REMARK ON ALGORITHM 500

Minimization of Unconstrained Multivariate Functions [E4] [D.F. Shanno and K.H. Phua, *ACM Trans. Math. Softw.* 2, 1 (March 1976), 87–94.]

D.F. Shanno and K.H. Phua [Received 3 January 1979; revised 5 June 1979; accepted 26 December 1979]
D.F. Shanno, Department of Management Information Systems, College of Business and Public Administration, The University of Arizona, Tucson, AZ 85721; K.H. Phua, Department of Computer Science, Nanyang University, Upper Jurong Road, Singapore 22, Republic of Singapore.

DESCRIPTION

1. Purpose

This subroutine finds a local minimizer of a nonlinear function of n variables $f(x)$ where $x = (x_1, \ldots, x_n)$, $n \geq 1$ can be any real numbers. This algorithm is meant to supersede the algorithm documented in [11].

2. Methods

The subroutine incorporates two nonlinear optimization methods, a conjugate gradient algorithm and a variable metric algorithm, with the choice of method left to the user.

The conjugate gradient algorithm is the Beale restarted memoryless variable metric algorithm documented in Shanno [7]. This method requires approximately $7n$ double-precision words of working storage to be provided by the user. The variable metric method is the BFGS algorithm with initial scaling documented in Shanno and Phua [10], and required approximately $n^2/2 + 11n/2$ double-precision words of working storage.

Whichever method is chosen, the same linear search technique is used for both methods, with two differences. The basic linear search uses Davidon's cubic interpolation to find a step length α, which satisfies

$$f(x + \alpha d) < f(x) + \alpha d'g(x)\ 0.0001, \tag{1}$$

where d is the chosen search direction, $g(x) = \nabla f(x)$, the gradient of f at x, and $d'g(x)$, the directional derivative of $f(x)$ at x along d, is always a negative number. In addition, α must satisfy

$$|d'g(x + \alpha d)/d'g(x)| < 0.9. \tag{2}$$

The convergence of the BFGS variable metric algorithm under the conditions (1) and (2) has been studied by Powell [6], while the convergence of the conjugate gradient method has been studied by Shanno [8].

The major difference between the two methods insofar as the linear search is concerned is that if the first trial α satisfies (1) and (2), it is accepted if a variable metric method is used, but at least two trial α's are required before accepting an α satisfying (1) and (2) for the conjugate gradient method. Reasons for requiring this are detailed in Shanno [9].

The second difference between the two methods is that for a variable metric method, $\alpha = 1$ is always the initial α tried, while for the conjugate gradient method, $\alpha = 1$ is tried only for restart iterations, whereas for nonrestart iterations the initial α at step $k + 1$, denoted by α_{k+1}, is chosen to be $\alpha_{k+1} = \alpha_k d'_k g_k / d'_{k+1} g_{k+1}$, where d_k and g_k and d_{k+1} and g_{k+1} are the search vectors and gradients at the kth and k + 1st points, respectively.

The linear search contains safeguards to ensure that the search procedure cannot become stuck or attempt to move past a local maximum to a more distant local minimum.

Convergence is determined to have occurred when $\|g\| < \epsilon \max(1, \|x\|)$, where $\|\cdot\|$ is the Euclidean norm and ϵ is user supplied.

3. Test Results

Table I contains test results for a variety of test problems for both the BFGS and conjugate gradient method. The test functions are Wood's function for the listed initial estimates, the extended Rosenbrock function documented in [7], Brodlie's [1] variable-dimensioned Watson function, Oren's [5] power function, Powell's

Table I

	BFGS		Conjugate Gradient	
	ITER	IFUN	ITER	IFUN
WOOD (n = 4)				
−3, −1, −3, −1	36	43	48	106
−3, 1, −3, 1	91	114	90	210
−1.2, 1, −1.2, 1	87	107	77	181
−1.2, 1, 1.2, 1	48	57	46	100
EROSEN				
−1.2, 1, 1, 1, 1 (n = 5)	117	150	132	278
−1, . . . , −1 (n = 10)	674	893	946	1940
WATSON				
0, . . . , 0 (n = 5)	37	39	34	69
0, . . . , 0 (n = 10)	92	95	179	360
POWER				
1, . . . , 1 (n = 20)	280	281	16	33
1, . . . , 1 (n = 50)	539	540	30	61
POWELL (n = 4)				
−3, −1, 0, 1	48	49	28	57
TRIG				
(n = 5)	20	22	20	41
(n = 10)	35	37	44	89
(n = 15)	57	59	100	201
MANCINO				
(n = 10)	9	10	12	28
(n = 20)	10	14	14	33
(n = 30)	11	17	18	49
BOUNDARY VALUE				
(n = 10)	28	30	25	51
(n = 20)	56	58	48	97
(n = 30)	86	88	121	243
BROYDEN-TOINT				
−1, . . . , −1 (n = 10)	27	28	23	47
−1, . . . , −1 (n = 20)	36	37	36	73
−1, . . . , −1 (n = 30)	47	48	46	93

four-dimensioned function [6], Fletcher and Powell's trigonometric functions with initial estimates as in [2], and for various dimensions, the Mancino function with initial estimates as documented in [7], Moré, Garbow, and Hillstrom's [4] boundary-value problem, and Toint's variation on Broyden's function [12]. The Wood and Powell functions are documented in [3].

In the table, ITER represents the number of search directions calculated, while IFUN represents the number of function and gradient evaluations that were performed. In all cases, $\epsilon = 10^{-5}$ was used, except for the boundary value problem, where $\epsilon = 10^{-4}$.

With the exception of the power function, on which the BFGS does not perform well for reasons documented in [10], it is clear from Table I that the variable metric method is quite a bit more efficient in terms of function and gradient calls, primarily due to the fact that the first trial α can be accepted, while at least two trial α's per iteration must be tried by the conjugate gradient algorithm.

In terms of execution time, however, the issue is not so clear-cut. On a DEC-10 computer, the Broyden–Toint function with $n = 30$ took 6.49 CPU seconds for the BFGS, while the conjugate gradient method took 5.32 due to the overhead of updating the approximate Hessian at each step. However, for the 30-variable Mancino function, the BFGS took 28.12 CPU seconds, while the conjugate gradient method took 72.31. As one would expect, evaluations of the Mancino function are quite expensive, while the Broyden–Toint function evaluations are quite inexpensive.

Thus for large problems where space limitations do not preclude using the BFGS algorithm, users are urged to experiment to determine the most efficient algorithm for a particular problem. For small problems, we recommend the BFGS method, while for very large problems, memory considerations generally mandate using the conjugate gradient algorithm.

ACKNOWLEDGMENT

The authors are deeply indebted to Professor M.J.D. Powell for many useful suggestions concerning the linear search method adopted and for illustrating several adopted methods for coding optimization routines.

REFERENCES

1. Brodlie, K.W. An assessment of two approaches to variable metric methods. *Math. Program. 12* (1977), 334–355.
2. Fletcher, R., and Powell, M.J.D. A rapidly convergent descent method for minimization. *Comput. J. 6* (1963), 163–168.
3. Himmelblau, D.M. *Applied Nonlinear Programming*. McGraw-Hill, New York, 1972.
4. Moré, J J., Garbow, B.S., and Hillstrom, K.E. Testing unconstrained minimization software TM-324, Appl. Math. Div , Argonne National Lab., Argonne, Ill., 1978.
5. Oren, S.S. Self-scaling variable metric algorithm. Part II. *Manage. Sci 20* (1974), 863–874.
6. Powell, M.J.D. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In *Nonlinear Programming*, R.W. Cottle and C.E. Lemke, Eds., American Mathematical Soc , Providence, R.I., 1976
7. Shanno, D.F. Conjugate gradient methods with inexact searches. *Math. Oper. Res. 3* (1978), 244–256.
8. Shanno, D.F. On the convergence of a new conjugate gradient method. *SIAM. J Numer Anal 15* (1978), 1247–1257.

9. SHANNO, D.F. Quadratic termination of conjugate gradient methods. In *Extrenal Methods and Systems Analysis*, A.V. Fiacco and K.O. Kortanek, Eds, Springer-Verlag, New York, 1980, pp. 433-441.
10. SHANNO, D.F., AND PHUA, K.H. Matrix conditioning and nonlinear optimization. *Math. Program 14* (1978), 149-160.
11. SHANNO, D.F., AND PHUA, K.H. Algorithm 500. Minimization of unconstrained multivariate functions. *ACM Trans. Math. Softw. 2*, 1 (1976), 87-94.
12. TOINT, PH.L. Some numerical results using a sparse matrix updating formula in unconstrained optimization. *Math. Comput. 32* (1978), 839-852.

ALGORITHM

[A part of the listing is printed here. The complete listing is available from the ACM Algorithms Distribution Service (see page 627 for order form) or may be found in "Collected Algorithms from ACM."]

```
      SUBROUTINE CONMIN(N,X,F,G,IFUN,ITER,EPS,NFLAG,MXFUN,W,
     1IOUT,MDIM,IDEV,ACC,NMETH)
C
C PURPOSE:      SUBROUTINE CONMIN MINIMIZES AN UNCONSTRAINED NONLINEAR
C               SCALAR VALUED FUNCTION OF A VECTOR VARIABLE X
C               EITHER BY THE BFGS VARIABLE METRIC ALGORITHM OR BY A
C               BEALE RESTARTED CONJUGATE GRADIENT ALGORITHM.
C
C USAGE:        CALL CONMIN(N,X,F,G,IFUN,ITER,EPS,NFLAG,MXFUN,W,
C               IOUT,MDIM,IDEV,ACC,NMETH)
C
C PARAMETERS: N       THE NUMBER OF VARIABLES IN THE FUNCTION TO
C                     BE MINIMIZED.
C             X       THE VECTOR CONTAINING THE CURRENT ESTIMATE TO
C                     THE MINIMIZER. ON ENTRY TO CONMIN,X MUST CONTAIN
C                     AN INITIAL ESTIMATE SUPPLIED BY THE USER.
C                     ON EXITING,X WILL HOLD THE BEST ESTIMATE TO THE
C                     MINIMIZER OBTAINED BY CONMIN. X MUST BE DOUBLE
C                     PRECISIONED AND DIMENSIONED N.
C             F       ON EXITING FROM CONMIN,F WILL CONTAIN THE LOWEST
C                     VALUE OF THE OBJECT FUNCTION OBTAINED.
C                     F IS DOUBLE PRECISIONED.
C             G       ON EXITING FROM CONMIN,G WILL CONTAIN THE
C                     ELEMENTS OF THE GRADIENT OF F EVALUATED AT THE
C                     POINT CONTAINED IN X. G MUST BE DOUBLE
C                     PRECISIONED AND DIMENSIONED N.
C             IFUN    UPON EXITING FROM CONMIN,IFUN CONTAINS THE
C                     NUMBER OF TIMES THE FUNCTION AND GRADIENT
C                     HAVE BEEN EVALUATED.
C             ITER    UPON EXITING FROM CONMIN,ITER CONTAINS THE
C                     TOTAL NUMBER OF SEARCH DIRECTIONS CALCULATED
C                     TO OBTAIN THE CURRENT ESTIMATE TO THE MINIZER.
C             EPS     EPS IS THE USER SUPPLIED CONVERGENCE PARAMETER.
C                     CONVERGENCE OCCURS WHEN THE NORM OF THE GRADIENT
C                     IS LESS THAN OR EQUAL TO EPS TIMES THE MAXIMUM
C                     OF ONE AND THE NORM OF THE VECTOR X. EPS
C                     MUST BE DOUBLE PRECISIONED.
C             NFLAG   UPON EXITING FROM CONMIN,NFLAG STATES WHICH
C                     CONDITION CAUSED THE EXIT.
C                     IF NFLAG=Ø, THE ALGORITHM HAS CONVERGED.
```

```
C                     IF NFLAG=1, THE MAXIMUM NUMBER OF FUNCTION
C                        EVALUATIONS HAVE BEEN USED.
C                     IF NFLAG=2, THE LINEAR SEARCH HAS FAILED TO
C                        IMPROVE THE FUNCTION VALUE. THIS IS THE
C                        USUAL EXIT IF EITHER THE FUNCTION OR THE
C                        GRADIENT IS INCORRECTLY CODED.
C                     IF NFLAG=3, THE SEARCH VECTOR WAS NOT
C                        A DESCENT DIRECTION. THIS CAN ONLY BE CAUSED
C                        BY ROUNDOFF,AND MAY SUGGEST THAT THE
C                        CONVERGENCE CRITERION IS TOO STRICT.
C            MXFUN    MXFUN IS THE USER SUPPLIED MAXIMUM NUMBER OF
C                     FUNCTION AND GRADIENT CALLS THAT CONMIN WILL
C                     BE ALLOWED TO MAKE.
C            W        W IS A VECTOR OF WORKING STORAGE.IF NMETH=Ø,
C                     W MUST BE DIMENSIONED 5*N+2. IF NMETH=1,
C                     W MUST BE DIMENSIONED N*(N+7)/2. IN BOTH CASES,
C                     W MUST BE DOUBLE PRECISIONED.
C            IOUT     IOUT IS A USER  SUPPLIED OUTPUT PARAMETER.
C                     IF IOUT = Ø, THERE IS NO PRINTED OUTPUT FROM
C                     CONMIN. IF IOUT J Ø,THE VALUE OF F AND THE
C                     NORM OF THE GRADIENT SQUARED,AS WELL AS ITER
C                     AND IFUN,ARE WRITTEN EVERY IOUT ITERATIONS.
C            MDIM     MDIM IS THE USER SUPPLIED DIMENSION OF THE
C                     VECTOR W. IF NMETH=Ø,MDIM=5*N+2. IF NMETH=1,
C                     MDIM=N*(N+7)/2.
C            IDEV     IDEV IS THE USER SUPPLIED NUMBER OF THE OUTPUT
C                     DEVICE ON WHICH OUTPUT IS TO BE WRITTEN WHEN
C                     IOUTJØ.
C            ACC      ACC IS A USER SUPPLIED ESTIMATE OF MACHINE
C                     ACCURACY. A LINEAR SEARCH IS UNSUCCESSFULLY
C                     TERMINATED WHEN THE NORM OF THE STEP SIZE
C                     BECOMES SMALLER THAN ACC. IN PRACTICE,
C                     ACC=1Ø.D-2Ø HAS PROVED SATISFACTORY. ACC IS
C                     DOUBLE PRECISIONED.
C            NMETH    NMETH IS THE USER SUPPLIED VARIABLE WHICH
C                     CHOOSES THE METHOD OF OPTIMIZATION. IF
C                     NMETH=Ø,A CONJUGATE GRADIENT METHOD IS
C                     USED. IF NMETH=1, THE BFGS METHOD IS USED.
C
C REMARKS:   IN ADDITION TO THE SPECIFIED VALUES IN THE ABOVE
C            ARGUMENT LIST, THE USER MUST SUPPLY A SUBROUTINE
C            CALCFG WHICH CALCULATES THE FUNCTION AND GRADIENT AT
C            X AND PLACES THEM IN F AND G(1),...,G(N) RESPECTIVELY.
C            THE SUBROUTINE MUST HAVE THE FORM:
C                     SUBROUTINE CALCFG(N,X,F,G)
C                     DOUBLE PRECISION X(N),G(N),F
C
C            AN EXAMPLE SUBROUTINE FOR THE ROSENBROCK FUNCTION IS:
C
C                     SUBROUTINE CALCFG(N,X,F,G)
C                     DOUBLE PRECISION X(N),G(N),F,T1,T2
C                     T1=X(2)-X(1)*X(1)
C                     T2=1.Ø-X(1)
C                     F=1ØØ.Ø*T1*T1+T2*T2
C                     G(1)=-4ØØ.Ø*T1*X(1)-2.Ø*T2
C                     G(2)=2ØØ.Ø*T1
C                     RETURN
C                     END
```