

Centre of Policy Studies Working Paper

No. G-257 November 2015

The Labour Module in a Dynamic Regional CGE Model

Glyn Wittwer

and

Janine Dixon

Centre of Policy Studies, Victoria University

ISSN 1 031 9034

ISBN 978-1-921654-65-7

The Centre of Policy Studies (CoPS), incorporating the IMPACT project, is a research centre at Victoria University devoted to quantitative analysis of issues relevant to economic policy. Address: Centre of Policy Studies, Victoria University, PO Box 14428, Melbourne, Victoria, 8001 home page: www.vu.edu.au/CoPS/ email: copsinfo@vu.edu.au Telephone +61 3 9919 1877

The Labour Module in a dynamic, regional CGE model

Prepared by Glyn Wittwer and Janine Dixon, Centre of Policy Studies, Victoria University

November 2015

The NSW Department of Industry commissioned CGE model development in order to enhance regional labour market analysis. The Centre of Policy Studies has developed VUEF (Victoria University Employment Forecasting), a dynamic model used in labour market forecasting. VUEF is a national dynamic model. It includes regional detail, but this is limited to a top-down module in which regions do not have separate labour markets and production functions. Rather, it relies on exogenous regional shares to distribute national results to each region.

Dynamic TERM (The Enormous Regional Model), with multi-regional bottom-up representation, has been used for policy analysis at the regional level at the Centre of Policy Studies since 2003. Until now, it has not been used to examine occupational or skills composition in the labour market.

We have adapted the recently developed VUEF labour market module for use in dynamic TERM. This paper elaborates on the regional labour market module, with minimal details of the linkages with the core TERM model. The labour market module has been developed as a standalone model for training purposes. In practice, it is likely that the fully dynamic TERM model with the labour market module will be used for analysis.

This paper also includes some guidance concerning the use of RunDynam.

JEL classification: R13, R15, R23, P25, P48

Keywords: skills shortages, occupational forecasting, regional modelling, RunDynam



Centre of Policy Studies
Campus: 300 Flinders St
PO Box 14428
Melbourne Victoria 8001
Australia

Contents

Introduction	5
Part 1: Demand and supply for labour units	9
Part 2: Translation of labour units into persons	10
Part 3: Market clearing equations	11
Part 4: Exogenous setting of skills supply	12
Part 6: Miscellaneous labour market aggregates	14
Part 7: Remaining equations from the core dynamic model	14
Replicating the baseline of 2014 from the full TERM model	15
Running the labour module in RunDynam	18
Appendix 1: Gathering shocks file from TERM skills run for LabModule	23
Appendix 2: Skills and occupations in labour market module	25

Tables

Table 1: ModuleBase2014.cmf	16
Table 2: Part of ModulePol2014.cmf	17
Table A1: Skills introduced to LabModule and VU-TERM	25
Table A2: Description of 3 digit ANZSCO occupations	26

Figures

Figure 1: RunDynam Sim Overview page	19
Figure 2: RunDynam Closure/Shock page	20
Figure A1: Results page	23

Introduction

CGE models may contain tens of thousands of equations and variables. The core theory of a CGE model is detailed in the short courses provided by the Centre of Policy Studies around the world. The main interest in the version of dynamic VU-TERM (Wittwer 2012) commissioned by NSW Department of Industry is in the labour market detail added to VU-TERM in response to the department's requirements. The purpose of this document is to detail the theory of the skills module. While there are some references to the core model, we concentrate on several dozen equations in this document. This document falls short of being fully self-contained. More comprehensive references to the theory of a CGE model and coding of it can be found in the ORANIG document (Horridge 2014).

Labour module: Overview

The labour market module provides a mechanism by which movements of labour between occupations can be restricted. In a typical CGE setting, the aggregate supply of labour is assumed either to be exogenous, driving movements in the average wage; or endogenous, with the national wage fixed. In comparative dynamic simulations, endogenous movements in the unemployment rate are possible, but a lagged wage adjustment mechanism ensures that changes in unemployment are only short term.

Within this framework, the occupational composition of employment can be determined in one of two ways – endogenously or exogenously. If occupation wages are assumed to move with the economy-wide average, the implication is that the supply of labour to any occupation is completely elastic. On the other hand, occupational relative wages can be determined endogenously if occupational labour supply is set exogenously (say as a fixed proportion of total employment). Neither setting is satisfactory for detailed modelling of labour markets: the first attributes too much fluidity to occupation movements, while the second is too rigid.

Dixon and Rimmer (2003) introduce constraints on the composition of labour supply in the MONASH model by explicitly modelling the transition of workers from one status to another (including unemployment) over each period of the simulation. The status of an employed worker is defined by his or her occupation at time $t-1$. The extent to which workers will transition into another occupation at time t is governed by movements in relative wages and parameters which determine the occupational preferences of workers.

Another remedy to this issue is to introduce skills to the model (Meagher and Pang, 2011). This is the approach used by Giesecke et al (2015) and the TERM labour market module. A requirement of Giesecke et al (2015) is that initial occupational wage rates are uniform across industries. Our approach relaxes this requirement, adding a level of complication to the model equations. Furthermore, the TERM labour market module includes a regional dimension.

Skills are identified as the highest post-school qualification, classified by level and field, held by an individual: for example, "Bachelor Degree, Management and Commerce". A matrix of headcounts of skill by occupation shows the occupational composition of employment for every skill group. In the labour market module, it is the supply of labour by skill, rather than by occupation, that is set exogenously.

Skill groups can allocate labour to occupations within the constraint of a transformation frontier. For each skill group, the initial position on the transformation frontier is given by the initial occupation shares. Movements are induced by changes in relative occupational wages. Movements away from the initial position are typically small, although in a dynamic setting large

changes may accumulate over many years. As such, the module provides an intermediate solution between the two extremes of fixed relative wages and fixed occupation shares.

The transformation frontier is described by the CET function. Wage bills, rather than headcounts, classified by skill and occupation are required to parameterise the function. The implementation of the transformation frontier is described in the remainder of this document. On/off shift variables, typical in GEMPACK models, enable the labour market module to be activated or de-activated as required. With the labour market module switched “off”, the resultant differences between labour supplied and demanded can be used to indicate occupations in shortage or surplus.

We use TABLO code to represent the model in this section. TABLO code is part of GEMPACK software. It resembles ordinary algebra. It can be coded and modified by practitioners without specialist computing skills.

A standalone version of the labour module has been developed. The TABLO file containing the model is in LabModule.tab

The code starts by declaring the file name JANINE from which the names of sets and the database will be read. Note that the SKILL and OCC sets are read from file, whereas the industry set is not. For the purposes of this module, the relatively aggregated industry set will remain constant. However, the SKILL and OCC sets may be aggregations of the full 56 SKILL and 97 OCC elements. These sets are derived from ABS labour market data.

```
File Janine;  
Set SKILL read elements from file Janine header "SKIL";  
Set OCC read elements from file Janine header "OCC";  
Set DST (SydneyNSW, CoastalNSW, InlandNSW, RoA);  
Set IND # Industries # (Livestock, Crops, ForestFish, Mining, FoodProds,  
TCFs, OthManuf, MetalPrds, Utilities, Construction, Trade, Accomodation,  
Restaurant, Transport, Communicatn, BankFinIns, OwnerDwelling, BusSrvces,  
GovAdmDef, Education, HealthComCre, OthService);
```

The next section declares some Coefficients, which are either database values or quantities, or model parameters. By convention, database coefficients, be they values or parameters, appear in upper case. Coefficients are either read from a database using a READ statement, or calculated using a FORMULA. Typically, we use a FORMULA statement when adding up across a dimension, as is seen in the calculation of QPERS_S, which is equal to the three dimensional PERS coefficient summed across the Skill dimension.¹

The labour market module requires data for the quantity of labour measured in both hours and value weights. WAGEBILL includes the initial year value of wage bills across occupations, skills and regions. PERSIND includes the number of people employed in the same dimensions. WAGEBILL, for example, will assign a higher value per person to managers than unskilled workers. A 10% decrease in unskilled workers, whose wages are lower than for other skills, therefore will have a larger proportional impact on PERSIND summed across occupations (PERSIND_O) than the WAGEBILL summed across occupations (WAGEBILL_O).

¹ The dynamic TERM model may be aggregated in the SKILL and OCC dimensions to reduce the time required to compute a model solution. Appropriate aggregations depend on the focus of a particular study. However, further model development and refining has enabled us to run the full TERM skills model with full SKILL and OCC, with acceptably short solution times.

Coefficient

```
(all,o,OCC)(all,s,SKILL)(all,d,DST) WAGEBILL(o,s,d) # Wage bill, occ x skill #;
(parameter) SIGMA_LABCET # CET substitution between occs #;
(all,o,OCC)(all,d,DST) WAGEBILL_S(o,d);
(all,s,SKILL)(all,d,DST) WAGEBILL_O(s,d);
(all,d,DST) WAGEBILL_OS(d);
(all,i,IND)(all,o,OCC)(all,d,DST) PERSIND(i,o,d);
(all,i,IND)(all,d,DST) PERSIND_O(i,d);
(all,o,OCC)(all,d,DST) PERSIND_I(o,d);
(all,d,DST) PERSIND_OI(d);
(all,o,OCC)(all,s,SKILL)(all,d,DST) PERS(o,s,d) # Head count, occ x skill #;
(all,o,OCC)(all,d,DST)QPERS_S(o,d);
(all,s,SKILL)(all,d,DST)PERS_O(s,d);
(all,d,DST)PERS_OS(d);
```

```
Read WAGEBILL from file JANINE header "VSKL";
SIGMA_LABCET from file JANINE header "LCET";
PERSIND from file JANINE header "NPRS";
PERS from file JANINE header "POSK";
```

```
Formula (all,o,OCC)(all,d,DST) WAGEBILL_S(o,d)=sum{s,SKILL,WAGEBILL(o,s,d)};
(all,s,SKILL)(all,d,DST) WAGEBILL_O(s,d)=sum{o,OCC,WAGEBILL(o,s,d)};
(all,o,OCC)(all,d,DST) PERSIND_I(o,d)=sum{i,IND,PERSIND(i,o,d)};
(all,i,IND)(all,d,DST) PERSIND_O(i,d)=sum{o,OCC,PERSIND(i,o,d)};
(all,d,DST) PERSIND_OI(d)=sum{o,OCC,PERSIND_I(o,d)};
(all,d,DST) WAGEBILL_OS(d)=sum{s,SKILL,WAGEBILL_O(s,d)};
(all,o,OCC)(all,d,DST)QPERS_S(o,d)=sum{s,SKILL,PERS(o,s,d)};
(all,s,SKILL)(all,d,DST)PERS_O(s,d)=sum{o,OCC,PERS(o,s,d)};
(all,d,DST)PERS_OS(d)=sum{o,OCC,QPERS_S(o,d)};
```

```
Coefficient (parameter) SIGMALABSK;
```

```
Formula (initial) SIGMALABSK = 2.0;
```

The next section declares some Variables. By convention, percentage change variables are lower case. To distinguish between various labour demands and labour supplies, we suspend this convention in the case of the variables `pers_sDEM`, `pers_iSUP`, `emp_persoSUP`, `xlab_iDEM`, `pers_isDEM` and `pers_iDEM`. To distinguish between labour units and persons we use `xlab` to identify labour units, and `pers` to identify persons.

Labour units and persons

The mathematical functions of the labour market module require quantity data to be identified in units with a consistent value. We define a “labour unit” to be a unit of labour worth \$1 in the initial solution. By following this convention, the CES production function and CET transformation frontier are relatively straightforward to define.

However, “labour units” are typically of little interest to policy-makers. Therefore, we transform many of the model’s results from labour units into hours or persons. In making this transformation, we find that an industry may employ more labour units but fewer people if it switches from relatively low skilled workers to high skilled workers (who provide more labour units per hour). The gradual transformation of the workforce from low-paid skills to high-paid skills is a source of technical change that can be seen in the growth of employment measured in labour units relative to employment measured in hours.

```

Variable (all,o,OCC)(all,d,DST)plabos_s(o,d) # Skill composite effective demand
price #;
(all,s,SKILL)(all,d,DST)plab_ave(s,d) # Average cost of unit of skill s #;
(all,o,OCC)(all,s,SKILL)(all,d,DST) plabos(o,s,d) # Nominal wage by occ and skill
#;
(all,o,OCC)(all,s,SKILL)(all,d,DST) xlabos(o,s,d) # Labour units (hours), occ and
skill #;
(all,s,SKILL)(all,d,DST) xlabo(s,d) # Supply of skill s, wage bill weights #;
(all,i,IND)(all,o,OCC)(all,d,DST)pers_sDEM(i,o,d);
(all,o,OCC)(all,s,SKILL)(all,d,DST)pers_iSUP(o,s,d) # Persons supplied, OCC x
Skill#;
(all,o,OCC)(all,d,DST) plab_i(o,d);
(all,o,OCC)(all,s,SKILL)(all,d,DST)f_plabos(o,s,d) # f_plabos exogenous = core on,
Lab module off #;
(all,s,SKILL) (all,d,DST) emp_pers_sk(s,d) # Total emp by skill, person weights #;
(all,o,OCC)(all,d,DST) emp_persoSUP(o,d) # Add-up of occupational supplies,
persons #;
(all,d,DST) emp_pers_o(d) # Add-up of regional supplies, persons #;
(all,o,OCC)(all,d,DST) pers_is(o,d) # Demand, persons by occ #;
(all,s,SKILL)(all,d,DST) r_skill_wpop(s,d) #Ratio of skill to working population#;
(all,d,DST)avE_skillsw(d) #Average of skill switches for given wpop#;
(all,o,OCC)(all,s,SKILL)(all,d,DST)xlabsk(o,s,d) # Skill shift #;
(all,o,OCC)(all,s,SKILL)(all,d,DST)xlab_iDEM(o,s,d) # Aggregate demand, OCC x
Skill #;
(all,o,OCC)(all,i,IND)(all,d,DST)ff1lab(i,o,d);
  (all,s,SKILL) (all,d,DST)f_pers_i(s,d);
(all,o,OCC)(all,s,SKILL)(all,d,DST)pers_iDEM(o,s,d);
(all,o,OCC)(all,d,DST)pers_isDEM(o,d);
(all,o,OCC)(all,d,DST)xlab_i(o,d);
(all,i,IND)(all,o,OCC)(all,d,DST)xlab(i,o,d);
(all,d,DST)r_hours_pers(d);
(all,d,DST)xlab_io(d);
(all,o,OCC)(all,d,DST)wlab_i(o,d);
(all,i,IND)(all,o,OCC)(all,d,DST)plab(i,o,d);
(all,d,DST)emp_hours(d);
(all,d,DST)pfin_hou(d);
(all,o,OCC)(all,s,SKILL)(all,d,DST)marketPERS(o,s,d);
(all,d,DST)aggmarketPRS(d);
  (all,o,OCC)(all,s,SKILL)(all,d,DST)mktSkillOCC(o,s,d);
(all,d,DST)agglabmkt(d);
(all,d,DST)aggxlab(d);
(all,d,DST)emp_pers(d);
(all,i,IND)(all,d,DST)pers_sh_o(i,d);
(all,d,DST)ffskill_wpop(d);
(all,s,SKILL)(all,d,DST)f_skill_wpop(s,d);
(all,i,IND)(all,o,OCC)(all,d,DST)flab(i,o,d);
(all,d,DST)flab_io(d);

```

GEMPACK software enables us to update a model during a simulation. Multi-step solution methods rely on this updating procedure. By using multi-step solutions, we are able to combine the relative simplicity of linearized algebra (often expressed in Equation statements in TABLO in percentage change terms) with solution accuracy comparable to that of computing equations in the levels form. The Update statements which follow update coefficients from the core database. Note that the syntax of each Update statement, unlike Equation statements, does not represent the underlying algebra. For example, the Update for WAGEBILL implies

$$\text{WAGEBILL}_1 = \text{WAGEBILL}_0 * [1 + (\text{xlabos} + \text{plabos})]$$

(dropping the dimensional subscripts).

```
Update (all,o,OCC)(all,s,SKILL)(all,d,DST) WAGEBILL(o,s,d) = xlabos(o,s,d) *
plabos(o,s,d);
(all,o,OCC)(all,s,SKILL)(all,d,DST) PERS(o,s,d) = pers_iSUP(o,s,d);
(all,i,IND)(all,o,OCC)(all,d,DST) PERSIND(i,o,d)=pers_sDEM(i,o,d);
```

Next, we introduce the Equation statements of this labour module. Note that each equation is named after a variable: if the relevant variable is pfarm, for example, the appropriate name is E_pfarm. This naming convention enables us to devise a default choice of exogenous variables automatically within GEMPACK. In a model consisting of n equations and $n+m$ variables, we can solve for n variables, leaving m exogenous variables. The TabMate program within GEMPACK can devise a suitable default list of the m exogenous variables if we follow the equation naming convention.

Scenarios

The labour module draws on the common GEMPACK method of making equations redundant by using “shift” variables. Depending on the closure, or choice of exogenous variables, the labour market module can be “switched on” or “switched off”. In this document, we refer to an “unrestricted” scenario in which the labour market module is switched off, and a “restricted” scenario in which the labour market module is switched on. To have the labour market module switched off is effectively to assume perfectly elastic, or *unrestricted*, supply of labour by occupation.

Although there appear to be a great many variables for which closure settings differ in the unrestricted and restricted scenarios, the closure files at the end of this document illustrate that we can switch between the two scenarios by implementing just six swaps. In these closures, the baseline represents the unrestricted scenario and the policy represents the restricted scenario.

We introduce many price mechanisms in CGE models through the use of constant elasticity of substitution (CES) or, on the supply side, constant elasticity of transformation (CET) equations. A CES equation follows the form

$$x(\text{specific}) = x(\text{average}) - \text{SIGMA}(p(\text{specific}) - p(\text{average})) \quad (1)$$

The specific demand x , be that a specific occupation or skill, is equal to the demand for the composite or average occupation or skill minus the price of the specific input relative to the average price of the composite input, in percentage change terms. SIGMA is the elasticity of substitution between different specific inputs.

Part 1: Demand and supply for labour units

!1 of JD's extension!

```
Equation E_plabos_s #Skill composite effective price #
(all,o,OCC)(all,d,DST)ID01[WAGEBILL_S(o,d)]*plabos_s(o,d)=
sum{s,SKILL,WAGEBILL(o,s,d)*plabos(o,s,d)};
```

!2!

```
Equation E_xlabosk # Skill shift #
(all,o,OCC)(all,s,SKILL)(all,d,DST)xlabosk(o,s,d) =
SIGMALABSK*[plabos(o,s,d) - plabos_s(o,d)];
```

!3!

```
E_xlab_iDEM # Aggregate demand, OCC x Skill #
(all,o,OCC)(all,s,SKILL)(all,d,DST)xlab_iDEM(o,s,d) =
xlab_i(o,d) - xlabsk(o,s,d);
```

A CES equation is appropriate for describing the demands of users, whose choice of specific inputs will depend on the price of a specific input relative to the average price of substitutable inputs. Equation E_plabos_s computes the skill-composite average price by occupation and region. In E_xlabsk, we see that the amount of labour hired of a specific skill type depends on the price of that skill type relative to the skill-composite price of labour plabos_s. Equation E_xlab_iDEM computes skill-specific demands as being equal to skill-composite demands minus the relative price-induced skill shift computed in E_xlabsk.

!4!

```
E_plab_ave # Average price received by skill cohort s #
(all,s,SKILL)(all,d,DST)
ID01[WAGEBILL_0(s,d)] * plab_ave(s,d) = sum{o,OCC, WAGEBILL(o,s,d)
*plabos(o,s,d)};
```

!5!

```
E_xlabos # Occ transformation frontier #
(all,o,OCC)(all,s,SKILL)(all,d,DST)
xlabos(o,s,d) = xlabo(s,d) + SIGMA_LABCET *{plabos(o,s,d) - plab_ave(s,d)};
```

The equations E_plab_ave and E_xlabos define CET transformations between occupations for a given skill level. E_xlabos is a supply equation, in which occupation-specific supplies rise as the occupation-specific wage rises relative to the average wage plab_ave of all occupations for a given skill type. Note that the expansion term, xlabo, is the wage-bill weighted sum of employment of skill s in region d.

Part 2: Translation of labour units into persons

!6!

```
E_xlabo # Total emp by skill, person weights #
(all,s,SKILL)(all,d,DST)ID01[PERS_0(s,d)]*emp_pers_sk(s,d)=
sum{o,OCC,PERS(o,s,d)*pers_iSUP(o,s,d)} + f_pers_i(s,d);
```

Equation E_xlabo is an add-up of persons employed across occupations. The terms xlabo and emp_pers_sk, both defined over skill and region, differ in the weights used to add them up. In the *restricted* scenario (in which the labour module is switched on), this equation forces employment by occupation and skill to add up to the exogenously determined total employment by skill. In this closure the variable f_pers_i is exogenous. However, in the *unrestricted* scenario (in which the labour market module is switched off), the restriction on skills is not imposed and the variable f_pers_i is endogenous, making this equation redundant.

```
Variable (all,o,OCC)(all,d,DST)xlab_iPRS(o,d);
!7!Equation E_xlab_iPRS
(all,o,OCC)(all,d,DST)ID01[PERSIND_I(o,d)]*xlab_iPRS(o,d)=
sum{i,IND, PERSIND(i,o,d)*xlab(i,o,d)};
```

!8!

```
E_pers_iDEM # Persons supplied, OCC x Skill#
(all,o,OCC)(all,s,SKILL)(all,d,DST)pers_iDEM(o,s,d) + r_hours_pers(d) =
xlab_iPRS(o,d) - xlabsk(o,s,d);
```

Total demand for persons by occupation and skill, pers_iDEM, cannot be derived by simply adding up demand for labour units. The first part of the transformation from labour units to persons is achieved by E_xlab_iPRS, in which labour units are aggregated using person weights. The second part is achieved in E_pers_iDEM, in which the term xlabsk provides an adjustment for the changing skill composition of labour in each occupation. The term r_hours_pers (ratio of hours to persons) translates the physical unit of labour from hours into persons.

Part 3: Market clearing equations

!9!

Equation E_pers_iSUP

```
(all,o,OCC)(all,s,SKILL)(all,d,DST)marketPERS(o,s,d)=
pers_iSUP(o,s,d) - pers_iDEM(o,s,d);
```

!10!

```
E_aggmarketPRS (all,d,DST)aggmarketPRS(d)=emp_pers(d) - emp_pers_o(d);
```

!11 !

E_emp_pers_o #Regional supply add-up, persons#

```
(all,d,DST)PERS_OS(d)*emp_pers_o(d) = sum{o,OCC,QPERS_S(o,d)*emp_persoSUP(o,d)};
```

!12!

E_emp_hours # Regional employment aggregate, hours #

```
(all,d,DST)PERSIND_OI(d) *emp_pers(d) =sum{o,OCC,PERSIND_I(o,d)*pers_isDEM(o,d)};
```

With marketPERS exogenous, equation E_pers_iSUP imposes market clearing, or demand = supply, for every occupation, skill and region. Under this setting, the following equation E_aggmarketPRS is redundant: if every individual market is balanced, then the sum of supply and demand will also be balanced. Therefore in the *restricted* scenario, aggmarketPRS is endogenous.

In the *unrestricted* scenario, marketPERS is endogenous, meaning that at the occupation, skill and region level, supply and demand of labour are determined separately. However, we require aggmarketPRS to be exogenous in this closure. The reason will become clear when we introduce equation E_r_skill_wpop

Equations E_emp_pers_o and E_emp_hours simply provide the definitions for the terms emp_pers_o (total regional supply) and emp_pers (total regional demand) respectively.

!13!

E_mktSkilOcc

```
(all,o,OCC)(all,s,SKILL)(all,d,DST)mktSkilOcc(o,s,d)=xlab_iDEM(o,s,d)-
xlabos(o,s,d);
```

!14!

E_fflabo

```
(all,d,DST)agglabmkt(d) =xlab_io(d) - aggxlab(d);
```

!15!

E_aggxlab

```
(all,d,DST)WAGEBILL_OS(d)*aggxlab(d) =sum{s,SKILL,WAGEBILL_O(s,d)*xlabo(s,d)};
```

By setting mktSkilOcc exogenous, equation E_mktSkilOcc imposes the market clearing condition on labour units. It is necessary to have explicit market clearing conditions on both

persons and labour units in order to correctly link the change in supply of persons by skill (typically exogenous) to the expansion term $xlabo(s,d)$ in the CET equation E_xlabos . This is the case in the *restricted* scenario. Equation E_fflabo is redundant if supply = demand for every occupation and skill. Therefore in the *restricted* scenario, $agglabmkt$ is endogenous.

However, in the *unrestricted* scenario, $mktSkilOcc$ is endogenous and $agglabmkt$ is exogenous.

Equation $E_aggxlab$ provides the definition for $aggxlab$, which appears on the RHS of E_fflabo . The reason E_fflabo is named after a variable that it does not appear to explain will become clearer when we discuss $E_f_pers_i$. Note that total demand for labour, $xlab_io$, is determined in the core model.

Part 4: Exogenous setting of skills supply

```
!16!
Equation E_emp_pers_sk # Ratio of skill supply to working population#
(all,s,SKILL)(all,d,DST) r_skill_wpop(s,d) = emp_pers_sk(s,d) - emp_pers_o(d);
!16a!
E_r_skill_wpop # Average of skill switches for given wpop #
(all,s,SKILL)(all,d,DST) r_skill_wpop(s,d) = f_skill_wpop(s,d) - avE_skilsw(d) +
ffskill_wpop(d);
```

A typical form of analysis to undertake using the labour market module concerns the skills composition of the workforce over time. There is often discussion of skills in short supply or excess supply. This may arise from a change in the composition of demand for skills over time, with a sluggish response in terms of skills training.

A simple way to impose exogenous shocks to the supply of skills is to do so via the share of skill groups in the labour force. This avoids the problem of imposing growth rates on the skill groups that are inconsistent with the growth in the aggregate or regional labour force. Equation $E_emp_pers_sk$ defines r_skill_wpop as the aggregate skills shares of total regional employment supplies. Equation $E_r_skill_wpop$ introduces a scalar for each region, $ffskill_wpop(d)$, which provides a correction to the ratio such that the sum of the ratios is one in every region. The modeller will typically exogenise and shock the term f_skill_wpop .

Part 5: Links to the core model

```
!17!
E_plab_i (all,o,OCC)(all,d,DST)plab_i(o,d) = wlab_i(o,d)-xlab_i(o,d);

!18!
E_plabos # F_plabos exogenous = core on, Lab module off; swap with flabmkt #
(all,o,OCC)(all,s,SKILL)(all,d,DST) plabos(o,s,d)+f_plabos(o,s,d) = plab_i(o,d);
!19!
E_ff1lab #Switch off E_plab with swap ff1lab = flab#
(all,i,IND)(all,o,OCC)(all,d,Dst)plab(i,o,d) +ff1lab(i,o,d) = plabos_s(o,d);

!copied from core model!
Equation E_plab # Wage fixing rule #
(all,i,IND)(all,o,OCC)(all,d,DST)
plab(i,o,d) = pfin("hou",d) + flab(i,o,d)+ flab_io(d);
```

Equation E_plab_i computes the occupational wage, averaged over industries, based on core model variables.

In the *unrestricted* scenario, we assume that wages move uniformly over all occupations and industries. In this scenario we require

```
plab(i,o,d) = pfin("hou",d) + flab_io(d);
```

which is achieved by exogenising $f_{lab}(i,o,d)$ in E_{plab} . Note that the regional scalar $f_{lab_io}(d)$ is required to accommodate forecasts for aggregate regional employment, and the term $p_{fin}("hou",d)$ represents the regional CPI.

Equation E_{plabos} links wages by occupation and skill to occupational wages calculated in the core part of the model. In the *unrestricted* scenario, we require

```
plabos(o,s,d) = plab_i(o,d)
```

meaning that f_{plabos} needs to be exogenous in E_{plabos} .

Finally, E_{fflab} defines p_{lab} , the regional wage at the industry by occupation level. In the *unrestricted* scenario, this wage is already defined in E_{plab} from the core model. Hence this equation is redundant and the shift term $fflab$ is endogenous.

Alternatively, in the *restricted* scenario, we use wage movements derived from the labour market module to inform wage movements in the core model. That is, we require

```
plab(i,o,d) = plabos_s(o,d);
```

This is achieved by exogenising $fflab$ in E_{fflab} . Note that we assume wage movements are uniform across industries.

Having determined p_{lab} in this way, equation E_{plab} in the core model is redundant. Therefore we endogenise f_{lab} . Equation E_{plabos} is also redundant, so we endogenise f_{plabos} .

!20!

Equation $E_{f_pers_i}$

```
(all,s,SKILL)(all,d,Dst)xlabo(s,d)=flabo(s,d) + emppers_sk(s,d) +fflabo(d);
```

!21!

$E_{marketPERS}$

```
(all,o,OCC)(all,s,Skill)(all,d,Dst)pers_iSUP(o,s,d)= emppers_sk(s,d)+fsup(o,s,d);
```

!22!

```
 $E_{emp\_pers}$  (all,d,Dst)rhours_pers(d) = emphours(d) - emppers(d);
```

!23!

```
 $E_{xlabsk\_s}$  (all,o,OCC)(all,d,Dst)QPERS_S(o,d)*xlabsk_s(o,d)=  
sum{s,Skill,PERS(o,s,d)*xlabsk(o,s,d)};
```

!24!

```
 $E_{pers\_sDEM}$  # i/o persons demanded # (all,i,IND)(all,o,OCC)(all,d,Dst)
```

```
pers_sDEM(i,o,d) + rhours_pers(d) =xlab(i,o,d)- xlabsk_s(o,d);
```

!25!

```
 $E_{pers\_isDEM}$  # i/o persons demanded # (all,o,OCC)(all,d,Dst)
```

```
ID01[PERSIND_I(o,d)]*pers_isDEM(o,d)= Sum{i,Ind,PERSIND(i,o,d)*pers_sDEM(i,o,d)};
```

The term x_{labo} in $E_{f_pers_i}$ is the expansion term in the CET equation E_{xlabos} , that is, the wage-weighted sum of employment over occupation. In the *restricted* scenario, $E_{f_pers_i}$ is not required because x_{labo} is determined by E_{xlabo} , that is, the link between persons and labour units is made at the occupation by skill level. In the *restricted* scenario therefore f_{labo} is endogenous.

In the *unrestricted* scenario, with no market clearing at the occupation by skill level, E_f_pers_i is required to find a solution for xlabo. The assumption in the *unrestricted* scenario is that labour units move in proportion with persons employed, that is xlabo moves in proportion with emp_pers_sk. As such, flabo is exogenous and fflabo is endogenous in the unrestricted scenario. Note that fflabo needs to be endogenous because an aggregate market clearing condition is already imposed in E_fflabo.

Equation E_marketPERS is activated during the *unrestricted* run, when fsup is exogenous and marketPERS endogenous. In this run we assume that there is no supply substitution between occupations. When marketPERS is endogenous, there may be excess supplies or excess demands of occupations by skills in each region. During the *restricted* run, pers_iSUP is determined by E_pers_iSUP. To make E_fsup redundant, fsup is endogenous.

Equation E_pers_sDEM is of a similar form to E_pers_iDEM, except that it includes an industry dimension instead of a skills dimension in computing occupations demanded in each region.

Part 6: Miscellaneous labour market aggregates

```
E_pers_is #Add-up of industry demands, persons#
(all,o,OCC)(all,d,DST) ID01[PERSIND_I(o,d)]*pers_is(o,d)=
sum{i,IND,PERSIND(i,o,d)*pers_sDEM(i,o,d)};

!26!
E_emp_persoSUP #Add-up of occupational supplies, persons#
(all,o,OCC)(all,d,DST) ID01[QPERS_S(o,d)]*emp_persoSUP(o,d)=
sum{s,SKILL,PERS(o,s,d)*pers_iSUP(o,s,d)};
!27!
E_pers_osDEM #Ind persons demanded# (all,i,IND)(all,d,DST)
ID01[PERSIND_O(i,d)]*pers_osDEM(i,d) = sum{o,OCC, PERSIND(i,o,d) *
pers_sDEM(i,o,d)};
!28!
E_pers_sh
(all,o,OCC)(all,d,DST)
pers_sh(o,d) = pers_isDEM(o,d) - emp_pers(d) + f_pers_sh(d);
!29!
E_pers_sh_o (all,i,IND)(all,d,DST)
pers_sh_o(i,d) = pers_osDEM(i,d) - emp_pers(d) +f_pers_sh_oi(d);
```

Equation E_emp_persoSUP adds up occupational supplies across the skills dimensions. E_pers_osDEM adds up industry demands across the occupational dimension.

Equations E_pers_sh and E_pers_sh_o are redundant in this module. They are carried over from the original skills module in the national version of the model, in which historical (observed) data are imposed on the model. Shocks to pers_sh and pers_sh_o ascribe these data. The two variables made endogenous in the historical years of the policy run of the national by swaps with labour technical change variables of appropriate dimensions.

Part 7: Remaining equations from the core dynamic model

```
!Equations from core dynamic model carried over to ease replication of main model
shocks!
Variable (all,q,DST)labslack(q);
(all,q,DST)flabsup(q) # State Labour supply #;
```



```

Equation E_flab_io # Inter-regional Labour migration or Labour supply #
(all,d,DST) ! 1.0 represents an elasticity value !
xlab_io(d) = 1.0*averealwage(d) + flabsup(d) + labslack(d);

Variable (all,i,IND)(all,d,DST)xlab_o(i,d);
(all,i,IND)(all,d,DST)plab_o(i,d);

Equation E_xlab # Demand for Labour by industry and occ group #
(all,i,IND)(all,o,OCC)(all,d,DST)
xlab(i,o,d) = xlab_o(i,d) - 0.5 !SIGMALAB!*[plab(i,o,d) - plab_o(i,d)];
Coefficient (all,i,IND)(all,o,OCC)(all,d,DST) LAB(i,o,d) # Wage matrix #;
Read LAB from file Janine header "1LAB";
Update (all,i,IND)(all,o,OCC)(all,d,DST) LAB(i,o,d)=plab(i,o,d)*xlab(i,o,d);

Coefficient (all,o,OCC)(all,d,DST) LAB_I(o,d) # Total wages by occ #;
Formula (all,o,OCC)(all,d,DST) LAB_I(o,d) = sum{i,IND, LAB(i,o,d)};
Equation E_wlab_i (all,o,OCC)(all,d,DST)
LAB_I(o,d)*wlab_i(o,d)=sum{i,IND, LAB(i,o,d)*[plab(i,o,d)+xlab(i,o,d)]};

E_xlab_i (all,o,OCC)(all,d,DST)
LAB_I(o,d)*xlab_i(o,d)=sum{i,IND, LAB(i,o,d)*xlab(i,o,d)};

```

Finally, this module repeats some equations found in the core of the dynamic TERM model. This is for convenience, to ease the replication in this module of simulations undertaken with the full model. E_plab is the standard wage fixing equation of the full model. Equation E_flab_io solves for the regional wage shifter flab_io which is usually endogenous in the baseline and exogenous in the policy run. E_xlab is the CES equation solving for occupation-specific labour demands in the core model.

If we run the full TERM model, as discussed below, with different skill and occupational dimensions than the standalone module, we can still feed in shocks from TERM to the module so long as no variables with skill or occupational dimensions are solved outside the module. The equations E_wlab_i and E_xlab_i solves for wlab_i and xlab_i, which both have an occupational dimension.

Replicating the baseline of 2014 from the full TERM model

This is equivalent to implementing the unrestricted scenario. As is evident in the preceding paragraphs, not all key variables in the skills module are solved within the module. Some variables appearing in the module are solved in the core part of the TERM model.

*Creating a *.cmf file for 2014*

One of the tasks in devising a CGE model or module as outlined above is in choosing appropriate combinations of endogenous and exogenous variables. Earlier, we noted that we can obtain a default closure for the model. We do so by opening LabModule.tab in TABMATE. Once we have generated a model using either *Tablo Sti* or *Tablo code* in the top menu, under *Tools* we go to *Closure*. This creates a file LabModule.clo. Section 4 of the *.clo file provides us with a default closure. It names the exogenous variables and then states that the remaining variables are endogenous.

TABMATE also creates a default *.cmf file for us under *Tools, View/Create CMF*. This is a blank canvas. The default *.cmf file asks us to insert a closure, shocks and a verbal description.

Table 1: ModuleBase2014.cmf

*! This is a template CMF file -- you will need to edit filenames
! and the closure and shocks.*

```

auxiliary files = LabModule;
check-on-read elements = warn; ! Optional: very often needed
cpu=yes ; ! Optional: Reports CPU times for various stages
log file = yes; ! Optional

! Input files:
File Janine = Janine97x4.har; !
! Updated files:
Updated File Janine = Janine.UPD;
start with mmnz = 70000;
! Solution method
method = Euler ;
steps = 4;

! Insert closure below
Exogenous      agglabmkt ; ! DST
Exogenous      alabos ; ! OCC*SKILL*DST Occupation demand change
Exogenous      averealwage ; ! DST
Exogenous      ffskill_wpop ; ! DST
Exogenous      flab ; ! IND*OCC*DST
Exogenous      flabo ; ! SKILL*DST
Exogenous      flabsup ; ! DST State labour supply
Exogenous      fsup ; ! OCC*SKILL*DST
Exogenous      f_pers_sh ; ! DST
Exogenous      f_pers_sh_oi ; ! DST
Exogenous      f_plabos ; ! OCC*SKILL*DST f_plabos exogenous = core on, Lab module off
Exogenous      f_skill_wpop ; ! SKILL*DST
Exogenous      labslack ; ! DST
Exogenous      pfin ; ! FINDEM*DST
Exogenous      plab_o ; ! IND*DST
Exogenous      r_hours_pers ; ! DST
Exogenous      xlab_io ; ! DST
Exogenous      xlab_o ; ! IND*DST
Rest endogenous; ! end of TABmate automatic closure

!Baseline swaps from structural closure
swap flabsup = emp_hours;
swap ffskill_wpop = aggmarketPRS;

Verbal Description = 2014 policy run ;
! Insert shocks below
shock emp_hours = select from file BaseShk14.rsl header "0208" ;
shock averealwage = select from file BaseShk14.rsl header "0110";
shock pfin = select from file BaseShk14.rsl header "0085";
shock xlab_o= select from file BaseShk14.rsl header "0034";
shock plab_o= select from file BaseShk14.rsl header "0033";
shock xlab_io = select from file BaseShk14.rsl header "0092";

```

Table 1 shows the full *.cmf file used to replicate the solution of the 2014 base of the full TERM skills model for the standalone skills module. Two changes are made to the default closure. We treat the quantity of labour used in each region as exogenous through the swap `swap flabsup = emp_hours;`

By making `aggmarketPRS` exogenous in the following line of the *.cmf file, we set aggregate labour demand equal to aggregate labour supply.

The shocks ascribed to the module are of two types. First, there are the shocks imposed on the full TERM skills model. Only `emp_hours` is in this group. The second group includes those variables that are exogenous in the module but endogenous in the full model. These are `averealwage`, `pfin`, `xlab_o`, `plab_o` and `xlab_io`. In `RunDynam`, these are written to *.rsl files, which have been renamed for convenience above.

Table 2: Part of ModulePol2014.cmf

The following swaps, applied to the unrestricted closure, create the restricted closure.

```
!Policy closure swaps
swap f_plabos = mktSkil0cc; !1 OCCxSKILLxDST clear Labour markets, endo wage
swap flabo = f_pers_i; !3 SKILLxDST link flabo to exo Labour supply
swap fsup = marketPERS ;!2 OCC*SKILL*DST
swap flab=ff1lab; !4 INDxOCCxDST alternative wage setting
swap aggmarketPRS = flab_io; !5 DST clear Labour markets (persons), link flab to
exo Labour supply
swap agglabmkt = fflabo; !6 DST

Verbal Description = 2014 policy run ;

! Insert shocks below
tshock mktSkil0cc = uniform 0;
tshock marketPERS = uniform 0;
tshock f_pers_i = uniform 0;
tshock ff1lab= uniform 0; !alternative wage setting (i,o)
tshock f_skill_wpop = select from file RegShock.har header "SSHK" slice "Y2014" ;

!shocks carried over from baseline simulation
shock flab_io = select from file ModulePol2014.sol header "0041" ;
shock ff1lab = select from file ModulePol2014.sol header "0019" ;
shock f_pers_i = select from file ModulePol2014.sol header "0020" ;
shock marketPERS = select from file ModulePol2014.sol header "0031" ;
shock mktSkil0cc = select from file ModulePol2014.sol header "0033" ;
shock fflabo = select from file ModulePol2014.sol header "0045" ;

shock emp_hours = select from file PolShk14.rsl header "0208" ;
shock averealwage = select from file PolShk14.rsl header "0110";
shock pfin = select from file PolShk14.rsl header "0085";
shock xlab_o= select from file PolShk14.rsl header "0034";
shock plab_o= select from file PolShk14.rsl header "0033";
shock xlab_io = select from file PolShk14.rsl header "0092";
```

When dynamics are introduced to a CGE model, the time dimension usually entails formal links between stocks and flows. For example, capital stocks in year t are pre-determined by the depreciated capital stocks plus investment flows of the previous year in each industry. However, the dynamic linkages are not solved in the standalone module of the model. Variables such as capital and investment do not appear in any equations in the module.

What is relevant in the skills module is the method we use to depict dynamics. First, we run the baseline, of which `ModuleBase2014.cmf` depicts a single year. Then we introduce closure swaps so that forecast variables, which are exogenous in forecast but normally endogenous in modelling, become endogenous with the swapped variable

An example of a variable that ought to be endogenous in usual analysis is regional employment, `xlab_io`. However, this is not solved in the skills module. Rather it is exogenous in forecast or base in the full model, and endogenous in the policy run of the full model. Consequently, we need to introduce a base shock to the `*.cmf` base shocks file and a policy shock to the `*.cmf` policy shocks file of the skills module. As shown in tables 1 and 2, the base shock is read from `BaseShk14.rsl` and the policy shock is read from `PolicyShk.rsl`.

We could devise a full dynamic run by preparing shocks files for all years of the run for the base and forecast. This would become a somewhat challenging task, as we would then need to assemble the year-by-year shocks for each year. Note that policy and base shocks brought over from the dynamic model are year specific.

Having devised the year-by-year in-files, our next task would be to compare the year-by-year solutions for the baseline with the policy run. In the context of labour market forecasting, year-by-year solutions themselves are of interest. In other types of modelling, the main interest is in the deviation of the solution from the baseline.

At this point, without additional software to ease the computing tasks, we are limiting dynamic simulations to a small group of GEMPACK specialists. This is not our aim, as we want others to use our models and software. We also wish to encourage users to check and interpret results, rather than put all their efforts into generating results.

Running the labour module in RunDynam

RunDynam software automates many of tasks required to put together a dynamic simulation.

To follow how commands in RunDynam are organised, we turn back to the file `ModuleBase2014.cmf` in Table 1.

Under *Model/Data* on the menu, we need to choose a model by clicking on *Change Model*. We choose a file for input data by clicking on *File Janine*. This serves the same function as the following command line in the `cmf` file:

```
File Janine = Janine97x4.har;
```

On the *Sim Overview* page, there are a number of self-explanatory decisions to make. These include the start year and the number of years (or other time intervals) in the simulation.

A less obvious choice is that of the three letter names for each of the base, re-run and policy names. It is our practice to re-run the baseline simulation with the policy closure. This means that where there are closure swaps, variables that were endogenous in base and are to be exogenous in the re-run and policy modes must be shocked in the new modes by the simulated change of the base run. In Figure 1, the three letter designation for each of the runs is the three letters “MOB”. Each solution and database file will be given a name unique to the run and year. For example, the *Janine* file generated for 2014 in base is `JAN-MOBb-`

2014.har. The re-run file for the same year is JAN-MOBB-MOBr-2014.har and the corresponding policy file is JAN-MOBB-MOBr-MOBp-2014.har.

This naming system has implications for file management. There is no need to run the entire model again if the policy shocks are changed but the closure is not. Moreover, by assigning a new set of three letters to *Policy [P]*, the existing files generated by the first policy run are preserved. If *Policy [P]* is not changed, all the policy simulations of the previous run will be over-written.

Figure 1: RunDynam Sim Overview page

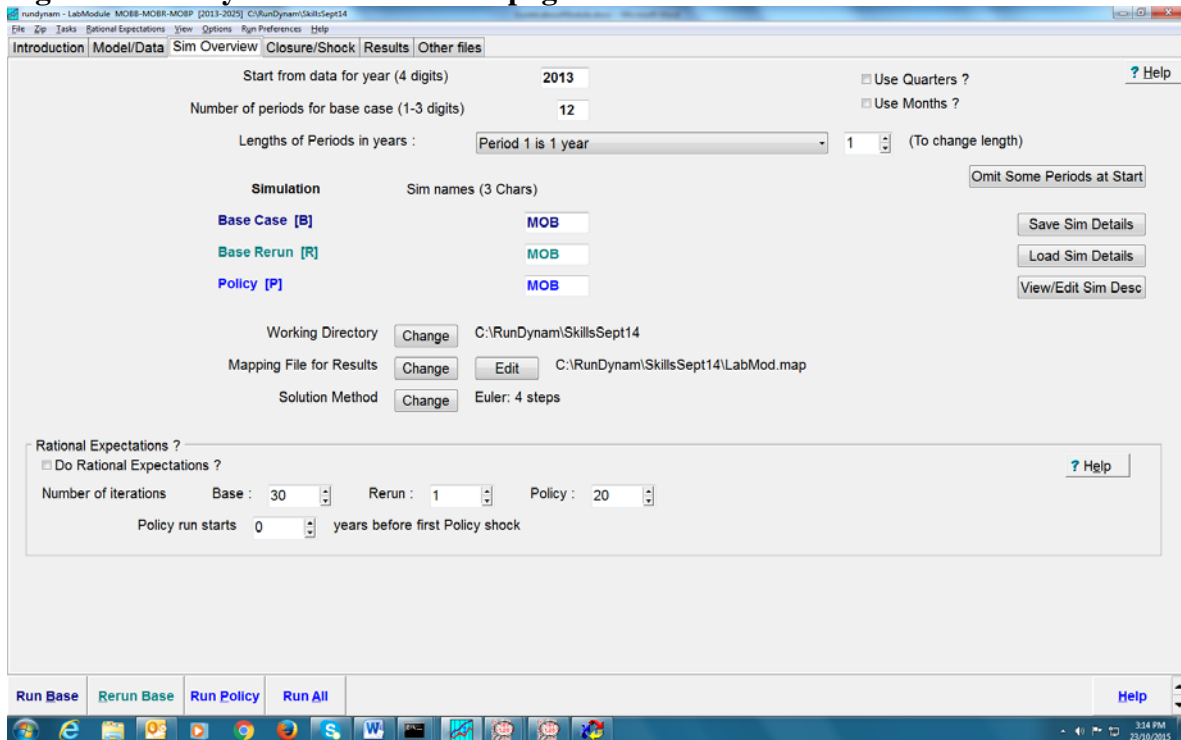


Figure 2: RunDynam Closure/Shock page

Part	Base Closure [MOB]	Base Shocks [MOB]	Policy Closure [MOB]	Policy Shocks [MOB]
Pattern	MOBYYYYY.CLS	MOBYYYYY.BSH	MOBYYYYY.CLS	MOBYYYYY.PSH
CMFStart	CMFSTARTB	N/A	CMFSTARTR	CMFSTARTP
Common	MODSTRUC.CLS	FORECAST.BSH	MODSTRUC.CLS	none
2014	MODBASE.CLS	common	MODPOLICY.CLS	TSHKBIG.PSH
2015	MODBASE.CLS	common	MODPOLICY.CLS	TSHKBIG.PSH
2016	MODBASE.CLS	common	MODPOLICY.CLS	TSHKBIG.PSH
2017	MODBASE.CLS	common	MODPOLICY.CLS	TSHKBIG.PSH
2018	MODBASE.CLS	common	MODPOLICY.CLS	TSHKBIG.PSH
2019	MODBASE.CLS	common	MODPOLICY.CLS	TSHKBIG.PSH
2020	MODBASE.CLS	common	MODPOLICY.CLS	TSHKBIG.PSH
2021	MODBASE.CLS	common	MODPOLICY.CLS	TSHKBIG.PSH
2022	MODBASE.CLS	common	MODPOLICY.CLS	TSHKBIG.PSH
2023	MODBASE.CLS	common	MODPOLICY.CLS	none
2024	MODBASE.CLS	common	MODPOLICY.CLS	none
2025	MODBASE.CLS	common	MODPOLICY.CLS	none

The button *Load Simulation Details* enables us to restore a previously completed simulation. If we are after database or solution files, there is no need to run the simulation again to restore these. The default name of the file that saves simulation details by default is *<model name>- <base name B>- <rerun name R>-<policy name P>.ds1*. In the case of our example, the default name is *LabModule-MobB-ModR-MobP.ds1*. If we wish to use simpler names, we can do so via the *Save Sim Details* button on the page.

The *Working Directory* is the directory in which all the closure (*.cls) and shocks files (*.bsh and *.psh) must reside. The *Mapping File for Results* must include a subset of the available variables of the model. Available variables are those which have not been omitted or substituted out if the model has been condensed.

The *Solution Method* performs the same task as the following *.cmf command (using the solution method we have chosen above as an example):

```
! Solution method
method = Euler ;
steps = 4;
```

On the *Closure/Shocks* page, the *Base Closure* column includes two files which cover the entire closure for each year. The structural closure copied from LabModule.clo, namely Modstruct.cls, is imposed on all years: hence, it appears in the *Common* row. Each year of the base adds the same file, with additional endogenous/exogenous swaps appearing in Modbase.cls to complete the baseline closure.

The second column brings in baseline shocks. These are a repeat of the shocks given in ModuleBase2014.cmf. The main difference between the shocks in Forecast.bsh and the shocks in ModuleBase2014.cmf is that the former point to file with a year dimension. That is,

in each shock statement, <year> appears, indicating that shocks may vary from year to year in the shocks file. Modstruc.cls is used again. There are more swaps so that the policy closure entails some market-clearing via price adjustments.

The fourth column includes the policy shocks. The shock to `f_skill_wpop` is assigned for each year, as it was for year 2014 in `ModulePol2014.cmf`. The additional shocks come from the full TERM skills module simulation. They are saved as year on year deviations. Note that the policy shocks are ascribed as *tshocks*. The deviations collected from the full model for variables that are endogenous in the full model and appear in the skills module as exogenous are ascribed as *ashocks*.

```
ashock <year> pfin = select from file devyony_4.har header "0085";
ashock <year> plab_o= select from file devyony_4.har header "0033";
ashock <year> averealwage = select from file devyony_4.har header "0110";
ashock <year> xlab_o= select from file devyony_4.har header "0034";
ashock <year> xlab_io = select from file devyony_4.har header "0092";
```

The fourth column includes the policy shocks. The policy shocks are ascribed as *tshocks* to `f_skill_wpop`, `mktSkilOcc`, `marketPERS` and `f_persi` are assigned for each year, as they were for year 2014 in `ModulePol2014.cmf`.

The additional shocks come from the full TERM skills module simulation. They are saved as year on year deviations. The deviations collected from the full model for variables that are endogenous in the full model and appear in the skills module as exogenous are ascribed as *ashocks*.

Since no solutions with either occupational or skill dimensions from the full TERM skills model are used in the standalone labour module run, it is possible to run the TERM skills version with an aggregation in the occupational and skill dimensions, and then run the labour module with the full 97 occupations and 56 skills. The advantage of this approach is that an aggregation with 13 occupations and 10 skills in the full TERM skills model solves about five times as quickly as the full dimensions. A disadvantage is that shocks must be introduced to the labour module version from the TERM run. However, once the *.bsh and *.psh shocks are set up, this process can be undertaken quickly. However, the current full version of the TERM skills database (97 occupations and 56 skills for 22 industries in 4 bottom-up regions) takes less than 15 minutes to run on relatively new PCs in total for all years from 2014 to 2025.² The aggregated version with 13 occupations and 10 skills solves in around 3 minutes, which is about the same as required for the standalone labour module to solve for all years.

It is probably better to confine LabModule to training purposes, and run the TERM skills model for usual analysis. Many of the variables examined in regional analysis such as a number of the macro variables do not appear in LabModule.

² The simulation takes less than 13 minutes for all years on a PC with an Intel 3.4 GHz dual processor, 32 gigabytes of RAM and a 64 bit operating system.

References

Dixon, P. B., & Rimmer, M. T. (2003). A New Specification of Labour Supply in the MONASH Model with an Illustrative Application. *The Australian Economic Review*, Vol. 36 (1) pp. 22-40.

Giesecke, J. A., Tran, N. H., Meagher, G. A., & Pang, F. (2015). A decomposition approach to labour market forecasting. *Journal of the Asia Pacific Economy*, 246-270.

Horridge, J. M. (2014). ORANI-G: A Generic Single-Country Computable General Equilibrium Model. available at <http://www.copsmodels.com/oranig.htm>.

Meagher, G. A., & Pang, F. (2011). Labour Market Forecasting, Reliability and Workforce Development. CoPS/IMPACT Working Paper Number G-225.

Wittwer, G. (2012). *Economic Modeling of Water: The Australian CGE Experience*. Springer.

Appendix 1: Gathering shocks file from TERM skills run for LabModule

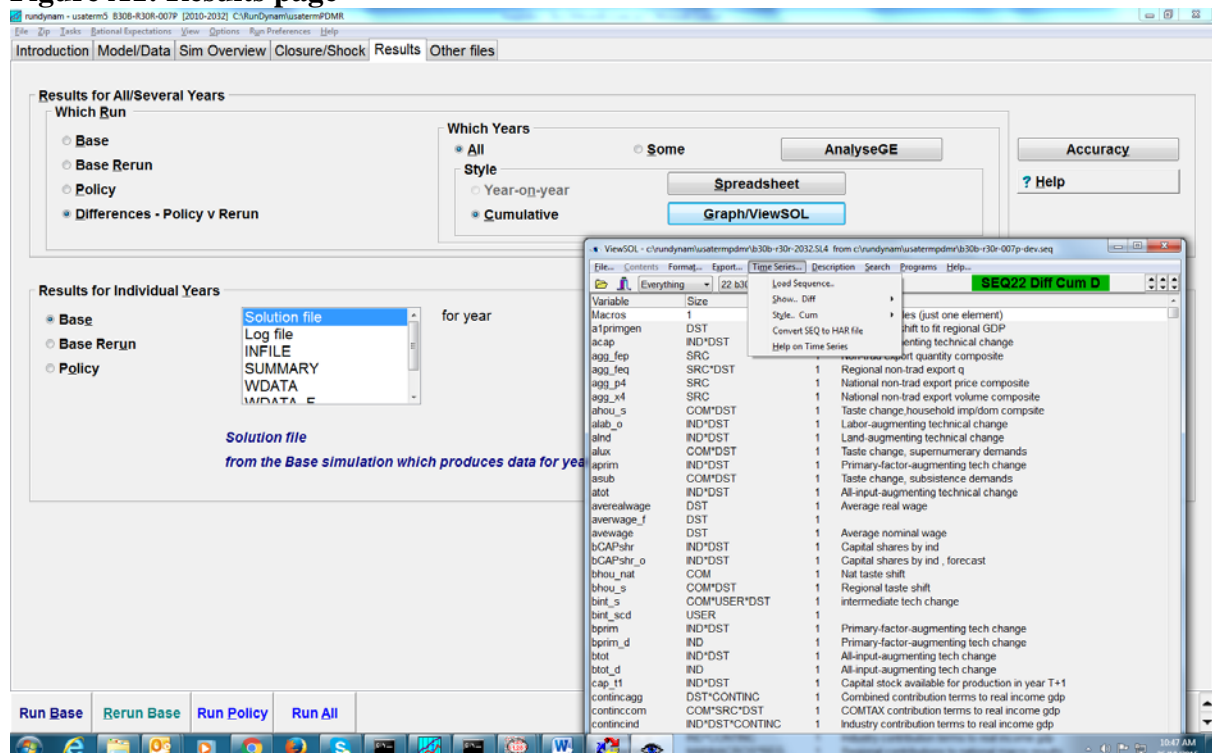
This explains how the LabModule shocks were carried over from DynTERMSk00. It may help users understand better the functions of RunDynam. Reading this may assist you in file management.

We need both the year-on-year baseline shocks, and the deviations solution file for each run from the full model (DynTERMSk00) in order to replicate the results in LabModule.

The ingredients from the DynTERMSk00 simulation are accessed in RunDynam by going into *Zip* on the top menu and choosing the appropriate zip file (*Restore Ingredients from ZIP Archive*), which may have a name such as TERMSkills.zip. When you unzip this file, it will create a directory and download all ingredients for the simulation.

Note there is no need to download the zip file each new time you wish to run the simulation or revisit results. You can restore the full simulation by going to *File* then *Load Simulation Details ...* Although simulation details are saved automatically in RunDynam, the names are not easy to follow. The default name of the file containing simulation details follows the convention <model name>_***b-***r-***p.ds1, where *** refers to the three letter/digit name given to each run on the *Sim Overview* page of RunDynam. In the current archive, each run has the three letters BIG so that the default simulation details name is DynTERMSk00-bigb-bigr-bigp.ds1. You have the option under *Sim Overview* to save simulation details with a name of your choosing.

Figure A1: Results page



After running DynTERMSK00 (the full TERM model with the skills module), on the *Results* page, we click on *Base/Year-on-year/Graph/Viewsol*. Then, in the Viewsol file, we choose

Time Series/Convert SEQ to HAR file. This gives us a complete baseline solution file in VIEWHAR format, which enables us to read in shocks in the LabModule model.³

In RunDynam, the shocks are set up automatically as inputs to the re-run and policy simulations. This implies that the only shocks we need to ascribe to LabModule from the core model (in addition to the policy simulation shocks) are the deviation shocks for variables not solved in LabModule. We obtain these with a time series conversion to a Viewhar file, this time using *Policy/Cumulative/Graph/Viewsol*.

An alternative way of viewing the solution file in deviations form: SEQ2HAR

For most applications, the use of RunDynam software provides a convenient way of viewing solutions files. In some circumstances, particularly when multiple scenarios are being analysed, the GEMPACK program SEQ2HAR is more convenient.

Consider an example in which the rerun/policy closure is identical for each of six runs. In the example below, on the Sim Overview page in RunDynam, the base run files are assigned the prefix b00 and the rerun files r00. Only the policy shocks vary between the policy scenarios: these are assigned the prefixes 001 to 006 on the Sim Overview page. The subset of model variables to be observed in the solution files are listed in the text file Skills.map.

The lines below are saved in a BATCH file with a name such AllRuns.bat. The name of the specific batch is typed into a DOS box in the working directory of RunDynam for the specific project. If the name of the file is AllRuns.bat, typing *AllRuns* in the working directory will generate each of the deviations solution files in a VIEWHAR-readable form.

```
seq2har b00b-r00r-001p-dev.seq run1.har 3 -c -Hskills.map
seq2har b00b-r00r-002p-dev.seq run2.har 3 -c -Hskills.map
seq2har b00b-r00r-003p-dev.seq run3.har 3 -c -Hskills.map
seq2har b00b-r00r-004p-dev.seq run4.har 3 -c -Hskills.map
seq2har b00b-r00r-005p-dev.seq run5.har 3 -c -Hskills.map
seq2har b00b-r00r-006p-dev.seq run6.har 3 -c -Hskills.map
```

Running multiple simulations in RunDynam using a batch file

A batch file run in the working directory could run each of the 6 policy scenarios as shown in the above box. The Sim Details *.ds1 files would have names of the form DynTERMSk00-b00b-r00r-00*p.ds1:

```
start/wait c:\RunDynam\RunDynam policy all DynTERMSk00-b00b-r00r-001p
start/wait c:\RunDynam\RunDynam policy all DynTERMSk00-b00b-r00r-002p
start/wait c:\RunDynam\RunDynam policy all DynTERMSk00-b00b-r00r-003p
start/wait c:\RunDynam\RunDynam policy all DynTERMSk00-b00b-r00r-004p
start/wait c:\RunDynam\RunDynam policy all DynTERMSk00-b00b-r00r-005p
start/wait c:\RunDynam\RunDynam policy all DynTERMSk00-b00b-r00r-006p
```

³ In creating the file ModuleBase2014.cmf referred to earlier, the *.cmf generated by the first year of the DynTERMSk base run was edited. The name of the generated file is BIGb-2014.cmf. This automatically generated file stores shocks in *.rsl files which are used in the re-run and policy runs.

Appendix 2: Skills and occupations in labour market module

In LabModule, dynamic TERM with a skills module and VUEF (Victoria University Employment Forecasting) the labour market is separated into many segments based on skills, where the term “skill” identifies post-school qualification by field and level. Each skill has a unique occupation profile. For example, within the skill “Bachelor degree: Health”, 37 per cent of employment is in the occupation “Midwifery and Nursing Professionals”, 13 per cent of employment is in “Medical Practitioners” and 11 per cent is in “Health Therapy Professionals”.

Table A1: Skills introduced to LabModule and VU-TERM

Discipline	Qualification	Skill label in database
Natural and physical sciences Information technology Engineering and related technologies Architecture and building Agriculture, environmental and related studies Health Education Management and commerce Society and culture Creative arts Food, hospitality and personal services	Postgraduate Degree	Q1SCI_Q1PGR Q2IT_Q1PGR Q3ENG_Q1PGR Q4ARCH_Q1PGR Q5QGRI_Q1PGR Q6HEAL_Q1PGR Q7EDUC_Q1PGR Q8MGT_Q1PGR Q9SOC_Q1PGR Q10ART_Q1PGR Q11FOO_Q1PGR
Natural and physical sciences Information technology Engineering and related technologies Architecture and building Agriculture, environmental and related studies Health Education Management and commerce Society and culture Creative arts Food, hospitality and personal services	Graduate Diploma/Graduate Certificate	Q1SCL_Q2GRD Q2IT_Q2GRD Q3ENG_Q2GRD Q4ARCH_Q2GRD Q5QGRI_Q2GRD Q6HEAL_Q2GRD Q7EDUC_Q2GRD Q8MGT_Q2GRD Q9SOC_Q2GRD Q10ART_Q2GRD Q11FOO_Q2GRD
Natural and physical sciences Information technology Engineering and related technologies Architecture and building Agriculture, environmental and related studies Health Education Management and commerce Society and culture Creative arts Food, hospitality and personal services	Bachelor Degree	Q1SCL_Q3BAC Q2IT_Q3BAC Q3ENG_Q3BAC Q4ARCH_Q3BAC Q5QGRI_Q3BAC Q6HEAL_Q3BAC Q7EDUC_Q3BAC Q8MGT_Q3BAC Q9SOC_Q3BAC Q10ART_Q3BAC Q11FOO_Q3BAC
Natural and physical sciences Information technology Engineering and related technologies Architecture and building Agriculture, environmental and related studies Health Education Management and commerce Society and culture Creative arts Food, hospitality and personal services	Advanced Diploma/Diploma	Q1SCL_Q4ADV Q2IT_Q4ADV Q3ENG_Q4ADV Q4ARCH_Q4ADV Q5QGRI_Q4ADV Q6HEAL_Q4ADV Q7EDUC_Q4ADV Q8MGT_Q4ADV Q9SOC_Q4ADV Q10ART_Q4ADV Q11FOO_Q4ADV
Natural and physical sciences Information technology Engineering and related technologies Architecture and building Agriculture, environmental and related studies Health Education Management and commerce Society and culture Creative arts Food, hospitality and personal services	Certificate	Q1SCL_Q5CER Q2IT_Q5CER Q3ENG_Q5CER Q4ARCH_Q5CER Q5QGRI_Q5CER Q6HEAL_Q5CER Q7EDUC_Q5CER Q8MGT_Q5CER Q9SOC_Q5CER Q10ART_Q5CER Q11FOO_Q5CER
No post school	No post school	NOPOSTSCHOOL

Table A2: Description of 3 digit ANZSCO occupations

MAN_Chi_Chie	111 Chief Executives, General Managers and	TEC_Ski_Hort	362 Horticultural Trades Workers
MAN_Far_Farm	121 Farmers and Farm Managers	TEC_Oth_Hair	391 Hairdressers
MAN_SpE_Adve	131 Advertising, Public Relations and Sales	TEC_Oth Prin	392 Printing Trades Workers
MAN_SpE_Busi	132 Business Administration Managers	TEC_Oth_Text	393 Textile, Clothing and Footwear Trades Workers
MAN_SpE_Cons	133 Construction, Distribution and Production	TEC_Oth_Wood	394 Wood Trades Workers
MAN_SpE_Educ	134 Education, Health and Welfare Services	TEC_Oth_Misc	399 Miscellaneous Technicians and Trades Workers
MAN_SpE_ICT	135 ICT Managers	COM_Hea_Heal	411 Health and Welfare Support Workers
MAN_SpE_Misc	139 Miscellaneous Specialist Managers	COM_Car_Chil	421 Child Carers
MAN_Hos_Acco	141 Accommodation and Hospitality Managers	COM_Car_Educ	422 Education Aides
MAN_Hos_Reta	142 Retail Managers	COM_Car_Pers	423 Personal Carers and Assistants
MAN_Hos_Misc	149 Miscellaneous Hospitality, Retail and Service	COM_Hos_Hosp	431 Hospitality Workers
PRO_Art_Arts	211 Arts Professionals	COM_Pro_Defe	441 Defence Force Members, Fire Fighters and Police
PRO_Art_Medi	212 Media Professionals	COM_Pro_Pris	442 Prison and Security Officers
PRO_Bus_Acco	221 Accountants, Auditors and Company Secretaries	COM_Spo_Pers	451 Personal Service and Travel Workers
PRO_Bus_Fina	222 Financial Brokers and Dealers, and Investment	COM_Spo_Spor	452 Sports and Fitness Workers
PRO_Bus_Huma	223 Human Resource and Training Professionals	CLE_Off_Cont	511 Contract, Program and Project Administrators
PRO_Bus_Info	224 Information and Organisation Professionals	CLE_Off_Off	512 Office and Practice Managers
PRO_Bus_Sale	225 Sales, Marketing and Public Relations	CLE_Per_Pers	521 Personal Assistants and Secretaries
PRO_Des_Air	231 Air and Marine Transport Professionals	CLE_Gen_Gene	531 General Clerks
PRO_Des_Arch	232 Architects, Designers, Planners and Surveyors	CLE_Gen_Keyb	532 Keyboard Operators
PRO_Des_Engi	233 Engineering Professionals	CLE_Inq_Call	541 Call or Contact Centre Information Clerks
PRO_Des_Natu	234 Natural and Physical Science Professionals	CLE_Inq_Rece	542 Receptionists
PRO_Edu_Scho	241 School Teachers	CLE_Num_Acco	551 Accounting Clerks and Bookkeepers
PRO_Edu_Tert	242 Tertiary Education Teachers	CLE_Num_Fina	552 Financial and Insurance Clerks
PRO_Edu_Misc	249 Miscellaneous Education Professionals	CLE_CIE_Cler	561 Clerical and Office Support Workers
PRO_Hea_Diag	251 Health Diagnostic and Promotion Professionals	CLE_Oth_Logi	591 Logistics Clerks
PRO_Hea_Ther	252 Health Therapy Professionals	CLE_Oth_Misc	599 Miscellaneous Clerical and Administrative Workers
PRO_Hea_Medi	253 Medical Practitioners	SAL_Sal_Insu	611 Insurance Agents and Sales Representatives
PRO_Hea_Midw	254 Midwifery and Nursing Professionals	SAL_Sal_Real	612 Real Estate Sales Agents
PRO_ICT_Busi	261 Business and Systems Analysts, and	SAL_Sal_Sale	621 Sales Assistants and Salespersons
PRO_ICT_Data	262 Database and Systems Administrators, and ICT Security Specialists		631 Checkout Operators and Office Cashiers
PRO_ICT_ICT	263 ICT Network and Support Professionals	SAL_Sal_Chec	639 Miscellaneous Sales Support Workers
PRO_Leg_Lega	271 Legal Professionals	SAL_Sal_Misc	711 Machine Operators
PRO_Leg_Soci	272 Social and Welfare Professionals	MAC_Mac_Mach	712 Stationary Plant Operators
TEC_Eng_Agri	311 Agricultural, Medical and Science Technicians	MAC_Mac_Stat	721 Mobile Plant Operators
TEC_Eng_Buil	312 Building and Engineering Technicians	MAC_Mob_Mobi	721 Mobile Plant Operators
TEC_Eng_ICT	313 ICT and Telecommunications Technicians	MAC_Roa_Auto	731 Automobile, Bus and Rail Drivers
TEC_Aut_Auto	321 Automotive Electricians and Mechanics	MAC_Roa_Deli	732 Delivery Drivers
TEC_Aut_Fabr	322 Fabrication Engineering Trades Workers	MAC_Roa_Truc	733 Truck Drivers
TEC_Aut_Mech	323 Mechanical Engineering Trades Workers	MAC_Sto_Stor	741 Storepersons
TEC_Aut_Pane	324 Panelbeaters, and Vehicle Body Builders,	LAB_CIE_Clea	811 Cleaners and Laundry Workers
TEC_Con_Bric	331 Bricklayers, and Carpenters and Joiners	LAB_Con_Cons	821 Construction and Mining Labourers
TEC_Con_Flo	332 Floor Finishers and Painting Trades Workers	LAB_Fac_Food	831 Food Process Workers
TEC_Con_Glaz	333 Glaziers, Plasterers and Tilers	LAB_Fac_Pack	832 Packers and Product Assemblers
TEC_Con_Plum	334 Plumbers	LAB_Fac_Misc	839 Miscellaneous Factory Process Workers
TEC_EIE_Elec	341 Electricians	LAB_Far_Farm	841 Farm, Forestry and Garden Workers
TEC_EIE_Tele	342 Electronics and Telecommunications Trades	LAB_Foo_Food	851 Food Preparation Assistants
TEC_Foo_Food	351 Food Trades Workers	LAB_Oth_Frei	891 Freight Handlers and Shelf Fillers
TEC_Ski_Anim	361 Animal Attendants and Trainers, and Shearers	LAB_Oth_Misc	899 Miscellaneous Labourers