Impact Project

Impact Centre
The University of Melbourne
153 Barry Street, Carlton
Vic. 3053 Australia
Phones: (0[illegible]) [illegible]
Telex: AA [illegible] UNIMEL
Telegrams: UNIMELB Parkville

IMPACT is an economic and demographic research project conducted by Commonwealth Government agencies in association with the Faculty of Economics and Commerce at The University of Melbourne, the Faculty of Economics and Commerce and the Departments of Economics in the Research Schools at the Australian National University and the School of Economics at La Trobe University.

HOW TO USE THE LONGRUN PROGRAMS TO EXTEND OR MODIFY AN ORANI SOLUTION

by

Mark Horridge
IMPACT Project Research Centre

Computing Document No. C7-01 Melbourne November 1985

CONTENTS

		Page
1.	INTRODUCTION	1
2.	OVERVIEW	2
3.	HOW TO USE THE EXAMPLE JOB DECKS	6
4.	PROGRAM SHARES	8
5.	PROGRAM EXPAND	17
6.	PROGRAM SWAP	24
7.	PROGRAMS TO MANIPULATE SOLUTION FILES	29
7.1	Program ROWTOT	32
7.2	Program APPEND	32
7.3	Program LINCOM	36
7.4	Program PIEDIT	36
8.	PROGRAMS TO PRINT SOLUTION FILES	40
8.1	Program FLASH	40
8.2	Program AGGREG	43
REFERENCES		45
APPENDIX I	Compatibility with Different Sizes of ORANI	46
APPENDIX II	Structure of the PIEOUT File	47
APPENDIX III	Derivation of New Variables Calculated by Program EXPAND	49
APPENDIX IV	Errors and Error Messages	66

LIST OF TABLES

		Page
Table 1	Example of Job Deck for Program SHARES	10
Table 2	Output from SHARES (a)	11
Table 3	Output from SHARES (b)	12
Table 4	Output from SHARES (c)	13
Table 5	Output from SHARES (d)	14
Table 6	BOTE Program for Macro Scenario	15
Table 7	The FINAL ORANI Variables	18
Table 8	The BACKSOLUTION ORANI Variables	20
Table 9	New Variables Added by Program EXPAND	21
Table 10	Example of Job Deck for Program EXPAND	22
Table 11	Input and Output of Program SWAP	25
Table 12	Example of Job Deck for Program SWAP	26
Table 13	Example of Job Deck for Program ROWTOT	33
Table 14	Example of Job Deck for Program APPEND	34
Table 15	Example of Job Deck for Program LINCOM	37
Table 16	Example of Job Deck for Program PIEDIT	38
Table 17	Example of Job Deck for Program FLASH	41
Table 18	Example of Job Deck for Program AGGREG	42

LIST OF FIGURES

Figure 1	Overview of Standard ORANI Programs	3
Figure 2	Overview of Longrun Extension Procedure	4
Figure 3	Overview of Programs to Manipulate Solution Files	30

HOW TO USE THE LONGRUN PROGRAMS TO EXTEND OR MODIFY AN ORANI SOLUTION

by

Mark Horridge

1. INTRODUCTION

This manual describes the use of a series of computer programs designed to implement a longrun closure of ORANI, a computable general-equilibrium model of the Australian economy. We assume familiarity with the standard version of ORANI as described in Dixon, Parmenter, Sutton and Vincent (DPSV) (1982) and implemented by the Industries Assistance Commission on CSIRONET under NCS (See Sutton (1985)). The theory lying behind the longrun closure is elaborated in Horridge (1985), which complements this guide.

In spite of their genesis, none of the programs is limited to longrun applications. Rather, the suite comprises a toolbox which may be used to extend or modify a wide range of ORANI solutions. The component programs may be used singly or in various combinations. Nevertheless, this manual allocates central place to the particular sequence of programs required for the longrun closure described in Horridge (1985).

To use the manual, skim through it quickly once, returning to the section which is of greatest interest. Key words which have a special meaning in the context of the ORANI computer programs appear in capital letters.

None of the programs described here are useful in isolation; they all form part of some sequence of computations, perhaps involving half a dozen programs or more. Careful planning of the sequence is therefore essential. Appendix II, which describes file structure, is required reading for the user of nearly all of the programs and should be consulted at the planning stage.

2. OVERVIEW

The longrun programs manipulate the solution produced by the standard ORANI programs in order to implement a model which is specified slightly differently from that described in DPSV. Figure 1 illustrates the generation of a standard ORANI solution, in a very simplified way. Three distinct forms of data are fed into the model solution procedure. Firstly, there is the database, comprising an extended input-out table on a CID file, lists of behavioural parameters on a PARAMS file, and a small amount of behavioural data in the user's input deck. Secondly, the user must specify which variables are to be exogenous, and which endogenous. Of the exogenous variables, only some may need to assume non-zero values. Of the endogenous variables, only a selection need be calculated. All this information appears in the user's input deck.

Using this data, the standard ORANI programs calculate a series of matrices of elasticities showing the response of each selected endogenous variable to a unit proportional change in any of the selected exogenous variables. These matrices are stored on the C and BACKSOL files.

Lastly, the user runs a PIE job, specifying the values of the non-zero exogenous variables (shocks). The PIE program prints out the results of these shocks, and optionally stores them on a PIEOUT or TAPE30 file. It is not necessary, at this stage, that every potentially non-zero exogenous variable be shocked - a second selection is made, from within the previously specified range. Similarly, at the PIE stage, a subset of the endogenous variables previously selected for storage on the C file is chosen for printing out, or storage on a PIEOUT file. The same C and BACKSOL files could form the input to a variety of PIE jobs, utilizing different shock vectors.

Figure 2 illustrates the operation of the longrun extension to ORANI described in Horridge (1985). First, program SHARES derives macro and industry specific data from the standard CID file, supplementing this by user-supplied data, to form a MACSHARE file. These data are not dependent on the closure desired or on the shocks chosen for investigation. Program EXPAND uses a MACSHARE file in conjunction with the C and PIEOUT files generated by a standard ORANI run to produce a MACPIEOUT file. The MACPIEOUT file is of the same format as the PIEOUT file from which it was derived; it contains all the same data <u>plus</u> the endogenous changes in a number of new variables which do not appear in the standard ORANI programs. All the endogenous changes in the MACPIEOUT file refer to the same closure and experiment as the parent PIEOUT file.

The introduction of new variables into the equation system increases the number of closures which are theoretically possible. Program SWAP transforms the data contained in the MACPIEOUT file into endogenous

FIGURE 1: OVERVIEW OF STANDARD ORANI PROGRAMS

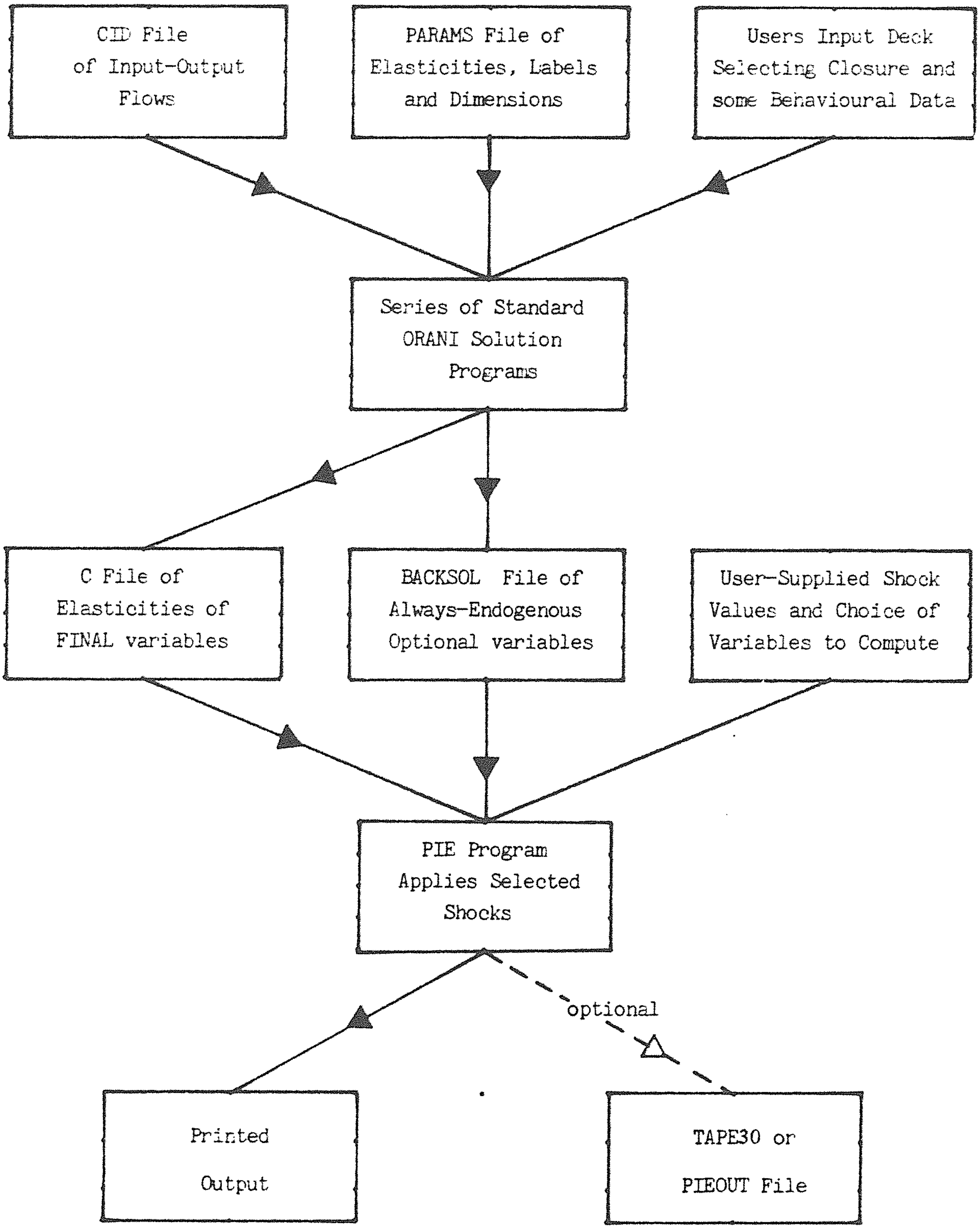
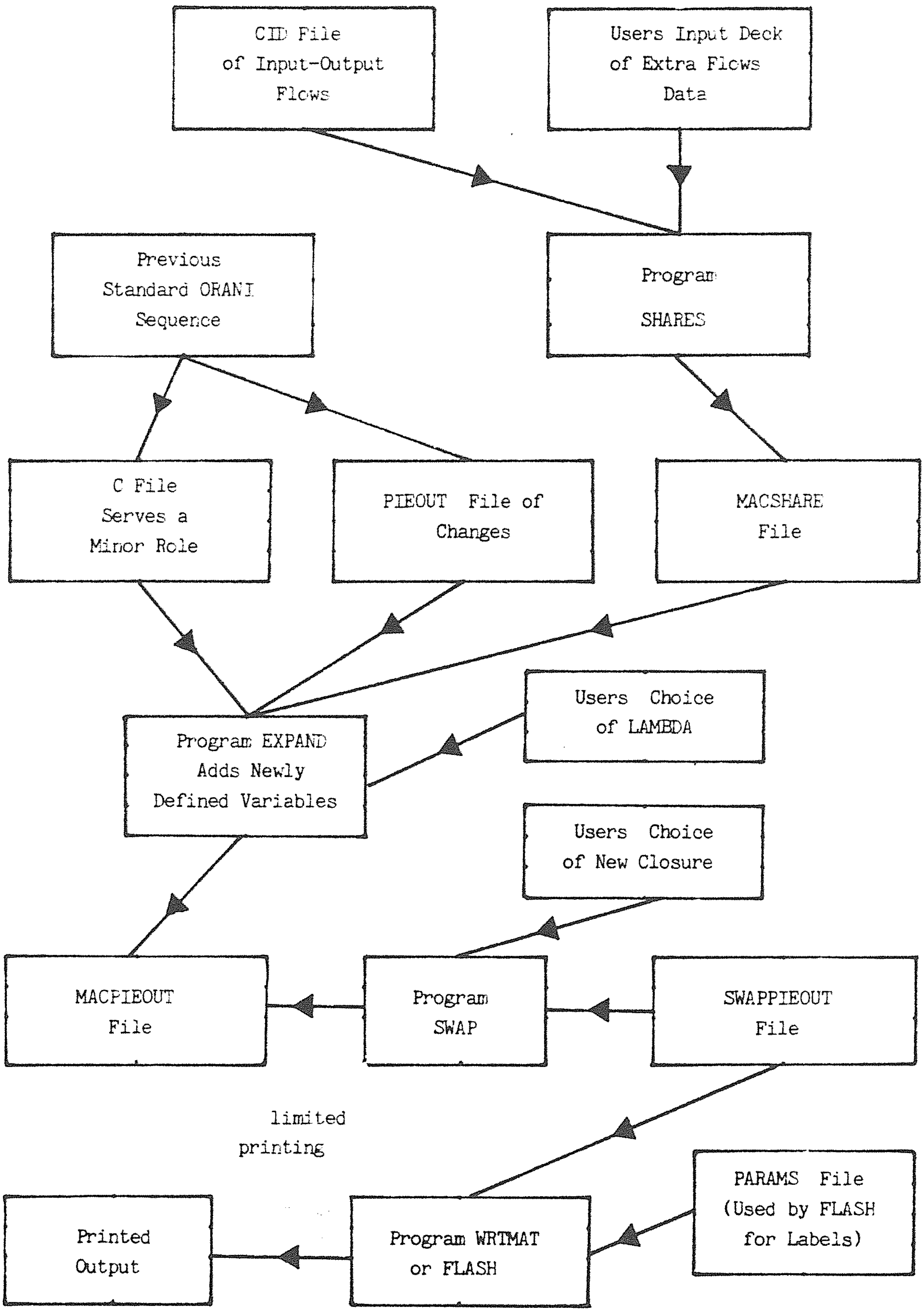


FIGURE 2: OVERVIEW OF LONGRUN EXTENSION PROCEDURE



changes reflecting a different, user specified, closure. The transformed result is stored on a SWAPPIEOUT file, sharing the format of its ancestors MACPIEOUT and PIEOUT. SWAP incorporates a rudimentary print routine; alternatively the SWAPPIEOUT file may be printed out by programs WRTMAT or FLASH.

This sequence was used by Horridge (1985) to transform a standard ORANI solution generated on the assumption that real consumption was held exogenous into a solution where instead the ratio of consumption to saving (F) was exogenous. F is among the new variables generated by program EXPAND, so the same result could not have been achieved using the standard ORANI programs alone.

Thus, the PIEOUT file which is the end result of the standard ORANI computer programs forms the starting point of post-solution manipulation. Many of the longrun programs expect an input file in PIEOUT format and produce a modified file in the same format. A knowledge of the structure of this file is necessary to use any of the longrun programs (except SHARES). It is described in Appendix II.

This manual is ordered as follows. Firstly, the structure of a job deck is described and guidance is offered for the adaptation of the examples provided to the user's needs. Next the programs SHARES, EXPAND, and SWAP are described - the order corresponding to their original use in the generation of a longrun solution. Then come four programs designed to manipulate PIEOUT files; each performs one primitive operation. They may be used singly or in combination at various stages in computation. Lastly, three print programs are described; these are used to print out PIEOUT format files.

Appendix I reassures the user that all the programs are compatible with different sizes of ORANI. Appendix II describes the PIEOUT file format. Appendix III shows the economic content of program EXPAND. Appendix IV comments on errors that may occur.

3. HOW TO USE THE EXAMPLE JOB DECKS

Following yesterday's terminology, we refer to a single line of instructions to the computer as a 'card'. A job deck is a group of cards making up a set of instructions. For each program, an example of the job deck is shown. These examples may be copied from an ED library partition held at the IAC, using the command:

U/DTBGAS/C/program-name,C78UMDE/

and adapted to the user's requirements. One such deck appears as Table 1. Each job deck consists of two parts. The first part of the deck is the job control language (JCL), terminated by the section delimiter card *EOS (End-Of-Sequence). Those parts of the JCL which need to be altered by the user are underlined. They consist merely of the specifications of the input and output files. For example, the card:

ATTACH,TAPE30=XXXXXXX/UN=YYYYYY,PW=ZZZZZZ.

is altered by the user to:

ATTACH,TAPE30=MYINPUT/UN=OWNER,PW=SECRET.

by the insertion, respectively, of the appropriate filename, user identity of the file owner, and the password associated with the file. Note that the tape number is not changed.

The data deck follows the first *EOS card. Any extra numerical data is fed in at this point. All the programs follow the convention that each data item (or group of related items) is preceded by a descriptive card which also shows the FORTRAN format to which the data should conform. Thus the data deck for Program ROWTOT (see Table 13) is of the form:

```
*EOS
MATRIX RANGE...PAIRS OF INTEGERS 2I5: A PAIR OF ZEROS TERMINATES THE LIST
    3    6
    9    9
   13   21
    0    0
COLUMN RANGE...PAIRS OF INTEGERS 2I5: A PAIR OF ZEROS TERMINATES THE LIST
    1  999
    0    0
*EOS
```

The comment lines make the programs easier to interpret. They are ignored by the program - which does however expect them to be there. Thus, if the first card (MATRIX RANGE...) was removed, ROWTOT would ignore instead the second card shown, interpreting this as the comment card. Data must appear in the order shown, as its meaning is determined by its position in the deck. For example, if the block of three COLUMN RANGE cards preceded the MATRIX RANGE cards, ROWTOT would process all matrices on the files, including only columns 3 to 6, 9 and 13 to 21. On the other hand, if the two comment cards alone were exchanged, the operation of ROWTOT would be quite unchanged. Thus the actual contents of the comment cards are irrelevant to the program, and no errors will result if they are copied incorrectly from the example decks. Nevertheless, it is in the user's interest to ensure that the comment cards do actually describe the data which follow them.

Throughout the programs, only three FORTRAN formats are needed for user input - nI5, nF10.5, and nA10. 'n' is the repeat factor, showing the number of entries per line, and is omitted if n = 1. The I5 format specifies a single integer of maximum length 5 digits (including any sign), justified to the right of the five columns allotted to it. 2I5 specifies two such numbers, ending on the fifth and tenth columns of one line.

The nF10.5 format is used for real numbers, which <u>must</u> contain a decimal point. Including the point and any sign, the maximum width of each number is 10 characters. The '5' has no significance in this context. It is not essential that each number is right justified, as long as it occupies only the 10 columns allotted to it. Thus, a single real in F10.5 format must appear within columns 1 to 10, whilst the 8F10.5 specifies 8 reals, occupying, respectively, columns 1 to 10, 11 to 20, ... , 71 to 80.

Lastly, the A10 format is used to read in character strings such as titles and names. Thus 4A10 allows a name to occupy any of the first 40 columns on a line. 4A10 is functionally equivalent to A40 or 2A20 as far as the user is concerned.

4. PROGRAM SHARES

Program SHARES assembles and files the data used by program EXPAND to supplement the variables reported by a standard ORANI run (see Figure 2). The output from SHARES is a MACSHARE file. Conceptually, the MACSHARE file forms part of the database, and so the same file could be used for several experiments perhaps involving different closures. However, a new MACSHARE file must be created for each CID file. Alterations to the PARAMS file do not occasion a rerunning of SHARES. Input data for SHARES falls into three categories:

a) The CID file, containing input-output data. This must be the same file as used in the standard ORANI run.

b) Items of macro data inserted by the user in the SHARES job deck which serve to determine entries in the foreign accounts.

c) User-supplied data which determine the distribution of foreign ownership between the ORANI industries.

Correspondingly, three tables are printed out by the program, showing:

(i) Macro information deducible from the CID file alone. An example is shown as Table 2.

(ii) Additional macro information deducible from data items (a) and (b). An example is shown as Table 3.

(iii) The share of each industry which is owned locally. An example is shown as Table 4.

While data item (a) influences (i), (ii) and (iii), item (b) affects only (ii) and (iii), and (c) influences only (iii).

Table 1 shows an example of a SHARES input deck. As noted in Section 3, user-dependent parts of the JCL are underlined. Comments identify the positions of the input (CID) and output (MACSHARE) files. Like most of the longrun suite of programs, the data deck of program SHARES is also interspersed with comment cards describing the item which is to follow. The program does not check these cards, although it does expect them to be there. Data items must appear in the order indicated.

The first item is the debug flag. When set (to one), the output shares listed in Table 5 will be printed for examination. The next four data constitute item (b) above. They supplement the CID data so that program SHARES may compute and print out the national accounting data presented in Table 2. It is anticipated that the user will obtain the supplementary data from the Australian National Accounts or a similar source. The national accounting conventions set out in Horridge (1985) must be followed. Two particular points to note are:

(i) The value of GDP implied by the CID file may or may not equal that contained in the national accounting data for the same year. Although the ABS input-output tables are compatible with ANA data, various aggregates are altered slightly as the ABS data are transformed into the ORANI database. For this reason the macro data items fed into program SHARES are supplied as fractions of GDP. Thus, although ORANI GDP (as implied by the CID file) may differ from that found in the national accounts, the ratio of, say, rentals from overseas to GDP may be set to the same value for the ORANI simulation as in the national accounts.

(ii) The first two data items are the gross values of capital rentals passing to and from the rest of the world, expressed as fractions of GDP. The second two items, also expressed as fractions, show the destination of national saving, which includes government saving. In concordance with Horridge (1985), this saving should be considered gross (of depreciation). The ANA show net saving and depreciation separately, and allowance must be made for this.

The data presented in Table 3 are determined solely from the CID file and the four components of item (b) above. This scenario should be examined carefully before the subsequent step of running program EXPAND is taken. In particular, the 'balancing item' should be a small fraction of GDP, if the spirit of Horridge (1985) is to be followed. Program SHARES is cheap to run, and several attempts could be made using different data for (b) above. Alternatively, a desk computer or programmable calculator could be used to generate these scenarios. An example program written in BASIC is shown in Table 6.

The third group of data determines the distribution of local ownership of capital amongst the industries. The overall share of capital rentals accruing locally is determined by items (a) and (b) above. The numbers in (c) are supposed to represent the individual shares for each industry of capital rentals accruing to Australians. Taken as a group they could imply a different overall share than that implied by (a) and (b) above. Consequently, program SHARES scales the industry shares so that they are in accordance with the given overall share. The program also offers a facility for setting unknown shares to a typical value. The following rules are observed.

(a) First, if any input share is negative, it is replaced by the overall average share.

TABLE 1: EXAMPLE OF JOB DECK FOR PROGRAM SHARES

```
SHARES.
*INC,USER
COMMENT.***************************************************
COMMENT.*   SHARES                                        *
COMMENT.*   A PROGRAM TO READ A PIEOUT FILE,              *
COMMENT.*   COMPUTE THE ROW TOTALS OF ITS MATRICES,       *
COMMENT.*   AND WRITE THEM TO A NEW FILE.                 *
COMMENT.*   INSTRUCTIONS...SEE IMPACT MANUAL C-701        *
COMMENT.*   NOS VERSION                         JMH 3/85 *
COMMENT.***************************************************
FTNAUG.
RFSGET,DTBIOM.
COMMENT. ATTACH SPECIAL LONGRUN ORANI LIBRARY
ATTACH,ULIB0=078LR4L/UN=DTBIOM.
COMMENT. ATTACH GENERAL PURPOSE ORANI LIBRARY
ATTACH,ULIB1=078UT4L/UN=DTBIOM.
COMMENT. ATTACH ORANI LARGE CHANGE LIBRARY
ATTACH,ULIB2=078LC4L/UN=DTBIOM.
RFSRET,DTBIOM.
LIBRARY,ULIB0,ULIB2,ULIB1/A.
COMMENT.******************************************************
COMMENT. THE FOLLOWING IS THE INPUT CID FILE
ATTACH,TAPE10=XXXXXXX/UN=YYYYYY,PW=ZZZZZZ.
COMMENT.******************************************************
COMMENT. THE FOLLOWING IS THE OUTPUT FILE OF SHARES USED AS INPUT TO EXPAND
DEFINE,TAPE20=XXXXXXX/CT=S,PW=ZZZZZZ.
LDSET,LIB=FTNAUG.
LIBLOAD,ULIB0,SHARES.
EXECUTE,SHARES.
*EOS
DEBUG FLAG- 1 PRINTS OUT ALL INPUT AND OUTPUT DATA - 0 DOESNT- I5 FORMAT
    0
DATA FROM ANA ACCOUNTS:
RENTALS FROM OVERSEAS AS FRACTION OF GDP F8.5
0.00302
RENTALS TO OVERSEAS AS FRACTION OF GDP F8.5
0.04
SAVING BY AUSTRALIANS INVESTED LOCALLY AS FRACTION OF GDP F8.5
0.18252
SAVING BY AUSTRALIANS INVESTED OVERSEAS AS FRACTION OF GDP F8.5
0.00329
ESTIMATED LOCAL SHARES OF EACH INDUSTRY'S CAPITAL STOCK  10F8.5
 -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7
 -0.7    +0.4    +0.47   +0.5    +0.41   -0.7    -0.7    -0.7    -0.7    -0.7
 -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7
 -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7
 -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7
 -0.7    -0.7    -0.7    -0.7    -0.7    +0.43   -0.7    -0.7    -0.7    -0.7
 -0.7    -0.7    -0.7    +0.4    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7
 -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7    -0.7
 -0.7    -0.7    -0.7    +1.0    +1.0    +1.0    -0.7    -0.7    -0.7    -0.7
 +1.0    +1.0    -0.7    +1.0    -0.7    +1.0    -0.7    -0.7    -0.7    -0.7
 -0.7    -0.7    +1.0    +1.0    +1.0    +1.0    +1.0    +1.0    -0.7    -0.7
 +1.0    -0.7
```

TABLE 2: OUTPUT FROM PROGRAM SHARES (a)

ORANI ACCOUNTS

AMOUNTS IN LEVELS:

G.D.P. FROM EXPENDITURE SIDE

HOUSEHOLD CONSUMPTION =	57344.07704
INVESTMENT =	21228.84213
GOVERNMENT (OTHER DEMANDS) =	16233.93636
EXPORTS AT FOREIGN PRICES ($A) =	14477.18856
IMPORTS AT FOREIGN PRICES ($A) =	14545.54512
TOTAL GDP AT PURCHASERS PRICES =	94738.49897

G.D.P. FROM INCOME SIDE

RENTALS TO LAND =	1447.45460
WAGES TO LABOUR =	54381.00370
RENTALS TO CAPITAL =	19653.56020
OTHER COST TICKET REVENUE =	12840.86093
TOTAL INDIRECT TAX REVENUE =	6414.61707
TOTAL =	94737.49650

INDIRECT TAXES BY SOURCE

TAXES ON INTERMEDIATE USAGE =	1238.87995
TAXES ON CAPITAL CREATION USAGE =	177.94900
TAXES ON HOUSEHOLD CONSUMPTION =	3852.38900
NET TAXES ON EXPORTS =	213.12211
NET TAXES ON IMPORTS =	932.27700
TOTAL COMMODITY TAXES =	6414.61707

AMOUNTS AS SHARES

G.D.P. FROM EXPENDITURE SIDE

HOUSEHOLD CONSUMPTION =	.60529
INVESTMENT =	.22408
GOVERNMENT (OTHER DEMANDS) =	.17136
EXPORTS AT FOREIGN PRICES ($A) =	.15281
IMPORTS AT FOREIGN PRICES ($A) =	-.15353
TOTAL GDP AT PURCHASERS PRICES =	1.00000

G.D.P. FROM INCOME SIDE

RENTALS TO LAND =	.01528
WAGES TO LABOUR =	.57402
RENTALS TO CAPITAL =	.20745
OTHER COST TICKET REVENUE =	.13554
TOTAL INDIRECT TAX REVENUE =	.06771
TOTAL =	1.00000

INDIRECT TAXES BY SOURCE

TAXES ON INTERMEDIATE USAGE =	.19313
TAXES ON CAPITAL CREATION USAGE =	.02774
TAXES ON HOUSEHOLD CONSUMPTION =	.60056
NET TAXES ON EXPORTS =	.03322
NET TAXES ON IMPORTS =	.14534
TOTAL COMMODITY TAXES =	1.00000

TABLE 3: OUTPUT FROM PROGRAM SHARES (b)

ORANI ACCOUNTS USING EXTRA DATA

AMOUNTS IN LEVELS:

G.N.P. FROM EXPENDITURE SIDE

HOUSEHOLD CONSUMPTION	=	57344.07704
GOVERNMENT CONSUMPTION-OTHER DEMANDS	=	16233.93636
NATIONAL SAVING INVESTED LOCALLY	=	17291.67083
NATIONAL SAVING INVESTED OVERSEAS	=	311.68966
BALANCING ITEM	=	53.69538
TOTAL	=	91235.06928

G.N.P. FROM INCOME SIDE

LOCAL CAPTL RENTALS ACCRUING LOCALLY	=	15865.02271
OTHER LOCAL INCOME	=	75083.93630
CAPITAL RENTALS FROM OVERSEAS	=	286.11027
TOTAL	=	91235.06928

OTHER QUANTITIES

GROSS CAPITAL RENTALS TO OVERSEAS	=	3789.53996
NET CAPITAL RENTALS TO OVERSEAS	=	3503.42969
NET FOREIGN INVESTMENT	=	3625.48164
LOCAL SHARE OF LOCAL CAPITAL RENTALS	=	.80723

FOREIGN ACCOUNT

INVESTMENT IN AUSTRALIA	=	21228.84213
BALANCE OF TRADE	=	-68.35656
TOTAL LEFT HAND SIDE	=	21160.48557
NATIONAL SAVING	=	17603.36049
NET RENTALS TO OVERSEAS	=	3503.42969
BALANCING ITEM	=	53.69538
TOTAL RIGHT HAND SIDE	=	21160.48557

AMOUNTS AS SHARES

G.N.P. FROM EXPENDITURE SIDE

HOUSEHOLD CONSUMPTION	=	.62853
GOVERNMENT CONSUMPTION-OTHER DEMANDS	=	.17794
NATIONAL SAVING INVESTED LOCALLY	=	.18953
NATIONAL SAVING INVESTED OVERSEAS	=	.00342
BALANCING ITEM	=	.00059
TOTAL	=	1.00000

G.N.P. FROM INCOME SIDE

LOCAL CAPTL RENTALS ACCRUING LOCALLY	=	.17389
OTHER LOCAL INCOME	=	.82297
CAPITAL RENTALS FROM OVERSEAS	=	.00314
TOTAL	=	1.00000

SHARES PROGRAM TERMINATED NORMALLY

TABLE 4: OUTPUT FROM PROGRAM SHARES (c)

VALUES,AS READ IN,OF QHAT - AUSTRALIAN SHARES OF INDUSTRY CAPITAL STOCKS

1	-.700000	2	-.700000	3	-.700000	4	-.700000	5	-.700000	6	-.700000	7	-.700000	8	-.700000	9	-.700000	10	-.700000
11	-.700000	12	.400000	13	.470000	14	.500000	15	.410000	16	-.700000	17	-.700000	18	-.700000	19	-.700000	20	-.700000
21	-.700000	22	-.700000	23	-.700000	24	-.700000	25	-.700000	26	-.700000	27	-.700000	28	-.700000	29	-.700000	30	-.700000
31	-.700000	32	-.700000	33	-.700000	34	-.700000	35	-.700000	36	-.700000	37	-.700000	38	-.700000	39	-.700000	40	-.700000
41	-.700000	42	-.700000	43	-.700000	44	-.700000	45	-.700000	46	-.700000	47	-.700000	48	-.700000	49	-.700000	50	-.700000
51	-.700000	52	-.700000	53	-.700000	54	-.700000	55	-.700000	56	.430000	57	-.700000	58	-.700000	59	-.700000	60	-.700000
61	-.700000	62	-.700000	63	-.700000	64	.400000	65	-.700000	66	-.700000	67	-.700000	68	-.700000	69	-.700000	70	-.700000
71	-.700000	72	-.700000	73	-.700000	74	-.700000	75	-.700000	76	-.700000	77	-.700000	78	-.700000	79	-.700000	80	-.700000
81	-.700000	82	-.700000	83	-.700000	84	1.000000	85	1.000000	86	1.000000	87	-.700000	88	-.700000	89	-.700000	90	-.700000
91	1.000000	92	1.000000	93	-.700000	94	1.000000	95	-.700000	96	1.000000	97	-.700000	98	-.700000	99	-.700000	100	-.700000
101	-.700000	102	-.700000	103	1.000000	104	1.000000	105	1.000000	106	1.000000	107	1.000000	108	1.000000	109	-.700000	110	-.700000
111	1.000000	112	-.700000																

VALUES OF (OZREV-ESTREV) AT EACH ITERATION

1	-824.10	2	-48.755	3	-12.509	4	-3.3055	5	-.87991
6	-.23468	7	-.62621E-01	8	-.16712E-01	9	-.44602E-02	10	-.11904E-02
11	-.31769E-03	12	-.84769E-04	13	-.22629E-04	14	-.60393E-05	15	-.16119E-05
16	-.43015E-06	17	-.11484E-06	18	-.30675E-07	19	-.81491E-08	20	-.22701E-08

EACH QHAT(J) IS SET, IF NEGATIVE, TO THE PREDETERMINED AVERAGE VALUE: .8072
THEN, EACH QHAT(J) IS RAISED TO THE POWER P= 1.36405

REVISED QHAT...AUSTRALIAN SHARES OF INDUSTRY CAPITAL STOCKS

1	.746694	2	.746694	3	.746694	4	.746694	5	.746694	6	.746694	7	.746694	8	.746694	9	.746694	10	.746694
11	.746694	12	.286543	13	.357047	14	.388491	15	.296359	16	.746694	17	.746694	18	.746694	19	.746694	20	.746694
21	.746694	22	.746694	23	.746694	24	.746694	25	.746694	26	.746694	27	.746694	28	.746694	29	.746694	30	.746694
31	.746694	32	.746694	33	.746694	34	.746694	35	.746694	36	.746694	37	.746694	38	.746694	39	.746694	40	.746694
41	.746694	42	.746694	43	.746694	44	.746694	45	.746694	46	.746694	47	.746694	48	.746694	49	.746694	50	.746694
51	.746694	52	.746694	53	.746694	54	.746694	55	.746694	56	.316252	57	.746694	58	.746694	59	.746694	60	.746694
61	.746694	62	.746694	63	.746694	64	.286543	65	.746694	66	.746694	67	.746694	68	.746694	69	.746694	70	.746694
71	.746694	72	.746694	73	.746694	74	.746694	75	.746694	76	.746694	77	.746694	78	.746694	79	.746694	80	.746694
81	.746694	82	.746694	83	.746694	84	1.000000	85	1.000000	86	1.000000	87	.746694	88	.746694	89	.746694	90	.746694
91	1.000000	92	1.000000	93	.746694	94	1.000000	95	.746694	96	1.000000	97	.746694	98	.746694	99	.746694	100	.746694
101	.746694	102	.746694	103	1.000000	104	1.000000	105	1.000000	106	1.000000	107	1.000000	108	1.000000	109	.746694	110	.746694
111	1.000000	112	.746694																

OVERALL AUSTRALIAN SHARE OF CAPITAL RENTALS = .807234

TABLE 5: OUTPUT FROM PROGRAM SHARES (d)

	DESCRIPTION	ROWS/COLS	SYMBOL
1	SHARES FOR GDP=C+I+G+X-M	5 x 1	SHRGDE
2	2,3: SHARES DOMESTIC/IMPORTED COMMODITIES	NG x 1	SHRGVD
3	IN OTHER DEMANDS (TWO VECTORS SUM TO UNITY)	NG x 1	SHRGVI
4	COMMODITY SHARES OF IMPORTS AT AT-PORT PRICES	NG x 1	SHRIMP
5	EXPORT SHARES (INC MARGINS AND SUBSIDIES)	NG x 1	SHREXP
6	WAGE BILL MATRIX	NH x 9	A
7	CAPITAL RENTALS - SHARES BY INDUSTRY	NH x 1	SHRREN
8	LAND RENTALS - SHARES BY INDUSTRY	NH x 1	SHRLAN
9	OCT SHARES BY INDUSTRY	NH x 1	SHROCT
10	TOTAL TARIFF REVENUE	1 x 1	TOTTAR
11	TOTAL EXPORT SUBSIDIES	1 x 1	TOTSUB
12	COMMODITY SHARES IN SALES TAX ON CONSUMPTION	NG x 1	SHRCTX
13	INDUSTRY SHARES IN SALES TAX ON INTERMEDIATE	NH x 1	SHRNTX
14	INDUSTRY SHARES IN SALES TAX ON INVESTMENT	NH x 1	SHRVTX
15	SHARES LAND,LABOUR,CAPITAL,OCT,TAXES IN GDP	5 x 1	SHRGDI
16	SHARES OF DIFFERENT INDIRECT TAXES IN ALL TAX ON INTERMEDIATE USAGE, INVESTMENT, CONSUMPTION, EXPORTS, IMPORTS	5 x 1	SHRTAX
17	SHARES BY COMMODITY OF LOCAL GOOD/BOTH GOODS(BASIC) IN CONSUMPTION	NG x 1	SHRCON
18	INDUSTRY SHARES IN VALUE OF CAPITAL STOCK	NH x 1	SHRCAP
19	DATABASE POWERS OF TARIFFS	NG x 1	POWTAR
20	DATABASE POWERS OF EXPORT SUBSIDIES	NG x 1	POWSUB
21	FLOWS OF IMPORTS AT FOREIGN PRICES	NG x 1	IMPORT
22	FLOWS OF EXPORTS AT FOREIGN PRICES	NG x 1	EXPORT
23	INDUSTRY SHARES OF LOCALLY-ACCRUING CAPITAL RENTALS (SUMMING TO ONE)	NH x 1	SHRLOC
24	SHARES C,G,SD,SF,AND BI IN GNP EXP	5 x 1	SHRGNE
25	SHARES OZREV, OTHINC, RIFANA IN GNP INC	3 x 1	SHRGNI
26	RITANA = Q1(1)*CAPTAL + Q1(2)*OZREN	2 x 1	Q1
27	RX = Q2(1)*RITANA + Q2(2)*RIFANA	2 x 1	Q2
28	RI = Q3(1)*AGINVN + Q3(2)*SAVING	2 x 1	Q3
29	INDUSTRY SHARES,TOTAL NOMINAL INVESTMENT	NH x 1	SHRINV

TABLE 6: BOTE PROGRAM FOR MACRO SCENARIO

```
100 GDP    = 95255.86                      :REM  ORANI GDP
    OTHINC = 75513.05                      :REM  ORANI GDP LESS RENTALS TO CAPITAL
    AGCONN = 57495.76                      :REM  ORANI AGGREGATE CONSUMPTION
    TOTOTH = 16221.65                      :REM  ORANI AGGREGATE OTHER DEMANDS
    AGINVN = 21244.77                      :REM  ORANI AGGREGATE INVESTMENT
    BT = 293.66                            :REM  ORANI BALANCE OF TRADE
    RIFANA = GDP*.0030163                  :REM  RIFANA = RENTALS FROM OVERSEAS
    BI = GDP*.00331466                     :REM  BI = BALANCING ITEM
            :REM NEXT ITEM IS ALTERED UNTIL SCENARIO IS SATISFACTORY
    INPUT 'RITANA/GDP';RITANA              :REM  RITANA = RENTALS TO OVERSEAS
    RITANA = RITANA*GDP                    :REM  CONVERT TO ABSOLUTE FORM
    GNP = GDP + RIFANA - RITANA            :REM
    OZREV = GNP - OTHINC - RIFANA          :REM  OZREV = ALL LOCAL RENTALS EARNED BY OZ
    SAVING = GNP - AGCONN - TOTOTH - BI    :REM
    RX = RITANA - RIFANA                   :REM  RX = NET RENTALS TO OVERSEAS
    RI = AGINVN - SAVING                   :REM  RI = NET FOREIGN INVESTMENT
    SAVDOM = SAVING/(1+RIFANA/OZREV)       :REM  AUSTRALIAN SAVING INVESTED LOCALLY
    SAVFOR = SAVING - SAVDOM               :REM  AUSTRALIAN SAVING INVESTED O'SEAS
    TOTREN = GDP - OTHINC                  :REM  ALL LOCAL RENTALS
    QTOT = OZREV/TOTREN                    :REM  OVERALL LOCAL OWNERSHIP SHARE
    FC = AGCONN/SAVING                     :REM  F= RATIO CONSUMPTION/SAVING
    LPRINTER                               :REM  NOW PRINT RESULTS
    REM: THE RESULTS PRINTING SECTION IS NOT REPRODUCED HERE
    REM: PRINT.......................
    CONSOLE
    GO TO 100           :REM LOOK AT THE RESULTS, ADJUST INPUTS AND TRY AGAIN
```

(c) Second, all the shares are raised to the same power P so that the overall local share is that which is already given by the macro data. P is printed out.Its value is found by an iterative process.

Corollaries of these rules are that:

An input share of zero (total foreign ownership) will be preserved unchanged; and that
An input share of one (total domestic ownership) will be preserved unchanged.
If all input shares are zero or one there is no room for adjustment and the routine is likely to crash.
The magnitude of a negative input share is irrelevant.

No loss of flexibility is implied by these rules since, if a vector of shares consistent with the overall share is fed in it will be virtually unchanged by the scaling procedure. On the other hand, the scaling may be used to generate scenarios consistent with macro data, as exemplified by Section 3 of Horridge (1985). The (adjusted) shares are printed out as in Table 4.

Optionally, Program SHARES also prints out the data filed for later use by program EXPAND. It appears mainly for diagnostic purposes. Table 5 lists this information by the labels under which it is printed out and indicates the uses to which it is put.

5. PROGRAM EXPAND

Program EXPAND takes as input a PIEOUT result file containing results for some or all of the standard ORANI variables listed in Tables 7 and 8 (see Figure 2). It then creates a second version, reproducing the first positions in the input file and at the same time extending the solution by the calculation of 30 new variables listed in Table 9. A full description of these new variables may be found in Appendix III. Note that the numbering of the variables, corresponding to their position on the PIEOUT file, continues the sequence of the standard ORANI variables listed in Tables 7 and 8.

The full data requirements for the program EXPAND are as follows.

a) A PIEOUT file containing information on all non-zero final vector variables plus information on the backsolution variables marked with an asterisk in Table 8. It is necessary that the first matrix in the input PIEOUT file be present, showing the responses of industry outputs to the shocks. This matrix is used to find the number of shocks for use in subsequent calculations. Thus, the (unlikely) possibility of holding all industry outputs exogenously constant is excluded.

Program EXPAND prints out warnings if any of variables 39 to 52 - the YUK terms - are non-zero. In this case certain of the results may be invalid. Consult Appendix III to discover which variables are invalidated by which YUK shocks.

b) The C file from which the input PIEOUT file was prepared. This file is used to recover data relating to the numbers of industries, commodities, and occupations. Values of export-elasticities and of the indexation parameters for wages, other cost tickets and government demands are also drawn from the C file. In fact, as long as this data is correct, any C file may be used.

c) A file of shares created by program SHARES (see Section 4). This MACSHARE file combines a digest of the ORANI data files with supplementary data used to calculate some of the new variables calculated by Program EXPAND.

d) The input deck for Program EXPAND contains a value for λ - a parameter used to relate changes in the stream of savings by Australians to changes in their stock of wealth. For shortrun work, this should be set to zero. Otherwise, choose a value between zero and one, using the discussion in pp. 11-15 of Horridge (1985) as a guide.

Table 10 shows an example of the job deck used to run program EXPAND. Those parts which are user-specific are underlined. Comment cards,

TABLE 7: THE FINAL ORANI VARIABLES

CODE NO.	COMPUTER MNEMONIC	DESCRIPTION	NO. OF COMPONENTS
1	Z	Industry outputs (real)	h
2	XA	Agricultural commodity outputs (real)	s
3	X4	Exogenous commodity exports (real)	g-NEX
4	X2	Exogenous commodity imports (real)	NIM
5	PA	Agricultural commodity prices ($BV)	s
6	S	Powers of exogenous export subsidies[d]	NEX
7	PM	Import prices (foreign currency)	g
8	T	Tariffs on imports (power of the tariff)[d]	g
9	ER	Exchange rate ($A/$US)	1
10	R	Rates of return by industry (real)	h
11	Q3	Rental prices on agricultural land ($)	1
12	N1	Employment by occupation (real)	m
13	K0	Base period capital stocks by industry(real)	h
14	N3	Quantity of agricultural land	1
15	BT	Balance of trade ($A billion)	1
16	M	Aggregate imports ($ value in foreign currency)	1
17	E	Aggregate exports ($ value in foreign currency)	1
18	IP	Private investment price index ($)	1
19	CP	Consumer price index ($)	1
20	IR	Aggregate real private investment	1
21	CR	Aggregate real household consumption	1
22	FR	Ratio of real consumption(21) to investment(20)	1
23	U1[b]	Aggregate employment (wagebill weights)	1
24	U2	Aggregate employment (persons weights)	1
25	GN[c]	Real National Output	1
26	K	Aggregate capital stock (real)	1
27	IM	Aggregate nominal private investment ($)	1
28	CM	Aggregate nominal household consumption ($)	1
29	Q	Number of households	1
30	LM	Economy-wide rate of return (real)	1
31	FE	Shift term for exports (real)	g

Continued.....

TABLE 7: (continued)

CODE NO.	COMPUTER MNEMONIC	DESCRIPTION	NO. OF COMPONENTS
32	FO	Shift term for occupation wage rates ($)	m
33	FI	Shift term for industry wage rates ($)	h
34	FG	Shift term for average wage rate ($)	1
35	FX	Shift term for other costs ($)	h
36	F2	Shift term for exogenous investment(real)	h
37	F5	Shift term for other usage(domestic,real)	g
38	F6	Shift term for other usage (imports) (real)	g
39	YI	YUK[d] - cost of capital equation	h
40	YP	YUK[d] - price equation	h
41	YC	YUK[d] - domestic consumption equation	g
42	ZC	YUK[d] - import consumption equation	g
43	YK	YUK[d] - labour demand equation	m
44	YK	YUK[d] - capital demand equation	h
45	YN	YUK[d] - land demand equation	1
46	YX	YUK[d] - domestic market clearing equation	g
47	ZX	YUK[d] - imports market clearing equation	g
48	YW	YUK[d] - export cost equation	g
49	Y3	YUK[d] - CPI equation	1
50	Y2	YUK[d] - import price equation	g
51	YA	YUK[d] - CRETH equation	s
52	XC	Extra variables	
53	W1	Shift term for wages by occupation and industry - occupation 1($)	h
to			
52+m	Wm	Shift term for wages by occupation and industry - occupation m ($)	h

a This variable does not appear in the ORANI theory as set out in DPSV.

b This variable does not appear in the ORANI theory as set out in DPSV. It is recommended that it be endogenous and that its results be ignored because the formula used is unsatisfactory in some circumstances.

c The YUK variables correspond to the shift variables bk in the condensed system of equations (see Table 32.1 in DPSV).

d The power of the tariff (or subsidy) is the factor by which prices are multiplied due to the tariff or subsidy. Thus a 25 per cent tariff rate corresponds to a power of the import tariff of 1.25, while a 10 per cent export subsidy corresponds to a power of the export subsidy of 0.90.

TABLE 8: THE BACKSOLUTION ORANI VARIABLES

CODE NO.	COMPUTER MNEMONIC	DESCRIPTION	NO. OF COMPONENTS
72	S1*	Powers of endogenous export subsidies	
73	XN*	Endogenous commodity exports (real)	NEX
74	XL*	Endogenous commodity imports (real)	h-NIM
75	PN	Non-agricultural commodity prices ($)	g-s
76	X3*	Household consumption of domestic commodities(real)	g
77	Y*	Investment by industry (real)	h
78	K1	Next period capital stocks by industry (real)	h
79	LI*	Employment by industry (based on wagebill weights)	h
80	P1*	Domestic prices ($ basic values)	g
81	P2*	Import prices ($ basic values)	g
82	PI*	Unit costs of capital by industry ($)	h
83	Q2*	Rental prices of capital by industry ($)	h
84	XM*	Household consumption of imports	g

* Variables marked with an asterisk must be present in the input PIEOUT file to program EXPAND.

TABLE 9: NEW SCALAR VARIABLES ADDED BY PROGRAM EXPAND TO PIEOUT[1] FILE

POSITION		DESCRIPTION	SYMBOL IN TEXT
85	*	Aggregate nominal other demands	g
86	*	Price index other demands	$\pi^{(5)}$
87	*	Price index exports(Australian dollars)	π^{e}
88	*	Price index inports(Australian dollars)	π^{m}
89	*	Price index GDP expenditure side	π^{gdp}
90	*	Nominal GDP from expenditure side	gdp
91	*	Aggregate nominal land rentals	
92	*	Aggregate nominal fixed capital rentals	r
93	*	Aggregate nominal other cost tickets	
94	*	Aggregate nominal wage payments	
95	*	Tax revenue intermediate usage	
96	*	Tax revenue investment inputs	
97	*	Tax revenue consumption	
98	*	Aggregate export subsidies	
99	*	Tariff revenue	
100	*	Total nominal indirect taxes	u
101	*	Nominal GDP from income side	gdp
102	*	Total nominal current capital stock	
103		GNP - national income	gnp
104		Ratio consumption/saving	f
105		Overall local share of local rentals	q
106		Total national saving	s
107	*	Total capital stock - rental shares	k
108		Average rental price of locally owned cap	p^{KL}
109	*	Average rental price of all local capital	p^{K}
110		Average creation price, locally owned cap	π^{L}
111		Locally owned capital stock	q+k
112		Net rentals to overseas	r_x
113		Net foreign investment	r_i
114		Rentals to overseas	r_t
115		Rentals from overseas	r_f
116	*	Total investment price index	π^{i}
117	*	Total nominal investment	i
118	*	Total real investment	i_r
119	*	Absorption price index	π_{abs}

* Variables marked by an asterisk are simply extensions of standard ORANI results. The computation of the other new variables requires information beyond that contained in the standard ORANI database. This information constitutes an additional input to program EXPAND.

[1] PIEOUT is an output file of the standard ORANI programs as implemented on CSIRONET (see Sutton (1985)).

TABLE 10: EXAMPLE OF JOB DECK FOR PROGRAM EXPAND

```
EXPAND,EC300.
*INC,USER
COMMENT.**************************************************
COMMENT.*   EXPAND                                       *
COMMENT.*   A PROGRAM TO CALCULATE P.C. CHANGES IN       *
COMMENT.*   VARIOUS MACRO QUANTITIES FROM AN ORANI       *
COMMENT.*   RESULT                                       *
COMMENT.*   INPUT   -  A PIEOUT FILE WITH ALL FINAL      *
COMMENT.*   VARIABLES AND ALL BACKSOLUTIONS EXCEPT       *
COMMENT.*   THOSE FOR 75 (PN) AND 78 (K1)                *
COMMENT.*   OUTPUT   -  A PIEOUT FILE WITH ALL THE       *
COMMENT.*   ORIGINAL INFORMATION  PLUS RESULTS FOR       *
COMMENT.*   29 NEW SCALAR (MACRO) VARIABLES              *
COMMENT.*                                                *
COMMENT.*   INSTRUCTIONS - SEE IMPACT MANUAL C7-01       *
COMMENT.*   NOS VERSION                      JMH/AS 7/85 *
COMMENT.**************************************************
FTNAUG.
RFSGET,DTBIOM.
COMMENT. ATTACH SPECIAL LONGRUN ORANI LIBRARY
ATTACH,ULIB0=O78LR4L/UN=DTBIOM.
COMMENT. ATTACH GENERAL PURPOSE ORANI LIBRARY
ATTACH,ULIB1=O78UT4L/UN=DTBIOM.
COMMENT. ATTACH ORANI LARGE CHANGE LIBRARY
ATTACH,ULIB2=O78LC4L/UN=DTBIOM.
RFSRET,DTBIOM.
LIBRARY,ULIB0,ULIB2,ULIB1/A.
COMMENT.*****************************************************
COMMENT. ATTACH NEXT A MACSHARES (INPUT) FILE PRODUCED BY PROGRAM SHARES
COMMENT. FROM THE APPROPRIATE CID FILE
ATTACH,TAPE20=XXXXXXX/UN=YYYYYY,PW=ZZZZZZ.
COMMENT. ATTACH NEXT THE INPUT C FILE - IT SHOULD MATCH YOUR INPUT TAPE30 FILE
COMMENT. IT IS USED TO GET DIMENSIONS, EXPORT ELASTICITIES, AND INDEXATION
COMMENT. PARAMETERS - SO NEED ONLY MATCH IN THESE RESPECTS
ATTACH,TAPE25=XXXXXXX/UN=YYYYYY,PW=ZZZZZZ.
COMMENT. ATTACH NEXT YOUR INPUT TAPE30 (PIEOUT) FILE
ATTACH,TAPE30=XXXXXXX/UN=YYYYYY,PW=ZZZZZZ.
COMMENT. THE FOLLOWING IS THE EXPANDED OUTPUT MACPIEOUT FILE
DEFINE,TAPE35=XXXXXXX/CT=S,PW=ZZZZZZ.
COMMENT.*****************************************************
RFL,EC=300.
LDSET,LIB=FTNAUG.
LIBLOAD,ULIB0,EXPAND.
EXECUTE,EXPAND.
*EOS
DEBUG FLAG - 1 PRINTS OUT INPUT DATA - 0 DOESNT- I5 FORMAT
    0
LONGRUN RUN CONSTANT ..LAMBDA...F10.5
0.5
*EOS
```

in the JCL, indicate the correct positions of the three input files, while the data deck contains only the value for λ, and a debug flag.

The PIEOUT file which is input to program EXPAND should have no more than 116 columns in each matrix. The output file contains the same number of columns as the input. As Table 9 shows, the new matrices present in the output MACPIEOUT files each have but one row. Thus, the MACPIEOUT file is only slightly larger than the input PIEOUT file. The PIEOUT file may be deleted once the MACPIEOUT file is created, as the MACPIEOUT file can serve any purpose to which the PIEOUT file could be applied.

The input deck allows the user to set a debug flag. If set to one the input data from the MACSHARE and C files will be printed. Whether the flag is set or not, EXPAND prints out values of all macro (scalar) variables. The original, pre-existing scalars on the input file are distinguished from newly calculated scalars. Some of the new variables have names which are very similar to those of the original ORANI variables, so users should identify these printed vectors both by name and variable/position number, using Tables 7 and 9 as a key.

EXPAND also prints out a number of coefficients used in the derivation of new macro variables. These coefficients are mentioned in Appendix III, in the section dealing with the condensed derivation of core variables.

6. PROGRAM SWAP

Program SWAP is used to change ORANI results generated by one closure into results compatible with another closure. For example, an ORANI result generated by holding the real wage constant, could be transformed into a result where aggregate employment was held constant. It is far cheaper to use program SWAP to modify an existing result in this way than to follow the alternative path of running two entirely separate parallel simulations. The theory behind the program is laid out in Horridge (1985) p. 23 and in DPSV (1982), pp. 246-247.

There are three main limitations to program SWAP:

a) Only one pair of scalar variables may exchange their exogenous/endogenous roles at a time. Thus, to move from a closure where the real wage and real consumption were exogenous to a closure where instead employment and the balance of trade were exogenous, two successive applications of SWAP would be required. The first could generate an intermediate result where the real wage was endogenous and real consumption remained exogenous. The second SWAP would then exogenize the balance of trade. The SWAPs could actually be done in either order.

b) Only single-valued (scalar) vector variables may be swapped. Examples are: 16 (aggregate imports), 23 (aggregate employment), 108 (aggregate indirect taxation). Results for all such variables are stored on a PIEOUT file as a matrix with a single row. Thus, it is impossible to use SWAP to move from a shortrun closure where all rates of return are endogenous and industry capital stocks exogenous, to a longrun closure where the reverse is true. Nor can SWAP be used to reverse the exogenous/endogenous status of the export level/export subsidy pair for a single commodity.

(c) Each column of the input PIEOUT file must contain a value for the exogenous variable which is to be endogenised. One column of the input PIEOUT file must contain a value for the endogenous variable which is to be exogenised.

Table 11 illustrates part of the contents of input and output PIEOUT files used by program SWAP. The example chosen is the transition from a closure where real consumption is exogenous and the ratio of consumption to saving endogenous to a closure where the reverse is the case. This example is pertinent to the generation of a longrun result as described in Horridge (1985).

Three matrices are shown for both the input and output file (in practice, there would be more matrices as the vertical columns of dots

TABLE 11: INPUT AND OUTPUT OF PROGRAM SWAP

INPUT FILE (1)

Matrix No.	21	77	120
Matrix Name	real consumption	industry investment	consumption/ saving
(rows,columns)	(1,3)	(112,3)	(1,3)
row 1	(0.0 , 2.0 , 0.0)	(1.9 ,-3.0 , 0.9)	(2.7 , 8.3 , 1.8)
row 2		(1.7 ,-2.1 , 2.1)	
...to..		(... , ... , ...)	
row 112		(-0.8 , 3.3 ,-0.9)	

OUTPUT FILE (2)

Matrix No.	21	77	120
Matrix Name	real consumption	industry investment	consumption/ saving
(rows,columns)	(1,3)	(112,3)	(1,3)
row 1	(-0.65, 0.77,-0.43)	(2.87,-1.16, 1.55)	(0.0 , 3.2 , 0.0)
row 2		(2.38,-0.81, 2.55)	
...to..		(... , ... , ...)	
row 112		(-1.87, 1.27,-1.61)	

Notation: (file,matrix,row,column)......thus:

(1,77,2,3) = value in input file at row 2, column 3 of matrix 77
(2,21,1,2) = value in output file at row 1, column 2 of matrix 21

(2,21,1,2) = (2,120,1,2)(1,21,1,2)/(1,120,1,2) = 3.2 x 2.0/8.3 = 0.77

(2,21,1,1) = - (1,21,1,1)(1,21,1,2)/(1,120,1,2) = -2.7 x 2.0/8.3 = -0.65
(2,21,1,3) = - (1,21,1,3)(1,21,1,2)/(1,120,1,2) = -1.8 x 2.0/8.3 = -0.43

(2,77,1,2) = (1,77,1,2)(2,120,1,2)/(1,120,1,2) = -3.0 x 3.2/8.3 = -1.16
(2,77,2,2) = (1,77,2,2)(2,120,1,2)/(1,120,1,2) = -2.1 x 3.2/8.3 = -0.81
(2,77,n,2) = (1,77,n,2)(2,120,1,2)/(1,120,1,2) = ...
....
(2,77,112,2) = (1,77,112,2)(2,120,1,2)/(1,120,1,2) = 3.3 x 3.2/8.3 = 1.27

(2,77,1,1) = (1,77,1,1) - (1,77,1,2)(1,120,1,1)/(1,120,1,2) = 1.9 - (-3.0 x 2.7/8.3) = 2.87
(2,77,n,m) = (1,77,n,m) - (1,77,n,2)(1,120,1,m)/(1,120,1,2) where n = 1,..,112,and m = 1 or 3

TABLE 12: EXAMPLE OF JOB DECK FOR PROGRAM SWAP

```
SWAP.
*INC,USER
COMMENT.**************************************************
COMMENT.*   SWAP                                         *
COMMENT.*   A PROGRAM TO SWAP AN ENDOGENOUS SCALAR       *
COMMENT.*   WITH AN EXOGENOUS SCALAR VARIABLE - TO       *
COMMENT.*   CHANGE CLOSURES                              *
COMMENT.*   INSTRUCTIONS - SEE IMPACT MANUAL C7-01       *
COMMENT.*   NOS VERSION                     JMH/AS 7/85 *
COMMENT.**************************************************
FTNAUG.
RFSGET,DTBIOM.
COMMENT. ATTACH SPECIAL LONGRUN ORANI LIBRARY
ATTACH,ULIB0=C78LR4L/UN=DTBIOM.
COMMENT. ATTACH GENERAL PURPOSE ORANI LIBRARY
ATTACH,ULIB1=C78UT4L/UN=DTBIOM.
COMMENT. ATTACH ORANI LARGE CHANGE LIBRARY
ATTACH,ULIB2=C78LC4L/UN=DTBIOM.
RFSRET,DTBIOM.
LIBRARY,ULIB0,ULIB2,ULIB1/A.
COMMENT.******************************************************
COMMENT. THE FOLLOWING IS THE INPUT FILE SHOWING RESULTS IN ORIGINAL CLOSURE
ATTACH,TAPE30=XXXXXXX/UN=YYYYYY,PW=ZZZZZZ.
COMMENT.******************************************************
COMMENT. THE FOLLOWING IS THE OUTPUT FILE CONTAINING THE NEW CLOSURE RESULTS
DEFINE,TAPE35=XXXXXXX/CT=S,PW=ZZZZZZ.
LDSET,LIB=FTNAUG.
LIBLOAD,ULIB0,SWAP.
EXECUTE,SWAP.
*EOS
SHOCK NO OF EXOGENOUS VARIABLE TO BE MADE ENDOGENOUS I5
    1
ORANI VECTOR VARIABLE NO OF EXOGENOUS VARIABLE TO BE MADE ENDOGENOUS I5
   21
ORANI VECTOR VARIABLE NO OF ENDOGENOUS VARIABLE TO BE MADE EXOGENOUS  I5
   22
VALUE OF NEWLY EXOGENOUS VARIABLE    F10.5
 1.0
ARE RESULTS TO BE CONVERTED TO ROW TOTALS OF SELECTED COLUMNS: 1=YES  0=NO
    0
MATRIX RANGE FOR FILING, PAIRS OF INTEGERS 2I5: TWO ZEROS ENDS LIST
    3    6
    9    9
   13   21
    0    0
MATRIX RANGE FOR PRINTING, PAIRS OF INTEGERS 2I5: TWO ZEROS ENDS LIST
    3    6
    9    9
   13   21
    0    0
COLUMNS TO INCLUDE IN COMPUTATIONS, PAIRS OF INTEGERS 2I5: TWO ZEROS ENDS LIST
    1   20
    0    0
*EOS
```

suggest). They contain arbitrary but logically consistent numbers. The columns of each matrix correspond to the shocks imposed on the model. In this example, there are two policy shocks corresponding to columns 1 and 3 in all matrices. The middle column of each matrix in the input file shows the response of the system to a change in the exogenous value of real consumption, while the middle column of the output file shows the results of an exogenous change in the ratio of consumption to saving.

The matrices themselves correspond to vector variables 21 (real consumption), 77 (investment by industry), and 120 (the ratio of consumption to saving). Zeros appear in columns 1 and 3 of matrix 21 in the input file and matrix 120 in the output file, showing that the input file contains the results of the policy shock where real consumption does not change, while the output file shows the results where the ratio of consumption to saving is held constant. Column 2 of matrix 21 in the input file shows that the value of the exogenous change to real consumption was 2.00. The entries in column 2 of the other matrices in this file show the response of the system to this change. Correspondingly, column 2 of the third matrix in the output file shows an exogenous change in the ratio of consumption to saving, to which all other entries in column 2 of this file are a response. The value of this exogenous change, 3.2, cannot be deduced from the input file; it is specified by the user of SWAP.

Denoting by the ordered quartuple (a,b,c,d) the value at column d and row c of matrix b in file a, we can verify the relationships set forth in Horridge (1985, p. 23). Some examples are shown at the bottom of Table 11.

Table 12 shows an example job deck for program SWAP. It requires the user to fill in input and output file names as indicated, and enter the following data:

1.SHOCK NO OF EXOGENOUS VARIABLE TO BE MADE ENDOGENOUS

Enter here the column number in the input file corresponding to the shock to be endogenized.

2.ORANI VECTOR VARIABLE NO OF EXOGENOUS VARIABLE TO BE MADE ENDOGENOUS I5

Enter here the matrix or position number of the originally exogenous variable which is to become endogenous. A difficulty arises if the column in item (1) corresponds, not to a change in a single exogenous variable, but is instead the combined effect of a number of such changes. It could,for instance, represent the total effect of specified reductions in a range of tariffs. We may wish to adjust the magnitude of all the changes together in such a way that, say, the balance of trade is held constant.

SWAP allows the desired effect to be achieved by entering a zero in this section of the data deck. A new 'dummy' composite variable is then appended to the output file, and is given the original exogenous value of unity. After SWAPping, this final position shows which multiple of the original composite shock is required to satisfy the new exogenous restriction.

3.ORANI VECTOR VARIABLE NO OF ENDOGENOUS VARIABLE TO BE MADE EXOGENOUS I5

Enter here the matrix or position number of the originally endogenous variable which is to become exogenous. Problems akin to that mentioned above may arise: one may wish to exogenize a variable which is not part of ORANI, or is an element of a vector. Here, the solution is to previously add the value oneself to a new position in the input file, using the FIEDIT program described in the next section.

4.VALUE OF NEWLY EXOGENOUS VARIABLE F10.5

Enter here the new, exogenous, value of the originally endogenous variable which is to become exogenous. If in doubt, enter 1.0, as this makes for simpler checking of the results. The results of this unit shock will appear in the output file in the column corresponding to that mentioned in item 1, as altered by the mapping specified in item 7.

5.ARE RESULTS TO BE CONVERTED TO ROW TOTALS OF SELECTED COLUMNS: 1=YES 0=NO

You get the row totals anyhow in the printed output, so only enter 1 to eliminate the individual columns, or if you want row totals in the output file.

The final three items correspond to a range selection format which is described more fully in the next section.

6.MATRIX RANGE FOR FILING, PAIRS OF INTEGERS 2I5: TWO ZEROS ENDS LIST

Allows you to avoid filing bulky information you won't need. You have to choose at least one variable for filing or the program will abort.

6.MATRIX RANGE FOR PRINTING, PAIRS OF INTEGERS 2I5: TWO ZEROS ENDS LIST

Allows you to save paper by not printing bulky information you won't need. It is wise to include here the matrices mentioned in items 2 and 3 above, as this facilitates checking. You have to choose at least one variable for printing or the program will abort.

8.COLUMNS TO INCLUDE IN COMPUTATIONS, PAIRS OF INTEGERS 2I5: TWO ZEROS ENDS LIST

Printing, filing, and row totalling apply only to the selected columns. Note that your column selection interacts with items 4 and 5.

7. PROGRAMS TO MANIPULATE SOLUTION FILES

This is a group of programs which take as input one or more PIEOUT files, combine or manipulate their contents, and produce a result file in the same PIEOUT format. Figure 3 illustrates their operation. Each program performs a fairly primitive operation, but as the output of one may serve as the input of another, the programs may be combined to perform more complex tasks. In brief:

ROWTOT outputs the row totals of all matrices in the input file, taking into account only selected columns.

APPEND combines two input files. Matrices at corresponding positions in the input file appear at the same position in the output file joined horizontally edge to edge. Thus the number of columns in the output file is the sum of the numbers of columns in the two input files.

LINCOM adds, or more generally forms a linear combination of, two input files. The user supplies coefficients α and β; each matrix in the output file is equal to $(\alpha.M1 + \beta.M2)$ where M1 and M2 are the matrices at the same position in the input files.

PIEDIT allows the addition or changing of a matrix in a PIEOUT file.

Example job decks of the four general programs are shown in Tables 13 to 16. User specific items in the JCL are underlined. Certain features are common to all four.

(a) Input and output files are specified in the JCL. Rename these to suit (limit 7 letters), but <u>don't</u> change the unit numbers (TAPEnn etc).

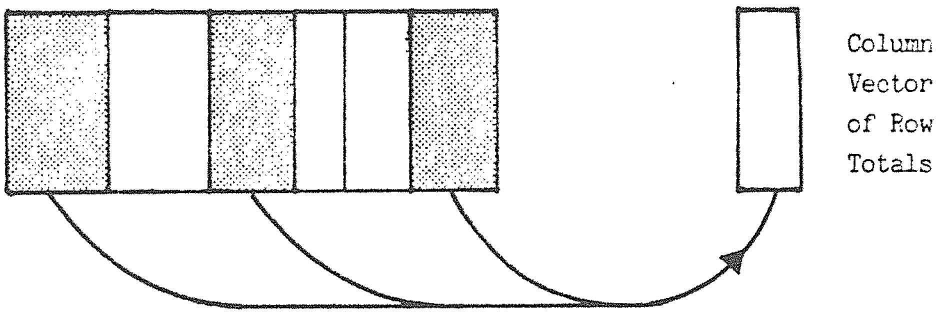
(b) The instruction cards in the data deck must appear in the positions shown, although their contents are only a guide to the user.

(c) Number formats are described on these cards according to FORTRAN conventions, as briefly described in Section 3.

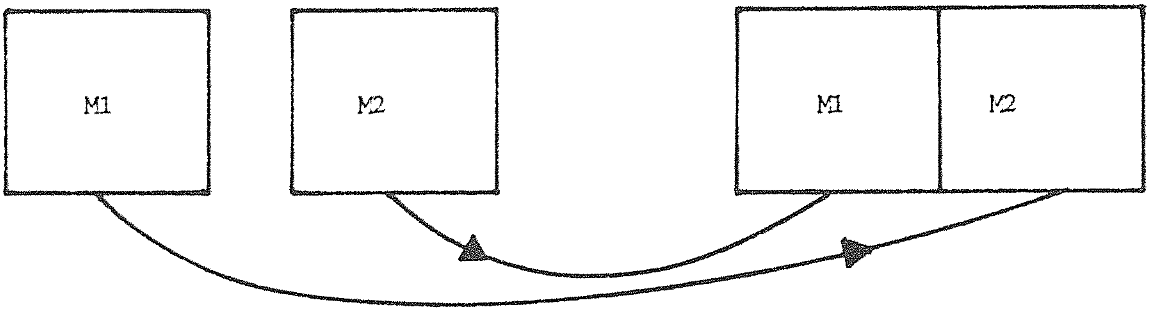
(d) The programs expect the input files to obey certain rules. For example, program LINCOM forms a linear combination of the matrices at corresponding positions in two input files and writes them to a third. Naturally LINCOM expects that matrices to be added together have the same dimensions. This may not always be the case. Generally, the programs do not abort in such cases; rather they resolve the conflict in the most obvious way (as described below) at the same time printing out a special warning message. In any event the dimensions and name of each matrix input or output is recorded in the printed output. This output should be carefully examined to check that the desired result is being achieved. Programs WRTMAT and FLASH

FIGURE 3: OVERVIEW OF PROGRAMS TO MANIPULATE SOLUTION FILES

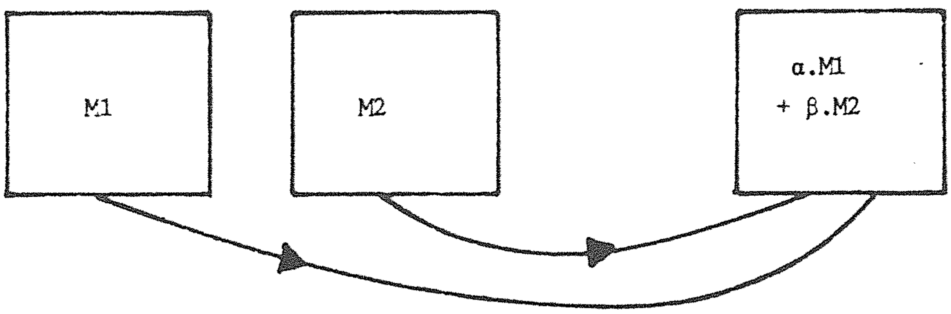
INPUT | OUTPUT



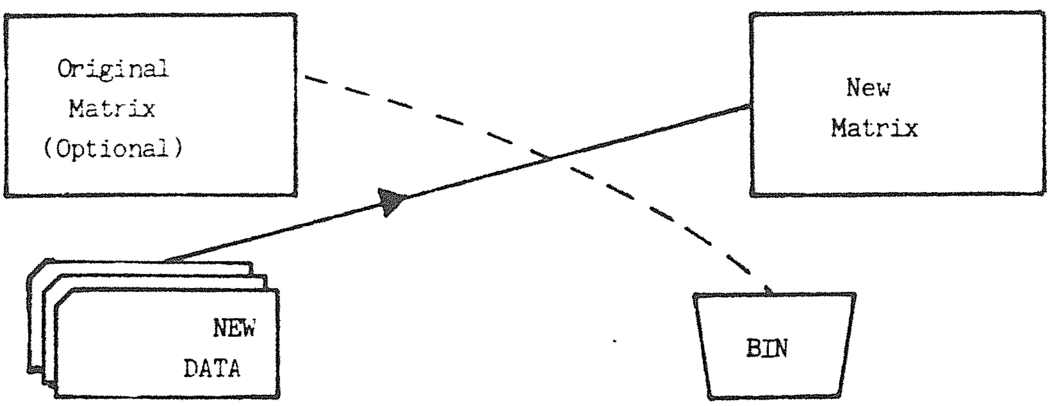
Program ROWTOT gets row totals of selected columns.



Program APPEND joins corresponding matrices side-by-side.



Program LINCOM forms a linear combination of corresponding matrices.



Program PIEDIT allows replacement or insertion of matrices.

may be used to check selected items from any files produced as intermediate stages.

(e) Matrix/Column range cards list a series of pairs of integers, specifying which matrices (or columns) are to be processed, terminated by a pair of zeros. The same range applies to all input and output files. Thus the cards:

```
MATRIX RANGE, PAIRS OF INTEGERS 2I5: TWO ZEROS ENDS LIST
    3    7
   20   25
    0    0
```

mean that only any matrices at positions 3, 4, 5, 6, 7, 20, 21, 22, 23, 24, and 25 will be processed. If a selected matrix is absent from the input file, a warning is issued, but processing continues. If a matrix is included twice, a message is issued, but the matrix will only be processed once. The order of the integer pairs is immaterial, except that the pair of zeros must terminate the list. The order of the numbers within each pair is immaterial, although a warning appears if they are reversed. If either is zero or negative the pair is interpreted as a pair of zeros (end of list). If nothing is selected the program will abort with a message. Requests for matrices the position numbers of which exceed the length of the input file are simply ignored. Thus, for the lazy, the following sequence will select all positions for processing:

```
MATRIX RANGE
    1  999
    0    0
```

In all cases where choice of matrix range is necessary, the program prints out a list of the selected matrices.

Processing of columns happens in much the same way. Note that the selection list is always internally sorted in ascending order. This order (printed out at the beginning) is also the order in which the columns appear in any output printing or filing.

(f) All the programs expect that matrices within any one file have the same number of columns. If this is not the case, a warning is issued, but processing continues. The current 'correct' number of columns in a file is equal to the number of columns in the last matrix to be processed. Thus the warning appears whenever the number of columns changes. Matrices are always processed in ascending order.

Specific instructions for the four programs, together with sample applications, appear below.

7.1 Program ROWTOT

Program ROWTOT adds up the row totals (using only selected columns) of chosen matrices present in an input file, filing the resulting column vectors with the same position and NAME as the input matrix. An example job deck appears as Table 13. The cards governing the selection of the columns operate in just the same manner as the MATRIX RANGE cards described above. In particular, if a requested column does not exist in the matrix currently being processed, a warning is printed out, although the program does not abort. The output consists of column vectors of the same depth as the corresponding matrices in the input files. Element (k) of such a vector is the sum of all elements in the kth row of the corresponding input matrix, which were in selected and existing columns.

EXAMPLES:

(I) A PIEOUT file showing the individual effects of 60 tariff changes could be processed by ROWTOT four times to produce PIEOUT files showing:

(a) the effect of changes in the tariffs protecting the T.C.F. sector;

(b) the effect of changes in the tariffs on the motor vehicles sector;

(c) the effect of all other tariffs.

These could be APPENDed to form one file with three columns. The computations load of subsequent processing (including interpretation) is reduced by a factor of up to 20 times.

(II) PIEOUT files which have been swapped typically contain a column showing the effects of an increase in the previously endogenous variable which has now become exogenous. Other columns show the effect of policy shocks under the new closure. We may wish to compute the total effect of the policy shocks, without including the effect of increasing the newly exogenous variable. ROWTOT provides a means of doing this.

7.2 Program APPEND

Table 14 shows an example job deck for program APPEND. The only data required are the specifications of the two input files and the output file, plus the matrix range. APPEND joins corresponding matrices in the two input files (of width W1 and W2 columns) at their edges to produce a single matrix (with W1+W2 columns). The first mentioned of the input files occupies the left hand side of the output file. Special cases are resolved as follows:

TABLE 13: EXAMPLE OF JOB DECK FOR PROGRAM ROWTOT

```
ROWTOT.
*INC,USER
COMMENT.*************************************************
COMMENT.*   ROWTOT                                      *
COMMENT.*   A PROGRAM TO READ A PIEOUT FILE,            *
COMMENT.*   COMPUTE THE ROW TOTALS OF ITS MATRICES,     *
COMMENT.*   AND WRITE THEM TO A NEW FILE.               *
COMMENT.*   INSTRUCTIONS - SEE IMPACT MANUAL C7-01      *
COMMENT.*   NOS VERSION                     JMH/AS 7/85 *
COMMENT.*************************************************
FTNAUG.
RFSGET,DTBIOM.
COMMENT. ATTACH SPECIAL LONGRUN ORANI LIBRARY
ATTACH,ULIB0=078LR4L/UN=DTBIOM.
COMMENT. ATTACH GENERAL PURPOSE ORANI LIBRARY
ATTACH,ULIB1=078UT4L/UN=DTBIOM.
COMMENT. ATTACH ORANI LARGE CHANGE LIBRARY
ATTACH,ULIB2=078LC4L/UN=DTBIOM.
RFSRET,DTBIOM.
LIBRARY,ULIB0,ULIB2,ULIB1/A.
COMMENT. THE FOLLOWING IS THE INPUT FILE
ATTACH,TAPE30=XXXXXXX/UN=YYYYYY,PW=ZZZZZZ.
COMMENT. THE FOLLOWING IS THE OUTPUT FILE
DEFINE,TAPE35=XXXXXXX/CT=S,PW=ZZZZZZ.
LDSET,LIB=FTNAUG.
LIBLOAD,ULIB0,ROWTOT.
EXECUTE,ROWTOT.
*EOS
MATRIX RANGE...PAIRS OF INTEGERS 2I5: A PAIR OF ZEROS TERMINATES THE LIST
    3    6
    9    9
   13   21
    0    0
COLUMN RANGE...PAIRS OF INTEGERS 2I5: A PAIR OF ZEROS TERMINATES THE LIST
    1  999
    0    0
*EOS
```

TABLE 14: EXAMPLE OF JOB DECK FOR PROGRAM APPEND

```
APPEND.
*INC,USER
COMMENT.*****************************************************
COMMENT.*   APPEND                                          *
COMMENT.*   A PROGRAM TO READ 2 FIEOUT FILES                *
COMMENT.*   JOIN CORRESPONDING MATRICES SIDE BY SIDE        *
COMMENT.*   AND WRITE EACH NEW MATRIX TO A NEW FILE         *
COMMENT.*   INSTRUCTIONS - SEE IMPACT MANUAL C7-01          *
COMMENT.*   NOS VERSION                          JMH/AS 7/85 *
COMMENT.*****************************************************
FTNAUG.
RFSGET,DTBIOM.
COMMENT. ATTACH SPECIAL LONGRUN ORANI LIBRARY
ATTACH,ULIB0=078LR4L/UN=DTBIOM.
COMMENT. ATTACH GENERAL PURPOSE ORANI LIBRARY
ATTACH,ULIB1=078UT4L/UN=DTBIOM.
COMMENT. ATTACH ORANI LARGE CHANGE LIBRARY
ATTACH,ULIB2=078LC4L/UN=DTBIOM.
RFSRET,DTBIOM.
LIBRARY,ULIB0,ULIB2,ULIB1/A.
COMMENT.*****************************************************
COMMENT. THE FOLLOWING IS THE FIRST INPUT FILE
COMMENT. IT GIVES THE LEFTHAND PART OF OUTPUT MATRICES
ATTACH,TAPE30=XXXXXXX/UN=YYYYYY,PW=ZZZZZZ.
COMMENT.*****************************************************
COMMENT. THE FOLLOWING IS THE SECOND INPUT FILE
COMMENT. IT GIVES THE RIGHTHAND PART OF OUTPUT MATRICES
ATTACH,TAPE31=XXXXXXX/UN=YYYYYY,PW=ZZZZZZ.
COMMENT.*****************************************************
COMMENT. THE FOLLOWING IS THE OUTPUT FILE CONTAINING THE JOINED MATRICES
DEFINE,TAPE35=XXXXXXX/CT=S,PW=ZZZZZZ.
LDSET,LIB=FTNAUG.
LIBLOAD,ULIB0,APPEND.
EXECUTE,APPEND.
*EOS
MATRIX RANGE...PAIRS OF INTEGERS 2I5: A PAIR OF ZEROS TERMINATES THE LIST
    3    6
    9    9
   13   21
    0    0
*EOS
```

(a) The output matrix in any position has a number of rows equal to the maximum of the two input matrices. The input matrix with least rows is padded at the bottom with zeros before joining. A warning is printed in this case.

(b) If both input files are empty at a particular position, so also will be the output file.

(c) If only one file is empty at some position, APPEND creates a dummy matrix of zeros to join to the existing matrix in the other, full, file. This dummy matrix has the same number of rows as its existing counterpart, and the same number of columns as the last existing matrix to be processed in the same input file. Note that matrices are processed in ascending order, and APPEND will only be aware of matrices within the matrix range chosen. Thus it is well to ensure that the first position within the matrix range is indeed occupied on both input files.

(d) If the number of columns in a matrix is found to differ from that found in the last matrix on that file to be processed, a warning is printed.

(e) If only one matrix is present at a given position in the input files, the output matrix bears the same NAME. If both input files contain a matrix at this position, the output file bears the name of the second file.

(f) The width of the output matrix is limited to 116 columns. Where the summed width of the input columns exceeds 116, only the leftmost 116 columns will be filed. A warning of this truncation is printed.

EXAMPLE: Several PIEOUT files from the standard ORANI programs, representing the effects of different shocks, may be APPENDed together and processed as one by the EXPAND or SWAP programs. The total effect of some or all of the shocks could then be found by ROWTOT.

7.3 Program LINCOM

LINCOM combines corresponding matrices from two input files (M1, M2), and writes a linear combination of the two to a third file (α.M1 + β.M2). The example deck in Table 15 is fairly self explanatory, bearing in mind the comments above applying to all four file manipulation programs. Special features of LINCOM are:

(a) If the corresponding matrices in the two input files are of different dimensions, a warning is printed. If necessary, the narrower matrix is padded with zeros at the right to the larger of the two widths. Similarly, the shallower matrix is padded with zeros at the bottom, so that the two matrices have the same depth. Again, a warning is printed. The linear combination thus has the smallest dimensions which can accommodate both input matrices.

(b) If only one input file has a matrix present at a given position, a dummy matrix of zeros is assigned to the empty position on the other file, of the same dimensions as its counterpart. A linear combination is formed of these two. A warning is issued.

EXAMPLE: LINCOM may be used to find the results of a two-instrument two-target problem. Each input file consists of one column showing the results of each instrument. Solving by hand the two simultaneous equations to find the amount of each instrument required to hit the two targets determines α and β. LINCOM now finds the effect on all endogenous variables of the combination policy.

7.4 Program PIEDIT

An example PIEDIT job deck is shown in Table 16. PIEDIT adds a matrix to a PIEOUT file or changes an existing matrix. If an existing matrix is to be changed, all its details must be respecified (PIEDIT will not change part of a matrix). However, PIEDIT can add or alter more than one matrix at a time.

The required input data are:

(a) The number of matrices to be altered, N (>0).
The succeeding block of cards must appear <u>in toto</u> N times.

(b) The position of the (next) new matrix. It is possible to enter a negative number here. In that case the matrix at the corresponding

TABLE 15: EXAMPLE OF JOB DECK FOR PROGRAM LINCOM

```
LINCOM.
*INC,USER
COMMENT.****************************************************
COMMENT.*   LINCOM                                         *
COMMENT.*   A PROGRAM TO READ 2 PIEOUT FILES               *
COMMENT.*   AND CREATE A NEW FILE CONTAINING               *
COMMENT.*   A LINEAR COMBINATION OF                        *
COMMENT.*   CORRESPONDING MATRICES.                        *
COMMENT.*   FACTORS ALPHA AND BETA ARE IN DATA DECK        *
COMMENT.*   FIRST INPUT FILE       M1                      *
COMMENT.*   SECOND INPUT FILE      M2                      *
COMMENT.*   OUTPUT FILE            ALPHA*M1 + BETA*M2      *
COMMENT.*   INSTRUCTIONS - SEE IMPACT MANUAL C7-01         *
COMMENT.*   NOS VERSION                        JMH/AS 7/85 *
COMMENT.****************************************************
FTNAUG.
RFSGET,DTBIOM.
COMMENT. ATTACH SPECIAL LONGRUN ORANI LIBRARY
ATTACH,ULIB0=078LR4L/UN=DTBIOM.
COMMENT. ATTACH GENERAL PURPOSE ORANI LIBRARY
ATTACH,ULIB1=078UT4L/UN=DTBIOM.
COMMENT. ATTACH ORANI LARGE CHANGE LIBRARY
ATTACH,ULIB2=078LC4L/UN=DTBIOM.
RFSRET,DTBIOM.
LIBRARY,ULIB0,ULIB2,ULIB1/A.
COMMENT.*******************************************************
COMMENT. THE FOLLOWING IS THE FIRST INPUT FILE
COMMENT. IT WILL BE MULTIPLIED BY ALPHA
ATTACH,TAPE30=XXXXXXX/UN=YYYYYY,PW=ZZZZZZ.
COMMENT.*******************************************************
COMMENT. THE FOLLOWING IS THE SECOND INPUT FILE
COMMENT. IT WILL BE MULTIPLIED BY BETA
ATTACH,TAPE31=XXXXXXX/UN=YYYYYY,PW=ZZZZZZ.
COMMENT.*******************************************************
COMMENT. THE FOLLOWING IS THE OUTPUT FILE CONTAINING THE LINEAR COMBINATION
DEFINE,TAPE35=XXXXXXX/CT=S,PW=ZZZZZZ.
LDSET,LIB=FTNAUG.
LIBLOAD,ULIB0,LINCOM.
EXECUTE,LINCOM.
*EOS
MATRIX RANGE...PAIRS OF INTEGERS 2I5: A PAIR OF ZEROS TERMINATES THE LIST
    3    6
    9    9
   13   21
    0    0
ALPHA...MULTIPLICATION FACTOR FOR FIRST FILE....FORMAT F10.5
       2.2
BETA....MULTIPLICATION FACTOR FOR SECOND FILE...FORMAT F10.5
      -3.1
*EOS
```

TABLE 16: EXAMPLE OF JOB DECK FOR PROGRAM PIEDIT

```
PIEDIT.
*INC,USER
COMMENT.*************************************************
COMMENT.*   PIEDIT                                      *
COMMENT.*   A PROGRAM TO ADD A MATRIX TO A PIEOUT       *
COMMENT.*   FILE OR REPLACE AN EXISTING MATRIX          *
COMMENT.*   CREATING A NEW MODIFIED FILE                *
COMMENT.*   INSTRUCTIONS - SEE IMPACT MANUAL C7-01      *
COMMENT.*   NOS VERSION                     JMH/AS 7/85 *
COMMENT.*************************************************
FTNAUG.
RFSGET,DTEIOM.
COMMENT. ATTACH SPECIAL LONGRUN ORANI LIBRARY
ATTACH,ULIB0=C78LR4L/UN=DTEIOM.
COMMENT. ATTACH GENERAL PURPOSE ORANI LIBRARY
ATTACH,ULIB1=078UT4L/UN=DTEIOM.
COMMENT. ATTACH ORANI LARGE CHANGE LIBRARY
ATTACH,ULIB2=C78LC4L/UN=DTEIOM.
RFSRET,DTEIOM.
LIBRARY,ULIB0,ULIB2,ULIB1/A.
COMMENT.*****************************************************
COMMENT. THE FOLLOWING IS THE INPUT FILE - IT WONT BE CHANGED
ATTACH,TAPE30=XXXXXXX/UN=YYYYYY,PW=ZZZZZZ.
COMMENT.*****************************************************
COMMENT. THE FOLLOWING IS THE OUTPUT FILE INCORPORATING ANY CHANGES
DEFINE,TAPE35=XXXXXXX/CT=S,PW=ZZZZZZ.
LDSET,LIB=FTNAUG.
LIBLOAD,ULIB0,PIEDIT.
EXECUTE,PIEDIT.
*EOS
NUMBER OF MATRICES TO BE ADDED OR ALTERED.....I5 FORMAT
    2
POSITION OF THE FIRST NEW MATRIX......I5 FORMAT
  121
NAME OF THE FIRST NEW MATRIX..40 CHARACTERS MAX
LAND TAX REVENUE
NUMBER OF ROWS OF NEW MATRIX.........I5 FORMAT
    1
NUMBER OF COLUMNS OF NEW MATRIX......I5 FORMAT
   11
CONTENTS OF MATRIX...8F10.5 FORMAT...START A NEW LINE FOR EACH ROW
       4.7      -9.3       6.2       0.5       6.4      -9.4      -5.1      5.78
       1.3      -3.3       2.3
POSITION OF MATRIX TO BE DELETED.....I5 FORMAT(NOTE,OTHER DETAILS ARE SKIPPED)
 -122
POSITION OF THE NEXT NEW MATRIX......I5 FORMAT
  123
NAME OF THE NEXT NEW MATRIX..40 CHARACTERS MAX
STATE TAX REVENUE
NUMBER OF ROWS OF NEW MATRIX.........I5 FORMAT
    6
NUMBER OF COLUMNS OF NEW MATRIX......I5 FORMAT
   11
CONTENTS OF MATRIX...8F10.5 FORMAT...START A NEW LINE FOR EACH ROW
       4.7      -9.3       6.2       0.5       6.4      -9.4      -5.1      5.78
       1.3      -3.3       2.3
       8.7      -9.3       6.2       0.5       6.8      -9.8      -5.1      5.78
       1.3      -3.3       2.3
       8.7      -9.9       6.2       0.6       2.8      -9.8      -5.4      5.76
       1.9      -2.4       2.9
       8.7      -4.9       2.2       0.5       2.3      -9.8      -5.1      5.78
       1.9      -5.3       2.9
       8.7      -6.9       2.2       0.7       2.8      -9.8      -5.2      5.78
       1.9      -7.5       2.9
       8.7      -7.9       2.2       0.5       2.8      -9.8      -5.1      5.98
       1.9      -9.9       2.9
```

(positive) position will not be copied across to the output file. This corresponds to deleting that position. If this option is chosen, cards corresponding to steps (c) to (f) must be omitted, as shown in the example deck.

(c) The name of the (next) new matrix - 40 characters maximum.

(d) The number of rows it will have.

(e) The number of columns it will have.

(f) The contents of the matrix in 8F10.5 format. Each row of the matrix must be started on a new line.

(g) Repeat entries (b) - (f) N-1 times.

The length of the INDEX vector in the output file will be 150, so no matrix can have a position number greater than 150.

It is envisaged that PIEDIT will be used to edit PIEOUT files, which are composed merely of real matrices. Nevertheless, some provision has been made for random access files which contain integer matrices. The program will copy these across unchanged from the input file, unless requested to delete them. However, PIEDIT cannot be used to enter new integer data.

EXAMPLE: PIEDIT can be combined with Program SWAP to produce solutions for a respecified ORANI model. Suppose it is desired to simulate a shortrun environment where Y (aggregate real private investment), instead of being exogenous, is tied to an overall profitability index, P. Then in percentage change form: y = p + fp, where FP is a shift variable. First run a normal ORANI experiment and print out the effects of (a) the policy shock and (b) a one per cent increase in Y. Next calculate by hand the implied changes (with respect to both shocks) in Y and FP. PIEDIT can be used to enter these values into spare positions on the PIEOUT solution file. Finally use SWAP to produce a new solution file, swapping Y with FP. This will show the results of the policy shock when fp is exogenous and zero, with Y endogenous.

8. PROGRAMS TO PRINT SOLUTION FILES

8.1 Program FLASH

Program FLASH is provided to print out TAPE30 or PIEOUT type files. It produces similar results to the WRTMAT program (see Sutton(1985)), but is improved in several ways:

(i) Labels are provided for the rows of matrices.

(ii) It accepts up to 150 matrices on a file.

(iii) It incorporates dynamic formatting so that large and small numbers can be printed out to good accuracy.

(iv) No unnecessary row totals for column vectors, or column totals for row vectors.

(v) Careful use of space gives up to 58 lines of numbers per page. This enables the 115 (114 + col., totals) rows of a commodity vector to fit neatly onto two pages.

Table 17 shows an example job deck for program FLASH. The user attaches a PIEOUT file, and a PARAMS file of matching dimensions, used to provide row labels. The program reads labels for industries, commodities, and occupations off the PARAMS file. As each vector is printed, its rows are labelled according as to whether its dimensions match any of these three.

Three items are also required in the data deck. Each is preceded by a dummy line of instructions, which must not be removed.

First, NDEC, a number between 1 and 6, shows the desired number of figures after the decimal point for the printed values. Where possible, these will be printed in the format F8.NDEC. If any value is too large for this to be possible, NDEC is temporarily decreased until the value can be printed successfully. If necessary, an exponential format is adopted. Unlike some other ORANI print programs, the adjustment to NDEC applies only to one value, not to the rest of the row in which it appears.

Second, a title (maximum length 80 characters) must be supplied, to appear at the top of each page.

Third, the user must supply cards indicating the selection of matrices to be printed. These cards follow the rules set forth in Section 7 above.

Program FLASH numbers the columns of printed matrices but does not supply column labels. It is up to the user to remember to which shock any column corresponds. FLASH may be used to print out any ORANI Random Access file, although it is tailored to suit PIEOUT files. It ignores requests to print integer matrices, and prints row vectors without labels, and in a less compact format.

TABLE 17: EXAMPLE OF JOB DECK FOR PROGRAM FLASH

```
FLASH.
*INC,USER
COMMENT.
COMMENT.***********************************************
COMMENT.*   FLASH                                     *
COMMENT.*   A PROGRAM TO PRINT OUT A RANDOM ACCESS    *
COMMENT.*   FILE WITH LABELS                          *
COMMENT.*                                             *
COMMENT.*   INSTRUCTIONS - SEE IMPACT MANUAL C7-01    *
COMMENT.*   NOS VERSION                JMH/AS 10/85   *
COMMENT.***********************************************
FTNAUG.
RFSGET,DTBIOM.
COMMENT. ATTACH SPECIAL LONGRUN ORANI LIBRARY
ATTACH,ULIB0=078LR4L/UN=DTBIOM.
COMMENT. ATTACH GENERAL PURPOSE ORANI LIBRARY
ATTACH,ULIB1=078UT4L/UN=DTBIOM.
COMMENT. ATTACH ORANI LARGE CHANGE LIBRARY
ATTACH,ULIB2=078LC4L/UN=DTBIOM.
RFSRET,DTBIOM.
LIBRARY,ULIB0,ULIB2,ULIB1/A.
COMMENT. ATTACH HERE THE APPROPRIATE PARAMS FILE
ATTACH,TAPE1=JS78114PARAMS/UN=DTBGJS,PW=LC.
COMMENT. ATTACH HERE THE INPUT PIEOUT FILE OF RESULTS
ATTACH,TAPE2=FILENAME/UN=MYNAME,PW=SECRET.
LDSET,LIB=FTNAUG.
LIBLOAD,ULIB0,FLASH.
EXECUTE,FLASH.
*EOS
NDEC = MAXIMUM NO OF DECIMAL PLACES (I5)
    4
EXPERIMENT TITLE (8A10)
EXAMPLE PRINTOUT FOR FLASH
MATRIX RANGE...PAIRS OF INTEGERS 2I5: A PAIR OF ZEROS TERMINATES THE LIST
    1    1
   12   32
    0    0
*EOS
*EOP
```

8.2 Program AGGREG

AGGREG reads in a PIEOUT file of results, and aggregates the rows of selected matrices, according to specified weights. The aggregated matrices are printed out with row labels in the same format as FLASH. Thus, an ORANI result reported at the 112 industry level may be presented as a series of sectoral results.

An example job deck appears as Table 18. The user must attach three files. A CID file is used to compute the weights, while the PARAMS file supplies the labels. The input PIEOUT file must match the dimensions of these.

The data deck falls into three parts. First, the user sets the debug flag, the desired number of decimal places in the printout, and a title to appear at the top of each page.

Next the details of the aggregation are specified. Here the details must be entered in a nested fashion. Firstly, the number of new sectors, NGROUP is entered. Secondly, and repeated NGROUP times, enter the name of each new sector, the number of individual industries within it, and the INDUSTRY numbers of the old ORANI industries to which it corresponds.

Lastly, the user specifies a series of tasks, each defined by an integer pair. The first integer is the ORANI variable number (see Tables 7 and 8) of the matrix to be aggregated. The second number assigns the weighting vector to be used, according to the list below.

OPTION NO	DESCRIPTION
1	LAND
2	LABOUR (WAGES)
3	LABOUR (PERSONS)
4	LABOUR (HOURS)
5	FIXED CAPITAL RENTALS
6	FIXED CAPITAL STOCKS
7	TOTAL FACTOR PAYMENTS
8	OTHER COST TICKETS
9	VALUE ADDED = TOTAL FACTOR + O.C.T
10	INTERMEDIATE USAGE BY INDUSTRY (P.PRICES)
11	OUTPUT = INDUSTRY COSTS
12	INVESTMENT BY INDUSTRY (PURCHASERS PRICES)
13	EXPORTS (BASIC VALUES)
14	IMPORTS (BASIC VALUES)
15	IMPORTS (FOREIGN PRICES)
16	DOMESTIC CONSUMPTION (PURCHASERS PRICES)
17	IMPORTED CONSUMPTION (PURCHASERS PRICES)

A pair of zeros signals the end of the task list.

TABLE 18: EXAMPLE OF JOB DECK FOR PROGRAM AGGREG

```
AGGREG.
*INC,USER
COMMENT.*************************************************
COMMENT.*   AGGREG                                      *
COMMENT.*   A PROGRAM TO AGGREGATE INDUSTRY             *
COMMENT.*   RESULTS FROM A PIEOUT FILE                  *
COMMENT.*                                               *
COMMENT.*   INSTRUCTIONS - SEE IMPACT MANUAL C7-01      *
COMMENT.*   NOS VERSION                    JMH/AS 10/85 *
COMMENT.*************************************************
COMMENT.
FTNAUG.
RFSGET,DTBIOM.
COMMENT. ATTACH SPECIAL LONGRUN ORANI LIBRARY
ATTACH,ULIB0=O78LR4L/UN=DTBIOM.
COMMENT. ATTACH GENERAL PURPOSE ORANI LIBRARY
ATTACH,ULIB1=O78UT4L/UN=DTBIOM.
COMMENT. ATTACH ORANI LARGE CHANGE LIBRARY
ATTACH,ULIB2=O78LC4L/UN=DTBIOM.
RFSRET,DTBIOM.
LIBRARY,ULIB0,ULIB2,ULIB1/A.
COMMENT. ATTACH HERE THE APPROPRIATE CID FILE
ATTACH,TAPE1=JS78114CID/UN=DTBGJS,PW=LC.
COMMENT. ATTACH HERE THE APPROPRIATE PARAMS FILE
ATTACH,TAPE2=JS78114PARAMS/UN=DTBGJS,PW=LC.
COMMENT. ATTACH HERE THE INPUT PIEOUT FILE OF RESULTS
ATTACH,TAPE3=FILENAME/UN=DTBGAS,PW=SECRET.
LDSET,LIB=FTNAUG.
LIBLOAD,ULIB0,AGGREG.
EXECUTE,AGGREG.
*EOS
DEBUG FLAG 1 = DEBUG OUTPUT 0 = NONE(I5)
    1
NDEC = MAXIMUM NO OF DECIMAL PLACES (I5)
    4
EXPERIMENT TITLE FOR PAGE HEADING - 80 CHARACTERS MAX
EXAMPLE JOB FOR AGGREG USING BISSA01
NGROUP - THE NUMBER OF NEW CATEGORIES - I5 FORMAT
    6
FOR EACH CATEGORY GIVE NAME (3A10), NO OF MEMBERS (I5), MEMBER NOS (16I5
AGRICULTURE
   11
    1    2    3    4    5    6    7    8    9   10   11
MINING
    6
   12   13   14   15   16   17
OTHER FOOD PRODUCTS
    4
   23   24   25   26
T. C. F.
    9
   31   32   33   34   35   36   37   38   39
MINERAL PRODUCTS
   11
   57   58   59   60   61   62   63   64   65   66   67
TRANSPORT EQUIPMENT
    4
   68   69   70   71
TASK DETAILS - ONE LINE PER TASK  VECTOR POSITION,WEIGHT OPTION NUMBER(2I5)
    1    9
    1   11
   20    1
   77   12
   79    4
   76   16
    0    0
*EOS
*EOP
```

A feature of the aggregation facility is that the user-supplied grouping need neither be exhaustive nor exclusive. Thus there is no need for every industry to be included in a new sector, and the newly defined sectors may overlap, or be subsets of one another. Results for a single industry may be presented alongside results for a wider grouping of which it is a member. Because so wide a range of groupings is allowed, errors in the cards which specify the grouping may pass unnoticed. Users are therefore advised to set the debug flag to 1 for the first run using any new grouping. This causes the program to print a list of the new sectors, showing which of the original industries are included in them.

Any ORANI variable which is either of industry or commodity dimensions may be selected for aggregation. The weight vectors are also of industry or commodity dimensions. However, the user-specified grouping is always on an industry basis. If any task specifies a variable or a weight vector which is of commodity dimensions AGGREG transforms the weight or variable to industry dimensions by using the MAKE matrix of industry outputs. Thus any of the weights may be used with any ORANI variable, whether it is of industry or commodity dimensions. Commonly the multiproduct (agricultural) industries will be grouped together as one so that this procedure introduces no problems. Otherwise, results for partly agricultural new sectors should be interpreted with caution. Of course, only a small proportion of the large number of possible tasks is economically sensible. The program is constructed to facilitate the addition of new weighting vectors to the option list, if this were desired. More information about the weights, and the database flows from which they were constructed, will be printed out if the debug flag is set to an integer larger than 1.

One application of AGGREG would be to compare the AGGREGated results of a simulation using fullsize ORANI, with the results from a smaller version of ORANI shrunk down with the IAC's AG78 facility. So far as the author is aware, this interesting experiment has not so far been performed.

REFERENCES

Bruce, I. and J. Sutton (1984), Disaggregation of Commodities and Industries in ORANI78. IMPACT Research Memorandum No. OA-226, ORANI Module, March, Canberra.

Bruce, I.A., J.M. Sutton and M. Da Cruz (1984), ORANI78 Large Change Aggregation Facility: Computing Manual. IMPACT Research Memorandum No. CA-72, Computer System, August, Canberra.

Dixon, P.B., B.R. Parmenter, J. Sutton and D.P. Vincent (1982), ORANI : A Multisectoral Model of the Australian Economy, North-Holland Publishing Company, Amsterdam, pp. xvii+372.

Horridge, Mark (1985), "Long-Run Closure of ORANI: First Implementation", IMPACT Preliminary Working Paper No. OP-50.

Sutton, John (1985) "ORANI 78 Large Change Computing Manual", (Revised) IMPACT Research Memorandum No. CA-79, Computer System, March, Canberra.

APPENDIX I

Compatibility with Different Sizes of ORANI

The Industries Assistance Commission maintains a program AG78 which may be used to reduce the number of industries and commodities in an ORANI database (Bruce, Sutton and Da Cruz 1984). Another program, DISAG78, is being developed which will use data supplied by the user to break up a standard ORANI industry classification into subclassifications (Bruce and Sutton 1984). Where a particular group of industries is the focus of attention, this can enhance the realism of an ORANI result. As the ORANI programs cannot handle much more than 115 industries, use of DISAG78 often implies the use of program AG78, which has the opposite purpose. Thus, the addition of 20 categories to the metal manufacturing sector has to be offset by the agglomeration of 20 or more industries in other sectors. AG78 may be used alone simply to reduce the size of the ORANI system. It is far cheaper to run ORANI if the number of industries is halved using program AG78.

All programs in the longrun suite of programs automatically handle these changes to model dimensions as long as the maximum number of commodities, industries, or occupations does not exceed 116. It is of course necessary that the same dimensions are adhered to throughout any sequence of computations.

APPENDIX II

Structure of the PIEOUT File

A PIEOUT file is a special case of the pseudo random-access file used by a number of ORANI programs. A library of programs and subroutines is available to handle these files: for further information consult the IAC programming staff.

The file consists of a series of matrices, each corresponding to one vector variable. The rows of the matrix correspond to the individual elements of the vector variables, whilst the columns correspond to the various policy shocks specified in the PIE input deck.

On creation of the file, the maximum number of matrices which potentially may be stored there is set and an INDEX vector, of length equal to this maximum number, is maintained.

The matrices are ordered according to the numbering system set out in Tables 7 to 9. Thus, matrix 1 shows changes in the industry outputs and the single-row matrix 34 the changes in the general real wage. A name is stored with each matrix, together with the numbers of rows and columns. Row and column totals are not stored.

Two particular aspects of the PIEOUT file may be stressed.

Firstly, it is necessary that each matrix on the file has the same number of columns. The values drawn from the same column of each matrix constitute an ORANI solution. No distinction is made, in the PIEOUT file, between the exogenous and endogenous variables - nor is any distinction necessary. To see this, compare two common shortrun closures of ORANI, differing only in that in the first the balance of trade is endogenous and real consumption exogenously shocked while in the second closure these roles are reversed. As long as the exogenous value of the balance of trade in the second simulation is equal to its endogenous change in the first, results for <u>all</u> variables will be identical in the two simulations.

Secondly, the PIEOUT file need not utilize every position allotted for matrices of results. Some positions are empty and this is recorded through the INDEX vector. Three distinct reasons may lie behind the absence of a matrix at some position.

(a) The file may have a maximum number of positions (the length of the index vector) greater than the number of ORANI vector variables. All positions greater than this number will be empty.

(b) A position may correspond to an endogenous variable for which values were not requested when the files were created.

(c) The position may correspond to an unshocked exogenous variable with zero values for all elements.

In the CSIRONET version of ORANI, a distinction is made between the first 71 vector variables, listed in Table 7, which comprise the FINAL system, and variables 72-84, listed in Table 8, which are optionally obtained by BACKSOLUTION from the FINAL variables. Exogenous variable must be chosen from the FINAL list, and the C file contains solutions for all these variables. Program EXPAND requires that we elevate common practice to the rule that values for all these FINAL variables appear - if non-zero - on the PIEOUT file. This makes it possible to distinguish between the possibilities (b) and (c) above. If an empty position on the PIEOUT file is numbered below 72 that variable is exogenous and zero. If the number is above 72, corresponding to a BACKSOLUTION variable, we can interpret the absence of information as the lack of any data concerning an endogenous variable.

The PIEOUT file generated by the standard ORANI PIE program is subject to some additional restrictions:

(a) The values of shocked exogenous variables must be entered on the PIE input deck in the order in which they appear in Table 9. Values for elements of any one vector variable also are entered in ascending order. The columns of the PIEOUT file which correspond to these shocks are ordered in the same way as they are specified on input to the PIE program.

(b) The PIE program always prints out the total effect of all specified shocks - optionally the effects of the component shocks are printed. The PIEOUT file <u>either</u> contains a column for each shock <u>or</u> a single totals column, <u>but not both</u> at once.

(c) The number of columns in a PIEOUT file is limited to the length of the largest vector variable - typically the number of commodities.

Programs ROWTOT, APPEND, LINCOM, and PIEDIT may be used to circumvent these additional restrictions.

APPENDIX III

Derivation of New Variables Calculated by Program EXPAND

This Appendix is a version of Appendix I of Horridge(1985). It incorporates certain important revisions, notably the addition of variables 116 to 119. A number of minor additions or corrections have also been made.

As set out in Horridge(1985), the longrun macro extension to ORANI consists of the addition of equal numbers of equations and variables. By means of the method of closure swapping, one of these new variables - the ratio of consumption to saving - is exogenized, whilst a standard ORANI variable - real consumption - is endogenized.

It follows that a sufficient description of the additional equations can be couched in terms of a set of definitions of the new variables listed in Table 9 above, and reproduced overleaf. To code program EXPAND it was necessary to define each new variable in terms of original ORANI variables or in terms of new variables already defined. This imposed a particular order on the work which is followed in the presentation below. By and large, the first new variables to be defined are logically dependent only on the variables computed in the standard ORANI computer programs. They do not depend on any additional data (such as data on foreign ownership) required for the macro extension. Such variables are annotated, on their heading lines, with a (1).

The method of computation of some new variables is not completely general. It is assumed, in some cases, that certain technical change and taste terms are exogenously constant (these correspond to ORANI variables 39 to 62 in Table 7). Therefore, the computed changes in these new variables may be invalid if technical change has been imposed. Variables which are defined in terms of these variables will also be invalid. The limitations are set out in the description of each variable below, but for quicker reference the annotation (2) has been attached to the heading line of each variable in this category. Program EXPAND checks that all the technical change terms are indeed zero. If not, the warning message 'YUK shock present at position nn - certain results may be invalid' is printed out.

TABLE 9: (replicated) NEW SCALAR VARIABLES ADDED BY PROGRAM EXPAND TO PIEOUT[1] FILE

POSITION		DESCRIPTION	SYMBOL IN TEXT
85	*	Aggregate nominal other demands	g
86	*	Price index other demands	$\pi^{(5)}$
87	*	Price index exports(Australian dollars)	π^{e}
88	*	Price index imports(Australian dollars)	π^{m}
89	*	Price index GDP expenditure side	π^{gdp}
90	*	Nominal GDP from expenditure side	gdp
91	*	Aggregate nominal land rentals	
92	*	Aggregate nominal fixed capital rentals	r
93	*	Aggregate nominal other cost tickets	
94	*	Aggregate nominal wage payments	
95	*	Tax revenue intermediate usage	
96	*	Tax revenue investment inputs	
97	*	Tax revenue consumption	
98	*	Aggregate export subsidies	
99	*	Tariff revenue	
100	*	Total nominal indirect taxes	u
101	*	Nominal GDP from income side	gdp
102	*	Total nominal current capital stock	
103		GNP - national income	gnp
104		Ratio consumption/saving	f
105		Overall local share of local rentals	q
106		Total national saving	s
107	*	Total capital stock - rental shares	k
108		Average rental price of locally owned cap	p^{KL}
109	*	Average rental price of all local capital	p^{K}
110		Average creation price, locally owned cap	π^{L}
111		Locally owned capital stock	$q+k$
112		Net rentals to overseas	r_x
113		Net foreign investment	r_i
114		Rentals to overseas	r_t
115		Rentals from overseas	r_f
116	*	Total investment price index	π^{i}
117	*	Total nominal investment	i
118	*	Total real investment	i_r
119	*	Absorption price index	π_{abs}

* Variables marked by an asterisk are simply extensions of standard ORANI results. The computation of the other new variables requires information beyond that contained in the standard ORANI database. This information constitutes an additional input to program EXPAND.

[1] PIEOUT is an output file of the standard ORANI programs as implemented on CSIRONET (see Sutton (1985)).

Lastly, only certain variables are used in the chain of definitions leading to the formula for calculating f - the ratio of consumption to saving. The longrun closure described in Horridge(1985) actually requires only these definitions. To assist the reader, such core variables are marked with the annotation (3) in the heading line. Variables not so marked yield useful information, but their value does not affect the values of the standard ORANI variables, after closure swapping to make f exogenous. Therefore, deficiencies in the computation of these noncore variables may have only limited significance. Note that no variable bears both annotations (2) and (3). Technical change does not preclude the application of the longrun closure described in Horridge(1985).

Notation follows the conventions of DPSV, as far as possible. As usual, variables in percentage change form appear in lower case. Note, however, that the symbol S_i is used to indicate the share of the ith component in a total, where the context determines both the relevant total and the nature of the ith component. Thus symbols such as S_1, S_2, and S_3 are constantly redefined as necessary. This seemed preferable to cluttering the notation with further affixes.

Variable 85: Aggregate Nominal Other Demands (1),(3)

Following DPSV (eqn. 16.1, p. 105) the demand for commodity i, source s, by the government is taken to be:

$$x^{(5)}_{(is)} = c_r h^{(5)}_{(is)} + f^{(5)}_{(is)} . \qquad i = 1,\ldots,g \quad s = 1,2$$

Advantage is taken of the circumstance that in the ORANI database there are no margins on other demands. The government sector therefore always pays the basic price for its consumption:

$$p^{(5)}_{(is)} = p^{(0)}_{(is)} . \qquad i = 1,\ldots,g \quad s = 1,2$$

The aggregate nominal expenditure is then computed as:

$$g = \sum_{i,s} S_{(is)} [x^{(5)}_{(is)} + p^{(0)}_{(is)}] , \qquad i = 1,\ldots,g \quad s = 1,2$$

where $S_{(is)}$ is the database share of good i from source s in total 'other' expenditure.

Variable 86: Price Index Other Demands (1)

$$\pi^{(5)} = \sum_{i,s} S_{(is)} p^{(0)}_{(is)} , \qquad i = 1,\ldots,g \quad s = 1,2$$

where the shares $S_{(is)}$ are as defined in the description of variable 85.

Variable 87: Price Index Exports (1)

The foreign price of Australian exports is found from:

$$p^{e}_{(i1)} = -\gamma_i x^{(4)}_{(i1)} + f^{e}_{(i1)} . \qquad i = 1,\ldots,g \quad \text{(DPSV 15.2, p. 104)}$$

The price index for exports is a weighted combination of these prices - which include subsidies - converted to Australian currency:

$$\pi^{e} = \phi + \sum_{i} S_i p^{e}_{(i1)} , \qquad i = 1,\ldots,g$$

where ϕ is the percentage change in the exchange rate and the S_i are the database shares of each commodity in total exports.

Variable 88: Price Index Imports (1)

The price index for imports is a weighted combination of c.i.f. import prices in Australian dollars. It is computed, in percentage change form, as:

$$\pi^{m} = \phi + \sum_{i} S_i p^{m}_{(i2)}, \qquad i = 1,\ldots,g$$

where ϕ is the percentage change in the exchange rate and the S_i are the database shares of each commodity in total imports.

Variable 89 GDP Deflator (1)

The gdp deflator is defined as:

$$\pi^{gdp} = S_1\pi^{(3)} + S_2\pi^{i} + S_3\pi^{(5)} + S_4\pi^{e} + S_5\pi^{m}$$

i.e., a weighted average of the price indices for household consumption, investment and other demands. $\pi_{(3)}$ is the standard ORANI CPI; π_i is the investment price index defined below as variable 116; while $\pi_{(5)}$, the goverment price index, and trade indices π^m and π^e are defined above; S_1 to S_5 are the component shares in GDP. Note that export and import prices are now included in this deflator, unlike in previous versions (See Variable No.119).

Variable 90 Nominal GDP from Expenditure Side (1),(3)

A standard definition of GDP is used:

$$GDP_{exp} = C + I + G + E - M .$$

All components except G and I are standard ORANI variables (although X and M need to be converted to Australian dollars). G is simply variable 85 described above - the nominal value of other demands - whilst I is variable 117 described below. The percentage change form is then:

$$gdp_{exp} = S_1c + S_2i + S_3g + S_4(e+\phi) + S_5(m+\phi).$$

This computation of gdp (rather than that from income side) is that used in the longrun closure equations.

Variable 91 Aggregate Nominal Land Rentals (1)

In percentage change form this variable is defined as:

$$\text{total rentals to land} = \sum_j S_j\, x^{(1)}_{(g+1,3)j} + p^{(1)}_{(g+1,3)j}, \qquad j= 1,\ldots,h$$

where $x^{(1)}_{(g+1,3)j}$ and $p^{(1)}_{(g+1,3)j}$ are, respectively, the percentage change in the quantity and the price of land used by industry j and the S_j are the shares of each industry in total rentals to land.

Variable 92 Aggregate Nominal Fixed Capital Rentals (1)

In percentage change form this variable is defined as:

$$r = \sum_j S_j\, x^{(1)}_{(g+1,2)j} + p^{(1)}_{(g+1,2)j}, \qquad j= 1,\ldots,h$$

where $x^{(1)}_{(g+1,2)j}$ and $p^{(1)}_{(g+1,2)j}$ are, respectively, the percentage change in the quantity and in the price of capital used by industry j and the S_j are the shares of each industry in total rentals to fixed capital.

Variable 93 Aggregate Nominal Other Cost Tickets (1),(2)

ORANI's 'other cost tickets' reflect the value of taxes and levies paid by firms plus the cost of working capital. EXPAND assumes that the quantity of tickets used by each industry is proportional to industry activity:

$$x^{(1)}_{g+2,j} = z_j. \qquad j = 1,\ldots,h$$

Thus the technical change terms $a^{(1)}_j$ and $a^{(1)}_{g+2,j}$ mentioned in DPSV eqn. 12.25 are assumed to be zero. The price of tickets is indexed to the consumer price index, via parameters $h^{(1)}_{g+2,j}$ (normally unity), and shifters $f^{(1)}_{g+2,j}$ (normally zero):

$$p^{(1)}_{g+2,j} = h^{(1)}_{g+2,j}\pi^{(3)} + f^{(1)}_{g+2,j}.$$

Thus the percentage change in aggregate ticket revenue may be expressed:

$$\text{ticket revenue} = \sum_j S_j(p^{(1)}_{g+2,j} + x^{(1)}_{g+2,j}), \qquad j = 1,\ldots,h$$

where the S_j are the database shares of each industry in total ticket revenue.

Variable 94: Aggregate Nominal Wage Payments (1),(2)

In percentage change form this variable is defined as:

$$\text{total wages to labour} = \sum_j S_j\, x^{(1)}_{(g+1,1)j} + p^{(1)}_{(g+1,1)j}, \quad j = 1,\ldots,h$$

where $x^{(1)}_{(g+1,1)j}$ and $p^{(1)}_{(g+1,1)j}$ are, respectively, the percentage changes in the quantity and the price of labour used by industry j and the S_j are the shares of each industry in the total wage bill.

The $x^{(1)}_{(g+1,1)j}$ are given by the ORANI vector variable no. 79 (LI) and are value weighted sums of industry j's demands for each occupation.

The $p^{(1)}_{(g+1,1)j}$ must be constructed as shareweighted sums of the prices of the m occupational labour grades. Building on DPSV, eqn. 22.6, p. 128):

$$p^{(1)}_{(g+1,1)j} = f^{(1)}_{(g+1,1)} + f^{(1)}_{(g+1,1)j} + \sum_m S_{mj}[cpi.h^{(1)}_{(g+1,1,m)j} + f^{(1)}_{(g+1,1,m)} + f^{(1)}_{(g+1,1,m)j}],$$

where:

$f^{(1)}_{(g+1,1)}$ = average wage shifter

$f^{(1)}_{(g+1,1)j}$ = industry-specific wage shifter

$f^{(1)}_{(g+1,1,m)}$ = occupation-specific wage shifter

$f^{(1)}_{(g+1,1,m)j}$ = (industry/occupation)-specific wage shifter

$h^{(1)}_{(g+1,1,m)j}$ = (industry/occupation)-specific indexation parameter

S_{mj} = share of occupation m in wage bill industry j .

The $f^{(1)}_{(g+1,1,m)j}$ although implemented by the standard ORANI programs, are assumed to be exogenously zero. If this is not the case, EXPAND prints out the the warning message 'YUK shock present at position nn - certain results may be invalid' where nn is a number from 53 to 62. Technological augmentation of the component grades of labour has also been excluded (the $a^{(1)}_{(g+1,1,m)j}$, see DPSV, p. 86, eqn. 12.56).

Variable 95: Tax Revenue Intermediate Usage (1),(2)

To calculate this flow, we assume that the tax is specific, fully indexed to the CPI, and is levied equally on domestic and imported equivalents. Thus the terms t(is,j1) and v(is,j1) in eqn. 18.20, DPSV p.116 are assumed to be zero. We also assume that the demand by an industry for any material input (disregarding source) is proportional to its activity level. Thus technical change terms appearing in equation 12.23 (DPSV, p. 81), namely the $a_j^{(1)}$, $a_{ij}^{(1)}$, and $a_{(is)j}^{(1)}$, are assumed to be zero. The total tax revenue is then:

$$\text{Intermediate Tax Revenue} = \pi^{(3)} + \sum_j S_j z_j , \qquad j = 1,\ldots,h$$

where $\pi^{(3)}$ is the consumer price index, z_j the activity level of industry j, and the S_j are the database shares, by industry, of the tax revenue.

Variable 96: Tax Revenue on Inputs to Capital Creation (1),(2)

To calculate this flow, we assume that the tax is specific, fully indexed to the CPI, and is levied equally on domestic and imported equivalents. Thus the terms t(is,j2) and v(is,j2) in eqn.18.20, DPSV pg.116 are assumed to be zero. We also assume that the demand by an industry for any material input to its capital creation (disregarding source) is proportional to its investment level. Thus technical change terms appearing in equation 13.4 (DPSV, p. 96), namely the $a_j^{(2)}$, $a_{ij}^{(2)}$, and $a_{(is)j}^{(2)}$, are assumed to be zero. The total tax revenue is then:

$$\text{Investment Tax Revenue} = \pi^{(3)} + \sum_j S_j y_j , \qquad j = 1,\ldots,h$$

where $\pi^{(3)}$ is the consumer price index, y_j the investment level of industry j, and the S_j are the database shares, by industry, of the tax revenue.

Variable 97: Tax Revenue on Household Consumption (1),(2)

To calculate this flow, we assume that the tax is specific, fully indexed to the CPI, and is levied equally on domestic and imported equivalents. Thus the terms t(is,3) and v(is,3) in eqn. 18.21, DPSV, p. 117 are assumed to be zero. The total tax revenue is then:

$$\text{Consumption Tax Revenue} = \pi^{(3)} + \sum_i S_i x_i^{(3)} , \qquad i = 1,\ldots,g$$

where $\pi^{(3)}$ is the consumer price index, $x_i^{(3)}$ the consumption of good i (regardless of source) and the S_i are the database shares, by commodity, of the tax revenue. The $x_i^{(3)}$ are calculated as Divisia aggregates of household consumption of domestic and imported equivalents:

$$x_i^{(3)} = \sum_s S_{is} x_{is}^{(3)} , \qquad s = 1,2 \quad i = 1,\ldots,g$$

where the s subscript differentiates imported from locally produced good i, and S_{is} is the database share of source s in the value of consumption of good i.

Variable 98: Aggregate Export Tax Revenue (1)

In calculating the revenue from export tax, we recognized that according to the ORANI database, the taxes were sometimes negative - actually subsidies. The tax rates are often endogenous, so that a subsidy may become a tax, or vice versa. Thus a special method of computation was designed. Contributions from tax on each commodity, ΔY_i, to the total change in export tax revenue, ΔY were calculated as follows.

For each commodity, the export tax revenue Y_i is equal to the difference between the value of exports at world prices, and the value at Australian prices (including margins). Call these F_i (foreign flow) and D_i (domestic flow). Then $Y_i = F_i - D_i$. Further, the power of the export subsidy, P_i is defined as the ratio D_i/F_i, so that $D_i = P_i F_i$. Hence:

$$Y_i = F_i - P_i F_i, \text{ and}$$

$$\Delta Y_i = F_i f_i - P_i F_i (p_i + f_i) ,$$

where the lower case letters denote **proportional** changes. Summing:

$$\Delta Y = \sum_i \Delta Y_i , \text{ and so}$$

$\Delta Y/Y$ is the proportional change in export tax revenue.

Both Y, the original revenue, F_i, the original 'foreign flows' and P_i, the original powers of the subsidies are calculated from the database. The f_i are derived via equation 15.2 of DPSV from the export prices and the world export demand shifters. The p_i are computed by the standard ORANI programs.

Variable 99: Aggregate Import Tax Revenue (1)

A similar method to that employed in calculating export tax was used, for the same reasons. Skipping the derivation:

$\Delta Y/Y$ is the proportional change in import tax revenue, where

$\Delta Y = \sum_i \Delta Y_i$ and

$\Delta Y_i = - F_i f_i + T_i F_i (t_i + f_i)$,

where the lower case letters denote proportional changes. T_i is the power of the tariff, whereas P_i, in the previous (export tax) derivation was the power of a subsidy; hence the sign differences.

Both Y, the original revenue, F_i, the original 'foreign flows' and T_i, the original powers of the tariffs are calculated from the database. Note that the T_i are not the IAC-generated equivalent rates used to compute the size of exogenous tariff shocks. The f_i are derived from the standard ORANI variables PM (import prices) and XL (import volumes).

Variable 100: Total Indirect Tax Revenue (1),(2)

Total indirect taxes are made up of the five components just described, namely, taxes on intermediate usage, investment inputs, household consumption, exports and imports. The percentage change in U, their sum, is simply:

$$u = \sum_w S_w u_w , \qquad w = 1,..,5$$

where the u_w are the percentage changes of the components, and the S_w are their database shares in indirect tax revenue.

Variable 101: Nominal GDP from Income Side (1),(2)

This is defined as the sum of payments to land, labour, capital, and other cost tickets, plus indirect taxes. The calculation of all these quantities is described above. The percentage change in GDP is then found as a weighted sum of the changes in its components, the weights being the component database shares in GDP.

The result should be equal to that obtained from the expenditure side - see above (Variable 90). In practice, tiny discrepancies arise. The revisions to program EXPAND described in the description below of variables 116 to 118 have reduced these discrepancies very greatly. However, results for Variable 101 are invalidated if any of several vectors of technical change or taste shocks are nonzero. Hence the longrun core equations use the calculated value of Variable 90 - GDP from the expenditure side.

Variable 102: Total Nominal Current Capital Stock (1)

This variable is simply defined as the change in the nominal value of the aggregate capital stock as computed by the standard ORANI programs (variable 26 - K0). In percentage change form:

$$\text{Total} = \sum_j (k_j(0) + \pi_j)\Phi_{2j} . \qquad j= 1,\ldots,h$$

(Refer eqn. 22.4, DPSV, p. 127). The $k_j(0)$ are the percentage changes in the quantity of capital employed in each industry, while the π_j are the changes in the price of creating a unit of such capital. The Φ_{2j} are the value shares of capital in each industry as utilized by the standard ORANI programs. That is, they are the normalized column sums of the capital stocks matrix. As detailed in Section 4 of Horridge (1985), these weights were deemed inappropriate for longrun applications. Thus this variable is computed for shortrun purposes only.

Variable 107: Total Capital Stock - Rental Shares (1),(3)

This variable is calculated in a similar manner to the variable k(0) - aggregate capital stock - computed by the standard ORANI programs, and described in DPSV, eqn. 22.4, p. 127. The difference lies in the weights attached to the changes in individual industries' capital stocks. For most purposes it is more appropriate to use weights based on the industry shares of total rentals to fixed capital. Thus:

$$k = \sum_j S_j k_j(0) , \qquad j = 1,\ldots,h$$

where the S_j are database industry shares of total rentals to fixed capital, and $k_j(0)$ is the percentage change in the capital stock in industry j.

Variable 108: Average Rental Price of Locally-Owned Capital (3)

This variable represents the overall rental price of all fixed capital owned by Australians. It is computed as:

$$p^{KL} = \sum_j S_j p_{(g+1,2)j} , \qquad j = 1,..,h$$

where the S_j are the shares accounted for by each industry in total revenue to fixed capital, accruing to Australians.

<u>Variable 109:</u> <u>Average Rental Price of Capital</u> (1)

This variable represents the overall rental price of all fixed capital located in Australia. It is computed as:

$$p^K = \sum_j S_j p_{(g+1,2)j} , \qquad j = 1,\dots,h$$

where the S_j are the shares accounted for by each industry in total revenue to fixed capital. It is the appropriate price index for Variable 107.

<u>Variable 110:</u> <u>Average Creation Price of Locally-Owned Capital</u> (1)

This variable represents the overall creation price of all fixed capital owned by Australians. It is computed as:

$$\pi^L = \sum_j S_j \pi_j , \qquad j = 1,\dots,h$$

where the S_j are the shares of each industry in total investment in Australia by Australians. According to the methodology set out in Section 4 of Horridge(1985), the distribution of this investment is the same as the distribution of Australians' capital assets between the industries, and the same too as the distribution, between the industries, of capital rentals accruing to Australians. This simplifying feature, however, is neither essential to the implementation of the longrun closure, nor particularly crucial to the results, in most instances. The exception would be a shock which caused a marked dispersion in the π_j.

Condensed Derivation of Core of Longrun Module

The following 8 equations (excerpted from Table 2 of Horridge(1985)) form the core of the longrun system. They are fully explained in Section 2 (*ibid*).

(1) $gnp = \beta^1 (q + k + p^{KL}) + \beta^2 othinc + \beta^3(k^F + \phi)$ [Income]
(2) $gnp = \alpha^1 c + \alpha^2 g + \alpha^3 s_d + \alpha^4 s_f + \alpha^5 bi$ [Expenditure]
(3) $c = f + s$ [Consumption]
(4) $bi = gnp$ [Balancing Item]
(5) $q + k = \lambda(s_d - \pi^L)$ [Australian Equity in Local Capital]
(6) $k^F = \lambda(s_f - \phi)$ [Australian Ownership of Overseas Capital]
(7) $s_f = s$ [Investment by Australians Overseas]
(8) $s_d = s$ [Local Investment by Australians].

Where β^1 = share locally generated capital rentals in GNP,
β^2 = share other locally generated income in GNP,
β^3 = share rentals from overseas in GNP,
α^1 = share consumption in GNP,
α^2 = share government spending in GNP,
α^3 = share saving invested locally in GNP,
α^4 = share saving invested overseas in GNP,
α^5 = share balancing item in GNP, and
λ = parameter reflecting assumed adjustment path of saving.

The equations above implicitly determine the variables:

gnp	national income	103*
q	Australian share of capital stock	105
q+k	Australian capital assets	111
k^F	overseas capital stock owned by Australians	
s_d	Australian saving used to purchase capital stock in Australia	106*
s_f	Australian saving used to purchase capital stock overseas	106*
s	Australian saving	106*
bi	residual component of national expenditure remitted overseas (not included in rentals to/from overseas or in investment flows)	103*
f	ratio of consumption to saving.	104

(* Some distinct variables share the same computer number, taking advantage of the fact that they always have the same percentage change.)

All other variables mentioned in the group of equations may be computed fairly directly from standard ORANI variables. In this section the condensed form of the 8 equations is presented and derived. The method used is to calculate f as an explicit function of the standard ORANI variables. Then the remaining new variables are successively computed as functions of f and the standard ORANI variables. In other words, the system is triangularised algebraically.

As a preliminary, we remind the reader that the variable 'othinc' appearing in the first equation is only a shorthand for the percentage change in all 'other income' - apart from capital rentals - making up GDP. By assumption, all this other income accrues to Australians. Thus, in levels:

GDP = OTHINC + (AGGREGATE NOMINAL FIXED CAPITAL RENTALS),

or in percentage change form:

$$gdp = B_1 othinc + B_2 r ,$$

where B_1 and B_2 are database shares and where r is the percentage change in capital rentals - variable 92 - and gdp is calculated from the expenditure side. (Refer to the descriptions of variables 90 and 101.) Rearrangement gives:

$$othinc = C_1 gdp + C_2 r ,$$

where $C_1 = 1/B_1$ and $C_2 = -B_2/B_1$. (C_1 and C_2 are printed out by program EXPAND under the names SHRNC1 and SHRNC2. Although EXPAND computes othinc it is not printed out.)

Using the equations:

$$q + k = \lambda(s_d - \pi^L) \qquad (5)$$

$$k^F = \lambda(s_f - \phi) \qquad (6)$$

$$s_f = s_d = s \qquad (7),(8)$$

we may rewrite equation (1) as:

$$gnp = S_1 c + S_2 f + S_3 \pi^L + S_4 p^{KL} + S_5 othinc + S_6 \phi,$$

where c and ϕ are standard ORANI variables - nominal household consumption and the exchange rate. p^{KL} and π^L are new variables 108 and 110 described above. The coefficients S_1 to S_6 are printed out by program EXPAND having been computed as:

$$S_1 = \lambda(\beta_1 + \beta_3)$$

$$S_2 = - S_1$$

$S_3 = -\beta_1\lambda$

$S_4 = \beta_1$

$S_5 = \beta_2$

$S_6 = \beta_3(1 - \lambda)$.

Values of S_1 to S_6 are printed out, whilst values of β_1 to β_3 appear under the names SHRGNI (1-3),

A similar transformation is applied to equation 2 of Table 2, making use of equation 4 :

$bi = gnp$, so that (4) becomes

$gnp = T_1c + T_2g + T_3f$.

Coefficients T_1 to T_3 are printed out and equal:

$T_1 = (\alpha_1 + \alpha_2 + \alpha_3)/(1 - \alpha_5)$

$T_2 = \alpha_2/(1 - \alpha_5)$

$T_3 = -(\alpha_3 + \alpha_4)/(1 - \alpha_5)$.

Values α_1 to α_5 are printed out as SHRGNE(1-5).

Combining these two rearrangements we have:

$$T_1c + T_2g + T_3f = gnp = S_1c + S_2f + S_3\pi^L + S_4p^{KL} + S_5othinc + S_6\phi.$$

Thus f may be expressed in terms of the already computed variables:

$$f = ((S_1-T_1)c + S_3\pi^L + S_4p^{KL} + S_5othinc + S_6\phi - T_2g)/(T_3 - S_2) .$$

Having performed these manipulations the going becomes easy:

$gnp = T_1c + T_2g + T_3f = bi$ defining variable 103

$s_d = s_f = s = c - f$ defining variable 106

$(q+k) = \lambda(s_d - \pi^L)$ defining variable 111

$(q+k+p^{KL}) = (q+k) + p^{KL} = r_a$,

= revenue from local capital accruing to Australians.

$q = (q+k) - k$, defining variable 105

where k,(q+k) are variables 107,111. Thus all eight variables are computed.

Variable 112: Net Rentals to Overseas

This is computed merely as the difference between rentals to and rentals from overseas. In percentage change form:

$r_x = Q_1^2 r_t + Q_2^2 r_f$, where Q_1^2 and Q_2^2 are computed from the database, and r_t and r_f are variables 114 and 115.

Variable 113: Net Foreign Investment

This is defined as the difference between national saving and aggregate investment. In percentage change form:

$r_i = Q_1^3 i + Q_2^3 s$, where Q_1^3 and Q_2^3 are computed from the database, and i and s are variables 117 and 106.

Variable 114: Rentals Paid to Overseas

This variable represents the capital income earned in Australia, yet accruing to foreigners. It is merely the difference between total capital income, R(variable 92), and that accruing to Australians, R_a. So in percentage change form:

$$r_t = Q_1^1 r + Q_2^1 r_a,$$

where Q_1^1 and Q_2^1 are computed from database shares. r_a is found by adding the results for variable 111 (q+k) and variable 108 (p^{KL}), which represent, respectively, the quantity and rental price of local capital.

Variable 115: Rentals from Overseas

These rentals represent the earnings of capital located overseas, yet owned by Australians. The unit rental price is deemed to be constant in overseas currency so the Australian value of the rentals is simply the product of the exchange rate, ϕ, and the amount of such capital, K^F. In percentage change terms:

$$r_f = k^F + \phi .$$

From equations (6) and (7) of Table 2 we deduce:

$$r_f = \lambda(s - \phi) + \phi,$$

where s and ϕ are variables 106 and 9.

Variable 116: Total Investment Price Index (1)

Variable 117: Total Nominal Investment (1),(3)

Variable 118: Total Real Investment (1)

The standard ORANI programs calculate values of nominal and real investment, and an investment price index, which take in all those industries where investment is governed by the theory set out in DPSV, p. 121, equation 19.9. These industries are known as the private, or endogenous-investment industries. The rest follow the rule given by equation 19.11 ibid. The membership of each group is user selectable. For the purposes of national accounting, it is necessary to define new variables which reflect the nominal and real value of investment in all industries. Fortunately, the standard ORANI programs calculate, for each industry, the change in real investment (y_j) and in its price (π_j). Thus it is simple enough to calculate the new aggregates:

116. $\pi^i = \sum_j S_j \pi_j \qquad j = 1,..,h$ Total Investment Price Index

117. $i = \sum_j S_j (\pi_j + y_j) \qquad j = 1,..,h$ Total Nominal Investment

118. $i_r = \sum_j S_j y_j \qquad j = 1,..,h$ Total Real Investment

Here the S_j are industry shares in total gross expenditure on investment.

The distinction between total investment and total 'private' investment was not made in Horridge (1985), the companion publication to this manual. References in that paper to investment, aggregate investment and so on, should be understood to refer to total investment, rather than the endogenous, or private component alone.

Variable 119: Absorption Price Index (1)

This deflator is defined as:

$$\pi^{abs} = S_1\pi^{(3)} + S_2\pi^i + S_3\pi^{(5)}$$

i.e., a weighted average of the price indices for household consumption, investment and other demands. $\pi^{(3)}$ is the standard ORANI CPI; π^i is the investment price index defined above as variable 116; while $\pi^{(5)}$, the goverment price index is defined above; S_1, S_2 and S_3 are the component shares in absorption. Note that export and import prices are not included in this deflator (See Variable No.89).

APPENDIX IV

Errors and Error Messages

This Section deals with errors which cause program execution to abort. It is quite possible for the programs to run successfully without performing the intended task - especially if the input files and data were deficient. Hence the necessity to carefully examine the printout from 'successful' runs, checking for non-fatal warning messages. Fatal errors fall into three main categories:

(a) Errors in the JCL.

These will cause processing to end before the program itself is executed. Typical causes include failure to specify user-specific files properly, and incorrect JCL syntax. As familiarity with NOS is assumed in this guide, such errors are not treated here. However, one point to note is that from time to time the storage location of standard (non-user-specific) data and program files held by the IAC may be changed.

(b) Errors in data input.

If the indicated comment cards are absent from the data deck, or if the user's numbers are not formatted correctly, a read error will result. Another cause could be failure to terminate a matrix or column range selector sequence (with a pair of zeros in I5 format - see Section 7. Generally, the programs print out an 'echo' of any data which is read in. Therefore, the offending line will probably be the line subsequent to the last line to be successfully echoed.

(c) File dimensions too large.

No more than 150 positions are allowed in any input file, whether the positions are occupied or not. Longer input files induce a fatal error message. Additionally, matrices are limited to a maximum size of 116 in either direction. If a larger matrix is found, an error will result. The message will resemble:

REDRAS Error No. 58 or 54

116 nr 116 nc 0 0 0 0.0

where nr and nc are the dimensions of the input matrix, exceeding the limit of 116. The message:

OPENRA Error No. 55

LU	0	0	0	0	0	0	0.0

indicates failure to open the file assigned to tape no. LU in the JCL.

The programs provide a rudimentary 'trace' facility in that a record is printed of all file operations. Thus the last printed output, before execution is terminated, should identify either the matrix which caused trouble, or its predecessor. The 'predecessor' may either be the preceding matrix on the file, or the last 'selected' matrix to be processed.