```
C **** SAMPLE CALLING PROGRAM FOR NEWTON ****
C
       IMPLICIT REAL*8(A-Z)
       INTEGER I,J,K,L,M,N,ITER,IFLAG,LIMIT
       LOGICAL PRNT
C
       DIMENSION X(4),H(4,4),G(4)
C
       N=4
       R1=0.00001
       R2=0.00005
       GAMMA=2.0
       BETA=0.5
       EPS=0.000001
       LIMIT=49
       PRNT=.FALSE.
       DO 10 I=1,N
      PRINT*,'ENTER INITIAL COORDINATE ',I
         READ(*,*)X(I)
10      CONTINUE
        CALL NEWTON(N,X,F,G,H,ITER,IFLAG,R1,R2,
     $ GAMMA,BETA,EPS,LIMIT,PRNT)
       PRINT*,'*****************************************'
       PRINT*,'NUMBER OF NEWTON ITERATIONS ',ITER
       PRINT*,'SOLUTION INDICATOR FLAG: ',IFLAG
       DO 20 I=1,N
         PRINT*,'FINAL COORDINATE',I,' = ',X(I)
20     CONTINUE
       PRINT*,'*****************************************'
         PRINT *, 'PROGRAM ENDED; PRESS <CR> TO EXIT '
       PAUSE

         END


       SUBROUTINE NEWTON(N,X,F,G,H,ITER,IFLAG,R1,R2,GAMMA,BETA,
     $EPS,LIMIT,PRNT)
C       SOLVES UNCONSTRAINED MINIMIZATION PROBLEMS USING A
C       MODIFIED VERSION OF NEWTON'S METHOD.
C      N: DIMENSION OF PROBLEM
C      F: FUNCTION VALUE
C      G: GRADIENT VECTOR
C      H: HESSIAN MATRIX
C      ITER: CURRENT ITER□ATION INDEX
C      IFLAG: INDICATES CONDITIONS ON EXIT
C            IFLAG=0  NORMAL EXIT HESSIAN NOT MODIFIED EVER
C            IFLAG=1  NORMAL EXIT HESSIAN MODIFIED ONCE OR MORE
C            IFLAG=2  EXIT DUE TO MAX # OF ITERATIONS REACHED
C            IFLAG□□=3  ABNORMAL EXIT / ARMIJO STEPSIZE RULE FAILED
C      R1: PARAMETER USED IN HESSIAN MODIFICATION
C          RECOMMENDED VALUE (0.0000001 TO 0.001)
C      R2: PARAMETER USED IN HESSIAN MODI□FICATION
C            RECOMMENDED VALUE (0.001 TO 10)
C        GAMMA: PARAMETER USED IN LINE SEARCH WHEN HESSIAN IS MODIFIED
C             RECOMMENDED VALUE (2 TO 10)
C               A LARGER VALUE OF GAMMA RESULTS IN A LINE
C               SEARCH OVER A WIDER RANGE
```

```
C     BETA: PARAMETER USED IN STEPSIZE RULE; IF THE CURRENT STEP
C           DOES NOT GIVE A REDUCTION IN FUNCTION VALUE
C             THE STEPSIZE IS CUT DOWN BY A FACTOR BETA
C             RECOMMENDED VALUE 0.1 TO 0.4
C       EPS: TERMINATION PARAMETER; IF THE NORM SQUARED OF THE
C          GRADIENT THAN EPS  THE ITERATIONS ARE STOPPED
C       LIMIT: MAX # OF ITERATIONS ALLOWED
C       PRNT: LOGICAL THAT SPECIFIES DETAILED PRINTING IF SET TO TRUE
C
C       NOTE: THE ABOVE RECOMMENDED RANGES FOR PARAMETER VALUES
C             ARE NOT FOOLPROOF. EXPERIMENTATION WITH OTHER VALUES
C             MAY BE REQUIRED IN UNUSUAL CASES.
C
       IMPLICIT REAL*8(A-Z)
       INTEGER I,J,K,L,M,N,ITER,LIMIT,IFLAG,NRED,INDEX
C
       DIMENSION X(N),Y(500),G(N),H(N,N),D(500),INDEX(500)
       LOGICAL MODHES,PRNT
C
C  ***** INITIALIZATION *****
C
       IFLAG=0
       MODHES=.FALSE.
       ITER=0
       CURF=FUNCT(N,X)
       CALL DERIVS(N,X,G,H)
       PRINT*,'STARTING FUNCTION VALUE',CURF
      DO 1000 I=1,N
         IF (PRNT) PRINT*,'STARTING COORDINATE',I,'=',X(I)
1000     CONTINUE
      DO 1010 I=1,N
      DO 1020 J=1,N
        IF (PRNT) PRINT*,'HESSIAN ELEMENT',I,J,' = ',H(I,J)
1020  CONTINUE
1010    CONTINUE
C
C  ****** START NEW ITERATION ******
C
10      CONTINUE
      ITER=ITER+1
      PRINT*,'************************************'
      PRINT*,'ITERATION ',ITER
C  **** GET MODIFIED CHOLESKY FACTORIZATION ****
C
       CALL MODCHLS(N,H,R1,R2,MODHES,INDEX)
C
C  **** GET NEWTON DIRECTION ****
C
       CALL DIRECT(N,G,H,D,INDEX)
C
C  **** GET INITIAL STEPSIZE ****
C
       IF (MODHES) THEN
       CALL INSTEP(N,X,D,GAMMA,S,PRNT)
         IFLAG=1
       PRINT*,'HESSIAN MODIFIED,  INITIAL STEPSIZE =',S
C    **** MODIFY PARAMETER R2 ****
```

```
          IF (S.LT.0.2) R2=R2*5.0
          IF (S.GT.0.9) R2=R2/5.0
        ELSE
          S=1.0
        END IF
C
C  **** GET INNER PRODUCT FOR THE ARMIJO RULE ****
C
        PRODUCT=0.0
        DO 12 I=1,N
          PRODUCT=PRODUCT+G(I)*D(I)
12      CONTINUE
C
C  STEPSIZE SELECTION BY THE ARMIJO RULE
C
        ALPHA=1.0
        NRED=0

15      STEPSIZE=ALPHA*S
        DO 20 I=1,N
          Y(I)=X(I)-STEPSIZE*D(I)
20      CONTINUE
        F=FUNCT(N,Y)
        IF (CURF.LT.F+0.0001*STEPSIZE*PRODUCT) THEN
          ALPHA=BETA*ALPHA
          NRED=NRED+1
          IF (NRED.GT.10) THEN
            IFLAG=3
            PRINT*,'EXCESSIVE # OF STEPSIZE REDUCTIONS'
            RETURN
          ELSE
            GO TO 15
          END IF
        END IF
C
C  **** LINE SEARCH HAS SUCCEEDED ****
C
        CURF=F
        DO 30 I=1,N
          X(I)=Y(I)
30      CONTINUE
        CALL DERIVS(N,X,G,H)
       PRINT*,'# OF STEPSIZE REDUCTIONS ',NRED
       PRINT*,'NEW FUNCTION VALUE = ',CURF
        DO 50 I=1,N
       IF (PRNT) PRINT*,'COORDINATE ',I,' = ',X(I)
50      CONTINUE
        IF (ITER.GT.LIMIT) THEN
          IFLAG=2
          PRINT*,'LIMIT ON THE NUMBER OF ITERATIONS ECXEEDED'
C
C  ***** MAX # OF ITERATIONS REACHED *****
C
          RETURN
        END IF
C
C   CHECK FOR GRAD BEING SMALL & IF NOT GO FOR ANOTHER ITER
```

```
C
      MAXGRAD=0
      DO 40 I=1,N
        MAXGRAD=MAXGRAD+G(I)**2
40    CONTINUE
      IF (MAXGRAD.LT.EPS) THEN
        PRINT*,' GRADIENT TOLERANCE ATTAINED; OPTIMIZATION COMPLETED'
        RETURN
      ELSE
        PRINT*,'SQUARED NORM OF THE GRADIENT =',MAXGRAD
        GO TO 10
      END IF
      END




      SUBROUTINE MODCHLS(N,H,R1,R2,MODHES,INDEX)

C     THIS ROUTINE CALCULATES A MODIFIED CHOLESKY FACTORIZATION
C     OF THE HESSIAN MATRIX H, AND RETURNS THE RESULT IN THE UPPER
C     TRIANGULAR PORTION OF H.
C     THE FACTORIZATION IS MODIFIED AS FOLLOWS:
C     THE MAXIMUM ABSOLUTE DIAGONAL ELEMENT IS FOUND. IF DURING THE
C     FACTORIZATION A DIAGONAL ELEMENT THAT IS LESS THAN R1*MAXGIAG IS
C     FOUND, THAT ELEMENT IS RAISED TO R2*MAXDIAG
C     THE USER SHOULD EXPERIMENT FOR PROPER SELECTION OF THE VALUES
C     OF R1 AND R2
C     ROW INTERCHANGES ARE INTRODUCED TO IMPROVE PERFORMANCE
C     ON RETURN INDEX(I) GIVES THE ORIGINAL POSITION OF THE
C     COORDINATE THAT HAS BEEN MOVED TO THE I-TH POSITION
C     AFTER INTERCHANGES
      IMPLICIT REAL*8(A-Z)
      INTEGER I,J,K,L,M,N,INDEX,PIVOT
      LOGICAL MODHES
      DIMENSION H(N,N),TEMP(500),INDEX(N)
C
C  **** INITIALIZATION ****
      MODHES=.FALSE.
      MAXDIAG=0.0
      DO 5 I=1,N
        IF (ABS(H(I,I)).GT.MAXDIAG) MAXDIAG=ABS(H(I,I))
        INDEX(I)=I
5     CONTINUE
      THRESH1=R1*MAXDIAG
      THRESH2=R2*MAXDIAG
C
C  **** START FACTORING ****
C
      DO 10 I=1,N
C
C   **** INTERCHANGE ROWS TO OPERATE NEXT ON THE ROW W/ LARGEST
C        DIAGONAL ELEMENT
C
      MAXDIAG=H(I,I)
      PIVOT=I
      DO 12 K=I,N
```

```
          IF (H(K,K).GT.MAXDIAG) THEN
            MAXDIAG=H(K,K)
            PIVOT=K
          END IF
12      CONTINUE
        IF (PIVOT.GT.I) THEN
          M=INDEX(I)
          INDEX(I)=INDEX(PIVOT)
          INDEX(PIVOT)=M
C  *** DO ROW INTERCHANGE ***
          DO 13 K=1,N
            TEMP(K)=H(I,K)
            H(I,K)=H(PIVOT,K)
13        CONTINUE
          DO 14 K=1,N
            H(PIVOT,K)=TEMP(K)
14        CONTINUE
C  *** DO COLUMN INTERCHANGE ***
       DO 15 K=1,N
         TEMP(K)=H(K,I)
         H(K,I)=H(K,PIVOT)
15     CONTINUE
       DO 16 K=1,N
         H(K,PIVOT)=TEMP(K)
16     CONTINUE
        END IF
C
C  ***** END OF ROW AND COLUMN  INTERCHANGE *****
C
        DO 18 J=I,N
          TEMP(J)=H(I,J)
          IF (I.GT.1) THEN
            DO 20 K=1,I-1
              TEMP(J)=TEMP(J)-H(K,J)*H(K,I)
20          CONTINUE
          END IF
18      CONTINUE
          IF (TEMP(I).LT.THRESH1) THEN
            TEMP(I)=THRESH2
            MODHES=.TRUE.
          END IF
          DIAG=SQRT(TEMP(I))
          H(I,I)=DIAG
          IF (I.LT.N) THEN
            DO 30 J=I+1,N
              H(I,J)=TEMP(J)/DIAG
30          CONTINUE
          END IF
10      CONTINUE
        RETURN
        END



       SUBROUTINE DIRECT(N,G,H,D,INDEX)
C      THIS ROUTINE CALCULATES THE NEWTON DIRECTION USING THE FACTORED
```

```
C     HESSIAN AND THE GRADIENT. THE NEWTON DIRECTION D IS THE
C     SOLUTION TO THE SYSTEM HD=G

      IMPLICIT REAL*8(A-Z)
      INTEGER I,J,K,L,M,N,INDEX
      DIMENSION H(N,N),G(N),D(N),DF(500),GI(500),INDEX(N)

C    REARRANGE GRAD COORDINATES AND STORE ORIGINAL GRAD IN ARRAY GI
      DO 5 I=1,N
        GI(I)=G(I)
5     CONTINUE
      DO 7 I=1,N
        G(I)=GI(INDEX(I))
7     CONTINUE
C
C
C  ***** FORWARD ELIMINATION *****
C
      DF(1)=G(1)/H(1,1)
      DO 20 I=2,N
        DF(I)=G(I)
        DO 10 J=1,I-1
          DF(I)=DF(I)-H(J,I)*DF(J)
10      CONTINUE
        DF(I)=DF(I)/H(I,I)
20    CONTINUE
C
C  **** BACK SUBSTITUTION ****
C
      D(N)=DF(N)/H(N,N)
      DO 40 K=1,N-1
      I=N-K
        D(I)=DF(I)
        DO 30 J=I+1,N
          D(I)=D(I)-H(I,J)*D(J)
30      CONTINUE
        D(I)=D(I)/H(I,I)
40    CONTINUE
C
C   ****** REARRANGE DIRECTION COORDINATES ******
C
      DO 50 I=1,N
        DF(I)=D(I)
          G(I)=GI(I)
50    CONTINUE
      DO 60 I=1,N
        M=INDEX(I)
        D(M)=DF(I)
60    CONTINUE
      RETURN
      END




      SUBROUTINE INSTEP(N,X,D,GAMMA,S,PRNT)
```

```
C     THIS SUBROUTINE CALCULATES AN INITIAL STEPSIZE FOR THE ARMIJO
C     STEPSIZE RULE. IT USES A QUADRATIC INTERPOLATION USING THE
C     FUNCTION VALUES AT THE VECTORS X, X-D, X-GAMMA*D.
      IMPLICIT REAL*8(A-Z)
      INTEGER I,J,K,L,M,N,COUNTER
      LOGICAL PRNT
      DIMENSION X(N),D(N),X2(500),X3(500)
C
C     GET A STEPSIZE THAT LEADS TO A COST REDUCTION
C
      F1=FUNCT(N,X)
C     START WITH A STEPSIZE OF 0.1
      STEP=0.1
      COUNTER=0
C
1     DO 2 I=1,N
        X2(I)=X(I)-STEP*D(I)
2     CONTINUE
C
      F2=FUNCT(N,X2)
      IF (PRNT) PRINT*,'STEPSIZE =',STEP,'COST =',F2
      IF (F1.LE.F2) THEN
        F3=F2
        STEP=0.2*STEP
        COUNTER=COUNTER+1
        IF (COUNTER.GE.15) THEN
          S=STEP
          RETURN
        END IF
        GO TO 1
      END IF
C
C     FIND THREE POINT PATTERN
C
      IF (COUNTER.GT.0) THEN
        S1=0
        S2=STEP
        S3=5*STEP
      END IF
C
      IF (COUNTER.EQ.0) THEN
        S1=0
        S2=STEP
        S3=GAMMA*STEP
C
3       CONTINUE
        DO 4 I=1,N
          X3(I)=X(I)-S3*D(I)
4       CONTINUE
        F3=FUNCT(N,X3)
        IF (PRNT) PRINT*,'STEPSIZE =',S3,'COST =',F3
        IF (F3.LT.F2) THEN
          COUNTER=COUNTER+1
C       RETURN IF THE LENGTH OF THE STEP EXCEEDS A LIMIT
          IF (COUNTER.GE.15) THEN
            S=S3
            RETURN
```

```
          END IF
          S1=S2
          S2=S3
          S3=GAMMA*S3
          F1=F2
          F2=F3
          GO TO 3
        END IF
      END IF
C
C       3 POINT PATTERN IS S1,S2,S3 W/ COSTS F1,F2,F3
C
        A1=S2-S3
        B1=S2**2-S3**2
        A2=S3-S1
        B2=S3**2-S1**2
        A3=S1-S2
        B3=S1**2-S2**2
      T=A1*F1+A2*F2+A3*F3
      IF (T.EQ.0) THEN
      S=1.0
      RETURN
      END IF
      S=0.5*(B1*F1+B2*F2+B3*F3)/T
C
C     FIND THE BEST STEPSIZE OUT OF S AND S2
C
      DO 10 I=1,N
        X2(I)=X(I)-S*D(I)
10    CONTINUE
      FS=FUNCT(N,X2)
      IF (PRNT) PRINT*,'INTERPOLATION STEP =',S,'COST =',FS
      IF (F2.LT.FS) S=S2
      RETURN
      END


C      THIS A SAMPLE PROBLEM FOR TESTING NEWTON'S METHOD
C     ROSENBROCK'S BANANA FUNCTION
       FUNCTION FUNCT(N,X)
       IMPLICIT REAL*8(A-Z)
       INTEGER I,J,K,L,M,N
       DIMENSION X(N)
C
      FUNCT=(1-X(1))**2
      DO 10 I=2,N
        FUNCT=FUNCT+100*(X(I)-X(1)**2)**2
10    CONTINUE
       RETURN
       END



       SUBROUTINE DERIVS(N,X,G,H)
       IMPLICIT REAL*8(A-Z)
       INTEGER I,J,K,L,M,N
       DIMENSION X(N),G(N),H(N,N)
```

```
C
      DO 2 I=1,N
      DO 4 J=1,N
        H(I,J)=0
4     CONTINUE
2     CONTINUE
C
C
      G(1)=2*(X(1)-1)
      H(1,1)=2
      DO 10 I=2,N
        G(1)=G(1)+400*X(1)*(X(1)**2-X(I))
        H(1,1)=H(1,1)+800*X(1)**2+400*(X(1)**2-X(I))
10    CONTINUE
       DO 20 I=2,N
         G(I)=200*(X(I)-X(1)**2)
       H(I,I)=200
       H(I,1)=-400*X(1)
       H(1,I)=H(I,1)
20     CONTINUE
       RETURN
       END
```