed to forecast the prospects of industries at the national level, at the state level, and at the level of 56 statistical divisions (see Adams, Horridge and others (1994) and CoPS/Syntec (1994)); and

- ORANI-INT, an intertemporal model of the Australian economy, designed to show time paths of adjustment of its 13 sectors to a wide range of different shocks (see Malakellis 1993, 1994).

Each of the above models has been implemented using GEMPACK, which is a suite of general-purpose economic modelling software designed for general equilibrium and partial equilibrium models. In this paper we discuss features of GEMPACK which are especially relevant for multiregional and/or intertemporal (or other large) models. These features include:

condensation	—	the ability to condense models to a manage-able size, which may be essential in order to solve them, even on a mainframe;
separation	—	a modular arrangement whereby the three principal ingredients of a model's specification (theory, data, and closure/shocks) are clearly identified in separate locales in the software;
automatic updating	—	automatic creation of updated (that is, post-simulation) data files as part of a simulation;
general-purpose solution algorithm	—	a solver that is well suited to intertemporal models irrespective of the amount of forward-looking behaviour incorporated within them.

Each feature is illustrated via references to one or more of the three models listed above.

We discuss *condensation* in general in section 2 and then in subsections 2.1 to 2.3 give examples of how this works with the three models. In section 3 we consider *separation* of theory from data and from the choice of closure and shocks, especially in relation to GTAP. We discuss *automatic data updates* in section 4, concentrating on the use of this facility to produce forecasts with MRES. Finally, in section 5 we discuss GEMPACK's *solution algorithm*, concentrating on its use for solving intertemporal models; in this case our illustration is drawn from ORANI-INT.

Before discussing condensation, we give in section 1.1 below a brief introduction to the procedure by which models are implemented and solved in GEMPACK.

1.1 Implementing and solving models with GEMPACK

An overview of the current version of GEMPACK (Release 5.1, April 1994) can be found in Harrison and Pearson (1994c). The GEMPACK user documentation is in Harrison and Pearson (1994a,b and 1993). Below we give a brief (but hopefully, self-contained) summary of the process of implementing and solving a model via GEMPACK.[1] Several of the points recur below in our illustrative examples.

A model is ready for implementation in GEMPACK when

(i)	***theory***	the equations describing its economic behaviour are written down in an algebraic form in a text file, and;
(ii)	***data***	data describing one solution of the model are assembled, to be used as a starting point for simulations.

The text file mentioned in (i) above contains the equations (written in an algebra-like syntax) and information about the data. This file is called a ***TABLO Input file*** since TABLO is the name of the GEMPACK program which processes this file and converts the information on it to a form suitable for running simulations with the model. The equations in the TABLO Input file can be levels equations, linearized equations or a mixture of the two. Examples of TABLO Input files can be found in Harrison *et al.* (1993), Harrison and Pearson (1994a,c), Harrison and Small (1993), Hertel and Tsigas (1994) and Horridge *et al.* (1993).

---

1 More details can be found in sections 2-5 of Harrison and Pearson (1994c).

Once the TABLO Input file and data files have been prepared, the model is ready for simulations. This involves

(iii) closure and shocks	preparing the closure (that is, choosing which variables are to be exogenous and which endogenous) and setting the shocks.

The details of closure and shocks, plus other information required to specify a simulation (such as the names of various output files), are usually put onto a text file known in GEMPACK nomenclature as a *Command file*,[2] as described in section 3.2 of Harrison and Pearson (1994c).

The steps involved in carrying out a simulation (that is, in solving the model) are listed below and are shown in Figure 1.1, which indicates where the theory, data, closure and shocks enter.

Step 1. Computer Implementation of the Model

Process the TABLO Input file for the model by running the GEMPACK program TABLO. TABLO first checks the syntax and semantics of the TABLO Input file (and reports any errors). If there are no errors, condensation (see section 2 below) can be carried out if desired. Then TABLO converts the information on the TABLO Input file to a form suitable for simulations. At the user's option, this is either a special-purpose Fortran program (the *TABLO-generated program* of the model) or a pair of *GEMSIM Auxiliary files*. (For most medium-sized or large models, the TABLO-generated route is chosen since it results in quicker simulations.) If the TABLO-generated program route is chosen, after TABLO has finished, the TABLO-generated program is compiled using an appropriate Fortran compiler, and linked to the GEMPACK library of subroutines; this produces an executable version of the TABLO-generated program.

Step 2. Simulation

Run the TABLO-generated program or the GEMPACK program GEMSIM. Specify which base data are to be read and describe the closure and the shocks. The program then computes the solution to the simulation and writes the results to a *Solution file*. It also produces updated (that is, post-simulation) data.

Step 3. Printing the Results of the Simulation

Run the GEMPACK program GEMPIE to convert the binary Solution file produced in Step 2 into a *Print file*. This is a text file which can be printed (or edited).

Often many different simulations are carried out on the same model, for example, with different closures and/or shocks, or starting from different base data. In these cases, Steps 2 and 3 are repeated but not Step 1. However Step 1 must be repeated if the TABLO Input file for the model is changed in any way, or if a different condensation is used.

---

2   File types which have technical definitions in GEMPACK nomenclature are shown in bold italics when first mentioned in this paper.
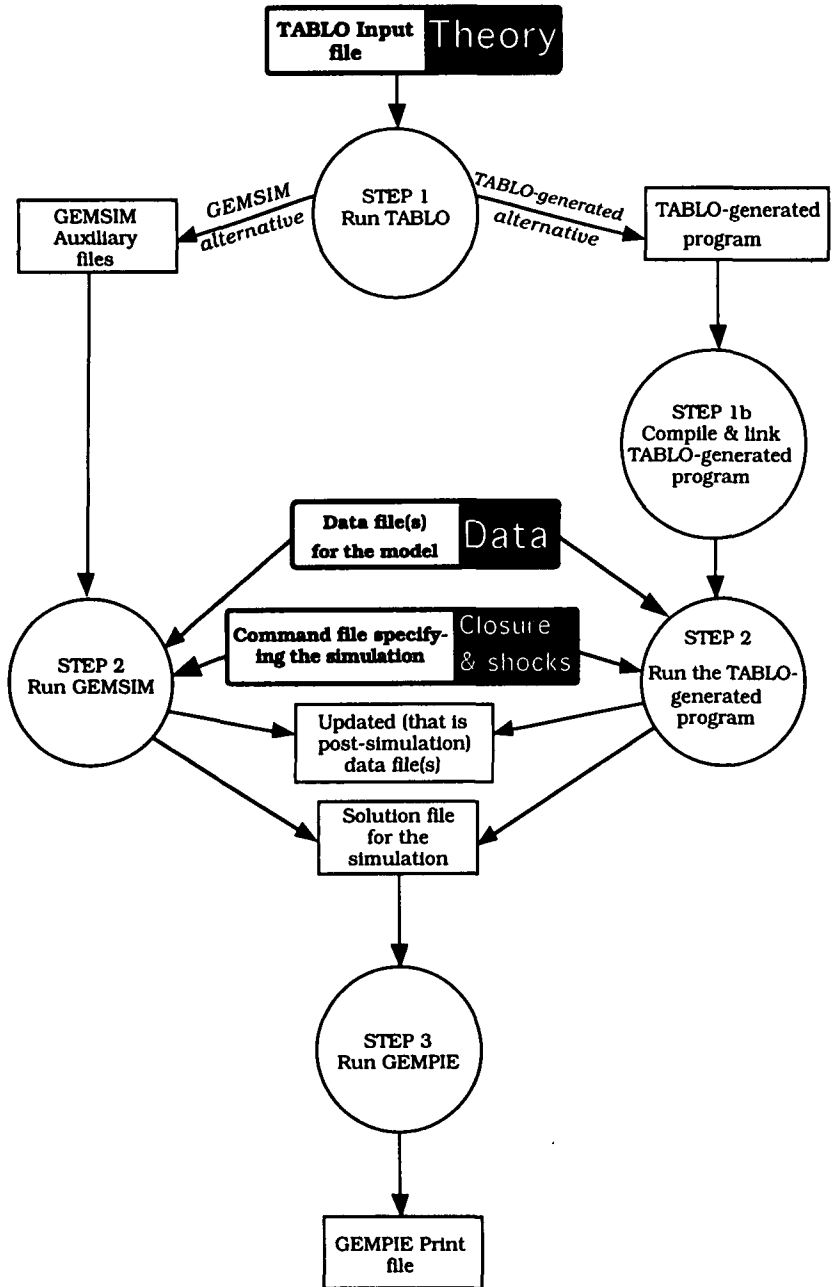
Figure 1.1: Steps in carrying out a simulation

2. Condensation

The basic idea behind condensation is to reduce the size of the model to satisfy computer memory constraints. In many cases, condensation also reduces the CPU time needed for each simulation.

Condensation refers to the process, familiar to all high school algebra students, of using one equation to find an expression for one variable in terms of other variables in the equation. This expression is then used to replace that one variable in all other equations. Instead of the modeller carrying out this laborious process with pen and paper, the GEMPACK program TABLO does these algebraic operations symbolically on the set of linearized equations derived from the TABLO Input file.

Condensation is easy to carry out in practice. There is a prompt-driven dialogue in which TABLO asks for the name of the variable to be substituted out and then the name of the equation to use to substitute it out. TABLO does the algebra quickly and accurately, and then asks if you want to do another substitution.

For example, consider the equation DMNDDPRICE in the TABLO Input file GTAP94.TAB for GTAP. This is

(all,i,TRAD_COMM) (all,j,PROD_COMM)(all,i,REG)

$pfd(i,j,r) = tfd(i,j,r) + pm(i,r)$;

Note that the one symbolic EQUATION DMNDDPRICE is actually several equations because of the indices i, j, r in the equation. In the 10 × 10 aggregation of GTAP, the sets TRAD_COMM, PROD_COMM and REG are of sizes 10, 11 and 10 respectively, which means that there are 1,100 individual equations corresponding to the single symbolic DMNDDPRICE equation.

One of the condensation actions in the standard condensation of GTAP94 is to use this equation to substitute out variable pfd. When this is done, for example, the equation INDDOM

(all,i,TRAD_COMM)(all,j,PROD_COMM)(all,r,REG)

$qfd(i,j,s) = qf(i,j,s) - ESUBD(i)*[\mathbf{pfd(i,j,s)} - pf(i,j,s)]$

is rewritten by TABLO as

$qfd(i,j,s) = qf(i,j,s) - ESUBD(i)*[\mathbf{tfd(i,j,s) + pm(i,s)} - pf(i,j,s)]$.

Variable pfd is replaced similarly in all other equations. Then we can ignore the equation DMNDDPRICE and work with the *condensed* system consisting of the remaining equations as rewritten, thus reducing the size of the condensed system by 1,100 equations and 1,100 variables. This process can be repeated for other substitutions.

Note that when the condensed system is solved, values are obtained for all endogenous variables in it. In the example above, once values for variables tfd and pm have been determined, we can put them in the right-hand side of equation DMNDDPRICE and hence find the solution for variable pfd; this is called ***backsolving*** for pfd. When doing condensation, a modeller can choose to backsolve

for the variable (which means that its values are available in the model's solution) or merely **substitute** it out (which means that its values are not available in the solution).

The other way of condensing a model is to **omit** certain variables. Omitting a variable from a linearized equation (that is, leaving out all terms involving the variable) is the same as exogenously fixing the level of that variable at its base-case solution value and hence setting the change (or percentage change[3]) in that variable to zero. For example, in ·GTAP, the variable tpm(i,r), which measures the percentage change in the tax on imports of composite commodity i purchased by private households in region r, is omitted in the standard condensation. The only equation this occurs in is the equation PHHIPRICES, which is:

$$(all,i,TRAD\_COMM)(all,r,REG) \quad ppm(i,r) = tpm(i,r) + pim(i,r) .$$

This equation links the percentage change in the price of imports of i by private households in region r [which is ppm(i,r)] to the percentage change in the market price of imported i in region r [namely, pim(i,r)]. When tpm(i,r) is omitted, the tax paid by households in r on imported i remains fixed at its pre-simulation level, and the equation PHHIPRICES becomes:

$$(all,i,TRAD\_COMM)(all,r,REG) \quad ppm(i,r) = pim(i,r) .$$

If a variable is to be set exogenous but not shocked in a group of simulations, it can be omitted from the condensed system for these simulations. Omitting a variable reduces the number of variables in the condensed system, but does not change the number of equations. For example, omitting the vector variable tpm from the condensed system for the 10 × 10 aggregation of GTAP reduces the number of variables in the condensed system by 100. In the standard condensation of GTAP, seven vector variables (all of them taxes) are omitted.

Condensation is optional and is quite separate from the process of specifying the model's equations via the TABLO Input file. This means that the variables to be condensed out can be chosen after setting up the model; moreover, different condensations can be used with the same model (just by listing different variables to condense out). This feature is advantageous since often it is found that different condensations suit different groups of simulations.

General points to note are:

(1)   (a) Variables backsolved for or substituted out are effectively hard-wired as endogenous. If variables backsolved for or substituted out in the standard condensation of the model need to be set exogenous in a group of simulations, it is possible to implement a different condensation in which these variables are not substituted out (but perhaps others are).

   (b) Variables omitted from the condensed system are effectively hard-wired as exogenous and given no shock. If variables omitted in the standard condensation need to be shocked (or set endogenous) in a group of simulations, it is possible to carry out a different condensation in which these variables are not omitted (but perhaps others are).

---

3 In this paper, *percentage change* means the percentage deviation in a variable from its value in the base-case or control solution.

(2) The system of linearized equations in the condensed system and the system of linearized equations of the entire model both can be written in the form

$$Cz = 0 \ . \tag{1}$$

Matrices like C are referred to below as an ***Equations matrix***, either for the uncondensed, or for the condensed system. In the latter case, C is the $m \times p$ matrix of coefficients of the m equations of the condensed system, while z is the $p \times 1$ vector of *all* the variables appearing in the condensed system. In general, m is less than p, so $(p - m)$ of the variables must be declared exogenous; shocks will be applied to some of these. The remaining m variables are endogenous.

Once the exogenous/endogenous split has been chosen, the system of equations $Cz = 0$, as in (1) above, becomes

$$A z_1 = -D z_2 \tag{2}$$

where z_1 and z_2 are respectively the (column) vectors of endogenous and exogenous variables. A is $m \times m$ and D is $m \times (p - m)$. The columns of the matrices A and D are just the columns of C corresponding to the endogenous and exogenous variables respectively. The matrix A here is called the ***left-hand-side matrix*** of the condensed system; similar nomenclature applies to the corresponding matrix for the entire model. The shocks are the values to use for z_2. Once these are known, we have a system

$$A z_1 = b \tag{3}$$

to solve (where b is an $m \times 1$ vector). The Johansen solution of the simulation is the solution z_1 of equation (3).

(3) GEMPACK produces accurate solutions of the underlying (usually non-linear) levels equations by carrying out multi-step calculations. At each step, just part of the shock is applied, a system of linear equations as in (3) above is solved, and the data are updated. The number of steps needed for an accurate solution depends on the model, the closure, and most importantly, on the size of the shocks. It is usual to carry out three different multi-step calculations (for example, 2-step, 4-step and 6-step) and then extrapolate on the basis of these to obtain the final solution. In subsections 2.1–2.3 we report CPU times for an accurate solution and for a Johansen solution, and indicate what multi-step calculations were carried out to obtain the accurate solution. The time taken for an accurate nonlinear solution is usually somewhere between 4 and 10 times as long as that for a Johansen solution, but could be longer in the case of some models and/or shocks. More details can be found in section 4 of Harrison and Pearson (1994c) and in section 5.2 of Harrison and Pearson (1994b).

(4) GEMPACK solves the system (3) of linear equations using the sparse linear-equations routines MA28 (see Duff (1977)) developed by AEA Technology at Harwell, UK. These solve the equations by calculating the so-called LU decomposition of A; they exploit the sparsity of A. This LU

decomposition often accounts for a significant part of the total CPU time
for each step of a multi-step calculation. The number of nonzeros in A is
one of the significant factors affecting the CPU time for LU decomposition.
In subsections 2.1–2.3 we report the CPU time for the LU decomposition
for each of the three models.

CPU times reported

The CPU times reported in this paper are those obtained on an 80486 DOS PC
with clockspeed 66MHz, 2Mb SMARTDRV, and VESA bus. The programs were
written by Release 5.1-001 (July 1994) of GEMPACK and compiled and linked using
Lahey Fortran F77L-EM/32; compiler options used were ones that in our
experience produce fastest run times.

2.1 GTAP — the Global Trade Analysis Project's model[4]

The Global Trade Analysis Project (GTAP) was established with the objective of
lowering the cost of entry for those seeking to conduct quantitative analyses of
international economic issues in an economy-wide context. It consists of several
components:

- a global data base,
- a standard modelling framework,
- software for manipulating the data and implementing the standard model,
  and
- a global network of researchers with a common interest in multiregional
  trade analysis and related issues.

The current GTAP data base recognises 37 commodities and 24 regions. It
incorporates recent updates of input-output data and bilateral trade, transport and
protection data.

In order to make this large data base operational, a standard modelling
structure has been developed. This permits users to conduct simulations in which
changes in policy, technology, population, and factor endowments are examined. In
addition to these shocks, the user specifies the split between endogenous and
exogenous variables (i.e., the closure). Behavioural parameters may also be altered.
Model results include a complete matrix of bilateral trade, activity levels by sector
and region, private and government consumption, regional welfare, and prices.

Although the disaggregated GTAP data set has 37 tradeable commodities and
24 regions, modellers work with aggregated versions of this data. The most
commonly used version of GTAP (this is the one used at the GTAP Short Course
held at Purdue University under the direction of Tom Hertel in August 1994) allows
for up to 10 tradeable commodities and 10 regions. We consider this version first
in section 2.1.1 below, and then consider briefly the fully disaggregated version in
section 2.1.2.

2.1.1 GTAP with up to 10 tradeable commodities and 10 regions

In the standard condensation of GTAP94[5], 7 vector variables (all of them taxes)
are omitted and 24 vector variables are backsolved for. The salient features of the
standard condensation of this model are shown in Table 2.1.1. The CPU times and

---

4   In the first three paragraphs below we quote directly from the August 1994 GTAP Short
    Course prospectus.
5   This is the condensation in the file TP1010.STI supplied to participants in the August
    1994 Short Course.

numbers of nonzeros reported there are for a simulation concerning MFA (Multifibre Agreement) liberalization. This simulation (see Yang *et al.* (1994)) is to analyse the removal of the MFA export quotas on textiles and clothing exported to North America and the European Union. In the GTAP data, these quotas are shown as export tax equivalents; shocks are given to these export taxes. The simulation is carried out with an aggregation containing 10 tradeable commodities and 10 regions, using the standard closure of GTAP.

Table 2.1.1: *Standard Condensation of GTAP94 with 10 Tradeable Commodities and 10 Regions*

Item	Uncondensed	Standard Condensation
Equations	14,411	3,151
Variables	22,652	8,462
Exogenous variables available for shocking	8,241	5,311
Variables omitted	nil	2,930
Variables backsolved for	nil	11,260
Endogenous results available	14,411	14,411
Nonzeros in Equations matrix	83,327	66,987
Nonzeros in left-hand-side matrix	68,796	35,136
Nonzeros after LU decomposition	153,453	90,837
Approximate memory required	8-9 Mb	6-7 Mb
CPU time for LU decomposition (66 MHz, 80486)	308 sec	38 sec
Total CPU time for Johansen solution (66 MHz, 80486)	339 sec	56 sec
Total CPU time for accurate solution (66 MHz, 80486)*	1,064 sec	494 sec

* This is obtained by extrapolating from 8,12,16-step Gragg calculations.

The following points are of interest:

(1) *Memory.* The memory required for the condensed version (6-7 Mb) is only slightly less than that required for the uncondensed version (8-9 Mb). However, in the context of the GTAP August 1994 Short Course, this was crucial since participants there worked in a lab of 8 Mb machines.

(2) *CPU time.* In Table 2.1.1 we show the total CPU time for an accurate nonlinear solution; in this case, the solution is obtained by extrapolating from 8,12 and 16-step Gragg calculations. In the case of this 10 × 10 aggregation of GTAP, the CPU saving from working with the condensed system is considerable (the condensed taking only 494 seconds for an accurate solution compared to 1,064 seconds with the uncondensed). This saving is achieved without losing access to any endogenous results (since all variables substituted out are backsolved for). However some flexibility in closure and shocks is sacrificed by working with the condensation because of the omissions and because the backsolved variables are hard-wired as endogenous. We also show in Table 2.1.1 the CPU time to LU decompose the left-hand-side matrix during step 1 and the total CPU time for a Johansen solution (which is the first step of a multi-step

calculation and includes the LU decomposition). The subsequent steps of a multi-step calculation are usually considerably faster than the first step because, when computing the LU decomposition, the sparse linear-equations routines MA28 (see Duff(1977)) are able to reuse pivots from the first step: in Table 2.1.1, the total CPU time for the 8,12,16-step calculations is only about three (for the uncondensed) or nine (for the condensed) times that of the Johansen CPU.

2.1.2 GTAP with the fully disaggregated data

The figures in Table 2.1.2 below are for the fully disaggregated data base with 37 tradeable commodities and 24 regions. The figures there are approximate since we did not actually solve the model with this data, but merely wrote a TABLO-generated program capable of so doing; TABLO produces the estimates shown in Table 2.1.2. The memory savings from working with the condensed system are considerable.

Table 2.1.2: *Standard Condensation of GTAP94 with*
37 Tradeable Commodities and 24 Regions

Item	Uncondensed	Standard Condensation
Equations (approx)	320,000	43,000
Variables (approx)	500,000	145,000
Exogenous variables available for shocking (approx)	180,000	102,000
Variables omitted (approx)	nil	78,000
Variables backsolved for (approx)	nil	277,000
Endogenous results available	all	all
Approximate memory required	70-75 Mb	43-48 Mb

2.2 MRES — a multiregional model of Australia

This model extends the 115-commodity, 113-sector MONASH model of the Australian economy by including equations to disaggregate results (in a tops-down manner) to the 6 states and 56 statistical divisions within Australia. This is, of course, a very large model and would remain so even if solved at a more aggregated level of industry and commodity detail.

The salient features of the standard condensation of this model are shown in Table 2.2. The CPU times and numbers of nonzeros are for the first year of the 10-year forecast reported in Adams, Horridge and others (1994) and Adams, Dixon and McDonald (1994).

Table 2.2: *The Standard Condensation of MRES*

Item	Uncondensed	Standard Condensation
Equations	7,982,020	25,176
Variables	8,049,255	35,968
Exogenous variables available for shocking	67,235	10,792
Variables omitted	nil	56,443
Variables backsolved for	nil	16,470
Endogenous results available	(a)	41,646
Nonzeros in Equations matrix	(a)	700,965
Nonzeros in left-hand-side matrix	(a)	623,767
Nonzeros after LU decomposition	(a)	669,975
Approximate memory required	(a)	40-42 Mb
CPU time for LU decomposition (66 MHz, 80486)	(a)	226 sec
Total CPU time for Johansen solution (66 MHz, 80486)[b]	(a)	2,331 sec

(a) Information not available. This version of the model is too large for us to solve without condensation.

(b) At the time of writing this paper, some of the statements for updating the data between steps of a multi-step calculation have not been included in the TABLO Input file for MRES. Accordingly we are not able to report the CPU time for an accurate solution.

For MRES, condensation is essential: without it the system (nearly 8 million equations) is far too large to solve even on most (possibly all) mainframes. The condensed system (25,176 equations with 623,767 nonzeros in the left-hand-side matrix) is quite sparse (only about 0.1 per cent of the entries in this matrix are nonzero).

2.3 ORANI-INT — an intertemporal model of Australia

This intertemporal model contains a 13-sector representation of the Australian economy with intertemporal optimization for investment and consumption; it can be solved over different time intervals. For this paper, we have solved it over a 30-year interval, using 31 grid points spaced a year apart. Experience shows that a minimum of 30 grid points is necessary to obtain satisfactory time paths in the solutions.

The salient features of the standard condensation of this model are shown in Table 2.3. The CPU times and numbers of nonzeros are for a simulation with the same shocks as in section 3 of Malakellis (1993), though with a different closure. In the simulation reported in Table 2.3, consumers are forward-looking as are investors in 10 of the 13 sectors[6], and real government expenditure is increased to 10 percent above control in each of years 10 through 30; this increase is anticipated 10 years in advance.

---

6 Investment at the second grid point is specified exogenously in one of these 10 sectors to keep it nonnegative.

Table 2.3: *Standard Condensation of ORANI-INT*

Item	Uncondensed	Standard Condensation
Equations	81,714	12,762
Variables	90,418	21,466
Exogenous variables available for shocking	8,704	8,704
Variables omitted	nil	nil
Variables backsolved for	nil	nil
Endogenous results available	81,714	12,762
Nonzeros in Equations matrix	308,023	119,078
Nonzeros in left-hand-side matrix	244,650	75,136
Nonzeros after LU decomposition	447,805	206,471
Approximate memory required	24-25 Mb	10-11 Mb
CPU time for LU decomposition (66 MHz, 80486)	334 sec	267 sec
Total CPU time for Johansen solution (66 MHz, 80486)	435 sec	319 sec
Total CPU time for accurate solution (66 MHz, 80486)*	1,416 sec	760 sec

*   This is obtained by extrapolating from 2,4,6-step Gragg calculations.

The following points are of interest.

(1) *Memory.* The condensed (10-11 Mb) requires considerably less memory than the uncondensed (24-25 Mb). Results for many endogenous variables are not available from the standard condensed version. However, most of those not available are of little interest; extra variables could be backsolved for (instead of being substituted out) with little extra cost in terms of memory or CPU.

(2) *CPU times.* The condensed version solves significantly more quickly that the uncondensed.[7] (For an accurate 2,4,6-step Gragg solution, the condensed takes 760 seconds compared to 1,416 seconds for the uncondensed.)

2.4 *Other general comments about condensation*

Condensation is always carried out on a system of linearized equations. Any levels equations in the TABLO Input file are automatically linearized by TABLO (which carries out symbolic differentiation) before condensation. For example, the levels equation

$$A = B + C*D$$

---

[7] In preparing the material for this paper, we found a surprising result when working with a variant of ORANI-INT with 41 grid points over a 40-year interval. For one simulation with this, the condensed version of the model solved more quickly than the uncondensed, as expected, on three machines (VAX Alpha, VAX/VMS and 33 MHz 40486), but the reverse was true on the 66 MHz 80486 PC used for the timings reported in this paper. We are especially surprised at the differences between the two 80486 PCs, where exactly the same executable image was running. We wonder if the anomaly is due to different size and/or memory caches on the two machines, and are grateful to Mark Horridge for discussions on this issue. We do not know of any other instances where the ranking of condensed and uncondensed solve times changes between machines.

is linearized to an equation equivalent to

A*p_A = B*p_B + C*D*[p_C+p_D]

(where p_A, p_B, p_C and p_D denote percentage changes in A, B, C, and D respectively).

During condensation, TABLO is able to rearrange equations to obtain an expression for the variable to be substituted out. For example, TABLO could rearrange the equation RORCURRENT from GTAP, namely

.   (all,r,REG) rorc(r) = GRNERATIO(r)*[rental(r) - pcgds(r)]

to obtain an expression equivalent to

(all,r,REG) rental(r) = {1/GRNERATIO(r)}*rorc(r) + pcgds(r)

for variable rental, if required.

There are rules restricting when an equation can be used to substitute out a variable. These are spelled out in section 2.3 of Harrison and Pearson (1994b).

The amount of condensation carried out can effect the CPU time for solving the model. In some cases, extra condensation increases the total CPU time. Section 5.4 of Harrison and Pearson (1994c) discusses this issue and describes a procedure used in TABLO to reduce the computational complexity during condensation.

The AMPL software (see section 13.2 of Fourer *et al.* (1993)) can also do automatic substitution,[8] in its case to reduce the number of constraints passed to its solver. Users of AMPL can ask it to make all substitutions possible or can indicate specific substitutions by using a special syntax.

As indicated above, GEMPACK currently uses the sparse linear-equations routines MA28 (Duff (1977)). The next release of GEMPACK will offer the newer MA48 routines — also developed by AEA Technology at Harwell (see Duff and Reid (1993)) — as an alternative to MA28; our initial testing indicates that MA48 is significantly faster than MA28 for several multiregional and intertemporal models (including GTAP, MRES and ORANI-INT).

3. Separation of Theory, Data and Simulation Specification

Within GEMPACK, there is a clear separation between the theory of a model, its data and the other information required to specify a simulation (notably the closure and shocks).

This separation leads to flexibility in modelling and simplifies the maintenance of a model once built. Each component, dealing with one well-defined aspect of the problem, is easily understood, while the connections between the three components are direct, obvious, and kept as simple as possible. As indicated in Figure 1.1, in

---

8   We are grateful to Paul Preckel for drawing this to our attention.

order to carry out a simulation, a modeller needs to specify the *theory* of the model in a TABLO Input file, attach the relevant *data* in one or more data files, and specify the *closure* and *shocks* in a Command file (or from the terminal).

The theory is written down in the TABLO Input file for the model. While this includes schematic information about the data, it rarely includes actual data. For example, the TABLO Input file GTAP94.TAB for GTAP contains the statements

COEFFICIENT (all,i,TRAD_COMM) (all,r,REG) VIPA (i,r)

! Private household expenditure on imported i ! ;

READ (all,i, TRAD_COMM) (all,r,REG) VIPA(i,r)

FROM FILE gtapdata HEADER "VIPA";

which indicate that a matrix of data is to be read from a file referred to as "gtapdata" at header "VIPA". The name "gtapdata" is merely the logical name by which this file is known within GTAP94.TAB; as will be seen below, the name of the actual file containing the data can be quite different. Binary data files in GEMPACK are so-called Header Array files; each array of data has an associated 4-character header ("VIPA" in the example above) which is used to locate the data quickly and to distinguish it from other arrays of data on the same file.

For each model (that is, each TABLO Input file) there can be several associated logical data files. The actual data files can be Header Array files (which are binary files) or text files. For example, for GTAP, there are three associated data files with logical names

GTAPSETS, GTAPDATA, GTAPPARM.

The first two correspond to Header Array files respectively containing information about the numbers and names of the commodities and regions (GTAPSETS) and the global data base including input-output and trade data (GTAPDATA). The third is a text file containing the values of the parameters (such as elasticities of substitution).

The use of logical file names enables us to keep data and theory separate. We are free to use two or more data sets with the same model theory. This is useful in situations where different problems require different aggregations. In routine policy-analytic work at the Australian Industry Commission, different applications of AGE models from the ORANI suite[9] regularly require differing degrees of resolution with respect to different sectors of the economy. With different aggregations of data, it is possible to focus on different details of the economy and to solve different problems on the one standard model. For an extensive discussion of how off-the-shelf modelling shells have been used in practice, see Dee (1994).

The ability to use data of different sizes with the same model assists in testing prototypes. Small data sets can be used to check the model framework and data structure initially, before all the data for the full-sized model have been assembled. Small data sets are also useful when teaching others about a model.

---

9   The core model is documented in Dixon *et al.* (1982).

Finally, the extra information required to specify a simulation with the model (notably, the closure and shocks) is separate again. Usually this is given as a Command file (or it can be given interactively when the TABLO-generated program or GEMSIM is run). For example, a Command file to carry out a numeraire simulation with GTAP may contain the statements

```
exogenous pop psave ao af afe ava atr
            to txs tms tx tm
            profitslack incomeslack endwslack
            cgdslack saveslack govslack tradslack
            qo(ENDW_COMM,REG);
rest endogenous;
shock psave = 10;                                                    .
```

These specify the closure by listing the exogenous variables and indicating that the remaining variables are endogenous. The last statement indicates that variable psave (this is the price of the global savings commodity, which is the numeraire of GTAP) is to be increased by 10 per cent.

To connect the three components — theory, data, and closure/shocks — together,

(a) the model theory is specified either by the name of the TABLO-generated program or by the Auxiliary file names for GEMSIM;

(b) the logical names of the data files are equated to the corresponding actual file names in the Command file for the particular simulation; and

(c) the closure and shocks are specified in the Command file mentioned in (b).

Once a modeller has prepared a suitable Command file, say called SIM.CMF, the *only responses needed to run the TABLO-generated program or GEMSIM are*

cmf   ! Take inputs from a Command file

sim.cmf   ! Name of the Command file

More information about specifying a simulation and Command files can be found in section 3.2 of Harrison and Pearson (1994c).

Below, in sections 3.1 and 3.2, we illustrate separation between theory, data and simulation by considering GTAP in this context in more detail.

3.1 GTAP94: separate specification of theory and data

During the 1994 GTAP Short Course, participants worked with 9 different aggregations of the GTAP data. However they only needed one version of the theory, the TABLO Input file GTAP94.TAB. This is one clear advantage of the separation between theory and data. The TABLO Input file GTAP94.TAB does not fix the numbers of tradeable commodities or regions; rather this information is read from the GTAPSETS file via statements such as

```
SET REG # Regions in the model #  MAXIMUM SIZE 10
            READ ELEMENTS FROM FILE gtapsets HEADER "H1";          .
```

(The only reason for "MAXIMUM SIZE 10" is to ensure that the TABLO-generated program for the model will run on a PC with 8 Mb of memory.) Thus simulations with GTAP94 can all be carried out by running the same TABLO-generated program.[10]

Which aggregation of the data base to use is indicated by the appropriate "file" statements in the Command file. These statements are of the form

file <logical name in TABLO Input file> =
            <name of file actually containing the data to be used> ;

In GTAP94, the following statements identify sets, data and parameters respectively:

file gtapsets = set2-04.har;
file gtapdata = dat2-04.har;
file gtapparm = par2-04.dat;

Of the nine aggregations used during the Short Course, these files jointly identify aggregation number 4, which has 8 tradeable commodities and 6 regions.

3.2 GTAP94: separate specification of shocks

One of the uses for GTAP is to simulate scenarios in which various wedges (for example, import tariffs or output subsidies) are removed. GTAP supplies a TABLO Input file (called SHOCKS.TAB) for calculating the sizes of the wedges in an existing data base and for calculating the sizes of the shocks to the appropriate instruments to simulate the removal of these wedges. For example, SHOCKS.TAB contains the following statements for calculating the power TMS_L of the import tariff (that is, one plus the *ad valorem* rate).

FORMULA (all,i,TRAD_COMM) (all,r,REG)(all,s,REG)

$$TMS\_L(i,r,s) = VIMS(i,r,s)/VIWS(i,r,s);$$

FORMULA (all,i,TRAD_COMM)(all,r,REG) (all,s,REG)

$$TMS\_HAT(i,r,s) = \{[1-TMS\_L(i,r,s)]/TMS\_L(i,r,s)\}*100;$$

Here TMS_L(i,r,s) is the power of the import tariff levied in region s on commodity i from region r. [VIMS and VIWS are respectively the dollar values at market and world (i.e., cif) prices.] Above, TMS_HAT(i,r,s) is the percentage change in this power of the tariff required to eliminate this wedge (that is, to change it to 1, which is its value when there is no tariff). For example, if VIMS(i,r,s) = 120 and VIWS(i,r,s) = 100 for some i, r and s, then TMS_L(i,r,s) = 1.2 (that is, a 20 per cent *ad valorem* tariff) and TMS_HAT(i,r,s) is –16.6667 since reducing 1.2 by 16.6667 per cent will reduce it from 1.2 to 1.0.

The file SHOCKS.TAB contains statements to write the shocks to various files. For example, it includes the statements

---

10 The TABLO-generated program TP1010.FOR is the one produced by running TABLO on GTAP94.TAB and carrying out the standard condensation, as discussed in section 2.1 above. The executable image TP1010.EXE is produced by compiling and linking this program, which is achieved by the command

ltg tp1010

FILE(NEW,TEXT) TMSHAT # The file with shocks to obtain TMS = 1 # ;
WRITE TMS_HAT TO FILE TMSHAT;

If the actual file containing these shocks is TMSHAT.DAT, then the statement

Shock tms = file tmshat.dat ;

in a Command file will give shocks to remove all import tariffs. If removal of only one tariff is desired, for example, removal of the tariff in the European Union E_U on food imported from the USA, this can be achieved via the statement

Shock tms("food","USA","E_U")= select from file tmshat.dat;

This illustrates the separability between theory, data and shocks. It also illustrates a different kind of TABLO Input file, namely files like SHOCKS.TAB, which do not specify the equations of a model, but rather do data-driven calculations.

4. *Using Post-Simulation Data*

There are two sorts of results from a simulation carried out using GEMPACK. The first are the percentage changes (or changes) in the endogenous variables of the model produced by the shocks; these are put on the Solution file — see Figure 1.1. The second are the updated (or post-simulation) data file(s); these typically show the input-output and other data as they would be given the changed values of the exogenous (and consequently the endogenous) variables.

For example, after a GTAP simulation, the updated global data file (the one with logical name GTAPDATA) contains post-simulation input-output and trade data (usually dollar values). This can be thought of as the world economy as it would be if the shocks were applied (for example, if the specified wedges were removed). In section 4.1 we discuss how these updated data are used in carrying out forecasts with MRES.

4.1 *Updated data and MRES forecasts*

The MRES model (see section 2.2 above) has been used recently to produce forecasts for the period 1992-93 to 2000-01 for 112 Australian industries at the national level, and also in the 6 states and in the 56 statistical divisions making up Australia.

Input (as exogenous shocks to the model) are

•   assumptions about export prospects for Australia's traditional exports (from ABARE, the Australian Bureau of Agricultural and Resource Economics),

•   assumptions about changes in protection (from the Australian Industry Commission),

•   assumptions about exports of services to international tourists (from the Australian Bureau of Tourism Research),

•   various macroeconomic assumptions (from Syntec Economic Services), and

- assumptions about changes in technology and tastes (from the Centre of Policy Studies, Monash University).

The exogenous assumptions may vary from year to year in the forecast period. For example, ABARE expects that exports of wool will fall by 2.3 per cent in 1996-97 but rise by 4.0 per cent in 1997-98.

In the forecasts reported in Adams, Horridge and others (1994), Adams, Dixon and McDonald (1994) and CoPS/Syntec (1994), the base year was 1991-92. Firstly the exogenous shocks for the year 1992-93 were fed in and the model was solved to produce a forecast of the 1992-93 data; this is the updated or post-simulation data output by the TABLO-generated program. Then these 1992-93 data were used as the new starting point for the model, the shocks for 1993-94 were fed in and the model was solved for this second year of the forecast. This was repeated to give nine year-on-year linked annual solutions. Results are typically reported as average-annual growth rates over the whole period. For example, domestic production of the Communication industry is forecast to have an average annual growth of 7.1 per cent for the period 1992-93 to 2000-01, compared to an average annual growth of 7.8 per cent for the historical period 1986-87 to 1992-93. Total value added in the Far North Statistical Division (i.e., the Cairns region) of Queensland is forecast to have an average-annual growth rate of 4 per cent over the period of the forecast.

The automatic production of post-simulation data makes the calculation of year-on-year linked annual simulations relatively straightforward. For example, the statements,

file *FID* = fid91.har;
updated file FID = fid92.har;

in the Command file for the first year of the forecast indicates that the 1991-92 data in FID91.HAR are the starting point and that the post-simulation data for this year of the forecast are to go in the file called FID92.HAR. The corresponding lines of the Command file for the second year of the forecast are

file FID = fid92.har;
updated file FID = fid93.har;

The role of FID92.HAR has changed here; now it is the starting point for this second year of forecast.

The use of updated data files in linked annual solutions for forecasting with an AGE model is a further example of the advantages of maintaining separation between data and the other components of a model (as discussed above in section 3).

5. *General-purpose Algorithm for Solving Intertemporal Models*

Dynamic equations (that is, those involving more than one time instant) can be forward- or backward-looking. For example, the capital accumulation equation

$$K(t+1) = K(t).D + I(t)$$

is backward-looking in the sense that the current stock of capital (at time $t+1$) depends on the stock of capital $K(t)$ and investment expenditure $I(t)$ from the

previous instant.[11] In ORANI-INT (see Malakellis (1993, 1994)), current investment decisions depend on future returns from investment, which results in forward-looking behavioural equations.

If all behaviour is backward-looking, a dynamic model can be solved as a series of linked single-period models, and no special intertemporal algorithm is required. This is how MRES forecasts (see section 4) are calculated.

However, as soon as any forward-looking behaviour is involved, the model cannot be solved in this way. The problem is that some initial-period behaviour is not known in the first period to be simulated (since it is dependent upon outcomes in the future). Terminal conditions holding at the end of the simulation (for example, convergence to a steady state or to balanced growth) are simple examples of conditions that must be satisfied if agents have forward-looking behaviour.

There are various ways of handling forward-looking behaviour numerically. The frequently used shooting or multiple shooting methods involve guessing the unknown initial values, solving forward and correcting these guesses depending on how far away from the terminal conditions this leaves the solution. A brief survey of these methods can be found in chapter 5 of Dixon *et al.* (1992). These methods tend to become slower as the amount of forward-looking behaviour increases.

GEMPACK adopts a different strategy for solving intertemporal models, namely to solve the equations *simultaneously at all time instants*. (This is the procedure recommended in Dixon *et al.* (1992) as used over the last several years by Wilcoxen – see, e.g., Wilcoxen (1985).) This has the advantage of being truly general-purpose. No user intervention is required to signal forward-looking behaviour or to cope with changes in the number of costate variables when the closure changes. Increasing the amount of forward-looking behaviour (for example, modelling sector-specific investment rather than its economy-wide counterpart) has little impact on solution times.

More details of the procedure for implementing intertemporal models via GEMPACK are given in Codsi *et al.* (1992).

5.1 *ORANI-INT*

As an example, consider the simulations with ORANI-INT reported in Malakellis (1993). There ORANI-INT is solved over a thirty-year time horizon. Real government expenditure is shocked so that its value is 10 percent above control in each of years 10 through 30. The effects of the shock are examined under alternative

---

11 A continuous-time model must be approximated by a discrete-time version before it can be solved numerically. This involves selecting a finite number of grid points representing time instants in the time-space of the original model. Any derivatives in the original model are replaced by finite difference approximations. For example, $K't$ is often replaced by

$$[K(t+1) - K(t)]/LNG(t)$$

where $K(t+1)$ and $K(t)$ are the values of K at the adjacent time instants and $LNG(t)$ is the length (say, number of years) of the interval between t and $t+1$. (See Exercise 5.12 in Dixon *et al.* (1992) and Codsi *et al.* (1992) for more details.) When creating a discrete-time approximation, a decision must be taken with flow variables (such as investment) as to whether their values at t represent the flow over the next period, the previous period, or some other alternative. Modellers must also be careful that the choice of grid points does not alter important properties of their models, such as steady-state values: see Mercenier and Michel (1994a,b).

specifications of the investment theory. Malakellis (1993) reports the results of the three simulations described below.

In the reference simulation, investors are assumed to have static expectations. The results of the reference simulation are contrasted with those of two simulations in which investors are forward-looking. In the first of the forward-looking simulations the shock to government expenditure is anticipated whilst in the second it is not. Because the focus of these simulations is on the investment theory, the path of consumption is set exogenously in all three simulations.

When investors are forward-looking, there is one forward-looking investment equation for each of 10 sectors (and exogenous investment in the remaining 3 sectors): thus 10 costate variables are involved.

Detailed explanations of the results in Malakellis (1993), and of results in which consumers are forward-looking, can be found in Malakellis (1994).

As can be seen from the CPU times in Table 2.3 above, the simultaneous solution strategy used in GEMPACK can handle models of this size and complexity quite satisfactorily. The ability to condense the model algebraically before solving it (see section 2) is important in reducing the size of the simultaneous system to be solved numerically.

References

Adams, Philip D., Peter B. Dixon and Daina McDonald (1994) 'MONASH Forecasts of Output and Employment for Australian Industries: 1992-93 to 2000-01', *Australian Bulletin of Labour*, Vol. 20, No.2, pp. 83-109.

Adams, Philip D., Mark Horridge and others (1994), *Guide to Growth*, Vol. 5. [Available from Syntec Economic Services, Melbourne.]

Codsi, G., K.R. Pearson and P.J. Wilcoxen (1992) 'General-Purpose Software for Intertemporal Economic Models', *Computer Science in Economics and Management* vol. 5, pp. 57-79.

CoPS [Centre of Policy Studies, Monash University]/Syntec (1994) *Briefing Folder*, June, pp. 300 (approx). [Available from Syntec Economic Services, Melbourne.]

Dee, Philippa (1994) 'General Equilibrium Models and Policy Advice in Australia', Industry Commission, Belconnen ACT 2616 Australia. Paper presented at International Federation of Automatic Control Workshop in Economics and Finance, Amsterdam (June).

Dixon, P.B., B.R. Parmenter, A.A. Powell and P.J. Wilcoxen (1992) *Notes and Problems in Applied General Equilibrium Economics*, North-Holland, Amsterdam.

Dixon, P.B., B.R. Parmenter, J. Sutton and D.P. Vincent (1982) *ORANI: A Multisectoral Model of the Australian Economy* , North-Holland, Amstredam.

Duff I.S. (1977) 'MA28 - A Set of FORTRAN Subroutines for Sparse Unsymmetric Linear Equations', *Harwell report* R.8730 (HMSO, London), pp. 104.

Duff, I.S. and J.K. Reid (1993), 'MA48, a Fortran Code for Direct Solution of Sparse Unsymmetric Linear Systems of Equations', Report No. RAL-93-072, Rutherford Appleton Laboratory.

Fourer, Robert, David M. Gray and Brian W. Kernighan (1993) *AMPL: A Modeling Language for Mathematical Programming*, The Scientific Press, South San Francisco.

Harrison, W.J. and K.R. Pearson (1993), 'How to Create and Modify GEMPACK Header Array Files Using the Program MODHAR', *GEMPACK Document* No. GPD-3, third edition, Monash University, April, 27+4.

Harrison, W.J. and K.R. Pearson (1994a), 'An Introduction to GEMPACK', *GEMPACK Document* No. GPD-1, second edition, Monash University, April, pp. 252+15.

Harrison, W.J. and K.R. Pearson (1994b), 'User's Guide to TABLO, GEMSIM and TABLO-generated Programs', *GEMPACK Document* No. GPD-2, second edition, Monash University, April, pp. 138+14.

Harrison, W.J. and K.R. Pearson (1994c), 'Computing Solutions for Large General Equilibrium Models Using GEMPACK', Centre of Policy Studies and the Impact Project, *Preliminary Working Paper* No. IP-64, Monash University, June, pp. 55.

Harrison, W.J., K.R. Pearson, A.A. Powell and E.J. Small (1993) 'Solving Applied General Equilibrium Models Represented as a Mixture of Linearized and Levels Equations', Centre of Policy Studies and the Impact Project, *Preliminary Working Paper* No. IP-61, Monash University, September, pp. 20. [A modified version is to appear in *Computational Economics.*]

Harrison, W. Jill and E. John Small (1993), 'TABLO Input Files for the Stylized Johansen, Miniature ORANI and ORANI-F Models', Centre of Policy Studies and the Impact Project, *Computing Document* No. C11-01, Monash University, September, pp. i + 29.

Hertel, Thomas W. and Marinos E. Tsigas (1994), 'Structure of the Standard GTAP Model', Short Course in Global Trade Analysis, Purdue University, August, pp. 56. [This is to be Chapter 2 of the forthcoming book: *Global Trade Analysis Using the GTAP Model*, edited by Thomas W. Hertel.]

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.

Malakellis, Michael (1993), 'Illustrative Results from ORANI-INT: An Intertemporal GE Model of the Australian Economy', Centre of Policy Studies and the Impact Project, *Preliminary Working Paper* No. OP-77, Monash University, July, pp. 40.

Malakellis, Michael (1994) 'ORANI-INT: An Intertemporal Computable General Equilibrium Model of the Australian Economy', Ph.D Thesis, Monash University. [Expected to be submitted in September 1994.]

Mercenier, Jean and Philippe Michel (1994a) 'Discrete-time Finite Horizon Approximations of Infinite-Horizon Optimization Problems with Steady-State Invariance', *Econometrica*, Vol. 62, pp. 635-656.

Mercenier, Jean and Philippe Michel (1994b) 'A Criterion for Time Aggregation in Intertemporal Dynamic Models', *Mathematical Programming* (to appear).

Wilcoxen, Peter J. (1985), 'Numerical Methods for Investment Models with Foresight', University of Melbourne, Impact Project *Preliminary Working Paper* No. IP-23, pp. 47.

Yang, Yongzheng, Will Martin and Koji Yanagishima (1994), 'Analysis of the Multifibre Agreement in the Context of the Uruguay Round'. [This is to be a chapter in the forthcoming book: *Global Trade Analysis Using the GTAP Model*, edited by Thomas W. Hertel.]