CENTRE of POLICY STUDIES and the IMPACT PROJECT

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

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

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

ADDING ACCOUNTING-RELATED BEHAVIOUR TO A MODEL IMPLEMENTED USING GEMPACK

by

W.J. HARRISON

and

K.R. PEARSON

Centre of Policy Studies, Monash University

Computing Document No. C12-01 June 2002

ISSN 1 031 9034 ISBN 0 7326 1534 8

Adding Accounting-Related Behaviour to a Model Implemented using GEMPACK

W. Jill Harrison and K. R. Pearson

September 1999; Minor revisions made June 2002

Abstract

Many modellers need to add new behaviour to an existing model which is implemented using GEMPACK. In many such cases, the new behaviour is intimately connected with the accounting relations in the model. An example is the task of adding new taxes to a model. The aim of this paper is to give some guidance to modellers faced with such model extensions.

Most models implemented in GEMPACK are written down as systems of linearised equations. Others are written as systems of mixed levels/linearised equations or of just levels equations. Adding accounting-related behaviour to linearised TABLO Input files is probably less straight forward than adding it to mixed or completely levels files.

Irrespective of the style of the equations in the original TABLO Input file, the first task is to write down the levels equations describing the new behaviour.

New data may be required, and care must also be taken with this. In particular, the new data must be set up so that the new levels equations of the model are satisfied in the pre-simulation data base (as well as the equations of the existing model).

In the paper a relatively simple example of accounting-related behaviour is considered, namely adding a form of consumption tax to the Stylized Johansen model. The paper describes how this behaviour can be added to the standard linearised TABLO Input file SJLN.TAB for Stylized Johansen. In particular the paper shows how the new behaviour can be added as linearised equations or as levels equations.

1 INTRODUCTION 1

2 ADDING A CONSUMPTION TAX TO STYLIZED JOHANSEN 2

2.1 Modified Data Base 2

2.2 New Variables and Equations 3

2.3 Modified equations 4

2.4 Extra Reads 4

2.5 Closure changes 5

2.6 Initial Levels Solution of Modified Model 5

2.7 Checking the Initial Data 5

2.8 Alternative TABLO Input Files for the New Model 5

2.9 Overview of the SJHT examples 6
2.9.1 Working with the Example Files Provided 6

2.10 Linearised TABLO Input file SJHT1.TAB 7
2.10.1 The New Linear Variables 7
2.10.2 Extra Coefficients and Formula 7
2.10.3 The New Equations 8
2.10.4 The Modified Equations and Formula 10
2.10.5 The Equation for GOVEXP_ALT 11
2.10.6 The Complete TAB file SJHT1.TAB 12

2.11 Linearised TABLO Input File SJHT2.TAB (handles zero taxes) 13
2.11.1 The New Variables 13
2.11.2 The New Equations 13
2.11.3 The Modified Equations and Formula 15
2.11.4 The Equation for GOVEXP_ALT 15
2.11.5 The Complete TAB file SJHT2.TAB 16

2.12 TABLO Input file SJHT3.TAB with New Equations in Levels 17
2.12.1 Default Statements 17
2.12.2 Levels Variables 17
2.12.3 Connections with the Original Model 18
2.12.4 Setting Initial Values of Levels Variables 19
2.12.5 The New Behavioural Equations 19
2.12.6 The Modified Equations and Formula 20
2.12.7 The Equation for GOVEXP_ALT 20
2.12.8 The Complete TAB file SJHT3.TAB 20

3 CHECKING THE IMPLEMENTATION 21

3.1 Data Files 21

3.2 SJHTCHK.TAB for Checking Initial and Updated Data 21

3.3 Command files and Simulations **22**
3.3.1 Checking the Reversal Simulations 23
3.3.2 Adding Tiny Values to the Data Base Is Surprisingly Successful 23
3.3.3 Could This Checking Detect Errors? 24
3.3.4 Problem with SJHT1-X.CMF 25

4 USING THE EXAMPLE FILES PROVIDED 25

5 APPENDIX A - ADDITIONS FOR SJHT1.TAB 26

5.1 Changes to the Original TAB File 26

5.2 Main Additions 26

6 APPENDIX B - ADDITIONS FOR SJHT2.TAB 29

6.1 Changes to the original TAB file 29

6.2 The Main Additions 29

7 APPENDIX C - ADDITIONS FOR SJHT3.TAB 31

7.1 Changes to the original TAB file 31

7.2 The Main Additions 31

8 REFERENCES 34

1 Introduction

Many modellers need to add new behaviour to an existing model which is implemented using GEMPACK. In many such cases, the new behaviour is intimately connected with the accounting relations in the model. An example is the task of adding new taxes to a model.[1] The aim of this note is to give some guidance to modellers faced with such model extensions.

Most models implemented in GEMPACK are written down as systems of linearised equations. Others are written as systems of mixed levels/linearised equations or of just levels equations. Adding accounting-related behaviour to linearised TABLO Input files is probably less straight forward than adding it to mixed or completely levels files.

Irrespective of the style of the equations in the original TABLO Input file, the first task is to write down the **levels** equations describing the new behaviour. If this behaviour is being added to a TABLO input file containing all linearised equations, there are two possibilities.

- The new equations can be added as linearised equations. In this case, the levels equations describing the new behaviour must be linearised following the usual procedures (which are based on differentiating the levels equations).

- The new equations can be added as levels equations, even though all the other equations are linearised. This may be the easier way to go since it does not require modellers to differentiate the new levels equations. However, it does require some familiarity with mixed levels/linearised models and with the somewhat specialised techniques required to append several levels equations to a linearised file. We describe this procedure later in this paper.

We recommend that all modellers faced with this sort of task consider seriously the option of adding the new behaviour as levels equations since especially accounting-related behaviour is usually more naturally written in levels form than in linearised form.

In either case, new data may be required. Care must also be taken with this. In particular, the new data must be set up so that the new levels equations of the model are satisfied in the pre-simulation data base (as well as the equations of the existing model).

In the rest of this note we consider a relatively simple example of accounting-related behaviour, namely adding a form of consumption tax to the Stylized Johansen model. We show how this behaviour can be added to the standard linearised TABLO Input file SJLN.TAB for Stylized Johansen. We describe how the new behaviour can be added as linearised equations or as levels equations.

The techniques and examples given in this paper can be used with Release 6.0 (October 1998) or later of GEMPACK. Minor simplifications are possible with Release 8.0 (October 2002) – see the footnote in section 2.12.3 below.

[1] This is not a discussion of the task of adding a CES nest to an existing model, since in that case it is natural to add the new equations as linearised equations, not levels. We have included "accounting-related" in the title to try to indicate that we are focusing here on behaviour (like taxes) which is most naturally expressed via levels equations.

2 Adding a Consumption Tax to Stylized Johansen

This example is written as an extension of the linearised TABLO Input file (usually called SJLN.TAB) for the Stylized Johansen model (see, for example, section 3.5 of GEMPACK document GPD-1). We use here the notation of that file SJLN.TAB without further introduction. A complete modelling description of the Stylized Johansen model can be found in chapter 3 of DPPW (Dixon *et al*, 1992).

The consumption tax to be modelled here is a tax levied only on purchases by households (not by firms). The revenue from this tax is assumed to be collected by a new agent, the **government**. The government is assumed to use the tax so collected to purchase an appropriate bundle of the commodities produced in the economy. (We assume that, as for households, government maximizes a Cobb-Douglas utility function subject to the obvious revenue constraint.)

2.1 Modified Data Base

It is vital to have an initial data base which represents a **levels** solution to the model in question.[2] We need a data base which represents a solution to the model with household taxes and government added. It is natural to start from the standard data base for Stylized Johansen and try to modify this to give a data base satisfying the model. Clearly this means we must add data showing the dollar value of taxes and government expenditure on commodities.

There are two ways of doing this.

- Try to add nonzero taxes and government expenditure data to the existing data base. To ensure that the data base created is a solution to the extended model, we must ensure that the relevant data balancing conditions hold. This means that total tax revenue should equal total government expenditure on commodities and, as for the basic model, the total value of primary factor inputs to production must equal the total value of household consumption. Whether or not the total value of household expenditure in that latter condition should include or exclude the new taxes requires a little thought – in fact, it should include the taxes. It would not too difficult to add tax and government data to achieve these ends.[3]

[2] The usual way of inferring a levels solution of the model is to set basic prices equal to one and then to infer quantities by dividing dollar values by prices. The levels values of non-basic prices can be inferred from the relative sizes of the dollar value amounts including and excluding the tax in question.

[3] One such data base is shown below. Taxes totalling 0.6 dollars are shown. [Note that while the initial tax rates on the two commodities are equal in this data, the theory does not assume they will remain equal.]

	Industry 1	2	Households	Government	Totals
Commodities	4	2	2	0.3	8.3
(Basic prices)	2	6	4	0.3	12.3
Primary factors					
Labour	1.2	3.2			4.4
Capital	1.1	1.1			2.2
Tax by			0.2		0.2
Commodity			0.4		0.4
	8.3	12.3	6.6	0.6	

- Try to use the existing data base as is. This would mean assuming that there are zero taxes and zero government expenditure in the data base. Clearly this will give us a solution to the new model in which the base power of the household taxes is 1 (that is, the *ad valorem* rates are zero).

Even in the second case where the new entries are zero, slots must be added in the data base scheme for these two items of data. Even when the initial data shows no taxes, post-simulation data will have taxes once the rates are shocked from zero to a nonzero value.

Of course, in a real model (rather than a pedagogical one such as Stylized Johansen), you may need to seek extra outside data to add to your data base. If this extra data is not readily available, there may be some virtue in starting with a data base in which the new taxes are initially zero (or nearly zero) and using a simulation with the model to introduce nonzero taxes (by giving appropriate and plausible shocks to the tax rates).

In the files associated with this note, this is what we recommend. First add slots (with zero values) for the taxes and government expenditure. Then shock the model (the powers of the taxes are naturally exogenous) to produce a data base containing nonzero taxes.

It is most important to start from initial data which represents a levels solution of the (extended) model. All GEMPACK simulations are calculated as perturbations (via the shocks) of the model from this initial solution. If the starting data base is not a solution of the model, the simulation results will not be meaningful.

2.2 New Variables and Equations

The new levels entities are

GOVINC total government income
GOVEXP total government expenditure
XGOV(i) **i in SECT**, quantity of government purchases of each commodity
DVGOV(i) **i in SECT**, value of government purchases of each commodity
DVHOUSTAX(i) **i in SECT**, value of taxes paid by households on each commodity
THOUS(i) **i in SECT**, power of the consumption taxes on each commodity
PHOUS(i) **i in SECT**, price (including tax) paid by households for commodities

The obvious levels equations are (in TABLO notation):

```
Equation E_DVGOV
(All,i,SECT)   DVGOV(i) = PC(i) * XGOV(i) ;
Equation E_DVHOUSTAX
(All,i,SECT)   DVHOUSTAX(i) = [THOUS(i) – 1] * DVHOUS(i) ;
Equation E_PHOUS
(All,i,SECT)   PHOUS(i) = PC(i) * THOUS(i) ;
Equation E_GOVINC
      GOVINC = SUM(i,SECT, DVHOUSTAX(i)) ;
Equation E_GOVEXP
      GOVEXP = GOVINC ;
Equation E_XGOV
(All,i,SECT)   XGOV(i) = AGOV(i) * GOVEXP / PC(i) ;
```

The first two equations E_DVGOV and E_DVHOUSTAX connect dollar values with the associated prices and quantities. The third E_PHOUS connects the dollar values at the two different prices (one including the tax and the other excluding it). The fourth E_GOVINC says

that government income consists of all tax revenue and the fifth E_GOVEXP says that the government spends all its income.

The last equation above E_XGOV is the solution to the government's purchasing decision, maximising utility U where

$$U = XGOV(1)^{AGOV(1)}\ XGOV(2)^{AGOV(2)}$$

subject to the budget constraint

```
GOVEXP = SUM(i,SECT, DVGOV(i)) .
```

Here the new parameters **AGOV(i)** are given by the formula

```
(All,i,SECT)   AGOV(i) = DVGOV(i) / GOVEXP ;
```

(which is analogous to the formula for the parameters ALPHAH(i) in the standard levels TABLO Input file SJLV.TAB for Stylized Johansen, as distributed with Release 6.0 of GEMPACK).

For checking purposes, it is probably useful to add a variable **GOVEXP_ALT** reporting an alternative measure of total government expenditure. [Of course, GOVEXP_ALT should equal GOVEXP.] The new variable is

GOVEXP_ALT alternative way of calculating total government expenditure

and the associated levels equation is

```
Equation E_GOVEXP_ALT
     GOVEXP_ALT = SUM(i,SECT, DVGOV(i)) ;
```

2.3 Modified equations

The price paid by households for commodities is now PHOUS(i) rather than the PC(i) in the original model. The price PHOUS(i) must replace the price PC(i) in the (linearised) equation **Consumer_demands** in the original model.

Market clearing of commodities (the equation called **Com_clear** in SJLN.TAB) must now include the term for government purchases. The levels version of this must be changed to

```
Equation Com_clear (All,i,SECT)
     XCOM(i) = SUM(j,SECT, XC(i,j)) + XH(i) + XGOV(i) ;
```

We describe suitable versions of this for each of the TABLO Input files described below.

2.4 Extra Reads

The new data vectors for **DVGOV(i)** and **DVHOUSTAX(i)** for **i in SECT** are assumed to be added to the usual Header Array data file for Stylized Johansen at headers "GCON" and "HTAX" respectively. These are read via the statements

```
READ  DVGOV from file data header  "GCON" ;
READ  DVHOUSTAX from file data header  "HTAX" ;
```

2.5 Closure changes

The normal change to the closure would be to take the two new tax rates **THOUS(i)** as exogenous. All other new variables are endogenous and all existing variables retain the same exogenous/endogenous status. [One alternative would be to set the government demands for commodities **XGOV(i)** exogenous and leave the tax rates to be determined endogenously.]

2.6 Initial Levels Solution of Modified Model

As with ordinary Stylized Johansen, this is easily obtained. The usual way is to set all basic prices [**PC(i), PF(f)**] equal to 1 and infer the obvious quantities from the dollar values.

2.7 Checking the Initial Data

Starting from the original TABLO Input file SJCHK.TAB used to check whether the data base for Stylised Johansen is balanced, a new extended TABLO Input file SJHTCHK.TAB has been written. After reading values of DVGOV(i) and DVHOUSTAX(i), new coefficients GOVINC and GOVEXP are calculated. For a balanced database, the government expenditure GOVEXP should equal the government income GOVINC.

```
Coefficient GOVEXP
 # Dollar value of total government expenditure # ;
Coefficient GOVINC
 # Dollar value of total government income # ;
Coefficient (All,i,SECT) THOUS(i)
 # power of tax paid by households for each commodity # ;
Formula GOVEXP = SUM(i,SECT, DVGOV(i) ) ;
Formula GOVINC = SUM(i,SECT, DVHOUSTAX(i) ) ;
```

There is also a formula to calculate the initial power of tax paid by households for each commodity:

```
Formula (All,i,SECT) THOUS(i) = 1 + {DVHOUSTAX(i)/DVHOUS(i)} ;
```

2.8 Alternative TABLO Input Files for the New Model

As indicated in the introduction, we can choose to add the new equations as linearised ones or as levels ones. We discuss the two alternatives in detail below. We call the two alternative TABLO Input files **SJHT1.TAB** and **SJHT3.TAB** respectively. [Here "**HT**" stands for household tax.]

- In SJHT1.TAB, all new equations are linearised.
- In **SJHT3.TAB**, the new equations are added as levels equations where this seems most natural.

In fact it makes a difference in the linearised version whether or not we allow for the cases where the taxes are zero. This leads to two linearised files

- **SJHT1.TAB** (which does not handle zero taxes) and
- **SJHT2.TAB** (where zero taxes are allowed for).

In each case, most of the additions can be added in a self-contained section at the end of the file. Of course, the original versions of the modified equations **Consumer_demands** and

`Com_clear` must be commented out. Also a couple of changes must be made to the original file, as we will see below.

2.9 Overview of the SJHT examples

This section ia a summary of points we would like you to watch as you work through the example TABLO Input files SJHT1.TAB, SJHT2.TAB and SJHT3.TAB below.

In the linearised additions in SJHT1.TAB and SJHT2.TAB

- The levels equations are differentiated.
- What is usually called dX we prefer to call c_X (the **change** in X).
- We are using the notation p_X for the percentage change in X.
- Convert c_X into p_X using c_X = X*p_X/100 where appropriate.
- Use change variables c_X instead of percentage changes when the associated levels coefficient X may be zero.
- When linearising equations you need to consider whether the variable is a percentage change or an ordinary change and differentiate appropriately.
- Some levels equations appear as UPDATEs instead of EQUATIONs.
- Each levels equation may need to be written as a FORMULA. This formula may be one which is calculated at each step of a multistep calculation (see section 2.10.2 below) or may be one which is used to calculate an initial levels solution for the model. In the second case, the linearised version of the levels equation is also needed; it is required so that the software can calculate the percentage changes or changes in the variables in the equation.

In the levels addition in SJHT3.TAB,

- Write the equations in levels form.
- Link linear percentage change or ordinary change variables in the old section of the TAB file with the new levels variables. Link old coefficients from the TAB file to the new levels variables.
- Make sure all entities in the levels equations are levels variables.
- Use FORMULA & EQUATION statements where possible instead of just EQUATION statements to calculate the initial solution.
- Order the FORMULA & EQUATION statements to build up the initial solution from the data values read in from the data files.
- It is easy to change between change and percentage change variables when using levels equations because TABLO differentiates with the correct variable p_X or c_X. [If you add or delete the qualifier "(Change)" in the declaration of a levels variable, this changes the associated linear variable between being a change and a percentage-change variable. Contrast this with the linearised case where you must recalculate the linearisation if a variable changes from change to percentage change, or *vice versa*.]

2.9.1 Working with the Example Files Provided

You will work through a detailed discussions of the alternative TABLO Input files in the next 3 sections. As you do this, you may find it helpful to have these TABLO files on your computer. Section 4 tells you how to do this.

2.10 Linearised TABLO Input file SJHT1.TAB

Here we discuss the additions and changes that must be made to SJLN.TAB to produce a TABLO Input file SJHT1.TAB in which the new equations are written down in linearised form. In this case, we work with percentage-change linearised variables. Since a percentage change cannot be defined if a variable starts from value 0, this file SJHT1.TAB will not handle zero taxes. [Later we will build SJHT2.TAB which can handle zero taxes.]

2.10.1 The New Linear Variables

These are defined in SJHT1.TAB as shown below. They are all percentage-change variables except for c_DVHOUSTAX.[4]

```
Variable p_GOVINC    # total government income # ;
Variable p_GOVEXP    # total government expenditure # ;
Variable p_GOVEXP_ALT
# Alternative version of gov. expenditure – should equal p_GOVEXP # ;
Variable (ORIG_LEVEL=DVGOV) (All,i,SECT) p_XGOV(i)
   # amount of government consumption of each commodity # ;
Variable (ORIG_LEVEL=THOUS) (All,i,SECT) p_PHOUS(i)
   # price (including tax) paid by households for each commodity # ;
Variable (ORIG_LEVEL=THOUS) (All,i,SECT) p_THOUS(i)
   # power of tax paid by households for each commodity # ;
Variable(Change,ORIG_LEVEL=DVHOUSTAX)(All,i,SECT) c_DVHOUSTAX(i)
   # Dollar value of household tax by commodity # ;
```

2.10.2 Extra Coefficients and Formula

We have seen two new coefficients DVGOV(i) and DVHOUSTAX(i) which are read in from the data files.[5]

```
Coefficient (GE 0) (All,i,SECT) DVGOV(i)
   # value of government expenditure on each commodity # ;
Coefficient (GE 0) (All,i,SECT) DVHOUSTAX(i)
  # value of tax on household consumption of each commodity # ;
```

In addition new coefficients GOVINC, GOVEXP and THOUS(i) are introduced and calculated using formulas derived from the levels equations.

```
Coefficient GOVINC   # total government income # ;
Coefficient GOVEXP   # total government expenditure # ;
Coefficient (GE 0) (All,i,SECT) THOUS(i)
   # power of tax paid by households for each commodity # ;

Formula   GOVINC = SUM(i,SECT, DVHOUSTAX(i) ) ;
Formula   GOVEXP = SUM(i,SECT, DVGOV(i) ) ;
Formula   (All,i,SECT) THOUS(i) = 1 + {DVHOUSTAX(i)/DVHOUS(i)} ;
```

[4] The "ORIG_LEVEL=DVGOV" in the declaration of p_XGOV(i) tells the software the pre-simulation levels value of the quantity of government use of each commodity. This assumes a volume measurement in which one unit of volume is the amount which can be purchased for one dollar in the base period. See chapter 2 of GEMPACK document GPD-4 for more details.

[5] The qualifier "(GE 0)" tells the software that these quantities must never be negative. See section 2.2 of GEMPACK document GPD-4 for more details.

These formulas are recalculated at each step of a multistep calculation so that the values of GOVEXP etc reflect the data base as updated at the start of each step of the multi-step calculation. [Note that an Update statement is not the appropriate way of ensuring up-to-date values of GOVEXP etc since these quantities GOVEXP etc are calculated by formulas, not reads. Update statements can only be used to update the values of Coefficients which are read[6].]

Note that, in TABLO Input files, the values of Coefficients are the (current) values of levels variables. [This is one definition of the term "Coefficient" as used in GEMPACK TABLO Input files.]

The formulas above are needed because (as you will see in the next section) the values of GOVEXP, GOVINC and THOUS are needed in the linearised versions of one or more equations.

2.10.3 The New Equations

As indicated above, the first new equation (in levels form) is:

```
(All,i,SECT)   DVGOV(i) = PC(i) * XGOV(i) ;
```

In a linearised TABLO Input file, this equation appears not as equations but rather as an **Update** since it is needed to tell how the software updates the values of the data items which are read initially. The values DVGOV(i) must be updated after each step of a multi-step simulation. In SJHT1.TAB this equation appears as

```
Update (All,i,SECT)   DVGOV(i) = p_PC(i) * p_XGOV(i) ;
```

The remaining equations (in levels form) can be linearised using standard techniques, as we describe below in detail.

```
Equation(Levels) E_PHOUS
(All,i,SECT)   PHOUS(i) = PC(i) * THOUS(i) ;
```

The linearised version of the equation E_PHOUS in SJHT1.TAB is

```
Equation(Linear) E_PHOUS
      # Relates commodity prices and taxes #
  (All,i,SECT)  p_PHOUS(i) = p_PC(i) + p_THOUS(i) ;
```

[This linearisation uses the well-known rule that the linearised version of the percentage change in a product is just the sum of the percentage changes.]

```
Equation(Levels) E_GOVEXP
GOVEXP = GOVINC ;
```

The linearisation of the equation E_GOVEXP above is, of course,

```
Equation(Linear) E_GOVEXP
  # Government expenditure equals government income #
   p_GOVEXP = p_GOVINC ;
```

```
Equation(Levels) E_XGOV
```

[6] Update statements can also be used to update Coefficients whose initial values are set via a Formula(Initial).

```
 (All,i,SECT)   XGOV(i) = AGOV(i) * GOVEXP / PC(i) ;
```

The linearisation of the equation E_XGOV above is

```
Equation(Linear) E_XGOV  # Government usage of commodities #
  (All,i,SECT)    p_XGOV(i) = p_GOVEXP - p_PC(i) ;
```

This linearisation uses the following well-known rules.

(i) The percentage change in a constant times a variable is just equal to the percentage change in the variable. [Here the constant is AGOV(i) and the variable part is GOVEXP/PC(i). This is why the constant AGOV(i) does not appear in the linearised equation in SJHT1.TAB.]

(ii) The percentage change in a ratio is just the difference of the percentage changes. [Here the ratio is GOVEXP/PC(i).]

The linearisation of E_DVHOUSTAX is fairly complicated. We work through the details below.[7]

```
Equation(Levels) E_DVHOUSTAX
 (All,i,SECT)   DVHOUSTAX(i) = [THOUS(i) – 1] * DVHOUS(i) ;
```

By the product rule for differentiation,[8]

```
c_DVHOUSTAX(i)=(THOUS(i)-1)*c_DVHOUS(i) + DVHOUS(i)*c_THOUS(i)
```

Changing to percentage changes on the right hand side,[9]

```
c_DVHOUSTAX(i)=(THOUS(i)-1)*DVHOUS(i)*p_DVHOUS(i)/100
                + DVHOUS(i)*THOUS(i)*p_THOUS(i)/100
```

From the levels equation above, replace (THOUS(i)-1)*DVHOUS(i)*THOUS(i) by DVHOUSTAX(i). Use the usual rule for the percentage change in a product to replace p_DVHOUS(i) by (p_PC(i)+p_XH(i)) since DVHOUS(i)=PC(i)*XH(i) is one of the levels equations underlying standard Stylized Johansen. Then multiply both sides by 100, which gives the linearised equation

```
Equation(Linear) E_DVHOUSTAX
100*c_DVHOUSTAX(i)= DVHOUSTAX(i)*(p_PC(i) + p_XH(i)) +
                + (DVHOUS(i) + DVHOUSTAX(i))* p_THOUS(i)
```

```
Equation(Levels) E_GOVINC
 GOVINC = SUM(i,SECT, DVHOUSTAX(i)) ;
```

Begin by differentiating both sides. This gives

```
  c_GOVINC = SUM(i,SECT, c_DVHOUSTAX(i) )
```

Substitute for c_GOVINC,

```
  c_GOVINC = GOVINC*p_GOVINC/100
```

[7] Details of a similar differentiation and derivation can be found in equation (28) in Excerpt 28 on page 108 in Horridge, Parmenter and Pearson (1993).

[8] This is the differentiation rule which says that d(X*Y) = X*dY + Y*dX. Or, using "c_" to denote changes, as we prefer to, c_(XY) = X*c_Y + Y*c_X.

[9] This is the usual formula that c_X = X*p_X/100.

where **p_** is used to denote the percentage change in a quantity. This completes the linearisation of the left-hand side.

Since we have introduced the variable c_DVHOUSTAX(i), and calculated it in the previous equation, it is a simple matter to complete the linearisation of the GOVINC levels equation. This is written in SJHT1.TAB as

```
Equation(Linear) E_GOVINC
# Government spending matches tax revenue #
   GOVINC*p_GOVINC = SUM(i,SECT, 100*c_DVHOUSTAX(i) ) ;
```

We also use c_HOUSTAX(i) to update the values of DVHOUSTAX(i).[10]

```
Update (Change) (All,i,SECT)
DVHOUSTAX(i) = c_DVHOUSTAX(i) ;
```

2.10.4 The Modified Equations and Formula

The modified equations are those called **Consumer_demands** and **Com_clear**. The modified version of the first of these is simple: just replace p_PC(i) in it by p_HOUS(i).

```
Equation Consumer_demands (All,I,SECT)
      P_XH(i) = p_Y - p_HOUS(i) ;
```

The levels version of **Com_clear** is

```
Equation(Levels) Com_clear (All,i,SECT)
     XCOM(i) = SUM(j,SECT, XC(i,j)) + XH(i) + XGOV(i) ;
```

The linearised version in SJLN.TAB is

```
EQUATION Com_clear # Commodity market clearing #
   (all,i,SECT) p_XCOM(i) =
       BHOUS(i)*p_XH(i) +  SUM(j,SECT,BCOM(i,j)*p_XC(i,j))
! This was (E3.2.4) in DPPW !  ;
```

The linearised version in SJHT1.TAB is as follows. The derivation of this equation is discussed in the paragraph after the equation.

```
Equation(Linear) Com_clear # Commodity market clearing #
(all,i,SECT) DVCOM(i)*p_XCOM(i) =
       SUM(j,SECT,DVCOMIN(i,j)*p_XC(i,j)) +
         DVHOUS(i)*p_XH(i) + DVGOV(i)*p_XGOV(i)
```

A levels equation A = B + C can be linearised in share form as

p_A = SB * p_B + SC * p_C

where SB and SC are the shares B/A and C/A respectively. It can also be linearised in "value" form

[10] See section 3.5.3 of GEMPACK document GPD-1 for information about the different types of Update statement and their meaning.

A * p_A = B * p_B + C * p_C

which you can think of as saying that 100 times the change in A is equal to 100 times the change in B plus 100 times the change in C. (This is because p_A = 100*c_A/A where c_A is the change in A.) The share form of linearisation is used in the original SJLN.TAB but we now prefer the "values" form – for example the "values" form is used in the ORANI-F paper referred to above. The version of Com_clear in SJHT1.TAB is the "values" version. The derivation is as follows.

First differentiate the levels equation Com_clear to give

c_XCOM(i) = SUM(j,SECT, c_XC(i,j)) + c_XH(i) + c_XGOV(i)

Then replace each c_X term by X*p_X/100 and multiply both sides by 100 to give

XCOM(i)*p_XCOM(i) =
SUM(j,SECT, XC(i,j)*p_XC(i,j)) + XH(i)*p_XH(i) + XGOV(i)*p_XGOV(i)

Finally multiply both sides by the common base price PC(i) which gives the linearised version of Com_clear shown above. [For example, PC(i)*XCOM(i)=DVCOM(i).] Note that this final step only works because there is a common price PC(i) for each quantity in the equation.

Note that the formula for DVCOM in SJLN.TAB must also be modified to include DVGOV(i). [This illustrates one of the possible traps in adding accounting-related behaviour to a linearised model. When a levels equation must be modified, as must Com_clear here, you may remember it modify the linearised equation but forget to modify the Formula which is really the same levels equation.[11]]

```
FORMULA (All,i,SECT)
DVCOM(i) = SUM(j,SECT,DVCOMIN(i,j)) + DVHOUS(i) + DVGOV(i);
```

2.10.5 The Equation for GOVEXP_ALT

In the levels we have the equation

```
Equation(Levels) E_GOVEXP_ALT
GOVEXP_ALT = SUM(i,SECT, DVGOV(i)) ;
```

First differentiate:

```
C_GOVEXP_ALT = SUM(i,SECT, c_DVGOV(i))

GOVEXP * p_GOVEXP_ALT = SUM(i,SECT,DVGOV(i)*p_DVGOV(i) )
```

The remaining step in the linearisation of the GOVEXP equation is the substitution of the linearised form of the percentage change in DVGOV(i) which is just the sum of the percentage changes p_PC(i)+p_XGOV(i).

```
Equation(Linear) E_GOVEXP_ALT  # Total government expenditure #
  GOVEXP*p_GOVEXP_ALT =
     SUM(i,SECT, DVGOV(i)*{p_PC(i) + p_XGOV(i)} ) ;
```

[11] This possible pitfall would be avoided if the original TABLO Input file were a mixed one in which the accounting identities were written as levels equations.

2.10.6 The Complete TAB file SJHT1.TAB

The complete file SJHT1.TAB is available in the ZIP file associated with this document. You can download it from the GEMPACK web site and study the details. The additions are shown in full in Appendix A.

2.11 Linearised TABLO Input File SJHT2.TAB (handles zero taxes)

In SJHT1.TAB the linearised variables p_XGOV(i), p_GOVINC report percentage changes in quantities which will be zero if there are no taxes in the pre-simulation data base. Since a percentage change in zero is meaningless, the file SJHT1.TAB will not perform adequately when zero taxes are present. Here we show how you can produce a file SJHT2.TAB which can handle zero taxes. Clearly the corresponding variables must be change variables rather than percentage-change variables. We must take this into account when we linearise the equations.

2.11.1 The New Variables

The relevant Change variables are as follows.[12]

```
Variable (Change) c_GOVINC   # total government income # ;
Variable (Change) c_GOVEXP   # total government expenditure # ;
Variable (Change, ORIG_LEVEL=DVGOV) (All,i,SECT) c_XGOV(i)
   # amount of government consumption of each commodity # ;
Variable (Change, ORIG_LEVEL=DVHOUSTAX)
     (All,i,SECT) c_DVHOUSTAX(i)
   # Dollar value of household tax by commodity # ;
```

The other new variables, p_PHOUS(i) and p_THOUS(i), are percentage-change variables. Their declaration in SJHT2.TAB is the same as it is in SJHT1.TAB.

2.11.2 The New Equations

As shown above, the levels versions of these are

```
Equation
E_DVGOV      (All,i,SECT)   DVGOV(i) = PC(i) * XGOV(i) ;
E_DVHOUSTAX  (All,i,SECT)
             DVHOUSTAX(i) = [THOUS(i) – 1] * DVHOUS(i) ;
E_PHOUS      (All,i,SECT)   PHOUS(i) = PC(i) * THOUS(i) ;
E_GOVINC     GOVINC = SUM(i,SECT, DVHOUSTAX(i)) ;
E_GOVEXP     GOVEXP = GOVINC ;
E_XGOV       (All,i,SECT)   XGOV(i) = AGOV(i) * GOVEXP / PC(i) ;
```

The linearisation of the equation `E_GOVINC` is

```
Equation E_GOVEXP
  # Government expenditure equals government income #
   c_GOVEXP = c_GOVINC ;
```

The equation `E_GOVINC` is very easy to linearise to give

```
Equation E_GOVINC  # Government spending matches tax revenue #
   c_GOVINC = SUM(i,SECT, c_DVHOUSTAX(i) ) ;
```

[This just says that the change is the sum of the changes.]

[12] The "ORIG_LEVEL=DVGOV" in the declaration of c_XGOV(i) tells the software the pre-simulation levels value of the quantity of government use of each commodity. This assumes a volume measurement in which one unit of volume is the amount which can be purchased for one dollar in the base period. See chapter 2 of GEMPACK document GPD-4 for more details.

Now consider the following levels equation.

```
Equation(Levels) E_XGOV
(All,i,SECT)   XGOV(i) = AGOV(i) * GOVEXP / PC(i) ;
```

In preparation for linearising it, multiply both sides by PC(i), which gives

```
  PC(i) * XGOV(i) = AGOV(i) * GOVEXP
```

Now differentiate both sides, which gives[13]

```
  PC(i)*c_XGOV(i) + XGOV(i)*c_PC(i) = AGOV(i)*c_GOVEXP
```

Now c_PC(i) = PC(i)*p_PC(i)/100 so that

```
XGOV(i)*c_PC(i) = [XGOV(I)*PC(i)]*p_PC(i)/100
                =  DVGOV(i) * p_PC(i)/100
```

In this TABLO Input file SJHT2.TAB there is no coefficient PC(i) corresponding to the percentage change p_PC(i), so we introduce PC_L(i) as follows:

```
Coefficient (All,i,SECT) PC_L(i)
# Basic prices of the commodities # ;
```

The initial value of PC_L(i) is set to 1 and it is updated using the percentage change p_PC(i) from the original model.

```
Formula (Initial) (All,i,SECT) PC_L(i) = 1 ;
Update (All,i,SECT) PC_L(i) = p_PC(i) ;
```

Hence the linearised version in SJHT2.TAB is as below.

```
Equation E_XGOV  # Government usage of commodities #
  (All,i,SECT)
100*PC_L(i)*c_XGOV(i) + DVGOV(i)*p_PC(i) = 100*AGOV(i)*c_GOVEXP ;
```

This leaves the following three levels equations.

```
(All,i,SECT)   DVGOV(i) = PC(i) * XGOV(i) ;
(All,i,SECT)   DVHOUSTAX(i) = [THOUS(i) – 1] * DVHOUS(i) ;
(All,i,SECT)   PHOUS(i) = PC(i) * THOUS(i) ;
```

The second and third of these are linearised in SJHT2.TAB in the same way as in SJHT1.TAB. The first needs a different treatment from that in SJHT1.TAB since SJHT2.TAB has c_XGOV(i) whereas SJHT1.TAB has p_XGOV(i). The quantity DVGOV(i) is read so this equation becomes an Update statement in SJHT2.TAB (as it is in SJHT1.TAB). To work out the Update formula, we must differentiate the equation to obtain an expression for the change c_DVGOV(i) in DVGOV(i). Differentiating the equation leads to

```
c_DVGOV(i) = PC(i)*c_XGOV(i) + XGOV(i)*c_PC(i)
           = PC(i)*c_XGOV(i) + XGOV(i)*PC(i)*p_PC(i)/100
           = PC(i)*c_XGOV(i) + DVGOV(i)*p_PC(i)/100
```

[13] The left-hand side is an example of d(XY) = X.dY + Y.dX. The right-hand side is an example of d(C.Z) = C.dZ when C is a constant.

Hence we have the following Update in SJHT2.TAB using the coefficient PC_L(i) as in the previous equation instead of PC(i)

```
Update (Change) (All,i,SECT)
  DVGOV(i) = PC_L(i) * c_XGOV(i) + DVGOV(i) * p_PC(i) / 100 ;
```

[The right-hand side of an Update(Change) must an expression for the change in the quantity being updated.]

2.11.3 The Modified Equations and Formula

The equation **Consumer_demands** is modified in the same way as in the file SJHT1.TAB.

```
Equation Consumer_demands # Household expenditure functions #
  ! (original) (all,i,SECT) p_XH(i) =p_Y - p_PC(i) !
  (all,i,SECT) p_XH(i) =p_Y - p_PHOUS(i);
```

In SJHT1.TAB, the linearised version of the equation **Com_clear** was

```
Equation(Linear) Com_clear # Commodity market clearing #
(all,i,SECT) DVCOM(i)*p_XCOM(i) =
       SUM(j,SECT,DVCOMIN(i,j)*p_XC(i,j)) +
         DVHOUS(i)*p_XH(i) + DVGOV(i)*p_XGOV(i)
```

Since the percentage change p_XGOV(i) is now replaced by a change variable c_XGOV(i), the last term in the equation is changed:

```
DVGOV(i) * p_XGOV(i) = PC(i)*XGOV(i)*p_XGOV(i)
                     = PC(i) * (100*c_XGOV(i))
                     = PC_L(i) * (100*c_XGOV(i))
```

using the value of PC_L(i) defined earlier, so the equation **Com_clear** becomes

```
 Equation Com_clear # Commodity market clearing #
(all,i,SECT) DVCOM(i)*p_XCOM(i) =
       SUM(j,SECT,DVCOMIN(i,j)*p_XC(i,j)) +
         DVHOUS(i)*p_XH(i) + 100*PC_L(i)*c_XGOV(i) ;
```

As in SJHT1.TAB, the formula for DVCOM(i) in the original part of the TAB file must be modified to have DVGOV(i) added on the right-hand side.

2.11.4 The Equation for GOVEXP_ALT

Differentiate the levels equation for GOVEXP_ALT:

```
Equation(Levels) E_GOVEXP_ALT
GOVEXP_ALT = SUM(i,SECT, DVGOV(i)) ;
```

This gives

```
c_GOVEXP_ALT = SUM(i,SECT,c_DVGOV(i))
        = SUM(i,SECT, (PC_L(i)*c_XGOV(i)+DVGOV(i)*p_PC(i)/100))
```

when we substitute in the value of c_DVGOV(i) derived for the Update statement for DVGOV(i) earlier. Hence:

```
Equation E_GOVEXP_ALT
  # Alternative calculation of government expenditure #
  c_GOVEXP_ALT =
     SUM(i,SECT, PC_L(i) * c_XGOV(i) + {DVGOV(i)/100} * p_PC(i) ) ;
```

2.11.5 The Complete TAB file SJHT2.TAB

The complete file SJHT2.TAB is available in the ZIP file associated with this document. You can download it from the GEMPACK web site and study the details. The additions are shown in full in Appendix B.

2.12 TABLO Input file SJHT3.TAB with New Equations in Levels

Here we discuss the additions and changes that must be made to SJLN.TAB to produce a TABLO Input file SJHT3.TAB in which the new equations are written down in levels form where this seems most natural.

2.12.1 Default Statements

Most of the variables and equations we will add are levels ones. Most of the Formulas will be Initial formulas. Hence the additions begin with the following default statements.

```
Variable (Default=Levels) ;
Equation (Default=Levels) ;
Formula (Default=Initial) ;
```

In the code which follows, if we say just "Equation" that will mean a levels equation. If we want a Linearised equation, we must put "Equation (Linear)" in the code. Similarly for variables.

2.12.2 Levels Variables

The new levels variables are shown below. [Note that, in the linearised verson SJHT1.TAB, most of these are declared as Coefficients.]

```
Variable (Change) GOVINC    # total government income # ;
Variable (Change) GOVEXP    # total government expenditure # ;
Variable (Change) GOVEXP_ALT
# Alternative version of gov expenditure – should equal GOVEXP # ;
Variable (Change) (All,i,SECT) XGOV(i)
# amount of government consumption of each commodity # ;
Variable (All,i,SECT) PHOUS(i)
   # price (including tax) paid by households for each commodity # ;
Variable  (All,i,SECT) THOUS(i)
   # power of tax paid by households for each commodity # ;

! The declaration and reading of DVGOV is added in
    the original part of the file !
Variable (Change, GE 0) (All,i,SECT) DVGOV(i)
# value of government expenditure on each commodity # ;
Variable (Change, GE 0) (All,i,SECT) DVHOUSTAX(i)
   # value of tax on household consumption of each commodity # ;
! Levels versions of variables in the original model !
Variable (All,i,SECT) PC_L(i)
   # price (excluding tax) for each commodity # ;
Variable (All,i,SECT) DVHOUS_L(i)
   # Basic value of each commodity used by households # ;
```

Several of the variables above have been declared as Change variables because we want the model to include the possibility that there are zero taxes shown in the pre-simulation data base. [If you wanted to start from a data base containing nonzero taxes and expected the taxes never to be zero in any updated data base, you might prefer to leave these as percentage-change variables.] There is no problem with having the prices or powers of the tax as percentage-change variables since these are never zero in the levels, even when no taxes are present.

After the declaration of these levels variables, we are free to refer to the associated Linear variables such as **c_DVHOUSTAX(i)** and **p_THOUS(i)**. [c_DVHOUSTAX(i) because

DVHOUSTAX is a Change variable and p_THOUS(i) since THOUS is a percentage-change variable.] We are also able to treat the quantities so declared (for example, **DVHOUSTAX(i)** and **THOUS(i)**) as Coefficients. Thus for example, we can Read the DVHOUSTAX values. See section 2.2 of GEMPACK document GPD-2 for more details.]

As with SJHT1.TAB, the new levels variable DVGOV(i), government expenditure on the commodities, must be declared and read before DVCOM(i) is calculated. The relevant statements (which are inserted near the "Reads" section of the original file) are as follows.

```
Variable (Levels, Change, GE 0) (All,i,SECT) DVGOV(i)
   # value of government expenditure on each commodity # ;

! Additional read of government consumption !
 READ DVGOV  FROM FILE iodata HEADER "GCON" ;
```

Here the qualifier "Levels" must be included since, in the original part of the file, variables are taken to be Linearised ones if no qualifier is present. As in SJHT1.TAB, the formula for DVCOM(i) which follows soon after is modified to include DVGOV(i) on the right-hand side.

2.12.3 Connections with the Original Model

We want to add the new equations as levels equations. Some of these equations involve variables or coefficients (**PC(i)**, the basic commodity prices and **DVHOUS(i)**, the value of commodities used by households) which have been declared as linear variables or coefficients in the original model. To include these in levels equations, we need also to introduce levels versions of these variables, and we need to connect the new levels variables to the original linearised variables in the original part of the file. We attach "**_L**" to the names of these levels variables. [If we tried to introduce a levels variable called simply **PC(i)**, as seems most natural, TABLO will automatically try to declare p_PC(i) as a linearised variable, and TABLO will then object because p_PC(i) has been declared earlier.]

```
! Levels versions of variables in the original model !

Variable (All,i,SECT) PC_L(i)
   # price (excluding tax) for each commodity # ;
Variable (All,i,SECT) DVHOUS_L(i)
   # Basic value of each commodity used by households # ;
```

The initial values of PC_L(i) and DVHOUS_L(i) are set using Formula(Initial).

```
Formula (All,i,SECT) PC_L(i) = 1 ;
Formula (All,i,SECT) DVHOUS_L(i) = DVHOUS(i) ;

! Connection between levels versions here and the linearised
    variables in the original model !
Equation (Linear) E_PC_L   (All,i,SECT) p_PC_L(i) = p_PC(i) ;
Equation (Linear) E_DVHOUS_L
  (All,i,SECT) p_DVHOUS_L(i) = p_PC(i) + p_XH(i) ;
```

This ensures that PC_L(i) and DVHOUS_L(i) are updated using the percentage change variables p_PC(i) and p_XH(i) from the original model. (Levels variables are automatically updated using the corresponding change or percentage change. For example, PC_L(i) is updated by p_PC_L(i)).

The above code is rather unnatural and artificial. It is necessary because we are grafting on levels equations to a file SJLN.TAB which contains only linearised equations.[14]

2.12.4 Setting Initial Values of Levels Variables

When levels variables are declared, it is necessary to give a sequence of reads and formulas to set their initial (that is, pre-simulaion) values. This is because, when you write down a levels equation, TABLO automatically linearises it; the associated linearisation usually involves the associated Coefficient (that is, levels value) as well as the associated linearised variables (changes or percentage changes).

We have seen above how some of the new levels variables have their initial values set by the following statements in SJHT3.TAB.

```
! Read of DVGOV has been added in the original part of the file !
 READ DVHOUSTAX FROM FILE iodata HEADER "HTAX" ;

! Basic prices equal 1 in the pre-sim model solution !
Formula (All,i,SECT) PC_L(i) = 1 ;
Formula (All,i,SECT) DVHOUS_L(i) = DVHOUS(i) ;
```

The initial values of the other new variables are set in Formula & Equation statements, as described in the section below.

2.12.5 The New Behavioural Equations

As indicated earlier, the new levels equations are as follows.

```
Equation
E_DVGOV      (All,i,SECT)     DVGOV(i) = PC(i) * XGOV(i) ;
E_DVHOUSTAX  (All,i,SECT) DVHOUSTAX(i) = [THOUS(i) – 1] * DVHOUS(i) ;
E_PHOUS      (All,i,SECT)   PHOUS(i) = PC(i) * THOUS(i) ;
E_GOVINC     GOVINC = SUM(i,SECT, DVHOUSTAX(i) ) ;
E_GOVEXP     GOVEXP = GOVINC ;
E_XGOV       (All,i,SECT)   XGOV(i) = AGOV(i) * GOVEXP / PC(i) ;
```

These are put directly into the TAB file as shown below. Care has to be taken about the order in which they are written because (as is common when writing down levels equations), these equations double as Formulas as well as equations. This is why you see the keyword "**Formula & Equation**" before each of them. The order of the sequence of Formulas so introduced is important. The right-hand side of any one can only contain quantities whose pre-simulation values have already been set (in earlier Reads or Formulas). This is why we needed the formulas above to set pre-simulation levels values for PC_L(i) and XH_L(i). Note that the first two equations have been rewritten to have XGOV and THOUS on their left-hand sides; this is so that the associated formula gives pre-simulation values for these. The equation E_PHOUS (with PHOUS on the left-hand side) must come after E_THOUS since it has THOUS on the right-hand side.

---

[14] Some simplification can be made if the TAB file is to be processed by Release 8.0 (or later) of GEMPACK. Then the declaration of PC_L can be made by the following statement.

```
   Variable (LINEAR_VAR=p_PC) (All,i,SECT) PC_L(i)
       # price (excluding tax) for each commodity # ;
```

Here the qualifier "(LINEAR_VAR=p_PC)" indicates that the linear variable associated with this new levels variable PC_L is simply the variable p_PC already declared. Then there is no need for the equation E_PC_L shown above. [For more details, see the Release 8.0 version of GPD-2.]

```
Formula & Equation E_DVGOV
     # Quantity of government comsumption by commodity #
 (All,i,SECT) XGOV(i) = DVGOV(i) / PC_L(i) ;
Formula & Equation E_DVHOUSTAX    # Powers of household taxes #
 (All,i,SECT) THOUS(i) = 1 + {DVHOUSTAX(i)/DVHOUS_L(i)} ;
Formula & Equation E_PHOUS
# Prices paid by households including taxes #
 (All,i,SECT) PHOUS(i) = PC_L(i) * THOUS(i) ;
Formula & Equation E_GOVINC  # Government income #
   GOVINC = SUM(i,SECT, DVHOUSTAX(i) ) ;
Formula & Equation E_GOVEXP
  # Government expenditure equals government income #
   GOVEXP = GOVINC ;
```

Before the last equation can be inserted, we need the declaration of AGOV(i) and a formula to calculate its values. This equation can be inserted as a levels equation or as a linearised equation (as the coments in the TAB file indicate).

```
ZeroDivide Default 0.5 ;
! If GOVEXP is zero in base data, set AGOV equal to 0.5 for each of
 the 2 commodities. Of course, 0.5 could be replaced by other values.
!
Coefficient (Parameter) (All,i,SECT)  AGOV(i)
   # share of each commodity in total government demand # ;
Formula (All,i,SECT)  AGOV(i) = DVGOV(i)/GOVEXP ;
ZeroDivide Off ;

Equation  E_XGOV   # Volume of government consumption by commodity #
   (All,i,SECT)  XGOV(i) = AGOV(i) * GOVINC / PC_L(i) ;
```

2.12.6 The Modified Equations and Formula

These equations **Consumer_demands** and **Com_clear** are the same as in SJHT2.TAB. As there, the modified equations are added at the end and the original versions are commented out.

As in SJHT1.TAB, the formula for DVCOM(i) in the original part of the TAB file must be modified to have DVGOV(i) added on the right-hand side.

2.12.7 The Equation for GOVEXP_ALT

The declaration of the levels variable GOVEXP_ALT and its associated equation in SJHT3.TAB are as below.

```
Variable (Change) GOVEXP_ALT
  # Alternative version of gov expenditure - should equal GOVEXP # ;
Formula & Equation E_GOVEXP_ALT
  # Alternative calculation of government expenditure #
   GOVEXP_ALT = SUM(i,SECT, DVGOV(i) ) ;
```

2.12.8 The Complete TAB file SJHT3.TAB

The complete file SJHT3.TAB is available in the ZIP file associated with this document. You can download it from the GEMPACK web site and study the details. The additions are shown in full in Appendix C.

3 Checking the Implementation

Whenever you extend a model, it is vital to carry out simulations to check that it is working. We have found (and recommend this as a strategy to you) that it is often useful to check that the reversal of a test simulation also works.

3.1 Data Files

We have supplied 3 versions of an initial data base for model SJHT. These data base files are

- **SJHT0.DAT**. This is identical with SJ.DAT except that zero vectors have been added at the new headers "GCON" and "HTAX".
- **SJHTTINY.DAT**. This contains the same data as in SJHT0.DAT except that a tiny positive number (0.000001) has been added to the values for DVGOV(i) and DVHOUSTAX(i) so they are no longer exactly zero. [This data SJHTTINY.DAT is used later with the file SJHT1.TAB.]
- **SJHT.DAT**. This contains nonzero values for DVGOV(i) and DVHOUSTAX(i) which we have made up to satisfy the underlying levels equations.

We show the data in SJHT0.DAT and SJHT.DAT in table form below. [That for SJHTTINY.DAT is the same as for SJHT0.DAT except that the two vectors of zeros are replaced by vectors containing 0.000001.]

Initial Data in SJHT0.DAT

	Industry 1	Industry 2	Households	Government	Totals
Commodities	4	2	2	**0.0**	8.0
(Basic prices)	2	6	4	**0.0**	12.0
Primary factors					
Labour	1.0	3.0			4.0
Capital	1.0	1.0			2.0
Tax by			**0.0**		0.0
Commodity			**0.0**		0.0
	8.0	12.0	6.0	0.0	

Initial Data in SJHT.DAT

	Industry 1	Industry 2	Households	Government	Totals
Commodities	4	2	2	0.3	8.3
(Basic prices)	2	6	4	0.3	12.3
Primary factors					
Labour	1.2	3.2			4.4
Capital	1.1	1.1			2.2
Tax by			0.2		0.2
Commodity			0.4		0.4
	8.3	12.3	6.6	0.6	

3.2 SJHTCHK.TAB for Checking Initial and Updated Data

The TABLO Input file SJHTCHK.TAB has been described briefly in section 2.7 above.

This TABLO file can be used to check that original data files or updated files produced by simulations are balanced – that is, are levels solutions of the model with a consumption tax. More precisely, this TABLO file checks that

- For each sector, total sales DVCOM(i) equals total costs DVCOSTS(i).
- Total government expenditure equals total tax revenue.

We have supplied three Command files SJHT0CK.CMF, SJHTTINY.CMF and SJHTCHK.CMF which can be used to check that the three sets of data described in the section above (SJHT0.DAT, SJHTTINY.DAT and SJHT.DAT) are balanced.

After running any simulation, you can also use the SJHTCHK.TAB to check the balance of the updated data. Simply copy say SJHS0CHK.CMF and replace the name of the file IODATA by the name of the updated data after the simulation in question, and replace the name of the Display file produced by something suitable.

It is an important test of any of the new TABLO Input files (SJHT1, SJHT2, SJHT3) that they produce balanced updated data (as verified by running SJHTCHK.TAB). When you carry out the test simulations, you should always run SJHTCHK to check the updated data.

3.3 Command files and Simulations

For each model, we supply several test Command files,

- one increasing the powers of the taxes by 10%, starting from the database with zeros
- and a corresponding one reversing this increase,
- one increasing the power of the taxes by 5 percent, starting from the nonzero database.

For TAB file SJHT1.TAB, use the Command files
SJHT1-A.CMF, SJHT1-AR.CMF and SJHT1-B.CMF.
For TAB file SJHT2.TAB, use the Command files
SJHT2-A.CMF, SJHT2-AR.CMF and SJHT2-B.CMF.
For TAB file SJHT3.TAB, use the Command files
SJHT3-A.CMF, SJHT3-AR.CMF and SJHT3-B.CMF.

We discuss these simulations with SJHT3.TAB. [Similar considerations apply when these shocks are applied to SJHT1.TAB or SJHT2.TAB.]

The file SJHT3-A.CMF increases THOUS(i) in each sector by 10 per cent.

```
shock p_thous = uniform 10 ;
```

Since there are no taxes in the pre-simulation data set SJHT0.DAT, the initial levels values of THOUS(i) are 1 for each i. The post-simulation data SJHT3-A.UPD will thus be a data set for the new model in which household taxes are levied on each commodity at the *ad valorem* rate of 10%.

The reversal simulation SJHT3-AR.CMF should reverse the shocks given. This is not a shock of –10% but rather a shock chosen to reverse the change in the powers. Initially each power was 1, then each power is 1.1 after the simulation. Thus a change of –0.1 from 1.1 back to 1.0 is required. This is, in fact, a shock of –9.090909 per cent, which is the shock introduced in Command file SJHT3-AR.CMF (R for “reversal”).

```
shock p_thous = uniform -9.09090909 ;
```

The final simulation SJHT3-B.CMF uses the nonzero database SJHT.DAT and applies a five percent increase to the power of the tax. This is a normal simulation where there are no difficulties with zero values to worry about.

3.3.1 Checking the Reversal Simulations

We discuss this in the context of the simulation in SJHT3-A.CMF and the reversal of this simulation in SJHT3-AR.CMF.

The simulation results in the reversal simulation should be the reverse of those in the original simulation. The data base updated after the reversal simulation should be the same as the starting data base before the original simulation. We describe how you can check both of these for this simulation.

Results

The c_GOVINC result for the original simulation is 0.545218 while that for the reversal simulation is -0.545219. These are changes in government income (measured in dollars). Clearly these are the reverse of each other.

The p_XH results for the first sector are -9.13032 in the original simulation and +10.0477 in the reversal simulation. Because these are percentage changes, it is not obvious whether the latter is the reversal of the former. To check this, it is legitimate to assume that the pre-simulation levels value (before the initial simulation) is 100. Then the levels value after the original simulation will be 9.13032 percent less, that is, 90.86968. To return this to 100, there must be an increase of 9.13032. As a percentage change, this is
100*(9.13032/90.86968) = 10.0477
Hence 10.0477 is indeed the reversal of –9.13032.

You can check other change and percentage change variables in the same way.

Data

The GEMPACK program CMPHAR can be used to compare the original data base SJHT0.DAT with that updated after the reversal simulation, namely SJHT3-AR.UPD. If you run CMPHAR to compare these two data files, you will see that the largest absolute differences is very small (less than 0.000002). This means that the two data files are identical to machine accuracy.

You should check the other reversal simulations (for example, those with Command files SJHT1-A.CMF and SJHT1-AR.CMF) in the same way. Check both simulation results and data.

3.3.2 Adding Tiny Values to the Data Base Is Surprisingly Successful

The starting data base for the SJHT1-A.CMF simulation (which increases the *ad valorem* tax on each commodity from 0 to 10) has very small values equal to 0.000001 for taxes and value of government consumption by commodity. When the *ad valorem* tax rates are increased, the total amount of tax collected increases from the pre-simulation value of 0.000002 to a post-simulation value of approximately 5.45. The simulation results say that the percentage increase in government consumption of each commodity [the p_XGOV(i) results] are approximately 27.2 million.

Should we have any faith in a simulation which reports such large percentage increases? Well, we think that the answer is “yes”. One way of checking this is to check the other

simulation results from SJHT1-A.CMF against those from SJHT3-A.CMF. You will find that the results for key variables such as p_XCOM(i) are in very good agreement. Another way of checking this is to compare the updated data after these two simulations (using CMPHAR). Again you will find good agreement.

Note, however, that the choice of values in the original DVCOM matrix does determine the post-simulation split of government expenditure between the two commodities. This is because of the value of the parameters AGOV(i) in the equation E_XGOV. If we had put values of 0.000001 (commodity 1) and 0.000002 (commodity 2) in the data base for government expenditure on the two commodities, the simulation results would be different since then the post-simulation data would then have these in the ratio 1:2.

3.3.3 Could This Checking Detect Errors?

The checks described above (running SJHTCHK.TAB on the updated data, and checking that the reversal simulation worked) are aimed at detecting errors in the new TABLO Input files. Would these tests in fact detect errors?

One way of gaining information is to make a deliberate error and repeat the checks. To try this, copy file SJHT3.TAB to **SJHT3E.TAB** ("E" for error) and then edit SJHT3E.TAB to leave out the term "**+ DVCOM(i)**" from the formula which has DVCOM(i) on the left-hand side. [Recall from the discussion near the end of section 2.10.4 that this is a likely error.]

Then copy SJHT3-A.CMF to **SJHT3EA.CMF** and edit this to change the Auxiliary files statement to

Auxiliary files = sjht3e ;

Then run the simulation in SJHT3EA.CMF. When it has finished, run SJHTCHK.TAB as recommended above. You will find that the DVCOSTS results in the Display file produced are

8.142574 11.884300 20.026875

while the DVCOSTS results are

8.145052 11.882955 20.028008

[The first two numbers are for the 2 sectors and the third number is the total of the other two.] Note that there are discrepancies of about 0.002 in each of the sectors. This is significant since it is in the 4th figure of the value for the first sector and in the 5th figure of the value for the second sector.

If you think that the above discrepancy is too small to show the error, repeat the above but this time increase the shock to THOUS from 10 to 40. If you do that, you will find that the discrepancies are even more obvious. The DVCOM(i) values are

8.453129 11.628621 20.081749

while the DVCOSTS(i) values are

8.478847 11.614412 20.093258

As you can see the discrepancies are larger – they are now in the 3rd and 4th figures.

3.3.4 Problem with SJHT1-X.CMF

The Command file SJHT1-X.CMF is used with the first model SJHT1.TAB and illustrates the problems caused by zeros. The left-hand side matrix is singular so it is impossible to carry out a simulation with SJHT1.TAB starting from zero taxes and government expenditure. [But it is possible to use SJHT1.TAB to do the same simulation starting from a the data base SJHTTINY.DAT, as discussed above in section 3.3.2.]

4 Using The Example Files Provided

All the example files are contained in the ZIP file SJHT-EX.ZIP which can be downloaded from the GEMPACK web site at address

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

This ZIP file contains

- the TABLO Input files SJHT1.TAB, SJHT2.TAB and SJHT3.TAB,
- the starting data files described in section 3.1 above, and
- the Command files described in sections 3.2 and 3.3 above.

To use these files, make a new directory (for example, C:\SJHT), put the ZIP file

SJHT-EX.ZIP

into that directory and issue the command (at a DOS prompt)

pkunzip sjht-ex

Then you can run TABLO to process any of the TAB files and can use the associated TABLO-generated program (or GEMSIM) to carry out simulations or to run SJHTCHK.

5 Appendix A - Additions for SJHT1.TAB

Below are the additions and changes made for SJHT1.TAB.

Most of the statements are added at the end of the standard SJLN.TAB. The original versions of the modified equations **Consumer_demands** and **Com_clear** must be commented out in the SJLN.TAB part.

Also the new DVGOV data must be read earlier in the file (because it is needed in calculating the modified DVCOM values).

5.1 Changes to the Original TAB File

It is necessary to read the DVGOV data as it is needed for calculating the modified DVCOM values (which need to take into account government purchases). These lines of code are inserted just before and just after the "Reads" section in SJLN.TAB. The section of the TABLO Input file containing the new and modified code is shown below. [Some of that shown below is also in SJLN.TAB. The new and modified parts are shown in bold]

```
! Additional declaration for reading government consumption !

Coefficient (GE 0) (All,i,SECT) DVGOV(i)
   # value of government expenditure on each commodity # ;

!-----------------------------------------------------------------------!
!  Reads from the data base                                             !
!-----------------------------------------------------------------------!

 READ DVCOMIN FROM FILE iodata HEADER "CINP" ;
 READ DVFACIN FROM FILE iodata HEADER "FINP" ;
 READ DVHOUS  FROM FILE iodata HEADER "HCON" ;
! Additional read of government consumption !
 READ DVGOV  FROM FILE iodata HEADER "GCON" ;

!-----------------------------------------------------------------------!
!  Other coefficients and formulas for them                             !
!-----------------------------------------------------------------------!

 COEFFICIENT  Y    # Total nominal household expenditure # ;
 FORMULA Y = SUM(i,SECT,DVHOUS(i)) ;

 COEFFICIENT (all,i,SECT) DVCOM(i)
  ! Dollar value of total demand for commodity i ! ;
 FORMULA   (all,i,SECT)
! Change for the household tax/government extension. Must
     add in DVGOV here. !
! (original)    DVCOM(i) = SUM(j,SECT, DVCOMIN(i,j)) + DVHOUS(i) ; !
    DVCOM(i) = SUM(j,SECT, DVCOMIN(i,j)) + DVHOUS(i) + DVGOV(i) ;
```

5.2 Main Additions

These are attached at the end of SJLN.TAB.

```
! Additions for household consumption tax and government purchases
!

Variable p_GOVINC
  # total government income # ;
Variable p_GOVEXP
  # total government expenditure # ;
```

```
Variable  p_GOVEXP_ALT
  # Alternative version of gov expenditure - should equal p_GOVEXP #
;
Variable (ORIG_LEVEL=DVGOV) (All,i,SECT) p_XGOV(i)
   # amount of government consumption of each commodity # ;
Variable (ORIG_LEVEL=THOUS) (All,i,SECT) p_PHOUS(i)
   # price (including tax) paid by households for each commodity # ;
Variable (ORIG_LEVEL=THOUS) (All,i,SECT) p_THOUS(i)
   # power of tax paid by households for each commodity # ;

! Next is introduced to simplify update of DVHOUSTAX and to
   simplify equation E_GOVINC !
Variable (Change, ORIG_LEVEL=DVHOUSTAX) (All,i,SECT) c_DVHOUSTAX(i)
   # Dollar value of household tax by commodity # ;

! Update statement for DVGOV which is read earlier !
Update  (All,i,SECT)
  DVGOV(i) = p_PC(i) * p_XGOV(i) ;

Coefficient GOVINC   # total government income # ;
Coefficient GOVEXP   # total government expenditure # ;

! The declaration and reading of DVGOV has been added in
    the original part of the file !

Coefficient (GE 0) (All,i,SECT) DVHOUSTAX(i)
   # value of tax on household consumption of each commodity # ;
Update (Change) (All,i,SECT)
    DVHOUSTAX(i) = c_DVHOUSTAX(i) ;

Coefficient (GE 0) (All,i,SECT) THOUS(i)
   # power of tax paid by households for each commodity # ;


! New DVGOV data is read in the original part of the file !
 READ DVHOUSTAX FROM FILE iodata HEADER "HTAX" ;

Formula GOVINC = SUM(i,SECT, DVHOUSTAX(i) ) ;
Formula GOVEXP = SUM(i,SECT, DVGOV(i) ) ;
Formula (All,i,SECT) THOUS(i) = 1 + {DVHOUSTAX(i)/DVHOUS(i)} ;


! Equations modified from the original model !
! Must comment the original versions of these out above !

 EQUATION Consumer_demands # Household expenditure functions #
  ! (original) (all,i,SECT) p_XH(i) =p_Y - p_PC(i) !
  (all,i,SECT) p_XH(i) =p_Y - p_PHOUS(i)         ;


 EQUATION Com_clear # Commodity market clearing #
   ! (original) (all,i,SECT) p_XCOM(i) =
       BHOUS(i)*p_XH(i) +  SUM(j,SECT,BCOM(i,j)*p_XC(i,j))  !
   (all,i,SECT) DVCOM(i)*p_XCOM(i) =
       SUM(j,SECT,DVCOMIN(i,j)*p_XC(i,j)) +
         DVHOUS(i)*p_XH(i) + DVGOV(i)*p_XGOV(i) ;


! New equations !
Equation E_PHOUS # Relates commodity prices and taxes #
  (All,i,SECT)  p_PHOUS(i) = p_PC(i) + p_THOUS(i) ;
```

```
Equation E_DVHOUSTAX  # Amount of household tax by commodity #
   (All,i,SECT)
    100*c_DVHOUSTAX(i) = DVHOUSTAX(i)*{p_PC(i) + p_XH(i)} +
         {DVHOUS(i)+DVHOUSTAX(i)}*p_THOUS(i) ;

Equation E_GOVINC  # Government income matches tax revenue #
   GOVINC*p_GOVINC = SUM(i,SECT, 100*c_DVHOUSTAX(i) ) ;

Equation E_GOVEXP
  # Government expenditure equals government income #
   p_GOVEXP = p_GOVINC ;

Equation E_XGOV  # Government usage of commodities #
  (All,i,SECT)
    p_XGOV(i) = p_GOVEXP - p_PC(i) ;

Equation E_GOVEXP_ALT  # Total government expenditure #
  GOVEXP*p_GOVEXP_ALT =
     SUM(i,SECT, DVGOV(i)*{p_PC(i) + p_XGOV(i)} ) ;
```

6 Appendix B - Additions for SJHT2.TAB

Below are the additions and changes made for SJHT2.TAB.

Most of the statements are added at the end of the standard SJLN.TAB. The original versions of the modified equations **Consumer_demands** and **Com_clear** must be commented out in the SJLN.TAB part.

Also the new DVGOV data must be read earlier in the file (because it is needed in calculating the modified DVCOM values).

6.1 Changes to the original TAB file

It is necessary to read the DVGOV data as it is needed for calculating the modified DVCOM values (which need to take into account government purchases). These lines of code are inserted just before and just after the "Reads" section in SJLN.TAB.

The changes here are exactly the same as those for SJHT1.TAB (see Appendix A).

6.2 The Main Additions

These are attached at the end of SJLN.TAB.

```
! Additions for household consumption tax and government purchases !
Variable (Change) c_GOVINC
  # total government income # ;
Variable (Change) c_GOVEXP
  # total government expenditure # ;
Variable (Change) c_GOVEXP_ALT
# Alternative version of gov expenditure - should equal c_GOVEXP # ;
Variable (Change, ORIG_LEVEL=DVGOV) (All,i,SECT) c_XGOV(i)
   # amount of government consumption of each commodity # ;
Variable (ORIG_LEVEL=THOUS) (All,i,SECT) p_PHOUS(i)
   # price (including tax) paid by households for each commodity # ;
Variable (ORIG_LEVEL=THOUS) (All,i,SECT) p_THOUS(i)
   # power of tax paid by households for each commodity # ;

! Next is introduced to simplify update of DVHOUSTAX and to
   simplify equation E_GOVBALANCE !
Variable (Change, ORIG_LEVEL=DVHOUSTAX) (All,i,SECT) c_DVHOUSTAX(i)
   # Dollar value of household tax by commodity # ;

! PC_L is needed to update DVGOV !
Coefficient (All,i,SECT) PC_L(i) # Basic prices of the commodities #;
! Basic prices start at 1 !
Formula (Initial) (All,i,SECT) PC_L(i) = 1 ;
Update (All,i,SECT) PC_L(i) = p_PC(i) ;

! Update statement for DVGOV which is read earlier !
Update (Change) (All,i,SECT)
  DVGOV(i) = PC_L(i) * c_XGOV(i) + DVGOV(i) * p_PC(i) / 100 ;

! The declaration and reading of DVGOV has been added in
    the original part of the file !

Coefficient (GE 0) (All,i,SECT) DVHOUSTAX(i)
   # value of tax on household consumption of each commodity # ;
```

```
Update (Change) (All,i,SECT)
    DVHOUSTAX(i) = c_DVHOUSTAX(i) ;

! New DVGOV data is read in the original part of the file !
 READ DVHOUSTAX FROM FILE iodata HEADER "HTAX" ;

Coefficient GOVEXP   # total government expenditure # ;
Coefficient (GE 0) (All,i,SECT) THOUS(i)
   # power of tax paid by households for each commodity # ;

Formula GOVEXP = SUM(i,SECT, DVGOV(i) ) ;
Formula (All,i,SECT) THOUS(i) = 1 + {DVHOUSTAX(i)/DVHOUS(i)} ;

! Equations modified from the original model !
! Must comment the original versions of these out above !

 EQUATION Consumer_demands # Household expenditure functions #
  ! (original) (all,i,SECT) p_XH(i) =p_Y - p_PC(i) !
  (all,i,SECT) p_XH(i) =p_Y - p_PHOUS(i)
   ! This was (E3.2.1) in DPPW !  ;

 EQUATION Com_clear # Commodity market clearing #
   ! (original) (all,i,SECT) p_XCOM(i) =
       BHOUS(i)*p_XH(i) +  SUM(j,SECT,BCOM(i,j)*p_XC(i,j))  !
   (all,i,SECT) DVCOM(i)*p_XCOM(i) =
       SUM(j,SECT,DVCOMIN(i,j)*p_XC(i,j)) +
         DVHOUS(i)*p_XH(i) + 100*PC_L(i)*c_XGOV(i) ;

! New equations !
Equation E_PHOUS # Relates commodity prices and taxes #
  (All,i,SECT)  p_PHOUS(i) = p_PC(i) + p_THOUS(i) ;

Equation E_DVHOUSTAX  # Amount of household tax by commodity #
   (All,i,SECT)
    100*c_DVHOUSTAX(i) = DVHOUSTAX(i)*{p_PC(i) + p_XH(i)} +
         {DVHOUS(i)+DVHOUSTAX(i)}*p_THOUS(i) ;

Equation E_GOVINC  # Government income matches tax revenue #
   c_GOVINC = SUM(i,SECT, c_DVHOUSTAX(i) ) ;

Equation E_GOVEXP
  # Government expenditure equals government income #
   c_GOVEXP = c_GOVINC ;

ZeroDivide Default 0.5 ;
! If GOVEXO is zero in base data, set AGOV equal to 0.5 for each of
 the 2 commodities. Of course, 0.5 could be replaced by other values!
Coefficient (Parameter) (All,i,SECT)  AGOV(i)
  # share of each commodity in total gov demand for commodities # ;
Formula (Initial) (All,i,SECT)  AGOV(i) = DVGOV(i)/GOVEXP ;
ZeroDivide Off ;

Equation E_XGOV  # Government usage of commodities #
  (All,i,SECT)
    100*PC_L(i)*c_XGOV(i) + DVGOV(i)*p_PC(i) = 100*AGOV(i)*c_GOVEXP ;

Equation E_GOVEXP_ALT
  # Alternative calculation of government expenditure #
  c_GOVEXP_ALT =
     SUM(i,SECT, PC_L(i) * c_XGOV(i) + {DVGOV(i)/100} * p_PC(i) ) ;
```

7 Appendix C - Additions for SJHT3.TAB

Below are the additions and changes made for SJHT3.TAB.

Most of the statements are added at the end of the standard SJLN.TAB. The original versions of the modified equations **Consumer_demands** and **Com_clear** must be commented out in the SJLNB.TAB part.

Also the new DVGOV data must be read earlier in the file (because it is needed in calculating the modified DVCOM values).

7.1 Changes to the original TAB file

It is necessary to read the DVGOV data as it is needed for calculating the modified DVCOM values (which need to take into account government purchases). These lines of code are inserted just before and just after the "Reads" section in SJLN.TAB. The section of the TABLO Input file containing the new and modified code is shown below. [Some of that shown below is also in SJLN.TAB. The new and modified parts are shown in bold]

```
! Additional declaration for reading government consumption !

Variable (Levels, Change, GE 0) (All,i,SECT) DVGOV(i)
   # value of government expenditure on each commodity # ;

!-------------------------------------------------------------------!
!  Reads from the data base                                         !
!-------------------------------------------------------------------!

 READ DVCOMIN FROM FILE iodata HEADER "CINP" ;
 READ DVFACIN FROM FILE iodata HEADER "FINP" ;
 READ DVHOUS  FROM FILE iodata HEADER "HCON" ;
! Additional read of government consumption !
 READ DVGOV  FROM FILE iodata HEADER "GCON" ;

!-------------------------------------------------------------------!
!  Other coefficients and formulas for them                         !
!-------------------------------------------------------------------!

 COEFFICIENT  Y    # Total nominal household expenditure # ;
 FORMULA Y = SUM(i,SECT,DVHOUS(i)) ;

 COEFFICIENT (all,i,SECT) DVCOM(i)
  ! Dollar value of total demand for commodity i ! ;
 FORMULA   (all,i,SECT)
! Change for the household tax/government extension. Must
     add in DVGOV here. !
! (original)    DVCOM(i) = SUM(j,SECT, DVCOMIN(i,j)) + DVHOUS(i) ; !
    DVCOM(i) = SUM(j,SECT, DVCOMIN(i,j)) + DVHOUS(i) + DVGOV(i) ;
```

7.2 The Main Additions

```
! Additions for household consumption tax and government purchases !

Variable (Default=Levels) ;
Equation (Default=Levels) ;
Formula (Default=Initial) ;

! New levels variables !
Variable (Change) GOVINC    # total government income # ;
Variable (Change) GOVEXP    # total government expenditure # ;
```

```
Variable (Change) GOVEXP_ALT
  # Alternative version of gov expenditure - should equal GOVEXP # ;
Variable (Change) (All,i,SECT) XGOV(i)
   # amount of government consumption of each commodity # ;
Variable (All,i,SECT) PHOUS(i)
   # price (including tax) paid by households for each commodity # ;
Variable  (All,i,SECT) THOUS(i)
   # power of tax paid by households for each commodity # ;

! The declaration and reading of DVGOV has been added in
    the original part of the file !
Variable (Change, GE 0) (All,i,SECT) DVHOUSTAX(i)
   # value of tax on household consumption of each commodity # ;

! Levels versions of variables in the original model !
Variable (All,i,SECT) PC_L(i)
   # price (excluding tax) for each commodity # ;
      ! Basic prices equal 1 in the pre-sim model soluiton !
Variable (All,i,SECT) DVHOUS_L(i)
   # Basic value of each commodity used by households # ;

! Connection between levels versions here and the linearised
    variables in the original model !
Equation (Linear) E_PC_L   (All,i,SECT) p_PC_L(i) = p_PC(i) ;
Equation (Linear) E_DVHOUS_L
 (All,i,SECT) p_DVHOUS_L(i) = p_PC(i) + p_XH(i) ;

! Read of DVGOV has been added in the original part of the file !
 READ DVHOUSTAX FROM FILE iodata HEADER "HTAX" ;

! Basic prices equal 1 in the pre-sim model solution !
Formula (All,i,SECT) PC_L(i) = 1 ;
Formula (All,i,SECT) DVHOUS_L(i) = DVHOUS(i) ;

Formula & Equation E_DVGOV
     # Dollar values of government comsumption by commodity #
 (All,i,SECT) XGOV(i) = DVGOV(i) / PC_L(i) ;
Formula & Equation E_DVHOUSTAX
     # Dollar values of household taxes by commodity #
 (All,i,SECT) THOUS(i) = 1 + {DVHOUSTAX(i)/DVHOUS_L(i)} ;
Formula & Equation E_PHOUS
     # Prices paid by households including taxes #
 (All,i,SECT) PHOUS(i) = PC_L(i) * THOUS(i) ;

Formula & Equation E_GOVINC # Government income equals tax revenue #
   GOVINC = SUM(i,SECT, DVHOUSTAX(i) ) ;

Formula & Equation E_GOVEXP
  # Government expenditure equals government income #
   GOVEXP = GOVINC ;

ZeroDivide Default 0.5 ;
! If GOVEXP is zero in base data, set AGOV equal to 0.5 for each of
 the 2 commodities. Of course, 0.5 could be replaced by other values.
!
Coefficient (Parameter) (All,i,SECT)  AGOV(i)
  # share of each commodity in total gov demand for commodities # ;
Formula (All,i,SECT)  AGOV(i) = DVGOV(i)/GOVEXP ;
ZeroDivide Off ;

Equation  E_XGOV  # Volume of government consumption by commodity #
```

```
   (All,i,SECT)  XGOV(i) = AGOV(i) * GOVEXP / PC_L(i) ;

Formula & Equation E_GOVEXP_ALT
  # Alternative calculation of government expenditure #
   GOVEXP_ALT = SUM(i,SECT, DVGOV(i) ) ;

! Equations modified from the original model !
! Leave these as linearised equations !
! Must comment the original versions of these out above !

 EQUATION (Linear) Consumer_demands # Household expenditure functions
#
  ! (original) (all,i,SECT) p_XH(i) =p_Y - p_PC(i) !
  (all,i,SECT) p_XH(i) =p_Y - p_PHOUS(i) ;

 EQUATION (Linear) Com_clear # Commodity market clearing #
   ! (original) (all,i,SECT) p_XCOM(i) =
       BHOUS(i)*p_XH(i) +  SUM(j,SECT,BCOM(i,j)*p_XC(i,j))  !
   (all,i,SECT) DVCOM(i)*p_XCOM(i) =
       SUM(j,SECT,DVCOMIN(i,j)*p_XC(i,j)) +
         DVHOUS(i)*p_XH(i) + 100*c_DVGOV(i) ;
```

8 References

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

Harrison, W.J. and K.R. Pearson (1998), *An Introduction to GEMPACK*, GEMPACK Document No. 1 **[GPD-1],** Monash University, Clayton, Fourth edition, October 1998.

Harrison, W.J. and K.R. Pearson (1994), *User's Guide to TABLO, GEMSIM and TABLO-generated Programs*, GEMPACK Document No. 2 **[GPD-2],** Monash University, Clayton, Second edition, April 1994.

Harrison, W.J. and K.R. Pearson (1998), *Release 6.0 of GEMPACK - New Features and Changes From Release 5.1 and 5.2*, GEMPACK Document No. 4 **[GPD-4],** Monash University, Clayton, Second edition, October 1998.

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.