International Institute for
Applied Systems Analysis
IIASA www.iiasa.ac.at

Computer programs for the IIASA health care resource allocation sub-model, MARK 1- a user's guide

Gibbs, R.J.

IIASA Working Paper

WP-78-015

May 1978

COMPUTER PROGRAMS FOR THE IIASA HEALTH CARE RESOURCE ALLOCATION SUB-MODEL, MARK 1--A USER'S GUIDE

Richard J. Gibbs

May 1978 WP-78-15

2361
Laxenburg
Austria

International Institute for Applied Systems Analysis

Preface

The aim of the IIASA Modeling Health Care Systems Task is to build a National Health Care System model and apply it in collaboration with national research centers as an aid to Health Service planners. The modeling work is proceeding along the lines proposed in earlier papers by Venedictov and Shigan [1] among others. It involves the construction of linked sub-models dealing with population, disease prevalence, resource need, resource supply and resource allocation.

A preliminary version of the resource allocation sub-model has already been built. It has been called DRAM Mark 1 and has been described in previous papers--Gibbs [2] and Gibbs [3]. (DRAM is an acronym for Disaggregated Resource Allocation Model.) This paper provides a user's guide to the computer programs that have been written for DRAM Mark 1. It should be read in conjunction with the description of the model given in Gibbs [3]. This guide provides all the information needed for the reader to be able to install and run the computer programs and use the sub-model, DRAM Mark 1, on its own.

Recent publications of the Modeling Health Care Systems Task are listed on the back pages of this Paper. By consulting them the reader will be able to learn how DRAM can be used in conjunction with the other sub-models of the Task.

Evgenii N. Shigan
Task Leader
April 1978

Abstract

The functions of the different files used in running DRAM Mark 1 are described. The definitions of the variables used in the programs are listed in a dictionary. Flow diagrams and listings of the programs are displayed. Some guidance is given on suitable values for the input data. Input and output for some illustrative runs of the programs are displayed. Finally a description is given of the error messages that can be encountered and appropriate remedial action is presented.

Contents

1.	PURPOSE AND LAYOUT OF THE PAPER	1
2.	ORGANISATION OF FILES	2
3.	DICTIONARY OF VARIABLES USED IN MAIN PROGRAMS	5
4.	MAIN PROGRAMS--FLOW DIAGRAMS AND LISTINGS	9
5.	DESCRIPTION OF SUBROUTINES	29
6.	SUITABLE VALUES FOR INPUT DATA	34
7.	CASE 1 ILLUSTRATIVE INPUT/OUTPUT	38
8.	CASE 2 ILLUSTRATIVE INPUT/OUTPUT	45
9.	CASE 3 ILLUSTRATIVE INPUT/OUTPUT	55
10.	ERROR MESSAGES	71
	REFERENCES	72
	PAPERS OF THE MODELING HEALTH CARE SYSTEMS STUDY	73

1. PURPOSE AND LAYOUT OF THE PAPER

This paper has been written for the reader who wishes to learn how to use the computer programs for the Mark 1 version of the resource allocation sub-model of the IIASA National Health Care System (HCS) Model. The sub-model is called DRAM (disaggregated resource allocation model). An earlier paper, Gibbs [2], described the function of DRAM within the IIASA HCS model. Another paper, Gibbs [3], describes the formulation, solution and parameter estimation procedure for DRAM and displays some illustrative model runs. It will be assumed here that the reader is familiar with at least the second paper, Gibbs [3].

The function of the model DRAM is to simulate the allocation of an HCS resource whose availability is limited. For example it can be used to simulate the allocation by the HCS of a limited number, B, of bed-days between admission rates and lengths of stay for a number of patient categories. The model contains parameters describing ideal admission rates and lengths of stay and power factors of the corresponding terms of the HCS utility function. In Gibbs [3] three possible data situations were envisaged:

- *Case 1*: exogenous estimates available for all parameters.
- *Case 2*: no exogenous estimates available for the parameters--parameters to be estimated endogenously from data on past allocations and elasticities of past allocations with respect to bed supply.
- *Case 3*: exogenous data available for ideal admission rates and lengths of stay but power factors to be estimated endogenously from data on elasticities of past allocations.

This paper describes the files for computer programs, and the input and output data that have been designed for each of these three cases. The organisation of the files is described in Section 2. A dictionary of the Fortran variables is given in Section 3. Section 4 provides flow diagrams and listings of the main programs for each of the three cases. Section 5 describes the subroutines, which are common to all three cases. Section 6 describes suitable values and restrictions on the input data. Section 7 lists input and output files for an illustrative model run for Case 1. Sections 8 and 9 provide input/output listings for Cases 2 and 3. Finally Section 10 describes the error messages that can be encountered and the appropriate remedies.

2. ORGANISATION OF FILES

The following files are used in running the model DRAM:

- Main Fortran program,
- Fortran subroutines 'EF' and 'STEP',
- Input files 'AUXPA' and 'INIDA' and
- Output files 'OUTPUT' and 'CALCNS'.

The use of these files is described below in terms of operation on the PDP-11 computer at IIASA, taking advantage of the UNIX operating system. However it should be possible to run the programs with little alteration on any computer with a Fortran IV compiler.

MAIN PROGRAM

There is a different main program for each of the three cases envisaged in Gibbs [3]. The main program reads auxiliary input data from file 'AUXPA' (unit 5) and main input data from file 'INIDA' (unit 7). It then writes the input data (and computed column totals) to the terminal (unit 6) and to the main output file 'OUTPUT'. It checks the main input data; if any errors are detected the appropriate error messages are written in the terminal, as described in Section 10, and the program stops. The program then computes values of the model parameters (for Cases 2 and 3) and then performs a simulation of resource allocation in which the final outputs are computed. In each of these computations--parameter estimation and simulation--the iterative Newton-Raphson procedure is used to solve the equation $f(\lambda) = 0$; for each iteration the subroutines 'EF' and 'STEP' are called and their results written to the auxiliary output file 'CALCNS'. The Newton-Raphson procedure is terminated when the computed value of $f(\lambda)$ is within a specified criterion of zero; at this point a message is written to the terminal and to the file 'OUTPUT'. If errors are encountered or the number of iterations exceeds the maximum specified in the file 'AUXPA' a message is written to the terminal and the program stops. Finally the final computed parameter values and final output results of the simulation are written to the file 'OUTPUT'.

SUBROUTINES

The subroutines 'EF' and 'STEP' are identical for each of the three cases. They are called for each iteration of the

Newton-Raphson procedure. The main purpose of 'EF' is to evaluate the function $f(\lambda)$ for a given value of λ. 'STEP' then evaluates the derivative $f'(\lambda)$ and computes a new value for λ as follows:

$$\lambda_{t+1} = \lambda_t - f(\lambda_t)/f'(\lambda_t) \quad , \tag{1}$$

where λ_t is the value of λ at iteration t.

For reasons given in Gibbs [3] we are searching for a value of λ greater than unity and the Newton-Raphson procedure can only be guaranteed to converge if the value of λ at each iteration is also greater than unity. Let us now consider the case where, at iteration t, $f(\lambda_t)/f'(\lambda_t) > \lambda_{t-1}$. In this case the application of (1) would lead to a value of λ_{t+1} less than unity. Accordingly, for these cases only, the new value of λ is computed as follows:

$$\lambda_{t+1} = (\lambda_t + 1) \quad . \tag{2}$$

FILE 'AUXPA'

The contents of this file are listed in full in a separate section of the dictionary given in the next section. They consist of auxiliary input data which would not normally need to be altered during a new series of model runs:

- unit numbers for files 'OUTPUT' and 'CALCNS',
- iteration maximum,
- criteria for checking main input data and for convergence of Newton-Raphson procedure,
- initial values for λ and (Cases 2 and 3 only) C and the value of $\overline{B}$.

Suitable values for these data are suggested in Section 6.

FILE 'INIDA'

This contains the main input data for a model run. It includes, for each category:

- the name of the patient category,
- values for admission rate and length of stay--ideal values for Cases 1 and 3, observed mean regional values for Case 2, and

- elasticities for admission rate and length of stay for Cases 2 and 3, values of the α_i and β_i parameters for Case 1.

It also contains the corresponding total admission rate (all categories) and the average length of stay (all categories, weighted by admission rate) to serve as checks on the category-specific data on admission rate and length of stay. Finally it contains the value of B, the supply of bed-days whose allocation is to be simulated, and a name for the model run.

FILE 'OUTPUT'

This is the main output file created by a run of the model. It contains:

- a list of the input data,
- information about each completed Newton-Raphson procedure,
- final computed parameter values (Cases 2 and 3),
- final simulation results--for each category the values of the x_i, the u_i, and their product (bed-days used by category) both in absolute terms and as fractions of the corresponding ideal values, and the corresponding figures for all categories.

FILE 'CALCNS'

This is the auxiliary output file created by a run of the model. It contains the values of the quantities computed in each execution of the subroutines 'EF' and 'STEP' and, for Cases 2 and 3, the values of parameters computed for each iteration of the parameter estimation procedure. This information will usually be of interest only in situations where the user suspects that the program is not working normally, e.g. when the parameter estimation procedure fails to converge.

3. DICTIONARY OF VARIABLES USED IN MAIN PROGRAMS

MAIN MODEL PARAMETERS AND VARIABLES USED IN MAIN PROGRAMS

Symbol	Fortran Name	Description	
		Subscript	
i	I	Patient category	
		Main Parameters and Variables	
X_i	XBIG[I]	Ideal admission rate for category i	Input data in Cases 1 and 3. Computed parameters in Case 2.
U_i	UBIG[I]	Ideal average length of stay for category i	
α_i	ALPHA[I]	Power factor for patient numbers	Input data in Case 1. Computed parameters in Cases 2 and 3.
β_i	BETA[I]	Power factor for length of stay	
$\hat{\gamma}_i$	GAMHAT[I]	Elasticity of admission rate for category i with respect to aggregate bed supply	Input data, used in Cases 2 and 3.
$\hat{\eta}_i$	ETAHAT[I]	Elasticity of length of stay for category i with respect to aggregate bed supply	
$\bar{x}_i$	XBAR[I]	Observed admission rate for category i	Input data, used in Case 2 only.
$\bar{u}_i$	UBAR[I]	Observed mean length of stay for category i	
B	BAVAIL	Number of bed-days available for occupation	

a_i	A[I]	Factor weighting patient numbers relative to length of stay*
x_i	XVAR[I]	Computed admission rate for category i
u_i	UVAR[I]	Computed average length of stay for category i
λ	ZLAMDA	The Lagrange Multiplier

AUXILIARY PARAMETERS FOR MAIN PROGRAM

Fortran Name	Description	
	Read from File 'AUXPA'	
IPRINT	Identifier of intermediate calculations file 'CALCNS'	
KPRINT	Identifier of output file 'OUTPUT'	
ITMAX	Maximum permitted number of iterations for the Newton-Raphson procedure	
IR	Number of patient categories	
CTEST	Criterion for testing accuracy of input data on admission rates and length of stay in file 'INIDA'	
CRIT	Criterion for convergence in solving $f(\lambda) = 0$ by the Newton-Raphson procedure	
ZLAMDA	Starting value of the Lagrange Multiplier (λ)	
BCALIB	Bed supply for which input data is strictly valid ($= \bar{B}$)	Not required for DRAM Case 1.
CRITAL	Criterion for convergence in parameter estimation process	Not required for DRAM Case 1.
C	Starting value for ratio $-\bar{B}/\lambda f'(\lambda)$	Not required for DRAM Case 1.

*For reasons given in Gibbs [3], $a_i = U_i$ in the current version of DRAM and this value is assigned in the main program. As explained in Gibbs [3], it is hoped to develop the model further to represent the allocation of more than one resource in which case a different value will need to be assigned to a_i.

Read from File 'INIDA' ('INIDA' also Contains Input Data as Described on Page 5)	
RUNAME	Label for model run (4 alphameric characters)
XTOTA	Column total for input data on admission rates
UBARA	Column total for input data on length of stay weighted by input data on admission rates
CATEG[I,J]	Part j of label for patient category i (4 alphameric characters); complete label is CATEG[I,1], CATEG[I,2], CATEG[I,3] (12 alphameric characters)

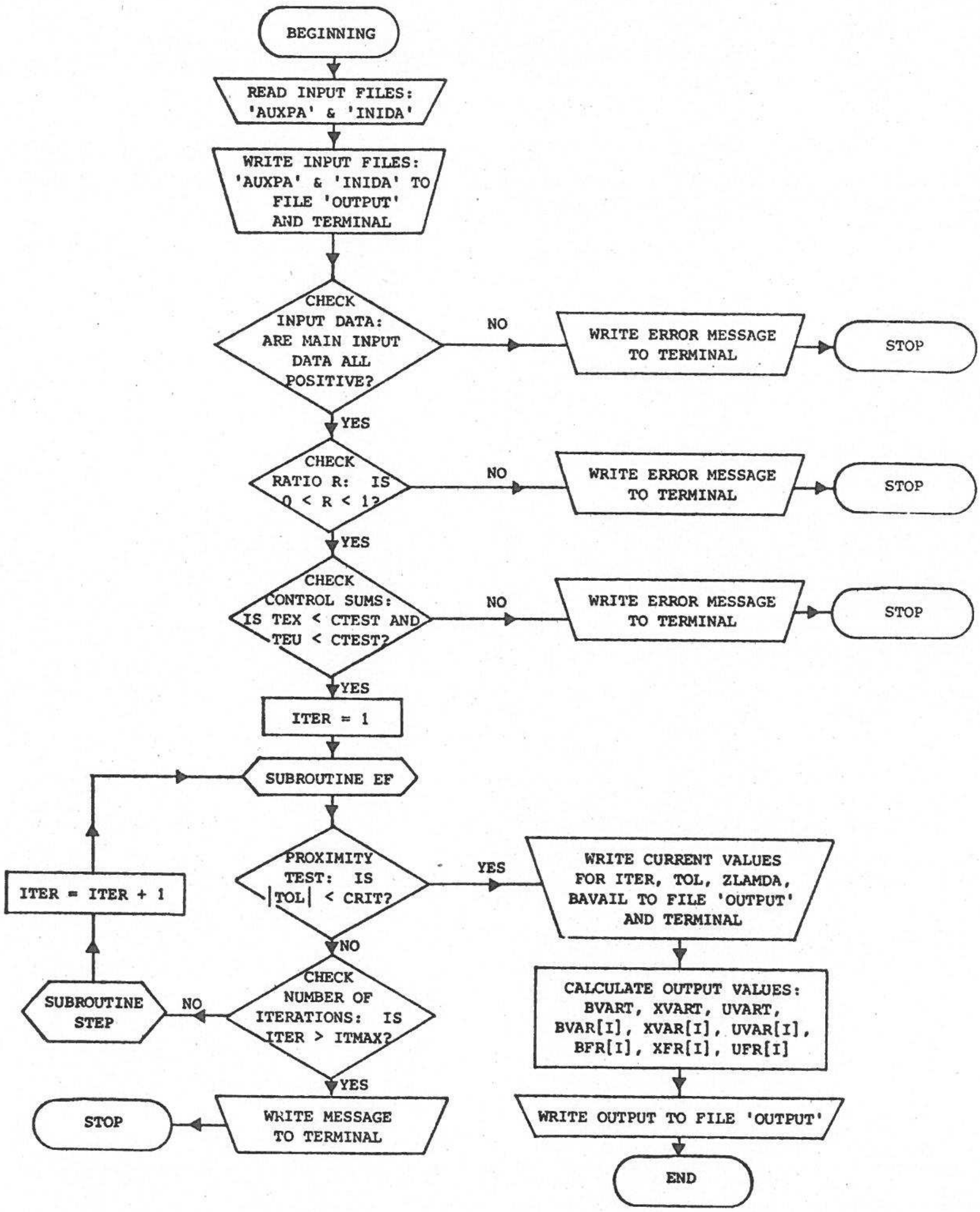
AUXILIARY QUANTITIES COMPUTED IN MAIN PROGRAM

Fortran Name	Description
AMOTOL	Absolute value of TOL
BFR[I]	Computed bed-days used by patients in category i, BVAR[I], as fraction of BNEED[I]
BNEED[I]	Bed-days needed for patients in category i at ideal admission rate and ideal length of stay (= $X_i U_i$)
BNEEDT	Total bed-days needed for patients in all categories at ideal admission rates and ideal lengths of stay (= $\sum_i X_i U_i$)
BTEMP	Temporary storage of the value of BAVAIL
BVAR[I]	Computed bed-days used by patients in category i (= $x_i u_i$)
BVART	Computed total bed-days used--all categories (= $\sum_i X_i U_i$)
IERR	Error count
IHAT	Iteration count for parameter estimation process in Cases 2 and 3
ITER	Iteration count for Newton-Raphson procedure
KERR	Error count
PWR	Power factor (= $-\beta_i/(\beta_i+1)$)
R	Ratio of bed-days available, BAVAIL, to BNEEDT

RED	Quantity used for scaling value of C in second and subsequent iterations of parameter estimation process for Case 3 $(= \sum_i X_i U_i (\hat{\gamma}_i + \hat{\eta}_i))$
SCALE	Quantity used to scale input elasticity data for Case 2 $(= \overline{B}/\sum_i \overline{x}_i \overline{u}_i (\hat{\gamma}_i + \hat{\eta}_i))$
TESTU	Percentage difference between computed and input values for the weighted column totals of the input data on length of stay
TESTX	Percentage difference between computed and input values for the weighted column totals of the input data on admission rates
TEU	Absolute value of TESTU
TEX	Absolute value of TESTX
UBARB	Computed value for weighted column total of input data on length of stay
UFR[I]	Computed length of stay of patients in category i, UVAR[I], as fraction of ideal, UBIG[I] $(= u_i/U_i)$
UVART	Overall computed average length of stay--all categories $(= \sum_i x_i u_i \big/ \sum_i x_i)$
XFR[I]	Computed number of patients in category i admitted to hospital, XVAR[I], as fraction of total number of individuals in category i, XBIG[I] $(= x_i/X_i)$
XTOTB	Computed value for the column total of input data on admission rates
XVART	Overall computed admission rate--all categories $(= \sum_i x_i)$

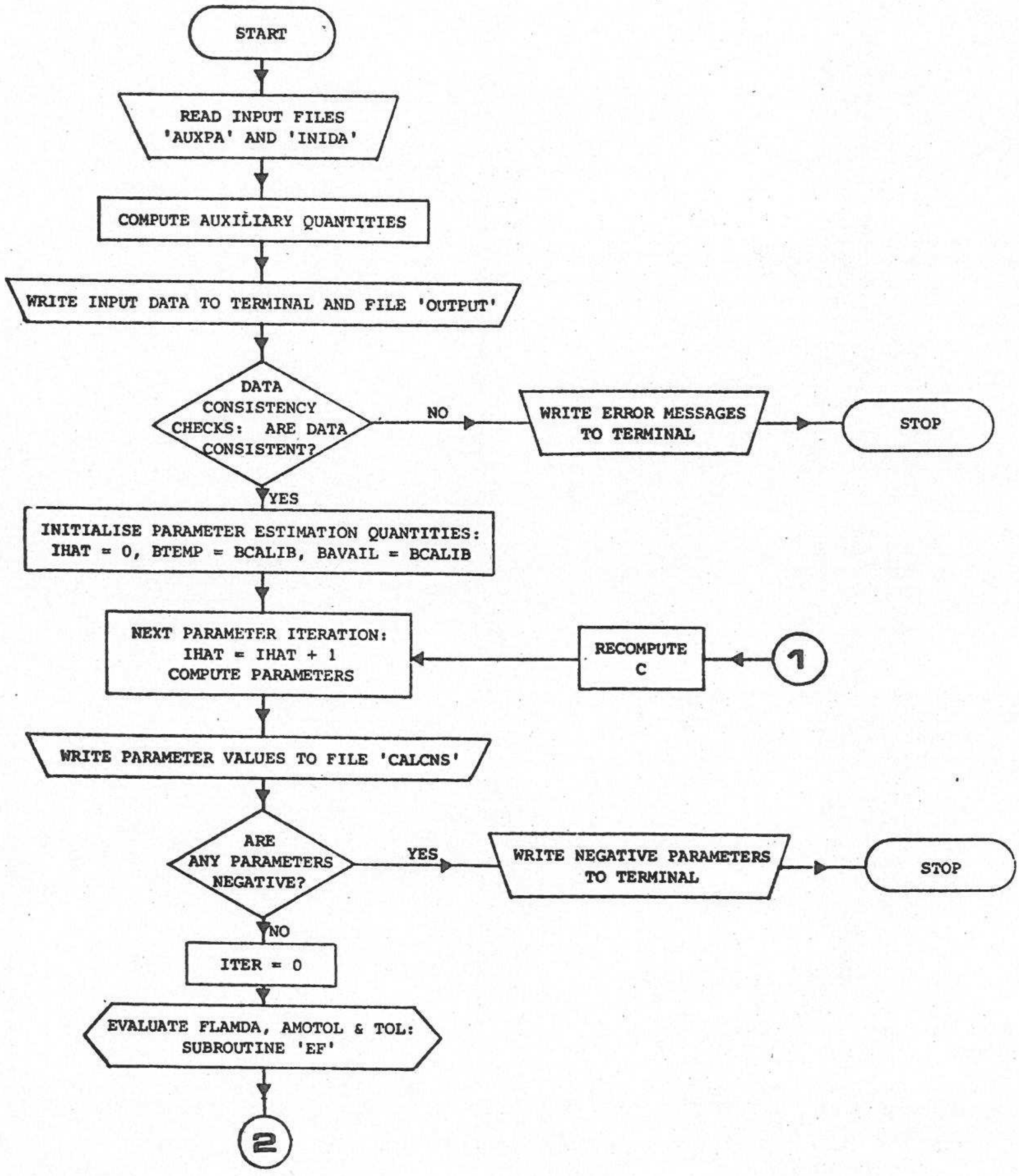
4. MAIN PROGRAMS--FLOW DIAGRAMS AND LISTINGS

In this section flow diagrams and listings are given for the main programs for each of the three cases. The programs for Cases 2 and 3 have a common flow diagram.

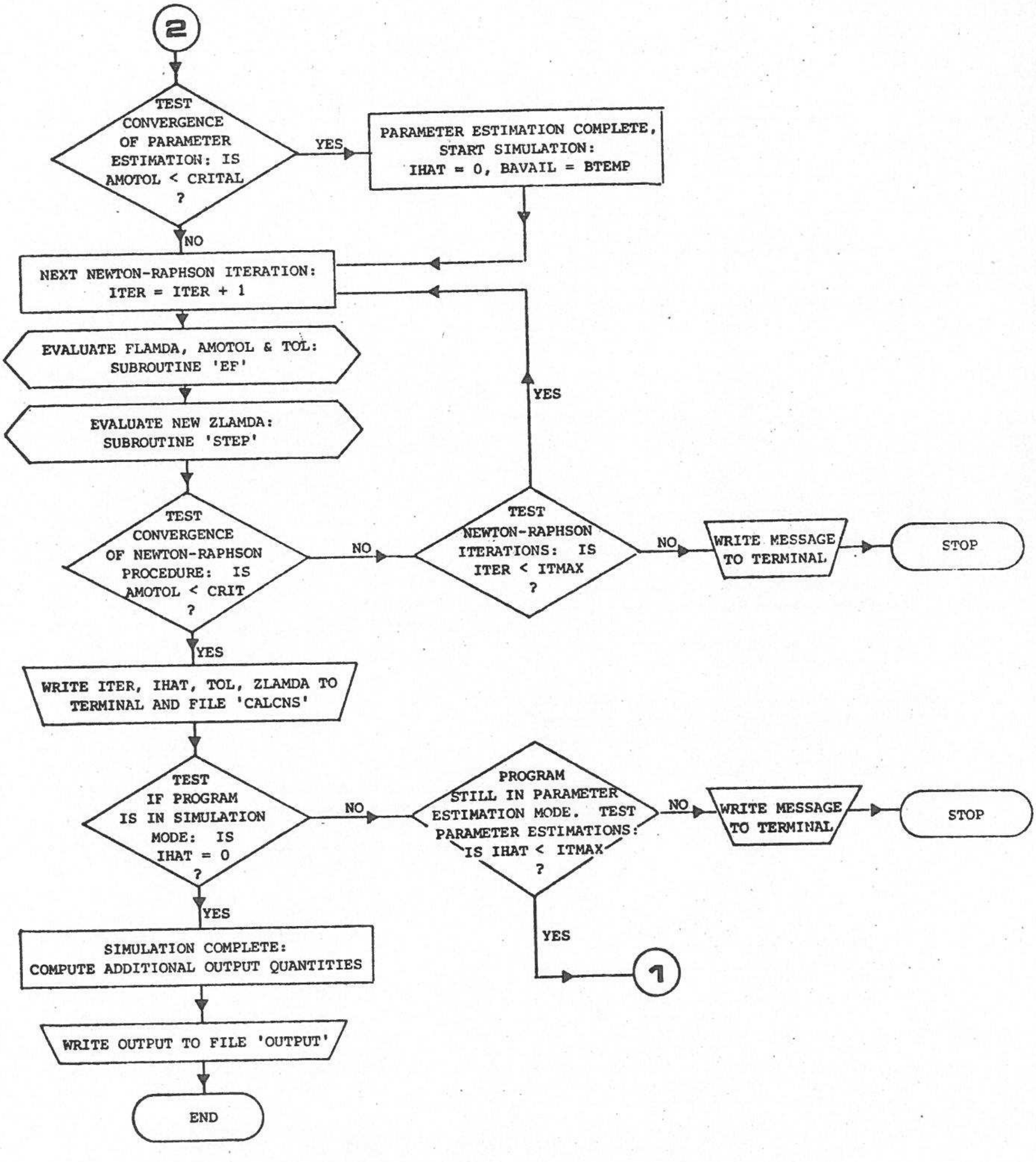


Main Program for Case 1*

*For definitions of variables and parameters used in the diagram see Section 3.



Main Program for Cases 2 and 3



Main Program for Cases 2 and 3 (continued)

MAIN PROGRAM FOR CASE 1

```
c    this is the main programme for the model 'dram' case 1.
c    it simulates the allocation of health care resources
c    using data on power parameters and ideal allocations.

c    it uses 2 subroutines 'ef.f' and 'step.f' iteratively to solve
c    the equation flamda = 0 , using the newton-raphson procedure,
c    to within a criterion, crit.

      dimension xfr(20),ufr(20),bfr(20),categ(20,3)

      common /cb1/ a(20)
      common /cb2/ alpha(20),beta(20)
      common /cb3/ xbig(20),ubig(20)
      common /cb4/ q(20),d(20)
      common /cb5/ xvar(20),uvar(20),bvar(20)
      common /cb6/ bneed(20)


c    read auxiliary parameters from file 5 = 'auxpa'.

      read(5,20) iprint,kprint,itmax,ir,crit,zlamda,ctest
c     write(6,20)iprint,kprint,itmax,ir,crit,zlamda,ctest

c    read input data from file 7 = 'inida'.

      read(7,21) runame,xtota, ubara,bavail,
     -((categ(i,j),j=1,3),xbig(i),ubig(i),
     -alpha(i),beta(i),i=1,ir)
c     write(6,21)runame,xtota,ubara,bavail,
c    -((categ(i,j),j=1,3),xbig(i),ubig(i),alpha(i),
c    -beta(i),i=1,ir)


c    write input data and auxiliary parameters
c    to file 6 (= terminal' ) and file kprint (= 'output' )

      write(6,141) runame,iprint,kprint,itmax,ir,crit,zlamda,ctest
      write(6,30) runame,(i,(categ(i,j),j=1,3),xbig(i),
     -ubig(i),alpha(i),beta(i),i=1,ir)
      write(6,140) xtota,ubara,bavail
      write(6,142)

      write(kprint,141)runame,iprint,kprint,itmax,ir,crit,zlamda,ctest
      write(kprint,30) runame,(i,(categ(i,j),j=1,3),xbig(i),
     -ubig(i),alpha(i),beta(i),i=1,ir)
      write(kprint,140) xtota,ubara,bavail
      write(kprint,142)

c    auxiliary quantities

      bneedt=.0
      ubarb=.0
      xtotb=.0

      do 70 i=1,ir
        xtotb=xtotb+xbig(i)
```

```
          bneed(i)=xbig(i)*ubig(i)
          bneedt=bneedt+bneed(i)
        a(i)=ubig(i)
   70   continue
        ubarb=bneedt/xtotb
        r=bavail/bneedt


c    3 checks of internal consistency of data in file 'inida'.
c    programme stops if any answer is 'yes'.

c    check 1- is any power parameter non-positive

        kerr=0
        do 71 i=1,ir
        ierr=0
        if(alpha(i).le.0.0)ierr=ierr+1
        if(beta(i).le.0.0)ierr=ierr+1
        kerr=kerr+ierr
        if(ierr.gt.0)write(6,22)ierr,i
   71   continue
        if(kerr.gt.0)stop

c    check 2- is r non-positive or greater than unity.

        if(r) 132,132,133
  132   write(6,23) r
        stop
  133   if(r-1.) 134,132,132

c    check 3- is either of the computed values xtotb and ubarb
c    different from the values xtota and ubara
c    by more tnan ctest percent.

  134   testx=((xtota-xtotb)/xtotb)*100.
        testu=((ubara-ubarb)/ubarb)*100.
        tex=abs(testx)
        teu=abs(testu)
        kerr=0
        if(tex.gt.ctest)kerr=1
        if(teu.gt.ctest)kerr=kerr+1
        if(kerr)135,135,136
  136   write(6,24)kerr
        stop

c    initialise iteration count, iter

  135   iter=1


cl   compute flamda and, for all i, xvar(i), uvar(i), d(i),and q(i).

  139   call ef(zlamda,bavail,ir,iter,itmax,flamda,tol,amotol
       -,iprint)
```

```
c   convergence test for newton-raphson procedure.
c   if test suceeds go to c3
c   if test fails proceed to next iteration at c2 unless iterations
c   exceed maximum in which case stop.

      if(amotol-crit) 137,138,138
      if(iter-itmax)138,138,131
  131 write(6,157)
      stop


c2  compute fdash and new lamda and return to cl

  138 call step (zlamda,flamda,ir,iter,fdash,iprint)
      iter=iter+1
      go to 139


c3  write information on final iteration to file 6 (=terminal)
c   and kprint (=output)

  137 write(kprint,126) iter,tol,zlamda,bavail
      write(6,126)iter,tol,zlamda,bavail


c   compute additional output quantities

      bvart=0.0
      xvart=0.0
      do 80 i=1,ir
        xfr(i)=xvar(i)/xbig(i)
        ufr(i)=uvar(i)/ubig(i)
        bfr(i)=xfr(i)*ufr(i)
        bvart=bvart+bvar(i)
        xvart=xvart+xvar(i)
   80 continue
      uvart=bvart/xvart
      xvartf=xvart/xtotb
      uvartf=uvart/ubarb
      bvartf=bvart/bneedt


c   write output

      write(kprint,29) runame
      write(kprint,27)(i,(categ(i,j),j=1,3),xvar(i),xfr(i),
     -uvar(i),ufr(i),bvar(i)
     -,bfr(i),i=1,ir)

      write(kprint,28) xvart,xvartf,uvart,uvartf,bvart,bvartf

   20 format(////4i10///////3f12.1)

   21 format(////1x,a4,8x,3f9.0//////(1x,3a4,4f9.0))

   22 format(1x,i1,' negative values for power parameter for',
```

```
   -i2,' type of patient')

 23 format(' ratio beds available : beds needed = ',f12.2/
   -' no allocation problem !')

 24 format(1x,i1,' errors in xbig, ubig data')

 27 format(//6x,'patient',12x,'actually',9x,'actual aver.',9x,
   -'beddays'/6x,'category',11x,'treated',10x,'length of stay',7x,
   -'actually used'/21x,'(number/fract.)',5x,'(number/fract.)',
   -6x,'(number/fract.)'//20(1x,i2,1x,3a4,5x,f7.1,'/',f7.4,5x,f7.3,
   -'/',f7.4,6x,f7.1,'/',f7.4//))

 28 format(/' all categories',5x,f8.1,'/',f7.4,5x
   -,f7.3,'/',f7.4,5x,f8.1,'/',f7.4/)

 29 format(1h1,15x,'simulation results',6x,'for run ',a4//)

 30 format(//126('*')/20x,'file =inida= for run ',a4
   -//' patient category',6x,'total sick',
   -5x,'ideal length of stay',3x,'alpha',5x,'beta'/
   -23x,'xbig(i)',11x,'ubig(i)',11x,'alpha(i)',3x,'beta(i)'//
   -(20(1x,i2,1x,3a4,5x,f7.1,12x,f7.3,12x,f7.4,3x,f7.4//)))

126 format(//2x,'no. iteration',2x,'tolerance',2x,'lambda',
   -2x,'beddays avail',/,6x,i3,8x,e9.2,1x,f7.3,1x,f7.1)

140 format(13x,'total= ',f8.1,5x,'mean= ',
   -f8.3,7x,'bed-days available=',f7.1/126('*'))

141 format(////126('*')/20x,'file=auxpa= for run ',a4//
   -'  identifiers for files    max no. of    no.of pat.',4x,
   -'newton-raphson     initial value    data'/
   -'  =calcns= and =output=     iterations     categories',4x,
   -'conv. criterion    of lambda',8x,'test'/
   -'    iprint',7x,'kprint',7x,'itmax',11x,'ir',12x,'crit',13x,
   -'zlamda',9x,'ctest'//
   -6x,i1,12x,i1,10x,i3,12x,i2,9x,f9.7,10x,f7.3,6x,f7.1//)

142 format(126('*'))

157 format(/'number of newton-raphson iterations exceeds maximum')

      stop
      end
```

MAIN PROGRAM FOR CASE 2

```
c    this is the main programme for the model 'dram' mark 1 case 2.
c    it simulates the allocation of a health care resource
c    using  data on elasticities and current mean allocations.


c    it uses 2 subroutines 'ef.f' and 'step.f' iteratively to solve
c    the equation flamda = Ø , using the newton-raphson procedure,
c    to within a given criterion.

        dimension xfr(2Ø),ufr(2Ø),bfr(2Ø),categ(2Ø,3),gamhat(2Ø),
       -etahat(2Ø),xbar(2Ø),ubar(2Ø)

        common /cb1/ a(2Ø)
        common /cb2/ alpha(2Ø),beta(2Ø)
        common /cb3/ xbig(2Ø),ubig(2Ø)
        common /cb4/ q(2Ø),d(2Ø)
        common /cb5/ xvar(2Ø),uvar(2Ø),bvar(2Ø)
        common /cb6/ bneed(2Ø)


c    read auxiliary parameters from file 5 = 'auxpa'.

        read(5,2Ø) iprint,kprint,itmax,ir,zlamda,ctest,crit,crital,c,
       -bcalib
c         write(6,2Ø)iprint,kprint,itmax,ir,zlamda,ctest,crit,crital,c,
c        -bcalib

c    read input data from file 7 = 'inida'.

        read(7,21) runame,xtota, ubara,bavail,
       -((categ(i,j),j=1,3),xbar(i),ubar(i),gamhat(i),etahat(i),i=1,ir)
c         write(6,21)runame,xtota,ubara,bavail,
c        -((categ(i,j),j=1,3),xbar(i),ubar(i),gamhat(i),etahat(i),i=1,ir)

c    calculate auxiliary quantities.

c    the variables bneed(i) are used here as temporary stores
c    for the products xbar(i)*ubar(i) and bneedt for the sum
c    of products. later in the program they take the values
c    of the products xbig(i)*ubig(i).

        bneedt=Ø.Ø
        ubarb=Ø.Ø
        xtotb=Ø.Ø
        scale=Ø.Ø

        do 7Ø i=1,ir
          xtotb=xtotb+xbar(i)
          bneed(i)=xbar(i)*ubar(i)
          scale=scale+(bneed(i)*(gamhat(i)+etahat(i)))
          bneedt=bneedt+bneed(i)
     7Ø continue
        ubarb=bneedt/xtotb
        r=bavail/bneedt

c    write input data and auxiliary parameters
```

```
c    to file 6 (= terminal' ) and file kprint (= 'output' )

       write(6,141) runame,iprint,kprint,itmax,ir,zlamda
      -,ctest,crit,crital,c,bcalib
       write(6,30) runame,(i,(categ(i,j),j=1,3),xbar(i),
      -ubar(i),gamhat(i),etahat(i),i=1,ir)
       write(6,140) xtotb,ubarb,bavail
       write(6,142)

       write(kprint,141)runame,iprint,kprint,itmax,ir,zlamda,
      -ctest,crit,crital,c,bcalib
       write(kprint,30) runame,(i,(categ(i,j),j=1,3),xbar(i),
      -ubar(i),gamhat(i),etahat(i),i=1,ir)
       write(kprint,140) xtotb,ubarb,bavail
       write(kprint,142)

c    scale the elasticity data and write scale to terminal.

       scale=bcalib/scale
       do 72 i=1,ir
       gamhat(i)=scale*gamhat(i)
       etahat(i)=scale*etahat(i)
    72 continue
       write(6,551)scale
       write(kprint,551)scale


c    2 checks of internal consistency of data .
c    programme stops if any answer is 'yes'.

c    check 1- are any elasticity data  non-positive?

       kerr=0
       do 71 i=1,ir
       ierr=0
       if(gamhat(i).le.0.0)ierr=1
       if(etahat(i).le.0.0)ierr=ierr+1
       kerr=kerr+ierr
       if(ierr.gt.0)write(6,22)ierr,i
    71 continue
       if(kerr.gt.0)stop

c    check 2- do either of the computed xtotb and ubarb differ
c    from the input xtota and ubara by more than ctest % ?

   134 testx=((xtota-xtotb)/xtotb)*100.
       testu=((ubara-ubarb)/ubarb)*100.
       tex=abs(testx)
       teu=abs(testu)
       kerr=0
       if(tex.gt.ctest)kerr=1
       if(teu.gt.ctest)kerr=kerr+1
       if(kerr)135,135,136
   136 write(6,24)kerr
       stop
```

```
c   initialise parameter iteration count and set bavail=bcalib.

  135 ihat=0
      btemp=bavail
      bavail=bcalib

ca  perform iteration of parameter estimation procedure

  500 ihat=ihat+1

c   calculate parameters`xbig(i), ubig(i), alpha(i), beta(i).
c   write to file iprint. stop if any are not positive.

      kerr=0
      do 502 i=1,ir
      ierr=0
      if(ihat-1)543,543,542
  542 c=(-1.0*bavail)/(zlamda*fdash)
  543 beta(i)=c/etahat(i)-1.0
      pwr=-(beta(i)/(beta(i)+1.0))
      alpha(i)=((c*beta(i))/((beta(i)+1.0-zlamda**pwr)*gamhat(i)))-1.0
      xbig(i)=xbar(i)*(((beta(i)+1.0)*(zlamda**(-pwr))-1.0)/
     -beta(i))**(1.0/(1.0+alpha(i)))
      ubig(i)=ubar(i)*zlamda**(1.0/(1.0+beta(i)))
      a(i)=ubig(i)
      bneed(i)=xbig(i)*ubig(i)
      if(alpha(i).le.0.0)ierr=ierr+1
      if(beta(i).le.0.0)ierr=ierr+1
      if(xbig(i).le.0.0)ierr=ierr+1
      if(ubig(i).le.0.0)ierr=ierr+1
      if(ierr)502,502,501
  501 kerr=kerr+1
      write(6,553) ierr,i
      write(6,552)ihat,i,alpha(i),beta(i),xbig(i),ubig(i)
  502 continue
      write(iprint,552)ihat,(i,alpha(i),beta(i),xbig(i),ubig(i),i=1,ir)
      if(kerr.gt.0)stop


c   initialise newton-raphson iteration count ,iter, compute flamda.

      iter=0
      if(ihat-1)540,540,539
  539 call ef(zlamda,bavail,ir,iter,itmax,flamda,tol,amotol,iprint)

c   test convergence of parameter estimation . if test succeeds
c   estimation is complete- commence simulation: set ihat=0, restore
c   bavail to input value & solve flamda=0 by newto-raphson.
c   if test fails perform newton-raphson and test again.

      if(amotol-crital)530,530,540
  530 ihat=0
      bavail=btemp
```

```
c1  perform iteration of newton-raphson procedure
c   compute flamda and, for all i, xvar(i),uvar(i),d(i),and q(i)

  540 iter=iter+1
      call ef(zlamda,bavail,ir,iter,itmax,flamda,tol,amotol,iprint)

c   compute fdash and new lamda

      call step(zlamda,flamda,ir,iter,fdash,iprint)

c   convergence test for newton-raphson procedure.
c   if test suceeds go to c2.
c   if test fails proceed to next iteration at c1, unless
c   iterations exceed maximum in which case stop.

      if(amotol-crit) 537,538,538
  538 if(iter-itmax)540,540,541
  541 write(6,557)
      stop


c2  write information on final newton-raphson iteration to file 6
c   and kprint (=output).if process is in simulation mode ( ihat=0)
c   simulation is now completed ; go to c3.

  537 write(kprint,126) ihat,iter,tol,zlamda
      write(6,126)      ihat,iter,tol,zlamda
      if(ihat)533,533,532


c   if no. of alpha iterations exceeds maximum stop,
c   if not perform next alpha iteration at ca

  532 if(ihat-itmax)500,500,531
  531 write(6,554)
      stop


c3  compute additional output quantities

  533 xvart=0.0
      bvart=0.0
      xtotb=0.0
      bneedt=0.0
      do 80 i=1,ir
         xfr(i)=xvar(i)/xbig(i)
         ufr(i)=uvar(i)/ubig(i)
         bfr(i)=xfr(i)*ufr(i)
         bvart=bvart+bvar(i)
         xvart=xvart+xvar(i)
         xtotb=xtotb+xbig(i)
         bneedt=bneedt+xbig(i)*ubig(i)
   80 continue
      uvart=bvart/xvart
      xvartf=xvart/xtotb
      ubarb=bneedt/xtotb
```

```
      uvartf=uvart/ubarb
      bvartf=bvart/bneedt

c   write output

      write(kprint,555) runame,(i,(categ(i,j),j=1,3),
     -alpha(i),beta(i),xbig(i),ubig(i),i=1,ir)
      write(kprint,29) runame
      write(kprint,27)(i,(categ(i,j),j=1,3),xvar(i),xfr(i),
     -uvar(i),ufr(i),bvar(i)
     -,bfr(i),i=1,ir)

      write(kprint,28) xvart,xvartf,uvart,uvartf,bvart,bvartf

   20 format(////4i10///////5f12.6,f12.3)

   21 format(////1x,a4,8x,3f10.5//////(1x,3a4,4f10.7))

   22 format(1x,i1,' negative values in elasticity data for ',
     -' patient category ',i2)

   24 format(1x,i1,' errors in xbig, ubig data')

   27 format(//6x,'patient',12x,'actually',9x,'actual aver.',9x,
     -'beddays'/6x,'category',11x,'treated',10x,'length of stay',7x,
     -'actually used'/21x,'(number/fract.)',5x,'(number/fract.)',
     -6x,'(number/fract.)'//20(1x,i2,1x,3a4,5x,f7.1,'/',f7.4,5x,f7.3,
     -'/',f7.4,6x,f7.1,'/',f7.4//))

   28 format(/' all categories',5x,f8.1,'/',f7.4,5x
     -,f7.3,'/',f7.4,5x,f8.1,'/',f7.4/)

   29 format(1h1,15x,'simulation results',6x,'for run ',a4//)

   30 format(//126('*')/20x,'file =inida= for run ',a4
     -//' patient category',3x,'mean admissions',
     -3x,'mean length of stay',4x,'elasticities'/
     -23x,'xbar(i)',11x,'ubar(i)',11x,'gamhat(i)',3x,'eta(i)'//
     -(20(1x,i2,1x,3a4,5x,f7.1,12x,f7.3,10x,f10.7,4x,f10.7//)))

  126 format(//' parameter iteration ',i2,' completed with ',i2,
     -' newton-raphson iterations'/' tolerance = ',e9.2,'
     -'     lamda = ',f7.3)
  140 format(13x,'total= ',f8.1,5x,'mean= ',
     -f8.3,7x,'bed-days available=',f7.1/126('*'))

  141 format(////126('*')/20x,'file=auxpa= for run ',a4//
     -'  identifiers for files     max no. of    no.of pat.',4x,
     -'initial value'/
     -'  =calcns= and =output=     iterations    categories',4x,
     -'of lamda'/
     -'    iprint',7x,'kprint',7x,'itmax',11x,'ir',11x,'zlamda'//
     -6x,i1,12x,i1,10x,i3,12x,i2,9x,f7.3///
     -5x,'data',7x,'newton-raphson',5x,'alpha',9x,'arbitrary',5x,
     -'calibration'/
```

```
   -5x,'test',7x,'conv. criterion',3x,'criterion',6x,'constant',
   -6x,'bed point'/
   -5x,'ctest',8x,'crit',12x,'crital',12x,'c',12x,'bcalib'//
   -2x,f10.5,5x,f10.7,5x,f10.7,5x,f10.5,5x,f10.2//
   -126('*'))

142 format(1h1)

551 format(/'elasticity data scaled by ',f8.5)

552 format(//126('*')//
  -' clculated parameters at iteration ',i2//
   -'  i',3x,'alpha(i)',12x,'beta(i)',14x,'xbig(i)',14x,'ubig(i)'//
   -(1x,i2,1x,f12.6,8x,f12.6,8x,f12.6,8x,f12.6))

553 format(/i2,' negative values of calculated parameters
   - for category ',i2)

554 foma(/'nmbr f pramte tertion exceeds maximum')

555 format(1h1,5x,'calculated parameter values for run ',a4///
   -2x,'i',4x,'category',7x,'alpha(i)',4x,'beta(i)',5x,
   -'xbig(i)',5x,'ubig(i)'//
   -(1x,i2,1x,3a4,5x,f7.3,5x,f7.3,5x,f7.3,5x,f7.3//))

557 format(/'number of newton-raphson iterations exceeds aximum')

      stop
      end
```

MAIN PROGRAM FOR CASE 3

```
c    this is the main programme for the model 'dram' mark 1 case 3.
c    it simulates the allocation of a health care resource
c    using data on elasticities and ideal allocations.


c    it uses 2 subroutines 'ef.f' and 'step.f' iteratively to solve
c    the equation flamda = 0 , using the newton-raphson procedure,
c    to within a given criterion.

        dimension xfr(20),ufr(20),bfr(20),categ(20,3),gamhat(20),
       -etahat(20)

        common /cb1/ a(20)
        common /cb2/ alpha(20),beta(20)
        common /cb3/ xbig(20),ubig(20)
        common /cb4/ q(20),d(20)
        common /cb5/ xvar(20),uvar(20),bvar(20)
        common /cb6/ bneed(20)


c    read auxiliary parameters from file 5 = 'auxpa'.

        read(5,20) iprint,kprint,itmax,ir,zlamda,ctest,crit,crital,c,
       -bcalib
c       write(6,20)iprint,kprint,itmax,ir,zlamda,ctest,crit,crital,c,
c      -bcalib

c    read input data from file 7 = 'inida'.

        read(7,21) runame,xtota, ubara,bavail,
       -((categ(i,j),j=1,3),xbig(i),ubig(i),gamhat(i),etahat(i),i=1,ir)
c       write(6,21)runame,xtota,ubara,bavail,
c      -((categ(i,j),j=1,3),xbig(i),ubig(i),gamhat(i),etahat(i),i=1,ir)


c    calculate auxiliary quantities.

        bneedt=0.0
        ubarb=0.0
        xtotb=0.0

        do 70 i=1,ir
          xtotb=xtotb+xbig(i)
          bneed(i)=xbig(i)*ubig(i)
          bneedt=bneedt+bneed(i)
c        beta(i)=c/etahat(i)-1.0
          a(i)=ubig(i)
     70 continue
        ubarb=bneedt/xtotb
        r=bavail/bneedt


c    write input data and auxiliary parameters
c    to file 6 (= terminal' ) and file kprint (= 'output' )

        write(6,141) runame,iprint,kprint,itmax,ir,zlamda
```

```
     -,ctest,crit,crital,c,bcalib
      write(6,30) runame,(i,(categ(i,j),j=1,3),xbig(i),
     -ubig(i),gamhat(i),etahat(i),i=1,ir)
      write(6,140) xtotb,ubarb,bavail
      write(6,142)

      write(kprint,141)runame,iprint,kprint,itmax,ir,zlamda,
     -ctest,crit,crital,c,bcalib
      write(kprint,30) runame,(i,(categ(i,j),j=1,3),xbig(i),
     -ubig(i),gamhat(i),etahat(i),i=1,ir)
      write(kprint,140) xtotb,ubarb,bavail
      write(kprint,142)

c   3 checks of internal consistency of data .
c   programme stops if any answer is 'yes'.

c   check 1- are any elasticity data non-positive?

      kerr=0
      do 71 i=1,ir
      ierr=0
      if(gamhat(i).le.0.0)ierr=1
      if(etahat(i).le.0.0)ierr=ierr+1
      kerr=kerr+ierr
      if(ierr.gt.0)write(6,22)ierr,i
   71 continue
      if(kerr.gt.0)stop

c   check 2- is r non-positive or greater than unity ?

      if(r) 132,132,133
  132 write(6,23) r
      stop
  133 if(r-1.) 134,132,132

c   check 3- do either of the computed xtotb and ubarb differ
c   from the input xtota and ubara by more than ctest % ?

  134 testx=((xtota-xtotb)/xtotb)*100.
      testu=((ubara-ubarb)/ubarb)*100.
      tex=abs(testx)
      teu=abs(testu)
      kerr=0
      if(tex.gt.ctest)kerr=1
      if(teu.gt.ctest)kerr=kerr+1
      if(kerr)135,135,136
  136 write(6,24)kerr
      stop


c   initialise parameter iteration count and set bavail=bcalib.

  135 ihat=0
      btemp=bavail
```

```
      bavail=bcalib

ca    perform iteration of parameter estimation procedure

  500 ihat=ihat+1

c     calculate parameters alpha(i) and beta(i) and write them to
c     file inprint and stop if any are not positive.

      red=0.0
      do 544 i=1,ir
      red=red+(xvar(i)*uvar(i)*(gamhat(i)+etahat(i)))
  544 continue
      kerr=0
      do 502 i=1,ir
      ierr=0
      if(inat-1)543,543,542
  542 c=(-1.0*red)/(zlamda*fdash)
  543 beta(i)=c/etahat(i)-1.0
      pwr=-(beta(i)/(beta(i)+1.0))
      alpha(i)=((c*beta(i))/((beta(i)+1.0-zlamda**pwr)*gamhat(i)))-1.0
      if(beta(i).le.0.0)ierr=ierr+1
      if(alpha(i).le.0.0)ierr=ierr+1
      if(ierr)502,502,501
  501 kerr=kerr+ierrr
      write(6,553) i,alpha(i),beta(i)
  502 continue
      write(iprint,552) ihat,(i,alpha(i),beta(i),i=1,ir)
      if(kerr.gt.0)stop


c   initialise newton-raphson iteration count ,iter, compute flamda.

      iter=0
      if(ihat-1)540,540,539
  539 call ef(zlamda,bavail,ir,iter,itmax,flamda,tol,amotol,iprint)

c   test convergence of parameter estimation. if test succeeds
c   estimation is complete- commence simulation: set ihat=0, restore
c   bavail to input value & solve flamda=0 by newton-raphson.
c   if test fails perform newton-raphson and test again.

      if(amotol-crital)530,530,540
  530 ihat=0
      bavail=btemp

cl  perform iteration of newton-raphson procedure
c   compute flamda and, for all i, xvar(i),uvar(i),d(i),and q(i)

  540 iter=iter+1
      call ef(zlamda,bavail,ir,iter,itmax,flamda,tol,amotol,iprint)

c   compute fdash and new lamda

      call step(zlamda,flamda,ir,iter,fdash,iprint)
```

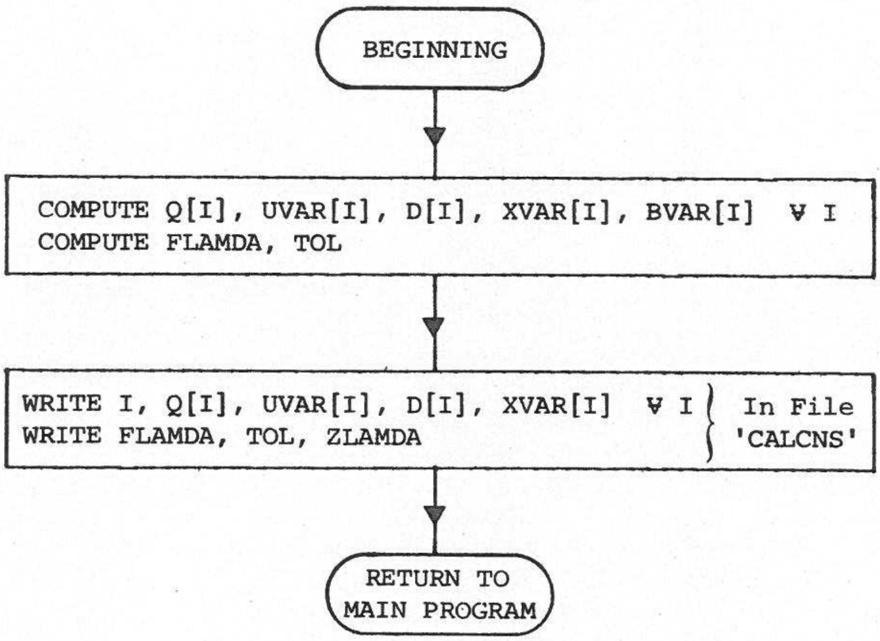
```
c   convergence test for newton-raphson procedure.
c   if test suceeds go to c2.
c   if test fails proceed to next iteration at cl, unless
c   iterations exceed maximum in which case stop.

       if(amotol-crit) 537,538,538
   538 if(iter-itmax)540,540,541
   541 write(6,557)
       stop


c2  write information on final newton-raphson iteration to file 6
c   and kprint. if process is in simulation mode ( ihat=0)
c   simulation is now completed ; go to c3.

   537 write(kprint,126) ihat,iter,tol,zlamda
       write(6,126)      ihat,iter,tol,zlamda
       if(ihat)533,533,532


c   if no. of alpna iterations exceeds maximum stop,
c   if not perform next alpha iteration at ca

   532 if(ihat-itmax)500,500,531
   531 write(6,554)
       stop


c3  compute additional output quantities

   533 xvart=0.0
       bvart=0.0
       do 80 i=1,ir
          xfr(i)=xvar(i)/xbig(i)
          ufr(i)=uvar(i)/ubig(i)
          bfr(i)=xfr(i)*ufr(i)
          bvart=bvart+bvar(i)
          xvart=xvart+xvar(i)
    80 continue
       uvart=bvart/xvart
       xvartf=xvart/xtotb
       uvartf=uvart/ubarb
       bvartf=bvart/bneedt


c   write output

       write(kprint,555) runame,(i,(categ(i,j),j=1,3),
      -alpha(i),beta(i),i=1,ir)
       write(kprint,29) runame
       write(kprint,27)(i,(categ(i,j),j=1,3),xvar(i),xfr(i),
      -uvar(i),ufr(i),bvar(i)
      -,bfr(i),i=1,ir)

       write(kprint,28) xvart,xvartf,uvart,uvartf,bvart,bvartf
```

```
 20 format(////4i10///////5f12.6,f12.3)

 21 format(////1x,a4,8x,3f9.2//////(1x,3a4,4f9.3))

 22 format(1x,i1,' negative values in elasticity data for ',
   -' patient category ',i2)

 23 format(' ratio beds available : beds needed = ',f12.2/
   -' no allocation problem !')

 24 format(1x,i1,' errors in xbig, ubig data')

 27 format(//6x,'patient',12x,'actually',9x,'actual aver.',9x,
   -'beddays'/6x,'category',11x,'treated',10x,'length of stay',7x,
   -'actually used'/21x,'(number/fract.)',5x,'(number/fract.)',
   -6x,'(number/fract.)'//20(1x,i2,1x,3a4,5x,f7.1,'/',f7.4,5x,f7.3,
   -'/',f7.4,6x,f7.1,'/',f7.4//))

 28 format(/' all categories',5x,f8.1,'/',f7.4,5x
   -,f7.3,'/',f7.4,5x,f8.1,'/',f7.4/)

 29 format(1h1,15x,'simulation results',6x,'for run ',a4//)

 30 format(//126('*')/20x,'file =inida= for run ',a4
   -//' patient category',6x,'total sick',
   -5x,'ideal length of stay',4x,'elasticities'/
   -23x,'xbig(i)',11x,'ubig(i)',11x,'gamhat(i)',3x,'eta(i)'//
   -(20(1x,i2,1x,3a4,5x,f7.1,12x,f7.3,12x,f7.4,3x,f7.4//)))

126 format(//' parameter iteration ',i2,' completed with ',i2,' newt',
   -'on-raphson iterations'/' tolerance = ',e9.2,'   lamda = ',f7.3)

140 format(13x,'total= ',f8.1,5x,'mean= ',
   -f8.3,7x,'bed-days available=',f7.1/126('*'))

141 format(////126('*')/20x,'file=auxpa= for run ',a4//
   -'  identifiers for files    max no. of    no.of pat.',4x,
   -'initial value'/
   -'  =calcns= and =output=    iterations    categories',4x,
   -'of lamda'/
   -'   iprint',7x,'kprint',7x,'itmax',11x,'ir',11x,'zlamda'//
   -6x,i1,12x,i1,10x,i3,12x,i2,9x,f7.3///
   -5x,'data',7x,'newton-raphson',5x,'alpha',9x,'arbitrary',5x,
   -'calibration'/
   -5x,'test',7x,'conv. criterion',3x,'criterion',6x,'constant',
   -6x,'bed point'/
   -5x,'ctest',8x,'crit',12x,'crital',12x,'c',12x,'bcalib'//
   -2x,f10.5,5x,f10.7,5x,f10.7,5x,f10.5,5x,f10.2//
   -126('*'))

142 format(1h1)


552 format(//126('*')//' parameter iteration number ',i2//
   ' i cacuatd aphai)  calculated beta(i)'//
   -(1x,i2,2x,f12.6,2x,f12.6))
```

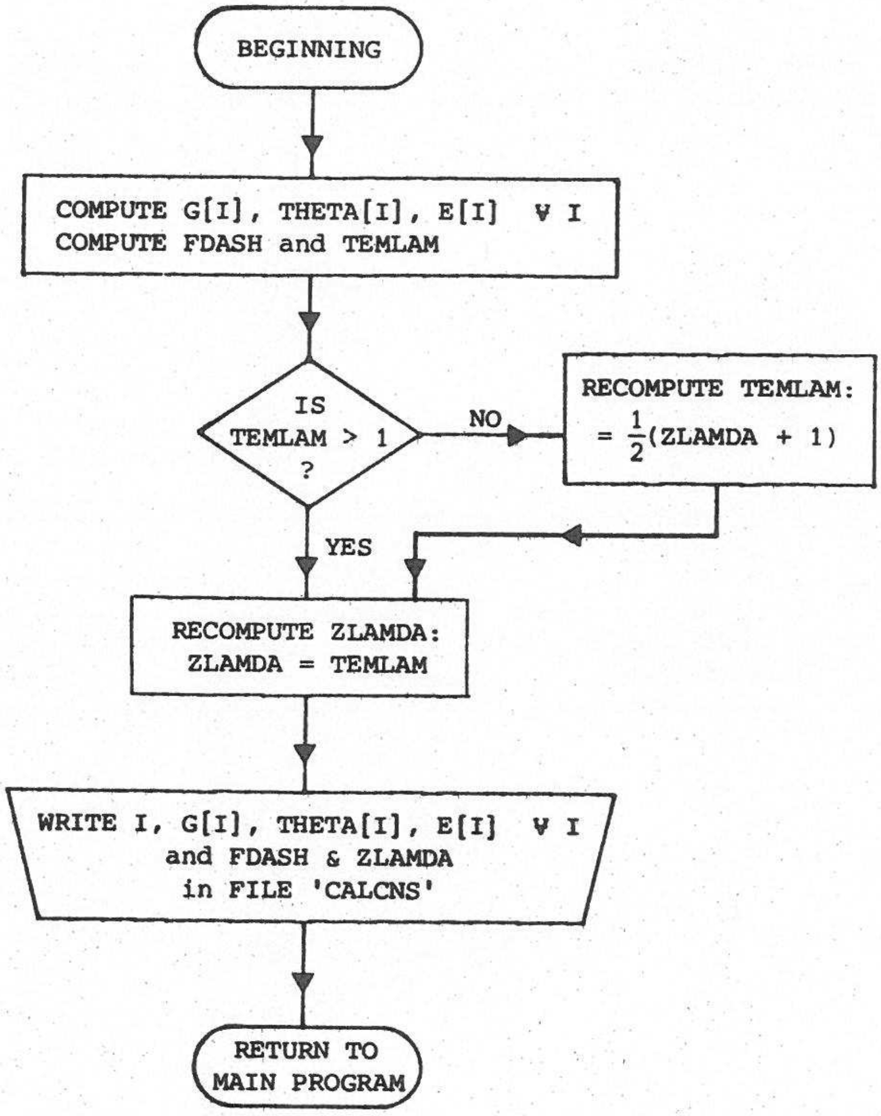
```
553 format(/' negative calculated parameter for category ',i2/
   -'alpha(i)= ',f12.6,'   beta(i)= ',f12.6)

554 format(/'number of parameter iterations exceeds maximum')

555 format(1h1,5x,'calculated parameter values for run ',a4///
   -2x,'i',4x,'category',7x,'alpha(i)',4x,'beta(i)'//
   -(1x,i2,1x,3a4,5x,f7.3,5x,f7.3//))

557 format(/'number of newton-raphson iterations exceeds maximum')

      stop
      end
```

5. DESCRIPTION OF SUBROUTINES

Flow diagrams, dictionaries for the computed quantities and listings are given below for the two subroutines 'EF' and 'STEP'.



Subroutine 'EF'



Subroutine 'STEP'

QUANTITIES COMPUTED IN SUBROUTINE 'EF'

AMOTOL = Absolute value of TOL

$$D[I] = d_i = \frac{U_i}{a_i\beta_i}\left[(\beta_i + 1)\lambda^{\beta_i/(\beta_i+1)} - 1\right]$$

$$FLAMDA = f(\lambda) = \sum_i x_i u_i - B$$

$$TOL = f(\lambda)/B$$

$$Q[I] = q_i = \lambda^{1/(\beta_i+1)}$$

$$UVAR[I] = u_i = U_i/q_i$$

$$XVAR[I] = x_i = X_i\Big/\left[d_i^{1/(\alpha_i+1)}\right]$$

QUANTITIES COMPUTED IN SUBROUTINE 'STEP'

$$E[I] = e_i = X_i U_i \Theta_i\Big/\left\{(\alpha_i + 1)g_i^{(\alpha_i+2)/(\alpha_i+1)}\right\}$$

$$FDASH = f'(\lambda) = -\sum_i e_i$$

$$G[I] = g_i = d_i q_i^{\alpha_i+1}$$

$$TEMLAM = \lambda - f(\lambda)/f'(\lambda)$$

$$THETA[I] = \Theta_i = \frac{U_i}{a_i\beta_i}\left\{(\alpha_i+\beta_i+1)\lambda^{\alpha_i/(\beta_i+1)} - \frac{\alpha_i+1}{\beta_i+1}\lambda^{(\alpha_i-\beta_i)/(\beta_i+1)}\right\}$$

$$ZLAMDA = \begin{cases} TEMLAM & \text{if } TEMLAM > 1 \\ \frac{1}{2}(ZLAMDA + 1) & \text{if } TEMLAM \le 1 \end{cases}$$

SUBROUTINE 'EF'

```
c   this subroutine computes flamda and amotol
c   and, for all i, xvar(i) uvar(i) d(i) and q(i)

        common /cbl/ a(2Ø)
        common /cb2/ alpha(2Ø),beta(2Ø)
        common /cb3/ xbig(2Ø),ubig(2Ø)
        common /cb4/ q(2Ø),d(2Ø)
        common /cb5/ xvar(2Ø),uvar(2Ø),bvar(2Ø)


        flamda=-bavail
      do 51 i=l,ir
        q(i)=zlamda**((1.Ø/(1.Ø+beta(i))))
        uvar(i)=ubig(i)/q(i)
        d(i)=(ubig(i)/(a(i)*beta(i)))*((beta(i)+1.Ø)*(zlamda**(beta(i)
     -/(beta(i)+1.Ø)))-1.Ø)
        xvar(i)=xbig(i)/(d(i)**(1.Ø/(alpha(i)+1.Ø)))
        bvar(i)=xvar(i)*uvar(i)
        flamda=flamda+bvar(i)
   51 continue
        tol=flamda/bavail
        amotol=abs(tol)


c   write computations on file iprint (= 'calcns')

        write(iprint,2) iter
        write(iprint,3) (i,q(i),uvar(i),d(i),xvar(i),i=l,ir)
        write(iprint,4) flamda,tol,zlamda

    2   format(///1Øx,6Ø('*')/' newton-raphson iteration ',i3//
     -'  category',1Øx,'q(i)',13x,'uvar(i)',1Øx,'d(i)',13x,'xvar(i)')

    3   format(4x,i2,llx,e12.5,5x,e12.5,5x,e12.5,5x,e12.5)

    4   format(//' flamda=',e12.5,5x,'tol=',e12.5,3x,'lamda=',e12.5)

        return
        end
```

SUBROUTINE 'STEP'

```
c  this subroutine computes fdash and new value of zlamda

        common /cbl/ a(20)
        common/cb2/ alpha(20),beta(20)
        common /cb3/ xbig(20),ubig(20)
        common/cb4/ q(20),d(20)
        common/cb6/ bneed(20)

        dimension g(20),theta(20),e(20)


c   compute fdash

        fdash=0.0
       do 61 i=1,ir
        g(i)=d(i)*(q(i)**(alpha(i)+1.0))
        theta(i)=(q(i)**alpha(i))*(alpha(i)+beta(i)+1.0-((alpha(i)+1.0)
     -/(beta(i)+1.0)/(q(i)**beta(i))))*ubig(i)/(a(i)*beta(i))
        e(i)=bneed(i)*theta(i)/((alpha(i)+1.0)*(g(i)**((alpha(i)+2.0)/(
     -alpha(i)+1.0))))
        fdash=fdash-e(i)
   61  continue


c   compute new zlamda

        temlam=zlamda-(flamda/fdash)
        if(temlam-1.) 116,116,117
  116   temlam=0.5+(0.5*zlamda)
  117   zlamda=temlam


c   write computations on file iprint (= 'calcns')

  111   write(iprint,11)
        write(iprint,12)(i,g(i),theta(i),e(i),i=1,ir)
        write(iprint,13)fdash,zlamda
  112   continue

   11   format(///'  category',6x,'g(i)',13x,'theta(i)',9x,'e(i)')

   12   format(5x,i2,5x,e12.5,5x,e12.5,5x,e12.5)

   13   format(//2x,'fdash=',e12.5,5x,'zlamda=',e12.5)

        return
        end
```

6. SUITABLE VALUES FOR INPUT DATA

FILE 'AUXPA'

This file contains input values for the following auxiliary parameters (see Section 3 for definitions):

- IPRINT,
- KPRINT,
- IR,
- ITMAX,
- ZLAMDA,
- CTEST,
- CRIT,
- CRITAL, } required for Cases 2 and 3 only
- C, }
- BCALIB. }

IPRINT and *KPRINT*: These define the unit numbers for the output files 'CALCNS' and 'OUTPUT', respectively, which are used in the main program subroutines. The suitable values depend on the computer installation. On the IIASA computer suitable values were 2 and 3, respectively.

IR must be set equal to the number of patient categories contained in the main input file 'INIDA'.

ITMAX: Computational experience with the test problem described in Gibbs [3] suggests that:

i. the Newton-Raphson procedure for solving the equation $f(\lambda) = 0$ normally converges to within an acceptable criterion (see below) in less than 10 iterations, and

ii. the parameter estimation process for Cases 2 and 3 usually converges in two or three iterations.

Thus a suitable value for ITMAX, which is used as an iteration limit for both processes, is <u>20</u>. If either of these processes fails to converge after 20 iterations there is probably some data error preventing convergence.

ZLAMDA: Any initial value of λ greater than unity will permit convergence but computational experience suggests that the values of λ of the order of 4 usually lead to rapid convergence in the case of Case 1. For Cases 2 and 3 the most suitable value of λ depends upon the value of C. For reasons given in Gibbs [3] a suitable value is given by $\lambda = 2^C$. Since a suitable value for C is usually of the order of 2, see below, this implies a value of λ of the order of 4 for cases 2 and 3 also.

CTEST: The minimum value to which this should be set is an order of magnitude greater than errors arising from rounding. For example if the input data on admissions and length of stay is provided with four-figure accuracy the column totals should be accurate to at least 1 part in 10^3 and a value for CTEST of 0.1 would be appropriate since this will stop the program for errors in column totals greater than 0.1%.

CRIT: Computational experience suggests that with a value for CRIT of 10^{-5} the model output is accurate to 1 part in 10^4.

CRITAL: This should be an order of magnitude greater than CRIT otherwise the parameter estimation procedure may never converge. Thus if CRIT is set to 10^{-5}, CRITAL could be set to 10^{-4}.

C: For reasons given in Gibbs [3] the most suitable value of C depends on the values of the elasticities supplied in the input data. A suitable value is given by:

$$C = \max_i \, (\hat{\gamma}_i + \hat{\eta}_i) \quad .$$

This usually implies a value of C of the order of 2, see Gibbs [3]. If C is set lower than this value there is a risk that negative values will be computed for one or more of the α_i and β_i, in which case an error message will appear and the program will stop (see Section 10).

A complication may arise for Case 2 where the input elasticities are scaled in order that they should be consistent with the equation:

$$\sum_i \bar{x}_i \bar{u}_i (\hat{\gamma}_i + \hat{\eta}_i) = \bar{B} \quad .$$

The equation for C quoted above applies, strictly speaking, to the *scaled* values of elasticities. Thus it may be necessary to adjust the input value of C by the scaling factor, which is written to the terminal.

BCALIB should be assigned the value of $\overline{B}$, the mean bed supply from the data from which the elasticities are estimated.

FILE 'INIDA'

This contains the main input data. Its contents are different for each case, as can be seen from the illustrative listings in the following sections.

The first data entered are RUNAME, XBARA, UBARA and B.

RUNAME: Any four-character alphameric value is permitted.

XBARA and *UBARA* are used to check the data on admission rates and lengths of stay by category which follow. Appropriate values are given by:

$$\left.\begin{aligned} XBARA &= \sum_i X_i \quad , \\ UBARA &= \sum_i X_i U_i \Big/ \sum_i X_i \quad , \end{aligned}\right\} \text{for Cases 1 and 3}$$

$$\left.\begin{aligned} XBARA &= \sum_i \overline{x}_i \quad , \\ UBARA &= \sum_i \overline{x}_i \overline{u}_i \Big/ \sum_i \overline{x}_i \quad . \end{aligned}\right\} \text{for Case 2}$$

Corresponding quantities XBARB and UBARB are computed within the program and if the computed and input values do not agree within CTEST percent an error message is printed and the program is stopped, see Section 10. If in a series of runs the user changes some of the data on category admission rates or lengths of stay without also recomputing XBARA and UBARA then the error will probably be encountered. However since the computed values are written both to the terminal and to the file OUTPUT the user can remedy the situation by substituting these values for XBARA and UBARA in the file INIDA.

B: This is the number of bed-days whose allocation is to be simulated by the model. For Cases 1 and 3 it should lie within the range:

$$0 < B < \sum_i X_i U_i \quad ,$$

otherwise an error message will be given (see Section 3). For Case 2, B can take any value in the range given by:

$$0 < \frac{B}{2} < \text{BCALIB} \quad ,$$

provided that the initial values for C and λ are set in accordance with the suggestions given above.

The data on each category commence with CATEG, the category name, for which any 12 alphameric characters are allowed. This is followed by data on admission rates, lengths of stay and power factors (Case 1) or elasticities (Cases 2 and 3). The only restriction on this data is that each value should be strictly positive although values for elasticities that are much in excess of unity should be regarded with suspicion.

7. CASE 1 ILLUSTRATIVE INPUT/OUTPUT

AUXPA

file identifiers iprint	kprint	max. iterations itmax	no. categories ir
2,	3,	2Ø,	6,

newton-raph. conv. crit. crit		initial lambda zlamda		data test ctest
Ø.ØØØØ1	,	4.ØØØ	,	1.Ø

INIDA

run name	total ideal no patients	overall ideal av.stay	beds available
test	98.Ø ,	22.78 ,	12ØØ.Ø

patient category	ideal no patients	ideal av.stay	alpha(i)	beta(i)
varic. veins	12.8	, 15.4	, 1.636	, 3.Ø323
haemorrhoids	7.7	, 13.1	, 2.1138	, 4.6818
ischm. heart	1Ø.4	, 52.1	, Ø.5445	, 1.3148
pneumonia	21.Ø	, 19.7	, 2.28Ø8	, 9.8696
bronchitis	21.3	, 34.2	, 1.179Ø	, 49.ØØØ
appendicitis	24.8	, 1Ø.1	, 44.433	, 7.Ø645

OUTPUT

```
*****************************************************************************************
                  file=auxpa= for run test

 identifiers for files    max no. of    no.of pat.    newton-raphson    initial value
 =calcns= and =output=    iterations    categories    conv. criterion   of lambda
  iprint       kprint       itmax           ir             crit            zlamda

     2            3           20            6           0.0000100          4.000

*****************************************************************************************
                  file =inida= for run test

patient category       total sick     ideal length of stay   alpha      beta
                       xbig(i)           ubig(i)            alpha(i)   beta(i)

 1 varic. veins         12.8             15.400               1.6360    3.0323

 2 haemorrhoids          7.7             13.100               2.1138    4.6818

 3 ischm. heart         10.4             52.100               0.5445    1.3148

 4 pneumonia            21.0             19.700               2.2808    9.8696

 5 bronchitis           21.3             34.200               1.1790   49.0000

 6 appendicitis         24.8             10.100              44.4330    7.0645

            total=      98.0     mean=   22.780      bed-days available= 1200.0
*****************************************************************************************
*****************************************************************************************


 no. iteration  tolerance  lambda  beddays avail
      4         -0.76e-07   3.356  1200.0
```

simulation results for run test

patient category	actually treated (number/fract.)	actual aver. length of stay (number/fract.)	beddays actually used (number/fract.)
1 varic. veins	8.5/ 0.6612	11.406/ 0.7406	96.5/ 0.4897
2 haemorrhoids	5.4/ 0.6970	10.586/ 0.8081	56.8/ 0.5632
3 ischm. heart	5.4/ 0.5205	30.880/ 0.5927	167.2/ 0.3085
4 pneumonia	14.7/ 0.7012	17.623/ 0.8946	259.5/ 0.6273
5 bronchitis	12.3/ 0.5764	33.382/ 0.9761	409.8/ 0.5626
6 appendicitis	24.2/ 0.9750	8.692/ 0.8606	210.2/ 0.8391
all categories	70.4/ 0.7186	17.039/ 0.7480	1200.0/ 0.5375

CALCNS

```
          ************************************************************
newton-raphson iteration    1
```

category	q(i)	uvar(i)	d(i)	xvar(i)
1	Ø.14103e Ø1	Ø.10920e Ø2	Ø.34419e Ø1	Ø.80089e Ø1
2	Ø.12763e Ø1	Ø.10264e Ø2	Ø.35898e Ø1	Ø.51077e Ø1
3	Ø.18201e Ø1	Ø.28625e Ø2	Ø.31086e Ø1	Ø.49902e Ø1
4	Ø.11360e Ø1	Ø.17341e Ø2	Ø.37765e Ø1	Ø.14006e Ø2
5	Ø.10281e Ø1	Ø.33265e Ø2	Ø.39496e Ø1	Ø.11340e Ø2
6	Ø.11876e Ø1	Ø.85048e Ø1	Ø.37035e Ø1	Ø.24096e Ø2

flamda=-Ø.92249e Ø2   tol=-Ø.76874e-Ø1   lamda= Ø.40000e Ø1

category	g(i)	theta(i)	e(i)
1	Ø.85187e Ø1	Ø.31472e Ø1	Ø.12257e Ø2
2	Ø.76739e Ø1	Ø.27263e Ø1	Ø.59814e Ø1
3	Ø.78394e Ø1	Ø.26932e Ø1	Ø.31773e Ø2
4	Ø.57386e Ø1	Ø.17706e Ø1	Ø.22842e Ø2
5	Ø.41956e Ø1	Ø.10789e Ø1	Ø.44518e Ø2
6	Ø.91293e Ø4	Ø.14934e Ø5	Ø.73784e Ø1

fdash=-Ø.12475e Ø3   zlamda= Ø.32605e Ø1

```
        ************************************************************
newton-raphson iteration   2

 category         q(i)             uvar(i)           d(i)              xvar(i)
     1          Ø.134Ø6e Ø1      Ø.11488e Ø2      Ø.29Ø45e Ø1       Ø.85416e Ø1
     2          Ø.12312e Ø1      Ø.1Ø64Øe Ø2      Ø.3ØØØ2e Ø1       Ø.541Ø7e Ø1
     3          Ø.16663e Ø1      Ø.31268e Ø2      Ø.26845e Ø1       Ø.54873e Ø1
     4          Ø.11149e Ø1      Ø.1767Øe Ø2      Ø.31196e Ø1       Ø.14846e Ø2
     5          Ø.1Ø239e Ø1      Ø.334Ø1e Ø2      Ø.32289e Ø1       Ø.12438e Ø2
     6          Ø.11578e Ø1      Ø.87232e Ø1      Ø.3Ø731e Ø1       Ø.24195e Ø2

flamda= Ø.161Ø8e Ø2     tol= Ø.13423e-Ø1   lamda= Ø.326Ø5e Ø1


 category      g(i)            theta(i)          e(i)
      1     Ø.62895e Ø1     Ø.28763e Ø1     Ø.17Ø23e Ø2
      2     Ø.57339e Ø1     Ø.2516Øe Ø1     Ø.81124e Ø1
      3     Ø.59Ø67e Ø1     Ø.25292e Ø1     Ø.47568e Ø2
      4     Ø.44568e Ø1     Ø.1694Øe Ø1     Ø.3Ø393e Ø2
      5     Ø.33996e Ø1     Ø.1Ø737e Ø1     Ø.6Ø216e Ø2
      6     Ø.23949e Ø4     Ø.48111e Ø4     Ø.93321e Ø1

fdash=-Ø.17265e Ø3     zlamda= Ø.33538e Ø1
```

```
          ***********************************************************
newton-raphson iteration   3

 category          q(i)            uvar(i)           d(i)            xvar(i)
    1          0.13500e 01      0.11407e 02      0.29738e 01      0.84655e 01
    2          0.12374e 01      0.10587e 02      0.30758e 01      0.53676e 01
    3          0.16867e 01      0.30889e 02      0.27402e 01      0.54149e 01
    4          0.11178e 01      0.17624e 02      0.32032e 01      0.14727e 02
    5          0.10245e 01      0.33382e 02      0.33200e 01      0.12280e 02
    6          0.11619e 01      0.86927e 01      0.31536e 01      0.24181e 02

flamda= 0.36009e 00     tol= 0.30008e-03   lamda= 0.33538e 01


 category      g(i)             theta(i)          e(i)
     1     0.65595e 01      0.29125e 01      0.16266e 02
     2     0.59700e 01      0.25441e 01      0.77774e 01
     3     0.61437e 01      0.25516e 01      0.44976e 02
     4     0.46154e 01      0.17044e 01      0.29216e 02
     5     0.34998e 01      0.10744e 01      0.57757e 02
     6     0.28810e 04      0.56257e 04      0.90342e 01

 fdash=-0.16503e 03     zlamda= 0.33560e 01
```

```
****************************************************************
newton-raphson iteration   4

 category          q(i)              uvar(i)             d(i)              xvar(i)
     1          Ø.135Ø2e Ø1       Ø.114Ø6e Ø2       Ø.29754e Ø1       Ø.84637e Ø1
     2          Ø.12375e Ø1       Ø.1Ø586e Ø2       Ø.3Ø776e Ø1       Ø.53666e Ø1
     3          Ø.16872e Ø1       Ø.3Ø88Øe Ø2       Ø.27415e Ø1       Ø.54133e Ø1
     4          Ø.11178e Ø1       Ø.17623e Ø2       Ø.32Ø51e Ø1       Ø.14724e Ø2
     5          Ø.1Ø245e Ø1       Ø.33382e Ø2       Ø.33222e Ø1       Ø.12277e Ø2
     6          Ø.1162Øe Ø1       Ø.8692Øe Ø1       Ø.31554e Ø1       Ø.24181e Ø2

flamda=-Ø.91553e-Ø4     tol=-Ø.76294e-Ø7    lamda= Ø.3356Øe Ø1
```

8. CASE 2 ILLUSTRATIVE INPUT/OUTPUT

AUXPA

file identifiers iprint	kprint	max. iterations itmax	no. categories ir
2,	3,	2Ø,	6,

initial lamda zlamda	data test ctest	newton-raph. conv. crit. crit	alpha est. crit. crital	arbitrary constant c	calibration bed point bcalib
4.ØØØ ,	1.Ø	, Ø.ØØØØ1	, Ø.ØØØ1Ø ,	2.22ØØ,	1Ø94.2

INIDA

run name	total mean no. patients	overall mean av.stay	beds available
test	63.9	,17.123ØØØ,	12ØØ.Ø

patient category	mea no patients	mean av.stay	elasticities gamhat(i)	etahat(i)
varic. veins	6.3	, 11.3	,Ø.78	,Ø.62
haemorrhoids	4.1	, 13.1	,Ø.7Ø	,Ø.44
ischm. heart	4.6	, 4Ø.2	,1.14	,1.Ø8
pneumonia	12.3	, 14.7	,Ø.71	,Ø.23
bronchitis	11.8	, 27.4	,1.13	,Ø.Ø5
appendicitis	24.8	, 11.3	,Ø.Ø5	,Ø.31

OUTPUT

```
*************************************************************************************
                  file=auxpa= for run test

 identifiers for files    max no. of    no.of pat.    initial value
 =calcns= and =output=    iterations    categories    of lamda
  iprint        kprint      itmax           ir            zlamda

     2             3          20             6             4.000

    data       newton-raphson      alpha          arbitrary     calibration
    test       conv. criterion    criterion       constant      bed point
    ctest        crit             crital            c             bcalib

    1.00000     0.0000100        0.0001000        2.22000        1094.20
*************************************************************************************
*************************************************************************************
                  file =inida= for run test

patient category   mean admissions   mean length of stay    elasticities
                      xbar(i)             ubar(i)          gamhat(i)   eta(i)

 1 varic. veins         6.3               11.300           0.7800000    0.6200000

 2 haemorrhoids         4.1               13.100           0.7000000    0.4400000

 3 ischm. heart         4.6               40.200           1.1400000    1.0800000

 4 pneumonia           12.3               14.700           0.7100000    0.2300000

 5 bronchitis          11.8               27.400           1.1300000    0.0500000

 6 appendicitis        24.8               11.300           0.0500000    0.3100000
           total=      63.9     mean=     17.123      bed-days available= 1200.0
*************************************************************************************
```

elasticity data scaled by Ø.89411

parameter iteration 1 completed with 1 newton-raphson iterations
tolerance = -Ø.91e-Ø5 lamda = 4.ØØØ

parameter iteration Ø completed with 4 newton-raphson iterations
tolerance = Ø.51e-Ø6 lamda = 3.27Ø

calculated parameter values for run test

i	category	alpha(i)	beta(i)	xbig(i)	ubig(i)
1	varic. veins	1.62Ø	3.ØØ5	1Ø.Ø95	15.974
2	haemorrhoids	2.Ø94	4.643	6.196	16.748
3	ischm. heart	Ø.536	1.299	9.616	73.468
4	pneumonia	2.259	9.795	18.49Ø	16.714
5	broncnitis	1.164	48.659	22.259	28.176
6	appendicitis	44.134	7.Ø1Ø	25.53Ø	13.435

simulation results for run test

patient category	actually treated (number/fract.)	actual aver. length of stay (number/fract.)	beddays actually used (number/fract.)
1 varic. veins	6.7/ 0.6652	11.883/ 0.7439	79.8/ 0.4948
2 haemorrhoids	4.3/ 0.7006	13.576/ 0.8106	58.9/ 0.5679
3 ischm. heart	5.1/ 0.5254	43.882/ 0.5973	221.7/ 0.3138
4 pneumonia	13.0/ 0.7048	14.977/ 0.8961	195.2/ 0.6315
5 bronchitis	12.9/ 0.5811	27.511/ 0.9764	355.8/ 0.5674
6 appendicitis	24.9/ 0.9754	11.588/ 0.8625	288.6/ 0.8413
all categories	67.0/ 0.7265	17.917/ 0.7339	1200.0/ 0.5332

CALCNS

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

calculated parameters at iteration  1

 i   alpha(i)             beta(i)              xbig(i)              ubig(i)

 1     1.619514            3.004692            10.094823            15.974129
 2     2.093667            4.642974             6.195897            16.748011
 3     0.535856            1.298990             9.616684            73.469368
 4     2.258924            9.795256            18.489796            16.714296
 5     1.164221           48.658176            22.259155            28.175692
 6    44.133141            7.009383            25.529669            13.435315
          ************************************************************
newton-raphson iteration   1

 category           q(i)             uvar(i)            d(i)            xvar(i)
    1           0.14136e 01      0.11300e 02      0.34385e 01      0.63000e 01
    2           0.12785e 01      0.13100e 02      0.35872e 01      0.41000e 01
    3           0.18276e 01      0.40200e 02      0.31037e 01      0.46000e 01
    4           0.11370e 01      0.14700e 02      0.37750e 01      0.12300e 02
    5           0.10283e 01      0.27400e 02      0.39493e 01      0.11800e 02
    6           0.11890e 01      0.11300e 02      0.37016e 01      0.24800e 02


flamda=-0.99487e-02      tol=-0.90922e-05    lamda= 0.40000e 01


 category        g(i)             theta(i)           e(i)
    1        0.85149e 01      0.31442e 01      0.10035e 02
    2        0.76706e 01      0.27239e 01      0.61651e 01
    3        0.78360e 01      0.26902e 01      0.41335e 02
    4        0.57368e 01      0.17695e 01      0.17113e 02
    5        0.41952e 01      0.10787e 01      0.38414e 02
    6        0.91415e 04      0.14955e 05      0.10158e 02


 fdasn=-0.12322e 03      zlamda= 0.39999e 01
```

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

calculated parameters at iteration  2

 i   alpha(i)           beta(i)            xbig(i)           ubig(i)

 1     1.619582          3.004773          10.094648         15.973933
 2     2.093741          4.643088           6.195804         16.747871
 3     0.535903          1.299036           9.616409         73.467857
 4     2.258996          9.795473          18.489527         16.714222
 5     1.164265         48.659180          22.258663         28.175667
 6    44.134174          7.009545          25.529646         13.435233


          ************************************************************
newton-raphson iteration   0

 category          g(i)            uvar(i)           d(i)            xvar(i)
    1         0.14136e 01     0.11300e 02     0.34384e 01     0.63000e 01
    2         0.12785e 01     0.13100e 02     0.35872e 01     0.41000e 01
    3         0.18276e 01     0.40200e 02     0.31037e 01     0.46000e 01
    4         0.11370e 01     0.14700e 02     0.37749e 01     0.12300e 02
    5         0.10283e 01     0.27400e 02     0.39492e 01     0.11800e 02
    6         0.11890e 01     0.11300e 02     0.37015e 01     0.24800e 02

flamda=-0.10010e-01     tol=-0.91480e-05   lamda= 0.39999e 01
```

```
************************************************************
newton-raphson iteration   1

 category        q(i)            uvar(i)          d(i)            xvar(i)
    1         Ø.14136e Ø1     Ø.113ØØe Ø2     Ø.34384e Ø1     Ø.63ØØØe Ø1
    2         Ø.12785e Ø1     Ø.131ØØe Ø2     Ø.35872e Ø1     Ø.41ØØØe Ø1
    3         Ø.18276e Ø1     Ø.4Ø2ØØe Ø2     Ø.31Ø37e Ø1     Ø.46ØØØe Ø1
    4         Ø.1137Øe Ø1     Ø.147ØØe Ø2     Ø.37749e Ø1     Ø.123ØØe Ø2
    5         Ø.1Ø283e Ø1     Ø.274ØØe Ø2     Ø.39492e Ø1     Ø.118ØØe Ø2
    6         Ø.1189Øe Ø1     Ø.113ØØe Ø2     Ø.37Ø15e Ø1     Ø.248ØØe Ø2

flamda=-Ø.1Ø581e Ø3     tol=-Ø.88175e-Ø1    lamda= Ø.39999e Ø1


 category     g(i)            theta(i)          e(i)
    1      Ø.85147e Ø1     Ø.31442e Ø1     Ø.1ØØ35e Ø2
    2      Ø.767Ø4e Ø1     Ø.27239e Ø1     Ø.61651e Ø1
    3      Ø.78359e Ø1     Ø.269Ø2e Ø1     Ø.41335e Ø2
    4      Ø.57367e Ø1     Ø.17695e Ø1     Ø.17113e Ø2
    5      Ø.41951e Ø1     Ø.1Ø787e Ø1     Ø.38414e Ø2
    6      Ø.914Ø5e Ø4     Ø.14954e Ø5     Ø.1Ø158e Ø2

 fdash=-Ø.12322e Ø3     zlamda= Ø.31412e Ø1
```

****

newton-raphson iteration 2

category	q(i)	uvar(i)	d(i)	xvar(i)
1	Ø.13308e Ø1	Ø.12ØØ3e Ø2	Ø.2813Øe Ø1	Ø.68918e Ø1
2	Ø.12249e Ø1	Ø.13673e Ø2	Ø.29Ø15e Ø1	Ø.4391Øe Ø1
3	Ø.16452e Ø1	Ø.44656e Ø2	Ø.26Ø93e Ø1	Ø.515Ø2e Ø1
4	Ø.11119e Ø1	Ø.15Ø33e Ø2	Ø.3Ø115e Ø1	Ø.13183e Ø2
5	Ø.1Ø233e Ø1	Ø.27534e Ø2	Ø.31122e Ø1	Ø.13173e Ø2
6	Ø.11536e Ø1	Ø.11646e Ø2	Ø.29687e Ø1	Ø.24922e Ø2

flamda= Ø.22776e Ø2 tol= Ø.1898Øe-Ø1 lamda= Ø.31412e Ø1

category	g(i)	theta(i)	e(i)
1	Ø.59475e Ø1	Ø.28272e Ø1	Ø.14815e Ø2
2	Ø.54343e Ø1	Ø.24776e Ø1	Ø.88476e Ø1
3	Ø.56Ø56e Ø1	Ø.2498Øe Ø1	Ø.66727e Ø2
4	Ø.42546e Ø1	Ø.16795e Ø1	Ø.24ØØ5e Ø2
5	Ø.32714e Ø1	Ø.1Ø726e Ø1	Ø.54946e Ø2
6	Ø.18782e Ø4	Ø.39177e Ø4	Ø.13414e Ø2

fdash=-Ø.18275e Ø3 zlamda= Ø.32658e Ø1

```
********************************************************************
newton-raphson iteration   3

 category          q(i)            uvar(i)          d(i)             xvar(i)
    1          Ø.13438e Ø1     Ø.11887e Ø2     Ø.29Ø62e Ø1     Ø.67177e Ø1
    2          Ø.12333e Ø1     Ø.13579e Ø2     Ø.3ØØ29e Ø1     Ø.43425e Ø1
    3          Ø.16733e Ø1     Ø.439Ø6e Ø2     Ø.26844e Ø1     Ø.5Ø559e Ø1
    4          Ø.11159e Ø1     Ø.14979e Ø2     Ø.31234e Ø1     Ø.13Ø36e Ø2
    5          Ø.1Ø241e Ø1     Ø.27512e Ø2     Ø.32339e Ø1     Ø.12941e Ø2
    6          Ø.11592e Ø1     Ø.1159Øe Ø2     Ø.3Ø765e Ø1     Ø.249Ø2e Ø2

flamda= Ø.71857e ØØ     tol= Ø.59881e-Ø3   lamda= Ø.32658e Ø1


 category      g(i)            theta(i)         e(i)
    1      Ø.63Ø29e Ø1     Ø.28763e Ø1     Ø.1391Øe Ø2
    2      Ø.57455e Ø1     Ø.25158e Ø1     Ø.83462e Ø1
    3      Ø.59188e Ø1     Ø.25284e Ø1     Ø.61742e Ø2
    4      Ø.44648e Ø1     Ø.16937e Ø1     Ø.22729e Ø2
    5      Ø.34Ø51e Ø1     Ø.1Ø736e Ø1     Ø.51868e Ø2
    6      Ø.24235e Ø4     Ø.48612e Ø4     Ø.12826e Ø2

 fdash=-Ø.17142e Ø3     zlamda= Ø.327ØØe Ø1
```

```
************************************************************
newton-raphson iteration   4
```

category	q(i)	uvar(i)	d(i)	xvar(i)
1	Ø.13443e Ø1	Ø.11883e Ø2	Ø.29Ø94e Ø1	Ø.67149e Ø1
2	Ø.12336e Ø1	Ø.13576e Ø2	Ø.3ØØ63e Ø1	Ø.434Ø9e Ø1
3	Ø.16742e Ø1	Ø.43882e Ø2	Ø.26869e Ø1	Ø.5Ø528e Ø1
4	Ø.1116Øe Ø1	Ø.14977e Ø2	Ø.31272e Ø1	Ø.13Ø31e Ø2
5	Ø.1Ø241e Ø1	Ø.27511e Ø2	Ø.3238Øe Ø1	Ø.12934e Ø2
6	Ø.11594e Ø1	Ø.11588e Ø2	Ø.3Ø8Ø1e Ø1	Ø.249Ø1e Ø2

flamda= Ø.61Ø35e-Ø3   tol= Ø.5Ø863e-Ø6   lamda= Ø.327ØØe Ø1

category	g(i)	theta(i)	e(i)
1	Ø.6315Øe Ø1	Ø.28779e Ø1	Ø.13882e Ø2
2	Ø.57561e Ø1	Ø.25171e Ø1	Ø.833Ø1e Ø1
3	Ø.59294e Ø1	Ø.25294e Ø1	Ø.61584e Ø2
4	Ø.44719e Ø1	Ø.16942e Ø1	Ø.22688e Ø2
5	Ø.34Ø96e Ø1	Ø.1Ø736e Ø1	Ø.51769e Ø2
6	Ø.24439e Ø4	Ø.48959e Ø4	Ø.128Ø7e Ø2

fdash=-Ø.171Ø6e Ø3   zlamda= Ø.327ØØe Ø1

9. CASE 3 ILLUSTRATIVE INPUT/OUTPUT

AUXPA

file identifiers iprint	kprint	max. iterations itmax	no. categories ir
2,	3,	20,	6,

initial lamda zlamda	data test ctest	newton-raph. conv. crit. crit	alpha est. crit. crital	arbitrary constant c	calibration bed point bcalib
4.000 ,	1.0	, 0.00001	, 0.00010 ,	2.2276,	1094.2

INIDA

run name	total ideal no patients	overall ideal av.stay	beds available
test	98.0	, 22.78	, 1200.0

patient category	ideal no patients	ideal av.stay	elasticities gamhat(i)	etahat(i)
varic. veins	12.8	, 15.4	, 0.7800	, 0.6200
haemorrhoids	7.7	, 13.1	, 0.7000	, 0.4400
ischm. heart	10.4	, 52.1	, 1.1400	, 1.0800
pneumonia	21.0	, 19.7	, 0.7100	, 0.2300
bronchitis	21.3	, 34.2	, 1.1300	, 0.0500
appendicitis	24.8	, 10.1	, 0.0500	, 0.3100

<u>OUTPUT</u>

```
*************************************************************************************
                   file=auxpa= for run test

 identifiers for files    max no. of    no.of pat.    initial value
 =calcns= and =output=    iterations    categories    of lamda
  iprint       kprint       itmax           ir            zlamda

    2            3            20            6            4.000

    data       newton-raphson     alpha          arbitrary     calibration
    test       conv. criterion   criterion       constant      bed point
    ctest        crit            crital             c             bcalib

    1.00000     0.0000100       0.0001000        2.22760        1094.20

*************************************************************************************
*************************************************************************************
                   file =inida= for run test

patient category       total sick     ideal length of stay    elasticities
                       xbig(i)            ubig(i)           gamhat(i)   eta(i)

 1 varic. veins         12.8              15.400              0.7800     0.6200

 2 haemorrhoids          7.7              13.100              0.7000     0.4400

 3 iscnm. heart         10.4              52.100              1.1400     1.0800

 4 pneumonia            21.0              19.700              0.7100     0.2300

 5 bronchitis           21.3              34.200              1.1300     0.0500

 6 appendicitis         24.8              10.100              0.0500     0.3100
            total=      98.0     mean=    22.780       bed-days available= 1200.0
*************************************************************************************
```

```
parameter iteration  1 completed with  4 newton-raphson iterations
tolerance =  Ø.14e-Ø6   lamda =    3.489

parameter iteration  2 completed with  2 newton-raphson iterations
tolerance =  Ø.42e-Ø7   lamda =    3.49Ø

parameter iteration  Ø completed with  4 newton-raphson iterations
tolerance =  Ø.18e-Ø6   lamda =    2.915
```

calculated parameter values for run test

i	category	alpha(i)	beta(i)
1	varic. veins	1.31Ø	2.576
2	haemorrhoids	1.738	4.Ø39
3	ischm. heart	Ø.342	1.Ø53
4	pneumonia	1.897	8.639
5	bronchitis	Ø.931	43.342
6	appendicitis	39.Ø53	6.152

simulation results for run test

patient category	actually treated (number/fract.)	actual aver. length of stay (number/fract.)	beddays actually used (number/fract.)
1 varic. veins	8.4/ 0.6599	11.418/ 0.7414	96.5/ 0.4893
2 haemorrhoids	5.4/ 0.6964	10.594/ 0.8087	56.8/ 0.5632
3 ischm. heart	5.4/ 0.5167	30.939/ 0.5938	166.3/ 0.3068
4 pneumonia	14.7/ 0.7013	17.631/ 0.8950	259.7/ 0.6276
5 pronchitis	12.3/ 0.5773	33.385/ 0.9762	410.5/ 0.5636
6 appendicitis	24.2/ 0.9750	8.697/ 0.8611	210.3/ 0.8395
all categories	70.4/ 0.7182	17.048/ 0.7484	1200.0/ 0.5375

CALCNS

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

 parameter iteration number  1

  i calculated alpha(i)  calculated beta(i)

  1      1.296010      2.592903
  2      1.731054      4.062727
  3      0.319989      1.062593
  4      1.899894      8.685218
  5      0.938300     43.552002
  6     39.041508      6.185807
          ******************************************************
 newton-raphson iteration   1

  category          q(i)              uvar(i)           d(i)             xvar(i)
     1           0.14709e 01       0.10470e 02      0.33827e 01      0.75283e 01
     2           0.13150e 01       0.99621e 01      0.35444e 01      0.48447e 01
     3           0.19584e 01       0.26604e 02      0.30236e 01      0.44977e 01
     4           0.11539e 01       0.17073e 02      0.37505e 01      0.13312e 02
     5           0.10316e 01       0.33152e 02      0.39435e 01      0.10494e 02
     6           0.12128e 01       0.83279e 01      0.36697e 01      0.24008e 02

 flamda=-0.72339e 02     tol=-0.66111e-01   lamda= 0.40000e 01


  category       g(i)            theta(i)          e(i)
      1       0.82035e 01      0.29594e 01      0.12385e 02
      2       0.74873e 01      0.26162e 01      0.61749e 01
      3       0.73422e 01      0.24146e 01      0.29812e 02
      4       0.56801e 01      0.17377e 01      0.23976e 02
      5       0.41887e 01      0.10752e 01      0.46073e 02
      6       0.83079e 04      0.13440e 05      0.80778e 01

  fdash=-0.12650e 03     zlamda= 0.34281e 01
```

****

newton-raphson iteration 2

category	q(i)	uvar(i)	d(i)	xvar(i)
1	Ø.14Ø9Øe Ø1	Ø.1Ø929e Ø2	Ø.29856e Ø1	Ø.7949Øe Ø1
2	Ø.12755e Ø1	Ø.1Ø27Øe Ø2	Ø.31Ø31e Ø1	Ø.5Ø865e Ø1
3	Ø.18172e Ø1	Ø.2867Øe Ø2	Ø.272Ø7e Ø1	Ø.48722e Ø1
4	Ø.11357e Ø1	Ø.17347e Ø2	Ø.32511e Ø1	Ø.13985e Ø2
5	Ø.1Ø28Øe Ø1	Ø.33267e Ø2	Ø.33883e Ø1	Ø.11349e Ø2
6	Ø.1187Øe Ø1	Ø.85Ø87e Ø1	Ø.31932e Ø1	Ø.24Ø91e Ø2

flamda= Ø.97266e Ø1 tol= Ø.88892e-Ø2 lamda= Ø.34281e Ø1

category	g(i)	theta(i)	e(i)
1	Ø.656Ø9e Ø1	Ø.27826e Ø1	Ø.16Ø48e Ø2
2	Ø.6Ø315e Ø1	Ø.2473Øe Ø1	Ø.78427e Ø1
3	Ø.59855e Ø1	Ø.2328Øe Ø1	Ø.41159e Ø2
4	Ø.47Ø15e Ø1	Ø.1684Øe Ø1	Ø.29964e Ø2
5	Ø.35748e Ø1	Ø.1Ø717e Ø1	Ø.58392e Ø2
6	Ø.3Ø6Ø2e Ø4	Ø.57815e Ø4	Ø.96717e Ø1

fdash=-Ø.163Ø8e Ø3 zlamda= Ø.34878e Ø1

```
***************************************************************
newton-raphson iteration   3
```

category	q(i)	uvar(i)	d(i)	xvar(i)
1	Ø.14158e Ø1	Ø.10877e Ø2	Ø.30279e Ø1	Ø.79005e Ø1
2	Ø.12799e Ø1	Ø.10235e Ø2	Ø.31497e Ø1	Ø.50587e Ø1
3	Ø.18325e Ø1	Ø.28431e Ø2	Ø.27534e Ø1	Ø.48283e Ø1
4	Ø.11377e Ø1	Ø.17316e Ø2	Ø.33036e Ø1	Ø.13908e Ø2
5	Ø.10284e Ø1	Ø.33254e Ø2	Ø.34463e Ø1	Ø.11250e Ø2
6	Ø.11899e Ø1	Ø.84883e Ø1	Ø.32434e Ø1	Ø.24082e Ø2

flamda= Ø.13782e ØØ   tol= Ø.12595e-Ø3   lamda= Ø.34878e Ø1

category	g(i)	theta(i)	e(i)
1	Ø.67274e Ø1	Ø.28019e Ø1	Ø.15588e Ø2
2	Ø.61794e Ø1	Ø.24886e Ø1	Ø.76354e Ø1
3	Ø.61247e Ø1	Ø.23377e Ø1	Ø.39694e Ø2
4	Ø.48021e Ø1	Ø.16899e Ø1	Ø.29225e Ø2
5	Ø.36388e Ø1	Ø.10720e Ø1	Ø.56865e Ø2
6	Ø.34219e Ø4	Ø.63536e Ø4	Ø.94789e Ø1

fdash=-Ø.15849e Ø3   zlamda= Ø.34887e Ø1

****

newton-raphson iteration 4

category	q(i)	uvar(i)	d(i)	xvar(i)
1	Ø.14159e Ø1	Ø.1Ø876e Ø2	Ø.3Ø285e Ø1	Ø.78998e Ø1
2	Ø.12799e Ø1	Ø.1Ø235e Ø2	Ø.315Ø4e Ø1	Ø.5Ø583e Ø1
3	Ø.18327e Ø1	Ø.28428e Ø2	Ø.27539e Ø1	Ø.48277e Ø1
4	Ø.11377e Ø1	Ø.17316e Ø2	Ø.33Ø43e Ø1	Ø.139Ø7e Ø2
5	Ø.1Ø284e Ø1	Ø.33254e Ø2	Ø.34471e Ø1	Ø.11249e Ø2
6	Ø.11899e Ø1	Ø.8488Øe Ø1	Ø.32441e Ø1	Ø.24Ø82e Ø2

flamda= Ø.15259e-Ø3 tol= Ø.13945e-Ø6 lamda= Ø.34887e Ø1

category	g(i)	theta(i)	e(i)
1	Ø.67299e Ø1	Ø.28Ø22e Ø1	Ø.15582e Ø2
2	Ø.61816e Ø1	Ø.24889e Ø1	Ø.76324e Ø1
3	Ø.61267e Ø1	Ø.23378e Ø1	Ø.39673e Ø2
4	Ø.48Ø36e Ø1	Ø.169ØØe Ø1	Ø.29215e Ø2
5	Ø.36397e Ø1	Ø.1Ø721e Ø1	Ø.56843e Ø2
6	Ø.34274e Ø4	Ø.63623e Ø4	Ø.94761e Ø1

fdash=-Ø.15842e Ø3 zlamda= Ø.34887e Ø1

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

parameter iteration number  2

 i calculated alpha(i)  calculated beta(i)

 1       1.310134       2.575888
 2       1.738257       4.038751
 3       0.341776       1.052824
 4       1.896724       8.639350
 5       0.930581      43.341011
 6      39.052784       6.151776


          ******************************************************
newton-raphson iteration   0

 category          q(i)            uvar(i)          d(i)            xvar(i)
    1          0.14183e 01     0.10858e 02     0.30266e 01     0.79254e 01
    2          0.12814e 01     0.10223e 02     0.31489e 01     0.50648e 01
    3          0.18380e 01     0.28346e 02     0.27511e 01     0.48919e 01
    4          0.11384e 01     0.17305e 02     0.33035e 01     0.13901e 02
    5          0.10286e 01     0.33250e 02     0.34469e 01     0.11220e 02
    6          0.11909e 01     0.84809e 01     0.32431e 01     0.24082e 02

flamda= 0.17265e 00     tol= 0.15779e-03   lamda= 0.34887e 01
```

```
 ************************************************************
newton-raphson iteration   1

 category        q(i)           uvar(i)          d(i)            xvar(i)
    1         Ø.14183e Ø1    Ø.1Ø858e Ø2    Ø.3Ø266e Ø1    Ø.79254e Ø1
    2         Ø.12814e Ø1    Ø.1Ø223e Ø2    Ø.31489e Ø1    Ø.5Ø648e Ø1
    3         Ø.1838Øe Ø1    Ø.28346e Ø2    Ø.27511e Ø1    Ø.48919e Ø1
    4         Ø.11384e Ø1    Ø.173Ø5e Ø2    Ø.33Ø35e Ø1    Ø.139Ø1e Ø2
    5         Ø.1Ø286e Ø1    Ø.3325Øe Ø2    Ø.34469e Ø1    Ø.1122Øe Ø2
    6         Ø.119Ø9e Ø1    Ø.848Ø9e Ø1    Ø.32431e Ø1    Ø.24Ø82e Ø2

flamda= Ø.17265e ØØ     tol= Ø.15779e-Ø3   lamda= Ø.34887e Ø1


 category      g(i)          theta(i)         e(i)
     1      Ø.67846e Ø1    Ø.2837Øe Ø1    Ø.15577e Ø2
     2      Ø.62Ø97e Ø1    Ø.25Ø62e Ø1    Ø.76315e Ø1
     3      Ø.62258e Ø1    Ø.23977e Ø1    Ø.398Ø1e Ø2
     4      Ø.48Ø89e Ø1    Ø.1693Øe Ø1    Ø.29236e Ø2
     5      Ø.36396e Ø1    Ø.1Ø72Øe Ø1    Ø.56916e Ø2
     6      Ø.35487e Ø4    Ø.66155e Ø4    Ø.95Ø62e Ø1

 fdash=-Ø.15867e Ø3     zlamda= Ø.34898e Ø1
```

****

newton-raphson iteration 2

category	q(i)	var(i)	d(i)	xvar(i)
1	Ø.14184e Ø1	Ø.1Ø857e Ø2	Ø.3Ø273e Ø1	Ø.79245e Ø1
2	Ø.12815e Ø1	Ø.1Ø222e Ø2	Ø.31498e Ø1	Ø.5Ø643e Ø1
3	Ø.18383e Ø1	Ø.28342e Ø2	Ø.27517e Ø1	Ø.48911e Ø1
4	Ø.11384e Ø1	Ø.173Ø4e Ø2	Ø.33Ø44e Ø1	Ø.139ØØe Ø2
5	Ø.1Ø286e Ø1	Ø.33249e Ø2	Ø.3448Øe Ø1	Ø.11218e Ø2
6	Ø.1191Øe Ø1	Ø.848Ø6e Ø1	Ø.3244Øe Ø1	Ø.24Ø82e Ø2

flamda= Ø.45776e-Ø4 tol= Ø.41835e-Ø7 lamda= Ø.34898e Ø1

category	g(i)	theta(i)	e(i)
1	Ø.67877e Ø1	Ø.28373e Ø1	Ø.15569e Ø2
2	Ø.62124e Ø1	Ø.25Ø65e Ø1	Ø.76278e Ø1
3	Ø.62284e Ø1	Ø.23979e Ø1	Ø.39775e Ø2
4	Ø.481Ø8e Ø1	Ø.16931e Ø1	Ø.29223e Ø2
5	Ø.364Ø8e Ø1	Ø.1Ø72Øe Ø1	Ø.56889e Ø2
6	Ø.35558e Ø4	Ø.66268e Ø4	Ø.95Ø27e Ø1

fdash=-Ø.15859e Ø3 zlamda= Ø.34898e Ø1

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

parameter iteration number   3


 i calculated alpha(i)  calculated beta(i)

 1       1.310101       2.575931
 2       1.738240       4.038811
 3       0.341724       1.052849
 4       1.896733       8.639465
 5       0.930600      43.341541
 6      39.052776       6.151861


        *****************************************************
newton-raphson iteration   0

 category          q(i)              uvar(i)             d(i)              xvar(i)
    1         0.14184e 01       0.10857e 02       0.30273e 01       0.79244e 01
    2         0.12815e 01       0.10222e 02       0.31498e 01       0.50643e 01
    3         0.18383e 01       0.28342e 02       0.27517e 01       0.48910e 01
    4         0.11384e 01       0.17304e 02       0.33044e 01       0.13900e 02
    5         0.10286e 01       0.33249e 02       0.34480e 01       0.11219e 02
    6         0.11910e 01       0.84806e 01       0.32440e 01       0.24082e 02

flamda= 0.15259e-04     tol= 0.13945e-07   lamda= 0.34898e 01
```

****

newton-raphson iteration 1

category	q(i)	uvar(i)	d(i)	xvar(i)
1	0.14184e 01	0.10857e 02	0.30273e 01	0.79244e 01
2	0.12815e 01	0.10222e 02	0.31498e 01	0.50643e 01
3	0.18383e 01	0.28342e 02	0.27517e 01	0.48910e 01
4	0.11384e 01	0.17304e 02	0.33044e 01	0.13900e 02
5	0.10286e 01	0.33249e 02	0.34480e 01	0.11219e 02
6	0.11910e 01	0.84806e 01	0.32440e 01	0.24082e 02

flamda=-0.10580e 03 tol=-0.88167e-01 lamda= 0.34898e 01

category	g(i)	theta(i)	e(i)
1	0.67875e 01	0.28372e 01	0.15569e 02
2	0.62123e 01	0.25064e 01	0.76278e 01
3	0.62282e 01	0.23977e 01	0.39774e 02
4	0.48108e 01	0.16931e 01	0.29223e 02
5	0.36408e 01	0.10720e 01	0.56889e 02
6	0.35555e 04	0.66262e 04	0.95026e 01

fdash=-0.15859e 03 zlamda= 0.28226e 01

****

newton-raphson iteration 2

category	q(i)	uvar(i)	d(i)	xvar(i)
1	Ø.13367e Ø1	Ø.11521e Ø2	Ø.25432e Ø1	Ø.85453e Ø1
2	Ø.12287e Ø1	Ø.1Ø662e Ø2	Ø.26185e Ø1	Ø.54178e Ø1
3	Ø.16578e Ø1	Ø.31428e Ø2	Ø.237ØØe Ø1	Ø.54667e Ø1
4	Ø.11137e Ø1	Ø.1769Øe Ø2	Ø.27122e Ø1	Ø.14881e Ø2
5	Ø.1Ø237e Ø1	Ø.334Ø9e Ø2	Ø.27979e Ø1	Ø.125Ø1e Ø2
6	Ø.11561e Ø1	Ø.87359e Ø1	Ø.26757e Ø1	Ø.24198e Ø2

flamda= Ø.2Ø287e Ø2   tol= Ø.169Ø6e-Ø1   lamda= Ø.28226e Ø1

category	g(i)	theta(i)	e(i)
1	Ø.49718e Ø1	Ø.269Ø4e Ø1	Ø.22291e Ø2
2	Ø.46Ø2Øe Ø1	Ø.23164e Ø1	Ø.1Ø618e Ø2
3	Ø.46698e Ø1	Ø.22699e Ø1	Ø.62241e Ø2
4	Ø.37Ø46e Ø1	Ø.162Ø9e Ø1	Ø.39761e Ø2
5	Ø.29272e Ø1	Ø.1Ø672e Ø1	Ø.78866e Ø2
6	Ø.89376e Ø3	Ø.2Ø622e Ø4	Ø.12178e Ø2

fdash=-Ø.22596e Ø3   zlamda= Ø.29124e Ø1

****

newton-raphson iteration 3

category	q(i)	uvar(i)	d(i)	xvar(i)
1	Ø.13484e Ø1	Ø.11421e Ø2	Ø.261Ø1e Ø1	Ø.84498e Ø1
2	Ø.12363e Ø1	Ø.1Ø596e Ø2	Ø.26913e Ø1	Ø.53637e Ø1
3	Ø.16833e Ø1	Ø.3Ø952e Ø2	Ø.24238e Ø1	Ø.53761e Ø1
4	Ø.11173e Ø1	Ø.17632e Ø2	Ø.27927e Ø1	Ø.14732e Ø2
5	Ø.1Ø244e Ø1	Ø.33385e Ø2	Ø.28855e Ø1	Ø.123Ø3e Ø2
6	Ø.11612e Ø1	Ø.36978e Ø1	Ø.27532e Ø1	Ø.24181e Ø2

flamda= Ø.52843e ØØ tol= Ø.44Ø36e-Ø3 lamda= Ø.29124e Ø1

category	g(i)	theta(i)	e(i)
1	Ø.52Ø68e Ø1	Ø.26343e Ø1	Ø.21135e Ø2
2	Ø.48112e Ø1	Ø.23437e Ø1	Ø.1Ø111e Ø2
3	Ø.48744e Ø1	Ø.22887e Ø1	Ø.58231e Ø2
4	Ø.385Ø6e Ø1	Ø.16314e Ø1	Ø.37991e Ø2
5	Ø.3Ø23Øe Ø1	Ø.1Ø679e Ø1	Ø.75152e Ø2
6	Ø.1Ø959e Ø4	Ø.245Ø2e Ø4	Ø.1174Øe Ø2

fdash=-Ø.21436e Ø3 zlamda= Ø.29149e Ø1

****

newton-raphson iteration 4

category	q(i)	uvar(i)	d(i)	xvar(i)
1	Ø.13487e Ø1	Ø.11418e Ø2	Ø.26119e Ø1	Ø.84472e Ø1
2	Ø.12365e Ø1	Ø.1Ø594e Ø2	Ø.26933e Ø1	Ø.53623e Ø1
3	Ø.16839e Ø1	Ø.3Ø939e Ø2	Ø.24252e Ø1	Ø.53737e Ø1
4	Ø.11174e Ø1	Ø.17631e Ø2	Ø.27949e Ø1	Ø.14727e Ø2
5	Ø.1Ø244e Ø1	Ø.33385e Ø2	Ø.28879e Ø1	Ø.12297e Ø2
6	Ø.11614e Ø1	Ø.86967e Ø1	Ø.27553e Ø1	Ø.2418Øe Ø2

flamda= Ø.21362e-Ø3 tol= Ø.178Ø2e-Ø6 lamda= Ø.29149e Ø1

category	g(i)	theta(i)	e(i)
1	Ø.52133e Ø1	Ø.26353e Ø1	Ø.211Ø5e Ø2
2	Ø.4817Øe Ø1	Ø.23444e Ø1	Ø.1ØØ97e Ø2
3	Ø.488Ø1e Ø1	Ø.22892e Ø1	Ø.58126e Ø2
4	Ø.38546e Ø1	Ø.16317e Ø1	Ø.37944e Ø2
5	Ø.3Ø256e Ø1	Ø.1Ø679e Ø1	Ø.75Ø54e Ø2
6	Ø.11Ø2Øe Ø4	Ø.24617e Ø4	Ø.11728e Ø2

fdash=-Ø.214Ø6e Ø3 zlamda= Ø.29149e Ø1

10. ERROR MESSAGES

The following error messages may be encountered, after any of which the program stops (** or ## denote a number supplied in the message):

i. *Case 1*: 'negative value of alpha or beta parameter for category **'.

Cases 2 and 3: 'negative value of elasticity data for category **'.

ii. *Cases 1 and 3*: 'ratio beds available/beds needed is **; no allocation problem'. This message is generated if the ratio $r = B/\sum_i X_i U_i$ lies outside the range given by $0 < r < 1$.

iii. *Cases 1 and 3*: '## errors in xbig, ubig data for category **'.

Case 2: '## errors in xbar, ubar data for category **'. This message occurs if one or both of the *computed* column totals, XBARB and UBARB, of the input data does not lie within CTEST percent of the *input* values, XBARA and UBARA. An error in the length of stay data (UBIG(I) or UBAR(I)) will cause the number 1 to appear at ##, whereas an error in the admission data (XBIG(I) or XBAR(I)) *may* cause either 1 or 2 to appear.

iv. *Cases 2 and 3*: 'negative values of calculated parameters for category **'.

v. *All cases*: 'number of newton-raphson iterations exceeds maximum'.

vi. *Cases 2 and 3*: 'number of parameter iterations exceeds maximum'.

Errors (i), (ii) and (iii) occur as the result of data errors in the file 'INIDA'. Errors (iv), (v) and (vi) are usually caused by inappropriate values of parameters in the file 'AUXPA'. The reader should refer to Section 6 for guidance on suitable values for the data in these files.

Note that the *computed* column totals, XBARB and UBARB, are written to the terminal and to the file OUTPUT along with the input data rather than the *input* values XBARA and UBARA. In case of error (iii) this facilitates location of the error.

Indeed if, in a sequence of model runs, the user changes the values of the input quantities $\overline{x}_i$ and $\overline{u}_i$ in Mark 2 or X_i and U_i in Mark 3 without also changing the input values of the column totals then this error is likely to be encountered; in this case the correct values for the column totals will be those written to the terminal and to the file OUTPUT and these can be inserted to obtain a successful run.

REFERENCES

[1] Venedictov, D.D., and E.N. Shigan, *The IIASA Health Care System Model*, paper presented at IIASA Conference on Modeling Health Care Systems, Laxenburg, Austria, November 22-24, 1977.

[2] Gibbs, R.J., *A Disaggregated Health Care Resource Allocation Model*, RM-78-1, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1978.

[3] Gibbs, R.J., *The IIASA Health Care Resource Allocation Sub-Model: Mark 1*, Research Report, International Institute for Applied Systems Analysis, Laxenburg, Austria (forthcoming).

Papers of the Modeling Health Care Systems Study

April 1978

Venedictov, D.D., Modeling of Health Care Systems, in IIASA Conference '76, Vol.2, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1976.

Kiselev, A., A Systems Approach to Health Care, RM-75-31, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1975.

Fleissner, P., Comparing Health Care Systems by Socio-Economic Accounting, RM-76-19, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1976.

Klementiev, A.A., A Computer Method for Projecting a Population Sex-Age Structure, RM-76-36, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1976.

Klementiev, A.A., Mathematical Approach to Developing a Simulation Model of a Health Care System, RM-76-65, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1976.

Kaihara, S., et al., An Approach to Building a Universal Health Care Model: Morbidity Model of Degenerative Diseases, RM-77-06, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1976.

Shigan, E.N., Alternative Analysis of Different Methods for Estimating Prevalence Rate, RM-77-40, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1977.

Klementiev, A.A., On the Estimation of Morbidity, RM-77-43, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1977.

Fleissner, P., and A. Klementiev, Health Care System Models: A Review, RM-77-49, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1977.

Gibbs, R.J., Health Care Resource Allocation Models - A Critical Review, RM-77-53, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1977.

Gibbs, R.J., A Disaggregated Health Care Resource Allocation Model, RM-78-1, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1978.

Kaihara, S., et al., Analysis and Future Estimation of Medical Demands Using a Health Care Simulation Model: A Case Study of Japan, RM-77-3, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1978.

Fujimasa, I., S. Kaihara, and K. Atsumi, A Morbidity Submodel of Infectious Diseases, RM-78-10, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1978.