IMPACT PROJECT

A Commonwealth Government inter-agency project in co-operation with the University of Melbourne, to facilitate the analysis of the impact of economic demographic and social changes on the structure of the Australian economy

COMPUTING MANUAL FOR THE SNAPSHOT MODEL

by

Alexandra Strzelecki and Bruce S. Coe

IMPACT Computing Document No. C2.02 (Supersedes C2.01)

Melbourne April 1981

The views expressed in this paper do not necessarily reflect the opinions of the participating agencies, nor of the Australian Government.

IMPACT PROJECT RESEARCH CENTRE 153 Barry Street Carlton 3053
Postal Address: Impact Centre, University of Melbourne, Parkville, Vic., 3052, Australia
Phones: (03) 345 4844 extensions 7417 & 7418
After hours (03) 341 7417 or 341 7418

C O N T E N T S

This document, C2.02, is a revised edition of C2.01 (April 1980). The main changes consist of corrections to the sample decks for running the computer programs and the addition of appendix 4 which summarizes the computer files used in SNAPSHOT.

A COMPUTING MANUAL FOR THE SNAPSHOT MODEL*

by

Alexandra Strzelecki and Bruce S. Coe†
Industries Assistance Commission

1. INTRODUCTION

This manual describes how to use the computer programs which calculate numerical solutions for SNAPSHOT, a long term economy-wide model of the Australian economy which has been developed within the IMPACT Project[1]. Inspection of the equations and inequalities which specify the SNAPSHOT model reveals that the system to be solved is very large and contains some non-linearities (apart from the inequalities). The model is solved by the method of joint maximization[2], which takes advantage of special features arising from the economic nature of the problem to reduce the programming effort and the computer costs involved. The solution procedure for the system of non-linear equations is iterative and based on a linear programming package[3].

---

* This manual has grown out of an earlier version written by Bruce Coe and John Harrower in 1977. The suite of programs and files has now been simplified and compacted for more convenient use.

The authors wish to thank David Vincent and John Harrower for assistance in describing the relevant details of the model, and John Sutton for editorial suggestions and for specifying the improvements to the programs, file structure and data inputs. We should also acknowledge the contribution of Clive Edington of CSIRO Division of Computing Research, Melbourne towards the iterative use of the APEX linear programming package.

† Now at Department of Industry and Commerce.

1. For details of the model see Dixon, Harrower and Powell (1976), Dixon, Harrower and Vincent (1978) and Dixon and Vincent (1979).
2. Dixon (1976).
3. The details of the iterative solution technique are documented in Edington and Harrower (1977).

A COMPUTING MANUAL FOR THE SNAPSHOT MODEL*

by

Alexandra Strzelecki and Bruce S. Coe†
Industries Assistance Commission

1. INTRODUCTION

This manual describes how to use the computer programs which calculate numerical solutions for SNAPSHOT, a long term economy-wide model of the Australian economy which has been developed within the IMPACT Project[1]. Inspection of the equations and inequalities which specify the SNAPSHOT model reveals that the system to be solved is very large and contains some non-linearities (apart from the inequalities). The model is solved by the method of joint maximization[2], which takes advantage of special features arising from the economic nature of the problem to reduce the programming effort and the computer costs involved. The solution procedure for the system of non-linear equations is iterative and based on a linear programming package[3].

---

* This manual has grown out of an earlier version written by Bruce Coe and John Harrower in 1977. The suite of programs and files has now been simplified and compacted for more convenient use.

The authors wish to thank David Vincent and John Harrower for assistance in describing the relevant details of the model, and John Sutton for editorial suggestions and for specifying the improvements to the programs, file structure and data inputs. We should also acknowledge the contribution of Clive Edington of CSIRO Division of Computing Research, Melbourne towards the iterative use of the APEX linear programming package.

† Now at Department of Industry and Commerce.

1. For details of the model see Dixon, Harrower and Powell (1976), Dixon, Harrower and Vincent (1978) and Dixon and Vincent (1979).
2. Dixon (1976).
3. The details of the iterative solution technique are documented in Edington and Harrower (1977).

In this manual, however, we shall not concern ourselves with the details of the actual mathematical problem or its solution algorithm. It is assumed that users of this manual are familiar with the economic specification of the model and its subsequent validation,[1] the jointmax algorithm[2] and the basic ideas behind the computing strategy[3] used to obtain numerical solutions to the model. All that is intended here is to detail the computer programming tasks needed to enable users to compute their own SNAPSHOT solutions.

The user will not need to alter any of the actual programming structure of the SNAPSHOT computer programs. What the user will need to acquire is an understanding of how the large SNAPSHOT data base is assembled on the appropriate input files. Hence, this manual is largely a description of the input files, their creation and their manipulation into the final forms required by the SNAPSHOT solution program.

All the computing is done on the CSIRO Cyber 76 and the programs are written in FORTRAN. The solution method makes much use of CSIRONET-specific options and hence the manual is applicable only to CSIRONET.

Throughout this guide, mention will be made of SUPERPASSION, APEX and UPDATE. SUPERPASSION is a matrix manipulation computer program originally developed at Harvard University. It has been adapted by the Tariff Board/Industries Assistance Commission for use on the CSIRO

---

1. Dixon, Harrower and Vincent (1978).
2. Dixon (1976) and Harrower and Vincent (1977).
3. Edington and Harrower (1977).

Computer Network. Appendix 1 describes how to read and write files in SUPERPASSION format. Also see program SNPMAT, which is described on page 33.

APEX (in this case APEX III) is a Control Data Corporation linear programming package which is available on the CSIRO network.

The UPDATE facility on the CSIRO Cyber 76 provides a means of maintaining program and data decks in conveniently amendable compressed format on magnetic disc and/or tape.

2. THE COMPUTING TASK

Inspection of the system of equations and inequalities which specify the SNAPSHOT model reveals that the computation of its solution involves a very large and non-linear system of equations and inequalities. This non-linear optimization problem can be approximated by an appropriately constructed linear model (see Dixon (1976)). The basic solution strategy is to make linear approximations to the non-linearities and then solve the linear program, hoping that the linear approximations are still valid in the optimal solution. If some of the approximations are inaccurate then they are changed and the linear model is re-run. This whole process of approximating, solving, testing the solution and refining the approximations is performed iteratively under computer control.

Thus the main computing task is to set up a system which makes it easy to transfer those inputs which are required to specify the linear program (LP) to the LP package (APEX), to solve the LP problem, and to transfer the LP solution back from the package. This is needed so that many iterations can be done in one computer run without the need for user intervention.

The iterative solution process is illustrated in Figure 1, whilst a more detailed algorithm for the computer system is shown in Figure 2.

LP packages normally expect input data on cards. However, the packages usually can be instructed to read data card images from an alternate (program defined) file which is already stored on the system.

Figure 1 : Basic Solution Strategy

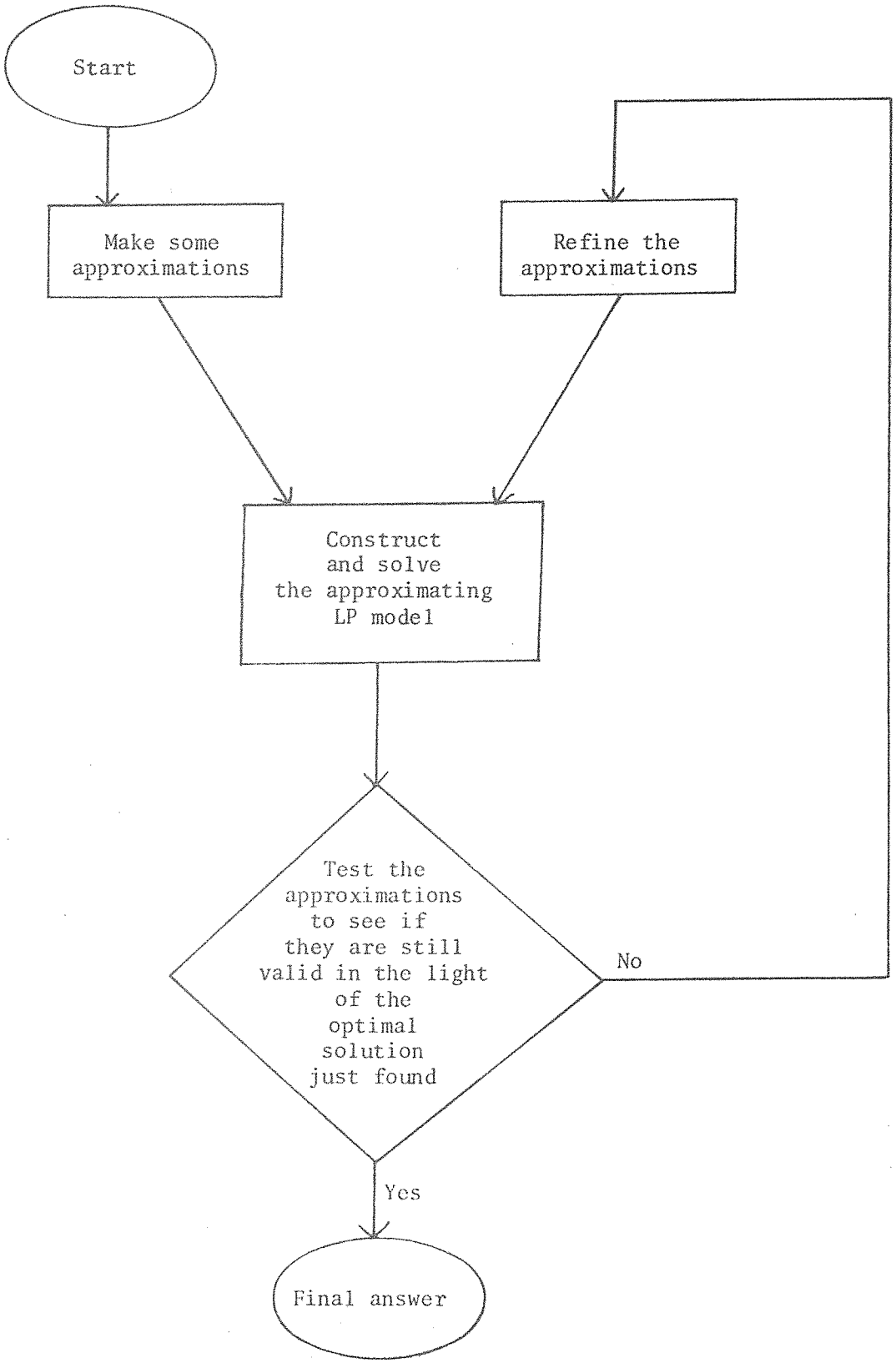
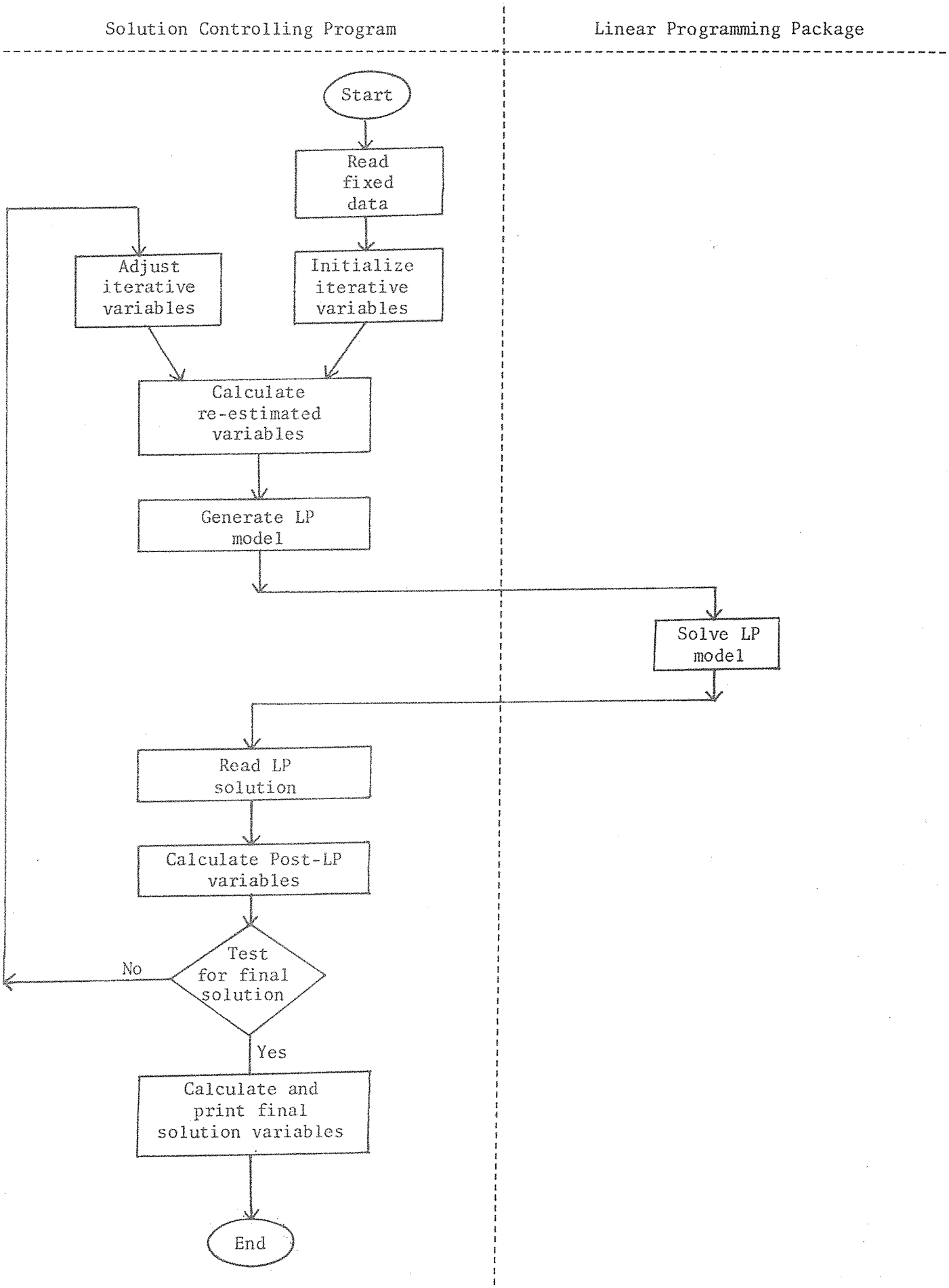


Figure 2 : Computing Algorithm



This feature can be used to get the package to read a file which is in fact prepared by a separate program. The FORTRAN programmer can for example use simple FORMAT statements[1] to write a file of "card images" which is acceptable to the LP package. Note that no cards are actually punched; files are just transferred from one program to another.

Similarly, after the LP problem has been solved, instead of printing the results on the printer, they can be written on to another file. The contents of that file can then be transferred back to the user's program and the results can be extracted by (say) READ statements in FORTRAN.[2]

Thus it is possible to repeatedly generate an LP problem, solve it, analyse the results and generate the next LP problem, etc. The programming to control this flow of steps is described in more detail later.

---

1. See Edington and Harrower (1977) for more details and examples of writing card images from FORTRAN.
2. See ibid. for an example of reading a solution file.

3. OVERVIEW OF THE SUITE OF PROGRAMS

Computation of a SNAPSHOT solution requires running four jobs in the appropriate order. These jobs are SNAP1, SNAP2, SNPCOVH and SNPPROG. They make use of a library file, AS1990SNAPLIB, an UPDATE file, AS1990SNAPUPDATE, and a data file ASSTDSNAPFIXED. The UPDATE file contains card images of the FORTRAN programs, and also some data. All three files[1] have ID=DTBIAS, are stored on the IAC disc (GETSET, DTB3006) and have a password which is available to authorised users.[2]

The inter-relationship between the various jobs and files[1] is illustrated in Figure 3. The functions of the 4 jobs are:

(1) SNAP1 produces the user's version of SNAPFIXED, using the standard data file ASSTDSNAPFIXED as the starting point,

(2) SNAP2 performs algebraic manipulation on SNAPFIXED to produce file SNAPFINAL,

(3) SNPCOVH creates file SNAPCONSVH, using UPDATE data as the starting point, and

(4) SNPPROG solves the model.

In addition,

(5) SNPMAT prints SUPERPASSION files, and

(6) LISTUPD (see Appendix 3) prints UPDATE decks.

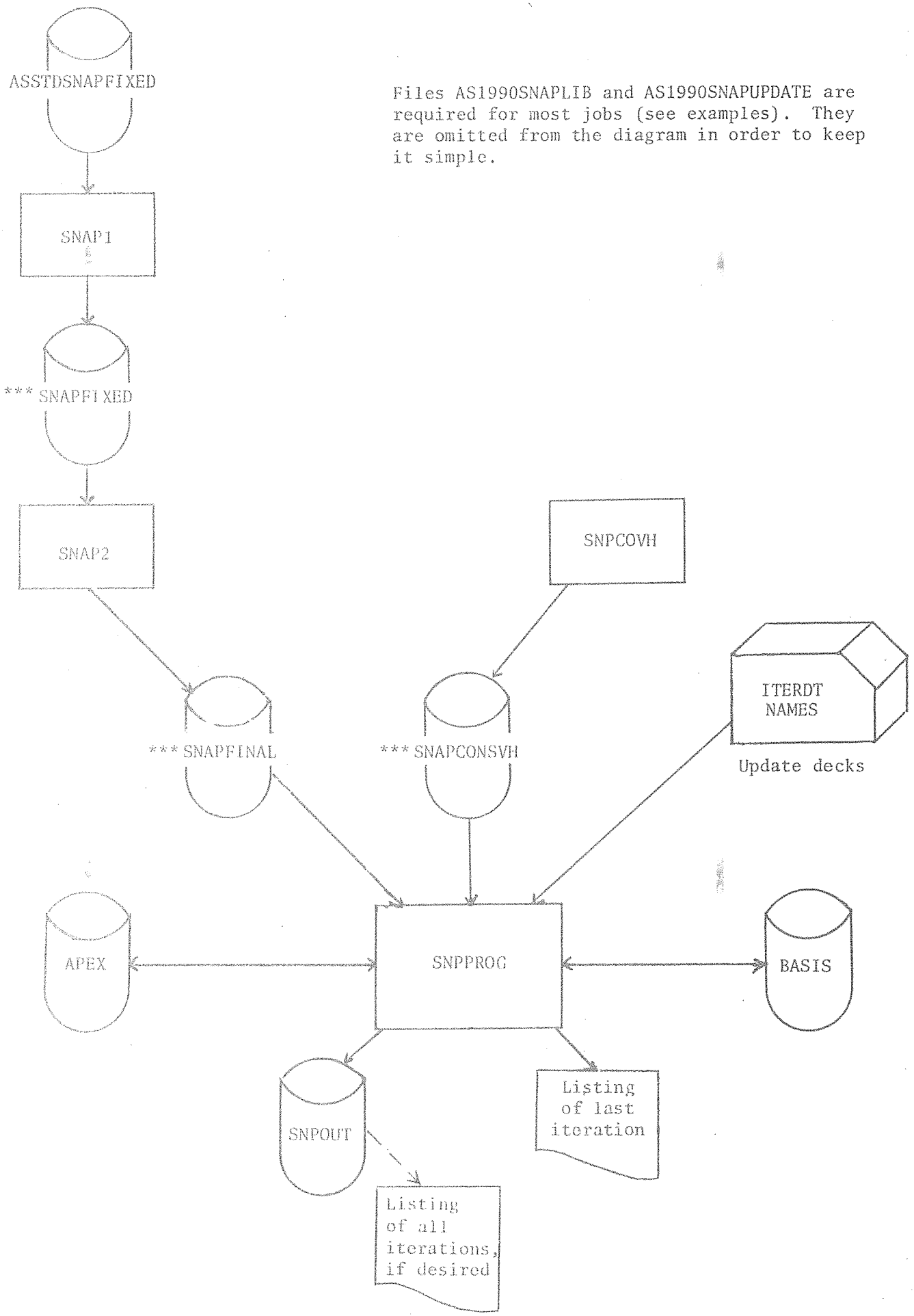
In normal usage it is expected that the user will want to provide special data for job SNAP1 and perhaps SNPCOVH. On the other hand, job SNAP2 and SNPPROG follow mechanically from the other jobs and there is little opportunity to provide data inputs.

---

1 Further notes on these files are provided in Appendix 4.

2 Intending users should check with the IMPACT information officer (Industries Assistance Commission, P.O. Box 80, Belconnen, ACT, 2616, Telephone (062) 641144) concerning passwords, availability of files, and possible revisions to setnames, etc.

Figure 3 : Flow Diagram for SNAPSHOT Computation



Files AS1990SNAPLIB and AS1990SNAPUPDATE are required for most jobs (see examples). They are omitted from the diagram in order to keep it simple.

It has been assumed that the numbers of industries, occupations, etc., would not have to be changed. If they were to be changed, the same four jobs would be run, but the user would have to provide many more changes.

The actual SNAPSHOT program needs the following inputs:

(1) two input files containing numerical data (SNAPFINAL and SNAPCONSVH),

(2) one input file containing starting values of iterative values (generated from UPDATE deck ITERDT),

(3) one input file holding the row and column names of the linear program (generated from UPDATE deck NAMES),

(4) a file on which the previous linear program basis was stored,

(5) the APEX linear programming package, and

(6) a library file on which the SNAPSHOT program is stored.

These input requirements are presented diagramatically in Figure 3.

The coefficients and iterative variables which are input to the linear program are changed after each iteration until convergence of the iterative variables is obtained.

Details of the LP matrix include the numbers of rows and columns, the equations which are treated as bounds, and the iterative variables, which, because they are held constant for an LP solution, are added to the 'standard' right hand side variable to form the programming problem's RHS for each LP iteration. (A description of the nomenclature used in the model is presented in Appendix 2.)

To assist the reader in the following discussion of the input files required for a SNAPSHOT solution we have set out in Figure 4 the SNAPSHOT LP tableau.

Figure 4 : The SNAPSHOT Linear Programming Matrix (Input to APEX)

NO.	NAME	NUMBER	[g*(t+1)]	(n)	(n)	(n)	(n)	(n)	(1)	(1)
		NAME	VILLAMB	VLIMP	VLXONE	VLXTWO	ITERATIVE VARIABLES (Added to RHS)			R.H.S.
		SYMBOL	λ	M	X_1	X_2	X^+	M^+	N^+	
(1)	OBJ*		V	$-\theta^+(\overline{p^m})'\hat{\tau}$						
(n)	PROD		QH	-1	$-1(I-A)$	$-(I-A)+K(\hat{\tau}\beta^+ + \hat{n})$	1			$\leq -(\bar{G}+\bar{E})$
(n)	✓ NBIMP1			1				-1		≤ 0
(1)	LABOUR				$\bar{w}'\ell$	$\bar{w}'\ell$			-1	≤ 0
(n)	✓ NBIMP1				1					$\leq -(I-\hat{n})^t\overline{K(0)}$
(1)	TRADBL			$(\overline{p^m})'$						$\leq (\overline{p^e})'\bar{E}+\bar{B}$
(g)	SLAM		1							$= 1$

* Objective function ✓ Bounds option used

4. DETAILS OF INDIVIDUAL PROGRAMS, INCLUDING EXAMPLES

In the examples which follow, stars in the control cards indicate information which must be supplied by the user, such as prefix for file name, user ID and retention period. No user passwords have been included in these examples. It should be noted that the examples are written on the assumption that user files are to be written on SYSTEM.

For 109 industries appropriate CPU times for the jobs and approximate sizes for the files created are:

CPU times

job SNAP1	5 seconds
job SNAP2	4 seconds
job SNPCOVH	2 seconds
job SNPPROG	For a typical run of 12 iterations the total CPU time is approximately 130 seconds. In the first iteration, the program takes 4 seconds and APEX takes 12 seconds. In subsequent iterations the program takes 4 seconds and APEX 6 seconds.

File sizes

file SNAPFIXED	641,920 characters
file SNAPFINAL	768,690 characters
file SNAPCONSVH	7,830 characters
file SNPOUT	136 characters per record

first iteration ; 1200 records

subsequent iterations : 830 records on average

Thus, for a typical run of 12 iterations the total is approximately 1.4 million characters.

IT IS STRONGLY RECOMMENDED THAT JOB SNPPROG ALWAYS BE RUN AT THE CHEAPEST COMPUTING RATE.

Many of the programs make use of UPDATE corrections to decks which are on UPDATE file AS1990SNAPUPDATE.

4.1 Program SNAP1

In order to compute a solution to the SNAPSHOT model, most of the numerical data must first be assembled on file SNAPFIXED. A standard set of data is provided on file ASSTDSNAPFIXED. As it is expected that in most instances users will want to use the standard set of data with only minor modifications, the computer program SNAP1 has been set up to create SNAPFIXED by using values from ASSTDSNAPFIXED as the default values. User supplied values may however be inserted when considered necessary.

Contents of file ASSTDSNAPFIXED

The file is written in SUPERPASSION format (see Appendix 1). The data held on file ASSTDSNAPFIXED is shown in Table 1.

Table 1 : Contents of File ASSTDSNAPFIXED

Position	Dimension	Variable	Brief Description	Symbol
1	N x 1	DFGPUR	government purchases	$\overline{G}$
2	N x 1	DFEXP	exports of commodities	$\overline{E}$
3	N x 1	DFIMPS	import shares of the domestic market [1]	γ
4	N x 1	DFEXPP	export prices (f.o.b.) in foreign currency	$\overline{p^e}$
5	N x 1	DFIMPP	import prices (c.i.f.) in foreign currency	$\overline{p^m}$
6	N x 1	DFTAR	ad valorem tariff rates in snapshot year	$\overline{\tau}$
7	H x 1	DFRWR	relative wage rates	$\overline{W}$
8	1 x 1	DFBOTF	balance of trade deficit	$\overline{B}$
9	N x 1	DFP	excess ad valorem tariff per unit domestic price	
10	N x 1	DFX	indicator - N, M or E	
11	N x 1	DFKBAS	base year capital stocks	$\overline{K(0)}$
12	N x 1	DFDEP	industry specific depreciation rates	η
13	N x N	DFKMAT	capital matrix	K
14	N x 1	DFKRRR	relative rates of return	$\overline{r}$
15	N x N	DFAMAT	I-O coefficients matrix	A
16	H x N-1	DFLMAT	labour requirements (also known as LMAT)	ℓ
17	N x G	DFQ	transformation matrix (also known as QMAT)	Q
18	N x 1	DFTE	export taxes	t_E
19	N x 1	DFTC	taxes on consumer goods	t_C
20	N x N	DFT1	ad valorem taxes on intermediate usage	T_1
21	N x N	DFT2	ad valorem taxes on creation of capital stocks	T_2

1 DFIMPS = (competing imports + duty)/(domestic production).

Inputs to SNAP1

1. Standard date file ASSTDSNAPFIXED on TAPE1.

2. UPDATE corrections to the program (optional). Appendix 3 describes how to obtain a listing of the update files.

3. UPDATE corrections to the data decks (optional).

4. TAPE 4 (optional) : User's version of any arrays he wants replaced (flag = 2, see card data) all in SUPERPASSION format (described in Appendix 1) in the following order: 1-3, 10, 4-9, 11-21. Not all of these have to be present. Only the arrays where the flag has the value 2 are required on TAPE 4.

5. Data on cards. Cards a and b are mandatory, card c depends on the setting of the flags in b.

   (a) N, H, G, Y in format 4I5 where:

   N is the number of industries,

   H is the number of occupations,

   G is the number of consumer goods, and

   Y is the number of years between the base year and the snapshot year.

   (b) 21 flags, in the format (21 I1), that indicate which arrays are to be changed. Flag K refers to matrix K.

   Each flag can have values 0-3 and flags for arrays 2, 4, 5 can have a value 4.

The values have the following meanings:

0 - no change,

1 - change individual cells, individual rows or individual columns in an already existing array,

2 - read the whole new array from TAPE 4,

3 - preset the array to zero before doing the changes described for flag 1, and

4 - for array 2 call subroutine EXPT before doing any changes described for flag 1;
for arrays 4, 5 call subroutine PRICES before doing any changes described for flag 1.

Whenever flags take the values 1, 3 or 4 subroutine SCAL is called. SCAL can take the following actions:

1. Replace individual cells,
2. Scale a row, or
3. Scale a column.

If both the row number and the column number are positive and non-zero, then only an individual cell is changed. If the row number is negative, all rows for the specified column are multiplied by the value. Similarly for columns if the column number is negative.

(c) Cards containing changes to data. These are required for all of the arrays that have their flags set to 1, 3, or 4.

For array 10, changes are read in the format (16 (I3, 1X, A1)), i.e., in sets of (row number, value), 16 pairs per card.

For all the other arrays the format used is (5 (2I3, F9.0)), i.e., sets of (row number, column number, value), 5 sets per card.

A card with 999 in columns 1, 2 and 3 terminates the card data for changes to an array. The 999 card is omitted if the flag is set to 2 (to denote that the array is to be read from TAPE 4).

The data cards describing changes to the arrays are read in the following order:

1-3, 10, 4-9, 11-21.

Array 10 appears early in the input stream because it is used in calculating arrays 4 and 5.

Not all arrays have to be changed in one run. Each array is independent of the others in this respect, with the following exceptions. If either 4 or 5 is to be changed, then both 4 and 5 should be changed, i.e., flags 4 and 5 set, and the appropriate data read from the update file. If 10 is changed, then 4 and 5 should also be changed, in the same manner.

6. Special data for arrays 2, 4 and 5 (optional).

Subroutine EXPT (for array 2) or PRICES (for arrays 4, 5) is called if the flag has a value of 4.

Subroutine EXPT (for array 2) - calculates the exports vector inputs:

(1) Base year values of exports, and

(2) Growth rates to convert the above into SNAPSHOT year exports.

The above input is read from TAPE 7 in card images, format (10F8.0). Default values are available in deck GEXPT of UPDATE file AS199OSNAPUPDATE.

Composition of deck GEXPT:

GEXPT. 2, 12 - Base year exports ($M),

GEXPT.13, 23 - Growth rates in exports between the base year and the export year, expressed as percentages.

Subroutine PRICES (for arrays 4, 5) - calculates import and export prices inputs:

(1) Tariff rates for the base year (default values are on UPDATE deck TAR).

(2) DFX - indicator N, M, or E (array 10) N, M and E denote that the industry is classified as non-traded, import competing or export, respectively.

(3) Growth rates for prices (default values are on UPDATE deck GPRICE).

Growth rates in world prices are read from TAPE 5. Default values are available on UPDATE deck GPRICE. (See the use of TAPE 5 and GPRICE in the sample deck.) DFX is read from TAPE 1 in the main program and can be subject to corrections only before it is passed to subroutine PRICES.

Composition of deck TAR:

TAR. 2, 12 - Base year tariff rates.

Composition of deck GPRICE:

GPRICE. 2, 12 - Growth rates in export and import prices between the base year and the snapshot year.

Output from SNAP1

1. Data file SNAPFIXED, written on TAPE 2 in SUPERPASSION format (see Appendix 1).

2. Printer output:

   (a) Component values of all 13 vectors are printed. The word "changed" is printed under the titles of the arrays that have been changed.

   (b) LMAT and QMAT arrays are printed in full.

   (c) Row and column totals of all other arrays, i.e., the large 2-dimensional arrays, are printed.

In this example some values in array 2 are replaced (flag = 1), all of array 3 is replaced (flag = 3) and arrays 4 and 5 are recalculated (flags = 4).

Arrays 13, 15 and 16 are totally replaced from TAPE 4. The formation of TAPE 4 is shown in this example in the SCOPE control cards (from FUSE. to REWIND(TAPE4)).

Sample deck for SNAP1 (*EOS means end of section card 7/8/9)
(*EOS means end of information card 6/7/8/9)

```
SNAP1.
NDFILE(2)
GETSET(DTB3006)
FUSE.
ATTACH (TAPE9,ASVALKMAT,ID=DTBIAS,SN=DTB3006,PW=*****)
COPYR(TAPE9,TAPE4,2)
DATA(TAPE4)
RETURN(TAPE9)
ATTACH(TAPE9,ASVALAMAT,ID=DTBIAS,SN=DTB3006,PW=*****)
COPYR(TAPE9,TAPE4,2)
DATA(TAPE4)
RETURN(TAPE9)
```

```
ATTACH (TAPE9,ASVALLMAT,ID=DTBIAS,SN=DTB3006,PW=*****)
COPYR(TAPE9,TAPE4,2)
DATA(TAPE4)
RETURN(TAPE9)
REWIND(TAPE4)
ATTACH (OLDPL,AS1990SNAPUPDATE,ID=DTBIAS,SN=DTB3006,PW=*****)
UPDATE(D,E,N,L=A124)
FTN(I)
COPYS(COMPILE,TAPE5)
COPYS(COMPILE,TAPE7)
REWIND(TAPE5,TAPE7)
REQUEST(TAPE2,*PF)
ATTACH (TAPE1,ASSTDSNAPFIXED,ID=DTBIAS,SN=DTB3006,PW=*****)
MAP(PART)
LGO.
CATALOG(TAPE2,***** SNAPFIXED,ID=******,RP=30)
*EOS
*ID GR1
*I SNAP1.461
C        22.   DFT1      *****    T1 MATRIX
*I SNAP1.462
      READ (1)  CODE
      READ (1)  NR, NC, NAME, ((T1  (I,J),J=1,NC),I=1,NR)
      IF (IL(22).EQ.0)  GO TO 522
      IF (IL(22).NE.2)  GO TO 622
      READ (4)  DUM
      READ (4)  NR, NC, NAME, ((T1  (I,J),J=1,NC),I=1,NR)
      GO TO 522
  622 CONTINUE
      CALL  SCAL (T1  ,IR,IC,NN,NN)
      IL(22) = 1
  522 CONTINUE
      WRITE (2)  CODE
      WRITE (2)  NR, NC, NAME, ((T1  (I,J),J=1,NC),I=1,NR)
C
C
*C SNAP1.PRICET
*C TOT.WIT
*C TAR
*C GPRICE
*C GEXPT
*EOS
  110    9    7   19
 1344        2 22
 44  1     21.7 47  1     155.8 49  1      77.2 52  1      22.5 65  1     368.5
 69  1     61.1 70  1      50.5 71  1      30.1 72  1      92.6 73  1      45.7
 74  1     65.2 75  1     341.4 78  1      23.2 80  1      89.3
999
  1  1    .0038  2  1     .0006  3  1     .0021  4  1        0  5  1         0
  6  1    .0816  7  1        0  8  1     .0290  9  1     .0584 10  1     .0003
 11  1    .0025 12  1     .1704 13  1     .1318 14  1        0 15  1     .0040
 16  1    .0108 17  1     .0795 18  1     .2203 19  1     .0082 20  1     .0132
 21  1    .1251 22  1     .0692 23  1     .0049 24  1     .0016 25  1     .5867
 26  1    .1153 27  1     .1233 28  1    1.1718 29  1     .9457 30  1     .0728
 31  1    .0014 32  1     .2893 33  1     .3465 34  1     .0985 35  1     .1064
 36  1    .2001 37  1     .2130 38  1     .1255 39  1     .0526 40  1     .0348
```

```
 41  1    .5651 42  1     .0584 43  1     .0750 44  1     .1846 45  1     .0412
 46  1    .0518 47  1     .5497 48  1     .0413 49  1     .2391 50  1     .0503
 51  1    .0734 52  1     .4472 53  1     .2295 54  1     .3574 55  1     .2175
 56  1    .0288 57  1     .0000 58  1     .0010 59  1     .1273 60  1     .1008
 61  1    .0265 62  1     .0358 63  1     .0141 64  1     .2026 65  1     .2239
 66  1    .0599 67  1     .0523 68  1     .7112 69  1    1.2724 70  1     .5066
 71  1    .1487 72  1     .3056 73  1     .3344 74  1     .4728 75  1     .6121
 76  1    .1960 77  1     .2306 78  1     .2229 79  1     .1210 80  1     .5487
 81  1       0 82  1        0 83  1        0 84  1        0 85  1        0
 86  1       0 87  1        0 88  1        0 89  1        0 90  1        0
 91  1    .0099 92  1     .1534 93  1     .3347 94  1     .0212 95  1        0
 96  1    .0043 97  1     .0066 98  1     .0538 99  1     .0265100  1        0
101  1    .0000102  1     .0000103  1        0104  1        0105  1        0
106  1    .0691107  1     .0021108  1     .0045109  1     .0000110  1  1000000
 -1  1     1.53
999
999
999
*EOI
```

4.2 Program SNAP2 (creates file SNAPFINAL)

This program performs algebraic manipulations on arrays which were produced by program SNAP1.

Sample deck for program SNAP2

This job follows automatically after SNAP1 and requires little action by the user. Normally it will be necessary to change only DAT2.2 to ensure that the 4 numbers are consistent with those used in SNAP1.

```
SNAP2.
NDFILE(2)
GETSET(DTB3006)
REQUEST(TAPE2,*PF)
ATTACH(OLDPL,AS1990SNAPUPDATE,ID=DTBIAS,SN=DTB3006,PW=******)
UPDATE(D,E,N,L=A124)
ATTACH(TAPE1,******SNAPFIXED,ID= ******)
ATTACH(ASLIB,AS1990SNAPLIB,ID=DTBIAS,SN=DTB3006,PW=******)
LIBRARY(*,ASLIB)
SNAP2(COMPILE)
CATALOG(TAPE2,****** SNAPFINAL,ID=****** ,RP=**)
*EOS
*ID SN2
*D DAT2.2
  110    9    7   19
*C DAT2
EOI
```

The inputs to SNAP2 are:

(1) file SNAPFIXED on TAPE1, and

(2) card data from deck DAT2 on UPDATE file.

The composition of deck DAT2 is

DAT2.2 N, H, G, Y, in format 4I5,

where N = number of industries,
H = number of occupations,
G = number of consumer goods, and
Y = number of years from base year to snapshot year.

These numbers should be consistent with those used in SNAP1. DAT2.3, 30 extra blank space name cards, one per variable, in format 12A6.

The outputs from SNAP2 are on file SNAPFINAL on TAPE2, written in SUPERPASSION format. It contains 28 arrays, in the following order:

DFIMPS, DFEXP, DFGE, DFEXPP, DFPEE, DFPEEB, DFRWR, DFLMAT, DFWL, DFLABI, DFPM, DFPMT, DFQ, DFDEP, DFKBAS, DFINKO, DFINTK, DFKRRR, DFTE, DFTC, DFP, DFX, DFT1A, DFT2K, DFKR, DFKNIA, DFIA, DFKMAT. (See Appendix 2 for a description of these arrays.)

4.3 Program SNPCOVH (creates file SNAPCONSVH)

This job is concerned primarily with the household consumption specification.[1] It calculates the consumption parameters, linearizes the utility function, and assembles miscellaneous consumption data in preparation for the solution program. File SNAPCONSVH alao contains some other data, namely the size of the workforce and parameters governing the speed of adjustment in certain adjustment rules.

SNPCOVH consists of two FORTRAN programs, CONPAR and VANDH, where CONPAR must be run first. CONPAR estimates the consumption parameters. VANDH performs algebraic manipulations on the output of CONPAR (on TAPE1), including formation of the parameters V and H, otherwise known as DRV and DRHM.

The following example of job SNPCOVH shows that the data for both programs is generated from the UPDATE file.

Deck DCNRW is the data for program CONPAR

Deck DVHRW is the data for program VANDH.

The output of the first program is passed to the second program by means of TAPE1. The output of the second program is data file SNAPCONSVH which is written in binary write on TAPE2. There is also a lineprinter output from each programme.

In this example modifications are made to the following sets of data:

(a) number of households,
(b) workforce, and
(c) ranges of values over which the consumption functions are to be linearized.

---

1. See Harrower (1977) and Williams, Vincent and Strzelecki (1978) for details.

```
SNPCOVH.
NDFILE(2)
GETSET(DTB3006)
REQUEST(TAPE2,*PF)
ATTACH (OLDPL, AS1990SNAPUPDATE,ID=DTBIAS,SN=DTB3006,PW=*****)
UPDATE(D,E,N,L=A124)
ATTACH(LIB,AS1990SNAPLIB,ID=DTBIAS,SN=DTB3006,PW=*****)
LIBRARY(*,LIB)
COMPAR(COMPILE)
REWIND(TAPE1)
VANDH(COMPILE)
CATALOG(TAPE2,******SNAPCONSVH,ID=******,RP=**)
*EOS
*ID C01
*D DCNRW.3
   0.587    0.822    1.267    0.466    0.713    0.285    0.280    0.559    0.510
```

(Number of households (million) by household type in the snapshot year,

1990/91.)

```
*D DVHRW.4.11
79000000.0      Workforce to be employed in the snapshot year (1990/91)
      4000      40000
      1200      12000
      1200      12000
      3200      30000
      1200      20000
      2500      20000
      1200      20000
*C DCNRW
*C DVHRW
*EOI
```

Range of values for which the consumption functions are to be linearized, i.e., Australia wide. Consumption of good 1 is assumed to lie in the range of $4,000 million to $40,000 million in the snapshot year in 1971/72 prices.

In the following description of the compositions of decks and files, M, G, LT, MT, and GT are used in the dimensioning of the arrays and have the following meaning:

- M is the number of consumer groups (9 for the standard set of data),
- G is the number of consumer goods (7),
- LT is the number of points on a consumption curve (11),
- MT = M + 1, and
- GT = G + 1.

Composition of deck DCNRW (standard data input to program CONPAR)

The formats are (10F8.0) for numeric data and (7A10) for alphanumeric data.

DCNRW.2	M,G format (1215)
DCNRW.3	HSEHLD,I=I,M Number of households in millions by household type in the snapshot year (1990/91).
DCNRW.4,10	AI(I,J),I=1,M;J=1,G Engel intercepts in 1974/75 prices (see Williams (February 1978), p.20).
DCNRW.11	CPI(J),J=1,G Price index numbers to convert Engel intercepts to base year (1971/72) units. (See Williams, Vincent and Strzelecki (1978), p.2.)
DCNRW.12,18	BI(I,J),I=1,M;J=1,G Matrix of marginal budget shares (see Williams (February 1978), p.13).
DCNRW.19	B Ratio of national personal disposable income to GNP.
DCNRW.20	MY(I),I=1,M Mean income.
DCNRW.21	SK(J),J=1,GT Commodity scaling factors, k (see Williams (May 1978), Table 9, p.17).
DCNRW.22	GI(I),I=1,M Total subsistence expenditure ($).
DCNRW.23	TCE(I),I=1,M Total consumption expenditure ($).
DCNRW.24	SL(I),I=1,M Consumption scaling factors, ℓ (see Williams (May 1978), Table 10, p.29).
DCNRW.25	ASP(I),I=1,M Average propensity to save.
DCNRW.26,27	GRPS(I),I=1,MT (numeric) headings for consumer groups.
DCNRW.28,29	COMM(I),J=1,GT Titles for commodities.
DCNRW.30	CPIGRP(J),J=1,G Consumer price index titles.
DCNRW.31,34	IHED(I,J),I=1,4 4 headings (1 per card).
DCNRW.35	CSRE(I),I=1,M Share of total consumption in each consumer group.
DCNRW.36	ANAC Level of total consumption ($ million).
DCNRW.37,45	IXA(M,K),M=1,7 9 headings (1 per card).

Composition of deck DVHRW (standard data input to program VANDH)

DVHRW.2 M,G,LT in format 3I5,

where M = number of consumer groups (9),

G = number of consumer goods (7), and

LT = number of points on a consumption curve (utility function).

DVHRW.3 DFONEL,DFTWOL,DFTHRL in format 3I5,

where DFONEL = λ_1, DFTWOL = λ_2, and DFTHRL = λ_3 are adjustment parameters described in Harrower (1977), Ch. 3, eqns. (1) (p.4) and (3) (p.5).

DVHRW.4 DFWORK, number of persons to be employed in the snapshot year (1990/1991), in format F8.0.

DVHRW.5,11 (DRH(I,1),DRH(I,11)), I=1,G, in the format (2F10.0), one card for each commodity (total number of cards = G). These define the range of values for which consumption function is to be linearised.

Composition of TAPE1 (output from CONPAR, input to VANDH)

(DFSA, DFSAT), DFHSHD, VIW, DFGAMB, DFGAMC, DFGAMD, DFBIJ. All these variables are written with binary writes, where brackets denote one binary write statement for a group of variables. Descriptions of these variables are given in Appendix 2.

Contents of file SNAPCONSVH (TAPE2)

This file is written using binary write statements, one statement for each of the following 11 groups of variables.

(DFONEL,DFWOL,DFTHRC),DFWORK,(DFSA,DFSAT),DFHSHD,VIW,DFGAMB,DFGAMC,DFGAMD, DFBIJ,DRHM,DRV. Descriptions of these variables are given in Appendix 2.

4.4 Program SNPPROG (solution program)

This job produces the solution to the non-linear problem defined by the input data contained in the files SNAPFINAL and SNAPCONSVH. It consists of (1) a FORTRAN program, SNPPROG, that creates the LP matrix for APEX (an LP package provided by CDC) and reads the solution file from APEX, and (2) APEX. SNPPROG is used iteratively with APEX until a desirable solution is achieved or until the maximum number of iterations (N) is reached.

The basis calculated in the first iteration of APEX is saved and then fed into the second iteration, and so on, i.e., the basis computed in iteration N-1 is used as the starting point for iteration N.

Once the files SNAPFINAL and SNAPCONSVH have been created, preparation of the deck for the solution program SNPPROG is essentially a mechanical operation with little user choice. Apart from ensuring that the correct files are attached, the user has only to:

(a) set the maximum number of iterations (see LET(N=20)),

(b) set the dimensions of N, G, T, etc., on card LUIN.2 to be consistent with those used to prepare the input files, and

(c) perhaps adjust the convergence parameters on cards LUIN.3 and 4.

The output from SNPPROG consists of:

(a) details of final iteration printed on the line printer, and

(b) details of all iterations on file SNPOUT, which can be printed using job SNAPOUT (see following section).

Sample deck for SNPPROG

In this example, the maximum number of iterations is set to 20, and the default values are used for the convergence parameters. This example shows that SNAPFINAL and SNAPCONSVH exist as permanent files while

the other data inputs, LUNACT, LUIN and LUNAME are generated from the update file.

```
SNPPROG(T777,P1,MS140000)
NDFILE(2)
COMMENT.    ************************************************************
COMMENT.
COMMENT.              S N A P S H O T    1 9 9 0
COMMENT.
COMMENT.    ************************************************************
GETSET(DTB3006)
REQUEST(T,*PF)
ATTACH (OLDPL,AS1990SNAPUPDATE,ID=DTBIAS,SN=DTB3006,PW=******)
UPDATE(D,K,N,L=A124)
COPYS(COMPILE,LUNACT)
COPYS(COMPILE,LUIN)
COPYS(COMPILE,LUNAME)
ATTACH(LIB,AS1990SNAPLIB,ID=DTBIAS,SN=DTB3006,PW=******)
LIBRARY,LIB.
LIBLOAD,LIB,SNPPROG.
NOGO(SNAP)
RETURN,LIB.
ATTACH,FUSE.
LIBRARY,FUSE.
LET(N=20)      MAXIMUM NUMBER OF ITERATIONS, SET BY USER
LET(E=0)
LET(T=1)                                      SET BY USER
REWIND(LUNACT,LUIN,LUNAME)
ATTACH(APEX,APEXIII,ID=PRDLIB)
ATTACH(LUNCVH,***** SNAPCONSVH,ID=******)
ATTACH (LUNFD,***** SNAPFINAL,ID=******)
SNAP(LUNACT,LUNFD,LUNLPM,LUNLPS,LUNSAV,LUNAME,LUNCVH,LUIN,LUOUT)
IF(E,GE,2,TEST)
RFL,140000.
APEX,SOLVE=LUNLPM,MAX,IG,SP,O=LUNLPS,SOF=LIST,YSB.
REWIND,LUNLPS.
REDUCE.
RETURN,TAPE3.
TAG(LOOP)
SNAP(LUNACT,LUNFD,LUNLPM,LUNLPS,LUNSAV,LUNAME,LUNCVH,LUIN,LUOUT)
REWIND(OUTPUT)
COPYP(OUTPUT,T)
IF(E,GE,2,TEST)
REWIND(OUTPUT,TAPE4)
COPY(TAPE4,TAPE3)
REWIND(TAPE3,TAPE4)
RETURN,TAPE4.
RFL,140000.
APEX,SOLVE=LUNLPM,MAX,SP,O=LUNLPS,SOF=LIST,YSB,INB.
REDUCE.
```

```
REWIND(TAPE3)
GOTO(LOOP)
TAG(TEST)
IF(E,EQ,2,END)
COMMENT,**************  ERRORS  ABOVE  ****************************************
TAG(END)
EXIT(U)
CATALOG(T,***** SNPOUT,ID=***** ,RP=**)
*EOS
*ID SC1
*D LUIN.2
    9  110      9     10      11      7      19
*C LUNACT
*C LUIN
*C ITERDT
*C NAMES
*EOI
```

Contents of the update decks LUNACT, LUIN, ITERDT and NAMES

It is anticipated that, in normal usage, only the contents of LUIN will be of relevance to the user. The contents of NAMES and ITERDT will be important if the user changes any of the dimensions. LUNACT is included here only to demonstrate the form of the "action commands".

Contents of deck LUIN

LUIN.2 variables defining dimensions N, G, T, LT, M, H, T, in the format (7I5), and

LUIN.3,4, DFHDEL, in the format (10F8.0) which are 20 convergence parameters (see Harrower (1977), p.21). These parameters are unlikely to require any changes.

List of cards in LUIN :

```
*DECK LUIN                                                                  LUIN  1
    9  110      9  10   11   7   19                                        LUIN  2
     .20      .10    .07    .05    .04    .46    .02    .02    .02    .02 LUIN  3
     .02      .02    .02    .02    .02    .02    .02    .02    .02    .02 LUIN  4
                                                                           LUIN  5
*WEOR
```

Contents of deck ITERDT (becomes first part of file LUNAME)

Initial values of iterative variables (see Appendix 2).

Card images follow the pattern:

1. Title card, dimension, in the format (7A10, 5X, I5), and
2. String of values, the length of which was specified by dimension, in the format (8F10.0), where the card images are

cards			
	2-16 ,	VICHI(N)	χ^+
	17-31 ,	VIM(N)	M^+
	32-34 ,	VIW(M)	W^+
	35-36 ,	VIBETA(1)	β^+
	37-38 ,	VITHET(1)	θ^+
	39-40 ,	VIN(1)	N^+
	41-55 ,	VIP(N)	P^+

See Figure 4 and Appendix 2.

N.B. There is no *WEOR present at the end of deck ITERDT.

Contents of deck NAMES (becomes second part of file LUNAME)

Row/column names needed for the LP matrix which is input to APEX, card images, all in format (8A10):

cards	2 ,	OBJ (1)
	3-16 ,	PROD (N)
	17 ,	TRABDL (1)
	18 ,	LABOUR (1)
	19 ,	SLAM (G)
	20 ,	TAX (1)
	21-30,	VLLAMB (G*LT)
	31 ,	VLSIGMA (1)
	32-45,	VLIMP (N)
	46-59,	VLXONE (N)
	60-73,	VLXTWO (N)
	74 ,	NBIMP (1)
	75 ,	*WEOR

} See Figure 4 and Appendix 2.

LUNACT, LUIN and NAMES are terminated by *WEOR (END-OF-SECTION mark), while ITERDT is not terminated in this way. Thus ITERDT and NAMES are later read from the same logical unit, LUNAME.

SNAPOUT

Only the final iteration plus some extra variables calculated after convergence has been achieved, are printed out by SNPPROG.

A print image of each iteration is catalogued under the name SNPOUT. If the job terminates unsuccessfully it may be desirable to print some or all of the preceding iterations. The following examples show how to do it.

Example 1

```
SNAPOUT.
NDFILE(2)
ATTACH(A,***** SNPOUT,ID=******)
COPY(A,OUTPUT)
*EOI
```

This example shows how to print all iterations.

Example 2

```
SNAPOUT.
NDFILE(2)
ATTACH(A,******SNPOUT,ID=******)
COPYS(A,B,7)
COPYS(A, ,2)
```

In this example the first seven iterations are skipped and the next two (8 and 9) are printed.

4.5 Program SNPMAT

Program SNPMAT has been provided in order to print some or all of the arrays present on SNAPSHOT data files, written in SUPERPASSION format.

The file containing the arrays to be printed has to be on TAPE1.

An additional card input is necessary to indicate which arrays on the file are to be printed. For that purpose a card containing: IPOS, IR1, IR2, IC1, IC2 is read in the format (5I5).

IPOS is the position of the array on the file. If IPOS = 99999 all arrays are printed.

Data is printed between rows IR1 to IR2 inclusive and between columns IC1 to IC2 inclusive. If the last four are left blank all rows and columns are printed. If any of the four exceeds the dimensions of an array they are made equal to those dimensions.

Only non-zero rows are printed. Row and column totals are computed and can be printed if requested. For a full printout of an array the totals are automatically included. Otherwise IR2 and/or IC2 can be set to row dimension plus one and/or column dimension plus one respectively (i.e., outside the dimension of an array). It should be noted that only a range of rows or columns can be printed. Thus rows 70, 85, 97 cannot be printed in one job unless a range of 70 to 97 is requested, which would also give all the extra unwanted information. Moreover totals can only be included as part of the range specifications. If rows 90, 91, 92, and the row total are wanted then two jobs must be run.

Example 1

In this example, arrays 13 and 20 from SNAPFIXED are printed in full, while columns 109 to 111 of array 21 are printed for all the rows (1 to 110). It should be noted that column 111 in this case represents row totals.

```
SNPMAT.
NDFILE(2)
COMMENT.
COMMENT.          S N A P S H O T    1 9 9 0
GETSET(DTB 3006)
ATTACH (TAPE1,******SNAPFIXED,ID=****)
ATTACH(ASLIB,AS1990SNAPLIB,ID=DIAXAS,SN=DTB3006,PW=*****)
LIBRARY(*,ASLIB)
SNPMAT.
*EOS
   13
   20
   21    1  110  109  111
*EOI
```

Example 2

This example illustrates how a complete listing of the data file SNAPFIXED can be obtained.

```
SNPMAT.
NDFILE(2)
COMMENT.
COMMENT.          S N A P S H O T    1 9 9 0
COMMENT.
GETSET(DTB 3006)
ATTACH (TAPE1,***** SNAPFIXED,ID=*****)
ATTACH(ASLIB,AS1990SNAPLIB,ID=DIAXAS,SN=DTB3006,PW=*****)
LIBRARY(*,ASLIB)
SNPMAT.
*EOS
99999
*EOI
```

Appendix 1 : SUPERPASSION Format

The following is an example of how to write an array A, of dimensions NR by NC, on file 2 in SUPERPASSION format using a FORTRAN program.

```
      DIMENSION NAME(12)
          .
          .
          .
          .
  100 FORMAT(12A6)
      READ(60,100) NAME
          .
          .
          .
          .
      WRITE(2) CODE
      WRITE(2) NR,NC,NAME((A(I,J),J=1,NC),I=1,NR)
```

SUPERPASSION format consists of two binary records for each array. The first record contains CODE (a real number), which is usually the position of the array on a file. The second record consists of: number of rows (NR), number of columns (NC), NAME in the format 12A6, and finally the array itself with its columns written first, i.e., in the inner DO loop. Note that the elements of A are stored row by row rather than in the usual manner of column by column.

N.B. There is a version of SUPERPASSION which can be used to read and write arrays on a file in both sequential mode and random access mode. In SNAPSHOT all the SUPERPASSION read and write statements are sequential. Hence if a user creates a SNAPSHOT file using the random access version of SUPERPASSION the write statements must occur in the correct sequence.

Appendix 2 : SNAPSHOT Notation/Nomenclature

For computing purposes the SNAPSHOT notation has been classified into the two broad groups of data and variables. These two groups have been subdivided as follows :

1. Data
   - Fixed data
   - Re-estimated data

2. Variables
   - Iterative variables
   - LP solution variables
   - Dual variables from the LP
   - Derived variables
   - Final SNAPSHOT variables .

The letters N, G, T, M, LT and H represent the dimensions of variables and data in the problem as follows (these dimensions can be changed in subsequent runs of the model):

N = 110 , the number of industry groups,

G = 7 , the number of consumer goods,

T = 10 , the number of segments of a consumption curve,

LT = T + 1 , the corresponding points on a consumption curve,

M = 9 , the number of consumer groups, and

H = 9 , the number of occupational groups .

A detailed listing of the notation is shown in Table A2.1.

Table A2.1 The SNAPSHOT NOTATION

Symbol	Description of Notation	Variable Name, if Used in Program	Size
	1. D A T A		
	(a) Fixed Data		
s_i	consumer group i's average propensity to save out of disposable income	DFSAV	M
α_i	share of GNP which is disposable income for group i	DFALPH	M
$\overline{K(0)}$	industry levels of capital stock in the base year	DFKBAS	N
t	number of years of the snapshot period	DFTIM	1
η	industry specific depreciation rates applicable to the industry capital stocks, K(t), over the t^{th} year	DFDEP	N
K	capital matrix in the snapshot year, K_{ij} is the input of good i required to create a unit of capital stock for industry j	DFKMAT	NxN
T_2	ad valorem taxes on creation of capital stock	DFT2	NxN
$\bar{r}$	relative rates of return to capital required to induce investment in each industry	DFKRRR	N
$\bar{E}$	exports of commodities	DFEXP	N
t_E	ad valorem taxes (net of subsidies) on exports	DFTE	N
γ	import shares of the domestic markets [1]	DFIMPS	N
t_c	ad valorem taxes on consumption	DFTC	N
$\overline{p_e}$	export prices (f.o.b.) in foreign currency	DFEXPP	N
$\overline{p_m}$	import prices (c.i.f.) in foreign currency	DFIMPP	N

contd.

1 γ = (competing imports + duty)/(domestic production)

Table A2.1 contd.

Symbol	Description of Notation	Variable Name, if Used in Program	Size
$\bar{\tau}$	ad valorem tariff rates	DFTAR	N
$\bar{B}$	balance of trade deficit in foreign currency	DFBOTF	1
A	input-output coefficients matrix	DFAMAT	NxN
T_1	ad valorem taxes and other costs on intermediate usage	DFT1	NxN
ℓ	labour requirements by occupation and industry per unit of output in the snapshot year	DFLMAT	HxN
$\bar{G}$	government purchases of commodities	DFGPUR	N
$\bar{N}$	total number of people in the workforce in the snapshot year	DFWORK	1
$\bar{w}$	relative wage rates, before taxes, for the various occupational groups	DFRWR	H
Q	transformation matrix between the number of consumer goods and the number of industry classifications	DFQ	NxG
b_{ij}	marginal budget share of good j for consumer group i	DFBIJ	MxG
γ_{ij}	subsistence expenditure for the i^{th} household on good j	DFGAMB	MxG
γ_j	total subsistence expenditure on good j	DFGAMC	G
$\bar{\gamma}_j$	average subsistence expenditure on good j by an individual consumer	DFGAMD	G
N_h	number of households by type in the snapshot year	DFSHD	1
g_c	convergence parameters	DFHDEL	20
λ_1	rate of adjustment of iterative variables	DFONEL	1
λ_2	rate of adjustment of iterative variables	DFTWOL	1
λ_3	rate of adjustment of iterative variables	DFTHRL	1

contd.

Table A2.1 contd.

Symbol	Description of Notation	Variable Name, if used in Program	Size
	(b) Re-estimated Data		
V_{jt}	parameter used in specification of the objective function	DRV	G(T+1)
H_{jt}	parameter used in the determination of the consumption of good j	DRHM	GxG(T+J)
b_j	marginal budget share of good j	DRBJ	G
	2. VARIABLES		
	(a) Iterative Variables		
β^+	absolute rate of return	VIBETA	1
Θ^+	exchange rate	VITHET	1
N^+	workforce (wage units)	VIN	1
W_i^+	reciprocal of the marginal utility of expenditure for the i^{th} consumer group	VIW	M
χ^+	see Dixon (1976)	VICHI	N
M^+	imports	VIM	N
p^+	domestic prices	VIP	N
	(b) LP Solution Variables		
λ	computational variable (consumption relationships)	VLLAMB	G*(T+1)
X_1	computational variable	VLXONE	N
X_2	computational variable	VLXTWO	N
M	imports of commodities	VLIMP	N

contd.

Table A2.1 contd.

Symbol	Description of Notation	Variable Name, if used in Program	Size
	(c) Dual Variables from the LP		
p	commodity prices	VDPRC	N
δ	variable reflecting the absolute level of wages before taxes for the Australian labour force	VDABWG	1
ϕ	excess tariff revenue per unit of imports	VDXSTR	N
Θ	exchange rate ($A per unit for foreign currency)	VDXCHM	1
π	rental prices on capital by industries	VDRPK	N
	(d) Derived Variables		
GNP	gross national product	VCGNP	1
h	average rate of growth of capital in each industry over the t-year snapshot period	VCARGK	N
C_{ij}	consumption of commodity j by consumer group i	VCCONS	MxG
C_j	consumption of commodity j	VCCONJ	G
C_{ni}	consumption by consumer group i of industry good n	VCCONI	NxM
C	aggregate consumption	VCAGGC	1

contd.

Table A2.1 contd.

Symbol	Description of Notation	Variable names, if used in Program	Size
	(e) Final SNAPSHOT Variables		
Z_i	total expenditure of consumer group i	VSZ	M
K(t)	industry levels of capital stock in the snapshot year	VSKSTK	N
K(t+1)	industry levels of capital stock in the year after the snapshot year	VSAKSK	N
J	gross investment by using industries	VSINV	N
I_s	gross investment by supplying industries	VSINVS	N
X	outputs of commodities	VSOUT	N
r	minimum acceptable rates of return by industry	VSMRR	N
β	variable reflecting the absolute rate of return demanded on new capital formation for Australian industry (final value of VIBETA)	VIBETA	1
E	exports of commodities (quantity)	DFEXP	N
ξ	exports tax (ξ_j positive) or subsidy (ξ_j negative)	VSEXFT	N
w	wage rates by occupation before taxes	VSWAGE	H
L	the number of labour units in each occupational group in the snapshot year	VSLAB	H

The following is a list of all relevant deck names, in the order in which they occur on the update file.

ARRAYS.ENDIT - program SNPPROG
ITERDT - data, iterative variables
NAMES - data, LP names
CONRW - program CONAPR
DCNRW - data for CONAPR
VHRW - program VANDH
DVHRW - data for VANDH
LUNACT - data, action commands
LUIN - data for SNPPROG
SNAP1.PRICET - program SNAP1
SNAP2.MAT - program SNAP2
TOT.WIT - subroutines used by SNAP1 and SNAP2
DAT1 - data for SNAP1
DAT2 - data for SNAP2
SNPMAT - program SNPMAT
TAR - data for SNAP1
GPRICE - data for SNAP1
GEXPT - data for SNAP1

Appendix 4 : Files Used in SNAPSHOT

The programs and data are held on three files:

a. a library file AS1990SNAPLIB,ID=DTBIAS,

b. an UPDATE file AS1990SNAPUPDATE,ID=DTBIAS,

c. a data file ASSTDSNAPFIXED,ID=DTBIAS.

These three files are currently held on set DTB3006. The output files from SNAP1, SNAP2 and SNPCOVH are held on SYSTEM. Therefore, in ATTACH statements involving these last three files no SN parameter is required.

AS1990SNAPLIB contains compiled versions of selected programs such as SNAP1, SNAP2, SNPCOVH,SNAP and SNPMAT. If no program alterations are made at the SNAP1 phase it is possible to attach AS1990SNAPLIB in order to run SNAP1, in which case the cards *ID GR1 to *C TOT.WIT (see "sample deck for SNAP1").

AS1990SNAPUPDATE contains the FORTRAN programs, subroutines and their data. This is an UPDATE file and can be amended only through UPDATE commands. An example of this updating method is found in "sample deck for SNAP1."

ASSTDSNAPFIXED is the file containing most of the data required by the SNAPSHOT suite of programs. Amendment of data in this file is by the SNAP1 program. Table 1 lists the contents of this file. The order of the vectors/matrices in the file is important.

Figure 3, "Flow Diagram for SNAPSHOT Computation" gives an indication of the use of these files by the major programs. It should be noted that the updating/amending examples given later in the manual do NOT change the data base permanently. The outputs from SNAP1, SNAP2 and SNPCOVH are amended copies of the data base and are held on SYSTEM. They are therefore subject to retention period restrictions and purging. It is recommended that the names given to these files distinguish them from the data base file names.

REFERENCES

Control Data Cyber 70 Computer Systems Models 72, 73, 74, 76, 6000 Series Computer Systems, 7600 Computer Systems, "APEX-III Reference Manual", Control Data Corporation, U.S.A., 1977.

Control Data Cyber 7600 Computer System, "Update Reference Manual", Control Data Corporation, U.S.A., 1975.

Dixon, Peter B., "A Jointmax Algorithm for the Solution of SNAPSHOT" IMPACT Preliminary Working Paper No. SP-03, Industries Assistance Commission, Melbourne, April 1976.

Dixon, Peter B., John D. Harrower and Alan A. Powell, "SNAPSHOT, A long-term Economy-wide Model of Australia : Preliminary Outline", IMPACT Preliminary Working Paper No. SP-01, Industries Assistance Commission, Melbourne, February 1976.

Dixon, Peter B., John D. Harrower and David P. Vincent, "Validation of the SNAPSHOT Model", IMPACT Preliminary Working Paper No. SP-12, Industries Assistance Commission, Melbourne, July 1978.

Dixon, P.B., and D.P. Vincent, "The SNAPSHOT Model : Underlying Theory and an Application to the Study of the Implications of Technical Change in Australia to 1990", IMPACT Preliminary Working Paper No. SP-14, Industries Assistance Commission, Melbourne, October 1979.

Edington, A.C., and John D. Harrower, "Solution of Non-Linear Problems by Iterative Use of a Linear Programming Package", IMPACT Preliminary Working Paper No. SP-08, Industries Assistance Commission, Melbourne, January 1977.

Harrower, John D., "Jointmax Algorithm : Solution Tests and Adjustment Rules", Research Memorandum, IMPACT Project, Industries Assistance Commission, Melbourne, August 1977 (mimeo).

Harrower, John D., and David Vincent, "Incorporation of Taxes, Margins and Non-Competing Imports into SNAPSHOT", Research Memorandum, IMPACT Project, Industries Assistance Commission, Melbourne, December 1977 (mimeo).

Harvard Economics Research Project, "Passion Manual", U.S.A., 1966.

Williams, Ross A., "Demographic Effects on Consumption Patterns in Australia : A Preliminary Analysis of the ABS 1974-75 Household Expenditure Survey", IMPACT Preliminary Working Paper No. SP-11, Industries Assistance Commission, Melbourne, February 1978.

Williams, Ross A. "The Use of disaggregated Cross-Section Data in Explaining Shifts in Australian Consumer Demand Patterns Over Time", IMPACT Preliminary Working Paper No. SP-13, Industries Assistance Commission, Melbourne, May 1978.

Williams, Ross, David Vincent and Alexandra Strzelecki, "Predictions of Consumer Expenditures for 1971-72 using SNAPSHOT", Research Memorandum, IMPACT Project, Industries Assistance Commission, Melbourne, May 1978 (mimeo).