CENTRE of POLICY STUDIES and the IMPACT PROJECT

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

Telephone:
(03) 990 52398, (03) 990 55112
from overseas:
61 3 990 52398 or
61 3 990 55112

Fax numbers:
(03) 990 52426, (03)990 55486
from overseas:
61 3 990 52426 or
61 3 990 55486

e-mail
impact@vaxc.cc.monash.edu.au

Notes on the Implementation of MONASH-MRF:

A Multiregional Model of Australia

by

Farzana Naqvi and Matthew W. Peter

Centre of Policy Studies
Monash University

Working Paper No. OP-82 April 1995

ISSN 1 031 9034

ISBN 0 7326 0723 X

The Centre of Policy Studies (COPS) is a research centre at Monash University devoted to quantitative analysis of issues relevant to Australian economic policy. The Impact Project is a cooperative venture between the Australian Federal Government and Monash University, La Trobe University, and the Australian National University. During the three years January 1993 to December 1995 COPS and Impact will operate as a single unit at Monash University with the task of constructing a new economy-wide policy model to be known as *MONASH*. This initiative is supported by the Industry Commission on behalf of the Commonwealth Government, and by several other sponsors. The views expressed herein do not necessarily represent those of any sponsor or government.

Abstract

This paper presents the computer implementation of the *MONASH-MRF* model. *MONASH-MRF* is a multiregional multisectoral model of the Australian Economy. Included is a complete documentation of the model in the TABLO language, consisting of the specification of the model's equation system, variables and coefficients. The paper also provides a step-by-step guide to the conventions of the TABLO language to allow the reader to interpret the specification of the model. Finally, the paper contains instructions and files necessary to generate a forecasting solution of the model.

Keywords: multiregional, CGE, GEMPACK, forecasting, computer implementation.

J.E.L. Classification numbers: C68, R10, R13.

Preamble

This document contains a draft version of chapter 3 and appendix 3 from the forthcoming monograph, *MONASH-MRF: A Multiregional Multisectoral Model of the Australian Economy*. Chapter 3 is concerned with the computer implementation of the model. Appendix 3 presents a complete documentation of the model in the TABLO language, including the specification of the model's equation system, variables and coefficients. Included in chapter 3 is a guide to the conventions of the TABLO language so that taken together, chapter 3 and appendix 3 allow the reader to interpret the specification of the model. Chapter 3 also contains instructions and files necessary to generate a forecasting solution of the model.

While the material presented in this document has been prepared by Naqvi and Peter, the development of MONASH-MRF has involved many researchers at the Centre of Policy Studies. The project was initiated in mid 1992 under the direction of B.R. Parmenter. The first version of the model, consisting of the CGE core module, was constructed by J.M. Horridge, with suggestions from P.B. Dixon. G.A. Meagher added the government finance module. S.H. Han undertook initial work on the input-output database. Short-run comparative-static simulations were performed by Meagher and Parmenter. These first efforts are documented in the annotated TABLO file of Appendix 3, Han (1992), Meagher and Parmenter (1993) and Meagher (1993 and 1994).

CONTENTS

Figure

```
! The following equations are used to update the final-year values of regional
growth, foreign migrtion and regional migration. This is their only purpose. !

E_popt # Update the regional population in final year #
                (all,q,REGDEST)
(C_POP(q)/100)*pop(q) =    del_poplt(q) + del_gt(q) + del_fmt(q)
                         + del_rmt(q)
                         + (C_G_0(q)+C_FM_0(q)+C_RM_0(q))*delpopfudget;

E_poplt # Update the regional population in final year in 2nd closure #
                (all,q,REGDEST)
C_POP(q)*pop(q) = 100*del_poplt(q) + (C_POP(q)/PRIOD)*pop(q) + del_fpoplt(q);
E_GT # Update the natural growth (change) in final year T #
                (all,q,REGDEST)
100*del_gt(q) = C_G_0(q)*pop(q) + 100*C_POP(q)*del_fgt(q);

E_RMT0 # Adjustment to reg.mig. to ensure adding-up condition #
                (all,q,REGDEST)
del_rmt(q) = del_rmt0(q)+ del_frmt0;

E_addup # Adding-up condition #
Sum(q,REGDEST, del_rmt(q)) = 0;



<< END OF FILE >>
```

References for Appendix 3

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

Horridge, J., B. Parmenter and K. Pearson (1993) 'ORANI-F: A General Equilibrium Model of the Australian Economy', *Economic and Financial Computing*, 3, 2, pp.71-140

Chapter 3 Notes on the Implementation of MONASH-MRF

3.1 Introduction

MONASH-MRF[1] (MMRF) has been implemented using GEMPACK software. GEMPACK is a suite of general purpose computer programs designed to solve general equilibrium models. Section 3.2 describes the various stages in the GEMPACK implementation of MMRF. To implement and solve MMRF, GEMPACK requires the specification of the model in TABLO language. TABLO language resembles algebraic language. Section 3.3 introduces the main conventions of the TABLO language. The aim of this section is to facilitate the interpretation of the MMRF TABLO file. An annotated version of the MMRF TABLO file is presented in Appendix 3.

GEMPACK is a user-friendly software and can be run either interactively or through stored input files. Stored input files used in the implementation and solution process have been prepared and Section 3.3 concludes with the presentation and description of these files. Full documentation of GEMPACK is contained in Harrison and Pearson (1994a, 1994b and 1994c) .

3.2 Technical notes on the computer implementation of MMRF

Figure 3.1 shows the steps involved in the computer implementation of MMRF using GEMPACK. Steps 1 and 2 cover the preparation of the computer code of the model, step 3 covers the solution of the model and step 4 covers the preparation of the text file containing simulation results. In subsections 3.2.1 to 3.2.4 we describe the steps in the order given in Figure 3.1. However, these steps need not always be repeated for each simulation of the model. Once the computer code has been prepared, the model can be solved for various closures and shocks, i.e., step 3 can be repeated for different experiments without the need to repeat steps 1 and 2. Steps 1 and 2 need only be repeated when regenerating the TABLO source code. This most usually occurs when changes are made to the model's structure, e.g., when adding new equations.

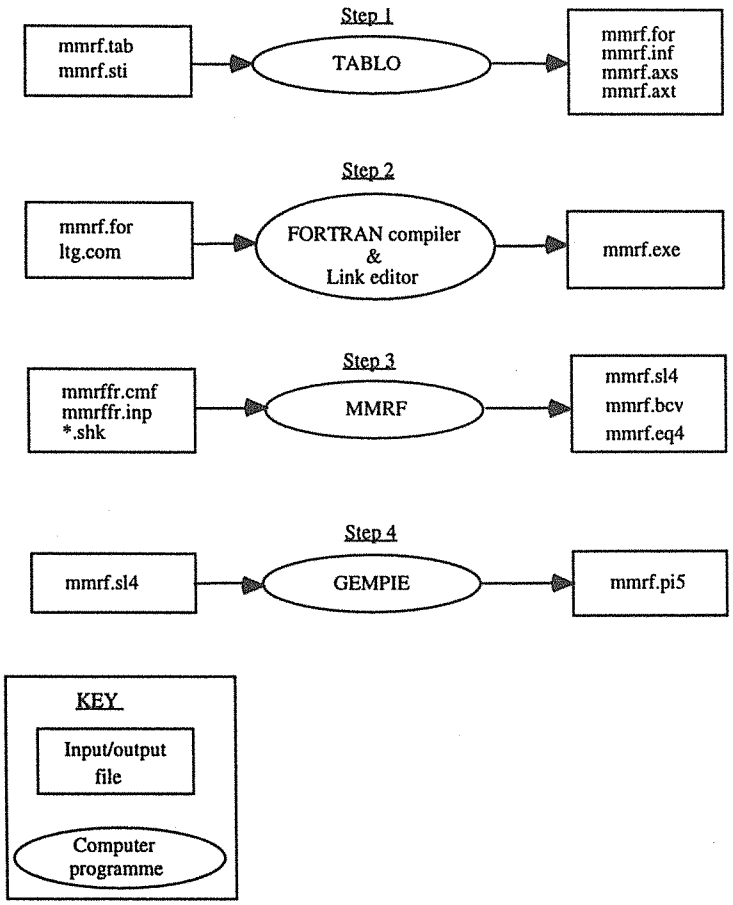
Included in the description of steps 1 to 4 are the DOS commands necessary to activate and run the various GEMPACK programs. The input/output file names used/created by GEMPACK are written in *italics*, all GEMPACK program names are in upper case and the user input commands, entered at the terminal, are written in **bold** script.

3.2.1 Step 1

The first step in the computer implementation is the preparation of a computer program containing all the equations of the model. Prior to the availability of general purpose software such as GEMPACK and GAMS, the practice was to write a tailor-made computer program for each model. General purpose software prepares a computer program from a source code which closely resembles ordinary algebra. This

[1] MONASH-MRF is an applied general equilibrium Multi Regional Forecasting model of the Australian economy.

Figure 3.1 Steps involved in the computer implementation and simulation of MMRF



means that the modeller no longer needs to master computer languages such as FORTRAN. In addition, the source code is sufficiently transparent that, with the knowledge of a few conventions, the model's equations (and hence structure) can be understood directly from the text of the source code. TABLO is the GEMPACK program which processes the description of the model documented in the simple algebraic form of the source code.

Two input files are required to run TABLO: a file containing the specification of the model coded in the TABLO language and; a file containing a set of stored input commands for the TABLO program. The TABLO file containing the specification of


```
C_WPOP(q)    = wpop(q);
```

! Section 6.4: Formulae for the labour market & regional migration module !

```
FORMULA
                (initial)(all,q,REGDEST)
C_FM_0(q) = C_FM(q);
                (initial)(all,q,REGDEST)
C_G_0(q)  = C_G(q);
C_NATEMPLOY = sum(q,REGDEST,C_EMPLOY(q));
C_NATLABSUP = sum(q,REGDEST,C_LABSUP(q));
C_PA2 = 50*(PRIOD+1);
                (initial)(all,q,REGDEST)
C_RM_0(q) = C_RM(q);
                (all,q,REGDEST)
C_PR1(q) = 100*PRIOD*(C_RM_0(q)+C_FM_0(q)+C_G_0(q));

DISPLAY
C_NATEMPLOY;
C_NATLABSUP;
```

! Section 6.5: Equations of the labour market & regional migration module !

```
EQUATION
E_delNatFM # Foreign mig. nationally is sum of Foreign mig. to regions #
del_natfm = sum(q,REGDEST,del_fm(q));

E_delNatg # Natural pop nationally is sum of natural pop of regions #
del_natg = sum(q,REGDEST,del_g(q));

E_natlabsup # National labour supply #
C_NATLABSUP*natlabsup = sum(q,REGDEST,C_labsup(q)*labsup(q));

E_Natemploy # National employment #
C_NATEMPLOY*Natemploy = sum(q,REGDEST,C_EMPLOY(q)*employ(q));

E_NatUnr # P-point change in natural unemployment rate #
C_NATLABSUP*del_natunr = C_NATEMPLOY*(natlabsup-natemploy);

E_rempl_interf # interface employ and l #
                (all,q,REGDEST)
l(q) = employ(q) + f_l(q);

E_del_labsup # P-point changes in regional unemployment rates #
                (all,q,REGDEST)
C_labsup(q)*del_unr(q)=C_EMPLOY(q)*(labsup(q)-employ(q));

E_wpop # regional labour supply #
                (all,q,REGDEST)
labsup(q) = pr(q) + wpop(q);

E_Pop # regional working age population #
                (all,q,REGDEST)
wpop(q) = pop(q) + f_wpop(q);

E_del_RM # Accumulation of regional population #
                (all,q,REGDEST)
C_POP(q)*pop(q) =   C_PR1(q)*delrpfudge(q)
                  + C_PA2*(del_rm(q)+del_fm(q)+del_g(q))+f_pop(q);

E_Pop_interf # Interface Pop and qhous #
                (all,q,REGDEST)
qhous(q) = pop(q) + f_qhous(q);

E_RM_addup # Adding-up condition on reg. mig. #
delf_rm = sum(q,REGDEST,del_rm(q));

E_RM_0 # ABS pop forecasts can drive inter-regional migration #
                (all,q,REGDEST)
del_rm(q) = del_rm_0(q) + delf_rm_0;
```

```
                (change) (all,q,REGDEST)
del_rmt0(q)     # Ordinary change in regional migration: ABS forecasts #;
                (change)(all,q,REGDEST)
delrpfudge(q)   # Dummies in equation E_del_RM #;
                (change) (all,q,REGDEST)
del_unr(q)      # Percentage-point changes in regional unemploy rate #;
                (all,q,REGDEST)
employ(q)   # regional employment: persons #;
                (all,q,REGDEST)
f_1(q)          # Shifter in regional employment #;
                (all,q,REGDEST)
f_pop(q)        # Shifter in equation E_del_RM #;
                (all,q,REGDEST)
f_wpop(q)       # Shifter in equation E_WPop #;
                (all,q,REGDEST)
f_qhous(q)      # Shifter in equation E_Pop_interf #;
                (all,q,REGDEST)
pop(q)          # regional population #;
                (all,q,REGDEST)
pr(q)           # regional workforce participation rate #;
                (all,q,REGDEST)
wpop(q)         # regional population of working age #;
```

! Section 6.2: Coefficient declarations for the labour market & regional migration module !

```
COEFFICIENT
                (all,q,REGDEST)
C_FM(q)         # regional foreign migration #;
                (all,q,REGDEST)
C_FM_0(q)       # Base-year regional foreign migration #;
                (all,q,REGDEST)
C_G(q)          # regional natural population growth (births-deaths) #;
                (all,q,REGDEST)
C_G_0(q)        # Base-year regional natural pop. growth #;
C_NATLABSUP     # National labour supply #;
C_NATEMPLOY     # National employment: Persons #;
C_PA2           # Coefficient in pop. accumulation equation #;
                (all,q,REGDEST)
C_POP(q)        # regional population #;
                (all,q,REGDEST)
C_PR1(q)        # constant term in regional pop. accumulation equation #;
                (all,q,REGDEST)
C_RM(q)         # Inter-regional migration #;
                (all,q,REGDEST)
C_RM_0(q)       # Base-year interregional migration #;
                (all,q,REGDEST)
C_WPOP(q)       # regional population of working age #;
```

! Section 6.3: Reads and updates for the labour market & regional migration module !

```
READ
C_FM    from file PDATA header "RFRM";
C_G     from file PDATA header "RGRO";
C_POP   from file PDATA header "RPOP";
C_RM    from file PDATA header "RRGM";
C_WPOP from file PDATA header "RWAP";

UPDATE
                (all,q,REGDEST)
C_POP(q) = pop(q);
                (change) (all,q,REGDEST)
C_RM(q)  = del_rmt(q);
                (change) (all,q,REGDEST)
C_FM(q)  = del_fmt(q);
                (change) (all,q,REGDEST)
C_G(q)   = del_gt(q);
! C_LABSUP and C_EMPLOY are read in government finance module to compute
  two coefficients for unemployment benefits equation. We update them here. !
                (all,q,REGDEST)
C_EMPLOY(q) = employ(q);
                (all,q,REGDEST)
C_LABSUP(q) = labsup(q);
                (all,q,REGDEST)
```


MMRF is called *mmrf.tab*. Both input files are text files and can be edited using any text editor.

As shown in Figure 3.1 at step 1, TABLO makes four output files. A file *mmrf.for* that contains the FORTRAN code of the model, an information file *mmrf.inf* and two auxiliary files; *mmrf.axt* and *mmrf.axs*. The first two files are text files and can be edited while the other two files are binary files. The information file reproduces the TABLO file with error messages if there are errors in the TABLO file. An error message follows the line containing an error.

Block 1 gives the commands to be entered at the terminal to perform step 1. After each command, a comment follows the exclamation mark. The comments and the greater-than signs are not entered.

Block 1 Commands for Step 1

> TABLO	! Run TABLO program
sti	! Read inputs from an stored input file
mmrf.sti	! The input file name

The last line of block 1 activates the stored input file *mmrf.sti*. This file is reproduced and discussed in section 3.3.3.

3.2.2 Step 2

In Step 2, the FORTRAN code prepared in step 1 is compiled and linked. A batch file (*ltg.bat*) has been created to perform these tasks. Block 2 contains the command, entered at the terminal, that runs the batch file.

Block 2 Command for Step 2

> LTG mmrf

The output of step 2 is the executable computer code of the model (*mmrf.exe*).

3.2.3 Step 3

In step 3, the TABLO generated computer code of the model (*mmrf.exe*) is run, producing a solution for a particular experiment. Block 3-A gives the command to perform step 3. The MMRF program reads commands from a text file *mmrffr.inp*.

Block 3-A Command to do Step 3

>MMRF <mmrffr.inp ! Run MMRF and read input from *mmrffr.inp* file

The *mmrffr.inp* file is reproduced in block 3-B. This file directs GEMPACK to another stored input file (*mmrfr.cmf*) which contains names of the input data files, the closure and the shock values for the exogenous variables. Apart from the data stored in binary files, the mmrf code reads data from the terminal at run time (i.e., during the actual processing). These data are stored in the *mmrffr.inp* file. The last three lines in the file give these data values. The comment at the end of each line gives the name of the coefficient to which the value is assigned. The role of these coefficients is discussed in detail in the TABLO file of Appendix 3.

Block 3-B The mmrffr.inp file

```
cmf             ! Read input from the cmf file
mmrffr.cmf      ! File name
6               ! Value for Fred coefficient
6               ! Value for Priod coefficient
1 2 3 4 5 6     ! Value for years coefficient
```

The stored input file (*mmrffr.cmf*) containing the commands for a forecasting experiment is reproduced and described in section 3.3.4.

The output of step 3 is a binary solution file (*mmrff1.sl4*) containing the values of the endogenous variables for a particular experiment (i.e., the solution of the model) and two intermediary binary files.

3.2.4 Step 4

GEMPIE is the GEMPACK program which converts the binary solution file (*mmrff1.sl4*) into a text file. Block 4-A gives the commands to run GEMPIE.

Block 4-A Commands to do Step 4

```
>GEMPIE
                ! A carriage return
mmrff1          ! The solution file name
a
mmrffr          ! The output file name
MMRF results    ! A title to be printed on each page of the results
4               ! Results up to 4 decimal places
```

As mentioned, the output of GEMPIE is a text file (*mmrffr.pi5*) that contains a list of all exogenous and endogenous variables, the exogenous variables that are shocked and their shock values, and the values of all the endogenous variables. This text file can be printed or edited using a text editor. Block 4-B reproduces a part of the output file for illustrative purposes.

Block 4-B Excerpt from mmrffr.pi5

```
 PAGE 1      MMRF results

 natlabind (IND)  Employment by Industry

 Agricult      Mining      Manufact     Publ_Util    Construct    Dom_Trade
 1.3928        19.0590     -3.4200      -11.0414     11.6610      12.2737

 Transp_Comm    Finance      Housing     Publ_Srv     Comm_Srv     Pers_Srv
 2.8777         11.0710      0.0000*     13.2322      12.3814      9.0038

 NC_Imports
 6.5531

yr_r (REGDEST)  Real GDP - regions

 NSW          VIC          QLD          SA           WA           TAS
 11.8782      10.4887      22.7328      10.4950      18.7424      13.4331

 NT           ACT
 24.8171      24.8251
```


! Part 6: LABOUR MARKET & REGIONAL MIGRATION MODULE !

```
! This module computes population in each region from natural growth,
  foreign migration and inter-regional migration. There are two main ways to use
  this module:

 (i) Take on board the forecasts of the three population flows from a
     demographic model thereby exogenously determining regional populations.
     For example, the ABS makes forecasts of these flows and of regional
     population. Allow the labour market & migration module to determine
     regional labour supply from the exogenously specified regional population
     and from given settings of regional participation rates and movements in the
     ratios of population to population of working age. With labour supply
     determined, the labour market and regional migration module will determine
     either inter-regional wage differentials, (given regional unemployment
     rates) or regional unemployment rates (given regional wage differentials).
     With given regional unemployment rates and regional labour supply, regional
     employment is determined as a residual and wage differentials adjust to
     accommodate the labour market outcome.  Fixing wage differentials determines
     the demand for labour so that with regional labour supply given, the model
     will determine regional unemployment rates as a residual.

(ii) Alternatively, exogenously specify inter-regional wage differentials and
     regional unemployment rates. The labour market and regional migration
     module will then determine regional labour supply and regional population
     for given settings of regional participation rates and ratios of population
     to population of working age.

The module has been designed with sufficient flexiblity to allow many variations
on the two general methods described above. Importantly, for forecasting, the
module allows for some regions to be subject to method (i) and other regions to
be subject to method (ii) in the same simulation.
!
```

! Section 6.1: Variable declarations for the labour market & regional migration module !

```
VARIABLE
                  (change)
del_natfm       # Ordinary change in foreign migration in Aust. #;
                  (change)
del_natg        # Ordinary change in natural pop. (births-deaths) in Aust.#;
                  (change)
del_natunr      # P-point change in Economy-wide unemployment rate #;
                  (change)
delf_rm         # Shifter in equation E_RM_Addup #;
del_frmt0       # Shifter for regional migration from ABS forecasts in T #;
                  (change)
delf_rm_0       # Shifter in equation E_RM_0 #;
                  (change)
delpopfudget    # Fudge factor in equation E_popT #;
natlabsup       # National labour supply #;
natemploy       # National employment #;
                  (change) (all,q,REGDEST)
del_fgt(q)      # Shift in nat. growth of regional population in year T #;
                  (change)(all,q,REGDEST)
del_fm(q)       # Ordinary changes in foreign migration: regions #;
                  (change) (all,q,REGDEST)
del_fmt(q)      # Ordinary change in foreign migration  for update #;
                  (change) (all,q,REGDEST)
del_fpop1t(q)   # Shift in regional population in year T-1 #;
                  (change)(all,q,REGDEST)
del_g(q)        # Ordinary changes in natural pop. (births-deaths): regions #;
                  (change) (all,q,REGDEST)
del_gt(q)       # Ordinary change in nat. growth in regional pop. for update #;
                  (change) (all,q,REGDEST)
del_pop1t(q)    # Ordinary change in regional population in year T-1 #;
                  (change)(all,q,REGDEST)
del_rm(q)       # Ordinary change in inter-regional migration #;
                  (change) (all,q,REGDEST)
del_rm_0(q)     # ABS forecast of inter-regional migration #;
                  (change) (all,q,REGDEST)
del_rmt(q)      # Ordinary change in regional migration  for update #;
```

```
deldebt_ratio # Change in Debt/GDP ratio#;
                (change)
deldebt       # Ordinary change in foreign debt  #;
                (change)
levdebt_ratio # Level of Debt/GDP ratio #;
```

! Section 5.2: Coefficients for the foreign-debt accumulation module !

```
COEFFICIENT
B0            # Original Real trade deficit#;
BT            # Real trade deficit#;
DEBT          # Real foreign Debt#;
DEBT_RATIO    # Debt/GDP ratio#;
DEBT0         # Original Real foreign debt#;
DEBT_RATIO_0  # Original Debt/GDP ratio#;
M_DEBT        # A constant #;
N_DEBT        # A constant#;
                (all,e,YEARS)
ORD(e);       ! = e for e=1,PRIOD !
P_GLOBAL      # Converts $A into 'Real' terms #;
R_WORLD       # World interest rate:number like 1.04#;
```

! Section 5.3: Read and update statements for the foreign-debt accumulation module !

```
READ
ORD From Terminal;       ! Enter by user at run-time !
DEBT_RATIO From File YDATA Header "DGDP";
R_WORLD From File YDATA Header "RWLD";

UPDATE
                (change)
DEBT_RATIO = deldebt_ratio;
P_GLOBAL = Natxim;
```

! Section 5.4: Formulae and displays for the foreign-debt accumulation module !

```
FORMULA
                (Initial)
P_GLOBAL = 1; ! Original value arbitrary !
BT = (NATAGGIMP - NATAGGEXP)/P_GLOBAL;
                (Initial)
B0 = BT;
M_DEBT = Sum(e,YEARS,([ORD(e)-1]/PRIOD)*R_WORLD^(PRIOD-ORD(e)));
                (Initial)
N_DEBT = Sum(e,YEARS,R_WORLD^(PRIOD-ORD(e)));
                (Initial)
DEBT_RATIO_0 = DEBT_RATIO;
DEBT = DEBT_RATIO*NatGDPEX/P_GLOBAL;
                (Initial)
DEBT0 = DEBT;
DISPLAY DEBT_RATIO_0;
```

! Section 5.5: Equations of teh foreign-debt accumulation module !

```
EQUATION
E_delbt # Ordinary change in Real trade deficit#
100*P_GLOBAL*delbt =  NATAGGIMP*(natimpvol)
                    - NATAGGEXP*(natexpvol);

E_deldebt_ratio # Change in Debt/GDP ratio#
deldebt_ratio = (DEBT_RATIO/DEBT)*deldebt-(DEBT_RATIO/100)*(natgdpexp-natxim);

E_deldebt # Ordinary change in foreign debt#
deldebt = (DEBT0*(R_WORLD^PRIOD-1)+B0*N_DEBT)*deldfudge+M_DEBT*delbt;

E_levdebt_ratio # Level Debt/GDP#
levdebt_ratio = DEBT_RATIO_0*delunity+deldebt_ratio;
```



```
employ (REGDEST)  regional employment: persons

 NSW          VIC          QLD          SA           WA           TAS
 6.5180       4.7859       16.2558      4.7955       10.4064      7.4105

 NT           ACT
 15.0419      21.3140
```

3.3 The MONASH-MRF TABLO file and stored input files

3.3.1 Overview of the MONASH-MRF TABLO file

In this section we describe the structure of the MMRF TABLO file; *mmrf.tab*. The file which we refer to in this section is the annotated version of *mmrf.tab* presented in Appendix 3.

The *mmrf.tab* contains the specification of the MMRF model in that it includes the model's equations, variables and coefficients. The *mmrf.tab* presents the elements of the model in five modules:

(i) the CGE core module;
(ii) the government finance module;
(iii) the capital and investment module.
(iv) the foreign debt accumulation module and
(v) the labour market and regional migration module.

Each equation within the MMRF system is uniquely allocated to one of the modules. Variables and coefficients, however, may appear in one or more modules.

The modules are presented in Parts 2 to 6 of *mmrf.tab*. The modules are preceded in the file by Part 1 (Preliminaries) which describes the basic content of each module. The CGE core module is presented in Part 2. Sections 2.1[2] and 2.3 identify the model's data files[3] and sets respectively. Subsets are also identified in section 2.2. The data files and sets are common to all modules, hence, sections 2.1 and 2.2 suffice for the remaining modules. This means that data and set declarations appear only in the CGE core module.

The structure of the remainder of Part 2 (the CGE core) module, is common to all modules of the model. The modules are ordered as follows:

- declarations of variables;
- declarations of coefficients;
- read statements associated with those coefficients taken directly from the database;
- update statements associated with those coefficients taken directly from the database;
- formulae which calculate coefficients derived from the database coefficients;
- equations.

The TABLO language, in which *mmrf.tab* is written, is easily understood once key notational conventions are explained. The explanation of these conventions is the purpose of the following section.

---

2 When referring to *mmrf.tab*, the sections are those listed in the 'Contents' on the first page of Appendix 3.

3 Section 2.1 also lists a display file sometimes used for interrogating database values.

3.3.2 The TABLO language

In this section we introduce the rudiments of the TABLO language by explaining examples from *mmrf.tab* presented in Appendix 3. We follow the ordering outlined in section 3.3.1 above.

Data files

Data is read from prepared files and used to assign values to coefficients in the model's equations. The TABLO language gives the logical name of the data files. For MMRF, there are four such files. The following example is extracted from section 2.1 of *mmrf.tab*:

```
FILE MDATA
  # File containing input-output tables and other data for the CGE core
module#;
```

The first word, FILE, is a key word which defines the statement type. Then follows the logical name of the file. In this example it is MDATA. The descriptive text between # symbols is optional. Finally, a semicolon denotes the end of TABLO statements.

Sets

Sets, variables and coefficients must be explicitly declared. An example of a set declaration, drawn from section 2.2 of *mnrf.tab* is as follows:

```
SET IND # Industries #
(Agricult,Mining,Manufact,Publ_Util,Construct,Dom_Trade,
 Transp_Comm,Finance,Housing,Publ_Srv,Comm_Srv,Pers_Srv,
 NC_Imports);
```

As in the 'File' declaration, the first word SET is a keyword defining the statement type. This is followed by the set identifier IND and an optional description between # symbols. In brackets, following the optional description, the elements of the set IND(dustry) are listed.

Variables and Coefficients

The following example shows two blocks of variable and coefficient declarations from the CGE core module. The variable block is extracted from section 2.3 and the coefficient block is from section 2.4 of *mmrf.tab*.

```
VARIABLE
natcR          # Real household consumption #;
                (all,q,REGDEST)
cr(q)          # Real household consumption #;

COEFFICIENT
                (all,j,IND)(all,q,REGDEST)
CAPITAL(j,q)   # Total capital in industry j #;
                (all,i,COM)
NATTARIFF(i)   # Tariff revenue #;
```

The above examples of variable and coefficient declarations demonstrate that keywords (e.g., VARIABLE, COEFFICIENT) need not be repeated when the following statement is of the same type. In *mmrf.tab*, the convention of naming


```
READ
VALKT From File YDATA Header "VALK";
DEP   From File YDATA Header "DPRC";
FRED  From Terminal; ! Enter by user at run-time !
PRIOD From Terminal; ! Enter by user at run-time !
! FRED=2 for comparative static simulations and PRIOD is a big number.
  Alternatively FRED = PRIOD for forecasting simulations. !

UPDATE
                (all,j,IND)(all,q,REGDEST)
VALKT(j,q) = pi(j,q)*curcap(j,q);
                (all,j,IND)(all,q,REGDEST)
VALK_T1(j,q) = pi(j,q)*curcap_t1(j,q);

FORMULA
K_TERM = 1+(1/PRIOD);
                (initial)(all,j,IND)(all,q,REGDEST)
VALK_0(j,q) = VALKT(j,q);
                (initial)(all,j,IND)(all,q,REGDEST)
VALK_T1(j,q) = VALKT(j,q);
                (initial)(all,j,IND)(all,q,REGDEST)
INVEST_0(j,q) = INVEST(j,q);
```

! Section 4.4: Equations of the capital accumulation and investment !

```
EQUATION
E_r0 # Definition of rates of return to capital #
                (all,j,IND)(all,q,REGDEST)
r0(j,q) = QCOEF(j,q)*(p1cap(j,q) - pi(j,q));

E_f_rate_xx # Capital growth rates related to rates of return #
                (all,j,IND)(all,q,REGDEST)
(r0(j,q) - natr_tot) = BETA_R(j,q)*[curcap(j,q) - kt(q)] + f_rate_xx(j,q);
! Notice regional subscript on kT but not on natr_tot !

E_naty # Total real investment #
                (all,j,IND)
(Tiny+NatINVEST(j))*naty(j) = sum(q,REGDEST,INVEST(j,q)*y(j,q));

E_curcapT1 # Capital stock in period T+1#
                (all,j,IND)(all,q,REGDEST)
  IF(FRED ne PRIOD, curcap_t1(j,q) - curcap(j,q))
+ IF(FRED eq PRIOD, curcap_t1(j,q) - K_TERM*curcap(j,q)) =0;

E_yT # Investment in period T#
                (all,j,IND)(all,q,REGDEST)
  IF(FRED ne PRIOD, curcap(j,q) - y(j,q)- 100*delf_rate(j,q))
+ IF(FRED eq PRIOD, + VALKT(j,q)*DEP(j)*curcap(j,q)
                    + (Tiny + INVEST(j,q))*y(j,q)
                    - 100*(VALK_0(j,q)*(1-DEP(j))-INVEST_0(j,q))*delkfudge
                    + 100*delf_rate(j,q) ) - VALK_T1(j,q)*curcap_t1(j,q) = 0;
```

! Part 5: FOREIGN DEBT ACCUMULATION !

```
! The accumulation equation compounds the base-year debt and the additonal
  debt accruing during the forecast period due to trade deficits at real
  foreign interest rate. That is,
    DEBT(T) = DEBT(0)(R_WORLD)^T + Σ_t BT*(R_WORLD)^(T-t-1)
!
```

! Section 5.1: Sets and variables for the foreign-debt accumulation module !

```
SET YEARS MAXIMUM SIZE 100 SIZE PRIOD;

VARIABLE
                (change)
deldfudge      # Dummy variable in equation E_delDebt #;
                (change)
delunity       # Dummy to activate foreign debt accum. equation #;
                (change)
delbt          # Ordinary change in Real trade deficit#;
                (change)
```

```
E_ig # Aggregate nominal government investment (residual) #
NATAGGINV * natin = C_IP * ip + C_IG * ig;

E_ig_r_fed # Nominal government investment - federal (residual) #
C_IG * iG = sum(q,domdest, C_IG_R(q) * ig_r(q));

E_c_b # Consumption function #
                (all,q,REGDEST)
c(q) = yd_r(q) + miscf001(q);

E_rl # Relative income tax rates #
rl = rk + miscf002;
```

! Part 4: CAPITAL ACCUMULATION AND INVESTMENT MODULE !

```
! MONASH-MRF can be into two modes:(i) compartive static and (ii) forecasting.
  The modes require alternative treatments of capital formation and
  investment which we have modelled using conditional equations.

  In forecasting mode, investment demand in the terminal year of the forecast
  period (year T) is determined via an accumulation relation linking investment
  in the terminal year to the difference between capital stock in the terminal
  year and capital sock in the year after the terminal year (year T+1). The
  capital stock in year T+1 is determined on the assumption that the ratio of
  the base-year capital stock to that in the terminal year is equal to
  the ratio of capital stocks in the terminal year and the year after the
  terminal year. That is K(0)/K(T) = K(T)/K(T+1)

  In comparative-static mode, there is no fixed relationship between capital
  and investment. The user decides the required realationship on the basis of
  the requirements of the specific simulation.  For example, it is typical
  in long-run comparative-static simulations to assume that the growth in
  capital and investment are equal.
!
```

! Section 4.1: Variable declarations specific to the cap. acc. and investment module !

```
VARIABLE
                (change) (all,j,IND)(all,q,REGDEST)
delf_rate(j,q)
                # A change shifter in capital_accum equation #;
                (change)
delkfudge       # Dummy variable to switch on capital accumulation equation #;
                (all,j,IND) (all,q,REGDEST)
f_rate_xx(j,q)
                # Shifter, rate of return eq. #;
                (all,j,IND) (all,q,REGDEST)
curcap_t1(j,q)
                # Capital stock in period T+1 #;
natr_tot        # Average rate of return #;
```

! Section 4.2: Coefficient declarations for the capital accumu. and investment module !

```
COEFFICIENT
                (all,j,IND)(all,q,REGDEST)
VALKT(j,q)      # Assest-value of capital stock in period T #;
                (all,j,IND)
DEP(j)          # Depreciation factor, uniform across regions #;
                (integer)
FRED            # Switch variable, forecast to comp. stat. #;
                (integer)
PRIOD           # Number of years in a forecast period #;
K_TERM          # A constant in T+1 capital equation #;
                (all,j,IND) (all,q,REGDEST)
VALK_0(j,q)     # Base-year asset value of capital stock #;
                (all,j,IND)(all,q,REGDEST)
VALK_T1(j,q)    # Value of capital stock in period T+1 #;
                (all,j,IND)(all,q,REGDEST)
INVEST_0(j,q) # Base-year value of investment #;
```

! Section 4.3: Reads, updates and formulae for the capital acc. and investment module !

variables in lower case (nater, cr) and coefficients in upper case (CAPITAL, NATTARIFF) is followed. GEMPACK ignores case so that the convention of naming variables in lower case and coefficients in upper case is only a useful memory device for the model user. The expression in brackets preceding the variable name cr (i.e., all, q, REGDEST), indicates that the variable is a vector variable containing 'all' the elements of the set REGDEST (regional destination). REGDEST is defined in section 2.2 of *mmrf.tab*. It has eight elements; NSW, VIC, QLD, SA, WA, TAS, NT and ACT. Likewise, the coefficient NATTARIFF is a vector with elements belonging to the set COM(modity). The variable natcr is a scalar, while the coefficient CAPITAL ranges over two dimensions; IND and REGDEST. The suffixes in brackets following the variable and coefficient names, (i.e., (q), (j,q) (i)), denote vector variables and coefficients and identify the sets and elements over which they range. Following the variables' and coefficients' names are optional descriptions contained within # symbols.

Reads and updates

Coefficients must be assigned values. This can be done either by reading values from a file or by calculating values in terms of other coefficients using formulae. Formulae are discussed in the following subsection. Below is an example of a READ statements from sections 2.5 of *mmrf.tab*.

```
READ
LAND from file MDATA header "LAND";
```

The coefficients read directly from the database can be updated in a multistep solution, thereby creating a new database representing post-solution values. This requires the inclusion of update statements such as:

```
UPDATE
                (all,j,IND)(all,q,REGDEST)
LAND(j,q) = p1land(j,q)*n(j,q);
```

which is the default update statement, causing LAND(j,q) to be increased after each step by [p1land(j,q)+n(j,q)]%, where pland(j,q) and n(j,q) are the percentage changes computed at the previous step.

Formulae

Coefficients defined in terms of other coefficients are calculated using formulae. For example, the formula

```
FORMULA
                (all,i,COM)
NATTARIFF(i) = sum(q,REGDEST,TARIFF(i,q));
```

defines the coefficient NATTARIFF as the sum of the TARIFF revenue in each region in the database. The right hand side of the above formula uses the TABLO summation notation equivalent of the Σ notation used in algebra. It defines the sum over an index q running over the set REGDEST of the TARIFF coefficients.

Equations

The *mmrf.tab* defines the percentage-change equations of the model. Below is the TABLO code for the CES-demand equation by firms for the domestic-composite input to capital creation:

```
EQUATION
E_x2c # Demand for domestic composite, User 2 #
              (all,i,COM)(all,j,IND)(all,q,REGDEST)
x2c(i,j,q) = x2o(i,j,q) - SIGMA2O(i)*(p2c(i,j,q)-p2o(i,j,q));
```

Following the keyword EQUATION is the equation identifier (E_x2c) and the optional equation description (between # symbols). The expression (all,i,COM)(all,j,IND)(all,q,REGDEST) indicates the equation is a matrix equation containing 1,352 scalar equations which is the product of the elements of the sets COM, IND and REGDEST (13x13x8) over which the equation ranges. Notice also that in TABLO language, the asterisk (*) replaces the multiplication symbol (x).

3.3.3 The stored input file to run TABLO

This section discusses the stored input file *mmrf.sti*. This file contains instructions required by GEMPACK for the processing of *mmrf.tab*. Processing of a TABLO file involves four tasks: (i) checking the TABLO file for syntax and other errors; (ii) omitting some exogenous variables; (iii) substituting out some endogenous variables; and (iv) generating a FORTRAN code of the model. The purpose of tasks (ii) and (iii) is to condense the model. While GEMPACK is capable of solving models of very large size, it is efficient from a computing and economic viewpoint to reduce the size of the model by omitting and substituting out variables. The variables that are omitted and substituted will vary depending on the particular application. The model user is able to specify the list of omitted and substituted variables in *mmrf.sti*. The list can be changed at the discretion of the user.

The *mmrf.sti* is reproduced below. The following is a description of the file. Each line starting with an exclamation mark is a comment. Each input command starts on a new line and is followed by a brief descriptive comment. The first input command chooses the default setting of running TABLO. The second command gives the TABLO file name. The next line contains the condensation command (c). This is followed by the command to omit variables (o) and the list of omitted variables. Following the list of omitted variables is the command to substitute out variables (s). Following each variable that is substituted is the name of the equation in the model used to substitute out the variable. If the user decides to change the list of omitted and/or substituted variables, steps 1 and 2 must be repeated to regenerate the code.

Given the condensation contained in *mmrf.sti*, the model's size exceeds the GEMPACK default size. However, users can change the default settings to accommodate the model size with which they wish to operate. For example, towards the end of *mmrf.sti*, we have entered commands which change the default settings of GEMPACK to accommodate the condensation of MMRF contained in *mmrf.sti*.


```
hhldy210(q) =  natxi3 + C_HHLDD001(q) * labsup(q)
                      - C_HHLDD002(q) * l(q) + hhldf001(q);

E_hhldy220 # Other personal benefit receipts #
              (all,q,REGDEST)
hhldy220(q) = natxi3 + hhldf002(q);

E_hhldy200 # Personal benefit receipts #
              (all,q,REGDEST)
C_HHLDY200(q) * hhldy200(q) =  C_HHLDY210(q) * hhldy210(q)
                             + C_HHLDY220(q) * hhldy220(q);

E_hhldy300 # Other Income (net) #
              (all,q,REGDEST)
hhldy300(q) = yn_r(q) + hhldf003(q);

E_hhldy410 # PAYE taxes #
              (all,q,REGDEST)
hhldy410(q) = hhldyl10(q) + rl;

E_hhldy420 # Taxes on non-wage primary factor income #
              (all,q,REGDEST)
hhldy420(q) = hhldyl20(q) + rk;

E_hhldy430 # Other direct taxes #
              (all,q,REGDEST)
hhldy430(q) = tod_r(q);

E_hhldy400 # Direct taxes #
              (all,q,REGDEST)
C_HHLDY400(q) * hhldy400(q) =  C_HHLDY410(q) * hhldy410(q)
                             + C_HHLDY420(q) * hhldy420(q)
                             + C_HHLDY430(q) * hhldy430(q);

E_hhldy000 # Disposable income #
              (all,q,REGDEST)
C_HHLDY000(q) * hhldy000(q) =  C_HHLDY100(q) * hhldy100(q)
                             + C_HHLDY200(q) * hhldy200(q)
                             + C_HHLDY300(q) * hhldy300(q)
                             - C_HHLDY400(q) * hhldy400(q);

E_ydr # Disposable income #
              (all,q,REGDEST)
yd_r(q) = hhldy000(q);

E_upb # Aggregate unemployment benefit payments #
C_UPB * upb = sum(q,REGDEST, C_HHLDY210(q) * hhldy210(q));

E_pbp_r # Personal benefit payments - regions #
              (all,q,REGDEST)
pbp_r(q) = hhldy200(q);

E_pbpA # Aggregate personal benefit payments #
C_PBP * pbp = sum(q,REGDEST, C_HHLDY200(q) * hhldy200(q));

E_pbpB # Personal benefit payments - federal (residual) #
C_PBP * pbp = sum(q,domdest, C_PBP_R(q) * pbp_r(q));

! Subsection 3.8.6: Miscellaneous equations of the government finance module !

E_g_rA # Nominal government consumption - regions #
              (all,q,REGDEST)
g_r(q) = othnom5(q);

E_g_rB # Nominal government consumption - federal #
g_r("federal") = natothnom6;

E_ip # Aggregate nominal private investment #
C_IP * iP = sum(q,REGDEST, C_IP_R(q) * in(q));

E_ig_r_reg # Nominal government investment - regions #
              (all,q,REGDEST)
ig_r(q) = in(q);
```

```
E_softq220 # Other personal benefits (residual) #
               (all,q,DOMDEST)
C_SOFTQ200(q) * softq200(q) =  C_SOFTQ210(q) * softq210(q)
                             + C_SOFTQ220(q) * softq220(q);

E_softq300A # Subsidies - regions #
               (all,q,REGDEST)
softq300(q) = ti_r(q);

E_softq300B # Subsidies - federal (residual) #
C_SUBSIDIES * tI = sum(q,domdest, C_SOFTQ300(q) * softq300(q));

E_softq400 # Interest payments #
               (all,q,DOMDEST)
softq400(q) = natxi3 + debtg(q) + softf007(q);

E_softq510 # Commonwealth grants to regions - current #
C_SOFTQ510("federal") * softq510("federal")
                                = sum(q,REGDEST, C_SOFTY141(q) * softy141(q));

E_softq520 # Commonwealth grants to regions - capital #
C_SOFTQ520("federal") * softq520("federal")
                                = sum(q,REGDEST, C_SOFTY142(q) * softy142(q));

E_softq500 # Commonwealth grants to regions #
C_SOFTQ500("federal") * softq500("federal")
                           =   C_SOFTQ510("federal") * softq510("federal")
                             + C_SOFTQ520("federal") * softq520("federal");

E_softq600 # Other outlays #
               (all,q,DOMDEST)
softq600(q) = softq000(q) + softf006(q);

E_softq000 # Summary of financial transactions : expenditure-side total #
               (all,q,DOMDEST)
C_SOFTQ000(q) * softq000(q)  =  C_SOFTQ100(q) * softq100(q)
                              + C_SOFTQ200(q) * softq200(q)
                              + C_SOFTQ300(q) * softq300(q)
                              + C_SOFTQ400(q) * softq400(q)
                              + C_SOFTQ500(q) * softq500(q)
                              + C_SOFTQ600(q) * softq600(q);

E_dGstar # Net borrowing to total outlays: percent point change #
               (all,q,DOMDEST)
C_SOFTQ000(q) * dgstar(q) = C_SOFTY310(q) * (softy310(q) - softq000(q));

E_tOD_r # Other direct taxes #
               (all,q,REGDEST)
tod_r(q) = softy112(q);

E_debtG # Government debt #
               (all,q,DOMDEST)
C_DEBTG(q) * debtg(q) = C_SOFTY310(q) * (softy310(q) - natxi3);
```

! Subsection 3.8.5: Household Income and its components !

```
! Household disposable income !
E_hhldy110 # Wages, salaries and supplements #
               (all,q,REGDEST)
hhldy110(q) = z01_r(q);

E_hhldy120 # Non-wage primary factor income #
               (all,q,REGDEST)
hhldy120(q) = zg_r(q);

E_hhldy100 # Primary factor income #
               (all,q,REGDEST)
C_HHLDY100(q) * hhldy100(q) =  C_HHLDY110(q) * hhldy110(q)
                             + C_HHLDY120(q) * hhldy120(q);

E_hhldy210 # Unemployment benefit receipts #
               (all,q,REGDEST)
```


mmrf.sti

```
! Stored Input file used to run TABLO with the TABLO Input file MMRF.TAB
bat     ! This runs the program in "batch" mode
      ! Carriage return to finish option selection
mmrf    ! Name of tab file
      ! Carriage return -default name for Information file MMRF.INF
c     ! Condense the equations of MMRF to take less memory
o     ! Omit these variables - equivalent to exogenous with zero shock
a1
a1cap
a1land
a1oct
a2ind
a3com
DelTax1all
DelTax2all
DelTax3all
DelTax4all
DelTax5all
DelTax6all
DelTaxDEST
DelTaxSOURCE

s     ! Substitute out the following variable (these must be endogenous)
p1oct  ! Variable name
E_p1oct ! Equation name
s
p11aboi
E_p11aboi
s
x1oct
E_x1oct
s
x1a
E_x1a
s
p1o
E_p1o
s
p1c
E_p1c
s
x1c
E_x1c
s
x2a
E_x2a
s
p2o
E_p2o
s
p2c
E_p2c
s
x2c
E_x2c
s
x3a
E_x3a
s
p3o
E_p3o
s
p3c
E_p3c
s
```

```
x3c
E_x3c
s
p1a
E_p1a
s
DelTax1
E_DelTax1
s
p2a
E_p2a
s
DelTax2
E_DelTax2
s
p3a
E_p3a
s
DelTax3
E_DelTax3
s
p4r
E_p4r
s
DelTax4
E_DelTax4
s
p5a
E_p5a
s
DelTax5
E_DelTax5
s
p6a
E_p6a
s
DelTax6
E_DelTax6
s
x1o
E_x1o
s
x2o
E_x2o
s
x3o
E_x3o
s
x5a
E_x5a
s
x6a
E_x6a
s
x1marg
E_x1marg
s
x2marg
E_x2marg
s
x3marg
E_x3marg
s
x4marg
E_x4marg
s
x5marg
```



```
softy150(q) = yn_r(q) + softf004(q);

E_softy150B # Other revenue - federal #
softy150("federal") = yn + softf004("federal");

E_softy100 # Government revenue #
                (all,q,DOMDEST)
C_SOFTY100(q) * softy100(q) =  C_SOFTY110(q) * softy110(q)
                               + C_SOFTY120(q) * softy120(q)
                               + C_SOFTY130(q) * softy130(q)
                               + C_SOFTY140(q) * softy140(q)
                               + C_SOFTY150(q) * softy150(q);

E_softy200 # Consumption of fixed capital - general government #
                (all,q,DOMDEST)
softy200(q) = g_r(q);

E_softy300 # Financing transactions #
                (all,q,DOMDEST)
C_SOFTY300(q) * softy300(q) =  C_SOFTQ000(q) * softq000(q)
                               - C_SOFTY100(q) * softy100(q)
                               - C_SOFTY200(q) * softy200(q);

! This equation is the default definition of the regional government's
  real budget deficit. !
E_realdefr # Real budget deficit for region #
                (All,q,REGDEST)
realdef(q) = softy300(q) - xi3(q);

! This equation is the default defination of the Fed. government's
  real budget deficit. !
E_realdeff # Real budget deficit for Fed. #
realdef("federal") = softy300("federal") - natxi3;

E_softy320 # Increase in provisions #
                (all,q,DOMDEST)
softy320(q) = softq100(q) + softf005(q);

E_softy330 # Other financing transactions #
                (all,q,DOMDEST)
softy330(q) = softy300(q) + f_oft(q);

E_softy310 # Net borrowing (residual) #
                (all,q,DOMDEST)
C_SOFTY300(q) * softy300(q) =   C_SOFTY310(q) * softy310(q)
                                + C_SOFTY320(q) * softy320(q)
                                + C_SOFTY330(q) * softy330(q);

E_softy000 # Summary of financial transactions : income-side total #
                (all,q,DOMDEST)
C_SOFTY000(q) * softy000(q) =   C_SOFTY100(q) * softy100(q)
                                + C_SOFTY200(q) * softy200(q)
                                + C_SOFTY300(q) * softy300(q);

E_softq110 # Government consumption #
                (all,q,DOMDEST)
softq110(q) = g_r(q);

E_softq120 # Government investment #
                (all,q,DOMDEST)
softq120(q) = ig_r(q);

E_softq100 # Expenditure on goods and services #
                (all,q,DOMDEST)
C_SOFTQ100(q) * softq100(q) =  C_SOFTQ110(q) * softq110(q)
                               + C_SOFTQ120(q) * softq120(q);

E_softq210 # Unemployment benefits #
softq210 ("federal") = upb;

E_softq200 # Personal benefit payments #
                (all,q,DOMDEST)
softq200(q) = pbp_r(q);
```

```
E_softy111 # Income taxes #
softy111("federal") = ty;

E_softy112A # Other direct taxes #
              (all,q,REGDEST)
softy112(q) = yn_r(q) + softf001(q);

E_softy112B # Other direct taxes federal #
softy112("federal") = yn + softf011;

E_softy110 # Direct taxes #
              (all,q,DOMDEST)
C_SOFTY110(q) * softy110(q) =  C_SOFTY111(q) * softy111(q)
                             + C_SOFTY112(q) * softy112(q);

E_softy121 # Tariff revenue #
softy121("federal") = nattaxrevm;

E_softy122A # Other commodity taxes - regions #
              (all,q,REGDEST)
softy122(q) = ti_r(q);

E_softy122B # Other commodity taxes - federal (residual) #
C_TI * ti + C_SUBSIDIES * tI = sum(q,domdest, C_SOFTY122(q) * softy122(q));
E_softy123a # Payroll taxes - regions #
              (all,q,REGDEST)
softy123(q) = z03_r(q);

E_softy123b # Fringe benefits taxes - federal #
softy123("federal") = z09;

E_softy124 # Property taxes - regions #
              (all,q,REGDEST)
softy124(q) = z05_r(q);

E_softy125 # Land taxes - regions #
              (all,q,REGDEST)
softy125(q) = z07_r(q);

E_softy126A # Other indirect taxes - regions #
              (all,q,REGDEST)
softy126(q) = z09_r(q);

E_softy126B # Other indirect taxes - federal (residual) #
C_Z09 * z09 = sum(q,domdest, C_SOFTY126(q) * softy126(q));

E_softy120 # Indirect taxes #
              (all,q,DOMDEST)
C_SOFTY120(q) * softy120(q) =  C_SOFTY121(q) * softy121(q)
                             + C_SOFTY122(q) * softy122(q)
                             + C_SOFTY123(q) * softy123(q)
                             + C_SOFTY124(q) * softy124(q)
                             + C_SOFTY125(q) * softy125(q)
                             + C_SOFTY126(q) * softy126(q);

E_softy130 # Interest received #
              (all,q,DOMDEST)
softy130(q) = softq400 (q);

E_softy141 # Commonwealth grants to regions - current #
              (all,q,REGDEST)
softy141(q) = yn_r(q) + softf002(q);

E_softy142 # Commonwealth grants to regions - capital #
              (all,q,REGDEST)
softy142(q) = yn_r(q) + softf003(q);

E_softy140 # Commonwealth grants to regions #
              (all,q,REGDEST)
C_SOFTY140(q) * softy140(q) =  C_SOFTY141(q) * softy141(q)
                             + C_SOFTY142(q) * softy142(q);

E_softy150A # Other revenue - regions #
              (all,q,REGDEST)
```



```
E_x5marg
s
x6marg
E_x6marg
s
efflab
E_efflab
s
x1laboi
E_x1laboi
s
utility
E_utility
e     ! Exit TABLO condensation
a     ! Automatic code generation
cdm   ! Change default values of parameters
c     ! Change value of "defmnz" (default max. no. of non-zero elements)
60000  ! DEFMNZ
n     ! Don't change "mmshk" (def. max. number of shocks)
n     ! Don't change "mmlist" (def. max. number of variable listed)
    ! Carriage return
mmrf  ! Name of FORTRAN file
```

3.3.4 The stored input file to run the MMRF code

In this section we discuss the stored input file containing the commands necessary to solve the model. This input file contains; the names of the model's data files; the output files used to store the updated database; the output file holding the solution; the list of exogenous variables (the closure); the shock values to the exogenous variables (including the names of prepared text files containing shock values for matrix variables). The file also contains commands indicating the solution method to be used. The file concludes with a verbal description of the experiment which the file is designed to perform. In the case of *mmrffr.cmf*, this is a forecasting experiment.

mmrffr.cmf

```
! Command file mmrffr.cmf for the TABLO-generated program MMRF.
! This file creates the equations file for the MMRF model and solves the model
! for a forecast closure.
!
log file = mmrfeq.log ;
!
! Data files
!
file mdata = mdata.har ;
file ndata = ndata.har ;
file pdata = pdata.har ;
file ydata = ydata.har ;
display file = mmrf.dsp ;
!
updated file mdata = mdata.upd ;
updated file ndata = ndata.upd ;
updated file pdata = pdata.upd ;
updated file ydata = ydata.upd ;
updated terminal data = term.upd;
!
! Equations file
!
equations file = mmrf ;
model = mrf ;
!
```

```
solution file = mmrff1;
method = euler;
steps = 1 2 4;
extrapolation accuracy file = yes;
! Closure
!
! Main macro variables
exogenous NATxi3   ! CPI
      NATr_tot  ! average rate of return on capital
      NATtot    ! terms of trade
!
! Taxes
      Deltax    ! p-point change in General Sales Tax rate
      rK      ! percentage change in income tax rate
      Powtaxm   ! percentage change in the power of the tariff
      rPR     ! percentage change in the payroll tax rate
!
! State and industry variables
      a1lab    ! labour-saving tech. change
      alprim    ! primary factor tech. change
      n      ! agricultural land
      pm      ! foreign price of imports
      miscf001 ; ! average prop. to consume in each State
!
! The following are a list of shift variables for the CGE core and
! the State accounts modules; except the wage shifters which are included
! in the labour-market section below
exogenous NATf5gen ! Overall shift term for state gov. consumption
      NATf6gen ! Overall shift term in fed. govt
         f5gen floct f6a f6gen
         fep  feq  f5a frPRi
         delf_rate
         softf001
      softf002  softf003
      softf004 softf005 softf006  softf007 softf011
      hhldf001 hhldf002 hhldf003 miscf002
      f_rate_xx;
!
! State and federal revenues and expenditure
!
! Income (only to federal, not to states)
exogenous softy111 1-8 ! Income taxes
      softy121 1-8 ! Tariff revenue
!
! Income (only to states, not to federal)
      softy124 9   ! Property taxes
      softy125 9   ! Land taxes
      softy140 9   ! Commonwealth grants to states
      softy141 9   ! Current grants to states
      softy142 9;   ! Capital grants to state
!
! Expenditure (by federal, not by states)
exogenous softq210 1-8 ! Unemployment benefits
      softq500 1-8 ! Commonwealth grants from federal
      softq510 1-8 ! Current grants from federal
      softq520 1-8; ! Capital grants from federal
!
! Govt. deficit is endogenous
exogenous softy330  ; ! Other financing transactions
!
! Fudge factors in K- and F.debt-accumulation equations
exogenous delkfudge delDfudge delunity;
!
! Labour market closure for hybrid simulation
! ABS forecasts
exogenous del_G     ! Change in regional pop. due to nat. growth
         del_FM    ! Change in reg.pop. due to foreign migration
```



```
      to map regional production variables to government finance accounts or
      to determine a government finance variable.
      For example "other commodity taxes" are determined by the gross regional
      production equations and are reported as an income component in the
      government finance statistics. Similarly, "nominal regional domestic
      product" is determined by the regional production equation and is used to
      compute "commonwealth grants" in the government finance module.
!

E_tir # Commodity taxes less subsidies (excl. tariffs) #
                (all,q,REGDEST)
ti_r(q) = dompy320(q);

E_ti # Commodity taxes less subsidies (excl. tariffs) #
C_TI * tI = sum(q,REGDEST, C_DOMPY320(q) * dompy320(q));

E_yn_r # Nominal regional domestic product #
                (all,q,REGDEST)
yn_r(q) = dompq000(q);

E_yn # Nominal GDP #
C_YN * yN = sum(q,REGDEST, C_DOMPQ000(q) * dompq000(q));
E_xiy_r # GDP deflator #
                (all,q,REGDEST)
C_DOMPQ000(q) * xiy_r(q) =  C_DOMPQ110(q) * xi3(q)
                          + C_DOMPQ120(q) * xi2(q)
                          + C_DOMPQ130(q) * xi5(q)
                          + C_DOMPQ140(q) * xi6(q)
                          + C_DOMPQ150(q) * xi2(q)
                          + C_DOMPQ210(q) * psexp(q)
                          - C_DOMPQ220(q) * psimp(q)
                          + C_DOMPQ310(q) * xi4(q)
                          - C_DOMPQ320(q) * xim(q);

E_xiy # GDP deflator #
C_YN * xiy = sum(q,REGDEST, C_DOMPQ000(q) * xiy_r(q));

E_yr_r # Real regional domestic product #
                (all,q,REGDEST)
yr_r(q) = yn_r(q) - xiy_r(q);

E_yr # Real GDP #
yr = yn - xiy;

E_yf # GDP at factor cost #
C_YF * yf =  sum(q,REGDEST, C_DOMPY100(q) * dompy100(q))
           + sum(q,REGDEST, C_DOMPY200(q) * dompy200(q));

E_bstar # Balance of trade surplus to GDP: percentage-point change #
C_YN * bstar = sum(q,REGDEST,C_DOMPQ300(q) * dompq300(q)) - NATBT * yn;

E_ty # Income taxes #
C_TY * tY =   sum(q,REGDEST, C_DOMPY120(q) * dompy120(q))
            + sum(q,REGDEST, C_DOMPY220(q) * dompy220(q));

E_yl # Pre-tax wage income #
C_YL * yl = sum(q,REGDEST, C_DOMPY100(q) * dompy100(q));

E_wn # Nominal pre-tax wage rate #
wn = yl - natl;

E_ylstar # Post-tax wage income #
C_YLSTAR * ylstar = sum(q,REGDEST, C_DOMPY110(q) * dompy110(q));

E_wnstar# Nominal post-tax wage rate #
wnstar= ylstar - natl;

E_wrstar # Real post-tax wage rate #
wrstar = wnstar- natxi3;
```

! Subsection 3.8.4: Income and expenditure compents of government finances !

```
! Summary of financial transactions !
```

```
! Note most of the expenditure components are determined in the core model,
  such as household consumption c(q), and are mapped to the regional production
  account arrays !
E_dompq110 # Private consumption #
             (all,q,REGDEST)
dompq110(q) = c(q);

E_dompq120 # Private investment #
             (all,q,REGDEST)
dompq120(q) = in(q);

E_dompq130 # Government consumption - regions #
             (all,q,REGDEST)
dompq130(q) = othnom5(q);

E_domq140 # Government consumption - federal #
             (all,q,REGDEST)
dompq140(q) = othnom6(q);

E_dompq150 # Government investment #
             (all,q,REGDEST)
dompq150(q) = in(q);

E_dompq100 # Domestic absorption #
             (all,q,REGDEST)
C_DOMPQ100(q) * dompq100(q) =  C_DOMPQ110(q) * dompq110(q)
                             + C_DOMPQ120(q) * dompq120(q)
                             + C_DOMPQ130(q) * dompq130(q)
                             + C_DOMPQ140(q) * dompq140(q)
                             + C_DOMPQ150(q) * dompq150(q);

E_dompq210 # Inter-regional exports #
             (all,q,REGDEST)
dompq210(q) = psexp(q) + xsexp(q);

E_dompq220 # Inter-regional imports #
             (all,q,REGDEST)
dompq220(q) = psimp(q) + xsimp(q);

E_dompq200 # Inter-regional trade balance #
             (all,q,REGDEST)
C_DOMPQ200(q) * dompq200(q) =   C_DOMPQ210(q) * dompq210(q)
                              - C_DOMPQ220(q) * dompq220(q);

E_dompq310 # International exports #
             (all,q,REGDEST)
dompq310(q) = export(q) + natphi;

E_dompq320 # International imports #
             (all,q,REGDEST)
dompq320(q) = imp(q) + natphi;

E_dompq300 # International trade balance #
             (all,q,REGDEST)
C_DOMPQ300(q) * dompq300(q) =   C_DOMPQ310(q) * dompq310(q)
                              - C_DOMPQ320(q) * dompq320(q);

E_dompq000 # GDP at market prices (expenditure side) #
             (all,q,REGDEST)
C_DOMPQ000(q) * dompq000(q) =   C_DOMPQ100(q) * dompq100(q)
                              + C_DOMPQ200(q) * dompq200(q)
                              + C_DOMPQ300(q) * dompq300(q);
```

! Subsection 3.8.3: Intermediary variables !

```
! This section documents equations and variables which are intermediary
  between the regional production and government finance equations. There are
  two types of equations which determine the intermediary variables:
(i)  Aggregating equations. The regional production equations documented above
     (section 3.7.2) determine regional macros, such as regional product.
     The equations of this subsection compute national macros by aggregating the
     regional macro values. These are used in the government finance module.
(ii) mapping equations. These equations compute the intermediary variables
```



```
        del_RM_0 1-7; ! migration to ACT endogenous
!
! Unemployment and wages
! National real wage endog.
exogenous Natfwage fwagei ;
! Regional real wage differentials
! next ties real wage in ACT to national real wage
exogenous wage_diff 8 ;
! unemployment exogenous
!
exogenous del_Unr ;
!
! Part. rate exogenous
exogenous PR ;
!
! Labour market and reg. pop shifters and fudge factors
exogenous f_l f_Wpop f_Pop f_qhous
         delrpfudge delf_RM ;
!
!The follwing variables are from linked-simulation module for hybrid scenario
exogenous del_GT 1-7 del_fGT 8 ! ABS for all natural growth except ACT
        del_FMT            ! ABS for all foreign migration
        del_RMT0 1-7 del_fpop1T 8  ! ABS for all reg mig except ACT
    delpopfudgeT ;
!
rest endogenous ;
!
! Shocks
!
shock delKfudge = 1.00                    ;
shock delDfudge = 1.00                    ;
shock delrpfudge = uniform 1.00           ;
shock delunity = 1.00                     ;
shock delpopfudgeT = 1.00                 ;
shock del_G = file del_ng1.shk            ;
shock del_FM = file del_FM1.shk           ;
shock del_RM_0 = file del_RM1H.shk        ;
shock f_Wpop = file f_RWpop1.shk          ;
shock del_Unr = file del_RUr1.shk         ;
shock a1prim = file a1prim1.shk           ;
shock Natxi3 = 18.565                     ;
shock powtaxm 3 = -5.85                   ;
shock pm = uniform 14.0                   ;
shock NATtot = -0.6065                    ;
shock del_GT = file delTNG1H.shk          ;
shock del_FMT = file delTFM1.shk          ;
shock del_RMT0 = file delTRM1H.shk        ;
!
! Verbal description
!
verbal description = MMRF forecast: hybrid forecast (H)     ;
!
! End of Command file
```

REFERENCES

Han, S (1992) 'A Multi-regional Input Output Data Base For the Australian Economy 1986-87', *COPS Discussion Paper No. D161,* Monash University, 60 pp.

Harrison, J. and K. Pearson (1994a) 'An Introduction to GEMPACK', *GEMPACK User Documentation GPD-1, Release 5.1., Volume 1*, Centre of Policy Studies and the IMPACT Project, Monash University, Clayton.

Harrison, J. and K. Pearson (1994b) 'User's Guide to TABLO, GEMSIM and TABLO-generated Programs' *GEMPACK User Documentation GPD-2, Release 5.1., Volume 2*, Centre of Policy Studies and the IMPACT Project, Monash University, Clayton.

Harrison, J. and K. Pearson (1994c) 'How to Create and Modify GEMPACK Header Array Files Using the Program MODHAR', *GEMPACK User Documentation GPD-3, Release 5.1. Volume 2* Centre of Policy Studies and the IMPACT Project, Monash University, Clayton.

Meagher, G.A. (1993) 'Fiscal Equalisation and the Distribution of Income', paper presented at the Conference of Economists, Perth, 50pp.

Meagher, G.A. (1994) 'Notes on the Construction of the 1986-87 Input-Output Database for the MONASH-MR Model', *Research Memorandum 94/01*, Centre of Policy Studies, Monash University, Clayton.

Meagher, G.A. and B. Parmenter (1993) 'MONASH-MR: A Multiregional CGE Model of Australia", a paper presented to the Workshop on Transportation and Spatial CGE Models, Venice, 24 pp.


```
E_zt_r # Production taxes - regions #
                (all,q,REGDEST)
C_ZT_R(q) * zt_r(q) =  C_Z03_R(q) * z03_r(q) + C_Z05_R(q) * z05_r(q)
                     + C_Z07_R(q) * z07_r(q) + C_Z09_R(q) * z09_r(q);

E_rpr # Payroll tax adjustment #
                (all,j,IND) (all,q,REGDEST)
C_Z03_I_R(j,q) * rpri(j,q) = (C_Z01_I_R(j,q) + C_Z02_I_R(j,q) + C_Z03_I_R(j,q)
                                + TINY) * arpri(j,q);

E_rpri # Setting of payroll tax rates #
                (all,j,IND) (all,q,REGDEST)
rpri(j,q) = rpr(q) + frPRi(j,q);

E_xisfb2 # Price index : sales by final buyers #
                (all,q,REGDEST)
C_Z10_R(q) * xisfb(q) = sum(j,IND, C_Z10_I_R(j,q) * ploct(j,q));
! Subsection 3.8.2: Gross regional product and its components !

! Domestic production account !
E_dompy100 # Wages, salaries and supplements #
                (all,q,REGDEST)
dompy100(q) = z01_r(q);

E_dompy120 # PAYE taxes #
                (all,q,REGDEST)
dompy120(q) = dompy100(q) + rl;

E_dompy110 # Disposable wage income (residual) #
                (all,q,REGDEST)
C_DOMPY100(q) * dompy100(q) =  C_DOMPY110(q) * dompy110(q)
                             + C_DOMPY120(q) * dompy120(q);

E_E_dompy200 # Non-wage primary factor income #
                (all,q,REGDEST)
dompy200(q) = zg_r(q);

E_dompy220 # Taxes on non-wage primary factor income #
                (all,q,REGDEST)
dompy220(q) = dompy200(q) + rk;

E_dompy210 # Disposable non-wage primary factor income (residual) #
                (all,q,REGDEST)
C_DOMPY200(q) * dompy200(q) =   C_DOMPY210(q) * dompy210(q)
                              + C_DOMPY220(q) * dompy220(q);

E_dompy310 # Tariff revenue #
                (all,q,REGDEST)
dompy310(q) = taxrevm(q);

E_dompy320 # Other commodity taxes less subsidies #
                (all,q,REGDEST)
C_DOMPY320(q) * dompy320(q) = AGGTAX1(q) * taxrev1(q) + AGGTAX2(q) * taxrev2(q)
                            + AGGTAX3(q) * taxrev3(q) + AGGTAX4(q) * taxrev4(q)
                            + AGGTAX5(q) * taxrev5(q) + AGGTAX6(q) * taxrev6(q);
! Note that there are no commodity taxes paid by users 5 & 6 in the database !

E_dompy330 # Production taxes #
                (all,q,REGDEST)
dompy330(q) = zt_r(q);

E_dompy300 # Indirect taxes less subsidies #
                (all,q,REGDEST)
C_DOMPY300(q) * dompy300(q) =  C_DOMPY310(q) * dompy310(q)
                             + C_DOMPY320(q) * dompy320(q)
                             + C_DOMPY330(q) * dompy330(q);

E_dompy000 # GDP at market prices (income side) - regions #
                (all,q,REGDEST)
C_DOMPY000(q) * dompy000(q) =   C_DOMPY100(q) * dompy100(q)
                              + C_DOMPY200(q) * dompy200(q)
                              + C_DOMPY300(q) * dompy300(q);
```

```
! Equations in this module are divided into 6 parts !
```

! Subsection 3.8.1: Value-added components of regional product !

```
! Primary factor incomes !
EQUATION
E_z01_r # Wages, salaries and supplements - regions #
                (all,q,REGDEST)
C_Z01_R(q) * z01_r(q)
                  = sum(j,IND, C_Z01_I_R(j,q) * (labind(j,q) + pwagei(j,q)));

E_z02_r # Imputed wages - regions #
                (all,q,REGDEST)
C_Z02_R(q) * z02_r(q)
                  = sum(j,IND, C_Z02_I_R(j,q) * (labind(j,q) + pwagei(j,q)));

E_z03_r # Payroll taxes - regions #
                (all,q,REGDEST)
C_Z03_R(q) * z03_r(q) = sum(j,IND, C_Z03_I_R(j,q) * (rpri(j,q)
                           + labind(j,q) + pwagei(j,q)));

E_z03 # Payroll taxes - national #
C_Z03 * z03 = sum(q,regdest, C_Z03_R(q) * z03_r(q));

E_z04_r # Returns to fixed capital - regions #
                (all,q,REGDEST)
C_Z04_R(q) * z04_r(q)
                = sum(j,IND, C_Z04_I_R(j,q) * (curcap(j,q) + plcap(j,q)));

E_z05_r # Property taxes - regions #
                (all,q,REGDEST)
C_Z05_R(q) * z05_r(q)
                = sum(j,IND, C_Z05_I_R(j,q) * (curcap(j,q) + plcap(j,q)));

E_z05  # Property taxes - national #
C_Z05 * z05 = sum(q,regdest, C_Z05_R(q) * z05_r(q));

E_z06_r # Returns to agricultural land - regions #
                (all,q,REGDEST)
C_Z06_R(q) * z06_r(q) = sum(j,IND, C_Z06_I_R(j,q) * (n(j,q) + plland(j,q)));

E_z07_r # Land taxes - regions #
                (all,q,REGDEST)
C_Z07_R(q) * z07_r(q) = sum(j,IND, C_Z07_I_R(j,q) * (n(j,q) + plland(j,q)));

E_z07 # Land taxes - national #
C_Z07 * z07 = sum(q,regdest, C_Z07_R(q) * z07_r(q));

E_z08_r # Returns to working capital - regions #
                (all,q,REGDEST)
C_Z08_R(q) * z08_r(q) = sum(j,IND, C_Z08_I_R(j,q) * (xloct(j,q) + ploct(j,q)));

E_z09_r # Other indirect taxes - regions #
                (all,q,REGDEST)
C_Z09_R(q) * z09_r(q) = sum(j,IND, C_Z09_I_R(j,q) * (xloct(j,q) + ploct(j,q)));

E_z09 # Other indirect taxes - national #
C_Z09 * z09 = sum(q,regdest, C_Z09_R(q) * z09_r(q));

E_z10_r # Sales by final buyers - regions #
                (all,q,REGDEST)
C_Z10_R(q) * z10_r(q) = sum(j,IND, C_Z10_I_R(j,q) * (xloct(j,q) + ploct(j,q)));

E_z10 # Sales by final buyers - national #
C_Z10 * z10 = sum(q,regdest, C_Z10_R(q) * z10_r(q));

! GOS is defined as sum of imputed wages and returns to fixed capital,
  working capital and agricultural land !
E_zg_r # Gross operating surplus - regions #
                (all,q,REGDEST)
C_ZG_R(q) * zg_r(q) =  C_Z02_R(q) * z02_r(q) + C_Z04_R(q) * z04_r(q)
                     + C_Z06_R(q) * z06_r(q) + C_Z08_R(q) * z08_r(q);
```


Appendix 3 An Annotated TABLO File for MONASH-MRF

CONTENTS

```
3: GOVERNMENT FINANCE MODULE
  3.1: Naming system in the government finance module
  3.2: Variable declarations for the government finance module
  3.3: Coefficient declarations for the government finance module
  3.4: Read statements for the government finance module
  3.5: Update statements for the government finance module
  3.6: Formulae for the government finance module
  3.7: Coeffs. common to the govt. finance and labour market & reg. mig. modules
  3.8: Equations of the government finance module
    3.8.1: Value-added components of regional product
    3.8.2: Gross regional product and its components
    3.8.3: Intermediary variables
    3.8.4: Income and expenditure components of government finances
    3.8.5: Household income and its components
    3.8.6: Miscellaneous equations of the government finance module
4: CAPITAL ACCUMULATION AND INVESTMENT MODULE
  4.1: Variable declarations specific to the cap. acc. and investment module
  4.2: Coefficient declarations for the capital accumu. and investment module
  4.3: Reads, updates and formulae for the capital acc. and investment module
  4.4: Equations of the capital accumulation and investment module
5: FOREIGN DEBT ACCUMULATION MODULE
  5.1: Sets and variables for the foreign-debt accumulation module
  5.2: Coefficient declarations for the foreign-debt accumulation
  5.3: Read and update statements for the foreign-debt accumulation module
  5.4: Formulae and displays for the foreign-debt accumlation module
  5.5: Equations of the foreign-debt accumulation module
6: LABOUR MARKET & REGIONAL MIGRATION MODULE
  6.1: Variable declarations for the labour market & regional migration module
  6.2: Coefficient declarations for the labour market & reg. migration module
  6.3: Reads and updates for the labour market & regional migration module
  6.4: Formulae for the labour market & regional migration module
  6.5: Equations of the labour market & regional migration module
!
```



```
C_PBP = sum(q,DOMDEST, C_PBP_R(q))    ;
C_SUBSIDIES = sum(q,DOMDEST, C_SOFTQ300(q));
C_TI  = sum(q,REGDEST, C_DOMPY320(q));
C_TY  = sum(q,REGDEST, C_DOMPY120(q)) + sum(q,REGDEST, C_DOMPY220(q));
C_UPB = sum(q,REGDEST, C_HHLDY210(q));
C_YF  = sum(q,REGDEST, C_DOMPY100(q)) + sum(q,REGDEST, C_DOMPY200(q));
C_YL  = sum(q,REGDEST, C_DOMPY100(q));
C_YLSTAR    = sum(q,REGDEST, C_DOMPY110(q));
C_YN  = sum(q,REGDEST, C_DOMPY000(q));
                (all,q,REGDEST)
C_Z01_R(q) = sum(j,IND, C_Z01_I_R(j,q));
                (all,q,REGDEST)
C_Z02_R(q) = sum(j,IND, C_Z02_I_R(j,q));
                (all,q,REGDEST)
C_Z03_R(q) = sum(j,IND, C_Z03_I_R(j,q));
                (all,q,REGDEST)
C_Z04_R(q) = sum(j,IND, C_Z04_I_R(j,q));
                (all,q,REGDEST)
C_Z05_R(q) = sum(j,IND, C_Z05_I_R(j,q));
                (all,q,REGDEST)
C_Z06_R(q) = sum(j,IND, C_Z06_I_R(j,q));
                (all,q,REGDEST)
C_Z07_R(q) = sum(j,IND, C_Z07_I_R(j,q));
                (all,q,REGDEST)
C_Z08_R(q) = sum(j,IND, C_Z08_I_R(j,q));
                (all,q,REGDEST)
C_Z09_R(q) = sum(j,IND, C_Z09_I_R(j,q));
                (all,q,REGDEST)
C_Z10_R(q) = sum(j,IND, C_Z10_I_R(j,q));
                (all,q,REGDEST)
C_ZG_R(q) = C_Z02_R(q) + C_Z04_R(q) + C_Z06_R(q) + C_Z08_R(q);
                (all,q,REGDEST)
C_ZT_R(q) = C_Z03_R(q) + C_Z05_R(q) + C_Z07_R(q) + C_Z09_R(q);
C_Z01 = sum(q,REGDEST, C_Z01_R(q));
C_Z02 = sum(q,REGDEST, C_Z02_R(q));
C_Z03 = sum(q,REGDEST, C_Z03_R(q));
C_Z04 = sum(q,REGDEST, C_Z04_R(q));
C_Z05 = sum(q,REGDEST, C_Z05_R(q));
C_Z06 = sum(q,REGDEST, C_Z06_R(q));
C_Z07 = sum(q,REGDEST, C_Z07_R(q));
C_Z08 = sum(q,REGDEST, C_Z08_R(q));
C_Z09 = sum(q,REGDEST, C_Z09_R(q));
C_Z10 = sum(q,REGDEST, C_Z10_R(q));
C_ZG = C_Z02 + C_Z04 + C_Z06 + C_Z08;
C_ZT = C_Z03 + C_Z05 + C_Z07 + C_Z09;
NATBT = sum(q,regdest, C_DOMPQ300(q));
```

! Section 3.7: Coeffs. common to govt. finance and labour market & reg. mig. modules !

```
! The following coefficients are used in the government finance and labour
  market & population modules. We read them here and update them in the labour
  market & population module after their corresponding variables are defined. !

COEFFICIENT
                (all,q,REGDEST)
C_LABSUP(q)    # regional labour supply #;
                (all,q,REGDEST)
C_EMPLOY(q)    # regional employment: persons #;

READ
C_LABSUP from file PDATA header "RLBS";
C_EMPLOY from file PDATA header "REMP";

FORMULA
                (all,q,DOMDEST)
C_DEBTG(q) = (C_SOFTQ400(q) - C_SOFTY130(q))/R_DEBTG;
                (all,q,REGDEST)
C_HHLDD001(q) = C_LABSUP(q)/(C_LABSUP(q)-C_EMPLOY(q));
                (all,q,REGDEST)
C_HHLDD002(q) = C_EMPLOY(q)/(C_LABSUP(q)-C_EMPLOY(q));
```

! Section 3.8: Equations of the government finance module !

```
C_SOFTQ520(q) = softq520(q);
                (all,q,DOMDEST)
C_SOFTQ600(q) = softq600(q);
                (all,q,DOMDEST)
C_SOFTY000(q) = softy000(q);
                (all,q,DOMDEST)
C_SOFTY100(q) = softy100(q);
                (all,q,DOMDEST)
C_SOFTY110(q) = softy110(q);
                (all,q,DOMDEST)
C_SOFTY111(q) = softy111(q);
                (all,q,DOMDEST)
C_SOFTY112(q) = softy112(q);
                (all,q,DOMDEST)
C_SOFTY120(q) = softy120(q);
                (all,q,DOMDEST)
C_SOFTY121(q) = softy121(q);
                (all,q,DOMDEST)
C_SOFTY122(q) = softy122(q);
                (all,q,DOMDEST)
C_SOFTY123(q) = softy123(q);
                (all,q,DOMDEST)
C_SOFTY124(q) = softy124(q);
                (all,q,DOMDEST)
C_SOFTY125(q) = softy125(q);
                (all,q,DOMDEST)
C_SOFTY126(q) = softy126(q);
                (all,q,DOMDEST)
C_SOFTY130(q) = softy130(q);
                (all,q,DOMDEST)
C_SOFTY140(q) = softy140(q);
                (all,q,DOMDEST)
C_SOFTY141(q) = softy141(q);
                (all,q,DOMDEST)
C_SOFTY142(q) = softy142(q);
                (all,q,DOMDEST)
C_SOFTY150(q) = softy150(q);
                (all,q,DOMDEST)
C_SOFTY200(q) = softy200(q);
                (all,q,DOMDEST)
C_SOFTY300(q) = softy300(q);
                (all,q,DOMDEST)
C_SOFTY310(q) = softy310(q);
                (all,q,DOMDEST)
C_SOFTY320(q) = softy320(q);
                (all,q,DOMDEST)
C_SOFTY330(q) = softy330(q);
                (all,q,REGDEST) (all,j,IND)
C_Z01_I_R(j,q) = labind(j,q) * pwagei(j,q);
                (all,q,REGDEST) (all,j,IND)
C_Z02_I_R(j,q) = labind(j,q) * pwagei(j,q);
                (all,q,REGDEST) (all,j,IND)
C_Z03_I_R(j,q) = rpri(j,q) * labind(j,q) * pwagei(j,q);
                (all,q,REGDEST)  (all,j,IND)
C_Z04_I_R(j,q) = curcap(j,q) * plcap(j,q);
                (all,q,REGDEST) (all,j,IND)
C_Z05_I_R(j,q) = curcap(j,q) * plcap(j,q);
                (all,q,REGDEST) (all,j,IND)
C_Z06_I_R(j,q) = n(j,q) * plland(j,q);
                (all,q,REGDEST) (all,j,IND)
C_Z07_I_R(j,q) = n(j,q) * plland(j,q);
                (all,q,REGDEST) (all,j,IND)
C_Z08_I_R(j,q) = xloct(j,q) * ploct(j,q);
                (all,q,REGDEST) (all,j,IND)
C_Z09_I_R(j,q) = xloct(j,q) * ploct(j,q);
                (all,q,REGDEST) (all,j,IND)
C_Z10_I_R(j,q) = xloct(j,q) * ploct(j,q);
```

! Section 3.6: Formulae for the government finance module !

```
FORMULA
C_IG  = sum(q,DOMDEST, C_IG_R(q))    ;
C_IP  = sum(q,REGDEST, C_IP_R(q))    ;
```


! Part 1: PRELIMINARIES !

! Section 1.1: MONASH Multi Regional Forecasting Model (MONASH-MRF) !

```
! MONASH-MRF is a computable general equilibrium (CGE) regional model of the
  Australian economy. The model recognises eight regions, comprising the six
  States and two Territories.

  In this tab file, the model's equations are presented in five modules:
   (i)   the CGE core module
   (ii)  the government finance module
   (iii) the capital and investment module
   (iv)  the debt accumulation module
   (v)   the labour market & regional migration module

  The CGE core module: consists of the equations and variables of the original
  ORANI model with a regional subscript added (see Dixon, Parmenter, Sutton and
  Vincent (DPSV), 1982 for the original version of ORANI). This module is
  separated into four main equation blocks determining: (a) consumer demands
  (b) producer and consumer prices (c) market clearing conditions
  (d) macroeconomic variables as summations of microeconomic variables.

  The government finance module incorporates equations determining: gross
  products of each region from the income and expenditure sides; sources of
  income and various expediture accounts for regional and federal governments
  as defined in the State Finance Statistics of the Australian Bureau of
  Statistics (ABS).

  The capital & investment and debt accumulation modules are added to
  endogenise (a) changes in total investment and capital stock over a forecast
  period, and (b) the accumulation of foreign debt. The capital accumulation
  section  of the capital & investment module is based on one of three
  alternative treatments implemented in the MONASH model. The entire debt
  accumulation module is based on ORANI-F (see Horridge, Parmenter and
  Pearson, 1993)

  The labour market & regional migration module defines equations
  determining regional population by taking into account (1) natural growth
  (2) inter-regional migration and (3) foreign migration. Regional labour
  supply is linked to regional population via accounting identities which
  allow for shifts in the relationship between regional population and the
  regional population of working age and the workforce-participation rate.
  The module also includes equations defining changes in regional unemployment
  rates.

Parts 2 to 6 of this TABLO file present the five modules in the order outlined
in Part 1.
!
```

! Part 2: MONASH-MRF CGE core module !

```
! First version by J.M. Horridge, July 1992
! The ORANI framework has been used to define the core module of MONASH-MRF with
  the addition of regional subscripts to all the equations and variables. In
  addition, we made the following changes to the original ORANI framwork:

* The Armington assumption has become 2 CES nests.  Industry, capital
  creators and households now combine goods from the eight regions to form an
  Australian composite. Imports are combined with the Australian composite
  to form an all-source composite.

* Each region exports independently to the Rest of the World (ROW). Aggregate
  demand for, and the price of, Australian exports have no behavioural
  content.

* 'Other' demands have been split by region; there is also a FEDERAL other
  demand by region which represents the Commonwealth government's expenditure in
  each region.

* A comprehensive set of national addups have been specified. The original ORANI
  variable's names now have a regional subscript. The orignal name, prefixed by
  "Nat", gives the national addup.
```

```
* Margins on goods used by industry, capital creators, regional governments
  and households are assumed to be produced at the point of consumption.
  Margins on exports are assumed to be produced at the point of production.

* Import tariff rates are assumed to be uniform over regions.

* Single product industries are assumed.

* CES substitution possibilities between different labour types, rather
  than CRESH.

* Not all the technical changes in ORANI are present here (they could be
  added as required).

* Tax changes are expressed in two ways:
    1. Percentage-point change in ad valorem tax rate. All such variable
       names have prefix of deltax. Thus deltax = 5 means the taxrate went
       from 20 to 25 percent, or from 24 to 29 percent. All such tax variables
       are in the CGE core module.
    2. Percent change in the tax rate. For example, income-tax rate = 25 means
       that if the intial tax rate was 40 per cent, then it has increased to
       50 per cent, i.e., 100*(50-40)/40 = 25. All such tax variables are
       in the government finance module.

* Tariff changes are expressed as percentage change in power of the tariff.
   The power of the tariff is defined as one plus the tariff rate. If the tariff
   rate is 20 percent, the power of tariff is 1.20. If the tariff rate is
   increased form 20 percent to 25 percent, percentage change in the power of
   the tariff is 4, i.e., 100*(1.25-1.20)/1.20 = 4.
!
```

! Section 2.1: Files for ALL modules !

```
FILE MDATA
  # File containing input-output tables and other data for the CGE core module#;
FILE YDATA
  # File containing data on debt, investment and the asset value of capital#;
FILE NDATA
  # File containing income and expenditure components of the State Fin. Stats.#;
FILE PDATA
  # File containing data on population & labour market #;
FILE (NEW,TEXT) DISFILE
  # Text output file for examining database values #;
```

! Section 2.2: Set and subset names common to ALL modules !

```
SET IND # Industries # (Agricult,Mining,Manufact,Publ_Util,Construct,Dom_Trade,
                        Transp_Comm,Finance,Housing,Publ_Srv,Comm_Srv,Pers_Srv,
                        NC_Imports);
SET MARGIND # Margin industries # (Dom_Trade,Transp_Comm);
SUBSET MARGIND is subset of IND;
SET NONMARGIND # NonMargin industries # (Agricult,Mining,Manufact,Publ_Util,
                                         Construct,Finance,Housing,Publ_Srv,
                                         Comm_Srv,Pers_Srv,NC_Imports);
SUBSET NONMARGIND is subset of IND;
SET COM # Commodities # (Agricult,Mining,Manufact,Publ_Util,Construct,Dom_Trade,
                         Transp_Comm,Finance,Housing,Publ_Srv,Comm_Srv,
                         Pers_Srv,NC_Imports);
SET MARGCOM # Margin Commodities # (Dom_Trade,Transp_Comm);
SUBSET MARGCOM is subset of COM;
SET NONMARGCOM # NonMargin Commodities # (Agricult,Mining,Manufact,Publ_Util,
    Construct,Finance,Housing,Publ_Srv,Comm_Srv,Pers_srv,NC_imports);
SUBSET NONMARGCOM is subset of COM;
SET FAC # Primary Factors # (labour,capital,land);
SET OCC # Occupation Types # SIZE 8;
SET ALLDEST # Destination of Goods #
              (NSW,VIC,QLD,SA,WA,TAS,NT,ACT,federal,foreign);
SET DOMDEST # Destination of Goods #
              (NSW,VIC,QLD,SA,WA,TAS,NT,ACT,federal);
SET REGDEST # 8 regional destinations #
              (NSW,VIC,QLD,SA,WA,TAS,NT,ACT);
SUBSET DOMDEST is subset of ALLDEST;
```



```
                (all,q,REGDEST)
C_DOMPY100(q) = dompy100(q);
                (all,q,REGDEST)
C_DOMPY110(q) = dompy110(q);
                (all,q,REGDEST)
C_DOMPY120(q) = dompy120(q);
                (all,q,REGDEST)
C_DOMPY200(q) = dompy200(q);
                (all,q,REGDEST)
C_DOMPY210(q) = dompy210(q);
                (all,q,REGDEST)
C_DOMPY220(q) = dompy220(q);
                (all,q,REGDEST)
C_DOMPY300(q) = dompy300(q);
                (all,q,REGDEST)
C_DOMPY310(q) = dompy310(q);
                (all,q,REGDEST)
C_DOMPY320(q) = dompy320(q);
                (all,q,REGDEST)
C_DOMPY330(q) = dompy330(q);
                (all,q,REGDEST)
C_HHLDY000(q) = hhldy000(q);
                (all,q,REGDEST)
C_HHLDY100(q) = hhldy100(q);
                (all,q,REGDEST)
C_HHLDY110(q) = hhldy110(q);
                (all,q,REGDEST)
C_HHLDY120(q) = hhldy120(q);
                (all,q,REGDEST)
C_HHLDY200(q) = hhldy200(q);
                (all,q,REGDEST)
C_HHLDY210(q) = hhldy210(q);
                (all,q,REGDEST)
C_HHLDY220(q) = hhldy220(q);
                (all,q,REGDEST)
C_HHLDY300(q) = hhldy300(q);
                (all,q,REGDEST)
C_HHLDY400(q) = hhldy400(q);
                (all,q,REGDEST)
C_HHLDY410(q) = hhldy410(q);
                (all,q,REGDEST)
C_HHLDY420(q) = hhldy420(q);
                (all,q,REGDEST)
C_HHLDY430(q) = hhldy430(q);
                (all,q,DOMDEST)
C_IG_R(q) = ig_r(q);
                (all,q,REGDEST)
C_IP_R(q) = in(q);
                (all,q,DOMDEST)
C_PBP_R(q) = pbp_r(q);
                (all,q,DOMDEST)
C_SOFTQ000(q) = softq000(q);
                (all,q,DOMDEST)
C_SOFTQ100(q) = softq100(q);
                (all,q,DOMDEST)
C_SOFTQ110(q) = softq110(q);
                (all,q,DOMDEST)
C_SOFTQ120(q) = softq120(q);
                (all,q,DOMDEST)
C_SOFTQ200(q) = softq200(q);
                (all,q,DOMDEST)
C_SOFTQ210(q) = softq210(q);
                (all,q,DOMDEST)
C_SOFTQ220(q) = softq220(q);
                (all,q,DOMDEST)
C_SOFTQ300(q) = softq300(q);
                (all,q,DOMDEST)
C_SOFTQ400(q) = softq400(q);
                (all,q,DOMDEST)
C_SOFTQ500(q) = softq500(q);
                (all,q,DOMDEST)
C_SOFTQ510(q) = softq510(q);
                (all,q,DOMDEST)
```

```
C_SOFTY126(Q) from file NDATA header "GA12";
              (all,q,DOMDEST)
C_SOFTY130(Q) from file NDATA header "GA13";
              (all,q,DOMDEST)
C_SOFTY140(Q) from file NDATA header "GA14";
              (all,q,DOMDEST)
C_SOFTY141(Q) from file NDATA header "GA15";
              (all,q,DOMDEST)
C_SOFTY142(Q) from file NDATA header "GA16";
              (all,q,DOMDEST)
C_SOFTY150(Q) from file NDATA header "GA17";
              (all,q,DOMDEST)
C_SOFTY200(Q) from file NDATA header "GA18";
              (all,q,DOMDEST)
C_SOFTY300(Q) from file NDATA header "GA19";
              (all,q,DOMDEST)
C_SOFTY310(Q) from file NDATA header "GA20";
              (all,q,DOMDEST)
C_SOFTY320(Q) from file NDATA header "GA21";
              (all,q,DOMDEST)
C_SOFTY330(Q) from file NDATA header "GA22";
              (all,j,IND) (all,q,REGDEST)
C_Z01_I_R(J,Q) from file NDATA header "FZ01";
              (all,j,IND) (all,q,REGDEST)
C_Z02_I_R(J,Q) from file NDATA header "FZ02";
              (all,j,IND) (all,q,REGDEST)
C_Z03_I_R(J,Q) from file NDATA header "FZ03";
              (all,j,IND) (all,q,REGDEST)
 C_Z04_I_R(J,Q) from file NDATA header "FZ04";
              (all,j,IND) (all,q,REGDEST)
C_Z05_I_R(J,Q) from file NDATA header "FZ05";
              (all,j,IND) (all,q,REGDEST)
C_Z06_I_R(J,Q) from file NDATA header "FZ06";
              (all,j,IND) (all,q,REGDEST)
C_Z07_I_R(J,Q) from file NDATA header "FZ07";
              (all,j,IND) (all,q,REGDEST)
C_Z08_I_R(J,Q) from file NDATA header "FZ08";
              (all,j,IND) (all,q,REGDEST)
C_Z09_I_R(J,Q) from file NDATA header "FZ09";
              (all,j,IND) (all,q,REGDEST)
C_Z10_I_R(J,Q) from file NDATA header "FZ10";
```

! Section 3.5: Update statements for the government finance module !

```
UPDATE
              (all,q,REGDEST)
C_DOMPQ000(q) = dompq000(q);
              (all,q,REGDEST)
C_DOMPQ100(q) = dompq100(q);
              (all,q,REGDEST)
C_DOMPQ110(q) = dompq110(q);
              (all,q,REGDEST)
C_DOMPQ120(q) = dompq120(q);
              (all,q,REGDEST)
C_DOMPQ130(q) = dompq130(q);
              (all,q,REGDEST)
C_DOMPQ140(q) = dompq140(q);
              (all,q,REGDEST)
C_DOMPQ150(q) = dompq150(q);
              (all,q,REGDEST)
C_DOMPQ200(q) = dompq200(q);
              (all,q,REGDEST)
C_DOMPQ210(q) = dompq210(q);
              (all,q,REGDEST)
C_DOMPQ220(q) = dompq220(q);
              (all,q,REGDEST)
C_DOMPQ300(q) = dompq300(q);
              (all,q,REGDEST)
C_DOMPQ310(q) = dompq310(q);
              (all,q,REGDEST)
C_DOMPQ320(q) = dompq320(q);
              (all,q,REGDEST)
C_DOMPY000(q) = dompy000(q);
```



```
SUBSET REGDEST is subset of ALLDEST;
SUBSET REGDEST is subset of DOMDEST;
SET ALLSOURCE # Origin of goods # (NSW,VIC,QLD,SA,WA,TAS,NT,ACT,foreign);
SET REGSOURCE # 8 regional sources # (NSW,VIC,QLD,SA,WA,TAS,NT,ACT);
SUBSET REGSOURCE is subset of ALLSOURCE;
SET TWOSOURCE # main sources # (domestic,foreign);
SUBSET MARGCOM is subset of MARGIND;
SUBSET MARGIND is subset of MARGCOM;
SUBSET REGSOURCE is subset of REGDEST;
SUBSET REGDEST is subset of REGSOURCE;
SUBSET COM is subset of IND;
SUBSET IND is subset of COM;
```

! Section 2.3: Variable declarations for the CGE core module !

! Subsection 2.3.1: Naming system for variables in the CGE core module !

```
! The following system indicates the source dimension of prices, quantities,
  shifters, and purchasers' value flows

Names consist of a prefix, a main user number, and a source dimension:

the prefixes are:
x       quantities
p       prices
a       shifters
pval    value of flow at purchasers prices

the main user numbers are:
0       basic                           1           column
1       intermediate inputs             IND * REG   columns
2       inputs to capital creation      IND * REG   columns
3       consumption demands             REG         columns
4       export demands                  1           column
5       regional "Other" demands        REG         columns
6       Fed. govt "Other" demands       REG         columns
The main user numbers also determine the column (destination) dimension, as
indicated.

The source dimensions are:
a       ALLSOURCE: regions 1 to 8 and foreign
r       REGSOURCE: regions 1 to 8
t       TWOSOURCE: domestic composite and foreign
c       CMPSOURCE: domestic composite only
o       ONESOURCE: domestic/foreign  composite

Examples:
x3t     usage by household of domestic composite and foreign
p1a     price to industries from 9 sources
pval2o  costs to capital creators of goods regardless of source
a1r     regional good decrementing tech change to industry
!
```

! Subsection 2.3.2: National macro variables in alphabetical order !

```
VARIABLE
              (change)
deltax1all    # Overall percent-point change in indirect tax rates, user 1 #;
              (change)
deltax2all    # Overall percent-point change in indirect tax rates, user 2 #;
              (change)
deltax3all    # Overall percent-point change in indirect tax rates, user 3 #;
              (change)
deltax4all    # Overall percent-point change in indirect tax rates, user 4 #;
              (change)
deltax5all    # Overall percent-point change in indirect tax rates, user 5 #;
              (change)
deltax6all    # Overall percent-point change in indirect tax rates, user 6 #;
natc          # Nominal total household consumption #;
natcaprev     # Aggregate payments to capital #;
natcR         # Real household consumption #;
              (change)
natdelB       # Ordinary change in balance of trade #;
```

```
natexport     # Foreign-currency value of exports #;
natexpvol     # Export volumes #;
natfep        # Economy-wide shifter of export demand curves #;
natf5gen      # Overall shift term for regional "Other" demands #;
natf6gen      # Overall shift term for federal "Other" demands #;
natfwage      # Overall wage shifter #;
natgdpexp     # Nominal GDP from expenditure side #;
natgdpinc     # Nominal GDP from income side #;
natgdpreal    # Real GDP from expenditure side #;
natimp        # Foreign currency value of imports #;
natimpvol     # Import volumes #;
natin         # Aggregate nominal investment #;
natiR         # Aggregate real investment expenditure #;
natkT         # Aggregate capital stock, rental weights #;
natl          # Aggregate employment, wage bill weights #;
natlabrev     # Aggregate payments to labour #;
natlndrev     # Aggregate payments to land #;
natoctrev     # Aggregate other cost ticket payments #;
natothnom5    # Aggregate nominal value of regional "Other" demands #;
natothnom6    # Aggregate nominal value of federal "Other" demands #;
natothreal5   # Aggregate real regional "Other" demands #;
natothreal6   # Aggregate real federal "Other" demands #;
natplcap      # Aggregate nominal capital rentals #;
natphi        # Exchange rate #;
natpwage      # Aggregate nominal wages#;
natrealwage   # National consumer real wage #;
natrwage_w    # National real wages for workers: deflated by CPI#;
natrwage_p    # National real wages for producers: deflated by GDP deflator#;
nattaxind     # Aggregate revenue from all indirect taxes #;
nattaxrev1    # Aggregate revenue from indirect taxes on intermediate #;
nattaxrev2    # Aggregate revenue from indirect taxes on investment #;
nattaxrev3    # Aggregate revenue from indirect taxes on households #;
nattaxrev4    # Aggregate revenue from indirect taxes on exports #;
nattaxrev5    # Aggregate revenue from indirect taxes on regional "Other" #;
nattaxrev6    # Aggregate revenue from indirect taxes on federal "Other" #;
nattaxrevm    # Aggregate tariff revenue #;
nattot        # Economy-wide terms of trade #;
natxi2        # Investment price index #;
natxi3        # Consumer price index #;
natxi4        # Exports price index #;
natxi5        # Regional "Other" demands price index #;
natxi6        # Fed "Other" demands price index #;
natxigdp      # GDP price index, expenditure side #;
natxim        # Imports price index #;
natximp0      # Duty-paid imports price index #;
natxiplpk     # Relative prices of labour and capital#;
natz_tot      # Aggregate Output: Value-Added Weights #;
```

! Subsection 2.3.3: National vector variables in alphabetical order !

```
              (all,j,IND)
natlabind(j)  # Employment by Industry #;
              (all,m,OCC)
natlambda(m)  # Employment in occupation M #;
              (all,i,COM)
natx0imp(i)   # Import volumes #;
              (all,j,IND)
naty(j)       # Capital creation by using industry #;
              (all,j,IND)
natz(j)       # Activity level or value-added #;
```

! Subsection 2.3.4: Regional macro variables in alphabetical order !

```
              (all,q,REGDEST)
c(q)          # Nominal total household consumption #;
              (all,q,REGDEST)
caprev(q)     # Aggregate payments to capital #;
              (all,q,REGDEST)
cr(q)         # Real household consumption #;
              (all,q,REGDEST)
cr_shr(q)     # Regional/national consumption ratio #;
              (change) (all,q,ALLDEST)
deltaxdest(q) # Reg. tax shifter (percentage-point change) #;
              (change) (all,s,ALLSOURCE)
```



```
C_HHLDY120(Q) from file NDATA header "HA04";
                (all,q,REGDEST)
C_HHLDY200(Q) from file NDATA header "HA05";
                (all,q,REGDEST)
C_HHLDY210(Q) from file NDATA header "HA06";
                (all,q,REGDEST)
C_HHLDY220(Q) from file NDATA header "HA07";
                (all,q,REGDEST)
C_HHLDY300(Q) from file NDATA header "HA08";
                (all,q,REGDEST)
C_HHLDY400(Q) from file NDATA header "HA09";
                (all,q,REGDEST)
C_HHLDY410(Q) from file NDATA header "HA10";
                (all,q,REGDEST)
C_HHLDY420(Q) from file NDATA header "HA11";
                (all,q,REGDEST)
C_HHLDY430(Q) from file NDATA header "HA12";
                (all,q,REGDEST)
C_IP_R(Q)     from file NDATA header "MI02";
                (all,q,DOMDEST)
C_IG_R(Q)     from file NDATA header "MI03";
                (all,q,DOMDEST)
C_PBP_R(Q)    from file NDATA header "MI04";
R_DEBTG       from file NDATA header "RGOV";
                (all,q,DOMDEST)
C_SOFTQ000(Q) from file NDATA header "GA23";
                (all,q,DOMDEST)
C_SOFTQ100(Q) from file NDATA header "GA24";
                (all,q,DOMDEST)
C_SOFTQ110(Q) from file NDATA header "GA25";
                (all,q,DOMDEST)
C_SOFTQ120(Q) from file NDATA header "GA26";
                (all,q,DOMDEST)
C_SOFTQ200(Q) from file NDATA header "GA27";
                (all,q,DOMDEST)
C_SOFTQ210(Q) from file NDATA header "GA28";
                (all,q,DOMDEST)
C_SOFTQ220(Q) from file NDATA header "GA29";
                (all,q,DOMDEST)
C_SOFTQ300(Q) from file NDATA header "GA30";
                (all,q,DOMDEST)
C_SOFTQ400(Q) from file NDATA header "GA31";
                (all,q,DOMDEST)
C_SOFTQ500(Q) from file NDATA header "GA32";
                (all,q,DOMDEST)
C_SOFTQ510(Q) from file NDATA header "GA33";
                (all,q,DOMDEST)
C_SOFTQ520(Q) from file NDATA header "GA34";
                (all,q,DOMDEST)
C_SOFTQ600(Q) from file NDATA header "GA35";
                (all,q,DOMDEST)
C_SOFTY000(Q) from file NDATA header "GA01";
                (all,q,DOMDEST)
C_SOFTY100(Q) from file NDATA header "GA02";
                (all,q,DOMDEST)
C_SOFTY110(Q) from file NDATA header "GA03";
                (all,q,DOMDEST)
C_SOFTY111(Q) from file NDATA header "GA04";
                (all,q,DOMDEST)
C_SOFTY112(Q) from file NDATA header "GA05";
                (all,q,DOMDEST)
C_SOFTY120(Q) from file NDATA header "GA06";
                (all,q,DOMDEST)
C_SOFTY121(Q) from file NDATA header "GA07";
                (all,q,DOMDEST)
C_SOFTY122(Q) from file NDATA header "GA08";
                (all,q,DOMDEST)
C_SOFTY123(Q) from file NDATA header "GA09";
                (all,q,DOMDEST)
C_SOFTY124(Q) from file NDATA header "GA10";
                (all,q,DOMDEST)
C_SOFTY125(Q) from file NDATA header "GA11";
                (all,q,DOMDEST)
```

```
                (all,q,REGDEST)
C_Z06_R(Q)      # Returns to agricultural land by region #;
                (all,q,REGDEST)
C_Z07_R(Q)      # Land taxes by region  #;
                (all,q,REGDEST)
C_Z08_R(Q)      # Returns to working capital by region #;
                (all,q,REGDEST)
C_Z09_R(Q)      # Other indirect taxes by region #;
                (all,q,REGDEST)
C_Z10_R(Q)      # Sales by final buyers by region #;
                (all,q,REGDEST)
C_ZG_R(Q)       # Gross operating surplus by region #;
                (all,q,REGDEST)
C_ZT_R(Q)       # Production taxes by region #;
```

! Section 3.4: Read statements for the government finance module!

```
READ
                (all,q,REGDEST)
C_DOMPQ000(Q) from file NDATA header "PA12";
                (all,q,REGDEST)
C_DOMPQ100(Q) from file NDATA header "PA13";
                (all,q,REGDEST)
C_DOMPQ110(Q) from file NDATA header "PA14";
                (all,q,REGDEST)
C_DOMPQ120(Q) from file NDATA header "PA15";
                (all,q,REGDEST)
C_DOMPQ130(Q) from file NDATA header "PA16";
                (all,q,REGDEST)
C_DOMPQ140(Q) from file NDATA header "PA17";
                (all,q,REGDEST)
C_DOMPQ150(Q) from file NDATA header "PA18";
                (all,q,REGDEST)
C_DOMPQ200(Q) from file NDATA header "PA19";
                (all,q,REGDEST)
C_DOMPQ210(Q) from file NDATA header "PA20";
                (all,q,REGDEST)
C_DOMPQ220(Q) from file NDATA header "PA21";
                (all,q,REGDEST)
C_DOMPQ300(Q) from file NDATA header "PA22";
                (all,q,REGDEST)
C_DOMPQ310(Q) from file NDATA header "PA23";
                (all,q,REGDEST)
C_DOMPQ320(Q) from file NDATA header "PA24";
                (all,q,REGDEST)
C_DOMPY000(Q) from file NDATA header "PA01";
                (all,q,REGDEST)
C_DOMPY100(Q) from file NDATA header "PA02";
                (all,q,REGDEST)
C_DOMPY110(Q) from file NDATA header "PA03";
                (all,q,REGDEST)
C_DOMPY120(Q) from file NDATA header "PA04";
                (all,q,REGDEST)
C_DOMPY200(Q) from file NDATA header "PA05";
                (all,q,REGDEST)
C_DOMPY210(Q) from file NDATA header "PA06";
                (all,q,REGDEST)
C_DOMPY220(Q) from file NDATA header "PA07";
                (all,q,REGDEST)
C_DOMPY300(Q) from file NDATA header "PA08";
                (all,q,REGDEST)
C_DOMPY310(Q) from file NDATA header "PA09";
                (all,q,REGDEST)
C_DOMPY320(Q) from file NDATA header "PA10";
                (all,q,REGDEST)
C_DOMPY330(Q) from file NDATA header "PA11";
                (all,q,REGDEST)
C_HHLDY000(Q) from file NDATA header "HA01";
                (all,q,REGDEST)
C_HHLDY100(Q) from file NDATA header "HA02";
                (all,q,REGDEST)
C_HHLDY110(Q) from file NDATA header "HA03";
                (all,q,REGDEST)
```



```
deltaxsource(s)
                # Reg. tax shifter (percentage-point change) #;
                (all,q,REGDEST)
export(q)       # Foreign currency value of exports #;
                (all,q,REGDEST)
expvoL(q)       # Export volumes #;
                (all,q,REGDEST)
f5gen(q)        # Overall shift term for regional "Other" demands #;
                (all,q,REGDEST)
fwage(q)        # Overall real wage shifter #;
                (all,q,REGDEST)
imp(q)          # Foreign currency value of imports #;
                (all,q,REGDEST)
impvol(q)       # Import volume index #;
                (all,q,REGDEST)
in(q)           # Aggregate nominal investment #;
                (all,q,REGDEST)
ir(q)           # Aggregate real investment expenditure #;
                (all,q,REGDEST)
kt(q)           # Aggregate capital stock, rental weights #;
                (all,q,REGDEST)
labrev(q)       # Aggregate payments to labour #;
                (all,q,REGDEST)
lndrev(q)       # Aggregate payments to land #;
                (all,q,REGDEST)
l(q)            # Aggregate employment- wage bill weights #;
                (all,q,REGDEST)
luxexp(q)       # Total supernumerary household expenditure #;
                (all,q,REGDEST)
octrev(q)       # Aggregate other cost ticket payments #;
                (all,q,REGDEST)
othnom5(q)      # Aggregate nominal regional "Other" demands #;
                (all,q,REGDEST)
othnom6(q)      # Aggregate nominal federal "Other" demand #;
                (all,q,REGDEST)
othreal5(q)     # Aggregate real regional "Other" demands #;
                (all,q,REGDEST)
othreal6(q)     # Aggregate real federal "Other" demand #;
                (all,q,REGDEST)
pwage(q)        # Region-wide nominal wage #;
                (all,s,REGSOURCE)
psexp(s)        # Price indices for inter-regional exports #;
                (all,q,REGDEST)
psimp(q)        # Price indices for inter-regional imports #;
                (all,q,REGDEST)
qhous(q)        # Number of households #;
                (all,q,REGDEST)
reg_plcap(q)    # Regional rental price of capital #;
                (all,q,REGDEST)
realwage_w(q)   # Real wages for workers: deflated by CPI #;
                (all,q,REGDEST)
realwage_p(q)   # Real wages for producers: def. by GDP def. #;
                (all,q,REGDEST)
totdom(q)       # Domestic terms of trade #;
                (all,q,REGDEST)
totfor(q)       # Foreign terms of trade #;
                (all,q,REGDEST)
r0_tot(q)       # Regional aggregate rate of return #;
                (all,q,REGDEST)
taxind(q)       # Aggregate revenue from all indirect taxes #;
                (all,q,REGDEST)
taxrev1(q)      # Aggregate rev., indirect taxes on intermediate #;
                (all,q,REGDEST)
taxrev2(q)      # Aggregate revenue, indirect taxes on investment #;
                (all,q,REGDEST)
taxrev3(q)      # Aggregate revenue, indirect taxes on households #;
                (all,s,REGSOURCE)
taxrev4(s)      # Aggregate revenue, ind. taxes on for. exports #;
                (all,q,REGDEST)
taxrev5(q)      # Agg. rev., indirect taxes on regional "Other" #;
                (all,q,REGDEST)
taxrev6(q)      # Aggregate rev., indirect tax on fed. "Other" #;
                (all,q,REGDEST)
```

```
taxrevm(q)      # Aggregate tariff revenue #;
                  (all,q,REGDEST)
utility(q)      # Utility per household #;
                  (all,q,REGDEST)
wage_diff(q)    # Regional real wage differential #;
                  (all,q,REGDEST)
xi2(q)          # Investment price index #;
                  (all,q,REGDEST)
xi3(q)          # Consumer price index #;
                  (all,q,REGDEST)
xi4(q)          # Exports Price Index #;
                  (all,q,REGDEST)
xi5(q)          # Regional "Other" demands price index #;
                  (all,q,REGDEST)
xi6(q)          # Fed.  "Other" demands price index #;
                  (all,q,REGDEST)
ximp0(q)        # Duty-paid imports price index #;
                  (all,q,REGDEST)
xim(q)          # Imports price index #;
                  (all,q,REGDEST)
xiplpk(q)       # Index of relative price of labour and capital #;
                  (all,s,REGSOURCE)
xsexp(s)        # Exports vol. in inter-regional trade #;
                  (all,q,REGDEST)
xsimp(q)        # Imports vol. in inter-regional trade #;
                  (all,q,REGDEST)
xiy_r(q)        # Regional GDP deflator #;
                  (all,q,REGDEST)
z_tot(q)        # Aggregate output: Value-added weights #;
```

! Subsection 2.3.5: Regional vector variables in alphabetical order !

```
                  (all,j,IND)(all,q,REGDEST)
a1cap(j,q)      # Capital augmenting tech. change #;
                  (all,j,IND)(all,q,REGDEST)
a1(j,q)         # All Input augmenting technical change #;
                  (all,j,IND)(all,q,REGDEST)
a1lab(j,q)      # Labor augmenting technical change #;
                  (all,j,IND)(all,q,REGDEST)
a1land(j,q)     # Land augmenting technical change #;
                  (all,j,IND)(all,q,REGDEST)
a1oct(j,q)      # "Other Cost" ticket techncal change#;
                  (all,j,IND)(all,q,REGDEST)
a1prim(j,q)     # All prim. factor technical change #;
                  (all,j,IND)(all,q,REGDEST)
a2ind(j,q)      # Neutral tech change, cap. creation #;
                  (all,i,COM)(all,q,REGDEST)
a3com(i,q)      # Change in household tastes #;
                  (all,i,COM)(all,q,REGDEST)
a3lux(i,q)      # Change in household tastes, luxury #;
                  (all,i,COM)(all,q,REGDEST)
a3sub(i,q)      # Change in household taste, subsist.#;
                  (all,j,IND)(all,q,REGDEST)
a(j,q)          # Average of tech change terms, prod. #;
                  (all,j,IND)(all,q,REGDEST)
arpri(j,q)      # Payroll tax adjustment factor #;
                  (all,j,IND)(all,q,REGDEST)
curcap(j,q)     # Current capital stock #;
                  (change) (all,i,COM)(all,s,REGSOURCE)
deltax4(i,s)    # Percentage-point change in export tax rates #;
                  (change) (all,i,COM)(all,s,ALLSOURCE) (all,q,REGDEST)
deltax6(i,s,q)
                  # Percent-point change in sales tax rate, Fed. gov. demands #;
                  (change) (all,i,COM)
deltax(i)       # Percentage-point change in the general sales tax rate #;
                  (all,j,IND)(all,q,REGDEST)
efflab(j,q)     # Effective labour input #;
                  (all,j,IND)(all,q,REGDEST)
f1oct(j,q)      # Shifters, "Other Cost" tickets #;
                  (all,i,COM)(all,s,ALLSOURCE) (all,q,REGDEST)
f6a(i,s,q)      # Shift, federal "Other" demand #;
                  (all,q,REGDEST)
f6gen(q)        # Shifter, fed. "Other" demand  #;
```



```
C_SOFTY122(Q) # Other commodity taxes #;
                (all,q,DOMDEST)
C_SOFTY123(Q) # Payroll taxes #;
                (all,q,DOMDEST)
C_SOFTY124(Q) # Property taxes #;
                (all,q,DOMDEST)
C_SOFTY125(Q) # Land taxes #;
                (all,q,DOMDEST)
C_SOFTY126(Q) # Other indirect taxes #;
                (all,q,DOMDEST)
C_SOFTY130(Q) # Interest received #;
                (all,q,DOMDEST)
C_SOFTY140(Q) # Commonwealth grants to the regions #;
                (all,q,DOMDEST)
C_SOFTY141(Q) # Current grants #;
                (all,q,DOMDEST)
C_SOFTY142(Q) # Capital grants #;
                (all,q,DOMDEST)
C_SOFTY150(Q) # Other revenue #;
                (all,q,DOMDEST)
C_SOFTY200(Q) # Depreciation - general government #;
                (all,q,DOMDEST)
C_SOFTY300(Q) # Financing transactions #;
                (all,q,DOMDEST)
C_SOFTY310(Q) # Net borrowing #;
                (all,q,DOMDEST)
C_SOFTY320(Q) # Increase in provisions #;
                (all,q,DOMDEST)
C_SOFTY330(Q) # Other financing transactions #;

! Components of value added: Region by industry !
                (all,j,IND)(all,q,REGDEST)
C_Z01_I_R(j,q)
                # Wages, salaries and supplements by region and industry #;
                (all,j,IND)(all,q,REGDEST)
C_Z02_I_R(J,Q)
                # Imputed wages by region and industry #;
                (all,j,IND)(all,q,REGDEST)
C_Z03_I_R(J,Q)
                # Payroll taxes by region and industry #;
                (all,j,IND)(all,q,REGDEST)
C_Z04_I_R(J,Q)
                # Returns to fixed capital by region and industry #;
                (all,j,IND) (all,q,REGDEST)
C_Z05_I_R(J,Q)
                # Property taxes by region and industry #;
                (all,j,IND) (all,q,REGDEST)
C_Z06_I_R(J,Q)
                # Returns to agricultural land by region and industry #;
                (all,j,IND) (all,q,REGDEST)
C_Z07_I_R(J,Q)
                # Land taxes by region and industry #;
                (all,j,IND) (all,q,REGDEST)
C_Z08_I_R(J,Q)
                # Returns to working capital by region and industry#;
                (all,j,IND) (all,q,REGDEST)
C_Z09_I_R(J,Q)
                # Other indirect taxes by region and industry #;
                (all,j,IND) (all,q,REGDEST)
C_Z10_I_R(J,Q)
                # Sales by final buyers by region and industry #;

! Regional components of value added !
                (all,q,REGDEST)
C_Z01_R(Q)      # Wages, salaries and supplements by region # ;
                (all,q,REGDEST)
C_Z02_R(Q)      # Imputed wages by region  #;
                (all,q,REGDEST)
C_Z03_R(Q)      # Payroll taxes by region  #;
                (all,q,REGDEST)
C_Z04_R(Q)      # Returns to fixed capital by region  #;
                (all,q,REGDEST)
C_Z05_R(Q)      # Property taxes by region  #;
```

```
C_HHLDY000(Q) # Disposable income #;
              (all,q,REGDEST)
C_HHLDY100(Q) # Primary factor income #;
              (all,q,REGDEST)
C_HHLDY110(Q) # Wages, salaries and supplements #;
              (all,q,REGDEST)
C_HHLDY120(Q) # Non-wage primary factor income #;
              (all,q,REGDEST)
C_HHLDY200(Q) # Personal benefit receipts #;
              (all,q,REGDEST)
C_HHLDY210(Q) # Unemployment benefits #;
              (all,q,REGDEST)
C_HHLDY220(Q) # Other personal benefits #;
              (all,q,REGDEST)
C_HHLDY300(Q) # Other income (net) #;
              (all,q,REGDEST)
C_HHLDY400(Q) # Direct taxes #;
              (all,q,REGDEST)
C_HHLDY410(Q) # PAYE taxes #;
              (all,q,REGDEST)
C_HHLDY420(Q) # Taxes on non-wage primary factor income #;
              (all,q,REGDEST)
C_HHLDY430(Q) # Other direct taxes #;

! Some miscellaneous coefficients !
              (all,q,REGDEST)
C_HHLDD001(Q) # Coeff. of labour supply in eqn HHLD007 #;
              (all,q,REGDEST)
C_HHLDD002(Q) # Coeff. of employment in eqn HHLD007 #;

! Total expenditure and its components for regional and Federal governments !
              (all,q,DOMDEST)
C_SOFTQ000(Q) # SOFT : expenditure side total #;
              (all,q,DOMDEST)
C_SOFTQ100(Q) # Expenditure on goods and sevices #;
              (all,q,DOMDEST)
C_SOFTQ110(Q) # Government consumption #;
              (all,q,DOMDEST)
C_SOFTQ120(Q) # Government investment #;
              (all,q,DOMDEST)
C_SOFTQ200(Q) # Personal benefit payments #;
              (all,q,DOMDEST)
C_SOFTQ210(Q) # Unemployment benefits #;
              (all,q,DOMDEST)
C_SOFTQ220(Q) # Other personal benefits #;
              (all,q,DOMDEST)
C_SOFTQ300(Q) # Subsidies #;
              (all,q,DOMDEST)
C_SOFTQ400(Q) # Interest payments  #;
              (all,q,DOMDEST)
C_SOFTQ500(Q) # Commonwealth grants to the regions #;
              (all,q,DOMDEST)
C_SOFTQ510(Q) # Current grants #;
              (all,q,DOMDEST)
C_SOFTQ520(Q) # Capital grants #;
              (all,q,DOMDEST)
C_SOFTQ600(Q) # Other outlays #;

! Total income and its components for regional and Federal governments !
              (all,q,DOMDEST)
C_SOFTY000(Q) # SOFT : income side total #;
              (all,q,DOMDEST)
C_SOFTY100(Q) # Government revenue #;
              (all,q,DOMDEST)
C_SOFTY110(Q) # Direct taxes #;
              (all,q,DOMDEST)
C_SOFTY111(Q) # Income taxes #;
              (all,q,DOMDEST)
C_SOFTY112(Q) # Other direct taxes #;
              (all,q,DOMDEST)
C_SOFTY120(Q) # Indirect taxes #;
              (all,q,DOMDEST)
C_SOFTY121(Q) # Tariff revenue #;
              (all,q,DOMDEST)
```



```
              (all,i,COM)
fep(i)        # Price (upward) shift in export demands #;
              (all,i,COM)
feq(i)        # Quantity (right) shift in export demands #;
              (all,j,IND)(all,q,REGDEST)
frpri(j,q)    # Payroll tax rate shifter #;
              (all,j,IND)(all,q,REGDEST)
fwagei(j,q)   # Industry-specific wage shifter #;
              (all,j,IND)(all,q,REGDEST)
labind(j,q)   # Employment by industry #;
              (all,m,OCC)(all,q,REGDEST)
lambda(m,q)   # Employment by occupation #;
              (all,j,IND)(all,q,REGDEST)
n(j,q)        # Use of land #;
              (all,j,IND)(all,q,REGDEST)
plcap(j,q)    # Rental price of capital #;
              (all,j,IND)(all,q,REGDEST)
pllab(j,q)    # Price of labour #;
              (all,j,IND)(all,q,REGDEST)
plland(j,q)   # Rental price of land #;
              (all,j,IND)(all,q,REGDEST)
ploct(j,q)    # Price of "Other Cost" tickets #;
              (all,i,COM)(all,s,REGSOURCE)
p4r(i,s)      # F.O.B. for. currency export prices #;
              (all,j,IND)(all,q,REGDEST)
pi(j,q)       # Costs of units of capital #;
              (all,i,COM)
pm(i)         # C.I.F. for. currency import prices #;
              (all,s,REGSOURCE)(all,q,REGDEST)
psflo(s,q)    # Price indices in inter-regional trade flows #;
              (all,i,COM)
powtaxm(i)    # Power of tariffs #;
              (all,j,IND)(all,q,REGDEST)
pwagei(j,q)   # Nominal wage rates #;
              (all,j,IND)(all,q,REGDEST)
r0(j,q)       # Current rates of return on capital #;
              (all,j,IND)(all,q,REGDEST)
rpri(j,q)     # Payroll tax rate (in per cent) #;
              (all,i,COM)(all,q,REGDEST)
x0imp(i,q)    # Import volumes #;
              (all,j,IND)(all,q,REGDEST)
xloct(j,q)    # Demand for "Other Cost" tickets #;
              (all,i,COM)(all,s,REGSOURCE)
x4r(i,s)      # Export volumes #;
              (all,j,IND)(all,q,REGDEST)
xi_fac(j,q)   # Index of factor costs #;
              (all,j,IND)(all,q,REGDEST)
xiplpk_ind(j,q)
              # Index of ratio of price of labour to price of capital #;
              (all,s,REGSOURCE)  (all,q,REGDEST)
xsflo(s,q)    # Volume of inter-regional trade flows #;
              (all,j,IND)(all,q,REGDEST)
y(j,q)        # Capital creation by using industry #;
              (all,j,IND)(all,q,REGDEST)
z(j,q)        # Activity level or value-added #;
```

! Subsection 2.3.6: Regional matrix variables in alphabetical order !

```
              (change)(all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
deltax1(i,s,j,q)
              # Percent-point change in tax rate on sales of inter. inputs #;
              (change)(all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
deltax2(i,s,j,q)
              # Percent-point change in tax rate on sales for cap. creat. #;
              (change)(all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
deltax3(i,s,q)
              # Percent-point change in tax rate on sales to households #;
              (change)(all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
deltax5(i,s,q)
              # Percent-point change: tax rate on sales to reg. Other demand #;
              (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
f5a(i,s,q)    # Shift in regional "Other" demands #;
```

```
                (all,i,COM)(all,s,ALLSOURCE)
p0a(i,s)        # Basic price of good i, source s #;
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
p1a(i,s,j,q)    # Prices of inputs for current production #;
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
p2a(i,s,j,q)    # Prices of inputs for capital creation #;
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
p3a(i,s,q)      # Purchasers prices by commodities and source for households #;
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
p5a(i,s,q)      # Purchasers' prices for commodities (by source) by "Other" #;
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
p6a(i,s,q)      # Purchasers' prices paid for commodities by Federal "Other" #;
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
p1c(i,j,q)      # Prices of composite inputs for current production #;
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
p2c(i,j,q)      # Prices of composite inputs for capital creation #;
                (all,i,COM)(all,q,REGDEST)
p3c(i,q)        # Prices of composite inputs for households #;
                (all,j,IND)(all,q,REGDEST)(all,m,OCC)
p1laboi(j,q,m)
                # Wage of occupation type m in industry j #;
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
p1o(i,j,q)      # Price, inputs for current production #;
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
p2o(i,j,q)      # Price, inputs for capital creation #;
                (all,i,COM)(all,q,REGDEST)
p3o(i,q)        # Price, inputs for households #;
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
x1a(i,s,j,q)    # Demands for inputs for current production #;
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
x2a(i,s,j,q)    # Demands for inputs for capital creation #;
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
x3a(i,s,q)      # Household demand for goods #;
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
x5a(i,s,q)      # Regional "Other" demands #;
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
x6a(i,s,q)      # Federal "Other" demands in each region #;
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
x1c(i,j,q)      # Demands for inputs for current production #;
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
x2c(i,j,q)      # Demands for inputs for capital creation #;
                (all,i,COM)(all,q,REGDEST)
x3c(i,q)        # Demands for inputs for households #;
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
x1o(i,j,q)      # Demands for inputs for current production #;
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
x2o(i,j,q)      # Demands for inputs for capital creation #;
                (all,i,COM)(all,q,REGDEST)
x3o(i,q)        # Demands for inputs for households #;
                (all,j,IND)(all,q,REGDEST)(all,m,OCC)
x1laboi(j,q,m)
                # Employment of occupation type m in industry j #;
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)
                (all,q,REGDEST)(all,r,MARGCOM)
x1marg(i,s,j,q,r)
                # Margins - current production #;
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)
                (all,q,REGDEST)(all,r,MARGCOM)
x2marg(i,s,j,q,r)
                # Margins - capital creation #;
                (all,i,COM)(all,s,ALLSOURCE)
                (all,r,MARGCOM)(all,q,REGDEST)
x3marg(i,s,q,r)
                # Margins - on sales to households #;
                (all,i,COM)(all,s,REGSOURCE)(all,r,MARGCOM)
x4marg(i,s,r)   # Margins - on exports #;
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)(all,r,MARGCOM)
x5marg(i,s,q,r)
                # Margins - regional "Other" #;
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)  (all,r,MARGCOM)
x6marg(i,s,q,r)
                # Margins - on sales to Federal " Other" demand #;
```



```
C_Z08           # Returns to working capital #;
C_Z09           # Other indirect taxes #;
C_Z10           # Sales by final buyers #;
C_ZG            # Gross operating surplus #;
C_ZT            # Production taxes #;

NATBT           # National nominal trade balance #;
R_DEBTG         # Nominal interest rate on government debt#;

! Some miscellaneous regional variables !
                (all,q,DOMDEST)
C_DEBTG(Q)      # Government debt #;
                (all,q,DOMDEST)
C_IG_R(Q)       # Government investment #;
                (all,q,REGDEST)
C_IP_R(Q)       # Private investment #;
                (all,q,DOMDEST)
C_PBP_R(Q)      # Personal benefit payments #;

! Gross Regional Product and their expenditure-side components !
                (all,q,REGDEST)
C_DOMPQ000(Q) # GDP at market prices (expenditure side) #;
                (all,q,REGDEST)
C_DOMPQ100(Q) # Domestic absorption #;
                (all,q,REGDEST)
C_DOMPQ110(Q) # Private consumption #;
                (all,q,REGDEST)
C_DOMPQ120(Q) # Private investment #;
                (all,q,REGDEST)
C_DOMPQ130(Q) # Government consumption - regions #;
                (all,q,REGDEST)
C_DOMPQ140(Q) # Government consumption - federal #;
                (all,q,REGDEST)
C_DOMPQ150(Q) # Government investment #;
                (all,q,REGDEST)
C_DOMPQ200(Q) # Inter-regional trade balance #;
                (all,q,REGDEST)
C_DOMPQ210(Q) # Inter-regional exports #;
                (all,q,REGDEST)
C_DOMPQ220(Q) # Inter-regional imports #;
                (all,q,REGDEST)
C_DOMPQ300(Q) # International trade balance #;
                (all,q,REGDEST)
C_DOMPQ310(Q) # International exports #;
                (all,q,REGDEST)
C_DOMPQ320(Q) # International imports #;

! Gross Regional Product and their income-side components !
                (all,q,REGDEST)
C_DOMPY000(Q) # GDP at market prices (income side) #;
                (all,q,REGDEST)
C_DOMPY100(Q) # Wages, salaries and supplements #;
                (all,q,REGDEST)
C_DOMPY110(Q) # Disposable wage income #;
                (all,q,REGDEST)
C_DOMPY120(Q) # PAYE taxes #;
                (all,q,REGDEST)
C_DOMPY200(Q) # GOS : non-wage primary factor income #;
                (all,q,REGDEST)
C_DOMPY210(Q) # Disposable non-wage primary factor income #;
                (all,q,REGDEST)
C_DOMPY220(Q) # Taxes on non-wage primary factor income #;
                (all,q,REGDEST)
C_DOMPY300(Q) # Indirect taxes less subsidies #;
                (all,q,REGDEST)
C_DOMPY310(Q) # Tariff revenue #;
                (all,q,REGDEST)
C_DOMPY320(Q) # Other commodity taxes less subsidies #;
                (all,q,REGDEST)
C_DOMPY330(Q) # Production taxes #;

! Regional household disposable income and it components !
                (all,q,REGDEST)
```

```
                (all,q,REGDEST)
softf003(q)     # Shift variable : capital Commonwealth grants #;
                (all,q,DOMDEST)
softf004(q)     # Shift variable : other revenue #;
                (all,q,DOMDEST)
softf005(q)     # Shift variable : increase in provisions #;
                (all,q,DOMDEST)
softf006(q)     # Shift variable : other outlays #;
                (all,q,DOMDEST)
softf007(q)     # Shift variable : government debt #;
softf011        # shifter for fed. collection of other ind. taxes#;
! Regional household disposable income and their components !
                (all,q,REGDEST)
hhldy000(q)     # Disposable income #;
                (all,q,REGDEST)
hhldy100(q)     # Primary factor income #;
                (all,q,REGDEST)
hhldy110(q)     # Wages, salaries and supplements #;
                (all,q,REGDEST)
hhldy120(q)     # Non-wage primary factor income #;
                (all,q,REGDEST)
hhldy200(q)     # Personal benefit receipts #;
                (all,q,REGDEST)
hhldy210(q)     # Unemployment benefits #;
                (all,q,REGDEST)
hhldy220(q)     # Other personal benefits #;
                (all,q,REGDEST)
hhldy300(q)     # Other income (net) #;
                (all,q,REGDEST)
hhldy400(q)     # Direct taxes #;
                (all,q,REGDEST)
hhldy410(q)     # PAYE taxes #;
                (all,q,REGDEST)
hhldy420(q)     # Taxes on non-wage primary factor income #;
                (all,q,REGDEST)
hhldy430(q)     # Other direct taxes #;

! Shift variables in household disposable income equations !
                (all,q,REGDEST)
hhldf001(q)     # Shift variable : unemployment benefits #;
                (all,q,REGDEST)
hhldf002(q)     # Shift variable : other personal benefits #;
                (all,q,REGDEST)
hhldf003(q)     # Shift variable : other income (net) - hholds #;
                (all,q,REGDEST)
miscf001(q)     # Shift variable : consumption function #;
miscf002        # Shift variable : relative income tax rates #;
```

! Section 3.3: Coefficient declarations for the government finance module!

```
COEFFICIENT

! National macro values !
C_IG            # Nominal government investment #;
C_IP            # Nominal private investment #;
C_PBP           # Personal benefit payments #;
C_SUBSIDIES     # Subsidies #;
C_TI            # Commodity taxes less subsidies (excl tariffs) #;
C_TY            # Income taxes #;
C_UPB           # Unemployment benefits #;
C_YF            # GDP at factor cost #;
C_YN            # Nominal GDP #;
C_YL            # Pre-tax wage income #;
C_YLSTAR        # Post-tax wage income #;

! National components of value added !
C_Z01           # Wages, salaries and supplements #;
C_Z02           # Imputed wages #;
C_Z03           # Payroll taxes #;
C_Z04           # Returns to fixed capital #;
C_Z05           # Property taxes #;
C_Z06           # Returns to agricultural land #;
C_Z07           # Land taxes #;
```


! Section 2.4: Coefficient declarations for the CGE core module !

! Subsection 2.4.1: Naming system for the input-output database flows of goods !

```
! The matrices showing flows of goods are arranged in three main rows,
  namely:
     BAS   flows at basic values
     MAR   marginal flows
     TAX   indirect taxes
  Each row is divided into 6 main columns, namely:
     1     intermediate inputs
     2     inputs to capital creation
     3     consumption demands
     4     export demands
     5     regional "Other" demands
     6     Federal "Other" demands
  Individual matrix names are formed by combining row and column names, thus:
     BAS2 shows inputs to capital creation at basic values
     MAR4 shows non-tax margins on export flows
     TAX1 shows indirect taxes on intermediate inputs
  The dimensions of the matrices vary according to their main row and column.
  Most are subscripted by COM, SOURCE, and DEST (in main columns 4 and 6, by
  COM and source only).  Those in the MAR row are also subscripted by MARGCOM.
  Those in columns 1 and 2 (intermediate and investment) are also subscripted
  by IND.  Hence MAR2(i,s,j,q,r) shows the amount of marginal commodity r used
  in delivering good i from source s to capital creator j in region q.
!

ZERODIVIDE OFF;
```

! Subsection 2.4.2: Input-output database flows of goods !

```
COEFFICIENT
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
BAS1(i,s,j,q);
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
BAS2(i,s,j,q);
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
BAS3(i,s,q);
                (all,i,COM)(all,s,REGSOURCE)
BAS4(i,s);
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
BAS5(i,s,q);
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
BAS6(i,s,q);
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)
                (all,q,REGDEST)(all,r,MARGCOM)
MAR1(i,s,j,q,r);
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)
                (all,q,REGDEST)(all,r,MARGCOM)
MAR2(i,s,j,q,r);
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)(all,r,MARGCOM)
MAR3(i,s,q,r);
                (all,i,COM)(all,s,REGSOURCE)(all,r,MARGCOM)
MAR4(i,s,r);
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)(all,r,MARGCOM)
MAR5(i,s,q,r);
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)(all,r,MARGCOM)
MAR6(i,s,q,r);
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
TAX1(i,s,j,q);
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
TAX2(i,s,j,q);
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
TAX3(i,s,q);
                (all,i,COM)(all,s,REGSOURCE)
TAX4(i,s);
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
TAX5(i,s,q);
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
TAX6(i,s,q);
```

! Subsection 2.4.3: Input-output database flows of primary factors !

```
                (all,m,OCC)(all,j,IND)(all,q,REGDEST)
LAB_OCC_IND(m,j,q)
                # Labour in industry j #;
                (all,j,IND)(all,q,REGDEST)
CAPITAL(j,q)    # Total capital in industry j #;
                (all,j,IND)(all,q,REGDEST)
LAND(j,q)       # Total land use in industry j #;
                (all,j,IND)(all,q,REGDEST)
OTHCOST(j,q)    # "Other Cost Tickets" paid by industry j #;
```

! Subsection 2.4.4: Input-output database flows of imports and tariffs !

```
                (all,i,COM)(all,q,REGDEST)
IMPORTS(i,q)    # Total basic-value imports of good i #;
                (all,i,COM)(all,q,REGDEST)
TARIFF(i,q)     # Tariffs #;
                (all,i,COM)
NATIMPCOST(i)   # Total ex-duty imports i #;
                (all,i,COM)
NATTARIFF(i)    # Tariff revenue #;
```

! Subsection 2.4.5: Elasticites of substitution (Armingtons) and other parameters !

```
                (all,j,IND)(all,q,REGDEST)
QCOEF(j,q)      # Ratio, gross to net rate of return #;
                (all,i,COM)
EXP_ELAST(i)    # Export Demand Elasticities: Typical Value -20.0 #;
                (all,j,IND)(all,q,REGDEST)
SIGMA1FAC(j,q)
                # CES substitution elasticities for primary factors #;
                (all,j,IND)(all,q,REGDEST)
SIGMA1LAB(j,q)
                # CES substitution elasticities between labour types #;
                (all,i,COM)
SIGMA1O(i)      # Armington Import/Domestic elast. of sub: Intermediate #;
                (all,i,COM)
SIGMA2O(i)      # Armington Import/Domestic elast. of sub: Investment #;
                (all,i,COM)
SIGMA3O(i)      # Armington Import/Domestic elast. of sub: Households #;
                (all,j,IND)(all,q,REGDEST)
BETA_R(j,q)     # Param. to dist. investment #;
                (all,i,COM)
SIGMA1C(i)      # Intra-Domestic substitution elasticities: User 1 #;
                (all,i,COM)
SIGMA2C(i)      # Intra-Domestic substitution elasticities: User 2 #;
                (all,i,COM)
SIGMA3C(i)      # Intra-Domestic substitution elasticities: User 3 #;
```

! Subsection 2.4.6: Purchasers' values !

```
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
PVAL1A(i,s,j,q)
                # Purchase value for current production #;
                (all,i,COM)(all,aa,TWOSOURCE)(all,j,IND)(all,q,REGDEST)
PVAL1T(i,aa,j,q)
                # Total purchase value for current production: Dom. & imp. #;
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
PVAL1O(i,j,q)   # Total purchase value for current production: Dom. & imp. #;
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
PVAL2A(i,s,j,q)
                # Purchase value for capital creation #;
                (all,i,COM)(all,aa,TWOSOURCE)(all,j,IND)(all,q,REGDEST)
PVAL2T(i,aa,j,q)
                # Total purchase value for capital creation: Dom. & imp. #;
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
PVAL2O(i,j,q)   # Total purchase value for cap. creat.: Domestic & imported #;
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
PVAL3A(i,s,q)   # Purchase value for household consumption #;
                (all,i,COM)(all,aa,TWOSOURCE)(all,q,REGDEST)
PVAL3T(i,aa,q)
                # Total purchase value for household consumption: Dom. & imp. #;
```



```
softy112(q)     # Other direct taxes #;
                  (all,q,DOMDEST)
softy120(q)     # Indirect taxes #;
                  (all,q,DOMDEST)
softy121(q)     # Tariff revenue #;
                  (all,q,DOMDEST)
softy122(q)     # Other commodity taxes #;
                  (all,q,DOMDEST)
softy123(q)     # Payroll taxes #;
                  (all,q,DOMDEST)
softy124(q)     # Property taxes #;
                  (all,q,DOMDEST)
softy125(q)     # Land taxes #;
                  (all,q,DOMDEST)
softy126(q)     # Other indirect taxes #;
                  (all,q,DOMDEST)
softy130(q)     # Interest received #;
                  (all,q,DOMDEST)
softy140(q)     # Commonwealth grants to regions #;
                  (all,q,DOMDEST)
softy141(q)     # Current grants #;
                  (all,q,DOMDEST)
softy142(q)     # Capital grants #;
                  (all,q,DOMDEST)
softy150(q)     # Other revenue #;
                  (all,q,DOMDEST)
softy200(q)     # Consumption of fixed capital - general govt #;
                  (all,q,DOMDEST)
softy300(q)     # Financing transactions #;
                  (all,q,DOMDEST)
softy310(q)     # Net borrowing #;
                  (all,q,DOMDEST)
softy320(q)     # Increase in provisions #;
                  (all,q,DOMDEST)
softy330(q)     # Other financing transactions #;
                  (all,q,DOMDEST)
f_oft(q)        # Other financing transactions shifter #;
                  (all,q,DOMDEST)
realdef(q)      # Real government budget deficit #;

! Regional and Federal government expenditures !
                (all,q,DOMDEST)
softq000(q)     # SOFT: expenditure side total #;
                  (all,q,DOMDEST)
softq100(q)     # Expenditure on goods and sevices #;
                  (all,q,DOMDEST)
softq110(q)     # Government consumption #;
                  (all,q,DOMDEST)
softq120(q)     # Government investment #;
                  (all,q,DOMDEST)
softq200(q)     # Personal benefit payments #;
                  (all,q,DOMDEST)
softq210(q)     # Unemployment benefits #;
                  (all,q,DOMDEST)
softq220(q)     # Other personal benefits #;
                  (all,q,DOMDEST)
softq300(q)     # Subsidies #;
                  (all,q,DOMDEST)
softq400(q)     # Interest payments #;
                  (all,q,DOMDEST)
softq500(q)     # Commonwealth grants to regions #;
                  (all,q,DOMDEST)
softq510(q)     # Current grants #;
                  (all,q,DOMDEST)
softq520(q)     # Capital grants #;
                  (all,q,DOMDEST)
softq600(q)     # Other outlays #;

! Shift variables in regional and Federal government finance equations !
                (all,q,REGDEST)
softf001(q)     # Shift variable : other direct taxes #;
                  (all,q,REGDEST)
softf002(q)     # Shift variable : current Commonwealth grants #;
```

```
z06_r(q)        # Returns to agricultural land #;
                  (all,q,REGDEST)
z07_r(q)        # Land taxes #;
                  (all,q,REGDEST)
z08_r(q)        # Returns to working capital #;
                  (all,q,REGDEST)
z09_r(q)        # Other indirect taxes #;
                  (all,q,REGDEST)
z10_r(q)        # Sales by final buyers #;
                  (all,q,REGDEST)
zg_r(q)         # Gross operating surplus #;
                  (all,q,REGDEST)
zt_r(q)         # Production taxes #;

! Gross Regional Products; income-side and expenditure-side components !
                  (all,q,REGDEST)
dompy000(q)     # GDP at market prices (income side) #;
                  (all,q,REGDEST)
dompy100(q)     # Wages, salaries and supplements #;
                  (all,q,REGDEST)
dompy110(q)     # Disposable wage income #;
                  (all,q,REGDEST)
dompy120(q)     # PAYE taxes #;
                  (all,q,REGDEST)
dompy200(q)     # GOS : non-wage primary factor income #;
                  (all,q,REGDEST)
dompy210(q)     # Disposable non-wage primary factor income #;
                  (all,q,REGDEST)
dompy220(q)     # Taxes on non-wage primary factor income #;
                  (all,q,REGDEST)
dompy300(q)     # Indirect taxes less subsidies #;
                  (all,q,REGDEST)
dompy310(q)     # Tariff revenue #;
                  (all,q,REGDEST)
dompy320(q)     # Other commodity taxes less subsidies #;
                  (all,q,REGDEST)
dompy330(q)     # Production taxes #;
                  (all,q,REGDEST)
dompq000(q)     # GDP at market prices (expenditure side) #;
                  (all,q,REGDEST)
dompq100(q)     # (all,q,REGDEST) domestic absorption #;
                  (all,q,REGDEST)
dompq110(q)     # Private consumption #;
                  (all,q,REGDEST)
dompq120(q)     # Private investment #;
                  (all,q,REGDEST)
dompq130(q)     # Government consumption - regions #;
                  (all,q,REGDEST)
dompq140(q)     # Government consumption - federal #;
                  (all,q,REGDEST)
dompq150(q)     # Government investment #;
                  (all,q,REGDEST)
dompq200(q)     # Inter-regional trade balance #;
                  (all,q,REGDEST)
dompq210(q)     # Inter-regional exports #;
                  (all,q,REGDEST)
dompq220(q)     # Inter-regional imports #;
                  (all,q,REGDEST)
dompq300(q)     # International trade balance #;
                  (all,q,REGDEST)
dompq310(q)     # International exports #;
                  (all,q,REGDEST)
dompq320(q)     # International imports #;

! Regional and Federal government revenues and deficits !
                  (all,q,DOMDEST)
softy000(q)     # SOFT : income side total #;
                  (all,q,DOMDEST)
softy100(q)     # Government revenue #;
                  (all,q,DOMDEST)
softy110(q)     # Direct taxes #;
                  (all,q,DOMDEST)
softy111(q)     # Income taxes #;
                  (all,q,DOMDEST)
```



```
                  (all,i,COM)(all,q,REGDEST)
PVAL3O(i,q)     # Total reg. purchase value for household consumption #;
                  (all,i,COM)(all,s,REGSOURCE)
PVAL4R(i,s)     # Purchase value for foreign exports #;
                  (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
PVAL5A(i,s,q)   # Purchase value for regional Other demands #;
                  (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
PVAL6A(i,s,q)   # Purchase value for Federal Other demands #;
```

! Subsection 2.4.7: Factor-payment aggregates !

```
                  (all,j,IND)(all,q,REGDEST)
LABOUR(j,q)     # Total labour bill in industry j #;
                  (all,j,IND)
NATLABOUR(j)    # National labour bill in industry j #;
                  (all,m,OCC)(all,q,REGDEST)
LAB_OCC(m,q)    # Total labour bill in occupation m #;
                  (all,m,OCC)
NATLAB_OCC(m)   # National labour bill in occupation m #;
                  (all,q,REGDEST)
AGGLAB(q)       # Total payments to labour #;
NATAGGLAB       # National wage bill #;
                  (all,q,REGDEST)
AGGCAP(q)       # Total payments to capital #;
NATAGGCAP       # National capital rentals #;
                  (all,q,REGDEST)
AGGLND(q)       # Total payments to land #;
NAtAGGLND       # National payments to land #;
                  (all,q,REGDEST)
TOTFAC(q)       # Total primary factor payments#;
NATTOTFAC       # National payments to primary factors #;
                  (all,q,REGDEST)
AGGOCT(q)       # Total other cost ticket payments #;
NATAGGOCT       # National payments to other costs #;
                  (all,j,IND)(all,q,REGDEST)
TOTFACIND(j,q)
                # Total factor input to industry j #;
                  (all,j,IND)
NATTOTFACIND(j)
                # National factor payments in industry j #;
```

! Subsection 2.4.8: Final-demand aggregates !

```
                  (all,j,IND)(all,q,REGDEST)
INVEST (j,q)    # Total capital created for each industry #;
                  (all,j,IND)
NATINVEST(j)    # Total capital created for each industry #;
                  (all,q,REGDEST)
AGGINV(q)       # Investment #;
NATAGGINV       # National investment #;
                  (all,q,REGDEST)
AGGCON(q)       # Total purchases by households #;
NATAGGCON       # National household demand #;
                  (all,q,REGDEST)
AGGEXP(q)       # Total export earnings #;
NATAGGEXP       # National exports #;
                  (all,q,REGDEST)
AGGOTH5(q)      # Value of reg. Other demands #;
NATAGGOTH5      # National value of regional Other demands #;
                  (all,q,REGDEST)
AGGOTH6(q)      # Total value of Fed. Other demands #;
NATAGGOTH6      # National value of Fed. Other demands #;
                  (all,i,COM)
NATIMPORTS(i)   # Total basic-value imports of good i #;
                  (all,i,COM)(all,q,REGDEST)
IMPCOST(i,q)    # Total ex-duty imports of good r #;
                  (all,q,REGDEST)
AGGIMP(q)       # Total foreign currency import costs #;
NATAGGIMP       # National foreign currency import costs #;
```

! Subsection 2.4.9: Tax aggregates !

```
                (all,q,REGDEST)
AGGTAX1(q)      # Sales tax on current production #;
                (all,q,REGDEST)
AGGTAX2(q)      # Sales tax on capital creation #;
                (all,q,REGDEST)
AGGTAX3(q)      # Sales tax on household consumption #;
                (all,s,REGSOURCE)
AGGTAX4(s)      # Sales tax on foreign exports #;
                (all,q,REGDEST)
AGGTAX5(q)      # Sales tax on regional Other demand #;
                (all,q,REGDEST)
AGGTAX6(q)      # Sales tax on Fed. Other demand #;
                (all,q,REGDEST)
AGGTAXM(q)      # Tariffs #;
NATAGGTAX1      # Sales tax on current production: National total #;
NATAGGTAX2      # Sales tax on capital creation: National total #;
NATAGGTAX3      # Sales tax on household consumption: National total #;
NATAGGTAX4      # Sales tax on foreign exports: National total #;
NATAGGTAX5      # Sales tax on regional Other demand: National total #;
NATAGGTAX6      # Sales tax on Fed. Other demand: National total #;
NATAGGTAXM      # Tariffs: National total #;
NATAGGTAX       # Sales tax: National total #;
                (all,q,DOMDEST)
AGGTAX(q)       # Aggregate indirect tax revenue collected by region #;
```

! Subsection 2.4.10: GDP !

```
NATGDPEX        # nominal gdp from expenditure side #;
NATGDPIN        # nominal gdp from income side #;
```

! Subsection 2.4.11: Inter-regional trade flows !

```
                (all,s,REGSOURCE) (all,q,REGDEST)
C_XSFLO(s,q)    # Trade: inter-regional trade flows #;
                (all,s,REGSOURCE)
C_XSEXP(s)      # Inter-regional exports #;
                (all,q,REGDEST)
C_XSIMP(q)      # Inter-regional imports #;
```

! Subsection 2.4.12: Elasticities and parameters for household demand !

```
                (all,q,REGDEST)
FRISCH(q)       # Frisch 'parameter' #;
                (all,i,COM)(all,q,REGDEST)
DELTA(i,q)      # Marginal household budget shares #;
                (all,i,COM)(all,q,REGDEST)
S3COM(i,q)      # Shares in total household expenditure #;
                (all,q,REGDEST)
ALPHA(q)        # share of supernumerary in total expenditure #;
                (all,i,COM)(all,q,REGDEST)
EPS(i,q)        # Household expenditure elasticities #;
                (all,i,COM)(all,q,REGDEST)
ALPHA_I(i,q)    # Supernumerary expend. good i/total expenditure good i #;
```

! Subsection 2.4.13: Costs and sales aggregates !

```
                (all,i,COM)(all,s,REGSOURCE)
DIRSALES(i,s)   # direct usage #;
                (all,r,MARGCOM)(all,s,REGSOURCE)
MARSALES(r,s)   # margin usage #;
                (all,i,COM)(all,s,REGSOURCE)
SALES(i,s)      # all usage #;
                (all,j,IND)(all,q,REGDEST)
COSTS(j,q)      # Total costs in industry j in region q #;
                (all,i,COM)(all,s,REGSOURCE)
LOSTGOODS(i,s)
                # discrepancy #;
                (all,i,COM)
LOSTGOODS1(i)   # subtotal #;
                (all,s,REGSOURCE)
LOSTGOODS2(s)   # subtotal #;
```



```
  (v)   hhldy***  (a household income component);
  (vi)  z_**_r    (a value-added component in gross regional production).
  In each array name "***" represents three or two digits component numbers.
!
```

! Section 3.2: Variable declarations for the government finance module !

```
VARIABLE
! National macro variables !
bstar           # Balance of trade surplus as percentage of GDP #;
ig              # Nominal government investment #;
ip              # Nominal private investment #;
pbp             # Personal benefit payments #;
rl              # Tax rate - wages, salaries and supplements #;
rk              # Tax rate - non-wage primary factor income #;
ti              # Commodity taxes less subsidies (excl tariffs) #;
ty              # Income taxes #;
upb             # Unemployment benefits #;
wn              # Nominal pre-tax wage rate #;
wnstar          # Nominal post-tax wage rate #;
wrstar          # Real post-tax wage rate #;
xiy             # GDP deflator #;
yf              # GDP at factor cost #;
yl              # Pre-tax wage income #;
ylstar          # Post-tax wage income #;
yn              # Nominal GDP #;
yr              # Real GDP #;
z03             # Payroll taxes #;
z05             # Property taxes #;
z07             # Land taxes #;
z09             # Other indirect taxes #;
z10             # Sales by final buyers #;

! Regional macro variables !
                  (all,q,DOMDEST)
dgstar(q)       # Government net borrowing / total outlays #;
                  (all,q,DOMDEST)
debtg(q)        # Government debt #;
                  (all,q,DOMDEST)
g_r(q)          # Government consumption #;
                  (all,q,DOMDEST)
ig_r(q)         # Government investment #;
                  (all,q,REGDEST)
labsup(q)       # labour supply #;
                  (all,q,DOMDEST)
pbp_r(q)        # Personal benefit payments #;
                  (all,q,REGDEST)
rpr(q)          # payroll tax rate #;
                  (all,q,REGDEST)
ti_r(q)         # Commodity taxes less subsidies (excl tariffs) #;
                  (all,q,REGDEST)
tod_r(q)        # Other direct taxes #;
                  (all,q,REGDEST)
xisfb(q)        # price index : sales by final buyers #;
                  (all,q,REGDEST)
yn_r(q)         # Nominal GDP - regions #;
                  (all,q,REGDEST)
yr_r(q)         # Real GDP - regions #;
                  (all,q,REGDEST)
yd_r(q)         # Household disposable income #;

! Factor payments and tax components of value added: regions !
                (all,q,REGDEST)
z01_r(q)        # Wages, salaries and supplements #;
                  (all,q,REGDEST)
z02_r(q)        # Imputed wages #;
                  (all,q,REGDEST)
z03_r(q)        # Payroll taxes #;
                  (all,q,REGDEST)
z04_r(q)        # Returns to fixed capital #;
                  (all,q,REGDEST)
z05_r(q)        # Property taxes #;
                  (all,q,REGDEST)
```

```
reg_plcap(q) = caprev(q) - kt(q);

E_realwage_w # Real wages for workers:deflated by CPI#
               (all,q,REGDEST)
realwage_w(q) = pwage(q) - xi3(q);

E_realwage_p # Real wages for producers: deflated by GDP def.#
               (all,q,REGDEST)
realwage_p(q) = pwage(q) - xiy_r(q);

E_totdom # Domestic terms of trade #
               (all,q,REGDEST)
totdom(q) = psexp(q) - psimp(q);

E_totfor # Foreign terms of trade #
               (all,q,REGDEST)
totfor(q) = xi4(q) - xim(q);

E_r0_tot # Regional industry-aggregate rate of return #
               (all,q,REGDEST)
r0_tot(q) = sum(j,IND,CAPITAL(j,q)*r0(j,q));

E_xiplpk_ind  # Relat. price of lab & cap #
               (all,j,IND)(all,q,REGDEST)
xiplpk_ind(j,q) = pwagei(j,q) - plcap(j,q);

E_xiplpk #Index of relat. price of lab & cap #
               (all,q,REGDEST)
xiplpk(q) = pwage(q) - reg_plcap(q);
```

! Subsection 2.8.19: Employment Aggregates !

```
E_lambda # Demand for labour by occupation #
               (all,m,OCC)(all,q,REGDEST)
LAB_OCC(m,q)*lambda(m,q) =
   sum(j,IND,LAB_OCC_IND(m,j,q)*x1laboi(j,q,m));

E_natlambda # National demand for labour by occupation #
               (all,m,OCC)
NATLAB_OCC(m)*natlambda(m)=sum(q,REGDEST,LAB_OCC(m,q)*lambda(m,q));
```

! Part 3: GOVERNMENT FINANCE MODULE !

```
! First version by G.A. Meagher, August 1992 !
! The equations in this module:
  (a) determine gross domestic products of regions from income and
  expenditure sides using the variables determined in the core module
  (b) describe income and expenditure accounts of regional and Fed. govts.
  as given in State Finance Statistics (ABS Cat No. 5512.0).

  All coefficients, variables and the read statements which are specific to
  this module are documented in this part.

 The module reads seven sets of data:
 * income and expenditure components of financial transactions
 * income and expenditure components of domestic production by region
 * components of value added by region and industries
 * components of household income by region
 * miscellaneous data !
```

! Section 3.1: Naming system in the government finance module !

```
! Following the style of the core module, all variable names are in lower case.
  However, the coefficient naming system is differnet. A variable name in
  upper case with a  prefix of "C_" defines the coefficient associated with the
  variable.  For example, the variable  "hhldy000" represents the percentage
  change in household disposable income. Its associated coefficient is
  "C_HHLDY00" which represents household disposable income in the base year.
  The following are the major array names for variables and coefficients:
  (i)   softy***  (summary of financial transaction, an income component);
  (ii)  softq***  (summary of financial transaction, an expenditure component);
  (iii) dompy***  (domestic regional production, an income component);
  (iv)  dompq***  (domestic regional production, an expenditure component);
```


! Subsection 2.4.14: Mapping coefficients for domestic/foreign and tiny !

```
                (all,s,ALLSOURCE)
IS_DOM(s)     # Binary dummy 1 #;
                (all,s,ALLSOURCE)
IS_IMP(s)     # Binary dummy 2 #;
TINY          # A very small number #;
```

! Section 2.5: Read statements for the CGE core module !

! Subsection 2.5.1: Reads of input-output database flows of goods and factor payments !

```
READ
BAS1 from file MDATA header "BAS1";
BAS2 from file MDATA header "BAS2";
BAS3 from file MDATA header "BAS3";
BAS4 from file MDATA header "BAS4";
BAS5 from file MDATA header "BAS5";
BAS6 from file MDATA header "BAS6";
MAR1 from file MDATA header "MAR1";
MAR2 from file MDATA header "MAR2";
MAR3 from file MDATA header "MAR3";
MAR4 from file MDATA header "MAR4";
MAR5 from file MDATA header "MAR5";
MAR6 from file MDATA header "MAR6";
TAX1 from file MDATA header "TAX1";
TAX2 from file MDATA header "TAX2";
TAX3 from file MDATA header "TAX3";
TAX4 from file MDATA header "TAX4";
TAX5 from file MDATA header "TAX5";
TAX6 from file MDATA header "TAX6";
LAB_OCC_IND from file MDATA HEADER "LABR";
CAPITAL from file MDATA header "CPTL";
LAND from file MDATA header "LAND";
OTHCOST from file MDATA header "OCTS";
TARIFF from file MDATA header "TARF";
```

! Subsection 2.5.2: Reads of parameters !

```
QCOEF from file MDATA header "p027";
FRISCH from file MDATA header "p021";
DELTA from file MDATA header "P044";
EXP_ELAST from file MDATA header "p018";
SIGMA1FAC from file MDATA header "p028";
SIGMA1LAB from file MDATA header "SLAB";
SIGMA1O from file MDATA header "p015";
SIGMA2O from file MDATA header "p016";
SIGMA3O from file MDATA header "p017";
BETA_R from file MDATA header "BETR";
SIGMA1C from file MDATA header "b015";
SIGMA2C from file MDATA header "b016";
SIGMA3C from file MDATA header "b017";
```

! Section 2.6: Update statements for the CGE core module !

```
UPDATE
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
BAS1(i,s,j,q) = p0a(i,s)*x1a(i,s,j,q);
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
BAS2(i,s,j,q) = p0a(i,s)*x2a(i,s,j,q);
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
BAS3(i,s,q) = p0a(i,s)*x3a(i,s,q);
                (all,i,COM)(all,s,REGSOURCE)
BAS4(i,s) = p0a(i,s)*x4r(i,s);
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
BAS5(i,s,q) = p0a(i,s)*x5a(i,s,q);
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
BAS6(i,s,q) = p0a(i,s)*x6a(i,s,q);
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)
                (all,q,REGDEST)(all,r,MARGCOM)
```

```
MAR1(i,s,j,q,r) = p0a(r,q)*x1marg(i,s,j,q,r);
                  (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)
                  (all,q,REGDEST)(all,r,MARGCOM)
MAR2(i,s,j,q,r) = p0a(r,q)*x2marg(i,s,j,q,r);
                  (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)(all,r,MARGCOM)
MAR3(i,s,q,r)= p0a(r,q)*x3marg(i,s,q,r);
                  (all,i,COM)(all,s,REGSOURCE)(all,r,MARGCOM)
MAR4(i,s,r)= p0a(r,s)*x4marg(i,s,r);
                  (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)(all,r,MARGCOM)
MAR5(i,s,q,r)= p0a(r,q)*x5marg(i,s,q,r);
                  (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)(all,r,MARGCOM)
MAR6(i,s,q,r)= p0a(r,s)*x6marg(i,s,q,r);
                  (explicit)(all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
TAX1(i,s,j,q) = TAX1(i,s,j,q) + [ BAS1(i,s,j,q)*deltax1(i,s,j,q) +
                        TAX1(i,s,j,q)*(x1a(i,s,j,q)+p0a(i,s))]/100;
                  (explicit)(all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
TAX2(i,s,j,q) = TAX2(i,s,j,q) + [ BAS2(i,s,j,q)*deltax2(i,s,j,q) +
                        TAX2(i,s,j,q)*(x2a(i,s,j,q)+p0a(i,s))]/100;
                  (explicit)(all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
TAX3(i,s,q) = TAX3(i,s,q) + [ BAS3(i,s,q)*deltax3(i,s,q) +
                        TAX3(i,s,q)*(x3a(i,s,q)+p0a(i,s))]/100;
                  (explicit)(all,i,COM)(all,s,REGSOURCE)
TAX4(i,s) = TAX4(i,s) + [ BAS4(i,s)*deltax4(i,s) +
                        TAX4(i,s)*(x4r(i,s)+p0a(i,s))]/100;
                  (explicit)(all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
TAX5(i,s,q) = TAX5(i,s,q) + [ BAS5(i,s,q)*deltax5(i,s,q) +
                        TAX5(i,s,q)*(x5a(i,s,q)+p0a(i,s))]/100;
                  (explicit)(all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
TAX6(i,s,q) = TAX6(i,s,q) + [BAS6(i,s,q)*deltax6(i,s,q) +
                        TAX6(i,s,q)*(x6a(i,s,q)+p0a(i,s))]/100;
                  (all,m,OCC)(all,j,IND)(all,q,REGDEST)
LAB_OCC_IND(m,j,q) = pllaboi(j,q,m)*x1laboi(j,q,m);
                  (all,j,IND)(all,q,REGDEST)
CAPITAL(j,q) = plcap(j,q)*curcap(j,q);
                  (all,j,IND)(all,q,REGDEST)
LAND(j,q) = plland(j,q)*n(j,q);
                  (all,j,IND)(all,q,REGDEST)
OTHCOST(j,q) = ploct(j,q)*xloct(j,q);
                  (change)(all,i,COM)(all,q,REGDEST)
TARIFF(i,q) = Tariff(i,q)*(pm(i) + natphi + x0imp(i,q))/100 +
                  IMPORTS(i,q)*powtaxm(i)/100;
                  (explicit)(all,j,IND)(all,q,REGDEST)
QCOEF(j,q) = QCOEF(j,q)*(1.0+ [1.0 - QCOEF(j,q)]*[plcap(j,q) - pi(j,q)]/100);
                  (explicit)(all,q,REGDEST)
FRISCH(q) = FRISCH(q)*(1.0 + [luxexp(q) - c(q)]/100.0);
                  (all,i,COM)(all,q,REGDEST)
DELTA(i,q) = a3lux(i,q);
```

! Section 2.7: Formulae for the CGE core module !

! Subsection 2.7.1: Formulae for tariffs !

```
FORMULA
                  (all,i,COM)(all,q,REGDEST)
IMPORTS(i,q) = sum(j,IND,BAS1(i,"foreign",j,q) + BAS2(i,"foreign",j,q))
    + BAS3(i,"foreign",q) + BAS5(i,"foreign",q)+BAS6(i,"foreign",q);
                  (all,i,COM)
NATTARIFF(i) = sum(q,REGDEST,TARIFF(i,q));
```

! Subsection 2.7.2: Formulae for purchasers' values !

```
                  (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
PVAL1A(i,s,j,q)=BAS1(i,s,j,q)+TAX1(i,s,j,q)+sum(r,MARGCOM,MAR1(i,s,j,q,r));
                  (all,i,COM)(all,j,IND)(all,q,REGDEST)
PVAL1T(i,"domestic",j,q)=sum(s,REGSOURCE,PVAL1A(i,s,j,q));
                  (all,i,COM)(all,j,IND)(all,q,REGDEST)
PVAL1T(i,"foreign",j,q)= PVAL1A(i,"foreign",j,q);
                  (all,i,COM)(all,j,IND)(all,q,REGDEST)
PVAL1O(i,j,q)=sum(aa,TWOSOURCE,PVAL1T(i,aa,j,q));
                  (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
PVAL2A(i,s,j,q)=BAS2(i,s,j,q)+TAX2(i,s,j,q)+sum(r,MARGCOM,MAR2(i,s,j,q,r));
                  (all,i,COM)(all,j,IND)(all,q,REGDEST)
PVAL2T(i,"domestic",j,q)=sum(s,REGSOURCE,PVAL2A(i,s,j,q));
```



```
E_natximP0 # National duty paid imports price index #
natximp0 = (1.0/[NATAGGIMP+NATAGGTAXM])
             *sum(i,COM,NATIMPORTS(i)*p0a(i,"foreign"));

E_psflo # Price index - inter-regional trade flows #
                  (all,s,REGSOURCE) (all,q,REGDEST)
(TINY + C_XSFLO(s,q)) * psflo(s,q)
    =  sum(i,com, sum(j,IND, BAS1(i,s,j,q) * p0a(i,s)))
     + sum(i,com, sum(j,IND, BAS2(i,s,j,q) * p0a(i,s)))
     + sum(i,com, BAS3(i,s,q) * p0a(i,s))
     + sum(i,com, BAS5(i,s,q) * p0a(i,s));

 E_psexp # Price index - inter-regional exports #
                  (all,s,REGSOURCE)
(TINY + C_XSEXP(s)) * psexp(s)
    =  sum(q,regdest, C_XSFLO(s,q) * psflo(s,q))
     - C_XSFLO(s,s) * psflo(s,s);

 E_psimp # Price index - inter-regional imports #
                  (all,q,REGDEST)
(TINY + C_XSIMP(q)) * psimp(q)
   = sum(s,regsource, C_XSFLO(s,q) * psflo(s,q))
    - C_XSFLO(q,q) * psflo(q,q);

E_nattot # National terms of trade #
nattot = natxi4 - natxim;
```

! Subsection 2.8.17: Money wage settings !

```
E_pllaboi # payroll tax adjustmemt #
                  (all,j,IND)(all,q,REGDEST) (all,m,OCC)
pllaboi(j,q,m)= pwagei(j,q) + arpri(j,q);

E_pwagei # flexible setting of money wages #
                  (all,j,IND)(all,q,REGDEST)
pwagei(j,q)= natxi3  + natfwage + fwage(q) + fwagei(j,q);
! Notice wages indexed to national cpi !

E_pwage # region-wide nominal wage #
                  (all,q,REGDEST)
AGGLAB(q)*pwage(q) = sum(j,IND,LABOUR(j,q)*pwagei(j,q));

! The following equations have been added to give flexibility to the regional
  labour market and regional migration module. !

E_natrealw # National real wage: consumer #
NATAGGLAB*natrealwage
    = sum(j,IND,sum(q,REGDEST,LABOUR(j,q)*
      (natfwage + fwage(q) + fwagei(j,q))));

E_wage_diff # region real-wage diff #
                  (all,q,REGDEST)
wage_diff(q) = pwage(q) - natxi3 - natrealwage;
```

! Subsection 2.8.18: Miscellaneous definitions of factor prices !

```
E_natplcap # Aggregate nominal capital rentals #
natplcap = natcaprev - natkt;
E_natpwage #Aggregate nominal wages#
NATAGGLAB * natpwage = sum(q,REGDEST, AGGLAB(q)*pwage(q));

E_natrwage_w # National real wages for workers:deflated by CPI #
NATrwage_w = natrealwage;

E_natrwage_p #National real wages for producers#
natrwage_p = natrwage_w + natxi3 -natxigdp;

E_natxiplpk # Relative prices of labour and capital#
natxiplpk = natpwage - natplcap;

E_reg_plcap # Reginal rental price of capital #
                  (all,q,REGDEST)
```

```
E_natimp # Foreign currency value of imports #
natimp = (1.0/NATAGGIMP)*sum(i,COM,NATIMPCOST(i)*[pm(i)+ natx0imp(i)]);

E_natimpvol # Import volume index #
natimpvol = natimp + natphi - natxim;

E_natgdpexp # aggregate nominal GDP from expenditure side #
natgdpexp =  (1.0/NATGDPEX)*
  (NATAGGCON*natc  +       NATAGGINV*natin
  +NATAGGOTH5*natothnom5 + NATAGGOTH6*natothnom6
  +NATAGGEXP*(natexport+natphi)-NATAGGIMP*(natimp+natphi));

E_natgdpreal # Real GDP,expenditure side #
natgdpreal = natgdpexp - natxigdp;

E_natdelb # Balance of trade in BILLIONS of dollars#
   1000.0*100.0*
natdelb = NATAGGEXP*natexport - NATAGGIMP*natimp;
     !  the factor 1000.0 converts millions to billions !
     !  the factor 100.0 converts percentages to proportional change !
```

! Subsection 2.8.16: Regional and national price indices !

```
E_xi3 # consumer price index # (all,q,REGDEST)
xi3(q) =(1.0/AGGCON(q))*sum(i,COM,sum(s,ALLSOURCE,PVAL3A(i,s,q)*p3a(i,s,q)));

E_natxi3 # consumer price index #
NATAGGCON*natxi3 = sum(q,REGDEST,AGGCON(q)*xi3(q));

E_xi2 # investment price index #
                (all,q,REGDEST)
xi2(q) = (1.0/AGGINV(q))*sum(j,IND,INVEST(j,q)*pi(j,q));
E_natxi2 # investment price index #
NATAGGINV*natxi2 = sum(q,REGDEST,AGGINV(q)*xi2(q));

E_xi4 # exports price index #
                (all,q,REGDEST)
xi4(q) - natphi = (1.0/AGGEXP(q))*sum(i,COM,PVAL4R(i,q)*p4r(i,q));
E_natxi4 # exports price index #
natxi4 = (1.0/NATAGGEXP)*sum(q,REGDEST,AGGEXP(q)*xi4(q));

E_xi5 # Regional Other demands price index #
                (all,q,REGDEST)
xi5(q) =(1.0/AGGOTH5(q))*sum(i,COM,sum(s,ALLSOURCE,PVAL5A(i,s,q)*p5a(i,s,q)));

E_natxi5 # National agg. regional Other demands price index #
natxi5 =(1.0/NATAGGOTH5)*sum(q,REGDEST,AGGOTH5(q)*xi5(q));

E_xi6 # Price index for Fed. Other demand #
                (all,q,REGDEST)
 xi6(q) = (1.0/AGGOTH6(q))*sum(i,com, sum(s,ALLSOURCE,PVAL6A(i,s,q) *
          p6a(i,s,q)));

E_natxi6 # National price index Fed Other demands #
NATAGGOTH6*natxi6 =sum(i,COM,sum(s,allsource,sum(q,REGDEST,
        PVAL6A(i,s,q)*p6a(i,s,q))));

E_natxigdp # Price index for GDP,expenditure side #
natxigdp =(1.0/NATGDPEX)*
    (NATAGGCON*natxi3 +  NATAGGINV*natxi2 + NATAGGOTH5*natxi5
   + NATAGGOTH6*natxi6 + NATAGGEXP*natxi4 - NATAGGIMP*natxim);

E_xim # imports price index #
                (all,q,REGDEST)
xim(q) - natphi =(1.0/AGGIMP(q))*sum(i,COM,IMPCOST(i,q)*pm(i));

E_natxim # National imports price index #
natxim = (1.0/NATAGGIMP)*sum(q,REGDEST,AGGIMP(q)*xim(q));

E_ximp0 # duty paid imports price index #
          (all,q,REGDEST)
ximp0(q) = (1.0/[AGGIMP(q)+AGGTAXM(q)])
             *sum(i,COM,IMPORTS(i,q)*p0a(i,"foreign"));
```



```
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
PVAL2T(i,"foreign",j,q)= PVAL2A(i,"foreign",j,q);
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
PVAL2O(i,j,q)=sum(aa,TWOSOURCE,PVAL2T(i,aa,j,q));
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
PVAL3A(i,s,q)=BAS3(i,s,q)+TAX3(i,s,q)+sum(r,MARGCOM,MAR3(i,s,q,r));
                (all,i,COM)(all,q,REGDEST)
PVAL3T(i,"domestic",q)=sum(s,REGSOURCE,PVAL3A(i,s,q));
                (all,i,COM)(all,q,REGDEST)
PVAL3T(i,"foreign",q)= PVAL3A(i,"foreign",q);
                (all,i,COM)(all,q,REGDEST)
PVAL3O(i,q)=sum(aa,TWOSOURCE,PVAL3T(i,aa,q));
                (all,i,COM)(all,s,REGSOURCE)
PVAL4R(i,s)=BAS4(i,s)+TAX4(i,s)+sum(r,MARGCOM,MAR4(i,s,r));
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
PVAL5A(i,s,q)=BAS5(i,s,q)+TAX5(i,s,q)+sum(r,MARGCOM,MAR5(i,s,q,r));
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
PVAL6A(i,s,q)=BAS6(i,s,q)+TAX6(i,s,q)+sum(r,MARGCOM,MAR6(i,s,q,r));
```

! Subsection 2.7.3: Formulae for factor-payment aggregates !

```
                (all,j,IND)(all,q,REGDEST)
LABOUR(j,q) = sum(m,OCC,LAB_OCC_IND(m,j,q));
                (all,j,IND)
NATLABOUR(j) = sum(q,REGDEST,LABOUR(j,q));
                (all,m,OCC)(all,q,REGDEST)
LAB_OCC(m,q) = sum(j,IND,LAB_OCC_IND(m,j,q));
                (all,m,OCC)
NATLAB_OCC(m) = sum(q,REGDEST,LAB_OCC(m,q));
                (all,q,REGDEST)
AGGLAB(q) = sum(j,IND,LABOUR(j,q));
NATAGGLAB = sum(q,REGDEST,AGGLAB(q));
                (all,q,REGDEST)
AGGCAP(q) = sum(j,IND,CAPITAL(j,q));
NATAGGCAP = sum(q,REGDEST,AGGCAP(q));
                (all,q,REGDEST)
AGGLND(q) = sum(j,IND,LAND(j,q));
NATAGGLND = sum(q,REGDEST,AGGLND(q));
                (all,q,REGDEST)
TOTFAC(q) = AGGLAB(q) + AGGCAP(q) + AGGLND(q);
NATTOTFAC = sum(q,REGDEST,TOTFAC(q));
                (all,q,REGDEST)
AGGOCT(q) = sum(j,IND,OTHCOST(j,q));
NATAGGOCT = sum(q,REGDEST,AGGOCT(q));
                (all,j,IND)(all,q,REGDEST)
TOTFACIND(j,q) = LABOUR(j,q) + CAPITAL(j,q) + LAND(j,q);
                (all,j,IND)
NATTOTFACIND(j) = sum(q,REGDEST,TOTFACIND(j,q));
```

! Subsection 2.7.4: Formulae for final-demand aggregates !

```
                (all,j,IND)(all,q,REGDEST)
INVEST(j,q) = sum(i,COM,PVAL2O(i,j,q));
                (all,j,IND)
NATINVEST (j) = sum(q,REGDEST,INVEST(j,q));
                (all,q,REGDEST)
AGGINV(q) = sum(j,IND,INVEST(j,q));
NATAGGINV = sum(q,REGDEST,AGGINV(q));
                (all,q,REGDEST)
AGGCON(q) = sum(i,COM,PVAL3O(i,q));
NATAGGCON = sum(q,REGDEST,AGGCON(q));
                (all,q,REGDEST)
AGGEXP(q) = sum(i,COM,PVAL4R(i,q));
NATAGGEXP = sum(q,REGDEST,AGGEXP(q));
                (all,q,REGDEST)
AGGOTH5(q) = sum(i,COM,sum(s,ALLSOURCE,PVAL5A(i,s,q)));
NATAGGOTH5 = sum(q,REGDEST,AGGOTH5(q));
                (all,q,REGDEST)
AGGOTH6(Q) = sum(i,COM,sum(s,ALLSOURCE,PVAL6A(i,s,q)));
NATAGGOTH6 = sum(i,COM,sum(s,ALLSOURCE,sum(q,REGDEST,PVAL6A(i,s,q))));
```

! Subsection 2.7.5: Formulae for import aggregates !

```
                (all,i,COM)
NATIMPORTS(i) = sum(q,REGDEST, IMPORTS(i,q));
                (all,i,COM)(all,q,REGDEST)
IMPCOST(i,q) = IMPORTS(i,q) - TARIFF(i,q);
                (all,i,COM)
NATIMPCOST(i) = NATIMPORTS(i) - NATTARIFF(i);
                (all,q,REGDEST)
AGGIMP(q) = sum(i,COM,IMPCOST(i,q));
NATAGGIMP = sum(i,COM,NATIMPCOST(i));
```

! Subsection 2.7.6: Formulae for tax revenue aggregates !

```
                (all,q,REGDEST)
AGGTAX1(q) = sum(i,COM,sum(s,ALLSOURCE,sum(j,IND,TAX1(i,s,j,q))));
                (all,q,REGDEST)
AGGTAX2(q) = sum(i,COM,sum(s,ALLSOURCE,sum(j,IND,TAX2(i,s,j,q))));
                (all,q,REGDEST)
AGGTAX3(q) = sum(i,COM,sum(s,ALLSOURCE,TAX3(i,s,q)));
                (all,s,REGSOURCE)
AGGTAX4(s) = sum(i,COM,TAX4(i,s));
                (all,q,REGDEST)
AGGTAX5(q) = sum(i,COM,sum(s,ALLSOURCE,TAX5(i,s,q)));
                (all,q,REGDEST)
AGGTAX6(q) = sum(i,COM,sum(s,ALLSOURCE,TAX6(i,s,q)));
                (all,q,REGDEST)
AGGTAXM(q) = sum(i,COM,Tariff(i,q));
NATAGGTAX1 = sum(q,REGDEST,AGGTAX1(q));
NATAGGTAX2 = sum(q,REGDEST,AGGTAX2(q));
NATAGGTAX3 = sum(q,REGDEST,AGGTAX3(q));
NATAGGTAX4 = sum(s,REGSOURCE,AGGTAX4(s));
NATAGGTAX5 = sum(q,REGDEST,AGGTAX5(q));
NATAGGTAX6 = sum(q,REGDEST,AGGTAX6(q));
NATAGGTAXM = sum(q,REGDEST, AGGTAXM(q));
NATAGGTAX  = NATAGGTAX1 + NATAGGTAX2 + NATAGGTAX3 + NATAGGTAX4 +
             NATAGGTAX5 + NATAGGTAX6 + NATAGGTAXM;
                (all,q,REGDEST)
AGGTAX(q) = AGGTAX1(q) + AGGTAX2(q) + AGGTAX3(q) + AGGTAX5(q);
AGGTAX("federal") = NATAGGTAX4 + NATAGGTAX6 + NATAGGTAXM;
```

! Subsection 2.7.7: Formulae for GDP !

```
NATGDPEX = NATAGGCON + NATAGGINV
           + NATAGGOTH5 + NATAGGOTH6 + NATAGGEXP - NATAGGIMP;
NATGDPIN = NATTOTFAC + NATAGGOCT + NATAGGTAX;
```

! Subsection 2.7.8: Formulae for inter-regional trade flows !

```
                (all,s,REGSOURCE)(all,q,REGDEST)
C_XSFLO(s,q) = sum(i,COM, sum(j,IND, BAS1(i,s,j,q)))
             + sum(i,COM, sum(j,IND, BAS2(i,s,j,q)))
             + sum(i,COM, BAS3(i,s,q))
             + sum(I,COM, BAS5(I,S,Q));
                (all,s,REGSOURCE)
C_XSEXP(s) = sum(q,REGDEST,C_XSFLO(s,q)) - C_XSFLO(s,s);
                (all,q,REGDEST)
C_XSIMP(q) = sum(s,REGSOURCE,C_XSFLO(s,q)) - C_XSFLO(q,q);
```


! Subsection 2.8.15: National GDP, real and nominal and its components !

```
E_natcaprev # aggregate payments to capital #
natcaprev = (1.0/NATAGGCAP)* sum(q,REGDEST,AGGCAP(q)*caprev(q));

E_natlabrev # aggregate payments to labour #
natlabrev = (1.0/NATAGGLAB)* sum(q,REGDEST, AGGLAB(q)*labrev(q));

E_natlndrev # aggregate payments to land #
natlndrev = (1.0/NATAGGLND)* sum(q,REGDEST, AGGLND(q)*lndrev(q));

E_natoctrev # aggregate other cost ticket payments #
natoctrev = (1.0/NATAGGOCT)*sum(q,REGDEST, AGGOCT(q)*octrev(q));

E_nattaxrevm # aggregate tariff revenue #
nattaxrevm = (1.0/NATAGGTAXM)* sum(q,REGDEST,AGGTAXM(q)*taxrevm(q));

E_nattaxind # aggregate value of indirect taxes #
nattaxind =(1.0/NATAGGTAX)*
     (NATAGGTAX1*nattaxrev1 + NATAGGTAX2*nattaxrev2
    + NATAGGTAX3*nattaxrev3 + NATAGGTAX4*nattaxrev4
    + NATAGGTAX5*nattaxrev5 + NATAGGTAX6*nattaxrev6
    + NATAGGTAXM*nattaxrevm);

E_natgdpinc # aggregate nominal GDP from income side #
natgdpinc = (1.0/NATGDPIN)*
     (NATAGGLND*natlndrev + NATAGGCAP*natcaprev + NATAGGLAB*natlabrev
     + NATAGGOCT*natoctrev + NATAGGTAX*nattaxind);

E_natkt # aggregate usage of capital,rental weights #
natkt = (1.0/NATAGGCAP)* sum(q,REGDEST,AGGCAP(q)*kt(q));

E_natl # 22.3 aggregate employment,wage bill weights #
natl = (1.0/NATAGGLAB)* sum(q,REGDEST,AGGLAB(q)*l(q));

E_natZ_TOT # aggregate output: value-added weights #
NATTOTFAC*natz_tot = sum(q,REGDEST,TOTFAC(q)*z_tot(q));

E_natZ # aggregate output: value-added weights #
                (all,j,IND)
(TINY+NATTOTFACIND(j))*natz(j) = sum(q,REGDEST,TOTFACIND(j,q)*z(j,q));

E_natc # aggregate nominal consumption #
NATAGGCON*natc = sum(q,REGDEST,AGGCON(q)*c(q) );
E_natcr # aggregate real consumption #
NATAGGCON*natcr = sum(q,REGDEST,AGGCON(q)*cr(q) );

E_natin # total nominal investment #
natin = natir + natxi2;

E_natir # total real investment #
natir = (1.0/NATAGGINV)*sum(j,IND,NATINVEST(j)*naty(j));

E_natothnom5 # aggregate nominal value of regional Other demands #
natothnom5 = natothreal5 + natxi5;

E_natothnom6 # aggregate nominal value of Fed. Other demands #
natothnom6 = natothreal6 + natxi6;

E_natothreal5 # aggregate real regional Other demands #
natothreal5 =(1.0/NATAGGOTH5)*sum(q,REGDEST,AGGOTH5(q)*othreal5(q));

E_natothreal6 # aggregate real Fed. Other demands #
NATAGGOTH6*natothreal6 = sum(i,COM,sum(s,allsource,sum(q,REGDEST,
   PVAL6A(i,s,q)*x6a(i,s,q))));

E_natexport # Foreign currency value of exports #
natexport =(1.0/NATAGGEXP)*sum(s,REGSOURCE,AGGEXP(s)*export(s));

E_natexpvol # Export volume index #
natexpvol = natexport + natphi - natxi4;
```

```
E_c_a # Household budget constraint #
                (all,q,REGDEST)
AGGCON(q)* c(q) =
          sum(i,COM, sum(s,ALLSOURCE, PVAL3A(i,s,q)*(x3a(i,s,q)+p3a(i,s,q)) ));

E_cr # Real (total) household consumption #
                (all,q,REGDEST)
cr(q) = c(q) - xi3(q);

E_in # Total nominal investment #
                (all,q,REGDEST)
in(q) = ir(q) + xi2(q);

E_ir # Total real investment #
                (all,q,REGDEST)
ir(q) = (1.0/AGGINV(q))*sum(j,IND,INVEST(j,q)*y(j,q));

E_othnom5 # aggregate nominal value of regional Other demands #
                (all,q,REGDEST)
othnom5(q) = othreal5(q) + xi5(q);

E_othreal5 # aggregate real regional Other demands #
                (all,q,REGDEST)
othreal5(q) = (1.0/AGGOTH5(q))*
                            sum(i,COM,sum(s,ALLSOURCE,PVAL5A(i,s,q)*x5a(i,s,q) ));

E_othnom6 # Nominal federal Other demand #
                (all,q,REGDEST)
othnom6(q) = othreal6(q) + xi6(q);

E_othreal6 # Real federal Other demand #
                (all,q,REGDEST)
AGGOTH6(q) * othreal6(q) =
              sum(i,com, sum(s,ALLSOURCE,PVAL6A(i,s,q) * x6a(i,s,q)));

E_export # Foreign currency value of exports #
                (all,q,REGDEST)
export(q) =(1.0/AGGEXP(q))*sum(i,COM,PVAL4R(i,q)*[p4r(i,q) + x4r(i,q)] );

E_expvol # Export volume index #
                (all,q,REGDEST)
expvol(q) = export(q) + natphi - xi4(q);

E_imp # foreign currency value of imports #
                (all,q,REGDEST)
imp(q) = (1.0/AGGIMP(q))*sum(i,COM,IMPCOST(i,q)*[pm(i)+ x0imp(i,q)]);

E_impvol # Import volume index #
                (all,q,REGDEST)
impvol(q) = imp(q) + natphi - xim(q);

 E_trd # Inter-regional trade flows (including diagonal term) #
                (all,s,REGSOURCE) (all,q,REGDEST)
(TINY + C_XSFLO(s,q)) * (psflo(s,q) + xsflo(s,q))
    =  sum(i,com, sum(j,IND, BAS1(i,s,j,q) * (p0a(i,s) + x1a(i,s,j,q))))
     + sum(i,com, sum(j,IND, BAS2(i,s,j,q) * (p0a(i,s) + x2a(i,s,j,q))))
     + sum(i,com, BAS3(i,s,q) * (p0a(i,s) + x3a(i,s,q)))
     + sum(i,com, BAS5(i,s,q) * (p0a(i,s) + x5a(i,s,q)));

E_int_exp # Inter-regional exports #
                (all,s,REGSOURCE)
(TINY + C_XSEXP(s)) * (psexp(s) + xsexp(s))
    =  sum(q,regdest, C_XSFLO(s,q) * (psflo(s,q) + xsflo(s,q)))
     - C_XSFLO(s,s) * (psflo(s,s) + xsflo(s,s));

E_int_imp # Inter-regional imports #
                (all,q,REGDEST)
(TINY + C_XSIMP(q)) * (psimp(q) + xsimp(q))
    =  sum(s,regsource, C_XSFLO(s,q) * (psflo(s,q) + xsflo(s,q)))
     - C_XSFLO(q,q) * (psflo(q,q) + xsflo(q,q));

E_cr_shr # shares in national Cr #
                (all,q,REGDEST)
cr(q) = natcr + cr_shr(q);
```


! Subsection 2.7.9: Formulae for household demands !

```
                ZERODIVIDE DEFAULT 0.0;
                FORMULA
                (all,i,COM)(all,q,REGDEST)
S3COM(i,q) = PVAL3O(i,q)/AGGCON(q);
                ZERODIVIDE OFF;
                FORMULA
                (all,q,REGDEST)
ALPHA(q) = -1/FRISCH(q);
                ZERODIVIDE DEFAULT 1.0;
                FORMULA
                (all,i,COM)(all,q,REGDEST)
EPS(i,q) = DELTA(i,q)/S3COM(i,q);
                !  Marginal/average shares !
                ZERODIVIDE OFF;
                FORMULA
                (all,i,COM)(all,q,REGDEST)
ALPHA_I(i,q) = ALPHA(q)*EPS(i,q);
```

! Subsection 2.7.10: Formulae for costs and sales aggregates !

```
                (all,i,COM)(all,s,REGSOURCE)
DIRSALES(i,s) = sum(j,IND, sum(q,REGDEST,  BAS1(i,s,j,q)) )
              +   sum(j,IND, sum(q,REGDEST,  BAS2(i,s,j,q)) )
              +              sum(q,REGDEST,  BAS3(i,s,q) )
              +                              BAS4(i,s)
              +              sum(q,REGDEST,  BAS5(i,s,q)
              +                              BAS6(i,s,q));
                (all,r,MARGCOM)(all,s,REGSOURCE)
MARSALES(r,s) = sum(j,IND, sum(i,COM, sum(ss,ALLSOURCE,MAR1(i,ss,j,s,r)) ))
              +   sum(j,IND, sum(i,COM, sum(ss,ALLSOURCE,MAR2(i,ss,j,s,r)) ))
              +   sum(i,COM, sum(ss,ALLSOURCE,           MAR3(i,ss,s,r) )  )
              +   sum(i,COM,                             MAR4(i,s,r) )
              +   sum(i,COM, sum(ss,ALLSOURCE,           MAR5(i,ss,s,r)
              +                                          MAR6(i,ss,s,r)));
                  ! In the above, we assume that margins are produced in the
                    destination region, except that MAR4 and MAR6 are produced
                    in the source region !
                (all,i,NONMARGCOM)(all,s,REGSOURCE)
SALES(i,s) = DIRSALES(i,s);
                (all,i,MARGCOM)(all,s,REGSOURCE)
SALES(i,s) = DIRSALES(i,s) + MARSALES(i,s);
                (all,j,IND)(all,q,REGDEST)
COSTS(j,q) =   sum(i,COM,sum(s,ALLSOURCE,  BAS1(i,s,j,q)) )
             +  sum(i,COM,sum(s,ALLSOURCE,sum(R,MARGCOM ,MAR1(i,s,j,q,r)) ) )
             +  sum(i,COM,sum(s,ALLSOURCE,  TAX1(i,s,j,q)) )
             +  sum(m,OCC, LAB_OCC_IND(m,j,q))
             +  CAPITAL(j,q)
             +  LAND(j,q)
             +  OTHCOST(j,q);
                ! Following formula used to check for balanced database !
                ! SHOULD EVALUATE TO ZERO !
                (all,i,COM)(all,s,REGSOURCE)
LOSTGOODS(i,s) = SALES(i,s) - COSTS(i,s);
                (all,i,COM)
LOSTGOODS1(i) = sum(s,REGSOURCE, LOSTGOODS(i,s));
                (all,s,REGSOURCE)
LOSTGOODS2(s) = sum(i,COM, LOSTGOODS(i,s));
```

! Subsection 2.7.11: Formulae for binary dummies and tiny !

```
                (all,s,REGSOURCE)
IS_DOM(s) = 1;
IS_DOM("foreign") = 0;
                (all,s,REGSOURCE)
IS_IMP(s) = 0;
IS_IMP("foreign") = 1;
TINY = 0.000000000001;
```

! Section 2.8: Equations of the CGE core module in thematic order !

! Subsection 2.8.1: Demands by industries for intermediate inputs-User 1 !

```
! Local and national Armington nests for intermediate inputs to current
  production. National nest gives substitution posibility between Natinal and
  foreign inputs and local nest gives substitution posibillity between inputs
  from 8 regions. !

EQUATION

E_x1a # Demand for goods by all source, User 1 #
              (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
x1a(i,s,j,q) = IS_DOM(s)*(x1c(i,j,q)-SIGMA1C(i)*(p1a(i,s,j,q)-p1c(i,j,q)))
          + IS_IMP(s)*(x1o(i,j,q)-SIGMA1O(i)*(p1a(i,"foreign",j,q)-p1o(i,j,q)));

E_p1o # Price of domestic/foreign composite, User 1 #
              (all,i,COM)(all,j,IND)(all,q,REGDEST)
(TINY+PVAL1O(i,j,q))*p1o(i,j,q) = sum(s,ALLSOURCE,PVAL1A(i,s,j,q)*p1a(i,s,j,q));

E_p1c # Price of domestic composite, User 1 #
              (all,i,COM)(all,j,IND)(all,q,REGDEST)
(TINY+PVAL1T(i,"domestic",j,q))*p1c(i,j,q)
                                  = sum(s,REGSOURCE,PVAL1A(i,s,j,q)*p1a(i,s,j,q));

E_x1c # Demand for domestic composite, User 1 #
              (all,i,COM)(all,j,IND)(all,q,REGDEST)
x1c(i,j,q) = x1o(i,j,q) - SIGMA1O(i)*(p1c(i,j,q)-p1o(i,j,q));

E_x1o # Demands for composite inputs, User 1 #
              (all,i,COM)(all,j,IND)(all,q,REGDEST)
x1o(i,j,q) =  z(j,q) + a1(j,q);

E_x1oct # Industry demands for other cost tickets #
              (all,j,IND)(all,q,REGDEST)
x1oct(j,q) = z(j,q) + a1(j,q) + a1oct(j,q);

E_p1oct # Indexing of prices of "Other Cost" tickets #
              (all,j,IND)(all,q,REGDEST)
p1oct(j,q) = xi3(q) + f1oct(j,q);
```

! Subsection 2.8.2: Primary factor demands, prices and supplies !

```
E_efflab # Industry demands for effective labour #
              (all,j,IND)(all,q,REGDEST)
efflab(j,q) = z(j,q) + a1(j,q) + a1prim(j,q) + a1lab(j,q)
               - SIGMA1FAC(j,q)*[p1lab(j,q) + a1lab(j,q) - xi_fac(j,q)];

E_curcap # Industry demands for capital #
              (all,j,IND)(all,q,REGDEST)
curcap(j,q) = z(j,q) + a1(j,q) + a1prim(j,q) + a1cap(j,q)
              - SIGMA1FAC(j,q)*[p1cap(j,q) + a1cap(j,q) - xi_fac(j,q)];

E_p1land # Industry demands for land #
              (all,j,IND)(all,q,REGDEST)
n(j,q) = z(j,q) + a1(j,q) + a1prim(j,q) + a1land(j,q)
          - SIGMA1FAC(j,q)*[p1land(j,q) + a1land(j,q) - xi_fac(j,q)];

E_xi_fac # Effective price term for factor demand equations #
              (all,j,IND)(all,q,REGDEST)
(TINY+TOTFACIND(j,q))*xi_fac(j,q) = LABOUR(j,q)*(p1lab(j,q)+a1lab(j,q))
                                  + CAPITAL(j,q)*(p1cap(j,q)+a1cap(j,q))
                                  + LAND(j,q)*(p1land(j,q)+a1land(j,q));

! Labour by occupation nest !
E_x1laboi # Demand for labour by industry and skill group #
              (all,m,OCC)(all,j,IND)(all,q,REGDEST)
x1laboi(j,q,m) = efflab(j,q) - SIGMA1LAB(j,q)*[p1laboi(j,q,m) - p1lab(j,q)];

E_p1lab # Price to each industry of labour in general #
              (all,j,IND)(all,q,REGDEST)
 (TINY+LABOUR(j,q))*p1lab(j,q) = sum(m,OCC,LAB_OCC_IND(m,j,q)*p1laboi(j,q,m));
```


! Subsection 2.8.13: Basic prices !

```
E_p0a # Zero pure profits in current production #
              (all,j,IND)(all,q,REGDEST)
(TINY+COSTS(j,q))*(p0a(j,q) - a(j,q)) =
                              sum(i,COM,sum(s,ALLSOURCE,PVAL1A(i,s,j,q)*p1a(i,s,j,q)))
                                  + sum(m,OCC,LAB_OCC_IND(m,j,q)* p1laboi(j,q,m))
    + CAPITAL(j,q)*p1cap(j,q) + LAND(j,q)*p1land(j,q) + OTHCOST(j,q)*p1oct(j,q);

E_a # Technical change by industry-current production #
              (all,j,IND)(all,q,REGDEST)
(TINY+COSTS(j,q))*[a(j,q) - a1(j,q)] =
                                        TOTFACIND(j,q)*a1prim(j,q)
                                      + LABOUR(j,q)*a1lab(j,q)
                                      + CAPITAL(j,q)*a1cap(j,q)
                                      + LAND(j,q)*a1land(j,q)
                                      + OTHCOST(j,q)*a1oct(j,q);

E_pi # Zero pure profits in capital creation #
              (all,j,IND)(all,q,REGDEST)
(TINY+INVEST(j,q))*(pi(j,q) - a2ind(j,q)) =
                                    sum(i,COM,sum(s,ALLSOURCE,
                                          PVAL2A(i,s,j,q)*p2a(i,s,j,q) ));

E_p0ab # Zero pure profits in importing #
              (all,i,COM)
p0a(i,"foreign") = pm(i) + natphi + powtaxm(i);
```

! Subsection 2.8.14: Components of regional GDP, real and nominal !

```
E_caprev # Aggregate payments to capital #
              (all,q,REGDEST)
caprev(q) = (1.0/AGGCAP(q))* sum(j,IND,CAPITAL(j,q)*(p1cap(j,q)+curcap(j,q)));

E_labrev # Aggregate payments to labour #
              (all,q,REGDEST)
labrev(q) = (1.0/AGGLAB(q))*
     sum(j,IND,sum(m,OCC,LAB_OCC_IND(m,j,q)*(p1laboi(j,q,m)+x1laboi(j,q,m))));

E_lndrev # Aggregate payments to land #
              (all,q,REGDEST)
lndrev(q) = (1.0/AGGLND(q))* sum(j,IND,LAND(j,q)*(p1land(j,q)+n(j,q)));

E_octrev # Aggregate other cost ticket payments #
              (all,q,REGDEST)
octrev(q) = (1.0/AGGOCT(q))* sum(j,IND,OTHCOST(j,q)*(p1oct(j,q)+x1oct(j,q)));

E_taxrevm # Aggregate tariff revenue #
              (all,q,REGDEST)
AGGTAXM(q)*taxrevm(q) = sum(i,COM,Tariff(i,q)*(pm(i) + natphi + x0imp(i,q))
                            + IMPORTS(i,q)*powtaxm(i));

E_taxind # Aggregate value of indirect taxes #
              (all,q,REGDEST)
taxind(q) = (1.0/AGGTAX(q))*(  AGGTAX1(q)*taxrev1(q) + AGGTAX2(q)*taxrev2(q)
                             + AGGTAX3(q)*taxrev3(q) + AGGTAX5(q)*taxrev5(q));

E_natlabind # Aggregate employment- wage bill weights #
              (all,j,IND)
(TINY+NATLABOUR(j))*natlabind(j) = sum(q,REGDEST,LABOUR(j,q)*labind(j,q));

E_l # Aggregate employment- wage bill weights #
              (all,q,REGDEST)
l(q) = (1.0/AGGLAB(q))* sum(j,IND,LABOUR(j,q)*labind(j,q));

E_kt # Aggregate usage of capital,rental weights #
              (all,q,REGDEST)
kt(q) = (1.0/AGGCAP(q))* sum(j,IND,CAPITAL(j,q)*curcap(j,q));

E_z_tot # Aggregate output: value-added weights #
              (all,q,REGDEST)
TOTFAC(q)*z_tot(q) = sum(j,IND,TOTFACIND(j,q)*z(j,q));
```

! Subsection 2.8.11: Margin usage of commodities !

```
E_x1marg # Margins on sales to producers #
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
                (all,s,ALLSOURCE)(all,r,MARGCOM)
x1marg(i,s,j,q,r)= x1a(i,s,j,q);

E_x2marg # Margins on sales to capital creators #
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
                (all,s,ALLSOURCE)(all,r,MARGCOM)
x2marg(i,s,j,q,r)= x2a(i,s,j,q);

E_x3marg # Margins on sales to household consumption #
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)(all,r,MARGCOM)
x3marg(i,s,q,r) = x3a(i,s,q);

E_x4marg # Margins on exports: factory gate to port #
                (all,i,COM)(all,r,MARGCOM)(all,s,REGSOURCE)
x4marg(i,s,r) = x4r(i,s);

E_x5marg # Margins on sales to regional Other demands #
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)(all,r,MARGCOM)
x5marg(i,s,q,r) = x5a(i,s,q);

E_x6marg # Margins on sales to fedral Other demands in each region #
                (all,i,COM)(all,r,MARGCOM)(all,s,ALLSOURCE)(all,q,REGDEST)
x6marg(i,s,q,r) = x6a(i,s,q);
```

! Subsection 2.8.12: Supply equals demand for domestic & imported commodities !

```
E_mkt_clear_margins # Demand equals supply for margin commodities #
                (all,r,MARGCOM)(all,s,REGSOURCE)
(TINY+SALES(r,s))*z(r,s) =
                          sum(j,IND, sum(q,REGDEST,  BAS1(r,s,j,q)*x1a(r,s,j,q)
                                                   +
BAS2(r,s,j,q)*x2a(r,s,j,q)))
                                   + sum(q,REGDEST,  BAS3(r,s,q)  *x3a(r,s,q))
                                                   + BAS4(r,s)    *x4r(r,s)
                                   + sum(q,REGDEST,  BAS5(r,s,q)  *x5a(r,s,q))
                                   + sum(q,REGDEST,  BAS6(r,s,q)  *x6a(r,s,q))
                             + sum(j,IND, sum(i,COM, sum(ss,ALLSOURCE,
MAR1(i,ss,j,s,r)*x1marg(i,ss,j,s,r) + MAR2(i,ss,j,s,r)*x2marg(i,ss,j,s,r) )))
 +           sum(i,COM, sum(ss,ALLSOURCE,MAR3(i,ss,s,r)  *x3marg(i,ss,s,r)  ))
 +           sum(i,COM,                  MAR4(i,s,r)     *x4marg(i,s,r)     )
 +           sum(i,COM, sum(ss,ALLSOURCE,MAR5(i,ss,s,r)  *x5marg(i,ss,s,r)  ))
 +           sum(i,COM, sum(ss,ALLSOURCE,MAR6(i,ss,s,r)  *x6marg(i,ss,s,r)  ));
               ! In the above, we assume that margins are produced in the
                 destination region, except that MAR4 and MAR6 are produced
                 in the source region !

E_mkt_clear_nomarg # Demand equals supply for non margin commodities #
                (all,r,NONMARGCOM)(all,s,REGSOURCE)
(TINY+SALES(r,s))*z(r,s) =
                         sum(j,IND, sum(q,REGDEST,  BAS1(r,s,j,q)*x1a(r,s,j,q)) )
                       + sum(j,IND, sum(q,REGDEST,  BAS2(r,s,j,q)*x2a(r,s,j,q)) )
                       +            sum(q,REGDEST,  BAS3(r,s,q)  *x3a(r,s,q)  )
                       +                            BAS4(r,s)    *x4r(r,s)
                       +             sum(q,REGDEST, BAS5(r,s,q)  *x5a(r,s,q) )
                       +             sum(q,REGDEST, BAS6(r,s,q)  *x6a(r,s,q) );

E_x0impa #  Import volumes of commodities by region #
                (all,i,COM)(all,q,REGDEST)
(TINY+IMPORTS(i,q))*x0imp(i,q) =
                            sum(j,IND,BAS1(i,"foreign",j,q)*x1a(i,"foreign",j,q)
                          +           BAS2(i,"foreign",j,q)*x2a(i,"foreign",j,q))
                          +           BAS3(i,"foreign",q)*x3a(i,"foreign",q)
                          +           BAS5(i,"foreign",q)*x5a(i,"foreign",q)
                          +           BAS6(i,"foreign",q)*x6a(i,"foreign",q);

E_natx0imp # Import volumes #
                (all,i,COM)
(TINY+NATIMPORTS(i))*natx0imp(i) = sum(q,REGDEST, IMPORTS(i,q)*x0imp(i,q));
```



```
E_labind # Employment by industry #
                (all,j,IND)(all,q,REGDEST)
(TINY+LABOUR(j,q))*labind(j,q) = sum(m,OCC,LAB_OCC_IND(m,j,q)*x1laboi(j,q,m));
```

! Subsection 2.8.3: Demands by industries for capital creation, User 2 !

```
! Local and national Armington nests for capital creation !

E_x2a # Demand for goods by source, User 2 #
                (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
x2a(i,s,j,q) =
     IS_DOM(s)*(x2c(i,j,q)-SIGMA2C(i)*(p2a(i,  s       ,j,q)-p2c(i,j,q)))
   + IS_IMP(s)*(x2o(i,j,q)-SIGMA2O(i)*(p2a(i,"foreign",j,q)-p2o(i,j,q)));

E_p2o # Price of domestic/foreign composite, User 2 #
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
(TINY+PVAL2O(i,j,q))*p2o(i,j,q) = sum(s,ALLSOURCE,PVAL2A(i,s,j,q)*p2a(i,s,j,q));

E_p2c # Price of domestic composite, User 2 #
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
(TINY+PVAL2T(i,"domestic",j,q))*p2c(i,j,q)
                                   = sum(s,REGSOURCE,PVAL2A(i,s,j,q)*p2a(i,s,j,q));

E_x2c # Demand for domestic composite, User 2 #
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
x2c(i,j,q) = x2o(i,j,q) - SIGMA2O(i)*(p2c(i,j,q)-p2o(i,j,q));

E_x2o # Demands for composite inputs, User 2 #
                (all,i,COM)(all,j,IND)(all,q,REGDEST)
x2o(i,j,q) =  y(j,q) + a2ind(j,q);
```

! Subsection 2.8.4: Household demands for commodities, User 3 !

```
! Household demand for subsistence and luxury consumption by commodities, and
  local and national Armingtion nests for commodity demand for household
  consumption. !

E_x3o # Household demand for composite commodities #
                (all,i,COM) (all,q,REGDEST)
x3o(i,q) =  [1-ALPHA_I(i,q)]*[qhous(q)+a3sub(i,q)]
          + ALPHA_I(i,q)*[luxexp(q) + a3lux(i,q) - p3o(i,q)];

E_a3lux # Default setting for luxury taste shifter #
                (all,i,COM)(all,q,REGDEST)
a3lux(i,q) = a3sub(i,q) - sum(k,COM, DELTA(k,q)*a3sub(k,q));

E_a3sub # Default setting for subsistence taste shifter #
                (all,i,COM)(all,q,REGDEST)
a3sub(i,q) = a3com(i,q) - sum(k,COM, S3COM(k,q)*a3com(k,q));

E_utility # Change in utility disregarding taste change terms #
                (all,q,REGDEST)
utility(q) = luxexp(q) - qhous(q) - sum(i,COM, DELTA(i,q)*p3o(i,q));

E_x3a # Demand for goods by source, User 3 #
                (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
x3a(i,s,q) =  IS_DOM(s)*(x3c(i,q)-SIGMA3C(i)*(p3a(i,  s       ,q)-p3c(i,q)))
            + IS_IMP(s)*(x3o(i,q)-SIGMA3O(i)*(p3a(i,"foreign",q)-p3o(i,q)));

E_p3o # Price of domestic/foreign composite, User 3 #
                (all,i,COM)(all,q,REGDEST)
(TINY+PVAL3O(i,q))*p3o(i,q) = sum(s,ALLSOURCE,PVAL3A(i,s,q)*p3a(i,s,q));

E_p3c # Price of domestic composite, User 3 #
                (all,i,COM)(all,q,REGDEST)
(TINY+PVAL3T(i,"domestic",q))*p3c(i,q)
                                   = sum(s,REGSOURCE,PVAL3A(i,s,q)*p3a(i,s,q));

E_x3c # Demand for domestic composite, User 3 #
                (all,i,COM)(all,q,REGDEST)
x3c(i,q) = x3o(i,q) - SIGMA3O(i)*(p3c(i,q)-p3o(i,q));
```

! Subsection 2.8.5: Tax rates !

```
E_deltax1 # Tax rate on sales to User 1 #
              (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
deltax1(i,s,j,q) = deltax(i) + deltax1all + deltaxsource(s)+deltaxdest(q);

E_deltax2 # Tax rate on sales to User 2 #
              (all,i,COM)(all,s,ALLSOURCE)(all,j,IND)(all,q,REGDEST)
deltax2(i,s,j,q) = deltax(i) + deltax2all + deltaxsource(s)+deltaxdest(q);

E_deltax3 # Tax rate on sales to User 3 #
              (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
deltax3(i,s,q) = deltax(i) + deltax3all + deltaxsource(s)+deltaxdest(q);

E_deltax4 # Tax rate on sales to User 4 #
              (all,i,COM)(all,s,REGSOURCE)
deltax4(i,s)=deltax(i)+deltax4all+deltaxsource(s)+deltaxdest("foreign");

E_deltax5 # Tax rate on sales to User 5 #
              (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
deltax5(i,s,q) = deltax(i) + deltax5all + deltaxsource(s)+deltaxdest(q);

E_deltax6 # Tax rate on sales to User 6 #
              (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
deltax6(i,s,q) = deltax(i) + deltax6all + deltaxsource(s)+deltaxdest("federal");
```

! Subsection 2.8.6: Purchasers' prices of commodities !

```
E_p1a # Purchasers prices - User 1 #
              (all,i,COM)(all,j,IND)(all,q,REGDEST)(all,s,ALLSOURCE)
(TINY+PVAL1A(i,s,j,q))*p1a(i,s,j,q) =
  [BAS1(i,s,j,q)+TAX1(i,s,j,q)]*p0a(i,s)
 + BAS1(i,s,j,q)*deltax1(i,s,j,q)     + sum(r,MARGCOM,MAR1(i,s,j,q,r)*p0a(r,q));

E_p2a # Purchasers prices - User 2 #
              (all,i,COM)(all,j,IND)(all,q,REGDEST)(all,s,ALLSOURCE)
(TINY+PVAL2A(i,s,j,q))*p2a(i,s,j,q) =
  [BAS2(i,s,j,q)+TAX2(i,s,j,q)]*p0a(i,s)
 + BAS2(i,s,j,q)*deltax2(i,s,j,q)     + sum(r,MARGCOM,MAR2(i,s,j,q,r)*p0a(r,q));

E_p3a # Purchasers prices - User 3 #
              (all,i,COM)(all,q,REGDEST)(all,s,ALLSOURCE)
(TINY+PVAL3A(i,s,q))*p3a(i,s,q) =
  [BAS3(i,s,q)+TAX3(i,s,q)]*p0a(i,s)
 + BAS3(i,s,q)*deltax3(i,s,q)     + sum(r,MARGCOM,MAR3(i,s,q,r)*p0a(r,q));

E_p4r # Purchasers prices - User 4 #
              (all,i,COM)(all,s,REGSOURCE)
(TINY+PVAL4R(i,s))*(natphi+p4r(i,s)) =
  [BAS4(i,s)+TAX4(i,s)]*p0a(i,s)
 + BAS4(i,s)*deltax4(i,s)     + sum(r,MARGCOM,MAR4(i,s,r)*p0a(r,s));

E_p5a # Purchasers prices - User 5 #
              (all,i,COM)(all,q,REGDEST)(all,s,ALLSOURCE)
(TINY+PVAL5A(i,s,q))*p5a(i,s,q) =
  [BAS5(i,s,q)+TAX5(i,s,q)]*p0a(i,s)
 + BAS5(i,s,q)*deltax5(i,s,q)     + sum(r,MARGCOM,MAR5(i,s,q,r)*p0a(r,q));

E_p6a # Purchasers prices - User 6 #
              (all,i,COM)(all,s,ALLSOURCE)  (all,q,REGDEST)
(TINY+PVAL6A(i,s,q))*p6a(i,s,q) =
  [BAS6(i,s,q)+TAX6(i,s,q)]*p0a(i,s)
 + BAS6(i,s,q)*deltax6(i,s,q)     + sum(r,MARGCOM,MAR6(i,s,q,r)*p0a(r,s));
```


! Subsection 2.8.7: Tax revenues !

```
E_taxrev1 # Aggregate revenue from indirect taxes levied on flows to User 1 #
              (all,q,REGDEST)
AGGTAX1(q)*taxrev1(q) = sum(i,COM, sum(s,ALLSOURCE, sum(j,IND,
                                    TAX1(i,s,j,q)*(p0a(i,s)+x1a(i,s,j,q))
                                  + BAS1(i,s,j,q)*deltax1(i,s,j,q) )));

E_nattaxrev1 # National revenue from indirect taxes levied on flows to User 1 #
NATAGGTAX1*nattaxrev1 = sum(q,REGDEST,AGGTAX1(q)*taxrev1(q));

E_taxrev2 # Aggregate revenue from indirect taxes levied on flows to User 2 #
              (all,q,REGDEST)
AGGTAX2(q)*taxrev2(q) = sum(i,COM, sum(s,ALLSOURCE, sum(j,IND,
                                    TAX2(i,s,j,q)*(p0a(i,s)+x2a(i,s,j,q))
                                  + BAS2(i,s,j,q)*deltax2(i,s,j,q) )));

E_nattaxrev2 # National revenue from indirect taxes levied on flows to User 2 #
NATAGGTAX2*nattaxrev2 = sum(q,REGDEST,AGGTAX2(q)*taxrev2(q));

E_taxrev3 # Aggregate revenue from indirect taxes levied on flows to User 3 #
              (all,q,REGDEST)
AGGTAX3(q)*taxrev3(q) = sum(i,COM, sum(s,ALLSOURCE,
                                    TAX3(i,s,q)*(p0a(i,s)+x3a(i,s,q))
                                  + BAS3(i,s,q)*deltax3(i,s,q) ));

E_nattaxrev3 # National revenue from indirect taxes levied on flows to User 3 #
NATAGGTAX3*nattaxrev3 = sum(q,REGDEST,AGGTAX3(q)*taxrev3(q));

E_taxrev4 # Aggregate revenue from indirect taxes levied on flows to User 4 #
              (all,s,REGSOURCE)
AGGTAX4(s)*taxrev4(s) = sum(i,COM,TAX4(i,s)*(p0a(i,s)+x4r(i,s))
                                  + BAS4(i,s)*deltax4(i,s) );

E_nattaxrev4 # National revenue from indirect taxes levied on flows to User 4 #
NATAGGTAX4*nattaxrev4 = sum(s,REGSOURCE,AGGTAX4(s)*taxrev4(s));

E_taxrev5 # Aggregate revenue from indirect taxes levied on flows to User 5 #
              (all,q,REGDEST)
(TINY+AGGTAX5(q))*taxrev5(q) = sum(i,COM, sum(s,ALLSOURCE,
                                      TAX5(i,s,q)*(p0a(i,s)+x5a(i,s,q))
                                    + BAS5(i,s,q)*deltax5(i,s,q) ));

E_nattaxrev5 # National revenue from indirect taxes levied on flows to User 5 #
(TINY+NatAGGTAX5)*nattaxrev5 = sum(q,REGDEST,AGGTAX5(q)*taxrev5(q));

E_taxrev6 # ggregate revenue from indirect taxes levied on flows to User 6 #
              (all,q,REGDEST)
(TINY+AGGTAX6(q))*taxrev6(q) = sum(i,COM,sum(s,ALLSOURCE,
                                      TAX6(i,s,q)*(p0a(i,s)+x6a(i,s,q))
                                    + BAS6(i,s,q)*deltax6(i,s,q) ));

E_nattaxrev6 # National revenue from indirect taxes levied on flows to User 6 #
(TINY+NATAGGTAX6)*nattaxrev6 = sum(q,REGDEST,AGGTAX6(q)*taxrev6(q));
```

! Subsection 2.8.8: Demands for exports !

```
E_x4r # Export demand functions #
              (all,i,COM)(all,s,REGSOURCE)
x4r(i,s) - feq(i) = EXP_ELAST(i)*[p4r(i,s) - fep(i) - natfep];
```

! Subsection 2.8.9: Demands for commodities for regional Other expenditure !

```
E_x5a # Regional Other demands #
              (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
x5a(i,s,q) = cr(q) + f5a(i,s,q) + f5gen(q) + natf5gen;
```

! Subsection 2.8.10: Demands for commodities for Fed. Other expenditure !

```
E_x6a # Fed. Other demands #
              (all,i,COM)(all,s,ALLSOURCE)(all,q,REGDEST)
x6a(i,s,q) = natcr + f6a(i,s,q) + f6gen(q) + natf6gen;
```