E04HCF – NAG Fortran Library Routine Document

Note. Before using this routine, please read the Users' Note for your implementation to check the interpretation of bold italicised terms and other implementation-dependent details.

1 Purpose

E04HCF checks that a user-supplied routine for evaluating an objective function and its first derivatives produces derivative values which are consistent with the function values calculated.

2 Specification

```
SUBROUTINE E04HCF(N, FUNCT, X, F, G, IW, LIW, W, LW, IFAIL)
INTEGER            N, IW(LIW), LIW, LW, IFAIL
real               X(N), F, G(N), W(LW)
EXTERNAL           FUNCT
```

3 Description

Routines for minimizing a function of several variables may require the user to supply a subroutine to evaluate the objective function $F(x_1, x_2, \ldots, x_n)$ and its first derivatives. E04HCF is designed to check the derivatives calculated by such user-supplied routines. As well as the routine to be checked (FUNCT), the user must supply a point $x = (x_1, x_2, \ldots, x_n)^T$ at which the check will be made. Note that E04HCF checks routines of the form required for E04KDF and E04LBF.

E04HCF first calls FUNCT to evaluate F and its first derivatives $g_j = \frac{\partial F}{\partial x_j}$, for $j = 1, 2, \ldots, n$ at x. The components of the user-supplied derivatives along two orthogonal directions (defined by unit vectors p_1 and p_2, say) are then calculated; these will be $g^T p_1$ and $g^T p_2$ respectively. The same components are also estimated by finite differences, giving quantities

$$v_k = \frac{F(x + hp_k) - F(x)}{h}, \qquad k = 1, 2$$

where h is a small positive scalar. If the relative difference between v_1 and $g^T p_1$ or between v_2 and $g^T p_2$ is judged too large, an error indicator is set.

4 References

None.

5 Parameters

1: N — INTEGER *Input*

On entry: the number n of independent variables in the objective function.

Constraint: N ≥ 1.

2: FUNCT — SUBROUTINE, supplied by the user. *External Procedure*

FUNCT must evaluate the function and its first derivatives at a given point. (The minimization routines mentioned in Section 3 give the user the option of resetting a parameter of FUNCT to cause the minimization process to terminate immediately. E04HCF will also terminate immediately, without finishing the checking process, if the parameter in question is reset.)

Its specification is:

```
SUBROUTINE FUNCT(IFLAG, N, XC, FC, GC, IW, LIW, W, LW)
INTEGER          IFLAG, N, IW(LIW), LIW, LW
real             XC(N), FC, GC(N), W(LW)
```

1: IFLAG — INTEGER *Input/Output*

On entry: IFLAG will be set to 2.

On exit: if the user resets IFLAG to a negative number in FUNCT and returns control to E04HCF, E04HCF will terminate immediately with IFAIL set to the user's setting of IFLAG.

2: N — INTEGER *Input*

On entry: the number n of variables.

3: XC(N) — ***real*** array *Input*

On entry: the point x at which F and its derivatives are required.

4: FC — ***real*** *Output*

On exit: unless FUNCT resets IFLAG, FC must be set to the value of the function F at the current point x.

5: GC(N) — ***real*** array *Output*

On exit: unless FUNCT resets IFLAG, GC(j) must be set to the value of the first derivative $\frac{\partial F}{\partial x_j}$ at the point x, for $j = 1, 2, \ldots, n$.

6: IW(LIW) — INTEGER array *Workspace*
7: LIW — INTEGER *Input*
8: W(LW) — ***real*** array *Workspace*
9: LW — INTEGER *Input*

These parameters are present so that FUNCT will be of the form required by the minimization routines mentioned in Section 3. FUNCT is called with E04HCF's parameters IW, LIW, W, LW as these parameters. If the advice given in the minimization routine documents is being followed, the user will have no reason to examine or change any elements of IW or W. In any case, FUNCT **must not change** the first 3 × N elements of W.

FUNCT must be declared as EXTERNAL in the (sub)program from which E04HCF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

3: X(N) — ***real*** array *Input*

On entry: X(j), for $j = 1, 2, \ldots, n$ must be set to the co-ordinates of a suitable point at which to check the derivatives calculated by FUNCT. 'Obvious' settings, such as 0.0 or 1.0, should not be used since, at such particular points, incorrect terms may take correct values (particularly zero), so that errors could go undetected. Similarly, it is preferable that no two elements of X should be the same.

4: F — ***real*** *Output*

On exit: unless the user sets IFLAG negative in the first call of FUNCT, F contains the value of the objective function $F(x)$ at the point given by the user in X.

5: G(N) — ***real*** array *Output*

On exit: unless the user sets IFLAG negative in the first call of FUNCT, G(j) contains the value of the derivative $\frac{\partial F}{\partial x_j}$ at the point given in X, as calculated by FUNCT, for $j = 1, 2, \ldots, n$.

6: IW(LIW) — INTEGER array *Workspace*

This array is in the parameter list so that it can be used by other library routines for passing INTEGER quantities to FUNCT. It is not examined or changed by E04HCF. The general user must provide an array IW but is advised not to use it.

7: LIW — INTEGER *Input*

On entry: the length of the array IW as declared in the (sub)program from which E04HCF is called.

Constraint: LIW ≥ 1.

8: W(LW) — ***real*** array *Workspace*

9: LW — INTEGER *Input*

On entry: the length of the array W as declared in the (sub)program from which E04HCF is called.

Constraint: LW $\geq 3 \times$ N.

10: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL $= 0$ unless the routine detects an error or gives a warning (see Section 6).

For this routine, because the values of output parameters may be useful even if IFAIL $\neq 0$ on exit, users are recommended to set IFAIL to -1 before entry. **It is then essential to test the value of IFAIL on exit**. To suppress the output of an error message when soft failure occurs, set IFAIL to 1.

6 Error Indicators and Warnings

Errors or warnings specified by the routine:

IFAIL < 0

A negative value of IFAIL indicates an exit from E04HCF because the user has set IFLAG negative in FUNCT. The setting of IFAIL will be the same as the user's setting of IFLAG. The check on FUNCT will not have been completed.

IFAIL $= 1$

On entry, N < 1,
or LIW < 1,
or LW $< 3 \times$ N.

IFAIL $= 2$

The user should check carefully the derivation and programming of expressions for the derivatives of $F(x)$, because it is very unlikely that FUNCT is calculating them correctly.

7 Accuracy

IFAIL is set to 2 if

$$(v_k - g^T p_k)^2 \geq h \times ((g^T p_k)^2 + 1)$$

for $k = 1$ or 2. (See Section 3 for definitions of the quantities involved.) The scalar h is set equal to $\sqrt{\epsilon}$, where ϵ is the ***machine precision*** as given by X02AJF.

8 Further Comments

The user-supplied routine FUNCT is called 3 times.

Before using E04HCF to check the calculation of first derivatives, the user should be confident that FUNCT is calculating F correctly. The usual way of checking the calculation of the function is to compare values of $F(x)$ calculated by FUNCT at non-trivial points x with values calculated independently. ('Non-trivial' means that, as when setting x before calling E04HCF, co-ordinates such as 0.0 or 1.0 should be avoided.)

E04HCF only checks the derivatives calculated by a user-supplied routine when IFLAG = 2. So, if FUNCT is intended for use in conjunction with a minimization routine which may set IFLAG to 1, the user must check that, for given settings of the XC(j), FUNCT produces the same values for the GC(j) when IFLAG is set to 1 as when IFLAG is set to 2.

9 Example

Suppose that it is intended to use E04KDF to minimize

$$F = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4.$$

The following program could be used to check the first derivatives calculated by the routine FUNCT. (The tests of whether IFLAG = 0 or 1 in FUNCT are present ready for when FUNCT is called by E04KDF. E04HCF will always call FUNCT with IFLAG set to 2.)

9.1 Program Text

Note. The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     E04HCF Example Program Text.
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          N, LIW, LW
      PARAMETER        (N=4,LIW=1,LW=3*N)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             F
      INTEGER          IFAIL, J
*     .. Local Arrays ..
      real             G(N), W(LW), X(N)
      INTEGER          IW(LIW)
*     .. External Subroutines ..
      EXTERNAL         E04HCF, FUNCT
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E04HCF Example Program Results'
*     Set up an arbitrary point at which to check the 1st derivatives
      X(1) = 1.46e0
      X(2) = -0.82e0
      X(3) = 0.57e0
      X(4) = 1.21e0
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'The test point is'
      WRITE (NOUT,99999) (X(J),J=1,N)
      IFAIL = 1
*
      CALL E04HCF(N,FUNCT,X,F,G,IW,LIW,W,LW,IFAIL)
*
      WRITE (NOUT,*)
      IF (IFAIL.LT.0) THEN
         WRITE (NOUT,99998) 'IFLAG was set to ', IFAIL, 'in FUNCT'
      ELSE IF (IFAIL.EQ.1) THEN
         WRITE (NOUT,*) 'A parameter is outside its expected range'
      ELSE
         IF (IFAIL.EQ.0) THEN
            WRITE (NOUT,*)
     +        '1st derivatives are consistent with function values'
         ELSE
            WRITE (NOUT,*)
     +        'Probable error in calculation of 1st derivatives'
         END IF
         WRITE (NOUT,*)
         WRITE (NOUT,99997)
     +     'At the test point, FUNCT gives the function value', F
         WRITE (NOUT,*) 'and the 1st derivatives'
         WRITE (NOUT,99996) (G(J),J=1,N)
      END IF
      STOP
*
99999 FORMAT (1X,4F10.4)
99998 FORMAT (1X,A,I3,A)
99997 FORMAT (1X,A,1P,e12.4)
99996 FORMAT (1X,1P,4e12.3)
      END
*
      SUBROUTINE FUNCT(IFLAG,N,XC,FC,GC,IW,LIW,W,LW)
*     Routine to evaluate objective function and its 1st derivatives.
*     .. Scalar Arguments ..
      real              FC
      INTEGER           IFLAG, LIW, LW, N
*     .. Array Arguments ..
      real              GC(N), W(LW), XC(N)
      INTEGER           IW(LIW)
*     .. Executable Statements ..
      IF (IFLAG.NE.1) THEN
         FC = (XC(1)+10.0e0*XC(2))**2 + 5.0e0*(XC(3)-XC(4))**2 + (XC(2)
     +        -2.0e0*XC(3))**4 + 10.0e0*(XC(1)-XC(4))**4
      END IF
      IF (IFLAG.NE.0) THEN
         GC(1) = 2.0e0*(XC(1)+10.0e0*XC(2)) + 40.0e0*(XC(1)-XC(4))**3
         GC(2) = 20.0e0*(XC(1)+10.0e0*XC(2)) + 4.0e0*(XC(2)-2.0e0*XC(3))
     +           **3
         GC(3) = 10.0e0*(XC(3)-XC(4)) - 8.0e0*(XC(2)-2.0e0*XC(3))**3
         GC(4) = 10.0e0*(XC(4)-XC(3)) - 40.0e0*(XC(1)-XC(4))**3
      END IF
      RETURN
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 E04HCF Example Program Results

 The test point is
     1.4600   -0.8200    0.5700    1.2100

 1st derivatives are consistent with function values

 At the test point, FUNCT gives the function value  6.2273E+01
 and the 1st derivatives
   -1.285E+01  -1.649E+02   5.384E+01   5.775E+00
```