

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 chosen) 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  $k$ th iteration the change  $\tilde{h}^k$  of the approximation  $\tilde{x}^{k-1}$  is determined as the solution of the linear minimax problem

$$\text{Minimize}_{\tilde{h}^k} \tilde{F}(\tilde{x}^{k-1}, \tilde{h}^k) = \max_{1 \leq i \leq m} |\tilde{f}_i(\tilde{x}^{k-1}) + \tilde{f}'_i(\tilde{x}^{k-1}) \tilde{h}^k|$$

subject to constraint

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

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

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

it is successful if

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

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  $\delta_x^0$  is defined by the user (argument DX). If the objective function  $F$  decreases, then  $x^k = \tilde{x}^{k-1} + h^k$ , otherwise  $x^k = \tilde{x}^{k-1}$ . There is no line search.

Switch to Stage 2

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

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

where  $\delta$  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  $\lambda_j^k \geq 0$ ,  $j \in A^k$ , such that

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

and, for  $k > 2$ ,

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

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

$$r(\underline{x}, \lambda, A) = \max \left( \max_{1 \leq i \leq n} \left| \sum_{j \in A} \lambda_j^k \bar{f}'_{ji}(\underline{x}) \right|, \max_{j \in A} (F_j(\underline{x}) - \bar{f}_j(\underline{x})) \right)$$

and

$$\bar{f}'_{ji} = \begin{cases} \frac{\partial f_j}{\partial x_i} & \text{if } f_j > 0 \\ -\frac{\partial f_j}{\partial x_i} & \text{otherwise.} \end{cases}$$

### Stage 2

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

$$\sum_{j \in A^k} \lambda_j^k \bar{f}'_{ji}(\underline{x}^k) = 0, \quad i = 1, \dots, n$$

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

$$\bar{f}_{j_0}(\underline{x}^k) - \bar{f}_j(\underline{x}^k) = 0, \quad j \in A^k, \quad j_0 \in A^k, \quad j \neq j_0,$$

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

### Switch to Stage 1

At each  $k$ th 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  $\lambda_j^k$  are nonnegative

$$\lambda_j^k \geq 0, \quad j \in A^k,$$

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

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

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

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

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  $\tilde{h}^k$  of the approximation  $\tilde{x}^k$  of the solution is sufficiently small

$$\|\tilde{h}^k\| \leq \epsilon \|\tilde{x}^k\|,$$

where  $\epsilon$  is defined by the user (argument EPS),

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

$$\|\tilde{h}^k\| \leq \epsilon_0 \|\tilde{x}^k\|,$$

where  $\epsilon_0$  is the smallest positive number such that

$$1 + \epsilon_0 > 1,$$

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

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

(when the solution  $\tilde{x}^*$  is equal to 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,

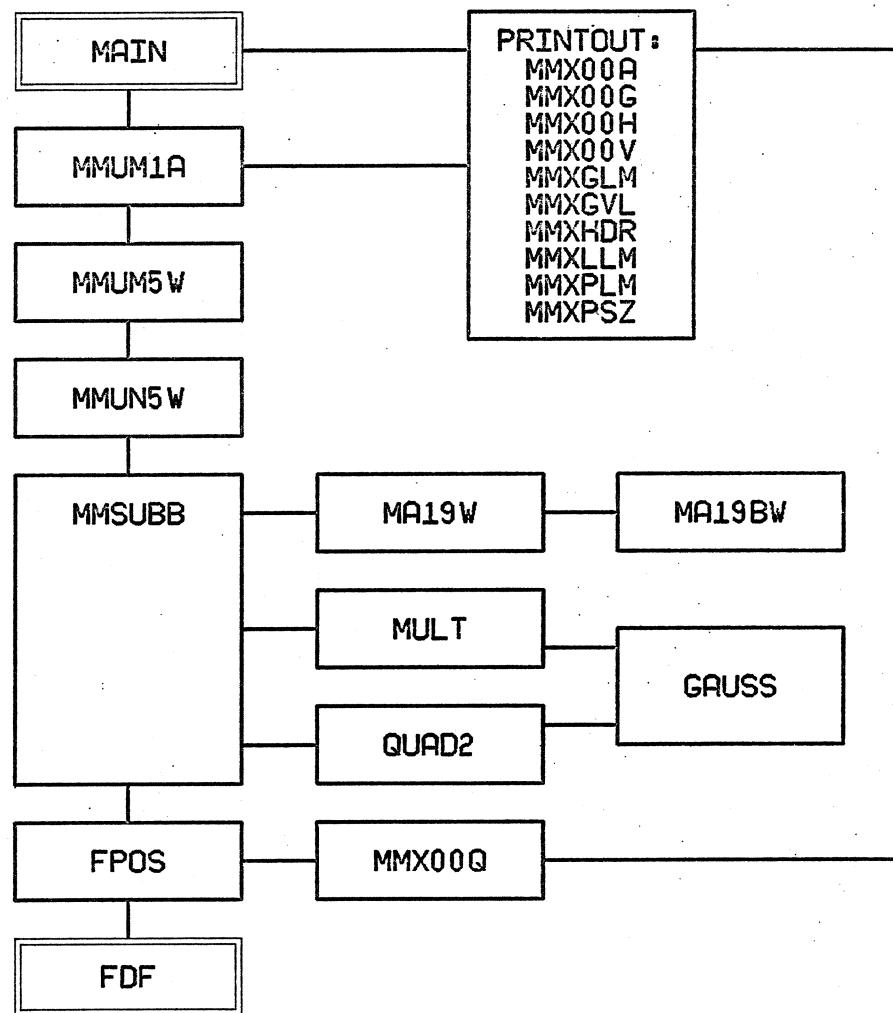


Fig. 1 Structure of the MMUM package corresponding to the standard entry (subroutine MMUM1A).

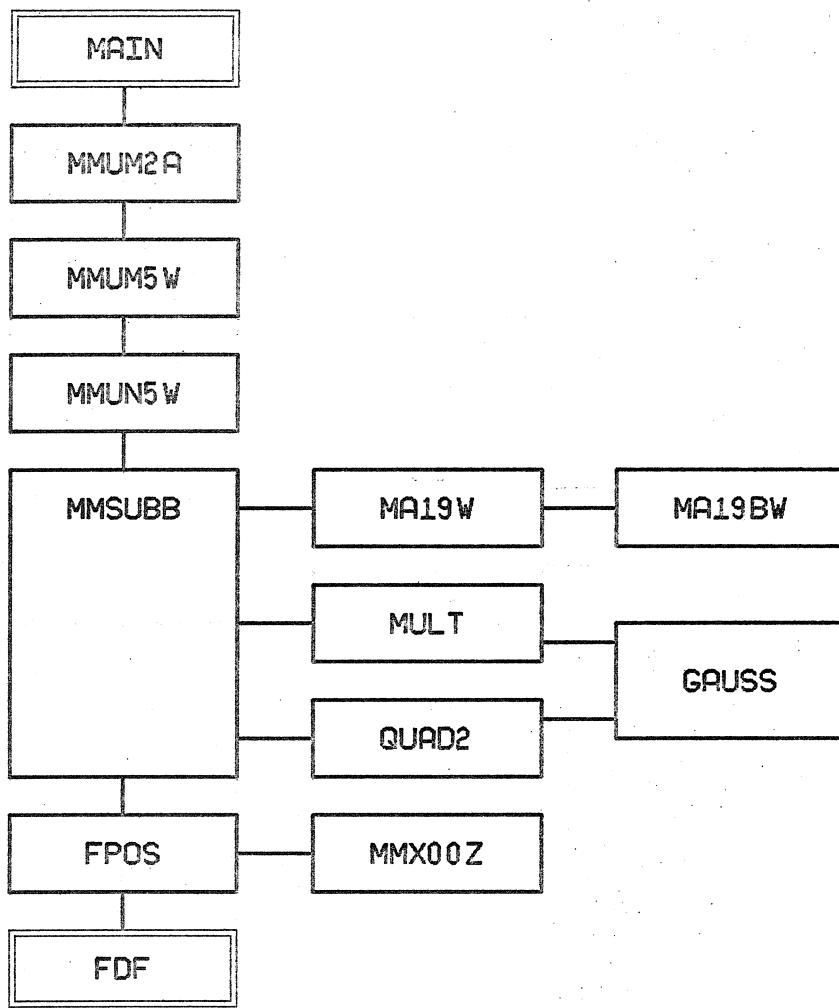


Fig. 2 Structure of the MMUM package corresponding to the basic entry (subroutine MMUM2A).

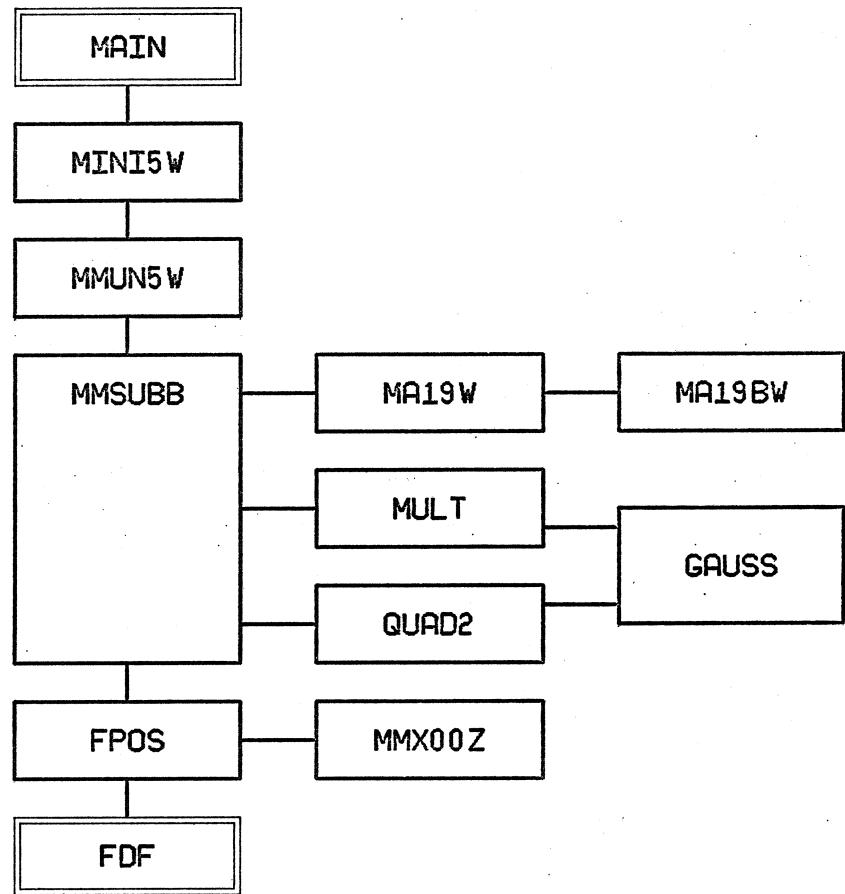


Fig. 3 Structure of the MMUM package corresponding to the original entry (subroutine **MINI5W**).

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 MMX00Q 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 MMUN5W 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  $f_i(x)$  and their derivatives  $\partial f_i(x)/\partial x_j$  at the point  $x$  corresponding to  $X(1), X(2), \dots, X(N)$ , and store the values in the following way:

$$F(I) = f_I(x), \quad I=1, \dots, M,$$
$$DF(I,J) = \partial f_I(x)/\partial x_J, \quad I=1, \dots, M, \quad J=1, \dots, N.$$

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,  $X(I)=x_I^0$ ,  $I=1, \dots, N$ . 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  $\{x | \|x-x^0\| < DX\}$  the residual functions  $f_i(x)$  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  $x^0$  and the solution, but if more curvature is present this value may be too large. Normally  $DX=0.1*\|x^0\|$  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  $\delta_x^k$ .

EPS is a REAL variable which on entry must be set to the required accuracy of the solution. The iteration terminates when  $\|h^k\| \leq EPS*\|x^k\|$ , where  $h^k$  is the correction to the kth approximation  $x^k$  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.,  $W(I)=f_I(x)$ ,  $I=1,\dots,M$ .

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	1753	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 MMJM PACKAGE

M:	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	2110	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

$$|IPR| = pqr s$$

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.,  $q=1, \dots, 9$ ) 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  $\underline{x}^0$  and corresponding residual function values  $f(\underline{x}^0)$  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 $\geq$ 0, MAXF>0, KEQS>0, IW $\geq$ IWR.

Date: March 1982.

## VII. EXAMPLES

### Example 1 [1, Example 1]

Minimize

$$F(x) = \max (|f_1(x)|, |f_2(x)|),$$

where

$$f_1(x) = x_1^2 + 2x_2^2 + x_1x_2,$$
$$f_2(x) = \sin(x_1) + \cos(x_2).$$

The starting point is

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

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  $\text{EPS} = 10^{-6}$ , as well as the numbers of shifts to Stage 2 are summarized in the following table (the numbers of shifts are given in parentheses):

DX	KEQS		
	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)          000001  
C                                              000002  
C K. MADSEN EXAMPLE                         000003  
C                                              000004  
C                                              000005  
C                                              000006  
C                                              000007  
C                                              000008  
C                                              000009  
C                                              000010  
C                                              000011  
C                                              000012  
C                                              000013  
C                                              000014  
C                                              000015  
C                                              000016  
C                                              000017  
C                                              000018  
C                                              000019  
C                                              000020  
C                                              000021  
C                                              000022  
C                                              000023  
C                                              000024  
C                                              000025  
C                                              000026  
C                                              000027  
C                                              000028  
C                                              000029  
C                                              000030  
C                                              000031  
C                                              000032  
C                                              000033  
C                                              000034  
C                                              000035  
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(IWS)	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

Example 2 [3, Example 1]

Minimize

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

where

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

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

The starting point is

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

The solution is

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

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

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

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  $f(\underline{x})$  is equal to 0, and the minimization of  $f(\underline{x})$  is thus equivalent to solving the nonlinear equation

$$f(\underline{x}) = 0.$$

However,  $f(\underline{x})$  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  $f(\underline{x})$  is equivalent to minimization of

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

or the minimization of

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

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:

DX	KEQS				
	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) 000001  
C 000002  
C HALD-MADSEN EXAMPLE 1 000003  
C 000004  
DIMENSION X(2),T(3),W(98) 000005  
EXTERNAL FDF 000006  
DATA T/10HTRMMU2 : H,10HALD-MADSEN,10H EXAMPLE 1/ 000007  
CALL MMXHDR(3,T) 000008  
N=2 000009  
M=2 000010  
DXX=0.6 000011  
DO 10 I=1,2 000012  
X(1)=-1.2 000013  
X(2)=1.0 000014  
DX=DXX 000015  
EPS=1.E-6 000016  
MAXF=50 000017  
KEQS=2 000018  
IW=98 000019  
ICH=5 000020  
IPR=-10 000021  
CALL MMUM1A(FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL) 000022  
DXX=DXX-1.E-10 000023  
10 CONTINUE 000024  
STOP 000025  
END 000026  
C 000027  
C SUBROUTINE FDF(N,M,X,DF,F)  
DIMENSION X(N),DF(M,N),F(M)  
X1=X(1) 000029  
X2=X(2) 000030  
F(1)=10.0\*(X2-X1\*X1) 000031  
F(2)=1.0-X1 000032  
DF(1,1)=-20.0\*X1 000033  
DF(1,2)=10.0 000034  
DF(2,1)=-1.0 000035  
DF(2,2)=0.0 000036  
RETURN 000037  
END 000038  
000039  
000040

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 (MFC)	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 (DX0)	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	2	-4.400000000000E+00
2	1.000000000000E+00	2	2.200000000000E+00

**VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.**

**SOLUTION**

	VARIABLES		FUNCTION VALUES
1	1.000000000000E+00	2	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

Example 3 [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

$$\begin{aligned} 4(x_1+x_2) &= 0, \\ (x_1-x_2)((x_1-2)^2 + x_2^2) + 3x_1 + 5x_2 &= 0. \end{aligned}$$

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

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

where

$$\begin{aligned} f_1(\underline{x}) &= 4(x_1+x_2), \\ f_2(\underline{x}) &= (x_1-x_2)((x_1-2)^2 + x_2^2) + 3x_1 + 5x_2. \end{aligned}$$

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

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

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

PROGRAM TRMMU3(OUTPUT,TAPE6=OUTPUT) 000001  
C 000002  
C BRENT EXAMPLE 000003  
C 000004  
DIMENSION X(2),XX(4,2),T(3),W(98) 000005  
EXTERNAL FDF 000006  
DATA XX/2.0,-2.0,2.0,2.0, 000007  
1 2.0,-2.0,0.0,1.0/ 000008  
DATA T/10HTRMMU3 : B,10HRENT EXAMP,10HLE 000009  
CALL MMXHDR(3,T) 000010  
N=2 000011  
M=2 000012  
IPR=-10 000013  
DO 20 I=1,4 000014  
X(1)=XX(I,1) 000015  
X(2)=XX(I,2) 000016  
DX=.2 000017  
EPS=1.E-6 000018  
MAXF=50 000019  
KEQS=2 000020  
IW=98 000021  
ICH=6 000022  
CALL MMUM1A(FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL) 000023  
IPR=0 000024  
20 CONTINUE 000025  
STOP 000026  
END 000027  
C 000028  
SUBROUTINE FDF(N,M,X,DF,F) 000029  
DIMENSION X(N),DF(M,N),F(M) 000030  
X1=X(1) 000031  
X2=X(2) 000032  
R1=X1-X2 000033  
R2=(X1-2.0)\*\*2+X2\*X2 000034  
F(1)=4.0\*(X1+X2) 000035  
F(2)=R1\*R2+3.0\*X1+5.0\*X2 000036  
DF(1,1)=4.0 000037  
DF(1,2)=4.0 000038  
DF(2,1)=R2+(R1+R1)\*(X1-2.0)+3.0 000039  
DF(2,2)=-R2+R1\*(X2+X2)+5.0 000040  
RETURN 000041  
END 000042  
000043

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 (NO)	2
NUMBER OF FUNCTIONS (MOC)	2
STEP LENGTH (DX0)	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 (MF)	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;  
UNCONSTRAINED MINIMAX OPTIMIZATION (MMUM PACKAGE)

PAGE : 1  
(V:82.03)

TRMMU3 : BRENT EXAMPLE

**INPUT DATA**

NUMBER OF VARIABLES (ND)	2
NUMBER OF FUNCTIONS (MD)	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	2.00000000000000E+00		1	8.00000000000000E+00
2	0.		2	6.00000000000000E+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:  
UNCONSTRAINED MINIMAX OPTIMIZATION (MMUM PACKAGE)

PAGE : 1  
(V:82.03)

TRMMU3 : BRENT EXAMPLE

**INPUT DATA**

NUMBER OF VARIABLES (N)	2
NUMBER OF FUNCTIONS (MF)	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

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

subject to constraints

$$\begin{aligned} -x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8 &\geq 0, \\ -x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10 &\geq 0, \\ -x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5 &\geq 0. \end{aligned}$$

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

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

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  $f(\underline{x}^*)$  cannot be obtained. Therefore, instead of the function  $f(\underline{x})$ , the function  $f_1(\underline{x}) = f(\underline{x}) + c$  can be used where  $c$  is a positive constant which is equal to at least  $f(\underline{x}^*)$ ;  $c=100$  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

$$\begin{aligned} f_2(\underline{x}) &= f_1(\underline{x}) - \alpha(-x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8), \\ f_3(\underline{x}) &= f_1(\underline{x}) - \alpha(-x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10), \\ f_4(\underline{x}) &= f_1(\underline{x}) - \alpha(-x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5), \end{aligned}$$

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

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

Finally, because the package minimizes the absolute values of residual functions  $f_i(\underline{x})$ , at the solution  $\underline{x}^*$  the absolute values of transformed

constraints  $|f_2(\tilde{x}^*)|$ ,  $|f_3(\tilde{x}^*)|$  and  $|f_4(\tilde{x}^*)|$  must not be greater than  $|f_1(\tilde{x}^*)|$ , otherwise the solution found by the package will be incorrect (if this condition is not satisfied the constant  $c$  in  $f_1(x)$  should be increased).

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

PROGRAM TRMMU4(OUTPUT,TAPE6=OUTPUT) 000001  
C 000002  
C ROSEN-SUZUKI PROBLEM 000003  
C 000004  
DIMENSION X(4),W(234) 000005  
EXTERNAL FDF 000006  
COMMON NCALL 000007  
CALL DATE(DAT) 000008  
CALL TIME(TIM) 000009  
N=4 000010  
M=4 000011  
X(1)=2.0 000012  
X(2)=2.0 000013  
X(3)=5.0 000014  
X(4)=0.0 000015  
DO 20 IX=1,2 000016  
NCALL=0 000017  
DX=0.5 000018  
EPS=1.E-6 000019  
MAXF=30 000020  
KEQS=2 000021  
IW=234 000022  
WRITE(6,111) DAT,TIM,N,M,MAXF,KEQS,DX,EPS 000023  
111 FORMAT(1H1/" PROGRAM : TRMMU4 - DATE :",A9," TIME :",A10//  
1 " ROSEN-SUZUKI PROBLEM"// 000024  
2 " NUMBER OF VARIABLES:",I4,10X,"NUMBER OF FUNCTIONS:",I4/ 000025  
3 " MAXF VALUE:",I4,10X,"KEQS VALUE:",I4/ 000026  
4 " DX VALUE:",E10.4,10X,"EPS VALUE:",E10.4/) 000027  
WRITE(6,222) (I,X(I),I=1,N) 000028  
222 FORMAT("// STARTING POINT:"/(20X,I2,2X,F12.8)) 000029  
CALL SECOND(TIME1) 000030  
CALL MMUM2A(FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,IFALL) 000031  
CALL SECOND(TIME2) 000032  
PTIME=TIME2-TIME1 000033  
WRITE(6,333) (I,X(I),I=1,N) 000034  
333 FORMAT("// SOLUTION:"/(20X,I2,2X,F12.8)) 000035  
WRITE(6,444) (I,W(I),I=1,M) 000036  
444 FORMAT("// FUNCTION VALUES:"/(20X,I2,2X,F12.8)) 000037  
DO 10 I=2,M 000038  
10 W(I)=0.1\*(W(I)-W(I-1)) 000039  
WRITE(6,555) (I,W(I),I=2,M) 000040  
555 FORMAT("// CONSTRAINTS:"/(20X,I2,2X,E12.5)) 000041  
WRITE(6,666) IFALL,MAXF,KEQS,PTIME 000042  
666 FORMAT("// TYPE OF SOLUTION (IFALL):",I4/ 000043  
1 " NUMBER OF FUNCTION EVALUATIONS:",I4/ 000044  
2 " NUMBER OF SHIFTS TO STAGE-2:",I4/ 000045  
3 " EXECUTION TIME (IN SECONDS):",F7.3/) 000046  
DO 15 J=1,4 000047  
15 X(J)=0.0 000048  
20 CONTINUE 000049  
STOP 000050  
END 000051  
C 000052  
C 000053

C SUBROUTINE FDF(N,M,X,DF,F)  
DIMENSION X(N),DF(M,N),F(MD)  
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

000054  
000055  
000056  
000057  
000058  
000059  
000060  
000061  
000062  
000063  
000064  
000065  
000066  
000067  
000068  
000069  
000070  
000071  
000072  
000073  
000074  
000075  
000076  
000077  
000078  
000079  
000080  
000081  
000082  
000083  
000084  
000085  
000086  
000087  
000088  
000089  
000090  
000091  
000092  
000093  
000094  
000095  
000096  
000097  
000098  
000099  
000100  
000101  
000102  
000103

PROGRAM : TRMMU4N - DATE : 82/04/22 TIME : 15:12:21.16  
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	5.00000005	0.00000000	333.00000000
2	1.50000000	1.50000000	4.50000004	.50000000	249.25000000
3	.50000000	.50000000	3.50000008	.25000000	128.93750000
4	1.57421875	2.50000000	1.61104911	-1.75000000	189.27274961
5	-.16666667	1.50000000	2.50000002	-.75000000	79.11805556
6	.83333333	.50000000	1.859953701	-1.75000000	87.04669817
7	.20833333	1.00000000	2.00000002	-1.18055556	62.39158951
8	-.29166667	.50000000	2.379700062	-.72946363	64.54807860
9	-.04166667	.75000000	2.111265312	-.94903600	57.76811046
10	.00340028	1.03032112	2.014010542	-.99577838	56.65489063
11	.00087176	.99953710	1.999836091	-1.00052898	56.00829245
12	-.00008510	1.00006030	2.000043862	-.99993569	56.00001186
13	.00000057	1.00000047	1.999999461	-1.00000055	56.00000013
14	-.00000001	.99999997	2.000000012	-.99999999	56.00000000

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 (IF ALL): 0  
NUMBER OF FUNCTION EVALUATIONS: 14  
NUMBER OF SHIFTS TO STAGE-2: 1  
EXECUTION TIME (IN SECONDS): .1858

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	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	100.0000000	1
2	.50000000	.50000000	.50000000	-.50000000	82.2500000	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	2.001887482	-.1.00068031	56.08312609	2
13	-.00008655	.99997837	2.000071332	-.99992134	56.00008885	2
14	.00000067	1.00000561	1.999997921	-.1.00000137	56.00000015	2
15	.00000063	.99999790	2.000000152	-.1.00000024	56.00000000	2
16	-.00000004	1.00000001	2.000000032	-.99999997	56.00000000	2
17	.00000000	1.00000000	2.000000002	-.1.00000000	56.00000000	1

SOLUTION:

1	.00000000
2	1.00000001
3	2.00000000
4	-1.00000001

FUNCTION VALUES:

1	56.00000000
2	56.00000000
3	46.00000036
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): .2088

Example 5

Minimize the Beale constrained function

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

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  $f_1(\underline{x}^*) = 1/9$  at the point  $\underline{x}^* = [4/3 \ 7/9 \ 4/9]^T$ .

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  $f_i(\underline{x})$ ,  $i=2, \dots, 5$ , are forced to be nonnegative

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

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

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

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

and the objective function is

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

The solution is shown for the starting point

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

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  $f_5(\underline{x})$ ) is active at the solution.

PROGRAM TRMMU5(OUTPUT,TAPE6=OUTPUT) 000001  
C 000002  
C BEALE FUNCTION 000003  
C 000004  
DIMENSION X(3),W(179) 000005  
EXTERNAL FDF 000006  
COMMON NCALL 000007  
CALL DATE(DAT) 000008  
CALL TIME(TIM) 000009  
N=3 000010  
M=5 000011  
DDX=0.25 000012  
DO 10 I=1,3 000013  
NCALL=0 000014  
X(1)=0.5 000015  
X(2)=0.5 000016  
X(3)=0.5 000017  
DX=DDX 000018  
EPS=1.E-6 000019  
MAXF=50 000020  
KEQS=2 000021  
IW=179 000022  
WRITE(6,111) DAT,TIM,N,M,MAXF,KEQS,DX,EPS 000023  
111 FORMAT(1H1/" PROGRAM : TRMMU5 - DATE : ",A9," TIME : ",A10) 000024  
1 " BEALE FUNCTION"// 000025  
2 " NUMBER OF VARIABLES:",I4,10X,"NUMBER OF FUNCTIONS:",I4/ 000026  
3 " MAXF VALUE: ",I4,10X,"KEQS VALUE: ",I4/ 000027  
4 " DX VALUE: ",E10.4,10X,"EPS VALUE: ",E10.4/) 000028  
WRITE(6,222) (J,X(J),J=1,N) 000029  
222 FORMAT(/" STARTING POINT: /(20X,I2,2X,F12.8)) 000030  
CALL SECOND(TIME1) 000031  
CALL MMUM2A(FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,IFALL) 000032  
CALL SECOND(TIME2) 000033  
PTIME=TIME2-TIME1 000034  
WRITE(6,333) (J,X(J),J=1,N) 000035  
333 FORMAT(/" SOLUTION: /(20X,I2,2X,F12.8)) 000036  
WRITE(6,444) (J,W(J),J=1,M) 000037  
444 FORMAT(/" FUNCTION VALUES: /(20X,I2,2X,F12.8)) 000038  
W(2)=X(1) 000039  
W(3)=X(2) 000040  
W(4)=X(3) 000041  
W(5)=3.0-X(1)-X(2)-2.0\*X(3) 000042  
WRITE(6,555) (J,W(J),J=2,M) 000043  
555 FORMAT(/" CONSTRAINTS: /(20X,I2,2X,E12.5)) 000044  
WRITE(6,666) IFALL,MAXF,KEQS,PTIME 000045  
666 FORMAT(/" TYPE OF SOLUTION (IFALL): ",I4/ 000046  
1 " NUMBER OF FUNCTION EVALUATIONS: ",I4/ 000047  
2 " NUMBER OF SHIFTS TO STAGE-2: ",I4/ 000048  
3 " EXECUTION TIME (IN SECONDS): ",F7.3/) 000049  
DDX=2.0\*DDX 000050  
10 CONTINUE 000051  
STOP 000052  
END 000053  
C 000054

C SUBROUTINE FDF(N,M,X,DF,F) 000055  
DIMENSION X(N),F(M),DF(M,N) 000056  
COMMON NCALL 000057  
X1=X(1) 000058  
X2=X(2) 000059  
X3=X(3) 000060  
F(1)=9.0-8.0\*X1-6.0\*X2-4.0\*X3+2.0\*(X1\*(X1+X2+X3)+X2\*X2)+X3\*X3 000061  
DF(1,1)=4.0\*X1+2.0\*(X2+X3)-8.0 000062  
DF(1,2)=4.0\*X2+2.0\*X1-6.0 000063  
DF(1,3)=2.0\*(X1+X3)-4.0 000064  
F(2)=F(1)-X1 000065  
F(3)=F(1)-X2 000066  
F(4)=F(1)-X3 000067  
F(5)=F(1)-(3.0-X1-X2-2.0\*X3) 000068  
DO 10 I=1,3 000069  
DO 10 J=1,3 000070  
R=DF(1,J) 000071  
IF(I.EQ.J) R=R-1.0 000072  
10 DF(I+1,J)=R 000073  
DF(5,1)=DF(1,1)+1.0 000074  
DF(5,2)=DF(1,2)+1.0 000075  
DF(5,3)=DF(1,3)+2.0 000076  
DO 20 I=2,5 000077  
IF(F(I).GT.0.0) GOTO 20 000078  
F(I)=0.0 000079  
DO 15 J=1,3 000080  
15 DF(I,J)=0.0 000081  
20 CONTINUE 000082  
IF(NCALL.EQ.0) WRITE(6,111) 000083  
111 FORMAT(/14X,"X(1)",9X,"X(2)",9X,"X(3)",10X,"MAX(F)") 000084  
NCALL=NCALL+1 000085  
FF=F(1) 000086  
J=1 000087  
DO 30 I=2,M 000088  
IF(FF.GE.F(I)) GOTO 30 000089  
FF=F(I) 000090  
J=I 000091  
30 CONTINUE 000092  
WRITE(6,222) NCALL,(X(I),I=1,N),FF,J 000093  
222 FORMAT(1X,I4,2X,3F13.8,F15.8,I4) 000094  
RETURN 000095  
END 000096  
000097

PROGRAM : TRMMU5 - DATE: 82/04/22 TIME: 15.1636.19  
BEALE FUNCTION

NUMBER OF VARIABLES: 3      NUMBER OF FUNCTIONS: UN5  
MAXF VALUE: .50      KEQS VALUE: 2  
DX VALUE: .2500E+00      EPS VALUE: 100 .1000E-05

STARTING POINT:

1	.50000000
2	.50000000
3	.50000000

	X(1)	X(2)	X(3)	MAX(F)	NC
1	.50000000	.50000000	.50000000	2.25000000	1
2	.71052632	.71052632	.71052632	.75415512	1
3	.80110159	.80110159	.80110159	.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 (IF ALL) : 0  
NUMBER OF FUNCTION EVALUATIONS: 19  
NUMBER OF SHIFTS TO STAGE=2: 3  
EXECUTION TIME (IN SECONDS): .18100

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)	ST
1	.50000000	.50000000	.50000000	.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	.441179054	.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): .1536

PROGRAM : TRMMU5 - DATED : 82/04/22 TIME : 15.16.36. 10  
BEALE FUNCTION

NUMBER OF VARIABLES: 3 NUMBER OF FUNCTIONS: 005  
MAXF VALUE: .50 KEQS VALUE: 2  
DX VALUE: .1000E+01 EPS VALUE: .1000E-05

STARTING POINT:

1	.50000000
2	.50000000
3	.50000000

	X(1)	X(2)	X(3)	MAX(F)	FC
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	.44444358	.11111111	1
15	1.33333328	.77777759	.44444456	.11111111	1
16	1.33333333	.77777778	.44444444	.11111111	1

SOLUTION:

1	1.33333331
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: 160  
NUMBER OF SHIFTS TO STAGE-2: 1  
EXECUTION TIME (IN SECONDS): .1568

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

<u>Subroutine</u>	<u>Number of Lines</u> (source text)	<u>Number of Words</u> (compiled code)	<u>Listing from Page</u>
MMUM1A	79	663	53
MMUM2A	9	105	54
MMUM5W	26	206	54
MINI5W	15	142	54
MMX00Z	9	23	55
MMX00Q	35	216	55
MMX00V	26	235	55
MMX00G	35	267	56
MMX00H	67	435	56
MMX00A	28	150	57
MMXP SZ	12	42	58
MMXP LM	11	37	58
MMXL LM	11	36	58
MMXHDR	16	47	58
MMXGLM	13	44	59
MMXGVL	11	41	59
MMUN5W	36	214	59
MMSUBB	288	1222	60
QUAD2	71	331	64
MULT	37	205	65
FPOS	15	143	66
GAUSS	84	335	66
MA19W	15	104	67
MA19BW	412	1672	68

SUBROUTINE MMUM1A (FDF, N, M, X, DX, EPS, MAXF, KEQS, W, IW, LCH, IPR, IFALL) 000001  
EXTERNAL FDF, MMX00Q, MMX00A 000002  
C C C  
LEVEL 1 INTERFACE (STANDARD ENTRY) 000003  
DIMENSION X(1), W(1) 000004  
COMMON /MMX00/ NCH, LV1, LV2, LG1, LG2, LMF, LMV, NRP, NRL, MXL, LMP, LML, LG 000005  
1H, DAT, TIM, LHT, H(8) 000006  
NCH=LCH 000007  
IF (LCH.LE.0) GO TO 40 000008  
I=1ABS(IPR) 000009  
J=I/10 000010  
LG2=MOD(I, 10) 000011  
I=J/10 000012  
LG1=MOD(J, 10) 000013  
J=I/10 000014  
LV2=MOD(I, 10) 000015  
LV1=J 000016  
LG1=LG1\*LV1 000017  
NRP=0 000018  
CALL MMXPZ (-1) 000019  
CALL MMXPLM (-1) 000020  
CALL MMXLIM (-1) 000021  
CALL MMXHDR (-1, HD) 000022  
CALL MMXGLM (-1, -1) 000023  
CALL MMXGVL (-1) 000024  
IF (MXL.NE.0) LML=MXL\*LMP+100 000025  
IF (MXL.EQ.0) MXL=LML+100 000026  
CALL DATE (DAT) 000027  
CALL TIME (TIM) 000028  
CALL MMX00A 000029  
WRITE (LCH, 10) N, M, DX, EPS, MAXF, KEQS, IW, IPR 000030  
10 FORMAT (11H INPUT DATA/1H -----) 000031  
1 27H NUMBER OF VARIABLES (N) ,25(2H. ), I4// 000032  
2 27H NUMBER OF FUNCTIONS (MD) ,25(2H. ), I4// 000033  
3 21H STEP LENGTH (DX) ,25(2H. ), 1PE10.3// 000034  
4 19H ACCURACY (EPS) ,26(2H. ), 1PE10.3// 000035  
5 45H MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) ,16(2H. ), I4// 000036  
6 43H NUMBER OF SUCCESSIVE ITERATIONS (KEQS) ,17(2H. ), I4// 000037  
7 22H WORKING SPACE (IW) ,26(2H. ), 1H., I6// 000038  
8 26H PRINTOUT CONTROL (IPR) ,24(2H. ), 1H., I6// 000039  
NRL=NRL-20 000040  
LML=LML-20 000041  
IF (LV2.NE.0.OR.LV1.EQ.1) GO TO 30 000042  
WRITE (LCH, 20) 000043  
20 FORMAT (19H STARTING POINT :) 000044  
NRL=NRL-1 000045  
LML=LML-1 000046  
CALL FDF (N, M, X, W(M+1), W(1)) 000047  
CALL MMX00V (MMX00A, X, N, W, MD) 000048  
IF (LG2.NE.0) CALL MMX00G (MMX00A, W(M+1), M, N) 000049  
30 IF (IPR.GE.0) GO TO 40 000050  
I=M\*N+M+1 000051  
J=I+M 000052  
K=J+M 000053  
CALL MMX00H (MMX00A, FDF, N, M, X, W(M+1), W(1), W(J), W(K), W(I)) 000054  
40 CALL SECOND (TBEG) 000055  
CALL MMUM5W (MMX00Q, MMX00A, FDF, N, M, X, DX, EPS, MAXF, KEQS, W, IW, IFALL) 000056  
CALL SECOND (TEND) 000057  
IF (LCH.LE.0) RETURN 000058  
IF (IFALL.LT.0) GO TO 70 000059  
IF (NRL.LT.9) CALL MMX00A 000060  
WRITE (LCH, 50) 000061  
50 FORMAT (//9H SOLUTION/9H -----) 000062  
NRL=NRL-4 000063  
000064  
000065

LML=LML-4  
CALL MMX00V (MMX00A,X,N,W,MD  
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 INCORRECT PARAMETERS)  
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)

000066  
000067  
000068  
000069  
000070  
000071  
000072  
000073  
000074  
000075  
000076  
000077  
000078  
000079  
000080  
000081  
000082  
000083  
000084  
000085  
000086  
000087  
000088  
000089  
000090  
000091  
000092  
000093  
000094  
000095  
000096  
000097  
000098  
000099  
000100  
000101  
000102  
000103  
000104  
000105  
000106  
000107  
000108  
000109  
000110  
000111  
000112  
000113  
000114  
000115  
000116  
000117  
000118  
000119  
000120  
000121  
000122  
000123  
000124  
000125  
000126  
000127  
000128  
000129  
000130

IF ( IWR.GT; IW.OR. N.LE.0.OR. M.LE.0.OR. DX.LE.XZERO.OR. MAXF.LE.0.OR.  
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) // 000131  
C DUMMY SUBROUTINE WHICH FOR BASIC AND ORIGINAL ENTRIES SUBSTITUTES 000132  
C SUBROUTINE MMX00Q/11Q. 000133  
C  
C EXTERNAL FUN 000134  
DIMENSION X(N), DF(M,N), F(MD) 000135  
RETURN 000136  
END 000137  
  
C  
C SUBROUTINE MMX00Q (FHH,N,M,X,DF,F,K,NS) // 000138  
C PRINT RESULTS OF FUNCTION EVALUATION. 000139  
C  
C EXTERNAL FHH 000140  
DIMENSION X(N), DF(M,N), F(MD) 000141  
COMMON /MMX000/ LCH, LV1, LV2, LG1, LG2, LMF, LMV, NRP, NRL, MXL, LMP, LML, LG 000142  
1H, DAT, TIM, LHT, H(8) 000143  
IF (LCH.LE.0) RETURN 000144  
IF (LV1+LV2.EQ.0) RETURN 000145  
IF (K.LE.LV2) GO TO 10 000146  
IF (LV1.EQ.0) RETURN 000147  
IF (MOD(K,LV1).NE.0) RETURN 000148  
10 IF (NRP.LE.LMP.AND.LML.GE.0) GO TO 30 000149  
LV1=0 000150  
LV2=0 000151  
WRITE (LCH,20) 000152  
20 FORMAT (/26H ( LISTING LIMIT REACHED )//) 000153  
NRL=NRL-5 000154  
LML=LML-5 000155  
RETURN 000156  
30 IF (NRL.LT.7) CALL FHH 000157  
WRITE (LCH,40) K,NS 000158  
40 FORMAT (22H0FUNCTION EVALUATION :,I4,2H0/,I2) 000159  
NRL=NRL-2 000160  
LML=LML-2 000161  
CALL MMX00V (FHH,X,N,F,MD) 000162  
IF (LG1+LG2.EQ.0) RETURN 000163  
IF (K.LE.LG2) GO TO 50 000164  
IF (K.LE.LV2) RETURN 000165  
IF (LG1.EQ.0) RETURN 000166  
IF (MOD(K,LG1).NE.0) RETURN 000167  
50 CALL MMX00G (FHH,DF,M,N) 000168  
RETURN 000169  
END 000170  
  
C  
C SUBROUTINE MMX00V (FHH,X,N,F,MD) 000171  
C PRINT VALUES OF VARIABLES AND RESIDUAL FUNCTIONS. F 000172  
C  
C DIMENSION X(N), F(MD) 000173  
COMMON /MMX000/ LCH, LV1, LV2, LG1, LG2, LMF, LMV, NRP, NRL, MXL, LMP, LML, LG 000174  
1H, DAT, TIM, LHT, H(8) 000175  
IF (LCH.LE.0) RETURN 000176  
K=MAX0(N,MD) 000177  
IF (NRL.LT.5) CALL FHH 000178  
000179  
000180  
000181  
000182  
000183  
000184  
000185  
000186  
000187  
000188  
000189  
000190  
000191  
000192  
000193  
000194  
000195

```
      WRITE (LCH, 10) 000196
10 FORMAT (/30X, 9H VARIABLES, 18X, 15H FUNCTION VALUES/) 000197
      NRL=NRL-3 000198
      LML=LML-3 000199
      DO 40 I=1,K 000200
      IF (NRL.LE.0) CALL FHH 000201
      IF (I.LE.N.AND.I.LE.M) WRITE (LCH,20) I,X(I),I,F(I) 000202
      IF (I.LE.N.AND.I.GT.M) WRITE (LCH,20) I,X(I) 000203
      IF (I.GT.N.AND.I.LE.M) WRITE (LCH,30) I,F(I) 000204
20 FORMAT (18X, I4,2X, 1PE19.12,5X,I4,2X, 1PE19.12) 000205
30 FORMAT (48X, I4,2X, 1PE19.12) 000206
      NRL=NRL-1 000207
      LML=LML-1 000208
40 CONTINUE 000209
      RETURN 000210
      END 000211
C 000212
C SUBROUTINE MMX00G (FHH,G,M,N) 000213
C PRINT PARTIAL DERIVATIVES OF RESIDUAL FUNCTIONS. 000214
C 000215
C DIMENSION C(M,N) 000216
COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MLX,LMP,LML,LG 000217
1H,DAT,TIM,LHT,H(8) 000218
      IF (LCH.LE.0) RETURN 000219
      IF (NRL.LT.7) CALL FHH 000220
      MM=MIN0(M,LMF) 000221
      NN=MIN0(N,LMV) 000222
      WRITE (LCH,10) 000223
10 FORMAT (30H0 GRADIENTS ( DF.I / DX.J ))::: 000224
      NRL=NRL-2 000225
      LML=LML-2 000226
      DO 60 K=1,NN,LGH 000227
      IF (NRL.LT.5) CALL FHH 000228
      J1=K 000229
      J2=MIN0(NN,K+LGH-1) 000230
      WRITE (LCH,20) (J,J=J1,J2) 000231
20 FORMAT (1H0,9X,12H VARIABLES(J),10(15,5X)) 000232
      WRITE (LCH,30) 000233
30 FORMAT (10X,12H FUNCTIONS(I)) 000234
      NRL=NRL-3 000235
      LML=LML-3 000236
      DO 50 I=1,MM 000237
      IF (NRL.LE.0) CALL FHH 000238
      WRITE (LCH,40) I,(C(I,J),J=J1,J2) 000239
40 FORMAT (10X, I6,4X, 10(1PE10.2)) 000240
      NRL=NRL-1 000241
      LML=LML-1 000242
50 CONTINUE 000243
60 CONTINUE 000244
      RETURN 000245
      END 000246
C 000247
C SUBROUTINE MMX00H (FHH,FDF,N,M,X,DF,F,DG,DH,G) 000248
C 000249
C NUMERICAL VERIFICATION OF USER-DEFINED PARTIAL DERIVATIVES 000250
C (VARIABLES ARE DISTURBED ONE BY ONE). 000251
C 000252
C DIMENSION X(N), DF(M,N), F(M), DG(M), DH(M,N), G(M) 000253
COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MLX,LMP,LML,LG 000254
1H,DAT,TIM,LHT,H(8) 000255
      IF (LCH.LE.0) RETURN 000256
      K=0 000257
      
```

```
CALL FDF (N, M, X, DF, F)          000261
DO 60 I=1,N                         000262
Z=X(I)
DX=1.E-6*Z
IF (ABS(DX).LT.1.E-10) DX=1.E+10  000263
DX2=DX+DX
X(I)=Z+DX
CALL FDF (N, M, X, DH, F)          000264
DO 10 J=1,M                         000265
DG(J)=DH(J,I)
10 CONTINUE                          000266
X(I)=Z-DX
CALL FDF (N, M, X, DH, G)          000267
X(I)=Z
DO 50 J=1,M                         000268
Y=DF(J,I)
Z=F(J)-G(J)
IF (ABS(Z).LE.0.5E-13*(F(J)+G(J)))(Z=0.0) 000269
Z=Z/DX2
IF (ABS(Y).LE.1.E-20.AND.ABS(Z).LE.1.E-20) GO TO 50 000270
IF (ABS(Z).LT.1.E-20) Z=SIGN(1.E-20,Z) 000271
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      CHANGE PAGE AND PRINT PAGE HEADER
C
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 (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 (1H0, 18X, 52H DAT, TIM, NRP)
```

```
30 FORMAT (1H1/7H DATED : ,A10,19X,6HTIME :,A10,20X,6HPAGE :,I3/1H1/7H  
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 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 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 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 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  
000326  
000327  
000328  
000329  
000330  
000331  
000332  
000333  
000334  
000335  
000336  
000337  
000338  
000339  
000340  
000341  
000342  
000343  
000344  
000345  
000346  
000347  
000348  
000349  
000350  
000351  
000352  
000353  
000354  
000355  
000356  
000357  
000358  
000359  
000360  
000361  
000362  
000363  
000364  
000365  
000366  
000367  
000368  
000369  
000370  
000371  
000372  
000373  
000374  
000375  
000376  
000377  
000378  
000379  
000380  
000381  
000382  
000383  
000384  
000385  
000386  
000387  
000388  
000389  
000390
```

```
DO 10 I=1,LL          000391
H(I)=T(I)          000392
10 CONTINUE          000393
RETURN             000394
END               000395
C
C
C SUBROUTINE MMXGLM (K,L)          000396
C
C DEFINE THE SIZE OF PRINTED JACOBIAN. 000397
C
C COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
1H,DAT,TIM,LHT,H(8)          000398
DATA KK/25/,LL/10/          000399
IF (K.GT.0) KK=K          000400
IF (L.GT.0) LL=L          000401
LMF=KK          000402
LMV=LL          000403
RETURN          000404
END               000405
C
C
C SUBROUTINE MMXGVL (L)          000406
C
C DEFINE THE NUMBER OF JACOBIAN COLUMNS PRINTED IN ONE LINE. 000407
C
C COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
1H,DAT,TIM,LHT,H(8)          000408
DATA LL/10/          000409
IF (L.GT.0) LL=MAX0(MIN0(10,L),5) 000410
LCH=LL          000411
RETURN          000412
END               000413
C
C
C SUBROUTINE MMUN5W (FQQ,FHH,FDF,N,M,X,RDX,EPS,MAXFUN,KEQS,DELTA,W,I
1W)          000414
C
C MINIMAX OPTIMIZATION USING QUADRATIC PROGRAMMING. 000415
C KAJ MADSEN, NUMERISK INSTITUT, LYNGBY, DENMARK. 000416
C
C DIMENSION X(N), W(IW)          000417
C
C IW MUST BE AT LEAST 16N+4M+2MN+2N**2+MAX(M,3N**2+6N+5)+13 000418
C
C EXTERNAL FQQ,FHH,FDF          000419
C COMMON /MMU000/ MARK          000420
C MARK=1          000421
C N1=N+1          000422
C N2=N+2          000423
C IIND1=5*N1+M          000424
C IWO=MAX0(N1*(N+5)+M,(2*N+3)**2+1) 000425
C IHLAM=2*N1          000426
C NFO=1          000427
C NF1=NFO+M          000428
C NDF0=NF1+M          000429
C NDF1=NDF0+M*N          000430
C NX1=NDF1+M*N          000431
C NH=NX1+N          000432
C NB=NH+N2          000433
C IIND1=NB+N*N          000434
C IIND0=IIND1+IIND1          000435
C NW0=NIND0+M          000436
C NY=NW0+IWO          000437
C NBH=NY+N          000438
C
C
```

NRLAM=NBR+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(NBHD),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,W0,IWO,Y,BH,RLAM,IRLAM,N1,N2,IEQUAL,DEL)  
EXTERNAL FQQ,FHH,FDF  
DIMENSION X0(N), F0(M), DF0(M,N), DF1(M,N), X1(N), H(N2), B  
1 (N,N), IND1(IIND1), IND0(M), W0(IWO), Y(N), BH(N), RLAGM(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 RLAGM HOLDS AN APPROXIMATION TO THE LAGRANGE MULTIPLIERS -  
C WHEN APPROPRIATE.  
C B IS THE APPROXIMATE HESSIAN, UPDATED BY POWELL METHOD.  
C Y,BH,W0 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  
C EPSFL=2.0\*16.0\*\*(-12)  
C IWO2=SQRT(FLOAT(IWO))  
C NEWTON=.FALSE.  
C KACT0=0  
C RDXFIX=RDX  
C KBOUND=MAX0(IEQUAL,2)  
C KEQUAL=0  
C NTAL=0  
C NQUAD=0

C  
C FIND THE LENGTH OF THE STARTING VECTOR  
C  
C X0MAX=XZERO  
C DO 10 I=1,N  
C X0MAX=AMAX1(X0MAX,ABS(X0(I)))  
10 CONTINUE

C  
C CALCULATE FUNCTION VALUES  
C  
C NTAL=NTAL+1  
C CALL FPOS (FQQ,FHH,FDF,NTAL,0,N,M,X0,DF0,F0)  
C F0MAX=XZERO  
C DO 20 J=1,M  
C IF (F0(J).GT.F0MAX).F0MAX=F0(J)  
20 CONTINUE

C  
000456  
000457  
000458  
000459  
000460  
000461  
000462  
000463  
000464  
000465  
000466  
000467  
000468  
000469  
000470  
000471  
000472  
000473  
000474  
000475  
000476  
000477  
000478  
000479  
000480  
000481  
000482  
000483  
000484  
000485  
000486  
000487  
000488  
000489  
000490  
000491  
000492  
000493  
000494  
000495  
000496  
000497  
000498  
000499  
000500  
000501  
000502  
000503  
000504  
000505  
000506  
000507  
000508  
000509  
000510  
000511  
000512  
000513  
000514  
000515  
000516  
000517  
000518  
000519  
000520

C INITIALIZE ARRAYS 000521  
C 000522  
DO 40 I=1,N 000523  
DO 30 J=1,N 000524  
B(I,J)=XZERO 000525  
30 CONTINUE 000526  
B(I,I)=XONE 000527  
40 CONTINUE 000528  
C ITERATIVE LOOP STARTS HERE ALL 000529  
C 000530  
50 CONTINUE 000531  
C 000532  
C FIND THE SOLUTION H OF THE LINEAR OR QUADRATIC SUBPROBLEM 000533  
C F1PRED IS THE MINIMUM PREDICTED BY THE SUBPROBLEM 000534  
C 000535  
C IF (NEWTON) GO TO 70 000536  
NSTAGE=1 000537  
60 CALL MA19W (N,M,DF0,M,F0,RDX,XZERO,H,W0,IND1,IWO,IIND1) 000538  
F1PRED=W0(M+1) 000539  
GO TO 80 000540  
70 CONTINUE 000541  
NSTAGE=2 000542  
CALL QUAD2 (N,M,DF0,F0,B,H,RLAM,N1,IND1,KACT1,W0,IWO2,EPSFL,I) 000543  
IF (I.NE.0) GO TO 100 000544  
F1PRED=H(N1) 000545  
80 CONTINUE 000546  
IF (F1PRED.GT.F0MAX) GO TO 410 000547  
C FIND THE NORM OF H, AND FIND THE POINT X+H 000548  
C 000549  
HMAX=XZERO 000550  
DO 90 I=1,N 000551  
HMAX=AMAX1(HMAX,ABS(H(I))) 000552  
X1(I)=X0(I)+H(I) 000553  
90 CONTINUE 000554  
C IF THE STEP LENGTH IS TOO LARGE UNDER THE NEWTON ITERATION 000555  
THEN USE THE LP DIRECTION 000556  
C 000557  
IF ((HMAX.LE.RDXFIX).OR. (.NOT. NEWTON)) GO TO 110 000558  
100 NEWTON=.FALSE. 000559  
KEQUAL=0 000560  
NSTAGE=3 000561  
GO TO 60 000562  
C FIND THE NEW FUNCTION VALUES 000563  
C 000564  
110 NTAL=NTAL+1 000565  
CALL FPOS (FQQ,FHH,FDF,NTAL,NSTAGE,N,M,X1,DF1,F1) 000566  
IF (MARK.EQ.0) GO TO 430 000567  
F1MAX=XZERO 000568  
DO 120 J=1,M 000569  
F1MAX=AMAX1(F1MAX,F1(J)) 000570  
120 CONTINUE 000571  
C TEST IF THE NEW POINT IS ACCEPTABLE 000572  
C 000573  
X1OK=(F0MAX-F1MAX).GE.0.01\*(F0MAX-F1PRED) 000574  
C 000575  
C FIND THE SET OF ACTIVE FUNCTIONS 000576  
KACT1=0 000577  
KACTX=0 000578  
DIFFX=XZERO 000579  
000580  
000581  
000582  
000583  
000584  
000585

```
DO 150 J=1,M          000586
SUM=F0(J)             000587
DO 130 I=1,N          000588
SUM=SUM+DF0(J,I)*H(I) 000589
130 CONTINUE           000590
DIFF=ABS(SUM-F1PRED)   000591
IF (KACTX.NE.0.AND.DIFFX.LE.DIFF) GO TO 140
KACTX=J               000592
DIFFX=DIFF             000593
140 IF (DIFF.GT.DEL*ABS(F1PRED)) GO TO 150
KACT1=KACT1+1          000594
IND1(KACT1)=J          000595
150 CONTINUE           000596
IF (KACT1.NE.0) GO TO 160
KACT1=1                000597
IND1(1)=KACTX          000598
160 CONTINUE           000599
C FIND THE MULTIPLIERS RLAM 000600
C IF (.NOT.NEWTON) CALL MULT (N,M,DF0,DF1,X1OK,IND1,N1,KACT1,RLAM,W0 000601
1,IW02,EPSFL)          000602
C FIND Y: THE DIFFERENCE IN THE LAGRANGIAN GRADIENTS 000603
C DO 170 I=1,N          000604
Y(I)=XZERO             000605
170 CONTINUE           000606
DO 190 J=1,KACT1        000607
JK=IND1(J)              000608
DO 180 I=1,N          000609
Y(I)=Y(I)+RLAM(J)*(DF1(JK,I)-DF0(JK,I)) 000610
180 CONTINUE           000611
190 CONTINUE           000612
C ADJUST THE LOCAL BOUND RDX 000613
C IF ((F0MAX-F1MAX).LE.0.25*(F0MAX-F1PRED)) RDX=RDX/XTWO 000614
IF ((F0MAX-F1MAX).GE.0.75*(F0MAX-F1PRED)) RDX=RDX*XTWO 000615
C TEST FOR NEWTON ITERATION 000616
C IF (KACT1.GT.N1) KACT0=0 000617
IF (KACT1.EQ.KACT0) GO TO 210 000618
KACT0=KACT1             000619
DO 200 I=1,KACT1        000620
IND0(I)=IND1(I)          000621
200 CONTINUE           000622
KEQUAL=0                000623
GO TO 270               000624
210 CONTINUE           000625
KEQUAL=KEQUAL+1          000626
DO 220 I=1,KACT1        000627
IF ((IND1(I).EQ.IND0(I)).AND.(RLAM(I).GE.XZERO)) GO TO 220 000628
IND0(I)=IND1(I)          000629
KEQUAL=0                000630
220 CONTINUE           000631
IF (KEQUAL.LT.(KBOUND-1)) GO TO 270 000632
C FIND THE RESIDUAL-NORM OF THE SET OF NON-LINEAR EQUATIONS 000633
C RES=XZERO             000634
DO 240 I=1,N          000635
S=XZERO                000636
DO 230 J=1,KACT1        000637
000638
000639
000640
000641
000642
000643
000644
000645
000646
000647
000648
000649
000650
```

```
JK=IND1(J)          000651
S=S+RLAM(J)*DF1(JK,I) 000652
230 CONTINUE        000653
RES=AMAX1(RES,ABS(S)) 000654
240 CONTINUE        000655
DO 250 J=1, KACT1   000656
JK=IND1(J)          000657
RES=AMAX1(RES,F1MAX-F1(JK)) 000658
250 CONTINUE        000659
IF (KEQUAL.GE.KBOUND) GO TO 260 000660
RES0=RES           000661
GO TO 270          000662
260 IF (RES.LE.0.999*RES0) GO TO 280 000663
270 NEWTON=.FALSE. 000664
GO TO 290          000665
280 IF (.NOT.NEWTON) NQUAD=NQUAD+1 000666
NEWTON=.TRUE.      000667
RES0=RES           000668
C
C     INTRODUCE THE NEW POINT IF IT IS ACCEPTABLE 000669
C
290 IF ((.NOT.X1OK).AND.(.NOT.NEWTON)) GO TO 330 000670
F0MAX=F1MAX        000671
X0MAX=XZERO         000672
DO 310 I=1,N       000673
X0(I)=X1(I)         000674
X0MAX=AMAX1(X0MAX,ABS(X0(I))) 000675
DO 300 J=1,M       000676
DF0(J,I)=DF1(J,I) 000677
300 CONTINUE        000678
310 CONTINUE        000679
DO 320 J=1,M       000680
F0(J)=F1(J)         000681
320 CONTINUE        000682
330 CONTINUE        000683
C
C     ADJUST THE MATRIX/B USING POWELL METHOD 000684
C
C     FIND BH AND YH 000685
C
YH=XZERO           000686
DO 350 J=1,N       000687
YH=YH+Y(J)*H(J)    000688
SUMB=XZERO         000689
DO 340 I=1,N       000690
SUMB=SUMB+B(J,I)*H(I) 000691
340 CONTINUE        000692
BH(J)=SUMB         000693
350 CONTINUE        000694
C
C     FIND T AND SEE IF THETA IS LESS THAN 1 000695
C
T=XZERO           000696
DO 360 I=1,N       000697
T=T+H(I)*BH(I)    000698
360 CONTINUE        000699
IF (YH.GE.XZERO02*T) GO TO 380 000700
THETA=XZERO08*T/(T-YH) 000701
C
C     IF THETA IS TOO SMALL, MATRIX B IS NOT ALTERED 000702
C
IF (THETA.LT.XZERO5) GO TO 400 000703
S=XONE-THETA      000704
YH=XZERO           000705
DO 370 I=1,N       000706
000707
000708
000709
000710
000711
000712
000713
000714
000715
```

Y(I)=THETA\*Y(I)+S\*BH(I) 000716  
YH=YH+Y(I)\*H(I) 000717  
370 CONTINUE 000718  
380 CONTINUE 000719  
C 000720  
C IF T OR YH IS TOO SMALL, TEST THE STOPPING CRITERION 000721  
C IF (ABS(T).LE.1.E-20) GO TO 400 000722  
IF (ABS(YH).LE.1.E-20) GO TO 400 000723  
C 000724  
C FINALLY WE CAN CALCULATE THE NEW B 000725  
C 000726  
DO 390 I=1,N 000727  
S1=BH(I)/T 000728  
S2=Y(I)/YH 000729  
DO 390 J=I,N 000730  
B(I,J)=B(I,J)-S1\*BH(J)+S2\*Y(J) 000731  
390 B(J,I)=B(I,J) 000732  
C 000733  
400 CONTINUE 000734  
C 000735  
C TEST THE STOPPING CRITERION 000736  
C 000737  
IF (NTAL.GE.MAXFUN) GO TO 420 000738  
IF (HMAX.LE.EPS\*X0MAX) GO TO 430 000739  
IF (HMAX.LE.EPSFL\*X0MAX) GO TO 410 000740  
IF (HMAX.GT.X1M50) GO TO 50 000741  
GO TO 430 000742  
C 000743  
410 EPS=XZERO 000744  
GO TO 430 000745  
420 NTAL=NTAL+1 000746  
430 F1(1)=NTAL 000747  
IEQUAL=NQUAD 000748  
RETURN 000749  
END 000750  
C 000751  
C SUBROUTINE QUAD2 (N,M,DF0,F0,B,H,RLAM,N1,IND,KACT1,W,IW2,EPSFL,IE) 000752  
C 000753  
MINIMIZE DELTA+HBH/2 SUBJECT TO F0+DF0\*H=DELTA FOR THE 000754  
C INDICES IN IND. 000755  
C 000756  
C IND GIVES THE INDICES CORRESPONDING TO ACTIVE FUNCTIONS; 000757  
C THERE IS IND(N+2) OF THESE; 000758  
C NI IS N+I; 000759  
C IW MUST BE AT LEAST 2\*N+2. 000760  
C 000761  
DIMENSION F0(M), DF0(M,N), B(N,N), H(N1), RLAM(N1), IND(N1), W(IW2 000762  
1, IW2) 000763  
DATA XZERO,XONE/0.0,1.0/ 000764  
NS=N+KACT1 000765  
NTOT=NS+1 000766  
NTOT1=NTOT+1 000767  
EPS1=EPSFL\*10\*NTOT 000768  
IE=-1 000769  
C 000770  
SET UP THE LINEAR SYSTEM 000771  
1: THE MATRIX (IT IS SYMMETRIC) 000772  
C 000773  
DO 10 I=1,N 000774  
W(I,NTOT)=XZERO 000775  
W(NTOT,I)=XZERO 000776  
DO 10 J=1,I 000777  
W(I,J)=B(I,J) 000778  
000779  
000780

```

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          2: RIGHT HAND SIDE

C          DO 40 I=1, N
C          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          SOLVE THE LINEAR SYSTEM
C          CALL GAUSS (W, IW2, NTOT, NTOT1, EPS1)
C          IF (EPS1.LE.XZERO) GO TO 80

C          STORE THE SOLUTION IN H AND RLAM
C          DO 60 I=1, N
C          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          SUBROUTINE MULT (N, M, DF0, DF1, X1OK, IND1, N1, KACT1, RLAM, W, IW2, EPSFL)
C          FIND THE MULTIPLIERS RLAM BY A LEAST SQUARES CALCULATION, SUBJECT 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)
C          LOGICAL X1OK
C          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)
10 S=S-1.0
20 CONTINUE
30 CONTINUE
40 CONTINUE
50 CONTINUE
60 CONTINUE
70 CONTINUE
80 CONTINUE
90 CONTINUE
100 CONTINUE
110 CONTINUE
120 CONTINUE
130 CONTINUE
140 CONTINUE
150 CONTINUE
160 CONTINUE
170 CONTINUE
180 CONTINUE
190 CONTINUE
200 CONTINUE
210 CONTINUE
220 CONTINUE
230 CONTINUE
240 CONTINUE
250 CONTINUE
260 CONTINUE
270 CONTINUE
280 CONTINUE
290 CONTINUE
300 CONTINUE
310 CONTINUE
320 CONTINUE
330 CONTINUE
340 CONTINUE
350 CONTINUE
360 CONTINUE
370 CONTINUE
380 CONTINUE
390 CONTINUE
400 CONTINUE
410 CONTINUE
420 CONTINUE
430 CONTINUE
440 CONTINUE
450 CONTINUE
460 CONTINUE
470 CONTINUE
480 CONTINUE
490 CONTINUE
500 CONTINUE
510 CONTINUE
520 CONTINUE
530 CONTINUE
540 CONTINUE
550 CONTINUE
560 CONTINUE
570 CONTINUE
580 CONTINUE
590 CONTINUE
600 CONTINUE
610 CONTINUE
620 CONTINUE
630 CONTINUE
640 CONTINUE
650 CONTINUE
660 CONTINUE
670 CONTINUE
680 CONTINUE
690 CONTINUE
700 CONTINUE
710 CONTINUE
720 CONTINUE
730 CONTINUE
740 CONTINUE
750 CONTINUE
760 CONTINUE
770 CONTINUE
780 CONTINUE
790 CONTINUE
800 CONTINUE
810 CONTINUE
820 CONTINUE
830 CONTINUE
840 CONTINUE
850 CONTINUE
860 CONTINUE
870 CONTINUE
880 CONTINUE
890 CONTINUE
900 CONTINUE
910 CONTINUE
920 CONTINUE
930 CONTINUE
940 CONTINUE
950 CONTINUE

```

```
GO TO 20 000846
10 S=S+DF1( IK, L)*DF1( JK, L) 000847
20 CONTINUE 000848
      W( I, J)=S 000849
      W( J, I)=S 000850
30 CONTINUE 000851
      W( I, K1)=-XONE 000852
      W( K1, I)=XONE 000853
      W( I, K2)=XZERO 000854
40 CONTINUE 000855
      W( K1, K1)=XZERO 000856
      W( K1, K2)=XONE 000857
      CALL GAUSS ( W, IW2, K1, K2, EPS1 ) 000858
      DO 50 I=1, KACT1 000859
      RLAM( I)=W( I, K2) 000860
50 CONTINUE 000861
      RETURN 000862
      END 000863
C 000864
C 000865
C SUBROUTINE FPOS (FQQ, FHH, FDF, NC, NS, N, M, X, DF, F) 000866
C EXTERNAL FHH 000867
C DIMENSION X( N ), DF( M, N ), F( M ) 000868
C DATA XZERO/0.0/ 000869
C CALL FDF ( N, M, X, DF, F ) 000870
C CALL FQQ ( FHH, N, M, X, DF, F, NC, NS ) 000871
C DO 20 J=1, M 000872
C IF ( F( J ) .GE. XZERO ) GO TO 20 000873
C F( J )=-F( J ) 000874
C DO 10 I=1, N 000875
C DF( J, I )=-DF( J, I ) 000876
10 CONTINUE 000877
20 CONTINUE 000878
      RETURN 000879
      END 000880
C 000881
C 000882
C SUBROUTINE GAUSS (A, IA, N, M, EPS) 000883
C 000884
C SOLUTION OF A SET OF N LINEAR EQUATIONS IN N UNKNOWN WITH M-N 000885
C RIGHT HAND SIDES. THE SET OF SOLUTIONS WILL BE STORED IN PLACE 000886
C OF THE RIGHT HAND SIDES: IN THE LAST M-N COLUMNS OF MATRIX A. 000887
C KAJ MADSEN, NUMERISK INSTITUT, LYNGBY, AUGUST 1980. 000888
C 000889
C DIMENSION A( IA, M ) 000890
C DATA XONE/1.0/ 000891
C N1=N+1 000892
C IF ( N.EQ.1 ) GO TO 100 000893
C 000894
C EQUILIBRATION 000895
C 000896
C DO 30 I=1, N 000897
C C=ABS(A( I, 1 )) 000898
C DO 10 J=2, N 000899
C IF ( ABS(A( I, J )) .GT. C ) C=ABS(A( I, J )) 000900
10 CONTINUE 000901
DO 20 J=1, M 000902
A( I, J )=A( I, J )/C 000903
20 CONTINUE 000904
30 CONTINUE 000905
C 000906
C PIVOTING AND REDUCTION TO TRIANGULAR FORM 000907
C 000908
C NM=N-1 000909
DO 90 K=1, NM 000910
```

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 TEST FOR SINGULARITY  
C IF (C.LT.EPS) GO TO 150  
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 END OF REDUCTION  
C 100 CONTINUE  
C TEST FOR SINGULARITY  
C IF (ABS(A(N,N)).LT.EPS) GO TO 150  
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 000976  
N2=N+2 000977  
NURHO=MAX0(M,3\*N1)+1 000978  
NH=1+NURHO 000979  
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(MD), 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 000986  
NN3=NN2+N1 000987  
LREF=NN3+N1 000988  
LBND=LREF+M 000989  
M1=M+1 000990  
SI4N=XONE/(XFOUR\*N) 000991  
NTAL=0 000992

C C FIND EQUATION I0 WHICH GOES INTO THE FIRST REFERENCE  
C C  
C=-XONE 000993  
DO 10 J=1,M 000994  
IREF(LREF+J)=0 000995  
IF (ABS(B(J)).LT.C) GO TO 10 000996  
C=ABS(B(J)) 000997  
I0=J 000998  
10 CONTINUE 000999

C C INITIALIZE REFERENCE ARRAYS  
C S=XZERO 001000  
T=B(I0) 001001  
XM=M 001002  
DO 20 I=1,N 001003  
D=A(I0,I) 001004  
S=S+ABS(D) 001005  
XM=XM+XONE 001006  
IF (D.EQ.XZERO) D=-XONE 001007  
IREF(I)=SIGN(XM,-D\*T) 001008  
IREF(LBND+I)=IREF(I) 001009

C 20 CONTINUE 001010  
XM=I0 001011  
IREF(N1)=SIGN(XM,T) 001012  
IREF(LREF+I0)=IREF(N1) 001013

C C INITIALIZE DH,DG, AND GAM  
C IF ((DX\*S).GT.C) GO TO 30 001014  
GAM=DX 001015  
DG=GAM 001016  
DH=C-DX\*S 001017  
GO TO 40 001018

C 30 DG=C/S 001019  
DH=XZERO 001020  
GAM=DG 001021

IF (DX\*S.GT.C) GO TO 30 001022  
GAM=DX 001023  
DG=GAM 001024  
DH=C-DX\*S 001025  
GO TO 40 001026

30 DG=C/S 001027  
DH=XZERO 001028  
GAM=DG 001029

IF ((DX\*S).GT.C) GO TO 30 001030  
GAM=DX 001031  
DG=GAM 001032  
DH=C-DX\*S 001033  
GO TO 40 001034

30 DG=C/S 001035  
DH=XZERO 001036  
GAM=DG 001037

IF ((DX\*S).GT.C) GO TO 30 001038  
GAM=DX 001039  
DG=GAM 001040

C FIND VECTOR X 001041  
C 001042  
C 001043  
C 40 DO 50 I=1,N 001044  
XM=IREF(I) 001045  
X(I)=SIGN(DG,XM) 001046  
50 CONTINUE 001047  
C FIND MATRIX H 001048  
C 001049  
C 001050  
S=XONE/(S+XONE) 001051  
H(N1,N1)=S 001052  
IREF(NN2+N1)=0 001053  
DO 80 I=1,N 001054  
XM=-IREF(I) 001055  
H(N1,I)=SIGN(S,XM) 001056  
T=ABS(A(I0,I))\*S 001057  
H(I,N1)=T 001058  
DO 60 J=1,N 001059  
XM=-IREF(J) 001060  
H(I,J)=SIGN(T,XM) 001061  
60 CONTINUE 001062  
IF (T.GT.XZERO) GO TO 70 001063  
IREF(NN2+I)=1 001064  
H(I,N2)=XONE 001065  
GO TO 80 001066  
70 IREF(NN2+I)=0 001067  
80 H(I,I)=ISIGN(1,IREF(I))+H(I,I) 001068  
C INITIALIZE SOME CONSTANTS 001069  
C 001070  
C 001071  
RSIG=XONE-S 001072  
DCH=DG 001073  
DH1=DH 001074  
ETA=GAM 001075  
ERRX=XZERO 001076  
GAMCH=.TRUE. 001077  
NBIN=N 001078  
GO TO 650 001079  
C ITERATIVE LOOP STARTS HERE 001080  
C FIND VECTOR RHO 001081  
C 001082  
C 001083  
90 DGH=DG 001084  
IF (I0S.LT.0) GO TO 120 001085  
DO 110 I=1,N1 001086  
S=-H(I,N1) 001087  
DO 100 J=1,N 001088  
S=S-H(I,J)\*A(I0,J) 001089  
100 CONTINUE 001090  
RHO(I)=S 001091  
110 CONTINUE 001092  
GO TO 190 001093  
120 DO 140 I=1,N1 001094  
S=-H(I,N1) 001095  
DO 130 J=1,N 001096  
S=S+H(I,J)\*A(I0,J) 001097  
130 CONTINUE 001098  
RHO(I)=S 001099  
140 CONTINUE 001100  
GO TO 190 001101  
C BOUNDS VIOLATED 001102  
C 001103  
C 001104  
150 I0=M+J0 001105

DHH=DH 001106  
IOS=SIGN(XONE,X(J0)) 001107  
IF (IOS.LT.0) GO TO 170 001108  
DO 160 I=1,N1 001109  
RHO(I)=-H(I,J0)-H(I,N1) 001110  
160 CONTINUE 001111  
GO TO 190 001112  
170 DO 180 I=1,N1 001113  
RHO(I)=H(I,J0)-H(I,N1) 001114  
180 CONTINUE 001115  
C 001116  
C FIND EQUATION L0 WHICH LEAVES THE REFERENCE 001117  
C FIND -H(I,N1)/RHO(I) FOR NEGATIVE VALUES OF RHO(I) 001118  
C 001119  
190 LB=0 001120  
RTAU=XZERO 001121  
IF (IO.GT.MD RTAU=XONE 001122  
DO 210 I=1,N1 001123  
IF (RHO(I).GE.XZERO) GO TO 200 001124  
LB=LB+1 001125  
IREF(N1+LB)=I 001126  
KK=N1 001127  
IF (IREF(NN2+I).GT.0) KK=N2 001128  
RHO(N1+I)=-H(I,KK)/RHO(I) 001129  
200 IF (IABS(IREF(I)).LE.MD GO TO 210 001130  
RTAU=RTAU+RHO(I) 001131  
210 CONTINUE 001132  
C 001133  
C FIND THE COEFFICIENTS IN THE RATIONAL EXPRESSION 001134  
C (TT+LAM\*SS)/(RSIG+LAM\*RTAU) 001135  
C 001136  
DG2=DG\*X TWO 001137  
IF (DH.GT.XZERO) GO TO 220 001138  
TT=DG\*RSIG 001139  
SS=DG\*RTAU+DHH+DGH-DG 001140  
NL=1 001141  
GO TO 230 001142  
220 RSIG=XONE-RSIG 001143  
RTAU=-RTAU 001144  
TT=DH\*RSIG 001145  
SS=DH\*RTAU+DHH-DH+DGH-DG 001146  
NL=2 001147  
230 SMAX=XZERO 001148  
L=1 001149  
240 LA=L 001150  
C 001151  
C FIND MINIMUM VALUE OF -H(I,N1)/RHO(I) 001152  
C 001153  
L0=IREF(N1+LA) 001154  
LAM=RHO(N1+L0) 001155  
ILAM=IREF(NN2+L0) 001156  
IF (LA.EQ.LB) GO TO 260 001157  
L=LA+1 001158  
LM=LA 001159  
DO 250 I=L,LB 001160  
IN=IREF(N1+I) 001161  
S=RHO(N1+IN) 001162  
IS=IREF(NN2+IN) 001163  
IF (((IS.EQ.ILAM).AND.(S.GE.LAM)).OR.((IS.LT.ILAM).GO TO 250 001164  
L0=IN 001165  
LAM=S 001166  
ILAM=IS 001167  
LM=I 001168  
250 CONTINUE 001169  
C 001170

C REORDER 001171  
C IREF(N1+LMD = IREF(N1+LA) 001172  
C IREF(N1+LA) = L0 001173  
C FIND MAX(-RHO(I)) 001174  
C 001175  
C 001176  
C 001177  
C 260 SMAX=AMAX1(SMAX,-RHO(L0)) 001178  
C REVISE THE COEFFICIENTS OF THE RATIONAL EXPRESSION 001179  
C 001180  
C 001181  
C ML=NL 001182  
C IF (IABS(IREF(L0)).GT.MD ML=NL+2 001183  
C GO TO (290,280,280,270), ML 001184  
C 270 TT=TT+DG2\*X(H(L0,N1) 001185  
C SS=SS+DG2\*RHO(L0) 001186  
C GO TO 290 001187  
C 280 RSIG=RSIG-XTWO\*X(H(L0,N1) 001188  
C RTAU=RTAU-XTWO\*RHO(L0) 001189  
C 290 IF ((RSIG\*SS.GT.RTAU\*TT).AND.(LA.LT.LB)) GO TO 240 001190  
C TEST IF -RHO(L0) IS TOO SMALL 001191  
C 001192  
C 001193  
C SMAX=SMAX/XFOUR 001194  
C 300 IF (-RHO(L0).GE.SMAX) GO TO 310 001195  
C LA=LA-1 001196  
C L0=IREF(N1+LA) 001197  
C GO TO 300 001198  
C UPDATE REFERENCE ARRAYS 001199  
C 001200  
C 001201  
C 310 IREF(LREF+I0)=ISIGN(1,I0S) 001202  
C IREF(LREF+IABS(IREF(L0)))=0 001203  
C IF (IABS(IREF(L0)).GT.MD NBIN=NBIN-1 001204  
C IF (I0.GT.MD NBIN=NBIN+1 001205  
C IREF(L0)=ISIGN(I0,I0S) 001206  
C UPDATE MATRIX H 001207  
C 001208  
C 001209  
C RHOL0=RHO(L0) 001210  
C DO 320 J=1,N1 001211  
C H(L0,J)=-H(L0,J)/RHOL0 001212  
C 320 CONTINUE 001213  
C DO 340 I=1,N1 001214  
C IF (I.EQ.L0) GO TO 340 001215  
C S=RHO(I) 001216  
C DO 330 J=1,N1 001217  
C H(I,J)=S\*H(L0,J)+H(I,J) 001218  
C 330 CONTINUE 001219  
C 340 CONTINUE 001220  
C IF ANY SIGNS HAVE BEEN CHANGED, UPDATE H 001221  
C 001222  
C 001223  
C RHON1=XONE 001224  
C DO 350 I=1,N1 001225  
C RHO(N1+I)=XONE 001226  
C 350 CONTINUE 001227  
C IF (LA.LE.1) GO TO 460 001228  
C K=LA-1 001229  
C DO 360 I=1,N1 001230  
C RHO(NN2+I)=XZERO 001231  
C 360 CONTINUE 001232  
C DO 370 I=1,K 001233  
C RHO(IREF(N1+I)+N1)=-XONE 001234  
C 370 CONTINUE 001235

DO 410 I=1,N1	001236
IF (RHO(N1+I).GT.XZERO) GO TO 390	001237
DO 380 J=1,N1	001238
RHO(NN2+J)=RHO(NN2+J)-H(I,J)	001239
380 CONTINUE	001240
GO TO 410	001241
390 DO 400 J=1,N1	001242
RHO(NN2+J)=RHO(NN2+J)+H(I,J)	001243
400 CONTINUE	001244
410 CONTINUE	001245
RHON1=RHO(NN2+N1)	001246
DO 420 J=1,N1	001247
H(J,N1)=H(J,N1)/RHON1	001248
420 CONTINUE	001249
DO 440 I=1,N	001250
S=RHO(NN2+I)	001251
DO 430 J=1,N1	001252
H(J,I)=-S*H(J,N1)+H(J,I)	001253
430 CONTINUE	001254
440 CONTINUE	001255
C C CHANGE SIGNS IN SOME ROWS OF MATRIX HF	001256
DO 450 L=1,K	001257
I=IREF(N1+L)	001258
J=-IREF(I)	001259
IREF(I)=J	001260
IREF(LREF+IABS(J))=J	001261
DO 450 J=1,N1	001262
450 H(I,J)=-H(I,J)	001263
C C UPDATE THE LAST COLUMN OF HF IN CASE OF DEGENERACIES	001264
460 IF (IREF(NN2+L0).EQ.0) GO TO 520	001265
H(L0,N2)=-H(L0,N2)/(RHO(L0)*RHON1)	001266
DO 510 I=1,N1	001267
IF ((IREF(NN2+I).EQ.0).OR.(I.EQ.L0)) GO TO 510	001268
IF (IREF(NN2+I)-IREF(NN2+L0)).EQ.0,470,490,480	001269
470 H(I,N2)=H(I,N2)/RHON1	001270
GO TO 510	001271
480 H(I,N2)=ABS(H(L0,N2)*RHO(I))	001272
IREF(NN2+I)=IREF(NN2+L0)	001273
GO TO 510	001274
490 C=H(I,N2)/RHON1+H(L0,N2)*RHO(I)	001275
IF ((LA.GT.1).AND.(RHO(N1+I).LT.XZERO)) C=-C	001276
IF (C.GT.XZERO) GO TO 500	001277
H(I,N2)=XONE	001278
IREF(NN2+I)=IREF(NN2+I)+1	001279
GO TO 510	001280
500 H(I,N2)=C	001281
510 CONTINUE	001282
GO TO 540	001283
C C TEST FOR DEGENERACIES	001284
520 DO 530 I=1,N1	001285
IREF(NN2+I)=0	001286
IF (H(I,N1).GT.XZERO) GO TO 530	001287
IREF(NN2+I)=1	001288
H(I,N1)=XZERO	001289
H(I,N2)=XONE	001290
530 CONTINUE	001291
C C UPDATE GAM	001292
	001293
	001294
	001295
	001296
	001297
	001298
	001299
	001300

540 GAMCH=.FALSE. 001301  
IF ((NBIN.EQ.0).OR.(GAMM.LT.XTWO\*GAM.AND.GAMM.LT.DX)) GO TO 550 001302  
IF (GAM.LT.GAMM) GAMCH=.TRUE. 001303  
GAM=GAMM 001304  
C  
C UPDATE DH AND DG 001305  
C  
550 S=XZERO 001308  
RSIG=XZERO 001309  
DO 570 I=1,N1 001310  
K=IABS(IREF(I)) 001311  
IF (K.GT.MD GO TO 560 001312  
RHO(I)=B(K)\*ISIGN(1,IREF(I)) 001313  
S=S+H(I,N1)\*RHO(I) 001314  
GO TO 570 001315  
560 RSIG=RSIG+H(I,N1) 001316  
570 CONTINUE 001317  
IF (RSIG.NE.XZERO) GO TO 580 001318  
DH=ABS(S) 001319  
DG=GAM 001320  
GO TO 590 001321  
580 DG=AMIN1(GAM,ABS(S)/RSIG) 001322  
DH=XZERO 001323  
IF (DG.EQ.GAM) DH=ABS(S-DG\*RSIG)/(XONE-RSIG) 001324  
C  
C CALCULATE PARAMETER VALUES 001325  
C  
590 DGH=XZERO 001328  
ERRX=XZERO 001329  
DO 610 I=1,N1 001330  
IF (IABS(IREF(I)).GT.MD GO TO 600 001331  
RHO(I)=DH-RHO(I) 001332  
GO TO 610 001333  
600 RHO(I)=DG 001334  
610 CONTINUE 001335  
DO 640 I=1,N 001336  
S=H(I,I)\*RHO(I) 001337  
DO 620 J=2,N1 001338  
S=S+H(J,I)\*RHO(J) 001339  
620 CONTINUE 001340  
IF (IREF(LBND+I).EQ.0) GO TO 630 001341  
T=S 001342  
XM=IREF(LBND+I) 001343  
S=SIGN(DG,XM) 001344  
ERRX=AMAX1(ERRX,ABS(S-T)) 001345  
630 X(I)=S 001346  
IF (ABS(S).LE.DGH) GO TO 640 001347  
DGH=ABS(S) 001348  
J0=I 001349  
640 CONTINUE 001350  
NTAL=NTAL+1 001351  
C  
C CALCULATE GAMM 001352  
C  
650 GAMM=AMIN1(AMAX1(XFIVE\*DG,GAM),DX) 001353  
C  
C FIND EQUATION 10 WHICH GOES INTO THE REFERENCE 001354  
C  
DHH=XZERO 001355  
T=DH 001356  
T1=DH 001357  
DO 680 I=1,M 001358  
S=B(I) 001360  
DO 660 J=1,N 001361  
S=S+A(I,J)\*X(J) 001362  
001363  
001364  
001365

660	CONTINUE	001366
	RHO(I)=S	001367
	IF (IREF(LREF+I).NE.0) GO TO 670	001368
	IF (ABS(S).LE.DHH) GO TO 680	001369
	DHH=ABS(S)	001370
	I0S=SIGN(XONE,S)	001371
	I0=I	001372
	GO TO 680	001373
670	T=AMAX1(T,ABS(S))	001374
	T1=A MIN1(T1,ABS(S))	001375
680	CONTINUE	001376
C		001377
C	CALCULATE DH1 AND ETA	001378
C	ETA=AMAX1(GAMM,SIGN(ETA+ERRX,DH1-T))	001379
	IF (.NOT.GAMCH) T=AMAX1(T,DH1+(T-T1))	001380
	DH1=AMAX1(T,DH+EPSH)	001381
C		001382
C	TEST IF CONSTRAINTS ARE VIOLATED	001383
C	IF (DGH.GT.ETA) GO TO 150	001384
C		001385
C	TEST OF CONVERGENCE CRITERION	001386
C	IF (DHH.GT.DH1) GO TO 90	001387
	IF ((GAM.LT.DX).AND.(DHH+DH.GT.(DH1-DH)*XTWO).AND.(NBIN.GT.0)) GO	001388
1TO 540		001389
	RHO(M1)=DH	001390
	IREF(N2)=NBIN	001391
	IREF(N+3)=NTAL	001392
	RETURN	001393
	END	001394
		001395
		001396
		001397

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.

