

INTERNAL REPORTS IN
SIMULATION, OPTIMIZATION
AND CONTROL

No. SOC-294

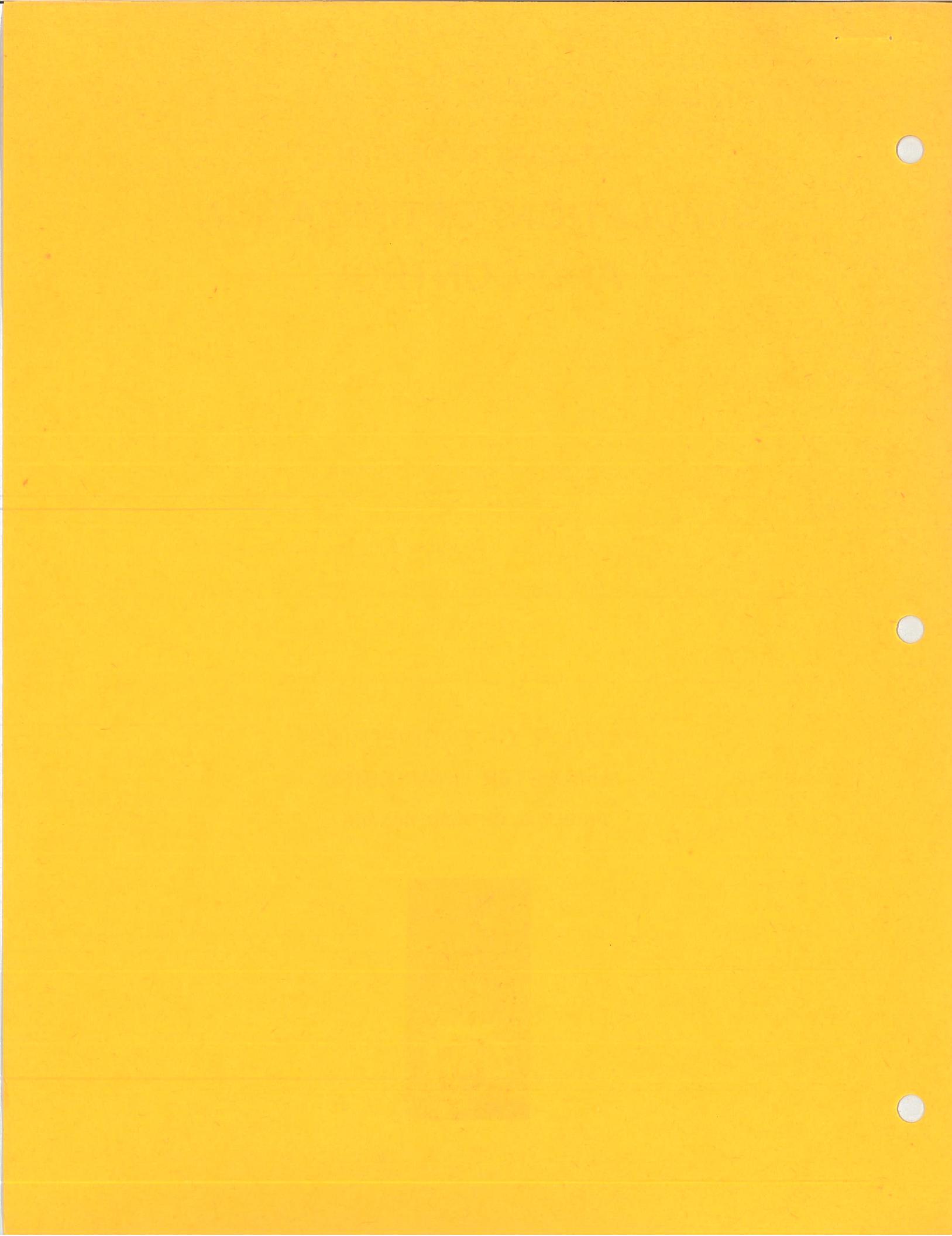
MFNC - A FORTRAN PACKAGE FOR MINIMIZATION
WITH GENERAL CONSTRAINTS

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 RJBAND. 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 $F(\underline{x})$ of n variables, $\underline{x} = [x_1 \dots x_n]^T$, subject to the general equality and inequality constraints

$$f_j(\underline{x}) = 0, \quad j=1, \dots, \ell_{eq},$$
$$f_j(\underline{x}) \geq 0, \quad j=\ell_{eq}+1, \dots, \ell,$$

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 k th iteration the search direction \underline{h}^k is determined as the solution of the linearly constrained quadratic minimization subproblem

$$\text{Minimize } \tilde{F}(\tilde{x}^{k-1}, \tilde{h}^k) = F(\tilde{x}^{k-1}) + \tilde{h}^{kT} \tilde{f}'(\tilde{x}^{k-1}) + 0.5 \tilde{h}^{kT} \tilde{B}^k \tilde{h}^k$$

subject to the constraints

$$\tilde{h}^{kT} \tilde{f}'_j(\tilde{x}^{k-1}) + \alpha^k f_j(\tilde{x}^{k-1}) = 0, \quad j=1, \dots, \ell_{\text{eq}},$$

$$\tilde{h}^{kT} \tilde{f}'_j(\tilde{x}^{k-1}) + \alpha^k f_j(\tilde{x}^{k-1}) \geq 0, \quad j=\ell_{\text{eq}}+1, \dots, \ell,$$

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

where $\tilde{F}'(\tilde{x})$ and $\tilde{f}'_j(\tilde{x})$, $j=1, \dots, \ell$, are the gradient vectors of the objective and constraint functions, respectively, \tilde{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, $\tilde{B}^0 = \mathbb{I}$), and α^k is an additional variable introduced in order to allow infeasibility in linearized constraints, while α_j^k , $j=\ell_{\text{eq}}+1, \dots, \ell$, are defined as

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

Usually the solution of the quadratic subproblem results in $\alpha^k = 1$. If the only feasible solution corresponds to $\alpha^k = 0$ and $\tilde{h}^k = \mathbb{0}$, 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 \tilde{x}^k of the solution

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

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

$$\bar{F}(\underline{x}^{k-1} + \beta^k \underline{h}^k, \underline{\mu}^k) < \bar{F}(\underline{x}^{k-1}, \underline{\mu}^k) ,$$

where

$$\bar{F}(\underline{x}, \underline{\mu}) = F(\underline{x}) + c(\underline{x}, \underline{\mu})$$

and

$$c(\underline{x}, \underline{\mu}) = \sum_{\substack{1 \leq j \leq l \\ eq}} \mu_j |f_j(\underline{x})| + \sum_{\substack{l \\ eq} < j \leq l} \mu_j |\min(0, f_j(\underline{x}))| .$$

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

$$\mu_j^1 = |\lambda_j^1|, \quad j=1, \dots, l,$$

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

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

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

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

$$\bar{F}(\underline{x}^{k-1} + \beta_i^k \underline{h}^k, \underline{\mu}^k) .$$

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

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

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

$$|h^k|^T F'(\tilde{x}^{k-1}) + \sum_{1 \leq j \leq l} |\lambda_j^k f_j(\tilde{x}^{k-1})| \leq \epsilon,$$

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.

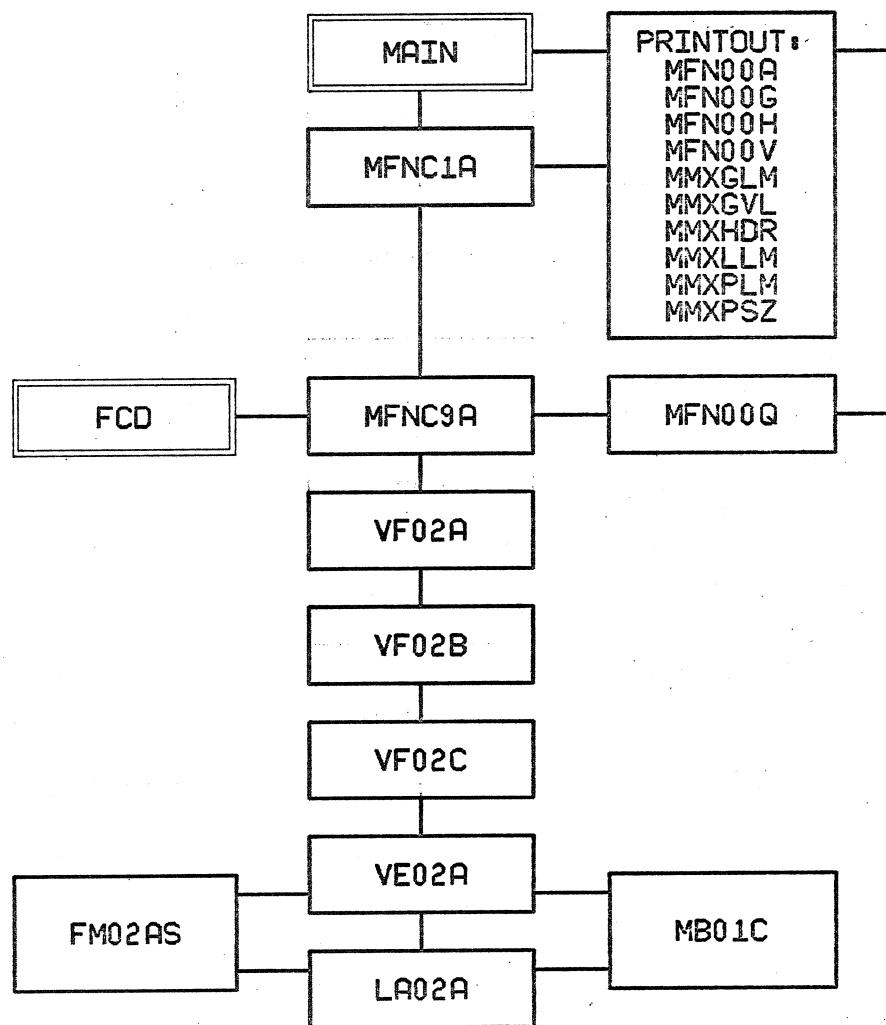


Fig. 1 Structure of the MFNC package corresponding to the standard entry (subroutine MFNC1A).

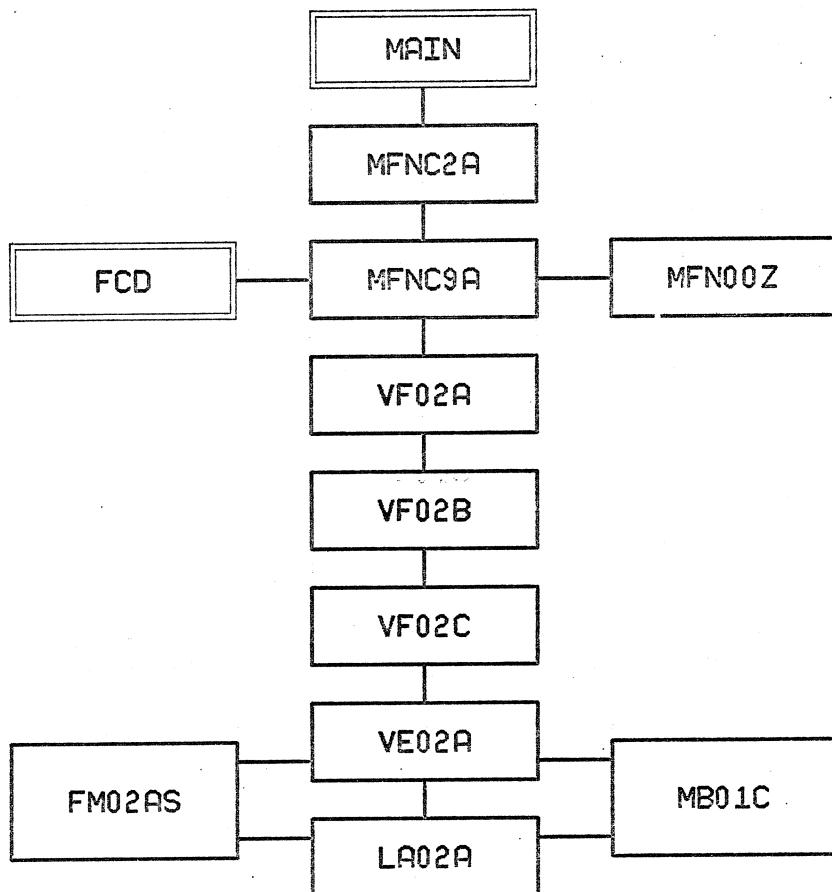


Fig. 2 Structure of the MFNC package corresponding to the basic entry (subroutine MFNC2A).

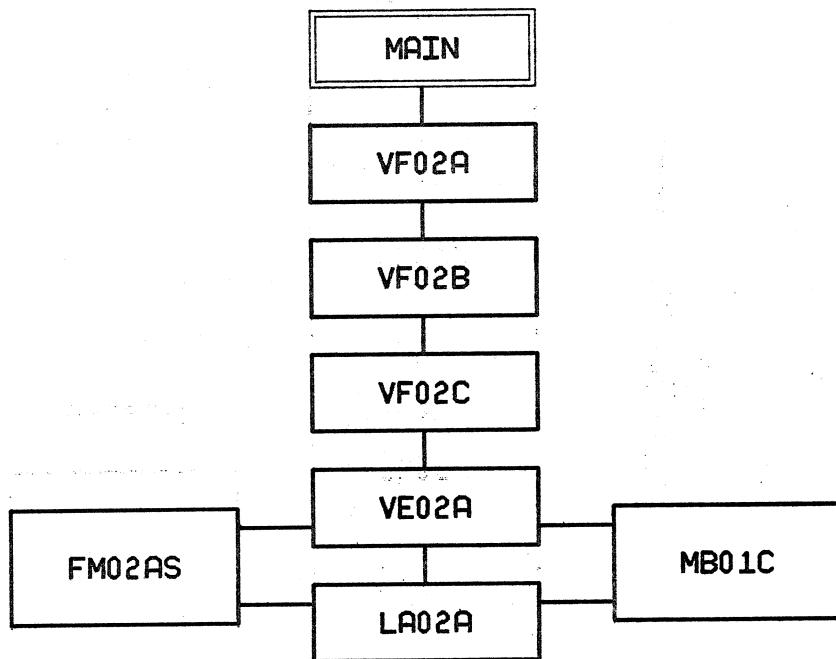


Fig. 3 Structure of the MFNC package corresponding to the original entry (subroutine VF02A).

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

$$G(J) = \partial F(\underline{x})/\partial x_J, \quad J=1, \dots, N,$$
$$C(I) = f_I(\underline{x}), \quad I=1, \dots, L,$$
$$D(J,I) = \partial f_I(\underline{x})/\partial x_J, \quad I=1, \dots, L, \quad J=1, \dots, N.$$

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 must 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, $X(I)=x_I^0$, $I=1,\dots,N$. 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., $W(1)=F(x)$ and $W(I+1)=f_I(x)$, $I=1,\dots,L$.

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. $q=1,\dots,9$) 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

N:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L:	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

N:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L:	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	1672	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 \underline{x}^0 and corresponding function values $f(\underline{x}^0)$ are printed; if, at the same time, s is greater than zero, the values of partial derivatives are included in the "starting point" information. It should be noted that the values of partial derivatives can only be printed for those evaluations for which printing of 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,
MFNOOA, MFNOOG, MFNOOH, MFNOOV, MMXPSZ, MMXPLM,
MMXLIM, MMXHDR, MMXGLM, MMXGVL;
b) for basic entry: MFNC2A, MFNC9A, MFN00Z.

Restrictions: N>0, L>0, LEQ>0, LEQ<L, LEQ<N, EPS>0, MAXF>0, IW>IWR.

Date: May 1982.

VII. EXAMPLES

Example 1 [3]

Minimize

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

subject to the constraints

$$x_1 x_2 - x_3 = 0 ,$$

$$x_3 - 1 \geq 0 .$$

For the starting point

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

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

PROGRAM TRMFN1 (OUTPUT,TAPE6=OUTPUT) 000001
C 000002
C HARWELL TEST PROGRAM. 000003
C 000004
DIMENSION X(3),T(3),W(166) 000005
EXTERNAL FCD 000006
DATA T/10HTRMFN1 : H,10HARWELL EXA,10HMPL. 000007
CALL MMXHDR(3,T) 000008
X(1)=1.0 000009
X(2)=2.0 000010
X(3)=3.0 000011
N=3 000012
LEQ=1 000013
L=2 000014
MAXF=25 000015
EPS=1.0E-7 000016
ICH=6 000017
IPR=-10 000018
LW=166 000019
CALL MFNC1A(FCD,N,L,LEQ,X,EPS,MAXF,W,LW,ICH,IPR,IFLAG) 000020
STOP 000021
END 000022
C 000023
SUBROUTINE FCD (N,M,X,F,G,C,D,K) 000024
DIMENSION X(N),G(N),C(MD),D(K,MD) 000025
X1=X(1) 000026
X2=X(2) 000027
X3=X(3) 000028
F=X1*X1+X2*X2+X3 000029
G(1)=X1+X1 000030
G(2)=X2+X2 000031
G(3)=1.0 000032
C(1)=X1*X2-X3 000033
D(1,1)=X2 000034
D(2,1)=X1 000035
D(3,1)=-1.0 000036
C(2)=X3-1.0 000037
D(1,2)=0.0 000038
D(2,2)=0.0 000039
D(3,2)=1.0 000040
RETURN 000041
END 000042
 000043

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

PAGE : 1
(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 : 8.00000000000E+00

VARIABLES	GRADIENT	CONSTRAINTS
1 1.00000000000E+00	2.0000000000E+00	1 -1.00000000000E+00
2 2.00000000000E+00	4.0000000000E+00	2 2.00000000000E+00
3 3.00000000000E+00	1.0000000000E+00	

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

SOLUTION

OBJECTIVE FUNCTION : 2.99999999999E+00

VARIABLES	GRADIENT	CONSTRAINTS
1 9.99999999993E-01	2.0000000000E+00	1 -4.618527782441E-13
2 1.00000000000E+00	2.0000000000E+00	2 0.
3 1.00000000000E+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

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

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 \underline{x}^0

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

as in [5].

PROGRAM TRMFN2 (OUTPUT, TAPE1=OUTPUT) 000001
C 000002
C BRENT EXAMPLE. 000003
C 000004
DIMENSION X(2),XX(4,2),T(3),W(104) 000005
EXTERNAL FFF 000006
DATA T/10HTRMFN2 : B,10HRENT EXAMP,10HLE 000007
DATA XX/2.0,-2.0,2.0,2.0, 000008
1 2.0,-2.0,0.0,1.0/ 000009
CALL MMXHDR(3,T) 000010
N=2 000011
LEQ=1 000012
L=1 000013
DO 10 I=1,4 000014
X(1)=XX(I,1) 000015
X(2)=XX(I,2) 000016
EPS=1.E-6 000017
MAXF=25 000018
IW=104 000019
ICH=1 000020
IPR=-10 000021
CALL MFNC1A(FFF,N,L,LEQ,X,EPS,MAXF,W,IW,ICH,IPR,IFLAG) 000022
10 CONTINUE 000023
STOP 000024
END 000025
C 000026
C SUBROUTINE FFF(N,L,X,F,G,C,D,K) 000027
DIMENSION X(N),G(N),C(L),D(K,L) 000028
X1=X(1) 000029
X2=X(2) 000030
R1=X1-X2 000031
R2=(X1-2.0)**2+X2*X2 000032
R3=X1+X2 000033
R4=R3+R3 000034
F=R3*R3 000035
G(1)=R4 000036
G(2)=R4 000037
C(1)=R1*R2+3.0*X1+5.0*X2 000038
D(1,1)=R2+(R1+R1)*(X1-2.0)+3.0 000039
D(2,1)=-R2+R1*(X2+X2)+5.0 000040
RETURN 000041
END 000042
 000043

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 : 1.60000000000E+01

	VARIABLES	GRADIENT	CONSTRAINTS
1	2.00000000000E+00	8.0000000000E+00	1 1.60000000000E+01
2	2.00000000000E+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.
MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE)

PAGE : 1
(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 : 1.60000000000E+01

VARIABLES	GRADIENT	CONSTRAINTS
1 -2.00000000000E+00	-8.0000000000E+00	1 -1.60000000000E+01
2 -2.00000000000E+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.
MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE)

PAGE : 1
(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 : 4.00000000000E+00

	VARIABLES	GRADIENT	CONSTRAINTS
1	2.00000000000E+00	4.0000000000E+00	1
2	0.	4.0000000000E+00	6.00000000000E+00

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

SOLUTION

		OBJECTIVE FUNCTION : 3.474781237047E-10	
	VARIABLES	GRADIENT	CONSTRAINTS
1	6.499981749980E-06	-3.7281530210E-05	1
2	-2.514074685511E-05	-3.7281530210E-05	2.035830275945E-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], slightly modified as indicated below. It is to minimize

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

subject to constraints

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

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

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

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

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

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

PROGRAM TRMFN3 (OUTPUT, TAPE6=OUTPUT) 000001
C 000002
C ROSEN-SUZUKI PROBLEM. 000003
C 000004
DIMENSION X(4),XX(4,2),T(3),W(300) 000005
EXTERNAL FCD 000006
DATA T/10HTRMFN3 : R, 10HOSEN-SUZUK, 10HI PROBLEM / 000007
DATA XX/2.0,2.0,5.0,0.0,0.0,0.0,0.0,0.0/ 000008
CALL MMXHDR(3,T) 000009
N=4 000010
LEQ=0 000011
L=3 000012
DO 10 II=1,2 000013
DO 20 JJ=1,4 000014
20 X(JJ)=XX(JJ,II) 000015
MAXF=30 000016
EPS=1.E-6 000017
LW=300 000018
ICH=6 000019
IPR=-10 000020
CALL MFNC1A(FCD,N,L,LEQ,X,EPS,MAXF,W,LW,ICH,IPR,IFLAG) 000021
10 CONTINUE 000022
STOP 000023
END 000024
C 000025
C SUBROUTINE FCD(N,L,X,F,G,C,D,K) 000026
DIMENSION X(N),G(N),C(L),D(K,L) 000027
X1=X(1) 000028
X2=X(2) 000029
X3=X(3) 000030
X4=X(4) 000031
R1=X1+X1+1.0 000032
R2=X2+X2-1.0 000033
R3=X3+X3 000034
R4=X4+X4-1.0 000035
F=X1*(X1-5.0)+X2*(X2-5.0)+X3*(R3-21.0)+X4*(X4+7.0) 000036
C(1)=R1-6.0 000037
C(2)=R2-4.0 000038
C(3)=4.0*X3-21.0 000039
C(4)=X4+X4+7.0 000040
C(1)=X1*(-X1-1.0)+X2*(1.0-X2)+X3*(-X3-1.0)+X4*(1.0-X4)+8.0 000041
C(2)=X1*(1.0-X1)-X2*(X2+X2)-X3*X3+X4*(1.0-X4-X4)+10.0 000042
C(3)=X1*(-X1-2.0)+X2*(1.0-X2)-X3*X3+X4+5.0 000043
D(1,1)=-R1 000044
D(2,1)=-R2 000045
D(3,1)=-R3-1.0 000046
D(4,1)=-R4 000047
D(1,2)=-R1+2.0 000048
D(2,2)=-4.0*X2 000049
D(3,2)=-R3 000050
D(4,2)=1.0-4.0*X4 000051
D(1,3)=-R1-1.0 000052
D(2,3)=-R2 000053
D(3,3)=-R3 000054
D(4,3)=1.0 000055
RETURN 000056
END 000057
000058

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 : -6.70000000000E+01

VARIABLES	GRADIENT	CONSTRAINTS
1 2.00000000000E+00	-1.0000000000E+00	1 -3.00000000000E+01
2 2.00000000000E+00	-1.0000000000E+00	2 -2.50000000000E+01
3 5.00000000000E+00	-1.0000000000E+00	3 -3.00000000000E+01
4 0.	7.0000000000E+00	

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

SOLUTION

OBJECTIVE FUNCTION : -4.40000001041E+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 : -4.40000000131E+01

VARIABLES	GRADIENT	CONSTRAINTS
1 -1.107425820962E-06	-5.0000022149E+00	1 -5.595527892402E-10
2 1.000000404734E+00	-2.999991905E+00	2 9.999990285584E-01
3 2.000000675463E+00	-1.299997298E+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

Example 4 [9, Example 5.5]

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

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

subject to the constraints

$$x_i \geq 0, \quad i=1, \dots, 15,$$

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

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

a_{ij}	j					b_i	
	1	2	3	4	5		
i	1	-16	2	0	1	0	-40
	2	0	-2	0	0.4	2	-2
	3	-3.5	0	2	0	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

		j				
		1	2	3	4	5
c _{ij}		1	2	3	4	5
i	1	30	-20	-10	32	-10
	2	-20	39	-6	-31	32
	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 $F(\underline{x}^*) = 32.34868$, and it is obtained for the starting point \underline{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) 000001
C 000002
C COLVILLE TEST PROBLEM 2. 000003
C 000004
DIMENSION X(15),T(4),W(2000) 000005
EXTERNAL FCD 000006
DATA T/10HTRMFN4 : C,10HOLVILLE TE,10HST PROBLEM,2H 2/ 000007
CALL MMXHDR(4,T) 000008
DO 10 I=1,15 000009
10 X(I)=0.0001 000010
X(12)=60.0 000011
N=15 000012
LEQ=0 000013
L=20 000014
MAXF=50 000015
EPS=1.0E-6 000016
ICH=6 000017
IPR=0 000018
LW=2000 000019
CALL MFNC1A(FCD,N,L,LEQ,X,EPS,MAXF,W,LW,ICH,IPR,IFLAG) 000020
STOP 000021
END 000022
C 000023
C SUBROUTINE FCD(N,L,X,F,C,C,D,KK) 000024
DIMENSION X(N),C(N),C(L),D(KK,L) 000025
DIMENSION A(10,5),B(10),C1(5,5),D1(5),E(5) 000026
DATA A/-16.0,0.0,-3.5,0.0,0.0,2.0,-1.0,-1.0,1.0,1.0, 000027
+ 2.0,-2.0,0.0,-2.0,-9.0,0.0,-1.0,-2.0,2.0,1.0, 000028
+ 0.0,0.0,2.0,0.0,-2.0,-4.0,-1.0,-3.0,3.0,1.0, 000029
+ 1.0,0.4,0.0,-4.0,1.0,0.0,-1.0,-2.0,4.0,1.0, 000030
+ 0.0,2.0,0.0,-1.0,-2.8,0.0,-1.0,-1.0,5.0,1.0/ 000031
DATA B/-40.0,-2.0,-0.25,-4.0,-4.0,-1.0,-40.0,-60.0,5.0,1.0/ 000032
DATA C1/30.0,-20.0,-10.0,32.0,-10.0, 000033
+ -20.0,39.0,-6.0,-31.0,32.0, 000034
+ -10.0,-6.0,10.0,-6.0,-10.0, 000035
+ 32.0,-31.0,-6.0,39.0,-20.0, 000036
+ -10.0,32.0,-10.0,-20.0,30.0/ 000037
DATA D1/4.0,8.0,10.0,6.0,2.0/ 000038
DATA E/-15.0,-27.0,-36.0,-18.0,-12.0/ 000039
C 000040
C OBJECTIVE FUNCTION F 000041
C 000042
F=0.0 000043
DO 10 I=1,10 000044
J=5+I 000045
10 F=F-B(I)*X(J) 000046
DO 30 I=1,5 000047
DO 20 J=1,5 000048
20 F=F+C1(I,J)*X(I)*X(J) 000049
30 F=F+2.0*D1(I)*X(I)**3 000050
C 000051
C GRADIENT G OF THE OBJECTIVE FUNCTION 000052
C 000053
DO 40 I=1,10 000054
J=5+I 000055
40 G(J)=-B(I) 000056
DO 60 I=1,5 000057
G(I)=6.0*D1(I)*X(I)*X(I) 000058
DO 60 J=1,5 000059
60 G(I)=G(I)+X(J)*(C1(I,J)+C1(J,I)) 000060
C 000061
C CONSTRAINTS C 000062
C 000063
DO 70 I=1,15 000064
C 000065

70 C(I)=X(I)	000066
DO 90 J=1,5	000067
K=J+15	000068
C(K)=E(J)+3.0*D1(J)*X(J)*X(J)	000069
DO 80 I=1,5	000070
80 C(K)=C(K)+2.0*C1(I,J)*X(I)	000071
DO 85 II=1,10	000072
JJ=5+II	000073
85 C(K)=C(K)-A(II,J)*X(JJ)	000074
90 CONTINUE	000075
 C	000076
C DERIVATIVES D OF THE CONSTRAINTS	000077
C	000078
DO 100 I=1,15	000079
100 D(I,I)=1.0	000080
DO 110 I=1,15	000081
DO 110 J=1,15	000082
IF(I.NE.J) D(I,J)=0.0	000083
110 CONTINUE	000084
DO 140 J=1,5	000085
K=15+J	000086
DO 120 I=1,5	000087
D(I,K)=2.0*C1(I,J)	000088
IF(I.EQ.J) D(I,K)=D(I,K)+6.0*D1(J)*X(J)	000089
120 CONTINUE	000090
DO 130 I=1,10	000091
II=5+I	000092
130 D(II,K)=-A(I,J)	000093
140 CONTINUE	000094
RETURN	000095
END	000096

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

PAGE : 1
(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

Example 5 [11, Example 5]

The problem is to determine an optimally centered point $\underline{x}^* = [\underline{x}_1^* \quad \underline{x}_2^*]^T$ that maximizes the relative tolerance r in the region R_c defined by the inequalities

$$2 + 2\underline{x}_1 - \underline{x}_2 \geq 0 ,$$

$$143 - 11\underline{x}_1 - 13\underline{x}_2 \geq 0 ,$$

$$-60 + 4\underline{x}_1 + 15\underline{x}_2 \geq 0 ,$$

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

$$R_\epsilon = \{ \underline{x} \mid (1-r) \underline{x}_i^* \leq \underline{x}_i \leq (1+r) \underline{x}_i^*, i=1,2 \}$$

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 \underline{x}_3) and then the vertices of the tolerance region R_ϵ are described by the nonlinear expressions

$$[(1 \pm \underline{x}_3) \underline{x}_1^* \quad (1 \pm \underline{x}_3) \underline{x}_2^*]^T .$$

Since \underline{x}_3 is to be maximized, the objective function can take the form

$$F(\underline{x}) = -\underline{x}_3$$

and it is to be minimized subject to the constraints

$$\begin{aligned} 2 + 2(1 \pm x_3)x_1 - (1 \pm x_3)x_2 &\geq 0, \\ 143 - 11(1 \pm x_3)x_1 - 13(1 \pm x_3)x_2 &\geq 0, \\ -60 + 4(1 \pm x_3)x_1 + 15(1 \pm x_3)x_2 &\geq 0, \\ x_3 &\geq 0. \end{aligned}$$

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

$$\begin{aligned} 2 + 2(1-x_3)x_1 - (1+x_3)x_2 &\geq 0, \\ 143 - 11(1+x_3)x_1 - 13(1+x_3)x_2 &\geq 0, \\ -60 + 4(1-x_3)x_1 + 15(1-x_3)x_2 &\geq 0. \end{aligned}$$

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

```

C PROGRAM TRMFN5 (* OUTPUT, TAPE6=OUTPUT)
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.
MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE)

PAGE : 1
(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

<u>Subroutine</u>	<u>Number of Lines</u> (source text)	<u>Number of Words</u> (compiled code)	<u>Listing from Page</u>
MFNC 1A	93	766	48
MFNC2A	9	77	49
MFNC9A	42	365	49
MFN00Z	8	23	50
MFN00Q	35	210	50
MFN00V	26	245	51
MFN00G	36	261	51
MFN00H	58	460	52
MFN00A	28	150	53
MMXPSZ	12	42	53
MMXPLM	11	37	53
MMXLLM	11	36	53
MMXHDR	16	47	54
MMXGLM	13	44	54
MMXGVL	11	41	54
VF02A	71	261	54
VF02B	261	1216	55
VF02C	122	462	60
VE02A	373	2512	61
LA02A	360	2115	67
FM02AS	25	60	73
MB01C	103	623	73

SUBROUTINE MFNC1A (FCD, N, L, LEQ, X, EPS, MAXF, W, IW, LCH, IPR, IFLAG) 000001
EXTERNAL FCD, MFN00Q 000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065

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 MMXPML (-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 (11H INPUT 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
JC=JC+L
JD=JC+N
IF (LV2.NE.0.OR.LV1.EQ.1) GO TO 30
CALL FCD (N, L, X, W(JF), W(JC), W(JC), W(JD), N)
WRITE (LCH, 20) W(JF)
20 FORMAT (19H STARTING POINT., 20X, 21H OBJECTIVE FUNCTION : ,
1 1PE19.12)
NRL=NRL-1
LML=LML-1
CALL MFN00V (X, N, W(JC), 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
JC1=JF1+L1
JC2=JC1+L1
LW=2*(N*L+N+L+1)
IF (LW.GT.IW) GO TO 80
CALL MFN00H (FCD, N, L, X, W(JF), W(JC), W(JF1), W(JC1), W(JG2), L1)
40 CALL SECOND (TBEG)

```
CALL MFNC9A (MFN00Q,FCD,N,L,LEQ,X,EPS,MAXF,W,IW,IFLAG)          000066
CALL SECOND (TEND)                                                 000067
IF (LCH.LE.0) RETURN                                              000068
IF (IFLAG.EQ.-1) GO TO 100                                         000069
IF (NRL.LT.9) CALL MFN00A                                         000070
WRITE (LCH,50)                                                       000071
50 FORMAT (//9H SOLUTION/9H -----)                                 000072
WRITE (LCH,60) W(1)                                                 000073
60 FORMAT (39X,21HOBJECTIVE FUNCTION : ,1PE19.12/)                000074
NRL=NRL-6                                                       000075
LML=LML-6                                                       000076
CALL MFN00V (X,N,W(L+2),W(2),L)                                  000077
CPU=TEND-TBEG                                                 000078
IF (NRL.LT.9) CALL MFN00A                                         000079
WRITE (LCH,70) IFLAG,MAXF,ITERQ,CPU                                000080
70 FORMAT (//29H TYPE OF SOLUTION (IFLAG) ,24(2H.)),I4//           000081
1 35H NUMBER OF FUNCTION EVALUATIONS ,21(2H. ),I4//              000082
2 35H NUMBER OF QUADRATIC ITERATIONS ,21(2H. ),I4//              000083
3 31H EXECUTION TIME (IN SECONDS) ,21(2H. ),1H.,F7.3/)          000084
RETURN                                                       000085
80 IFLAG=-1                                                 000086
WRITE (LCH,90) LW                                                 000087
90 FORMAT(/55H0INSUFFICIENT WORKSPACE. IT SHOULD HAVE LENGTH AT LEAST
1 ,16/)                                         000088
100 WRITE (LCH,110)                                               000089
110 FORMAT (///40H INCORRECT PARAMETERS)                           000090
RETURN                                                       000091
END                                                       000092
C                                                       000093
C                                                       000094
SUBROUTINE MFNC2A (FCD,N,L,LEQ,X,EPS,MAXF,W,IW,IFLAG)          000095
EXTERNAL FCD,MFN00Z                                             000096
C                                                       000097
C                                                       000098
LEVEL 2 INTERFACE (BASIC ENTRY)                               000099
C                                                       000100
C                                                       000101
DIMENSION X(1), W(1)                                         000102
CALL MFNC9A (MFN00Z,FCD,N,L,LEQ,X,EPS,MAXF,W,IW,IFLAG)        000103
RETURN                                                       000104
C                                                       000105
C                                                       000106
C                                                       000107
SUBROUTINE MFNC9A (FQQ,FCD,N,M,LEQ,X,EPS,MAXF,W,LW,IFLAG)        000108
DIMENSION X(1), W(1)                                         000109
EXTERNAL FQQ,FCD                                             000110
DIMENSION IEE(8)                                              000111
COMMON /LA02B/ LP1,IEE1                                         000112
COMMON /MB01D/ LP2,IEE2                                         000113
COMMON /VP02D/ VLN,LP3                                         000114
COMMON /VE02X/ LP4                                           000115
COMMON /MFN000/ MARK                                         000116
COMMON /MFN111/ LP0                                           000117
DATA LP0/0/,IEE/0,2,-3,1,-2,-4,-1,-1/                         000118
DATA XZERO/0.0/                                              000119
MARK=1                                                       000120
LP1=LP0                                                       000121
LP2=LP0                                                       000122
LP3=LP0                                                       000123
LP4=LP0                                                       000124
LWR=5*N*N+24*N+6*M+N*M+19+MAX0(M,3*N+3)                   000125
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                                000126
IF (LWR.LE.LW) GO TO 30                                         000127
IF (LP0.GT.0) WRITE (LP0,10) LWR                                000128
10 FORMAT(/55H0INSUFFICIENT WORKSPACE. IT SHOULD HAVE LENGTH AT LEAST
1 ,16/)                                         000129
000130
```

20	IFLAG=-1	000131
	RETURN	000132
30	IF=1	000133
	IC=2	000134
	IC= IC+M	000135
	ID= IC+N	000136
	IW= ID+N*M+M	000137
	LL=LW- IW+1	000138
	IE=-1	000139
	NCALL=0	000140
40	CALL FCD (N,M,X,W(IF),W(IC),W(IC),W(ID),N+1)	000141
	NCALL=NCALL+1	000142
	CALL FQQ (N,M,X,W(IF),W(IC),W(IC),W(ID),N+1,NCALL)	000143
	IF (MARK.EQ.0) GO TO 50	000144
	CALL VF02A (N,M,LEQ,X,W(IF),W(IC),W(IC),W(ID),N+1,MAXF,EPS,0,IE,W(IW),LL)	000145
	IF (IE.EQ.0) GO TO 40	000146
	IFLAG= IEE(IE)	000147
	MAXF=NCALL	000148
	RETURN	000149
50	IFLAG=3	000150
	MAXF=NCALL	000151
	RETURN	000152
	END	000153
C		000154
C	SUBROUTINE MFN00Z (N,M,X,F,G,C,D,L,K)	000155
C	DUMMY SUBROUTINE WHICH FOR BASIC ENTRY SUBSTITUTES SUBROUTINE	000156
C	MFN00Q/11Q.	000157
C	DIMENSION X(N), G(N), C(MD), D(L,MD)	000158
	RETURN	000159
	END	000160
C		000161
C	SUBROUTINE MFN00Q (N,M,X,F,G,C,D,L,K)	000162
C	PRINT RESULTS OF FUNCTION EVALUATION.	000163
C	DIMENSION X(N), G(N), C(MD), D(L,MD)	000164
	COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MLX,LMP,LML,LG	000165
1H,DAT,TIM,LHT,H(8)		000166
	IF (LCH.LE.0) RETURN	000167
	IF (LV1+LV2.EQ.0) RETURN	000168
	IF (K.LE.LV2) GO TO 10	000169
	IF (LV1.EQ.0) RETURN	000170
	IF (MOD(K,LV1).NE.0) RETURN	000171
10	IF (NRP.LE.LMP.AND.LML.GE.0) GO TO 30	000172
	LV1=0	000173
	LV2=0	000174
	WRITE (LCH,20)	000175
20	FORMAT (//26H(LISTING LIMIT REACHED)//)	000176
	NRL=NRL-5	000177
	LML=LML-5	000178
	RETURN	000179
30	IF (NRL.LT.7) CALL MFN00A	000180
	WRITE (LCH,40) K,F	000181
40	FORMAT (22H0FUNCTION EVALUATION :,I4,13X,21HOBJECTIVE FUNCTION :,	000182
1	1PE19.12)	000183
	NRL=NRL-2	000184
	LML=LML-2	000185
	CALL MFN00V (X,N,G,C,MD)	000186
	IF (LG1+LG2.EQ.0) RETURN	000187
	IF (K.LE.LG2) GO TO 50	000188
		000189
		000190
		000191
		000192
		000193
		000194
		000195

```
IF (K.LE.LV2) RETURN 000196
IF (LG1.EQ.0) RETURN 000197
IF (MOD(K,LG1).NE.0) RETURN 000198
50 CALL MFN00G (D,N,M,L) 000199
RETURN 000200
END 000201
C 000202
C SUBROUTINE MFN00V (X,N,G,C,MD) 000203
C PRINT VALUES OF VARIABLES AND CONSTRAINTS. 000204
C DIMENSION X(1), G(1), C(1) 000205
COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG 000206
1H,DAT,TIM,LHT,H(8) 000207
IF (LCH.LE.0) RETURN 000208
K=MAX0(N,MD) 000209
IF (NRL.LT.5) CALL MFN00A 000210
WRITE (LCH,10) 000211
10 FORMAT (/17X,9HVARIABLES,11X,8HGRADIENT,20X,11HCONSTRAINTS/) 000212
NRL=NRL-3 000213
LML=LML-3 000214
DO 40 I=1,K 000215
IF (NRL.LE.0) CALL MFN00A 000216
IF (I.LE.N.AND.I.LE.MD) WRITE (LCH,20) I,X(I),G(I),I,C(I) 000217
IF (I.LE.N.AND.I.GT.MD) WRITE (LCH,20) I,X(I),G(I) 000218
IF (I.GT.N.AND.I.LE.MD) WRITE (LCH,30) I,C(I) 000219
20 FORMAT (5X,I4,2X,1PE19.12,2X,1PE17.10,5X,I4,2X,1PE19.12) 000220
30 FORMAT (54X,I4,2X,1PE19.12) 000221
NRL=NRL-1 000222
LML=LML-1 000223
40 CONTINUE 000224
RETURN 000225
END 000226
C
C SUBROUTINE MFN00G (D,N,M,L) 000227
C PRINT PARTIAL DERIVATIVES OF CONSTRAINTS. 000228
C DIMENSION D(L,M) 000229
COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG 000230
1H,DAT,TIM,LHT,H(8) 000231
IF (LCH.LE.0) RETURN 000232
IF (M.LE.0) RETURN 000233
IF (NRL.LT.7) CALL MFN00A 000234
MM=MIN0(M,LMF) 000235
NN=MIN0(N,LMV) 000236
WRITE (LCH,10) 000237
10 FORMAT (43H0 CONSTRAINT DERIVATIVES ( DF.I / DX.J ) :)
NRL=NRL-2 000238
LML=LML-2 000239
DO 60 K=1,NN,LGH 000240
IF (NRL.LT.5) CALL MFN00A 000241
J1=K 000242
J2=MIN0(NN,K+LGH-1) 000243
WRITE (LCH,20) (J,J=J1,J2) 000244
20 FORMAT (1H0,9X,12HVARIBLES(J),10(I5,5X)) 000245
WRITE (LCH,30) 000246
30 FORMAT (8X,14HCONSTRAINTS(I)) 000247
NRL=NRL-3 000248
LML=LML-3 000249
DO 50 I=1,MM 000250
IF (NRL.LE.0) CALL MFN00A 000251
WRITE (LCH,40) I,(D(J,I),J=J1,J2) 000252
20 FORMAT (1H0,9X,12HVARIBLES(J),10(I5,5X)) 000253
WRITE (LCH,30) 000254
30 FORMAT (8X,14HCONSTRAINTS(I)) 000255
NRL=NRL-3 000256
LML=LML-3 000257
DO 50 I=1,MM 000258
IF (NRL.LE.0) CALL MFN00A 000259
WRITE (LCH,40) I,(D(J,I),J=J1,J2) 000260
```

```
40 FORMAT (10X, I6, 4X, 10(1PE10.2)) 000261
NRL=NRL-1 000262
LML=LML-1 000263
50 CONTINUE 000264
60 CONTINUE 000265
RETURN 000266
END 000267
C 000268
C SUBROUTINE MFN00H (FDF, N, M, X, F, DF, G, DG, DH, M1) 000269
C NUMERICAL VERIFICATION OF USER-DEFINED PARTIAL DERIVATIVES 000270
C (VARIABLES ARE DISTURBED ONE BY ONE). 000272
C 000273
C DIMENSION X(N), F(M1), DF(N,M1), G(M1), DG(M1), DH(N,M1) 000274
COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG 000275
1H,DAT,TIM,LHT,H(8) 000276
IF (LCH.LE.0) RETURN 000277
K=0 000278
CALL FDF (N,M,X,F(1),DF(1,1),F(2),DF(1,2),N) 000279
DO 60 I=1,N 000280
Z=X(I) 000281
DX=1.E-6*Z 000282
IF (ABS(DX).LT.1.E-10) DX=1.E-10 000283
DX2=DX+DX 000284
X(I)=Z+DX 000285
CALL FDF (N,M,X,F(1),DH(1,1),F(2),DH(1,2),N) 000286
DO 10 J=1,M1 000287
DG(J)=DH(I,J) 000288
000289
10 CONTINUE 000290
X(I)=Z-DX 000291
CALL FDF (N,M,X,G(1),DH(1,1),G(2),DH(1,2),N) 000292
X(I)=Z 000293
DO 50 J=1,M1 000294
Y=DF(I,J) 000295
Z=F(J)-G(J) 000296
IF (ABS(Z).LE.0.5E-13*(F(J)+G(J))) Z=0.0 000297
Z=Z/DX2 000298
IF (ABS(Y).LE.1.E-20.AND.ABS(Z).LE.1.E-20) GO TO 50 000299
IF (ABS(Z).LT.1.E-20) Z=SIGN(1.E-20,Z) 000300
R=100.0*ABS((Z-Y)/Z) 000301
IF (R.LE.1.0) GO TO 50 000302
IF (SIGN(1.0,DG(J))+SIGN(1.0,DH(I,J)).EQ.0.0) GO TO 50 000303
IF (K.NE.0) GO TO 30 000304
IF (NRL.LT.5) CALL MFN00A 000305
WRITE (LCH,20) 000306
20 FORMAT(38H0VERIFICATION OF PARTIAL DERIVATIVES ://)
1 1H0,18X,52H DF. I / DX.J : USER DEFINED NUMERICAL DIFFERENCE) 000307
NRL=NRL-4 000308
LML=LML-4 000309
30 K=K+1 000310
IF (NRL.LE.0) CALL MFN00A 000311
L=J-1 000312
WRITE (LCH,40) L,I,Y,Z,R 000313
40 FORMAT (19X, I5, 3X, I4, 5X, 1PE10.3, 2X, 1PE10.3, 4X, 0PF6.1, 2H %) 000314
NRL=NRL-1 000315
LML=LML-1 000316
50 CONTINUE 000317
60 CONTINUE 000318
IF (K.NE.0) GO TO 80 000319
IF (NRL.LT.2) CALL MFN00A 000320
WRITE (LCH,70) 000321
70 FORMAT (47H0VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.) 000322
NRL=NRL-2 000323
LML=LML-2 000324
000325
```

```
80 RETURN 000326
END 000327
C 000328
C SUBROUTINE MFN00A 000329
C CHANGE PAGE AND PRINT PAGE HEADER. 000330
C 000331
C COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG 000332
1H,DAT,TIM,LHT,H(8) 000333
IF (LCH.LE.0) RETURN 000334
IF (NRP.LT.LMP) GO TO 20 000335
LV1=0 000336
LV2=0 000337
WRITE (LCH,10) 000338
10 FORMAT (/27H ( LIMIT OF PAGES REACHED )) 000339
20 NRP=NRP+1 000340
NRL=MXL-5 000341
LML=LML-5 000342
WRITE (LCH,30) DAT,TIM,NRP 000343
30 FORMAT (1H1/7H DATE :,A10,19X,6HTIME :,A10,20X,6HPAGE :,I3// 000344
1 55H MINIMIZATION WITH NONLINEAR CONSTRAINTS (MFNC PACKAGE),17X, 000345
2 9H(V:82.05)) 000346
IF (LHT.LE.0) GO TO 50 000347
WRITE (LCH,40) (H(J),J=1,LHT) 000348
40 FORMAT (1H0,8A10) 000349
NRL=NRL-2 000350
LML=LML-2 000351
50 WRITE (LCH,60) 000352
60 FORMAT (/1X) 000353
RETURN 000354
END 000355
C 000356
C SUBROUTINE MMXPSZ (L) 000357
C 000358
C DEFINE THE PAGE SIZE ( I.E. THE NUMBER OF LINES PER PAGE). 000359
C 000360
C COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG 000361
1H,DAT,TIM,LHT,H(8) 000362
DATA LL/65/ 000363
IF (L.GT.0) LL=MAX0(25,L) 000364
IF (L.EQ.0) LL=0 000365
MXL=LL 000366
RETURN 000367
END 000368
C 000369
C SUBROUTINE MMXPLM (L) 000370
C 000371
C DEFINE THE LIMIT OF PRINTED PAGES. 000372
C 000373
C COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG 000374
1H,DAT,TIM,LHT,H(8) 000375
DATA LL/10/ 000376
IF (L.GT.0) LL=MIN0(50,L) 000377
LMP=LL 000378
RETURN 000379
END 000380
C 000381
C SUBROUTINE MMXLLM (L) 000382
C 000383
C DEFINE THE LIMIT OF PRINTED LINES. 000384
C 000385
C 000386
C 000387
C 000388
C 000389
C 000390
```

```
COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG 000391
1H,DAT,TIM,LHT,H(8) 000392
DATA LL/750/ 000393
IF (L.GT.0) LL=L 000394
LML=LL 000395
RETURN 000396
END 000397
000398
C 000399
C SUBROUTINE MMXHDR (L,T) 000400
C DEFINE THE HEADER LINE. 000401
C 000402
C DIMENSION T(1) 000403
COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG 000404
1H,DAT,TIM,LHT,H(8) 000405
DATA LL/0/ 000406
IF (L.GE.0) LL=MIN0(8,L) 000407
LHT=LL 000408
IF (L.LE.0) RETURN 000409
DO 10 I=1,LL 000410
H(I)=T(I) 000411
10 CONTINUE 000412
RETURN 000413
END 000414
000415
C 000416
C SUBROUTINE MMXGLM (K,L) 000417
C DEFINE THE SIZE OF PRINTED JACOBIAN. 000418
C 000419
C COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG 000420
1H,DAT,TIM,LHT,H(8) 000421
DATA KK/25/,LL/10/ 000422
IF (K.GT.0) KK=K 000423
IF (L.GT.0) LL=L 000424
LMF=KK 000425
LMV=LL 000426
RETURN 000427
END 000428
000429
000430
000431
C 000432
C SUBROUTINE MMXGVL (L) 000433
C DEFINE THE NUMBER OF JACOBIAN COLUMNS PRINTED IN ONE LINE. 000434
C 000435
C COMMON /MMX000/ LCH,LV1,LV2,LG1,LG2,LMF,LMV,NRP,NRL,MXL,LMP,LML,LG 000436
1H,DAT,TIM,LHT,H(8) 000437
DATA LL/10/ 000438
IF (L.GT.0) LL=MAX0(MIN0(10,L),5) 000439
LCH=LL 000440
RETURN 000441
END 000442
000443
000444
000445
000446
000447
000448
000449
000450
000451
000452
000453
000454
000455
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 000449
C M IS THE TOTAL NUMBER OF CONSTRAINTS 000450
C MEQ IS THE NUMBER OF EQUALITY CONSTRAINTS 000451
C X IS THE VECTOR OF VARIABLES 000452
C IT MUST BE SET BY THE USER BEFORE THE INITIAL CALL AND LEFT 000453
C UNCHANGED THEREAFTER. 000454
000455
```

C F IS THE VALUE OF THE OBJECTIVE FUNCTION 000456
C G IS THE GRADIENT OF THE OBJECTIVE FUNCTION 000457
C C IS THE VECTOR OF CONSTRAINT FUNCTIONS 000458
C CN IS THE MATRIX OF CONSTRAINT NORMALS 000459
C LCN IS THE FIRST DIMENSION OF CN 000460
C
C F, G, C, CN MUST ALL BE SET BY THE USER BEFORE EACH CALL. 000461
C
C MAXFUN BOUNDS THE NUMBER OF CALLS OF VF02A/AD 000462
C ACC CONTROLS THE FINAL ACCURACY - THE CALCULATION ENDS WHEN THE 000463
C OBJECTIVE FUNCTION PLUS SUITABLY WEIGHTED MULTIPLES OF THE 000464
C CONSTRAINT FUNCTIONS ARE PREDICTED TO DIFFER FROM THEIR 000465
C OPTIMAL VALUES BY AT MOST ACC 000466
C IPRINT CONTROLS THE AMOUNT OF PRINTING 000467
C .LT. 0 NO PRINTING 000468
C .EQ. 0 DIAGNOSTICS ONLY 000469
C .GT. 0 X, F AND C AT START OF EVERY IPRINT ITERATIONS 000470
C INF CONTROLS THE CALCULATION 000471
C =-1 ON INITIAL CALL 000472
C =0 DURING CALCULATION 000473
C =1 WHEN REQUIRED ACCURACY IS ACHIEVED 000474
C =2 WHEN VF02A/AD IS CALLED MAXFUN TIMES 000475
C =3 WHEN A LINE SEARCH REQUIRES 5 CALLS OF VF02A/AD 000476
C =4 WHEN AN UPHILL SEARCH DIRECTION IS CALCULATED 000477
C =5 WHEN NO FEASIBLE POINT IS FOUND BY VF02C/CD 000478
C =6 WHEN AN ARTIFICIAL BOUND RESTRICTS VF02C/CD 000479
C =7 WHEN LW IS TOO SMALL 000480
C =8 WHEN N, M OR MEQ HAS A SILLY VALUE 000481
C W IS A WORKSPACE ARRAY OF LENGTH LW 000482
C LW IS THE LENGTH OF ARRAY W 000483
C
C COMMON /VF02D/ VLARGE, LP 000484
DATA VLARGE/1.0E6/, LP/0/ 000485
C
C LP IS THE UNIT NUMBER FOR PRINTING MESSAGES 000486
C
C IW=5*N*N+23*N+18+4*M+MAX0(M, 3*N+3) 000487
IF (IW.LE.LW) GO TO 20 000488
IF (IPRINT.GE.0. AND. LP.GT.0) WRITE (LP, 10) IW 000489
10 FORMAT (49H ERROR RETURN FROM VF02A/AD BECAUSE W SHOULD HAVE, 000490
1 16H LENGTH AT LEAST, 18) 000491
INF=7 000492
GO TO 50 000493
20 IF (N.GT.0. AND. M.GE. MEQ. AND. MEQ.GE.0) GO TO 40 000494
IF (IPRINT.GE.0. AND. LP.GT.0) WRITE (LP, 30) N, M, MEQ 000495
30 FORMAT (38H ERROR RETURN FROM VF02A/AD BECAUSE N=, I6.5H M=, I6, 000496
1 7H MEQ=, I6) 000497
INF=8 000498
GO TO 50 000499
40 IVLAM=1 000500
IVMU= IVLAM+M 000501
IB= IVMU+M 000502
IDELTA= IB+(N+1)*(N+1) 000503
ICLAG= IDELTA+N+1 000504
IGLAGA= ICLAG+N 000505
IXA= IGLAGA+N 000506
CALL VF02B (N, M, MEQ, X, F, G, C, CN, LCN, MAXFUN, ACC, IPRINT, INF, W, LW, N+1 000507
1, W(IVLAM), W(IVMU), W(IB), W(IDELTA), W(ICLAG), W(IGLAGA), W(IGLACA), W(I 000508
2GLAGA), W(IXA), W(IXA)) 000509
50 RETURN 000510
END 000511
C
C SUBROUTINE VF02B (N, M, MEQ, X, F, G, C, CN, LCN, MAXFUN, ACC, IPRINT, INF, W, 000512
1LW, NP, VLAM, VMU, B, DELTA, GLAG, GLACA, GAMMA, ETA, XA, BDELTAA) 000513
000514
000515
000516
000517
000518
000519
000520

```
REAL X(N), F, G(N), C(M), CN(LCN, M), ACC, W(LW)          000521
DIMENSION VLAM(M), VMU(M), DELTA(NP), GLAG(N), GLAGA(N), XA(N), GAMMA(N) 000522
1, BDELTA(N), ETA(N), B(NP, NP)                           000523
C
C      VLAM   IS THE VECTOR OF LAGRANGE MULTIPLIERS           000524
C      VMU    HOLDS THE PARAMETERS FOR THE LINE SEARCH FUNCTION 000525
C      DELTA  IS THE SEARCH DIRECTION TIMES THE STEP-LENGTH     000526
C      GLAG   IS THE GRADIENT OF THE LAGRANGIAN FUNCTION        000527
C      GLAGA  IS THIS GRADIENT AT THE START OF AN ITERATION     000528
C      XA    IS THE VECTOR OF VARIABLES AT THE START OF AN ITERATION 000529
C      GAMMA IS THE CHANGE IN GRADIENT OF THE LAGRANGIAN FUNCTION 000530
C      BDELTA IS B TIMES DELTA                                000531
C      ETA    REPLACES GAMMA IN THE B-F-C-S FORMULA FOR REVISING B 000532
C      B      IS THE VARIABLE METRIC MATRIX                   000533
C
C      (GLAGA, GAMMA, ETA) AND (XA, BDELTA) ARE ESSENTIALLY EQUIVALENT, 000534
C      SINCE THEY ARE ASSOCIATED WITH THE SAME PART OF ARRAY W.       000535
C
C      COMMON /VF02D/ VLARGE, LP                               000536
C      COMMON /VF02E/ FLS, SUM, FLSA, ALPHA, DFLSA, SPGDEL, NF, ITER, ITERP, NFIN 000537
1 IT, MACT
DATA XZERO, XONE, XHALF, XZONE, XZTWO/0.0E0, 1.0E0, 0.5E0, 0.1E0, 0.2E0/ 000538
C
C      SET SOME PARAMETERS FOR THE CALCULATION               000539
C      NFLINE CONTROLS THE ERROR RETURN FROM THE LINE SEARCH 000540
C      PARACC IS THE SLOPE OF THE ARMJO CHORD              000541
C      PARSTP LIMITS THE REDUCTION IN THE LINE SEARCH STEP-LENGTH 000542
C      PARB   BOUNDS THE REDUCTION IN THE DETERMINANT OF B    000543
C
C      NFLINE=5                                         000544
C      PARACC=XZONE                                     000545
C      PARSTP=XZONE                                     000546
C      PARB=XZTWO                                      000547
C      IF (INF.EQ.0) GO TO 270                         000548
C
C      SET INITIAL VALUES OF SOME VARIABLES               000549
C      NF    IS THE NUMBER OF CALLS OF VF02A/AD          000550
C      ITER  IS THE ITERATION NUMBER                     000551
C      ITERP IS THE NEXT ITERATION ON WHICH PRINTING OCCURS 000552
C
C      NF=1                                           000553
C      ITER=0                                         000554
C      ITERP=MIN0(1, IPRINT)                         000555
C
C      SET THE INITIAL ELEMENTS OF B AND VMU            000556
C
C      DO 20 I=1,N                                     000557
C      DO 10 J=1,N                                     000558
C      B(I,J)=XZERO                                 000559
10 CONTINUE
C      B(I,I)=XONE                                 000560
20 CONTINUE
C      IF (M.EQ.0) GO TO 40                         000561
C      DO 30 K=1,M                                     000562
C      VMU(K)=XZERO                                 000563
30 CONTINUE
C
C      START THE ITERATION BY PROVIDING PRINTING        000564
C
C      40 ITER=ITER+1                                000565
C      IF (ITER.NE.ITERP) GO TO 100                  000566
C      50 IF (LP.GT.0) WRITE (LP,60) ITER, NF          000567
C      60 FORMAT (/5X, 12HITERATIONS =, I5, 5X, 19HCALLS OF VF02A/AD =, I5) 000568
C      IF (LP.GT.0) WRITE (LP,70) (X(I), I=1, N)      000569
C      70 FORMAT (/5H X =, 5E20.10/(5X, 5E20.10))    000570
C
```

```
      IF (LP.GT.0) WRITE (LP,80) F          000586
80 FORMAT (/5H F =,E20.10)                000587
      IF (M.GT.0.AND.LP.GT.0) WRITE (LP,90) (C(K),K=1,M) 000588
90 FORMAT (/5H C =,5E20.10/(5X,5E20.10))    000589
      IF (ITER.NE.ITERP) GO TO 440        000590
      ITERP=ITER+IPRINT                  000591
C
C       CALL THE QUADRATIC PROGRAMMING SUBROUTINE
C
100 IDELTA=1+2*M+NP*NP                  000592
C
C       THIS CHOICE OF IDELTA MEANS THAT ARRAYS DELTA OF THIS SUBROUTINE
C       AND OF VF02C/CD ARE DYNAMICALLY EQUIVALENT
C
110 LDELTA=4*NP+MAX0(M,3*NP)            000593
      IGM= IDELTA+LDELTA                000594
      ICM= IGM+NP                      000595
      IBDL= ICM+M                      000596
      IBDU= IBDL+N+1                  000597
      IH= IBDU+NP                      000598
      ILT= IH+NP*NP*4                  000599
      LLT=M+NP*6                      000600
      CALL VF02C (N,M,MEQ,G,C,CN,LGN,IPRINT,INF,NP,2*NP,VLAM,B,W( IDELTA
1),LDELTA,W(IGM),W(ICM),W(IBDL),W(IBDU),W(IH),W(ILT),LLT) 000601
      IF (INF.LE.1) GO TO 120          000602
      IF (IPRINT.LT.0) GO TO 460        000603
      IF (LP.GT.0) WRITE (LP,110)        000604
110 FORMAT (/5X,24HERROR CONDITION FOUND IN,
1 34H QUADRATIC PROGRAMMING CALCULATION)
      GO TO 430                        000605
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                          000606
130 DO 140 I=1,N                      000607
      GLAG(I)=G(I)                    000608
140 CONTINUE
      IF (M.EQ.0) GO TO 170          000609
      DO 160 K=1,M                  000610
      IF (VLAM(K).EQ.XZERO) GO TO 160 000611
      DO 150 I=1,N                  000612
      GLAG(I)=GLAG(I)-CN(I,K)*VLAM(K) 000613
150 CONTINUE
160 CONTINUE
170 IF (NF.NE.NFINIT) GO TO 350        000614
C
C       STORE THE ELEMENTS OF GLAG AND X
C       SET SPGDEL TO THE SCALAR PRODUCT OF G AND DELTA
C
180 SPGDEL=XZERO                      000615
      DO 180 I=1,N                  000616
      SPGDEL=SPGDEL+G(I)*DELTA(I)    000617
      GLAGA(I)=GLAG(I)              000618
      XA(I)=X(I)                    000619
180 CONTINUE
C
C       REVISE THE VECTOR VMU
C       TEST FOR CONVERGENCE
C
      SUM=ABS(SPGDEL)              000620
      IF (M.LE.0) GO TO 200          000621
      DO 190 K=1,M                  000622
      AUX=ABS(VLAM(K))             000623
      VMU(K)=AMAX1(AUX,XHALF*(AUX+VMU(K))) 000624
190 CONTINUE                            000625
      IF (VMU(K).LT.1.E-10) GO TO 200 000626
      VMU(K)=VMU(K)/XHALF*(AUX+VMU(K)) 000627
      VMU(K)=VMU(K)+XHALF*(AUX+VMU(K)) 000628
      VMU(K)=VMU(K)/XHALF*(AUX+VMU(K)) 000629
      VMU(K)=VMU(K)+XHALF*(AUX+VMU(K)) 000630
      VMU(K)=VMU(K)/XHALF*(AUX+VMU(K)) 000631
      VMU(K)=VMU(K)+XHALF*(AUX+VMU(K)) 000632
      VMU(K)=VMU(K)/XHALF*(AUX+VMU(K)) 000633
      VMU(K)=VMU(K)+XHALF*(AUX+VMU(K)) 000634
      VMU(K)=VMU(K)/XHALF*(AUX+VMU(K)) 000635
      VMU(K)=VMU(K)+XHALF*(AUX+VMU(K)) 000636
      VMU(K)=VMU(K)/XHALF*(AUX+VMU(K)) 000637
      VMU(K)=VMU(K)+XHALF*(AUX+VMU(K)) 000638
      VMU(K)=VMU(K)/XHALF*(AUX+VMU(K)) 000639
      VMU(K)=VMU(K)+XHALF*(AUX+VMU(K)) 000640
      VMU(K)=VMU(K)/XHALF*(AUX+VMU(K)) 000641
      VMU(K)=VMU(K)+XHALF*(AUX+VMU(K)) 000642
      VMU(K)=VMU(K)/XHALF*(AUX+VMU(K)) 000643
      VMU(K)=VMU(K)+XHALF*(AUX+VMU(K)) 000644
      VMU(K)=VMU(K)/XHALF*(AUX+VMU(K)) 000645
      VMU(K)=VMU(K)+XHALF*(AUX+VMU(K)) 000646
      VMU(K)=VMU(K)/XHALF*(AUX+VMU(K)) 000647
      VMU(K)=VMU(K)+XHALF*(AUX+VMU(K)) 000648
      VMU(K)=VMU(K)/XHALF*(AUX+VMU(K)) 000649
      VMU(K)=VMU(K)+XHALF*(AUX+VMU(K)) 000650
```

```
SUM=SUM+ABS(VLAM(K)*C(K))          000651
190 CONTINUE                         000652
200 IF (SUM.GT.ACC) GO TO 320        000653
      INF=1                           000654
      GO TO 430                      000655
C                                     000656
C             SET THE INITIAL CONDITIONS FOR THE LINE SEARCH
C             FLSA   IS THE INITIAL VALUE OF THE LINE SEARCH FUNCTION 000657
C             DFLSA  IS USUALLY ITS FIRST DERIVATIVE                 000658
C             ALPHA   IS THE NEXT REDUCTION IN THE STEP-LENGTH       000659
C                                     000660
C                                     000661
210 FLSA=FLS                         000662
      DFLSA=SPGDEL-DELTA(N+1)*SUM    000663
      IF (DFLSA.GE.XZERO) GO TO 410  000664
      ALPHA=XONE                     000665
C                                     000666
C             MULTIPLY DELTA BY ALPHA AND CALCULATE THE NEW X
C                                     000667
C                                     000668
220 DO 230 I=1,N                     000669
      DELTA(I)=ALPHA*DELTA(I)        000670
      X(I)=XA(I)+DELTA(I)           000671
230 CONTINUE                          000672
      DFLSA=ALPHA*DFLSA              000673
C                                     000674
C             TEST NF AGAINST MAXFUN AND RETURN FOR MORE VALUES OF F,G,C,CN
C                                     000675
C                                     000676
      IF (NF.LT.MAXFUN) GO TO 260   000677
240 DO 250 I=1,N                     000678
      X(I)=XA(I)                   000679
250 CONTINUE                          000680
      IF (NF.EQ.NFINIT) GO TO 280  000681
260 NF=NF+1                          000682
      GO TO 460                    000683
270 IF (NF.LE.MAXFUN) GO TO 300   000684
280 INF=2                            000685
      IF (IPRINT.LT.0) GO TO 460   000686
      IF (LP.GT.0) WRITE (LP,290) NF 000687
290 FORMAT (/5X,33HERROR RETURN FROM VF02A/AD DUE TO,I5,
      1 21H FUNCTION EVALUATIONS)  000688
      GO TO 430                    000689
C                                     000690
C             TEST FOR ERROR RETURN FROM LINE SEARCH
C                                     000691
C                                     000692
300 IF (NF.LE.NFINIT+NFLINE) GO TO 320 000693
      INF=3                          000694
      IF (IPRINT.LT.0) GO TO 460   000695
      IF (LP.GT.0) WRITE (LP,310) NFLINE 000696
310 FORMAT (/5X,49HRETURN FROM VF02A/AD BECAUSE LINE SEARCH REQUIRES,
      1 10H MORE THAN,13,6H STEPS)  000697
      GO TO 430                    000698
C                                     000699
C                                     000700
C             SET SUM TO THE WEIGHTED SUM OF INFEASIBILITIES
C             SET FLS TO THE LINE SEARCH OBJECTIVE FUNCTION
C                                     000701
C                                     000702
C                                     000703
C                                     000704
320 SUM=XZERO                         000705
      IF (M.LE.0) GO TO 340          000706
      DO 330 K=1,M                  000707
      AUX=C(K)                      000708
      IF (K.GT.MEQ) AUX=XZERO       000709
      SUM=SUM+VMU(K)*AMAX1(AUX,-C(K)) 000710
330 CONTINUE                          000711
340 FLS=F+SUM                         000712
      IF (NF.EQ.NFINIT) GO TO 210  000713
C                                     000714
C             CALCULATE THE GRADIENT OF THE LAGRANGIAN FUNCTION 000715
```

C IF THE LINE SEARCH IS COMPLETE 000716
C 000717
C AUX=FLS-FLSA 000718
IF (AUX.LE.PARACC*DFLSA) GO TO 130 000719
IF (NF.GE.NFINIT+NFLINE) GO TO 240 000720
C 000721
C CALCULATE THE NEXT REDUCTION IN THE STEP-LENGTH 000722
C 000723
C ALPHA=AMAX1(PARSTP,XHALF*DFLSA/(DFLSA-AUX)) 000724
GO TO 220 000725
C 000726
C CALCULATE GAMMA AND BDELTA IN ORDER TO REVISE B 000727
C SET DG TO THE SCALAR PRODUCT OF DELTA AND GAMMA 000728
C SET DBD TO THE SCALAR PRODUCT OF DELTA AND BDELTA 000729
C 000730
350 DG=XZERO 000731
DBD=XZERO 000732
DO 370 I=1,N 000733
GAMMA(I)=GLAG(I)-GLAGA(I) 000734
BDELTA(I)=XZERO 000735
DO 360 J=1,N 000736
BDELTA(I)=BDELTA(I)+B(I,J)*DELTA(J) 000737
360 CONTINUE 000738
DG=DG+DELTA(I)*GAMMA(I) 000739
DBD=DBD+DELTA(I)*BDELTA(I) 000740
370 CONTINUE 000741
C 000742
C CALCULATE THE VECTOR ETA FOR THE B-F-G-S FORMULA 000743
C REPLACE DG BY THE SCALAR PRODUCT OF DELTA AND ETA 000744
C 000745
AUX=PARB*DBD 000746
IF (DG.GE.AUX) GO TO 390 000747
THCOMP=(DBD-AUX)/(DBD-DG) 000748
THCOMP=XONE-THETA 000749
DO 380 I=1,N 000750
ETA(I)=THETA*GAMMA(I)+THCOMP*BDELTA(I) 000751
380 CONTINUE 000752
DG=AUX 000753
C 000754
REVISE THE MATRIX B 000755
C 000756
390 DO 400 I=1,N 000757
AUX=BDELTA(I)/DBD 000758
AUXA=ETA(I)/DG 000759
DO 400 J=I,N 000760
B(I,J)=B(I,J)-AUX*BDELTA(J)+AUXA*ETA(J) 000761
400 B(J,I)=B(I,J) 000762
GO TO 40 000763
C 000764
TEST IF ANY PRINTING IS REQUIRED BEFORE THE RETURN 000765
C 000766
410 INF=4 000767
IF (IPRINT.LT.0) GO TO 460 000768
IF (LP.GT.0) WRITE (LP,420) 000769
420 FORMAT (/5X,43HERROR RETURN FROM VF02A/AD DUE TO AN UPHILL, 000770
1 17H SEARCH DIRECTION/5X,30HUSUALLY THIS IS DUE TO LOSS OF, 000771
2 50H ACCURACY IN THE QUADRATIC PROGRAMMING CALCULATION) 000772
430 IF (IPRINT.LE.0) GO TO 460 000773
IF (ITERP.NE.ITER+IPRINT) GO TO 50 000774
440 IF (LP.GT.0) WRITE (LP,450) 000775
450 FORMAT (/5X,44HTHE PRINTING OF THE LAST ITERATION GIVES THE, 000776
1 48H VALUES THAT ARE RETURNED BY SUBROUTINE VF02A/AD) 000777
460 RETURN 000778
END 000779
C 000780

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(MD, B(NP, NP)), DELTA(LDELTA), GM(NP
1), CM(MD, 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
C MODE=1
C VSMALL=XONEM6
C FEASP=XZNINE
C MTOTAL=M+NPP
C NSIX=6*NP
C IF (INF.GE.0) GO TO 50
C
C SET INITIAL VALUES OF SOME VARIABLES
C
C INF=0
C MACT=MEQ+1
C
C SET THE INITIAL ELEMENTS OF BDL, BDU, DELTA AND LT
C
C DO 10 I=1,N
C BDL(I)=-VLARGE
C BDU(I)=VLARGE
C DELTA(I)=XZERO
10 CONTINUE
C BDL(NP)=XZERO
C DELTA(NP)=XONE
C IF (MEQ.LE.0) GO TO 30
C DO 20 K=1,MEQ
C LT(K)=K+NPP
20 CONTINUE
C 30 LT(MACT)=NPP
C
C EXTEND GM AND B BECAUSE OF THE EXTRA VARIABLE THAT IS
C INTRODUCED TO ALLOW FOR INFEASIBILITY
C
C GM(NP)=VLARGE
C DO 40 I=1,NP
C B(I,NP)=XZERO
C B(NP,I)=XZERO
40 CONTINUE
C
C SET THE ELEMENTS OF GM, CM AND CN(NP,*)
C
C 50 DO 60 I=1,N
C GM(I)=-G(I)
60 CONTINUE
C IF (M.LE.0) GO TO 90

000781
000782
000783
000784
000785
000786
000787
000788
000789
000790
000791
000792
000793
000794
000795
000796
000797
000798
000799
000800
000801
000802
000803
000804
000805
000806
000807
000808
000809
000810
000811
000812
000813
000814
000815
000816
000817
000818
000819
000820
000821
000822
000823
000824
000825
000826
000827
000828
000829
000830
000831
000832
000833
000834
000835
000836
000837
000838
000839
000840
000841
000842
000843
000844
000845

```
DO 80 K=1,M          000846
IF (K.LE.MEQ) GO TO 70 000847
IF (C(K).LT.XZERO) GO TO 70 000848
CM(K)=-C(K)           000849
CN(NP,K)=XZERO       000850
GO TO 80              000851
70 CM(K)=XZERO       000852
CN(NP,K)=C(K)         000853
80 VLAM(K)=XZERO     000854
C
C      CALL SUBROUTINE VE02A/AD 000855
C
C 90 BDU(NP)=XONE    000856
IFLAG=-1              000857
100 CALL VE02A (NP,MTOTAL,B,NP,GM,CN,LCN,CM,BDL,BDU,DELTA,MACT,MEQ,H,
1NPP,LT,MODE)        000858
000859
000860
000861
000862
C
C      CHECK WHETHER THE REQUIRED FEASIBILITY CONDITIONS HOLD 000863
C
C      IF (DELTA(NP).LE.VSMALL) GO TO 150 000864
DO 110 J=1,MACT      000865
IF (LT(J).GT.NPP) GO TO 110 000866
IF (LT(J).LT.NPP) GO TO 170 000867
IFLAG=1              000868
000869
110 CONTINUE          000870
IF (IFLAG.GE.1) GO TO 120 000871
IF (IFLAG.GE.0) GO TO 150 000872
BDU(NP)=FEASP*DELTA(NP) 000873
IFLAG=0              000874
GO TO 100             000875
C
C      CALCULATE THE LAGRANGE MULTIPLIERS 000876
C
C 120 DO 140 J=1,MACT 000877
K=LT(J)-NPP          000878
NPJ=NP+J              000879
IF (K.LE.0) GO TO 140 000880
000881
DO 130 I=1,N          000882
NSIXI=NSIX+I          000883
VLAM(K)=VLAM(K)+H(NPJ,I)*DELTA(NSIXI) 000884
000885
130 CONTINUE          000886
140 CONTINUE          000887
GO TO 190             000888
C
C      RETURN FROM THE SUBROUTINE 000889
C
C 150 INF=5            000890
IF (IPRINT.LT.0) GO TO 190 000891
IF (LP.GT.0) WRITE (LP,160) 000892
000893
160 FORMAT (/5X,45HTHE GIVEN CONSTRAINTS SEEM TO BE INCONSISTENT) 000894
GO TO 190             000895
000896
170 INF=6            000897
IF (IPRINT.LT.0) GO TO 190 000898
IF (LP.GT.0) WRITE (LP,180) VLARGE 000899
180 FORMAT (/5X,46HVE02A FINDS THAT AN ARTIFICIAL BOUND IS ACTIVE
1 /5X,46HTHE PREDICTED CHANGE IN THE VARIABLES EXCEEDS,1PE12.4) 000900
000901
190 RETURN            000902
END                  000903
000904
000905
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 000906
000907
000908
000909
000910
```

```
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
```

000911
000912
000913
000914
000915
000916
000917
000918
000919
000920
000921
000922
000923
000924
000925
000926
000927
000928
000929
000930
000931
000932
000933
000934
000935
000936
000937
000938
000939
000940
000941
000942
000943
000944
000945
000946
000947
000948
000949
000950
000951
000952
000953
000954
000955
000956
000957
000958
000959
000960
000961
000962
000963
000964
000965
000966
000967
000968
000969
000970
000971
000972
000973
000974
000975

```
DO 110 J=1,N          000976
H(N+I,J)=C(J,LI)    000977
H(J,N+I)=C(J,LI)    000978
110 CONTINUE          000979
X(NN+I)=D(LI)        000980
120 DO 130 J=1,K      000981
130 H(N+I,N+J)=XZERO 000982
140 NK=N+K            000983
C
C     INVERT MATRIX GIVING OPERATORS H AND CSTAR
C
CALL MB01C (H,NK,IH,LT(NNMD,X(N3+1)) 000984
GO TO 200             000985
C
C     SET UP RHS ONLY
C
150 DO 160 I=1,N      000986
X(N+I)=B(I)           000987
160 CONTINUE          000988
DO 190 I=1,K          000989
LI=LT(I)              000990
IF (LI.GT.NN) GO TO 180 000991
IF (LI.GT.N) GO TO 170 000992
X(NN+I)=BDL(LI)       000993
GO TO 190             000994
170 X(NN+I)=-BDU(LI-N) 000995
GO TO 190             000996
180 X(NN+I)=D(LI-NN)   000997
190 CONTINUE          000998
C
C     SOLVE FOR SOLUTION POINT X
C
NK=N+K                000999
200 DO 210 I=1,N      001000
X(I)=FM02AS(NK,H(1,I),1,X(N+1),1) 001001
210 CONTINUE          001002
C
C     CHECK FEASIBILITY, IF NOT EXIT TO 10
C
DO 250 I=1,M          001003
IF (LT(NN+I).LE.0) GO TO 250 001004
IF (I.GT.N) GO TO 220 001005
Z=X(I)-BDL(I)         001006
GO TO 240             001007
220 IF (I.GT.NN) GO TO 230 001008
Z=BDU(I-N)-X(I-N)    001009
GO TO 240             001010
230 J=I-NN             001011
Z=FM02AS(N,C(1,J),1,X(1),1)-D(J) 001012
240 IF (Z.LT.XZERO) GO TO 10 001013
250 CONTINUE          001014
260 CONTINUE          001015
C
C     CALCULATE GRADIENT G AND LAGRANGE MULTIPLIERS -CSTAR.G,
C     FIND LARGEST MULTIPLIER, EXIT IF NOT POSITIVE
C
DO 270 I=1,N          001016
X(N6+I)=FM02AS(N,A(I,1),IA,X(1),1)-B(I) 001017
270 CONTINUE          001018
IF (K.EQ.0) RETURN    001019
Z=-XONE75             001020
DO 280 I=1,K          001021
IF (LT(NN+LT(I)).EQ.-1) GO TO 280 001022
ZZ=-FM02AS(N,H(N+I,1),IH,X(N6+1),1) 001023
IF (ZZ.LE.Z) GO TO 280 001024
001025
001026
001027
001028
001029
001030
001031
001032
001033
001034
001035
001036
001037
001038
001039
001040
```

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 (43H SOLUTION 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
C IF (POSTIV.AND.ALPHA.GE.XONE) GO TO 680
C
C CALCULATE H.C AND CSTAR.C
C

001041
001042
001043
001044
001045
001046
001047
001048
001049
001050
001051
001052
001053
001054
001055
001056
001057
001058
001059
001060
001061
001062
001063
001064
001065
001066
001067
001068
001069
001070
001071
001072
001073
001074
001075
001076
001077
001078
001079
001080
001081
001082
001083
001084
001085
001086
001087
001088
001089
001090
001091
001092
001093
001094
001095
001096
001097
001098
001099
001100
001101
001102
001103
001104
001105

DO 430 I=1,N	001106
X(I)=X(I)+ALPHA*X(N5+I)	001107
430 CONTINUE	001108
ALPHA=ALPHA*Y	001109
J=1	001110
IF (K.EQ.N) J=N+1	001111
IF (IAL.GT.N) GO TO 450	001112
DO 440 I=J,NK	001113
X(N3+I)=H(I,IAL)	001114
440 CONTINUE	001115
CHC=X(N3+IAL)	001116
GO TO 500	001117
450 IB=IAL-N	001118
IF (IB.GT.N) GO TO 470	001119
DO 460 I=J,NK	001120
X(N3+I)=-H(I,IB)	001121
460 CONTINUE	001122
CHC=-X(N3+IB)	001123
GO TO 500	001124
470 IB=IB-N	001125
DO 480 I=1,N	001126
X(N5+I)=C(I,IB)	001127
480 CONTINUE	001128
DO 490 I=J,NK	001129
X(N3+I)=FM02AS(N,H(I,1),IH,X(N5+1),1)	001130
490 CONTINUE	001131
IF (K.NE.N) CHC=FM02AS(N,X(N5+1),1,X(N3+1),1)	001132
500 LT(NN+IAL)=0	001133
IF (K.EQ.N) GO TO 770	001134
IF (PASSIV) GO TO 600	001135
C	001136
C APPLY FORMULA FOR ADDING A CONSTRAINT	001137
C	001138
510 IF (K.EQ.0) GO TO 530	001139
DO 520 I=1,K	001140
ALPHA=X(N4+I)/CHC	001141
N1=N+I	001142
DO 520 J=1,N	001143
H(N1,J)=H(N1,J)-ALPHA*X(N3+J)	001144
530 K=K+1	001145
LT(K)=IAL	001146
DO 540 J=1,N	001147
H(N+K,J)=X(N3+J)/CHC	001148
540 CONTINUE	001149
IF (K.LT.N) GO TO 560	001150
DO 550 I=1,N	001151
DO 550 J=1,N	001152
550 H(I,J)=XZERO	001153
GO TO 580	001154
560 DO 570 I=1,N	001155
ALPHA=X(N3+I)/CHC	001156
DO 570 J=1,I	001157
H(I,J)=H(I,J)-ALPHA*X(N3+J)	001158
570 H(J,I)=H(I,J)	001159
580 IF (.NOT.PASSIV) GO TO 650	001160
C	001161
C REMOVAL OF A CONSTRAINT HAS BEEN DEFERRED, SETUP AS IF	001162
C THE CONSTRAINT IS BEING REMOVED FROM AUGMENTED BASIS	001163
C	001164
DO 590 I=1,N	001165
X(N6+I)=FM02AS(N,A(I,1),IA,X(1),1)-B(I)	001166
X(NN+I)=H(N+II,I)	001167
590 CONTINUE	001168
Z=-FM02AS(N,X(N6+1),1,X(NN+1),1)	001169
IF (Z.EQ.XZERO) GO TO 700	001170

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.CH*C(Z-ALPHA*CAC)+CHC*CC) GO TO 510
001171
001172
001173
001174
001175
001176
001177
001178
001179
001180
001181
001182
001183
001184
001185
001186
001187
001188
001189
001190
001191
001192
001193
001194
001195
001196
001197
001198
001199
001200
001201
001202
001203
001204
001205
001206
001207
001208
001209
001210
001211
001212
001213
001214
001215
001216
001217
001218
001219
001220
001221
001222
001223
001224
001225
001226
001227
001228
001229
001230
001231
001232
001233
001234
001235

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)
LTC(II)=IAL
650 IF (K.EQ.N) GO TO 260

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
LTC(II)=LTC(K)
740 K=K-1

```
DO 750 I=1,K          001236
NI=N+I                001237
X(N3+I)=FM02AS(N,H(NI,1),IH,X(N+1),1) 001238
750 CONTINUE           001239
DO 760 I=1,K          001240
ALPHA=X(N3+I)/CAC   001241
NI=N+I                001242
DO 760 J=1,N          001243
H(NI,J)=H(NI,J)-ALPHA*X(NN+J) 001244
GO TO 260             001245
770 Z=XONE/X(N4+II)  001246
C
C      APPLY SIMPLEX FORMULA TO EXCHANGE CONSTRAINTS 001247
C
DO 810 I=1,N          001248
NI=N+I                001249
IF (I.NE.II) GO TO 790 001250
DO 780 J=1,N          001251
H(NI,J)=H(NI,J)*Z   001252
780 CONTINUE           001253
GO TO 810             001254
790 ZZ=Z*X(N4+I)      001255
DO 800 J=1,N          001256
H(NI,J)=H(NI,J)-ZZ*X(NN+J) 001257
800 CONTINUE           001258
810 CONTINUE           001259
LT(I)=IAL             001260
GO TO 260             001261
C
820 K=0                001262
IF (KE.NE.0.AND.LPR.GT.0) WRITE (LPR,830) 001263
830 FORMAT (30H0KE MUST BE 0 IN MODES 2 AND 3) 001264
KE=0                  001265
DO 840 I=1,M          001266
LT(NN+I)=1             001267
840 CONTINUE           001268
CALL MB01C (H,N,IH,LT(NN),X(N6+1)) 001269
C
C      START WITH EMPTY BASIS FROM FEASIBLE POINT 001270
C      SEARCH DIRECTION IS -A(-1).B 001271
C
GO TO 650             001272
END                  001273
C
SUBROUTINE LA02A (N,M,C,IC,D,BDL,BDU,X,K,KE,H,IH,LT) 001274
COMMON /LA02B/ LP,IFLAG 001275
REAL C(IC,1),D(1),BDL(N),BDU(N),X(1),H(IH,1) 001276
INTEGER LT(1)          001277
COMMON /MB01D/ LPMB01,IFMB01 001278
REAL ZERO,ONE,RANGE/0.0,1.0,1.0E75/ 001279
DATA ZERO,ONE,RANGE/0.0,1.0,1.0E75/ 001280
DATA LP/0/              001281
C
C      SUPPRESS ERROR MESSAGES FROM MB01C/CD 001282
C
LPMB1=LPMB01            001283
LPMB01=0                001284
10 IFLAG=0               001285
NN=N+N                 001286
N3=NN+N                001287
DO 20 I=1,M              001288
N1=NN+I                001289
LT(N1)=1                001290
20 CONTINUE              001291
C
C      LPMB1=LPMB01 001292
C      LPMB01=0 001293
10 IFLAG=0               001294
NN=N+N                 001295
N3=NN+N                001296
DO 20 I=1,M              001297
N1=NN+I                001298
LT(N1)=1                001299
20 CONTINUE              001300
```

```

C CONSTRAINTS INDEXED AS
C -1=EQUALITY, 0=ACTIVE, 1=INACTIVE, 2=VIOLATED
C
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
C DO 60 I=1,N
C DO 30 J=1,N
C H(I,J)=ZERO
C
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) 001366
150 H(J,I)=H(I,J) 001367
CALL MB01C (H,K,IH,LT(N+1),X(N+1)) 001368
IF (IFMB01.EQ.0) GO TO 170 001369
160 IF (K.EQ.KE) GO TO 690 001370
K=KE 001371
GO TO 10 001372
001373
C CALCULATE GENERALIZED INVERSE OF V, VPLUS=M.VTRANSPOSE 001374
C 001375
170 DO 190 I=1,K 001376
DO 180 J=1,K 001377
N1=N+J 001378
X(N1)=H(I,J) 001379
180 CONTINUE 001380
DO 190 J=1,N 001381
190 H(I,J)=FM02AS(K,X(N+1),1,H(J,N+1),IH) 001382
001383
C SET UP DIAGONAL ELEMENTS OF THE PROJECTION MATRIX P=V.VPLUS 001384
C 001385
DO 200 I=1,N 001386
N1=N+I 001387
X(N1)=FM02AS(K,H(1,I),1,H(I,N+1),IH) 001388
200 CONTINUE 001389
DO 210 I=1,N 001390
N1=N+I 001391
LT(N1)=0 001392
210 CONTINUE 001393
KV=K 001394
001395
C ADD BOUND E(I) CORRESPONDING TO THE SMALLEST DIAG(P) 001396
C 001397
220 Z=ONE 001398
DO 230 I=1,N 001399
N1=N+I 001400
IF (LT(N1).EQ.1) GO TO 230 001401
IF (X(N1).GE.Z) GO TO 230 001402
Z=X(N1) 001403
II=I 001404
230 CONTINUE 001405
Y=ONE 001406
IF (X(II)-BDL(II).GT.BDU(II)-X(II)) Y=-ONE 001407
C CALCULATE VECTORS VPLUS.E(I) AND U=E(I)-V.VPLUS.E(I) 001408
C 001409
IF (Y.NE.ONE) GO TO 250 001410
DO 240 I=1,K 001411
N1=NN+I 001412
X(N1)=H(I,II) 001413
240 CONTINUE 001414
GO TO 270 001415
250 DO 260 I=1,K 001416
N1=NN+I 001417
X(N1)=-H(I,II) 001418
260 CONTINUE 001419
270 CONTINUE 001420
DO 280 I=1,N 001421
N1=N+I 001422
IF (LT(N1).EQ.1) GO TO 280 001423
N1=N3+I 001424
X(N1)=-FM02AS(KV,H(I,N+1),IH,X(NN+1),1) 001425
280 CONTINUE 001426
DO 290 I=1,N 001427
H(I,II)=ZERO 001428
290 CONTINUE 001429
001430

I1=N+II
LT(I1)=1
I3=N3+II
Z=ONE+X(I3)*Y
001431
001432
001433
001434
001435
001436
001437
001438
001439
001440
001441
001442
001443
001444
001445
001446
001447
001448
001449
001450
001451
001452
001453
001454
001455
001456
001457
001458
001459
001460
001461
001462
001463
001464
001465
001466
001467
001468
001469
001470
001471
001472
001473
001474
001475
001476
001477
001478
001479
001480
001481
001482
001483
001484
001485
001486
001487
001488
001489
001490
001491
001492
001493
001494
001495

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 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 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 CALCULATE THE CONSTRAINT RESIDUALS, THE NUMBER OF VIOLATED
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	001496
Z=X(I)-BDL(I)	001497
GO TO 420	001498
400 IF (I.GT.NN) GO TO 410	001499
L= I-N	001500
Z=BDU(L)-X(L)	001501
GO TO 420	001502
410 J= I-NN	001503
Z=-D(J)+FM02AS(N,C(1,J),1,X(1),1)	001504
420 X(N1)=Z	001505
IF (Z.GE.ZERO) GO TO 460	001506
KV=KV+1	001507
LT(N1)=2	001508
IF (I.GT.N) GO TO 430	001509
N2=N+I	001510
X(N2)=X(N2)+ONE	001511
GO TO 460	001512
430 IF (I.GT.NN) GO TO 440	001513
X(I)=X(I)-ONE	001514
GO TO 460	001515
440 DO 450 II=1,N	001516
N1I=N+II	001517
X(NII)=X(NII)+C(II,J)	001518
450 CONTINUE	001519
460 CONTINUE	001520
IF (KV.NE.0) GO TO 470	001521
GO TO 710	001522
C	001523
C POSSIBLE DIRECTIONS OF SEARCH OBTAINABLE BY REMOVING A	001524
C CONSTRAINT ARE ROWS OF H, CALCULATE THE OPTIMUM DIRECTION	001525
C	001526
470 Z=ZERO	001527
DO 480 I=1,N	001528
N1=NN+LT(I)	001529
IF (LT(N1).EQ.-1) GO TO 480	001530
Y=FM02AS(N,H(I,1),IH,X(N+1),1)	001531
IF (Y.LE.Z) GO TO 480	001532
Z=Y	001533
II=I	001534
480 CONTINUE	001535
IF (Z.LE.ZERO) GO TO 670	001536
C	001537
C SEARCH FOR THE NEAREST OF THE FURTHEST VIOLATED CONSTRAINT	001538
C AND THE NEAREST NONVIOLATED NONBASIC CONSTRAINT	001539
C	001540
ALPHA=RANGE	001541
BETA=ZERO	001542
DO 490 I=1,N	001543
N1=N+I	001544
X(N1)=H(II,I)	001545
490 CONTINUE	001546
DO 540 I=1,M	001547
N1=NN+I	001548
IF (LT(N1).LE.0) GO TO 540	001549
IF (I.GT.N) GO TO 500	001550
N2=N+I	001551
Z=-X(N2)	001552
GO TO 520	001553
500 IF (I.GT.NN) GO TO 510	001554
Z=X(I)	001555
GO TO 520	001556
510 JJ= I-NN	001557
Z=-FM02AS(N,X(N+1),1,C(1,JJ),1)	001558
520 IF (LT(N1).EQ.2) GO TO 530	001559
IF (Z.LE.ZERO) GO TO 540	001560

Z=X(N1)/Z
IF (Z.GE.ALPHA) GO TO 540
ALPHA=Z
IAL=I
GO TO 540
530 LTC(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 EXCHANGE WITH THE CONSTRAINT BEING REMOVED FROM THE BASIS,
C USING SIMPLEX FORMULA FOR NEW H
C
550 I1=NN+LT(I1)
LTC(I1)=1
I2=NN+IB
LT(I2)=0
LT(I1)=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+I
Z=ONE/X(I2)
DO 660 I=1,N
N1=N+I
X(I)=X(I)+BETA*X(N1)
IF (I.NE.I2) 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
001561
001562
001563
001564
001565
001566
001567
001568
001569
001570
001571
001572
001573
001574
001575
001576
001577
001578
001579
001580
001581
001582
001583
001584
001585
001586
001587
001588
001589
001590
001591
001592
001593
001594
001595
001596
001597
001598
001599
001600
001601
001602
001603
001604
001605
001606
001607
001608
001609
001610
001611
001612
001613
001614
001615
001616
001617
001618
001619
001620
001621
001622
001623
001624
001625

```
670 K=0          001626
    IFLAG=1        001627
    IF (LP.GT.0) WRITE (LP,680) 001628
680 FORMAT (50H ERROR RETURN FROM LA02A/AD ; THERE IS NO SOLUTION)
    GO TO 710      001629
690 IFLAG=2        001630
    IF (LP.GT.0) WRITE (LP,700) 001631
700 FORMAT (50H ERROR RETURN FROM LA02A/AD BECAUSE GIVEN EQUALITY,
    1 41H CONSTRAINTS ARE NOT LINEARLY INDEPENDENT) 001632
C
C       RESTORE UNIT NUMBER FOR MESSAGES FROM MB01C 001633
C
710 LPMB01=LPMB1 001634
    RETURN         001635
    END            001636
C
C       REAL FUNCTION FM02AS (N,A,IA,B,IB) 001637
C       DIMENSION A(N), B(N) 001638
C
C       N      IS THE LENGTH OF THE VECTORS (IF N <= 0 FM02AS/AD = 0) 001639
C       A      IS THE FIRST VECTOR 001640
C       IA     IS SUBSCRIPT DISPLACEMENT BETWEEN ELEMENTS OF A 001641
C       B      IS THE SECOND VECTOR 001642
C       IB     IS SUBSCRIPT DISPLACEMENT BETWEEN ELEMENTS OF B 001643
C       FM02AS/AD IS THE RESULT 001644
C
C       R1=0D0        001645
    IF (N.LE.0) GO TO 20 001646
    JA=1           001647
    IF (IA.LT.0) JA=1-(N-1)*IA 001648
    JB=1           001649
    IF (IB.LT.0) JB=1-(N-1)*IB 001650
    I=0             001651
10   I=I+1         001652
    R1=R1+A(JA)*B(JB) 001653
    JA=JA+IA        001654
    JB=JB+IB        001655
    IF (I.LT.N) GO TO 10 001656
20   FM02AS=R1      001657
    RETURN          001658
    END            001659
C
C       SUBROUTINE MB01C (A,M,IA,IND,C) 001660
C       REAL A,AMAX,C,DIV,STO,W,W1,FM02AS,ZERO,ONE 001661
C       DIMENSION A(IA,1), IND(1), C(1) 001662
C       COMMON /MB01D/ LP,IFLAG 001663
C       DATA ZERO,ONE/0.0,1.0/ 001664
C       DATA LP/0/ 001665
C       IFLAG=0        001666
10   IF (M-1) 310,10,20 001667
10   IF (A(1,1).EQ.ZERO) GO TO 330 001668
    A(1,1)=ONE/A(1,1) 001669
    GO TO 350        001670
20   M1=M-1         001671
    AMAX=ZERO        001672
    DO 40 I=1,M      001673
    IND(I)=I         001674
    IF (ABS(A(I,1))-ABS(AMAX)) 40,40,30 001675
30   AMAX=A(I,1)      001676
    IMAX=I           001677
40   CONTINUE        001678
    IF (AMAX.EQ.ZERO) GO TO 330 001679
    DO 120 J=1,M1    001680
C
```

50 IF (IMAX-J) 70,70,50	001691
50 IW=IND(IMAX)	001692
50 IND(IMAX)=IND(J)	001693
50 IND(J)=IW	001694
50 DO 60 K=1,M	001695
50 W=A(I MAX,K)	001696
50 A(I MAX,K)=A(J,K)	001697
50 A(J,K)=W	001698
60 CONTINUE	001699
70 J1=J+1	001700
70 IF (J.EQ.1) GO TO 90	001701
70 DO 80 I=J1,M	001702
70 A(J,I)=A(J,I)-FM02AS(J-1,A(J,1),IA,A(1,I),1)	001703
80 CONTINUE	001704
90 DIV=AMAX	001705
90 AMAX=ZERO	001706
90 DO 110 I=J1,M	001707
90 A(I,J)=A(I,J)/DIV	001708
90 A(I,J+1)=A(I,J+1)-FM02AS(J,A(I,1),IA,A(1,J+1),1)	001709
90 IF (ABS(A(I,J+1))-ABS(AMAX)) 110,110,100	001710
100 AMAX=A(I,J1)	001711
100 IMAX=I	001712
110 CONTINUE	001713
110 IF (AMAX.EQ.ZERO) GO TO 330	001714
120 CONTINUE	001715
120 DO 170 I1=1,M1	001716
120 I=M+1-I1	001717
120 I2=I-1	001718
120 DO 150 J1=1,I2	001719
120 J=I2+1-J1	001720
120 J2=J+1	001721
120 W1=-A(I,J)	001722
120 IF (I2-J2) 140,130,130	001723
130 W1=W1-FM02AS(I2-J2+1,A(J2,J),1,C(J2),1)	001724
140 C(J)=W1	001725
150 CONTINUE	001726
150 DO 160 K=1,I2	001727
150 A(I,K)=C(K)	001728
160 CONTINUE	001729
170 CONTINUE	001730
170 DO 260 I1=1,M	001731
170 I=M+1-I1	001732
170 I2=I+1	001733
170 W=A(I,I)	001734
170 DO 240 J=1,M	001735
170 IF (I-J) 180,190,200	001736
180 W1=ZERO	001737
180 GO TO 210	001738
190 W1=ONE	001739
190 GO TO 210	001740
200 W1=A(I,J)	001741
210 IF (I1-1) 230,230,220	001742
220 W1=W1-FM02AS(M-I2+1,A(I,I2),IA,A(I2,J),1)	001743
230 C(J)=W1	001744
240 CONTINUE	001745
240 DO 250 J=1,M	001746
240 A(I,J)=C(J)/W	001747
250 CONTINUE	001748
260 CONTINUE	001749
260 DO 300 I=1,M	001750
270 IF (IND(I)-I) 280,300,280	001751
280 J=IND(I)	001752
280 DO 290 K=1,M	001753
280 STO=A(K,I)	001754
280 A(K,I)=A(K,J)	001755

A(K,J)=STO	001756
290 CONTINUE	001757
ISTO= IND(J)	001758
IND(J)=J	001759
IND(I)= ISTO	001760
GO TO 270	001761
300 CONTINUE	001762
GO TO 350	001763
310 IF (LP.GT.0) WRITE (LP,320)	001764
320 FORMAT (53H ERROR RETURN FROM MB01C/CD BECAUSE M IS NOT POSITIVE)	001765
IFLAG=1	001766
GO TO 350	001767
330 IF (LP.GT.0) WRITE (LP,340)	001768
340 FORMAT (54H ERROR RETURN FROM MB01C/CD BECAUSE MATRIX IS SINGULAR)	001769
IFLAG=2	001770
350 RETURN	001771
END	001772

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.

