

**FMCGA - A FORTRAN PACKAGE FOR
UNCONSTRAINED MINIMIZATION
BY CONJUGATE GRADIENTS**

J.W. Bandler and M.A. El-Gamal

SOS-84-7-L

June 1984

© J.W. Bandler and M.A. El-Gamal 1984

No part of this document, computer program, source code, compiled code, related documentation and user manuals, magnetic tape, constituent subprograms, test programs, data and data files may be acquired, copied, reproduced, duplicated, executed, lent, disclosed, circulated, translated, transcribed or entered in any form into any machine without written permission. Address enquiries in this regard to Dr. J.W. Bandler. Neither the authors nor any other person, company, agency or institution make any warranty, express or implied, or assume any legal responsibility for the accuracy, completeness or usefulness of the material presented herein, or represent that its use would not infringe upon privately owned rights. This title page and original cover may not be separated from the contents of this document.

FMCGA - A FORTRAN PACKAGE FOR UNCONSTRAINED
MINIMIZATION BY CONJUGATE GRADIENTS

J.W. Bandler and M.A. El-Gamal

Abstract

This report contains a listing of the computer package FMCGA for minimization of an unconstrained function by the new three term conjugate gradient method. The package has been developed for the CDC 170/815 system with the NOS 2.1 level 558 operating system and the Fortran Extended (FTN) version 4.8 compiler. The listing contains a total of 532 lines (including 191 comments) constituting six subroutines.

This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant A7239.

The authors are with the Simulation Optimization Systems Research Laboratory and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7.

I. INTRODUCTION

This document contains a Fortran listing of all the subroutines of the FMCGA package. The FMCGA package executes unconstrained function minimization by the new three term conjugate gradient method [1]. The user's manual of the FMCGA package together with illustrative examples is found in [2].

The FMCGA package has been developed for the CDC 170/815 system with the NOS 2.1 level 558 operating system and the Fortran Extended (FTN) version 4.8 compiler. The package is available at McMaster University in the form of a library of binary relocatable subroutines. The library is in the group indirect file LIBFMCG accessible under the charge RJWBAND.

The FMCGA package contains 532 lines of which 191 are comments. It has been modularized into 6 subroutines. The list of all subroutines is given in Table I.

TABLE I
LIST OF SUBROUTINES OF THE FMCGA PACKAGE

Subroutine	Number of Lines (source text)	Listing (page)
1 ANORM	14	10
2 BNORM	15	10
3 FMCG1	100	4
4 FMCG2	257	6
5 LSERCH	129	11
6 VALUE	17	10

II. REFERENCES

- [1] L.C.W. Dixon, P.G. Ducksbury and P. Singh, "A new three term conjugate gradient method", Numerical Optimization Centre, Hatfield Polytechnic, England, Technical Report No. 130, 1983.
- [2] J.W. Bandler and M.A. El-Gamal, "FMCGA - A Fortran package for unconstrained minimization by conjugate gradients", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-84-7-U, 1984.

III. LISTING OF THE FMCGA PACKAGE

```

SUBROUTINE FMCG1 (FUNCT, GRAD, N, M, X, EPS, C1, ITERM, W, LW, ICH, IFLAG, IPR  A  1
1) A  2
C A  3
C A  4
C SUBROUTINE FMCG1 IS THE HIGHEST LEVEL SUBROUTINE TO MINIMIZE AN A  5
C UNCONSTRIANED FUNCTION USING THE METHOD OF CONJUGATE GRADIENTS. A  6
C IT SUBDIVIDES THE WORK SPACE SUPPLIED BY THE USER INTO A SET OF A  7
C VECTORS USED BY THE REMAINING SUBROUTINES AND INITIALIZES THE A  8
C MINIMIZATION PROCEDURE BY CALLING SUBROUTINE FMCG2. A  9
C A 10
C A 11
C DIMENSION X(N), W(LW) A 12
C EXTERNAL FUNCT, GRAD A 13
C A 14
C LIST OF MAIN VARIABLES A 15
C ----- A 16
C FUNCT THE NAME OF THE SUBROUTINE SUPPLIED BY THE USER A 17
C TO CALCULATE THE VALUES OF THE OBJECTIVE FUNCTION A 18
C A 19
C GRAD THE NAME OF THE SUBROUTINE SUPPLIED BY THE USER A 20
C TO CALCULATE THE GRADIENT OF THE OBJECTIVE FUNCTION A 21
C A 22
C N NUMBER OF OPTIMIZATION VARIABLES A 23
C A 24
C M THE DIMENSION OF THE VECTOR GD A 25
C A 26
C X VECTOR OF OPTIMIZATION VARIABLES A 27
C A 28
C EPS THE REQUIRED ACCURACY A 29
C A 30
C C1 DETERMINES THE ANGLE BETWEEN THE CURRENT SEARCH DIRECTION A 31
C AND THE STEEPEST DESCENT DIRECTION A 32
C A 33
C ITERM THE MAX. NUMBER OF ITERATIONS A 34
C A 35
C W THE WORKING SPACE A 36
C A 37
C LW THE LENGTH OF THE WORKING SPACE A 38
C A 39
C ICH UNIT NUMBER FOR PRINTED OUTPUT A 40
C A 41
C IFLAG CONTAINS THE INFORMATION ABOUT THE SOLUTION A 42
C = -1 INCORRECT INPUT ARGUMENTS A 43
C = 0 REQUIRED ACCURACY OBTAINED A 44
C = 1 LIMIT OF THE NUMBER OF ITERATIONS IS REACHED A 45
C A 46
C IPR CONTROLS THE PRINTED OUTPUT A 47
C = 0 THE PRINTED OUTPUT IS SUPRESSED A 48
C = 1 EACH ITERATION IS PRINTED A 49
C = 2 THE SOLUTION (LAST ITERATION) IS PRINTED ONLY A 50
C A 51
C JX1=1 A 51
C JXD=JX1+N A 52
C JZ0=JXD+N A 53
C JZ1=JZ0+N A 54
C JW0=JZ1+N A 55
C JW1=JW0+N A 56
C JY0=JW1+N A 57
C JY1=JY0+N A 58
C JT0=JY1+N A 59
C JT1=JT0+N A 60
C JT2=JT1+N A 61
C JPH=JT2+N A 62
C JGH=JPH+N A 63
C JD0=JGH+N A 64
C JC0=JD0+N A 65

```

	JG1=JG0+N	A	66
	JG=JG1+N	A	67
	JG0STR=JG+N	A	68
	JXSTR=JG0STR+N	A	69
	JGD=JXSTR+N	A	70
	IFLAG=0	A	71
	LMAX=JGD+M-1	A	72
	IF (LMAX.LE.LW) GO TO 10	A	73
	IFLAG=-1	A	74
	WRITE (ICH,30)	A	75
	RETURN	A	76
10	CONTINUE	A	77
C	PRINT THE INPUT DATA	A	78
	WRITE (ICH,20)	A	79
	WRITE (ICH,40)	A	80
	WRITE (ICH,50)	A	81
	WRITE (ICH,60) N	A	82
	WRITE (ICH,70) EPS	A	83
	WRITE (ICH,90) LW	A	84
	WRITE (ICH,80) IPR	A	85
	CALL FMCG2 (FUNCT,GRAD,N,M,X,EPS,C1,ITERM,ICH,IFLAG,IPR,W(JX1),W(J	A	86
	1XD),W(JZ0),W(JZ1),W(JW0),W(JW1),W(JY0),W(JY1),W(JT0),W(JT1),W(JT2)	A	87
	2,W(JPD),W(JGD),W(JD0),W(JG0),W(JG1),W(JC),W(JG0STR),W(JXSTR),W(JGD	A	88
	3))	A	89
20	FORMAT (1H1,/////,7X,"UNCONSTRAINED MINIMIZATION BY CONJUGATE GRAD	A	90
	1IENTS (FMCG PACKAGE) ",////)	A	91
30	FORMAT (2X,"INSUFFICIENT WORK SPACE")	A	92
40	FORMAT (7X,"INPUT DATA")	A	93
50	FORMAT (7X,10(1H-),/)	A	94
60	FORMAT (10X,"NUMBER OF VARIABLES (N)",34(1H.),2X,17,/)	A	95
70	FORMAT (10X,"ACCURACY (EPS)",42(1H.),1PE10.3,/)	A	96
80	FORMAT (10X,"PRINTOUT CONTROL (IPR)",42(1H.),12,/)	A	97
90	FORMAT (10X,"WORKING SPACE (LW)",41(1H.),17,/)	A	98
	RETURN	A	99
	END	A	100

C		B	1
C		B	2
	SUBROUTINE FMCG2 (FUNCT, GRAD, N, M, X, EPS, C1, ITERM, ICH, IFLAG, IPR, X1, X	B	3
	1D, Z0, Z1, W0, W1, Y0, Y1, T0, T1, T2, PH, GH, D0, G0, G1, G, C0STAR, XSTAR, GD)	B	4
		B	5
		B	6
	THIS IS THE MAIN SUBROUTINE OF THE FMCG PACKAGE. IT IMPLEMENTS	B	7
	THE DIXON ALGORITHM PUBLISHED IN TECHNICAL REPORT NO. 130 OF	B	8
	THE NUMERICAL OPTIMIZATION CENTRE, HATFIELD POLYTECHNIC, ENGLAND.	B	9
	A DETAILED DESCRIPTION OF THE DIFFERENT STEPS IN THE ALGORITHM ARE	B	10
	PROVIDED ALONG WITH THE LISTING.	B	11
	REFER TO [1] FOR THE THEORETICAL BACKGROUND AND THE DETAILED	B	12
	DISCUSSION OF THE ALGORITHM.	B	13
		B	14
		B	15
	DIMENSION X(N), X1(N), XD(N), Z0(N), Z1(N), W0(N), W1(N), Y0(N), Y	B	16
	11(N), T0(N), T1(N), T2(N), PH(N), GH(N), D0(N), G0(N), G1(N), G(N)	B	17
	2, C0STAR(N), XSTAR(N), CD(M)	B	18
	EXTERNAL FUNCT, GRAD	B	19
	COMMON NFE, NGE	B	20
		B	21
	LIST OF MAIN VARIABLES	B	22
	-----	B	23
	X VECTOR OF OPTIMIZATION VARIABLES	B	24
		B	25
	X1 VECTOR OF OPTIMIZATION VARIABLES AT ITERATION K+1	B	26
		B	27
	XD DUMMY VECTOR TO EVALUATE THE OBJECTIVE FUNCTION AT	B	28
	DIFFERENT POINTS IN SUBROUTINE LSEARCH	B	29
		B	30
	Z0 IS THE CORRECTION VECTOR Z AT ITERATION K	B	31
		B	32
	Z1 IS THE CORRECTION VECTOR Z AT ITERATION K-1	B	33
		B	34
	G0 GRADIENT OF THE OBJECTIVE FUNCTION AT ITERATION K	B	35
		B	36
	G1 GRADIENT OF THE OBJECTIVE FUNCTION AT ITERATION K-1	B	37
		B	38
	T0 SEARCH DIRECTION AT ITERATION K	B	39
		B	40
	T1 SEARCH DIRECTION AT ITERATION K-1	B	41
		B	42
	T2 SEARCH DIRECTION AT ITERATION K-2	B	43
		B	44
	W0 CORRECTION VECTOR W AT ITERATION K	B	45
		B	46
	W1 CORRECTION VECTOR W AT ITERATION K-1	B	47
		B	48
	INITIALIZATION OF VARIABLES.	B	49
		B	50
	CALL SECOND (TM1)	B	51
	J=0	B	52
	NFE=0	B	53
	NCE=0	B	54
	EPS1=EPS	B	55
	IF (EPS.GE.1.E-8) EPS1=1.E-10	B	56
	DO 10 I=1,N	B	57
	G0STAR(I)=0.	B	58
10	CONTINUE	B	59
	ITER=0	B	60
	DO 20 I=1,N	B	61
	Z1(I)=0.	B	62
	W1(I)=0.	B	63
20	CONTINUE	B	64
		B	65

C	STEP 1	B	66
C	EVALUATE THE OBJECTIVE FUNCTION AND ITS GRADIENT AT	B	67
C	THE STARTING POINT.	B	68
C		B	69
	CALL FUNCT (N,M,X,GD,F)	B	70
	CALL GRAD (N,M,X,GD,G1)	B	71
	NFE=NFE+1	B	72
	NGE=NGE+1	B	73
C		B	74
C	PRINT THE STARTING POINT,THE OBJECTIVE FUNCTION AND ITS GRADIENT.	B	75
C		B	76
	IF (IPR.EQ.3) GO TO 40	B	77
	WRITE (ICH,290) F	B	78
	WRITE (ICH,300)	B	79
	DO 30 I=1,N	B	80
	WRITE (ICH,310) I,X(I),G1(I)	B	81
	30 CONTINUE	B	82
	40 CONTINUE	B	83
C		B	84
C	STEP 2	B	85
C	START WITH THE STEEPEST DESCENT DIRECTION.	B	86
C		B	87
	DO 50 I=1,N	B	88
	G0(I)=G1(I)	B	89
	GH(I)=G1(I)	B	90
	PH(I)=-G1(I)	B	91
	T0(I)=-G1(I)	B	92
	T1(I)=-G1(I)	B	93
	50 CONTINUE	B	94
C		B	95
C	STEP 3	B	96
C	PERFORM AN APPROXIMATE LINE SEARCH.	B	97
C		B	98
	60 CALL LSERCH (FUNCT,GRAD,D0,X,N,M,T0,XD,G0,GD,F)	B	99
C		B	100
C	STEP 4	B	101
C	DETERMINE THE NEW POINT AND THE GRADIENT AT THIS POINT.	B	102
C	INCREASE THE NUMBER OF ITERATIONS BY 1.	B	103
C		B	104
	DO 70 I=1,N	B	105
	X1(I)=X(I)+D0(I)	B	106
	G1(I)=G0(I)	B	107
	70 CONTINUE	B	108
	IF (ANORM(D0,N).LT.EPS1) GO TO 220	B	109
	ITER=ITER+1	B	110
	J=J+1	B	111
	CALL GRAD (N,M,X1,GD,G0)	B	112
	NGE=NGE+1	B	113
C		B	114
C	STEP 5	B	115
C	DETERMINATION OF THE NEW CONJUGATE SEARCH DIRECTION	B	116
C	USING THE NAZARETH THREE TERM FORMULA.	B	117
C	REFER TO [1] FOR THE THEORETICAL BACKGROUND.	B	118
C		B	119
	DO 80 I=1,N	B	120
	Y0(I)=G0(I)-G1(I)	B	121
	80 CONTINUE	B	122
	GAMA=VALUE(Y0,Y0,Y0,T0,N)	B	123
	THETA=-VALUE(G0,T0,Y0,T0,N)	B	124
	DO 90 I=1,N	B	125
	Z0(I)=Z1(I)-THETA*D0(I)	B	126
	W0(I)=W1(I)-THETA*Y0(I)	B	127
	G0STAR(I)=G0(I)-W0(I)	B	128
	XSTAR(I)=X1(I)-Z0(I)	B	129
	T2(I)=T1(I)	B	130

	T1(I)=T0(I)	B 131
90	CONTINUE	B 132
	BETA=0.	B 133
	IF (J.GE.2) BETA=VALUE(Y0,Y1,Y1,T2,N)	B 134
	DO 100 I=1,N	B 135
	T0(I)=-Y0(I)+BETA*T2(I)+GAMA*T1(I)	B 136
100	CONTINUE	B 137
	IF (IPR.EQ.0) GO TO 120	B 138
	IF (IPR.GT.1) GO TO 120	B 139
	WRITE (ICH,320) ITER	B 140
	WRITE (ICH,300)	B 141
	DO 110 I=1,N	B 142
	WRITE (ICH,310) I,XSTAR(I),G0STAR(I)	B 143
110	CONTINUE	B 144
120	CONTINUE	B 145
C		B 146
C	STEP 6	B 147
C	CHECK THE STOPPING CRITERIA.	B 148
C	IF G0 < EPS, STOP.	B 149
C	IF THE NUMBER OF ITERATIONS EXCEEDS THE LIMIT DEFINED BY	B 150
C	USER, STOP.	B 151
C		B 152
	IF (ANORM(G0,N).LT.EPS.OR.ITER.GT.ITERMD GO TO 220	B 153
C		B 154
C	STEP 7	B 155
C	RESTART TESTS.	B 156
C	CHECK WHETHER THE NEW DIRECTION IS SUFFICIENTLY DOWNHILL.	B 157
C	IF ANY ONE OF THE RESTART TESTS IS SATISFIED THEN GO TO STEP 10.	B 158
C		B 159
	IF (C1*ANORM(T0,N)*ANORM(G0,N).GE.-BNORM(T0,G0,N)) GO TO 160	B 160
	IF (ANORM(G0STAR,N).LE.C1*ANORM(G0,N).OR.ANORM(T0,N).LE.C1*ANORM(G	B 161
	0,N)) GO TO 160	B 162
	IF (J.CT.1) GO TO 130	B 163
	GO TO 140	B 164
130	IF (BNORM(PH,Y0,N)**2..GT..2*(ANORM(PH,N)**2.)*(ANORM(Y0,N)**2.))	B 165
	160 TO 160	B 166
C		B 167
C	STEP 8	B 168
C	IF THE NUMBER OF ITERATIONS \geq N+1 THEN GO TO STEP 10	B 169
C		B 170
140	IF (J.GE.N+1) GO TO 160	B 171
C		B 172
C	STEP 9	B 173
C	RESTART TESTS ARE NOT SATISFIED	B 174
C	UPDATE THE VARIABLES AND START LINE SEARCH IN THE NEW DIRECTION.	B 175
C		B 176
	DO 150 I=1,N	B 177
	Y1(I)=Y0(I)	B 178
	Z1(I)=Z0(I)	B 179
	W1(I)=W0(I)	B 180
	X(I)=X1(I)	B 181
150	CONTINUE	B 182
	GO TO 60	B 183
C		B 184
C	STEP 10	B 185
C	TEST Z0 FOR WOLFE CONDITION. IF SATISFIED GO TO STEP 12.	B 186
C		B 187
160	IF (C1*ANORM(Z0,N)*ANORM(G0,N).LT.-BNORM(Z0,G0,N)) GO TO 180	B 188
C		B 189
C	STEP 11	B 190
C	RESTART WITH STEEPEST DESCENT DIRECTION.	B 191
C		B 192
	DO 170 I=1,N	B 193
	T0(I)=-G0(I)	B 194
	CH(I)=-C0(I)	B 195

	PH(I)=-G0(I)	B 196
170	CONTINUE	B 197
	J=0	B 198
	GO TO 200	B 199
C		B 200
C	STEP 12	B 201
C	RESTART WITH Z0 AS THE NEW SEARCH DIRECTION	B 202
C		B 203
180	DO 190 I=1,N	B 204
	T0(I)=Z0(I)	B 205
	T1(I)=Z0(I)	B 206
	GH(I)=G0(I)	B 207
	PH(I)=Z0(I)	B 208
190	CONTINUE	B 209
C		B 210
C	STEP 13	B 211
C	INITIALIZE SOME OF THE VARIABLES BEFORE RESTARTING	B 212
C	GO TO STEP 3	B 213
C		B 214
	J=0	B 215
200	DO 210 I=1,N	B 216
	W1(I)=0.	B 217
	Z1(I)=0.	B 218
	X(I)=X1(I)	B 219
210	CONTINUE	B 220
	GO TO 60	B 221
220	CONTINUE	B 222
	IF (IPR.EQ.0) GO TO 360	B 223
C	PRINT OUT THE SOLUTION FOUND BY THE PACKAGE	B 224
	WRITE (ICH,330)	B 225
	WRITE (ICH,340)	B 226
	CALL FUNCT (N,M,XSTAR,GD,F)	B 227
	NFE=NFE+1	B 228
	NEFE=NFE+N*NCE	B 229
	CALL SECOND (TM2)	B 230
	CPU=TM2-TM1	B 231
	WRITE (ICH,350) F	B 232
	WRITE (ICH,300)	B 233
	DO 230 I=1,N	B 234
	WRITE (ICH,310) I,XSTAR(I),COSTAR(I)	B 235
230	CONTINUE	B 236
	IF (ITER.EQ.ITERMD) IFLAG=1	B 237
	WRITE (ICH,240) IFLAG	B 238
	WRITE (ICH,250) ITER	B 239
	WRITE (ICH,260) NEFE	B 240
	WRITE (ICH,270) CPU	B 241
	WRITE (ICH,280)	B 242
240	FORMAT (/,10X,"TYPE OF SOLUTION (IFLAG)",38(1H.),2X,12,/) B 243	
250	FORMAT (10X,"NUMBER OF ITERATIONS ",38(1H.),17,/) B 244	
260	FORMAT (10X,"EFFECTIVE FUNCTION EVALUATIONS (EFE) ",20(1H.),19,/) B 245	
270	FORMAT (10X,"EXECUTION TIME (IN SECONDS) ",30(1H.),F9.3) B 246	
280	FORMAT (1H1) B 247	
290	FORMAT (/,10X,"STARTING POINT.",9X,"OBJECTIVE FUNCTION :",2X,1PE12 B 248	
	1.5,/) B 249	
300	FORMAT (/,26X,"VARIABLES",28X,"GRADIENT",/) B 250	
310	FORMAT (13X,16,4X,E12.5,25X,E12.5) B 251	
320	FORMAT (///,10X,"SOLUTION AT ITERATION NUMBER ",1X,14) B 252	
330	FORMAT (////,7X,"SOLUTION") B 253	
340	FORMAT (7X,9(1H-),/) B 254	
350	FORMAT (34X,"OBJECTIVE FUNCTION :",2X,1PE12.5,/) B 255	
360	RETURN B 256	
	END B 257	

C		C	1
C		C	2
	FUNCTION VALUE (A,B,C,D,N)	C	3
C		C	4
C	THIS FUNCTION CALCULATES THE VALUE OF THE QUOTIENT OF THE PRODUCT	C	5
C	OF THE TWO VECTORS OF A AND B DIVIDED BY THE PRODUCT OF C AND D.	C	6
		C	7
	REAL A(N),B(N),C(N),D(N)	C	8
	P=0.	C	9
	R=0.	C	10
	DO 10 I=1,N	C	11
	P=P+A(I)*B(I)	C	12
	R=R+C(I)*D(I)	C	13
10	CONTINUE	C	14
	VALUE=P/R	C	15
	RETURN	C	16
	END	C	17
C		D	1
C		D	2
	FUNCTION ANORM (A,N)	D	3
C		D	4
C	THIS FUNCTION EVALUATES THE NORM OF A VECTOR A.	D	5
C		D	6
	REAL A(N)	D	7
	P=0.	D	8
	DO 10 I=1,N	D	9
	P=P+A(I)*A(I)	D	10
10	CONTINUE	D	11
	ANORM=SQRT(P)	D	12
	RETURN	D	13
	END	D	14
C		E	1
C		E	2
	FUNCTION BNORM (A,B,N)	E	3
C		E	4
C	THIS FUNCTION EVALUATES THE VALUE OF THE PRODUCT OF THE TWO	E	5
C	VECTORS A AND B.	E	6
		E	7
	REAL A(N),B(N)	E	8
	P=0.	E	9
	DO 10 I=1,N	E	10
	P=P+A(I)*B(I)	E	11
10	CONTINUE	E	12
	BNORM=P	E	13
	RETURN	E	14
	END	E	15

```

C
C
SUBROUTINE LSERCH ( FUNCT, GRAD, D0, X, N, M, T0, XD, G, GD, F)
C
C
SUBROUTINE LINESERCH IMPLEMENTS THE LINE SEARCH AT STEP 3
C
C
IN THE SUBROUTINE FMGG2. IT FIRST PERFORMES A PARABOLIC
C
C
INTERPOLATION USING THE VALUES OF THE FUNCTION AND ITS
C
C
DERIVATIVE AT THE STARTING POINT AND THE FUNCTION VALUE
C
C
AT AN OFFSET POINT. IF THE PREDICTED FUNCTION VALUE
C
C
DOES NOT SATISFY THE WOLFE CONDITIONS THEN THE ARMIJIO PROC-
C
C
EDURE IS ADOPTED BASED ON THAT STEP.
C
C
DIMENSION D0(N), X(N), T0(N), XD(N), G(N), GD(M)
EXTERNAL FUNCT, GRAD
COMMON NFE, NGE
LK=0
LL=0
CONS=10.
ALMIN=0.
FMIN=F
G01=BNORM(G, T0, N)
EST=0.
AL1=-2.*(F-EST)/G01
AL=AMIN1(1., AL1)
DO 10 I=1, N
XD(I)=X(I)+AL*T0(I)
10 CONTINUE
CALL FUNCT (N, M, XD, GD, F1)
NFE=NFE+1
B=-G01
ALS=AL*AL
C=2.*(F1-F-AL*G01)/ALS
IF (C.LE.1E-10) A1=AL
IF (C.LE.1E-10) GO TO 50
C
C
START THE QUADRATIC INTERPOLATION
C
C
A1=B/C
DO 20 I=1, N
XD(I)=X(I)+A1*T0(I)
20 CONTINUE
CALL FUNCT (N, M, XD, GD, F2)
NFE=NFE+1
P2=F+A1*G01+A1*A1*C/2.
E0=ABS(F-P2)/CONS
E1=ABS(F2-P2)
E2=ABS(F+A1*G01-F2)
E3=ABS(A1*G01)/CONS
IF (E1.LT.E0.AND.E3.LT.E2) GO TO 30
IF (F2.LT.F1) F1=F2
IF (F2.GT.F1) A1=AL
GO TO 50
30 CONTINUE
AL=A1
DO 40 I=1, N
D0(I)=AL*T0(I)
40 CONTINUE
F=F2
RETURN
C
C
THE QUADRATIC INTERPOLATION IS NOT SATISFACTORY
C
C
ARMIJIO PROCEDURE IS ADOPTED
C
50 CONTINUE
FMIN=F1

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```

IF (F1.GT.F) ALMIN=0.	F 66
IF (F1.GT.F) FMIN=F	F 67
IF (F1.LT.F) ALMIN=A1	F 68
E1=F-F1	F 69
E2=A1*ABS(G01)/CONS	F 70
IF (E1.LE.E2) GO TO 100	F 71
60 CONTINUE	F 72
AL=2.*A1	F 73
DO 70 I=1,N	F 74
XD(I)=X(I)+AL*T0(I)	F 75
70 CONTINUE	F 76
CALL FUNCT (N,M,XD,GD,F3)	F 77
NFE=NFE+1	F 78
LK=LK+1	F 79
E3=F-F3	F 80
E4=AL*ABS(G01)/CONS	F 81
IF (F3.LT.FMIN) ALMIN=AL	F 82
IF (F3.LT.FMIN) FMIN=F3	F 83
IF (E3.LT.E4.OR.LK.GE.10) GO TO 80	F 84
A1=AL	F 85
F1=F3	F 86
GO TO 60	F 87
80 CONTINUE	F 88
AL=A1	F 89
F=F1	F 90
IF (F1.GT.FMIN) F=FMIN	F 91
IF (F1.GT.FMIN) AL=ALMIN	F 92
DO 90 I=1,N	F 93
D0(I)=AL*T0(I)	F 94
90 CONTINUE	F 95
RETURN	F 96
100 CONTINUE	F 97
AL=A1/2.	F 98
DO 110 I=1,N	F 99
XD(I)=X(I)+AL*T0(I)	F 100
110 CONTINUE	F 101
CALL FUNCT (N,M,XD,GD,F4)	F 102
NFE=NFE+1	F 103
LL=LL+1	F 104
E5=F-F4	F 105
E6=AL*ABS(G01)/CONS	F 106
IF (FMIN.GT.F4) ALMIN=AL	F 107
IF (FMIN.GT.F4) FMIN=F4	F 108
IF (E5.GT.E6) GO TO 120	F 109
IF (LL.GE.10) GO TO 140	F 110
A1=AL	F 111
F1=F4	F 112
GO TO 100	F 113
120 CONTINUE	F 114
F=F1	F 115
AL=A1	F 116
IF (F1.GT.FMIN) AL=ALMIN	F 117
IF (F1.GT.FMIN) F=FMIN	F 118
DO 130 I=1,N	F 119
D0(I)=AL*T0(I)	F 120
130 CONTINUE	F 121
RETURN	F 122
140 CONTINUE	F 123
F=F4	F 124
DO 150 I=1,N	F 125
D0(I)=AL*T0(I)	F 126
150 CONTINUE	F 127
RETURN	F 128
END	F 129