The Sym Specification for the G-Cubed Model

This document provides an example of the way Sym has been used in the past. It contains an annotated copy of the full Sym code for the standard nine-region, twelve-sector version of the G-Cubed model.[1] Lines of code and annotations can be distinguished by font and indenting:

```
// Example of Sym (a comment)
```

- Example annotation: the line above is a comment

For clarity, the appendix is divided into four sections: sets, parameters, variables, and equations. However, Sym itself is order-independent and the definitions of parameters, variables and equations can be freely mixed in the file. Also, Sym allows white space (including carriage returns) to be used freely throughout its input but requires that each statement be terminated by a semicolon.

1.1 Sets:

```
set regions (UU,JJ,AA,EE,OO,CC,LL,BB,PP) 'countries and regions' ;
```

- Declares set "regions" having 8 elements, one for each region in the model. (For historical reasons, the Sym code for G-Cubed uses doubled letters for regions.) Set declarations can be followed by comments in quotes.

```
set oecd (UU,JJ,AA,EE,OO) 'OECD countries';
set AnnexB (UU,JJ,AA,EE,OO,BB) 'Annex B countries';
set notAnnexB (CC,LL,PP) 'Non Annex B countries';
```

- All of the sets above will be recognized as subsets of "regions". They are used to set up equations that only apply to selected regions.

[1] McKibbin, Warwick J. and Peter J. Wilcoxen, "A Global Approach to Energy and Environment: The G-Cubed Model," in Peter B. Dixon and Dale W. Jorgenson, (eds), Handbook of Computational General Equilibrium Modeling, Amsterdam: North-Holland, pp. 995-1068, 2013.

```
set notUS = regions - (UU) 'used with REXC equation';
```

- Set “notUS” is a subset of “regions” that includes all elements except “UU”.

```
set sectors (a01,a02,a03,a04,a05,a06,a07,a08,a09,a10,a11,a12) 'ordinary sectors' ;
```

- Defines the sectors in the model. The prefix “a” is used to denote “agent”, indicating that the subsequent number refers to a sector rather than a good.

```
set goods (g01,g02,g03,g04,g05,g06,g07,g08,g09,g10,g11,g12) 'ordinary products' ;
```

- Defines the goods in the model. The prefix “g” is used to distinguish goods from sectors.

```
set goods_e = goods( g01,g02,g03,g04,g05) 'energy goods' ;
set goods_m = goods( g06,g07,g08,g09,g10,g11,g12) 'materials' ;
```

- These define key subsets of the list of goods.

```
set time (t0,t1,t2,t3,t4,t5,t6,t7,t8,t9) 'tablo time periods' ;
```

- Define a list of time periods. The Ox solution algorithm for G-Cubed does not use this set but Sym can also produce GEMPACK output files, which do use the set. Unlike all other sets in Sym, the time set is ordered: lead and lag operators can be used in equations to pick up variables from adjacent periods.

```
set last = time(t9) 'steady state' ;
```

- Defines a subset of the list of periods that corresponds to the steady state.

```
set dest = regions 'alias for regions' ;
set orig = regions 'alias for regions' ;
```

- Sym allows additional names, or “aliases”, to be defined for sets. Aliases are used to distinguish between semantically different uses of a given set. For example, the set of regions is used in two logically distinct ways when referring to imports: the region of origin and the region of destination. The “dest” and “orig” aliases allow that indexing to be made explicit and unambiguous. See variable IMP below for an example application.

- Because they are used for semantic purposes, aliases are not simply synonyms for the original set. Rather, they indicate a narrow interpretation of the set. A variable defined over the parent of an alias can be explicitly mapped to the alias but not vice versa. For example, “dest” is a particular use of “regions”. All variables defined over “regions” could potentially be used over “dest”. However, variables defined over “dest” can’t be used over “orig” (a parallel but distinct alias) or “region” (the parent set of “dest”).

```
set currency = regions 'currency of assets' ;
set owner = regions 'owner of assets ASSE' ;
```

- Two more aliases used in accounting for bilateral holdings of international assets. In this case, each region can potentially hold assets in the currencies of every other region. See variable ASSE .

```
set abroad = regions 'used in asset calculations' ;
```

- Used in each country's asset calculations to distinguish between the home country and the rest of the world.

```
set agents = sectors 'needed by msgproc' ;
set products = goods 'needed by msgproc' ;
```

- These sets are used in linking to the Ox solution algorithm.

```
set factors (K,L,E,M) 'top tier factors of production' ;
set varfac = factors - (K) 'variable factors' ;
set domfor (D,F) 'domestic vs foreign' ;
```

1.2 Parameters:

```
parameter adapt (regions) 'rate of closure TOB-TPA gap' ;
```

- Parameters are defined over sets. The declaration can optionally include a comment in quotes. This line defines the rate at which adaptive expectations by firms converge to full foresight.

```
parameter aeye(currency,owner) 'identity matrix' ;
```

- Sym's strict conformability rules occasionally make it necessary to use an identity matrix to map one set to another. In this case, AEYE is used to link each country ("owner") to its currency. In essence, it indicates that the dollar is the currency of the US, the yen is the currency of Japan, and so on.

```
parameter ainv(owner,abroad) 'inverse of I-ashr' ;
parameter ashr(currency,owner) 'asset split among currencies' ;
parameter base_jk(sectors,regions) 'base J/K' ;
parameter btucoef(goods,regions) 'emissions coefficients, energy' ;
parameter carcoef(goods,regions) 'emissions coefficients, carbon' ;

parameter cd_df (goods,regions) '1 if cd, 0 if ces' ;
parameter cd_e(sectors,regions) '1 if cd, 0 if ces' ;
parameter cd_eH(regions) '1 if cd, 0 if ces' ;
parameter cd_eR(regions) '1 if cd, 0 if ces' ;
parameter cd_ff (goods,dest ) '1 if cd, 0 if ces' ;
parameter cd_m(sectors,regions) '1 if cd, 0 if ces' ;
parameter cd_mH(regions) '1 if cd, 0 if ces' ;
```

```
parameter cd_mR(regions) '1 if cd, 0 if ces' ;
parameter cd_o(sectors,regions) '1 if cd, 0 if ces' ;
parameter cd_oH(regions) '1 if cd, 0 if ces' ;
parameter cd_oR(regions) '1 if cd, 0 if ces' ;
```

- In some circumstances, one or more of G-Cubed's production or consumption nodes are set to Cobb-Douglas. Because the functional form for Cobb-Douglas demand equations differs from that for the CES, these parameters enable Sym to select the correct form when the model is built.

```
parameter delta (regions) 'depreciation rate' ;
parameter delta_dom(goods,regions) 'domestic weight in DF' ;
parameter delta_e(regions,sectors,goods_e) 'input weights, individ goods' ;
parameter delta_eG(goods_e,regions) 'gov budget shares on E goods' ;
parameter delta_eH(goods_e,regions) 'input weights, individ goods' ;
parameter delta_eR(goods_e,regions) 'input weights, individ goods' ;
parameter delta_ff(goods,dest,orig) 'foreign source weights in FF' ;
parameter delta_m(regions,sectors,goods_m) 'input weights, individ goods' ;
parameter delta_mG(goods_m,regions) 'gov budget shares on M goods' ;
parameter delta_mH(goods_m,regions) 'input weights, individ goods' ;
parameter delta_mR(goods_m,regions) 'input weights, individ goods' ;
parameter delta_o(factors,sectors,regions) 'input weights, KLEM tier' ;
parameter delta_oG(factors,regions) 'gov KLEM budget shares' ;
parameter delta_oH(factors,regions) 'input weights, KLEM tier' ;
parameter delta_oR(factors,regions) 'input weights, KLEM tier' ;
parameter eer_weight(dest,orig) 'trade weights, NEER and REER' ;
parameter eurowt(regions) 'weight on countries in Eurozone';
parameter fore_c(regions) 'shr of C driven by foresight' ;
parameter fore_i(regions) 'shr of I driven by foresight' ;

parameter FTA (dest,orig) 'free trade area tariff flag' ;
```

- FTA is a key parameter for the FTAA and NAFTA simulations. When the value for a given origin and destination is set to 1, trade between those regions is subject to a reduced or zero tariff.

```
parameter int_elast(regions) 'interest elast of money demand' ;
parameter labgrow(regions) 'growth of effective labor' ;

parameter makeinv(goods,sectors,regions) 'inverse of make table' ;
```

- G-Cubed's structure allows for joint production via a make table linking producing sectors and goods. In the main version of the model, however, the make table is an identity matrix.

```
parameter mongdp(regions) 'coefficient on MONE in WELT' ;
parameter mpc(regions) 'marginal propensity to consume' ;
parameter mrule_ex(regions) 'weight on exchange rate in M rule' ;
parameter mrule_in(regions) 'weight on inflation in M rule' ;
parameter mrule_m(regions) 'weight on money in M rule' ;
parameter mrule_ny(regions) 'weight on nominal income in M rule';
parameter mrule_r(regions) 'weight on lag interest rate in M rule';
parameter mrule_y(regions) 'weight on output in M rule' ;

parameter MUL (dest,orig) 'multilateral tariff flag' ;
```

- Used to indicate that two trade partners are not part of a free trade area.

```
parameter phi (regions) 'adjustment cost parameter' ;
parameter prid_weight(goods,regions) 'weights in PRID' ;
parameter prim_weight(goods,regions) 'weights in PRIM' ;
parameter prix_weight(goods,regions) 'weights in PRIX' ;
parameter sigma_df(goods,regions) 'subs elast between D&F srcs' ;
parameter sigma_e(sectors,regions) 'subs elast, E tier' ;
parameter sigma_eH(regions) 'subs elast, E tier' ;
parameter sigma_eR(regions) 'subs elast, E tier' ;
parameter sigma_ff(goods,dest ) 'subs elast between F sources' ;
parameter sigma_m(sectors,regions) 'subs elast, M tier' ;
parameter sigma_mH(regions) 'subs elast, M tier' ;
parameter sigma_mR(regions) 'subs elast, M tier' ;
parameter sigma_o(sectors,regions) 'subs elast, KLEM tier' ;
parameter sigma_oH(regions) 'subs elast, KLEM tier' ;
parameter sigma_oR(regions) 'subs elast, KLEM tier' ;
parameter timepref(regions) 'time preference rate' ;
parameter transgdp(regions) 'transfers per unit of GDP' ;
parameter wage_p(regions) 'exp infl weight in wage eqn' ;
parameter wage_q(regions) 'employment param in wage eqn' ;
```

1.3 Variables:

```
variable ABUY(owner) 'new purchases of foreign assets' end,gdp ;
```

- Variables are also defined over sets and comments are allowed in declarations. In addition, variable declarations can optionally include a list of attributes; in this case, the attributes are “end” and “gdp”. The attributes are provided for the convenience of the solution algorithm and have no meaning to Sym. For G-Cubed’s Ox solution algorithm, “end” indicates an ordinary endogenous variable and “gdp” indicates that the units of the variable are normalized by the corresponding country’s GDP. The unit flag is used by programs that prepare input for, or process output from, the solution algorithm.

```
variable ASSE(currency,owner) 'assets in each currency, including own' sta,usgdp ;
```

- ASSE is the bilateral asset matrix. The attribute “sta” flags it as a state variable and “usgdp” indicates that it is normalized by US GDP.

```
variable ASSU(regions) 'assets held by region, in US$' end,gdp ;
variable BCOL(regions) 'Quads from coal' exo,btu;
```

- Attribute “exo” indicates that BCOL is exogenous (used by the Ox solution but not by the GEMPACK version). The units are BTUs.

```
variable BGAS(regions) 'Quads from gas' exo,btu;
variable BOIL(regions) 'Quads from oil' exo,btu;
variable BOND(regions) 'bonds' sta,gdp ;
variable CAP(sectors,regions) 'capital stock' sta,gdp ;
variable CAPY(regions) 'capital stock' sta,gdp ;
variable CAPZ(regions) 'capital stock' sta,gdp ;
variable CNPE(regions) 'HH demand for E' end,gdp ;
variable CNPK(regions) 'HH cap services' end,gdp ;
variable CNPL(regions) 'HH demand for L' end,gdp ;
variable CNPO(regions) 'HH demand for M' end,gdp ;
variable CON(goods,regions) 'HH demand for goods' end,gdp ;
variable CONP(regions) 'national accounts C' end,gdp ;
variable CURN(regions) 'current account, dom currency, using INAS' end,gdp ;
variable CURR(regions) 'current account, US$, using IRAS' end,gdp ;
variable DEFI(regions) 'fiscal deficit, using INTR' end,gdp ;
variable DEFN(regions) 'fiscal deficit, using INTN' end,gdp ;
variable DEFX(regions) 'fiscal deficit, using INTR (exog)' exo,gdp ;
variable ECOL(regions) 'CO2 emissions from coal' exo,mmt;
```

- Attribute “mmt” indicates that the units of emissions are million metric tonnes.

```
variable EGAS(regions) 'CO2 emissions from gas' exo,mmt;
variable EMCO(regions) 'CO2 emissions' end,mmtgdp ;
```

- Internally, emissions are in MMT normalized by GDP. They are converted back to MMT after the model is solved.

```
variable EMCOW 'CO2 emissions, global' end,mmtusgdp ;
variable EMCOZ 'CO2 emissions, Annex B' end,mmtusgdp ;
variable EMIS(regions) 'carbon emissions' end,mmtgdp ;
variable EMISW 'carbon emissions, global' end,mmtusgdp ;
variable EMISZ 'carbon emissions, Annex B' end,mmtusgdp ;
variable EMISZX 'carbon emissions, Annex B' end,mmtusgdp ;
variable EMIT(regions) 'carbon emissions' end,mmtgdp ;
```

```
variable EMIX(regions) 'carbon emissions, exogenous' exo,mmtgdp ;
variable EMZT(regions) 'carbon emissions' end,mmtusgdp ;
variable EN(sectors,goods_e,regions) 'sector demand for E goods' end,gdp ;
variable ENER(regions) 'energy consumption' end,btugdp ;
variable ENERW 'energy consumption, global' end,btuusgdp ;
variable ENERZ 'energy consumption, Annex B' end,btuusgdp ;
variable ENT(sectors,regions) 'sector demand for E' end,gdp ;
variable EOIL(regions) 'CO2 emissions from oil' exo,mmt;
variable EXCH(regions) 'exch rate as US$ per unit of region currency' stl,pct ;
```

- Attribute "stl" indicates a state variable that may appear as the target of an earlier period's lead() operator. The unit "pct" indicates that the internal form of the variable in the model is the natural log of its ordinary value. All prices in G-Cubed are expressed in log form and have their units set to "pct". The code "pct" is used because it is often convenient to interpret the changes in such variables as percentages.

```
variable EXCL(regions) 'exch rate as US$ per unit of region currency - lagged'
    sta,pct ;
variable EXCR(regions) 'exchange rate shock' exo,pct ;
variable EXCX(regions) 'exch rate target ' exo,pct ;
variable EXNA(regions) 'value of exports, deflated by PRIX' end,gdp ;
variable EXQ(goods,regions) 'exports, EXP.xx' end,gdp ;
variable EXQT(regions) 'value of exports at source' end,gdp ;
variable GCE(goods,regions) 'government purchases' end,gdp ;
variable GCET(regions) 'national accounts G except gov labor' end,gdp ;
variable GDPN(regions) 'GDP, nominal' end,gdp ;
variable GDPR(regions) 'GDP, individual component deflation' end,gdp ;
variable GNPR(regions) 'GNP, GDPR plus PBAL and interest' end,gdp ;
variable GOVL(regions) 'labor used by gov' exo,gdp ;
variable GOVS(regions) 'total gov spending' end,gdp ;
variable IIN(goods,regions) 'R demand for goods' end,gdp ;
variable IITE(regions) 'R demand for E' end,gdp ;
variable IITL(regions) 'R demand for L' end,gdp ;
variable IITO(regions) 'R demand for M' end,gdp ;
variable IMNA(regions) 'value of imports at dest, deflated by PRIM' end,gdp ;
variable IMP(goods,dest,orig) 'individual imports, IMP.xy' end,gdp ;
```

- IMP is a set of bilateral trade matrices. It contains the amount of each import to every region from every other region. The set aliases "dest" and "orig" are used to eliminate the ambiguity about indexing that would arise if IMP were defined over regions twice.

```
variable IMQ(goods,regions) 'DF imports by good, IMQ.xx' end,gdp ;
variable IMQT(regions) 'value of imports at dest' end,gdp ;
variable INAS(regions) 'net int pmts on for assets using INTN (USD)' end,gdp ;
variable INCM(regions) 'after tax household income' end,gdp ;
```

```
variable INFL(regions) 'inflation rate in PRCT' end,del ;
variable INFP(regions) 'inflation rate in PRID' end,del ;
variable INFX(regions) 'inflation target' exo,del ;
```

- The units of inflation rates, interest rates and related variables are "del" which indicates to the pre- and post-processing programs that the variable is a pure number and has not been normalized.

```
variable INS (regions,goods) 'agg I demand shocks' exo,gdp ;
variable INTF(regions) 'risk free interest rate, short-run, real' end,del ;
variable INTL(regions) 'interest rate, short-run, nominal (lagged)' sta,del ;
variable INTN(regions) 'interest rate, short-run, nominal' stl,del ;
variable INTR(regions) 'risk adjusted interest rate, short-run, real' end,del ;
variable INTX(regions) 'interest rate target' exo,del ;
variable INV(sectors,regions) 'raw investment' end,gdp ;
variable INVT(regions) 'total cap goods demanded' end,gdp ;
variable INVY(regions) 'raw investment' end,gdp ;
variable INVZ(regions) 'raw investment' end,gdp ;
variable IRAS(regions) 'net int pmts on for assets using INTR (USD)' end,gdp ;
variable JNV(sectors,regions) 'installed investment' end,gdp ;
variable JNVY(regions) 'installed investment' end,gdp ;
variable JNVZ(regions) 'installed investment' end,gdp ;
variable LAB(sectors,regions) 'sector demand for L' end,gdp ;
variable LABO(regions) 'total labor demanded' end,gdp ;
variable LABX(regions) 'total labor supply' exo,gdp ;

variable LAM(sectors,regions) 'costate var lambda' cos,del ;
variable LAMY(regions) 'costate var lambda' cos,del ;
variable LAMZ(regions) 'costate var lambda' cos,del ;
```

- Attribute "cos" indicates a costate variable.

```
variable LGDPN(regions) 'GDP, nominal' exo,del;
variable LGDPP(regions) 'GDP, PPP' exo,del;
variable LGDPR(regions) 'GDP, real' exo,del;
variable LGYL(regions) 'ln of output (lagged)' sta,pct ;
variable LGYX(regions) 'ln of output target' exo,pct ;
variable LOGY(regions) 'ln of output' stl,pct ;
variable MONE(regions) 'money demanded IN LOGS' end,pct ;
variable MONL(regions) 'money demanded as %GDP' end,gdp ;
variable MONX(regions) 'money target IN LOGS' exo,pct ;
variable NB02(regions) 'interest rate, 2-year, nominal' end,del ;
variable NB05(regions) 'interest rate, 5-year, nominal' end,del ;
```

```
variable NB10(regions) 'interest rate, 10-year, nominal' end,del ;
variable NEER(regions) 'trade-weighted exch rate, FC/domestic' end,pct ;
variable OI(sectors,goods_m,regions) 'sector demand for M goods' end,gdp ;
variable OIN(sectors,regions) 'sector demand for M' end,gdp ;
variable OUG(goods,regions) 'domestic supply of goods' end,gdp ;
variable OUP(sectors,regions) 'domestic production' end,gdp ;
variable OUTP(regions) 'agg output Q index to match PRID' end,gdp ;
variable OUY(goods,regions) 'composite supply' end,gdp ;
variable PBAL(regions) 'value of permits traded, USD' exo,usgdp ;
variable PGDP(regions) 'GDP, implicit price deflator GDPN/GDPR' end,pct ;
variable PIM(goods,dest,orig) 'landed price of IMP' end,pct ;
variable PMQ(goods,regions) 'landed price index for IMQ' end,pct ;
variable PMR(goods,regions) 'price of imports incl tariffs' end,pct ;
variable POI(sectors,regions) 'sector price of M' end,pct ;
variable PRCE(regions) 'HH price of E' end,pct ;
variable PRCL(regions) 'aggregate price of consumer goods lag' sta,pct ;
variable PRCO(regions) 'HH price of M' end,pct ;
variable PRCT(regions) 'aggregate price of consumer goods' ets,pct ;
```

- Attribute “ets” indicates that the variable is used in both lag and lead contexts.

```
variable PRD(goods,regions) 'pretax price of goods' end,pct ;
variable PRDL(regions) 'weighted price of domestic output expected' sta,pct ;
variable PRDX(regions) 'weighted price of domestic output' exo,pct ;
variable PRE(sectors,regions) 'sector price of E' end,pct ;
variable PRF(sectors,regions) 'short run profits' end,gdp ;
variable PRFY(regions) 'profits on R capital' end,gdp ;
variable PRFZ(regions) 'imputed short run profits on HH cap' end,gdp ;
variable PRGT(regions) 'aggregate price of government goods' end,pct ;
variable PRID(regions) 'weighted price of domestic output' ets,pct ;
variable PRIE(regions) 'R price of E' end,pct ;
variable PRII(regions) 'prod price of inv goods' end,pct ;
variable PRIM(regions) 'weighted price of imports' end,pct ;
variable PRIX(regions) 'weighted price of exports' end,pct ;
variable PRK(sectors,regions) 'implicit rental price' end,pct ;
variable PRKY(regions) 'implicit rental price' end,pct ;
variable PRKZ(regions) 'implicit rental price' end,pct ;
variable PROI(regions) 'R price of M' end,pct ;
variable PRP(sectors,regions) 'producer price' end,pct ;
variable PRS(goods,regions) 'producer price of composite' end,pct ;
variable PRX(goods,regions) 'price of exports at source' end,pct ;
variable PRY(goods,regions) 'purchase price of composite' end,pct ;
variable RB10(regions) 'interest rate, 10-year, real' end,del ;
```

```
variable REER(regions) 'trade-weighted real exch rate, FC/domestic' end,pct ;
variable REXC(regions) 'real exch rate as US$ per unit of regions currency' cos,pct ;
variable RISE(sectors,regions) 'equity risk premium by sector by region' exo,del ;
variable RISH(regions) 'risk premium shock for human wealth' exo,del ;
variable RISR(regions) 'risk premium in the yield curve' exo,del ;
variable RISW(regions) 'risk premium for human wealth' exo,del ;
variable RISY(regions) 'risk premium for R capital' exo,del ;
variable RISZ(regions) 'risk premium for HH capital' exo,del ;
variable ROGY(regions) 'potential output growth' exo,pct ;
variable SAVI(regions) 'domestic saving, real' end,gdp ;
variable SAVN(regions) 'domestic saving, nominal' end,gdp ;
variable SAVT(regions) 'domestic non-governmen saving, nominal' end,gdp ;
variable SHEF(regions,sectors) 'aeei shock' exo,pct ;
variable SHEFC(regions) 'aeei shock - consumption' exo,pct ;
variable SHKC(regions) 'agg con shock' exo,gdp ;
variable SHKI(regions,sectors) 'shock in inv equation' exo,gdp ;
variable SHKM(regions) 'shock to money demand' exo,pct ;
variable SHL (regions,sectors) 'labor shock' exo,pct ;
variable SHLY(regions) 'labor shock, R' exo,pct ;
variable SHLZ(regions) 'labor shock, HH' exo,pct ;
variable SHY (regions,sectors) 'tfp shock' exo,pct ;
variable SHYY(regions) 'tfp shock, R' exo,pct ;
variable SHYZ(regions) 'tfp shock, HH' exo,pct ;
variable STM(sectors,regions) 'stock market value' end,gdp ;
variable STMT(regions) 'Total stock market value' end,gdp ;
variable TAX(goods,regions) 'sales tax on composite' exo,pct ;
variable TAXC(regions) 'revenue from sales & corp taxes' end,gdp ;
variable TAXE(regions) 'revenue from externality taxes' end,gdp ;
variable TAXH(regions) 'revenue from taxes on wages' end,gdp ;
variable TAXL(regions) 'revenue for bond interest plus TAXS' end,gdp ;
variable TAXM(regions) 'tariff revenue' end,gdp ;
variable TAXS(regions) 'exogenous lump sum tax' exo,gdp ;
variable TAXT(regions) 'total revenue' end,gdp ;
variable TAXX(regions) 'extern tax revenue' end,gdp ;
variable TBAL(regions) 'trade surplus, domestic currency' end,gdp ;
variable TBAU(regions) 'trade surplus, USD' end,gdp ;
variable TBFD(regions) 'unit tax on energy' exo,cent ;
variable TBFI(regions) 'unit tax on energy, imports' exo,cent ;
variable TBFX(regions) 'unit tax on energy, exports' exo,cent ;
variable TBNA(regions) 'trade surplus, deflated by PRIX and PRIM' end,gdp ;
variable TCAI(regions) 'unit tax on carbon, imports' end,cent ;
variable TCAR(regions) 'unit tax on carbon' end,cent ;
variable TCAW 'unit tax on carbon' end,cent ;
```

```
variable TCAX(regions) 'unit tax on carbon' exo,cent ;
variable TCEX(regions) 'unit tax on carbon, exports' end,cent ;
variable TCOR(regions) 'tax rate on corporate profits' exo,pct ;
variable TEX(goods,regions) 'extern tax on composite' exo,pct ;
variable TIF(goods,dest ) 'fta tariff rates' exo,pct ;
variable TIM(goods,dest ) 'multilateral tariff rates' exo,pct ;
variable TINC(regions) 'tax rate on labor income' exo,pct ;
variable TITC(sectors,regions) 'rate of ITC' exo,pct ;
variable TITCH(regions) 'rate of ITC for households' exo,pct ;
variable TOB(sectors,regions) 'marginal q, TOB.NC' end,del ;
variable TOBY(regions) 'marginal q, TOB.NC' end,del ;
variable TOBZ(regions) 'marginal q, TOB.NC' end,del ;
variable TPA(sectors,regions) 'partial adjustment q' sta,del ;
variable TPAY(regions) 'partial adjustment q' sta,del ;
variable TPAZ(regions) 'partial adjustment q' sta,del ;
variable TRAN(regions) 'total transfers' end,gdp ;
variable TRANX(regions) 'exogenous transfers' exo,gdp ;
variable TXM(goods,dest) 'tariff revenue by good' end,gdp ;
variable WAG(sectors,regions) 'sector wage' end,pct ;
variable WAGE(regions) 'wage rate' sta,pct ;
variable WELH(regions) 'human wealth' cos,gdp ;
variable WELT(regions) 'total wealth' end,gdp ;
variable YRATN(regions) 'GDP ratio, nominal' exo,pct;
variable YRATP(regions) 'GDP ratio, PPP' exo,pct;
variable YRATR(regions) 'GDP ratio, real' exo,pct;
```

1.4 Equations:

```
EXCH(UU) = 0;
```

- This equation is unusual in several respects and is not ideal for illustrating how Sym works. However, for historical reasons it happens to be first in the file. The “UU” in this equation indicates that it holds for a subset of possible values of EXCH; in this case, for the US only. Exchange rates in G-Cubed are all expressed relative to the US dollar. This equation essentially sets the model’s numeraire by setting the nominal US exchange rate to 1. However, EXCH is actually the log of the exchange rate, so it is set to 0.

```
goods_e: OUY = CON + IIN + INS + GCE + EXQ + sum(sectors,EN);
goods_m: OUY = CON + IIN + INS + GCE + EXQ + sum(sectors,OI);
```

- This pair of equations represents all of the market clearing conditions for goods in the model (that is, all goods in all regions). Sym allows equations to be restricted to subsets of the possible space over which they hold by prefixing them with one or more sets. In this case, the first equation holds for all energy goods and the second holds for all

materials. The distinction is only needed because the intermediate demands for energy and materials use different variable names (EN and OI).

```
IMQ = (1-delta_dom) * OUY * exp( PRS - PMR )^sigma_df ;
```

- A standard equation with no special adjustments. It holds for all goods in all regions. The exponentiation is needed because PRS and PMR are expressed in logs.

```
PIM = EXCH(orig) + PRX(orig)#dest - EXCH(dest);
```

- In some circumstances, a variable may need to be replicated over an additional dimension. In this case, PRX is the price of exports and varies over goods and region of origin. However, this equation calculates the price of imports, which varies over goods, region of origin and region of destination. The term "#dest" appended to PRX indicates that all elements of set "dest" use the same matrix PRX.

```
IMP = delta_ff * IMQ(dest)*(carcoef(dest)*TCAI(dest)+btucoef(dest)*TBFI(dest)+ (1 +
    TIM*MUL#goods + TIF*FTA#goods)*exp(PIM-PMR(dest)))^(-sigma_ff(dest));
```

- Here, matrices MUL and FTA are replicated over goods: if two trade partners are members of a free trade area, the set of FTA tariffs TIF applies to all goods they trade.

```
PMQ = ln( sum(orig, IMP*exp(PIM) )) - ln(IMQ+.00007) ;
```

- This equation holds for all goods and all regions. The sum adds up the value of imports of the good from all of the country's trade partners. The small number in this equation is present to avoid numerical errors when IMQ happens to be zero in the input data.

```
IMQT = sum(goods, exp(PMQ)*IMQ )/exp(PRID);
EXQT = sum(goods, exp(PRX)*EXQ )/exp(PRID) ;
```

- The total values of imports to, and exports from, each country.

```
EXQ(orig) = sum(dest,IMP);
```

- Exports from a given country of origin are equal to the sum of imports from that country to all possible destinations. This equation illustrates an important feature of set aliases: a variable defined over the parent of the alias can be coerced to hold over the alias. In this case, EXQ is defined over "regions" which is the parent of "orig". The expression "EXQ(orig)" tells Sym that in this context it should treat EXQ as though it had been defined over "orig" for conformability purposes.

```
PRX = ln( (1+TEX)*exp(PRS) + carcoef*TCEX + btucoef*TBFX);
OUG = delta_dom * OUY * exp( PRS - PRD )^sigma_df ;
OUP = sum( goods, makeinv*OUG );
LAB = exp(SHL+SHY )^(sigma_o-1 ) * delta_o(l) * OUP * exp(PRP-WAG)^sigma_o ;
ENT = exp(SHEF+SHY )^(sigma_o-1 ) * delta_o(e) * OUP * exp(PRP-PRE)^sigma_o ;
OIN = exp(SHY )^(sigma_o-1 ) * delta_o(m) * OUP * exp(PRP-POI)^sigma_o ;
```

```
IITL = exp(SHLY+SHYY)^(sigma_oR-1) * delta_oR(l) * INVT * exp(PRII-WAGE)^sigma_oR ;
IITE = exp(SHYY )^(sigma_oR-1) * delta_oR(e) * INVT * exp(PRII-PRIE)^sigma_oR ;
IITO = exp(SHYY )^(sigma_oR-1) * delta_oR(m) * INVT * exp(PRII-PROI)^sigma_oR ;
goods_e: IIN = delta_eR * IITE * exp( PRIE - PRY )^sigma_eR;
goods_m: IIN = delta_mR * IITO * exp( PROI - PRY )^sigma_mR;
LABO = IITL + CNPL + GOVL + sum(sectors,LAB);

PRS = cd_df*( delta_dom *PRD + (1-delta_dom)*PMR ) + (1-cd_df)*ln( delta_dom *
    exp(PRD)^(1-sigma_df) + (1-delta_dom) * exp(PMR)^(1-sigma_df) ) / (1-sigma_df*(1-
    cd_df)) ;
```

- Illustrates the use of the "cd" parameters. The value of "cd_df" (at the level of individual goods in particular regions) will be set to 1 if the Cobb-Douglas specification is desired and set to 0 for the default CES equation.

```
PRY = ln(1+TAX) + PRS ;
PMR(dest) = cd_ff*sum(orig, delta_ff*ln( carcoef(dest)*TCAI(dest) +
    btucoef(dest)*TBFI(dest) +(1+TIM*MUL#goods+TIF*FTA#goods)*exp(PIM))) + (1-
    cd_ff)*ln( sum(orig, delta_ff*(carcoef(dest)*TCAI(dest) +
    btucoef(dest)*TBFI(dest) +(1+TIM*MUL#goods+TIF*FTA#goods)*exp(PIM))^(1-sigma_ff))
    ) / (1-sigma_ff*(1-cd_ff)) ;
PRIM = sum(goods, prim_weight*PMR );
PRIE = cd_eR*sum(goods_e, delta_eR*PRY(goods_e)) + (1-cd_eR)*ln( sum(goods_e,
    delta_eR*exp(PRY(goods_e))^(1-sigma_eR)) ) / (1-sigma_eR*(1-cd_eR)) ;
PROI = cd_mR*sum(goods_m, delta_mR*PRY(goods_m) ) + (1-cd_mR)*ln( sum(goods_m,
    delta_mR*exp(PRY(goods_m))^(1-sigma_mR)) ) / (1-sigma_mR*(1-cd_mR)) ;
PRD = ln( exp(sum(sectors, makeinv*PRP)) + carcoef*TCAR + btucoef*TBFD );
PRP = cd_o*( delta_o(k)*PRK + delta_o(l)*WAG + delta_o(e)*PRE + delta_o(m)*POI -
    delta_o(l)*SHL - delta_o(e)*SHEF ) + (1-cd_o)*ln( delta_o(k)*exp(PRK )^(1-
    sigma_o) + delta_o(l)*exp(WAG-SHL )^(1-sigma_o) + delta_o(e)*exp(PRE-SHEF)^(1-
    sigma_o) + delta_o(m)*exp(POI )^(1-sigma_o) ) / (1-sigma_o*(1-cd_o)) - SHY ;
PRII = cd_oR*( delta_oR(k)*PRKY + delta_oR(l)*WAGE + delta_oR(e)*PRIE +
    delta_oR(m)*PROI - delta_oR(l)*SHLY ) + (1-cd_oR)*ln( delta_oR(k)*exp(PRKY )^(1-
    sigma_oR) + delta_oR(l)*exp(WAGE-SHLY)^(1-sigma_oR) + delta_oR(e)*exp(PRIE)^(1-
    sigma_oR) + delta_oR(m)*exp(PROI )^(1-sigma_oR) ) / (1-sigma_oR*(1-cd_oR)) - SHYY
    ;
PRK = PRP + ln( delta_o(k)*OUP/CAP )/sigma_o + SHY*(sigma_o-1)/sigma_o ;
PRKY = PRII + ln( delta_oR(k)*INVT/CAPY )/sigma_oR + SHYY*(sigma_oR-1)/sigma_oR ;
PRKZ = PRCT + ln( delta_oH(k)*CONP/CAPZ )/sigma_oH + SHYZ*(sigma_oH-1)/sigma_oH ;


lead(WAGE) = WAGE + wage_p*(lead(PRCT)-PRCT) + (1-wage_p)*(PRCT-PRCL) +
    wage_q*ln(LABO) ;
```

- Sym allows lead and lag operators for use in constructing intertemporal equations.

```
last: LABO = LABX {miss} ;
```

- Terminal conditions can be specified by defining equations that hold only in the last period of the simulation. Like variables, equations may have attributes that are not used by Sym but are passed to the solution algorithm. Attributes for equations are listed in curly brackets. The attribute in this equation, "miss", is experimental and not used by the current solution algorithm.

```
WAG = WAGE#sectors;
```

- In this version of G-Cubed, the wage is the same across all sectors within a region. WAGE is the aggregate regional wage rate; WAG is a matrix of sector-level wages in each region; and "#sectors" maps the aggregate wage to each of the sectors.

```
lead(CAP ) = JNV + (1-delta-labgrow)*CAP ;
lead(CAPY) = JNVY + (1-delta-labgrow)*CAPY ;
lead(CAPZ) = JNVZ + (1-delta-labgrow)*CAPZ ;
last: CAP = JNV / (delta+labgrow) {miss} ;
last: CAPY = JNVY / (delta+labgrow) {miss} ;
last: CAPZ = JNVZ / (delta+labgrow) {miss} ;
INV = ( (1+.5*(TOB-1))*(TOB-1)*CAP/phi )*fore_i + ( (1+.5*(TPA-1))*(TPA-1)*CAP/phi
    )*(1-fore_i) + SHKI ;
JNV = ( INV + phi*0.5*(base_jk^2)*CAP )/(1+phi*base_jk);
INVY = JNVY*( 1 + (phi/2)*JNVY/CAPY ) ;
INVZ = JNVZ*( 1 + (phi/2)*JNVZ/CAPZ ) ;
JNVY = CAPY*( fore_i*TOBY + (1-fore_i)*TPAY - 1 )/phi;
JNVZ = CAPZ*( fore_i*TOBZ + (1-fore_i)*TPAZ - 1 )/phi;
lead(LAM) = (1+INTR+RISE+delta)*LAM - (1-TCOR)*exp(PRK-PRID) - (1-TITC)*exp(PRII-
    PRID)*(phi/2)*(JNV/CAP)^2;
last: LAM = ( (1-TCOR)*exp(PRK-PRID) + (1-TITC)*exp(PRII-PRID)*(phi/2)*(JNV/CAP)^2 )
    / (INTR+RISE+delta) {miss} ;
lead(LAMY) = (1+INTR+RISY+delta)*LAMY - (1-TCOR)*exp(PRKY-PRID) - exp(PRII-
    PRID)*(phi/2)*(JNVY/CAPY)^2;
last: LAMY = ( (1-TCOR)*exp(PRKY-PRID) + exp(PRII-PRID)*(phi/2)*(JNVY/CAPY)^2 ) /
    (INTR+RISY+delta) {miss} ;
lead(LAMZ) = (1+INTR+RISZ+delta)*LAMZ - exp(PRKZ-PRID) - (1-TITCH)*exp(PRII-
    PRID)*(phi/2)*(JNVZ/CAPZ)^2;
last: LAMZ = ( exp(PRKZ-PRID) + (1-TITCH)*exp(PRII-PRID)*(phi/2)*(JNVZ/CAPZ)^2 ) /
    (INTR+RISZ+delta) {miss} ;
TOB = LAM *exp(PRID-PRII)/(1-TITC) ;
TOBY = LAMY*exp(PRID-PRII);
TOBZ = LAMZ*exp(PRID-PRII)/(1-TITCH);
lead(TPA) = TPA + adapt*( TOB - TPA );
lead(TPAY) = TPAY + adapt*( TOBY - TPAY );
lead(TPAZ) = TPAZ + adapt*( TOBZ - TPAZ );
last: TPA = TOB {miss} ;
last: TPAY = TOBY {miss} ;
```

```
last: TPAZ = TOBZ {miss} ;
PRF = ( OUP*exp(PRP) - ENT*exp(PRE) - LAB*exp(WAG) - OIN*exp(POI) - (1-
    TITC)*INV*exp(PRII) )/exp(PRID) ;
PRFY = ( INVT*exp(PRII) - IITE*exp(PRIE) - IITL*exp(WAGE) - IITO*exp(PROI) -
    INVY*exp(PRII) )/exp(PRID);
PRFZ = ( CONP*exp(PRCT) - CNPE*exp(PRCE) - CNPL*exp(WAGE) - CNPO*exp(PRCO) - (1-
    TITCH)*INVZ*exp(PRII) )/exp(PRID) ;
STM = LAM*CAP/(1-TITC) ;
STMT = sum(sectors,STM) ;
INVT = INVY + INVZ + sum(sectors,INV) ;
PRID = sum( goods, prid_weight*PRD ) ;
PRIX = sum( goods, prix_weight*PRX ) ;
OUTP = sum ( goods, exp(PRD)*OUG )/exp(PRID) ;
LOGY = ln(OUTP);
lead(LGYL)= LOGY;
INCM = TRAN - TAXL + INTR*BOND + IRAS/exp(REXC) + PRFZ + (1-TCOR)*( sum(sectors,PRF)
    + PRFY ) + (1-TINC)*( exp(WAGE)*(IITL+CNPL+GOVL) + sum(sectors,exp(WAG)*LAB)
    )/exp(PRID) ;
CONP = fore_c*(timepref+RISW)*WELT*exp(PRID-PRCT) + (1-fore_c)* mpc * INCM*exp(PRID-
    PRCT) + SHKC ;


IRAS = sum( currency, INTR(currency)*exp(REXC(currency))*ASSE ) ;
INAS = sum( currency, INTN(currency)*exp(REXC(currency))*ASSE ) ;
```

- Sym will automatically coerce variables defined over parent sets to aliases when it can do so unambiguously. In these equations, the left hand side variables are defined over "regions". After the summation, the right hand sides will be defined over "owner", an alias of "regions". Because there is no ambiguity about the meaning of the equation, for conformability tests, Sym will automatically treat IRAS and INAS as though they had been defined over "owner".

```
CNPL = exp(SHLZ+SHYZ)^(sigma_oH-1) * delta_oH(l) * CONP * exp(PRCT-WAGE)^sigma_oH ;
CNPE = exp(SHEFC+SHYZ)^(sigma_oH-1) * delta_oH(e) * CONP * exp(PRCT-PRCE)^sigma_oH ;
CNPO = exp(SHYZ )^(sigma_oH-1) * delta_oH(m) * CONP * exp(PRCT-PRCO)^sigma_oH ;
CNPK = exp(SHYZ )^(sigma_oH-1) * delta_oH(k) * CONP * exp(PRCT-PRKZ)^sigma_oH ;
goods_e: CON = delta_eH * CNPE * exp( PRCE - PRY )^sigma_eH;
goods_m: CON = delta_mH * CNPO * exp( PRCO - PRY )^sigma_mH;
PRCT = cd_oH*( delta_oH(k)*PRKZ + delta_oH(l)*WAGE + delta_oH(e)*PRCE +
    delta_oH(m)*PRCO - delta_oH(l)*SHLZ - delta_oH(e)*SHEFC ) + (1-cd_oH)*ln(
    delta_oH(k)*exp(PRKZ )^(1-sigma_oH) + delta_oH(l)*exp(WAGE-SHLZ)^(1-sigma_oH) +
    delta_oH(e)*exp(PRCE-SHEFC)^(1-sigma_oH) + delta_oH(m)*exp(PRCO )^(1-sigma_oH) )
    / (1-sigma_oH*(1-cd_oH)) - SHYZ ;
PRCE = cd_eH*sum( goods_e, delta_eH*PRY(goods_e) ) + (1-cd_eH)*ln( sum(goods_e,
    delta_eH*exp(PRY(goods_e))^(1-sigma_eH)) ) / (1-sigma_eH*(1-cd_eH)) ;
PRCO = cd_mH*sum( goods_m, delta_mH*PRY(goods_m) ) + (1-cd_mH)*ln( sum(goods_m,
    delta_mH*exp(PRY(goods_m))^(1-sigma_mH)) ) / (1-sigma_mH*(1-cd_mH)) ;
```

```
MONE = PRID + LOGY + int_elast*INTN + SHKM ;
INTN = mrule_r*INTL + mrule_ex*(EXCH -EXCL - EXCX) + mrule_in*(INFL - INFX) +
    mrule_m*(MONE - MONX) + mrule_ny*(LOGY + PRID - LGYX - PRDX- ROGY) +
    mrule_y*(LOGY - LGYL - ROGY) + INTX ;
lead(INTL) = INTN;
INTF = INTN - lead(PRID) + PRID ;
INTR = INTF + RISR ;
last: INTR = INTN {miss} ;
INFL = PRCT - PRCL ;
INFP = PRID - PRDL ;
lead (PRCL) = PRCT ;
lead (PRDL) = PRID ;
GDPN = exp(PRCT)*CONP + exp(PRGT)*GCET + exp(WAGE)*GOVL + exp(PRII)*INVT +
    exp(PRID)*TBNA + sum(goods,exp(PRII)*INS) ;
GDPR = (exp(PRCT)*CONP + exp(PRGT)*GCET + exp(WAGE)*GOVL + exp(PRII)*INVT +
    exp(PRID)*TBNA + sum(goods,exp(PRII)*INS))/exp(PRID) ;
PGDP = PRID ;
GNPR = GDPR + (PBAL+IRAS)/exp(REXC) ;
CURN = TBAL + (PBAL+INAS)/exp(REXC) ;
CURR = TBAU + PBAL + IRAS ;
WELT = BOND + ASSU/exp(REXC) + mongdp*exp(MONE)/exp(PRID) + WELH + LAMY*CAPY +
    LAMZ*CAPZ + sum(sectors,STM) ;
lead(WELH) = ( 1 + RISW + RISH + INTR - labgrow )*WELH - ( TRAN - TAXH - TAXL ) - (
    exp(WAGE)*(IITL+CNPL+GOVL) + sum(sectors,exp(WAG)*LAB) )/exp(PRID) ;
last: WELH = ( TRAN - TAXH - TAXL + ( exp(WAGE)*(IITL+CNPL+GOVL) +
    sum(sectors,exp(WAG)*LAB) )/exp(PRID) ) / ( RISW + RISH + INTR - labgrow ) {miss}
    ;
TXM = sum(orig, (TIM*MUL#goods + TIF*FTA#goods)*exp(PIM)*IMP )/exp(PRID) ;
TAXM(dest) = sum(goods,TXM) ;
TAXX = sum(goods, TEX*exp(PRS)*EXQ )/exp(PRID) ;
TAXT = TAXC+TAXH+TAXL+TAXM+TAXX+TAXE;
TCAR = TCAX/exp(REXC);
TCAI = TCAR;
TCEX = -TCAR;
TAXH = TINC*( exp(WAGE)*(IITL+CNPL+GOVL) + sum(sectors,exp(WAG)*LAB) )/exp(PRID) -
    (TITCH*exp(PRII)*INVZ)/exp(PRID);
TRAN = TRANX - transgdp*OUTP;
DEFN = GCET + GOVL*exp(WAGE-PRID) + TRAN - TAXT + INTN*BOND ;
DEFI = DEFX;
GCET = sum(goods, GCE*exp(PRY) )/exp(PRID) ;
goods_e: GCE = delta_eG*(delta_oG(e)/(delta_oG(e)+delta_oG(m))) *
    GOVS*exp(PRID)/exp(PRY);
goods_m: GCE = delta_mG*(delta_oG(m)/(delta_oG(e)+delta_oG(m))) *
    GOVS*exp(PRID)/exp(PRY);
```

```
PRGT = (delta_oG(e)/(delta_oG(e)+delta_oG(m))) * sum(goods_e,delta_eG*PRY(goods_e)) +
    (delta_oG(m)/(delta_oG(e)+delta_oG(m))) * sum(goods_m,delta_mG*PRY(goods_m));
lead(BOND) = DEFI + BOND*(1-labgrow) ;
last: BOND = DEFI/labgrow {miss} ;
SAVI = INVT*exp(PRII-PRID) + CURR/exp(REXC) + DEFI ;
SAVN = INVT*exp(PRII-PRID) + CURN + DEFN ;
SAVT = INVT*exp(PRII-PRID) + CURN ;
IMNA = sum(goods,exp(PMR)*IMQ)/exp(PRID) ;
EXNA = EXQT ;
TBNA = EXNA-IMNA ;
TBAL = EXQT-IMQT ;
TBAU = (EXQT-IMQT)*exp(REXC) ;
EMIS = 1000*sum( goods, carcoef*(OUY-EXQ) );
EMIT = EMIS - EMIX;
ENER = 1000*sum( goods, btucoef*(OUY-EXQ) ) ;
EMCO = EMIS*44/12 ;
EMISZX = sum( AnnexB, EMIX);
EMISZ = sum( AnnexB, EMIS);
EMISW = sum( regions, EMIS );
ENERZ = sum( AnnexB, ENER);
ENERW = sum( regions, ENER );
EMCOW = EMISW*44/12;
EMCOZ = EMISZ*44/12;


EXCH(notUS) = REXC(notUS) - PRID(notUS) + PRID(UU) ;
```

- This equation matches the first equation in the file and fills in the nominal exchange rate for all regions other than the US. REXC is the real exchange rate and PRID is the domestic price level.

```
lead(EXCL) = EXCH;
lead(REXC) = REXC - INTR + INTR(UU) + EXCR ;
last: INTR(notUS) = INTR(UU) + EXCR(notUS) {miss} ;
NEER(dest) = - sum(orig, eer_weight*(EXCH(orig)#dest-EXCH(dest)) );
REER(dest) = - sum(orig, eer_weight*(REXC(orig)#dest-REXC(dest)) );
ABUY = sum( abroad, ainv*CURR(abroad) );
lead(ASSE) = ASSE*( 1-labgrow(owner) ) + ( ashr*ABUY + aeye*(CURR(owner)-ABUY) ) /
    exp(REXC(currency)) ;
last: ASSE = ( ashr*ABUY/labgrow(owner) + aeye*(CURR(owner)-ABUY)/labgrow(owner) ) /
    exp(REXC(currency)) {miss} ;
ASSU = sum( currency, exp(REXC(currency))*ASSE ) ;
```

```
goods_e: EN = delta_e * ENT * exp( PRE - PRY#sectors )^sigma_e;
goods_m: OI = delta_m * OIN * exp( POI - PRY#sectors )^sigma_m ;
goods_e: PRE = cd_e*sum(goods_e, delta_e*PRY#sectors) + (1-cd_e)*ln( sum(goods_e,
    delta_e*exp(PRY#sectors)^(1-sigma_e)) )/(1-sigma_e*(1-cd_e));
goods_m: POI = cd_m*sum(goods_m, delta_m*PRY#sectors) + (1-cd_m)*ln( sum(goods_m,
    delta_m*exp(PRY#sectors)^(1-sigma_m)) )/(1-sigma_m*(1-cd_m));
```

- These are low-level intermediate demands and price indices. PRY is purchase price of inputs and in G-Cubed it is constant across all purchasing sectors. The "#sectors" syntax is used to expand it to reflect that.

```
TAXC = TCOR*( sum(sectors,PRF) + PRFY ) + sum(goods , TAX *(exp(PRD)*OUG +
    exp(PMR)*IMQ) )/exp(PRID) - sum(sectors, TITC*exp(PRII)*INV )/exp(PRID) -
    sum(goods , TAX *exp(PRS)*EXQ )/exp(PRID) ;
TAXL = INTR*BOND + TAXS;
GOVS = DEFI - GOVL*exp(WAGE-PRID) - TRAN + TAXT - INTR*BOND ;
dest: TAXE = sum( goods, (carcoef*TCAR + btucoef*TBFD)*OUG + (carcoef*TCEX +
    btucoef*TBFX)*EXQ + (carcoef*TCAI + btucoef*TBFI)*IMQ ) / exp(PRID) ;
```