INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-294

MFNC - A FORTRAN PACKAGE FOR MINIMIZATION WITH GENERAL CONSTRAINTS

J.W. Bandler and W.M. Zuberek

June 1982

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

MFNC - A FORTRAN PACKAGE FOR MINIMIZATION WITH GENERAL CONSTRAINTS

J.W. Bandler and W.M. Zuberek

Abstract

MFNC is a package of subroutines for minimization of a nonlinear objective function subject to nonlinear constraints. It is an extension and modification of a set of subroutines from the Harwell Subroutine Library (subroutines VF02AD, VF02BD, VF02CD, VE02A, LA02A, MB01C, FM02AS). First derivatives of all functions with respect to all variables are assumed to be available. The solution is found by an iteration that minimizes a quadratic approximation of the objective function subject to linearized constraints. The method was presented by Han and Powell. The package and documentation have been 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.

I. INTRODUCTION

The package of Harwell subroutines (with the main subroutine VF02AD) for minimization with nonlinear constraints [1,2,3] has recently been modified and extended to provide a uniform printed output of input parameters as well as intermediate and final results of optimization. The modifications include:

(1) conversion to single precision,

(2) replacement of the subroutine MB01B by MB01C, which supersedes MB01B and removes the restriction on the matrix order,

(3) adjustments in the subroutines LA02A and VE02A required by MB01C,

(4) standardization of the source code.

The extensions, in the form of additional subroutines, contain:

(1) more flexible and more detailed printed output generated by the package,

(2) numerical verification of partial derivatives,

(3) replacement of the "reverse communication" by the separate user-defined subroutine that evaluates the functions and their first-order derivatives.

Consequently, the calling sequences have been changed appropriately, however, the original call of the subroutine VF02A (in single precision) has been preserved with slight modifications only.

The whole package is written in Fortran IV for the CDC 170/730 system. At McMaster University it is available in the form of a library of binary relocatable subroutines which is linked with the user's program by the appropriate call of the main subroutine in the package. The name of the library is LIBRMFN. The library is available as a group indirect file under the charge RJWBAND. The general sequence of NOS commands to use the package can be as follows:

/GET(LIBRMFN/GR) - fetch the library,

/LIBRARY(LIBRMFN) - indicate library to the loader,

/FTN(...,GO) - compile, load and execute the program.

The user's program should be composed (at least) of:

- the main segment that prepares arguments and calls the main subroutine of the package,
- the subroutine which evaluates the objective and constraint functions and their partial derivatives at points determined by the package; the name of this subroutine can be arbitrary because it is transferred to the package as one of the arguments.

II. GENERAL DESCRIPTION

The purpose of the package is to minimize the objective function $F(\underset{\sim}{x})$ of n variables, $\underset{\sim}{x} = [x_1 \ldots x_n]^T$, subject to the general equality and inequality constraints

$$f_j(\underset{\sim}{x}) = 0, \qquad j=1, \ldots, \ell_{eq},$$
$$f_j(\underset{\sim}{x}) \geq 0, \qquad j=\ell_{eq}+1, \ldots, \ell,$$

where the objective and the constraint functions are differentiable and their first-order derivatives are available.

The algorithm used in the package is Powell's [1,4] variable metric method for constrained optimization which is based on the results of Han [2]. In each kth iteration the search direction $\underset{\sim}{h}^k$ is determined as the solution of the linearly constrained quadratic minimization subproblem

$$\underset{\underset{\sim}{h}^k}{\text{Minimize}}\ \tilde{F}(\underset{\sim}{x}^{k-1}, \underset{\sim}{h}^k) = F(\underset{\sim}{x}^{k-1}) + \underset{\sim}{h}^{k^T} \underset{\sim}{F}'(\underset{\sim}{x}^{k-1}) + 0.5\ \underset{\sim}{h}^{k^T} \underset{\sim}{B}^k \underset{\sim}{h}^k$$

subject to the constraints

$$\underset{\sim}{h}^{k^T} \underset{\sim}{f}'_j(\underset{\sim}{x}^{k-1}) + \alpha^k f_j(\underset{\sim}{x}^{k-1}) = 0 , \qquad j=1,\dots,\ell_{eq} ,$$

$$\underset{\sim}{h}^{k^T} \underset{\sim}{f}'_j(\underset{\sim}{x}^{k-1}) + \alpha^k_j f_j(\underset{\sim}{x}^{k-1}) \geq 0 , \qquad j=\ell_{eq}+1,\dots,\ell ,$$

$$0 \leq \alpha^k \leq 1 ,$$

where $\underset{\sim}{F}'(\underset{\sim}{x})$ and $\underset{\sim}{f}'_j(\underset{\sim}{x})$, $j=1,\dots,\ell$, are the gradient vectors of the objective and constraint functions, respectively, $\underset{\sim}{B}^k$ is a positive definite square matrix of dimension n containing second-order derivative information which is updated in consecutive iterations according to the BFGS formula (initially the matrix is set to the unit matrix, $\underset{\sim}{B}^0 = \underset{\sim}{1}$), and α^k is an additional variable introduced in order to allow infeasibility in linearized constraints, while α^k_j, $j=\ell_{eq}+1,\dots,\ell$, are defined as

$$\alpha^k_j = \begin{cases} 1, \text{ if } f_j(\underset{\sim}{x}^{k-1}) > 0 , \\ \alpha^k, \text{ if } f_j(\underset{\sim}{x}^{k-1}) \leq 0 . \end{cases}$$

Usually the solution of the quadratic subproblem results in $\alpha^k = 1$. If the only feasible solution corresponds to $\alpha^k = 0$ and $\underset{\sim}{h}^k = \underset{\sim}{0}$, the algorithm terminates and it is assumed that the constraints are inconsistent. Positive values of α^k are used in a subsequent one-dimensional search of the consecutive approximations $\underset{\sim}{x}^k$ of the solution

$$\underset{\sim}{x}^k = \underset{\sim}{x}^{k-1} + \beta^k \underset{\sim}{h}^k ,$$

where β^k is a positive multiplier, $0 < \beta^k \leq 1$, which is chosen in such a way that

$$\overline{F}(\underset{\sim}{x}^{k-1} + \beta^k \underset{\sim}{h}^k, \underset{\sim}{\mu}^k) < \overline{F}(\underset{\sim}{x}^{k-1}, \underset{\sim}{\mu}^k) ,$$

where

$$\overline{F}(\underset{\sim}{x}, \underset{\sim}{\mu}) = F(\underset{\sim}{x}) + c(\underset{\sim}{x}, \underset{\sim}{\mu})$$

and

$$c(\underset{\sim}{x}, \underset{\sim}{\mu}) = \sum_{1 \leq j \leq \ell_{eq}} \mu_j |f_j(\underset{\sim}{x})| + \sum_{\ell_{eq} < j \leq \ell} \mu_j |\min(0, f_j(\underset{\sim}{x}))| .$$

$c(\underset{\sim}{x}, \underset{\sim}{\mu})$ is equal to zero when all the constraints are satisfied, and is positive otherwise. The vectors $\underset{\sim}{\mu}^k$ depend on the Lagrangian multipliers $\underset{\sim}{\lambda}^k$ (determined at the solution $\underset{\sim}{h}^k$ of the quadratic subproblem) in the following way:

$$\mu_j^1 = |\lambda_j^1|, \ j=1,\ldots,\ell,$$
$$\mu_j^k = \max(|\lambda_j^k|, \ 0.5 \ (\mu_j^{k-1} + |\lambda_j^k|)), \ j=1,\ldots,\ell, \ k=2,3,\ldots .$$

The multiplier β^k is determined iteratively (line search) starting with the value $\beta_1^k = 1$. In each step i of the search

$$\beta_{i+1}^k = \max(0.1 \ \beta_i^k, \ \overline{\beta}_i^k)$$

where $\overline{\beta}_i^k$ is the value that minimizes the quadratic approximation of the function

$$\overline{F}(\underset{\sim}{x}^{k-1} + \beta_i^k \ \underset{\sim}{h}^k, \underset{\sim}{\mu}^k) .$$

The value β^k is equal to the first β_i^k that satisfies the condition

$$\bar{F}(\underset{\sim}{x}^{k-1} + \beta_i^k \underset{\sim}{h}^k, \underset{\sim}{\mu}^k) \leq \bar{F}(\underset{\sim}{x}^{k-1}, \underset{\sim}{\mu}^k) + 0.1\ \beta_i^k (\underset{\sim}{h}^{k^T} \underset{\sim}{F}'(\underset{\sim}{x}^{k-1}) - \alpha^k\ c(\underset{\sim}{x}^{k-1}, \mu^k)).$$

Usually the condition is satisfied in the first step of the line search and $\beta^k = \beta_1^k = 1$. However, when the starting point is far from the solution, more line search steps can be required. The algorithm terminates if the number of required line search steps is greater than 5 since it is assumed then that the gradient vectors are incorrect.

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

(1) the required accuracy is obtained

$$|\underset{\sim}{h}^{k^T} \underset{\sim}{F}'(\underset{\sim}{x}^{k-1})| + \sum_{1 \leq j \leq \ell} |\lambda_j^k f_j(\underset{\sim}{x}^{k-1})| \leq \varepsilon ,$$

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

(2) an uphill search direction is obtained, which can only be due to rounding errors; the required accuracy cannot be obtained in this case,

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

(4) the line search procedure requires more than 5 steps, which is usually due to incorrect derivatives but can also occur when the required accuracy cannot be achieved and the function values are dominated by rounding errors,

(5) a vector of variables that satisfy the constraints cannot be determined, which is usually due to inconsistent constraints but can also occur when constraint function derivatives are incorrect,

(6) the changes of the values of variables are restricted by an artificial bound (with default value 10^6) which is usually due to an unbounded solution but may also occur when the problem is badly scaled.

Moreover, the user can terminate the iterative procedure and cause the return from the package by setting one of parameters during evaluation of functions and their first-order derivatives (see argument FCD).

III. STRUCTURE OF THE PACKAGE

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

1. subroutine MFNC1A - standard entry which provides uniform printing of input parameters as well as intermediate and final results,
2. subroutine MFNC2A - 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 VF02A - original entry, as defined in VF02AD subroutine specification [3].

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 MFNC1A) is called, and that the subroutine MFN00Q (Fig. 1), which is responsible for printing the values of the functions and their first-order derivatives, is replaced by the dummy subroutine MFN00Z (Fig. 2) when entry 2 is used.

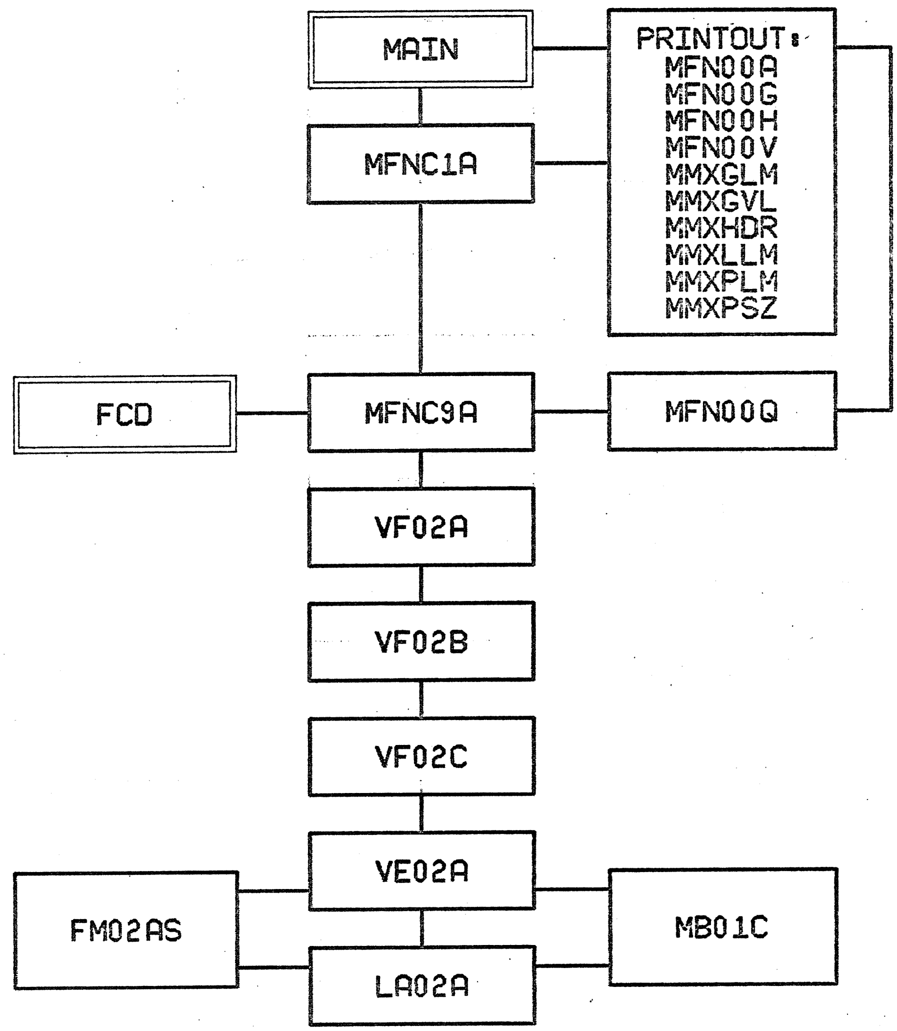

Fig. 1 Structure of the MFNC package corresponding to the standard entry (subroutine MFNC1A).

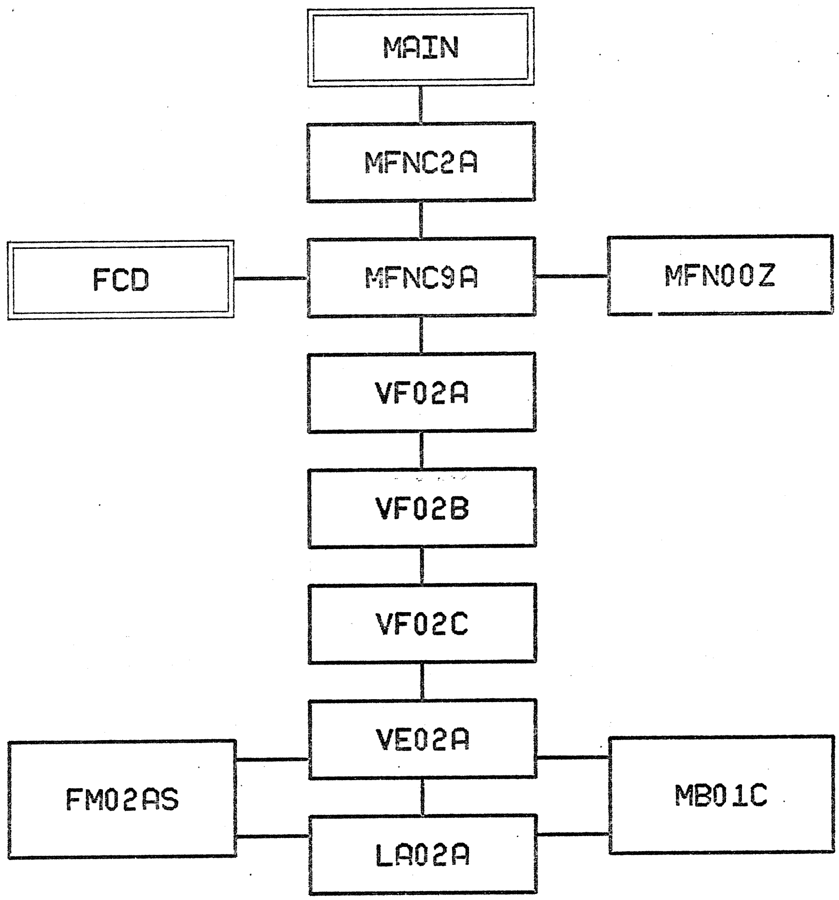

Fig. 2 Structure of the MFNC package corresponding to the basic entry (subroutine MFNC2A).

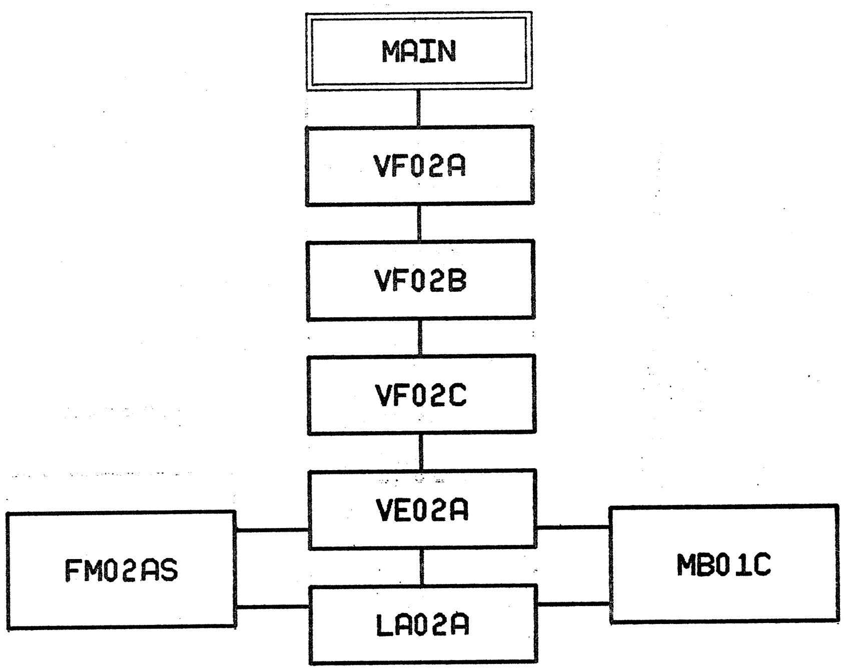

Fig. 3 Structure of the MFNC package corresponding to the original entry (subroutine VF02A).

The common part of the package is composed of subroutines VF02A, VF02B, VF02C, VE02A, LA02A, MB01C and FM02AS. VF02A subdivides the workspace (defined by the user) into a set of vectors and matrices used by the remaining subroutines and checks formal correctness of some parameters. VF02B controls the minimization procedure, implements the line search, calls VF02C for solution of quadratic subproblems, updates the approximation of the Hessian matrix, and checks the convergence of the algorithm. VF02C determines linear approximations of the constraint functions, calls VE02A to solve linearly constrained quadratic minimization, and calculates Lagrangian multipliers; it also checks whether the required feasibility conditions hold for the solution returned by VE02A. VE02A finds a minimum of the quadratic function subject to linear equality and inequality constraints using the method of Fletcher. The method requires an initial feasible point, and this is obtained by calling LA02A. MB01C is used for matrix inversion, and FM01AS for evaluation of the inner product of two real vectors.

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

When the standard entry (Fig. 1) is used, the subroutine MFNC1A and the set of subroutines PRINTOUT provide printed output containing principal input parameters of the problem to be solved, and the solution obtained by the package. Moreover, the subroutine MFN00Q outputs the values of functions and their derivatives according to the argument IPR in the call of MFNC1A.

For the standard entry (Fig. 1) and the basic entry (Fig. 2) the subroutine MFNC9A checks the formal correctness of input parameters, calls the user-defined subroutine FCD for evaluations required by the package, and sets the output parameters to the values corresponding to the solution found by the package.

IV. LIST OF ARGUMENTS

Standard entry (subroutine MFNC1A)

The subroutine call is

```
CALL MFNC1A (FCD,N,L,LEQ,X,EPS,MAXF,W,IW,ICH,IPR,IFLAG)
```

The arguments are as follows.

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

```
SUBROUTINE FCD(N,L,X,F,G,C,D,K)
DIMENSION X(N),G(N),C(L),D(K,L)
```

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

$G(J) = \partial F(\underset{\sim}{x})/\partial x_J$, J=1,...,N,

$C(I) = f_I(\underset{\sim}{x})$, I=1,...,L,

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

Note: The name FCD can be arbitrary (user's choice) and must appear in an EXTERNAL statement in the segment calling MFNC1A.

The user can terminate the iterative procedure and force the return from the package by setting to zero (in the subroutine FCD) the variable MARK in the common area MFN000

COMMON /MFN000/ MARK

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

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

L is an INTEGER argument which must be set to ℓ, the total number of equality and inequality constraints. Its value must be positive or zero and it is not changed by the package

LEQ is an INTEGER argument which must be set to ℓ_{eq}, the number of equality constraints. Its value must be positive or zero and not greater than L, and not greater than N. Its value is not changed by the package.

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

EPS is a REAL variable which on entry must be set to the required accuracy of the solution. The iteration terminates when the objective function is predicted to be within EPS of its final value and allowance is made for any constraint violation. If EPS is chosen too small, the iteration terminates when no better estimation of the solution can be obtained because of rounding errors.

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

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

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

IWR = 19+5*N*N+24*N+6*L+N*L+max(L,3*N+3).

The values of IWR for a set of initial values of arguments L 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.

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

$$|IPR| = pqrs$$

where q, r and s are the least significant one-digit fields, and p is the remaining part of the number. If q is not equal to zero (i.e. q=1,...,9) then the first q evaluations of functions (i.e., the first q calls of FCD) are reported in the printed output. Further, if p is not equal to zero then every

TABLE I

MINIMUM WORKSPACE FOR THE MFNC PACKAGE

L: \ N:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	61	104	157	220	293	376	469	572	685	808	941	1084	1237	1400	1573	1756	1949	2152	2365	2588
2	68	112	166	230	304	388	482	586	700	824	958	1102	1256	1420	1594	1778	1972	2176	2390	2614
3	75	120	175	240	315	400	495	600	715	840	975	1120	1275	1440	1615	1800	1995	2200	2415	2640
4	82	128	184	250	326	412	508	614	730	856	992	1138	1294	1460	1636	1822	2018	2224	2440	2666
5	89	136	193	260	337	424	521	628	745	872	1009	1156	1313	1480	1657	1844	2041	2248	2465	2692
6	96	144	202	270	348	436	534	642	760	888	1026	1174	1332	1500	1678	1866	2064	2272	2490	2718
7	104	152	211	280	359	448	547	656	775	904	1043	1192	1351	1520	1699	1888	2087	2296	2515	2744
8	112	160	220	290	370	460	560	670	790	920	1060	1210	1370	1540	1720	1910	2110	2320	2540	2770
9	120	168	229	300	381	472	573	684	805	936	1077	1228	1389	1560	1741	1932	2133	2344	2565	2796
10	128	177	238	310	392	484	586	698	820	952	1094	1246	1408	1580	1762	1954	2156	2368	2590	2822
11	136	186	247	320	403	496	599	712	835	968	1111	1264	1427	1600	1783	1976	2179	2392	2615	2848
12	144	195	256	330	414	508	612	726	850	984	1128	1282	1446	1620	1804	1998	2202	2416	2640	2874
13	152	204	266	340	425	520	625	740	865	1000	1145	1300	1465	1640	1825	2020	2225	2440	2665	2900
14	160	213	276	350	436	532	638	754	880	1016	1162	1318	1484	1660	1846	2042	2248	2464	2690	2926
15	168	222	286	360	447	544	651	768	895	1032	1179	1336	1503	1680	1867	2064	2271	2488	2715	2952
16	176	231	296	371	458	556	664	782	910	1048	1196	1354	1522	1700	1888	2086	2294	2512	2740	2978
17	184	240	306	382	469	568	677	796	925	1064	1213	1372	1541	1720	1909	2108	2317	2536	2765	3004
18	192	249	316	393	480	580	690	810	940	1080	1230	1390	1560	1740	1930	2130	2340	2560	2790	3030
19	200	258	326	404	492	592	703	824	955	1096	1247	1408	1579	1760	1951	2152	2363	2584	2815	3056
20	208	267	336	415	504	604	716	838	970	1112	1264	1426	1598	1780	1972	2174	2386	2608	2840	3082

TABLE I

MINIMUM WORKSPACE FOR THE MFNC PACKAGE

L: \ N:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	216	276	346	426	516	616	729	852	985	1128	1281	1444	1617	1800	1993	2196	2409	2632	2865	3108
22	224	285	356	437	528	629	742	866	1000	1144	1298	1462	1636	1820	2014	2218	2432	2656	2890	3134
23	232	294	366	448	540	642	755	880	1015	1160	1315	1480	1655	1840	2035	2240	2455	2680	2915	3160
24	240	303	376	459	552	655	768	894	1030	1176	1332	1498	1674	1860	2056	2262	2478	2704	2940	3186
25	248	312	386	470	564	668	782	908	1045	1192	1349	1516	1693	1880	2077	2284	2501	2728	2965	3212
26	256	321	396	481	576	681	796	922	1060	1208	1366	1534	1712	1900	2098	2306	2524	2752	2990	3238
27	264	330	406	492	588	694	810	936	1075	1224	1383	1552	1731	1920	2119	2328	2547	2776	3015	3264
28	272	339	416	503	600	707	824	951	1090	1240	1400	1570	1750	1940	2140	2350	2570	2800	3040	3290
29	280	348	426	514	612	720	838	966	1105	1256	1417	1588	1769	1960	2161	2372	2593	2824	3065	3316
30	288	357	436	525	624	733	852	981	1120	1272	1434	1606	1788	1980	2182	2394	2616	2848	3090	3342
31	296	366	446	536	636	746	866	996	1136	1288	1451	1624	1807	2000	2203	2416	2639	2872	3115	3368
32	304	375	456	547	648	759	880	1011	1152	1304	1468	1642	1826	2020	2224	2438	2662	2896	3140	3394
33	312	384	466	558	660	772	894	1026	1168	1320	1485	1660	1845	2040	2245	2460	2685	2920	3165	3420
34	320	393	476	569	672	785	908	1041	1184	1337	1502	1678	1864	2060	2266	2482	2708	2944	3190	3446
35	328	402	486	580	684	798	922	1056	1200	1354	1519	1696	1883	2080	2287	2504	2731	2968	3215	3472
36	336	411	496	591	696	811	936	1071	1216	1371	1536	1714	1902	2100	2308	2526	2754	2992	3240	3498
37	344	420	506	602	708	824	950	1086	1232	1388	1554	1732	1921	2120	2329	2548	2777	3016	3265	3524
38	352	429	516	613	720	837	964	1101	1248	1405	1572	1750	1940	2140	2350	2570	2800	3040	3290	3550
39	360	438	526	624	732	850	978	1116	1264	1422	1590	1768	1959	2160	2371	2592	2823	3064	3315	3576
40	368	447	536	635	744	863	992	1131	1280	1439	1608	1787	1978	2180	2392	2614	2846	3088	3340	3602

pth evaluation of 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 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 FCD 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 FCD is not reported by the package), then the "starting point" values of optimization variables $\underset{\sim}{x}^0$ and corresponding function values $\underset{\sim}{f}(\underset{\sim}{x}^0)$ are printed; if, at the same time, s is greater than zero, the values of partial derivatives are included in the "starting point" information. It should be noted that the values of partial derivatives can only be printed for those evaluations for which printing of function values is indicated.

If the value of IPR is negative, the partial derivatives calculated by FCD are verified numerically by comparing values supplied by FCD with the differences of 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. Partial derivatives of the objective function are indicated by the subscript equal to zero.

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

IFLAG = -4 artificial bound reached (usually because of unbounded solution),

IFLAG = -3 line search requires more than 5 steps (usually because of incorrect derivatives),

IFLAG = -2 feasible region is empty (usually because of inconsistent constraints),

IFLAG = -1 incorrect input arguments,

IFLAG = 0 required accuracy obtained,

IFLAG = 1 machine accuracy reached,

IFLAG = 2 limit of function evaluations reached,

IFLAG = 3 iteration terminated by the user.

Basic entry (subroutine MFNC2A)

The subroutine call is

```
CALL MFNC2A (FCD,N,L,LEQ,X,EPS,MAXF,W,IW,IFLAG)
```

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, however, diagnostic messages can be obtained by setting the variable LPR in the common area MFN111

```
COMMON /MFN111/ LPR
```

to the unit number of the output file (LPR has a preset value 0).

Original entry (subroutine VF02A)

The subroutine call is

CALL VF02A (N,L,LEQ,X,F,G,C,D,K,MAXF,EPS,IP,W,IW)

The arguments are described in the documentation of the subroutine VF02AD [3] from the Harwell Subroutine Library, however:

(1) the length IW of the workspace W must be at least

18+5*N*N+23*N+4*L+max(L,3*N+3),

(2) to obtain printed output, the variable LPR in the common area VF02D

COMMON /VF02D/ VLN,LPR

must be set to the unit number of the output file (LPR has a preset value 0); VLN controls the artificial bound and is preset to 10^6.

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 variable and the number of constraint 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 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 line. 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 /MFN000/ (see argument FCD), COMMON /MFN111/ (for basic entry), COMMON /MMX000/ (for standard entry), COMMON /VF02D/ COMMON /VF02E/ COMMON /VE02X/ COMMON /LA02B/ COMMON /MB01D/
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:	VF02A, VF02B, VF02C, VE02A, LA02A, MB01C, FM01AS and: a) for standard entry: MFNC1A, MFNC9A, MFN00Q, MFN00A, MFN00G, MFN00H, MFN00V, MMXPSZ, MMXPLM, MMXLLM, MMXHDR, MMXGLM, MMXGVL; b) for basic entry: MFNC2A, MFNC9A, MFN00Z.
Restrictions:	$N>0$, $L\geq 0$, $LEQ\geq 0$, $LEQ\leq L$, $LEQ\leq N$, $EPS\geq 0$, $MAXF>0$, $IW\geq IWR$.
Date:	May 1982.

VII. EXAMPLES

Example 1 [3]

Minimize

$$F(\underset{\sim}{x}) = x_1^2 + x_2^2 + x_3$$

subject to the constraints

$$x_1 x_2 - x_3 = 0 ,$$

$$x_3 - 1 \geq 0 .$$

For the starting point

$$\underset{\sim}{x}^0 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

and the required accuracy 10^{-7} the solution is obtained after 12 evaluations of the functions (as in [3]).

```
      PROGRAM TRMFN1 (OUTPUT,TAPE6=OUTPUT)
C
C   HARWELL TEST PROGRAM.
C
      DIMENSION X(3),T(3),W(166)
      EXTERNAL FCD
      DATA T/10HTRMFN1 : H,10HARWELL EXA,10HMPLE.      /
      CALL MMXHDR(3,T)
      X(1)=1.0
      X(2)=2.0
      X(3)=3.0
      N=3
      LEQ=1
      L=2
      MAXF=25
      EPS=1.0E-7
      ICH=6
      IPR=-10
      LW=166
      CALL MFNC1A(FCD,N,L,LEQ,X,EPS,MAXF,W,LW,ICH,IPR,IFLAG)
      STOP
      END
C
C
      SUBROUTINE FCD (N,M,X,F,G,C,D,K)
      DIMENSION X(N),G(N),C(M),D(K,M)
      X1=X(1)
      X2=X(2)
      X3=X(3)
      F=X1*X1+X2*X2+X3
      G(1)=X1+X1
      G(2)=X2+X2
      G(3)=1.0
      C(1)=X1*X2-X3
      D(1,1)=X2
      D(2,1)=X1
      D(3,1)=-1.0
      C(2)=X3-1.0
      D(1,2)=0.0
      D(2,2)=0.0
      D(3,2)=1.0
      RETURN
      END
```

DATE : 82/05/19. TIME : 15.07.22. PAGE : 1
MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE) (V:82.05)

TRMFN1 : HARWELL EXAMPLE.

INPUT DATA

NUMBER OF VARIABLES (N) 3

NUMBER OF EQUALITY CONSTRAINTS (LEQ) 1

TOTAL NUMBER OF CONSTRAINTS (L) 2

ACCURACY (EPS) 1.000E-07

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) 25

WORKING SPACE (IW) 166

PRINTOUT CONTROL (IPR) . . . . -10

STARTING POINT. OBJECTIVE FUNCTION : 8.000000000000E+00

	VARIABLES	GRADIENT		CONSTRAINTS
1	1.000000000000E+00	2.0000000000E+00	1	-1.000000000000E+00
2	2.000000000000E+00	4.0000000000E+00	2	2.000000000000E+00
3	3.000000000000E+00	1.0000000000E+00		

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

SOLUTION

OBJECTIVE FUNCTION : 2.999999999999E+00

	VARIABLES	GRADIENT		CONSTRAINTS
1	9.999999999993E-01	2.0000000000E+00	1	-4.618527782441E-13
2	1.000000000000E+00	2.0000000000E+00	2	0.
3	1.000000000000E+00	1.0000000000E+00		

TYPE OF SOLUTION (IFLAG) 0

NUMBER OF FUNCTION EVALUATIONS 12

NUMBER OF QUADRATIC ITERATIONS 10

EXECUTION TIME (IN SECONDS) 268

Example 2 [5, Example 3]

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

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

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

$$F(\underset{\sim}{x}) = (x_1+x_2)^2$$

subject to the nonlinear constraint

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

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

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

as in [5].

```
      PROGRAM TRMFN2 (OUTPUT,TAPE1=OUTPUT)
C
C  BRENT EXAMPLE.
C
      DIMENSION X(2),XX(4,2),T(3),W(104)
      EXTERNAL FFF
      DATA T/10HTRMFN2 : B,10HRENT EXAMP,10HLE        /
      DATA XX/2.0,-2.0,2.0,2.0,
     1        2.0,-2.0,0.0,1.0/
      CALL MMXHDR(3,T)
      N=2
      LEQ=1
      L=1
      DO 10 I=1,4
      X(1)=XX(I,1)
      X(2)=XX(I,2)
      EPS=1.E-6
      MAXF=25
      IW=104
      ICH=1
      IPR=-10
      CALL MFNC1A(FFF,N,L,LEQ,X,EPS,MAXF,W,IW,ICH,IPR,IFLAG)
   10 CONTINUE
      STOP
      END
C
C
      SUBROUTINE FFF(N,L,X,F,G,C,D,K)
      DIMENSION X(N),G(N),C(L),D(K,L)
      X1=X(1)
      X2=X(2)
      R1=X1-X2
      R2=(X1-2.0)**2+X2*X2
      R3=X1+X2
      R4=R3+R3
      F=R3*R3
      G(1)=R4
      G(2)=R4
      C(1)=R1*R2+3.0*X1+5.0*X2
      D(1,1)=R2+(R1+R1)*(X1-2.0)+3.0
      D(2,1)=-R2+R1*(X2+X2)+5.0
      RETURN
      END
```

```
DATE : 82/05/19.                    TIME : 14.53.31.                PAGE :  1
MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE)              (V:82.05)

TRMFN2 : BRENT EXAMPLE


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

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

  NUMBER OF EQUALITY CONSTRAINTS (LEQ). . . . . . . . . . . . . . . . . . .     1

  TOTAL NUMBER OF CONSTRAINTS (L) . . . . . . . . . . . . . . . . . . . . .     1

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

  MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) . . . . . . . . . . . . . . . .    25

  WORKING SPACE (IW) . . . . . . . . . . . . . . . . . . . . . . . . . . .   104

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

  STARTING POINT.                   OBJECTIVE FUNCTION :  1.600000000000E+01

              VARIABLES             GRADIENT                     CONSTRAINTS

        1  2.000000000000E+00   8.0000000000E+00          1  1.600000000000E+01
        2  2.000000000000E+00   8.0000000000E+00

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.


SOLUTION
--------
                                    OBJECTIVE FUNCTION :  1.186745702620E-08

              VARIABLES             GRADIENT                     CONSTRAINTS

        1 -1.082246213238E-04   2.1787571711E-04          1 -5.405507481288E-04
        2  2.171624798772E-04   2.1787571711E-04


  TYPE OF SOLUTION (IFLAG)  . . . . . . . . . . . . . . . . . . . . . . . .     0

  NUMBER OF FUNCTION EVALUATIONS  . . . . . . . . . . . . . . . . . . . . .     9

  NUMBER OF QUADRATIC ITERATIONS  . . . . . . . . . . . . . . . . . . . . .     8

  EXECUTION TIME (IN SECONDS) . . . . . . . . . . . . . . . . . . . . . .  .121
```

```
DATE : 82/05/19.                    TIME : 14.53.32.                        PAGE :   1
MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE)                      (V:82.05)

TRMFN2 : BRENT EXAMPLE


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

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

  NUMBER OF EQUALITY CONSTRAINTS (LEQ). . . . . . . . . . . . . . . . . . . .     1

  TOTAL NUMBER OF CONSTRAINTS (L) . . . . . . . . . . . . . . . . . . . . . .     1

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

  MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) . . . . . . . . . . . . . . . . .    25

  WORKING SPACE (IW)  . . . . . . . . . . . . . . . . . . . . . . . . . . .     104

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

  STARTING POINT.                   OBJECTIVE FUNCTION :  1.600000000000E+01

                VARIABLES             GRADIENT                   CONSTRAINTS

         1  -2.000000000000E+00  -8.0000000000E+00          1  -1.600000000000E+01
         2  -2.000000000000E+00  -8.0000000000E+00

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.


SOLUTION
--------
                                    OBJECTIVE FUNCTION :  4.270746189808E-12

                VARIABLES             GRADIENT                   CONSTRAINTS

         1  -2.117626035708E-05  -4.1331567547E-06          1  -1.291275529750E-04
         2   1.910968197971E-05  -4.1331567547E-06


  TYPE OF SOLUTION (IFLAG)  . . . . . . . . . . . . . . . . . . . . . . . . .     0

  NUMBER OF FUNCTION EVALUATIONS  . . . . . . . . . . . . . . . . . . . . . .     7

  NUMBER OF QUADRATIC ITERATIONS  . . . . . . . . . . . . . . . . . . . . . .     6

  EXECUTION TIME (IN SECONDS) . . . . . . . . . . . . . . . . . . . . . . .    .094
```

DATE : 82/05/19.   TIME : 14.56.16.   PAGE : 1
MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE)   (V:82.05)

TRMFN2 : BRENT EXAMPLE

INPUT DATA

NUMBER OF VARIABLES (N) 2

NUMBER OF EQUALITY CONSTRAINTS (LEQ) 1

TOTAL NUMBER OF CONSTRAINTS (L) 1

ACCURACY (EPS) 1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) 25

WORKING SPACE (IW) 104

PRINTOUT CONTROL (IPR) . . . . -10

STARTING POINT.   OBJECTIVE FUNCTION : 4.000000000000E+00

	VARIABLES	GRADIENT		CONSTRAINTS
1	2.000000000000E+00	4.0000000000E+00	1	6.000000000000E+00
2	0.	4.0000000000E+00		

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

SOLUTION

OBJECTIVE FUNCTION : 3.474781237047E-10

	VARIABLES	GRADIENT		CONSTRAINTS
1	6.499981749980E-06	-3.7281530210E-05	1	2.035830275945E-05
2	-2.514074685511E-05	-3.7281530210E-05		

TYPE OF SOLUTION (IFLAG) 0

NUMBER OF FUNCTION EVALUATIONS 6

NUMBER OF QUADRATIC ITERATIONS 6

EXECUTION TIME (IN SECONDS) 095

```
DATE : 82/05/19.                 TIME : 14.56.17.                 PAGE :  1
MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE)            (V:82.05)

TRMFN2 : BRENT EXAMPLE


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

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

  NUMBER OF EQUALITY CONSTRAINTS (LEQ). . . . . . . . . . . . . . . . . . .     1

  TOTAL NUMBER OF CONSTRAINTS (L) . . . . . . . . . . . . . . . . . . . . .     1

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

  MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) . . . . . . . . . . . . . . . .    25

  WORKING SPACE (IW) . . . . . . . . . . . . . . . . . . . . . . . . . . .    104

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

  STARTING POINT.                      OBJECTIVE FUNCTION :  9.000000000000E+00

                VARIABLES              GRADIENT                  CONSTRAINTS

       1   2.000000000000E+00   6.0000000000E+00       1   1.200000000000E+01
       2   1.000000000000E+00   6.0000000000E+00

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.


SOLUTION
--------
                                       OBJECTIVE FUNCTION :  2.811748620719E-07

                VARIABLES              GRADIENT                  CONSTRAINTS

       1  -2.157351558771E-04   1.0605184809E-03       1  -7.649821902926E-04
       2   7.459943963171E-04   1.0605184809E-03

  TYPE OF SOLUTION (IFLAG)  . . . . . . . . . . . . . . . . . . . . . . . .     0

  NUMBER OF FUNCTION EVALUATIONS  . . . . . . . . . . . . . . . . . . . . .     6

  NUMBER OF QUADRATIC ITERATIONS  . . . . . . . . . . . . . . . . . . . . .     6

  EXECUTION TIME (IN SECONDS) . . . . . . . . . . . . . . . . . . . . . .   .087
```

Example 3 [7, Example 4]

This is the Rosen-Suzuki constrained minimization problem [8], slighly modified as indicated below. It is to minimize

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

subject to constraints

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

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

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

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

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

Two solutions are shown, which correspond to starting points $\underset{\sim}{x}^0 = [2\ 2\ 5\ 0]^T$ and $\underset{\sim}{x}^0 = \underset{\sim}{0}$, as in [7]. Both the solutions require slightly different numbers of function evaluations. The differences, however, are not significant.

```
      PROGRAM TRMFN3 (OUTPUT,TAPE6=OUTPUT)
C
C  ROSEN-SUZUKI PROBLEM.
C
      DIMENSION X(4),XX(4,2),T(3),W(300)
      EXTERNAL FCD
      DATA T/10HTRMFN3 : R,10HOSEN-SUZUK,10HI PROBLEM /
      DATA XX/2.0,2.0,5.0,0.0,0.0,0.0,0.0,0.0/
      CALL MMXHDR(3,T)
      N=4
      LEQ=0
      L=3
      DO 10 II=1,2
      DO 20 JJ=1,4
   20 X(JJ)=XX(JJ,II)
      MAXF=30
      EPS=1.E-6
      LW=300
      ICH=6
      IPR=-10
      CALL MFNC1A(FCD,N,L,LEQ,X,EPS,MAXF,W,LW,ICH,IPR,IFLAG)
   10 CONTINUE
      STOP
      END
C
C
      SUBROUTINE FCD(N,L,X,F,G,C,D,K)
      DIMENSION X(N),G(N),C(L),D(K,L)
      X1=X(1)
      X2=X(2)
      X3=X(3)
      X4=X(4)
      R1=X1+X1+1.0
      R2=X2+X2-1.0
      R3=X3+X3
      R4=X4+X4-1.0
      F=X1*(X1-5.0)+X2*(X2-5.0)+X3*(R3-21.0)+X4*(X4+7.0)
      G(1)=R1-6.0
      G(2)=R2-4.0
      G(3)=4.0*X3-21.0
      G(4)=X4+X4+7.0
      C(1)=X1*(-X1-1.0)+X2*(1.0-X2)+X3*(-X3-1.0)+X4*(1.0-X4)+8.0
      C(2)=X1*(1.0-X1)-X2*(X2+X2)-X3*X3+X4*(1.0-X4-X4)+10.0
      C(3)=X1*(-X1-2.0)+X2*(1.0-X2)-X3*X3+X4+5.0
      D(1,1)=-R1
      D(2,1)=-R2
      D(3,1)=-R3-1.0
      D(4,1)=-R4
      D(1,2)=-R1+2.0
      D(2,2)=-4.0*X2
      D(3,2)=-R3
      D(4,2)=1.0-4.0*X4
      D(1,3)=-R1-1.0
      D(2,3)=-R2
      D(3,3)=-R3
      D(4,3)=1.0
      RETURN
      END
```

DATE : 82/06/29 TIME : 13.07.19 PAGE : 1
MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE) (V:82.05)

TRMFN3 : ROSEN-SUZUKI PROBLEM

INPUT DATA

NUMBER OF VARIABLES (N) 4

NUMBER OF EQUALITY CONSTRAINTS (LEQ) 0

TOTAL NUMBER OF CONSTRAINTS (L) 3

ACCURACY (EPS) 1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) 30

WORKING SPACE (IW) 300

PRINTOUT CONTROL (IPR) . . . . -10

STARTING POINT. OBJECTIVE FUNCTION : -6.700000000000E+01

	VARIABLES	GRADIENT		CONSTRAINTS
1	2.000000000000E+00	-1.0000000000E+00	1	-3.000000000000E+01
2	2.000000000000E+00	-1.0000000000E+00	2	-2.500000000000E+01
3	5.000000000000E+00	-1.0000000000E+00	3	-3.000000000000E+01
4	0.	7.0000000000E+00		

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

SOLUTION

OBJECTIVE FUNCTION : -4.400000001041E+01

	VARIABLES	GRADIENT		CONSTRAINTS
1	-4.667889315509E-08	-5.0000000934E+00	1	-3.596164868861E-09
2	1.000001144072E+00	-2.9999977119E+00	2	9.999959321476E-01
3	1.999999707412E+00	-1.3000001170E+01	3	-3.411145144128E-09
4	-1.000000123047E+00	4.9999997539E+00		

TYPE OF SOLUTION (IFLAG) 0

NUMBER OF FUNCTION EVALUATIONS 16

NUMBER OF QUADRATIC ITERATIONS 11

EXECUTION TIME (IN SECONDS) 485

DATE : 82/06/29 TIME : 13.07.20 PAGE : 1
MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE) (V:82.05)

TRMFN3 : ROSEN-SUZUKI PROBLEM

INPUT DATA

NUMBER OF VARIABLES (N) 4

NUMBER OF EQUALITY CONSTRAINTS (LEQ). 0

TOTAL NUMBER OF CONSTRAINTS (L) 3

ACCURACY (EPS) 1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) 30

WORKING SPACE (IW) 300

PRINTOUT CONTROL (IPR) . . . . -10

STARTING POINT. OBJECTIVE FUNCTION : 0.

	VARIABLES	GRADIENT		CONSTRAINTS
1	0.	-5.0000000000E+00	1	8.000000000000E+00
2	0.	-5.0000000000E+00	2	1.000000000000E+01
3	0.	-2.1000000000E+01	3	5.000000000000E+00
4	0.	7.0000000000E+00		

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

SOLUTION

OBJECTIVE FUNCTION : -4.400000000131E+01

	VARIABLES	GRADIENT		CONSTRAINTS
1	-1.107425820962E-06	-5.0000022149E+00	1	-5.595527892402E-10
2	1.000000404734E+00	-2.9999991905E+00	2	9.999990285584E-01
3	2.000000675463E+00	-1.2999997298E+01	3	-3.801403636317E-10
4	-9.999991086450E-01	5.0000017827E+00		

TYPE OF SOLUTION (IFLAG) 0

NUMBER OF FUNCTION EVALUATIONS 12

NUMBER OF QUADRATIC ITERATIONS 10

EXECUTION TIME (IN SECONDS) 424

<u>Example 4</u> [9, Example 5.5]

This is Colville's test problem 2 [10]. It is to minimize the objective function

$$F(\underset{\sim}{x}) = -\sum_{1\le i\le 10} b_i\, x_{5+i} + \sum_{1\le i\le 5}\ \sum_{1\le j\le 5} c_{ij}x_i x_j + 2\sum_{1\le j\le 5} d_j x_j^3$$

subject to the constraints

$$x_i \ge 0\ , \qquad i=1,\dots,15,$$

$$\sum_{1\le i\le 10} a_{ij}x_{5+i} \le e_j + 2\sum_{1\le i\le 5} c_{ij}x_i + 3d_j x_j^2,\ j=1,\dots,5,$$

where a_{ij}, b_i, c_{ij}, d_j, e_j are as follows:

a_{ij}	j = 1	2	3	4	5	b_i
i = 1	-16	2	0	1	0	-40
2	0	-2	0	0.4	2	-2
3	-3.5	0	2	0	0	-.25
4	0	-2	0	-4	-1	-4
5	0	-9	-2	1	-2.8	-4
6	2	0	-4	0	0	-1
7	-1	-1	-1	-1	-1	-40
8	-1	-2	-3	-2	-1	-60
9	1	2	3	4	5	5
10	1	1	1	1	1	1

c_{ij}		j=1	2	3	4	5
	1	30	-20	-10	32	-10
	2	-20	39	-6	-31	32
i	3	-10	-6	10	-6	-10
	4	32	-31	-6	39	-20
	5	-10	32	-10	-20	30
d_j		4	8	10	6	2
e_j		-15	-27	-36	-18	-12

The solution is $F(\underset{\sim}{x}^*) = 32.34868$, and it is obtained for the starting point $\underset{\sim}{x}^0$ where $x_i = 0.0001$, $i \neq 12$, and $x_{12} = 60.0$ (as in [9]), and for the accuracy 10^{-6} after 16 iterations (as in [1]).

```
      PROGRAM TRMFN4 (OUTPUT,TAPE6=OUTPUT)
C
C   COLVILLE TEST PROBLEM 2.
C
      DIMENSION X(15),T(4),W(2000)
      EXTERNAL FCD
      DATA T/10HTRMFN4 : C,10HOLVILLE TE,10HST PROBLEM,2H 2/
      CALL MMXHDR(4,T)
      DO 10 I=1,15
   10 X(I)=0.0001
      X(12)=60.0
      N=15
      LEQ=0
      L=20
      MAXF=50
      EPS=1.0E-6
      ICH=6
      IPR=0
      LW=2000
      CALL MFNC1A(FCD,N,L,LEQ,X,EPS,MAXF,W,LW,ICH,IPR,IFLAG)
      STOP
      END
C
C
      SUBROUTINE FCD(N,L,X,F,G,C,D,KK)
      DIMENSION X(N),G(N),C(L),D(KK,L)
      DIMENSION A(10,5),B(10),C1(5,5),D1(5),E(5)
      DATA A/-16.0,0.0,-3.5,0.0,0.0,2.0,-1.0,-1.0,1.0,1.0,
     +       2.0,-2.0,0.0,-2.0,-9.0,0.0,-1.0,-2.0,2.0,1.0,
     +       0.0,0.0,2.0,0.0,-2.0,-4.0,-1.0,-3.0,3.0,1.0,
     +       1.0,0.4,0.0,-4.0,1.0,0.0,-1.0,-2.0,4.0,1.0,
     +       0.0,2.0,0.0,-1.0,-2.8,0.0,-1.0,-1.0,5.0,1.0/
      DATA B/-40.0,-2.0,-0.25,-4.0,-4.0,-1.0,-40.0,-60.0,5.0,1.0/
      DATA C1/30.0,-20.0,-10.0,32.0,-10.0,
     +        -20.0,39.0,-6.0,-31.0,32.0,
     +        -10.0,-6.0,10.0,-6.0,-10.0,
     +        32.0,-31.0,-6.0,39.0,-20.0,
     +        -10.0,32.0,-10.0,-20.0,30.0/
      DATA D1/4.0,8.0,10.0,6.0,2.0/
      DATA E/-15.0,-27.0,-36.0,-18.0,-12.0/
C
C     OBJECTIVE FUNCTION F
C
      F=0.0
      DO 10 I=1,10
      J=5+I
   10 F=F-B(I)*X(J)
      DO 30 I=1,5
      DO 20 J=1,5
   20 F=F+C1(I,J)*X(I)*X(J)
   30 F=F+2.0*D1(I)*X(I)**3
C
C     GRADIENT G OF THE OBJECTIVE FUNCTION
C
      DO 40 I=1,10
      J=5+I
   40 G(J)=-B(I)
      DO 60 I=1,5
      G(I)=6.0*D1(I)*X(I)*X(I)
      DO 60 J=1,5
   60 G(I)=G(I)+X(J)*(C1(I,J)+C1(J,I))
C
C     CONSTRAINTS C
C
      DO 70 I=1,15
```

```
   70 C(I)=X(I)
      DO 90 J=1,5
      K=J+15
      C(K)=E(J)+3.0*D1(J)*X(J)*X(J)
      DO 80 I=1,5
   80 C(K)=C(K)+2.0*C1(I,J)*X(I)
      DO 85 II=1,10
      JJ=5+II
   85 C(K)=C(K)-A(II,J)*X(JJ)
   90 CONTINUE
C
C     DERIVATIVES D OF THE CONSTRAINTS
C
      DO 100 I=1,15
  100 D(I,I)=1.0
      DO 110 I=1,15
      DO 110 J=1,15
      IF(I.NE.J) D(I,J)=0.0
  110 CONTINUE
      DO 140 J=1,5
      K=15+J
      DO 120 I=1,5
      D(I,K)=2.0*C1(I,J)
      IF(I.EQ.J) D(I,K)=D(I,K)+6.0*D1(J)*X(J)
  120 CONTINUE
      DO 130 I=1,10
      II=5+I
  130 D(II,K)=-A(I,J)
  140 CONTINUE
      RETURN
      END
```

DATE : 82/06/28. TIME : 15.11.05. PAGE : 1
MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE) (V:82.05)

TRMFN4 : COLVILLE TEST PROBLEM 2

INPUT DATA

NUMBER OF VARIABLES (N) 15

NUMBER OF EQUALITY CONSTRAINTS (LEQ) 0

TOTAL NUMBER OF CONSTRAINTS (L) 20

ACCURACY (EPS) 1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) 50

WORKING SPACE (IW) 2000

PRINTOUT CONTROL (IPR) 0

STARTING POINT. OBJECTIVE FUNCTION : 2.400010525500E+03

	VARIABLES	GRADIENT		CONSTRAINTS
1	1.000000000000E-04	4.4002400000E-03	1	1.000000000000E-04
2	1.000000000000E-04	2.8004800000E-03	2	1.000000000000E-04
3	1.000000000000E-04	-4.3994000000E-03	3	1.000000000000E-04
4	1.000000000000E-04	2.8003600000E-03	4	1.000000000000E-04
5	1.000000000000E-04	4.4001200000E-03	5	1.000000000000E-04
6	1.000000000000E-04	4.0000000000E+01	6	1.000000000000E-04
7	1.000000000000E-04	2.0000000000E+00	7	1.000000000000E-04
8	1.000000000000E-04	2.5000000000E-01	8	1.000000000000E-04
9	1.000000000000E-04	4.0000000000E+00	9	1.000000000000E-04
10	1.000000000000E-04	4.0000000000E+00	10	1.000000000000E-04
11	1.000000000000E-04	1.0000000000E+00	11	1.000000000000E-04
12	6.000000000000E+01	4.0000000000E+01	12	6.000000000000E+01
13	1.000000000000E-04	6.0000000000E+01	13	1.000000000000E-04
14	1.000000000000E-04	-5.0000000000E+00	14	1.000000000000E-04
15	1.000000000000E-04	-1.0000000000E+00	15	1.000000000000E-04
			16	4.500605012000E+01
			17	3.300380024000E+01
			18	2.399590030000E+01
			19	4.200266018000E+01
			20	4.800408006000E+01

SOLUTION

OBJECTIVE FUNCTION : 3.234867906597E+01

	VARIABLES	GRADIENT		CONSTRAINTS
1	2.999918085990E-01	2.1753848156E+01	1	2.999918085990E-01
2	3.334635341015E-01	2.3265889599E+00	2	3.334635341015E-01
3	3.999890835345E-01	-2.0212949897E+00	3	3.999890835345E-01
4	4.283149306028E-01	2.4779014835E+01	4	4.283149306028E-01
5	2.239687464390E-01	4.2495198946E+00	5	2.239687464390E-01
6	1.399403725291E-16	4.0000000000E+01	6	1.399403725291E-16
7	1.494513241190E-14	2.0000000000E+00	7	1.494513241190E-14

DATE : 82/06/28 TIME : 15.11.05 PAGE : 2
MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE) (V:82.05)

TRMFN4 : COLVILLE TEST PROBLEM 2

8	5.174131652324E+00	2.5000000000E-01	8	5.174131652324E+00
9	0.	4.0000000000E+00	9	0.
10	3.061114284129E+00	4.0000000000E+00	10	3.061114284129E+00
11	1.183971715885E+01	1.0000000000E+00	11	1.183971715885E+01
12	0.	4.0000000000E+01	12	0.
13	0.	6.0000000000E+01	13	0.
14	1.039335785344E-01	-5.0000000000E+00	14	1.039335785344E-01
15	4.635547259249E-15	-1.0000000000E+00	15	4.635547259249E-15
			16	2.061274154105E-08
			17	7.416374783705E-08
			18	1.652826842332E-07
			19	8.141103620909E-10
			20	1.229539830694E-09

TYPE OF SOLUTION (IFLAG) 0

NUMBER OF FUNCTION EVALUATIONS 16

NUMBER OF QUADRATIC ITERATIONS 16

EXECUTION TIME (IN SECONDS) 12.129

<u>Example 5</u> [11, Example 5]

The problem is to determine an optimally centered point $\underset{\sim}{x}^* = [x_1^* \; x_2^*]^T$ that maximizes the relative tolerance r in the region R_c defined by the inequalities

$$2 + 2x_1 - x_2 \geq 0 ,$$

$$143 - 11x_1 - 13x_2 \geq 0 ,$$

$$-60 + 4x_1 + 15x_2 \geq 0 ,$$

i.e., to find a point $\underset{\sim}{x}^*$ and a tolerance r such that the tolerance region R_ε

$$R_\varepsilon = \{ \underset{\sim}{x} \mid (1-r) \; x_i^* \leq x_i \leq (1+r) x_i^*, \; i=1,2 \}$$

is in the constraint region R_c and is as large as possible.

It can be shown [12] that if the constraint region R_c is one-dimensionally convex (and it is in this case) then it is sufficient that all vertices of R_ε belong to R_c to guarantee that the whole tolerance region R_ε is in the constraint region R_c.

It is convenient to assume that the tolerance r is an additional optimization variable (say x_3) and then the vertices of the tolerance region R_ε are described by the nonlinear expressions

$$[(1 \pm x_3) \; x_1^* \quad (1 \pm x_3) \; x_2^*]^T .$$

Since x_3 is to be maximized, the objective function can take the form

$$F(\underset{\sim}{x}) = -x_3$$

and it is to be minimized subject to the constraints

$$2 + 2(1 \pm x_3)x_1 - (1 \pm x_3)x_2 \geq 0 \ ,$$

$$143 - 11(1 \pm x_3)x_1 - 13(1 \pm x_3)x_2 \geq 0 \ ,$$

$$-60 + 4(1 \pm x_3)x_1 + 15(1 \pm x_3)x_2 \geq 0 \ ,$$

$$x_3 \geq 0 \ .$$

It should be observed that due to $x_3 \geq 0$ the first 3 constraints (and, in fact, 12 constraints) can be simplified to the form

$$2 + 2(1-x_3)x_1 - (1+x_3)x_2 \geq 0 \ ,$$

$$143 - 11(1+x_3)x_1 - 13(1+x_3)x_2 \geq 0 \ ,$$

$$-60 + 4(1-x_3)x_1 + 15(1-x_3)x_2 \geq 0 \ .$$

The solution is shown for the starting point $\underset{\sim}{x}^0 = \underset{\sim}{0}$. The resulting relative tolerance r is equal to 0.3414 or 34.1% (as in [11]).

```
      PROGRAM TRMFN5 (OUTPUT,TAPE6=OUTPUT)
C
C   TOLERANCING EXAMPLE.
C
      DIMENSION X(3),T(3),W(184)
      EXTERNAL FCD
      DATA T/10HTRMFN5 : T,10HOLERANCING,10H EXAMPLE  /
      CALL MMXHDR(3,T)
      X(1)=0.0
      X(2)=0.0
      X(3)=0.0
      N=3
      LEQ=0
      L=4
      MAXF=25
      EPS=1.0E-6
      ICH=6
      IPR=-10
      LW=184
      CALL MFNC1A(FCD,N,L,LEQ,X,EPS,MAXF,W,LW,ICH,IPR,IFLAG)
      STOP
      END
C
C
      SUBROUTINE FCD(N,L,X,F,G,C,D,K)
      DIMENSION X(N),G(N),C(L),D(K,L)
      X1=X(1)
      X2=X(2)
      X3=X(3)
      R1=1.0-X3
      R2=1.0+X3
      F=-X3
      G(1)=0.0
      G(2)=0.0
      G(3)=-1.0
      C(1)=2.0*(1.0+X1*R1)-X2*R2
      D(1,1)=R1+R1
      D(2,1)=-R2
      D(3,1)=-X1-X1-X2
      C(2)=143.0-R2*(11.0*X1+13.0*X2)
      D(1,2)=-11.0*R2
      D(2,2)=-13.0*R2
      D(3,2)=-11.0*X1-13.0*X2
      C(3)=-60.0+R1*(4.0*X1+15.0*X2)
      D(1,3)=4.0*R1
      D(2,3)=15.0*R1
      D(3,3)=-4.0*X1-15.0*X2
      C(4)=X3
      D(1,4)=0.0
      D(2,4)=0.0
      D(3,4)=1.0
      RETURN
      END
```

```
DATE : 82/06/28.                    TIME : 15.17.11.                        PAGE :  1
MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE)                      (V:82.05)

TRMFN5 : TOLERANCING EXAMPLE


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

  NUMBER OF VARIABLES (N) . . . . . . . . . . . . . . . . . . . . . . . . . .     3

  NUMBER OF EQUALITY CONSTRAINTS (LEQ). . . . . . . . . . . . . . . . . . . .     0

  TOTAL NUMBER OF CONSTRAINTS (L) . . . . . . . . . . . . . . . . . . . . . .     4

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

  MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) . . . . . . . . . . . . . . . . .    25

  WORKING SPACE (IW) . . . . . . . . . . . . . . . . . . . . . . . . . . . .   184

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

  STARTING POINT.                      OBJECTIVE FUNCTION :  0.

                VARIABLES            GRADIENT                    CONSTRAINTS

       1   0.                    0.                        1   2.000000000000E+00
       2   0.                    0.                        2   1.430000000000E+02
       3   0.                   -1.0000000000E+00          3  -6.000000000000E+01
                                                           4   0.

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

SOLUTION
--------
                                       OBJECTIVE FUNCTION : -3.414065195318E-01

                VARIABLES            GRADIENT                    CONSTRAINTS

       1   3.670138928676E+00    0.                        1  -3.183515673300E-10
       2   5.094845628439E+00    0.                        2   2.412889443804E-09
       3   3.414065195318E-01   -1.0000000000E+00          3   6.580421541003E-09
                                                           4   3.414065195318E-01

  TYPE OF SOLUTION (IFLAG) . . . . . . . . . . . . . . . . . . . . . . . . . .     0

  NUMBER OF FUNCTION EVALUATIONS  . . . . . . . . . . . . . . . . . . . . . .    12

  NUMBER OF QUADRATIC ITERATIONS  . . . . . . . . . . . . . . . . . . . . . .    11

  EXECUTION TIME (IN SECONDS) . . . . . . . . . . . . . . . . . . . . . . . .  .376
```

VIII. REFERENCES

[1] M.J.D. Powell, "A fast algorithm for nonlinearly constrained optimization calculations", in Numerical Analysis, Proc. Biennial Conf. (Dundee, Scotland, 1977), Lecture Notes in Mathematics 630, G.A. Watson, Ed. Berlin: Springer-Verlag, 1978, pp. 144-157.

[2] S.P. Han, "Superlinearly convergent variable metric algorithms for general nonlinear programming problem", Mathematical Programming, vol. 11, 1976, pp. 263-282.

[3] VF02AD subroutine specification, Harwell Subroutine Library, AERE, Harwell, Oxfordshire, England, February 1978.

[4] J.W. Bandler, "Nonlinear programming using Lagrangian functions", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-284, 1981.

[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.W. Bandler and W.M. Zuberek, "MMUM - a Fortran package for unconstrained minimax optimization", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-291, 1982.

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

[9] C. Charalambous and A.R. Conn, "An efficient method to solve the minimax problem directly", SIAM J. Numerical Analysis, vol. 15, 1978, pp. 162-187.

[10] A.R. Colville, "A comparative study of nonlinear programming codes", IBM Scientific Center, Yorktown Heights, New York, Report 320-2949, 1968.

[11] J.W. Bandler and W.M. Zuberek, "MMLC - a Fortran package for linearly constrained minimax optimization", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-292, 1982.

[12] J.W. Bandler, "Optimization of design tolerances using nonlinear programming", J. Optimization Theory and Applications, vol. 14, 1974, pp. 99-114.

APPENDIX

LISTING OF THE MFNC PACKAGE

```
      SUBROUTINE MFNC1A (FCD,N,L,LEQ,X,EPS,MAXF,W,IW,LCH,IPR,IFLAG)
      EXTERNAL FCD,MFN00Q
C
C          LEVEL 1 INTERFACE (STANDARD ENTRY)
C
      DIMENSION X(1), W(1)
      COMMON /MMX000/ NCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      COMMON /VF02E/ DUMMY(6),IDUMMY,ITERQ,JDUMMY(3)
      NCH=LCH
      IF (LCH.LE.0) GO TO 40
      I=IABS(IPR)
      J=I/10
      LG2=MOD(I,10)
      I=J/10
      LG1=MOD(J,10)
      J=I/10
      LV2=MOD(I,10)
      LV1=J
      LG1=LG1*LV1
      NRP=0
      CALL MMXPSZ (-1)
      CALL MMXPLM (-1)
      CALL MMXLLM (-1)
      CALL MMXHDR (-1,H)
      CALL MMXGLM (-1,-1)
      CALL MMXGVL (-1)
      IF (MXL.NE.0) LML=MXL*LMP+100
      IF (MXL.EQ.0) MXL=LML+100
      CALL DATE (DAT)
      CALL TIME (TIM)
      CALL MFN00A
      WRITE (LCH,10) N,LEQ,L,EPS,MAXF,IW,IPR
   10 FORMAT (11H0INPUT DATA/11H ----------//
     1 27H    NUMBER OF VARIABLES (N),25(2H. ),I4//
     2 39H    NUMBER OF EQUALITY CONSTRAINTS (LEQ),19(2H. ),I4//
     3 35H    TOTAL NUMBER OF CONSTRAINTS (L) ,21(2H. ),I4//
     4 19H    ACCURACY (EPS)  ,26(2H. ),1PE10.3//
     5 45H    MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) ,16(2H. ),I4//
     6 22H    WORKING SPACE (IW) ,26(2H. ),1H.,I6//
     7 26H    PRINTOUT CONTROL (IPR) ,24(2H. ),1H.,I6/)
      NRL=NRL-18
      LML=LML-18
      JF=1
      JC=JF+1
      JG=JC+L
      JD=JG+N
      IF (LV2.NE.0.OR.LV1.EQ.1) GO TO 30
      CALL FCD (N,L,X,W(JF),W(JG),W(JC),W(JD),N)
      WRITE (LCH,20) W(JF)
   20 FORMAT (19H    STARTING POINT.  ,20X,21HOBJECTIVE FUNCTION :  ,
     1 1PE19.12)
      NRL=NRL-1
      LML=LML-1
      CALL MFN00V (X,N,W(JG),W(JC),L)
      IF (LG2.NE.0) CALL MFN00G (W(JD),N,L,N)
   30 IF (IPR.GE.0) GO TO 40
      L1=L+1
      JF1=JD+N*L
      JG1=JF1+L1
      JG2=JG1+L1
      LW=2*(N*L+N+L+1)
      IF (LW.GT.IW) GO TO 80
      CALL MFN00H (FCD,N,L,X,W(JF),W(JG),W(JF1),W(JG1),W(JG2),L1)
   40 CALL SECOND (TBEG)
```

```
      CALL MFNC9A (MFN00Q,FCD,N,L,LEQ,X,EPS,MAXF,W,IW,IFLAG)
      CALL SECOND (TEND)
      IF (LCH.LE.0) RETURN
      IF (IFLAG.EQ.-1) GO TO 100
      IF (NRL.LT.9) CALL MFN00A
      WRITE (LCH,50)
   50 FORMAT (//9H SOLUTION/9H --------)
      WRITE (LCH,60) W(1)
   60 FORMAT (39X,21HOBJECTIVE FUNCTION : ,1PE19.12/)
      NRL=NRL-6
      LML=LML-6
      CALL MFN00V (X,N,W(L+2),W(2),L)
      CPU=TEND-TBEG
      IF (NRL.LT.9) CALL MFN00A
      WRITE (LCH,70) IFLAG,MAXF,ITERQ,CPU
   70 FORMAT (//29H   TYPE OF SOLUTION (IFLAG)   ,24(2H. ),I4//
     1 35H   NUMBER OF FUNCTION EVALUATIONS   ,21(2H. ),I4//
     2 35H   NUMBER OF QUADRATIC ITERATIONS   ,21(2H. ),I4//
     3 31H   EXECUTION TIME (IN SECONDS) ,21(2H. ),1H.,F7.3/)
      RETURN
   80 IFLAG=-1
      WRITE (LCH,90) LW
   90 FORMAT(/55H0INSUFFICIENT WORKSPACE. IT SHOULD HAVE LENGTH AT LEAST
     1 ,I6/)
  100 WRITE (LCH,110)
  110 FORMAT (///40H I N C O R R E C T   P A R A M E T E R S/)
      RETURN
      END
C
C
      SUBROUTINE MFNC2A (FCD,N,L,LEQ,X,EPS,MAXF,W,IW,IFLAG)
      EXTERNAL FCD,MFN00Z
C
C          LEVEL 2 INTERFACE (BASIC ENTRY)
C
      DIMENSION X(1), W(1)
      CALL MFNC9A (MFN00Z,FCD,N,L,LEQ,X,EPS,MAXF,W,IW,IFLAG)
      RETURN
      END
C
C
      SUBROUTINE MFNC9A (FQQ,FCD,N,M,LEQ,X,EPS,MAXF,W,LW,IFLAG)
      DIMENSION X(1), W(1)
      EXTERNAL FQQ,FCD
      DIMENSION IEE(8)
      COMMON /LA02B/ LP1,IE1
      COMMON /MB01D/ LP2,IE2
      COMMON /VF02D/ VLN,LP3
      COMMON /VE02X/ LP4
      COMMON /MFN000/ MARK
      COMMON /MFN111/ LP0
      DATA LP0/0/,IEE/0,2,-3,1,-2,-4,-1,-1/
      DATA XZERO/0.0/
      MARK=1
      LP1=LP0
      LP2=LP0
      LP3=LP0
      LP4=LP0
      LWR=5*N*N+24*N+6*M+N*M+19+MAX0(M,3*N+3)
      IF (N.LE.0.OR.LEQ.GT.N.OR.LEQ.GT.M.OR.M.LT.0.OR.LEQ.LT.0.OR.EPS.LT
     1.XZERO.OR.MAXF.LE.0) GO TO 20
      IF (LWR.LE.LW) GO TO 30
      IF (LP0.GT.0) WRITE (LP0,10) LWR
   10 FORMAT(/55H0INSUFFICIENT WORKSPACE. IT SHOULD HAVE LENGTH AT LEAST
     1 ,I6/)
```

```
   20 IFLAG=-1
      RETURN
   30 IF=1
      IG=2
      IC=IG+M
      ID=IC+N
      IW=ID+N*M+M
      LL=LW-IW+1
      IE=-1
      NCALL=0
   40 CALL FCD (N,M,X,W(IF),W(IG),W(IC),W(ID),N+1)
      NCALL=NCALL+1
      CALL FQQ (N,M,X,W(IF),W(IG),W(IC),W(ID),N+1,NCALL)
      IF (MARK.EQ.0) GO TO 50
      CALL VF02A (N,M,LEQ,X,W(IF),W(IG),W(IC),W(ID),N+1,MAXF,EPS,0,IE,W(
     1IW),LL)
      IF (IE.EQ.0) GO TO 40
      IFLAG=IEE(IE)
      MAXF=NCALL
      RETURN
   50 IFLAG=3
      MAXF=NCALL
      RETURN
      END
C
C
      SUBROUTINE MFN00Z (N,M,X,F,G,C,D,L,K)
C
C     DUMMY SUBROUTINE WHICH FOR BASIC ENTRY SUBSTITUTES SUBROUTINE
C     MFN00Q/11Q.
C
      DIMENSION X(N), G(N), C(M), D(L,M)
      RETURN
      END
C
C
      SUBROUTINE MFN00Q (N,M,X,F,G,C,D,L,K)
C
C     PRINT RESULTS OF FUNCTION EVALUATION.
C
      DIMENSION X(N), G(N), C(M), D(L,M)
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      IF (LCH.LE.0) RETURN
      IF (LV1+LV2.EQ.0) RETURN
      IF (K.LE.LV2) GO TO 10
      IF (LV1.EQ.0) RETURN
      IF (MOD(K,LV1).NE.0) RETURN
   10 IF (NRP.LE.LMP.AND.LML.GE.0) GO TO 30
      LV1=0
      LV2=0
      WRITE (LCH,20)
   20 FORMAT (//26H ( LISTING LIMIT REACHED )//)
      NRL=NRL-5
      LML=LML-5
      RETURN
   30 IF (NRL.LT.7) CALL MFN00A
      WRITE (LCH,40) K,F
   40 FORMAT (22H0FUNCTION EVALUATION :,I4,13X,21HOBJECTIVE FUNCTION : ,
     1 1PE19.12)
      NRL=NRL-2
      LML=LML-2
      CALL MFN00V (X,N,G,C,M)
      IF (LG1+LG2.EQ.0) RETURN
      IF (K.LE.LG2) GO TO 50
```

```
      IF (K.LE.LV2) RETURN
      IF (LG1.EQ.0) RETURN
      IF (MOD(K,LG1).NE.0) RETURN
   50 CALL MFN00G (D,N,M,L)
      RETURN
      END
C
C
      SUBROUTINE MFN00V (X,N,G,C,M)
C
C     PRINT VALUES OF VARIABLES AND CONSTRAINTS.
C
      DIMENSION X(1), G(1), C(1)
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      IF (LCH.LE.0) RETURN
      K=MAX0(N,M)
      IF (NRL.LT.5) CALL MFN00A
      WRITE (LCH,10)
   10 FORMAT (/17X,9HVARIABLES,11X,8HGRADIENT,20X,11HCONSTRAINTS/)
      NRL=NRL-3
      LML=LML-3
      DO 40 I=1,K
      IF (NRL.LE.0) CALL MFN00A
      IF (I.LE.N.AND.I.LE.M) WRITE (LCH,20) I,X(I),G(I),I,C(I)
      IF (I.LE.N.AND.I.GT.M) WRITE (LCH,20) I,X(I),G(I)
      IF (I.GT.N.AND.I.LE.M) WRITE (LCH,30) I,C(I)
   20 FORMAT (5X,I4,2X,1PE19.12,2X,1PE17.10,5X,I4,2X,1PE19.12)
   30 FORMAT (54X,I4,2X,1PE19.12)
      NRL=NRL-1
      LML=LML-1
   40 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE MFN00G (D,N,M,L)
C
C     PRINT PARTIAL DERIVATIVES OF CONSTRAINTS.
C
      DIMENSION D(L,M)
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      IF (LCH.LE.0) RETURN
      IF (M.LE.0) RETURN
      IF (NRL.LT.7) CALL MFN00A
      MM=MIN0(M,LMF)
      NN=MIN0(N,LMV)
      WRITE (LCH,10)
   10 FORMAT (43H0  CONSTRAINT DERIVATIVES ( DF.I / DX.J ) :)
      NRL=NRL-2
      LML=LML-2
      DO 60 K=1,NN,LGH
      IF (NRL.LT.5) CALL MFN00A
      J1=K
      J2=MIN0(NN,K+LGH-1)
      WRITE (LCH,20) (J,J=J1,J2)
   20 FORMAT (1H0,9X,12HVARIABLES(J),10(I5,5X))
      WRITE (LCH,30)
   30 FORMAT (8X,14HCONSTRAINTS(I))
      NRL=NRL-3
      LML=LML-3
      DO 50 I=1,MM
      IF (NRL.LE.0) CALL MFN00A
      WRITE (LCH,40) I,(D(J,I),J=J1,J2)
```

```
   40 FORMAT (10X,I6,4X,10(1PE10.2))
      NRL=NRL-1
      LML=LML-1
   50 CONTINUE
   60 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE MFN00H (FDF,N,M,X,F,DF,G,DG,DH,M1)
C
C     NUMERICAL VERIFICATION OF USER-DEFINED PARTIAL DERIVATIVES
C     (VARIABLES ARE DISTURBED ONE BY ONE).
C
      DIMENSION X(N), F(M1), DF(N,M1), G(M1), DG(M1), DH(N,M1)
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      IF (LCH.LE.0) RETURN
      K=0
      CALL FDF (N,M,X,F(1),DF(1,1),F(2),DF(1,2),N)
      DO 60 I=1,N
      Z=X(I)
      DX=1.E-6*Z
      IF (ABS(DX).LT.1.E-10) DX=1.E-10
      DX2=DX+DX
      X(I)=Z+DX
      CALL FDF (N,M,X,F(1),DH(1,1),F(2),DH(1,2),N)
      DO 10 J=1,M1
      DG(J)=DH(I,J)
   10 CONTINUE
      X(I)=Z-DX
      CALL FDF (N,M,X,G(1),DH(1,1),G(2),DH(1,2),N)
      X(I)=Z
      DO 50 J=1,M1
      Y=DF(I,J)
      Z=F(J)-G(J)
      IF (ABS(Z).LE.0.5E-13*(F(J)+G(J))) Z=0.0
      Z=Z/DX2
      IF (ABS(Y).LE.1.E-20.AND.ABS(Z).LE.1.E-20) GO TO 50
      IF (ABS(Z).LT.1.E-20) Z=SIGN(1.E-20,Z)
      R=100.0*ABS((Z-Y)/Z)
      IF (R.LE.1.0) GO TO 50
      IF (SIGN(1.0,DG(J))+SIGN(1.0,DH(I,J)).EQ.0.0) GO TO 50
      IF (K.NE.0) GO TO 30
      IF (NRL.LT.5) CALL MFN00A
      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 MFN00A
      L=J-1
      WRITE (LCH,40) L,I,Y,Z,R
   40 FORMAT (19X,I5,3X,I4,5X,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 MFN00A
      WRITE (LCH,70)
   70 FORMAT (47H0VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.)
      NRL=NRL-2
      LML=LML-2
```

```
   80 RETURN
      END
C
C
      SUBROUTINE MFN00A
C
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 (1H1/7H DATE :,A10,19X,6HTIME :,A10,20X,6HPAGE :,I3/
     1 55H MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE),17X,
     2 9H(V:82.05))
      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 (/1X)
      RETURN
      END
C
C
      SUBROUTINE MMXPSZ (L)
C
C     DEFINE THE PAGE SIZE (I.E. THE NUMBER OF LINES PER PAGE).
C
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      DATA LL/65/
      IF (L.GT.0) LL=MAX0(25,L)
      IF (L.EQ.0) LL=0
      MXL=LL
      RETURN
      END
C
C
      SUBROUTINE MMXPLM (L)
C
C     DEFINE THE LIMIT OF PRINTED PAGES.
C
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      DATA LL/10/
      IF (L.GT.0) LL=MIN0(50,L)
      LMP=LL
      RETURN
      END
C
C
      SUBROUTINE MMXLLM (L)
C
C     DEFINE THE LIMIT OF PRINTED LINES.
C
```

```
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      DATA LL/750/
      IF (L.GT.0) LL=L
      LML=LL
      RETURN
      END
C
C
      SUBROUTINE MMXHDR (L,T)
C
C     DEFINE THE HEADER LINE.
C
      DIMENSION T(1)
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      DATA LL/0/
      IF (L.GE.0) LL=MIN0(8,L)
      LHT=LL
      IF (L.LE.0) RETURN
      DO 10 I=1,LL
      H(I)=T(I)
   10 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE MMXGLM (K,L)
C
C     DEFINE THE SIZE OF PRINTED JACOBIAN.
C
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      DATA KK/25/,LL/10/
      IF (K.GT.0) KK=K
      IF (L.GT.0) LL=L
      LMF=KK
      LMV=LL
      RETURN
      END
C
C
      SUBROUTINE MMXGVL (L)
C
C     DEFINE THE NUMBER OF JACOBIAN COLUMNS PRINTED IN ONE LINE.
C
      COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG
     1H,DAT,TIM,LHT,H(8)
      DATA LL/10/
      IF (L.GT.0) LL=MAX0(MIN0(10,L),5)
      LGH=LL
      RETURN
      END
C
C
      SUBROUTINE VF02A  (N,M,MEQ,X,F,G,C,CN,LCN,MAXFUN,ACC,IPRINT,INF,W,
     1LW)
      REAL X(N),F,G(N),C(M),CN(LCN,M),ACC,W(LW)
C
C     N        IS THE NUMBER OF VARIABLES
C     M        IS THE TOTAL NUMBER OF CONSTRAINTS
C     MEQ      IS THE NUMBER OF EQUALITY CONSTRAINTS
C     X        IS THE VECTOR OF VARIABLES
C              IT MUST BE SET BY THE USER BEFORE THE INITIAL CALL AND LEFT
C              UNCHANGED THEREAFTER.
```

```
C     F       IS THE VALUE OF THE OBJECTIVE FUNCTION
C     G       IS THE GRADIENT OF THE OBJECTIVE FUNCTION
C     C       IS THE VECTOR OF CONSTRAINT FUNCTIONS
C     CN      IS THE MATRIX OF CONSTRAINT NORMALS
C     LCN     IS THE FIRST DIMENSION OF CN
C
C     F,G,C,CN MUST ALL BE SET BY THE USER BEFORE EACH CALL.
C
C     MAXFUN BOUNDS THE NUMBER OF CALLS OF VF02A/AD
C     ACC     CONTROLS THE FINAL ACCURACY - THE CALCULATION ENDS WHEN THE
C             OBJECTIVE FUNCTION PLUS SUITABLY WEIGHTED MULTIPLES OF THE
C             CONSTRAINT FUNCTIONS ARE PREDICTED TO DIFFER FROM THEIR
C             OPTIMAL VALUES BY AT MOST ACC
C     IPRINT CONTROLS THE AMOUNT OF PRINTING
C          .LT.0  NO PRINTING
C          .EQ.0  DIAGNOSTICS ONLY
C          .GT.0  X, F AND C AT START OF EVERY IPRINT ITERATIONS
C     INF CONTROLS THE CALCULATION
C          =-1 ON INITIAL CALL
C          =0 DURING CALCULATION
C          =1 WHEN REQUIRED ACCURACY IS ACHIEVED
C          =2 WHEN VF02A/AD IS CALLED MAXFUN TIMES
C          =3 WHEN A LINE SEARCH REQUIRES 5 CALLS OF VF02A/AD
C          =4 WHEN AN UPHILL SEARCH DIRECTION IS CALCULATED
C          =5 WHEN NO FEASIBLE POINT IS FOUND BY VF02C/CD
C          =6 WHEN AN ARTIFICIAL BOUND RESTRICTS VF02C/CD
C          =7 WHEN LW IS TOO SMALL
C          =8 WHEN N,M OR MEQ HAS A SILLY VALUE
C     W        IS A WORKSPACE ARRAY OF LENGTH LW
C     LW       IS THE LENGTH OF ARRAY W
C
      COMMON /VF02D/ VLARGE,LP
      DATA VLARGE/1.0E6/,LP/0/
C
C     LP IS THE UNIT NUMBER FOR PRINTING MESSAGES
C
      IW=5*N*N+23*N+18+4*M+MAX0(M,3*N+3)
      IF (IW.LE.LW) GO TO 20
      IF (IPRINT.GE.0.AND.LP.GT.0) WRITE (LP,10) IW
   10 FORMAT (49H ERROR RETURN FROM VF02A/AD BECAUSE W SHOULD HAVE,
     1 16H LENGTH AT LEAST,I8)
      INF=7
      GO TO 50
   20 IF (N.GT.0.AND.M.GE.MEQ.AND.MEQ.GE.0) GO TO 40
      IF (IPRINT.GE.0.AND.LP.GT.0) WRITE (LP,30) N,M,MEQ
   30 FORMAT (38H ERROR RETURN FROM VF02A/AD BECAUSE N=,I6,5H   M=,I6,
     1 7H    MEQ=,I6)
      INF=8
      GO TO 50
   40 IVLAM=1
      IVMU=IVLAM+M
      IB=IVMU+M
      IDELTA=IB+(N+1)*(N+1)
      IGLAG=IDELTA+N+1
      IGLAGA=IGLAG+N
      IXA=IGLAGA+N
      CALL VF02B  (N,M,MEQ,X,F,G,C,CN,LCN,MAXFUN,ACC,IPRINT,INF,W,LW,N+1
     1,W(IVLAM),W(IVMU),W(IB),W(IDELTA),W(IGLAG),W(IGLAGA),W(IGLAGA),W(I
     2GLAGA),W(IXA),W(IXA))
   50 RETURN
      END
C
C
      SUBROUTINE VF02B  (N,M,MEQ,X,F,G,C,CN,LCN,MAXFUN,ACC,IPRINT,INF,W,
     1LW,NP,VLAM,VMU,B,DELTA,GLAG,GLAGA,GAMMA,ETA,XA,BDELTA)
```

```
      REAL X(N),F,G(N),C(M),CN(LCN,M),ACC,W(LW)
      DIMENSION VLAM(M),VMU(M),DELTA(NP),GLAG(N),GLAGA(N),XA(N),GAMMA(N)
     1,BDELTA(N),ETA(N),B(NP,NP)
C
C     VLAM   IS THE VECTOR OF LAGRANGE MULTIPLIERS
C     VMU    HOLDS THE PARAMETERS FOR THE LINE SEARCH FUNCTION
C     DELTA  IS THE SEARCH DIRECTION TIMES THE STEP-LENGTH
C     GLAG   IS THE GRADIENT OF THE LAGRANGIAN FUNCTION
C     GLAGA  IS THIS GRADIENT AT THE START OF AN ITERATION
C     XA     IS THE VECTOR OF VARIABLES AT THE START OF AN ITERATION
C     GAMMA  IS THE CHANGE IN GRADIENT OF THE LAGRANGIAN FUNCTION
C     BDELTA IS B TIMES DELTA
C     ETA    REPLACES GAMMA IN THE B-F-G-S FORMULA FOR REVISING B
C     B      IS THE VARIABLE METRIC MATRIX
C
C     (GLAGA,GAMMA,ETA) AND (XA,BDELTA) ARE ESSENTIALLY EQUIVALENT,
C          SINCE THEY ARE ASSOCIATED WITH THE SAME PART OF ARRAY W.
C
      COMMON /VF02D/ VLARGE,LP
      COMMON /VF02E/  FLS,SUM,FLSA,ALPHA,DFLSA,SPCDEL,NF,ITER,ITERP,NFIN
     1IT,MACT
      DATA XZERO,XONE,XHALF,XZONE,XZTWO/0.0E0,1.0E0,0.5E0,0.1E0,0.2E0/
C
C        SET SOME PARAMETERS FOR THE CALCULATION
C           NFLINE  CONTROLS THE ERROR RETURN FROM THE LINE SEARCH
C           PARACC  IS THE SLOPE OF THE ARMIJO CHORD
C           PARSTP  LIMITS THE REDUCTION IN THE LINE SEARCH STEP-LENGTH
C           PARB    BOUNDS THE REDUCTION IN THE DETERMINANT OF B
C
      NFLINE=5
      PARACC=XZONE
      PARSTP=XZONE
      PARB=XZTWO
      IF (INF.EQ.0) GO TO 270
C
C        SET INITIAL VALUES OF SOME VARIABLES
C           NF    IS THE NUMBER OF CALLS OF VF02A/AD
C           ITER  IS THE ITERATION NUMBER
C           ITERP IS THE NEXT ITERATION ON WHICH PRINTING OCCURS
C
      NF=1
      ITER=0
      ITERP=MIN0(1,IPRINT)
C
C        SET THE INITIAL ELEMENTS OF B AND VMU
C
      DO 20 I=1,N
      DO 10 J=1,N
      B(I,J)=XZERO
   10 CONTINUE
      B(I,I)=XONE
   20 CONTINUE
      IF (M.EQ.0) GO TO 40
      DO 30 K=1,M
      VMU(K)=XZERO
   30 CONTINUE
C
C        START THE ITERATION BY PROVIDING PRINTING
C
   40 ITER=ITER+1
      IF (ITER.NE.ITERP) GO TO 100
   50 IF (LP.GT.0) WRITE (LP,60) ITER,NF
   60 FORMAT (/5X,12HITERATIONS =,I5,5X,19HCALLS OF VF02A/AD =,I5)
      IF (LP.GT.0) WRITE (LP,70) (X(I),I=1,N)
   70 FORMAT (/5H  X =,5E20.10/(5X,5E20.10))
```

```
      IF (LP.GT.0) WRITE (LP,80) F
   80 FORMAT (/5H  F =,E20.10)
      IF (M.GT.0.AND.LP.GT.0) WRITE (LP,90) (C(K),K=1,M)
   90 FORMAT (/5H  C =,5E20.10/(5X,5E20.10))
      IF (ITER.NE.ITERP) GO TO 440
      ITERP=ITER+IPRINT
C
C        CALL THE QUADRATIC PROGRAMMING SUBROUTINE
C
  100 IDELTA=1+2*M+NP*NP
C
C        THIS CHOICE OF IDELTA MEANS THAT ARRAYS DELTA OF THIS SUBROUTINE
C        AND OF VF02C/CD ARE DYNAMICALLY EQUIVALENT
C
      LDELTA=4*NP+MAX0(M,3*NP)
      IGM=IDELTA+LDELTA
      ICM=IGM+NP
      IBDL=ICM+M
      IBDU=IBDL+N+1
      IH=IBDU+NP
      ILT=IH+NP*NP*4
      LLT=M+NP*6
      CALL VF02C  (N,M,MEQ,G,C,CN,LCN,IPRINT,INF,NP,2*NP,VLAM,B,W(IDELTA
     1),LDELTA,W(IGM),W(ICM),W(IBDL),W(IBDU),W(IH),W(ILT),LLT)
      IF (INF.LE.1) GO TO 120
      IF (IPRINT.LT.0) GO TO 460
      IF (LP.GT.0) WRITE (LP,110)
  110 FORMAT (/5X,24HERROR CONDITION FOUND IN,
     1 34H QUADRATIC PROGRAMMING CALCULATION)
      GO TO 430
C
C        CALCULATE THE GRADIENT OF THE LAGRANGIAN FUNCTION
C        NFINIT IS THE VALUE OF NF AT THE START OF AN ITERATION
C
  120 NFINIT=NF
  130 DO 140 I=1,N
      GLAG(I)=G(I)
  140 CONTINUE
      IF (M.EQ.0) GO TO 170
      DO 160 K=1,M
      IF (VLAM(K).EQ.XZERO) GO TO 160
      DO 150 I=1,N
      GLAG(I)=GLAG(I)-CN(I,K)*VLAM(K)
  150 CONTINUE
  160 CONTINUE
  170 IF (NF.NE.NFINIT) GO TO 350
C
C        STORE THE ELEMENTS OF GLAG AND X
C        SET SPGDEL TO THE SCALAR PRODUCT OF G AND DELTA
C
      SPGDEL=XZERO
      DO 180 I=1,N
      SPGDEL=SPGDEL+G(I)*DELTA(I)
      GLAGA(I)=GLAG(I)
      XA(I)=X(I)
  180 CONTINUE
C
C        REVISE THE VECTOR VMU
C        TEST FOR CONVERGENCE
C
      SUM=ABS(SPGDEL)
      IF (M.LE.0) GO TO 200
      DO 190 K=1,M
      AUX=ABS(VLAM(K))
      VMU(K)=AMAX1(AUX,XHALF*(AUX+VMU(K)))
```

```
      SUM=SUM+ABS(VLAM(K)*C(K))
  190 CONTINUE
  200 IF (SUM.GT.ACC) GO TO 320
      INF=1
      GO TO 430
C
C        SET THE INITIAL CONDITIONS FOR THE LINE SEARCH
C          FLSA    IS THE INITIAL VALUE OF THE LINE SEARCH FUNCTION
C          DFLSA   IS USUALLY ITS FIRST DERIVATIVE
C          ALPHA   IS THE NEXT REDUCTION IN THE STEP-LENGTH
C
  210 FLSA=FLS
      DFLSA=SPGDEL-DELTA(N+1)*SUM
      IF (DFLSA.GE.XZERO) GO TO 410
      ALPHA=XONE
C
C        MULTIPLY DELTA BY ALPHA AND CALCULATE THE NEW X
C
  220 DO 230 I=1,N
      DELTA(I)=ALPHA*DELTA(I)
      X(I)=XA(I)+DELTA(I)
  230 CONTINUE
      DFLSA=ALPHA*DFLSA
C
C        TEST NF AGAINST MAXFUN AND RETURN FOR MORE VALUES OF F,G,C,CN
C
      IF (NF.LT.MAXFUN) GO TO 260
  240 DO 250 I=1,N
      X(I)=XA(I)
  250 CONTINUE
      IF (NF.EQ.NFINIT) GO TO 280
  260 NF=NF+1
      GO TO 460
  270 IF (NF.LE.MAXFUN) GO TO 300
  280 INF=2
      IF (IPRINT.LT.0) GO TO 460
      IF (LP.GT.0) WRITE (LP,290) NF
  290 FORMAT (/5X,33HERROR RETURN FROM VF02A/AD DUE TO,I5,
     1 21H FUNCTION EVALUATIONS)
      GO TO 430
C
C        TEST FOR ERROR RETURN FROM LINE SEARCH
C
  300 IF (NF.LE.NFINIT+NFLINE) GO TO 320
      INF=3
      IF (IPRINT.LT.0) GO TO 460
      IF (LP.GT.0) WRITE (LP,310) NFLINE
  310 FORMAT (/5X,49HRETURN FROM VF02A/AD BECAUSE LINE SEARCH REQUIRES,
     1 10H MORE THAN,I3,6H STEPS)
      GO TO 430
C
C        SET SUM TO THE WEIGHTED SUM OF INFEASIBILITIES
C        SET FLS TO THE LINE SEARCH OBJECTIVE FUNCTION
C
  320 SUM=XZERO
      IF (M.LE.0) GO TO 340
      DO 330 K=1,M
      AUX=C(K)
      IF (K.GT.MEQ) AUX=XZERO
      SUM=SUM+VMU(K)*AMAX1(AUX,-C(K))
  330 CONTINUE
  340 FLS=F+SUM
      IF (NF.EQ.NFINIT) GO TO 210
C
C        CALCULATE THE GRADIENT OF THE LAGRANGIAN FUNCTION
```

```
C        IF THE LINE SEARCH IS COMPLETE
C
      AUX=FLS-FLSA
      IF (AUX.LE.PARACC*DFLSA) GO TO 130
      IF (NF.GE.NFINIT+NFLINE) GO TO 240
C
C        CALCULATE THE NEXT REDUCTION IN THE STEP-LENGTH
C
      ALPHA=AMAX1(PARSTP,XHALF*DFLSA/(DFLSA-AUX))
      GO TO 220
C
C        CALCULATE GAMMA AND BDELTA IN ORDER TO REVISE B
C        SET DG TO THE SCALAR PRODUCT OF DELTA AND GAMMA
C        SET DBD TO THE SCALAR PRODUCT OF DELTA AND BDELTA
C
  350 DG=XZERO
      DBD=XZERO
      DO 370 I=1,N
      GAMMA(I)=GLAG(I)-GLAGA(I)
      BDELTA(I)=XZERO
      DO 360 J=1,N
      BDELTA(I)=BDELTA(I)+B(I,J)*DELTA(J)
  360 CONTINUE
      DG=DG+DELTA(I)*GAMMA(I)
      DBD=DBD+DELTA(I)*BDELTA(I)
  370 CONTINUE
C
C        CALCULATE THE VECTOR ETA FOR THE B-F-G-S FORMULA
C        REPLACE DG BY THE SCALAR PRODUCT OF DELTA AND ETA
C
      AUX=PARB*DBD
      IF (DG.GE.AUX) GO TO 390
      THETA=(DBD-AUX)/(DBD-DG)
      THCOMP=XONE-THETA
      DO 380 I=1,N
      ETA(I)=THETA*GAMMA(I)+THCOMP*BDELTA(I)
  380 CONTINUE
      DG=AUX
C
C        REVISE THE MATRIX B
C
  390 DO 400 I=1,N
      AUX=BDELTA(I)/DBD
      AUXA=ETA(I)/DG
      DO 400 J=I,N
      B(I,J)=B(I,J)-AUX*BDELTA(J)+AUXA*ETA(J)
  400 B(J,I)=B(I,J)
      GO TO 40
C
C        TEST IF ANY PRINTING IS REQUIRED BEFORE THE RETURN
C
  410 INF=4
      IF (IPRINT.LT.0) GO TO 460
      IF (LP.GT.0) WRITE (LP,420)
  420 FORMAT (/5X,43HERROR RETURN FROM VF02A/AD DUE TO AN UPHILL,
     1 17H SEARCH DIRECTION/5X,30HUSUALLY THIS IS DUE TO LOSS OF,
     2 50H ACCURACY IN THE QUADRATIC PROGRAMMING CALCULATION)
  430 IF (IPRINT.LE.0) GO TO 460
      IF (ITERP.NE.ITER+IPRINT) GO TO 50
  440 IF (LP.GT.0) WRITE (LP,450)
  450 FORMAT (/5X,44HTHE PRINTING OF THE LAST ITERATION GIVES THE,
     1 48H VALUES THAT ARE RETURNED BY SUBROUTINE VF02A/AD)
  460 RETURN
      END
C
```

```
C
      SUBROUTINE VF02C  (N,M,MEQ,G,C,CN,LCN,IPRINT,INF,NP,NPP,VLAM,B,DEL
     1TA,LDELTA,GM,CM,BDL,BDU,H,LT,LLT)
      DIMENSION G(N),C(M),CN(LCN,M),VLAM(M),B(NP,NP),DELTA(LDELTA),GM(NP
     1),CM(M),BDL(NP),BDU(NP),H(NPP,NPP)
      INTEGER LT(LLT)
      COMMON /VF02D/ VLARGE,LP
      COMMON /VF02E/ DUMMY(6),IDUMMY(4),MACT
      DATA XZERO,XONE,XONEM6,XZNINE/0.0E0,1.0E0,1.0E-6,0.9E0/
C
C        NP = N + 1
C        NPP = NP + NP
C
C        GM IS SET TO MINUS THE VECTOR G
C        CM IS SET TO MINUS THE VECTOR C
C        BDL AND BDU GIVE LOWER AND UPPER BOUNDS ON DELTA
C        H AND LT ARE USED AS WORKING SPACE BY VE02A/AD
C
C        SET SOME PARAMETERS THAT ARE USED BY VF02C/CD AND VE02A/AD
C          VLARGE IS ASSUMED TO BE A LARGE NUMBER
C          VSMALL IS ASSUMED TO BE A SMALL POSITIVE NUMBER
C          FEASP IS A SCALING FACTOR THAT IS USED TO ACHIEVE FEASIBILITY
C
      MODE=1
      VSMALL=XONEM6
      FEASP=XZNINE
      MTOTAL=M+NPP
      NSIX=6*NP
      IF (INF.GE.0) GO TO 50
C
C        SET INITIAL VALUES OF SOME VARIABLES
C
      INF=0
      MACT=MEQ+1
C
C        SET THE INITIAL ELEMENTS OF BDL, BDU, DELTA AND LT
C
      DO 10 I=1,N
      BDL(I)=-VLARGE
      BDU(I)=VLARGE
      DELTA(I)=XZERO
   10 CONTINUE
      BDL(NP)=XZERO
      DELTA(NP)=XONE
      IF (MEQ.LE.0) GO TO 30
      DO 20 K=1,MEQ
      LT(K)=K+NPP
   20 CONTINUE
   30 LT(MACT)=NPP
C
C        EXTEND GM AND B BECAUSE OF THE EXTRA VARIABLE THAT IS
C        INTRODUCED TO ALLOW FOR INFEASIBILITY
C
      GM(NP)=VLARGE
      DO 40 I=1,NP
      B(I,NP)=XZERO
      B(NP,I)=XZERO
   40 CONTINUE
C
C        SET THE ELEMENTS OF GM, CM AND CN(NP,*)
C
   50 DO 60 I=1,N
      GM(I)=-G(I)
   60 CONTINUE
      IF (M.LE.0) GO TO 90
```

```
      DO 80 K=1,M
      IF (K.LE.MEQ) GO TO 70
      IF (C(K).LT.XZERO) GO TO 70
      CM(K)=-C(K)
      CN(NP,K)=XZERO
      GO TO 80
   70 CM(K)=XZERO
      CN(NP,K)=C(K)
   80 VLAM(K)=XZERO
C
C        CALL SUBROUTINE VE02A/AD
C
   90 BDU(NP)=XONE
      IFLAG=-1
  100 CALL VE02A  (NP,MTOTAL,B,NP,GM,CN,LCN,CM,BDL,BDU,DELTA,MACT,MEQ,H,
     1NPP,LT,MODE)
C
C        CHECK WHETHER THE REQUIRED FEASIBILITY CONDITIONS HOLD
C
      IF (DELTA(NP).LE.VSMALL) GO TO 150
      DO 110 J=1,MACT
      IF (LT(J).GT.NPP) GO TO 110
      IF (LT(J).LT.NPP) GO TO 170
      IFLAG=1
  110 CONTINUE
      IF (IFLAG.GE.1) GO TO 120
      IF (IFLAG.GE.0) GO TO 150
      BDU(NP)=FEASP*DELTA(NP)
      IFLAG=0
      GO TO 100
C
C        CALCULATE THE LAGRANGE MULTIPLIERS
C
  120 DO 140 J=1,MACT
      K=LT(J)-NPP
      NPJ=NP+J
      IF (K.LE.0) GO TO 140
      DO 130 I=1,N
      NSIXI=NSIX+I
      VLAM(K)=VLAM(K)+H(NPJ,I)*DELTA(NSIXI)
  130 CONTINUE
  140 CONTINUE
      GO TO 190
C
C        RETURN FROM THE SUBROUTINE
C
  150 INF=5
      IF (IPRINT.LT.0) GO TO 190
      IF (LP.GT.0) WRITE (LP,160)
  160 FORMAT (/5X,45HTHE GIVEN CONSTRAINTS SEEM TO BE INCONSISTENT)
      GO TO 190
  170 INF=6
      IF (IPRINT.LT.0) GO TO 190
      IF (LP.GT.0) WRITE (LP,180) VLARGE
  180 FORMAT (/5X,46HVE02A FINDS THAT AN ARTIFICIAL BOUND IS ACTIVE
     1 /5X,45HTHE PREDICTED CHANGE IN THE VARIABLES EXCEEDS,1PE12.4)
  190 RETURN
      END
C
C
      SUBROUTINE VE02A  (N,M,A,IA,B,C,IC,D,BDL,BDU,X,K,KE,H,IH,LT,MODE)
      DIMENSION A(IA,1), B(1), C(IC,1), D(1), BDL(1), BDU(1), X(1), H(IH
     1,1), LT(1)
      LOGICAL RETEST,PASSIV,POSTIV
      COMMON /VE02X/ LPR
```

```
      DATA LPR/0/
      DATA XZERO,XONE,XONE75/0.0E0,1.0E0,1.0E75/
      RETEST=.FALSE.
      NN=N+N
      N3=NN+N
      N4=NN+NN
      N5=N4+N
      N6=N5+N
      NNM=NN+M+1
      IF (MODE.GE.3) GO TO 30
C
C        CALL FEASIBLE VERTEX ROUTINE
C
   10 CALL LA02A  (N,M,C,IC,D,BDL,BDU,X,K,KE,H,IH,LT)
      IF (K.EQ.0) RETURN
      IF (MODE.EQ.2.AND..NOT.RETEST) GO TO 60
C
C        INITIAL OPERATORS H=0 AND CSTAR=C(-1)
C
      DO 20 I=1,N
      DO 20 J=1,N
      H(N+I,J)=H(I,J)
   20 H(I,J)=XZERO
      GO TO 260
C
   30 DO 40 I=1,M
      LT(NN+I)=1
   40 CONTINUE
C
C        CONSTRAINTS INDEXED AS  -1=EQUALITY, 0=ACTIVE, 1=INACTIVE
C
      IF (K.EQ.0) GO TO 60
      DO 50 I=1,K
      J=0
      IF (I.LE.KE) J=-1
      LT(NN+LT(I))=J
   50 CONTINUE
   60 IF (MODE.EQ.5.AND..NOT.RETEST) GO TO 150
C
C        SET UP MATRIX AND RHS OF EQUATIONS GOVERNING EQUALITY PROBLEM
C
      DO 70 I=1,N
      X(N+I)=B(I)
      DO 70 J=1,N
   70 H(I,J)=A(I,J)
      IF ((MODE.EQ.2.OR.MODE.EQ.3).AND..NOT.RETEST) GO TO 820
      IF (K.EQ.0) GO TO 140
      DO 130 I=1,K
      LI=LT(I)
      IF (LI.GT.NN) GO TO 100
      DO 80 J=1,N
      H(J,N+I)=XZERO
      H(N+I,J)=XZERO
   80 CONTINUE
      IF (LI.GT.N) GO TO 90
      H(N+I,LI)=XONE
      H(LI,N+I)=XONE
      X(NN+I)=BDL(LI)
      GO TO 120
   90 LI=LI-N
      H(N+I,LI)=-XONE
      H(LI,N+I)=-XONE
      X(NN+I)=-BDU(LI)
      GO TO 120
  100 LI=LI-NN
```

```
      DO 110 J=1,N
      H(N+I,J)=C(J,LI)
      H(J,N+I)=C(J,LI)
  110 CONTINUE
      X(NN+I)=D(LI)
  120 DO 130 J=1,K
  130 H(N+I,N+J)=XZERO
  140 NK=N+K
C
C        INVERT MATRIX GIVING OPERATORS H AND CSTAR
C
      CALL MB01C  (H,NK,IH,LT(NNM),X(N3+1))
      GO TO 200
C
C        SET UP RHS ONLY
C
  150 DO 160 I=1,N
      X(N+I)=B(I)
  160 CONTINUE
      DO 190 I=1,K
      LI=LT(I)
      IF (LI.GT.NN) GO TO 180
      IF (LI.GT.N) GO TO 170
      X(NN+I)=BDL(LI)
      GO TO 190
  170 X(NN+I)=-BDU(LI-N)
      GO TO 190
  180 X(NN+I)=D(LI-NN)
  190 CONTINUE
C
C        SOLVE FOR SOLUTION POINT X
C
      NK=N+K
  200 DO 210 I=1,N
      X(I)=FM02AS(NK,H(1,I),1,X(N+1),1)
  210 CONTINUE
C
C        CHECK FEASIBILITY, IF NOT EXIT TO 10
C
      DO 250 I=1,M
      IF (LT(NN+I).LE.0) GO TO 250
      IF (I.GT.N) GO TO 220
      Z=X(I)-BDL(I)
      GO TO 240
  220 IF (I.GT.NN) GO TO 230
      Z=BDU(I-N)-X(I-N)
      GO TO 240
  230 J=I-NN
      Z=FM02AS(N,C(1,J),1,X(1),1)-D(J)
  240 IF (Z.LT.XZERO) GO TO 10
  250 CONTINUE
  260 CONTINUE
C
C        CALCULATE GRADIENT G AND LAGRANGE MULTIPLIERS -CSTAR.G,
C        FIND LARGEST MULTIPLIER,  EXIT IF NOT POSITIVE
C
      DO 270 I=1,N
      X(N6+I)=FM02AS(N,A(I,1),IA,X(1),1)-B(I)
  270 CONTINUE
      IF (K.EQ.0) RETURN
      Z=-XONE75
      DO 280 I=1,K
      IF (LT(NN+LT(I)).EQ.-1) GO TO 280
      ZZ=-FM02AS(N,H(N+I,1),IH,X(N6+1),1)
      IF (ZZ.LE.Z) GO TO 280
```

```
      Z=ZZ
      II=I
  280 CONTINUE
      IF (Z.GT.XZERO) GO TO 310
      IF (RETEST.OR.MODE.GE.4) GO TO 290
      RETEST=.TRUE.
      GO TO 60
  290 IF (Z.NE.XZERO) RETURN
      IF (LPR.GT.0) WRITE (LPR,300)
  300 FORMAT (43H0SOLUTION MAY BE A DEGENERATE LOCAL MINIMUM)
      RETURN
C
C         SET DIRECTION OF SEARCH AS CORRESPONDING ROW OF CSTAR
C
  310 DO 320 I=1,N
      X(NN+I)=H(N+II,I)
  320 CONTINUE
  330 DO 340 I=1,N
      X(N+I)=FM02AS(N,A(I,1),IA,X(NN+1),1)
  340 CONTINUE
      CAC=FM02AS(N,X(NN+1),1,X(N+1),1)
      IF (CAC.GT.XZERO) GO TO 350
      POSTIV=.FALSE.
      Y=XONE
      GO TO 360
  350 POSTIV=.TRUE.
      Y=Z/CAC
  360 DO 370 I=1,N
      X(N5+I)=X(NN+I)*Y
  370 CONTINUE
      PASSIV=.TRUE.
  380 ALPHA=XONE75
      NK=N+K
C
C         LINEAR SEARCH ALONG DIRECTION OF SEARCH,  PASSIV INDICATES
C         A CONSTRAINT HAS BEEN REMOVED TO GET SEARCH DIRECTION,
C         POSTIV INDICATES POSITIVE CURVATURE ALONG THE DIRECTION
C
      DO 420 I=1,M
      IF (LT(NN+I).LE.0) GO TO 420
      IF (I.GT.N) GO TO 390
      IF (X(N5+I).GE.XZERO) GO TO 420
      CC=(BDL(I)-X(I))/X(N5+I)
      GO TO 410
  390 IF (I.GT.NN) GO TO 400
      IF (X(N4+I).LE.XZERO) GO TO 420
      CC=(BDU(I-N)-X(I-N))/X(N4+I)
      GO TO 410
  400 J=I-NN
      ZZ=FM02AS(N,C(1,J),1,X(N5+1),1)
      IF (ZZ.GE.XZERO) GO TO 420
      CC=D(J)-FM02AS(N,C(1,J),1,X(1),1)
      CC=CC/ZZ
  410 IF (CC.GE.ALPHA) GO TO 420
      ALPHA=CC
      IAL=I
  420 CONTINUE
      IF (PASSIV) LT(NN+LT(II))=1
C
C          IF MINIMUM FOUND, GO TO 680
C
      IF (POSTIV.AND.ALPHA.GE.XONE) GO TO 680
C
C          CALCULATE H.C AND CSTAR.C
C
```

```
      DO 430 I=1,N
      X(I)=X(I)+ALPHA*X(N5+I)
  430 CONTINUE
      ALPHA=ALPHA*Y
      J=1
      IF (K.EQ.N) J=N+1
      IF (IAL.GT.N) GO TO 450
      DO 440 I=J,NK
      X(N3+I)=H(I,IAL)
  440 CONTINUE
      CHC=X(N3+IAL)
      GO TO 500
  450 IB=IAL-N
      IF (IB.GT.N) GO TO 470
      DO 460 I=J,NK
      X(N3+I)=-H(I,IB)
  460 CONTINUE
      CHC=-X(N3+IB)
      GO TO 500
  470 IB=IB-N
      DO 480 I=1,N
      X(N5+I)=C(I,IB)
  480 CONTINUE
      DO 490 I=J,NK
      X(N3+I)=FM02AS(N,H(I,1),IH,X(N5+1),1)
  490 CONTINUE
      IF (K.NE.N) CHC=FM02AS(N,X(N5+1),1,X(N3+1),1)
  500 LT(NN+IAL)=0
      IF (K.EQ.N) GO TO 770
      IF (PASSIV) GO TO 600
C
C        APPLY FORMULA FOR ADDING A CONSTRAINT
C
  510 IF (K.EQ.0) GO TO 530
      DO 520 I=1,K
      ALPHA=X(N4+I)/CHC
      NI=N+I
      DO 520 J=1,N
  520 H(NI,J)=H(NI,J)-ALPHA*X(N3+J)
  530 K=K+1
      LT(K)=IAL
      DO 540 J=1,N
      H(N+K,J)=X(N3+J)/CHC
  540 CONTINUE
      IF (K.LT.N) GO TO 560
      DO 550 I=1,N
      DO 550 J=1,N
  550 H(I,J)=XZERO
      GO TO 580
  560 DO 570 I=1,N
      ALPHA=X(N3+I)/CHC
      DO 570 J=1,I
      H(I,J)=H(I,J)-ALPHA*X(N3+J)
  570 H(J,I)=H(I,J)
  580 IF (.NOT.PASSIV) GO TO 650
C
C        REMOVAL OF A CONSTRAINT HAS BEEN DEFERRED,  SET UP AS IF
C        THE CONSTRAINT IS BEING REMOVED FROM AUGMENTED BASIS
C
      DO 590 I=1,N
      X(N6+I)=FM02AS(N,A(I,1),IA,X(1),1)-B(I)
      X(NN+I)=H(N+II,I)
  590 CONTINUE
      Z=-FM02AS(N,X(N6+1),1,X(NN+1),1)
      IF (Z.EQ.XZERO) GO TO 700
```

```
      GO TO 330
  600 CC=X(N4+II)
      Y=CHC*CAC+CC**2
      GHC=FM02AS(N,X(N6+1),1,X(N3+1),1)
      IF (ALPHA*Y.LT.CHC*(Z-ALPHA*CAC)+GHC*CC) GO TO 510
C
C        APPLY FORMULA FOR EXCHANGING NEW CONSTRAINT
C        WITH PASSIVE CONSTRAINT
C
      DO 610 I=1,K
      NI=N+I
      X(N5+I)=FM02AS(N,H(NI,1),IH,X(N+1),1)
  610 CONTINUE
      DO 620 I=1,N
      X(N+I)=(CHC*X(NN+I)-CC*X(N3+I))/Y
      X(N6+I)=(CAC*X(N3+I)+CC*X(NN+I))/Y
  620 CONTINUE
      DO 630 I=1,N
      DO 630 J=1,I
      H(I,J)=H(I,J)+X(N+I)*X(NN+J)-X(N6+I)*X(N3+J)
  630 H(J,I)=H(I,J)
      X(N4+II)=X(N4+II)-XONE
      DO 640 I=1,K
      NI=N+I
      DO 640 J=1,N
  640 H(NI,J)=H(NI,J)-X(N4+I)*X(N6+J)-X(N5+I)*X(N+J)
      LT(II)=IAL
  650 IF (K.EQ.N) GO TO 260
C
C        CALCULATE G,  NEW SEARCH DIRECTION IS -H.G
C
      DO 660 I=1,N
      X(N+I)=FM02AS(N,A(I,1),IA,X(1),1)-B(I)
  660 CONTINUE
      Z=XZERO
      DO 670 I=1,N
      X(N5+I)=-FM02AS(N,H(I,1),IH,X(N+1),1)
      IF (X(N5+I).NE.XZERO) Z=XONE
  670 CONTINUE
      PASSIV=.FALSE.
      IF (Z.EQ.XZERO) GO TO 260
      POSTIV=.TRUE.
      GO TO 380
  680 DO 690 I=1,N
      X(I)=X(I)+X(N5+I)
  690 CONTINUE
C
C        X IS NOW THE MINIMUM POINT IN THE BASIS
C        UPDATE THE OPERATORS IF A CONSTRAINT HAD BEEN REMOVED
C
      IF (.NOT.PASSIV) GO TO 260
  700 DO 710 I=1,N
      ALPHA=X(NN+I)/CAC
      DO 710 J=1,I
      H(I,J)=H(I,J)+ALPHA*X(NN+J)
  710 H(J,I)=H(I,J)
      IF (K.GT.1) GO TO 720
      K=0
      GO TO 260
  720 IF (II.EQ.K) GO TO 740
      DO 730 I=1,N
      H(N+II,I)=H(N+K,I)
  730 CONTINUE
      LT(II)=LT(K)
  740 K=K-1
```

```
      DO 750 I=1,K
      NI=N+I
      X(N3+I)=FM02AS(N,H(NI,1),IH,X(N+1),1)
  750 CONTINUE
      DO 760 I=1,K
      ALPHA=X(N3+I)/CAC
      NI=N+I
      DO 760 J=1,N
  760 H(NI,J)=H(NI,J)-ALPHA*X(NN+J)
      GO TO 260
  770 Z=XONE/X(N4+II)
C
C       APPLY SIMPLEX FORMULA TO EXCHANGE CONSTRAINTS
C
      DO 810 I=1,N
      NI=N+I
      IF (I.NE.II) GO TO 790
      DO 780 J=1,N
      H(NI,J)=H(NI,J)*Z
  780 CONTINUE
      GO TO 810
  790 ZZ=Z*X(N4+I)
      DO 800 J=1,N
      H(NI,J)=H(NI,J)-ZZ*X(NN+J)
  800 CONTINUE
  810 CONTINUE
      LT(II)=IAL
      GO TO 260
C
  820 K=0
      IF (KE.NE.0.AND.LPR.GT.0) WRITE (LPR,830)
  830 FORMAT (30H0KE MUST BE 0 IN MODES 2 AND 3)
      KE=0
      DO 840 I=1,M
      LT(NN+I)=1
  840 CONTINUE
      CALL MB01C  (H,N,IH,LT(NNM),X(N6+1))
C
C       START WITH EMPTY BASIS FROM FEASIBLE POINT
C       SEARCH DIRECTION IS -A(-1).B
C
      GO TO 650
      END
C
C
      SUBROUTINE LA02A (N,M,C,IC,D,BDL,BDU,X,K,KE,H,IH,LT)
      COMMON /LA02B/ LP,IFLAG
      REAL C(IC,1),D(1),BDL(N),BDU(N),X(1),H(IH,1)
      INTEGER LT(1)
      COMMON /MB01D/ LPMB01,IFMB01
      REAL ZERO,ONE,RANGE
      DATA ZERO,ONE,RANGE/0.0,1.0,1.0E75/
      DATA LP/0/
C
C       SUPPRESS ERROR MESSAGES FROM MB01C/CD
C
      LPMB1=LPMB01
      LPMB01=0
   10 IFLAG=0
      NN=N+N
      N3=NN+N
      DO 20 I=1,M
      N1=NN+I
      LT(N1)=1
   20 CONTINUE
```

```
C
C        CONSTRAINTS INDEXED AS
C        -1=EQUALITY,  0=ACTIVE,  1=INACTIVE,  2=VIOLATED
C
      IF (K.NE.0) GO TO 70
C
C        NO DESIGNATED CONSTRAINTS, VERTEX CHOSEN FROM UPPER AND
C        LOWER BOUNDS,  INVERSE MATRIX TRIVIAL
C
      DO 60 I=1,N
      DO 30 J=1,N
      H(I,J)=ZERO
   30 CONTINUE
      IF (X(I)-BDL(I).GT.BDU(I)-X(I)) GO TO 40
      LT(I)=I
      H(I,I)=ONE
      GO TO 50
   40 LT(I)=N+I
      H(I,I)=-ONE
   50 N1=NN+LT(I)
      LT(N1)=0
   60 CONTINUE
      K=N
      GO TO 330
C
C        SET UP NORMALS V OF THE K DESIGNATED CONSTRAINTS IN BASIS
C
   70 DO 120 I=1,K
      J=0
      IF (I.LE.KE) J=-1
      N1=NN+LT(I)
      LT(N1)=J
      LI=LT(I)
      NI=N+I
      IF (LI.GT.NN) GO TO 100
      DO 80 J=1,N
      H(J,NI)=ZERO
   80 CONTINUE
      IF (LI.GT.N) GO TO 90
      H(LI,NI)=ONE
      GO TO 120
   90 L=LI-N
      H(L,NI)=-ONE
      GO TO 120
  100 LI=LI-NN
      DO 110 J=1,N
      H(J,NI)=C(J,LI)
  110 CONTINUE
  120 CONTINUE
      IF (K.NE.N) GO TO 140
      DO 130 J=1,N
      NJ=N+J
      DO 130 I=1,N
  130 H(I,J)=H(I,NJ)
      CALL MB01C (H,N,IH,LT(N+1),X(N+1))
      IF (IFMB01.NE.0) GO TO 160
      GO TO 330
  140 CONTINUE
C
C        FORM M=(VTRANSPOSE.V)(-1)
C
      DO 150 I=1,K
      N1=N+I
      DO 150 J=I,K
      N2=N+J
```

```
      H(I,J)=FM02AS(N,H(1,N1),1,H(1,N2),1)
  150 H(J,I)=H(I,J)
      CALL MB01C (H,K,IH,LT(N+1),X(N+1))
      IF (IFMB01.EQ.0) GO TO 170
  160 IF (K.EQ.KE) GO TO 690
      K=KE
      GO TO 10
C
C        CALCULATE GENERALIZED INVERSE OF V,  VPLUS=M.VTRANSPOSE
C
  170 DO 190 I=1,K
      DO 180 J=1,K
      N1=N+J
      X(N1)=H(I,J)
  180 CONTINUE
      DO 190 J=1,N
  190 H(I,J)=FM02AS(K,X(N+1),1,H(J,N+1),IH)
C
C        SET UP DIAGONAL ELEMENTS OF THE PROJECTION MATRIX  P=V.VPLUS
C
      DO 200 I=1,N
      N1=N+I
      X(N1)=FM02AS(K,H(1,I),1,H(I,N+1),IH)
  200 CONTINUE
      DO 210 I=1,N
      N1=N+I
      LT(N1)=0
  210 CONTINUE
      KV=K
C
C        ADD BOUND E(I) CORRESPONDING TO THE SMALLEST DIAG(P)
C
  220 Z=ONE
      DO 230 I=1,N
      N1=N+I
      IF (LT(N1).EQ.1) GO TO 230
      IF (X(N1).GE.Z) GO TO 230
      Z=X(N1)
      II=I
  230 CONTINUE
      Y=ONE
      IF (X(II)-BDL(II).GT.BDU(II)-X(II)) Y=-ONE
C
C        CALCULATE VECTORS VPLUS.E(I) AND  U=E(I)-V.VPLUS.E(I)
C
      IF (Y.NE.ONE) GO TO 250
      DO 240 I=1,K
      N1=NN+I
      X(N1)=H(I,II)
  240 CONTINUE
      GO TO 270
  250 DO 260 I=1,K
      N1=NN+I
      X(N1)=-H(I,II)
  260 CONTINUE
  270 CONTINUE
      DO 280 I=1,N
      N1=N+I
      IF (LT(N1).EQ.1) GO TO 280
      N1=N3+I
      X(N1)=-FM02AS(KV,H(I,N+1),IH,X(NN+1),1)
  280 CONTINUE
      DO 290 I=1,N
      H(I,II)=ZERO
  290 CONTINUE
```

```
      I1=N+II
      LT(I1)=1
      I3=N3+II
      Z=ONE+X(I3)*Y
C
C        UPDATE VPLUS AND DIAG(P)
C
      DO 310 I=1,N
      N1=N+I
      IF (LT(N1).EQ.1) GO TO 310
      L=N3+I
      ALPHA=X(L)/Z
      H(K+1,I)=ALPHA
      DO 300 J=1,K
      N2=NN+J
      H(J,I)=H(J,I)-X(N2)*ALPHA
  300 CONTINUE
  310 CONTINUE
      DO 320 I=1,N
      N1=N+I
      IF (LT(N1).EQ.1) GO TO 320
      N2=N3+I
      X(N1)=X(N1)+X(N2)**2/Z
  320 CONTINUE
      K=K+1
      H(K,II)=Y
      IF (Y.NE.ONE) II=II+N
      I2=NN+II
      LT(I2)=0
      LT(K)=II
      IF (K.NE.N) GO TO 220
C
C        SET UP RHS OF CONSTRAINTS IN BASIS
C
  330 DO 360 I=1,N
      LI=LT(I)
      N1=N+I
      IF (LI.GT.N) GO TO 340
      X(N1)=BDL(LI)
      GO TO 360
  340 IF (LI.GT.NN) GO TO 350
      L=LI-N
      X(N1)=-BDU(L)
      GO TO 360
  350 LL=LI-NN
      X(N1)=D(LL)
  360 CONTINUE
C
C        CALCULATE POSITION OF VERTEX
C
      DO 370 I=1,N
      X(I)=FM02AS(N,H(1,I),1,X(N+1),1)
  370 CONTINUE
C
C        CALCULATE THE CONSTRAINT RESIDUALS, THE NUMBER OF VIOLATED
C        CONSTRAINTS, AND THE SUM OF THEIR NORMALS
C
  380 KV=0
      DO 390 I=1,N
      N1=N+I
      X(N1)=ZERO
  390 CONTINUE
      DO 460 I=1,M
      N1=NN+I
      IF (LT(N1).LE.0) GO TO 460
```

```
      IF (I.GT.N) GO TO 400
      Z=X(I)-BDL(I)
      GO TO 420
  400 IF (I.GT.NN) GO TO 410
      L=I-N
      Z=BDU(L)-X(L)
      GO TO 420
  410 J=I-NN
      Z=-D(J)+FM02AS(N,C(1,J),1,X(1),1)
  420 X(N1)=Z
      IF (Z.GE.ZERO) GO TO 460
      KV=KV+1
      LT(N1)=2
      IF (I.GT.N) GO TO 430
      N2=N+I
      X(N2)=X(N2)+ONE
      GO TO 460
  430 IF (I.GT.NN) GO TO 440
      X(I)=X(I)-ONE
      GO TO 460
  440 DO 450 II=1,N
      NII=N+II
      X(NII)=X(NII)+C(II,J)
  450 CONTINUE
  460 CONTINUE
      IF (KV.NE.0) GO TO 470
      GO TO 710
C
C        POSSIBLE DIRECTIONS OF SEARCH OBTAINABLE BY REMOVING A
C        CONSTRAINT ARE ROWS OF H,  CALCULATE THE OPTIMUM DIRECTION
C
  470 Z=ZERO
      DO 480 I=1,N
      N1=NN+LT(I)
      IF (LT(N1).EQ.-1) GO TO 480
      Y=FM02AS(N,H(I,1),IH,X(N+1),1)
      IF (Y.LE.Z) GO TO 480
      Z=Y
      II=I
  480 CONTINUE
      IF (Z.LE.ZERO) GO TO 670
C
C        SEARCH FOR THE NEAREST OF THE FURTHEST VIOLATED CONSTRAINT
C        AND THE NEAREST NONVIOLATED NONBASIC CONSTRAINT
C
      ALPHA=RANGE
      BETA=ZERO
      DO 490 I=1,N
      N1=N+I
      X(N1)=H(II,I)
  490 CONTINUE
      DO 540 I=1,M
      N1=NN+I
      IF (LT(N1).LE.0) GO TO 540
      IF (I.GT.N) GO TO 500
      N2=N+I
      Z=-X(N2)
      GO TO 520
  500 IF (I.GT.NN) GO TO 510
      Z=X(I)
      GO TO 520
  510 JJ=I-NN
      Z=-FM02AS(N,X(N+1),1,C(1,JJ),1)
  520 IF (LT(N1).EQ.2) GO TO 530
      IF (Z.LE.ZERO) GO TO 540
```

```
      Z=X(N1)/Z
      IF (Z.GE.ALPHA) GO TO 540
      ALPHA=Z
      IAL=I
      GO TO 540
  530 LT(N1)=1
      IF (Z.GE.ZERO) GO TO 540
      I1=NN+I
      Z=X(I1)/Z
      IF (Z.LE.BETA) GO TO 540
      BETA=Z
      IB=I
  540 CONTINUE
      IF (ALPHA.GT.BETA) GO TO 550
      IB=IAL
      BETA=ALPHA
C
C        EXCHANGE WITH THE CONSTRAINT BEING REMOVED FROM THE BASIS,
C        USING SIMPLEX FORMULA FOR NEW H
C
  550 I1=NN+LT(II)
      LT(I1)=1
      I2=NN+IB
      LT(I2)=0
      LT(II)=IB
      IF (IB.GT.N) GO TO 570
      DO 560 I=1,N
      N1=NN+I
      X(N1)=H(I,IB)
  560 CONTINUE
      GO TO 620
  570 IB=IB-N
      IF (IB.GT.N) GO TO 590
      DO 580 I=1,N
      N1=NN+I
      X(N1)=-H(I,IB)
  580 CONTINUE
      GO TO 620
  590 IB=IB-N
      DO 600 I=1,N
      N1=N3+I
      X(N1)=C(I,IB)
  600 CONTINUE
      DO 610 I=1,N
      N1=NN+I
      X(N1)=FM02AS(N,H(I,1),IH,X(N3+1),1)
  610 CONTINUE
  620 I2=NN+II
      Z=ONE/X(I2)
      DO 660 I=1,N
      N1=N+I
      X(I)=X(I)+BETA*X(N1)
      IF (I.NE.II) GO TO 640
      DO 630 J=1,N
      H(I,J)=H(I,J)*Z
  630 CONTINUE
      GO TO 660
  640 L=NN+I
      ZZ=Z*X(L)
      DO 650 J=1,N
      N2=N+J
      H(I,J)=H(I,J)-ZZ*X(N2)
  650 CONTINUE
  660 CONTINUE
      GO TO 380
```

```
  670 K=0
      IFLAG=1
      IF (LP.GT.0) WRITE (LP,680)
  680 FORMAT (50H ERROR RETURN FROM LA02A/AD ; THERE IS NO SOLUTION)
      GO TO 710
  690 IFLAG=2
      IF (LP.GT.0) WRITE (LP,700)
  700 FORMAT (50H ERROR RETURN FROM LA02A/AD BECAUSE GIVEN EQUALITY,
     1 41H CONSTRAINTS ARE NOT LINEARLY INDEPENDENT)
C
C        RESTORE UNIT NUMBER FOR MESSAGES FROM MB01C
C
  710 LPMB01=LPMB1
      RETURN
      END
C
C
      REAL FUNCTION FM02AS (N,A,IA,B,IB)
      DIMENSION A(N), B(N)
C
C        N    IS THE LENGTH OF THE VECTORS (IF N <= 0  FM02AS/AD = 0)
C        A    IS THE FIRST VECTOR
C        IA   IS SUBSCRIPT DISPLACEMENT BETWEEN ELEMENTS OF A
C        B    IS THE SECOND VECTOR
C        IB   IS SUBSCRIPT DISPLACEMENT BETWEEN ELEMENTS OF B
C        FM02AS/AD  IS THE RESULT
C
      R1=0D0
      IF (N.LE.0) GO TO 20
      JA=1
      IF (IA.LT.0) JA=1-(N-1)*IA
      JB=1
      IF (IB.LT.0) JB=1-(N-1)*IB
      I=0
   10 I=I+1
      R1=R1+A(JA)*B(JB)
      JA=JA+IA
      JB=JB+IB
      IF (I.LT.N) GO TO 10
   20 FM02AS=R1
      RETURN
      END
C
C
      SUBROUTINE MB01C (A,M,IA,IND,C)
      REAL A,AMAX,C,DIV,STO,W,W1,FM02AS,ZERO,ONE
      DIMENSION A(IA,1), IND(1), C(1)
      COMMON /MB01D/ LP,IFLAG
      DATA ZERO,ONE/0.0,1.0/
      DATA LP/0/
      IFLAG=0
      IF (M-1) 310,10,20
   10 IF (A(1,1).EQ.ZERO) GO TO 330
      A(1,1)=ONE/A(1,1)
      GO TO 350
   20 M1=M-1
      AMAX=ZERO
      DO 40 I=1,M
      IND(I)=I
      IF (ABS(A(I,1))-ABS(AMAX)) 40,40,30
   30 AMAX=A(I,1)
      IMAX=I
   40 CONTINUE
      IF (AMAX.EQ.ZERO) GO TO 330
      DO 120 J=1,M1
```

```
      IF (IMAX-J) 70,70,50
 50   IW=IND(IMAX)
      IND(IMAX)=IND(J)
      IND(J)=IW
      DO 60 K=1,M
      W=A(IMAX,K)
      A(IMAX,K)=A(J,K)
      A(J,K)=W
 60   CONTINUE
 70   J1=J+1
      IF (J.EQ.1) GO TO 90
      DO 80 I=J1,M
      A(J,I)=A(J,I)-FM02AS(J-1,A(J,1),IA,A(1,I),1)
 80   CONTINUE
 90   DIV=AMAX
      AMAX=ZERO
      DO 110 I=J1,M
      A(I,J)=A(I,J)/DIV
      A(I,J+1)=A(I,J+1)-FM02AS(J,A(I,1),IA,A(1,J+1),1)
      IF (ABS(A(I,J1))-ABS(AMAX)) 110,110,100
100   AMAX=A(I,J1)
      IMAX=I
110   CONTINUE
      IF (AMAX.EQ.ZERO) GO TO 330
120   CONTINUE
      DO 170 I1=1,M1
      I=M+1-I1
      I2=I-1
      DO 150 J1=1,I2
      J=I2+1-J1
      J2=J+1
      W1=-A(I,J)
      IF (I2-J2) 140,130,130
130   W1=W1-FM02AS(I2-J2+1,A(J2,J),1,C(J2),1)
140   C(J)=W1
150   CONTINUE
      DO 160 K=1,I2
      A(I,K)=C(K)
160   CONTINUE
170   CONTINUE
      DO 260 I1=1,M
      I=M+1-I1
      I2=I+1
      W=A(I,I)
      DO 240 J=1,M
      IF (I-J) 180,190,200
180   W1=ZERO
      GO TO 210
190   W1=ONE
      GO TO 210
200   W1=A(I,J)
210   IF (I1-1) 230,230,220
220   W1=W1-FM02AS(M-I2+1,A(I,I2),IA,A(I2,J),1)
230   C(J)=W1
240   CONTINUE
      DO 250 J=1,M
      A(I,J)=C(J)/W
250   CONTINUE
260   CONTINUE
      DO 300 I=1,M
270   IF (IND(I)-I) 280,300,280
280   J=IND(I)
      DO 290 K=1,M
      STO=A(K,I)
      A(K,I)=A(K,J)
```

```
      A(K,J)=STO
  290 CONTINUE
      ISTO=IND(J)
      IND(J)=J
      IND(I)=ISTO
      GO TO 270
  300 CONTINUE
      GO TO 350
  310 IF (LP.GT.0) WRITE (LP,320)
  320 FORMAT (53H ERROR RETURN FROM MB01C/CD BECAUSE M IS NOT POSITIVE)
      IFLAG=1
      GO TO 350
  330 IF (LP.GT.0) WRITE (LP,340)
  340 FORMAT (54H ERROR RETURN FROM MB01C/CD BECAUSE MATRIX IS SINGULAR)
      IFLAG=2
  350 RETURN
      END
```

SOC-294

MFNC - A FORTRAN PACKAGE FOR MINIMIZATION WITH GENERAL CONSTRAINTS

J.W. Bandler and W.M. Zuberek

June 1982, No. of Pages: 75

Revised:

Key Words: Constrained optimization, nonlinear programming, optimization program, computer-aided design, Han-Powell algorithm

Abstract: MFNC is a package of subroutines for minimization of a nonlinear objective subject to nonlinear constraints. It is an extension and modification of a set of subroutines from the Harwell Subroutine Library (subroutines VF02AD, VF02BD, VF02CD, VE02A, LA02A, MB01C, FM02AS). First derivatives of all functions with respect to all variables are assumed to be available. The solution is found by an iteration that minimizes a quadratic approximation of the objective function subject to linearized constraints. The method was presented by Han and Powell. The package and documentation have been 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 1772 lines, of which 394 are comments.

Related Work: SOC-218, SOC-280, SOC-281, SOC-291, SOC-292.

Price: $100.00.