A Single Country Computable General Equilibrium Modeling Course Exercises S4

STAGE_t Version: July 2025

cgemod

Addresses for correspondence:

Scott McDonald,
cgemod
url: www.cgemod.org.uk
Email: scott@cgemod.org.uk

The objectives of the project are to develop your ability to (i) set up and implement policy experiments in a small CGE model; and (ii) interpret the results of your policy experiments. There are four elements to the project: recalibration (20%); programming (15%); policy experiment and interpretation (50%); and sensitivity analysis (30%). The percentages indicate the approximate shares of your time to allocate to each task.

Table of Contents

1. Introduction

CGE models are mostly used for policy analysis. A policy change, such as an increase in tax in one sector will have direct impacts on the sector taxed as well as indirect effects on other sectors. Analysis using a general equilibrium model quantifies those effects based on economic theory and data and provides insights about the relative magnitudes of different effects. CGE models are 'laboratories' in which researchers can explore the effects of policy changes and conduct sensitivity analysis to determine how model closure conditions and parameters such as elasticities of substitution impact the results. The models are tools which can provide background information and guidance to policy makers who negotiate *inter alia* trade agreements, implement policies on climate change, determine migration policies, etc.

You must determine your target audience before writing the report. In general terms, reports for government policy advisors\makers will want information about the implications for welfare, consumption, taxation, factor markets etc., but may be less interested in the generic implications of the analyses and results. Reports for applied economics, and related disciplines, may want the same information about the implications but may also seek information about the generic implications of the analyses for economic policy formulation. Contrastingly, reports for academic journals with a stronger focus on economic theory may emphasise contributions that extend the theoretical aspects of GE models and/or extensions to the range of policy issues that can be addressed by GE models.

Throughout the course exercises, you have implemented a variety of policy shocks and progressively extended your abilities to interpret results. So far you have always worked with databases that have been prepared for you, set up in an Excel workbook and tested to ensure it produces results consistent with the learning objectives at each stage. For the course project you will work with data that have been prepared for you, but you will be required to set up the Excel workbook. Also, although the data have been tested, we cannot be certain that the experiments you choose will always generate meaningful solutions; this is a problem that can arise when a CGE model is used with different data and in the process of use identifies issues with the data.

Hence, this part of the course not only synthesizes the coding skills you have developed in the previous exercises, but it will also help develop your ability to set up a database for a model. The emphasis will be on **defining and coding policy experiments and interpreting**

results, particularly in the context of the structure of the economy. The policy simulations, closure assumptions and sensitivity analysis conducted are not prescribed. Instead, you, the researcher, must design the simulation to answer a policy question. The database provided has Social Accounting Matrices (SAMs) for 8 countries.[1]

[1] We can only supply data with the agreement of the developers, and for which we have the time to customise. We plan to increase the number of SAMs, but …….

2. Setup

The setup process for this exercise should be familiar: establish a working directory and New Project and populate the directory from the GAMS User Library.

1. Create a subdirectory `project` in the directory `C:\cgemod\stg_t\project`.
2. Create a New Project using GAMS Studio.
3. Open the GAMS User Library `Single CGE Course Library` and Download `stg_t_10` from the GAMS User Library (SeqNr: 04)
4. Unzip the file `stg_t_4.zip` in your working directory.
5. It is assumed that you will use Save and Restart to run the experiments.

WE ARE DELIBERATLEY DUPLICATING THE FILES USED. THIS IS BECAUSE YOU MAY, LEGITIMATELY, HAVE MADE CHANGES WHILE USING THESE FILES DURING MODULE S2, THAT MAY CAUSE THE RESULTS TO CHANGE.

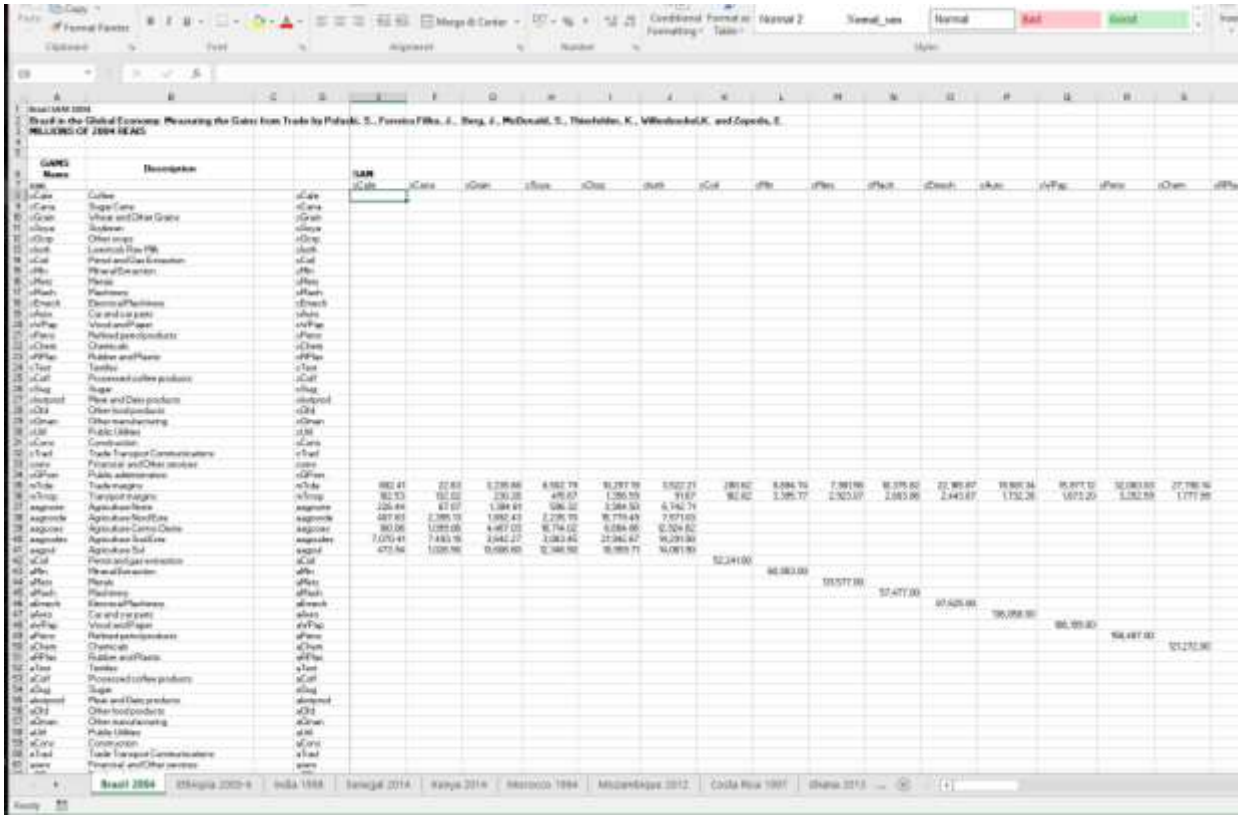
3. Backstory and Proposal

The data and set information for all the project countries are stored in an Excel workbook – `stg_t_4_data.xlsx`, that will have been downloaded into your working directory (Figure S4.1). Each country has a single worksheet that has three pieces of information

1. the labels for the set ***sac***;
2. the descriptions for the set ***sac***; and
3. the ***SAM*** transactions matrix.

Inspect the data and decide on your choice of country.

Figure S4.1 Project Database



‘Backstory’ and Proposal

Develop a summary of the economic structure and challenges facing your chosen country; it should reflect the country around the date of the SAM, not those in the present time. This summary, maximum of two pages, should provide the basis for the development of your proposed policy experiments.

For your proposal

1. Identify a policy issue important to the country, at the time of the SAM, you are analyzing.
2. Write a one-page proposal for the policy experiments you propose to implement. This proposal should summarise the planned experiments and model settings (macroeconomic closures and market clearing conditions). **Explain why** you have made these choices.

This step, and the associated notes, is a brief version of a process you might undertake when conceptualising a project. It is very easy to waste large amounts of time when conducting a study because the groundwork has not been laid.

4. Setting up the Project Model's Database

The data and set information in the worksheet you have chosen from the Excel workbook – `stg_t_4_data.xlsx` – has all the data you need for your project. You need to use these data to develop your project's (Excel) database.

Create the Excel Database

The first task you need to complete is setting up the Excel database for your chosen country; this can be tricky, and it is easy to make errors; many of the common typing errors can be avoided by using copy and paste (or Paste/Special/Values) to avoid typing errors. **NB**: there are series of Hints and Help comments below to help you; you may like to read these before going further.

You should open the Excel file `stg_t_data_sam_3_2.xlsx` and save it with a new name, e.g., `stg_t_data_***.xlsx`. What you will then need to do is replace the data for South Africa with the data for your chosen country. The following steps need to be followed.

1. sets sheet:
    a. copy the labels and descriptions from `stg_t_4_data.xlsx` into columns A and B (make sure that the first label is in cell A6);
    b. populate the subsets of *sac*, *c*, *a*, *f* and *g*;
    c. note that the set *ff* must include the element *flab*;
    d. note that the set *sac* provided contains elements that may not be in the SAM but are needed for the model.
2. maps sheet:
    a. populate the mapping set *map_va_ff*;
    b. populate the mapping set *map_fagg_ff*.
3. comelast sheet:
    a. populate the labels for the set *c*;
    b. populate the descriptions for the set *c*;
    c. determine the values for the elasticities to be used for your project (a series of identical elasticities for each *c* have been used in the template; you need to make decisions about the appropriate values).

4. actelast sheet:
    a. populate the labels for the set *a*;
    b. populate the descriptions for the set *a*;
    c. determine the values for the elasticities to be used for your project (a series of identical elasticities for each *a* have been used in the template; you need to make decisions about the appropriate values).
5. elastf sheet:
    a. populate the labels for the set *a*;
    b. determine the values for the elasticities to be used for your project (a series of identical elasticities for each *a* have been used in the template; you need to make decisions about the appropriate values)
6. frischelast sheet:
    a. populate the labels for the set *h*;
    b. populate the descriptions for the set *h*;
    c. determine the values for the elasticities to be used for your project (the frisch elasticities for each *h* have been set equal to minus 1, i.e., consistent with Cobb Douglas utility functions for each household). You need to make decisions about the appropriate values.
7. hoelast sheet:
    a. populate the labels for the set *c*;
    b. populate the descriptions for the set *c*;
    c. populate the labels for the set *h*;
    d. determine the values for the elasticities to be used for your project (the income elasticities for each *h* and each *c* have been set equal to 1, i.e., consistent with Cobb Douglas utility functions for each household). You need to make decisions about the appropriate values.
8. popn sheet: leave blank;
9. ue_rate sheet:
    a. populate the labels for the set *f* (not *ff*);
    b. set the un/underemployment rates for each factor to zero.
10. SAM sheet: copy and paste from `ModO10_data.xlsx`;
11. stab_sets sheet: do NOT change;

12. controls sheet: do NOT change;
13. factuse sheet: leave blank or copy the transaction values from the SAM. (HINT: in the file `stg_t_data_load.inc` search for `factuse` and see how `factuse` can be assigned if it is empty in the worksheet `factuse`.)
14. factins sheet: this is a tricky worksheet to assign. A worked example is provided in the `factins` worksheet of the workbook `stg_t_data_sam_3_2.xlsx`. (See Appendix 1 for an explanation of the derivation of the `factins` worksheet.)

Testing the Model

Run the model. You are highly likely to have errors and on the first run there could be a very large number (although if you have configured Studio to stop after 5 to 10 errors you will not see them all – this is not a problem). Do not be disheartened, this is normal and with some care can be rapidly changed. Most of these apparent errors are cumulative, i.e., one simple error early in the process can generate a very large number of subsequent errors. The process of resolving these errors can be speeded up by using the stop two feature in GAMS

1. use the command `$stop` in the code. (This can be done anywhere in the code. The places that may initially prove helpful are: immediately after the $GDXIN command to load data from the `data_in.gdx` file, at the end of the `Data Entry` section, after the `parmcalib.inc` file and after the `stg_int_samchk.inc`.
2. It may also be useful to use `$stop` at intermediate points in the files to help pinpoint the error.

The most common errors encountered are likely to be

1. a failure to read a file,
2. failures of domain checks, and
3. failures to convert data in Excel into the `data_in.gdx` file.

Hints and help for some common problems are provided in the next section.

BE SYSTEMATIC; MAKE ONE CHANGE AT A TIME; DO NOT CHANGE THE LAYOUT SHEET; DO NOT PANIC AND START TO MAKE CHANGES IN THE MODEL FILES.

Once the model appears to be running correctly, carry out all the standard tests. There are SEVEN checks **all** of which need to be passed. You should know all these checks by now, so we do not repeat the details here; look at previous instructions if you are uncertain.

Hints and Help with Common Problems

This list of common problems and hints as to solutions is not complete; rather it is list of problems we have encountered at various times (as new ones are discovered we may revise these 'hints'). The most important skill needed to resolve these problems is learning to read the information provided by GAMS; it is not always obvious what the information means, and it can be the case that the error message is generated by an error earlier in the code. **You must also learn to read the log file**; this is especially important when converting Excel data to gdx.

Failure to generate `data_in.gdx` file

1. Incorrect file names: make sure the file names are correct and that they are in the correct directory.
2. Duplicate entries for symbols: look at the **log** file, wherein duplicates will be identified. (We have added the 'trace' option to the GDXXRW call statement in file `stg_t_data_load.inc` file to increase the amount of information produced in the log file. The default value is ONE; the amount of information increases with the value. The choice of THREE is a compromise that is usually adequate but sometime FOUR is helpful)

```
Reading Slice: A1:BQ71
Par  sam            2      sam                  B5:BQ71       A5:A71        B4:BQ4
**** There were at least 10 duplicate entries for symbol sam
 1: cagric,total
 2: cmins,total
 3: cfood,total
 4: ctext,total
 5: cpetchem,total
 6: cmprod,total
 7: cmach,total
 8: cveh,total
 9: comanu,total
10: cutil,total
Remaining duplicate entries skipped (use trace>=4 to get more information)
Loading sheet factuse
```

3. Bad data range: look in the **log** file wherein the bad data range is identified (typically caused by either entering data in a worksheet that is not consistent with the range specified in the LAYOUT sheet OR changing the LAYOUT sheet wrongly).

```
par=elastx Rng=actelast!B5 rdim=1 cdim=1
par=elastf Rng=elastf!B5 rdim=1 cdim=1
par=elasty Rng=hoelast!B5 rdim=1 cdim=1
par=elastmu Rng=frischelast!B5 rdim=1 cdim=1
**** Bad data range for symbol mcons : control!A7 (Sheet name not found control)
Total time = 1797 Ms
--- .stg_int_data_load_prpj.inc(147) 2 Mb
*** GDXIN failed C:\cgemod\stg_int_proj\data_in.gdx
*** Msg: No such file or directory
```

4. Error with investment-savings sets: the set `i(sac)` should not include `dstoc` (stock changes), the set `in(i)` should not include `i_s` (savings). A failure to assign these sets correctly will typically result in a failure of the SAM checks. (These set assignments are used for recursive dynamics to allow for a richer specification of types of productive capital and investment decisions.)

Failure to load expected data from `data_in.gdx` file

These errors are of various types:

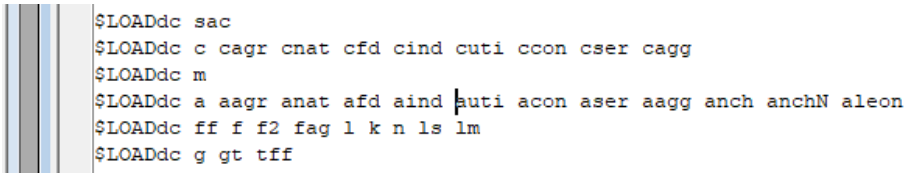
1. a parameter and/or elasticity that appears in the Excel workbook does not appear in the GAMS programme:
    a. data are not entered in the Excel workbook where the LAYOUT sheet tells GDX to expect the data;
    b. the wrong labels have been used in the Excel workbook, e.g., using the labels for the set *a* instead of set *c*;
    c. the LAYOUT sheet has been changed;
2. the SAM does not balance despite apparently being balanced in Excel (all those in the project database do balance)
    a. one or more elements are not included in the set *sac* (hint: subtract the SAM from the model from the SAM in Excel – remember to scale the SAM from Excel by the same scaling factor (`samscal`) as used in the model);
3. the data used in the model files differs from that in the `data_in.gdx` file (where the gdx data are consistent with the Excel data)
    a. check that there have been no code changes that mean the wrong data are assigned to the parameter for which errors are reported;

b. the LAYOUT sheet has been changed.

Failure to pass domain checks

These failures can have multiple possible causes, and they can be difficult to find.

1. Element is a subset not included in the parent set, e.g., *flab* included in *ff* but not in *sac*, *ccrop* in *cagr* but not in *c*, etc. When double clicking on the error message, the error will be identified by the cursory flashing immediately before the NEXT (sub)set to be read (see screenshot below). (Note: elements may appear in a parent set, e.g., *sac*, but not in subsets and this should not cause a failure of a domain check.). The main causes are:
    a. an element has been omitted in the parent set;
    c. an element's label has been mistyped – copy and paste helps reduce this type of error;
    d. the LAYOUT sheet has been changed;



2. Labels for the elements in parameter tables are not those expected by GAMS
    a. check that there was not an error when changing the row and/or column labels (these can be either typing errors or simply forgetting to update the labels for a table);
    b. the LAYOUT sheet has been changed;
3. Labels for the set elements in the maps are wrong
    a. check that there was not an error when changing the sets in each map (these can be either typing errors or simply forgetting to update the labels for a table);
    b. check that *map_va_ff* and *map_fagg_ff* have been extended to cover all relevant factors;
    c. check that the LAYOUT sheet has not been changed.

Undefined Elasticity Values

1. CES/CET elasticities: these functions are undefined for elasticity values of ONE. (Avoid elasticities too 'close' to ONE – avoid range c 0.85 to c 1.15).
2. Frisch Elasticities: values must be negative and less than or equal to MINUS ONE.

5. Policy Experiments and Interpretation[2]

You are an economic consultant hired to analyse policy issues that are relevant and current to your chosen country.

Tasks

1. Identify a relevant and current policy issue and justify your decision.
2. Define and code appropriate policy experiments.
3. Specify appropriate macroeconomic closure conditions.
4. Specify appropriate factor market clearing conditions.
5. Interpret the economic and policy implications from the results of the experiments.

Develop the experiment file

1. Develop the GAMS code needed to run the agreed upon experiments using a `sim` loop.
2. Ensure that all the experiments are implementing the intended shocks and producing coherent solutions. The code can be developed from `stg_t_exp*.inc` files used and developed for use with the previous modules (S2 and S3 in particular, but you may want to reuse code developed for earlier modules). (Copy the required files into your project directory using Windows (file) Explorer to avoid making errors with the paths.)

[2] As is common in economics; the questions often do not change, but the answers do!!

6. Sensitivity Analysis

The sensitivity of the results to assumptions embedded within the model needs to be examined.

Tasks

1. Devise a series of policy experiments that systematically examine the sensitivity of your policy experiment results to these assumptions.
2. Explain carefully how you decided upon the policy experiments used to conduct the sensitivity analyses.
3. Comment critically on the results of the sensitivity analyses

7. Project Report

In all cases you should use tables, diagrams etc., to report your results, and then comment on the policy implications of your results.

Remember the purpose of this project is to communicate information to your target audience

Appendix 1 'factins' worksheet

The 'factins' worksheet arises from the desire/need for a richer specification of the endowments owned by representative household groups (RHG), but doing so requires accounting for endowments owned by all institutions. The 'factins' worksheet is key for the Factor Supply by Institutions (FSI) variable in the model. This variable was introduced to address a series of issues, *inter alia*

1. changes in the supply of surplus labour to activities in the SNA production boundary leading to changes in the functional distribution of factor incomes;
2. relaxing the strong separability assumption to account for Home Production for Home Consumption (HPHC), which restricts the availability of RHG endowments for use by other activities in the SNA production boundary;
3. including labour-leisure trade-offs in the model so that RHGs can change the supply of endowments for use by activities in the SNA production boundary;
4. including economic activities that are outside of the SNA production boundary in the model so that RHGs can change the supply of endowments for use by activities inside and outside the SNA production boundary;
5. recursive dynamics need to track the evolution of endowments – labour, human capital, physical capital – over time to evaluate the implications for welfare and income distribution (the equivalent parameter to FSI was used in earlier versions of recursive dynamic code, but this meant losing other aspects captured by FSI as a variable).

Hence, FSI has been deemed useful and therefore the `factins(insa,f)` matrix needs to be populated. In an ideal world data adequate to estimate this matrix may be available from surveys and censuses, but this is rarely the case although such data may still be useful.

The pragmatic method outlined here applies to applications of the STAGE model that are limited to working inside the SNA production boundary.

Known Information

The following data are known from the SAM and factuse matrices

1. the payments received by institutions from the factor accounts,

2. the factor quantities, in physical units or value quantities, used by activities inside the SNA production boundary.

Given these data the following can be derived

1. The total quantities of each type of factor use inside the SNA production boundary, i.e., $FS_f = \sum_a factuse_{f,a}$
2. The shares of each factor payments made to each institution, i.e., $sh\_f\_exp_{insa,f} = SAM_{insa,f} / \sum_{insa} SAM_{insa,f}$

If it is assumed that factor ownership is in the same proportions as factor payments to institutions by factors, then

$$factins_{insa,f} = FS_f * sh\_f\_exp_{insa,f}$$