STAGE_t: A User Guide

Scott McDonald
Feb 2024

Abstract

This paper is a user's guide to the STAGE_t CGE model: it is not a manual for the learning about STAGE or CGE models in general. The User Guide assumes that the user is familiar with the structure of the model database, the behavioural relationships in the model and CGE models implemented using the GAMS programming language. The User Guide refers to an implementation of the model that uses GAMS Studio as the text editor, GDX as the source of transactions data and destination of the model results, and MS Excel – in conjunction with GDXXRW - as the source data relating to sets and various exogenous parameters. It is assumed that the user will present the (transactions) database in a SAM format and access the model results using GDX. This User Guide does not provide any guidance on how to frame policy experiments using CGE models.

This is a draft that is undergoing continuing development: it will never satisfy everyone. It is provided on that basis. Comments on the current content are encouraged.

Email: scott@cgemod.org.uk
Url: www.cgemod.org.uk

Table of Contents

1. Introduction

STAGE_t is a ‘‘STandard’ Applied General Equilibrium’ model that is implement in the General Equilibrium Model System (GAMS) and calibrated using a Social Accounting Matrix (SAM) and associated satellite accounts. The STAGE_t variant has been adapted to meet pedagogic requirements consistent with cgemod’s Single Country CGE modelling course. It is nevertheless a fully functioning single country CGE model with many modern features that allows users to easily progress to using the latest versions of STAGE. STAGE_t is a ‘template’ model, so users are encouraged to adapt and develop variants of the STAGE_t model.

This User Guide is intended to support use of the STAGE_t model. It is not an objective of this guide to provide technical information about the STAGE_t model, nor is it an objective to provide introductions to the principles of CGE modelling or the mechanics of the General Algebraic Modelling System (GAMS). The guide provides details about the structure of the model and experiment programmes, the data, aggregating a SAM, setting up and configuring the model, setting up and configuring an experiment file and arranging data for the model and experiments.

The STAGE_t model has been developed using GAMS Studio as the editor.[1] This guide presumes that users of STAGE_t will use GAMS Studio as the editor; all screen shots for the User Guide are from applications running in Studio.

The User Guide is compiled under the presumption that users will implement experiments/simulations using GAMS’s ‘save and restart’ facility. This entails implementing the base model from the file `stg_**.gms` using the SAVE facility with simple and flexible macroeconomic closure and factor market clearing conditions. Experiments are then implemented using the file `stg_expt_**.gms` and the RESTART facility that can support multiple experiment files (`expt_***.inc`) implemented in a system of three loops – sensitivity loop, closure loop and simulation loop. Each experiment file is be calibrated from an Excel file.

[1] Previously the STAGE model was developed using GAMSIDE, while the earlier development of the PROVIDE/STAGE model used Programmers File Editor.

The rest of this Guide is organised as follows. The next section, 2, provides information about the file structure used for the model's code and reviews three facilities in GAMS Studio that make accessing parts of the code easy. The third section reviews the SAM and ancillary data used by the model, and this followed, section 4, with guidance on compiling an Excel workbook for use with this model family.

2. Directory and File Structure

The STAGE_t model adopts the multiple directory structure used in the later STAGE and ANARRES models, see Table 1. At first sight this arrangement may appear difficult to navigate, but by using features available in GAMS Studio it is straightforward while avoiding many files in a single directory.

It is my **preference** to avoid overly large files of code – they can become cumbersome – and to retain the option to switch code in and out of the programme without needing to create a completely new (large) file.

2.1 Directory Structure and Using $SETGLOBAL

The first thing to note is that the directory contains TWO `***.gms` files – see Figure 2.1.1. This reflects the presumption that users will run TWO projects from the directory: the base model – `stg_t.gms` – project and the experiment model – `stg_t_expt.gms` – project. Each of project is supported by a data subdirectory: `1_1_Model_data` can contain the data for multiple applications, e.g., different countries and/or different aggregations and/or different model configurations, while `1_2_Expt_data` can contain the data for multiple experiments, e.g., trade or policies, applied to different applications. The codes for the base model and base experiment files use the `$SETGLOBAL`[2] option to select the data used. This avoids the need to edit codes in various places in the base model or experiment files. The base experiment file also uses `$SETGLOBAL` to identify the experiment used and condition the files that carry out post solution analysis (see Figure 2.2).

[2] Guidance on the use of $SEGLOBAL can be found in the GAMS Studio User Guide (http://www.cgemod.org.uk/gams_studio.html).

Table 2.1.1 STAGE_t Directory Structure

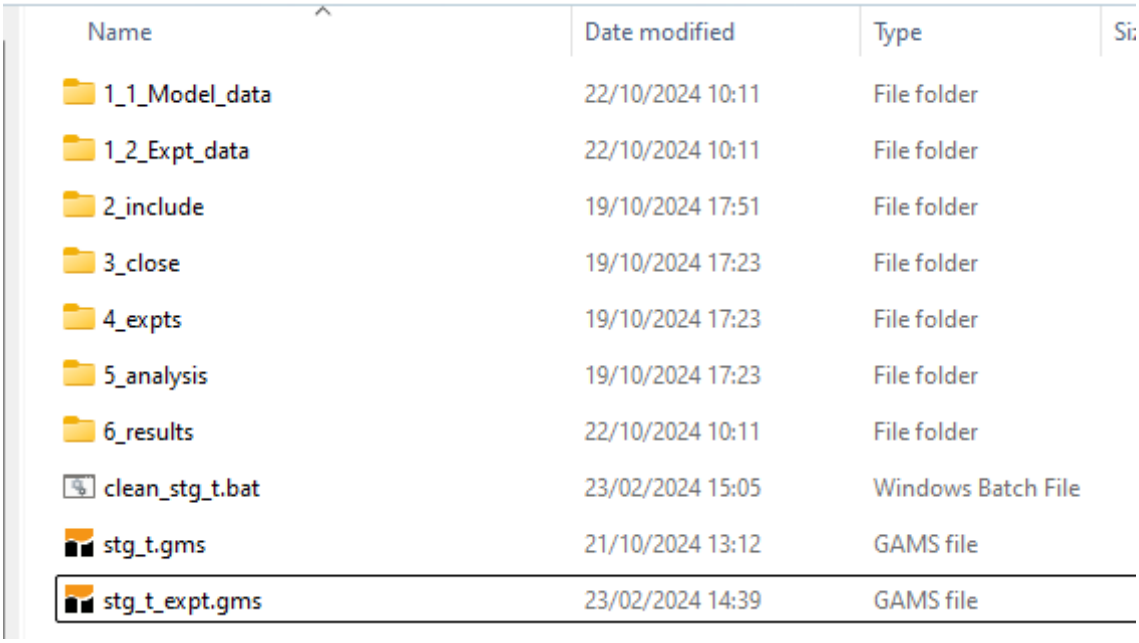


Figure 2.1.2 $SETGLOBAL in Base Experiment File

```
21  *#### SETTING DIMENSIONS FOR THE RESULTS FILES
22  * Set dimensions for the results analysis files
23
24  $SETGLOBAL res_dimen      simc,clos
25
26  * Set experiment File
27
28  $SETGLOBAL expt_inc   stg_t_expt_base_2
29
30  * Set experiment data
31
32  $SETGLOBAL expt_data   stg_t_expt_7_6
33
```

The subdirectory `2_include` contains the various `***.inc` files, e.g., `stg_t_parmcalib.inc`, used by the model file `stg_t.gms`. All these files can be opened directly from the `stg_t.gms` file by using `Ctrl+Click` on the link to the chosen include file, e.g.,

```
*################ 8. MODEL CALIBRATION ###################################
* Calibration proceeds by the blocks of equations
* each block contains all the related parameter declarations & assignments

$INCLUDE 2_include/stg_t_parmcalib.inc
```

This option means that the core GMS file also operates as 'indexing system' in Studio and thereby avoids needing to search the subdirectories to find files.

Similarly, the subdirectory `3_close` can contain multiple closure files that can be used by different experiments. In this instance the `$SETGLOBAL` option is not used, but rather the closure file choices are set manually for each experiment; this choice was made because in this instance it was deemed that using $`$SETGLOBAL` was fiddly, too liable to produce errors and not robust.

The subdirectory `4_expts` is a repository of experiment files that can be called from the base experiment file – `stg_t_expt.gms` – by using `$SETGLOBAL`. This system makes it easy to store multiple experiment files and easily move between different experiments. Storing experiment files is a good practise to acquire, it allows you to replicate experiments and adjust experiments without needing to reproduce the experiment file.

The subdirectory `5_analysis` contains a series of files that process the results from experiments and write out the processed results to GDX files. The selection of analysis files included with `stg_t.gms` report the levels and percentage changes for all model variables, and selections of summary macroeconomic and welfare results. The dimensions of the result parameters are assigned using $SETGLOBAL, see below

```
* Table of scalar results
 resSCAL(scalres,%res_dimen%)   table of scalar results
* Table of adjuster results
 resADJ(adjres,%res_dimen%)     table of adjuster results
* ------- TRADE BLOCK ----------------------------------------------------
* #### Exports Block
 resPWE(c,%res_dimen%)        World price of exports in dollars
 resPE(c,%res_dimen%)         Domestic price of exports by activity a
 resPD(c,%res_dimen%)         Consumer price for domestic supply of commodity c
 resQE(c,%res_dimen%)         Domestic output exported by commodity c
 resQD(c,%res_dimen%)         Domestic demand for commodity c
```

Similarly, the analysis files make use of $SETGLOBAL when calibrating the results parameters, see below

```
resADJ("aADVAADJ",%res_dimen%)$ADVAADJ0
                          = (resADJ("aADVAADJ",%res_dimen%)/ADVAADJ0-1)*100 ;
 resADJ("aDADVA",%res_dimen%)$DADVA0
                          = (resADJ("aDADVA",%res_dimen%)/DADVA0-1)*100 ;
* ------- TRADE BLOCK ----------------------------------------------------
* #### Exports Block
 resPWE(c,%res_dimen%)$PWE0(c)   = (resPWE(c,%res_dimen%)/PWE0(c)-1)*100 ;
 resPE(c,%res_dimen%)$PE0(c)     = (resPE(c,%res_dimen%)/PE0(c)-1)*100 ;
 resPD(c,%res_dimen%)$PD0(c)     = (resPD(c,%res_dimen%)/PD0(c)-1)*100 ;
```

This method allows for dynamic changes to the results files according to the specific experiments being implemented.

The final subdirectory 6_results is where the output from the analyses of the results is recorded. For the course the results are all recorded in the subdirectory 6_results, e.g.,

```
*################### EXPORT OF RESULT FILES TO GDX ####################

* This writes the results data to GDX

Execute_Unload  '6_results/stg_t_percent.gdx',

* Table of scalar results
 resSCAL,
```

this is an acceptable practice since storing the experiment files means that results can be readily recreated.

However, it is often good practice to store the results from experiments in discrete subdirectories that are labelled to link them to specific directories. This can be done easily by manually copying results from each experiment to specific subdirectories: this is the method used in this course.

Alternatively, an extra `$SETGLOBAL` option can be used to label results and write them out to a specified subdirectory, e.g.,

```
* This writes the results data to GDX

Execute_Unload  '6_results/%expt1%/%expt1%_stg_t_percent.gdx',

* Table of scalar results
 resSCAL,
```

where ‘expt1’ is a $SETGLOBAL string that can be assigned at the beginning of the file stg_t_expt.gms.

Note that this requires the creation of the subdirectory whose name is that defined by the $SETGLOBAL option. In this illustration the string substitution was used twice – once the determined the destination subdirectory and then the prefix the name of the GDX file. (A risk averse tactic.)

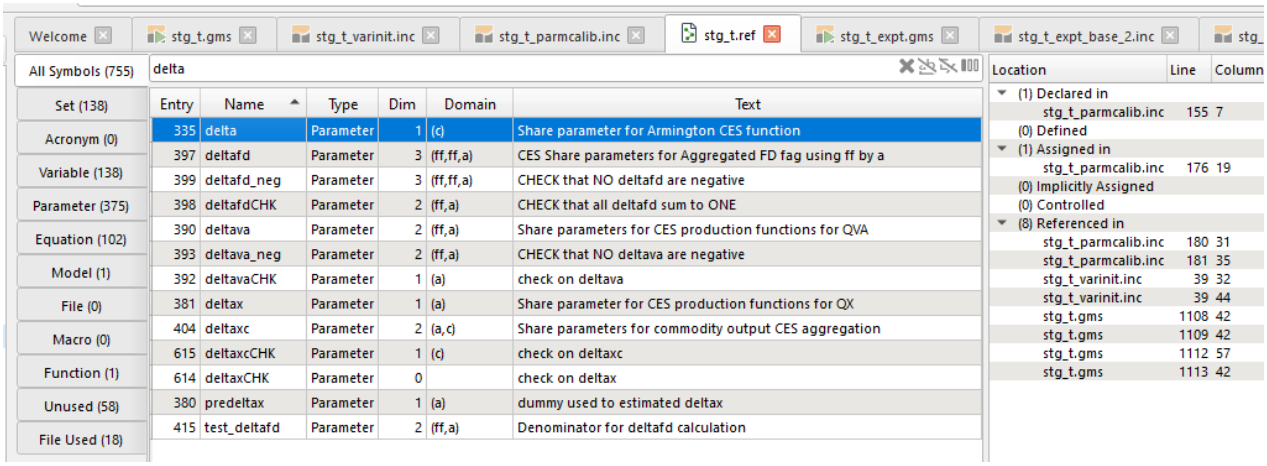
2.2 Useful Navigation Facilities in GAMS Studio

GAMS and GAMS Studio provide several features that aid navigated programme files. In no order, there are three facilities that are especially useful: reference files, find and search and Pin Right/Pin Below.

Reference Files

References files are a staple product of GAMS; they are so useful we recommend that they are created every time a model is run (use the GAMS>GAMS Default Configuration menu and set reference as a default command line parameter). Typically, when using/searching a reference file the user is guided by the model equations

On opening a reference file, you can search for a specific symbol. The screenshot below illustrates a search for the parameters containing the word ‘delta’, where the presumption is that the user starts by wanting to know where the symbol delta is declared and assigned in the code. For each variant of delta selected the Location pane identifies all references to that variant of delta. Doubling clicking on items in the Location pane will open the file in which delta appears at the point in the files where delta is referenced.

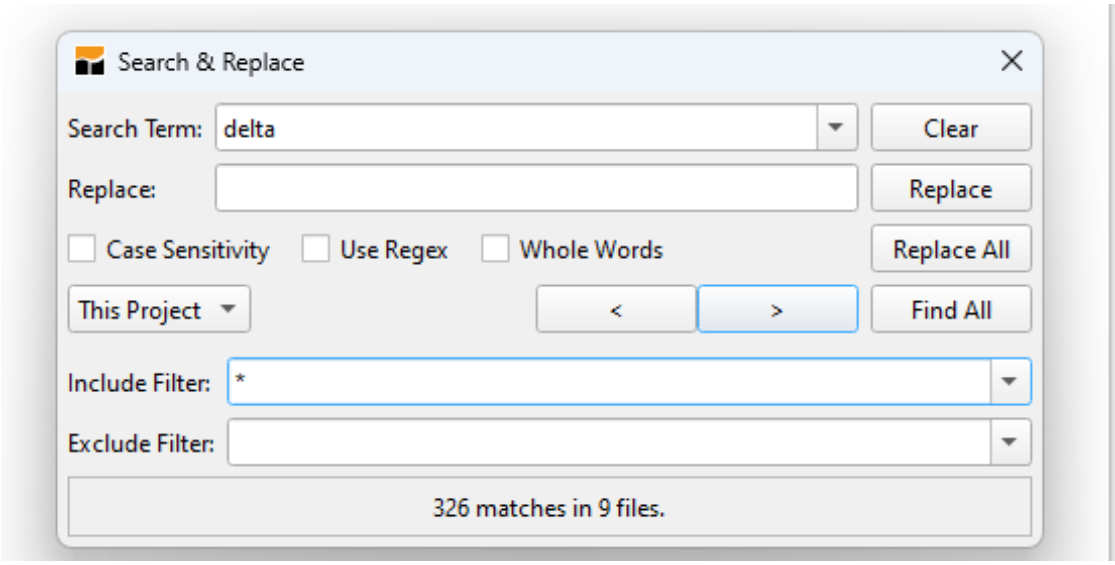


The reference file method is especially useful when learning about a model.

Find and Search

The Find and Search system in Studio allows the user to search through one or multiple files. In the illustration below the search term is for `delta` across all files in 'This Project', which is one of six options, and returned 326 matches. Searches can be refined using various options, e.g., if searching for all occurrences of a variable the option of Case Sensitivity may be chosen while using Case Sensitivity to search for delta reduced the matches to 32.

The forward and reverse arrows move the view through the matches starting from the point in the file where the search was initiated. The Process Pane provides more information about the matches.



The Search and Find method is especially useful when the user know what they are looking, i.e., they have a good understanding of a model.

Pin Right/Pin Below

The Pin Right/Pin Below View (`View>Pin Right or Pin Below` – the choice will depend on the user screen characteristics and case specific needs) option allows the user to view different parts of the code. The illustration below assumes that the user wants to view simultaneously the `QVAPRODFN` and `QVAFOC` equations in `stg_t.gms`, and the code for calibrating the parameters in those equations in the file `stg_t_parmcalib.inc`. In addition, the pin view can be used to view simultaneously different parts of the code within a single file.[3]

An advantage of the pin view option is that the user can edit the content of both files while in pin view. Its disadvantage is that it requires the user to have a good understanding of the model codes.

[3] Thus, if users want to embed the content of all the include files in the master file, the pin view allows the user to work on different parts of the file at the same time without a lot of scrolling.

The Pin View option is especially useful when the user is modifying the model since it allows the user to adjust different parts of the code at the same time.

3. Data

3.1 Social Accounting Matrix

The model is designed for calibration using a Social Accounting Matrix (SAM) that broadly conforms to the UN System of National Accounts (SNA). Since the databases for ALL whole economy models can always be represented in a SAM format, and some modellers choose to present their model databases in SAM format, such a transformation has attractions. The three most obvious attractions are:

i) the increased use by economists of the SAM format, especially now that it is formally part of the System of National Accounts (SNA) (UN, 1993 and 2008);

ii) the greater ease with which the data for a single region can be assessed and related to national account aggregates; and

the (arguably) greater accessibility of the information for policy makers.

Table 3.1 contains a macro-SAM in which the active sub matrices are identified by *X* and the inactive sub matrices are identified by 0. In general, the model will run for any SAM that contains information in the active sub matrices and conforms to the rules of a SAM.[4] In some cases a SAM might contain payments from and to both transacting parties, in which case recording the transactions as net payments between the parties will render the SAM consistent with the structure laid out in Table 1.

The most notable differences between this SAM and one fully consistent with the SNA are:

1) The SAM is assumed to contain only a single 'level' of income distribution. (SAMs consistent with multi-level income distributions, as outlined in the SNA can be readily transformed using apportionment (see Pyatt, 1989)).

2) A series of tax accounts are identified (see below for details), each of which relates to specific tax instruments. Thereafter a consolidated government account is used to bring together the different forms of tax revenue and to record

[4] If users have a SAM that does not run with no information in inactive sub matrices the author would appreciate a copy of the SAM to further generalise the model.

government expenditures. These adjustments do not change the information content of the SAM, but they do simplify the modelling process. However, they do have the consequence of creating a series of reserved names that are required for the operation of the model.[5]

Table 3.1 Macro SAM for the Standard Model

	Commodities	Activities	Factors	Households	Enterprises	Government	Capital Accounts	RoW
Commodities	0	X	0	X	X	X	X	X
Activities	X	0	0	0	0	0	0	0
Factors	0	X	0	0	0	0	0	X
Households	0	0	X	0	X	X	0	X
Enterprises	0	0	X	0	0	X	0	X
Government	X	X	X	X	X	0	0	X
Capital Accounts	0	0	X	X	X	X	0	X
RoW	X	0	X	X	X	X	X	0
Total	X	X	X	X	X	X	X	X

A SAM is a transactions matrix; hence each cell in a SAM simply records the values of the transactions between the two agents identified by the row and column accounts. The selling agents are identified by the rows, i.e., the row entries record the incomes received by the identified agent, while the purchasing agents are identified by the columns, i.e., the column entries record the expenditures made by agents. As such a SAM is a relatively compact form of double entry bookkeeping that is complete and consistent and can be used to present the National Accounts of a country in a single two-dimensional matrix (see UN, 1993, for a detailed explanation of the relationship between conventional and SAM presentations of National Accounts). A SAM is *complete* in the sense that the SAM should record ALL the transactions within the production boundary of the National Accounts, and *consistent* in the sense that income transactions by every agent are exactly matched by expenditure transactions

[5] These and other reserved names are specified below as part of the description of the model.

of other agents. A fundamental consequence of these conditions is that the row and column totals of the SAM for each region must be identical, and hence the SAM provides a complete characterisation of current account transactions of an economy as a circular (flow) system.

Given these definitions of a SAM the transactions recorded in a SAM are readily interpreted. In Table 3.1 the row entries for the commodity accounts are the values of commodity sales to the agents identified in the columns, i.e., intermediate inputs are purchased by activities (industries etc.,), final (consumption) demand is provided by households, the government and investment demand and export demand is provided by the all the other regions in the global SAM and the export of margin services. The commodity column entries deal with the supply side, i.e., they identify the accounts from which commodities are purchased to satisfy demand. Specifically, commodities can be purchased from either domestic activities – the domestic supply matrix valued inclusive of domestic trade and transport margins – or they can be imported – valued exclusive of international trade and transport margins. In addition to payments to the producing agents – domestic or foreign – the commodity accounts need to make expenditures with respect to the trade and transport services needed to import the commodities and any commodity specific taxes.

An important feature of the construction of a SAM can be deduced from the nature of the entries in the commodity account columns. The column and row totals must equate, and these transaction totals can be expressed as an implicit price multiplied by a quantity, and the quantity of a commodity supplied must be identical to the quantity of a commodity demanded. The column entries represent the expenditures incurred to supply a commodity to the economy and hence the implicit (average) price must be exactly equal to the average cost incurred to supply a commodity. Moreover, since the row and column totals equate, and the quantity represented by each corresponding entry must be the same for the row and column totals the implicit price for the row total must be identical to average cost incurred to supply the commodity. Hence the column entries identify the components that enter the formation of the implicit prices in the rows, and therefore identify the price formation process for each price in the system. Typically, a SAM is defined such that the commodities in the rows are homogenous and that all agents purchase a commodity at the same price.

The model contains a section of code, immediately after the data have been read in, that resolves a number of common 'problems' encountered with SAM databases by transforming the SAM so that it is consistent with the model structure. Specifically, all transactions between an account with itself are eliminated by setting the appropriate cells in the SAM equal to zero. Second, all transfers from domestic institutions to the Rest of the World and between the Rest of the World and domestic institutions are treated net as transfers to the Rest of the World and domestic institutions, by transposing and changing the sign of the payments to the Rest of the World. And third, all transfers between domestic institutions and the government are treated as net and as payments from government to the respective institution. Since these adjustments change the account totals, which are used in calibration, the account totals are recalculated within the model.

Ancillary programmes are available to

1. aggregate a SAM;
2. removal minor differences the row and column totals of the SAM; and
3. estimate a SAM from incomplete and imperfect data (using entropy).

3.2 Satellite Account Data

In addition to the SAM, which records transactions in value terms, there are additional data that can be used by the model. These data are record as satellite accounts, where, by definition, the dimensions of each satellite account are consistent with sub matrices of the SAM. Typically, these accounts will record data on 'quantities' related to the transactions recorded in the SAM.

The first two satellite accounts are critical to the model and record factor quantities. These are

1. FACTUSE: this satellite account records the (natural) quantities of factors used by each activity, i.e., the dimensions are $(f*a)$[6]; and
2. FACTINS: this satellite account records the (natural) quantities of factors owned by each representative household group (RHG), i.e., the dimensions are $(h*f)$[7].

[6] f is the number of natural factors and a the number of activities.
[7] h is the number of RHGs.

If the factor quantity data are not available, the 'values' used in the model are derived from the transactions data using the Harberger convention, i.e., it is assumed that the 'value' quantities derived from the transaction values and normalised prices are proportionate to physical quantities. The process is automatically implemented in the model if the FACTUSE and/or FACTINS data are not recorded in the model database.

The model is informed about the availability of these satellite accounts by controlling parameters that are configured in the model's Excel database.

3.3 Exogenous Model Data

All CGE models require exogenous elasticity data for the functional forms used to define behavioural relationships; the models will only be implemented if these data are made available to the model (or are implicit in the model's behavioural relationships[8]). These are

1. ELASTC: Elasticities indexed on commodities for imports, exports, export demand functions and differentiated commodities produced in the economy, dimensions are (*c*,**);
2. ELASTX: Elasticities indexed on activities for first and second level of production systems and output mix (of commodities) by activities, dimensions are (*a*,**);
3. ELASTF: Elasticities for CES Production function level 3 and below, dimensions are (*fag***a*)[9];
4. ELASTY: Income Demand Elasticities in Linear Expenditure System (LES) for households, dimensions are (c**h*);
5. ELASTMU: Elasticity of the MU of income (Frisch parameter), dimensions are (*h**1); and
6. ELASTCES: Elasticities of substitution in CES functions for households, dimensions are (*cag***h***1*)[10].

These data are rarely supported by empirical estimates that are country specific or for groups of countries.

[8] Cobb-Douglas functions have elasticities equal to ONE.
[9] *fag* is the number of aggregates in the production system in levels 3 and below.
[10] *cag* is the number of aggregates in the utility system.

3.4 Database Presentation

All the data are accessed by the model from data recorded in Excel and GDX (GAMS data exchange) file. All the data recorded in Excel are converted into GDX format as part of the model.

4. Compiling the Model's Excel Workbook[11]

4.1 Excel Workbooks

The model uses two Excel workbooks to contain the information used by the model; the first workbook contains information used for calibrating the model, while the second provides information used to run simulations. This arrangement presumes the user will implement simulations/experiments from a dedicated Excel workbook. This section is concerned with the Excel workbook that has information used to calibrate the model.

The Excel workbook for the STAGE model typically has the following worksheets

1. Layout: instructs GDXXRW where data are stored in the workbook
2. Sets: 'static' model sets
3. Nest_va: mapping set for second level of production system
4. Nest_fagg: mapping set for third level of production system
5. stab_sets: sets used to report structural information about the model and data
6. res_sets: sets used to report results (these sets are used to report on model calibration and experiments)
7. controls: parameters used to control/condition aspects of the model
8. SAM: the SAM transactions data (can be stored separately in GDX)
9. factuse: factor use by activities (can be stored separately in GDX)
10. factins: factor ownership by institutions (can be stored separately in GDX)
11. pop: population data by RHG
12. comelast: elasticities relating to commodities
13. actelast: elasticities relating to activities
14. elastf: elasticities used in level 3 and below of the production system
15. frischelast: Frisch parameters for the LES

[11] GAMS uses a programme, GDXXRW, to convert the information in an Excel workbook into a GDX file using information drawn from a worksheet in the workbook – for STAGE this 'master' worksheet is called 'Layout'. The 'Layout' worksheet contains information that instructs GDXXRW about the 'Data Type', 'Name (of the data)', 'Location (of the data)', 'Row dimension (of the data)', 'Column dimension (of the data)', 'Total dimension (of the data)'. If the user chooses to put data in different places to those in the sample Excel workbook errors will be generated unless the 'Layout' workbook is modified. For instructions about using GDXXRW see the help menu in GAMS Studio (> Help > docs> gams > gdxutils.chm or gdxutils.pdf).

16. ihoelast: income elasticities of demand for the LES3.1 Layout Sheet

The syntax for the 'layout' worksheet is described in the gdx utilities documentation supplied with GAMS. Unless the user wants to pass ADDITIONAL sets and data to the model there is no reason for the user to alter the 'layout' worksheet. If changes are made to the structure of the database, it is important to ensure that all the syntax etc., is fully consistent; this is especially the case when working with GDXXRW since the error messages may be slightly opaque. If the user does get errors, then it is wise to review the associated *.log file since the detail therein is the most comprehensive available.

Figure 4.1.1 Data 'Layout' Worksheet

	A	B	C	D	E	F	G	H
1	LAYOUT							
2								
3	Data Type	Name	Location	Row dimension	Column dimension	Total dimension		
4				rdim	cdim	dim		
5								
6	dset	sac	Sets!A6	1				
7	dset	ss	Sets!D6	1				
8	dset	c	Sets!G6	1				
9	dset	cagr	Sets!H6	1				
10	dset	cnat	Sets!I6	1				
11	dset	cfd	Sets!J6	1				
12	dset	cind	Sets!K6	1				
13	dset	cuti	Sets!L6	1				
14	dset	ccon	Sets!M6	1				
15	dset	cser	Sets!N6	1				

See section 6.1 (Data Entry Section) for guidance on reading information from the layout sheet and triggering information in the GAMS log file to interpret any issues that arise when reading data into GDX and/or GAMS from Excel.

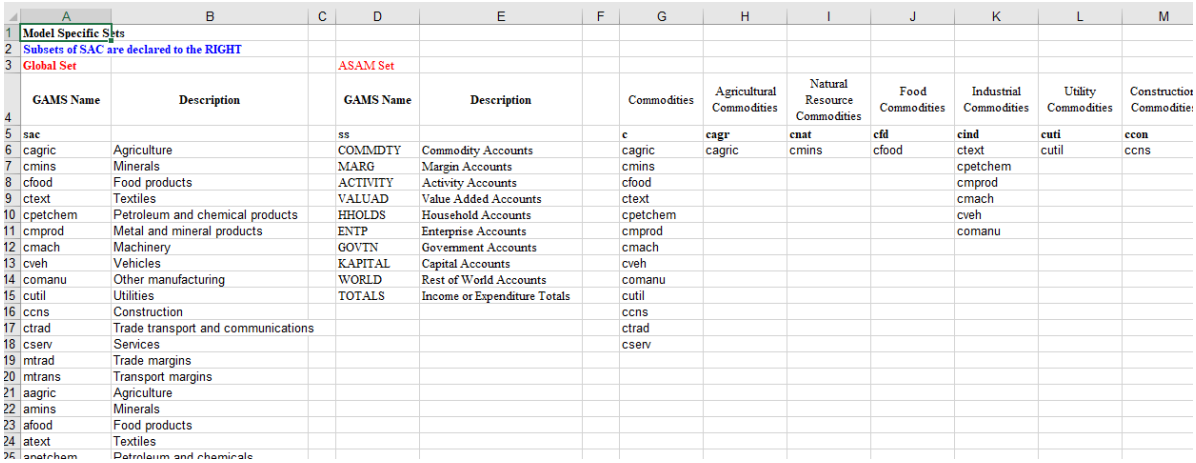
4.2 Model Sets

4.2.1 Sets Worksheet

The 'sets' worksheet contains most of the sets and subsets used by the model. There is a 'global' set, *sac*, that contains all the accounts for the SAM and other sets that are declared as subsets of I; this facilitates domain checking. It is important to ensure that the set member names and descriptions are in the correct cells in the workbook and worksheet as defined in

the layout worksheet, if not errors will be generated. The first few columns of the 'sets' worksheet are illustrated in shown in Figure 4.2.1.1.

Figure 4.2.1.1 Sets Worksheet Part 1



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Model Specific Sets**												
2	**Subsets of SAC are declared to the RIGHT**												
3	**Global Set**			ASAM Set									
4	**GAMS Name**	**Description**		**GAMS Name**	**Description**		Commodities	Agricultural Commodities	Natural Resource Commodities	Food Commodities	Industrial Commodities	Utility Commodities	Construction Commodities
5	**sac**			**ss**			**c**	**cagr**	**cnat**	**cfd**	**cind**	**cuti**	**ccon**
6	cagric	Agriculture		COMMDTY	Commodity Accounts		cagric	cagric	cmins	cfood	ctext	cutil	ccns
7	cmins	Minerals		MARG	Margin Accounts		cmins				cpetchem		
8	cfood	Food products		ACTIVITY	Activity Accounts		cfood				cmprod		
9	ctext	Textiles		VALUAD	Value Added Accounts		ctext				cmach		
10	cpetchem	Petroleum and chemical products		HHOLDS	Household Accounts		cpetchem				cveh		
11	cmprod	Metal and mineral products		ENTP	Enterprise Accounts		cmprod				comanu		
12	cmach	Machinery		GOVTN	Government Accounts		cmach						
13	cveh	Vehicles		KAPITAL	Capital Accounts		cveh						
14	comanu	Other manufacturing		WORLD	Rest of World Accounts		comanu						
15	cutil	Utilities		TOTALS	Income or Expenditure Totals		cutil						
16	ccns	Construction					ccns						
17	ctrad	Trade transport and communications					ctrad						
18	cserv	Services					cserv						
19	mtrad	Trade margins											
20	mtrans	Transport margins											
21	aagric	Agriculture											
22	amins	Minerals											
23	afood	Food products											
24	atext	Textiles											
25	apetchem	Petroleum and chemicals											

The model uses a substantial number of subsets of *sac*; some of these are shown in Figure 4.2.1.2. The main subsets are

- *c(sac)* commodity accounts
- *m(sac)* Margins
- *a(sac)* activity accounts
- *ff(sac)* factor accounts (natural and aggregates)
- *insa(sac)* all domestic institutions and rest of the world
- *h(sac)* household accounts
- *e(sac)* enterprise accounts
- *g(sac)* government accounts
- *gt(g)* tax accounts
- *i(sac)* investment accounts
- *w(sac)* Rest of world trade partners

The user needs to manually assign these subsets when setting up a new version of the STAGE model. HINT: copying and pasting can reduce typing errors. NB the STAGE model will abort if the accounts names in *sac* and ALL the subsets are NOT identical.

In addition to the subsets of *sac* there are several static subsets of the subsets of *sac*. These are

- *cagr(c)* Agricultural Commodities
- *cnat(c)* Natural Resource Commodities
- *cfd(c)* Food Commodities
- *cind(c)* Industrial Commodities
- *cuti(c)* Utility Commodities
- *ccon(c)* Construction Commodities
- *cser(c)* Service Commodities
- *cagg* Aggregate commodity groups
- *aagr(a)* Agricultural Activities
- anat(a) Natural Resource Activities
- afd(a) Food Activities
- aind(a) Industrial Activities
- auti(a) Utility Activities
- acon(a) Construction Activities
- aser(a) Service Activities
- aagg Aggregate activity groups
- anch(a) Anchor activity for fixing 1 WFDIST in various factor closures
- aleon(a) Activities with Leontief prodn function at Level 1
- ff(sac) factors factor commodities and aggregates
- f(ff) natural factor accounts
- fag(ff) aggregate factors
- l(f) Labour Factors
- ls(l) Skilled Labour Factors
- lm(l) Skilled or Unskilled Labour Factors
- lu(l) Unskilled Labour Factors
- k(f) Capital Factors
- n(f) Land Factors
- insa(sac) All Domestic Institutions and Rest of World
- insw(insa) Domestic Non Government Institutions and Rest of World
- insg(insa) Domestic Institutions including Government

- ins(insg) Domestic Non Government Institutions
- h(insa) Households
- g(sac) Government
- gt(g) Government tax accounts
- tff(g) factor tax account used in GDX program
- e(insa) Enterprises
- i(sac) Investment categories
- in(i) Investment categories excluding i_s
- w(insa) Rest of the world

The user needs to define the memberships of these subsets manually following conventions that are obvious from the naming of the subsets. HINT: copying and pasting reduces typing errors. Empty subsets are legitimate.[12]

NB: The are two sets *cagg* and *aagg* that are fixed, and the user should not change them unless they intend to make substantial changes throughout the files that are used to analyse results. If the user wants additional subsets for the analysis of specific aspects of the results they should declare and assign those subsets – typically in the experiment files and associated Excel workbooks.

Figure 4.1.1.2 Sets Worksheet Part 2 (Outtake)

	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	
1	Natural & Aggregate Factors	Natural Factors	Aggregate Factors	Factors Used in level 2	Labour Factors	skilled labour	semi-skilled labour	Capital Factors	Immobile capital factors	Land	ALL Domestic Instiutions and RoW	Domestic Instiutions and RoW	Domestic Institutions	Domestic Non Govt Institutions	Households	Government	Government Taxes	U
	ff	**f**	**fag**	**f2**	**l**	**ls**	**lm**	**k**	**kfx**	**n**	**insa**	**insw**	**insg**	**ins**	**h**	**g**	**gt**	**tfl**
	fcap_fix	fcap_fix	flab	fcap_fix	fafunsk	fafskl		fcap_fix	fcap_fix	fland	hafflow	hafflow	hafflow	hafflow	hafflow	imptax	imptax	tfl
	fcap_flex	fcap_flex		fcap_flex	fafskl	fcaskl		fcap_flex			haffmed	haffmed	haffmed	haffmed	haffmed	exptax	exptax	tfl
	fland	fland		fland	fcaunsk	fwhskl					haffhigh	haffhigh	haffhigh	haffhigh	haffhigh	saltax	saltax	tfl
	fafunsk	fafunsk		flab	fcaskl						hafmlow	hafmlow	hafmlow	hafmlow	hafmlow	ssaltax	ssaltax	tfl
	fafskl	fafskl			fwhunsk						hafmmed	hafmmed	hafmmed	hafmmed	hafmmed	vattax	vattax	tfl
	fcaunsk	fcaunsk			fwhskl						hafmhigh	hafmhigh	hafmhigh	hafmhigh	hafmhigh	ectax	ectax	tfl
	fcaskl	fcaskl									hcalow	hcalow	hcalow	hcalow	hcalow	indtax	indtax	tfl
	fwhunsk	fwhunsk									hcahigh	hcahigh	hcahigh	hcahigh	hcahigh	tffcap_fix	tffcap_fix	tfl
	fwhskl	fwhskl									hwhlow	hwhlow	hwhlow	hwhlow	hwhlow	tffcap_flex	tffcap_flex	tfl
	flab										hwhhigh	hwhhigh	hwhhigh	hwhhigh	hwhhigh	tffland	tffland	
											ent	ent	ent	ent		tffafunsk	tffafunsk	
											govt	row	govt			tffafskl	tffafskl	
											row					tffcaunsk	tffcaunsk	
																tffcaskl	tffcaskl	
																tffwhunsk	tffwhunsk	
																tffwhskl	tffwhskl	
																facttax	facttax	
																dirtax	dirtax	
																govt		

[12] STAGE also uses GAMS ONEMPTY option to facilitate set declaration and assignment.

4.3 Maps Worksheets

The model uses several mapping sets. Three are used allow the matching of data that are recorded using different but related labels, while two mapping sets are used to control the nested of the production system. allow the matching of data that are recorded using different but related labels.

The standard format for a mapping set is '`map_set1_set2(set1,set2)`'. '*`map_set1_set2`*' is the name of the mapping set, and (*`set1,set2`*) define the two sets that are mapped where *`set1`* is the destination set and *`set2`* is the source set. Thus '`map_set1_set2(set1,set2)`' defines the mapping of the elements in *`set2`* onto the elements in *set1*.

Maps with Related Labels

The generation of the first three of the required mapping sets has been simplified in GAMS. Specifically, a mapping set can be defined directly from a pair of related labels using the assignment statement `#x:#tx` if the mapping set has been declared. Below is an example where the mapping sets are defined, e.g., `map_f_tff(f,tff)` and then assigned, e.g., `#f:#tff`, in one block.

```
Sets
 map_f_tff(f,tff) Factor taxes to factors /
 #f:#tff
  /
 map_tff_f(tff,f) Factor taxes to factors reverse  /
 #tff:#f
  /
 map_i_k(in,k)     Mapping from investment type to capital type /
 #in:#k
  /
 ;
```

The weakness of this method is its reliance on the sets being ordered consistently, e.g., the order elements must be such that the order of each element in *f* must be the same as the order of the elements in the set *tff*. This requires diligence when declaring and assigning the elements in sets that will be mapped.

Maps for Production System Nesting

This requires assigning elements to two mapping sets that determine the arguments (factors) that are used in the second level of the production system (*QVAPRODFN*), *map_va_ff(ff,a)*, and the arguments used in the third level of the production system (*FDPRODFN*), *map_fag_ff(ff,ff,a)*.

This method of setting up a nested production system requires the declaration of a set *ff* that consists of all 'natural' factors, *f*, and all 'aggregate' factors, *fag*, and ensuring that the 'aggregate' factors are included in the set *sac*, i.e., declaring *ff(sac)* and *f(ff)*. This is useful because only natural factors (*f*) receive incomes that are parts of factor incomes that are passed down to institutions and therefore all other equations based on natural factors do not need to be changed.

Note that the calibration requires the imposition of exogenous data for the elasticities.

The identification of inputs used in the second level is achieved by setting a 'mapping' set that identifies those members the set *ff* that are used by each activity, *a*: *map_va_ff(ff,a)*. This mapping set can contain members of *ff* that are either natural (*f*) or aggregate (*fagg*) factors. The application of this mapping set can be seen in the calibration code below.

```
rhocva(a)    = (1/ELASTX(a,"sigmava")) - 1 ;

deltava(ff,a)$[map_va_ff(ff,a) AND FD0(ff,a)]
  = (WFDIST0(ff,a)*WF0(ff)*(FD0(ff,a))**(1+rhocva(a)))
     /SUM(ffp$map_va_ff(ffp,a),WFDIST0(ffp,a)*WF0(ffp)*(FD0(ffp,a))
             **(1+rhocva(a))) ;

ADVA0(a)$SUM(ff$map_va_ff(ff,a),deltava(ff,a)*FD0(ff,a))
     = QVA0(a)/(SUM(ff$map_va_ff(ff,a),deltava(ff,a)*FD0(ff,a)
                    **(-rhocva(a))))**(-1/rhocva(a));
```

When calibrating *deltava* the 'mapping' set defines the elements of *ff* in the numerator by virtue of the $ control on the LHS, while the use of the 'mapping' set in the denominator defines the elements of *ff* that are aggregated. The same logic applies for the shift parameter where the 'mapping' set in the denominator defines the elements of *ff* that are aggregated while the use on the $ control on the LHS simply avoids division by zero. Note this 'mapping' set allows for different elements of *ff* used by each activity.

Checking on the values for *deltava* is critical. They should sum to one for all activities and none can be negative: a 'negative input is an output'. Code is included in the files to carry out these checks and cause the programme to abort with error message if the checks fail. Such problems should only be caused by data[13] and mapping set issues.

The third level is concerned with the production of aggregate inputs (*fag*) that will be used, with or without natural factors (*f*), at the second level. Hence the need for a second 'mapping' set that identifies natural factors (*f*) that are used to produce the aggregate factors (*fag*): *map_fagg_ff(ff,ff)*. As before, when calibrating the share parameters, *deltafd*, the 'mapping' set defines the elements of *f* in the numerator, i.e., only natural factors, by virtue of the $ control on the LHS, while the use of the 'mapping' set in the denominator defines the elements of *f* that are aggregated to form the aggregate factors (*fag*). The same logic applies for the shift parameter where the 'mapping' set in the denominator defines the elements of *f* that are aggregated while the use on the $ control on the LHS simply avoids division by zero. Note this 'mapping' set allows for different elements of *f* used by each activity to produce each aggregate.

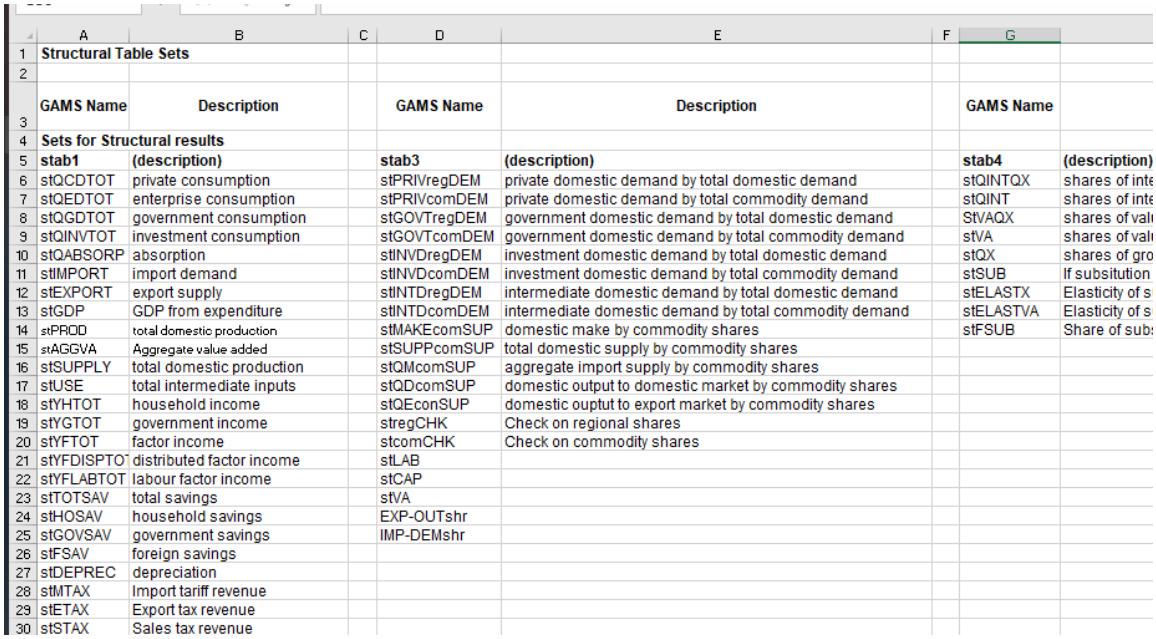
NOTE: the setup used for calibrating *stg_t.gms* only allows the production of aggregates (*fag*) at the third level from natural factors (*f*), while at the second level aggregates and natural factors can be used. With appropriate set assignments the third level equations can be extended for an *n*-level nest.

4.4 Structural tables (Stab) Sets

The stab_sets worksheet contains the sets used in the reports about the structure of the system being evaluated. These tables are generated by the model to provide information about the structure of the economy prior to the experiments. They are useful not only by providing data used to comment on economic structures but also when analysing the results from experiments. The sets in this worksheet will only change if the user chooses to extend the files that generate the structural tables.

[13] In SAMs with factor use taxes negative delta values can arise if the factor use tax exceeds the payment to the factor, i.e., the more the activity uses of the factor the less its costs of production with no limit. Such events are data errors.

Figure 4.4.1 Stab_Sets Worksheet



	A	B	C	D	E	F	G	
1	**Structural Table Sets**							
2								
3	**GAMS Name**	**Description**		**GAMS Name**	**Description**		**GAMS Name**	
4	**Sets for Structural results**							
5	**stab1**	**(description)**		**stab3**	**(description)**		**stab4**	**(description)**
6	stQCDTOT	private consumption		stPRIVregDEM	private domestic demand by total domestic demand		stQINTQX	shares of inte
7	stQEDTOT	enterprise consumption		stPRIVcomDEM	private domestic demand by total commodity demand		stQINT	shares of inte
8	stQGDTOT	government consumption		stGOVTregDEM	government domestic demand by total domestic demand		StVAQX	shares of val
9	stQINVTOT	investment consumption		stGOVTcomDEM	government domestic demand by total commodity demand		stVA	shares of val
10	stQABSORP	absorption		stINVDregDEM	investment domestic demand by total domestic demand		stQX	shares of gro
11	stIMPORT	import demand		stINVDcomDEM	investment domestic demand by total commodity demand		stSUB	If subsitution
12	stEXPORT	export supply		stINTDregDEM	intermediate domestic demand by total domestic demand		stELASTX	Elasticity of s
13	stGDP	GDP from expenditure		stINTDcomDEM	intermediate domestic demand by total commodity demand		stELASTVA	Elasticity of s
14	stPROD	total domestic production		stMAKEcomSUP	domestic make by commodity shares		stFSUB	Share of subs
15	stAGGVA	Aggregate value added		stSUPPcomSUP	total domestic supply by commodity shares			
16	stSUPPLY	total domestic production		stQMcomSUP	aggregate import supply by commodity shares			
17	stUSE	total intermediate inputs		stQDcomSUP	domestic output to domestic market by commodity shares			
18	stYHTOT	household income		stQEconSUP	domestic ouptut to export market by commodity shares			
19	stYGTOT	government income		stregCHK	Check on regional shares			
20	stYFTOT	factor income		stcomCHK	Check on commodity shares			
21	stYFDISPTOT	distributed factor income		stLAB				
22	stYFLABTOT	labour factor income		stCAP				
23	stTOTSAV	total savings		stVA				
24	stHOSAV	household savings		EXP-OUTshr				
25	stGOVSAV	government savings		IMP-DEMshr				
26	stFSAV	foreign savings						
27	stDEPREC	depreciation						
28	stMTAX	Import tariff revenue						
29	stETAX	Export tax revenue						
30	stSTAX	Sales tax revenue						

4.4.1 Results (res) Sets

This worksheet of sets is something of an anomaly; while it is included in the data workbook it apparently refers to actions that take place in the experiment file. The reason is simple. This worksheet contains sets that are used in the generation of the descriptive statistics module (`stg_t_struct.inc`) that provides structural information about the underlying database, and this module is implemented during the model set up and calibration phase. This allows the analyst to evaluate the data, and choice of aggregation, before conducting experiments.

The sets in this worksheet will only change if the analyst chooses to extend the files that generate aggregated results. This is a straightforward process if the objective is to simply add another summary statistic to the already define parameters: the additional set member is added to the already defined set and the calculations required to compute the new summary statistics are added to the appropriate results analyses' files and/or the descriptive structural file.

Figure 4.4.2 Results Set Worksheet

1	**Experiment Results Sets**				
2					
3	**GAMS Name**	**Description**		**GAMS Name**	**Description**
4	**Set for adjuster results**			**Set for scalar results**	
5	**adjres**	**(description)**		**scalres**	**(description)**
6	aTEADJ	Export subsidy Scaling Factor		sER	Exchange rate (domestic per world unit)
7	aTMADJ	Tarrif rate Scaling Factor		sCPI	Consumer price index
8	aTSADJ	Sales tax rate scaling factor		sPPI	Producer (domestic) price index
9	aTSSADJ	Sales tax 2 rate scaling factor		sMTAX	Tariff revenue
10	aTEXADJ	Excise tax rate scaling factor		sETAX	Export tax revenue
11	aTVADJ	Value added tax rate scaling factor		sDTAX	Direct Income tax revenue
12	aTXADJ	Indirect Tax Scaling Factor		sFYTAX	Factor Income tax revenue
13	aTFADJ	Factor Use Tax Scaling Factor		sITAX	Indirect tax revenue
14	aTYFADJ	Factor Tax Scaling Factor		sSTAX	Sales tax revenue
15	aTYHADJ	Household Income Tax Scaling Factor		sSSTAX	Sales tax 2 revenue
16	aTYADJ	Household and Enterprise Income Tax Scaling Factor		sVTAX	Value added tax
17	aTYEADJ	Enterprise income tax Scaling Factor		sFTAX	Factor use tax revenue
18	aDTE	Partial Export tax rate scaling factor		sEXTAX	Excise tax revenue
19	aDTM	Partial Tariff rate scaling factor		sINVEST	Total investment expenditure
20	aDTS	Partial Sales tax rate scaling factor		sTOTSAV	Total savings
21	aDTSS	Partial Sales tax 2 rate scaling factor		sYG	Government income
22	aDTEX	Partial Value added tax rate scaling factor		sEG	Expenditure by government
23	aDTV	Partial Excise tax rate scaling factor		sVGD	Value of Government consumption expendi

4.5 Elasticities

In addition to the transactions data derived from the (aggregate) SAM the model also needs a series of elasticities. The user needs to assign the elasticities used by the model.

Commodity Elasticities

Substitution/transformation elasticities are needed for commodities (see Figure 4.5.1). These are recorded in the worksheet, 'comelast', where the column *sigma* records the elasticities of substitution between **im**ported and domestically produced commodities, the column *omega* records the elasticities of transformation between **ex**ported and domestically produced commodities. the column *exdem* records the response elasticities for changes in export volumes that impact on world prices (a large country assumption) and the column sigmaxc records the elasticities of substitution between imported and domestically produced commodities s that of substitution between domestically produced commodities produced by different activities. All are components of the model parameter *ELASTC*.

The determination of appropriate elasticities is the responsibility of the user; the numbers in the model template are placeholders **only**.

Figure 4.5.1 Commodity Elasticities

	A	B	C	D	E	F
1	**Model Specific Commodity Elasticities**					
2	Elasticities					
3						
4	ELASTC	ELASTC	ELASTC	ELASTC	ELASTC	ELASTC
5			sigma	omega	exdem	sigmaxc
6	Agriculture	cagric	1.75	2	0	4
7	Minerals	cmins	1.5	1.5	0	4
8	Food products	cfood	1.5	1.75	0	4
9	Textiles	ctext	1.75	2	0	4
10	Petroleum and chemical products	cpetchem	1.25	2	0	4
11	Metal and mineral products	cmprod	0.9	1.1	0	4
12	Machinery	cmach	0.75	1.2	0	4
13	Vehicles	cveh	0.8	1.25	0	4
14	Other manufacturing	comanu	1.1	1.1	0	4
15	Utilities	cutil	2	2	0	4
16	Construction	ccns	0.75	0.75	0	4
17	Trade transport and communications	ctrad	1.5	1.5	0	4
18	Services	cserv	1.5	1.5	0	4
19						

Activity Elasticities

Substitution/transformation elasticities are required for the activities, i.e., production system (see in Figure 4.5.2). These are recorded in one worksheet, ‘actelast’, where the columns *sigmax* and *sigmava* are the elasticities of substitution between aggregate intermediate inputs and aggregate value added and the elasticities of substitution between primary inputs (model parameter *ELASTVA*). The column *omegaout* has the elasticities of transformation that allow activities to adjust the composition of outputs to price changes. All are components of the model parameter *ELASTX*.

In addition, the nested production system requires identifying elasticities of substitution between labour types that are activity specific. These are recorded in the worksheet ‘elastf’ (see below Figure 3.2.2.2). They could have been included in the worksheet ‘actelast’ but this becomes more complicated with systems of complex production nests, e.g., in energy, water, climate, etc., models, so retaining a separate worksheet has some advantages.

The determination of appropriate elasticities is the responsibility of the user; the numbers in the model template are placeholders **only**.

Figure 4.5.2 Activity Elasticities

	A	B	C	D	E
1	**Model Specific Activity Elasticities**				
2	Elasticities				
3					
4	ELASTX	ELASTX	ELASTX		
5			sigmax	sigmava	omegaout
6	Agriculture	aagric	0.9	1.5	0
7	Minerals	amins	0.5	0.8	0
8	Food products	afood	0.9	1.6	0.25
9	Textiles	atext	0.9	1.5	0.25
10	Petroleum and chemical products	apetchem	0.5	0.8	0.25
11	Metal and mineral products	amprod	0.5	0.8	0.25
12	Machinery	amach	0.65	1.25	0.25
13	Vehicles	aveh	0.5	0.8	0.25
14	Other manufacturing	aomanu	0.5	0.8	0.25
15	Utilities	autil	0.5	0.8	0
16	Construction	acns	0.9	1.5	0
17	Trade transport and communications	atrad	0.65	1.2	0
18	Services	aserv	0.9	1.6	0

Figure 4.5.3 Third Level Production Function Elasticities

	A	B	C	D	E	F	G	H	I
1	**Model Specific Factor Elasticities by Activity**								
2	Elasticities								
3									
4	ELASTF	ELASTF	ELASTF						
5			aagric	amins	afood	atext	apetchem	amprod	amach
6	Aggregate Labour	flab	4	4	4	4	4	4	4
7									

Demand System Elasticities

The elasticities used for the linear expenditure systems (LES) are recorded in two worksheets, ‘hoelast’ and ‘frischelast’ (see Figures 4.5.4 and 4.5.5). The income elasticities of demand are recorded in the worksheet ‘hocelast’, while the income elasticity of money elasticities are recorded in the worksheet ‘frischelast’.

The default values for ‘hoelast’ are all one and for ‘frischelast’ are all MINUS one. This is a special case where the LES utility functions reduce to Cobb-Douglas utility functions.

The determination of appropriate elasticities is the responsibility of the user; the numbers in the model template are placeholders **only**.

Figure 4.5.4 Income Elasticities

	A	B	C	D	E	F	G	H	I	J
1	**Model Specific Household Elasticities**									
2	Elasticities									
3										
4	ELASTY	ELASTY	ELASTY	ELASTY	ELASTY					
5			hafflow	haffmed	haffhigh	hafmlow	hafmmed	hafmhigh	hcalow	hcahigh
6	Agriculture	cagric	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0
7	Minerals	cmins	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0
8	Food products	cfood	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0
9	Textiles	ctext	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1
10	Petroleum and chemical products	cpetchem	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1
11	Metal and mineral products	cmprod	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1
12	Machinery	cmach	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1
13	Vehicles	cveh	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1
14	Other manufacturing	comanu	2	2	2	2	2	2	2	
15	Utilities	cutil	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1
16	Construction	ccns	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0
17	Trade transport and communications	ctrad	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1
18	Services	cserv	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2
19										

Figure 4.5.5 Frisch Elasticities

	A	B	C
1	**PROVIDE: SA SAM 2000**		
2	Elasticities		
3			
4	ELASTMU	ELASTMU	ELASTMU
5			frisc
6	African female low education	hafflow	-4.59
7	African female mid education	haffmed	-4.30
8	African femal high education	haffhigh	-2.76
9	African male low education	hafmlow	-4.07
10	African male mid education	hafmmed	-3.43
11	African male high education	hafmhigh	-2.32
12	Coloured and Asian low education	hcalow	-3.48
13	Coloured and Asian high education	hcahigh	-1.88
14	White low education	hwhlow	-1.96
15	White high education	hwhhigh	-1.05

5. Controls Worksheet

There are numerous aspects of the model structure and the flow of the programme that the user might wish to control. The approach used in the STAGE model is to concentrate as many of these aspects of the programme as practical in an Excel worksheet – the 'controls' worksheet – in a table `mod_cont(mcons)` that contains values for parameters controlling model content.

The `mod_cont` parameters are contained in a data table that consists of various elements – Figure 5.1. Also included are examples of typical default values and brief descriptions of the role of each parameter in the configuration of the model.

Figure 5.1 Model Controls

	A	B	C	D	E	F	G
1	**Control Parameters**						
2	**These values are used to initialise various parameters that condition the model and control the operation of the model programme**						
3							
4							
5	**mod_control**						
6							
7	**mcons**						
8	numerchk	1			IF 1 then default, if NOT 1 then check on numeraire		
9	minaqxsh	0.1			Minimum share of intermediates in cost for aqx		
10	samscal	0.001			SAM scaling factor and initial value for autoscaling		
11	scaltarg	100			target level for auto scaling		
12	scalprop	0.95			proportion of NON zero elements that must be below the target level		
13	setpop	2			If 1 then POP data used; If 2 then adult equivalent data used		
14	toldiffsam	0.000001			Tolerance on differences in base and solution SAM entries		
15	micsam1	1			If 1 checks micro SAM at calibration; If 0 does not check micro SAM at calibration		
16	micsam2	1			If 1 checks micro SAM at solution; If 0 does not check micro SAM at solution		
17							

6. Structure of the STAGE Model Code

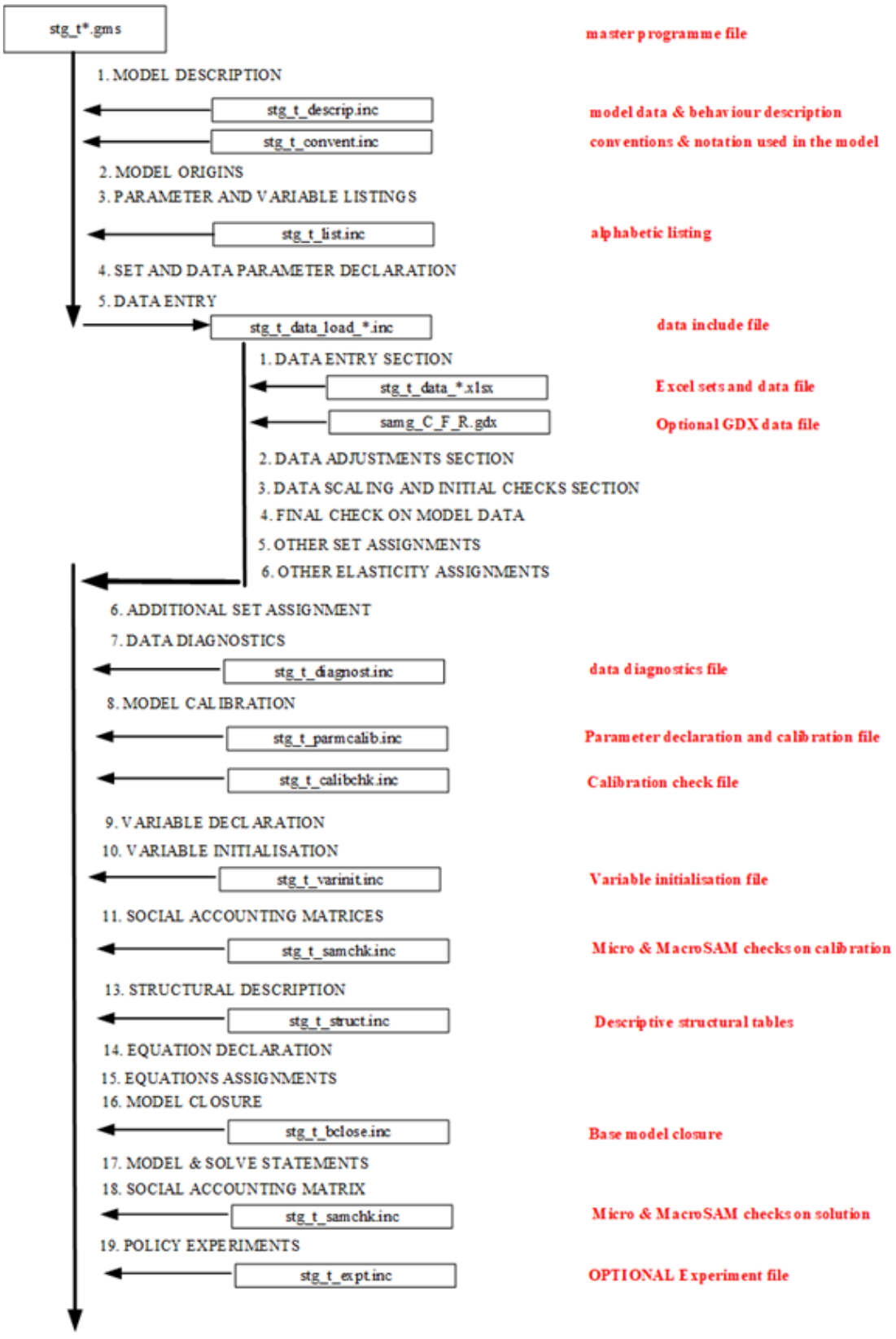
The STAGE programme code is relatively complex and consequently a modular structure is adopted; this takes the form of a series of *INCLUDE* files that are called from the main programme – `***.gms` – file and two MS Excel workbooks that contain model and experiment/simulation data. In addition, the model makes uses of number of *IF* statements that are used to control which modules are implemented in any run of the model. Finally, the model set up assumes users will implement the model using GAMS's *SAVE* and *RESTART* facility – this facility and its uses are not discussed in this User Guide.

A key element of the programming philosophy is that as far as possible all the data entry required by a typical user should be concentrated. For the base programme all the data entry is concentrated in two places – one *INCLUDE* file and one Excel workbook.[14] In the experiment programme file all the data entry AND programming takes place in three places – one *INCLUDE* file for each set of simulations, multiple *INCLUDE* files for closure choices and one Excel workbook.

The schematic illustration of the structure of the STAGE model file – stg_*.gms – in Figure 6.1 is a reproduction of a schematic of the file structure included in the programme file. The place in the core programme where each *INCLUDE* file is called is shown together with the titles to each of the sections of the core programme file. In addition, the descriptions down the right-hand side indicate the role each *INCLUDE* file plays in the programme. The majority of the *INCLUDE* files only need attention if the user is making changes to the core model code. However, there are two 'areas' to which the user needs to pay attention:

[14] Optionally some additional data can be input from a GDX file if this is more convenient.

Figure 6.1 STAGE File Structure



- `stg1_data_load_*.inc` – data entry file
- `stg1_data_*.inc` – Excel data file

these are explored below, using extracts from the model code files.

The model follows a standard format for the presentation of a GAMS programme. All sets and parameters are declared, using the $ONEMPTY option, followed by section wherein the data are loaded, i.e., the sets and parameters are populated. After loading the data various adjustments and checks[15] are conducted to ensure the data are consistent AND that the data do not encompass transactions for which behavioural relationships are not included in the model (`stg1_diagnost.inc`); these checks will cause the model to abort and then print a message in the `*.lst` file that identifies the check that has been failed.[16]

After the data has been conditioned and checked all the model's parameters are declared and assigned (`stg_t_parmcalib.inc`) followed by the declaration of all model variables[17] and then the variables are initialised (`stg_t_varinit1.inc`). This initialisation of the variables includes an (implicit) double check on the parameter calibration of the initial values for (nearly) all variables. The number of display statements in the programme is limited; rather it is assumed the user will run the model with GDX creation (F10 in Studio) in which case all the parameters and (initial values for) variables are reported, automatically, in a GDX file (`stg_t.gdxinc`).

This is followed by two further `*.inc` files. The first (`stg_t_samchk.inc`) checks that the calibrated parameter and variable values generates 'macro' and 'micro' SAMs that are consistent with the data (after any adjustments) that were loaded and then again after the model is first solved. The second (`stg_t_struct.inc`) computes are series of reports that describe the structure of the economy contained within the base data; these reports provide

15 Domain checking on the sets and data are conducted when the data are being loaded through options included in GDXXRW.

16 Before the abort statements are implemented a full listing of checks that have been failed are printed to the `*.lst` file; this is because the first occurrence of an error that triggers an abort command stops the model.

17 Note that there are more variables than equations; this will be resolved when the case specific model (macroeconomic) closure rules and market clearing conditions are specified.

data for use when analysing the results from simulation exercises and input to written project reports.

7. Model Data and Conditioning File

The data entry file – smod1_data_load_*.inc - is extensively documented at the top of the file; the notes here are designed to assist the user rather than replace the information provided there.

Data Entry Section

In this section the user makes the necessary entries to change the data sets provided to the model. There are 6 sub sections.

1. Excel workbook sets and data converted to gdx here

   The user assigns the name for the input (i) Excel file using $SETGLOBAL in stg_t.gms and this is converted to data_in.gdx using the information in the index sheet Layout starting from the cell A4, i.e.,

```
$CALL "GDXXRW i=1_1_Model_data/%mod_data%.xlsx o=1_1_Model_data/data_in.gdx
                                    1. index=Layout!A4 trace=3 "
```

   The trace option is added so that the log file reports details about the data loaded. The default is zero and there four higher value each producing more information.

2. GAMS can also check for errors and abort the programme if certain errors are detected

```
$if errorLevel 1 $abort problems with GDXXRW
```

3. All data, elasticities and sets from Excel are assigned here

   There should be no need to make changes here unless the model is being changed.

4. Defining SETS by exclusions from previously defined SETS

   There should be no need to make changes here unless the model is being changed

5. Defining MAPPING SETS by using subset information

   There should be no need to make changes here unless the model is being changed.

6. Initial SAM check

The user needs to enter the appropriate Excel file name in to link the GAMS programme to the Excel database (see above for details about compiling model's Excel file).

The rest of this section typically only needs explicit attention if the user is making changes to the model's behavioural relationships and/or structure.

Data adjustments section

This section may need the user to make some changes to accommodate specific aspects of the database used for a particular model. Typically, many of these changes can be made in the Excel data file but this section provides a facility for making them within the model code. There are however a series of standard 'adjustments' included in the default setting that are designed to remove common 'errors'. These should be left unchanged since they are neutral with respect to the information content of the SAM but avoid some common problems.

Data scaling and factor quantities section

This section scales the transactions data and loads and scales the factor use data; if there is no satellite accounts for factor quantities the model by default uses the factor transactions data. The scaling is to improve algorithm performance. The data scaling routines use `mod_cont("scalprop")` and `mod_cont("scaltarg")` that are set in the Excel workbook to control automatic scaling. The default values of `mod_cont("scalprop")` and `mod_cont("scaltarg")` are 0.95 and 100.

There is usually no need for the user to make changes in this section, but the parameters may need tweaking especially if the SAM is badly scaled.

Final check on model data

This section provides a simple check that ensure the programme only continues if the SAM is balanced. If it is not balanced the programme aborts with the error message:

```
"Totals Check failed - Check SAM after adjustments"
```

Other set assignments

This section assigns a number of set memberships. These (sub) sets typically control aspects of the model including the choice of behavioural relationships that are implemented. These include:

1. 5a. Assign set members for unskilled labour by exclusion
2. 5b. Defining sets to control production nesting structure

This uses the data and the sets aleon and rleon from the worksheet 'mod_sets' to set aqx and aqxn. A manual option is also available. aqxn is the complement to aqx.

3. 5c. Defining sets to control aggregation of commodities
   Linear aggregation of homogenous commodities

Other elasticity assignments

This section allows the user to override the elasticity data provided in the Excel workbook; this section needs using with care to avoid introducing parameter changes that can be overlooked.

8. Model Calibration Checks

The STAGE model has several aspects that facilitate checking that the model is correctly specified. Whenever the user makes any changes to the model or the model data these checks should be conducted BEFORE carrying out any simulations; failure to do so may mean that the simulations are conducted using an incorrectly specified model.

1. Slack variables: the slack variable should equal zero, or very nearly zero. Search for '`var walras` which should be zero.
2. Check the Left hand sides: Search for 'LHS', then after finding the first occurrence of 'LHS' search for '***'. If any equations are incorrectly specified, they are identified. (Note: if the version of GAMS used has indexing for the list file select SolEQU and then the first named equation, this will move the cursor to the first equation.)
3. Check data replication using the ***.gdx file produced if run with GDX creation (F10): First check the Macro SAM: search for 'ASAMG2CHK' – all the values should equal 1; then search for and check DIFFASAMG2 and CNTASAMG2 – these should be zeros or very close to zero. Second check the Micro SAM: search for and check DIFFSAMG2 and CNTSAMG2 – these should be zeros or very close to zero.
4. Check the numéraire: The Excel workbook go to the worksheet 'mcontrols' and change the value of 'numerchk' to 2, save the Excel file and rerun the model. Then check the Macro SAM: search for 'ASAMG2CHK' – all the values should equal 2; note that DIFFASAMG2 and CNTASAMG2 are no longer meaningful and therefore the micro SAM calculations have not been implemented.

If the model passes all these checks the model will (usually) be correct.

9. Experiment File Template

The experiment file is where the user has to be most active. The model file is largely straightforward and any changes the user would make are standardised, but the range of potential changes in the experiment file are very large and defy standardisation.

9.1 Experiment File Structure

The structure of the experiment file is illustrated in Figure 9.1.1.

The experiment file is set up so that a set of simulations are run in a LOOP that is nested within 2 other the LOOPS; the first allows the user to change the closure conditions while the other allows the user to change the elasticities. All the LOOPS have controlling sets that can be multi or single member sets – *sim* for simulations, *clos* for closure and *elst* for elasticities. When changing closure conditions, it is necessary to reset the choice of 'fixed' variables; this is done for ALL variables that can enter into any closure condition by the file `glb1_reset.inc`.

With the completion of each simulation the levels values for the results are assigned to the appropriate results parameter – all results parameters use the same name as the associated variable prefixed with 'res'. These levels results are subsequently used to derived additional (analytical) results (see below).

9.2. Analysis File

The analysis file uses the values for the variables after each solution plus the model's parameters to develop a series of derived results that assist in analyses of the model simulations. The various components of the analysis file are illustrated in Figure 9.2.1 and the user chooses which files to run by setting the 0/1 parameters in the worksheet 'econtrols'. Each component exports the results it generates to a GDX file with a default name linked to the name of the respective *INCLUDE* file; these can be changed but it is often easier to collect the GDX files containing the results together after a run (NB: if the GDX files are not collected and moved to another location/directory they will be over written).

The user should expect to extend the analyses files to meet the needs of the analyses they are engaged in conducting.

Figure 9.1.1 STAGE Experiment File Structure

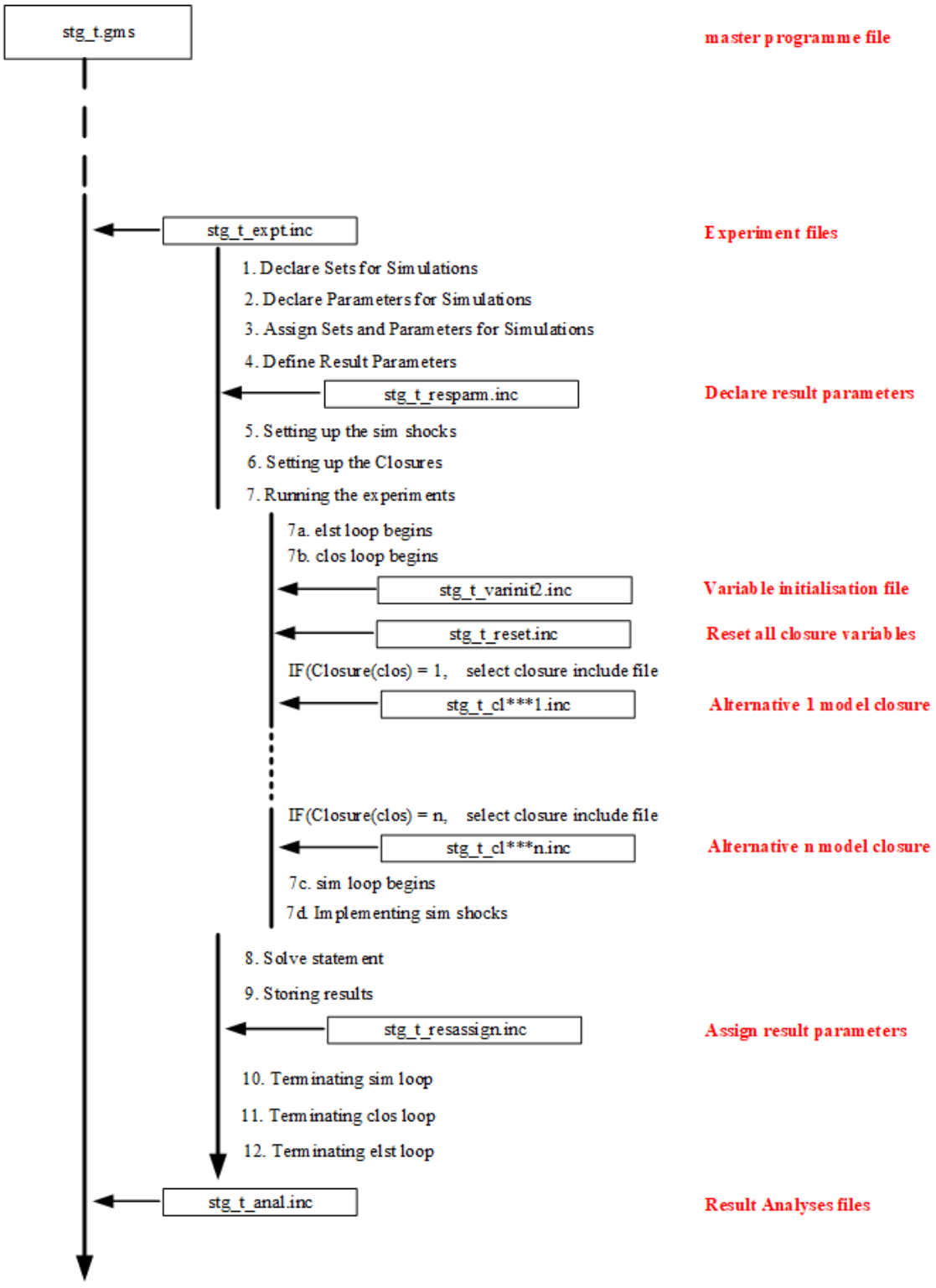
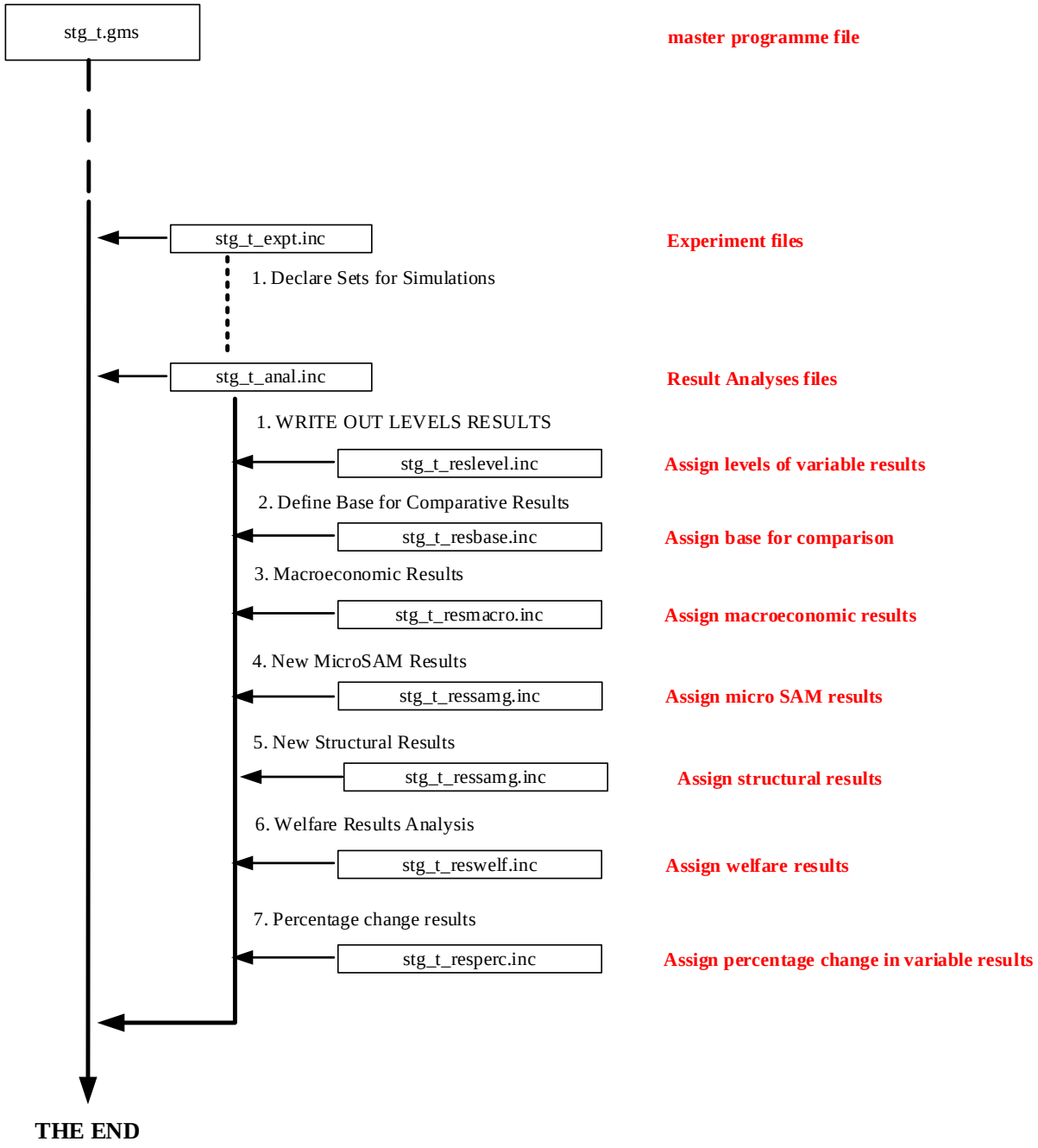


Figure 9.2.1 STAGE Results Analysis File Structure



10. Compiling an Experiment Excel Workbook

Despite the proliferation of files that arises from the use of a separate Excel workbook to contain sets and data used by each experiment there are substantial advantages. First, each worksheet with sets and data for an experiment can be related to a single experiment file, thereby assisting with replicability. Second, it facilitates programming of the experiment files by increasing the amount of generic code that can be used. And third, this approach allows the user to exploit the `save and restart` facility in GAMS while retaining the advantages of passing set and data information to the experiment files from Excel; this can provide appreciable savings in time when building up a series of simulations.

The arrangement of the Excel workbooks for experiments is essentially the same as for the model data; a series of worksheets containing structured information together with a 'layout' worksheet that provides an 'index'. Note that the model is not designed so that changes in the behavioural structure can be affected during the experiment stage, except for changes to the series of closure rules.

Simulation Sets

The worksheet for the simulation sets, 'exp_sets', contains only generic simulation sets; these are sets that are declared as standard in the model code. These sets are

- sim simulations set
- simc applied Simulations set
- elst elasticities set
- clos closures set

Changing the membership of these sets allows for the running of different combinations of shocks, through 'simc', elasticities, through 'elst', and closures, through 'clos' without having to entry the specific choices directly in the experiment file.

In addition to the *sim* sets closure (*clos*) and elasticity (*elst*) sets are assigned to control the other LOOPS in the experiment file – see Figure 10.1.

Figure 10.1 Simulations Sets Worksheet Part 1

	A	B	C	D	E	F
1	**Model Experiment Sets**					
2						
3	**GAMS Name**	**Description**		**GAMS Name**	**Description**	
4		The full set of sims defined			Set of sims run	
5	**sim**			**simc**		
6	base	baseline for comparisons		base	baseline for comparisons	
7	sim01	50% cut in trade barriers by RSA		sim01	50% cut in trade barriers by RSA	
8	sim02	25% cut in trade barriers by RSA		sim02	25% cut in trade barriers by RSA	
9	sim03	50% cut in trade barriers by RSA		sim03	50% cut in trade barriers by RSA	
10	sim04	75% cut in trade barriers by RSA		sim04	75% cut in trade barriers by RSA	
11	sim05	100% cut in trade barriers by RSA		sim05	100% cut in trade barriers by RSA	
12	sim06	50% cut in Agric trade barriers by RSA		sim06	50% cut in Agric trade barriers by RSA	
13	sim07	100% cut in Agric trade barriers by RSA		sim07	100% cut in Agric trade barriers by RSA	
14	sim08	50% cut in Minearls trade barriers by RSA		sim08	50% cut in Minearls trade barriers by RSA	
15	sim09	100% cut in Minerals trade barriers by RSA		sim09	100% cut in Minerals trade barriers by RSA	
16	sim10	50% cut in Manuf trade barriers by RSA		sim10	50% cut in Manuf trade barriers by RSA	
17	sim11	100% cut in Manuf trade barriers by RSA		sim11	100% cut in Manuf trade barriers by RSA	
18	sim12	50% cut in Serv trade barriers by RSA		sim12	50% cut in Serv trade barriers by RSA	
19	sim13	100% cut in Ser trade barriers by RSA		sim13	100% cut in Ser trade barriers by RSA	

Figure 10.2 Simulations Sets Worksheet Part 2

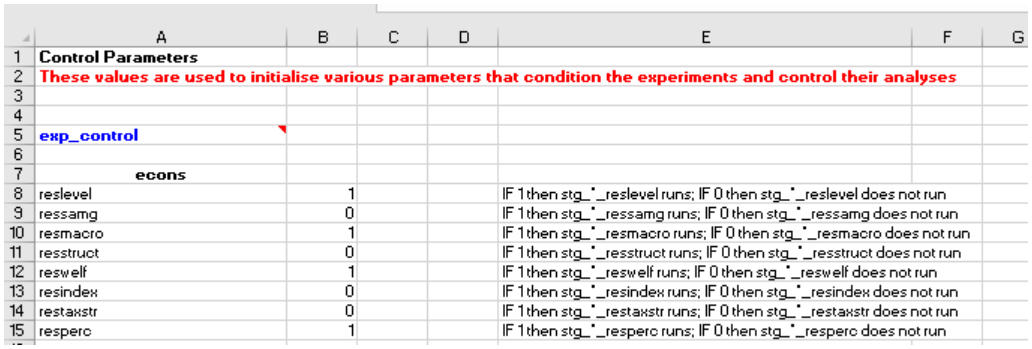
	F	G	H	I	J	K
1						
2						
3		**GAMS Name**	**Description**		**GAMS Name**	**Description**
4			Set elements for closure selection			Set elements for elaticity selection
5		**clos**			**elst**	
6		cl_base	Base liberal free market closure		elst_01	50% cut in elasto
7		cl_tyh	Basic tax replacement with TYH		elst_02	50% cut in elastx
8		cl_ts	Basic tax replacement with TS			
9		cl_tv	Basic tax replacement with TV			
10						

When defining experiments, it is often extremely useful to create (sub)sets to abbreviate the coding of the shocks; the sheet ‘exp_sets’ can be easily extended for this purpose. This approach requires the user to declare the set in the experiment file, assign the set in Excel and extend the layout/index sheet to include the new set and then load the set using GDXIN. Some users prefer to declare AND assign sets directly in the experiment file. Both methods achieve the same objective.

Flow Controls

The flow controls in the worksheet ‘exp_controls’ are included so that the user can select which components of the analyses file are implemented. These are simple 0 or 1 parameters used to trigger IF statements. Figure 10.3 provides a screen shot of the ‘exp_controls’ worksheet; as can be seen there are descriptions for each parameter.

Figure 10.3 Exp(eriment)_controls



	A	B	C	D	E	F	G
1	**Control Parameters**						
2	**These values are used to initialise various parameters that condition the experiments and control their analyses**						
3							
4							
5	**exp_control**						
6							
7	**econs**						
8	reslevel	1			IF 1 then stg_*_reslevel runs; IF 0 then stg_*_reslevel does not run		
9	ressamg	0			IF 1 then stg_*_ressamg runs; IF 0 then stg_*_ressamg does not run		
10	resmacro	1			IF 1 then stg_*_resmacro runs; IF 0 then stg_*_resmacro does not run		
11	resstruct	0			IF 1 then stg_*_resstruct runs; IF 0 then stg_*_resstruct does not run		
12	reswelf	1			IF 1 then stg_*_reswelf runs; IF 0 then stg_*_reswelf does not run		
13	resindex	0			IF 1 then stg_*_resindex runs; IF 0 then stg_*_resindex does not run		
14	restaxstr	0			IF 1 then stg_*_restaxstr runs; IF 0 then stg_*_restaxstr does not run		
15	respero	1			IF 1 then stg_*_respero runs; IF 0 then stg_*_respero does not run		

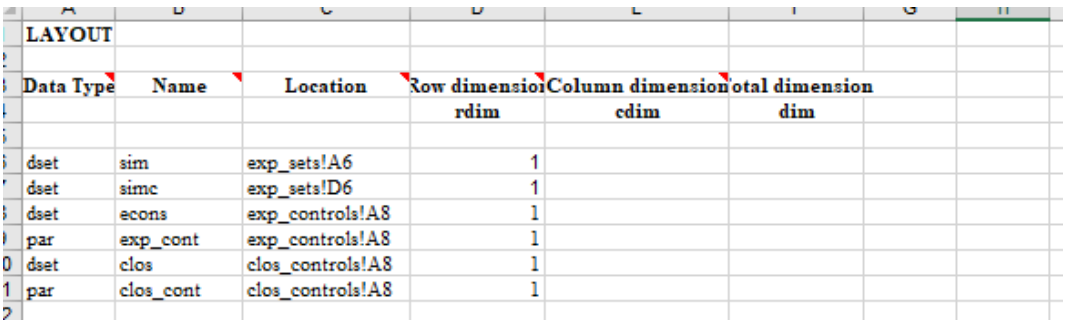
Data

For some experiments it is convenient to provide the values for the shocks from a table of data. If a user chooses this method a parameter must be declared in the experiment file, a new worksheet must be added in which to assign values, the layout/index sheet must be extended to include the new parameter and then the data loaded using GDXIN.

Layout Sheet

The syntax for the 'layout' worksheet is described in the gdx utilities documentation supplied with GAMS. When making changes it is important to ensure that all the syntax etc., is fully consistent; this is especially the case when working with GDXXRW since the error messages are not always as informative as the user might wish. If the user does get errors, then it is wise to review the associated *.log file since the detail therein is the most comprehensive available.

Figure 10.4 Experiment Layout/Index Sheet



	A	B	C	D	E	F	G	H
	LAYOUT							
	Data Type	**Name**	**Location**	**Row dimensio rdim**	**Column dimension cdim**	**otal dimension dim**		
	dset	sim	exp_sets!A6	1				
	dset	simc	exp_sets!D6	1				
	dset	econs	exp_controls!A8	1				
	par	exp_cont	exp_controls!A8	1				
0	dset	clos	clos_controls!A8	1				
1	par	clos_cont	clos_controls!A8	1				
2								

11. Market Clearing and Model Closure Rules

The model is programmed to provide a wide degree of flexibility for the user in the selection of market clearing and model (macroeconomic) closure rules. The supplied version of the model adopts the principle that the model will be calibrated using a default version of these rules, and thereafter the user selects one or more sets of the rules that are implemented in a system of LOOPs that are embedded in the experiment file. This allows the user to conduct the same set of simulations using a range of different market clearing and model closure rules; this can be viewed as fulfilling one or more of the following objectives:

- identifying the contributions of different components of the model to the overall results;
- exploring properties of the model;
- conducting sensitivity analyses with respect to the (exogenously) imposed assumptions about economic systems; and
- allowing for uncertainty about the nature of the market clearing or macroeconomic mechanisms.

For any given model there are a large number of different permutations for these ‘rules’. It is therefore important to be systematic in identifying the rules that are to apply for each model and to include checks in the model simulations to ensure that the chosen ‘rules’ and those actually applied.

The extracts of code reported below are taken from the template file for closure conditions; greater detail is provided in the technical document for the STAGE model. At the top of the file a simple ‘table’ provides a basis for summarising the chosen closure conditions, i.e.,

```
$ontext
stg_t_cl_base - minimal neo classical closure for model initialisation
                     CONDITION

     FEX         - Exchange rate                      - FLEX
                 - CAPWOR                             - FIX

     Investment  - absorption share                   - FIX

     Government  - absorption share                   - FIX
                 - Tax rates                          - ALL FIX
```

```
                - KAPGOV                                - FLEX

        Factors - Land sector                           - mobile & full
                - Capital                               - mobile & full
                - Unskilled labour                      - mobile & full
                - Skilled labour                        - mobile & full

        Numeraire - CPI                                 - FIX
                - PPI                                   - FLEX

   It is assumed that the closures used for experiments will be set in
   the experiment files.
   $offtext
```

Repeating this where the closure *INCLUDE* file enters the experiment file provides one why of documenting the model and experiment.

11.1 Macroeconomic Closures

Foreign Exchange Closure

This, essentially, determines whether the foreign exchange market clears by variations in the exchange rate or the current account balance. There are situations where both need to be fixed, which means consideration needs to be given to the closure settings that allow to foreign exchange market to clear, e.g., a reduction in domestic absorption.

```
*## FOREIGN EXCHANGE MARKET CLOSURE
$ontext

* ER.FX            = ER0 ;

* alternatively the external balance is fixed

 CAPWOR.FX       = CAPWOR0 ;

* Fixing of world prices
 PWM.FX(c)         = PWM0(c) ;
 PWE.FX(cedn)      = PWE0(cedn) ;
```

Investment-Savings Closure

This determines whether the investment-savings market clears by variations in savings or investment.

```
*## INVESTMENT-SAVINGS CLOSURE
$ontext
 IF Aggregate investment is determined by aggregate savings,
 i.e., the model is savings driven, then fix ALL of SADJ, SHADJ, SEADJ,
 DSHH, DS and DSEN.
```

```
 IF the model is investment driven only one of SADJ, SHADJ, SEADJ, DSHH,
 DS or DSEN should be variable.

 MULTIPLICATIVE adjustment
 SADJ variable - h'hold and enterprise savings rates vary equiproportionately
 SHADJ variable - h'hold savings rates vary equiproportionately
 SEADJ variable - enterprise savings rates vary equiproportionately

 ADDITIVE adjustment
 DS variable - selected h'hold and enterprise savings rates vary
 DSHH variable - selected h'hold savings rates vary
 DSEN variable - selected enterprise savings rates vary

 For investment driven closures the 'level' of investment can be defined
 in terms of the volume, value or share of final demand expenditure.
$offtext

* Savings Rate adjustments

* SADJ.FX          = SADJ0 ;
 SHADJ.FX         = SHADJ0 ;
 SEADJ.FX         = SEADJ0 ;

 DSHH.FX          = DSHH0 ;
 DS.FX            = DS0 ;
 DSEN.FX          = DSEN0 ;

* Investment 'level' adjustments

* IF the model is investment driven fix the investment scaling factor

* IADJ.FX         = IADJ0 ;

* OR the value or share of domestic final demand of investment is
fixed

* INVEST.FX = INVEST0 * mod_cont("numerchk") ;
 INVESTSH.FX     = INVESTSH0 ;
```

Enterprise Closure

This determines how the enterprise account clears. In this case the determinants included in this section relate to expenditure decisions with savings and tax consideration, which partly determine funds available for expenditure set in the investment-savings and government closure settings.

```
*## ENTERPRISE CLOSURE RULES
$ontext
 ONE of the volume, value or share of final demand should be fixed
$offtext

 QEDADJ.FX        = QEDADJ0 ;
* VED.FX(e)       = VED0(e) ;
* VEDSH.FX(e)      = VEDSH0(e) ;
```

```
* Enterprise transfers to households

 HEADJ.FX        = HEADJ0 ;
```

Government Closure Rules

This identifies the government's decisions about tax rates, tax revenues and government expenditures.

```
*## GOVT CLOSURE RULES
$ontext
 IF ALL tax rates are fixed AND Government consumption expenditures and
transfers are fixed  the equilibriating variable is Government Savings

 IF a tax rate is variable then an alternative equilibriating variable
 is required.

 Note that the tax rates can be varied multiplicatively or additively
$offtext

* Tax rate scaling factors

 TEADJ.FX         = TEADJ0 ;
 TMADJ.FX         = TMADJ0 ;
 TSADJ.FX         = TSADJ0 ;
 TSSADJ.FX        = TSSADJ0 ;
 TVADJ.FX         = TVADJ0 ;
 TEXADJ.FX        = TEXADJ0 * mod_cont("numerchk") ;
 TYFADJ.FX        = TYFADJ0 ;
 TXADJ.FX         = TXADJ0 ;
 TFADJ.FX         = TFADJ0 ;
 TYHADJ.FX        = TYHADJ0 ;
 TYADJ.FX         = TYADJ0 ;
 TYEADJ.FX        = TYEADJ0 ;

 DTE.FX           = DTE0 ;
 DTM.FX           = DTM0 ;
 DTS.FX           = DTS0 ;
 DTSS.FX          = DTSS0 ;
 DTV.FX           = DTV0 ;
 DTEX.FX          = DTEX0 ;
 DTX.FX           = DTX0 ;
 DTF.FX           = DTF0 ;
 DTYF.FX          = DTYF0 ;
 DTYH.FX          = DTYH0 ;
 DTY.FX           = DTY0 ;
 DTYE.FX          = DTYE0 ;

    Tax revenue options

* ETAX.FX         = ETAX0     ;
* MTAX.FX         = MTAX0     ;
* STAX.FX         = STAX0     ;
* SSTAX.FX        = SSTAX0    ;
```

```
* DTAX.FX         = DTAX0     ;
* EXTAX.FX        = EXTAX0    ;
* FYTAX.FX        = FYTAX0    ;
* ITAX.FX         = ITAX0     ;
* FTAX.FX         = FTAX0     ;
* VTAX.FX         = VTAX0     ;

* Government expenditure scaling factor
* EITHER the volume, value or share of final demand should be fixed

* QGDADJ.FX        = QGDADJ0 ;
* VGD.FX           = VGD0 ;
 VGDSH.FX         = VGDSH0 ;

* Government transfers scaling factors are fixed

 HGADJ.FX          = HGADJ0 ;
 EGADJ.FX          = EGADJ0 ;

* alternatively the internal balance / Govt savings can be fixed

* KAPGOV.FX        = KAPGOV0 ;
```

11.2 Other Closure Conditions

Technology Variables

```
* # Technology Closures for Factor Market

* Technology for CES production functions for Level 1 of production
nest

 ADXADJ.FX         = ADXADJ0 ;
 DADX.FX           = DADX0 ;

* Technology for CES aggregation functions for Level 2 of production
nest

 ADVAADJ.FX        = ADVAADJ0 ;
 DADVA.FX          = DADVA0 ;

* Technology for factor activity and region specific factor efficiency

 ADFD.FX(ff,a)      = ADFD0(ff,a) ;
```

Miscellaneous Fixed Variables

```
*## MISCELLANEOUS FIXED VARIABLES

* To use CPI as the numeraire fix CPI
```

```
 CPI.FX(r)          = CPI0(r)*numerchk ;

* To fix the real exchange rate fix ER and PPI

* PPI.FX(r)           = PPI0(r)*numerchk ;
```

11.3 Factor Market Closure

This, essentially, determines whether the foreign exchange market clears by variations in the exchange rate or the current account balance. There are situations where both need to be fixed, which means consideration needs to be given to the closure settings that allow to foreign exchange market to clear, e.g., a reduction in domestic absorption

```
*## FACTOR MARKET CLOSURE

*# Basic Factor Market Closure

 FSI.FX(insw,f)           = FSI0(insw,f) ;

* FS.FX(f)                 = FS0(f) ;
 WFDIST.FX(f,a)           = WFDIST0(f,a) ;
 WF.LO(f)                 = -inf ;
 WF.UP(f)                 = +inf ;

* activity specific capital selected activities (not all activities)
*$ontext
 FD.FX(k,a)              = FD0(k,a) ;
 WF.FX(k)                = WF0(k) ;
 WFDIST.LO(k,a)          = - inf ;
 WFDIST.UP(k,a)          = + inf ;
*$offtext

$ontext
* Surplus Labour

 FSI.LO(h,l)         = -inf ;
 FSI.UP(h,l)         = +inf ;

 FSISH.FX(h,l)       = FSISH0(h,l) ;

 WF.FX(l)               = WF0(l) ;
* FS.LO(l)                = -inf ;
* FS.UP(l)                = +inf ;

$offtext

$ontext
 WF.LO(f)$(NOT UEF(f))    = -inf ;
 WF.UP(f)$(NOT UEF(f))    = +inf ;
 WF.LO(f)$UEF(f)          = WF0(f) * mod_cont("numerchk") ;
 WF.UP(f)$UEF(f)          = + INF ;
$offtext
```