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

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 $\bar{B}$.

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.
U_i	UBIG[I]	Ideal average length of stay for category i	Computed parameters in Case 2.
α_i	ALPHA[I]	Power factor for patient numbers	Input data in Case 1.
β_i	BETA[I]	Power factor for length of stay	Computed parameters in Cases 2 and 3.
$\hat{\gamma}_i$	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	
C	Starting value for ratio $-\bar{B}/\lambda f'(\lambda)$	

*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)

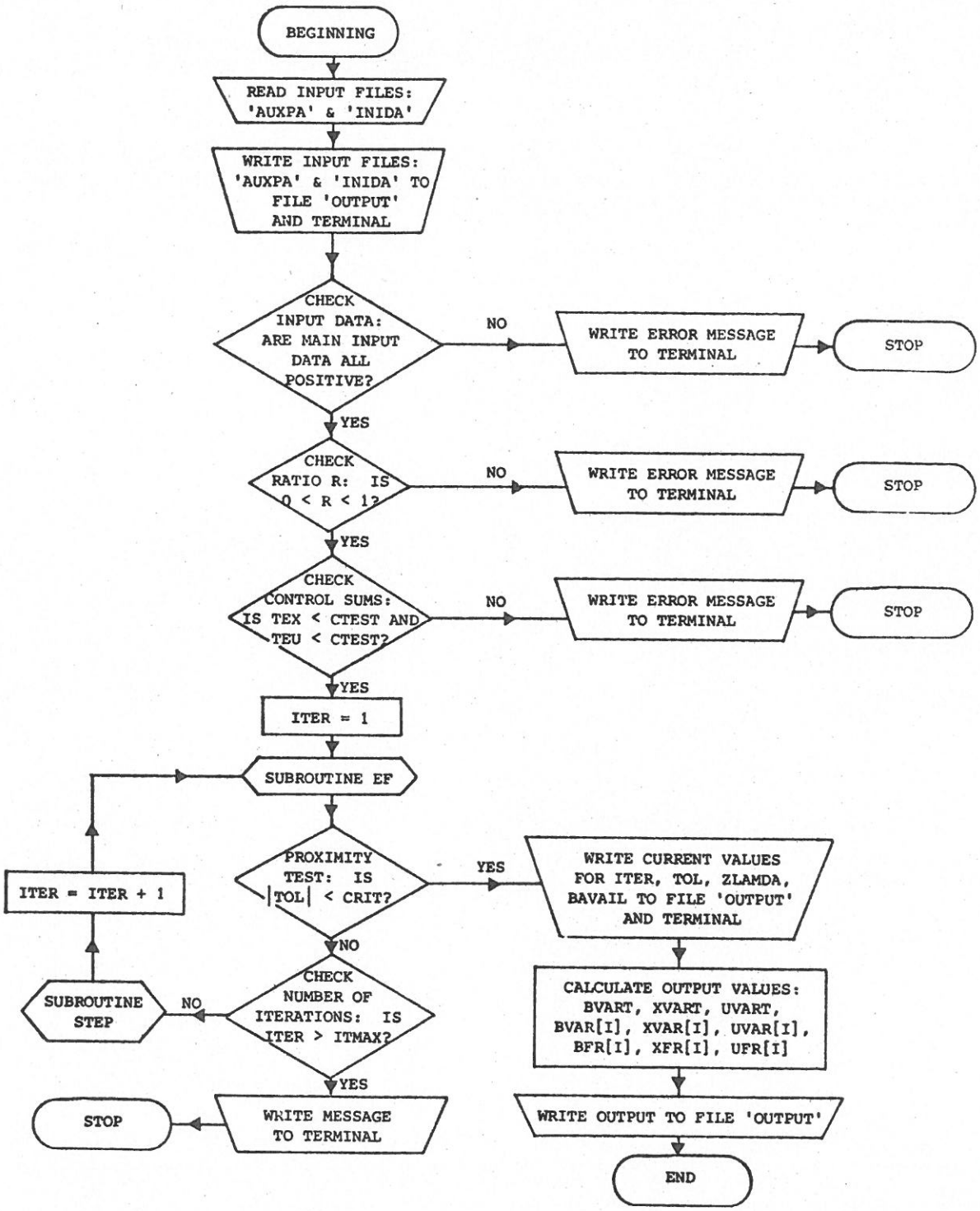
AUXILIARY QUANTITIES COMPUTED IN MAIN PROGRAM

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 ($= \bar{B} / \sum_i \bar{x}_i \bar{u}_i (\hat{\gamma}_i + \hat{\eta}_i)$
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 ($= \sum_i x_i u_i / \sum_i x_i$)
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 ($= \sum_i x_i$)

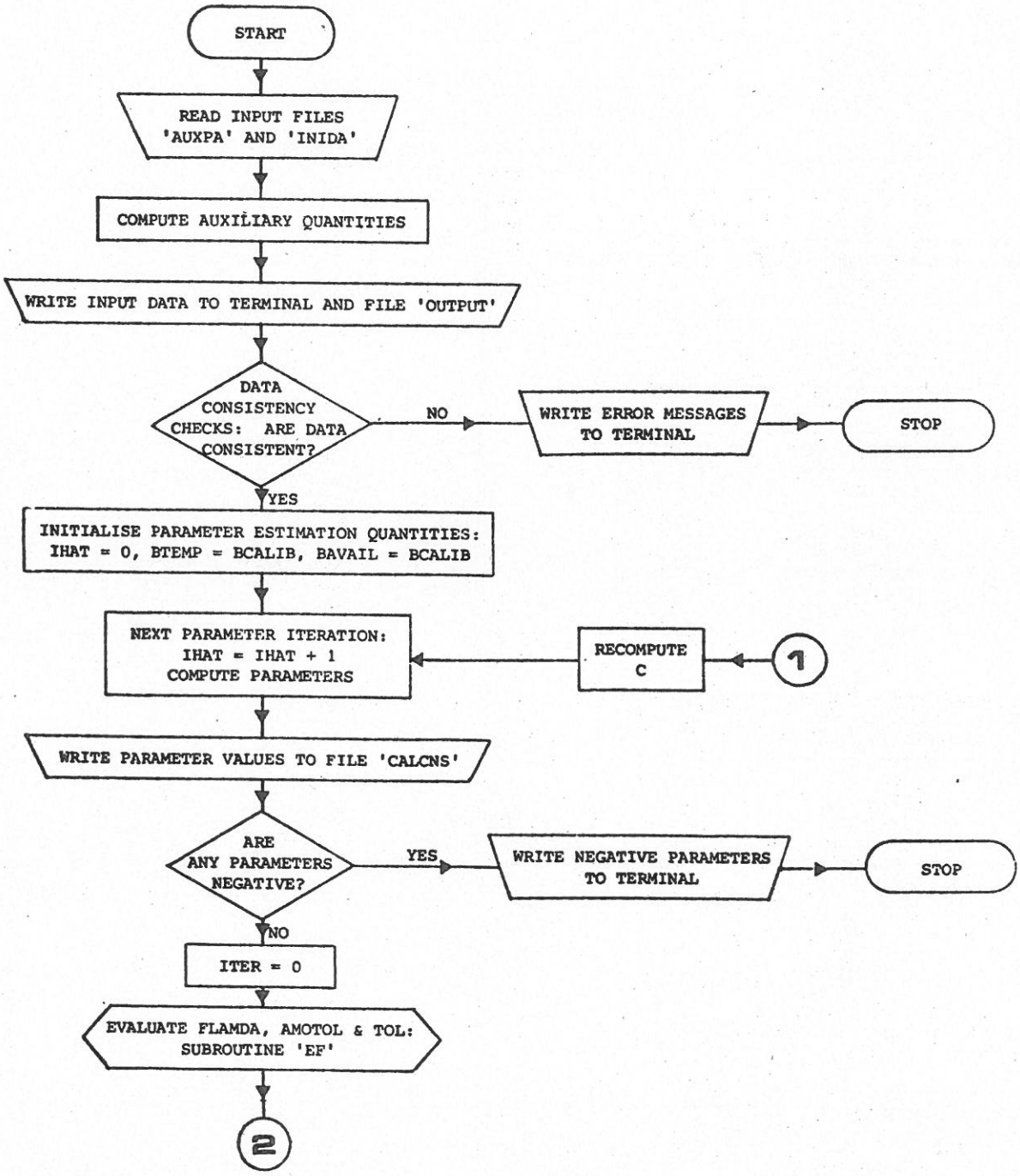
4. MAIN PROGRAMS--FLOW DIAGRAMS AND LISTINGS

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.

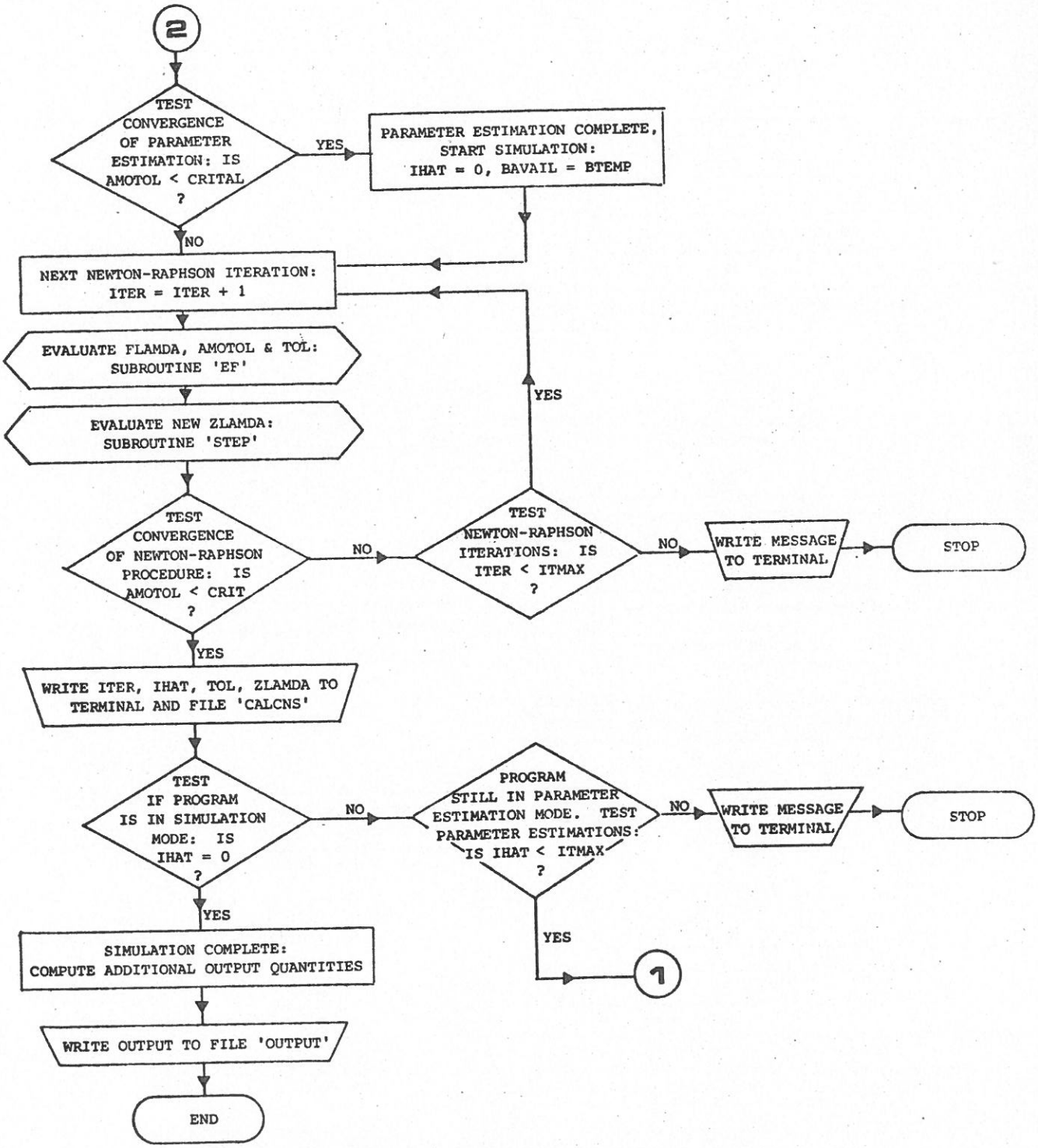


Main Program for Case 1*

*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 than 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


c1  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 c1

  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 = 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),xbar(20),ubar(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),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=0.0
      ubarb=0.0
      xtotb=0.0
      scale=0.0

      do 70 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)
   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),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

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)
```

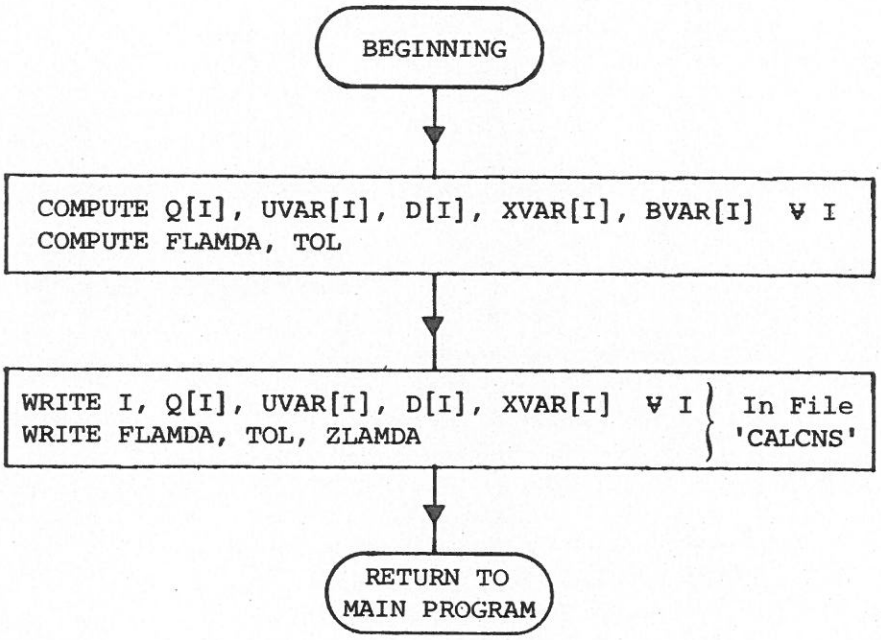
```
c   convergence test for newton-raphson procedure.
c   if test suceeds go to c2.
c   if test fails proceed to next iteration at c1, unless
c   iterations exceed maximum in which case stop.

      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. if process is in simulation mode ( ihat=Ø)
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=Ø.Ø
      bvart=Ø.Ø
      do 8Ø 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)
   8Ø 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))
```

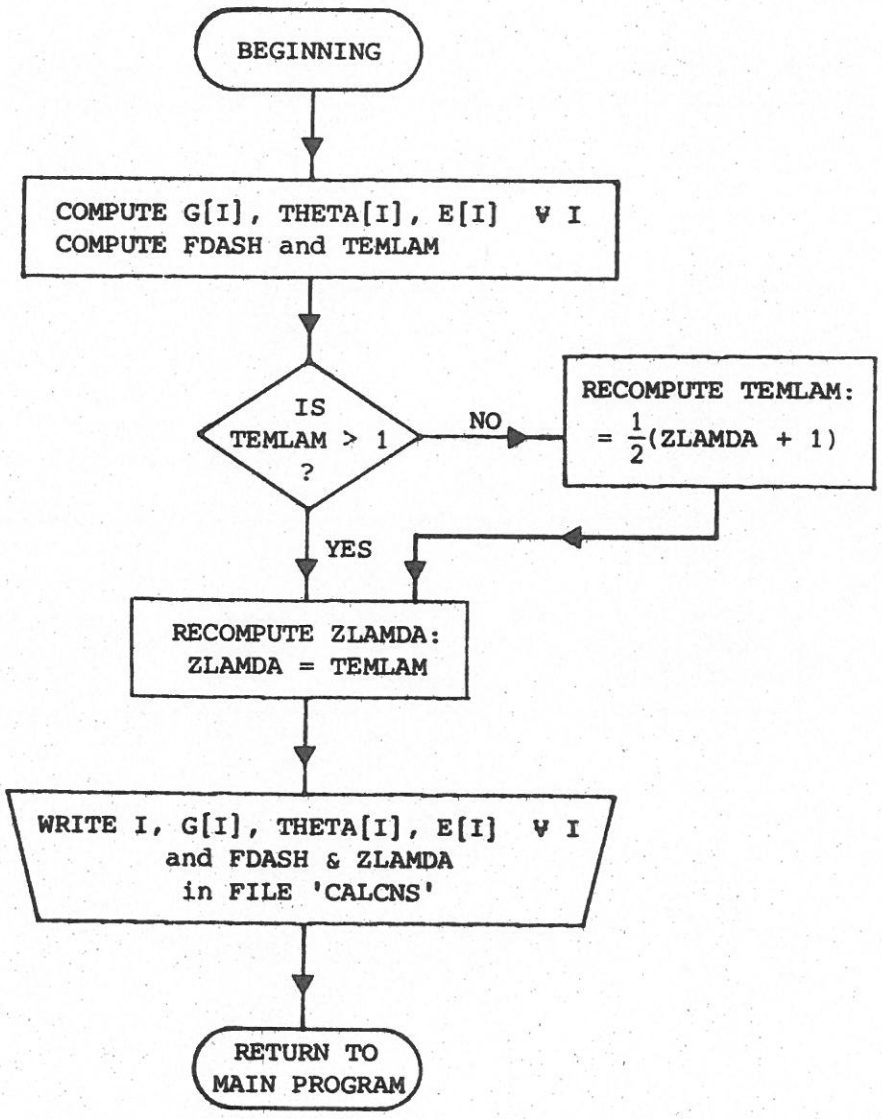
```
553 format(/' negative calculated parameter for category ',i2/
   -'alpha(i)= ',f12.6,'   beta(i)= ',f12.6)

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'

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

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

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

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

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

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

SUBROUTINE 'EF'

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

        common /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)


        flamda=-bavail
      do 51 i=1,ir
        q(i)=zlamda**((1.0/(1.0+beta(i))))
        uvar(i)=ubig(i)/q(i)
        d(i)=(ubig(i)/(a(i)*beta(i)))*((beta(i)+1.0)*(zlamda**(beta(i)
     -/(beta(i)+1.0)))-1.0)
        xvar(i)=xbig(i)/(d(i)**(1.0/(alpha(i)+1.0)))
        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=1,ir)
        write(iprint,4) flamda,tol,zlamda

   2    format(///10x,60('*')/' newton-raphson iteration ',i3//
     -'  category',10x,'q(i)',13x,'uvar(i)',10x,'d(i)',13x,'xvar(i)')

   3    format(4x,i2,11x,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 /cb1/ 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,
- C,
- BCALIB.

} required for Cases 2 and 3 only

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 20. 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 $\bar{B}$, 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:

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

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

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,	20,	6,

newton-raph. conv. crit. crit	initial lambda zlamda	data test ctest
0.00001 ,	4.000 ,	1.0

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	alpha(i)	beta(i)
varic. veins	12.8 ,	15.4 ,	1.636 ,	3.0323
haemorrhoids	7.7 ,	13.1 ,	2.1138 ,	4.6818
ischm. heart	10.4 ,	52.1 ,	0.5445 ,	1.3148
pneumonia	21.0 ,	19.7 ,	2.2808 ,	9.8696
bronchitis	21.3 ,	34.2 ,	1.1790 ,	49.000
appendicitis	24.8 ,	10.1 ,	44.433 ,	7.0645

OUTPUT

****

file=auxpa= for run test

identifiers for files =calcns= and =output= iprint	kprint	max no. of iterations itmax	no.of pat. categories ir	newton-raphson conv. criterion crit	initial value of lambda zlamda
2	3	20	6	0.0000100	4.000

****

file =inida= for run test

patient category	total sick xbig(i)	ideal length of stay ubig(i)	alpha alpha(i)	beta 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 iscnm. 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           0.14103e 01       0.10920e 02       0.34419e 01       0.80089e 01
    2           0.12763e 01       0.10264e 02       0.35898e 01       0.51077e 01
    3           0.18201e 01       0.28625e 02       0.31086e 01       0.49902e 01
    4           0.11360e 01       0.17341e 02       0.37765e 01       0.14006e 02
    5           0.10281e 01       0.33265e 02       0.39496e 01       0.11340e 02
    6           0.11876e 01       0.85048e 01       0.37035e 01       0.24096e 02

flamda=-0.92249e 02     tol=-0.76874e-01   lamda= 0.40000e 01


 category       g(i)              theta(i)          e(i)
    1        0.85187e 01       0.31472e 01       0.12257e 02
    2        0.76739e 01       0.27263e 01       0.59814e 01
    3        0.78394e 01       0.26932e 01       0.31773e 02
    4        0.57386e 01       0.17706e 01       0.22842e 02
    5        0.41956e 01       0.10789e 01       0.44518e 02
    6        0.91293e 04       0.14934e 05       0.73784e 01

 fdash=-0.12475e 03     zlamda= 0.32605e 01
```

****

newton-raphson iteration 2

category	q(i)	uvar(i)	d(i)	xvar(i)
1	0.13406e 01	0.11488e 02	0.29045e 01	0.85416e 01
2	0.12312e 01	0.10640e 02	0.30002e 01	0.54107e 01
3	0.16663e 01	0.31268e 02	0.26845e 01	0.54873e 01
4	0.11149e 01	0.17670e 02	0.31196e 01	0.14846e 02
5	0.10239e 01	0.33401e 02	0.32289e 01	0.12438e 02
6	0.11578e 01	0.87232e 01	0.30731e 01	0.24195e 02

flamda= 0.16108e 02 tol= 0.13423e-01 lamda= 0.32605e 01

category	g(i)	theta(i)	e(i)
1	0.62895e 01	0.28763e 01	0.17023e 02
2	0.57339e 01	0.25160e 01	0.81124e 01
3	0.59067e 01	0.25292e 01	0.47568e 02
4	0.44568e 01	0.16940e 01	0.30393e 02
5	0.33996e 01	0.10737e 01	0.60216e 02
6	0.23949e 04	0.48111e 04	0.93321e 01

fdash=-0.17265e 03 zlamda= 0.33538e 01

****

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	0.13502e 01	0.11406e 02	0.29754e 01	0.84637e 01
2	0.12375e 01	0.10586e 02	0.30776e 01	0.53666e 01
3	0.16872e 01	0.30880e 02	0.27415e 01	0.54133e 01
4	0.11178e 01	0.17623e 02	0.32051e 01	0.14724e 02
5	0.10245e 01	0.33382e 02	0.33222e 01	0.12277e 02
6	0.11620e 01	0.86920e 01	0.31554e 01	0.24181e 02

flamda=-0.91553e-04 tol=-0.76294e-07 lamda= 0.33560e 01

8. CASE 2 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 ,	2.2200,	1094.2

INIDA

run name	total mean no. patients	overall mean av.stay	beds available
test	63.9 ,	17.123000,	1200.0

patient category	mea no patients	mean av.stay	elasticities gamhat(i)	etahat(i)
varic. veins	6.3 ,	11.3 ,	0.78 ,	0.62
haemorrhoids	4.1 ,	13.1 ,	0.70 ,	0.44
ischm. heart	4.6 ,	40.2 ,	1.14 ,	1.08
pneumonia	12.3 ,	14.7 ,	0.71 ,	0.23
bronchitis	11.8 ,	27.4 ,	1.13 ,	0.05
appendicitis	24.8 ,	11.3 ,	0.05 ,	0.31

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	q(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	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=-0.10581e 03 tol=-0.88175e-01 lamda= 0.39999e 01

category	g(i)	theta(i)	e(i)
1	0.85147e 01	0.31442e 01	0.10035e 02
2	0.76704e 01	0.27239e 01	0.61651e 01
3	0.78359e 01	0.26902e 01	0.41335e 02
4	0.57367e 01	0.17695e 01	0.17113e 02
5	0.41951e 01	0.10787e 01	0.38414e 02
6	0.91405e 04	0.14954e 05	0.10158e 02

fdash=-0.12322e 03 zlamda= 0.31412e 01

****

newton-raphson iteration 2

category	q(i)	uvar(i)	d(i)	xvar(i)
1	0.13308e 01	0.12003e 02	0.28130e 01	0.68018e 01
2	0.12249e 01	0.13673e 02	0.29015e 01	0.43910e 01
3	0.16452e 01	0.44656e 02	0.26093e 01	0.51502e 01
4	0.11119e 01	0.15033e 02	0.30115e 01	0.13183e 02
5	0.10233e 01	0.27534e 02	0.31122e 01	0.13173e 02
6	0.11536e 01	0.11646e 02	0.29687e 01	0.24922e 02

flamda= 0.22776e 02 tol= 0.18980e-01 lamda= 0.31412e 01

category	g(i)	theta(i)	e(i)
1	0.59475e 01	0.28272e 01	0.14815e 02
2	0.54343e 01	0.24776e 01	0.88476e 01
3	0.56056e 01	0.24980e 01	0.66727e 02
4	0.42546e 01	0.16795e 01	0.24005e 02
5	0.32714e 01	0.10726e 01	0.54946e 02
6	0.18782e 04	0.39177e 04	0.13414e 02

fdash=-0.18275e 03 zlamda= 0.32658e 01

****

newton-raphson iteration 3

category	q(i)	uvar(i)	d(i)	xvar(i)
1	0.13438e 01	0.11887e 02	0.29062e 01	0.67177e 01
2	0.12333e 01	0.13579e 02	0.30029e 01	0.43425e 01
3	0.16733e 01	0.43906e 02	0.26844e 01	0.50559e 01
4	0.11159e 01	0.14979e 02	0.31234e 01	0.13036e 02
5	0.10241e 01	0.27512e 02	0.32339e 01	0.12941e 02
6	0.11592e 01	0.11590e 02	0.30765e 01	0.24902e 02

flamda= 0.71857e 00 tol= 0.59881e-03 lamda= 0.32658e 01

category	g(i)	theta(i)	e(i)
1	0.63029e 01	0.28763e 01	0.13910e 02
2	0.57455e 01	0.25158e 01	0.83462e 01
3	0.59188e 01	0.25284e 01	0.61742e 02
4	0.44648e 01	0.16937e 01	0.22729e 02
5	0.34051e 01	0.10736e 01	0.51868e 02
6	0.24235e 04	0.48612e 04	0.12826e 02

fdash=-0.17142e 03 zlamda= 0.32700e 01

****

newton-raphson iteration 4

category	q(i)	uvar(i)	d(i)	xvar(i)
1	0.13443e 01	0.11883e 02	0.29094e 01	0.67149e 01
2	0.12336e 01	0.13576e 02	0.30063e 01	0.43409e 01
3	0.16742e 01	0.43882e 02	0.26869e 01	0.50528e 01
4	0.11160e 01	0.14977e 02	0.31272e 01	0.13031e 02
5	0.10241e 01	0.27511e 02	0.32380e 01	0.12934e 02
6	0.11594e 01	0.11588e 02	0.30801e 01	0.24901e 02

flamda= 0.61035e-03 tol= 0.50863e-06 lamda= 0.32700e 01

category	g(i)	theta(i)	e(i)
1	0.63150e 01	0.28779e 01	0.13882e 02
2	0.57561e 01	0.25171e 01	0.83301e 01
3	0.59294e 01	0.25294e 01	0.61584e 02
4	0.44719e 01	0.16942e 01	0.22688e 02
5	0.34096e 01	0.10736e 01	0.51769e 02
6	0.24439e 04	0.48959e 04	0.12807e 02

fdash=-0.17106e 03 zlamda= 0.32700e 01

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

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         2.22760         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 = 0.14e-06 lamda = 3.489

parameter iteration 2 completed with 2 newton-raphson iterations
tolerance = 0.42e-07 lamda = 3.490

parameter iteration 0 completed with 4 newton-raphson iterations
tolerance = 0.18e-06 lamda = 2.915

calculated parameter values for run test

i	category	alpha(i)	beta(i)
1	varic. veins	1.310	2.576
2	haemorrhoids	1.738	4.039
3	ischm. heart	0.342	1.053
4	pneumonia	1.897	8.639
5	bronchitis	0.931	43.342
6	appendicitis	39.053	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 bronchitis	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	0.14090e 01	0.10929e 02	0.29856e 01	0.79490e 01
2	0.12755e 01	0.10270e 02	0.31031e 01	0.50865e 01
3	0.18172e 01	0.28670e 02	0.27207e 01	0.48722e 01
4	0.11357e 01	0.17347e 02	0.32511e 01	0.13985e 02
5	0.10280e 01	0.33267e 02	0.33883e 01	0.11349e 02
6	0.11870e 01	0.85087e 01	0.31932e 01	0.24091e 02

flamda= 0.97266e 01 tol= 0.88892e-02 lamda= 0.34281e 01

category	g(i)	theta(i)	e(i)
1	0.65609e 01	0.27826e 01	0.16048e 02
2	0.60315e 01	0.24730e 01	0.78427e 01
3	0.59855e 01	0.23280e 01	0.41159e 02
4	0.47015e 01	0.16840e 01	0.29964e 02
5	0.35748e 01	0.10717e 01	0.58392e 02
6	0.30602e 04	0.57815e 04	0.96717e 01

fdash=-0.16308e 03 zlamda= 0.34878e 01

****

newton-raphson iteration 3

category	q(i)	uvar(i)	d(i)	xvar(i)
1	0.14158e 01	0.10877e 02	0.30279e 01	0.79005e 01
2	0.12799e 01	0.10235e 02	0.31497e 01	0.50587e 01
3	0.18325e 01	0.28431e 02	0.27534e 01	0.48283e 01
4	0.11377e 01	0.17316e 02	0.33036e 01	0.13908e 02
5	0.10284e 01	0.33254e 02	0.34463e 01	0.11250e 02
6	0.11899e 01	0.84883e 01	0.32434e 01	0.24082e 02

flamda= 0.13782e 00 tol= 0.12595e-03 lamda= 0.34878e 01

category	g(i)	theta(i)	e(i)
1	0.67274e 01	0.28019e 01	0.15588e 02
2	0.61794e 01	0.24886e 01	0.76354e 01
3	0.61247e 01	0.23377e 01	0.39694e 02
4	0.48021e 01	0.16899e 01	0.29225e 02
5	0.36388e 01	0.10720e 01	0.56865e 02
6	0.34219e 04	0.63536e 04	0.94789e 01

fdash=-0.15849e 03 zlamda= 0.34887e 01

****

newton-raphson iteration 4

category	q(i)	uvar(i)	d(i)	xvar(i)
1	0.14159e 01	0.10876e 02	0.30285e 01	0.78998e 01
2	0.12799e 01	0.10235e 02	0.31504e 01	0.50583e 01
3	0.18327e 01	0.28428e 02	0.27539e 01	0.48277e 01
4	0.11377e 01	0.17316e 02	0.33043e 01	0.13907e 02
5	0.10284e 01	0.33254e 02	0.34471e 01	0.11249e 02
6	0.11899e 01	0.84880e 01	0.32441e 01	0.24082e 02

flamda= 0.15259e-03 tol= 0.13945e-06 lamda= 0.34887e 01

category	g(i)	theta(i)	e(i)
1	0.67299e 01	0.28022e 01	0.15582e 02
2	0.61816e 01	0.24889e 01	0.76324e 01
3	0.61267e 01	0.23378e 01	0.39673e 02
4	0.48036e 01	0.16900e 01	0.29215e 02
5	0.36397e 01	0.10721e 01	0.56843e 02
6	0.34274e 04	0.63623e 04	0.94761e 01

fdash=-0.15842e 03 zlamda= 0.34887e 01

****

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	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

category	g(i)	theta(i)	e(i)
1	0.67846e 01	0.28370e 01	0.15577e 02
2	0.62097e 01	0.25062e 01	0.76315e 01
3	0.62258e 01	0.23977e 01	0.39801e 02
4	0.48089e 01	0.16930e 01	0.29236e 02
5	0.36396e 01	0.10720e 01	0.56916e 02
6	0.35487e 04	0.66155e 04	0.95062e 01

fdash=-0.15867e 03 zlamda= 0.34898e 01

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

 category          q(i)              xvar(i)           d(i)              xvar(i)
    1          0.14184e 01       0.10857e 02       0.30273e 01       0.79245e 01
    2          0.12815e 01       0.10222e 02       0.31498e 01       0.50643e 01
    3          0.18383e 01       0.28342e 02       0.27517e 01       0.48911e 01
    4          0.11384e 01       0.17304e 02       0.33044e 01       0.13900e 02
    5          0.10286e 01       0.33249e 02       0.34480e 01       0.11218e 02
    6          0.11910e 01       0.84806e 01       0.32440e 01       0.24082e 02

flamda= 0.45776e-04     tol= 0.41835e-07   lamda= 0.34898e 01


 category       g(i)              theta(i)          e(i)
    1       0.67877e 01       0.28373e 01       0.15569e 02
    2       0.62124e 01       0.25065e 01       0.76278e 01
    3       0.62284e 01       0.23979e 01       0.39775e 02
    4       0.48108e 01       0.16931e 01       0.29223e 02
    5       0.36408e 01       0.10720e 01       0.56889e 02
    6       0.35558e 04       0.66268e 04       0.95027e 01

 fdash=-0.15859e 03     zlamda= 0.34898e 01
```

****

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	0.13367e 01	0.11521e 02	0.25432e 01	0.85453e 01
2	0.12287e 01	0.10662e 02	0.26185e 01	0.54178e 01
3	0.16578e 01	0.31428e 02	0.23700e 01	0.54667e 01
4	0.11137e 01	0.17690e 02	0.27122e 01	0.14881e 02
5	0.10237e 01	0.33409e 02	0.27979e 01	0.12501e 02
6	0.11561e 01	0.87359e 01	0.26757e 01	0.24198e 02

flamda= 0.20287e 02 tol= 0.16906e-01 lamda= 0.28226e 01

category	g(i)	theta(i)	e(i)
1	0.49718e 01	0.26004e 01	0.22291e 02
2	0.46020e 01	0.23164e 01	0.10618e 02
3	0.46698e 01	0.22699e 01	0.62241e 02
4	0.37046e 01	0.16209e 01	0.39761e 02
5	0.29272e 01	0.10672e 01	0.78866e 02
6	0.89376e 03	0.20622e 04	0.12178e 02

fdash=-0.22596e 03 zlamda= 0.29124e 01

****

newton-raphson iteration 3

category	q(i)	uvar(i)	d(i)	xvar(i)
1	0.13484e 01	0.11421e 02	0.26101e 01	0.84498e 01
2	0.12363e 01	0.10596e 02	0.26913e 01	0.53637e 01
3	0.16833e 01	0.30952e 02	0.24238e 01	0.53761e 01
4	0.11173e 01	0.17632e 02	0.27927e 01	0.14732e 02
5	0.10244e 01	0.33385e 02	0.28855e 01	0.12303e 02
6	0.11612e 01	0.86978e 01	0.27532e 01	0.24181e 02

flamda= 0.52843e 00 tol= 0.44036e-03 lamda= 0.29124e 01

category	g(i)	theta(i)	e(i)
1	0.52068e 01	0.26343e 01	0.21135e 02
2	0.48112e 01	0.23437e 01	0.10111e 02
3	0.48744e 01	0.22887e 01	0.58231e 02
4	0.38506e 01	0.16314e 01	0.37991e 02
5	0.30230e 01	0.10679e 01	0.75152e 02
6	0.10959e 04	0.24502e 04	0.11740e 02

fdash=-0.21436e 03 zlamda= 0.29149e 01

****

newton-raphson iteration 4

category	q(i)	uvar(i)	d(i)	xvar(i)
1	0.13487e 01	0.11418e 02	0.26119e 01	0.84472e 01
2	0.12365e 01	0.10594e 02	0.26933e 01	0.53623e 01
3	0.16839e 01	0.30939e 02	0.24252e 01	0.53737e 01
4	0.11174e 01	0.17631e 02	0.27949e 01	0.14727e 02
5	0.10244e 01	0.33385e 02	0.28879e 01	0.12297e 02
6	0.11614e 01	0.86967e 01	0.27553e 01	0.24180e 02

flamda= 0.21362e-03 tol= 0.17802e-06 lamda= 0.29149e 01

category	g(i)	theta(i)	e(i)
1	0.52133e 01	0.26353e 01	0.21105e 02
2	0.48170e 01	0.23444e 01	0.10097e 02
3	0.48801e 01	0.22892e 01	0.58126e 02
4	0.38546e 01	0.16317e 01	0.37944e 02
5	0.30256e 01	0.10679e 01	0.75054e 02
6	0.11020e 04	0.24617e 04	0.11728e 02

fdash=-0.21406e 03 zlamda= 0.29149e 01

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 $\bar{x}_i$ and $\bar{u}_i$ 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.