# STAGE: A User Guide

Scott McDonald
Sept 2020

**Abstract**

This paper is a user's guide to the STAGE CGE model. The User Guide refers to an implementation of the model that uses GAMSIDE as the text editor[1], 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. 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.**



Email: scott@cgemod.org.uk ; jrs.mcdonald@gmail.com
Url: www.cgemod.org.uk

[1] A version will be developed for GAMS Studio when GAMS Studio has the full functionality of GAMSIDE.

# Table of Contents

# 1. Introduction

STAGE 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. Since STAGE is a 'template' model that users are encouraged to adapt and develop, this User Guide may not encompass all dimensions of all variants of the STAGE_CC model.

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.

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.

The rest of this Guide is organised as follows.

# 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 contains information in the active sub matrices and conforms to the rules of a SAM.[2] 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. (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 government expenditures. These adjustments do not change the information content of the SAM, but they do simplify the modeling process. However, they do

[2] 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.[3]

**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

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

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 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. 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 (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 for use in STAGE;
2. removal minor differences the row and column totals of the SAM (using RAS); and
3. estimate a SAM from incomplete and imperfect data (using entropy).

### *2.1.3 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 $(f*a)$[4]; and
2. FACTINS: this satellite account records the (natural) quantities of factors owned by each representative household group (RHG), i.e., the dimensions are $(h*f)$[5].

---

4 *f* is the number of natural factors and *a* the number of activities.

5 *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 other satellite accounts currently available for use in the model are optional and require the user to 'tell' the model that these data are available. These are

1. POP by RHG: these data can be recorded in adult equivalents or simple numbers;
2. EN_USE; recorded quantity units according to the nature of the energy input, e.g., oil equivalents for fossil fuels and kilowatt hours for electricity, the dimensions are (*c***h*+*ins***c_en*)[6]; and
3. EMIT: emissions, e.g., CO2, NO2, etc., associated with each energy input and its use by different consumers, the dimensions are (*c***h*+*ins***c_en***emiss*)[7].

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

### *2.1.4 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*,**);

---

6 *ins* is the number of domestic institutions (RHG, government and investment) and *a* the number of commodities, and *c_en* the number of different types of energy, e.g., coal, oil, gas, electricity, biofuels etc.

7 *emiss* is the type of emission, e.g., CO2, NO2.

8 Cobb-Douglas functions have elasticities equal to ONE.

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.

### 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.

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

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: identifies factors used in the 2nd level of the production system
4. nest_fagg: identifies natural and aggregate factors used in the 3rd and lower levels of the production system
5. nest_agg: provides nesting structure data for calibrating the production system
6. fact_comm: identifies commodities included in the value added 'arm' of the production system
7. stab_sets: sets used to report structural information about the model and data
8. res_sets: sets used to report results (these sets are used to report on model calibration and experiments)
9. controls: parameters used to control/condition aspects of the model
10. SAM: the SAM transactions data (can be stored separately in GDX)
11. factuse: factor use by activities (can be stored separately in GDX)
12. factins: factor ownership by institutions (can be stored separately in GDX)
13. pop: population data by RHG
14. EN_USE: energy use by consumers (activities and institutions)

---

[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 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 generates 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).

15. EMIT: emissions data by energy source, emissions type and consumer
16. comelast: elasticities relating to commodities
17. actelast: elasticities relating to activities
18. elastf: elasticities used in level 3 and below of the production system
19. frischelast: Frisch parameters for the LES
20. incelast: income elasticities of demand for the LES
21. comelasth: CES elasticities for nested utility function

## 3.1 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 3.1.1 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 | cc | Sets!G6 | 1 | | | | |
| 9 | dset | c | Sets!H6 | 1 | | | | |
| 10 | dset | cces | Sets!I6 | 1 | | | | |
| 11 | dset | cag | Sets!J6 | 1 | | | | |
| 12 | dset | cagr | Sets!K6 | 1 | | | | |
| 13 | dset | cnat | Sets!L6 | 1 | | | | |
| 14 | dset | cfd | Sets!M6 | 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.

## 3.2 Model Sets

### *3.2.1 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 ar illustrated in shown in Figure 3.1.1.1.

**Figure 3.2.1.1 Sets Worksheet Part 1**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | Subsets of SAC are declared to the RIGHT | | | | | | |
| 3 | Global Set | | ASAM Set | | | | |
| 4 | GAMS Name | Description | GAMS Name | Description | Commodities - Natural and aggregates | Natural Commodities | Natural Commoditi CES Utili |
| 5 | sac | | ss | | cc | c | cces |
| 6 | c_whea | Wheat | COMMDTY | Commodity Accounts | c_whea | c_whea | |
| 7 | c_maiz | Maize | MARG | Margin Accounts | c_maiz | c_maiz | |
| 8 | c_rice | Rice | ACTIVITY | Activity Accounts | c_rice | c_rice | |
| 9 | c_ogrn | Other cereals | VALUAD | Value Added Accounts | c_ogrn | c_ogrn | |
| 10 | c_oils | Pulses & oil seeds | HHOLDS | Household Accounts | c_oils | c_oils | |
| 11 | c_cott | Cotton | ENTP | Enterprise Accounts | c_cott | c_cott | |
| 12 | c_sugr | Sugarcane | GOVTN | Government Accounts | c_sugr | c_sugr | |
| 13 | c_root | Roots & tubers | KAPITAL | Capital Accounts | c_root | c_root | |
| 14 | c_vege | Vegetables | WORLD | Rest of World Accounts | c_vege | c_vege | |
| 15 | c_frui | Fruits | TOTALS | Income or Expenditure Totals | c_frui | c_frui | |
| 16 | c_coff | Coffee | | | c_coff | c_coff | |
| 17 | c_tea | Tea | | | c_tea | c_tea | |
| 18 | c_ocrp | Others crops | | | c_ocrp | c_ocrp | |
| 19 | c_goat | Sheep, goat and lamb for slaughter | | | c_goat | c_goat | |
| 20 | c_beef | Beef | | | c_beef | c_beef | |
| 21 | c_poul | Poultry | | | c_poul | c_poul | |
| 22 | c_oliv | Other livestock | | | c_oliv | c_oliv | |
| 23 | c_fore | Forestry | | | c_fore | c_fore | |
| 24 | c_fish | Fishing | | | c_fish | c_fish | |
| 25 | c_mine | Mining | | | c_mine | c_mine | |
| 26 | c_gmil | Grain milling | | | c_gmil | c_gmil | |
| 27 | c_meat | Meat & dairy | | | c_meat | c_meat | |
| 28 | c_dair | Dairy | | | c_dair | c_dair | |
| 29 | c_bake | Sugar & bakery & confectionary | | | c_bake | c_bake | |
| 30 | c_omfd | Other manufactured food | | | c_omfd | c_omfd | |
| 31 | c_bevt | Beverages & tobacco | | | c_bevt | c_bevt | |

The model uses a substantial number of subsets of *sac*; some of these are shown in Figure 3.2.1.2. 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: copy and paste 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
- *cf(c)* commodity factors
- *cfn(c)* NOT commodity factors
- *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
- a_h(sac) Activities and households

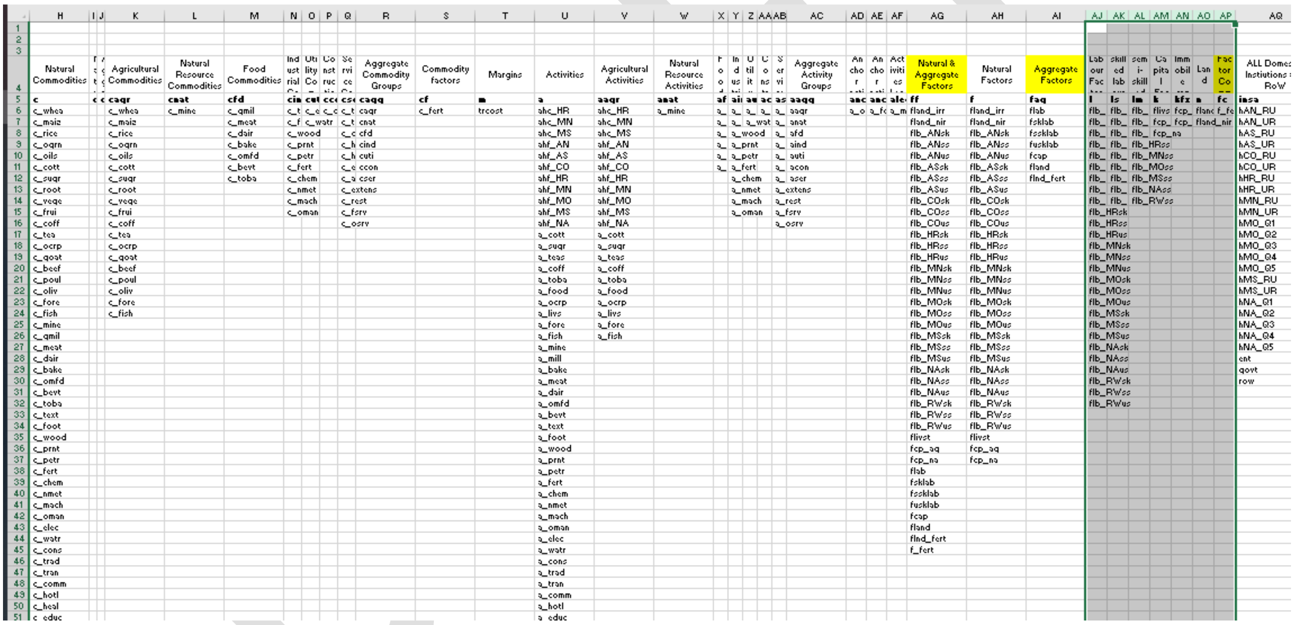
- cfa(c,a) commodity factors used by activity a
- cfan(c,a) NOT commodity factors used by activity a
- anch(a) Anchor activity for fixing 1 WFDIST in various factor closures
- anchN(a) Anchor activity for fixing 1 WFDIST in land factor closures
- aleon(a) Activities with Leontief prodn function at Level 1
- ff(sac) factors factor commodities and aggregates
- f(ff) natural factor accounts
- fag(ff) aggregate factors
- fc(ff) factor commodities
- ffc(ff) natural factors and factor commodities
- fcn(ff) natural factors and aggregates
- 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
- kfx(k) Immobile capital factors
- kfxn(k) Mobile 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 3.1.1.2  Sets Worksheet Part 2**



### *3.1.2  Maps Worksheet*

The model uses several mapping sets that allow the matching of data that are recorded using different but related labels. The generation of the required mapping sets has been simplified in GAMS (see below) so this worksheet is only retained if users need specific mapping sets that are not assigned elsewhere. A stylised version of the maps worksheet is illustrated in Figure 3.1.2.1.

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

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*.

**Figure 3.1.2.1** **Maps Worksheet** (illustration only)

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | **Maps** | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | **map_f_tff(f,tff)** | | | **map_tff_f(tff,f)** | | |
| 5 | fland_irr | tfland_irr | | | fland_irr | tfland_irr |
| 6 | fland_nir | tfland_nir | | | fland_nir | tfland_nir |
| 7 | flb_ANsk | tflb_ANsk | | | flb_ANsk | tflb_ANsk |
| 8 | flb_ANss | tflb_ANss | | | flb_ANss | tflb_ANss |
| 9 | flb_ANus | tflb_ANus | | | flb_ANus | tflb_ANus |
| 10 | flb_ASsk | tflb_ASsk | | | flb_ASsk | tflb_ASsk |
| 11 | flb_ASss | tflb_ASss | | | flb_ASss | tflb_ASss |
| 12 | flb_ASus | tflb_ASus | | | flb_ASus | tflb_ASus |

GAMS has introduced syntax to allow creation of mapping sets from CORRECTLY ordered sets. The mapping set `map_set1_set2(`*`set1`*`,`*`set2`*`)` can be assigned dynamically in GAMS using

```
Sets
map_set1_set2(set1,set2)   set 2 to set 1/
set1#set2
/;
```

However, it is important to note that this syntax only produces the correct output if `set1` and `set2` are in matching orders, i.e., the $n^{th}$ element in `set1` is paired with the $n^{th}$ element of `set2`.

### *3.1.3 nest_va, nest_fagg, nest_agg and fact_comm worksheets*

These four worksheets detail the sets and mappings used to control the production system. The sets used are *ff*, set of all natural (*f*) and aggregate factors (*fag*), `map_va_ff(ff,a)`, which identifies the arguments in Value added level nest, `map_fagg_ff(ff,ff,a)`, which identifies the arguments in nests below the value added level in respect to which natural factors (*f*) and aggregate factors (*fag*) enter each level and the natural factors and aggregates that enter into the production of each aggregate factor. The set `map_agg(ff,ff,a)` is developed from `map_va_ff(ff,a)` and `map_fagg_ff(ff,ff,a)` and identifies all the natural factors that enter into the production of each aggregate, i.e., each aggregate factor is defined solely in terms of natural factors by resolving the any aggregate factors at a lower level in the production system.

This method for setting up a nested production system may appear complicated, and it is prone to user error, but the advantage is that all levels 3 and below of a nested production system can be collapsed into two equation blocks: a production function and its associated first-order conditions.

The model also provides the facility to include commodities in the value added ‘arm’ of the production system; these are termed commodity_factors (*cf*) when considered from the commodity side of the model and factor_commodities (*fc*) when considered from the commodity side of the model. This facility is necessary when it is deemed necessary to allow for the demand for commodities to be sensitive to their purchaser prices and/or when they are substitutes for natural factors, e.g., fertilisers and phytosanitary chemicals can be regarded as substitutes for land the substitutions maybe price responsive. The other category of commodities that are traditionally included in the value added ‘arm’ of the production system are energy commodities, e.g., fossil fuels such as coal, oil and gas and electricity.

Figure 3.1.3.1a illustrates a production system for agriculture and Figure 3.1.3.1b illustrates a production system for agriculture, while Table 3.1.3.1 reports the sets used to control the production system.

**Figure 3.1.3.1a Production System for Agriculture**

**Figure 3.1.3.1a Production System for Services**

**Table 3.1.3.1 Sets for Production Systems for Agriculture and Services**

| Factors | | Activities | | | |
|---|---|---|---|---|---|
| | | map_va_ff(ff,a) | | | |
| | | a_agric | a_mine | a_manu | a_serv |
| | flnd_fer | 1 | 1 | 1 | 1 |
| | flab | 1 | 1 | 1 | 1 |
| | fcap_en | 1 | 1 | 1 | 1 |
| | | map_fagg_ff(ff,ff,a) | | | |
| | | a_agric | a_mine | a_manu | a_serv |
| flnd_fert | fland | 1 | 0 | 0 | 0 |
| flnd_fert | f_fert | 1 | 0 | 0 | 0 |
| flab | fskill | 1 | 1 | 1 | 1 |
| flab | fusklab | 1 | 1 | 1 | 1 |
| fskill | fsklab | 1 | 1 | 1 | 1 |
| fskill | fssklab | 1 | 1 | 1 | 1 |
| fcap_en | fcap | 1 | 1 | 1 | 1 |
| fcap_en | fenerg | 1 | 1 | 1 | 1 |
| fenerg | f_foss | 1 | 1 | 1 | 0 |
| fenerg | f_elect | 1 | 1 | 1 | 1 |

| Factors | | Activities | | | |
|---|---|---|---|---|---|
| | | map_agg(ff,ff,a) | | | |
| | | a_agric | a_mine | a_manu | a_serv |
| flnd_fert | fland | 1 | 0 | 0 | 0 |
| flnd_fert | f_fert | 1 | 0 | 0 | 0 |
| flab | fusklab | 1 | 1 | 1 | 1 |
| flab | fssklab | 1 | 1 | 1 | 1 |
| fskill | fsklab | 1 | 1 | 1 | 1 |
| fskill | fssklab | 1 | 1 | 1 | 1 |
| fcap_en | fcap | 1 | 1 | 1 | 1 |
| fusklab | f_foss | 1 | 1 | 1 | 0 |
| fusklab | f_elect | 1 | 1 | 1 | 1 |
| fenerg | f_foss | 1 | 1 | 1 | 0 |
| fenerg | f_elect | 1 | 1 | 1 | 1 |

An examination of these figures and table is that there are three factor_commodities (fossil fuels (*f_foss*), electricity (*f_elect*) and fertilisers (*f_fert*)) in the value added nest. Three natural types of labour (skilled (*fsklab*) and semi-skilled (*fssklab*) aggregated into *fskill*, and unskilled (*fusklab*) that combine to form aggregate labour (flab)); one type of capital that combines with aggregate energy that is formed from fossil fuels and electricity, and one type of land (*fland*) that is aggregate with fertilisers.

Note also that the nesting structure can be different for each activity; this is illustrated by the figures illustration the production systems for agriculture and services.

### *3.1.4 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 uses chooses to extend the files that generate the structural tables.

## Figure 3.1.4.1 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 valu |
| 9 | stQINVTOT | investment consumption | | stGOVTcomDEM | government domestic demand by total commodity demand | | stVA | shares of valu |
| 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 sub |
| 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 | | | | | | |

### *3.1.5 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_*_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 3.1.5.1 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 | Tariff 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 |

## 3.2 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.[13]

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

13 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.

through the mod_control parameters that are described in section 7.3. This section describes the format used to record elasticity data in Excel.

### *3.2.1 Commodity Elasticities*

Import substitution elasticities are needed for both levels of the Armington functions. These are recorded in two worksheets, 'comelastm' and 'comelastrm', 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.[14] 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**.

**Figure 3.2.1.1 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 | Wheat | c_whea | 1.75 | 2 | 0 | 4 |
| 7 | Maize | c_maiz | 1.75 | 2 | 0 | 4 |
| 8 | Rice | c_rice | 1.75 | 2 | 0 | 4 |
| 9 | Other cereals | c_ogrn | 1.75 | 2 | 0 | 4 |
| 10 | Pulses & oil seeds | c_oils | 1.75 | 2 | 0 | 4 |
| 11 | Cotton | c_cott | 1.75 | 2 | 0 | 4 |
| 12 | Sugarcane | c_sugr | 1.75 | 2 | 0 | 4 |
| 13 | Roots & tubers | c_root | 1.75 | 2 | 0 | 4 |
| 14 | Vegetables | c_vege | 1.75 | 2 | 0 | 4 |
| 15 | Fruits | c_frui | 1.75 | 2 | 0 | 4 |

### *3.2.2 Activity 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

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

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.[15] 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**.

**Figure 3.2.2.1 Activity Elasticities**

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | **Model Specific Activity Elasticities** | | | | | |
| 2 | Elasticities | | | | | |
| 3 | | | | | | |
| 4 | ELASTX | ELASTX | ELASTX | | | |
| 5 | | | sigmax | sigmava | omegaout | |
| 6 | ahc_HR | ahc_HR | 0.9 | 1.5 | 0 | |
| 7 | ahc_MN | ahc_MN | 0.5 | 0.8 | 0 | |
| 8 | ahc_MS | ahc_MS | 0.9 | 1.6 | 0 | |
| 9 | ahf_AN | ahf_AN | 0.9 | 1.5 | 0 | |
| 10 | ahf_AS | ahf_AS | 0.5 | 0.8 | 0 | |
| 11 | ahf_CO | ahf_CO | 0.5 | 0.8 | 0 | |
| 12 | ahf_HR | ahf_HR | 0.65 | 1.25 | 0 | |
| 13 | ahf_MN | ahf_MN | 0.5 | 0.8 | 0 | |
| 14 | ahf_MO | ahf_MO | 0.5 | 0.8 | 0 | |
| 15 | ahf_MS | ahf_MS | 0.5 | 0.8 | 0 | |
| 16 | ahf_NA | ahf_NA | 0.9 | 1.5 | 0 | |
| 17 | a_cott | a_cott | 0.65 | 1.2 | 0 | |
| 18 | a_sugr | a_sugr | 0.65 | 1.2 | 0 | |
| 19 | a_teas | a_teas | 0.65 | 1.2 | 0 | |
| 20 | a_coff | a_coff | 0.65 | 1.2 | 0 | |
| 21 | a_toba | a_toba | 0.65 | 1.2 | 0 | |
| 22 | a_food | a_food | 0.65 | 1.2 | 0 | |

### *3.2.3 Production Function Elasticities*

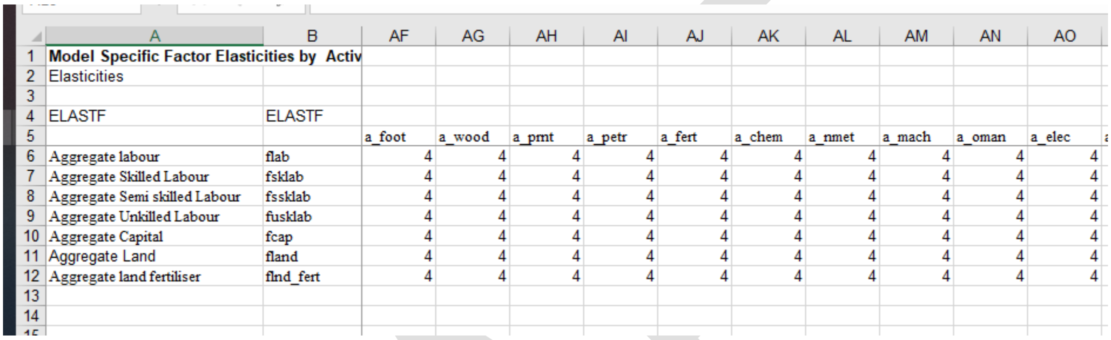
Substitution elasticities are required for both levels of the nested CES production functions. These are recorded in two worksheets, 'actelastx' and 'actelastva', where the former contains the elasticities of substitution between aggregate intermediate inputs and aggregate value added (model parameter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

[15] 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.

for differences in the substitution elasticities by activity and region.[16] 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**.

**Figure 3.2.2.1 Production Function Elasticities**



| | A | B | AF | AG | AH | AI | AJ | AK | AL | AM | AN | AO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **Model Specific Factor Elasticities by Activ** | | | | | | | | | | | |
| 2 | Elasticities | | | | | | | | | | | |
| 3 | | | | | | | | | | | | |
| 4 | ELASTF | ELASTF | | | | | | | | | | |
| 5 | | | a_foot | a_wood | a_pmt | a_petr | a_fert | a_chem | a_nmet | a_mach | a_oman | a_elec |
| 6 | Aggregate labour | flab | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 7 | Aggregate Skilled Labour | fsklab | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 8 | Aggregate Semi skilled Labour | fssklab | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 9 | Aggregate Unkilled Labour | fusklab | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 10 | Aggregate Capital | fcap | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 11 | Aggregate Land | fland | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 12 | Aggregate land fertiliser | flnd_fert | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 13 | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | |

### *3.2.4 Demand 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 – "hous" – 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**.

[16] 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 3.2.2.1 Income Elasticities

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **Model Specific Household Income Elasticities for LES** | | | | | | | | | | |
| 2 | Elasticities | defined over cles(cc) which here is the aggregate commodities | | | | | | | | | |
| 3 | | | | | | | | | | | |
| 4 | ELASTY | ELASTY | ELASTY | ELASTY | ELASTY | | | | | | |
| 5 | | | hAN_RU | hAN_UR | hAS_RU | hAS_UR | hCO_RU | hCO_UR | hHR_RU | hHR_UR | hMN_RU |
| 6 | Wheat | c_whea | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |
| 7 | Maize | c_maiz | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |
| 8 | Rice | c_rice | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |
| 9 | Other cereals | c_ogrn | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |
| 10 | Pulses & oil seeds | c_oils | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |
| 11 | Cotton | c_cott | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |
| 12 | Sugarcane | c_sugr | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |
| 13 | Roots & tubers | c_root | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |
| 14 | Vegetables | c_vege | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| 15 | Fruits | c_frui | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| 16 | Coffee | c_coff | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| 17 | Tea | c_tea | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| 18 | Others crops | c_ocrp | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| 19 | Sheep, goat and lamb for slaught | c_goat | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| 20 | Beef | c_beef | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| 21 | Poultry | c_poul | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| 22 | Other livestock | c_oliv | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| 23 | Forestry | c_fore | 2.2 | 2.2 | 2.2 | 2.2 | 2.2 | 2.2 | 2.2 | 2.2 | 2.2 |
| 24 | Fishing | c_fish | 2.2 | 2.2 | 2.2 | 2.2 | 2.2 | 2.2 | 2.2 | 2.2 | 2.2 |

## Figure 3.2.2.1 Frisch Elasticities

| | A | B | C |
|---|---|---|---|
| 1 | **FRISCH Parameters** | | |
| 2 | Elasticities | | |
| 3 | | | |
| 4 | ELASTMU | ELASTMU | ELASTMU |
| 5 | | | frisc |
| 6 | Arid North - Rural | hAN_RU | -3 |
| 7 | Arid North - Urban | hAN_UR | -3 |
| 8 | Arid South - Rural | hAS_RU | -3 |
| 9 | Arid South - Urban | hAS_UR | -3 |
| 10 | Coast - Rural | hCO_RU | -3 |
| 11 | Coast - Urban | hCO_UR | -3 |
| 12 | High Rainfall - Rural | hHR_RU | -3 |
| 13 | High Rainfall - Urban | hHR_UR | -3 |

## Figure 3.2.2.1 CES Demand Elasticities

| B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|
| **Model Specific Commodity Elasticities for CES Utility Functions** | | | | | | | | | |
| Elasticities | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | hh_RuRDA | hh_RuRCe | hh_RuRNc | hh_RuRSw | hh_RuREa | hh_UrbDA | hh_UrbCe | hh_UrbNo | hh_UrbSw |
| c_AgrF | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| c_Mfg | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| c_Svc | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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.

# 4 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.

## *4.1 Model Controls*

The `mod_cont` parameters are contained in a data table that consists of various elements – Figure 4.1.1. Also included are examples of typical default values and brief descriptions of the role of each parameter in the configuration of the model.

**Figure 4.1.1 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 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 | 1E-06 | | | 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 | trunc | 2 | | | This INTEGER defines the number of decimal places to which the SAM will be truncated |
| 18 | ras_solve | 1 | | | Choice of Solver for RAS module: 1 = CONOPT; 2 = PATHNLP; |

# 5. 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 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 it 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.[17] 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.

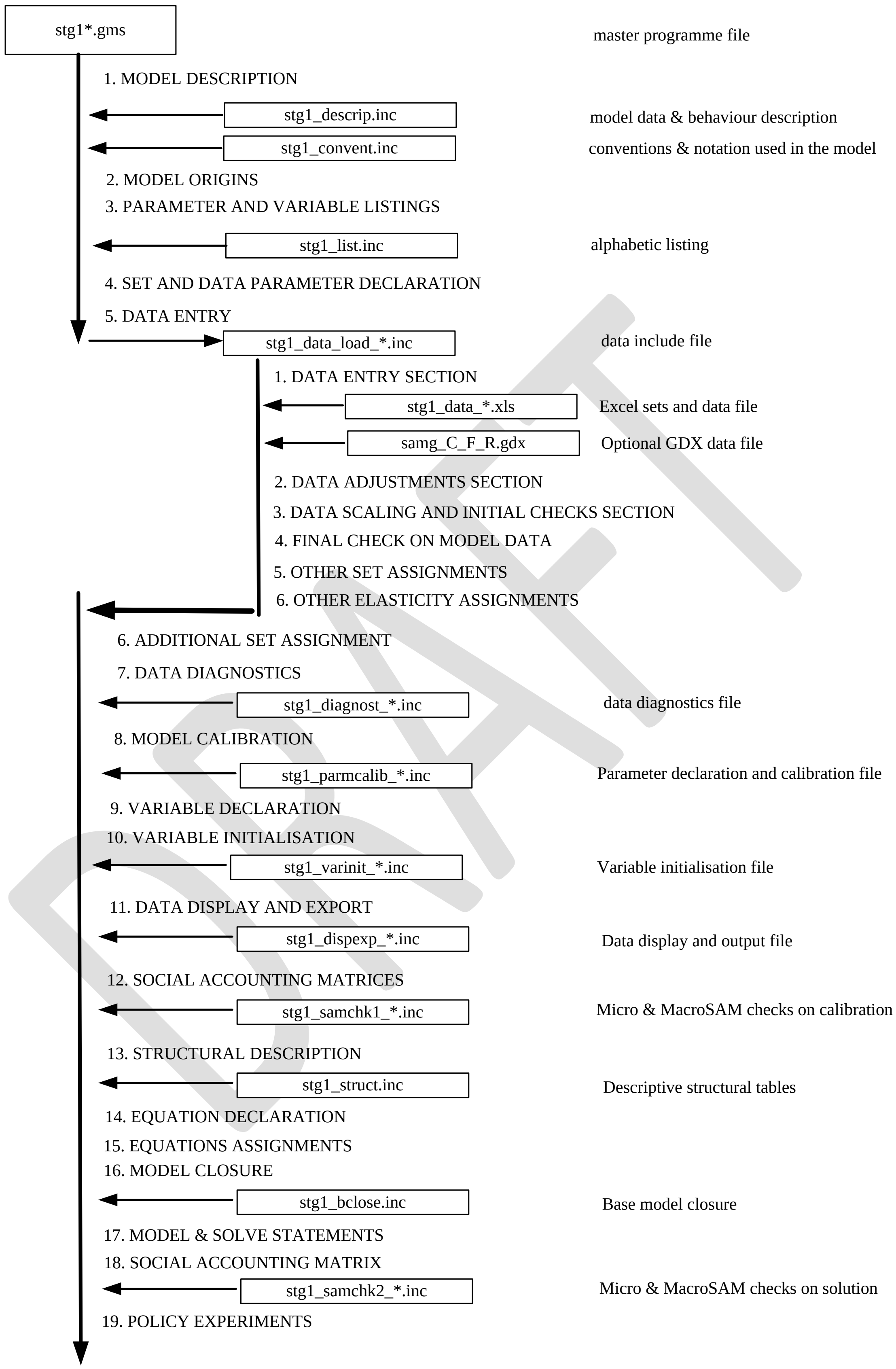
In this section 2 schematics are presented to assist the user in understanding the structure of the programme's code. The first deals with the code for the basic model while the second relates to the template experiment file.

## 5.1 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:

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

**Figure 5.1.1 STAGE File Structure**




- `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 checks[18] 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.[19]

After the data has been conditioned and checked all the model's parameters are declared and assigned (stg1_parmcalib.inc) followed by the declaration of all model variables[20] 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 consistent with the data (after any adjustments) that were loaded. The second (stg1_struct.inc) computes are 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.

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

19 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.

20 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.

# 6. 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.

# 7. 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.

# 8. Experiment File Template

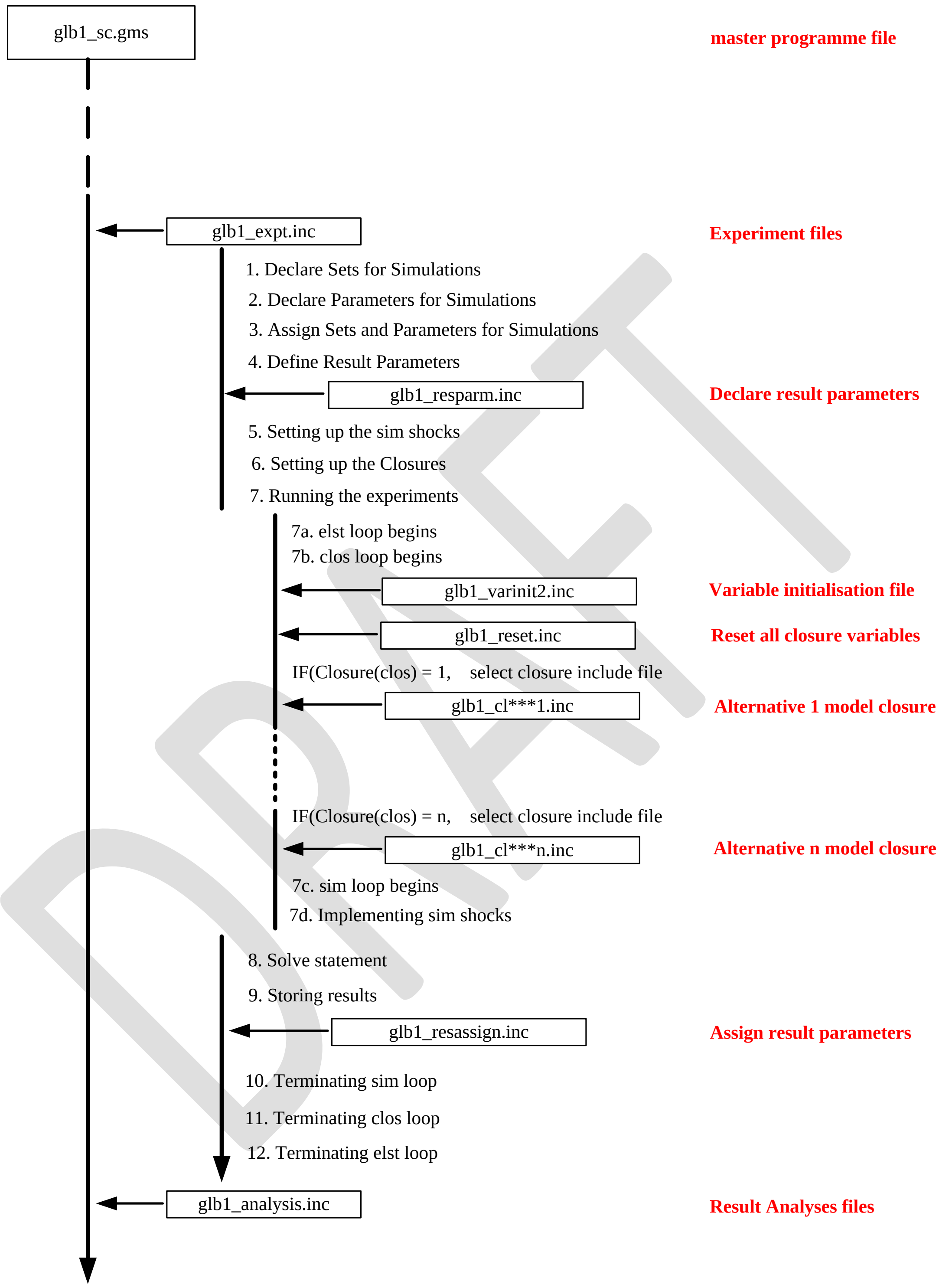
The experiment file is where the user has to be most active. The model file is largely straightforward and any changes the user would make are standardised, but the range of potential changes in the experiment file are very large and defy standardisation.

## 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).

# 9. 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

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | **Simulation sets** | | | | | | | |
| 2 | | | | | | | | |
| 3 | **Simulations set** | | | **Applied Simulations set** | | | **Reported Simulations set** | |
| 4 | **sim** | **(description)** | | **sim1** | **(description)** | | **sim2** | **(description)** |
| 5 | base | base simulation | | base | base simulation | | base | base simulation |
| 6 | sim02 | remove EU27 export taxes | | sim02 | remove EU27 export taxes | | sim02 | remove EU27 export taxes |
| 7 | sim03a | half remove EU27 Import duties | | sim03a | half remove EU27 Import duties | | sim03 | remove EU27 Import duties |
| 8 | sim03 | remove EU27 Import duties | | sim03 | remove EU27 Import duties | | sim04 | remove EU27 trade taxes |
| 9 | sim04 | remove EU27 trade taxes | | sim04 | remove EU27 trade taxes | | sim05 | common EU27 Average trade taxes |
| 10 | sim05 | common EU27 Average trade taxes | | sim05 | common EU27 Average trade taxes | | sim07 | common EU27 Average factor use taxes |
| 11 | sim06 | common EU27 Exogenous trade taxes | | sim06 | common EU27 Exogenous trade taxes | | sim09 | EU27 harmonisation on Average tax rates |
| 12 | sim07a | half common EU27 Average factor use taxes | | sim07a | half common EU27 Average factor use taxes | | | |
| 13 | sim07 | common EU27 Average factor use taxes | | sim07 | common EU27 Average factor use taxes | | | |
| 14 | sim08 | common EU27 Exogenous factor use taxes | | sim08 | common EU27 Exogenous factor use taxes | | | |
| 15 | sim09a | part EU27 harmonisation on Average tax rates | | sim09a | part EU27 harmonisation on Average tax rates | | | |
| 16 | sim09 | EU27 harmonisation on Average tax rates | | sim09 | EU27 harmonisation on Average tax rates | | | |
| 17 | sim10 | EU27 harmonisation on Exogenous tax rates | | | | | | |
| 18 | | | | | | | | |

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.[21] 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

| | I | J | K | L | M | N |
|---|---|---|---|---|---|---|
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | **Closures set** | | | **Elasticities set** | |
| 4 | | **clos** | **(description)** | | **elst** | **(description)** |
| 5 | | clbase | Balanced macro closure | | elst01 | base elasticities |
| 6 | | clos02 | IPTS long run closure full emplyt | | | |
| 7 | | clos03 | IPTS long run closure umemplyt | | | |
| 8 | | clos04 | IPTS short run closure umemplyt | | | |
| 9 | | | | | | |
| 10 | | | | | | |
| 11 | | | | | | |
| 12 | | | | | | |
| 13 | | | | | | |
| 14 | | | | | | |
| 15 | | | | | | |
| 16 | | | | | | |
| 17 | | | | | | |

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.

---

21 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**

D4 fx reunew

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | | cze | Czech Republic | | aut | Austria | | roecd | Rest of the OECD |
| 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 .

## Figure 10.2.1 econtrols

A7 | econs

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | **Control Parameters** | | | | |
| 2 | **These values are used to initialise various parameters that condition experiments using the GLOBE programme** | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | **exp_control** | | | | |
| 6 | | | | | |
| 7 | **econs** | | | | |
| 8 | reslevel | 1 | | | IF 1 then womod_reslevel runs; IF 0 then womod_reslevel does not run |
| 9 | ressamg | 0 | | | IF 1 then womod_ressamg runs; IF 0 then womod_ressamg does not run |
| 10 | resmacro | 1 | | | IF 1 then womod_resmacro runs; IF 0 then womod_resmacro does not run |
| 11 | resstruct | 1 | | | IF 1 then womod_resstruct runs; IF 0 then womod_resstruct does not run |
| 12 | reswelf | 1 | | | IF 1 then womod_reswelf runs; IF 0 then womod_reswelf does not run |
| 13 | resindex | 0 | | | IF 1 then womod_resindex runs; IF 0 then womod_resindex does not run |
| 14 | restaxstr | 0 | | | IF 1 then womod_restaxstr runs; IF 0 then womod_restaxstr does not run |
| 15 | resperc | 1 | | | IF 1 then womod_resperc runs; IF 0 then womod_resperc does not run |
| 16 | resbase | 1 | | | IF 1 then basis for comparative results are initial values; IF 2 then bases are from an experiment |
| 17 | | | | | |

## 10.3 Data

For some experiments it is convenient to provide the values for the shocks from a table of data, an example of such a data table is provided in Figure 10.3.1. If a user chooses this method a parameter must be declared in the experiment file, assigned in the Excel worksheet, the layout/index sheet must be extended to include the new parameter and then the data loaded using GDXIN.

## Figure 10.3.1 Experiment Data Sheet

A1 | Common External Tariff of EU15 and EU25 with Trade Partners in accordance with the Regional and ( ANEFIS, ad valorem in %

| | A | B | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **Common External Tariff of EU15 and EU25 with Trade Partners in accordance with the Regional and Commodity Aggregation of ANEFIS, a** | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | |
| 4 | **EU15** | | | | | | | | | | | | | |
| 5 | | cgran | cscb | cocrp | cpbf | clstk | cmlk | caprd | cmins | cmeat | cmprd | cvof | cdair | csu |
| 6 | wdeu | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 7 | wita | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 8 | waut | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 9 | wgbr | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 10 | wfra | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 11 | wbnl | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 12 | wespt | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 13 | wreu | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 14 | wpol | 11.656 | 10.113 | 0.984 | 0 | 33.506 | 0 | 1.844 | 0 | 69.09 | 16.814 | 3.408 | 41.365 | |
| 15 | whun | 6.642 | 0 | 2.686 | 0 | 16.129 | 0 | 2.955 | 0 | 52.264 | 14.134 | 0.573 | 29.953 | |
| 16 | wcze | 3.763 | 0 | 0.758 | 0 | 50.422 | 0 | 2.802 | 0 | 52.851 | 11.972 | 8.733 | 44.389 | |
| 17 | wreur | 1.468 | 63.055 | 1.149 | 0 | 21.137 | 0 | 1.249 | 0 | 35.94 | 11.504 | 3.578 | 40.121 | |
| 18 | wrobu | 15.366 | 2.057 | 18.084 | 0.204 | 3.373 | 0 | 7.091 | 2.032 | 30.062 | 42.726 | 10.192 | 26.932 | |
| 19 | wtur | 31.866 | 19.022 | 19.46 | 0 | 3.776 | 0 | 2.12 | 0 | 65.09 | 72.894 | 13.564 | 84.425 | |
| 20 | wroecd | 17.616 | 13.909 | 11.108 | 0.445 | 1.958 | 0 | 4.218 | 0.353 | 35.176 | 50.065 | 4.334 | 34.186 | |
| 21 | wcis | 55.272 | 52.808 | 16.304 | 1.944 | 1.897 | 0 | 3.826 | 1.936 | 26.106 | 28.241 | 15.681 | 23.847 | |
| 22 | wmerc | 10.863 | 9.544 | 7.445 | 9.271 | 5.342 | 0 | 7.045 | 4.729 | 11.965 | 14.517 | 12.941 | 19.203 | |
| 23 | wrow | 14.462 | 30.776 | 13.024 | 5.597 | 0.985 | 0 | 8.143 | 6.6 | 15.121 | 15.259 | 15.785 | 11.04 | |
| 24 | wglo | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 25 | | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | | |
| 28 | | | | | | | | | | | | | | |
| 29 | | | | | | | | | | | | | | |
| 30 | **EU25** | | | | | | | | | | | | | |
| 31 | | cgran | cscb | cocrp | cpbf | clstk | cmlk | caprd | cmins | cmeat | cmprd | cvof | cdair | csu |
| 32 | wdeu | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

## 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**

A1 fx LAYOUT

| | 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 | rerfx | clsets!A5 | 1 | | | |
| 19 | dset | rsavfx | clsets!D5 | 1 | | | |
| 20 | dset | rinvshfx | clsets!G5 | 1 | | | |
| 21 | dset | riadjfx | clsets!J5 | 1 | | | |
| 22 | dset | rkapgfx | clsets!M5 | 1 | | | |
| 23 | dset | rluen | 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 | reu15 | expsets!A5 | 1 | | | |
| 30 | dset | reunew | expsets!D5 | 1 | | | |

# 10. 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
     glb1_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 multiplictive 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 equilibriating variable is Government Savings

* Tax rate scaling factors

* T*ADJ control multiplictive 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.
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)            = FD0(lu,a,rluex) ;
*  WF.FX(lu,rluex)                = WF0(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)            = FD0(lu,a,rluen) ;
  WF.FX(lu,rluen)              = WF0(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) ;
*  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

* 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 ;
```

# 11 Predefined Closure Files

# 12. Predefined Parameters for Experiments

## 12.1 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*.

## 12.2 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*.

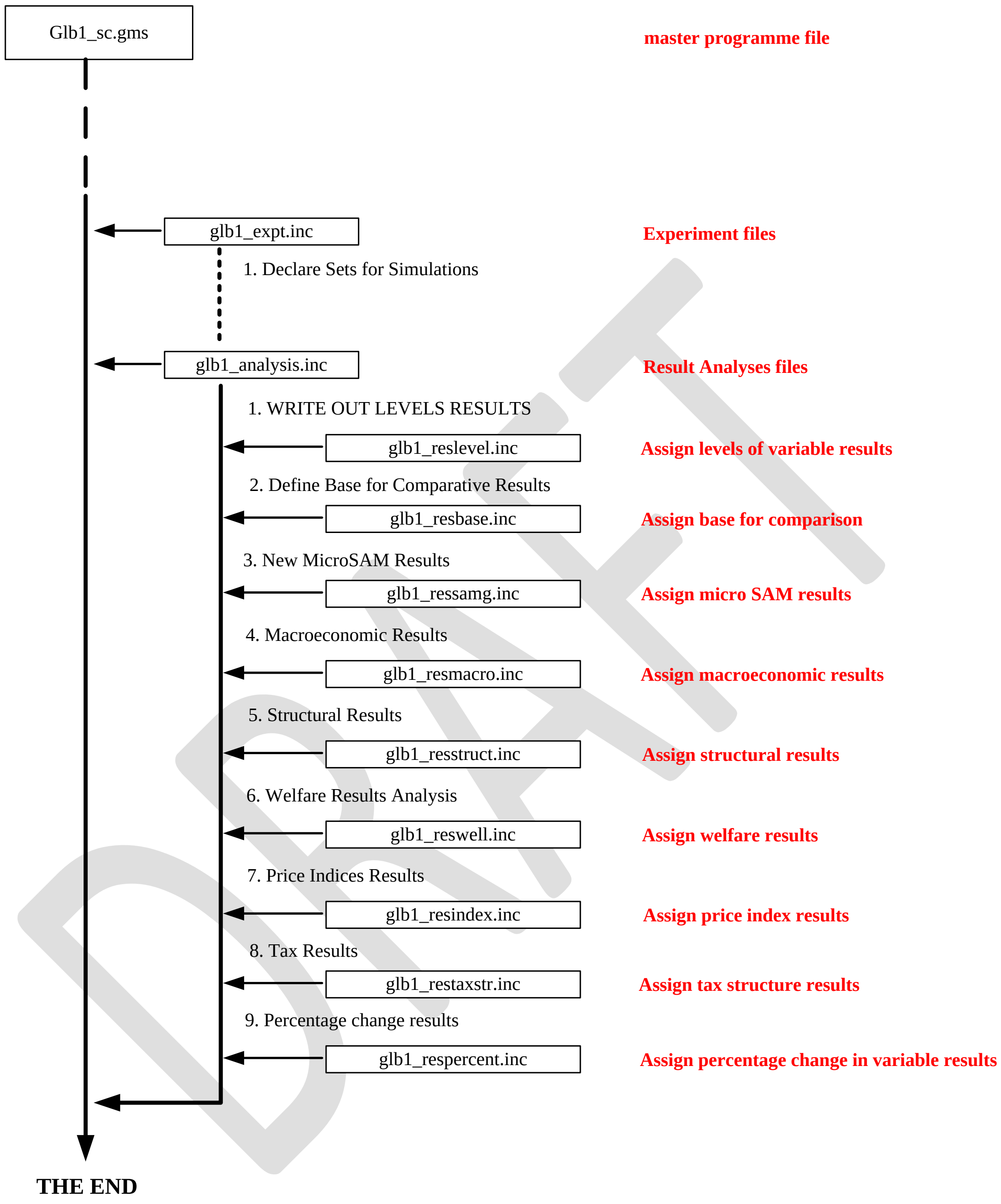
# 13. Analysis File

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).

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

**Figure 14.1 STAGE Results Analysis File Structure**



# 14. STAGE_RDYN Technical Notes

The technical notes regarding the STAGE_RDYN codes focus exclusively on the recursive dynamic elements, i.e., those elements required for the comparative static core model are not considered. The technical notes are classified under five headings: Data Requirements, Code Organisation and File Structure, Recursive Dynamics, Business as Usual Scenario (Baseline) and Presentation of the Results. The notes refer to the current implementation and therefore do not embrace the options for future developments discussed below (section 6).

Data Requirements

The additional data required for implementing STAGE_RDYN simulations are relatively limited: a series of projections for the baseline – typically macroeconomic aggregates and relative productivity growth rates; a series of parameters that drive the responsiveness of parameters in the recursive dynamic phases, i.e., those updated between solutions; a series of trends that reflect underlying changes in relationships that would be expected under the BaU and simulation scenarios and subsets and data relating to policy instruments that can influence the recursive dynamics. All these data are included in a single Excel workbook (stg_dev_dyn_calib.xlsx); Figure 15.1 shows the format of the 'layout' worksheet that defines the structure of the workbook.

**Figure 15.1 Recursive Dynamic Workbook Layout Sheet**

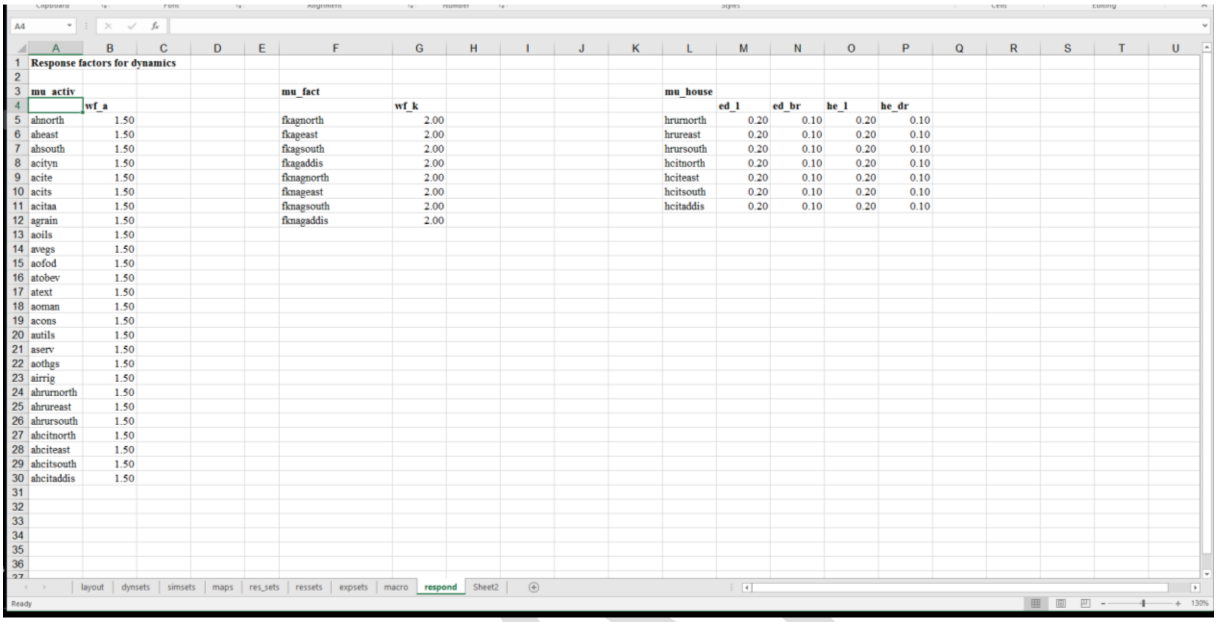
LAYOUT

| Data Type | Name | Location | Row dimension | Column dimension | Total dimension |
|---|---|---|---|---|---|
| | | | rdim | cdim | dim |
| dset | educate | dynsets!A5 | 1 | | |
| dset | health | dynsets!D5 | 1 | | |
| dset | sim | simsets!A5 | 1 | | |
| dset | sim1 | simsets!D5 | 1 | | |
| dset | simr | simsets!G5 | 1 | | |
| dset | elst | simsets!J5 | 1 | | |
| dset | clos | simsets!M5 | 1 | | |
| dset | closelem | simsets!P5 | 1 | | |
| set | map_k_i | maps!A5 | 2 | | |
| set | map_i_k | maps!D5 | 2 | | |
| dset | adjres | res_sets!A6 | 1 | | |
| dset | scalres | res_sets!D6 | 1 | | |
| dset | stab1 | res_sets!G6 | 1 | | |
| dset | stab3 | res_sets!J6 | 1 | | |
| dset | stab4 | res_sets!M6 | 1 | | |
| dset | stab5 | res_sets!P6 | 1 | | |
| dset | mbas | macro!A7 | 1 | | |
| par | gr_macro | macro!A7 | 1 | | |
| par | mu_activ | respond!A4 | 1 | 1 | |
| par | mu_fact | respond!F4 | 1 | 1 | |
| par | mu_house | respond!L4 | 1 | 1 | |

The projections for the baseline are typically expressed as equilibrium growth rates; this reflects the underlying presumption of a steady-state growth path and that the economy in the base year is an equilibrium on the steady-state growth path (see below for details on the implementation). The number of exogenous variables for which projections can be imposed is flexible; but the greater the number of imposed projections the less flexible the model is in terms of both finding solutions and macroeconomic and market clearing options.

Changes in policies and/or patterns of expenditures, e.g., on education and healthcare, will drive endogenously change behaviours, e.g., rate of increase in human capital accumulation. The user needs to define the responsiveness of endogenously determined behaviours to changes in the drivers associated with each behaviour. The user needs to set these responsiveness parameters – typically elasticities – in the worksheet 'respond'. As illustrated in the Figure 15.2 these parameters are activity, factor and household (RHG) specific. But the empirical evidence underpinning these parameter values is limited and/or non-existent, and therefore there are strong arguments for sensitivity analyses of these parameters.[22]

**Figure 15.2 Responsiveness Parameter Worksheet**



| | A | B | F | G | L | M | N | O | P |
|---|---|---|---|---|---|---|---|---|---|
| 1 | **Response factors for dynamics** | | | | | | | | |
| 2 | | | | | | | | | |
| 3 | **mu_activ** | | **mu_fact** | | **mu_house** | | | | |
| 4 | | **wf_a** | | **wf_k** | | **ed_l** | **ed_br** | **he_l** | **he_dr** |
| 5 | ahnorth | 1.50 | fkagnorth | 2.00 | hrurnorth | 0.20 | 0.10 | 0.20 | 0.10 |
| 6 | aheast | 1.50 | fkageast | 2.00 | hrureast | 0.20 | 0.10 | 0.20 | 0.10 |
| 7 | ahsouth | 1.50 | fkagsouth | 2.00 | hrursouth | 0.20 | 0.10 | 0.20 | 0.10 |
| 8 | acityn | 1.50 | fkagaddis | 2.00 | hcitnorth | 0.20 | 0.10 | 0.20 | 0.10 |
| 9 | acite | 1.50 | fknagnorth | 2.00 | hciteast | 0.20 | 0.10 | 0.20 | 0.10 |
| 10 | acits | 1.50 | fknageast | 2.00 | hcitsouth | 0.20 | 0.10 | 0.20 | 0.10 |
| 11 | acitaa | 1.50 | fknagsouth | 2.00 | hcitaddis | 0.20 | 0.10 | 0.20 | 0.10 |
| 12 | agrain | 1.50 | fknagaddis | 2.00 | | | | | |
| 13 | aoils | 1.50 | | | | | | | |
| 14 | avegs | 1.50 | | | | | | | |
| 15 | aofod | 1.50 | | | | | | | |
| 16 | atobev | 1.50 | | | | | | | |
| 17 | atext | 1.50 | | | | | | | |
| 18 | aoman | 1.50 | | | | | | | |
| 19 | acons | 1.50 | | | | | | | |
| 20 | autils | 1.50 | | | | | | | |
| 21 | aserv | 1.50 | | | | | | | |
| 22 | aothgs | 1.50 | | | | | | | |
| 23 | airrig | 1.50 | | | | | | | |
| 24 | ahrurnorth | 1.50 | | | | | | | |
| 25 | ahrureast | 1.50 | | | | | | | |
| 26 | ahrursouth | 1.50 | | | | | | | |
| 27 | ahcitnorth | 1.50 | | | | | | | |
| 28 | ahciteast | 1.50 | | | | | | | |
| 29 | ahcitsouth | 1.50 | | | | | | | |
| 30 | ahcitaddis | 1.50 | | | | | | | |



Existing policies represent part of the BaU scenario and hence should be represented. In addition, there are ongoing transformations to the economic system that emerge as an evolution of existing changes in social relations. It is very difficult to define the underlying causes of these changes in economic terms and their quantification is extremely difficult. In keeping with the backward-looking logic of a recursive dynamic model these are represented in STAGE_RDYN as simple trends.[23] For example, it is assumed that the year on year decline in birth rates persists, e.g., a one percent year on year reduction, which gives an underlying trend that can be modified by changes in policies, e.g., the volume of healthcare and/or education expenditures per capita, that increase or decrease the trend rate.

There are also two worksheets to allow for sets and or subsets ('dyn_sets') and data ('dyn_data') required for the recursive dynamics. Examples of the roles that these worksheets

---

22 Unlike standard CES/CET elasticities for which experience has shown that the results are often not particularly sensitive to elasticity assumptions, the sensitivity of results to these parameters is not well understood and preliminary analyses suggest the results can be sensitive to chosen values.

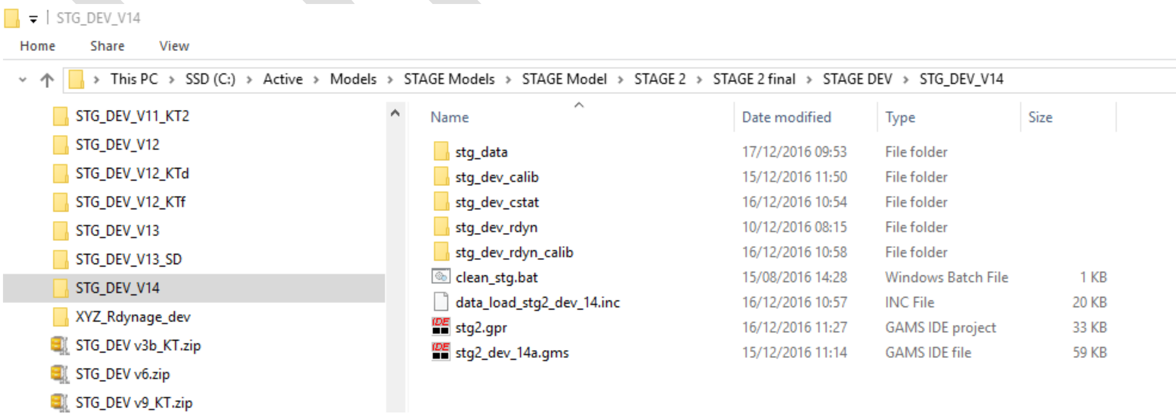
23 "From a quantitative, empirical point of view, we are left with time as an explanatory variable. Now trend projections, however necessary they may be in practice, are basically a confession of ignorance, and, what is worse from a practical viewpoint, are not policy variables." (Arrow, 1962, p 155).

can fill are given for the situation where the underlying database does not include separate accounts for the commodities ‘education’ and ‘healthcare’: the dyn_sets worksheet identifies the commodity account that encompasses ‘education’ and ‘healthcare’, in this case ‘marketed services’ (*cmserv*), while the dyn_data worksheet identifies the proportions of ‘marketed services’ accounted for by ‘education’ and ‘healthcare’. This is an illustration of an important principle: just because the data are poor or missing does not require the behaviour to be excluded from the model, rather it is better to include the behaviour using poor quality data, or even crude estimates.

<u>Code Organisation and Directory and File Structure</u>

The STAGE_DEV model uses an approach to the organisation of the code that is being adopted across the STAGE/GLOBE family of models; the principle underlying the organisation is that each model should be able to be implemented in either a comparative static or recursive dynamic mode using the same set of files AND each file should only appear once in the system of files. This is achieved by organising the files within a system of directories and using the save/restart facilities provided by GAMS. This is shown in Figure 15.31, for the situation BEFORE the programmes have been run (the commentary below assumes the use of GAMSIDE and Windows Explorer; other choices of EMAC and directory viewer will look different).

**Figure 15.3 Directory and File Structure**




The master directory, in this case ‘STG_DEV_V14’, contains four files and five sub-directories. The four files in the master directory are the core comparative static model (‘*stg2_dev_14a.gms*’), the include file that manages the loading of sets and data

(‘*data_load_stg2_dev_14.inc*’), the project file (‘*stg2.gpr*’) and a batch file that is used to clean out surplus files (‘*clean_stg.bat*’). All the other required files are collected within appropriate sub directories.

1. ‘*stg_data*’: this sub directory contains all the Excel files used to provide sets and data used to calibrate the comparative and recursive dynamic models and to implement comparative static and recursive dynamic experiments.
2. ‘*stg_dev_calib*’: this sub directory contains all the code files required to calibrate the comparative static model and prepare the files required for the recursive dynamic components.
3. ‘*stg_dev_cstat*’: this sub directory contains all the code files required implement comparative static experiments. Within this sub directory there is a file (‘*stg2_dev_cstat_expt.gms*’) that is used to run comparative static experiments.
4. ‘*stg_dev_rdyn_calib*’: this sub directory contains all the code files required to calibrate the recursive dynamic model.
5. ‘*stg_dev_rdyn*’: this sub directory contains all the code files required to implement recursive dynamic experiments. Within this sub directory there is a file (‘*stg2_dev_rdyn_expt.gms*’) that is used to run recursive dynamic experiments.

Operating this arrangement requires the use of multiple INCLUDE file (‘*.*inc*’) that are called in from the appropriate directory, i.e., the various include statements in the programmes require the specification of the path that identifies the directory location of the file to be included. It also requires the use of GAMS’s save and restart facility to pass work files (*.*g00*) to required sub directories; this requires the user to ensure that the appropriate commands are included in the command line, see Figure 15.4. The save instructions in this illusatration require that the file ‘*save.g00*’ is saved into the sub directory ‘*stg_dev_cstat*’ (`s=stg_dev_cstat\save`) and the sub directory ‘*stg_dev_rdyn_calib*’ (`s=stg_dev_rdyn_calib\save`).[24]

[24] Consideration is being given to adding an additional sub directory – *stg_save* – for the storage of all save files. This has the advantage of reducing the number of save instructions BUT requires the use of path information in the restart instructions.

**Figure 15.4 Command line Instructions**


Another aspect of the coding principles is a dramatic reduction in the use of Display statements in the code. Rather the code makes use of the GAMS Data Exchange facilities to record all the information relating to sets (and alias), parameters, variables and equations in a single GDX file. This is implemented by adding the instruction `gdx=input_base` to the command line, see Figure 15.4, which ‘automatically’ records the information if the file *input_base.gdx*. The file *input_base.gdx* can be opened in GAMSIDE and easily searched using the `Symbol search` box in the lower left-hand corner.

Given this method of organising and running programmes the file structures inevitably become complex and difficult to follow, and it can be taxing to identify where in the system of files symbols are declared (declared’), defined (‘defined’), assigned (‘assigned’), referenced (‘ref’), assigned in implementation (impl-asn’) and used as a control (‘control’). Therefore, the command line is used to generate a reference file (‘**.ref*’), using the instruction `rf=ref_base` that collects all this information for the model’s symbols and identifies all the files used by the model. The resultant reference file can be opened in GAMSIDE.

<u>Recursive Dynamics</u>

In any backward-looking model agents are assumed to respond to incentives in the current period to determine their actions in the next period. Thus, if the price of some factor increases agents should try to increase the quantity of that factor; but if agents are assumed to respond fully to such changes in incentives the resultant system could experience wild fluctuations, e.g., the ‘hog cycle’. The favoured approach is that of partial adjustment, where the extent of adjustment is determined by exogenous adjustment parameters (*mu_**); consequently, if a single shock is introduced the system will adjust to this shock overtime and that a number of time periods are required before the system returns to a long run equilibrium state.[25] The time taken to return the long run equilibrium will depend on adjustment rates (parameters) and the size and nature of the shock.

The standard presumption is that the model is solved for annual steps under the maintained assumption that agents make decisions about resource allocation at the start of the **current** period, e.g., the amount of physical capital available to each activity. These decisions are then revised based on the outcomes, primarily prices, in the **current** period to determine the decisions about resource allocation at the start of the **next** period.[26]

It is relevant to note that the existence of stock changes in base data sets implies that the base data do not represent a long run equilibrium situation. Therefore, it can be appropriate to include a trend rate that eliminates the stock changes over time. However, while this option achieves an objective – getting rid of stock changes – it does not, and cannot, address the fundamental problem about stock changes: namely that their existence demonstrates that the initial database may not have been in a steady-state equilibrium.

Demographics

At its simplest level demographics is about little more than changes in the size of populations. On a slightly more complex level it is concerned with how the structures of populations change with respect to age and gender profiles, while at a still more complex level it involves tracking how populations evolve in terms of their ownership of assets through *inter alia* the processes of education, training and health. The approach embedded in this version of STGAE_RDYN is a compromise that seeks to limit the data requirements for its implementation while providing enough additional information to capture aspects of the evolution of the populations of and labour supplies by RHGs.[27]

25 A solution from a comparative static model after a shock represents such a long run equilibrium.

26 This approach is rooted in the tradition of Harrod-Domar growth models where backward looking decisions are important. This does raise important question about the appropriate definitions for investment functions (see Khan, ??).

27 It is pertinent to note that while demographics were not an explicit component of the contract they have been added because otherwise he endogenous functional distribution of income element of the STAGE_DEV model in the contract would have been substantially weakened.

The demographics are implemented through two include files in the recursive dynamic codes[28]:

1. '*stg2_dyn_demog_14.inc*': this file defines all the parameter and other symbols referring to demography used in STAGE_RDYN and assigns their initial values, there is an option to load additional demographic specific data within this file.

2. '*stg2_dyn_pop_14.inc*': this file is implemented within the time loop when running the recursive dynamic model. It implements any trend rates of change in birth and death rates by RHG, and then further adjusts the birth and death rates by RHG in response to changes in economic incentives – the volumes of healthcare and education expenditures by the government. The age profile of the population by RHG is updated given the birth and death rates.

The derivation of the population profiles depends upon the information available when the databases are being compiled. If the RHGs are defined using data derived from household income and expenditure surveys (HIES), or censuses that can be linked to the HIES, then there is usually enough data to estimate the numbers of persons in certain age bands and from those data to infer the numbers in each annual cohort. Given the numbers in each annual cohort it is also possible to derive estimates of the numbers of children borne each year and hence the estimates of the crude birth rate adjusted for early year infant mortality.

The methods coded in the file '' assume very limited information. The numbers in the age group - 0-15, 16-55, and 55+ - the crude birth and death rates; these data are freely available from the World Bank's database.

The process by which birth and death rates are adjusted uses a simple constant elasticity function to generate adjustment factors, e.g.,

$$factor_h^t = factor_h^{t-1} * \left\{ \frac{q_{h,i}^t}{q_{h,i}^{t-1}} \right\}^{\mu_h^t} \qquad (15.1)$$

28 GAMS requires that symbols are declared and defined outside off conditional statements, e.g., loops. Because the changes through time are implemented inside a (time) loop it is necessary to have two files – one outside and one inside the loop.

where $factor_h^t$ is the adjustment factor for RHG h in period *t*, $q_{h,i}^t$ is the volume the intervention per head of population in the RHG and $\mu_h^t$ is the responsiveness of the rate to the change in the volume of per capital intervention. Then the birth or death rate is updated, when there are no trend changes in the rate, using

$$rate_h^t = \left( factor_h^t \right) * rate_h^{t+1} \qquad \textbf{OR} \qquad rate_h^t = \left( \frac{1}{factor_h^t} \right) * rate_h^{t+1} . \tag{15.2}$$

The different formulae in (15.2) depend upon whether it is assumed the intervention increases or reduces the rate. Thus, for instance, increases in healthcare expenditure may be assumed to increase the **effective** birth rates because of reduced foetal and early years (under 5) deaths, while increases in education expenditure may be assumed to decrease the **effective** birth rates because of increases in the opportunity costs of pregnancy and female empowerment.

Note how Eqn 15.3 ensures that the impact of changes in the volume of intervention per capita is cumulative. If the volume of intervention per capita increases continuously so will the adjustment factor, because the ratio of the current to the previous volume of intervention is greater than one, but if at any time the volume of intervention declines for one period to the next the ratio of the current to the previous volume of intervention will be less than one and the adjustment factor will reduce.

Having determined the birth and death rates for period *t*, the populations by each RHG are updated. Given the populations of each RHG in (*t*-1) the number of deaths and births can be derived. Deaths are recorded as removing the enough persons from the oldest annual cohorts of the populations, then the remaining populations in (*t*-1) are all aged by one annual cohort and births are recorded as populating the youngest (zero to one) cohort. Thus, the profiles of the populations for each RHG evolve, and with them the cohort sizes for dependent children, children of school age, dependent (old) adults and working aged adults.

<u>Human Capital Accumulation</u>

Human capital accumulation presents several conceptual issues and problems with the classification of working population. It has become common, for economists, to equate human capital accumulation with educational attainment, e.g., Barro (2001) and Barro and

Lee (1993 and 2012), but there is also a longstanding current in the human capital literature about the role of 'learning-by-doing' (Arrow, 1962) that highlights the role of human capital accumulation through experience, practice and on-going learning. Thus, while educational attainment may, on average, provide indications of the potential productivity of 'new' members of the workforce, it may be a poor indicator of the average flow of labour services from any given stock of labour. One approach may be to suggest that groups of individuals enter the workforce with a specific flow of labour services associated with their terminal educational attainment, and thereafter they continue to acquire skills that increase their productivity until age starts to degrade their productivity.

But the empirical basis available to calibrate such evolving labour productivity is sparse, to say the least. Moreover, there is a classification problem. Very often, following ILO convention, workers are classified by their occupations not their levels of skills; so even data can be collected on the educational attainment of labour services owned by households, it is rare to find data on the employment of labour by educational attainment by activities.

The approach taken for STAGE_RDYN is, again, a pragmatic one that seeks a compromise between data requirements and behavioural precision. The process can be conceived of as consisting of three elements; a trend rate (based on historical evidence) of human capital accumulation, educational expenditure that raises skill levels and health expenditures that enhance the ability of workers to generate labour services. All three elements contribute to increases in the efficiency of workers so that the *de facto* flow of services from any given stock of workers increase. As such the model carries around two measures of labour; the physical numbers of workers of each type supplied by each RHG and the efficiency factor for each type of labour supplied by each RHG. The former defines the ownership of each type of labour by each RHG in natural units – person hours – while the latter defines the flows of each type of labour service available for production.

The education component is defined as

$$eff\_l\_ed_{h,l}^{t} = eff\_l\_ed_{h,l}^{t-1} * \left( \frac{educ_h^t}{educ_h^{t-1}} \right)^{\mu_{h,t}^{ed\_l}} \tag{15.4}$$

where $eff\_l\_ed^{t}_{h,l}$ is the efficiency factor due to education for labour of type *l* provided by RHG *h* in period *t*; $educ^{t}_{h}$ is volume of education services per child in RHG *h* in period t and $\mu^{ed\_l}_{h,t}$ is the responsiveness of the efficiency factor to changes in the per child volume of education services. Similarly, the health component is defined as

$$eff\_l\_he^{t}_{h,l} = eff\_l\_he^{t-1}_{h,l} * \left( \frac{heal^{t}_{h}}{heal^{t-1}_{h}} \right)^{\mu^{he\_l}_{h,t}} \tag{15.5}$$

where $eff\_l\_he^{t}_{h,l}$ is the efficiency factor for labour due to health of type *l* provided by RHG *h* in period *t*; $heal^{t}_{h}$ is volume of health services per person in RHG *h* in period t and $\mu^{he\_l}_{h,t}$ is the responsiveness of the efficiency factor to changes in the per person volume of education services. Not how, in both cases, the efficiency factor is cumulative as in the case of the demographics.

The enhancement to productivity due to education and healthcare expenditures are assumed to be multiplication, hence

$$fsia^{t}_{h,l} = fsia^{t-1}_{h,l} * eff\_l\_ed^{t}_{h,l} * eff\_l\_he^{t}_{h,l} \tag{15.6}$$

where $fsia^{t}_{h,l}$ is the supply of labour services of type *l* by RHG *h* in period *t*.

The trend rate of human capital accumulation can be conceived of as ‘learning-by-doing’ and as being cumulative, thereby further enhancing the efficiency factor.

Physical Capital Accumulation

The updating of physical capital follows a standard perpetual inventory model. At the end of each period the existing capital is depreciated using economic depreciation rates that are functions of the intensity of use of the stock in the previous period and then gross investment is added to determine the stock of capital in the next period. It is assumed that new investment is delivered to activities in the same ratio as the existing capital stock **unless** the relative rates of return to capital by activity have changed. If the relative rates of return have changed then the pattern of investment by activity is different to the existing capital stock but does not wholly adjust to the new rates of return, i.e., there is a partial adjustment process. It is typically assumed that the stock:flow ratio is constant, i.e., the quality of the capital is constant.

Capital can be divided into 'fixed' and 'mobile' capital. It is assumed that once 'fixed' capital is located within an activity it can only exit that activity by economic and/or accelerated depreciation; examples of such capital would include power generating plants, machines for specific operations etc. This reflects the fact that the adjustment processes with respect to 'fixed' capital can be slow.[29] On the other hand, 'variable' capital enters a pool that can be allocated across activities in the current period in response to changes in the rates of return.

With multiple forms of capital goods, the commodities required to produce each form of capital will differ. But the commodities used for investment in capital for the next period must be produced in the current period which mean that the demand for commodities for investment

The first step is to convert investment expenditures into a volume of new capital. It important to be careful at this stage to ensure that there is no confusion between the value of capital stocks and the rate of return; this is important since the units in which value of capital stocks and the rates of return need to be consistent. The volume of new capital is defined as

$$gfcf_k^t = {invest\_f_k^t}/{pk_k^t} \tag{15.7}$$

where $gfcf_k^t$ is the gross fixed capital stock of type *k* in period *t*, $invest\_f_k^t$ is the value of investment in *k* in period *t*, which is defined as the summation of the expenditures on the different commodities (*QINVD*) used to produce capital of type *k*, and $pk_k^t$ is the price index of capital of type *k* produced in period *t*.

Then the new capital has to be allocated across activities. The investment shares for capital of type *k* by activity *a* are

$$inv\_sh\_ka_{k,a}^t = cap\_sh\_ka_{k,a}^{t-1} * \left(1 + \mu_a^t * \left(wfa\_rel_{k,a}^{t-1} - 1\right)\right) \tag{15.8}$$

29 This is one of the reasons to prefer solving the model year by year rather than in, say, 5 or 10 year periods.

where $inv\_sh\_ka_{k,a}^{t}$ is the share of the volume of investment of capital type *k* by activity type *a* in period *t*, $cap\_sh\_ka_{k,a}^{t-1}$ is the share of the volume of capital of type *k* used by activity of type *a* in period (*t*-1), $wfa\_rel_{k,a}^{t-1}$ is the relative rates of return to capital of type *k* in activity of type *a* in period (*t*-1) and $\mu_{a}^{t}$ is the partial adjustment (response) parameter for activity of type *a*. The quantities of new capital allocated to each activity $\left(d\_kap\_ka_{k,a}^{t}\right)$ are then defined as

$$d\_kap\_ka_{k,a}^{t} = inv\_sh\_ka_{k,a}^{t} * gfcf_{k}^{t} \tag{15.9}$$

and the new quantities of capital by activity for period are then given by

$$FD_{k,a}^{t} = \left(FD_{k,a}^{t} * \left(1 - deprec_{k}\right)\right) + d\_kap\_ka_{k,a}^{t} \tag{15.10}$$

where $FD_{k,a}^{t}$ is the stock of capital of type *k* available to activity *a* at the start of period t and $deprec_{k}$ is the depreciation rate for capital of type *k*.

Note that the Eqn 5.3.3.4 only defines the availability of capital of type *k* to activity *a* at the start of period *t*. If capital of *k* is made activity specific then the quantity of capital of type *k* in activity *a* at the end of *t* will be the same, but if capital of type *k* is deemed mobile then capital of type k will be reallocated within period *t*. Thus, this allocation method works for either fixed or mobile capital.[30],[31]

However, it is not enough to allocate investment capital to each activity, it is also necessary to define the ownership of new capital by RHG so that the functional distribution of the incomes from capital goods is consistent with savings by domestic institutions. The savings rates for institutions, which can be either endogenous or exogenous variables, and their incomes and income tax rates from the period specific solutions, the value of savings by each institution can be defined and the ownership of capital by each institution in the next period can be defined as

[30] If capital of type *k* is mobile, then is the relative rates of return to capital of type *k* in activity of type *a* in periods *t* and (*t*-1) will be identical and so will be the investment and capital shares in (5.3.3.2).

[31] The alternative approach of defining the total supply of capital of type *k* and then allowing the solution to define its allocation in use produces an identical allocation. The difficulty with implementing this through the total supply is that it limits the factor market clearing options available when using the model.

$$FSI_{ins,k}^{t} = \left( FSI_{ins,k}^{t-1} * \left(1 - deprec_{k}\right)\right) + d\_kap\_ins_{ins,k}^{t} \tag{15.11}$$

where $FSI_{ins,k}^{t}$ is the supply of capital of type *k* by institution (*ins*) and $d\_kap\_ins_{ins,k}^{t}$ is the change in the supply of capital by institution. Note how the depreciation rates are specific to the type of capital and the same as those used when determining the stocks of capital available to activities.

Finally, it is necessary to identify the pattern of capital goods produced in the next period; this is a necessary bridge between the identification of savings at the level of the institution and the composition of capital stocks. The key assumption is that expectations as the pattern of capital good production in the current period for investment in the next period partially adjust to the rates of return to the types of capital, i.e., the pattern of capital goods production is determined at the beginning of each period. Thus

$$inv\_sh\_k_{k}^{t} = cap\_sh\_k_{k}^{t-1} * \left(1 + \mu\_k_{k}^{t} * \left(wf\_rel_{k}^{t-1} - 1\right)\right) \tag{15.12}$$

where $inv\_sh\_k_{k}^{t}$ is the share of investment devoted to capital of type *k*, $cap\_sh\_k_{k}^{t-1}$ is the share of capital of type *k* in the previous period, $wf\_rel_{k}^{t-1}$ is the relative rate of return to capital of type *k* and $\mu\_k_{k}^{t}$ is the partial adjustment (response) parameter for capital of type *k*. Thereafter the base levels of investment volume for the next period can be defined as

$$qinvb_{i}^{t} = \sum_{i,k\$map\_i\_k} inv\_sh\_k_{k} * \sum_{i} qinvb_{i}^{t-1} \tag{15.13}$$

where $qinvb_{i}^{t}$ is the share of investment funds devoted to investment account *i*. Note how there is a unique mapping between investment accounts and capital goods defined by the set $map\_i\_k_{i,k}$.

<u>Business as Usual Scenario (Baseline)</u>

The implementation of the BaU scenario is an experiment in which the shocks are defined as the time paths for a series of exogenous variables with appropriate settings for the macroeconomic closure rules and the factor market clearing conditions. The range of exogenous variables that can be set depends overwhelmingly on the perspective of the analyst and, possibly, the requirements of the policy makers. Typically, these might include forecasts for several macroeconomic aggregates, e.g., GDP, investment, household and government consumption, internal and external balances, etc., plus, possibly, some presumptions about relative rates of activity and/or factor specific productivity growth. There is no unique or correct set of exogenous variables relative rates of productivity growth, so while certain options may be preferred by one analyst, another may well prefer a different set of options.

Having defined the exogenous variables, it is necessary to pair each of these with an endogenous variable; an action that is *de facto* the determination of the appropriate setting of the macroeconomic closure rules. If any relative rates of productivity growth are set then it is necessary to make sure the settings for the derivation of productivity adjustment are set correctly.

The BaU scenario can then be implemented and the results can be saved; note since this is an experiment the process for collecting the results is identical to that for a standard recursive dynamic experiment. It is important that once the BaU scenario has been implemented that the results are analysed **as if** they are a standard experiment, i.e., they need to be checked for logical coherence – the evolution of the economy – and the economic implications. There is no point to running recursive dynamic experiments off a BaU/baseline that is incoherent and/or produces unsustainable economic implications.

Once the BaU scenario has been implemented and checked the model can be configured to run policy simulations. This requires changing the closure rules so that the endogenous variables for the BaU scenario are exogenous variables in the policy experiments and that the exogenous variables for the BaU scenario are endogenous variables in the policy experiments; the requisite data can be accessed form the results or, as in STAGE_RDYN from a specific GDX file that records the required endogenous variables for the BaU scenario. It is important to make sure any relative rates of productivity growth are carried over into the policy experiment settings.

Collection of the Results

The collection of results follows the same principles adopted for the STAGE/GLOBE family of models. All the values for the levels of the variables after each solution are record as parameters indexed on the simulation set, the closure set, the sensitivity analyses set and the time set. Thereafter derived results are calculated using the levels results and base results, in the recursive dynamic case the base results include not only the value of all variables in the first year and all model parameters, but also the BaU results. The BaU results are important because the standard method for presenting recursive dynamic results is in terms of the deviations from the BaU scenario.

## 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.

# Appendices

## A1 Aggregating SAMs

### A1.2 SAMGator

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,[32] 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 version[33] of STAGE with a new aggregation, is to use the SAMgator software (PROVIDE, 2004).

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

[33] 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.

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 and destination data files and then runs GAMS remotely from within the Excel programme[34]. 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).

```
NEWSAM(sp,spp)          = SUM((ss,ssp)
                                  $(MAPSAMAG(sp,ss)
                                  $MAPSAMAG(spp,ssp)),
                                  SAM(ss,ssp)) ;
```

The parameter *SAM*(*ss*,*ssp*) contains the disaggregated database where *ss*[35] is the set that defines the row and columns labels for the SAM. The parameter *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 *MAPSAMAG*(*sp*,*ss*) defines the members of *ss* that aggregated into *sp* by the rows of *SAM*(*ss*,*ssp*), while the set *MAPSAMAG*(*spp*,*ssp*) defines the members of *ssp* that aggregated into *ssp* by the columns of *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.

[34] The GAMS code at the heart of SAMgator is report in the Appendices.
[35] The set *ssp* is an alias for *ss*.

### *3.1.1 Control*

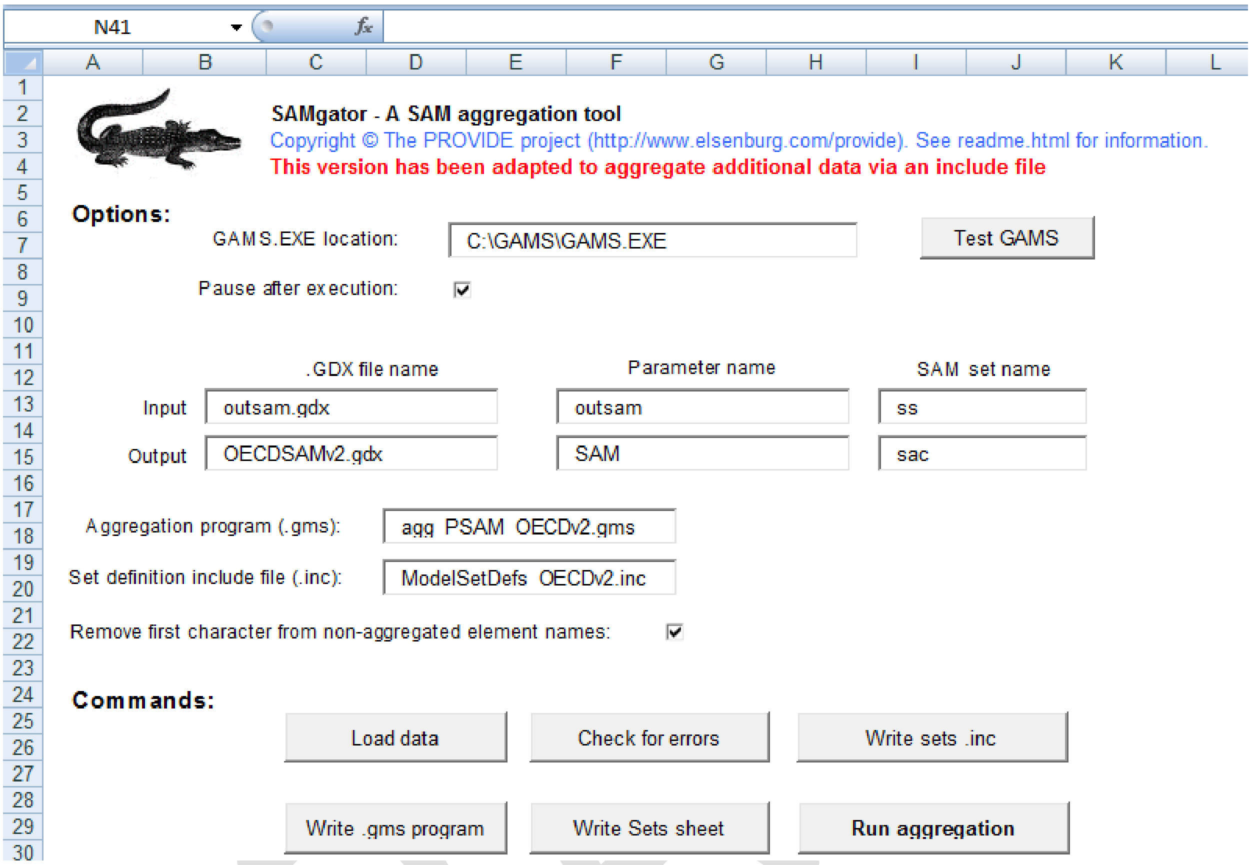
The ‘Control’ worksheet is used to identify the location of GAMS.EXE, input and output file names, check for errors, write output and to run the aggregation programme. A screen shot of the worksheet is given in Figure 3.1.1.1.

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.[36] 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.



[36] 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.

**Figure 3.1.1.1 Control Worksheet for SAMgator**

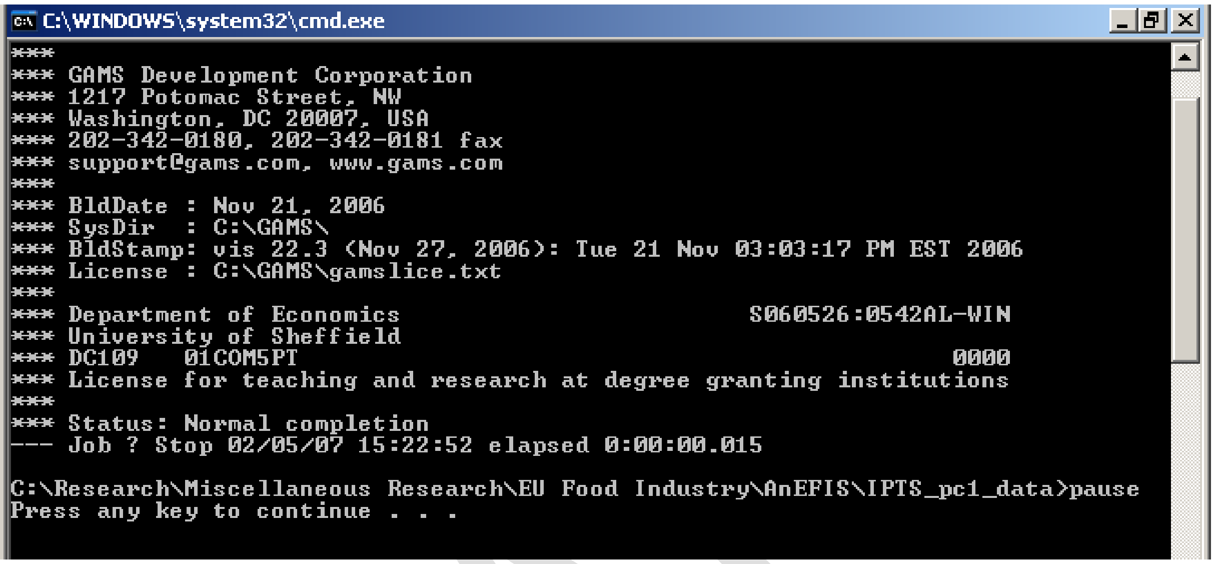



The user should now define the input file, which is required to be a GAMS Data Exchange (GDX) file; in this illustration the file is called `'outsam.gdx'`.[37] Then name the output file; in this illustration `'OECDSAMv2.gdx'`. In both instances it is necessary for the user to identify the parameter within 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.

37 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.

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**

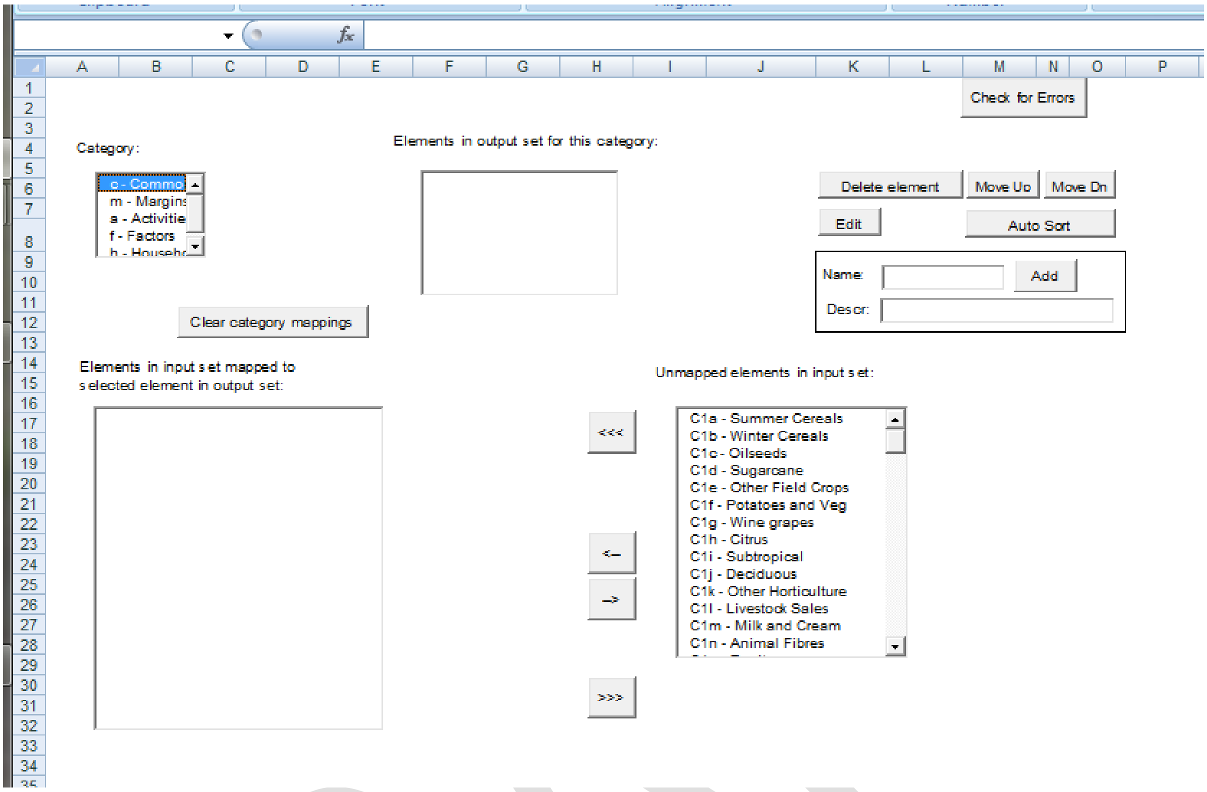



### *3.1.2 MappingConfig*

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' bow (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.[38]
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.



[38] 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.

### 3.1.3 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.

## 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 aggregations[39].
- 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. Te 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.

## A1.2 Using GAMS and Excel

39 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.

## A2. RAS Routine

# A3. Entropy SAM Estimation