INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-280

MINI5W - A FORTRAN PACKAGE FOR MINIMAX OPTIMIZATION

K. Madsen

(Adapted and Edited by J.W. Bandler and W.M. Zuberek)

December 1981

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

No. SOC-280

MINI5W - A FORTRAN PACKAGE FOR MINIMAX OPTIMIZATION

K. Madsen

(Adapted and Edited by J.W. Bandler and W.M. Zuberek)

December 1981

MINI5W - A FORTRAN PACKAGE FOR MINIMAX OPTIMIZATION

K. Madsen

Adapted and Edited by

J.W. Bandler and W.M. Zuberek

Abstract

MINI5W is a package of subroutines for solving unconstrained, nonlinear minimax optimization problems. First derivatives of all functions w.r.t. all variables are assumed to be known. The solution is found by an iteration that uses either linear programming applied in connection with first derivatives or a Newton step applied in connection with first derivatives and approximate second derivatives. The method has been described by Hald and Madsen. The Fortran IV package and documentation have been adapted for the CDC 170/730 system.

The work of verifying, editing and adapting the original material due to K. Madsen was supported by the Natural Sciences and Engineering Research Council of Canada under Grant G0647.

K. Madsen is with the Institute of Datalogy, University of Copenhagen, Copenhagen, Denmark.

J.W. Bandler and W.M. Zuberek are with the Group on Simulation, Optimization and Control and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7.

W.M. Zuberek is on leave from the Institute of Computer Science, Technical University of Warsaw, Warsaw, Poland.

I. INTRODUCTION

Prepared by J.W. Bandler and W.M. Zuberek

This report gives a user-oriented description of a program package for unconstrained minimax optimization of absolute values of a set of differentiable nonlinear functions. The package has been developed in Fortran IV by Kaj Madsen at the Institute of Datalogy of the University of Copenhagen* and has been adapted for the CDC 170/730 (System B) installation at McMaster University. Sections II, IV-VII contain the body of Madsen's description edited and arranged for use at McMaster University. Also given is the listing (Appendix) and tests of the package.

The package is available as a permanent group file in the form of a library of binary relocatable subroutines. The name of the library is LIBRMMU. The package is linked with the user's program by the appropriate call of the main subroutine of the package, namely, subroutine MINI5W. The general sequence of NOS commands to use the package can be as follows:

```
/GET(LIBRMMU/GR)   - fetch the library LIBRMMU,
/LIBRARY(LIBRMMU) - indicate the library to the loader,
/FTN(...,GO)       - compile, load and execute the program.
```

---

* K. Madsen, "Documentation of the Fortran subroutine MINI5W for minimax optimization", Inst. of Datalogy, University of Copenhagen, Copenhagen, Denmark, April 1981.

The user must prepare programs which should be composed (at least) of:

- the main segment, which prepares parameters and calls the main subroutine of the package (subroutine MINI5W),
- the segment which calculates the values of the nonlinear functions and their first derivatives w.r.t. all variables at points determined by the package; the name of this subroutine can be arbitrary because it is transferred to the package as one of the parameters.

II. GENERAL DESCRIPTION

The purpose of the package is to find a local minimum of the minimax objective function

$$F(\underset{\sim}{x}) = \max_{1 \leq j \leq m} |f_j(\underset{\sim}{x})| ,$$

where $f_j(\underset{\sim}{x})$, $j = 1, \ldots, m$, is a set of nonlinear functions of the n-dimensional vector of variables $\underset{\sim}{x}$. First-order derivatives w.r.t. $\underset{\sim}{x}$ are assumed to be available.

The solution is found by an iteration process that uses either linear programming applied in connection with the first derivatives, or a Newton step applied in connection with first derivatives and approximate second derivatives. For a description, see Hald and Madsen [1].

III. STRUCTURE OF THE PACKAGE

Prepared by J.W. Bandler and W.M. Zuberek

A block diagram of the package is shown in Fig. 1. The subroutine MINI5W is the main subroutine, the aim of which is only to subdivide the work space (defined by user) into a set of vectors and arrays used by

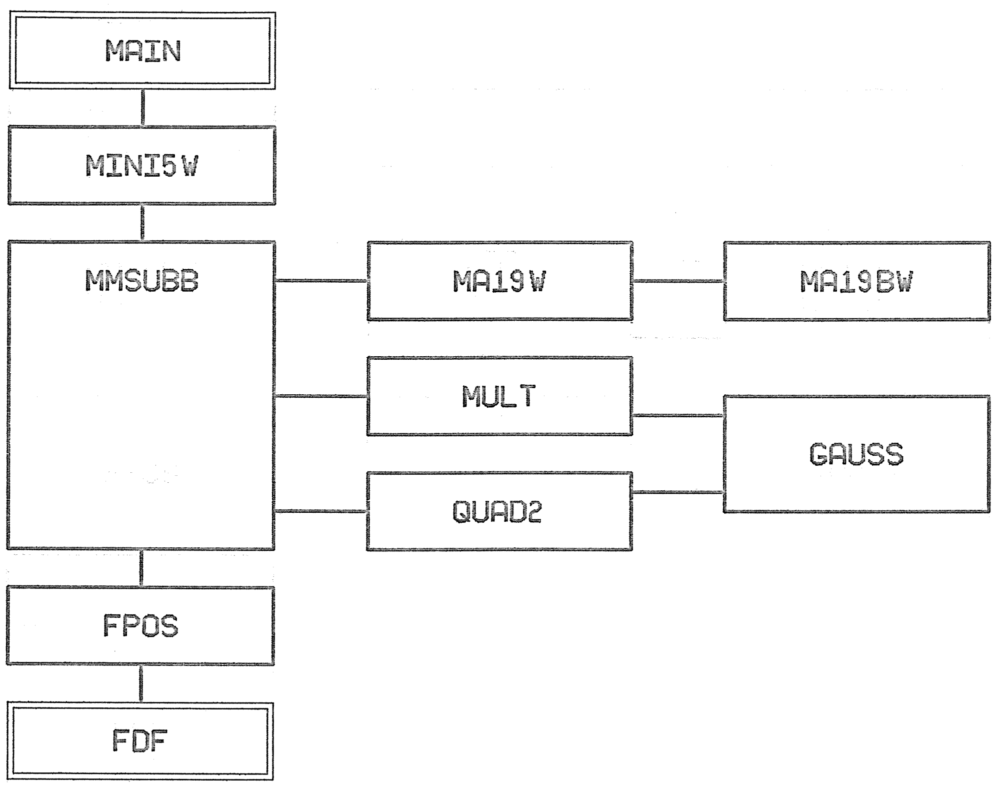

Fig. 1 Structure of the MINI5W package for unconstrained minimax optimization of absolute values of a set of nonlinear functions.

the package. Minimax optimization is performed by MMSUBB, which calls FPOS (and FDF) for evaluation of the nonlinear functions and their first derivatives. Linear subproblems are solved by MA19BW and quadratic subproblems by QUAD2. MA19W is only used to simplify the call of MA19BW. MULT determines Lagrangian multipliers for linear subproblems. GAUSS is used to solve a system of linear equations.

The main program MAIN and the subroutine FDF must be supplied by the user.

IV. LIST OF ARGUMENTS

The main subroutine call is

```
CALL MINI5W (FDF,N,M,X,DX,EPS,MAXFUN,W,IW)
```

The arguments of this call statement are defined as follows.

FDF is the name of a subroutine written by the user. It must have the form

```
SUBROUTINE FDF(N,M,X,DF,F)
REAL X(N), DF(M,N), F(M)
```

and it must calculate the values of the nonlinear functions and their derivatives at the point $\underset{\sim}{x}$ corresponding to X(1), X(2),...,X(N), and store these in the following way:

$F(J) = f_J(\underset{\sim}{x})$, $J = 1,\ldots,M$,

$DF(J,I) = \partial f_J/\partial x_I(\underset{\sim}{x})$, $J = 1,\ldots,M$, $I = 1,\ldots,N$.

Note: The name of this user-supplied subroutine, which can be any name of the user's choice, must appear in an EXTERNAL statement in the calling program.

N is an INTEGER variable and must be set to n, the number of optimization parameters. Its value must be positive, and it is not changed by the package.

M is the INTEGER variable and must be set to m, the number of nonlinear functions defining the minimax objective function. Its value must be positive, and it is not changed by the package.

X is a REAL array of length at least N. <u>On entry</u> it must be set by the user to an initial approximation of the solution, X(I) = x_I^0, I=1,...,N. <u>On exit</u> X contains the best solution found by the package.

DX is a REAL variable which controls the step length of the iterative method used. It must be set by the user to an initial value corresponding to the starting vector $\underset{\sim}{x}^0$. DX should be chosen so that in the region $\{\underset{\sim}{x} \mid \|\underset{\sim}{x}-\underset{\sim}{x}^0\| < DX\}$ the functions f_j can be approximated reasonably well by linear functions. If the functions are nearly linear, DX should be set to an approximate value of the distance between the starting vector and the solution, but if more curvature is present this value may be too large. In general, DX = $0.1*\|\underset{\sim}{x}^0\|$ is a reasonable choice. However, it is normally not severe to choose a bad initial value of DX, since DX is adjusted by the package during the iteration. The value of DX must be positive.

EPS is a REAL variable which must be set by the user to indicate the required accuracy of the solution. The iteration is stopped when the change $\underset{\sim}{h}^k$ of the approximate solution $\underset{\sim}{x}^k$ is smaller than EPS*$\|\underset{\sim}{x}^k\|$. If EPS is chosen too small, the package will stop when the calculation is dominated by rounding errors, and EPS will be set to 0.

MAXFUN is an INTEGER variable. It must be set by the user, and its value gives an upper bound for the number of calls of FDF. If the number of calls required exceeds MAXFUN, the package will return and W(M+1) will be set to (MAXFUN+1).

W is a REAL array which is used for workspace. Its length must be at least

(16*N+4*M+2*M*N+2*N**2+max{M,3*N**2+6*N+5}+13).

On entry W(1) must be set to a positive value δ which is used to determine the set of active functions at each iterate $\underset{\sim}{x}^k$. If

$$|f_j(\underset{\sim}{x}^k) - F(\underset{\sim}{x}^k)| \leq \delta * |F(\underset{\sim}{x}^k)|$$

then f_j goes into the set of active functions. Normally δ = W(1) = 0.01 is an appropriate value. W(2) must be set to the number of iterations with identical set of active functions that is required before a Newton iteration is tried (see reference [1]). Normally W(2) = 3 is an appropriate value. If W(2) $\geq$ MAXFUN then the Newton iteration is never used. On exit W will contain the function values at the solution, i.e.,

$$W(J) = f_J(\underset{\sim}{x}), \quad J = 1,..,M.$$

Further, W(M+1) will give the number of calls of FDF used (see variable MAXFUN).

IW is an INTEGER variable which must be set to the length of W.

V. GENERAL INFORMATION

Use of COMMON:	None.
Workspace:	Provided by the user; see arguments W and IW.
Other subroutines:	MMSUBB, QUAD2, MULT, FPOS, GAUSS, MA19W, MA19BW.
Input/Output:	None.
Restrictions:	$N \geq 1$, $M \geq 1$, $DX > 0$, $EPS \geq 0$, $MAXFUN > 0$.
Date:	March 1981.

VI. EXAMPLES

Example 1

Minimize

$$F(\underset{\sim}{x}) = \max(|f_1(\underset{\sim}{x})|, |f_2(\underset{\sim}{x})|),$$

where

$$f_1(\underset{\sim}{x}) = x_1^2 + 2x_2^2 + x_1x_2,$$

$$f_2(\underset{\sim}{x}) = \sin(x_1) + \cos(x_2).$$

For this example,

$$N = 2$$

$$M = 2$$

$$IW = 98$$

The starting point is

$$\underset{\sim}{x}^0 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

```
      PROGRAM T1RMMU(OUTPUT,TAPE3=OUTPUT)
C
C    K.MADSEN - EXAMPLE
C
      DIMENSION X(2),W(98)
      EXTERNAL DFF
      COMMON NCOUNT
      NCOUNT=0
      N=2
      M=2
      X(1)=3.
      X(2)=1.
      DX=1.0
      EPS=1.0E-10
      MAXF=30
      IW=98
      W(1)=0.01
      W(2)=3.0
      WRITE(3,100)
  100 FORMAT(" PROGRAM T1RMMU (K.MADSEN EXAMPLE)"//
     1 13X,"X1",12X,"X2",15X,"F1",12X,"F2")
      CALL SECOND(TM1)
      CALL MINI5W(DFF,N,M,X,DX,EPS,MAXF,W,IW)
      CALL SECOND(TM2)
      CPU=TM2-TM1
      LCALL=W(M+1)
      WRITE(3,200) CPU
  200 FORMAT(/" CPU TIME:",F8.3," SECONDS")
      WRITE(3,300) EPS,LCALL
  300 FORMAT(/" EPS VAL=",E13.6/" F.EVAL.=",I5)
      WRITE(3,400) (X(I),I=1,N),(W(I),I=1,M)
  400 FORMAT(/" SOLUTION:",2(/F20.10)//" FUNCTION VALUES:",2(/F20.10)//)
      STOP
      END
C
C
      SUBROUTINE DFF(N,M,X,DF,F)
      DIMENSION X(N),DF(M,N),F(M)
      COMMON NCOUNT
      F(1)=X(1)**2+2*X(2)**2+X(1)*X(2)
      F(2)=SIN(X(1))+COS(X(2))
      DF(1,1)=2*X(1)+X(2)
      DF(1,2)=4*X(2)+X(1)
      DF(2,1)=COS(X(1))
      DF(2,2)=-SIN(X(2))
      NCOUNT=NCOUNT+1
      WRITE(3,100) NCOUNT,(X(I),I=1,N),(F(I),I=1,M)
  100 FORMAT(1X,I5,2(F13.8,1X),1X,2(1X,F13.8))
      RETURN
      END
```

PROGRAM T1RMMUB(K.MADSEN EXAMPLE)

	X1	X2	F1	F2
1	3.00000000	1.00000000	14.00000000	.68142231
2	2.31450415	0.00000000	5.35692946	1.73596335
3	2.06828201	-1.00000000	4.20950847	1.41908752
4	1.06828201	-1.29690028	3.11967189	1.14685866
5	.06828201	-.97915801	1.85530438	.62595058
6	-.93171799	-.32302946	1.37776682	.14563227
7	-1.40213805	.67697054	1.93336319	-.20633683
8	-.94555372	.17697054	.78937382	.17356040
9	-.69143806	.68801162	.94908915	.13486454
10	-.69555372	.42697054	.55142171	.26941384
11	-.73863115	.17697054	.47749717	.31110511
12	-.63924467	.42697054	.50030280	.31363504
13	-.64814920	.30197054	.40674784	.35104032
14	-.67685894	.17697054	.40099108	.35803401
15	-.64828496	.23947054	.37972052	.36764349
16	-.64244065	.23720201	.37287137	.37284796
17	-.64232155	.23756544	.37285825	.37285786
18	-.64233723	.23751138	.37285803	.37285802
19	-.64233723	.23751138	.37285803	.37285803
20	-.64233723	.23751138	.37285803	.37285803

CPU TIME: .151 SECONDS

EPS VAL= .100000E-09
F.EVAL.= 20

SOLUTION:
-.6423372301
.2375113809

FUNCTION VALUES:
.3728580268
.3728580268

<u>Example 2</u> (Prepared by J.W. Bandler and W.M. Zuberek)

This is the design of a 3-section 100-percent relative bandwidth 10:1 transmission-line transformer problem [2,3]. In this case the error functions f_i represent the modulus of the reflection coefficient sampled at the 11 normalized frequencies (w.r.t. 1 GHz)

$$\{ 0.5, 0.6, 0.7, 0.77, 0.9, 1.0, 1.1, 1.23, 1.3, 1.4, 1.5 \}.$$

Gradient vectors with respect to section lengths and characteristic impedances are obtained using the adjoint network method.

The known quarter-wave solution is given by

$$\ell_1 = \ell_2 = \ell_3 = \ell_q,$$
$$Z_1 = 1.63471,$$
$$Z_2 = 3.16228,$$
$$Z_3 = 6.11729,$$

where ℓ_q is the quarter wavelength at centre frequency, namely

$$\ell_q = 7.49481 \text{ cm for 1 GHz.}$$

The corresponding maximum reflection coefficient is 0.19729. The vector of optimization parameters is

$$\underset{\sim}{x} = \begin{bmatrix} \ell_1/\ell_q \\ Z_1 \\ \ell_2/\ell_q \\ Z_2 \\ \ell_3/\ell_q \\ Z_3 \end{bmatrix}.$$

For this example,

$$N = 6$$
$$M = 11$$
$$IW = 506$$

The starting point is

```
      PROGRAM T2RMMU(OUTPUT,TAPE1=OUTPUT)
C
C     3-SECTION MICROWAVE TRANSMISSION-LINE TRANSFORMER
C
      DIMENSION X(6),W(506)
      EXTERNAL FGR
      COMMON NCOUNT,RG,RL,FREQ(11)
      DATA X/0.8,1.5,1.2,3.0,0.8,6.0/
      NCOUNT=0
      RG=1.0
      RL=10.0
      FREQ(1)=0.5
      FREQ(2)=0.6
      FREQ(3)=0.7
      FREQ(4)=0.77
      FREQ(5)=0.9
      FREQ(6)=1.0
      FREQ(7)=1.1
      FREQ(8)=1.23
      FREQ(9)=1.3
      FREQ(10)=1.4
      FREQ(11)=1.5
      N=6
      M=11
      DX=0.1
      EPS=1.0E-7
      MAXF=150
      AX=0.01
      KEQS=2
      W(1)=AX
      W(2)=KEQS
      IW=506
      WRITE(1,100)
  100 FORMAT(" PROGRAM T2RMMU (3-SECTION MICROWAVE TRANSFORMER)"/)
      CALL SECOND(T1)
      CALL MINI5W(FGR,N,M,X,DX,EPS,MAXF,W,IW)
      CALL SECOND(T2)
      CPU=T2-T1
      WRITE(1,200) CPU
  200 FORMAT("0CPU TIME:",F8.3," SECONDS")
      NFC=W(M+1)
      WRITE(1,300) EPS,NFC
  300 FORMAT("0EPS VAL=",E13.6/" F.EVAL.=",I5)
      WRITE(1,400) (X(I),I=1,N),(W(I),I=1,M)
  400 FORMAT("0SOLUTION:",6(/F20.8)/"0FUNCTION VALUES:",11(/F20.8)//)
      STOP
      END
C
```

```
C
      SUBROUTINE FGR(N,M,X,DF,F)
      DIMENSION X(N),DF(M,N),F(M)
      COMPLEX AI(4),V(4),A,B,C,RH,CRH,VG,TVG
      DIMENSION AL(3),TH(3),Z(3)
      COMMON NCOUNT,RG,RL,FREQ(11)
      DATA FACT1,FACT2/7.4948125,0.2095844728/
      NS=3
      NS1=NS+1
      DO 40 I=1,M
      BETA=FACT2*FREQ(I)
      ALFA=FACT1*BETA
      DO 10 J=1,NS
      JJ=J+J
      AL(J)=FACT1*X(JJ-1)
      Z(J)=X(JJ)
   10 CONTINUE
      AI(NS1)=CMPLX(1.0,0.0)
      V(NS1)=CMPLX(RL,0.0)
      DO 20 J=1,NS
      K=NS1-J
      K1=K+1
      T=BETA*AL(K)
      TH(K)=T
      CT=COS(T)
      ST=SIN(T)
      A=CMPLX(CT,0.0)
      B=CMPLX(0.0,Z(K)*ST)
      C=CMPLX(0.0,ST/Z(K))
      V(K)=A*V(K1)+B*AI(K1)
      AI(K)=C*V(K1)+A*AI(K1)
   20 CONTINUE
      VG=V(1)+AI(1)*RG
      TVG=(RG+RG)/VG
      RH=1.0-AI(1)*TVG
      CRH=TVG*CONJG(RH)
      FM=CABS(RH)
      F(I)=FM
      DO 30 J=1,NS
      T=TH(J)
      JJ=J+J
      J1=J+1
      J2=JJ-1
      DF(I,JJ)=REAL(CRH*(V(J)*AI(J)-V(J1)*AI(J1))/VG)/(Z(J)*FM)
      DF(I,J2)=ALFA*REAL(CRH*(V(J)*AI(J1)-V(J1)*AI(J))/VG)/(SIN(T)*FM)
   30 CONTINUE
   40 CONTINUE
      FM=0.0
      DO 50 I=1,M
      IF(F(I).GT.FM) FM=F(I)
   50 CONTINUE
      IF(NCOUNT.EQ.0) WRITE(1,200)
  200 FORMAT(//13X,"X1",12X,"X2",12X,"X3",12X,"X4",12X,"X5",12X,"X6",
     1 12X,"MAX(F)")
      NCOUNT=NCOUNT+1
      WRITE(1,100) NCOUNT,(X(I),I=1,N),FM
  100 FORMAT(1X,I5,6(F13.8,1X),1X,F13.8,2X,11A1)
      RETURN
      END
```

PROGRAM T2RMMU (3-SECTION MICROWAVE TRANSFORMER)

	X1	X2	X3	X4	X5	X6	MAX(F)
1	.80000000	1.50000000	1.20000000	3.00000000	.80000000	6.00000000	.38813233
2	.90000000	1.51604930	1.10000000	2.90000000	.90000000	6.10000000	.25859383
3	.92577041	1.63907010	.96590917	3.10000000	1.10000000	5.90000000	.27440370
4	1.00000000	1.60944075	1.00117844	3.00000000	1.00000000	6.00000000	.20190611
5	.85589970	1.55971115	1.00902065	3.20000000	1.08792370	5.80000000	.27854952
6	.90000000	1.58268512	1.00627271	3.10000000	1.06102097	5.90000000	.23975004
7	.95000000	1.59425478	1.00313508	3.05000000	1.03051281	5.95000000	.20909191
8	.97500000	1.60003961	1.00156626	3.02500000	1.01525873	5.97500000	.20154102
9	1.00430879	1.59249966	.99822726	3.07531822	1.00311132	5.95116460	.20260379
10	1.00088815	1.59399712	1.00008572	3.07621267	.99916669	5.96002514	.19802040
11	1.00016800	1.59682950	1.00001135	3.08463835	.99983466	5.97164943	.19789703
12	.99955105	1.61283336	.99996044	3.12193821	1.00041262	6.03551229	.19761384
13	.99974907	1.63349063	1.00000664	3.16262159	1.00025859	6.11381691	.19743702
14	.99992489	1.63463073	1.00000073	3.16279629	1.00007439	6.11701215	.19729193
15	.99999551	1.63471930	1.00000007	3.16233863	1.00000434	6.11734875	.19729074
16	.99999995	1.63470852	1.00000000	3.16228041	1.00000005	6.11730887	.19729063
17	1.00000000	1.63470720	1.00000000	3.16227774	1.00000000	6.11730393	.19729063
18	1.00000000	1.63470714	1.00000000	3.16227766	1.00000000	6.11730369	.19729063

CPU TIME: .704 SECONDS

EPS VAL= .100000E-06
F.EVAL.= 18

SOLUTION:
1.00000000
1.63470714
1.00000000
3.16227766
1.00000000
6.11730369

FUNCTION VALUES:
.19729063
.03946026
.17197713
.19729063
.12388802
.00000000
.12388802
.19729063
.17197713
.03946026
.19729063

$$\underset{\sim}{x}^0 = [0.8 \quad 1.5 \quad 1.2 \quad 3.0 \quad 0.8 \quad 6.0]^T .$$

VII. REFERENCES

[1] J. Hald and K. Madsen, "Combined LP and quasi-Newton methods for minimax optimization", Mathematical Programming, vol. 20, 1981, pp. 49-62.

[2] J.W. Bandler, T.V. Srinivasan and C. Charalambous, "Minimax optimization of networks by grazor search", IEEE Trans. Microwave Theory Tech., vol. MTT-20, 1972, pp. 596-604.

[3] J.W. Bandler and D. Sinha, "FLOPT5 - a program for minimax optimization using the accelerated least pth algorithm", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-218, 1978.

APPENDIX

LISTING OF THE MINI5W PACKAGE

```
      SUBROUTINE MINI5W (FDF,N,M,X,RDX,EPS,MAXFUN,W,IW)
C
C     MINIMAX OPTIMIZATION USING QUADRATIC PROGRAMMING.
C     KAJ MADSEN, NUMERISK INSTITUT,
C     THE TECHNICAL UNIVERSITY OF DENMARK, LYNGBY, DENMARK.
C     MARCH 1981.
C
      DIMENSION X(N), W(IW)
C
C        IW MUST BE AT LEAST  16N+4M+2MN+2N**2+MAX(M,3N**2+6N+5)+13
C
      EXTERNAL FDF
      N1=N+1
      N2=N+2
      IIND1=5*N1+M
      IWO=MAX0(N1*(N+5)+M,(2*N+3)**2+1)
      IRLAM=2*N1
      NF0=1
      NF1=NF0+M
      NDF0=NF1+M
      NDF1=NDF0+M*N
      NX1=NDF1+M*N
      NH=NX1+N
      NB=NH+N2
      NIND1=NB+N*N
      NIND0=NIND1+IIND1
      NWO=NIND0+M
      NY=NWO+IWO
      NBH=NY+N
      NRLAM=NBH+N
      CALL MMSUBB (FDF,N,M,X,RDX,EPS,MAXFUN,W(NF0),W(NF1),W(NDF0),W(NDF1
     1),W(NX1),W(NH),W(NB),W(NIND1),W(NIND0),IIND1,W(NWO),IWO,W(NY),W(NB
     2H),W(NRLAM),IRLAM,N1,N2)
      RETURN
      END
C
C
      SUBROUTINE MMSUBB (FDF,N,M,X0,RDX,EPS,MAXFUN,F0,F1,DF0,DF1,X1,H,B,
     1IND1,IND0,IIND1,WO,IWO,Y,BH,RLAM,IRLAM,N1,N2)
      EXTERNAL FDF
      DIMENSION X0(N), F0(M), F1(M), DF0(M,N), DF1(M,N), X1(N), H(N2), B
     1(N,N), IND1(IIND1), IND0(M), WO(IWO), Y(N), BH(N), RLAM(N1)
      LOGICAL X1OK,NEWTON
C
C     X0       IS THE CURRENT APPROXIMATION OF THE SOLUTION.
C     F0,DF0   ARE THE CORRESPONDING SETS OF FUNCTION VALUES AND DERIVA-
C              TIVES.
C     X1       IS THE CURRENT APPROXIMATION OF THE SOLUTION
C     F1,DF1   ARE THE CORRESPONDING SETS OF FUNCTION VALUES AND DERIVA-
C              TIVES.
C     H        IS THE TRIAL INCREMENT TO ADD TO X0.
C     IND0     HOLDS THE INDICES CORRESPONDING TO ACTIVE CONSTRAINTS AT
C              X0 - WHEN APPROPRIATE. (THERE IS KACT0 OF THESE).
C     IND1     AS IND0, BUT AT THE POINT X1.
C     RLAM     HOLDS AN APPROXIMATION TO THE LAGRANGE MULTIPLIERS -
C              WHEN APPROPRIATE.
C     B        IS THE APPROXIMATE HESSIAN, UPDATED BY POWELL'S METHOD.
C     Y,BH,WO  ARE WORK AREAS.
C     X1OK     IS TRUE IF X1 IS ACCEPTED AS A NEW ITERATE.
C     NEWTON   IS TRUE IF THE NEXT STEP OF THE ITERATION WILL BE A
C              NEWTON STEP.
C
C     EPSFL IS THE SMALLEST MACHINE NUMBER X FOR WHICH 1+X>1
C
      EPSFL=16.E0**(-12)
```

```
      IWO2=SQRT(IWO+0.E0)
      NEWTON=.FALSE.
      KACT0=0
      RDXFIX=RDX
      DEL=F0(1)
      KBOUND=F0(2)
      KBOUND=MAX0(KBOUND,2)
      NTAL=0
C
C       FIND THE LENGTH OF THE STARTING VECTOR
C
      X0MAX=0.E0
      DO 1 I=1,N
      X0MAX=AMAX1(X0MAX,ABS(X0(I)))
    1 CONTINUE
C
C       CALCULATE FUNCTION VALUES
C
      CALL FPOS (FDF,N,M,X0,DF0,F0)
      NTAL=NTAL+1
      F0MAX=0.E0
      DO 2 J=1,M
      IF (F0(J).GT.F0MAX) F0MAX=F0(J)
    2 CONTINUE
C
C       INITIALIZE ARRAYS
C
      DO 4 I=1,N
      DO 3 J=1,N
      B(I,J)=0.E0
    3 CONTINUE
      B(I,I)=1.E0
    4 CONTINUE
C
C       ITERATIVE LOOP STARTS HERE
C
    5 CONTINUE
C
C           FIND THE SOLUTION H OF THE LINEAR OR QUADRATIC SUBPROBLEM
C           F1PRED IS THE MINIMUM PREDICTED BY THE SUBPROBLEM
C
      IF (NEWTON) GO TO 7
    6 CALL MA19W (N,M,DF0,M,F0,RDX,0.E0,H,WO,IND1,IWO,IIND1)
      F1PRED=WO(M+1)
      GO TO 8
    7 CONTINUE
      CALL QUAD2 (N,M,DF0,F0,B,H,RLAM,N1,IND1,KACT1,WO,IWO2,EPSFL)
      F1PRED=H(N1)
    8 CONTINUE
      IF (F1PRED.GT.F0MAX) GO TO 38
C
C           FIND THE NORM OF H, AND FIND THE POINT X+H
C
      HMAX=0.E0
      DO 9 I=1,N
      HMAX=AMAX1(HMAX,ABS(H(I)))
      X1(I)=X0(I)+H(I)
    9 CONTINUE
C
C           IF THE STEP LENGTH IS TOO LARGE UNDER THE NEWTON ITERATION
C           THEN USE THE LP DIRECTION
C
      IF ((HMAX.LE.RDXFIX).OR.(.NOT.NEWTON)) GO TO 10
      NEWTON=.FALSE.
      GO TO 6
```

```
C
C     FIND THE NEW FUNCTION VALUES
C
   10 CALL FPOS (FDF,N,M,X1,DF1,F1)
      NTAL=NTAL+1
      F1MAX=0.E0
      DO 11 J=1,M
      F1MAX=AMAX1(F1MAX,F1(J))
   11 CONTINUE
C
C     TEST IF THE NEW POINT IS ACCEPTABLE
C
      X1OK=(F0MAX-F1MAX).GE.0.01*(F0MAX-F1PRED)
C
C     FIND THE SET OF ACTIVE FUNCTIONS
C
      KACT1=0
      DO 13 J=1,M
      SUM=F0(J)
      DO 12 I=1,N
      SUM=SUM+DF0(J,I)*H(I)
   12 CONTINUE
      IF (ABS(SUM-F1PRED).GT.DEL*ABS(F1PRED)) GO TO 13
      KACT1=KACT1+1
      IND1(KACT1)=J
   13 CONTINUE
C
C    FIND THE MULTIPLIERS RLAM
C
      IF (.NOT.NEWTON) CALL MULT (N,M,DF0,DF1,X1OK,IND1,N1,KACT1,RLAM,WO
     1,IWO2,EPSFL)
C
C       FIND Y: THE DIFFERENCE IN THE LAGRANGIAN GRADIENTS
C
      DO 14 I=1,N
      Y(I)=0.E0
   14 CONTINUE
      DO 16 J=1,KACT1
      JK=IND1(J)
      DO 15 I=1,N
      Y(I)=Y(I)+RLAM(J)*(DF1(JK,I)-DF0(JK,I))
   15 CONTINUE
   16 CONTINUE
C
C     ADJUST THE LOCAL BOUND RDX
C
      IF ((F0MAX-F1MAX).LE.0.25*(F0MAX-F1PRED)) RDX=RDX/2.
      IF ((F0MAX-F1MAX).GE.0.75*(F0MAX-F1PRED)) RDX=RDX*2.
C
C     TEST FOR NEWTON ITERATION
C
      IF (KACT1.GT.N1) KACT0=0
      IF (KACT1.EQ.KACT0) GO TO 18
      KACT0=KACT1
      DO 17 I=1,KACT1
      IND0(I)=IND1(I)
   17 CONTINUE
      KEQUAL=0
      GO TO 24
   18 CONTINUE
      KEQUAL=KEQUAL+1
      DO 19 I=1,KACT1
      IF ((IND1(I).EQ.IND0(I)).AND.(RLAM(I).GE.0E0)) GO TO 19
      IND0(I)=IND1(I)
      KEQUAL=0
```

```
   19 CONTINUE
      IF (KEQUAL.LT.(KBOUND-1)) GO TO 24
C
C        FIND THE RESIDUAL-NORM OF THE SET OF NON-LINEAR EQUATIONS
C
      RES=0.E0
      DO 21 I=1,N
      S=0.E0
      DO 20 J=1,KACT1
      JK=IND1(J)
      S=S+RLAM(J)*DF1(JK,I)
   20 CONTINUE
      RES=AMAX1(RES,ABS(S))
   21 CONTINUE
      DO 22 J=1,KACT1
      JK=IND1(J)
      RES=AMAX1(RES,F1MAX-F1(JK))
   22 CONTINUE
      IF (KEQUAL.GE.KBOUND) GO TO 23
      RES0=RES
      GO TO 24
   23 IF (RES.LE.0.999*RES0) GO TO 25
   24 NEWTON=.FALSE.
      GO TO 26
   25 NEWTON=.TRUE.
      RES0=RES
C
C        INTRODUCE THE NEW POINT IF IT IS ACCEPTABLE
C
   26 IF ((.NOT.X1OK).AND.(.NOT.NEWTON)) GO TO 30
      F0MAX=F1MAX
      X0MAX=0.E0
      DO 28 I=1,N
      X0(I)=X1(I)
      X0MAX=AMAX1(X0MAX,ABS(X0(I)))
      DO 27 J=1,M
      DF0(J,I)=DF1(J,I)
   27 CONTINUE
   28 CONTINUE
      DO 29 J=1,M
      F0(J)=F1(J)
   29 CONTINUE
   30 CONTINUE
C
C        ADJUST THE MATRIX B USING POWELL'S METHOD
C
C        FIND BH AND YH
C
      YH=0
      DO 32 J=1,N
      YH=YH+Y(J)*H(J)
      SUMB=0.E0
      DO 31 I=1,N
      SUMB=SUMB+B(J,I)*H(I)
   31 CONTINUE
      BH(J)=SUMB
   32 CONTINUE
C
C        FIND T AND SEE IF THETA IS LESS THAN 1
C
      T=0.E0
      DO 33 I=1,N
      T=T+H(I)*BH(I)
   33 CONTINUE
      IF (YH.GE.0.2E0*T) GO TO 35
```

```
      THETA=0.8E0*T/(T-YH)
C
C       IF THETA IS TOO SMALL WE DON'T ALTER MATRIX B
C
      IF (THETA.LT.0.50E0) GO TO 37
      S=1E0-THETA
      YH=0.E0
      DO 34 I=1,N
      Y(I)=THETA*Y(I)+S*BH(I)
      YH=YH+Y(I)*H(I)
   34 CONTINUE
   35 CONTINUE
C
C       FINALLY WE CAN CALCULATE THE NEW B
C
      DO 36 I=1,N
      S1=BH(I)/T
      S2=Y(I)/YH
      DO 36 J=I,N
      B(I,J)=B(I,J)-S1*BH(J)+S2*Y(J)
   36 B(J,I)=B(I,J)
C
   37 CONTINUE
C
C       TEST THE STOPPING CRITERION
C
      IF ((NTAL.LT.MAXFUN).AND.(HMAX.GT.X0MAX*EPS)) GO TO 5
C
      IF (HMAX.GT.X0MAX*EPS) NTAL=NTAL+1
      GO TO 39
   38 EPS=0.E0
   39 F1(1)=NTAL
      RETURN
      END
C
C
      SUBROUTINE QUAD2 (N,M,DF0,F0,B,H,RLAM,N1,IND,KACT1,W,IW2,EPSFL)
C
C                     MINIMIZE    DELTA+HBH/2
C                     SUBJECT TO F0+DF0^H = DELTA
C                     FOR THE INDICES IN IND.
C          IND   GIVES THE INDICES CORRESPONDING TO ACTIVE CONSTRAINTS.
C                THERE IS IND(N+2) OF THESE.
C          NI    IS  N+I.
C          IW    MUST BE AT LEAST  2*N+2
C
      DIMENSION F0(M), DF0(M,N), B(N,N), H(N1), RLAM(N1), IND(N1), W(IW2
     1,IW2)
      NS=N+KACT1
      NTOT=NS+1
      NTOT1=NTOT+1
      EPS1=EPSFL*10*NTOT
C
C     SET UP THE THE LINEAR SYSTEM
C     1: THE MATRIX (IT IS SYMMETRIC)
C
      DO 1 I=1,N
      W(I,NTOT)=0.E0
      W(NTOT,I)=0.E0
      DO 1 J=1,I
      W(I,J)=B(I,J)
      W(J,I)=W(I,J)
    1 CONTINUE
      DO 3 J=N1,NS
      W(J,NTOT)=-1.E0
```

```
      W(NTOT,J)=-1.E0
      DO 2 I=N1,J
      W(I,J)=0.E0
      W(J,I)=0.E0
    2 CONTINUE
      DO 3 I=1,N
      JA=IND(J-N)
      W(J,I)=DF0(JA,I)
      W(I,J)=W(J,I)
    3 CONTINUE
      W(NTOT,NTOT)=0.E0
C
C    2: RIGHT HAND SIDE
C
      DO 4 I=1,N
      W(I,NTOT1)=0.E0
    4 CONTINUE
      DO 5 J=N1,NS
      JA=IND(J-N)
      W(J,NTOT1)=-F0(JA)
    5 CONTINUE
      W(NTOT,NTOT1)=-1.E0
C
C     SOLVE THE LINEAR SYSTEM
C
      CALL GAUSS (W,IW2,NTOT,NTOT1,EPS1)
C
C     STORE THE SOLUTION IN H AND RLAM
C
      DO 6 I=1,N
      H(I)=W(I,NTOT1)
    6 CONTINUE
      H(N1)=W(NTOT,NTOT1)
      DO 7 J=1,KACT1
      RLAM(J)=W(N+J,NTOT1)
    7 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE MULT (N,M,DF0,DF1,X1OK,IND1,N1,KACT1,RLAM,W,IW2,EPSFL)
C
C     FIND THE MULTIPLIERS RLAM BY A LEAST SQUARES CALCULATION, SUB-
C     JECT TO THE CONSTRAINT THAT THE SUM OF THE MULTIPLIERS IS 1.
C
      DIMENSION DF0(M,N), DF1(M,N), IND1(N1), RLAM(KACT1), W(IW2,IW2)
      LOGICAL X1OK
      K1=KACT1+1
      K2=KACT1+2
      EPS1=EPSFL*10*K1
      DO 4 I=1,KACT1
      IK=IND1(I)
      DO 3 J=1,I
      JK=IND1(J)
      S=0.E0
      DO 2 L=1,N
      IF (X1OK) GO TO 1
      S=S+DF0(IK,L)*DF0(JK,L)
      GO TO 2
    1 S=S+DF1(IK,L)*DF1(JK,L)
    2 CONTINUE
      W(I,J)=S
      W(J,I)=S
    3 CONTINUE
      W(I,K1)=-1.E0
```

```
      W(K1,I)=1.E0
      W(I,K2)=0.E0
    4 CONTINUE
      W(K1,K1)=0.E0
      W(K1,K2)=1.E0
      CALL GAUSS (W,IW2,K1,K2,EPS1)
      DO 5 I=1,KACT1
      RLAM(I)=W(I,K2)
    5 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE FPOS (FDF,N,M,X,DF,F)
      DIMENSION X(N), DF(M,N), F(M)
      CALL FDF (N,M,X,DF,F)
      DO 2 J=1,M
      IF (F(J).GE.0E0) GO TO 2
      F(J)=-F(J)
      DO 1 I=1,N
      DF(J,I)=-DF(J,I)
    1 CONTINUE
    2 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE GAUSS (A,IA,N,M,EPS)
C
C       SOLUTION OF A SET OF N LINEAR EQUATIONS IN N UNKNOWNS WITH M-N
C       RIGHT HAND SIDES. THE SET OF SOLUTIONS WILL BE STORED IN PLACE
C       OF THE RIGHT HAND SIDES: IN THE LAST M-N COLUMNS OF MATRIX A.
C       KAJ MADSEN, NUMERISK INSTITUT, LYNGBY, AUGUST 1980.
C
      DIMENSION A(IA,M)
      N1=N+1
      IF (N.EQ.1) GO TO 10
C
C       EQUILIBRATION
C
      DO 3 I=1,N
      C=ABS(A(I,1))
      DO 1 J=2,N
      IF (ABS(A(I,J)).GT.C) C=ABS(A(I,J))
    1 CONTINUE
      DO 2 J=1,M
      A(I,J)=A(I,J)/C
    2 CONTINUE
    3 CONTINUE
C
C       PIVOTING AND REDUCTION TO TRIANGULAR FORM
C
      NM=N-1
      DO 9 K=1,NM
      K1=K+1
      IPIV=K
      C=ABS(A(K,K))
      DO 4 I=K1,N
      IF (C.GE.ABS(A(I,K))) GO TO 4
      IPIV=I
      C=ABS(A(I,K))
    4 CONTINUE
C
C       TEST FOR SINGULARITY
C
```

```
      IF (C.LT.EPS) GO TO 15
C
C         PIVOTING CONTINUED
C
      IF (IPIV.EQ.K) GO TO 6
      DO 5 J=K,M
      C=A(K,J)
      A(K,J)=A(IPIV,J)
      A(IPIV,J)=C
    5 CONTINUE
    6 CONTINUE
      DO 8 I=K1,N
      C=A(I,K)/A(K,K)
      DO 7 J=K1,M
      A(I,J)=A(I,J)-C*A(K,J)
    7 CONTINUE
    8 CONTINUE
    9 CONTINUE
C
C         END OF REDUCTION
C
   10 CONTINUE
C
C         TEST FOR SINGULARITY
C
      IF (ABS(A(N,N)).LT.EPS) GO TO 15
C
C         BACKSUBSTITUTION
C
      DO 14 II=1,N
      I=N-II+1
      DO 13 J=N1,M
      C=A(I,J)
      IF (I.EQ.N) GO TO 12
      I1=I+1
      DO 11 K=I1,N
      C=C-A(I,K)*A(K,J)
   11 CONTINUE
   12 A(I,J)=C/A(I,I)
   13 CONTINUE
   14 CONTINUE
      GO TO 16
   15 EPS=-1.E0
   16 RETURN
      END
C
C
      SUBROUTINE MA19W (N,M,A,IA,B,DX,EPS,X,RES,IREF,NURES,NUIREF)
      DIMENSION A(IA,N), B(M), X(N), RES(NURES)
      INTEGER IREF(NUIREF)
C
C        NURES=(N+1)*(N+5)+M
C       NUIREF=5*(N+1)+M
C
      N1=N+1
      N2=N+2
      NURHO=MAX0(M,3*N1)+1
      NH=1+NURHO
      CALL MA19BW (N,M,A,IA,B,DX,EPS,X,RES(NH),N1,RES(1),IREF,NURHO,NUIR
     1EF,N2)
      RETURN
      END
C
C
      SUBROUTINE MA19BW (N,M,A,IA,B,DX,EPSH,X,H,N1,RHO,IREF,NURHO,NUIREF
```

```
     1,N2)
      DIMENSION A(IA,N), B(M), X(N), H(N1,N2), RHO(NURHO)
      REAL LAM
      INTEGER IREF(NUIREF)
      LOGICAL GAMCH
      IF ((DX.LT.0E0).OR.(EPSH.LT.0E0)) RETURN
      IF ((N.LT.1).OR.(M.LT.1)) RETURN
C
      NN2=N+N2
      NN3=NN2+N1
      LREF=NN3+N1
      LBND=LREF+M
      M1=M+1
      SI4N=1./(4.*N)
      NTAL=0
C
C       FIND EQUATION I0  WHICH GOES INTO THE FIRST REFERENCE
C
      C=-1.
      DO 1 J=1,M
      IREF(LREF+J)=0
      IF (ABS(B(J)).LT.C) GO TO 1
      C=ABS(B(J))
      I0=J
    1 CONTINUE
C
C       INITIALIZE REFERENCE ARRAYS
C
      S=0.E0
      T=B(I0)
      XM=M
      DO 2 I=1,N
      D=A(I0,I)
      S=S+ABS(D)
      XM=XM+1E0
      IF (D.EQ.0E0) D=-1.
      IREF(I)=SIGN(XM,-D*T)
      IREF(LBND+I)=IREF(I)
    2 CONTINUE
      XM=I0
      IREF(N1)=SIGN(XM,T)
      IREF(LREF+I0)=IREF(N1)
C
C       INITIALIZE DH,DG, AND GAM
C
      IF ((DX*S).GT.C) GO TO 3
      GAM=DX
      DG=GAM
      DH=C-DX*S
      GO TO 4
    3 DG=C/S
      DH=0.
      GAM=DG
C
C       FIND VECTOR X
C
    4 DO 5 I=1,N
      XM=IREF(I)
      X(I)=SIGN(DG,XM)
    5 CONTINUE
C
C       FIND MATRIX H
C
      S=1./(S+1.)
      H(N1,N1)=S
```

```
      IREF(NN2+N1)=0
      DO 8 I=1,N
      XM=-IREF(I)
      H(N1,I)=SIGN(S,XM)
      T=ABS(A(I0,I))*S
      H(I,N1)=T
      DO 6 J=1,N
      XM=-IREF(J)
      H(I,J)=SIGN(T,XM)
    6 CONTINUE
      IF (T.GT.0.) GO TO 7
      IREF(NN2+I)=1
      H(I,N2)=1.
      GO TO 8
    7 IREF(NN2+I)=0
    8 H(I,I)=ISIGN(1,IREF(I))+H(I,I)
C
C        INITIALIZE SOME CONSTANTS
C
      RSIG=1E0-S
      DGH=DG
      DH1=DH
      ETA=GAM
      ERRX=0.E0
      GAMCH=.TRUE.
      NBIN=N
      GO TO 65
C
C        ITERATIVE LOOP STARTS HERE
C        FIND VECTOR RHO
C
    9 DGH=DG
      IF (IOS.LT.0) GO TO 12
      DO 11 I=1,N1
      S=-H(I,N1)
      DO 10 J=1,N
      S=S-H(I,J)*A(I0,J)
   10 CONTINUE
      RHO(I)=S
   11 CONTINUE
      GO TO 19
   12 DO 14 I=1,N1
      S=-H(I,N1)
      DO 13 J=1,N
      S=S+H(I,J)*A(I0,J)
   13 CONTINUE
      RHO(I)=S
   14 CONTINUE
      GO TO 19
C
C        BOUNDS VIOLATED
C
   15 I0=M+J0
      DHH=DH
      IOS=SIGN(1.E0,X(J0))
      IF (IOS.LT.0) GO TO 17
      DO 16 I=1,N1
      RHO(I)=-H(I,J0)-H(I,N1)
   16 CONTINUE
      GO TO 19
   17 DO 18 I=1,N1
      RHO(I)=H(I,J0)-H(I,N1)
   18 CONTINUE
C
C        FIND EQUATION L0 WHICH LEAVES THE REFERENCE
```

```
C       FIND -H(I,N1)/RHO(I)  FOR NEGATIVE VALUES OF RHO(I)
C
   19 LB=0
      RTAU=0.E0
      IF (I0.GT.M) RTAU=1.
      DO 21  I=1,N1
      IF (RHO(I).GE.0.E0) GO TO 20
      LB=LB+1
      IREF(N1+LB)=I
      KK=N1
      IF (IREF(NN2+I).GT.0) KK=N2
      RHO(N1+I)=-H(I,KK)/RHO(I)
   20 IF (IABS(IREF(I)).LE.M) GO TO 21
      RTAU=RTAU+RHO(I)
   21 CONTINUE
C
C       FIND THE COEFFICIENTS IN THE RATIONAL EXPRESSION
C       (TT+LAM*SS)/(RSIG+LAM*RTAU)
C
      DG2=DG*2.
      IF (DH.GT.0E0) GO TO 22
      TT=DG*RSIG
      SS=DG*RTAU+DHH+DGH-DG
      NL=1
      GO TO 23
   22 RSIG=1.-RSIG
      RTAU=-RTAU
      TT=DH*RSIG
      SS=DH*RTAU+DHH-DH+DGH-DG
      NL=2
   23 SMAX=0.
      L=1
   24 LA=L
C
C       FIND MINIMUM VALUE OF  -H(I,N1)/RHO(I)
C
      L0=IREF(N1+LA)
      LAM=RHO(N1+L0)
      ILAM=IREF(NN2+L0)
      IF (LA.EQ.LB) GO TO 26
      L=LA+1
      LM=LA
      DO 25  I=L,LB
      IN=IREF(N1+I)
      S=RHO(N1+IN)
      IS=IREF(NN2+IN)
      IF (((IS.EQ.ILAM).AND.(S.GE.LAM)).OR.(IS.LT.ILAM)) GO TO 25
      L0=IN
      LAM=S
      ILAM=IS
      LM=I
   25 CONTINUE
C
C       REORDER
C
      IREF(N1+LM)=IREF(N1+LA)
      IREF(N1+LA)=L0
C
C       FIND  MAX(-RHO(I))
C
   26 SMAX=AMAX1(SMAX,-RHO(L0))
C
C       REVISE THE COEFFICIENTS OF THE RATIONAL EXPRESSION
C
      ML=NL
```

```
      IF (IABS(IREF(L0)).GT.M) ML=NL+2
      GO TO (29,28,28,27), ML
   27 TT=TT+DG2*H(L0,N1)
      SS=SS+DG2*RHO(L0)
      GO TO 29
   28 RSIG=RSIG-2.*H(L0,N1)
      RTAU=RTAU-2.*RHO(L0)
   29 IF ((RSIG*SS.GT.RTAU*TT).AND.(LA.LT.LB)) GO TO 24
C
C        TEST IF  -RHO(L0)  IS TOO SMALL
C
      SMAX=SMAX/4.
   30 IF (-RHO(L0).GE.SMAX) GO TO 31
      LA=LA-1
      L0=IREF(N1+LA)
      GO TO 30
C
C        UPDATE REFERENCE ARRAYS
C
   31 IREF(LREF+I0)=ISIGN(1,I0S)
      IREF(LREF+IABS(IREF(L0)))=0
      IF (IABS(IREF(L0)).GT.M) NBIN=NBIN-1
      IF (I0.GT.M) NBIN=NBIN+1
      IREF(L0)=ISIGN(I0,I0S)
C
C        UPDATE MATRIX H
C
      RHOL0=RHO(L0)
      DO 32 J=1,N1
      H(L0,J)=-H(L0,J)/RHOL0
   32 CONTINUE
      DO 34 I=1,N1
      IF (I.EQ.L0) GO TO 34
      S=RHO(I)
      DO 33 J=1,N1
      H(I,J)=S*H(L0,J)+H(I,J)
   33 CONTINUE
   34 CONTINUE
C
C        IF ANY SIGNS HAVE BEEN CHANGED, UPDATE H
C
      RHON1=1E0
      DO 35 I=1,N1
      RHO(N1+I)=1.
   35 CONTINUE
      IF (LA.LE.1) GO TO 46
      K=LA-1
      DO 36 I=1,N1
      RHO(NN2+I)=0.E0
   36 CONTINUE
      DO 37 I=1,K
      RHO(IREF(N1+I)+N1)=-1.
   37 CONTINUE
      DO 41 I=1,N1
      IF (RHO(N1+I).GT.0.) GO TO 39
      DO 38 J=1,N1
      RHO(NN2+J)=RHO(NN2+J)-H(I,J)
   38 CONTINUE
      GO TO 41
   39 DO 40 J=1,N1
      RHO(NN2+J)=RHO(NN2+J)+H(I,J)
   40 CONTINUE
   41 CONTINUE
      RHON1=RHO(NN2+N1)
      DO 42 J=1,N1
```

```
      H(J,N1)=H(J,N1)/RHON1
   42 CONTINUE
      DO 44 I=1,N
      S=RHO(NN2+I)
      DO 43 J=1,N1
      H(J,I)=-S*H(J,N1)+H(J,I)
   43 CONTINUE
   44 CONTINUE
C
C        CHANGE SIGNS IN SOME ROWS OF MATRIX H
C
      DO 45 L=1,K
      I=IREF(N1+L)
      J=-IREF(I)
      IREF(I)=J
      IREF(LREF+IABS(J))=J
      DO 45 J=1,N1
   45 H(I,J)=-H(I,J)
C
C        UPDATE THE LAST COLUMN OF H  IN CASE OF DEGENERACIES
C
   46 IF (IREF(NN2+L0).EQ.0) GO TO 52
      H(L0,N2)=-H(L0,N2)/(RHO(L0)*RHON1)
      DO 51 I=1,N1
      IF ((IREF(NN2+I).EQ.0).OR.(I.EQ.L0)) GO TO 51
      IF (IREF(NN2+I)-IREF(NN2+L0)) 47,49,48
   47 H(I,N2)=H(I,N2)/RHON1
      GO TO 51
   48 H(I,N2)=ABS(H(L0,N2)*RHO(I))
      IREF(NN2+I)=IREF(NN2+L0)
      GO TO 51
   49 C=H(I,N2)/RHON1+H(L0,N2)*RHO(I)
      IF ((LA.GT.1).AND.(RHO(N1+I).LT.0E0)) C=-C
      IF (C.GT.0) GO TO 50
      H(I,N2)=1.E0
      IREF(NN2+I)=IREF(NN2+I)+1
      GO TO 51
   50 H(I,N2)=C
   51 CONTINUE
      GO TO 54
C
C        TEST FOR DEGENERACIES
C
   52 DO 53 I=1,N1
      IREF(NN2+I)=0
      IF (H(I,N1).GT.0E0) GO TO 53
      IREF(NN2+I)=1
      H(I,N1)=0.E0
      H(I,N2)=1E0
   53 CONTINUE
C
C        UPDATE GAM
C
   54 GAMCH=.FALSE.
      IF ((NBIN.EQ.0).OR.((GAMM.LT.2.*GAM).AND.(GAMM.LT.DX))) GO TO 55
      IF (GAM.LT.GAMM) GAMCH=.TRUE.
      GAM=GAMM
C
C        UPDATE DH AND DG
C
   55 S=0.
      RSIG=0.E0
      DO 57 I=1,N1
      K=IABS(IREF(I))
      IF (K.GT.M) GO TO 56
```

```
      RHO(I)=B(K)*ISIGN(1,IREF(I))
      S=S+H(I,N1)*RHO(I)
      GO TO 57
   56 RSIG=RSIG+H(I,N1)
   57 CONTINUE
      IF (RSIG.NE.0E0) GO TO 58
      DH=ABS(S)
      DG=GAM
      GO TO 59
   58 DG=AMIN1(GAM,ABS(S)/RSIG)
      DH=0.
      IF (DG.EQ.GAM) DH=ABS(S-DG*RSIG)/(1E0-RSIG)
C
C        CALCULATE PARAMETER VALUES
C
   59 DGH=0.E0
      ERRX=0.E0
      DO 61 I=1,N1
      IF (IABS(IREF(I)).GT.M) GO TO 60
      RHO(I)=DH-RHO(I)
      GO TO 61
   60 RHO(I)=DG
   61 CONTINUE
      DO 64 I=1,N
      S=H(1,I)*RHO(1)
      DO 62 J=2,N1
      S=S+H(J,I)*RHO(J)
   62 CONTINUE
      IF (IREF(LBND+I).EQ.0) GO TO 63
      T=S
      XM=IREF(LBND+I)
      S=SIGN(DG,XM)
      ERRX=AMAX1(ERRX,ABS(S-T))
   63 X(I)=S
      IF (ABS(S).LE.DGH) GO TO 64
      DGH=ABS(S)
      J0=I
   64 CONTINUE
      NTAL=NTAL+1
C
C        CALCULATE GAMM
C
   65 GAMM=AMIN1(AMAX1(5.*DG,GAM),DX)
C
C        FIND EQUATION I0 WHICH GOES INTO THE REFERENCE
C
      DHH=0.E0
      T=DH
      T1=DH
      DO 68 I=1,M
      S=B(I)
      DO 66 J=1,N
      S=S+A(I,J)*X(J)
   66 CONTINUE
      RHO(I)=S
      IF (IREF(LREF+I).NE.0) GO TO 67
      IF (ABS(S).LE.DHH) GO TO 68
      DHH=ABS(S)
      I0S=SIGN(1.E0,S)
      I0=I
      GO TO 68
   67 T=AMAX1(T,ABS(S))
      T1=AMIN1(T1,ABS(S))
   68 CONTINUE
C
```

```
C         CALCULATE DH1 AND ETA
C
         ETA=AMAX1(GAMM,SIGN(ETA+ERRX,DH1-T))
         IF (.NOT.GAMCH) T=AMAX1(T,DH1+(T-T1))
         DH1=AMAX1(T,DH+EPSH)
C
C         TEST IF CONSTRAINTS ARE VIOLATED
C
         IF (DGH.GT.ETA) GO TO 15
C
C         TEST OF CONVERGENCE CRITERION
C
         IF (DHH.GT.DH1) GO TO 9
         IF ((GAM.LT.DX).AND.(DHH+DH.GT.(DH1-DH)*2.).AND.(NBIN.GT.0)) GO TO
        1 54
         RHO(M1)=DH
         IREF(N2)=NBIN
         IREF(N+3)=NTAL
         RETURN
         END
```

SOC-280

MINI5W - A FORTRAN PACKAGE FOR MINIMAX OPTIMIZATION

K. Madsen (Adapted and Edited by J.W. Bandler and W.M. Zuberek)

December 1981, No. of Pages: 31

Revised:

Key Words: Minimax optimization, nonlinear optimization, computer-aided design, optimization program

Abstract: MINI5W is a package of subroutines for solving unconstrained, nonlinear minimax optimization problems. First derivatives of all functions w.r.t. all variables are assumed to be known. The solution is found by an iteration that uses either linear programming applied in connection with first derivatives or a Newton step applied in connection with first derivatives and approximate second derivatives. The method has been described by Hald and Madsen. The Fortran IV package and documentation have been adapted for the CDC 170/730 system.

Description: Contains Fortran listing, user's manual. Source deck or magnetic tape available for $100.00. The listing contains 930 lines, of which 261 are comments.

Related Work: SOC-218, SOC-281.

Price: $ 30.00.