1 TABLO APPENDIX

1.1 The TABLO language

In this document we provide a formal description of the linear form of the model, organised around the TABLO file which implements the model in GEMPACK, via a sequence of excerpts.

The use of the TABLO language in the description of the model ensures that the description represents the theory accurately, and also facilitates the mastering of the model by model users.

The TABLO language is essentially conventional algebra, with names for variables and coefficients chosen to be suggestive of their economic interpretations. We continue this section with a short introduction to the TABLO language.

The TABLO model description defines the percentage-change equations of the model. For example, the CES demand equations, (10) and (11), would appear as:

```
Equation E_x # input demands #
 (all, f, FAC)  x(f) = z - SIGMA*[p(f) - p_f];

Equation E_p_f # input cost index #
  V_F*p_f = sum{f,FAC, V(f)*p(f)};
```

The first word, 'Equation', is a keyword which defines the statement type. Then follows the identifier for the equation, which must be unique. The descriptive text between '#' symbols is optional—it appears in certain report files. The expression '(all, f, FAC)' signifies that the equation is a matrix equation, containing one scalar equation for each element of the set FAC.[1]

Within the equation, the convention is followed of using lower-case letters for the percentage-change variables (x, z, p and p_f), and upper case for the coefficients (SIGMA, V and V_F). Since GEMPACK ignores case, this practice assists only the human reader. An implication is that we cannot use the same sequence of characters, distinguished only by case, to define a variable and a coefficient. The '(f)' suffix indicates that variables and coefficients are vectors, with elements corresponding to the set FAC. A semicolon signals the end of the TABLO statement.

To facilitate portability between computing environments, the TABLO character set is quite restricted—only alphanumerics and a few punctuation marks may be used. The use of Greek letters and subscripts is precluded, and the asterisk, '*', must replace the multiplication symbol '×'.

Sets, coefficients and variables must be explicitly declared, *via* statements such as:

[1] For equation E_x we could have written: (all, j, FAC) x(j) = z - SIGMA*[p(j) - p_f], without affecting simulation results. Our convention that the index, (f), be the same as the initial letter of the set it ranges over, aids comprehension but is not enforced by GEMPACK. By contrast, GAMS (a competing software package) enforces consistent usage of set indices by rigidly connecting indices with the corresponding sets.

```
Set FAC # inputs # (capital, labour, energy);

Coefficient

 (all,f,FAC) V(f) # cost of inputs #;
           V_F    # total cost #;
         SIGMA    # substitution elasticity #;

Variable

 (all,f,FAC) p(f) # price of inputs  #;
 (all,f,FAC) x(f) # demand for inputs  #;
             z    # output #;
           p_f    # input cost index #;
```

As the last two statements in the 'Coefficient' block and the last three in the 'Variable' block illustrate, initial keywords (such as 'Coefficient' and 'Variable') may be omitted if the previous statement was of the same type.

Coefficients must be assigned values, either by reading from file:

```
Read V from file FLOWDATA;

Read SIGMA from file PARAMS;
```

or in terms of other coefficients, using formulae:

```
Formula V_F = sum{f, FAC, V(f)};  ! used in cost index equation !
```

The right hand side of the last statement employs the TABLO summation notation, equivalent to the Σ notation used in standard algebra. It defines the sum over an index f running over the set FAC of the input-cost coefficients, V(f). The statement also contains a comment, i.e., the text between exclamation marks (!). TABLO ignores comments.

Some of the coefficients will be updated during multistep computations. This requires the inclusion of statements such as:

```
Update (all,f,FAC) V(f) = x(f)*p(f);
```

which is the default update statement, causing V(f) to be increased after each step by [x(f) + p(f)]%, where x(f) and p(f) are the percentage changes computed at the previous step.

The sample statements listed above introduce most of the types of statement required for the model. But since all sets, variables and coefficients must be defined before they are used, and since coefficients must be assigned values before appearing in equations, it is necessary for the order of the TABLO statements to be almost the reverse of the order in which they appear above.

The ORANI-IT TABLO Input file is ordered as follows:

- definition of sets;
- declarations of variables;
- declarations of often-used coefficients which are read from files, with associated Read and Update statements;
- declarations of other often-used coefficients which are computed from the data, using associated Formulae; and
- groups of topically-related equations, with some of the groups including statements defining coefficients which are used only within that group.

1.2 Dimensions of the model

Excerpt 1 of the TABLO Input file begins by defining logical names for input and output files. Initial data are stored in the BASEDATA input file. The SUMMARY output file is used to store summary and diagnostic information. Note that BASEDATA and SUMMARY are *logical* names. The actual locations of these files (disk, folder, filename) are chosen by the model user.

The rest of Excerpt 1 defines sets: lists of descriptors for the components of vector variables. Set names appear in upper-case characters. For example, the first Set statement is to be read as defining a set named 'COM' which contains commodity descriptors. The elements of COM (a list of commodity names) are read from the input file (this allows the model to use databases with different numbers of sectors). By contrast the two elements of the set SRC—dom and imp—are listed explicitly.

```
! Excerpt 1 of TABLO input file: !
! Files and sets !

File  BASEDATA  # Input data file #;
(new) SUMMARY  # Output for summary and checking data #;

Set                                                                    !Index!
COM # Commodities# read elements from file BASEDATA header "COM";        ! c !
SRC # Source of commodities # (dom,imp);                                 ! s !
IND # Industries # read elements from file BASEDATA header "IND";        ! i !
OCC # Occupations # read elements from file BASEDATA header "OCC";       ! o !
MAR # Margin commodities # read elements from file BASEDATA header "MAR";! m !
Subset MAR is subset of COM;
Set NONMAR   # Non-margin commodities # = COM - MAR;                      ! n !
```

The commodity and industry classifications of ORANI-IT, reported in Table 1, follow the SUT classification. Among the 63 commodities, six are margin commodities (set MAR, marked by bold interface in Table 1).

Table 1. Industry and commodity classification in ORANI-IT

63 industries and commodities	**Description**
1 Agriculture	Crop and animal production, hunting and related service activities
2 Forestry	Forestry and logging
3 Fishing	Fishing and aquaculture
4 Mining	Mining and quarrying
5 FoodBevCigs	Manufacture of food products, beverages and tobacco products
6 TCF	Manufacture of textiles, wearing apparel and leather products
7 WoodPrd	Manufacture of wood and of products of wood and cork, except furniture; manufacture of articles of straw and plaiting materials
8 PaperPrd	Manufacture of paper and paper products
9 Printing	Printing and reproduction of recorded media
10 CokePetrlRef	Manufacture of coke and refined petroleum products
11 Chemicals	Manufacture of chemicals and chemical products
12 Pharma	Manufacture of basic pharmaceutical products and pharmaceutical preparations
13 RubberPlast	Manufacture of rubber and plastic products
14 ONonMetalPrd	Manufacture of other non-metallic mineral products
15 BasicMetals	Manufacture of basic metals
16 FabMetalPrd	Manufacture of fabricated metal products, except machinery and equipment
17 Electronics	Manufacture of computer, electronic and optical products
18 ElectEquip	Manufacture of electrical equipment

19 OthMachinery	Manufacture of machinery and equipment n.e.c.
20 MotorVehic	Manufacture of motor vehicles, trailers and semi-trailers
21 OthTrnsEquip	Manufacture of other transport equipment
22 FurnitOthMan	Manufacture of furniture; other manufacturing
23 EqpRepair	Repair and installation of machinery and equipment
24 ElectricGas	Electricity, gas, steam and air conditioning supply
25 Water	Water collection, treatment and supply.
26 WasteTreatmt	Sewerage; waste collection, treatment and disposal activities; materials recovery; remediation activities and other waste management services
27 Construction	Construction
28 CarTrade	**Wholesale and retail trade and repair of motor vehicles and motorcycles**
29 WholeslTrade	**Wholesale trade, except of motor vehicles and motorcycles**
30 RetailTrade	**Retail trade, except of motor vehicles and motorcycles**
31 LandTransprt	**Land transport and transport via pipelines**
32 WatrTransprt	**Water transport**
33 AirTransprt	**Air transport**
34 TransprtSvc	Warehousing and support activities for transportation
35 Post	Postal and courier activities
36 HotelRestrnt	Accommodation and food service activities
37 Publishing	Publishing activities
38 MediaFilm	Motion picture, video and television programme production, sound recording and music publishing activities; programming and broadcasting activities
39 Telecomms	Telecommunications
40 ITservices	Computer programming, consultancy and related activities; information service activities
41 FinancialSvc	Financial service activities, except insurance and pension funding
42 InsuranceSvc	Insurance, reinsurance and pension funding, except compulsory social security
43 OthFinInsSvc	Activities auxiliary to financial services and insurance activities
44 RealEstate	Real estate activities
45 LegalAccount	Legal and accounting activities; activities of head offices; management consultancy activities
46 ArchEngSvc	Architectural and engineering activities; technical testing and analysis
47 ScientficSvc	Scientific research and development
48 Advertising	Advertising and market research
49 OtherProfSvc	Other professional, scientific and technical activities; veterinary activities
50 RentlLeasing	Rental and leasing activities
51 EmploymntSvc	Employment activities
52 TravelSvc	Travel agency, tour operator reservation service and related activities
53 OthBusSvc	Security and investigation activities; services to buildings and landscape activities; office administrative, office support and other business support activities
54 PubAdmin	Public administration and defence; compulsory social security
55 Education	Education
56 Health	Human health activities
57 SocialWork	Social work activities
58 ArtLibBets	Creative, arts and entertainment activities; libraries, archives, museums and other cultural activities; gambling and betting activities
59 SportsRec	Sports activities and amusement and recreation activities
60 ActivMembOrg	Activities of membership organisations
61 Repairs	Repair of computers and personal and household goods
62 OthPersnlSvc	Other personal service activities
63 MaidsOwnProd	Activities of households as employers; undifferentiated goods- and services-producing activities of households for own use

1.3 The ORANI-IT naming system

The TABLO Input file defines a multitude of variables and coefficients that are used in the model's equations. As far as possible, names for variables and coefficients conform to a system in which each name consists of 2 or more parts, as follows:

first, a letter or letters indicating the type of variable, for example,

a	technical change
del	ordinary (rather than percentage) change
f	shift variable
H	indexing parameter
p	price, local currency
pf	price, foreign currency
S	input share
SIGMA	elasticity of substitution
t	tax
V	levels value, local currency
w	percentage-change value, local currency
x	input quantity;

second, one of the digits 0 to 6 indicating user, that is,

1	current production
2	investment
3	consumption
4	export
5	government
6	inventories
0	all users, or user distinction irrelevant;

third (optional), three or more letters giving further information, for example,

bas	(often omitted) basic—not including margins or taxes
cap	capital
cif	imports at border prices
imp	imports (duty paid)
lab	labour
lnd	land
lux	linear expenditure system (supernumerary part)
mar	margins
oct	other cost tickets
prim	all primary factors (land, labour or capital)
pur	at purchasers' prices
sub	linear expenditure system (subsistence part)
tar	tariffs
tax	indirect taxes
tot	total or average over all inputs for some user;

fourth (optional), an underscore character, indicating that this variable is an aggregate or average, with subsequent letters showing over which sets the underlying variable has been summed or averaged, for example,

_c	over COM (commodities),
_s	over SRC (dom + imp),
_i	over IND (industries),
_io	over IND and OCC (skills).

Although GEMPACK does not distinguish between upper and lower case, we use:

lower case for variable names and set indices;
upper case for set and coefficient names; and
initial letter upper case for TABLO keywords.

1.4 Core data coefficients and related variables

The TABLO file opens with statements indicating data to be read from file. The data items defined in these statements appear as coefficients in the model's equations. The statements define coefficient names (which all appear in upper-case characters), the locations from which the data are to be read, variable names (in lower-case), and formulae for the data updates which are necessary in computing multi-step solutions to the model.

1.4.1 *Basic flows*

The excerpts group the data according to the rows of Figure 1. Excerpt 2 begins by defining coefficients representing the basic commodity flows corresponding to row 1 (direct flows) of the figure, i.e., the flow matrices V1BAS, V2BAS, and so on. Preceding the coefficient names are their dimensions, indicated using the "all" qualifier and the sets defined in Excerpt 1. For example, the first 'Coefficient' statement defines a data item V1BAS(c,s,i) which is the basic value (indicated by 'BAS') of a flow of intermediate inputs (indicated by '1') of commodity c from source s to user industry i. The first 'Read' statement indicates that this data item is stored on file BASEDATA with header '1BAS'. (A GEMPACK data file consists of a number of data items such as arrays of real numbers. Each data item is identified by a unique key or 'header').

Each of these flows is the product of a price and a quantity. The excerpt goes on to define these variables. Unless otherwise stated, all variables are percentage changes—to indicate this, their names appear in lower-case letters. Preceding the names of the variables are their dimensions, indicated using the sets defined in Excerpt 1. For example, the first variable statement defines a matrix variable x1 (indexed by commodity, source, and using industry) the elements of which are percentage changes in the direct demands by producers for source-specific intermediate inputs. This is the quantity variable corresponding to V1BAS.

```
! Excerpt 2 of TABLO input file: !
! Data coefficients and variables relating to basic commodity flows !

Coefficient  ! Basic flows of commodities (excluding margin demands)!
 (all,c,COM)(all,s,SRC)(all,i,IND)  V1BAS(c,s,i)  # Intermediate basic flows #;
 (all,c,COM)(all,s,SRC)(all,i,IND)  V2BAS(c,s,i)  # Investment basic flows #;
 (all,c,COM)(all,s,SRC)             V3BAS(c,s)    # Household basic flows #;
 (all,c,COM)                        V4BAS(c)      # Export basic flows #;
 (all,c,COM)(all,s,SRC)             V5BAS(c,s)    # Government basic flows #;
 (all,c,COM)(all,s,SRC)             V6BAS(c,s)    # Inventories basic flows #;
Read
 V1BAS from file BASEDATA header "1BAS";
 V2BAS from file BASEDATA header "2BAS";
 V3BAS from file BASEDATA header "3BAS";
 V4BAS from file BASEDATA header "4BAS";
 V5BAS from file BASEDATA header "5BAS";
 V6BAS from file BASEDATA header "6BAS";
Variable  ! Variables used to update above flows !
 (all,c,COM)(all,s,SRC)(all,i,IND)  x1(c,s,i)  # Intermediate basic demands #;
 (all,c,COM)(all,s,SRC)(all,i,IND)  x2(c,s,i)  # Investment basic demands #;
 (all,c,COM)(all,s,SRC)             x3(c,s)    # Household basic demands #;
 (all,c,COM)                        x4(c)      # Export basic demands #;
 (all,c,COM)(all,s,SRC)             x5(c,s)    # Government basic demands #;
```

```
 (change) (all,c,COM)(all,s,SRC)    delx6(c,s) # Inventories demands #;
 (all,c,COM)(all,s,SRC)             p0(c,s)    # Basic prices for local users #;
 (all,c,COM)                        pe(c)      # Basic price of exportables #;
 (change)(all,c,COM)(all,s,SRC)  delV6(c,s)    # Value of inventories #;
Update
 (all,c,COM)(all,s,SRC)(all,i,IND)  V1BAS(c,s,i)  = p0(c,s)*x1(c,s,i);
 (all,c,COM)(all,s,SRC)(all,i,IND)  V2BAS(c,s,i)  = p0(c,s)*x2(c,s,i);
 (all,c,COM)(all,s,SRC)             V3BAS(c,s)    = p0(c,s)*x3(c,s);
 (all,c,COM)                        V4BAS(c)      = pe(c)*x4(c);
 (all,c,COM)(all,s,SRC)             V5BAS(c,s)    = p0(c,s)*x5(c,s);
 (change)(all,c,COM)(all,s,SRC)     V6BAS(c,s)    = delV6(c,s);
```

The last in the group of quantity variables, delx6, is preceded by the 'Change' qualifier to indicate that it is an ordinary (rather than percentage) change. Changes in inventories may be either positive or negative. Our multistep solution procedure requires that large changes be broken into a sequence of small changes. However, no sequence of small *percentage* changes allows a number to change sign—at least one change must exceed -100%. Thus, for variables that may, in the levels, change sign, we prefer to use ordinary changes. The names of ordinary change variables often start with the letters "del".

Next come two price variables. A matrix variable p0 (indexed by commodity and source), shows percentage changes in the basic prices which are common to all local users. These basic prices do not include the cost of margins and taxes. Exports have their own basic prices, pe. Potentially, the pe could be different from the domestic part of p0[2].

Finally, the variable delV6 is used in the update statements which appear next.

The first 'Update' statement indicates that the flow V1BAS(c,s,i) should be updated using the default update formula, which is used for a data item which is a product of two (or more) of the model's variables. For an item of the form V = PX, the formula for the updated value V^U is:

$$V^U = V^0 + \Delta(PX) = V^0 + X^0\Delta P + P^0\Delta X$$

$$= V^0 + P^0X^0\left(\frac{\Delta P}{P^0}+\frac{\Delta X}{\Delta^0}\right) = V^0 + V^0\left(\frac{p}{100}+\frac{x}{100}\right) \qquad (1)$$

where V^0, P^0 and X^0 are the pre-update values, and p and x are the percentage changes of the variables P and X. For the data item V1BAS(c,s,i) the relevant percentage-change variables are p0(c,s) (the basic-value price of commodity c from source s) and x1(c,s,i) (the demand by user industry i for intermediate inputs of commodity c from source s).

Not all of the model's data items are amenable to update *via* default Updates. For example, the inventories flows, V6BAS, might change sign, and so must not be updated with percentage change variables. In such a case, the Update statement must contain an explicit formula for the ordinary change in the data item: this is indicated by the word 'Change' in parentheses. For V6BAS we represent the change by an ordinary-change variable, delV6. The Update formula (1) then becomes simply:

$$V^U = V^0 + \Delta V. \qquad (2)$$

An equation defining the delV6 variable appears later on.

---

[2] Exports (V4BAS) are valued with price vector pe. Unless we activate the optional CET transformation between goods destined for export and those for local use, the pe are identical to the domestic part of p0. See Excerpt 13.

1.4.2 *Margin flows*

The coefficients and variables of Excerpt 3 are associated with row 2 (margins) of Figure 1, i.e., the flow matrices V1MAR, V2MAR, and so on. These are the quantities of retail and wholesale services or transport needed to deliver each basic flow to the user. For example V3MAR(c,s,m) is the value of margin type m used to deliver commodity type c from source s to households (user 3). The model assumes that margin services are domestically produced and are valued at basic prices—represented by the variable p0dom, which (we shall see later) is simply a synonym for the domestic part of the basic price matrix, p0 [i.e., p0dom(c) = p0(c,"dom")].

```
! Excerpt 3 of TABLO input file: !
! Data coefficients and variables relating to margin flows !

Coefficient
 (all,c,COM)(all,s,SRC)(all,i,IND)(all,m,MAR)
                                   V1MAR(c,s,i,m)  # Intermediate margins #;
 (all,c,COM)(all,s,SRC)(all,i,IND)(all,m,MAR)
                                   V2MAR(c,s,i,m)  # Investment margins #;
 (all,c,COM)(all,s,SRC)(all,m,MAR) V3MAR(c,s,m)    # Households margins #;
 (all,c,COM)(all,m,MAR)            V4MAR(c,m)      # Export margins #;
 (all,c,COM)(all,s,SRC)(all,m,MAR) V5MAR(c,s,m)    # Government margins #;
Read
 V1MAR from file BASEDATA header "1MAR";
 V2MAR from file BASEDATA header "2MAR";
 V3MAR from file BASEDATA header "3MAR";
 V4MAR from file BASEDATA header "4MAR";
 V5MAR from file BASEDATA header "5MAR";
Variable ! Variables used to update above flows !
 (all,c,COM)(all,s,SRC)(all,i,IND)(all,m,MAR)
                                   x1mar(c,s,i,m)# Intermediate margin demand #;
 (all,c,COM)(all,s,SRC)(all,i,IND)(all,m,MAR)
                                   x2mar(c,s,i,m)# Investment margin demands #;
 (all,c,COM)(all,s,SRC)(all,m,MAR) x3mar(c,s,m)  # Household margin demands #;
 (all,c,COM)(all,m,MAR)            x4mar(c,m)    # Export margin demands #;
 (all,c,COM)(all,s,SRC)(all,m,MAR) x5mar(c,s,m)  # Government margin demands #;
 (all,c,COM)           p0dom(c) # Basic price of domestic goods = p0(c,"dom") #;
Update
 (all,c,COM)(all,s,SRC)(all,i,IND)(all,m,MAR)
                                   V1MAR(c,s,i,m)  = p0dom(m)*x1mar(c,s,i,m);
 (all,c,COM)(all,s,SRC)(all,i,IND)(all,m,MAR)
                                   V2MAR(c,s,i,m)  = p0dom(m)*x2mar(c,s,i,m);
 (all,c,COM)(all,s,SRC)(all,m,MAR) V3MAR(c,s,m)    = p0dom(m)*x3mar(c,s,m);
 (all,c,COM)(all,m,MAR)            V4MAR(c,m)      = p0dom(m)*x4mar(c,m);
 (all,c,COM)(all,s,SRC)(all,m,MAR) V5MAR(c,s,m)    = p0dom(m)*x5mar(c,s,m);
```

1.4.3 *User-specific commodity taxes*

Excerpt 4 contains coefficients and variables associated with row 3 (commodity taxes) of Figure 1, i.e., the flow matrices V1TAX, V2TAX, and so on. These all have the same dimensions as the corresponding basic flows.

```
! Excerpt 4 of TABLO input file: !
! Data coefficients and variables relating to commodity taxes !

Coefficient  ! Taxes on Basic Flows!
 (all,c,COM)(all,s,SRC)(all,i,IND)(all,t,ALLCTAX)  V1TAX(c,s,i,t) # Taxes on
intermediate #;
 (all,c,COM)(all,s,SRC)(all,i,IND)(all,t,ALLCTAX)  V2TAX(c,s,i,t) # Taxes on investment;
 (all,c,COM)(all,s,SRC)(all,t,ALLCTAX)             V3TAX(c,s,t)   # Taxes on households;
 (all,c,COM)(all,t,ALLCTAX)                        V4TAX(c,t)     # Taxes on export #;
 (all,c,COM)(all,s,SRC)(all,t,ALLCTAX)             V5TAX(c,s,t)   # Taxes on government;
Read
 V1TAX from file BASEDATA header "1TXC";
 V2TAX from file BASEDATA header "2TXC";
 V3TAX from file BASEDATA header "3TXC";
 V4TAX from file BASEDATA header "4TXC";
 V5TAX from file BASEDATA header "5TXC";

Variable
 (change)(all,c,COM)(all,s,SRC)(all,i,IND)(all,t,ALLCTAX) delV1TAX(c,s,i,t) # Interm tax
rev #;
 (change)(all,c,COM)(all,s,SRC)(all,i,IND)(all,t,ALLCTAX) delV2TAX(c,s,i,t) # Invest tax
rev #;
 (change)(all,c,COM)(all,s,SRC)(all,t,ALLCTAX)            delV3TAX(c,s,t)   # H'hold tax
rev #;
 (change)(all,c,COM)(all,t,ALLCTAX)                       delV4TAX(c,t)     # Export tax
rev #;
 (change)(all,c,COM)(all,s,SRC)(all,t,ALLCTAX)            delV5TAX(c,s,t)   # Govmnt tax
rev #;
Update
 (change)(all,c,COM)(all,s,SRC)(all,i,IND)(all,t,ALLCTAX)  V1TAX(c,s,i,t) =
delV1TAX(c,s,i,t);
 (change)(all,c,COM)(all,s,SRC)(all,i,IND)(all,t,ALLCTAX)  V2TAX(c,s,i,t) =
delV2TAX(c,s,i,t);
 (change)(all,c,COM)(all,s,SRC)(all,t,ALLCTAX)             V3TAX(c,s,t)   =
delV3TAX(c,s,t);
 (change)(all,c,COM)(all,t,ALLCTAX)                        V4TAX(c,t)     =
delV4TAX(c,t);
 (change)(all,c,COM)(all,s,SRC)(all,t,ALLCTAX)             V5TAX(c,s,t)   =
delV5TAX(c,s,t);
```

ORANI-IT treats commodity taxes in great detail—the tax levied on each basic flow is separately identified. The disaggregated structure allows us to simulate the effects of commodity—user-and-tax type-specific tax changes, such as an increased in the excise tax on Cokes and refined petroleum used by the Land transport industry.

The tax flows are updated by corresponding ordinary change variables: equations determining these appear later on.

1.4.4 *Factor payments and other flows data*

Excerpt 5 of the TABLO Input file corresponds to the remaining rows of Figure 1. There are coefficient matrices for payments to labour, capital, and land, and two sorts of production tax. Then are listed the corresponding price and quantity variables. For the production tax V1PTX, the corresponding ordinary change variable delV1PTX is used in a Change Update statement: an equation determining this variable appears later. The other flow coefficients in this group are simply the products of prices and quantities. Hence, they can be updated *via* default Update statements.

Excerpt 5 also defines the import duty vector V0TAR. Treatment of the last item in the flows database, the multiproduction matrix MAKE, showing output of commodities by each industry, is deferred to a later section.

```
! Excerpt 5 of TABLO input file: !
! Data coefficients for primary-factor flows, other industry costs, and tariffs!

Coefficient
 (all,i,IND)(all,o,OCC)  V1LAB(i,o)  # Wage bill matrix #;
 (all,i,IND)             V1CAP(i)    # Capital rentals #;
 (all,i,IND)             V1LND(i)    # Land rentals #;
 (all,i,IND)(all,p,ALLPTAX) V1PTX(i,p)    # Production tax #;;
 (all,i,IND)             V1OCT(i)    # Other cost tickets #;
Read
 V1LAB from file BASEDATA header "1LAB";
 V1CAP from file BASEDATA header "1CAP";
 V1LND from file BASEDATA header "1LND";
 V1PTX from file BASEDATA header "1PTX";
 V1OCT from file BASEDATA header "1OCT";
Variable
 (all,i,IND)(all,o,OCC)   x1lab(i,o)  # Employment by industry and occupation #;
 (all,i,IND)(all,o,OCC)   p1lab(i,o)  # Wages by industry and occupation #;
 (all,i,IND)  x1cap(i)        # Current capital stock #;
 (all,i,IND)  p1cap(i)        # Rental price of capital #;
 (all,i,IND)  x1lnd(i)        # Use of land #;
 (all,i,IND)  p1lnd(i)        # Rental price of land #;
(change)(all,i,IND)(all,p,ALLPTAX)
              delV1PTX(i,p)   # Ordinary change in production tax revenue #;
 (all,i,IND)  x1oct(i)        # Demand for "other cost" tickets #;
 (all,i,IND)  p1oct(i)        # Price of "other cost" tickets #;
Update
 (all,i,IND)(all,o,OCC)  V1LAB(i,o)  = p1lab(i,o)*x1lab(i,o);
 (all,i,IND)             V1CAP(i)    = p1cap(i)*x1cap(i);
 (all,i,IND)             V1LND(i)    = p1lnd(i)*x1lnd(i);
(change)(all,i,IND)(all,p,ALLPTAX)   V1PTX(i,p)  = delV1PTX(i,p);
 (all,i,IND)             V1OCT(i)    = p1oct(i)*x1oct(i);

! Data coefficients relating to import duties !
Coefficient (all,c,COM) V0TAR(c)  # Tariff revenue #;
Read V0TAR from file BASEDATA header "0TAR";
Variable (all,c,COM) (change) delV0TAR(c) # Ordinary change in tariff revenue #;
Update (change)  (all,c,COM) V0TAR(c) = delV0TAR(c);
```

1.4.5 *Purchasers' values*

Excerpt 6 defines the values at purchasers' prices of the commodity flows identified in Figure 1. These aggregates will be used in several different equation blocks. The definitions use the TABLO summation notation, explained in Section 1. For example, the first formula in Excerpt 6 contains the term:

```
sum{m,MAR, V1MAR(c,s,i,m) }
```

This defines the sum, over an index *m* running over the set of margins commodities (MAR), of the input-output data flows V1MAR(c,s,i,m). This sum is the total value of margins commodities required to facilitate the flow of intermediate inputs of commodity *c* from source *s* to user industry *i.* Adding this sum to the basic value of the intermediate-input flow and the associated indirect tax, gives the purchaser's-price value of the flow.

Next are defined purchasers' price variables, which include basic, margin and tax components. Equations to determine these variables appear in Excerpt 22 below.

```
! Excerpt 6 of TABLO input file: !
! Coefficients and variables for purchaser's prices (basic + margins + taxes) !

Coefficient ! Flows at purchasers prices !
 (all,c,COM)(all,s,SRC)(all,i,IND)  V1PUR(c,s,i)  # Intermediate purch. value #;
 (all,c,COM)(all,s,SRC)(all,i,IND)  V2PUR(c,s,i)  # Investment purch. value #;
 (all,c,COM)(all,s,SRC)             V3PUR(c,s)    # Households purch. value #;
 (all,c,COM)                        V4PUR(c)      # Export purch. value #;
 (all,c,COM)(all,s,SRC)             V5PUR(c,s)    # Government purch. value #;
Formula
(all,c,COM)(all,s,SRC)(all,i,IND)
   V1PUR(c,s,i)  = V1BAS(c,s,i) + Sum{t,ALLCTAX, V1TAX(c,s,i,t)} + sum{m,MAR,
V1MAR(c,s,i,m)};
 (all,c,COM)(all,s,SRC)(all,i,IND)
   V2PUR(c,s,i)  = V2BAS(c,s,i) + Sum{t,ALLCTAX, V2TAX(c,s,i,t)} + sum{m,MAR,
V2MAR(c,s,i,m)};
 (all,c,COM)(all,s,SRC)
   V3PUR(c,s)    = V3BAS(c,s)   + Sum{t,ALLCTAX, V3TAX(c,s,t)}   + sum{m,MAR,
V3MAR(c,s,m)};
 (all,c,COM)
   V4PUR(c)      = V4BAS(c)     + Sum{t,ALLCTAX, V4TAX(c,t)}     + sum{m,MAR,
V4MAR(c,m)};
 (all,c,COM)(all,s,SRC)
   V5PUR(c,s)    = V5BAS(c,s)   + Sum{t,ALLCTAX, V5TAX(c,s,t)}   + sum{m,MAR,
V5MAR(c,s,m)};
Variable ! Purchasers prices !
 (all,c,COM)(all,s,SRC)(all,i,IND) p1(c,s,i)# Purchaser's price, intermediate #;
 (all,c,COM)(all,s,SRC)(all,i,IND) p2(c,s,i)# Purchaser's price, investment #;
 (all,c,COM)(all,s,SRC)            p3(c,s)  # Purchaser's price, household #;
 (all,c,COM)                       p4(c)    # Purchaser's price, exports,loc$ #;
 (all,c,COM)(all,s,SRC)            p5(c,s)  # Purchaser's price, government #;
```

1.5 The equation system

The rest of the TABLO Input file is an algebraic specification of the linear form of the model, with the equations organised into a number of blocks. Each Equation statement begins with a name and (optionally) a description. For ORANI-IT, the equation name normally consists of the characters E_ followed by the name of the left-hand-side variable. Except where indicated, the variables are percentage changes. Variables are in lower-case characters and coefficients in upper case. Variables and coefficients are defined as the need arises. Readers who have followed the TABLO file so far should have no difficulty in reading the equations in the TABLO notation.

```
! Excerpt 7 of TABLO input file: !
! Occupational composition of labour demand !

Coefficient
 (parameter)(all,i,IND) SIGMA1LAB(i) # CES substitution between skill types #;
 (all,i,IND) V1LAB_O(i)   # Total labour bill in industry i #;
             TINY    # Small number to prevent zerodivides or singular matrix #;
Read SIGMA1LAB from file BASEDATA header "SLAB";
Formula
 (all,i,IND) V1LAB_O(i) = sum{o,OCC, V1LAB(i,o)};
             TINY       = 0.000000000001;
Variable
 (all,i,IND) p1lab_o(i) # Price to each industry of labour composite #;
 (all,i,IND) x1lab_o(i) # Effective labour input #;
```

```
Equation
 E_x1lab   # Demand for labour by industry and skill group #
  (all,i,IND)(all,o,OCC)
   x1lab(i,o) = x1lab_o(i) - SIGMA1LAB(i)*[p1lab(i,o) - p1lab_o(i)];
 E_p1lab_o # Price to each industry of labour composite #
  (all,i,IND) [TINY+V1LAB_O(i)]*p1lab_o(i) = sum{o,OCC, V1LAB(i,o)*p1lab(i,o)};



! Excerpt 8 of TABLO input file: !
! Primary factor proportions !

Coefficient
(parameter)(all,i,IND) SIGMA1PRIM(i) # CES substitution, primary factors #;
Read SIGMA1PRIM from file BASEDATA header "P028";
Coefficient (all,i,IND) V1PRIM(i) # Total factor input to industry i#;
Formula     (all,i,IND) V1PRIM(i) = V1LAB_O(i)+ V1CAP(i) + V1LND(i);
Variable
 (all,i,IND) p1prim(i)  # Effective price of primary factor composite #;
 (all,i,IND) x1prim(i)  # Primary factor composite #;
 (all,i,IND) a1lab_o(i) # Labor-augmenting technical change #;
 (all,i,IND) a1cap(i)   # Capital-augmenting technical change #;
 (all,i,IND) a1lnd(i)   # Land-augmenting technical change #;
(change)(all,i,IND) delV1PRIM(i)# Ordinary change in cost of primary factors #;

Equation
 E_x1lab_o  # Industry demands for effective labour #
  (all,i,IND)  x1lab_o(i) - a1lab_o(i) =
   x1prim(i) - SIGMA1PRIM(i)*[p1lab_o(i) + a1lab_o(i) - p1prim(i)];

 E_p1cap  # Industry demands for capital #
  (all,i,IND)  x1cap(i) - a1cap(i) =
   x1prim(i) - SIGMA1PRIM(i)*[p1cap(i) + a1cap(i) - p1prim(i)];

 E_p1lnd  # Industry demands for land #
  (all,i,IND)  x1lnd(i) - a1lnd(i) =
   x1prim(i) - SIGMA1PRIM(i)*[p1lnd(i) + a1lnd(i) - p1prim(i)];

 E_p1prim  # Effective price term for factor demand equations #
  (all,i,IND)  V1PRIM(i)*p1prim(i) = V1LAB_O(i)*[p1lab_o(i) + a1lab_o(i)]
    + V1CAP(i)*[p1cap(i) + a1cap(i)] + V1LND(i)*[p1lnd(i) + a1lnd(i)];

 E_delV1PRIM  # Ordinary change in total cost of primary factors #
  (all,i,IND) 100*delV1PRIM(i) = V1CAP(i)  * [p1cap(i) + x1cap(i)]
                               +  V1LND(i)  * [p1lnd(i) + x1lnd(i)]
                    + sum{o,OCC, V1LAB(i,o)* [p1lab(i,o) + x1lab(i,o)]};


! Excerpt 9 of TABLO input file: !
! Import/domestic composition of intermediate demands !

Variable
 (all,c,COM)(all,s,SRC)(all,i,IND) a1(c,s,i) # Intermediate basic tech change #;
 (all,c,COM)(all,i,IND) x1_s(c,i)    # Intermediate use of imp/dom composite #;
 (all,c,COM)(all,i,IND) p1_s(c,i)    # Price, intermediate imp/dom composite #;
 (all,i,IND)            p1mat(i)     # Intermediate cost price index #;
 (all,i,IND)            p1var(i)     # Short-run variable cost price index #;
Coefficient
 (parameter)(all,c,COM) SIGMA1(c)    # Armington elasticities: intermediate #;
 (all,c,COM)(all,i,IND) V1PUR_S(c,i) # Dom+imp intermediate purch. value #;
 (all,c,COM)(all,s,SRC)(all,i,IND) S1(c,s,i) # Intermediate source shares #;
 (all,i,IND)            V1MAT(i)     # Total intermediate cost for industry i #;
```

```
 (all,i,IND)            V1VAR(i)     # Short-run variable cost for industry i #;
Read SIGMA1 from file BASEDATA header "1ARM";
Zerodivide default 0.5;
Formula
 (all,c,COM)(all,i,IND)            V1PUR_S(c,i) = sum{s,SRC, V1PUR(c,s,i)};
 (all,c,COM)(all,s,SRC)(all,i,IND) S1(c,s,i)    = V1PUR(c,s,i) / V1PUR_S(c,i);
 (all,i,IND)                       V1MAT(i)     = sum{c,COM, V1PUR_S(c,i)};
 (all,i,IND)                       V1VAR(i)     = V1MAT(i) + V1LAB_O(i);
Zerodivide off;

Equation E_x1  # Source-specific commodity demands #
 (all,c,COM)(all,s,SRC)(all,i,IND)
  x1(c,s,i)-a1(c,s,i) = x1_s(c,i) -SIGMA1(c)*[p1(c,s,i) +a1(c,s,i) -p1_s(c,i)];

Equation E_p1_s  # Effective price of commodity composite #
 (all,c,COM)(all,i,IND)
  p1_s(c,i) = sum{s,SRC, S1(c,s,i)*[p1(c,s,i) + a1(c,s,i)]};

Equation E_p1mat  # Intermediate cost price index #
 (all,i,IND)
  p1mat(i) = sum{c,COM, sum{s,SRC, (V1PUR(c,s,i)/ID01[V1MAT(i)])*p1(c,s,i)}};

Equation E_p1var  # Short-run variable cost price index #
 (all,i,IND)
  p1var(i) = [1/V1VAR(i)]*[V1MAT(i)*p1mat(i) + V1LAB_O(i)*p1lab_o(i)];


! Excerpt 10 of TABLO input file: !
! Top nest of industry input demands !
Variable
 (all,i,IND)  x1tot(i)        # Activity level or value-added #;
 (all,i,IND)  a1prim(i)       # All factor augmenting technical change #;
 (all,i,IND)  a1tot(i)        # All input augmenting technical change #;
 (all,i,IND)  p1tot(i)        # Average input/output price #;
 (all,i,IND)  a1oct(i)        # "Other cost" ticket augmenting techncal change#;
 (all,c,COM)(all,i,IND) a1_s(c,i) # Tech change, int'mdiate imp/dom composite #;

Equation E_x1_s  # Demands for commodity composites #
 (all,c,COM)(all,i,IND)  x1_s(c,i) - [a1_s(c,i) + a1tot(i)] = x1tot(i);

Equation E_x1prim  # Demands for primary factor composite #
 (all,i,IND)  x1prim(i) - [a1prim(i) + a1tot(i)] = x1tot(i);

Equation E_x1oct  # Demands for other cost tickets #
 (all,i,IND)  x1oct(i) - [a1oct(i) + a1tot(i)] = x1tot(i);


! Excerpt 11 of TABLO input file: !
! Output cost inclusive of production tax !

Coefficient
 (all,i,IND)  V1CST(i)    # Total cost of industry i #;
 (all,i,IND)  V1TOT(i)    # Total industry cost plus tax #;
 (all,i,IND)(all,p,ALLPTAX) PTXRATE(i,p)   # Rate of production tax #;
Formula
 (all,i,IND)  V1CST(i)    = V1PRIM(i) + V1OCT(i) + V1MAT(i);
 (all,i,IND)  V1TOT(i)    = V1CST(i) + V1PTX(i);
(all,i,IND)(all,p,ALLPTAX)  PTXRATE(i,p)  = V1PTX(i,p)/V1CST(i);
Variable
 (change)(all,i,IND) delV1CST(i)   # Change in ex-tax cost of production #;
 (change)(all,i,IND) delV1TOT(i)   # Change in tax-inc cost of production #;
(change)(all,i,IND)(all,p,ALLPTAX) delPTXRATE(i,p) # Change in rate of production tax #;
```

```
Equation
 E_delV1CST  (all,i,IND) delV1CST(i) = delV1PRIM(i) +
  sum{c,COM,sum{s,SRC,  0.01*V1PUR(c,s,i)*[p1(c,s,i) + x1(c,s,i)]}}
                    +  0.01*V1OCT(i)    *[p1oct(i)  + x1oct(i)];

E_delV1PTX  (all,i,IND)(all,p,ALLPTAX) delV1PTX(i,p) =
                      PTXRATE(i,p)*delV1CST(i)  + V1CST(i) * delPTXRATE(i,p);

 E_delV1TOT  (all,i,IND) delV1TOT(i) = delV1CST(i) + Sum{p,ALLPTAX, delV1PTX(i,p)};

 E_p1tot     (all,i,IND) V1TOT(i)*[p1tot(i) + x1tot(i)] = 100*delV1TOT(i);

Variable (all,i,IND) p1cst(i) # Index of production costs (for AnalyseGE) #;
Equation E_p1cst  (all,i,IND) p1cst(i) =  [1/V1CST(i)]*[
  sum{c,COM,sum{s,SRC,  V1PUR(c,s,i)*p1(c,s,i)}}
                    +  V1OCT(i)    *p1oct(i)
                    +  V1CAP(i)    *p1cap(i)
                    +  V1LND(i)    *p1lnd(i)
          + sum{o,OCC, V1LAB(i,o)  *p1lab(i,o)}];



! Excerpt 12 of TABLO input file: !
! Output mix of commodities !

Coefficient (all,c,COM)(all,i,IND) MAKE(c,i) # Multiproduction matrix #;
Variable
 (all,c,COM)(all,i,IND)  q1(c,i)   # Output by commodity and industry #;
 (all,c,COM)(all,i,IND)  pq1(c,i)  # Price of com c produced by ind i #;
 (all,c,COM)  p0com(c)   # General output price of locally-produced commodity #;
Read MAKE from file BASEDATA header "MAKE";
Update (all,c,COM)(all,i,IND) MAKE(c,i)= pq1(c,i)*q1(c,i);
Variable
 (all,c,COM)  x0com(c)        # Output of commodities #;
Coefficient
 (parameter)(all,i,IND) SIGMA1OUT(i) # CET transformation elasticities #;
Read SIGMA1OUT from file BASEDATA header "SCET";
Equation E_q1  # Supplies of commodities by industries #
 (all,c,COM)(all,i,IND)
  q1(c,i) = x1tot(i) + SIGMA1OUT(i)*[p0com(c) - p1tot(i)];
Coefficient
 (all,i,IND) MAKE_C(i) # All production by industry i #;
 (all,c,COM) MAKE_I(c) # Total production of commodities #;
Formula
 (all,i,IND) MAKE_C(i) = sum{c,COM, MAKE(c,i)};
 (all,c,COM) MAKE_I(c) = sum{i,IND, MAKE(c,i)};

Equation E_x1tot # Average price received by industries #
 (all,i,IND) p1tot(i) = sum{c,COM, [MAKE(c,i)/MAKE_C(i)]*pq1(c,i)};

Equation
 E_pq1  # Each industry gets the same price for a given commodity #
 (all,c,COM)(all,i,IND) pq1(c,i) = p0com(c);
 E_x0com # Total output of commodities (as simple addition) #
 (all,c,COM) x0com(c) = sum{i,IND, [MAKE(c,i)/MAKE_I(c)]*q1(c,i)};
```

```
! Excerpt 13 of TABLO input file: !
! CET between outputs for local and export markets !
Variable
 (all,c,COM)  x0dom(c)  # Output of commodities for local market #;
Coefficient
 (all, c,COM) EXPSHR(c) # Share going to exports #;
 (all, c,COM) TAU(c)    # 1/Elast. of transformation, exportable/locally used #;
Zerodivide default 0.5;
Formula
 (all,c,COM) EXPSHR(c) = V4BAS(c)/MAKE_I(c);
 (all,c,COM) TAU(c) = 0.0; ! if zero, p0dom = pe, and CET is nullified !
Zerodivide off;
Equation E_x0dom  # Supply of commodities to export market #
 (all,c,COM) TAU(c)*[x0dom(c) - x4(c)] = p0dom(c) - pe(c);
Equation E_pe     # Supply of commodities to domestic market #
 (all,c,COM) x0com(c) = [1.0-EXPSHR(c)]*x0dom(c) + EXPSHR(c)*x4(c);
Equation E_p0com  # Zero pure profits in transformation #
 (all,c,COM) p0com(c) = [1.0-EXPSHR(c)]*p0dom(c) + EXPSHR(c)*pe(c);
```

The names of the prices, quantities and flows in the two CET nests of Excerpts 12 and 13 are shown below:

	Joint Production CET Nest				
Type of Variable	Industry Output	Commodity Outputs	Undifferentiated Commodity	Local Destination	Export Destination
% quantity	x1tot(i)	q1(c,i)	x0com(c)	x0dom(c)	x4(c)
% price	p1tot(i)	p0com(c)	p0com(c)	p0dom(c) = p0(c,"dom")	pe(c)
Value of flow	V1TOT(i)	MAKE(c,i)	SALES(c)	DOMSALES(c)	V4BAS(c)
			Export/Domestic CET nest		

Note that $V1TOT(i) = \Sigma_c MAKE(c,i)$ and $SALES(c) = \Sigma_i MAKE(c,i) = DOMSALES(c) + V4BAS(c)$

```
! Excerpt 14 of TABLO input file: !
! Investment demands !

Variable
 (all,c,COM)(all,i,IND)  x2_s(c,i)  # Investment use of imp/dom composite #;
 (all,c,COM)(all,i,IND)  p2_s(c,i)  # Price, investment imp/dom composite #;
 (all,c,COM)(all,s,SRC)(all,i,IND)  a2(c,s,i) # Investment basic tech change #;
Coefficient
 (parameter) (all,c,COM)  SIGMA2(c) # Armington elasticities: investment #;
Read SIGMA2 from file BASEDATA header "2ARM";
Coefficient ! Source Shares in Flows at Purchaser's prices !
 (all,c,COM)(all,i,IND)     V2PUR_S(c,i) # Dom+imp investment purch. value #;
 (all,c,COM)(all,s,SRC)(all,i,IND)  S2(c,s,i) # Investment source shares #;
Zerodivide default 0.5;
Formula
  (all,c,COM)(all,i,IND)     V2PUR_S(c,i)  = sum{s,SRC, V2PUR(c,s,i)};
  (all,c,COM)(all,s,SRC)(all,i,IND)  S2(c,s,i) = V2PUR(c,s,i) / V2PUR_S(c,i);
Zerodivide off;

Equation E_x2  # Source-specific commodity demands #
(all,c,COM)(all,s,SRC)(all,i,IND)
x2(c,s,i)-a2(c,s,i) - x2_s(c,i) = - SIGMA2(c)*[p2(c,s,i)+a2(c,s,i) - p2_s(c,i)];

Equation E_p2_s  # Effective price of commodity composite #
(all,c,COM)(all,i,IND)
p2_s(c,i) = sum{s,SRC, S2(c,s,i)*[p2(c,s,i)+a2(c,s,i)]};
```

```
! Investment top nest !

Variable
 (all,i,IND)  a2tot(i)        # Neutral technical change - investment #;
 (all,i,IND)  p2tot(i)        # Cost of unit of capital #;
 (all,i,IND)  x2tot(i)        # Investment by using industry #;
 (all,c,COM)(all,i,IND) a2_s(c,i) # Tech change, investment imp/dom composite #;
Coefficient (all,i,IND) V2TOT(i)  # Total capital created for industry i #;
Formula     (all,i,IND) V2TOT(i)  = sum{c,COM, V2PUR_S(c,i)};
Equation
 E_x2_s  (all,c,COM)(all,i,IND)  x2_s(c,i) - [a2_s(c,i) + a2tot(i)] = x2tot(i);
 E_p2tot (all,i,IND) p2tot(i)
  = sum{c,COM, (V2PUR_S(c,i)/ID01[V2TOT(i)])*[p2_s(c,i) +a2_s(c,i) +a2tot(i)]};



! Excerpt 15 of TABLO input file: !
! Import/domestic composition of household demands !

Variable
(all,c,COM)(all,s,SRC)  a3(c,s)   # Household basic taste change #;
(all,c,COM)             x3_s(c)   # Household use of imp/dom composite #;
(all,c,COM)             p3_s(c)   # Price, household imp/dom composite #;
Coefficient
 (parameter)(all,c,COM) SIGMA3(c) # Armington elasticities: households #;
Read SIGMA3 from file BASEDATA header "3ARM";

Coefficient ! Source Shares in Flows at Purchaser's prices !
 (all,c,COM)           V3PUR_S(c) # Dom+imp households purch. value #;
 (all,c,COM)(all,s,SRC)   S3(c,s) # Household source shares #;
Zerodivide default 0.5;
Formula
 (all,c,COM)           V3PUR_S(c) = sum{s,SRC, V3PUR(c,s)};
 (all,c,COM)(all,s,SRC)   S3(c,s) = V3PUR(c,s)   / V3PUR_S(c);
Zerodivide off;

Equation E_x3  # Source-specific commodity demands #
(all,c,COM)(all,s,SRC)
x3(c,s)-a3(c,s) = x3_s(c) - SIGMA3(c)*[ p3(c,s)+a3(c,s) - p3_s(c) ];

Equation E_p3_s  # Effective price of commodity composite #
(all,c,COM) p3_s(c) = sum{s,SRC, S3(c,s)*[p3(c,s)+a3(c,s)]};


! Excerpt 16 of TABLO input file: !
! Household demands for composite commodities !
Variable        p3tot # Consumer price index #;
                x3tot # Real household consumption #;
                w3lux # Total nominal supernumerary household expenditure #;
                w3tot # Nominal total household consumption #;
                    q # Number of households #;
              utility # Utility per household #;
 (all,c,COM) x3lux(c) # Household - supernumerary demands #;
 (all,c,COM) x3sub(c) # Household - subsistence demands #;
 (all,c,COM) a3lux(c) # Taste change, supernumerary demands #;
 (all,c,COM) a3sub(c) # Taste change, subsistence demands #;
 (all,c,COM)  a3_s(c) # Taste change, household imp/dom composite #;
Coefficient
                V3TOT # Total purchases by households #;
               FRISCH # Frisch LES 'parameter'= - (total/luxury) #;
 (all,c,COM)    EPS(c) # Household expenditure elasticities #;
 (all,c,COM)   S3_S(c) # Household average budget shares #;
 (all,c,COM) B3LUX(c) # Ratio, (supernumerary expenditure/total expenditure) #;
```

```
 (all,c,COM) S3LUX(c) # Marginal household budget shares #;
Read   FRISCH from file BASEDATA header "P021";
          EPS from file BASEDATA header "XPEL";
Update (change)            FRISCH = FRISCH*[w3tot - w3lux]/100.0;
       (change)(all,c,COM) EPS(c) = EPS(c)*[x3lux(c)-x3_s(c)+w3tot-w3lux]/100.0;
Formula
               V3TOT = sum{c,COM, V3PUR_S(c)};
 (all,c,COM)  S3_S(c) = V3PUR_S(c)/V3TOT;
 (all,c,COM) B3LUX(c) = EPS(c)/ABS[FRISCH]; ! initial sign of Frisch ignored !
 (all,c,COM) S3LUX(c) = EPS(c)*S3_S(c);
Write  S3LUX   to file SUMMARY header "LSHR";
       S3_S    to file SUMMARY header "CSHR";
Equation
 E_x3sub # Subsistence demand for composite commodities #
  (all,c,COM) x3sub(c) = q + a3sub(c);

 E_x3lux # Luxury demand for composite commodities #
  (all,c,COM) x3lux(c) + p3_s(c) = w3lux + a3lux(c);

 E_x3_s  # Total household demand for composite commodities #
  (all,c,COM) x3_s(c) = B3LUX(c)*x3lux(c) + [1-B3LUX(c)]*x3sub(c);

 E_utility # Change in utility disregarding taste change terms #
  utility + q = sum{c,COM, S3LUX(c)*x3lux(c)};

 E_a3lux # Default setting for luxury taste shifter #
  (all,c,COM) a3lux(c) = a3sub(c) - sum{k,COM, S3LUX(k)*a3sub(k)};

 E_a3sub # Default setting for subsistence taste shifter #
  (all,c,COM) a3sub(c) = a3_s(c) - sum{k,COM, S3_S(k)*a3_s(k)};

 E_x3tot # Real consumption #
  x3tot = sum{c,COM, sum{s,SRC, [V3PUR(c,s)/V3TOT]*x3(c,s)}};

 E_p3tot # Consumer price index #
  p3tot = sum{c,COM, sum{s,SRC, [V3PUR(c,s)/V3TOT]*p3(c,s)}};
 E_w3tot # Household budget constraint: determines w3lux #
  w3tot = x3tot + p3tot;




! Excerpt 17 of TABLO input file: !
! Export demands !

Coefficient
 (parameter)(all,c,COM) IsIndivExp(c) # >0.5 For individual export commodities#;
Read IsIndivExp from file BASEDATA header "ITEX";
! This way of defining a set facilitates aggregation of the data base !
Set TRADEXP # Individual export commodities # = (all,c,COM: IsIndivExp(c)>0.5);
Write (Set) TRADEXP  to file SUMMARY header "TEXP";

Variable
              phi     # Exchange rate, local currency/$world #;
 (all,c,COM)  f4p(c)  # Price (upward) shift in export demand schedule #;
 (all,c,COM)  f4q(c)  # Quantity (right) shift in export demands #;
Coefficient (parameter)(all,c,COM) EXP_ELAST(c)
  # Export demand elasticities: typical value -5.0 #;
Read EXP_ELAST from file BASEDATA header "P018";
Equation E_x4A  # Individual export demand functions #
(all,c,TRADEXP) x4(c) - f4q(c) = -ABS[EXP_ELAST(c)]*[p4(c) - phi - f4p(c)];
! note: ABS function above fixes common mistake: positive EXP_ELAST values !
```

```
Set NTRADEXP # Collective Export Commodities # = COM - TRADEXP;
Write (Set) NTRADEXP  to file SUMMARY header "NTXP";
Variable
 x4_ntrad     # Quantity, collective export aggregate #;
 f4p_ntrad    # Upward demand shift, collective export aggregate #;
 f4q_ntrad    # Right demand shift, collective export aggregate #;
 p4_ntrad     # Price, collective export aggregate #;

Coefficient V4NTRADEXP # Total collective export earnings #;
Formula     V4NTRADEXP = sum{c,NTRADEXP, V4PUR(c)};

Equation E_X4B  # Collective export demand functions #
 (all,c,NTRADEXP) x4(c) - f4q(c) = x4_ntrad;

Equation E_p4_ntrad  # Average price of collective exports #
     [TINY+V4NTRADEXP]*p4_ntrad = sum{c,NTRADEXP, V4PUR(c)*p4(c)};

Coefficient (parameter) EXP_ELAST_NT # Collective export demand elasticity #;
Read EXP_ELAST_NT from file BASEDATA header "EXNT";
Equation E_x4_ntrad # Demand for collective export aggregate #
        x4_ntrad - f4q_ntrad = -ABS[EXP_ELAST_NT]*[p4_ntrad - phi - f4p_ntrad];



! Excerpt 18 of TABLO input file: !
! Government and inventory demands !
Variable
 f5tot  # Overall shift term for government demands #;
 f5tot2 # Ratio between f5tot and x3tot #;
 (all,c,COM)(all,s,SRC) f5(c,s) # Government demand shift #;
 (change) (all,c,COM)(all,s,SRC)  fx6(c,s)    # Shifter on rule for stocks #;
Equation
 E_x5 # Government demands # (all,c,COM)(all,s,SRC) x5(c,s) = f5(c,s) + f5tot;
 E_f5tot # Overall government demands shift # f5tot = x3tot + f5tot2;

Coefficient (all,c,COM)(all,s,SRC) LEVP0(c,s) # Levels basic prices #;
Formula (initial) (all,c,COM)(all,s,SRC) LEVP0(c,s) = 1; ! arbitrary setting !
Update   (all,c,COM)(all,s,SRC) LEVP0(c,s) = p0(c,s);
Equation
 E_delx6 # Stocks follow domestic output # (all,c,COM)(all,s,SRC)
  100*LEVP0(c,s)*delx6(c,s) = V6BAS(c,s)*x0com(c) + fx6(c,s);
 E_delV6 # Update formula for stocks #     (all,c,COM)(all,s,SRC)
  delV6(c,s) = 0.01*V6BAS(c,s)*p0(c,s) + LEVP0(c,s)*delx6(c,s);



! Excerpt 19 of TABLO input file: !
! Margin demands !

Variable
 (all,c,COM)(all,s,SRC)(all,i,IND)(all,m,MAR)
                           a1mar(c,s,i,m)  # Intermediate margin tech change #;
 (all,c,COM)(all,s,SRC)(all,i,IND)(all,m,MAR)
                           a2mar(c,s,i,m)  # Investment margin tech change #;
 (all,c,COM)(all,s,SRC)(all,m,MAR) a3mar(c,s,m) # Household margin tech change#;
 (all,c,COM)(all,m,MAR)            a4mar(c,m)   # Export margin tech change #;
 (all,c,COM)(all,s,SRC)(all,m,MAR) a5mar(c,s,m) # Governmnt margin tech change#;

Equation
 E_x1mar  # Margins to producers #  (all,c,COM)(all,s,SRC)(all,i,IND)(all,m,MAR)
  x1mar(c,s,i,m) = x1(c,s,i) + a1mar(c,s,i,m);
 E_x2mar  # Margins to investment # (all,c,COM)(all,s,SRC)(all,i,IND)(all,m,MAR)
```

```
  x2mar(c,s,i,m) = x2(c,s,i) + a2mar(c,s,i,m);
 E_x3mar  # Margins to households # (all,c,COM)(all,s,SRC)(all,m,MAR)
  x3mar(c,s,m)   = x3(c,s) + a3mar(c,s,m);
 E_x4mar  # Margins to exports #    (all,c,COM)(all,m,MAR)
  x4mar(c,m)     = x4(c) + a4mar(c,m);
 E_x5mar  # Margins to government # (all,c,COM)(all,s,SRC)(all,m,MAR)
  x5mar(c,s,m)   = x5(c,s) + a5mar(c,s,m);



! Excerpt 20 of TABLO input file: !
! Sales Aggregates !

Coefficient (all,c,COM) MARSALES(c) # Total usage for margins purposes #;
Formula
 (all,n,NONMAR) MARSALES(n) = 0.0;
 (all,m,MAR)    MARSALES(m) = sum{c,COM, V4MAR(c,m) +
                                 sum{s,SRC, V3MAR(c,s,m) + V5MAR(c,s,m) +
                                  sum{i,IND, V1MAR(c,s,i,m) + V2MAR(c,s,i,m)}}};

Set DEST # Sale Categories #
(Interm, Invest, HouseH, Export, GovGE, Stocks, Margins);

Coefficient (all,c,COM)(all,s,SRC)(all,d,DEST) SALE(c,s,d) # Sales aggregates #;
Formula
 (all,c,COM)(all,s,SRC) SALE(c,s,"Interm")      = sum{i,IND, V1BAS(c,s,i)};
 (all,c,COM)(all,s,SRC) SALE(c,s,"Invest")      = sum{i,IND, V2BAS(c,s,i)};
 (all,c,COM)(all,s,SRC) SALE(c,s,"HouseH")      = V3BAS(c,s);
 (all,c,COM)            SALE(c,"dom","Export")  = V4BAS(c);
 (all,c,COM)            SALE(c,"imp","Export")  = 0;
 (all,c,COM)(all,s,SRC) SALE(c,s,"GovGE")       = V5BAS(c,s);
 (all,c,COM)(all,s,SRC) SALE(c,s,"Stocks")      = V6BAS(c,s);
 (all,c,COM)            SALE(c,"dom","Margins") = MARSALES(c);
 (all,c,COM)            SALE(c,"imp","Margins") = 0;
Write SALE to file SUMMARY header "SALE";

Coefficient (all,c,COM) V0IMP(c) # Total basic-value imports of good c #;
Formula     (all,c,COM) V0IMP(c) = sum{d,DEST, SALE(c,"imp",d)};
Coefficient (all,c,COM) SALES(c) # Total sales of domestic commodities #;
Formula     (all,c,COM) SALES(c) = sum{d,DEST, SALE(c,"dom",d)};
Coefficient (all,c,COM) DOMSALES(c) # Total sales to local market #;
Formula     (all,c,COM) DOMSALES(c) = SALES(c) - V4BAS(c);



! Excerpt 21 of TABLO input file: !
! Market clearing equations !

Variable (change)
 (all,c,COM)(all,s,SRC)(all,d,DEST) delSale(c,s,d) # Sales aggregates #;
Equation
E_delSaleA (all,c,COM)(all,s,SRC) delSale(c,s,"Interm") =
                  0.01*sum{i,IND,V1BAS(c,s,i)*x1(c,s,i)};
E_delSaleB (all,c,COM)(all,s,SRC) delSale(c,s,"Invest") =
                  0.01*sum{i,IND,V2BAS(c,s,i)*x2(c,s,i)};
E_delSaleC (all,c,COM)(all,s,SRC) delSale(c,s,"HouseH")=0.01*V3BAS(c,s)*x3(c,s);
E_delSaleD (all,c,COM)            delSale(c,"dom","Export")=0.01*V4BAS(c)*x4(c);
E_delSaleE (all,c,COM)            delSale(c,"imp","Export")=   0;
E_delSaleF (all,c,COM)(all,s,SRC) delSale(c,s,"GovGE") =0.01*V5BAS(c,s)*x5(c,s);
E_delSaleG (all,c,COM)(all,s,SRC) delSale(c,s,"Stocks") = LEVP0(c,s)*delx6(c,s);
E_delSaleH (all,m,MAR)            delSale(m,"dom","Margins") = 0.01*
  sum{c,COM, V4MAR(c,m)*x4mar(c,m)   ! note nesting of sum parentheses !
```

```
 + sum{s,SRC, V3MAR(c,s,m)*x3mar(c,s,m) + V5MAR(c,s,m)*x5mar(c,s,m)
  + sum{i,IND, V1MAR(c,s,i,m)*x1mar(c,s,i,m) + V2MAR(c,s,i,m)*x2mar(c,s,i,m)}}};
E_delSaleI (all,n,NONMAR)         delSale(n,"dom","Margins") = 0;
E_delSaleJ (all,c,COM)            delSale(c,"imp","Margins") = 0;

Set LOCUSER # Non-export users #(Interm, Invest, HouseH, GovGE, Stocks,Margins);
Subset LOCUSER is subset of DEST;

Equation E_p0A # Supply = Demand for domestic commodities #
(all,c,COM) 0.01*[TINY+DOMSALES(c)]*x0dom(c) =sum{u,LOCUSER,delSale(c,"dom",u)};

Variable (all,c,COM)  x0imp(c)        # Total supplies of imported goods #;
Equation E_x0imp # Import volumes #
 (all,c,COM) 0.01*[TINY+V0IMP(c)]*x0imp(c) = sum{u,LOCUSER,delSale(c,"imp",u)};


! Excerpt 22 of TABLO input file: !
! Purchasers prices !

Variable  ! Powers of Commodity Taxes on Basic Flows !
(all,c,COM)(all,s,SRC)(all,i,IND)(all,t,ALLCTAX) t1(c,s,i,t) # Power of tax on
intermediate #;
 (all,c,COM)(all,s,SRC)(all,i,IND)(all,t,ALLCTAX) t2(c,s,i,t) # Power of tax on
investment #;
 (all,c,COM)(all,s,SRC)(all,t,ALLCTAX)           t3(c,s,t)   # Power of tax on
household #;
 (all,c,COM)(all,t,ALLCTAX)                      t4(c,t)     # Power of tax on export
#;
 (all,c,COM)(all,s,SRC)(all,t,ALLCTAX)           t5(c,s,t)   # Power of tax on
government #;

Equation E_p1 # Purchasers prices - producers #
 (all,c,COM)(all,s,SRC)(all,i,IND)
  [V1PUR(c,s,i)+TINY]*[x1(c,s,i) + p1(c,s,i)] =
  V1BAS(c,s,i) * [x1(c,s,i) + p0(c,s)] + 100* Sum{t,ALLCTAX, delV1TAX(c,s,i,t)}
    +  Sum{m,MAR, V1MAR(c,s,i,m)*[x1(c,s,i) + p0dom(m) + a1mar(c,s,i,m)] };

Equation E_p2 # Purchasers prices - producers #
 (all,c,COM)(all,s,SRC)(all,i,IND)
  [V2PUR(c,s,i)+TINY]*[x2(c,s,i) + p2(c,s,i)] =
  V2BAS(c,s,i) * [x2(c,s,i) + p0(c,s)] + 100* Sum{t,ALLCTAX, delV2TAX(c,s,i,t)}
    +  Sum{m,MAR, V2MAR(c,s,i,m)*[x2(c,s,i) + p0dom(m) + a2mar(c,s,i,m)] };

Equation E_p3 # Purchasers prices - producers #
 (all,c,COM)(all,s,SRC)
  [V3PUR(c,s)+TINY]*[x3(c,s) + p3(c,s)] =
    V3BAS(c,s) * [x3(c,s) + p0(c,s)] + 100* Sum{t,ALLCTAX, delV3TAX(c,s,t)}
    +  Sum{m,MAR, V3MAR(c,s,m)*[x3(c,s) + p0dom(m) + a3mar(c,s,m)] };

Equation E_p4 # Purchasers prices - producers #
 (all,c,COM)
  [V4PUR(c)+TINY]*[x4(c) + p4(c)] =
    V4BAS(c) * [x4(c) + pe(c)] + 100* Sum{t,ALLCTAX, delV4TAX(c,t)}
    +  Sum{m,MAR, V4MAR(c,m)*[x4(c) + p0dom(m) + a4mar(c,m)] };

Equation E_p5 # Purchasers prices - producers #
 (all,c,COM)(all,s,SRC)
  [V5PUR(c,s)+TINY]*[x5(c,s) + p5(c,s)] =
    V5BAS(c,s) * [x5(c,s) + p0(c,s)] + 100* Sum{t,ALLCTAX, delV5TAX(c,s,t)}
    +  Sum{m,MAR, V5MAR(c,s,m)*[x5(c,s) + p0dom(m) + a5mar(c,s,m)] };
```

```
! Excerpt 23 of TABLO input file: !
! Tax rate equations !

Variable
f1tax_csi    # Uniform % change in powers of taxes on intermediate usage #;
f2tax_csi    # Uniform % change in powers of taxes on investment #;
f3tax_cs     # Uniform % change in powers of taxes on household usage #;
f4tax_ntrad  # Uniform % change in powers of taxes on nontradtnl exports #;
f4tax_trad   # Uniform % change in powers of taxes on tradtnl exports #;
f5tax_cs     # Uniform % change in powers of taxes on government usage #;
(all,c,COM)             f0tax_s(c) # General comm tax shifter, by commodity #;
(all,t,ALLCTAX)         f0tax_cu(t) # General comm tax shifter, by tax type #;
(all,c,COM)(all,t,ALLCTAX) f3tax_u(c,t) # shifters for household taxes #;

(All,c,COM)(All,s,SRC)(All,i,IND)(all,t,ALLCTAX)
                ft1(c,s,i,t) # Tax power shifter, intermediate #;
(All,c,COM)(All,s,SRC)(All,i,IND)(all,t,ALLCTAX)
                ft2(c,s,i,t) # Tax power shifter, investment #;
(All,c,COM)(All,s,SRC)(all,t,ALLCTAX)
                ft3(c,s,t) # Tax power shifter, household #;
(All,c,COM)(all,t,ALLCTAX)
                ft4(c,t) # Tax power shifter, export #;
(All,c,COM)(All,s,SRC)(all,t,ALLCTAX)
                ft5(c,s,t) # Tax power shifter, government #;

Equation
E_t1  # Power of tax on sales to intermediate #
  (all,c,COM)(all,s,SRC)(all,i,IND)(all,t,ALLCTAX) t1(c,s,i,t)  =  ft1(c,s,i,t) +
f0tax_s(c)  +f0tax_cu(t) + f1tax_csi;

E_t2  # Power of tax on sales to investment #
  (all,c,COM)(all,s,SRC)(all,i,IND)(all,t,ALLCTAX)  t2(c,s,i,t)  = ft2(c,s,i,t) +
f0tax_s(c) +f0tax_cu(t)+ f2tax_csi;

E_t3  # Power of tax on sales to households #
  (all,c,COM)(all,s,SRC)(all,t,ALLCTAX)             t3(c,s,t)    = ft3(c,s,t) +
f3tax_u(c,t) + f0tax_s(c) +f0tax_cu(t)+ f3tax_cs;

E_t4A # Power of tax on sales to individual exports #
  (all,c,TRADEXP)(all,t,ALLCTAX)                    t4(c,t)      = ft4(c,t) + f0tax_s(c)
+f0tax_cu(t)+ f4tax_trad;

E_t4B # Power of tax on sales to collective exports #
  (all,c,NTRADEXP)(all,t,ALLCTAX)                   t4(c,t)      = ft4(c,t) + f0tax_s(c)
+f0tax_cu(t)+ f4tax_ntrad;

E_t5  # Power of tax on sales to government #
  (all,c,COM)(all,s,SRC)(all,t,ALLCTAX)             t5(c,s,t)    = ft5(c,s,t) +
f0tax_s(c) +f0tax_cu(t)+ f5tax_cs;
```

```
! Excerpt 24 of TABLO input file: !
! Update formulae for commodity taxes !
Equation
 E_delV1TAXA (all,c,COM)(all,s,SRC)(all,i,IND)(all,t,NOTVAT)
   delV1TAX(c,s,i,t) = 0.01*V1TAX(c,s,i,t)* [x1(c,s,i) + p0(c,s)] +
     0.01*[V1BAS(c,s,i)+ V1TAX(c,s,i,t)]*t1(c,s,i,t);

 E_delV2TAXA (all,c,COM)(all,s,SRC)(all,i,IND)(all,t,NOTVAT)
   delV2TAX(c,s,i,t)=  0.01*V2TAX(c,s,i,t)* [x2(c,s,i) + p0(c,s)] +
     0.01*[V2BAS(c,s,i)+V2TAX(c,s,i,t)]*t2(c,s,i,t);

 E_delV3TAXA (all,c,COM)(all,s,SRC)(all,t,NOTVAT)
   delV3TAX(c,s,t)  = 0.01*V3TAX(c,s,t)* [x3(c,s) + p0(c,s)] +
     0.01*[V3BAS(c,s)+V3TAX(c,s,t)]*t3(c,s,t);

 E_delV4TAXA (all,c,COM)(all,t,NOTVAT)
   delV4TAX(c,t) = 0.01*V4TAX(c,t)* [x4(c) + pe(c)] +
     0.01*[V4BAS(c)+V4TAX(c,t)]*t4(c,t);

 E_delV5TAXA (all,c,COM)(all,s,SRC)(all,t,NOTVAT)
   delV5TAX(c,s,t) = 0.01*V5TAX(c,s,t)*[x5(c,s) + p0(c,s)] +
     0.01*[V5BAS(c,s)+V5TAX(c,s,t)]*t5(c,s,t);

Equation
 E_delV1TAXB (all,c,COM)(all,s,SRC)(all,i,IND)
   100* delV1TAX(c,s,i,"VAT") = V1TAX(c,s,i,"VAT")* [1/[TINY + V1BAS(c,s,i) +
Sum{m,MAR,V1MAR(c,s,i,m)}]]
    * [V1BAS(c,s,i)*(x1(c,s,i) + p0(c,s)) + Sum{m,MAR, V1MAR(c,s,i,m) * (x1mar(c,s,i,m)
+p0dom(m))}]
    + [V1BAS(c,s,i)+ Sum{m,MAR,V1MAR(c,s,i,m)} + V1TAX(c,s,i,"VAT")]*t1(c,s,i,"VAT");

 E_delV2TAXB (all,c,COM)(all,s,SRC)(all,i,IND)
   100* delV2TAX(c,s,i,"VAT") = V2TAX(c,s,i,"VAT")* [1/[TINY + V2BAS(c,s,i) +
Sum{m,MAR,V2MAR(c,s,i,m)}]]
    * [V2BAS(c,s,i)*(x2(c,s,i) + p0(c,s)) + Sum{m,MAR, V2MAR(c,s,i,m) * (x2mar(c,s,i,m)
+p0dom(m))}]
    + [V2BAS(c,s,i)+ Sum{m,MAR,V2MAR(c,s,i,m)} + V2TAX(c,s,i,"VAT")]*t2(c,s,i,"VAT");

 E_delV3TAXB (all,c,COM)(all,s,SRC)
      100* delV3TAX(c,s,"VAT") = V3TAX(c,s,"VAT")* [1/[TINY + V3BAS(c,s) +
Sum{m,MAR,V3MAR(c,s,m)}]]
    * [V3BAS(c,s)*(x3(c,s) + p0(c,s)) + Sum{m,MAR, V3MAR(c,s,m) * (x3mar(c,s,m)
+p0dom(m))}]
    + [V3BAS(c,s)+ Sum{m,MAR,V3MAR(c,s,m)} + V3TAX(c,s,"VAT")]*t3(c,s,"VAT");

 E_delV4TAXB (all,c,COM)
   100* delV4TAX(c,"VAT") = V4TAX(c,"VAT")* [1/[TINY + V4BAS(c) +
Sum{m,MAR,V4MAR(c,m)}]]
    * [V4BAS(c)*(x4(c) + pe(c)) + Sum{m,MAR, V4MAR(c,m) * (x4mar(c,m) +p0dom(m))}]
    + [V4BAS(c)+ Sum{m,MAR,V4MAR(c,m)} + V4TAX(c,"VAT")]*t4(c,"VAT");

E_delV5TAXB (all,c,COM)(all,s,SRC)
   100* delV5TAX(c,s,"VAT") = V5TAX(c,s,"VAT")* [1/[TINY + V5BAS(c,s) +
Sum{m,MAR,V5MAR(c,s,m)}]]
    * [V5BAS(c,s)*(x5(c,s) + p0(c,s)) + Sum{m,MAR, V5MAR(c,s,m) * (x5mar(c,s,m)
+p0dom(m))}]
    + [V5BAS(c,s)+ Sum{m,MAR,V5MAR(c,s,m)} + V5TAX(c,s,"VAT")]*t5(c,s,"VAT");
```

```
! Excerpt 25 of TABLO input file: !
! Import prices and tariff revenue !

Variable
(all,c,COM)  pf0cif(c) # C.I.F. foreign currency import prices #;
(all,c,COM)  t0imp(c)  # Power of tariff #;

Equation E_p0B # Zero pure profits in importing #
(all,c,COM) p0(c,"imp") = pf0cif(c) + phi + t0imp(c);

Equation E_delV0TAR   (all,c,COM)
 delV0TAR(c) = 0.01*V0TAR(c)*[x0imp(c)+pf0cif(c)+phi] + 0.01*V0IMP(c)*t0imp(c);




! Excerpt 26 of TABLO input file: !
! Indirect tax revenue aggregates !

 V1TAX_CSI  # Total intermediate tax revenue #;
 V2TAX_CSI  # Total investment tax revenue #;
 V3TAX_CS   # Total households tax revenue #;
 V4TAX_C    # Total export tax revenue #;
 V5TAX_CS   # Total government tax revenue #;
 V0TAR_C    # Total tariff revenue #;Formula
 V1TAX_CSI  = sum{c,COM, sum{s,SRC, sum{i,IND, sum{t,ALLCTAX,      V1TAX(c,s,i,t)}}}};
 V2TAX_CSI  = sum{c,COM, sum{s,SRC, sum{i,IND, sum{t,ALLCTAX,V2TAX(c,s,i,t)}}}};
 V3TAX_CS   = sum{c,COM, sum{s,SRC,   sum{t,ALLCTAX,V3TAX(c,s,t)}}};
 V4TAX_C    = sum{c,COM,              sum{t,ALLCTAX,V4TAX(c,t)}};
 V5TAX_CS   = sum{c,COM, sum{s,SRC,   sum{t,ALLCTAX,V5TAX(c,s,t)}}};
 V0TAR_C    = sum{c,COM, V0TAR(c)};Variable
(change) delV1tax_csi # Aggregate revenue from indirect taxes on intermediate ;
(change) delV2tax_csi # Aggregate revenue from indirect taxes on investment #;
(change) delV3tax_cs  # Aggregate revenue from indirect taxes on households #;
(change) delV4tax_c   # Aggregate revenue from indirect taxes on export #;
(change) delV5tax_cs  # Aggregate revenue from indirect taxes on government #;
(change) delV0tar_c   # Aggregate tariff revenue #;

Equation
E_delV1tax_csi delV1tax_csi = sum{c,COM,sum{s,SRC,sum{i,IND,sum{t,ALLCTAX,
delV1TAX(c,s,i,t)}}}};
E_delV2tax_csi delV2tax_csi = sum{c,COM,sum{s,SRC,sum{i,IND,sum{t,ALLCTAX,
delV2TAX(c,s,i,t)}}}};
E_delV3tax_cs  delV3tax_cs  = sum{c,COM,sum{s,SRC,sum{t,ALLCTAX, delV3TAX(c,s,t)}}};
E_delV4tax_c   delV4tax_c   = sum{c,COM,sum{t,ALLCTAX, delV4TAX(c,t)}};
E_delV5tax_cs  delV5tax_cs  = sum{c,COM,sum{s,SRC,sum{t,ALLCTAX, delV5TAX(c,s,t)}}};
E_delV0tar_c   delV0tar_c   = sum{c,COM, delV0TAR(c)};

Coefficient
(all,t,ALLCTAX) V1TAXc_CSI(t)  # Total intermediate tax revenue #;
(all,t,ALLCTAX) V2TAXc_CSI(t)  # Total investment tax revenue #;
(all,t,ALLCTAX) V3TAXc_CS(t)   # Total households tax revenue #;
(all,t,ALLCTAX) V4TAXc_C(t)    # Total export tax revenue #;
(all,t,ALLCTAX) V5TAXc_CS(t)   # Total government tax revenue #;

Formula
(all,t,ALLCTAX) V1TAXc_CSI(t)   = sum{c,COM, sum{s,SRC, sum{i,IND,  V1TAX(c,s,i,t)}}};
(all,t,ALLCTAX) V2TAXc_CSI(t)   = sum{c,COM, sum{s,SRC, sum{i,IND, V2TAX(c,s,i,t)}}};
(all,t,ALLCTAX) V3TAXc_CS(t)    = sum{c,COM, sum{s,SRC,  V3TAX(c,s,t)}};
(all,t,ALLCTAX) V4TAXc_C(t)     = sum{c,COM, V4TAX(c,t)};
(all,t,ALLCTAX) V5TAXc_CS(t)    = sum{c,COM, sum{s,SRC,  V5TAX(c,s,t)}};
```

```
Variable
(change) (all,t,ALLCTAX) delV1TAXc_csi(t) # Aggregate revenue from indirect taxes on
intermediate #;
(change) (all,t,ALLCTAX) delV2TAXc_csi(t) # Aggregate revenue from indirect taxes on
investment #;
(change) (all,t,ALLCTAX) delV3TAXc_cs(t)  # Aggregate revenue from indirect taxes on
households #;
(change) (all,t,ALLCTAX) delV4TAXc_c(t)   # Aggregate revenue from indirect taxes on
export #;
(change) (all,t,ALLCTAX) delV5TAXc_cs(t)  # Aggregate revenue from indirect taxes on
government #;

Equation
E_delV1TAXc_csi (all,t,ALLCTAX) delV1TAXc_csi(t)  = sum{c,COM,sum{s,SRC,sum{i,IND,
delV1TAX(c,s,i,t)}}};
E_delV2TAXc_csi (all,t,ALLCTAX) delV2TAXc_csi(t) = sum{c,COM,sum{s,SRC,sum{i,IND,
delV2TAX(c,s,i,t)}}};
E_delV3TAXc_cs  (all,t,ALLCTAX) delV3TAXc_cs(t) =
sum{c,COM,sum{s,SRC, delV3TAX(c,s,t)}};
E_delV4TAXc_c   (all,t,ALLCTAX) delV4TAXc_c(t)  =                          sum{c,COM,
delV4TAX(c,t)};
E_delV5TAXc_cs  (all,t,ALLCTAX) delV5TAXc_cs(t)  =                 sum{c,COM,sum{s,SRC,
delV5TAX(c,s,t)}};


! Excerpt 27 of TABLO input file: !
! Factor incomes and GDP !

Coefficient
  V1CAP_I   # Total payments to capital #;
  V1LAB_IO  # Total payments to labour #;
  V1LND_I   # Total payments to land #;
  V1PTX_I   # Total production tax/subsidy #;
  V1OCT_I   # Total other cost ticket payments #;
  V1PRIM_I  # Total primary factor payments#;
  V0GDPINC  # Nominal GDP from income side #;
  V0TAX_CSI # Total indirect tax revenue #;
 (all,t,ALLCTAX)  V0TAXC_CSI(t) # Total commodity tax revenue, by tax type #;
 (all,p,ALLPTAX)  V1PTXP_I(p)   # Total production tax revenue, by tax type #;

Formula
  V1CAP_I   = sum{i,IND, V1CAP(i)};
  V1LAB_IO  = sum{i,IND, V1LAB_O(i)};
  V1LND_I   = sum{i,IND, V1LND(i)};
  V1PTX_I   = sum{i,IND,sum{p,ALLPTAX, V1PTX(i,p)}};
  V1OCT_I   = sum{i,IND, V1OCT(i)};
  V1PRIM_I  = V1LAB_IO + V1CAP_I + V1LND_I;
  V0TAX_CSI  = V1TAX_CSI + V2TAX_CSI + V3TAX_CS + V4TAX_C + V5TAX_CS
               + V0TAR_C + V1OCT_I + V1PTX_I;
  V0GDPINC  = V1PRIM_I + V0TAX_CSI;
 (all,t,ALLCTAX)  V0TAXC_CSI(t) = V1TAXc_CSI(t) + V2TAXc_CSI(t) + V3TAXc_CS(t) +
V4TAXc_C(t)  + V5TAXc_CS(t);
 (all,p,ALLPTAX)   V1PTXP_I(p) = Sum{i,IND, V1PTX(i,p)};
```

```
Variable
 w1lab_io  # Aggregate payments to labour #;
 w1cap_i   # Aggregate payments to capital #;
 w1lnd_i   # Aggregate payments to land #;
 w1prim_i  # Aggregate primary factor payments #;
 w1oct_i   # Aggregate "other cost" ticket payments #;
 (change) delV1PTX_i # Ordinary change in all-industry production tax revenue #;
 (change) delV0tax_csi # Aggregate revenue from all indirect taxes #;
 w0tax_csi # Aggregate revenue from all indirect taxes #;
 w0gdpinc  # Nominal GDP from income side #;
(all,t,ALLCTAX) w0taxc_csi(t) # Aggregate revenue from commodity taxes by tax type #;
 (change) (all,t,ALLCTAX)  delV0taxc_csi(t) # Aggregate revenue from commodity taxes #;

Equation
 E_w1lab_io     V1LAB_IO*w1lab_io =
                    sum{i,IND,sum{o,OCC, V1LAB(i,o)*[x1lab(i,o)+p1lab(i,o)]}};
 E_w1cap_i      V1CAP_I*w1cap_i = sum{i,IND, V1CAP(i)*[x1cap(i)+p1cap(i)]};
 E_w1lnd_i          ID01[V1LND_I]*w1lnd_i = sum{i,IND, V1LND(i)*[x1lnd(i)+p1lnd(i)]};
 E_w1prim_i     V1PRIM_I*w1prim_i=V1LAB_IO*w1lab_io+V1CAP_I*w1cap_i+V1LND_I*w1lnd_i;
 E_w1oct_i          ID01[V1OCT_I]*w1oct_i = sum{i,IND, V1OCT(i)*[x1oct(i)+p1oct(i)]};
 E_delV1PTX_i   delV1PTX_i = sum{i,IND,sum{p,ALLPTAX, delV1PTX(i,p)}};
 E_delV0tax_csi  delV0tax_csi = delV1tax_csi + delV2tax_csi + delV3tax_cs + delV4tax_c
                    + delV5tax_cs  + delV0tar_c + delV1PTX_i + 0.01*V1OCT_I*w1oct_i;
 E_w0tax_csi    V0TAX_CSI*w0tax_csi = 100*delV0tax_csi;
 E_w0gdpinc     V0GDPINC*w0gdpinc = V1PRIM_I*w1prim_i + 100*delV0tax_csi;
 E_w0taxc_csi   (all,t,ALLCTAX) [TINY+V0TAXC_CSI(t)]*w0taxc_csi(t) =
100*delV0taxc_csi(t);
 E_delV0taxc_csi (all,t,ALLCTAX) delV0taxc_csi(t) =
    delV1taxc_csi(t) + delV2taxc_csi(t) + delV3taxc_cs(t) + delV4taxc_c(t) +
delV5taxc_cs(t);


! Excerpt 28 of TABLO input file: !
! GDP expenditure aggregates !

Coefficient ! Expenditure Aggregates at Purchaser's Prices !
 (all,c,COM) V0CIF(c) # Total ex-duty imports of good c #;
 V0CIF_C  # Total local currency import costs, excluding tariffs #;
 V0IMP_C  # Total basic-value imports (includes tariffs) #;
 V2TOT_I  # Total investment usage #;
 V4TOT    # Total export earnings #;
 V5TOT    # Total value of government demands #;
 V6TOT    # Total value of inventories #;
 V0GNE    # GNE from expenditure side #;
 V0GDPEXP # GDP from expenditure side #;
Formula
 (all,c,COM) V0CIF(c) = V0IMP(c) - V0TAR(c);
 V0CIF_C   = sum{c,COM, V0CIF(c)};
 V0IMP_C   = sum{c,COM, V0IMP(c)};
 V2TOT_I   = sum{i,IND, V2TOT(i)};
 V4TOT     = sum{c,COM, V4PUR(c)};
 V5TOT     = sum{c,COM, sum{s,SRC, V5PUR(c,s)}};
 V6TOT     = sum{c,COM, sum{s,SRC, V6BAS(c,s)}};
 V0GNE     = V3TOT + V2TOT_I + V5TOT + V6TOT;
 V0GDPEXP  = V0GNE + V4TOT - V0CIF_C;

Variable
 x2tot_i    # Aggregate real investment expenditure #;
 p2tot_i    # Aggregate investment price index #;
 w2tot_i    # Aggregate nominal investment #;
```

```
Equation
 E_x2tot_i  V2TOT_I*x2tot_i = sum{i,IND, V2TOT(i)*x2tot(i)};
 E_p2tot_i  V2TOT_I*p2tot_i = sum{i,IND, V2TOT(i)*p2tot(i)};
 E_w2tot_i  w2tot_i = x2tot_i + p2tot_i;

Variable
 x4tot      # Export volume index #;
 p4tot      # Exports price index, local currency #;
 w4tot      # Local currency border value of exports #;
Equation
 E_x4tot    V4TOT*x4tot = sum{c,COM, V4PUR(c)*x4(c)};
 E_p4tot    V4TOT*p4tot = sum{c,COM, V4PUR(c)*p4(c)};
 E_w4tot    w4tot = x4tot + p4tot;

Variable
 x5tot      # Aggregate real government demands #;
 p5tot      # Government price index #;
 w5tot      # Aggregate nominal value of government demands #;
Equation
 E_x5tot    V5TOT*x5tot = sum{c,COM, sum{s,SRC, V5PUR(c,s)*x5(c,s)}};
 E_p5tot    V5TOT*p5tot = sum{c,COM, sum{s,SRC, V5PUR(c,s)*p5(c,s)}};
 E_w5tot    w5tot = x5tot + p5tot;

Variable
 x6tot      # Aggregate real inventories #;
 p6tot      # Inventories price index #;
 w6tot      # Aggregate nominal value of inventories #;
Equation
 E_x6tot    [TINY+V6TOT]*x6tot =100*sum{c,COM,sum{s,SRC,LEVP0(c,s)*delx6(c,s)}};
 E_p6tot    [TINY+V6TOT]*p6tot = sum{c,COM, sum{s,SRC, V6BAS(c,s)*p0(c,s)}};
 E_w6tot    w6tot = x6tot + p6tot;

Variable
 x0cif_c    # Import volume index, C.I.F. weights #;
 p0cif_c    # Imports price index, C.I.F., local currency #;
 w0cif_c    # C.I.F. local currency value of imports #;
Equation
 E_x0cif_c  V0CIF_C*x0cif_c = sum{c,COM, V0CIF(c)*x0imp(c)};
 E_p0cif_c  V0CIF_C*p0cif_c = sum{c,COM, V0CIF(c)*[phi+pf0cif(c)]};
 E_w0cif_c  w0cif_c = x0cif_c + p0cif_c;

Variable  !section added Oct 2002!
 x0gne   # Real GNE from expenditure side #;
 p0gne   # GNE price index, expenditure side #;
 w0gne   # Nominal GNE from expenditure side #;
Equation
 E_x0gne V0GNE*x0gne = V3TOT*x3tot + V2TOT_I*x2tot_i + V5TOT*x5tot +V6TOT*x6tot;
 E_p0gne V0GNE*p0gne = V3TOT*p3tot + V2TOT_I*p2tot_i + V5TOT*p5tot +V6TOT*p6tot;
 E_w0gne w0gne = x0gne + p0gne;

Variable
 x0gdpexp   # Real GDP from expenditure side #;
 p0gdpexp   # GDP price index, expenditure side #;
 w0gdpexp   # Nominal GDP from expenditure side #;
Equation
 E_x0gdpexp x0gdpexp = [1/V0GDPEXP]*[V3TOT*x3tot + V2TOT_I*x2tot_i + V5TOT*x5tot
             + V6TOT*x6tot + V4TOT*x4tot - V0CIF_C*x0cif_c];
 E_p0gdpexp p0gdpexp = [1/V0GDPEXP]*[V3TOT*p3tot + V2TOT_I*p2tot_i + V5TOT*p5tot
             + V6TOT*p6tot + V4TOT*p4tot - V0CIF_C*p0cif_c];
 E_w0gdpexp w0gdpexp = x0gdpexp + p0gdpexp;
```

```
! Excerpt 29 of TABLO input file: !
! Trade balance and other indices !

Variable
 (change) delB # (Nominal balance of trade)/{nominal GDP} #;
       x0imp_c # Import volume index, duty-paid weights #;
       w0imp_c # Value of imports plus duty #;
       p0imp_c # Duty-paid imports price index, local currency #;
     p0realdev # Real devaluation #;
        p0toft # Terms of trade #;
Equation
 E_delB 100*V0GDPEXP*delB=V4TOT*w4tot -V0CIF_C*w0cif_c-[V4TOT-V0CIF_C]*w0gdpexp;
 E_x0imp_c       x0imp_c = sum{c,COM, [V0IMP(c)/V0IMP_C]*x0imp(c)};
 E_p0imp_c       p0imp_c = sum{c,COM, [V0IMP(c)/V0IMP_C]*p0(c,"imp")};
 E_w0imp_c       w0imp_c = x0imp_c + p0imp_c;
 E_p0toft        p0toft = p4tot - p0cif_c;
 E_p0realdev     p0realdev =  p0cif_c - p0gdpexp;



! Excerpt 30 of TABLO input file: !
! Primary factor aggregates !

Variable
 (all,i,IND) employ(i) # Employment by industry #;
 employ_i   # Aggregate employment: wage bill weights #;
 x1cap_i    # Aggregate capital stock, rental weights #;
 x1lnd_i    # Aggregate land stock, rental weights #;
 x1prim_i   # Aggregate effective primary factor use #;
 xgdpfac    # Real GDP at factor cost (inputs) = x1prim_i #;
 p1prim_i   # Index of factor cost #;
 p1lab_io   # Average nominal wage #;
 realwage   # Average real wage #;
 p1cap_i    # Average capital rental #;
 p1lnd_i    # Average land rental #;
Equation
 E_employ  (all,i,IND) V1LAB_O(i)*employ(i) = sum{o,OCC, V1LAB(i,o)*x1lab(i,o)};
 E_employ_i V1LAB_IO*employ_i = sum{i,IND, V1LAB_O(i)*employ(i)};
 E_x1cap_i  V1CAP_I*x1cap_i   = sum{i,IND, V1CAP(i)*x1cap(i)};
 E_x1lnd_i  ID01[V1LND_I]*x1lnd_i   = sum{i,IND, V1LND(i)*x1lnd(i)};
 E_x1prim_i V1PRIM_I*x1prim_i = sum{i,IND, V1PRIM(i)*x1prim(i)};
 E_xgdpfac  xgdpfac = [1/V1PRIM_I]*
              [V1LAB_IO*employ_i + V1CAP_I*x1cap_i + V1LND_I*x1lnd_i];
 E_p1prim_i V1PRIM_I*p1prim_i = sum{i,IND, V1PRIM(i)*p1prim(i)};
 E_p1lab_io V1LAB_IO*p1lab_io = sum{i,IND, sum{o,OCC, V1LAB(i,o)*p1lab(i,o)}};
 E_realwage realwage = p1lab_io - p3tot;
 E_p1cap_i  V1CAP_I*p1cap_i   = sum{i,IND, V1CAP(i)*p1cap(i)};
 E_p1lnd_i  ID01[V1LND_I]*p1lnd_i   = sum{i,IND, V1LND(i)*p1lnd(i)};



! Excerpt 31 of TABLO input file: !
! Investment equations !

Variable
 (all,i,IND) ggro(i) # Gross growth rate of capital = Investment/capital #;
 (all,i,IND) gret(i) # Gross rate of return = Rental/[Price of new capital] #;
Equation
 E_ggro (all,i,IND) ggro(i) = x2tot(i) - x1cap(i);
 E_gret (all,i,IND) gret(i) = p1cap(i) - p2tot(i);
```

```
! Three alternative rules for investment:
  Choose which applies to each industry by setting JUST ONE of
  the corresponding elements of x2tot, finv1, finv2, or finv3 exogenous.
  Iff aggregate investment x2tot_i is exogenous, invslack must be endogenous. !
Variable
 (all,i,IND)  finv1(i)# Shifter to enforce DPSV investment rule #;
 (all,i,IND)  finv2(i)# Shifter for "exogenous" investment rule #;
 (all,i,IND)  finv3(i)# Shifter for longrun investment rule #;
 invslack # Investment slack variable for exogenizing aggregate investment #;

! Rule 1: Follows Section 19 of DPSV. The ratios Q and G are treated as
  parameters, just as in the original ORANI-IT implementation.  Attempts to
  improve the theory by updating these parameters have been found to
  occasionally lead to perversely signed coefficients !
Equation E_finv1 # DPSV investment rule #
 (all,i,IND) ggro(i) = finv1(i) + 0.33*[2.0*gret(i) - invslack];
! Note: above equation comes from substituting together DPSV
  equations 19.7-9. The value 0.33 and 2.0 correspond to the DPSV ratios
  [1/G.Beta] and Q (= ratio, gross to net rate of return) and are typical
  values of this ratio. In DPSV invslack was called "omega" and was interpreted
  as the "economy-wide rate of return" !

! Rule 2: For industries where investment is not mainly driven by current
  profits (eg, Education) make investment follow aggregate investment. !
Equation E_finv2 # Alternative rule for "exogenous" investment industries #
 (all,i,IND) x2tot(i) = x2tot_i + finv2(i);
! NB: you must not set ALL of finv2 exogenous else above would conflict with
  Equation E_x2tot_i !

! Rule 3: longrun investment rule: investment/capital ratios are exogenous !
Equation E_finv3 # Alternative long-run investment rule #
 (all,i,IND) ggro(i) = finv3(i) + invslack;

Variable f2tot   # Ratio, investment/consumption #;
Equation E_f2tot  x2tot_i = x3tot + f2tot;
! set f2tot exogenous and invslack endogenous
  to link aggregate investment to real consumption !

! Mechanism to allow fixed total capital to flow between sectors !
Variable
 (all,i,IND) fgret(i) # Shifter to lock together industry rates of return #;
             capslack # Slack variable to allow fixing aggregate capital #;
Equation E_fgret # Equation to force rates of return to move together #
 (all,i,IND) gret(i) = fgret(i) + capslack;



! Excerpt 32 of TABLO input file: !
! Labour market !

Variable
 (all,i,IND)(all,o,OCC) f1lab(i,o) # Wage shift variable #;
            (all,o,OCC) f1lab_i(o) # Occupation-specific wage shifter #;
            (all,o,OCC) x1lab_i(o) # Employment by occupation #;
 (all,i,IND)            f1lab_o(i) # Industry-specific wage shifter #;
                        f1lab_io   # Overall wage shifter #;

Coefficient (all,o,OCC) V1LAB_I(o) # Total wages, occupation o #;
Formula     (all,o,OCC) V1LAB_I(o) = sum{i,IND, V1LAB(i,o)};
Equation
 E_x1lab_i # Demand equals supply for labour of each skill #
  (all,o,OCC) V1LAB_I(o)*x1lab_i(o) = sum{i,IND, V1LAB(i,o)*x1lab(i,o)};
```

```
 E_p1lab   # Flexible setting of money wages #
  (all,i,IND)(all,o,OCC)
    p1lab(i,o)= p3tot + f1lab_io + f1lab_o(i) + f1lab_i(o) + f1lab(i,o);

Variable (all,o,OCC) p1lab_i(o) # Average wage of occupation #;
Equation E_p1lab_i # Average wage of occupation #
  (all,o,OCC) V1LAB_I(o)*p1lab_i(o) = sum{i,IND, V1LAB(i,o)*p1lab(i,o)};

! Excerpt 33 of TABLO input file: !
! Miscellaneous equations !

Variable (all,i,IND) f1oct(i)# Shift in price of "other cost" tickets #;
Equation E_p1oct # Indexing of prices of "other cost" tickets #
 (all,i,IND) p1oct(i) = p3tot + f1oct(i);   ! assumes full indexation !

Variable f3tot # Ratio, consumption/ GDP #;
Equation E_f3tot # Consumption function #
 w3tot = w0gdpexp + f3tot;

! Map between vector and matrix forms of basic price variables !
Variable
 (all,c,COM)  p0imp(c) # Basic price of imported goods = p0(c,"imp") #;
Equation E_p0dom  # Basic price of domestic goods = p0(c,"dom") #
 (all,c,COM) p0dom(c) = p0(c,"dom");
Equation E_p0imp  # Basic price of imported goods = p0(c,"imp") #
 (all,c,COM) p0imp(c) = p0(c,"imp");
```

1.5.1 *Adding variables for explaining results*

Part of the ORANI tradition is that simulation results, although voluminous, must all be capable of verbal explanation based on model equations and data. It is customary to examine and present results in great detail. The aim is to dispel any tendency to treat the model as a black box. These detailed analyses sometimes yield theoretical insights; for example, we may find that some mechanism which we thought to be of minor significance exerts a dominant force in certain sectors. Sometimes we discover errors—either in the data or in the model equations. Inappropriate theory may also lead to implausible results.

Results analysis, then, is an indispensable (but laborious) part of quality control for an economic model. To make it less painful, we often add equations and variables merely to help explain results. The next two excerpts are examples of this type of addition.

1.5.2 *Sales decomposition*

The sales decomposition, implemented in Excerpt 34, breaks down the percentage change in the total sales of each commodity into various intermediate and final demand categories. This would be useful, for example, if we wondered whether an increase in motor vehicle output was due to increases in either investment or household demand.

To decompose a percentage change in this way (i.e., find parts which add up to the whole), we have to perform some arithmetical manipulations. Suppose we have a variable X which is the sum of 2 parts:

$$X = A + B \quad \text{or} \quad PX = PA + PB \text{ (where P is a common price)} \tag{3}$$

then, for small percentage changes, we can write:

$$x = conta + contb \quad \text{where } conta = (PA/PX)a \text{ and } contb = (PB/PX)b \tag{4}$$

We call conta and contb the *contributions* of A and B to the percentage change in X.

For larger changes, which require a multistep computation, equation (36) would result in values for conta and contb which did not quite add up to the total percentage change in X^3. To avoid this, it is useful to specify both conta and contb as ordinary change variables and to define a new ordinary change variable, q, in such a way that the final result for q (after results for several computational steps have been acccumulated) is identical to that for x. This leads to the small change equation:

$$X^0 q = Xx \qquad \text{where } X^0 \text{ is the initial value of X,} \tag{5}$$

and to the revised decomposition:

$$q = \text{conta} + \text{contb} \tag{6}$$

$$\text{where} \quad \text{conta} = (PA/PX^0)a \quad \text{and} \quad \text{contb} = (PB/PX^0)b \tag{7}$$

In Excerpt 34, INITSALES is the *initial quantity* of SALES *measured in current prices*, analogous to PX^0 above. The last, Total, column of the SalesDecomp variable is the sum of the preceding 7 and should be identical to x0com(c), the percentage change in commodity output. Each of the first 7 columns shows how some demand category contributed to x0com. The delSale variable, calculated earlier, contains the ordinary changes in quantities, measured in current prices.

```
! Excerpt 34 of TABLO input file: !
! Decomposition of sales change by destination !

Coefficient
 (all,c,COM) INITSALES(c) # Initial volume of SALES at current prices #;
Formula
 (initial) (all,c,COM) INITSALES(c) = SALES(c);
Update (all,c,COM) INITSALES(c) = p0com(c);
Set DESTPLUS # Sale Categories #
 (Interm, Invest, HouseH, Export, GovGE, Stocks, Margins, Total);
Subset DEST is subset of DESTPLUS;
Variable (change)
  (all,c,COM)(all,d,DESTPLUS) SalesDecomp(c,d) # Sales decomposition #;
Equation E_SalesDecompA
(all,c,COM)(all,d,DEST) INITSALES(c)*SalesDecomp(c,d) = 100*delSale(c,"dom",d);
Equation E_SalesDecompB
 (all,c,COM) SalesDecomp(c,"Total")= sum{d,DEST, SalesDecomp(c,d)};
```

1.5.3 *The Fan decomposition*

Suppose our simulation predicts an increase in domestic production of Textiles. This could be due to three causes:

- the local market effect: an increase in local usage of Textiles, whether domestically-produced or imported;
- the export effect: an increase in exports of Textiles; or
- the domestic share effect: a shift in local usage of Textiles, from imported to domestically-produced.

Very often these 3 effects will work in different directions; for example, a increase in foreign demand might pull local producers up the supply curve, so increasing the domestic price and

[3] The reason is that during a multistep computation percentage changes are compounded, whilst ordinary changes are added.

facilitating import penetration. The decomposition of Fan[4] aims to show the relative magnitude of these 3 contributions to output change.

Excerpt 35 starts by defining x0loc, the percentage change in local sales from both sources. Equation E_fandecompA says that this percentage, weighted by the value of local domestic sales, is the local market component of the percentage change in domestic production. Similarly, equation E_fandecompB defines the export component. In these equations INITSALES corresponds to the term PX^0 in equation (39): it is the initial value of sales, updated only by the change in price. Equation E_fandecompC defines the domestic share component as a residual[5]. Finally, equation E_fandecompD corresponds to equation (37).

```
! Excerpt 35 of TABLO input file: !
! Decomposition of Fan !

Set FANCAT # Parts of Fan decomposition #
  (LocalMarket, DomShare, Export, Total);
Variable
(all,c,COM) x0loc(c)  # Real percent change in LOCSALES (dom+imp)  #;
(change)(all,c,COM)(all,f,FANCAT) fandecomp(c,f) # Fan decomposition #;
Coefficient (all,c,COM) LOCSALES(c)  # Total local sales of dom + imp good c #;
Formula     (all,c,COM) LOCSALES(c) = DOMSALES(c) + V0IMP(c);

Equation
 E_x0loc # %Growth in local market #
  (all,c,COM) LOCSALES(c)*x0loc(c) = DOMSALES(c)*x0dom(c) + V0IMP(c)*x0imp(c);

 E_fandecompA # Growth in local market effect #
  (all,c,COM)  INITSALES(c)*fandecomp(c,"LocalMarket") = DOMSALES(c)*x0loc(c);
! The local market effect is the % change in output that would have occurred
if local sales of the domestic product had followed dom+imp sales (x0loc) !

 E_fandecompB # Export effect #
  (all,c,COM) INITSALES(c)*fandecomp(c,"Export") = V4BAS(c)*x4(c);

 E_fandecompC # Import leakage effect - via residual #
  (all,c,COM) fandecomp(c,"Total") = fandecomp(c,"LocalMarket")
     + fandecomp(c,"DomShare") + fandecomp(c,"Export");

 E_fandecompD # Fan total = x0com #
  (all,c,COM) INITSALES(c)*fandecomp(c,"Total") = SALES(c)*x0com(c);
```

1.5.4 *The expenditure side GDP decomposition*

Excerpt 36 breaks down changes in real GDP into the contributions of the main expenditure-side aggregates (contGDPexp). This enables us to quickly see how much of the change in real expenditure-side GDP is due, say, to a change in aggregate investment. The mathematics is the

---

[4] Named after Fan Ming-Tai of the Academy of Social Sciences, Beijing; their PRCGEM is one of the most elaborate versions of ORANI.

[5] No interactive term is concealed in the residual. Because these decompositions are specified in small change terms, the changes due to each part add up to the change in the whole. To convince yourself, retrace the example starting at equation (35) with the multiplicative form X = AB, leading to $X^0q = Xa + Xb$, with contribution terms $(X/X^0)a$ and $(X/X^0)b$. However, the cumulative results of these contributions can be defined only as a path integral of the contribution terms computed at each solution step. Hence they are not (quite) invariant to the details of our solution procedure. See also footnote 17.

same as that of the Sales decomposition in Excerpt 34. For convenience, the contributions of exports and imports are combined into the variable contBOT. Note that contBOT, like each component of contGDPexp, is a *real* contribution: it takes no account of changes in export or import prices.

```
! Excerpt 36 of TABLO input file: !
! GDP decomposition !

Set EXPMAC # Expenditure Aggregates #
 (Consumption, Investment, Government, Stocks, Exports, Imports);
Variable (change) (all,e,EXPMAC)
  contGDPexp(e) # Contributions to real expenditure-side GDP #;
Coefficient INITGDP # Initial real GDP at current prices #;
Formula (initial) INITGDP = V0GDPEXP;
Update INITGDP = p0gdpexp;
Equation
  E_contGDPexpA  INITGDP*contGDPexp("Consumption") = V3TOT*x3tot;
  E_contGDPexpB  INITGDP*contGDPexp("Investment")  = V2TOT_I*x2tot_i;
  E_contGDPexpC  INITGDP*contGDPexp("Government")  = V5TOT*x5tot;
  E_contGDPexpD  INITGDP*contGDPexp("Stocks")      = V6TOT*x6tot;
  E_contGDPexpE  INITGDP*contGDPexp("Exports")     = V4TOT*x4tot;
  E_contGDPexpF  INITGDP*contGDPexp("Imports")     = - V0CIF_C*x0cif_c;

Variable (change) contBOT # Contribution of BOT to real expenditure-side GDP #;
Equation  E_contBOT contBOT =  contGDPexp("Exports") + contGDPexp("Imports");

! Exact GDP decomposition from income side !
Set ContInc (Land, Labour, Capital, IndTax, TechChange);
Variable
(change) x0gdpinc    # Real GDP from the income side #;
(change) (all,c,ContInc) contGDPinc(c) # Income side real GDP decomposition #;
Equation
  E_contGDPincA  contGDPinc("Land")       = [V1LND_I/INITGDP]*x1lnd_i;
  E_contGDPincB  contGDPinc("Labour")     = [V1LAB_IO/INITGDP]*employ_i;
  E_contGDPincC  contGDPinc("Capital")    = [V1CAP_I/INITGDP]*x1cap_i;
  E_contGDPincD  contGDPinc("IndTax")     =
    sum{i,IND, [V1OCT(i)/INITGDP]*x1oct(i)}
  + sum{i,IND, [sum{p,ALLPTAX,V1PTX(i,p)}/INITGDP]*x1tot(i)}
  + sum{c,COM, [V0TAR(c)/INITGDP]*x0imp(c)}
  + sum{c,COM,sum{s,SRC, sum{i,IND, [sum{t,ALLCTAX,V1TAX(c,s,i,t)}/INITGDP]*x1(c,s,i)}}}
  + sum{c,COM,sum{s,SRC, sum{i,IND, [sum{t,ALLCTAX,V2TAX(c,s,i,t)}/INITGDP]*x2(c,s,i)}}}
  + sum{c,COM,sum{s,SRC, [sum{t,ALLCTAX,V3TAX(c,s,t)}/INITGDP]*x3(c,s)}}
  + sum{c,COM, [sum{t,ALLCTAX,V4TAX(c,t)}/INITGDP]*x4(c)}
  + sum{c,COM,sum{s,SRC, [sum{t,ALLCTAX,V5TAX(c,s,t)}/INITGDP]*x5(c,s)}};
 E_contGDPincE  contGDPinc("TechChange") =
  -sum{c,COM,sum{s,SRC,sum{i,IND,[V1PUR(c,s,i)/INITGDP]*[a1(c,s,i)+a1_s(c,i)]}}}
  -sum{c,COM,sum{s,SRC,sum{i,IND,[V2PUR(c,s,i)/INITGDP]*[a2(c,s,i)+a2_s(c,i)]}}}
  -sum{i,IND,[V1LAB_O(i)/INITGDP]*a1lab_o(i)}
  -sum{i,IND,[V1CAP(i)/INITGDP]*a1cap(i)}
  -sum{i,IND,[V1LND(i)/INITGDP]*a1lnd(i)}
  -sum{i,IND,[V1OCT(i)/INITGDP]*a1oct(i)}
  -sum{i,IND,[V1PRIM(i)/INITGDP]*a1prim(i)}
  -sum{c,COM,sum{s,SRC,sum{i,IND,sum{m,MAR,
   [V1MAR(c,s,i,m)/INITGDP]*a1mar(c,s,i,m)}}}}
  -sum{c,COM,sum{s,SRC,sum{i,IND,sum{m,MAR,
   [V2MAR(c,s,i,m)/INITGDP]*a2mar(c,s,i,m)}}}}
  -sum{c,COM,sum{s,SRC,sum{m,MAR,[V3MAR(c,s,m)/INITGDP]*a3mar(c,s,m)}}}
  -sum{c,COM,sum{m,MAR,[V4MAR(c,m)/INITGDP]*a4mar(c,m)}}
  -sum{c,COM,sum{s,SRC,sum{m,MAR,[V5MAR(c,s,m)/INITGDP]*a5mar(c,s,m)}}}
  -sum{i,IND,[V2TOT(i)/INITGDP]*a2tot(i)}
  -sum{i,IND,[V1CST(i)/INITGDP]*a1tot(i)};
  E_x0gdpinc  # Decomposition of real GDP from income side #
   x0gdpinc = sum{c,ContInc, contGDPinc(c)};
```

The second part of Excerpt 36 breaks down changes in real GDP from the income side: into the contributions due to primary factor usage, indirect taxes, and technological change. If primary factor endowments and technology are fixed, the indirect tax terms may be identified with changes in allocative efficiency.

1.5.5 *Checking the data*

A model rendered in the TABLO language is a type of computer program, and like other computer programs tends to contain errors. We employ a number of strategies to prevent errors and to make errors apparent. One strategy is to check all conditions which the initial data must satisfy. This is done in Excerpt 37. The conditions are:

- The row sums of the MAKE matrix must equal the row sums of the BAS and MAR rows of Figure 1. That is, the output of domestically produced commodities must equal the total of the demands for them.
- The column sums of the MAKE matrix must equal the sum of the first, producers', column of Figure 1. That is, the value of output by each industry must equal the total of production costs.
- The average value of the household expenditure elasticities, EPS, should be one. The average should be computed using the expenditure weights, V3PUR_S.

To check these conditions, the items PURE_PROFITS, LOST_GOODS, and EPSTOT are stored on the SUMMARY file. Their values should be near to zero (or one, for EPSTOT).

The Assertion statements which come next enforce the above rules. If the specified condition is not met during some simulation, GEMPACK will stop with an error message indicating which condition is not satisfied. The first two Assertions are checked before every stage of a multi-step simulation. The last one, for EPSTOT, should be true before and after (but not necessarily *during*) a simulation.

```
! Excerpt 37 of TABLO input file: !
! Check identities !

Coefficient                    ! coefficients for checking !
 (all,i,IND) DIFFIND(i)  # COSTS-MAKE_C : should be zero #;
 (all,c,COM) DIFFCOM(c)  # SALES-MAKE_I : should be zero #;
             EPSTOT      # Average Engel elasticity: should = 1 #;
Formula
 (all,i,IND) DIFFIND(i)  = V1TOT(i) - MAKE_C(i);
 (all,c,COM) DIFFCOM(c)  = SALES(c) - MAKE_I(c);
             EPSTOT      = sum{c,COM, S3_S(c)*EPS(c)};

Write ! we file these numbers BEFORE the assertions below !
 DIFFIND to file SUMMARY header "DIND";
 DIFFCOM to file SUMMARY header "DCOM";
 EPSTOT  to file SUMMARY header "ETOT";

Assertion ! if below not true, program will stop with message !
 # DIFFIND = V1TOT-MAKE_C = tiny # (all,i,IND) ABS[DIFFIND(i)/V1TOT(i)] <0.001;
 # DIFFCOM = SALES-MAKE_I = tiny # (all,c,COM) ABS[DIFFCOM(c)/SALES(c)] <0.001;
 (initial) # Average Engel elasticity = 1 #  ABS[1-EPSTOT] <0.001;
```

1.5.6 *Summarizing the data*

The next few Excerpts collect together various summaries of the data and store these on file in a form that is convenient for later viewing. These summaries are useful for checking the plausibility of data and for explaining simulation results. Excerpt 38 groups into vectors the various components of, first, GDP from the expenditure side; second, GDP from the income side; and third, the components of total indirect taxes[6].

```
! Excerpt 38 of TABLO input file: !
! Summary: components of GDP from income and expenditure sides !

Coefficient (all,e,EXPMAC)  EXPGDP(e) # Expenditure Aggregates #;
Formula
 EXPGDP("Consumption") = V3TOT;
 EXPGDP("Investment")  = V2TOT_I;
 EXPGDP("Government")  = V5TOT;
 EXPGDP("Stocks")      = V6TOT;
 EXPGDP("Exports")     = V4TOT;
 EXPGDP("Imports")     = -V0CIF_C;
Write EXPGDP to file SUMMARY header "EMAC";

Set INCMAC # Income Aggregates #
 (Land, Labour, Capital,  IndirectTax);
Coefficient (all,i,INCMAC)  INCGDP(i) # Income Aggregates #;
Formula
 INCGDP("Land")        = V1LND_I;
 INCGDP("Labour")      = V1LAB_IO;
 INCGDP("Capital")     = V1CAP_I;
 INCGDP("IndirectTax") = V0TAX_CSI;
Write INCGDP to file SUMMARY header "IMAC";

Set TAXMAC # Tax Aggregates #
  (Intermediate,Investment,Consumption,Exports,Government,OCT,ProdTax,Tariff);
Coefficient (all,t,TAXMAC)  TAX(t) # Tax Aggregates #;
Formula
 TAX("Intermediate") = V1TAX_CSI;
 TAX("Investment")   = V2TAX_CSI;
 TAX("Consumption")  = V3TAX_CS;
 TAX("Exports")      = V4TAX_C;
 TAX("Government")   = V5TAX_CS;
 TAX("OCT")          = V1OCT_I;
 TAX("ProdTax")      = V1PTX_I;
 TAX("Tariff")       = V0TAR_C;
Write TAX to file SUMMARY header "TMAC";
```

Excerpt 39 forms a matrix showing the main parts of production cost for each industry.

```
! Excerpt 39 of TABLO input file: !
! Summary: matrix of industry costs !

Set COSTCAT # Cost Categories #
 (IntDom, IntImp, Margin, ComTax, Lab, Cap, Lnd, ProdTax, OCT);
Coefficient (all,i,IND)(all,co,COSTCAT) COSTMAT(i,co) # Cost Matrix #;
```

[6] GEMPACK stores data in its own, binary, format. A Windows program, ViewHAR, is normally used for viewing or modifying these so-called HAR files. The data matrices created here are designed to be convenient for examining with ViewHAR. ViewHAR automatically calculates and displays subtotals, so that total GDP, for example, does not need to be included in the summary vectors defined here.

```
Formula
 (all,i,IND) COSTMAT(i,"IntDom")  = sum{c,COM, V1BAS(c,"dom",i)};
 (all,i,IND) COSTMAT(i,"IntImp")  = sum{c,COM, V1BAS(c,"imp",i)};
 (all,i,IND) COSTMAT(i,"Margin")  =
                      sum{c,COM, sum{s,SRC, sum{m,MAR, V1MAR(c,s,i,m)}}};
 (all,i,IND) COSTMAT(i,"ComTax")  = sum{c,COM, sum{s,SRC,
sum{t,ALLCTAX,V1TAX(c,s,i,t)}}};
 (all,i,IND) COSTMAT(i,"Lab")     = V1LAB_O(i);
 (all,i,IND) COSTMAT(i,"Cap")     = V1CAP(i);
 (all,i,IND) COSTMAT(i,"Lnd")     = V1LND(i);
 (all,i,IND) COSTMAT(i,"ProdTax") = sum{p,ALLPTAX,V1PTX(i,p)};
 (all,i,IND) COSTMAT(i,"OCT")     = V1OCT(i);
Write COSTMAT to file SUMMARY header "CSTM";
```

Excerpt 40 calculates for each main destination, the basic, margin, and commodity tax components of purchases by commodity and source.

```
! Excerpt 40 of TABLO input file: !
! Summary: basic, margins and taxes !
Set
SALECAT2 # SALE Categories # (Interm, Invest, HouseH, Export, GovGE, Stocks);
FLOWTYPE # Type of flow # (Basic, Margin, TAX);
Coefficient
(all,c,COM)(all,f,FLOWTYPE)(all,s,SRC)(all,sa,SALECAT2) SALEMAT2(c,f,s,sa)
    # Basic, margin and tax components of purchasers' values #;
Formula

all,c,COM)(all,f,FLOWTYPE)(all,s,SRC)(all,sa,SALECAT2) SALEMAT2(c,f,s,sa)=0;
(all,c,COM)(all,s,SRC) SALEMAT2(c,"Basic",s,"Interm") = sum{i,IND,V1BAS(c,s,i)};
(all,c,COM)(all,s,SRC) SALEMAT2(c,"Tax"  ,s,"Interm") =
sum{i,IND,sum{t,ALLCTAX,V1TAX(c,s,i,t)}};
(all,c,COM)(all,s,SRC) SALEMAT2(c,"Margin",s,"Interm") =
                                    sum{i,IND, sum{m,MAR, V1MAR(c,s,i,m)}};
(all,c,COM)(all,s,SRC) SALEMAT2(c,"Basic",s,"Invest") = sum{i,IND,V2BAS(c,s,i)};
(all,c,COM)(all,s,SRC) SALEMAT2(c,"Tax"  ,s,"Invest") =
sum{i,IND,sum{t,ALLCTAX,V2TAX(c,s,i,t)}};
(all,c,COM)(all,s,SRC) SALEMAT2(c,"Margin",s,"Invest") =
                                    sum{i,IND, sum{m,MAR, V2MAR(c,s,i,m)}};
(all,c,COM)(all,s,SRC) SALEMAT2(c,"Basic",s,"HouseH") = V3BAS(c,s);
(all,c,COM)(all,s,SRC) SALEMAT2(c,"Tax"  ,s,"HouseH") = sum{t,ALLCTAX,V3TAX(c,s,t)};
(all,c,COM)(all,s,SRC) SALEMAT2(c,"Margin",s,"HouseH")= sum{m,MAR,V3MAR(c,s,m)};
(all,c,COM)(all,s,SRC) SALEMAT2(c,"Basic",s,"GovGE") = V5BAS(c,s);
(all,c,COM)(all,s,SRC) SALEMAT2(c,"Tax"  ,s,"GovGE") = sum{t,ALLCTAX,V5TAX(c,s,t)};
(all,c,COM)(all,s,SRC) SALEMAT2(c,"Margin",s,"GovGE")= sum{m,MAR,V5MAR(c,s,m)};
(all,c,COM)            SALEMAT2(c,"Basic","dom","Export") = V4BAS(c);
(all,c,COM)            SALEMAT2(c,"Tax"  ,"dom","Export") = sum{t,ALLCTAX,V4TAX(c,t)};
(all,c,COM)          SALEMAT2(c,"Margin","dom","Export")= sum{m,MAR,V4MAR(c,m)};
(all,c,COM)(all,s,SRC) SALEMAT2(c,"Basic",s,"Stocks") = V6BAS(c,s);
Write SALEMAT2 to file SUMMARY header "MKUP";
```

1.5.7 *Import shares and short-run supply elasticities*

Excerpt 41 computes and stores 2 more items which help explain results. The higher is the import share IMPSHR for some commodity, the more vulnerable is the associated local industry to import competition.

Short-run simulation results are often coloured by the effects of inelastic supply schedules for capital-intensive industries. We compute and store the short-run supply elasticity SUPPLYELAST to help identify such industries. The formula used is derived in Appendix E. The closely-related primary factor shares can be seen from the FACTOR matrix (using the *row shares* feature of the GEMPACK program ViewHAR).

```
! Excerpt 41 of TABLO input file: !
! Import shares and short-run supply elasticities !

Coefficient  (all,c,COM) IMPSHR(c) # Share of imports in local market #;
Formula      (all,c,COM) IMPSHR(c) = V0IMP(c)/[TINY+DOMSALES(c)+V0IMP(c)];
Write IMPSHR to file SUMMARY header "MSHR";

Coefficient (all,i,IND) SUPPLYELAST(i) # Short-run supply elasticity #;
Zerodivide default 999;
Zerodivide (nonzero_by_zero) default 999;
Formula     (all,i,IND) SUPPLYELAST(i) =
    SIGMA1PRIM(i)*V1LAB_O(i)*V1TOT(i)/[V1PRIM(i)*{V1CAP(i)+V1LND(i)}];
Zerodivide off;
Zerodivide (nonzero_by_zero) off;
Write SUPPLYELAST to file SUMMARY header "SRSE";

Set FAC # Primary Factors #  (Lab, Cap, Lnd);
Coefficient (all,i,IND)(all,f,FAC) FACTOR(i,f) # Primary Factor Costs #;
Formula
 (all,i,IND) FACTOR(i,"Lab")     = V1LAB_O(i);
 (all,i,IND) FACTOR(i,"Cap")     = V1CAP(i);
 (all,i,IND) FACTOR(i,"Lnd")     = V1LND(i);
Write FACTOR to file SUMMARY header "FACT";
```

1.5.8 *Storing Data for Other Computations*

It is often useful to extract data from the model for other calculations. For example, we might wish to combine levels data with change results in our presentation of simulation results. Another important use of such data is in aggregating the model database. It is wise to prepare the initial model database at the highest level of disaggregation supported by available Input-Output tables. This large database can be aggregated later, if desired, to a smaller number of sectors. A specialized program, AggHAR, may ease the aggregation task.

For flows data, each item in the aggregated database is simply the sum of corresponding sectors in the original database. Parameters, however, can not normally be added together. Instead, aggregated parameters are normally weighted averages of the original parameters. The purpose of Excerpt 42 is store such weights on a file. For example, the parameter SIGMA2 (Armington elasticity between domestic and imported commodities used for investment) could be aggregated using the weight vector V2PUR_SI. Information about which weight to use for each elasticity is conveyed to AggHAR by the WAGGSET vector of strings. For example, the first element of WAGGSET, "SCET@@@@1TOT" indicates that the MAKE CET elasticities stored at header SCET on the ORANI-ITG input data file should be aggregated using the industry output values stored at header 1TOT on the summary file[7].

[7] The set WAGGSET is used to work around a GEMPACK limitation: the only way to write lists of strings (such as headers) to a file is to write them as set elements. Furthermore, set elements may not begin with a digit.

```
! Excerpt 42 of TABLO input file: !
! Weight vectors for use in aggregation and other calculations !

Coefficient (all,c,COM) V1PUR_SI(c)  # Dom+imp intermediate purch. value #;
            (all,c,COM) V2PUR_SI(c)  # Dom+imp investment purch. value #;
            (all,c,COM) V5PUR_S(c)   # Dom+imp government purch. value #;
            (all,c,COM) V6BAS_S(c)   # Dom+imp inventories #;
Formula     (all,c,COM) V1PUR_SI(c)  = sum{i,IND, V1PUR_S(c,i)};
            (all,c,COM) V2PUR_SI(c)  = sum{i,IND, V2PUR_S(c,i)};
            (all,c,COM) V5PUR_S(c)   = sum{s,SRC, V5PUR(c,s)};
            (all,c,COM) V6BAS_S(c)   = sum{s,SRC, V6BAS(c,s)};

Write                                  ! weight to aggregate..... !
 V1TOT    to file SUMMARY header "1TOT";!MAKE CET elasticities at SCET !
 V4PUR    to file SUMMARY header "4PUR";!export elasticities at header P018!
 V1LAB_O  to file SUMMARY header "LAB1";!CES inter-skill elasticities at P018!
 V1PRIM   to file SUMMARY header "VLAD";!CES primary-factor elasts at P018!
 V1PUR_SI to file SUMMARY header "1PUR";!Armington elasticities at 1ARM !
 V2PUR_SI to file SUMMARY header "2PUR";!Armington elasticities at 2ARM !
 V3PUR_S  to file SUMMARY header "3PUR";!Armington elasticities at 2ARM !

Set WAGGSET # Instructions to AGGHAR for weighted aggregation of parameters # (
    SCET@@@@1TOT,
    P018@@@@4PUR,
    ITEX@@@@4PUR,
    XPEL@@@@3PUR,
    P028@@@@VLAD,
    SLAB@@@@LAB1,
    xx1ARM@@1PUR,  ! leading "xx" is ignored by AGGHAR !
    xx2ARM@@2PUR,
    xx3ARM@@3PUR);
Write (set) WAGGSET to file SUMMARY header "WAGG";


! Other useful values !
Coefficient  (all,c,COM) TARFRATE(c) # Ad valorem tariff rate #;
Formula      (all,c,COM) TARFRATE(c) = V0TAR(c)/[TINY+V0CIF(c)];
Write
  TARFRATE to file SUMMARY header "TRAT";
  V0TAR    to file SUMMARY header "0TAR";
  V0CIF    to file SUMMARY header "0CIF";
  V0IMP    to file SUMMARY header "0IMP";
  MAKE     to file SUMMARY header "MAKE";
  V1CAP    to file SUMMARY header "1CAP";
  V6BAS_S  to file SUMMARY header "6BSS";
  V2TOT    to file SUMMARY header "2TOT";
  V5PUR_S  to file SUMMARY header "5PUR";
```

1.6 Model condensation

Excerpt 49 of the model contains condensation instructions for GEMPACK. In many cases models need to be reduced in size before it is practical to solve them. For example, the ORANI-IT model has hundreds of equations, and thousands of variables. Solving such a system is time-consuming and requires a large amount of computer memory. Solving the system in its entirety is also unnecessary because in any given simulation there would be many variables which are exogenous and not shocked. Also, if a variable is not of interest for a particular simulation, it may be substituted out, thus reducing the size of the system of equations. You can condense the model by instructing the TABLO program to:

a) *Omit specified variables from the system*. This option is useful for variables which will be exogenous and unshocked (zero percentage change). Normally it allows us to dispense with the bulk of the technical and taste change terms (denoted 'a' in ORANI-IT).
b) *Substitute out specified variables using specified equations*. This results in fewer equations (although sometimes the equations become more complex).

For example, consider the following equations in ORANI-IT:

Equation E_x1lab_o *# Industry demands for effective labour #*
(**all**,i,IND) x1lab_o(i) - a1lab_o(i) =
x1prim(i) - SIGMA1PRIM(i)***[**p1lab_o(i) + a1lab_o(i) - p1prim(i)**]**;

Equation E_x1prim *# Demands for primary factor composite #*
(**all**,i,IND) x1prim(i) - **[**a1prim(i) + a1tot(i)**]** = x1tot(i);

In these equations, the 'a' terms are exogenous technology variables. In a simulation when they are not shocked, we can omit them. If we also substitute out the variable *x1prim*, then the two equations become one equation:

(**all**,i,IND) x1lab_o(i) = x1tot(i) - SIGMA1PRIM(i)***[**p1lab_o(i) - p1prim(i)**]**;

- *Backsolving*

In our example, the substitution in (b) means that the values for *x1prim* are not calculated, and so cannot be reported when we carry out a simulation. If we want to be able to look at the results for *x1prim*, but still want to condense the system, we should use the *backsolving* command instead of the *substitution* command. Under the backsolving command, the system still substitutes the variable *x1prim* out in solving the system, but after that it uses the results of the remaining variables (*x1tot* in this example) to solve for *x1prim*.

The variables for omission and the equation-variable pairs for substitution or backsolving are listed in the tablo file ORANI-IT.tab.

1.7 Equations for aggregations of simulation results

To assist interpretation and reporting, we may wish to look at simulation results for a set of sectors, which are at a more aggregated level than the 63 sectors of the core model. In this version of ORANI-IT, for example, we aggregate the results of the 63 sectors to 6 broad sectors, namely Agriculture, Mining, Manufactures, Utilities, Construction and Services.

Suppose that we want to aggregate industry-specific results for variable X (X_i) to the results for broad industry sectors (X_s). Then the aggregation is conducted by using equations of the form:

$x_s = \sum_{i \in SEC} S_{i/s} x_i$ or in intermediate form: $V_s x_s = \sum_{i \in SEC} V_i x_i$

where

- X_i and x_s are the percentage change in variable x at the industrial and sectoral level respectively;
- V_i and V_s are the relevant values used for calculating sectoral weights. For example, in aggregating exports, V_i might be the value of Clothing exports, in which case V_s would be the value of *Textile, Clothing and Footwear* exports.
- $S_{i/s} = V_i / V_s$ is the shares of relevant values of X at the industry level in the value of X at the sectoral level.