FMCGA - A FORTRAN PACKAGE FOR UNCONSTRAINED MINIMIZATION BY CONJUGATE GRADIENTS

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

SOS-84-7-L

June 1984

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
     1)
C
C
C     SUBROUTINE FMCG1 IS THE HIGHEST LEVEL SUBROUTINE TO MINIMIZE AN
C     UNCONSTRIANED FUNCTION USING THE METHOD OF CONJUGATE GRADIENTS.
C     IT SUBDIVIDES THE WORK SPACE SUPPLIED BY THE USER INTO A SET OF
C     VECTORS USED BY THE REMAINING SUBROUTINES AND INITIALIZES THE
C     MINIMIZATION PROCEDURE BY CALLING SUBROUTINE FMCG2.
C
C
      DIMENSION X(N), W(LW)
      EXTERNAL FUNCT,GRAD
C
C     LIST OF MAIN VARIABLES
C     ----------------------
C     FUNCT  THE NAME OF THE SUBROUTINE  SUPPLIED BY THE USER
C            TO CALCULATE THE VALUES OF THE OBJECTIVE FUNCTION
C
C     GRAD   THE NAME OF THE SUBROUTINE SUPPLIED BY THE USER
C            TO CALCULATE THE GRADIENT OF THE OBJECTIVE FUNCTION
C
C     N      NUMBER OF OPTIMIZATION VARIABLES
C
C     M      THE DIMENSION OF THE VECTOR GD
C
C     X      VECTOR OF OPTIMIZATION VARIABLES
C
C     EPS    THE REQUIRED ACCURACY
C
C     C1     DETERMINES THE ANGLE BETWEEN THE CURRENT SEARCH DIRECTION
C            AND THE STEEPEST DESCENT DIRECTION
C
C     ITERM  THE MAX. NUMBER OF ITERATIONS
C
C     W      THE WORKING SPACE
C
C     LW     THE LENGTH OF THE WORKING SPACE
C
C     ICH    UNIT NUMBER FOR PRINTED OUTPUT
C
C     IFLAG  CONTAINS THE INFORMATION ABOUT THE SOLUTION
C            = -1 INCORRECT INPUT ARGUMENTS
C            = 0 REQUIRED ACCURACY OBTAINED
C            = 1 LIMIT OF THE NUMBER OF ITERATIONS IS REACHED
C
C     IPR    CONTROLS THE PRINTED OUTPUT
C            = 0 THE PRINTED OUTPUT IS SUPRESSED
C            = 1 EACH ITERATION IS PRINTED
C            = 2 THE SOLUTION (LAST ITERATION) IS PRINTED ONLY
      JX1=1
      JXD=JX1+N
      JZ0=JXD+N
      JZ1=JZ0+N
      JW0=JZ1+N
      JW1=JW0+N
      JY0=JW1+N
      JY1=JY0+N
      JT0=JY1+N
      JT1=JT0+N
      JT2=JT1+N
      JPH=JT2+N
      JGH=JPH+N
      JD0=JGH+N
      JC0=JD0+N
```

```
      JG1=JG0+N
      JG=JG1+N
      JG0STR=JG+N
      JXSTR=JG0STR+N
      JGD=JXSTR+N
      IFLAG=0
      LMAX=JGD+M-1
      IF (LMAX.LE.LW) GO TO 10
      IFLAG=-1
      WRITE (ICH,30)
      RETURN
   10 CONTINUE
C     PRINT THE INPUT DATA
      WRITE (ICH,20)
      WRITE (ICH,40)
      WRITE (ICH,50)
      WRITE (ICH,60) N
      WRITE (ICH,70) EPS
      WRITE (ICH,90) LW
      WRITE (ICH,80) IPR
      CALL FMCG2 (FUNCT,GRAD,N,M,X,EPS,C1,ITERM,ICH,IFLAG,IPR,W(JX1),W(J
     1XD),W(JZ0),W(JZ1),W(JW0),W(JW1),W(JY0),W(JY1),W(JT0),W(JT1),W(JT2)
     2,W(JPH),W(JGH),W(JD0),W(JG0),W(JG1),W(JG),W(JG0STR),W(JXSTR),W(JGD
     3))
   20 FORMAT (1H1,/////,7X,"UNCONSTRAINED MINIMIZATION BY CONJUGATE GRAD
     1IENTS (FMCG PACKAGE) ",///)
   30 FORMAT (2X,"INSUFFICIENT WORK SPACE")
   40 FORMAT (7X,"INPUT DATA")
   50 FORMAT (7X,10(1H-),/)
   60 FORMAT (10X,"NUMBER OF VARIABLES (N)",34(1H.),2X,I7,/)
   70 FORMAT (10X,"ACCURACY (EPS)",42(1H.),1PE10.3,/)
   80 FORMAT (10X,"PRINTOUT CONTROL (IPR)",42(1H.),I2,/)
   90 FORMAT (10X,"WORKING SPACE (LW)",41(1H.),I7,/)
      RETURN
      END
```

```
C
C
      SUBROUTINE FMCG2 (FUNCT,GRAD,N,M,X,EPS,C1,ITERM,ICH,IFLAG,IPR,X1,X
     1D,Z0,Z1,W0,W1,Y0,Y1,T0,T1,T2,PH,GH,D0,G0,G1,G,G0STAR,XSTAR,GD)
C
C
C     THIS IS THE MAIN SUBROUTINE OF THE FMCG PACKAGE. IT IMPLEMENTS
C     THE DIXON ALGORITHM PUBLISHED IN TECHNICAL REPORT NO. 130 OF
C     THE NUMERICAL OPTIMIZATION CENTRE, HATFIELD POLYTECHNIC,ENGLAND.
C     A DETAILED DESCRIPTION OF THE DIFFERENT STEPS IN THE ALGORITHM ARE
C     PROVIDED ALONG WITH THE LISTING.
C     REFER TO [1] FOR THE THEORTICAL BACKGROUND AND THE DETAILED
C     DISCUSSION OF THE ALGORITHM.
C
C
      DIMENSION X(N), X1(N), XD(N), Z0(N), Z1(N), W0(N), W1(N), Y0(N), Y
     11(N), T0(N), T1(N), T2(N), PH(N), GH(N), D0(N), G0(N), G1(N), G(N)
     2, G0STAR(N), XSTAR(N), GD(M)
      EXTERNAL FUNCT,GRAD
      COMMON NFE,NGE
C
C     LIST OF MAIN VARIABLES
C     ----------------------
C     X          VECTOR OF OPTIMIZATION VARIABLES
C
C     X1         VECTOR OF OPTIMIZATION VARIABLES AT ITERATION K+1
C
C     XD         DUMMY VECTOR TO EVALUATE THE OBJECTIVE FUNCTION AT
C                DIFFERENT POINTS IN SUBROUTINE LSERCH
C
C     Z0         IS THE CORRECTION VECTOR Z AT ITERATION K
C
C     Z1         IS THE CORRECTION VECTOR Z AT ITERATION K-1
C
C     G0         GRADIENT OF THE OBJECTIVE FUNCTION AT ITERATION K
C
C     G1         GRADIENT OF THE OBJECTIVE FUNCTION AT ITERATION K-1
C
C     T0         SEARCH DIRECTION AT ITERATION K
C
C     T1         SEARCH DIRECTION AT ITERATION K-1
C
C     T2         SEARCH DIRECTION AT ITERATION K-2
C
C     W0         CORRECTION VECTOR W AT ITERATION K
C
C     W1         CORRECTION VECTOR W AT ITERATION K-1
C
C     INITIALIZATION OF VARIABLES.
C
      CALL SECOND (TM1)
      J=0
      NFE=0
      NGE=0
      EPS1=EPS
      IF (EPS.GE.1.E-8) EPS1=1.E-10
      DO 10 I=1,N
      G0STAR(I)=0.
   10 CONTINUE
      ITER=0
      DO 20 I=1,N
      Z1(I)=0.
      W1(I)=0.
   20 CONTINUE
C
```

```
C       STEP 1
C       EVALUATE THE OBJECTIVE FUNCTION AND ITS GRADIENT AT
C       THE STARTING POINT.
C
        CALL FUNCT (N,M,X,GD,F)
        CALL GRAD (N,M,X,GD,G1)
        NFE=NFE+1
        NGE=NGE+1
C
C       PRINT THE STARTING POINT,THE OBJECTIVE FUNCTION AND ITS GRADIENT.
C
        IF (IPR.EQ.3) GO TO 40
        WRITE (ICH,290) F
        WRITE (ICH,300)
        DO 30 I=1,N
        WRITE (ICH,310) I,X(I),G1(I)
   30   CONTINUE
   40   CONTINUE
C
C       STEP 2
C       START WITH THE STEEPEST DESCENT DIRECTION.
C
        DO 50 I=1,N
        G0(I)=G1(I)
        GH(I)=G1(I)
        PH(I)=-G1(I)
        T0(I)=-G1(I)
        T1(I)=-G1(I)
   50   CONTINUE
C
C       STEP 3
C       PERFORM AN APPROXIMATE LINE SEARCH.
C
   60   CALL LSERCH (FUNCT,GRAD,D0,X,N,M,T0,XD,G0,GD,F)
C
C       STEP 4
C       DETERMINE THE NEW POINT AND THE GRADIENT AT THIS POINT
C       INCREASE THE NUMBER OF ITERATIONS BY 1.
C
        DO 70 I=1,N
        X1(I)=X(I)+D0(I)
        G1(I)=G0(I)
   70   CONTINUE
        IF (ANORM(D0,N).LT.EPS1) GO TO 220
        ITER=ITER+1
        J=J+1
        CALL GRAD (N,M,X1,GD,G0)
        NGE=NGE+1
C
C       STEP 5
C       DETERMINATION OF THE NEW CONJUGATE SEARCH DIRECTION
C       USING THE NAZARETH THREE TERM FORMULA.
C       REFER TO [1] FOR THE THEORETICAL BACKGROUND.
C
        DO 80 I=1,N
        Y0(I)=G0(I)-G1(I)
   80   CONTINUE
        GAMA=VALUE(Y0,Y0,Y0,T0,N)
        THETA=-VALUE(G0,T0,Y0,T0,N)
        DO 90 I=1,N
        Z0(I)=Z1(I)-THETA*D0(I)
        W0(I)=W1(I)-THETA*Y0(I)
        G0STAR(I)=G0(I)-W0(I)
        XSTAR(I)=X1(I)-Z0(I)
        T2(I)=T1(I)
```

```
      T1(I)=T0(I)
   90 CONTINUE
      BETA=0.
      IF (J.GE.2) BETA=VALUE(Y0,Y1,Y1,T2,N)
      DO 100 I=1,N
      T0(I)=-Y0(I)+BETA*T2(I)+GAMA*T1(I)
  100 CONTINUE
      IF (IPR.EQ.0) GO TO 120
      IF (IPR.GT.1) GO TO 120
      WRITE (ICH,320) ITER
      WRITE (ICH,300)
      DO 110 I=1,N
      WRITE (ICH,310) I,XSTAR(I),G0STAR(I)
  110 CONTINUE
  120 CONTINUE
C
C     STEP 6
C     CHECK THE STOPPING CRITERIA.
C     IF ||G0|| < EPS, STOP.
C     IF THE NUMBER OF ITERATIONS EXCEEDS THE LIMIT DEFINED BY
C     USER, STOP.
C
      IF (ANORM(G0,N).LT.EPS.OR.ITER.GT.ITERM) GO TO 220
C
C     STEP 7
C     RESTART TESTS.
C     CHECK WHETHER THE NEW DIRECTION IS SUFFICIENTLY DOWNHILL.
C     IF ANY ONE OF THE RESTART TESTS IS SATISFIED THEN GO TO STEP 10.
C
      IF (C1*ANORM(T0,N)*ANORM(G0,N).GE.-BNORM(T0,G0,N)) GO TO 160
      IF (ANORM(G0STAR,N).LE.C1*ANORM(G0,N).OR.ANORM(T0,N).LE.C1*ANORM(G
     10,N)) GO TO 160
      IF (J.GT.1) GO TO 130
      GO TO 140
  130 IF (BNORM(PH,Y0,N)**2..GT..2*(ANORM(PH,N)**2.)*(ANORM(Y0,N)**2.))
     1GO TO 160
C
C     STEP 8
C     IF THE NUMBER OF ITERATIONS  ≥ N+1 THEN GO TO STEP 10
C
  140 IF (J.GE.N+1) GO TO 160
C
C     STEP 9
C     RESTART TESTS ARE NOT SATISFIED
C     UPDATE THE VARIABLES AND START LINE SEARCH IN THE NEW DIRECTION.
C
      DO 150 I=1,N
      Y1(I)=Y0(I)
      Z1(I)=Z0(I)
      W1(I)=W0(I)
      X(I)=X1(I)
  150 CONTINUE
      GO TO 60
C
C     STEP 10
C     TEST Z0 FOR WOLFE CONDITION. IF SATISFIED GO TO STEP 12.
C
  160 IF (C1*ANORM(Z0,N)*ANORM(G0,N).LT.-BNORM(Z0,G0,N)) GO TO 180
C
C     STEP 11
C     RESTART WITH STEEPEST DESCENT DIRECTION.
C
      DO 170 I=1,N
      T0(I)=-G0(I)
      GH(I)=-G0(I)
```

```
      PH(I)=-G0(I)
  170 CONTINUE
      J=0
      GO TO 200
C
C     STEP 12
C     RESTART WITH Z0 AS THE NEW SEARCH DIRECTION
C
  180 DO 190 I=1,N
      T0(I)=Z0(I)
      T1(I)=Z0(I)
      GH(I)=G0(I)
      PH(I)=Z0(I)
  190 CONTINUE
C
C     STEP 13
C     INITIALIZE SOME OF THE VARIABLES BEFORE RESTARTING
C     GO TO STEP 3
C
      J=0
  200 DO 210 I=1,N
      W1(I)=0.
      Z1(I)=0.
      X(I)=X1(I)
  210 CONTINUE
      GO TO 60
  220 CONTINUE
      IF (IPR.EQ.0) GO TO 360
C     PRINT OUT THE SOLUTION FOUND BY THE PACKAGE
      WRITE (ICH,330)
      WRITE (ICH,340)
      CALL FUNCT (N,M,XSTAR,GD,F)
      NFE=NFE+1
      NEFE=NFE+N*NGE
      CALL SECOND (TM2)
      CPU=TM2-TM1
      WRITE (ICH,350) F
      WRITE (ICH,300)
      DO 230 I=1,N
      WRITE (ICH,310) I,XSTAR(I),G0STAR(I)
  230 CONTINUE
      IF (ITER.EQ.ITERM) IFLAG=1
      WRITE (ICH,240) IFLAG
      WRITE (ICH,250) ITER
      WRITE (ICH,260) NEFE
      WRITE (ICH,270) CPU
      WRITE (ICH,280)
  240 FORMAT (/,10X,"TYPE OF SOLUTION (IFLAG)",38(1H.),2X,I2,/)
  250 FORMAT (10X,"NUMBER OF ITERATIONS ",38(1H.),I7,/)
  260 FORMAT (10X,"EFFECTIVE FUNCTION EVALUATIONS (EFE)  ",20(1H.),I9,/)
  270 FORMAT (10X,"EXECUTION TIME (IN SECONDS)",30(1H.),F9.3)
  280 FORMAT (1H1)
  290 FORMAT (/,10X,"STARTING POINT.",9X,"OBJECTIVE FUNCTION :",2X,1PE12
     1.5,/)
  300 FORMAT (/,26X,"VARIABLES",28X,"GRADIENT",/)
  310 FORMAT (13X,I6,4X,E12.5,25X,E12.5)
  320 FORMAT (//,10X,"SOLUTION AT ITERATION NUMBER ",1X,I4)
  330 FORMAT (/////,7X,"SOLUTION")
  340 FORMAT (7X,9(1H-),/)
  350 FORMAT (34X,"OBJECTIVE FUNCTION :",2X,1PE12.5,/)
  360 RETURN
      END
```

```
C
C
      FUNCTION VALUE (A,B,C,D,N)
C
C     THIS FUNCTION CALCULATES THE VALUE OF THE QUOTIENT OF THE PRODUCT
C     OF THE TWO VECTORS OF A AND B DIVIDED BY THE PRODUCT OF C AND D.
C
      REAL A(N),B(N),C(N),D(N)
      P=0.
      R=0.
      DO 10 I=1,N
      P=P+A(I)*B(I)
      R=R+C(I)*D(I)
   10 CONTINUE
      VALUE=P/R
      RETURN
      END
C
C
      FUNCTION ANORM (A,N)
C
C     THIS FUNCTION EVALUATES THE NORM OF A VECTOR A.
C
      REAL A(N)
      P=0.
      DO 10 I=1,N
      P=P+A(I)*A(I)
   10 CONTINUE
      ANORM=SQRT(P)
      RETURN
      END
C
C
      FUNCTION BNORM (A,B,N)
C
C     THIS FUNCTION EVALUATES THE VALUE OF THE PRODUCT OF THE TWO
C     VECTORS A AND B.
C
      REAL A(N),B(N)
      P=0.
      DO 10 I=1,N
      P=P+A(I)*B(I)
   10 CONTINUE
      BNORM=P
      RETURN
      END
```

```
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     IN THE SUBROUTINE FMCG2.IT FIRST PERFORMES A PARABOLIC
C     INTERPOLATION USING THE VALUES OF THE FUNCTION AND ITS
C     DERIVATIVE AT THE STARTING POINT AND THE FUNCTION VALUE
C     AT AN OFFSET POINT. IF THE PREDICTED FUNCTION VALUE
C     DOES NOT SATISFY THE WOLFE CONDITIONS THEN THE ARMIJO[1] PROC-
C     EDURE IS ADOPTED BASED ON THAT STEP.
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
      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     ARMIJIO PROCEDURE IS ADOPTED
C
   50 CONTINUE
      FMIN=F1
```

```
      IF (F1.GT.F) ALMIN=0.
      IF (F1.GT.F) FMIN=F
      IF (F1.LT.F) ALMIN=A1
      E1=F-F1
      E2=A1*ABS(G01)/CONS
      IF (E1.LE.E2) GO TO 100
   60 CONTINUE
      AL=2.*A1
      DO 70 I=1,N
      XD(I)=X(I)+AL*T0(I)
   70 CONTINUE
      CALL FUNCT (N,M,XD,GD,F3)
      NFE=NFE+1
      LK=LK+1
      E3=F-F3
      E4=AL*ABS(G01)/CONS
      IF (F3.LT.FMIN) ALMIN=AL
      IF (F3.LT.FMIN) FMIN=F3
      IF (E3.LT.E4.OR.LK.GE.10) GO TO 80
      A1=AL
      F1=F3
      GO TO 60
   80 CONTINUE
      AL=A1
      F=F1
      IF (F1.GT.FMIN) F=FMIN
      IF (F1.GT.FMIN) AL=ALMIN
      DO 90 I=1,N
      D0(I)=AL*T0(I)
   90 CONTINUE
      RETURN
  100 CONTINUE
      AL=A1/2.
      DO 110 I=1,N
      XD(I)=X(I)+AL*T0(I)
  110 CONTINUE
      CALL FUNCT (N,M,XD,GD,F4)
      NFE=NFE+1
      LL=LL+1
      E5=F-F4
      E6=AL*ABS(G01)/CONS
      IF (FMIN.GT.F4) ALMIN=AL
      IF (FMIN.GT.F4) FMIN=F4
      IF (E5.GT.E6) GO TO 120
      IF (LL.GE.10) GO TO 140
      A1=AL
      F1=F4
      GO TO 100
  120 CONTINUE
      F=F1
      AL=A1
      IF (F1.GT.FMIN) AL=ALMIN
      IF (F1.GT.FMIN) F=FMIN
      DO 130 I=1,N
      D0(I)=AL*T0(I)
  130 CONTINUE
      RETURN
  140 CONTINUE
      F=F4
      DO 150 I=1,N
      D0(I)=AL*T0(I)
  150 CONTINUE
      RETURN
      END
```