DEPARTMENT OF MATHEMATICS

Final Report to the Ministry of State for
Science and Technology
on
THE QUEEN'S MATHEMATICS DEPARTMENT
POPULATION MODEL
by

J.H. Davis and J.H. Verner
Queen's University, Kingston, Ontario

Queen's Mathematical Preprints No. 1975-33
VOLUME II

Queen's University

AT KINGSTON
ONTARIO, CANADA

Final Report to the Ministry of State for
Science and Technology

on

THE QUEEN'S MATHEMATICS DEPARTMENT POPULATION MODEL

by

J.H. Davis and J.H. Verner
Queen's University, Kingston, Ontario

Queen's Mathematical Preprints No. 1975-33

VOLUME II

APPENDIX E: Program Listings and Comments

This Appendix consists of the computer program listings for the programs used in conjunction with this project.

The large amounts of data which must be handled in connection with this work have required the use of file systems in the programs. For this reason, facsimiles of the appropriate job control language (work flow language) cards are included with each program.

The program listings are heavily documented; also included are supplementary comments concerning implementation of the basic algorithms developed for use in this work. Most of the programs return considerably more printed output than has been included here. An amount of output sufficient to indicate the nature and format of the returned data has been retained with each program.

Appendix E: Outline

Index of Punched Cards

Box	Data	Programs
1	CSRS	
2	CSNF	
3	BS	
4	SSNF	
5	DSCN	
6	XXIT	
7	XXIT	
8		
9		2.4, 2.5, 2.6
10		3.1, 3.2, 3.3
11		3.4, 3.5, 3.6
12		3.7, PLOTT, 4.1
13		4.2, 4.3, 4.4, 4.5

1. RAW DATA

1.0 Introduction.

Primarily to provide portable programs the computer language adopted for this study was FORTRAN. For adequate accuracy in some of the programs, it was found desirable to use the Burroughs 6700 computer which provides much higher accuracy in arithmetic processes than does the 360 series of IBM computers. Hence job control language (JCL) given is that appropriate for the B6700. Occasionally extensions to standard FORTRAN are used to facilitate input/output, but these could be easily modified by a competent FORTRAN programmer.

In the Interim Report [3] computer programs for several basic procedures were presented. These procedures had been developed under the assumptions that data would be available in a particular format. During the course of extracting data relevant to the evolution of Canadian population, considerable processing of more data was required for the use of these procedures. Furthermore, the computer programs had to be modified, and extended to accommodate the full scope of the model proposed.

This appendix to the final report describes the data obtained, and its incorporation into the procedures. Further it includes source listings with data and selected output (excepting only a program for plotting three-dimensional profiles) needed for processing additional data as it becomes available.

The algorithms used were among the most accurate and reliable known to the authors; parallel programming with other algorithms did not yield more promising results. While the numerical techniques may be improved upon through intensive investigation of each of the algorithms used and through improved programming, the authors expect that inaccuracies in the data currently available would inhibit improvement of the final results. Certainly the limited experimentation by the authors failed to indicate significant changes in the results.

To implement the model proposed in the Interim Report, estimations of parameters characterizing evolution of a population from historical data required three large data sets:

For the evolution of fertility:

1. <u>Fertility</u>: age-specific fertility as a function of time: in particular, the rate at which females of age x give birth at time t for each value of x and t in years.

and for the evolution of population:

2. <u>Population Density</u>: age-, income- specific population density as a function of time: in particular, the number of people at age x with an income of s at time t ; in practice this data might be estimated from the number of people between ages x and x + 1 coming between s and s + ds at time t for each value of x and t in years,

and ds = $1000 (for example); and

3. <u>Mortality</u>: age-specific death rates as a function of time.

As well, certain other sets of data forming economic indicators were required for estimating the dynamics of the evolution, and these are discussed in section 4.1.

1.1 Fertility Data

Data available in annual issues of Vital Statistics [23] provides fertility rates for five year age groups. Annual data of this type is available only for a limited period, and data in single years of age was required if accurate estimates of the changes over time were to be obtained.

To obtain this data in the smaller age groupings, the authors communicated directly with Statistics Canada. This ministry provided the following data:

1. (a) for each census year (1956, 1961, 1966, 1971), the numbers of femalses (and males) of age x (in single years) at June 1, Canada and the Provinces;

   (b) for each non-census year (during 1957-1972) inter-censal revised estimates of the numbers of females (and males) of age x (in single years), Canada and the Provinces;

2. for each year (1956-1972) the numbers of live births to mothers between ages x and x + 1 in Canada excluding Newfoundland.

This data defined* the age-specific fertility as

f(x,t) = number of live births to mothers between ages x and x + 1 during year t in Canada excluding Newfoundland

numbers of females of age x at time t in Canada excluding Newfoundland

To create computer files of this data, handwritten tables were transcribed via punched cards, and checked with totals provided in the tables.

For the age-specific distributions of males and females two data-sets were created:

1. Census Singles/Revised Singles: Canada Format:

I4 ,	8X , F9.1 ,	2X , F9.1 ,	1X , F9.1 ,	30X , A6	
Age	Total	Males	Females	Year	'CS' 'RS'

for ages 0-89 in years 1956-66 and 1971-72,

and ages 0-69 in years 1967-1970.

*These births to mothers of unstated age were distributed among those of stated age according to the "stated-age" distribution of mothers.

2. Census Singles/Revised Singles: Newfoundland Format:

I4	8X	F9.1	2X	F9.1	1X	F9.1	30X	A8
Age		Total		Males		Females		Year 'CSNF' 'RSNF'

For the age-specific distribution of births, one data set was created in three different formats:

Birth Singles (excluding Newfoundland)

1. 1956-65

I4 , 8X , F7.0 , 53X , A6

2. 1966-72

I4	7X	F7.0	54X	A6
Age		Births to mothers of given age		Year 'BS'

3. Age unstated

6X	F3.0	1X	1X	F6.0	57X	A6
	Births to mothers of unstated age			Total number of births		Year 'BS'

All of these data sets were combined in Program 1 to give punched (or printed) output of Summary Singles excluding Newfoundland in the format:

F4.0 ,	F9.1 ,	F9.1 ,	F9.1 ,	F9.3 ,	32X ,	A8
Age	Total Population	Males	Females	Births		Year, 'SSNF'

This summary data forms the input for programmes described in the next chapter.

```
¶JOB   ACCT-NUM,VERNER,TIME=60
C      ****************************************************************************
C
C          PROGRAM 1.1 :   CHECK
C
C      ****************************************************************************
```

```
3JOB CHECK; CHARGE BERGHOUT; CLASS 3;
USER 080002/VHJ;
BEGIN
COMPILE CHECK FORTRAN;
FILE FILE9(KIND=PACK,TITLE=DATA/XXCSRS,FILETYPE=7);
FILE FILE10(KIND=PACK,TITLE=DATA/XXCSNF,FILETYPE=7);
DATA
FILE   5=FILE5
FILE   6=FILE6
FILE   7=PUNCH
FILE   9=FILE9,UNIT=DISKPACK
FILE   10=FILE10,UNIT=DISKPACK
```

```
3DATA FILE5
```

```
C       ******************************************************************
C
C       PROGRAM :   1.1
C       TITLE : CHECK
C       PURPOSE : INPUT AND CALCULATE DATA FOR FERTILITY.
C                  CHECKS THE  INPUT DATA FOR ERRORS
C       DATA : CSRS.CSNF.BS
C       OUTPUT : SSNF (PRINTED OR PUNCHED)
C       NOTE : THE DATA FROM THE FILES CSRS.CSNF.AND BS ARE READ IN
C              THE FOLLOWING ORDER
C                  ONE YEAR FROM CSRS
C                  ONE YEAR FROM CSNF
C                  ONE YEAR FROM BS
C              THE PROGRAM WILL READ AND PROCESS DATA FOR THE YEARS
C              1956 TO 1972
C       FILES : FILE5 = BS DATA (FILE5=READER)
C                FILE9=CSRS DATA
C                FILE10=CSNF DATA
C
C       ******************************************************************
        INTEGER YEAR,AGE,NIN(17)
        INTEGER SYEAR
        REAL PDATA(90,4), NFDATA(90,4), BIRTHS(90,4)
        DATA NIN/90,90,90,90,90,90,90,90,90,90,90,70,70,70,70,90,90/
        II = 7
        II = 6
        SYEAR=1955
        DO 9 I = 1,17
        SYEAR=SYEAR+1
           LL=9
           CALL DATAIN(PDATA,NIN(I),4,'     ',0.0,LL,SYEAR)
           LL=10
           CALL DATAIN(NFDATA,NIN(I),4,'     ',0.0,LL,SYEAR)
           LL=5
           CALL BORNI(BIRTHS,12,51,SYEAR,LL)
           DO 1 J = 1,90
                PDATA(J,2) = PDATA(J,2) - NFDATA(J,2)
                PDATA(J,3) = PDATA(J,3) - NFDATA(J,3)
                PDATA(J,4) = PDATA(J,4) - NFDATA(J,4)
        IF (J.EQ.52) WRITE(6,101)
              WRITE(II,100)(PDATA(J,L),L=1,4),BIRTHS(J,2),SYEAR
    1           CONTINUE
  100 FORMAT(F4.0,3F9.1,F9.3,32X,I4,'SSNF')
  101 FORMAT('1',1X)
    9 CONTINUE
        STOP
        END

        SUBROUTINE BORNI(BIRTHS,NS,NN,SYEAR,LL)
C       ******************************************************************
C
C   BIRTHS = THE MATRIX IN WHICH THE DATA IS TO BE STORED
C  NS = THE START POSITION (AGE ) OF THE DATA
C    NN = THE END POSITION (AGE) OF THE DATA
C       SYEAR = THE YEAR OF THE DATA SET
C       LL = THE UNIT # FROM WHICH THE DATA IS TO BE READ
C       CODE IS SET TO BS FOR ALL YEARS
C
C       ******************************************************************
```

```
      REAL BIRTHS(90,4)
      REAL CODE,CODEIN
      INTEGER YEAR,SYEAR
      DATA CODE/'RS'/
      DO 50 I = 1,NS
   50 BIRTHS(I,2) = 0.0
      DO 51 I = NN,90
   51    BIRTHS(I,2) = 0.0
      SUM = 0.0
      DO 1 I = NS,NN
      IF (SYEAR.GT.1965) GO TO 5
      READ(LL,10,END=3)(BIRTHS(I,J),J=1,2),YEAR,CODEIN
      GO TO 6
    5 READ(LL,11,END=3)(BIRTHS(I,J),J=1,2),YEAR,CODEIN
    6 CONTINUE
C    CHECK THE CARD CODE
      IF (YEAR.NE.SYEAR.OR.CODEIN.NE.CODE) GO TO 3
C   3  AL THE BIRHTS
    1 SUM = SUM + BIRTHS(I,2)
      READ(LL,20)TNN,TOTAL,YEAR,CODEIN
      TOTAL = TOTAL - TNN
      IF (SUM.NE.TOTAL) WRITE(6,21)SYEAR,CODE,SUM,TOTAL
      SUM = 0
      DO 2 I = NS,NN
      BIRTHS(I,2) = ( BIRTHS(I,2) * TNN) / TOTAL + BIRTHS(I,2)
    2 SUM = SUM + BIRTHS(I,2)
      WRITE(6,22)YEAR,CODE,TOTAL,TNN
*EXTENSION*    IO-3
      I = NN
    3 CONTINUE
      RETURN
   10 FORMAT(F4.0,8X,F7.0,53X,I4,A2)
   11 FORMAT(F4.0,7X,F7.0,54X,I4,A2)
   20 FORMAT(6X,F3.0,1X,F6.0,56X,I4,A2)
   21 FORMAT('0',' ERROR IN  ',I4,A2,' TOTALS ',F13.5,5X,F13.5)
   22 FORMAT('1',' DATA ',I4,A2,' TOTAL = ',5X,F13.5)
      END

      SUBROUTINE DATAIN(DATAS,ROWS,COLS,NAME,ERRORB,LL,SYEAR)
C     ********************************************************************
C
C     DATAS IS THE DATA TO BE INPUT MATRIX , INPUT BY ROW
C     ROWS IS THE # OF ROWS IN DATAS
C     COLS IS THE # OF COLUMNS IN DATAS
C     NAME IF NOT BLANK THEN DATA WILL BE CHECKED AS IT IS READ
C     ERRORB IS THE MAXIMUM ERROR ALLOWED BEFORE BEING FLAGGED
C     LL IS THE UNIT # FROM WHICH THE DATA IS TO BE READ
C     SYEAR IS THE YEAR OF THE DATA SET
C     CODE (THE EXPECTED CODE) IS SET TO CS FOR CENSUS YEARS AND TO
C     RS OTHERWISE
C
C     ********************************************************************
      REAL DATAS(90,4)
      INTEGER ROWS,COLS,YEAR,SYEAR,LL
      REAL NAME,B/'    '/,ERRORB,ERR,CODE,CODEIN
      CODE='RS'

      IF (MOD(SYEAR,5).EQ.1) CODE='CS'

      IF (NAME.NE.B) GO TO 5
```

```
      READ(LI,100)((DATAS(I,J),J=1,COLS),YEAR,CODEIN,I=1,ROWS)
      IF (YEAR.EQ.SYEAR.AND.CODE.EQ.CODEIN) GO TO 15
      WRITE(6,210)SYEAR,CODE,YEAR,CODEIN
      STOP
    5 DO 10 I = 1,ROWS
      READ(LI,100)(DATAS(I,J),J=1,COLS),YEAR,CODEIN
      IF (YEAR.EQ.SYEAR.AND.CODE.EQ.CODEIN) GO TO 10
      WRITE(6,210) SYEAR,CODE,YEAR,CODEIN,I
      STOP
   10 CONTINUE
C
C     SCALE THE DATA
C
   15 IF(MOD(SYEAR,5).EQ.1) GO TO 16
      DO 30 I = 1,ROWS
          DO 30 J=2,COLS
   30         DATAS(I,J) = DATAS(I,J) * 1000
   16 IF(NAME .EQ. B ) RETURN
      DO 20 I = 1, ROWS
      ERR = DATAS(I,2) - ( DATAS(I,3) + DATAS(I,4) )
      IF ( ABS(ERR) .LT. ERRORB ) GO TO 20
      WRITE(6,200) NAME, ( DATAS(I,J), J=1,4)
   20 CONTINUE
      RETURN
  100 FORMAT(F4.0,8X,F9.1,2X,F9.1,1X,F9.1,30X,I4,A2)
  200 FORMAT(/,'  ',A4,5X,F5.0,3F11.1)
  210 FORMAT(1X,10('* '),'CARD CODES DISAGREE, EXPECTED CODE = ',I4,A2,
     & ' GOT ',I4,A2, ',CARD # ',I2)
      END
```

```
****************************************************************************

DATA FOR
     PROGRAM 1.1 :   CHECK

****************************************************************************
```

DATA 1956RS TOTAL - 436198.00000

0.	401629.0	204917.0	196712.0	0.000	1956SSNF
1.	400163.0	204130.0	196033.0	0.000	1956SSNF
2.	388573.0	198064.0	190509.0	0.000	1956SSNF
3.	371619.0	189639.0	181980.0	0.000	1956SSNF
4.	358205.0	183152.0	175053.0	0.000	1956SSNF
5.	373355.0	189960.0	183395.0	0.000	1956SSNF
6.	363146.0	184789.0	178357.0	0.000	1956SSNF
7.	350992.0	178683.0	172309.0	0.000	1956SSNF
8.	337349.0	171848.0	165501.0	0.000	1956SSNF
9.	322672.0	164502.0	158170.0	0.000	1956SSNF
10.	307176.0	156769.0	150407.0	0.000	1956SSNF
11.	291081.0	148784.0	142297.0	0.000	1956SSNF
12.	276012.0	141150.0	134862.0	1.002	1956SSNF
13.	262882.0	134234.0	128648.0	21.033	1956SSNF
14.	251446.0	128012.0	123434.0	124.192	1956SSNF
15.	240328.0	121984.0	118344.0	584.906	1956SSNF
16.	229538.0	116128.0	113410.0	1958.034	1956SSNF
17.	221712.0	111829.0	109883.0	5070.859	1956SSNF
18.	217935.0	109691.0	108244.0	9501.725	1956SSNF
19.	217128.0	109142.0	107986.0	13907.553	1956SSNF
20.	216869.0	108840.0	108029.0	18731.029	1956SSNF
21.	217344.0	108859.0	108485.0	22738.239	1956SSNF
22.	218918.0	109608.0	109310.0	24773.393	1956SSNF
23.	221414.0	111089.0	110325.0	27164.098	1956SSNF
24.	224547.0	113036.0	111511.0	28278.825	1956SSNF
25.	228071.0	115171.0	112900.0	27940.301	1956SSNF
26.	231894.0	117542.0	114352.0	27090.984	1956SSNF
27.	235152.0	119264.0	115888.0	25138.959	1956SSNF
28.	237345.0	119862.0	117483.0	24870.543	1956SSNF
29.	238621.0	119650.0	118971.0	22515.894	1956SSNF
30.	239852.0	119504.0	120348.0	21774.746	1956SSNF
31.	241123.0	119361.0	121762.0	18902.294	1956SSNF
32.	240530.0	118568.0	121962.0	18436.572	1956SSNF
33.	237304.0	116936.0	120368.0	16330.308	1956SSNF
34.	232323.0	114749.0	117574.0	15034.299	1956SSNF
35.	227139.0	112424.0	114715.0	13223.493	1956SSNF
36.	221301.0	109780.0	111521.0	11660.070	1956SSNF
37.	216369.0	107675.0	108694.0	9231.306	1956SSNF
38.	213232.0	106543.0	106689.0	8311.881	1956SSNF
39.	211116.0	105969.0	105147.0	6567.178	1956SSNF
40.	208427.0	105136.0	103291.0	5209.073	1956SSNF
41.	205599.0	104259.0	101340.0	3749.811	1956SSNF
42.	201817.0	102838.0	98979.0	3046.722	1956SSNF
43.	196567.0	100567.0	96000.0	1938.003	1956SSNF
44.	190351.0	97722.0	92629.0	1202.864	1956SSNF
45.	184247.0	94923.0	89324.0	646.001	1956SSNF
46.	178116.0	92082.0	86034.0	298.463	1956SSNF
47.	171980.0	89176.0	82804.0	116.180	1956SSNF
48.	166001.0	86271.0	79730.0	63.098	1956SSNF
49.	160164.0	83368.0	76796.0	31.048	1956SSNF
50.	154285.0	80412.0	73873.0	13.020	1956SSNF

51.	148313.0	77389.0	70924.0	0.000	1956SSNF
52.	142968.0	74578.0	68390.0	0.000	1956SSNF
53.	138564.0	72096.0	66468.0	0.000	1956SSNF
54.	134786.0	69843.0	64943.0	0.000	1956SSNF
55.	131075.0	67620.0	63455.0	0.000	1956SSNF
56.	127635.0	65521.0	62114.0	0.000	1956SSNF
57.	123860.0	63302.0	60558.0	0.000	1956SSNF
58.	119422.0	60840.0	58582.0	0.000	1956SSNF
59.	114653.0	58283.0	56370.0	0.000	1956SSNF
60.	110103.0	55822.0	54281.0	0.000	1956SSNF
61.	105508.0	53321.0	52187.0	0.000	1956SSNF
62.	101760.0	51362.0	50398.0	0.000	1956SSNF
63.	99307.0	50218.0	49089.0	0.000	1956SSNF
64.	97655.0	49576.0	48079.0	0.000	1956SSNF
65.	95819.0	48837.0	46982.0	0.000	1956SSNF
66.	93967.0	48118.0	45849.0	0.000	1956SSNF
67.	91685.0	47074.0	44611.0	0.000	1956SSNF
68.	88664.0	45467.0	43197.0	0.000	1956SSNF
69.	85067.0	43452.0	41615.0	0.000	1956SSNF
70.	81508.0	41476.0	40032.0	0.000	1956SSNF
71.	78013.0	39525.0	38488.0	0.000	1956SSNF
72.	73690.0	37197.0	36493.0	0.000	1956SSNF
73.	68195.0	34350.0	33845.0	0.000	1956SSNF
74.	61979.0	31190.0	30789.0	0.000	1956SSNF
75.	55826.0	28046.0	27780.0	0.000	1956SSNF
76.	49597.0	24869.0	24728.0	0.000	1956SSNF
77.	43840.0	21904.0	21936.0	0.000	1956SSNF
78.	38920.0	19315.0	19605.0	0.000	1956SSNF
79.	34628.0	17022.0	17606.0	0.000	1956SSNF
80.	30367.0	14756.0	15611.0	0.000	1956SSNF
81.	26242.0	12559.0	13683.0	0.000	1956SSNF
82.	22500.0	10603.0	11897.0	0.000	1956SSNF
83.	19194.0	8942.0	10252.0	0.000	1956SSNF
84.	16277.0	7524.0	8753.0	0.000	1956SSNF
85.	13623.0	6236.0	7387.0	0.000	1956SSNF
86.	11254.0	5090.0	6164.0	0.000	1956SSNF
87.	9176.0	4095.0	5081.0	0.000	1956SSNF
88.	7385.0	3246.0	4139.0	0.000	1956SSNF
89.	5874.0	2535.0	3339.0	0.000	1956SSNF

DATA 1957BS TOTAL - 453778.00000

0.	417900.0	213100.0	204800.0	0.000	1957SSNF
1.	403200.0	205800.0	197400.0	0.000	1957SSNF
2.	405300.0	206900.0	198400.0	0.000	1957SSNF
3.	393900.0	201000.0	192900.0	0.000	1957SSNF
4.	379000.0	194000.0	185000.0	0.000	1957SSNF
5.	382400.0	194800.0	187600.0	0.000	1957SSNF
6.	372600.0	189800.0	182800.0	0.000	1957SSNF
7.	361300.0	184100.0	177200.0	0.000	1957SSNF
8.	349300.0	178100.0	171200.0	0.000	1957SSNF
9.	336800.0	171800.0	165000.0	0.000	1957SSNF
10.	323800.0	165300.0	158500.0	0.000	1957SSNF
11.	310300.0	158600.0	151700.0	0.000	1957SSNF
12.	296600.0	151500.0	145100.0	3.005	1957SSNF
13.	282600.0	144400.0	138200.0	31.049	1957SSNF
14.	268100.0	136600.0	131500.0	137.216	1957SSNF
15.	255000.0	129800.0	125200.0	644.012	1957SSNF
16.	241600.0	122700.0	118900.0	2375.733	1957SSNF
17.	231700.0	117400.0	114300.0	5677.921	1957SSNF
18.	226400.0	114500.0	111900.0	10499.497	1957SSNF
19.	224800.0	113400.0	111400.0	15654.597	1957SSNF
20.	223500.0	112500.0	111000.0	20089.566	1957SSNF
21.	222800.0	111700.0	111100.0	24144.938	1957SSNF
22.	223700.0	112000.0	111700.0	26352.406	1957SSNF
23.	225600.0	113300.0	112300.0	27772.638	1957SSNF
24.	228700.0	115400.0	113300.0	28701.097	1957SSNF
25.	232200.0	117600.0	114600.0	28971.522	1957SSNF
26.	236100.0	120000.0	116100.0	27823.718	1957SSNF
27.	239600.0	122100.0	117500.0	26786.088	1957SSNF
28.	241800.0	122600.0	119200.0	25091.425	1957SSNF
29.	243400.0	122600.0	120800.0	23163.396	1957SSNF
30.	244900.0	122700.0	122200.0	21838.314	1957SSNF
31.	246500.0	122900.0	123600.0	19240.231	1957SSNF
32.	246100.0	122100.0	124000.0	18750.462	1957SSNF
33.	243600.0	120500.0	123100.0	16880.524	1957SSNF
34.	239500.0	118600.0	120900.0	14984.545	1957SSNF
35.	235000.0	116600.0	118400.0	13560.307	1957SSNF
36.	229800.0	113900.0	115900.0	12234.223	1957SSNF
37.	225300.0	111900.0	113400.0	10558.590	1957SSNF
38.	221300.0	110200.0	111100.0	8329.087	1957SSNF
39.	217500.0	108700.0	108800.0	6788.667	1957SSNF
40.	214100.0	107500.0	106600.0	5381.456	1957SSNF
41.	209700.0	105700.0	104000.0	3766.919	1957SSNF
42.	205100.0	103900.0	101200.0	3155.959	1957SSNF
43.	200100.0	101800.0	98300.0	2016.168	1957SSNF
44.	194500.0	99300.0	95200.0	1170.840	1957SSNF
45.	189500.0	97100.0	92400.0	608.957	1957SSNF
46.	184200.0	94800.0	89400.0	331.521	1957SSNF
47.	178500.0	92100.0	86400.0	150.236	1957SSNF
48.	172500.0	89500.0	83000.0	67.105	1957SSNF
49.	165800.0	86100.0	79700.0	23.036	1957SSNF
50.	159800.0	83000.0	76800.0	21.033	1957SSNF

51.	153600.0	80000.0	73600.0	0.000	1957SSNF
52.	147700.0	76900.0	70800.0	0.000	1957SSNF
53.	142800.0	74100.0	68700.0	0.000	1957SSNF
54.	138700.0	71700.0	67000.0	0.000	1957SSNF
55.	134600.0	69500.0	65100.0	0.000	1957SSNF
56.	130800.0	67200.0	63600.0	0.000	1957SSNF
57.	126500.0	64700.0	61800.0	0.000	1957SSNF
58.	121500.0	61900.0	59600.0	0.000	1957SSNF
59.	116900.0	59400.0	57500.0	0.000	1957SSNF
60.	112400.0	56900.0	55500.0	0.000	1957SSNF
61.	107800.0	54400.0	53400.0	0.000	1957SSNF
62.	103800.0	52200.0	51600.0	0.000	1957SSNF
63.	101000.0	50900.0	50100.0	0.000	1957SSNF
64.	99000.0	50000.0	49000.0	0.000	1957SSNF
65.	96800.0	49000.0	47800.0	0.000	1957SSNF
66.	94300.0	48000.0	46300.0	0.000	1957SSNF
67.	91700.0	46800.0	44900.0	0.000	1957SSNF
68.	89000.0	45400.0	43600.0	0.000	1957SSNF
69.	85500.0	43300.0	42200.0	0.000	1957SSNF
70.	82700.0	41800.0	40900.0	0.000	1957SSNF
71.	79500.0	39900.0	39600.0	0.000	1957SSNF
72.	75100.0	37700.0	37400.0	0.000	1957SSNF
73.	70000.0	34900.0	35100.0	0.000	1957SSNF
74.	63900.0	32000.0	31900.0	0.000	1957SSNF
75.	57800.0	28800.0	29000.0	0.000	1957SSNF
76.	51800.0	25800.0	26000.0	0.000	1957SSNF
77.	46100.0	23000.0	23100.0	0.000	1957SSNF
78.	41100.0	20300.0	20800.0	0.000	1957SSNF
79.	36300.0	17900.0	18400.0	0.000	1957SSNF
80.	31800.0	15400.0	16400.0	0.000	1957SSNF
81.	27300.0	13000.0	14300.0	0.000	1957SSNF
82.	23200.0	11000.0	12200.0	0.000	1957SSNF
83.	19700.0	9300.0	10400.0	0.000	1957SSNF
84.	16900.0	7800.0	9100.0	0.000	1957SSNF
85.	14200.0	6400.0	7800.0	0.000	1957SSNF
86.	11800.0	5200.0	6600.0	0.000	1957SSNF
87.	9300.0	4200.0	5100.0	0.000	1957SSNF
88.	7400.0	3200.0	4200.0	0.000	1957SSNF
89.	6000.0	2700.0	3300.0	0.000	1957SSNF

DATA 1958BS TOTAL - 455303.00000

0.	445100.0	228100.0	217000.0	0.000	1958SSNF
1.	421300.0	215400.0	205900.0	0.000	1958SSNF
2.	406700.0	207700.0	199000.0	0.000	1958SSNF
3.	408800.0	208800.0	200000.0	0.000	1958SSNF
4.	397400.0	202700.0	194700.0	0.000	1958SSNF
5.	393600.0	200500.0	193100.0	0.000	1958SSNF
6.	384000.0	195700.0	188300.0	0.000	1958SSNF
7.	373700.0	190500.0	183200.0	0.000	1958SSNF
8.	362300.0	184800.0	177500.0	0.000	1958SSNF
9.	350600.0	178900.0	171700.0	0.000	1958SSNF
10.	339000.0	173100.0	165900.0	0.000	1958SSNF
11.	327500.0	167400.0	160100.0	0.000	1958SSNF
12.	314300.0	160700.0	153600.0	4.006	1958SSNF
13.	299600.0	153000.0	146600.0	22.036	1958SSNF
14.	283700.0	144700.0	139000.0	178.288	1958SSNF
15.	268900.0	136900.0	132000.0	711.150	1958SSNF
16.	253800.0	129000.0	124800.0	2368.829	1958SSNF
17.	241900.0	122500.0	119400.0	5782.347	1958SSNF
18.	235400.0	119300.0	116100.0	10657.227	1958SSNF
19.	232100.0	117000.0	115100.0	16246.262	1958SSNF
20.	229000.0	115000.0	114000.0	21145.181	1958SSNF
21.	226500.0	113300.0	113200.0	24281.251	1958SSNF
22.	226000.0	113000.0	113000.0	27002.650	1958SSNF
23.	227400.0	114000.0	113400.0	28031.313	1958SSNF
24.	230200.0	116100.0	114100.0	28163.527	1958SSNF
25.	233500.0	118200.0	115300.0	28479.037	1958SSNF
26.	237100.0	120700.0	116400.0	28186.564	1958SSNF
27.	241000.0	122900.0	118100.0	26451.759	1958SSNF
28.	243400.0	123800.0	119600.0	25647.459	1958SSNF
29.	245100.0	123900.0	121200.0	23116.368	1958SSNF
30.	247000.0	124300.0	122700.0	22104.732	1958SSNF
31.	248400.0	124300.0	124100.0	19034.770	1958SSNF
32.	248900.0	124100.0	124800.0	18460.842	1958SSNF
33.	247300.0	123100.0	124200.0	16762.096	1958SSNF
34.	244600.0	121600.0	123000.0	15125.450	1958SSNF
35.	241000.0	119700.0	121300.0	13219.369	1958SSNF
36.	237100.0	117700.0	119400.0	12046.473	1958SSNF
37.	233300.0	115900.0	117400.0	10440.878	1958SSNF
38.	228400.0	113400.0	115000.0	9033.603	1958SSNF
39.	223700.0	111400.0	112300.0	6537.568	1958SSNF
40.	219000.0	109600.0	109400.0	5339.632	1958SSNF
41.	213400.0	107300.0	106100.0	3551.741	1958SSNF
42.	208000.0	104900.0	103100.0	2927.733	1958SSNF
43.	203000.0	102800.0	100200.0	1872.026	1958SSNF
44.	198200.0	100800.0	97400.0	1188.922	1958SSNF
45.	194300.0	99200.0	95100.0	634.025	1958SSNF
46.	189600.0	97200.0	92400.0	309.500	1958SSNF
47.	184100.0	94900.0	89200.0	146.236	1958SSNF
48.	177900.0	91800.0	86100.0	60.097	1958SSNF
49.	171100.0	88400.0	82700.0	19.031	1958SSNF
50.	164800.0	85300.0	79500.0	13.021	1958SSNF

51.	158600.0	82200.0	76400.0	0.000	1958SSNF
52.	152300.0	78900.0	73400.0	0.000	1958SSNF
53.	147100.0	76200.0	70900.0	0.000	1958SSNF
54.	142200.0	73500.0	68700.0	0.000	1958SSNF
55.	137700.0	71100.0	66600.0	0.000	1958SSNF
56.	133400.0	68500.0	64900.0	0.000	1958SSNF
57.	128900.0	65900.0	63000.0	0.000	1958SSNF
58.	124100.0	63300.0	60800.0	0.000	1958SSNF
59.	119000.0	60400.0	58600.0	0.000	1958SSNF
60.	114600.0	57900.0	56700.0	0.000	1958SSNF
61.	109900.0	55200.0	54700.0	0.000	1958SSNF
62.	105700.0	53000.0	52700.0	0.000	1958SSNF
63.	102400.0	51300.0	51100.0	0.000	1958SSNF
64.	100000.0	50200.0	49800.0	0.000	1958SSNF
65.	97300.0	49100.0	48200.0	0.000	1958SSNF
66.	94800.0	47800.0	47000.0	0.000	1958SSNF
67.	91700.0	46400.0	45300.0	0.000	1958SSNF
68.	89000.0	44800.0	44200.0	0.000	1958SSNF
69.	85900.0	43100.0	42800.0	0.000	1958SSNF
70.	83400.0	41800.0	41600.0	0.000	1958SSNF
71.	80400.0	40000.0	40400.0	0.000	1958SSNF
72.	76400.0	37900.0	38500.0	0.000	1958SSNF
73.	71500.0	35500.0	36000.0	0.000	1958SSNF
74.	65500.0	32600.0	32900.0	0.000	1958SSNF
75.	59600.0	29600.0	30000.0	0.000	1958SSNF
76.	54000.0	26900.0	27100.0	0.000	1958SSNF
77.	48100.0	23900.0	24200.0	0.000	1958SSNF
78.	42900.0	21100.0	21800.0	0.000	1958SSNF
79.	37900.0	18600.0	19300.0	0.000	1958SSNF
80.	33200.0	16100.0	17100.0	0.000	1958SSNF
81.	28400.0	13600.0	14800.0	0.000	1958SSNF
82.	24200.0	11400.0	12800.0	0.000	1958SSNF
83.	20900.0	9800.0	11100.0	0.000	1958SSNF
84.	17400.0	7900.0	9500.0	0.000	1958SSNF
85.	14600.0	6600.0	8000.0	0.000	1958SSNF
86.	12000.0	5300.0	6700.0	0.000	1958SSNF
87.	9900.0	4300.0	5600.0	0.000	1958SSNF
88.	8000.0	3400.0	4600.0	0.000	1958SSNF
89.	6200.0	2700.0	3500.0	0.000	1958SSNF

DATA 1959BS TOTAL - 464449.00000

0.	444800.0	227600.0	217200.0	0.000	1959SSNF
1.	442200.0	226500.0	215700.0	0.000	1959SSNF
2.	419600.0	214500.0	205100.0	0.000	1959SSNF
3.	405400.0	207000.0	198400.0	0.000	1959SSNF
4.	407300.0	207900.0	199400.0	0.000	1959SSNF
5.	401500.0	204800.0	196700.0	0.000	1959SSNF
6.	392700.0	200300.0	192400.0	0.000	1959SSNF
7.	383500.0	195800.0	187700.0	0.000	1959SSNF
8.	373800.0	190800.0	183000.0	0.000	1959SSNF
9.	363400.0	185700.0	177700.0	0.000	1959SSNF
10.	353300.0	180500.0	172800.0	0.000	1959SSNF
11.	343100.0	175600.0	167500.0	1.002	1959SSNF
12.	331200.0	169400.0	161800.0	3.005	1959SSNF
13.	315700.0	161500.0	154200.0	24.037	1959SSNF
14.	297800.0	152000.0	145800.0	189.289	1959SSNF
15.	281600.0	143700.0	137900.0	789.205	1959SSNF
16.	264800.0	134900.0	129900.0	2570.925	1959SSNF
17.	251500.0	128000.0	123500.0	6111.329	1959SSNF
18.	242900.0	123200.0	119700.0	11416.428	1959SSNF
19.	237600.0	119800.0	117800.0	16880.769	1959SSNF
20.	232500.0	116700.0	115800.0	21880.401	1959SSNF
21.	228100.0	113900.0	114200.0	25770.339	1959SSNF
22.	225900.0	112500.0	113400.0	27568.084	1959SSNF
23.	226500.0	113300.0	113200.0	29057.357	1959SSNF
24.	228900.0	115300.0	113600.0	29092.411	1959SSNF
25.	232200.0	117500.0	114700.0	28309.215	1959SSNF
26.	236300.0	120600.0	115700.0	27663.229	1959SSNF
27.	239500.0	122700.0	116800.0	26921.096	1959SSNF
28.	242200.0	123700.0	118500.0	25864.483	1959SSNF
29.	244700.0	124400.0	120300.0	24085.768	1959SSNF
30.	247000.0	125100.0	121900.0	21938.490	1959SSNF
31.	249600.0	125800.0	123800.0	19426.656	1959SSNF
32.	250800.0	126000.0	124800.0	18157.718	1959SSNF
33.	250200.0	125400.0	124800.0	16653.422	1959SSNF
34.	248400.0	123900.0	124500.0	15340.418	1959SSNF
35.	245900.0	122200.0	123700.0	13404.462	1959SSNF
36.	243100.0	120400.0	122700.0	11804.019	1959SSNF
37.	239900.0	118700.0	121200.0	10648.255	1959SSNF
38.	235100.0	116600.0	118500.0	9293.186	1959SSNF
39.	229300.0	114100.0	115200.0	7316.168	1959SSNF
40.	223400.0	111400.0	112000.0	5210.955	1959SSNF
41.	217200.0	108900.0	108300.0	3785.779	1959SSNF
42.	211400.0	106300.0	105100.0	2969.533	1959SSNF
43.	206500.0	104200.0	102300.0	1886.880	1959SSNF
44.	203200.0	103100.0	100100.0	1167.783	1959SSNF
45.	199100.0	101400.0	97700.0	708.081	1959SSNF
46.	195000.0	99600.0	95400.0	276.422	1959SSNF
47.	190400.0	97700.0	92700.0	153.234	1959SSNF
48.	184200.0	94800.0	89400.0	75.115	1959SSNF
49.	177500.0	91400.0	86100.0	23.035	1959SSNF
50.	171200.0	88300.0	82900.0	11.017	1959SSNF

51.	164700.0	85000.0	79700.0	0.000	1959SSNF
52.	158100.0	81800.0	76300.0	0.000	1959SSNF
53.	152500.0	78800.0	73700.0	0.000	1959SSNF
54.	147100.0	75900.0	71200.0	0.000	1959SSNF
55.	142400.0	73300.0	69100.0	0.000	1959SSNF
56.	137300.0	70500.0	66800.0	0.000	1959SSNF
57.	132100.0	67700.0	64400.0	0.000	1959SSNF
58.	127300.0	65100.0	62200.0	0.000	1959SSNF
59.	122100.0	62000.0	60100.0	0.000	1959SSNF
60.	117800.0	59600.0	58200.0	0.000	1959SSNF
61.	112900.0	56800.0	56100.0	0.000	1959SSNF
62.	108700.0	54300.0	54400.0	0.000	1959SSNF
63.	105200.0	52500.0	52700.0	0.000	1959SSNF
64.	101700.0	50700.0	51000.0	0.000	1959SSNF
65.	99300.0	49700.0	49600.0	0.000	1959SSNF
66.	95800.0	47900.0	47900.0	0.000	1959SSNF
67.	92600.0	46200.0	46400.0	0.000	1959SSNF
68.	89900.0	44900.0	45000.0	0.000	1959SSNF
69.	87200.0	43400.0	43800.0	0.000	1959SSNF
70.	84500.0	42100.0	42400.0	0.000	1959SSNF
71.	82000.0	40600.0	41400.0	0.000	1959SSNF
72.	78100.0	38500.0	39600.0	0.000	1959SSNF
73.	72900.0	35900.0	37000.0	0.000	1959SSNF
74.	67100.0	33000.0	34100.0	0.000	1959SSNF
75.	61800.0	30600.0	31200.0	0.000	1959SSNF
76.	55800.0	27500.0	28300.0	0.000	1959SSNF
77.	50200.0	24800.0	25400.0	0.000	1959SSNF
78.	45000.0	22200.0	22800.0	0.000	1959SSNF
79.	39800.0	19500.0	20300.0	0.000	1959SSNF
80.	35000.0	16900.0	18100.0	0.000	1959SSNF
81.	29900.0	14100.0	15800.0	0.000	1959SSNF
82.	25400.0	12000.0	13400.0	0.000	1959SSNF
83.	21700.0	10000.0	11700.0	0.000	1959SSNF
84.	18500.0	8500.0	10000.0	0.000	1959SSNF
85.	15300.0	7000.0	8300.0	0.000	1959SSNF
86.	12800.0	5700.0	7100.0	0.000	1959SSNF
87.	10400.0	4600.0	5800.0	0.000	1959SSNF
88.	8200.0	3600.0	4600.0	0.000	1959SSNF
89.	6300.0	2600.0	3700.0	0.000	1959SSNF

DATA 1960BS TOTAL · 463378.00000

0.	448600.0	229600.0	219000.0	0.000	1960SSNF
1.	442400.0	226400.0	216000.0	0.000	1960SSNF
2.	440900.0	225900.0	215000.0	0.000	1960SSNF
3.	418500.0	213900.0	204600.0	0.000	1960SSNF
4.	404400.0	206600.0	197800.0	0.000	1960SSNF
5.	409800.0	209300.0	200500.0	0.000	1960SSNF
6.	402100.0	205200.0	196900.0	0.000	1960SSNF
7.	394200.0	201300.0	192900.0	0.000	1960SSNF
8.	386100.0	197100.0	189000.0	0.000	1960SSNF
9.	376800.0	192500.0	184300.0	0.000	1960SSNF
10.	367900.0	188100.0	179800.0	0.000	1960SSNF
11.	359300.0	183600.0	175700.0	0.000	1960SSNF
12.	347600.0	177600.0	170000.0	4.006	1960SSNF
13.	331600.0	169500.0	162100.0	27.040	1960SSNF
14.	313300.0	160100.0	153200.0	179.265	1960SSNF
15.	295800.0	151100.0	144700.0	751.109	1960SSNF
16.	278000.0	142000.0	136000.0	2645.906	1960SSNF
17.	262900.0	134000.0	128900.0	6377.414	1960SSNF
18.	251800.0	127900.0	123900.0	11559.063	1960SSNF
19.	244500.0	123400.0	121100.0	17602.984	1960SSNF
20.	237000.0	118800.0	118200.0	22510.228	1960SSNF
21.	230200.0	114400.0	115800.0	26312.841	1960SSNF
22.	226300.0	112200.0	114100.0	27997.327	1960SSNF
23.	226100.0	112400.0	113700.0	28555.151	1960SSNF
24.	228100.0	114500.0	113600.0	29013.828	1960SSNF
25.	231300.0	116800.0	114500.0	28532.117	1960SSNF
26.	234900.0	119500.0	115400.0	26971.814	1960SSNF
27.	238600.0	121900.0	116700.0	26348.894	1960SSNF
28.	241300.0	123400.0	117900.0	25446.562	1960SSNF
29.	243700.0	124100.0	119600.0	23788.114	1960SSNF
30.	246400.0	125100.0	121300.0	22696.503	1960SSNF
31.	248900.0	126000.0	122900.0	18659.544	1960SSNF
32.	250800.0	126600.0	124200.0	17932.470	1960SSNF
33.	251200.0	126400.0	124800.0	16268.013	1960SSNF
34.	250800.0	125500.0	125300.0	14882.969	1960SSNF
35.	249700.0	124400.0	125300.0	13418.808	1960SSNF
36.	248800.0	123300.0	125500.0	11747.340	1960SSNF
37.	246500.0	121800.0	124700.0	10140.969	1960SSNF
38.	241500.0	119500.0	122000.0	9172.540	1960SSNF
39.	234900.0	116600.0	118300.0	7383.899	1960SSNF
40.	227900.0	113300.0	114600.0	5881.682	1960SSNF
41.	220400.0	110000.0	110400.0	3520.196	1960SSNF
42.	214000.0	107300.0	106700.0	2883.256	1960SSNF
43.	209900.0	105500.0	104400.0	1911.822	1960SSNF
44.	206700.0	104400.0	102300.0	1112.642	1960SSNF
45.	203600.0	103100.0	100500.0	622.920	1960SSNF
46.	199800.0	101600.0	98200.0	307.454	1960SSNF
47.	195500.0	99700.0	95800.0	125.185	1960SSNF
48.	189600.0	97100.0	92500.0	51.075	1960SSNF
49.	182700.0	93700.0	89000.0	19.028	1960SSNF
50.	176600.0	90700.0	85900.0	16.024	1960SSNF

51.	170200.0	87500.0	82700.0	0.000	1960SSNF
52.	163700.0	84200.0	79500.0	0.000	1960SSNF
53.	157500.0	81100.0	76400.0	0.000	1960SSNF
54.	152200.0	78500.0	73700.0	0.000	1960SSNF
55.	146400.0	75400.0	71000.0	0.000	1960SSNF
56.	140600.0	72300.0	68300.0	0.000	1960SSNF
57.	135300.0	69500.0	65800.0	0.000	1960SSNF
58.	130100.0	66400.0	63700.0	0.000	1960SSNF
59.	125100.0	63600.0	61500.0	0.000	1960SSNF
60.	120500.0	61000.0	59500.0	0.000	1960SSNF
61.	115500.0	58100.0	57400.0	0.000	1960SSNF
62.	111300.0	55600.0	55700.0	0.000	1960SSNF
63.	107500.0	53600.0	53900.0	0.000	1960SSNF
64.	104200.0	51800.0	52400.0	0.000	1960SSNF
65.	100800.0	50100.0	50700.0	0.000	1960SSNF
66.	96900.0	48100.0	48800.0	0.000	1960SSNF
67.	94100.0	46600.0	47500.0	0.000	1960SSNF
68.	90900.0	45000.0	45900.0	0.000	1960SSNF
69.	88100.0	43500.0	44600.0	0.000	1960SSNF
70.	85300.0	42000.0	43300.0	0.000	1960SSNF
71.	82500.0	40600.0	41900.0	0.000	1960SSNF
72.	79000.0	38700.0	40300.0	0.000	1960SSNF
73.	74100.0	36300.0	37800.0	0.000	1960SSNF
74.	68700.0	33800.0	34900.0	0.000	1960SSNF
75.	63200.0	31000.0	32200.0	0.000	1960SSNF
76.	57600.0	28400.0	29200.0	0.000	1960SSNF
77.	52000.0	25700.0	26300.0	0.000	1960SSNF
78.	46700.0	22900.0	23800.0	0.000	1960SSNF
79.	41500.0	20200.0	21300.0	0.000	1960SSNF
80.	36400.0	17400.0	19000.0	0.000	1960SSNF
81.	31200.0	14800.0	16400.0	0.000	1960SSNF
82.	26500.0	12400.0	14100.0	0.000	1960SSNF
83.	22900.0	10600.0	12300.0	0.000	1960SSNF
84.	19300.0	8900.0	10400.0	0.000	1960SSNF
85.	16000.0	7300.0	8700.0	0.000	1960SSNF
86.	13500.0	6100.0	7400.0	0.000	1960SSNF
87.	10900.0	4800.0	6100.0	0.000	1960SSNF
88.	8600.0	3800.0	4800.0	0.000	1960SSNF
89.	6700.0	2900.0	3800.0	0.000	1960SSNF

DATA 1961BS TOTAL - 460109.00000

0.	450509.0	230518.0	219991.0	0.000	1961SSNF
1.	443035.0	226504.0	216531.0	0.000	1961SSNF
2.	439987.0	225077.0	214910.0	0.000	1961SSNF
3.	435454.0	222501.0	212953.0	0.000	1961SSNF
4.	419721.0	215089.0	204632.0	0.000	1961SSNF
5.	415670.0	212768.0	202902.0	0.000	1961SSNF
6.	410194.0	209997.0	200197.0	0.000	1961SSNF
7.	403586.0	206544.0	197042.0	0.000	1961SSNF
8.	396614.0	202889.0	193725.0	0.000	1961SSNF
9.	389054.0	199020.0	190034.0	0.000	1961SSNF
10.	381581.0	195112.0	186469.0	0.000	1961SSNF
11.	374675.0	191445.0	183230.0	0.000	1961SSNF
12.	364059.0	185933.0	178126.0	7.009	1961SSNF
13.	347883.0	177678.0	170205.0	25.031	1961SSNF
14.	328337.0	167737.0	160600.0	198.247	1961SSNF
15.	309628.0	158222.0	151406.0	789.986	1961SSNF
16.	291087.0	148877.0	142210.0	2655.313	1961SSNF
17.	274638.0	140266.0	134372.0	6390.973	1961SSNF
18.	261732.0	132948.0	128784.0	11955.915	1961SSNF
19.	251645.0	126781.0	124864.0	17655.025	1961SSNF
20.	241853.0	120832.0	121021.0	23200.944	1961SSNF
21.	232421.0	114980.0	117441.0	26395.930	1961SSNF
22.	226504.0	111479.0	115025.0	28625.711	1961SSNF
23.	225284.0	111268.0	114016.0	29285.535	1961SSNF
24.	227346.0	113292.0	114054.0	28361.382	1961SSNF
25.	229976.0	115499.0	114477.0	28270.268	1961SSNF
26.	233526.0	118210.0	115316.0	26731.348	1961SSNF
27.	237081.0	120658.0	116423.0	25194.431	1961SSNF
28.	239862.0	122239.0	117623.0	24435.484	1961SSNF
29.	242133.0	123233.0	118900.0	23097.815	1961SSNF
30.	244800.0	124510.0	120290.0	21889.308	1961SSNF
31.	247518.0	125877.0	121641.0	19514.345	1961SSNF
32.	249828.0	126798.0	123030.0	17619.981	1961SSNF
33.	251542.0	127096.0	124446.0	15936.882	1961SSNF
34.	252551.0	126852.0	125699.0	14191.705	1961SSNF
35.	253274.0	126480.0	126794.0	12796.965	1961SSNF
36.	254004.0	126061.0	127943.0	11872.812	1961SSNF
37.	252272.0	124717.0	127555.0	10228.761	1961SSNF
38.	247033.0	122121.0	124912.0	8858.051	1961SSNF
39.	239518.0	118722.0	120796.0	7157.930	1961SSNF
40.	231889.0	115226.0	116663.0	5764.191	1961SSNF
41.	223515.0	111380.0	112135.0	4046.048	1961SSNF
42.	216656.0	108303.0	108353.0	2801.495	1961SSNF
43.	212579.0	106579.0	106000.0	1903.375	1961SSNF
44.	210181.0	105665.0	104516.0	1086.355	1961SSNF
45.	207139.0	104468.0	102671.0	608.759	1961SSNF
46.	204000.0	103242.0	100758.0	292.365	1961SSNF
47.	199975.0	101520.0	98455.0	157.196	1961SSNF
48.	194470.0	98974.0	95496.0	67.084	1961SSNF
49.	187995.0	95874.0	92121.0	22.027	1961SSNF
50.	181712.0	92854.0	88858.0	17.021	1961SSNF

51.	175467.0	89823.0	85644.0	0.000	1961SSN
52.	169132.0	86714.0	82418.0	0.000	1961SSN
53.	162788.0	83568.0	79220.0	0.000	1961SSN
54.	156483.0	80405.0	76078.0	0.000	1961SSN
55.	150170.0	77208.0	72962.0	0.000	1961SSN
56.	143952.0	73997.0	69955.0	0.000	1961SSN
57.	137951.0	70889.0	67062.0	0.000	1961SSN
58.	132675.0	67944.0	64731.0	0.000	1961SSN
59.	127862.0	65139.0	62723.0	0.000	1961SSN
60.	123137.0	62379.0	60758.0	0.000	1961SSN
61.	118550.0	59669.0	58881.0	0.000	1961SSN
62.	114205.0	57143.0	57062.0	0.000	1961SSN
63.	110112.0	54858.0	55254.0	0.000	1961SSN
64.	106225.0	52754.0	53471.0	0.000	1961SSN
65.	102418.0	50684.0	51734.0	0.000	1961SSN
66.	98607.0	48625.0	49982.0	0.000	1961SSN
67.	95114.0	46752.0	48362.0	0.000	1961SSN
68.	92062.0	45127.0	46935.0	0.000	1961SSN
69.	89217.0	43637.0	45580.0	0.000	1961SSN
70.	86318.0	42133.0	44185.0	0.000	1961SSN
71.	83508.0	40681.0	42827.0	0.000	1961SSN
72.	79952.0	38910.0	41042.0	0.000	1961SSN
73.	75249.0	36641.0	38608.0	0.000	1961SSN
74.	69782.0	34030.0	35752.0	0.000	1961SSN
75.	64436.0	31490.0	32946.0	0.000	1961SSN
76.	59084.0	28965.0	30119.0	0.000	1961SSN
77.	53704.0	26351.0	27353.0	0.000	1961SSN
78.	48382.0	23642.0	24740.0	0.000	1961SSN
79.	43162.0	20907.0	22255.0	0.000	1961SSN
80.	38004.0	18221.0	19783.0	0.000	1961SSN
81.	32926.0	15569.0	17357.0	0.000	1961SSN
82.	28270.0	13182.0	15088.0	0.000	1961SSN
83.	24209.0	11179.0	13030.0	0.000	1961SSN
84.	20643.0	9480.0	11163.0	0.000	1961SSN
85.	17303.0	7879.0	9424.0	0.000	1961SSN
86.	14249.0	6417.0	7832.0	0.000	1961SSN
87.	11546.0	5133.0	6413.0	0.000	1961SSN
88.	9201.0	4027.0	5174.0	0.000	1961SSN
89.	7222.0	3101.0	4121.0	0.000	1961SSN

1.2 Population Data

To estimate the economic mobility of the population, values of age-, income-specific population density are required. The Interim Report included an algorithm for estimating densities from given populations; hence it is sufficient if numbers of individuals between ages x and x + 1 having incomes between s and s + ds are obtained.

Unfortunately, data in single years of age is unavailable (except for recent censes); published data of this type is aggregated in larger age groups. For this study, data is taken from the annual reports on taxation [24] provided by Revenue Canada. Hence for each of the years 1963-1973, the numbers of males and females in the age groups:

less than 25

25-29
30-34
35-39
40-44
45-49
50-54
55-59
60-64
65-69
over 69

earning non-taxable incomes in the brackets

$ 0-1000	1963-1965	$ 0-1000	1966-1973
1000-2000		1000-2000	
2000-3000		2000-3000	
over 3000		3000-4000	
		4000-5000	
		over 5000	

or taxable incomes in the brackets

$ 0-2000
2000-3000
3000-4000
4000-5000
5000-6000
6000-7000
7000-8000
8000-9000
9000-10000
10000-15000
15000-20000
20000-25000
over -25000

are available.* This data has been transcribed from these annual reports to punched cards, and checked with respect to totals for each income bracket using the format

```
2I2 , 4X , I7 , 3X , I7 , 4X , A1 , 9X , 2I2 , 29X , A6
```

Income Start End ($1000.)	Males Canada	Females Canada	Type T=Taxable N=non-taxable A=all returns	Age Start End	Year 'I1'

This data forms direct input to the programs used for estimating economic mobility (Chapter 4).

*Since 1972 the lower age brackets have been subdivided into two groups: less than 19, 19-25.

1.3 Mortality Data

The estimation of economic mobility from population data also requires estimates of age-specific death rates. These have been estimated using the numbers of deaths between ages x and x + 1 given annual in Vital Statistics [23]. As in the case of fertility, the mortality of individuals at age x at time t is estimated by the ratio

$$R(x,t) = \frac{\text{number of individuals dying at age } x \text{ during year } t}{\text{number of individuals of age } x \text{ alive at time } t} .$$

To accommodate the possibility that Newfoundland might have been excluded (it has not), punched cards include data for Canada and for Newfoundland in the following format:

Death Singles Canada including Newfoundland

2X,	I2,	T10,	I4,	I3,	T20,	I4,	I3,	T29,	I5,	I3,	T73,	A8
	Age		Male deaths Canada	Male deaths Nfld.		Female deaths Canada	Female deaths Nfld.		Total deaths Canada	Total deaths Nfld.		Year 'DSCN'

This data forms direct input to subroutine CALCR which is used in estimating economic mobility, and in the aggregation experiments discussed in Chapter III.

2. ESTIMATION OF FERTILITY PARAMETERS

2.0 Introduction.

Programs for the Interim Report have been modified and in some cases extended to accommodate the data described in Section 1.1. Here, listings of programs and description for their use are provided. These have been developed with the objective of estimating dominant components of the fertility parameters.

The results have been produced and presented in a form which readily demonstrates the consistency of the parameters over a period of years. Using the estimated values of the parameters, values of the fertility are simulated and compared to the actual values. Further support for the model is presented graphically through profile plots of the actual and simulated fertility, as well as for the components of fertility.

2.1 Normalized Fertility

The nominal model representing the evaluation of the fertility is

$$\frac{df}{dt}(x,t) = -\frac{d}{dx}\left(\sum_{r=1}^{q} q_r(t)\ d_r(t)\ f(x,t)\right) - b(t)\ f(x,t)\ . \qquad (1)$$

By modifying (or normalizing) the fertility to

$$\bar{f}(x,t) = e^{-\int_0^t b(\tau)d\tau} f(x,t) \quad , \qquad (2)$$

it has been shown that

$$\frac{d\bar{f}}{dt}(x,t) = \frac{-d}{dx}\left(\sum_{r=1}^{q} a_r(t)\ d_r(x)\ \bar{f}(x,t)\right). \qquad (3)$$

If the distribution of the normalized fertility were known exactly, the function of two variables

$$m(x,t) = \sum_{r=1}^{q} d_r(t)\ d_r(x)$$

would be evaluated using the equivalent formulae

$$m(x,t) \quad \frac{-1}{\overline{f}(x,t)} \int_{\underline{x}}^{x} \frac{d\overline{f}}{dt}(\xi,t)\,d\zeta \qquad (4a)$$

$$= \frac{-1}{\overline{f}(x,t)} \frac{d}{dt} \int_{\underline{x}}^{x} \overline{f}(\xi,t)\,d\zeta \qquad (4b)$$

where $\underline{x}$ = minimum value of x. Either (4a) or (4b) could in principle be used algorithmically to estimate $m(x,t)$ from values of $\overline{f}(x,t)$ given only for a discrete set of ages x and time t.

As the integral (4) is required for all discrete values of x for each t, the integral was approximated by a cubic spline in x for each t and integrated exactly (rather than approximating each of the integrals directly by numerical quadrature).

In limited testing, numerical algorithms based on formula (4b) yielded larger errors. Hence the programs developed have been based on formula (4a).

To obtain estimates of the normalized fertility $\overline{f}(x,t)$, an estimate of the integration factor in (2) was needed. Integration of (1) or (1') over the range of ages for which fertility is non-zero gives

$$b(t) = \frac{-\frac{d}{dt}\int_0^{50} f(x,t)\,dx}{\int_0^{50} f(x,t)\,dx} \tag{5}$$

$$= -\frac{F'(t)}{F(t)}$$

with

$$F(t) = \int_0^{50} f(x,t)\,dx \quad . \tag{6a}$$

This integral was approximated using the midpoint quadrature rule for each t

$$F(t) = \sum_{i=0}^{50} f(i,t) \quad . \tag{6b}$$

To estimate the integrating factor, (5) implies

$$e^{\int_0^t b(\tau)d\tau} = e^{-\int_0^t \frac{F'(\tau)}{F(\tau)}d\tau} = e^{-\ell n(F(\tau))\big|_0^t}$$

$$= \frac{F(0)}{F(t)} \tag{7}$$

so that

$$\bar{f}(x,t) = \frac{F(0)}{F(t)} f(x,t) \tag{8}$$

and $\bar{f}(x,t)$ was estimated using (6b) and (8) . Here, $F(0) = F(1956)$, the first year for which fertility data was acquired. For the subsequent simulation required to validate the model, values of $b(t)$ were estimated by the formula

$$b(t) = -\frac{F'(t)}{F(t)} = \frac{F(t) - F(t+1)}{F(t)} \tag{5'}$$

Computation yielded the values given in the following table:

t	F(t)	b(t)
1956	3.85865	
1957	3.92929	-.01831
1958	3.88774	.01055
1959	3.94680	-.01519
1960	3.91022	.00927
1961	3.85704	.01360
1962	3.77676	.02084
1963	3.68987	.02301
1964	3.52102	.04576
1965	3.16255	.10181
1966	2.81195	.11086
1967	2.60412	.07391
1968	2.46239	.05443
1969	2.41282	.02013
1970	2.33947	.03040
1971	2.18869	.06445
1972	2.02195	.07618

Table 2.1: Fractional Rate of Decline of Family Size (b(t))

Program 2.1 reads data from SSNF and prints $F(x,t)$ and $\tilde{F}(x,t)$, and punches $\bar{F}(x,t)$. This punched output is used as data for the estimation of $m(x,t)$, and later, transformed by $F(t)$ to retrieve the fertility, as comparison data for the simulated fertility.

```
$ JOB    ACCT-NUM,VERNER,TIME=60
C        ************************************************************
C
C              PROGRAM 2.1  :   FBAR
C
C        ************************************************************
```

```
3JOB FBAR; CHARGE BERGHOUT; CLASS 3;
USER 080002/VHJ;
BEGIN
3COMPILE FBAR FORTRAN;
DATA
FILE  5=FILE5
```

```
3DATA FILE5
```

```
      C     ##########################################################################
      C
      C     PROGRAM 2.1
      C     TITLE : FBAR
      C     PURPOSE : CREATE F(X.T),FBAR(X.T) AND F(T)
      C     DATA : SSNF
      C     OUTPUT : PRINTS FERTILITY AND FERTILITY TOTALS
      C                 PUNCHES NORMALIZED FERTILITY
      C
      C     ##########################################################################
            REAL CODES(17,2)
            COMMON CODES
            COMMON /RESULT/FBAR(51,17), FBARST(51,17), G(51,17), GT(17,51)
            DATA CODES/'1956','1957','1958','1959','1960','1961',
           & '1962','1963','1964','1965','1966','1967','1968','1969',
           & '1970','1971','1972',
           &           'CS','RS','RS','RS','RS','CS','RS','RS','RS','RS',
           & 'CS','RS','RS','RS','RS','CS','RS'/
*EXTENSION*  CM-1   CODES
            NX = 51
            NT= 17
            CALL GTBAR(NX,NT)
            CALL DUMPS(2,NX,NT)
            WRITE(7,102)((FBAR(I,J),I=1,NX),J=1,NT)
        102 FORMAT(4F17.7)
            STOP
            END

            SUBROUTINE DUMPS(LL,NX,NT)
            REAL CODES(17,2)
            INTEGER NAMER(10)/'FERT','ILIT','Y   ','    ','    ','NORM',
           &'ALIZ','ED F','ERTI','LITY'/
            COMMON CODES
            COMMON /RESULT/FBAR(51,17), FBARST(51,17), G(51,17), GT(17,51)
            IPAGE = 0
            LP=5*(LL-1)+1
            LPE=LP+4
            IED=0
            DO 20 I=1,4
                 IST=IED+1
                 IED=IED+5
                 IF (IED.GT.NT) IED=NT
                 DO 20 J = 1,NX
            JM1=J-1
                 IF (MOD(JM1,26).NE.0) GO TO 20
            IPAGE = IPAGE + 1
            WRITE(6,200)(NAMER(K),K=LP,LPE),IPAGE,((CODES(K,L),L=1,2),
           &    K=IST,IED)
         20 WRITE(6,100) JM1,(FBAR(J,K),K=IST,IED)
        100 FORMAT(/,I5,5F14.5)
        200 FORMAT('1',5A4,15X,'YEAR',20X,'PAGE',I3,//,
           & '      AGE',5(8X,4('_'),2X),/,'+',5X,5(8X,A4,A2),//)
            RETURN
            END

            SUBROUTINE GTBAR(NX,NT)
            REAL CODES(17,2)
            COMMON /RESULT/FBAR(51,17), FBARST(51,17), G(51,17), GT(17,51)
            COMMON CODES
```

```
      REAL F(17),XDATA(90,5)
      INTEGER MIN(17)
      DATA MIN/11*90,4*70,2*90/
C     NX = 51, AGES 0 --> 50
C     NT = 17, YEARS 1956 --> 1972
C
C     GET THE FEMALES DATA AND THE BIRTHS DATA AND CALCULATE
C     THE FERTILITY RATES
C
      DO 10 I = 1,NT
         CODES(I,2)='SS'

         F(I) = 0.0
         LL=5
         CALL DATAIN(XDATA,90,5,'      ',0.0,LL,CODES(I,1),CODES(I,2))
         DO 10 J = 1,NX
            FBAR(J,I) = XDATA(J,5) / XDATA(J,4)
   10       F(I) = F(I) + FBAR(J,I)
C
C    CHANGE THE CODES FOR PRINTING THE YEAR
C
      DO 40 I = 1,NT
   40 CODES(I,2)='  '

      CALL DUMPS(1,NX,NT)
      WRITE(6,103) (F(I),I=1,NT)
  103 FORMAT('1',10X, 'F(T)',//,(/,5F13.5))
C
C     CALCULATE FBAR(X,T) = F(T)/F(0) * FBAR(X,T)
C
      DO 30 I = 2,NT
         SCALAR = F(1) / F(I)
         DO 30 J = 12,NX
   30       FBAR(J,I) = SCALAR * FBAR(J,I)
      RETURN
      END

      SUBROUTINE DATAIN(DATAS,ROWS,COLS,NAME,ERRORB,LL,SYEAR,CODE)
C
C     DATAS IS THE DATA TO BE INPUT MATRIX , INPUT BY ROW
C     ROWS IS THE # OF ROWS IN DATAS
C     COLS IS THE # OF COLUMNS IN DATAS
C     NAME IF NOT BLANK THEN DATA WILL BE CHECKED AS IT IS READ
C     ERRORB IS THE MAXIMUM ERROR ALLOWED BEFORE BEING FLAGGED
C     LL IS THE UNIT NUMBER FROM WHICH THE DATA IS READ
C     SYEAR IS THE YEAR OF THE DATA
C     CODE IS THE EXPECTED CODE
C
      REAL SYEAR,YEAR
      REAL NAME,B/'      '/
      REAL CODE,CODEIN
      INTEGER ROWS, COLS
      REAL DATAS(90,5), ERRORB,ERR
      IF( NAME .NE. B ) GO TO 5
      READ(LL,100)((DATAS(I,J),J=1,COLS),YEAR,CODEIN,I=1,ROWS)
      IF (YEAR.EQ.SYEAR.AND.CODE.EQ.CODEIN) GO TO 15
      WRITE(6,210)SYEAR,CODE,YEAR,CODEIN
      STOP
    5 DO 10 I = 1,ROWS
      READ(LL,100)(DATAS(I,J),J=1,COLS),YEAR,CODEIN
```

```
      IF (YEAR.EQ.SYEAR.AND.CODE.EQ.CODEIN) GO TO 10
      WRITE(6,210)SYEAR,CODE,YEAR,CODEIN,I
      STOP
   10 CONTINUE
C
C     SCALE THE DATA
C
   15 CONTINUE
   16 IF(NAME .EQ. B ) RETURN
      DO 20 I = 1, ROWS
      ERR = DATAS(I,2) - ( DATAS(I,3) + DATAS(I,4) )
      IF ( ABS(ERR) .LT. ERRORB ) GO TO 20
      WRITE(6,200) NAME, ( DATAS(I,J), J=1,4)
   20 CONTINUE
      RETURN
  100 FORMAT(F4.0,3F9.1,F9.3,32X,A4,A2)
  200 FORMAT(/,'  ',A4,5X, F5.0, 3F11.1)
  210 FORMAT(1X,10('* '),'CARD CODES DISAGREE, EXPECTED CODE=',A4,A2,
     &  ' GOT ',A4,A2,', CARD #',I2)
      END
```

```
C     ****************************************************************************
C
C     DATA FOR
C           PROGRAM 2.1  :   FBAR
C
C     ****************************************************************************
```

AGE	1956	1957	1958	1959	1960
0	0.00000	0.00000	0.00000	0.00000	0.00000
1	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000
5	0.00000	0.00000	0.00000	0.00000	0.00000
6	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000
10	0.00000	0.00000	0.00000	0.00000	0.00000
11	0.00000	0.00000	0.00000	0.00001	0.00000
12	0.00001	0.00002	0.00003	0.00002	0.00002
13	0.00016	0.00022	0.00015	0.00016	0.00017
14	0.00101	0.00104	0.00128	0.00130	0.00117
15	0.00494	0.00514	0.00539	0.00572	0.00519
16	0.01727	0.01998	0.01898	0.01979	0.01946
17	0.04615	0.04968	0.04843	0.04948	0.04948
18	0.08778	0.09383	0.09179	0.09538	0.09329
19	0.12879	0.14053	0.14115	0.14330	0.14536
20	0.17339	0.18099	0.18548	0.18895	0.19044
21	0.20960	0.21733	0.21450	0.22566	0.22723
22	0.22663	0.23592	0.23896	0.24310	0.24538
23	0.24622	0.24731	0.24719	0.25669	0.25114
24	0.25360	0.25332	0.24683	0.25610	0.25540
25	0.24748	0.25281	0.24700	0.24681	0.24919

FERTILITY | YEAR | PAGE 2

AGE	1956	1957	1958	1959	1960
26	0.23691	0.23965	0.24215	0.23909	0.23372
27	0.21692	0.22797	0.22398	0.23049	0.22578
28	0.21169	0.21050	0.21444	0.21827	0.21583
29	0.18926	0.19175	0.19073	0.20021	0.19890
30	0.18093	0.17871	0.18015	0.17997	0.18711
31	0.15524	0.15567	0.15338	0.15692	0.15183
32	0.15117	0.15121	0.14792	0.14549	0.14438
33	0.13567	0.13713	0.13496	0.13344	0.13035
34	0.12787	0.12394	0.12297	0.12322	0.11878
35	0.11527	0.11453	0.10898	0.10836	0.10709
36	0.10455	0.10556	0.10089	0.09620	0.09360
37	0.08493	0.09311	0.08893	0.08786	0.08132
38	0.07791	0.07497	0.07855	0.07842	0.07518
39	0.06246	0.06240	0.05822	0.06351	0.06242
40	0.05043	0.05048	0.04881	0.04653	0.05132
41	0.03700	0.03622	0.03348	0.03496	0.03189
42	0.03078	0.03119	0.02840	0.02825	0.02702
43	0.02019	0.02051	0.01868	0.01844	0.01831
44	0.01299	0.01230	0.01221	0.01167	0.01088
45	0.00723	0.00659	0.00667	0.00725	0.00620
46	0.00347	0.00371	0.00335	0.00290	0.00313
47	0.00140	0.00174	0.00164	0.00165	0.00131
48	0.00079	0.00081	0.00070	0.00084	0.00055
49	0.00040	0.00029	0.00023	0.00027	0.00021
50	0.00018	0.00027	0.00016	0.00013	0.00019

FERTILITY YEAR PAGE 3

AGE	1961	1962	1963	1964	1965
0	0.00000	0.00000	0.00000	0.00000	0.00000
1	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000
5	0.00000	0.00000	0.00000	0.00000	0.00000
6	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000
10	0.00000	0.00000	0.00000	0.00000	0.00000
11	0.00000	0.00000	0.00000	0.00000	0.00001
12	0.00004	0.00000	0.00002	0.00002	0.00002
13	0.00015	0.00016	0.00012	0.00013	0.00017
14	0.00123	0.00108	0.00093	0.00100	0.00086
15	0.00522	0.00523	0.00514	0.00449	0.00438
16	0.01867	0.01730	0.01924	0.01688	0.01640
17	0.04756	0.04299	0.04310	0.04529	0.04146
18	0.09284	0.08734	0.08087	0.07813	0.08204
19	0.14139	0.13916	0.13294	0.11990	0.11364
20	0.19171	0.18669	0.18128	0.16648	0.14347
21	0.22476	0.22520	0.21589	0.20579	0.17990
22	0.24887	0.24308	0.24105	0.22510	0.20407
23	0.25685	0.25455	0.24884	0.23647	0.21057
24	0.24867	0.25279	0.24811	0.23636	0.21156
25	0.24695	0.23840	0.23645	0.23089	0.20542

AGE	1961	1962	1963	1964	1965
26	0.23181	0.23226	0.22497	0.21610	0.19444
27	0.21640	0.21513	0.21371	0.20254	0.18336
28	0.20774	0.20268	0.19804	0.19179	0.16914
29	0.19426	0.18551	0.17999	0.17388	0.15578
30	0.18197	0.17499	0.16951	0.16178	0.14525
31	0.16043	0.15678	0.15189	0.14052	0.12403
32	0.14322	0.14827	0.14392	0.13979	0.12086
33	0.12806	0.12415	0.12803	0.12299	0.10988
34	0.11290	0.11313	0.10964	0.11061	0.09794
35	0.10093	0.09699	0.09774	0.09135	0.08710
36	0.09280	0.08902	0.08751	0.08337	0.07407
37	0.08019	0.07704	0.07417	0.07161	0.06510
38	0.07091	0.06648	0.06605	0.06250	0.05758
39	0.05926	0.05562	0.05400	0.05182	0.04617
40	0.04941	0.04637	0.04292	0.04145	0.03637
41	0.03608	0.03451	0.03186	0.03021	0.02690
42	0.02586	0.02772	0.02541	0.02479	0.02160
43	0.01796	0.01671	0.01741	0.01664	0.01408
44	0.01039	0.00924	0.00906	0.01004	0.00913
45	0.00593	0.00527	0.00541	0.00571	0.00570
46	0.00290	0.00286	0.00281	0.00277	0.00239
47	0.00160	0.00135	0.00103	0.00111	0.00104
48	0.00070	0.00042	0.00054	0.00044	0.00039
49	0.00024	0.00020	0.00017	0.00018	0.00016
50	0.00019	0.00012	0.00012	0.00010	0.00015

AGE	1966	1967	1968	1969	1970
0	0.00000	0.00000	0.00000	0.00000	0.00000
1	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000
5	0.00000	0.00000	0.00000	0.00000	0.00000
6	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000
10	0.00000	0.00000	0.00000	0.00000	0.00000
11	0.00000	0.00000	0.00000	0.00000	0.00000
12	0.00002	0.00001	0.00001	0.00000	0.00001
13	0.00016	0.00017	0.00014	0.00017	0.00014
14	0.00098	0.00096	0.00111	0.00114	0.00126
15	0.00464	0.00468	0.00505	0.00564	0.00584
16	0.01653	0.01573	0.01582	0.01667	0.01935
17	0.03882	0.03667	0.03662	0.03746	0.04043
18	0.07227	0.06925	0.06552	0.06401	0.06372
19	0.10695	0.10517	0.09698	0.09138	0.08859
20	0.13264	0.14373	0.12648	0.11610	0.10818
21	0.16135	0.15035	0.15647	0.13970	0.12824
22	0.17508	0.15950	0.15495	0.16619	0.15044
23	0.18944	0.17472	0.15847	0.15936	0.17190
24	0.19392	0.18329	0.17154	0.16253	0.16380
25	0.18263	0.17503	0.17163	0.16900	0.15824

AGE	1966	1967	1968	1969	1970
26	0.17603	0.16373	0.16252	0.16630	0.16029
27	0.15867	0.15131	0.14802	0.14816	0.15297
28	0.15264	0.14219	0.13730	0.13954	0.13908
29	0.14438	0.12729	0.12102	0.12270	0.12261
30	0.11581	0.11229	0.11116	0.11128	0.10928
31	0.11723	0.09970	0.09190	0.09287	0.09089
32	0.10141	0.09064	0.08699	0.08265	0.08156
33	0.09406	0.08224	0.07469	0.07220	0.06740
34	0.08783	0.07407	0.06717	0.06594	0.05888
35	0.07136	0.06735	0.05829	0.05665	0.05354
36	0.06860	0.05859	0.05199	0.04850	0.04373
37	0.05513	0.05243	0.04472	0.04243	0.03768
38	0.04906	0.04236	0.03912	0.03645	0.03408
39	0.04239	0.03446	0.03090	0.02976	0.02623
40	0.02875	0.02688	0.02456	0.02230	0.02028
41	0.02639	0.02035	0.01688	0.01599	0.01445
42	0.01736	0.01499	0.01383	0.01188	0.01127
43	0.01319	0.01060	0.00877	0.00801	0.00686
44	0.00816	0.00643	0.00526	0.00466	0.00390
45	0.00408	0.00367	0.00316	0.00269	0.00242
46	0.00226	0.00183	0.00195	0.00139	0.00114
47	0.00112	0.00078	0.00091	0.00074	0.00052
48	0.00034	0.00040	0.00041	0.00025	0.00020
49	0.00018	0.00024	0.00007	0.00010	0.00006
50	0.00010	0.00005	0.00003	0.00005	0.00002

AGE	1971	1972	----	----	----
0	0.00000	0.00000			
1	0.00000	0.00000			
2	0.00000	0.00000			
3	0.00000	0.00000			
4	0.00000	0.00000			
5	0.00000	0.00000			
6	0.00000	0.00000			
7	0.00000	0.00000			
8	0.00000	0.00000			
9	0.00000	0.00000			
10	0.00000	0.00000			
11	0.00000	0.00000			
12	0.00002	0.00002			
13	0.00015	0.00016			
14	0.00120	0.00118			
15	0.00564	0.00570			
16	0.01776	0.01838			
17	0.03754	0.03804			
18	0.06080	0.05890			
19	0.08345	0.07502			
20	0.10442	0.08924			
21	0.12262	0.10313			
22	0.14144	0.12232			
23	0.15147	0.13495			
24	0.15205	0.15218			
25	0.16623	0.16528			

AGE	1971	1972	----	----	----
26	0.14962	0.14523			
27	0.14599	0.13065			
28	0.12846	0.12538			
29	0.11764	0.11361			
30	0.10190	0.09748			
31	0.08717	0.08062			
32	0.07557	0.07049			
33	0.06349	0.05841			
34	0.05551	0.04924			
35	0.04655	0.03971			
36	0.04106	0.03522			
37	0.03268	0.02849			
38	0.02665	0.02304			
39	0.02152	0.01820			
40	0.01770	0.01422			
41	0.01135	0.00977			
42	0.00857	0.00785			
43	0.00570	0.00454			
44	0.00370	0.00246			
45	0.00183	0.00148			
46	0.00063	0.00079			
47	0.00037	0.00033			
48	0.00017	0.00015			
49	0.00003	0.00007			
50	0.00004	0.00003			

F(T)

3.95867	3.92931	3.88775	3.94681	3.91024
3.85705	3.77677	3.68988	3.52103	3.16257
2.81197	2.60413	2.46241	2.41283	2.33948
2.18870	2.02197			

AGE	1956	1957	1958	1959	1960
0	0.00000	0.00000	0.00000	0.00000	0.00000
1	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000
5	0.00000	0.00000	0.00000	0.00000	0.00000
6	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000
10	0.00000	0.00000	0.00000	0.00000	0.00000
11	0.00000	0.00000	0.00000	0.00001	0.00000
12	0.00001	0.00002	0.00003	0.00002	0.00002
13	0.00016	0.00022	0.00015	0.00015	0.00016
14	0.00101	0.00102	0.00127	0.00127	0.00115
15	0.00494	0.00505	0.00535	0.00560	0.00512
16	0.01727	0.01962	0.01884	0.01935	0.01920
17	0.04615	0.04878	0.04807	0.04838	0.04882
18	0.08778	0.09214	0.09111	0.09325	0.09206
19	0.12879	0.13800	0.14009	0.14010	0.14344
20	0.17339	0.17773	0.18410	0.18473	0.18793
21	0.20960	0.21342	0.21289	0.22062	0.22423
22	0.22663	0.23168	0.23717	0.23768	0.24214
23	0.24622	0.24286	0.24534	0.25096	0.24783
24	0.25360	0.24877	0.24499	0.25038	0.25203
25	0.24748	0.24826	0.24515	0.24130	0.24590

AGE	1956	1957	1958	1959	1960
26	0.23691	0.23534	0.24034	0.23375	0.23064
27	0.21692	0.22387	0.22230	0.22534	0.22281
28	0.21169	0.20671	0.21284	0.21339	0.21299
29	0.18926	0.18830	0.18930	0.19574	0.19627
30	0.18093	0.17550	0.17880	0.17595	0.18464
31	0.15524	0.15287	0.15223	0.15342	0.14982
32	0.15117	0.14849	0.14682	0.14225	0.14248
33	0.13567	0.13466	0.13395	0.13046	0.12863
34	0.12787	0.12171	0.12205	0.12046	0.11721
35	0.11527	0.11247	0.10817	0.10594	0.10568
36	0.10455	0.10366	0.10014	0.09405	0.09237
37	0.08493	0.09144	0.08827	0.08589	0.08025
38	0.07791	0.07362	0.07797	0.07667	0.07419
39	0.06246	0.06127	0.05778	0.06209	0.06159
40	0.05043	0.04958	0.04844	0.04549	0.05065
41	0.03700	0.03557	0.03323	0.03418	0.03147
42	0.03078	0.03062	0.02818	0.02762	0.02667
43	0.02019	0.02014	0.01854	0.01803	0.01807
44	0.01299	0.01208	0.01212	0.01141	0.01073
45	0.00723	0.00647	0.00662	0.00709	0.00612
46	0.00347	0.00364	0.00332	0.00283	0.00309
47	0.00140	0.00171	0.00163	0.00162	0.00129
48	0.00079	0.00079	0.00069	0.00082	0.00054
49	0.00040	0.00028	0.00023	0.00026	0.00021
50	0.00018	0.00027	0.00016	0.00013	0.00018

AGE	1961	1962	1963	1964	1965
0	0.00000	0.00000	0.00000	0.00000	0.00000
1	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000
5	0.00000	0.00000	0.00000	0.00000	0.00000
6	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000
10	0.00000	0.00000	0.00000	0.00000	0.00000
11	0.00000	0.00000	0.00000	0.00000	0.00001
12	0.00004	0.00000	0.00002	0.00002	0.00002
13	0.00015	0.00016	0.00012	0.00014	0.00021
14	0.00123	0.00110	0.00097	0.00109	0.00104
15	0.00522	0.00534	0.00537	0.00492	0.00534
16	0.01868	0.01767	0.02012	0.01850	0.02001
17	0.04758	0.04392	0.04507	0.04964	0.05059
18	0.09288	0.08923	0.08457	0.08562	0.10010
19	0.14145	0.14218	0.13902	0.13140	0.13866
20	0.19179	0.19074	0.18957	0.18244	0.17505
21	0.22485	0.23009	0.22576	0.22552	0.21949
22	0.24897	0.24835	0.25208	0.24668	0.24899
23	0.25696	0.26007	0.26022	0.25914	0.25692
24	0.24877	0.25827	0.25946	0.25903	0.25812
25	0.24705	0.24357	0.24726	0.25303	0.25063

AGE	1961	1962	1963	1964	1965
26	0.23191	0.23729	0.23526	0.23682	0.23723
27	0.21649	0.21979	0.22349	0.22196	0.22372
28	0.20783	0.20708	0.20710	0.21018	0.20637
29	0.19434	0.18953	0.18822	0.19055	0.19007
30	0.18205	0.17878	0.17726	0.17729	0.17722
31	0.16049	0.16018	0.15884	0.15399	0.15133
32	0.14328	0.15148	0.15050	0.15319	0.14747
33	0.12812	0.12684	0.13389	0.13478	0.13406
34	0.11295	0.11558	0.11466	0.12122	0.11950
35	0.10097	0.09910	0.10221	0.10011	0.10627
36	0.09284	0.09095	0.09151	0.09137	0.09038
37	0.08022	0.07871	0.07757	0.07848	0.07943
38	0.07094	0.06793	0.06907	0.06850	0.07025
39	0.05928	0.05683	0.05647	0.05679	0.05633
40	0.04943	0.04737	0.04488	0.04543	0.04438
41	0.03610	0.03526	0.03332	0.03310	0.03282
42	0.02587	0.02832	0.02658	0.02717	0.02635
43	0.01796	0.01707	0.01821	0.01823	0.01717
44	0.01040	0.00944	0.00948	0.01100	0.01114
45	0.00593	0.00538	0.00566	0.00626	0.00695
46	0.00290	0.00292	0.00294	0.00303	0.00292
47	0.00160	0.00138	0.00107	0.00121	0.00127
48	0.00070	0.00043	0.00056	0.00049	0.00048
49	0.00024	0.00020	0.00018	0.00020	0.00019
50	0.00019	0.00012	0.00012	0.00011	0.00018

AGE	1966	1967	1968	1969	1970
0	0.00000	0.00000	0.00000	0.00000	0.00000
1	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000
5	0.00000	0.00000	0.00000	0.00000	0.00000
6	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000
10	0.00000	0.00000	0.00000	0.00000	0.00000
11	0.00000	0.00000	0.00000	0.00000	0.00000
12	0.00002	0.00002	0.00002	0.00000	0.00002
13	0.00022	0.00025	0.00022	0.00027	0.00023
14	0.00134	0.00142	0.00173	0.00182	0.00208
15	0.00637	0.00694	0.00791	0.00902	0.00962
16	0.02268	0.02330	0.02479	0.02665	0.03192
17	0.05328	0.05434	0.05738	0.05990	0.06669
18	0.09918	0.10262	0.10268	0.10237	0.10510
19	0.14676	0.15583	0.15198	0.14613	0.14612
20	0.18202	0.21298	0.19820	0.18567	0.17843
21	0.22142	0.22278	0.24519	0.22341	0.21151
22	0.24024	0.23633	0.24281	0.26578	0.24813
23	0.25995	0.25889	0.24832	0.25486	0.28352
24	0.26611	0.27160	0.26881	0.25993	0.27017
25	0.25061	0.25935	0.26895	0.27028	0.26099

AGE	1966	1967	1968	1969	1970
26	0.24155	0.24261	0.25468	0.26596	0.26438
27	0.21773	0.22421	0.23195	0.23694	0.25231
28	0.20946	0.21069	0.21515	0.22315	0.22939
29	0.19813	0.18862	0.18964	0.19622	0.20223
30	0.15892	0.16638	0.17419	0.17797	0.18024
31	0.16086	0.14773	0.14402	0.14851	0.14991
32	0.13916	0.13431	0.13631	0.13218	0.13452
33	0.12908	0.12185	0.11704	0.11546	0.11116
34	0.12052	0.10975	0.10526	0.10546	0.09712
35	0.09792	0.09980	0.09135	0.09060	0.08831
36	0.09413	0.08681	0.08147	0.07756	0.07212
37	0.07565	0.07769	0.07008	0.06785	0.06215
38	0.06732	0.06276	0.06130	0.05829	0.05621
39	0.05817	0.05106	0.04842	0.04759	0.04326
40	0.03946	0.03982	0.03848	0.03567	0.03345
41	0.03622	0.03015	0.02645	0.02557	0.02383
42	0.02382	0.02221	0.02167	0.01900	0.01860
43	0.01810	0.01571	0.01374	0.01281	0.01131
44	0.01119	0.00952	0.00825	0.00745	0.00643
45	0.00560	0.00543	0.00496	0.00430	0.00400
46	0.00310	0.00272	0.00305	0.00222	0.00188
47	0.00153	0.00115	0.00142	0.00118	0.00086
48	0.00046	0.00059	0.00065	0.00040	0.00032
49	0.00024	0.00036	0.00010	0.00016	0.00010
50	0.00014	0.00007	0.00005	0.00008	0.00003

AGE	1971	1972	----	----	----
0	0.00000	0.00000			
1	0.00000	0.00000			
2	0.00000	0.00000			
3	0.00000	0.00000			
4	0.00000	0.00000			
5	0.00000	0.00000			
6	0.00000	0.00000			
7	0.00000	0.00000			
8	0.00000	0.00000			
9	0.00000	0.00000			
10	0.00000	0.00000			
11	0.00000	0.00001			
12	0.00003	0.00003			
13	0.00026	0.00030			
14	0.00211	0.00225			
15	0.00994	0.01088			
16	0.03131	0.03507			
17	0.06619	0.07260			
18	0.10720	0.11241			
19	0.14711	0.14316			
20	0.18409	0.17030			
21	0.21618	0.19681			
22	0.24935	0.23344			
23	0.26703	0.25753			
24	0.26806	0.29041			
25	0.29306	0.31542			

AGE	1971	1972
26	0.26379	0.27714
27	0.25739	0.24933
28	0.22648	0.23926
29	0.20739	0.21680
30	0.17965	0.18602
31	0.15369	0.15384
32	0.13324	0.13453
33	0.11194	0.11147
34	0.09787	0.09396
35	0.08207	0.07579
36	0.07238	0.06721
37	0.05762	0.05436
38	0.04699	0.04398
39	0.03795	0.03474
40	0.03121	0.02714
41	0.02002	0.01865
42	0.01510	0.01499
43	0.01005	0.00866
44	0.00652	0.00469
45	0.00323	0.00283
46	0.00111	0.00151
47	0.00064	0.00062
48	0.00029	0.00029
49	0.00006	0.00014
50	0.00008	0.00007

2.2 Estimation of the Fertility Parameters

A matrix $M(=M(i,j))$ representing approximate values of $m(x,t)$ at $i \in [0,50]$, and $j \in [1956, 1972]$ is obtained using an algorithm based on formula (4a) as follows:

1. $\bar{f}(x,t)$ is approximated by an interpolatory cubic spline $\bar{f}_3(x,t)$ as a function of t for each x. (This interpolates the data points $\bar{f}(x_i,t_j)$ satisfying boundary conditions so that third-order derivatives are approximated by third-order finite differences at $x_0 - \frac{h}{2}$ and $x_n - \frac{3h}{2}$.*)

2. Exact values of $\frac{d\bar{f}_3}{dt}(x,t)$ are obtained for x and t in single years: $x \in (0,50)$, $t \in [1956, 1972]$.

3. $\frac{d\bar{f}_3}{dt}(x,t)$ is approximated by an interpolatory cubic spline $\left[\frac{d\bar{f}_3}{dt}\right]_3(x,t)$ as a function of x for each t with boundary conditions as above.

---

*The selection of appropriate boundary conditions caused considerable difficulty. This choice appears to yield acceptable derivatives of the data sets used, and has been used for all cubic spline approximations.

4. M is the result obtained on replacing $\frac{d\bar{f}}{dt}$ of formula (4a) by the cubic spline $\left[\frac{d\bar{f}}{dt}\right]_3$ and integrating exactly.

The array M (denoted by G in output of program 2.2) forms input via a computer file to program 2.3.

```
$JOB   ACCT-NUM,VERNER,TIME=60
C      *******************************************************************
C
C             PROGRAM 2.2 :   POP1
C
C      *******************************************************************
```

```
3JOB POP1; CHARGE BERGHOUT;CLASS 3;
USER 080002/VHJ;
PROCESSTIME=600;
PRINTLIMIT=13200;
BEGIN
COMPILE POP1 FORTRAN;
FILE FILE3 (KIND=PACK,TITLE=DATA/XXG,MAXRECSIZE=30,BLOCKSIZE=990,
AREASIZE=33,AREAS=8,SAVEFACTOR=99);
OPTION=AUTORM;
DATA
```

```
3DATA FILE5
```

```
3END JOB
```

```
C        ****************************************************************************
C
C        PROGRAM : 2.2
C        TITLE : POP1
C        PURPOSE : CALCULATE N(X,T)
C        DATA : FBAR(0->50,1->17)
C        OUTPUT : DATA/XXG ON DISKPACK
C
C        ****************************************************************************
C
C
C        HOW TO SET UP A RUN
C        === == === == = ===
C
C    1)  PREPARE A NAMELIST CALLED VALUES SPECIFYING THE
C        FOLLOWING PARAMETERS WHICH ARE USED FOR OUTPUT REQUESTS
C        AND FOR INITIALIZATION OF VARIABLES.
C
C        LFBAR    = .TRUE. => LIST FBAR
C        LFBARS   = .TRUE. => LIST FBARST
C        LG       = .TRUE. => LIST G
C        NT       = # OF YEARS TO BE COVERED
C        SAGE     = STARTING AGE + 1
C        EAGE     = ENDING AGE + 1
C
C        FOR EXAMPLE USE FOR DATA THE FOLLOWING INFORMATION (OMIT C)
C        PUNCHED ON CARDS BEGINNING IN COLUMN 2
C        &VALUES
C        LG=.TRUE.,LFBAR=.TRUE.,LFBARS=.TRUE.,
C        &END
C        &VALUS1
C        NT=10,SAGE=16,EAGE=48,&END
C
C    2)  IF NT IS GREATER THAN 10 THE SECOND DIMENSION OF FBAR,
C        FBARST AND G MUST CHANGED TO CORRESPOND TO THE VALUE OF NT.
C
C    3)  IF EAGE-SAGE+1 IS GREATER THAN 33 THE FIRST DIMENSION OF
C        FBAR,FBARST AND G MUST CHANGED TO CORRESPOND TO THIS NEW VALUE.
C
C        THE DIMENSIONS OF G, FBAR AND FBARST MUST BE
C        CHANGED IN THE DECLARATION STATEMENT OF THE MAIN PROGRAM,
C        BUT NOT IN THE SUBROUTINES.
C
C
         REAL FBAR(33,10),FBARST(33,10),G(33,10),DATA(51,17)
         LOGICAL LG,LFBAR,LFBARS
         INTEGER SAGE,EAGE,NX,NT
         NAMELIST/VALUS1/NT,SAGE,EAGE
         NAMELIST/VALUES/LG,LFBAR,LFBARS
         READ(5,100)((DATA(I,J),I=1,51),J=1,17)
     100 FORMAT(4E17.7)
         LG = .FALSE.
         LFBAR = .FALSE.
         LFBARS = .FALSE.
         READ(5,VALUES)
         WRITE(6,VALUES)
         READ(5,VALUS1)
         WRITE(6,VALUS1)
         NX = EAGE - SAGE + 1
```

```
      IX = 17 - NT + 1
      DO 1 I = 1,IX
      II = I + NT - 1
      DO 2 J = I,II
         NN = J - I + 1
         DO 2 K = SAGE,EAGE
         MM = K - SAGE + 1
    2    FBAR(MM,NN) = DATA(K,J)
      CALL OBS(NX,NT,FBAR,FBARST,G)
      DO 3 K = 1,NX
    3 WRITE(3)(G(K,J),J=1,NT)
      IF (LFBAR) CALL DUMPS(1,NX,NT,FBAR,I,SAGE,EAGE)
      IF (LFBARS) CALL DUMPS(7,NX,NT,FBARST,I,SAGE,EAGE)
      IF (LG) CALL DUMPS(5,NX,NT,G,I,SAGE,EAGE)
    1 CONTINUE
      LOCK 3

      STOP
      END

      SUBROUTINE DUMPS(LL,NX,NT,FBAR,N,ISAGE,IEAGE)
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C   DUMPS PURPOSE IS TO LIST A MATRIX IN A RADABLE FORM WITH A PAGE # &
C   HEADING.  THE HEADING IS CHOOSEN ACCORDING TO THE VALUE OF LL:
C        LL          HEADING
C        ==          =======
C         1          FBAR
C         3          D(X)A(T)
C         5          G(X,T)
C         7          FBARST
C
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INTEGER YEAR(17),HI,LO , AGE
      INTEGER TITLE(8)/'FBAR','    ','D(X)','A(T)','G(X,',',T)   ','FBAR',
     * 'ST  '/
      REAL FBAR(NX,NT)
      IAGE = ISAGE - 2
      DO 1 I = 1, 17
    1    YEAR(I) = 1955 + I
      LLL = NT / 5
      IF( MOD(NT,5 ) .NE. 0 ) LLL = LLL + 1
      IPAGE = 0
      HI = 0
      DO 5 IE = 1 , LLL
      LO = HI + 1
      HI = HI + 5
      IF( HI .GT. NT ) HI = NT
      DO 5 I = 1,NX
         AGE = I + IAGE
         IF( MOD(I,20) .NE. 1 ) GO TO 5
         IPAGE  = IPAGE + 1
         WRITE(6,100) TITLE(LL), TITLE (LL+1), IPAGE,
     *                ( YEAR( J+N-1), J = LO,HI )
    5 WRITE(6,101) AGE, ( FBAR(I,J), J = LO, HI)
  100 FORMAT('1',1X,///,15X,2A4,10X,'Y E A R',T57,'PAGE ',I2,//,
     &  5X,'AGE',2X,5(3X,'____',4X),/,'+',9X,5(I7,4X),///)
  101 FORMAT(/,5X,I3,  5F11.5)
      RETURN
      END
```

```
      SUBROUTINE OBS(NX,NT,FBAR,FBARST,G)
C     ****************************************************************
C     THIS ROUTINE COMPUTES THE 'OBSERVATIONS' OF THE FORM
C     INTEGRAL FROM 0 TO X OF -D/DT(FBAR(X,T)) ALL DIVIDED BY FBAR(X,T)
C      THE ARRAY G STORES THE OBSERVATIONS
C     ****************************************************************
      REAL K(2,33),A(3,33),H(33),COF(4,33),D(33)
      REAL FBAR(NX,NT),FBARST(NX,NT),G(NX,NT)
      DT=1.
      DO 5 I=1,NT
    5     K(1,I)=DT*(I-1.)
      CALL SETUP(NT,A,H,K)
      DO 25 IX=1,NX
          DO 15 IT=1,NT
   15         K(2,IT)=FBAR(IX,IT)
          CALL SOLVE(A,K,H,NT,D,COF)
          DO 24 J=1,NT
              T=DT*(J-1.)
              FBARST(IX,J)=3.*COF(1,J)*T*T+2.*COF(2,J)*T+COF(3,J)
   24         CONTINUE
   25     CONTINUE
      DO 35 I=1,NX
          K(1,I)=I-1.
   35     CONTINUE
      CALL SETUP(NX,A,H,K)
      DO 55 IT=1,NT
          DO 45 IX=1,NX
   45         K(2,IX)=FBARST(IX,IT)
          CALL SOLVE(A,K,H,NX,D,COF)
          NXM1=NX-1
          DD=0.
C    PROGRAM WILL INTEGRATE FORWARD OR BACKWARDS BY FLIPING CARDS MARKED
C    CB(BACKWARD INTEGRATION) AND CF(FORWARD INTEGRATION)
C    IF ALL CB CARDS COMMENT THEN FORWARD INTEGRATION
          G(1,IT)=0.
CB                                                            0.0 = )TI,XN(G
            FBARST( 1,IT) = 0.0
CB                                                      0.0 = )TI,XN(TSRABF
          X=0.
CB                                                                    XN = X
CB                                                                    XN = J
          DO 55 J=1,NXM1
CB                                                               1MXN,1=L 55 OD
CB                                                                 1 - J = J
              DD=DD-X*(COF(4,J)+X*(COF(3,J)/2.+X*(COF(2,J)/3.+
     2        X*COF(1,J)/4.)))
              X=X+1.
CB                                                                 1 - X = X
              DD=DD+X*(COF(4,J)+X*(COF(3,J)/2.+X*(COF(2,J)/3.+
     2        X*COF(1,J)/4.)))
              G(J+1,IT)=-DD/FBAR(J+1,IT)
CB                                         )TI,  J(RABF/DD-=)TI,  J(G
              FBARST(J+1,IT) = DD
CB                                                  DD = )TI,J(TSRABF
   55         CONTINUE
      RETURN
      END
      SUBROUTINE SETUP(N,A,H,K)
C
```

```
C     ******************************************************************
C     THIS ROUTINE SETS UP A TRIDIAGONAL MATRIX A OF THE SPLINE EQUATIO
C     IN A  3 BY N  ARRAY  -  FIRST ROW IS SUPERDIAGONAL
C                             SECOND ROW IS DIAGONAL
C                             THIRD ROW IS SUBDIAGONAL
C     AND THEN DECOMPOSES A TO  LU  SO THAT
C        FIRST ROW IS SUPERDIAGONAL OF U, SECOND ROW IS DIAGONAL OF U
C       THIRD ROW IS SUBDIAGONAL OF L, DIAGONAL OF L IS UNITY
C     ******************************************************************
C
      REAL A(3,N),H(N),K(2,N)
      NM1=N-1
      H(2)=K(1,2)-K(1,1)
      DO 10 I=2,NM1
           H(I+1)=K(1,I+1)-K(1,I)
           A(1,I)=H(I+1)/(H(I+1)+H(I))
           A(2,I)=2.0
           A(3,I)=1-A(1,I)
 10        CONTINUE
      A(1,1)  =  A(1,2)
      A(2,1)  =  A(2,2)
      A(3,1)  =  A(3,2)
      A(1,N)  =  A(1,N-1)
      A(2,N)  =  A(2,N-1)
      A(3,N)  =  A(3,N-1)
      A(1,2)  =  -2
      A(2,2)  =  2
      A(3,2)  =  0
      A(1,N-1)  =  0
      A(2,N-1)  =  2
      A(3,N-1)  =  -2
      LL  =  N - 1
      DO 11  I  =  3, LL
           A(3,I)=A(3,I)/A(2,I-1)
           A(2,I)=A(2,I)-A(3,I)*A(1,I-1)
 11        CONTINUE
      RETURN
      END

      SUBROUTINE SOLVE(A,K,H,N,D,COF)
C
C     ******************************************************************
C     THIS ROUTINE CALCULATES THE SECOND ORDER FINITE DIFFERENCES OF THE
C     SPLINE, AND THEN SOLVES   AM=D   , (BY FORWARD AND BACKWARD
C     SUBSTITUTION), PLACING M(VECTOR OF SECOND DERIVATIVES) IN D
C     END POINT CONDITIONS NOW USE THIRD ORDER FINITE DIFFERENCES
C     TO ESTIMATE THIRD ORDER DERIVATIVES AT X0+3H/2 AND XN-3H/2
C     ******************************************************************
C
      REAL H(N),D(N),A(3,N),K(2,N),COF(4,N)
      D(2)=(K(2,2)-K(2,1))/H(2)
      NM1=N-1
      DO 12 I=2,NM1
           D(I+1)=(K(2,I+1)-K(2,I))/H(I+1)
           D(I)=6*(D(I+1)-D(I))/(H(I+1)+H(I))
 12        CONTINUE
      D(1) =-2.*(-K(2,1)+K(2,4)+3*(K(2,2)-K(2,3)))/(H(2)*H(2))
      D(N) =-2.*(-K(2,N)+K(2,N-3)+3.*(K(2,N-1)-K(2,N-2)))/(H(N-1)**2)
      T1  =  D(2)
      T2  =  D(N-1)
```

```
      D(2) = D(1)
      D(N-1) = D(N)
      LL = N - 1
      DO 13 I = 3, LL
         D(I)=D(I)-A(3,I)*D(I-1)
 13      CONTINUE
      D(N-1) = D(N-1) / A(2,N-1)
      DO 14 I = 3,LL
         J=N+1-I
         D(J)=(D(J)-A(1,J)*D(J+1))/A(2,J)
 14      CONTINUE
      D(1) = ( T1 - A(2,1) * D(2) - A(1,1) *.D(3) ) / A(3,1)
      D(N ) = ( T2 - A(2,N) * D(LL) - A(3,N) * D(N-2) ) / A(1,N)
      CALL POLLY(N,D,K,H,COF)
      RETURN
      END

      SUBROUTINE POLLY(N,M,K,H,COF)
C
C     ****************************************************************
C     THIS ROUTINE COMPUTES THE COEFFICIENTS OF THE SPLINE POLYNOMIAL
C        ON EACH SUBINTERVAL
C     K IS THE ARRAY OF DATA POINTS
C     H IS THE VECTOR OF SUBINTERVAL LENGTHS
C     M IS THE SOLUTION VECTOR TO THE EQUATION AM=D
C     ****************************************************************
C
      REAL M(N),K(2,N),H(N),COF(4,N)
      NM1=N-1
      DO 11 I=1,NM1
      COF(1,I)=(M(I+1)-M(I))/(6.*H(I+1))
      COF(2,I)=(K(1,I+1)*M(I)-K(1,I)*M(I+1))/(2.*H(I+1))
      DD=M(I+1)*K(1,I)*K(1,I)-M(I)*K(1,I+1)*K(1,I+1)
     &+2.*K(2,I+1)-2.*K(2,I)
      COF(3,I)=(DD/(2.*H(I+1)))+H(I+1)*(M(I)-M(I+1))/6.
      DD=M(I)*(K(1,I+1)**3)-M(I+1)*(K(1,I)**3)+6.*K(1,I+1)*K(2,I)
     &-6.*K(1,I)*K(2,I+1)+K(1,I)*M(I+1)*(H(I+1)**2)-K(1,I+1)*M(I)*
     &(H(I+1)**2)
      COF(4,I)=DD/(H(I+1)*6.)
 11   CONTINUE
      DO 16 J=1,4
 16   COF(J,N)=COF(J,NM1)
      RETURN
         END
```

```
C     **********************************************************************
C
C     DATA FOR
C          PROGRAM 2.2 :   POP1
C
C     **********************************************************************
    0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
    0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
    0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
    0.7429816E-05    0.1634927E-03    0.1006141E-02    0.4942421E-02
    0.1726509E-01    0.4614777E-01    0.8778054E-01    0.1287903E 00
    0.1733888E 00    0.2095979E 00    0.2266342E 00    0.2462188E 00
    0.2535967E 00    0.2474782E 00    0.2369086E 00    0.2169245E 00
    0.2116948E 00    0.1892552E 00    0.1809314E 00    0.1552396E 00
    0.1511665E 00    0.1356698E 00    0.1278709E 00    0.1152725E 00
    0.1045548E 00    0.8492929E-01    0.7790750E-01    0.6245709E-01
    0.5043102E-01    0.3700228E-01    0.3078149E-01    0.2018753E-01
    0.1298582E-01    0.7232107E-02    0.3469116E-02    0.1403072E-02
    0.7913960E-03    0.4042918E-03    0.1762484E-03    0.0000000E 00
    0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
    0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
    0.0000000E 00    0.0000000E 00    0.0000000E 00    0.2033752E-04
    0.2206279E-03    0.1024708E-02    0.5051389E-02    0.1962170E-01
    0.4878248E-01    0.9214234E-01    0.1379995E 00    0.1777333E 00
    0.2134188E 00    0.2316796E 00    0.2428612E 00    0.2487651E 00
    0.2482604E 00    0.2353446E 00    0.2238681E 00    0.2067142E 00
    0.1883027E 00    0.1754966E 00    0.1528666E 00    0.1484947E 00
    0.1346632E 00    0.1217132E 00    0.1124704E 00    0.1036606E 00
    0.9143513E-01    0.7362139E-01    0.6127410E-01    0.4957513E-01
    0.3556923E-01    0.3062474E-01    0.2014163E-01    0.1207764E-01
    0.6471954E-02    0.3641624E-02    0.1707581E-02    0.7939595E-03
    0.2838371E-03    0.2689431E-03    0.0000000E 00    0.0000000E 00
    0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
    0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
    0.0000000E 00    0.0000000E 00    0.2588560E-04    0.1491891E-03
    0.1273050E-02    0.5347177E-02    0.1883900E-01    0.4806599E-01
    0.9110671E-01    0.1400929E 00    0.1840963E 00    0.2128937E 00
    0.2371737E 00    0.2453404E 00    0.2449850E 00    0.2451515E 00
    0.2403407E 00    0.2223017E 00    0.2128390E 00    0.1893021E 00
    0.1788046E 00    0.1522348E 00    0.1468166E 00    0.1339506E 00
    0.1220511E 00    0.1081653E 00    0.1001368E 00    0.8826882E-01
    0.7796538E-01    0.5777969E-01    0.4844318E-01    0.3322498E-01
    0.2818455E-01    0.1854310E-01    0.1211527E-01    0.6617039E-02
    0.3324505E-02    0.1627151E-02    0.6927683E-03    0.2283992E-03
    0.1625607E-03    0.0000000E 00    0.0000000E 00    0.0000000E 00
    0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
    0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
    0.5848482E-05    0.1815752E-04    0.1524006E-03    0.1269283E-02
    0.5595200E-02    0.1934955E-01    0.4837923E-01    0.9324521E-01
    0.1400996E 00    0.1847297E 00    0.2206196E 00    0.2376751E 00
```

0.2509574E 00	0.2503754E 00	0.2412987E 00	0.2337543E 00
0.2253411E 00	0.2133908E 00	0.1957424E 00	0.1759517E 00
0.1534150E 00	0.1422451E 00	0.1304606E 00	0.1204642E 00
0.1059424E 00	0.9405357E-01	0.8589458E-01	0.7667202E-01
0.6208997E-01	0.4548723E-01	0.3417568E-01	0.2762331E-01
0.1803264E-01	0.1140561E-01	0.7085640E-02	0.2832796E-02
0.1616091E-02	0.8214479E-03	0.2615624E-03	0.1299269E-03
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.2325392E-04	0.1646105E-03	0.1154704E-02	0.5122337E-02
0.1919859E-01	0.4882310E-01	0.9206295E-01	0.1434420E 00
0.1879300E 00	0.2242298E 00	0.2421390E 00	0.2478324E 00
0.2520348E 00	0.2459021E 00	0.2306418E 00	0.2228054E 00
0.2129852E 00	0.1962739E 00	0.1846426E 00	0.1498247E 00
0.1424794E 00	0.1286335E 00	0.1172121E 00	0.1056809E 00
0.9236968E-01	0.8025026E-01	0.7419306E-01	0.6159350E-01
0.5064670E-01	0.3146533E-01	0.2666571E-01	0.1807094E-01
0.1073283E-01	0.6116446E-02	0.3089603E-02	0.1289498E-02
0.5448801E-03	0.2109780E-03	0.1840822E-03	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.3936497E-04
0.1471253E-03	0.1234930E-02	0.5219840E-02	0.1867957E-01
0.4758165E-01	0.9287566E-01	0.1414530E 00	0.1917900E 00
0.2248529E 00	0.2489690E 00	0.2569618E 00	0.2487700E 00
0.2470545E 00	0.2319062E 00	0.2164945E 00	0.2078308E 00
0.1943436E 00	0.1820471E 00	0.1604927E 00	0.1432767E 00
0.1281160E 00	0.1129494E 00	0.1009693E 00	0.9283632E-01
0.8022434E-01	0.7094383E-01	0.5928110E-01	0.4942951E-01
0.3609700E-01	0.2586606E-01	0.1796386E-01	0.1039849E-01
0.5931694E-02	0.2902867E-02	0.1597295E-02	0.7027730E-03
0.2392092E-03	0.1916328E-03	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.1611304E-03
0.1100139E-02	0.5341694E-02	0.1767021E-01	0.4392144E-01
0.8923173E-01	0.1421814E 00	0.1907406E 00	0.2300875E 00
0.2483481E 00	0.2600701E 00	0.2582705E 00	0.2435674E 00
0.2372926E 00	0.2197912E 00	0.2070794E 00	0.1895335E 00
0.1787836E 00	0.1601778E 00	0.1514808E 00	0.1268388E 00
0.1155834E 00	0.9909540E-01	0.9094930E-01	0.7870620E-01
0.6792599E-01	0.5682655E-01	0.4737460E-01	0.3526177E-01
0.2832320E-01	0.1706897E-01	0.9436630E-02	0.5381081E-02
0.2923402E-02	0.1375800E-02	0.4280740E-03	0.2046375E-03
0.1230038E-03	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.2249860E-04	0.1204436E-03	0.9681717E-03
0.5371928E-02	0.2012049E-01	0.4507285E-01	0.8456761E-01

0.1390227E 00	0.1895710E 00	0.2257624E 00	0.2520762E 00
0.2602199E 00	0.2594635E 00	0.2472610E 00	0.2352608E 00
0.2234864E 00	0.2071005E 00	0.1882225E 00	0.1772634E 00
0.1588416E 00	0.1505039E 00	0.1338879E 00	0.1146557E 00
0.1022107E 00	0.9151363E-01	0.7756662E-01	0.6906772E-01
0.5646708E-01	0.4487861E-01	0.3331536E-01	0.2657641E-01
0.1820700E-01	0.9477235E-02	0.5660690E-02	0.2936016E-02
0.1072021E-02	0.5649622E-03	0.1822288E-03	0.1219036E-03
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.1736314E-04	0.1416877E-03	0.1094503E-02	0.4919004E-02
0.1849992E-01	0.4963557E-01	0.8561850E-01	0.1313989E 00
0.1824397E 00	0.2255227E 00	0.2466831E 00	0.2591423E 00
0.2590284E 00	0.2530327E 00	0.2368238E 00	0.2219617E 00
0.2101821E 00	0.1905518E 00	0.1772934E 00	0.1539934E 00
0.1531928E 00	0.1347782E 00	0.1212198E 00	0.1001142E 00
0.9136510E-01	0.7847703E-01	0.6849623E-01	0.5679385E-01
0.4542725E-01	0.3310285E-01	0.2717116E-01	0.1823463E-01
0.1100488E-01	0.6262664E-02	0.3032315E-02	0.1214716E-02
0.4865748E-03	0.1983496E-03	0.1126833E-03	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.6215397E-05	0.1900049E-04
0.2125569E-03	0.1044521E-02	0.5343676E-02	0.2001397E-01
0.5058838E-01	0.1000955E 00	0.1386551E 00	0.1750458E 00
0.2194927E 00	0.2489910E 00	0.2569161E 00	0.2581220E 00
0.2506329E 00	0.2372339E 00	0.2237182E 00	0.2063728E 00
0.1900717E 00	0.1772207E 00	0.1513262E 00	0.1474659E 00
0.1340637E 00	0.1194983E 00	0.1062744E 00	0.9037650E-01
0.7943255E-01	0.7025135E-01	0.5633250E-01	0.4437540E-01
0.3281593E-01	0.2634889E-01	0.1717417E-01	0.1114297E-01
0.6952133E-02	0.2916923E-02	0.1267667E-02	0.4754490E-03
0.1937536E-03	0.1845764E-03	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.2060735E-04	0.2163455E-03
0.1342522E-02	0.6368209E-02	0.2268060E-01	0.5327580E-01
0.9917772E-01	0.1467553E 00	0.1820194E 00	0.2214158E 00
0.2402436E 00	0.2599525E 00	0.2661082E 00	0.2506132E 00
0.2415536E 00	0.2177314E 00	0.2094588E 00	0.1981256E 00
0.1589200E 00	0.1608635E 00	0.1391610E 00	0.1290765E 00
0.1205251E 00	0.9791631E-01	0.9413302E-01	0.7565463E-01
0.6731683E-01	0.5816970E-01	0.3945719E-01	0.3621977E-01
0.2382486E-01	0.1810371E-01	0.1119237E-01	0.5598128E-02
0.3102446E-02	0.1532112E-02	0.4638946E-03	0.2444400E-03
0.1360976E-03	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.2193426E-04	0.2478703E-03	0.1424491E-02

0.6939832E-02	0.2330431E-01	0.5433539E-01	0.1026180E 00
0.1558304E 00	0.2129785E 00	0.2227834E 00	0.2363340E 00
0.2588930E 00	0.2715958E 00	0.2593470E 00	0.2426100E 00
0.2242088E 00	0.2106870E 00	0.1886187E 00	0.1663788E 00
0.1477292E 00	0.1343106E 00	0.1218542E 00	0.1097493E 00
0.9979910E-01	0.8680904E-01	0.7769436E-01	0.6276107E-01
0.5106154E-01	0.3982237E-01	0.3015380E-01	0.2221074E-01
0.1571323E-01	0.9520765E-02	0.5431309E-02	0.2716314E-02
0.1153191E-02	0.5903142E-03	0.3588586E-03	0.7260863E-04
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.2246628E-04	0.2236100E-03	0.1733763E-02	0.7913966E-02
0.2478982E-01	0.5738202E-01	0.1026796E 00	0.1519760E 00
0.1982002E 00	0.2451878E 00	0.2428117E 00	0.2483238E 00
0.2688071E 00	0.2689522E 00	0.2546778E 00	0.2319517E 00
0.2151463E 00	0.1896352E 00	0.1741874E 00	0.1440163E 00
0.1363129E 00	0.1170444E 00	0.1052563E 00	0.9134817E-01
0.8146679E-01	0.7008237E-01	0.6129551E-01	0.4842377E-01
0.3848107E-01	0.2644810E-01	0.2167465E-01	0.1374268E-01
0.8248139E-02	0.4956193E-02	0.3049610E-02	0.1419656E-02
0.6481190E-03	0.1044638E-03	0.4576219E-04	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.2671527E-03	0.1822941E-02	0.9021621E-02	0.2665320E-01
0.5989974E-01	0.1023681E 00	0.1461299E 00	0.1856654E 00
0.2234071E 00	0.2657835E 00	0.2548606E 00	0.2599273E 00
0.2702767E 00	0.2659580E 00	0.2369433E 00	0.2231550E 00
0.1962218E 00	0.1779684E 00	0.1485134E 00	0.1321834E 00
0.1154602E 00	0.1054579E 00	0.9060305E-01	0.7756007E-01
0.6784779E-01	0.5828715E-01	0.4759468E-01	0.3566820E-01
0.2557244E-01	0.1899502E-01	0.1280696E-01	0.7447124E-02
0.4304644E-02	0.2219763E-02	0.1180247E-02	0.3952421E-03
0.1624286E-03	0.7658264E-04	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.2269397E-04	0.2308429E-03
0.2075878E-02	0.9624846E-02	0.3191707E-01	0.6668711E-01
0.1050972E 00	0.1461197E 00	0.1784285E 00	0.2115127E 00
0.2481316E 00	0.2835237E 00	0.2701750E 00	0.2609929E 00
0.2643824E 00	0.2523072E 00	0.2293909E 00	0.2022312E 00
0.1802447E 00	0.1499075E 00	0.1345164E 00	0.1111640E 00
0.9712094E-01	0.8830923E-01	0.7212412E-01	0.6215255E-01
0.5621058E-01	0.4325967E-01	0.3344726E-01	0.2382999E-01
0.1859563E-01	0.1131049E-01	0.6426115E-02	0.3995836E-02
0.1880914E-02	0.8638178E-03	0.3228260E-03	0.1022436E-03
0.3061809E-04	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00

```
    0.0000000E 00    0.3218444E-04    0.2596709E-03    0.2107170E-02
    0.9936295E-02    0.3130761E-01    0.6618536E-01    0.1071982E 00
    0.1471151E 00    0.1840913E 00    0.2161806E 00    0.2493536E 00
    0.2670345E 00    0.2680613E 00    0.2930642E 00    0.2637855E 00
    0.2573879E 00    0.2264757E 00    0.2073898E 00    0.1796527E 00
    0.1536861E 00    0.1332374E 00    0.1119388E 00    0.9786659E-01
    0.8207232E-01    0.7238299E-01    0.5762001E-01    0.4698988E-01
    0.3794770E-01    0.3120529E-01    0.2001548E-01    0.1510441E-01
    0.1004597E-01    0.6521642E-02    0.3233414E-02    0.1113423E-02
    0.6435823E-03    0.2916171E-03    0.5741679E-04    0.7639593E-04
    0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
    0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
    0.0000000E 00    0.0000000E 00    0.0000000E 00    0.8601016E-05
    0.3439712E-04    0.3028943E-03    0.2249576E-02    0.1088468E-01
    0.3506811E-01    0.7260209E-01    0.1124092E 00    0.1431631E 00
    0.1702980E 00    0.1968122E 00    0.2334363E 00    0.2575313E 00
    0.2904121E 00    0.3154174E 00    0.2771440E 00    0.2493343E 00
    0.2392632E 00    0.2168028E 00    0.1860240E 00    0.1538441E 00
    0.1345306E 00    0.1114675E 00    0.9396076E-01    0.7578874E-01
    0.6721002E-01    0.5436332E-01    0.4397735E-01    0.3473518E-01
    0.2713976E-01    0.1855069E-01    0.1498656E-01    0.8664083E-02
    0.4691057E-02    0.2833721E-02    0.1506527E-02    0.6247566E-03
    0.2911384E-03    0.1415498E-03    0.6554931E-04
&VALUES
LG=.TRUE.,
LFBAR=.FALSE.,
LFBARS=.FALSE.,
&END
&VALUS1
NT=10,
SAGE=13,FAGE=45,
&END
```

```
&VALUES LG = .TRUE.. LEBAR = .FALSE.. LEBARS = .FALSE..&END
&VALUS1 NT =  10. CAGE =  13. EAGE =  45.&END
```

G(X,T)			Y E A R		PAGE 1
AGE	1956	1957	1958	1959	1960
12	0.00000	0.00000	0.00000	0.00000	0.00000
13	0.43654	0.35973	0.05159	-0.07967	0.30985
14	-0.03086	-0.01209	-0.01101	0.01835	0.02156
15	0.12942	-0.06085	-0.06323	0.04081	0.03170
16	-0.20649	0.00402	-0.02999	0.00979	0.02433
17	-0.23983	0.00921	-0.00267	-0.00873	0.01156
18	-0.25516	0.01973	-0.00719	-0.00865	0.00644
19	-0.30333	0.00444	-0.00695	-0.01774	0.00530
20	-0.23356	-0.04463	-0.01038	-0.02060	-0.01316
21	-0.22797	-0.04551	-0.02838	-0.03699	-0.02106
22	-0.25610	-0.05052	-0.03808	-0.05538	-0.03500
23	-0.22197	-0.06570	-0.05519	-0.05019	-0.05917
24	-0.22393	-0.03874	-0.07619	-0.06164	-0.05160
25	-0.24922	-0.01796	-0.06643	-0.08163	-0.05547
26	-0.19335	-0.04277	-0.05098	-0.06705	-0.07226
27	-0.22466	-0.05820	-0.05517	-0.05784	-0.05456
28	-0.23277	-0.07191	-0.06793	-0.06711	-0.03271
29	-0.20510	-0.08785	-0.10310	-0.08213	-0.02359
30	-0.15193	-0.10437	-0.11578	-0.11874	-0.03538
31	-0.07585	-0.13312	-0.13536	-0.13552	-0.07923

AGE	1956	1957	1958	1959	1960
32	-0.06945	-0.12044	-0.13571	-0.12392	-0.09665
33	-0.02003	-0.13881	-0.11841	-0.11574	-0.10554
34	0.06072	-0.15555	-0.12255	-0.09939	-0.09221
35	0.13677	-0.14448	-0.12221	-0.10413	-0.06144
36	0.17673	-0.13009	-0.08042	-0.09306	-0.06494
37	0.07373	-0.12426	-0.04089	-0.04491	-0.05895
38	0.08254	-0.18149	-0.04381	0.00080	-0.02189
39	0.19057	-0.19554	-0.10019	-0.00424	0.01775
40	0.15136	-0.17617	-0.08770	-0.07232	0.01033
41	0.27887	-0.22076	-0.07080	-0.09064	-0.06810
42	0.10958	-0.11569	-0.08114	-0.06591	-0.05818
43	0.02486	-0.08957	-0.02722	-0.10174	-0.03240
44	-0.12714	-0.01054	-0.04097	-0.09598	-0.12165

G(X,T)			YEAR		PAGE 3
AGE	1961	1962	1963	1964	1965
12	0.00000	0.00000	0.00000	0.00000	0.00000
13	-0.63561	0.07112	0.68041	-1.47289	2.24870
14	-0.03794	0.10174	0.06027	-0.23021	0.43328
15	-0.04220	0.04999	0.03666	-0.08787	0.15576
16	0.00974	-0.02910	0.00187	0.09905	-0.37072
17	0.05761	-0.01050	-0.06329	0.03601	-0.19406
18	0.05108	0.04285	-0.04386	-0.05446	-0.18166
19	0.03958	0.04528	0.01395	-0.04434	-0.33803
20	0.03090	0.03442	0.03754	0.01892	-0.35239
21	0.01261	0.03087	0.04849	0.02961	-0.20367
22	-0.00048	0.02230	0.05111	0.03856	-0.18546
23	-0.02149	0.01970	0.04703	0.06052	-0.23574
24	-0.05083	0.00267	0.05398	0.06018	-0.20893
25	-0.04878	-0.01264	0.04439	0.05409	-0.18735
26	-0.05674	-0.01085	0.03529	0.03876	-0.14910
27	-0.07049	-0.03129	0.04268	0.03925	-0.16925
28	-0.05668	-0.04863	0.03822	0.04247	-0.17272
29	-0.04350	-0.03961	0.03450	0.03061	-0.12268
30	-0.01744	-0.02497	0.03493	0.02156	-0.11257
31	-0.02582	-0.01177	0.04947	0.05016	-0.16236

G(X,T) — Y E A R

AGE	1961	1962	1963	1964	1965
32	-0.08438	-0.01779	0.06708	0.05506	-0.08172
33	-0.10427	-0.05618	0.05316	0.06138	-0.03360
34	-0.10766	-0.07738	0.02382	0.04618	0.02439
35	-0.09603	-0.10450	0.01644	0.01822	0.03419
36	-0.06795	-0.12095	0.00662	0.05682	-0.13432
37	-0.08179	-0.11678	0.00396	0.05585	-0.11118
38	-0.06425	-0.11601	-0.00144	0.06300	-0.19685
39	-0.01951	-0.12154	-0.01672	0.09415	-0.32638
40	0.03522	-0.10348	-0.00895	0.09352	-0.30282
41	0.03894	-0.05998	0.02280	0.10308	-0.29320
42	-0.05974	-0.03675	0.08008	0.07966	-0.19298
43	-0.08975	-0.10306	0.14024	0.07047	-0.00761
44	-0.08814	-0.09408	0.06629	0.24844	-0.25464

If $M(x,t)$ were identical to $m(x,t)$ for a discrete set of x and t, then the vectors $\underline{a}_r$ (and thence d_r) could be obtained as the orthogonal eigenvectors of $M^T M$.

However, discrepancies in the data used, and use of the algorithms for estimating m from data, lead to some differences between $m(x,t)$ and M particularly at or near the boundaries. (Evidence of this is given in estimating an artificial economic mobility in the Interim Report.) If these errors were small enough, the dominant modes could still be segregated from the noise: in the presence of noise, $M^T M$ will have some eigenvalues distinct from zero, and some close to zero (the latter corresponding to the errors in M).

As described in detail in Chapter II above, the following procedure was adopted:

1. to form $M^T M$, only a subset of the values of M in the age distribution were used;
2. the two outside borders of $M^T M$ were replaced by zeros.

These modifications resulted from considerable experimentation directed at obtaining consistent results over eight successive ten year periods (see below for further description).

The first modification was an attempt to remove anomolous results occurring when $\overline{f}(x,t)$ was small; the second modification minimized the effect of inaccuracies in the boundary values of splines in the time variable.

To determine the extent to which consistent estimates could be obtained from the data, eight submatrices of the matrix M were defined, each corresponding to entries for ten successive years in time: e.g. the submatrix M_2 contains the entries of M for ages SAGE $\leq$ x $\leq$ EAGE, and time $1957 \leq t \leq 1966$.

For each submatrix M_r , the first four dominant eigenvectors $\underline{a}_r$ of $M_i^T M_i$ were obtained. Corresponding vectors $\underline{d}_r$ were determined as

$$\underline{d}_r = \frac{M_i \underline{a}_r}{\underline{a}_r^T \underline{a}_r} \quad . \tag{9}$$

Each pair (a_r, d_r) obtained in this manner is only unique up to multiplication and division respectively by an arbitrary constant. In order to obtain consistent scaling for the values of $a_r(t)$ obtained from different time intervals, d_r was scaled so that $\sum_{i=SAGE+5}^{EAGE-5} d_r^2(i) = 2.0$, further, the orientation of contiguous vectors d_r is the same.

While the selection of the ages SAGE and EAGE is optional, SAGE = 12 EAGE = 44 have been chosen here, and program 2.3 estimates the pairs $(\underline{a}_r, \underline{d}_r)$, using M_i for ages [12,44] and setting two borders of $M_i^T M_i$ to zero. The digit given in the summary plots for $\underline{a}_r$ corresponds to suffix of M_i (for duplicate values, lower values of i are suppressed) while the digit 9 represents the average value taken from each of the eight cases. The second set of graphs represents the four corresponding vectors $\underline{d}_r$. (The subroutine PLOTT is listed as SUBROUTINE 3.7.)

```
$JOB    ACCT-NUM,VERNER,TIME=60
C       ************************************************************************
C
C               PROGRAM 2.3 :  POP2
C
C       ************************************************************************
```

```
3JOB POP2; CHARGE BERGHOUT;CLASS 3;
USER 080002/VHJ;
PROCESSTIME=300;
PRINTLIMIT=13200;
BEGIN
COMPILE POP2 FORTRAN;
FILE FILE3(KIND=PACK,TITLE=DATA/XXG,FILETYPE=7);
FILE FILE4(KIND=PACK,TITLE=DATA/XXAV,FLEXIBLE,MAXRECSIZE=90,
BLOCKSIZE=720,AREASIZE=8,AREAS=1,SAVEFACTOR=99);
OPTION=AUTORM;
DATA
$RESET LIST $
$INCLUDE 'QLIB/FS/PLOTT'
$SET LIST




3DATA FILE5




3END JOB
```

```
$RESET LIST $
$INCLUDE 'OLIB/ES/PLOTT'
$SET LIST
C      ********************************************************************************
C
C      PROGRAM : 2.3
C      TITLE : POP2
C      PURPOSE : CALCULATE DOMINANT COMPONENTS OF FERTILITY
C      DATA : DATA/XXG
C      OUTPUT : EIGENVECTORS AI,D-VECTORS DI
C               AVERAGES -> DATA/XXAV,PLOTS OF THE VECTORS
C
C      ********************************************************************************
C
C
C      HOW TO SET UP A RUN
C      === == === = ===
C
C      PREPARE NAMELISTS CALLED VALS2,VALS3 AND VALS4
C      SPECIFYING THE FOLLOWING PARAMETERS.
C
C     &VALS2
C      PEIGEN  = .TRUE. => PLOT EIGENVECTORS FOR EACH TIME PERIOD
C      PD      = .TRUE => PLOT D VECTORS FOR EACH TIME PERIOD
C      PEIGES  = .TRUE. => PLOT SUMMARY EIGEN VECTORS
C      PDS     = .TRUE. => PLOT SUMMARY D VECTORS
C      &END
C     &VALS3
C      NT      = # OF YEARS PER TIME PERIOD
C      SAGE    = STARTING AGE
C      EAGE    = ENDING AGE
C      SYEAR   = STARTING YEAR
C      EYEAR   = ENDING YEAR
C      &END
C     &VALS4
C      NEV     = # OF EIGEN VECTORS TO BE CONSIDERED (MAXIMUM = NT)
C      ICL     = # OF LEADING COLUMNS OF G TO BE ZEROED
C      ICT     = # OF TRAILING COLUMNS OF G TO BE ZEROED
C      IRL     = # OF LEADING ROWS OF G TO BE ZEROED
C      IRT     = # OF TRAILING ROWS OF G TO BE ZEROED
C      &END
C
C
C      SUMMARY OF VARIABLES USED
C      ======= == ========= ====
C
C      G(33,10):  READ FROM A DISK FILE.  CONTAINS 'OBSERVATIONS' OF THE
C      FORM INTEGRAL FROM 0 TO X OF -D/DT(FBAR(X,T)) ALL DIVIDED BY
C      FBAR(X,T).  8 OF THESE ARRAYS ARE READ.
C      THESE ARRAYS HAVE BEEN CALCULATED BY THE PROGRAM POP1.
C      EVT: AFTER THE SUBROUTINE EIGEN IS CALLED, EVT WILL CONTAIN
C      EIGEN VECTORS STORED COLUMNWISE.
C      MEIG:  THE EIGENVALUES OF M ARE STORED IN THE DIAGONAL OF MEIG
C      IN DESCENDING ORDER.
C      A:  USED TO REARRANGE THE ELEMENTS OF M BEFORE CALLING EIGEN.
C      M:  USED IN CALCM.  WILL CONTAIN G TRANSPOSE.G
C      D:  COLUMNS OF D ARE USED TO STORE THE D-VECTORS CALCULATED
C      IN CALCD.
C      PLOTD:  USED TO STORE ALL D-VECTORS CALCULATED OVER 8 TIME
```

```
C        PERIODS.
C        PLOTE:  USED TO STORE ALL EIGEN VECTORS OBTAINED OVER THE 8 TIME
C        PERIODS.
C        EVALM:  EIGENVALUES FOR EACH TIME PERIOD ARE STORED IN THE
C        COLUMNS OF EVALM.
C        D1(33,10):  USED IN SUBROUTINE CALCD TO 'NORMALIZE' THE D-VECTORS.
C
C
         REAL G(33,10)
         REAL EVT(10,10),D(33,10),PLOTD(33,10,8),PLOTE(10,10,8)
         REAL MEIG(10,10),A(55),M(10,10)
         REAL EVALM(10,8)
         REAL D1(33,10)
         LOGICAL PEIGEN,PD,PEIGES,PDS
         INTEGER NT,SAGE,EAGE,NX,SYEAR,EYEAR,NEV
         NAMELIST/VALS2/PEIGEN,PD,PEIGES,PDS
         NAMELIST/VALS3/NT,SAGE,EAGE,SYEAR,EYEAR
         NAMELIST/VALS4/NEV,ICL,ICT,IRL,IRT
C
C        READ PARAMETERS TO BE USED IN THIS RUN.
C
         READ(5,VALS2)
         READ(5,VALS3)
         READ(5,VALS4)
         WRITE(6,100)
         WRITE(6,VALS2)
         WRITE(6,VALS3)
         WRITE(6,VALS4)
     100 FORMAT('0','PARAMETERS USED IN THIS RUN',///)
         NX = EAGE - SAGE + 1
C
C        INITIALIZE D1 MATRIX
C
         LL = SAGE + 1
         DO 20 J = 1,NX
         VAL = 0.0
         IF (J.GE.(35-LL).AND.J.LE.(38-LL)) VAL = 1.0
         DO 20 K = 1,NT
      20 D1(J,K) = VAL
         DO 1 I = 1,8
C
C        THE G MATRIX FOR EACH TIME PERIOD IS READ FROM THE DISK FILE
C        DATA/XXG.
C
         DO 2 J = 1,NX
       2 READ(3)(G(J,K),K=1,NT)
         CALL ZEROG(G,NX,NT,ICL,ICT,IRL,IRT)
         CALL CALCM(G,M,NX,NT)
         DO 3 K = 1,NT
         DO 3 J = K,NT
         IA = K + (J * J - J) / 2
       3    A(IA) = M(K,J)
         N1 = (NT*NT+NT)/2
         N2 = NT*NT
         CALL EIGEN(A,EVT,NT,0,N1,N2)
         DO 4 K = 1,NT
         DO 4 J = K,NT
         IA = K + (J * J - J) / 2
       4    MEIG(K,J) = A(IA)
         CALL CALCD(G,EVT,D,D1,NX,NT,NEV)
```

```
C
C      PRINT EIGEN VALUES, EIGEN VECTORS AND D
C
       IYEAR = SYEAR + I - 1
       DO 5 K = 1,NEV
       IF (MOD(K,2).EQ.1) WRITE(6,150) IYEAR
       WRITE(6,200) MEIG(K,K)
       WRITE(6,210) NT,NT
       WRITE(6,220)(EVT(J,K),J=1,NT)
       WRITE(6,230)
       WRITE(6,220)(D(J,K), J=1,NX)
   150 FORMAT('1','EIGEN VALUES,EIGEN VECTORS AND D MATRIX FOR TIME PERIO
      $D STARTING',//' AT THE YEAR 19',I2)
   200 FORMAT(///,10X,'EIGEN VALUE = ',F10.5)
   210 FORMAT('0',' A(T) OBTAINED FROM ',I2,' BY ',I2,' MATRIX IS')
   220 FORMAT('0',5F13.6)
   230 FORMAT('0',' D(X) OBTAINED AS (G(X,T).A(T))/A(T).A(T))' )
C
C      STORE EIGENVALUES FOR SUMMARY
C
       EVALM(K,I) = MEIG(K,K)
C
C      STORE THE EIGEN VECTORS AND D-VECTORS ONLY IF SUMMARY PLOTS
C      ARE REQUESTED.
C
       IF (.NOT.PEIGES) GO TO 9
       DO 7 J = 1,NT
     7    PLOTE(J,K,I) = EVT(J,K)
     9 IF (.NOT.PDS) GO TO 5
       DO 8 J = 1,NX
     8    PLOTD(J,K,I) = D(J,K)
     5 CONTINUE
C
C      PLOT EIGEN VECTORS IF REQUESTED.
C
       NEV1 = NEV+1
       IF (.NOT.PEIGEN) GO TO 12
       NN = NT - 1
       LL = NT
       DO 10 J = 1,NN
       LL = LL - 1
       DO 10 K = 1,NT
    10    EVT(K,LL+1) = EVT(K,LL)
       DO 11 J = 1,NT
    11     EVT(J,1) = SYEAR + I + J - 2
       CALL PLOTA(1,EVT,NT,NEV1,0.0,0.0,1.0,-1.0)
C
C      PLOT D-VECTORS IF REQUESTED.
C
    12 IF (.NOT.PD) GO TO 15
       NN = NT - 1
       LL = NT
       DO 13 J = 1,NN
       LL = LL - 1
       DO 13 K = 1,NX
    13    D(K,LL+1) = D(K,LL)
       DO 14 K = 1,NX
    14    D(K,1) = SAGE + K - 1
       CALL PLOTA(2,D,NX,NEV1,0.0,0.0,0.5,-0.5)
    15 CONTINUE
```

```
      1 CONTINUE
C
C        PRINT SUMMARY OF EIGEN VALUES
C
        WRITE(6,240)
        DO 19 J=1,8
     19 WRITE(6,260) (EVALM(K,J),K=1,NEV)
    240 FORMAT('1',28X,'SUMMARY OF EIGEN VALUES',////)
    260 FORMAT('0',4F10.5)
C
C        PLOT SUMMARY EIGEN VECTORS IF REQUESTED.
C
        IF (.NOT.PEIGES) GO TO 16
        CALL PLTER(PLOTE,NT,NT,SYEAR,EYEAR,NEV,0)
C
C        PLOT SUMMARY D-VECTORS IF REQUESTED.
C
     16 IF (.NOT.PDS) GO TO 17
        CALL PLTER(PLOTD,NX,NT,SAGE,EAGE,NEV,1)
     17 CONTINUE
        LOCK 4

        STOP
        END

        SUBROUTINE PLTER(ARRAY,NX,NT,START,FINISH,NEV,CODE)
C
C        THE PURPOSE OF THIS ROUTINE IS TO PLOT THE SUMMARY VECTORS
C        WHICH HAVE BEEN STORED IN PLOTE OR PLOTD.
C        EACH PLOT WILL CONTAIN 9 VECTORS, ONE WHICH IS THE AVERAGE AND
C        ONE FROM EACH OF THE TIME PERIODS.
C        NEV      = # OF PLOTS
C        START    = FIRST VALUE OF BASE VECTOR
C        FINISH   = LAST VALUE OF BASE VECTOR
C        CODE     = 0 => PLOT EIGEN VECTORS
C        CODE     = 1 => PLOT D VECTORS
C        ARRAY(NX,NT,8) CONTAINS VECTORS TO BE PLOTTED
C
        INTEGER START,FINISH,CODE
        REAL ARRAY(NX,NT,8)
        REAL PLT(33,10)
        REAL MEANS(33,10)
        XMAX=FINISH
        XMIN=START
C
C        TO SUPPRESS THE PLOTTING OF ZEROES, INITIALIZE PLT TO 99.9.
C        (THE VALUE 99.9 LIES OUTSIDE THE RANGE OF VALUES WHICH WILL
C        BE PLOTTED.)
C
        DO 5 I = 1,33
        DO 5 J = 1,NT
      5   PLT(I,J) = 99.9
        DO 1 I = 1,NEV
C
C        MOVE THE ITH EIGEN OR D VECTOR FROM EACH OF THE 8 TIME PERIODS
C        TO THE COLUMNS OF PLT
C
        IF (CODE.EQ.1) GO TO 7
        DO 2 K = 1,8
        DO 2 J = 1,NX
```

```
            PLT(J+K-1,K+1) = ARRAY(J,I,K)
    2       IF (ARRAY(J,I,K).EQ.0.0) PLT(J+K-1,K+1) = 99.9
         II = NX+7
         GO TO 9
    7 DO 8 K = 1,8
         DO 8 J = 1,NX
            PLT(J,K+1) = ARRAY(J,I,K)
    8       IF (ARRAY(J,I,K).EQ.0.0) PLT(J,K+1) = 99.9
         II = NX
C
C        CALCULATE THE MEAN OF THE 8 VECTORS AND STORE THE RESULT IN
C        COLUMN 10 OF PLT.  THE BASE VECTOR FOR PLOTTING IS STORED IN
C        COLUMN 1 OF PLT.
C
    9 DO 3 J = 1,II
         SUM = 0.0
         NUM = 0
         LL = NT-1
         DO 4 K = 2,LL
            IF (PLT(J,K).EQ.99.9) GO TO 4
            SUM = SUM + PLT(J,K)
            NUM = NUM+1
    4 CONTINUE
         IF (NUM.NE.0) SUM = SUM/NUM
         MEANS(J,I) = SUM
         IF (NUM.EQ.0) SUM = 99.9
         PLT(J,NT) = SUM
         PLT(J,1) = START + J - 1
    3 CONTINUE
C
C        CALCULATE THE MAXIMUM AND MINIMUM Y VALUES TO BE PLOTTED AND
C        THEN CALL PLOTA.
C
         YMIN = 10**10
         YMAX = -YMIN
         DO 6 K = 2,NT
         DO 6 J = 1,II
            IF (PLT(J,K).EQ.99.9) GO TO 6
            IF (PLT(J,K).LT.YMIN) YMIN = PLT(J,K)
            IF (PLT(J,K).GT.YMAX) YMAX = PLT(J,K)
    6 CONTINUE
         LL = CODE * 100 + 100 + I
         CALL PLOTA(LL,PLT,33,10,XMAX,XMIN,YMAX,YMIN)
    1 CONTINUE
C
C        PRINT SUMMARY OF AVERAGES STORED IN MEANS(33,10)
C
         IF (CODE.EQ.0) WRITE(6,100)
         IF (CODE.EQ.1) WRITE(6,200)
         LL = II - 2
         DO 10 I = 1,NEV
         WRITE(6,300)(MEANS(J,I),J=1,II)
         WRITE(4)(MEANS(J,I),J=3,LL)
   10 WRITE(6,400)
  100 FORMAT('1',22X,'AVERAGES OF EIGEN VECTORS',//)
  200 FORMAT('1',23X,'AVERAGES OF D-VECTORS',//)
  300 FORMAT(/,6F10.5)
  400 FORMAT('0',1X)
         RETURN
         END
```

```
      SUBROUTINE CALCD(G,EVT,D,D1,NX,NT,NEV)
C
C
C     THIS SUBROUTINE CALCULATES THE NUMBER OF D-VECTORS SPECIFIED BY
C      NEV FOR ONE TIME PERIOD.
C     EACH D IS CALCULATED AS:
C     D(I) = SUM OVER T OF A(I).G(XI,T)/A(I).A(I)
C     WHERE A(I) IS THE ITH EIGEN VECTOR.  THE D-VECTORS ARE STORED
C     IN THE MATRIX D(NX,NT)
C
C
      REAL EVT(NT,NT),D(NX,NT)
      REAL D1(NX,NT)
      REAL G(NX,NT)
      DO 1 IV = 1,NEV
      SUM2 = 0.0
      SUM3 = 0.0
      DO 4 IX = 1,NX
      SUM = 0.0
      SUM1 = 0.0
         DO 2 IT = 1,NT
         SUM = SUM + EVT(IT,IV) * G(IX,IT)
         SUM1 = SUM1 + (EVT(IT,IV))**2
    2    CONTINUE
      D(IX,IV) = SUM/SUM1
C
C     'NORMALIZE' THE D VECTORS
C
      IF (IX.LT.6.OR.IX.GT.(NX-6)) GO TO 4
      SUM2 = SUM2 + (D(IX,IV))**2
      SUM3 = SUM3 + D(IX,IV) * D1(IX,IV)
    4 CONTINUE
      SUM = SUM2 / 2.0
      SUM = SQRT(SUM) * SGN(SUM3)
      SUM2 = SUM3/SUM2
      IF (D1(NX-5,NEV).EQ.0.0) SUM2 = 1.0/SUM
      DO 3 IT = 1,NT
    3 EVT(IT,IV) = EVT(IT,IV) / SUM2
      DO 5 IX = 1,NX
      D(IX,IV) = D(IX,IV) * SUM2
      IF ((IX.GT.5.AND.IX.LT.(NX-5)).AND.D1(NX-5,NEV).EQ.0.0)
     $  D1(IX,IV) = D(IX,IV)
    5 CONTINUE
    1 CONTINUE
      RETURN
      END

      SUBROUTINE ZEROG(G,NX,NT,ICL,ICT,IRL,IRT)
C
C
C     THIS SUBROUTINE ZEROES OUT ROWS AND COLUMNS OF G(NX,NT) AS
C     SPECIFIED BY ICL,ICT,IRL AND IRT.
C
C
      REAL G(NX,NT)
      DO 1 IX = 1,NX
      IF (ICL.EQ.0) GO TO 3
      DO 2 I = 1,ICL
    2    G(IX,I) = 0.0
    3 IF (ICT.EQ.0) GO TO 1
```

```
      LL = NT + 1
      DO 4 I = 1,ICT
         LL = LL - 1
    4    G(IX,LL) = 0.0
    1 CONTINUE
      DO 5 IT = 1,NT
      IF (IRL.EQ.0) GO TO 7
      DO 6 I = 1,IRL
    6    G(I,IT) = 0.0
    7 IF (IRT.EQ.0) GO TO 5
      LL = NX + 1
      DO 8 I = 1,IRT
         LL = LL - 1
    8    G(LL,IT) = 0.0
    5 CONTINUE
      RETURN
      END

      REAL FUNCTION SGN(A)
C
C     RETURNS +1 OR -1 AS THE SIGN OF A.  THE SIGN OF 0 = +1.
C
      IF (A.EQ.0.0) GO TO 1
      SGN = A / ABS(A)
      RETURN
    1 SGN = 1.0
      RETURN
      END

      SUBROUTINE CALCM(G,M,NX,NT)
C
C
C     THIS SUBROUTINE CALCULATES THE MATRIX M AS:
C     M(NT,NT) = G(NX,NT)TRANSPOSE.G(NX,NT)
C
C
      REAL G(NX,NT),M(NT,NT)
      DO 1 I = 1,NT
      DO 1 J = 1,NT
         SUM = 0.0
         DO 2 K = 1,NX
    2    SUM = SUM + G(K,I) * G(K,J)
      M(J,I) = SUM
    1 CONTINUE
      RETURN
      END
C
C     ..................................................................
C
C        SUBROUTINE EIGEN
C
C        PURPOSE
C           COMPUTE EIGENVALUES AND EIGENVECTORS OF A REAL SYMMETRIC
C           MATRIX
C
C        USAGE
C           CALL EIGEN(A,R,N,MV,N1,N2)
C
C        DESCRIPTION OF PARAMETERS
C           A - ORIGINAL MATRIX (SYMMETRIC), DESTROYED IN COMPUTATION.
```

```
C                 RESULTANT EIGENVALUES ARE DEVELOPED IN DIAGONAL OF
C                 MATRIX A IN DESCENDING ORDER.
C              R - RESULTANT MATRIX OF EIGENVECTORS (STORED COLUMNWISE,
C                 IN SAME SEQUENCE AS EIGENVALUES)
C              N - ORDER OF MATRICES A AND R
C              MV- INPUT CODE
C                      0   COMPUTE EIGENVALUES AND EIGENVECTORS
C                      1   COMPUTE EIGENVALUES ONLY (R NEED NOT BE
C                          DIMENSIONED BUT MUST STILL APPEAR IN CALLING
C                          SEQUENCE)
C              N1:  DIMENSION OF A.  N1 = (N*N+N)/2
C              N2:  # OF ELEMENTS IN R.  N2 = N*N.
C
C           REMARKS
C              ORIGINAL MATRIX A MUST BE REAL SYMMETRIC (STORAGE MODE=1)
C              MATRIX A CANNOT BE IN THE SAME LOCATION AS MATRIX R
C
C           SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C              NONE
C
C           METHOD
C              DIAGONALIZATION METHOD ORIGINATED BY JACOBI AND ADAPTED
C              BY VON NEUMANN FOR LARGE COMPUTERS AS FOUND IN 'MATHEMATICAL
C              METHODS FOR DIGITAL COMPUTERS', EDITED BY A. RALSTON AND
C              H.S. WILF, JOHN WILEY AND SONS, NEW YORK, 1962, CHAPTER 7
C
C        ..................................................................
C

      SUBROUTINE EIGEN(A,R,N,MV,N1,N2)
      DIMENSION A(N1),R(N2)
C
C           ..............................................................
C
C           IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C           C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
C           STATEMENT WHICH FOLLOWS.
C
C     DOUBLE PRECISION A,R,ANORM,ANRMX,THR,X,Y,SINX,SINX2,COSX,
C    1                 COSX2,SINCS
C
C           THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C           APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C           ROUTINE.
C
C           THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
C           CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  SQRT IN STATEMENTS
C           40, 68, 75, AND 78 MUST BE CHANGED TO DSQRT.  ABS IN STATEMENT
C           62 MUST BE CHANGED TO DABS.
C
C           ..............................................................
C
C           GENERATE IDENTITY MATRIX
C
      IF(MV-1) 10,25,10
   10 IQ=-N
      DO 20 J=1,N
      IQ=IQ+N
      DO 20 I=1,N
      IJ=IQ+I
```

```
      R(IJ)=0.0
      IF(I-J) 20,15,20
   15 R(IJ)=1.0
   20 CONTINUE
C
C        COMPUTE INITIAL AND FINAL NORMS (ANORM AND ANORMX)
C
   25 ANORM=0.0
      DO 35 I=1,N
      DO 35 J=I,N
      IF(I-J) 30,35,30
   30 IA=I+(J*J-J)/2
      ANORM=ANORM+A(IA)*A(IA)
   35 CONTINUE
      IF(ANORM) 165,165,40
   40 ANORM=1.414* SQRT(ANORM)
      ANRMX=ANORM*1.0E-10/FLOAT(N)
C
C        INITIALIZE INDICATORS AND COMPUTE THRESHOLD, THR
C
      IND=0
      THR=ANORM
   45 THR=THR/FLOAT(N)
   50 L=1
   55 M=L+1
C
C        COMPUTE SIN AND COS
C
   60 MQ=(M*M-M)/2
      LQ=(L*L-L)/2
      LM=L+MQ
   62 IF( ABS(A(LM))-THR) 130,65,65
   65 IND=1
      LL=L+LQ
      MM=M+MQ
      X=0.5*(A(LL)-A(MM))
   68 Y=-A(LM)/ SQRT(A(LM)*A(LM)+X*X)
      IF(X) 70,75,75
   70 Y=-Y
   75 SINX=Y/ SQRT(2.0*(1.0+( SQRT(1.0-Y*Y))))
      SINX2=SINX*SINX
   78 COSX= SQRT(1.0-SINX2)
      COSX2=COSX*COSX
      SINCS =SINX*COSX
C
C        ROTATE L AND M COLUMNS
C
      ILQ=N*(L-1)
      IMQ=N*(M-1)
      DO 125 I=1,N
      IQ=(I*I-I)/2
      IF(I-L) 80,115,80
   80 IF(I-M) 85,115,90
   85 IM=I+MQ
      GO TO 95
   90 IM=M+IQ
   95 IF(I-L) 100,105,105
  100 IL=I+LQ
      GO TO 110
  105 IL=L+IQ
```

```
  110 X=A(IL)*COSX-A(IM)*SINX
      A(IM)=A(IL)*SINX+A(IM)*COSX
      A(IL)=X
  115 IF(MV-1) 120,125,120
  120 ILR=ILQ+I
      IMR=IMQ+I
      X=R(ILR)*COSX-R(IMR)*SINX
      R(IMR)=R(ILR)*SINX+R(IMR)*COSX
      R(ILR)=X
  125 CONTINUE
      X=2.0*A(LM)*SINCS
      Y=A(LL)*COSX2+A(MM)*SINX2-X
      X=A(LL)*SINX2+A(MM)*COSX2+X
      A(LM)=(A(LL)-A(MM))*SINCS+A(LM)*(COSX2-SINX2)
      A(LL)=Y
      A(MM)=X
C
C        TESTS FOR COMPLETION
C
C        TEST FOR M = LAST COLUMN
C
  130 IF(M-N) 135,140,135
  135 M=M+1
      GO TO 60
C
C        TEST FOR L = SECOND FROM LAST COLUMN
C
  140 IF(L-(N-1)) 145,150,145
  145 L=L+1
      GO TO 55
  150 IF(IND-1) 160,155,160
  155 IND=0
      GO TO 50
C
C        COMPARE THRESHOLD WITH FINAL NORM
C
  160 IF(THR-ANRMX) 165,165,45
C
C        SORT EIGENVALUES AND EIGENVECTORS
C
  165 IQ=-N
      DO 185 I=1,N
      IQ=IQ+N
      LL=I+(I*I-I)/2
      JQ=N*(I-2)
      DO 185 J=I,N
      JQ=JQ+N
      MM=J+(J*J-J)/2
      IF(A(LL)-A(MM)) 170,185,185
  170 X=A(LL)
      A(LL)=A(MM)
      A(MM)=X
      IF(MV-1) 175,185,175
  175 DO 180 K=1,N
      ILR=IQ+K
      IMR=JQ+K
      X=R(ILR)
      R(ILR)=R(IMR)
  180 R(IMR)=X
  185 CONTINUE
```

```
      RETURN
      END
```

```
C       ************************************************************************
C
C       DATA FOR
C            PROGRAM 2.3 :   POP2
C
C       ************************************************************************
 &VALS2
 PEIGEN=.FALSE..
 PD=.FALSE..
 PEIGES=.TRUE..
 PDS=.TRUE..
 &END
 &VALS3
 NT=10.
 SAGE=12.
 EAGE=44.
 SYEAR=56.
 FYEAR=72.
 &END
 &VALS4
 NEV=4.
 ICL=2.ICT=2.IRL=2.IRT=2.
 &END
```

```
EOF1PROG2                          00010001000100 75225 75230 0000510001051 B6500
EOF2F00022000223010000000000000000006810              00
VOL1                               652020601
HDR1PROG3                          00010001000100 75225 75230 0000000000000 B6500
HDR2F00022000223010000000000000000006810              00
```

PARAMETERS USED IN THIS RUN

```
&VALS2 PEIGEN = .FALSE., PD = .FALSE., PEIGES = .TRUE., PDS = .TRUE.,&END
&VALS3 NT =  10, SAGE =  12, EAGE =  44, SYEAR =  56, EYEAR =  72,&END
&VALS4 NEV =  4, ICL =  2, ICT =  2, IRL =  2, IRT =  2,&END
```

EIGEN VALUES,EIGEN VECTORS AND D MATRIX FOR TIME PERIOD STARTING AT THE YEAR 1956

EIGEN VALUE = 0.50339

A(T) OBTAINED FROM 10 BY 10 MATRIX IS

0.000000	0.000000	-0.267580	-0.244634	-0.172451
-0.163833	-0.154151	0.103356	0.000000	0.000000

D(X) OBTAINED AS (G(X,T).A(T))/A(T).A(T))

0.000000	0.000000	-0.039202	0.020085	0.020310
-0.060708	-0.074521	-0.030278	0.016176	0.082982
0.142469	0.191471	0.260078	0.277968	0.256768
0.265662	0.274054	0.309113	0.343627	0.423394
0.480056	0.491918	0.472137	0.459891	0.386322
0.287249	0.195803	0.202221	0.218597	0.261105
0.321830	0.000000	0.000000		

EIGEN VALUE = 0.08680

A(T) OBTAINED FROM 10 BY 10 MATRIX IS

0.000000	0.000000	0.010185	0.027454	0.025076
0.001099	-0.139714	-0.073444	0.000000	0.000000

D(X) OBTAINED AS (G(X,T).A(T))/A(T).A(T))

0.000000	0.000000	-0.672286	-0.320124	0.170898
0.234904	-0.108263	-0.292862	-0.323207	-0.367120
-0.365730	-0.365635	-0.308870	-0.221815	-0.201111
-0.089330	0.021610	-0.035787	-0.167464	-0.344784
-0.368976	-0.121208	0.100550	0.289485	0.429352
0.485120	0.578374	0.662674	0.474759	0.069363
-0.185877	0.000000	0.000000		

EIGEN VALUES,EIGEN VECTORS AND D MATRIX FOR TIME PERIOD STARTING

AT THE YEAR 1956

EIGEN VALUE = 0.04346

A(T) OBTAINED FROM 10 BY 10 MATRIX IS

0.000000	0.000000	0.025419	0.050181	-0.005807
-0.094823	-0.001681	0.022079	0.000000	0.000000

D(X) OBTAINED AS (G(X,T).A(T))/A(T).A(T))

0.000000	0.000000	0.415849	0.393008	-0.098282
-0.584839	-0.515639	-0.364268	-0.266599	-0.207611
-0.189848	-0.042230	0.100669	0.012959	0.152727
0.291072	0.110087	-0.130250	-0.491298	-0.490639
0.031987	0.232708	0.260057	0.131696	0.035973
0.401302	0.418763	-0.092841	-0.731979	-0.712996
0.194209	0.000000	0.000000		

EIGEN VALUE = 0.02183

A(T) OBTAINED FROM 10 BY 10 MATRIX IS

0.000000	0.000000	0.032563	-0.015180	-0.031825
0.001066	-0.002643	-0.006981	0.000000	0.000000

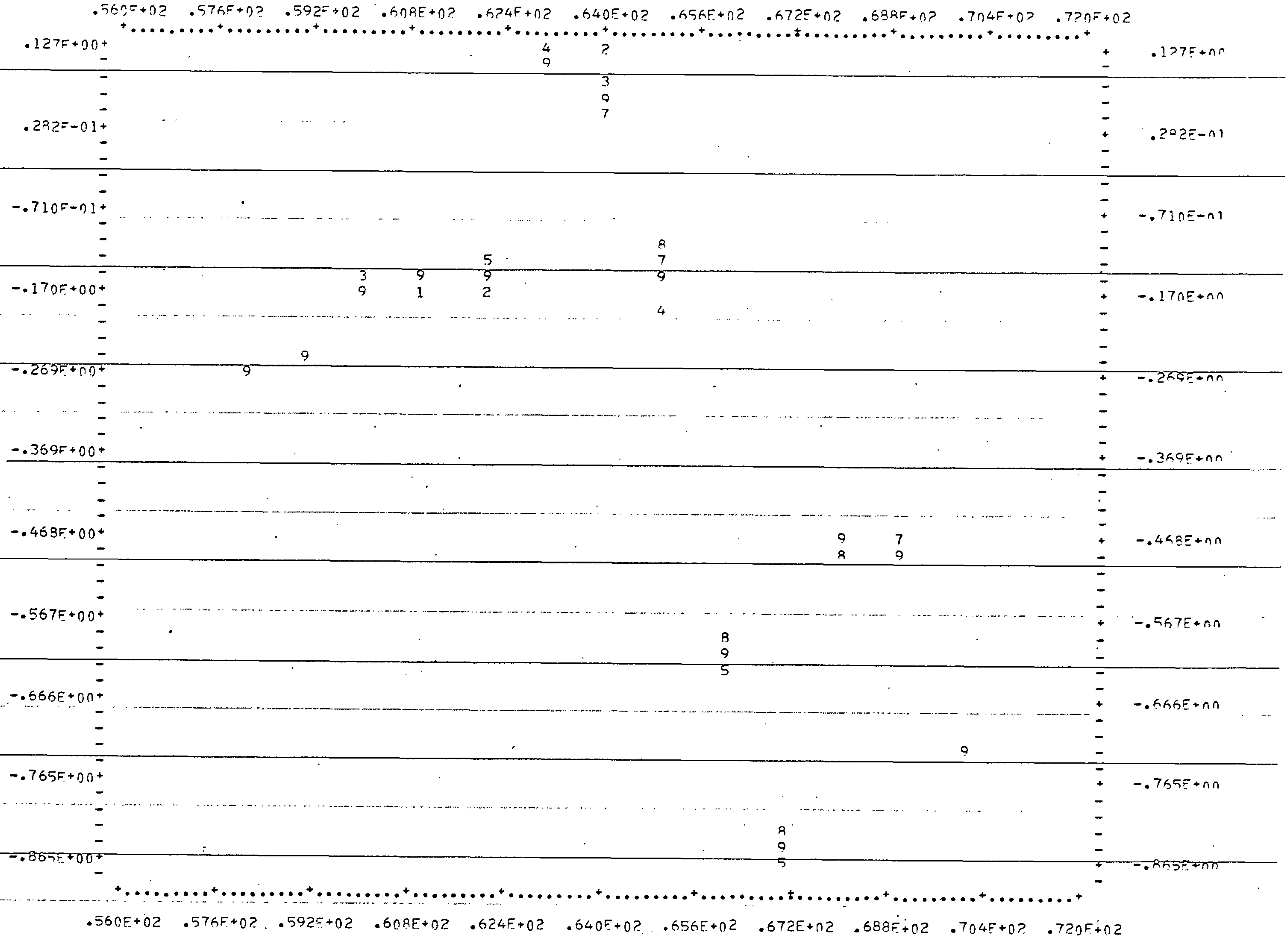
D(X) OBTAINED AS (G(X,T).A(T))/A(T).A(T))

0.000000	0.000000	-0.869929	-1.745564	-0.773252
0.088287	-0.025691	-0.127382	0.031162	-0.041943
0.126342	0.188378	-0.144499	0.217249	0.584324
0.223465	-0.148576	-0.653320	-0.439793	-0.072333
0.011259	0.392439	0.159370	-0.162865	0.449829
0.601646	-0.209182	-1.417385	-0.725837	0.541649
-0.133590	0.000000	0.000000		

SUMMARY OF EIGEN VALUES

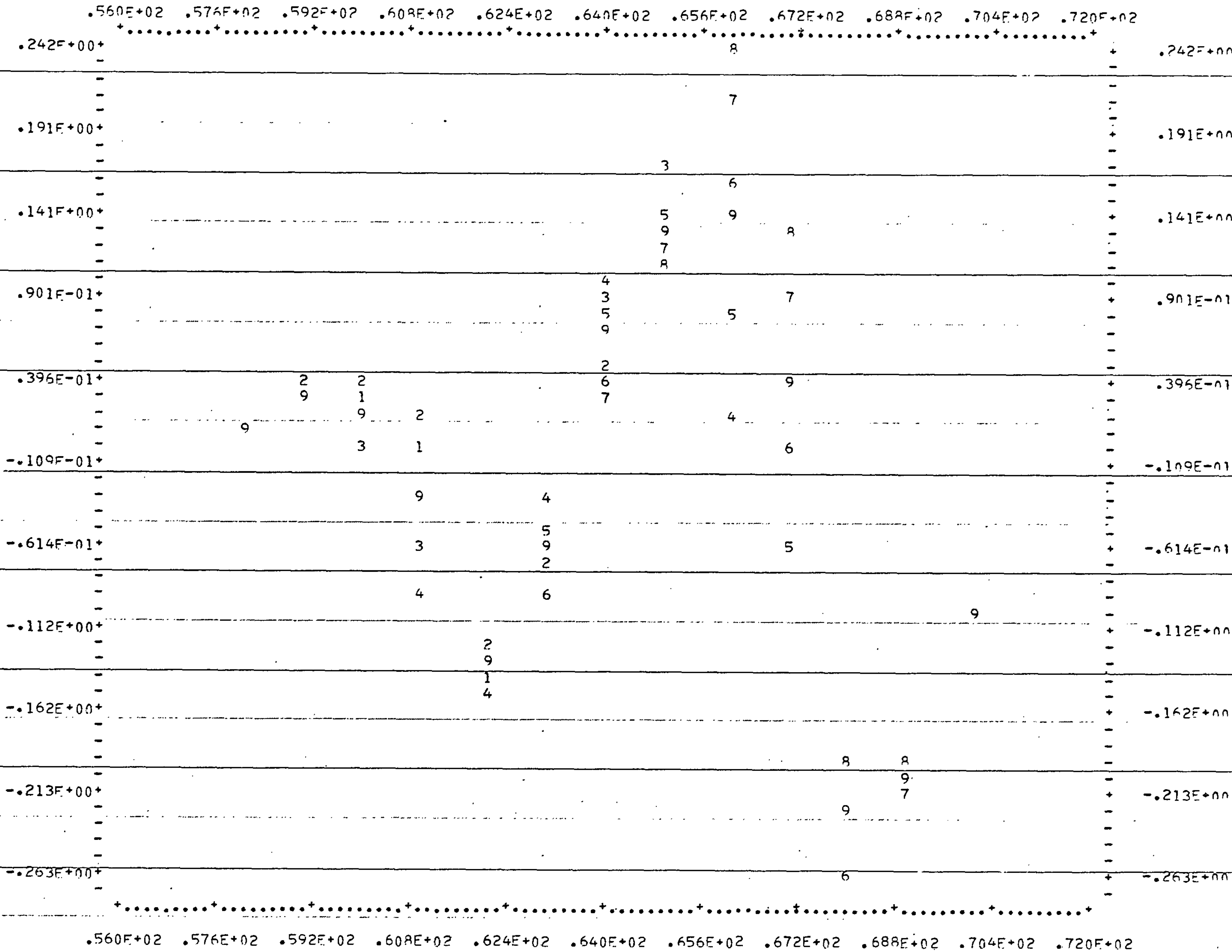
0.50339	0.08680	0.04346	0.02183
0.40210	0.09033	0.04305	0.02114
0.30959	0.14562	0.06767	0.02533
1.06030	0.15260	0.05429	0.02452
2.88158	0.15698	0.04239	0.03453
3.28012	0.25510	0.07369	0.02426
3.77877	0.35535	0.09139	0.02242
5.07600	0.37180	0.07918	0.04842

CHART 101



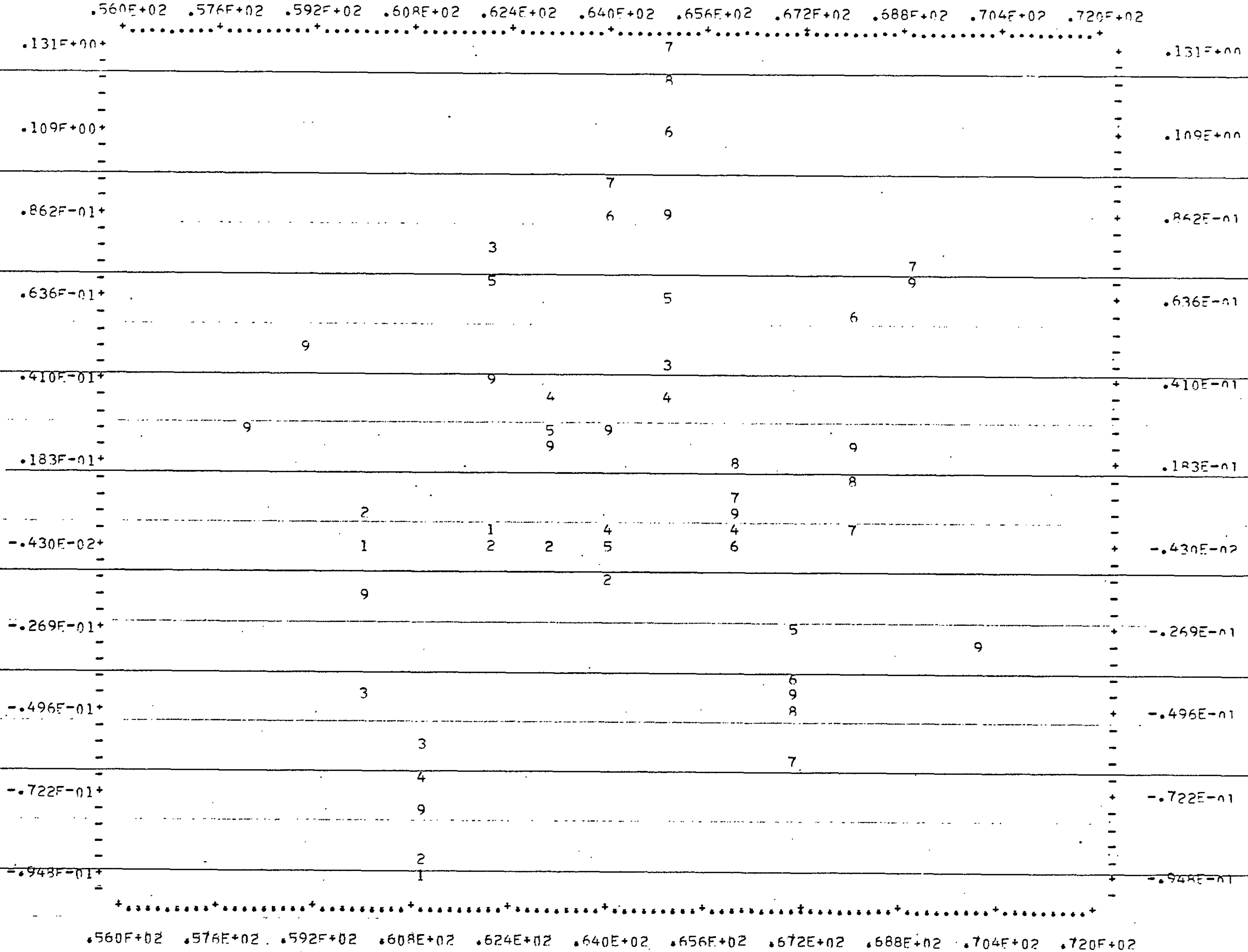
$a_1(t)$ - First Time Component

CHART 102



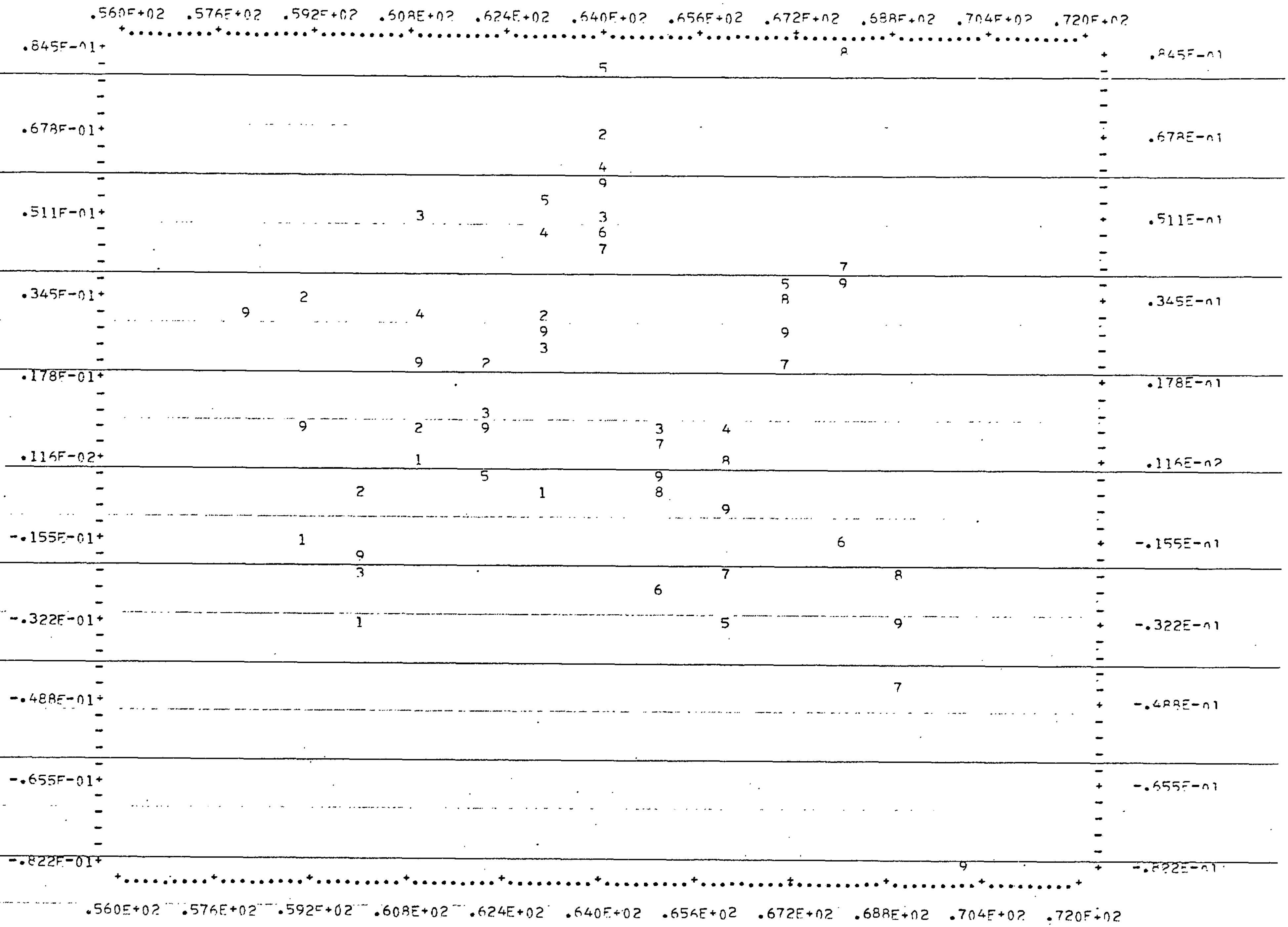
$a_2(t)$ - Second Time Component

CHART 103



$a_3(t)$ - Third Time Component

CHART 104



$a_4(t)$ - Fourth Time Component

AVERAGES OF EIGEN VECTORS

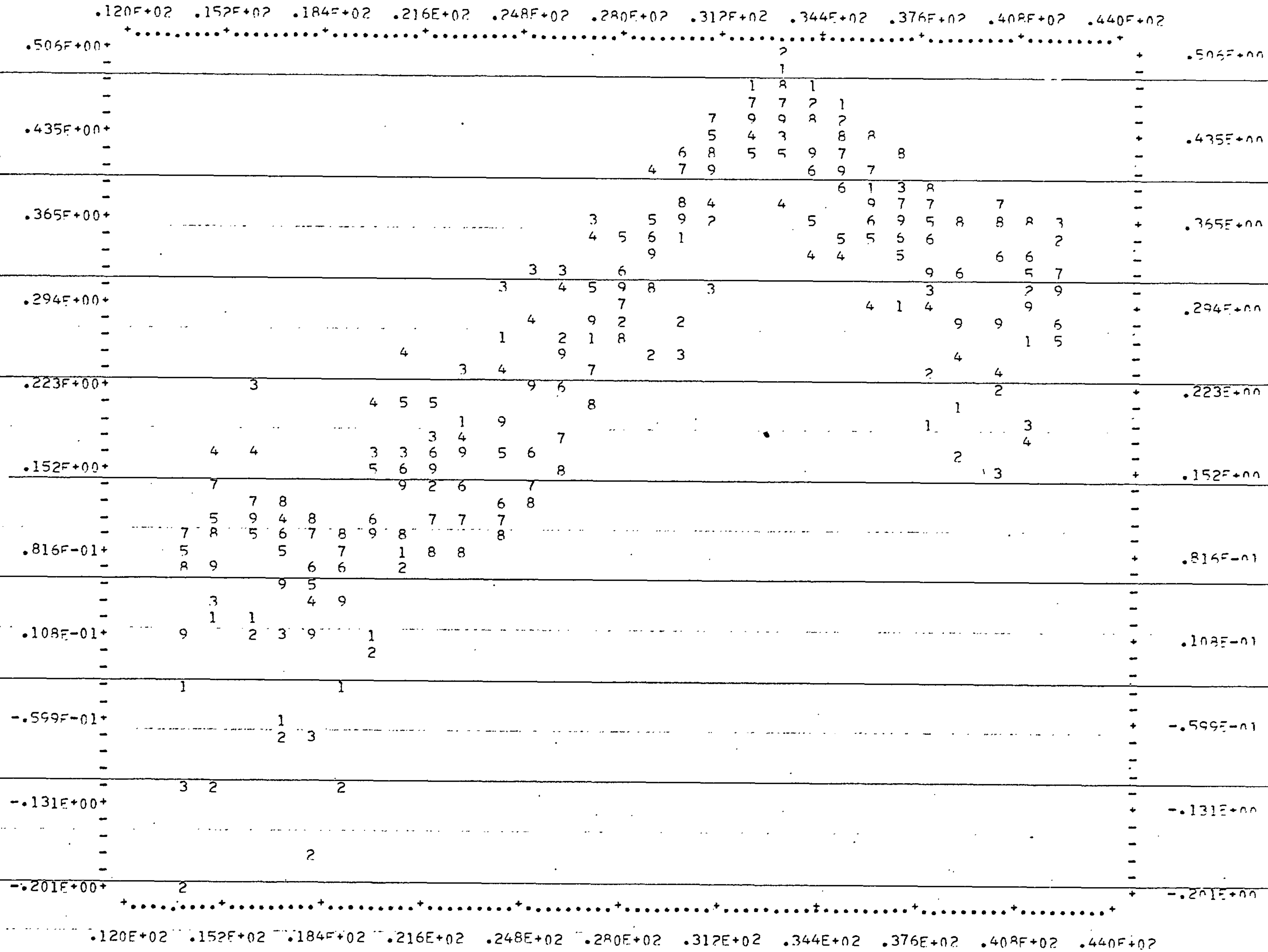
0.00000	0.00000	-0.26758	-0.24528	-0.16406	-0.15926
-0.15094	0.11465	0.07610	-0.15359	-0.60086	-0.83705
-0.47282	-0.47894	-0.72492	0.00000	0.00000	

0.00000	0.00000	0.01018	0.03214	0.02180	-0.03447
-0.13266	-0.05895	0.06681	0.13455	0.14122	0.03928
-0.22728	-0.20267	-0.10317	0.00000	0.00000	

0.00000	0.00000	0.02542	0.05024	-0.01673	-0.07732
0.04185	0.02415	0.02653	0.08466	0.00695	-0.04477
0.02393	0.06994	-0.03349	0.00000	0.00000	

0.00000	0.00000	0.03256	0.00908	-0.01962	0.02260
0.00704	0.02947	0.05790	-0.00358	-0.00935	0.02805
0.03691	-0.03337	-0.08216	0.00000	0.00000	

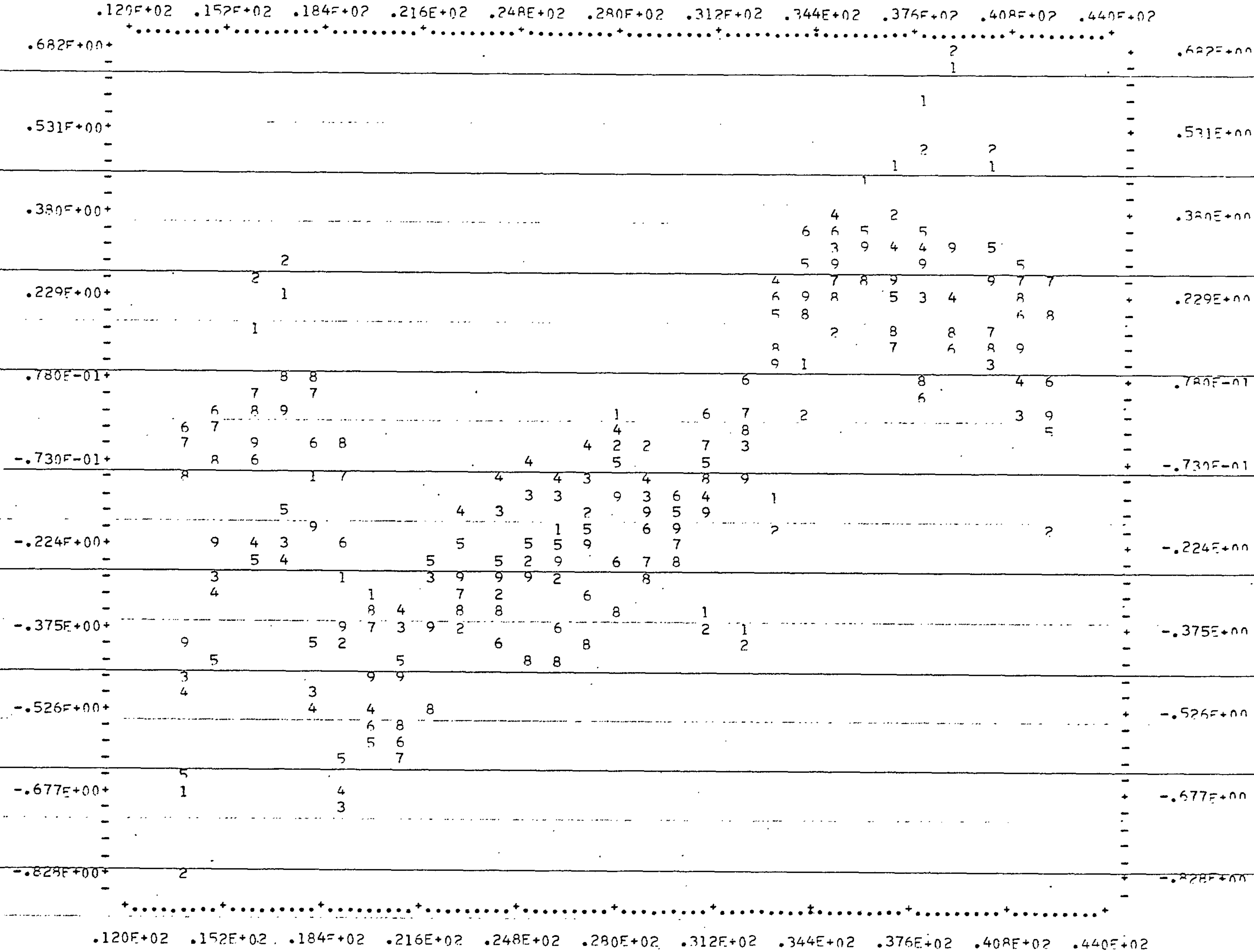
CHART 201



$d_1(x)$ - First Age Component

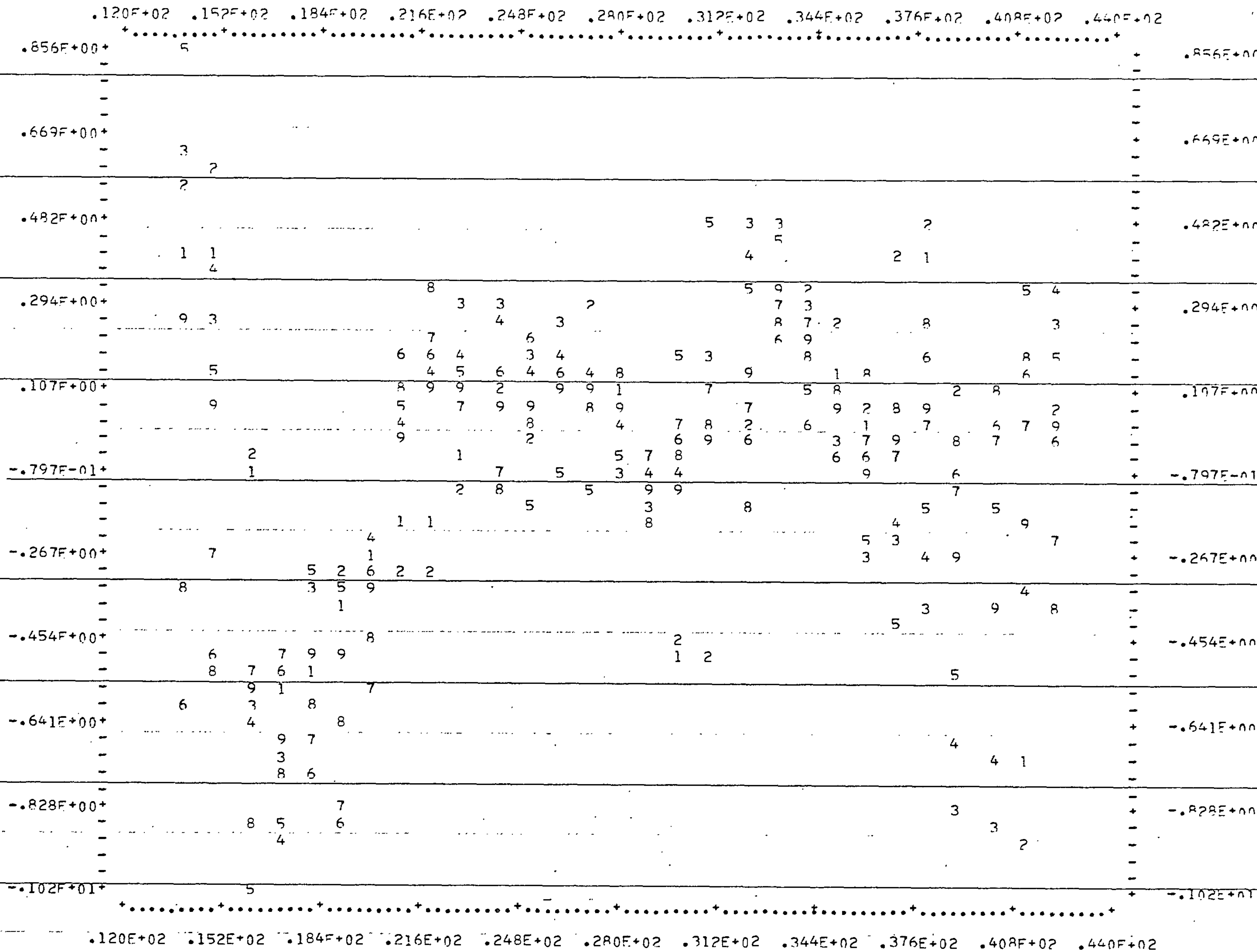
-402-

CHART 202



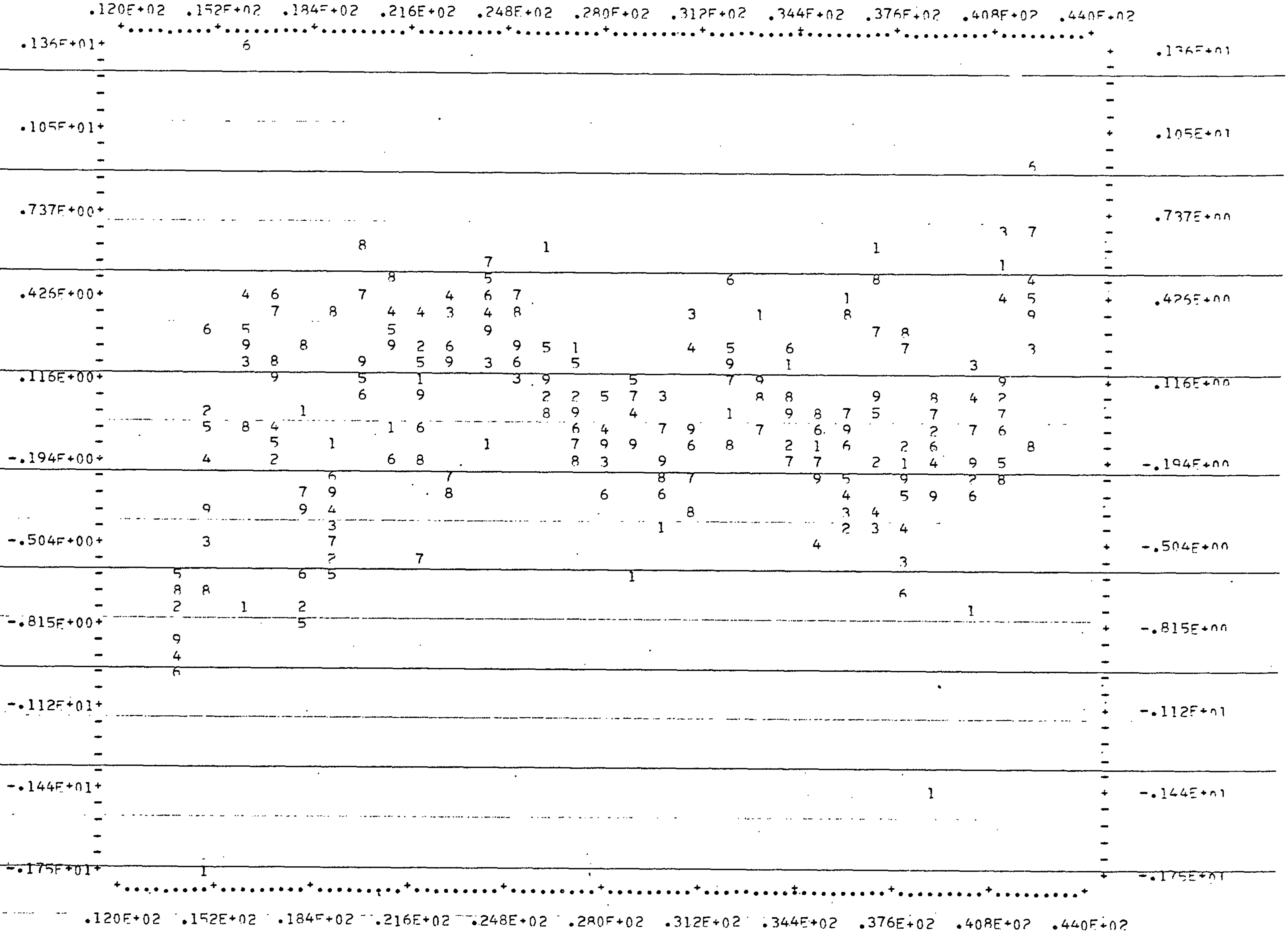
$d_2(x)$ – Second Age Component

CHART 203



$d_3(x)$ - Third Age Component

CHART 204



$d_4(x)$ - Fourth Age Component

AVERAGES OF D-VECTORS

0.00000	0.00000	0.00999	0.07335	0.11198	0.05014
0.00662	0.03942	0.10248	0.14117	0.15363	0.16157
0.19377	0.22440	0.24663	0.28580	0.30769	0.33060
0.36291	0.40647	0.44860	0.45266	0.41882	0.40010
0.38033	0.35944	0.31556	0.28220	0.27988	0.29109
0.30911	0.00000	0.00000			

0.00000	0.00000	-0.41022	-0.21447	-0.03870	0.00690
-0.20510	-0.38710	-0.45814	-0.45404	-0.36460	-0.28699
-0.27815	-0.27881	-0.26905	-0.21340	-0.13591	-0.15874
-0.18430	-0.14956	-0.10872	0.10356	0.22101	0.28481
0.30890	0.26572	0.27523	0.30806	0.24850	0.14890
0.02760	0.00000	0.00000			

0.00000	0.00000	0.25144	0.05421	-0.54968	-0.66729
-0.50392	-0.50564	-0.33419	-0.00593	0.09335	0.09792
0.07904	0.05757	0.11871	0.12398	0.07301	-0.10975
-0.11735	-0.01413	0.15247	0.31754	0.21523	0.08038
-0.06883	-0.00293	0.07875	-0.28314	-0.38251	-0.20447
0.04975	0.00000	0.00000			

0.00000	0.00000	-0.84741	-0.40283	0.25564	0.11106
-0.40335	-0.31948	0.19937	0.22344	0.04916	0.15554
0.29483	0.24972	0.10948	-0.02702	-0.12488	-0.10456
-0.18221	-0.05456	0.17612	0.09927	-0.03720	-0.27040
-0.09396	0.04689	-0.23273	-0.30901	-0.20341	0.14194
0.35895	0.00000	0.00000			

2.3 Simulation from Estimated Parameters

To test the accuracy of the estimated $\underline{a}_r$, $\underline{d}_r$, and $\underline{b}$, a simulation was performed integrating forward beginning with the fertility profile of 1958 from $\overline{f}$ year by year to 1970 (allowing for two years time lost in "bordering" $M^T M$).

The numerical procedure, based on the Lax-Wendroff [15] conservation formula is described in sufficient detail in the Interim Report [3]. For the values used, the condition for stability is satisfied by using equal steps in time and age.

While errors at some individual age levels become significant over the period of integration, the effect of this model on an overall population simulation will only be reflected through an integral over the population density which gives the annual birthrate, and hence the actual propagated error would be considerably smaller.

It was observed that little improvement results from either reducing the step size to .25 or on using the subdominant vector pairs $(\underline{a}_2,\underline{d}_2)$, $(\underline{a}_3,\underline{d}_3)$, $(\underline{a}_4,\underline{d}_4)$. This result also holds for modified versions of the algorithm, and suggests that further efforts to improve accuracy are probably not justified by the increased storage requirements and execution times likely to be required.

```
$JOB    ACCT-NUM,VERNER,TIME=60
C       ****************************************************************************
C
C            PROGRAM 2.4 :  SIM
C
C       ****************************************************************************
```

```
3JOB SIM; CHARGE BERGHOUT; CLASS 3;
USER 080002/VHJ;
BEGIN
PRINTLIMIT=10000;
PROCESSTIME=300;
COMPILE SIM FORTRAN;
FILE FILE4(KIND=PACK,TITLE=DATA/XXAV,FILETYPE=7);
DATA
$SET $


3DATA FILE5


3END JOB
```

```
$SET $
C       ************************************************************
C
C       PROGRAM : 2.4
C       TITLE : SIM
C       PURPOSE : SIMULATE FERTILITY FROM ESTIMATES
C       DATA : DATA/XXAV,FBAR(X,T),F(T)
C       OUTPUT : COMPARISON OF ACTUAL AND SIMULATED F(X,T)
C
C       ************************************************************
C
C       THIS PROGRAM RUNS A SIMULATION OF FERTILITY FROM ESTIMATED
C       VALUES OF THE PARAMETERS  A,B,  AND  D.
C
C       REAL FA(AGES,YRS),FS(AGES),FT(YRS)
C
C       ************************************************************
        INTEGER IR,AGES,YRS
        REAL FA(29,13),FS(29),FT(13)
C       ************************************************************
C
C       SET UP INTERVALS OF SIMULATION AND INPUT VALUES OF FERTILITY
C       FROM WHICH PARAMETERS ARE ESTIMATED, AND WITH WHICH SIMULATED
C       VALUES ARE TO BE COMPARED
C
C       (IN THIS SIMULATION IN AGES  12-44  AND YEARS  1956-1972  TWO
C       YEARS AT EACH END OF EACH INTERVAL ARE SUPPRESSED IN THE
C       ESTIMATION.  HENCE COMPARISON OCCURS FOR AGES  14-42  AND YEARS
C       1958-1970.)
C
C       ************************************************************
        AGES=29
        YRS=13
        CALL INACT(AGES,YRS,FA,FT)
C       ************************************************************
C
C       SET UP INITIAL VALUES OF FERTILITY FOR SIMULATION
C
C       ************************************************************
        DO 11 NDOM = 1,4
        DO 10 I=1,AGES
            FS(I)=FA(I,1)
   10       CONTINUE
C       ************************************************************
C
C       RUN SIMULATION AND COMPARE WITH ACTUAL VALUES AFTER EACH YEAR
C       NOTE:  IF  IR=0  DATA GIVES STARTING VALUES ONLY
C
C       ************************************************************
        DO 11 I=1,13
        IR=I-1
        CALL DATA(FS,IR,AGES,IF)
        CALL COMPAR(IR,AGES,YRS,FA,FS)
  11    CONTINUE
        STOP
        END

        SUBROUTINE INACT(AGES,YRS,FA,FT)
C       ************************************************************
```

```
C
C      THIS SUBROUTINE MUST PROVIDE THE ACTUAL VALUES OF FERTILITY
C      FROM WHICH THE ESTIMATION WAS MADE, AND THE TOTAL FERTILITY
C      FOR EACH YEAR.
C
C      THE FORMAT OF THESE ARRAYS IS:
C
C         FA - ACTUAL FERTILITY IS AN (AGES BY YRS) ARRAY IN WHICH
C      THE FERTILITY FROM WHICH FERTILITY IS SIMULATED IS THE FIRST
C      COLUMN.
C
C         FT - TOTAL FERTILITY IS A VECTOR OF LENGTH YRS,EACH ELEMENT
C      BEING THE SUM OF ALL FERTILITY FOR A PARTICULAR YEAR, STARTING
C      WITH THAT FOR THE INITIAL FERTILITY.
C
C      *********************************************************************
       INTEGER AGES,YRS
       REAL FBAR(51,17)
       REAL FT(YRS),FA(AGES,YRS)
C
C      READREADREADREADREADREADREADREADREADREADREADREADREADREADREADREAD
C      READ  FBAR(X,T)
       READ(5,100)((FBAR(I,J),I=1,51),J=1,17)
   100 FORMAT(4F17.7)
C      READ  F(T)  AND SET EQUAL TO FT
       READ(5,101) (FT(I),I=1,YRS)
   101 FORMAT(8F10.8)
C      READREADREADREADREADREADREADREADREADREADREADREADREADREADREADREAD
C
       DO 102 J=1,YRS
          L=J+2
       XX=FT(J)/3.85865
       DO 102 I=1,AGES
          K=I+14
          FA(I,J)=FBAR(K,L)*XX
   102    CONTINUE
       RETURN
       END

       SUBROUTINE DATA(F,IR,IC,IF)
C      *********************************************************************
C
C      THIS SUBROUTINE PROPOGATES THE FERTILITY ONE YEAR FORWARD IN TIME
C
C      *********************************************************************
$SET OWN
       INTEGER IR,IC,NDOM,IF,NK
       INTEGER YRS
       DATA NDOM/0/
       DATA NK/1/
       DATA YRS/13/
C      *********************************************************************
C
C      DIMENSIONED ARRAYS ARE
C
C      REAL D(AGES,4),A(YRS,4),F(AGES),B(YRS),FT(YRS+4)
C
C      IF STEP-NUMBER > 1 THE FOLLOWING EXTENDED VECTORS HAVE
C      DIFFERENT DIMENSIONS.
C      REAL D(NK*(AGES-1)+1,4),DE5(NK*(AGES-1)+2,4),AE(NK*(YRS-1)+1,4)
```

```
C      REAL BE(NK*(YRS-1)+1),FE(NK*(AGES-1)+1),USE(NK*(AGES-1)+3,4)
C      REAL FTE(NK*(YRS+3)+1),FET(NK*(AGES-1)+1)
C
C      IF NK IS NOT EQUAL TO  1 THE FOLLOWING ARRAYS ARE ALSO NEEDED:
C      REAL A1(3,AGES),H1(AGES),K1(2,AGES),D1(AGES),COF(4,AGES)
C
C      ******************************************************************
       REAL D(29,4),A(13,4),F(IC),B(13),HH,KK
       REAL FT(17)
       REAL DE(29,4),DE5(30,4),AE(13,4),FE(29),BE(13),USE(31,4),C
       REAL FTE(17),FET(29)
       REAL A1(3,29),H1(29),K1(2,29),D1(29),COF(4,29)
       IF (IR.GT.0) GO TO 61
       NDOM=NDOM+1
       IRE=NK*IR
       IF (NDOM.GT.1) GO TO 59
       CALL INPUT(29,13,4,A,B,D,FT)
       KK=1./NK
       HH=KK
       ICE=NK*(IC-1)+1
       ICEM1=ICE-1
       ICEP1=ICE+1
       ICEP2=ICE+2
       IREP=1+NK*(YRS-1)
$POP OWN
       IF (NK.EQ.1) GO TO 29
C      ******************************************************************
C
C      USE SPLINE ROUTINES TO INTERPOLATE VECTORS  A,B,D,F,FT  WHEN
C      THE STEP-NUMBER IS GREATER THAN 1.
C
C      ******************************************************************
       DO 11 I=1,13
   11     K1(1,I)=I-1
       CALL SETUP (13,A1,H1,K1)
       DO 14 J=1,4
          DO 12 I=1,13
   12        K1(2,I)=A(I,J)
          CALL SOLVE (A1,K1,H1,13,D1,COF)
          X=-KK
          I1=0
          DO 14 I=1,12
   13        X=X+KK
             I1=I1+1
             AE(I1,J)=COF(4,I)+X*(COF(3,I)+X*(COF(2,I)+X*COF(1,I)))
             IF (X.LT.I-.0001) GO TO 13
   14        CONTINUE
       DO 17 I=1,13
   17     K1(2,I)=B(I)
       CALL SOLVE(A1,K1,H1,13,D1,COF)
       X=-KK
       I1=0
       DO 18 I=1,12
   16     X=X+KK
          I1=I1+1
          BE(I1)=COF(4,I)+X*(COF(3,I)+X*(COF(2,I)+X*COF(1,I)))
       IF (X.LT.I-.0001) GO TO 16
   18     CONTINUE
       DO 21 I=1,IC
   21     K1(1,I)=I-1
```

```
      CALL SETUP(IC,A1,H1,K1)
      DO 24 J=1,4
          DO 22 I=1,IC
   22         K1(2,I)=D(I,J)
          CALL SOLVE(A1,K1,H1,IC,D1,COF)
          X=-HH
          I1=0
          ICM1=IC-1
          DO 24 I=1,ICM1
   23         X=X+HH
              I1=I1+1
              DE(I1,J)=COF(4,I)+X*(COF(3,I)+X*(COF(2,I)+X*COF(1,I)))
              IF (X.LT.I-.0001) GO TO 23
   24     CONTINUE
      DO 25 I=1,IC
   25     K1(2,I)=F(I)
      CALL SOLVE(A1,K1,H1,IC,D1,COF)
      X=-HH
      I1=0
      DO 27 I=1,ICM1
   28     X=X+HH
          I1=I1+1
          FE(I1)=COF(4,I)+X*(COF(3,I)+X*(COF(2,I)+X*COF(1,I)))
          FET(I1)=FE(I1)
          IF (X.LT.I-.0001) GO TO 28
   27     CONTINUE
C     *****************************************************************
C
C     END OF SPLINE INTERPOLATION
C
C     *****************************************************************
      GO TO 34
   29 DO 30 I=1,ICE
          FE(I)=F(I)
          FET(I)=F(I)
          DO 30 J=1,4
              DE(I,J)=D(I,J)
   30         CONTINUE
      DO 33 I=1,IREP
          BE(I)=B(I)
          DO 33 J=1,4
              AE(I,J)=A(I,J)
   33         CONTINUE
   34 CONTINUE
      WRITE(6,201)
      WRITE(6,211) (FE(I),I=1,ICE)
      WRITE(6,202)
      WRITE(6,211) (BE(I),I=1,IREP)
      WRITE(6,203)
      WRITE(6,211) (AE(I,1),I=1,IREP)
      WRITE(6,204)
      WRITE(6,211) (DE(I,1),I=1,ICE)
      DO 32 J=1,4
      DO 31 I=2,ICE
          DE5(I,J)=.5*(DE(I,J)+DE(I-1,J))
   31     CONTINUE
          DE5(1,J)=.5*(3.*DE(1,J)-DE(2,J))
          DE5(ICEP1,J)=.5*(3.*DE(ICE,J)-DE(ICEM1,J))
   32     CONTINUE
  201 FORMAT('1',1X,///,' EXTENDED FERTILITY IS',//)
```

```
  202 FORMAT('1',1X,///,' EXTENDED B(T) IS',//)
  203 FORMAT('1',1X,///,' EXTENDED A1(T) IS',//)
  204 FORMAT('1',1X,///,' EXTENDED D1(T) IS',//)
  211 FORMAT(4F10.6,/)
C     ############################################################################
C
C     SET UP  C  FOR CHECKING TOTAL FERTILITY
C
C     ############################################################################
   59 CONTINUE
      DO 60 I=1,ICE
   60      FE(I)=FET(I)
      SUM=0.
      DO 67 I=1,IC
           SUM=SUM+FE(I)
 67        CONTINUE
C
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
      WRITE(6,301) IR,NDOM
  301 FORMAT('1','FERTILITY AT TIME',I5,'      # OF EIGEN VECTORS = ',I2)
      WRITE(6,302) SUM
  302 FORMAT('0','  THE SUM OF THOSE STARTING FERTILITIES SIMULATED IS',
     *F14.7)
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
C
      IF (NDOM.GT.1) GO TO 65
      C=SUM/FT(IR+2)
      GOTO 65
 61   CONTINUE
C     ############################################################################
C
C     THE PROPAGATION SCHEME USES A LAX-WENDROFF FINITE DIFFERENCE
C     APPROXIMATION FOR THE PARTIAL DIFFERENTIAL EQUATION WITH A TIME
C     AND AGE INCREMENT OF  1./NK  YEAR IN EACH DIRECTION
C
C     ############################################################################
      DO 48 K=1,NK
           IRE=IRE+1
      FAC=1.-BE(IRE)*KK
      DO 44 I=1,ICE
           DO 43 J=1,NDOM
   43           USE(I+1,J)=DE(I,J)*FE(I)
   44      FE(I)=FAC*FE(I)
      DO 46 J=1,NDOM
           USE(1,J)=2.*USE(2,J)-USE(3,J)
           USE(ICEP2,J)=2.*USE(ICEP1,J)-USE(ICE,J)
   46      CONTINUE
      DO 47 I=1,ICE
      SUM1=0.
      SUM2=0.
      DO 59 J1=1,NDOM
           SUM1=SUM1+AE(IRE,J1)*(USE(I+2,J1)-USE(I+1,J1))
           SUM2=SUM2+AE(IRE,J1)*(USE(I+1,J1)-USE(I   ,J1))
   59      CONTINUE
      DO 47 J=1,NDOM
           FE(I)=FE(I)-.5*(AE(IRE,J)*(USE(I+2,J)-USE(I,J)-
     &               (DE5(I+1,J)*SUM1-DE5(I,J)*SUM2)) )
   47      CONTINUE
   48      CONTINUE
      I1=1-NK
```

```
      DO 49 I=1,IC
      I1=I1+NK
         F(I)=FF(I1)
   49    CONTINUE
C     ******************************************************************
C
C     THE ESTIMATE OF WHAT THE TOTAL FERTILITY SHOULD BE IS JUST THE
C     PRO-RATED FRACTION OF THE ACTUAL TOTAL FERTILITY GIVEN BY THE
C     FRACTION OF THE SUM OF STARTING FERTILITY VALUES SIMULATED TO THE
C     TOTAL STARTING FERTILITY
C
C     ******************************************************************
      SUM=0.
      DO 63 I=1,IC
         SUM=SUM+F(I)
   63    CONTINUE
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
      WRITE(6,301) IR,NDOM
      WRITE(6,303) SUM
  303 FORMAT('0',' TOTAL FERTILITY IS  ',E14.7)
      SUM=C*FT(IR+2)
      WRITE(6,304) SUM
  304 FORMAT(' ',' AND SHOULD BE APPROXIMATELY  ',E14.7)
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
   62    CONTINUE
   65 CONTINUE
      RETURN
      END

      SUBROUTINE INPUT(AGES,YRS,NDOM,A,B,D,FT)
C     ******************************************************************
C
C     THIS SUBROUTINE MUST PROVIDE THE PARAMETER ARRAYS  A,B, D
C     IN THE FOLLOWING FORMAT:
C
C        A  IS AN (YRS BY NDOM) ARRAY WITH EACH COLUMN ONE OF THE
C        NDOM  DOMINANT MODES
C
C        B  IS A VECTOR HAVING  YRS  ELEMENTS
C
C        D IS AN (AGES BY NDOM) ARRAY WITH EACH COLUMN EQUAL TO ONE OF
C           THE DOMINANT MODES
C
C     THESE MAY BE OBTAINED, FOR EXAMPLE, BY CALLING THE ESTIMATION
C     PROCEDURE FOR A(T) AND D(X),  OR OTHERWISE BY READING THE
C     PREVIOUSLY STORED RESULTS
C
C     ******************************************************************
      INTEGER CH(80)
      INTEGER AGES,YRS,NDOM
      REAL A(YRS,NDOM),D(AGES,NDOM),B(YRS)
      REAL FT(17)
  109 FORMAT(10F8.6)
  110 FORMAT(80A1)
C
C     READREADREADREADREADREADREADREADREADREADREADREADREADREADREADREAD
      READ(5,110) CH
      READ(5,109) (FT(I),I=1,17)
      DO 71 I = 1,NDOM
   71 READ(4)(A(J,I),J=1,YRS)
```

```
      DO 72 I = 1,NDOM
   72 READ(4)(D(J,I),J=1,AGES)
C     READREADREADREADREADREADREADREADREADREADREADREADREADREADREADREAD
C
      DO 28 I=1,YRS
      B(I)=1.-FT(I+3)/FT(I+2)
 28         CONTINUE
C
C
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
      WRITE(6,201)
  201 FORMAT('1',' TOTAL FERTILITY FOR ALL YEARS IS',/)
      WRITE(6,111)(FT(I),I=1,17)
      WRITE(6,202)
  202 FORMAT('0','FAMILY SIZE REDUCTION RATE FOR SIMULATION YEARS IS',/
      WRITE(6,111)(B(I),I=1,YRS)
      WRITE(6,204)
  204 FORMAT('1','A(T) IS',/)
      DO 81 J = 1,NDOM
            WRITE(6,111) (A(I,J),I=1,YRS)
      WRITE(6,200)
  200 FORMAT(' ',1X)
   81       CONTINUE
      WRITE(6,205)
  205 FORMAT('0','D(X) IS',/)
      DO 82 J =1,NDOM
            WRITE(6,111) (D(I,J),I=1,AGES)
      WRITE(6,200)
   82       CONTINUE
  111 FORMAT (5F12.6)
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
C
C
      RETURN
      END

      SUBROUTINE COMPAR(IR,AGES,YRSP1,FA,FS)
C     ******************************************************************
C
C     THIS ROUTINE PRINTS COMPARISON OF ACTUAL AND SIMULATED FERTILITY
C
C     ******************************************************************
      INTEGER IR,AGES,YRSP1
      REAL ACT,ERR,PCT,FA(AGES,YRSP1),FS(AGES)
      WRITE(6,201)
  201 FORMAT('0',' AGE            ACTUAL        SIMULATED        ERROR        FRAC ER
     $ROR',//)
      DO 104 I=1,AGES
      ACT=FA(I,IR+1)
      SIM=FS(I)
      ERR=ACT-SIM
      IF (ACT.EQ.0) GOTO 46
      PCT=ERR/ACT
      GO TO 47
   46 PCT=999
   47 CONTINUE
      IA=I+13
  104 WRITE(6,103) IA,ACT,SIM,ERR,PCT
  103 FORMAT(I4,6X,4F12.8)
      RETURN
```

```
      END
      SUBROUTINE SETUP(N,A,H,K)
C     ******************************************************************
C
C     THIS ROUTINE SETS UP A TRIDIAGONAL MATRIX OF THE SPLINE EQUATION
C     IN A  3 BY N  ARRAY  -  FIRST ROW IS SUPERDIAGONAL
C                             SECOND ROW IS DIAGONAL
C                             THIRD ROW IS SUBDIAGONAL
C     AND THEN DECOMPOSES A TO  LU  SO THAT
C        FIRST ROW IS SUPERDIAGONAL OF U, SECOND ROW IS DIAGONAL OF U
C       THIRD ROW IS SUBDIAGONAL OF L, DIAGONAL OF L IS UNITY
C
C     ******************************************************************
      REAL A(3,N),H(N),K(2,N)
      NM1=N-1
      H(2)=K(1,2)-K(1,1)
      DO 10 I=2,NM1
           H(I+1)=K(1,I+1)-K(1,I)
           A(1,I)=H(I+1)/(H(I+1)+H(I))
           A(2,I)=2.0
           A(3,I)=1-A(1,I)
   10      CONTINUE
      A(1,1) = A(1,2)
      A(2,1) = A(2,2)
      A(3,1) = A(3,2)
      A(1,N) = A(1,N-1)
      A(2,N) = A(2,N-1)
      A(3,N) = A(3,N-1)
      A(1,2) = -2.
      A(2,2) = 2.
      A(3,2) = 0.
      A(1,N-1) = 0.
      A(2,N-1) = 2.
      A(3,N-1) = -2.
      NM1 = N-1
      DO 11 I=3,NM1
           A(3,I)=A(3,I)/A(2,I-1)
           A(2,I)=A(2,I)-A(3,I)*A(1,I-1)
   11      CONTINUE
      RETURN
      END
C
      SUBROUTINE SOLVE(A,K,H,N,D,COF)
C     ******************************************************************
C
C     THIS ROUTINE CALCULATES THE SECOND ORDER FINITE DIFFERENCES OF
C     THE SPLINE, AND THEN SOLVES  AM=D,  (BY FORWARD AND BACKWARD
C     SUBSTITUTION), PLACING M(VECTOR OF SECOND DERIVATIVES) IN D
C     END POINT CONDITIONS NOW USE THIRD ORDER FINITE DIFFERENCES
C     TO ESTIMATE THE THIRD ORDER DERIVATIVES AT X0+3H/2 AND XN-3H/2.
C
C     ******************************************************************
      REAL H(N),D(N),A(3,N),K(2,N),COF(4,N)
      D(2)=(K(2,2)-K(2,1))/H(2)
      NM1=N-1
      DO 12 I=2,NM1
           D(I+1)=(K(2,I+1)-K(2,I))/H(I+1)
           D(I)=6*(D(I+1)-D(I))/(H(I+1)+H(I))
```

```
   12         CONTINUE
          CA=-6./(H(2)*(H(2)+H(3))*(H(2)+H(3)+H(4)))
          CB=6./(H(2)*H(3)*(H(4)+H(3)))
          CC=-6./(H(3)*H(4)*(H(2)+H(3)))
          CD=6./(H(4)*(H(4)+H(3))*(H(2)+H(3)+H(4)))
          D(1)=-2.*H(3)*(CA*K(2,1)+CB*K(2,2)+CC*K(2,3)+CD*K(2,4))
          CA=-6./(H(N-2)*(H(N-2)+H(NM1))*(H(N-2)+H(NM1)+H(N)))
          CB=6./(H(N-2)*H(NM1)*(H(N)+H(NM1)))
          CC=-6./(H(NM1)*H(N)*(H(N-2)+H(NM1)))
          CD=6./(H(N)*(H(N)+H(NM1))*(H(N-2)+H(NM1)+H(N)))
          D(N)=2.*H(NM1)*(CA*K(2,N-3)+CB*K(2,N-2)+CC*K(2,NM1)+CD*K(2,N))
          T1 = D(2)
          T2 = D(NM1)
          D(2) = D(1)
          D(NM1) = D(N)
          DO 13 I=3,NM1
              D(I)=D(I)-A(3,I)*D(I-1)
   13         CONTINUE
          D(NM1) = D(NM1)/A(2,NM1)
          DO 14 I=3,NM1
              J=N+1-I
              D(J)=(D(J)-A(1,J)*D(J+1))/A(2,J)
   14         CONTINUE
          D(1) = (T1 - A(2,1)*D(2) - A(1,1)*D(3))/A(3,1)
          D(N) = (T2 - A(2,N)*D(NM1) - A(3,N)*D(N-2))/A(1,N)
          CALL POLLY(N,D,K,H,COF)
          RETURN
          END
C

          SUBROUTINE POLLY(N,M,K,H,COF)
C         ************************************************************
C
C         THIS ROUTINE COMPUTES THE COEFFICIENTS OF THE SPLINE POLYNOMIAL
C           ON EACH SUBINTERVAL
C         K IS THE ARRAY OF DATA POINTS
C         H IS THE VECTOR OF SUBINTERVAL LENGTHS
C         M IS THE SOLUTION VECTOR TO THE EQUATION AM=D
C
C         ************************************************************
          REAL M(N),K(2,N),H(N),COF(4,N)
          NM1=N-1
          DO 11 I=1,NM1
          COF(1,I)=(M(I+1)-M(I))/(6.*H(I+1))
          COF(2,I)=(K(1,I+1)*M(I)-K(1,I)*M(I+1))/(2.*H(I+1))
          DD=M(I+1)*K(1,I)*K(1,I)-M(I)*K(1,I+1)*K(1,I+1)
         &+2.*K(2,I+1)-2.*K(2,I)
          COF(3,I)=(DD/(2.*H(I+1)))+H(I+1)*(M(I)-M(I+1))/6.
          DD=M(I)*(K(1,I+1)**3)-M(I+1)*(K(1,I)**3)+6.*K(1,I+1)*K(2,I)
         &-6.*K(1,I)*K(2,I+1)+K(1,I)*M(I+1)*(H(I+1)**2)-K(1,I+1)*M(I)
         &*(H(I+1)**2)
          COF(4,I)=DD/(H(I+1)*6.)
   11     CONTINUE
          DO 16 J=1,4
   16     COF(J,N)=COF(J,NM1)
          RETURN
            END
```

```
C     ******************************************************************
C
C     DATA FOR
C           PROGRAM 2.4 :   SIM
C
C     ******************************************************************
   0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
   0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
   0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
   0.7429816E-05    0.1634927E-03    0.1006141E-02    0.4942421E-02
   0.1726509E-01    0.4614777E-01    0.8778054E-01    0.1287903E 00
   0.1733888E 00    0.2095979E 00    0.2266342E 00    0.2462188E 00
   0.2535967E 00    0.2474782E 00    0.2369086E 00    0.2169245E 00
   0.2116948E 00    0.1892552E 00    0.1809314E 00    0.1552396E 00
   0.1511665E 00    0.1356698E 00    0.1278709E 00    0.1152725E 00
   0.1045548E 00    0.8492929E-01    0.7790750E-01    0.6245709E-01
   0.5043102E-01    0.3700228E-01    0.3078149E-01    0.2018753E-01
   0.1298582E-01    0.7232107E-02    0.3469116E-02    0.1403072E-02
   0.7913960E-03    0.4042918E-03    0.1762484E-03    0.0000000E 00
   0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
   0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
   0.0000000E 00    0.0000000E 00    0.0000000E 00    0.2033752E-04
   0.2206279E-03    0.1024708E-02    0.5051389E-02    0.1962170E-01
   0.4878248E-01    0.9214234E-01    0.1379995E 00    0.1777333E 00
   0.2134188E 00    0.2316796E 00    0.2428612E 00    0.2487651E 00
   0.2482604E 00    0.2353446E 00    0.2238681E 00    0.2067142E 00
   0.1883027E 00    0.1754966E 00    0.1528666E 00    0.1484947E 00
   0.1346632E 00    0.1217132E 00    0.1124704E 00    0.1036606E 00
   0.9143513E-01    0.7362139E-01    0.6127410E-01    0.4957513E-01
   0.3556923E-01    0.3062474E-01    0.2014163E-01    0.1207764E-01
   0.6471954E-02    0.3641624E-02    0.1707581E-02    0.7939595E-03
   0.2838371E-03    0.2689431E-03    0.0000000E 00    0.0000000E 00
   0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
   0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
   0.0000000E 00    0.0000000E 00    0.2588560E-04    0.1491891E-03
   0.1273050E-02    0.5347177E-02    0.1883900E-01    0.4806599E-01
   0.9110671E-01    0.1400929E 00    0.1840963E 00    0.2128937E 00
   0.2371737E 00    0.2453404E 00    0.2449850E 00    0.2451515E 00
   0.2403407E 00    0.2223017E 00    0.2128390E 00    0.1893021E 00
   0.1788046E 00    0.1522348E 00    0.1468166E 00    0.1339506E 00
   0.1220511E 00    0.1081653E 00    0.1001368E 00    0.8826882E-01
   0.7796538E-01    0.5777969E-01    0.4844318E-01    0.3322498E-01
   0.2818455E-01    0.1854310E-01    0.1211527E-01    0.6617039E-02
   0.3324505E-02    0.1627151E-02    0.6927683E-03    0.2283992E-03
   0.1625607E-03    0.0000000E 00    0.0000000E 00    0.0000000E 00
   0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
   0.0000000E 00    0.0000000E 00    0.0000000E 00    0.0000000E 00
   0.5848482E-05    0.1815752E-04    0.1524006E-03    0.1269283E-02
   0.5595200E-02    0.1934955E-01    0.4837923E-01    0.9324521E-01
   0.1400996E 00    0.1847297E 00    0.2206196E 00    0.2376751E 00
```

0.2509574E 00	0.2503754E 00	0.2412987E 00	0.2337543E 00
0.2253411E 00	0.2133908E 00	0.1957424E 00	0.1759517E 00
0.1534150E 00	0.1422451E 00	0.1304606E 00	0.1204642E 00
0.1059424E 00	0.9405357E-01	0.8589458E-01	0.7667202E-01
0.6208997E-01	0.4548723E-01	0.3417568E-01	0.2762331E-01
0.1803264E-01	0.1140561E-01	0.7085640E-02	0.2832796E-02
0.1616091E-02	0.8214479E-03	0.2615624E-03	0.1299269E-03
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.2325392E-04	0.1646105E-03	0.1154704E-02	0.5122337E-02
0.1919859E-01	0.4882310E-01	0.9206295E-01	0.1434420E 00
0.1879300E 00	0.2242298E 00	0.2421390E 00	0.2478324E 00
0.2520348E 00	0.2459021E 00	0.2306418E 00	0.2228054E 00
0.2129852E 00	0.1962739E 00	0.1846426E 00	0.1498247E 00
0.1424794E 00	0.1286335E 00	0.1172121E 00	0.1056809E 00
0.9236968E-01	0.8025026E-01	0.7419306E-01	0.6159350E-01
0.5064670E-01	0.3146533E-01	0.2666571E-01	0.1807094E-01
0.1073283E-01	0.6116446E-02	0.3089603E-02	0.1289498E-02
0.5448801E-03	0.2109780E-03	0.1840822E-03	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.3936497E-04
0.1471253E-03	0.1234930E-02	0.5219840E-02	0.1867957E-01
0.4758165E-01	0.9287566E-01	0.1414530E 00	0.1917900E 00
0.2248529E 00	0.2489690E 00	0.2569618E 00	0.2487700E 00
0.2470545E 00	0.2319062E 00	0.2164945E 00	0.2078308E 00
0.1943436E 00	0.1820471E 00	0.1604927E 00	0.1432767E 00
0.1281160E 00	0.1129494E 00	0.1009693E 00	0.9283632E-01
0.8022434E-01	0.7094383E-01	0.5928110E-01	0.4942951E-01
0.3609700E-01	0.2586606E-01	0.1796386E-01	0.1039849E-01
0.5931694E-02	0.2902867E-02	0.1597295E-02	0.7027730E-03
0.2392092E-03	0.1916328E-03	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.1611304E-03
0.1100139E-02	0.5341694E-02	0.1767021E-01	0.4392144E-01
0.8923173E-01	0.1421814E 00	0.1907406E 00	0.2300875E 00
0.2483481E 00	0.2600701E 00	0.2582705E 00	0.2435674E 00
0.2372926E 00	0.2197912E 00	0.2070794E 00	0.1895335E 00
0.1787836E 00	0.1601778E 00	0.1514808E 00	0.1268388E 00
0.1155834E 00	0.9909540E-01	0.9094930E-01	0.7870620E-01
0.6792599E-01	0.5682655E-01	0.4737460E-01	0.3526177E-01
0.2832320E-01	0.1706897E-01	0.9436630E-02	0.5381081E-02
0.2923402E-02	0.1375800E-02	0.4280740E-03	0.2046375E-03
0.1230038E-03	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.2249860E-04	0.1204436E-03	0.9681717E-03
0.5371928E-02	0.2012049E-01	0.4507285E-01	0.8456761E-01

0.1390227E 00	0.1895710E 00	0.2257624E 00	0.2520762E 00
0.2602199E 00	0.2594635E 00	0.2472610E 00	0.2352608E 00
0.2234864E 00	0.2071005E 00	0.1882225E 00	0.1772634E 00
0.1588416E 00	0.1505039E 00	0.1338879E 00	0.1146557E 00
0.1022107E 00	0.9151363E-01	0.7756662E-01	0.6906772E-01
0.5646708E-01	0.4487861E-01	0.3331536E-01	0.2657641E-01
0.1820700E-01	0.9477235E-02	0.5660690E-02	0.2936016E-02
0.1072021E-02	0.5649622E-03	0.1822288E-03	0.1219036E-03
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.1736314E-04	0.1416877E-03	0.1094503E-02	0.4919004E-02
0.1849992E-01	0.4963557E-01	0.8561850E-01	0.1313989E 00
0.1824397E 00	0.2255227E 00	0.2466831E 00	0.2591423E 00
0.2590284E 00	0.2530327E 00	0.2368238E 00	0.2219617E 00
0.2101821E 00	0.1905518E 00	0.1772934E 00	0.1539934E 00
0.1531928E 00	0.1347782E 00	0.1212198E 00	0.1001142E 00
0.9136510E-01	0.7847703E-01	0.6849623E-01	0.5679385E-01
0.4542725E-01	0.3310285E-01	0.2717116E-01	0.1823463E-01
0.1100488E-01	0.6262664E-02	0.3032315E-02	0.1214716E-02
0.4865748E-03	0.1983496E-03	0.1126833E-03	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.6215397E-05	0.1900049E-04
0.2125569E-03	0.1044521E-02	0.5343676E-02	0.2001397E-01
0.5058838E-01	0.1000955E 00	0.1386551E 00	0.1750458E 00
0.2194927E 00	0.2489910E 00	0.2569161E 00	0.2581220E 00
0.2506329E 00	0.2372339E 00	0.2237182E 00	0.2063728E 00
0.1900717E 00	0.1772207E 00	0.1513262E 00	0.1474659E 00
0.1340637E 00	0.1194983E 00	0.1062744E 00	0.9037650E-01
0.7943255E-01	0.7025135E-01	0.5633250E-01	0.4437540E-01
0.3281593E-01	0.2634889E-01	0.1717417E-01	0.1114297E-01
0.6952133E-02	0.2916923E-02	0.1267667E-02	0.4754490E-03
0.1937536E-03	0.1845764E-03	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.2060735E-04	0.2163455E-03
0.1342522E-02	0.6368209E-02	0.2268060E-01	0.5327580E-01
0.9917772E-01	0.1467553E 00	0.1820194E 00	0.2214158E 00
0.2402436E 00	0.2599525E 00	0.2661082E 00	0.2506132E 00
0.2415536E 00	0.2177314E 00	0.2094588E 00	0.1981256E 00
0.1589200E 00	0.1608635E 00	0.1391610E 00	0.1290765E 00
0.1205251E 00	0.9791631E-01	0.9413302E-01	0.7565463E-01
0.6731683E-01	0.5816970E-01	0.3945719E-01	0.3621977E-01
0.2382486E-01	0.1810371E-01	0.1119237E-01	0.5598128E-02
0.3102446E-02	0.1532112E-02	0.4638946E-03	0.2444400E-03
0.1360976E-03	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.2193426E-04	0.2478703E-03	0.1424491E-02

0.6939832E-02	0.2330431E-01	0.5433539E-01	0.1026180E 00
0.1558304E 00	0.2129785E 00	0.2227834E 00	0.2363340E 00
0.2588930E 00	0.2715958E 00	0.2593470E 00	0.2426100E 00
0.2242088E 00	0.2106870E 00	0.1886187E 00	0.1663788E 00
0.1477292E 00	0.1343106E 00	0.1218542E 00	0.1097493E 00
0.9979910E-01	0.8680904E-01	0.7769436E-01	0.6276107E-01
0.5106154E-01	0.3982237E-01	0.3015380E-01	0.2221074E-01
0.1571323E-01	0.9520765E-02	0.5431309E-02	0.2716314E-02
0.1153191E-02	0.5903142E-03	0.3588586E-03	0.7260863E-04
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.2246628E-04	0.2236100E-03	0.1733763E-02	0.7913966E-02
0.2478982E-01	0.5738202E-01	0.1026796E 00	0.1519760E 00
0.1982002E 00	0.2451878E 00	0.2428117E 00	0.2483238E 00
0.2688071E 00	0.2689522E 00	0.2546778E 00	0.2319517E 00
0.2151463E 00	0.1896352E 00	0.1741874E 00	0.1440163E 00
0.1363129E 00	0.1170444E 00	0.1052563E 00	0.9134817E-01
0.8146679E-01	0.7008237E-01	0.6129551E-01	0.4842377E-01
0.3848107E-01	0.2644810E-01	0.2167465E-01	0.1374268E-01
0.8248139E-02	0.4956193E-02	0.3049610E-02	0.1419656E-02
0.6481190E-03	0.1044638E-03	0.4576219E-04	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.2671527E-03	0.1822941E-02	0.9021621E-02	0.2665320E-01
0.5989974E-01	0.1023681E 00	0.1461299E 00	0.1856654E 00
0.2234071E 00	0.2657835E 00	0.2548606E 00	0.2599273E 00
0.2702767E 00	0.2659580E 00	0.2369433E 00	0.2231550E 00
0.1962218E 00	0.1779684E 00	0.1485134E 00	0.1321834E 00
0.1154602E 00	0.1054579E 00	0.9060305E-01	0.7756007E-01
0.6784779E-01	0.5828715E-01	0.4759468E-01	0.3566820E-01
0.2557244E-01	0.1899502E-01	0.1280696E-01	0.7447124E-02
0.4304644E-02	0.2219763E-02	0.1180247E-02	0.3952421E-03
0.1624286E-03	0.7658264E-04	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.2269397E-04	0.2308429E-03
0.2075878E-02	0.9624846E-02	0.3191707E-01	0.6668711E-01
0.1050972E 00	0.1461197E 00	0.1784285E 00	0.2115127E 00
0.2481316E 00	0.2835237E 00	0.2701750E 00	0.2609929E 00
0.2643824E 00	0.2523072E 00	0.2293909E 00	0.2022312E 00
0.1802447E 00	0.1499075E 00	0.1345164E 00	0.1111640E 00
0.9712094E-01	0.8830923E-01	0.7212412E-01	0.6215255E-01
0.5621058E-01	0.4325967E-01	0.3344726E-01	0.2382999E-01
0.1859563E-01	0.1131049E-01	0.6426115E-02	0.3995836E-02
0.1880914E-02	0.8638178E-03	0.3228260E-03	0.1022436E-03
0.3061809E-04	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00

0.0000000E 00	0.3218444E-04	0.2596709E-03	0.2107170E-02
0.9936295E-02	0.3130761E-01	0.6618536E-01	0.1071982E 00
0.1471151E 00	0.1840913E 00	0.2161806E 00	0.2493536E 00
0.2670345E 00	0.2680613E 00	0.2930642E 00	0.2637855E 00
0.2573879E 00	0.2264757E 00	0.2073898E 00	0.1796527E 00
0.1536861E 00	0.1332374E 00	0.1119388E 00	0.9786659E-01
0.8207232E-01	0.7238299E-01	0.5762001E-01	0.4698988E-01
0.3794770E-01	0.3120529E-01	0.2001548E-01	0.1510441E-01
0.1004597E-01	0.6521642E-02	0.3233414E-02	0.1113423E-02
0.6435823E-03	0.2916171E-03	0.5741679E-04	0.7639593E-04
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.8601016E-05
0.3439712E-04	0.3028943E-03	0.2249576E-02	0.1098468E-01
0.3506811E-01	0.7260209E-01	0.1124092E 00	0.1431631E 00
0.1702980E 00	0.1968122E 00	0.2334363E 00	0.2575313E 00
0.2904121E 00	0.3154174E 00	0.2771440E 00	0.2493343E 00
0.2392632E 00	0.2168028E 00	0.1860240E 00	0.1538441E 00
0.1345306E 00	0.1114675E 00	0.9396076E-01	0.7578874E-01
0.6721002E-01	0.5436332E-01	0.4397735E-01	0.3473518E-01
0.2713976E-01	0.1865069E-01	0.1498656E-01	0.8664083E-02
0.4691057E-02	0.2833721E-02	0.1506527E-02	0.6247566E-03
0.2911384E-03	0.1415498E-03	0.6554931E-04	

```
3.88774   3.9468   3.91022   3.85704   3.77676   3.68987   3.52102   3.16255
2.81195   2.60412  2.46239   2.41282   2.33947   2.18869
FR FOR 1956 TO 1972
3.85865 3.92929 3.88774 3.9468 3.91022 3.85704 3.77676 3.68987 3.52102 3.16255
2.81195 2.60412 2.46239 2.41282 2.33947 2.18869 2.02195
```

TOTAL FERTILITY FOR ALL YEARS IS

3.858650	3.920290	3.887740	3.946800	3.910220
3.857040	3.776760	3.689870	3.521020	3.162550
2.811950	2.604120	2.462390	2.412820	2.339470
2.188690	2.021950			

FAMILY SIZE REDUCTION RATE FOR SIMULATION YEARS IS

-0.015191	0.009268	0.013600	0.020814	0.023006
0.045760	0.101809	0.110860	0.073910	0.054425
0.020131	0.030400	0.064450		

A(T) IS

-0.267580	-0.245284	-0.164061	-0.159264	-0.150935
0.114646	0.076100	-0.153589	-0.600859	-0.837052
-0.472820	-0.479944	-0.724917		

0.010185	0.032143	0.021795	-0.034471	-0.132655
-0.058951	0.066811	0.134553	0.141224	0.039275
-0.227281	-0.202670	-0.103166		

0.025419	0.050240	-0.016732	-0.077323	0.041845
0.024148	0.026526	0.084657	0.006947	-0.044769
0.023930	0.069941	-0.033489		

0.032563	0.009084	-0.019622	0.022599	0.007039
0.029470	0.057903	-0.003575	-0.009346	0.028054
0.036906	-0.033371	-0.082158		

D(X) IS

0.009989	0.073350	0.111975	0.050145	0.006624
0.039422	0.102479	0.141169	0.153635	0.161569
0.193774	0.224397	0.246626	0.285798	0.307690
0.330599	0.362906	0.406465	0.448598	0.452659
0.418818	0.400096	0.380325	0.359445	0.315561
0.282202	0.279884	0.291089	0.309107	

-0.410221	-0.214475	-0.038699	0.006899	-0.205102
-0.387104	-0.459139	-0.454042	-0.364596	-0.286995
-0.278147	-0.279808	-0.269046	-0.213400	-0.135908
-0.158739	-0.184298	-0.149561	-0.108717	0.103561
0.221014	0.284806	0.308897	0.265724	0.275233
0.308064	0.249496	0.148897	0.027601	

0.251437	0.054209	-0.549679	-0.667290	-0.503924
-0.505644	-0.334192	-0.005928	0.093354	0.097917
0.079036	0.057571	0.118708	0.123983	0.073013
-0.109751	-0.117346	-0.014132	0.152474	0.317542
0.215226	0.080382	-0.068830	-0.002928	0.078746
-0.283141	-0.382506	-0.204473	0.049753	

-0.847410	-0.402827	0.255639	0.111065	-0.403355
-0.319476	0.199372	0.223445	0.049164	0.155543
0.294825	0.249717	0.109481	-0.027024	-0.124876
-0.104559	-0.182205	-0.054559	0.176123	0.099271
-0.037198	-0.270404	-0.093962	0.046889	-0.232727
-0.309010	-0.203413	0.141938	0.358949	

FERTILITY AT TIME 0 # OF EIGEN VECTORS = 1

THE SUM OF THOSE STARTING FERTILITIES SIMULATED IS .3843938E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00128265	0.00128265	0.00000000	0.00000000
15	0.00538749	0.00538749	0.00000000	0.00000000
16	0.01898103	0.01898103	0.00000000	0.00000000
17	0.04842835	0.04842835	0.00000000	0.00000000
18	0.09179355	0.09179355	0.00000000	0.00000000
19	0.14114905	0.14114905	0.00000000	0.00000000
20	0.18548418	0.18548418	0.00000000	0.00000000
21	0.21449869	0.21449869	0.00000000	0.00000000
22	0.23896173	0.23896173	0.00000000	0.00000000
23	0.24719000	0.24719000	0.00000000	0.00000000
24	0.24683192	0.24683192	0.00000000	0.00000000
25	0.24699967	0.24699967	0.00000000	0.00000000
26	0.24215261	0.24215261	0.00000000	0.00000000
27	0.22397761	0.22397761	0.00000000	0.00000000
28	0.21444357	0.21444357	0.00000000	0.00000000
29	0.19072923	0.19072923	0.00000000	0.00000000
30	0.18015259	0.18015259	0.00000000	0.00000000
31	0.15338248	0.15338248	0.00000000	0.00000000
32	0.14792344	0.14792344	0.00000000	0.00000000
33	0.13496044	0.13496044	0.00000000	0.00000000
34	0.12297123	0.12297123	0.00000000	0.00000000
35	0.10898075	0.10898075	0.00000000	0.00000000
36	0.10089172	0.10089172	0.00000000	0.00000000
37	0.08893427	0.08893427	0.00000000	0.00000000
38	0.07855315	0.07855315	0.00000000	0.00000000
39	0.05821529	0.05821529	0.00000000	0.00000000
40	0.04880839	0.04880839	0.00000000	0.00000000
41	0.03347546	0.03347546	0.00000000	0.00000000
42	0.02839703	0.02839703	0.00000000	0.00000000

FERTILITY AT TIME 1 # OF EIGEN VECTORS = 1

TOTAL FERTILITY IS .3904589E+01
AND SHOULD BE APPROXIMATELY .3903290E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00129828	0.00140531	-0.00010703	-0.08244199
15	0.00572302	0.00575715	-0.00003412	-0.00596266
16	0.01979159	0.01953654	0.00025505	0.01288658
17	0.04948444	0.04895830	0.00052614	0.01063240
18	0.09537538	0.09361352	0.00176186	0.01847291
19	0.14330014	0.14578512	-0.00248498	-0.01734107
20	0.18894981	0.19162375	-0.00267394	-0.01415159
21	0.22565961	0.22011069	0.00554892	0.02458976
22	0.24310473	0.24386823	-0.00076349	-0.00314060
23	0.25669047	0.25246446	0.00422601	0.01646345
24	0.25609517	0.25266041	0.00343476	0.01341204
25	0.24681112	0.25232224	-0.00551112	-0.02232930
26	0.23909436	0.24698466	-0.00789030	-0.03300076
27	0.23048897	0.22819784	0.00229113	0.00994029
28	0.21826567	0.21751879	0.00074687	0.00342186
29	0.20021409	0.19360818	0.00660591	0.03299425
30	0.17997128	0.18272372	-0.00275244	-0.01529378
31	0.15691973	0.15594687	0.00097286	0.00619975
32	0.14549466	0.14985646	-0.00436180	-0.02997914
33	0.13344094	0.13495853	-0.00151758	-0.01137271
34	0.12321618	0.12253334	0.00068284	0.00554177
35	0.10836263	0.10892225	-0.00055962	-0.00516436
36	0.09620220	0.10085592	-0.00465372	-0.04837432
37	0.08785682	0.08846605	-0.00060923	-0.00693437
38	0.07842358	0.07766487	0.00075870	0.00967442
39	0.06350840	0.05767250	0.00583591	0.09189187
40	0.04652638	0.04864341	-0.00211703	-0.04550172
41	0.03495642	0.03336035	0.00159607	0.04565884
42	0.02825436	0.02856915	-0.00031479	-0.01114136

FERTILITY AT TIME 2 # OF EIGEN VECTORS = 1

TOTAL FERTILITY IS .3870496E+01
AND SHOULD BE APPROXIMATELY .3961067E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00117014	0.00149320	-0.00032307	-0.27609305
15	0.00519080	0.00597477	-0.00078397	-0.15103101
16	0.01945517	0.01960050	-0.00014532	-0.00746955
17	0.04947561	0.04831008	0.00116553	0.02355767
18	0.09329335	0.09315475	0.00013860	0.00148563
19	0.14535907	0.14679237	-0.00143330	-0.00986039
20	0.19044164	0.19296599	-0.00252435	-0.01325524
21	0.22722658	0.22024370	0.00698288	0.03073091
22	0.24537513	0.24278718	0.00258795	0.01054692
23	0.25114462	0.25156189	-0.00041727	-0.00166147
24	0.25540319	0.25226464	0.00313855	0.01228861
25	0.24918853	0.25143189	-0.00224337	-0.00900268
26	0.23372428	0.24575408	-0.01202980	-0.05147007
27	0.22578314	0.22680139	-0.00101825	-0.00450985
28	0.21583170	0.21531084	0.00052086	0.00241328
29	0.19889706	0.19179020	0.00710685	0.03573132
30	0.18711031	0.18089631	0.00621399	0.03321032
31	0.15182707	0.15469680	-0.00286973	-0.01890128
32	0.14438361	0.14805333	-0.00366972	-0.02541647
33	0.13035266	0.13171202	-0.00135936	-0.01042831
34	0.11877861	0.11928285	-0.00050424	-0.00424518
35	0.10709330	0.10635720	0.00073610	0.00687344
36	0.09360418	0.09846466	-0.00486048	-0.05192590
37	0.08132279	0.08594649	-0.00462371	-0.05685622
38	0.07518463	0.07504124	0.00014340	0.00190725
39	0.06241668	0.05585350	0.00656318	0.10515107
40	0.05132358	0.04737646	0.00394712	0.07690662
41	0.03188586	0.03249007	-0.00060421	-0.01894915
42	0.02702209	0.02808806	-0.00106597	-0.03944811

FERTILITY AT TIME 3 # OF EIGEN VECTORS = 1

TOTAL FERTILITY IS .3819247E+01
AND SHOULD BE APPROXIMATELY .3825282E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00123441	0.00154271	-0.00030829	-0.24974904
15	0.00521766	0.00607428	-0.00085661	-0.16417585
16	0.01867178	0.01949475	-0.00082297	-0.04407582
17	0.04756180	0.04752269	0.00003911	0.00082226
18	0.09283691	0.09216609	0.00067082	0.00722574
19	0.14139398	0.14637924	-0.00498526	-0.03525795
20	0.19170998	0.19242257	-0.00071259	-0.00371703
21	0.22475908	0.21867972	0.00607936	0.02704833
22	0.24886512	0.24026362	0.00860150	0.03456289
23	0.25685458	0.24910325	0.00775134	0.03017792
24	0.24866620	0.25012940	-0.00146320	-0.00588420
25	0.24695142	0.24896638	-0.00201496	-0.00815935
26	0.23180944	0.24310251	-0.01129307	-0.04871706
27	0.21640417	0.22417009	-0.00776592	-0.03588621
28	0.20774408	0.21224866	-0.00450458	-0.02168332
29	0.19426251	0.18915521	0.00510730	0.02629072
30	0.18197114	0.17836807	0.00360307	0.01980023
31	0.16042574	0.15269059	0.00773515	0.04821637
32	0.14321692	0.14571091	-0.00249399	-0.01741411
33	0.12806254	0.12855513	-0.00049259	-0.00384648
34	0.11290227	0.11627638	-0.00337410	-0.02988517
35	0.10092717	0.10389858	-0.00297141	-0.02944112
36	0.09279758	0.09616321	-0.00336563	-0.03626849
37	0.08019087	0.08364803	-0.00345717	-0.04311171
38	0.07091423	0.07278036	-0.00186614	-0.02631539
39	0.05925637	0.05426150	0.00499487	0.08429249
40	0.04940889	0.04620982	0.00319906	0.06474669
41	0.03608194	0.03168416	0.00439778	0.12188306
42	0.02585527	0.02757868	-0.00172341	-0.06665600

FERTILITY AT TIME 4 # OF EIGEN VECTORS = I

TOTAL FERTILITY I= .3741084E+01
AND SHOULD BE APPROXIMATELY .3773257E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00107679	0.00157945	-0.00050266	-0.46681173
15	0.00522833	0.00612227	-0.00089393	-0.17097903
16	0.01729520	0.01924144	-0.00194623	-0.11253006
17	0.04298932	0.04640750	-0.00341818	-0.07951221
18	0.08733801	0.09051966	-0.00318164	-0.03642910
19	0.13916396	0.14486523	-0.00570127	-0.04096802
20	0.18669262	0.19042097	-0.00372836	-0.01997055
21	0.22520448	0.21549137	0.00971311	0.04313017
22	0.24307755	0.23600484	0.00707271	0.02909650
23	0.25455078	0.24485050	0.00970027	0.03810741
24	0.25278937	0.24616818	0.00662118	0.02619250
25	0.23839830	0.24469172	-0.00629342	-0.02639875
26	0.23225667	0.23869725	-0.00644058	-0.02773043
27	0.21512669	0.21992117	-0.00479448	-0.02228678
28	0.20268467	0.20769300	-0.00500833	-0.02470996
29	0.18551114	0.18519317	0.00031797	0.00171401
30	0.17498937	0.17459536	0.00039402	0.00225165
31	0.15677843	0.14959398	0.00718445	0.04582553
32	0.14826601	0.14231087	0.00595513	0.04016519
33	0.12414697	0.12454071	-0.00039374	-0.00317158
34	0.11313044	0.11254799	0.00058245	0.00514845
35	0.09699235	0.10078278	-0.00379043	-0.03907966
36	0.08901913	0.09323992	-0.00422079	-0.04741436
37	0.07703586	0.08082360	-0.00378774	-0.04916848
38	0.06648443	0.07009200	-0.00360757	-0.05426183
39	0.05562055	0.05235367	0.00326688	0.05873507
40	0.04636920	0.04475812	0.00161107	0.03474443
41	0.03451343	0.03068440	0.00382903	0.11094313
42	0.02772211	0.02689331	0.00082880	0.02989685

FERTILITY AT TIME 5 # OF EIGEN VECTORS = 1

TOTAL FERTILITY IS .3656251E+01
AND SHOULD BE APPROXIMATELY .3694721E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00092582	0.00160882	-0.00068300	-0.73772324
15	0.00513696	0.00614442	-0.00100746	-0.19612020
16	0.01924041	0.01893885	0.00030156	0.01567328
17	0.04310133	0.04522175	-0.00212042	-0.04919614
18	0.08086856	0.08869438	-0.00782582	-0.09677208
19	0.13294175	0.14296965	-0.01002790	-0.07543076
20	0.18127903	0.18790883	-0.00662980	-0.03657235
21	0.21588740	0.21179200	0.00409540	0.01897007
22	0.24105021	0.23126106	0.00978916	0.04061044
23	0.24883770	0.24009143	0.00874627	0.03514848
24	0.24811439	0.24166368	0.00645071	0.02599894
25	0.23644563	0.23989849	-0.00345286	-0.01460319
26	0.22497033	0.23380669	-0.00883636	-0.03927789
27	0.21371095	0.21523311	-0.00152216	-0.00712250
28	0.19804178	0.20277852	-0.00473674	-0.02391789
29	0.17998952	0.18090970	-0.00092018	-0.00511240
30	0.16950978	0.17052720	-0.00101743	-0.00600217
31	0.15189376	0.14621447	0.00567929	0.03738990
32	0.14392076	0.13864935	0.00527141	0.03662718
33	0.12803155	0.12040738	0.00762417	0.05954915
34	0.10964058	0.10876134	0.00087924	0.00801931
35	0.09773993	0.09759317	0.00014677	0.00150162
36	0.08751076	0.09023879	-0.00272803	-0.03117364
37	0.07417380	0.07795750	-0.00378370	-0.05101125
38	0.06604665	0.06740361	-0.00135696	-0.02054542
39	0.05399717	0.05044318	0.00355399	0.06581802
40	0.04291559	0.04328299	-0.00036740	-0.00856093
41	0.03185812	0.02967000	0.00218812	0.06868331
42	0.02541394	0.02618104	-0.00076710	-0.03018406

FERTILITY AT TIME 6 # OF EIGEN VECTORS = 1

TOTAL FERTILITY IS .3488019E+01
AND SHOULD BE APPROXIMATELY .3609718E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00099873	0.00148556	-0.00048682	-0.48743928
15	0.00448859	0.00574350	-0.00125491	-0.27957698
16	0.01688119	0.01796711	-0.00108592	-0.06432720
17	0.04529248	0.04323988	0.00205260	0.04531878
18	0.07812692	0.08444368	-0.00631675	-0.08085245
19	0.11990156	0.13536271	-0.01546115	-0.12894866
20	0.16647631	0.17792142	-0.01144510	-0.06874914
21	0.20578957	0.20116442	0.00462515	0.02247515
22	0.22509845	0.22016661	0.00493183	0.02190968
23	0.23646747	0.22846310	0.00800438	0.03384980
24	0.23636354	0.22974308	0.00662047	0.02800967
25	0.23089246	0.22829587	0.00259659	0.01124589
26	0.21610183	0.22266814	-0.00656630	-0.03038523
27	0.20254016	0.20510778	-0.00256762	-0.01267708
28	0.19179127	0.19358987	-0.00179860	-0.00937791
29	0.17387861	0.17267045	0.00120816	0.00694832
30	0.16178031	0.16273450	-0.00095419	-0.00589806
31	0.14051905	0.13951969	0.00099936	0.00711193
32	0.13978850	0.13255660	0.00723190	0.05173460
33	0.12298517	0.11584888	0.00713629	0.05802564
34	0.11061313	0.10467854	0.00593458	0.05365172
35	0.09135426	0.09377645	-0.00242219	-0.02651426
36	0.08337070	0.08673823	-0.00336753	-0.04039230
37	0.07161033	0.07513855	-0.00352822	-0.04926967
38	0.06250284	0.06511062	-0.00260777	-0.04172249
39	0.05182442	0.04866962	0.00315480	0.06087477
40	0.04145239	0.04162066	-0.00016827	-0.00405934
41	0.03020637	0.02854827	0.00165810	0.05489243
42	0.02479370	0.02504527	-0.00025157	-0.01014658

FERTILITY AT TIME 7 # OF EIGEN VECTORS = 1

TOTAL FERTILITY IS .3132326E+01
AND SHOULD BE APPROXIMATELY .3444536E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00085609	0.00130346	-0.00044737	-0.52257266
15	0.00437968	0.00508315	-0.00070348	-0.16062297
16	0.01640345	0.01607104	0.00033241	0.02026485
17	0.04146224	0.03889279	0.00256945	0.06197085
18	0.08203828	0.07572649	0.00631179	0.07693712
19	0.11364174	0.12091147	-0.00726973	-0.06397059
20	0.14346756	0.15893092	-0.01546337	-0.10778301
21	0.17989624	0.18008962	-0.00019337	-0.00107492
22	0.20407305	0.19742689	0.00664616	0.03256756
23	0.21056847	0.20479923	0.00576923	0.02739836
24	0.21155682	0.20580868	0.00574815	0.02717070
25	0.20541875	0.20465623	0.00076253	0.00371207
26	0.19443693	0.19971775	-0.00528082	-0.02715955
27	0.18335946	0.18404707	-0.00068761	-0.00375007
28	0.16914317	0.17393605	-0.00479288	-0.02833623
29	0.15578279	0.15511473	0.00066806	0.00428838
30	0.14525011	0.14617640	-0.00092629	-0.00637721
31	0.12402697	0.12530591	-0.00127894	-0.01031182
32	0.12086307	0.11921110	0.00165197	0.01366811
33	0.10987862	0.10464726	0.00523137	0.04761041
34	0.09794092	0.09459245	0.00334838	0.03418775
35	0.08710251	0.08464451	0.00245800	0.02821960
36	0.07407259	0.07830621	-0.00423362	-0.05715498
37	0.06510293	0.06796232	-0.00285939	-0.04392108
38	0.05757801	0.05898727	-0.00140926	-0.02447563
39	0.04617012	0.04405737	0.00211275	0.04576010
40	0.03637008	0.03758866	-0.00121858	-0.03350489
41	0.02689594	0.02579275	0.00110319	0.04101714
42	0.02159555	0.02253867	-0.00094312	-0.04367186

FERTILITY AT TIME 8 # OF EIGEN VECTORS = 1

TOTAL FERTILITY IS .2786129E+01
AND SHOULD BE APPROXIMATELY .3093853E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00097835	0.00121449	-0.00023614	-0.24136974
15	0.00464076	0.00465822	-0.00001745	-0.00376041
16	0.01652825	0.01440913	0.00211912	0.12821196
17	0.03882417	0.03448083	0.00434334	0.11187198
18	0.07227470	0.06754938	0.00472532	0.06538003
19	0.10694636	0.10872796	-0.00178160	-0.01665881
20	0.13264469	0.14290166	-0.01025697	-0.07732661
21	0.16135440	0.16119885	0.00015555	0.00096402
22	0.17507496	0.17612547	-0.00105051	-0.00600036
23	0.18943761	0.18283761	0.00660000	0.03483998
24	0.19392351	0.18397909	0.00994442	0.05128014
25	0.18263169	0.18268236	-0.00005067	-0.00027745
26	0.17602961	0.17809098	-0.00206138	-0.01171039
27	0.15866943	0.16396371	-0.00529428	-0.03336672
28	0.15264086	0.15454055	-0.00189969	-0.01244546
29	0.14438192	0.13789828	0.00648364	0.04490618
30	0.11581125	0.12992773	-0.01411647	-0.12189207
31	0.11722756	0.11146994	0.00575761	0.04911486
32	0.10141209	0.10567644	-0.00426435	-0.04204971
33	0.09406312	0.09197425	0.00208887	0.02220711
34	0.08783138	0.08308100	0.00475038	0.05408523
35	0.07135547	0.07451452	-0.00315905	-0.04427204
36	0.06859843	0.06889590	-0.00029747	-0.00433642
37	0.05513250	0.05957061	-0.00443810	-0.08049881
38	0.04905642	0.05152811	-0.00247168	-0.05038453
39	0.04239055	0.03856708	0.00382347	0.09019617
40	0.02875401	0.03303956	-0.00428555	-0.14904195
41	0.02639477	0.02266869	0.00372608	0.14116720
42	0.01736211	0.01995681	-0.00259470	-0.14944604

FERTILITY AT TIME 9 # OF EIGEN VECTORS = 1

TOTAL FERTILITY IS .2583853E+01
AND SHOULD BE APPROXIMATELY .2750869E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00096136	0.00132651	-0.00036515	-0.37983095
15	0.00468354	0.00481381	-0.00013027	-0.02781398
16	0.01572758	0.01374137	0.00198621	0.12628819
17	0.03666979	0.03157380	0.00509599	0.13896969
18	0.06925468	0.06334763	0.00590706	0.08529470
19	0.10516659	0.10507383	0.00009276	0.00088204
20	0.14373461	0.13793455	0.00580007	0.04035261
21	0.15035173	0.15295052	-0.00259879	-0.01728472
22	0.15949674	0.16510252	-0.00560578	-0.03514667
23	0.17472132	0.17203059	0.00269073	0.01540013
24	0.18329417	0.17382767	0.00946650	0.05164649
25	0.17502772	0.17158820	0.00343952	0.01965129
26	0.16373228	0.16670770	-0.00297542	-0.01817247
27	0.15131370	0.15283117	-0.00151747	-0.01002864
28	0.14218813	0.14258657	-0.00039843	-0.00280216
29	0.12729471	0.12779720	-0.00050249	-0.00394743
30	0.11228548	0.12001484	-0.00772936	-0.06883671
31	0.09969926	0.10359764	-0.00389838	-0.03910139
32	0.09064334	0.09613012	-0.00548678	-0.06053155
33	0.08223678	0.08131976	0.00091703	0.01115106
34	0.07406745	0.07355893	0.00050852	0.00686559
35	0.06735227	0.06653961	0.00081266	0.01206574
36	0.05858556	0.06121448	-0.00262893	-0.04487330
37	0.05243425	0.05218669	0.00024757	0.00472144
38	0.04235610	0.04458038	-0.00222428	-0.05251393
39	0.03446034	0.03381676	0.00064358	0.01867599
40	0.02687526	0.02925678	-0.00238151	-0.08861362
41	0.02035015	0.02018162	0.00016854	0.00828179
42	0.01498955	0.01822216	-0.00323261	-0.21565727

FERTILITY AT TIME 10 # OF EIGEN VECTORS = 1

TOTAL FERTILITY IS .2447947E+01
AND SHOULD BE APPROXIMATELY .2547553E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00110640	0.00154633	-0.00043993	-0.39762310
15	0.00505028	0.00522378	-0.00017350	-0.03435457
16	0.01581958	0.01347112	0.00234845	0.14845225
17	0.03661822	0.02937417	0.00724405	0.19782635
18	0.06552479	0.06101250	0.00451229	0.06886386
19	0.09698319	0.10531394	-0.00833075	-0.08589890
20	0.12648107	0.13780042	-0.01131935	-0.08949443
21	0.15646612	0.14920286	0.00726326	0.04642066
22	0.15494981	0.15865210	-0.00370229	-0.02389348
23	0.15846735	0.16638137	-0.00791402	-0.04994103
24	0.17153873	0.16883568	0.00270305	0.01575766
25	0.17163132	0.16522184	0.00640949	0.03734451
26	0.16252214	0.15982417	0.00269798	0.01660066
27	0.14801953	0.14544860	0.00257093	0.01736886
28	0.13729519	0.13402651	0.00326868	0.02380771
29	0.12101533	0.12104845	-0.00003312	-0.00027367
30	0.11115735	0.11307087	-0.00191352	-0.01721455
31	0.09190372	0.09812602	-0.00622230	-0.06770453
32	0.08698781	0.08753179	-0.00054398	-0.00625347
33	0.07469166	0.07181980	0.00287186	0.03844948
34	0.06716910	0.06560743	0.00156168	0.02324990
35	0.05829366	0.05991189	-0.00161824	-0.02776010
36	0.05198787	0.05446217	-0.00247429	-0.04759367
37	0.04472293	0.04552183	-0.00079890	-0.01786327
38	0.03911561	0.03837815	0.00073746	0.01885331
39	0.03090153	0.02985602	0.00104552	0.03383387
40	0.02455662	0.02603106	-0.00147444	-0.06004250
41	0.01687780	0.01821948	-0.00134168	-0.07949378
42	0.01383164	0.01702628	-0.00319464	-0.23096634

FERTILITY AT TIME 11 # OF EIGEN VECTORS = 1

TOTAL FERTILITY IS .2401225E+01
AND SHOULD BE APPROXIMATELY .2408902E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00113989	0.00169167	-0.00055178	-0.48406455
15	0.00564123	0.00548152	0.00015971	0.02831146
16	0.01666629	0.01344560	0.00322069	0.19324578
17	0.03745540	0.02851871	0.00893669	0.23859560
18	0.06401094	0.06043058	0.00358036	0.05593358
19	0.09137526	0.10650626	-0.01513100	-0.16559191
20	0.11609687	0.13903981	-0.02294294	-0.19761895
21	0.13969682	0.14858319	-0.00888637	-0.06361186
22	0.16619485	0.15682363	0.00937121	0.05638691
23	0.15936474	0.16507567	-0.00571093	-0.03583558
24	0.16253296	0.16783269	-0.00529973	-0.03260711
25	0.16900445	0.16343966	0.00556479	0.03292691
26	0.16630396	0.15767149	0.00863248	0.05190782
27	0.14816102	0.14287631	0.00528471	0.03566871
28	0.13953918	0.13092934	0.00860984	0.06170196
29	0.12269780	0.11864618	0.00405163	0.03302118
30	0.11128392	0.11067429	0.00060963	0.00547817
31	0.09286567	0.09575269	-0.00288703	-0.03108819
32	0.08265449	0.08371538	-0.00106089	-0.01283522
33	0.07219745	0.06768220	0.00451525	0.06254032
34	0.06594299	0.06235249	0.00359050	0.05444854
35	0.05665423	0.05712513	-0.00047090	-0.00831185
36	0.04849844	0.05152940	-0.00303096	-0.06249613
37	0.04242533	0.04259115	-0.00016582	-0.00390840
38	0.03644705	0.03576644	0.00068061	0.01867402
39	0.02976103	0.02820164	0.00155939	0.05239710
40	0.02230338	0.02474153	-0.00243815	-0.10931735
41	0.01599049	0.01743641	-0.00144592	-0.09042394
42	0.01187762	0.01666427	-0.00478666	-0.40299803

FERTILITY AT TIME 12 # OF EIGEN VECTORS = 1

TOTAL FERTILITY IS .2330801E+01
AND SHOULD BE APPROXIMATELY .2360408E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00125859	0.00182752	-0.00056893	-0.45203562
15	0.00583547	0.00568127	0.00015420	0.02642516
16	0.01935108	0.01327060	0.00608048	0.31421926
17	0.04043189	0.02738440	0.01304749	0.32270296
18	0.06371963	0.05926987	0.00444976	0.06983343
19	0.08859126	0.10665652	-0.01806526	-0.20391698
20	0.10817984	0.13884275	-0.03066292	-0.28344393
21	0.12823853	0.14638255	-0.01814402	-0.14148649
22	0.15044029	0.15341402	-0.00297373	-0.01976684
23	0.17189825	0.16214826	0.00974999	0.05671956
24	0.16380504	0.16510670	-0.00130166	-0.00794640
25	0.15823800	0.15995522	-0.00171722	-0.01085212
26	0.16029303	0.15387381	0.00641921	0.04004674
27	0.15297193	0.13878995	0.01418198	0.09270968
28	0.13907795	0.12654346	0.01253449	0.09012567
29	0.12261123	0.11508651	0.00752472	0.06137060
30	0.10928098	0.10714468	0.00213631	0.01954876
31	0.09088777	0.09220477	-0.00131700	-0.01449037
32	0.08155627	0.07889625	0.00266002	0.03261575
33	0.06739788	0.06301654	0.00438135	0.06500718
34	0.05888368	0.05866652	0.00021717	0.00368804
35	0.05354121	0.05383527	-0.00029406	-0.00549224
36	0.04372830	0.04811961	-0.00439131	-0.10042250
37	0.03768262	0.03933224	-0.00164962	-0.04377676
38	0.03408004	0.03296050	0.00111954	0.03285045
39	0.02622801	0.02638055	-0.00015255	-0.00581614
40	0.02027882	0.02327176	-0.00299294	-0.14758944
41	0.01444794	0.01654473	-0.00209679	-0.14512754
42	0.01127439	0.01619399	-0.00491960	-0.43635224

FERTILITY AT TIME 12 # OF EIGEN VECTORS = 2

TOTAL FERTILITY IS .2331060E+01
AND SHOULD BE APPROXIMATELY .2360408E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00125859	0.00173154	-0.00047295	-0.37578140
15	0.00583547	0.00570417	0.00013130	0.02250067
16	0.01935108	0.01349502	0.00585605	0.30262155
17	0.04043189	0.02623808	0.01419380	0.35105463
18	0.06371963	0.05526083	0.00845880	0.13275023
19	0.08859126	0.10131907	-0.01272782	-0.14366900
20	0.10817984	0.13614706	-0.02796722	-0.25852526
21	0.12823853	0.14697715	-0.01873862	-0.14612314
22	0.15044029	0.15547241	-0.00503211	-0.03344925
23	0.17189825	0.16333944	0.00855882	0.04979002
24	0.16380504	0.16536375	-0.00155871	-0.00951566
25	0.15823800	0.16012835	-0.00189035	-0.01194626
26	0.16029303	0.15490119	0.00539183	0.03363736
27	0.15297193	0.14159236	0.01137956	0.07438988
28	0.13907795	0.12823008	0.01084787	0.07799848
29	0.12261123	0.11457522	0.00803601	0.06554058
30	0.10928098	0.10697763	0.00230336	0.02107739
31	0.09088777	0.09243088	-0.00154311	-0.01697819
32	0.08155627	0.08084573	0.00071054	0.00871227
33	0.06739798	0.06579787	0.00160001	0.02373979
34	0.05888368	0.06007903	-0.00119535	-0.02030021
35	0.05354121	0.05432548	-0.00078427	-0.01464803
36	0.04372830	0.04757973	-0.00385142	-0.08807621
37	0.03768262	0.03876712	-0.00108450	-0.02877986
38	0.03408004	0.03283997	0.00124007	0.03638700
39	0.02622801	0.02627659	-0.00004858	-0.00185218
40	0.02027882	0.02280535	-0.00252653	-0.12458968
41	0.01444794	0.01607364	-0.00162570	-0.11252114
42	0.01127439	0.01578524	-0.00451085	-0.40009720

FERTILITY AT TIME 12 # OF EIGEN VECTORS = 3

TOTAL FERTILITY IS .2329979E+01
AND SHOULD BE APPROXIMATELY .2360408E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00125859	0.00187647	-0.00061788	-0.49092830
15	0.00583547	0.00683936	-0.00100389	-0.17203229
16	0.01935108	0.01581133	0.00353975	0.18292248
17	0.04043199	0.02832069	0.01211119	0.29954556
18	0.06371963	0.05834006	0.00537957	0.08442560
19	0.08859126	0.10212408	-0.01353282	-0.15275573
20	0.10817994	0.13019791	-0.02201808	-0.20353218
21	0.12823853	0.14119941	-0.01296088	-0.10106851
22	0.15044029	0.15434809	-0.00390779	-0.02597571
23	0.17189825	0.16379825	0.00810000	0.04712090
24	0.16380504	0.16582428	-0.00201924	-0.01232712
25	0.15823800	0.15927430	-0.00103630	-0.00654899
26	0.16029303	0.15464091	0.00565212	0.03526118
27	0.15297193	0.14376439	0.00920754	0.06019102
28	0.13907795	0.13153125	0.00754670	0.05426239
29	0.12261123	0.11502809	0.00758314	0.06184706
30	0.10928098	0.10408517	0.00519581	0.04754543
31	0.09088777	0.08956199	0.00132578	0.01458706
32	0.08155627	0.07987190	0.00168437	0.02065286
33	0.06739798	0.06741393	-0.00001604	-0.00023803
34	0.05888368	0.06228542	-0.00340174	-0.05777045
35	0.05354121	0.05538935	-0.00184814	-0.03451802
36	0.04372830	0.04732649	-0.00359819	-0.08228508
37	0.03768262	0.03857224	-0.00088962	-0.02360826
38	0.03408004	0.03430067	-0.00022063	-0.00647379
39	0.02622801	0.02690077	-0.00067277	-0.02565069
40	0.02027892	0.02153227	-0.00125345	-0.06181100
41	0.01444794	0.01489360	-0.00044565	-0.03084557
42	0.01127439	0.01492659	-0.00365220	-0.32393803

FERTILITY AT TIME 12 # OF EIGEN VECTORS = 4

TOTAL FERTILITY IS .2328504E+01
AND SHOULD BE APPROXIMATELY .2360408E+01

AGE	ACTUAL	SIMULATED	ERROR	FRAC ERROR
14	0.00125859	0.00194224	-0.00068365	-0.54318493
15	0.00583547	0.00645364	-0.00061817	-0.10593304
16	0.01935108	0.01575472	0.00359636	0.18584799
17	0.04043189	0.03063904	0.00979284	0.24220594
18	0.06371963	0.06039005	0.00332958	0.05225355
19	0.08859126	0.09766204	-0.00907079	-0.10238919
20	0.10817994	0.12598271	-0.01780287	-0.16456736
21	0.12823853	0.14295217	-0.01471364	-0.11473649
22	0.15044029	0.15404540	-0.00360511	-0.02396374
23	0.17189825	0.16059925	0.01129900	0.06573076
24	0.16380504	0.16596149	-0.00215646	-0.01316478
25	0.15823800	0.16271799	-0.00447999	-0.02831173
26	0.16029303	0.15810055	0.00219247	0.01367791
27	0.15297193	0.14554351	0.00742842	0.04856066
28	0.13907795	0.13147878	0.00759917	0.05463967
29	0.12261123	0.11455013	0.00806110	0.06574520
30	0.10928098	0.10197870	0.00730228	0.06682114
31	0.09088777	0.08767531	0.00321246	0.03534540
32	0.08155627	0.08130037	0.00025589	0.00313760
33	0.06739798	0.06945953	-0.00206165	-0.03058918
34	0.05888368	0.06284414	-0.00396046	-0.06725900
35	0.05354121	0.05414639	-0.00060517	-0.01130296
36	0.04372830	0.04675074	-0.00302244	-0.06911867
37	0.03768262	0.03978056	-0.00209795	-0.05567413
38	0.03408004	0.03448349	-0.00040344	-0.01183814
39	0.02622801	0.02581934	0.00040867	0.01558150
40	0.02027882	0.02047632	-0.00019750	-0.00973943
41	0.01444794	0.01427472	0.00017322	0.01198914
42	0.01127439	0.01474062	-0.00346623	-0.30744315

2.4 Fertility Profiles

For an overall perspective on the change in fertility over time, a modified version of the SYMVU program for profile plotting was used with both actual data, and with simulated values of the fertility.

The program SYMVU as modified at Queen's has been further modified for this project to give plots with time scale increasing from front to back, and age scale increasing from left to right. Programme 2.5 is a listing of the program using SYMVU to plot fertility from actual data for ages 14-42 over the years 1958-1970; Program 2.6 is a listing to plot fertility using simulated values over the same period.

```
$ JOB    ACCT-NUM,VERNER,TIME=60
C        ***********************************************************************
C
C        PROGRAM 2.5 : PLOT
C
C        ***********************************************************************
```

```
3JOB PLOT; CHARGE BERGHOUT; CLASS 3;
USER 080002/VHJ;
BEGIN
COMPILE MY/DATA FORTRAN LIBRARY;
DATA
```

```
3DATA FILE5
```

```
3REMOVE MY/DATA FROM PACK;
REMOVE MAIN FROM PACK;
END JOB
```

```
C       ***********************************************************************
C
C       PROGRAM : 2.5
C       TITLE : PLOT
C       PURPOSE : PLOT FERTILITY DATA
C       DATA : : PROGRAM 2.7 : SURFACE
C       OUTPUT : SURFACE PLOT
C
C       ***********************************************************************
$SET $
$SET SEPARATE
        SUBROUTINE DATA(X,IR,IC,IF)
C       ***********************************************************************
C
C       THIS ROUTINE IS USED TO CREATE A PLOT OF FA ON THE
C       CALCOMP PLOTTER FOR AGES 14 TO 42 AND YEARS 1958 TO 1970
C
C       ***********************************************************************
        REAL X(130)
$SET OWN
        REAL FBAR(51,17),FT(14)
$POP OWN
        IF (IR.NE.1) GO TO 10
        READ(5,100)((FBAR(I,J),I=1,51),J=1,17)
  100   FORMAT(4E17.7)
        READ(5,101)(FT(I),I=1,14)
  101   FORMAT(8F10.8)
   10   CONTINUE
        L=IR+2
        XX=FT(IR)/3.85865
        DO 20 I=1,IC
           K=I+14
           X(I)=FBAR(K,L)*XX
   20      CONTINUE
        RETURN
        END

        END
```

```
C       ********************************************************************
C
C       DATA FOR
C       PROGRAM 2.5 : PLOT
C
C       ********************************************************************
FERTILITY FROM VITAL STATISTICS DATA - 1958 TO 1970
  13  29   2   2        0                                     1 1      3        0
 30.0330.0  7.0          4.0                            20.0
     0.0000000E 00     0.0000000E 00     0.0000000E 00     0.0000000E 00
     0.0000000E 00     0.0000000E 00     0.0000000E 00     0.0000000E 00
     0.0000000E 00     0.0000000E 00     0.0000000E 00     0.0000000E 00
     0.7429816E-05     0.1634927E-03     0.1006141E-02     0.4942421E-02
     0.1726509E-01     0.4614777E-01     0.8778054E-01     0.1287903E 00
     0.1733888E 00     0.2095979E 00     0.2266342E 00     0.2462188E 00
     0.2535967E 00     0.2474782E 00     0.2369086E 00     0.2169245E 00
     0.2116948E 00     0.1892552E 00     0.1809314E 00     0.1552396E 00
     0.1511665E 00     0.1356698E 00     0.1278709E 00     0.1152725E 00
     0.1045548E 00     0.8492929E-01     0.7790750E-01     0.6245709E-01
     0.5043102E-01     0.3700228E-01     0.3078149E-01     0.2018753E-01
     0.1298582E-01     0.7232107E-02     0.3469116E-02     0.1403072E-02
     0.7913960E-03     0.4042918E-03     0.1762484E-03     0.0000000E 00
     0.0000000E 00     0.0000000E 00     0.0000000E 00     0.0000000E 00
     0.0000000E 00     0.0000000E 00     0.0000000E 00     0.0000000E 00
     0.0000000E 00     0.0000000E 00     0.0000000E 00     0.2033752E-04
     0.2206279E-03     0.1024708E-02     0.5051389E-02     0.1962170E-01
     0.4878248E-01     0.9214234E-01     0.1379995E 00     0.1777333E 00
     0.2134188E 00     0.2316796E 00     0.2428612E 00     0.2487651E 00
     0.2482604E 00     0.2353446E 00     0.2238681E 00     0.2067142E 00
     0.1883027E 00     0.1754966E 00     0.1528666E 00     0.1484947E 00
     0.1346632E 00     0.1217132E 00     0.1124704E 00     0.1036606E 00
     0.9143513E-01     0.7362139E-01     0.6127410E-01     0.4957513E-01
     0.3556923E-01     0.3062474E-01     0.2014163E-01     0.1207764E-01
     0.6471954E-02     0.3641624E-02     0.1707581E-02     0.7939595E-03
     0.2838371E-03     0.2689431E-03     0.0000000E 00     0.0000000E 00
     0.0000000E 00     0.0000000E 00     0.0000000E 00     0.0000000E 00
     0.0000000E 00     0.0000000E 00     0.0000000E 00     0.0000000E 00
     0.0000000E 00     0.0000000E 00     0.2588560E-04     0.1491891E-03
     0.1273050E-02     0.5347177E-02     0.1883900E-01     0.4806599E-01
     0.9110671E-01     0.1400929E 00     0.1840963E 00     0.2128937E 00
     0.2371737E 00     0.2453404E 00     0.2449850E 00     0.2451515E 00
     0.2403407E 00     0.2223017E 00     0.2128390E 00     0.1893021E 00
     0.1788046E 00     0.1522348E 00     0.1468166E 00     0.1339506E 00
     0.1220511E 00     0.1081653E 00     0.1001368E 00     0.8826882E-01
     0.7796538E-01     0.5777969E-01     0.4844318E-01     0.3322498E-01
     0.2818455E-01     0.1854310E-01     0.1211527E-01     0.6617039E-02
     0.3324505E-02     0.1627151E-02     0.6927683E-03     0.2283992E-03
     0.1625607E-03     0.0000000E 00     0.0000000E 00     0.0000000E 00
     0.0000000E 00     0.0000000E 00     0.0000000E 00     0.0000000E 00
     0.0000000E 00     0.0000000E 00     0.0000000E 00     0.0000000E 00
```

0.5848482E-05	0.1815752E-04	0.1524006E-03	0.1269283E-02
0.5595200E-02	0.1934955E-01	0.4837923E-01	0.9324521E-01
0.1400996E 00	0.1847297E 00	0.2206196E 00	0.2376751E 00
0.2509574E 00	0.2503754E 00	0.2412987E 00	0.2337543E 00
0.2253411E 00	0.2133908E 00	0.1957424E 00	0.1759517E 00
0.1534150E 00	0.1422451E 00	0.1304606E 00	0.1204642E 00
0.1059424E 00	0.9405357E-01	0.8589458E-01	0.7667202E-01
0.6208997E-01	0.4548723E-01	0.3417568E-01	0.2762331E-01
0.1803264E-01	0.1140561E-01	0.7085640E-02	0.2832796E-02
0.1616091E-02	0.8214479E-03	0.2615624E-03	0.1299269E-03
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.2325392E-04	0.1646105E-03	0.1154704E-02	0.5122337E-02
0.1919859E-01	0.4882310E-01	0.9206295E-01	0.1434420E 00
0.1879300E 00	0.2242298E 00	0.2421390E 00	0.2478324E 00
0.2520348E 00	0.2459021E 00	0.2306418E 00	0.2228054E 00
0.2129852E 00	0.1962739E 00	0.1846426E 00	0.1498247E 00
0.1424794E 00	0.1286335E 00	0.1172121E 00	0.1056809E 00
0.9236968E-01	0.8025026E-01	0.7419306E-01	0.6159350E-01
0.5064670E-01	0.3146533E-01	0.2666571E-01	0.1807094E-01
0.1073283E-01	0.6116446E-02	0.3089603E-02	0.1289498E-02
0.5448801E-03	0.2109780E-03	0.1840822E-03	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.3936497E-04
0.1471253E-03	0.1234930E-02	0.5219840E-02	0.1867957E-01
0.4758165E-01	0.9287566E-01	0.1414530E 00	0.1917900E 00
0.2248529E 00	0.2489690E 00	0.2569618E 00	0.2487700E 00
0.2470545E 00	0.2319062E 00	0.2164945E 00	0.2078308E 00
0.1943436E 00	0.1820471E 00	0.1604927E 00	0.1432767E 00
0.1281160E 00	0.1129494E 00	0.1009693E 00	0.9283632E-01
0.8022434E-01	0.7094383E-01	0.5928110E-01	0.4942951E-01
0.3609700E-01	0.2586606E-01	0.1796386E-01	0.1039849E-01
0.5931694E-02	0.2902867E-02	0.1597295E-02	0.7027730E-03
0.2392092E-03	0.1916328E-03	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.1611304E-03
0.1100139E-02	0.5341694E-02	0.1767021E-01	0.4392144E-01
0.8923173E-01	0.1421814E 00	0.1907406E 00	0.2300875E 00
0.2483481E 00	0.2600701E 00	0.2582705E 00	0.2435674E 00
0.2372926E 00	0.2197912E 00	0.2070794E 00	0.1895335E 00
0.1787836E 00	0.1601778E 00	0.1514808E 00	0.1268388E 00
0.1155834E 00	0.9909540E-01	0.9094930E-01	0.7870620E-01
0.6792599E-01	0.5682655E-01	0.4737460E-01	0.3526177E-01
0.2832320E-01	0.1706897E-01	0.9436630E-02	0.5381081E-02
0.2923402E-02	0.1375800E-02	0.4280740E-03	0.2046375E-03
0.1230038E-03	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00

0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.2249860E-04	0.1204436E-03	0.9681717E-03
0.5371928E-02	0.2012049E-01	0.4507285E-01	0.8456761E-01
0.1390227E 00	0.1895710E 00	0.2257624E 00	0.2520762E 00
0.2602199E 00	0.2594635E 00	0.2472610E 00	0.2352608E 00
0.2234864E 00	0.2071005E 00	0.1882225E 00	0.1772634E 00
0.1588416E 00	0.1505039E 00	0.1338879E 00	0.1146557E 00
0.1022107E 00	0.9151363E-01	0.7756662E-01	0.6906772E-01
0.5646708E-01	0.4487861E-01	0.3331536E-01	0.2657641E-01
0.1820700E-01	0.9477235E-02	0.5660690E-02	0.2936016E-02
0.1072021E-02	0.5649622E-03	0.1822288E-03	0.1219036E-03
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.1736314E-04	0.1416877E-03	0.1094503E-02	0.4919004E-02
0.1849992E-01	0.4963557E-01	0.8561850E-01	0.1313989E 00
0.1824397E 00	0.2255227E 00	0.2466831E 00	0.2591423E 00
0.2590284E 00	0.2530327E 00	0.2368238E 00	0.2219617E 00
0.2101821E 00	0.1905518E 00	0.1772934E 00	0.1539934E 00
0.1531928E 00	0.1347782E 00	0.1212198E 00	0.1001142E 00
0.9136510E-01	0.7847703E-01	0.6849623E-01	0.5679385E-01
0.4542725E-01	0.3310285E-01	0.2717116E-01	0.1823463E-01
0.1100488E-01	0.6262664E-02	0.3032315E-02	0.1214716E-02
0.4865748E-03	0.1983496E-03	0.1126833E-03	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.6215397E-05	0.1900049E-04
0.2125569E-03	0.1044521E-02	0.5343676E-02	0.2001397E-01
0.5058838E-01	0.1000955E 00	0.1386551E 00	0.1750458E 00
0.2194927E 00	0.2489910E 00	0.2569161E 00	0.2581220E 00
0.2506329E 00	0.2372339E 00	0.2237182E 00	0.2063728E 00
0.1900717E 00	0.1772207E 00	0.1513262E 00	0.1474659E 00
0.1340637E 00	0.1194983E 00	0.1062744E 00	0.9037650E-01
0.7943255E-01	0.7025135E-01	0.5633250E-01	0.4437540E-01
0.3281593E-01	0.2634889E-01	0.1717417E-01	0.1114297E-01
0.6952133E-02	0.2916923E-02	0.1267667E-02	0.4754490E-03
0.1937536E-03	0.1845764E-03	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.2060735E-04	0.2163455E-03
0.1342522E-02	0.6368209E-02	0.2268060E-01	0.5327580E-01
0.9917772E-01	0.1467553E 00	0.1820194E 00	0.2214158E 00
0.2402436E 00	0.2599525E 00	0.2661082E 00	0.2506132E 00
0.2415536E 00	0.2177314E 00	0.2094588E 00	0.1981256E 00
0.1589200E 00	0.1608635E 00	0.1391610E 00	0.1290765E 00
0.1205251E 00	0.9791631E-01	0.9413302E-01	0.7565463E-01
0.6731683E-01	0.5816970E-01	0.3945719E-01	0.3621977E-01
0.2382486E-01	0.1810371E-01	0.1119237E-01	0.5598128E-02
0.3102446E-02	0.1532112E-02	0.4638946E-03	0.2444400E-03
0.1360976E-03	0.0000000E 00	0.0000000E 00	0.0000000E 00

0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.2193426E-04	0.2478703E-03	0.1424491E-02
0.6939832E-02	0.2330431E-01	0.5433539E-01	0.1026180E 00
0.1558304E 00	0.2129785E 00	0.2227834E 00	0.2363340E 00
0.2588930E 00	0.2715958E 00	0.2593470E 00	0.2426100E 00
0.2242088E 00	0.2106870E 00	0.1886187E 00	0.1663788E 00
0.1477292E 00	0.1343106E 00	0.1218542E 00	0.1097493E 00
0.9979910E-01	0.8680904E-01	0.7769436E-01	0.6276107E-01
0.5106154E-01	0.3982237E-01	0.3015380E-01	0.2221074E-01
0.1571323E-01	0.9520765E-02	0.5431309E-02	0.2716314E-02
0.1153191E-02	0.5903142E-03	0.3588586E-03	0.7260863E-04
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.2246628E-04	0.2236100E-03	0.1733763E-02	0.7913966E-02
0.2478982E-01	0.5738202E-01	0.1026796E 00	0.1519760E 00
0.1982002E 00	0.2451878E 00	0.2428117E 00	0.2483238E 00
0.2688071E 00	0.2689522E 00	0.2546778E 00	0.2319517E 00
0.2151463E 00	0.1896352E 00	0.1741874E 00	0.1440163E 00
0.1363129E 00	0.1170444E 00	0.1052563E 00	0.9134817E-01
0.8146679E-01	0.7008237E-01	0.6129551E-01	0.4842377E-01
0.3848107E-01	0.2644810E-01	0.2167465E-01	0.1374268E-01
0.8248139E-02	0.4956193E-02	0.3049610E-02	0.1419656E-02
0.6481190E-03	0.1044638E-03	0.4576219E-04	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.2571527E-03	0.1822941E-02	0.9021621E-02	0.2665320E-01
0.5989974E-01	0.1023681E 00	0.1461299E 00	0.1856654E 00
0.2234071E 00	0.2657835E 00	0.2548606E 00	0.2599273E 00
0.2702767E 00	0.2659580E 00	0.2369433E 00	0.2231550E 00
0.1962218E 00	0.1779684E 00	0.1485134E 00	0.1321834E 00
0.1154602E 00	0.1054579E 00	0.9060305E-01	0.7756007E-01
0.6784779E-01	0.5828715E-01	0.4759468E-01	0.3566820E-01
0.2557244E-01	0.1899502E-01	0.1280696E-01	0.7447124E-02
0.4304644E-02	0.2219763E-02	0.1180247E-02	0.3952421E-03
0.1624286E-03	0.7658264E-04	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.2269397E-04	0.2308429E-03
0.2075878E-02	0.9624846E-02	0.3191707E-01	0.6668711E-01
0.1050972E 00	0.1461197E 00	0.1784285E 00	0.2115127E 00
0.2481316E 00	0.2835237E 00	0.2701750E 00	0.2609929E 00
0.2543824E 00	0.2523072E 00	0.2293909E 00	0.2022312E 00
0.1802447E 00	0.1499075E 00	0.1345164E 00	0.1111640E 00
0.9712094E-01	0.8830923E-01	0.7212412E-01	0.6215255E-01
0.5621058E-01	0.4325967E-01	0.3344726E-01	0.2382999E-01
0.1859563E-01	0.1131049E-01	0.6426115E-02	0.3995836E-02
0.1880914E-02	0.8638178E-03	0.3228260E-03	0.1022436E-03

0.3061809E-04	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.3218444E-04	0.2596709E-03	0.2107170E-02
0.9936295E-02	0.3130761E-01	0.6618536E-01	0.1071982E 00
0.1471151E 00	0.1940913E 00	0.2161806E 00	0.2493536E 00
0.2670345E 00	0.2680613E 00	0.2930642E 00	0.2637855E 00
0.2573879E 00	0.2264757E 00	0.2073898E 00	0.1796527E 00
0.1536861E 00	0.1332374E 00	0.1119388E 00	0.9786659E-01
0.8207232E-01	0.7238299E-01	0.5762001E-01	0.4698988E-01
0.3794770E-01	0.3120529E-01	0.2001548E-01	0.1510441E-01
0.1004597E-01	0.6521642E-02	0.3233414E-02	0.1113423E-02
0.6435823E-03	0.2916171E-03	0.5741679E-04	0.7639593E-04
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.0000000E 00
0.0000000E 00	0.0000000E 00	0.0000000E 00	0.8601016E-05
0.3439712E-04	0.3028943E-03	0.2249576E-02	0.1088468E-01
0.3506811E-01	0.7260209E-01	0.1124092E 00	0.1431631E 00
0.1702980E 00	0.1968122E 00	0.2334363E 00	0.2575313E 00
0.2904121E 00	0.3154174E 00	0.2771440E 00	0.2493343E 00
0.2392632E 00	0.2168028E 00	0.1860240E 00	0.1538441E 00
0.1345306E 00	0.1114675E 00	0.9396076E-01	0.7578874E-01
0.6721002E-01	0.5436332E-01	0.4397735E-01	0.3473518E-01
0.2713976E-01	0.1865069E-01	0.1498656E-01	0.8664083E-02
0.4691057E-02	0.2833721E-02	0.1506527E-02	0.6247566E-03
0.2911384E-03	0.1415498E-03	0.6554931E-04	

3.88774	3.9468	3.91022	3.85704	3.77676	3.68987	3.52102	3.16255
2.81195	2.60412	2.46239	2.41282	2.33947	2.18869		

```
$JOB   ACCT-NUM,VERNER,TIME=60
C      ***********************************************************************
C
C      PROGRAM 2.6 : PLOT
C
C      ***********************************************************************
```

```
3JOB PLOT; CHARGE BERGHOUT; CLASS 3;
USER 080002/VHJ;
PROCESSTIME=600;
BEGIN
3REMOVE MY/DATA FROM PACK;
COMPILE MY/DATA FORTRAN LIBRARY;
DATA



3DATA FILE5



3REMOVE MY/DATA FROM PACK;
3END JOB
```

```
C       ****************************************************************
C
C       PROGRAM : 2.6
C       TITLE : PLOT
C       PURPOSE : PLOT SIMULATED FERTILITY
C       DATA : DATA/XXAV : PROGRAM 2.7 : SURFACE
C       OUTPUT : SURFACE PLOT
C
C       ****************************************************************
$SET %
$SET SEPARATE
FILE  4=FILE4


        SUBROUTINE DATA(X,IR,IC,IF)
C       ****************************************************************
C
C       THIS ROUTINE IS USED TO PLOT FERTILITY PARAMETERS FOR
C       AGES 14 - 42 AND YEARS 1958 - 1970 ON THE CALCOMP PLOTTER
C       THE DATA FOR THE ARRAYS A (EIGEN VECTOR AVERAGES) AND
C       D (D VECTOR AVERAGES) WERE CALCULATED USING PROGRAM 2.3
C
C       ****************************************************************

        REAL X(130)
$SET OWN
        REAL A(13,4),D(29,4)
        INTEGER NDOM
$POP OWN
        DATA NDOM/0/
        IF (IR.GT.1) GO TO 10
        NDOM=NDOM+1
        IF (NDOM.GT.1) GO TO 10
        DO 60 J=1,4
        READ(IF)(A(I,J),I=1,13)
     60 CONTINUE
        DO 70 J=1,4
        READ(IF)(D(I,J),I=1,29)
     70 CONTINUE
     10 CONTINUE
        GO TO (1,2,3,4) NDOM


      1 DO 20 J=1,IC
     20     X(J)=A(IR,1)*D(J,1)
        RETURN
      2 DO 30 J=1,IC
     30     X(J)=A(IR,1)*D(J,1)+A(IR,2)*D(J,2)
        RETURN
      3 DO 40 J=1,IC
     40     X(J)=A(IR,1)*D(J ,1)+A(IR,2)*D(J ,2)+A(IR,3)*D(J ,3)
        RETURN
      4 DO 50 J=1,IC
     50     X(J)=A(IR,1)*D(J ,1)+A(IR,2)*D(J ,2)+A(IR,3)*D(J ,3)
      &     +A(IR,4)*D(J ,4)
        RETURN
        END
```

```
C      ******************************************************************
C
C      DATA FOR
C      PROGRAM 2.6 : PLOT
C
C      ******************************************************************
FERTILITY PARAMETERS - AGES 14 - 42   -  YEARS 1958 - 1970  -  1 COMPONENT
  13  29   2   2                                   1 1     3                 4
 30.0330.0  7.0       4.0                     20.0
FERTILITY PARAMETERS - AGES 14 - 42   -  YEARS 1958 - 1970  -  2 COMPONENTS
  13  29   2   2                                   1 1     3                 4
 30.0330.0  7.0       4.0                     20.0
FERTILITY PARAMETERS - AGES 14 - 42   -  YEARS 1958 - 1970  -  3 COMPONENTS
  13  29   2   2                                   1 1     3                 4
 30.0330.0  7.0       4.0                     20.0
FERTILITY PARAMETERS - AGES 14 - 42   -  YEARS 1958 - 1970  -  4 COMPONENTS
  13  29   2   2                                   1 1     3                 4
 30.0330.0  7.0       4.0                     20.0
```

```
$ JOB   ACCT_NUM,WERNER,TIME=60
C       *******************************************************************
C
C       PROGRAM 2.7 : SURFACE
C
C       *******************************************************************


3JOB SURFACE; CHARGE BERGHOUT; CLASS 3;
USER 080002/VHJ;
BEGIN
COMPILE SURFACE FORTRAN LIBRARY;
DATA
FILE   1=TEMP1,UNIT=DISKPACK,RECORD=130,BLOCKING=3,AREA=130              00001000
FILE   2=TEMP2,UNIT=DISKPACK,RECORD=130,BLOCKING=3,AREA=130              00002000
FILE   3=TEMP3,UNIT=DISKPACK,RECORD=400,AREA=1                           00003000
$INCLUDE   "QLIB/FS/CALCOMP"                                             00004000
$RESET LIST                                                              00005000
$SET AUTOBIND                                                            00006000
$BIND PLOTER FROM QLIB/=;                                                00007000


3END JOB
```

PROGRAM SURFACE

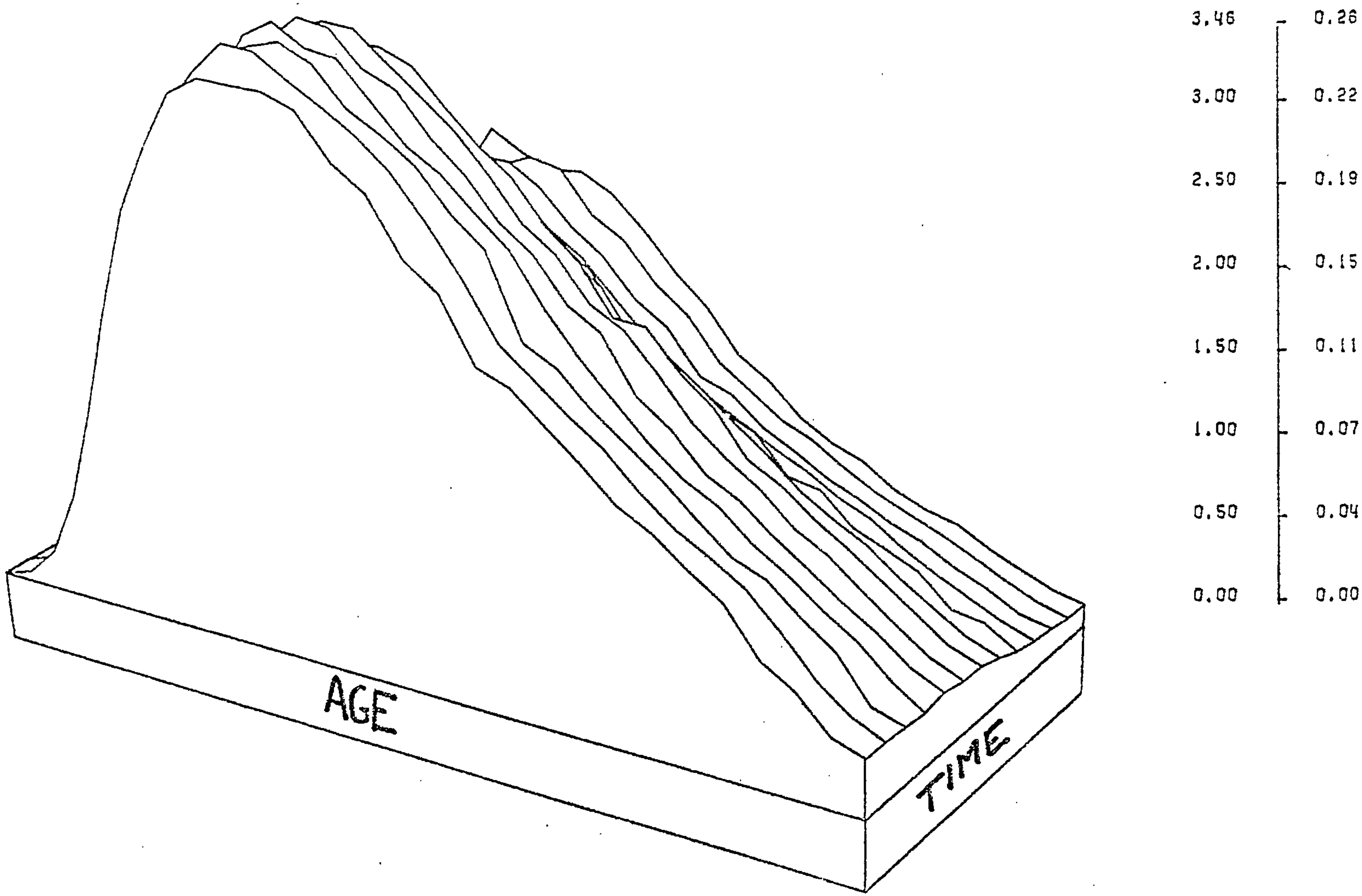
This program is a version of SYMVU modified to provide three-dimensional plots of two dimensional functions with the origin in front-left corner. This modification allows data to be specified row by row with that row with its profile in the foreground to be specified first.

Sourcelisting for the program SYMVU may be obtained from the

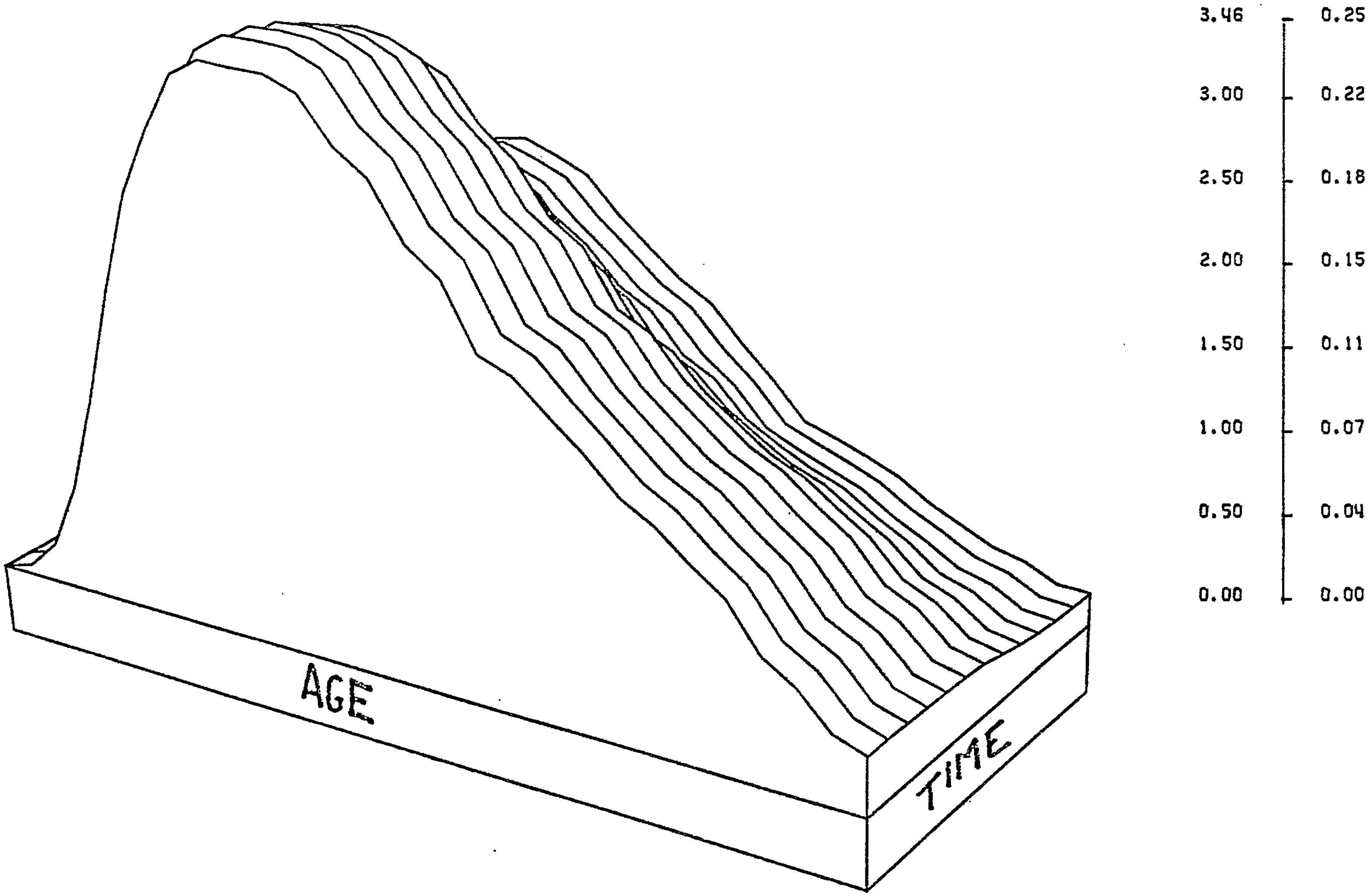
Laboratory for Computer Graphics and Spatial Analysis,
Graduate School of Design,
Harvard University,
520 Grund Hall, 48 Quincy Street,
Cambridge, Massachusetts
U.S.A. 02138

Source listing for the appropriate modifications may be obtained from

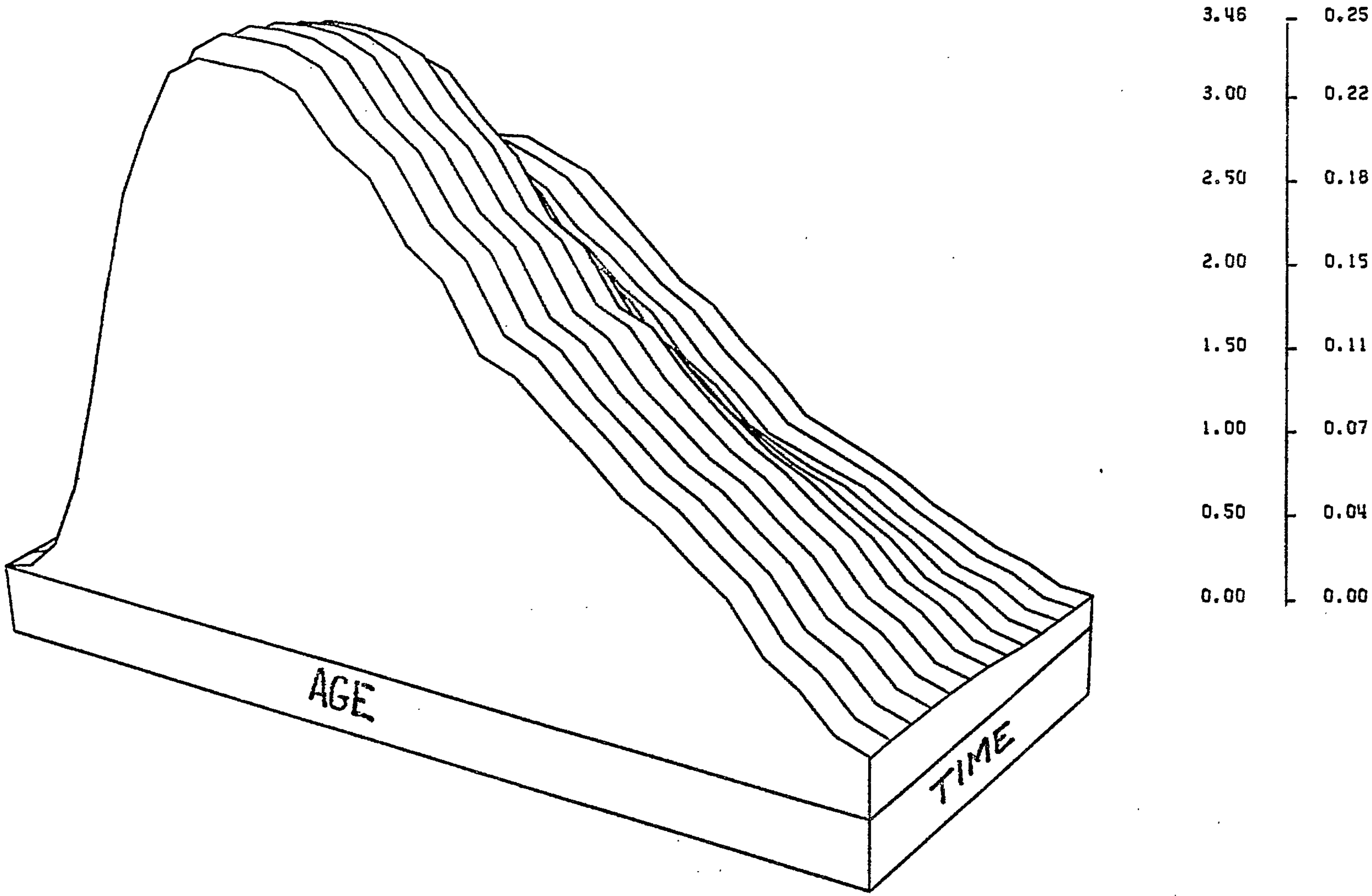
Dr. Henry Castner,
Department of Geography,
Queen's University,
Kingston, Ontario
Canada K7L 3N6



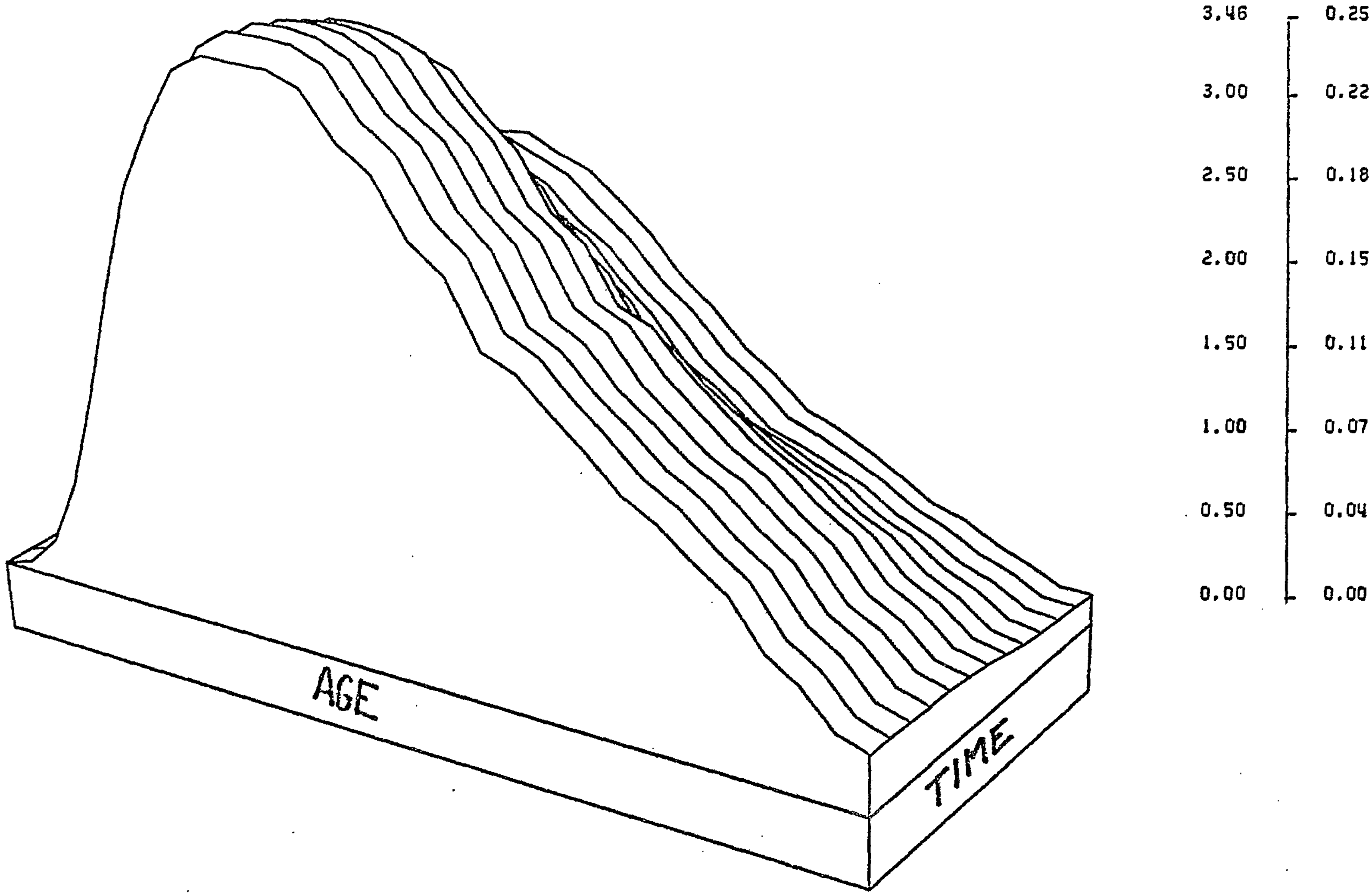
FERTILITY FROM VITAL STATISTICS DATA - 1958 TO 1970



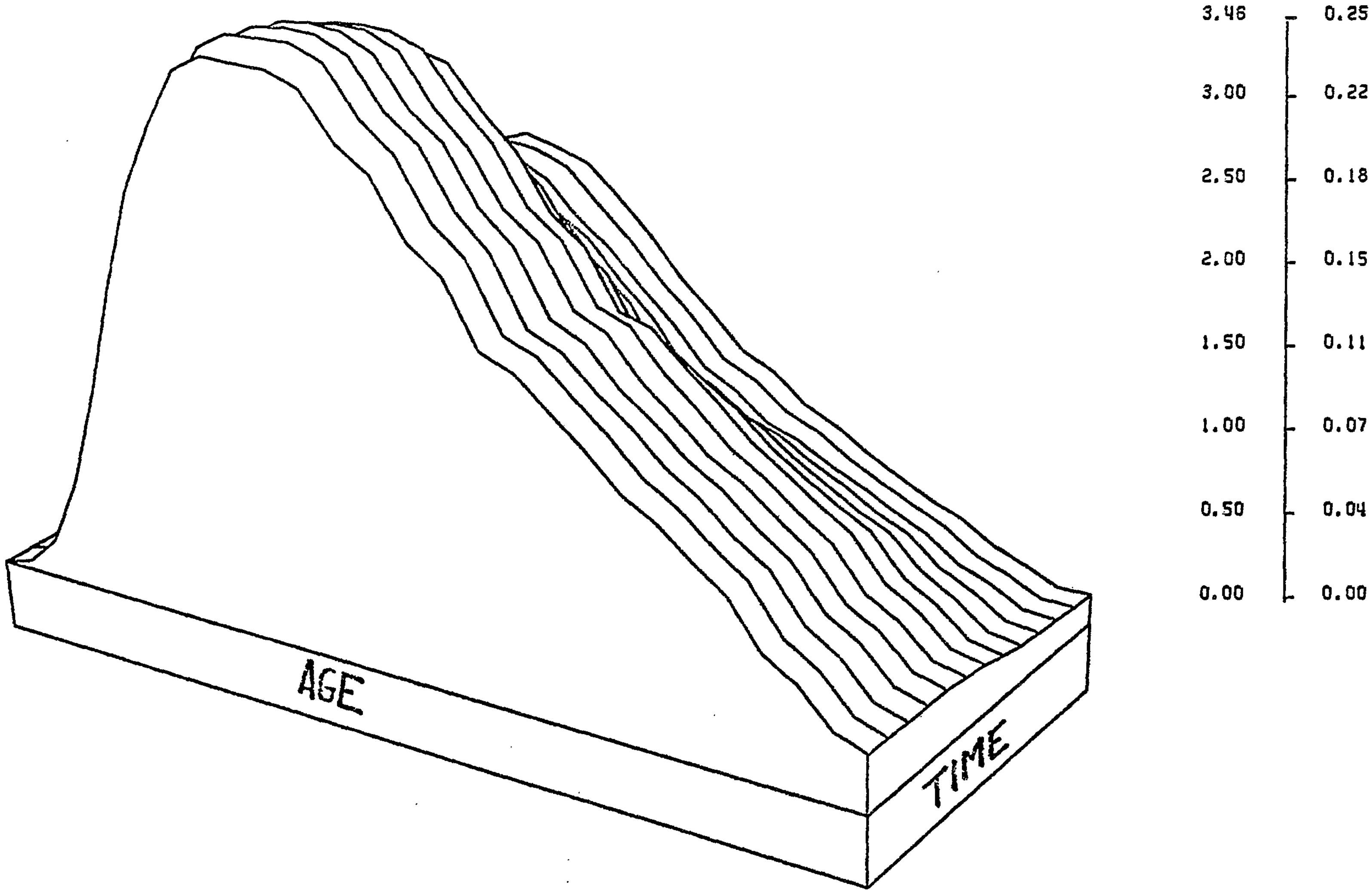
SIMULATED FERTILITY - AGES 14 - 42 - YEARS 1958 - 1970 - 1 COMPONENT



SIMULATED FERTILITY - AGES 14 - 42 - YEARS 1958 - 1970 - 2 COMPONENTS



SIMULATED FERTILITY - AGES 14 - 42 - YEARS 1958 - 1970 - 3 COMPONENTS



SIMULATED FERTILITY - AGES 14 - 42 - YEARS 1958 - 1970 - 4 COMPONENTS

3. ESTIMATION OF ECONOMIC MOBILITY

3.0 Introduction

This chapter contains source listings of programs used to estimate and display the economic mobility function. Further, to test the validity of the estimation procedures used, some aggregation experiments were run including, first, a simulation of age-, income-specific population propogated from a realistic mobility function, followed by a procedure for estimating the (economic) mobility, again followed by procedures for estimating the dominant modes of the mobility function.

The differential equation modelling economic mobility given in Chapter III leads to the formula

$$\mu(x,s,t) = \frac{-1}{\rho(x,s,t)} \int_{s}^{\bar{s}} \left(\frac{d\rho}{dt} + \frac{d\rho}{dx} + r(x,t)\ \rho(x,s,t)\right)ds \tag{10}$$

for estimating the mobility. Observe that mobility in the early years of age is definitely different from zero, and hence the upper limit of integration $\bar{s}$ is that value of the income for which the mobility (or else the population density) is assumed to be zero. (For the purposes of the aggregation experiments $\bar{s} = 20.$) This equation for estimating the mobility is used in much the same way that formula (4a) of section 2.1 is used for estimating the fertility parameters.

To remain consistent with Chapter II we begin with the programs for the aggregation experiments.

3.1 A Realistic Mobility Function

Motivated by preliminary results of values of the economic mobility estimated from data acquired (and our view of mobility at the boundaries), the following function was deemed to be a mobility function of realistic magnitude and shape:

$$u(x,s,t) = a_1(t)\, d_1(x,s) + .1\, a_2(t)\, d_2(x,s) + .02\, a_3(t)\, d_3(x,s)$$

where

$$\bar{a}_1(t) = (5+t)(15-t)$$

$$\bar{a}_2(t) = \sin \frac{2\pi t}{10} \qquad\qquad t \in (0,10)$$

$$\bar{a}_3(t) = \sin \frac{2\pi t}{5}$$

$$a_i(t) = \frac{\bar{a}_i(t)}{\sqrt{\sum_{j=0}^{10} a_i(j)^2}}$$

$$\bar{\eta}_1(x,s) = [\cos(.01(x+5s) + \cos(.005(4x+3s))]\ f(x,s)$$

$$\bar{\eta}_2(x,s) = [.000004x(100-x)s(25-s)]\ f(x,s)$$

$$\bar{\eta}_3(x,s) = [.0004xs+\exp(-(s-6)^2) + \exp(-\frac{(x-21)^2}{10})]\ f(x,s)$$

$$f(x,s) = \exp(-.05(4.5-s)^2 - .005(30.5-x)^2)$$

$$\eta_1(x,s) = \bar{\eta}_1(x,s)$$

$$\eta_2(x,s) = \bar{\eta}_2(x,s) - \frac{(\bar{\eta}_1,\bar{\eta}_2)}{(\bar{\eta}_1,\bar{\eta}_1)}\ \bar{\eta}_1$$

$$\eta_3(x,s) = \bar{\eta}_3(x,s) - \frac{(\eta_1,\bar{\eta}_3)}{(\eta_1,\eta_1)}\ \eta_1(x,s) - \frac{(\eta_2,\bar{\eta}_3)}{(\eta_2,\eta_2)}\ \eta_2(x,s)$$

$$(\eta_1,\eta_2) = \sum_{i=17}^{78} \sum_{j=0}^{19} \eta_1(i,j)\ \eta_2(i,j) \quad .$$

It will be observed that the sets

$$(a_1(t),\ a_2(t),\ a_3(t) \quad \text{and} \quad (\eta_1(x,s),\ \eta_2(x,s),\ \eta_3(x,s))$$

are orthogonal under the inner products implied. As mentioned in Chapter III above, this is necessary in order to facilitate recognition of the components in the computed estimates.

The returned time and age-income components are then unique up to multiplication and division by an arbitrary constant for each component.

The program will print values for all full intervals, and will store on a computer file all values corresponding to steps of length 1/NSTEP.

```
$JOB    ACCT-NUM,VERNER,TIME=60
C       ****************************************************************************
C
C               PROGRAM 3.1 :   MUTEST
C
C       ****************************************************************************
```

```
3JOB MUTEST; CHARGE VERNER; CLASS 3
3USER 080002/VHJ
3PROCESSTIME=600;
5PRINTLIMIT=13200
3BEGIN
3REMOVE DATA/XXMU9 FROM PACK;
3COMPILE MUTEST FORTRAN;
FILE FILE3(KIND=PACK,TITLE=DATA/XXMU9,MAXRECSIZE=39,BLOCKSIZE=1209,
AREASIZE=31,AREAS=84,FLEXIBLE,SAVEFACTOR=5)
$FILE FILE4(KIND=PACK,TITLE=PLOT/SIMMU,MAXRECSIZE=40,BLOCKSIZE=120,
AREASIZE=20, AREAS=1,FLEXIBLE,SAVEFACTOR=10);
OPTION=AUTORM;
3DATA



3DATA



3END JOB
```

```
C       ********************************************************************
C
C       PROGRAM : 3.1
C       TITLE : MUTEST
C       PURPOSE : CALCULATE VALUES OF SIMULATED MU
C       DATA : DIMENSIONS OF MU ARRAY - MU(X,S,T)
C               X = AGE : S = INCOME : T = TIME
C       OUTPUT : DATA/XXMU9 - VALUES OF MU :
C                    PLOT/SIMMU ON PACK : TO PLOT THREE DOMINANT MODES OF MU
C
C       ********************************************************************
$INCLUDE 'OLIB/FS/PLOTT'
C       ********************************************************************
C
C       THIS PROGRAM CREATES A SIMULATED MU  EVALUATED IN STEPS OF
C       1./NSTEP  FOR THE VARIABLES
C
C               AGE:  17 - 78 (YEARS)
C               INCOME:    0 - 19 ($1000)
C               TIME:  1963 - 1973
C
C       REAL AE(3,TM),NU(3,XE,SE),MU(XE,SE,TM)
C       XL = NSTEP*XE-1   $   SL = NSTEP*SE-1   $   TL = NSTEP*TM-1
C       REAL AE1(TL),AE2(TL),AE3(TL),NU1(XL,SL),NU2(XL,SL),NU3(XL,S.)
C       MUT(SL)
C
C       ********************************************************************
        REAL AE(3,11),NU(3,62,20),MU(62,20,11)
        INTEGER LS,MS,LX,MX,XE,SE,TM
        INTEGER XL,SL,TL,NSTEP
        REAL AE1(21),AE2(21),AE3(21),NU1(123,39),NU2(123,39),NU3(123,39)
        REAL MUT(39)
        REAL AF(3),NF(3)
        REAL PT(11,4)
        READ(5,131) LS,MS
        READ(5,131) LX,MX
        READ(5,132) XE,SE,TM
        READ(5,133) NSTEP
        XL=NSTEP*XE-1
        SL=NSTEP*SE-1
        TL=NSTEP*TM-1
        WRITE(6,141) LS,MS
        WRITE(6,142) LX,MX
        WRITE(6,143) XE,SE,TM
    131 FORMAT(2I5)
    132 FORMAT(3I5)
    133 FORMAT(I2)
    141 FORMAT(' ','LS = ',I5,' AND MS = ',I5,//)
    142 FORMAT(' ','LX = ',I5,' AND MX = ',I5,//)
    143 FORMAT(' ','XE = ',I5,' AND SE = ',I5,' AND TM = ',I5,//)
        CALL MUTEST(LX,MX,LS,MS,XE,SE,TM,AE,NU,MU,AF,NF)
        WRITE(6,111)
        DO 10 I=1,3
     10     WRITE(6,101) (AE(I,J),J=1,TM)
        WRITE(6,112)
        WRITE(6,102)  (NU(1,IX,5),IX=1,XE)
        WRITE(6,102)  (NU(2,IX,5),IX=1,XE)
        WRITE(6,102)  (NU(3,IX,5),IX=1,XE)
        WRITE(6,113)
```

```
            WRITE(6,102)  (NU(1,IX,7),IX=1,XF)
            WRITE(6,102)  (NU(2,IX,7),IX=1,XF)
            WRITE(6,102)  (NU(3,IX,7),IX=1,XF)
            DO 89 I=10,32
                 I1=I+LX-1
            WRITE(6,117) I1
            WRITE(6,118)
                 DO 88 J=1,SE
                      WRITE(6,116) J-1, (MU(I,J,KK),KK=1,6)
*EXTENSION*   IO-3
        88            CONTINUE
            WRITE(6,119)
                 DO 89 J=1,SE
                      WRITE(6,116) J-1, (MU(I,J,KK),KK=7,TM)
*EXTENSION*   IO-3
        89            CONTINUE
            DO 59 I=1,TM
                 PT(I,1)=1962+I
                 DO 59 J=2,4
        59            PT(I,J)=AE(J-1,I)
            CALL PLOTB(100,PT,TM,4,0.,0.,0.,0.)
       101 FORMAT(6F11.6,/,5F11.6,//)
       102 FORMAT(12(5F11.6,/),2F11.6,//)
       111 FORMAT('0','THE THREE TIME COMPONENTS ARE',//)
       112 FORMAT('1','THREE AGE VECTORS AT S=$4000 ARE',//)
       113 FORMAT('1','THREE AGE VECTORS AT S=$6000 ARE',//)
       116 FORMAT(I4,5X,6F10.6)
       117 FORMAT('1',' VALUES OF SIMULATED ECONOMIC MOBILITY AT AGE = 'I2,/
          &)
       118 FORMAT(' INCOME/TIME 1963       1964       1965       1966       1967
          &     1968',//)
       119 FORMAT(//,' INCOME/TIME 1969       1970       1971       1972       1
          &73',//)
            IF (NSTEP.NE.1)
          &CALL STORMU(AF,NF,AE1,AE2,AE3,NU1,NU2,NU3,XL,SL,TL,NSTEP,MUT,
          &             LX,LS)
            IF (NSTEP.NE.1) GO TO 98
            DO 99 KK=1,TM
                 DO 99 I=1,XE
                      WRITE(3)  (MU(I,J,KK),J=1,SE)
        99            CONTINUE
        98       CONTINUE
            DO 79 K=1,3
                 DO 79 I=1,20
        79            WRITE(4)  (NU(K,J,I),J=1,40)
            LOCK 3

            LOCK 4

            STOP
            END

            SUBROUTINE MUTEST(LX,MX,LS,MS,XE,SE,TM,AE,NU,MU,AF,NF)
            INTEGER LX,MX,LS,MS,TM,XF,SE
            REAL AE(3,TM),NU(3,XF,SE),MU(XE,SE,TM)
            REAL AF(3),NF(3)
            PI=3.1415926536
            DO 10 IT=1,TM
                 AE(1,IT)=(4.+IT)*(16.-IT)
                 AE(2,IT)=SIN(PI*(IT-1)/5.)
```

```
                AF(3,IT)=SIN(PI*(IT-1)/2.5)
   10           CONTINUE
                DO 15 I=1,3
                    SUM=0.
                DO 14 J=1,IM
                    SUM=SUM+AF(I,J)**2
   14               CONTINUE
                SUM=1./SQRT(SUM)
                AF(I)=SUM
                DO 15 J=1,IM
                    AF(I,J)=AF(I,J)*SUM
   15               CONTINUE
            SUM=0
            SUM1=0
            SUM2=0
            I=0
            DO 20 IX=LX,MX
                I=I+1
                FX=EXP(-.005*(30.5-IX)**2)
                J=0
                DO 20 IS=LS,MS
                    J=J+1
                    FS=EXP(-.05*(4.5-IS)**2)*FX
                    NU(1, I, J)=COS(.01*(IX+5*IS))+COS(.005*(4*IX+3.*IS))
                    NU(2, I, J)=.000004*IX*(100.-IX)*IS*(25.-IS)
                    NU(3, I, J)=.0004*IX*IS+EXP(-(IS-6.)**2)+EXP(-((IX-21.)
     &                       **2)/10.)
                    NU(1,I,J)=NU(1,I,J)*FS
                    NU(2,I,J)=NU(2,I,J)*FS
                    NU(3,I,J)=NU(3,I,J)*FS
                    IF(I.LT.10.OR.I.GT.52.OR.J.LT.3.OR.J.GT.12) GOTO 20
                    SUM=SUM+NU(1,I,J)*NU(1,I,J)
                    SUM1=SUM1+NU(1,I,J)*NU(2,I,J)
                    SUM2=SUM2+NU(1,I,J)*NU(3,I,J)
   20               CONTINUE
            SUM1-SUM1/SUM
            NF(1)=SUM1
            SUM2=SUM2/SUM
            NF(2)=SUM2
            SUM -0
            SUM3=0
            I=0
            DO 30 IX=LX,MX
                I=I+1
                J=0
                DO 30 IS=LS,MS
                    J=J+1
                    NU(2,I,J)=NU(2,I,J)-SUM1*NU(1,I,J)
                    NU(3,I,J)=NU(3,I,J)-SUM2*NU(1,I,J)
                    IF(I.LT.10.OR.I.GT.52.OR.J.LT.3.OR.J.GT.12) GOTO 30
                    SUM=SUM+NU(2,I,J)*NU(2,I,J)
                    SUM3=SUM3+NU(2,I,J)*NU(3,I,J)
   30               CONTINUE
            SUM3=SUM3/SUM
            NF(3)=SUM3
            I=0
            DO 40 IX=LX,MX
                I=I+1
                J=0
                DO 40 IS=LS,MS
```

```
                    J=J+1
                    NU(3,I,J)=NU(3,I,J)-SUM3*NU(2,I,J)
                    DO 40 IT=1,TM
                          MU(I,J,IT)=AE(1,IT)*NU(1, I, J) +
      &                                      .1*AE(2,IT)*NU(2,I,J) +
      &                                      .02*AE(3,IT)*NU(3,I,J)
   40               CONTINUE
        RETURN
        END

        SUBROUTINE STORMU(AF,NF,AE1,AE2,AE3,NU1,NU2,NU3,XL,SL,TL,NSTEP,
      & MUT,LX,LS)
        INTEGER XL,SL,TL,NSTEP
        REAL IX1,IS1,IT1
        REAL AE1(TL),AE2(TL),AE3(TL),NU1(XL,SL),NU2(XL,SL),NU3(XL,SL)
        REAL MUT(SL),AF(3),NF(3)
        PI=3.1415926536
        DO 10 IT=1,TL
              IT1=(IT-1.)/NSTEP
              AE1(IT)=(5.+IT1)*(15.-IT1)*AF(1)
              AE2(IT)=SIN(PI*(IT1/5.))*.1*AF(2)
              AE3(IT)=SIN(PI*(IT1/2.5))*.02*AF(3)
   10         CONTINUE
        DO 20 IX=1,XL
              IX1=LX+(IX-1.)/NSTEP
              FX=EXP(-.005*(30.5-IX1)**2)
              DO 20 IS=1,SL
                    IS1=LS+(IS-1.)/NSTEP
                    FS=EXP(-.05*(4.5-IS1)**2)*FX
                    NU1(IX,IS)=(COS(.01*(IX1+5.*IS1))
      &                        +COS(.005*(4.*IX1+3.*IS1)))*FS
                    NU2(IX,IS)=.000004*IX1*(100.-IX1)*IS1*(25.-IS1)*FS
      &                        -NF(1)*NU1(IX,IS)
                    NU3(IX,IS)=(.0004*IX1*IS1+EXP(-(IS1-6.)**2)
      &                        +EXP(-((IX1-21.)**2)/10.))*FS
      &                        -NF(2)*NU1(IX,IS)-NF(3)*NU2(IX,IS)
   20               CONTINUE
        DO 30 IT=1,TL
              DO 30 IX=1,XL
                    DO 25 IS=1,SL
   25                     MUT(IS)=AE1(IT)*NU1(IX,IS)+AE2(IT)*NU2(IX,IS)
      &                        +AE3(IT)*NU3(IX,IS)
        IF (IX.EQ.28.AND.IT.EQ. 8) WRITE(6,124) (MUT(IS),IS=1,SL)
        IF (IX.EQ.29.AND.IT.EQ. 9) WRITE(6,123) (MUT(IS),IS=1,SL)
                    WRITE(3) (MUT(IS),IS=1,SL)
   30               CONTINUE
  124 FORMAT('1','EXTENDED MU FOR X = 30.5  AND T = 1966.5',//,
      &8(5F12.6,/),//)
  123 FORMAT(//,' EXTENDED MU FOR X = 31  AND  T = 1967',//,
      &8(5F12.6,/),//)
        LOCK 3

        RETURN
        END
```

```
********************************************************************

DATA FOR
     PROGRAM 3.1 :  MUTEST

********************************************************************
  0    19
 17    78
 62    20    11
2
```

LS = 0 AND MS = 19

LX = 17 AND MX = 78

XF = 62 AND SF = 20 AND TM = 11

THE THREE TIME COMPONENTS ARE

0.250058	0.280065	0.303404	0.320075	0.330077	0.333411
0.330077	0.320075	0.303404	0.280065	0.250058	
0.000000	0.262866	0.425325	0.425325	0.262866	-0.000000
-0.262866	-0.425325	-0.425325	-0.262866	0.000000	
0.000000	0.425325	0.262866	-0.262866	-0.425325	0.000000
0.425325	0.262866	-0.262866	-0.425325	-0.000000	

THREE AGE VECTORS AT S=$4000 ARE

0.735846	0.832742	0.932749	1.034067	1.134646
1.232249	1.324527	1.409113	1.483719	1.546237
1.594839	1.628064	1.644895	1.644803	1.627781
1.594342	1.545496	1.482695	1.407764	1.322814
1.230140	1.132122	1.031125	0.929402	0.829022
0.731802	0.639266	0.552616	0.472731	0.400173
0.335210	0.277853	0.227895	0.184956	0.148527
0.118016	0.092791	0.072170	0.055542	0.042290
0.031856	0.023739	0.017501	0.012763	0.009207
0.006569	0.004636	0.003236	0.002234	0.001525
0.001030	0.000687	0.000454	0.000296	0.000191
0.000122	0.000077	0.000048	0.000029	0.000018
0.000011	0.000006			

-0.193502	-0.207742	-0.220261	-0.230505	-0.238039
-0.242477	-0.243526	-0.241005	-0.234860	-0.225177
-0.212174	-0.196203	-0.177727	-0.157301	-0.135546
-0.113113	-0.090654	-0.068792	-0.048086	-0.029017
-0.011960	0.002815	0.015156	0.025020	0.032464
0.037630	0.040730	0.042023	0.041798	0.040356
0.037990	0.034979	0.031569	0.027973	0.024367
0.020887	0.017631	0.014664	0.012024	0.009725
0.007759	0.006110	0.004749	0.003645	0.002763
0.002068	0.001529	0.001117	0.000806	0.000574
0.000405	0.000282	0.000194	0.000132	0.000088
0.000059	0.000038	0.000025	0.000016	0.000010
0.000006	0.000004			

0.066328	0.161800	0.310001	0.472033	0.573508
0.553517	0.415853	0.225849	0.056520	-0.057197
-0.120126	-0.152468	-0.170675	-0.182926	-0.191997
-0.198479	-0.202373	-0.203623	-0.202240	-0.198327
-0.192071	-0.183731	-0.173626	-0.162111	-0.149564
-0.136364	-0.122878	-0.109439	-0.096346	-0.083844
-0.072130	-0.061344	-0.051578	-0.042874	-0.035236
-0.028630	-0.023000	-0.018267	-0.014345	-0.011137
-0.008548	-0.006487	-0.004866	-0.003609	-0.002646
-0.001917	-0.001373	-0.000972	-0.000680	-0.000470
-0.000321	-0.000217	-0.000144	-0.000095	-0.000062
-0.000040	-0.000025	-0.000016	-0.000010	-0.000006
-0.000004	-0.000002			

THREE AGE VECTORS AT S=$6000 ARE

0.646833	0.731276	0.818265	0.906210	0.993307
1.077597	1.157030	1.229550	1.293180	1.346108
1.386776	1.413955	1.426805	1.424919	1.408344
1.377580	1.333555	1.277576	1.211268	1.136494
1.055267	0.969664	0.881736	0.793430	0.706521
0.622562	0.542842	0.468372	0.399879	0.337813
0.282375	0.233544	0.191116	0.154737	0.123952
0.098234	0.077021	0.059741	0.045840	0.034794
0.026124	0.019401	0.014251	0.010353	0.007438
0.005284	0.003712	0.002579	0.001771	0.001202
0.000807	0.000535	0.000351	0.000227	0.000145
0.000092	0.000057	0.000035	0.000021	0.000013
0.000008	0.000004			

-0.104410	-0.104060	-0.100768	-0.094427	-0.084788
-0.071788	-0.055498	-0.036138	-0.014075	0.010186
0.036017	0.062696	0.089440	0.115442	0.139912
0.162118	0.181419	0.197299	0.209389	0.217480
0.221525	0.221636	0.218066	0.211188	0.201468
0.189434	0.175646	0.160667	0.145033	0.129237
0.113708	0.098802	0.084800	0.071903	0.060239
0.049869	0.040800	0.032991	0.026368	0.020832
0.016270	0.012562	0.009588	0.007236	0.005399
0.003983	0.002905	0.002095	0.001494	0.001053
0.000734	0.000506	0.000345	0.000232	0.000155
0.000102	0.000066	0.000043	0.000027	0.000017
0.000011	0.000007			

0.376070	0.504533	0.682332	0.873563	1.010094
1.035908	0.953465	0.820883	0.703248	0.631239
0.599651	0.589497	0.585343	0.579434	0.569106
0.553889	0.534056	0.510128	0.482739	0.452587
0.420401	0.386912	0.352827	0.318806	0.285444
0.253255	0.222664	0.194003	0.167511	0.143339
0.121557	0.102164	0.085099	0.070253	0.057481
0.046612	0.037461	0.029839	0.023556	0.018431
0.014291	0.010983	0.008364	0.006313	0.004722
0.003500	0.002570	0.001871	0.001349	0.000964
0.000683	0.000479	0.000333	0.000229	0.000156
0.000106	0.000071	0.000047	0.000031	0.000020
0.000013	0.000008			

VALUES OF SIMULATED ECONOMIC MOBILITY AT AGE = 26

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.150590	0.161300	0.169954	0.178941	0.189715	0.200786
1	0.222007	0.239726	0.254125	0.267929	0.282513	0.296009
2	0.295680	0.321707	0.342944	0.362016	0.379808	0.394240
3	0.355750	0.389823	0.417700	0.441416	0.460972	0.474334
4	0.386649	0.426642	0.459257	0.485635	0.504945	0.515533
5	0.379592	0.424113	0.457290	0.480600	0.496802	0.506123
6	0.336606	0.382636	0.412167	0.427970	0.439217	0.448807
7	0.269589	0.304631	0.330917	0.348221	0.357469	0.359452
8	0.194998	0.220001	0.240431	0.254962	0.261479	0.259998
9	0.127372	0.144298	0.158177	0.167876	0.171726	0.169829
10	0.075126	0.085453	0.093909	0.099702	0.101746	0.100168
11	0.040007	0.045669	0.050297	0.053412	0.054395	0.053343
12	0.019234	0.022024	0.024299	0.025806	0.026235	0.025645
13	0.008347	0.009582	0.010586	0.011243	0.011413	0.011129
14	0.003269	0.003761	0.004159	0.004416	0.004478	0.004359
15	0.001156	0.001331	0.001472	0.001563	0.001584	0.001541
16	0.000368	0.000425	0.000470	0.000498	0.000505	0.000491
17	0.000106	0.000122	0.000135	0.000143	0.000145	0.000141
18	0.000027	0.000032	0.000035	0.000037	0.000038	0.000037
19	0.000006	0.000007	0.000008	0.000009	0.000009	0.000009

INCOME/TIME	1969	1970	1971	1972	1973
0	0.207842	0.206568	0.195477	0.176021	0.150590
1	0.303584	0.300409	0.284611	0.257569	0.222007
2	0.400787	0.394925	0.374573	0.340616	0.295680
3	0.478209	0.469305	0.445587	0.407057	0.355750
4	0.515810	0.504188	0.479013	0.439453	0.386649
5	0.505322	0.491157	0.463854	0.426174	0.379592
6	0.449421	0.433741	0.404663	0.371361	0.336606
7	0.354246	0.341927	0.323287	0.299248	0.269589
8	0.253317	0.244233	0.232765	0.216795	0.194998
9	0.164535	0.158195	0.150911	0.141014	0.127372
10	0.096586	0.092620	0.088396	0.082829	0.075126
11	0.051224	0.049006	0.046787	0.043947	0.040007
12	0.024543	0.023433	0.022376	0.021060	0.019234
13	0.010623	0.010126	0.009669	0.009115	0.008347
14	0.004153	0.003954	0.003775	0.003563	0.003269
15	0.001467	0.001395	0.001332	0.001257	0.001156
16	0.000468	0.000445	0.000424	0.000401	0.000368
17	0.000135	0.000128	0.000122	0.000115	0.000106
18	0.000035	0.000033	0.000032	0.000030	0.000027
19	0.000008	0.000008	0.000007	0.000007	0.000006

VALUES OF SIMULATED ECONOMIC MOBILITY AT AGE = 27

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.155647	0.166560	0.175565	0.185048	0.196244	0.207530
1	0.229350	0.247485	0.262505	0.277073	0.292187	0.305800
2	0.305305	0.332032	0.354223	0.374354	0.392733	0.407073
3	0.367135	0.402215	0.431385	0.456419	0.476546	0.489514
4	0.398803	0.440060	0.474225	0.502075	0.521864	0.531737
5	0.391297	0.437318	0.472108	0.496770	0.513272	0.521730
6	0.346775	0.394436	0.425438	0.442251	0.453589	0.462367
7	0.277557	0.313853	0.341450	0.359761	0.369052	0.370076
8	0.200627	0.226535	0.247988	0.263334	0.269849	0.267502
9	0.130955	0.148503	0.163082	0.173324	0.177140	0.174607
10	0.077181	0.087891	0.096776	0.102893	0.104899	0.102908
11	0.041069	0.046942	0.051805	0.055094	0.056047	0.054758
12	0.019727	0.022621	0.025012	0.026604	0.027014	0.026303
13	0.008553	0.009834	0.010889	0.011583	0.011744	0.011404
14	0.003347	0.003856	0.004274	0.004546	0.004604	0.004462
15	0.001182	0.001363	0.001512	0.001608	0.001627	0.001575
16	0.000376	0.000435	0.000482	0.000512	0.000518	0.000502
17	0.000108	0.000125	0.000138	0.000147	0.000149	0.000144
18	0.000028	0.000032	0.000036	0.000038	0.000038	0.000037
19	0.000007	0.000008	0.000008	0.000009	0.000009	0.000009

INCOME/TIME	1969	1970	1971	1972	1973
0	0.214665	0.213409	0.202140	0.182091	0.155647
1	0.313298	0.310064	0.294052	0.266260	0.229350
2	0.413271	0.407226	0.386649	0.351850	0.305305
3	0.492691	0.483447	0.459530	0.420168	0.367135
4	0.530975	0.518860	0.493537	0.453258	0.398803
5	0.519753	0.504951	0.477440	0.439189	0.391297
6	0.461897	0.445493	0.416069	0.382340	0.346775
7	0.363698	0.350785	0.332088	0.307875	0.277557
8	0.259805	0.250271	0.238866	0.222869	0.200627
9	0.168581	0.161921	0.154702	0.144836	0.130955
10	0.098859	0.094691	0.090517	0.084994	0.077181
11	0.052374	0.050042	0.047855	0.045052	0.041069
12	0.025066	0.023899	0.022860	0.021568	0.019727
13	0.010837	0.010314	0.009866	0.009325	0.008553
14	0.004232	0.004022	0.003847	0.003641	0.003347
15	0.001493	0.001417	0.001355	0.001283	0.001182
16	0.000475	0.000451	0.000431	0.000408	0.000376
17	0.000137	0.000130	0.000124	0.000117	0.000108
18	0.000036	0.000034	0.000032	0.000030	0.000028
19	0.000008	0.000008	0.000008	0.000007	0.000007

VALUES OF SIMULATED ECONOMIC MOBILITY AT AGE = 28

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.159228	0.170332	0.179567	0.189342	0.200818	0.212304
1	0.234509	0.253029	0.268475	0.283497	0.298941	0.312679
2	0.312010	0.339378	0.362246	0.383012	0.401725	0.416013
3	0.374995	0.410988	0.441095	0.466928	0.487335	0.499994
4	0.407111	0.449510	0.484815	0.513559	0.533526	0.542815
5	0.399214	0.446570	0.482559	0.508023	0.524551	0.532285
6	0.353571	0.402663	0.434766	0.452139	0.463348	0.471428
7	0.282812	0.320223	0.348809	0.367717	0.376872	0.377083
8	0.204284	0.231004	0.253232	0.269074	0.275457	0.272379
9	0.133245	0.151351	0.166461	0.177032	0.180732	0.177660
10	0.078470	0.089523	0.098734	0.105046	0.106967	0.104626
11	0.041720	0.047782	0.052824	0.056217	0.057116	0.055626
12	0.020022	0.023010	0.025488	0.027130	0.027511	0.026696
13	0.008673	0.009995	0.011089	0.011804	0.011950	0.011564
14	0.003390	0.003916	0.004349	0.004629	0.004681	0.004520
15	0.001196	0.001383	0.001537	0.001635	0.001652	0.001594
16	0.000380	0.000440	0.000489	0.000520	0.000526	0.000507
17	0.000109	0.000126	0.000140	0.000149	0.000151	0.000145
18	0.000028	0.000033	0.000036	0.000038	0.000039	0.000038
19	0.000007	0.000008	0.000008	0.000009	0.000009	0.000009

INCOME/TIME	1969	1970	1971	1972	1973
0	0.219544	0.218282	0.206826	0.186338	0.159228
1	0.320163	0.316846	0.300600	0.272271	0.234509
2	0.421981	0.415732	0.394898	0.359524	0.312010
3	0.502653	0.493060	0.468893	0.429001	0.374995
4	0.541247	0.528646	0.503108	0.462419	0.407111
5	0.529374	0.513964	0.486199	0.447669	0.399214
6	0.470081	0.453004	0.423234	0.389337	0.353571
7	0.369752	0.356283	0.337482	0.313276	0.282812
8	0.263854	0.253894	0.242498	0.226593	0.204284
9	0.171035	0.164075	0.156980	0.147118	0.133245
10	0.100194	0.095837	0.091686	0.086249	0.078470
11	0.053024	0.050585	0.048416	0.045670	0.041720
12	0.025348	0.024127	0.023100	0.021840	0.020022
13	0.010946	0.010399	0.009957	0.009432	0.008673
14	0.004269	0.004050	0.003878	0.003678	0.003390
15	0.001504	0.001425	0.001364	0.001295	0.001196
16	0.000478	0.000453	0.000434	0.000412	0.000380
17	0.000137	0.000130	0.000124	0.000118	0.000109
18	0.000036	0.000034	0.000032	0.000031	0.000028
19	0.000008	0.000008	0.000008	0.000007	0.000007

VALUES OF SIMULATED ECONOMIC MOBILITY AT AGE = 29

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.161223	0.172448	0.181906	0.191724	0.203351	0.214963
1	0.237326	0.256108	0.271805	0.287057	0.302652	0.316434
2	0.315590	0.343410	0.366699	0.387794	0.406619	0.420787
3	0.379090	0.415742	0.446453	0.472703	0.493143	0.505454
4	0.411320	0.454554	0.490611	0.519827	0.539722	0.548426
5	0.403097	0.451437	0.488231	0.514104	0.530439	0.537463
6	0.356785	0.406929	0.439780	0.457411	0.468327	0.475713
7	0.285192	0.323434	0.352698	0.371912	0.380797	0.380255
8	0.205857	0.233188	0.255951	0.272055	0.278211	0.274476
9	0.134170	0.152697	0.168176	0.178920	0.182444	0.178893
10	0.078951	0.090265	0.099702	0.106116	0.107917	0.105268
11	0.041940	0.048146	0.053312	0.056759	0.057587	0.055920
12	0.020109	0.023168	0.025707	0.027374	0.027717	0.026813
13	0.008702	0.010055	0.011176	0.011901	0.012030	0.011603
14	0.003398	0.003936	0.004379	0.004663	0.004708	0.004530
15	0.001197	0.001389	0.001546	0.001646	0.001660	0.001596
16	0.000380	0.000442	0.000492	0.000523	0.000528	0.000507
17	0.000109	0.000127	0.000141	0.000150	0.000151	0.000145
18	0.000028	0.000033	0.000036	0.000039	0.000039	0.000038
19	0.000007	0.000008	0.000008	0.000009	0.000009	0.000009

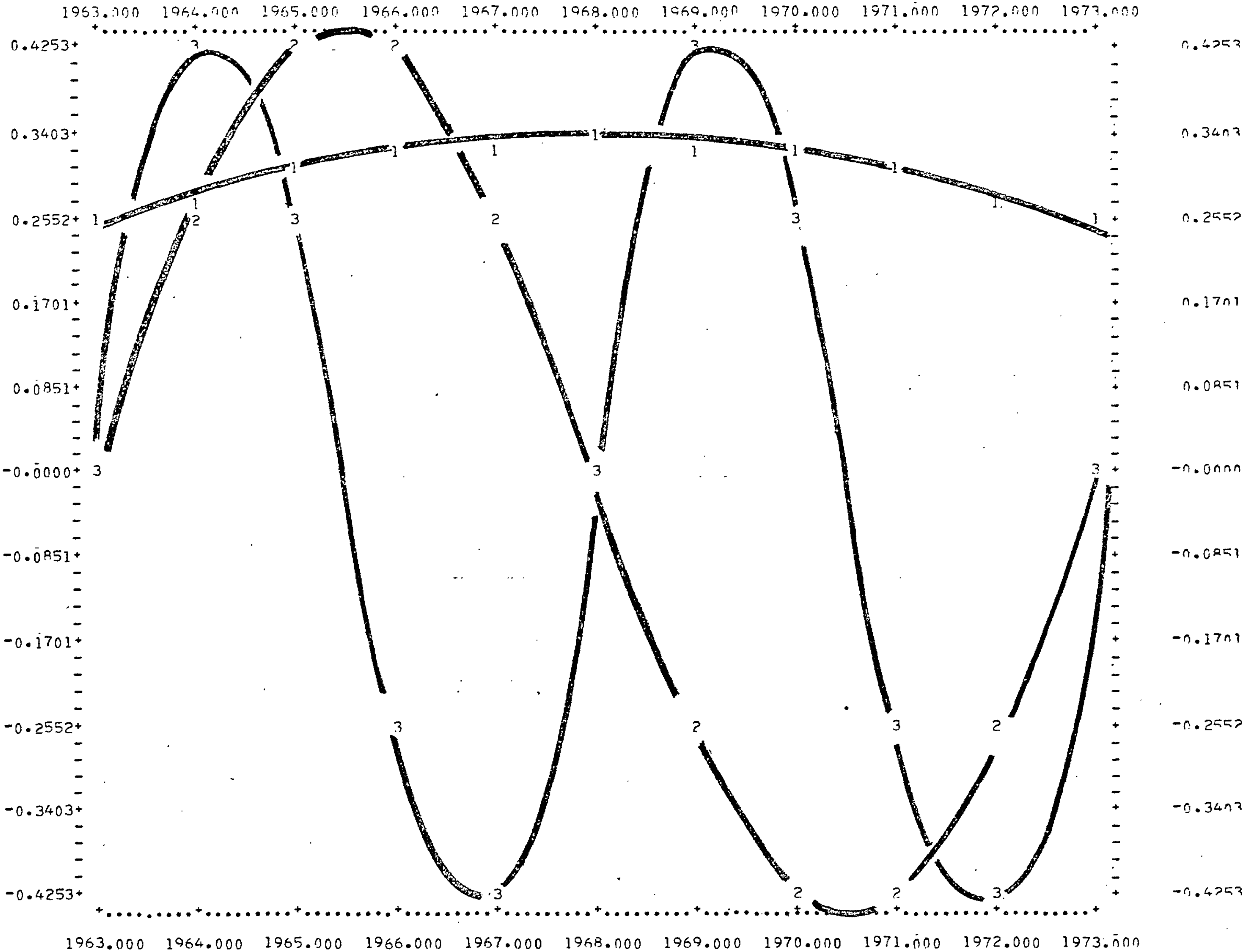
INCOME/TIME	1969	1970	1971	1972	1973
0	0.222277	0.221006	0.209428	0.188690	0.161223
1	0.323888	0.320497	0.304106	0.275502	0.237326
2	0.426540	0.420117	0.399134	0.363512	0.315590
3	0.507655	0.497768	0.473473	0.433420	0.379090
4	0.546162	0.533151	0.507524	0.466802	0.411320
5	0.533738	0.517825	0.489952	0.451501	0.403097
6	0.473584	0.455957	0.426017	0.392268	0.356785
7	0.372109	0.358179	0.339367	0.315395	0.285192
8	0.265253	0.254940	0.243596	0.227932	0.205857
9	0.171765	0.164555	0.157410	0.147844	0.134170
10	0.100514	0.096000	0.091887	0.086586	0.078951
11	0.053134	0.050607	0.048462	0.045799	0.041940
12	0.025372	0.024106	0.023092	0.021878	0.020109
13	0.010943	0.010376	0.009941	0.009437	0.008702
14	0.004262	0.004035	0.003866	0.003675	0.003398
15	0.001499	0.001418	0.001358	0.001292	0.001197
16	0.000476	0.000450	0.000431	0.000410	0.000380
17	0.000137	0.000129	0.000124	0.000117	0.000109
18	0.000035	0.000033	0.000032	0.000030	0.000028
19	0.000008	0.000008	0.000007	0.000007	0.000007

VALUES OF SIMULATED ECONOMIC MOBILITY AT AGE = 30

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.161569	0.172815	0.182194	0.192139	0.203792	0.215426
1	0.237713	0.256585	0.272364	0.287667	0.303246	0.316951
2	0.315936	0.343950	0.367411	0.388586	0.407319	0.421248
3	0.379292	0.416261	0.447248	0.473607	0.493856	0.505722
4	0.411297	0.454961	0.491388	0.520731	0.540333	0.548395
5	0.402826	0.451693	0.488900	0.514870	0.530826	0.537101
6	0.356313	0.407034	0.440283	0.457944	0.468439	0.475084
7	0.284619	0.323329	0.352962	0.372248	0.380757	0.379492
8	0.205295	0.232977	0.256034	0.272209	0.278063	0.273726
9	0.133700	0.152472	0.168155	0.178945	0.182249	0.178267
10	0.078610	0.090076	0.099639	0.106078	0.107737	0.104813
11	0.041722	0.048012	0.053248	0.056706	0.057452	0.055629
12	0.019986	0.023086	0.025659	0.027331	0.027632	0.026648
13	0.008639	0.010011	0.011147	0.011874	0.011983	0.011519
14	0.003370	0.003915	0.004364	0.004648	0.004685	0.004493
15	0.001186	0.001380	0.001539	0.001639	0.001650	0.001581
16	0.000376	0.000438	0.000489	0.000520	0.000524	0.000502
17	0.000108	0.000125	0.000140	0.000149	0.000150	0.000143
18	0.000028	0.000032	0.000036	0.000038	0.000038	0.000037
19	0.000006	0.000007	0.000008	0.000009	0.000009	0.000009

INCOME/TIME	1969	1970	1971	1972	1973
0	0.222751	0.221478	0.209881	0.189100	0.161569
1	0.324317	0.320879	0.304487	0.275893	0.237713
2	0.426751	0.420210	0.399260	0.363746	0.315936
3	0.507475	0.497380	0.473167	0.433353	0.379292
4	0.545490	0.532188	0.506692	0.466343	0.411297
5	0.532635	0.516365	0.488624	0.450637	0.402826
6	0.472227	0.454217	0.424370	0.391107	0.356313
7	0.370637	0.356377	0.337714	0.314217	0.284619
8	0.263916	0.253345	0.242148	0.226883	0.205295
9	0.170719	0.163327	0.156290	0.147016	0.133700
10	0.099793	0.095163	0.091121	0.086010	0.078610
11	0.052692	0.050101	0.047997	0.045444	0.041722
12	0.025130	0.023833	0.022840	0.021682	0.019986
13	0.010825	0.010243	0.009818	0.009341	0.008639
14	0.004211	0.003978	0.003813	0.003633	0.003370
15	0.001479	0.001396	0.001337	0.001276	0.001186
16	0.000469	0.000442	0.000424	0.000404	0.000376
17	0.000134	0.000127	0.000121	0.000116	0.000108
18	0.000035	0.000033	0.000031	0.000030	0.000028
19	0.000008	0.000008	0.000007	0.000007	0.000006

CHART 100



Three Time Components of Simulated Mobility

3.2 Propagation of Population Density

Program 3.2 will propagate a given age-, income-specific population density at time t (=1963 here) over a p period of years from specified mobility and mortality functions using a finite-difference scheme based on the Lax-Wendroff conservation formula [15]. While the program is used here only for aggregation experiments with the mobility function of the previous section, it could be used with estimates of the mobility obtained from actual data if it were available. Further, the program provides values of the population aggregated from the density function using the product rule for multiple integration based on Simpson's Rule [22]. Values of density and aggregated population are placed on a computer file for use in estimating $\mu(x,s,t)$.

```
$JOB   ACCT-NUM,VERNER,TIME=60
C      ****************************************************************************
C
C           PROGRAM 3.2 :   SIMMU
C
C      ****************************************************************************


3JOB SIMMU; CHARGE VERNER; CLASS 3;
USER 080002/VHJ;
PROCESSTIME=600;
PRINTLIMIT=5000;
BEGIN
3REMOVE DATA/XXGRAL FROM PACK;
COMPILE SIMMU FORTRAN;
FILE FILE2(KIND=PACK,TITLE=DATA/XXRHO2,MAXRECSIZE=11,BLOCKSIZE=330,
AREASIZE=42,AREAS= 1,FLEXIBLE,SAVEFACTOR=30);
FILE FILE3 (KIND=PACK,TITLE=DATA/XXMU9 , FILETYPE=7);
3FILE FILE4(KIND=PACK,TITLE=DATA/XXGRAL,MAXRECSIZE=20,BLOCKSIZE=60,
            AREASIZE=21,AREAS=11,FLEXIBLE,SAVEFACTOR=30);
FILE FILE8 (KIND=PACK,TITLE=DATA/XXDSCN,FILETYPE=7);
FILE FILE9 (KIND=PACK,TITLE=DATA/XXCSRS,FILETYPE=7);
OPTION=AUTORM
3DATA



3DATA



3END JOB
```

```
C       ****************************************************************
C
C       PROGRAM : 3.2
C       TITLE : SIMMU
C       PURPOSE : TO PROPAGATE A POPULATION DENSITY AND AGGREGATION
C                 FROM A GIVEN ECONOMIC MOBILITY AND STARTING DENSITY
C       DATA : DATA/XXMU9
C       OUTPUT : DATA/XXRHO2 - DENSITY
C                DATA/XXGRAL - AGGREGATED POPULATION
C
C       ****************************************************************
        REAL MU(123,40),MU5(123,40)
        REAL R(62,11),RI(123,21)
        REAL PS(123,40),P(62,20,11),PT(123,40)
        REAL K1(2,62),A1(3,62),H1(62),D1(62),COF(4,62)
        REAL FT(22)
        REAL STARTS(40)
        REAL CST(3),PI(62,20,11)
        REAL STEP
        INTEGER SA
        NAMELIST/PRTLIM/IPS,IPE
C       ****************************************************************
C
C       THIS SUBROUTINE SIMULATES THE CHANGE IN POPULATION DENSITY AS A
C       FUNCTION OF AGES (17-78) AND INCOME ($0-$19,000) OVER THE YEARS
C       1963-1973.
C
C          AN ARTIFICIAL BUT REALISTIC ECONOMIC MOBILITY WAS CREATED
C          IN PROGRAM 3.1 TO DETERMINE AT WHICH POINT IN THE
C          ESTIMATION PROCEDURE THAT ERRORS OCCURRED WHICH CAUSED
C          POOR RESULTS WHEN ESTIMATING FROM REAL DATA.
C          THIS PROGRAM RUNS A SIMULATION OF DENSITY USING THIS
C          SIMULATED MOBILITY.
C
C          NSTEP = NUMBER OF SUBINTERVALS PER BASIC INTERVAL IN
C                  PROPAGATING THE DENSITY FROM THE SIMULATED MU.
C                = SAME VALUE AS IN PROGRAM  3.1.
C
C       ****************************************************************
        READ(5,PRTLIM)
        READ(5,111) NX,NS,NT
        READ(5,112) NSTEP
        STEP=NSTEP
        NSTP1=NSTEP+1
        READ(5,111) NX1,NS1,NT1
        WRITE(6,113) NX,NS,NT
        READ(5,112) SA
        WRITE(6,114) SA
        READ(5,115) (CST(I),I=1,3),SUM
        DO 3 I=1,3
           CST(I)=CST(I)/SUM
    3      CONTINUE
  111 FORMAT(3I3)
  112 FORMAT(I3)
  113 FORMAT('1','NX = ',I3,///,' NS = ',I3,///,' NT = ',I3,///)
  114 FORMAT(//,' STARTING AGE IS ',I3,//)
  115 FORMAT(4F10.4)
      NXM1=NX-1
      NTM1=NT-1
```

```
      NXP1=NX+1
      NSP1=NS+1
      NTP1=NT+1
      NX1P1=NX1+1
      NS1P1=NS1+1
      NT1P1=NT1+1
      CALL CALCR(R,NX,NT,SA)
      DO 51 I=1,NX
51         K1(1,I)=SA+I-1.
      CALL SETUP(NX,A1,H1,K1)
      DO 53 K=1,NT
           DO 52 I=1,NX
52              K1(2,I)=R(I,K)
           CALL SOLVE(A1,K1,H1,NX,D1,COF)
           I1=0
           COUNT=0
           DO 53 I=1,NX
57              I1=I1+1
                COUNT=COUNT+1
               XX=SA+(I1-1.)/STEP
                R1(I1,K)=COF(4,I)+XX*(COF(3,I)+XX*(COF(2,I)+XX*COF(1,I))
     &)
                IF (COUNT.LT.NSTEP.AND.I1.LT.NX1) GO TO 57
                COUNT=0
53              CONTINUE
      DO 54 K=1,NT
54         K1(1,K)=K-1.
      CALL SETUP(NT,A1,H1,K1)
      DO 56 I=1,NX1
           DO 55 K=1,NT
55              K1(2,K)=R1(I,K)
           CALL SOLVE(A1,K1,H1,NT,D1,COF)
           K2=0
           COUNT=0
           DO 56 K=1,NT
58              K2=K2+1
                COUNT=COUNT+1
                TT=(K2-1.)/STEP
                R1(I,K2)=COF(4,K)+TT*(COF(3,K)+TT*(COF(2,K)+TT*COF(1,K))
     &)
                IF (COUNT.LT.NSTEP.AND.K2.LT.NT1) GO TO 58
                COUNT=0
56              CONTINUE
      CALL        INPUT(NX1,NS1P1,NT1,PS,FT,NTP1,R1,NSTEP,SA)
      I1=1-NSTEP
      DO 61 I=1,NX
           I1=I1+NSTEP
           J1=1-NSTEP
           DO 61 J=1,NS
                J1=J1+NSTEP
                P(I,J,1)=PS(I1,J1)
61              CONTINUE
      IPRT=0
      K2=1
      CALL        INTGXS(NX,NS,NT,NX1,NS1P1,1,NSTP1,PS,PI,CST)
      DO 10 K=2,NT1
           CALL INPMU(NX1,NS1P1,MU,MU5)
           DO 8 I=1,NX1
                DO 8 J=1,NS1P1
                     PT(I,J)=MU(I,J)*PS(I,J)
```

```
        8              CONTINUE
                  XX=FT(K)/FT(1)
                  CALL BIRTHS(NS1P1,K,STARTS,XX,NSTEP)
                  NX1M1=NX1-1
                  DO 11 I1=1,NX1M1
                      I=NX1-I1
                  DO 12 J=2,NS1
       12             PS(I+1,J)=PS(I,J)*(1.-R1(I,K-1)/NSTEP)-.5*
         &                           (PT(I,J+1)-PT(I,J-1)-(MU5(I,J+1)*
         &                           PT(I,J+1)+MU5(I,J)*PT(I,J-1)-(MU5(I,J+1)+
         &                           MU5(I,J))*PT(I,J)))
                      PS(I+1,1)=PS(I,1)*(1.-R1(I,K-1)/NSTEP)-.5*(PT(I,2)+
         &                     PT(I,1)-MU5(I,2)*(PT(I,2)-PT(I,1)))
                      PS(I+1,NS1P1)=PS(I,NS1P1)*(1.-R1(I,K-1)/NSTEP)+.5*
         &                          (PT(I,NS1P1)+PT(I,NS1)-MU5(I,NS1P1)*
         &                          (PT(I,NS1P1)-PT(I,NS1)))
       11                 CONTINUE
              DO 5 J=1,NS1P1
        5         PS(1,J)=STARTS(J)
                  IPRT=IPRT+1
                  IF (IPRT.NE.NSTEP) GOTO 10
                  IPRT=0
                  K2=K2+1
                  I1=1-NSTEP
                  DO 71 I=1,NX
                      I1=I1+NSTEP
                      J1=1-NSTEP
                      DO 71 J=1,NS
                          J1=J1+NSTEP
                          P(I,J,K2)=PS(I1,J1)
       71                 CONTINUE
              CALL        INTGXS(NX,NS,NT,NX1,NS1P1,K2,NSTP1,PS,PT,CST)
       10         CONTINUE
              WRITE(6,121)
              WRITE(6,120)
              DO 21 I=IPS,IPE
                  I1=SA+I-1
                  WRITE(6,116) I1,(R(I,K),K=1,6)
       21         CONTINUE
              WRITE(6,122)
              DO 22 I=IPS,IPE
                  I1=SA+I-1
                  WRITE(6,116) I1,(R(I,K),K=7,NT)
       22         CONTINUE
              DO 19 I=IPS,IPE
                  DO 14 K=1,NT
       14             FT(K)=0.
                  I1=I+SA-1
                  WRITE(6,117) I1
                  WRITE(6,118)
                  DO 16 J=1,NS
                      WRITE(6,116) J-1, (P(I,J,K),K=1,6)
*EXTENSION*    IO-3
                  DO 16 K=1,NT
       16             FT(K)=FT(K)+P(I,J,K)
                  WRITE(6,129) (FT(K),K=1,6)
                  WRITE(6,119)
                  DO 17  J=1,NS
                      WRITE(6,116) J-1, (P(I,J,K),K=7,NT)
*EXTENSION*    IO-3
```

```
 17               CONTINUE
            WRITE(6,129) (FT(K),K=7,NT)
  18        CONTINUE
 116 FORMAT(I4,5X,6F10.6)
 117 FORMAT('1',' DENSITY PROPAGATED FROM SIMULATED MOBILITY AT AGE = '
    &,I2,//)
 118 FORMAT(' INCOME/TIME 1963         1964         1965         1966         1967
    &        1968',//)
 119 FORMAT(//,' INCOME/TIME 1969         1970         1971         1972         19
    &73',//)
 120 FORMAT('  AGE/TIME   1963         1964         1965         1966         1967
    &        1968',//)
 121 FORMAT('1',///,'      DEATH RATES',///)
 122 FORMAT(//,'  AGE/TIME   1969         1970         1971         1972         19
    &73',//)
 129 FORMAT(//,'TOTALS    ',6F10.6,//)
      DO 15 I=1,NX
            DO 15 J=1,NS
                  WRITE(2)  (P(I,J,K),K=1,NT)
  15              CONTINUE
      DO 19 K=1,NT
            DO 19 I=1,NX
            WRITE(4)  (PI(I,J,K),J=1,NS)
  19              CONTINUE
      LOCK 2

      LOCK 4

      STOP
      END

      SUBROUTINE BIRTHS(NS,K,STARTS,XX,NSTEP)
      INTEGER NS,K
      REAL STARTS(NS),XX
      REAL J1,STEP
      STEP=NSTEP
C         XX = FERTILITY RATIO (1963 = TOTAL FERTILITY)
      DO 1 J=1,NS
            J1=(J-1.)/STEP
            STARTS(J)=EXP(-.05*(J1-17./4.1+.8)**2)*XX
   1        CONTINUE
      RETURN
      END

      SUBROUTINE INPUT(NX1,NS1P1,NT1,PS,FT,NTP1,R1,NSTEP,SA)
      INTEGER NX1,NS1P1,NT1,NTP1,NSTEP,SA
      REAL PS(NX1,NS1P1),FT(NT1),R1(NX1,NT1)
      REAL K1(2,43),H1(43),A1(43),D1(43),COF(4,43)
      REAL I1,J1,STEP
      STEP=NSTEP
      NT=NTP1-1
      READ(5,114) (FT(I),I=1,NTP1)
 114 FORMAT(8F10.6)
      IF (NSTEP.EQ.1) GO TO 10
      DO 9 K=1,NTP1
            K1(1,K)=K-1.
   9        CONTINUE
      CALL SETUP(NTP1,A1,H1,K1)
      DO 12 K=1,NTP1
  12        K1(2,K)=FT(K)
```

```
      CALL SOLVE(A1,K1,H1,NTP1,D1,COF)
      K2=0
      COUNT=0
      DO 13 K=1,NT
 14         K2=K2+1
            COUNT=COUNT+1
            TT=(K2-1.)/STEP
            FT(K2)=COF(4,K)+TT*(COF(3,K)+TT*(COF(2,K)+TT*COF(1,K)))
            IF (COUNT.LT.NSTEP.AND.K2.LT.NT1) GO TO 14
            COUNT=0
 13         CONTINUE
 10         CONTINUE
      WRITE(6,212)
      WRITE(6,122) (FT(I),I=1,NT1)
212 FORMAT(//,' BIRTH RATE FRACTIONS IN HALF YEARS',//)
122 FORMAT(' ', 5F12.6,/)
      SUM=0.
      DO 1 J=1,NS1P1
            J1=(J-1.)/STEP
            PS(1,J)=EXP(-.05*(J1-17./4.1+.8)**2)
            SUM=SUM+PS(1,J)
  1   CONTINUE
      SUM=SUM/STEP
      FT(1)=FT(1)*SUM
      DO 2 J=1,NS1P1
         PS(1,J)=PS(1,J)/SUM
  2         CONTINUE
      FAC=STEP
      DO 5 I=2,NX1
            I1=SA+(I-1.)/STEP
            SUM=0
            DO 4 J=1,NS1P1
                  J1=(J-1.)/STEP
                  PS(I,J)=(EXP(-.05*(J1-I1/4.1+.8)**2)*STEP/(1.+I)+
     &                          (1.-STEP/(1.+I))*SIN(EXP(-.01*((I1-61.25)**2+
     &                          (J1-10.25)**2))))
                  SUM=SUM+PS(I,J)
  4               CONTINUE
             FAC=FAC*(1.-R1(I-1,1)/STEP)
             SUM=SUM/FAC
            DO 5 J=1,NS1P1
                  PS(I,J)=PS(I,J)/SUM
  5         CONTINUE
      RETURN
      END

      SUBROUTINE INPMU(NX,NSP1,MU,MU5)
      INTEGER NX,NSP1
      REAL MU(NX,NSP1),MU5(NX,NSP1)
      NS=NSP1-1
      DO 2 I=1,NX
                  READ(3)  (MU(I,J),J=1,NS)
  1               CONTINUE
            DO 2 K=1,NT
                  MU(I,NSP1)=2.*MU(I,NS)-MU(I,NS-1)
  2         CONTINUE
      DO 4 I=1,NX
            DO 4 J=1,NS
                  MU5(I,J+1)=.5*(MU(I,J)+MU(I,J+1))
  4                     CONTINUE
```

```
      RETURN
      END
C
      SUBROUTINE CALCR(R,NX,NT,SA)
      INTEGER NX,NT,SA
      REAL R(NX,NT)
C
C     ****************************************************************
C
C     THIS SUBROUTINE CALCULATES DEATH RATES FOR THE YEARS
C     1963 TO 1963+NT-1
C     R(I,J) = FRACTION OF PEOPLE OF AGE  I+SA  DYING AT TIME  J+1962
C
C     ****************************************************************
      REAL XY(20),XZ(20)
      INTEGER YY,D
      INTEGER TEST/'R'/
      I1=SA+203
      DO 202 I=1,I1
  202 READ(8,100) XX
  100 FORMAT(A1)
      I1=SA+631
      DO 203 I=1,I1
  203 READ(9,100) XX
      DO 206 I=1,20
            XY(I)=0.
  206       CONTINUE
C     READ DATA FOR EACH YEAR
      DO 200 LT=1,NT
      DO 221 LX=1,NX
C     READ DEATHS FOR EACH YEAR
      READ(8,101) D
  101 FORMAT(T29,I5)
      IF (LX.LE.70-SA) GOTO 210
C     ADJUSTMENT TO ACCOMMODATE LACK OF POPULATION DATA ABOVE AGE
C     69  FOR YEARS  1967 TO 1970.
      IF (LT.GT.4.AND.LT.LE.8) GOTO 211
C     READ POPULATION FOR EACH YEAR
      READ(9,102) P,YY
  102 FORMAT(12X,F9.1,55X,A1)
C     POPULATION IN NON-CENSUS YEARS MUST BE MULTIPLIED BY  1000.
      IF (YY.EQ.TEST) P=P*1000.
C     LINEAR EXTRAPOLATION FROM 1966 DATA IS USED FOR 1967 TO 1970.
      J=LX-51
      XZ(J)=P-XY(J)
      XY(J)=P
      GOTO 201
  211 J=LX-51
      P=XY(J)+XZ(J)
      XY(J)=P
      GOTO 201
  210 READ(9,102) P,YY
      IF (YY.EQ.TEST) P=P*1000.
  201 R(LX,LT)=D/P
  221 CONTINUE
      IF (LT.EQ.NT) GOTO 200
C     CLEAR REMAINING DATA ON DEATHS.
      NN=101-NX
      IF (LT.GT.2) NN=NN-1
```

```
      DO 204 I=1,NN
  204 READ(8,100) XX
C     CLEAR REMAINING DATA ON POPULATION.
      NM=90-NX
      IF (LT.GT.4.AND.LT.LE.8) NM=SA
      DO 205 I=1,NM
  205 READ(9,100) XX
  200 CONTINUE
      RETURN
      END

      SUBROUTINE INTGXS(NX,NS,NT,NX1,NS1P1,K,NSTP1,PS,PI,CST)
      INTEGER NX,NS,NT,NX1,NS1P1,K,NSTP1
      REAL PS(NX1,NS1P1),PI(NX,NS,NT),CST(NSTP1)
      NSTEP=NSTP1-1
      NXM1=NX-1
      NXM2=NX-2
      NSM1=NS-1
      NSM2=NS-2
      DO 5 I=1,NX
      DO 5 J=1,NS
    5          PI(I,J,K)=0.
      DO 10 I1=1,NSTP1
           DO 10 J1=1,NSTP1
                C=CST(I1)*CST(J1)
                DO 10 I=1,NXM1
                     I2=(I-1)*NSTEP+I1
                     DO 10 J=1,NSM1
                          J2=(J-1)*NSTEP+J1
                          PI(I,J,K)=PI(I,J,K)+C*PS(I2,J2)
                          IF (PI(I,J,K).LT.0.) PI(I,J,K)=0.
   10                     CONTINUE
      DO 20 I=1,NXM1
   20      PI(I,NS,K)=2.*PI(I,NSM1,K)-PI(I,NSM2,K)
      DO 30 J=1,NS
   30      PI(NX,J,K)=2.*PI(NXM1,J,K)-PI(NXM2,J,K)
      RETURN
      END

      SUBROUTINE SETUP(N,A,H,K)
C
C     ****************************************************************
C     THIS ROUTINE SETS UP A TRIDIAGONAL MATRIX A OF THE SPLINE EQUATION
C     IN A  3 BY N  ARRAY   -   FIRST ROW IS SUPERDIAGONAL
C                               SECOND ROW IS DIAGONAL
C                               THIRD ROW IS SUBDIAGONAL
C     AND THEN DECOMPOSES A TO  LU  SO THAT
C        FIRST ROW IS SUPERDIAGONAL OF U, SECOND ROW IS DIAGONAL OF U
C       THIRD ROW IS SUBDIAGONAL OF L, DIAGONAL OF L IS UNITY
C     ****************************************************************
C
      REAL A(3,N),H(N),K(2,N)
      NM1=N-1
      H(2)=K(1,2)-K(1,1)
      DO 10 I=2,NM1
           H(I+1)=K(1,I+1)-K(1,I)
           A(1,I)=H(I+1)/(H(I+1)+H(I))
           A(2,I)=2.0
           A(3,I)=1-A(1,I)
   10      CONTINUE
```

```
      A(1,1) = A(1,2)
      A(2,1) = A(2,2)
      A(3,1) = A(3,2)
      A(1,N) = A(1,N-1)
      A(2,N) = A(2,N-1)
      A(3,N) = A(3,N-1)
      A(1,2) = -2
      A(2,2) = 2
      A(3,2) = 0
      A(1,N-1) = 0
      A(2,N-1) = 2
      A(3,N-1) = -2
      LL = N - 1
      DO 11 I = 3, LL
           A(3,I)=A(3,I)/A(2,I-1)
           A(2,I)=A(2,I)-A(3,I)*A(1,I-1)
   11      CONTINUE
      RETURN
      END

      SUBROUTINE SOLVE(A,K,H,N,D,COF)
C
C     *******************************************************************
C     THIS ROUTINE CALCULATES THE SECOND ORDER FINITE DIFFERENCES OF THE
C     SPLINE, AND THEN SOLVES  AM=D  , (BY FORWARD AND BACKWARD
C     SUBSTITUTION), PLACING M(VECTOR OF SECOND DERIVATIVES) IN D
C     END POINT CONDITIONS NOW USE THIRD ORDER FINITE DIFFERENCES
C     TO ESTIMATE THIRD ORDER DERIVATIVES AT X0+H/2 AND XN+H/2
C     *******************************************************************
C
      REAL H(N),D(N),A(3,N),K(2,N),COF(4,N)
      D(2)=(K(2,2)-K(2,1))/H(2)
      NM1=N-1
      DO 12 I=2,NM1
           D(I+1)=(K(2,I+1)-K(2,I))/H(I+1)
           D(I)=6*(D(I+1)-D(I))/(H(I+1)+H(I))
   12      CONTINUE
      D(1) =-2.*(-K(2,1)+K(2,4)+3*(K(2,2)-K(2,3)))/(H(2)*H(2))
      D(N)=-2.*(-K(2,N)+K(2,N-3)+3.*(K(2,N-1)-K(2,N-2)))/(H(N-1)**2)
      T1 = D(2)
      T2 = D(N-1)
      D(2) = D(1)
      D(N-1) = D(N)
      LL = N - 1
      DO 13 I = 3, LL
           D(I)=D(I)-A(3,I)*D(I-1)
   13      CONTINUE
      D(N-1) = D(N-1) / A(2,N-1)
      DO 14 I = 3,LL
           J=N+1-I
           D(J)=(D(J)-A(1,J)*D(J+1))/A(2,J)
   14      CONTINUE
      D(1) = ( T1 - A(2,1) * D(2) - A(1,1) * D(3) ) / A(3,1)
      D(N ) = ( T2 - A(2,N) * D(LL) - A(3,N) * D(N-2) ) / A(1,N)
      CALL POLLY(N,D,K,H,COF)
      RETURN
      END

      SUBROUTINE POLLY(N,M,K,H,COF)
C
```

```
C        ****************************************************************
C        THIS ROUTINE COMPUTES THE COEFFICIENTS OF THE SPLINE POLYNOMIAL
C          ON EACH SUBINTERVAL
C        K IS THE ARRAY OF DATA POINTS
C        H IS THE VECTOR OF SUBINTERVAL LENGTHS
C        M IS THE SOLUTION VECTOR TO THE EQUATION AM=D
C        ****************************************************************
C
         REAL M(N),K(2,N),H(N),COF(4,N)
         NM1=N-1
         DO 11 I=1,NM1
         COF(1,I)=(M(I+1)-M(I))/(6.*H(I+1))
         COF(2,I)=(K(1,I+1)*M(I)-K(1,I)*M(I+1))/(2.*H(I+1))
         DD=M(I+1)*K(1,I)*K(1,I)-M(I)*K(1,I+1)*K(1,I+1)
        &+2.*K(2,I+1)-2.*K(2,I)
         COF(3,I)=(DD/(2.*H(I+1)))+H(I+1)*(M(I)-M(I+1))/6.
         DD=M(I)*(K(1,I+1)**3)-M(I+1)*(K(1,I)**3)+6.*K(1,I+1)*K(2,I)
        &-6.*K(1,I)*K(2,I+1)+K(1,I)*M(I+1)*(H(I+1)**2)-K(1,I+1)*M(I)*
        &(H(I+1)**2)
         COF(4,I)=DD/(H(I+1)*6.)
   11    CONTINUE
         DO 16 J=1,4
   16    COF(J,N)=COF(J,NM1)
         RETURN
            END
```

```
C       ***********************************************************************
C
C       DATA FOR
C       PROGRAM 3.2 : STMMU
C
C       ***********************************************************************
 &PRTLIM IPS=10, IPC=32, &END
 62 20 11
  2
123 39 21
 17
    1.          4.          1.          6.
  1.000000    .99885      .997693     .996588     .995288     .99422      .993154     .992034
   .991098    .990059     .988883     .977777
```

NX = 62

NS = 20

NT = 11

STARTING AGE IS 17

BIRTH RATE FRACTIONS IN HALF YEARS

1.000000	0.999458	0.998850	0.998255	0.997693
0.997159	0.996588	0.995936	0.995288	0.994730
0.994220	0.993697	0.993154	0.992604	0.992084
0.991612	0.991098	0.990498	0.990058	0.989842
0.988883				

DEATH RATES

AGE/TIME	1963	1964	1965	1966	1967	1968
26	0.001107	0.001003	0.001122	0.001003	0.001039	0.000982
27	0.001088	0.000947	0.001110	0.001069	0.001062	0.001198
28	0.001078	0.001000	0.000976	0.001166	0.001098	0.001112
29	0.001159	0.001140	0.001234	0.001049	0.001096	0.001057
30	0.001015	0.001115	0.001050	0.001300	0.001049	0.001106
31	0.001156	0.001269	0.001166	0.001170	0.001181	0.001146
32	0.001221	0.001325	0.001204	0.001179	0.001169	0.001250
33	0.001392	0.001272	0.001416	0.001466	0.001194	0.001418
34	0.001449	0.001483	0.001457	0.001406	0.001507	0.001333
35	0.001502	0.001494	0.001615	0.001449	0.001462	0.001560
36	0.001722	0.001608	0.001490	0.001570	0.001526	0.001595
37	0.001810	0.001770	0.001776	0.001851	0.001838	0.001643
38	0.001882	0.001749	0.001984	0.002031	0.001787	0.002015
39	0.002065	0.001992	0.002041	0.001923	0.002005	0.002116
40	0.002093	0.002256	0.002126	0.002341	0.002341	0.002285
41	0.002533	0.002620	0.002535	0.002447	0.002722	0.002377
42	0.003055	0.003117	0.002846	0.002744	0.002602	0.002769
43	0.003265	0.003314	0.003267	0.003418	0.003237	0.002983
44	0.003225	0.003453	0.003571	0.003115	0.003645	0.003449
45	0.003634	0.003483	0.003934	0.003877	0.003686	0.003791
46	0.003797	0.004186	0.004034	0.004684	0.004639	0.004274
47	0.004526	0.004395	0.004180	0.004591	0.004669	0.004735
48	0.005380	0.005119	0.004754	0.005218	0.004907	0.005209

AGE/TIME	1969	1970	1971	1972	1973
26	0.001098	0.001023	0.001061	0.001077	0.001262
27	0.001223	0.001022	0.001070	0.001086	0.001114
28	0.001082	0.001178	0.001119	0.001216	0.001059
29	0.001161	0.001112	0.001102	0.001215	0.001061
30	0.001225	0.001191	0.001178	0.001365	0.001191
31	0.001180	0.001260	0.001267	0.001289	0.001259
32	0.001228	0.001242	0.001348	0.001172	0.001219
33	0.001284	0.001405	0.001295	0.001349	0.001345
34	0.001335	0.001371	0.001378	0.001462	0.001354
35	0.001561	0.001422	0.001378	0.001546	0.001605
36	0.001672	0.001437	0.001642	0.001543	0.001675
37	0.001690	0.001856	0.001563	0.001733	0.001888
38	0.002002	0.001904	0.001861	0.001957	0.001911
39	0.002110	0.002130	0.002363	0.002177	0.002048
40	0.002067	0.002321	0.002377	0.002239	0.002282
41	0.002503	0.002521	0.002653	0.002699	0.002484
42	0.002819	0.002800	0.002793	0.002807	0.002921
43	0.003065	0.002977	0.003128	0.003119	0.003142
44	0.003237	0.003388	0.003352	0.003418	0.003093
45	0.003854	0.003528	0.003614	0.003436	0.003562
46	0.004069	0.004054	0.004075	0.004085	0.003951
47	0.004862	0.004348	0.004353	0.004255	0.004590
48	0.005274	0.005149	0.004756	0.004782	0.004978

DENSITY PROPAGATED FROM SIMULATED MOBILITY AT AGE = 26

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.027818	0.020878	0.014310	0.008526	0.003720	-0.000029
1	0.046054	0.043658	0.041340	0.038936	0.036236	0.033095
2	0.068989	0.063838	0.058907	0.054576	0.051023	0.048252
3	0.093512	0.087124	0.080205	0.073535	0.067723	0.063139
4	0.114690	0.110737	0.104906	0.097825	0.090795	0.084601
5	0.127279	0.129211	0.128326	0.125531	0.121500	0.116412
6	0.127808	0.135958	0.143926	0.148025	0.151232	0.153040
7	0.116127	0.126224	0.138360	0.150759	0.161859	0.171722
8	0.095473	0.102308	0.111207	0.122022	0.134177	0.146707
9	0.071023	0.073149	0.076365	0.080607	0.085634	0.090931
10	0.047808	0.046713	0.046044	0.045704	0.045530	0.045282
11	0.029119	0.026971	0.025067	0.023345	0.021736	0.020160
12	0.016049	0.014179	0.012520	0.011040	0.009708	0.008496
13	0.008005	0.006798	0.005758	0.004861	0.004087	0.003420
14	0.003614	0.002969	0.002430	0.001981	0.001608	0.001299
15	0.001477	0.001179	0.000937	0.000742	0.000584	0.000458
16	0.000548	0.000425	0.000329	0.000253	0.000194	0.000148
17	0.000185	0.000139	0.000105	0.000079	0.000059	0.000044
18	0.000058	0.000042	0.000031	0.000022	0.000016	0.000012
19	0.000017	0.000012	0.000008	0.000006	0.000004	0.000003
OTALS	0.995653	0.992514	0.990081	0.988374	0.987426	0.987192

INCOME/TIME	1969	1970	1971	1972	1973
0	-0.002704	-0.004392	-0.005329	-0.005844	-0.005894
1	0.029493	0.025693	0.022146	0.019408	0.019453
2	0.046113	0.044492	0.043336	0.042857	0.042871
3	0.059855	0.057926	0.056890	0.057039	0.057075
4	0.079775	0.076563	0.074842	0.074535	0.074631
5	0.110879	0.106405	0.104014	0.103678	0.104302
6	0.153130	0.151725	0.150070	0.149648	0.150546
7	0.181049	0.189558	0.195624	0.198680	0.197600
8	0.158599	0.169303	0.178097	0.184031	0.182668
9	0.095807	0.099802	0.102494	0.103404	0.102429
10	0.044705	0.043709	0.042239	0.040245	0.039910
11	0.018557	0.016937	0.015311	0.013695	0.013610
12	0.007382	0.006370	0.005455	0.004635	0.004614
13	0.002844	0.002351	0.001933	0.001579	0.001574
14	0.001043	0.000834	0.000664	0.000525	0.000524
15	0.000357	0.000278	0.000215	0.000165	0.000165
16	0.000113	0.000085	0.000064	0.000048	0.000048
17	0.000032	0.000024	0.000018	0.000013	0.000013
18	0.000009	0.000006	0.000004	0.000003	0.000003
19	0.000002	0.000001	0.000001	0.000001	0.000001
OTALS	0.987041	0.987570	0.988087	0.988345	0.986141

DENSITY PROPAGATED FROM SIMULATED MOBILITY AT AGE = 27

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.024069	0.017752	0.011776	0.006554	0.002290	-0.000915
1	0.040831	0.038433	0.036081	0.033601	0.030763	0.027409
2	0.062674	0.057486	0.052499	0.048098	0.044438	0.041483
3	0.087049	0.080365	0.073132	0.066171	0.060093	0.055247
4	0.109399	0.104816	0.098119	0.090335	0.082547	0.075676
5	0.124404	0.125687	0.123809	0.119747	0.114264	0.107657
6	0.128006	0.136028	0.142604	0.146662	0.148522	0.148585
7	0.119178	0.129860	0.142661	0.155624	0.166968	0.176652
8	0.100402	0.108096	0.118162	0.130533	0.144653	0.159412
9	0.076535	0.079281	0.083348	0.088749	0.095305	0.102466
10	0.052791	0.051892	0.051509	0.051560	0.051886	0.052204
11	0.032950	0.030697	0.028713	0.026936	0.025288	0.023668
12	0.019611	0.016533	0.014683	0.013028	0.011534	0.010163
13	0.009514	0.008122	0.006916	0.005870	0.004964	0.004177
14	0.004403	0.003636	0.002991	0.002450	0.001999	0.001622
15	0.001847	0.001480	0.001182	0.000940	0.000744	0.000586
16	0.000703	0.000548	0.000425	0.000329	0.000253	0.000194
17	0.000245	0.000185	0.000139	0.000105	0.000079	0.000059
18	0.000080	0.000058	0.000042	0.000030	0.000022	0.000016
19	0.000026	0.000017	0.000012	0.000008	0.000006	0.000004
OTALS	0.993716	0.990973	0.988802	0.987331	0.986618	0.986366

INCOME/TIME	1969	1970	1971	1972	1973
0	-0.003042	-0.004209	-0.004698	-0.004847	-0.004956
1	0.023583	0.019562	0.015843	0.012889	0.011054
2	0.039097	0.037083	0.035438	0.034334	0.034186
3	0.051737	0.049415	0.048156	0.047863	0.048684
4	0.070349	0.066678	0.064589	0.063887	0.064610
5	0.100851	0.095329	0.092199	0.091237	0.091757
6	0.146717	0.143199	0.139841	0.138210	0.138696
7	0.185575	0.193295	0.198558	0.200957	0.202126
8	0.173716	0.186665	0.197720	0.205856	0.210084
9	0.109487	0.115615	0.120428	0.123274	0.123341
10	0.052224	0.051726	0.050680	0.048975	0.046468
11	0.022007	0.020277	0.018517	0.016733	0.014929
12	0.008899	0.007730	0.006668	0.005706	0.004840
13	0.003495	0.002905	0.002401	0.001972	0.001610
14	0.001310	0.001052	0.000841	0.000669	0.000529
15	0.000459	0.000358	0.000278	0.000215	0.000166
16	0.000149	0.000113	0.000085	0.000064	0.000048
17	0.000044	0.000032	0.000024	0.000018	0.000013
18	0.000012	0.000008	0.000006	0.000004	0.000003
19	0.000003	0.000002	0.000001	0.000001	0.000001
OTALS	0.986672	0.986837	0.987576	0.988019	0.988190

DENSITY PROPAGATED FROM SIMULATED MOBILITY AT AGE = 28

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.020720	0.015074	0.009722	0.005064	0.001327	-0.001379
1	0.036014	0.033696	0.031403	0.028940	0.026089	0.022700
2	0.056643	0.051553	0.046649	0.042287	0.038633	0.035631
3	0.080613	0.073804	0.066451	0.059364	0.053191	0.048266
4	0.103811	0.098719	0.091453	0.083062	0.074743	0.067467
5	0.120964	0.121571	0.118774	0.113531	0.106750	0.098904
6	0.127539	0.135247	0.141199	0.144142	0.144487	0.142732
7	0.121677	0.132717	0.145944	0.159114	0.170249	0.179253
8	0.105039	0.113453	0.124562	0.138276	0.154063	0.170723
9	0.082050	0.085377	0.090296	0.096835	0.104902	0.113972
10	0.057996	0.057297	0.057236	0.057710	0.058585	0.059555
11	0.037096	0.034739	0.032693	0.030872	0.029200	0.027561
12	0.021474	0.019173	0.017127	0.015285	0.013616	0.012080
13	0.011253	0.009654	0.008264	0.007052	0.005996	0.005074
14	0.005340	0.004430	0.003662	0.003015	0.002472	0.002017
15	0.002299	0.001850	0.001484	0.001185	0.000942	0.000746
16	0.000901	0.000703	0.000548	0.000425	0.000329	0.000253
17	0.000325	0.000245	0.000185	0.000139	0.000105	0.000079
18	0.000111	0.000079	0.000057	0.000042	0.000030	0.000022
19	0.000039	0.000025	0.000017	0.000012	0.000008	0.000006
OTALS	0.991901	0.989406	0.987726	0.986353	0.985717	0.985662

INCOME/TIME	1969	1970	1971	1972	1973
0	-0.003039	-0.003796	-0.003944	-0.003823	-0.003703
1	0.018822	0.014785	0.011066	0.008102	0.006120
2	0.033086	0.030815	0.028758	0.027147	0.026357
3	0.044627	0.042138	0.040570	0.039872	0.040143
4	0.061792	0.057849	0.055421	0.054371	0.054656
5	0.091027	0.084744	0.080929	0.079400	0.079387
6	0.138765	0.133251	0.128083	0.125168	0.124786
7	0.187026	0.193411	0.197041	0.198080	0.198429
8	0.186951	0.201926	0.214906	0.224838	0.231021
9	0.123192	0.131806	0.139083	0.144355	0.146519
10	0.060261	0.060479	0.060046	0.058862	0.056671
11	0.025855	0.024064	0.022191	0.020266	0.018273
12	0.010648	0.009320	0.008095	0.006979	0.005963
13	0.004269	0.003570	0.002966	0.002451	0.002011
14	0.001637	0.001322	0.001061	0.000848	0.000674
15	0.000588	0.000461	0.000359	0.000279	0.000216
16	0.000194	0.000148	0.000113	0.000085	0.000064
17	0.000059	0.000044	0.000032	0.000024	0.000018
18	0.000016	0.000012	0.000008	0.000006	0.000004
19	0.000004	0.000003	0.000002	0.000001	0.000001
OTALS	0.985778	0.986350	0.986688	0.987311	0.987613

DENSITY PROPAGATED FROM SIMULATED MOBILITY AT AGE = 29

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.017745	0.012795	0.008066	0.003959	0.000706	-0.001567
1	0.031599	0.029437	0.027258	0.024908	0.022143	0.018845
2	0.050921	0.046065	0.041337	0.037135	0.033572	0.030612
3	0.074254	0.067512	0.060191	0.053173	0.047042	0.042159
4	0.097980	0.092559	0.084893	0.076164	0.067532	0.060055
5	0.116987	0.116974	0.113324	0.107131	0.099273	0.090467
6	0.126391	0.133666	0.138937	0.140679	0.139479	0.135944
7	0.123561	0.134760	0.148088	0.161220	0.171772	0.179672
8	0.109302	0.118297	0.130202	0.145058	0.162150	0.180257
9	0.087492	0.091358	0.097037	0.104693	0.114187	0.125085
10	0.063375	0.062884	0.063134	0.064082	0.065528	0.067198
11	0.041545	0.039090	0.036980	0.035145	0.033460	0.031819
12	0.024651	0.022117	0.019857	0.017828	0.015974	0.014261
13	0.013244	0.011416	0.009821	0.008427	0.007203	0.006129
14	0.006449	0.005372	0.004462	0.003692	0.003042	0.002494
15	0.002852	0.002303	0.001854	0.001488	0.001188	0.000945
16	0.001152	0.000901	0.000703	0.000548	0.000426	0.000329
17	0.000432	0.000324	0.000245	0.000185	0.000139	0.000105
18	0.000156	0.000111	0.000079	0.000057	0.000042	0.000030
19	0.000060	0.000039	0.000025	0.000017	0.000012	0.000008
OTALS	0.990147	0.987980	0.986392	0.985593	0.984868	0.984847

INCOME/TIME	1969	1970	1971	1972	1973
0	-0.002853	-0.003303	-0.003216	-0.002913	-0.002638
1	0.015080	0.011174	0.007612	0.004773	0.002815
2	0.028033	0.025601	0.023284	0.021286	0.020001
3	0.038527	0.035931	0.034153	0.033081	0.032850
4	0.054251	0.050131	0.047494	0.046114	0.045988
5	0.081872	0.075014	0.070729	0.068666	0.068147
6	0.130102	0.122715	0.115998	0.111782	0.110468
7	0.185904	0.190307	0.191889	0.190904	0.189807
8	0.197959	0.214255	0.228500	0.239767	0.247441
9	0.136519	0.147605	0.157655	0.165676	0.170450
10	0.068696	0.069724	0.070142	0.069708	0.068109
11	0.030104	0.028266	0.026330	0.024292	0.022144
12	0.012656	0.011151	0.009757	0.008471	0.007293
13	0.005186	0.004360	0.003645	0.003027	0.002499
14	0.002035	0.001651	0.001332	0.001070	0.000854
15	0.000748	0.000589	0.000462	0.000360	0.000280
16	0.000254	0.000194	0.000148	0.000113	0.000085
17	0.000078	0.000059	0.000044	0.000032	0.000024
18	0.000022	0.000016	0.000012	0.000008	0.000006
19	0.000006	0.000004	0.000003	0.000002	0.000001
OTALS	0.985177	0.985445	0.985971	0.986219	0.986626

DENSITY PROPAGATED FROM SIMULATED MOBILITY AT AGE = 30

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.015121	0.010861	0.006742	0.003152	0.000333	-0.001583
1	0.027580	0.025626	0.023622	0.021429	0.018847	0.015732
2	0.045530	0.041017	0.036570	0.032581	0.029208	0.026352
3	0.068024	0.061525	0.054414	0.047572	0.041640	0.036884
4	0.091968	0.086405	0.078577	0.069685	0.061010	0.053494
5	0.112516	0.111970	0.107655	0.100672	0.092098	0.082610
6	0.124561	0.131315	0.135688	0.136383	0.133868	0.128716
7	0.124780	0.135938	0.149127	0.161853	0.171708	0.178239
8	0.113111	0.122520	0.135003	0.150602	0.168727	0.187786
9	0.092785	0.097118	0.103472	0.112061	0.122923	0.135469
10	0.068880	0.068588	0.069148	0.070528	0.072606	0.074983
11	0.046281	0.043731	0.041564	0.039703	0.038050	0.036418
12	0.028153	0.025374	0.022891	0.020656	0.018622	0.016720
13	0.015514	0.013432	0.011610	0.010010	0.008606	0.007361
14	0.007756	0.006486	0.005410	0.004497	0.003725	0.003069
15	0.003529	0.002857	0.002308	0.001859	0.001492	0.001192
16	0.001474	0.001152	0.000901	0.000703	0.000548	0.000426
17	0.000578	0.000431	0.000324	0.000244	0.000184	0.000139
18	0.000224	0.000156	0.000111	0.000079	0.000057	0.000042
19	0.000096	0.000060	0.000039	0.000025	0.000017	0.000012
OTALS	0.988461	0.986560	0.985177	0.984294	0.984270	0.984058

INCOME/TIME	1969	1970	1971	1972	1973
0	-0.002583	-0.002822	-0.002589	-0.002182	-0.001816
1	0.012182	0.008523	0.005211	0.002592	0.000761
2	0.023815	0.021353	0.018904	0.016682	0.015043
3	0.033328	0.030728	0.028812	0.027466	0.026789
4	0.047686	0.043539	0.040768	0.039153	0.038632
5	0.073556	0.066400	0.061785	0.059332	0.058323
6	0.121252	0.112412	0.104408	0.099133	0.096864
7	0.182634	0.184888	0.184103	0.180905	0.177839
8	0.206370	0.223432	0.238239	0.250120	0.258489
9	0.148933	0.162425	0.175171	0.186186	0.193881
10	0.077302	0.079248	0.080636	0.081217	0.080487
11	0.034707	0.032862	0.030888	0.028794	0.026528
12	0.014929	0.013242	0.011665	0.010204	0.008849
13	0.006262	0.005294	0.004450	0.003718	0.003086
14	0.002516	0.002052	0.001664	0.001343	0.001078
15	0.000948	0.000750	0.000591	0.000463	0.000361
16	0.000329	0.000254	0.000194	0.000148	0.000113
17	0.000105	0.000078	0.000059	0.000044	0.000032
18	0.000030	0.000022	0.000016	0.000012	0.000008
19	0.000008	0.000006	0.000004	0.000003	0.000002
OTALS	0.984308	0.984686	0.984979	0.985332	0.985349

3.3 Estimation of Mobility

The procedures used for estimating the mobility from a population density function parallel those of section 2.2. Briefly, from a density function $\rho(x,s,t)$,

1. $\rho(x,s,t)$ is splined (cubic spline) in the x variable for each s and t to obtain $\left[\frac{\partial \rho}{\partial x}\right]_3$;

2. $\rho(x,s,t)$ is splined (cubic spline) in the t variable for each x and s to obtain $\left[\frac{\partial \rho}{\partial t}\right]_3$;

3. $\rho(x,s,t)$ is multiplied by the mortality $r(x,t)$;

4. the sum $\left[\frac{\partial \rho}{\partial t}\right]_3 + \left[\frac{\partial \rho}{\partial x}\right]_3 + \rho(x,s,t)\ r(x,t)$ is splined (cubic spline) in the s variable, and integrated exactly on $(s,\overline{s})$ for each s ;

5. these values are divided by the density of $\rho(x,s,t)$ in each case.

To estimate $\mu(x,s,t)$ when only aggregated values of population are known requires estimation of the density at the outset. The procedure described in the Interim Report which computes a fourth-degree spline whose piecewise integrals are equal to values of aggregated population seems to provide reasonable estimates <u>when the level of aggregation is low</u>.

In this case, the numbers of individuals P(I,J,R) between ages XX(i) and XX(i+1) earning incomes between ss(j) and ss(j+1) at time TT(k) are available. These are used to estimate the density as follows:

1. P(x,s,t) is splined by quartic splines for histogram data in the x variable to give $P_4(x,s,t)$,

2. $P_4(x,s,t)$ is splined by quartic splines for histogram data is the s variable to give $P_{4,4}(x,s,t)$.

3. $\rho(x,s,t)$ is replaced by $P_{4,4}(x,s,t)$ and the previous estimation procedure for $\mu(x,s,t)$ is used.

Program 3.3 estimates $\mu(x,s,t)$ directly from a density function obtained in the aggregation experiments was used.

Program 3.5 estimates $\mu(x,s,t)$ form values of population aggregated according to values of age and income provided as data. This program may be used either with actual population data, or with values obtained from aggregation experiments.

```
$JOB    ACCT-NUM,VERNER,TIME=60
C       ************************************************************************
C
C       PROGRAM 3.3 : ESTMU
C
C       ************************************************************************
```

```
3JOB ESTMU; CHARGE VERNER; CLASS 3
2USER 080002/VHJ
3PRINTLIMIT=100000;
3PROCESSTIME=600;
3BEGIN
3COMPILE ESTMU FORTRAN
3FILE FILE1 (KIND=READER,MAXRECSIZE=80,UNITS=CHARACTERS,BLOCKSIZE=80);
FILE FILE2(KIND=PACK,TITLE=DATA/XXMU7,FILETYPE=7);
FILE FILE3(KIND=PACK,TITLE=DATA/XXRHO2,FILETYPE=7);
2FILE FILE8 (KIND=PACK,TITLE=DATA/XXDSCN,FILETYPE=7);
2FILE FILE9 (KIND=PACK,TITLE=DATA/XXCSRS,FILETYPE=7);
DATA



DATA FILE1



END JOB
```

```
C       ****************************************************************
C
C       PROGRAM : 3.3
C       TITLE : ESTMU
C       PURPOSE : TO ESTIMATE MU FROM A (SIMULATED) DENSITY
C       DATA : DATA/XXRHO2
C               DATA/XXDSCN,DATA/XXCSRS - FOR ESTIMATING DEATH RATES
C       OUTPUT : DATA/XXMU7
C
C       ****************************************************************
C       ****************************************************************
C
C       THIS PROGRAM IS USED TO ESTIMATE THE ECONOMIC MOBILITY
C       PARAMETER MU.  AGE- TIME- AND INCOME-SPECIFIC VALUES OF THE
C       POPULATION ARE REQUIRED.
C
C       REAL XL(XE),SL(SE),TT(TM),MU(XE,SE,TM),P1(XE,SE,TM)
C       REAL R(XE,TM)
C
C       ****************************************************************
        INTEGER XE,SE,TM,SA,TMM1
        REAL XL(62),SL(20),TT(11),MU(62,20,11),P1(62,20,11)
        REAL R(62,11)
        REAL K(2,62),H(62),COF(4,62),A(3,62),D(62)
        LOGICAL PRTR,PRTPX,PRTPXT,PRTXTR,PRTMU
        NAMELIST/VALUES/PRTR,PRTPX,PRTPXT,PRTXTR,PRTMU
        NAMELIST/PRTLIM/IPS,IPE
        READ(1,VALUES)
        READ(1,PRTLIM)
        READ(1,101) XE,SE,TM
        READ(1,101) SA
        WRITE(6,202) XE,SE,TM
        WRITE(6,203) SA
    101 FORMAT(3I3)
    202 FORMAT(///,' NUMBER OF AGE DENSITIES = ',I2,//,' NUMBER OF INCOME
       &DENSITIES = ',I2,//,' NUMBER OF YEARS = ',I2,//)
    203 FORMAT(///,' STARTING AGE = ',I2,//)
        XMP1-XM+1
        SMP1-SM+1
        TMM1-TM-1
        TMP1-TM+1
        DO 2 I=1,XE
      2     XL(I)=SA+I-1.
        DO 3 I=1,SE
      3     SL(I)=I-1.
            DO 4 I=1,TM
      4     TT(I)=I
C       READREADREADREADREADREADREADREADREADREADREADREADREADREADREADREADR
C
        DO 66 IX=1,XE
        DO 66 IS=1,SE
                READ(3) (P1(IX,IS,IT),IT=1,TM)
     66         CONTINUE
C
C       READREADREADREADREADREADREADREADREADREADREADREADREADREADREADREADR
C       ****************************************************************
C
C       NOW OBTAIN DP/DX
C
```

```
C     ******************************************************************
      DO 60 IX=1,XF
      K(1,IX)=XL(IX)
   60 CONTINUE
      CALL SETUP(XF,A,H,K)
      DO 61 IS=1,SE
      DO 61 IT=1,TM
      DO 62 IX=1,XF
      K(2,IX)=P1(IX,IS,IT)
   62 CONTINUE
      CALL SOLVE(A,K,H,XF,D,COF)
      DO 63 IX=1,XF
      X=XL(IX)
   63 MU(IX,IS,IT)=COF(3,IX)+X*(2.*COF(2,IX)+X*3.*COF(1,IX))
   61 CONTINUE
C     PRINTPRINTPRINTIRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
C
      IF(.NOT.PRTPX) GO TO 72
      WRITE(6,122)
      DO 22 I=IPS,IPF
            I1=I+SA-1
            WRITE(6,117) I1
            WRITE(6,118)
            DO 26 J=1,SE
                  WRITE(6,116) J-1,(MU(I,J,IT),IT=1,6)
*EXTENSION*    IO-3
   26             CONTINUE
            WRITE(6,119)
            DO 22 J=1,SE
                  WRITE(6,116) J-1,(MU(I,J,IT),IT=7,TM)
*EXTENSION*    IO-3
   22             CONTINUE
C
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
   72 CONTINUE
C     ******************************************************************
C
C        SPLINE  P(X,S,T)  AGAINST  T  AND DIFFERENTIATE THE SPLINE
C          MU(X,S,T)  IS  DP/DX+DP/DT
C
C     ******************************************************************
      DO 30 IT=1,TM
      K(1,IT)=TT(IT)
   30 CONTINUE
      CALL SETUP(TM,A,H,K)
      DO 31 IX=1,XF
      DO 31 IS=1,SE
      DO 32 IT=1,TM
      K(2,IT)=P1(IX,IS,IT)
   32 CONTINUE
      CALL SOLVE(A,K,H,TM,D,COF)
      DO 33 IT=1,TM
      T=TT(IT)
   33 MU(IX,IS,IT)=MU(IX,IS,IT)+(COF(3,IT)+T*(2.*COF(2,IT)+T*3.*COF(1,I
     1))))
   31 CONTINUE
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
C
      IF(.NOT.PRTPXT) GO TO 73
      WRITE(6,123)
```

```
              DO 73 I=IPS,IPE
                    I1=I+SA-1
                    WRITE(6,117) I1
                    WRITE(6,118)
                    DO 27 J=1,SE
                          WRITE(6,116) J-1,(MU(I,J,IT),IT=1,6)
*EXTENSION*    IO-3
          27              CONTINUE
                    WRITE(6,119)
                    DO 23 J=1,SE
                          WRITE(6,116) J-1,(MU(I,J,IT),IT=7,TM)
*EXTENSION*    IO-3
          23              CONTINUE
          73 CONTINUE
      C
      C      PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
      C      *******************************************************************************
      C
      C            ADD IN THE DEATH RATE FACTOR
      C
      C      *******************************************************************************
              CALL CALCR(R,XE,TM,SA)
              DO 34 LX=1,XE
              DO 34 LS=1,SE
              DO 35 LT=1,TM
          35 MU(LX,LS,LT)=MU(LX,LS,LT)+R(LX,LT)*P1(LX,LS,LT)
          34 CONTINUE
      C      PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
      C
              IF(.NOT.PRTXTR) GO TO 74
              WRITE(6,124)
              DO 24 I=IPS,IPE
                    I1=I+SA-1
                    WRITE(6,117) I1
                    WRITE(6,118)
                    DO 28 J=1,SE
                          WRITE(6,116) J-1,(MU(I,J,IT),IT=1,6)
*EXTENSION*    IO-3
          28              CONTINUE
                    WRITE(6,119)
                    DO 24 J=1,SE
                          WRITE(6,116) J-1,(MU(I,J,IT),IT=7,TM)
*EXTENSION*    IO-3
          24              CONTINUE
          74 CONTINUE
      C
      C      PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
      C      *******************************************************************************
      C
      C          SPLINE  DP/DX+DP/DT+R(X,S,T)*P(X,S,T)   AGAINST  S  AND
      C          INTEGRATE THE SPLINE FROM  S  TO THE MAXIMUM VALUE OF  S.
      C                  MU(X,S,T) IS  END CONDITION PLUS THIS INTEGRAL
      C
      C      *******************************************************************************
              DO 40 IS=1,SE
              K(1,IS)=SL(IS)
        40    CONTINUE
              CALL SETUP(SE,A,H,K)
              DO 41 IX=1,XE
              DO 41 IT=1,TM
```

```
      DO 42 IS=1,SE
      K(2,IS)=MU(IX,IS,IT)
   42 CONTINUE
      CALL SOLVE(A,K,H,SE,D,COF)
      MU(IX,SE,IT)=0.
      DO 43 IS=2,SE
      I=SE+1-IS
      S1=SL(I)
      S2=S1+1.
         MU(IX,I,IT)=MU(IX,I+1,IT)+(((S2*COF(1,I)/4.+COF(2,I)/3.)*
     &   S2+COF(3,I)/2.)*S2+COF(4,I))*S2-(((S1*COF(1,I)/4.
     &   +COF(2,I)/3.)*S1+COF(3,I)/2.)*S1+COF(4,I))*S1
   43    CONTINUE
   41 CONTINUE
      DO 55 KS=1,SE
      DO 55 KT=1,TM
      DO 56 KX=1,XE
      IF (P1(KX,KS,KT).LE.0.0) GO TO 57
      IF  (KX.EQ.XE) GO TO 58
      IF (P1(KX,KS,KT).LE.0.1*P1(KX+1,KS,KT)) GO TO 57
   58    CONTINUE
      IF (KX.EQ.1) GO TO 59
      IF (P1(KX,KS,KT).LE. 0.1*P1(KX-1,KS,KT)) GOTO 57
   59    CONTINUE
      MU(KX,KS,KT)=  MU(KX,KS,KT) / P1(KX,KS,KT)
      GO TO 56
   57 MU(KX,KS,KT)=0.0
   56 CONTINUE
   55    CONTINUE
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRI
C
      IF(.NOT.PRTR) GO TO 71
      WRITE(6,125)
      WRITE(6,120)
      DO 20  I=IPS,IPE
         I1=I+SA-1
         WRITE(6,116) I1,(R(I,IT),IT=1,6)
   20    CONTINUE
      WRITE(6,121)
      DO 21 I=IPS,IPE
         I1=I+SA-1
         WRITE(6,116) I1,(R(I,IT),IT=7,TM)
   21    CONTINUE
   71 CONTINUE
C
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
C
      IF(.NOT.PRTMU) GO TO 75
      DO 25 I=IPS,IPE
         I1=I+SA-1
         WRITE(6,115) I1
         WRITE(6,118)
         DO 29 J=1,SE
            WRITE(6,116) J-1,(MU(I,J,IT),IT=1,6)
*EXTENSION*   IO-3
   29       CONTINUE
         WRITE(6,119)
         DO 25 J=1,SE
            WRITE(6,116) J-1,(MU(I,J,IT),IT=7,TM)
```

```
 25        CONTINUE
 75  CONTINUE
C
C    PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
 115 FORMAT('1',' MOBILITY ESTIMATED FROM SIMULATED DENSITY AT AGE = ',
    &I2,//)
 116 FORMAT(I4,5X,6F10.6)
 117 FORMAT('1',//,' AGE = ',I2,//)
 118 FORMAT(' INCOME/TIME 1963       1964       1965       1966       1967
    &      1968',//)
 119 FORMAT(//,' INCOME/TIME 1969       1970       1971       1972       19
    &73',//)
 120 FORMAT('   AGE/TIME    1963       1964       1965       1966       1967
    &      1968',//)
 121 FORMAT(//,'   AGE/TIME    1969       1970       1971       1972       19
    &73',//)
 122 FORMAT(//,' VALUES OF DP/DX',//)
 123 FORMAT(//,' VALUES OF DP/DX+DP/DT',//)
 124 FORMAT(//,' VALUES OF DP/DX+DP/DT+R*P',//)
 125 FORMAT(//,' DEATH RATES',//)
     DO 9999 I=1,XE
     DO 9999 J=1,SE
     WRITE(2) (MU(I,J,KK),KK=1,TM)
9999 CONTINUE
     LOCK 2

     STOP
     END
C

     SUBROUTINE CALCR(R,NX,NT,SA)
     INTEGER NX,NT,SA
     REAL R(NX,NT)
C
C    ***********************************************************************
C    THIS SUBROUTINE CALCULATES DEATH RATES FOR THE YEARS
C    1963 TO 1963+NT-1
C
C    R(I,J) = FRACTION OF PEOPLE OF AGE  I+SA  DYING AT TIME  J+1962
C    ***********************************************************************
C
     REAL XY(20),XZ(20)
     INTEGER YY,D
     INTEGER TEST/'R'/
     I1=SA+203
     DO 202 I=1,I1
 202 READ(8,100)XX
 100 FORMAT(A1)
     I1=SA+631
     DO 203 I=1,I1
 203 READ(9,100)XX
     DO 206 I=1,20
         XY(I)=0.
 206     CONTINUE
C    READ DATA FOR EACH YEAR
     DO 200 LT=1,NT
C    READ DEATHS FOR EACH YEAR
     DO 201 LX=1,NX
     READ(8,101)D
```

```
      101 FORMAT(T29,I5)
          IF (LX.LE.70-SA) GO TO 210
C         ADJUSTMENT TO ACCOMMODATE LACK OF POPLUATION DATA ABOVE AGE
C         69  FOR YEARS  1967 TO 1970.
          IF(LT.GT.4.AND.LT.LE.8)GO TO 211
C         READ POPULATION FOR EACH YEAR
          READ(9,102)P,YY
      102 FORMAT(12X,F9.1,55X,A1)
C         POPULATION IN NON-CENSUS YEARS MUST BE MULTIPLIED BY  1000.
          IF(YY.EQ.TEST)P=P*1000.
C         LINEAR EXTRAPOLATION FROM 1966 DATA IS USED FOR 1967 TO 1970.
          J=LX-51
          XZ(J)=P-XY(J)
          XY(J)=P
          GO TO 201
      211 J=LX-51
          P=XY(J)+XZ(J)
          XY(J)=P
          GO TO 201
      210 READ(9,102)P,YY
          IF(YY.EQ.TEST)P=P*1000.
      201 R(LX,LT)=D/P
      221 CONTINUE
             IF (LT.EQ.NT) GO TO 200
C         CLEAR REMAINING DATA ON DEATHS.
          NN=1 1-NX
          IF (LT.GT.2) NN=NN-1
          DO 204 I=1,NN
      204 READ(8,100)XX
C         CLEAR REMAINING DATA ON POPULATION.
          NM=9 -NX
          IF (LT.GT.4.AND.LT.LE.8) NM=SA
          DO 205 I=1,NM
      205 READ(9,100)XX
      200 CONTINUE
          RETURN
          END

          SUBROUTINE SETUP(N,A,H,K)
C
C         ******************************************************************
C         THIS ROUTINE SETS UP A TRIDIAGONAL MATRIX OF THE SPLINE EQUATION
C         IN A  3 BY N  ARRAY  -  FIRST ROW IS SUPERDIAGONAL
C                                 SECOND ROW IS DIAGONAL
C                                 THIRD ROW IS SUBDIAGONAL
C         AND THEN DECOMPOSES A TO  LU  SO THAT
C            FIRST ROW IS SUPERDIAGONAL OF U. SECOND ROW IS DIAGONAL OF U
C           THIRD ROW IS SUBDIAGONAL OF L, DIAGONAL OF L IS UNITY
C         ******************************************************************
C
          REAL A(3,N),H(N),K(2,N)
          NM1=N-1
          H(2)-K(1,2)-K(1,1)
          DO 10 I=2,NM1
               H(I+1)=K(1,I+1)-K(1,I)
               A(1,I)=H(I+1)/(H(I+1)+H(I))
               A(2,I)=2.0
               A(3,I)=1-A(1,I)
   10          CONTINUE
          A(1,1) = A(1,2)
```

```
      A(2,1) = A(2,2)
      A(3,1) = A(3,2)
      A(1,N) = A(1,N-1)
      A(2,N) = A(2,N-1)
      A(3,N) = A(3,N-1)
      A(1,2) = -2.
      A(2,2) = 2.
      A(3,2) = 0.
      A(1,N-1) = 0.
      A(2,N-1) = 2.
      A(3,N-1) = -2.
      NM1 = N-1
      DO 11 I=3,NM1
            A(3,I)=A(3,I)/A(2,I-1)
            A(2,I)=A(2,I)-A(3,I)*A(1,I-1)
   11       CONTINUE
      RETURN
      END
C

      SUBROUTINE SOLVE(A,K,H,N,D,COF)
C
C     ******************************************************************
C     THIS ROUTINE CALCULATES THE SECOND ORDER FINITE DIFFERENCES OF
C     THE SPLINE, AND THEN SOLVES  AM=D.  (BY FORWARD AND BACKWARD
C     SUBSTITUTION), PLACING M(VECTOR OF SECOND DERIVATIVES) IN D
C     END POINT CONDITIONS NOW USE THIRD ORDER FINITE DIFFERENCES
C     TO ESTIMATE THE THIRD ORDER DERIVATIVES AT X0+3H/2 AND XN-3H/2.
C     ******************************************************************
C
      REAL H(N),D(N),A(3,N),K(2,N),COF(4,N)
      D(2)=(K(2,2)-K(2,1))/H(2)
      NM1=N-1
      DO 12 I=2,NM1
            D(I+1)=(K(2,I+1)-K(2,I))/H(I+1)
            D(I)=6*(D(I+1)-D(I))/(H(I+1)+H(I))
   12       CONTINUE
      CA=-6./(H(2)*(H(2)+H(3))*(H(2)+H(3)+H(4)))
      CB=6./(H(2)*H(3)*(H(4)+H(3)))
      CC=-6./(H(3)*H(4)*(H(2)+H(3)))
      CD=6./(H(4)*(H(4)+H(3))*(H(2)+H(3)+H(4)))
      D(1)=-2.*H(3)*(CA*K(2,1)+CB*K(2,2)+CC*K(2,3)+CD*K(2,4))
      CA=-6./(H(N-2)*(H(N-2)+H(NM1))*(H(N-2)+H(NM1)+H(N)))
      CB=6./(H(N-2)*H(NM1)*(H(N)+H(NM1)))
      CC=-6./(H(NM1)*H(N)*(H(N-2)+H(NM1)))
      CD=6./(H(N)*(H(N)+H(NM1))*(H(N-2)+H(NM1)+H(N)))
      D(N)=2.*H(NM1)*(CA*K(2,N-3)+CB*K(2,N-2)+CC*K(2,NM1)+CD*K(2,N))
      T1 = D(2)
      T2 = D(NM1)
      D(2) = D(1)
      D(NM1) = D(N)
      DO 13 I=3,NM1
            D(I)=D(I)-A(3,I)*D(I-1)
   13       CONTINUE
      D(NM1) = D(NM1)/A(2,NM1)
      DO 14 I=3,NM1
            J=N+1-I
            D(J)=(D(J)-A(1,J)*D(J+1))/A(2,J)
   14       CONTINUE
      D(1) = (T1 - A(2,1)*D(2) - A(1,1)*D(3))/A(3,1)
```

```
      D(N) = (T2 - A(2,N)*D(NM1) - A(3,N)*D(N-2))/A(1,N)
      CALL POLLY(N,D,K,H,COF)
      RETURN
      END
C
      SUBROUTINE POLLY(N,M,K,H,COF)
C
C     ***************************************************************
C     THIS ROUTINE COMPUTES THE COEFFICIENTS OF THE SPLINE POLYNOMIAL
C        ON EACH SUBINTERVAL
C     K IS THE ARRAY OF DATA POINTS
C     H IS THE VECTOR OF SUBINTERVAL LENGTHS
C     M IS THE SOLUTION VECTOR TO THE EQUATION AM=D
C     ***************************************************************
C
      REAL M(N),K(2,N),H(N),COF(4,N)
      NM1=N-1
      DO 11 I=1,NM1
      COF(1,I)=(M(I+1)-M(I))/(6.*H(I+1))
      COF(2,I)=(K(1,I+1)*M(I)-K(1,I)*M(I+1))/(2.*H(I+1))
      DD=M(I+1)*K(1,I)*K(1,I)-M(I)*K(1,I+1)*K(1,I+1)
     &+2.*K(2,I+1)-2.*K(2,I)
      COF(3,I)=(DD/(2.*H(I+1)))+H(I+1)*(M(I)-M(I+1))/6.
      DD=M(I)*(K(1,I+1)**3)-M(I+1)*(K(1,I)**3)+6.*K(1,I+1)*K(2,I)
     &-6.*K(1,I)*K(2,I+1)+K(1,I)*M(I+1)*(H(I+1)**2)-K(1,I+1)*M(I)
     &*(H(I+1)**2)
      COF(4,I)=DD/(H(I+1)*6.)
   11 CONTINUE
      DO 16 J=1,4
   16 COF(J,N)=COF(J,NM1)
      RETURN
         END
```

```
**************************************************************************

DATA FOR
     PROGRAM 3.3 :  ESTMU

**************************************************************************
&VALUES
PRTR=.FALSE.,PRTPX=.FALSE.,PRTPXT=.FALSE.,PRTXTR=.TRUE.,PRTMU=.TRUE.,
&END
&PRTLIM IPS=10,IPE=32, &END
11 13 11
62 20 17
```

NUMBER OF INCOME GROUPS =

NUMBER OF INCOME DENSITIES = 20

STARTING AGE = 17

MOBILITY ESTIMATED FROM SIMULATED DENSITY AT AGE = 26

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.072791	0.120657	0.189769	0.276194	0.540420	0.000000
1	0.221319	0.239642	0.252742	0.257842	0.271820	0.279733
2	0.287052	0.314241	0.337511	0.354622	0.375727	0.388790
3	0.342370	0.378445	0.408407	0.431023	0.453637	0.467394
4	0.369774	0.411577	0.446482	0.473703	0.496839	0.508850
5	0.359630	0.408244	0.443589	0.467313	0.485951	0.498303
6	0.317196	0.366557	0.398929	0.417819	0.430364	0.438636
7	0.257083	0.294389	0.321570	0.337939	0.347976	0.351927
8	0.188839	0.213788	0.235781	0.251907	0.260818	0.260124
9	0.124349	0.142221	0.157553	0.168510	0.174736	0.174851
10	0.074393	0.085426	0.095130	0.102235	0.106262	0.105745
11	0.040246	0.046515	0.051952	0.055747	0.057971	0.057643
12	0.019768	0.022871	0.025622	0.027489	0.028669	0.028156
13	0.008806	0.010189	0.011443	0.012165	0.012746	0.012398
14	0.003605	0.004087	0.004626	0.004857	0.005193	0.004847
15	0.001415	0.001460	0.001702	0.001692	0.001900	0.001652
16	0.000645	0.000397	0.000560	0.000477	0.000650	0.000387
17	0.000473	-0.000063	0.000126	-0.000012	0.000134	-0.000109
18	0.000330	-0.000543	-0.000347	-0.000484	-0.000345	-0.000821
19	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

INCOME/TIME	1969	1970	1971	1972	1973
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.291173	0.294730	0.272133	0.325693	0.019812
2	0.399325	0.395270	0.372735	0.372113	0.199023
3	0.476183	0.469566	0.448777	0.427716	0.309736
4	0.513543	0.504706	0.483784	0.453750	0.371659
5	0.503580	0.492556	0.467344	0.434342	0.377384
6	0.442096	0.431592	0.408460	0.377553	0.336248
7	0.350078	0.337626	0.320576	0.294620	0.272441
8	0.254349	0.244301	0.236786	0.212625	0.221191
9	0.171809	0.164773	0.163021	0.135137	0.193080
10	0.103425	0.097895	0.101535	0.064883	0.183091
11	0.056957	0.052693	0.061136	0.013104	0.190279
12	0.027905	0.023854	0.035449	-0.023050	0.200645
13	0.012848	0.009310	0.022930	-0.042145	0.210547
14	0.005367	0.001929	0.016220	-0.052610	0.215937
15	0.002365	-0.000749	0.013728	-0.056325	0.218678
16	0.000960	-0.002081	0.011892	-0.057916	0.216702
17	0.000385	-0.002383	0.010532	-0.055719	0.205780
18	-0.001268	-0.004140	0.004200	-0.046044	0.148271
19	0.000000	0.000000	0.000000	0.000000	0.000000

MOBILITY ESTIMATED FROM SIMULATED DENSITY AT AGE = 27

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.081170	0.122179	0.208709	0.325238	0.705301	0.000000
1	0.232679	0.245703	0.262816	0.270039	0.281375	0.296765
2	0.299542	0.323822	0.350641	0.370073	0.389707	0.407347
3	0.355842	0.390594	0.423556	0.448538	0.470491	0.487645
4	0.383476	0.424958	0.462709	0.492375	0.515328	0.529336
5	0.372285	0.421455	0.459320	0.485151	0.503960	0.517746
6	0.327976	0.378169	0.412781	0.433418	0.445871	0.454605
7	0.265638	0.303357	0.332220	0.349784	0.359784	0.363928
8	0.194964	0.220055	0.243466	0.260619	0.269310	0.268189
9	0.128314	0.146196	0.162580	0.174253	0.180438	0.180553
10	0.076774	0.087668	0.098150	0.105695	0.109627	0.109180
11	0.041583	0.047610	0.053600	0.057630	0.059770	0.059706
12	0.020500	0.023306	0.026467	0.028429	0.029500	0.029298
13	0.009225	0.010286	0.011864	0.012605	0.013091	0.013103
14	0.003881	0.004034	0.004853	0.005057	0.005305	0.005305
15	0.001639	0.001345	0.001849	0.001787	0.001929	0.002012
16	0.000868	0.000257	0.000681	0.000524	0.000645	0.000674
17	0.000727	-0.000215	0.000242	0.000014	0.000128	0.000134
18	0.000604	-0.000655	-0.000214	-0.000460	-0.000354	-0.000679
19	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

INCOME/TIME	1969	1970	1971	1972	1973
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.304297	0.300655	0.314738	0.234725	0.473315
2	0.413063	0.405410	0.395287	0.338315	0.391326
3	0.492305	0.484194	0.468144	0.419819	0.423462
4	0.530446	0.520727	0.501833	0.459368	0.434386
5	0.520126	0.507990	0.483577	0.444249	0.413284
6	0.455860	0.445524	0.422538	0.388342	0.361209
7	0.359754	0.347505	0.328387	0.306059	0.279413
8	0.260011	0.250609	0.238498	0.227162	0.194468
9	0.175796	0.170280	0.161514	0.160273	0.118790
10	0.105348	0.102108	0.095160	0.103010	0.042501
11	0.057687	0.056681	0.050697	0.066586	-0.015448
12	0.027743	0.027531	0.021551	0.042495	-0.059407
13	0.012337	0.012977	0.007378	0.031445	-0.081944
14	0.004679	0.005426	-0.000090	0.024841	-0.095280
15	0.001671	0.002630	-0.002483	0.022576	-0.099100
16	0.000285	0.001062	-0.003872	0.020161	-0.100892
17	-0.000211	0.000448	-0.004013	0.018069	-0.096294
18	-0.001671	-0.002201	-0.006406	0.007627	-0.080181
19	0.000000	0.000000	0.000000	0.000000	0.000000

MOBILITY ESTIMATED FROM SIMULATED DENSITY AT AGE = 28

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.047848	0.146391	0.201829	0.382633	0.920535	0.000000
1	0.218789	0.260310	0.263383	0.280954	0.286161	0.298729
2	0.295130	0.337322	0.355828	0.382744	0.398520	0.413655
3	0.356672	0.403770	0.431766	0.462284	0.481867	0.497104
4	0.387022	0.437631	0.472670	0.506668	0.528199	0.540429
5	0.376594	0.433080	0.469494	0.498560	0.516716	0.529376
6	0.331989	0.387985	0.421946	0.445011	0.456851	0.464411
7	0.268821	0.310800	0.339158	0.358376	0.368013	0.371283
8	0.196946	0.225300	0.248354	0.266867	0.275027	0.272698
9	0.129182	0.149652	0.165611	0.178380	0.184253	0.183721
10	0.076871	0.089808	0.099803	0.108213	0.111799	0.110864
11	0.041228	0.048882	0.054323	0.059058	0.060881	0.060534
12	0.019941	0.024072	0.026666	0.029212	0.029951	0.029551
13	0.008619	0.010777	0.011802	0.013053	0.013224	0.013117
14	0.003309	0.004385	0.004687	0.005345	0.005283	0.005204
15	0.001144	0.001617	0.001653	0.002007	0.001855	0.001891
16	0.000479	0.000476	0.000493	0.000711	0.000548	0.000550
17	0.000479	-0.000051	0.000085	0.000181	0.000035	0.000031
18	0.000540	-0.000537	-0.000280	-0.000314	-0.000425	-0.000729
19	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

INCOME/TIME	1969	1970	1971	1972	1973
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.306685	0.325710	0.306834	0.357273	0.069812
2	0.419090	0.420591	0.395000	0.380900	0.268023
3	0.501640	0.499202	0.474050	0.448718	0.358009
4	0.541193	0.534896	0.510632	0.480378	0.403332
5	0.531290	0.520379	0.492541	0.459862	0.402325
6	0.465520	0.455994	0.431646	0.400475	0.359126
7	0.366646	0.353969	0.334927	0.312850	0.283628
8	0.263705	0.253276	0.242014	0.226495	0.205817
9	0.178665	0.172224	0.165751	0.155531	0.147772
10	0.106947	0.102667	0.099351	0.091895	0.094708
11	0.058662	0.056553	0.055561	0.049728	0.062227
12	0.028234	0.026820	0.026661	0.020873	0.039249
13	0.012644	0.012092	0.012725	0.007444	0.029780
14	0.004875	0.004468	0.005171	-0.000154	0.023262
15	0.001830	0.001714	0.002620	-0.002269	0.021679
16	0.000418	0.000214	0.000924	-0.003933	0.019055
17	-0.000092	-0.000284	0.000314	-0.004147	0.017164
18	-0.001509	-0.002555	-0.003262	-0.008186	0.004348
19	0.000000	0.000000	0.000000	0.000000	0.000000

MOBILITY ESTIMATED FROM SIMULATED DENSITY AT AGE = 29

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.106794	0.128960	0.246751	0.350264	1.507626	0.000000
1	0.254438	0.255116	0.278034	0.275061	0.295077	0.304053
2	0.319031	0.336749	0.368315	0.382236	0.407727	0.419846
3	0.374310	0.405972	0.442947	0.465045	0.491423	0.504522
4	0.400854	0.441275	0.482828	0.511619	0.537706	0.548123
5	0.387493	0.437148	0.478358	0.504175	0.525445	0.536974
6	0.340530	0.391601	0.429259	0.450307	0.463950	0.470417
7	0.275342	0.313392	0.344413	0.362163	0.373025	0.375536
8	0.201609	0.226922	0.252047	0.269467	0.278257	0.274822
9	0.132391	0.150491	0.168021	0.179905	0.186452	0.185268
10	0.079083	0.090111	0.101323	0.108916	0.113088	0.111642
11	0.042801	0.048862	0.055256	0.059232	0.061637	0.060929
12	0.021122	0.023901	0.027265	0.029090	0.030373	0.029688
13	0.009568	0.010549	0.012220	0.012803	0.013499	0.013156
14	0.004114	0.004150	0.005014	0.005050	0.005481	0.005193
15	0.001862	0.001389	0.001932	0.001712	0.002030	0.001867
16	0.001149	0.000254	0.000744	0.000428	0.000706	0.000519
17	0.001114	-0.000267	0.000316	-0.000073	0.000186	0.000004
18	0.001050	-0.000681	-0.000067	-0.000480	-0.000291	-0.000708
19	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

INCOME/TIME	1969	1970	1971	1972	1973
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.323264	0.319513	0.341357	0.290598	0.572427
2	0.429885	0.419016	0.406433	0.361119	0.358080
3	0.511991	0.501020	0.483813	0.440086	0.411760
4	0.550420	0.538440	0.519208	0.477459	0.438434
5	0.539326	0.524456	0.499300	0.460207	0.424990
6	0.471632	0.460457	0.437534	0.402754	0.373181
7	0.370506	0.356754	0.338363	0.315891	0.291513
8	0.264957	0.253529	0.241928	0.226912	0.205154
9	0.179734	0.172761	0.165841	0.157430	0.143361
10	0.107460	0.102870	0.098901	0.094653	0.085015
11	0.058965	0.056690	0.054824	0.053484	0.046865
12	0.028387	0.026888	0.025690	0.025366	0.019427
13	0.012747	0.012144	0.011670	0.012246	0.007338
14	0.004967	0.004535	0.004157	0.004762	-0.000053
15	0.001913	0.001768	0.001647	0.002471	-0.001655
16	0.000511	0.000291	0.000077	0.000626	-0.003396
17	0.000005	-0.000203	-0.000399	-0.000013	-0.003446
18	-0.001286	-0.002223	-0.003305	-0.004554	-0.009018
19	0.000000	0.000000	0.000000	0.000000	0.000000

MOBILITY ESTIMATED FROM SIMULATED DENSITY AT AGE = 30

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.051234	0.147577	0.207147	0.473681	1.988387	0.000000
1	0.225021	0.263701	0.266633	0.291783	0.286605	0.312078
2	0.302318	0.342905	0.362178	0.394348	0.403369	0.425137
3	0.363915	0.410663	0.439888	0.474195	0.489388	0.509155
4	0.393753	0.444904	0.481640	0.518674	0.537251	0.551814
5	0.382072	0.439910	0.478062	0.509380	0.525890	0.540022
6	0.336221	0.393509	0.429295	0.454076	0.464492	0.472211
7	0.271879	0.314480	0.344140	0.364360	0.373120	0.376362
8	0.198768	0.227501	0.251622	0.270821	0.277827	0.274414
9	0.130139	0.150763	0.167481	0.180798	0.186056	0.185058
10	0.077348	0.090251	0.100780	0.109533	0.112622	0.111439
11	0.041484	0.048952	0.054749	0.059719	0.061215	0.060843
12	0.020121	0.023987	0.026826	0.029507	0.029986	0.029687
13	0.008804	0.010636	0.011848	0.013189	0.013170	0.013221
14	0.003532	0.004232	0.004703	0.005408	0.005189	0.005296
15	0.001433	0.001454	0.001670	0.002046	0.001773	0.001987
16	0.000869	0.000291	0.000530	0.000733	0.000471	0.000647
17	0.000986	-0.000264	0.000155	0.000198	-0.000016	0.000133
18	0.001049	-0.000668	-0.000123	-0.000256	-0.000414	-0.000539
19	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

INCOME/TIME	1969	1970	1971	1972	1973
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.321925	0.337310	0.342718	0.487429	-0.479835
2	0.428744	0.424641	0.403595	0.389126	0.273394
3	0.511909	0.505232	0.482463	0.456738	0.365362
4	0.550624	0.541376	0.518560	0.488521	0.408491
5	0.539853	0.526071	0.498721	0.467176	0.407656
6	0.471917	0.461495	0.437671	0.406613	0.363847
7	0.370250	0.356695	0.338348	0.318218	0.289026
8	0.263345	0.251537	0.239691	0.225332	0.202179
9	0.178723	0.171532	0.164559	0.156227	0.142440
10	0.106707	0.102053	0.098193	0.093797	0.086032
11	0.058457	0.056166	0.054412	0.052503	0.048810
12	0.028087	0.026649	0.025634	0.024598	0.022680
13	0.012550	0.012001	0.011672	0.011342	0.010737
14	0.004861	0.004516	0.004310	0.004062	0.003772
15	0.001830	0.001754	0.001767	0.001724	0.001889
16	0.000468	0.000340	0.000280	0.000096	0.000254
17	-0.000018	-0.000134	-0.000189	-0.000430	-0.000058
18	-0.001114	-0.001814	-0.002551	-0.003759	-0.004542
19	0.000000	0.000000	0.000000	0.000000	0.000000

3.4 Estimating Components of Mobility

As for fertility the estimated values of mobility obtained from a (three-dimensional) array MU(x,s,t) . To estimate the modes $\{a_r(t)\}$, of $\mu(x,s,t)$, the dominant eigenvectors $\{\bar{a}_r\}$ the square matrix*

$$MTM(k,\ell) = \sum_{i=LX}^{MX} \sum_{j=LS}^{MS} MU(i,j,k)\ MU(i,j,\ell)$$

are obtained. From these time components of the estimated mobility, the age-income components are estimated as

$$\bar{d}_r(x,s) = \sum_{k=ITS}^{TE+ITS} MU(x,s,k)\ \bar{a}_r(k) \quad .$$

The program 3.4 allows partitioning the array MU(x,s,t) into (NT-TE+1) arrays so that estimations of the dominant components of MU(x,s,t) on successively overlapping time intervals may be computed. Further, bordering of the arrays MTM in the manner described in the estimation of fertility parameters is possible.

The output following the source listing for Program 3.4 is that obtained on estimating the components of that MU(x,s,t) computed using the density function from the aggregation experiments.

Lineprinter plots of the time components of MU(x,s,t) are plotted using Subroutine 3.7. Surface plots for the corresponding age-income components of MU(x,s,t) are plotted using Program 2.7.

```
$JOB    ACCT-NUM,VERNER,TIME=60
C       ****************************************************************
C
C       PROGRAM 3.4 : POP
C
C       ****************************************************************
```

```
3JOB POP; CHARGE BERGHOUT; CLASS 3;
USER 080002/VHJ;
PRINTLIMIT=13200;
3PROCESSTIME=300;
BEGIN
COMPILE POP FORTRAN;
3FILE FILE2(KIND=PACK,TITLE=DATA/XXMU7,FILETYPE=7);
FILE FILE3(KIND=PACK,TITLE=PLOT/DENS,MAXRECSIZE=40,BLOCKWISE=120,
AREASIZE=35,AREAS=1,FLEXIBLE,SAVEFACTOR=10);
OPTION=AUTORM;
DATA



3DATA FILE5



3END JOB
```

```
C       **************************************************************************
C
C       PROGRAM : 3.4
C       TITLE : POP
C       PURPOSE : TO ESTIMATE DOMINANT COMPONENTS OF MU
C       DATA : DATA/XXMU7
C       OUTPUT : TIME COMPONENTS PRINTED AND PLOTTED ON LINEPRINTER
C                  PLOT/DENS - FOR SURFACE PLOTS
C
C       **************************************************************************
$RESET LIST
$INCLUDE 'OLIB/ES/PLOTT'
$SET LIST
        INTEGER NX,NS,NT,TE
        INTEGER SYEAR,NEV
        INTEGER LX,LS
        INTEGER MX,MS
        COMMON/ZERO/ICL,ICT,IRL,IRT
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C       NX:  TOTAL NUMBER OF AGES AVAILABLE
C       NS:  TOTAL NUMBER OF INCOME GROUPS
C       NT:  NUMBER OF YEARS COVERED
C       TE:  NUMBER OF YEARS PER ESTIMATION
C
C       LX:  LOWEST AGE
C       MX:  HIGHEST AGE
C       LS:  LOWEST INCOME GROUP
C       MS:  HIGHEST INCOME GROUP
C       SYEAR:  FIRST YEAR
C
C       NEV:  NUMBER OF EIGEN VALUES TO BE USED
C
C       ICL:  # OF LEADING COLUMNS OF M MATRIX TO BE ZEROED
C       ICT:  # OF TRAILING COLUMNS OF M MATRIX TO BE ZEROED
C       IRL:  # OF LEADING ROWS OF M MATRIX TO BE ZEROED
C       IRT:  # OF TRAILING ROWS TO BE ZEROED
C
C       REAL MU(NX,NS,TE),D(NX,NS,NEV)
C       REAL M(TE,TE),MEIG(TE,TE),EVT(TE,TE),A(TE*(TE+1)/2)
C       FOR ONE RUN OVER THE WHOLE TIME PERIOD
C       REAL PT(TE,4,NEV),PL(TE,4)
C       FOR SEVERAL RUNS OVER SUB-INTERVALS OF TIME PERIOD
C       REAL DMESH(NT,NT-TE+3,3,NEV),  DSUM(NT,NT-TE+3)
C       REAL D1(NX,NS,NEV),DDSUM(NX,NS,NT-TE+1,NEV)
C
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
        REAL MU(62,20,11),D(62,20,5)
        REAL M(11,11),MEIG(11,11),EVT(11,11),A(66)
        REAL PT(11,4,5),PL(11,4)
        REAL DMESH(1,1,1,1),DSUM(1,1)
        REAL D1(1,1,1),DDSUM(1,1,1,1)
C
C       READ MU(X,S,T) WHICH IS STORED ON PACK IN FILE DATA/XXMU.
C       THE MU VALUES WERE OBTAINED FROM THE PROGRAM MUTEST
C       X = AGES 17-78; S = INCOMES $ 000-19000; T = YEARS 1963-1973.
C
        READ(5,105) NX,NS,NT,TE
C
```

```
C       READ AND PRINT PARAMETERS USED IN THIS RUN
C
        READ(5,102) LX,MX
        READ(5,102) LS,MS
        READ(5,103) NEV
        READ(5,103) SYEAR
        LL = SYEAR + NT - 1
C       PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
C
        WRITE(6,201)
        WRITE(6,202) LX,MX
        WRITE(6,203) LS,MS
        WRITE(6,204) NEV
        WRITE(6,205) SYEAR,LL
        WRITE(6,206)  TE
        LL=NT-TE+1
        WRITE(6,207) LL
        LLP1-LL+1
        LLP2-LL+2
    102 FORMAT(2I3)
    103 FORMAT(I3)
    104 FORMAT(4I2)
    105 FORMAT(4I3)
    201 FORMAT('1','PARAMETERS USED FOR THIS RUN',////)
    202 FORMAT('0','RANGE OF AGES USED:      LX = ',I2,'   MX = ',I2)
    203 FORMAT('0','RANGE OF INCOMES USED:  LS = ',I2,'   MS = ',I2)
    204 FORMAT('0','NUMBER OF EIGEN VALUES USED:    = ',I2)
    205 FORMAT('0','RANGE OF YEARS 19',I2,' TO 19',I2)
    206 FORMAT('0','NUMBER OF YEARS PER ESTIMATION:  TE = ',I2)
    207 FORMAT('0','NUMBER OF ESTIMATIONS:  LL = ',I2)
C
C       PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
C
        IF (TE.EQ.NT) GO TO 67
        DO 12 I=1,NT
            DO 12 J=1,3
                DO 12 NV=1,NEV
                    DMESH(I,1,J,NV)=I+SYEAR-1.
     12             CONTINUE
     67 CONTINUE
        ITS=
        ITE=TE-1
     49 CONTINUE
        DO 11 I=1,TE
            DO 11 J=1,NEV
     11         PT(I,1,J)=I+SYEAR-1.+ITS
        ITS=ITS+1
        ITE=ITE+1
        DO 1 I=1,NX
        DO 1 J=1,NS
          READ(2)(MU(I,J,K),K=1,TE)
      1     CONTINUE
        DO 49 II=2,4
        READ(5,104) ICL,ICT,IRL,IRT
        CALL CALCM(NX,NS,TE,MU,LX,MX,LS,MS,M)
        CALL ZEROM(M,TE)
        DO 5 I=1,TE
            DO 5 J=I,TE
                IA=I+(J*J-J)/2
      5         A(IA)=M(I,J)
```

```
      CALL EIGEN(A,EVT,TE,0)
      DO 6 I=1,TE
         DO 6 J=I,TE
            IA=I+(J*J-J)/2
    6       MEIG(I,J)=A(IA)
      DO 9 J=1,NEV
         SUM=0.
         DO 10 I=1,TE
            SUM=SUM+EVT(I,J)**2
   10       CONTINUE
         SUM=1./SQRT(SUM)
         DO 9 I=1,TE
            EVT(I,J)=EVT(I,J)*SUM
    9       CONTINUE
      CALL CALCD(NX,NS,TE,MU,EVT,NEV,D)
C     ********************************************************************
C
C     NORMALIZE VECTORS: EG.  D2=:D2*(D1,D2)/(D2,D2)
C                        HENCE IF D2=K*D1, K CONSTANT
C                        D1  AND  D2  WILL HAVE THE SAME NORMS
C
C     ********************************************************************
      IF (TE.EQ.NT) GO TO 48
      IF (ITS.GT.1) GO TO 43
      DO 42 I=1,NX
         DO 42 J=1,NS
            DO 42 K=1,NEV
   42          D1(I,J,K)=D(I,J,K)
      GO TO 48
   43 DO 46 K=1,NEV
         SUM=0
         SUM1=0.
         DO 44 I=LX,MX
            DO 44 J=LS,MS
               SUM=SUM+D(I,J,K)*D(I,J,K)
   44          SUM1=SUM1+D(I,J,K)*D1(I,J,K)
         SUM=SUM/SUM1
         DO 45 I=1,TE
   45       EVT(I,K)=EVT(I,K)*SUM
         DO 46 I=1,NX
            DO 46 J=1,NS
   46          D(I,J,K)=D(I,J,K)/SUM
   48 CONTINUE
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
C
      WRITE(6,100) TF,TE,ICL
      DO 3 I=1,NEV
      WRITE(6,400)MEIG(I,I)
      WRITE(6,500) I
      WRITE(6,600)(EVT(J,I),J=1,TF)
      DO 3 J=1,TF
         PT(J,II,I)=EVT(J,I)
    3 CONTINUE
      IF ( NT.EQ.TF) GO TO 68
      DO 4 I=1,NEV
      J1=0
      DO 4 J=ITS,ITE
         J1=J1+1
    4    DMESH(J,ITS+1,II-1,I)=EVT(J1,I)
   68 CONTINUE
```

```
C
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
      IF (II.GT.2.OR.NT.NE.TE) GO TO 77
      DO 21 K=1,4
         DO 21 J=1,20
            WRITE(3)  (D(I,J,K),I=1,40)
   21       CONTINUE
   77 IF (II.NE.3.OR.NT.EQ.TE) GO TO 41
C     ***********************************************************************
C
C     IN MULTIPLE RUNS D(X,S) OBTAINED WITH SINGLE BORDERING IS
C     USED FOR AVERAGING AND PLOTTING OF PERSPECTIVE GRAPHS
C
C     ***********************************************************************
      DO 78 I=1,NX
         DO 78 J=1,NS
            DO 78 K=1,NEV
   78          DDSUM(I,J,ITS,K)=D(I,J,K)
   41 CONTINUE
C
C     CALL PLOTTING ROUTINE HERE
C
C     ***********************************************************************
C
C     PLOT TIME VECTORS WITH DIFFERENT BORDERING ON PRINTER
C
C     ***********************************************************************
      DO 35 K=1,5
         DO 31 I=1,TE
            DO 31 J=1,4
   31          PL(I,J)=PT(I,J,K)
         DO 33 J1=3,4
            SUM=0.
            DO 32 I1=1,TE
               SUM=SUM+PL(I1,2)*PL(I1,J1)
   32          CONTINUE
            IF (SUM.GE.0) GOTO 33
            DO 34 I1=1,TE
   34          PL(I1,J1)=-PL(I1,J1)
   33          CONTINUE
         CALL PLOTR(K,PL,TE,4,0.,0.,0.,0.)
   35    CONTINUE
      IF (NT.GT.ITE) GO TO 49
      IF (TE.EQ.NT) GO TO 59
C     ***********************************************************************
C
C     SUMMARIZE TIME VECTORS AND PLOT ON PRINTER
C
C     ***********************************************************************
      DO 59 NV=1,NEV
         DO 58 KK=1,3
            DO 51 I=1,NT
               DO 51 J=1,LLP1
   51             DSUM(I,J)=DMESH(I,J,KK,NV)
            DO 57 I=1,NT
               SUM=0.
               NC=0
               DO 56 J=2,LLP1
                  IF(I.LT.J+KK-2 .OR.I.GT.TE+J-KK-1)  GO TO 56
                  NC=NC+1
```

```
                           SUM=SUM+DSUM(I,J)
      56                   CONTINUE
                   IF (NC.EQ.0) NC=1
                   DSUM(I,LLP2)=SUM/NC
      57           CONTINUE
               KKM1=KK-1
               WRITE(6,301) NV,KKM1
               WRITE(6,302)
               DO 55  I=1,NT
                   WRITE(6,107) (DSUM(I,J),J=1,LLP2)
      55           CONTINUE
               INT=100*NV+KK
                   CALL PLOTB(INT,DSUM,NT,LLP2,0.,0.,0.,0.)
      58 CONTINUE
C        ******************************************************************
C
C        SUMMARIZE SPACE VECTORS AND WRITE TO DISK FOR PERSPECTIVE PLOTS
C
C        ******************************************************************
         DO 83 K=1,4
               DO 82 I=1,NX
                   DO 82 J=1,NS
                       SUM=0.
                       DO 81 II=1,LL
      81                   SUM=SUM+DDSUM(I,J,II,K)
      82               D(I,J,K)=SUM/LL
               DO 83 J=1,20
                   WRITE(3) (D(I,J,K),I=1,40)
      83           CONTINUE
      59 CONTINUE
     100 FORMAT('1','EIGENPAIRS OF MTM(',I2,',',I2,')     BORDERING ',I2,
        &' ROWS AND COLUMNS',//)
     107 FORMAT(4X,'19',I2,4X,5F12.6)
     301 FORMAT('1','SUMMARY - COMPONENT ',I2,', BORDERING ',I2,
        &        ' BOUNDARY VALUES',//)
     302 FORMAT(T5,'TIME',T17,'RUN1',T29,'RUN2',T39,'SUMMARY',//)
     400 FORMAT('0',10X,'EIGENVALUE = ',F15.5)
     500 FORMAT('0','EVT(I,',I2,')')
     600 FORMAT('0',5F12.6)
         LOCK 3

         STOP
         END

         SUBROUTINE ZEROM(M,NT)
C
C        THIS SUBROUTINE WILL ZERO OUT ROWS AND COLUMNS OF THE
C        M MATRIX AS SPECIFIED BY ICL,ITL,IRL AND IRT
C
         COMMON/ZERO/ICL,ICT,IRL,IRT
         REAL M(NT,NT)
         DO 1 I = 1,NT
         IF (ICL.EQ.0) GO TO 2
         DO 3 J = 1,ICL
       3    M(I,J) = 0.0
       2 IF (ICT.EQ.0) GO TO 1
         LL = NT + 1
         DO 4 J = 1,ICT
            LL = LL - 1
       4    M(I,LL) = 0.0
```

```
    1 CONTINUE
      DO 5 I = 1,NT
      DO 9  J=1,IRL
      IF (IRL.EQ.0) GO TO 6
    9    M(J,I)=0.0
    6 IF (IRT.EQ.0) GO TO 5
      LL = NT + 1
      DO 7 J = 1,IRT
         LL = LL - 1
    7    M(LL,I) = 0.0
    5 CONTINUE
      RETURN
      END

      SUBROUTINE CALCM(NX,NS,NT,MU,LX,MX,LS,MS,M)
      INTEGER NX,NS,NT,LX,MX,LS,MS
      REAL MU(NX,NS,NT),M(NT,NT)
C
C     THIS SUBROUTINE CALCULATES THE MATRIX M
C     M(I,J) = THE SUM OVER X AND S OF MU(X,S,TI) * MU(X,S,TJ)
C     THE VALUES OF X RANGE FROM LX TO MX
C     THE VALUES OF S RANGE FROM LS TO MS
C
      DO 10 I=1,NT
      DO 10  J=I,NT
         SUM=0
         DO 20 K = LX,MX
         DO 20 L = LS,MS
         SUM=SUM+MU(K,L,I)*MU(K,L,J)
   20    CONTINUE
      M(I,J)=SUM
      M(J,I)=SUM
   10 CONTINUE
      RETURN
      END

      SUBROUTINE CALCD(NX,NS,NT,MU,EVT,NEV,D)
      INTEGER NX,NS,NT,NEV
      REAL MU(NX,NS,NT),D(NX,NS,NEV),EVT(NT,NT)
C
C     CALCULATE A D-MATRIX FOR EACH EIGENVECTOR
C     THE NUMBER OF EIGEN VECTORS TO BE USED IS GIVEN BY NEV
C
      DO 1 IV = 1,NEV
      DO 2 IX = 1,NX
      DO 2 IS=1,NS
      SUM = 0.0
      SUM1 = 0.0
      DO 3 IT = 1,NT
         SUM = SUM + EVT(IT,IV) * MU(IX,IS,IT)
         SUM1 = SUM1 + (EVT(IT,IV))**2
    3    CONTINUE
      D(IX,IS,IV) = SUM / SUM1
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
C
C     .................................................................
C
```

```
C        SUBROUTINE EIGEN
C
C        PURPOSE
C           COMPUTE EIGENVALUES AND EIGENVECTORS OF A REAL SYMMETRIC
C           MATRIX
C
C        USAGE
C           CALL EIGEN(A,R,N,MV)
C
C        DESCRIPTION OF PARAMETERS
C           A - ORIGINAL MATRIX (SYMMETRIC), DESTROYED IN COMPUTATION.
C               RESULTANT EIGENVALUES ARE DEVELOPED IN DIAGONAL OF
C               MATRIX A IN DESCENDING ORDER.
C           R - RESULTANT MATRIX OF EIGENVECTORS (STORED COLUMNWISE,
C               IN SAME SEQUENCE AS EIGENVALUES)
C           N - ORDER OF MATRICES A AND R
C           MV- INPUT CODE
C                   0   COMPUTE EIGENVALUES AND EIGENVECTORS
C                   1   COMPUTE EIGENVALUES ONLY (R NEED NOT BE
C                       DIMENSIONED BUT MUST STILL APPEAR IN CALLING
C                       SEQUENCE)
C
C        REMARKS
C           ORIGINAL MATRIX A MUST BE REAL SYMMETRIC (STORAGE MODE=1)
C           MATRIX A CANNOT BE IN THE SAME LOCATION AS MATRIX R
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           DIAGONALIZATION METHOD ORIGINATED BY JACOBI AND ADAPTED
C           BY VON NEUMANN FOR LARGE COMPUTERS AS FOUND IN 'MATHEMATICAL
C           METHODS FOR DIGITAL COMPUTERS', EDITED BY A. RALSTON AND
C           H.S. WILF, JOHN WILEY AND SONS, NEW YORK, 1962, CHAPTER 7
C
C     ..................................................................
C
      SUBROUTINE EIGEN(A,R,N,MV)
      DIMENSION A(1),R(1)
C
C        ...............................................................
C
C        IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C        C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
C        STATEMENT WHICH FOLLOWS.
C
C     DOUBLE PRECISION A,R,ANORM,ANRMX,THR,X,Y,SINX,SINX2,COSX,
C    1                 COSX2,SINCS
C
C        THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C        APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C        ROUTINE.
C
C        THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
C        CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  SQRT IN STATEMENTS
C        40, 68, 75, AND 78 MUST BE CHANGED TO DSQRT.  ABS IN STATEMENT
C        62 MUST BE CHANGED TO DABS.
C
C        ...............................................................
```

```
C
C         GENERATE IDENTITY MATRIX
C
      IF(MV-1) 10,25,10
   10 IQ=-N
      DO 20 J=1,N
      IQ=IQ+N
      DO 20 I=1,N
      IJ=IQ+I
      R(IJ)=0.0
      IF(I-J) 20,15,20
   15 R(IJ)=1.0
   20 CONTINUE
C
C         COMPUTE INITIAL AND FINAL NORMS (ANORM AND ANORMX)
C
   25 ANORM=0.0
      DO 35 I=1,N
      DO 35 J=I,N
      IF(I-J) 30,35,30
   30 IA=I+(J*J-J)/2
      ANORM=ANORM+A(IA)*A(IA)
   35 CONTINUE
      IF(ANORM) 165,165,40
   40 ANORM=1.414* SQRT(ANORM)
      ANRMX=ANORM*1.0E-6/FLOAT(N)
C
C         INITIALIZE INDICATORS AND COMPUTE THRESHOLD, THR
C
      IND=0
      THR=ANORM
   45 THR=THR/FLOAT(N)
   50 L=1
   55 M=L+1
C
C         COMPUTE SIN AND COS
C
   60 MQ=(M*M-M)/2
      LQ=(L*L-L)/2
      LM=L+MQ
   62 IF( ABS(A(LM))-THR) 130,65,65
   65 IND=1
      LL=L+LQ
      MM=M+MQ
      X=0.5*(A(LL)-A(MM))
   68 Y=-A(LM)/ SQRT(A(LM)*A(LM)+X*X)
      IF(X) 70,75,75
   70 Y=-Y
   75 SINX=Y/ SQRT(2.0*(1.0+( SQRT(1.0-Y*Y))))
      SINX2=SINX*SINX
   78 COSX= SQRT(1.0-SINX2)
      COSX2=COSX*COSX
      SINCS =SINX*COSX
C
C         ROTATE L AND M COLUMNS
C
      ILQ=N*(L-1)
      IMQ=N*(M-1)
      DO 125 I=1,N
      IQ=(I*I-I)/2
```

```
      IF(I-L) 80,115,80
   80 IF(I-M) 85,115,90
   85 IM=I+MQ
      GO TO 95
   90 IM=M+IQ
   95 IF(I-L) 100,105,105
  100 IL=I+LQ
      GO TO 110
  105 IL=L+IQ
  110 X=A(IL)*COSX-A(IM)*SINX
      A(IM)=A(IL)*SINX+A(IM)*COSX
      A(IL)=X
  115 IF(MV-1) 120,125,120
  120 ILR=ILQ+I
      IMR=IMQ+I
      X=R(ILR)*COSX-R(IMR)*SINX
      R(IMR)=R(ILR)*SINX+R(IMR)*COSX
      R(ILR)=X
  125 CONTINUE
      X=2.0*A(LM)*SINCS
      Y=A(LL)*COSX2+A(MM)*SINX2-X
      X=A(LL)*SINX2+A(MM)*COSX2+X
      A(LM)=(A(LL)-A(MM))*SINCS+A(LM)*(COSX2-SINX2)
      A(LL)=Y
      A(MM)=X
C
C         TESTS FOR COMPLETION
C
C         TEST FOR M = LAST COLUMN
C
  130 IF(M-N) 135,140,135
  135 M=M+1
      GO TO 60
C
C         TEST FOR L = SECOND FROM LAST COLUMN
C
  140 IF(L-(N-1)) 145,150,145
  145 L=L+1
      GO TO 55
  150 IF(IND-1) 160,155,160
  155 IND=0
      GO TO 50
C
C         COMPARE THRESHOLD WITH FINAL NORM
C
  160 IF(THR-ANRMX) 165,165,45
C
C         SORT EIGENVALUES AND EIGENVECTORS
C
  165 IQ=-N
      DO 185 I=1,N
      IQ=IQ+N
      LL=I+(I*I-I)/2
      JQ=N*(I-2)
      DO 185 J=I,N
      JQ=JQ+N
      MM=J+(J*J-J)/2
      IF(A(LL)-A(MM)) 170,185,185
  170 X=A(LL)
      A(LL)=A(MM)
```

```
      A(MM)=X
      IF(MM-1) 175,185,175
  175 DO 180 K=1,N
      ILR=IO+K
      IMR=IO+K
      X=R(ILR)
      R(ILR)=R(IMR)
  180 R(IMR)=X
  185 CONTINUE
      RETURN
      END
C                        **********  AUGUST/1973  **********
```

```
C        ******************************************************************************
C
C        DATA FOR
C        PROGRAM 3.4 : POP
C
C        ******************************************************************************
 62 20  11  11
 10  32
  3  12
  5
 63
 0  0  0  0
 1  1  1  1
 2  2  2  2
 0  0  0  0
 1  1  1  1
 2  2  2  2
 0  0  0  0
 1  1  1  1
 2  2  2  2
 0  0  0  0
 1  1  1  1
 2  2  2  2
```

PARAMETERS USED FOR THIS RUN

RANGE OF AGES USED: LX = 10   MX = 32

RANGE OF INCOMES USED: LS = 3   MS = 12

NUMBER OF EIGEN VALUES USED: = 5

RANGE OF YEARS 1963 TO 1973

NUMBER OF YEARS PER ESTIMATION: TE = 11

NUMBER OF ESTIMATIONS: LL = 1

EIGENPAIRS OF MTM(11,11)   BORDERING 0 ROWS AND COLUMNS

EIGENVALUE = 174.05992

EVT(I, 1)

0.239829	0.272251	0.297082	0.315881	0.327710
0.333164	0.331738	0.323744	0.309388	0.286738
0.262928				

EIGENVALUE = 3.04108

EVT(I, 2)

-0.097195	-0.033286	-0.072723	-0.061406	-0.059136
-0.066506	-0.089405	-0.049618	-0.026633	-0.274218
0.941332				

EIGENVALUE = 0.06293

EVT(I, 3)

0.057430	-0.235166	-0.390337	-0.465885	-0.287360
-0.080192	0.219742	0.380751	0.440560	0.304240
0.055381				

EIGENVALUE = 0.02424

EVT(I, 4)

0.834103	-0.262712	0.192624	-0.306187	0.094858
0.014052	0.041937	-0.115356	-0.068572	-0.274605
-0.005339				

EIGENVALUE = 0.00879

EVT(I, 5)

-0.097950	-0.066195	-0.038261	0.121791	-0.031294
0.035732	0.264228	-0.534619	0.686787	-0.367915
-0.095616				

EIGENPAIRS OF MTM(11,11)   BORDERING 1 ROWS AND COLUMNS

EIGENVALUE = 152.19181

EVT(I, 1)

0.000000	0.291157	0.317873	0.337928	0.350545
0.356410	0.354991	0.346258	0.330805	0.307788
0.000000				

EIGENVALUE = 0.15333

EVT(I, 2)

0.000000	-0.100845	-0.080960	-0.148951	-0.150109
-0.102633	0.049189	-0.076567	-0.170180	0.929800
0.000000				

EIGENVALUE = 0.05688

EVT(I, 3)

0.000000	-0.163576	-0.381445	-0.431903	-0.243686
-0.028239	0.230417	0.449320	0.568850	-0.049509
0.000000				

EIGENVALUE = 0.00866

EVT(I, 4)

0.000000	-0.160658	0.194413	-0.252446	0.512799
-0.448370	0.293849	-0.459775	0.332684	-0.015709
0.000000				

EIGENVALUE = 0.00729

EVT(I, 5)

0.000000	-0.030626	-0.181117	0.411980	-0.423063
0.236193	0.133403	-0.596849	0.432582	0.024681
0.000000				

EIGENPAIRS OF MIM(11,11)   BORDERING 2 ROWS AND COLUMNS

EIGENVALUE = 124.88297

EVT(I, 1)

0.000000	0.000000	0.350914	0.373066	0.387004
0.393472	0.391866	0.382267	0.365243	0.000000
0.000000				

EIGENVALUE = 0.05522

EVT(I, 2)

0.000000	0.000000	-0.413275	-0.464572	-0.280449
-0.058980	0.218693	0.434881	0.542493	0.000000
0.000000				

EIGENVALUE = 0.00856

EVT(I, 3)

0.000000	0.000000	0.029651	-0.100552	0.487961
-0.477947	0.098854	-0.507049	0.507422	0.000000
0.000000				

EIGENVALUE = 0.00813

EVT(I, 4)

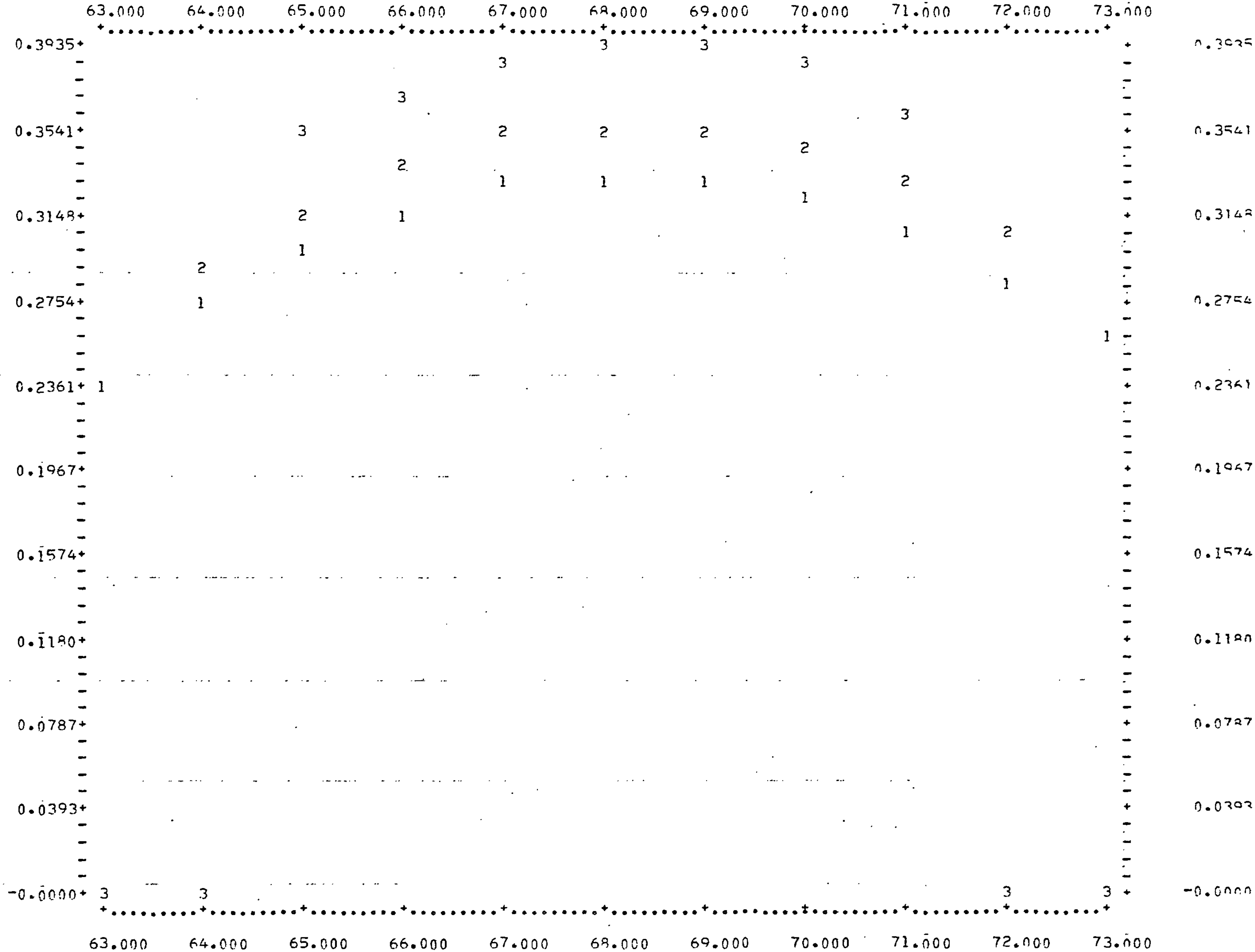
0.000000	0.000000	0.338733	-0.530340	0.224210
-0.261647	0.476163	0.205297	-0.465180	0.000000
0.000000				

EIGENVALUE = 0.00663

EVT(I, 5)

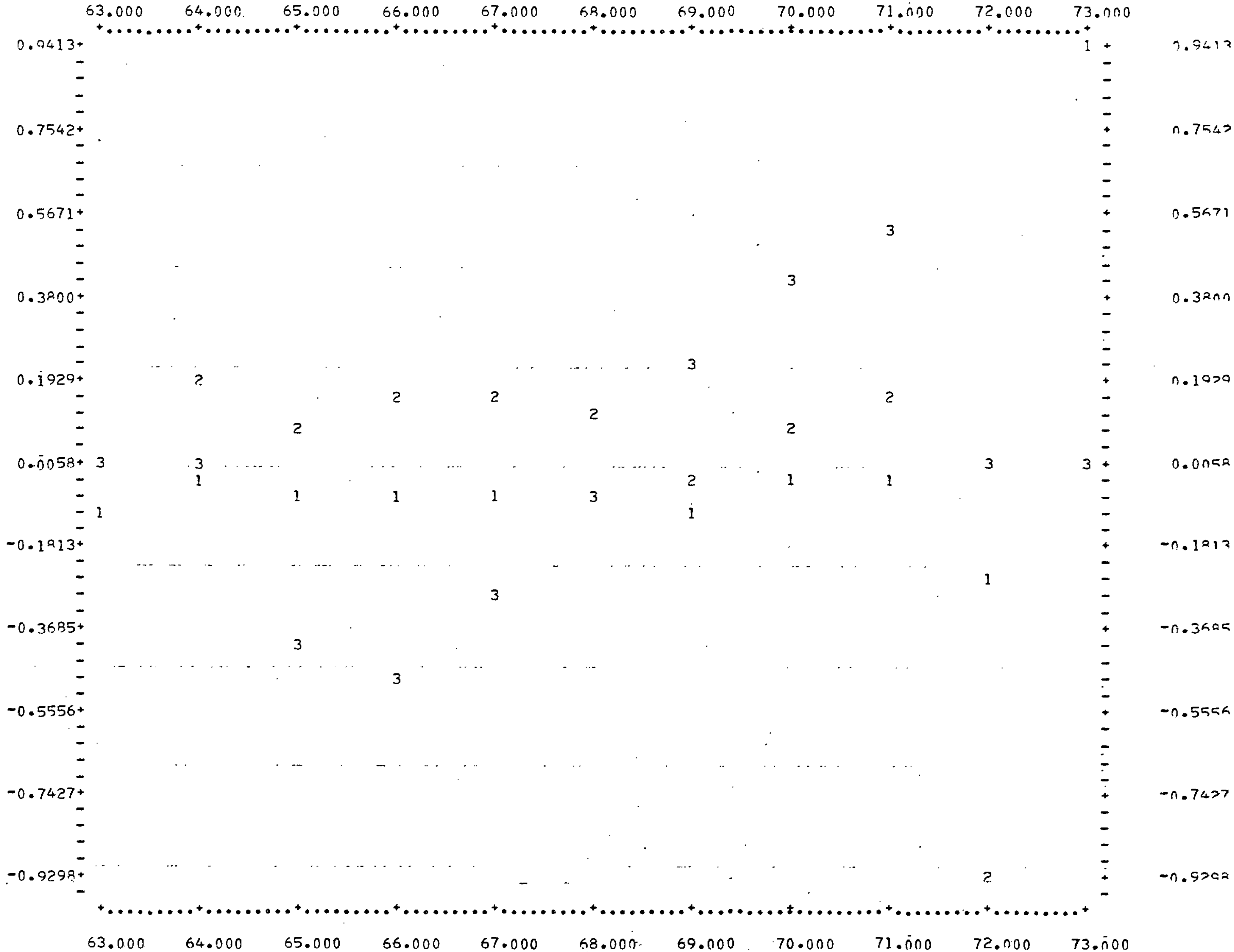
0.000000	0.000000	0.099373	0.054112	-0.513592
0.208384	0.627400	-0.532487	0.053130	0.000000
0.000000				

CHART 1



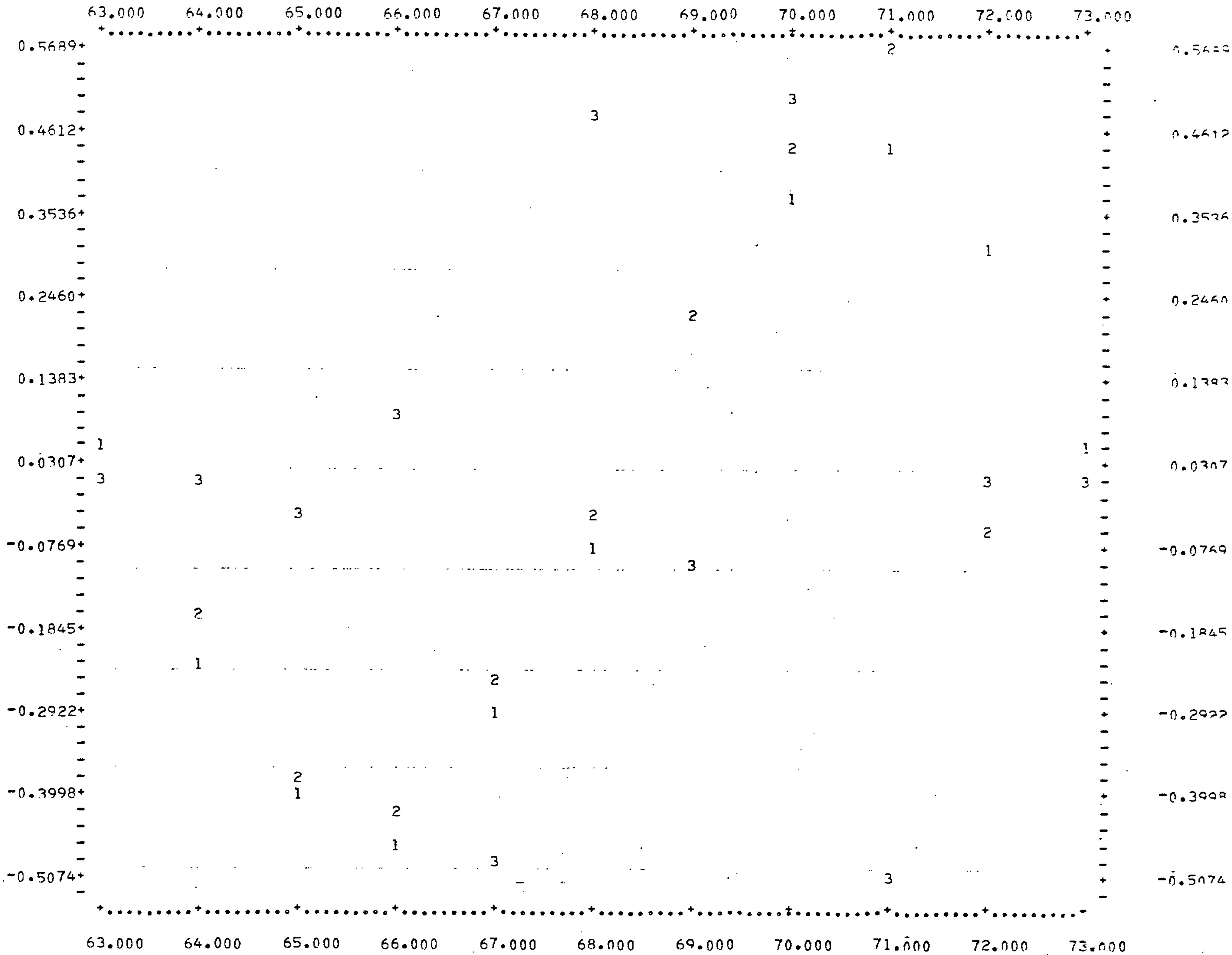
63.000 64.000 65.000 66.000 67.000 68.000 69.000 70.000 71.000 72.000 73.000
0.3935
0.3541
0.3148
0.2754
0.2361
0.1967
0.1574
0.1180
0.0787
0.0393
-0.0000

CHART 2



63.000
64.000
65.000
66.000
67.000
68.000
69.000
70.000
71.000
72.000
73.000
0.9413
0.7542
0.5671
0.3800
0.1929
0.0058
-0.1813
-0.3685
-0.5556
-0.7427
-0.9298

CHART 3



63.000
64.000
65.000
66.000
67.000
68.000
69.000
70.000
71.000
72.000
73.000
0.5689
0.4612
0.3536
0.2460
0.1383
0.0307
-0.0769
-0.1845
-0.2922
-0.3998
-0.5074

CHART 4

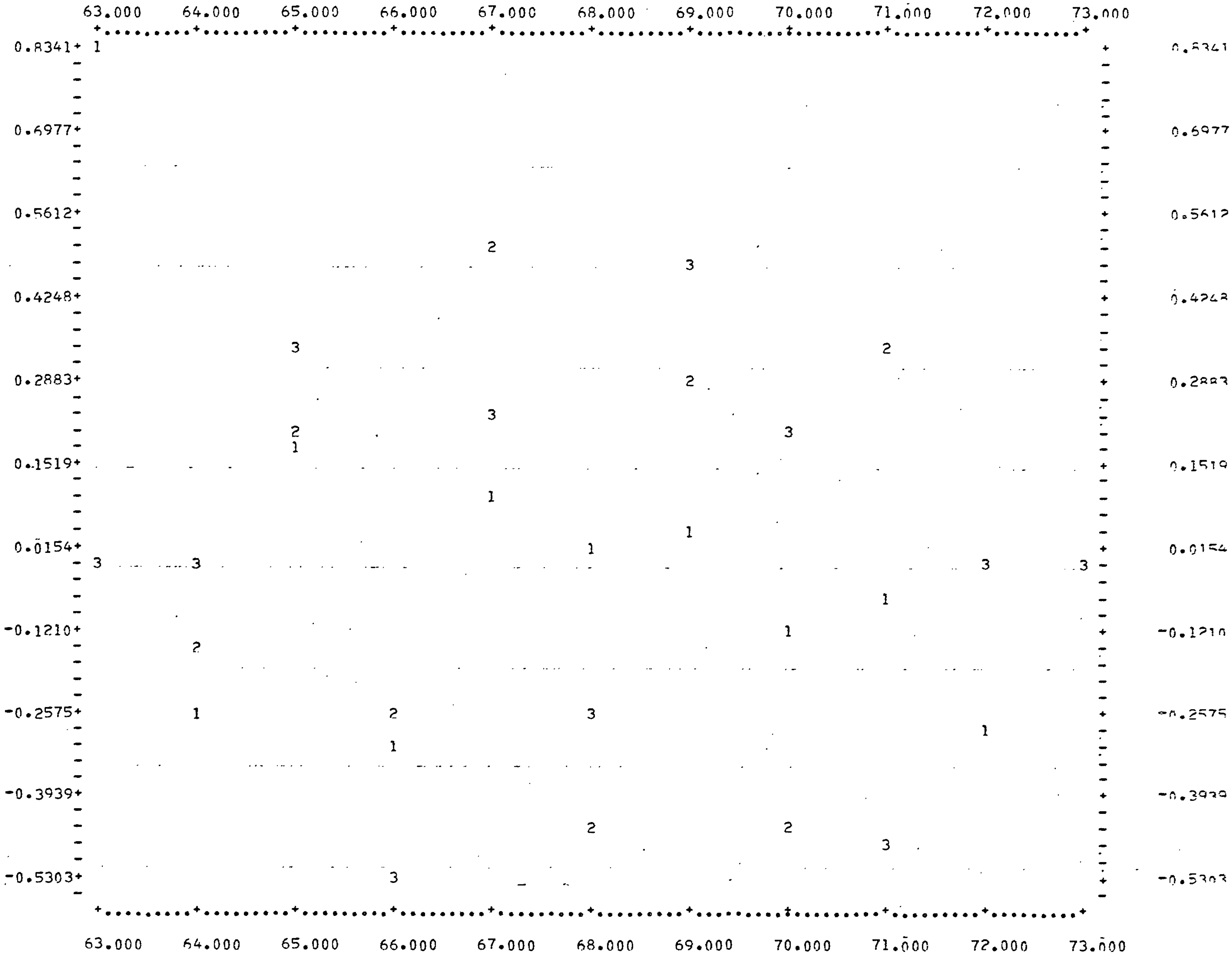
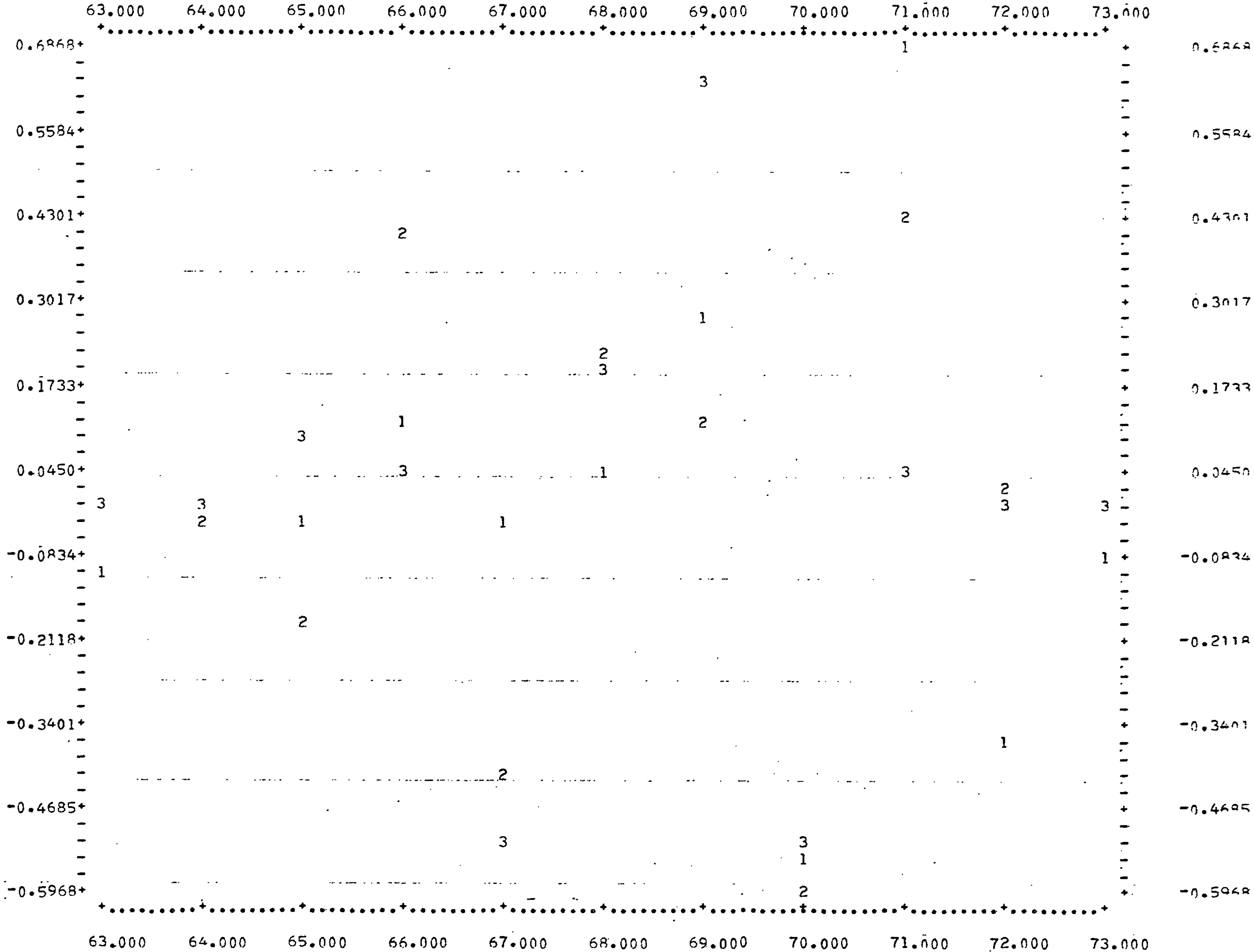

63.000
64.000
65.000
66.000
67.000
68.000
69.000
70.000
71.000
72.000
73.000
0.8341
0.6977
0.5612
0.4248
0.2883
0.1519
0.0154
-0.1210
-0.2575
-0.3939
-0.5303

CHART 5


63.000 64.000 65.000 66.000 67.000 68.000 69.000 70.000 71.000 72.000 73.000
0.6868
0.5584
0.4301
0.3017
0.1733
0.0450
-0.0834
-0.2118
-0.3401
-0.4685
-0.5968

3.5. <u>Estimating the Mobility from Data</u>

As stated above the economic mobility can be estimated using Program 3.5. Program 3.4 may be used to estimate the dominant modes. Graphs for both time and age-income components obtained using Programs 3.5, 3.4 and 3.7 from actual data are included in Chapter III.

The output of Program 3.5 and that of Program 3.4 immediately thereafter result from an aggregation experiment in which MU is calculated from an intermediate-level of aggregation of population resulting from the simulated mobility function.

```
$JOB   ACCT-NUM,VERNER,TIME=60
C      ****************************************************************
C      PROGRAM 3.5 : REALMU
C
C
C      ****************************************************************
```

```
3JOB REALMU; CHARGE VERNER; CLASS 3
3USER 080002/VHJ
3PROCESSTIME=500;
PRINTLIMIT=9000;
3BEGIN
3COMPILE REALMU FORTRAN
3FILE FILE1 (KIND=READER,MAXRECSIZE=80,UNITS=CHARACTERS,BLOCKSIZE=80);
3FILE FILE2(KIND=PACK,TITLE=DATA/XXMU5,MAXRECSIZE=11,BLOCKSIZE=330,
AREASIZE=42,AREAS=1,FLEXIBLE,SAVEFACTOR=10);
3FILE FILE4(KIND=PACK,TITLE=DATA/XXGRAL,FILETYPE=7);
3FILE FILE5(KIND=PACK,TITLE=DATA/XXIT,  FILETYPE=7);
3FILE FILE8(KIND=PACK,TITLE=DATA/XXDSCN,FILETYPE=7);
3FILE FILE9 (KIND=PACK,TITLE=DATA/XXCSRS,FILETYPE=7);
3DATA



3DATA FILE1



3END JOB
```

```
C       ************************************************************
C
C       PROGRAM : 3.5
C       TITLE : REALMU
C       PURPOSE : TO ESTIMATE MU FROM AGGREGATED POPULATIONS
C                 1. REAL DATA (TYPE=.TRUE.)
C                 2. SIMULATED POPULATION (TYPE=.FALSE.)
C       DATA : 1. DATA/XXIT,DATA/XXDSCN,DATA/XXCSRS
C              2. DATA/XXGRAL, DATA/XXDSCN,DATA/XXCSRS
C       OUTPUT : DATAXXMU2
C
C       ************************************************************
C       ************************************************************
C
C       THIS PROGRAM IS USED TO ESTIMATE THE ECONOMIC MOBILTITY
C       PARAMETER MU.  AGE- TIME- AND INCOME-SPECIFIC VALUES OF THE
C       POPULATION ARE REQUIRED.
C
C       ************************************************************
        INTEGER XM,SM,TM,XE,SE,TE,XMP1,SMP1,TMP1,SA,SI,S3,X3
C       ************************************************************
C
C       INTEGER  XM,SM,TM:  NUMBER OF INTERVALS IN AGE-INCOME-TIME
C                           DIMENSIONS FOR WHICH POPULATION IS GIVEN
C
C       INTEGER  XE,SE,TE:  NUMBER OF INDEPENDENT VARIABLE POINTS
C                           FOR WHICH DENSITIES ARE CALCULATED AND USED.
C
C       REAL XX(XM+1),SS(SM+1),TT(TM),XL(XE),SL(SE),X,S,T
C       REAL PP(XM,SM),PT(XE,SM),P2(SM),P4(SE),Q(3*(SM+1)),A2(3*(SM+1),9)
C       REAL PI(XM,SM,TM),P(XE,SM,TE),P1(XE,SE,TE)
C       REAL R1(XE),PS(XM),Q1(3*(XM+1)),H1(XM+1),A1(3*(XM+1),9)
C       REAL R(XE,TM),MU(XE,SE,TE)
C       REAL K(2,XE),H(XE),COF(4,XE),A(3,XE),D(XE)
C
C       ************************************************************
        REAL XX(20),SS(20),TT(11),XL(62),SL(20),X,S,T
        REAL PP(19,19),PT(62,19),P2(19),P4(20),Q(60),A2(60,9)
        REAL PI(19,19,11),P(62,19,11),P1(62,20,11)
        REAL R1(62),PS(19),Q1(60),H1(20),A1(60,9)
        REAL R(62,11),MU(62,20,11)
        REAL K(2,62),H(62),COF(4,62),A(3,62),D(62)
        INTEGER IYRS(11)
        DATA IYRS/1963,1964,1965,1966,1967,1968,1969,1970,1971,1972,1973/
        LOGICAL PRTSDN,PRTPDN
        LOGICAL PRTR,PRTPX,PRTPXT,PRTXTR,PRTMU,TYPE
        NAMELIST/DNSITY/PRTSDN,PRTPDN
        NAMELIST/VALUES/PRTR,PRTPX,PRTPXT,PRTXTR,PRTMU
        NAMELIST/POP/TYPE
        NAMELIST/PRTLIM/IPS,IPE
        READ(1,DNSITY)
        READ(1,VALUES)
        READ(1,POP)
        READ(1,PRTLIM)
        READ(1,101) XM,SM,TM
        XMP1-XM+1
        SMP1-SM+1
        TMP1-TM+1
        S3=3*SMP1
```

```
      X3=3*XMP1
      READ(1,101) XF,SF,TF
  101 FORMAT(3I3)
      READ(1,102) (XX(I),I=1,XMP1)
      READ(1,102) (SS(I),I=1,SMP1)
  102 FORMAT(20F4.0)
      SA=IFIX(XX(1)+.01)
      SI=IFIX(SS(1)+.01)
      DO 4 I=1,XF
         XL(I)=SA+I-1
    4    CONTINUE
      DO 5 I=1,SF
         SL(I)=SI+I-1
    5    CONTINUE
      DO 6 I=1,TM
         TT(I)=I
    6    CONTINUE
      WRITE(6,201) XM,SM,TM
      WRITE(6,203)
      WRITE(6,103) (XX(I),I=1,XMP1)
      WRITE(6,204)
      WRITE(6,103) (SS(I),I=1,SMP1)
      WRITE(6,202) XF,SF,TF
      WRITE(6,205)
      WRITE(6,103) (XL(I),I=1,XF)
      WRITE(6,206)
      WRITE(6,103) (SL(I),I=1,SF)
  103 FORMAT(10F6.0,/)
  201 FORMAT('1',/,' NUMBER OF AGE GROUPS = ',I3,//,' NUMBER OF INCOME
     &      ,'GROUPS = ',I3,//,' NUMBER OF TIME VALUES = ',I3,//)
  202 FORMAT('1',/,' NUMBER OF AGE DENSITITES = ',I3,//,' NUMBER OF INC
     &ME DENSITIES = ',I3,//,' NUMBER OF TIME VALUES PER ESTIMATION = '
     &I3,//)
  203 FORMAT(///,' BOUNDARIES OF AGE GROUPS',/)
  204 FORMAT(///,' BOUNDARIES OF INCOME GROUPS',/)
  205 FORMAT(///,' BOUNDARIES OF AGE DENSITIES',/)
  206 FORMAT(///,' BOUNDARIES OF INCOME DENSITIES',/)
C     ******************************************************************
C
C              CALCULATE DEATH RATES
C
C     ******************************************************************
      CALL CALCR(R,XF,TM,SA)
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
C
      IF(.NOT.PRTR) GO TO 71
      WRITE(6,125)
      WRITE(6,120)
      DO 26 I=IPS,IPF
         WRITE(6,116) XL(I) ,(R(I,IT),IT=1,6)
   26    CONTINUE
      WRITE(6,121)
      DO 27 I=IPS,IPF
         WRITE(6,116) XL(I) ,(R(I,IT),IT=7,TM)
   27    CONTINUE
C
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
   71 CONTINUE
C     *****************************************************************
C
```

```
C     INPUT DATA AND SPLINEGRAM TO ESTIMATE DENSITY IS X VARIABLE
C
C     ************************************************************************
C
      IF (TYPE) CALL INPOP(XM,XMP1,SM,SMP1,TM,XX,SS,PI)
      ITS=
      ITSP5=5
      ITSP6=6
      ITE=TE-1
   49 ITS=ITS+1
      ITSP5=ITSP5+1
      ITE=ITE+1
      ITSP6=ITSP6+1
      DO 13 IT=1,TE
            LT=ITS+IT-1
            IF (.NOT. TYPE) GO TO 15
            DO 10 I=1,XM
                  DO 10 J=1,SM
   10                   PP(I,J)=PI(I,J,LT)
            GO TO 14
   15       IF (.NOT.TYPE) CALL INAGGR(XM,XMP1,SM,SMP1,IT,XX,SS,SA,SI,PP
            DO 11 IX=1,XM
                  DO 11 IS=1,SM
   11                   PI(IX,IS,LT)=PP(IX,IS)
   14       CONTINUE
            CALL DENS(XM,XMP1,X3,SM,XE,XX,PP,XL,PT,R1,PS,Q1,H1,A1)
            DO 12 LX=1,XE
                  DO 12 LS=1,SM
                        P(LX,LS,IT)=PT(LX,LS)
   12                   CONTINUE
   13       CONTINUE
      IF (.NOT.PRTPDN) GO TO 77
      DO 85 I=IPS,IPE
            WRITE(6,131) XL(I)
            WRITE(6,118) (IYRS(IT),IT=ITS,ITSP5)
            DO 86 J=1,SM
                  WRITE(6,126) SS(J),(P(I,J,IT),IT=1,6)
   86             CONTINUE
            WRITE(6,119) (IYRS(IT),IT=ITSP6,ITE)
            DO 85 J=1,SM
                  WRITE(6,126) SS(J),(P(I,J,IT),IT=7,TE)
   85       CONTINUE
   77 CONTINUE
C     ************************************************************************
C
C     USE SPLINE GRAM TO DETERMINE DENSITY IN S DIRECTION
C
C     ************************************************************************
      DO 84 IT=1,TE
            DO 84 IX=1,XE
                  DO 81 IS=1,SM
                        P2(IS)=P(IX,IS,IT)
   81                   CONTINUE
      CALL ISPLIN(SM,S3,P2,Q,SS,SMP1,H,A2)
      CALL RHO(Q,H,SM,SMP1,S3,1.,SS,SE,P4)
      DO 84 IS=1,SE
            P1(IX,IS,IT)=P4(IS)
   84       CONTINUE
      IF (.NOT.PRTSDN) GO TO 78
      DO 89 I=IPS,IPE
```

```
              WRITE(6,132) XL(I)
              WRITE(6,118) (IYRS(IT),IT=ITS,ITSP5)
              DO 87 J=1,SE
                   WRITE(6,126) SL(J),(P1(I,J,IT),IT=1,6)
     87            CONTINUE
              WRITE(6,119) (IYRS(IT),IT=ITSP6,ITE)
              DO 88 J=1,SE
                   WRITE(6,126) SL(J),(P1(I,J,IT),IT=7,TE)
     88            CONTINUE
     78 CONTINUE
C       ******************************************************************
C
C       NOW OBTAIN DP/DX
C
C       ******************************************************************
        DO 60 IX=1,XE
        K(1,IX)=XL(IX)
     60 CONTINUE
        CALL SETUP(XE,A,H,K)
        DO 61 IS=1,SE
        DO 61 IT=1,TE
        DO 62 IX=1,XE
        K(2,IX)=P1(IX,IS,IT)
     62 CONTINUE
        CALL SOLVE(A,K,H,XE,D,COF)
        DO 63 IX=1,XE
        X=XL(IX)
     63 MU(IX,IS,IT)=COF(3,IX)+X*(2.*COF(2,IX)+X*3.*COF(1,IX))
     61 CONTINUE
C       PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPPINT
C
        IF(.NOT.PRTPX) GO TO 72
        DO 21 I=IPS,IPE
             WRITE(6,133) XL(I)
             WRITE(6,118) (IYRS(IT),IT=ITS,ITSP5)
             DO 20 J=1,SE
                  WRITE(6,126) SL(J),(MU(I,J,IT),IT=1,6)
     20           CONTINUE
             WRITE(6,119) (IYRS(IT),IT=ITSP6,ITE)
             DO 21 J=1,SE
                  WRITE(6,126) SL(J),(MU(I,J,IT),IT=7,TE)
     21      CONTINUE
C
C       PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
     72 CONTINUE
C       *************************************************************
C
C          SPLINE  P(X,S,T)  AGAINST  T  AND DIFFERENTIATE THE SPLINE
C            MU(X,S,T)  IS  DP/DX+DP/DT
C
C       *************************************************************
        DO 30 IT=1,TE
             LT=ITS+IT-1
     30      K(1,IT)=TT(LT)
        CALL SETUP(TM,A,H,K)
        DO 31 IX=1,XE
        DO 31 IS=1,SE
        DO 32 IT=1,TE
        K(2,IT)=P1(IX,IS,IT)
     32 CONTINUE
```

```
      CALL SOLVF(A,K,H,TM,D,COF)
      DO 33 IT=1,TE
         LT=ITS+IT-1
         T=TT(LT)
   33 MU(IX,IS,IT)=MU(IX,IS,IT)+(COF(3,IT)+T*(2.*COF(2,IT)+T*3.*COF(1,IT
     1)))
   31 CONTINUE
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
C
      IF(.NOT.PRTPXT) GO TO 73
      DO 23 I=IPS,IPE
         WRITE(6,134) XL(I)
         WRITE(6,118) (IYRS(IT),IT=ITS,ITSP5)
         DO 22 J=1,SE
            WRITE(6,126) SL(J),(MU(I,J,IT),IT=1,6)
   22       CONTINUE
         WRITE(6,119) (IYRS(IT),IT=ITSP6,ITE)
         DO 23 J=1,SE
            WRITE(6,126) SL(J),(MU(I,J,IT),IT=7,TE)
   23       CONTINUE
C
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
   73 CONTINUE
C     ******************************************************************
C
C          ADD IN THE DEATH RATE FACTOR
C
C     ******************************************************************
      DO 34 LX=1,XE
      DO 34 LS=1,SE
      DO 35 IT=1,TE
         LT=ITS+IT-1
   35                  MU(LX,LS,IT)=MU(LX,LS,IT)+R(LX,LT)*P1(LX,LS,IT)
   34 CONTINUE
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
C
      IF(.NOT.PRTXTR) GO TO 74
      DO 25 I=IPS,IPE
         WRITE(6,135) XL(I)
         WRITE(6,118) (IYRS(IT),IT=ITS,ITSP5)
         DO 24 J=1,SE
            WRITE(6,126) SL(J),(MU(I,J,IT),IT=1,6)
   24       CONTINUE
         WRITE(6,119) (IYRS(IT),IT=ITSP6,ITE)
         DO 25 J=1,SE
            WRITE(6,126) SL(J),(MU(I,J,IT),IT=7,TE)
   25       CONTINUE
C
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
   74 CONTINUE
C     ****************************************************************
C
C        SPLINE  DP/DX+DP/DT+R(X,S,T)*P(X,S,T)  AGAINST  S  AND
C        INTEGRATE THE SPLINE FROM  S  TO THE MAXIMUM VALUE OF  S.
C                 MU(X,S,T) IS  END CONDITION PLUS THIS INTEGRAL
C
C     ****************************************************************
      DO 40 IS=1,SE
      K(1,IS)=SL(IS)
   40 CONTINUE
```

```
      CALL SETUP(SE,A,H,K)
      DO 41 IX=1,XE
      DO 41 IT=1,TE
          IT=ITS+IT-1
      DO 42 IS=1,SE
      K(2,IS)=MU(IX,IS,IT)
   42 CONTINUE
      CALL SOLVE(A,K,H,SE,0,COF)
      MU(IX,SE,IT)=0.
      DO 43 IS=2,SE
      I=SE+1-IS
      S1=SL(I)
      S2=S1+1.
              MU(IX,I,IT)=MU(IX,I+1,IT)+(((S2*COF(1,I)/4.+COF(2,I)/3.)*
     &        S2+COF(3,I)/2.)*S2+COF(4,I))*S2-(((S1*COF(1,I)/4.
     &        +COF(2,I)/3.)*S1+COF(3,I)/2.)*S1+COF(4,I))*S1
   43         CONTINUE
   41 CONTINUE
      DO 55 KS=1,SE
      DO 55 KT=1,TE
      DO 56 KX=1,XE
      IF (P1(KX,KS,KT).LE.0.0) GO TO 57
      IF  (KX.EQ.XE) GO TO 58
      IF (P1(KX,KS,KT).LE.0.1*P1(KX+1,KS,KT)) GO TO 57
   58         CONTINUE
      IF (KX.EQ.1) GO TO 59
      IF (P1(KX,KS,KT).LE. 0.1*P1(KX-1,KS,KT)) GOTO 57
   59         CONTINUE
      MU(KX,KS,KT)=  MU(KX,KS,KT) / P1(KX,KS,KT)
      GO TO 56
   57 MU(KX,KS,KT)=0.0
   56 CONTINUE
   55         CONTINUE
C
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
C
      IF(.NOT.PRTMU) GO TO 75
      DO 29 I=IPS,IPE
          WRITE(6,137) XL(I)
          WRITE(6,118)  (IYRS(IT),IT=ITS,ITSP5)
          DO 28 J=1,SE
              WRITE(6,126) SL(J),(MU(I,J,IT),IT=1,6)
   28         CONTINUE
          WRITE(6,119)  (IYRS(IT),IT=ITSP6,ITE)
          DO 29 J=1,SE
              WRITE(6,126) SL(J),(MU(I,J,IT),IT=7,TE)
   29         CONTINUE
C
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
   75 CONTINUE
  116 FORMAT(I4,5X,6F10.6)
  118 FORMAT(' INCOME/TIME ',6(I4,6X),//)
  119 FORMAT(//,' INCOME/TIME ',6(I4,6X),//)
  120 FORMAT('  AGE/TIME    1963        1964        1965        1966        1967
     &      1968',//)
  121 FORMAT(//,'  AGE/TIME    1969        1970        1971        1972        19
     &73',//)
  125 FORMAT('1','DEATH RATES',//)
  126 FORMAT(I4,5X,6F10.6)
  131 FORMAT('1','DENSITY IN X AT AGE = ',I2,//)
```

```
  132 FORMAT('1','DENSITY IN X & S AT AGE = ',I2,//)
  133 FORMAT('1',' DP/DX   AT AGE = '          ,I2,//)
  134 FORMAT('1','DP/DX + DP/DT   AT AGE =' ,I2,//)
  135 FORMAT('1','DP/DX + DP/DT + R*P AT AGE = ',I2,//)
  137 FORMAT('1','ECONOMIC MOBILITY AT AGE = ',I2,//)
      DO 90 I=1,XF
           DO 90 J=1,SF
                WRITE(2) (MU(I,J,KK),KK=1,TF)
   90           CONTINUE
      IF (TM.GE.ITS+TE) GO TO 49
      LOCK 2

      STOP
      END

      SUBROUTINE CALCR(R,NX,NT,SA)
C     ******************************************************************
C
C     THIS SUBROUTINE CALCULATES DEATH RATES FOR THE YEARS
C     1963 TO 1963+NT-1
C
C     R(I,J) = FRACTION OF PEOPLE OF AGE  I+SA  DYING AT TIME  J+1962
C
C     ******************************************************************
      INTEGER NX,NT,SA
      REAL R(NX,NT)
      REAL XY(20),XZ(20)
      INTEGER YY,D
      INTEGER TEST/'R'/
      I1=SA+203
      DO 202 I=1,I1
  202 READ(8,100)XX
  100 FORMAT(A1)
      I1=SA+631
      DO 203 I=1,I1
  203 READ(9,100)XX
      DO 206 I=1,20
           XY(I)=0.
  206      CONTINUE
C     READ DATA FOR EACH YEAR
      DO 200 LT=1,NT
C     READ DEATHS FOR EACH YEAR
      DO 221 LX=1,NX
      READ(8,101)D
  101 FORMAT(T29,I5)
      IF (LX.LE.70-SA) GO TO 210
C     ADJUSTMENT TO ACCOMMODATE LACK OF POPLUATION DATA ABOVE AGE
C     69  FOR YEARS  1967 TO 1970.
      IF(LT.GT.4.AND.LT.LE.8)GO TO 211
C     READ POPULATION FOR EACH YEAR
      READ(9,102)P,YY
  102 FORMAT(12X,F9.1,55X,A1)
C     POPULATION IN NON-CENSUS YEARS MUST BE MULTIPLIED BY  1000.
      IF(YY.EQ.TEST)P=P*1000.
C     LINEAR EXTRAPOLATION FROM 1966 DATA IS USED FOR 1967 TO 1970.
      J=LX-51
      XZ(J)=P-XY(J)
      XY(J)=P
      GO TO 201
  211 J=LX-51
```

```
      P=XY(J)+XZ(J)
      XY(J)=P
      GO TO 201
  210 READ(9,102)P,YY
      IF(YY.EQ.TEST)P=P*1000.
  201 R(LX,LT)=D/P
  221 CONTINUE
         IF (LT.EQ.NT) GO TO 200
C     CLEAR REMAINING DATA ON DEATHS.
      NN=101-NX
      IF (LT.GT.2) NN=NN-1
      DO 204 I=1,NN
  204 READ(8,100)XX
C     CLEAR REMAINING DATA ON POPULATION.
      NM=90-NX
      IF (LT.GT.4.AND.LT.LE.8) NM=SA
      DO 205 I=1,NM
  205 READ(9,100)XX
  200 CONTINUE
      RETURN
      END

      SUBROUTINE SETUP(N,A,H,K)
C     ****************************************************************
C
C     THIS ROUTINE SETS UP A TRIDIAGONAL MATRIX OF THE SPLINE EQUATION
C     IN A  3 BY N  ARRAY  -  FIRST ROW IS SUPERDIAGONAL
C                             SECOND ROW IS DIAGONAL
C                             THIRD ROW IS SUBDIAGONAL
C     AND THEN DECOMPOSES A TO  LU  SO THAT
C        FIRST ROW IS SUPERDIAGONAL OF U, SECOND ROW IS DIAGONAL OF U
C       THIRD ROW IS SUBDIAGONAL OF L, DIAGONAL OF L IS UNITY
C
C     ****************************************************************
      REAL A(3,N),H(N),K(2,N)
      NM1=N-1
      H(2)=K(1,2)-K(1,1)
      DO 10 I=2,NM1
          H(I+1)=K(1,I+1)-K(1,I)
          A(1,I)=H(I+1)/(H(I+1)+H(I))
          A(2,I)=2.0
          A(3,I)=1-A(1,I)
   10     CONTINUE
      A(1,1) = A(1,2)
      A(2,1) = A(2,2)
      A(3,1) = A(3,2)
      A(1,N) = A(1,N-1)
      A(2,N) = A(2,N-1)
      A(3,N) = A(3,N-1)
      A(1,2) = -2.
      A(2,2) = 2.
      A(3,2) = 0.
      A(1,N-1) = 0.
      A(2,N-1) = 2.
      A(3,N-1) = -2.
      NM1 - N-1
      DO 11 I=3,NM1
          A(3,I)=A(3,I)/A(2,I-1)
          A(2,I)=A(2,I)-A(3,I)*A(1,I-1)
   11     CONTINUE
```

```
      RETURN
      END

      SUBROUTINE SOLVE(A,K,H,N,D,COF)
C     ******************************************************************
C
C     THIS ROUTINE CALCULATES THE SECOND ORDER FINITE DIFFERENCES OF
C     THE SPLINE, AND THEN SOLVES  AM=D.  (BY FORWARD AND BACKWARD
C     SUBSTITUTION), PLACING M(VECTOR OF SECOND DERIVATIVES) IN D
C     END POINT CONDITIONS NOW USE THIRD ORDER FINITE DIFFERENCES
C     TO ESTIMATE THE THIRD ORDER DERIVATIVES AT X0+3H/2 AND XN-3H/2.
C
C     ******************************************************************
      REAL H(N),D(N),A(3,N),K(2,N),COF(4,N)
      D(2)=(K(2,2)-K(2,1))/H(2)
      NM1=N-1
      DO 12 I=2,NM1
           D(I+1)=(K(2,I+1)-K(2,I))/H(I+1)
           D(I)=6*(D(I+1)-D(I))/(H(I+1)+H(I))
   12      CONTINUE
      CA=-6./(H(2)*(H(2)+H(3))*(H(2)+H(3)+H(4)))
      CB=6./(H(2)*H(3)*(H(4)+H(3)))
      CC=-6./(H(3)*H(4)*(H(2)+H(3)))
      CD=6./(H(4)*(H(4)+H(3))*(H(2)+H(3)+H(4)))
      D(1)=-2.*H(3)*(CA*K(2,1)+CB*K(2,2)+CC*K(2,3)+CD*K(2,4))
      CA=-6./(H(N-2)*(H(N-2)+H(NM1))*(H(N-2)+H(NM1)+H(N)))
      CB=6./(H(N-2)*H(NM1)*(H(N)+H(NM1)))
      CC=-6./(H(NM1)*H(N)*(H(N-2)+H(NM1)))
      CD=6./(H(N)*(H(N)+H(NM1))*(H(N-2)+H(NM1)+H(N)))
      D(N)=2.*H(NM1)*(CA*K(2,N-3)+CB*K(2,N-2)+CC*K(2,NM1)+CD*K(2,N))
      T1 = D(2)
      T2 = D(NM1)
      D(2) = D(1)
      D(NM1) = D(N)
      DO 13 I=3,NM1
           D(I)=D(I)-A(3,I)*D(I-1)
   13      CONTINUE
      D(NM1) = D(NM1)/A(2,NM1)
      DO 14 I=3,NM1
           J=N+1-I
           D(J)=(D(J)-A(1,J)*D(J+1))/A(2,J)
   14      CONTINUE
      D(1) = (T1 - A(2,1)*D(2) - A(1,1)*D(3))/A(3,1)
      D(N) = (T2 - A(2,N)*D(NM1) - A(3,N)*D(N-2))/A(1,N)
      CALL POLLY(N,D,K,H,COF)
      RETURN
      END

      SUBROUTINE POLLY(N,M,K,H,COF)
C     ******************************************************************
C
C     THIS ROUTINE COMPUTES THE COEFFICIENTS OF THE SPLINE POLYNOMIAL
C       ON EACH SUBINTERVAL
C     K IS THE ARRAY OF DATA POINTS
C     H IS THE VECTOR OF SUBINTERVAL LENGTHS
C     M IS THE SOLUTION VECTOR TO THE EQUATION AM=D
C
C     ******************************************************************
      REAL M(N),K(2,N),H(N),COF(4,N)
      NM1=N-1
```

```
      DO 11 I=1,NM1
      COF(1,I)=(M(I+1)-M(I))/(6.*H(I+1))
      COF(2,I)=(K(1,I+1)*M(I)-K(1,I)*M(I+1))/(2.*H(I+1))
      DD=M(I+1)*K(1,I)*K(1,I)-M(I)*K(1,I+1)*K(1,I+1)
     &+2.*K(2,I+1)-2.*K(2,I)
      COF(3,I)=(DD/(2.*H(I+1)))+H(I+1)*(M(I)-M(I+1))/6.
      DD=M(I)*(K(1,I+1)**3)-M(I+1)*(K(1,I)**3)+6.*K(1,I+1)*K(2,I)
     &-6.*K(1,I)*K(2,I+1)+K(1,I)*M(I+1)*(H(I+1)**2)-K(1,I+1)*M(I)
     &*(H(I+1)**2)
      COF(4,I)=DD/(H(I+1)*6.)
   11 CONTINUE
      DO 16 J=1,4
   16 COF(J,N)=COF(J,NM1)
      RETURN
        END

      SUBROUTINE INPOP(XM,XMP1,SM,SMP1,TM,XX,SS,PI)
C     ************************************************************************
C
C     THIS ROUTINE SELECTS THE APPROPRIATE INCOME DATA FROM THE
C     FILE  XXIT  FOR USE IN ESTIMATING THE ECONOMIC MOBILITY.
C
C     AT PRESENT P(I,J) = NUMBER OF INDIVIDULAS BETWEEN AGES  XX(I)
C     AND  XX(I+1)  EARNING BETWEEN  SS(J)  AND SS(J+1) AT YEAR LT.
C
C     TO OBTAIN THIS  THE FOLLOWING CALCULATIONS ARE MADE:
C            1963 - 1965:    ADD NON-TAXABLE RETURNS TO TAXABLE RETURNS
C                            FOR INCOMES $0-$2,$2-$3,>$3
C
C            1966 - 1973:    REPLACE TAXABLE RETURNS BY ALL RETURNS
C                            FOR INCOMES $0-$2,$2-$3,$3-$4,$4-$5
C
C            1972 - 1973:    AGGREGATE THE TWO AGE GROUPS BELOW AGE 25
C
C     ************************************************************************
      INTEGER XM,XMP1,SM,SMP1,TM
      REAL XX(XMP1),SS(SMP1),PI(XM,SM,TM)
      INTEGER TAX
      INTEGER TITLE
      INTEGER TEST/'N'/
      INTEGER TEST1/'T'/
      DO 20 LT=1,TM
      NX=4
      IF (LT.LE.3) NX=3
      READ(5,200) TITLE
      DO 15 I=1,XM
           DO 10 J=1,SM
                READ(5,103) IB,IE,MM,MF,TAX,IAS,IAE
                PI(I,J,LT)=MM+MF
   10           CONTINUE
      READ(5,103) IB,IE,MM,MF,TAX,IAS,IAE
      IF (TAX.NE.TEST) GO TO 13
      PI(I,1,LT)=PI(I,1,LT)+MM+MF
      DO 11 J=1,NX
           READ(5,103) IB,IE,MM,MF,TAX,IAS,IAE
           PI(I,J,LT)=PI(I,J,LT)+MM+MF
   11      CONTINUE
      GO TO 15
   13 TU=MM+MF
      DO 14 J=1,NX
```

```
                  READ(5,103) IB,IE,MM,MF,TAX,IAS,IAF
                  PI(I,J,LT)=MM+MF
       14         CONTINUE
            PI(I,1,LT)=PI(I,1,LT)+TU
            READ(5,103) IB,IE,MM,MF,TAX,IAS,IAF
       15 CONTINUE
C           NOTE:  DATA FOR I=12  IS USED FOR CHECKING DATA TOTALS
       16 READ(5,103) IB,IE,MM,MF,TAX,IAS,IAF
            IF (IE.NE.99.OR.TAX.EQ.TEST1) GO TO 16
       20 CONTINUE
      103 FORMAT(2I2,4X,I7,3X,I7,4X,A1,9X,2I2)
      200 FORMAT(A1)
            RETURN
            END

            SUBROUTINE INAGGR(XM,XMP1,SM,SMP1,LT,XX,SS,SA,SI,PP)
C           ****************************************************************
C
C           THIS SUBROUTINE READS VALUES OF SIMULATED POPULATIONS FROM
C           STORED DATA, AND AGGREGATES ACCORDING TO AGE GROUPS AND INCOME
C           GROUPS READ IN CARD FILE.
C
C           FOR SAMPLE VALUES OF AGGREGATED POPULATION SET ICH=1
C
C           ****************************************************************
            INTEGER XM,XMP1,SM,SMP1,LT,SA,SI
            REAL XX(XMP1),SS(SMP1),PP(XM,SM)
            REAL PI(62,20)
            INTEGER ICH
            DATA ICH/1/
            ICH=0
            DO 10 I=1,62
       10         READ(4) (PI(I,J),J=1,20)
            DO 30 I=1,XM
                  I1=IFIX(XX(I)-SA+.01)
                  DO 30 J=1,SM
                        J1=IFIX(SS(J)-SI+.01)
                        SUM=0.
                        DO 28 I2=1,8
                              I3=I1+I2
                              IF(I3.GE.XX(I+1)-SA+.01) GO TO 28
                              DO 27 J2=1,8
                                    J3=J1+J2
                                    IF (J3.GE.SS(J+1)-SI+.01) GO TO 27
                                    SUM=SUM+PI(I3,J3)
       27                           CONTINUE
       28                     CONTINUE
       30               PP(I,J)=SUM
            IF (ICH.EQ.0) GO TO 40
            IF (LT.NE.1) GO TO 40
            WRITE(6,205)
            DO 50 J=1,4
            WRITE(6,104) ( PI(I,J),I=1,23)
       50         CONTINUE
            WRITE(6,206)
            DO 51 J=1,5
                  WRITE(6,104) (PP(I,J),I=1,5)
       51         CONTINUE
       40 CONTINUE
      104 FORMAT(5F12.6)
```

```
  205 FORMAT(//,' INTEGRALS IN SINGLE YEARS AND $1000',//)
  206 FORMAT(//,' INTEGRALS IN GROUPS ',//)
      RETURN
      END

      SUBROUTINE DENS(XM,XMP1,X3,SM,XF,XX,PP,XL,P, R,PS,Q,H,A)
      INTEGER XM,XMP1,X3,SM,XF
      REAL XX(XMP1),PP(XM,SM),XL(XF),P(XF,SM)
      REAL R(XF),PS(XM),Q(X3),H(XMP1),A(X3,9)
C     ***************************************************************
C
C     THIS SUBROUTINE YIELDS THE DENSITY IN THE X VARIABLE FROM
C     VALUES OF THE POPULATION IN AGE AND INCOME COHORTS.
C
C     THIS IS ACHIEVED BY CALLING THE SPLINEGRAM ROUTINE.  AS WELL
C     THERE IS THE OPTION OF COMPARING THE EXACT INTEGRAL OF THE
C     SPLINE OBTAINED WITH THE ORIGINAL (HISTOGRAM TYPE) DATA.
C
C     FOR THIS COMPARISON SET   ICH = 1
C     OTHERWISE SET   ICH = 0
      DATA ICH/1/
      ICH=0
C     ***************************************************************
C     FOR DENSITY IN SINGLE UNITS SET DELX = 1.
      DELX=1.
      IF(ICH.EQ.0) GOTO 18
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTP
      WRITE(6,106)
 106  FORMAT('1','THE GIVEN POPULATION IS')
      WRITE(6,102) ((PP(I,J),I=1,XM),J=1,SM)
  102 FORMAT(//,5F8.0)
   18 CONTINUE
      DO 15 J=1,SM
      DO 14 I=1,XM
   14 PS(I)=PP(I,J)
      CALL ISPLIN(XM,X3,PS,Q,XX,XMP1,H,A)
      CALL PHO(Q,H,XM,XMP1,X3,DELX,XX,XF,R)
      IF (ICH.EQ.0) GOTO 19
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTP
      CALL CHECK(PS,Q,X3,XM,XMP1,H)
      WRITE(6,991) XX(1),XX(XMP1)
  991 FORMAT(///,'THE POPULATION DENSITY FOR AGES',F6.0,' TO ',F6.0,//)
      WRITE(6,102) (R(I),I=1,XF)
   19 CONTINUE
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTP
      DO 13 I=1,XF
          P(I,J)=R(I)
   13     CONTINUE
   15 CONTINUE
      RETURN
      END

      SUBROUTINE ISPLIN(TN,T3,P,Q,X,TNP1,H,A)
C     ***************************************************************
C
C     THIS ROUTINE SOLVES THE INTEGRAL SPLINE PROBLEM FOR A POPULATION
C
C     ***************************************************************
      INTEGER T3,TNP1,TN
      REAL A(T3,9),P(TN),Q(T3),X(TNP1),H(TNP1)
```

```
            CALL BC(X,P,TNP1,TN,R10,R20,R1N,R2N)
            CALL CREATE(A,TN,T3,P,Q,X,TNP1,H,R10,R20,R1N,R2N)
            DO 33 I=1,T3
                 Q(I)=Q(I)/A(I,5)
            DO 34 J=6,9
                      A(I,J)=A(I,J)/A(I,5)
   34            CONTINUE
            A(I,5)=1
            DO 33 K=1,4

                 IF((I+K).GT.T3) GOTO 33
                 Q(I+K)=Q(I+K)-A(I+K,5-K)*Q(I)
                 DO 35 J=2,5
                           A(I+K,4-K+J)=A(I+K,4-K+J)-A(I+K,5-K)*A(I,4+J)
   35                 CONTINUE
                        A(I+K,5-K)=0
   33            CONTINUE
C           ***************************************************************************
C           BACKSUBSTITUTION
C           ***************************************************************************
            N=T3
            Q(N-1)=Q(N-1)-Q(N)*A(N-1,6)
            Q(N-2)=Q(N-2)-(Q(N)*A(N-2,7)+Q(N-1)*A(N-2,6))
            Q(N-3)=Q(N-3)-(Q(N)*A(N-3,8)+Q(N-1)*A(N-3,7)+Q(N-2)*A(N-3,6))
            DO 36 I=5,N
                 K=N-I+1
                 DO 36 J=1,4
                      Q(K)=Q(K)-A(K,5+J)*Q(K+J)
   36            CONTINUE
            RETURN
            END

            SUBROUTINE BC(X,P,TNP1,TN,R10,R20,R1N,R2N)
C           ***************************************************************************
C           THIS SUBROUTINE CALCULATES ESTIMATES OF THE DERIVATIVES OF
C           THE DENSITY FUNCTION IN TERMS OF THE HISTOGRAM VALUES
C           AT OR NEAR THE ENDPOINTS.
C           VALUES OBTAINED HERE ARE
C
C                R'(.5) = R10
C                R'(N-.5) = R1N
C                R''(N) = R2N
C                R''(0) =R20
C
C           THESE ARE ESTIMATED FROM  A NEWTON FINITE DIFFERENCE
C           POLYNOMIAL OF DEGREE THREE INTERPOLATING THE FIRST FOUR
C           HISTOGRAM VALUES AT THE MIDPOINTS OF THE SUBINTERVALS.
C
C           ***************************************************************************
            INTEGER TN,TNP1
            REAL P(TN),X(TNP1)
            IK=1
            A=(X(1)+X(2))/2.
            B=(X(2)+X(3))/2.
            C=(X(3)+X(4))/2.
            D=(X(4)+X(5))/2.
            E=P(1)/(X(2)-X(1))
            F=P(2)/(X(3)-X(2))
            G=P(3)/(X(4)-X(3))
            H=P(4)/(X(5)-X(4))
```

```
      34 CONTINUE
         H=(H-G)/(D-C)
         G=(G-F)/(C-B)
         F=(F-E)/(B-A)
         H=(H-G)/(D-B)
         G=(G-F)/(C-A)
         H=(H-G)/(D-A)
         IF (IK.EQ.0) GO TO 33
         R10=E+(A-B)*(G+H*(A-C))
         R20=2.*(G+H*(3.*X(1)-A-B-C))
         IK=0
         A=(X(IN)+X(INP1))/2.
         B=(X(IN-1)+X(IN  ))/2.
         C=(X(IN-2)+X(IN-1))/2.
         D=(X(IN-3)+X(IN-2))/2.
         E=P(IN  )/(X(IN  )-X(IN+1))
         F=P(IN-1)/(X(IN-1)-X(IN  ))
         G=P(IN-2)/(X(IN-2)-X(IN-1))
         H=P(IN-3)/(X(IN-3)-X(IN-2))
         GO TO 34
      33 CONTINUE
         R1N=E+(A-B)*(G+H*(A-C))
         R2N=2.*(G+H*(3.*X(INP1)-A-B-C))
         RETURN
         END

         SUBROUTINE CREATE(A,IN,I3,P,Q,X,INP1,H,R10,R20,R1N,R2N)
C        ***********************************************************************
C
C        THIS ROUTINE CREATES THE MATRIX FOR THE UNKNOWNS REQUIRED
C        TO APPROXIMATE A HISTOGRAM BY A DENSITY FUNCTION.
C
C        THE UNKNOWN VALUES  Q(I)  ARE THE VALUES
C             R(0),R2(0),R4(0),R(1),R2(1),...,R(INP1),R2(INP1),R4(INP1)
C
C        ***********************************************************************
         INTEGER IN,I3,INP1
         REAL A(I3,9),P(IN),Q(I3),X(INP1),H(INP1)
         REAL B(8,9),D(8),M(3,9),C(3,9)
C        ***********************************************************************
C        CAUTION:  DATA STATEMENTS ARE NON-EXECUTABLE
C                  ASSIGNMENT IS MADE ONLY AT COMPILE TIME
C        ***********************************************************************
         DATA D/24.,1.,1.,60.,24.,2.,24.,1./
         DATA B/0.   ,0.   ,0.   ,0.   ,24.  ,2.   ,-24.,0.   ,
     &0.   ,0.   ,0.   ,60.  ,-4.  ,0.   ,1.   ,0.   ,
     &0.   ,0.   ,0.   ,-5.  ,0.   ,0.   ,0.   ,0.   ,
     &0.   ,0.   ,0.   ,0.   ,-24.,-2.  ,24.  ,1.   ,
     &-24.,1.   ,1.   ,60.  ,-8.  ,-1.  ,-1.  ,0.   ,
     &1.   ,0.   ,0.   ,-5.  ,1.   ,0.   ,0.   ,0.   ,
     &0.   ,0.   ,0.   ,1.   ,24.  ,2.   ,0.   ,0.   ,
     &24.  ,0.   ,-1.  ,0.   ,-4.  ,-1.  ,0.   ,0.   ,
     &-1.  ,0.   ,0.   ,0.   ,1.   ,0.   ,0.   ,0./
*EXTENSION*    CC-1
         DO 31 I=2,INP1
         IA=I-1
              H(I)=X(I)-X(IA)
      31      CONTINUE
         H(1)=H(2)
         DO 35 J=1,9
```

```
                DO 32 I=1,3
                     A(I,J)=B(I,J)/D(I)
                     M(I,J)=1.
         32          CONTINUE
    C           PLACE CONSTANTS WHICH RECUR IN MATRIX IN THE FIRST
    C           THREE ROWS OF B
                DO 33 I=4,6
                     I1=I-4
                     IF (I.EQ.4) I1=3
                     C(I1,J)=B(I,J)/D(I)
         33          CONTINUE
                DO 34 I=7,8
                     A(I3-8+I,J)=B(I,J)/D(I)
         34          CONTINUE
         35     CONTINUE
    C           INDICES OF  X   AND  H  ARE INCREASED BY 1 IN PROGRAM
            A(1,5)=A(1,5)/H(2)
            A(1,8)=A(1,8)/H(2)
            A(1,6)=A(1,6)*H(2)
            A(1,9)=A(1,9)*H(2)
            A(I3-1,1)=A(I3-1,1)/H(INP1)
            A(I3-1,4)=A(I3-1,4)/H(INP1)
            A(I3-1,2)=A(I3-1,2)*H(INP1)
            A(I3-1,5)=A(I3-1,5)*H(INP1)
            A(4,2)=C(3,2)*H(2)
            A(4,5)=C(3,5)*H(2)
            A(4,3)=C(3,3)*(H(2)**3)
            A(4,6)=C(3,6)*(H(2)**3)
            A(4,7)=C(3,7)*(H(2)**5)
            DO 37 K=2,IN
                H1=H(K)
                H2=H(K+1)
                M(1,1)=1./H1
                M(1,2)=H1
                M(1,7)=1./H2
                M(1,8)=H2
                M(1,4)=M(1,1)+M(1,7)
                M(1,5)=M(1,2)+M(1,8)
                M(1,6)=H1**3
                M(1,9)=H2**3
                M(2,1)=M(1,1)
                M(2,4)=M(1,4)
                M(2,5)=M(1,2)
                M(2,7)=M(1,7)
                M(2,8)=M(1,8)
                M(3,2)=M(1,8)
                M(3,3)=M(1,9)
                M(3,5)=M(1,8)
                M(3,6)=M(1,9)
                M(3,7)=H2**5
                KT=3*(K-1)+1
                DO 36 I=1,3
                          DO 36 J=1,9
                               A(KT+I,J)=C(I,J)*M(I,J)
         36                    CONTINUE
                Q(KT)=2*P(K-1)
                Q(KT+1)=0.
                Q(KT+2)=0.
         37     CONTINUE
            Q(1)=R10
```

```
      Q(2)=R20
      Q(I3-2)=2.*P(IN)
      Q(I3-1)=R1N
      Q(I3)=R2N
      RETURN
      END

      SUBROUTINE RHO(Q,H,IN,INP1,I3,DELX,X,I2,R)
      INTEGER I3,IN,INP1,I2
      REAL Q(I3),H(INP1),X(INP1),R(I2),DELX
C     **********************************************************************
C
C     THIS SUBROUTINE USES THE RESULTS FROM THE INTEGRAL SPLINE
C     TO CALCULATE THE VALUES OF THE DENSITY FUNCTION AT INTERVALS
C     OF DELX.
C
C     PRINT CHECK IS MADE IF ICH = 1
      ICH = 1
      ICH = 0
C
C     **********************************************************************
      I2M1=I2-1
      IF (ICH.NE.1) GOTO 19
  101 FORMAT(10X,10F12.4)
      WRITE(6,112)
  112 FORMAT(' ','INDEPENDENT VARIABLES ARE   ')
      WRITE(6,101)(X(I),I=1,INP1)
      WRITE(6,113)
  113 FORMAT(' ','AGE INTERVALS USED IN ESTIMATION   ')
      WRITE(6,101) (H(I),I=1,INP1)
      WRITE(6,111)
  111 FORMAT(' ','SOLUTION OF EQUATIONS IS   ')
      WRITE(6,101)(Q(I),I=1,I3)
   19 CONTINUE
      I1=0
      DO 20 I=1,IN
      J=3*I
      IP1=I+1
      JP1=J+1
      JP2=J+2
      JP3=J+3
      JM1=J-1
      JM2=J-2
      X1=X(I)
   30 I1=I1+1
      R(I1)=Q(JM2)*(X(IP1)-X1)/H(IP1)+Q(JP1)*(X1-X(I))/H(IP1)+Q(JM1)*((
     1(X(IP1)-X1)**3)/(6.*H(IP1))-(X(IP1)-X1)*H(IP1)/6.)+Q(JP2)*
     2(((X1-X(I))**3)/(6.*H(IP1))-(X1-X(I))*H(IP1)/6.)+Q(JP3)*
     3(((X(IP1)-X1)**4)/24.-((X(IP1)-X1)**3)*H(IP1)/12.+(X(IP1)-X1)*
     4(H(IP1)**3)/24.)
      IF (I1.EQ.I2) GOTO 21
      X1=X1+DELX
      IF(I1.EQ.I2M1)GO TO 30
      XX=X(IP1)-X1
      IF(XX.GT.1E-3)GO TO 30
   20 CONTINUE
   21 IF (ICH.NE.1) GOTO 22
C     CHECK
      WRITE(6,104) R(I2),Q(I3-2)
  104 FORMAT(' RHO(END) =   ',F10.4,' AND SHOULD BE ',F10.4)
```

```
   22 CONTINUE
      RETURN
      END

      SUBROUTINE CHECK(P,Q,T3,TN,TNP1,H)
C     ******************************************************************************
C
C     THIS SUBROUTINE REGENERATES THE POPULATION HISTOGRAM FROM THE
C     DENSITY FUNCTION BY EXACT INTEGRATION OF THE FOURTH ORDER SPLINE.
C     HENCE THE ERRORS ARE THOSE SUFFERED AS A RESULT OF ROUND-OFF ERRO
C
C     ******************************************************************************
      INTEGER T3,TN
      INTEGER TNP1
      REAL Q(T3),P(TN),H(TNP1)
      DO 26 I=1,TN
      I1=I+1
      X1=H(I1)
      X3=X1*X1*X1
      X5=X3*X1*X1
      J=3*I
      JP1= J+1
      JP2= J+2
      JP3= J+3
      JM1= J-1
      JM2= J-2
      P(I)-(Q(JP1)+Q(JM2))*X1/2.-(Q(JP2)+Q(JM1))*X3/24.+Q(JP3)*X5/120.
   26       CONTINUE
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINT
      WRITE(6,103)
  103 FORMAT(//,' THE POPULATION REGENERATED FROM THE FOURTH ORDER SPLIN
     &E IS',//)
      WRITE(6,102) (P(I),I=1,TN)
  102 FORMAT(//,5F8.0)
C     PRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTPRINTP
      RETURN
      END
```

```
C       ****************************************************************
C
C       DATA FOR
C       PROGRAM 3.5 : REALMU
C
C       ****************************************************************
 &DNSITY PRTSDN=.FALSE.,PRTPDN=.FALSE.. &END
 &VALUES
PRTR=.FALSE.,PRTPX=.FALSE.,PRTPXT=.FALSE.,PRTXTR=.FALSE.,PRTMU=.TRUE.,
 &END
 &POP TYPE=.FALSE., &END
 &PRTLIM IPS=10,IPE=32, &END
 19 19 11
 62 20 11
 17. 20. 23. 26. 29. 32. 35. 38. 41. 44. 47. 50. 53. 56. 59. 62. 65. 69. 73. 78.
  0.  1.  2.  3.  4.  5.  6.  7.  8.  9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 20.
```

NUMBER OF AGE GROUPS = 19

NUMBER OF INCOME GROUPS = 19

NUMBER OF TIME VALUES = 11

BOUNDARIES OF AGE GROUPS

17.	20.	23.	26.	29.	32.	35.	38.	41.	44.
47.	50.	53.	56.	59.	62.	65.	69.	73.	78.

BOUNDARIES OF INCOME GROUPS

0.	1.	2.	3.	4.	5.	6.	7.	8.	9.
10.	11.	12.	13.	14.	15.	16.	17.	18.	20.

NUMBER OF AGE DENSITITES = 62

NUMBER OF INCOME DENSITIES = 20

NUMBER OF TIME VALUES PER ESTIMATION = 11

BOUNDARIES OF AGE DENSITIES

17.	18.	19.	20.	21.	22.	23.	24.	25.	26.
27.	28.	29.	30.	31.	32.	33.	34.	35.	36.
37.	38.	39.	40.	41.	42.	43.	44.	45.	46.
47.	48.	49.	50.	51.	52.	53.	54.	55.	56.
57.	58.	59.	60.	61.	62.	63.	64.	65.	66.
67.	68.	69.	70.	71.	72.	73.	74.	75.	76.
77.	78.								

BOUNDARIES OF INCOME DENSITIES

0.	1.	2.	3.	4.	5.	6.	7.	8.	9.
10.	11.	12.	13.	14.	15.	16.	17.	18.	19.

ECONOMIC MOBILITY AT AGE = 26

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.111277	0.116252	0.116913	0.114969	0.172819	0.000000
1	0.215519	0.241495	0.258235	0.266549	0.282623	0.296840
2	0.285226	0.315682	0.337683	0.352117	0.370427	0.385740
3	0.340868	0.379701	0.409919	0.431449	0.451815	0.467097
4	0.368720	0.412395	0.446589	0.472363	0.494464	0.508159
5	0.358901	0.409110	0.444860	0.467768	0.484032	0.495949
6	0.316581	0.367359	0.399156	0.416191	0.427648	0.437814
7	0.257054	0.294882	0.322176	0.337942	0.346783	0.350241
8	0.189184	0.214279	0.236610	0.251158	0.258041	0.257476
9	0.124612	0.142561	0.157866	0.167511	0.172383	0.172108
10	0.074906	0.085538	0.095700	0.101463	0.102951	0.100683
11	0.040464	0.046388	0.051898	0.054506	0.055095	0.053696
12	0.019897	0.022195	0.025415	0.026595	0.025420	0.022033
13	0.008325	0.008622	0.010109	0.010744	0.010492	0.008753
14	0.002328	0.000727	0.002234	0.004067	0.003724	-0.000338
15	-0.001835	-0.005261	-0.003484	0.000928	0.003688	0.001172
16	-0.006223	-0.013256	-0.008877	0.002261	0.007497	-0.001334
17	-0.012181	-0.023234	-0.016111	0.004155	0.015993	0.005285
18	-0.292241	0.000000	0.000000	0.000000	0.000000	0.000000
19	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

INCOME/TIME	1969	1970	1971	1972	1973
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.296315	0.266596	0.249676	0.256219	0.236537
2	0.391619	0.379343	0.367164	0.357292	0.323521
3	0.474394	0.465438	0.448026	0.422979	0.377151
4	0.511276	0.500847	0.482095	0.452644	0.402194
5	0.501293	0.491624	0.467458	0.432174	0.387747
6	0.442170	0.431563	0.407731	0.377301	0.345820
7	0.349004	0.339950	0.322598	0.299660	0.273339
8	0.255286	0.251548	0.240626	0.221254	0.197328
9	0.172792	0.174369	0.167027	0.147444	0.126693
10	0.103640	0.111473	0.106929	0.083024	0.061774
11	0.059019	0.071322	0.069681	0.043363	0.021838
12	0.027342	0.042824	0.042166	0.009930	-0.014835
13	0.014924	0.031157	0.032378	0.001892	-0.020895
14	0.003128	0.018473	0.018788	-0.016034	-0.041611
15	0.002676	0.014356	0.016365	-0.009176	-0.027724
16	-0.011083	-0.008625	-0.010541	-0.037676	-0.058305
17	-0.018399	-0.036829	-0.039338	-0.029043	-0.023248
18	0.000000	0.000000	0.000000	0.000000	0.000000
19	0.000000	0.000000	0.000000	0.000000	0.000000

ECONOMIC MOBILITY AT AGE = 27

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.116933	0.119594	0.121776	0.128833	0.225150	0.000000
1	0.224535	0.249514	0.266602	0.276619	0.292243	0.304564
2	0.295981	0.325959	0.348151	0.364419	0.384381	0.401668
3	0.353309	0.392260	0.423159	0.446717	0.468901	0.486391
4	0.381791	0.425870	0.461127	0.489367	0.513771	0.529162
5	0.371146	0.422335	0.459403	0.484655	0.502914	0.516149
6	0.326927	0.378850	0.411693	0.430866	0.444611	0.456386
7	0.265074	0.303647	0.331655	0.349244	0.360053	0.364707
8	0.194557	0.220308	0.243027	0.259163	0.267960	0.268972
9	0.127645	0.146222	0.161591	0.172656	0.179760	0.182093
10	0.076199	0.087424	0.097472	0.104274	0.107922	0.109241
11	0.040513	0.047071	0.052485	0.055985	0.058794	0.061692
12	0.018994	0.022238	0.025622	0.027434	0.028004	0.029003
13	0.006500	0.008442	0.010727	0.011871	0.012673	0.014738
14	-0.000770	0.000819	0.004255	0.006116	0.005145	0.003546
15	-0.006966	-0.004190	0.001665	0.005252	0.004227	0.001367
16	-0.014977	-0.009674	0.002773	0.010648	0.005233	-0.009055
17	-0.024510	-0.015504	0.004769	0.017902	0.008877	-0.015908
18	-0.250857	-0.640106	0.000000	0.000000	0.000000	0.000000
19	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

INCOME/TIME	1969	1970	1971	1972	1973
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.294514	0.269484	0.276482	0.291897	0.206930
2	0.404584	0.396084	0.395686	0.386580	0.320565
3	0.491499	0.482556	0.468817	0.439807	0.373368
4	0.529900	0.518199	0.501078	0.469435	0.407765
5	0.520073	0.509289	0.485531	0.448775	0.398806
6	0.459839	0.448171	0.425052	0.394693	0.359798
7	0.361789	0.349862	0.331199	0.307469	0.278970
8	0.264742	0.255954	0.241450	0.221790	0.200725
9	0.180872	0.175444	0.162670	0.145128	0.135115
10	0.110741	0.109197	0.096063	0.075488	0.074147
11	0.065914	0.066887	0.053915	0.031415	0.037622
12	0.033642	0.036557	0.022109	-0.006955	0.002342
13	0.020262	0.023966	0.011320	-0.015420	-0.001524
14	0.006756	0.010936	-0.001965	-0.033673	-0.021815
15	0.002984	0.007389	0.000190	-0.021525	-0.007207
16	-0.016399	-0.011885	-0.015284	-0.041264	-0.042357
17	-0.033523	-0.029936	-0.014534	-0.008132	-0.024949
18	0.000000	0.000000	0.000000	0.000000	0.000000
19	0.000000	0.000000	0.000000	0.000000	0.000000

ECONOMIC MOBILITY AT AGE = 28

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.123162	0.120658	0.113347	0.145383	0.296523	0.000000
1	0.232398	0.254934	0.267939	0.286312	0.299781	0.304790
2	0.304456	0.333317	0.353371	0.376342	0.396916	0.411267
3	0.362621	0.401377	0.431537	0.460418	0.483522	0.498886
4	0.391243	0.435589	0.471026	0.503740	0.529535	0.543373
5	0.379735	0.431821	0.469684	0.498424	0.517525	0.529387
6	0.334024	0.386994	0.420696	0.442693	0.457404	0.468215
7	0.270490	0.309781	0.338570	0.358412	0.369729	0.372895
8	0.198094	0.224542	0.247904	0.265985	0.275176	0.274755
9	0.129626	0.148898	0.164733	0.177622	0.185379	0.186688
10	0.077137	0.089067	0.099418	0.107679	0.111925	0.112570
11	0.040884	0.048204	0.053823	0.058518	0.061919	0.064351
12	0.019201	0.023465	0.026937	0.029355	0.030131	0.030883
13	0.006972	0.010403	0.012634	0.013578	0.013944	0.015778
14	0.000507	0.004507	0.007448	0.007400	0.004726	0.003320
15	-0.003881	0.002870	0.007292	0.005859	0.000820	-0.001308
16	-0.008173	0.004040	0.012747	0.009005	-0.004768	-0.015954
17	-0.012599	0.006393	0.019496	0.012118	-0.010809	-0.029221
18	-0.083334	0.064201	0.639828	0.000000	0.000000	0.000000
19	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

INCOME/TIME	1969	1970	1971	1972	1973
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.294296	0.301066	0.317688	0.317114	0.128684
2	0.413892	0.420158	0.420696	0.399845	0.301981
3	0.501981	0.499710	0.483168	0.445539	0.363597
4	0.541530	0.533245	0.514761	0.480266	0.411543
5	0.531288	0.522129	0.496949	0.458487	0.405000
6	0.469663	0.458262	0.434629	0.403688	0.366366
7	0.366954	0.353524	0.334340	0.310943	0.281648
8	0.266882	0.255236	0.240602	0.223761	0.204918
9	0.181479	0.172380	0.160649	0.149898	0.145216
10	0.109672	0.103062	0.091404	0.082074	0.090307
11	0.063762	0.058391	0.046924	0.039188	0.059132
12	0.030915	0.026887	0.014080	0.003126	0.030552
13	0.017178	0.014172	0.002974	-0.005658	0.026517
14	0.004224	0.003265	-0.007544	-0.020242	0.011088
15	0.001047	0.003189	-0.002633	-0.010866	0.017206
16	-0.014675	-0.005256	-0.006434	-0.024045	-0.015323
17	-0.026282	-0.006077	0.008105	-0.003227	-0.032300
18	0.000000	0.000000	0.000000	0.000000	0.000000
19	0.000000	0.000000	0.000000	0.000000	0.000000

ECONOMIC MOBILITY AT AGE = 29

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.133854	0.121995	0.133062	0.136656	0.398855	0.000000
1	0.241167	0.259080	0.280592	0.287376	0.303664	0.309516
2	0.311852	0.338466	0.364719	0.381678	0.404868	0.418926
3	0.369493	0.407406	0.442320	0.467680	0.492344	0.506474
4	0.397490	0.441740	0.480882	0.511886	0.538962	0.551501
5	0.385000	0.437692	0.478627	0.506406	0.525723	0.535925
6	0.338230	0.391908	0.428033	0.449619	0.464427	0.473716
7	0.273717	0.313400	0.344118	0.363675	0.374546	0.375631
8	0.200265	0.227075	0.251979	0.269831	0.278393	0.275783
9	0.131044	0.150617	0.167670	0.180448	0.187719	0.186934
10	0.078175	0.090355	0.101561	0.109536	0.113280	0.111934
11	0.041948	0.049395	0.055488	0.059752	0.062708	0.063209
12	0.020704	0.024944	0.028286	0.029825	0.030216	0.029576
13	0.009528	0.012553	0.013733	0.013170	0.013268	0.014390
14	0.005063	0.007898	0.008026	0.005222	0.002732	0.002521
15	0.004390	0.008330	0.006762	0.000502	-0.003441	-0.001342
16	0.006941	0.013074	0.009506	-0.003075	-0.013283	-0.012847
17	0.010265	0.018752	0.011848	-0.008686	-0.024427	-0.021582
18	0.047160	0.106648	0.091306	-0.279940	0.000000	0.000000
19	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

INCOME/TIME	1969	1970	1971	1972	1973
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.312977	0.319988	0.327746	0.259917	0.000398
2	0.426726	0.429771	0.425830	0.383844	0.285695
3	0.510980	0.504470	0.485086	0.437463	0.360252
4	0.549598	0.538212	0.519963	0.482372	0.417902
5	0.537169	0.525603	0.499961	0.459154	0.407753
6	0.473984	0.461053	0.437519	0.405390	0.369197
7	0.367741	0.353355	0.335409	0.313074	0.284891
8	0.265568	0.253089	0.240621	0.226467	0.207707
9	0.179280	0.169819	0.162003	0.156014	0.149460
10	0.106369	0.099466	0.093682	0.092381	0.096850
11	0.059644	0.053793	0.049405	0.052269	0.066470
12	0.026944	0.022516	0.018131	0.021288	0.042189
13	0.013616	0.009984	0.006234	0.010607	0.034248
14	0.002944	0.001630	-0.002029	-0.000369	0.023547
15	0.002857	0.003899	0.000670	0.000432	0.020039
16	-0.003762	0.004018	0.000266	-0.011132	-0.001204
17	-0.003380	0.012492	0.009072	-0.014645	-0.030834
18	0.000000	0.000000	0.000000	0.000000	0.000000
19	0.000000	0.000000	0.000000	0.000000	0.000000

ECONOMIC MOBILITY AT AGE = 30

INCOME/TIME	1963	1964	1965	1966	1967	1968
0	0.128801	0.113497	0.118311	0.184479	0.545564	0.000000
1	0.239379	0.256408	0.276280	0.300851	0.300961	0.317733
2	0.311375	0.337883	0.363744	0.392311	0.405036	0.423812
3	0.369589	0.407981	0.442971	0.475878	0.492807	0.508968
4	0.397628	0.442676	0.482204	0.518247	0.540368	0.553796
5	0.384903	0.438728	0.480291	0.511156	0.526670	0.536654
6	0.338030	0.392638	0.429405	0.452963	0.465317	0.474202
7	0.273537	0.313716	0.344967	0.365638	0.374430	0.374501
8	0.200043	0.227189	0.252436	0.270931	0.277687	0.273858
9	0.130972	0.150648	0.167871	0.181173	0.186979	0.185016
10	0.078379	0.090439	0.101529	0.109842	0.112372	0.110011
11	0.042592	0.049530	0.055193	0.059761	0.061786	0.061330
12	0.021929	0.025086	0.027465	0.029252	0.029308	0.028318
13	0.011620	0.012493	0.011981	0.011815	0.012449	0.013794
14	0.008437	0.007251	0.004399	0.002375	0.002296	0.003968
15	0.009773	0.006291	-0.000305	-0.004723	-0.003013	0.003102
16	0.015440	0.008079	-0.004237	-0.012740	-0.010446	-0.000687
17	0.021192	0.009624	-0.009630	-0.022864	-0.017514	0.000419
18	0.064790	0.028325	-0.074374	-0.265502	-1.308428	0.000000
19	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

INCOME/TIME	1969	1970	1971	1972	1973
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.326134	0.328001	0.298336	0.167918	-0.040440
2	0.431986	0.432153	0.418045	0.369135	0.305441
3	0.512818	0.505265	0.481545	0.433306	0.375141
4	0.551088	0.539733	0.520924	0.484840	0.430186
5	0.537087	0.525365	0.498780	0.457958	0.411097
6	0.473858	0.461200	0.437594	0.405577	0.372442
7	0.365963	0.352486	0.335910	0.315080	0.288484
8	0.262942	0.251350	0.240404	0.227104	0.206644
9	0.176925	0.168954	0.163549	0.158097	0.146534
10	0.104243	0.099487	0.097580	0.097448	0.093104
11	0.057501	0.054006	0.054148	0.058125	0.058949
12	0.025657	0.023791	0.025060	0.030962	0.035853
13	0.012790	0.010795	0.011352	0.017099	0.022578
14	0.004307	0.003496	0.003798	0.009017	0.017262
15	0.006319	0.004351	0.000934	0.002151	0.008709
16	0.006461	0.005187	-0.002328	-0.005700	0.007532
17	0.013637	0.010190	-0.006688	-0.021535	-0.010160
18	0.000000	0.000000	0.000000	0.000000	0.000000
19	0.000000	0.000000	0.000000	0.000000	0.000000

```
PARAMETERS USED FOR THIS RUN

RANGE OF AGES USED:      LX = 10   MX = 32

RANGE OF INCOMES USED:  LS =  3   MS = 12

NUMBER OF EIGEN VALUES USED:    =  5

RANGE OF YEARS 1963 TO 1973

NUMBER OF YEARS PER ESTIMATION:  TE = 11

NUMBER OF ESTIMATIONS:  LL =  1
```

```
3JOB POP; CHARGE BERGHOUT; CLASS 3;
USER 080002/VHJ;
PRINTLIMIT=13200;
3PROCESSTIME=300;
BEGIN
COMPILE POP FORTRAN;
FILE FILE6(KIND=BACKUP PETAPE,TITLE=PROG2);
3FILE FILE2(KIND=PACK,TITLE=DATA/XXMU5,FILETYPE=7);
FILE FILE3(KIND=PACK,TITLE=PLOT/DAGI,MAXRECSIZE=40,BLOCKSIZE=120,
     AREASIZE=35,AREAS=1,FLEXIBLE,SAVEFACTOR=10);
OPTION=AUTORM;
DATA



3DATA FILE5



3END JOB
```

EIGENPAIRS OF MTM(11,11)   BORDERING 0 ROWS AND COLUMNS

EIGENVALUE = 174.18530

EVT(I, 1)

0.240035	0.272162	0.297006	0.315574	0.327634
0.333026	0.331168	0.323798	0.310515	0.288468
0.260980				

EIGENVALUE = 0.25253

EVT(I, 2)

-0.121761	-0.037432	-0.064747	-0.009109	0.100907
0.088894	-0.069293	-0.272540	-0.265404	-0.138341
0.890369				

EIGENVALUE = 0.12880

EVT(I, 3)

-0.028302	-0.100302	-0.131120	0.115095	0.022306
-0.222591	-0.377783	-0.303316	0.267736	0.771837
0.080738				

EIGENVALUE = 0.08255

EVT(I, 4)

0.043671	-0.170104	-0.330951	-0.478325	-0.373591
-0.077233	0.210428	0.417059	0.414679	0.074875
0.299191				

EIGENVALUE = 0.02418

EVT(I, 5)

0.238003	0.195577	0.292003	0.324144	-0.277778
-0.515889	-0.311626	0.074049	0.325829	-0.362659
0.187077				

EIGENPAIRS OF MTM(11,11)   BORDERING 1 ROWS AND COLUMNS

EIGENVALUE = 152.29859

EVT(I, 1)

0.000000	0.291068	0.317654	0.337495	0.350361
0.356120	0.354174	0.346350	0.332131	0.308528
0.000000				

EIGENVALUE = 0.13034

EVT(I, 2)

0.000000	-0.109095	-0.144014	0.068675	-0.047038
-0.261289	-0.347265	-0.202755	0.351343	0.779110
0.000000				

EIGENVALUE = 0.09950

EVT(I, 3)

0.000000	-0.105072	-0.198592	-0.409257	-0.415026
-0.137163	0.248921	0.570667	0.449952	-0.029853
0.000000				

EIGENVALUE = 0.02957

EVT(I, 4)

0.000000	-0.223191	-0.427650	-0.418983	0.212913
0.525394	0.307218	-0.144583	-0.230826	0.319084
0.000000				

EIGENVALUE = 0.00787

EVT(I, 5)

0.000000	-0.293639	-0.335307	0.093449	0.529329
0.046257	-0.1[illegible]125	[illegible]073201	0.543508	-0.414535
0.000000				

EIGENPAIRS OF MTM(II,II) BORDERING 2 ROWS AND COLUMNS

EIGENVALUE = 124.90621

EVT(I, 1)

0.000000	0.000000	0.350776	0.372634	0.386865
0.393277	0.391162	0.382503	0.366676	0.000000
0.000000				

EIGENVALUE = 0.09829

EVT(I, 2)

0.000000	0.000000	-0.212471	-0.414913	-0.435153
-0.170657	0.214935	0.551467	0.462504	0.000000
0.000000				

EIGENVALUE = 0.05508

EVT(I, 3)

0.000000	0.000000	0.051634	0.377855	0.027689
-0.405794	-0.492499	-0.161582	0.655904	0.000000
0.000000				

EIGENVALUE = 0.01794

EVT(I, 4)

0.000000	0.000000	-0.607322	-0.277903	0.377629
0.399851	0.018439	-0.338461	0.369526	0.000000
0.000000				

EIGENVALUE = 0.00563

EVT(I, 5)

0.000000	0.000000	0.135736	0.174944	-0.555624
0.139569	0.488484	-0.581534	0.214416	0.000000
0.000000				

CHART 1

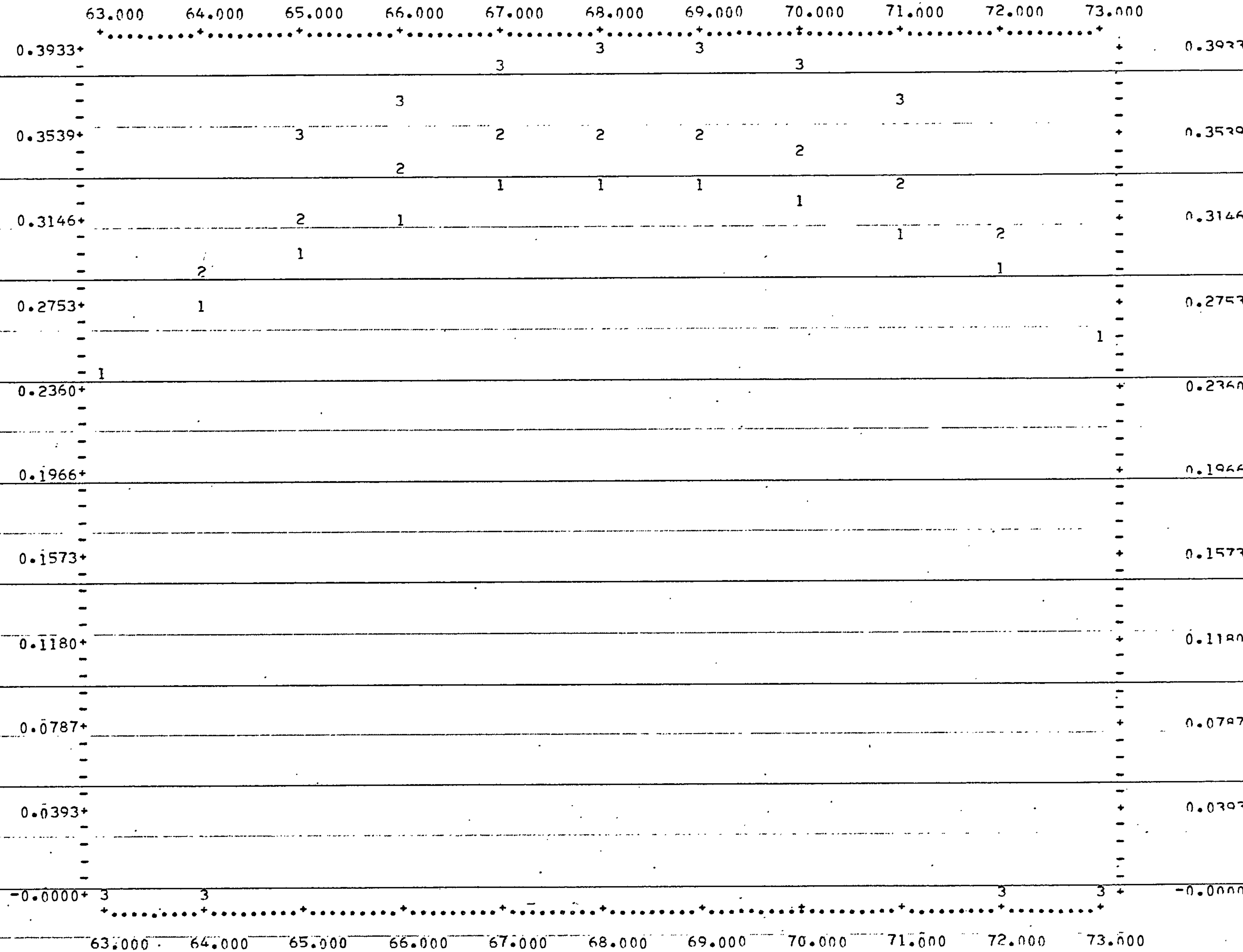

63.000 64.000 65.000 66.000 67.000 68.000 69.000 70.000 71.000 72.000 73.000
0.3933
0.3539
0.3146
0.2753
0.2360
0.1966
0.1573
0.1180
0.0787
0.0393
-0.0000

CHART 2

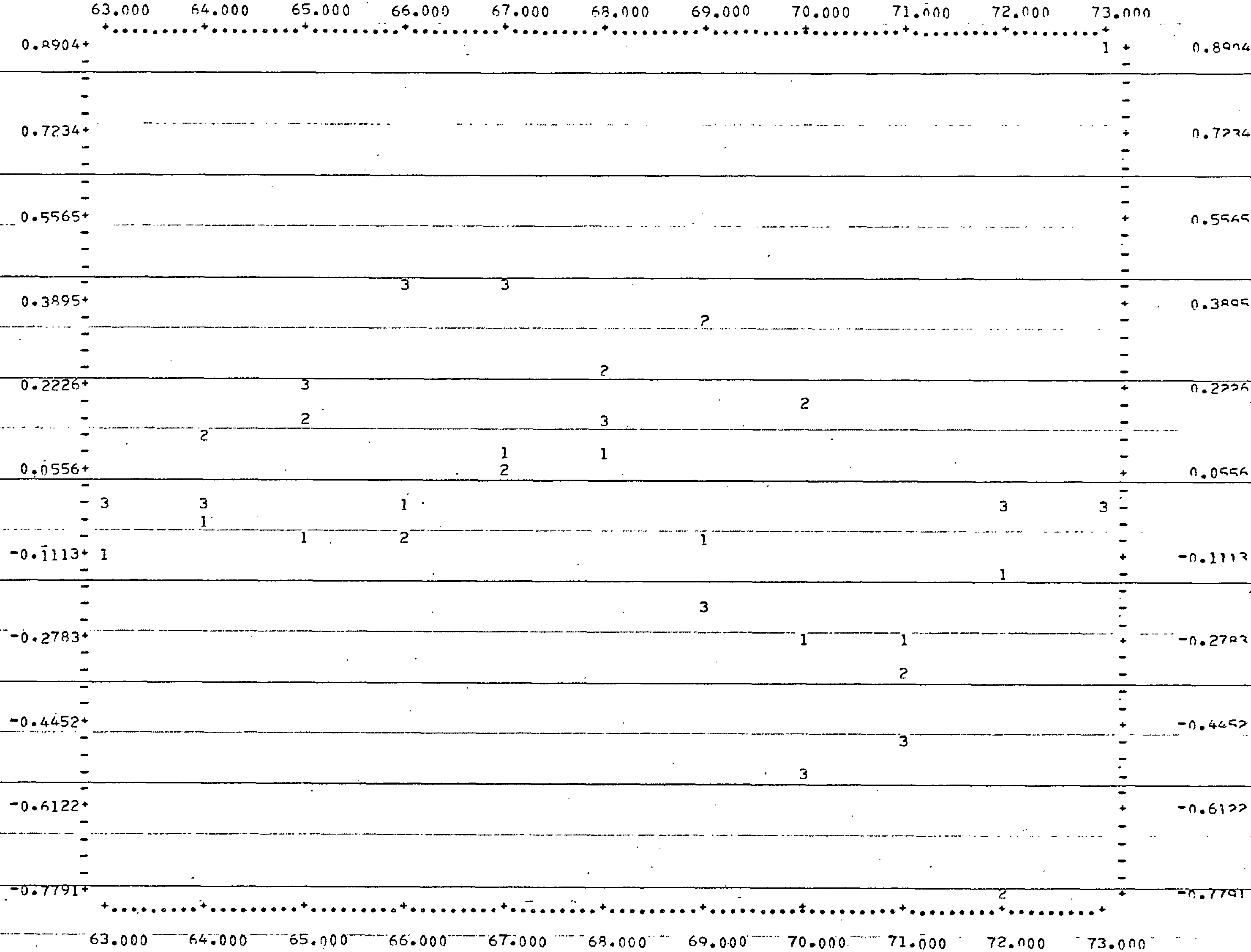
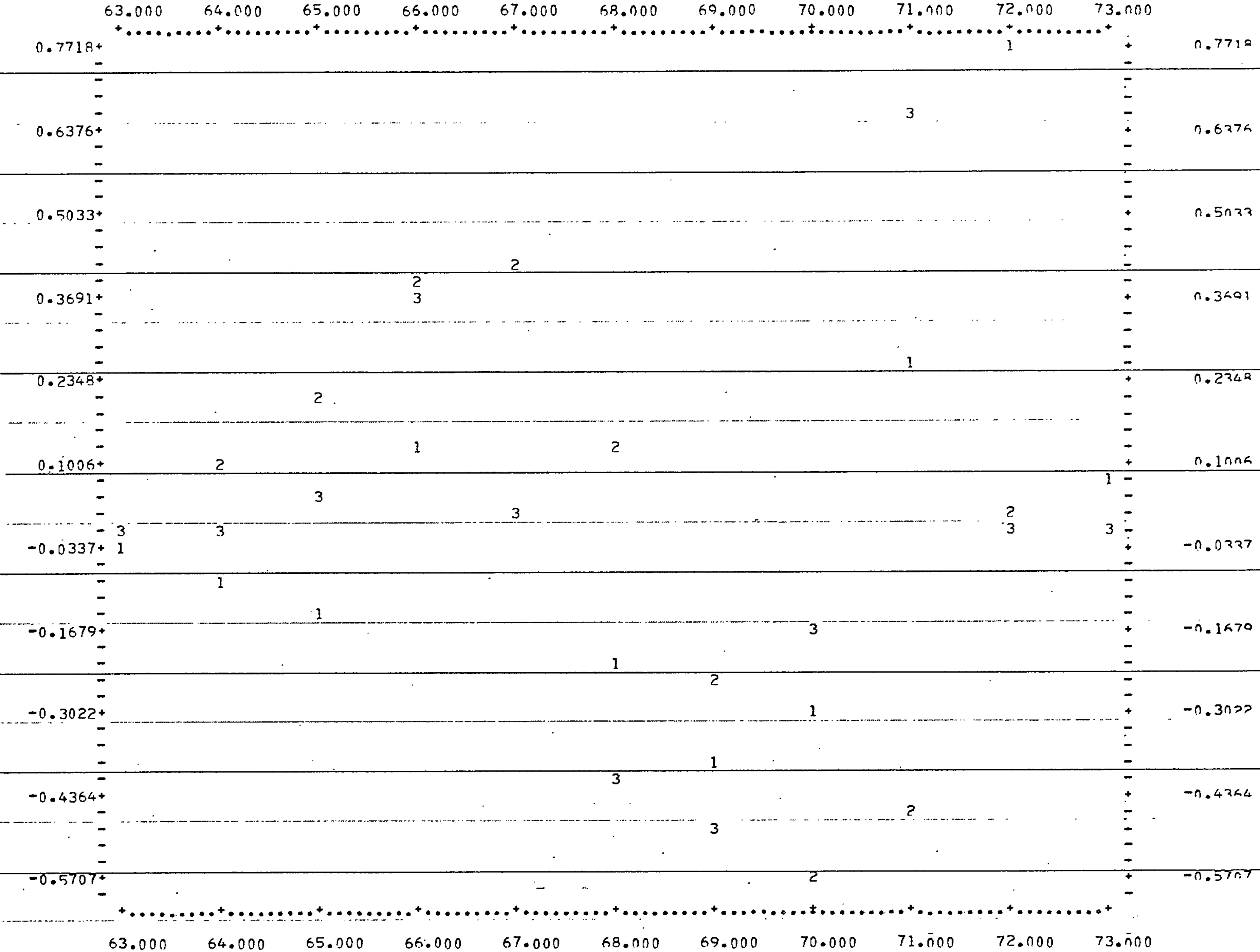
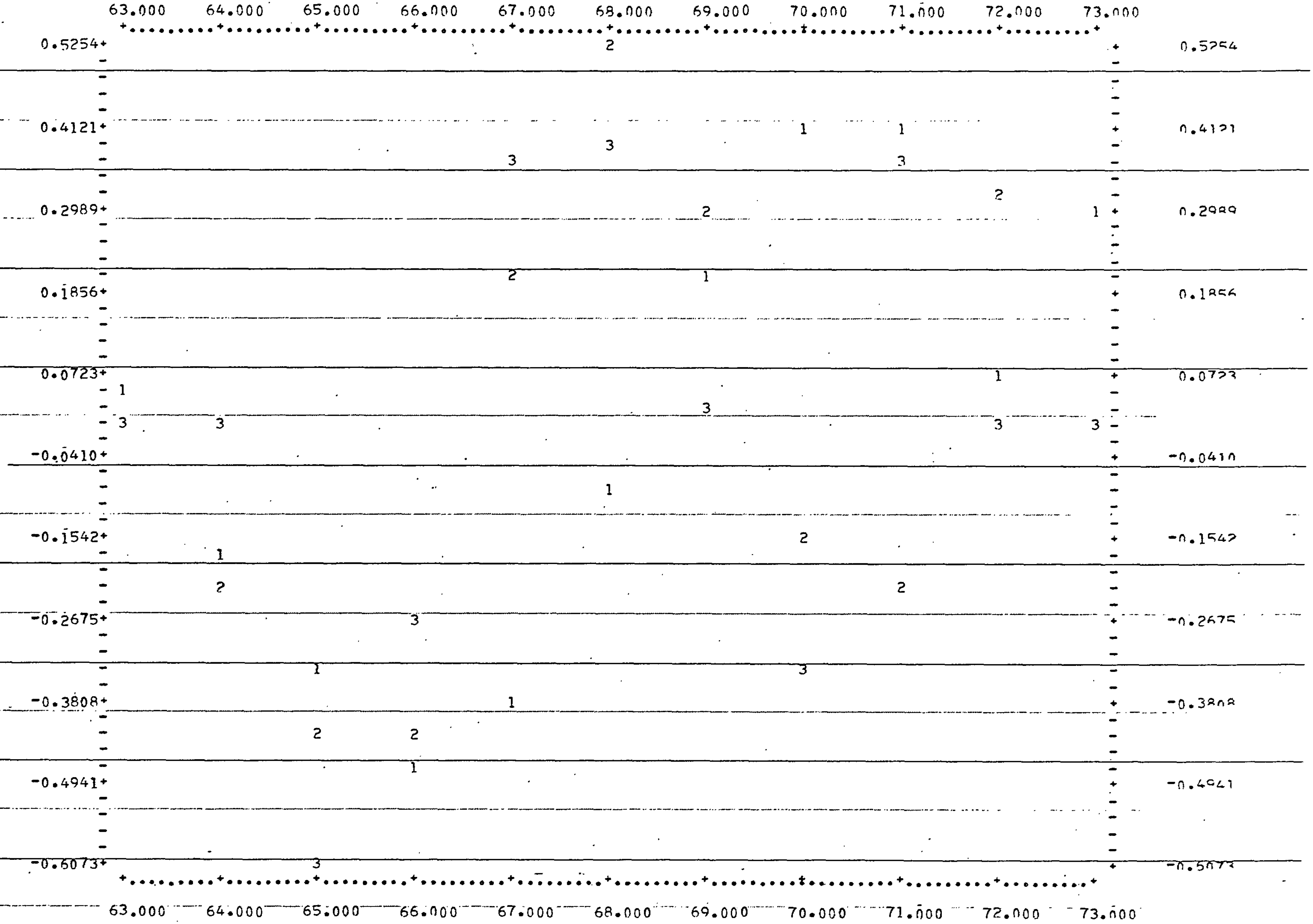

63.000
64.000
65.000
66.000
67.000
68.000
69.000
70.000
71.000
72.000
73.000
0.8904
0.7234
0.5565
0.3895
0.2226
0.0556
-0.1113
-0.2783
-0.4452
-0.6122
-0.7791

CHART 3



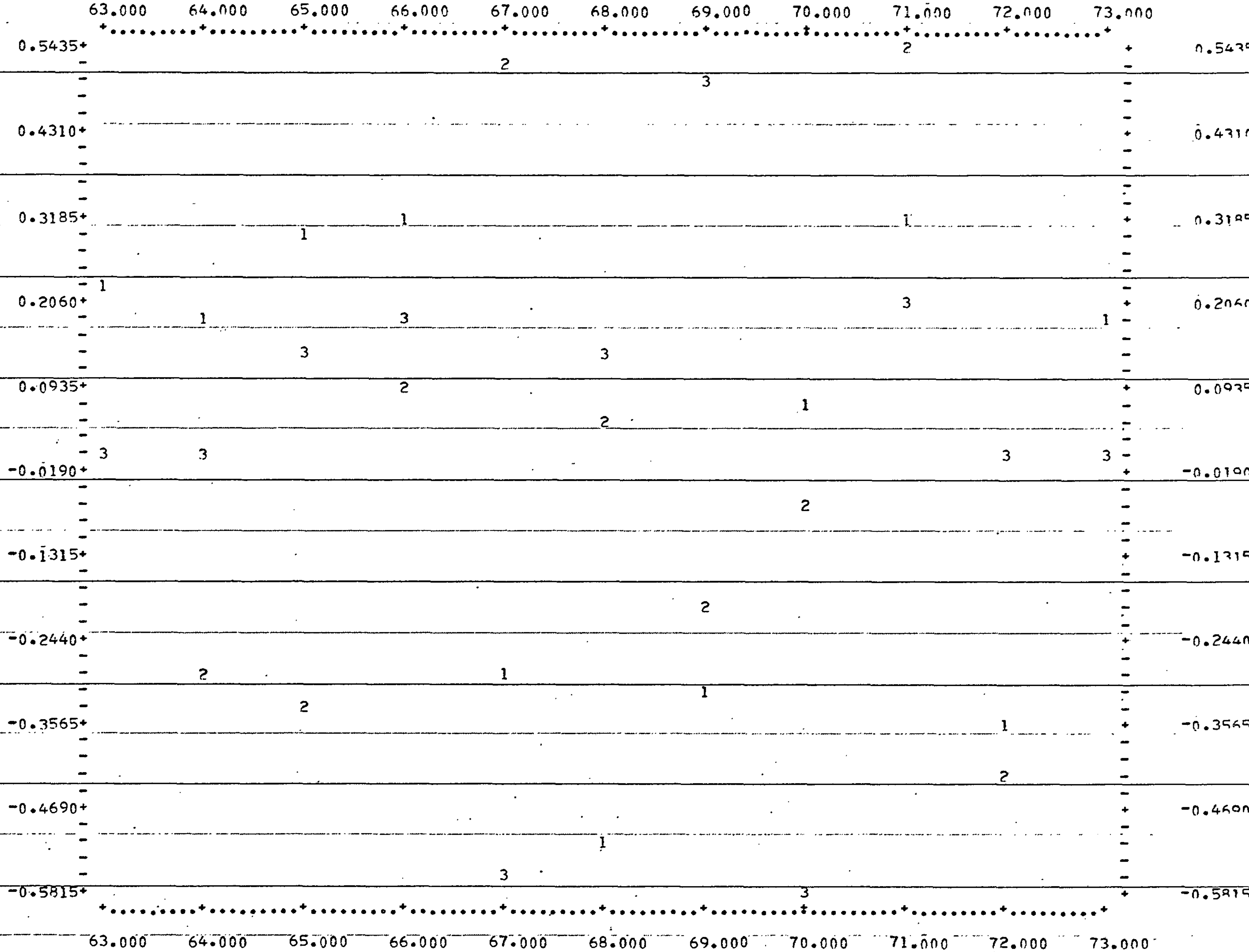
63.000
64.000
65.000
66.000
67.000
68.000
69.000
70.000
71.000
72.000
73.000
0.7718
0.6376
0.5033
0.3691
0.2348
0.1006
-0.0337
-0.1679
-0.3022
-0.4364
-0.5707

CHART 4



63.000 64.000 65.000 66.000 67.000 68.000 69.000 70.000 71.000 72.000 73.000
0.5254
0.4121
0.2989
0.1856
0.0723
-0.0410
-0.1542
-0.2675
-0.3808
-0.4941
-0.6073

CHART 5


63.000
64.000
65.000
66.000
67.000
68.000
69.000
70.000
71.000
72.000
73.000
0.5435
0.4310
0.3185
0.2060
0.0935
-0.0190
-0.1315
-0.2440
-0.3565
-0.4690
-0.5815

```
$ JOB    ACCT-NUM,VERNER,TIME=60
C        ******************************************************************
C
C        PROGRAM 3.6 : PLOT
C
C        ******************************************************************
```

```
3JOB PLOT; CHARGE BERGHOUT; CLASS 3;
USER 080002/VHJ;
PROCESSTIME=600;
BEGIN
COMPILE MY/DATA FORTRAN LIBRARY;
FORTRAN OPTION=AUTORM;
DATA
$SET $
$SET SEPARATE
FILE 4=FILE4
FILE 8=FILE8
FILE 9=FILE9
FILE 10=FILE10




3BIND MAIN WITH BINDER;
BINDER FILE HOST(TITLE=SURFACE);
FILE FILE4(KIND=PACK,TITLE=PLOT/DAGI,FILETYPE=7);
FILE FILE8(KIND=PACK,TITLE=PLOT/DAGH,FILETYPE=7);
FILE FILE9(KIND=PACK,TITLE=PLOT/DRELI,FILETYPE=7);
FILE FILE10(KIND=PACK,TITLE=PLOT/DENS,FILETYPE=7);
DATA
BIND DATA FROM MY/DATA;
3DATA FILE5



3END JOB
```

```
      SUBROUTINE DATA(X,IR,IC,IF)
C     ****************************************************************************
C
C     THIS ROUTINE READS ONE ROW OF DATA FROM THE FILE WITH
C     THE UNIT NUMBER SPECIFIED BY IF (IFORM IN SURFACE), EVERY TIME IT
C     IS CALLED.  THIS DATA IS USED TO PLOT ON THE CALCOMP PLOTTER.
C
C     ****************************************************************************
      REAL X(130)
      IF (IR.NE.1) GO TO 10
      READ(IF)(X(I),I=1,IC)
      READ(IF)(X(I),I=1,IC)
   10 READ(IF)(X(I),I=1,IC)
      IF (IR.NE.10) RETURN
      DO 20 J=1,8
   20 READ(IF) Y
      RETURN
      END
```

```
C       ****************************************************************************************
C
C       DATA FOR
C       PROGRAM 3.6 : PLOT
C
C       ****************************************************************************************
COMPONENTS OF MU ESTIMATED FROM DENSITY  COMPONENT 1
  10  40   2   2                                              1 1      3                 10
 30.0330.0  7.0        4.0                           20.0
COMPONENTS OF MU ESTIMATED FROM DENSITY  COMPONENT 2
  10  40   2   2                                              1 1      3                 10
 30.0330.0  7.0        4.0                           20.0
COMPONENTS OF MU ESTIMATED FROM DENSITY  COMPONENT 3
  10  40   2   2                                              1 1      3                 10
 30.0330.0  7.0        4.0                           20.0
COMPONENTS OF MU ESTIMATED FROM DENSITY  COMPONENT 4
  10  40   2   2                                              1 1      3                 10
 30.0330.0  7.0        4.0                           20.0
```

SYMVU RUN TITLE 'COMPONENTS OF MU ESTIMATED FROM DENSITY COMPONENT 1

ROWS = 10	COLS = 40
VIEW = 2	TYPE = 2
SMOOTH = 0	REPEAT = 0
LNINT = 0	NOZERO = 0
SORTT = 0	SAMEH = 0
ENDLIN = 0	SYMAP = 0
LEG = 0	NSCAL = 1
FLP = 0	BASE = 3
PLTDOT = 0	FDATA = 0
NAMIN = 0	NAMAX = 0
IFORM = 1	NLEG = 0
ALTITUDE = 30.000	AZIMUTH = 330.000
WIDTH = 7.000	HEIGHT = 4.000
AMIN = 0.000	AMAX = 0.000
VDIST = 20.000	STEP = 0.000
SEPSM = 0.000	SIZE = 0.000

FLIP MATRIX = 1

END OF PLOT

SYMVU RUN TITLE •COMPONENTS OF MU ESTIMATED FROM DENSITY COMPONENT 2

ROWS = 10	COLS = 40
VIEW = 2	TYPE = 2
SMOOTH = 0	REPEAT = 0
LNINT = 0	NOZERO = 0
SORTT = 0	SAMEH = 0
ENDLIN = 0	SYMAP = 0
LEG = 0	NSCAL = 1
FLP = 0	BASF = 3
PLTDOT = 0	FDATA = 0
NAMIN = 0	NAMAX = 0
IFORM = 1	NLEG = 0
ALTITUDE = 30.000	AZIMUTH = 330.000
WIDTH = 7.000	HEIGHT = 4.000
AMIN = 0.000	AMAX = 0.000
VDIST = 20.000	STEP = 0.000
SEPSM = 0.000	SIZE = 0.000

FLIP MATRIX = 1

END OF PLOT

SYMVU RUN TITLE 'COMPONENTS OF MU ESTIMATED FROM DENSITY COMPONENT 3

ROWS = 10	COLS = 40
VIEW = 2	TYPE = 2
SMOOTH = 0	REPEAT = 0
LNINT = 0	NOZERO = 0
SQRTT = 0	SAMEH = 0
ENDLIN = 0	SYMAP = 0
LEG = 0	NSCAL = 1
FLP = 0	BASE = 3
PLTDOT = 0	FDATA = 0
NAMIN = 0	NAMAX = 0
IFORM = 1	NLEG = 0
ALTITUDE = 30.000	AZIMUTH = 330.000
WIDTH = 7.000	HEIGHT = 4.000
AMIN = 0.000	AMAX = 0.000
VDIST = 20.000	STEP = 0.000
SEPSM = 0.000	SIZE = 0.000

FLIP MATRIX = 1

END OF PLOT

```
$ JOB   ACCT-NUM,VERNER,TIME=60
C       ****************************************************************************************
C
C       SUBROUTINE 3.7 : PLOTT
C
C       ****************************************************************************************
```

```
C         ****************************************************************
          SUBROUTINE PLOTT(NO,A,N,M,IFUNC,EQWQWQ,XMX,XMN,YMX,YMN)
C
C         ****************************************************************
C
          DIMENSION ISYM(10)
          DATA ISYM(1),ISYM(2),ISYM(3),ISYM(4),ISYM(5),ISYM(6),ISYM(7),ISYM
         1(8),ISYM(9),ISYM(10)/'1      ','2      ','3      ','4      ','5      '
         1,'6      ','7      ','8      ','9      ','10     '/
          DIMENSION IOUT(101), XPR(11), A(N,M)
C
        1 FORMAT(1H1,60X,7H CHART ,I3)
        2 FORMAT(1H ,12X,2H- ,101A1,' -')
        3 FORMAT(1H ,F12.3,'+ ',101A1,' +',F12.3)
       33 FORMAT(' ',F12.4,'+ ',101A1,' +',F12.4)
        4 FORMAT(1H ,14X,10('+.........'),1H+)
        5 FORMAT(1H0,9X,11E10.3)
       55 FORMAT('0',9X,11F10.3)
        6 FORMAT(2X,'FUNCTION VALUES NOT PLOTTED.  N, THE NUMBER OF VALUES'
         1' TO BE PLOTTED MUST BE GREATER THAN 1.')
C
C                   SKIP ALTERNATE ENTRY POINT.
C
          IFLAG=0
          GO TO 1000
C
          ENTRY PLOTB(NO,A,N,M,XMX,XMN,YMX,YMN)
          IFLAG=1
          GO TO 2000
C
          ENTRY PLOTA(NO,A,N,M,XMX,XMN,YMX,YMN)
          IFLAG=0
     2000 IFUNC=-1
C
     1000 WRITE(6,1) NO
C
C                   STORE CALLING ARGUMENTS SO THEY WILL NOT BE ALTERED UPON
C                   RETURN TO CALLING PROGRAM.
C
          XMAX-XMX
          XMIN-XMN
          YMAX-YMX
          YMIN-YMN
C
C                   INITIALIZE CONSTANTS
C
          R50=50.0
          R100-100.0
          IBLK=' '
C
C                   IF MAX AND MIN VALUES FOR INDEPENDENT VARIABLE HAVE BEEN
C                   GIVEN - SKIP SCAN WHICH DETERMINES THEM.
C

          IF (XMAX.GT.XMIN) GO TO 60
       10 XMIN = A(1,1)
          XMAX = XMIN
          DO 50  J=2,N
          IF (A(J,1).GT.XMAX) GO TO 30
```

```
   20 IF (A(J,1).GE.XMIN) GO TO 50
      GO TO 40
   30 XMAX = A(J,1)
      GO TO 50
   40 XMIN = A(J,1)
   50 CONTINUE
   60 IF (IFUNC.LE.0) GO TO 110
   70 IF (M.GT.1) GO TO 90
   80 WRITE(6,6)
      IF (M.GT.2) GO TO 110
      GO TO 340
   90 DO 100  J=1,N
C
C                IF ONLY FUNCTION CURVE BEING PLOTTED COMPUTE INDEPENDENT
C                VARIABLE VALUES TO BE USED.
C
      IF (M - 2) 340,95,100
   95 A(J,1) = ((XMAX - XMIN)/(N - 1))*(J - 1) + XMIN
C
C     ******************************************************************
  100 CALL FOWOWO(A(J,1),A(J,M))
C     ******************************************************************
C
C                CALCULATE SCALING FOR INDEPENDENT VARIABLE (IE ARGUMENT
C                WIDTH PER PRINT POSITION WIDTH).
C
  110 XSCAL = (XMAX - XMIN)/R100
C
C                IF MAX AND MIN VALUES FOR DEPENDENT VARIABLES HAVE BEEN
C                GIVEN - SKIP SCAN WHICH DETERMINES THEM.
C
      IF (YMAX.GT.YMIN) GO TO 170
  120 YMIN = A(1,2)
      YMAX = YMIN
      DO 160  J=2,M
      DO 160  L=1,N
      IF (A(L,J).GT.YMAX) GO TO 140
  130 IF (A(L,J).GE.YMIN) GO TO 160
  150 YMIN = A(L,J)
      GO TO 160
  140 YMAX = A(L,J)
  160 CONTINUE
C
C                CALCULATE SCALING FOR DEPENDENT VARIABLES (IE FUNCTION
C                WIDTH PER PRINT POSITION HEIGHT)
C
  170 YSCAL = (YMAX - YMIN)/R50
C
C                SET UP AND PRINT INDEPENDENT VARIABLE AXIS FORM
C
      XPR(1) = XMIN
      DO 190  J=2,11
      XPR(J) = XPR(J-1) + XSCAL*10.
      IF (ABS(XPR(J)).GT..5*XSCAL) GO TO 190
  180 XPR(J)=0.0
  190 CONTINUE
      IF (IFLAG-1) 185,175,185
  175 WRITE (6,55) (XPR(J),J=1,11)
      GO TO 195
  185 WRITE (6,5) (XPR(J),J=1,11)
```

```
      195 WRITE (6,4)
C
C                 INITIALIZE DEPENDENT VARIABLE AXIS VALUE
C
          YPR = YMAX
C
C                 INITIALIZE MOD 5 COUNTER FOR DEPENDENT VARIABLE AXIS.
C
          ICOUNT = 4
      200 DO 210  J=1,101
      210 IOUT(J) = IBLK
C
C                 SCAN ARRAY FOR DEPENDENT VARIABLES WITH VALUES IN RANGE OF
C                 PRESENT INDEPENDENT VARIABLE AXIS VALUE
C
          DO 260  J=2,M
          DO 260  L=1,N
          IF (ABS(YPR - A(L,J)) - .5*YSCAL) 230,220,260
C
C                 IF VALUE LIES BETWEEN 2 PRINT LINES PUT IT WITH THE HIGHER
C                 VALUED LINE.
C
      220 IF (YPR.LT.A(L,J)) GO TO 260
C
C                 CALCULATE CORRESPONDING PRINT POSITON FOR INDEPENDENT
C                 VARIABLE AXIS.
C
      230 JP = (A(L,1) - XMIN)/XSCAL + 1.5
C
C                 INSERT DEPENDENT VARIABLE'S SUBSCRIPT IN APPROPRIATE
C                 POSITION IN PRINT LINE ONLY IF IN LEGAL RANGE OF
C                 INDEPENDENT VARIABLE AXIS.
C
          IF (JP.LT.1.OR.JP.GT.101) GO TO 260
      250 IF (IFUNC.LE.0 .OR. L.NE.M) GO TO 255
C
C                 PUT AN ASTERISK IN OUTPUT LINE (VARIABLE WAS INPUT THROUGH
C                 FUNCTION)
C
          IOUT(JP)='*'

          GO TO 260
C
C                 PUT APPROPRIATE SUBSCRIPT IN OUTPUT LINE.
C
      255 IOUT(JP)=ISYM(J-1)
      260 CONTINUE
C
C                 ON EVERY 5TH LINE PRINT DEPENDENT VARIABLE VALUE.
C
          IF (ICOUNT.EQ.4) GO TO 300
      270 ICOUNT = ICOUNT + 1
          WRITE(6,2) (IOUT(JP), JP=1,101)
C
C                 CHECK IF FINISHED
C
      280 IF (YPR.LE.YMIN) GO TO 310
      290 YPR = YPR - YSCAL
          GO TO 200
      300 IF (IFLAG-1) 305,295,305
```

```
  295 WRITE (6,33) YPR,(IOUT(JP),JP=1,101),YPR
      GO TO 315
  305 WRITE (6,3) YPR,(IOUT(JP),JP=1,101),YPR
  315 ICOUNT=0
C
C                SET UP AND PRINT INDEPENDENT VARIABLE AXIS FORM.
C
      GO TO 280
  310 WRITE(6,4)
      IF (IFLAG-1) 325,335,325
  325 WRITE (6,5) (XPR(J),J=1,11)
      GO TO 340
  335 WRITE (6,55) (XPR(J),J=1,11)
  340 RETURN
      END
```

4. ESTIMATION OF DYNAMICS OF FERTILITY

4.0 Introduction.

This chapter contains source listings of programs for interpolating the fertility parameters, and for listing and in certain cases smoothing economic indicators. Also included is a program for the determination of the dynamics of the vectors $\underline{b}$ and $\underline{a}_1$ of Chapter 2 using the algorithm based on estimation of linear models from input-output data described in Chapter IV above.

4.1 Extended Fertility Parameters

The averages of vectors a_r , $r = 1,2,3,4$ obtained in Program 2. and $\underline{b}$ of Table 2. represent estimated annual values of the functions $a_r(t)$, $b(t)$. To obtain quarterly values, the interpolatory cubic spline routine was used. The corresponding subroutines were used in the following program to produce extended vectors ae and be . Plots of these vectors are obtained through the next program.

```
$JOB   ACCT-NUM,VERNER,TIME=60
C      ************************************************************************
C
C               PROGRAM 4.1 :   ABEXT
C
C      ************************************************************************
```

```
3JOB ABEXT; CHARGE BERGHOUT; CLASS 3;
USER 080002/VHJ;
BEGIN
3COMPILE ABEXT FORTRAN;
FILE FILE4(KIND=PACK,TITLE=DATA/XXAV,FILETYPE=7);
DATA




3DATA FILE5




3END JOB
```

```
C       ************************************************************************
C
C       PROGRAM : 4.1
C       TITLE : ABEXT
C       PURPOSE : EXTEND A(T) AND B(T) BY SPLINES
C       DATA : DATA/XXAV,B(T)
C       OUTPUT : A(T) AND B(T) EXTENDED
C
C       ************************************************************************
        REAL K(2,13),A(3,13),H(13),COF(4,13),D(13)
        REAL AT(4,13),ATE(4,49)
        INTEGER NDOM,YRS
        NDOM = 4
        YRS = 13
C
C       READ AT:  THE AVERAGES OF EIGENVECTORS
C
        DO 10 I = 1,NDOM
        READ(4)(AT(I,J),J=1,YRS)
     10 CONTINUE
        DO 20 I = 1,YRS
           K(1,I) = I - 1
     20    CONTINUE
        CALL SETUP(YRS,A,H,K)
        DO 30 I = 1,NDOM
           DO 40 J = 1,YRS
           K(2,J) = AT(I,J)
     40    CONTINUE
           CALL SOLVE(A,K,H,YRS,D,COF)
           T = -.25
           DO 50 J = 1,YRS
           DO 50 L = 1,4
           T = T + .25
           ATE(I,4*(J-1)+L)=COF(1,J)*T**3 + COF(2,J)*T**2 + COF(3,J)*T
     $     + COF(4,J)
           IF (J.EQ.13) GO TO 30
     50    CONTINUE
     30 CONTINUE
C
C       PRINT AT AND ATE
C
        DO 60 I = 1,NDOM
        WRITE(6,900) I
        WRITE(6,901)
        DO 70 J = 1,YRS
           IJ = 4*(J-1)+1
           NJ = 4*(J-1)+4
           IF (J.EQ.13) GO TO 80
           WRITE(6,902) AT(I,J),(ATE(I,L),L=IJ,NJ)
           GO TO 70
     80    WRITE(6,902) AT(I,J),ATE(I,IJ)
     70    CONTINUE
     60 CONTINUE
    900 FORMAT('1',5X,'EIGEN VECTOR = ',I2,/)
    901 FORMAT('0',11X,'AT',35X,'ATE',//)
    902 FORMAT(/,5X,F13.8,10X,4F13.8)
    903 FORMAT(5F16.8)
        CALL BEXTE
        STOP
```

```
            END
**WARNING**    FM-7      903

            SUBROUTINE BEXTE
            REAL B(13),BEXT(49)
            REAL K(2,13),A(3,13),H(13),COF(4,13),D(13)
            INTEGER YRS
            YRS = 13
      C
      C     READ B FROM CARDS
      C
            READ(5,900)(B(I),I=1,YRS)
            DO 20 I=1,YRS
               K(1,I) = I-1
         20    CONTINUE
            CALL SETUP(YRS,A,H,K)
            DO 30 J = 1,YRS
               K(2,J) = B(J)
         30    CONTINUE
            CALL SOLVE(A,K,H,YRS,D,COF)
            T = -.25
            DO 40 J = 1,YRS
            DO 50 L = 1,4
               T = T + .25
               BEXT(4*(J-1)+L) = COF(1,J)*T**3 + COF(2,J)*T**2 + COF(3,J)*T
           $   + COF(4,J)
               IF (J.EQ.13) GO TO 40
         50    CONTINUE
         40 CONTINUE
      C
      C     PRINT B AND B EXTENDED
      C
            WRITE(6,901)
            DO 60 J = 1,YRS
               IJ = 4*(J-1)+1
               NJ = 4*(J-1)+4
               IF (J.EQ.13) GO TO 70
               WRITE(6,902) B(J),(BEXT(L),L=IJ,NJ)
               GO TO 60
         70    WRITE(6,902) B(J),BEXT(IJ)
         60 CONTINUE
        900 FORMAT(8F10.6)
        901 FORMAT('0',11X,'B',35X,'BEXT',//)
        902 FORMAT(/,5X,F13.8,10X,4F13.8)
            RETURN
            END

            SUBROUTINE SETUP(N,A,H,K)
      C
      C     ****************************************************************
      C     THIS ROUTINE SETS UP A TRIDIAGONAL MATRIX OF THE SPLINE EQUATION
      C     IN A  3 BY N  ARRAY   -   FIRST ROW IS SUPERDIAGONAL
      C                               SECOND ROW IS DIAGONAL
      C                               THIRD ROW IS SUBDIAGONAL
      C     AND THEN DECOMPOSES A TO  LU  SO THAT
      C        FIRST ROW IS SUPERDIAGONAL OF U, SECOND ROW IS DIAGONAL OF U
      C       THIRD ROW IS SUBDIAGONAL OF L, DIAGONAL OF L IS UNITY
      C     ****************************************************************
      C
            REAL A(3,N),H(N),K(2,N)
```

```
      NM1=N-1
      H(2)=K(1,2)-K(1,1)
      DO 10 I=2,NM1
         H(I+1)=K(1,I+1)-K(1,I)
         A(1,I)=H(I+1)/(H(I+1)+H(I))
         A(2,I)=2.0
         A(3,I)=1-A(1,I)
 10      CONTINUE
      A(1,1) = A(1,2)
      A(2,1) = A(2,2)
      A(3,1) = A(3,2)
      A(1,N) = A(1,N-1)
      A(2,N) = A(2,N-1)
      A(3,N) = A(3,N-1)
      A(1,2) = -2.
      A(2,2) = 2.
      A(3,2) = 0.
      A(1,N-1) = 0.
      A(2,N-1) = 2.
      A(3,N-1) = -2.
      NM1 = N-1
      DO 11 I=3,NM1
         A(3,I)=A(3,I)/A(2,I-1)
         A(2,I)=A(2,I)-A(3,I)*A(1,I-1)
 11      CONTINUE
      RETURN
      END
C
      SUBROUTINE SOLVE(A,K,H,N,D,COF)
C
C     ******************************************************************
C     THIS ROUTINE CALCULATES THE SECOND ORDER FINITE DIFFERENCES OF
C     THE SPLINE, AND THEN SOLVES  AM=D,  (BY FORWARD AND BACKWARD
C     SUBSTITUTION), PLACING M(VECTOR OF SECOND DERIVATIVES) IN D
C     END POINT CONDITIONS NOW USE THIRD ORDER FINITE DIFFERENCES
C     TO ESTIMATE THE THIRD ORDER DERIVATIVES AT X0+3H/2 AND XN-3H/2.
C     ******************************************************************
C
      REAL H(N),D(N),A(3,N),K(2,N),COF(4,N)
      D(2)=(K(2,2)-K(2,1))/H(2)
      NM1=N-1
      DO 12 I=2,NM1
         D(I+1)=(K(2,I+1)-K(2,I))/H(I+1)
         D(I)=6*(D(I+1)-D(I))/(H(I+1)+H(I))
 12      CONTINUE
      CA=-6./(H(2)*(H(2)+H(3))*(H(2)+H(3)+H(4)))
      CB=6./(H(2)*H(3)*(H(4)+H(3)))
      CC=-6./(H(3)*H(4)*(H(2)+H(3)))
      CD=6./(H(4)*(H(4)+H(3))*(H(2)+H(3)+H(4)))
      D(1)=-2.*H(3)*(CA*K(2,1)+CB*K(2,2)+CC*K(2,3)+CD*K(2,4))
      CA=-6./(H(N-2)*(H(N-2)+H(NM1))*(H(N-2)+H(NM1)+H(N)))
      CB=6./(H(N-2)*H(NM1)*(H(N)+H(NM1)))
      CC=-6./(H(NM1)*H(N)*(H(N-2)+H(NM1)))
      CD=6./(H(N)*(H(N)+H(NM1))*(H(N-2)+H(NM1)+H(N)))
      D(N)=2.*H(NM1)*(CA*K(2,N-3)+CB*K(2,N-2)+CC*K(2,NM1)+CD*K(2,N))
      T1 = D(2)
      T2 = D(NM1)
      D(2) = D(1)
      D(NM1) = D(N)
```

```
      DO 13 I=3,NM1
          D(I)=D(I)-A(3,I)*D(I-1)
13        CONTINUE
      D(NM1) = D(NM1)/A(2,NM1)
      DO 14 I=3,NM1
          J=N+1-I
          D(J)=(D(J)-A(1,J)*D(J+1))/A(2,J)
14        CONTINUE
      D(1) = (T1 - A(2,1)*D(2) - A(1,1)*D(3))/A(3,1)
      D(N) = (T2 - A(2,N)*D(NM1) - A(3,N)*D(N-2))/A(1,N)
      CALL POLLY(N,D,K,H,COF)
      RETURN
      END
C

      SUBROUTINE POLLY(N,M,K,H,COF)
C
C     ******************************************************************
C     THIS ROUTINE COMPUTES THE COEFFICIENTS OF THE SPLINE POLYNOMIAL
C       ON EACH SUBINTERVAL
C     K IS THE ARRAY OF DATA POINTS
C     H IS THE VECTOR OF SUBINTERVAL LENGTHS
C     M IS THE SOLUTION VECTOR TO THE EQUATION AM=D
C     ******************************************************************
C
      REAL M(N),K(2,N),H(N),COF(4,N)
      NM1=N-1
      DO 11 I=1,NM1
      COF(1,I)=(M(I+1)-M(I))/(6.*H(I+1))
      COF(2,I)=(K(1,I+1)*M(I)-K(1,I)*M(I+1))/(2.*H(I+1))
      DD=M(I+1)*K(1,I)*K(1,I)-M(I)*K(1,I+1)*K(1,I+1)
     &+2.*K(2,I+1)-2.*K(2,I)
      COF(3,I)=(DD/(2.*H(I+1)))+H(I+1)*(M(I)-M(I+1))/6.
      DD=M(I)*(K(1,I+1)**3)-M(I+1)*(K(1,I)**3)+6.*K(1,I+1)*K(2,I)
     &-6.*K(1,I)*K(2,I+1)+K(1,I)*M(I+1)*(H(I+1)**2)-K(1,I+1)*M(I)
     &*(H(I+1)**2)
      COF(4,I)=DD/(H(I+1)*6.)
11    CONTINUE
      DO 16 J=1,4
16    COF(J,N)=COF(J,NM1)
      RETURN
        END
```

```
*****************************************************************

DATA FOR
     PROGRAM 4.1 :  ABEXT

*****************************************************************
-0.015191  0.009268  0.013600  0.020814  0.023007  0.045760  0.101809  0.110860
 0.073909  0.054425  0.020131  0.030400  0.064450
```

EIGEN VECTOR = 1

AT	ATE			
-0.26758019	-0.26758019	-0.27968979	-0.27733178	-0.26452390
-0.24528391	-0.24528391	-0.22331243	-0.20104147	-0.18058595
-0.16406074	-0.16406074	-0.15326360	-0.14872370	-0.15065306
-0.15926373	-0.15926373	-0.17298921	-0.18314899	-0.17928405
-0.15093535	-0.15093535	-0.09235999	-0.01667949	0.05826847
0.11464625	0.11464625	0.13896615	0.13514029	0.11143077
0.07609967	0.07609967	0.03526234	-0.01355276	-0.07496391
-0.15358941	-0.15358941	-0.25198907	-0.36448888	-0.48335636
-0.60085903	-0.60085903	-0.70824066	-0.79264999	-0.84021198
-0.83705162	-0.83705162	-0.77606159	-0.67720542	-0.56721436
-0.47281965	-0.47281965	-0.41595747	-0.39938367	-0.42105907
-0.47894446	-0.47894446	-0.56620555	-0.65682777	-0.72000148
-0.72491702	-0.72491702			

EIGFN VECTOR = 2

AT	ATE			
0.01018497	0.01018497	0.02103964	0.02763786	0.03099923
0.03214338	0.03214338	0.03188454	0.03021540	0.02692328
0.02179549	0.02179549	0.01441395	0.00353906	-0.01227416
-0.03447072	-0.03447072	-0.06319349	-0.09337694	-0.11865344
-0.13265535	-0.13265535	-0.13073982	-0.11516316	-0.08990646
-0.05895082	-0.05895082	-0.02589342	0.00720430	0.03866480
0.06681054	0.06681054	0.09030554	0.10917991	0.12380530
0.13455337	0.13455337	0.14171794	0.14528152	0.14514880
0.14122446	0.14122446	0.13268773	0.11581598	0.08616112
0.03927506	0.03927506	-0.02644243	-0.10120019	-0.17235925
-0.22728059	-0.22728059	-0.25599494	-0.25921183	-0.24031051
-0.20267023	-0.20267024	-0.15233995	-0.10604743	-0.08319018
-0.10316569	-0.10316569			

EIGEN VECTOR = 3

AT	ATE			
0.02541859	0.02541859	0.03365331	0.04269781	0.04930801
0.05023987	0.05023987	0.04304565	0.02846297	0.00802581
-0.01673187	-0.01673187	-0.04347976	-0.06670222	-0.08008723
-0.07732280	-0.07732280	-0.05490357	-0.02055065	0.01520818
0.04184518	0.04184518	0.05160645	0.04783366	0.03664231
0.02414793	0.02414793	0.01557659	0.01259662	0.01598688
0.02652624	0.02652624	0.04388273	0.06328092	0.07883455
0.08465734	0.08465734	0.07664478	0.05781931	0.03298518
0.00694659	0.00694659	-0.01598599	-0.03347715	-0.04368523
-0.04476856	-0.04476857	-0.03574564	-0.01907552	0.00192261
0.02392956	0.02392956	0.04384923	0.05947799	0.06883530
0.06994065	0.06994065	0.06103661	0.04125822	0.00996362
-0.03348906	-0.03348906			

EIGEN VECTOR = 4

AT	ATE			
0.03256336	0.03256336	0.04479571	0.04058059	0.02648697
0.00908378	0.00908378	-0.00595651	-0.01654750	-0.02149932
-0.01962205	-0.01962205	-0.01062232	0.00220720	0.01467734
0.02259894	0.02259894	0.02306442	0.01829267	0.01178418
0.00703945	0.00703945	0.00683650	0.01106350	0.01888614
0.02947013	0.02947013	0.04159320	0.05248114	0.05897180
0.05790302	0.05790302	0.04738817	0.03064277	0.01215791
-0.00357535	-0.00357535	-0.01300617	-0.01634464	-0.01474107
-0.00934579	-0.00934579	-0.00129494	0.00833207	0.01847001
0.02805366	0.02805366	0.03597661	0.04096780	0.04171499
0.03690593	0.03690593	0.02570913	0.00921601	-0.01100128
-0.03337056	-0.03337056	-0.05583895	-0.07443062	-0.08468901
-0.08215756	-0.08215756			

B	BEXT			
-0.01519100	-0.01519100	-0.00401279	0.00303857	0.00708990
0.00926800	0.00926800	0.01057182	0.01148875	0.01237830
0.01360000	0.01360000	0.01538547	0.01745482	0.01940026
0.02081400	0.02081400	0.02144468	0.02166659	0.02201046
0.02300700	0.02300700	0.02521233	0.02928419	0.03590570
0.04576000	0.04576000	0.05907149	0.07422966	0.08916525
0.10180900	0.10180900	0.11044760	0.11479142	0.11490679
0.11086000	0.11086000	0.10302255	0.09298665	0.08264967
0.07390900	0.07390900	0.06806152	0.06400211	0.06002514
0.05442500	0.05442500	0.04609685	0.03633904	0.02705071
0.02013100	0.02013100	0.01710841	0.01802886	0.02256763
0.03040000	0.03040000	0.04083062	0.05168154	0.06040419
0.06445000	0.06445000			

```
$JOB   ACCT-NUM,VERNER,TIME=60
C      **********************************************************************
C
C             PROGRAM 4.2 :  PLOTFP
C
C      **********************************************************************
```

```
3JOB PLOTFP; CHARGE BERGHOUT; CLASS 3;
USER 080002/VHJ
BEGIN
3COMPILE PLOTFP FORTRAN;
DATA



3DATA FILE5



3END JOB
```

```
C       ********************************************************************
C
C       PROGRAM : 4.2
C       TITLE : PLOTFP
C       PURPOSE : PLOT FERTILITY PARAMETERS
C       DATA : A(T) AND B(T) EXTENDED
C       OUTPUT : LINEPRINTER PLOTS OF A(T) AND B(T) EXTENDED
C
C       ********************************************************************
$RESET LIST
$INCLUDE 'OLIB/FS/PLOTT'
$SET LIST
        REAL PL(49,2)
        X=58.25
        DO 10 I=1,49
        X=X+.25
     10 PL(I,1)=X
        READ(5,100)(PL(J,2),J=1,49)
        N=21
        CALL PLOTR(N,PL,49,2,70.5,58.5,-0.5,0.5)
        DO 20 I=1,4
        READ(5,200)(PL(J,2),J=1,49)
        N=I+10
        CALL PLOTR(N,PL,49,2,70.5,58.5,-0.5,0.5)
     20 CONTINUE
    100 FORMAT(4F20.8)
    200 FORMAT(5F16.8)
        STOP
        END
```

****

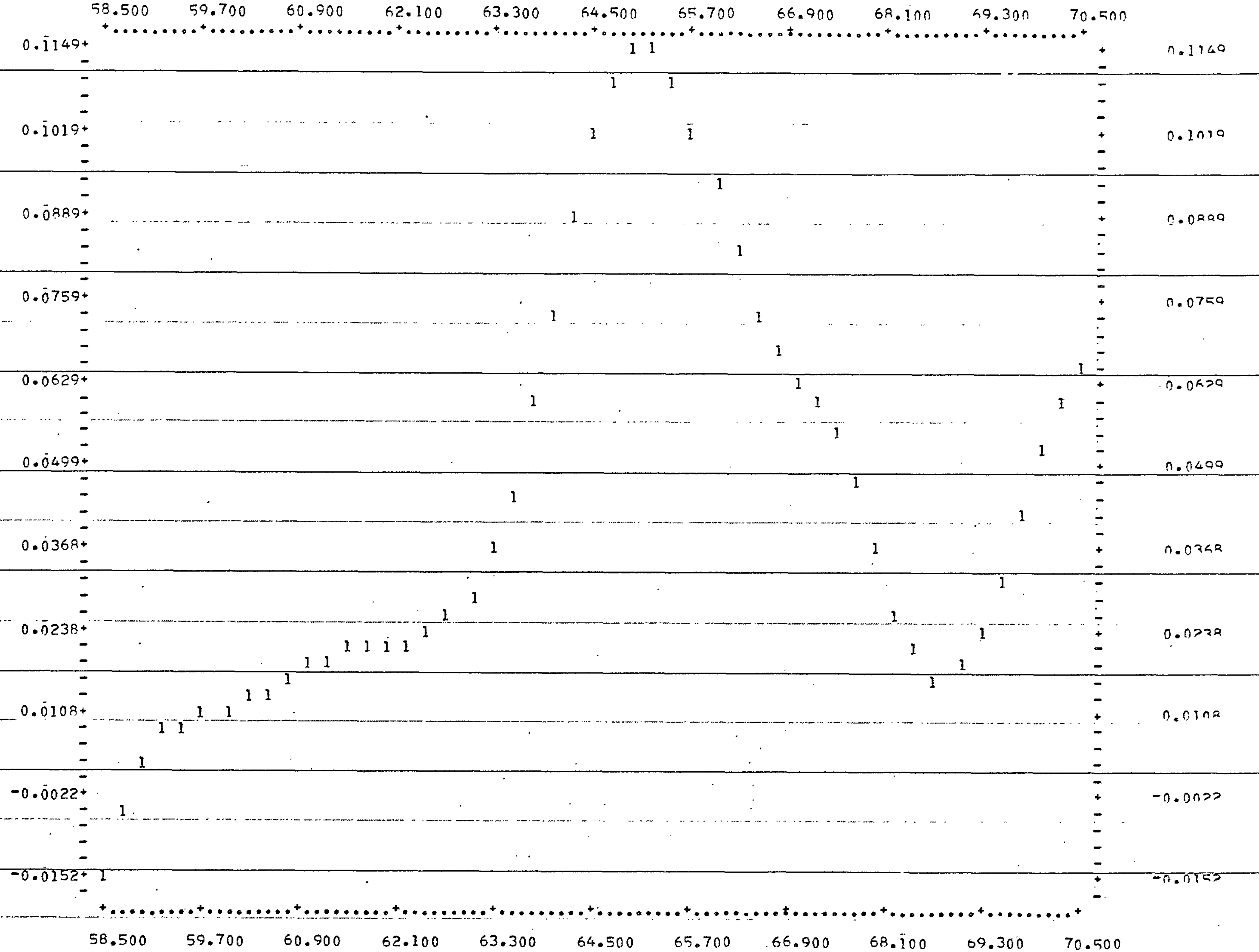
DATA FOR
    PROGRAM 4.2 : PLOTFP

****

```
 -0.01519100   -0.00401279    0.00303857    0.00708990
  0.00926800    0.01057182    0.01148875    0.01237830
  0.01360000    0.01538547    0.01745482    0.01940026
  0.02081400    0.02144468    0.02166659    0.02201046
  0.02300700    0.02521233    0.02928419    0.03590570
  0.04576000    0.05907149    0.07422966    0.08916525
  0.10180900    0.11044760    0.11479142    0.11490679
  0.11086000    0.10302255    0.09298665    0.08264967
  0.07390900    0.06806152    0.06400211    0.06002514
  0.05442500    0.04609685    0.03633904    0.02705071
  0.02013100    0.01710841    0.01802886    0.02256763
  0.03040000    0.04083062    0.05168154    0.06040419
  0.06445000
-0.26758019  -0.27968979  -0.27733178  -0.26452390  -0.24528391
-0.22331243  -0.20104147  -0.18058595  -0.16406074  -0.15326360
-0.14872370  -0.15065306  -0.15926373  -0.17298921  -0.18314899
-0.17928405  -0.15093535  -0.09235999  -0.01667949   0.05826847
 0.11464625   0.13896615   0.13514029   0.11143077   0.07609967
 0.03526234  -0.01355276  -0.07496391  -0.15358941  -0.25198907
-0.36448888  -0.48335636  -0.60085903  -0.70824066  -0.79264999
-0.84021198  -0.83705162  -0.77606159  -0.67720542  -0.56721436
-0.47281965  -0.41595746  -0.39938367  -0.42105907  -0.47894447
-0.56620556  -0.65682779  -0.72000150  -0.72491703
 0.01018497   0.02103964   0.02763786   0.03099923   0.03214338
 0.03188454   0.03021540   0.02692328   0.02179549   0.01441395
 0.00353906  -0.01227416  -0.03447072  -0.06319349  -0.09337694
-0.11865344  -0.13265535  -0.13073982  -0.11516316  -0.08990646
-0.05895082  -0.02589342   0.00720430   0.03866480   0.06681054
 0.09030554   0.10917991   0.12380530   0.13455337   0.14171794
 0.14528152   0.14514880   0.14122446   0.13268773   0.11581598
 0.08616112   0.03927506  -0.02644243  -0.10120019  -0.17235925
-0.22728059  -0.25599494  -0.25921183  -0.24031051  -0.20267024
-0.15233995  -0.10604743  -0.08319018  -0.10316569
 0.02541859   0.03365331   0.04269781   0.04930801   0.05023987
 0.04304565   0.02846297   0.00802581  -0.01673187  -0.04347976
-0.06670222  -0.08008723  -0.07732280  -0.05490357  -0.02055065
 0.01520818   0.04184518   0.05160645   0.04783366   0.03664231
 0.02414793   0.01557659   0.01259662   0.01598688   0.02652624
 0.04388273   0.06328092   0.07883455   0.08465734   0.07664478
 0.05781931   0.03298518   0.00694659  -0.01598599  -0.03347715
-0.04368523  -0.04476857  -0.03574564  -0.01907552   0.00192261
 0.02392956   0.04384923   0.05947799   0.06883530   0.06994065
 0.06103661   0.04125822   0.00996362  -0.03348906
 0.03256336   0.04479571   0.04058059   0.02648697   0.00908378
```

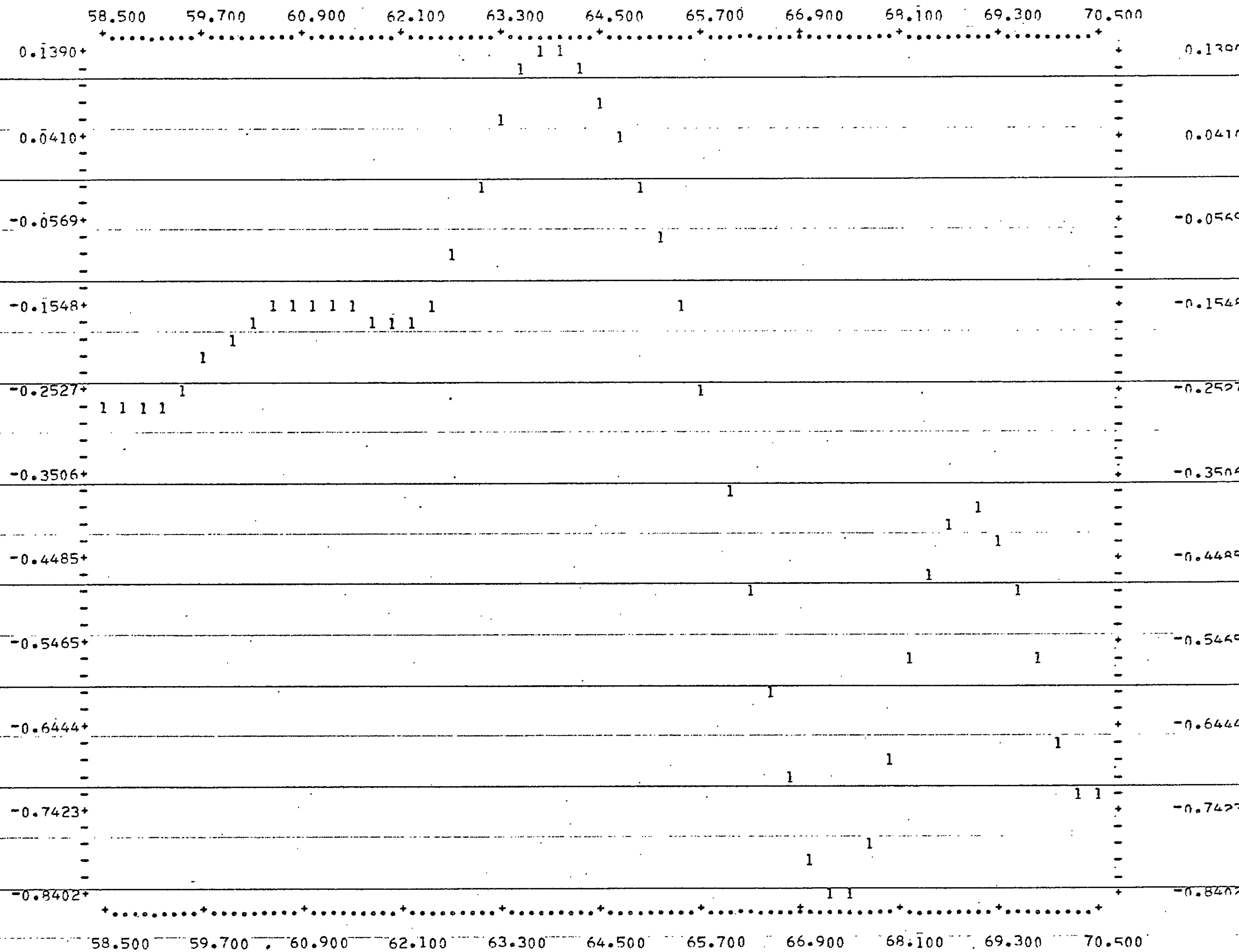
-0.00595651	-0.01654750	-0.02149932	-0.01962205	-0.01062232
0.00220720	0.01467734	0.02259894	0.02306442	0.01829267
0.01178418	0.00703945	0.00683650	0.01106350	0.01888614
0.02947013	0.04159320	0.05248114	0.05897180	0.05790302
0.04738817	0.03064277	0.01215791	-0.00357535	-0.01300617
-0.01634464	-0.01474107	-0.00934579	-0.00129494	0.00833207
0.01847001	0.02805366	0.03597661	0.04096780	0.04171499
0.03690593	0.02570913	0.00921601	-0.01100128	-0.03337056
-0.05583895	-0.07443062	-0.08468901	-0.08215755	

CHART 21



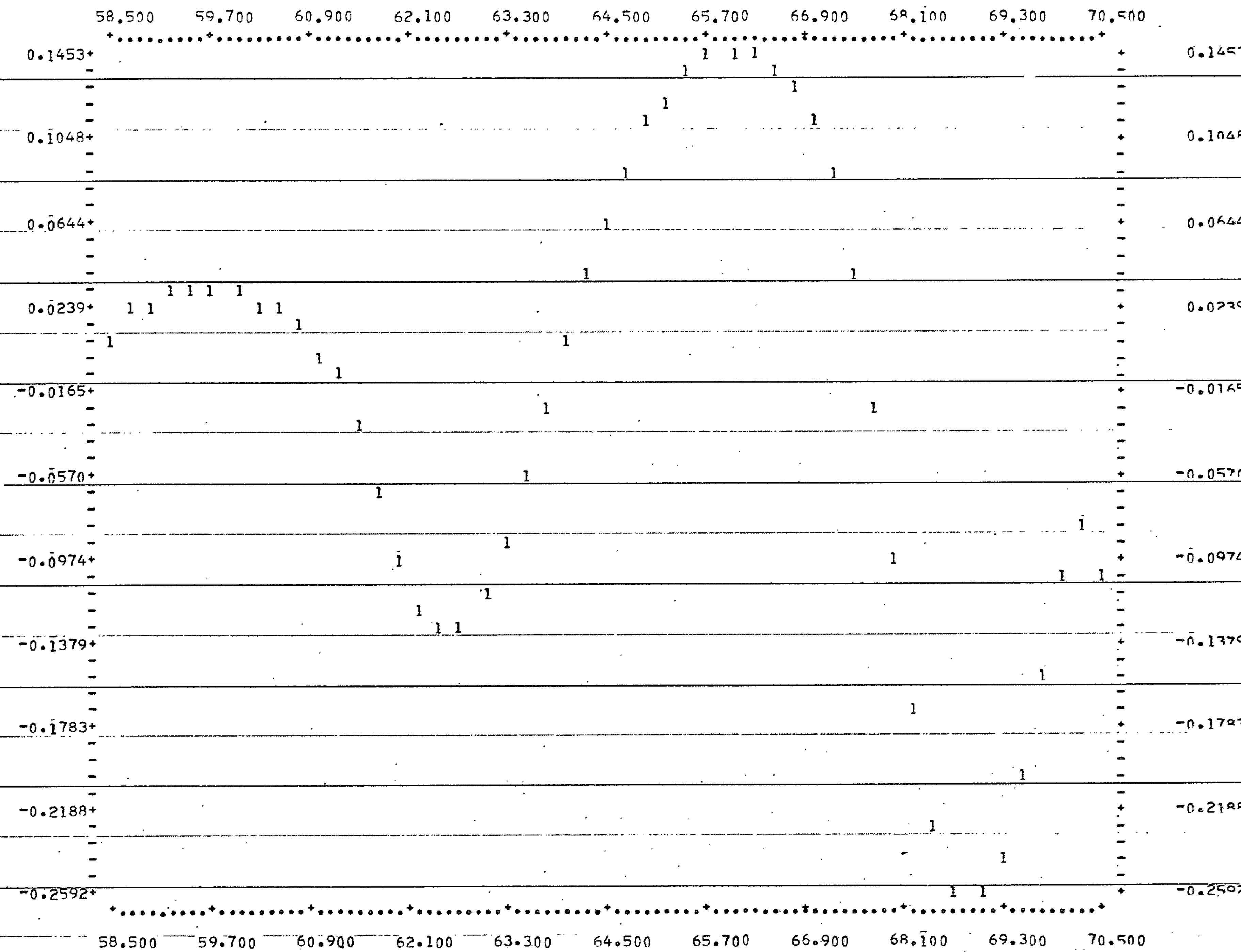
b(t) - Rate of Change of "Family Size"

CHART II



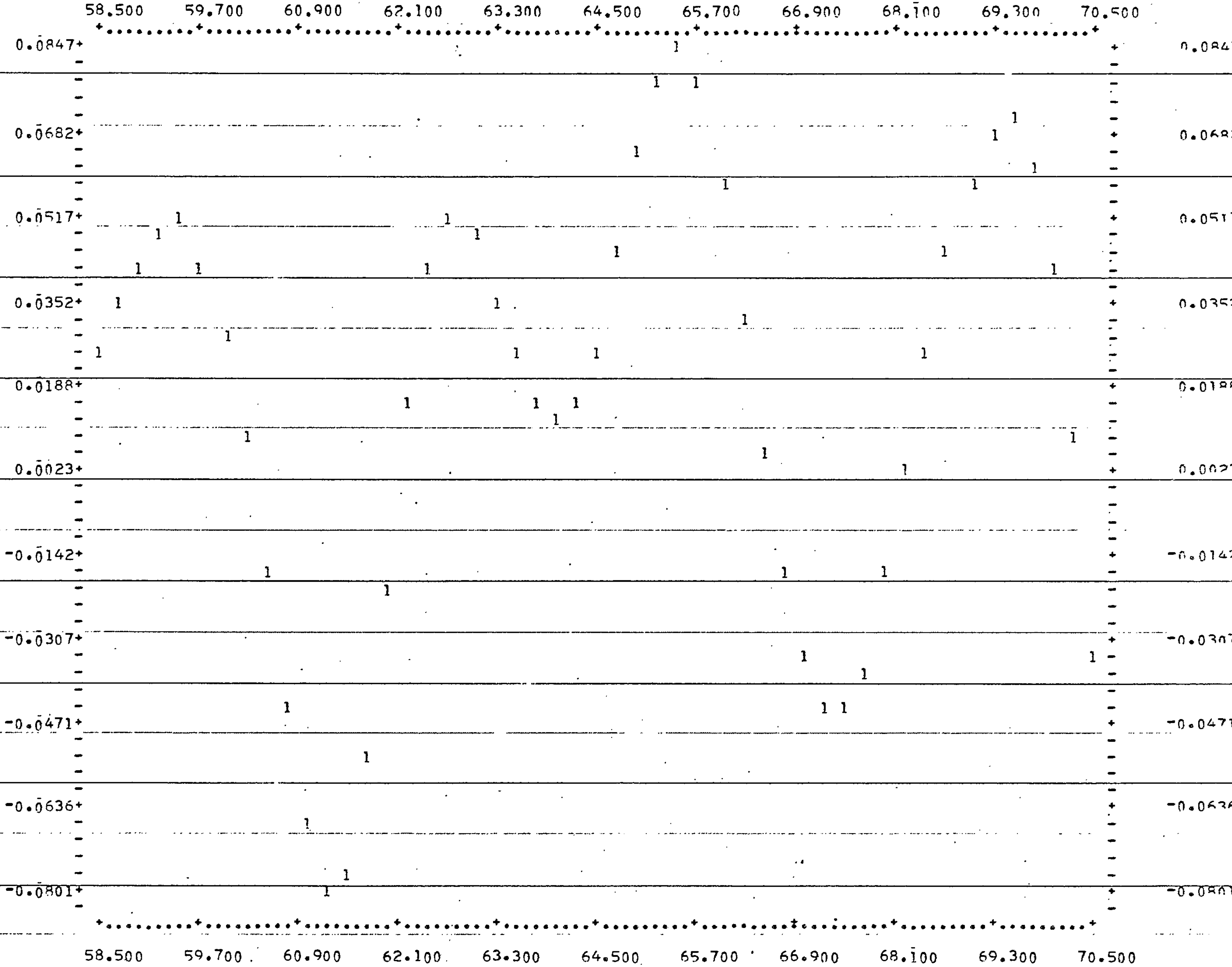
$a_1(t)$ - First Fertility Time Component

CHART 12



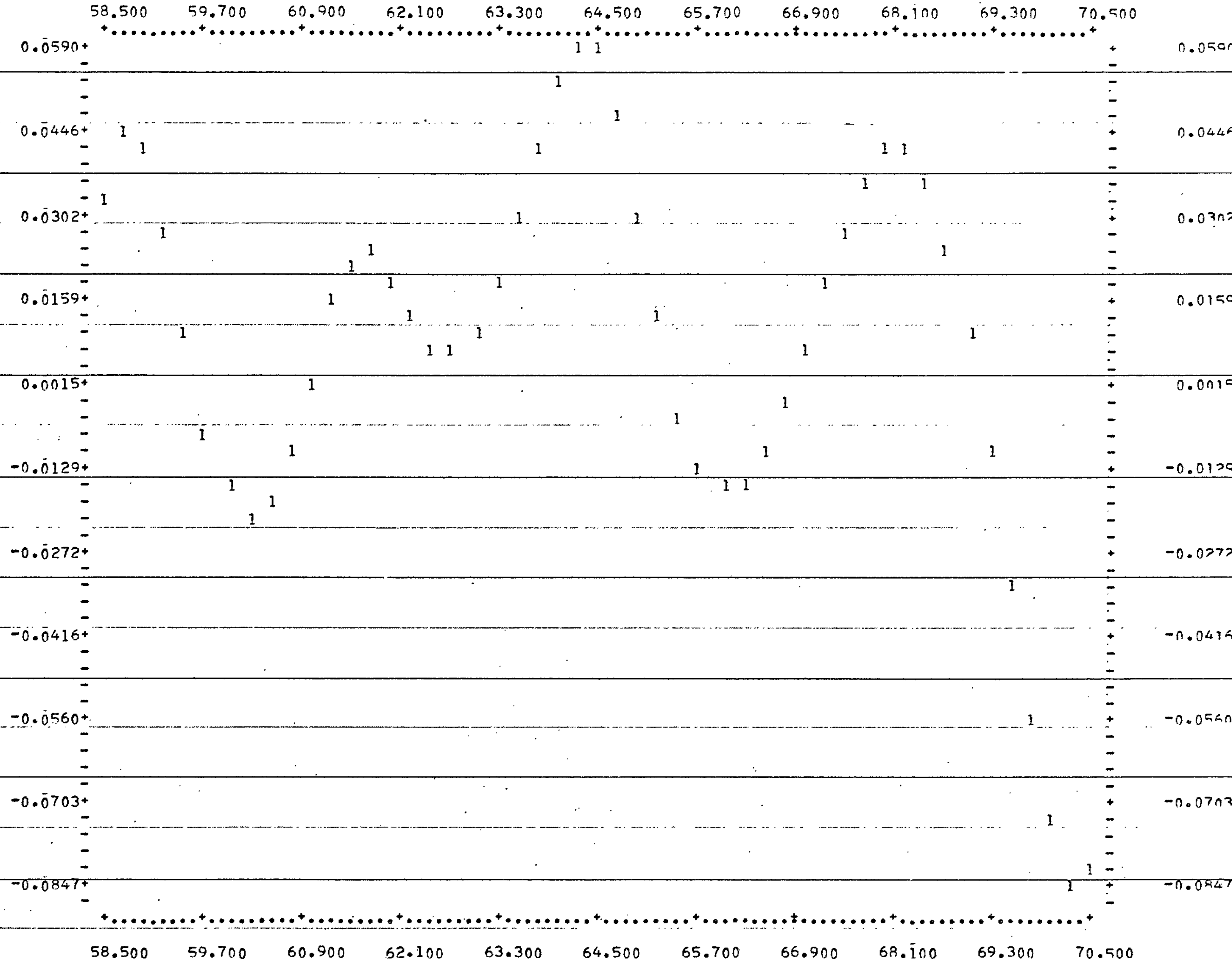
$a_2(t)$ - Second Fertility Time Component

CHART 13



$a_3(t)$ - Third Fertility Time Component

CHART 14



$a_4(t)$ - Fourth Fertility Time Component

4.2 Selected Economic Indicators

Nine different quartlerly economic indicators were selected as likely candidates as input time series for a model for the evolution of the fertility parameters $a_1(t)$, $b(t)$. Because of periodic changes in the methods of determination and reporting of these indicators, the most recently revised values were selected, and in one case, a scale change was required for earlier data. These nine indicators are:

1. Industrial Weekly Wages 1958-1970

   From: Canadian Statistical Review 11-003

   (June 60, Sept. 60, June 63, June 64, June 66, Jan. 69, June 69, June 71, Dec. 72)

   Format: 4F72, T73, A5, 3X

   Jan., Apr., July, Oct. XXIWW

2. Manufacturing Weekly Wages 1958-1970

   From: Canadian Statistical Review 11-003

   (June 60, Sept. 60, June 63, June 64, June 66, Jan. 69, June 69, June 71, Dec. 72)

   Format: 4F72, T73, A5, 3X

   Jan., Apr.,, July, Oct. XXMWW

3. Dwelling Units Completed 1958-1970

From: Canadian Statistical Review 11-003

(Dec. 60, June 64, June 66, June 69,

June 71, Dec. 72)

Format: 4I7, T73, A5, 3X

4 quarters XXDUC

4. Total Personal Income 1958-1970

From: Canadian Statistical Review 11-003

(Dec. 60, June 63, June 64, June 66,

Dec. 68, June 69, Dec. 69, Jan. 71, Mar. 71)

Format: 4I7, T73, A5, 3X

4 quarters XXTPI

5. Personal Disposable Income 1958-1970

From: Canadian Statistical Review 11-003

(Dec. 60, June 63, June 64, June 66, Dec. 68,

June 69, Dec. 69, Jan. 71, Mar. 71)

Format: 4I7, T73, A5, 3X

4 quarters XXPDI

6. Participation Rate Total 1958-1970

From: Historical labour force statistics,

actual data, seasonal factors, seasonally

adjusted data. 71-201 Annual

Format: 4F5.1, T73, A5, 3X

Jan., Apr., July, Oct. XXPRT

7. Participation Rate Females 1958-1970

From: Historical labour force statistics

actual data, seasonal factors, seasonally

adjusted data. 71-201 Annual

Format: 4F5.1, T73, A5, 3X

Jan., Apr., July, Oct. XXPRF

8. Unemployment Rate 1958-1970

- seasonally adjusted

From: Catalogue 71-201 Annual (see above)

Format: 4F5.1, T73, A5, 3X

Jan., April, July, Oct. XXUER

9. Index of Industrial Production 1958-1970

- seasonally adjusted

From: Canadian Statistical Review 11-003

(1958-1962 figures)

Dominion Bureau of Statistics - Catalogue

61-510 Occasional (1963-1967 figures)

Dominion Bureau of Statistics 61-005

Monthly - 1973 Supplement

(1968-1970 figures)

Format: 4F6.1, T73, A5, 3X

1963-1970 - 4 quarters XXIIP

1958-1962 - Jan., Apr., July, Oct.

Note: 1958 → 1960 figures multiplied by

101.6/173.4

101.6 = Index (revised)

173.4 = Index for July 1961 (unrevised)

Personal Disposable Income and Total Personal Income in millions of dollars.

Manufacturing and Industrial Wages in dollars.

In additon four modification of three of these indicators were calculated and used:

10. Dwelling Units Completed 1958-1970

- Running Averages

From: Canadian Statistica Review 11-003

(Dec. 60, June 64, June 66, June 69,

June 71, Dec. 72)

Format: 4F9.2, T73, A5, 3X

4 quarters XXDUA

-averages taken over four previous quarters.

11. Dwelling Units Completed 1958-1970

- Running Averages - Smoothed

Values obtained in 10.were used to obtain the fourth degree polynomial approximation which minimizes the sum of the squares of the residuals. XXHA

12. Participation Rates - Smoothed 1958-1970

Values given by 6. were used to obtain the corresponding least squares polynomial of degree four. XXPR

13. Unemployment - Smoothed 1958-1970

Values given in 8. were used to obtain the corresponding least squares polynomial of degree six. XXUR

```
$JOB    ACCT-NUM,VERNER,TIME=60
C       ***********************************************************************
C
C             PROGRAM 4.3 :   UPRINT
C
C       ***********************************************************************
```

```
3JOB UPRINT; CHARGE BERGHOUT; CLASS 3;
USER 080002/VHJ;
BEGIN
3COMPILE UPRINT FORTRAN;
DATA



3DATA FILE5



3END JOB
```

```
C       ****************************************************************
C
C       PROGRAM : 4.3
C       TITLE : UPRINT
C       PURPOSE : PRINT ECONOMIC INDICATORS
C       DATA : U-ARRAY
C       OUTPUT : PRINTS U
C
C       ****************************************************************
        DIMENSION SEL(7,13)
        DIMENSION A(20)
        DATA SEL/'INDU','STRI','AL W','AGES','    ','    ','    ',
     $           'MANU','FACT','URIN','G WA','GES ','    ','    ',
     $           'HOUS','ING ','COMP','LETI','ONS ','    ','    ',
     $           'PERS','ONAL',' INC','OME ','    ','    ','    ',
     $           'DISP','OSAB','LE I','NCOM','E   ','    ','    ',
     &     'PART','ICIP','ATIO','N RA','TE T','OTAL','    ',
     $           'PART','ICIP','ATIO','N RA','TE F','EMAL','ES  ',
     $           'UNEM','PLOY','MENT',' RAT','E   ','    ','    ',
     $           'INDU','STRI','AL P','RODU','CTIO','N   ','    ',
     $           'HOUS','ING ','AVER','AGES','    ','    ','    ',
     $           'HOUS','ING ','AVER','AGES',' - S','MOOT','HED ',
     $           'PART','ICIP','ATIO','N - ','SMOO','THED','DEG4',
     $           'UNEM','PLOY','MENT',' - S','MOOT','HED ','DEG6'/
*EXTENSION*    CC-1
        DO 10 I=1,13
           WRITE(6,200) (SEL(K,I),K=1,7)
           DO 10 J=1,13
           READ(5,100) A
           WRITE(6,101) A
     10 CONTINUE
    100 FORMAT(20A4)
    101 FORMAT('0',20A4)
    200 FORMAT('1',1X,7A4,//)
        STOP
        END
```

69.25	70.35	70.76	71.13	58IWW
72.34	73.26	73.76	74.66	59IWW
75.13	75.98	76.28	76.60	60IWW
76.99	78.12	78.32	79.07	61IWW
79.39	80.20	80.97	81.63	62IWW
82.01	83.53	83.35	84.54	63IWW
85.04	86.19	86.56	88.33	64IWW
89.07	90.32	90.95	93.56	65IWW
93.71	95.90	96.97	99.43	66IWW
99.57	102.32	103.35	105.10	67IWW
106.11	108.95	109.98	112.90	68IWW
114.02	116.43	118.21	120.71	69IWW
123.08	125.53	127.02	130.23	70IWW

MANUFACTURING WAGES

71.61	72.92	72.62	73.36	58MWW
75.16	75.69	75.56	77.07	59MWW
77.90	78.40	78.18	78.95	60MWW
79.65	80.95	80.34	81.78	61MWW
82.29	83.11	83.13	84.34	62MWW
85.12	86.72	85.30	88.23	63MWW
89.56	90.39	89.81	91.89	64MWW
93.18	94.88	93.59	97.10	65MWW
97.68	99.90	99.24	102.86	66MWW
103.06	106.05	106.12	109.32	67MWW
110.33	114.33	113.20	118.08	68MWW
119.47	121.84	122.18	125.93	69MWW
129.28	131.43	131.83	135.72	70MWW

26315	31993	37698	50680	58DUC
24697	38254	37382	45338	59DUC
28783	28747	31091	35136	60DUC
22835	24594	32981	35198	61DUC
26776	25698	34558	39700	62DUC
25093	30094	35398	37606	63DUC
44385	32050	35337	39191	64DUC
43762	36850	34719	37706	65DUC
35964	47614	39586	39028	66DUC
30003	35552	37600	46087	67DUC
35212	40442	43862	51477	68DUC
37651	52553	50011	55611	69DUC
37127	44019	47968	46713	70DUC

PERSONAL INCOME

23892	24340	24568	24960	58TPI
25424	26060	26024	26252	59TPI
27096	27216	27472	27860	60TPI
27636	28084	28876	29376	61TPI
30224	30864	30752	31428	62TPI
31688	32932	33180	33936	63TPI
34520	34632	35384	36076	64TPI
37996	37872	39476	40264	65TPI
41888	42392	43412	44560	66TPI
46040	46828	47456	48484	67TPI
49412	51024	52368	53692	68TPI
54500	60568	62568	63624	69TPI
64688	65428	66344	67940	70TPI

DISPOSABLE INCOME

22060	22652	22794	23088	58PDI
23468	24016	23876	24048	59PDI
24780	24832	25120	25472	60PDI
25524	25576	26308	26812	61PDI
27584	28188	27992	28624	62PDI
28828	30156	30240	30848	63PDI
31324	31236	31884	32456	64PDI
34196	34020	35544	36200	65PDI
37632	38184	38808	39692	66PDI
40508	41848	41916	42564	67PDI
43000	44788	45652	46416	68PDI
46932	49816	51490	51988	69PDI
53008	52816	53656	54900	70PDI

PARTICIPATION RATE TOTAL

54.0	54.3	53.8	53.8	58PRT
53.9	53.8	53.7	53.8	59PRT
54.0	54.1	53.8	54.5	60PRT
54.5	54.6	53.9	53.8	61PRT
53.8	54.0	53.9	53.4	62PRT
53.6	53.6	53.9	54.2	63PRT
54.2	54.0	54.1	54.0	64PRT
54.2	54.3	54.4	54.3	65PRT
54.7	55.0	55.0	55.5	66PRT
55.3	55.2	55.7	55.[illegible]	67PRT
55.2	55.2	55.5	55.8	68PRT
56.1	56.3	55.6	55.4	69PRT
55.3	55.7	55.8	55.9	70PRT

PARTICIPATION RATE FEMALES

25.9	26.5	26.4	26.6	58PRF
26.4	26.6	26.7	26.9	59PRF
27.3	27.1	27.5	28.5	60PRF
28.8	29.0	28.6	28.4	61PRF
28.8	29.0	29.2	28.9	62PRT
28.8	28.8	29.8	30.5	63PRF
30.4	30.4	30.3	30.5	64PRF
30.9	31.4	31.4	31.5	65PRF
32.1	32.6	32.8	33.3	66PRF
33.2	33.5	34.1	33.5	67PRF
34.0	33.8	34.3	34.9	68PRF
35.2	35.8	35.1	35.1	69PRF
34.9	35.7	35.7	35.6	70PRF

UNEMPLOYMENT RATE

6.5	6.9	7.5	7.5	58UER
6.5	5.8	5.5	5.5	59UER
6.1	6.8	7.1	7.6	60UER
7.6	7.6	7.2	6.5	61UER
6.0	5.9	6.0	5.7	62UER
5.9	5.7	5.4	5.3	63UER
4.9	4.9	4.6	4.9	64UER
4.2	4.4	3.9	3.7	65UER
3.7	3.4	3.7	3.4	66UER
3.8	4.1	4.0	4.3	67UER
4.6	4.8	5.1	4.6	68UER
4.6	4.6	4.6	4.9	69UER
4.8	5.8	6.6	6.3	70UER

INDUSTRIAL PRODUCTION

150.8	151.3	152.0	152.4	58IIP
159.6	166.2	164.8	171.1	59IIP
173.5	164.3	168.0	166.3	60IIP
95.9	98.9	101.6	103.4	61IIP
105.2	107.6	110.2	109.8	62IIP
111.5	113.9	115.9	119.5	63IIP
123.2	125.9	127.7	129.7	64IIP
133.1	134.4	138.2	142.3	65IIP
144.9	145.4	145.7	148.0	66IIP
147.8	150.1	151.9	153.4	67IIP
155.1	160.9	163.7	168.1	68IIP
171.7	172.1	172.5	174.0	69IIP
177.0	174.7	174.8	174.6	70IIP

HOUSING AVERAGES

		32002.00	36671.50	58DUA
36267.00	37832.25	37753.25	36417.75	59DUA
37439.25	35062.50	33489.75	30939.25	60DUA
29452.25	28414.00	28886.50	28902.00	61DUA
29887.25	30163.25	30545.00	31670.50	62DUA
31249.75	32348.75	32571.25	32047.75	63DUA
36870.75	37359.75	37344.50	37740.75	64DUA
37585.00	38785.00	38630.50	38259.25	65DUA
36309.75	39000.75	40217.50	40548.00	66DUA
39057.75	36042.25	35545.75	37310.50	67DUA
38612.75	39835.25	41400.75	42748.25	68DUA
43358.00	46385.75	47923.00	48956.50	69DUA
48825.50	46692.00	46181.25		70DUA

HOUSING AVERAGES - SMOOTHED

		38134.272	36850.172	58HA
35739.444	34789.608	33988.659	33325.067	59HA
32787.781	32366.225	32050.300	31830.382	60HA
31697.324	31642.457	31657.585	31734.990	61HA
31867.432	32048.145	32270.841	32529.705	62HA
32819.404	33135.076	33472.337	33827.282	63HA
34196.479	34576.972	34966.285	35362.416	64HA
35763.837	36169.502	36578.835	36991.742	65HA
37408.601	37830.269	38258.078	38693.838	66HA
39139.832	39598.822	40074.047	40569.220	67HA
41088.532	41636.649	42218.714	42840.347	68HA
43507.644	44227.176	45005.993	45851.618	69HA
46772.052	47775.774	48871.737		70HA

PARTICIPATION - SMOOTHEDDFG4

		53.66	53.78	58PR
53.87	53.94	53.99	54.02	59PR
54.04	54.04	54.03	54.02	60PR
54.00	53.98	53.96	53.94	61PR
53.92	53.90	53.89	53.89	62PR
53.89	53.91	53.93	53.96	63PR
53.99	54.04	54.10	54.17	64PR
54.24	54.33	54.42	54.52	65PR
54.63	54.74	54.85	54.97	66PR
55.08	55.20	55.31	55.42	67PR
55.51	55.60	55.68	55.74	68PR
55.78	55.80	55.79	55.76	69PR
55.69	55.59	55.45		70PR

UNEMPLOYMENT - SMOOTHED DEG6

		7.671	6.849	58UR
6.370	6.145	6.102	6.178	59UR
6.323	6.497	6.670	6.817	60UR
6.924	6.978	6.976	6.916	61UR
6.800	6.635	6.426	6.184	62UR
5.917	5.635	5.349	5.068	63UR
4.801	4.555	4.336	4.151	64UR
4.001	3.890	3.816	3.780	65UR
3.777	3.804	3.856	3.927	66UR
4.010	4.100	4.189	4.275	67UR
4.352	4.419	4.478	4.533	68UR
4.595	4.676	4.797	4.985	69UR
5.276	5.714	6.353		70UR

```
$JOB   ACCT-NUM,VERNER,TIME=60
C      ************************************************************************
C
C           PROGRAM 4.4 :  PLOTEI
C
C      ************************************************************************
```

```
3JOB PLOTEI; CHARGE BERGHOUT; CLASS 3;
USER 080002/VHJ;
PROCESSTIME=300;
BEGIN
3COMPILE PLOTEI FORTRAN;
DATA



3DATA FILE5



3END JOB
```

```
C       *************************************************************************
C
C       PROGRAM : 4.4
C       TITLE : PLOTFI
C       PURPOSE : PLOTS ECONOMIC INDICATORS
C       DATA : U-ARRAY
C       OUTPUT : PLOTS U
C
C       *************************************************************************
$RESET LIST
$INCLUDE '0LIB/FS/PLOTT'
$SET LIST
        REAL PL(49,2),TEMP(52)
        DIMENSION VAR(13),VF(3)
        DATA VAR/'  F7.2','  F7.2','     I7','     I7','     I7','  F5.1',
     &  '  F5.1','  F5.1','  F6.1','  F9.2',' F10.3','  F6.2','  F6.2'/
        DATA VF/'    (4','      ',')     '/
        X=58.25
        DO 10 I=1,49
           X = X + .25
           PL(I,1) = X
   10      CONTINUE
        DO 20 I=1,13
        VF(2)=VAR(I)
        READ(5,VF)X,XX,(PL(J,2),J=1,49)
        IF (I.NE.9) GO TO 30
           X=101.6/173.4
           DO 40 J=1,10
   40      PL(J,2)=PL(J,2) * X
   30   CONTINUE
        CALL PLOTB(I,PL,49,2,70.5,58.5,-0.5,0.5)
   20   CONTINUE
        STOP
        END
```

CHART 1

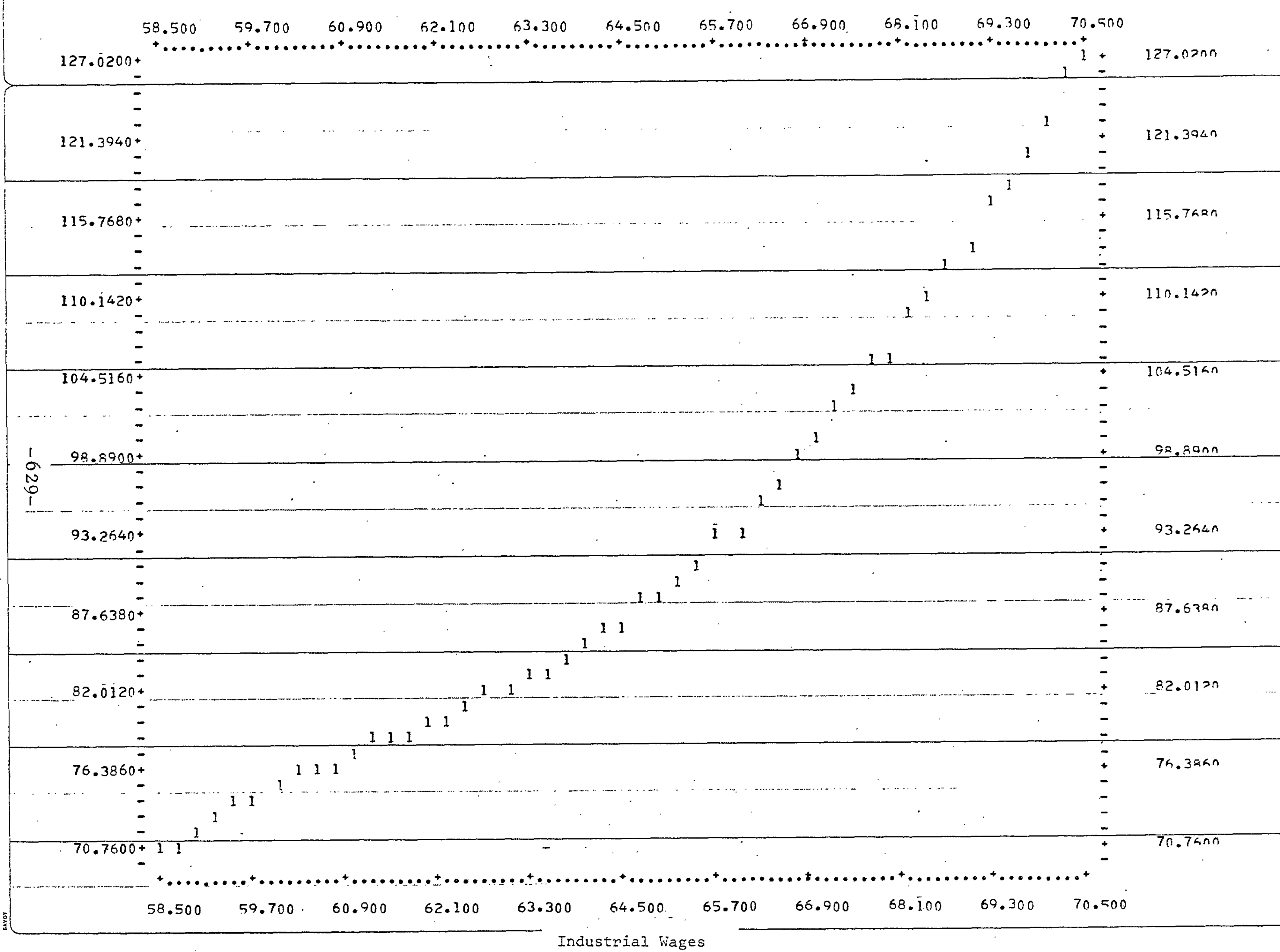
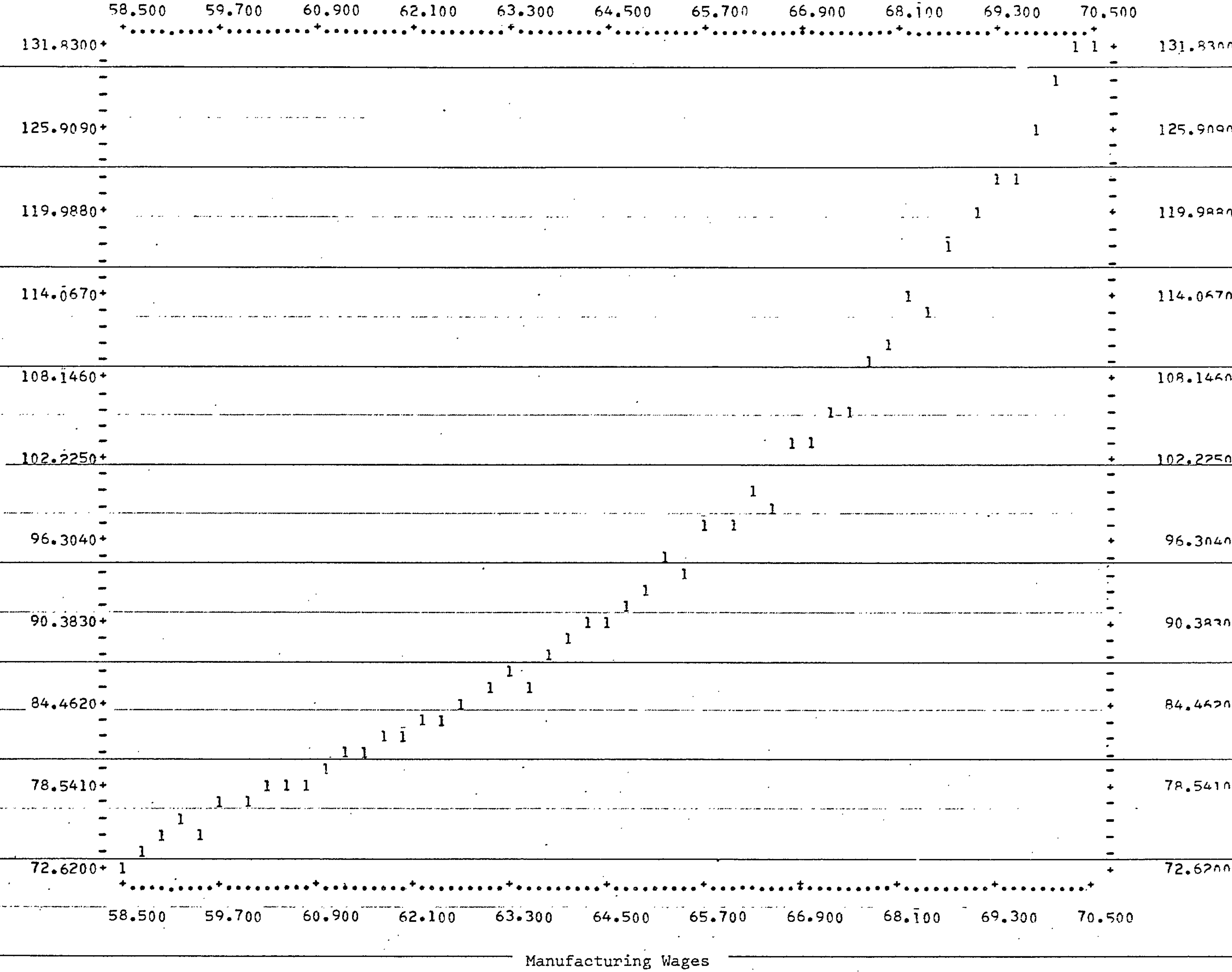

58.500
59.700
60.900
62.100
63.300
64.500
65.700
66.900
68.100
69.300
70.500
127.0200
121.3940
115.7680
110.1420
104.5160
98.8900
93.2640
87.6380
82.0120
76.3860
70.7600
Industrial Wages

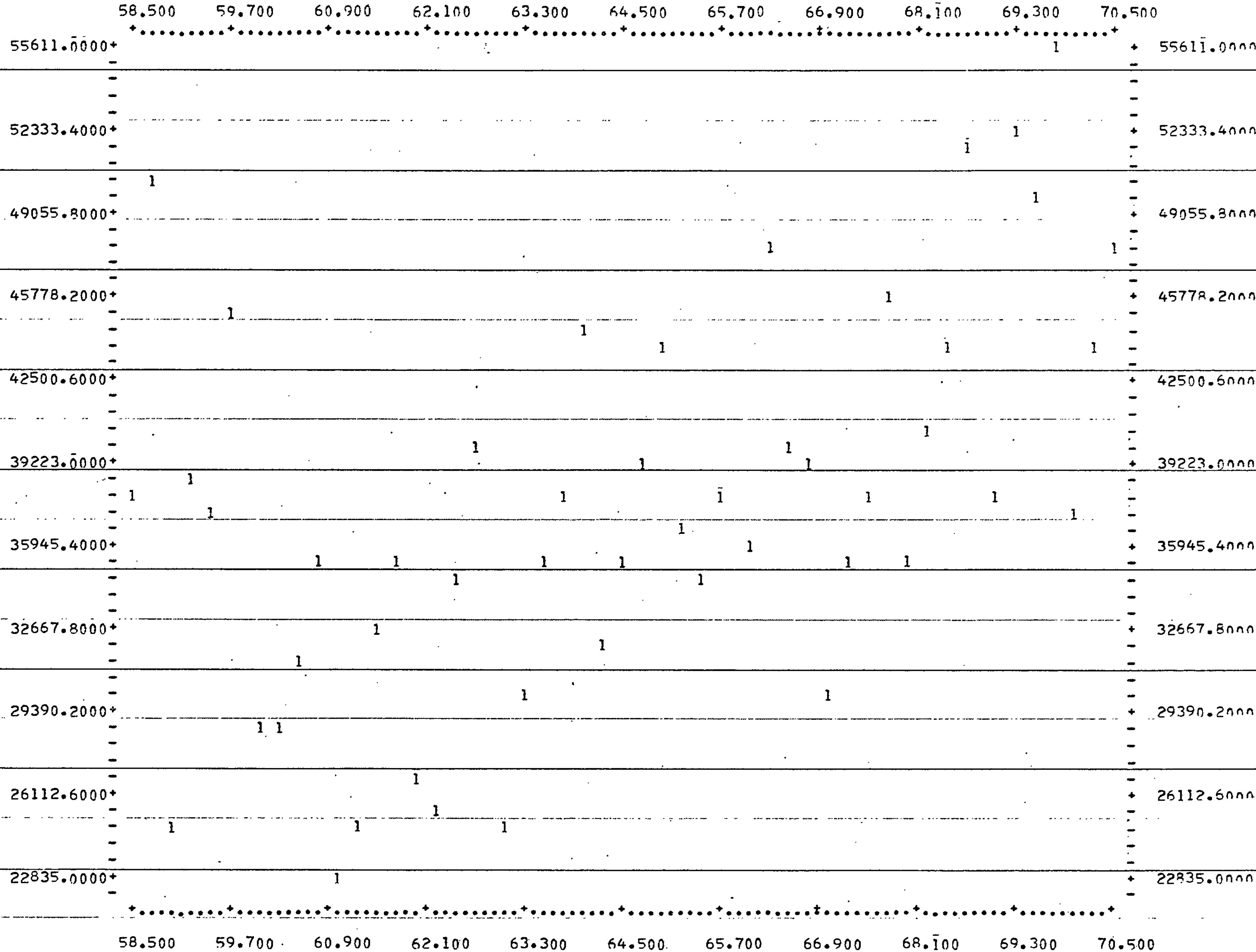
CHART 2



58.500
59.700
60.900
62.100
63.300
64.500
65.700
66.900
68.100
69.300
70.500
131.8300
125.9090
119.9880
114.0670
108.1460
102.2250
96.3040
90.3830
84.4620
78.5410
72.6200
Manufacturing Wages

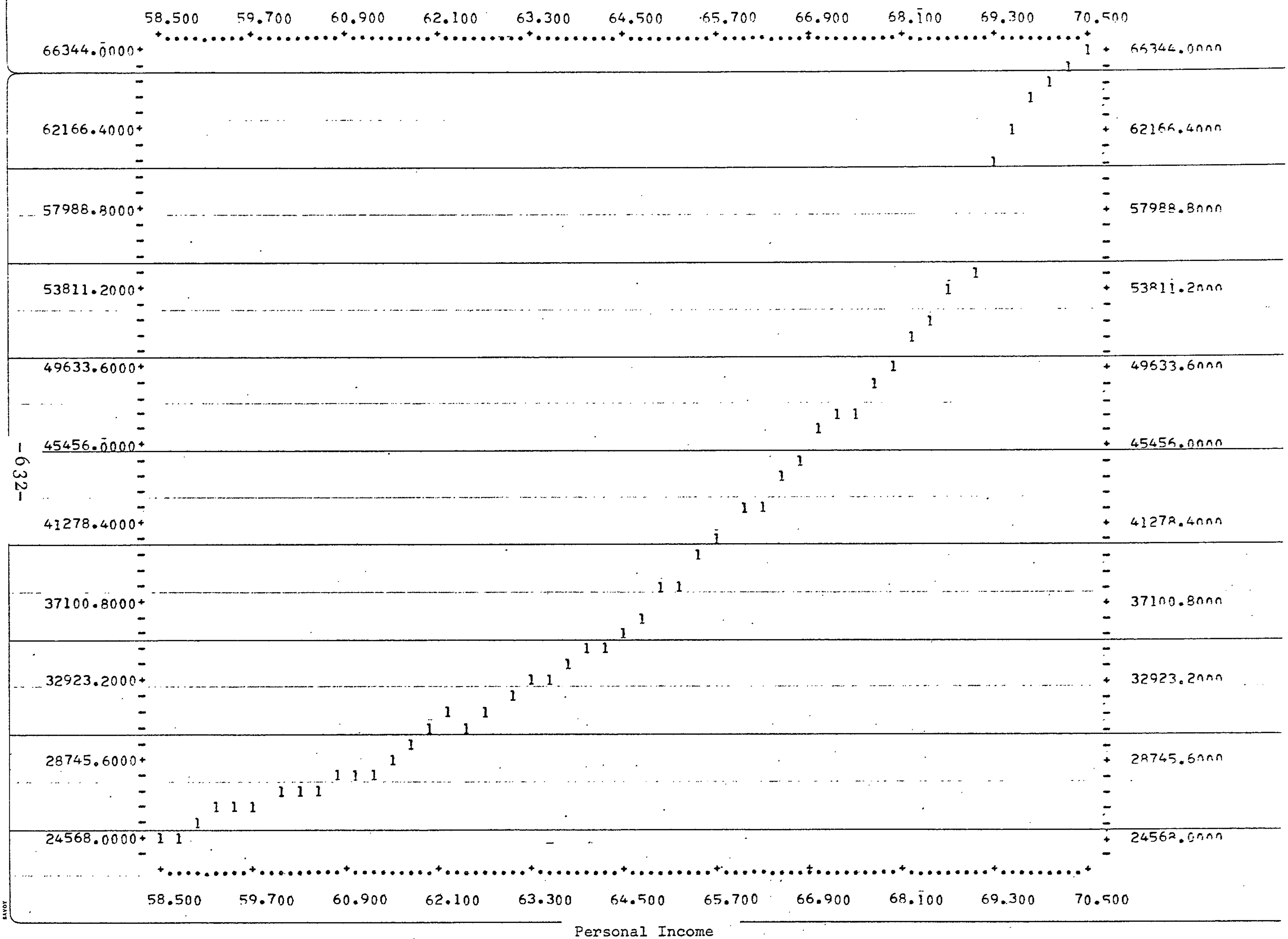
-630-

CHART 3



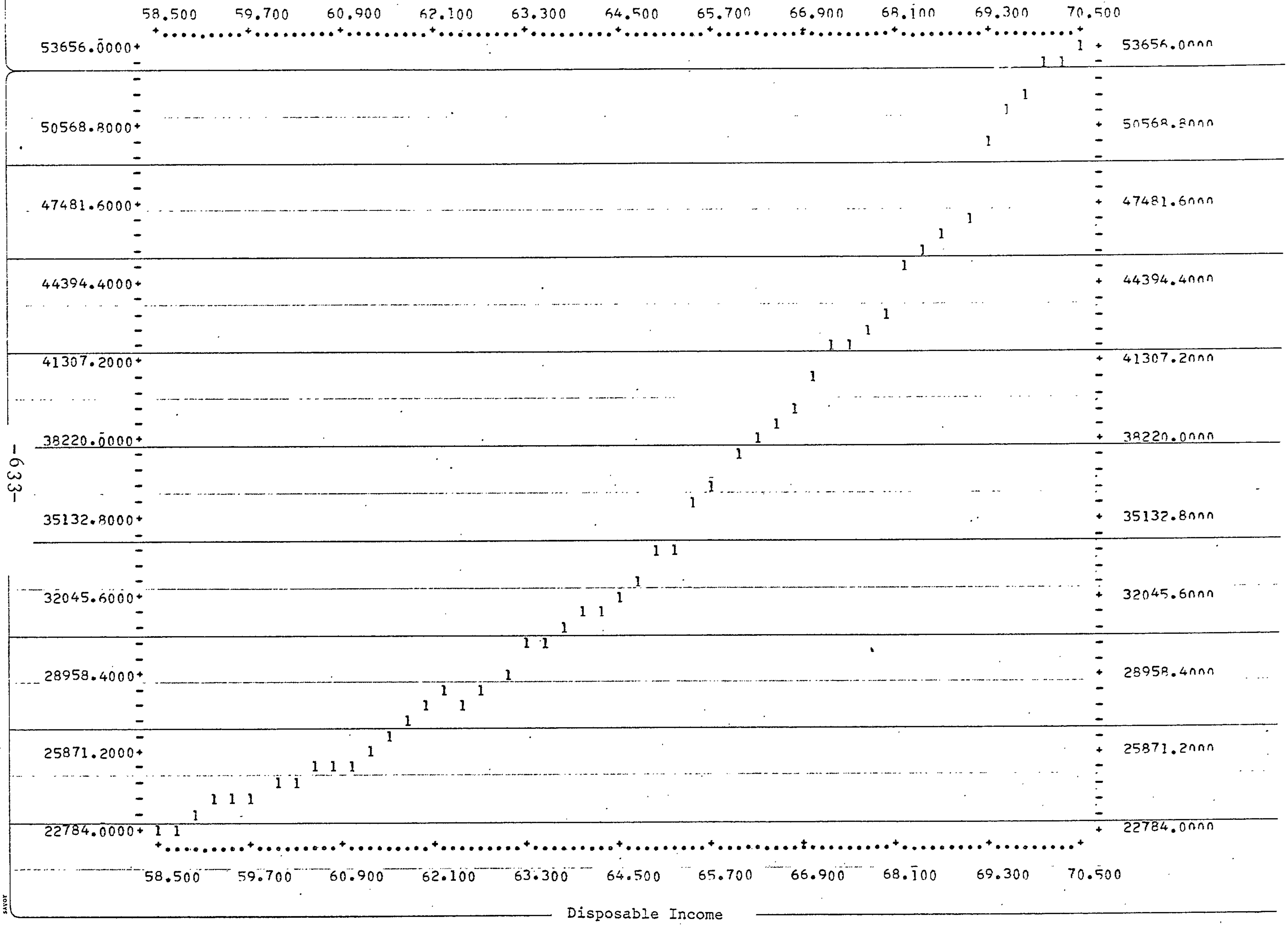
Housing Completions

CHART 4


58.500
59.700
60.900
62.100
63.300
64.500
65.700
66.900
68.100
69.300
70.500
66344.0000
62166.4000
57988.8000
53811.2000
49633.6000
45456.0000
41278.4000
37100.8000
32923.2000
28745.6000
24568.0000
Personal Income

-632-

CHART 5



58.500
59.700
60.900
62.100
63.300
64.500
65.700
66.900
68.100
69.300
70.500
53656.0000
50568.8000
47481.6000
44394.4000
41307.2000
38220.0000
35132.8000
32045.6000
28958.4000
25871.2000
22784.0000
Disposable Income

CHART 6

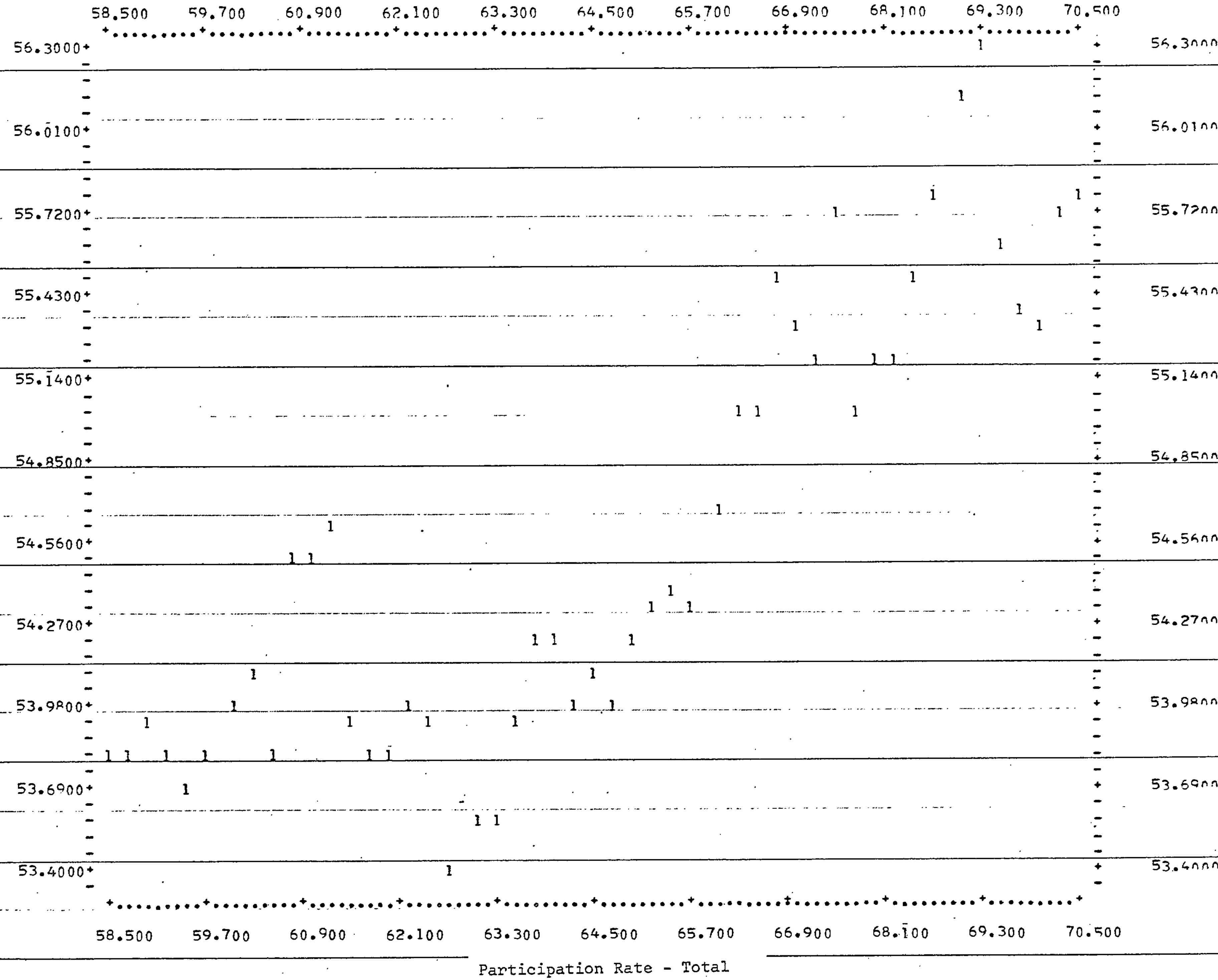

58.500
59.700
60.900
62.100
63.300
64.500
65.700
66.900
68.100
69.300
70.500
56.3000
56.0100
55.7200
55.4300
55.1400
54.8500
54.5600
54.2700
53.9800
53.6900
53.4000
Participation Rate - Total

CHART 7

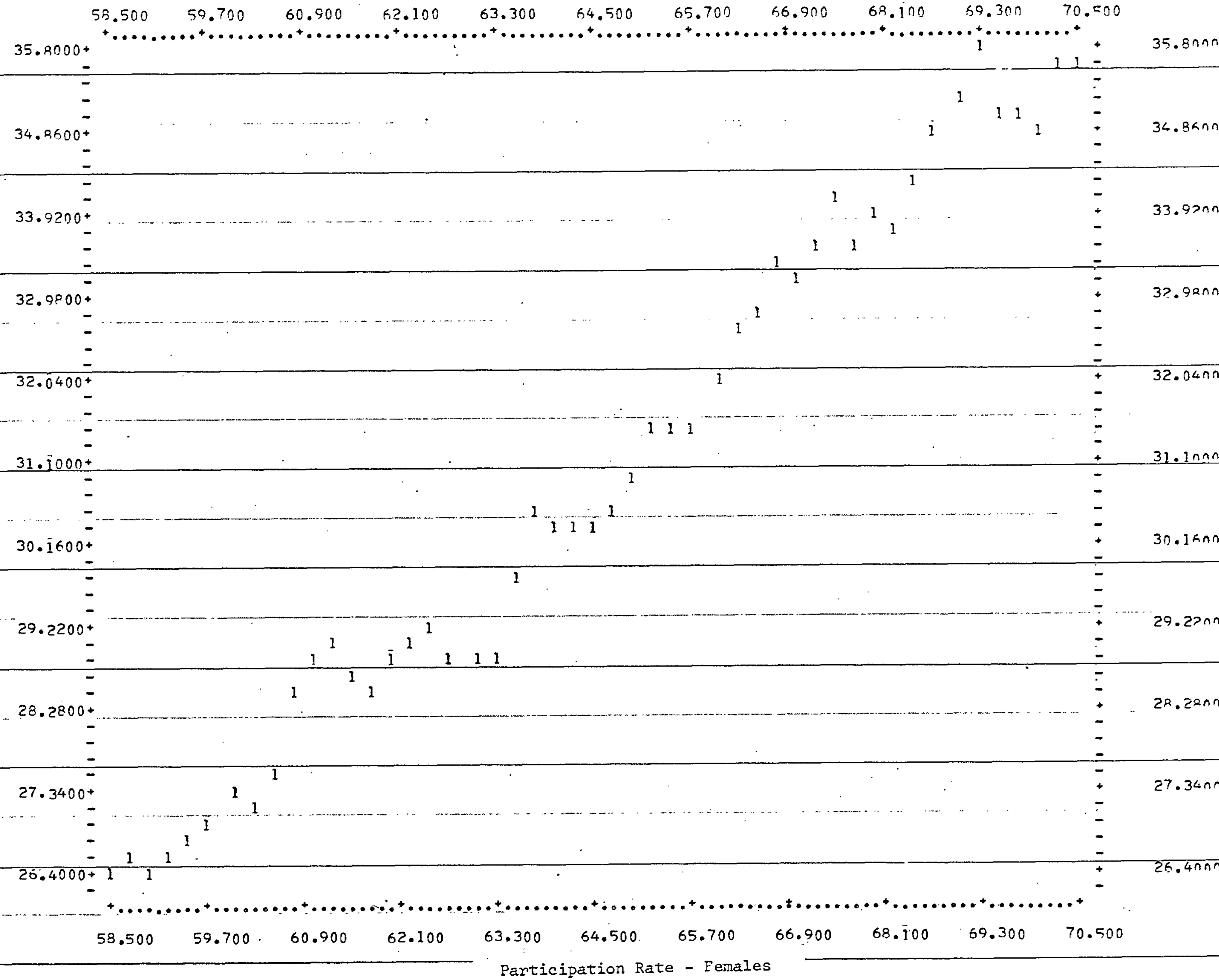
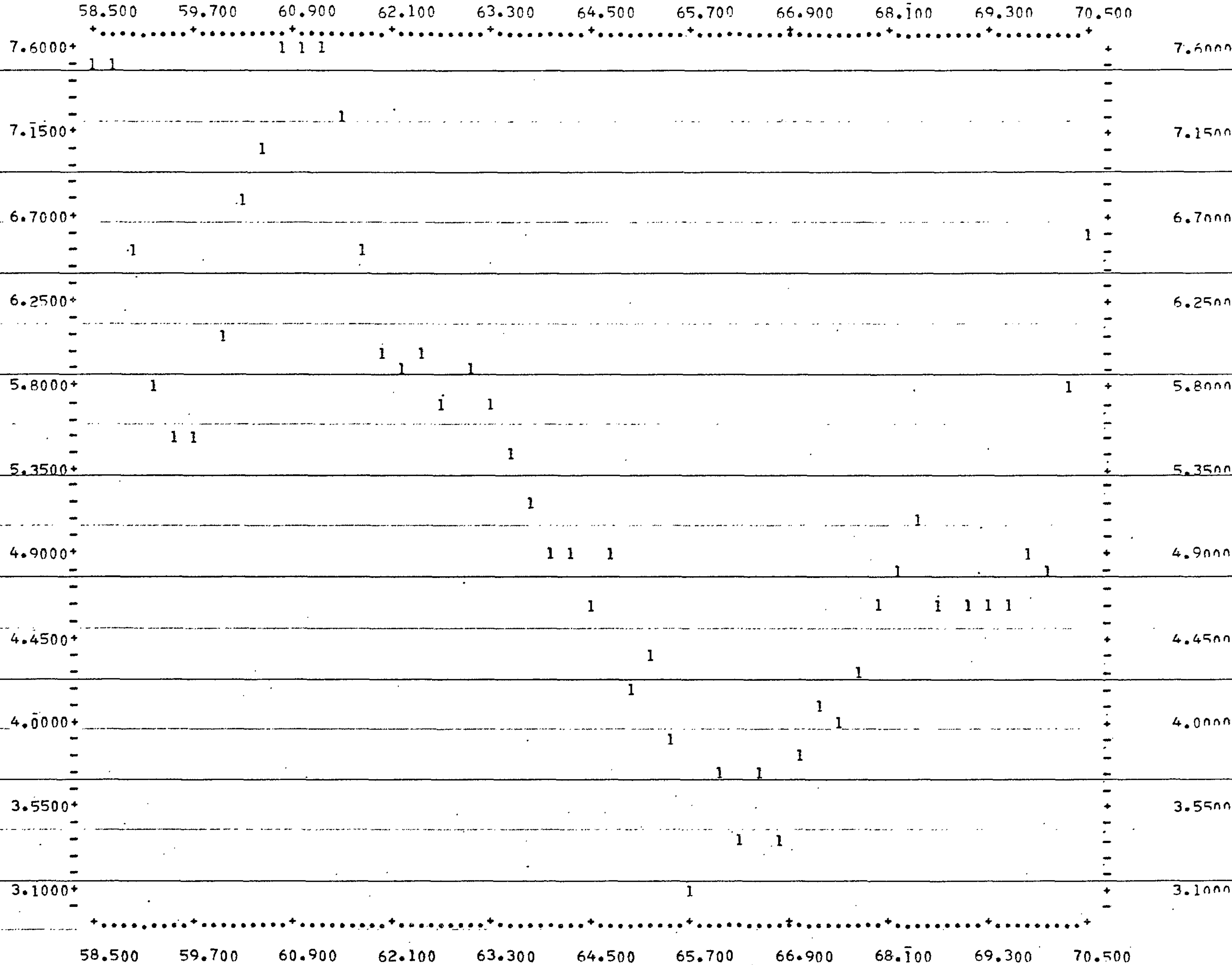
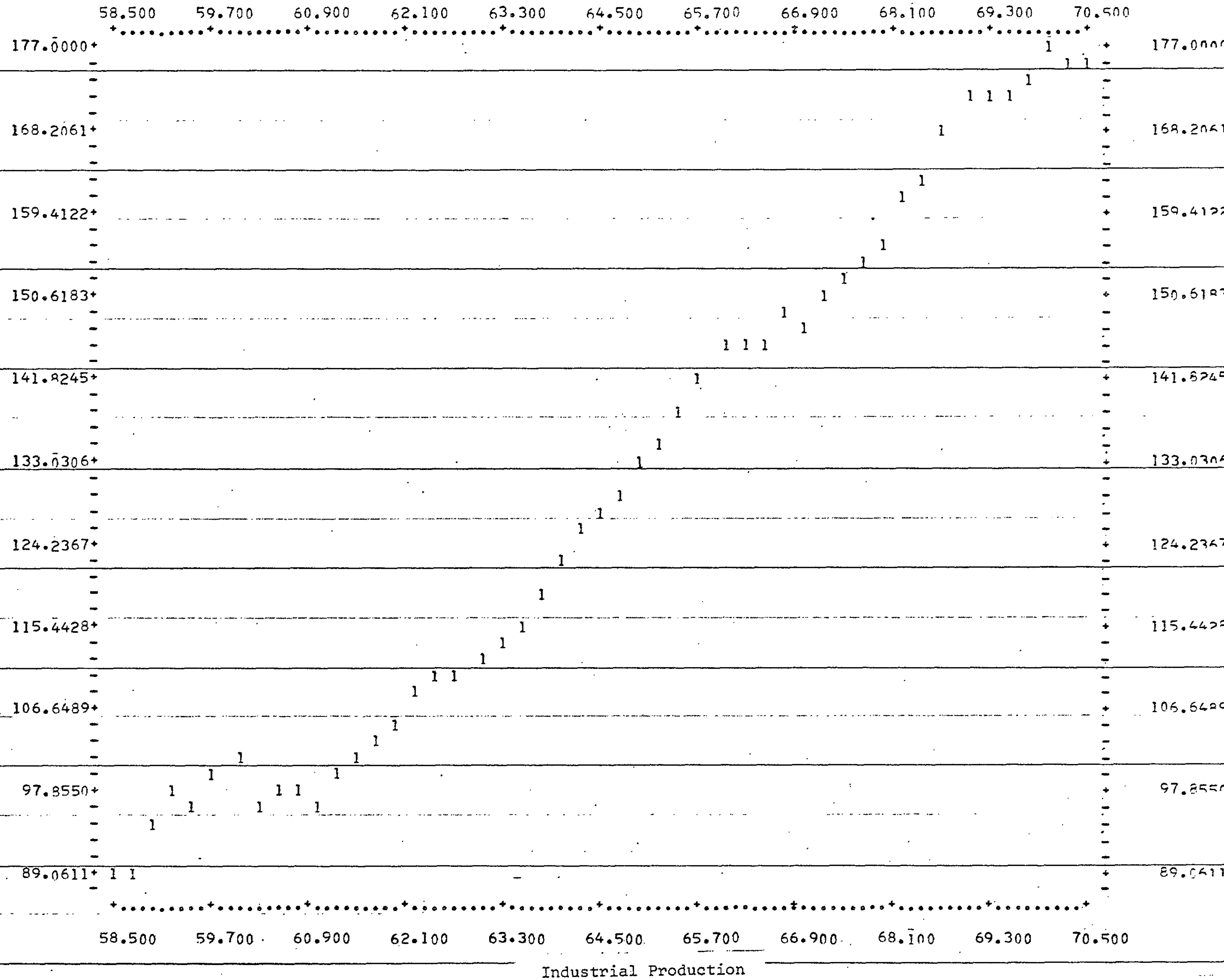

58.500
59.700
60.900
62.100
63.300
64.500
65.700
66.900
68.100
69.300
70.500
35.8000
34.8600
33.9200
32.9800
32.0400
31.1000
30.1600
29.2200
28.2800
27.3400
26.4000
Participation Rate - Females

CHART 8



Unemployment Rate

CHART 9



58.500
59.700
60.900
62.100
63.300
64.500
65.700
66.900
68.100
69.300
70.500
177.0000
168.2061
159.4122
150.6183
141.8245
133.0306
124.2367
115.4428
106.6489
97.8550
89.0611
Industrial Production

CHART 10

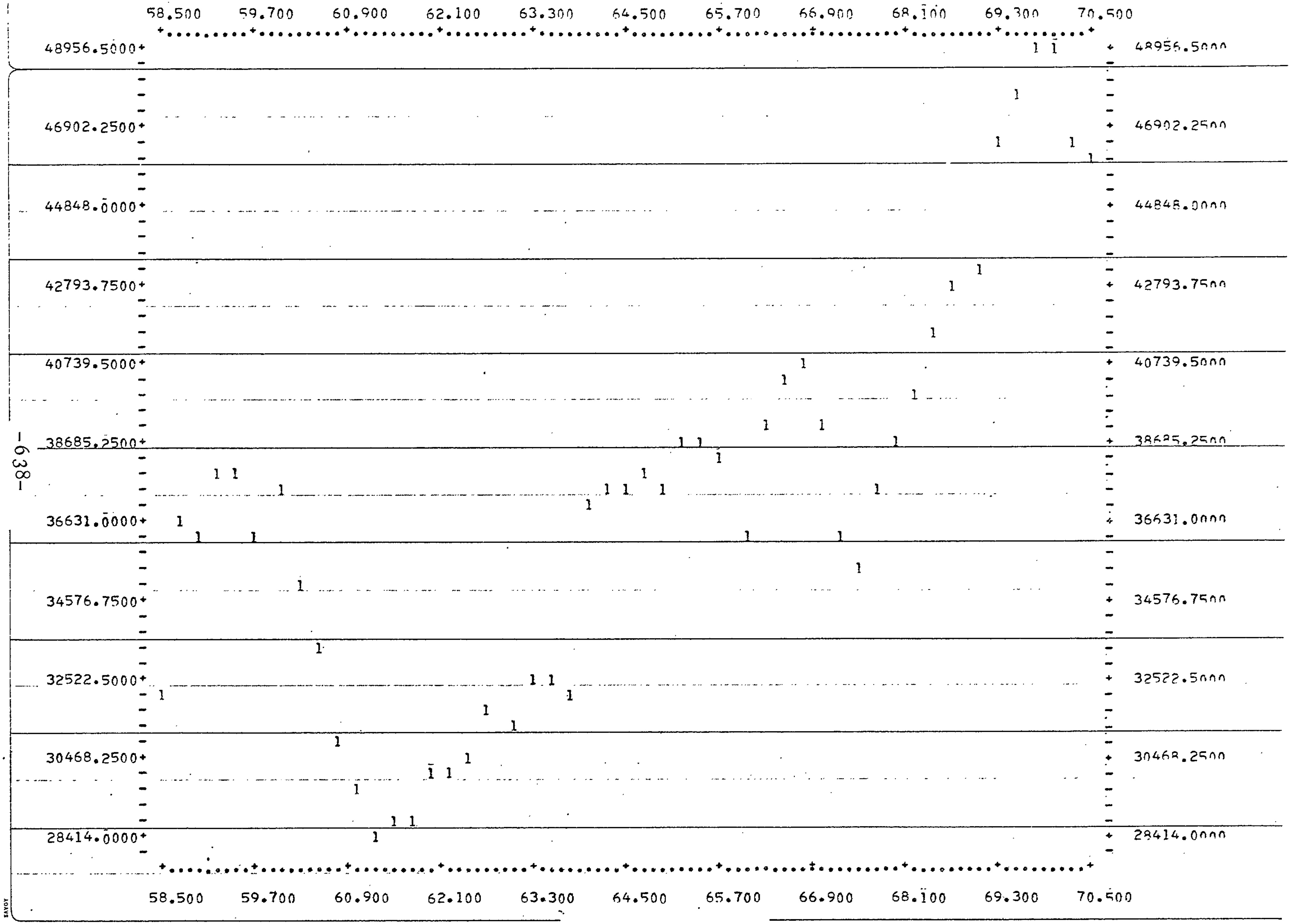
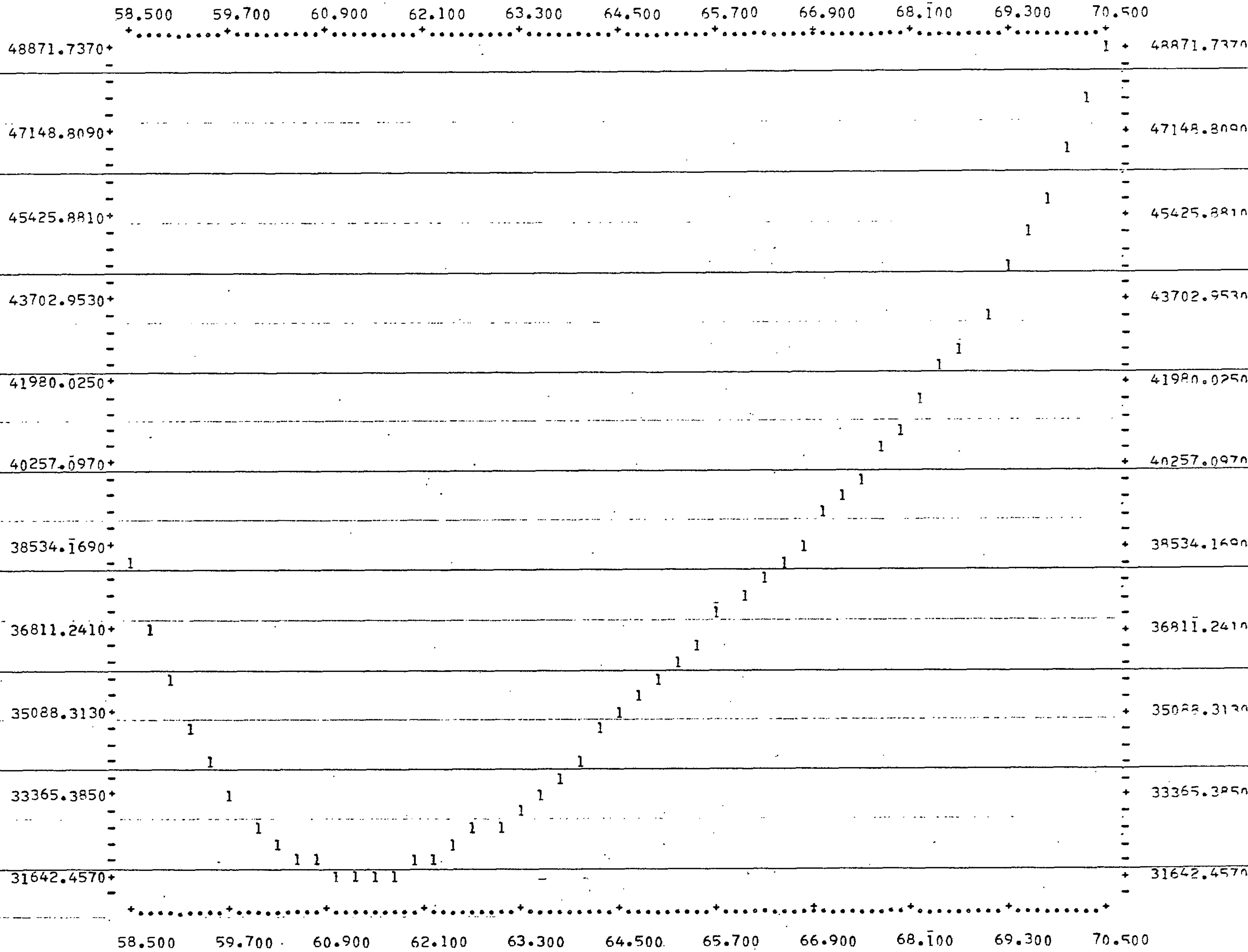
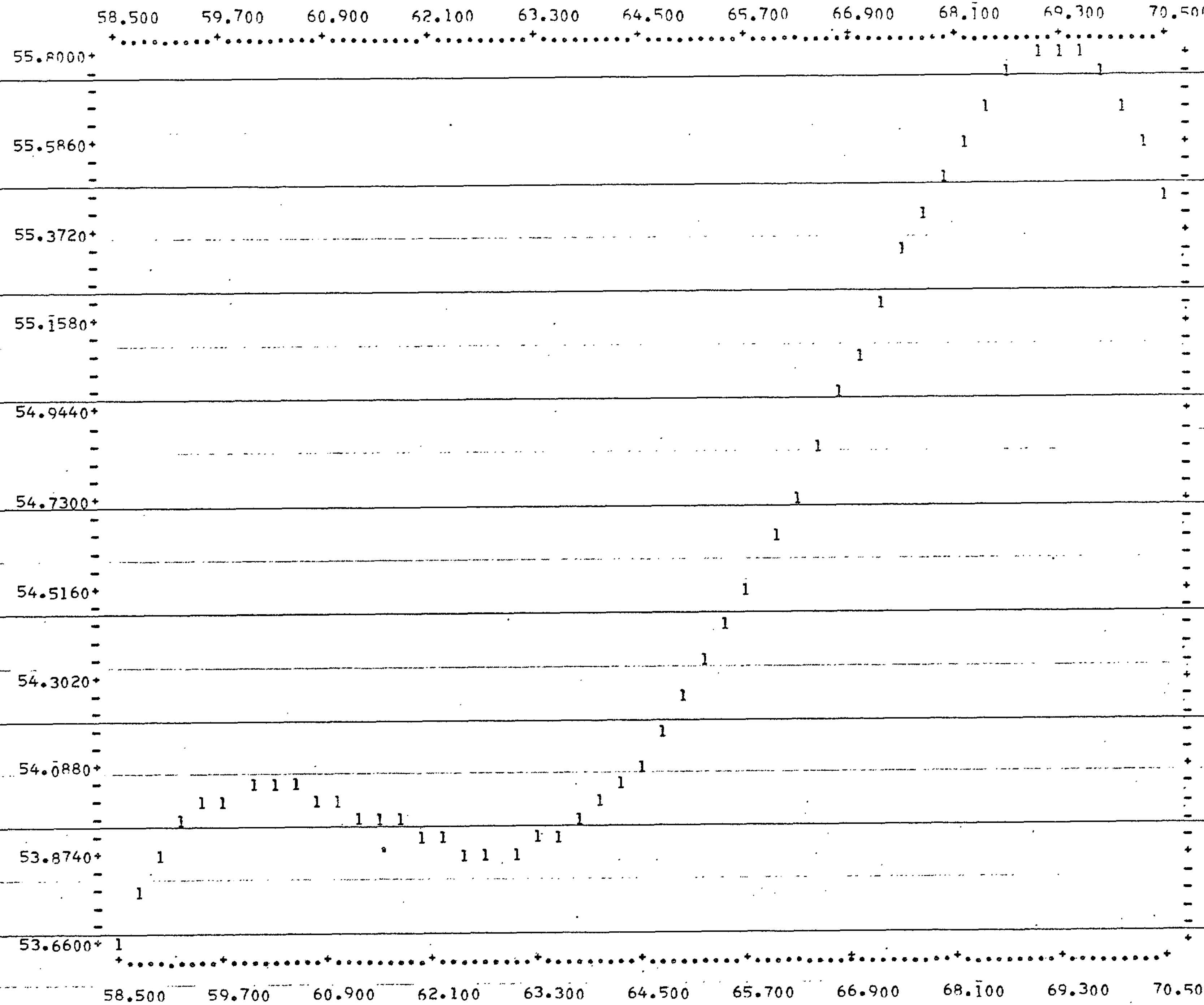

58.500
59.700
60.900
62.100
63.300
64.500
65.700
66.900
68.100
69.300
70.500
48956.5000
46902.2500
44848.0000
42793.7500
40739.5000
38685.2500
36631.0000
34576.7500
32522.5000
30468.2500
28414.0000
Housing Averages

CHART 11



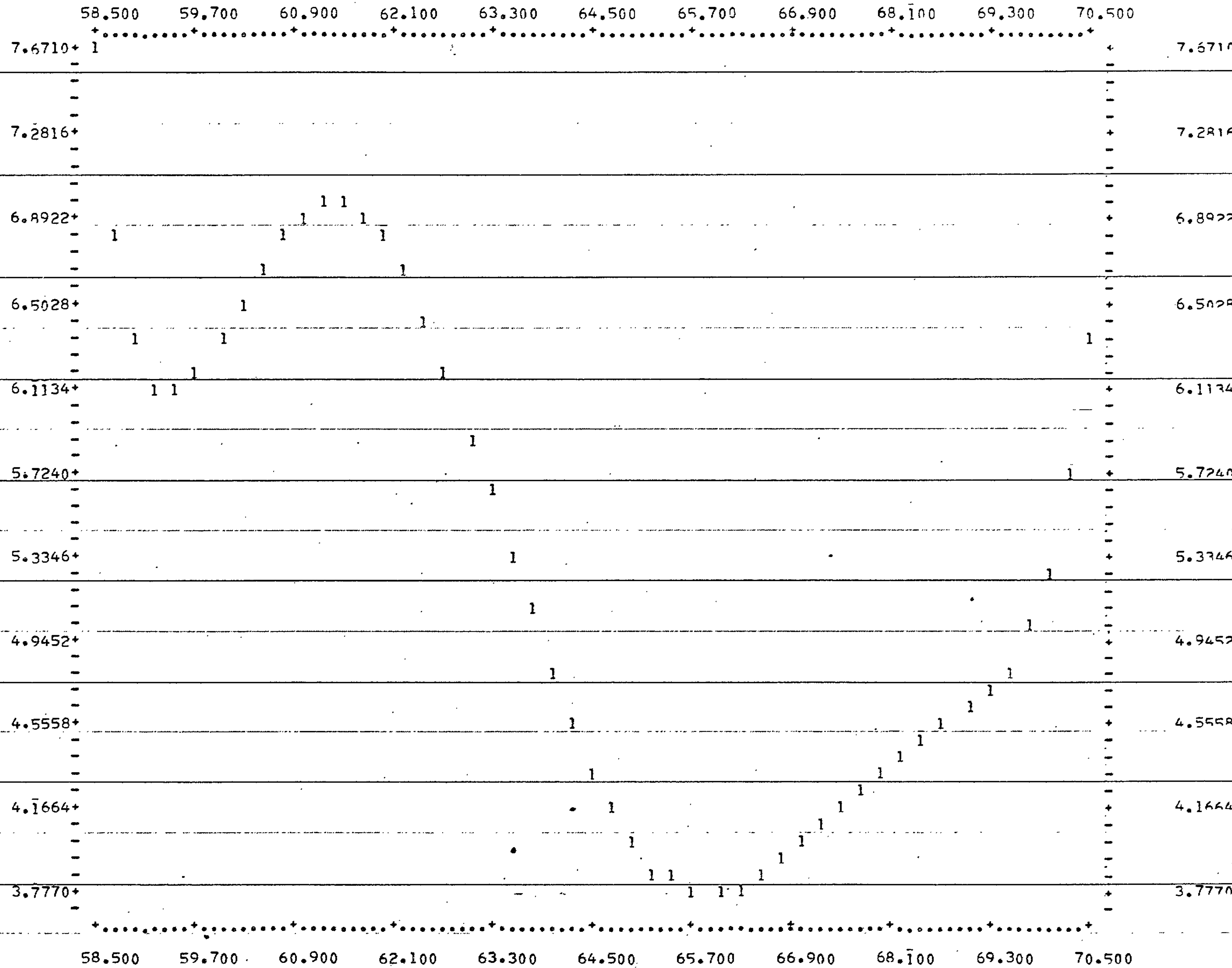
Housing Averages - Smoothed

CHART 12



Participation Rate - Total - Smoothed

CHART 13



Unemployment Rate - Smoothed

4.3 Input-Output Model Identification

The following programs were written for the problem of input-output model identification described in Chapter IV of this report.

The three programs ESTIMATE, ESTIMATEA, and ESTIMATEB were developed using the APL programming language; ESTIMATEA has been implemented in FORTRAN as well.

The program ESTIMATE consists essentially of an implementation of the algorithm described in [19], for parameter identification of minimal, cycli state variable models. Since the relevant algebraic formulae are somewhat involved, and are readily available from the cited references, we do not repeat them here.

The program ESTIMATEA contains the modification to the basic algorithm described in Chapter IV. These modifications allow the program user to supply a dynamical model for an exogenous drift term, in the form of the coefficients $\beta_q,\ldots,\beta_1$ in the characteristic polynomial

$$\lambda^q + \beta_q\lambda^{q-1} + \ldots + \beta_2\lambda + \beta_1$$

of the drift matrix. This information is used to reduce the dimension of the required computation.

The program ESTIMATEB incorporates a computation of the matrix Q (QMAT in the program) used as described in Chapter IV in connection with decoupling problems. It was not felt that the results of computations with real data described in Chapter IV using the FORTRAN version of ESTIMATEA justified the effort involved in a FORTRAN implementation of ESTIMATEB.

```
$JOB    ACCT-NUM,VERNER,TIME=60
C       ***********************************************************************
C
C            APL PROGRAMS : ESTIMATE, ESTIMATEA, ESTIMATEB, AUGMENT
C
C       ***********************************************************************
```

```
    ∇ ESTIMATE;F;G;H;PART
[1]    'STATE DIMENSION'
[2]    N̲←N←⎕
[3]    'INPUT DIMENSION'
[4]    M←⎕
[5]    'OUTPUT DIMENSION'
[6]    P←⎕
[7]    'SELECTOR COEFFICIENTS'
[8]    ALPHA←⎕
[9]    'DATA SEGMENT POINTERS'
[10]   JJ←⎕
[11]   ALPY←(1,ND←ρALPY)ρALPY←+/[1]ALPHA×Y
[12]   ARRAY←(1,N)ρALT←ALPY
[13]   I←0
[14]  L1:→((¯1+ρALPY)>I←I+1+0×1↑ρARRAY←ARRAY,[1](1,N)ρALT←1⌽ALT)/L1
[15]   YPART←((JJ[2]-ND),0)↓(JJ[1],0)↓ARRAY
[16]   UARRAY←(1,M×N)ρUT←U
[17]   I←0
[18]  L2:→((¯1+ρALPY)>I←I+1+0×1↑ρUARRAY←UARRAY,[1](1,M×N)ρUT←M⌽UT)/L2
[19]   UPART←((JJ[2]-ND),0)↓(JJ[1],0)↓UARRAY
[20]   RHS←(JJ[2]-ND)↓JJ[1]↓N̲⌽ALPY[1;]
[21]   AR←,RHS LSQ DATAMAT←YPART,UPART
[22]   A←N̲↑AR
[23]   R←N̲↓AR
[24]   H←(P,N)ρJ←0
[25]  L5:→(P<J←J+1)/L6
[26]   H[J;]←(1,N)ρN↑((JJ[2]-ND)↓JJ[1]↓Y[J;])LSQ DATAMAT
[27]   →L5
[28]  L6:EH←H
[29]   F←(((N̲-1),N̲)ρ0,1,(N̲-1)ρ0),[1]A
[30]   R←(((N̲-1),N̲×M)ρ0),[1]R
[31]   G←(N,M)↑R
[32]   J←1
[33]  L4:→(N>J←J+1+0×ρρG←G+(N,M)↑F+.×R←(0,M)↓R)/L4
[34]   EF←F+0×ρρEG←G
[35]   ⎕L;'H IS   ';H
[36]   ⎕L;'F IS   ';F
[37]   ⎕L;'G IS   ';G
    ∇
```

```
    ∇ ESTIMATEA;F;G;H;YPART;UPART
      'STATE DIMENSION'
      N̲←N←⎕
      'INPUT DIMENSION'
      M←⎕
      'OUTPUT DIMENSION'
      P←⎕
      'DRIFT DYNAMICS POLYNOMIAL'
      P̲←ρBB←,B←⎕
      B←⌽[1](N,1)ρB,Nρ0
      L←⌽[1]⍉((N-P̲),N)ρ1,BB,(N-P̲)ρ0
      'SELECTOR COEFFICIENTS'
      ALPHA←⎕
      'DATA SEGMENT POINTERS'
      JJ←⎕
      ALPY←(1,ND←ρALPY)ρALPY←+/[1]ALPHA×Y
      ARRAY←(1,N)ρALT←ALPY
      I←0
     L1:→((¯1+ρALPY)>I←I+1+0×1↑ρARRAY←ARRAY,[1](1,N)ρALT←1⌽ALT)/L1
      →L8
     L7:JJ←JJ+1
     L8:
      TYPART←YPART←((JJ[2]-ND),0)↓(JJ[1],0)↓ARRAY
      YPART←YPART+.×L
      UARRAY←(1,M×N)ρUT←U
      I←0
     L2:→((¯1+ρALPY)>I←I+1+0×1↑ρUARRAY←UARRAY,[1](1,M×N)ρUT←M⌽UT)/L2
      UPART←-((JJ[2]-ND),0)↓(JJ[1],0)↓UARRAY
      RHS←(JJ[2]-ND)↓JJ[1]↓N̲⌽ALPY[1;]
      RHS←-RHS+TYPART+.×B
      AR←,RHS LSQ DATAMAT←YPART,UPART
      AD←(N̲-P̲)↑AR
      A←-B+L+.×AD
      R←(N̲-P̲)↓AR
      H←(P,N)ρJ←0
     L5:→(P<J←J+1)/L6
      H[J;]←(1,N)ρN↑((JJ[2]-ND)↓JJ[1]↓Y[J;])LSQ DATAMAT←TYPART,UPART
      →L5
     L6:EH←H
      F←(((N̲-1),N̲)ρ0,1,(N̲-1)ρ0),[1],A
      R←(((N̲-1),N̲×M)ρ0),[1]R
      G←(N,M)↑R
      J←1
     L4:→(N>J←J+1+0×ρρG←G+(N,M)↑R←F+.×R←(0,M)↓R)/L4
      EF←F+0×ρρEG←G
      ⎕L;⎕L;⎕L;⎕L;⎕L
      'SEG START IS ';1+JJ[1];'      SEG END IS ';JJ[2];⎕L
      'A IS ';A;⎕L
      'AD IS ';AD
      ⎕L;'H IS  ';H
      ⎕L;'F IS  ';F
      ⎕L;' EIGENVALUES ARE    ';0 EIG F
      ⎕L;'G IS  ';G
      →(STOP>¯2+N+JJ[2])/L7
    ∇
```

```
    ∇ ESTIMATER;F;G;H;YPART;UPART
[1]   'STATE DIMENSION'
[2]   N←N←⎕
[3]   'INPUT DIMENSION'
[4]   M←⎕
[5]   'OUTPUT DIMENSION'
[6]   P←⎕
[7]   'DRIFT DYNAMICS POLYNOMIAL'
[8]   P←ρBB←,B←⎕
[9]   B←⌽[1](N,1)ρB,Nρ0
[10]  L←⌽[1]⍉((Q←N-P),N)ρ1,BB,(N-P)ρ0
[11]  'SELECTOR COEFFICIENTS'
[12]  ALPHA←⎕
[13]  'DATA SEGMENT POINTERS'
[14]  JJ←⎕
[15]  ALPY←(1,ND←ρALPY)ρALPY←+/[1]ALPHA×Y
[16]  ARRAY←(1,N)ρALT←ALPY
[17]  I←0
[18] L1:→((¯1+ρALPY)>I←I+1+0×1↑ρARRAY←ARRAY,[1](1,N)ρALT←1⌽ALT)/L1
[19]  TYPART←YPART←((JJ[2]-ND),0)↓(JJ[1],0)↓ARRAY
[20]  YPART←YPART+.×L
[21]  UARRAY←(1,M×N)ρUT←U
[22]  I←0
[23] L2:→((¯1+ρALPY)>I←I+1+0×1↑ρUARRAY←UARRAY,[1](1,M×N)ρUT←M⌽UT)/L2
[24]  UPART←-((JJ[2]-ND),0)↓(JJ[1],0)↓UARRAY
[25]  RHS←(JJ[2]-ND)↓JJ[1]↓N⌽ALPY[1;]
[26]  RHS←-RHS+TYPART+.×B
[27]  AR←,RHS LSQ DATAMAT←YPART,UPART
[28]  AD←(N-P)↑AR
[29]  AUGMENT
[30]  A←-B+L+.×AD
[31]  R←(N-P)↓AR
[32]  H←(P,N)ρJ←0
[33] L5:→(P<J←J+1)/L6
[34]  H[J;]←(1,N)ρN↑((JJ[2]-ND)↓JJ[1]↓Y[J;])LSQ DATAMAT←TYPART,-UPART
[35]  →L5
[36] L6:EH←H
[37]  F←(((N-1),N)ρ0,1,(N-1)ρ0),[1],A
[38]  R←(((N-1),N×M)ρ0),[1]R
[39]  G←(N,M)↑R
[40]  J←1
[41] L4:→(N>J←J+1+0×ρρG←G+(N,M)↑F+.×R←(0,M)↓R)/L4
[42]  EF←F+0×ρρEG←G
[43]  ⎕L;'H IS    ';H
[44]  ⎕L;'F IS    ';F
[45]  ⎕L;'G IS    ';G
    ∇
```

```
    ∇ AUGMENT
[1]   APOLY←1,AD
[2]   JPOLY←1,BB
[3]   M1←⍉(Q,P+Q)⍴JPOLY,Q⍴0
[4]   M2←⍉(P,P+Q)⍴APOLY,P⍴0
[5]   RS←(((P+Q-1)⍴0),1)⌹M1,M2
[6]   ⍝ CHECK EUCLIDEAN ALGORITHM
[7]   ⎕←(RHO←(RPOLY←Q↑RS)POLYMULT JPOLY)+SIGMA←(SPOLY←(-P)↑RS)POLYMULT APOLY
[8]   SIGMA←SIGMA MOD POLY←JPOLY POLYMULT APOLY
[9]   RHO←RHO MOD POLY
[10]  SN←(1,N)⍴I←0
[11] L9:→(P≥1↑⍴SN←SN,[1](1,N)⍴((⌽⍳P)=I←I+1)POLYMULT SIGMA MOD POLY)/L9
[12]  RN←(1,N)⍴I←0
[13] L99:→(Q≥1↑⍴RN←RN,[1](1,N)⍴((⌽⍳Q)=I←I+1)POLYMULT RHO MOD POLY)/L99
[14]  SN←((-P),¯1)↑QMAT←⌽(1,0)↓RN,[1](1,0)↓SN
[15]  COMPJ←(((P-1),P)⍴0,1,(P-1)⍴0),[1]-⌽BB
[16]  V←TEMP←SN+I←0
[17] L10:→((N-1)>I←I+1+0×⍴⍴V←V,TEMP←COMPJ+.×TEMP)/L10
[18]  E←(⍳M)∘.=(M×N)⍴⍳M
[19]  TEMP←E[I←1;]\V
[20] L11:→((1<⍴⍴E)×M>I+0×⍴⍴TEMP←TEMP,[1]E[I←I+1;]\V)/L11
[21]  V←((P,Q)⍴0),TEMP
[22]  DATAMAT←(YPART,UPART),[1]V
[23]  RHS←(,RHS),(M×P)⍴0
[24]  AR←RHS LSQ DATAMAT
[25]  AD←(N-P)↑AR
    ∇
```

```
$JOB   ACCT-NUM,VERNER,TIME=60
C      ********************************************************************************
C
C            PROGRAM 4.5 :   INOUT
C
C      ********************************************************************************
```

```
3JOB INOUT; CHARGE BERGHOUT; CLASS 3;
USER 080002/VHJ;
PRINTLIMIT=12000;
PROCESSTIME=600;
BEGIN
3COMPILE INOUT FORTRAN;
FILE FILE4(KIND=PACK,TITLE=DATA/XXAV,FILETYPE=7);
DATA
$SET $



3DATA FILE5



3END JOB
```

```
C
C       ###############################################################
C       PROGRAM : 4.5
C       TITLE : INOUT
C       PURPOSE : TO ESTIMATE DYNAMICS
C       DATA : U-ARRAY AND A(T) OR B(T)
C       OUTPUT : DYNAMICS
C
C            A T T R I B U T E S    OF    E S T I M A T E A
C
C            N - STATE VARIABLE DIMENSION
C            M - INPUT VARIABLE DIMENSION
C            P - OUTPUT VARIABLE DIMENSION
C
C
C                 ##############################################
C                 #                                            #
C                 #               N O T E                      #
C                 #                                            #
C                 #    SEG+1      MUST BE GREATER THAN  (M+1)*N #
C                 #                                            #
C                 #   STOP-N+2    MUST BE GREATER THAN    SEG  #
C                 #                                            #
C                 #      N        MUST BE GREATER THAN     D   #
C                 #                                            #
C                 ##############################################
C
C            D - DRIFT DYNAMICS DIMENSION
C            SEGS.SEG - DATA SEGMENT START,LENGTH
C
C       REAL ALPHA(P),ALPY(STOP),B(N),B1(N+D-1),BB(D)
C       REAL YARRAY(SPMNP1,N),YPART(SEG,NMD),TYPART(SEG,N)
C       REAL DATAMX(SEG,(M+1)*N-D),ARX((M+1)*N-D),DATAMT(SEG,(M+1)*N),
C         ART((M+1)*N)
C       REAL U1(MBSTOP),UARRAY(SPMNP1,MBN),UPART(SEG,MBN)
C       REAL L(N,NMD),LT(NMD,N),RHS(SEG),AD(NMD)
C       REAL A(N),R1(MBN),F1(N,N),G1(N,M),H1(P,N)
C       REAL F2(N,N),F3(N,N)
C       REAL IPIV(MP1BN),AUX(2*MP1BN)
C
C       ###############################################################
C
C
C            D I M E N S I O N    E S T I M A T E A
C
        REAL ALPHA(4),ALPY(56),B(11),BT(20),BB(4)
        REAL YARRAY(50,11),YPART(37,9),TYPART(37,11)
        REAL DATAMX(37,33),APX(33),DATAMT(37,35),ART(35)
        REAL U1(200),UARRAY(50,28),UPART(37,28)
        REAL L(11,9),LT(9,11),RHS(37),AD(9)
        REAL A(11),R1(28),F1(11,11),G1(11,4),H1(4,11)
        REAL F2(11,11),F3(11,11)
        INTEGER IPIV(35)
        REAL AUX(70)
        REAL RR(10),RI(10)
C
C            D I M E N S I O N    E S T I M A T E A
C
C       ###############################################################
```

```
C
C                  D A T A
C
C       ISEL(M)   -   (MI3) FORMAT
C           1 => INDUSTRIAL WAGES
C           2 => MANUFACTURING WAGES
C           3 => HOUSING COMPLETIONS
C           4 => PERSONAL INCOME
C           5 => DISPOSABLE INCOME
C           6 => PARTICIPATION RATE TOTAL
C           7 => PARTICIPATION RATE FEMALES
C           8 => UNEMPLOYMENT RATE
C           9 => INDUSTRIAL INDEX
C           10 => HOUSING COMPLETIONS (AVERAGES OF 4 PREVIOUS QUARTERS)
C           11 => HOUSING AVERAGES (SMOOTHED - FOURTH DEGREE POLYNOMIAL)
C           12 => PARTICIPATION RATES (SMOOTHED - FOURTH DEGREE POLY)
C           13 => UNEMPLOYMENT (SMOOTHED - SIXTH DEGREE POLYNOMIAL)
C
C       N,M,P,D,STOP  BY (5I3)
C
C       SHIFT VARIABLE   -   (I3) FORMAT
C
C       ONE OF THE FOLLOWING:
C       1)  A(T,X)    4 ROWS BY 49 COLUMNS IN (5F16.8) FORMAT
C       2)  B(T)     49 ELEMENTS IN (4F20.8) FORMAT
C
C       U-ARRAY     13 ROWS BY 52 COLUMNS
C           ROWS 1 & 2    (4F7.2) FORMAT
C           ROWS 3, 4 & 5    (4I7) FORMAT
C           ROWS 6,7 & 8    (4F5.1) FORMAT
C           ROW 9    (4F6.1) FORMAT
C           ROW 10   (4F9.2) FORMAT
C           ROW 11   (4F10.3) FORMAT
C           ROW 12   (4F6.2) FORMAT
C           ROW 13   (4F6.1) FORMAT
C
C       BB   BY   (DF10.6)
C
C       ALPHA   BY   (PF10.6)
C
C       TITLE   -   (3A4) FORMAT
C
C       ***************************************************************
C
      INTEGER D,NMD,MBN,MBSTOP,SPMNP1,SEG,SEGS,MP1BN,M1BNMD,NPDM1,MP1B2N
      INTEGER N,M,P,Q,R,STOP
      REAL Y(1,49),U(4,49)
      REAL YY(1,49),UU(13,49)
      INTEGER IX,SEED
      DIMENSION ISEL(4)
      DIMENSION SEL(7,13)
      REAL ADSUM(35,9)
      INTEGER SHIFT
      DIMENSION IMONTH(4)
      DIMENSION TITLE(3)
      INTEGER COUNT
      COMMON IX,SEED
      DATA IMONTH/'APR ','JULY','OCT ','JAN '/
      DATA COUNT/0/
      DATA SEL/'INDU','STRI','AL W','AGES','    ','    ','    ',
```

```
     $              'MANU','FACT','URIN','G WA','GES ','  ','  ',
     $              'HOUS','ING ','COMP','LETI','ONS ','  ','  ',
     $              'PERS','ONAL',' INC','OME ','  ','  ','  ',
     $              'DISP','OSAB','LE I','NCOM','E ','  ','  ',
     $              'PART','ICIP','ATIO','N RA','TE T','OTAL','  ',
     $              'PART','ICIP','ATIO','N RA','TE F','EMAL','ES  ',
     $              'UNEM','PLOY','MENT',' RAT','E ','  ','  ',
     $              'INDU','STRI','AL P','RODU','CTIO','N ','  ',
     $              'HOUS','ING ','AVER','AGES','  ','  ','  ',
     $              'HOUS','ING ','AVER','AGES',' - S','MOOT','HED ',
     $              'PART','ICIP','ATIO','N - ','SMOO','THED','DEG4',
     $              'UNEM','PLOY','MENT',' - S','MOOT','HED ','DEG6'/
*EXTENSION*    CC-1
               IX=0
               CALL INPUT(1,49,UU,YY)
               I1=0
               DO 901 I=1,49
               Y(1,I)=YY(1,I)
          901  CONTINUE
            1  READ(5,101)(ISEL(I),I=1,4)
               COUNT=0
               READ(5,104) N,M,P,D,STOP
               READ(5,102) SHIFT
               SEGS = 1
               SEG = (M+1)*N+4
               DO 60 I=1,M
                    J=ISEL(I)
                    DO 60 I1=1,STOP
                         U(I,I1)=UU(J,I1)
           60            CONTINUE
               WRITE(6,249)
               WRITE(6,113) SEGS,SEG
               NMD=N-D
               MBN=M*N
               NPDM1=N+D-1
               SPMNP1=STOP-N+1
               MP1BN=MBN+N
               M1BNMD=MP1BN-D
               MP1B2N=(M+1)*2*N
               MBSTOP=M*STOP
           93  CONTINUE
               IF (COUNT.EQ.0) GO TO 94
               SEGS = SEGS+SHIFT
               IF (SEGS+SEG-1.GT.STOP-N+1) GO TO 90
               WRITE(6,249)
               WRITE(6,113) SEGS,SEG
           94  CONTINUE
               CALL       ESTA(N,M,P,D,STOP,NMD,MBN,SEGS,SEG,NPDM1,SPMNP1,M1BNMD,
              &COUNT,SHIFT,MP1BN,MP1B2N,MBSTOP,RR,RI,
              &F2,F3,       Y,U,ALPHA,ALPY,B,BT,BB,YARRAY,YPART,TYPART,DATAMX,
              &ARX,DATAMT,ART,U1,UARRAY,UPART,L,LT,RHS,AD,A,R1,F1,G1,H1,IPIV,AUX)
               COUNT = COUNT + 1
               DO 95 I = 1,NMD
           95  ADSUM(COUNT,I) = AD(I)
               GO TO 93
           90  CONTINUE
               READ(5,256) TITLE
               WRITE(6,252) TITLE
               WRITE(6,251)
               DO 10 I = 1,M
```

```
      J = ISEL(I)
   10 WRITE(6,103)(SEL(K,J),K=1,7)
      WRITE(6,111) N,M,P,D,STOP,SEG,SHIFT
      SEGS = -SHIFT + 1
      DO 96 I = 1,COUNT
         SEGS = SEGS + SHIFT
         SEGE=SEGS+SEG+N-2
         JY=INT((SEGS+1)/4)+58
         KY=INT((SEGE+1)/4)+58
         JM = MOD(SEGS,4) + 1
         KM = MOD(SEGE,4) + 1
   96 WRITE(6,253) IMONTH(JM),JY,IMONTH(KM),KY,(ADSUM(I,LL),LL=1,NMD)
      READ(5,102) N
      IF(N.EQ.1) GO TO 1
  101 FORMAT(4I3)
  102 FORMAT(I3)
  103 FORMAT('+',T18,7A4,/)
  104 FORMAT(5I3)
  111 FORMAT('0',' N = ',I2,'     M = ',I2,'     P = ',I2,'     D = ',I2,
     & '     STOP = ',I2,'     SEGMENT LENGTH = ',I2,'     SHIFT = ',I2,//)
  113 FORMAT(' ','START OF SEGMENT = ',I2,///,' SEGMENT LENGTH = ',I2)
  249 FORMAT('1',1X,////,' ESTIMATION',//)
  251 FORMAT('0','FACTORS USED :')
  252 FORMAT('1','SUMMARY OF AD: ',10X,3A4,///)
  253 FORMAT(' ',A4,1X,I2,' TO ',A4,1X,I2,5X,9F11.6)
  256 FORMAT(3A4)
      STOP
      END

      SUBROUTINE INPUT(P,STOP,U,Y)
C     ****************************************************************
C     THIS ROUTINE READS (1) EITHER A(T) OR B(T) AND (2) THE U-ARRAY.
C     ****************************************************************
      INTEGER P,STOP
      REAL U(13,STOP),Y(1,STOP)
      REAL TEMP(52)
      DIMENSION VAR(13),VF(3)
      DIMENSION ISEL(4)
      INTEGER CODE,LL
      DATA VAR/'  F7.2','  F7.2',',   I7',',   I7',',   I7',',  F5.1',
     & '  F5.1',',  F5.1',',  F6.1',',  F9.2',', F10.3',',  F6.2',
     & '  F6.1'/
      DATA VF/'   (4','   ',')     '/
      DATA CODE/0/
C     READREADREADREADREADREADREADREADREADREADREADREADREADREADREADREAD
C
C     READ A(T,X)
C
      DO 10 I = 1,P
   10 READ(5,200)(Y(I,K),K=1,STOP)
      LL=INT((50-STOP)/5)
      IF (LL.EQ.0) GO TO 100
      DO 101 J=1,LL
  101 READ(5,200) X
  100 CONTINUE
      MM=4-P
      IF (MM.EQ.0) GO TO 90
      DO 80 I = 1,MM
      DO 80 K=1,10
   80 READ(5,200)X1,X2,X3,X4,X5
```

```
   90 CONTINUE
  200 FORMAT(5F16.8)
C
C     READREADREADREADREADREADREADREADREADREADREADREADREADREADREADREAD
C     ****************************************************************
C     TO READ B(T) REMOVE THE READ BLOCK ABOVE, AND REMOVE THE C
C     FROM THE FOLLOWING CARDS:
C     READ(5,200)(Y(1,I),I=1,STOP)
C     LL=INT((52-STOP)/4)
C     IF (LL.EQ.0) GO TO 100
C     DO 101 J=1,LL
C 101 READ(5,200) X
C 100 CONTINUE
C 200 FORMAT(4F20.8)
C     ****************************************************************
      DO 20 I=1,13
      VF(2) = VAR(I)
      READ(5,VF)X1,X2,(U(I,K),K=1,STOP)
      L=INT((50-STOP)/4)
      IF (L.EQ.0) GO TO 70
      DO 40 J=1,L
   40 READ(5,VF)X
   70 IF (I.NE.9) GO TO 20
      X = 101.6 / 173.4
      DO 50 J = 1,10
   50 U(LL,J) =U(LL,J) * X
   20 CONTINUE
      RETURN
      END

      SUBROUTINE ESTA(N,M,P,D,STOP,NMD,MBN,SEGS,SEG,NPDM1,SPMNP1,M1BNMD,
     &COUNT,SHIFT,MP1BN,MP1B2N,MBSTOP,RR,RI,
     &F2,F3,          Y,U,ALPHA,ALPY,B,B1,BB,YARRAY,YPART,TYPART,DATAMX,
     &ARX,DATAMT,ART,U1,UARRAY,JPART,L,LT,RHS,AD,A,R,F,G,H,IPIV,AUX)
      INTEGER N,M,P,D,STOP,NMD,MBN,SEGS,SEG,NPDM1,SPMNP1,M1BNMD
      INTEGER MP1BN,MP1B2N,MBSTOP
      REAL Y(1,STOP),U(4,STOP)
      REAL ALPHA(P),ALPY(STOP),B(N),B1(NPDM1),BB(D)
      REAL YARRAY(SPMNP1,N),YPART(SEG,NMD),TYPART(SEG,N)
      REAL DATAMX(SEG,M1BNMD),ARX(M1BNMD),DATAMT(SEG,MP1BN),ART(MP1BN)
      REAL U1(MBSTOP),UARRAY(SPMNP1,MBN),UPART(SEG,MBN)
      REAL L(N,NMD),LT(NMD,N),RHS(SEG),AD(NMD)
      REAL A(N),R(MBN),F(N,N),G(N,M),H(P,N)
      REAL F2(N,N),F3(N,N)
      INTEGER IPIV(MP1BN)
      REAL AUX(MP1B2N)
      REAL RR(N),RI(N)
      INTEGER SHIFT,COUNT,SEGE
      DIMENSION NVAR(10),VF(3),VF1(3)
      DATA NVAR/'1','2','3','4','5','6','7','8','9','10'/
      DATA VF/'(1X  ,','        ','F13.8)'/
      DATA VF1/'(      ','      ','F10.6)'/
      IF (COUNT.NE.0) GO TO 10
C
C     READREADREADREADREADREADREADREADREADREADREADREADREADREADREADREAD
      VF1(2)=NVAR(D)
      READ(5,VF1) (BB(I),I=1,D)
      VF1(2)=NVAR(P)
      READ(5,VF1) (ALPHA(I),I=1,P)
C     READREADREADREADREADREADREADREADREADREADREADREADREADREADREADREAD
```

```
   10 CONTINUE
C
      DO 31 I=1,NMD
   31      B(I)=0.
      DO 32 I=1,D
           J=N+1-I
   32      B(J)=BB(I)
      B1(N)=1.
      DO 25 I=1,D
           J=N-I
           B1(J)=BB(I)
   25      CONTINUE
      NMDM1=NMD-1
      IF (NMDM1.EQ.0) GO TO 27
      DO 26 I=1,NMDM1
           J=N+I
           B1(I)=0.
           B1(J)=0.
   26      CONTINUE
   27 CONTINUE
      NPMM1=N+NMD-1
      CALL SHFTMX(NPMM1,B1,NMD,N,1,LT)
      DO 34 I=1,NMD
           DO 34 J=1,N
   34      L(J,I)=LT(I,J)
      DO 36 J=1,STOP
           SUM=0.
           DO 35 I=1,P
   35           SUM=SUM+ALPHA(I)*Y(I,J)
   36      ALPY(J)=SUM
      CALL SHFTMX(STOP,ALPY,SPMNP1,N,1,YARRAY)
      SEGE=SEGS+SEG-1
      I=0
      DO 41 K=SEGS,SEGE
      I=I+1
           DO 41 J=1,N
   41           TYPART(I,J)=YARRAY(K,J)
      CALL MULTM(SEG,N,NMD,TYPART,L,YPART)
      DO 43 I=1,STOP
           DO 43 J=1,M
                K=M*(I-1)+J
   43           U1(K)=U(J,I)
      CALL SHFTMX(MBSTOP,U1,SPMNP1,MBN,M,UARRAY)
      I=0
      DO 45 K=SEGS,SEGE
           I=I+1
           DO 45 J=1,MBN
                UPART(I,J)=-UARRAY(K,J)
   45           CONTINUE
      CALL MULT(SEG,N,   TYPART,B,RHS)
      I=0
      DO 46 K=SEGS,SEGE
           I=I+1
      RHS(I)=-(RHS(I)+ALPY(K+N))
   46      CONTINUE
      I=0
      DO 49 K=SEGS,SEGE
           I=I+1
           DO 47 J=1,NMD
   47           DATAMX(I,J)=YPART(I,J)
```

```
          DO 48 J=1,MBN
48              DATAMX(I,J+NMD)=UPART(I,J)
      CALL LLSQ(DATAMX,RHS,SEG,M1BNMD,1,ARX,IPIV,1.E-10 ,IER,AUX)
      WRITE(6,245) IER
245 FORMAT(////,' IFLAG = ',I2,' AND SHOULD BE  0  IF LEAST SQUARES',
   &' IS SUCCESSFUL',////)
C     CHECK BEGINS
      DO 49 I=1,NMD
          AD(I)=ARX(I)
49        CONTINUE
      CALL MULT(N,NMD,L,AD,A)
      DO 50 I=1,N
50        A(I)=-(B(I)+A(I))
      WRITE(6,270)
      VF(2)=NVAR(NMD)
      WRITE(6,VF) (AD(I),I=1,NMD)
      WRITE(6,271)
      VF(2)=NVAR(N)
      WRITE(6,VF) (A(I),I=1,N)
      I1=SEGS+NMD
      K=NMD
      DO 52 I=1,MBN
          K=K+1
          R(I)=ARX(K)
52        CONTINUE
      WRITE(6,272)
      VF(2)=NVAR(M)
      WRITE(6,VF) (R(I),I=1,MBN)
      DO 59 JJ=1,P
          I=0
          DO 58 K=SEGS,SEGE
              I=I+1
              DO 56 J=1,N
56                DATAMT(I,J)=TYPART(I,J)
              DO 57 J=1,MBN
57                DATAMT(I,J+N)=UPART(I,J)
      RHS(I)=Y(JJ,K)
58            CONTINUE
      CALL LLSQ(DATAMT,RHS,SEG,MP1BN,1,ART,IPIV,1.E-10 ,IER,AUX)
          DO 59 I=1,N
59            H(JJ,I)=ART(I)
      NM1=N-1
      DO 62 J=1,N
          DO 61 I=1,NM1
              F(I,J)=0.
61            IF (I.EQ.J-1) F(I,J)=1.
62        F(N,J)=A(J)
      DO 64 I=1,N
          DO 63 J=1,N
          F3(I,J)=0.
63            IF (I.EQ.J) F3(I,J)=1.
          DO 64 J=1,M
64            G(I,J)=0.
      DO 69 KK=1,N
          DO 65 J=1,M
          K1=(KK-1)*M+J
              DO 68 I=1,N
                  G(I,J)=G(I,J)+F3(I,N)*R(K1)
68                CONTINUE
65                CONTINUE
```

```
      CALL MULTM(N,N,N,F,F3,F2)
            DO 67 I=1,N
            DO 67 J=1,N
   67             F3(I,J)=F2(I,J)
   69       CONTINUE
      WRITE(6,121)
      VF(2)=NVAR(N)
      WRITE(6,VF)  ((F(I,J),J=1,N),I=1,N)
      WRITE(6,261)
      VF(2)=NVAR(N)
C     PUT F IN UPPER HESSENBERG FORM
      DO 77 I=1,N
            DO 77 J=1,N
                  F2(I,J)=F(J,I)
   77             CONTINUE
      CALL ATEIG(N,F2,RR,RI,IPIV,N)
      WRITE(6,VF)  (RR(I),I=1,N)
      WRITE(6,VF)  (RI(I),I=1,N)
      WRITE(6,122)
      VF(2)=NVAR(M)
      WRITE(6,VF)  ((G(I,J),J=1,M),I=1,N)
      WRITE(6,123)
      VF(2)=NVAR(N)
      WRITE(6,VF)  ((H(I,J),J=1,N),I=1,P)
  121 FORMAT(' ','F IS')
  122 FORMAT(' ','G IS')
  123 FORMAT(' ','H IS')
  261 FORMAT(///,' EIGENVALUES ARE',//)
  270 FORMAT(' ','AD IS')
  271 FORMAT(' ','A IS')
  272 FORMAT(' ','R IS')
      RETURN
      END

      SUBROUTINE SHFTMX(N,B,P,M,SHFT,A)
      INTEGER N,M,P,SHFT
      REAL B(N),A(P,M)
      DO 10 I=1,P
            DO 10 J=1,M
                  A(I,J)=B(SHFT*(I-1)+J)
   10             CONTINUE
      RETURN
      END

      SUBROUTINE MULT(IA,JA,A,B,PROD)
      INTEGER IA,JA
      REAL A(IA,JA),B(JA),PROD(IA)
      DO 1 I=1,IA
            SUM=0.
            DO 2 J=1,JA
    2             SUM=SUM+A(I,J)*B(J)
            PROD(I)=SUM
    1       CONTINUE
      RETURN
      END

      SUBROUTINE MULTM(M,N,P,A,B,C)
      INTEGER M,N,P
      REAL A(M,N),B(N,P),C(M,P)
      DO 10 I=1,M
```

```
            DO 10 K=1,P
               SUM=0
               DO 9 J=1,N
    9             SUM=SUM+A(I,J)*B(J,K)
   10          C(I,K)=SUM
      RETURN
      END

      SUBROUTINE LLSQ(A,B,M,N,L,X,IPIV,EPS,IER,AUX)
C
      DIMENSION A(1),B(1),X(1),IPIV(1),AUX(1)
C
C     ERROR TEST
      IF(M-N)30,1,1
C
C     GENERATION OF INITIAL VECTOR S(K) (K=1,2,....N) IN STORAGE
C     LOCATIONS AUX(K) (K=1,2,...,N)
    1 PIV=0.
      IEND=0
      DO 4 K=1,N
      IPIV(K)=K
      H=0.
      IST=IEND+1
      IEND=IEND+M
      DO 2 I=IST,IEND
    2 H=H+A(I)*A(I)
      AUX(K)=H
      IF(H-PIV)4,4,3
    3 PIV=H
      KPIV=K
    4 CONTINUE
C
C     ERROR TEST
      IF(PIV)31,31,5
C
C     DEFINE TOLERANCE FOR CHECKING RANK OF A
    5 SIG=SQRT(PIV)
      TOL=SIG*ABS(EPS)
C
C
C     DECOMPOSITION LOOP
      LM=L*M
      IST=-M
      DO 21 K=1,N
      IST=IST+M+1
      IEND=IST+M-K
      I=KPIV-K
      IF(I)8,8,6
C
C     INTERCHANGE K-TH COLUMN OF A WITH KPIV-TH IN CASE KPIV.GT.K
    6 H=AUX(K)
      AUX(K)=AUX(KPIV)
      AUX(KPIV)=H
      ID=I*M
      DO 7 I=IST,IEND
      J=I+ID
      H=A(I)
      A(I)=A(J)
    7 A(J)=H
C
```

```
C        COMPUTATION OF PARAMETER SIG
       8 IF(K-1)11,11,9
       9 SIG=0.
         DO 10 I=IST,IEND
      10 SIG=SIG+A(I)*A(I)
         SIG=SQRT(SIG)
C
C        TEST ON SINGULARITY
         IF(SIG-TOL)32,32,11
C
C        GENERATE CORRECT SIGN OF PARAMETER SIG
      11 H=A(IST)
         IF(H)12,13,13
      12 SIG=-SIG
C
C        SAVE INTERCHANGE INFORMATION
C
C        GENERATION OF VECTOR UK IN K-TH COLUMN OF MATRIX A AND OF
C        PARAMETER BETA
      13 IPIV(KPIV)=IPIV(K)
         IPIV(K)=KPIV
         BETA=H+SIG
         A(IST)=BETA
         BETA=1./(SIG*BETA)
         J=N+K
         AUX(J)=-SIG
         IF(K-N)14,19,19
C
C        TRANSFORMATION OF MATRIX A
      14 PIV=0.
         ID=0
         JST=K+1
         KPIV=JST
         DO 18 J=JST,N
         ID=ID+M
         H=0.
         DO 15 I=IST,IEND
         II=I+ID
      15 H=H+A(I)*A(II)
         H=BETA*H
         DO 16 I=IST,IEND
         II=I+ID
      16 A(II)=A(II)-A(I)*H
C
C        UPDATING OF ELEMENT S(J) STORED IN LOCATION AUX(J)
         II=IST+ID
         H=AUX(J)-A(II)*A(II)
         AUX(J)=H
         IF(H-PIV)18,18,17
      17 PIV=H
         KPIV=J
      18 CONTINUE
C
C        TRANSFORMATION OF RIGHT HAND SIDE MATRIX B
      19 DO 21 J=K,LM,M
         H=0.
         IEND=J+M-K
         II=IST
         DO 20 I=J,IEND
         H=H+A(II)*B(I)
```

```
   20 II=II+1
      H=BETA*H
      II=IST
      DO 21 I=J,IEND
      B(I)=B(I)-A(II)*H
   21 II=II+1
C     END OF DECOMPOSITION LOOP
C
C
C     BACK SUBSTITUTION AND BACK INTERCHANGE
      IER=0
      I=N
      LN=L*N
      PIV=1./AUX(2*N)
      DO 22 K=N,LN,N
      X(K)=PIV*B(I)
   22 I=I+M
      IF(N-1)26,26,23
   23 JST=(N-1)*M+N
      DO 25 J=2,N
      JST=JST-M-1
      K=N+N+1-J
      PIV=1./AUX(K)
      KST=K-N
      ID=IPIV(KST)-KST
      IST=2-J
      DO 25 K=1,L
      H=B(KST)
      IST=IST+N
      IEND=IST+J-2
      II=JST
      DO 24 I=IST,IEND
      II=II+M
   24 H=H-A(II)*X(I)
      I=IST-1
      II=I+ID
      X(I)=X(II)
      X(II)=PIV*H
   25 KST=KST+M
C
C
C     COMPUTATION OF LEAST SQUARES
   26 IST=N+1
      IEND=0
      DO 29 J=1,L
      IEND=IEND+M
      H=0.
      IF(M-N)29,29,27
   27 DO 28 I=IST,IEND
   28 H=H+B(I)*B(I)
      IST=IST+M
   29 AUX(J)=H
      RETURN
C
C     ERROR RETURN IN CASE M LESS THAN N
   30 IER=-2
      RETURN
C
C     ERROR RETURN IN CASE OF ZERO-MATRIX A
   31 IER=-1
```

```
      RETURN
C
C     ERROR RETURN IN CASE OF RANK OF MATRIX A LESS THAN N
   32 IER=K-1
      RETURN
      END

      SUBROUTINE ATEIG(M,A,RR,RI,IANA,IA)
      DIMENSION A(1),RR(1),RI(1),PRR(2),PRI(2),IANA(1)
      INTEGER P,P1,Q
C
      E7=1.0E-8
      E6=1.0E-6
      E10=1.0E-10
      DELTA=0.5
      MAXIT=30
C
C        INITIALIZATION
C
      N=M
   20 N1=N-1
      IN=N1*IA
      NN=IN+N
      IF(N1) 30,1300,30
   30 NP=N+1
C
C        ITERATION COUNTER
C
      IT=0
C
C        ROOTS OF THE 2ND ORDER MAIN SUBMATRIX AT THE PREVIOUS
C        ITERATION
C
      DO 40 I=1,2
      PRR(I)=0.0
   40 PRI(I)=0.0
C
C        LAST TWO SUBDIAGONAL ELEMENTS AT THE PREVIOUS ITERATION
C
      PAN=0.0
      PAN1=0.0
C
C        ORIGIN SHIFT
C
      R=0.0
      S=0.0
C
C        ROOTS OF THE LOWER MAIN 2 BY 2 SUBMATRIX
C
      N2=N1-1
      IN1=IN-IA
      NN1=IN1+N
      N1N=IN+N1
      N1N1=IN1+N1
   60 T=A(N1N1)-A(NN)
      U=T*T
      V=4.0*A(N1N)*A(NN1)
      IF(ABS(V)-U*E7) 100,100,65
   65 T=U+V
      IF(ABS(T)-AMAX1(U,ABS(V))*E6) 67,67,68
```

```
   67 T=0.0
   68 U=(A(N1N1)+A(NN))/2.0
      V=SQRT(ABS(T))/2.0
      IF(T)140,70,70
   70 IF(U) 80,75,75
   75 RR(N1)=U+V
      RR(N)=U-V
      GO TO 130
   80 RR(N1)=U-V
      RR(N)=U+V
      GO TO 130
  100 IF(T)120,110,110
  110 RR(N1)=A(N1N1)
      RR(N)=A(NN)
      GO TO 130
  120 RR(N1)=A(NN)
      RR(N)=A(N1N1)
  130 RI(N)=0.0
      RI(N1)=0.0
      GO TO 160
  140 RR(N1)=U
      RR(N)=U
      RI(N1)=V
      RI(N)=-V
  160 IF(N2)1280,1280,180
C
C         TESTS OF CONVERGENCE
C
  180 N1N2=N1N1-IA
      RMOD=RR(N1)*RR(N1)+RI(N1)*RI(N1)
      EPS=E10*SQRT(RMOD)
      IF(ABS(A(N1N2))-EPS)1280,1280,240
  240 IF(ABS(A(NN1))-E10*ABS(A(NN))) 1300,1300,250
  250 IF(ABS(PAN1-A(N1N2))-ABS(A(N1N2))*E6) 1240,1240,260
  260 IF(ABS(PAN-A(NN1))-ABS(A(NN1))*E6)1240,1240,300
  300 IF(IT-MAXIT) 320,1240,1240
C
C         COMPUTE THE SHIFT
C
  320 J=1
      DO 360 I=1,2
      K=NP-I
      IF(ABS(RR(K)-PRR(I))+ABS(RI(K)-PRI(I))-DELTA*(ABS(RR(K))
     1    +ABS(RI(K)))) 340,360,360
  340 J=J+I
  360 CONTINUE
      GO TO (440,460,460,480),J
  440 R=0.0
      S=0.0
      GO TO 500
  460 J=N+2-J
      R=RR(J)*RR(J)
      S=RR(J)+RR(J)
      GO TO 500
  480 R=RR(N)*RR(N1)-RI(N)*RI(N1)
      S=RR(N)+RR(N1)
C
C         SAVE THE LAST TWO SUBDIAGONAL TERMS AND THE ROOTS OF THE
C         SUBMATRIX BEFORE ITERATION
C
```

```
  500 PAN=A(NN1)
      PAN1=A(N1N2)
      DO 520 I=1,2
      K=NP-I
      PRR(I)=RR(K)
  520 PRI(I)=RI(K)
C
C          SEARCH FOR A PARTITION OF THE MATRIX, DEFINED BY P AND Q
C
      P=N2
      IF(N-3)600,600,525
525   IPI=N1N2
      DO 580 J=2,N2
      IPI=IPI-IA-1
      IF(ABS(A(IPI))-EPS) 600,600,530
  530 IPIP=IPI+IA
      IPIP2=IPIP+IA
      D=A(IPIP)*(A(IPIP)-S)+A(IPIP2)*A(IPIP+1)+R
      IF(D)540,560,540
  540 IF(ABS(A(IPI)*A(IPIP+1))*(ABS(A(IPIP)+A(IPIP2+1)-S)+ABS(A(IPIP2
     1 ))  -ABS(D)*EPS) 620,620,560
  560 P=N1-J
  580 CONTINUE
  600 Q=P
      GO TO 680
  620 P1=P-1
      Q=P
      DO 660 I=2,P1
      IPI=IPI-IA-1
C     ***************************************************************
C
      IF  (IPI.LE.0)  GO TO 680
C
C     THIS CARD WAS INSERTED TO MAKE PROGRAM RUN -
C
C     WHILE NO ANALYSIS HAS BEEN DONE TO DETERMINE IF THIS MODIFICATI
C     IS THE APPPOPRIATE ONE, THE EIGENVALUES OBTAINED SEEM REASONABL
C
C     ***************************************************************
      IF(ABS(A(IPI))-EPS)680,680,660
  660 Q=Q-1
C
C          OR DOUBLE ITERATION
C
  680 II=(P-1)*IA+P
      DO 1220 I=P,N1
      II1=II-IA
      IIP=II+IA

      IF(I-P)720,700,720
  700 IPI=II+1
      IPIP=IIP+1
C
C          INITIALIZATION OF THE TRANSFORMATION
C
      G1=A(II)*(A(II)-S)+A(IIP)*A(IPI)+R
      G2=A(IPI)*(A(IPIP)+A(II)-S)
      G3=A(IPI)*A(IPIP+1)
      A(IPI+1)=0.0
      GO TO 780
```

```
  720 G1=A(II1)
      G2=A(II1+1)
      IF(I-N2)740,740,760
  740 G3=A(II1+2)
      GO TO 780
  760 G3=0.0
  780 CAP=SQRT(G1*G1+G2*G2+G3*G3)
      IF(CAP)800,860,800
  800 IF(G1)820,840,840
  820 CAP=-CAP
  840 T=G1+CAP
      PSI1=G2/T
      PSI2=G3/T
      ALPHA=2.0/(1.0+PSI1*PSI1+PSI2*PSI2)
      GO TO 880
  860 ALPHA=2.0
      PSI1=0.0
      PSI2=0.0
  880 IF(I-Q)900,960,900
  900 IF(I-P)920,940,920
  920 A(III)=-CAP
      GO TO 960
  940 A(III)=-A(II1)
C
C          ROW OPERATION
C
  960 IJ=II
      DO 1040 J=I,N
      T=PSI1*A(IJ+1)
      IF(I-N1)980,1000,1000
  980 IP2J=IJ+2
      T=T+PSI2*A(IP2J)
 1000 ETA=ALPHA*(T+A(IJ))
      A(IJ)=A(IJ)-ETA
      A(IJ+1)=A(IJ+1)-PSI1*ETA
      IF(I-N1)1020,1040,1040
 1020 A(IP2J)=A(IP2J)-PSI2*ETA
 1040 IJ=IJ+IA
C
C          COLUMN OPERATION
C
      IF(I-N1)1080,1060,1060
 1060 K=N
      GO TO 1100
 1080 K=I+2
 1100 IP=IIP-I
      DO 1180 J=Q,K
      JIP=IP+J
      JI=JIP-IA
      T=PSI1*A(JIP)
      IF(I-N1)1120,1140,1140
 1120 JIP2=JIP+IA
      T=T+PSI2*A(JIP2)
 1140 ETA=ALPHA*(T+A(JI))
      A(JI)=A(JI)-ETA
      A(JIP)=A(JIP)-ETA*PSI1
      IF(I-N1)1160,1180,1180
 1160 A(JIP2)=A(JIP2)-ETA*PSI2
 1180 CONTINUE
      IF(I-N2)1200,1220,1220
```

```
      1200 JI=II+3
           JIP=JI+IA
           JIP2=JIP+IA
           FTA=ALPHA*PSI2*A(JIP2)
           A(JI)=-FTA
           A(JIP)=-FTA*PSI1
           A(JIP2)=A(JIP2)-FTA*PSI2
      1220 II=IIP+1
           IT=IT+1
           GO TO 60
C
C              END OF ITERATION
C
      1240 IF(ABS(A(NN1))-ABS(A(N1N2))) 1300,1280,1280
C
C              TWO EIGENVALUES HAVE BEEN FOUND
C
      1280 IANA(N)=0
           IANA(N1)=2
           N=N2
           IF(N2)1400,1400,20
C
C              ONE EIGENVALUE HAS BEEN FOUND
C
      1300 RR(N)=A(NN)
           RI(N)=0.0
           IANA(N)=1
           IF(N1)1400,1400,1320
      1320 N=N1
           GO TO 20
      1400 RETURN
           END
```

```
*****************************************************************

DATA FOR
     PROGRAM 4.5 :   INOUT

*****************************************************************
 -0.26758019   -0.27968979   -0.27733178   -0.26452390   -0.24528391
 -0.22331243   -0.20104147   -0.18058595   -0.16406074   -0.15326360
 -0.14872370   -0.15065306   -0.15926373   -0.17298921   -0.18314899
 -0.17928405   -0.15093535   -0.09235999   -0.01667949    0.05826847
  0.11464625    0.13896615    0.13514029    0.11143077    0.07609967
  0.03526234   -0.01355276   -0.07496391   -0.15358941   -0.25198907
 -0.36448888   -0.48335636   -0.60085903   -0.70824066   -0.79264999
 -0.84021198   -0.83705162   -0.77606159   -0.67720542   -0.56721436
 -0.47281965   -0.41595746   -0.39938367   -0.42105907   -0.47894447
 -0.56620556   -0.65682779   -0.72000150   -0.72491703
  0.01018497    0.02103964    0.02763786    0.03099923    0.03214338
  0.03188454    0.03021540    0.02692328    0.02179549    0.01441395
  0.00353906   -0.01227416   -0.03447072   -0.06319349   -0.09337694
 -0.11865344   -0.13265535   -0.13073982   -0.11516316   -0.08990646
 -0.05895082   -0.02589342    0.00720430    0.03866480    0.06681054
  0.09030554    0.10917991    0.12380530    0.13455337    0.14171794
  0.14528152    0.14514880    0.14122446    0.13268773    0.11581598
  0.08616112    0.03927506   -0.02644243   -0.10120019   -0.17235925
 -0.22728059   -0.25599494   -0.25921183   -0.24031051   -0.20267024
 -0.15233995   -0.10604743   -0.08319018   -0.10316569
  0.02541859    0.03365331    0.04269781    0.04930801    0.05023987
  0.04304565    0.02846297    0.00802581   -0.01673187   -0.04347976
 -0.06670222   -0.08008723   -0.07732280   -0.05490357   -0.02055065
  0.01520818    0.04184518    0.05160645    0.04783366    0.03664231
  0.02414793    0.01557659    0.01259662    0.01598688    0.02652624
  0.04388273    0.06328092    0.07883455    0.08465734    0.07664478
  0.05781931    0.03298518    0.00694659   -0.01598599   -0.03347715
 -0.04368523   -0.04476857   -0.03574564   -0.01907552    0.00192261
  0.02392956    0.04384923    0.05947799    0.06883530    0.06994065
  0.06103661    0.04125822    0.00996362   -0.03348906
  0.03256336    0.04479571    0.04058059    0.02648697    0.00908378
 -0.00595651   -0.01654750   -0.02149932   -0.01962205   -0.01062232
  0.00220720    0.01467734    0.02259894    0.02306442    0.01829267
  0.01178418    0.00703945    0.00683650    0.01106350    0.01888614
  0.02947013    0.04159320    0.05248114    0.05897180    0.05790302
  0.04738817    0.03064277    0.01215791   -0.00357535   -0.01300617
 -0.01634464   -0.01474107   -0.00934579   -0.00129494    0.00833207
  0.01847001    0.02805366    0.03597661    0.04096780    0.04171499
  0.03690593    0.02570913    0.00921601   -0.01100128   -0.03337056
 -0.05583895   -0.07443062   -0.08468901   -0.08215755
69.25   70.35   70.76   71.13                                 58IWW
72.34   73.26   73.76   74.66                                 59IWW
75.13   75.98   76.28   76.60                                 60IWW
76.99   78.12   78.32   79.07                                 61IWW
```

79.39	80.20	80.97	81.63	62IWW
82.01	83.53	83.35	84.54	63IWW
85.04	86.19	86.56	88.33	64IWW
89.07	90.32	90.95	93.56	65IWW
93.71	95.90	96.97	99.43	66IWW
99.57	102.32	103.35	105.10	67IWW
106.11	108.95	109.99	112.90	68IWW
114.02	116.43	118.21	120.71	69IWW
123.08	125.53	127.02	130.23	70IWW
71.61	72.92	72.62	73.36	58MWW
75.16	75.69	75.56	77.07	59MWW
77.90	78.40	78.19	78.95	60MWW
79.65	80.95	80.34	81.78	61MWW
82.29	83.11	83.13	84.34	62MWW
85.12	86.72	85.30	88.23	63MWW
89.56	90.39	89.81	91.89	64MWW
93.18	94.88	93.59	97.10	65MWW
97.68	99.90	99.24	102.86	66MWW
103.06	106.05	106.12	109.32	67MWW
110.33	114.33	113.20	118.08	68MWW
119.47	121.84	122.19	125.93	69MWW
129.28	131.43	131.83	135.72	70MWW
26315	31993	37698	50680	58DUC
24697	38254	37382	45338	59DUC
28783	28747	31091	35136	60DUC
22835	24594	32981	35198	61DUC
26776	25698	34509	39700	62DUC
25093	30094	35398	37606	63DUC
44385	32050	35337	39191	64DUC
43762	36850	34719	37706	65DUC
35964	47614	39586	39028	66DUC
30003	35552	37600	46087	67DUC
35212	40442	43862	51477	68DUC
37651	52553	50011	55611	69DUC
37127	44019	47969	46713	70DUC
23892	24340	24569	24960	58TPI
25424	26060	26024	26252	59TPI
27096	27216	27472	27860	60TPI
27636	28084	28876	29376	61TPI
30224	30864	30752	31428	62TPI
31688	32932	33186	33936	63TPI
34520	34632	35384	36076	64TPI
37996	37872	39476	40264	65TPI
41888	42392	43412	44560	66TPI
46040	46828	47456	48484	67TPI
49412	51024	52368	53692	68TPI
54500	60568	62568	63624	69TPI
64688	65428	66344	67940	70TPI
22060	22652	22784	23088	58PDI
23468	24016	23876	24048	59PDI

24780	24832	25120	25472	60PDI
25524	25576	26308	26812	61PDI
27584	28188	27992	28624	62PDI
28828	30156	30240	30848	63PDI
31324	31236	31884	32456	64PDI
34196	34020	35544	36200	65PDI
37632	38184	38809	39692	66PDI
40508	41848	41916	42564	67PDI
43000	44788	45652	46416	68PDI
46932	49816	51480	51988	69PDI
53008	52816	53656	54900	70PDI
54.0	54.3	53.8	53.8	58PRT
53.9	53.8	53.7	53.8	59PRT
54.0	54.1	53.8	54.5	60PRT
54.5	54.6	53.9	53.8	61PRT
53.8	54.0	53.9	53.4	62PRT
53.6	53.6	53.9	54.2	63PRT
54.2	54.0	54.1	54.0	64PRT
54.2	54.3	54.4	54.3	65PRT
54.7	55.0	55.0	55.5	66PRT
55.3	55.2	55.7	55.0	67PRT
55.2	55.2	55.5	55.8	68PRT
56.1	56.3	55.6	55.4	69PRT
55.3	55.7	55.8	55.9	70PRT
25.9	26.5	26.4	26.6	58PRF
26.4	26.6	26.7	26.9	59PRF
27.3	27.1	27.5	28.5	60PRF
28.8	29.0	28.6	28.4	61PRF
28.8	29.0	29.2	28.9	62PRT
28.8	28.8	29.8	30.5	63PRF
30.4	30.4	30.3	30.5	64PRF
30.9	31.4	31.4	31.5	65PRF
32.1	32.6	32.8	33.3	66PRF
33.2	33.5	34.1	33.5	67PRF
34.0	33.8	34.3	34.9	68PRF
35.2	35.8	35.1	35.1	69PRF
34.9	35.7	35.7	35.6	70PRF
6.5	6.9	7.5	7.5	58UFR
6.5	5.8	5.5	5.5	59UFR
6.1	6.8	7.1	7.6	60UFR
7.6	7.6	7.2	6.5	61UFR
6.0	5.9	6.0	5.7	62UFR
5.9	5.7	5.4	5.2	63UFR
4.9	4.9	4.6	4.9	64UFR
4.2	4.4	3.9	3.1	65UFR
3.7	3.4	3.7	3.4	66UFR
3.8	4.1	4.0	4.3	67UFR
4.6	4.8	5.1	4.6	68UFR
4.6	4.6	4.6	4.9	69UFR
4.8	5.8	6.6	6.2	70UFR

150.8	151.3	152.0	152.4	58IIP
159.6	166.2	164.8	171.1	59IIP
173.5	164.3	168.0	166.3	60IIP
95.9	98.9	101.6	103.4	61IIP
105.2	107.6	110.2	109.8	62IIP
111.5	113.9	115.9	119.5	63IIP
123.2	125.9	127.7	129.7	64IIP
133.1	134.4	138.2	142.3	65IIP
144.9	145.4	145.7	148.0	66IIP
147.8	150.1	151.9	153.4	67IIP
155.1	160.9	163.7	168.1	68IIP
171.7	172.1	172.5	174.0	69IIP
177.0	174.7	174.8	174.6	70IIP
		32002.00	36671.50	58DUA
36267.00	37832.25	37753.25	36417.75	59DUA
37439.25	35062.50	33489.75	30939.25	60DUA
29452.25	28414.00	28886.50	28902.00	61DUA
29887.25	30163.25	30545.00	31670.50	62DUA
31249.75	32348.75	32571.25	32047.75	63DUA
36870.75	37359.75	37344.50	37740.75	64DUA
37585.00	38785.00	38630.50	38259.25	65DUA
36309.75	39000.75	40217.50	40548.00	66DUA
39057.75	36042.25	35545.75	37310.50	67DUA
38612.75	39835.25	41400.75	42748.25	68DUA
43358.00	46385.75	47923.00	48956.50	69DUA
48825.50	46692.00	46181.25		70DUA
		38134.272	36850.172	58HA
35739.444	34789.608	33988.659	33325.067	59HA
32787.781	32366.225	32050.300	31830.382	60HA
31697.324	31642.457	31657.585	31734.990	61HA
31867.432	32048.145	32270.841	32529.705	62HA
32819.404	33135.076	33472.337	33827.282	63HA
34196.479	34576.972	34966.285	35362.416	64HA
35763.837	36169.502	36578.835	36991.742	65HA
37408.601	37830.269	38258.078	38693.838	66HA
39139.832	39598.822	40074.047	40569.220	67HA
41088.532	41636.649	42218.714	42840.347	68HA
43507.644	44227.176	45005.993	45851.618	69HA
46772.052	47775.774	48871.737		70HA
		53.66	53.78	58PR
53.87	53.94	53.99	54.02	59PR
54.04	54.04	54.03	54.02	60PR
54.00	53.98	53.96	53.94	61PR
53.92	53.90	53.89	53.89	62PR
53.89	53.91	53.93	53.96	63PR
53.99	54.04	54.10	54.17	64PR
54.24	54.33	54.42	54.52	65PR
54.63	54.74	54.85	54.97	66PR
55.08	55.20	55.31	55.42	67PR
55.51	55.60	55.68	55.74	68PR

```
55.78 55.80 55.79 55.76                                                                   69PR
55.69 55.59 55.45                                                                         70PR
                  7.671 6.849                                                             58UR
6.370 6.145 6.102 6.178                                                                   59UR
6.323 6.497 6.670 6.817                                                                   60UR
6.924 6.978 6.976 6.916                                                                   61UR
6.800 6.635 6.426 6.184                                                                   62UR
5.917 5.635 5.349 5.068                                                                   63UR
4.801 4.555 4.336 4.151                                                                   64UR
4.001 3.890 3.816 3.780                                                                   65UR
3.777 3.804 3.856 3.927                                                                   66UR
4.010 4.100 4.189 4.275                                                                   67UR
4.352 4.419 4.478 4.533                                                                   68UR
4.595 4.676 4.797 4.985                                                                   69UR
5.276 5.714 6.353                                                                         70UR
 7   8
 5   2   1   3 49
 1
-3.          3.        -1.
 1.
1(T)  USED
 1
 7   8
 6   2   1   3 49
 1
-3.          3.        -1.
 1.
1(T)  USED
 0
```

```
*******************************************************************

PROGRAM CHANGE CARDS AND DATA FOR RUNNING B EXTENDED IN
     PROGRAM 4.5 :  INOUT

*******************************************************************
READREADREADREADREADREADREADREADREADREADREADREADERADREADREADREAD

READ B(T)

    READ(5,200) (Y(1,I),I=1,STOP)
    LL=INT((52-STOP)/4)
    IF (LL.EQ.0) GO TO 100
    DO 101 J=1,LL
101 READ(5,200) X
100 CONTINUE
200 FORMAT(4F20.8)

READREADREADREADREADREADREADREADREADREADREADREADERADREADREADREAD
*******************************************************************
TO READ A(T,X) REMOVE THE READ BLOCK ABOVE, AND REMOVE C FROM
THE FOLLOWING CARDS:
    DO 10 I=1,P
 10 READ(5,200)(Y(I,K),K=1,STOP)
    LL=INT((50-STOP)/5)
    IF (LL.EQ.0) GO TO 100
    DO 101 J=1,LL
101 READ(5,200) X
100 CONTINUE
    MM=4-P
    IF (MM.EQ.0) GO TO 90
    DO 80 I=1,MM
    DO 80 K=1,10
 80 READ(5,200) X1,X2,X3,X4,X5
 90 CONTINUE
200 FORMAT(5F16.8)
*******************************************************************
```

-0.01519100	-0.00401279	0.00303857	0.00708990
0.00926800	0.01057182	0.01148875	0.01237830
0.01360000	0.01538547	0.01745482	0.01940026
0.02081400	0.02144468	0.02166659	0.02201046
0.02300700	0.02521233	0.02928419	0.03590570
0.04576000	0.05907149	0.07422966	0.08916525
0.10180900	0.11044760	0.11479142	0.11490679
0.11086000	0.10302255	0.09298665	0.08264967
0.07390900	0.06806152	0.06400211	0.06002514
0.05442500	0.04609685	0.03633904	0.02705071
0.02013100	0.01710841	0.01802886	0.02256763
0.03040000	0.04083062	0.05168154	0.06040419
0.06445000			

```
START OF SEGMENT =  1

SEGMENT LENGTH = 19


IFLAG =   0 AND SHOULD BE   0  IF LEAST SQUARES IS SUCCESSFUL


AD IS
  -1.26542462    0.89678966
A  IS
   0.89678966   -3.95579361    7.48664285   -7.69306353    4.26542462
R  IS
   0.00688032   -0.00248872
  -0.00461925   -0.00216065
  -0.00269036    0.01736763
   0.00553795   -0.01741406
  -0.00570185    0.00762743
F  IS
   0.00000000    1.00000000    0.00000000    0.00000000    0.00000000
   0.00000000   00.00000000    1.00000000    0.00000000    0.00000000
   0.00000000   00.00000000    0.00000000    1.00000000    0.00000000
   0.00000000   00.00000000    0.00000000    0.00000000    1.00000000
   0.89678966   -3.95579361    7.48664285   -7.69306353    4.26542462


EIGENVALUES ARE


   0.63271231    0.63271231    1.00024391    1.00024391    0.99951217
   0.70460258   -0.70460258    0.00042352   -0.00042352    0.00000000
G  IS
  -0.00570185    0.00762743
  -0.01878285    0.01512019
  -0.03894251    0.02318331
  -0.06891566    0.03750933
  -0.10555216    0.06218072
H  IS
   1.00000000   00.00000000    0.00000000    0.00000000   -0.00000000
```

```
STIMATION

START OF SEGMENT =  2

SEGMENT LENGTH = 19


IFLAG =   0 AND SHOULD BE   0  IF LEAST SQUARES IS SUCCESSFUL


AD IS
   -1.19203837   0.74518333
A IS
    0.74518333  -3.42758835   6.81166509  -7.32129844   4.19203837
B IS
    0.00201709  -0.00821408
   -0.00490707   0.01418823
    0.00602397   0.00359489
   -0.00444955  -0.01592521
    0.00094864   0.00801562
F IS
    0.00000000   1.00000000   0.00000000   0.00000000   0.00000000
    0.00000000  00.00000000   1.00000000   0.00000000   0.00000000
    0.00000000  00.00000000   0.00000000   1.00000000   0.00000000
    0.00000000  00.00000000   0.00000000   0.00000000   1.00000000
    0.74518333  -3.42758835   6.81166509  -7.32129844   4.19203837


EIGENVALUES ARE


    0.59601918   0.59601918   1.00044051   0.99977974   0.99977974
    0.62445533  -0.62445533   0.00000000   0.00038130  -0.00038130
G IS
    0.00094864   0.00801562
   -0.00047281   0.01767658
   -0.00290334   0.01901103
   -0.00715459   0.01906743
   -0.01319117   0.02546457
H IS
    1.00000000  00.00000000  -0.00000000  -0.00000000   0.00000000
```

```
STIMATION

START OF SEGMENT =   3

SEGMENT LENGTH = 19



IFLAG =   0 AND SHOULD BE   0   IF LEAST SQUARES IS SUCCESSFUL


AD IS
  -1.26378220    0.68129906
A IS
   0.68129906   -3.30767938    6.83524379   -7.47264567    4.26378220
B IS
   0.00167375   -0.00970013
  -0.00262245    0.01435832
   0.00437364    0.00215318
  -0.00436637   -0.01304308
   0.00099355    0.00613211
F IS
   0.00000000    1.00000000    0.00000000    0.00000000    0.00000000
   0.00000000   00.00000000    1.00000000    0.00000000    0.00000000
   0.00000000   00.00000000    0.00000000    1.00000000    0.00000000
   0.00000000   00.00000000    0.00000000    0.00000000    1.00000000
   0.68129906   -3.30767938    6.83524379   -7.47264567    4.26378220


EIGENVALUES ARE


   0.63189110    0.63189110    1.00067668    0.99966166    0.99966166
   0.53104867   -0.53104867    0.00000000    0.00058508   -0.00058508
G IS
   0.00099355    0.00613211
  -0.00013011    0.01310291
  -0.00360553    0.01219805
  -0.01023225    0.01036919
  -0.01918717    0.01263871
H IS
   1.00000000   00.00000000    0.00000000   -0.00000000   -0.00000000
```

STIMATION

START OF SEGMENT = 4

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AD IS

-1.14564597	0.64560558

A IS

0.64560558	-3.08246272	6.37375466	-7.08254350	4.14564597

B IS

-0.00148331	-0.00723149
-0.00106349	0.01618186
0.00477291	-0.00364489
-0.00641438	-0.00901489
0.00387717	0.00494888

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.64560558	-3.08246272	6.37375466	-7.08254350	4.14564597

EIGENVALUES ARE

0.57282299	0.57282299	1.00070651	0.99964675	0.99964675
0.56345311	-0.56345311	0.00000000	0.00061141	-0.00061141

G IS

0.00387717	0.00494888
0.00965899	0.01150144
0.01735544	0.00898531
0.02718793	0.00351531
0.03792036	0.00175550

H IS

1.00000000	00.00000000	-0.00000000	0.00000000	-0.00000000

```
STIMATION

START OF SEGMENT =   5

SEGMENT LENGTH = 19


IFLAG =   0 AND SHOULD BE   0  IF LEAST SQUARES IS SUCCESSFUL


AD IS
  -1.04951774    0.40512567
A IS
   0.40512567   -2.26489475    5.36393023   -6.55367889    4.04951774
B IS
  -0.00051694   -0.01195577
  -0.00568333    0.02303912
   0.01047364   -0.00695677
  -0.01331973   -0.00330739
   0.00889982   -0.00061177
F IS
   0.00000000    1.00000000    0.00000000    0.00000000    0.00000000
   0.00000000   00.00000000    1.00000000    0.00000000    0.00000000
   0.00000000   00.00000000    0.00000000    1.00000000    0.00000000
   0.00000000   00.00000000    0.00000000    0.00000000    1.00000000
   0.40512567   -2.26489475    5.36393023   -6.55367889    4.04951774


EIGENVALUES ARE


   1.01483609    1.01483609    0.97032783    0.52475887    0.52475887
   0.02569686   -0.02569686    0.00000000    0.36021355   -0.36021355
G IS
   0.00889982   -0.00061177
   0.02272025   -0.00578477
   0.04415314   -0.02637294
   0.07195236   -0.04912857
   0.10320263   -0.06770648
H IS
   1.00000000   00.00000000    0.00000000    0.00000000    0.00000000
```

STIMATION

```
START OF SEGMENT =  6

SEGMENT LENGTH = 19


IFLAG =  0 AND SHOULD BE  0  IF LEAST SQUARES IS SUCCESSFUL


AD IS
  -0.90449731    0.23133341
A IS
   0.23133341   -1.59849754    4.40749216   -5.94482534    3.90449731
B IS
  -0.00178924   -0.01330019
  -0.00417770    0.02451357
   0.00621195   -0.00166101
  -0.00921082   -0.00727592
   0.00882883   -0.00245669
F IS
   0.00000000    1.00000000    0.00000000    0.00000000    0.00000000
   0.00000000   00.00000000    1.00000000    0.00000000    0.00000000
   0.00000000   00.00000000    0.00000000    1.00000000    0.00000000
   0.00000000   00.00000000    0.00000000    0.00000000    1.00000000
   0.23133341   -1.59849754    4.40749216   -5.94482534    3.90449731


EIGENVALUES ARE

   1.00013708    1.00013708    0.99972583    0.45224865    0.45224865
   0.00023891   -0.00023891    0.00000000    0.16372100   -0.16372100
G IS
   0.00882883   -0.00245669
   0.02526133   -0.01686806
   0.05235890   -0.05291772
   0.08899626   -0.09265367
   0.13165820   -0.13089846
H IS
   1.00000000   00.00000000    0.00000000   -0.00000000    0.00000000
```

STIMATION

START OF SEGMENT = 7

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AD IS

-0.92287544	0.26531569

A IS

0.26531569	-1.71882249	4.56457336	-6.03394199	3.92287544

B IS

-0.00215354	-0.00926941
-0.00199594	0.01724665
0.00292348	0.00244602
-0.00641292	-0.00704448
0.00736643	-0.00292197

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.26531569	-1.71882249	4.56457336	-6.03394199	3.92287544

EIGENVALUES ARE

1.00030085	1.00030085	0.99939830	0.46143772	0.46143772
0.00052113	-0.00052113	0.00000000	0.22889062	-0.22889062

G IS

0.00736643	-0.00292197
0.02248466	-0.01850702
0.04567941	-0.05252370
0.07907502	-0.09046457
0.11635838	-0.12667997

H IS

1.00000000	00.00000000	-0.00000000	0.00000000	-0.00000000

```
STIMATION

START OF SEGMENT =   8

SEGMENT LENGTH = 19


IFLAG =   0 AND SHOULD BE   0   IF LEAST SQUARES IS SUCCESSFUL


AD IS
   -0.79577604    0.20781404
A IS
    0.20781404   -1.41921815    4.01077023   -5.59514216    3.79577604
R IS
   -0.00348059   -0.00556941
   -0.00192222    0.01269075
    0.00344716    0.00636407
   -0.00491632   -0.00601200
    0.00661731   -0.00748983
F IS
    0.00000000    1.00000000    0.00000000    0.00000000    0.00000000
    0.00000000   00.00000000    1.00000000    0.00000000    0.00000000
    0.00000000   00.00000000    0.00000000    1.00000000    0.00000000
    0.00000000   00.00000000    0.00000000    0.00000000    1.00000000
    0.20781404   -1.41921815    4.01077023   -5.59514216    3.79577604


EIGENVALUES ARE


    1.00030025    1.00030025    0.99939950    0.39788802    0.39788802
    0.00052062   -0.00052062    0.00000000    0.22248406   -0.22248406
G IS
    0.00661731   -0.00748983
    0.02020150   -0.03444171
    0.04310275   -0.08246228
    0.07519640   -0.13765134
    0.11241427   -0.19418295
H IS
    1.00000000   00.00000000   -0.00000000    0.00000000   -0.00000000
```

STIMATION

START OF SEGMENT = 9

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AD IS

```
  -0.75501246    0.18308996
A IS
   0.18308996   -1.30428233    3.81430724   -5.44812733    3.75501246
B IS
  -0.00286706   -0.00394512
  -0.00224163    0.01040686
   0.00218671    0.00654601
  -0.00290080   -0.00384390
   0.00550181   -0.00888767
F IS
   0.00000000    1.00000000    0.00000000    0.00000000    0.00000000
   0.00000000   00.00000000    1.00000000    0.00000000    0.00000000
   0.00000000   00.00000000    0.00000000    1.00000000    0.00000000
   0.00000000   00.00000000    0.00000000    0.00000000    1.00000000
   0.18308996   -1.30428233    3.81430724   -5.44812733    3.75501246
```

EIGENVALUES ARE

```
   1.03152876    1.03152876    0.93694248    0.37750623    0.37750623
   0.05460941   -0.05460941    0.00000000    0.20144231   -0.20144231
G IS
   0.00550181   -0.00888767
   0.01775857   -0.03721721
   0.03889579   -0.08478391
   0.06804720   -0.13909402
   0.10130252   -0.19469717
H IS
   1.00000000   00.00000000   -0.00000000    0.00000000   -0.00000000
```

STIMATION

START OF SEGMENT = 10

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AD IS

-0.78491628	0.21657469

A IS

0.21657469	-1.43464036	4.00447292	-5.57132353	3.78491628

B IS

-0.00248161	-0.00149081
-0.00213180	0.00988762
0.00259589	0.00195632
-0.00512035	-0.00166530
0.00691360	-0.00897101

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.21657469	-1.43464036	4.00447292	-5.57132353	3.78491628

EIGENVALUES ARE

1.00036063	1.00036063	0.99927873	0.39245814	0.39245814
0.00062522	-0.00062522	0.00000000	0.25010258	-0.25010258

G IS

0.00691360	-0.00897101
0.02104705	-0.03561984
0.04373930	-0.08288136
0.07384320	-0.14128593
0.10768674	-0.20425587

H IS

1.00000000	00.00000000	-0.00000000	0.00000000	-0.00000000

```
STIMATION

START OF SEGMENT = 11

SEGMENT LENGTH = 19


IFLAG =  0 AND SHOULD BE  0  IF LEAST SQUARES IS SUCCESSFUL


AD IS
  -0.76442227   0.21014709
A IS
   0.21014709  -1.39486353   3.92370808  -5.50341390   3.76442227
R IS
  -0.00413231   0.00302385
   0.00297152   0.00229573
  -0.00610970   0.00364021
   0.00114086  -0.00473537
   0.00606700  -0.00545998
F IS
   0.00000000   1.00000000   0.00000000   0.00000000   0.00000000
   0.00000000  00.00000000   1.00000000   0.00000000   0.00000000
   0.00000000  00.00000000   0.00000000   1.00000000   0.00000000
   0.00000000  00.00000000   0.00000000   0.00000000   1.00000000
   0.21014709  -1.39486353   3.92370808  -5.50341390   3.76442227


EIGENVALUES ARE


   1.00018930   0.99990535   0.99990535   0.38221114   0.38221114
   0.00000000   0.00016366  -0.00016366   0.25310420  -0.25310420
G IS
   0.00606700  -0.00545998
   0.02397962  -0.02528903
   0.05077047  -0.06150986
   0.08592842  -0.11150072
   0.12555398  -0.16980857
H IS
   1.00000000  00.00000000   0.00000000   0.00000000  -0.00000000
```

STIMATION

START OF SEGMENT = 32

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AO IS

-0.80888476	0.15073370

A IS

0.15073370	-1.26108585	3.87885537	-5.57738798	3.80888476

B IS

-0.00363325	-0.00070504
0.00004890	0.00424556
-0.00297352	0.00343481
-0.00155812	-0.00264680
0.00808057	-0.00553324

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.15073370	-1.26108585	3.87885537	-5.57738798	3.80888476

EIGENVALUES ARE

1.00019889	1.00019889	0.99960222	0.51775585	0.29112891
0.00034467	-0.00034467	0.00000000	0.00000000	0.00000000

G IS

0.00808057	-0.00553324
0.02921986	-0.02372226
0.06325306	-0.05605953
0.10934540	-0.09843311
0.16321322	-0.14799699

H IS

1.00000000	00.00000000	0.00000000	0.00000000	-0.00000000

STIMATION

START OF SEGMENT = 73

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AD IS

-0.79276325	0.25982407

A IS

0.25982407	-1.57223547	4.15776196	-5.63811382	3.79276325

B IS

-0.00610185	-0.00018833
0.00254886	0.00039951
-0.00339034	0.00201749
0.00000250	-0.00165301
0.00704195	-0.00241387

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.25982407	-1.57223547	4.15776196	-5.63811382	3.79276325

EIGENVALUES ARE

1.02500919	1.02500919	0.94998161	0.39638162	0.39638162
0.04331720	-0.04331720	0.00000000	0.32047727	-0.32047727

G IS

0.00704195	-0.00241387
0.02671096	-0.01080823
0.05821466	-0.02536592
0.10202264	-0.04490566
0.15261116	-0.06863181

H IS

1.00000000	00.00000000	-0.00000000	0.00000000	-0.00000000

STIMATION

START OF SEGMENT = 14

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AD IS

-0.80929245	0.44512379

A IS

0.44512379	-2.14466383	4.76324873	-5.87300115	3.80929245

B IS

-0.00692699	-0.00348113
0.00422584	-0.00121345
-0.00183177	0.00214217
0.00006015	0.00097632
0.00488809	-0.00173978

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.44512379	-2.14466383	4.76324873	-5.87300115	3.80929245

EIGENVALUES ARE

0.40464623	0.40464623	1.00050856	0.99974572	0.99974572
0.53045756	-0.53045756	0.00000000	0.00044039	-0.00044039

G IS

0.00488809	-0.00173978
0.01868033	-0.00565103
0.04061930	-0.00916651
0.07253023	-0.01122988
0.10930041	-0.01561012

H IS

1.00000000	00.00000000	0.00000000	-0.00000000	0.00000000

STIMATION

START OF SEGMENT = 75

SEGMENT LENGTH = 12

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AD IS

-0.80597118	0.27424870

A IS

0.27424870	-1.62871729	4.24065964	-5.69216223	3.80597118

B IS

-0.00534495	-0.00231371
0.00142911	0.00128048
-0.00220794	0.00175577
0.00303043	-0.00017794
0.00324326	-0.00221329

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.27424870	-1.62871729	4.24065964	-5.69216223	3.80597118

EIGENVALUES ARE

1.02529247	1.02529247	0.94941505	0.40298559	0.40298559
0.04380785	-0.04380785	0.00000000	0.33444180	-0.33444180

G IS

0.00324326	-0.00221329
0.01537417	-0.00860164
0.03784458	-0.01838345
0.07170574	-0.02911025
0.11206183	-0.04133672

H IS

1.00000000	00.00000000	0.00000000	-0.00000000	0.00000000

```
STIMATION

START OF SEGMENT = 56

SEGMENT LENGTH = 19


IFLAG =   0 AND SHOULD BE   0   IF LEAST SQUARES IS SUCCESSFUL


AD IS
  -0.90855742    0.47199793
A IS
   0.47199793   -2.32455119    5.14166602   -6.19767017    3.90855742
B IS
  -0.01064642    0.00365678
   0.00555548   -0.00140596
   0.00082738   -0.00373271
  -0.00658391    0.00150659
   0.01081033   -0.00144471
F IS
   0.00000000    1.00000000    0.00000000    0.00000000    0.00000000
   0.00000000   00.00000000    1.00000000    0.00000000    0.00000000
   0.00000000   00.00000000    0.00000000    1.00000000    0.00000000
   0.00000000   00.00000000    0.00000000    0.00000000    1.00000000
   0.47199793   -2.32455119    5.14166602   -6.19767017    3.90855742


EIGENVALUES ARE

   1.00031797    1.00031797    0.99936407    0.45427871    0.45427871
   0.00055140   -0.00055140    0.00000000    0.51539187   -0.51539187
G IS
   0.01081033   -0.00144471
   0.03566887   -0.00414016
   0.07324237   -0.01096088
   0.12634670   -0.02601612
   0.18752311   -0.04802576
H IS
   1.00000000   00.00000000  -0.00000000    0.00000000   -0.00000000
```

STIMATION

START OF SEGMENT = 17

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AD IS

-1.06019872	0.45166356

A IS

0.45166356	-2.41518941	5.53558685	-6.63225973	4.06019872

B IS

-0.00915776	0.00298306
0.00412089	-0.00061051
-0.00009474	-0.00144210
-0.00700694	0.00148354
0.01202623	-0.00355216

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.45166356	-2.41518941	5.53558685	-6.63225973	4.06019872

EIGENVALUES ARE

1.00035862	1.00035862	0.99928276	0.53009936	0.53009936
0.00062135	-0.00062135	0.00000000	0.41310801	-0.41310801

G IS

0.01202623	-0.00355216
0.04182193	-0.01293894
0.08994954	-0.03041793
0.15853222	-0.05796221
0.24040916	-0.09366092

H IS

1.00000000	00.00000000	-0.00000000	0.00000000	-0.00000000

STIMATION

START OF SEGMENT = 18

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AD IS

-1.18675231	0.64918787

A IS

0.64918787	-3.13431593	6.50782055	-7.20944481	4.18675231

B IS

-0.00658740	0.00242062
0.00254869	-0.00243235
0.00262506	-0.00053045
-0.01179143	0.00358204
0.01310804	-0.00371020

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.64918787	-3.13431593	6.50782055	-7.20944481	4.18675231

EIGENVALUES ARE

0.59337616	0.59337616	1.00027965	1.00027965	0.99944070
0.54506202	-0.54506202	0.00048495	-0.00048495	0.00000000

G IS

0.01310804	-0.00371020
0.04308869	-0.01195166
0.08852503	-0.02382058
0.14784034	-0.04014372
0.21349586	-0.06006835

H IS

1.00000000	00.00000000	0.00000000	0.00000000	0.00000000

STIMATION

START OF SEGMENT = 79

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQJARES IS SUCCESSFUL

AD IS

-1.23565501	0.65762039

A IS

0.65762039	-3.20851616	6.67982618	-7.36458541	4.23565501

B IS

-0.00615294	0.00489107
0.00249312	-0.00339661
0.00713415	-0.00304371
-0.01143807	0.00341863
0.00818639	-0.00462968

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.65762039	-3.20851616	6.67982618	-7.36458541	4.23565501

EIGENVALUES ARE

0.61782750	0.61782750	1.00073351	0.99963325	0.99963325
0.52527094	-0.52527094	0.00000000	0.00063465	-0.00063465

G IS

0.00818639	-0.00462968
0.02323667	-0.01619109
0.04526728	-0.03752791
0.07778494	-0.07403669
0.11889323	-0.12562453

H IS

1.00000000	00.00000000	-0.00000000	0.00000000	-0.00000000

STIMATION

START OF SEGMENT = 20

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AD IS

-1.26593214	0.72338848

A IS

0.72338848	-3.43609759	6.96796186	-7.52118490	4.26593214

B IS

-0.00724240	0.00370138
0.00274523	-0.00338283
0.00575655	-0.00218670
-0.01102021	0.00480343
0.00983892	-0.00475739

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.72338848	-3.43609759	6.96796186	-7.52118490	4.26593214

EIGENVALUES ARE

0.63296607	0.63296607	1.00060054	0.99969973	0.99969973
0.56810425	-0.56810425	0.00000000	0.00051966	-0.00051966

G IS

0.00983892	-0.00475739
0.03095196	-0.01549130
0.06379518	-0.03249028
0.11065292	-0.05862059
0.16684474	-0.09360059

H IS

1.00000000	00.00000000	0.00000000	0.00000000	0.00000000

STIMATION

START OF SEGMENT = 21

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AD IS

-1.23953147	0.70320544

A IS

0.70320544	-3.34914779	6.82821072	-7.42179984	4.23953147

B IS

-0.01455222	0.00671237
0.00288408	-0.00463978
0.00989943	-0.00545427
-0.00910062	0.00532703
0.01025634	0.00067154

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.70320544	-3.34914779	6.82821072	-7.42179984	4.23953147

EIGENVALUES ARE

0.61976573	0.61976573	1.00069662	0.99965169	0.99965169
0.56488572	-0.56488572	0.00000000	0.00060245	-0.00060245

G IS

0.01025634	0.00067154
0.03438147	0.00817405
0.07954023	0.02421584
0.15495746	0.04194328
0.25247706	0.05837216

H IS

1.00000000	00.00000000	0.00000000	0.00000000	0.00000000

STIMATION

START OF SEGMENT = 22

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AD IS

-1.24762821 0.69291061

A IS

0.69291061 -3.32636004 6.82161646 -7.43579524 4.24762821

B IS

-0.00911961	0.00584047
0.00459404	-0.00554571
0.00787710	-0.00298897
-0.01127271	0.00563374
0.00790458	-0.00400688

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.69291061	-3.32636004	6.82161646	-7.43579524	4.24762821

EIGENVALUES ARE

0.62381420	0.62381420	1.00013866	1.00013866	0.99972248
0.55115017	-0.55115017	0.00023768	-0.00023768	0.00000000

G IS

0.00790458	-0.00400688
0.02230301	-0.01138601
0.04383515	-0.02155814
0.07887086	-0.03978606
0.12579438	-0.06719666

H IS

1.00000000 00.00000000 0.00000000 0.00000000 0.00000000

```
STIMATION

START OF SEGMENT = 33

SEGMENT LENGTH = 19


IFLAG =   0 AND SHOULD BE   0   IF LEAST SQUARES IS SUCCESSFUL


A0 IS
  -1.34900934    0.77182465
A IS
   0.77182465   -3.66448329    7.36250196   -7.81885266    4.34900934
B IS
  -0.01205962    0.00832378
   0.00554537   -0.00680497
   0.01092044   -0.00511122
  -0.00949317    0.00687790
   0.00468272   -0.00118089
F IS
   0.00000000    1.00000000    0.00000000    0.00000000    0.00000000
   0.00000000   00.00000000    1.00000000    0.00000000    0.00000000
   0.00000000   00.00000000    0.00000000    1.00000000    0.00000000
   0.00000000   00.00000000    0.00000000    0.00000000    1.00000000
   0.77182465   -3.66448329    7.36250196   -7.81885266    4.34900934


EIGENVALUES ARE


   0.67450467    0.67450467    1.00037978    1.00037978    0.99924043
   0.56291039   -0.56291039    0.00065829   -0.00065829    0.00000000
G IS
   0.00468272   -0.00118089
   0.01087204    0.00174218
   0.02158953    0.01169878
   0.04890810    0.02175695
   0.09472248    0.02862810
H IS
   1.00000000   00.00000000    0.00000000    0.00000000    0.00000000
```

STIMATION

START OF SEGMENT = 24

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AD IS

-1.56812105	0.97485305

A IS

0.97485305	-4.49268019	8.62892230	-8.67921621	4.56812105

R IS

-0.01379519	0.00969487
0.00658813	-0.00828286
0.01735761	-0.00652260
-0.01278995	0.00716304
0.00212127	0.00136748

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.97485305	-4.49268019	8.62892230	-8.67921621	4.56812105

EIGENVALUES ARE

0.78406052	0.78406052	1.00081445	0.99959278	0.99959278
0.60008511	-0.60008511	0.00000000	0.00070409	-0.00070409

G IS

0.00212127	0.00136748
-0.00309975	0.01340986
-0.01521336	0.04286661
-0.01770067	0.08294979
0.00110833	0.12614001

H IS

1.00000000	00.00000000	-0.00000000	0.00000000	-0.00000000

STIMATION

START OF SEGMENT = 25

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQJARES IS SUCCESSFUL

AD IS

```
  -1.62572935   1.32624782
A IS
   1.32624782  -5.60447282   9.85593153  -9.20343589   4.62572935
B IS
  -0.01744433   0.00801470
   0.00565165  -0.00794285
   0.02420694  -0.01381690
  -0.00945590   0.00831978
  -0.00231418  -0.00096906
F IS
   0.00000000   1.00000000   0.00000000   0.00000000   0.00000000
   0.00000000  00.00000000   1.00000000   0.00000000   0.00000000
   0.00000000  00.00000000   0.00000000   1.00000000   0.00000000
   0.00000000  00.00000000   0.00000000   0.00000000   1.00000000
   1.32624782  -5.60447282   9.85593153  -9.20343589   4.62572935
```

EIGENVALUES ARE

```
   0.81286468   0.81286468   1.00025117   1.00025117   0.99949766
   0.81578112  -0.81578112   0.00043580  -0.00043580   0.00000000
G IS
  -0.00231418  -0.00096906
  -0.02016068   0.00383717
  -0.04775249   0.01285150
  -0.05249935   0.00663856
  -0.00653762  -0.03630511
H IS
   1.00000000  00.00000000  -0.00000000   0.00000000  -0.00000000
```

STIMATION

START OF SEGMENT = 26

SEGMENT LENGTH = 19

IFLAG = 0 AND SHOULD BE 0 IF LEAST SQUARES IS SUCCESSFUL

AD IS

-1.58949624	0.91423642

A IS

0.91423642	-4.33220552	8.51119800	-8.68272515	4.58949624

B IS

-0.01218463	0.01063012
0.00434178	-0.01095251
0.01734079	-0.00582620
-0.01539899	0.00565207
0.00518362	0.00526354

F IS

0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
0.00000000	00.00000000	1.00000000	0.00000000	0.00000000
0.00000000	00.00000000	0.00000000	1.00000000	0.00000000
0.00000000	00.00000000	0.00000000	0.00000000	1.00000000
0.91423642	-4.33220552	8.51119800	-8.68272515	4.58949624

EIGENVALUES ARE

0.79474812	0.79474812	1.00033966	1.00033966	0.99932067
0.53161250	-0.53161250	0.00058832	-0.00058832	0.00000000

G IS

0.00518362	0.00526354
0.00839120	0.02980909
0.01084425	0.09528058
0.02537174	0.16641738
0.05906388	0.26484246

H IS

1.00000000	80.00000000	0.00000000	-0.00000000	0.00000000

```
STIMATION

START OF SEGMENT = 57

SEGMENT LENGTH = 19


IFLAG =   0 AND SHOULD BE   0  IF LEAST SQUARES IS SUCCESSFUL


AD IS
  -4.72483214    3.44868766
A IS
   3.44868766 -15.07089512  25.52055939 -20.62318407   7.72483214
B IS
  -0.15423973  -0.02198008
   0.02640355   0.03077331
  -0.04575441  -0.20855752
   0.12131908   0.00256785
   0.03280270   0.31457940
F IS
   0.00000000   1.00000000   0.00000000   0.00000000   0.00000000
   0.00000000  00.00000000   1.00000000   0.00000000   0.00000000
   0.00000000  00.00000000   0.00000000   1.00000000   0.00000000
   0.00000000  00.00000000   0.00000000   0.00000000   1.00000000
   3.44868766 -15.07089512  25.52055939 -20.62318407   7.72483214


EIGENVALUES ARE

   3.82266332   1.00028130   1.00028130   0.99943739   0.90216883
   0.00000000   0.00048739  -0.00048739   0.00000000   0.00000000
G IS
   0.03280270   0.31457940
   0.37471440   2.43264088
   2.17235540  12.09555615
   9.91682343  51.32635562
  40.71922649 204.35798347
H IS
   1.00000000  00.00000000   0.00000000   0.00000000   0.00000000
```

SUMMARY OF AD: A1(T) USED

FACTORS USED : PARTICIPATION RATE FEMALES
UNEMPLOYMENT RATE

N = 5 M = 2 P = 1 D = 3 STOP = 49 SEGMENT LENGTH = 19 SHIFT =

JULY	58	TO	JAN	64	-1.265425	0.896790
OCT	58	TO	APR	64	-1.192038	0.745183
JAN	59	TO	JULY	64	-1.263782	0.681299
APR	59	TO	OCT	64	-1.145646	0.645606
JULY	59	TO	JAN	65	-1.049518	0.405126
OCT	59	TO	APR	65	-0.904497	0.231333
JAN	60	TO	JULY	65	-0.922875	0.265316
APR	60	TO	OCT	65	-0.795776	0.207814
JULY	60	TO	JAN	66	-0.755012	0.183090
OCT	60	TO	APR	66	-0.784916	0.216575
JAN	61	TO	JULY	66	-0.764422	0.210147
APR	61	TO	OCT	66	-0.808885	0.150734
JULY	61	TO	JAN	67	-0.792763	0.259824
OCT	61	TO	APR	67	-0.809292	0.445124
JAN	62	TO	JULY	67	-0.805971	0.274249
APR	62	TO	OCT	67	-0.908557	0.471998
JULY	62	TO	JAN	68	-1.060199	0.451664
OCT	62	TO	APR	68	-1.186752	0.649188
JAN	63	TO	JULY	68	-1.235655	0.657620
APR	63	TO	OCT	68	-1.265932	0.723388
JULY	63	TO	JAN	69	-1.239531	0.703205
OCT	63	TO	APR	69	-1.247628	0.692911
JAN	64	TO	JULY	69	-1.349009	0.771825
APR	64	TO	OCT	69	-1.568121	0.974853
JULY	64	TO	JAN	70	-1.625729	1.326248
OCT	64	TO	APR	70	-1.589496	0.914236
JAN	65	TO	JULY	70	-4.724832	3.448688

Bibliography

[1] A. J. Coale, The Growth and Structure of Human Populations, Princeton Univ. Press 1972.

[2] J. H. Pollard, Mathematical Models for the Growth of Human Populations, Cambridge University Press, 1973.

[3] J. Davis and J. Verner, "Interim Report to the Ministry of State for Science and Technology on the Queen's Mathematics Department Population Model", Queen's University, Kingston, Ontario, Queen's Mathematical Preprint No. 1974-8. (also included as Appendix D above.)

[4] Economic Council of Canada, Ninth Annual Review, "The Years to 1980", 1972.

[5] Economic Council of Canada, "The Economy to 1980: Staff Papers", 1972.

[6] J. Henripin, Trends and Factors of Fertility in Canada, 1961 Census Monograph, Statistics Canada, 1972.

[7] W. E. Kalbach, The Impact of Immigration on Canada's Population, 1961 Census Monograph, Dominion Bureau of Statistics, 1970.

[8] Economic Council of Canada, Fourth Annual Review, "The Canadian Economy from the 1960's to the 1970's".

[9] D. Walters, Canadian Income Levels and Growth, Economic Council of Canada.

[10] F. Denton et. Al., Population and Labour Force Projections to 1970, Economic Council of Canada.

[11] F. Denton, An Analysis of Interregional Differences in Manpower Utilization and Earnings, Economic Council of Canada.

[12] R. Courant and D. Hilbert, Methods of Mathematical Physics, Volume II, Partial Differential Equations, Wiley-Interscience, New York, 1965.

[13] S. Lang, Linear Algebra, Addison-Wesley, 1971.

[14] E. I. Jury, Theory and Applications of the z-transform John Wiley and Sons, New York 1964.

[15] K. W. Morton and R. D. Richtmeyer, Difference Methods for Initial Value Problems, 2nd Edition, Interscience Publishers, New York, 1967.

[16] B. Gopinath, Identification and Control of Linear Systems, Ph.D. Thesis, Stanford University, 1986.

[17] M. A. Budin, "Minimal Realization of Discrete Linear Systems from Input-Output Observations", IEEE Transactions on Automatic Control, Vol. AC-16, No. 5, October 1971.

[18] R. D. Gupta and F. W. Fairman, "Parameter estimation for Multivariable Systems" IEEE Transactions on Automatic Control, Vol. AC-19, 1974.

[19] R. D. Gupta, Parameter Estimation in Linear Models, Ph.D. Thesis, Queen's University, 1974.

[20] T. W. Anderson, The Statistical Analysis of Time Series, John Wiley and Sons, New York, 1971.

[21] A. Zellner, An Introduction to Bayesian Inference in Econometrics, John Wiley and Sons, New York, 1971.

[22] A. H. Stroud, Approximate Calculation of Multiple Integrals, Prentice Hall, Englewood Cliff, N. J., 1971.

[23] Vital Statistics 1957-60, 1961-65, 1966-69, Dominion Bureau of Statistics, Ottawa.

[24] Taxation Statistics 1960-1963, 1964-1966, 1967-1969, 1967-1970, 1970-1973, Department of National Revenue, Taxation Division, Ottawa.