A Single Country Computable General Equilibrium Modeling Course Exercises S1

STAGE_t Version: Sept 2025

cgemod

Addresses for correspondence:

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

This document is subject to ongoing revisions. If you find any errors, please let us know so that we can correct them ASAP. If you have suggestions as to how the exercises can be improved, please send them to the email address above.

Table of Contents

Introduction

These exercises use the STAGE_t model, which is a member of the STAGE family of models. STAGE_t has been compiled for training purposes. It is licensed under the Creative Commons BY (v4) license (details of the license are included with the model code).

The exercises for module S1, S2 and S3 use a SAM for South Africa, which has relatively rich details about household, factors and taxes. It is an aggregated version of the full SAM developed as part of the PROVIDE project for the government of South Africa. Module S4, the course project, has c 10 SAMs from which the participants can choose. We cannot guarantee that the model will work with other SAMs, although experience has shown that adapting the model to other compliant SAMs is often relatively straightforward.

There are six exercises in module S1, seven in module S2, two in module S3 and one in module S4. The exercises for modules S1 and S2 are designed to develop the technical skills required to build comprehensive experiments. Through these exercises you will progressively build up your own core experiment file; this will provide you with a base for the course project (module S4). The experiment file you build will, eventually, include code for implementing shocks, conducting sensitivity analyses with respect to choices about closure and factor market clearing settings and elasticities, and post simulation analyses of the results. There will be some guided interpretation of the results as part of the exercises for modules S1 and S2.

Module S3 concentrates on the interpretation of results. There are two exercises with predefined experiments. The experiments are designed to develop your skill at interpreting results, i.e., explaining why the results were produced, and both experiments produce results that may not be immediately explicable. Each module has a mini project.

Module S4 is the course project. You will choose one of the available databases[1], design and implement experiments, interpret the results and write up your analyses.

1 The databases have been checked for conformity with the stg_t model. Within the course it is not possible to use participant specific databases.

Setting up the Course Library

These instructions are for GAMS 51.4 and Studio 1.22.2. Earlier or later versions of GAMS and GAMS Studio may operate slightly differently.

All work for the course should be kept in the directory `C:\cgemod` with appropriate subdirectories as described in the exercises. During this course, the files we will use are in a User Library, which you will download from the Moodle site.

The library for the single country CGE modules contains zipped directories that will need to be unpacked AFTER they are downloaded from the User Library – `single_lib`.

The models for this module, and the module for the global model, use a file structure that requires most files to be in sub directories. This reflects the fact that by the end of the Practical CGE course working directories were starting to contain large numbers of files, and hence finding files in Windows Explorer was becoming less easy. The transition to a system of nested sub directories is simply a move to a filing system deemed less likely to confuse.

NB: the single_lib directory contains sub directories. These sub-directories must NOT be unpacked.

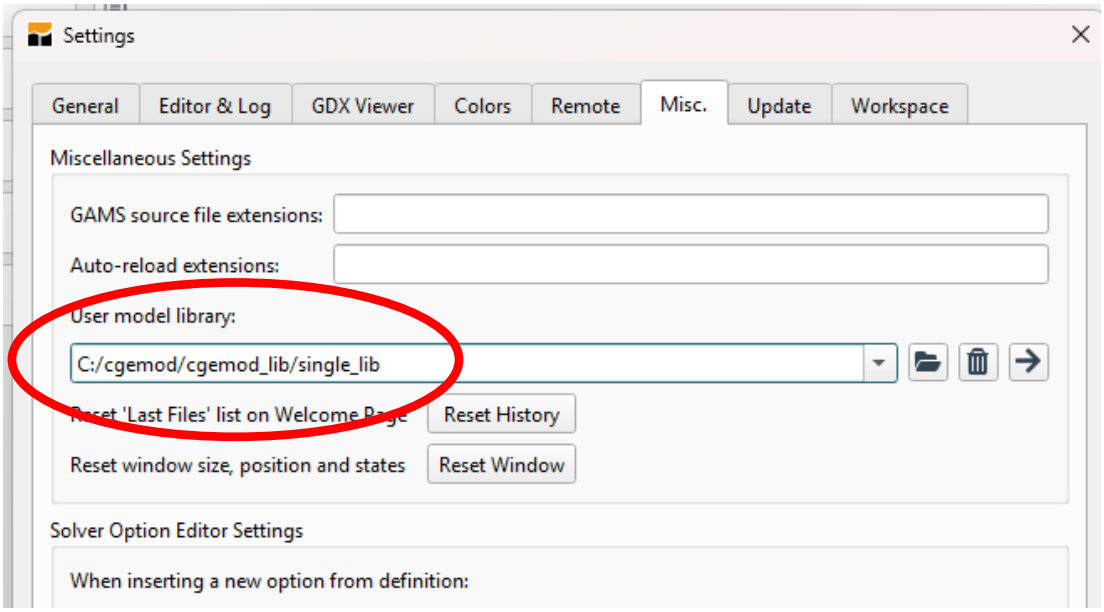
The first stage is to establish the User Library that will be used for the four modules of the Single Country CGE course and tell GAMS Studio where the User Library will be found (see section 11 of 'Introduction to GAMS and GAMS Studio'). Carry out the following actions

1. Then download the zip file `single_lib.zip` from the website (it is one of the files in the folder of downloads for Module S1) into your downloads directory (`C:\cgemod\downloads`).
2. Unzip the contents of the file `single_lib.zip` into the library directory `C:\cgemod\cgemod_lib\single_lib`. WinZip may by default unzip these files into a directory, often called `single`. If so, you will need to copy these files and paste them into your library directory `C:\cgemod\cgemod_lib\single_lib`.[2]

[2] The default settings in WinZip may make you do this as a two-stage process. In WinZip 'classic view' you can avoid the two-step process. It is called progress!!!!

3. **The files in `single_lib` that are in sub directories must NOT be extracted from the sub directories**.
4. Open Studio settings - `File>Settings` or `F7` or the `Settings Wheel` on the toolbar and go to the `Misc` tab.
5. Change the path for the User Model library from something like `C:\Users\...\Documents\GAMS\modellibs` (the exact path will depend on the settings on your PC) to `C:\cgemod\cgemod_lib\singel_lib`.
6. Close the Studio `Settings` window; remember to click `Apply` and `OK` at the bottom of the `Misc` tab.

Figure S1:1 Setting the Path for single_lib

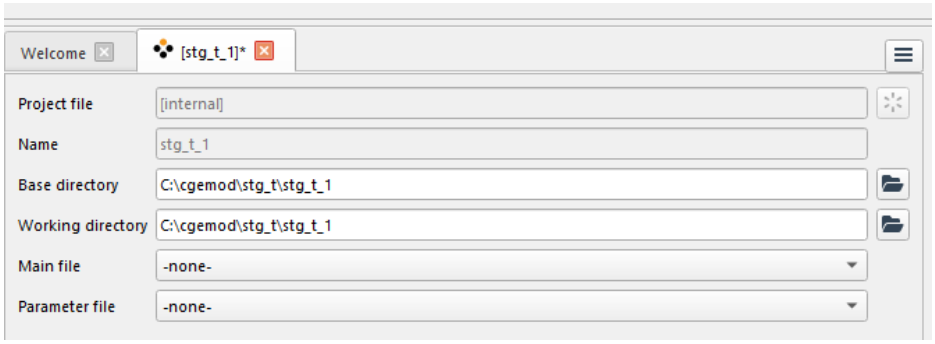


The second stage is to establish the master directory for the STAGE_t models and then create a first project for the exercises.

1. in Windows Explorer create the directory for this course, i.e., `C:\cgemod\single`,
2. in Windows Explorer create the directory for these exercises, i.e., `C:\cgemod\single\stg_t\stg_t1`,
3. open the Settings (`F7`)
4. on the General tab make sure that the option '`Open file in current project by default`' is selected (this should already be the case),

5. choose `File>New Project` and this opens the project options in the edit viewer,
6. the default name for the project is `newProject`,
7. rename the project name as `stg_t1` (the name in Project Explorer will not change until the project is saved),
8. set the Working directory to the new project directory, e.g., `C:\cgemod\single\stage\stg_t1`, using the BROWSE button,
9. note the Base directory will default to the same name as the Working directory,
10. choose `File>Save` **– DO NOT USE SAVE AS**[3]
11. now choose `File>New` (see below) that will open a Windows Explorer window with defaults File name, `new***.gms`. (this step together with step 3 above is necessary to ensure that Studio is directed to use the new project's directory),
12. click `SAVE` and `new**.gms` will appear in the `clmod1` project in the Project Explorer.
13. You now have a New Project that is in a working directory that is effectively empty. This method will be used repeatedly although he instructions to create a New Project will be truncated.

Figure S1:2 Setting the Project Directory



The working directory can now be populated from the User Model Library (`single_lib`).

[3] As of Studio version 1.22.2 using Save As produces an error.

1. In the project `stg_t1` choose `F6` and in the `Model Library Explorer` find the library `Single Country CGE Course`.
2. Open item 01 (`stg_t_1`) in the `Single Country CGE Course` library.
3. The `stg_t.gms` model will now be displayed in the editor window and be listed in the Project Explorer as being in the project `stg_t1`.
4. If you right-click on the project name and select `Open Location`, you will see 5 files – 2 `gms` files, 1 `dat` file and 1 `zip` file - have been downloaded to the directory `cgemod\stg_t\stg_t_1`, one of which is a zipped archive (Figure S1.3).
5. You can delete the file `new*.gms`.
6. Open the directory `cgemod\stg_t\stg_t_1` and unzip the archive `stg_t_1.zip`. This will add 6 sub directories to the project's directory (see Figure S1.4). **NB**: when using Windows Explorer to extract the files they can be put into a sub directory: if this happens you need to move the sub directory to the working directory. **NB: the files in each sub directory must remain in the sub directory.**
7. You also needed to change the file extension for `clean_stg_t.dat` to `clean_stg_t.bat` (this is a batch file used to 'clean' working directories but will be treated as an executable if downloaded with the `bat` file extension).
8. Studio can now be used to work with the STAGE_t model.
9. Throughout the course we will work with the model user library to access files needed for each exercise. The instructions for setting up modules S2, S3 and S4 will be less detailed.
10. **If you work directly from the library directory, you will end up using the wrong files at some point in the course and thereby corrupt the library. Never work directly from the library directory.**

11. RESIST THE TEMPTATION TO EXPLORE THE CONTENTS OF THE LIBRARY; THIS WAY LIES CONFUSION. YOU GET TO SEE AND USE ALL THE FILES.

Figure S1:3 Project Directory for stg_t_1

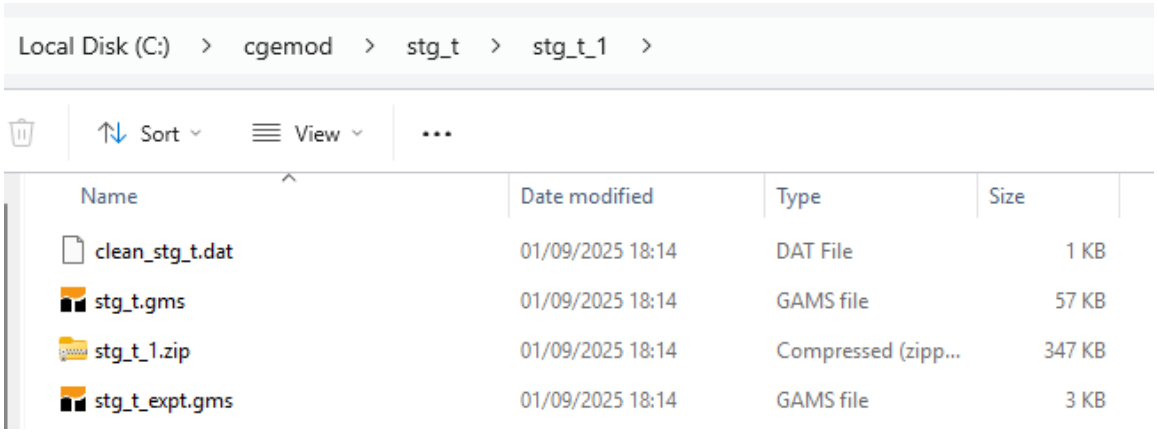
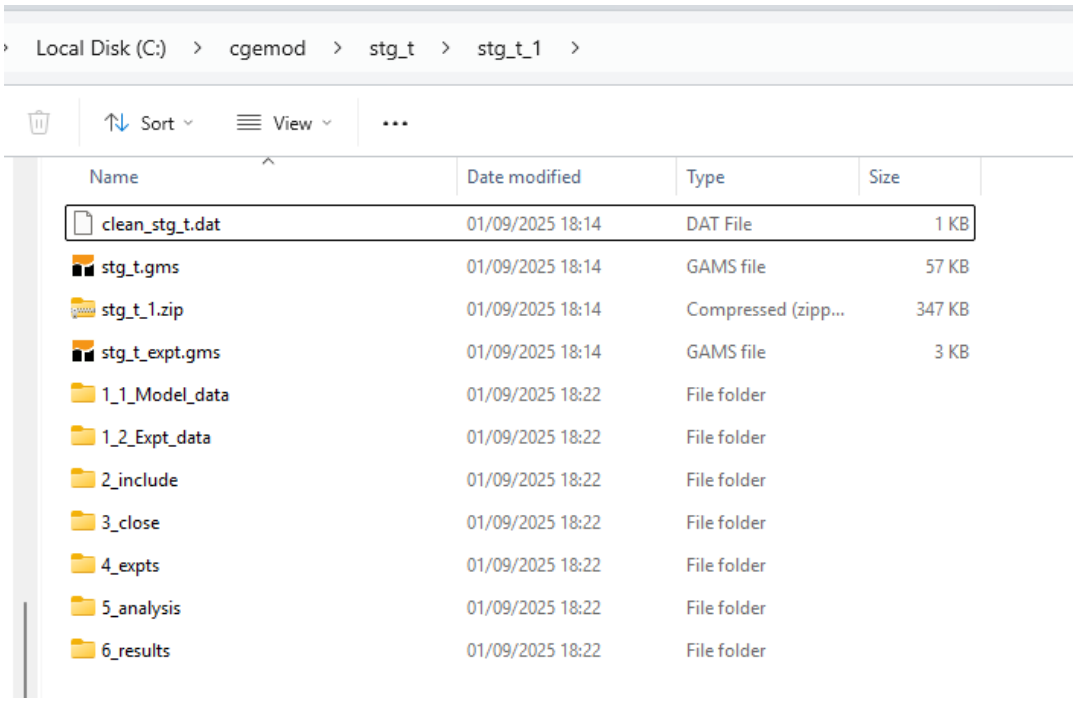


Figure S1:4 Project Directory for stg_t_1 - unzipped



Ex S1:1 Review the Model stg_t Model

This exercise is concerned with reviewing the stg_t model and familiarising yourself with the directory structure. This exercise operates solely in comparative static mode.

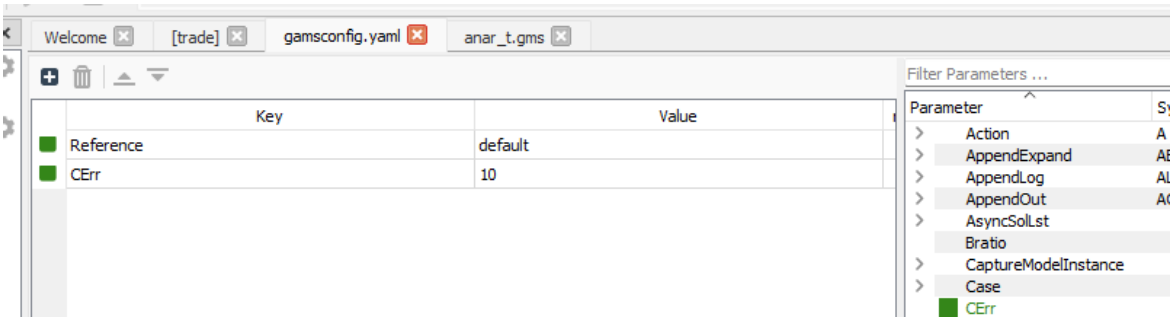
GAMS Configuration

It will be helpful if GAMS is configured to carry out two actions automatically when a programme is run (you should already be using this configuration, so this is a reminder).

1. In GAMS Studio menu select `GAMS>Default GAMS Configuration` (see Figure S1.4).
2. The course assumes you select one item (`CErr`) from the right-hand frame in the editor window and drag then to the left-hand frame.
3. The `CErr` instruction automatically stops a programme when a specified number of syntax errors have been identified; we suggest setting the entry in the Value column to 10 or less – since once one syntax error occurs others will inevitably follow it is good practice to resolve errors in the order they occur.

We assume that participants will make extensive use of the reference file. The use of reference files is covered in the course 'Introduction to GAMS/GAMS Studio (http://www.cgemod.org.uk/gams_studio.html).

Figure S1:4 GAMS Default Configuration



Running the Model

The first task is to run the model and use the output to explore the model.

1. Run the model `stg_t.gms`, with GDX and RF creation (`F10`), to ensure that it solves correctly.
2. Use the reference file (`stg_t.ref`) and the GDX (`stg_t.gdx`) and Reference (`stg_t.ref`) to explore the contents of the model files

Checking the Model

All the checks used to verify the model has run correctly are detailed in this section. You should get into the habit of checking that the model has correctly run every time you start a session and every time you make changes to the data and/or model. Once used to making the checks they only take a couple of minutes, whereas hours can be wasted trying to work out why an 'incorrect' model has produced unexpected results.

IN FUTURE EXERCISES YOU WILL BE ADVISED TO CHECK THE MODEL, BUT DETAILS ABOUT THE CHECKS WILL NOT BE INCLUDED.

1. Check that the value for `VAR WALRAS` is zero.
   This check can be done from the `lst` file or `stg_t.gdx`. In the `lst` file search for '`Var Walras`' or use the index for `SolVAR`. In `stg_t.gdx` search for '`Walras`'. If `WALRAS` does not equal zero (less than E-08, and ideally even smaller), then the model fails to satisfy Walras's law, and some transactions are not accounted for.
2. Check that the basic prices (*PE*, *PD*, *PM*) are equal to one.
   This check can be done from the `lst` file or `stg_t.gdx`. In the `lst` file search for '`Var PE`' or use the index for `SolVAR`. In `stg_t.gdx` search for '`PE`'. (Do the same for `PD` and `PM`.) If values are not one, then the standard price normalisation rule has not been applied correctly.
3. Check that all entries in `ASAM1CHK` are equal to one.
   This checks for differences between the macro-SAM from the database for the model and the macro-SAM calculated from the values of the calibrated parameters and variables. They should be identical and therefore the ratios should all equal one.
4. Check that all entries in `SAM1CHK` are equal to one.
   This checks for differences between the micro-SAM from the database for the model and the micro-SAM calculated from the values of the calibrated parameters and variables. They should be identical and therefore the ratios should all equal one

5. Check that all entries in `ASAM2CHK` are equal to one.

   This checks the ratio of the entries in the macro-SAM calculated from the values of the calibrated parameters and the macro-SAM calculated from the values of the parameters and variables in the model. They should all equal one.

6. Check that all entries in `SAM2CHK` are equal to one.

   This checks the ratio of the entries in the micro-SAM calculated from the values of the calibrated parameters and the micro-SAM calculated from the values of the parameters and variables in the model. They should all equal one.

7. Check the `LHS` values. Search from the beginning for the first string `LHS` in the `lst` file; then search from the cursor for the string `'***'`.

   In the output, GAMS reports the equations after all equations are organised so that all endogenous terms are on the `RHS` and hence the `LHS` should equal zero **or** a known exogenous value. Those that do not satisfy that condition are marked with '***'. In the `lst` file GAMS reports the values of the `LHS`: the search for `LHS` finds the first such entry, and second search finds the equations with incorrect `LHS` values.

8. For additional certainty search for `infes`. (This will return "`Search string 'infes' not found`" if the model is calibrated correctly.)

9. Check that the model is homogenous of degree 0: in the workbook `stg_t_data_sam.xlsx` go to the worksheet 'controls' and change the entry in cell B8 to 2. We are doubling the numéraire; what should this do?

10. Rerun the program and find `ASAM2CHK` and `SAM2CHK`; what are their values? What should these values be?

11. Check the results for var `WALRAS`, `LHS` and `infes`.

12. Restore prices to original values, in the controls sheet, change the numéraire check back to 1 and save the file. Verify that the values of `ASAM2CHK` and `SAM2CHK` all equal 1.

Exploring the Input

The identity of the Excel data file used to calibrate the model is supplied to the model code using the GAMS option $SETGLOBAL. This GAMS option allows the substitution of a

global compile-time variable with a user defined string of text. This is implemented when the model is compiled. The syntax is

```
$SETGLOBAL  VARNAME   text
```

When the programme is compiled, the text is substituted for %VARNAME%, **NB**: the use of % signs to bracket VARNAME. The Excel file name is the text string, less the .xlsx suffix, and the VARNAME is 'mod_data'. If the VARNAME is changed an error will occur and the dataload file will need editing, i.e., the purpose for using $SETGLOBAL will be lost.

When searching for $SETGLOBAL in the GAMS documentation you need to include the $ sign.

The Excel file used by the model is identified by the $SETGLOBAL assignment on line 68 of the file stg_t.gms; the Excel file is in the sub directory 1_1_Model_data. Open this file and explore the contents; it is like the Excel file used for smod_t/2/3.gms in modules O5 and O6.

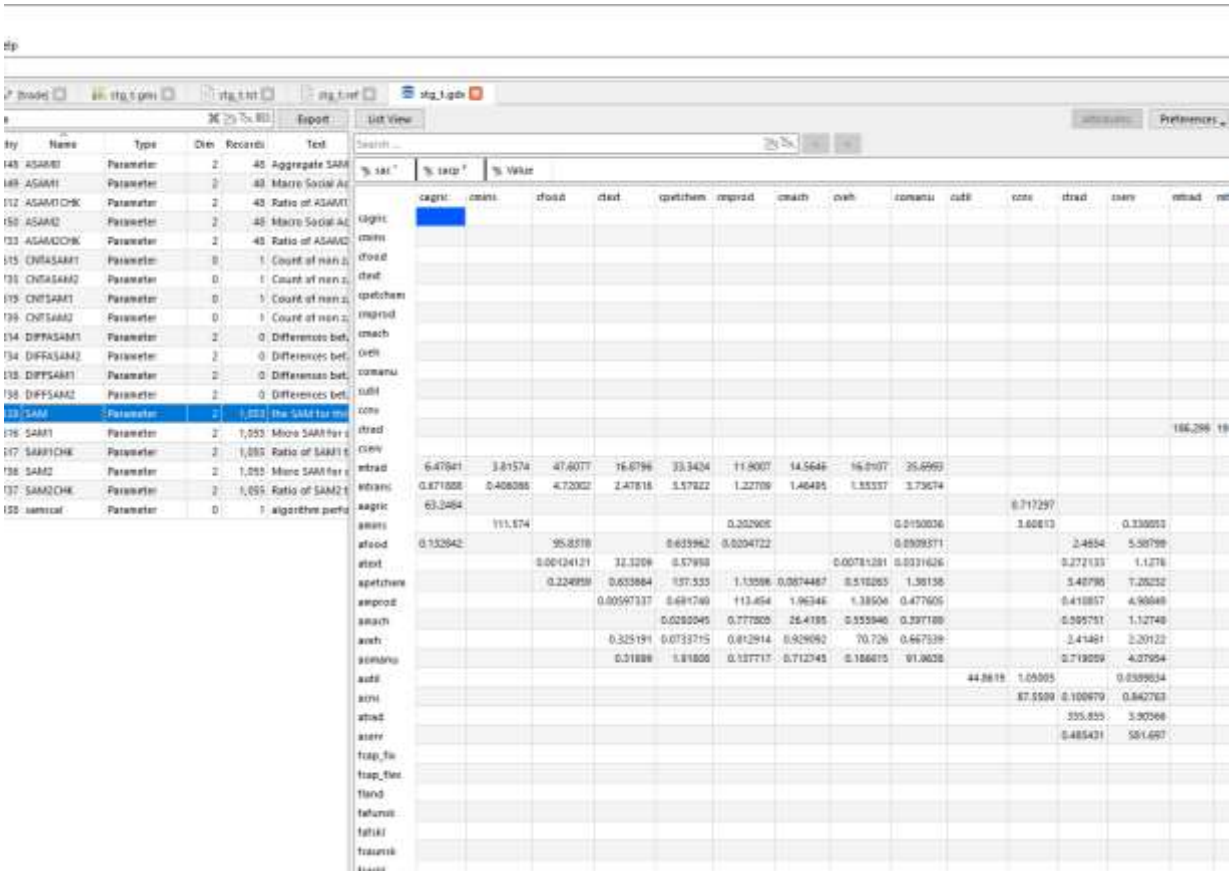
The load instructions are embedded in an INCLUDE file – stg_t_data_load.inc. The instruction for this is on line 442 of the file stg_t.gms but is easier to access using the reference file stg_t.ref and filtering on Files Used. Open the file stg_t_data_load.inc and search for GDXXRW. This is the start of the block where data are entered into the programme. One advantage of $SETGLOBAL is that the data used by the model can be changed without needing to modify a file embedded within the model code, which is especially useful if data are entered in more than one location (something we discourage).

Exploring the Output

This set of exercises is intended to help you learn the model.

1. The file stg_t.gms will be open on your screen in GAMS Studio. It is wise to rerun the model using F10 (Run with GDX and RF Creation). This will generate a reference file, stg_t.ref and a GDX file with all model information, stg_t.gdx.
2. What files were used running this programme? HINT: open the file stg_t.ref and select the File Used tab.

3. Examine the content of the directory `C:\cgemod\stg_t\stg_t_1`. Open the file `1_1_Model Data\stg_t_data_sam.xlsx` and look at the various worksheets and answer the questions below for each worksheet:
   a. `Layout` – where are the members of set *f* found?
   b. `Sets` – who are the members of set *cagg*?
   c. `nest_va and nest_fagg`– what do the sets *map_va_f(ff,)* and *map_fagg_ff(ff,ff,a)* do?
   d. `Controls`– what does *samscal* do?
   e. Go back to GAMS Studio and open the file `stg_t.gdx`; your screen will look as follows if you select the SAM and choose `Table View`. You should find the same data in the workbook `stg_t_data_sam.xlsx`. Why do the numbers differ in terms of decimal places?



We will look at more parts of the data as the course progresses.

4. To assist with navigating the files in a GAMS programme it is useful to use the reference file which provides a comprehensive indexing of the contents of all the files used when running a GAMS programme. You created a reference file when you ran the model with `Reference` set to default in the GAMS Configuration (if in doubt see the Guide to GAMS Studio). Open the reference file, `stg_t.ref`. A reference file has tabs for `Symbols, Files Used, Sets, Parameters, Variables, Equations, Models and Unused`. For Sets, Parameters, Variables, Equations and Models each element is `Id(entified)` and then where they are `Declared, Defined, Assigned, Ref(erenced), Control` and `Impl-Asn` is detailed. Thus, to find out where the variable `DTAX` is Declared go to the tab `Variables` and double click on `DTAX` in the second column (`Name`); this will open a frame that identifies all occurrence of DTAX in the model. If you click on the highlighted row under the label `Declared in`, you will be taken to the point in the programme where the variable `DTAX` is declared. If you then click on the highlighted row under the label `Assigned in` you will be taken to where `DTAX` is assigned. **NOTE** how Studio automatically opens the file (`stg_t_varinit.inc`) where `DTAX` is assigned. For the equation `Armington`, go to the `Equations` tab and then click on the 'name' `ARMINGTON`. This opens a frame that can direct you to all references to the equation `ARMINGTON` in the programme.
5. Identify what each of the following (tax rate) symbols are, how they are assigned and explain their meaning in the model (use a reference file to guide you):
    - `TX`
    - `TS`
    - `TSS`
    - `TV`
    - `TM`
    - `TMS`
    - `TE`
    - `TF`
    - `TYH`

6. Find the values of the following taxes in the `lst` file and the code that generates them in the `gms` file:
    - production taxes for each activity
    - two sales taxes for each commodity
    - value added taxes for each commodity
    - *ad valorem* import tariffs for each commodity
    - specific import tariffs for each commodity
    - export taxes for each commodity
    - factor use taxes for each factor and activity
    - income tax
7. Review the default macro closures given in the core model. HINT: they are in an `INCLUDE` file that can be opened using `Ctrl+click`. It should have the following features:
    - Fix *CAPWOR*: lets exchange rate clear the foreign exchange market
    - Fix *INVESTSH*, the investment consumption share of final demand is fixed
    - Flex *SADJ*, the savings rates must be flexible for investment driven savings. Why must S*ADJ* be flexible in this case? Could **one** of *SHADJ* or *SEADJ* been flexible rather than *SADJ* and what difference might this make?
    - Fix *VGDSH*, the government consumption share of final demand is fixed: ALL tax rates and transfer scalars are fixed so government borrowing (*KAPGOV*) clears the government account. Why must *KAPGOV* be flexible in this case?
    - Fix *QEDADJ*. Why in this case can *VEDSH* not be flexible?
8. Review the factor market clearing conditions. Verify that there is full employment of all factors and that all factors are fully mobile across all production activities and that technology is fixed.
9. What is the model numéraire?

<u>Other Activities</u>

You should devote time, *inter alia*, to ensuring that you understand

1. the behavioural relationships in the model – you will not understand them all from the beginning, but time spent exploring the model, and the documentation, will be useful in the future (and can save time).
2. the content and location of the various `INCLUDE` files used by the model (note that IF the list of files used in the reference file is collapsed the `INCLUDE` files are in the order the files are used, i.e., those called directly from `stg_t.gms`, but files called indirectly, i.e., from within `INCLUDE` files follow the order they are called from within the `INCLUDE` file);
3. how the links for the declaration, assignment and use of parameters and variables in the reference file ensure that the relevant file will open in Studio; and
4. the relationships used to endogenise the functional distribution of income (*FSI*, etc.).

`$SETGLOBAL`

Explore the use of `$SETGLOBAL` in this model

1. how is `$SETGLOBAL` used to assign the database used in the model; and
2. how are the `$SETGLOBAL` and `$DROPGLOBAL` commands used to facilitate checking the SAM in different places in the code.

Cleanup

Use `clean_stg_t.bat` (note you will need to change the suffix on `clean_stg.dat` to `bat` – this is because files with the suffix `bat` can trigger errors when downloading files). To use `clean_stg.bat` double click on the file name in Windows Explorer.

This deletes selected files; it is safer than doing it manually.

Ex S2:2 Trade Policy: Ad Valorem Tariff Changes

> *This is a first experiment using stg_t. It repeats much that was done with experiments in the Practical CGE course: this is deliberate since the aim is to get started with this model rather than introduce new ideas/concepts.*

This exercise provides a summary of the information needed to begin using `STAGE_t`. For more detail, see the STAGE_t Technical document (*STAGE_t Model.pdf*) and the STAGE_t User Guide (*STAGE_t User Guide.pdf*).

The project directory created for Exercise S1:1 (`C:\cgemod\stg_t\stg_t_1`) contains the files needed for this exercise. All the exercises for Mod O7 are run in the directory `C:\cgemod\stg_t\stg_t_1.` This is one of the advantages of using the directory structure used by the STAGE and ANARRES models

Ad Valorem Tariff Changes

Taxes in the basic version of `stg_t` are *ad valorem*, based on values of transactions (in an exercise for module O8 you will re code an *ad valorem* tax as a specific tax - a tax per unit quantity imported). In this exercise, we will consider increases in *ad valorem* tariffs. Policy simulations will be run in a loop. Results will be accessed using GAMS Studio.

1. If the project form Ex S1:1 is not open use `File>Open in New Project` to open `C:\cgemod\stg_t\stg_t_1\stg_t.gms` in Studio.
2. Run the model `stg_t.gms` and check that the model has solved correctly – use all the checks detailed in Ex S1:1.
3. In Studio open the file `4_expts\stg_t_expt_base.inc`; this is the experiment file that you will start from and build on for Exercises S1 and Exercises S2. Rename the file as `stg_t_expt_1_2.inc` and save it in the directory `4_expts.`
4. Some of the information needed for the experiments will come from an Excel workbook; each set of experiments will have its own workbook. Open the file `1_2_Expt_data\stg_t_expt.xlsx` and save it as `1_2_Expt_data\stg_t_expt_1_2.xlsx`.

5. Before running the experiments, we need to assign some `$SETGLOBAL` instructions. But before doing so it is a good idea to review the experiment file and experiment data file.
6. Review the contents of the file `stg_t_expt_1_2.inc`. What experiments are implemented by this file?
7. Find where the experiment data file (`stg_t_expt_1_2.xlsx`) is read into the experiment file `stg_t_expt_1_2.inc`.
8. We will loop over a (controlling) set called `sim` (simulations) and `simc` (simulations currently run).
9. Answer the following questions (you may need to use the GAMS help system)
    a. How many elements are there in `sim` and `simc`?
    b. Where are the sets `sim`, `simc` and `clos` loaded into GAMS?
    c. How does GAMS know where to find the sets `sim`, `simc` and `clos`?
    d. How is the set `econs` assigned?
    e. How is the parameter `mod_cont` assigned?
10. In section `19a Policy experiments` of the model file `stg_t.gms` add two instructions. The first `res_dim` is used to declare and assign sets used in the results analysis files; this is an example of using `$SETGLOBAL` to customise declaration and assignment statements for different configurations of the results analysis files. The second `expt_data` is used to identify the Excel file that contains the experiment data.

```
$SETGLOBAL    res_dimen     sim
$SETGLOBAL    expt_data     stg_t_expt_1_2
```

11. In section `19a Policy experiments` of the model file `stg_t.gms` add an instruction to call in the experiment file (see Figure 7.2.1)

```
$INCLUDE 4_expts/stg_t_expt_1_2.inc
```

12. Now you need to modify the experiment file `stg_t_expt_1_2.inc` to make it use the `$SETGLOBAL` assignment for the experiment data. Modify the `$CALL "GDXXRW ....` Line by replacing `stg_t_expt_1_2` with `%expt_data%` so it reads

```
$CALL "GDXXRW i=1_2_Expt_data\%expt_data%.xlsx o=1_2_Expt_data\expt_in.gdx index=Layout!A4 trace=3"
```

13. Run the model and verify that there are 3 solutions found.
14. The results from the experiment will be in the file `stg_t.gdx`.
15. Answer the following questions:
    a. What values should the import tariff rates take?
    b. How can you verify that those are the rates applied?
    c. In what file will you find these values?
    d. Where in the experiment file are the results parameter declared?
    e. Where in the experiment file are the results parameters assigned?
    f. What does the instruction `stg_t.HOLDFIXED` mean?
    g. What does the instruction `stg_t.TOLINFREP` mean?
    h. What does the instruction `stg_t.TOLINFEAS` mean?
16. Our example version of the experiment file is called `stg_t_expt_1_3_sol.inc`.

Figure S1:2.1 Policy Experiments Section

```
*################ 19. POLICY EXPERIMENTS ################################

*$ontext

$SETGLOBAL   res_dimen    sim
$SETGLOBAL   expt_data    stg_t_expt_1_2

$INCLUDE 4_expts/stg_t_expt_1_2_sol.inc

*$offtext
```

Interpreting Results

1. Open the file `stg_t`.
2. Review `resTM, resPWM, resPM, resQM, resPE, resQE, resQXC, resHEXP, resEV, resPVA, resWF, resYH, resQINVD, resSCAL, resMACRO`
3. What impact does an increase in tariffs have on the economy:
    a. What happens to imports and why?
    b. What happens to the exchange rate and why?
    c. What happens to exports and why?
    d. What happens to production and why?

e. What happens to government consumption and why?

f. What happens to household consumption and why?

g. What happens to investment and why?

h. What happens to welfare and why? Which variable is the nearest equivalent to a welfare measure?

Ex S1:3 Accessing Model Results

This exercise uses the same experiment as developed for Ex 7:2. For this exercise the amount of information available as results has increased greatly.

The project directory created for Exercise S1:2 (`C:\cgemod\stg_t\stg_t_1`) contains all the files needed for this exercise.

Analysis of Results

The previous exercise produced a series of tables of results where each parameter `res***(*,*,sim)` contains the results for one or more variables. These results were accessed from the file `stg_t.gdx`. But this only reported the results for the levels values of the model variables: the analyses of the results can be enhanced by some processing of the levels values of the variables.

1. Open the file `5_analysis\stg_t_anal.inc` and examine its contents. Then answer the following questions:
   a. What is the role of the parameter `exp_cont`?
   b. How and where is the parameter `exp_cont` assigned?
   c. Why are the parameter declarations made outside of the `IF` statements?
   d. What does the file `stg_t_reslevel.inc` do?
   e. What does the file `stg_t_resmacro.inc` do?
   f. What does `%res_dimen%` do in the file `stg_t_resmacro.inc` do?
   g. What does the file `stg_t_reswelfare.inc` do?
   h. What does `%res_dimen%` do in the file `stg_t_reswelfare.inc` do?
   i. What does the file `stg_t_respercent.inc` do?
   j. What does `%res_dimen%` do in the file `stg_t_respercent.inc` do?
   k. Why must the file `stg_t_respercent.inc` be the last file in the file `stg_t_anal.inc`?
2. In section `19a Policy experiments` of the model file `stg_t.gms` add two more instructions. The first, `expt_res`, is used to identify the subdirectory where the experiment results are saved.

```
$SETGLOBAL    expt_res      trade
```

3. The second is to call in the file - `stg_t_anal.inc` - that controls the analysis of the results (we will look at this in more detail later)

```
$INCLUDE 5_analysis/stg_t_anal.inc
```

4. The modified section `19a Policy experiments` should be equivalent to that in Figure 7.3.1.
5. In the directory `6_results` create a subdirectory `trade`
6. The results will be stored in a subdirectory, `6_results\trade`.
7. Run the model and verify that there are 3 solutions found.
8. Answer the following questions using `stg_t.gdx` (this is a simple check to make sure that you ran the experiment that you intended to run):
    a. What values should the import tariff rates take?
    b. Verify that the intended import tariff were the rates applied?

Figure S1:3.1 Policy Experiments Section

```
*################# 19. POLICY EXPERIMENTS ##################################

*$ontext

$SETGLOBAL   res_dimen    sim
$SETGLOBAL   expt_data    stg_t_expt_1_2
$SETGLOBAL   expt_res     trade

$INCLUDE 4_expts/stg_t_expt_1_2_sol.inc

$INCLUDE 5_analysis/stg_t_anal.inc

*$offtext
```

Interpreting Results

1. The files referenced in this section are all in the sub directory `6_results\trade`.
2. Open the file `stg_t_reslevel.gdx`. and review the results *resTM, resPWM, resPM, resQM, resPE, resQE, resQXC, resHEXP, resEV, resPVA, resWF, resYH, resQINVD, resSCAL, resADJ*
3. Open the file `stg_t_respercent.gdx`. and review the results `resTM, resPWM, resPM, resQM, resPE, resQE, resQXC, resHEXP, resEV, resPVA, resWF, resYH, resQINVD, resSCAL, resADJ`

4. How does this additional information inform your understanding of the impact of an increase in tariffs on the economy:
    a. What happens to imports and why?
    b. What happens to the exchange rate and why?
    c. What happens to exports and why?
    d. What happens to production and why?
    e. What happens to government consumption and why?
    f. What happens to household consumption and why?
    g. What happens to investment and why?
5. Open the file `stg_t_resmacro.gdx`. and review the content. How does this additional information inform your understanding of the results.
6. What are the impacts of an increase in tariffs on
    a. Real and nominal GDP?
    b. Real and nominal demand by households?
    c. Real and nominal demand by government?
7. Open the file `stg_t_reswelfare.gdx`. and review the contents. How does this additional information inform your understanding of the results.
8. What are the impacts of an increase in tariffs on
    a. Household welfare?
    b. What are the empirical differences between (Hicks) EV and Slutsky EV?
    c. What are the theoretical differences between (Hicks) EV and Slutsky EV?

Saving Results

The addition of subdirectory `6_results\trade` means that once this series of policy experiments is complete the results are saved for future reference. Moreover, provided the experiment data file - `stg_t_expt_1_2.xlsx` - and the experiment file - `stg_t_expt_1_2.inc` - are saved, the results can be easily replicated.

Ex S1:4 Using Save and Restart

The Save and Restart facility in GAMS can be extremely useful, especially when developing code extensions and/or experiment files. It is not uncommon to encounter 'issues' when developing a series of experiments and this can be especially time consuming when testing to ensure experiments are doing what was intended. One approach is to use the set `sim` to identify the full set of possible experiments but to only run subsets (`simc`) while testing that the experiments have been coded correctly. In such situations, the base model is run every time a subset of experiments is run, which with large models (that may also be burdensome to solve) can take a lot of time.

The Save and Restart facility avoids the need to repeatedly solve the base model. While also separating the base model from the experiment model. This introduction to the Save and Restart facility uses the same model and experiments used in Exercises S1:2 and S1:3, i.e., the experiments will be same as in the previous exercises and hence there will be no need to do any analysis of results. The aim here is to develop a GAMS skill that can be used repeatedly in subsequent exercises and will be a useful option for you during the project phase (Module S4).

Save and Restart

"GAMS saves the information provided in input files in intermediate … files, called work files or scratch files. ….. They are usually deleted just before a GAMS run is complete. However, these intermediate files may be used to process an input file, save the result, and later reload this file and continue with processing another input file. ….. This is a useful feature that can help to reduce the required time when, e.g., several runs of similar models are being made, all of them sharing an equal large initialization part." (GAMS documentation, accessed Jan 2021).

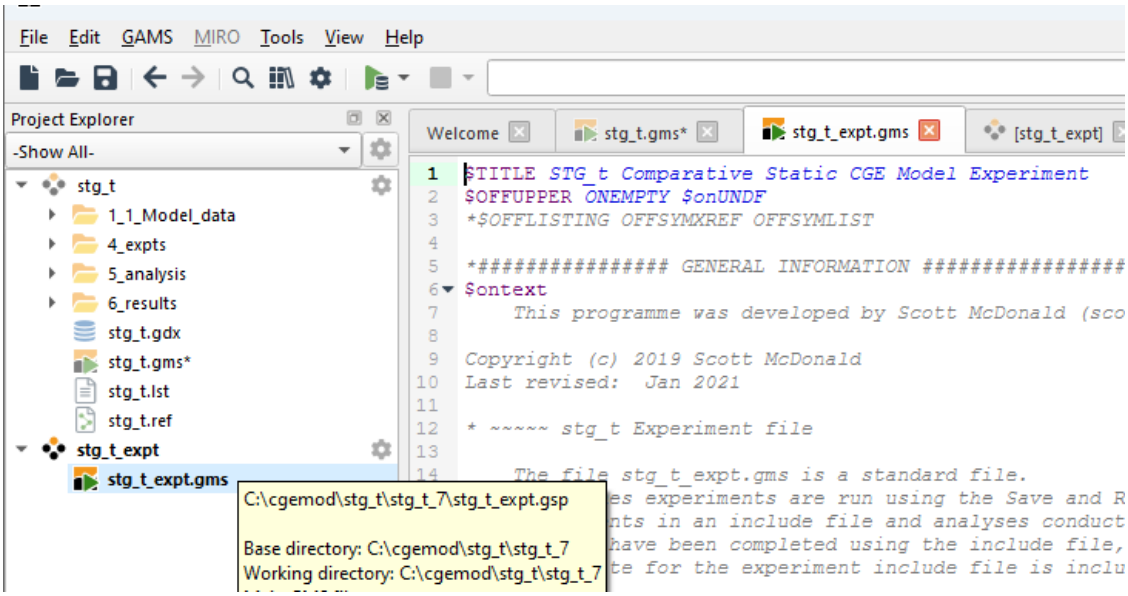
In the context of this CGE model the facility can be used to separate the model, i.e., the "initialization part" from the experiment, or simulation, part. This can be helpful when implementing large numbers of simulations and engaging in sensitivity analyses: sensitivity to macroeconomic closure choices, factor market clearing choices, and parameter, e.g., elasticity, settings. The Save and Restart facility will be central to the exercises for Module

O8, which concentrates on sensitivity analyses, and Module O9, which will concentrate on the interpretation of model results. It is anticipated that you will make use of the Save and Restart facility for your project, Module O10.

Setup

The project directory created for Exercise S1:2 (`C:\cgemod\stg_t\stg_t_1`) contains the files needed for this exercise.

1. In the model `'stg_t.gms'` switch off the experiment code added to section 19 (Policy Experiments) by bracketing the code with `$ontext` and `$offtext`.
2. Using `File>Open in New Project` [4]open the file `stg_t_expt.gms` will now be open in the project directory `stg_t_7`. Verify this by clicking on the settings wheel for the project `stg_t_expt`.
3. This means that you will be running two projects, each with its own `Main File` from a single directory. This turns out to be very useful.



4 NB: in Settings (`F7`) on the tab General in the Behavior section you can choose the option of the default use of `Ctl+O`; we choose that `Ctl+O` opens a file in the current project and therefore that `Ctl+Shift+O` opens in a new project).

Using Save and Restart

1. Select the `stg_t` project (double click on the file `stg_t.gms`) and this will result in `stg_t.gms` being the `Main File` – shown by the GREEN arrowhead being dark green. Because the project title is in BOLD this means `stg_t` is the active project and that when F10 (or F9) is chosen the programme that will run is the `Main File` in the active project.
2. Type `'s=save'` in the command line box and run the model `'stg_t.gms'` (without any experiments); the instruction `s=save` means that GAMS will save the work file from the solution generated by the model, `stg_t.gms`, that was solved.
3. Run the model and verify that the model has solved correctly, i.e., carry out the standard checks. The file `'save.g00'` will have been created in your working directory; this is a GAMS work file.
4. Now select the project `stg_t_expt` by double clicking on the file `'stg_t_expt.gms'`; this will now be the active project.
5. Review the content of the file `'stg_t_expt.gms'`; note that it is similar to the 'Policy Experiments' section you added to the model `'stg_t.gms'` in the previous exercise. The differences are only in terms of layout.
6. Open the experiment file `stg_t_expt_base_2.inc` and examine the contents of the file.
7. Now open the file `stg_t_expt_1_4.xlsx` in Excel and examine the contents, noting the extra worksheet.
8. Note that results are being directed to the subdirectory `trade_2`; hence you need to create the directory `6_results\trade_2` to avoid an error.
9. Type `'r=save'` in the command line box, and then run the programme `'stg_t_expt.gms'`. The instruction `r=save` means that GAMS will restart from the work file (`save.g00`) generated by the model that was solved when running `stg_t.gms`.
10. An incidental, but potentially useful, by-product of the save and restart system for running experiments is that a separate GDX file (`stg_t_expt.gdx`) can be generated.

11. Examine the results and compare them with those produced for Ex S1.3. HINT: you only need to visually check the results for several variables. Are the results different?
12. How many different closure settings were run?

The save and restart facility is especially useful when building up a sequence of experiments/simulations. Each simulation can be tested by controlling the member of the set `sim` that is implemented, to ensure it is doing as intended with the need to run only one solve command rather than two. With these small models this is not an issue, but with larger models, say 50,000 to 1,000,000 variables, the time savings can be substantial.

Ex S1:5 Trade Policy and Tax Replacement

> *Macroeconomic closure choices are major determinants of the precise patterns of the results produced by CGE studies, i.e., the results may be sensitive to the choice of closure setting. The method explored in these courses presumes that well-formulated studies carefully explore the implications of different closure choices rather than asserting a unidimensional vision of the operation of economies.*

There are many alternative macro closures that a modeler can impose. The STAGE, and ANARRES, family of models are coded from an agnostic perspective for two reasons: first, the world macroeconomic view of a model's developers should not be imposed on the user, and second model users should undertake sensitivity analyses with respect to macroeconomic closure choices. This exercise starts from a template experiment file, `stg_t_expt_base_2.inc`, that is set up as a two-loop system that runs the simulations loop inside a closure loop that allows for multiple different closures: two different tax replacement macroeconomic closures will be implemented.

The template experiment file is setup, so the closure selected is a repeat of the closure from the base model, i.e., `stg_t_cl_base.inc`. Note how the main characteristics of the closure are reported in a block of text before the instruction to include the closure file; this text is a copy of the text at the top of the closure file `stg_t_cl_base.inc`. Documentation is important; it is tempting to save a few seconds by not documenting what you are doing in a model's codes, but it is a false economy - "short cuts make for long delays" (Tolkien, The Lord of the Rings). Our practice is to run the simulations using this 'base' neo/new classical world view to provide a comparison with macroeconomic closures and factor market clearing conditions that, arguably, may better reflect the realities of the economy being analysed. In this exercise, no changes are made to the default factor market clearing assumptions.

We will investigate the government's closure choices, considering adjustments by different tax instruments when tariffs increase. Under the default settings, a tariff decrease produces an increase in government borrowing, i.e., an increase in debt going forward. Thus, there is a case for employing another tax instrument to offset the loss of tariff revenue and

avoid the government increasing its borrowings, i.e., a tax replacement instrument, and thereby avoiding manna from heaven from passing increased borrowings to future generations. With changes in tax instruments, different outcomes will be observed, and the outcomes will be influenced by the choice of tax replacement instrument; particularly with respect to the 'efficiency' of different tax instruments where a more 'efficient' is defined as one that produces less distortions.

The method used for different closure choices developed here run in a loop – for each policy simulation (also run in a loop). This way the results allow for straightforward analyses of the impacts on the economy for a variety of different assumptions about macroeconomic behavior.

Closure loops are more complicated than simulation loops. Recall, in the simulation loop, a parameter value was assigned a new level. In a closure loop, variables that were flexible will be held constant\fixed while other variables, which had been constant, will be made flexible. It is important to be sure that, because of the closure changes, the total number of equations equals the total number of variables in the model, otherwise the model will not solve. Further recall that when implementing simulations GAMS starts each search for a new solution from the previous solution; hence, in most cases, the values of the model's variables progressively diverge from the initial values.

The closure loops in this method always start by re-initializing all variables to their base values and all variables that can be fixed in a closure choice are made flexible. For a small model the gains from this decision may often be minimal, but for complex models, and models run in recursive dynamic mode, the gains can be substantial. Then the loop calls an "IF" statement which assigns an include file with the desired variables fixed and unfixed for a chosen closure setting.

The project directory created for Exercise S1:2 (`C:\cgemod\stg_t\stg_t_1`) contains all the files needed for this exercise.

Tax Replacement Closures in LOOP

In this exercise we will use the tariff reduction simulation from exercise S1:3, i.e., focus on the different closure options. This exercise will require some coding in the experiment file,

but again this coding will contribute to the development of a generic experiment file that will contribute to your project and your library of code resources. Policy simulations will be run in two loops, and results will be accessed from GDX files using GAMS Studio.

1. We will work in the project `stg_t_expt`, i.e., we will be separating the model from the experiment by using Save and Restart.
2. In the project `stg_t` rerum the model `stg_t.gms` using the save command in the command line.
3. Open the file `4_expts\stg_t_expt_base_2.inc`; this is our version of the experiment file at the end of Ex 7.4 plus some blocks of code that provide a template within which you can code the closure LOOP.[5] start from and build on for Exercises O7.
4. Rename the file as `stg_t_expt_1_5.inc` making sure it is saved in the subdirectory `4_expts`.
5. Review the contents of the file `stg_t_expt_1_5.inc`. What experiments and closures are implemented by this file?
6. Examine the contents of the three files included immediately after the LOOP keyword for the closures – `LOOP(clos),`. Answer the following questions
    a. What does the file `stg_t_varinit2.inc` do?
    b. What does the file `stg_t_reset.inc` do?
    c. What does the file `stg_t_cl_base.inc` do?
7. Open the file `1_2_Expt_data/stg_t_expt_1_4.xlsx` and save as `1_2_Expt_data/stg_t_expt_1_5.xlsx`. Review the contents of the file. Answer the following questions
    a. How is the set `clos` assigned?
    b. How is the parameter `clos_cont` assigned?
8. Make sure you have run the of the model file `stg_t.gms` using the Save facility.

[5] By starting from our version, it is easier to ensure that the instructions are clear because we then know where items are in the code, while still building up the code for Exercises O7 and Exercises O8.

9. We now need to update the `$SETGLOBAL` instructions in the master experiment file (`stg_t_expt.gms`). We need to call in different data and a different experiment file. The statements should be the same as those below.

```
$SETGLOBAL   res_dimen    simc,clos
$SETGLOBAL   expt_data    stg_t_expt_1_5
$INCLUDE 4_expts/stg_t_expt_1_6
$INCLUDE 5_analysis/stg_t_anal.inc
```

10. Run the experiment file, `stg_t_expt.gms`, using Restart and verify that the experiments ran as intended.
11. Answer the following questions:
    a. How many simulations should be implemented?
    b. How many closure LOOPS should be implemented?
    c. How can you verify that those variables that should be fixed by the closure settings were in fact fixed in the simulations?

Adding a Second Closure to the Model

The second closure that will be added to the closure loop is a tax replacement closure that replaces the revenues from import duties with a multiplicative increase in the income (direct) tax rates paid by households. The policy objective will be to find a set of income tax rates that ensures that the real value of the government's borrowing/saving does not change.

1. Open the file `3_close/stg_t_cl_base.inc` and save the file as `3_close/stg_t_cl_tyh.inc`.
2. Review the file `stg_t_cl_tyh.inc` to refresh your understanding of the base closure settings.
3. In the file `stg_t_cl_tyh.inc` fix `KAPGOV` at its base value and unfix/flex `TYHADJ`.
4. In the file `stg_t_expt_1_5.inc` go to the section "4b. Selecting the closure" and after the end of the first IF statement (`IF(clos_cont(clos) = 1,`) replicate the content of the first IF statement using '`IF(clos_cont(clos) = 2`' and modifying the `$INCLUDE` command to call the income tax replacement closure - `stg_t_cl_tyh.inc`.

5. Make sure you document the content of the file `stg_t_cl_tyh.inc` in the file, and repeat the documentation on the file `stg_t_expt_1_5.inc`.
6. Go to the file `stg_t_expt_1_5.xlsx`. It is necessary to update the assignment of `clos` and `clos_cont`. A label is needed for the element in `clos`, e.g., `cl_tyh`, and a value is needed for the value in `clos_cont(clos)`. Number the closures sequentially so that the new closure has the number **2**. **NB**: the IF condition used for this closure.
7. Run the experiment file, `stg_t_expt.gms`, using Restart.
8. Verify that the closure settings that were implemented were those you intended. Use the results to verify the closure setting for each member of the closure LOOP.
9. If the closure settings have not been implemented, find out why.

Adding a Third Closure to the Model

The third closure will use a different tax replacement instrument: General Sales Taxes (GST) which in the model are the variable *TS*. Again, the policy objective will be to find a set of sales tax rates that ensures that the real value of the government's borrowing/saving does not change.

1. Open the file `stg_t_cl_tyh.inc` and resave the file as `3_close/stg_t_cl_ts.inc`.
2. In the file `stg_t_cl_ts.inc` fix `KAPGOV` and `TYHADJ` at their base values and unfix/flex `TSADJ`.
3. In the file `stg_t_expt_1_5.inc` go to the section "4b. Selecting the closure" and after the end of the second IF statement (`IF(clos_cont(clos) = 2,`) replicate the content of the first IF statement using '`IF(clos_cont(clos) = 3`' and modifying the $INCLUDE command to call the income tax replacement closure – `3_close/stg_t_cl_ts.inc`.
4. Go to the file `stg_t_expt_1_5.xlsx`. It is necessary to update the assignment of `clos` and `clos_cont`. A label is needed for the element in `clos`, e.g., `cl_ts`, and a value is needed for the value in `clos_cont(clos)`. Number the closures sequentially so that the new closure has the number **3**. NB**:** the IF condition used for this closure.

5. Run the experiment file, `stg_t_expt.gms`, using Restart.
6. Verify that the closure settings that were implemented were those you intended. Use the results to verify the closure setting for each member of the closure LOOP.
7. If the closure settings have not been implemented find out why.
8. Our version of the experiment file is called `stg_t_expt_1_5_sol.inc`. It contains all three different closures. Our versions of the closure files are called `stg_t_cl_tyh_sol.inc` and `stg_t_cl_ts_sol.inc`. There is also a closure version for VAT - `stg_t_cl_tv.inc`.

Interpreting Results

1. Review the results and answer the following questions:
    a. How does an increase in tariffs affect imports under the three different tax regimes? Why?
    b. What happens to household expenditure under the three different tax regimes? Why?
    c. What happens to household income under the three different tax regimes? Why?
    d. What happens to the exchange rate under the three different tax regimes? Why?
    e. What happens to exports under the three different tax regimes? Why?
    f. What happens to government consumption and why?
    g. How does production change under the three different tax regimes? Why?
    h. What happens to investment under the three different tax regimes? Why?
    i. How does welfare change under the three different tax regimes? Why?
2. What conclusions can you draw about the relative efficiency of income taxes and sales taxes as tax replacement instruments? Why?
3. How do the different closure settings impact on the cost of living and the costs of production? Why?

Saving Results

You may choose to copy and save the directory of results, e.g., a directory `6_results\trade_5`, for future reference. It will be easy enough to recreate the results: if you do not, the next exercise will overwrite the results from previous experiments.

Ex S1:6 Project S1:1: Interpreting Tax Replacement Options

Tax replacement (macroeconomic) closures are standard in CGE studies.
But not all tax replacement closures produce the same results: why?

This exercise is the project for module S1. For this exercise you need to create an experiment file, e.g., `stg_t_expt_1_**.inc` (where ** is a unique identifier), that contains the following:

1. 2 loops, one for the shocks and one for the different closures;
2. 6 to 10 different trade tax shocks that explore a **common** theme;
3. 3 macroeconomic closure settings – the first will be the base closure, the second will use income taxes as the tax replacement instrument and the third will use another **indirect** tax replacement instrument, but not GST;
4. you can use multiplicative or additive changes in the tax replacement instruments); and
5. all factors will be fully employed and mobile.

Run the model and save the results.

Analyse the results and identify any differences between the results with different tax replacement instruments. Then **explain**, and illustrate, why the different tax replacement instruments produce different results. HINT: you will need to understand how prices are related in the model.

Project S1:1

Compile your results and comments in a Word document with a maximum of, say, 2,500 words, excluding tables and figures but an absolute maximum of 10 sides – using small print is cheating!!!. The reasons for the limits are to encourage you to identify the important results and concentrate the report on the important implications of the results.

An example of how we might have done this project is available on the course page for Mod S1.