

STAGE_t: A User Guide

Scott McDonald

Addresses for correspondence:

cgemod
url: www.cgemod.org.uk
Email: scott@cgemod.org.uk
Email: jrs.mcdonald@gmail.com
Tel: +44 789 979 3671

Abstract

This paper is a user's guide to the STAGE CGE model. 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 GAMSIDE 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, database aggregations can be produced using SAMGator and access the model results using SeeResults. 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 is provided on that basis. Comments on the current content are encouraged.

Table of Contents

STAGE_t: A User Guide.....	1
Table of Contents	2
1. Introduction.....	3
2. Social Accounting Matrix	4
3. Structure of the STAGE Model Code	8
4. Model Data and Conditioning File	19
5. Compiling the Model's Excel Workbook	19
6. Model Calibration Checks	39
7. Experiment File Template.....	15
8. Compiling an Experiment Excel Workbook.....	34
9. Market Clearing and Model Closure Rules	39
10. Tax Rate Shocks	48
11. Technology Shocks	49
12. Analysis File	17
Annexes.....	50
A1. Aggregating a SAMs.....	50
References	57

1. Introduction

This User Guide is intended to support use of the STAGE model. It is not an objective of this guide to provide technical information about the STAGE 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.

DRAFT

2. 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
- iii) the (arguably) greater accessibility of the information for policy makers.

Table 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 does not contain information in the inactive sub matrices and conforms to the rules of a SAM.¹ 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 consistent with the SNA are:

- 1) The SAM is assumed to contain only a single 'level' of income distribution. However, fixed proportions are used in the functional distribution of income within the model and therefore a reduced form of an SNA SAM using apportionment (see Pyatt, 1989) will not violate the model's behavioural assumptions.
- 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 government expenditures. These adjustments do not change the information content of the SAM, but they do simplify the modeling process. However, they do

¹ 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 so as to further generalise the model.

have the consequence of creating a series of reserved names that are required for the operation of the model.²

Table 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 each and every agent are exactly matched by expenditure transactions 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.

² These and other reserved names are specified below as part of the description of the model.

Given these definitions of a SAM the transactions recorded in a SAM are easily interpreted. In Table 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 so as 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. By definition the column and row totals must equate and these transaction totals can be expressed as an implicit price times 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 in order to supply a commodity to the economy and hence the implicit 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 into 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.

2.1.3 Other Data

In addition to the SAM, which records transactions in value terms, two additional databases are used by the model. The first records the ‘quantities’ of primary inputs used by each activity. If such quantity data are not available then the entries in the factor use matrix are the same as those in the corresponding sub matrix of the SAM. The second series of additional data are the elasticities of substitution for imports and exports relative to domestic commodities, the elasticities of substitution for the CES production functions, the income elasticities of demand for the linear expenditure system and the Frisch (marginal utility of income) parameters for each household.

Other data of interest to modellers include estimates of energy usage and emissions, biofuels and land use. None of these data are used in this variant of the model, but there are variants of the STAGE model that do use such satellite data to model energy use and emissions and other relevant policy considerations.

2.1.4 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.

Structure of the Computer Code Files

3. Structure of the STAGE Model Code

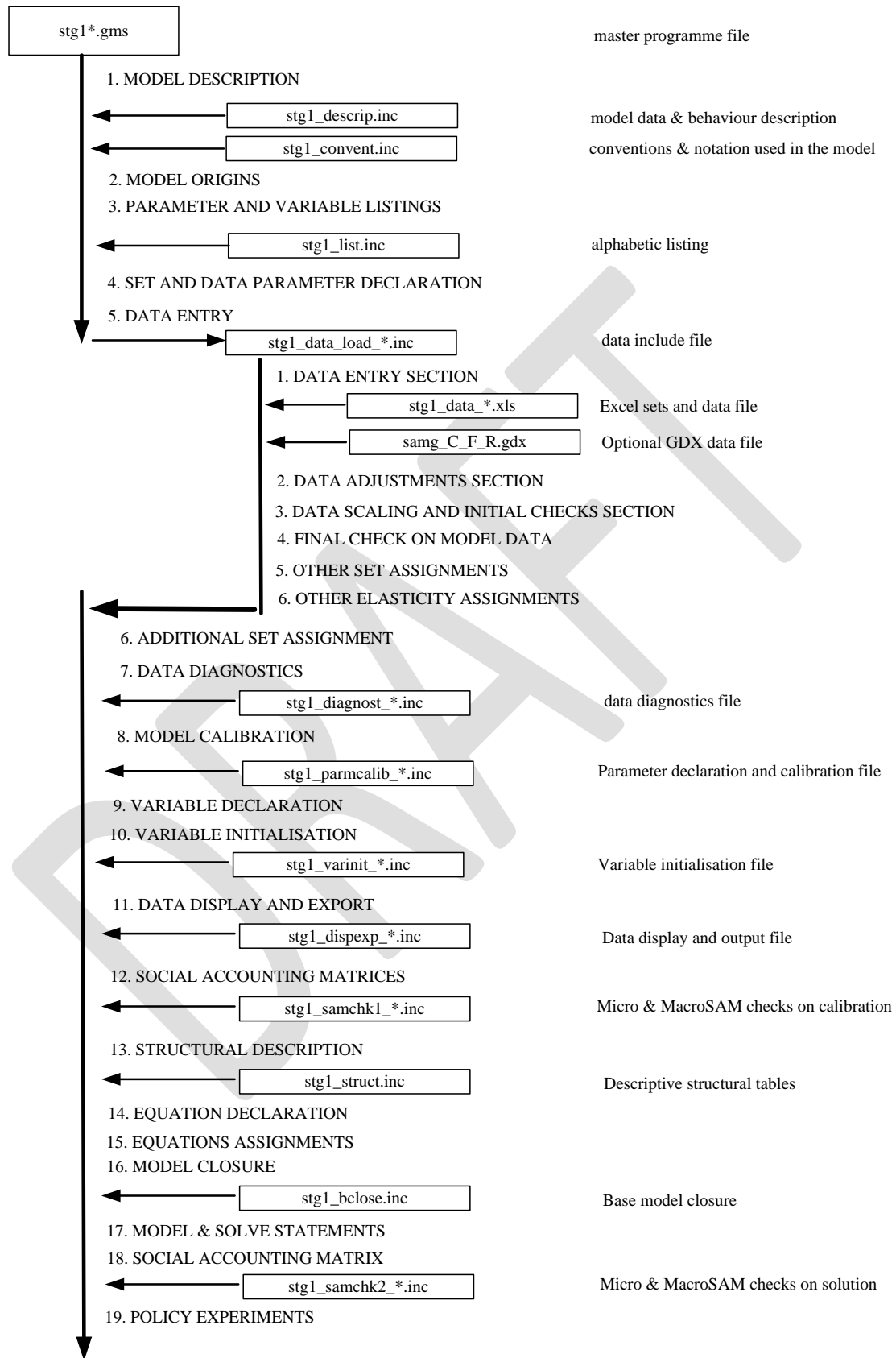
The STAGE programme code is relatively complex and consequently a modular structure is adopted; this takes the form of a series of *INCLUDE* files that are called from the main programme – ****.gms* – file and two MS Excel workbooks that contain model and experiment/simulation data. In addition the model makes use of number of *IF* statements that are used to control which modules are implemented in any particular run of the model. Finally the model is set up to allow users, with minor adjustments, to 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.³ 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.

In this section 2 schematics are presented to assist the user in understanding the structure of the programme's code. The first deals with the code for the basic model while the second relates to the template experiment file.

³ Optionally some additional data can be input from a GDX file if this is more convenient.

Figure 4.1.1 STAGE File Structure



4.1 STAGE Programme Structure

The schematic illustration of the structure of the STAGE model file – `smod1_*.gms` – in Figure 4.1.1 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:

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

these are explored in detail 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 checks⁴ 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.⁵

After the data has been conditioned and checked all the models parameters are declared and assigned (`stg1_parmcalib.inc`) followed by the declaration of all model variables⁶ and then the variables are initialised (`stg1_varinit1.inc`). This initialisation of the variables includes an (implicit) double check on the parameter calibration of the initial values for (nearly) all variables. All the parameters and (initial values for) variables are then displayed and exported to GDX (`stg1_disexp.inc`).

This is followed by two further `*.inc` files. The first (`stg1_samchk1.inc`) checks that the calibrated parameter and variable values generates ‘macro’ and ‘micro’ SAMs that are

⁴ Domain checking on the sets and data are conducted when the data are being loaded through options included in `GDXXRW`.

⁵ 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.

⁶ 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.

consistent with the data (after any adjustments) that were loaded. The second (`stg1_struct.inc`) computes a series of reports that describe the structure of the economy contained within the base data; these reports provide data for use when analysing the results from simulation exercises and input to written project reports.

DRAFT

4. 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.

6.1 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.

- 1a. Excel workbook sets and data converted to.gdx here
The user replaces [filename] with the appropriate filename for the new Excel data file in GDXXRW call statement. i.e.,
`$CALL "GDXXRW i=[filename].xls o=data_in.gdx INDEX=LAYOUT!A4"`
- 1b. 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.
- 1c. Defining SETS by exclusions from previously defined SETS
There should be no need to make changes here unless the model is being changed
- 1d. Defining MAPPING SETS by using subset information
There should be no need to make changes here unless the model is being changed.
- 1e. Initial SAM check

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

```
*---- 1a. Excel workbook sets and data converted to.gdx here

$CALL "GDXXRW i=smod1_data_*.xls o=data_in.gdx index=Layout!A4"
```

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.

6.2. 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.

6.3. 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.

6.4 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"
```

6.5 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

6.6. 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.

DRAFT

7. Experiment File Template

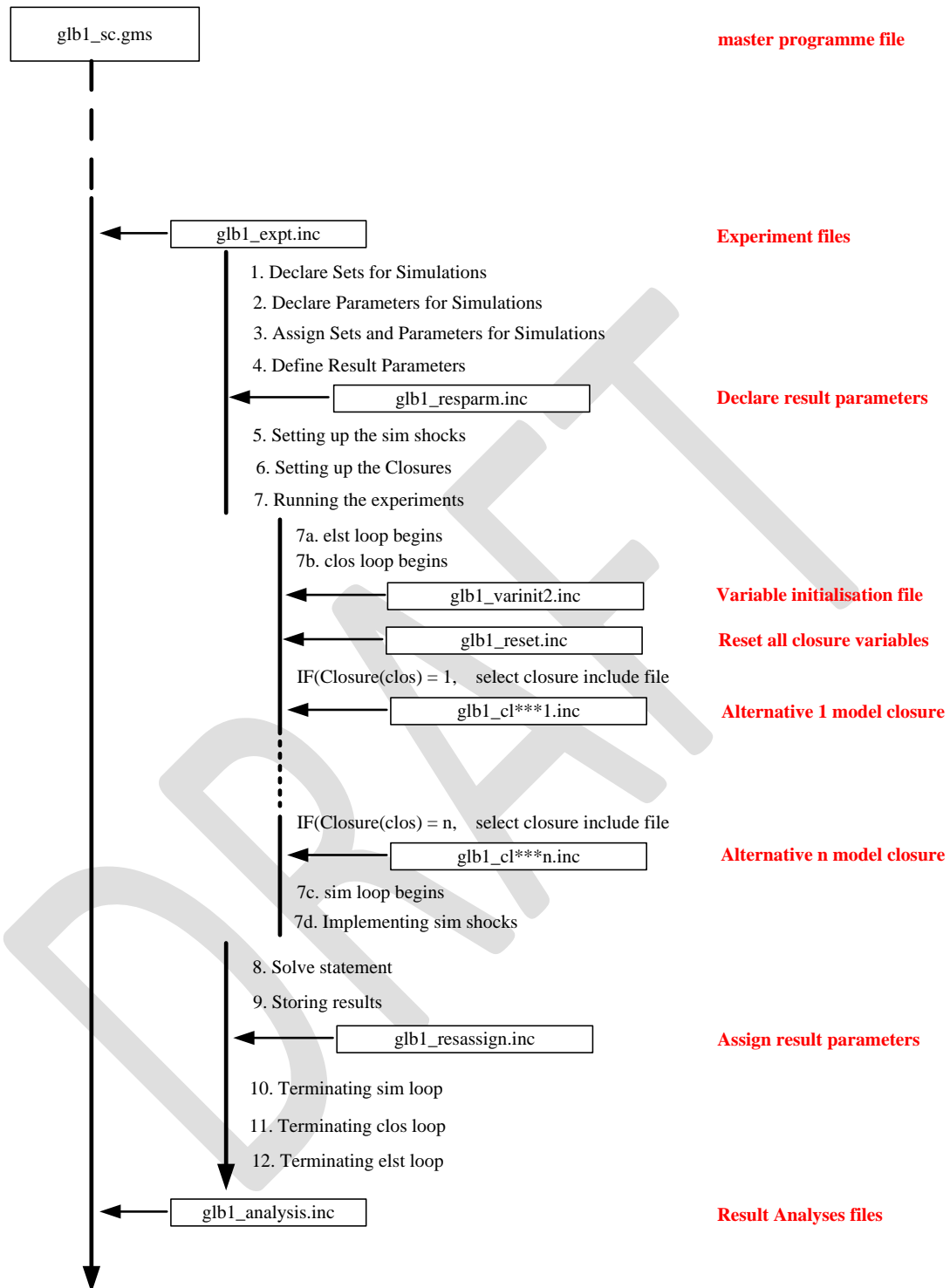
The experiment file is where the user has to be most active. The model file is largely straightforward and any changes the user would make are standardised, but the range of potential changes in the experiment file are very large and defy standardisation.

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`.

Figure 9.1.1 GLOBE Experiment File Structure



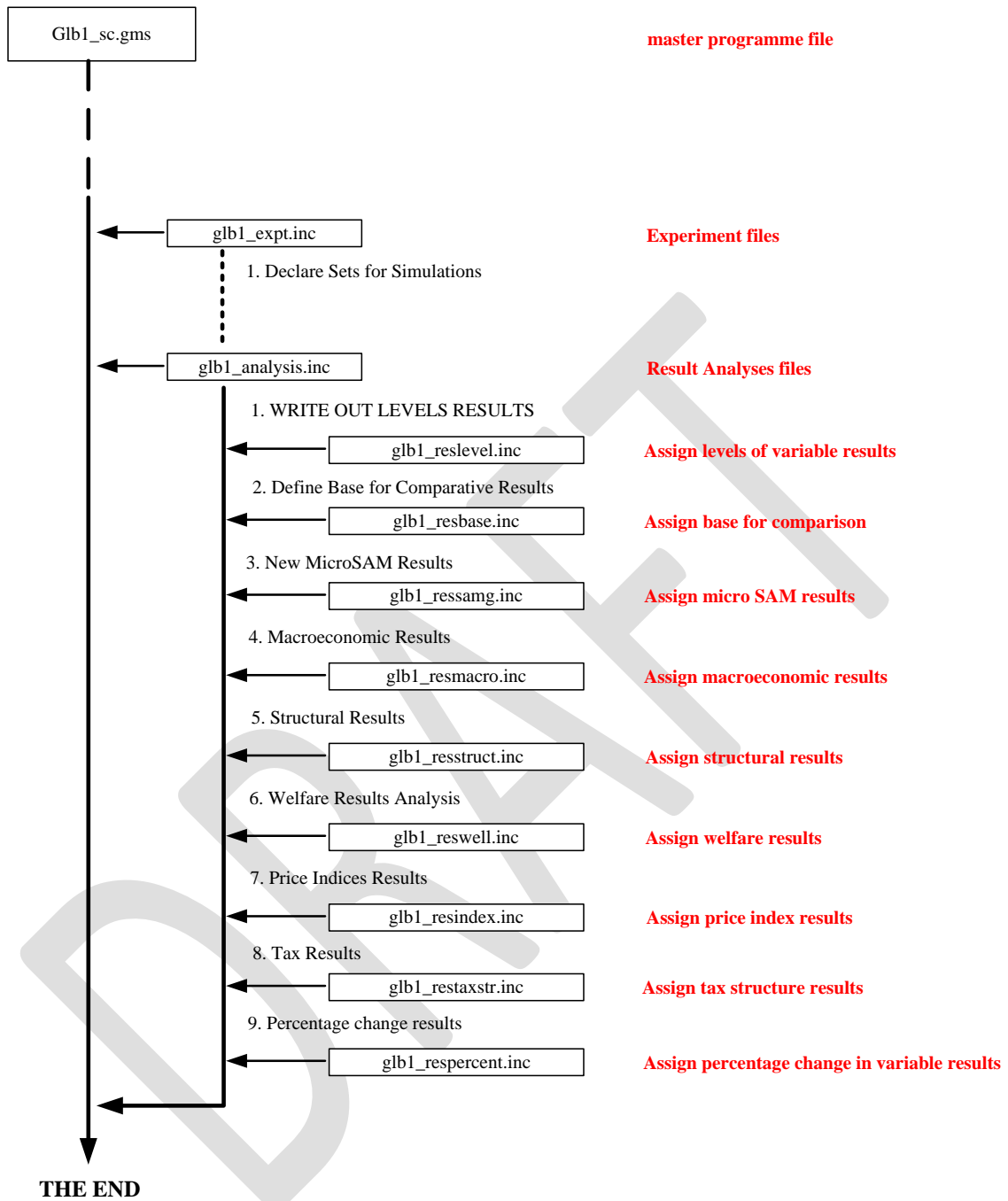
With the completion of each simulation the levels values for the results are assigned to the appropriate results parameter – all results parameters use the same name as the associated variable prefixed with ‘res’. These levels results are subsequently used to derived additional (analytical) results (see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

12. 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 14.1 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).

DRAFT

Figure 14.1 GLOBE Results Analysis File Structure



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

Model and Experiment Data Files

5. Compiling the Model's Excel Workbook⁷

Typically 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. While this arrangement increases the number of files a user has to access and pass to the model it does have advantages in flexibility and replicability. This section is concerned with the Excel workbook that has information used to calibrate the model.

7.1 Model Sets

7.1.1 Sets Worksheet

The 'sets' worksheet contains the vast majority of the sets and subsets used by the model. There are two major sets, *sac* and *r*, where *sac* contains all the accounts for a single region and *r* the set of regions in the model. The member and descriptions for *sac* and *r* can be obtained by copying and pasting from the SAMgator file used to generate the aggregate SAM. It is important to ensure that the set member names and descriptions are pasted into the correct cells in the workbook and worksheet; if not errors will be generated. How the first few columns of the 'sets' worksheet look to the user is shown in Figure 7.1.1.1.

⁷ 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 GLOBE 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 GAMSIDE (> Help > docs > gams > gdxutils.chm or gdxutils.pdf).

Figure 7.1.1.1 Sets Worksheet Part 1

New SAM accounts		Regions		Regional Accounts for numeraire		Commodities
sac	(description)	r	(description)	ref		c
ogran	Grains	deu	Germany	deu		ogran
cscb	Sugar cane and beet	ita	Italy	ita		cscb
coocrp	Other crops	aut	Austria	aut		coocrp
cpbf	Plant based fibres	gbr	United Kingdom	gbr		cpbf
clstk	Livestock	fra	France	fra		clstk
cmilk	Raw milk	bnl	Benelux	bnl		cmilk
caprd	Other animal products	espt	Spain and Portugal	espt		caprd
cmns	Minerals	reu	Rest of EU 15	reu		cmns
cmeat	Meat	pol	Poland	reur		cmeat
cmprd	Meat products	hun	Hungary	roecd		cmprd
cvof	Vegetable oils and fats	cze	Czech Republic			cvof
odair	Dairy products	reur	Rest of EU 25			odair
csuga	Sugar	robu	Romania and Bugaria			csuga
cofd	Other food products	tur	Turkey			cofd
cbtob	Beverages and tobacco	roecd	Rest of the OECD			cbtob
cbind	Base industries	cis	Former communist block			cbind
cmanu	Manufactures	merc	MERCUSOR			cmanu
cmach	Machinery	row	Rest of the World			cmach
cutil	Utilities	glo	Globe			cutil
ccns	Construction					ccns
ctrd	Trade and communication					ctrd
ctran	Transport					ctran
cserv	Services					cserv
agran	Grains					
ascb	Sugar cane and beet					
aoocrp	Other crops					
apbf	Plant based fibres					
alstk	Livestock					
amilk	Raw milk					
aaprd	Other animal products					
amins	Minerals					

Note that in addition to the sets *sac* and *r* Figure 7.1.1.1 also contains a third set *ref*. This set is a subset of *r* that defines the regions that are included in the exchange rate index used by the model. The members of *ref* are selected by the user subject to the following conditions:

- *ref* must contain at least one member of *r*; and
- *ref* must NOT contain *glo*.

The model uses a substantial number of subsets of *sac*; some of these are shown in Figure 7.1.1.2. The main subsets are

- *c(sac)* commodity accounts
- *a(sac)* activity accounts
- *f(sac)* factor accounts
- *tmr(sac)* tariff accounts used in GDX program
- *ter(sac)* export subsidy accounts used in GDX program
- *tff(sac)* factor tax account used in GDX program
- *h(sac)* household accounts

- $g(sac)$ government accounts
- $i(sac)$ investment accounts
- $owatpmarg(sac)$ Trade and transportation margin services
- $w(sac)$ Rest of world trade partners

The user needs to manually assign these subsets when setting up a new version of the GLOBE model. It is a sensible practice to copy and paste the subsets from the column containing the *sac* account names so as to avoid typing errors. NB the GLOBE programme will abort if the accounts names in *sac* and ALL the subsets are NOT identical.

In addition to the subsets of *sac* there are a number of subsets of the subsets of *sac*. These are

- $cagr(c)$ Agricultural commodities
- $cnat(c)$ Natural resource commodities
- $cfcd(c)$ Food commodities
- $cind(c)$ Industrial commodities
- $cuti(c)$ Utility commodities
- $cser(c)$ Service commodities
- $aagr(a)$ Agricultural activities
- $anat(a)$ Natural resource activities
- $afd(a)$ Food activities
- $aind(a)$ Industrial activities
- $auti(a)$ Utility activities
- $aser(a)$ Service activities
- $l(f)$ Labour Factors
- $ls(l)$ Skilled Labour Factors
- $lu(l)$ Unskilled Labour Factors
- $k(f)$ Capital Factors
- $Ind(f)$ Land factors

The user needs to define the memberships of these subsets manually following conventions that are obvious from the naming of the subsets. Again copying and pasting is a way to avoid typing errors. The subsets of *c* and *a* are used extensively in standardised summary statistics used to describe the database and/or analyse results from model simulations; inappropriate assignments will produce tables of results that contain limited and/or misleading information.

Empty subsets of *c* and *a* are legitimate. The subsets of *f* are primarily used for model and simulation purposes.

NB: The sets *cagg* and *aagg* 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 7.1.1.2 Sets Worksheet Part 2

	I	J	P	Q	R	X	Y	Z	AE	AF	AG	AH	AI	AJ	AK	AL
1																
2																
3	Commodities	Agricultural Commodities	Commodity Aggregates	Activities	Agricultural Activities	Activity Aggregates	Factors	Labour Factors	Households	Import taxes	Export taxes	Taxes on Factors	Government	Savings/ Investment	all margins on imports	Rest of the World
4	c	cagr	cagg	a	aagr	aagg	f	l	h	tmr	ter	tff	g	i	owatpmarg	w
5	cgran	cgran	cagr	agran	agran	aagr	land	UnSkLab	HOUS	tmdeu	tedeu	tfland	SALTAX	KAP	owatpdeu	wdeu
6	oscb	oscb	onat	asob	asob	anat	UnSkLab	SKLab		tmita	teita	tUnSkLab	PRODTAX		owatpita	wita
7	coerp	coerp	ofd	aoerp	aoerp	afd	SKLab			tmaut	teaut	tSKLab	FACTAX		owatpaut	waut
8	opbf	opbf	cind	apbf	apbf	aind	cap			tmgbr	tegbr	tfoap	DIRTAX		owatpabr	wabr
9	clstk	clstk	cuti	alstk	alstk	auti	natres			tmfra	tefra	tfnatres	Govt		owatpfra	wfra
10	omlk	omlk	cser	amlk	amlk	aser				tmbnl	tebnl				owatpbnl	wbnl
11	caprd	caprd		aaprd	aaprd					tmespt	teespt				owatpespt	wespt
12	omins			amins						tmreu	tereu				owatpreu	wreu
13	omeat			ameat						tmpol	tepol				owatppol	wpol
14	emprd			amprd						tmhun	tehun				owatphun	whun
15	evof			avof						tmoze	tecze				owatpze	wze
16	edair			adair						tmreur	tereur				owatpreur	wreur
17	csuga			asuga						tmrobu	terobu				owatprobu	wrobu
18	cofd			afod						tmur	teur				owatptur	wtur
19	cbtob			sbtob						tmroecd	teroeed				owatproecd	wroecd
20	cbind			abind						tmcis	tecis				owatpcis	wcis
21	cmanu			amanu						tmmerc	temerc				owatpmerc	wmerc
22	cmach			amach						tmrow	terow				owatprow	wrow
23	cutil			autil						tmglo	teglo				owatpglo	wglo
24	cens			aens												
25	ctrd			atrd												
26	ctran			atran												
27	cserv			aserv												
28																
29																

7.1.2 Maps Worksheet

In a number of calibration statements and equations in the model it is necessary to match data that are recorded using slightly different but related labels. This requires the provision of a series of mapping sets that define the matching of set elements. These mapping sets are defined in the ‘maps’ worksheet. The user needs to compile this worksheet; this is illustrated in Figure 7.1.2.1.

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*. The user needs to enter the sets in the appropriate columns of the ‘maps’ worksheet. When compiling these mapping sets it is

important to be very careful to ensure the mappings are correct. Notice how in all cases bar one (see below) that *set1* and *set2* have common suffixes; these must always match. If the set *sac* is constructed from correctly ordered output from SAMgator then the ordering within the subsets of *sac* will be consistent and the ‘maps’ worksheet can be compiled by copying and pasting from the subsets of *sac*.

Figure 7.1.2.1 Maps Worksheet

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Maps																
2																	
3																	
4	MAP_V	TMR(w,tmr)		MAP_TMR	w(tmr,w)		MAP_V	TER(w,ter)		MAP_TER	V(ter,w)		MAPR	V(r,w)		MAPV	R(w,r)
5	wdeu	tmdeu		tmdeu	wdeu		wdeu	tedeu		tedeu	wdeu		deu	wdeu		wdeu	deu
6	wita	tmita		tmita	wita		wita	teita		teita	wita		ita	wita		wita	ita
7	waut	tmaut		tmaut	waut		waut	teaut		teaut	waut		aut	waut		waut	aut
8	wgbr	tmgbr		tmgbr	wgbr		wgbr	tegbr		tegbr	wgbr		gbr	wgbr		wgbr	gbr
9	wfra	tmfra		tmfra	wfra		wfra	tefra		tefra	wfra		fra	wfra		wfra	fra
10	wbnl	tmbnl		tmbnl	wbnl		wbnl	tebnl		tebnl	wbnl		bnl	wbnl		wbnl	bnl
11	wespt	tmespt		tmespt	wespt		wespt	teespt		teespt	wespt		espt	wespt		wespt	espt
12	wreu	tmreu		tmreu	wreu		wreu	tereu		tereu	wreu		reu	wreu		wreu	reu
13	wpol	tmpol		tmpol	wpol		wpol	tepol		tepol	wpol		pol	wpol		wpol	pol
14	whun	tmhun		tmhun	whun		whun	tehun		tehun	whun		hun	whun		whun	hun
15	wcze	tmcze		tmcze	wcze		wcze	tecze		tecze	wcze		cze	wcze		wcze	cze
16	wreur	tmreur		tmreur	wreur		wreur	tereur		tereur	wreur		reur	wreur		wreur	reur
17	wrobu	tmrobu		tmrobu	wrobu		wrobu	terobu		terobu	wrobu		robu	wrobu		wrobu	robu
18	wtur	tmtur		tmtur	wtur		wtur	tetur		tetur	wtur		tur	wtur		wtur	tur
19	wroecd	tmroecd		tmroecd	wroecd		wroecd	teroeecd		teroeecd	wroecd		roecd	wroecd		wroecd	roecd
20	wcis	tmcis		tmcis	wcis		wcis	tecis		tecis	wcis		cis	wcis		wcis	cis
21	wmerc	tmmerc		tmmerc	wmerc		wmerc	temerc		temerc	wmerc		merc	wmerc		wmerc	merc
22	wrow	tmrow		tmrow	wrow		wrow	terow		terow	wrow		row	wrow		wrow	row
23	wglo	tmglo		tmglo	wglo		wglo	teglo		teglo	wglo		glo	wglo		wglo	glo
24																	
25																	
26																	

One complication in the ‘maps’ worksheet is the mapping set $\text{map_c_w_marg}(c, w, \text{owatpmarg})$, which is a triple. The complexity here derives from the presence of the mapping to the set *c*. Since both *w* and *owatpmarg* are sets based on the regions the mapping here is straightforward since it involves matching regions, but *c* refers to types of trade margins of which there can be three. Thus when constructing this mapping set each pairing of *w* and *owatpmarg* needs to be repeated for each member of *c*, i.e., for each type of trade margin.

7.1.3 Model Sets

The final worksheet of sets for the model is ‘mod_sets’. This worksheet defines subsets that are used to control behavioural attributes of the model. Specifically there are two behavioural attributes that are covered; the identification of activities and regions for which the top level of the production nests are characterised by zero substitution. The default setting in the model is that the top level nest is a CES nest and therefore the user specifies only those activities and regions for which the Leontief option is wanted. (NB There is a minimum intermediate input

share for which the CES default operates; this can be changed in the ‘controls’ worksheet – see below).

Figure 7.1.3.1 Model Sets Worksheet

	A	B	C
1	Model Structure Sets		
2			
3	Sets to control production structure Level 1		
4	aleon	rleon	
5			
6			
7			
8			
9			
10			

7.1.4 Results 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 (`womod_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 7.1.4.1 Results Set Worksheet

1	A	B	C	D	E	F	G	H
1	Results sets							
2								
3	Sets for Structural results							
4	stab1	(description)	stab2	(description)	stab3	(description)		
5	stQCDTOT	private consumption	stIMPregFOB	Imports valued fob by region shares	stPRIVregDEM	private domestic demand by total domestic de		
6	stQGDTOT	government consumption	stIMPcomFOB	Imports valued fob by commodity shares	stPRIVcomDEM	private domestic demand by total commodity c		
7	stQINVTOT	investment consumption	stIMPregCIF	Imports valued cif by region shares	stGOVTregDEM	government domestic demand by total domes		
8	stQABSORP	absorption	stIMPcomCIF	Imports valued cif by commodity shares	stGOVTcomDEM	government domestic demand by total commo		
9	stIMPORP	import demand	stIMPregDOM	Imports valued at domestic prices by region shares	stINVDregDEM	investment domestic demand by total domesti		
10	stEXPORT	export supply	stIMPcomDOM	Imports valued at domestic prices by commodity shares	stINVDcomDEM	investment domestic demand by total commo		
11	stGDPVA	GDP from value added	stEXPregFOB	Exports valued fob by region shares	stINTDregDEM	intermediate domestic demand by total domes		
12	stGDPEXP	GDP from expenditure	stEXPcomFOB	Exports valued fob by commodity shares	stINTDcomDEM	intermediate domestic demand by total commv		
13	stSUPPLY	total domestic production	stEXPregDOM	Exports valued at domestic prices by region shares	stMAKEcomSUP	domestic make by commodity shares		
14	stUSE	total intermediate inputs	stEXPcomDOM	Exports valued at domestic prices by commodity shares	stSUPPcomSUP	total domestic supply by commodity shares		
15	stYHTOT	household income	stSmImpSh	Small Import shares by commodity shares	stQcomSUP	aggregate import supply by commodity shares		
16	stYGTOT	government income	stSmExpSh	Small Export shares by commodity shares	stQcomSUP	domestic output to domestic market by comm		
17	stYFTOT	factor income			stQeconSUP	domestic output to export market by commod		
18	stYFDISTTOT	distributed factor income						
19	stYFLABTOT	labour factor income						
20	stTOTSAV	total savings						
21	stHOSAV	household savings						
22	stGOVSAV	government savings						
23	stFSAV	foreign savings						
24	stDEPPREC	depreciation						
25	stMTAX	Import tariff revenue						
26	stETAX	Export tax revenue						
27	stSTAX	Sales tax revenue						
28	stITAX	Indirect tax revenue						
29	stFTAX	Factor tax revenue						
30	stHTAX	Household income tax revenue						
31								
32								

Factor Use and Ownership Satellite Accounts

Factor Use Accounts (FACTUSE)

Factor Ownership Accounts (FACTINS)

7.2 Elasticities

In addition to the transactions data derived from the (aggregate) SAM the model also needs a series of elasticities. The elasticities included in the GTAP database are made available to the model through the GDX database produced by the SAMgator programme, but the GTAP model, and hence the GTAP database, does not use as many exogenous elasticities as GLOBE. In particular GLOBE has a two level production nest where the top level can be CES – the default – or Leontief – as in the GTAP model, CET functions on the export side and a linear expenditure system. Thus GLOBE requires a number of elasticities that are not included in the GTAP database and moreover, because of the use of CET functions for

exports, the ‘high’ values of the import substitution elasticities in the GTAP database can be reduced.⁸

The user has (broadly) three choices as to the elasticities used by the model; user defined elasticities that are supplied to the model from Excel, GTAP based elasticities that are supplied in the GDX database, or some combination of the two. The user makes this choice through the `mod_control` parameters that are described in section 7.3. This section describes the format used to record elasticity data in Excel.

7.2.1 Import Substitution Elasticities

Import substitution elasticities are needed for both levels of the Armington functions. These are recorded in two worksheets, ‘`comelastm`’ and ‘`comelastm`’, where the former contains the elasticities of substitution between aggregate imports and domestic commodities (model parameter *ELASTM*) and the latter the elasticities of substitution between aggregate imports from different regions (model parameter *ELASTRM*). Both are two dimensional matrices that allow for differences in the substitution elasticities by commodity and region.⁹ The user needs to define the row and column accounts from the sets *c* and *r* that are determined by the aggregation used; this is illustrated in Figure 7.2.1.1.

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

⁸ It is argued that GTAP uses ‘high’ substitution elasticities so that the terms of trade effects are ‘damped down’ and that part of reason that the GTAP model can produce large terms of trade effects is the fact that export supplies are driven by import demands. When there is a CET function on exports domestic producers are able to reallocate output between export and domestic markets in response to changes in export prices, thereby ‘damping down’ the terms of trade effects for any given set of import substitution elasticities. But the interaction effects between Armington and CET elasticities are not fully worked out.

⁹ This is different to the case in the GTAP model where the substitution elasticities are common across regions and are only differentiated by commodity.

Figure 7.2.1.1 Substitution Elasticities for GLOBE Model - illustration

Model Specific Commodity Elasticities									
	A	B	C	D	E	F	G	H	I
1	Model Specific Commodity Elasticities								
2	Elasticities								
3									
4	ELASTM	ELASTM	ELASTM	ELASTM	ELASTM	ELASTM	ELASTM	ELASTM	
5			deu	ita	aut	gbr	fra	bnl	espt
6	Grains	cgran	2	2	2	2	2	2	2
7	Sugar cane and beet	cscb	2	2	2	2	2	2	2
8	Other crops	coorp	2	2	2	2	2	2	2
9	Plant based fibres	cpbf	2	2	2	2	2	2	2
10	Livestock	clstk	2	2	2	2	2	2	2
11	Raw milk	cmilk	2	2	2	2	2	2	2
12	Other animal products	caprd	2	2	2	2	2	2	2
13	Minerals	cmins	2	2	2	2	2	2	2
14	Meat	cmeat	2	2	2	2	2	2	2
15	Meat products	cmprd	2	2	2	2	2	2	2
16	Vegetable oils and fats	cvof	2	2	2	2	2	2	2
17	Dairy products	cdair	2	2	2	2	2	2	2
18	Sugar	csuga	2	2	2	2	2	2	2
19	Other food products	cofd	2	2	2	2	2	2	2
20	Beverages and tobacco	cbtob	2	2	2	2	2	2	2
21	Base industries	cbind	2	2	2	2	2	2	2
22	Manufactures	cmanu	2	2	2	2	2	2	2
23	Machinery	cmach	2	2	2	2	2	2	2
24	Utilities	cutil	2	2	2	2	2	2	2

7.2.2 Export Transformation Elasticities

Export transformation elasticities are needed for both levels of the CET functions. These are recorded in two worksheets, ‘comelaste’ and ‘comelastre’, where the former contains the elasticities of transformation between aggregate exports and domestic commodities (model parameter *ELASTE*) while the latter worksheet contains the elasticities of transformation between aggregate exports to different regions (model parameter *ELASTRE*). Both are two dimensional matrices that allow for differences in the transformation elasticities by commodity and region.¹⁰ The user needs to define the row and column accounts from the sets *c* and *r* used determined by the aggregation used; this is illustrated in Figure 7.2.1.1.

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

7.2.3 Production Function Elasticities

Substitution elasticities are required for both levels of the nested CES production functions. These are recorded in two worksheets, ‘actelastx’ and ‘actelastva’, where the former contains the elasticities of substitution between aggregate intermediate inputs and aggregate value added (model parameter *ELASTX*) and the latter the elasticities of substitution between

¹⁰ There is no equivalent to the CET elasticities in the GTAP model because of differences in behavioural assumptions. When CET elasticities are derived from the GTAP database they are in fact based on the Armington elasticities used by GTAP.

primary inputs (model parameter *ELASTVA*). Both are two dimensional matrices that allow for differences in the substitution elasticities by activity and region.¹¹ The user needs to define the row and column accounts from the sets *a* and *r* used determined by the aggregation used; this is illustrated in Figure 7.2.1.1.

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

74.2.4 Linear Expenditure System Elasticities

The elasticities of substitution used for the linear expenditure systems (LES) are only supplied from Excel, and require the user to provide both income elasticities of demand and Frisch elasticities. The income elasticities of demand are recorded in the worksheet ‘*incelast*’ and supply data for the model parameter *ELASTY*; this table make be slightly confusing because of the additional column of labels – “*hou*” – but since there is only a single household in each region for the GTAP model this is only included for model convenience and (future) flexibility. Except for the additional column of identifiers for this table, *ELASTY*, is identical to those for commodities and activities see Figure 7.2.1.1. The determination of appropriate elasticities is the responsibility of the user; the numbers in the model template are placeholders **only**.

The Frisch elasticities are recorded in the worksheet ‘*frischelast*’ and supply data for the model parameter *ELASTF*; note how these elasticities are defined by region and household; the worksheet ‘*frischelast*’ is shown in Figure . The determination of appropriate elasticities is the responsibility of the user; the numbers in the model template are placeholders **only**.

The default values for *ELASTY* and *ELASTF* are all minus one. This is a special case whereby the LES utility functions reduce to Cobb-Douglas utility functions for all regions.

¹¹ This is different to the case in the GTAP model where the top level of the production nest is fixed as Leontief, i.e., zero substitution elasticities. When CES elasticities are derived from the GTAP database they are in fact based on the value added substitution elasticities used by GTAP.

Figure 7.2.4.1 Frisch Elasticities for GLOBE

	A	B	C	D
1	Model Specific Household Elasticities			
2	Elasticities			
3				
4	ELASTF	ELASTF	ELASTF	
5			hous	
6	Germany	deu	-1	
7	Italy	ita	-1	
8	Austria	aut	-1	
9	United Kingdom	gbr	-1	
10	France	fra	-1	
11	Benelux	bnl	-1	
12	Spain and Portugal	espt	-1	
13	Rest of EU 15	reu	-1	
14	Poland	pol	-1	
15	Hungary	hun	-1	
16	Czech Republic	cze	-1	
17	Rest of EU 25	reur	-1	
18	Romania and Bugaria	robu	-1	
19	Turkey	tur	-1	
20	Rest of the OECD	roecd	-1	
21	Former communist block	cis	-1	
22	MERCUSOR	merc	-1	
23	Rest of the World	row	-1	
24	Globe	glo	-1	
25				
26				

7.3 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 GLOBE model is to concentrate as many of these aspects of the programme as practical in an Excel worksheet – the ‘controls’ worksheet. This has two tables of entries, the first, `mod_control`, contains a number of elements that pass parameters to the model that control aspects of model structure, while the second, `flow_control`, passes parameters to the model that are used to determine whether various IF statements are implemented, and hence whether certain non crucial components of the programme run.

7.3.1 Model Controls

The `mod_control` parameters are contained in a data table that consists of various elements – Figure 7.3.1.1. Also included are examples of typical default values and brief descriptions of the role of each parameter in the configuration of the model.

Figure 7.3.1.1 Model Controls

	A	B	C	D	E
1	Control Parameters				
2	These values are used to initialise various parameters that condition the model and control th				
3					
4					
5	mod_control				
6					
7	mcons				
8	numerchk	1			IF 1 then default, if NOT 1 then check on numeraire
9	mintrans	0.00001			Minimum values for transactions in unscaled SAMG
10	minaqxsh	0.1			Minimum share of intermediates in cost for aqx
11	samscale	0.001			SAM scaling factor and initial value for autoscaling
12	scaltarget	100			target level for auto scaling
13	scalprop	0.95			proportion of NON zero elements that must be below the target level
14	armelast	2			IF 1 then ALL Armington elasticities from Excel; IF 2 ALL from 'GTAP'
15	armscal	2			Scaling value for GTAP Armington elasticities - divided by armscal
16	cetelast	2			IF 1 then ALL CET elasticities from Excel; IF 2 ALL from 'GTAP'
17	cetscal	2			Scaling value for GTAP CET elasticities - divided by cetscal
18	qxelast	2			IF 1 then ALL Level 1 production elasticities from Excel; IF 2 ALL from 'GTAP'
19	qxscal	2			Scaling value for GTAP Level 1 Production elasticities - divided by qxscal
20	qvaelast	2			IF 1 then ALL Level 2 production elasticities from Excel; IF 2 ALL from 'GTAP'
21	toldiffsam	0.00001			Tolerance level for differences between data and solution SAMs
22					

The parameters in lines 14 to 20 (inclusive) are of particular importance. These parameters determine which elasticities are used to calibrate the model; if *armelst*, *cetelast*, *qcelast* and *qvaelast* are equal to 1 then the respective elasticities come from Excel whereas if they are equal to 2 the respective elasticities come from GDX. The *mod_control* parameter also allows the user the scale any elasticities taken from GTAP using uniform scaling parameters *armscal*, *cetscal* and *qxscal*.

7.3.2 Flow Controls

The flow controls are 1/0 parameters; if the parameter has a value of 1 then an action is implemented and if its value is zero it is not implemented. Overwhelmingly the flow control parameters determine whether an If statement is implemented or not; when a statement is implemented it typically controls output, either by running an *INCLUDE* file that produces output or implementing a display or export command in GAMS. The parameters in the *flow_control* table are shown in Figure 7.3.2.1.

Figure 7.3.2.1 Flow Controls

28				
29	flow_control			
30				
31	fcns			
32	disp1	1	IF 1 then implemented; IF 0 NOT implemented	Control on DISPLAY Statements for read in SETS and DATA
33	disp2	1	IF 1 then implemented; IF 0 NOT implemented	Control on parameter DISPLAY Statements
34	disp3	1	IF 1 then implemented; IF 0 NOT implemented	Control on variable initialisation DISPLAY Statements
35	outGD%modS	1	IF 1 then implemented; IF 0 NOT implemented	Control on GD% Unload for Model SAM
36	outGD%parm	1	IF 1 then implemented; IF 0 NOT implemented	Control on GD% Unload for ALL model parameters
37	outGD%var0	1	IF 1 then implemented; IF 0 NOT implemented	Control on GD% Unload for ALL base values for variables
38	outGD%varL	1	IF 1 then implemented; IF 0 NOT implemented	Control on GD% Unload for ALL initialised values for variables
39	outGD%data	1	IF 1 then implemented; IF 0 NOT implemented	Control on GD% Unload for ALL model base data
40	mic%gchk	1	IF 1 then implemented; IF 0 NOT implemented	Control for checking at micro SAM% level
41	reslevel	1	IF 1 then womod_reslevel runs; IF 0 then womod_reslevel does not run	
42	ressamg	0	IF 1 then womod_ressamg runs; IF 0 then womod_ressamg does not run	
43	resmacro	1	IF 1 then womod_resmacro runs; IF 0 then womod_resmacro does not run	
44	resstruct	1	IF 1 then womod_resstruct runs; IF 0 then womod_resstruct does not run	
45	reswelf	1	IF 1 then womod_reswelf runs; IF 0 then womod_reswelf does not run	
46	resindex	0	IF 1 then womod_resindex runs; IF 0 then womod_resindex does not run	
47	restaxstr	0	IF 1 then womod_restaxstr runs; IF 0 then womod_restaxstr does not run	
48	resperc	1	IF 1 then womod_resperc runs; IF 0 then womod_resperc does not run	
49	resbase	1	IF 1 then basis for comparative results are initial values; IF 2 then bases are from an experiment	
50				
51				

7.4 Layout Worksheet

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 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 7.4.1 Data ‘Layout’ Worksheet

	A	B	C	D	E	F	G	H
1	LAYOUT							
2								
3	Data Typ	Name	Location	Row dim	Column	Total dimension		
4				rdim	odim	dim		
5								
6	dset	sac	sets!A5	1				
7	dset	r	sets!D5	1				
8	dset	ref	sets!G5	1				
9	dset	c	sets!I5	1				
10	dset	oagr	sets!J5	1				
11	dset	onac	sets!K5	1				
12	dset	ofd	sets!L5	1				
13	dset	ocind	sets!M5	1				
14	dset	ocuti	sets!N5	1				
15	dset	oser	sets!O5	1				
16	dset	ocagg	sets!P5	1				
17	dset	a	sets!Q5	1				
18	dset	aagr	sets!R5	1				
19	dset	anat	sets!S5	1				
20	dset	afd	sets!T5	1				
21	dset	aind	sets!U5	1				
22	dset	auti	sets!V5	1				
23	dset	aser	sets!W5	1				
24	dset	aagg	sets!X5	1				
25	dset	f	sets!Y5	1				
26	dset	l	sets!Z5	1				
27	dset	ls	sets!AA5	1				
28	dset	lu	sets!AB5	1				
29	dset	k	sets!AC5	1				
30	dset	lnd	sets!AD5	1				
31	dset	h	sets!AE5	1				
32	dset	tmr	sets!AF5	1				
33	dset	ter	sets!AG5	1				
34	dset	tff	sets!AH5	1				
35	dset	g	sets!AI5	1				
36	dset	i	sets!AJ5	1				
37	dset	owatpmarg	sets!AK5	1				
38	dset	w	sets!AL5	1				
39	dset	aleon	mod_sets!A5	1				
40	dset	rleon	mod_sets!B5	1				
41	set	map_w_tmr	maps!A5	2				
42	set	map_tmr_w	maps!D5	2				

DRAFT

DRAFT

8. 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.

10.1 Sets

10.1.1 Simulation Sets

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

- `sim` simulations set
- `sim1` applied Simulations set
- `sim2` reported Simulations set
- `elst` elasticities set
- `clos` closures set

Changing the membership of these sets allows for the running of different combinations of shocks, through ‘sim1’, elasticities, through ‘elst’, and closures, through ‘clos’ without having to entry the specific choices directly in the experiment file.

Figure 10.1.1.1 Simulations Sets Worksheet Part 1

Simulation sets		Applied Simulations set		Reported Simulations set	
sim	(description)	sim1	(description)	sim2	(description)
base	base simulation	base	base simulation	base	base simulation
sim02	remove EU27 export taxes	sim02	remove EU27 export taxes	sim02	remove EU27 export taxes
sim03a	half remove EU27 Import duties	sim03a	half remove EU27 Import duties	sim03	remove EU27 Import duties
sim03	remove EU27 Import duties	sim03	remove EU27 Import duties	sim04	remove EU27 trade taxes
sim04	remove EU27 trade taxes	sim04	remove EU27 trade taxes	sim05	common EU27 Average trade taxes
sim05	common EU27 Average trade taxes	sim05	common EU27 Average trade taxes	sim07	common EU27 Average factor use taxes
sim06	common EU27 Exogenous trade taxes	sim06	common EU27 Exogenous trade taxes	sim09	EU27 harmonisation on Average tax rates
sim07a	half common EU27 Average factor use taxes	sim07a	half common EU27 Average factor use taxes		
sim07	common EU27 Average factor use taxes	sim07	common EU27 Average factor use taxes		
sim08	common EU27 Exogenous factor use taxes	sim08	common EU27 Exogenous factor use taxes		
sim09a	part EU27 harmonisation on Average tax rates	sim09a	part EU27 harmonisation on Average tax rates		
sim09	EU27 harmonisation on Average tax rates	sim09	EU27 harmonisation on Average tax rates		
sim10	EU27 harmonisation on Exogenous tax rates				

The set ‘sim2’ is included to avoid reporting the results from intermediate steps. For some simulations using some aggregations it may be necessary to use a number of intermediate steps in order to achieve a solution and the user would typically not wish to consider the results from the intermediate steps.¹² In addition when using `SeeResults` to access the results it is relatively easy to overfill the pivot cache (max 60,000 observations).

Figure 10.1.1.2 Simulations Sets Worksheet Part 2

Closures set		Elasticities set	
clos	(description)	elst	(description)
clbase	Balanced macro closure	elst01	base elasticities
clos02	IP TS long run closure full emptyt		
clos03	IP TS long run closure umemptyt		
clos04	IP TS short run closure umemptyt		

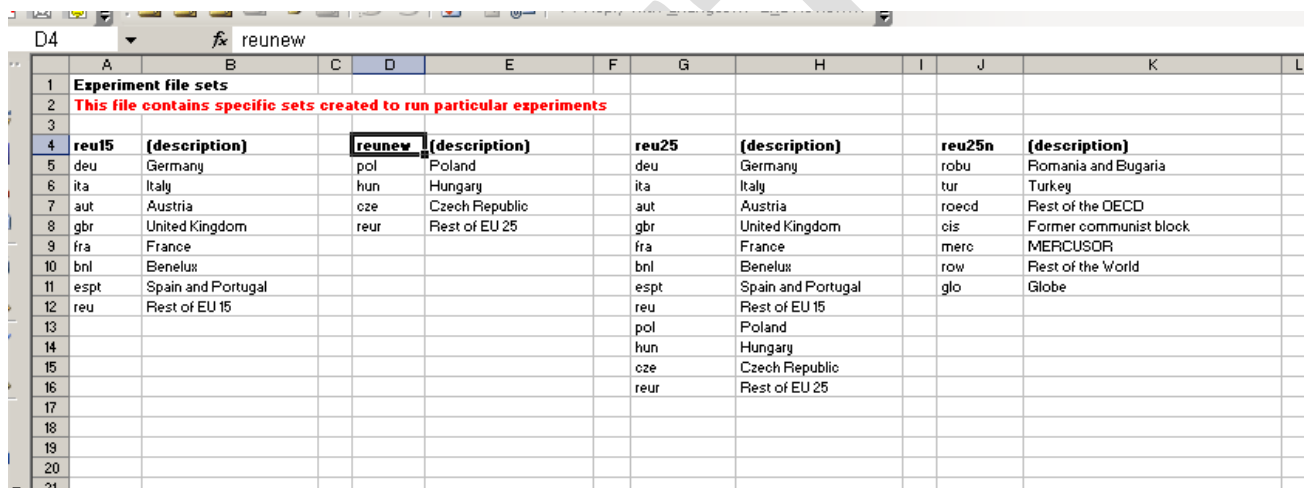
In addition to the *sim* sets closure (*clos*) and elasticity (*elst*) sets are assigned to control the other LOOPS in the experiment file – see Figure 10.1.1.2.

¹² Experience suggests that for models with less than some 70,000 variables there are relatively few instances where a solution is not achieved relatively quickly, i.e., less than about 100 iterations, even when the shock is substantial.

10.1.2 Experiment Sets

GAMS is a set based language and hence much of its power and flexibility comes from the use of sets. When defining experiments it is often extremely useful to create (sub)sets to abbreviate the coding of the shocks; the sheet ‘expsets’ exists 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.

Figure 10.1.2 Experiment Sets



	A	B	C	D	E	F	G	H	I	J	K	L
1	Experiment file sets											
2	This file contains specific sets created to run particular experiments											
3												
4	reu15	(description)		reunew	(description)		reu25	(description)		reu25n	(description)	
5	deu	Germany		pol	Poland		deu	Germany		robu	Romania and Bugaria	
6	ita	Italy		hun	Hungary		ita	Italy		tur	Turkey	
7	aut	Austria		oze	Czech Republic		aut	Austria		roecd	Rest of the DECD	
8	gbr	United Kingdom		reur	Rest of EU 25		gbr	United Kingdom		cis	Former communist block	
9	fra	France					fra	France		merc	MERCUSOR	
10	bnl	Benelux					bnl	Benelux		row	Rest of the World	
11	espt	Spain and Portugal					espt	Spain and Portugal		glo	Globe	
12	reu	Rest of EU 15					reu	Rest of EU 15				
13							pol	Poland				
14							hun	Hungary				
15							cze	Czech Republic				
16							reur	Rest of EU 25				
17												
18												
19												
20												
21												

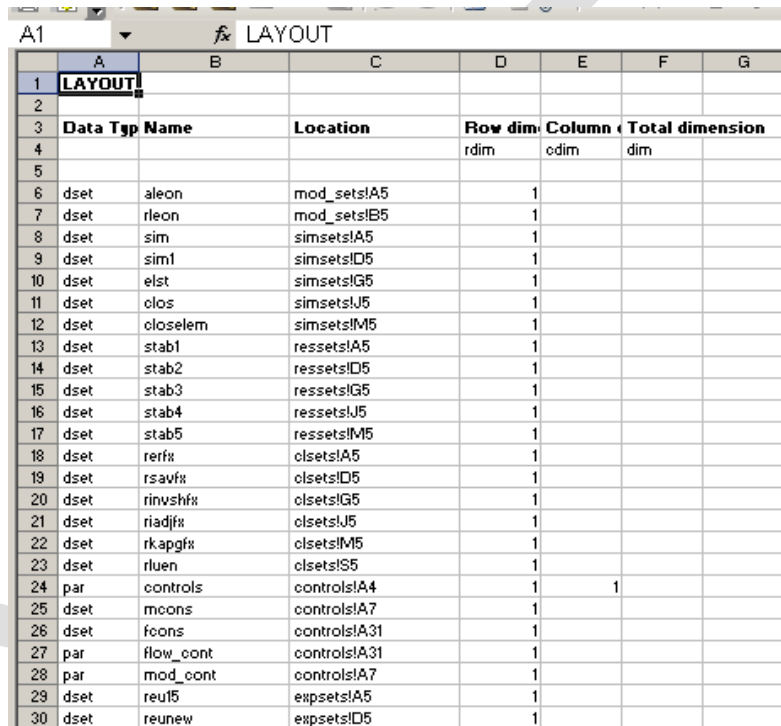
10.2 Flow Controls

The flow controls in the worksheet ‘econtrols’ 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 10.2.1 provides a screen shot of the ‘econtrols’ worksheet; as can be seen there are descriptions for each parameter .

10.4 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 10.4.1 Experiment Layout/Index Sheet



	A	B	C	D	E	F	G
1	LAYOUT						
2							
3	Data Typ	Name	Location	Row dim	Column	Total dimension	
4				rdim	cdim	dim	
5							
6	dset	aleon	mod_sets!A5	1			
7	dset	rleon	mod_sets!B5	1			
8	dset	sim	simsets!A5	1			
9	dset	sim1	simsets!D5	1			
10	dset	elst	simsets!G5	1			
11	dset	clos	simsets!J5	1			
12	dset	closelem	simsets!M5	1			
13	dset	stab1	ressets!A5	1			
14	dset	stab2	ressets!D5	1			
15	dset	stab3	ressets!G5	1			
16	dset	stab4	ressets!J5	1			
17	dset	stab5	ressets!M5	1			
18	dset	refix	clsets!A5	1			
19	dset	rsavfix	clsets!D5	1			
20	dset	rinvsfix	clsets!G5	1			
21	dset	riadfix	clsets!J5	1			
22	dset	rkappfix	clsets!M5	1			
23	dset	rlien	clsets!S5	1			
24	par	controls	controls!A4	1	1		
25	dset	mcons	controls!A7	1			
26	dset	fcons	controls!A31	1			
27	par	flow_cont	controls!A31	1			
28	par	mod_cont	controls!A7	1			
29	dset	reut5	expsets!A5	1			
30	dset	reunew	expsets!D5	1			

Ancillary Code and Experiment Files

6. Model Calibration Checks

The GLOBE model has a number of 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: All the slack variables should equal zero, or very nearly zero. Search for 'var walras', 'var kapworsys', 'var globeslack' – all should be zero. (Note: if the version of GAMS used has indexing for the list file select SolVar and the slacks are reported at the end of the list of variables.)
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: 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 close to zero. Second check the Micro SAM: search for and check DIFFSAMG2 and CNTSAMG2 – these should be zeros or close to zero. (Note: if the version of GAMS used has indexing for the list file select DISPLAY.)
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. 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 region there are a large number of different permutations for these ‘rules’ and the number of permutations increase ‘exponentially’ as the number of regions in the model increases. It is therefore important to be systematic in identifying the rules that are to apply for each region 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 GLOBE model. At the top of the file a simple ‘table’ provides a basis for summarising the chosen closure conditions, i.e.,

```
$ontext
  glbl_cl_template_1.inc
```

	CONDITION	REGIONS
FEX	- Exchange rate fixed	rerfx
	- KAPWOR fixed	rerfxn

Investment - absorption share fixed	All
Government - absorption share fixed	All
- TYH flexible - Additive form	All
- CAPGOV fixed	All
Factors - Land sector - mobile & full	All
- Capital - mobile & full	All
- Unskilled labour - mobile & full	All
- Skilled labour - mobile & full	All
Numeraire - CPI	All
\$offtext	

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

11.1 Foreign Exchange Closure

```

### FOREIGN EXCHANGE MARKET CLOSURE

* fix world numeraire in which foreign transactions are valued

ERPI.FX          = ERPI0 * numerchk ;

* globe transactions are in the numeraire currency

ER.FX("glo")     = ERPI0 * numerchk;

* globes trade balance is zero by definition of tship and the fact
* that there is only one imported commodity

* In this MODEL the exchange rate for all regions EXCEPT GLO is fixed
* and the current account balance is the equilibrating variable.

* ER.FX(rgn)      = ER0(rgn) ;
ER.FX(rerfx)     = ER0(rerfx) ;

* alternatively the external balance is fixed for non-reference
countries

```

```
* KAPWOR.FX(rgn)           = KAPWOR0(rgn) ;
KAPWOR.FX(rerfxn)         = KAPWOR0(rerfxn) ;
```

11.2 Investment-Savings Closure

```
*## INVESTMENT-SAVINGS CLOSURE
* IF aggregate investment is determined by aggregate savings
* i.e., the model is savings driven, then fix SAdj and DSHH
* Typically the actual/model savings rates are NOT fixed directly

* SAdj controls multiplicative changes in savings rates

* SAdj.FX(r)           = SAdj0(r) ;

* DSHH controls additive changes in savings rates
* NB DSHH needs to be used in conjunction with shh01

DSHH.FX(r)           = DSHH0(r) ;

* ALTERNATIVELY the model is made investment driven and the investment
* scaling factor is fixed

* IAdj.FX(r)           = IAdj0(r) ;

* OR the shares of domestic final demand of investment is fixed

INVESTSH.FX(r)       = INVESTSH0(r) ;

* OR the value of domestic investment can be fixed

INVEST.FX(r)         = INVEST0(r) ;

* Closure rule for Government savings is part of the Govt Closure
Rules
```

11.3 Government Closure Rules

```
*## GOVT CLOSURE RULES
* IF ALL tax rates are fixed
* AND Government consumption expenditure is fixed
* Then the equilibrating variable is Government Savings
```

* Tax rate scaling factors

* T*ADJ control multiplicative changes in tax rates

TEADJ.FX(r) = TEADJ0(r) ;

TMADJ.FX(r) = TMADJ0(r) ;

TSADJ.FX(r) = TSADJ0(r) ;

TXADJ.FX(r) = TXADJ0(r) ;

TYFADJ.FX(r) = TYFADJ0(r) ;

TYHADJ.FX(r) = TYHADJ0(r) ;

TFADJ.FX(r) = TFADJ0(r) ;

* DT* control additive changes in tax rates

* NB DT* needs to be used in conjunction with t*01

DTE.FX(r) = DTE0(r) ;

DTM.FX(r) = DTM0(r) ;

DTS.FX(r) = DTS0(r) ;

DTX.FX(r) = DTX0(r) ;

DTYF.FX(r) = DTYF0(r) ;

* DTYH.FX(r) = DTYH0(r) ;

DTF.FX(r) = DTF0(r) ;

* GOVERNMENT EXPENDITURE

* FIX volume

* QGDADJ.FX(r) = QGDADJ0(r) ;

* OR FIX nominal

* EG.FX(r) = EG0(r) ;

* OR FIX shares of final demand

VGDSH.FX(r) = VGDSH0(r) ;

* GOVERNMENT SAVINGS

* OR the internal balance / Govt savings can be fixed

```
KAPGOV.FX(r)          = KAPGOV0(r) ;
```

11.4 Factor Market Closure

```
*** FACTOR MARKET CLOSURE
```

```
*# Basic Factor Market Closure
```

```
*$ontext
```

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

```
*$offtext
```

```
* # Alternative Factor Market Closure
```

```
$ontext
```

```
* This code allows for controlling individual factors
```

When changing factor market closure rules be careful to count how many conditions you relax, i.e., unfix, and how many you fix.

```
TO MAKE A FACTOR ACTIVITY SPECIFIC AND FIXED
```

- i) Unfix FS.FX("factor") and WFDIST.FX("factor",a)
- ii) Fix FD.FX("factor",a) AND ONE WFDIST.FX("??",a)

```
TO ALLOW FOR AN UNEMPLOYED FACTOR
```

- i) Unfix FS.FX("factor")
- ii) Fix WF.FX("factor") AND FS.LO("factor") and FS.UP("factor")

```
TO CONTROL FACTOR USE BY ACTIVITY
```

Adapt the procedure for making a factor activity specific and fixed by specifying the activities.

NB GAMS reads the programme files from the top and hence if a condition is over written it is the last statement that determines the models behaviour.

STAGE_t: A User Guide

Hence, it is often convenient to specify a general case for all factors and/or activities and then OVERWRITE the general case with the conditions for the specific case.

\$offtext

\$ontext

The example below assumes that all factors for all regions and activities are fully employed and mobile, EXCEPT for unskilled labour in the subset of regions rluen (regions with unskilled labour supply endogenously determined). Note membership of rluen is specified in the Excel worksheet for the experiment.

\$offtext

\$ontext

```

    FS.FX(lu,rluex)           = FS0(lu,rluex) ;
    WFDIST.FX(lu,a,rluex)    = WFDIST0(lu,a,rluex) ;
    WF.LO(lu,rluex)         = -inf ;
    WF.UP(lu,rluex)         = +inf ;
*   WFDIST.FX(lu,"aotserv",rluex) = WFDIST0(lu,"aotserv",rluex) ;
*   FD.FX(lu,a,rluex)       = FDO(lu,a,rluex) ;
*   WF.FX(lu,rluex)        = WFO(lu,rluex) ;
*   FS.LO(lu,rluex)        = -inf ;
*   FS.UP(lu,rluex)        = +inf ;

*   FS.FX(lu,rluen)         = FS0(lu,rluen) ;
    WFDIST.FX(lu,a,rluen)    = WFDIST0(lu,a,rluen) ;
*   WF.LO(lu,rluen)        = -inf ;
*   WF.UP(lu,rluen)        = +inf ;
*   WFDIST.FX(lu,"aotserv",rluen) = WFDIST0(lu,"aotserv",rluen) ;
*   FD.FX(lu,a,rluen)      = FDO(lu,a,rluen) ;
    WF.FX(lu,rluen)         = WFO(lu,rluen) ;
    FS.LO(lu,rluen)         = -inf ;
    FS.UP(lu,rluen)         = +inf ;

    FS.FX(ls,r)             = FS0(ls,r) ;
    WFDIST.FX(ls,a,r)       = WFDIST0(ls,a,r) ;
    WF.LO(ls,r)             = -inf ;
    WF.UP(ls,r)             = +inf ;
*   WFDIST.FX(ls,"aotserv",r) = WFDIST0(ls,"aotserv",r) ;

```

STAGE_t: A User Guide

```

* FD.FX(ls, a, r) = FD0(ls, a, r) ;
* WF.FX(ls, r) = WF0(ls, r) ;
* FS.LO(ls, r) = -inf ;
* FS.UP(ls, r) = +inf ;

FS.FX(k, r) = FS0(k, r) ;
WFDIST.FX(k, a, r) = WFDIST0(k, a, r) ;
WF.LO(k, r) = -inf ;
WF.UP(k, r) = +inf ;
* WFDIST.FX(k, "aotserv", r) = WFDIST0(k, "aotserv", r) ;
* FD.FX(k, a, r) = FD0(k, a, r) ;
* WF.FX(k, r) = WF0(k, r) ;
* FS.LO(k, r) = -inf ;
* FS.UP(k, r) = +inf ;

FS.FX(lnd, r) = FS0(lnd, r) ;
WFDIST.FX(lnd, a, r) = WFDIST0(lnd, a, r) ;
WF.LO(lnd, r) = -inf ;
WF.UP(lnd, r) = +inf ;
* WFDIST.FX(lnd, "aagr", r) = WFDIST0(lnd, "aagr", r) ;
* FD.FX(lnd, a, r) = FD0(lnd, a, r) ;
* WF.FX(lnd, r) = WF0(lnd, r) ;
* FS.LO(lnd, r) = -inf ;
* FS.UP(lnd, r) = +inf ;

FS.FX("NatRes", r) = FS0("NatRes", r) ;
WFDIST.FX("NatRes", a, r) = WFDIST0("NatRes", a, r) ;
WF.LO("NatRes", r) = -inf ;
WF.UP("NatRes", r) = +inf ;
* WFDIST.FX("NatRes", "APUBL", r) = WFDIST0("NatRes", "APUBL", r) ;
* FD.FX("NatRes", a, r) = FD0("NatRes", a, r) ;
* WF.FX("NatRes", r) = WF0("NatRes", r) ;
* FS.LO("NatRes", r) = -inf ;
* FS.UP("NatRes", r) = +inf ;

```

\$offtext

11.5 Technology Variables

```
* # Technology Closures for Factor Market
```

STAGE_t: A User Guide

* NB SHIFT factors are fixed indirectly through the adjustment variables

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

* To FIX ADX fix BOTH ADXADJ and DADX

* NB DADX needs to be used in conjunction with adx01

ADXADJ.FX(r) = ADXADJ0(r) ;

DADX.FX(r) = DADX0(r) ;

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

* To FIX ADVA fix BOTH ADVAADJ and DADVA

* NB DADVA needs to be used in conjunction with adva01

ADVAADJ.FX(r) = ADVAADJ0(r) ;

DADVA.FX(r) = DADVA0(r) ;

* Technology for factor activity and region specific factor efficiency

ADFD.FX(f, a, r) = ADFD0(f, a, r) ;

11.6 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 ;

10. Tax Rate Shocks

In order to assist the user the template experiment file comes complete with declaration statements for tax rate shocks, i.e.,

```

Parameter
* Standard parameters for tax instrument shocks
TESIM(c,w,r,sim)      Export taxes on exported comm'y c from r to w
TMSIM(w,c,r,sim)      Tariff rates on imported comm'y c from w by r
TSSIM(c,r,sim)        Sales tax rate
TXSIM(a,r,sim)         Indirect tax rate
TYFSIM(f,r,sim)       Direct tax rate on factor income
TYHSIM(h,r,sim)       Direct tax rate on households
TFSIM(f,a,r,sim)      Tax rate on factor use

```

together with statements in the LOOP that set the tax rates equal to those in base case.

The user is free to choose any name for the parameter that carries information used to shock a parameter; as a matter of good practice it is advisable to choose a name that links to the parameter being shocked, e.g., *TMSIM* refers to the variable *TM*.

11. Technology Shocks

In order to assist the user the template experiment file comes complete with declaration statements for technology shocks, i.e.,

```
* Standard parameters for efficiency rate shocks
ADXSIM(a,r,sim)      Shift parameter for CES prodn fns for QX in r
ADVASIM(a,r,sim)    Shift parameter for CES prodn fns for QVA
ADFDSIM(f,a,r,sim)  Shift parameter for factor & activity efficiency
```

The user is free to choose any name for the parameter that carries information used to shock a parameter; as a matter of good practice it is advisable to choose a name that links to the parameter being shocked, e.g., *ADVASIM* refers to the variable *ADVA*.

Annexes

A1. Aggregating a SAMs

The dimensions of a SAM may be such as to require, for practical purposes, that the database is aggregated before being used to calibrate a CGE model. There are two main reasons for this:

1. If the database is not aggregated the resultant model would be too large to
 - a. solve within realistic limits of computer capacity,¹³ OR
 - b. yield results that would be susceptible to practical analyses and interpretation.
2. It is argued that models should be focused upon specific issues rather than being general and that part of the process of focus depends upon the identification of an appropriate aggregation for the purposes at hand.

An aggregation of a SAM requires that the database is (simultaneously) aggregated in two dimensions, i.e., the income and expenditure accounts are simultaneously aggregated to reduce the number commodities/activities, factors, trade partners and associated transactions. GAMS is an extremely efficient medium for implementing the calculations required to aggregate a large SAM, but setting up the sets and (set) mappings to control the aggregation is potentially time consuming and subject to errors. One method for reducing the time costs involved in setting up a new aggregation, and simultaneously reducing the time required to configure the sets and other data needed to implement a version¹⁴ of STAGE with a new aggregation, is to use the SAMgator software (PROVIDE, 2004).

SAMgator is a Visual Basic for Applications (VBA) programme implemented using Microsoft (MS) Excel. The user declares, describes and defines the new (aggregated) sectors (commodities and activities), factors and regions in an Excel template, identifies the source

¹³ Developments in solver technologies have allowed solutions to be derived for ever larger models so this reason is increasingly not relevant.

¹⁴ The term “version of STAGE” is used to refer to an implementation of the STAGE model with a specific aggregation of the database. A “version of STAGE” is used to refer to an implementation of a version of the STAGE model that contains either ‘limited’ or no variations in the behavioural relationships of the STAGE model. More substantial changes in the STAGE model are identified by ‘extending’ the STAGE name, e.g., GLOBE-AGR.

and destination data files and then runs GAMS remotely from within the Excel programme¹⁵. The SAMgator programme generates the set and mapping files, checks to ensure that all the mappings are ‘legal’ and then generates the GAMS programme file.

3.1 The Mechanics of SAMgator

At the heart of SAMgator is a single (GAMS) equation that aggregates the SAM database in two dimensions (see below).

$$\begin{aligned} \text{NEWSAM}(sp, spp) &= \text{SUM}((ss, ssp) \\ &\quad \$(\text{MAPSAMAG}(sp, ss) \\ &\quad \$\text{MAPSAMAG}(spp, ssp)), \\ &\quad \text{SAM}(ss, ssp)) ; \end{aligned}$$

The parameter $\text{SAM}(ss, ssp)$ contains the disaggregated database where ss ¹⁶ is the set that defines the row and columns labels for the SAM. The parameter $\text{NEWSAM}(sp, spp)$ contains the aggregated database where s is the set that defines the row and columns labels for the aggregated (NEW)SAM. The set $\text{MAPSAMAG}(sp, ss)$ defines the members of ss that aggregated into sp by the rows of $\text{SAM}(ss, ssp)$, while the set $\text{MAPSAMAG}(spp, ssp)$ defines the members of ssp that aggregated into ssp by the columns of $\text{SAM}(ss, ssp)$. (Note that there is only one mapping set MAPSAMAG with its implementation being solely driven by the sets ss and s and their aliases.).

3.1 Using SAMgator

The user interface to SAMgator consists of three worksheets ‘Control’, ‘MappingConfig’ and ‘Sets’. A user only needs to access these three worksheets; although more experienced users may wish to use other features of SAMgator the discussion here is limited to these three worksheets.

3.1.1 Control

The ‘Control’ worksheet is used to identify the location of GAMS.EXE, input and output file names, check for errors, write output and to run the aggregation programme. A screen shot of the worksheet is given in Figure 3.1.1.1.

¹⁵ The GAMS code at the heart of SAMgator is report in the Appendices.

¹⁶ The set ssp is an alias for ss .

In order to run GAMS from within MS Excel it is necessary for Excel to know the location of the GAMS.EXE file; this information is provided in the ‘GAMS.EXE location’ box.¹⁷ In order to test that Excel can implement a GAMS programme first click on the ‘Pause after execution’ box – so it has a tick in the box as in Figure 3.1.1.1 – and then click on the ‘Test GAMS’ button. A DOS window will appear (Figure 3.1.1.2); if this indicates a normal completion this is confirmation that Excel can trigger GAMS correctly and the user can press any key to continue.

Figure 3.1.1.1 Control Worksheet for SAMgator

The screenshot shows an Excel spreadsheet with the following content:

- Row 2:** SAMgator - A SAM aggregation tool
- Row 3:** Copyright © The PROVIDE project (<http://www.elsenburg.com/provide>). See [readme.html](#) for information.
- Row 4:** This version has been adapted to aggregate additional data via an include file
- Options:**
 - GAMS.EXE location: C:\GAMS\GAMS.EXE
 - Pause after execution:
 - Test GAMS button
- File Name Fields:**
 - .GDX file name: Input (outsam.gdx), Output (OECD SAMv2.gdx)
 - Parameter name: (outsam), (SAM)
 - SAM set name: (ss), (sac)
- Aggregation program (.gms):** (agg PSAM OECDv2.gms)
- Set definition include file (.inc):** (ModelSetDefs OECDv2.inc)
- Remove first character from non-aggregated element names:
- Commands:**
 - Load data, Check for errors, Write sets .inc
 - Write .gms program, Write Sets sheet, Run aggregation

The user should now define the input file, which is required to be a GAMS Data Exchange (GDX) file; in this illustration the file is called 'outsam.gdx'.¹⁸ Then name the

¹⁷ With some versions of GAMS spaces in the paths can lead to problems. Note how the screen shot indicates that GAMS.exe is stored in a GAMS directory on the C drive and not in the ‘Program Files’ directory that is the default destination when installing GAMS.

¹⁸ GAMS has changed the formatting of GDX files over recent years. The changes are backward compatible but not forward compatible. This can lead to problem if the user wishes to access a more recent version of a GDX file that is not consistent with the version of GAMS being used; this should be rare. If it does happen there are two most obvious solutions; (1) recompile the input database using the more recent version of GAMS or (2) recompile the input database using the options in GAMS to write out GDX files in the earlier formats.

output file; in this illustration 'OECD SAMv2.gdx'. In both instances it is necessary for the user to identify the parameter within the GDX file that is the source of the data and the destination and the set labels used in the source and destination parameters. And finally the user should provide a name for the GAMS programme file; in this illustration 'agg_PSAM_OECDv2.gms' and the name for the include file that will be generated to contain the set definitions.

Then select each of the Command buttons in turn – do so for each row in turn starting each time with the lefthand button.

Figure 3.1.1.2 Test GAMS

```

C:\WINDOWS\system32\cmd.exe
****
**** GAMS Development Corporation
**** 1217 Potomac Street, NW
**** Washington, DC 20007, USA
**** 202-342-0180, 202-342-0181 fax
**** support@gams.com, www.gams.com
****
**** BldDate : Nov 21, 2006
**** SysDir : C:\GAMS\
**** BldStamp: vis 22.3 (Nov 27, 2006): Tue 21 Nov 03:03:17 PM EST 2006
**** License : C:\GAMS\gamslice.txt
****
**** Department of Economics                S060526:0542AL-WIN
**** University of Sheffield
**** DC109 01COM5PT                          0000
**** License for teaching and research at degree granting institutions
****
**** Status: Normal completion
--- Job ? Stop 02/05/07 15:22:52 elapsed 0:00:00.015

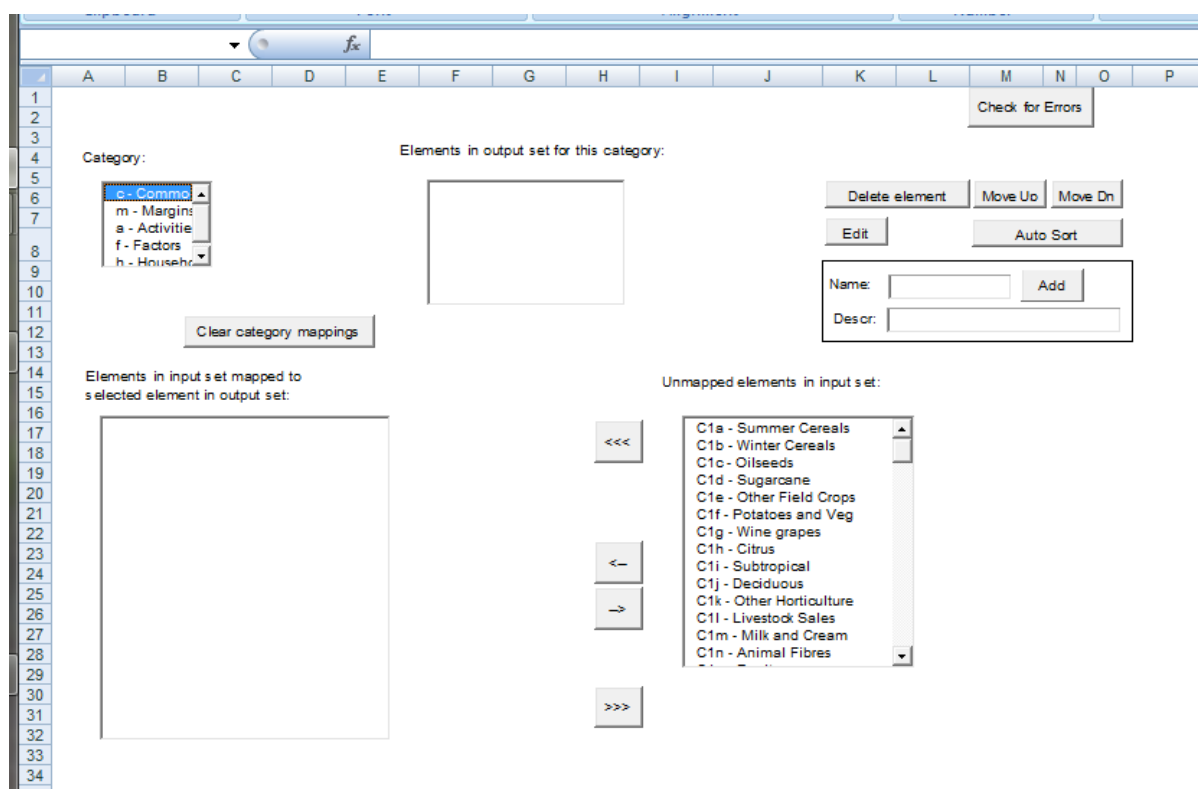
C:\Research\Miscellaneous Research\EU Food Industry\AnEFIS\IPTS_pc1_data>pause
Press any key to continue . . .

```

3.1.2 MappingConfig

The role of the 'MappingConfig' worksheet is to generate the sets for the aggregated SAM and set up the mappings, which will control the aggregation. The process has a number of safeguards built in to ensure there the mappings are legitimate and that there are no errors. Although users may choose to adapt a previous aggregation the description given here assumes that the user starts with a version of SAMgator that contains no aggregation specific set or mapping data.

Figure 3.1.2.1 illustrates how the 'MappingConfig' worksheet might look when there is no aggregation specific set or mapping data. The user needs to create sets and mappings for the commodities/activities, factors and households.

Figure 3.1.2.1 Mapping and Configuration Worksheet 1

The steps undertaken to declare and define aggregate sets and to generate the mapping file are as follows.

1. Select a category: choose commodities/activities, factors or households in the 'Category' box (top left).
2. Declare (and describe) an aggregate set element: type the name in the 'Name' box and below it a description in the 'Descr' box and then click the 'Add' button. This element will be a member of an output set for which each element will appear with its description in the 'Elements in output set for this category' box.
3. Select elements in the input set that are to be mapped to the selected member of the output set: select an element in the 'Elements in output set for this category' box and then select those elements that are to be mapped to that output set element from the 'Unmapped elements of input set' box (bottom right) by clicking on each in turn (NB the SHIFT and Ctrl click options do not operate). Click on the <- button to define the mapping. Note how the selected element of the input set are moved to the 'Elements in input set mapped to selected element in output set' box (bottom left).

- The <- button moves selected elements in the unmapped element box to the mapped elements box; the -> button moves selected elements in the mapped element box to the unmapped elements box; the <<< button moves all elements in the unmapped element box to the mapped elements box; >>> button moves all elements in the mapped element box to the unmapped elements box.
 - Figure 3.1.2.1 illustrates that for the category ‘c – Commodities’.
4. Continue adding elements to the output set in each category until all the elements in the input set are mapped to elements in the output set. Note how each element in the input set can only be mapped to one element in the output set – thereby avoiding ‘double counting’. The error checking also checks to avoid duplicates in the output set.
 5. Ordering of the output set: as each new element of an output set is declared it is added to the end of the list of members if the output set for that category and this defines the set ordering that will be used by GAMS. The order of elements within an output set can be altered by selecting elements in the ‘Elements in output set for this category’ box and using the ‘Move Up’ and/or ‘Move Down’ buttons.¹⁹
 6. Deleting an output set element: select the element to delete in ‘Elements in output set for this category’ box and click on the ‘Delete element’ button.
 7. Select another category and complete steps 2 to 7 as appropriate until output sets and mapping sets for all three categories have been generated.

3.1.3 Sets

The worksheet sets provides a full listing of the set names and descriptions required by the STAGE model. These can be copied and pasted into the worksheet that contains set information for STAGE.

¹⁹ Note that GAMS lists set members in the order that they were declared in the programme. If the same name, say ‘abc’, is used in 2 different sets this can create ordering problems for the user: assume that the order wanted for reports is the order of members in the second set declared, but because ‘abc’ is in the first set declared it will be the first member of the second set. Use names for members of sets that are unique to the sets to which they belong.

3.2 Aggregation Guidelines

Some general guidelines for aggregations are given below. In the main these are common sense.

The aggregation of accounts in a SAM is overwhelmingly a matter of the context for which the aggregation is being compiled; consequently the notes below are general and are not a replacement for economic reasoning and/or common sense.

- Aggregations across the categories (commodities, activities, factors, regions) of accounts are NOT economically appropriate even though they are mathematically possible. SAMgator prohibits such aggregations, and where the aggregation is conducted using a different method the checks in the STAGE model code will often identify such inappropriate aggregations²⁰.
- Aggregations within categories should typically avoid mappings that are difficult to rationalise, e.g., aggregating unskilled labor with capital and keeping skilled labour separate, aggregating an agricultural commodity with a heavy manufacturing commodity.

3.3 GAMS Data Exchange (GDX) Database

The GDX file produced by the SAMgator programme contains an aggregated SAM. The user can then choose that the model accesses the SAM data directly from GDX or the user can extract the SAM from GDX, the easiest option being to use the CUBE and EXPORT facilities in GDXViewer, and add the SAM data to the model Excel workbook (see below).

²⁰ The checks in the model code are not designed to identify such errors but they do identify entries in cells that are inappropriate and in such circumstances will cause the model to abort. However the error messages are not geared to the identification of problems associated with inappropriate aggregations.

References

- Francois, J.F. and Reinert, K.A. (ed) (1997). *Applied Methods for Trade Policy Analysis: A Handbook*. Cambridge: Cambridge University Press.
- King, B.B., (1985). 'What is a SAM?', in Pyatt, G. and Round, J.I. (ed), *Social Accounting Matrices: A Basis for Planning*. Washington: World Bank.
- McDonald, S. and Thierfelder, K., (2004). 'Deriving a Global Social Accounting Matrix from GTAP version 5 Data', *Global Trade Analysis Project Technical Paper 23*, 2004 Center for Global Trade Analysis: Purdue University.
- McDonald (2007). 'Deriving Reduced Form Global Social Accounting Matrices from GTAP Data'. *Mimeo*.
- McDonald. S. and Sonmez, Y., (2004). 'Augmenting the GTAP Database with Data on Inter-Regional Transactions', *Sheffield Economics Research Paper 2004:009*. The University of Sheffield.
- McDonald, S., (2007b). 'GLOBE-EN: GLOBE: Global CGE Energy Model', *mimeo* (with K Thierfelder).
- McDonald, S., (2007c). 'GLOBE-IC: Global CGE Model with Imperfect Competition', *mimeo*.
- PROVIDE (2004). 'SeeResults: A Spreadsheet Application for the Analysis of CGE Model Results', *PROVIDE Project Technical Paper 2004:1*. PROVIDE Project: Elsenburg.
- Pyatt, G. and Round, J.I. (1985) (ed), *Social Accounting Matrices: A Basis for Planning*. Washington: World Bank.
- Pyatt, G., (1991). 'Fundamentals of Social Accounting', *Economic Systems Research*, Vol 3, pp 315-341.
- Rutherford, T., (1998). GTAP6inGAMS, <http://www.mpsge.org/gtap6/>.
- Sadoulet, E. and de Janvry, A., (1995). *Quantitative Development Policy Analysis*. Baltimore: John Hopkins University Press.
- UN, (1993). *System of National Accounts 1993*. New York: UN.