INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-252

QUSNTN - A PROGRAM FOR UNCONSTRAINED FUNCTION MINIMIZATION

J.W. Bandler, M.A. El-Kady and M.R.M. Rizk

September 1980

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

QUSNTN - A PROGRAM FOR UNCONSTRAINED FUNCTION MINIMIZATION

J.W. Bandler, M.A. El-Kady and M.R.M. Rizk

Abstract

QUSNTN is a package of a compiled library subroutine and a user-supplied subroutine. The package implements the 1972 version of Fletcher's quasi-Newton method for unconstrained optimization [1] with the minor changes outlined in FLOPT5 [2], a previously produced package for minimax optimization. The program has been tested on a CYBER 170 computer. The report includes a FORTRAN listing of the program, a user's guide and illustrating examples.

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

The authors are with the Group on Simulation, Optimization and Control and Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, L8S 4L7.

INTRODUCTION

The purpose of producing this package is to provide a simple user-oriented program for unconstrained function minimization. Although the available FLOPT-series of programs, constructed originally for constrained optimization, may also be used for unconstrained optimization, the user has to realize more general features, supply more information and define more dimensional variables than what is actually needed. In the QUSNTN program, we have extracted the relevant computing statements and reduced the information supplied by the user to a minimum. The subroutines of the package can, however, be attached to a general constrained optimization program via suitable calling statements.

The user has to supply the external subroutine XUGU which defines the objective function and its gradients with respect to the variables. Subroutine QUSNTN is called to perform the function minimization. The user defines the number of variables, the maximum allowable number of function evaluations, the initial values of the variables, and the vector EPS used to check the convergence criterion. On output, the values of variables and objective function at the local minimum reached, the number of iterations and number of function evaluations performed are available.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the effort provided by Mr. D. Sinha in the programming of the FLOPT5 package of subroutines, from which the present package has evolved.

DEFINITION OF VARIABLES

Integer Variables

IFN Index for the number of function evaluations performed

ITN Index for the number of iterations performed

MAX Maximum allowable number of function evaluations

N Number of variables

Real Variables

EPS Array of N elements of "small" positive values used for testing the convergence with respect to variations in variables between successive iterations

GU Array of N elements containing the current values of partial derivatives of the objective function w.r.t. the variables

H Array of dimension N*(N+1)/2 used to store the current estimate of the Hessian matrix

U Current value of the objective function to be minimized

X Array of N elements containing the current values of variables

W Array of dimension 4*N used as working area

EXAMPLES

We present three examples to illustrate the use of the package for unconstrained minimization.

Example 1 (Rosenbrock's function [3])

Minimize

$$U = 100(x_1^2 - x_2)^2 + (1 - x_1)^2.$$

The function has a minimum value of zero at $x_1 = x_2 = 1.0$. The starting point used is $x_1 = -1.2$, $x_2 = 1.0$.

```
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION X(2),GU(2),H(3),W(8),EPS(2)
      N=2
      X(1)=-1.2
      X(2)=1.0
      EPS(1)=1.0E-8
      EPS(2)=1.0E-8
      MAX=100
      CALL QUSNTN(N,X,U,GU,H,W,EPS,MAX,ITN,IFN)
      WRITE(6,1) ITN,IFN,(X(I),I=1,N),U,(GU(I),I=1,N)
      STOP
    1 FORMAT(1H1,///,3X,*THE OBTAINED RESULTS*,/,3X,20(*-*),//,3X,
     **NO. OF ITERATIONS PERFORMED = *,I3,//,3X,
     **NO. OF FUNCTION EVALUATIONS = *,I3,//,3X,
     **VARIABLES AT SOLUTION = *,2(E13.6,3X),//,3X,
     **FUNCTION AT SOLUTION = *,E13.6,//,3X,
     **GRADIENTS AT SOLUTION = *,2(E13.6,3X),5(/))
      END
```

```
      SUBROUTINE XUGU(N,X,U,GU)
C
C     THIS SUBROUTINE DEFINES FUNCTION U AND ITS GRADIENTS
C
      DIMENSION X(1),GU(1)
      A=X(1)*X(1)
      B=A-X(2)
      C=1.0-X(1)
      U=100.0*B*B+C*C
      GU(1)=400.0*X(1)*(A-X(2))-C-C
      GU(2)=-200.0*B
      RETURN
      END
```

THE OBTAINED RESULTS

NO. OF ITERATIONS PERFORMED = 37

NO. OF FUNCTION EVALUATIONS = 47

VARIABLES AT SOLUTION = .100000E+01 .100000E+01

FUNCTION AT SOLUTION = .258746E-24

GRADIENTS AT SOLUTION = -.760281E-11 .426326E-11

Example 2

Minimize

$$U = [x_1 - (x_2 - x_3)^2]^2 + [x_3 - (1 + x_2 - x_4)^2]^2 + x_1^2 + x_3^2.$$

The function has a minimum of 0.0 at $x_1 = x_2 = x_3 = 0.0$ and $x_4 = 1.0$. The starting point used is $x_1 = x_2 = x_3 = x_4 = 2.0$.

```
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION X(4),GU(4),H(10),W(16),EPS(4)
      N=4
      MAX=100
      DO 1 I=1,N
      X(I)=2.0
      EPS(I)=1.0E-8
    1 CONTINUE
      CALL QUSNTN(N,X,U,GU,H,W,EPS,MAX,ITN,IFN)
      WRITE(6,2) ITN,IFN,(X(I),I=1,N),U,(GU(I),I=1,N)
      STOP
    2 FORMAT(1H1,///,3X,*THE OBTAINED RESULTS*,/,3X,20(*-*),//,3X,
     **NO. OF ITERATIONS PERFORMED = *,I3,//,3X,
     **NO. OF FUNCTION EVALUATIONS = *,I3,//,3X,
     **VARIABLES AT SOLUTION = *,4(E10.3,2X),//,3X,
     **FUNCTION AT SOLUTION = *,E13.6,//,3X,
     **GRADIENTS AT SOLUTION = *,4(E10.3,2X),5(/))
      END
```

```
      SUBROUTINE XUGU(N,X,U,GU)
C
C     THIS SUBROUTINE DEFINES FUNCTION U AND ITS GRADIENTS
C
      DIMENSION X(1),GU(1)
      A=X(2)-X(3)
      B=X(1)-A*A
      C=1.0+X(2)-X(4)
      D=X(3)-C*C
      U=B*B+D*D+X(1)*X(1)+X(3)*X(3)
      GU(1)=2.0*(B+X(1))
      GU(3)=4.0*A*B
      GU(4)=4.0*C*D
      GU(2)=-GU(3)-GU(4)
      GU(3)=GU(3)+2.0*(D+X(3))
      RETURN
      END
```

THE OBTAINED RESULTS

NO. OF ITERATIONS PERFORMED = 79

NO. OF FUNCTION EVALUATIONS = 90

VARIABLES AT SOLUTION = .253E-12 .711E-06 .333E-12 .100E+01

FUNCTION AT SOLUTION = .346758E-24

GRADIENTS AT SOLUTION = -.692E-16 -.351E-18 .803E-14 .107E-17

Example 3 (Quadratic function)

Minimize

$$U = x_1^2 + 4\, x_1 x_2 + 5\, x_2^2 + 2\, x_1 - x_2 + 7.25.$$

The function has a minimum of 0.0 at $x_1 = -6.0$ and $x_2 = 2.5$. The starting point used is $x_1 = x_2 = 0.0$. The output message indicates that U cannot be minimized any further. The zero gradients on output indicate that the optimum has been already achieved.

```
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION X(2),GU(2),H(3),W(8),EPS(2)
      N=2
      X(1)=0.0
      X(2)=0.0
      EPS(1)=1.0E-8
      EPS(2)=1.0E-8
      MAX=100
      CALL QUSNTN(N,X,U,GU,H,W,EPS,MAX,ITN,IFN)
      WRITE(6,1) ITN,IFN,(X(I),I=1,N),U,(GU(I),I=1,N)
      STOP
    1 FORMAT(1H1,///,3X,*THE OBTAINED RESULTS*,/,3X,20(*-*),//,3X,
     **NO. OF ITERATIONS PERFORMED = *,I3,//,3X,
     **NO. OF FUNCTION EVALUATIONS = *,I3,//,3X,
     **VARIABLES AT SOLUTION = *,2(E13.6,3X),//,3X,
     **FUNCTION AT SOLUTION = *,E13.6,//,3X,
     **GRADIENTS AT SOLUTION = *,2(E13.6,3X),5(/))
      END
```

```
      SUBROUTINE XUGU(N,X,U,GU)
C
C     THIS SUBROUTINE DEFINES FUNCTION U AND ITS GRADIENTS
C
      DIMENSION X(1),GU(1)
      U=X(1)*X(1)+4.0*X(1)*X(2)+5.0*X(2)*X(2)+2.0*X(1)-X(2)+7.25
      GU(1)=2.0*X(1)+4.0*X(2)+2.0
      GU(2)=4.0*X(1)+10.0*X(2)-1.0
      RETURN
      END
```

U CANNOT BE DECREASED ANY FURTHER-CHECK OPTIMALITY

THE OBTAINED RESULTS

NO. OF ITERATIONS PERFORMED = 5

NO. OF FUNCTION EVALUATIONS = 6

VARIABLES AT SOLUTION = -.600000E+01 .250000E+01

FUNCTION AT SOLUTION = -.198952E-12

GRADIENTS AT SOLUTION = 0. 0.

REFERENCES

[1] R. Fletcher, "FORTRAN Subroutines for minimization by quasi-Newton methods", Atomic Energy Research Establishment, Harwell, Berkshire, England, Report AERE-R7125, 1972.

[2] J.W. Bandler and D. Sinha, "FLOPT5 - a program for minimax optimization using the accelerated least pth algorithm", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-218, 1978.

[3] H.H. Rosenbrock, "An automatic method for finding the greatest or least value of a function", Comp. J., vol. 3, Oct. 1960, pp. 175-184.

```
      SUBROUTINE QUSNTN(N,X,U,GU,H,W,EPS,MAX,ITN,IFN)
      DIMENSION X(1),GU(1),H(1),W(1),EPS(1)
C
C     INITIALIZATION
C
      NP=N+1
      N1=N-1
      NN=N*NP/2
      IV=N+N
      IB=IV+N
      IEXIT=0
C
C     SETTING INITIAL H TO IDENTITY MATRIX
C
      IJ=NN+1
      DO 2 I=1,N
      DO 1 J=1,I
      IJ=IJ-1
      H(IJ)=0.0
    1 CONTINUE
      H(IJ)=1.0
    2 CONTINUE
      DMIN=1.0
      ITN=0
      IFN=0
      CALL XUGU(N,X,U,GU)
      IFN=IFN+1
      DF=U
      IF(DF.LE.0.0) DF=1.0
C
C     START OF ITERATIONS
C
    3 ITN=ITN+1
C
C     DETERMINATION OF DIRECTION OF SEARCH
C
      W(1)=-GU(1)
      DO 5 I=2,N
      IJ=I
      I1=I-1
      Z=-GU(I)
      DO 4 J=1,I1
      Z=Z-H(IJ)*W(J)
      IJ=IJ+N-J
    4 CONTINUE
      W(I)=Z
    5 CONTINUE
      W(IV)=W(N)/H(NN)
      IJ=NN
```

```
      DO 7 I=1,N1
      IJ=IJ-1
      Z=0.0
      DO 6 J=1,I
      Z=Z+H(IJ)*W(N+NP-J)
      IJ=IJ-1
    6 CONTINUE
      W(IV-I)=W(N-I)/H(IJ)-Z
    7 CONTINUE
C
C     EVALUATION OF PRODUCT OF GU WITH DIRECTION OF SEARCH
C
      GS=0.0
      DO 8 I=1,N
      GS=GS+W(N+I)*GU(I)
    8 CONTINUE
      IEXIT=2
      IF(GS.GT.0.0) GO TO 31
      IF(GS.EQ.0.0) GO TO 33
C
C     CALCULATION OF LINE SEARCH PARAMETER ALPHA
C
      GS0=GS
      ALPHA=-2.0*DF/GS
      IF(ALPHA.GT.1.0) ALPHA=1.0
      DF=U
      TOT=0.0
    9 IEXIT=3
      IF(IFN.GT.MAX) GO TO 32
      ICON=0
      IEXIT=1
      DO 10 I=1,N
      Z=ALPHA*W(N+I)
      IF(ABS(Z).GE.EPS(I)) ICON=1
      X(I)=X(I)+Z
   10 CONTINUE
      CALL XUGU(N,X,FY,W)
      IFN=IFN+1
      GYS=0.0
      DO 11 I=1,N
      GYS=GYS+W(I)*W(N+I)
   11 CONTINUE
```

```
      IF(FY.GE.U) GO TO 12
      IF(ABS(GYS/GS0).LE.0.9) GO TO 14
      IF(GYS.GT.0.0) GO TO 12
C
C     PERFORMANCE OF LINEAR EXTRAPOLATION FOR SEARCH PARAMETER
C
      TOT=TOT+ALPHA
      Z=10.0
      IF(GS.LT.GYS) Z=GYS/(GS-GYS)
      IF(Z.GT.10.0) Z=10.0
      ALPHA=ALPHA*Z
      U=FY
      GS=GYS
      GO TO 9
C
C     CUBIC INTERPOLATION FOR SEARCH PARAMETER
C
   12 DO 13 I=1,N
      X(I)=X(I)-ALPHA*W(N+I)
   13 CONTINUE
      IF(ICON.EQ.0) GO TO 30
      Z=3.0*(U-FY)/ALPHA+GYS+GS
      ZZ=SQRT(Z*Z-GS*GYS)
      GZ=GYS+ZZ
      Z=1.0-(GZ-Z)/(ZZ+GZ-GS)
      ALPHA=ALPHA*Z
      GO TO 9
C
C     UPDATING THE MATRIX H
C
   14 ALPHA=TOT+ALPHA
      U=FY
      IF(ICON.EQ.0) GO TO 28
      DF=DF-U
      DGS=GYS-GS0
      LINK=1
C
C     TEST FOR THE USE OF COMPLEMENTARY DFP FORMULA
C
      IF(DGS+ALPHA*GS0.GT.0.0) GO TO 16
```

```
      DO 15 I=1,N
      W(N+I)=W(I)-GU(I)
   15 CONTINUE
      SIG=1.0/(ALPHA*DGS)
      GO TO 23
   16 ZZ=ALPHA/(DGS-ALPHA*GS0)
      Z=DGS*ZZ-1.0
      DO 17 I=1,N
      W(N+I)=Z*GU(I)+W(I)
   17 CONTINUE
      SIG=1.0/(ZZ*DGS*DGS)
      GO TO 23
   18 LINK=2
      DO 19 I=1,N
      W(N+I)=GU(I)
   19 CONTINUE
      IF(DGS+ALPHA*GS0.GT.0.0) GO TO 20
      SIG=1.0/GS0
      GO TO 23
   20 SIG=-ZZ
      GO TO 23
   21 DO 22 I=1,N
      GU(I)=W(I)
   22 CONTINUE
      GO TO 3
   23 W(IV+1)=W(N+1)
      DO 25 I=2,N
      IJ=I
      I1=I-1
      Z=W(N+I)
      DO 24 J=1,I1
      Z=Z-H(IJ)*W(IV+J)
      IJ=IJ+N-J
   24 CONTINUE
      W(IV+I)=Z
   25 CONTINUE
      IJ=1
```

```
      DO 26 I=1,N
      IVI=IV+I
      IBI=IB+I
      Z=H(IJ)+SIG*W(IVI)*W(IVI)
      IF(Z.LE.0.0) Z=DMIN
      IF(Z.LT.DMIN) DMIN=Z
      H(IJ)=Z
      W(IBI)=W(IVI)*SIG/Z
      SIG=SIG-W(IBI)*W(IBI)*Z
      IJ=IJ+NP-I
   26 CONTINUE
      IJ=1
      DO 27 I=1,N1
      IJ=IJ+1
      I1=I+1
      DO 27 J=I1,N
      W(N+J)=W(N+J)-H(IJ)*W(IV+I)
      H(IJ)=H(IJ)+W(IB+I)*W(N+J)
   27 IJ=IJ+1
      IF(LINK-2) 18,21,21
   28 DO 29 I=1,N
      GU(I)=W(I)
   29 CONTINUE
   30 IF(IEXIT-2) 34,31,32
   31 WRITE(6,35) IEXIT
      CALL EXIT
   32 WRITE(6,36) IEXIT
      CALL EXIT
   33 IEXIT=4
      WRITE(6,37) IEXIT
   34 RETURN
   35 FORMAT(1H1,*IEXIT=*,I2/1H0,*EPS CHOSEN IS TOO SMALL*)
   36 FORMAT(1H1,*IEXIT=*,I2/1H0,*FUNCTION EVALUATIONS EXEEDED MAX*)
   37 FORMAT(1H1,*IEXIT=*,I2/1H0,*U CANNOT BE DECREASED ANY FURTHER-C[illegible]
     *K OPTIMALITY*)
      END
```

SOC-252

QUSNTN - A PROGRAM FOR UNCONSTRAINED FUNCTION MINIMIZATION

J.W. Bandler, M.A. El-Kady and M.R.M. Rizk

September 1980, No. of Pages: 22

Revised:

Key Words: Unconstrained optimization, quasi-Newton method, gradient minimization

Abstract: QUSNTN is a package of a compiled library subroutine and a user-supplied subroutine. The package implements the 1972 version of Fletcher's quasi-Newton method for unconstrained optimization [1] with the minor changes outlined in FLOPT5 [2], a previously produced package for minimax optimization. The program has been tested on a CYBER 170 computer. The report includes a FORTRAN listing of the program, a user's guide and illustrating examples.

Description: Contains FORTRAN listing, user's manual.
Source deck available for $50.00
The listing contains 203 statements of which 30 are comments.

Related Work: SOC-151, SOC-218.

Price: $30.00.