INDUSTRY
COMMISSION

Creating Synthetic Single Region Input-Output Data for SALTER: Hong Kong and the Rest of the World

by

James Tormey

SALTER Working Paper No. 20

JUNE 1993

CONTENTS

Figures

Tables

1 INTRODUCTION

The final SALTER database is constructed from 16 single region input-output databases, one for each SALTER region, and a trade database describing bilateral trade flows between these regions. The regions currently in the SALTER model are Australia, New Zealand, Canada, the United States, Japan, Korea, the European Community, Indonesia, Malaysia, Philippines, Singapore, Thailand, the People's Republic of China, Hong Kong, Taiwan and the Rest of the World.

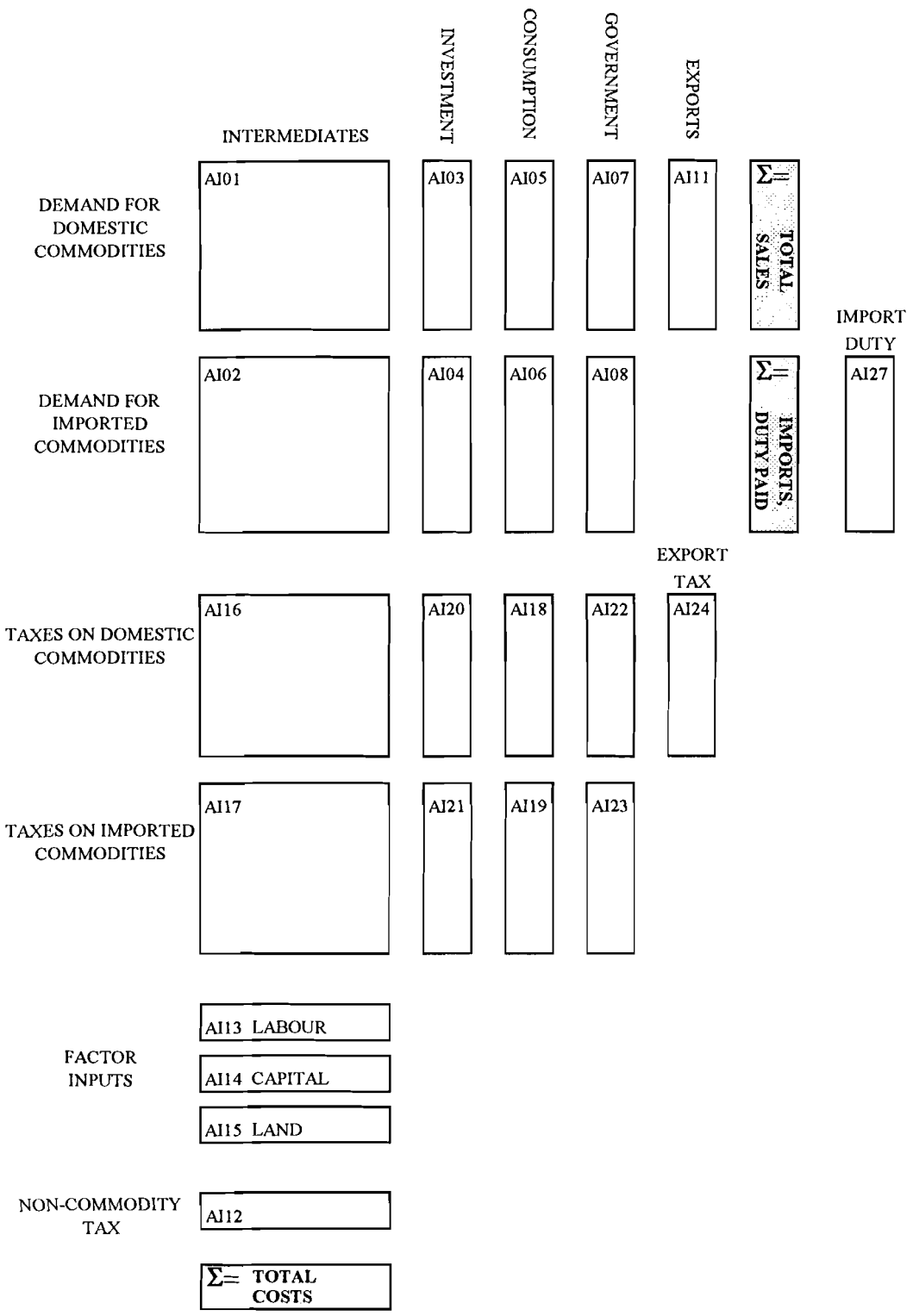
The single region databases of Hong Kong and the rest of the world (ROW) differ in construction from the other single region databases. No published input-output data exists for these regions so their databases contain very high proportions of estimated data. Each single region database contains about 3000 individual numbers. About half the numbers in the databases for Hong Kong and the Rest of the World involve some degree of approximation and the remaining data are set to zero. The estimation process employs other single region databases and raw statistics published for Hong Kong and the Rest of the World.

This paper summarises the structure of a single region database, explains the logic underlying the data estimation process and provides specific information on the construction of the Hong Kong and ROW databases. Hambley (1993) explains how the single region databases for other region were obtained from published input-output data and Brown *et al* (1993) provides details on how the final database is assembled.

The structure of a single region database

Figure 1 illustrates the organisation of a single region database. Each box in Figure 1 represents an array of values. These arrays are stored as "header arrays", described in Mikkelsen and Pearson (1986). For example, AI03 is the header array containing the values of investment expenditure on each domestically produced commodity. The header arrays in Figure 1 are either of size 37*37 or 37*1 because the SALTER model uses 37 commodity and industry groups. Square matrices are drawn as such. Note that some headers have been excluded from the figure. Values for depreciation, income tax, general government surplus and capital mobility data are added at a later stage (see Brown *et al* 1993).

Figure 1: **The structure of a single region input-output database**



By construction, there are the same number of commodities as industries in the SALTER model and they have the same names. A numbered list of SALTER industries and commodities appears in Table 1.

Table 1: **The ordered list of SALTER industries and commodities**

Number	*Name*	*Sector*
1	Paddy rice	Agriculture
2	Wheat	Agriculture
3	Other grains	Agriculture
4	Non-grain crops	Agriculture
5	Wool	Agriculture
6	Other livestock products	Agriculture
7	Forestry	Agriculture
8	Fishing	Agriculture
9	Coal	Mining
10	Oil	Mining
11	Gas	Mining
12	Other minerals	Mining
13	Processed rice	Manufacturing (Food)
14	Meat products	Manufacturing (Food)
15	Milk products	Manufacturing (Food)
16	Other food products	Manufacturing (Food)
17	Beverages and tobacco	Manufacturing (Food)
18	Textiles	Manufacturing (Non-food)
19	Wearing apparel	Manufacturing (Non-food)
20	Leather products	Manufacturing (Non-food)
21	Lumber products	Manufacturing (Non-food)
22	Pulp, paper and printing	Manufacturing (Non-food)
23	Petroleum and coal products	Manufacturing (Non-food)
24	Chemicals, rubber and plastic	Manufacturing (Non-food)
25	Non-metallic minerals	Manufacturing (Non-food)
26	Primary iron and steel	Manufacturing (Non-food)
27	Non-ferrous metals	Manufacturing (Non-food)
28	Fabricated metal products	Manufacturing (Non-food)
29	Transport equipment	Manufacturing (Non-food)
30	Other machinery and equipment	Manufacturing (Non-food)
31	Other manufacturing	Manufacturing (Non-food)
32	Electricity, gas and water	Services
33	Construction	Services
34	Trade and transport	Services
35	Other services (Private)	Services
36	Other services (Government)	Services
37	Ownership of dwellings	Services

The database gives values for all flows of goods and services produced domestically or imported by a country. These elements of demand are shown in Figure 1 as headers AI01-AI08 and AI11. Domestic and imported commodities are demanded for final demand and as inputs to production (intermediate demand). Final demand for imported commodities is represented by headers AI04, AI06 and AI08. Headers AI03, AI05, AI07 and AI11 represent final demand for domestically produced commodities. Note that all headers, including the tax headers, should ultimately report values in 1988 $USmillion.

The database also details all commodity taxes, import and export taxes and non-commodity indirect taxes. Subsidies are represented by a negative tax. By assumption, most taxes for Hong Kong and the ROW are zero. This assumption is necessitated by a scarcity of the required data but can at least be justified in the case of Hong Kong as an approximation to the actual tax rates. The only non-zero tax rates are import duties in ROW (see Chapter 3). As noted, headers containing other fiscal data and capital mobility data are added in a later stage.

All SALTER databases obey the accounting identity that the value of total sales of each domestically produced commodity is equal to the value of total costs of producing that commodity. The total cost vector is the sum by industry of payments to farm land, capital, and labour, and the tax-inclusive cost of all intermediate inputs to the production process. The vector of total sales is the sum by commodity of intermediate and final demand for domestically produced goods. Figure 1 shows these sums.

Principles of data estimation

Statistics collected for Hong Kong and ROW come from World Bank and United Nations sources. These statistics shall be referred to as the *published statistics*. The published statistics report Gross Domestic Product at market prices (GDP), value added, value of industry output, payments to capital, payments to labour, private consumption, government consumption and investment. Value added is the sum of payments to primary factors of production, that is, capital (which includes land used in the SALTER agricultural industries) plus labour. Value added is equivalent to GDP at factor cost.

Statistics reported by commodity are private consumption, government consumption and investment. Statistics reported by industry are output, total value added and the components of value added. The published statistics cover a range of industry and commodity classifications. Some statistics, like GDP, are reported at the economy-wide level of aggregation. At the other end of the scale are statistics reported for single SALTER industries. Many of the United

Nations manufacturing statistics are reported at this level. Other United Nations data and World Bank data appear at levels of aggregation between the national level the and SALTER industry level.

Each statistic may be relevant to several headers in the single country database. For example, private consumption reported in the published statistics refers to headers AI05 (domestic consumption) and AI06 (imported consumption) of Figure 1. Thus each of the published statistics is distributed over several headers. The GDP figure and values in other single country databases determine the shares by which the published statistics are distributed over the database headers. The disaggregation technique used in any particular case depends upon the extent of data availability.

Often, distributing a published statistic over domestic and imported headers is not sufficient. The published statistics typically refer to industry or commodity classifications that differ from the SALTER industry or commodity classification. For example, the statistic published for private consumption could be a vector of four numbers adding up to total private consumption, distributed over headers AI05 (domestic) and AI06 (imported), each vectors of 37 numbers. Published statistics are distributed over SALTER industries or commodities before being shared out between imported and domestic sources.

Disaggregating the published statistics to the SALTER level requires some estimation. The economic structures of Hong Kong and the ROW are estimated by identifying another SALTER country or region of similar industrial or demand structure. For example, Singapore is identified as being similar in industrial structure to Hong Kong. The *proxy* industry structure is used to disaggregate the published statistics. For example, if payments to labour in Hong Kong are published for SALTER industries A and B as a single figure, the proxy (ie. Singapore) database is used to estimate the share of payments to labour by industry A. The Singapore share allocates the Hong Kong labour statistic across SALTER industries A and B in the Hong Kong database.

The published statistics do not report data for some SALTER industries in Hong Kong and the ROW. The value of GDP at factor cost (total value added) serves as a data source when a SALTER industry is not represented at all by the published statistics. For example, the value of paddy rice output in Hong Kong is omitted from the published statistics, so the value of rice output is estimated as a portion of total value added. Specifically, we find the ratio of the omitted statistic to total value added as defined by the proxy industry structure. This ratio is multiplied by the Hong Kong or ROW total value added to yield the estimated statistic. For example, the value of paddy rice output in Hong Kong is found by multiplying the proxy (ie. Singapore) ratio of paddy rice output to total value added (ie. GDP at factor cost) by the Hong Kong figure for GDP.

Each element of each header array in Figure 1 is assigned a value estimated from the data available. This value is initially assigned without regard to other values in the database. As a result, the database will be internally inconsistent. For example, one would expect the value of output of industry 1 (paddy rice) to equal the value of paddy rice absorbed domestically, plus exports of paddy rice. The paddy rice elements of headers AI03-AI11 are assigned values independently of values assigned to the paddy rice elements of header AI02, AI12-AI17. That is, column values (industry costs) are assigned independently of row values (commodity sales).

Total costs (columns) and total sales (rows) are reconciled for an industry and commodity by repeatedly scaling the rows and columns in question. For the Rest of the World individual industry values of total costs are assumed to be correct, and commodity values of total sales are scaled up or down to meet the industry-defined *target values*. For Hong Kong the larger of the row and column total is used as the target value for each industry and commodity.

If total costs of the oil industry exceed total sales of oil, for example, then for either region each element of the domestic oil sales row vector would be scaled up to equate total sales and costs in oil. However, by scaling each row of the total sales headers, the sum of total costs will change because of the changes in header AI01. The target values for total costs and total sales are attained by repeated scaling of the rows and then the columns. This procedure is known as a RAS (UN 1973). The RAS can be modified by deliberately excluding elements of the database from the scaling process. This is useful when the unmodified RAS greatly distorts specific values in the database. Details of the RAS procedure are described in Chapters 2 and 3.

2 THE HONG KONG SINGLE REGION DATABASE

The Hong Kong single country database is derived from the industry structure of Singapore and national accounts statistics for Hong Kong. The industry structure of Singapore is assumed to be reasonably representative of Hong Kong's because both are small, trading economies with high values of imports and exports relative to GDP. This chapter explains how the Hong Kong database is estimated from the available data.

The order of construction

The strategy to estimate data for the Hong Kong database is to collect economic statistics, translate the statistics to SALTER classification and allocate the translated statistics to elements of the Hong Kong single-region database. The Hong Kong database evolves in four stages. Stage one deals with the collection of statistics published for Hong Kong by the World Bank and the United Nations. Stage two uses the industry structure of Singapore to allocate the published statistics across SALTER commodities and industries. Stage three reallocates the statistics to individual elements of the Hong Kong database and stage four ensures that the information in the database is consistent. The sections that follow describe each stage of the data estimation process in detail.

Collecting the published statistics

Table 2 lists information about the statistics collected for Hong Kong, referred to in this section as the published statistics. The first and second columns of Table 2 show the names and sources of the statistics collected. The third column of Table 2 shows the form in which the statistics are reported by the source. The fourth column provides a concordance between each source statistic and their associated SALTER classifications. The published statistics of Table 2 appear in header array format in the file HK.DAT. HK.DAT is created by the GEMPACK (Codsi and Pearson 1988) program MODHAR and the input file HK.STI.

Table 2: **Sources of statistics for Hong Kong**

Statistic	*Source*	*Observations*	*Corresponding SALTER categories*
		6 commodities:	**Commodities:**
Private consumption	*UN (1990c)*	food	1-4 and 13-16
in 1988 $HK	*Table 2.5*	beverages & tobacco	17
		clothing	18-20
		rent, fuel & power	9-11, 23 and 37
		transport & communication	29 and 34
		miscellaneous commodities	5-8, 12, 21-22, 24-28, 30-33, 35-36
		18 industries:	**Industries:**
Output	*UN (1991c)*	mining	9-12
	Column 8	food	13-16
		beverages and tobacco	17
Labour	*UN (1991c)*	textiles	18
	Column 4+5	wearing apparel	19
		leather and footwear	20
Value added	*UN (1991c)*	wood and furniture	21
	Column 9	paper and products	22
		petroleum and coal products	23
(Units in 1988 $HK)		chemicals, rubber and plastic	24
		non-metal products	25
		iron and steel	26
		non-ferrous metals	27
		metal products	28
		transport equipment	29
		machinery n.e.c.	30
		other industries & prof. goods	31
		electricity, gas and water	32
		3 industries:	**Industries:**
Value added	*UN (1990c)*	construction	33
in 1988 $HK	*Table 1.10*	transport, storage & comm.	34
		wholesale & retail trade, finance, insurance & real estate, community & personal services.	35
Government consumption in 1988 $HK	*UN (1990c) Table 1.1*	1 commodity	All SALTER commodities
Investment in 1988 $HK	*UN (1990c) Table 1.1*	1 commodity	All SALTER commodities

Table 2: **Sources of statistics for Hong Kong** (continued)

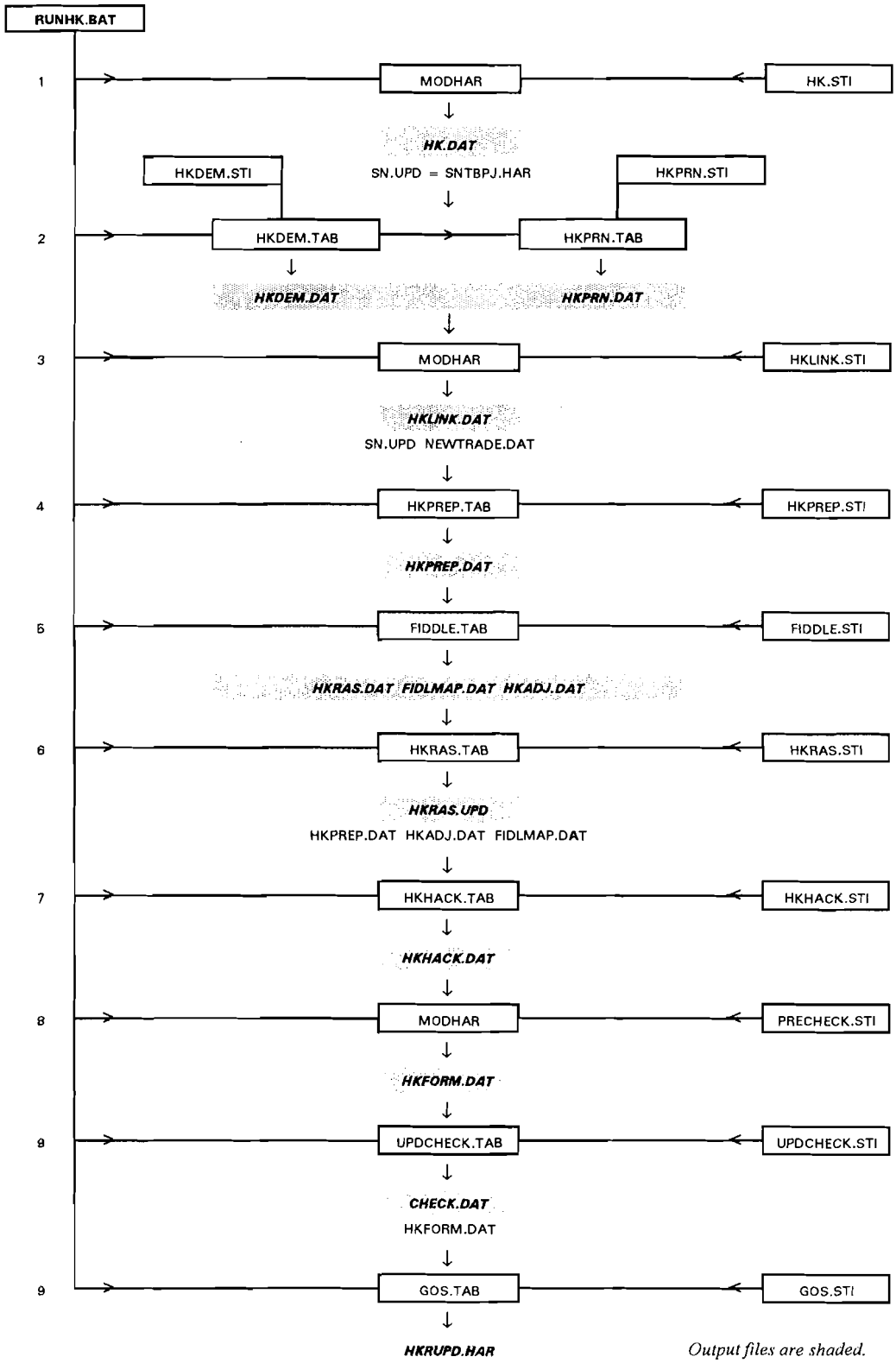
Statistic	*Source*	*Observations*	*Corresponding SALTER categories*
GDP in 1988 $US	*World Bank (1990)*	Hong Kong	All SALTER commodities/ industries
GDP in 1988 $HK	*UN (1990c) Table 1.1*	Hong Kong	All SALTER commodities/ industries
Share of GDP by sector (1980-89 average)	*World Bank (1991)*	**4 sectors:** agriculture mining manufacturing services	**Industries:** 1-8 9-12 13-31 32-37

Disaggregating the published statistics

As evident in Table 2, the published statistics do not match the standard SALTER classification. The published statistics refer to industry or commodity classifications that are generally more aggregated than the SALTER classifications. Hence we need to distribute each published statistic over the SALTER industries or commodities it encompasses. The SALTER classification figures produced by this process are private consumption by commodity, government consumption by commodity, investment demand by commodity, industry output by industry, value added by industry and payments by industry to capital, land and labour.

Figure 2 illustrates the progression of files used to create the Hong Kong database. Appendix B holds a printed copy of each file. As already mentioned, the input file HK.STI directs MODHAR to convert the published statistics into header array format (file HK.DAT). Next, the file HKPRN.TAB computes industry values for output and payments to capital, labour and land. HKPRN.TAB also determines value added by industry, but this information is only used within the file and is not reported as an output. The file HKDEM.TAB uses the published statistics in HK.DAT to disaggregate private consumption, government consumption and investment across SALTER commodities. The remaining files in Figure 2 are referred to later in the chapter.

Figure 2: **Files that create the Hong Kong single region database**



Output files are shaded.

HKPRN.TAB first estimates value added by SALTER industries in Hong Kong for use in the remainder of the file. The advantage of having figures for value added available is that we know the relative size of each industry in the overall economy. Equation 1 describes the general calculation of value added for the SALTER industry i'. The prime superscript in equation (1) denotes a specific (single) industry. An industry without a superscript can include any number of industries of that type.

$$\hat{VA}(i',HK) = \frac{VA(WB',HK)}{fGDP(HK)} \cdot \frac{VA(UN',HK)}{\sum_{UN \in WB'} VA(UN,HK)} \cdot \frac{VA(i',Sin)}{\sum_{i \in UN'} VA(i,Sin)} \cdot GDP(HK) \qquad (1)$$

where $\hat{VA}(i',HK)$ = value added estimated for the single SALTER industry i' in Hong Kong (1988 $USm)

$VA(WB',HK)$ = value added reported by the World Bank for sector WB' encompassing SALTER industry i'

$fGDP(HK)$ = Hong Kong GDP in $HKm

$VA(UN',HK)$ = value added reported by the United Nations for the UN industry UN' encompassing SALTER industry i'

$VA(i',Sin)$ = value added for SALTER industry i' in the Singapore database

$GDP(HK)$ = Hong Kong GDP in 1988 US$m

The published statistics for Hong Kong appear in equation (1) as World Bank and United Nations statistics for value added, expressed as $A(WB',HK)$ and $A(UN',HK)$. The value added terms on the right hand side proceed in descending order of aggregation. That is, the World Bank statistic includes one or more United Nations industries, and the United Nations statistic includes one or more SALTER industries. The World Bank source reports the value added statistic for four sectors and the United Nations reports the value added at a more disaggregated level but not to the level required in SALTER.

The first term on the right hand side is the share in GDP of value added for the World Bank sector WB', where the numerator and denominator are expressed in $HK. Note that the published statistics report the first term as a ratio for each World Bank sector of the economy, as shown in the last row in Table 2. The second term decomposes the World Bank figure into its component UN industries. Specifically, it is the share of UN industry UN' in all UN industries belonging to the World Bank sector WB'. The third term uses SALTER value

added data for Singapore to disaggregate the United Nations industry aggregate. Together the first three terms estimate the share in GDP of value added for SALTER industry *i'*. The last term is Hong Kong GDP expressed in 1988 US$m and is used with the share terms to find the estimated value added for industry *i'* in 1988 US$m. The Singapore value added term in equation (1) is omitted if the United Nations industry *UN'* corresponds exactly to the SALTER industry *i'*.

Two Hong Kong industries, numbers 5 (wool) and 23 (petroleum and coal products), have no published data available, and data from the Singapore database cannot be used as a substitute. The Singapore database has no information for industry 5. The industrial structure of industry 23 is unsuitable because of the relatively high values of intermediate usage and exports, reflecting Singapore's substantial production of petroleum. Value added for these industries is set to unity in Hong Kong so that an industry structure can be established prior to the actual values being assigned. That is, output, capital and labour are found as ratios of a unitary value added and later scaled up to the desired level, as described later in the chapter. The Singapore database is the SNTBPJ.HAR version of the Singapore input-output table (see Brown *et al* 1993).

HKPRN.TAB uses the estimates of value added by industry to find industry values for output, labour, capital and land. Equation (2) describes the calculation of output and labour for mining, food, non-food manufactures and industry 32 (electricity, gas and water). United Nations data on output is available for these industries.

$$\hat{OL}(i', HK) = \frac{OL(UN', HK)}{VA(UN', HK)} \cdot \hat{VA}(i', HK) \qquad (2)$$

$\hat{OL}(i', HK)$ is the estimated value of output *or* labour for the SALTER industry *i'* in Hong Kong. United Nations data provides the proportion of output or labour to value added for each UN industry *UN'*. Where more than one SALTER industry belongs to the UN industry aggregate, we assume that the component SALTER industries have the same ratio of labour or output to value added. This is the case for the individual SALTER mining and food industries.

Equation (3) calculates the values of labour and output for agriculture, services and industries 5 and 23 by using the industry structure of Singapore.

$$\hat{OL}(i',HK) = \frac{OL(i',Sin)}{VA(i',Sin)} \cdot \hat{VA}(i',HK) \tag{3}$$

HKPRN.TAB calculates the value of capital plus land as value added less labour for each SALTER industry. The total is separated into estimates of capital and land by applying the same capital to land ratios found in the Singapore data. That is, land usage is zero for non-agricultural industries and equal to half the value of capital for the agricultural industries.

The file HKDEM.TAB disaggregates the statistics for private consumption, government expenditure and investment. Equation (4) specifies the method used to estimate private consumption for each SALTER commodity *i'* in Hong Kong.

$$\hat{C}(i',HK) = C(UN',HK) \cdot \frac{C(i',Sin)}{\sum_{i \in UN'} C(i,Sin)} \cdot \frac{GDP(HK)}{fGDP(HK)} \tag{4}$$

where

$\hat{C}(i',HK)$	=	value of consumption estimated for the single SALTER commodity *i'* in Hong Kong (1988 $USm)
$C(UN',HK)$	=	value of consumption reported by the United Nations for the UN commodity *UN'* encompassing SALTER commodity *i'*
$C(i',Sin)$	=	value of consumption for SALTER commodity *i'* in the Singapore database
$fGDP(HK)$	=	Hong Kong GDP in 1988 $HKm
$GDP(HK)$	=	Hong Kong GDP in 1988 $USm

The consumption structure of Singapore guides the distribution of the United Nations aggregate statistics across SALTER commodities. The United Nations consumption statistics represent all SALTER commodities in aggregates, as shown in Table 2. The second term on the right hand side of equation (4) determines what part of the total United Nations statistic can be assigned to a particular SALTER commodity. The third term translates the United Nations consumption statistic from $HKm to 1988 $USm. Equation (4) also explains the method used to estimate government expenditure in Hong Kong for each SALTER commodity. A single United Nations government expenditure figure is available and it is disaggregated using shares from the Singapore database.

HKDEM.TAB estimates investment figures for Hong Kong by assuming that the Singaporean structure of investment applies to Hong Kong. No special cases apply when disaggregating investment to the SALTER classification.

Assigning values to the database headers

Referring to Figure 2, MODHAR combines the output from HKPRN.TAB and HKDEM.TAB using the input file HKLINK.STI. The output file, HKLINK.DAT, contains the estimates of capital, land, labour, output, consumption, government expenditure and investment in the SALTER classification. HKPREP.TAB distributes each estimate across the database headers described by Figure 1 and listed in Table 6.

Intermediate inputs occupy headers AI01 (domestic) and AI02 (imported). The estimated value of output less labour, capital and land determines the total usage of intermediates for each industry. HKPREP.TAB then disaggregates intermediate usage by commodity. Data on intermediate usage in the Singapore database is used to disaggregate each Hong Kong industry's intermediate usage across commodities. That is, the share of each SALTER commodity in any industry's total intermediate usage is assumed to be the same in Hong Kong and Singapore. HKPREP.TAB also allocates intermediate inputs across domestic and imported sources. Again the industry structure of Singapore guides the decomposition; the Singaporean shares of domestic to imported intermediates are applied to Hong Kong. In the Hong Kong database, elements of intermediate usage equal to zero are changed to 0.1 for the services industries. This is necessary to ensure that the arrays of intermediate usage in services are flexible enough for the RAS procedure, described below.

HKPREP.TAB divides the SALTER estimates of consumption, investment and government expenditure into domestic and imported components. This process is the same as for domestic and imported intermediate inputs. That is, the ratio of domestic to imported sources is assumed to be the same as in the Singapore database. The disaggregated estimates occupy headers AI03 to AI08. Headers for the primary factors labour (AI13), capital (AI14) and land (AI15) are transferred directly from the output of HKPRN.TAB as held in HKLINK.DAT.

UN based international trade data in SALTER's trade database (NEWTRADE.DAT) dictate the value of exports by SALTER commodity (see Hanslow 1993). In this database Hong Kong exports exclude re-exports, which have been converted back into direct trade between the ultimate source and destination regions (Hambley 1993). Note that the trade database is expressed in thousands of US$ and the Hong Kong database is expressed in millions of US$. Thus the value of exports is divided by 1000 before being transferred to

the export header (AI11). As mentioned above, all elements of each tax header are set to zero by assumption.

Recall that HKPRN.TAB sets the value added for industries 5 and 23 to unity, so the estimates of output, labour and capital for these industries must be revised. The trade database gives a value of domestic exports for these industries. Thus Hong Kong requires output from these industries. Output is estimated as 1.5 times the value of exports to allow for any intermediate usage and final demand for the domestic product.

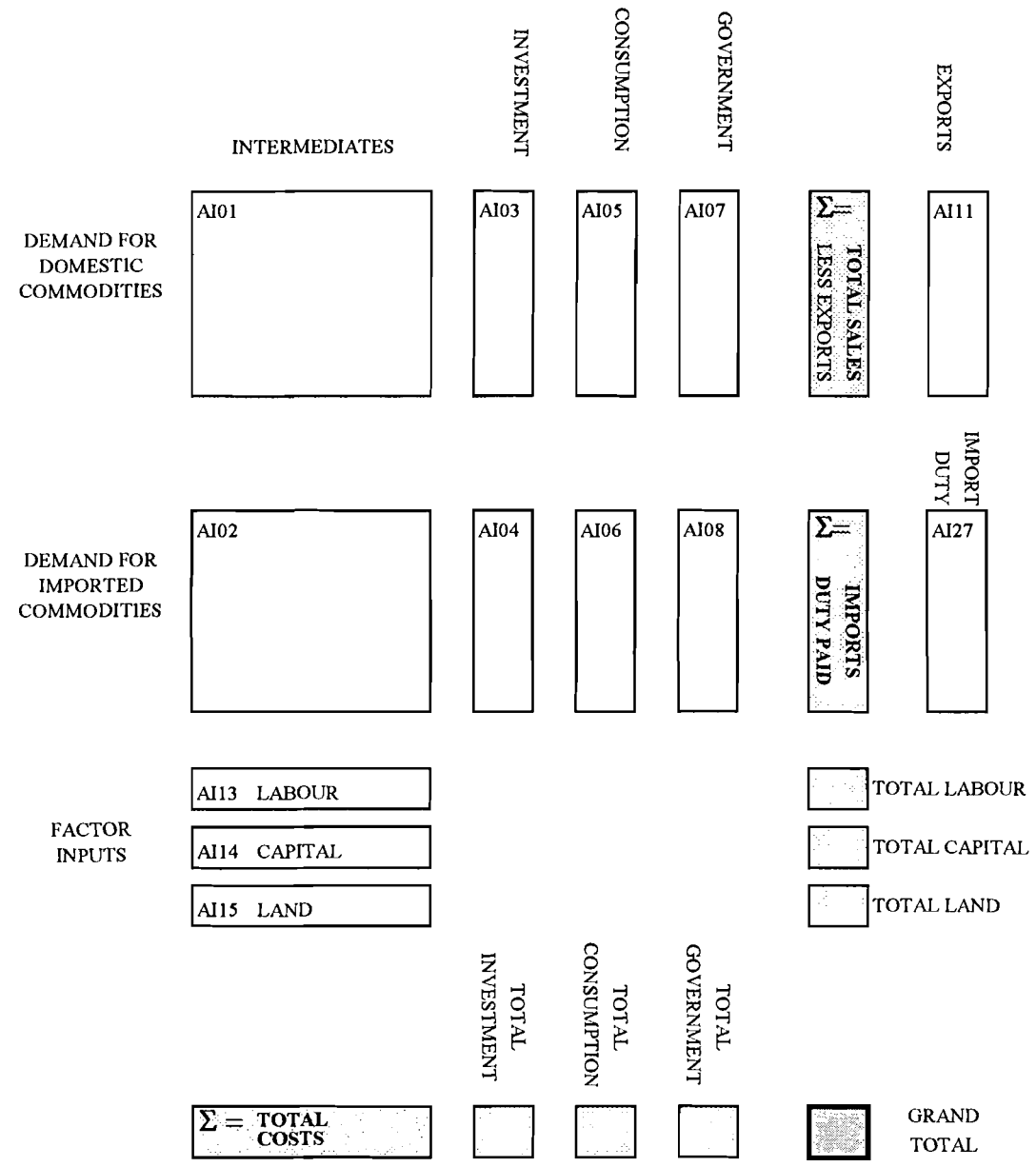
The RAS procedure

An accounting identity common to all SALTER databases is that total costs equal total sales (see Figure 1), or equivalently GDP from the income side (GDP at factor cost plus indirect taxes) must equal GDP from the expenditure side (GDP at market prices). Equality of total costs and total sales is not ensured during the estimation process. The function of the RAS is to enforce this identity, thus *balancing* the database. Appendix B lists the TABLO code of the RAS program used to balance the Hong Kong database.

Figure 3 describes the schematic form of a RAS table. Tax terms (all zero) have been omitted for simplicity. Exports appear in the margin because they are excluded from the RAS to prevent the export values from changing. The sums down the columns, shown as the last row of Figure 3, are the column totals. The row totals are the sums across the rows of Figure 3, shown as the column of shaded boxes. It is true of any database, balanced or otherwise, that the row totals and the column totals will sum to a common grand total. This is true because the sum of the row and column totals simply adds up every value in the database to a single number.

The RAS process alters the value of elements of the header arrays to conform to the total sales equals total costs constraint. Balancing the Hong Kong database involves a modified RAS process. The modified RAS allows the elements of some headers to be excluded from the RAS. This is necessary since the unmodified RAS changes elements in the database enough to significantly distort the industrial structure of the database. The elements suffering large changes are omitted by the modified RAS. The modified RAS has four stages. The first (HKPREP.TAB) prepares the data in HKLINK.DAT so that a RAS table can be constructed. The second (FIDDLE.TAB) specifies the elements to be excluded from the RAS. The third (HKRAS.TAB) is the RAS proper and the fourth (HKHACK.TAB) restores the excluded elements to the database.

Figure 3: **The RAS table**



The trade data in HKLINK.DAT and NEWTRADE.DAT originate from different sources and thus do not agree. For example, the total value of imports derived from the data in HKLINK.DAT differs from the value of imports reported by NEWTRADE.DAT. The RAS alters one data set to agree with the alternative set. In particular, the value of imports for each commodity reported by NEWTRADE.DAT is taken to be correct for Hong Kong. The NEWTRADE.DAT value of imports by commodity becomes a *target* value and the RAS alters the component values of imports (headers AI02, AI04, AI06, AI08) reported by HKLINK.DAT. HKPREP.TAB chooses the set of targets from the list in Table 3. Target values of exports are also provided by NEWTRADE.DAT to ensure trade flow consistency between single region databases.

Table 3: **The target set for Hong Kong's RAS procedure**

Target name	*Source of target values*	*Row (by commodity) or column (by industry) targets*
Total Consumption	HKDEM.TAB	Column
Total Government	HKDEM.TAB	Column
Total Investment	HKDEM.TAB	Column
Output (Total Costs)	HKPRN.TAB	Column
Output less Exports (Total Sales less Exports)	HKPRN.TAB and NEWTRADE.DAT	Row
Imports, Duty Paid	NEWTRADE.DAT	Row
Total Labour	HKPRN.TAB	Row
Total Capital	HKPRN.TAB	Row
Total Land	HKPRN.TAB	Row

The non-trade header arrays in the RAS table are constructed from the header arrays in HKLINK.DAT. Sales data in the RAS table originate from HKPRN.TAB and cost data originate from HKDEM.TAB. Hence, total costs by industry disagree with total sales by commodity. HKPREP.DAT selects the larger of the two totals as the target total for the RAS process. If the row total is larger, then the totals for consumption, investment and government usage are scaled up to equalise the grand totals in Table 3. HKPREP.TAB scales up the totals for labour, capital and land if the column total is larger. This yields a consistent set of targets, represented by the shaded boxes in Figure 3.

The file FIDDLE.TAB uses a map of the same dimensions as the RAS table, so it is schematically represented by Figure 3. Elements to be excluded from the RAS are set equal to unity in the map. All other elements of the map are zero. The RAS table is written by FIDDLE.TAB, the map setting to zero the elements to be excluded from the RAS. Note that the RAS process itself cannot change zero elements of the RAS table to be non-zero. This is why HKPREP.TAB changes intermediate usages with values of zero to values of 0.1 in the pre-RAS (unbalanced) database. By allowing all intermediate usages to change, the RAS can arrive at a solution that may otherwise involve significantly distorting the industrial structure. Note that this practice is only applied to the Hong Kong database.

The file HKRAS.TAB is the RAS proper. Referring to Figure 3, the RAS imposes equality between total costs and total sales, subject to the target totals defined by HKPREP.TAB. HKRAS.TAB first scales the elements of each row by a row-specific constant to arrive at the target value for that row. Next the process is repeated for each column, but in doing so, the sums of the rows are displaced from the target values. The row and column scaling process is repeated often enough to ensure that the last column scaling operation leaves the row totals almost unchanged from their target values. This process takes about 20 to 50 steps, where a step is defined as one row operation and one column operation. Figure 3 is further explained in Chapter 3.

The elements excluded from the RAS by FIDDLE.TAB are also subtracted from the target totals by FIDDLE.TAB. After the RAS, the file HKHACK.TAB completes the Hong Kong database by restoring the elements excluded from the RAS by FIDDLE.TAB. The subsequent MODHAR operation and UPDCHECK.TAB check the completed Hong Kong database for balance. The MODHAR operation adds missing headers (all zero) to the Hong Kong database so that UPDCHECK.TAB will run successfully. UPDCHECK.TAB outputs a file containing a header named PDIF. This gives the percent difference between total costs and total sales by industry, by which the success of the RAS can be judged. The final file, GOS.TAB, ensures that the agricultural industries have equal values of land and capital inputs.

3 THE REST OF THE WORLD SINGLE REGION DATABASE

The rest of the world (ROW) is defined as all sub-regions that are not encompassed by the 15 explicitly defined SALTER regions: Australia, New Zealand, Canada, United States, Japan, Korea, EEC, Indonesia, Malaysia, Philippines, Thailand, Singapore, China, Hong Kong, and Taiwan. The ROW single region database is more complex in construction than the single country database for Hong Kong, but follows the same principles. Appendix B holds all copies of the files required to create the ROW single country database.

The order of construction

Six sub-regions define the ROW region: Africa, the Middle East, some Asian and Pacific countries (known as *South Asia* in the text), Latin America, West Europe excluding the EEC countries and the formerly Centrally Planned Economies (CPE) excluding China. The countries included in each sub-region are as defined in Table 1 of UN (1991a), with the following exceptions. South Asia includes the countries defined in the source as Eastern, Southeastern and Southern Asia and Oceania, but excludes the countries already represented by SALTER. Countries under the heading *Eastern Europe and USSR* define the CPE sub-region. Latin America includes the Caribbean nations. West Europe is defined under the heading *Other Europe* in the source.

The ROW database is built in four distinct stages. Stage one collects published statistics for countries chosen to represent each ROW sub-region and relates them to proxy industry structures. The proxy industry structure is a SALTER country thought to be similar in industrial structure to the sub-region representative country in question. Stage two derives basic economic statistics for the ROW from the published statistics. Stage three decomposes the economic data into the database categories, or header arrays, shown in Figure 1. Stage three also assigns values to database header arrays that are not obtained from the published statistics. Stage four ensures that information in the database is consistent.

Stage one selects a representative country from the countries within each sub-region to indicate the typical economic structure of all countries included in the sub-region. Each representative country is then linked to the SALTER country that best describes its structure of industry. The SALTER country approximates the representative country when published statistics for the representative

country are unavailable. The sub-regions, representative countries and proxy countries are defined in Table 4.

Table 4: **Representative and proxy countries for the Rest of the World**

sub-region	*representative*	*proxy*
Africa	Zimbabwe	Thailand
Middle East	Iran	Thailand
South Asia	India	Thailand
Latin America	Chile	Korea
CPE	Hungary	Korea
West Europe	Sweden	EC

Stage two produces basic economic statistics for each sub-region in the ROW. GDP data for the sub-regions, published statistics for the representative countries and pre-existing SALTER data for the proxy countries determine economic statistics for each sub-region. The statistics produced by this process are private consumption by commodity, government consumption by commodity, investment demand by commodity, industry output, value added by industry and payments by industry to capital and labour. Summing these statistics over all sub-regions in ROW produces data for the ROW. This process of generating basic economic data for ROW is explained in the next section.

Stage three of the process decomposes the estimated economic statistics into header arrays of the ROW database. The complete ROW database has a structure equivalent to that of the other SALTER single region databases represented by Figure 1. The estimated economic statistics for ROW exist in aggregate form as output from the previous stage. Stage three distributes single statistics for ROW over several database headers. The industry structure of the proxy countries is vital to this stage of the process.

Stage four ensures that information in the ROW database is consistent. In terms of Figure 1, total sales in the database should agree with total costs. When this condition is satisfied the database is said to be *balanced* (see the previous chapter). The following sections explain further each stage of the database construction process.

Collecting the published statistics

An informal approach is used to decide the representative and proxy countries for each sub-region. A representative country is selected on the basis of data availability and similarity in industrial structure to countries in the sub-region for which the representative country acts. A proxy country is the SALTER region considered as closest in industrial structure to the representative country.

We collect statistics for the sub-regions and representative countries. Data for the SALTER proxy countries are available from existing single region databases. GDP is the only statistic sought at the sub-region level. GDP data are sourced from United Nations (1991*a*, Table 1). Statistics collected for the representative countries are summarised in Table 5.

The GEMPACK data manipulation software program MODHAR and input file ROWDATA.STI put the published statistics collected for the ROW sub-regions and representative countries into header array format. Appendix B contains a copy of ROWDATA.STI, which lists the published statistics used as input. RW92.DAT holds the header array output of the MODHAR operation. Data for the proxy countries exist in header array format in the SALTER single region databases of Thailand, Korea and the European Community (Brown *et al* 1993).

Table 5: **Sources of statistics for representative countries in the Rest of the World**

Statistic	*Source*	*Observations*	*For representatives*
		14 commodities:	
Government consumption in domestic currency	UN (1991b) Table 2.3	general public services defence public order and safety education health social security and welfare housing recreation fuel and energy agriculture other mining, manufacturing and construction transport and communication other functions	Zimbabwe (1986), Iran (1987), India (1987), Chile (1986), Sweden (1987)

Table 5: **Sources of statistics for representative countries in the Rest of the World** (cont)

Statistic	*Source*	*Observations*	*For representatives*
		8 commodities:	
Private consumption in domestic currency	UN (1991b) Table 2.5	food beverages and tobacco clothing and footwear fuel and power gross rent transport equipment communication other (household equipment, medical services, recreational services, miscellaneous goods and services)	Zimbabwe (1986), Iran (1987), India (1987), Chile (1986), Sweden (1987)
GDP in domestic currency (1988)	UN (1991b) Table 2	5 countries	Zimbabwe, Iran, India, Chile, Sweden
GDP in $USm (1988)	UN (1991a) Table 1	6 countries	Zimbabwe, Iran, India, Chile, Sweden, Hungary
		4 sectors:	
Value added in $USm (1988)	WB (1990) Table 3	agriculture manufacturing other industry services	Zimbabwe, India, Chile, Sweden, Hungary
		20 industries:	
Value added in domestic currency	UN (1990b) Volume I. For example, Column 6 in Zimbabwe	food beverages textiles wearing apparel leather and footwear wood products paper products chemical products petroleum and coal non-metal products n.e.c. iron and steel non-ferrous metals metal products transport equipment other machinery other manufacturing electricity, gas and water	Zimbabwe (1986), India (1985), Chile (1986), Hungary (1988), Sweden (1988)

Table 5: **Sources of statistics for representative countries in the Rest of the World** (cont)

Statistic	*Source*	*Observations*	*For representatives*
		6 industries:	
Output in domestic currency	UN (1990c) Table 4.1	agriculture mining electricity, gas and water construction wholesale and retail trade transport and communication	Zimbabwe (1984), Hungary (1987), Chile (1982), Sweden (1987)
		16 industries:	
Output in domestic currency	UN (1990b) For example, Column 5 in Zimbabwe	food beverages textiles wearing apparel leather and footwear wood products paper products chemical products petroleum and coal non-metal products n.e.c. iron and steel non-ferrous metals metal products transport equipment other machinery other manufacturing	Zimbabwe (1986), India (1985), Chile (1986), Hungary (1988), Sweden (1988)
		6 industries:	
Value added in domestic currency	UN (1990c) Table 4.1	agriculture mining electricity, gas and water construction wholesale and retail trade transport and communication	Zimbabwe (1984), Hungary (1987), Chile (1982), Sweden (1987)
		6 industries:	
Labour in domestic currency	UN (1990c) Table 4.3	agriculture mining electricity, gas and water construction wholesale and retail trade transport and communication	Zimbabwe (1984), Hungary (1987), Chile (1982), Sweden (1987)

Table 5: **Sources of statistics for representative countries in the Rest of the World** (cont)

Statistic	*Source*	*Observations*	*For representatives*
		6 industries:	
Capital (net operating surplus) in domestic currency	UN (1990c) Table 4.3	agriculture mining electricity, gas and water construction wholesale and retail trade transport and communication	Zimbabwe (1984), Hungary (1987), Chile (1982), Sweden (1987)
		16 industries:	
Labour in domestic currency	UN (1990b) For example, Column 4 in Zimbabwe	food beverages textiles wearing apparel leather and footwear wood products paper products chemical products petroleum and coal non-metal products n.e.c. iron and steel non-ferrous metals metal products transport equipment other machinery other manufacturing	Zimbabwe (1986), India (1985), Chile (1986), Hungary (1988), Sweden (1988)
		16 industries:	
Capital (Value added less labour) in domestic currency	UN (1990b) For example, Column 6 *less* Column 4 in Zimbabwe	food beverages textiles wearing apparel leather and footwear wood products paper products chemical products petroleum and coal non-metal products n.e.c. iron and steel non-ferrous metals metal products transport equipment other machinery other manufacturing	Zimbabwe (1986), India (1985), Chile (1986), Hungary (1988), Sweden (1988)

Disaggregating the published statistics

The published statistics held in RW92.DAT and the proxy data provide a basis for estimating the economic statistics for sub-regions in the ROW. The economic statistics to be estimated for each sub-region are private consumption, government consumption, investment demand, industry output and payments by industry to capital and labour. Each estimated statistic has 37 elements that correspond to the SALTER industries or commodities, as appropriate.

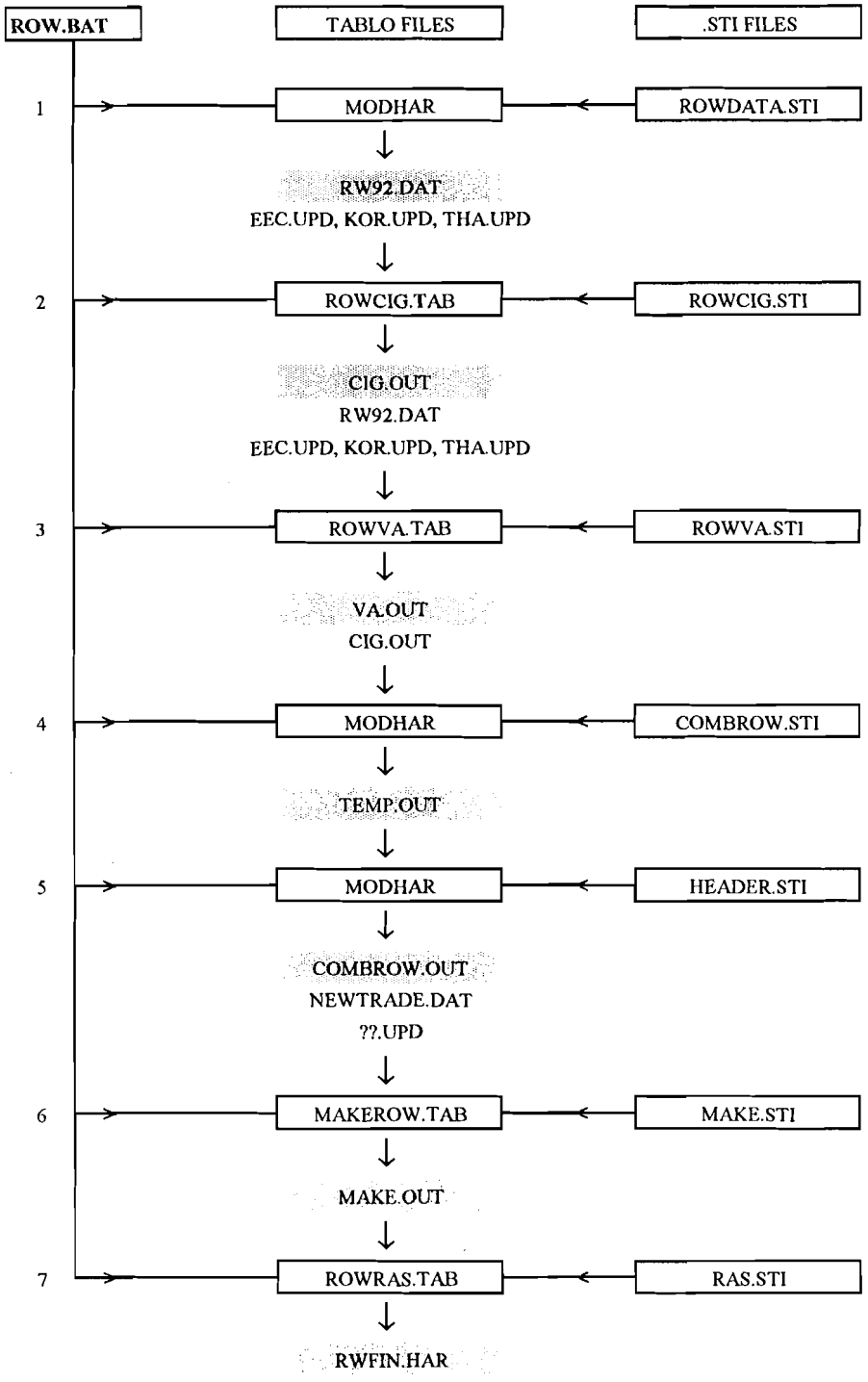
The estimated statistics appear by name in the published statistics (RW92.DAT) but refer to industry and commodity classifications that differ from the SALTER classification. The industrial structure of the proxy country determines how a statistic published for the representative country will be allocated across corresponding SALTER classifications. After estimating economic statistics in SALTER classification for the representative country, the estimates are scaled up to sub-regional level. Sub-regional values of a representative statistic result from transferring the share of the statistic in GDP at the representative level to that of the sub-regional level. Recall that GDP is available at both levels (representative and sub-regional) from the published statistics.

The file ROWVA.TAB finds value added, output, payments to capital and payments to labour. The file ROWCIG.TAB finds private and government consumption data and investment data for ROW subregions. Figure 4 shows all files used to make the final ROW database. Note that the files EEC.HAR, THA.HAR and KOR.HAR are single region databases at the ??.UPD stage (Brown *et al* 1993).

ROWVA.TAB first determines value added for each SALTER industry in each sub-region of ROW. Appendix C gives a complete description of ROWVA.TAB. ROWVA.TAB uses the value added statistic internally and does not report the statistic as output. Value added, calculated by SALTER industry, facilitates estimation of output, labour and capital. The generic calculation of value added for SALTER industry i' in representative r and sub-region s is:

$$VA(i',s) = \left[\frac{VA(WB',r)}{GDP(r)} \cdot \frac{VA(UN',r)}{\sum_{UN \in WB'} VA(UN,r)} \cdot \frac{VA(i',p)}{\sum_{i \in UN'} VA(i,p)} \right] \cdot GDP(s) \qquad (5)$$

Figure 4: **Files that create the ROW single region database**



Output files appear shaded.

?? = AS, CN, NZ, US, JP, KR, EC, IN, MS, PH, SN, TH, CH, HK, TW.

where $VA(i',s)$ = value added estimated for the single SALTER industry i' in sub-region s (1988 $USm)

$VA(WB',r)$ = value added reported by the World Bank for the representative country r and sector WB' encompassing SALTER industry i'

$VA(UN',r)$ = value added reported by the United Nations for the representative country r and UN industry UN' encompassing SALTER industry i'

$VA(i',p)$ = value added for SALTER industry i' in the proxy country (1988 $USm)

$GDP(s)$ = Sub-region GDP in 1988 US$m

The prime superscript in equation (5) denotes a specific (single) commodity. A commodity without a superscript can include any number of commodities of that type. The right hand side of equation (5) proceeds in descending order of industry aggregation. World Bank value added data for the representative country is most highly aggregated. The United Nations valued added statistic is less aggregated in terms of industry definition. SALTER value added data for the proxy country are least aggregated. The first term of the right hand side is the share of value added in GDP for a World Bank industry in the representative country. The second and third terms of the right hand side decompose the World Bank industry share to a SALTER industry share of value added in GDP for the representative country. The final term of the right hand side translates the SALTER industry share into an estimated SALTER statistic for the sub-region. This assumes that the shares of value added in GDP at the representative and sub-regional levels are equal.

If SALTER commodity i' is not represented by the UN published statistics, the UN term is simply omitted from equation (5). The WB term is omitted if i' is not represented in the WB statistics.

Estimation of value added by SALTER industry at the sub-regional level involves some inaccuracy. The value added information is drawn from up to three independent sources and determined as a sub-regional share of GDP for each SALTER industry. The independence of the data sources used means that the shares of value added in GDP will probably not sum to unity.

ROWVA.TAB uses the estimates of value added to calculate the basic economic statistics M of industry output, payments to land and payments to capital. The benefit of having estimated value added for sub-regions is apparent in equation (6).

$$M(i',s) = \hat{VA}(i',s) \cdot \frac{M(PS',r)}{VA(PS',r)} \qquad (6)$$

where $M(i',s)$ = estimated statistic for single SALTER industry i' in sub-region s (1988 $USm)

$VA(i',s)$ = estimate of value added for industry i' in sub-region s

$M(PS',r)$ = economic statistic published for the representative country r and the UN or WB industry PS' that includes SALTER industry i'

$VA(PS',r)$ = value added published for the representative country r and the UN or WB industry PS' that includes SALTER industry i'

Equation (6) demonstrates how statistic M is estimated for SALTER industry i' in sub-region s. Value added is available for each published statistic PS, so the ratio of M to value added can be found for the representative country. When data for the representative country is not available, proxy country data is used instead. Multiplying the ratio by the estimate of value added for the sub-region yields a sub-regional estimate of the economic statistic. When the published statistic contains several SALTER commodities, as in the case of agriculture, the commodities will have common values of the M to value added ratio, but different estimated values of M for the sub-region. See Appendix C for full details.

Published statistics used in equations (5) and (6) above were not available for the South Asia and Middle East sub-regions of ROW. Data from the Thailand single region database substitutes for the published statistics, except for the SALTER mining industries (industries 9-12). The industrial structure of mining in Thailand is an inappropriate proxy for the South Asia and Middle East sub-regions. In particular, Thailand data produce estimates of oil (industry 10) and gas (industry 11) output that cause ROW output values to be less than the value of exports reported by NEWTRADE.DAT. Data from the Indonesia single region database are used as a proxy for mining data in the South Asia and Middle East sub-regions.

The file ROWCIG.TAB estimates ROW statistics for private expenditure, government consumption and investment. ROWCIG.TAB is independent of the data used and produced by ROWVA.TAB.

The published statistics do not provide investment data. Estimated values of investment depend on data for the proxy countries. Equation (7) specifies the method of estimating investment for the sub-regions. That is, data from the proxy countries is used in place of the missing published statistics.

$$I(i',s) = \frac{I(i',p)}{VA(i',p)} \cdot GDP(s) \tag{7}$$

The published statistics list private and government consumption statistics for representative countries. Table 5 indicates the statistics that are compiled by the United Nations. In the source, these figures are unavailable for Hungary. Estimates of private and government consumption for the CPE sub-region follow the principle of equation (7). That is, data are supplied by the proxy country (Korea).

Private and government consumption for the remaining sub-regions of the ROW depend on the published statistics, which are mostly reported in classifications more aggregated than the SALTER classification of commodities or industries. Data from the proxy countries guide the disaggregation of a UN consumption statistic. Equation (8) shows the procedure.

$$C(i',s) = C(UN',r) \cdot \frac{C(i',p)}{\sum_{i \in UN'} C(i,p)} \cdot \frac{GDP(s)}{fGDP(s)} \tag{8}$$

C(i',s) represents private or government consumption of SALTER commodity *i'* in sub-region *s*. The first term on the right hand side is the value of consumption in the representative country for a single UN commodity aggregate. The second term demonstrates the method that allocates consumption to SALTER commodities included by the UN aggregate in question. The third term translates the UN value of consumption, expressed in domestic currency, to $US. Appendix C gives a complete description of the calculations used in ROWCIG.TAB.

We have obtained the value of output, capital, labour, investment, private consumption, government consumption and investment for each SALTER industry or commodity in the ROW. The final output of each variable has dimensions (1*37) because the statistics calculated for the ROW are summed over each sub-region. The following section describes stage three of the construction process.

Assigning estimated statistics to database headers

This section explains how the estimated data from the previous stage are converted to the ROW single-region database. The content of the final ROW database is shown in Table 6 and represented schematically in Figure 1.

MODHAR and the input file COMBROW.STI combine the output from ROWVA.TAB and ROWCIG.TAB into a single file called COMBROW.DAT. COMBROW.DAT also includes the $US GDP of each sub-region in the ROW under header "gdp". Figure 4 illustrates the usage of files in the database creation process.

The file MAKEROW.TAB calculates the header arrays to be included in the ROW database. MAKEROW.TAB draws data from COMBROW.DAT and all existing single region databases. The single region database for Hong Kong is replaced with that of Singapore. Most tax terms (headers AI12, AI16-AI24) are set equal to zero. Headers AI28, AI29, AI31 and AI32 are uni-dimensional. Refer to Table 6 for a translation of header codes. Depreciation (AI31 = 578490.4) and income tax (AI28 = 643011.2) are taken from national accounts statistics. The figures in all SALTER single region databases are denominated in $USm. Transfers (AI29) and savings (AI32) are set to zero.

Headers AI13-AI15 are primary factor inputs. Labour is taken directly from the output of ROWVA.TAB as kept in COMBROW.DAT. For the non-agricultural industries, land is set equal to zero and capital is taken directly from COMBROW.DAT. The agricultural industries in SALTER use land, so capital from COMBROW.DAT is allocated between capital and land in the database. The share of land in the capital values from COMBROW.DAT depends on the share of land in industry land and capital as defined by all other single region databases. Currently the split is 50-50 in each region.

Intermediate inputs occupy headers AI01 and AI02 in the database. The value of output less payments to capital and labour in COMBROW.DAT determines the aggregate value of intermediate inputs. Intermediates are then disaggregated by commodity. This disaggregation is guided by the proportion of industry use of a commodity in total intermediate use of that commodity as reported in the other single region databases.

MAKEROW.TAB also allocates intermediates across imported and domestic sources. Intermediates are separated according to the average proportion of imported to total intermediate demand for all SALTER regions other than ROW.

Table 6: **Header names and codes for a standard SALTER input-output database**

TABLO name	*Dimension*	*Description*
AI01	37*37	Usage of domestically produced intermediate inputs
AI02	37*37	Usage of imported intermediates
AI03	37*1	Usage of domestically produced goods for investment
AI04	37*1	Usage of imported goods for investment
AI05	37*1	Usage of domestically produced goods for consumption
AI06	37*1	Usage of imported goods for consumption
AI07	37*1	Usage of domestically produced goods by government
AI08	37*1	Usage of imported goods by government
AI11	37*1	Exports
AI12	37*1	Non-commodity indirect taxes plus subsidies
AI13	37*1	Payments to labour
AI14	37*1	Payments to capital
AI15	37*1	Rental of land
AI16	37*37	Commodity taxes on domestically produced intermediates
AI17	37*37	Commodity taxes on imported intermediates
AI18	37*1	Commodity taxes on domestically produced consumption goods
AI19	37*1	Commodity taxes on imported consumption goods
AI20	37*1	Commodity taxes on domestically produced investment goods
AI21	37*1	Commodity taxes on imported investment goods
AI22	37*1	Commodity taxes on domestically produced government consumption goods
AI23	37*1	Commodity taxes on imported government consumption goods
AI24	37*1	Export tax
AI27	37*1	Import duty
AI28	1*1	Income tax
AI29	1*1	Transfers
AI31	1*I	Depreciation
AI32	1*1	Savings

The file MAKEROW.TAB also separates the final demands in COMBROW.DAT into domestic and imported components. A weighted share of imported demand in total (domestic and imported) demand is used, as defined

by the proxy countries. The weight for each proxy country is the ratio of the GDP of all representative countries covered by the proxy country to total GDP for the ROW. For example, the share of imported private consumption in total ROW private consumption is:

$$SHICON = \frac{Imported\ consumption("Thailand")}{Total\ consumption\ ("Thailand")} \cdot \frac{\sum_{\forall i\ with\ proxy = Thailand} ROWGDP_i}{\sum_{\forall i} ROWGDP_i}$$
$$+ \frac{Imported\ consumption("Korea")}{Total\ consumption\ ("Korea")} \cdot \frac{\sum_{\forall i\ with\ proxy = Korea} ROWGDP_i}{\sum_{\forall i} ROWGDP_i}$$
$$+ \frac{Imported\ consumption("EC")}{Total\ consumption\ ("EC")} \cdot \frac{\sum_{\forall i\ with\ proxy = EC} ROWGDP_i}{\sum_{\forall i} ROWGDP_i} \qquad (9)$$

Exports (AI11) for ROW are taken directly from the trade database (NEWTRADE.DAT). Note that the figures in the ROW database are expressed in $USm and the figures in NEWTRADE.DAT are expressed in thousands of $US. Also, the values of imports in the trade database are duty free. The value of import duty (AI27) is found by determining the simple average percentage import duty (by commodity) for SALTER regions other than ROW and multiplying this by the value of ROW imports net of duty.

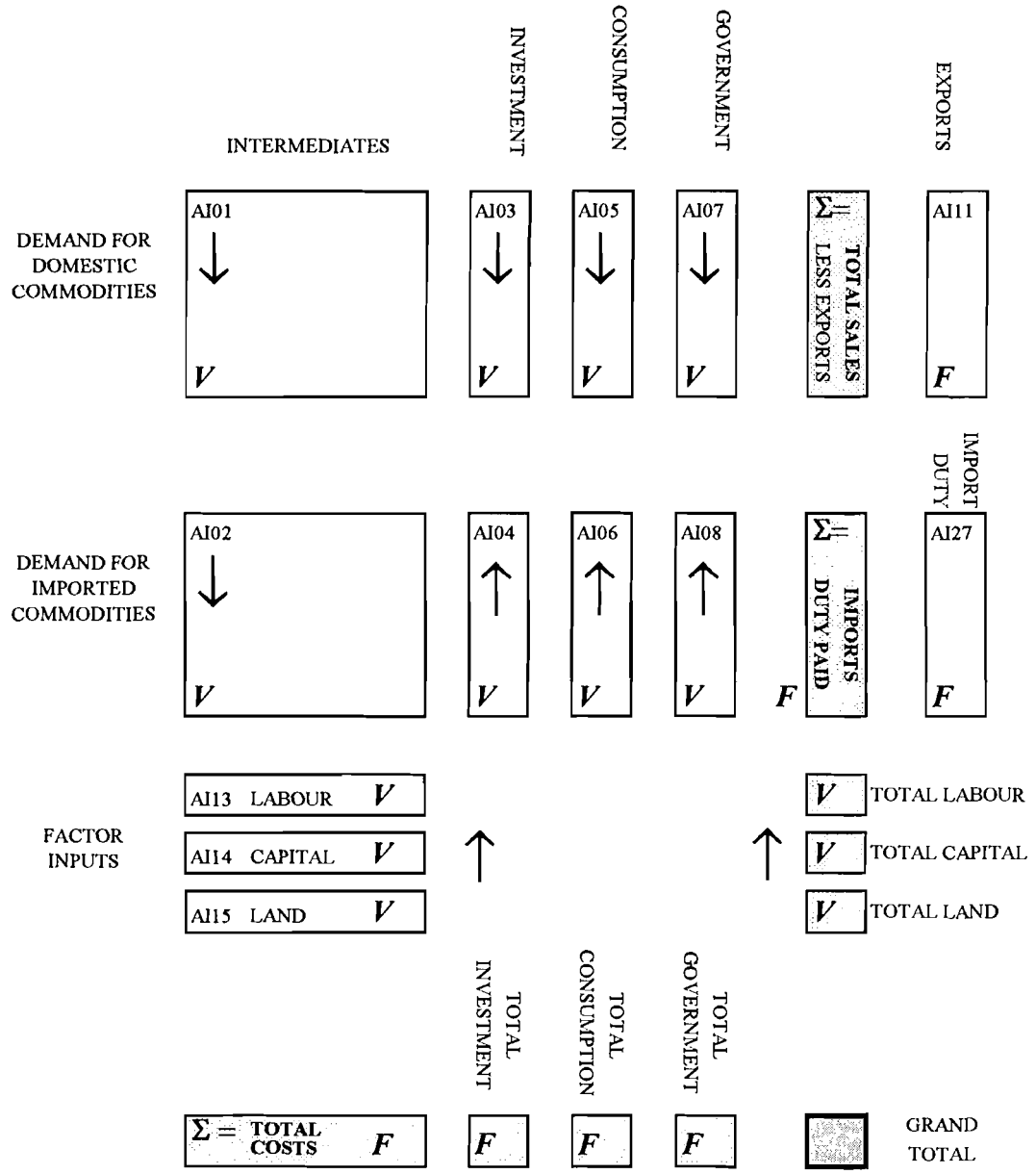
The output of MAKEROW.TAB contains all header arrays included in the final ROW database. However, the sum of total costs and total sales, shown in Figure 1, will not agree because they have been determined independently. The last step in creating the ROW database involves applying a RAS procedure to the output of MAKEROW.TAB that will equate total costs and total sales.

The RAS procedure

Equality of total costs and total sales is not ensured during the estimation process. An accounting identity common to all SALTER databases is that total costs must equal total sales. The function of the RAS is to enforce this identity, thus *balancing* the database. Appendix B holds the TABLO code of the RAS program used to balance the ROW database.

Figure 5 reproduces the schematic form of a RAS table. Recall that the sums down the columns, shown as the last row of Figure 5, are the column totals. The row totals are the sums across the rows of Figure 5, shown as the column of

Figure 5: **The RAS procedure**



shaded boxes. It is true of any database, balanced or otherwise, that the row totals and the column totals will add to a common grand total.

The RAS process equates costs and sales by changing values in the header arrays of the database. The trade database (NEWTRADE.DAT) dictates the value of exports in the ROW database. These cannot be altered so they are excluded from the RAS. Remaining in the RAS table are final and intermediate

demands and primary factor inputs. Figure 5 omits taxes, but in practice they remain in the table. However, the RAS will not change a zero value to be non-zero and all taxes are zero except import duties and income tax. The RAS alters the non-zero values remaining in the table to equate total sales less exports and total costs less exports.

The RAS process adjusts the database values with reference to a set of targets. A target set is simply the set of row totals or column totals, whichever is most appropriate. A total that is used as a target will not be changed by the RAS. For the ROW database, the set of column totals is chosen as the target set because these totals in the final database should agree with the published national accounts statistics. For example, total consumption is derived in ROWCIG.TAB from published statistics and thus should not be distorted by the RAS.

A target is implicitly defined for total sales (less exports) by selecting the column totals as the target set because the RAS equates total sales and total costs. The other components of the row totals are total imports duty paid and the total primary factors. The totals of primary factors are not fixed in the RAS. Total imports are fixed because they are taken from the trade database, which is predetermined. Figure 5 indicates the fixed *(F)* and variable *(V)* elements of the RAS table, the same breakdown as was used for Hong Kong. The more elements that are variable in a table, the easier is the adjustment of the table to satisfy the targets.

To illustrate the function of the RAS, say that the pre-RAS total costs are less than total sales. The RAS process will attempt to scale down the variable elements of the first row in Figure 5 to meet the target (total costs less exports), which is initially less than the initial row total (total sales less exports). Given the constraints imposed on the column totals, it is possible to discover how the RAS process will change the database. The RAS operates by scaling up the variable elements of each row (or column) by a constant number. The constant can be different for different rows (or columns) in the table. The arrows in Figure 5 show whether the *final* values in a header array will increase or decrease. Figure 5 is illustrative of what happens during the RAS, but simplifies somewhat from the actual process. See Appendix B for the TABLO code of the RAS.

Making total sales equal to the value of total costs causes the first row of Figure 5 to be scaled down. Assume that the second row of the table does not need scaling. That is, import targets are reached. The RAS then scales each column so that they meet the target totals again, having been displaced by the changes due to the previous row operation.

In this example, the columns are each multiplied by numbers greater than unity to compensate for smaller values in the first row of Figure 5. Each subsequent step of the RAS process involves scaling the rows and then the columns to meet the targets (the RAS for the ROW database is run in about 200 steps). The second step in the RAS example scales down the first row to meet the total sales less exports target again, and scales down the second row to meet the imports duty paid target. These row operations are a result of the column scaling in the first step. Next, a column operation restores the column totals to their target values by scaling up the first four columns of the figure. This process is repeated many times, resulting in the changes in total magnitude represented by arrows in Figure 5. In the figure, the net effect of the RAS is to balance the database at the cost of increasing the share of imports in final demand, decreasing the use of imported intermediates and increasing the use of primary factor inputs.

Sometimes elements in the database are highly distorted by the RAS. This is the case in the ROW with government consumption. The RAS inflates elements of domestic government consumption to values that are very high relative to the initial values, especially in agricultural commodities. A solution is to interrupt the RAS after it scales the row totals and before it scales the column totals. During the interruption, the shares of government expenditure in total government expenditure for each SALTER commodity are restored to their pre-RAS shares. Because each row is scaled up (or down) by a different multiplier, the shares of government consumption are changed by the row adjustment. The shares restored by the interruption are not disturbed by the subsequent column adjustment because every element in the government consumption column is multiplied by the same number.

Usually a RAS will take about 20 steps before total costs and total sales are almost equal. The interruption to restore shares of government consumption made the RAS less efficient in equating costs and sales, hence the use of 200 steps. Note that in the ROW it turned out that pre-RAS total costs were less than total sales by about one seventeenth, requiring the row totals to be scaled down to their target. The reader should be aware that the RAS program provided in Appendix B may break down if the target value is significantly lower than the initial total of the row or column. This can be overcome by selecting the larger of total sales and total costs as the target row or column, but for the ROW this would mean abandoning the desired targets in favour of less appropriate targets.

APPENDIX A: FILES USED TO CREATE THE HONG KONG SINGLE COUNTRY DATABASE

The TABLO code for all files listed in Figure 2 follows.

RUNHK.BAT

```
rem   this file uses a suite of programs
rem   to generate a database for hong kong
rem   each file serves a particular
rem   function as described below
up modhar <hk.sti >out0
rem   production structure
up hkprn < hkprn.sti > out1
rem   demand structure
up hkdem < hkdem.sti > out2
rem   link output from both procedures and
rem   prepare data for the RAS procedure
up modhar < hklink.sti > out3
up hkprep < hkprep.sti > out4
up fiddle < fiddle.sti > out5
rem   run the RAS with 100 steps
up hkras < hkras.sti > out6
rem   hack the ras table into its component parts
up hkhack < hkhack.sti > out7
rem   format the output to run it through
rem   total sales = total cost check and
rem   prepare for reporting table or inclusion
rem   in the Salter database
up modhar < precheck.sti > out8
up gos <gos.sti>out9
```

HK.STI

f
i
n
HK.dat
aw
t
r
1
6
n
34590
4241
48067
32244
17222
80553
HK1
Private consumption for Hong Kong
w
n
AW
t
r
1
1
n
25451
HK2
Hong Kong government consumption
W
n
AW
t
r
1
1
n
95041
HK3
Hong Kong investment
W
n
AW
t
r
1
1
n
55594
HK4

Hong Kong GDP in US$
W
n
AW
t
r
1
1
n
368177
HK5
Hong Kong GDP in $Hong Kong
W
n
AW
t
r
1
4
n
0.01
0.07
0.22
0.7
HK6
Hong Kong GDP by sector
W
n
AW
t
r
1
16
n
2203 12578 17835
802 951 4987 0 7341
663 209 225 5146 1615
13964 6239 9842
HK7
NFM Hong Kong Value Added UN industrial statistics
W
n
AW
t
r
1
2
n
248 1793
HK8
Mining and food Hong Kong Value Added UN industrial statistics
w

n
AW
t
r
l
3
n
16400 203624 30272
HK9
VA UN nat ac construction, private and transport services
W
n
AW
t
r
l
16
n
4556 44376 60806
2884 3408 14506 0 28455
4226 1395 1502 17235 2868
60448 30980 13557
HK10
NFM Hong Kong output UN industrial statistics
W
n
AW
t
r
l
2
n
449 6484
HK11
Mining and food man Hong Kong output UN industrial statistics
W
n
AW
t
r
l
16
n
443 7220 12969
584 603 2681 0 4268
383 108 130 3029 1200
7420 1909 1941
HK12
NFM Hong Kong labour UN industrial statistics
W
n
AW

t
r
1
2
n
59 986
HK13
NFM Hong Kong labour UN industrial statistics
W
n
ex
PAJ
25792
Hong Kong input data
**end
y

HKDEM.STI

hk.dat
sn.upd
hkdem.dat
checkdem.dat

HKDEM.TAB

```
!PURPOSE: TO FIND CONSUMPTION, GOVERNMENT CONSUMPTION, INVESTMENT FOR
HONG KONG!

                !FILE HKDEM.TAB!

!*****************************************************************!
!                    FILES                      !
!*****************************************************************!
FILE HKIO #FILE CONTAINING HK DATA#;

FILE SNIO #FILE CONTAINING SINGAPORE DATA#;

FILE (NEW) HKDEM #FILE CONTAINING DEMAND DATA FOR HONG KONG#;

FILE (NEW) CHECK # FILE CONTAINING SHARES#;

!*************************************************************!
!                    SETS                       !
!*************************************************************!

SET IND (IND1 - IND37) ;

SET HK(HONGKONG);

SET SING(SINGAPORE);

SET HKCD(HKCD1-HKCD6);

SET HKC1(IND1-IND4, IND13-IND16);
SUBSET HKC1 IS SUBSET OF IND;

SET HKC3(IND18-IND20);
SUBSET HKC3 IS SUBSET OF IND;

SET HKC4(IND9-IND11, IND23, IND37);
SUBSET HKC4 IS SUBSET OF IND;

SET HKC5(IND29, IND34);
SUBSET HKC5 IS SUBSET OF IND;

SET HKC6(IND5-IND8,IND12, IND21-IND22, IND24-IND28, IND30-IND33,IND35-IND36);
```

```
SUBSET HKC6 IS SUBSET OF IND;

SET NI(NIV1-NIV5);

SET NI1(IND33);
SUBSET NI1 IS SUBSET OF IND;

SET NI2(IND29);
SUBSET NI2 IS SUBSET OF IND;

SET   NI3( IND30);
SUBSET NI3 IS SUBSET OF IND;

SET   NI4(IND1- IND8);
SUBSET NI4 IS SUBSET OF IND;

SET   NI5(IND9-IND28,IND30,IND32-IND37);
SUBSET NI5 IS SUBSET OF IND;

!----------------------------------------------------------------------
                INPUT-OUTPUT DATABASE
 ----------------------------------------------------------------------!


COEFFICIENT (ALL,I,HKCD)                        HKCON(I)
!PRIVATE CONSUMPTION BY HONG KONG!;

COEFFICIENT (ALL,I,IND)                         DCON(I)
!DOM CON!;

COEFFICIENT (ALL,I,IND)                         ICON(I)
!IMPORTED CON!;

COEFFICIENT                                HKGOV
!HONG KONG GOVERNMENT CONSUMPTION!;

COEFFICIENT  (ALL,I,IND)                        DGOV(I)
!DOM GOVT!;

COEFFICIENT (ALL,I,IND)                         IGOV(I)
!IMPORTED GOVT!;

COEFFICIENT                               HKINV
!HONG KONG INVESTMENT!;

COEFFICIENT  (ALL,I,IND)                        DINV(I)
!DOM INVESTMENT!;

COEFFICIENT (ALL,I,IND)                         IINV(I)
!IMPORTED INVESTMENT!;

COEFFICIENT                                USGDP
```

```
!HONG KONG GDP IN US DOLLARS!;

COEFFICIENT                                HKGDP
!HONG KONG GDP IN HONG KONG DOLLARS!;

COEFFICIENT (ALL,J,IND)          SINLAB(J)
!SINGAPORE LABOUR!;


COEFFICIENT (ALL,J,IND)          SINCAP(J)
!SINGAPORE CAPITAL!;

!************************************************************
                  READING DATA BASE
 ************************************************************!
READ

  (ALL,I,HKCD)
  HKCON(I)
       FROM FILE HKIO HEADER "HK1";


  HKGOV
      FROM FILE HKIO HEADER "HK2";

  HKINV
     FROM FILE HKIO HEADER "HK3";


  (ALL,I,IND)
  DCON(I)
       FROM FILE SNIO HEADER "AI05";

  (ALL,I,IND)
  ICON(I)
       FROM FILE SNIO HEADER "AI06";

  (ALL,I,IND)
   DGOV(I)
        FROM FILE SNIO HEADER "AI07";

        (ALL,I,IND)
   IGOV(I)
        FROM FILE SNIO HEADER "AI08";

  (ALL,I,IND)
   DINV(I)
        FROM FILE SNIO HEADER "AI03";

  (ALL,I,IND)
   IINV(I)
        FROM FILE SNIO HEADER "AI04";
```

```
   USGDP
      FROM FILE HKIO HEADER "HK4";

   HKGDP
      FROM FILE HKIO HEADER "HK5";

 (ALL,J,IND)
SINLAB(J)
      FROM FILE SNIO HEADER "AI13";


      (ALL,J,IND)
SINCAP(J)
      FROM FILE SNIO HEADER "AI14";

!****************************************************************
            COEFFICIENTS
****************************************************************!

COEFFICIENT (ALL,I,IND)                    SINCON(I)
!PRIVATE CONSUMPTION BY SINGAPORE!;
FORMULA (ALL,I,IND)
SINCON(I)=DCON(I)+ICON(I);

COEFFICIENT (ALL,I,IND)                    SINGOV(I)
!SINGAPORE GOVERNMENT CONSUMPTION!;
FORMULA (ALL,I,IND)
SINGOV(I)=DGOV(I)+IGOV(I);

COEFFICIENT (ALL,I,IND)                    SININV(I)
!SINGAPORE INVESTMENT!;
FORMULA (ALL,I,IND)
SININV(I)=DINV(I)+IINV(I);

COEFFICIENT (ALL,J,IND)          GDPSIN(J)
!GDP FOR SINGAPORE AT FACTOR COST!;
FORMULA (ALL,J,IND)
GDPSIN(J)=SINLAB(J)+SINCAP(J);

!SHARES FOR CONSUMPTION!

COEFFICIENT (ALL,I,HKC1)           SC1(I)
!SHARES OF FOOD FOR HONG KONG!;
FORMULA (ALL,I,HKC1)
SC1(I)=SINCON(I)/SUM(G,HKC1,SINCON(G));


COEFFICIENT (ALL,I,HKC3)           SC2(I)
!SHARES FOR CLOTHING & FOOTWEAR!;
FORMULA (ALL,I,HKC3)
SC2(I)=SINCON(I)/SUM(G,HKC3,SINCON(G));
```

```
COEFFICIENT (ALL,I,HKC4)          SC3(I)
!SHARES FOR RENT, FUEL AND POWER!;
FORMULA (ALL,I,HKC4)
SC3(I)=SINCON(I)/SUM(G,HKC4,SINCON(G));

COEFFICIENT (ALL,I,HKC5)          SC4(I)
!SHARES FOR TRANSPORT & COMMUNICATION!;
FORMULA (ALL,I,HKC5)
SC4(I)=SINCON(I)/SUM(G,HKC5,SINCON(G));

COEFFICIENT (ALL,I,HKC6)          SC5(I)
!SHARES OF MISCELLANEOUS COMMODITIES!;
FORMULA (ALL,I,HKC6)
SC5(I)=SINCON(I)/SUM(G,HKC6,SINCON(G));

!SHARES FOR GOVERNMENT CONSUMPTION!

COEFFICIENT (ALL,I,IND)           SG1(I)
!SHARES FOR GOVERNMENT CONSUMPTION!;
FORMULA (ALL,I,IND)
SG1(I)=SINGOV(I)/SUM(G,IND,SINGOV(G));

 !COEFFICIENTS TO CALCULATE CONSUMPTION!

COEFFICIENT (ALL,I,IND)                CHK(I)
!CONSUMPTION OF FOOD COMMODITIES FOR HONG KONG!;
FORMULA (ALL,I,HKC1)
CHK(I)=SC1(I)*HKCON("HKCD1")*(USGDP/HKGDP);

!CONSUMPTION OF BEVERAGES AND TOBACCO!
FORMULA
CHK("IND17")=HKCON("HKCD2")*(USGDP/HKGDP);


!CONSUMPTION OF CLOTHING & FOOTWEAR COMMODITIES FOR HONG KONG!
FORMULA (ALL,I,HKC3)
CHK(I)=SC2(I)*HKCON("HKCD3")*(USGDP/HKGDP);

!CONSUMPTION OF RENT, FUEL & POWER COMMODITIES FOR HONG KONG!
FORMULA (ALL,I,HKC4)
CHK(I)=SC3(I)*HKCON("HKCD4")*(USGDP/HKGDP);

!CONSUMPTION OF TRANSPORT & COMMUNICATION COMMODITIES FOR HONG KONG!
FORMULA (ALL,I,HKC5)
CHK(I)=SC4(I)*HKCON("HKCD5")*(USGDP/HKGDP);

!CONSUMPTION OF MISCELLANEOUS COMMODITIES FOR HONG KONG!
FORMULA (ALL,I,HKC6)
CHK(I)=SC5(I)*HKCON("HKCD6")*(USGDP/HKGDP);

        !GOVERNMENT CONSUMPTION!
```

```
COEFFICIENT (ALL,I,IND)                GHK(I)
!GOVERNMENT CONSUMPTION FOR HONG KONG!;
FORMULA (ALL,I,IND)
GHK(I)=SG1(I)*HKGOV*(USGDP/HKGDP);

                !INVESTMENT!

COEFFICIENT (ALL,I,IND)                IHK(I)
! ALL OTHER SALTER COMMODITIES FOR HONG KONG!;
FORMULA (ALL,I,IND)
IHK(I)=HKINV*(SININV(I)/SUM(G,IND,SININV(G)))*(USGDP/HKGDP);


!----------------------------------------------------------------
            WRITE STATEMENTS
 ----------------------------------------------------------------!
WRITE CHK TO FILE HKDEM HEADER "CON";
WRITE GHK TO FILE HKDEM HEADER "GOV";
WRITE IHK TO FILE HKDEM HEADER "INV";

WRITE SC1 TO FILE CHECK HEADER "SC1";
WRITE SC2 TO FILE CHECK HEADER "SC2";
WRITE SC3 TO FILE CHECK HEADER "SC3";
WRITE SC4 TO FILE CHECK HEADER "SC4";
WRITE SC5 TO FILE CHECK HEADER "SC5";
```

HKPRN.STI

hk.dat
sn.upd
hkprn.dat

HKPRN.TAB

```
!PURPOSE: TO FIND VALUE ADDED, OUTPUT, LABOUR RETURNS, RETURNS TO LAND
AND CAPITAL RETURNS FOR  HONG KONG!

                !FILE HKPRN.TAB!

!*****************************************************************!
!                    FILES                      !
!*****************************************************************!
FILE HKIO #FILE CONTAINING HK DATA#;

FILE SNIO #FILE CONTAINING SINGAPORE DATA#;

FILE (NEW) HKPRN #FILE CONTAINING PRODUCTION DATA FOR HONG KONG#;

!*****************************************************************!
!                    SETS                       !
!*****************************************************************!

SET IND (IND1 - IND37) ;

SET MF(MINE,FOOD);

SET SV (IND33, IND34, IND35);
SUBSET SV IS SUBSET OF IND;

SET WBHK(WB1-WB4);

!SUBSET FOR PRIMARY INDUSTRIES!
SET WBN1(IND1-IND8);
SUBSET WBN1 IS SUBSET OF IND;

SET WBN2(IND9-IND12);
SUBSET WBN2 IS SUBSET OF IND;

SET WBN3(IND35-IND37);
SUBSET WBN3 IS SUBSET OF IND;

SET NFMEGW(IND17-IND32);
SUBSET NFMEGW IS SUBSET OF IND;

! SET FOR FOOD!
SET FOOD(IND13-IND16);
```

```
SUBSET FOOD IS SUBSET OF IND;

! SET FOR agr!
SET agr(IND1-IND6);
SUBSET agr IS SUBSET OF IND;


!SET FOR SERVICES EXCLUDING EGW!
SET SERV(IND33-IND37);
SUBSET SERV IS SUBSET OF IND;

!SET FOR CREATING AN INDUSTRY IN HK EVEN THOUGH THERE IS NO OUTPUT DATA!
!THE APPROPRIATE LEVEL OF THIS INDUSTRY WILL BE SET IN HKPREP!
SET CORRECT(IND5,IND23);
SUBSET CORRECT IS SUBSET OF IND;

!----------------------------------------------------------------
                INPUT-OUTPUT DATABASE
 ----------------------------------------------------------------!

COEFFICIENT                 USGDP
! HONG KONG GDP IN US DOLLARS!;

COEFFICIENT                 HKGDP
!HONG KONG GDP IN HONG KONG DOLLARS!;

COEFFICIENT (ALL,I,WBHK)        WBVA(I)
!WORLD BANK VALUE ADDED!;

COEFFICIENT (ALL,I,NFMEGW)      UNVA(I)
!UNIDO VALUE ADDED!;
COEFFICIENT (ALL,I,MF)          UNVAMF(I);
COEFFICIENT (ALL,I,SV)          UNVASV(I);

COEFFICIENT (ALL,I,NFMEGW)      UNTC(I)
!UN OUTPUT (TOTAL COSTS) FOR HONG KONG!;
COEFFICIENT (ALL,I,MF)          UNTCMF(I);

COEFFICIENT (ALL,I,NFMEGW)      UNLAB(I)
!UN LABOUR RETURNS!;
COEFFICIENT (ALL,I,MF)          UNLABMF(I);

COEFFICIENT (ALL,J,IND)         SINLAB(J)
!SINGAPORE LABOUR!;

COEFFICIENT (ALL,J,IND)         SINCAP(J)
!SINGAPORE CAPITAL!;

COEFFICIENT (ALL,J,IND)         SINLN(J)
!SINGAPORE LAND!;

COEFFICIENT (ALL,I,IND) (ALL,J,IND)  SDINT(I,J)
```

```
!DOM INT SINGAPORE!;

COEFFICIENT (ALL,I,IND) (ALL,J,IND)  SIINT(I,J)
!IMPORTED INT SINGAPORE!;

COEFFICIENT (ALL,I,IND) (ALL,J,IND)  DINT(I,J)
!HK DINT!;

COEFFICIENT  (ALL,I,IND)   DCON(I)
!HK DCON!;

COEFFICIENT (ALL,I,IND)    DGOV(I)
!HK DGOV!;

COEFFICIENT  (ALL,I,IND)    DINV(I)
!HK DINV!;

COEFFICIENT  (ALL,I,IND)    EXP_(I)
!HK EXP!;


!-----------------------------------------------------------------
               READING DATABASE
-----------------------------------------------------------------!
READ
  USGDP FROM FILE HKIO HEADER "HK4";

  HKGDP FROM FILE HKIO HEADER "HK5";

   (ALL,I,WBHK) WBVA(I)
  FROM FILE HKIO HEADER "HK6";

   (ALL,I,NFMEGW) UNVA(I)
  FROM FILE HKIO HEADER "HK7";
   (ALL,I,MF) UNVAMF(I)
  FROM FILE HKIO HEADER "HK8";
   (ALL,I,SV) UNVASV(I)
  FROM FILE HKIO HEADER "HK9";

  (ALL,I,NFMEGW) UNTC(I)
  FROM FILE HKIO HEADER "HK10";
   (ALL,I,MF) UNTCMF(I)
  FROM FILE HKIO HEADER "HK11";

  (ALL,I,NFMEGW) UNLAB(I)
  FROM FILE HKIO HEADER "HK12";
   (ALL,I,MF) UNLABMF(I)
  FROM FILE HKIO HEADER "HK13";


  (ALL,J,IND) SINLAB(J)
       FROM FILE SNIO HEADER "AI13";
```

```
  (ALL,J,IND) SINCAP(J)
       FROM FILE SNIO HEADER "AI14";

  (ALL,J,IND) SINLN(J)
       FROM FILE SNIO HEADER "AI15";

  (ALL,I,IND) (ALL,J,IND) SDINT(I,J)
      FROM FILE SNIO HEADER "AI01";

  (ALL,I,IND) (ALL,J,IND) SIINT(I,J)
      FROM FILE SNIO HEADER "AI02";

!-----------------------------------------------------------------
              COEFFICIENTS
 ----------------------------------------------------------------!
formula sinln("ind5")=sinln("ind6");sincap("ind5")=sincap("ind6");
sinlab("ind5")=sinlab("ind6");
(all,i,ind) sdint(i,"ind5")=sdint(i,"ind6");
(all,i,ind) siint(i,"ind5")=siint(i,"ind6");

COEFFICIENT (ALL,J,IND)              SOUT(J)
!SINGAPORE OUTPUT!;
FORMULA (ALL,J,IND)
SOUT(J)=SINLN(J)+SINLAB(J)+SINCAP(J)+SUM(I,IND,SDINT(I,J))
                          +SUM(I,IND,SIINT(I,J));

COEFFICIENT (ALL,J,IND)          VASIN(J)
!GDP FOR SINGAPORE AT FACTOR COST!;
FORMULA (ALL,J,IND)
VASIN(J)=SINLAB(J)+SINCAP(J)+SINLN(J);

COEFFICIENT                      UNNFMAN
!HK VALUE ADDED FOR NON FOOD MANUFACTURING!;
FORMULA
UNNFMAN=SUM(I,NFMEGW,UNVA(I))+UNVAMF("FOOD")-UNVA("IND32");

COEFFICIENT                      UNSER
!HK VALUE ADDED FOR SERVICES!;
FORMULA
UNSER=SUM(I,SV,UNVASV(I))+UNVA("IND32");


!*****************************************************************
       COEFFICIENTS TO CALCULATE VALUE ADDED
 *****************************************************************!

COEFFICIENT (ALL,I,IND)              VA(I)
!VALUE ADDED FOR AGRICULTURE!;
!ASSUME THAT HK HAS THE SAME AG PROFILE AS SN BUT ITS OWN SHARE OF AG
IN ITS ECONOMY!
FORMULA (ALL,I,WBN1)
```

```
VA(I)=WBVA("WB1")*(VASIN(I)/SUM(G,WBN1,VASIN(G)))*USGDP;

!VALUE ADDED FOR MINING!
!ASSUME THE SAME FOR MINING AS FOR AG!
FORMULA (ALL,I,WBN2)
VA(I)=WBVA("WB2")*(VASIN(I)/SUM(G,WBN2,VASIN(G)))*USGDP;

!VALUE ADDED FOR FOOD!
!USE THE SHARE OF ALL FOOD PROCESSING IN ALL MANUFACTURING (SITC 3)
TO DETERMINE THE SIZE OF THE FOOD INDUSTRIES IN HK!
FORMULA (ALL,I,FOOD)
VA(I)=WBVA("WB3")*UNVAMF("FOOD")/(UNNFMAN+UNVAMF("FOOD"))*
   (VASIN(I)/SUM(G,FOOD,VASIN(G)))*USGDP ;

!VALUE ADDED FOR NON FOOD MANUFACTURING INDUSTRIES!
FORMULA(ALL,I,NFMEGW)
VA(I)=WBVA("WB3")*(UNVA(I)/UNNFMAN)*USGDP;

!VALUE ADDED FOR ELECTRICITY, WATER AND GAS!
FORMULA
VA("IND32")= WBVA("WB4")*(UNVA("IND32")/UNSER)*USGDP;

! SERVICES OTHER THAN EGW !
!VALUE ADDED FOR CONSTRUCTION!
FORMULA
VA("IND33")=WBVA("WB4")*(UNVASV("IND33")/UNSER)*USGDP ;

!VALUE ADDED FOR TRANSPORT SERVICES!
FORMULA
VA("IND34")=WBVA("WB4")*(UNVASV("IND34")/UNSER)*USGDP ;

!VALUE ADDED FOR SERVICE INDUSTRIES 35-37!
!VALUE ADDED FOR IND35-IND37 IS STORED UNDER ELEMENT IND35 IN UN DATA!
FORMULA   (ALL,I,WBN3)
VA(I)=WBVA("WB4")*(UNVASV("IND35")/UNSER)*
(VASIN(I)/SUM(F,WBN3,VASIN(F)))*USGDP;

!CREATE CORRECTED INDUSTRIES IN HK!
FORMULA (ALL,I,CORRECT) VA(I)=1.0;

!*************************************************************
            COEFFICIENTS TO CALCULATE OUTPUT
 *************************************************************!

COEFFICIENT (ALL,I,IND)                OUT(I)
!HONG KONG OUTPUT FOR AGRICULTURE!;
FORMULA (ALL,I,WBN1)
OUT(I)=SOUT(I)/VASIN(I)*VA(I);

!HONG KONG OUTPUT FOR MINING!
!ASSUME ALL MINING INDUSTRIES HAVE SAME STRUCTURE AS THE OVERALL
MINING INDUSTRY FROM INDUSTRIAL STATISTICS!
```

```
FORMULA (ALL,I,WBN2)
OUT(I)=UNTCMF("MINE")/UNVAMF("MINE")*VA(I);

!HONG KONG OUTPUT FOR FOOD!
!ASSUME THE SAME FOR STRUCTURE OF FOOD AS FOR MINING!
FORMULA (ALL,I,FOOD)
OUT(I)=UNTCMF("FOOD")/UNVAMF("FOOD")*VA(I);

!HONG KONG OUTPUT FOR NON FOOD MANUFACTURING AND EGW!
FORMULA (ALL,I,NFMEGW)
OUT(I)=UNTC(I)/UNVA(I)*VA(I);

!USE SINGAPORE STRUCTURE FOR ALL OTHER SERVICES!
FORMULA (ALL,I,SERV)
OUT(I)=SOUT(I)/VASIN(I)*VA(I);

!USE SINGAPORE STRUCTURE FOR CORRECTED INDUSTRIES!
FORMULA (ALL,I,CORRECT)
OUT(I)=SOUT(I)/VASIN(I)*VA(I);

!*****************************************************************
            COEFFICIENTS FOR LABOUR
*****************************************************************!

COEFFICIENT (ALL,I,IND)                LAB(I)
!HONG KONG LABOUR FOR PRIMARY INDUSTRIES!;
FORMULA (ALL,I,WBN1)
LAB(I)=SINLAB(I)/VASIN(I)*VA(I);

!HONG KONG LABOUR FOR MINING!
FORMULA (ALL,I,WBN2)
LAB(I)=UNLABMF("MINE")/UNVAMF("MINE")*VA(I);

!HONG KONG LABOUR FOR FOOD!
FORMULA (ALL,I,FOOD)
LAB(I)=UNLABMF("FOOD")/UNVAMF("FOOD")*VA(I);

!HONG KONG LABOUR FOR BEVERAGES AND TOBACCO!
FORMULA (ALL,I,NFMEGW)
LAB(I)=UNLAB(I)/UNVA(I)*VA(I);

!HONG KONG LABOUR FOR CONSTRUCTION!
FORMULA (ALL,I,SERV)
LAB(I)=SINLAB(I)/VASIN(I)*VA(I);

!HONG KONG LABOUR FOR CORRECTED INDUSTRIES!
FORMULA (ALL,I,CORRECT)
LAB(I)=SINLAB(I)/VASIN(I)*VA(I);

!*****************************************************************
          COEFFICIENTS FOR CAPITAL AND LAND
*****************************************************************!
```

```
COEFFICIENT (ALL,I,IND)                CAPLND(I)
!HONG KONG CAPITAL AND LAND USE FOR ALL INDUSTRIES IS THE COMPLEMENT
OF LABOUR IN VALUE ADDED!;
FORMULA (ALL,I,IND)
CAPLND(I)=VA(I)-LAB(I);

COEFFICIENT (ALL,I,IND)                CAP(I)
!HONG KONG CAPITAL FOR ALL INDUSTRIES IS THE PROPORTION USED IN SN!;
FORMULA (ALL,I,IND)
CAP(I)=CAPLND(I)*SINCAP(I)/(SINCAP(I)+SINLN(I));

COEFFICIENT (ALL,I,IND)                LAND(I)
!HONG KONG LAND FOR ALL INDUSTRIES IS THE PROPORTION USED IN SN!;
FORMULA (ALL,I,IND)
LAND(I)=CAPLND(I)*SINLN(I)/(SINCAP(I)+SINLN(I));

!redistribute Gross Operating Surplus to 50% land, 50% cap
for agr indys!
formula (all,i,agr)
land(i)=(land(i)+cap(i))/2;
(all,i,agr)
cap(i)=land(i);


!*****************************************************************
                WRITE STATEMENTS
*****************************************************************!

WRITE OUT TO FILE HKPRN HEADER "OUT";
WRITE LAB TO FILE HKPRN HEADER "LAB";
WRITE CAP TO FILE HKPRN HEADER "CAP";
WRITE LAND TO FILE HKPRN HEADER "LAND";
```

HKLINK.STI

```
f
b
n
hklink.dat
aw
h
hkprn.dat
cap
w
n
aw
h
hkprn.dat
land
w
n
aw
h
hkprn.dat
lab
w
n
aw
h
hkprn.dat
out
w
n
aw
h
hkdem.dat
con
w
n
aw
h
hkdem.dat
gov
w
n
aw
h
hkdem.dat
inv
w
n
ex
paj
23.7.92
**end
```

HKPREP.STI

hklink.dat
newtrade.dat
sn.upd
hkprep.dat

HKPREP.TAB

```
                !HKRPREP.TAB!
!*********************************************************!
!                   FILES                    !
!*********************************************************!

file HKD1
#Initial Hong Kong io data#;

file trade
#Trade data#;

FILE SNIO
#INCLUDES SINGAPORE DATA#;

file (new) HKD2 #Hong Kong io data for RASing#;

!*************************************************************!
!                   SETS                    !
!*************************************************************!

set primary (labour,capital,land);

SET IND(IND1-IND37);



set usage (IND1-IND37,inv,con,gov);
subset ind is subset of usage;

set final (inv,con,gov);
subset final is subset of usage;

set serv(ind33-ind37);
subset serv is subset of ind;

set correct(ind5,ind23);
subset correct is subset of ind;

SET SEC (AUSTRALIA,
NZ,
CANADA,
```

```
USA,
JAPAN,
KOREA,
EC,
Indonesia,
Malaysia,
Philippines,
Singapore,
Thailand,
China,
HK,
Taiwan,
ROFW) ;

!*************************************************************!
!                    COEFFICIENTS                           !
!*************************************************************!


coefficient no_ind; recip_ind;
formula no_ind=sum(i,ind,1); recip_ind=1/no_ind;

coefficient
totcost; totfinal; totdom; totimp; totprim; grandtot;
(all,u,final) tmp_final(u);
(all,i,primary) tmp_prim(i);

(all,u,usage) col_tot(u);

(all,j,ind) hk_lab(j);
(all,j,ind) hk_cap(j);
(all,j,ind) hk_land(j);

(ALL,I,IND)(ALL,J,IND)  hk_int(I,J);
(all,i,ind) hk_inv(i);
(all,i,ind) hk_con(i);
(all,i,ind) hk_gov(i);

(all,i,ind)  hk_out(i);

(all,i,ind)(all,j,ind)  hk_iint(i,j);
(all,i,ind)(all,j,ind)  hk_dint(i,j);

(all,i,ind) hk_iinv(i);
(all,i,ind)hk_dinv(i);

(all,i,ind) hk_icon(i);
(all,i,ind) hk_dcon(i);

(all,i,ind) hk_igov(i);
(all,i,ind) hk_dgov(i);
```

```
(all,i,ind) hk_dom(i);
(ALL,S,SEC)(ALL,D,SEC)(ALL,I,IND)   HK_EXPS(s,d,i);
(ALL,S,SEC)(ALL,D,SEC)(ALL,I,IND)   HK_IMPS(s,d,i);


(all,i,ind) hk_exp(i);
(all,s,sec) margs(s);
(all,i,ind) hk_imp(i);
(all,i,primary) hk_prim(i);

(ALL,I,IND) (ALL,J,IND)  SGDINT(I,J);

(ALL,I,IND) (ALL,J,IND)  SGIINT(I,J);

(ALL,I,IND)          SGDCON(I);

(ALL,I,IND)          SGICON(I);

(ALL,I,IND)          SGDINV(I);

(ALL,I,IND)          SGIINV(I);

(ALL,I,IND)          SGDGOV(I);

(ALL,I,IND)          SGIGOV(I);

(ALL,I,IND)          SGLAB(I);
(ALL,I,IND)          SGCAP(I);
(ALL,I,IND)          SGLAND(I);


!************************************************************
                  READING DATA BASE
 ************************************************************!


read hk_lab from file hkd1 header "lab";
read hk_cap from file hkd1 header "cap";
read hk_land from file hkd1 header "land";

read hk_out from file hkd1 header "out";

read hk_inv from file hkd1 header "inv";
read hk_con from file hkd1 header "con";
read hk_gov from file hkd1 header "gov";

read hk_exps from file trade header "exp";
    margs from file trade header "emrg";
READ hk_IMPs FROM FILE trade HEADER "imp";

READ SGDINT FROM FILE SNIO HEADER "AI01";
READ SGIINT  FROM FILE SNIO HEADER "AI02";
```

```
READ SGDCON FROM FILE SNIO HEADER "AI05";
READ SGICON  FROM FILE SNIO HEADER "AI06";
READ SGDINV FROM FILE SNIO HEADER "AI03";
READ SGIINV  FROM FILE SNIO HEADER "AI04";
READ SGDGOV FROM FILE SNIO HEADER "AI07";
READ SGIGOV  FROM FILE SNIO HEADER "AI08";


!----------------------------------------------------------------
                    CALCULATING COEFFICIENTS
 ----------------------------------------------------------------!
 zerodivide default recip_ind;

FORMULA (ALL,I,IND)
hk_exp(i) = sum(d,sec,HK_EXPS("HK",d,i))/1000;
formula
hk_exp("ind34") = hk_exp("ind34")+margs("hk")/1000;

formula (all,i,ind)
hk_imp(i)=sum(s,sec,hk_imps(s,"HK",i))/1000;

 COEFFICIENT (ALL,I,IND) (ALL,J,IND)  SGINTS(I,J)
 !SG TOTAL INT INPUT USE OF COMMODITY I BY INDUSTRY J!;
 FORMULA (ALL,I,IND) (ALL,J,IND)
 SGINTS(I,J)=SGIINT(I,J) +SGDINT(I,J);

 COEFFICIENT (ALL,I,IND) (ALL,J,IND)  SGINT_SH(I,J)
 !SG SHARE OF INT INPUT I IN TOTAL INT INPUT USE BY EACH INDUSTRY!;
 FORMULA (ALL,I,IND) (ALL,J,IND)
 SGINT_SH(I,J)=SGINTS(I,J)/SUM(F,IND,SGINTS(F,J));

coefficient (all,j,ind) intj(j)
!total int inputs demand by each hk industry!;
formula (all,j,ind)
intj(j)=hk_out(j)-hk_lab(j)-hk_cap(j)-hk_land(j);

!demand for each int input by each hk industry!
(all,i,ind)(all,j,ind) hk_int(i,j)= SGINT_SH(I,J)*intj(j);

zerodivide default 0.5;

!HK IMPORTED INTERMEDIATE INPUTS!
FORMULA (ALL,I,IND) (ALL ,J,IND)
HK_IINT(I,J)=SGIINT(I,J)/(SGDINT(I,J)+SGIINT(I,J))*HK_INT(I,J);

FORMULA (ALL,I,IND)(ALL,J,IND)
HK_DINT(I,J)=HK_INT(I,J)-HK_IINT(I,J);

FORMULA (ALL,I,IND)
HK_ICON(I)=SGICON(I)/(SGDCON(I)+SGICON(I))*HK_CON(I);

FORMULA (ALL,I,IND)
```

```
HK_DCON(I)=HK_CON(I)-HK_ICON(I);

FORMULA (ALL,I,IND)
HK_IGOV(I)=SGIGOV(I)/(SGDGOV(I)+SGIGOV(I))*HK_GOV(I);

FORMULA (ALL,I,IND)
HK_DGOV(I)=HK_GOV(I)-HK_IGOV(I);

FORMULA (ALL,I,IND)
HK_IINV(I)=SGIINV(I)/(SGDINV(I)+SGIINV(I))*HK_INV(I);

FORMULA (ALL,I,IND)
HK_DINV(I)=HK_INV(I)-HK_IINV(I);

coefficient (all,i,ind) totsales(i);
formula (all,i,ind) totsales(i)=
hk_dgov(i)+hk_dinv(i)+hk_dcon(i)+hk_exp(i)+sum(j,ind,hk_dint(i,j));

(all,j,ind) col_tot(j)=max(totsales(j),hk_out(j));

!FOR CORRECTED INDUSTRIES, WE SET OUTPUT TO BE 1.5 TIMES EXPORTS TO ALLOW
FOR THE INTERMEDIATE INPUTS NEEDED BY THE NEWLY CREATED INDUSTRIES!
(all,i,correct) col_tot(i)=1.5*hk_exp(i);

!RECALCULATE THE INDUSTRY STRUCTURE BASED ON THE NEW COLUMN TOTALS!
coefficient

(all,j,ind) n_hk_lab(j);
(all,j,ind) n_hk_cap(j);
(all,j,ind) n_hk_land(j);

formula
(all,j,ind) n_hk_lab(j)=hk_lab(j)/hk_out(j)*col_tot(j);
(all,j,ind) n_hk_cap(j)=hk_cap(j)/hk_out(j)*col_tot(j);
(all,j,ind) n_hk_land(j)=hk_land(j)/hk_out(j)*col_tot(j);

(all,i,ind) hk_dom(i)=col_tot(i)-hk_exp(i);
hk_prim("labour")=sum(j,ind,n_hk_lab(j));
hk_prim("capital")=sum(j,ind,n_hk_cap(j));
hk_prim("land")=sum(j,ind,n_hk_land(j));

col_tot("inv")=sum(i,ind,hk_inv(i));
col_tot("con")=sum(i,ind,hk_con(i));
col_tot("gov")=sum(i,ind,hk_gov(i));

totcost=sum(u,ind,col_tot(u));
totfinal=sum(u,final,col_tot(u));
totdom=sum(i,ind,hk_dom(i));
totimp=sum(i,ind,hk_imp(i));
totprim=sum(i,primary,hk_prim(i));

grandtot=max(totcost+totfinal,totdom+totimp+totprim);
```

```
(all,u,final) tmp_final(u)=col_tot(u);
(all,i,primary) tmp_prim(i)=hk_prim(i);

(all,u,final) col_tot(u)=tmp_final(u)*(grandtot-totcost)/totfinal;
(all,i,primary) hk_prim(i)=tmp_prim(i)*(grandtot-totdom-totimp)/totprim;

!-----------------------------------------------------------------
                    WRITE STATEMENTS
 -----------------------------------------------------------------!

formula (all,i,ind)(all,j,serv:hk_dint(i,j)=0) hk_dint(i,j)=0.1;
write hk_dint to file hkd2 header "ai01";
write hk_dinv to file hkd2 header "ai03";
write hk_dcon to file hkd2 header "ai05";
write hk_dgov to file hkd2 header "ai07";

formula (all,i,ind)(all,j,serv:hk_iint(i,j)=0) hk_iint(i,j)=0.1;
write hk_iint to file hkd2 header "ai02";
write hk_iinv to file hkd2 header "ai04";
write hk_icon to file hkd2 header "ai06";
write hk_igov to file hkd2 header "ai08";

write hk_imp to file hkd2 header "imp";
write hk_dom to file hkd2 header "dom";
write hk_exp to file hkd2 header "ai11";
write hk_prim to file hkd2 header "prim";

write col_tot to file hkd2 header "colt";

write n_hk_lab to file hkd2 header "ai13";
write n_hk_cap to file hkd2 header "ai14";
write n_hk_land to file hkd2 header "ai15";
```

FIDDLE.STI

hkprep.dat
hkras.dat
fidlmap.dat
hkadj.dat

FIDDLE.TAB

```
!This program allows the user to specify elements to be excluded from the ras
procedure. This is known as the modified ras procedure. It is used to
maintain certain characteristics from the original database in the rased
database. Only a few elements can be excluded otherwise the procedure will
not converge !

file hk_input #Hong Kong data from preparatory (hkprep.tab) stage#;
file (new) hk_ras #Hong Kong io data for RASing#;
file (new) adj_map #map for adjusting individual cells#;
file (new) hk_adj #original data from preparatory stage in table form#;

set ind (d1-d37);
set ind2 (i1-i37);
set primary (labour,capital,land);

set usage (d1-d37,inv,con,gov);
subset ind is subset of usage;

set inputs (d1-d37,i1-i37,labour,capital,land);
subset ind is subset of inputs;
subset ind2 is subset of inputs;
subset primary is subset of inputs;

coefficient

(all,u,usage)(all,i,inputs) table(i,u);
(all,u,usage)(all,i,inputs) map(i,u);
(all,u,usage) col_tot(u);
(all,i,inputs) row_tot(i);
(all,u,usage)(all,i,inputs) ras_table(i,u);
(all,u,usage) ras_col(u);
(all,i,inputs) ras_row(i);

(all,j,ind) lab(j);
(all,j,ind) cap(j);
(all,j,ind) land(j);

(all,i,ind2)(all,j,ind) iint(i,j);
(all,i,ind)(all,j,ind) dint(i,j);

(all,i,ind2) iinv(i);
```

```
(all,i,ind) dinv(i);

(all,i,ind2) icon(i);
(all,i,ind) dcon(i);

(all,i,ind2) igov(i);
(all,i,ind) dgov(i);

(all,i,ind) hk_dom(i);
(all,i,ind) hk_exp(i);
(all,i,ind2) hk_imp(i);
(all,i,primary) hk_prim(i);

read lab from file hk_input header "ai13";
read cap from file hk_input header "ai14";
read land from file hk_input header "ai15";

read dint from file hk_input header "ai01";
read dinv from file hk_input header "ai03";
read dcon from file hk_input header "ai05";
read dgov from file hk_input header "ai07";

read iint from file hk_input header "ai02";
read iinv from file hk_input header "ai04";
read icon from file hk_input header "ai06";
read igov from file hk_input header "ai08";

read hk_imp from file hk_input header "imp";
read hk_dom from file hk_input header "dom";
read hk_exp from file hk_input header "ai11";
read hk_prim from file hk_input header "prim";

read col_tot from file hk_input header "colt";

formula

(all,u,usage)(all,i,inputs) table(i,u)=0;

(all,u,ind)(all,i,ind) table(i,u)=dint(i,u);
(all,i,ind) table(i,"inv")=dinv(i);
(all,i,ind) table(i,"con")=dcon(i);
(all,i,ind) table(i,"gov")=dgov(i);

(all,u,ind)(all,i,ind2) table(i,u)=iint(i,u);
(all,i,ind2) table(i,"inv")=iinv(i);
(all,i,ind2) table(i,"con")=icon(i);
(all,i,ind2) table(i,"gov")=igov(i);

(all,u,ind) table("labour",u)=lab(u);
(all,u,ind) table("capital",u)=cap(u);
(all,u,ind) table("land",u)=land(u);
```

```
(all,i,ind) row_tot(i)=hk_dom(i);
(all,i,ind2) row_tot(i)=hk_imp(i);
(all,i,primary) row_tot(i)=hk_prim(i);

(all,u,usage)(all,i,inputs) map(i,u)=0;
map("capital","d37")=1;

(all,i,inputs)(all,u,usage) ras_table(i,u)=0;
(all,i,inputs)(all,u,usage:map(i,u)=0) ras_table(i,u)=table(i,u);
(all,u,usage) ras_col(u)=col_tot(u)-sum(i,inputs:map(i,u) ne 0,table(i,u));
(all,i,inputs) ras_row(i)=row_tot(i)-sum(u,usage:map(i,u) ne 0,table(i,u));

!transform into subsets because write cannot take subsets as options!
coefficient
(all,i,ind) dom_row(i);
(all,i,ind2) imp_row(i);
(all,i,primary) prim_row(i);
formula
(all,i,ind) dom_row(i)=ras_row(i);
(all,i,ind2) imp_row(i)=ras_row(i);
(all,i,primary) prim_row(i)=ras_row(i);

write ras_table to file hk_ras header "tabl";
write dom_row to file hk_ras header "dom";
write imp_row to file hk_ras header "imp";
write prim_row to file hk_ras header "prim";
write ras_col to file hk_ras header "colt";
write hk_exp to file hk_ras header "exp";

write map to file adj_map header "map";

write table to file hk_adj header "tabl";
write col_tot to file hk_adj header "colt";
write row_tot to file hk_adj header "rowt";
```

HKRAS.STI

```
s
100
n
aaa
aaa
1
aaa
aaa
hkras.dat
hkras.upd

p
a
n
u
n
0.1
a
y
a
**end
```

HKRAS.TAB

```
file hk_ras #Hong Kong io data for RASing#;

SET IND  (d1-d37);
set ind2 (i1-i37);
set primary (labour,capital,land);

set usage (d1-d37,inv,con,gov);
subset ind is subset of usage;

set inputs (d1-d37,i1-i37,labour,capital,land);
subset ind is subset of inputs;
subset ind2 is subset of inputs;
subset primary is subset of inputs;

coefficient

(all,u,usage)(all,i,inputs) table(i,u);
(all,u,usage)(all,i,inputs) ras_table(i,u);
(all,u,usage)(all,i,inputs) n_ras_table(i,u);
(all,u,usage) col_tot(u);
(all,i,inputs) row_tot(i);

(all,i,ind) hk_dom(i);
(all,i,ind) hk_exp(i);
(all,i,ind2) hk_imp(i);
(all,i,primary) hk_prim(i);

read table from file hk_ras header "tabl";

read hk_dom from file hk_ras header "dom";
read hk_imp from file hk_ras header "imp";
read hk_prim from file hk_ras header "prim";

read hk_exp from file hk_ras header "exp";
read col_tot from file hk_ras header "colt";

formula

(all,i,ind) row_tot(i)=hk_dom(i);
(all,i,ind2) row_tot(i)=hk_imp(i);
(all,i,primary) row_tot(i)=hk_prim(i);

(all,u,usage)(all,i,inputs) ras_table(i,u)=table(i,u);

! First step of RAS - scale to row totals !
formula
(all,i,inputs)(all,u,usage) n_ras_table(i,u)=ras_table(i,u)*row_tot(i)
                    /sum(u_,usage,ras_table(i,u_));

! Second step of RAS - scale to column totals !
```

```
(all,i,inputs)(all,u,usage) ras_table(i,u)=n_ras_table(i,u)*col_tot(u)
                    /sum(i_,inputs,n_ras_table(i_,u));

! Update data as consequence of RAS !

update (explicit) (all,u,usage)(all,i,inputs) table(i,u)=ras_table(i,u);

update (explicit) (all,u,ind) hk_exp(u)=hk_exp(u);

variable y;x;

equation dummy y=x;
```

HKHACK.STI

```
s
1
n
aaa
aaa
1
aaa
aaa
hkprep.dat
hkras.upd
hkadj.dat
fidlmap.dat
hkhack.dat
hkras.tmp
hkadj.tmp
fidlmap.tmp

p
a
n
u
n
0.5
a
y
a
**end
```

HKHACK.TAB

```
! This program takes the output from the modified RAS procedure and
reintegrates the bits left out in the fiddle.tab program!

file hk_input #dummy file to read elements that will be updated#;
file hk_ras #Hong Kong data after RASing#;
file hk_adj #Hong Kong table before RASing held for adjustment#;
file hk_map #map for data held for adjustment#;


SET IND  (d1-d37);
set ind2 (i1-i37);
set agr (d1-d6);
subset agr is subset of ind;
set primary (labour,capital,land);

set usage (d1-d37,inv,con,gov);
subset ind is subset of usage;

set inputs (d1-d37,i1-i37,labour,capital,land);
subset ind is subset of inputs;
subset ind2 is subset of inputs;
subset primary is subset of inputs;

set single (one);

coefficient

(all,u,usage)(all,i,inputs) adj_table(i,u);
(all,u,usage)(all,i,inputs) map(i,u);
(all,u,usage)(all,i,inputs) ras_table(i,u);

(all,u,usage) col_tot(u);
(all,i,inputs) row_tot(i);

(all,j,ind) lab(j);
(all,j,ind) cap(j);
(all,j,ind) land(j);

(all,i,ind2)(all,j,ind) iint(i,j);
(all,i,ind)(all,j,ind) dint(i,j);

(all,i,ind2) iinv(i);
(all,i,ind) dinv(i);

(all,i,ind2) icon(i);
(all,i,ind) dcon(i);

(all,i,ind2) igov(i);
(all,i,ind) dgov(i);
```

```
(all,i,ind) hk_dom(i);
(all,i,ind) hk_exp(i);
(all,i,ind2) hk_imp(i);
(all,i,primary) hk_prim(i);

read adj_table from file hk_adj header "tabl";
read map from file hk_map header "map";
read ras_table from file hk_ras header "tabl";

read lab from file hk_input header "ai13";
read cap from file hk_input header "ai14";
read land from file hk_input header "ai15";

read dint from file hk_input header "ai01";
read dinv from file hk_input header "ai03";
read dcon from file hk_input header "ai05";
read dgov from file hk_input header "ai07";

read iint from file hk_input header "ai02";
read iinv from file hk_input header "ai04";
read icon from file hk_input header "ai06";
read igov from file hk_input header "ai08";

read hk_exp from file hk_input header "ai11";


update (explicit) (all,u,ind)(all,i,ind)
dint(i,u)=sum(s_,single:map(i,u) ne 0,adj_table(i,u))+
sum(s_,single:map(i,u)=0,ras_table(i,u));
update (explicit) (all,i,ind)
dinv(i)=sum(s_,single:map(i,"inv") ne 0,adj_table(i,"inv"))+
sum(s_,single:map(i,"inv")=0,ras_table(i,"inv"));
update (explicit) (all,i,ind)
dcon(i)=sum(s_,single:map(i,"con") ne 0,adj_table(i,"con"))+
sum(s_,single:map(i,"con")=0,ras_table(i,"con"));
update (explicit) (all,i,ind)
dgov(i)=sum(s_,single:map(i,"gov") ne 0,adj_table(i,"gov"))+
sum(s_,single:map(i,"gov")=0,ras_table(i,"gov"));

update (explicit) (all,u,ind)(all,i,ind2)
iint(i,u)=sum(s_,single:map(i,u) ne 0,adj_table(i,u))+
sum(s_,single:map(i,u)=0,ras_table(i,u));
update (explicit) (all,i,ind2)
iinv(i)=sum(s_,single:map(i,"inv") ne 0,adj_table(i,"inv"))+
sum(s_,single:map(i,"inv")=0,ras_table(i,"inv"));
update (explicit) (all,i,ind2)
icon(i)=sum(s_,single:map(i,"con") ne 0,adj_table(i,"con"))+
sum(s_,single:map(i,"con")=0,ras_table(i,"con"));
update (explicit) (all,i,ind2)
igov(i)=sum(s_,single:map(i,"gov") ne 0,adj_table(i,"gov"))+
sum(s_,single:map(i,"gov")=0,ras_table(i,"gov"));
```

```
update (explicit) (all,u,ind)
lab(u)=sum(s_,single:map("labour",u) ne 0,adj_table("labour",u))+
sum(s_,single:map("labour",u)=0,ras_table("labour",u));
update (explicit) (all,u,ind)
cap(u)=sum(s_,single:map("capital",u) ne 0,adj_table("capital",u))+
sum(s_,single:map("capital",u)=0,ras_table("capital",u));
update (explicit) (all,u,ind)
land(u)=sum(s_,single:map("land",u) ne 0,adj_table("land",u))+
sum(s_,single:map("land",u)=0,ras_table("land",u));

update (explicit) (all,u,ind) hk_exp(u)=hk_exp(u);

variable y;x;

equation dummy y=(lab("d1")+cap("d1")+land("d1")+dint("d1","d1")
+iint("i1","d1")+dcon("d1")+icon("i1")+dgov("d1")+igov("i1")+
dinv("d1")+iinv("i1"))*x;
```

PRECHECK.STI

f
b
y
hkhack.dat
hkform.dat
aw
t
r
2
37 37
y
0
ai16
com tax on dom interm
w
n
aw
t
r
2
37 37
y
0
ai17
com tax on imp interm
w
n
aw
t
r
1
37
y
0
ai12
ind taxes
w
n
aw
t
r
1
37
y
0
ai18
com tax on dom cons
w
n
aw

t
r
1
37
y
0
ai19
com tax on imp cons
w
n
aw
t
r
1
37
y
0
ai20
dom inv taxes
w
n
aw
t
r
1
37
y
0
ai21
imp inv tax
w
n
aw
t
r
1
37
y
0
ai22
dom gov taxes
w
n
aw
t
r
1
37
y
0
ai23
imp gov tax

w
n
aw
t
r
1
37
y
0
ai24
exp taxes
w
n
aw
t
r
1
37
y
0
ai27
duties
w
n
aw
t
r
1
1
y
0
ai28
income tax
w
n
aw
t
r
1
1
y
0
ai29
transfers
w
n
aw
t
r
1
1
y

0
ai30

w
n
aw
t
r
1
1
y
0
ai31
depreciation
w
n
aw
t
r
1
1
y
0
ai32
savings
w
n
aw
t
r
1
37
y
0
ai09

w
n
aw
t
r
1
37
y
0
ai10

w
n
aw
t
r

```
l
37
y
0
ai25

w
n
aw
t
r
l
37
y
0
ai26

w
n
ex
a
f
paj
12792
**end
y
```

UPDCHECK.STI

hkrupd.har
taxio.dat
chkbal.dat

UPDCHECK.TAB

```
!************************************************!
!                    FILES                       !
!************************************************!

FILE DATIO # FILE CONTAINING ALL UPDATED I/O INFORMATION # ;
FILE (NEW) TAXES # FILE CONTAINING OUTPUT TAX INFORMATION # ;
FILE (NEW) TRADE # FILE CONTAINING OUTPUT TRADE INFORMATION # ;

!************************************************************!
!                    SETS                        !
!************************************************************!

SET IND (IND1 - IND37) ;

SET REG (REGION) ;

!************************************************************!
!                  DATA BASE                     !
!************************************************************!


!-------------------------------------------------------------
            INPUT-OUTPUT DATA BASE
 -------------------------------------------------------------!

! SUBSCRIPT RANGE                          COEFFICIENT
            DETAILED DESCRIPTION              NAME
 ------------------------------------      ----------- !

COEFFICIENT (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)          DINT(I,J,Z)
! INTERMEDIATE USAGE (DOMESTIC) !      ;

COEFFICIENT (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)  IINTS(I,J,Z)
! INTERMEDIATE USAGE (BY IMPORTING SECTOR) !      ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                  DINV(I,Z)
! INVESTMENT USAGE (DOMESTIC) ! ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)          IINVS(I,Z)
! INVESTMENT USAGE (BY IMPORTING SECTOR) ! ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                  DCON(I,Z)
```

```
! CONSUMPTION USAGE (DOMESTIC)   ! ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)          ICONS(I,Z)
! CONSUMPTION USAGE (BY IMPORTING SECTOR)   ! ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                DGOV(I,Z)
! GOVERNMENT USAGE OF DOMESTIC COMMODITIES   ! ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)          IGOVS(I,Z)
! GOVERNMENT USAGE OF IMPORTED COMMODITIES   ! ;


COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                DSTOCK(I,Z)
! DOMESTIC CHANGES IN STOCKS   ! ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)          ISTOCK(I,Z)
! IMPORTED CHANGES IN STOCKS   ! ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)          EXPS(I,Z)
! EXPORTS OF COMMODITY I  ! ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                TSR(I,Z)
! NON COMMODITY INDIRECT TAXES, NET OF SUBSIDIES ! ;
COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                COSTINP(I,Z)
! TOTAL VALUE OF DOMESTIC OUTPUT ! ;

COEFFICIENT (ALL,J,IND)(ALL,Z,REG)                LAB(J,Z)
! USAGE OF LABOUR  ! ;

COEFFICIENT (ALL,J,IND)(ALL,Z,REG)                CAP(J,Z)
! USAGE OF CAPITAL ! ;

COEFFICIENT (ALL,J,IND)(ALL,Z,REG)                LAND(J,Z)
! USAGE OF LAND   ! ;

COEFFICIENT (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)          TRD(I,J,Z)
! COMMODITY TAXES ON DOMESTICALLY PRODUCED GOODS USED AS INTERMEDIATE
USAGE ! ;
COEFFICIENT (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)          VALDOMINT(I,J,Z)
! TOTAL VALUE OF DOMESTIC INTER USAGE INCL. TAXES ! ;


COEFFICIENT (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)          TRI(I,J,Z)
! COMMODITY TAXES ON IMPORTS USED AS INTERMEDIATE USAGE ! ;
COEFFICIENT (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)          VALIMPINT(I,J,Z)
! TOTAL VALUE OF IMPORTED INTER USAGE INCL. TAXES ! ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                TIRD(I,Z)
! COMMODITY TAXES ON DOMESTICALLY PRODUCED GOODS USED AS INVESTMENT ! ;
COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                VALDOMINV(I,Z)
! TOTAL VALUE OF DOMESTIC INVESTMENT USAGE INCL. TAXES ! ;
```

```
COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                TIRI(I,Z)
! COMMODITY TAXES ON IMPORTED PRODUCED GOODS USED AS INVESTMENT ! ;
COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                VALIMPINV(I,Z)
! TOTAL VALUE OF IMPORTED INVESTMENT INCL. TAXES ! ;


COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                TCRD(I,Z)
! COMMODITY TAXES ON DOMESTICALLY PRODUCED GOODS SOLD AS CONSUMPTION
! ;
COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                VALDOMCON(I,Z)
! TOTAL VALUE OF DOMESTIC CONSUMPTION INCL. TAXES ! ;


COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                TCRI(I,Z)
! COMMODITY TAXES ON IMPORTS SOLD AS CONSUMPTION ! ;
COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                VALIMPCON(I,Z)
! TOTAL VALUE OF IMPORTED CONSUMPTION INCL. TAXES ! ;


COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                TGRD(I,Z)
! COMMODITY TAXES ON DOMESTICALLY PRODUCED GOODS SOLD TO GOVERNMENT !
;
COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                VALDOMGOV(I,Z)
! TOTAL VALUE OF DOMESTIC GOVERNMENT USAGE INCL. TAXES ! ;


COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                TGRI(I,Z)
! COMMODITY TAXES ON IMPORTS SOLD TO GOVERNMENT ! ;
COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                VALIMPGOV(I,Z)
! TOTAL VALUE OF IMPORTED GOVERNMENT USAGE INCL. TAXES ! ;


COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                TSTD(I,Z)
! COMMODITY TAXES ON DOMESTICALLY PRODUCED CHANGES IN STOCKS ! ;
COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                VALDOMST(I,Z)
! TOTAL VALUE OF DOMESTIC CHANGES IN STOCKS INCL. TAXES ! ;


COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                TSTI(I,Z)
! COMMODITY TAXES ON IMPORTED CHANGES IN STOCKS ! ;
COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                VALIMPST(I,Z)
! TOTAL VALUE OF CHANGES IN STOCKS USAGE INCL. TAXES ! ;


COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                TER(I,Z)
! COMMODITY TAXES EXPORTS ! ;
COEFFICIENT (ALL,I,IND)(ALL,Z,REG)                VALEXP(I,Z)
! TOTAL VALUE OF EXPORTS INCL. TAXES ! ;


COEFFICIENT (ALL,I,IND)(ALL,Z,REG)          DR(I,Z)
```

```
! DUTY BY SECTOR ! ;
COEFFICIENT (ALL,I,IND)(ALL,Z,REG)          IMPLS(I,Z)
! TOTAL VALUE OF IMPORTS INCL DUTY ! ;


!************************************************************
                READING DATA BASE
 ************************************************************!

READ (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)
    DINT(I,J,Z)
          FROM FILE DATIO HEADER "AI01" ;

   (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)
    IINTS(I,J,Z)
          FROM FILE DATIO HEADER "AI02" ;

   (ALL,I,IND)(ALL,Z,REG)
    DINV(I,Z)
          FROM FILE DATIO HEADER "AI03" ;

   (ALL,I,IND)(ALL,Z,REG)
    IINVS(I,Z)
          FROM FILE DATIO HEADER "AI04" ;

   (ALL,I,IND)(ALL,Z,REG)
    DCON(I,Z)
          FROM FILE DATIO HEADER "AI05" ;

   (ALL,I,IND)(ALL,Z,REG)
    ICONS(I,Z)
          FROM FILE DATIO HEADER "AI06" ;

   (ALL,I,IND)(ALL,Z,REG)
    DGOV(I,Z)
          FROM FILE DATIO HEADER "AI07" ;

   (ALL,I,IND)(ALL,Z,REG)
    IGOVS(I,Z)
          FROM FILE DATIO HEADER "AI08" ;

   (ALL,I,IND)(ALL,Z,REG)
    DSTOCK(I,Z)
          FROM FILE DATIO HEADER "AI09" ;

   (ALL,I,IND)(ALL,Z,REG)
    ISTOCK(I,Z)
          FROM FILE DATIO HEADER "AI10" ;

   (ALL,I,IND)(ALL,Z,REG)
    EXPS(I,Z)
```

```
        FROM FILE DATIO HEADER "AI11" ;

(ALL,I,IND)(ALL,Z,REG)
 TSR(I,Z)
        FROM FILE DATIO HEADER "AI12" ;

(ALL,I,IND)(ALL,Z,REG)
 LAB(I,Z)
        FROM FILE DATIO HEADER "AI13" ;

(ALL,I,IND)(ALL,Z,REG)
 CAP(I,Z)
        FROM FILE DATIO HEADER "AI14" ;

(ALL,I,IND)(ALL,Z,REG)
 LAND(I,Z)
        FROM FILE DATIO HEADER "AI15" ;

(ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)
 TRD(I,J,Z)
        FROM FILE DATIO HEADER "AI16" ;

(ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)
 TRI(I,J,Z)
        FROM FILE DATIO HEADER "AI17" ;

(ALL,I,IND)(ALL,Z,REG)
 TCRD(I,Z)
       FROM FILE DATIO HEADER  "AI18" ;

(ALL,I,IND)(ALL,Z,REG)
 TCRI(I,Z)
       FROM FILE DATIO HEADER  "AI19" ;

(ALL,I,IND)(ALL,Z,REG)
 TIRD(I,Z)
        FROM FILE DATIO HEADER "AI20" ;

(ALL,I,IND)(ALL,Z,REG)
 TIRI(I,Z)
        FROM FILE DATIO HEADER "AI21" ;

(ALL,I,IND)(ALL,Z,REG)
 TGRD(I,Z)
       FROM FILE DATIO HEADER  "AI22" ;

(ALL,I,IND)(ALL,Z,REG)
 TGRI(I,Z)
       FROM FILE DATIO HEADER  "AI23" ;

(ALL,I,IND)(ALL,Z,REG)
 TER(I,Z)
```

```
        FROM FILE  DATIO  HEADER  "AI24" ;

   (ALL,I,IND)(ALL,Z,REG)
    TSTD(I,Z)
        FROM FILE  DATIO  HEADER  "AI25" ;

   (ALL,I,IND)(ALL,Z,REG)
    TSTI(I,Z)
        FROM FILE  DATIO  HEADER  "AI26" ;

   (ALL,I,IND)(ALL,Z,REG)
    DR(I,Z)
        FROM FILE  DATIO  HEADER  "AI27" ;


!************************************************************
          WIDELY USED DATABASE AGGREGATES
************************************************************!

  ZERODIVIDE DEFAULT 0.0;

 ZERODIVIDE (nonzero_by_zero) DEFAULT 0.0 ;

!-------------------------------------------------------------
       AGGREGATION OF IMPORTS BY USER AND BY SECTOR
 -------------------------------------------------------------!

 COEFFICIENT (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG) IINT(I,J,Z)
! INTERMEDIATE USAGE (IMPORTED) ! ;
 FORMULA  (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)
 IINT(I,J,Z) = IINTS(I,J,Z)  ;

 COEFFICIENT (ALL,I,IND)(ALL,Z,REG) IINV(I,Z)
! INVESTMENT USAGE (IMPORTED) ! ;
 FORMULA  (ALL,I,IND)(ALL,Z,REG)
 IINV(I,Z) = IINVS(I,Z)  ;

 COEFFICIENT (ALL,I,IND)(ALL,Z,REG) ICON(I,Z)
! CONSUMPTION (IMPORTED) ! ;
 FORMULA  (ALL,I,IND)(ALL,Z,REG)
 ICON(I,Z) = ICONS(I,Z)  ;

 COEFFICIENT (ALL,I,IND)(ALL,Z,REG) IGOV(I,Z)
! GOVERNMENT USAGE (IMPORTED) ! ;
 FORMULA  (ALL,I,IND)(ALL,Z,REG)
 IGOV(I,Z) = IGOVS(I,Z)  ;

!-------------------------------------------------------------
            GOVERNMENT AGGREGATES
 -------------------------------------------------------------!

 COEFFICIENT (ALL,Z,REG) AG(Z)
```

```
! GOVERNMENT SPENDING ON GOODS!  ;
 FORMULA  (ALL,Z,REG)
 AG(Z) = SUM(I,IND,DGOV(I,Z)) + SUM(I,IND,IGOV(I,Z))
     + SUM(I,IND,TGRD(I,Z)) + SUM(I,IND,TGRI(I,Z));

!-----------------------------------------------------------------
              TRADE AGGREGATES
 -----------------------------------------------------------------!

COEFFICIENT (ALL,I,IND)(ALL,Z,REG) VALIMP(I,Z);
 FORMULA (ALL,I,IND)(ALL,Z,REG)
 VALIMP(I,Z) =  SUM(J,IND,IINTS(I,J,Z)) + ICONS(I,Z)
          + IINVS(I,Z) + IGOVS(I,Z) + ISTOCK(I,Z)-dr(i,z) ;

 FORMULA (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)
 VALDOMINT(I,J,Z) =  DINT(I,J,Z) + TRD(I,J,Z) ;

 FORMULA (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)
 VALIMPINT(I,J,Z) =  IINT(I,J,Z) + TRI(I,J,Z) ;

 FORMULA (ALL,I,IND)(ALL,Z,REG)
 VALDOMINV(I,Z) =  DINV(I,Z) + TIRD(I,Z) ;

 FORMULA (ALL,I,IND)(ALL,Z,REG)
 VALIMPINV(I,Z) =  IINV(I,Z) + TIRI(I,Z) ;

 FORMULA (ALL,I,IND)(ALL,Z,REG)
 VALDOMCON(I,Z) =  DCON(I,Z) + TCRD(I,Z) ;

 FORMULA (ALL,I,IND)(ALL,Z,REG)
 VALIMPCON(I,Z) =  ICON(I,Z) + TCRI(I,Z) ;

 FORMULA (ALL,I,IND)(ALL,Z,REG)
 VALDOMGOV(I,Z) =  DGOV(I,Z) + TGRD(I,Z) ;

 FORMULA (ALL,I,IND)(ALL,Z,REG)
 VALIMPGOV(I,Z) = IGOV(I,Z) + TGRI(I,Z) ;

 FORMULA (ALL,I,IND)(ALL,Z,REG)
 VALEXP(I,Z) = EXPS(I,Z) + TER(I,Z) ;

! IMPORTS FROM THE STH TO THE VTH SECTOR !
 FORMULA  (ALL,I,IND)(ALL,Z,REG)
! IMPORTS(CIF) OF THE ITH COMMODITY BY SECTOR !
 IMPLS(I,Z) = VALIMP(I,Z)  + DR(I,Z) ;

 COEFFICIENT (ALL,I,IND) IMPL(I)
! IMPORTS(CIF) OF COMMODITY I BY SECTOR S!  ;
 FORMULA  (ALL,I,IND)
 IMPL(I) = SUM(Z,REG,IMPLS(I,Z))  ;

 COEFFICIENT  (ALL,Z,REG) IMPTT(Z)
```

```
! TOTAL IMPORTS(CIF) !  ;
 FORMULA  (ALL,Z,REG)
 IMPTT(Z) = SUM(I,IND,IMPL(I)) ;

 COEFFICIENT (ALL,Z,REG) EXPFOB(Z)
! TOTAL EXPORTS AT FOB PRICES !   ;
 FORMULA  (ALL,Z,REG)
 EXPFOB(Z) = SUM(I,IND,VALEXP(I,Z)) ;


!----------------------------------------------------------------
            PRODUCTION AGGREGATES
 ----------------------------------------------------------------!

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)   DOMPN(I,Z)
! VALUE OF DOMESTIC PRODUCTION OF THE ITH COMMODITY !   ;
 FORMULA (ALL,I,IND)(ALL,Z,REG)
 DOMPN(I,Z) = SUM(J,IND,DINT(I,J,Z)) + DINV(I,Z) + DGOV(I,Z)
        + DCON(I,Z) + EXPS(I,Z) + DSTOCK(I,Z);


! TOTAL COSTS OF INPUTS TO INDUSTRY EXCLUDING INDIRECT TAXES!
 FORMULA (ALL,J,IND)(ALL,Z,REG)
 COSTINP(J,Z) = SUM(I,IND,DINT(I,J,Z)) + SUM(I,IND,IINT(I,J,Z))
        + SUM(I,IND,TRD(I,J,Z)) + SUM(I,IND,TRI(I,J,Z))
        + LAB(J,Z) + CAP(J,Z) + LAND(J,Z) ;

COEFFICIENT (ALL,J,IND)(ALL,Z,REG)          TOTCOSTS(J,Z)
! TOTAL COSTS OF ALL INPUTS TO INDUSTRY ! ;
 FORMULA (ALL,J,IND)(ALL,Z,REG)
 TOTCOSTS(J,Z) = COSTINP(J,Z) + TSR(J,Z) ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)  PDIFF(I,Z)
! PERCENTAGE DIFFERENCE BETWEEN COSTS AND SALES ! ;
FORMULA (ALL,I,IND)(ALL,Z,REG)
PDIFF(I,Z) = 100*(TOTCOSTS(I,Z) - DOMPN(I,Z))/DOMPN(I,Z) ;

!----------------------------------------------------------------
            EXPENDITURE AGGREGATES
 ----------------------------------------------------------------!

 COEFFICIENT (ALL,Z,REG) CTT(Z)
! TOTAL CONSUMPTION !  ;
 FORMULA  (ALL,Z,REG)
 CTT(Z) = SUM(I,IND,DCON(I,Z)) + SUM(I,IND,ICON(I,Z))
       + SUM(I,IND,TCRD(I,Z)) + SUM(I,IND,TCRI(I,Z)) ;

 COEFFICIENT (ALL,Z,REG) INVTT(Z)
! TOTAL INVESTMENT !   ;
 FORMULA  (ALL,Z,REG)
 INVTT(Z) = SUM(I,IND,DINV(I,Z)) + SUM(I,IND,IINV(I,Z))
       + SUM(I,IND,TIRD(I,Z)) + SUM(I,IND,TIRI(I,Z))  ;
```

```
 COEFFICIENT (ALL,Z,REG) STOCKTT(Z)
! TOTAL CHANGES IN STOCKS !  ;
 FORMULA  (ALL,Z,REG)
 STOCKTT(Z) = SUM(I,IND,DSTOCK(I,Z)) + SUM(I,IND,ISTOCK(I,Z))
       + SUM(I,IND,TIRD(I,Z)) + SUM(I,IND,TIRI(I,Z))  ;

!----------------------------------------------------------------
              INCOME AGGREGATES
 ----------------------------------------------------------------!


 ZERODIVIDE DEFAULT 1.0 ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)              T_INDTAX(I,Z);
 FORMULA (ALL,I,IND)(ALL,Z,REG)
 T_INDTAX(I,Z) = TOTCOSTS(I,Z)/COSTINP(I,Z)  ;

COEFFICIENT (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)        T_DOMINT(I,J,Z);
 FORMULA (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)
 T_DOMINT(I,J,Z) = VALDOMINT(I,J,Z) / DINT(I,J,Z)  ;

COEFFICIENT (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)        T_IMPINT(I,J,Z);
 FORMULA (ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)
 T_IMPINT(I,J,Z) = VALIMPINT(I,J,Z) / IINT(I,J,Z)  ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)              T_DOMINV(I,Z);
 FORMULA (ALL,I,IND)(ALL,Z,REG)
 T_DOMINV(I,Z) = VALDOMINV(I,Z) / DINV(I,Z)  ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)              T_IMPINV(I,Z) ;
 FORMULA (ALL,I,IND)(ALL,Z,REG)
 T_IMPINV(I,Z) = VALIMPINV(I,Z) / IINV(I,Z)  ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)              T_DOMCON(I,Z) ;
 FORMULA (ALL,I,IND)(ALL,Z,REG)
 T_DOMCON(I,Z) = VALDOMCON(I,Z) / DCON(I,Z)  ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)              T_IMPCON(I,Z) ;
 FORMULA (ALL,I,IND)(ALL,Z,REG)
 T_IMPCON(I,Z) = VALIMPCON(I,Z) / ICON(I,Z)  ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)              T_DOMGOV(I,Z) ;
 FORMULA (ALL,I,IND)(ALL,Z,REG)
 T_DOMGOV(I,Z) = VALDOMGOV(I,Z) / DGOV(I,Z)  ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)              T_IMPGOV(I,Z) ;
 FORMULA (ALL,I,IND)(ALL,Z,REG)
 T_IMPGOV(I,Z) = VALIMPGOV(I,Z) / IGOV(I,Z)  ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)              V_EXPSUB(I,Z) ;
 FORMULA (ALL,I,IND)(ALL,Z,REG)
```

```
 V_EXPSUB(I,Z) = VALEXP(I,Z) / EXPS(I,Z) ;

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)          T_IMPDUTY(I,Z) ;
 FORMULA (ALL,I,IND)(ALL,Z,REG)
 T_IMPDUTY(I,Z) = IMPLS(I,Z)/ (VALIMP(I,Z) ) ;


WRITE (ALL,I,IND)(ALL,Z,REG)
     VALEXP(I,Z)
          TO FILE TRADE HEADER "EXP" ;
    (ALL,I,IND)(ALL,Z,REG)
     VALIMP(I,Z)
          TO FILE TRADE HEADER "IMP" ;

    (ALL,Z,REG)
     CTT(Z)
          TO FILE TRADE HEADER "CON" ;

     (ALL,Z,REG)
     INVTT(Z)
          TO FILE TRADE HEADER "INV" ;

    (ALL,Z,REG)
     AG(Z)
          TO FILE TRADE HEADER "GOV" ;

   (ALL,I,IND)(ALL,Z,REG)
    TOTCOSTS(I,Z)
          TO FILE TRADE HEADER "TOTC" ;


   (ALL,I,IND)(ALL,Z,REG)
    PDIFF(I,Z)
          TO FILE TRADE HEADER "PDIF" ;

   (ALL,I,IND)(ALL,Z,REG)
    DOMPN(I,Z)
          TO FILE TRADE HEADER "TOTS" ;


   (ALL,I,IND)(ALL,Z,REG)
    DSTOCK(I,Z)
          TO FILE TRADE HEADER "DSTK" ;


   (ALL,I,IND)(ALL,Z,REG)
    ISTOCK(I,Z)
          TO FILE TRADE HEADER "ISTK" ;



    (ALL,I,IND)(ALL,Z,REG)
```

```
 T_INDTAX(I,Z)
      TO FILE TAXES HEADER "0001" ;

(ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)
 T_DOMINT(I,J,Z)
      TO FILE TAXES HEADER "0002" ;

(ALL,I,IND)(ALL,J,IND)(ALL,Z,REG)
 T_IMPINT(I,J,Z)
      TO FILE TAXES HEADER "0003" ;

(ALL,I,IND)(ALL,Z,REG)
 T_DOMINV(I,Z)
      TO FILE TAXES HEADER "0004" ;

(ALL,I,IND)(ALL,Z,REG)
 T_IMPINV(I,Z)
      TO FILE TAXES HEADER "0005" ;

(ALL,I,IND)(ALL,Z,REG)
 T_DOMCON(I,Z)
      TO FILE TAXES HEADER "0006" ;

(ALL,I,IND)(ALL,Z,REG)
 T_IMPCON(I,Z)
      TO FILE TAXES HEADER "0007" ;

(ALL,I,IND)(ALL,Z,REG)
 T_DOMGOV(I,Z)
      TO FILE TAXES HEADER "0008" ;

(ALL,I,IND)(ALL,Z,REG)
 T_IMPGOV(I,Z)
      TO FILE TAXES HEADER "0009" ;

(ALL,I,IND)(ALL,Z,REG)
 V_EXPSUB(I,Z)
      TO FILE TAXES HEADER "0010" ;

(ALL,I,IND)(ALL,Z,REG)
 T_IMPDUTY(I,Z)
      TO FILE TAXES HEADER "0011" ;
```

GOS.STI

hkform.dat
hkrupd.har

GOS.TAB

```
!*********************************************************!
!                    FILES                                !
!*********************************************************!

FILE DATIO # FILE CONTAINING ALL I/O INFORMATION # ;
FILE (NEW) RUPD #FILE CONTAINING SINGLE COUNTRY DATABASES POST
ADJUSTMENT # ;


!*****************************************************!
!                    SETS                             !
!*****************************************************!

SET IND (IND1 - IND37) ;

SET AGR (IND1 - IND6) ;
SUBSET AGR IS SUBSET OF IND  ;

SET REG (REGION) ;



!*********************************************************!
!                  DATA BASE                              !
!*********************************************************!


!-----------------------------------------------------------
               INPUT-OUTPUT DATA BASE
 -----------------------------------------------------------!

! SUBSCRIPT RANGE                          COEFFICIENT
               DETAILED DESCRIPTION              NAME
 ------------------------------------          ---------- !

COEFFICIENT (ALL,I,IND)(ALL,J,IND)          DINT(I,J)
! INTERMEDIATE USAGE (DOMESTIC) !  ;

COEFFICIENT (ALL,I,IND)(ALL,J,IND)          IINTS(I,J)
! INTERMEDIATE USAGE (BY IMPORTING SECTOR) !     ;

COEFFICIENT (ALL,I,IND)                     DINV(I)
```

```
! INVESTMENT USAGE (DOMESTIC) ! ;

COEFFICIENT (ALL,I,IND)                IINVS(I)
! INVESTMENT USAGE (BY IMPORTING SECTOR) ! ;

COEFFICIENT (ALL,I,IND)                DCON(I)
! CONSUMPTION USAGE (DOMESTIC)  ! ;

COEFFICIENT (ALL,I,IND)          ICONS(I)
! CONSUMPTION USAGE (BY IMPORTING SECTOR)  ! ;

COEFFICIENT (ALL,I,IND)                DGOV(I)
! GOVERNMENT USAGE OF DOMESTIC COMMODITIES  ! ;

COEFFICIENT (ALL,I,IND)          IGOVS(I)
! GOVERNMENT USAGE OF IMPORTED COMMODITIES  ! ;

COEFFICIENT (ALL,I,IND)               CDOMS(I)
! DOMESTIC CHANGES IN STOCKS ! ;

COEFFICIENT (ALL,I,IND)          CIMPS(I)
! CHANGE IN IMPORTED STOCKS  ! ;

COEFFICIENT (ALL,I,IND)          EXPS(I)
! EXPORTS OF COMMODITY I  ! ;

COEFFICIENT (ALL,J,IND)                LAB(J)
! USAGE OF LABOUR  ! ;

COEFFICIENT (ALL,J,IND)                CAP(J)
! USAGE OF CAPITAL ! ;

COEFFICIENT (ALL,J,IND)                LAND(J)
! USAGE OF LAND   ! ;

COEFFICIENT (ALL,I,IND)                TSR(I)
! NON  COMMODITY INDIRECT TAXES, NET OF SUBSIDIES ! ;

COEFFICIENT (ALL,I,IND)(ALL,J,IND)          TRD(I,J)
! COMMODITY TAXES ON DOMESTICALLY PRODUCED GOODS USED AS INTERMEDIATE
USAGE ! ;

COEFFICIENT (ALL,I,IND)(ALL,J,IND)          TRI(I,J)
! COMMODITY TAXES ON IMPORTS USED AS INTERMEDIATE USAGE ! ;

COEFFICIENT (ALL,I,IND)                TIRD(I)
! COMMODITY TAXES ON DOMESTICALLY PRODUCED GOODS USED AS INVESTMENT ! ;

COEFFICIENT (ALL,I,IND)                TIRI(I)
! COMMODITY TAXES ON IMPORTED PRODUCED GOODS USED AS INVESTMENT ! ;

COEFFICIENT (ALL,I,IND)                TCRD(I)
```

```
! COMMODITY TAXES ON DOMESTICALLY PRODUCED GOODS SOLD AS CONSUMPTION
! ;

COEFFICIENT (ALL,I,IND)                 TCRI(I)
! COMMODITY TAXES ON IMPORTS SOLD AS CONSUMPTION ! ;

COEFFICIENT (ALL,I,IND)                 TGRD(I)
! COMMODITY TAXES ON DOMESTICALLY PRODUCED GOODS SOLD TO GOVERNMENT !
;

COEFFICIENT (ALL,I,IND)                 TGRI(I)
! COMMODITY TAXES ON IMPORTS SOLD TO GOVERNMENT ! ;

COEFFICIENT (ALL,I,IND)                 TER(I)
! COMMODITY TAXES EXPORTS ! ;

COEFFICIENT (ALL,I,IND)                 TSTD(I)
! COMMODITY TAXES ON DOMESTIC STOCKS ! ;

COEFFICIENT (ALL,I,IND)                 TSTI(I)
! COMMODITY TAXES ON IMPORTED STOCKS ! ;


COEFFICIENT (ALL,I,IND)                 DR(I)
! DUTY ! ;



!*****************************************************
                READING DATA BASE
*****************************************************!


READ (ALL,I,IND)(ALL,J,IND)
    DINT(I,J)
            FROM FILE  DATIO  HEADER  "AI01" ;

   (ALL,I,IND)(ALL,J,IND)
    IINTS(I,J)
            FROM FILE  DATIO  HEADER  "AI02" ;

   (ALL,I,IND)
    DINV(I)
            FROM FILE  DATIO  HEADER  "AI03" ;

   (ALL,I,IND)
    IINVS(I)
            FROM FILE  DATIO  HEADER  "AI04" ;

   (ALL,I,IND)
    DCON(I)
            FROM FILE  DATIO  HEADER  "AI05" ;
```

```
(ALL,I,IND)
 ICONS(I)
        FROM FILE DATIO HEADER "AI06" ;

(ALL,I,IND)
 DGOV(I)
        FROM FILE DATIO HEADER "AI07" ;

(ALL,I,IND)
 IGOVS(I)
        FROM FILE DATIO HEADER "AI08" ;
(ALL,I,IND)
 CDOMS(I)
        FROM FILE DATIO HEADER "AI09" ;

(ALL,I,IND)
 CIMPS(I)
        FROM FILE DATIO HEADER "AI10" ;

(ALL,I,IND)
 EXPS(I)
        FROM FILE DATIO HEADER "AI11" ;


(ALL,I,IND)
 TSR(I)
        FROM FILE DATIO HEADER "AI12" ;


(ALL,I,IND)
 LAB(I)
        FROM FILE DATIO HEADER "AI13" ;

(ALL,I,IND)
 CAP(I)
        FROM FILE DATIO HEADER "AI14" ;

(ALL,I,IND)
 LAND(I)
        FROM FILE DATIO HEADER "AI15" ;

(ALL,I,IND)(ALL,J,IND)
 TRD(I,J)
        FROM FILE DATIO HEADER "AI16" ;

(ALL,I,IND)(ALL,J,IND)
 TRI(I,J)
        FROM FILE DATIO HEADER "AI17" ;

(ALL,I,IND)
 TCRD(I)
```

```
        FROM FILE DATIO HEADER  "AI18" ;

 (ALL,I,IND)
  TCRI(I)
        FROM FILE DATIO HEADER  "AI19" ;

 (ALL,I,IND)
  TIRD(I)
        FROM FILE DATIO HEADER "AI20" ;

 (ALL,I,IND)
  TIRI(I)
        FROM FILE DATIO HEADER "AI21" ;

 (ALL,I,IND)
  TGRD(I)
        FROM FILE DATIO HEADER  "AI22" ;

 (ALL,I,IND)
  TGRI(I)
        FROM FILE DATIO HEADER  "AI23" ;

 (ALL,I,IND)
  TER(I)
        FROM FILE DATIO HEADER  "AI24" ;

(ALL,I,IND)
  TSTD(I)
        FROM FILE DATIO HEADER  "AI25" ;

(ALL,I,IND)
  TSTI(I)
        FROM FILE DATIO HEADER  "AI26" ;


 (ALL,I,IND)
  DR(I)
        FROM FILE DATIO HEADER  "AI27" ;


!*********************************************************
            GOS ADJUSTMENTS
*********************************************************!

COEFFICIENT (ALL,A,AGR)              GOS(A)
! TOTAL GOS ! ;
FORMULA (ALL,A,AGR)
GOS(A) = LAND(A) + CAP(A) ;


FORMULA (ALL,A,AGR)
```

```
LAND(A) = GOS(A)/2 ;

FORMULA (ALL,A,AGR)
CAP(A) = GOS(A)/2 ;

!*********************************************************
                    WRITE STATEMENTS
*********************************************************!


WRITE

   (ALL,I,IND)(ALL,J,IND)
    DINT(I,J)
            TO FILE RUPD HEADER "AI01" ;

   (ALL,I,IND)(ALL,J,IND)
    IINTS(I,J)
            TO FILE RUPD HEADER "AI02" ;

   (ALL,I,IND)
    DINV(I)
            TO FILE RUPD HEADER "AI03" ;

   (ALL,I,IND)
    IINVS(I)
            TO FILE RUPD HEADER "AI04" ;

   (ALL,I,IND)
    DCON(I)
            TO FILE RUPD HEADER "AI05" ;

   (ALL,I,IND)
    ICONS(I)
            TO FILE RUPD HEADER "AI06" ;

   (ALL,I,IND)
    DGOV(I)
            TO FILE RUPD HEADER "AI07" ;

   (ALL,I,IND)
    IGOVS(I)
            TO FILE RUPD HEADER "AI08" ;
   (ALL,I,IND)
    CDOMS(I)
            TO FILE RUPD HEADER "AI09" ;

   (ALL,I,IND)
    CIMPS(I)
            TO FILE RUPD HEADER "AI10" ;

   (ALL,I,IND)
```

```
EXPS(I)
        TO FILE RUPD HEADER "AI11";

(ALL,I,IND)
TSR(I)
        TO FILE RUPD HEADER "AI12";


(ALL,I,IND)
LAB(I)
        TO FILE RUPD HEADER "AI13";

(ALL,I,IND)
CAP(I)
        TO FILE RUPD HEADER "AI14";

(ALL,I,IND)
LAND(I)
        TO FILE RUPD HEADER "AI15";

(ALL,I,IND)(ALL,J,IND)
TRD(I,J)
        TO FILE RUPD HEADER "AI16";

(ALL,I,IND)(ALL,J,IND)
TRI(I,J)
        TO FILE RUPD HEADER "AI17";

(ALL,I,IND)
TCRD(I)
        TO FILE RUPD HEADER "AI18";

(ALL,I,IND)
TCRI(I)
        TO FILE RUPD HEADER "AI19";

(ALL,I,IND)
TIRD(I)
        TO FILE RUPD HEADER "AI20";

(ALL,I,IND)
TIRI(I)
        TO FILE RUPD HEADER "AI21";

(ALL,I,IND)
TGRD(I)
        TO FILE RUPD HEADER "AI22";

(ALL,I,IND)
TGRI(I)
        TO FILE RUPD HEADER "AI23";
```

```
(ALL,I,IND)
 TER(I)
        TO FILE RUPD HEADER "AI24" ;

(ALL,I,IND)
 TSTD(I)
        TO FILE RUPD HEADER "AI25" ;

(ALL,I,IND)
 TSTI(I)
        TO FILE RUPD HEADER "AI26" ;

(ALL,I,IND)
 DR(I)
        TO FILE RUPD HEADER "AI27" ;
```

APPENDIX B: FILES USED TO CREATE THE ROW SINGLE COUNTRY DATABASE

The TABLO code for all files listed in Figure 4 follows.

ROW.BAT

```
up modhar <rowdata.sti>out1
up rowcig <rowcig.sti>out2
up rowva <rowva.sti >out3
up modhar <combrow.sti>out4
up modhar <header.sti>out5
up makerow <make.sti>out6
up rowras <ras.sti>out7

grep error out?
del rwfin.har
ren ras100.out rwfin.har


@echo off
echo   The RW database is in file RWFIN.HAR.  Now checking for balance.
echo   Check header "PDIF" in file BAL.SEE

up modhar <precheck.sti
echo ras100.new >updcheck.sti
echo temp.out >>updcheck.sti
echo bal.out  >>updcheck.sti
up updcheck <updcheck.sti
echo. >see.sti
echo bal.out >>see.sti
echo bal.see >>see.sti
echo 6 >>see.sti
echo. >>see.sti
echo r >>see.sti
up seehar <see.sti

del ras100.new
del temp.out
del updcheck.sti
del see.sti
```

ROWDATA.STI

```
f
i
n
rw92.dat
aw
t
r
2
8 5
n
590   847   478   426   275   0     57    1757
5736  0  1136  2913  0  0  721  1580
1140  67  259  99  154  11  176  318
83166  26518  25971  34844  0  43774  0  117954
88530  28732  40040  24729  105095  27865  65988  138149
UN1
Private consumption for rest of the world rep. countries
w
n
AW
t
r
2
14 5
n
870  0  0  554  176
0  67  35  0  0
0  0  160  16
87  1099  136  561  152
214  9  37  6  28
6  29  145  200
73  133  0  49  24
10  7  4  6  17
4  10  9  1
4424  5287  5343  18603  10717
4122  1807  1034  72  1032
164  1210  926  50
24373  24853  12654  54138  65835
48099  3244  14761  328  1353
753  8067  11293  1468
UN2
Government consumption for rest of world rep. countries
W
N
AW
t
r
2
5 1
n
```

8295
21270
3326
493765
1020890
UN3
GDP for rep. countries of the rest of the world in foreign currency
W
N
AW
t
r
2
1 6
n
419304
736359
602832
909114
1878585
753660
UN4
GDP for regions of rest of the world in $US
W
N
AW
t
r
2
4 5
n
0.11 0.12 0.31 0.46
0.32 0.11 0.19 0.38
0.13 0.115 0.315 0.44
0.14 0.11 0.26 0.49
0.04 0.13 0.3 0.53
UN5
WB value added for row rep. countries
W
N
AW
t
r
2
20 5
n
340 387 156 112 8
48 76 417 0 63
209 16 159 89 81
20 239 270 768 436
17764 1666 25200 1073 641
899 2877 35554 12044 9062

22137 1424 5260 15226 18627
2406 150 170 484 275
159115 39396 33402 15465 3102
20406 63113 90127 50481 24437
45640 196500 27914 8872 11904
1056 40137 69670 217389 57502
19940 5650 12770 9930 2140
3180 6210 56510 9800 9940
19680 5180 13210 23580 33690
8450 54293 92233 128107 94425
21470 3120 1680 1440 300
13220 27050 30930 3780 5680
12290 4170 22410 40110 30350
1000 28739 62335 114746 60435
UN6
Value added for man & services (UNIDO)
W
N
AW
t
r
2
6 4
n
1317 729 405 555 1452 857
127901 178509 65229 128325 314727 140629
500916 97333 128785 210487 220658 167880
47596 6233 41226 127018 177238 116419
UN7
Output for row rep countries (National Accounts)
W
N
AW
t
r
2
16 5
n
1011 492 509 235 27
88 161 983 0 147
387 33 367 167 111 32
150285 7829 131586 7345 6693
4307 20349 192978 116504 30135
126429 16756 23514 63650 68907 4900
468600 73800 65500 33300 10400
45800 118700 205100 251800 40100
102400 454400 67000 27100 30600 2500
220200 25400 49600 25300 10100
10000 22800 175000 72600 29600
87600 48100 39100 87800 89800 19100
73730 6590 3070 2670 730
35650 67430 59050 16600 10170

35340 13240 43220 86680 62480 1770
UN8
Output for rep. countries of row (UNIDO)
W
N
AW
t
r
2
6 4
n
673 457 239 270 768 436
69371 95212 40137 69670 217389 57502
189222 56360 54293 92233 128107 94425
28726 3467 28739 62335 114746 60435
UN9
Value added for ag & mining (National accounts)
W
N
AW
t
r
2
6 3
n
313 193 46 167 356 332
27148 35698 8793 34698 58369 30886
7414 1965 5336 43755 75710 41862
UN10
Labour returns for Row rep countries (National accounts)
W
N
AW
t
r
2
6 3
n
360 264 193 103 412 104
42223 59514 31344 34972 159020 26616
21312 1502 23403 18580 39036 18573
UN11
Capital returns for Row rep countries (National Accounts)
W
N
AW
t
r
2
16 5
n
122 48 73 58 3

24 35 126 0 35
106 5 83 48 19 13
7325 597 17592 494 347
434 1766 12982 1302 3187
10275 1132 2615 9755 8412 389
13256 1681 3688 2317 382
2926 1801 7990 274 1208
4442 6127 3996 1225 2232 134
11602 1098 1136 1349 231
8461 12133 13065 785 3036
7152 2248 14213 22370 19853 660
19476 2463 6393 4820 1090
1164 1818 12052 1019 3164
7166 2880 5364 8920 14132 3712
UN12
Labour returns for row rep. countries (UNIDO)
W
N
AW
t
r
2
16 5
n
218 339 83 54 6
24 41 290 0 29
103 11 76 41 62 7
10439 1069 7608 579 294
465 1111 22572 10742 5875
11862 292 2645 5471 10215 2017
145859 37715 29714 13148 2720
17480 61312 82137 50207 23229
41198 190373 23918 7647 9672 922
464 3187 6377 5110 1050
2016 4392 44458 8781 6776
12514 2300 7846 14660 19558 4738
9868 2022 544 91 69
4759 14917 17865 2995 2644
5138 1922 8197 17740 10497 340
UN13
Capital returns for rep.row countries (UNIDO)
W
N
ex

27/1/92
rest of the world data
**end
y

ROWCIG.STI

```
rw92.dat
ec.upd
th.upd
kr.upd
cig.out
```

ROWCIG.TAB

```
! This file obtains the consumption, government and investment demand
  for the rest of the world (ROW).
-------------------------------------------------------------------------!

!*************************************************************!
!                    FILES                                    !
!*************************************************************!
 FILE UNIO #FILE CONTAINING ALL UN I/O INFORMATION#;
 FILE EC #FILE CONTAINING EC DATA#;
 FILE Thailand #FILE CONTAINING Thailand DATA#;
 FILE KOREA #FILE CONTAINING KOREAN DATA#;

 FILE (NEW) ROWD92 # DEMAND INFORMATION  FOR ROW #;

!*****************************************************!
!                    SETS                             !
!*****************************************************!
SET IND (IND1 - IND37) ;

SET SEC (AUSTRALIA,
NZ,
CANADA,
USA,
JAPAN,
KOREA,
EC,
Indonesia,
Malaysia,
Philippines,
Singapore,
Thailand,
China,
HK,
Taiwan,
ROFW) ;

SET REG (AUSTRALIA,
NZ,
CANADA,
USA,
```

```
JAPAN,
KOREA,
EC,
Indonesia,
Malaysia,
Philippines,
Singapore,
Thailand,
China,
HK,
Taiwan);
SUBSET REG IS SUBSET OF SEC;

SET Proxy(EC,
Thailand,
KOREA);
SUBSET Proxy IS SUBSET OF SEC;
SUBSET Proxy IS SUBSET OF REG;

SET FD(C,G,I,TC,TG,TI);

SET FO(LAB,LN,K);

SET SCR(D,IM);

SET ROFW (AFRICA,
MIDEAST,
SASIA,
LATAMER,
CPE,
WEURO) ;

SET ROFWA (AFRICA,
MIDEAST,
SASIA,
LATAMER,
WEURO);
SUBSET ROFWA IS SUBSET OF ROFW;

SET ROFWB (CPE);
SUBSET ROFWB IS SUBSET OF ROFW;

SET Af_MidE_asia  (AFRICA,
MIDEAST,
SASIA);
SUBSET Af_MidE_asia  IS SUBSET OF ROFW;
SUBSET Af_MidE_asia  IS SUBSET OF ROFWA;

SET Latin_Am ( LATAMER);
SUBSET Latin_Am IS SUBSET OF ROFW;
SUBSET Latin_Am IS SUBSET OF ROFWA;
```

```
SET West_Europe (WEURO);
SUBSET West_Europe IS SUBSET OF ROFW;
SUBSET West_Europe IS SUBSET OF ROFWA;

SET CPE_LatinAm(LATAMER,CPE);
SUBSET CPE_LatinAm IS SUBSET OF ROFW;


!SETS NEEDED ONLY FOR CONSUMPTION; maps UN data to SALTER industries!

!UN commodity aggregates!
SET UNCons (UNfood,
UNBev_Tob,
UN_TCF,
UNfuel,
UNdwellings,
UNtransport,
UNmisc,
UNtrade);

!Food and food products!
SET Food_Prod(IND1 - IND4, IND13 - IND16);
SUBSET Food_Prod IS SUBSET OF IND;

!Textiles, clothing and footwear!
SET TCF (IND18 - IND20);
SUBSET TCF IS SUBSET OF IND;

!Beverages and tobacco!
SET Bev_Tob(IND17);
SUBSET Bev_Tob IS SUBSET OF IND;

!Fuel!
SET Fuel (IND9 - IND11, IND23);
SUBSET Fuel IS SUBSET OF IND;

!Ownership of dwellings!
SET Dwellings(IND37);
SUBSET Dwellings IS SUBSET OF IND;

!Transport industries!
SET Transport(IND29);
SUBSET Transport IS SUBSET OF IND;

!Miscellaneous industries!
SET Misc(IND5 - IND8, IND12, IND21 - IND22, IND24 - IND28, IND30 - IND33,
IND35 - IND36);
SUBSET Misc IS SUBSET OF IND;

!Trade and transport!
SET Trade(IND34);
SUBSET Trade IS SUBSET OF IND;
```

```
!SETS ONLY NEEDED FOR GOVERNMENT CONSUMPTION; maps UN to SALTER!

!UN aggregate commodities for govt. consumption!
SET UNG(UNGovt1-UNGovt6,
UNdwell_govt,
UNgovt8,
UNpetrol_gov,
UNagr_govt,
UNmanuf_govt,
UNtrade_govt,
UNGovt13-UNGovt14);

!Agriculture industries!
SET agriculture(IND1 - IND8);
SUBSET agriculture IS SUBSET OF IND;

!manufacturing mining and construction industries!
SET man_mine_con(IND9 - IND22, IND24-IND33, IND35);
SUBSET man_mine_con IS SUBSET OF IND;

!UN government services commodity aggregates!
SET UNgovService( UNGovt1 - UNGovt6, UNGovt8, UNGovt13 - UNGovt14);
SUBSET UNgovService IS SUBSET OF UNG;

SET OSgovt(IND36);
SUBSET OSgovt IS SUBSET OF IND;

SET Petrol(IND23);
SUBSET Petrol IS SUBSET OF IND;


!----------------------------------------------------------------
              INPUT-OUTPUT FILES
----------------------------------------------------------------!

COEFFICIENT (ALL,X,ROFWA)                    ROFWGDP(X)
!GDP FOR REST OF WORLD REPRESENTATIVE COUNTRIES  IN LOCAL CURRENCY!;

COEFFICIENT  (ALL,X,ROFW)                    RWUSGDP(X)
!GDP FOR REST OF WORLD REGIONS IN US$!;

COEFFICIENT  (ALL,I,UNCons) (ALL,X,ROFWA)          Pri_Cons(I,X)
!PRIVATE CONSUMPTION OF COMMODITY I!;

COEFFICIENT (ALL,I,UNG) (ALL,X,ROFWA)              Gov_Cons(I,X)
!GOVERNMENT CONSUMPTION!;


COEFFICIENT (ALL,F,FD) (ALL,SC,SCR) (ALL,I,IND) (ALL,Z,Proxy)
```

```
                                        FINDEM(F,SC,I,Z)
 !INCLUDES SETS DEFINED TO READ FROM SINGLE COUNTRY DATABASE!;

 COEFFICIENT (ALL,G,FO) (ALL,I,IND) (ALL,Z,Proxy)    FINDO(G,I,Z)
 !INCLUDES SETS DEFINED TO READ FROM SINGLE COUNTRY DATABASE!;


!*****************************************************
                READING DATA BASE
 *****************************************************!
READ

!READ GDP DATA FROM FILES!
   (ALL,X,ROFWA)
   ROFWGDP(X)
         FROM FILE UNIO HEADER "UN3";

    (ALL,X,ROFW)
     RWUSGDP(X)
         FROM FILE UNIO HEADER "UN4";


!READ CONSUMPTION DATA FROM FILES!
   (ALL,I,UNCons) (ALL,X,ROFWA)
   Pri_Cons(I,X)
           FROM FILE UNIO HEADER   "UN1";


!*********************************************
read single country data
*********************************************!

   (ALL,I,IND)
   FINDEM("C","D",I,"EC")
          FROM FILE EC HEADER   "AI05";

  (ALL,I,IND)
  FINDEM ("C","IM",I,"EC")
         FROM FILE EC HEADER    "AI06";

   (ALL,I,IND)
   FINDEM("C","D",I,"Thailand")
          FROM FILE Thailand HEADER   "AI05";

  (ALL,I,IND)
  FINDEM ("C","IM",I,"Thailand")
         FROM FILE Thailand HEADER    "AI06";

  (ALL,I,IND)
   FINDEM("C","D",I,"KOREA")
          FROM FILE KOREA HEADER   "AI05";
```

```
 (ALL,I,IND)
 FINDEM ("C","IM",I,"KOREA")
        FROM FILE KOREA HEADER    "AI06";

(ALL,I,IND)
  FINDEM("TC","D",I,"EC")
         FROM FILE EC HEADER   "AI18";

 (ALL,I,IND)
 FINDEM ("TC","IM",I,"EC")
        FROM FILE EC HEADER    "AI19";

 (ALL,I,IND)
 FINDEM("TC","D",I,"Thailand")
         FROM FILE Thailand HEADER   "AI18";

 (ALL,I,IND)
 FINDEM ("TC","IM",I,"Thailand")
        FROM FILE Thailand HEADER    "AI19";

 (ALL,I,IND)
 FINDEM("TC","D",I,"KOREA")
         FROM FILE KOREA HEADER   "AI18";

 (ALL,I,IND)
 FINDEM ("TC","IM",I,"KOREA")
        FROM FILE KOREA HEADER    "AI19";

           (ALL,I,IND)
 FINDEM("G","D",I,"EC")
         FROM FILE EC HEADER   "AI07";

 (ALL,I,IND)
 FINDEM ("G","IM",I,"EC")
        FROM FILE EC HEADER    "AI08";

 (ALL,I,IND)
 FINDEM("G","D",I,"Thailand")
         FROM FILE Thailand HEADER   "AI07";

 (ALL,I,IND)
 FINDEM ("G","IM",I,"Thailand")
        FROM FILE Thailand HEADER    "AI08";

 (ALL,I,IND)
 FINDEM("G","D",I,"KOREA")
         FROM FILE KOREA HEADER   "AI07";

 (ALL,I,IND)
 FINDEM ("G","IM",I,"KOREA")
        FROM FILE KOREA HEADER    "AI08";
```

```
(ALL,I,IND)
  FINDEM("TG","D",I,"EC")
        FROM FILE EC HEADER   "AI22";

 (ALL,I,IND)
 FINDEM ("TG","IM",I,"EC")
       FROM FILE EC HEADER    "AI23";

  (ALL,I,IND)
  FINDEM("TG","D",I,"Thailand")
        FROM FILE Thailand HEADER   "AI22";

 (ALL,I,IND)
 FINDEM ("TG","IM",I,"Thailand")
       FROM FILE Thailand HEADER    "AI23";

 (ALL,I,IND)
  FINDEM("TG","D",I,"KOREA")
        FROM FILE KOREA HEADER   "AI22";

 (ALL,I,IND)
 FINDEM ("TG","IM",I,"KOREA")
       FROM FILE KOREA HEADER    "AI23";

 (ALL,I,UNG) (ALL,X,ROFWA)
 Gov_Cons(I,X)
      FROM FILE UNIO HEADER "UN2";

!READ INVESTMENT DATA FROM FILES!
(ALL,I,IND)
  FINDEM("I","D",I,"EC")
        FROM FILE EC HEADER   "AI03";

 (ALL,I,IND)
 FINDEM ("I","IM",I,"EC")
       FROM FILE EC HEADER    "AI04";

  (ALL,I,IND)
  FINDEM("I","D",I,"Thailand")
        FROM FILE Thailand HEADER   "AI03";

 (ALL,I,IND)
 FINDEM ("I","IM",I,"Thailand")
       FROM FILE Thailand HEADER    "AI04";

 (ALL,I,IND)
  FINDEM("I","D",I,"KOREA")
        FROM FILE KOREA HEADER   "AI03";

 (ALL,I,IND)
 FINDEM ("I","IM",I,"KOREA")
       FROM FILE KOREA HEADER    "AI04";
```

```
(ALL,I,IND)
  FINDEM("TI","D",I,"EC")
          FROM FILE EC HEADER   "AI20";

 (ALL,I,IND)
 FINDEM ("TI","IM",I,"EC")
         FROM FILE EC HEADER    "AI21";

  (ALL,I,IND)
  FINDEM("TI","D",I,"Thailand")
          FROM FILE Thailand HEADER   "AI20";

 (ALL,I,IND)
 FINDEM ("TI","IM",I,"Thailand")
         FROM FILE Thailand HEADER    "AI21";

 (ALL,I,IND)
  FINDEM("TI","D",I,"KOREA")
          FROM FILE KOREA HEADER   "AI20";

 (ALL,I,IND)
 FINDEM ("TI","IM",I,"KOREA")
         FROM FILE KOREA HEADER    "AI21";

(ALL,I,IND)
FINDO("LN",I,"EC")
          FROM FILE EC HEADER "AI15";

(ALL,I,IND)
FINDO("LN",I,"Thailand")
          FROM FILE Thailand HEADER "AI15";

(ALL,I,IND)
FINDO("LN",I,"KOREA")
          FROM FILE KOREA HEADER "AI15";

(ALL,I,IND)
FINDO("LAB",I,"EC")
          FROM FILE EC HEADER "AI13";

(ALL,I,IND)
FINDO("LAB",I,"Thailand")
          FROM FILE Thailand HEADER "AI13";

(ALL,I,IND)
FINDO("LAB",I,"KOREA")
          FROM FILE KOREA HEADER "AI13";

(ALL,I,IND)
FINDO("K ",I,"EC")
          FROM FILE EC HEADER "AI14";
```

```
(ALL,I,IND)
FINDO("K",I,"Thailand")
        FROM FILE Thailand HEADER "AI14";
(ALL,I,IND)
FINDO("K",I,"KOREA")
        FROM FILE KOREA HEADER "AI14";



!*********************************************************
            DATABASE AGGREGATES
*********************************************************!

COEFFICIENT (ALL,I,IND) (ALL,Z,Proxy) LAB(I,Z)
!LABOUR RETURNS FROM SINGLE COUNTRY DATA!;
FORMULA (ALL,I,IND) (ALL,Z,Proxy)
LAB(I,Z)=FINDO("LAB",I,Z);

COEFFICIENT (ALL,I,IND) (ALL,Z,Proxy) LAND(I,Z)
!LAND RETURNS FROM SINGLE COUNTRY DATA!;
FORMULA (ALL,I,IND) (ALL,Z,Proxy)
LAND(I,Z)=FINDO("LN",I,Z);

COEFFICIENT (ALL,I,IND) (ALL,Z,Proxy) CAP(I,Z)
!CAPITAL RETURNS FROM SINGLE COUNTRY DATA!;
FORMULA (ALL,I,IND) (ALL,Z,Proxy)
CAP(I,Z)=FINDO("K",I,Z);

COEFFICIENT (ALL,Z,Proxy) LTT(Z)
! TOTAL LABOUR USAGE !  ;
FORMULA  (ALL,Z,Proxy)
LTT(Z) = SUM(J,IND,LAB(J,Z))  ;

COEFFICIENT (ALL,Z,Proxy) KTT(Z)
! TOTAL CAPITAL USAGE  !  ;
FORMULA  (ALL,Z,Proxy)
KTT(Z) = SUM(J,IND,CAP(J,Z))  ;

COEFFICIENT (ALL,Z,Proxy) MTT(Z)
! TOTAL LAND USAGE  !  ;
FORMULA  (ALL,Z,Proxy)
MTT(Z)  = SUM(J,IND,LAND(J,Z))  ;

COEFFICIENT (ALL,Z,Proxy) GDPFCL(Z)
! GDPFC AT FACTOR COST ! ;
FORMULA (ALL,Z,Proxy)
GDPFCL(Z) = LTT(Z) + KTT(Z) + MTT(Z)  ;

COEFFICIENT (ALL,I,IND)   (ALL,Z,Proxy)        DCONS(I,Z)
!DOMESTIC CONSUMPTION FOR Proxy COUNTRIES!;
FORMULA (ALL,I,IND) (ALL,Z,Proxy)
```

```
 DCONS(I,Z) =FINDEM("C","D",I,Z);

 COEFFICIENT (ALL,I,IND)  (ALL,Z,Proxy)    ICONS(I,Z)
 !DOMESTIC CONSUMPTION FOR Proxy COUNTRIES!;
 FORMULA (ALL,I,IND) (ALL,Z,Proxy)
ICONS(I,Z) =FINDEM("C","IM",I,Z) ;

 COEFFICIENT (ALL,I,IND)  (ALL,Z,Proxy)    TCRD(I,Z)
 !DOMESTIC CONSUMPTION FOR Proxy COUNTRIES!;
 FORMULA (ALL,I,IND) (ALL,Z,Proxy)
 TCRD(I,Z)=FINDEM("TC","D",I,Z);

 COEFFICIENT (ALL,I,IND)  (ALL,Z,Proxy)    TCRI(I,Z)
 !DOMESTIC CONSUMPTION FOR Proxy COUNTRIES!;
 FORMULA (ALL,I,IND) (ALL,Z,Proxy)
 TCRI(I,Z)=FINDEM("C","IM",I,Z);

COEFFICIENT (ALL,I,IND) (ALL,Z,Proxy)  CTL(I,Z)
! TOTAL CONSUMPTION !  ;
 FORMULA (ALL,I,IND)  (ALL,Z,Proxy)
 CTL(I,Z) = DCONS(I,Z) + ICONS(I,Z)
        + TCRD(I,Z) + TCRI(I,Z) ;


!SHARES FOR CONSUMPTION!

COEFFICIENT (ALL,I,Food_Prod) (ALL,X,ROFWA)    foodShare(I,X);
FORMULA  (ALL,I,Food_Prod) (ALL,X,Af_MidE_asia )
foodShare(I,X)=CTL(I,"Thailand")/SUM(G,Food_Prod,CTL(G,"Thailand"));
FORMULA  (ALL,I,Food_Prod) (ALL,X,Latin_Am)
foodShare(I,X)=CTL(I,"KOREA")/SUM(G,Food_Prod,CTL(G,"KOREA"));
FORMULA  (ALL,I,Food_Prod) (ALL,X,West_Europe)
foodShare(I,X)=CTL(I,"EC")/SUM(G,Food_Prod,CTL(G,"EC"));

COEFFICIENT (ALL,I,TCF) (ALL,X,ROFWA)    TCFshare(I,X);
FORMULA  (ALL,I,TCF) (ALL,X,Af_MidE_asia )
TCFshare(I,X)=CTL(I,"Thailand")/SUM(G,TCF,CTL(G,"Thailand"));
FORMULA  (ALL,I,TCF) (ALL,X,Latin_Am)
TCFshare(I,X)=CTL(I,"KOREA")/SUM(G,TCF,CTL(G,"KOREA"));
FORMULA  (ALL,I,TCF) (ALL,X,West_Europe)
TCFshare(I,X)=CTL(I,"EC")/SUM(G,TCF,CTL(G,"EC"));

COEFFICIENT (ALL,I,Fuel) (ALL,X,ROFWA)    fuelShare(I,X);
FORMULA  (ALL,I,Fuel) (ALL,X,Af_MidE_asia )
fuelShare(I,X)=CTL(I,"Thailand")/SUM(G,Fuel,CTL(G,"Thailand"));
FORMULA  (ALL,I,Fuel) (ALL,X,Latin_Am)
fuelShare(I,X)=CTL(I,"KOREA")/SUM(G,Fuel,CTL(G,"KOREA"));
FORMULA  (ALL,I,Fuel) (ALL,X,West_Europe)
fuelShare(I,X)=CTL(I,"EC")/SUM(G,Fuel,CTL(G,"EC"));

COEFFICIENT (ALL,I,Misc) (ALL,X,ROFWA)    miscShare(I,X);
FORMULA  (ALL,I,Misc) (ALL,X,Af_MidE_asia )
```

```
miscShare(I,X)=CTL(I,"Thailand")/SUM(G,Misc,CTL(G,"Thailand"));
FORMULA  (ALL,I,Misc) (ALL,X,Latin_Am)
miscShare(I,X)=CTL(I,"KOREA")/SUM(G,Misc,CTL(G,"KOREA"));
FORMULA  (ALL,I,Misc) (ALL,X,West_Europe)
miscShare(I,X)=CTL(I,"EC")/SUM(G,Misc,CTL(G,"EC"));


!APPLY CONSUMPTION SHARES TO UN DATA!

COEFFICIENT (ALL,I,IND)  (ALL,X,ROFW)              CROFW(I,X);
FORMULA (ALL,I,Food_Prod)  (ALL,X,ROFWA)
CROFW(I,X)=foodShare(I,X)*Pri_Cons("UNfood",X)*(RWUSGDP(X)/ROFWGDP(X));
FORMULA (ALL,I,Food_Prod) (ALL,X,ROFWB)
CROFW(I,X)=CTL(I,"KOREA")*(RWUSGDP(X)/GDPFCL("KOREA"));

FORMULA (ALL,I,Bev_Tob) (ALL,X,ROFWA)
CROFW(I,X)= Pri_Cons("UNBev_Tob",X)*(RWUSGDP(X)/ROFWGDP(X));
FORMULA (ALL,I,Bev_Tob) (ALL,X,ROFWB)
CROFW(I,X)=CTL(I,"KOREA")*(RWUSGDP(X)/GDPFCL("KOREA"));

FORMULA (ALL,I,TCF) (ALL,X,ROFWA)
CROFW(I,X)=TCFshare(I,X)*Pri_Cons("UN_TCF",X)*(RWUSGDP(X)/ROFWGDP(X));
FORMULA (ALL,I,TCF) (ALL,X,ROFWB)
CROFW(I,X)=CTL(I,"KOREA")*(RWUSGDP(X)/GDPFCL("KOREA"));

FORMULA (ALL,I,Fuel)  (ALL,X,ROFWA)
CROFW(I,X)=fuelShare(I,X)*Pri_Cons("UNfuel",X)*(RWUSGDP(X)/ROFWGDP(X));
FORMULA (ALL,I,Fuel) (ALL,X,ROFWB)
CROFW(I,X)=CTL(I,"KOREA")*(RWUSGDP(X)/GDPFCL("KOREA"));

FORMULA  (ALL,I,Dwellings) (ALL,X,ROFWA)
CROFW(I,X)=Pri_Cons("UNdwellings",X)*(RWUSGDP(X)/ROFWGDP(X));
FORMULA  (ALL,I,Dwellings) (ALL,X,ROFWB)
CROFW(I,X)=CTL(I,"KOREA")*(RWUSGDP(X)/GDPFCL("KOREA"));

FORMULA  (ALL,I,Transport) (ALL,X,ROFWA)
CROFW(I,X)=Pri_Cons("UNtransport",X)*(RWUSGDP(X)/ROFWGDP(X)) ;
FORMULA (ALL,I,Transport) (ALL,X,ROFWB)
CROFW(I,X)=CTL(I,"KOREA")*(RWUSGDP(X)/GDPFCL("KOREA")) ;

FORMULA (ALL,I,Misc) (ALL,X,ROFWA)
CROFW(I,X)=miscShare(I,X)*Pri_Cons("UNmisc",X)*(RWUSGDP(X)/ROFWGDP(X));
FORMULA (ALL,I,Misc) (ALL,X,ROFWB)
CROFW(I,X)=CTL(I,"KOREA")*(RWUSGDP(X)/GDPFCL("KOREA")) ;

FORMULA  (ALL,I,Trade) (ALL,X,ROFWA)
CROFW(I,X)=Pri_Cons("UNtrade",X)*(RWUSGDP(X)/ROFWGDP(X));
FORMULA (ALL,I,Trade) (ALL,X,ROFWB)
CROFW(I,X)=CTL(I,"KOREA")*(RWUSGDP(X)/GDPFCL("KOREA")) ;

COEFFICIENT  (ALL,I,IND)              CTROFW(I)
 !TOTAL CONSUMPTION FOR THE REST OF THE WORLD!;
```

```
FORMULA (ALL,I,IND)
CTROFW(I)=SUM(X,ROFWA,CROFW(I,X))
          +SUM(X,ROFWB,CROFW(I,X));



!COEFFICIENTS FOR GOVERNMENT CONSUMPTION!

COEFFICIENT (ALL,I,IND)  (ALL,Z,Proxy)          DGOV(I,Z)
 !DOMESTIC CONSUMPTION FOR Proxy COUNTRIES!;
 FORMULA (ALL,I,IND) (ALL,Z,Proxy)
 DGOV(I,Z) =FINDEM("G","D",I,Z);

 COEFFICIENT (ALL,I,IND)  (ALL,Z,Proxy)   IGOVS(I,Z)
 !DOMESTIC CONSUMPTION FOR Proxy COUNTRIES!;
 FORMULA (ALL,I,IND) (ALL,Z,Proxy)
 IGOVS(I,Z)=FINDEM("G","IM",I,Z);

 COEFFICIENT (ALL,I,IND)  (ALL,Z,Proxy)          TGRD(I,Z)
 !DOMESTIC CONSUMPTION FOR Proxy COUNTRIES!;
 FORMULA (ALL,I,IND) (ALL,Z,Proxy)
 TGRD(I,Z)=FINDEM("TG","D",I,Z);

 COEFFICIENT (ALL,I,IND)  (ALL,Z,Proxy)   TGRI(I,Z)
 !DOMESTIC CONSUMPTION FOR Proxy COUNTRIES!;
 FORMULA (ALL,I,IND) (ALL,Z,Proxy)
 TGRI(I,Z)=FINDEM("TG","IM",I,Z);

COEFFICIENT  (ALL,I,IND) (ALL,Z,Proxy)  GTL(I,Z)
!TOTAL GOVERNMENT CONSUMPTION!;
FORMULA (ALL,I,IND) (ALL,Z,Proxy)
GTL(I,Z)=DGOV(I,Z)+IGOVS(I,Z)+TGRD(I,Z)+TGRI(I,Z);


!SALTER SHARES FOR GOVERNMENT CONSUMPTION!

COEFFICIENT (ALL,I,agriculture) (ALL,X,ROFW)      agrShare(I,X);
FORMULA (ALL,I,agriculture) (ALL,X,Af_MidE_asia )
agrShare(I,X)=GTL(I,"Thailand")/SUM(G,agriculture,GTL(G,"Thailand"));
FORMULA (ALL,I,agriculture) (ALL,X,CPE_LatinAm)
agrShare(I,X)=GTL(I,"KOREA")/SUM(G,agriculture,GTL(G,"KOREA"));
FORMULA (ALL,I,agriculture) (ALL,X,West_Europe)
agrShare(I,X)=GTL(I,"EC")/SUM(G,agriculture,GTL(G,"EC"));

COEFFICIENT (ALL,I,man_mine_con) (ALL,X,ROFW)    manufShare(I,X);
FORMULA (ALL,I,man_mine_con) (ALL,X,Af_MidE_asia )
manufShare(I,X)=GTL(I,"Thailand")/SUM(G,man_mine_con,GTL(G,"Thailand"));
FORMULA (ALL,I,man_mine_con) (ALL,X,CPE_LatinAm)
manufShare(I,X)=GTL(I,"KOREA")/SUM(G,man_mine_con,GTL(G,"KOREA"));
FORMULA (ALL,I,man_mine_con) (ALL,X,West_Europe)
manufShare(I,X)=GTL(I,"EC")/SUM(G,man_mine_con,GTL(G,"EC"));
```

```
!APPLY SHARES TO UN DATA!

COEFFICIENT (ALL,I,IND) (ALL,X,ROFW)     GROFW(I,X);

FORMULA (ALL,I,OSgovt) (ALL,X,ROFWA)
GROFW(I,X)=SUM(h,UNgovService,Gov_Cons(h,X))*(RWUSGDP(X)/ROFWGDP(X));
FORMULA (ALL,I,OSgovt) (ALL,X,ROFWB)
GROFW(I,X)=GTL(I,"KOREA")*(RWUSGDP(X)/GDPFCL("KOREA"));

FORMULA  (ALL,I,Petrol) (ALL,X,ROFWA)
 GROFW(I,X)=Gov_Cons("UNpetrol_gov",X)*(RWUSGDP(X)/ROFWGDP(X));
FORMULA   (ALL,I,Petrol) (ALL,X,ROFWB)
 GROFW(I,X)=GTL(I,"KOREA")*(RWUSGDP(X)/GDPFCL("KOREA")) ;

FORMULA (ALL,I,agriculture) (ALL,X,ROFWA)
 GROFW(I,X)=agrShare(I,X)*Gov_Cons("UNagr_govt",X)*RWUSGDP(X)/ROFWGDP(X);
FORMULA (ALL,I,agriculture) (ALL,X,ROFWB)
GROFW(I,X)=GTL(I,"KOREA")* (RWUSGDP(X)/GDPFCL("KOREA"));

FORMULA (ALL,I,man_mine_con) (ALL,X,ROFWA)
GROFW(I,X)=manufShare(I,X)*Gov_Cons("UNmanuf_govt",X)*(RWUSGDP(X)/ROFWGDP(X));
FORMULA (ALL,I,man_mine_con) (ALL,X,ROFWB)
GROFW(I,X)=GTL(I,"KOREA")*(RWUSGDP(X)/GDPFCL("KOREA"));

FORMULA  (ALL,I,Trade) (ALL,X,ROFWA)
 GROFW(I,X)=Gov_Cons("UNtrade_govt",X)*(RWUSGDP(X)/ROFWGDP(X));
FORMULA (ALL,I,Trade) (ALL,X,ROFWA)
 GROFW(I,X)=GTL(I,"KOREA")*(RWUSGDP(X)/GDPFCL("KOREA"));

FORMULA (ALL,I,Dwellings) (ALL,X,ROFWA)
 GROFW(I,X)=Gov_Cons("UNdwell_govt",X)*(RWUSGDP(X)/ROFWGDP(X));
FORMULA (ALL,I,Dwellings) (ALL,X,ROFWA)
 GROFW(I,X)=GTL(I,"KOREA")*(RWUSGDP(X)/GDPFCL("KOREA"));

COEFFICIENT (ALL,I,IND)       GTROFW(I)
!TOTAL GOVERNMENT CONSUMPTION!;
FORMULA (ALL,I,IND)
GTROFW(I)=SUM(X,ROFW,GROFW(I,X));


    !COEFFICIENTS FOR INVESTMENT!

 COEFFICIENT (ALL,I,IND)  (ALL,Z,Proxy)    DINV(I,Z)
 !DOMESTIC CONSUMPTION FOR Proxy COUNTRIES!;
 FORMULA (ALL,I,IND) (ALL,Z,Proxy)
 DINV(I,Z)=FINDEM("I","D",I,Z);

 COEFFICIENT (ALL,I,IND)  (ALL,Z,Proxy)    IINVS(I,Z)
 !DOMESTIC CONSUMPTION FOR Proxy COUNTRIES!;
 FORMULA (ALL,I,IND) (ALL,Z,Proxy)
```

```
 IINVS(I,Z)=FINDEM("I","IM",I,Z);

 COEFFICIENT (ALL,I,IND)  (ALL,Z,Proxy)    TIRD(I,Z)
 !DOMESTIC CONSUMPTION FOR Proxy COUNTRIES!;
 FORMULA (ALL,I,IND) (ALL,Z,Proxy)
 TIRD(I,Z)=FINDEM("TI","D",I,Z);

 COEFFICIENT (ALL,I,IND)  (ALL,Z,Proxy)    TIRI(I,Z)
 !DOMESTIC CONSUMPTION FOR Proxy COUNTRIES!;
 FORMULA (ALL,I,IND) (ALL,Z,Proxy)
 TIRI(I,Z)=FINDEM("TI","IM",I,Z);

 COEFFICIENT (ALL,I,IND) (ALL,Z,Proxy)  INVTL(I,Z)
! TOTAL INVESTMENT !    ;
 FORMULA (ALL,I,IND) (ALL,Z,Proxy)
 INVTL(I,Z)= DINV(I,Z) +IINVS(I,Z)
        + TIRD(I,Z) + TIRI(I,Z)  ;

COEFFICIENT (ALL,I,IND) (ALL,X,ROFW)        IROFW(I,X);
FORMULA (ALL,I,IND) (ALL,X,Af_MidE_asia )
IROFW(I,X)= INVTL(I,"Thailand")*(RWUSGDP(X)/GDPFCL("Thailand"));
FORMULA (ALL,I,IND) (ALL,X,Latin_Am)
IROFW(I,X)=INVTL(I,"KOREA")* (RWUSGDP(X)/GDPFCL("KOREA"));
FORMULA (ALL,I,IND) (ALL,X,West_Europe)
IROFW(I,X)=INVTL(I,"EC")* (RWUSGDP(X)/GDPFCL("EC"));


COEFFICIENT (ALL,I,IND)    ITROFW(I)
!TOTAL INVESTMENT BY INDUSTRY!;
FORMULA (ALL,I,IND)
ITROFW(I)=SUM(X,ROFW,IROFW(I,X));



!-----------------------------------------------------------------
              WRITE STATEMENTS
 -----------------------------------------------------------------!
WRITE CTROFW TO FILE ROWD92 HEADER "CON";
WRITE GTROFW TO FILE ROWD92 HEADER "GOVT";
WRITE ITROFW TO FILE ROWD92 HEADER "INV";
```

ROWVA.STI

```
disp1
rw92.dat
ec.upd
th.upd
kr.upd
in.upd
va.out
```

ROWVA.TAB

```
!-------------------------------------------------------------------------!
!                                                  !
!                                                  !
!                THE SALTER MODEL                          !
!          ---------------------------------               !
!This file calculates value added, output, labour and capital for the rest !
!of the world (ROW).                                !
!-------------------------------------------------------------------------!

!*************************************************************************!
!                     FILES                        !
!*************************************************************************!
 FILE UNIO #FILE CONTAINING ALL UN I/O INFORMATION#;
 FILE EC #FILE CONTAINING EC DATA#;
 FILE Thailand #FILE CONTAINING Thailand DATA#;
 FILE KOREA #FILE CONTAINING KOREAN DATA#;
 FILE Indonesia;

 FILE (NEW) ROWP92 # ALL ROW PRODUCTION INFORMATION #;

!*************************************************************************!
!                     SETS                         !
!*************************************************************************!
SET IND (IND1 - IND37) ;

SET SEC (AUSTRALIA,
NZ,
CANADA,
USA,
JAPAN,
KOREA,
EC,
Indonesia,
Malaysia,
Philippines,
Singapore,
Thailand,
China,
```

```
HK,
Taiwan,
ROFW) ;


SET REG (AUSTRALIA,
NZ,
CANADA,
USA,
JAPAN,
KOREA,
EC,
Indonesia,
Malaysia,
Philippines,
Singapore,
Thailand,
China,
HK,
Taiwan);
SUBSET REG IS SUBSET OF SEC;

SET ROFW (AFRICA,
MIDEAST,
SASIA,
LATAMER,
CPE,
WEURO) ;

SET AfAsLatCpeEu(AFRICA,
SASIA,
LATAMER,
CPE,
WEURO);
SUBSET AfAsLatCpeEu IS SUBSET OF ROFW;

SET Afr_Sasia(AFRICA,
SASIA);
SUBSET Afr_Sasia IS SUBSET OF ROFW;
SUBSET Afr_Sasia IS SUBSET OF AfAsLatCpeEu;

SET Afr(AFRICA);
SUBSET Afr IS SUBSET OF ROFW;
SUBSET Afr IS SUBSET OF AfAsLatCpeEu;

SET LatAm_CPE(LATAMER,
CPE);
SUBSET LatAm_CPE IS SUBSET OF ROFW;
SUBSET LatAm_CPE IS SUBSET OF AfAsLatCpeEu;

SET MidE_Sasia(MIDEAST,
```

```
SASIA);
SUBSET MidE_Sasia IS SUBSET OF ROFW;

SET AfLatWeuCPE(AFRICA,
LATAMER,
CPE,
WEURO);
SUBSET AfLatWeuCPE IS SUBSET OF ROFW;

SET Afr_LAm_Weu (AFRICA,LATAMER,WEURO);
SUBSET Afr_LAm_Weu IS SUBSET OF ROFW;
SUBSET Afr_LAm_Weu IS SUBSET OF AfLatWeuCPE;

SET CentPlan (CPE);
SUBSET CentPlan IS SUBSET OF ROFW;

SET WEurope (WEURO);
SUBSET WEurope IS SUBSET OF ROFW;
SUBSET WEurope IS SUBSET OF AfAsLatCpeEu;

SET MidE(MIDEAST);
SUBSET MidE IS SUBSET OF ROFW;

SET Proxy(EC,
Thailand,
KOREA);
SUBSET Proxy IS SUBSET OF SEC;
SUBSET Proxy IS SUBSET OF REG;

SET PRN(IN,TI);

SET FO(LAB,LN,K);

SET SCR(D,IM);


SET WBT(WBagr,
WBmining,
WBmanuf,
WBservices);

SET agr(IND1 - IND8);
SUBSET agr IS SUBSET OF IND;

SET mining(IND9 - IND12);
SUBSET mining IS SUBSET OF IND;

SET otherServs(IND35 - IND37);
SUBSET otherServs IS SUBSET OF IND;

SET OSgovPriv(IND35-IND36);
SUBSET OSgovPriv IS SUBSET OF IND;
```

SET OSdwelling(IND37);
SUBSET OSdwelling IS SUBSET OF IND;

!UNIDO SET CONTAINING ELEMENTS FOR MANUFACTURING AND SERVICES!

SET UNmanServ(UNfood,
UNbev,
UNtextiles,
UNapparel,
UNleather,
UNwood,
UNpaper,
UNchem,
UNpetrol,
UNnonMet,
UNiron,
UNnonFerr,
UNfabMet,
UNtrans,
UNothMach,
UNothManuf,
UNewg,
UNconstr,
UNsales,
UNtrade);

!UNIDO SET CONTAINING ELEMENTS FOR MANUFACTURING!

SET UNmanuf(UNfood,
UNbev,
UNtextiles,
UNapparel,
UNleather,
UNwood,
UNpaper,
UNchem,
UNpetrol,
UNnonMet,
UNiron,
UNnonFerr,
UNfabMet,
UNtrans,
UNothMach,
UNothManuf);
SUBSET UNmanuf IS SUBSET OF UNmanServ;

SET manuf_serv(IND13 - IND33);
SUBSET manuf_serv IS SUBSET OF IND;

```
!SET FOR FOOD!

SET foodProd (IND13 - IND16);
SUBSET foodProd IS SUBSET OF IND;

!UNIDO SET CONTAINING SERVICES!

SET UNserv(UNewg,
UNconstr,
UNsales,
UNtrade);
SUBSET UNserv IS SUBSET OF UNmanServ;

SET OSandTrade(IND34 - IND37);
SUBSET OSandTrade IS SUBSET OF IND;

SET Bev_Tob (IND17);
SUBSET Bev_Tob IS SUBSET OF IND;

SET Textiles (IND18);
SUBSET Textiles IS SUBSET OF IND;

SET Apparel (IND19);
SUBSET Apparel IS SUBSET OF IND;

SET Leather (IND20);
SUBSET Leather IS SUBSET OF IND;

SET Wood (IND21);
SUBSET Wood IS SUBSET OF IND;

SET Paper (IND22);
SUBSET Paper IS SUBSET OF IND;

SET Petrol (IND23);
SUBSET Petrol IS SUBSET OF IND;

SET Chem (IND24);
SUBSET Chem IS SUBSET OF IND;

SET NonMetal (IND25);
SUBSET NonMetal IS SUBSET OF IND;

SET Iron (IND26);
SUBSET Iron IS SUBSET OF IND;

SET NonFerr (IND27);
SUBSET NonFerr IS SUBSET OF IND;

SET FabMet (IND28);
SUBSET FabMet IS SUBSET OF IND;
```

```
SET Trans (IND29);
SUBSET Trans IS SUBSET OF IND;

SET Mach (IND30);
SUBSET Mach IS SUBSET OF IND;

SET OthManuf (IND31);
SUBSET OthManuf IS SUBSET OF IND;

SET EWG (IND32);
SUBSET EWG IS SUBSET OF IND;

SET Constr (IND33);
SUBSET Constr IS SUBSET OF IND;

SET Trade (IND34);
SUBSET Trade IS SUBSET OF IND;

!-----------------------------------------------------------------
                INPUT-OUTPUT DATABASE
 ----------------------------------------------------------------!
 !DEFINE COEFFICIENTS TO BE READ FROM SALTER!

COEFFICIENT (ALL,P,PRN) (ALL,SC,SCR) (ALL,I,IND) (ALL,J,IND)(ALL,Z,Proxy)
PR(P,SC,I,J,Z)
!INCLUDES SETS DEFINED TO READ FROM SINGLE COUNTRY DATABASE!;

COEFFICIENT (ALL,G,FO) (ALL,I,IND)  (ALL,Z,Proxy)
FINDO (G,I,Z)
!INCLUDES SETS DEFINED TO READ FROM SINGLE COUNTRY DATABASE!;


!COEFFICIENTS TO BE READ FROM WORLD BANK!

COEFFICIENT (ALL,I,WBT) (ALL,X,AfAsLatCpeEu)       wbVAinGDP(I,X)
!VALUE ADDED OVER GDP FOR REST OF THE WORLD !;

!COEFFICENTS TO BE READ FROM UNIDO!

COEFFICIENT (ALL,I,UNmanServ) (ALL,X,AfAsLatCpeEu)      UNvalueAdded(I,X)
!VALUE ADDED FOR REST OF THE WORLD!;

COEFFICIENT  (ALL,X,ROFW)                       RWUSGDP(X)
!GDP FOR REST OF WORLD REGIONS IN US$!;


!*****************************************************************************
                READING DATA BASE
 *****************************************************************************!
READ
```

```
   (ALL,I,IND)
FINDO("LN",I,"EC")
          FROM FILE EC HEADER "AI15";

(ALL,I,IND)
FINDO("LN",I,"Thailand")
          FROM FILE Thailand HEADER "AI15";

(ALL,I,IND)
FINDO("LN",I,"KOREA")
          FROM FILE KOREA HEADER "AI15";

(ALL,I,IND)
FINDO("LAB",I,"EC")
          FROM FILE EC HEADER "AI13";

(ALL,I,IND)
FINDO("LAB",I,"Thailand")
          FROM FILE EC HEADER "AI13";

(ALL,I,IND)
FINDO("LAB",I,"KOREA")
          FROM FILE KOREA HEADER "AI13";

(ALL,I,IND)
FINDO("K ",I,"EC")
          FROM FILE EC HEADER "AI14";


(ALL,I,IND)
FINDO("K",I,"Thailand")
          FROM FILE EC HEADER "AI14";
(ALL,I,IND)
FINDO("K",I,"KOREA")
          FROM FILE KOREA HEADER "AI14";

 (ALL,I,IND) (ALL,J,IND)
 PR("IN","D",I,J,"EC")
             FROM FILE EC HEADER "AI01";

             (ALL,I,IND) (ALL,J,IND)
 PR("IN","IM",I,J,"EC")
             FROM FILE EC HEADER "AI02";
             (ALL,I,IND) (ALL,J,IND)

 PR("IN","D",I,J,"Thailand")
             FROM FILE Thailand HEADER "AI01";

             (ALL,I,IND) (ALL,J,IND)
 PR("IN","IM",I,J,"Thailand")
             FROM FILE Thailand HEADER "AI02";
```

```
            (ALL,I,IND) (ALL,J,IND)
            PR("IN","D",I,J,"KOREA")
            FROM FILE KOREA HEADER "AI01";

            (ALL,I,IND) (ALL,J,IND)
             PR("IN","IM",I,J,"KOREA")
            FROM FILE KOREA HEADER "AI02";

            (ALL,I,IND) (ALL,J,IND)
PR("TI","D",I,J,"EC")
            FROM FILE EC HEADER "AI16";

            (ALL,I,IND) (ALL,J,IND)
PR("TI","IM",I,J,"EC")
            FROM FILE EC HEADER "AI17";
            (ALL,I,IND) (ALL,J,IND)

PR("TI","D",I,J,"Thailand")
            FROM FILE Thailand HEADER "AI16";

            (ALL,I,IND) (ALL,J,IND)
PR("TI","IM",I,J,"Thailand")
            FROM FILE Thailand HEADER "AI17";

            (ALL,I,IND) (ALL,J,IND)
            PR("TI","D",I,J,"KOREA")
            FROM FILE KOREA HEADER "AI16";

            (ALL,I,IND) (ALL,J,IND)
             PR("TI","IM",I,J,"KOREA")
            FROM FILE KOREA HEADER "AI17";


   (ALL,I,WBT) (ALL,X,AfAsLatCpeEu)
   wbVAinGDP(I,X)
         FROM FILE UNIO HEADER  "UN5" ;


   (ALL,I,UNmanServ) (ALL,X,AfAsLatCpeEu)
   UNvalueAdded(I,X)
            FROM FILE UNIO HEADER "UN6";

!READ GDP DATA FROM FILES!

   (ALL,X,ROFW)
   RWUSGDP(X)
        FROM FILE UNIO HEADER "UN4";

!Read data from the Indonesian database.  Used to estimate
oil and gas output of the Middle East !

COEFFICIENT
```

```
(all,i,ind)(all,j,ind)  IndonDINT(i,j);
(all,i,ind)(all,j,ind)  IndonIINT(i,j);
(all,i,ind)(all,j,ind)  IndonTRD(i,j);
(all,i,ind)(all,j,ind)  IndonTRI(i,j);
(all,i,ind)  IndonLAB(i);
(all,i,ind)  IndonCAP(i);
(all,i,ind)  IndonLAND(i);

READ
   (ALL,I,IND)(all,j,ind)
 IndonDINT(I,j)
         FROM FILE Indonesia HEADER "AI01";
   (ALL,I,IND)(all,j,ind)
 IndonIINT(I,j)
         FROM FILE Indonesia HEADER "AI02";
   (ALL,I,IND)(all,j,ind)
 IndonTRD(I,j)
         FROM FILE Indonesia HEADER "AI16";
   (ALL,I,IND)(all,j,ind)
 IndonTRI(I,j)
         FROM FILE Indonesia HEADER "AI17";
   (ALL,I,IND)
 IndonLAB(I)
         FROM FILE Indonesia HEADER "AI13";
   (ALL,I,IND)
 IndonCAP(I)
         FROM FILE Indonesia HEADER "AI14";
   (ALL,I,IND)
 IndonLAND(I)
         FROM FILE Indonesia HEADER "AI15";


!*********************************************************************

            WIDELY USED DATABASE AGGREGATES
**********************************************************************!

!---------------------------------------------------------------
               FACTOR AGGREGATES
 ---------------------------------------------------------------!

COEFFICIENT (ALL,J,IND) (ALL,Z,Proxy) LAB(J,Z)
 !LABOUR RETURNS FROM SINGLE COUNTRY DATA!;
 FORMULA (ALL,J,IND) (ALL,Z,Proxy)
 LAB(J,Z)=FINDO("LAB",J,Z);

 COEFFICIENT (ALL,J,IND) (ALL,Z,Proxy) LAND(J,Z)
 !LAND RETURNS FROM SINGLE COUNTRY DATA!;
 FORMULA (ALL,J,IND) (ALL,Z,Proxy)
 LAND(J,Z)=FINDO("LN",J,Z);

 COEFFICIENT (ALL,J,IND) (ALL,Z,Proxy) CAP(J,Z)
```

```
 !CAPITAL RETURNS FROM SINGLE COUNTRY DATA!;
 FORMULA (ALL,J,IND) (ALL,Z,Proxy)
 CAP(J,Z)=FINDO("K",J,Z);

COEFFICIENT (ALL,Z,Proxy) LTT(Z)
! TOTAL LABOUR USAGE ! ;
 FORMULA (ALL,Z,Proxy)
 LTT(Z) = SUM(J,IND,LAB(J,Z)) ;

 COEFFICIENT (ALL,Z,Proxy) KTT(Z)
! TOTAL CAPITAL USAGE ! ;
 FORMULA (ALL,Z,Proxy)
 KTT(Z) = SUM(J,IND,CAP(J,Z)) ;

 COEFFICIENT (ALL,Z,Proxy) MTT(Z)
! TOTAL LAND USAGE ! ;
 FORMULA (ALL,Z,Proxy)
 MTT(Z) = SUM(J,IND,LAND(J,Z)) ;

 COEFFICIENT (ALL,Z,Proxy) GDPFCL(Z)
! GDPFC AT FACTOR COST ! ;
 FORMULA (ALL,Z,Proxy)
 GDPFCL(Z) = LTT(Z) + KTT(Z) + MTT(Z) ;



 COEFFICIENT (ALL,J,IND) (ALL,Z,Proxy) GDPFCS(J,Z)
! GDPFC AT FACTOR COST ! ;
 FORMULA (ALL,J,IND) (ALL,Z,Proxy)
 GDPFCS(J,Z) =LAB (J,Z) + CAP(J,Z) + LAND(J,Z) ;

 COEFFICIENT (ALL,J,IND) IndonGDPFCS(J)
! GDPFC AT FACTOR COST ! ;
 FORMULA (ALL,J,IND)
 IndonGDPFCS(J) = IndonLAB (J) + IndonCAP(J) + IndonLAND(J) ;

 COEFFICIENT IndonGDPFCL;
 FORMULA
 IndonGDPFCL = SUM(J,IND,IndonGDPFCS(J));

 !COEFFICIENTS TO CALCULATE VALUE ADDED!

 COEFFICIENT (ALL,X,AfAsLatCpeEu)          UNmanufVA(X)
 !TOTAL VALUE ADDED FOR MANUFACTURING FROM UNIDO!;
 FORMULA (ALL,X,AfAsLatCpeEu)
 UNmanufVA(X)=SUM(I,UNmanuf,UNvalueAdded(I,X));

 COEFFICIENT (ALL,X,AfAsLatCpeEu)          UNservVA(X)
 !TOTAL VALUE ADDED FOR SERVICES FROM NATIONAL ACCOUNTS!;
 FORMULA (ALL,X,AfAsLatCpeEu)
 UNservVA(X)=SUM(I,UNserv,UNvalueAdded(I,X));
```

```
COEFFICIENT (ALL,I,IND) (ALL,X,ROFW)        VA(I,X);
FORMULA (ALL,I,agr) (ALL,X,Afr_Sasia)
VA(I,X)=(wbVAinGDP("WBagr",X))
        *(GDPFCS(I,"Thailand")/SUM(G,agr,GDPFCS(G,"Thailand")))
        *(RWUSGDP(X));
FORMULA (ALL,I,agr) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))
        *(RWUSGDP(X));
FORMULA (ALL,I,agr) (ALL,X,LatAm_CPE)
VA(I,X)=(wbVAinGDP("WBagr",X))
        *(GDPFCS(I,"KOREA")/SUM(G,agr,GDPFCS(G,"KOREA")))
        *(RWUSGDP(X));
FORMULA (ALL,I,agr) (ALL,X,WEurope)
VA(I,X)=(wbVAinGDP("WBagr",X))
        *(GDPFCS(I,"EC")/SUM(G,agr,GDPFCS(G,"EC")))
        *(RWUSGDP(X));

FORMULA (ALL,I,mining) (ALL,X,Afr)
VA(I,X)=(wbVAinGDP("WBmining",X))
        *(GDPFCS(I,"Thailand")/SUM(G,mining,GDPFCS(G,"Thailand")))
        *(RWUSGDP(X));
!%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&!
!Use Indonesian industry structure for mining
so that output of oil and gas in ROW is greater
than the value of exports reported by the trade database.!
!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&!
FORMULA (ALL,I,mining) (ALL,X,MidE_Sasia)
VA(I,X)=(IndonGDPFCS(I)/IndonGDPFCL)
        *(RWUSGDP(X));

FORMULA (ALL,I,mining) (ALL,X,LatAm_CPE)
VA(I,X)=(wbVAinGDP("WBmining",X))
        *(GDPFCS(I,"KOREA")/SUM(G,mining,GDPFCS(G,"KOREA")))
        *(RWUSGDP(X));
FORMULA (ALL,I,mining) (ALL,X,WEurope)
VA(I,X)=(wbVAinGDP("WBmining",X))
        *(GDPFCS(I,"EC")/SUM(G,mining,GDPFCS(G,"EC")))
        *(RWUSGDP(X));

FORMULA (ALL,I,foodProd) (ALL,X,Afr_Sasia)
VA(I,X)=(wbVAinGDP("WBmanuf",X))*(UNvalueAdded("UNfood",X)
        /SUM(G,UNmanuf,UNvalueAdded(G,X)))
        *(GDPFCS(I,"Thailand")/SUM(G,foodProd,GDPFCS(G,"Thailand")))
        *(RWUSGDP(X));
FORMULA (ALL,I,foodProd) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))
        *(RWUSGDP(X));
FORMULA (ALL,I,foodProd) (ALL,X,LatAm_CPE)
VA(I,X)=(wbVAinGDP("WBmanuf",X))*(UNvalueAdded("UNfood",X)
        /SUM(G,UNmanuf,UNvalueAdded(G,X)))
        *GDPFCS(I,"KOREA")/SUM(G,foodProd,GDPFCS(G,"KOREA"))
```

```
        *RWUSGDP(X);
FORMULA (ALL,I,foodProd) (ALL,X,WEurope)
VA(I,X)=(wbVAinGDP("WBmanuf",X))*(UNvalueAdded("UNfood",X)
        /SUM(G,UNmanuf,UNvalueAdded(G,X)))
        *(GDPFCS(I,"EC")/SUM(G,foodProd,GDPFCS(G,"EC"))
        *RWUSGDP(X));

FORMULA  (ALL,I,Bev_Tob) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
        *(UNvalueAdded("UNbev",X)/UNmanufVA(X))*(RWUSGDP(X));
FORMULA  (ALL,I,Bev_Tob) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));

FORMULA  (ALL,I,Textiles) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
        *(UNvalueAdded("UNtextiles",X)/UNmanufVA(X))*(RWUSGDP(X));
FORMULA  (ALL,I,Textiles) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))
        *(RWUSGDP(X));

FORMULA  (ALL,I,Apparel) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
        *(UNvalueAdded("UNapparel",X)/UNmanufVA(X))*(RWUSGDP(X));
FORMULA  (ALL,I,Apparel) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));

FORMULA  (ALL,I,Leather) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
        *(UNvalueAdded("UNleather",X)/UNmanufVA(X))*(RWUSGDP(X));
FORMULA  (ALL,I,Leather) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));

FORMULA  (ALL,I,Wood) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
        *(UNvalueAdded("UNwood",X)/UNmanufVA(X))*(RWUSGDP(X));
FORMULA (ALL,I,Wood) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));

FORMULA  (ALL,I,Paper) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
        * (UNvalueAdded("UNpaper",X)/UNmanufVA(X))*(RWUSGDP(X));
FORMULA  (ALL,I,Paper) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));

FORMULA  (ALL,I,Chem) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
        *(UNvalueAdded("UNchem",X)/UNmanufVA(X))*(RWUSGDP(X));
FORMULA  (ALL,I,Chem) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));

FORMULA  (ALL,I,Petrol) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
```

```
        *(UNvalueAdded("UNpetrol",X)/UNmanufVA(X))*(RWUSGDP(X));
FORMULA  (ALL,I,Petrol) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));

FORMULA  (ALL,I,NonMetal) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
        *(UNvalueAdded("UNnonMet",X)/UNmanufVA(X))*(RWUSGDP(X));
FORMULA  (ALL,I,NonMetal) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))
        *(RWUSGDP(X));

FORMULA  (ALL,I,Iron) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
        *(UNvalueAdded("UNiron",X)/UNmanufVA(X))*RWUSGDP(X);
FORMULA  (ALL,I,Iron) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand")*RWUSGDP(X));

FORMULA (ALL,I,NonFerr) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
        *(UNvalueAdded("UNnonFerr",X)/UNmanufVA(X))*(RWUSGDP(X));
FORMULA (ALL,I,NonFerr) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));

FORMULA (ALL,I,FabMet) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
        *(UNvalueAdded("UNfabMet",X)/UNmanufVA(X))*(RWUSGDP(X));
FORMULA (ALL,I,FabMet) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));

FORMULA  (ALL,I,Trans) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
        *(UNvalueAdded("UNtrans",X)/UNmanufVA(X))*(RWUSGDP(X));
FORMULA  (ALL,I,Trans) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));

FORMULA  (ALL,I,Mach) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
        *(UNvalueAdded("UNothMach",X)/UNmanufVA(X))*(RWUSGDP(X));
FORMULA  (ALL,I,Mach) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));

FORMULA  (ALL,I,OthManuf) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBmanuf",X))
        *(UNvalueAdded("UNothManuf",X)/UNmanufVA(X))*(RWUSGDP(X));
FORMULA  (ALL,I,OthManuf) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));

FORMULA  (ALL,I,EWG) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBservices",X))*(UNvalueAdded("UNewg",X)/UNservVA(X))
        *(RWUSGDP(X));
FORMULA  (ALL,I,EWG) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));
```

```
FORMULA (ALL,I,Constr) (ALL,X,AfAsLatCpeEu)
VA(I,X)=(wbVAinGDP("WBservices",X))
        *(UNvalueAdded("UNconstr",X)/UNservVA(X))*(RWUSGDP(X));
FORMULA (ALL,I,Constr) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));

FORMULA (ALL,I,otherServs) (ALL,X,Afr_Sasia)
VA(I,X)= (wbVAinGDP("WBservices",X))*(UNvalueAdded("UNsales",X)/UNservVA(X))
        *(GDPFCS(I,"Thailand")/SUM(G,otherServs,GDPFCS(G,"Thailand"))
        *RWUSGDP(X));
FORMULA (ALL,I,otherServs) (ALL,X,LatAm_CPE)
VA(I,X)= (wbVAinGDP("WBservices",X))*(UNvalueAdded("UNsales",X)/UNservVA(X))
        *(GDPFCS(I,"KOREA")/SUM(G,otherServs,GDPFCS(G,"KOREA"))
        *RWUSGDP(X));
FORMULA (ALL,I,otherServs) (ALL,X,WEurope)
VA(I,X)= (wbVAinGDP("WBservices",X))*(UNvalueAdded("UNconstr",X)/UNservVA(X))
        *(GDPFCS(I,"EC")/SUM(G,otherServs,GDPFCS(G,"EC"))
        *RWUSGDP(X));
FORMULA (ALL,I,otherServs) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/SUM(G,otherServs,GDPFCS(G,"Thailand")))
        *RWUSGDP(X);

FORMULA  (ALL,I,Trade) (ALL,X,AfAsLatCpeEu)
VA(I,X)= (wbVAinGDP("WBservices",X))*(UNvalueAdded("UNtrade",X)/UNservVA(X))
        *(RWUSGDP(X));
FORMULA  (ALL,I,Trade) (ALL,X,MidE)
VA(I,X)=(GDPFCS(I,"Thailand")/GDPFCL("Thailand"))*(RWUSGDP(X));



!*************************************************************************!
!find output!

!*************************************************************************!
!                    SETS                         !
!*************************************************************************!
!SET OF DATA FROM NATIONAL ACCOUNTS!

SET NAcommod(NAagr,
NAmining,
NAewg,
NAconstr,
NAsales,
NAtrade);

!-------------------------------------------------------------
              INPUT-OUTPUT DATABASE
 -------------------------------------------------------------!

!DEFINE COEFFICIENTS TO BE READ FROM NATIONAL ACCOUNTS!
```

```
COEFFICIENT (ALL,I,NAcommod)   (ALL,X,AfLatWeuCPE)   NAvalueAdded(I,X);

COEFFICIENT   (ALL,I,NAcommod) (ALL,X,AfLatWeuCPE)   NAoutput(I,X);

COEFFICIENT (ALL,I,UNmanuf) (ALL,X, AfAsLatCpeEu)     UNoutput(I,X);


!*************************************************************************
                  READING DATA BASE
*************************************************************************!
READ
!READ FROM NATIONAL ACCOUNTS!


   (ALL,I,NAcommod) (ALL,X,AfLatWeuCPE)
   NAoutput(I,X)
            FROM FILE UNIO HEADER "UN7";

(ALL,I,NAcommod) (ALL,X,AfLatWeuCPE)
  NAvalueAdded(I,X)
            FROM FILE UNIO HEADER "UN9";



!READ FROM UNIDO!
(ALL,I,UNmanuf) (ALL,X,AfAsLatCpeEu)
      UNoutput(I,X)
              FROM FILE UNIO HEADER "UN8";


!*************************************************************************
            WIDELY USED DATABASE AGGREGATES
*************************************************************************!

COEFFICIENT (ALL,I,IND) (ALL,J,IND) (ALL,Z,Proxy) DINT(I,J,Z)
 !DOMESTIC INT INPUTS FROM SINGLE COUNTRY DATA!;
 FORMULA (ALL,I,IND) (ALL,J,IND) (ALL,Z,Proxy)
 DINT(I,J,Z)=PR("IN","D",I,J,Z);


 COEFFICIENT(ALL,I,IND) (ALL,J,IND) (ALL,Z,Proxy) IINT(I,J,Z)
 !IMPORTED INT INPUTS FROM SINGLE COUNTRY DATA!;
 FORMULA (ALL,I,IND) (ALL,J,IND) (ALL,Z,Proxy)
 IINT(I,J,Z)=PR("IN","IM",I,J,Z);

 COEFFICIENT (ALL,I,IND) (ALL,J,IND) (ALL,Z,Proxy) TRD(I,J,Z)
 !TAXES ON DOMESTIC INT INPUTS  FROM SINGLE COUNTRY DATA!;
 FORMULA (ALL,I,IND) (ALL,J,IND) (ALL,Z,Proxy)
 TRD(I,J,Z)=PR("TI","D",I,J,Z);

 COEFFICIENT (ALL,I,IND) (ALL,J,IND) (ALL,Z,Proxy) TRI(I,J,Z)
```

```
 !IMPORTED INT INPUTS FROM SINGLE COUNTRY DATA!;
 FORMULA (ALL,I,IND) (ALL,J,IND) (ALL,Z,Proxy)
 TRI(I,J,Z)=PR("TI","IM",I,J,Z);

COEFFICIENT (ALL,J,IND)(ALL,Z,Proxy)                COSTINP(J,Z)
! TOTAL COSTS OF INPUTS TO INDUSTRY EXCLUDING INDIRECT TAXES!   ;
 FORMULA (ALL,J,IND)(ALL,Z,Proxy)
 COSTINP(J,Z) =SUM(I,IND,DINT(I,J,Z)) + SUM(I,IND,IINT(I,J,Z))
        +SUM(I,IND,TRD(I,J,Z)) +SUM(I,IND,TRI(I,J,Z))
        + LAB(J,Z) + CAP(J,Z) + LAND(J,Z) ;

COEFFICIENT (ALL,J,IND)                 IndonCOSTINP(J)
! TOTAL COSTS OF INPUTS TO INDUSTRY EXCLUDING INDIRECT TAXES!   ;
 FORMULA (ALL,J,IND)
 IndonCOSTINP(J) =SUM(I,IND,IndonDINT(I,J)) + SUM(I,IND,IndonIINT(I,J))
        +SUM(I,IND,IndonTRD(I,J)) +SUM(I,IND,IndonTRI(I,J))
        + IndonLAB(J) + IndonCAP(J) + IndonLAND(J) ;




!output for ROW!
COEFFICIENT (ALL,I,IND) (ALL,X,ROFW)            ROFWO(I,X);

FORMULA (ALL,I,agr) (ALL,X,AfLatWeuCPE)
ROFWO(I,X)=NAoutput("NAagr",X)/NAvalueAdded("NAagr",X)*VA(I,X);
FORMULA (ALL,I,agr) (ALL,X,MidE_Sasia)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA (ALL,I,mining) (ALL,X,AfLatWeuCPE)
ROFWO(I,X)=NAoutput("NAmining",X)/NAvalueAdded("NAmining",X)*VA(I,X);
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%!
!Use Indonesian data for Mid East & S Asia to ensure that
the value of mining exports from ROW as reported in the trade
database is not greater than mining output!
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%!
FORMULA (ALL,I,mining)(ALL,X,MidE_Sasia)
ROFWO(I,X)=IndonCOSTINP(I) /IndonGDPFCS(I)*VA(I,X);

coefficient out11;
formula out11=sum(x,rofw,rofwo("ind11",X));
display out11;
coefficient out10;
formula out10=sum(x,rofw,rofwo("ind10",X));
display out10;

FORMULA (ALL,I,foodProd) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNfood",X)/UNvalueAdded("UNfood",X)*VA(I,X);
FORMULA (ALL,I,foodProd) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);
```

```
FORMULA  (ALL,I,Bev_Tob) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNbev",X)/UNvalueAdded("UNbev",X)*VA(I,X);
FORMULA  (ALL,I,Bev_Tob) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Textiles) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNtextiles",X)/UNvalueAdded("UNtextiles",X)*VA(I,X);
FORMULA  (ALL,I,Textiles) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Apparel) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNapparel",X)/UNvalueAdded("UNapparel",X)*VA(I,X);
FORMULA  (ALL,I,Apparel) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Leather) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNleather",X)/UNvalueAdded("UNleather",X)*VA(I,X);
FORMULA (ALL,I,Leather) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Wood) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNwood",X)/UNvalueAdded("UNwood",X)*VA(I,X);
FORMULA  (ALL,I,Wood) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X) ;

FORMULA  (ALL,I,Paper) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNpaper",X)/UNvalueAdded("UNpaper",X)*VA(I,X);
FORMULA (ALL,I,Paper) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Chem) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNchem",X)/UNvalueAdded("UNchem",X)*VA(I,X);
FORMULA  (ALL,I,Chem) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Petrol) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNpetrol",X)/UNvalueAdded("UNpetrol",X)*VA(I,X);
FORMULA  (ALL,I,Petrol) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,NonMetal) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNnonMet",X)/UNvalueAdded("UNnonMet",X)*VA(I,X);
FORMULA  (ALL,I,NonMetal) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Iron) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNiron",X)/UNvalueAdded("UNiron",X)*VA(I,X);
FORMULA  (ALL,I,Iron) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,NonFerr) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNnonFerr",X)/UNvalueAdded("UNnonFerr",X)*VA(I,X);
```

```
FORMULA  (ALL,I,NonFerr) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,FabMet) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNfabMet",X)/UNvalueAdded("UNfabMet",X)*VA(I,X);
FORMULA  (ALL,I,FabMet) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Trans) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNtrans",X)/UNvalueAdded("UNtrans",X)*VA(I,X);
FORMULA  (ALL,I,Trans) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Mach) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNothMach",X)/UNvalueAdded("UNothMach",X)*VA(I,X) ;
FORMULA  (ALL,I,Mach) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,OthManuf) (ALL,X,AfAsLatCpeEu)
ROFWO(I,X)=UNoutput("UNothManuf",X)/UNvalueAdded("UNothManuf",X)*VA(I,X);
FORMULA (ALL,I,OthManuf) (ALL,X,MidE)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA (ALL,I,EWG) (ALL,X,AfLatWeuCPE)
ROFWO(I,X)=NAoutput("NAewg",X)/NAvalueAdded("NAewg",X)*VA(I,X);
FORMULA (ALL,I,EWG) (ALL,X,MidE_Sasia)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA (ALL,I,Constr) (ALL,X,AfLatWeuCPE)
ROFWO(I,X)=NAoutput("NAconstr",X)/NAvalueAdded("NAconstr",X)*VA(I,X);
FORMULA (ALL,I,Constr) (ALL,X,MidE_Sasia)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA (ALL,I,OSgovPriv) (ALL,X,AfLatWeuCPE)
ROFWO(I,X)=NAoutput("NAsales",X)/NAvalueAdded("NAsales",X)*VA(I,X);
FORMULA (ALL,I,OSdwelling)
ROFWO(I,"Africa")=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,"Africa");
FORMULA (ALL,I,OSdwelling)
ROFWO(I,"LatAmer")=COSTINP(I,"Korea")/GDPFCS(I,"Korea")*VA(I,"LatAmer");
FORMULA (ALL,I,OSdwelling)
ROFWO(I,"Weuro")=COSTINP(I,"EC")/GDPFCS(I,"EC")*VA(I,"Weuro");
FORMULA (ALL,I,OSdwelling)
ROFWO(I,"CPE")=COSTINP(I,"Korea")/GDPFCS(I,"Korea")*VA(I,"CPE");
FORMULA (ALL,I,otherServs) (ALL,X,MidE_Sasia)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Trade) (ALL,X,AfLatWeuCPE)
ROFWO(I,X)=NAoutput("NAtrade",X)/NAvalueAdded("NAtrade",X)*VA(I,X);
FORMULA (ALL,I,Trade) (ALL,X,MidE_Sasia)
ROFWO(I,X)=COSTINP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);
```

```
COEFFICIENT (ALL,I,IND)          TRLOFWO(I)
!OUTPUT RETURNS FOR THE REST OF THE WORLD BY INDUSTRY!;
FORMULA (ALL,I,IND)
TRLOFWO(I)=SUM(X,ROFW,ROFWO(I,X));




!*************************************************************************!
!find labour and capital!


!-----------------------------------------------------------------
                INPUT-OUTPUT DATABASE
 ----------------------------------------------------------------!

!DEFINE COEFFICIENTS TO BE READ FROM NATIONAL ACCOUNTS!


COEFFICIENT (ALL,I,NAcommod)   (ALL,X,Afr_LAm_Weu)      NAlabour(I,X)
!LABOUR RETURNS FOR COUNTRIES IN GROUP C!;

COEFFICIENT (ALL,I,NAcommod)   (ALL,X,Afr_LAm_Weu)      NAcapital(I,X)
!CAPITAL RETURNS FOR COUNTRIES IN GROUP C!;

!DEFINE COEFFICIENTS TO BE READ FROM UNIDO!

COEFFICIENT (ALL,I,UNmanuf) (ALL,X, AfAsLatCpeEu)       UNlabour(I,X)
!LABOUR FOR COUNTRIES IN GROUP E!;

COEFFICIENT (ALL,I,UNmanuf) (ALL,X, AfAsLatCpeEu)       UNcapital(I,X)
!CAPITAL FOR COUNTRIES IN GROUP E!;

!*********************************************************************
                READING DATA BASE
 *********************************************************************!
READ


!READ FROM NATIONAL ACCOUNTS!



   (ALL,I,NAcommod) (ALL,X,Afr_LAm_Weu)
   NAlabour(I,X)
```

```
            FROM FILE UNIO HEADER "UN10";

   (ALL,I,NAcommod) (ALL,X,Afr_LAm_Weu)
   NAcapital(I,X)
            FROM FILE UNIO HEADER "UN11";

!READ FROM UNIDO!


(ALL,I,UNmanuf) (ALL,X,AfAsLatCpeEu)
      UNlabour(I,X)
               FROM FILE UNIO HEADER "UN12";

(ALL,I,UNmanuf) (ALL,X,AfAsLatCpeEu)
      UNcapital(I,X)
               FROM FILE UNIO HEADER "UN13";



!labour!
COEFFICIENT (ALL,I,IND) (ALL,X,ROFW)          ROFWL(I,X);

FORMULA (ALL,I,agr) (ALL,X,Afr_LAm_Weu)
ROFWL(I,X)=(NAlabour("NAagr",X)/NAvalueAdded("NAagr",X))*VA(I,X);
FORMULA (ALL,I,agr) (ALL,X,MidE_Sasia)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X) ;
FORMULA (ALL,I,agr) (ALL,X,CentPlan)
ROFWL(I,X)=LAB(I,"KOREA")/GDPFCS(I,"KOREA")*VA(I,X) ;

FORMULA (ALL,I,mining) (ALL,X,Afr_LAm_Weu)
ROFWL(I,X)=(NAlabour("NAmining",X)/NAvalueAdded("NAmining",X))*VA(I,X);
FORMULA (ALL,I,mining) (ALL,X,MidE_Sasia)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X) ;
FORMULA (ALL,I,mining) (ALL,X,CentPlan)
ROFWL(I,X)=LAB(I,"KOREA")/GDPFCS(I,"KOREA")*VA(I,X) ;

FORMULA (ALL,I,foodProd)  (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNlabour("UNfood",X)/UNvalueAdded("UNfood",X)*VA(I,X);
FORMULA (ALL,I,foodProd)  (ALL,X,MidE)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X) ;

FORMULA (ALL,I,Bev_Tob) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNlabour("UNbev",X)/UNvalueAdded("UNbev",X)*VA(I,X);
FORMULA  (ALL,I,Bev_Tob) (ALL,X,MidE)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Textiles) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNlabour("UNtextiles",X)/UNvalueAdded("UNtextiles",X)*VA(I,X);
FORMULA  (ALL,I,Textiles) (ALL,X,MidE)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);
```

```
FORMULA  (ALL,I,Apparel) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNlabour("UNapparel",X)/UNvalueAdded("UNapparel",X)*VA(I,X);
FORMULA  (ALL,I,Apparel) (ALL,X,MidE)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Leather) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNlabour("UNleather",X)/UNvalueAdded("UNleather",X)*VA(I,X);
FORMULA  (ALL,I,leather) (ALL,X,MidE)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Wood) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNlabour("UNwood",X)/UNvalueAdded("UNwood",X)*VA(I,X);
FORMULA  (ALL,I,Wood) (ALL,X,MidE)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Paper) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNcapital("UNpaper",X)/UNvalueAdded("UNpaper",X)*VA(I,X);
FORMULA  (ALL,I,Paper) (ALL,X,MidE)
ROFWL(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Chem) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNcapital("UNchem",X)/UNvalueAdded("UNchem",X)*VA(I,X);
FORMULA  (ALL,I,Chem) (ALL,X,MidE)
ROFWL(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Petrol) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNcapital("UNpetrol",X)/UNvalueAdded("UNpetrol",X)*VA(I,X);
FORMULA  (ALL,I,Petrol)(ALL,X,MidE)
ROFWL(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,NonMetal) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNlabour("UNnonMet",X)/UNvalueAdded("UNnonMet",X)*VA(I,X);
FORMULA  (ALL,I,NonMetal) (ALL,X,MidE)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Iron) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNlabour("UNiron",X)/UNvalueAdded("UNiron",X)*VA(I,X);
FORMULA  (ALL,I,Iron) (ALL,X,MidE)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,NonFerr) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNlabour("UNnonFerr",X)/UNvalueAdded("UNnonFerr",X)*VA(I,X);
FORMULA  (ALL,I,NonFerr) (ALL,X,MidE)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,FabMet) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNlabour("UNfabMet",X)/UNvalueAdded("UNfabMet",X)*VA(I,X);
FORMULA  (ALL,I,FabMet) (ALL,X,MidE)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Trans) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNlabour("UNtrans",X)/UNvalueAdded("UNtrans",X)*VA(I,X);
```

```
FORMULA  (ALL,I,Trans) (ALL,X,MidE)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Mach) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNlabour("UNothMach",X)/UNvalueAdded("UNothMach",X)*VA(I,X);
FORMULA  (ALL,I,Mach) (ALL,X,MidE)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,OthManuf) (ALL,X,AfAsLatCpeEu)
ROFWL(I,X)=UNlabour("UNothManuf",X)/UNvalueAdded("UNothManuf",X)*VA(I,X);
FORMULA  (ALL,I,OthManuf) (ALL,X,MidE)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,EWG) (ALL,X,Afr_LAm_Weu)
ROFWL(I,X)=NAlabour("NAewg",X)/NAvalueAdded("NAewg",X)*VA(I,X);
FORMULA (ALL,I,EWG) (ALL,X,MidE_Sasia)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);
FORMULA (ALL,I,EWG) (ALL,X,CentPlan)
ROFWL(I,X)=LAB(I,"KOREA")/GDPFCS(I,"KOREA")*VA(I,X);

FORMULA (ALL,I,Constr) (ALL,X,Afr_LAm_Weu)
ROFWL(I,X)=NAlabour("NAconstr",X)/NAvalueAdded("NAconstr",X)*VA(I,X);
FORMULA (ALL,I,Constr) (ALL,X,MidE_Sasia)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);
FORMULA (ALL,I,Constr) (ALL,X,CentPlan)
ROFWL(I,X)=LAB(I,"KOREA")/GDPFCS(I,"KOREA")*VA(I,X);

FORMULA (ALL,I,OSgovPriv) (ALL,X,Afr_LAm_Weu)
ROFWL(I,X)=NAlabour("NAsales",X)/NAvalueAdded("NAsales",X)*VA(I,X);
FORMULA (ALL,I,OSdwelling)
ROFWL(I,"Africa")=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,"Africa");
FORMULA (ALL,I,OSdwelling)
ROFWL(I,"LatAmer")=LAB(I,"Korea")/GDPFCS(I,"Korea")*VA(I,"LatAmer");
FORMULA (ALL,I,OSdwelling)
ROFWL(I,"Weuro")=LAB(I,"EC")/GDPFCS(I,"EC")*VA(I,"Weuro");
FORMULA (ALL,I,otherServs) (ALL,X,MidE_Sasia)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);
FORMULA (ALL,I,otherServs) (ALL,X,CentPlan)
ROFWL(I,X)=LAB(I,"KOREA")/GDPFCS(I,"KOREA")*VA(I,X);

FORMULA  (ALL,I,Trade) (ALL,X,Afr_LAm_Weu)
ROFWL(I,X)=NAlabour("NAtrade",X)/NAvalueAdded("NAtrade",X)*VA(I,X);
FORMULA  (ALL,I,Trade) (ALL,X,MidE_Sasia)
ROFWL(I,X)=LAB(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);
FORMULA  (ALL,I,Trade) (ALL,X,CentPlan)
ROFWL(I,X)=LAB(I,"KOREA")/GDPFCS(I,"KOREA")*VA(I,X);



COEFFICIENT (ALL,I,IND)          TROFWL(I)
```

```
!LABOUR RETURNS FOR THE REST OF THE WORLD BY INDUSTRY!;
FORMULA (ALL,I,IND)
TROFWL(I)=SUM(X,ROFW,ROFWL(I,X));


!*****************************************************************!

!COEFFICIENT FOR CAPITAL!
COEFFICIENT (ALL,I,IND) (ALL,X,ROFW)          ROFWK(I,X);

FORMULA (ALL,I,agr) (ALL,X,Afr_LAm_Weu)
ROFWK(I,X)=(NAcapital("NAagr",X)/NAvalueAdded("NAagr",X))*VA(I,X);
FORMULA (ALL,I,agr) (ALL,X,MidE_Sasia)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X) ;
FORMULA (ALL,I,agr) (ALL,X,CentPlan)
ROFWK(I,X)=CAP(I,"KOREA")/GDPFCS(I,"KOREA")*VA(I,X) ;

FORMULA (ALL,I,mining) (ALL,X,Afr_LAm_Weu)
ROFWK(I,X)=(NAcapital("NAmining",X)/NAvalueAdded("NAmining",X))*VA(I,X);
FORMULA (ALL,I,mining) (ALL,X,MidE_Sasia)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X) ;

FORMULA (ALL,I,foodProd)  (ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNfood",X)/UNvalueAdded("UNfood",X)*VA(I,X);
FORMULA (ALL,I,foodProd)  (ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X) ;

FORMULA   (ALL,I,Bev_Tob)(ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNtextiles",X)/UNvalueAdded("UNtextiles",X)*VA(I,X);
FORMULA  (ALL,I,Textiles)(ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA   (ALL,I,Apparel)(ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNapparel",X)/UNvalueAdded("UNapparel",X)*VA(I,X);
FORMULA  (ALL,I,Apparel)(ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA   (ALL,I,Leather)(ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNleather",X)/UNvalueAdded("UNleather",X)*VA(I,X);
FORMULA  (ALL,I,Leather)(ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA   (ALL,I,Wood)(ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNwood",X)/UNvalueAdded("UNwood",X)*VA(I,X);
FORMULA  (ALL,I,Wood)(ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA   (ALL,I,Paper)(ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNpaper",X)/UNvalueAdded("UNpaper",X)*VA(I,X);
FORMULA  (ALL,I,Paper)(ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);
```

```
FORMULA  (ALL,I,Chem)(ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNchem",X)/UNvalueAdded("UNchem",X)*VA(I,X);
FORMULA  (ALL,I,Chem)(ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Petrol)(ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNpetrol",X)/UNvalueAdded("UNpetrol",X)*VA(I,X);
FORMULA  (ALL,I,Petrol)(ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,NonMetal)(ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNnonMet",X)/UNvalueAdded("UNnonMet",X)*VA(I,X);
FORMULA  (ALL,I,NonMetal)(ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Iron)(ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNiron",X)/UNvalueAdded("UNiron",X)*VA(I,X);
FORMULA  (ALL,I,Iron)(ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,NonFerr)(ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNnonFerr",X)/UNvalueAdded("UNnonFerr",X)*VA(I,X);
FORMULA  (ALL,I,NonFerr)(ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,FabMet)(ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNfabMet",X)/UNvalueAdded("UNfabMet",X)*VA(I,X);
FORMULA  (ALL,I,FabMet)(ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Trans)(ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNtrans",X)/UNvalueAdded("UNtrans",X)*VA(I,X);
FORMULA  (ALL,I,Trans)(ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,Mach)(ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNothMach",X)/UNvalueAdded("UNothMach",X)*VA(I,X);
FORMULA  (ALL,I,Mach)(ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA  (ALL,I,OthManuf)(ALL,X,AfAsLatCpeEu)
ROFWK(I,X)=UNcapital("UNothManuf",X)/UNvalueAdded("UNothManuf",X)*VA(I,X);
FORMULA  (ALL,I,OthManuf)(ALL,X,MidE)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);

FORMULA (ALL,I,EWG)(ALL,X,Afr_LAm_Weu)
ROFWK(I,X)=NAcapital("NAewg",X)/NAvalueAdded("NAewg",X)*VA(I,X);
FORMULA (ALL,I,EWG)(ALL,X,MidE_Sasia)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);
FORMULA (ALL,I,EWG)(ALL,X,CentPlan)
ROFWK(I,X)=CAP(I,"KOREA")/GDPFCS(I,"KOREA")*VA(I,X);
```

```
FORMULA (ALL,I,Constr)(ALL,X,Afr_LAm_Weu)
ROFWK(I,X)=NAcapital("NAconstr",X)/NAvalueAdded("NAconstr",X)*VA(I,X);
FORMULA (ALL,I,Constr)(ALL,X,MidE_Sasia)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);
FORMULA (ALL,I,Constr)(ALL,X,CentPlan)
ROFWK(I,X)=CAP(I,"KOREA")/GDPFCS(I,"KOREA")*VA(I,X);

FORMULA (ALL,I,OSgovPriv) (ALL,X,Afr_LAm_Weu)
ROFWK(I,X)=NAcapital("NAsales",X)/NAvalueAdded("NAsales",X)*VA(I,X);
FORMULA (ALL,I,OSdwelling)
ROFWK(I,"Africa")=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,"Africa");
FORMULA (ALL,I,OSdwelling)
ROFWK(I,"LatAmer")=CAP(I,"Korea")/GDPFCS(I,"Korea")*VA(I,"LatAmer");
FORMULA (ALL,I,OSdwelling)
ROFWK(I,"Weuro")=CAP(I,"EC")/GDPFCS(I,"EC")*VA(I,"Weuro");
FORMULA (ALL,I,otherServs) (ALL,X,MidE_Sasia)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);
FORMULA (ALL,I,otherServs) (ALL,X,CentPlan)
ROFWK(I,X)=CAP(I,"KOREA")/GDPFCS(I,"KOREA")*VA(I,X);

FORMULA  (ALL,I,Trade)(ALL,X,Afr_LAm_Weu)
ROFWK(I,X)=NAcapital("NAtrade",X)/NAvalueAdded("NAtrade",X)*VA(I,X);
FORMULA (ALL,I,Trade)(ALL,X,MidE_Sasia)
ROFWK(I,X)=CAP(I,"Thailand")/GDPFCS(I,"Thailand")*VA(I,X);
FORMULA (ALL,I,Trade)(ALL,X,CentPlan)
ROFWK(I,X)=CAP(I,"KOREA")/GDPFCS(I,"KOREA")*VA(I,X);


COEFFICIENT (ALL,I,IND)          TROFWK(I)
!CAPITAL RETURNS FOR THE REST OF THE WORLD BY INDUSTRY!;
FORMULA (ALL,I,IND)
TROFWK(I)=SUM(X,ROFW,ROFWK(I,X));


!----------------------------------------------------------------
              WRITE STATEMENTS

 ----------------------------------------------------------------!
WRITE TROFWL TO FILE ROWP92 HEADER "LAB";
WRITE TROFWK TO FILE ROWP92 HEADER "CAP";
WRITE TRLOFWO TO FILE ROWP92 HEADER "OUT";
```

COMBROW.STI

```
f          [full prompts
b          [error recovery
y          [based on old HA file?
va.out
temp.out
aw          [add and write
h          [from header file
cig.out     [header file name
con         [header name
w          [write to new file
n          [use as basis for next change?
aw          [add and write
h          [from header file
cig.out     [header file name
inv         [header name
w          [write to new file
n          [use as basis for next change?
aw          [add and write
h          [from header file
cig.out     [header file name
govt        [header name
w          [write to new file
n          [use as basis for next change?
aw          [add and write
h          [from header file
rw92.dat    [header file name
UN4         [header name
w          [write to new file
n          [use as basis for next change?
ex         [exit
a          [write all remaining headers to the new file
name
date
**end
y
```

MAKE.STI

```
disp
combrow.out
newtrade.dat
as.upd
nz.upd
jp.upd
ec.upd
cn.upd
us.upd
kr.upd
in.upd
ms.upd
ph.upd
sn.upd
th.upd
ch.upd
hk.upd
tw.upd
make.out
```

MAKEROW.TAB

```
!                                                   !

!                THE SALTER MODEL                   !
!           ----------------------------------      !
!file to seperate C,I,G for ROW into imports and exports; to generate    !
!import duty and to report primary factors for ROW                 !


!***********************************************************************!
!                      FILES                        !
!***********************************************************************!

file row_data #Initial rest of world io data#;
file trade #Trade data#;

FILE AUSIO #INCLUDES AUST DATA#;
FILE NZIO #INCLUDES NZ DATA#;
FILE JPNIO #INCLUDES JAPANESE DATA#;
FILE ECIO #INCLUDES EC DATA#;
FILE CANIO #INCLUDES CANADA DATA#;
FILE USAIO #INCLUDES USA DATA#;
FILE KORIO #INCLUDES KOREA DATA#;
FILE INDIO #INCLUDES Indonesia DATA#;
FILE MALIO;
FILE PHIIO;
FILE SINIO;
FILE THAIO;
```

```
FILE CHIIO;
FILE HKIO;
FILE TAIIO;

file (new) row_ras #Rest of world io data for RASing#;

!*************************************************************************!
!                    SETS                          !
!*************************************************************************!
SET IND(IND1-IND37);

SET FD (I,C,G);
SET INP(IIP);
SET SCR(D,IM);

SET PRF(LABO,K,LND);
SET AGR(IND1 - IND6);
SUBSET AGR IS SUBSET OF IND;

SET NAGR(IND7 - IND37);
SUBSET NAGR IS SUBSET OF IND;

SET SEC (AUSTRALIA,
NZ,
CANADA,
USA,
JAPAN,
KOREA,
EC,
Indonesia,
Malaysia,
Philippines,
Singapore,
Thailand,
China,
HK,
Taiwan,
ROFW) ;

SET REG (AUSTRALIA,
NZ,
CANADA,
USA,
JAPAN,
KOREA,
EC,
Indonesia,
Malaysia,
Philippines,
Singapore,
Thailand,
China,
```

```
HK,
Taiwan);
SUBSET REG IS SUBSET OF SEC;

SET REGN(EC,
KOREA,
Thailand);
SUBSET REGN IS SUBSET OF SEC;
SUBSET REGN IS SUBSET OF REG;

SET ROW(Africa,
MidE,
Sasia,
LatAm,
CPE,
Weurope);

!*************************************************************************!
!                    COEFFICIENTS                    !
!*************************************************************************!


COEFFICIENT (all,i,ind)(all,z,reg)          ODT(i,z);
!import duty!

(ALL,F,FD)(ALL,SC,SCR)(ALL,I,IND)(ALL,Z,REG)      FINDEM(F,SC,I,Z)
!final demands to be read from single country databases!;

(ALL,IIP,INP)(ALL,SC,SCR)(ALL,I,IND)(ALL,J,IND)(ALL,Z,REG) FINDO(IIP,SC,I,J,Z)
!intermediate inputs to be read from single country databases!      ;

(ALL,F,PRF) (ALL,I,IND) (ALL,Z,REG)              PR(F,I,Z)
!primary factors labour, capital and land!;

(ALL,I,IND) (ALL,Z,REG)                    SAVINGS(I,Z);


!coefficients to be read from the existing ROW data file!
COEFFICIENT (ALL,I,IND)              R_OUT(i);
        (ALL,I,IND)              r_lab(i);
        (ALL,I,IND)              CAP(i);
        (ALL,I,IND)              R_INV(i);
        (ALL,I,IND)              R_CON(i);
        (ALL,I,IND)              R_GOVT(i);

        (ALL,S,SEC)(ALL,D,SEC)(ALL,I,IND)   EXPS(s,d,i);
        (ALL,S,SEC)(ALL,D,SEC)(ALL,I,IND)   IMPS(s,d,i);

COEFFICIENT (ALL,X,ROW)                ROWGDP(X);
!GDP for the ROW countries!
```

```
!*************************************************************************
                    READING DATABASES
 *************************************************************************!

read R_OUT from file row_data header "out";
read r_lab from file row_data header "lab";
read CAP from file row_data header "cap";
read R_INV from file row_data header "inv";
read R_CON from file row_data header "con";
read R_GOVT from file row_data header "govt";
read ROWGDP from file row_data header "gdp";

!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%!
!note that the trade database is in thousands!
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%!
read EXPS from file trade header "exp";
read IMPS  from file trade header "imp";


coefficient (all,s,sec)  freight(s);
read freight from file trade header "emrg";
coefficient row_freight;
formula
row_freight = freight("rofw")/1000;

FORMULA     (ALL,S,SEC)(ALL,D,SEC)(ALL,I,IND)
EXPS(S,D,I) = EXPS(S,D,I)/1000;

FORMULA     (ALL,S,SEC)(ALL,D,SEC)(ALL,I,IND)
IMPS(S,D,I) = IMPS(S,D,I)/1000;


!-------------------------------------------------------------------
                    AUSTRALIAN DATA
 -------------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","D",I,J,"AUSTRALIA")
              FROM FILE  AUSIO  HEADER  "AI01" ;

   (ALL,I,IND)(ALL,J,IND)
   FINDO("IIP","IM",I,J,"AUSTRALIA")
              FROM FILE  AUSIO  HEADER  "AI02" ;

   (ALL,I,IND)
    FINDEM("I","IM",I,"AUSTRALIA")
              FROM FILE  AUSIO  HEADER  "AI04" ;

     (ALL,I,IND)
    FINDEM("C","IM",I,"AUSTRALIA")
              FROM FILE  AUSIO  HEADER  "AI06" ;
```

```
   (ALL,I,IND)
   FINDEM("G","IM",I,"AUSTRALIA")
            FROM FILE  AUSIO  HEADER  "AI08" ;

    (ALL,I,IND)
    ODT(I,"AUSTRALIA")
           FROM FILE  AUSIO  HEADER  "AI27" ;

    (ALL,I,IND)
    PR("K",I,"AUSTRALIA")
            FROM FILE  AUSIO  HEADER  "AI14" ;

   (ALL,I,IND)
    PR("LND",I,"AUSTRALIA")
            FROM FILE  AUSIO  HEADER  "AI15" ;

   (ALL,I,IND)
    PR("LABO",I,"AUSTRALIA")
            FROM FILE  AUSIO  HEADER  "AI13" ;


!-----------------------------------------------------------------
                NEW ZEALAND DATA
-----------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
   FINDO("IIP","D",I,J,"NZ")
            FROM FILE  NZIO  HEADER  "AI01" ;

   (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","IM",I,J,"NZ")
            FROM FILE  NZIO  HEADER  "AI02" ;

   (ALL,I,IND)
    FINDEM("I","IM",I,"NZ")
            FROM FILE  NZIO  HEADER  "AI04" ;


   (ALL,I,IND)
    FINDEM("C","IM",I,"NZ")
            FROM FILE  NZIO  HEADER  "AI06" ;

   (ALL,I,IND)
   FINDEM("G","im",I,"NZ")
            FROM FILE  NZIO  HEADER  "AI08" ;



   (ALL,I,IND)
    ODT(I,"NZ")
           FROM FILE  NZIO  HEADER  "AI27" ;
```

```
  (ALL,I,IND)
 PR("K",I,"NZ")
          FROM FILE  NZIO  HEADER  "AI14" ;

  (ALL,I,IND)
 PR("LND",I,"NZ")
          FROM FILE  NZIO  HEADER  "AI15" ;

  (ALL,I,IND)
  PR("LABO",I,"NZ")
          FROM FILE  NZIO  HEADER  "AI13" ;


!-----------------------------------------------------------------
                JAPAN DATA
 -----------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
   FINDO("IIP","D",I,J,"JAPAN")
          FROM FILE  JPNIO  HEADER  "AI01" ;

  (ALL,I,IND)(ALL,J,IND)
   FINDO("IIP","IM",I,J,"JAPAN")
          fROM FILE  JPNIO  HEADER  "AI02" ;

  (ALL,I,IND)
   FINDEM("I","IM",I,"JAPAN")
          FROM FILE  JPNIO  HEADER  "AI04" ;


  (ALL,I,IND)
   FINDEM("C","IM",I,"JAPAN")
          FROM FILE  JPNIO  HEADER  "AI06" ;

  (ALL,I,IND)
  FINDEM("G","IM",I,"JAPAN")
          FROM FILE  JPNIO  HEADER  "AI08" ;

   (ALL,I,IND)
   ODT(I,"JAPAN")
         FROM FILE  JPNIO  HEADER  "AI27" ;

   (ALL,I,IND)
   PR("K",I,"JAPAN")
          FROM FILE  JPNIO  HEADER  "AI14" ;

  (ALL,I,IND)
  PR("LND",I,"JAPAN")
          FROM FILE  JPNIO  HEADER  "AI15" ;

   (ALL,I,IND)
   PR("LABO",I,"JAPAN")
          FROM FILE  JPNIO  HEADER  "AI13" ;
```

```
!-----------------------------------------------------------------
                    EC DATA
 ----------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","D",I,J,"EC")
            FROM FILE ECIO HEADER "AI01" ;

    (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","IM",I,J,"EC")
            FROM FILE ECIO HEADER "AI02" ;

    (ALL,I,IND)
    FINDEM("I","D",I,"EC")
            FROM FILE ECIO HEADER "AI03" ;

    (ALL,I,IND)
    FINDEM("I","IM",I,"EC")
            FROM FILE ECIO HEADER "AI04" ;

    (ALL,I,IND)
    FINDEM("C","D",I,"EC")
            FROM FILE ECIO HEADER "AI05" ;

    (ALL,I,IND)
    FINDEM("C","IM",I,"EC")
            FROM FILE ECIO HEADER "AI06" ;

    (ALL,I,IND)
    FINDEM("G","D",I,"EC")
            FROM FILE ECIO HEADER "AI07" ;

    (ALL,I,IND)
    FINDEM("G","IM",I,"EC")
            FROM FILE ECIO HEADER "AI08" ;


    (ALL,I,IND)
    ODT(I,"EC")
            FROM FILE ECIO HEADER "AI27" ;


    (ALL,I,IND)
    PR("K",I,"EC")
            FROM FILE ECIO HEADER "AI14" ;

    (ALL,I,IND)
    PR("LND",I,"EC")
            FROM FILE ECIO HEADER "AI15" ;

    (ALL,I,IND)
```

```
    PR("LABO",I,"EC")
            FROM FILE ECIO HEADER "AI13" ;


!-----------------------------------------------------------------
                CANADIAN DATA
 ----------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","D",I,J,"CANADA")
            FROM FILE CANIO HEADER "AI01" ;

   (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","IM",I,J,"CANADA")
            FROM FILE CANIO HEADER "AI02" ;

   (ALL,I,IND)
    FINDEM("I","IM",I,"CANADA")
            FROM FILE CANIO HEADER "AI04" ;

   (ALL,I,IND)
    FINDEM("C","IM",I,"CANADA")
            FROM FILE CANIO HEADER "AI06" ;

   (ALL,I,IND)
   FINDEM("G","IM",I,"CANADA")
            FROM FILE CANIO HEADER "AI08" ;



   (ALL,I,IND)
    ODT(I,"CANADA")
            FROM FILE CANIO HEADER "AI27" ;


    (ALL,I,IND)
    PR("K",I,"CANADA")
            FROM FILE CANIO HEADER "AI14" ;

   (ALL,I,IND)
    PR("LND",I,"CANADA")
            FROM FILE CANIO HEADER "AI15" ;

   (ALL,I,IND)
    PR("LABO",I,"CANADA")
            FROM FILE CANIO HEADER "AI13" ;


!-----------------------------------------------------------------
                US DATA
 ----------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","D",I,J,"USA")
```

```
          FROM FILE  USAIO  HEADER  "AI01" ;

   (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","IM",I,J,"USA")
          FROM FILE  USAIO  HEADER  "AI02" ;

   (ALL,I,IND)
    FINDEM("I","IM",I,"USA")
          FROM FILE  USAIO  HEADER  "AI04" ;


   (ALL,I,IND)
    FINDEM("C","IM",I,"USA")
          FROM FILE  USAIO  HEADER  "AI06" ;

   (ALL,I,IND)
   FINDEM("G","IM",I,"USA")
          FROM FILE  USAIO  HEADER  "AI08" ;



   (ALL,I,IND)
    ODT(I,"USA")
          FROM FILE  USAIO  HEADER  "AI27" ;



    (ALL,I,IND)
    PR("K",I,"USA")
          FROM FILE  USAIO  HEADER  "AI14" ;

   (ALL,I,IND)
    PR("LND",I,"USA")
          FROM FILE  USAIO  HEADER  "AI15" ;

   (ALL,I,IND)
    PR("LABO",I,"USA")
          FROM FILE  USAIO  HEADER  "AI13" ;



!-----------------------------------------------------------------
                KOREAN DATA
 -----------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","D",I,J,"KOREA")
          FROM FILE  KORIO  HEADER  "AI01" ;

   (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","IM",I,J,"KOREA")
          FROM FILE  KORIO  HEADER  "AI02" ;
```

```
   (ALL,I,IND)
    FINDEM("I","D",I,"KOREA")
            FROM FILE KORIO HEADER "AI03" ;

   (ALL,I,IND)
    FINDEM("I","IM",I,"KOREA")
            FROM FILE KORIO HEADER "AI04" ;

   (ALL,I,IND)
    FINDEM("C","D",I,"KOREA")
            FROM FILE KORIO HEADER "AI05" ;

   (ALL,I,IND)
    FINDEM("C","IM",I,"KOREA")
            FROM FILE KORIO HEADER "AI06" ;

   (ALL,I,IND)
    FINDEM("G","D",I,"KOREA")
            FROM FILE KORIO HEADER "AI07" ;

   (ALL,I,IND)
   FINDEM("G","IM",I,"KOREA")
            FROM FILE KORIO HEADER "AI08" ;


   (ALL,I,IND)
    ODT(I,"KOREA")
            FROM FILE KORIO HEADER "AI27" ;


    (ALL,I,IND)
    PR("K",I,"KOREA")
            FROM FILE KORIO HEADER "AI14" ;

   (ALL,I,IND)
    PR("LND",I,"KOREA")
            FROM FILE KORIO HEADER "AI15" ;

   (ALL,I,IND)
    PR("LABO",I,"KOREA")
            FROM FILE KORIO HEADER "AI13" ;


!-----------------------------------------------------------------
                  Thailand DATA
 -----------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","D",I,J,"Thailand")
            FROM FILE THAIO HEADER "AI01" ;

   (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","IM",I,J,"Thailand")
```

```
          FROM FILE  THAIO  HEADER  "AI02" ;

   (ALL,I,IND)
    FINDEM("I","D",I,"Thailand")
          FROM FILE  THAIO  HEADER  "AI03" ;

   (ALL,I,IND)
    FINDEM("I","IM",I,"Thailand")
          FROM FILE  THAIO  HEADER  "AI04" ;

   (ALL,I,IND)
    FINDEM("C","D",I,"Thailand")
          FROM FILE  THAIO  HEADER  "AI05" ;

   (ALL,I,IND)
    FINDEM("C","IM",I,"Thailand")
          FROM FILE  THAIO  HEADER  "AI06" ;

   (ALL,I,IND)
    FINDEM("G","D",I,"Thailand")
          FROM FILE  THAIO  HEADER  "AI07" ;

   (ALL,I,IND)
   FINDEM("G","IM",I,"Thailand")
          FROM FILE  THAIO  HEADER  "AI08" ;


   (ALL,I,IND)
    ODT(I,"Thailand")
          FROM FILE  THAIO  HEADER  "AI27" ;


   (ALL,I,IND)
    PR("K",I,"Thailand")
          FROM FILE  THAIO  HEADER  "AI14" ;

   (ALL,I,IND)
    PR("LND",I,"Thailand")
          FROM FILE  THAIO  HEADER  "AI15" ;

   (ALL,I,IND)
    PR("LABO",I,"Thailand")
          FROM FILE  THAIO  HEADER  "AI13" ;


!-----------------------------------------------------------------
                IndonesiaN DATA
 -----------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","D",I,J,"Indonesia")
          FROM FILE  INDIO  HEADER  "AI01" ;
```

```
(ALL,I,IND)(ALL,J,IND)
 FINDO("IIP","IM",I,J,"Indonesia")
          FROM FILE INDIO HEADER "AI02" ;

(ALL,I,IND)
 FINDEM("I","D",I,"Indonesia")
          FROM FILE INDIO HEADER "AI03" ;

(ALL,I,IND)
 FINDEM("I","IM",I,"Indonesia")
          FROM FILE INDIO HEADER "AI04" ;

(ALL,I,IND)
 FINDEM("C","D",I,"Indonesia")
          FROM FILE INDIO HEADER "AI05" ;

(ALL,I,IND)
 FINDEM("C","IM",I,"Indonesia")
          FROM FILE INDIO HEADER "AI06" ;

(ALL,I,IND)
 FINDEM("G","D",I,"Indonesia")
          FROM FILE INDIO HEADER "AI07" ;

(ALL,I,IND)
FINDEM("G","IM",I,"Indonesia")
          FROM FILE INDIO HEADER "AI08" ;


(ALL,I,IND)
 ODT(I,"Indonesia")
          FROM FILE INDIO HEADER "AI27" ;


 (ALL,I,IND)
 PR("K",I,"Indonesia")
          FROM FILE INDIO HEADER "AI14" ;

(ALL,I,IND)
 PR("LND",I,"Indonesia")
          FROM FILE INDIO HEADER "AI15" ;

(ALL,I,IND)
 PR("LABO",I,"Indonesia")
          FROM FILE IndIO HEADER "AI13" ;

!-------------------------------------------------------------
                MalaysiaN DATA
-------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
```

```
 FINDO("IIP","D",I,J,"Malaysia")
         FROM FILE MALIO HEADER "AI01" ;

(ALL,I,IND)(ALL,J,IND)
 FINDO("IIP","IM",I,J,"Malaysia")
         FROM FILE MALIO HEADER "AI02" ;

(ALL,I,IND)
 FINDEM("I","D",I,"Malaysia")
         FROM FILE MALIO HEADER "AI03" ;

(ALL,I,IND)
 FINDEM("I","IM",I,"Malaysia")
         FROM FILE MALIO HEADER "AI04" ;

(ALL,I,IND)
 FINDEM("C","D",I,"Malaysia")
         FROM FILE MALIO HEADER "AI05" ;

(ALL,I,IND)
 FINDEM("C","IM",I,"Malaysia")
         FROM FILE MALIO HEADER "AI06" ;

(ALL,I,IND)
 FINDEM("G","D",I,"Malaysia")
         FROM FILE MALIO HEADER "AI07" ;

(ALL,I,IND)
FINDEM("G","IM",I,"Malaysia")
         FROM FILE MALIO HEADER "AI08" ;


(ALL,I,IND)
 ODT(I,"Malaysia")
         FROM FILE MALIO HEADER "AI27" ;


 (ALL,I,IND)
 PR("K",I,"Malaysia")
         FROM FILE MALIO HEADER "AI14" ;

(ALL,I,IND)
 PR("LND",I,"Malaysia")
         FROM FILE MALIO HEADER "AI15" ;

(ALL,I,IND)
 PR("LABO",I,"Malaysia")
         FROM FILE MALIO HEADER "AI13" ;


!-----------------------------------------------------------------
```

PhilippinesN DATA

```
 ----------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","D",I,J,"Philippines")
            FROM FILE PHIIO HEADER "AI01" ;

   (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","IM",I,J,"Philippines")
            FROM FILE PHIIO HEADER "AI02" ;

   (ALL,I,IND)
    FINDEM("I","D",I,"Philippines")
            FROM FILE PHIIO HEADER "AI03" ;

   (ALL,I,IND)
    FINDEM("I","IM",I,"Philippines")
            FROM FILE PHIIO HEADER "AI04" ;

   (ALL,I,IND)
    FINDEM("C","D",I,"Philippines")
            FROM FILE PHIIO HEADER "AI05" ;

   (ALL,I,IND)
    FINDEM("C","IM",I,"Philippines")
            FROM FILE PHIIO HEADER "AI06" ;

   (ALL,I,IND)
    FINDEM("G","D",I,"Philippines")
            FROM FILE PHIIO HEADER "AI07" ;

   (ALL,I,IND)
   FINDEM("G","IM",I,"Philippines")
            FROM FILE PHIIO HEADER "AI08" ;


   (ALL,I,IND)
    ODT(I,"Philippines")
           FROM FILE PHIIO HEADER "AI27" ;


    (ALL,I,IND)
    PR("K",I,"Philippines")
            FROM FILE PHIIO HEADER "AI14" ;

   (ALL,I,IND)
    PR("LND",I,"Philippines")
            FROM FILE PHIIO HEADER "AI15" ;

   (ALL,I,IND)
    PR("LABO",I,"Philippines")
            FROM FILE PHIIO HEADER "AI13" ;
```

```
!-----------------------------------------------------------------
                SingaporeN DATA
 ----------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","D",I,J,"Singapore")
            FROM FILE  SINIO  HEADER  "AI01" ;

   (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","IM",I,J,"Singapore")
            FROM FILE  SINIO  HEADER  "AI02" ;

   (ALL,I,IND)
    FINDEM("I","D",I,"Singapore")
            FROM FILE  SINIO  HEADER  "AI03" ;

   (ALL,I,IND)
    FINDEM("I","IM",I,"Singapore")
            FROM FILE  SINIO  HEADER  "AI04" ;

   (ALL,I,IND)
    FINDEM("C","D",I,"Singapore")
            FROM FILE  SINIO  HEADER  "AI05" ;

   (ALL,I,IND)
    FINDEM("C","IM",I,"Singapore")
            FROM FILE  SINIO  HEADER  "AI06" ;

   (ALL,I,IND)
    FINDEM("G","D",I,"Singapore")
            FROM FILE  SINIO  HEADER  "AI07" ;

   (ALL,I,IND)
   FINDEM("G","IM",I,"Singapore")
            FROM FILE  SINIO  HEADER  "AI08" ;


   (ALL,I,IND)
    ODT(I,"Singapore")
           FROM FILE  SINIO  HEADER  "AI27" ;


    (ALL,I,IND)
    PR("K",I,"Singapore")
            FROM FILE  SINIO  HEADER  "AI14" ;

   (ALL,I,IND)
    PR("LND",I,"Singapore")
            FROM FILE  SINIO  HEADER  "AI15" ;

   (ALL,I,IND)
    PR("LABO",I,"Singapore")
            FROM FILE  SINIO  HEADER  "AI13" ;
```

```
!-----------------------------------------------------------------
                ChinaN DATA
 ----------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","D",I,J,"China")
            FROM FILE  CHIIO  HEADER  "AI01" ;

   (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","IM",I,J,"China")
            FROM FILE  CHIIO  HEADER  "AI02" ;

   (ALL,I,IND)
    FINDEM("I","D",I,"China")
            FROM FILE  CHIIO  HEADER  "AI03" ;

   (ALL,I,IND)
    FINDEM("I","IM",I,"China")
            FROM FILE  CHIIO  HEADER  "AI04" ;

   (ALL,I,IND)
    FINDEM("C","D",I,"China")
            FROM FILE  CHIIO  HEADER  "AI05" ;

   (ALL,I,IND)
    FINDEM("C","IM",I,"China")
            FROM FILE  CHIIO  HEADER  "AI06" ;

   (ALL,I,IND)
    FINDEM("G","D",I,"China")
            FROM FILE  CHIIO  HEADER  "AI07" ;

   (ALL,I,IND)
   FINDEM("G","IM",I,"China")
            FROM FILE  CHIIO  HEADER  "AI08" ;


   (ALL,I,IND)
    ODT(I,"China")
           FROM FILE  CHIIO  HEADER  "AI27" ;


    (ALL,I,IND)
    PR("K",I,"China")
            FROM FILE  CHIIO  HEADER  "AI14" ;

   (ALL,I,IND)
    PR("LND",I,"China")
            FROM FILE  CHIIO  HEADER  "AI15" ;

   (ALL,I,IND)
```

```
    PR("LABO",I,"China")
            FROM FILE  CHIIO  HEADER  "AI13" ;


!-----------------------------------------------------------------
                HKN DATA
 ----------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","D",I,J,"HK")
            FROM FILE  HKIO  HEADER  "AI01" ;

    (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","IM",I,J,"HK")
            FROM FILE  HKIO  HEADER  "AI02" ;

    (ALL,I,IND)
    FINDEM("I","D",I,"HK")
            FROM FILE  HKIO  HEADER  "AI03" ;

    (ALL,I,IND)
    FINDEM("I","IM",I,"HK")
            FROM FILE  HKIO  HEADER  "AI04" ;

    (ALL,I,IND)
    FINDEM("C","D",I,"HK")
            FROM FILE  HKIO  HEADER  "AI05" ;

    (ALL,I,IND)
    FINDEM("C","IM",I,"HK")
            FROM FILE  HKIO  HEADER  "AI06" ;

    (ALL,I,IND)
    FINDEM("G","D",I,"HK")
            FROM FILE  HKIO  HEADER  "AI07" ;

    (ALL,I,IND)
    FINDEM("G","IM",I,"HK")
            FROM FILE  HKIO  HEADER  "AI08" ;


    (ALL,I,IND)
    ODT(I,"HK")
            FROM FILE  HKIO  HEADER  "AI27" ;


    (ALL,I,IND)
    PR("K",I,"HK")
            FROM FILE  HKIO  HEADER  "AI14" ;

    (ALL,I,IND)
    PR("LND",I,"HK")
            FROM FILE  HKIO  HEADER  "AI15" ;
```

```
   (ALL,I,IND)
    PR("LABO",I,"HK")
            FROM FILE  HKIO  HEADER  "AI13" ;


!-------------------------------------------------------------
                  TaiwanN DATA
 -------------------------------------------------------------!
READ (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","D",I,J,"Taiwan")
            FROM FILE  TAIIO  HEADER  "AI01" ;

   (ALL,I,IND)(ALL,J,IND)
    FINDO("IIP","IM",I,J,"Taiwan")
            FROM FILE  TAIIO  HEADER  "AI02" ;

   (ALL,I,IND)
    FINDEM("I","D",I,"Taiwan")
            FROM FILE  TAIIO  HEADER  "AI03" ;

   (ALL,I,IND)
    FINDEM("I","IM",I,"Taiwan")
            FROM FILE  TAIIO  HEADER  "AI04" ;

   (ALL,I,IND)
    FINDEM("C","D",I,"Taiwan")
            FROM FILE  TAIIO  HEADER  "AI05" ;

   (ALL,I,IND)
    FINDEM("C","IM",I,"Taiwan")
            FROM FILE  TAIIO  HEADER  "AI06" ;

   (ALL,I,IND)
    FINDEM("G","D",I,"Taiwan")
            FROM FILE  TAIIO  HEADER  "AI07" ;

   (ALL,I,IND)
   FINDEM("G","IM",I,"Taiwan")
            FROM FILE  TAIIO  HEADER  "AI08" ;


   (ALL,I,IND)
    ODT(I,"Taiwan")
           FROM FILE  TAIIO  HEADER  "AI27" ;


    (ALL,I,IND)
    PR("K",I,"Taiwan")
            FROM FILE  TAIIO  HEADER  "AI14" ;

   (ALL,I,IND)
```

```
    PR("LND",I,"Taiwan")
            FROM FILE  TAIIO  HEADER  "AI15" ;

   (ALL,I,IND)
    PR("LABO",I,"Taiwan")
            FROM FILE  TAIIO  HEADER  "AI13" ;


!*********************************************************************!
!calculate capital, labour and land for ROW!

COEFFICIENT (ALL,I,IND) (ALL,Z,REG)     KAP(I,Z)
!CAPITAL FROM SINGLE REGIONS!;
FORMULA (ALL,I,IND) (ALL,Z,REG)
KAP(I,Z)= PR("K",I,Z);

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)     LN(I,Z)
!LAND FROM SINGLE REGIONS!;
FORMULA (ALL,I,IND) (ALL,Z,REG)
LN(I,Z)= PR("LND",I,Z);

COEFFICIENT (ALL,I,IND)(ALL,Z,REG)     LAB(I,Z)
!LABOUR FROM SINGLE REGIONS!;
FORMULA (ALL,I,IND) (ALL,Z,REG)
LAB(I,Z)= PR("LABO",I,Z);

COEFFICIENT (ALL,I,AGR)              REGLAND(I)
!AG LAND FOR SALTER REGIONS!;
FORMULA (ALL,I,AGR)
REGLAND(I)=SUM(Z,REG,LN(I,Z));

COEFFICIENT (ALL,I,AGR)              REGCAP(I)
!AG CAPIATL FOR SALTER REGIONS!;
FORMULA (ALL,I,AGR)
REGCAP(I)=SUM(Z,REG,KAP(I,Z));

COEFFICIENT (ALL,I,AGR)              REGCL(I);
FORMULA (ALL,I,AGR)
REGCL(I)=REGLAND(I)+REGCAP(I) ;

COEFFICIENT (ALL,I,AGR)              SHLAND(I);
FORMULA (ALL,I,AGR)
SHLAND(I)=REGLAND(I)/REGCL(I);

COEFFICIENT (ALL,I,AGR)             SHCAP(I);
FORMULA (ALL,I,AGR)
SHCAP(I)=REGCAP(I)/REGCL(I);

COEFFICIENT (ALL,I,IND)             r_cap(I);
FORMULA (ALL,I,AGR)  r_cap(I)=CAP(I)*SHCAP(I);
       (ALL,I,NAGR) r_cap(I)=CAP(I);
```

```
COEFFICIENT (ALL,I,IND)            r_land(I);
FORMULA (ALL,I,AGR)  r_land(I)=CAP(I)*SHLAND(I);
       (ALL,I,NAGR) r_land(I)=0.0;



!*************************************************************!
!calculate intermediates, investment, consumption and govt. consumption!

COEFFICIENT (ALL,I,IND) (ALL,J,IND)(ALL,Z,REG)  DINT(I,J,Z)
!DOMESTIC INT INPUTS!;
FORMULA (ALL,I,IND) (ALL,J,IND)(ALL,Z,REG)
DINT(I,J,Z)=FINDO("IIP","D",I,J,Z);

COEFFICIENT (ALL,I,IND) (ALL,J,IND) (ALL,Z,REG)  SIINT(I,J,Z)
!IMPORTED INT INPUTS!;
FORMULA (ALL,I,IND) (ALL,J,IND)(ALL,Z,REG)
SIINT(I,J,Z)=FINDO("IIP","IM",I,J,Z);

COEFFICIENT (ALL,I,IND)(ALL,Z,REGN)          DINV(I,Z)
!DOMESTIC INVESTMENT!;
FORMULA (ALL,I,IND)(ALL,Z,REGN)
DINV(I,Z)=FINDEM("I","D",I,Z);

COEFFICIENT (ALL,I,IND) (ALL,Z,REG)          SIINV(I,Z)
!IMPORTED INVESTMENT!;
FORMULA (ALL,I,IND)(ALL,Z,REG)
SIINV(I,Z)=FINDEM("I","IM",I,Z);

COEFFICIENT (ALL,I,IND) (ALL,Z,REGN)          DCON(I,Z)
!DOMESTIC CONSUMPTION!;
FORMULA (ALL,I,IND)(ALL,Z,REGN)
DCON(I,Z)=FINDEM("C","D",I,Z);

COEFFICIENT (ALL,I,IND) (ALL,Z,REG)          SICON(I,Z)
!IMPORTED CONSUMPTION!;
FORMULA (ALL,I,IND)(ALL,Z,REG)
SICON(I,Z)=FINDEM("C","IM",I,Z);

COEFFICIENT (ALL,I,IND) (ALL,Z,REGN)          DGOV(I,Z)
!DOMESTIC GOVERNMENT CONSUMPTION!;
FORMULA (ALL,I,IND)(ALL,Z,REGN)
DGOV(I,Z)=FINDEM("G","D",I,Z);

COEFFICIENT (ALL,I,IND) (ALL,Z,REG)          SIGOV(I,Z)
!IMPORTED GOVERNMENT CONSUMPTION!;
FORMULA (ALL,I,IND)(ALL,Z,REG)
SIGOV(I,Z)=FINDEM("G","IM",I,Z);


COEFFICIENT (ALL,I,IND)(ALL,J,IND)          INTS(I,J);
!intermediates for the world excl. ROW!
```

```
FORMULA (ALL,I,IND)(ALL,J,IND)
INTS(I,J) = sum(s,reg,DINT(I,J,S)) + sum(s,reg,SIINT(I,J,S));

COEFFICIENT (ALL,I,IND)(ALL,J,IND)          INT_SH(I,J);
FORMULA (all,i,ind)(all,j,ind)
!sum over commodity!
INT_SH(i,j) = INTS(i,j) / sum(F,ind,INTS(F,j));

COEFFICIENT (ALL,I,IND)(ALL,J,IND)          R_INT(i,j);
!disaggregate commodity output to be industry-specific!
FORMULA (all,i,ind)(all,j,ind)
R_INT(i,j) = INT_SH(i,j)*[R_OUT(j) - r_lab(j) - CAP(j)];

COEFFICIENT (ALL,X,REGN)              GDPshare(X);
FORMULA
GDPshare("Thailand") = (ROWGDP("Africa")+ROWGDP("MidE")+ROWGDP("Sasia"))/
                sum(Y,ROW,ROWGDP(Y));
GDPshare("KOREA") =  (ROWGDP("LatAm")+ROWGDP("CPE"))/
                sum(Y,ROW,ROWGDP(Y));
GDPshare("EC") =  (ROWGDP("Weurope"))/
                sum(Y,ROW,ROWGDP(Y));

COEFFICIENT (ALL,I,IND) (ALL,J,IND)      SHIINT(I,J);
!share of imported intermediates in total intermediates for ROW!
FORMULA (ALL,I,IND) (ALL,J,IND)
SHIINT(I,J)=(SIINT(I,J,"Thailand")/(DINT(I,J,"Thailand")+SIINT(I,J,"Thailand"))
        *GDPshare("Thailand"))
        +(SIINT(I,J,"KOREA")/(DINT(I,J,"KOREA")+SIINT(I,J,"KOREA"))
        *GDPshare("KOREA"))
        + (SIINT(I,J,"EC")/(DINT(I,J,"EC")+SIINT(I,J,"EC"))
        *GDPshare("EC"));

COEFFICIENT (ALL,I,IND)(ALL,J,IND)              r_iint(I,J);
!imported intermediates for ROW!
FORMULA (all,i,ind)(all,j,ind)
r_iint(i,j) = SHIINT(i,j)*R_INT(i,j);

COEFFICIENT (ALL,I,IND)(ALL,J,IND)              r_dint(I,J);
!domestic intermediates for ROW!
FORMULA (all,i,ind)(all,j,ind)
r_dint(i,j)=R_INT(i,j)-r_iint(i,j);

COEFFICIENT (ALL,I,IND)                     SHIINV(I);
!share of imported investment!
FORMULA (ALL,I,IND)
SHIINV(I)=(SIINV(I,"Thailand")/(DINV(I,"Thailand")+SIINV(I,"Thailand"))
        *GDPshare("Thailand"))+
        (SIINV(I,"KOREA")/(DINV(I,"KOREA")+SIINV(I,"KOREA"))
        *GDPshare("KOREA"))+
        (SIINV(I,"EC")/(DINV(I,"EC")+SIINV(I,"EC"))
        *GDPshare("EC"));
```

```
COEFFICIENT (ALL,I,IND)                    r_iinv(I);
!imported investment for ROW!
FORMULA (all,i,ind)
r_iinv(i) = SHIINV(i)*R_INV(i);

COEFFICIENT (ALL,I,IND)                    r_dinv(I);
!domestic investment for ROW!
FORMULA (all,i,ind)
r_dinv(i)=R_INV(i)-r_iinv(i);

COEFFICIENT (ALL,I,IND)                    SHICON(I);
!share of imported consumption!
FORMULA (ALL,I,IND)
SHICON(I)=(SICON(I,"Thailand")/(DCON(I,"Thailand")+SICON(I,"Thailand"))
        *GDPshare("Thailand"))+
        (SICON(I,"KOREA")/(DCON(I,"KOREA")+SICON(I,"KOREA"))
        *GDPshare("KOREA"))+
        (SICON(I,"EC")/(DCON(I,"EC")+SICON(I,"EC"))
        *GDPshare("EC"));

COEFFICIENT (ALL,I,IND)                    r_icon(I);
!imported consumption for ROW!
FORMULA (all,i,ind)
r_icon(i) = SHICON(i)*R_CON(i);

COEFFICIENT (ALL,I,IND)                    r_dcon(I);
!domestic consumption for ROW!
FORMULA (all,i,ind)
r_dcon(i) = R_CON(i)-r_icon(i);

COEFFICIENT (ALL,I,IND)                    SHIGOV(I)
!SHARE OF IMPORTED GOVERNMENT CONSUMPTION!;
FORMULA (ALL,I,IND)
SHIGOV(I) = (SIGOV(I,"Thailand")/(DGOV(I,"Thailand")+SIGOV(I,"Thailand"))
        *GDPshare("Thailand"))+
        (SIGOV(I,"KOREA")/(DGOV(I,"KOREA")+SIGOV(I,"KOREA"))
        *GDPshare("KOREA"))+
        (SIGOV(I,"EC")/(DGOV(I,"EC")+SIGOV(I,"EC"))
        *GDPshare("EC"));

COEFFICIENT (ALL,I,IND)                    r_igov(I);
!imported govt consumption for ROW!
FORMULA (all,i,ind)
r_igov(i) = SHIGOV(i)*r_govt(i);

COEFFICIENT  (ALL,I,IND)                   r_dgov(I);
!domestic govt consumption for ROW!
FORMULA (all,i,ind)
r_dgov(i) = R_GOVT(i) - r_igov(i);


!*********************************************************************!
```

```
!calculate nominal rate of protection for ROW!

COEFFICIENT (all,i,ind)(all,z,reg)              VALIMPS(i,z);
!total value of imports including duty (not for ROW)  !
FORMULA (all,i,ind)(all,z,reg)
VALIMPS(i,z) = sum(j,ind,SIINT(i,j,z))
          + SICON(i,z) + SIGOV(i,z) + SIINV(i,z);

COEFFICIENT (all,i,ind)                         NETIMPS(i);
!imports to ROW net of duty!
FORMULA (all,i,ind)
NETIMPS(i) = sum(s,sec,IMPS(s,"rofw",i));


COEFFICIENT (all,i,ind)                         np(i);
!estimates of the value of nominal protection for the ROW by industry!
FORMULA (all,i,ind)
np(i) = sum(z,reg,(ODT(i,z)/(VALIMPS(i,z)-ODT(i,z))))/15 * NETIMPS(i);

coefficient (all,i,ind) rates(i);
formula (all,i,ind)
rates(i) = np(i)/netimps(i);
display rates;

COEFFICIENT (all,i,ind)                         row_imps(i);
!required for the RAS!
formula (all,i,ind)
row_imps(i) = sum(s,sec,IMPS(s,"rofw",i)) + np(i);

!*************************************************************************!
!calculate exports for ROW!

COEFFICIENT (ALL,D,SEC)(ALL,I,IND)              row_exp(D,I);
!exports by destination for ROW!
FORMULA (ALL,D,SEC)(ALL,I,IND)
row_exp(d,i) = EXPS("rofw",d,i);


coefficient  (all,i,ind)                        exports(I);
formula (all,i,ind)
exports(I) = sum(d,sec,row_exp(d,i));
exports("ind34") = exports("ind34") + row_freight;

!*************************************************************************!
!tax coefficients!

COEFFICIENT (ALL,I,IND)                         TSR(I);
FORMULA (ALL,I,IND)
TSR(I) = 0;

COEFFICIENT (ALL,I,IND)(ALL,J,IND)              TRD(I,J);
FORMULA (ALL,I,IND)(ALL,J,IND)
```

```
TRD(I,J) = 0;

COEFFICIENT (ALL,I,IND)(ALL,J,IND)        TRI(I,J);
FORMULA (ALL,I,IND)(ALL,J,IND)
TRI(I,J) = 0;

COEFFICIENT (ALL,I,IND)                   TCRD(I);
FORMULA (ALL,I,IND)
TCRD(I) = 0;

COEFFICIENT (ALL,I,IND)                   TCRI(I);
FORMULA (ALL,I,IND)
TCRI(I) = 0;

COEFFICIENT (ALL,I,IND)                   TIRD(I);
FORMULA (ALL,I,IND)
TIRD(I) = 0;

COEFFICIENT (ALL,I,IND)                   TIRI(I);
FORMULA (ALL,I,IND)
TIRI(I) = 0;

COEFFICIENT (ALL,I,IND)                   TGRD(I);
FORMULA (ALL,I,IND)
TGRD(I) = 0;

COEFFICIENT (ALL,I,IND)                   TGRI(I);
FORMULA (ALL,I,IND)
TGRI(I) = 0;

COEFFICIENT (ALL,I,IND)                   TER(I);
FORMULA (ALL,I,IND)
TER(I) = 0;


!*****************************************************************!
!income tax, depreciation, transfers and savings!

COEFFICIENT               DEPRW
!depreciation!;
FORMULA
DEPRW=578490.4;

COEFFICIENT               TAXYL
!income tax!;
FORMULA
TAXYL=643011.2 ;

COEFFICIENT               TRANSF;
!transfers!
FORMULA
```

```
TRANSF=0;

!savings
COEFFICIENT (ALL,S,SEC)                 WorldGDP(s);
FORMULA (ALL,Z,REG)
WorldGDP(Z) = sum(i,ind,(sum(j,ind,DINT(I,J,Z))+
                    sum(j,ind,SIINT(I,J,Z))))+
          sum(i,ind,KAP(I,Z))+
          sum(i,ind,LN(I,Z))+
          sum(i,ind,LAB(I,Z));
FORMULA
WorldGDP("ROFW") = sum(i,ind,sum(j,ind,R_INT(i,j)))+
                sum(i,ind,r_lab(i))+
                sum(i,ind,r_cap(i))+
                sum(i,ind,r_land(i));

COEFFICIENT (ALL,Z,REG)         Sratio(Z);
FORMULA (ALL,Z,REG)
Sratio(Z) = sum(i,ind,SAVINGS(I,Z))/WorldGDP(Z);

COEFFICIENT                     Saverage;
FORMULA
Saverage = sum(z,reg,Sratio(Z))/15 ;

COEFFICIENT                     SVT;
FORMULA
SVT = WorldGDP("ROFW")*Saverage;
!

COEFFICIENT                     SVT;
FORMULA
SVT = 0;

coefficient out11;
formula out11 = sum(i,ind,r_dint(i,"ind11"))+r_dinv("ind11")
                +r_dcon("ind11")+r_dgov("ind11")+exports("ind11");
display out11;
coefficient out10;
formula out10 = sum(i,ind,r_dint(i,"ind10"))+r_dinv("ind10")
                +r_dcon("ind10")+r_dgov("ind10")+exports("ind10");
display out10;
coefficient out9;
formula out9 = sum(i,ind,r_dint(i,"ind9"))+r_dinv("ind9")
                +r_dcon("ind9")+r_dgov("ind9");
display out9;

!*************************************************************************!

write r_dint to file row_ras header  "AI01";
write r_dinv to file row_ras header  "AI03";
write r_dcon to file row_ras header  "AI05";
write r_dgov to file row_ras header  "AI07";
```

```
write r_iint to file row_ras header  "AI02";
write r_iinv to file row_ras header  "AI04";
write r_icon to file row_ras header  "AI06";
write r_igov to file row_ras header  "AI08";

write exports to file row_ras header "AI11";

write r_lab to file row_ras header   "AI13";
write r_cap to file row_ras header   "AI14";
write r_land to file row_ras header  "AI15";

write np to file row_ras header      "AI27";

write TSR to file row_ras header     "AI12";
write TRD to file row_ras header     "AI16";
write TRI to file row_ras header     "AI17";
write TCRD to file row_ras header    "AI18";
write TCRI to file row_ras header    "AI19";
write TIRD to file row_ras header    "AI20";
write TIRI to file row_ras header    "AI21";
write TGRD to file row_ras header    "AI22";
write TGRI to file row_ras header    "AI23";
write TER to file row_ras header     "AI24";

write TRANSF to file row_ras header  "AI29";
write TAXYL to file row_ras header   "AI28";
write DEPRW to file row_ras header   "AI31";
write SVT to file row_ras header     "AI32";

!headers required for the RAS!

write row_imps to file row_ras header "imp";
write r_out to file row_ras header "out";
write r_int to file row_ras header "rint";
write int_sh to file row_ras header "ints";
write dint to file row_ras header "dint";
write siint to file row_ras header "iint";
write r_lab to file row_ras header "lab";
write cap to file row_ras header "cap";

display row_iinps;
display r_out;
display exports;
```

RAS.STI

```
s            [ single stage
100
n            [ number of steps; use BCV file?
temp           [ name of .eq4 file
temp           [ name of model
1            [ model number; version
1
temp           [ name of .sl4 file
make.out       [ name of file to be RASed
ras100.out     [ RAS output; same name for working files

p            [respond to prompts
a            [set all of variable y exog
n            [set all of variable x endog
u            [use this closure
n            [save an LU file?
0.1          [accuracy factor for matrix ops
a
y
a
**end
```

ROWRAS.TAB

```
s            [ single stage
100
n            [ number of steps; use BCV file?
temp           [ name of .eq4 file
temp           [ name of model
1            [ model number; version
1
temp           [ name of .sl4 file
make.out       [ name of file to be RASed
ras100.out     [ RAS output; same name for working files

p            [respond to prompts
a            [set all of variable y exog
n            [set all of variable x endog
u            [use this closure
n            [save an LU file?
0.1          [accuracy factor for matrix ops
a
y
a
**end
```

APPENDIX C: A DESCRIPTION OF THE CALCULATIONS USED IN ROWVA.TAB AND ROWCIG.TAB

The file ROWVA.TAB finds the following coefficients for each SALTER commodity *(i)* in each subregion *(s)* for ROW:

$$VA(i,s) = \frac{VA(WB,r)}{GDP(r)} \cdot \frac{VA(i,p)}{\sum_{i \in WB} VA(i,p)} \cdot GDP(s)$$

where i = single SALTER mining and agriculture industries
s = Africa, S.Asia, Latin America, CPE and W.Europe
r = representative country
p = proxy country
WB = single World Bank mining or agriculture industry
VA = value added.

$$VA(i,s) = \frac{VA(i,p)}{\sum_{\substack{i \in SALTER \\ industries}} VA(i,p)} \cdot GDP(s)$$

where i = single SALTER mining, food, manufacturing and agriculture industries
s = Mid East
r = representative country
p = proxy country
VA = value added.

$$VA(i,s) = \frac{VA(WB,r)}{GDP(r)} \cdot \frac{VA(UN,r)}{\sum_{\substack{UN \in UN \\ manufactures}} VA(UN,r)} \cdot \frac{VA(i,p)}{\sum_{\substack{i \in SALTER \\ food}} VA(i,p)} \cdot GDP(s)$$

where i = single SALTER food industries
WB = World Bank food industry
UN = single United Nations food industry
s = Africa, S.Asia, Latin America, CPE, W.Europe
r = representative country
p = proxy country
VA = value added.

$$VA(i,s) = \frac{VA(WB,r)}{GDP(r)} \cdot \frac{VA(UN,r)}{\sum_{\substack{UN \in UN \\ manufactures}} VA(UN,r)} \cdot GDP(s)$$

where i = single SALTER manufacturing industry
WB = World Bank manufacturing industry
UN = single United Nations manufacturing industry
s = Africa, S.Asia, Latin America, CPE, W.Europe
r = representative country
VA = value added.

$$VA(i,s) = \frac{VA(WB,r)}{GDP(r)} \cdot \frac{VA(UN,r)}{\sum_{\substack{UN \in UN \\ services}} VA(UN,r)} \cdot GDP(s)$$

where i = single SALTER services industry
WB = World Bank services industry
UN = single United Nations services industry
s = Africa, S.Asia, Latin America, CPE, W.Europe
r = representative country
VA = value added.

$$O(i,s) = \frac{O(NA,r)}{VA(NA,r)} \cdot VA(i,s)$$

where i = single SALTER agriculture, mining and services industries
NA = single National Accounts agriculture, mining and services industries
s = Africa, Latin America, CPE, W.Europe
r = representative country
VA = value added
O = value of output.

$$O(i,s) = \frac{O(UN,r)}{VA(UN,r)} \cdot VA(i,s)$$

where i = single SALTER manufacturing industries
UN = single United Nations manufacturing industries
s = Africa, Latin America, CPE, W.Europe
r = representative country
VA = value added
O = value of output.

$$O(i,s) = \frac{COSTINP(i,p)}{VA(i,p)} \cdot VA(i,s)$$

where i = all single SALTER industries
s = S.Asia, Mid East
p = proxy country
$COSTINP$ = cost of all inputs to production
VA = value added
O = value of output.

$$L(i,s) = \frac{L(NA,r)}{VA(NA,r)} \cdot VA(i,s)$$

where i = single SALTER agriculture, mining and services industries
NA = single National Accounts agriculture, mining and services industries
s = Africa, Latin America, W.Europe
r = representative country
VA = value added
L= payments to labour.

$$L(i,s) = \frac{L(i,p)}{VA(i,p)} \cdot VA(i,s)$$

where i = single SALTER agriculture, mining and services industries
s = S.Asia, Mid East, CPE
p = proxy country
VA = value added
L= payments to labour.

$$L(i,s) = \frac{L(UN,r)}{VA(UN,r)} \cdot VA(i,s)$$

where i = single SALTER food and manufacturing industries
UN = single United Nations manufacturing industries
s = Africa, Latin America, S.Asia, CPE, W.Europe
r = representative country
VA = value added
L= payments to labour.

$$L(i,s) = \frac{L(i,p)}{VA(i,p)} \cdot VA(i,s)$$

where i = single SALTER food and manufacturing industries
s = Mid East
p = proxy country
VA = value added
L= payments to labour.

$$K(i,s) = \frac{K(NA,r)}{VA(NA,r)} \cdot VA(i,s)$$

where i = single SALTER agricultural, mining and services industries
s = Africa, Latin America, W.Europe
r = representative country
VA = value added
K= payments to capital.

$$K(i,s) = \frac{K(i,p)}{VA(i,p)} \cdot VA(i,s)$$

where i = single SALTER agricultural, mining and services industries
s = S.Asia, Mid East and CPE
p = proxy country
VA = value added
K= payments to capital.

$$K(i,s) = \frac{K(UN,r)}{VA(UN,r)} \cdot VA(i,s)$$

where i = single SALTER food and manufacturing industries
s = Africa, S.Asia, Latin America, W.Europe and CPE
r = representative country
VA = value added
K= payments to capital.

$$K(i,s) = \frac{K(i,p)}{VA(i,p)} \cdot VA(i,s)$$

where i = single SALTER food and manufacturing industries
s = Mid East
p = proxy country
VA = value added
K= payments to capital.

The file ROWCIG.TAB finds the following coefficients for each SALTER commodity in each sub-region of ROW:

$$C(i,s) = \frac{C(i,p)}{VA(i,p)} \cdot GDP(s)$$

where C = private consumption
i = all single SALTER industries
s = CPE
p = proxy country
VA = value added.

$$C(i,s) = \frac{C(i,p)}{\sum_{i \in UN} C(i,p)} \cdot C(UN,r) \cdot \frac{GDP(s)}{fGDP(s)}$$

where C = private consumption
$fGDP$ = GDP in domestic currency
UN = United Nations mining, food, agriculture, services and manufacturing industries in domestic currency
i = all single SALTER industries
s = Africa, Latin America, S.Asia, Mid East, W.Europe
p = proxy country
r = representative country.

$$G(i,s) = \frac{G(i,p)}{\sum_{i \in UN} G(i,p)} \cdot G(UN,r) \cdot \frac{GDP(s)}{fGDP(s)}$$

where G = government consumption
$fGDP$ = GDP in domestic currency
UN = United Nations mining, food, agriculture, services and manufacturing industries in domestic currency
i = all single SALTER industries except industries 23,33,34,36
s = Africa, Latin America, S.Asia, Mid East, W.Europe
p = proxy country
r = representative country.

$$G(i,s) = \sum_{UN \in i} G(UN,r) \cdot \frac{GDP(s)}{fGDP(s)}$$

where G = government consumption
i = single SALTER industries 23,33,34,36
UN = single UN commodity corresponding to single SALTER commodity
s = Africa, Latin America, S.Asia, Mid East, W.Europe
r = representative country.

$$G(i,s) = \frac{G(i,p)}{VA(i,p)} \cdot GDP(s)$$

where G = government consumption
i = all single SALTER industries
s = CPE
p = proxy country
VA = value added.

$$I(i,s) = \frac{I(i,p)}{VA(i,p)} \cdot GDP(s)$$

where I = investment
i = all single SALTER industries
s = Africa, Latin America, S.Asia, Mid East, W.Europe, CPE
p = proxy country
VA = value added.

REFERENCES

Brown, S., Strzelecki, A. and Watts, G. 1993, *Matching input-output to international trade data and assembling a Salter Database*, Salter Working Paper No. 19, Industry Commission, Canberra.

Calder, W., McDougall, R. and Strzelecki, A. 1993, *Procedures for later stage processing of single region input-output data for Salter*, Salter Working Paper No. 18, Industry Commission, Canberra.

Hambley, J. 1993, *Early stage processing of international trade and input-output data for Salter*, Salter Working Paper No. 15, Industry Commission, Canberra.

Hanslow, K. 1993, *Later stage processing of the international trade data for Salter*, Salter Working Paper No. 16, Industry Commission, Canberra.

Mikkelsen, A. and Pearson, K.R. 1986, *How to Use Header Array Files to Access Data for Economic Models*, GEMPACK Document Number GED-16, Impact Project, Melbourne.

Codsi, G. and Pearson, K.R. 1988, *An Overview of GEMPACK - A Software System for Implementing and Solving Economic Models*, GEMPACK Document Number GED-22, Impact Project, Melbourne.

United Nations 1973, *Input-Output Tables and Analysis, Study of Methods Series F*, Number 14, Rev 1, New York.

United Nations 1990a, *National Accounts Statistics: Main Aggregates and Detailed Tables, 1987*, New York.

United Nations 1990b, *Industrial Statistics Yearbook, 1988*, New York.

United Nations 1990c, *National Accounts Statistics: Main Aggregates and Detailed Tables, 1988*, New York.

United Nations 1991a, *National Accounts Statistics: Analysis of Main Aggregates, 1988-1989*, New York.

United Nations 1991b, *National Accounts Statistics: Main Aggregates and Detailed Tables, 1989*, New York.

United Nations 1991c, *Industrial Statistics Yearbook, 1989*, New York.

World Bank 1990, *World Development Report, 1990*, Washington DC.

World Bank 1991, *World Development Report, 1991*, Washington DC.