INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-291

MMUM - A FORTRAN PACKAGE FOR UNCONSTRAINED MINIMAX OPTIMIZATION

J.W. Bandler and W.M. Zuberek

May 1982

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

MMUM - A FORTRAN PACKAGE FOR UNCONSTRAINED MINIMAX OPTIMIZATION

J.W. Bandler and W.M. Zuberek

Abstract

MMUM is a package of subroutines for solving unconstrained minimax optimization problems. It is an extension and modification of the MINI5W package due to Madsen. First derivatives of all functions with respect to all variables are assumed to be known. The solution is found by an iteration that uses either linear programming applied in connection with first-order derivatives or a quasi-Newton method applied in connection with first-order and approximate second-order derivatives. The method has been described by Hald and Madsen. The package and documentation are developed for the CDC 170/730 system with the NOS 1.4 operating system and the Fortran 4.8508 compiler.

This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant G0647.

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.

used [4], and at each point the nonlinear residual functions are approximated by linear functions using the first derivative information. However, if a smooth valley through the solution is detected, a switch to Stage 2 is made and the quasi-Newton iteration is used. If it turns out that the Stage 2 iteration is unsuccessful (for instance, if the set of active functions has been wrongly choosen) then a switch is made back to Stage 1. The algorithm may switch several times between Stage 1 and Stage 2 but normally only a few switches will take place and the iteration will terminate either in Stage 1 with quadratic rate of convergence or in Stage 2 with superlinear rate of convergence [3].

Stage 1

The Stage 1 algorithm is described in detail in [4]. At the kth iteration the change $\underset{\sim}{h}^k$ of the approximation $\underset{\sim}{x}^{k-1}$ is determined as the solution of the linear minimax problem

$$\underset{\underset{\sim}{h}^k}{\text{Minimize}}\ \tilde{F}(\underset{\sim}{x}^{k-1}, \underset{\sim}{h}^k) = \max_{1 \leq i \leq m} |\bar{f}_i(\underset{\sim}{x}^{k-1}) + \bar{\underset{\sim}{f}}_i'^{T}(\underset{\sim}{x}^{k-1})\, \underset{\sim}{h}^k|$$

subject to constraint

$$\|\underset{\sim}{h}^k\| \leq \delta_x^{k-1}$$

where δ_x^k is equal to $0.5\delta_x^{k-1}$, δ_x^{k-1} or $2\delta_x^{k-1}$ according to an unsuccessful, not unsuccessful or successful (k-1)th iteration. The jth iteration is unsuccessful if

$$F(\underset{\sim}{x}^{j-1}) - F(\underset{\sim}{x}^{j-1} + \underset{\sim}{h}^j) \leq 0.25\ (F(\underset{\sim}{x}^{j-1}) - \tilde{F}(\underset{\sim}{x}^{j-1}, \underset{\sim}{h}^j))$$

it is successful if

$$F(\underset{\sim}{x}^{j-1}) - F(\underset{\sim}{x}^{j-1} + \underset{\sim}{h}^j) \geq 0.75\ (F(\underset{\sim}{x}^{j-1}) - \tilde{F}(\underset{\sim}{x}^{j-1}, \underset{\sim}{h}^j))$$

and is not unsuccessful otherwise. In each iteration of Stage 1, the step size is thus updated according to the goodness of the linear approximation. If the change of the objective function F slightly differs from the change predicted by linear approximation, the step size is increased; if it differs significantly, the step size is decreased. The initial step size δ_x^0 is defined by the user (argument DX). If the objective function F decreases, then $\underset{\sim}{x}^k = \underset{\sim}{x}^{k-1}+\underset{\sim}{h}^k$, otherwise $\underset{\sim}{x}^k=\underset{\sim}{x}^{k-1}$. There is no line search.

Switch to Stage 2

For the kth Stage 1 iteration the set A^k of active residual functions is defined as

$$A^k = \{ i \mid | \tilde{F}(\underset{\sim}{x}^{k-1}, \underset{\sim}{h}^k) - (\bar{f}_i(\underset{\sim}{x}^{k-1}) + \underset{\sim}{\bar{f}}_i'^T(\underset{\sim}{x}^{k-1}) \underset{\sim}{h}^k)| \leq \delta \tilde{F}(\underset{\sim}{x}^{k-1}, \underset{\sim}{h}^k) \} ,$$

where δ is a small positive number (normally $\delta = 0.01$ is an appropriate value). A switch to Stage 2 is made if the following conditions are simultaneously satisfied:

(1) The sets of active functions for the last t Stage 1 iterations are identical

$$A^{k-t+1} = A^{k-t+2} = \dots = A^k$$

(normally t=3 is an appropriate value; t cannot be less than 2).

(2) There exist nonnegative multipliers $\lambda_j^k \geq 0$, $j\epsilon A^k$, such that

$$\sum_{j\epsilon A^k} \lambda_j^k = 1$$

and, for $k > 2$,

$$r(\underset{\sim}{x}^{k-1} + \underset{\sim}{h}^k, \underset{\sim}{\lambda}^k, A^k) \leq 0.999\, r(\underset{\sim}{x}^{k-2} + \underset{\sim}{h}^{k-1}, \underset{\sim}{\lambda}^{k-1}, A^{k-1}) ,$$

where the residual $r(\underset{\sim}{x}, \underset{\sim}{\lambda}, A)$ is defined as

$$r(\underset{\sim}{x}, \underset{\sim}{\lambda}, A) = \max \left(\max_{1 \le i \le n} \left| \sum_{j\varepsilon A} \lambda_j \bar{f}'_{ji}(\underset{\sim}{x}) \right|, \ \max_{j\varepsilon A} (F(\underset{\sim}{x}) - \bar{f}_j(\underset{\sim}{x})) \right)$$

and

$$\bar{f}'_{ji} = \begin{cases} \partial f_j/\partial x_i, \text{ if } f_j \geqslant 0 , \\ -\partial f_j/\partial x_i, \text{ otherwise.} \end{cases}$$

Stage 2

At the kth Stage 2 iteration, an approximate Newton method is applied to the following system of equations

$$\sum_{j\varepsilon A^k} \lambda_j^k \bar{f}'_{ji} (\underset{\sim}{x}^k) = 0, \quad i = 1, \ldots, n$$

$$\sum_{j\varepsilon A^k} \lambda_j^k = 1$$

$$\bar{f}_{j_0}(\underset{\sim}{x}^k) - \bar{f}_j(\underset{\sim}{x}^k) = 0 , \ j\varepsilon A^k, \ j_0\varepsilon A^k, \ j \neq j_0,$$

where the unknowns are $[\underset{\sim}{x}^k, \underset{\sim}{\lambda}^k]$. The iteration is approximate because instead of $\bar{f}''_j(\underset{\sim}{x}^k)$ the approximated second-order derivatives are used.

Switch to Stage 1

At each kth Stage 2 iteration, the following conditions are checked:

(1) whether the set of active residual functions is preserved

$$A^k = A^{k-1} ,$$

(2) whether all multipliers λ_j^k are nonnegative

$$\lambda_j^k \geq 0, \ j\varepsilon A^k ,$$

(3) whether residuals $r(\underset{\sim}{x},\underset{\sim}{\lambda},A)$ are decreasing

$$r(\underset{\sim}{x}^{k-1} + \underset{\sim}{h}^{k}, \underset{\sim}{\lambda}^{k}, A^{k}) \leq 0.999\, r(\underset{\sim}{x}^{k-2} + \underset{\sim}{h}^{k-1}, \underset{\sim}{\lambda}^{k-1}, A^{k-1}),$$

(4) whether the step length $\underset{\sim}{h}^{k} = \underset{\sim}{x}^{k} - \underset{\sim}{x}^{k-1}$ is not greater than the value δ_x^0 (defined by the user)

$$\|\underset{\sim}{h}^{k}\| \leq \delta_x^0.$$

The Stage 2 iteration is continued when all the conditions are satisfied, otherwise the algorithm returns to Stage 1.

Termination

The iterative procedure terminates when any one of the following conditions is satisfied:

(1) the number of residual function evaluations exceeds the limit defined by the user (argument MAXF),

(2) the consecutive change $\underset{\sim}{h}^{k}$ of the approximation $\underset{\sim}{x}^{k}$ of the solution is sufficiently small

$$\|\underset{\sim}{h}^{k}\| \leq \varepsilon\, \|\underset{\sim}{x}^{k}\|,$$

where ε is defined by the user (argument EPS),

(3) the consecutive change $\underset{\sim}{h}^{k}$ reaches the machine accuracy

$$\|\underset{\sim}{h}^{k}\| \leq \varepsilon_0\, \|\underset{\sim}{x}^{k}\|,$$

where ε_0 is the smallest positive number such that

$$1 + \varepsilon_0 > 1,$$

(4) the consecutive change $\underset{\sim}{h}^{k}$ is insignificantly small

$$\|\underset{\sim}{h}^{k}\| \leq 10^{-50}$$

(when the solution $\underset{\sim}{x}^{*}$ is equal to $\underset{\sim}{0}$, the conditions (2) and (3) may be insufficient to terminate the iteration),

(5) the consecutive solution found by the package does not decrease the value of the objective function F.

Moreover, the user can terminate the iterative procedure and cause the return from the package by setting one of parameters during evaluation of residual functions (see argument FDF).

III. STRUCTURE OF THE PACKAGE

There are 3 different entries to the package and 3 corresponding "main" (or interfacing) subroutines:

1. subroutine MMUM1A - standard entry which provides uniform printing of input parameters as well as intermediate and final results,
2. subroutine MMUM2A - basic entry which does not provide any form of printed output (it is the user's responsibility to organize printing of data and results in this case),
3. subroutine MINI5W - original entry, as defined by Madsen [1]; this entry is preserved to ensure the compatibility with the previous version of the package.

Block diagrams of the package, corresponding to entries 1, 2 and 3 are shown in Fig. 1, 2 and 3, respectively. It can be observed that the PRINTOUT package of subroutines is used only when entry 1 (subroutine MMUM1A) is called, and that the subroutine MMX00Q (Fig. 1), which is responsible for printing the values of functions and their first derivatives, is replaced by dummy subroutine MMX00Z (Fig. 2,3) when entry 2 or 3 is used.

The common part of the package is composed of subroutines MMUN5W,

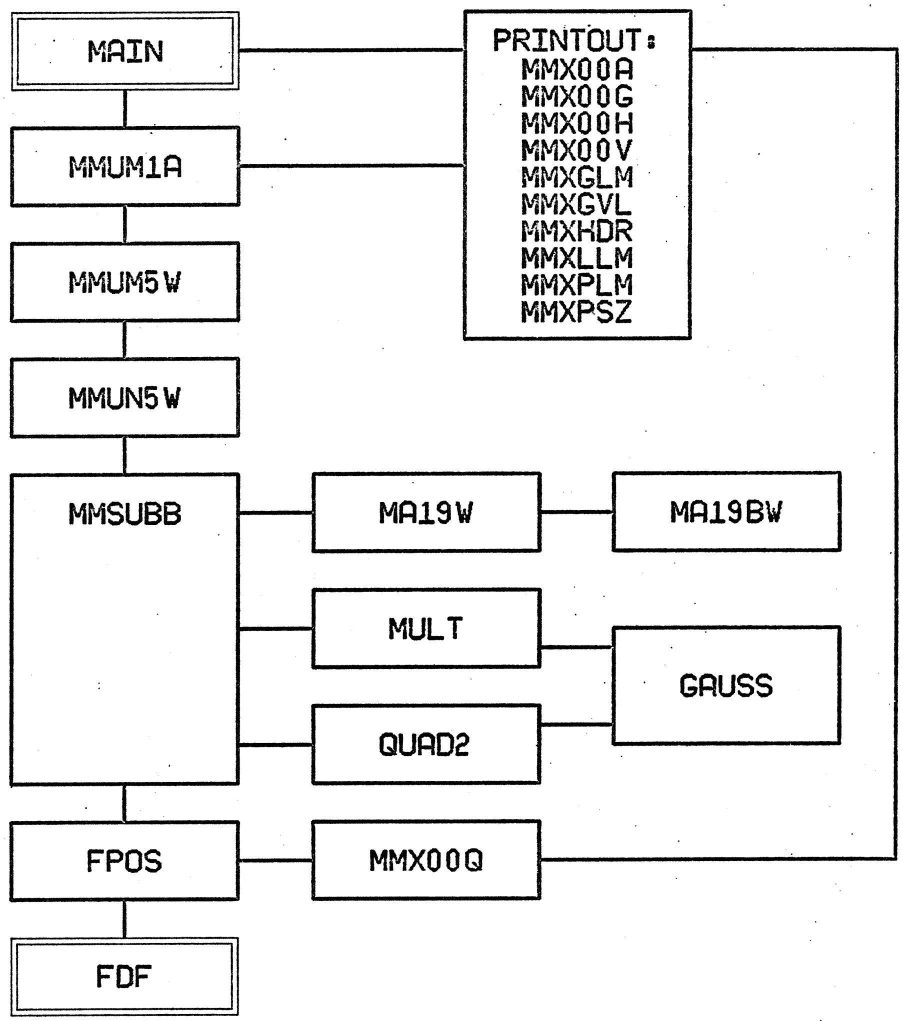

Fig. 1 Structure of the MMUM package corresponding to the standard entry (subroutine MMUM1A).

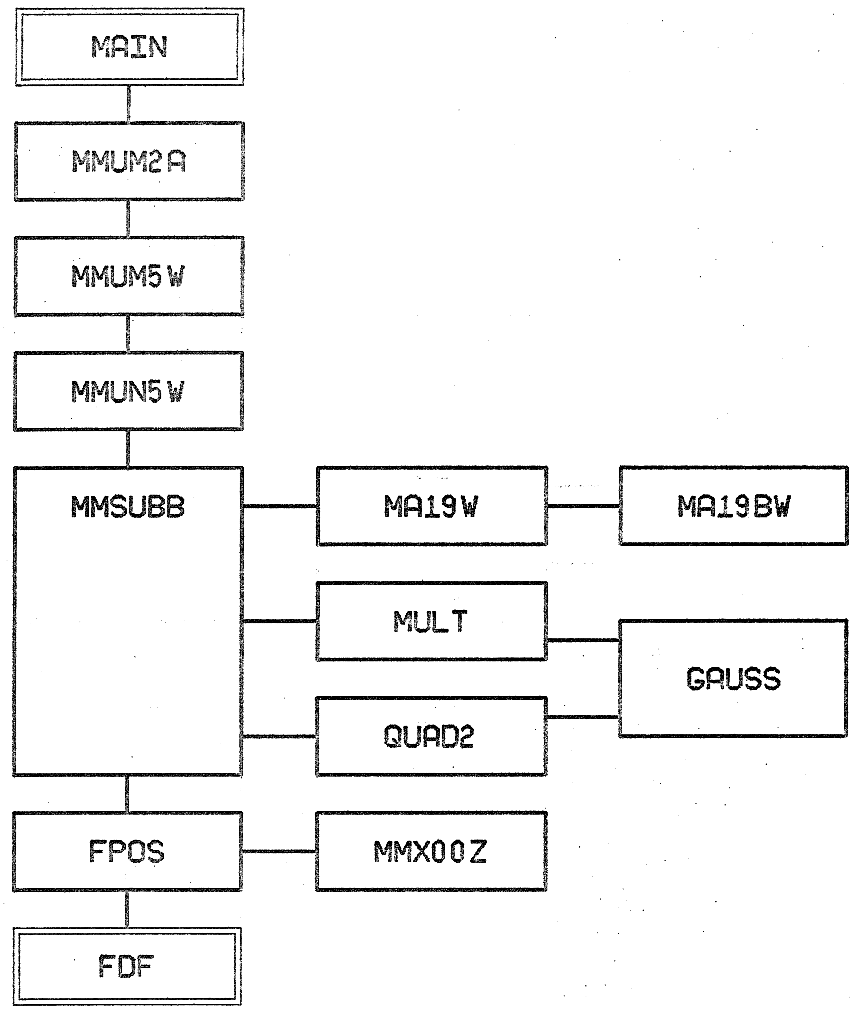

Fig. 2 Structure of the MMUM package corresponding to the basic entry (subroutine MMUM2A).

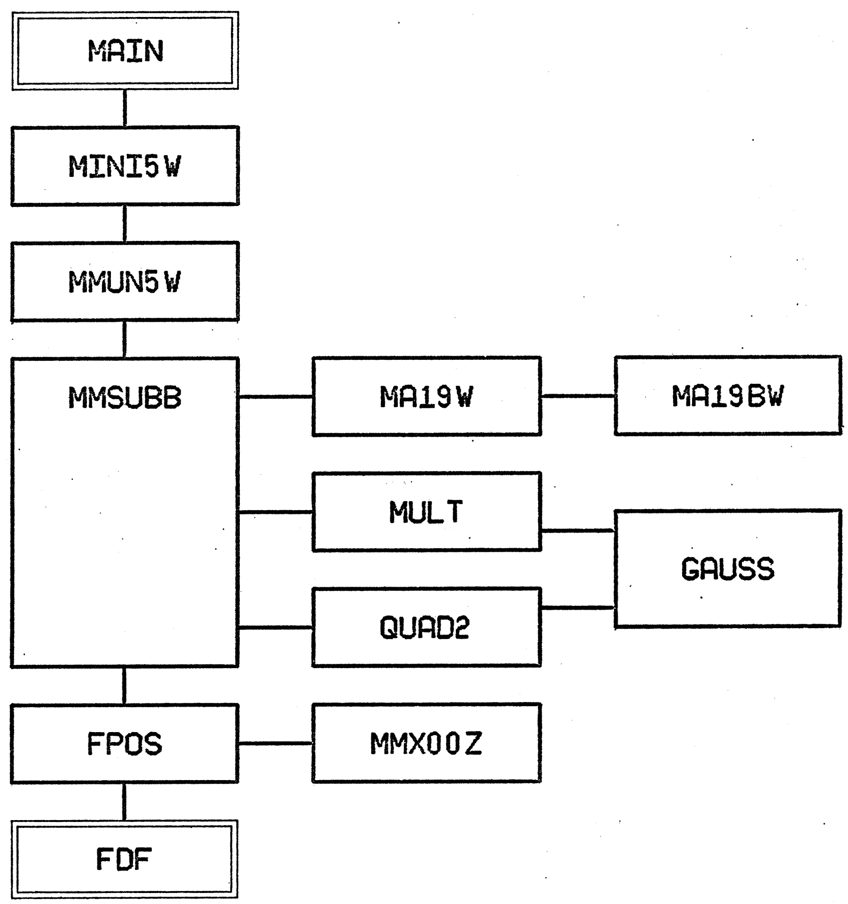

Fig. 3 Structure of the MMUM package corresponding to the original entry (subroutine MINI5W).

MMSUBB, MA19W, MA19BW, MULT, QUAD2, GAUSS and FPOS. MMUN5W subdivides the workspace (defined by the user) into a set of vectors and matrices used by the remaining subroutines. MMSUBB controls the minimax optimization, calls FPOS (and FDF) for evaluation of functions and their first derivatives, checks the conditions for switching from Stage 1 to Stage 2 and from Stage 2 to Stage 1, updates approximation of the Hessian matrix, and tests accuracy of the iterations. Stage 1 iterations are solved by MA19BW, and MA19W only simplifies the calling sequence of MA19BW. QUAD2 is called to solve Stage 2 iterations. MULT determines the multipliers during the Stage 1 iterations, and GAUSS is used to solve systems of linear equations.

The main segment MAIN and the subroutine FDF for evaluation of residual functions and their first-order derivatives must be supplied by the user.

When the standard entry (Fig. 1) is used, the subroutine MMUM1A and the set of subroutines PRINTOUT provide printed output containing principal input parameters of the minimax problem to be solved, and the solution obtained by the package. Moreover, the subroutine MMXOOQ outputs the values of residual functions and their derivatives according to the argument IPR in the call of MMUM1A.

For the standard entry (Fig. 1) and the basic entry (Fig. 2) the subroutine MMUM5W checks the formal correctness of input parameters and sets the output parameters to the values corresponding to the solution found by the package.

IV. LIST OF ARGUMENTS

Standard entry (subroutine MMUM1A)

The subroutine call is

```
CALL MMUM1A (FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
```

The arguments are as follows.

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

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

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

F(I) = $f_I(\underset{\sim}{x})$, I=1,...,M,

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

Note: The name FDF can be arbitrary (user's choice) and must appear in the EXTERNAL statement in the segment calling MMUM1A.

The user can terminate the iterative procedure and force the return from the package by setting to zero (in the subroutine FDF) the variable MARK in the common area MMU000

```
COMMON /MMU000/ MARK
```

(on entry to the package MARK is set to 1).

N is an INTEGER argument which 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 an INTEGER argument which must be set to m, the number of residual 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 the length at least N which on entry must be set to the initial approximation of the solution, $X(I)=x_I^0$, I=1,...,N. On exit X contains the best solution found by the package.

DX is a REAL variable which controls the step length of the iterative algorithm. On entry it must be set to such an initial value that in the region $\{\underset{\sim}{x} \mid \|\underset{\sim}{x}-\underset{\sim}{x}^0\| < DX\}$ the residual functions $f_i(\underset{\sim}{x})$ can be approximated reasonably well by linear functions. If the residual functions are nearly linear, DX should be set to an approximate value of the distance between the initial approximation $\underset{\sim}{x}^0$ and the solution, but if more curvature is present this value may be too large. Normally $DX=0.1*\|\underset{\sim}{x}^0\|$ is an appropriate value, but an improper choice of DX is usually not critical, since the value of DX is adjusted by the package during the iteration. The value of DX must be positive. On exit DX contains the last value of the step size δ_x^k.

EPS is a REAL variable which on entry must be set to the required accuracy of the solution. The iteration terminates when $\|\underset{\sim}{h}^k\| \leq EPS*\|\underset{\sim}{x}^k\|$, where $\underset{\sim}{h}^k$ is the correction to the kth approximation $\underset{\sim}{x}^k$ of the solution. If EPS is chosen too small, the iteration terminates when no better estimation of the solution can be obtained because of rounding errors, and then EPS will be set to 0.

MAXF is an INTEGER variable which must be set to an upper bound on the number of calls of FDF (i.e., the maximum number of residual functions evaluations). On exit MAXF contains the number of calls of FDF that have been performed by the package.

KEQS is an INTEGER variable which must be set to the number of successive iterations with identical sets of active residual functions that is required before a switch to Stage 2 is made. Normally, KEQS=3 is an appropriate value. If KEQS $\geq$ MAXF, the Stage 2 is never used. On exit KEQS contains the number of switches to Stage 2 that have taken place.

W is a REAL array which is used for working space. Its length is given by IW. On exit the first M elements of W contain the residual function values at the solution, i.e., $W(I)=f_I(\underset{\sim}{x})$, I=1,...,M.

IW is an INTEGER argument which must be set to the length of W. Its value must be at least

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

The values of IWR for a set of initial values of arguments M and N are given in Table 1.

ICH is an INTEGER argument which must be set to the unit number (or channel number) that is to be used for the printed output generated by the package. Usually it is the unit number of the file OUTPUT. If ICH is less than or equal to zero, no printed output will be generated by the package. The value of ICH is not changed by the package.

TABLE I

MINIMUM WORKSPACE FOR THE MMUM PACKAGE

M: \ N:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	51	90	139	198	267	346	435	534	643	762	891	1030	1179	1338	1507	1686	1875	2074	2283	2502
2	57	98	149	210	281	362	453	554	665	786	917	1058	1209	1370	1541	1722	1913	2114	2325	2546
3	63	106	159	222	295	378	471	574	687	810	943	1086	1239	1402	1575	1758	1951	2154	2367	2590
4	69	114	169	234	309	394	489	594	709	834	969	1114	1269	1434	1609	1794	1989	2194	2409	2634
5	75	122	179	246	323	410	507	614	731	858	995	1142	1299	1466	1643	1830	2027	2234	2451	2678
6	81	130	189	258	337	426	525	634	753	882	1021	1170	1329	1498	1677	1866	2065	2274	2493	2722
7	87	138	199	270	351	442	543	654	775	906	1047	1198	1359	1530	1711	1902	2103	2314	2535	2766
8	93	146	209	282	365	458	561	674	797	930	1073	1226	1389	1562	1745	1938	2141	2354	2577	2810
9	99	154	219	294	379	474	579	694	819	954	1099	1254	1419	1594	1779	1974	2179	2394	2619	2854
10	105	162	229	306	393	490	597	714	841	978	1125	1282	1449	1626	1813	2010	2217	2434	2661	2898
11	111	170	239	318	407	506	615	734	863	1002	1151	1310	1479	1658	1847	2046	2255	2474	2703	2942
12	117	178	249	330	421	522	633	754	885	1026	1177	1338	1509	1690	1881	2082	2293	2514	2745	2986
13	123	186	259	342	435	538	651	774	907	1050	1203	1366	1539	1722	1915	2118	2331	2554	2787	3030
14	129	194	269	354	449	554	669	794	929	1074	1229	1394	1569	1754	1949	2154	2369	2594	2829	3074
15	136	202	279	366	463	570	687	814	951	1098	1255	1422	1599	1786	1983	2190	2407	2634	2871	3118
16	143	210	289	378	477	586	705	834	973	1122	1281	1450	1629	1818	2017	2226	2445	2674	2913	3162
17	150	218	299	390	491	602	723	854	995	1146	1307	1478	1659	1850	2051	2262	2483	2714	2955	3206
18	157	226	309	402	505	618	741	874	1017	1170	1333	1506	1689	1882	2085	2298	2521	2754	2997	3250
19	164	234	319	414	519	634	759	894	1039	1194	1359	1534	1719	1914	2119	2334	2559	2794	3039	3294
20	171	242	329	426	533	650	777	914	1061	1218	1385	1562	1749	1946	2153	2370	2597	2834	3081	3338

TABLE I

MINIMUM WORKSPACE FOR THE MMUM PACKAGE

M: \ N:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	178	250	339	438	547	666	795	934	1083	1242	1411	1590	1779	1978	2187	2406	2635	2874	3123	3382
22	185	258	349	450	561	682	813	954	1105	1266	1437	1618	1809	2010	2221	2442	2673	2914	3165	3426
23	192	266	359	462	575	698	831	974	1127	1290	1463	1646	1839	2042	2255	2478	2711	2954	3207	3470
24	199	274	369	474	589	714	849	994	1149	1314	1489	1674	1869	2074	2289	2514	2749	2994	3249	3514
25	206	282	379	486	603	730	867	1014	1171	1338	1515	1702	1899	2106	2323	2550	2787	3034	3291	3558
26	213	290	389	498	617	746	885	1034	1193	1362	1541	1730	1929	2138	2357	2586	2825	3074	3333	3602
27	220	298	399	510	631	762	903	1054	1215	1386	1567	1758	1959	2170	2391	2622	2863	3114	3375	3646
28	227	306	409	522	645	778	921	1074	1237	1410	1593	1786	1989	2202	2425	2658	2901	3154	3417	3690
29	234	314	419	534	659	794	939	1094	1259	1434	1619	1814	2019	2234	2459	2694	2939	3194	3459	3734
30	241	323	429	546	673	810	957	1114	1281	1458	1645	1842	2049	2266	2493	2730	2977	3234	3501	3778
31	248	332	439	558	687	826	975	1134	1303	1482	1671	1870	2079	2298	2527	2766	3015	3274	3543	3822
32	255	341	449	570	701	842	993	1154	1325	1506	1697	1898	2109	2330	2561	2802	3053	3314	3585	3866
33	262	350	459	582	715	858	1011	1174	1347	1530	1723	1926	2139	2362	2595	2838	3091	3354	3627	3910
34	269	359	469	594	729	874	1029	1194	1369	1554	1749	1954	2169	2394	2629	2874	3129	3394	3669	3954
35	276	368	479	606	743	890	1047	1214	1391	1578	1775	1982	2199	2426	2663	2910	3167	3434	3711	3998
36	283	377	489	618	757	906	1065	1234	1413	1602	1801	2010	2229	2458	2697	2946	3205	3474	3753	4042
37	290	386	499	630	771	922	1083	1254	1435	1626	1827	2038	2259	2490	2731	2982	3243	3514	3795	4086
38	297	395	509	642	785	938	1101	1274	1457	1650	1853	2066	2289	2522	2765	3018	3281	3554	3837	4130
39	304	404	519	654	799	954	1119	1294	1479	1674	1879	2094	2319	2554	2799	3054	3319	3594	3879	4174
40	311	413	529	666	813	970	1137	1314	1501	1698	1905	2122	2349	2586	2833	3090	3357	3634	3921	4218

IPR is an INTEGER argument which controls the printed output generated by the package. It must be set by the user and is not changed by the package. The absolute value of IPR, as a decimal number, is "logically" composed of 4 fields

$$|IPR| = pqrs$$

where q, r and s are the least significant one-digit fields, and p is the remaining part of the number. If q is not equal to zero (i.e., q=1, ..., 9) then the first q evaluations of residual functions (i.e., the first q calls of FDF) are reported in the printed output. Further, if p is not equal to zero then every pth evaluation of residual functions is reported in the printed output. Consequently, if p=1, the value of q is insignificant because all function evaluations will be reported by the package. The fields p and q control the printing of residual function values only. Printing of partial derivatives is controlled by the fields r and s. If s is not equal to zero (and is not greater than q) then the values of partial derivatives calculated in the first s calls of FDF are reported in the printed output. If r is not equal to zero (and p is greater than zero) then every (p*r)th evaluation of partial derivatives is reported as well. Moreover, if q is equal to zero and p is not equal to 1 (i.e., when the first call of FDF is not reported by the package), then the "starting point" values of optimization variables $\underset{\sim}{x}^0$ and corresponding residual function values $\underset{\sim}{f}(\underset{\sim}{x}^0)$ are printed; if, at the same time, s is greater than zero, the values of partial derivatives are included in the "starting point" information. It should be noted that the values of partial derivatives can only be printed for those evaluations for which printing of residual function values is indicated.

Note: The function evaluations reported by the package are indexed by two numbers in the form i/j where

i is the consecutive number of function evaluation,

j indicates the stage of the iterative algorithm preceding the evaluation:

0 - initial function evaluation,

1 - Stage 1 iteration,

2 - Stage 2 iteration,

3 - unsuccessful Stage 2 iteration followed by Stage 1 iteration.

If the value of IPR is negative, the partial derivatives calculated by FDF are verified numerically by comparing values supplied by FDF with the differences of residual function values in the small environment of the starting point. All partial derivatives which differ from the numerically approximated ones by more than 1% (with respect to the numerical approximation) are reported in the printed output.

IFALL is an INTEGER variable which on exit contains information about the solution:

IFALL = -1 incorrect input data,

IFALL = 0 required accuracy obtained,

IFALL = 1 machine accuracy reached,

IFALL = 2 maximum number of function evaluations reached,

IFALL = 3 iteration terminated by the user.

Basic entry (subroutine MMUM2A)

The subroutine call is

```
CALL MMUM2A (FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,IFALL)
```

All arguments are the same as for the standard entry. It should be noted, however, that 2 arguments of the standard entry do not exist in this case (arguments ICH and IPR), since no printed output is generated for the basic entry to the package.

Original entry (subroutine MINI5W)

The subroutine call is

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

The arguments are generally the same as for the foregoing standard entry but some of them (MAXF,W) are used in a slightly different way. The detailed description is given in [1].

V. AUXILIARY SUBROUTINES

The package contains several auxiliary subroutines which can be used to change or to set the values of additional parameters controlling the form of the printed output generated by the package. All these subroutines (if used) should be called before the standard entry to the package.

Subroutine MMXHDR

Subroutine MMXHDR defines the title line which is printed within the page header. The title must be a string of up to 80 characters which is stored in consecutive elements of a REAL array, 10 characters in one element.

The subroutine call is

```
CALL MMXHDR(L,T)
```

where L is the number of array elements required for the title, and T is the name of an array or the first element storing the title. If L is equal to zero, no title line is printed by the package.

Subroutine MMXPSZ

Subroutine MMXPSZ defines the "page size", that is the maximum number of lines printed on a page. The preset value is 65.

The subroutine call is

```
CALL MMXPSZ(L)
```

where L is the defined page size. If the value of L is equal to zero, the printed output is generated without page control.

Subroutine MMXPLM

Subroutine MMXPLM defines the limit of printed pages. The preset value of this limit is 10, and it cannot be changed to more than 50.

The subroutine call is

```
CALL MMXPLM (L)
```

where L is the defined limit of pages.

When the limit of pages is reached the further output generated by the package is suppressed except of the results of optimization.

Subroutine MMXLLM

Subroutine MMXLLM defines the limit of printed lines. The preset value of this limit is 750.

The subroutine call is

```
CALL MMXLLM(L)
```

where L is the defined limit of lines.

When the limit of printed lines is reached the further output generated by the package is suppressed except of the results of optimization.

Subroutine MMXGLM

Subroutine MMXGLM defines the bounds on the number of variables and the number of residual functions when the matrix of partial derivatives is printed by the package (for some problems this matrix can be quite large and it can be reasonable to print the initial part of it only). The preset bound on the number of variables is 10, and on the number of functions is 25.

The subroutine call is

```
CALL MMXGLM(K,L)
```

where K is the defined bound on the number of variables, and L is the defined bound on the number of residual functions.

Subroutine MMXGVL

Subroutine MMXGVL defines, for the matrix of partial derivatives, the number of columns printed in one line. The preset value is 10, and it corresponds to 120 character lines. If the standard form of generated output is to be preserved this number should be defined as 6.

The subroutine call is

```
CALL MMXGVL(K)
```

where K is the defined number of columns per line.

VI. GENERAL INFORMATION

Use of COMMON: COMMON/MMX000/ (for standard entry only), COMMON/MMU000/ (see argument FDF).

Workspace: Provided by the user; see arguments W and IW.

Input/output: Output (for standard entry only) as defined by the user; see argument ICH.

Subroutines: MMUN5W, MMSUBB, MA19W, MA19BW, MULT, QUAD2, GAUSS, FPOS and:

a) for standard entry: MMUM1A, MMUM5W, MMX00Q, MMX00V, MMX00G, MMX00H, MMX00A, MMXPSZ, MMXPLM, MMXLLM, MMXHDR, MMXGLM, MMXGVL;

b) for basic entry: MMUM2A, MMUM5W, MMX00Z;

c) for original entry: MINI5W, MMX00Z.

Restrictions: N>0, M>0, DX>0, EPS≥0, MAXF>0, KEQS>0, IW≥IWR.

Date: March 1982.

VII. EXAMPLES

Example 1 [1, Example 1]

Minimize

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

where

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

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

The starting point is

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

To show the influence of the parameters DX and KEQS, the optimization has been performed several times for different values of DX and KEQS. The resulting numbers of residual function evaluations required to achieve the accuracy EPS = 10^{-6}, as well as the numbers of shifts to Stage 2 are summarized in the following table (the numbers of shifts are given in parentheses):

	KEQS		
DX	2	3	4
0.25	22(4)	23(3)	23(3)
0.5	20(4)	20(3)	20(2)
1.0	18(3)	19(3)	19(2)
2.0	20(3)	21(2)	21(2)

It can be observed that the increasing values of KEQS correspond, generally, to smaller numbers of shifts to Stage 2 (some too early shifts are eliminated), and to slightly increased numbers of residual function evaluations (see also Example 2). Moreover, too small and too large values of DX require more residual function evaluations because of adjustments which are performed by the package.

```
      PROGRAM TRMMU1(OUTPUT,TAPE6=OUTPUT)
C
C  K.MADSEN EXAMPLE
C
      DIMENSION X(2),W(98),T(3)
      EXTERNAL FUN
      DATA T/10HTRMMU1 : K,10H.MADSEN EX,10HAMPLE     /
      CALL MMXHDR(3,T)
      N=2
      M=2
      X(1)=3.
      X(2)=1.
      DX=1.0
      EPS=1.E-6
      MAXF=30
      KEQS=2
      IW=98
      IPR=-10
      CALL MMUM1A(FUN,N,M,X,DX,EPS,MAXF,KEQS,W,IW,6,IPR,IFALL)
      STOP
      END
C
C
      SUBROUTINE FUN(N,M,X,DF,F)
      DIMENSION X(N),DF(M,N),F(M)
      X1=X(1)
      X2=X(2)
      F(1)=X1*X1+X2*(X1+X2+X2)
      F(2)=SIN(X1)+COS(X2)
      DF(1,1)=X1+X1+X2
      DF(1,2)=4.0*X2+X1
      DF(2,1)=COS(X1)
      DF(2,2)=-SIN(X2)
      RETURN
      END
```

DATE : 82/04/29 TIME : 12.24.51 PAGE : 1
UNCONSTRAINED MINIMAX OPTIMIZATION (MMUM PACKAGE) (V:82.03)

TRMMU1 : K.MADSEN EXAMPLE

INPUT DATA

NUMBER OF VARIABLES (N) 2

NUMBER OF FUNCTIONS (M) 2

STEP LENGTH (DX) . . . . 1.000E+00

ACCURACY (EPS) 1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) 30

NUMBER OF SUCCESSIVE ITERATIONS (KEQS) 2

WORKING SPACE (IW) 98

PRINTOUT CONTROL (IPR) . . . . -10

STARTING POINT :

	VARIABLES		FUNCTION VALUES
1	3.000000000000E+00	1	1.400000000000E+01
2	1.000000000000E+00	2	6.814223139280E-01

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

SOLUTION

	VARIABLES		FUNCTION VALUES
1	-6.423372301388E-01	1	3.728580267894E-01
2	2.375113808568E-01	2	3.728580267894E-01

TYPE OF SOLUTION (IFALL) 0

NUMBER OF FUNCTION EVALUATIONS 18

NUMBER OF SHIFTS TO STAGE-2 3

EXECUTION TIME (IN SECONDS) 076

<u>Example 2</u> [3, Example 1]

Minimize

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

where

$$f_1(\underset{\sim}{x}) = 10(x_2 - x_1^2),$$

$$f_2(\underset{\sim}{x}) = 1 - x_1 \ .$$

The starting point is

$$\underset{\sim}{x}^0 = \begin{bmatrix} -1.2 \\ 1.0 \end{bmatrix} .$$

The solution is

$$\underset{\sim}{x}^* = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}$$

with $F(\underset{\sim}{x}^*) = 0$. The function $F(\underset{\sim}{x})$ has a "banana-shaped" valley like the Rosenbrock function

$$f(\underset{\sim}{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \ .$$

In fact, it is derived from the Rosenbrock function in the following way. It is well known (and easy to check) that the minimum value of $f(\underset{\sim}{x})$ is equal to 0, and the minimization of $f(\underset{\sim}{x})$ is thus equivalent to solving the nonlinear equation

$$f(\underset{\sim}{x}) = 0 \ .$$

However, $f(\underset{\sim}{x})$ is equal to the sum of 2 nonnegative terms, $100(x_2 - x_1^2)^2$ and $(1 - x_1)^2$. At the solution both these terms have to be equal to zero, and therefore the minimization of $f(\underset{\sim}{x})$ is equivalent to minimization of

$$\max(100(x_2 - x_1^2)^2,\ (1 - x_1)^2)$$

or the minimization of

$$F(\underset{\sim}{x}) = \max(|10(x_2 - x_1^2)|,\ |1 - x_1|) = \max(|f_1(\underset{\sim}{x})|,\ |f_2(\underset{\sim}{x})|).$$

The numbers of residual function evaluations and the numbers of shifts to Stage 2 corresponding to several values of the parameters KEQS and DX are given in the following table:

	KEQS				
DX	2	3	4	5	6
0.2	17(4)	20(0)	17(2)	20(0)	20(0)
0.4	22(5)	22(3)	22(2)	22(2)	22(1)
0.6	17(5)	27(4)	27(2)	27(1)	27(1)
0.8	17(4)	15(3)	27(4)	27(2)	27(2)
1.0	14(3)	14(2)	14(1)	14(1)	14(1)
1.2	12(2)	12(1)	12(1)	12(1)	12(0)
1.4	28(5)	23(5)	28(4)	26(2)	26(1)
1.6	46(14)	24(5)	16(2)	37(6)	33(3)

The numbers differ significantly more than those in Example 1, which is due to more nonlinearity; the selection of "good" parameters is thus more difficult in this case. It should be noted that the table contains many better results than reported in [3]. Moreover, the example shows that sometimes a very small change of the value of DX can significantly influence the required number of function evaluations.

```
      PROGRAM TRMMU2(OUTPUT,TAPE5=OUTPUT)
C
C  HALD-MADSEN EXAMPLE 1
C
      DIMENSION X(2),T(3),W(98)
      EXTERNAL FDF
      DATA T/10HTRMMU2 : H,10HALD-MADSEN,10H EXAMPLE 1/
      CALL MMXHDR(3,T)
      N=2
      M=2
      DXX=0.6
      DO 10 I=1,2
      X(1)=-1.2
      X(2)=1.0
      DX=DXX
      EPS=1.E-6
      MAXF=50
      KEQS=2
      IW=98
      ICH=5
      IPR=-10
      CALL MMUM1A(FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
      DXX=DXX-1.E-10
   10 CONTINUE
      STOP
      END
C
C
      SUBROUTINE FDF(N,M,X,DF,F)
      DIMENSION X(N),DF(M,N),F(M)
      X1=X(1)
      X2=X(2)
      F(1)=10.0*(X2-X1*X1)
      F(2)=1.0-X1
      DF(1,1)=-20.0*X1
      DF(1,2)=10.0
      DF(2,1)=-1.0
      DF(2,2)=0.0
      RETURN
      END
```

DATE : 82/04/29 TIME : 12.27.08 PAGE : 1
UNCONSTRAINED MINIMAX OPTIMIZATION (MMUM PACKAGE) (V:82.03)

TRMMU2 : HALD-MADSEN EXAMPLE 1

INPUT DATA

NUMBER OF VARIABLES (N) 2

NUMBER OF FUNCTIONS (M) 2

STEP LENGTH (DX) 6.000E-01

ACCURACY (EPS) 1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) 50

NUMBER OF SUCCESSIVE ITERATIONS (KEQS) 2

WORKING SPACE (IW) 98

PRINTOUT CONTROL (IPR) . . . . -10

STARTING POINT :

	VARIABLES		FUNCTION VALUES
1	-1.200000000000E+00	1	-4.400000000000E+00
2	1.000000000000E+00	2	2.200000000000E+00

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

SOLUTION

	VARIABLES		FUNCTION VALUES
1	1.000000000000E+00	1	3.552713678801E-14
2	1.000000000000E+00	2	7.105427357601E-15

TYPE OF SOLUTION (IFALL) 0

NUMBER OF FUNCTION EVALUATIONS 17

NUMBER OF SHIFTS TO STAGE-2 5

EXECUTION TIME (IN SECONDS) 087

```
DATE : 82/04/29                    TIME : 12.27.09                         PAGE :   1
UNCONSTRAINED MINIMAX OPTIMIZATION (MMUM PACKAGE)                          (V:82.03)

TRMMU2 : HALD-MADSEN EXAMPLE 1


INPUT DATA
----------

  NUMBER OF VARIABLES (N) . . . . . . . . . . . . . . . . . . . . . . . . .      2

  NUMBER OF FUNCTIONS (M) . . . . . . . . . . . . . . . . . . . . . . . . .      2

  STEP LENGTH (DX)  . . . . . . . . . . . . . . . . . . . . . . . . . 6.000E-01

  ACCURACY (EPS)  . . . . . . . . . . . . . . . . . . . . . . . . . . 1.000E-06

  MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) . . . . . . . . . . . . . . .     50

  NUMBER OF SUCCESSIVE ITERATIONS (KEQS) . . . . . . . . . . . . . . . . .      2

  WORKING SPACE (IW) . . . . . . . . . . . . . . . . . . . . . . . . . . .     98

  PRINTOUT CONTROL (IPR) . . . . . . . . . . . . . . . . . . . . . . . . .    -10

  STARTING POINT :

                          VARIABLES                       FUNCTION VALUES

                   1  -1.200000000000E+00            1  -4.400000000000E+00
                   2   1.000000000000E+00            2   2.200000000000E+00

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.


SOLUTION
--------

                          VARIABLES                       FUNCTION VALUES

                   1   1.000000000000E+00            1   3.552713678801E-14
                   2   1.000000000000E+00            2   7.105427357601E-15


  TYPE OF SOLUTION (IFALL)  . . . . . . . . . . . . . . . . . . . . . . . .      0

  NUMBER OF FUNCTION EVALUATIONS  . . . . . . . . . . . . . . . . . . . . .     11

  NUMBER OF SHIFTS TO STAGE-2 . . . . . . . . . . . . . . . . . . . . . . .      2

  EXECUTION TIME (IN SECONDS) . . . . . . . . . . . . . . . . . . . . . . .   .052
```

<u>Example 3</u> [5, Example 3]

This is the problem proposed by Brent [6] as an example in which the continuous analogue of the Newton-Raphson method is not globally convergent. The problem is to solve a system of 2 nonlinear equations

$$4(x_1+x_2) = 0 ,$$
$$(x_1-x_2)((x_1-2)^2 + x_2^2) + 3x_1 + 5x_2 = 0 .$$

More details and some solutions are given in [5]. It can be observed, however, that the solution can be obtained by minimizing the objective function

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

where

$$f_1(\underset{\sim}{x}) = 4(x_1+x_2) ,$$
$$f_2(\underset{\sim}{x}) = (x_1-x_2)((x_1-2)^2 + x_2^2) + 3x_1 + 5x_2 .$$

The solutions are shown for 4 different starting points $\underset{\sim}{x}^0$

$$\begin{bmatrix} 2 \\ 2 \end{bmatrix},\ \begin{bmatrix} -2 \\ -2 \end{bmatrix},\ \begin{bmatrix} 2 \\ 0 \end{bmatrix},\ \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

as in [5]. For this example all the solutions have been found in Stage 1 only.

```
      PROGRAM TRMMU3(OUTPUT,TAPE6=OUTPUT)
C
C   BRENT EXAMPLE
C
      DIMENSION X(2),XX(4,2),T(3),W(98)
      EXTERNAL FDF
      DATA XX/2.0,-2.0,2.0,2.0,
     1        2.0,-2.0,0.0,1.0/
      DATA T/10HTRMMU3 : B,10HRENT EXAMP,10HLE
      CALL MMXHDR(3,T)
      N=2
      M=2
      IPR=-10
      DO 20 I=1,4
      X(1)=XX(I,1)
      X(2)=XX(I,2)
      DX=0.2
      EPS=1.E-6
      MAXF=50
      KEQS=2
      IW=98
      ICH=6
      CALL MMUM1A(FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
      IPR=0
   20 CONTINUE
      STOP
      END
C
C
      SUBROUTINE FDF(N,M,X,DF,F)
      DIMENSION X(N),DF(M,N),F(M)
      X1=X(1)
      X2=X(2)
      R1=X1-X2
      R2=(X1-2.0)**2+X2*X2
      F(1)=4.0*(X1+X2)
      F(2)=R1*R2+3.0*X1+5.0*X2
      DF(1,1)=4.0
      DF(1,2)=4.0
      DF(2,1)=R2+(R1+R1)*(X1-2.0)+3.0
      DF(2,2)=-R2+R1*(X2+X2)+5.0
      RETURN
      END
```

DATE : 82/04/29 TIME : 16.13.30 PAGE : 1
UNCONSTRAINED MINIMAX OPTIMIZATION (MMUM PACKAGE) (V:82.03)

TRMMU3 : BRENT EXAMPLE

INPUT DATA

NUMBER OF VARIABLES (N) 2

NUMBER OF FUNCTIONS (M) 2

STEP LENGTH (DX) 2.000E-01

ACCURACY (EPS) 1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) 50

NUMBER OF SUCCESSIVE ITERATIONS (KEQS) 2

WORKING SPACE (IW) 98

PRINTOUT CONTROL (IPR) . . . . -10

STARTING POINT :

	VARIABLES		FUNCTION VALUES
1	2.000000000000E+00	1	1.600000000000E+01
2	2.000000000000E+00	2	1.600000000000E+01

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

SOLUTION

	VARIABLES		FUNCTION VALUES
1	0.	1	0.
2	0.	2	0.

TYPE OF SOLUTION (IFALL) 0

NUMBER OF FUNCTION EVALUATIONS 9

NUMBER OF SHIFTS TO STAGE-2 0

EXECUTION TIME (IN SECONDS) 029

DATE : 82/04/29 TIME : 16.13.31 PAGE : 1
UNCONSTRAINED MINIMAX OPTIMIZATION (MMUM PACKAGE) (V:82.03)

TRMMU3 : BRENT EXAMPLE

INPUT DATA

NUMBER OF VARIABLES (N) 2

NUMBER OF FUNCTIONS (M) 2

STEP LENGTH (DX) 2.000E-01

ACCURACY (EPS) 1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) 50

NUMBER OF SUCCESSIVE ITERATIONS (KEQS) 2

WORKING SPACE (IW) 98

PRINTOUT CONTROL (IPR) 0

STARTING POINT :

	VARIABLES		FUNCTION VALUES
1	-2.000000000000E+00	1	-1.600000000000E+01
2	-2.000000000000E+00	2	-1.600000000000E+01

SOLUTION

	VARIABLES		FUNCTION VALUES
1	0.	1	0.
2	0.	2	0.

TYPE OF SOLUTION (IFALL) 0

NUMBER OF FUNCTION EVALUATIONS 7

NUMBER OF SHIFTS TO STAGE-2 0

EXECUTION TIME (IN SECONDS) 023

DATE : 82/04/29 TIME : 16.13.31 PAGE : 1
UNCONSTRAINED MINIMAX OPTIMIZATION (MMUM PACKAGE) (V:82.03)

TRMMU3 : BRENT EXAMPLE

INPUT DATA

NUMBER OF VARIABLES (N) 2

NUMBER OF FUNCTIONS (M) 2

STEP LENGTH (DX) 2.000E-01

ACCURACY (EPS) 1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) 50

NUMBER OF SUCCESSIVE ITERATIONS (KEQS) 2

WORKING SPACE (IW) 98

PRINTOUT CONTROL (IPR) 0

STARTING POINT :

	VARIABLES		FUNCTION VALUES
1	2.000000000000E+00	1	8.000000000000E+00
2	0.	2	6.000000000000E+00

SOLUTION

	VARIABLES		FUNCTION VALUES
1	0.	1	0.
2	0.	2	0.

TYPE OF SOLUTION (IFALL) 0

NUMBER OF FUNCTION EVALUATIONS 15

NUMBER OF SHIFTS TO STAGE-2 0

EXECUTION TIME (IN SECONDS) 054

DATE : 82/04/29.   TIME : 16.14.15.   PAGE : 1
UNCONSTRAINED MINIMAX OPTIMIZATION (MMUM PACKAGE)   (V:82.03)

TRMMU3 : BRENT EXAMPLE

INPUT DATA

NUMBER OF VARIABLES (N) 2

NUMBER OF FUNCTIONS (M) 2

STEP LENGTH (DX) 2.000E-01

ACCURACY (EPS) 1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) 50

NUMBER OF SUCCESSIVE ITERATIONS (KEQS) 2

WORKING SPACE (IW) 98

PRINTOUT CONTROL (IPR) 0

STARTING POINT :

	VARIABLES		FUNCTION VALUES
1	2.000000000000E+00	1	1.200000000000E+01
2	1.000000000000E+00	2	1.200000000000E+01

SOLUTION

	VARIABLES		FUNCTION VALUES
1	0.	1	0.
2	0.	2	0.

TYPE OF SOLUTION (IFALL) 0

NUMBER OF FUNCTION EVALUATIONS 14

NUMBER OF SHIFTS TO STAGE-2 0

EXECUTION TIME (IN SECONDS) 056

Example 4 [3, Example 4]

This is the Rosen-Suzuki constrained minimization problem [7], slightly modified as indicated below. It is to minimize

$$f(\underset{\sim}{x}) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$

subject to constraints

$$- x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8 \geq 0 ,$$
$$- x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10 \geq 0 ,$$
$$- x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5 \geq 0 .$$

(The coefficient of x_1^2 in the third constraint is -1 not -2.)

The solution is $\underset{\sim}{x}^* = [0\ 1\ 2\ -1]^T$ with $f(\underset{\sim}{x}^*) = -44$.

To use the package, the formulation of the original problem has to be modified in several ways. Since the package minimizes the absolute values of residual functions, the negative solution $f(\underset{\sim}{x}^*)$ cannot be obtained. Therefore, instead of the function $f(\underset{\sim}{x})$, the function $f_1(\underset{\sim}{x}) = f(\underset{\sim}{x}) + c$ can be used where c is a positive constant which is equal to at least $f(\underset{\sim}{x}^*)$; c=100 is used in the example (as in [3]). Moreover, the constraints must be expressed in another form because the package performs unconstrained optimization. The common technique [8] is to transform constraints into additional residual functions

$$f_2(\underset{\sim}{x}) = f_1(\underset{\sim}{x}) - \alpha(- x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8) ,$$
$$f_3(\underset{\sim}{x}) = f_1(\underset{\sim}{x}) - \alpha(- x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10) ,$$
$$f_4(\underset{\sim}{x}) = f_1(\underset{\sim}{x}) - \alpha(- x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5) ,$$

where α is a positive constant. α=10 is used in the example (as in [3]). The minimax objective function is then

$$F(\underset{\sim}{x}) = \max_{1 \leq i \leq 4} |f_i(\underset{\sim}{x})| .$$

Finally, because the package minimizes the absolute values of residual functions $f_i(\underset{\sim}{x})$, at the solution $\underset{\sim}{x}^*$ the absolute values of transformed constraints $|f_2(\underset{\sim}{x}^*)|$, $|f_3(\underset{\sim}{x}^*)|$ and $|f_4(\underset{\sim}{x}^*)|$ must not be greater than $|f_1(\underset{\sim}{x}^*)|$, otherwise the solution found by the package will be incorrect (if this condition is not satisfied the constant c in $f_1(\underset{\sim}{x})$ should be increased).

Two solutions are shown which correspond to starting points $\underset{\sim}{x}^0=[2\ 2\ 5\ 0]^T$ and $\underset{\sim}{x}^0 = \underset{\sim}{0}$, as in [3], and both solutions require less residual function evaluations than reported in [3].

```
      PROGRAM TRMMU4(OUTPUT,TAPE6=OUTPUT)
C
C  ROSEN-SUZUKI PROBLEM
C
      DIMENSION X(4),W(234)
      EXTERNAL FDF
      COMMON NCALL
      CALL DATE(DAT)
      CALL TIME(TIM)
      N=4
      M=4
      X(1)=2.0
      X(2)=2.0
      X(3)=5.0
      X(4)=0.0
      DO 20 IX=1,2
      NCALL=0
      DX=0.5
      EPS=1.E-6
      MAXF=30
      KEQS=2
      IW=234
      WRITE(6,111) DAT,TIM,N,M,MAXF,KEQS,DX,EPS
  111 FORMAT(1H1/" PROGRAM : TRMMU4  -  DATE : ",A9," TIME : ",A10/
     1 " ROSEN-SUZUKI PROBLEM"//
     2 " NUMBER OF VARIABLES:",I4,10X,"NUMBER OF FUNCTIONS:",I4/
     3 " MAXF VALUE:         ",I4,10X,"KEQS VALUE:         ",I4/
     4 " DX VALUE:      ",E10.4,   10X,"EPS VALUE:    ",E10.4/)
      WRITE(6,222) (I,X(I),I=1,N)
  222 FORMAT(/" STARTING POINT:"/(20X,I2,2X,F12.8))
      CALL SECOND(TIME1)
      CALL MMUM2A(FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,IFALL)
      CALL SECOND(TIME2)
      PTIME=TIME2-TIME1
      WRITE(6,333) (I,X(I),I=1,N)
  333 FORMAT(//" SOLUTION:"/(20X,I2,2X,F12.8))
      WRITE(6,444) (I,W(I),I=1,M)
  444 FORMAT(/" FUNCTION VALUES:"/(20X,I2,2X,F12.8))
      DO 10 I=2,M
   10 W(I)=0.1*(W(1)-W(I))
      WRITE(6,555) (I,W(I),I=2,M)
  555 FORMAT(/" CONSTRAINTS:"/(20X,I2,2X,E12.5))
      WRITE(6,666) IFALL,MAXF,KEQS,PTIME
  666 FORMAT(/" TYPE OF SOLUTION (IFALL):        ",I4/
     1        " NUMBER OF FUNCTION EVALUATIONS:",I4/
     2        " NUMBER OF SHIFTS TO STAGE-2:   ",I4/
     3        " EXECUTION TIME (IN SECONDS):",F7.3/)
      DO 15 J=1,4
   15 X(J)=0.0
   20 CONTINUE
      STOP
      END
C
```

```
C
      SUBROUTINE FDF(N,M,X,DF,F)
      DIMENSION X(N),DF(M,N),F(M)
      COMMON NCALL
      DATA A /10.0/
      X1=X(1)
      X2=X(2)
      X3=X(3)
      X4=X(4)
      A2X1=A*(X1+X1+1.0)
      A2X2=A*(X2+X2-1.0)
      A2X3=A*(X3+X3)
      FF=100.0+X1*(X1-5.0)+X2*(X2-5.0)+X3*(X3+X3-21.0)+X4*(X4+7.0)
      F(1)=FF
      F(2)=FF-A*(X1*(-X1-1.0)+X2*(1.0-X2)+X3*(-X3-1.0)+X4*(1.0-X4)+8.0)
      F(3)=FF-A*(X1*(1.0-X1)-X2*(X2+X2)-X3*X3+X4*(1.0-X4-X4)+10.0)
      F(4)=FF-A*(X1*(-X1-2.0)+X2*(1.0-X2)-X3*X3+X4+5.0)
      D=X1+X1-5.0
      DF(1,1)=D
      DF(2,1)=D+A2X1
      DF(3,1)=D+A2X1-A-A
      DF(4,1)=D+A2X1+A
      D=X2+X2-5.0
      DF(1,2)=D
      DF(2,2)=D+A2X2
      DF(3,2)=D+A*4.0*X2
      DF(4,2)=D+A2X2
      D=4.0*X3-21.0
      DF(1,3)=D
      DF(2,3)=D+A2X3+A
      DF(3,3)=D+A2X3
      DF(4,3)=D+A2X3
      D=X4+X4+7.0
      DF(1,4)=D
      DF(2,4)=D-A*(1.0-X4-X4)
      DF(3,4)=D-A*(1.0-4.0*X4)
      DF(4,4)=D-A
      IF(NCALL.EQ.0) WRITE(6,111)
  111 FORMAT(/14X,"X(1)",9X,"X(2)",9X,"X(3)",9X,"X(4)",10X,"MAX(F)")
      NCALL=NCALL+1
      J=1
      DO 30 I=2,M
      IF(FF.GE.F(I)) GOTO 30
      FF=F(I)
      J=I
   30 CONTINUE
      WRITE(6,222) NCALL,(X(I),I=1,N),FF,J
  222 FORMAT(1X,I4,2X,4F13.8,F15.8,I4)
      RETURN
      END
```

PROGRAM : TRMMU4 - DATE : 82/04/22 TIME : 15.12.21.
ROSEN-SUZUKI PROBLEM

NUMBER OF VARIABLES:	4	NUMBER OF FUNCTIONS:	4
MAXF VALUE:	30	KEQS VALUE:	2
DX VALUE:	.5000E+00	EPS VALUE:	.1000E-05

STARTING POINT:

1	2.00000000
2	2.00000000
3	5.00000000
4	0.00000000

	X(1)	X(2)	X(3)	X(4)	MAX(F)	
1	2.00000000	2.00000000	5.00000000	0.00000000	333.00000000	2
2	1.50000000	1.50000000	4.50000000	.50000000	249.25000000	2
3	.50000000	.50000000	3.50000000	.25000000	128.93750000	2
4	1.57421875	2.50000000	1.61104911	-1.75000000	189.27274961	3
5	-.16666667	1.50000000	2.50000000	-.75000000	79.11805556	3
6	.83333333	.50000000	1.85995370	-1.75000000	87.04669817	2
7	.20833333	1.00000000	2.00000000	-1.18055556	62.39158951	2
8	-.29166667	.50000000	2.37970006	-.72946363	64.54807860	2
9	-.04166667	.75000000	2.11126531	-.94903600	57.76811046	2
10	.00340028	1.03032112	2.01401054	-.99577838	56.65489063	2
11	.00087176	.99953710	1.99983609	-1.00052898	56.00829245	2
12	-.00008510	1.00006030	2.00004386	-.99993569	56.00001186	2
13	.00000057	1.00000047	1.99999946	-1.00000055	56.00000013	2
14	-.00000001	.99999997	2.00000001	-.99999999	56.00000000	2

SOLUTION:

1	-.00000001
2	.99999997
3	2.00000001
4	-.99999999

FUNCTION VALUES:

1	56.00000000
2	56.00000000
3	45.99999892
4	56.00000000

CONSTRAINTS:

2	-.11369E-11
3	.10000E+01
4	-.72760E-12

TYPE OF SOLUTION (IFALL): 0
NUMBER OF FUNCTION EVALUATIONS: 14
NUMBER OF SHIFTS TO STAGE-2: 1
EXECUTION TIME (IN SECONDS): .185

PROGRAM : TRMMU4 - DATE : 82/04/22 TIME : 15.12.21.
ROSEN-SUZUKI PROBLEM

NUMBER OF VARIABLES: 4
MAXF VALUE: 30
DX VALUE: .5000E+00

NUMBER OF FUNCTIONS: 4
KEQS VALUE: 2
EPS VALUE: .1000E-05

STARTING POINT:

1	0.00000000
2	0.00000000
3	0.00000000
4	0.00000000

	X(1)	X(2)	X(3)	X(4)	MAX(F)	
1	0.00000000	0.00000000	0.00000000	0.00000000	100.00000000	1
2	.50000000	.50000000	.50000000	-.50000000	82.25000000	1
3	.91666667	1.50000000	1.50000000	-1.50000000	82.49305556	3
4	1.00000000	1.00000000	1.00000000	-1.00000000	67.00000000	1
5	.34482759	1.44827586	2.00000000	-1.17241379	71.32461356	3
6	.56666667	1.23333333	1.50000000	-1.50000000	68.05888889	3
7	.75000000	1.25000000	1.25000000	-1.25000000	63.68750000	4
8	.50495050	1.16212871	1.50000000	-1.42698020	60.32389288	3
9	.25495050	1.20620979	1.75000000	-1.24081013	58.05910359	2
10	.07158069	.95620979	2.00000000	-1.05221194	57.46411259	2
11	-.05341931	1.08120979	2.02029019	-.96626943	56.33023544	2
12	.00206446	.99797419	2.00188748	-1.00068031	56.08312609	2
13	-.00008655	.99997837	2.00007133	-.99992134	56.00008885	2
14	.00000067	1.00000561	1.99999792	-1.00000137	56.00000015	2
15	.00000063	.99999790	2.00000015	-1.00000024	56.00000000	2
16	-.00000004	1.00000001	2.00000003	-.99999997	56.00000000	2
17	.00000000	1.00000000	2.00000000	-1.00000000	56.00000000	1

SOLUTION:

1	.00000000
2	1.00000000
3	2.00000000
4	-1.00000000

FUNCTION VALUES:

1	56.00000000
2	56.00000000
3	46.00000003
4	56.00000000

CONSTRAINTS:

2	.11369E-12
3	.10000E+01
4	.45475E-13

TYPE OF SOLUTION (IFALL): 0
NUMBER OF FUNCTION EVALUATIONS: 17
NUMBER OF SHIFTS TO STAGE-2: 1
EXECUTION TIME (IN SECONDS): .208

Example 5

Minimize the Beale constrained function

$$f_1(\underset{\sim}{x}) = 9 - 8x_1 - 6x_2 - 4x_3 + 2x_1^2 + 2x_2^2 + x_3^2 + 2x_1x_2 + 2x_1x_3$$

subject to constraints

$$x_i \geq 0 \; , \; i = 1,2,3 \; ,$$

$$3 - x_1 - x_2 - 2x_3 \geq 0 \; .$$

The function has a minimum $f_1(\underset{\sim}{x}^*) = 1/9$ at the point $\underset{\sim}{x}^* = [4/3 \;\; 7/9 \;\; 4/9]^T$.

The same transformation of constraints into additional residual functions is used as in Example 4, but in this case $\alpha=1$ is assumed. Moreover, another technique is used to avoid the undesired effects of transformed constraints on the minimax optimization, due to the absolute value operator in the objective function; in this case the transformed constraints $f_i(\underset{\sim}{x})$, i=2, ..., 5, are forced to be nonnegative

$$f_2(\underset{\sim}{x}) = \begin{cases} f_1(\underset{\sim}{x})-x_1, & \text{if } f_1(\underset{\sim}{x})-x_1 \geq 0, \\ 0 & \text{, otherwise;} \end{cases}$$

$$f_3(\underset{\sim}{x}) = \begin{cases} f_1(\underset{\sim}{x})-x_2, & \text{if } f_1(\underset{\sim}{x})-x_2 \geq 0, \\ 0 & \text{, otherwise;} \end{cases}$$

$$f_4(\underset{\sim}{x}) = \begin{cases} f_1(\underset{\sim}{x})-x_3, & \text{if } f_1(\underset{\sim}{x})-x_3 \geq 0, \\ 0 & \text{, otherwise;} \end{cases}$$

$$f_5(\underset{\sim}{x}) = \begin{cases} f_1(\underset{\sim}{x})-(3-x_1-x_2-2x_3), & \text{if } f_1(\underset{\sim}{x})-(3-x_1-x_2-2x_3) \geq 0, \\ 0 & \text{, otherwise;} \end{cases}$$

and the objective function is

$$F(\underset{\sim}{x}) = \max_{1 \leq i \leq 5} |f_i(\underset{\sim}{x})| \; .$$

The solution is shown for the starting point

$$\underset{\sim}{x}^0 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

and for 3 values of DX, namely 0.25, 0.5 and 1.0. The least number of residual function evaluations as well as shifts to Stage 2 corresponds to the largest value of DX in this case. It can be observed that only one constraint (corresponding to $f_5(\underset{\sim}{x})$) is active at the solution.

```
      PROGRAM TRMMU5(OUTPUT,TAPE6=OUTPUT)
C
C    BEALE FUNCTION
C
      DIMENSION X(3),W(179)
      EXTERNAL FDF
      COMMON NCALL
      CALL DATE(DAT)
      CALL TIME(TIM)
      N=3
      M=5
      DDX=0.25
      DO 10 I=1,3
      NCALL=0
      X(1)=0.5
      X(2)=0.5
      X(3)=0.5
      DX=DDX
      EPS=1.E-6
      MAXF=50
      KEQS=2
      IW=179
      WRITE(6,111) DAT,TIM,N,M,MAXF,KEQS,DX,EPS
  111 FORMAT(1H1/" PROGRAM : TRMMU5 -  DATE :",A9," TIME :",A10/
     1 " BEALE FUNCTION"//
     2 " NUMBER OF VARIABLES:",I4,10X,"NUMBER OF FUNCTIONS:",I4/
     3 " MAXF VALUE:          ",I4,10X,"KEQS VALUE:          ",I4/
     4 " DX VALUE:        ",E10.4,    10X,"EPS VALUE:      ",E10.4/)
      WRITE(6,222) (J,X(J),J=1,N)
  222 FORMAT(/" STARTING POINT:"/(20X,I2,2X,F12.8))
      CALL SECOND(TIME1)
      CALL MMUM2A(FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,IFALL)
      CALL SECOND(TIME2)
      PTIME=TIME2-TIME1
      WRITE(6,333) (J,X(J),J=1,N)
  333 FORMAT(//" SOLUTION:"/(20X,I2,2X,F12.8))
      WRITE(6,444) (J,W(J),J=1,M)
  444 FORMAT(/" FUNCTION VALUES:"/(20X,I2,2X,F12.8))
      W(2)=X(1)
      W(3)=X(2)
      W(4)=X(3)
      W(5)=3.0-X(1)-X(2)-2.0*X(3)
      WRITE(6,555) (J,W(J),J=2,M)
  555 FORMAT(/" CONSTRAINTS:"/(20X,I2,2X,E12.5))
      WRITE(6,666) IFALL,MAXF,KEQS,PTIME
  666 FORMAT(/" TYPE OF SOLUTION (IFALL):          ",I4/
     1        " NUMBER OF FUNCTION EVALUATIONS:",I4/
     2        " NUMBER OF SHIFTS TO STAGE-2:     ",I4/
     3        " EXECUTION TIME (IN SECONDS):",F7.3/)
      DDX=2.0*DDX
   10 CONTINUE
      STOP
      END
C
```

```
C
      SUBROUTINE FDF(N,M,X,DF,F)
      DIMENSION X(N),F(M),DF(M,N)
      COMMON NCALL
      X1=X(1)
      X2=X(2)
      X3=X(3)
      F(1)=9.0-8.0*X1-6.0*X2-4.0*X3+2.0*(X1*(X1+X2+X3)+X2*X2)+X3*X3
      DF(1,1)=4.0*X1+2.0*(X2+X3)-8.0
      DF(1,2)=4.0*X2+2.0*X1-6.0
      DF(1,3)=2.0*(X1+X3)-4.0
      F(2)=F(1)-X1
      F(3)=F(1)-X2
      F(4)=F(1)-X3
      F(5)=F(1)-(3.0-X1-X2-2.0*X3)
      DO 10 I=1,3
      DO 10 J=1,3
      R=DF(1,J)
      IF(I.EQ.J) R=R-1.0
   10 DF(I+1,J)=R
      DF(5,1)=DF(1,1)+1.0
      DF(5,2)=DF(1,2)+1.0
      DF(5,3)=DF(1,3)+2.0
      DO 20 I=2,5
      IF(F(I).GT.0.0) GOTO 20
      F(I)=0.0
      DO 15 J=1,3
   15 DF(I,J)=0.0
   20 CONTINUE
      IF(NCALL.EQ.0) WRITE(6,111)
  111 FORMAT(/14X,"X(1)",9X,"X(2)",9X,"X(3)",10X,"MAX(F)")
      NCALL=NCALL+1
      FF=F(1)
      J=1
      DO 30 I=2,M
      IF(FF.GE.F(I)) GOTO 30
      FF=F(I)
      J=I
   30 CONTINUE
      WRITE(6,222) NCALL,(X(I),I=1,N),FF,J
  222 FORMAT(1X,I4,2X,3F13.8,F15.8,I4)
      RETURN
      END
```

PROGRAM : TRMMU5 - DATE: 82/04/22 TIME : 15.16.36.
BEALE FUNCTION

NUMBER OF VARIABLES: 3 NUMBER OF FUNCTIONS: 5
MAXF VALUE: 50 KEQS VALUE: 2
DX VALUE: .2500E+00 EPS VALUE: .1000E-05

STARTING POINT:

1	.50000000
2	.50000000
3	.50000000

	X(1)	X(2)	X(3)	MAX(F)	
1	.50000000	.50000000	.50000000	2.25000000	1
2	.71052632	.71052632	.71052632	.75415512	1
3	.80110159	.80110159	.69889841	.44780289	1
4	1.02624310	1.02624310	.47375690	.25344350	1
5	1.26704769	.78543852	.47375690	.11597369	1
6	1.79617754	1.31456838	-.05537296	1.45526499	4
7	1.76704769	1.28543852	-.02624310	1.28262773	4
8	1.51704769	1.03543852	.22375690	.37367916	5
9	1.39204769	.91043852	.34875690	.16670143	1
10	1.32954769	.84793852	.41125690	.12180631	1
11	1.26704769	.78543852	.47375690	.11597369	1
12	1.29829769	.81668852	.44250690	.11400719	1
13	1.32954769	.78543852	.44250690	.11121757	1
14	1.33102462	.77700821	.44598358	.11112177	1
15	1.33459646	.77746119	.44397118	.11111273	1
16	1.33329800	.77782368	.44443916	.11111111	1
17	1.33333368	.77777482	.44444575	.11111111	1
18	1.33333336	.77777778	.44444443	.11111111	1
19	1.33333333	.77777778	.44444444	.11111111	1

SOLUTION:

1	1.33333333
2	.77777778
3	.44444444

FUNCTION VALUES:

1	.11111111
2	0.00000000
3	0.00000000
4	0.00000000
5	.11111111

CONSTRAINTS:

2	.13333E+01
3	.77778E+00
4	.44444E+00
5	.35527E-14

TYPE OF SOLUTION (IFALL): 0
NUMBER OF FUNCTION EVALUATIONS: 19
NUMBER OF SHIFTS TO STAGE-2: 3
EXECUTION TIME (IN SECONDS): .181

PROGRAM : TRMMU5 - DATE : 82/04/22 TIME : 15.16.36.
BEALE FUNCTION

NUMBER OF VARIABLES: 3
NUMBER OF FUNCTIONS: 5
MAXF VALUE: 50
KEQS VALUE: 2
DX VALUE: .5000E+00
EPS VALUE: .1000E-05

STARTING POINT:

1	.50000000
2	.50000000
3	.50000000

	X(1)	X(2)	X(3)	MAX(F)	
1	.50000000	.50000000	.50000000	2.25000000	1
2	.71052632	.71052632	.71052632	.75415512	1
3	.80110159	.80110159	.69889841	.44780289	1
4	1.02624310	1.02624310	.47375690	.25344350	1
5	1.26704769	.78543852	.47375690	.11597369	1
6	1.79617754	1.31456838	-.05537296	1.45526499	4
7	1.79617754	1.31456838	-.05537296	1.45526499	4
8	1.76704769	1.28543852	-.02624310	1.28262773	4
9	1.51704769	1.03543852	.22375690	.37367916	5
10	1.39204769	.91043852	.34875690	.16670143	5
11	1.68578033	.54281312	.38570328	.26638471	1
12	1.36079769	.87918852	.38000690	.13937106	1
13	1.34925678	.76716215	.44179054	.11142806	1
14	1.32833497	.77966516	.44599994	.11113620	1
15	1.33334949	.77779957	.44442547	.11111111	5
16	1.33333327	.77777708	.44444482	.11111111	1
17	1.33333333	.77777778	.44444444	.11111111	1

SOLUTION:

1	1.33333333
2	.77777778
3	.44444444

FUNCTION VALUES:

1	.11111111
2	0.00000000
3	0.00000000
4	0.00000000
5	.11111111

CONSTRAINTS:

2	.13333E+01
3	.77778E+00
4	.44444E+00
5	.10658E-13

TYPE OF SOLUTION (IFALL): 0
NUMBER OF FUNCTION EVALUATIONS: 17
NUMBER OF SHIFTS TO STAGE-2: 2
EXECUTION TIME (IN SECONDS): .153

PROGRAM : TRMMU5 - DATE : 82/04/22 TIME : 15.16.36.
BEALE FUNCTION

NUMBER OF VARIABLES: 3
MAXF VALUE: 50
DX VALUE: .1000E+01

NUMBER OF FUNCTIONS: 5
KEQS VALUE: 2
EPS VALUE: .1000E-05

STARTING POINT:

1	.50000000
2	.50000000
3	.50000000

	X(1)	X(2)	X(3)	MAX(F)	
1	.50000000	.50000000	.50000000	2.25000000	1
2	.71052632	.71052632	.71052632	.75415512	1
3	.80110159	.80110159	.69889841	.44780289	1
4	1.02624310	1.02624310	.47375690	.25344350	1
5	1.26704769	.78543852	.47375690	.11597369	1
6	1.79617754	1.31456838	-.05537296	1.45526499	4
7	1.79617754	1.31456838	-.05537296	1.45526499	4
8	1.79617754	1.31456838	-.05537296	1.45526499	4
9	1.76704769	1.28543852	-.02624310	1.28262773	4
10	1.51704769	1.03543852	.22375690	.37367916	5
11	1.39204769	.91043852	.34875690	.16670143	1
12	1.34990932	.76672712	.44168178	.11145457	5
13	1.32732170	.78049533	.44609149	.11114840	1
14	1.33334130	.77778794	.44443538	.11111111	1
15	1.33333328	.77777759	.44444456	.11111111	1
16	1.33333333	.77777778	.44444444	.11111111	1

SOLUTION:

1	1.33333333
2	.77777778
3	.44444444

FUNCTION VALUES:

1	.11111111
2	0.00000000
3	0.00000000
4	0.00000000
5	.11111111

CONSTRAINTS:

2	.13333E+01
3	.77778E+00
4	.44444E+00
5	.35527E-14

TYPE OF SOLUTION (IFALL): 0
NUMBER OF FUNCTION EVALUATIONS: 16
NUMBER OF SHIFTS TO STAGE-2: 1
EXECUTION TIME (IN SECONDS): .156

VIII. REFERENCES

[1] K. Madsen (Adapted and Edited by J.W. Bandler and W.M. Zuberek), "MINI5W - A Fortran package for minimax optimization", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-280, 1981.

[2] K. Madsen and H. Schjaer-Jacobsen, "Linearly constrained minimax optimization", Mathematical Programming, vol. 14, 1978, pp. 208-223.

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

[4] K. Madsen, "An algorithm for minimax solution of overdetermined systems of non-linear equations", J. Inst. of Mathematics and its Applications, vol. 16, 1975, pp. 321-328.

[5] S. Incerti, V. Parisi and F. Zirilli, A new method for solving nonlinear simultaneous equations", SIAM J. Numerical Analysis, vol. 16, 1979, pp. 779-789.

[6] R.P. Brent, "On the Davidenko-Branin method for solving simultaneous nonlinear equations", IBM J. Research and Development, vol. 16, 1972, pp. 434-436.

[7] J.B. Rosen and S. Suzuki, "Construction of non-linear programming test problems", Comm. ACM, vol. 8, 1965, p. 113.

[8] J.W. Bandler and C. Charalambous, "Nonlinear programming using minimax techniques", J. Optimization Theory and Applications, vol. 13, 1974, pp. 607-619.

APPENDIX

LISTING OF THE MMUM PACKAGE

```
      SUBROUTINE MMUM1A (FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,LCH,IPR,IFALL)
      EXTERNAL FDF,MMX00Q,MMX00A
C
C          LEVEL 1 INTERFACE (STANDARD ENTRY)
C
      DIMENSION X(1), W(1)
      COMMON /MMX000/ NCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      NCH=LCH
      IF (LCH.LE.0) GO TO 40
      I=IABS(IPR)
      J=I/10
      LG2=MOD(I,10)
      I=J/10
      LG1=MOD(J,10)
      J=I/10
      LV2=MOD(I,10)
      LV1=J
      LG1=LG1*LV1
      NRP=0
      CALL MMXPSZ (-1)
      CALL MMXPLM (-1)
      CALL MMXLLM (-1)
      CALL MMXHDR (-1,H)
      CALL MMXGLM (-1,-1)
      CALL MMXGVL (-1)
      IF (MXL.NE.0) LML=MXL*LMP+100
      IF (MXL.EQ.0) MXL=LML+100
      CALL DATE (DAT)
      CALL TIME (TIM)
      CALL MMX00A
      WRITE (LCH,10) N,M,DX,EPS,MAXF,KEQS,IW,IPR
   10 FORMAT (11H0INPUT DATA/11H ----------//
     1 27H    NUMBER OF VARIABLES (N),25(2H. ),I4//
     2 27H    NUMBER OF FUNCTIONS (M) ,25(2H. ),I4//
     3 21H    STEP LENGTH (DX)  ,25(2H. ),1PE10.3//
     4 19H    ACCURACY (EPS)  ,26(2H. ),1PE10.3//
     5 45H    MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) ,16(2H. ),I4//
     6 43H    NUMBER OF SUCCESSIVE ITERATIONS (KEQS)  ,17(2H. ),I4//
     7 22H    WORKING SPACE (IW)  ,26(2H. ),1H.,I6//
     8 26H    PRINTOUT CONTROL (IPR) ,24(2H. ),1H.,I6/)
      NRL=NRL-20
      LML=LML-20
      IF (LV2.NE.0.OR.LV1.EQ.1) GO TO 30
      WRITE (LCH,20)
   20 FORMAT (19H    STARTING POINT :)
      NRL=NRL-1
      LML=LML-1
      CALL FDF (N,M,X,W(M+1),W(1))
      CALL MMX00V (MMX00A,X,N,W,M)
      IF (LG2.NE.0) CALL MMX00G (MMX00A,W(M+1),M,N)
   30 IF (IPR.GE.0) GO TO 40
      I=M*N+M+1
      J=I+M
      K=J+M
      CALL MMX00H (MMX00A,FDF,N,M,X,W(M+1),W(1),W(J),W(K),W(I))
   40 CALL SECOND (TBEG)
      CALL MMUM5W (MMX00Q,MMX00A,FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,IFALL)
      CALL SECOND (TEND)
      IF (LCH.LE.0) RETURN
      IF (IFALL.LT.0) GO TO 70
      IF (NRL.LT.9) CALL MMX00A
      WRITE (LCH,50)
   50 FORMAT (//9H SOLUTION/9H ---------)
      NRL=NRL-4
```

```
      LML=LML-4
      CALL MMX00V (MMX00A,X,N,W,M)
      CPU=TEND-TBEG
      IF (NRL.LT.9) CALL MMX00A
      WRITE (LCH,60) IFALL,MAXF,KEQS,CPU
   60 FORMAT (//29H    TYPE OF SOLUTION (IFALL)    ,24(2H. ),I4//
     1 35H    NUMBER OF FUNCTION EVALUATIONS    ,21(2H. ),I4//
     2 31H    NUMBER OF SHIFTS TO STAGE-2 ,23(2H. ),I4//
     3 31H    EXECUTION TIME (IN SECONDS) ,21(2H. ),1H.,F7.3/)
      RETURN
   70 WRITE (LCH,80)
   80 FORMAT (///40H I N C O R R E C T   P A R A M E T E R S/)
      RETURN
      END
C
C
      SUBROUTINE MMUM2A (FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,IFALL)
      EXTERNAL FDF,MMX00Z
C
C          LEVEL 2 INTERFACE (BASIC ENTRY)
C
      DIMENSION X(1), W(1)
      CALL MMUM5W (MMX00Z,MMX00Z,FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,IFALL)
      RETURN
      END
C
C
      SUBROUTINE MMUM5W (FQQ,FHH,FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,IFALL)
      DIMENSION X(1), W(1)
      EXTERNAL FQQ,FHH,FDF
      COMMON /MMU000/ MARK
      DATA XZERO/0.0/
      DELTA=0.01
      IFALL=0
      L=13+2*M*(2+N)+2*N*(8+N)+MAX0(M,3*N*(N+2)+5)
      IF (N.LE.0) GO TO 10
      IF (M.LE.0) GO TO 10
      IF (DX.LE.XZERO) GO TO 10
      IF (EPS.LT.XZERO) GO TO 10
      IF (MAXF.LE.0) GO TO 10
      IF (L.LE.IW.AND.IW.GT.0) GO TO 20
   10 IFALL=-1
      MAXF=0
      KEQS=0
      RETURN
   20 CALL MMUN5W (FQQ,FHH,FDF,N,M,X,DX,EPS,MAXF,KEQS,DELTA,W,IW)
      MIT=W(M+1)
      IF (MIT.LE.MAXF.AND.EPS.EQ.XZERO) IFALL=1
      IF (MIT.GT.MAXF) IFALL=2
      IF (MIT.LT.MAXF) MAXF=MIT
      IF (MARK.EQ.0) IFALL=3
      RETURN
      END
C
C
      SUBROUTINE MINI5W (FDF,N,M,X,DX,EPS,MAXF,W,IW)
C
C          LEVEL 3 INTERFACE (K.MADSEN ENTRY)
C
      DIMENSION X(1), W(1)
      EXTERNAL FDF,MMX00Z
      DATA XZERO/0.0/
      DELTA=W(1)
      KEQS=W(2)
      IWR=13+2*M*(2+N)+2*N*(8+N)+MAX0(M,3*N*(N+2)+5)
```

```
      IF (IWR.GT.IW.OR.N.LE.0.OR.M.LE.0.OR.DX.LE.XZERO.OR.MAXF.LE.0.OR.
     1    EPS.LT.XZERO) STOP 77777
      CALL MMUN5W (MMX00Z,MMX00Z,FDF,N,M,X,DX,EPS,MAXF,KEQS,DELTA,W,IW)
      RETURN
      END
C
C
      SUBROUTINE MMX00Z (FUN,N,M,X,DF,F,K,NS)
C
C     DUMMY SUBROUTINE WHICH FOR BASIC AND ORIGINAL ENTRIES SUBSTITUTES
C     SUBROUTINE MMX00Q/11Q.
C
      EXTERNAL FUN
      DIMENSION X(N), DF(M,N), F(M)
      RETURN
      END
C
C
      SUBROUTINE MMX00Q (FHH,N,M,X,DF,F,K,NS)
C
C     PRINT RESULTS OF FUNCTION EVALUATION.
C
      EXTERNAL FHH
      DIMENSION X(N), DF(M,N), F(M)
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      IF (LCH.LE.0) RETURN
      IF (LV1+LV2.EQ.0) RETURN
      IF (K.LE.LV2) GO TO 10
      IF (LV1.EQ.0) RETURN
      IF (MOD(K,LV1).NE.0) RETURN
   10 IF (NRP.LE.LMP.AND.LML.GE.0) GO TO 30
      LV1=0
      LV2=0
      WRITE (LCH,20)
   20 FORMAT (//26H ( LISTING LIMIT REACHED )//)
      NRL=NRL-5
      LML=LML-5
      RETURN
   30 IF (NRL.LT.7) CALL FHH
      WRITE (LCH,40) K,NS
   40 FORMAT (22H0FUNCTION EVALUATION :,I4,2H /,I2)
      NRL=NRL-2
      LML=LML-2
      CALL MMX00V (FHH,X,N,F,M)
      IF (LG1+LG2.EQ.0) RETURN
      IF (K.LE.LG2) GO TO 50
      IF (K.LE.LV2) RETURN
      IF (LG1.EQ.0) RETURN
      IF (MOD(K,LG1).NE.0) RETURN
   50 CALL MMX00G (FHH,DF,M,N)
      RETURN
      END
C
C
      SUBROUTINE MMX00V (FHH,X,N,F,M)
C
C     PRINT VALUES OF VARIABLES AND RESIDUAL FUNCTIONS.
C
      DIMENSION X(N), F(M)
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      IF (LCH.LE.0) RETURN
      K=MAX0(N,M)
      IF (NRL.LT.5) CALL FHH
```

```
      WRITE (LCH,10)
   10 FORMAT (/30X,9HVARIABLES,18X,15HFUNCTION VALUES/)
      NRL=NRL-3
      LML=LML-3
      DO 40 I=1,K
      IF (NRL.LE.0) CALL FHH
      IF (I.LE.N.AND.I.LE.M) WRITE (LCH,20) I,X(I),I,F(I)
      IF (I.LE.N.AND.I.GT.M) WRITE (LCH,20) I,X(I)
      IF (I.GT.N.AND.I.LE.M) WRITE (LCH,30) I,F(I)
   20 FORMAT (18X,I4,2X,1PE19.12,5X,I4,2X,1PE19.12)
   30 FORMAT (48X,I4,2X,1PE19.12)
      NRL=NRL-1
      LML=LML-1
   40 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE MMX00G (FHH,G,M,N)
C
C     PRINT PARTIAL DERIVATIVES OF RESIDUAL FUNCTIONS.
C
      DIMENSION G(M,N)
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      IF (LCH.LE.0) RETURN
      IF (NRL.LT.7) CALL FHH
      MM=MIN0(M,LMF)
      NN=MIN0(N,LMV)
      WRITE (LCH,10)
   10 FORMAT (30H0  GRADIENTS ( DF.I / DX.J ) :)
      NRL=NRL-2
      LML=LML-2
      DO 60 K=1,NN,LGH
      IF (NRL.LT.5) CALL FHH
      J1=K
      J2=MIN0(NN,K+LGH-1)
      WRITE (LCH,20) (J,J=J1,J2)
   20 FORMAT (1H0,9X,12HVARIABLES(J),10(I5,5X))
      WRITE (LCH,30)
   30 FORMAT (10X,12HFUNCTIONS(I))
      NRL=NRL-3
      LML=LML-3
      DO 50 I=1,MM
      IF (NRL.LE.0) CALL FHH
      WRITE (LCH,40) I,(G(I,J),J=J1,J2)
   40 FORMAT (10X,I6,4X,10(1PE10.2))
      NRL=NRL-1
      LML=LML-1
   50 CONTINUE
   60 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE MMX00H (FHH,FDF,N,M,X,DF,F,DG,DH,G)
C
C     NUMERICAL VERIFICATION OF USER-DEFINED PARTIAL DERIVATIVES
C     (VARIABLES ARE DISTURBED ONE BY ONE).
C
      DIMENSION X(N), DF(M,N), F(M), DG(M), DH(M,N), G(M)
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      IF (LCH.LE.0) RETURN
      K=0
```

```
      CALL FDF (N,M,X,DF,F)
      DO 60 I=1,N
      Z=X(I)
      DX=1.E-6*Z
      IF (ABS(DX).LT.1.E-10) DX=1.E+10
      DX2=DX+DX
      X(I)=Z+DX
      CALL FDF (N,M,X,DH,F)
      DO 10 J=1,M
      DG(J)=DH(J,I)
   10 CONTINUE
      X(I)=Z-DX
      CALL FDF (N,M,X,DH,G)
      X(I)=Z
      DO 50 J=1,M
      Y=DF(J,I)
      Z=F(J)-G(J)
      IF (ABS(Z).LE.0.5E-13*(F(J)+G(J))) Z=0.0
      Z=Z/DX2
      IF (ABS(Y).LE.1.E-20.AND.ABS(Z).LE.1.E-20) GO TO 50
      IF (ABS(Z).LT.1.E-20) Z=SIGN(1.E-20,Z)
      R=100.0*ABS((Z-Y)/Z)
      IF (R.LE.1.0) GO TO 50
      IF (SIGN(1.0,DG(J))+SIGN(1.0,DH(J,I)).EQ.0.0) GO TO 50
      IF (K.NE.0) GO TO 30
      IF (NRL.LT.5) CALL FHH
      WRITE (LCH,20)
   20 FORMAT (38H0VERIFICATION OF PARTIAL DERIVATIVES :/
     1 1H0,18X,52H DF.I / DX.J  :  USER DEFINED  NUMERICAL  DIFFERENCE)
      NRL=NRL-4
      LML=LML-4
   30 K=K+1
      IF (NRL.LE.0) CALL FHH
      WRITE (LCH,40) J,I,Y,Z,R
   40 FORMAT (19X,I5,3X,I4,6X,1PE10.3,2X,1PE10.3,4X,0PF6.1,2H %)
      NRL=NRL-1
      LML=LML-1
   50 CONTINUE
   60 CONTINUE
      IF (K.NE.0) GO TO 80
      IF (NRL.LT.2) CALL FHH
      WRITE (LCH,70)
   70 FORMAT (47H0VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.)
      NRL=NRL-2
      LML=LML-2
   80 RETURN
      END
C
C
      SUBROUTINE MMX00A
C
C     CHANGE PAGE AND PRINT PAGE HEADER.
C
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LC
     1H,DAT,TIM,LHT,H(8)
      IF (LCH.LE.0) RETURN
      IF (NRP.LT.LMP) GO TO 20
      LV1=0
      LV2=0
      WRITE (LCH,10)
   10 FORMAT (//27H ( LIMIT OF PAGES REACHED ))
   20 NRP=NRP+1
      NRL=MXL-5
      LML=LML-5
      WRITE (LCH,30) DAT,TIM,NRP
```

```
   30 FORMAT (1H1/7H DATE :,A10,19X,6HTIME :,A10,20X,6HPAGE :,I3/
     1 50H UNCONSTRAINED MINIMAX OPTIMIZATION (MMUM PACKAGE),22X,
     2 9H(V:82.03))
      IF (LHT.LE.0) GO TO 50
      WRITE (LCH,40) (H(J),J=1,LHT)
   40 FORMAT (1H0,8A10)
      NRL=NRL-2
      LML=LML-2
   50 WRITE (LCH,60)
   60 FORMAT (1H0)
      RETURN
      END
C
C
      SUBROUTINE MMXPSZ (L)
C
C     DEFINE THE PAGE SIZE (I.E. THE NUMBER OF LINES PER PAGE).
C
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      DATA LL/65/
      IF (L.GT.0) LL=MAX0(25,L)
      IF (L.EQ.0) LL=0
      MXL=LL
      RETURN
      END
C
C
      SUBROUTINE MMXPLM (L)
C
C     DEFINE THE LIMIT OF PRINTED PAGES.
C
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      DATA LL/10/
      IF (L.GT.0) LL=MIN0(50,L)
      LMP=LL
      RETURN
      END
C
C
      SUBROUTINE MMXLLM (L)
C
C     DEFINE THE LIMIT OF PRINTED LINES.
C
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      DATA LL/750/
      IF (L.GT.0) LL=L
      LML=LL
      RETURN
      END
C
C
      SUBROUTINE MMXHDR (L,T)
C
C     DEFINE THE HEADER LINE.
C
      DIMENSION T(1)
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      DATA LL/0/
      IF (L.GE.0) LL=MIN0(8,L)
      LHT=LL
      IF (L.LE.0) RETURN
```

```
      DO 10 I=1,LL
      H(I)=T(I)
   10 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE MMXGLM (K,L)
C
C     DEFINE THE SIZE OF PRINTED JACOBIAN.
C
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      DATA KK/25/,LL/10/
      IF (K.GT.0) KK=K
      IF (L.GT.0) LL=L
      LMF=KK
      LMV=LL
      RETURN
      END
C
C
      SUBROUTINE MMXGVL (L)
C
C     DEFINE THE NUMBER OF JACOBIAN COLUMNS PRINTED IN ONE LINE.
C
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      DATA LL/10/
      IF (L.GT.0) LL=MAX0(MIN0(10,L),5)
      LGH=LL
      RETURN
      END
C
C
      SUBROUTINE MMUN5W (FQQ,FHH,FDF,N,M,X,RDX,EPS,MAXFUN,KEQS,DELTA,W,I
     1W)
C
C     MINIMAX OPTIMIZATION USING QUADRATIC PROGRAMMING.
C     KAJ MADSEN, NUMERISK INSTITUT, LYNGBY, DENMARK.
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 FQQ,FHH,FDF
      COMMON /MMU000/ MARK
      MARK=1
      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 (FQQ,FHH,FDF,N,M,X,RDX,EPS,MAXFUN,W(NF0),W(NF1),W(NDF0
     1),W(NDF1),W(NX1),W(NH),W(NB),W(NIND1),W(NIND0),IIND1,W(NWO),IWO,W(
     2NY),W(NBH),W(NRLAM),IRLAM,N1,N2,KEQS,DELTA)
      RETURN
      END
C
C
      SUBROUTINE MMSUBB (FQQ,FHH,FDF,N,M,X0,RDX,EPS,MAXFUN,F0,F1,DF0,DF1
     1,X1,H,B,IND1,IND0,IIND1,WO,IWO,Y,BH,RLAM,IRLAM,N1,N2,IEQUAL,DEL)
      EXTERNAL FQQ,FHH,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 FUNCTIONS 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 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
      COMMON /MMU000/ MARK
      DATA XZERO,XONE,XTWO,XZERO2,XZERO5,XZERO8/0.0,1.0,2.0,0.2,0.5,0.8/
     1 ,X1M50/1.0E-50/
C
C        EPSFL IS THE SMALLEST MACHINE NUMBER X FOR WHICH  1+X>1
C
      EPSFL=2.0*16.0**(-12)
      IWO2=SQRT(FLOAT(IWO))
      NEWTON=.FALSE.
      KACT0=0
      RDXFIX=RDX
      KBOUND=MAX0(IEQUAL,2)
      KEQUAL=0
      NTAL=0
      NQUAD=0
C
C        FIND THE LENGTH OF THE STARTING VECTOR
C
      X0MAX=XZERO
      DO 10 I=1,N
      X0MAX=AMAX1(X0MAX,ABS(X0(I)))
   10 CONTINUE
C
C        CALCULATE FUNCTION VALUES
C
      NTAL=NTAL+1
      CALL FPOS (FQQ,FHH,FDF,NTAL,0,N,M,X0,DF0,F0)
      F0MAX=XZERO
      DO 20 J=1,M
      IF (F0(J).GT.F0MAX) F0MAX=F0(J)
   20 CONTINUE
C
```

```
C        INITIALIZE ARRAYS
C
      DO 40 I=1,N
      DO 30 J=1,N
      B(I,J)=XZERO
   30 CONTINUE
      B(I,I)=XONE
   40 CONTINUE
C
C        ITERATIVE LOOP STARTS HERE
C
   50 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 70
      NSTAGE=1
   60 CALL MA19W (N,M,DF0,M,F0,RDX,XZERO,H,WO,IND1,IWO,IIND1)
      F1PRED=WO(M+1)
      GO TO 80
   70 CONTINUE
      NSTAGE=2
      CALL QUAD2 (N,M,DF0,F0,B,H,RLAM,N1,IND1,KACT1,WO,IWO2,EPSFL,I)
      IF (I.NE.0) GO TO 100
      F1PRED=H(N1)
   80 CONTINUE
      IF (F1PRED.GT.F0MAX) GO TO 410
C
C        FIND THE NORM OF H, AND FIND THE POINT X+H
C
      HMAX=XZERO
      DO 90 I=1,N
      HMAX=AMAX1(HMAX,ABS(H(I)))
      X1(I)=X0(I)+H(I)
   90 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 110
  100 NEWTON=.FALSE.
      KEQUAL=0
      NSTAGE=3
      GO TO 60
C
C        FIND THE NEW FUNCTION VALUES
C
  110 NTAL=NTAL+1
      CALL FPOS (FQQ,FHH,FDF,NTAL,NSTAGE,N,M,X1,DF1,F1)
      IF (MARK.EQ.0) GO TO 430
      F1MAX=XZERO
      DO 120 J=1,M
      F1MAX=AMAX1(F1MAX,F1(J))
  120 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
      KACTX=0
      DIFFX=XZERO
```

```
      DO 150 J=1,M
      SUM=F0(J)
      DO 130 I=1,N
      SUM=SUM+DF0(J,I)*H(I)
  130 CONTINUE
      DIFF=ABS(SUM-F1PRED)
      IF (KACTX.NE.0.AND.DIFFX.LE.DIFF) GO TO 140
      KACTX=J
      DIFFX=DIFF
  140 IF (DIFF.GT.DEL*ABS(F1PRED)) GO TO 150
      KACT1=KACT1+1
      IND1(KACT1)=J
  150 CONTINUE
      IF (KACT1.NE.0) GO TO 160
      KACT1=1
      IND1(1)=KACTX
  160 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 170 I=1,N
      Y(I)=XZERO
  170 CONTINUE
      DO 190 J=1,KACT1
      JK=IND1(J)
      DO 180 I=1,N
      Y(I)=Y(I)+RLAM(J)*(DF1(JK,I)-DF0(JK,I))
  180 CONTINUE
  190 CONTINUE
C
C       ADJUST THE LOCAL BOUND RDX
C
      IF ((F0MAX-F1MAX).LE.0.25*(F0MAX-F1PRED)) RDX=RDX/XTWO
      IF ((F0MAX-F1MAX).GE.0.75*(F0MAX-F1PRED)) RDX=RDX*XTWO
C
C       TEST FOR NEWTON ITERATION
C
      IF (KACT1.GT.N1) KACT0=0
      IF (KACT1.EQ.KACT0) GO TO 210
      KACT0=KACT1
      DO 200 I=1,KACT1
      IND0(I)=IND1(I)
  200 CONTINUE
      KEQUAL=0
      GO TO 270
  210 CONTINUE
      KEQUAL=KEQUAL+1
      DO 220 I=1,KACT1
      IF ((IND1(I).EQ.IND0(I)).AND.(RLAM(I).GE.XZERO)) GO TO 220
      IND0(I)=IND1(I)
      KEQUAL=0
  220 CONTINUE
      IF (KEQUAL.LT.(KBOUND-1)) GO TO 270
C
C       FIND THE RESIDUAL-NORM OF THE SET OF NON-LINEAR EQUATIONS
C
      RES=XZERO
      DO 240 I=1,N
      S=XZERO
      DO 230 J=1,KACT1
```

```
      JK=IND1(J)
      S=S+RLAM(J)*DF1(JK,I)
  230 CONTINUE
      RES=AMAX1(RES,ABS(S))
  240 CONTINUE
      DO 250 J=1,KACT1
      JK=IND1(J)
      RES=AMAX1(RES,F1MAX-F1(JK))
  250 CONTINUE
      IF (KEQUAL.GE.KBOUND) GO TO 260
      RES0=RES
      GO TO 270
  260 IF (RES.LE.0.999*RES0) GO TO 280
  270 NEWTON=.FALSE.
      GO TO 290
  280 IF (.NOT.NEWTON) NQUAD=NQUAD+1
      NEWTON=.TRUE.
      RES0=RES
C
C         INTRODUCE THE NEW POINT IF IT IS ACCEPTABLE
C
  290 IF ((.NOT.X1OK).AND.(.NOT.NEWTON)) GO TO 330
      F0MAX=F1MAX
      X0MAX=XZERO
      DO 310 I=1,N
      X0(I)=X1(I)
      X0MAX=AMAX1(X0MAX,ABS(X0(I)))
      DO 300 J=1,M
      DF0(J,I)=DF1(J,I)
  300 CONTINUE
  310 CONTINUE
      DO 320 J=1,M
      F0(J)=F1(J)
  320 CONTINUE
  330 CONTINUE
C
C         ADJUST THE MATRIX B USING POWELL METHOD
C
C         FIND BH AND YH
C
      YH=XZERO
      DO 350 J=1,N
      YH=YH+Y(J)*H(J)
      SUMB=XZERO
      DO 340 I=1,N
      SUMB=SUMB+B(J,I)*H(I)
  340 CONTINUE
      BH(J)=SUMB
  350 CONTINUE
C
C         FIND T AND SEE IF THETA IS LESS THAN 1
C
      T=XZERO
      DO 360 I=1,N
      T=T+H(I)*BH(I)
  360 CONTINUE
      IF (YH.GE.XZERO2*T) GO TO 380
      THETA=XZERO8*T/(T-YH)
C
C         IF THETA IS TOO SMALL, MATRIX B IS NOT ALTERED
C
      IF (THETA.LT.XZERO5) GO TO 400
      S=XONE-THETA
      YH=XZERO
      DO 370 I=1,N
```

```
      Y(I)=THETA*Y(I)+S*BH(I)
      YH=YH+Y(I)*H(I)
  370 CONTINUE
  380 CONTINUE
C
C        IF T OR YH IS TOO SMALL, TEST THE STOPPING CRITERION
C
      IF (ABS(T).LE.1.E-20) GO TO 400
      IF (ABS(YH).LE.1.E-20) GO TO 400
C
C        FINALLY WE CAN CALCULATE THE NEW B
C
      DO 390 I=1,N
      S1=BH(I)/T
      S2=Y(I)/YH
      DO 390 J=I,N
      B(I,J)=B(I,J)-S1*BH(J)+S2*Y(J)
  390 B(J,I)=B(I,J)
C
  400 CONTINUE
C
C        TEST THE STOPPING CRITERION
C
      IF (NTAL.GE.MAXFUN) GO TO 420
      IF (HMAX.LE.EPS*X0MAX) GO TO 430
      IF (HMAX.LE.EPSFL*X0MAX) GO TO 410
      IF (HMAX.GT.X1M50) GO TO 50
      GO TO 430
C
  410 EPS=XZERO
      GO TO 430
  420 NTAL=NTAL+1
  430 F1(1)=NTAL
      IEQUAL=NQUAD
      RETURN
      END
C
C
      SUBROUTINE QUAD2 (N,M,DF0,F0,B,H,RLAM,N1,IND,KACT1,W,IW2,EPSFL,IE)
C
C     MINIMIZE   DELTA+HBH/2   SUBJECT TO   F0+DF0*H=DELTA    FOR THE
C     INDICES IN IND.
C
C         IND   GIVES THE INDICES CORRESPONDING TO ACTIVE FUNCTIONS;
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)
      DATA XZERO,XONE/0.0,1.0/
      NS=N+KACT1
      NTOT=NS+1
      NTOT1=NTOT+1
      EPS1=EPSFL*10*NTOT
      IE=-1
C
C        SET UP THE LINEAR SYSTEM
C        1: THE MATRIX (IT IS SYMMETRIC)
C
      DO 10 I=1,N
      W(I,NTOT)=XZERO
      W(NTOT,I)=XZERO
      DO 10 J=1,I
      W(I,J)=B(I,J)
```

```
      W(J,I)=W(I,J)
   10 CONTINUE
      DO 30 J=N1,NS
      W(J,NTOT)=-XONE
      W(NTOT,J)=-XONE
      DO 20 I=N1,J
      W(I,J)=XZERO
      W(J,I)=XZERO
   20 CONTINUE
      DO 30 I=1,N
      JA=IND(J-N)
      W(J,I)=DF0(JA,I)
      W(I,J)=W(J,I)
   30 CONTINUE
      W(NTOT,NTOT)=XZERO
C
C       2: RIGHT HAND SIDE
C
      DO 40 I=1,N
      W(I,NTOT1)=XZERO
   40 CONTINUE
      DO 50 J=N1,NS
      JA=IND(J-N)
      W(J,NTOT1)=-F0(JA)
   50 CONTINUE
      W(NTOT,NTOT1)=-XONE
C
C       SOLVE THE LINEAR SYSTEM
C
      CALL GAUSS (W,IW2,NTOT,NTOT1,EPS1)
      IF (EPS1.LE.XZERO) GO TO 80
C
C       STORE THE SOLUTION IN H AND RLAM
C
      DO 60 I=1,N
      H(I)=W(I,NTOT1)
   60 CONTINUE
      H(N1)=W(NTOT,NTOT1)
      DO 70 J=1,KACT1
      RLAM(J)=W(N+J,NTOT1)
   70 CONTINUE
      IE=0
   80 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
      DATA XZERO,XONE/0.0,1.0/
      K1=KACT1+1
      K2=KACT1+2
      EPS1=EPSFL*10*K1
      DO 40 I=1,KACT1
      IK=IND1(I)
      DO 30 J=1,I
      JK=IND1(J)
      S=XZERO
      DO 20 L=1,N
      IF (X1OK) GO TO 10
      S=S+DF0(IK,L)*DF0(JK,L)
```

```
      GO TO 20
   10 S=S+DF1(IK,L)*DF1(JK,L)
   20 CONTINUE
      W(I,J)=S
      W(J,I)=S
   30 CONTINUE
      W(I,K1)=-XONE
      W(K1,I)=XONE
      W(I,K2)=XZERO
   40 CONTINUE
      W(K1,K1)=XZERO
      W(K1,K2)=XONE
      CALL GAUSS (W,IW2,K1,K2,EPS1)
      DO 50 I=1,KACT1
      RLAM(I)=W(I,K2)
   50 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE FPOS (FQQ,FHH,FDF,NC,NS,N,M,X,DF,F)
      EXTERNAL FHH
      DIMENSION X(N), DF(M,N), F(M)
      DATA XZERO/0.0/
      CALL FDF (N,M,X,DF,F)
      CALL FQQ (FHH,N,M,X,DF,F,NC,NS)
      DO 20 J=1,M
      IF (F(J).GE.XZERO) GO TO 20
      F(J)=-F(J)
      DO 10 I=1,N
      DF(J,I)=-DF(J,I)
   10 CONTINUE
   20 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)
      DATA XONE/1.0/
      N1=N+1
      IF (N.EQ.1) GO TO 100
C
C       EQUILIBRATION
C
      DO 30 I=1,N
      C=ABS(A(I,1))
      DO 10 J=2,N
      IF (ABS(A(I,J)).GT.C) C=ABS(A(I,J))
   10 CONTINUE
      DO 20 J=1,M
      A(I,J)=A(I,J)/C
   20 CONTINUE
   30 CONTINUE
C
C       PIVOTING AND REDUCTION TO TRIANGULAR FORM
C
      NM=N-1
      DO 90 K=1,NM
```

```
      K1=K+1
      IPIV=K
      C=ABS(A(K,K))
      DO 40 I=K1,N
      IF (C.GE.ABS(A(I,K))) GO TO 40
      IPIV=I
      C=ABS(A(I,K))
   40 CONTINUE
C
C        TEST FOR SINGULARITY
C
      IF (C.LT.EPS) GO TO 150
C
C        PIVOTING CONTINUED
C
      IF (IPIV.EQ.K) GO TO 60
      DO 50 J=K,M
      C=A(K,J)
      A(K,J)=A(IPIV,J)
      A(IPIV,J)=C
   50 CONTINUE
   60 CONTINUE
      DO 80 I=K1,N
      C=A(I,K)/A(K,K)
      DO 70 J=K1,M
      A(I,J)=A(I,J)-C*A(K,J)
   70 CONTINUE
   80 CONTINUE
   90 CONTINUE
C
C        END OF REDUCTION
C
  100 CONTINUE
C
C        TEST FOR SINGULARITY
C
      IF (ABS(A(N,N)).LT.EPS) GO TO 150
C
C        BACKSUBSTITUTION
C
      DO 140 II=1,N
      I=N-II+1
      DO 130 J=N1,M
      C=A(I,J)
      IF (I.EQ.N) GO TO 120
      I1=I+1
      DO 110 K=I1,N
      C=C-A(I,K)*A(K,J)
  110 CONTINUE
  120 A(I,J)=C/A(I,I)
  130 CONTINUE
  140 CONTINUE
      GO TO 160
  150 EPS=-XONE
  160 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
      DATA XZERO,XONE,XTWO,XFOUR,XFIVE/0.0,1.0,2.0,4.0,5.0/
      IF ((DX.LT.XZERO).OR.(EPSH.LT.XZERO)) 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=XONE/(XFOUR*N)
      NTAL=0
C
C       FIND EQUATION I0  WHICH GOES INTO THE FIRST REFERENCE
C
      C=-XONE
      DO 10 J=1,M
      IREF(LREF+J)=0
      IF (ABS(B(J)).LT.C) GO TO 10
      C=ABS(B(J))
      I0=J
   10 CONTINUE
C
C       INITIALIZE REFERENCE ARRAYS
C
      S=XZERO
      T=B(I0)
      XM=M
      DO 20 I=1,N
      D=A(I0,I)
      S=S+ABS(D)
      XM=XM+XONE
      IF (D.EQ.XZERO) D=-XONE
      IREF(I)=SIGN(XM,-D*T)
      IREF(LBND+I)=IREF(I)
   20 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 30
      GAM=DX
      DG=GAM
      DH=C-DX*S
      GO TO 40
   30 DC=C/S
      DH=XZERO
      GAM=DG
```

```
C
C        FIND VECTOR X
C
   40 DO 50 I=1,N
      XM=IREF(I)
      X(I)=SIGN(DG,XM)
   50 CONTINUE
C
C        FIND MATRIX H
C
      S=XONE/(S+XONE)
      H(N1,N1)=S
      IREF(NN2+N1)=0
      DO 80 I=1,N
      XM=-IREF(I)
      H(N1,I)=SIGN(S,XM)
      T=ABS(A(I0,I))*S
      H(I,N1)=T
      DO 60 J=1,N
      XM=-IREF(J)
      H(I,J)=SIGN(T,XM)
   60 CONTINUE
      IF (T.GT.XZERO) GO TO 70
      IREF(NN2+I)=1
      H(I,N2)=XONE
      GO TO 80
   70 IREF(NN2+I)=0
   80 H(I,I)=ISIGN(1,IREF(I))+H(I,I)
C
C        INITIALIZE SOME CONSTANTS
C
      RSIG=XONE-S
      DGH=DG
      DH1=DH
      ETA=GAM
      ERRX=XZERO
      GAMCH=.TRUE.
      NBIN=N
      GO TO 650
C
C        ITERATIVE LOOP STARTS HERE
C        FIND VECTOR RHO
C
   90 DGH=DG
      IF (I0S.LT.0) GO TO 120
      DO 110 I=1,N1
      S=-H(I,N1)
      DO 100 J=1,N
      S=S-H(I,J)*A(I0,J)
  100 CONTINUE
      RHO(I)=S
  110 CONTINUE
      GO TO 190
  120 DO 140 I=1,N1
      S=-H(I,N1)
      DO 130 J=1,N
      S=S+H(I,J)*A(I0,J)
  130 CONTINUE
      RHO(I)=S
  140 CONTINUE
      GO TO 190
C
C        BOUNDS VIOLATED
C
  150 I0=M+J0
```

```
      DHH=DH
      I0S=SIGN(XONE,X(J0))
      IF (I0S.LT.0) GO TO 170
      DO 160 I=1,N1
      RHO(I)=-H(I,J0)-H(I,N1)
  160 CONTINUE
      GO TO 190
  170 DO 180 I=1,N1
      RHO(I)=H(I,J0)-H(I,N1)
  180 CONTINUE
C
C        FIND EQUATION L0 WHICH LEAVES THE REFERENCE
C        FIND -H(I,N1)/RHO(I)  FOR NEGATIVE VALUES OF RHO(I)
C
  190 LB=0
      RTAU=XZERO
      IF (I0.GT.M) RTAU=XONE
      DO 210 I=1,N1
      IF (RHO(I).GE.XZERO) GO TO 200
      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)
  200 IF (IABS(IREF(I)).LE.M) GO TO 210
      RTAU=RTAU+RHO(I)
  210 CONTINUE
C
C        FIND THE COEFFICIENTS IN THE RATIONAL EXPRESSION
C        (TT+LAM*SS)/(RSIG+LAM*RTAU)
C
      DG2=DG*XTWO
      IF (DH.GT.XZERO) GO TO 220
      TT=DG*RSIG
      SS=DG*RTAU+DHH+DGH-DG
      NL=1
      GO TO 230
  220 RSIG=XONE-RSIG
      RTAU=-RTAU
      TT=DH*RSIG
      SS=DH*RTAU+DHH-DH+DGH-DG
      NL=2
  230 SMAX=XZERO
      L=1
  240 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 260
      L=LA+1
      LM=LA
      DO 250 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 250
      L0=IN
      LAM=S
      ILAM=IS
      LM=I
  250 CONTINUE
C
```

```
C       REORDER
C
      IREF(N1+LM)=IREF(N1+LA)
      IREF(N1+LA)=L0
C
C       FIND  MAX(-RHO(I))
C
  260 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 (290,280,280,270), ML
  270 TT=TT+DG2*H(L0,N1)
      SS=SS+DG2*RHO(L0)
      GO TO 290
  280 RSIG=RSIG-XTWO*H(L0,N1)
      RTAU=RTAU-XTWO*RHO(L0)
  290 IF ((RSIG*SS.GT.RTAU*TT).AND.(LA.LT.LB)) GO TO 240
C
C       TEST IF  -RHO(L0)  IS TOO SMALL
C
      SMAX=SMAX/XFOUR
  300 IF (-RHO(L0).GE.SMAX) GO TO 310
      LA=LA-1
      L0=IREF(N1+LA)
      GO TO 300
C
C       UPDATE REFERENCE ARRAYS
C
  310 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 320 J=1,N1
      H(L0,J)=-H(L0,J)/RHOL0
  320 CONTINUE
      DO 340 I=1,N1
      IF (I.EQ.L0) GO TO 340
      S=RHO(I)
      DO 330 J=1,N1
      H(I,J)=S*H(L0,J)+H(I,J)
  330 CONTINUE
  340 CONTINUE
C
C       IF ANY SIGNS HAVE BEEN CHANGED, UPDATE H
C
      RHON1=XONE
      DO 350 I=1,N1
      RHO(N1+I)=XONE
  350 CONTINUE
      IF (LA.LE.1) GO TO 460
      K=LA-1
      DO 360 I=1,N1
      RHO(NN2+I)=XZERO
  360 CONTINUE
      DO 370 I=1,K
      RHO(IREF(N1+I)+N1)=-XONE
  370 CONTINUE
```

```
      DO 410 I=1,N1
      IF (RHO(N1+I).GT.XZERO) GO TO 390
      DO 380 J=1,N1
      RHO(NN2+J)=RHO(NN2+J)-H(I,J)
  380 CONTINUE
      GO TO 410
  390 DO 400 J=1,N1
      RHO(NN2+J)=RHO(NN2+J)+H(I,J)
  400 CONTINUE
  410 CONTINUE
      RHON1=RHO(NN2+N1)
      DO 420 J=1,N1
      H(J,N1)=H(J,N1)/RHON1
  420 CONTINUE
      DO 440 I=1,N
      S=RHO(NN2+I)
      DO 430 J=1,N1
      H(J,I)=-S*H(J,N1)+H(J,I)
  430 CONTINUE
  440 CONTINUE
C
C        CHANGE SIGNS IN SOME ROWS OF MATRIX H
C
      DO 450 L=1,K
      I=IREF(N1+L)
      J=-IREF(I)
      IREF(I)=J
      IREF(LREF+IABS(J))=J
      DO 450 J=1,N1
  450 H(I,J)=-H(I,J)
C
C        UPDATE THE LAST COLUMN OF H IN CASE OF DEGENERACIES
C
  460 IF (IREF(NN2+L0).EQ.0) GO TO 520
      H(L0,N2)=-H(L0,N2)/(RHO(L0)*RHON1)
      DO 510 I=1,N1
      IF ((IREF(NN2+I).EQ.0).OR.(I.EQ.L0)) GO TO 510
      IF (IREF(NN2+I)-IREF(NN2+L0)) 470,490,480
  470 H(I,N2)=H(I,N2)/RHON1
      GO TO 510
  480 H(I,N2)=ABS(H(L0,N2)*RHO(I))
      IREF(NN2+I)=IREF(NN2+L0)
      GO TO 510
  490 C=H(I,N2)/RHON1+H(L0,N2)*RHO(I)
      IF ((LA.GT.1).AND.(RHO(N1+I).LT.XZERO)) C=-C
      IF (C.GT.XZERO) GO TO 500
      H(I,N2)=XONE
      IREF(NN2+I)=IREF(NN2+I)+1
      GO TO 510
  500 H(I,N2)=C
  510 CONTINUE
      GO TO 540
C
C        TEST FOR DEGENERACIES
C
  520 DO 530 I=1,N1
      IREF(NN2+I)=0
      IF (H(I,N1).GT.XZERO) GO TO 530
      IREF(NN2+I)=1
      H(I,N1)=XZERO
      H(I,N2)=XONE
  530 CONTINUE
C
C        UPDATE GAM
C
```

```
  540 GAMCH=.FALSE.
      IF ((NBIN.EQ.0).OR.(GAMM.LT.XTWO*GAM.AND.GAMM.LT.DX)) GO TO 550
      IF (GAM.LT.GAMM) GAMCH=.TRUE.
      GAM=GAMM
C
C        UPDATE DH AND DG
C
  550 S=XZERO
      RSIG=XZERO
      DO 570 I=1,N1
      K=IABS(IREF(I))
      IF (K.GT.M) GO TO 560
      RHO(I)=B(K)*ISIGN(1,IREF(I))
      S=S+H(I,N1)*RHO(I)
      GO TO 570
  560 RSIG=RSIG+H(I,N1)
  570 CONTINUE
      IF (RSIG.NE.XZERO) GO TO 580
      DH=ABS(S)
      DG=GAM
      GO TO 590
  580 DG=AMIN1(GAM,ABS(S)/RSIG)
      DH=XZERO
      IF (DG.EQ.GAM) DH=ABS(S-DG*RSIG)/(XONE-RSIG)
C
C        CALCULATE PARAMETER VALUES
C
  590 DGH=XZERO
      ERRX=XZERO
      DO 610 I=1,N1
      IF (IABS(IREF(I)).GT.M) GO TO 600
      RHO(I)=DH-RHO(I)
      GO TO 610
  600 RHO(I)=DG
  610 CONTINUE
      DO 640 I=1,N
      S=H(1,I)*RHO(1)
      DO 620 J=2,N1
      S=S+H(J,I)*RHO(J)
  620 CONTINUE
      IF (IREF(LBND+I).EQ.0) GO TO 630
      T=S
      XM=IREF(LBND+I)
      S=SIGN(DG,XM)
      ERRX=AMAX1(ERRX,ABS(S-T))
  630 X(I)=S
      IF (ABS(S).LE.DGH) GO TO 640
      DGH=ABS(S)
      J0=I
  640 CONTINUE
      NTAL=NTAL+1
C
C        CALCULATE GAMM
C
  650 GAMM=AMIN1(AMAX1(XFIVE*DG,GAM),DX)
C
C        FIND EQUATION I0 WHICH GOES INTO THE REFERENCE
C
      DHH=XZERO
      T=DH
      T1=DH
      DO 680 I=1,M
      S=B(I)
      DO 660 J=1,N
      S=S+A(I,J)*X(J)
```

```
  660 CONTINUE
      RHO(I)=S
      IF (IREF(LREF+I).NE.0) GO TO 670
      IF (ABS(S).LE.DHH) GO TO 680
      DHH=ABS(S)
      I0S=SIGN(XONE,S)
      I0=I
      GO TO 680
  670 T=AMAX1(T,ABS(S))
      T1=AMIN1(T1,ABS(S))
  680 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 150
C
C        TEST OF CONVERGENCE CRITERION
C
      IF (DHH.GT.DH1) GO TO 90
      IF ((GAM.LT.DX).AND.(DHH+DH.GT.(DH1-DH)*XTWO).AND.(NBIN.GT.0)) GO
     1TO 540
      RHO(M1)=DH
      IREF(N2)=NBIN
      IREF(N+3)=NTAL
      RETURN
      END
```

SOC-291

MMUM - A FORTRAN PACKAGE FOR UNCONSTRAINED MINIMAX OPTIMIZATION

J.W. Bandler and W.M. Zuberek

May 1982, No. of Pages: 74

Revised:

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

Abstract: MMUM is a package of subroutines for solving unconstrained minimax optimization problems. It is an extension and modification of the MINI5W package due to Madsen. First derivatives of all functions with respect to all variables are assumed to be known. The solution is found by an iteration that uses either linear programming applied in connection with first-order derivatives or a quasi-Newton method applied in connection with first-order and approximate second-order derivatives. The method has been described by Hald and Madsen. The package and documentation are developed for the CDC 170/730 system with the NOS 1.4 operating system and the Fortran 4.8508 compiler.

Description: Contains Fortran listing, user's manual. Source deck or magnetic tape available for $150.00. The listing contains 1397 lines, of which 342 are comments.

Related Work: SOC-218, SOC-280, SOC-281, SOC-292, SOC-294.

Price: $100.00.