HANDS-ON COMPUTING WITH ORANIG
PART B: SIMULATIONS

Implementing ORANIG

The GEMPACK program TABLO is used to process the TABLO Input file ORANIG.TAB. We call this *implementing* the model. TABLO prepares files which contain information about the variables and the equations of the model, in a form ready to be used when running a simulation using the program GEMSIM. These files are called GEMSIM Auxiliary files (they are suffixed GST and GSS).

Set the Default Directory

First select in the main WinGEM menu,

File | Change both default directories

and set the working directory as subdirectory **C:\GPWork\ORANIG**
The following box contains a set of instructions which are the wrong way to implement the ORANI-G model. This method does not work because the model is too big to be run without condensation[1]:

The WRONG Way to run TABLO for ORANI-G

To implement ORANI-G, open a TABLO Window by choosing

Simulation | TABLO Implement

Choose the TABLO Input file ORANIG.TAB, *Run* TABLO. **!*WRONG***

If you do this, when you try to run the model using the program GEMSIM,
you will get an error message saying that the model has exceeded the dimensions of your type of Executable-Image licence. [Or you may get a message that the closure is not valid.[2]]

Condensation

ORANI-G is quite a large model that will not run on the laboratory PC unless it is condensed. So the moral of this exercise is don't choose the TAB file, use a STI file instead to carry out condensation.

The Correct Way to Run TABLO for ORANI-G

To implement ORANI-G correctly, carrying out the condensation, open a TABLO Window by choosing

Simulation | TABLO Implement

In doing a condensation, you need to say which variables are to be condensed out and which equations are to be used. This information is prepared in a Stored Input file called ORANIG.STI. To tell TABLO to use this file, choose **in the menu for the TABLO Window**

Options | Run from STI file ***!CORRECT***

and then ***Select*** the name of the Stored Input file as ORANIG.STI

Now select ***Run*** and wait while TABLO processes ORANIG.TAB and performs the condensation.

[1] "Condensation" reduces the number of model variables by using equations to "substitute out" some variables. See Section 7 of the ORANI-G document.
[2] The reason why you get a non-valid closure is because during condensation some of the variables are omitted and this alters the closure.

When large models such as ORANI-G or GTAP are implemented, it is usually necessary to **condense** the model. This means carrying out algebraic substitutions to reduce the size of the system of linear equations to be solved at each step of a multi-step calculation. This condensation is carried out by TABLO after the stage where the syntax of the TABLO Input file is checked and before the GEMSIM Auxiliary files are written.

Here TABLO is taking all of its instructions from the Stored-input file ORANIG.STI. In particular, this contains a large number of instructions to substitute out and/or backsolve for different variables.

TABLO will produce GEMSIM Auxiliary files ORANIG.GSS and ORANIG.GST. These contain all the theory in ORANIG.TAB and the condensation actions in ORANIG.STI.

When TABLO has finished, look at the Log file. Go to the end (Ctrl-End) to see that TABLO has completed successfully. In the Information file, search for "Final status" to see variables and equations that are in the model after condensation.

Now return to WinGem and choose ***Go to GEMSIM***

Preparing the Command File

Before you run a simulation, you need to prepare a small text file called a Command file. Because it is a text file, you can use the text editor to prepare it. Select, in the main WinGEM menu,

File | Edit file....

and select the *incomplete* Command file called ORANIG.CMF to edit.

You are going to run a Homogeneity simulation on ORANI-G as a check. In this simulation all Nominal variables are to be increased by one percent while keeping all Real variables fixed.

Each simulation needs its own Command file. Follow the steps below to modify the file in the editor.

(1) Choose ***File | Save As...*** and change the name of this file[3] to ORANIGHO.CMF.
This simulation will produce a Solution file with the name ORANIGHO.SL4 (an abbreviation for "ORANIG homogeneity test").

(2) Type in the name of the GEMSIM Auxiliary files for the model we are using for this simulation.

```
auxiliary files = ??? ;
```

(3) We would like to include a description of the simulation to remind us of the purpose of this CMF file. Edit the line

```
verbal description = ???? ;
```

to include the description – ORANI-G homogeneity test. You can add more than one line if you like. The verbal description ends with a semi-colon ;

(4) Add the closure and shocks as described in the next two sections.

(5) Save your Command file

Closure

[3] It is now possible to use longer filenames in GEMPACK. It is probably better not to use file or folder names containing spaces since then you have to remember to add quotes " " whenever you refer to the name. However, the Fortran compilers cannot handle non-English (eg Asian or Scandinavian) characters in your file or directory names. Strange characters (such as '&') might also cause problems.

When you are creating your own model, it can be difficult to find a valid closure. The first test your closure needs to satisfy is that the number of endogenous variables (actually the number of separate components of variables) must be equal to the number of equations (the number of separate equations). The second test is whether you can solve the equations, that is, that the set of equations is not structurally or numerically singular.

In the ORANI-G model the equations are all called "E_xxx" where xxx is the name of some variable in the model. We can usefully imagine that each equation explains a particular variable. Variables not explained by any equation are deemed to be exogenous in the standard closure. Although this is not really true since the solution of a set of equations depends on all the equations in the set, it gives us a starting point in finding a closure.

TABmate has a tool that helps us find a closure when this system[4] of naming equations is used. In TABmate, leave the Command file ORANIGHO.CMF open. Open the file ORANIG.TAB and click on the button marked **TABLO STI**. Select the STI file ORANIG.STI and run TABLO within TABmate. When the run has finished, we are ready to find a closure.

In the TABmate menu, select ***Tools | Closure*** This will produce a report on the Closure. Briefly read this report, then search for the section which starts

```
Section 4. Unmatched variables
******************************

Following variables are unmatched - probably exogenous
(text below is formatted so you can paste straight into the CMF file):

! Automatic closure generated by TABmate
!                Variable     Size
Exogenous           a1cap ; ! IND   Capital-augmenting technical change
Exogenous         a1lab_o ; ! IND   Labor-augmenting technical change
Exogenous           a1lnd ; ! IND   Land-augmenting technical change
Exogenous          a1prim ; ! IND   All factor augmenting technical change
Exogenous           a1tot ; ! IND   All input augmenting technical change
Exogenous           a2tot ; ! IND   Neutral technical change - investment
Exogenous            a3_s ; ! COM   Taste change, household imp/dom composite
Exogenous        capslack ; ! 1   Slack variable to allow fixing aggregate capital
Exogenous      delPTXRATE ; ! IND   Change in rate of production tax
```

Select all the lines in this section starting with the word "Exogenous", copy them and then paste them into the file ORANIGHO.CMF. This is a list of exogenous variables. Add the following line to tell the program that the remainder of the variables are endogenous:

```
rest endogenous ;
```

This closure is sometimes called the "structural" closure. See Section 5 of the ORANI-G document for further discussion on closures.

Shocks

The final edit we need to make is to add appropriate shock commands. We need to complete the line beginning with the word "shock" in the CMF file. The way to write shock statements is

shock <*variable name*> = <*list of shock values*> ;

Look through the list of exogenous variables in the Command file to find the names of the variables you wish to shock in order to increase all Nominal variables by one percent while keeping all Real variables

[4] In order for TABmate to know which variable is explained by a given equation, the modeller must follow a naming convention for equations. The convention is that the equation which explains variable "p1", say, is named "E_p1" (prefix E_). If you do not follow this convention, the Closure command will be no use to you, although the rest of TABmate will work normally. Each equation must be of the same dimensions (range over the same sets) as the variable which it explains. TABmate checks that the dimensions match.

fixed. (You can only shock variables which are exogenous.) For each shocked variable, complete a shock statement in ORANIGHO.CMF, replacing the question marks below with this variable name.

```
shock  ????  =  1 ;
```

Now save this file and exit from the editor.

Running a Homogeneity Simulation

In WinGEM select ***Simulation | GEMSIM Solve*** In the GEMSIM window, click on the ***Select*** button and select the Command file that you have just made, called ORANIGHO.CMF.

Now run this simulation by clicking on the ***Run*** button.

When the simulation has completed, choose ***Go to ViewSOL***.
View the results. Check the results for some real variables and some nominal ones. Is this model simulation homogeneous?

The homogeneity simulation illustrates the economic proposition that *only relative prices matter*. The general idea is that:

if all nominal variables on the exogenous list are shocked by X%,
and if the model is homogeneous,
then all nominal %change variables should change by X%
but real variables will not change.

Check the exogenous list to see if ALL nominal exogenous variables have been shocked.
Nominal variables are expressed in local currency.]
[Hints: Look at section 4.4 of the main ORANIG document to see the naming system used.
Variables whose names begin with "**a**" are tech change variables. Are these real or nominal?
Variables whose names begin with "**f**" are shifters.

Continue working along these lines until your results show homogeneity.

Using Swap Statements

Once we have one valid closure it is possible to "**swap**" variables to obtain a different valid closure.

In the previous simulation, the variable *f5tot2* was exogenous. In the next simulation we want *f5tot2* to be endogenous and instead we want *x5tot* to be exogenous. (Both of these variables have one component so the number of exogenous variables will stay the same.) One way to swap these variables is to delete the line

```
Exogenous  f5tot2 ;
```

and replace it with the line

```
Exogenous  x5tot ;
```

A better way is to leave the closure in the file as it is and add "swap" statements after the closure:

```
exogenous ......   ;
exogenous.......   ;
.....
rest endogenous ;
swap f5tot2 = x5tot ;
```

The advantage of this is that you can easily see how this closure is different from the original closure. Add the swap statement to the file ORANIGHO.CMF and save it.In the GEMSIM window, run a simulation with this file. Look at the results in ViewSOL. In general if you change the closure, you would expect to get different results, even if you have the same shocks. Why do you get the same results here?

Notes on Closure - Using Swaps to get the Short Run Closure

You do not need to carry out the swaps in this section as part of the exercises in the computer labs.

This section is for your information at some time when you are thinking about closures, perhaps when you are developing your own closure for some simulation, or for your own model. The short-run closure is described in section 5 of the ORANI-G document and is listed in Table 3. In the command file ORANIGSR.CMF above, it is given in the simplest possible form, just a list of exogenous variables followed by "rest endogenous". It doesn't give you many ideas as to how this closure was made or how ir relates to the structural closure used in the Homogeneity Simulation.

To derive the short run closure from the structural closure, we could use the following swap statements: (In each case we follow the convention that the variable on the left-hand side is exogenous in the old closure, and the variable on the right-hand side is exogenous in the new closure.)

```
!       Old exogenous      New Exogenous
       swap w3lux    =     x3tot ;
       swap invslack =     x2tot_i ;
       swap f5tot2   =     x5tot ;
       swap fx6      =     delx6 ;
       swap f1lab_io =     realwage ;

xSet EXOGINV  # 'exogenous' investment industries #
 (ElecGasWater, Construction, FinanceInsur, OwnerDwellng,
  PropBusSrvc, GovAdminDfnc, Education, HealthCommun);
xSubset EXOGINV is subset of IND;
xSet ENDOGINV # 'endogenous' investment industries # = IND - EXOGINV;

! investment linked to profits
       swap x2tot(ENDOGINV) = finv1(ENDOGINV);
! investment follows aggregate investment
       swap x2tot(EXOGINV) = finv2(EXOGINV);
```

Here the ***xSet*** and ***xSubset*** statements are used to divide the industry set IND into two subsets ENDOGINV and EXOGINV. In set ENDOGINV, finv1 is exogenous and the ENDOGINV set of industries follow Rule 1 in the TAB file. In set EXOGINV, finv2 is exogenous and the EXOGINV set of industries follows Rule 2 in the TAB file. See Excerpt 31 in the TAB file and section 4.26 in the ORANI-G document.

Having trouble finding a closure

You do not need to carry out the swaps in this section as part of the exercises in the computer labs.

If you are developing your own model at home or changing an existing model or changing the closure in a model, you may have difficulties finding a good closure. One way of proceeding is to return back to an existing closure that works. To proceed to your new closure, put in one swap statement at a time and run a test simulation until you reach the new closure that you want to use.

[However sometimes this does not work. For example if you add the swap

```
       swap invslack = x2tot_i ;
```

which makes the aggregate investment x2tot_i exogenous, you would need to free up at least one component of x2tot by a swap to make it endogenous at the same time. See Excerpt 31 for more details.]

Another tip, for when you add a new equation which will affect the rest of the system: include a new shift variable which can be left endogenous. This prevents the new equation from interacting with others, and allows the previous closure to be used. To 'switch on' your new equation, ‘swap’ the shift variable with another model variable of equal size (that your new equation will determine).