NAG Fortran Library Routine Document

E04YBF

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

E04YBF checks that a user-supplied routine for evaluating the second derivative term of the Hessian matrix of a sum of squares is consistent with a user-supplied routine for calculating the corresponding first derivatives.

2 Specification

```
SUBROUTINE E04YBF(M, N, LSQFUN, LSQHES, X, FVEC, FJAC, LJ, B, LB, IW,
1                 LIW, W, LW, IFAIL)
INTEGER           M, N, LJ, LB, IW(LIW), LIW, LW, IFAIL
real              X(N), FVEC(M), FJAC(LJ,N), B(LB), W(LW)
EXTERNAL          LSQFUN, LSQHES
```

3 Description

Routines for minimizing a sum of squares of m nonlinear functions (or 'residuals'), $f_i(x_1, x_2, \ldots, x_n)$, for $i = 1, 2, \ldots, m$; $m \geq n$, may require the user to supply a subroutine to evaluate the quantities

$$b_{jk} = \sum_{i=1}^{m} f_i \frac{\partial^2 f_i}{\partial x_j \partial x_k}$$

for $j = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, j$. E04YBF is designed to check the b_{jk} calculated by such user-supplied routines. As well as the routine to be checked (LSQHES), the user must supply a routine (LSQFUN) to evaluate the f_i and their first derivatives, and a point $x = (x_1, x_2, \ldots, x_n)^T$ at which the checks will be made. Note that E04YBF checks routines of the form required by E04HEF. E04YBF is essentially identical to CHKLSH in the NPL Algorithms Library.

E04YBF first calls LSQFUN and LSQHES to evaluate the first derivatives and the b_{jk} at x. Let J denote the m by n matrix of first derivatives of the residuals. The Hessian matrix of the sum of squares,

$$G = J^T J + B,$$

is calculated and projected onto two orthogonal vectors y and z to give the scalars $y^T G y$ and $z^T G z$ respectively. The same projections of the Hessian matrix are also estimated by finite differences, giving

$$p = (y^T g(x + hy) - y^T g(x))/h \quad \text{and}$$
$$q = (z^T g(x + hz) - z^T g(x))/h$$

respectively, where $g(\ )$ denotes the gradient vector of the sum of squares at the point in brackets and h is a small positive scalar. If the relative difference between p and $y^T G y$ or between q and $z^T G z$ is judged too large, an error indicator is set.

4 References

None.

5 Parameters

1: M – INTEGER *Input*
2: N – INTEGER *Input*

On entry: the number m of residuals, $f_i(x)$, and the number n of variables, x_j.

Constraint: $1 \leq \mathrm{N} \leq \mathrm{M}$.

3: LSQFUN – SUBROUTINE, supplied by the user. *External Procedure*

LSQFUN must calculate the vector of values $f_i(x)$ and their first derivatives $\dfrac{\partial f_i}{\partial x_j}$ at any point x. (E04HEF gives the user the option of resetting a parameter of LSQFUN to cause the minimization process to terminate immediately. E04YBF will also terminate immediately, without finishing the checking process, if the parameter in question is reset.)

Its specification is:

```
SUBROUTINE LSQFUN(IFLAG, M, N, XC, FVECC, FJACC, LJC, IW, LIW, W,
1                 LW)
INTEGER           IFLAG, M, N, LJC, IW(LIW), LIW, LW
real              XC(N), FVECC(M), FJACC(LJC,N), W(LW)
```

1: IFLAG – INTEGER *Input/Output*

On entry: to LSQFUN, IFLAG will be set to 2.

On exit: if the user resets IFLAG to some negative number in LSQFUN and returns control to E04YBF, the routine will terminate immediately with IFAIL set to the user's setting of IFLAG.

2: M – INTEGER *Input*
3: N – INTEGER *Input*

On entry: the numbers m and n of residuals and variables, respectively.

4: XC(N) – ***real*** array *Input*

On entry: the point x at which the values of the f_i and the $\dfrac{\partial f_i}{\partial x_j}$ are required.

5: FVECC(M) – ***real*** array *Output*

On exit: unless IFLAG is reset to a negative number, FVECC(i) must contain the value of f_i at the point x, for $i = 1, 2, \ldots, m$.

6: FJACC(LJC,N) – ***real*** array *Output*

On exit: unless IFLAG is reset to a negative number, FJACC(i, j) must contain the value of $\dfrac{\partial f_i}{\partial x_j}$ at the point x, for $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$.

7: LJC – INTEGER *Input*

On entry: the first dimension of the array FJACC as declared in the (sub)program from which E04YBF is called.

8: IW(LIW) – INTEGER array *Workspace*
9: LIW – INTEGER *Input*
10: W(LW) – ***real*** array *Workspace*
11: LW – INTEGER *Input*

These parameters are present so that LSQFUN will be of the form required by E04HEF. LSQFUN is called with E04YBF's parameters IW, LIW, W, LW as these parameters. If the recommendation in E04HEF is followed, the user will have no reason to examine or change the elements of IW or W. In any case, LSQFUN **must** not change the first $5 \times \mathrm{N} + \mathrm{M} + \mathrm{M} \times \mathrm{N} + \mathrm{N} \times (\mathrm{N}-1)/2$ (or $6 + 2 \times \mathrm{M}$ if $\mathrm{N} = 1$) elements of W.

LSQFUN must be declared as EXTERNAL in the (sub)program from which E04YBF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

Note: E04YAF should be used to check the first derivatives calculated by LSQFUN before E04YBF is used to check the b_{jk} since E04YBF assumes that the first derivatives are correct.

4: LSQHES – SUBROUTINE, supplied by the user. *External Procedure*

LSQHES must calculate the elements of the symmetric matrix

$$B(x) = \sum_{i=1}^{m} f_i(x) G_i(x),$$

at any point x, where $G_i(x)$ is the Hessian matrix of $f_i(x)$. (As with LSQFUN, a parameter can be set to cause immediate termination.)

Its specification is:

```
SUBROUTINE LSQHES(IFLAG, M, N, FVECC, XC, B, LB, IW, LIW, W, LW)
INTEGER           IFLAG, M, N, LB, IW(LIW), LIW, LW
real              FVECC(M), XC(N), B(LB), W(LW)
```

1: IFLAG – INTEGER *Input/Output*

On entry: IFLAG is set to a non-negative number.

On exit: if LSQHES resets IFLAG to some negative number, E04YBF will terminate immediately, with IFAIL set to the user's setting of IFLAG.

2: M – INTEGER *Input*
3: N – INTEGER *Input*

On entry: the numbers m and n of residuals and variables, respectively.

4: FVECC(M) – ***real*** array *Input*

On entry: the value of the residual f_i at the point x, for $i = 1, 2, \ldots, m$, so that the values of the f_i can be used in the calculation of the elements of B.

5: XC(N) – ***real*** array *Input*

On entry: the point x at which the elements of B are to be evaluated.

6: B(LB) – ***real*** array *Output*

On exit: unless IFLAG is reset to a negative number B must contain the lower triangle of the matrix $B(x)$, evaluated at the point in XC, stored by rows. (The upper triangle is not needed because the matrix is symmetric.) More precisely, $\mathrm{B}(j(j-1)/2 + k)$ must contain $\sum_{i=1}^{m} f_i \dfrac{\partial^2 f_i}{\partial x_j \partial x_k}$ evaluated at the point x, for $j = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, j$.

7: LB – INTEGER *Input*

On entry: LB gives the length of the array B.

8: IW(LIW) – INTEGER array *Workspace*
9: LIW – INTEGER *Input*
10: W(LW) – ***real*** array *Workspace*
11: LW – INTEGER *Input*

As in LSQFUN, these parameters correspond to the parameters IW, LIW, W, LW of E04YBF. LSQHES **must not change** the first $5 \times \mathrm{N} + \mathrm{M} \times \mathrm{N} + \mathrm{N} \times (\mathrm{N} - 1)/2$ (or $6 + 2 \times \mathrm{M}$ if $\mathrm{N} = 1$) elements of W.

LSQHES must be declared as EXTERNAL in the (sub)program from which E04YBF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

5: X(N) – ***real*** array *Input*

On entry: X(j) ($j = 1, 2, \ldots, n$) must be set to the co-ordinates of a suitable point at which to check the b_{jk} calculated by LSQHES. 'Obvious' settings, such as 0 or 1, should not be used since, at such particular points, incorrect terms may take correct values (particularly zero), so that errors could go undetected. For a similar reason, it is preferable that no two elements of X should have the same value.

6: FVEC(M) – ***real*** array *Output*

On exit: unless the user sets IFLAG negative in the first call of LSQFUN, FVEC(i) contains the value of f_i at the point given by the user in X, for $i = 1, 2, \ldots, m$.

7: FJAC(LJ,N) – ***real*** array *Output*

On exit: unless the user sets IFLAG negative in the first call of LSQFUN, FJAC(i, j) contains the value of the first derivative $\dfrac{\partial f_i}{\partial x_j}$ at the point given in X, as calculated by LSQFUN, for $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$.

8: LJ – INTEGER *Input*

On entry: the first dimension of the array FJAC as declared in the (sub)program from which E04YBF is called.

Constraint: LJ $\geq$ M.

9: B(LB) – ***real*** array *Output*

On exit: unless the user sets IFLAG negative in LSQHES, B($j \times (j - 1)/2 + k$) contains the value of b_{jk} at the point given in X as calculated by LSQHES, for $j = 1, 2, \ldots, n$; $k = 1, 2, \ldots, j$.

10: LB – INTEGER *Input*

On entry: the dimension of the array B as declared in the (sub)program from which E04YBF is called.

Constraint: LB $\geq (\mathrm{N} + 1) \times \mathrm{N}/2$.

11: IW(LIW) – INTEGER array *Workspace*

This array appears in the parameter list purely so that, if E04YBF is called by another library routine, the library routine can pass quantities to LSQFUN and LSQHES via IW. IW is not examined or changed by E04YBF. The general user must provide an array IW, but is advised not to use it.

12: LIW – INTEGER *Input*

On entry: the actual length of IW as declared in the (sub)program from which E04YBF is called.

Constraint: LIW ≥ 1.

13: W(LW) – ***real*** array *Workspace*

14: LW – INTEGER *Input*

On entry: the actual length of W as declared in the (sub)program from which E04YBF is called.

Constraints:

$$\text{LW} \geq 5 \times \text{N} + \text{M} + \text{M} \times \text{N} + \text{N} \times (\text{N} - 1)/2, \text{ if N} > 1$$
$$\text{LW} \geq 6 + 2 \times \text{M}, \text{ if N} = 1.$$

15: 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 (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, because for this routine the values of the output parameters may be useful even if IFAIL $\neq 0$ on exit, the recommended value is -1. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL < 0

A negative value of IFAIL indicates an exit from E04YBF because the user has set IFLAG negative in LSQFUN or LSQHES. The setting of IFAIL will be the same as the user's setting of IFLAG. The check on LSQHES will not have been completed.

IFAIL $= 1$

On entry, $\text{M} < \text{N}$,
or $\text{N} < 1$,
or $\text{LJ} < \text{M}$,
or $\text{LB} < (\text{N} + 1) \times \text{N}/2$,
or $\text{LIW} < 1$,
or $\text{LW} < 5 \times \text{N} + \text{M} + \text{M} \times \text{N} + \text{N} \times (\text{N} - 1)/2$, if $\text{N} > 1$,
or $\text{LW} < 6 + 2 \times \text{M}$, if $\text{N} = 1$.

IFAIL $= 2$

The user should check carefully the derivation and programming of expressions for the b_{jk}, because it is very unlikely that LSQHES is calculating them correctly.

7 Accuracy

IFAIL is set to 2 if

$$|y^T G y - p| \geq \sqrt{h}(|y^T G y| + 1.0) \quad \text{or}$$
$$|z^T G z - q| \geq \sqrt{h}(|z^T G z| + 1.0)$$

where h is set equal to $\sqrt{\epsilon}$ (ϵ being the ***machine precision*** as given by X02AJF) and other quantities are defined as in Section 3.

8 Further Comments

E04YBF calls LSQHES once and LSQFUN three times.

9 Example

Suppose that it is intended to use E04HEF to find least-squares estimates of x_1, x_2 and x_3 in the model

$$y = x_1 + \frac{t_1}{x_2 t_2 + x_3 t_3}$$

using the 15 sets of data given in the following table.

y	t_1	t_2	t_3
0.14	1.0	15.0	1.0
0.18	2.0	14.0	2.0
0.22	3.0	13.0	3.0
0.25	4.0	12.0	4.0
0.29	5.0	11.0	5.0
0.32	6.0	10.0	6.0
0.35	7.0	9.0	7.0
0.39	8.0	8.0	8.0
0.37	9.0	7.0	7.0
0.58	10.0	6.0	6.0
0.73	11.0	5.0	5.0
0.96	12.0	4.0	4.0
1.34	13.0	3.0	3.0
2.10	14.0	2.0	2.0
4.39	15.0	1.0	1.0

The following program could be used to check the b_{jk} calculated by the routine LSQHES required. (The call of E04YBF is preceded by a call of E04YAF to check the routine LSQFUN which calculates the first derivatives.)

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.

```
*     E04YBF Example Program Text.
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          MDEC, NDEC, LJ, LB, LIW, LW
      PARAMETER        (MDEC=15,NDEC=3,LJ=MDEC,LB=NDEC*(NDEC+1)/2,LIW=1,
     +                 LW=5*NDEC+MDEC+MDEC*NDEC+NDEC*(NDEC-1)/2)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Arrays in Common ..
      real             T(MDEC,NDEC), Y(MDEC)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, K, M, N
*     .. Local Arrays ..
      real             B(LB), FJAC(LJ,NDEC), FVEC(MDEC), W(LW), X(NDEC)
      INTEGER          IW(LIW)
*     .. External Subroutines ..
      EXTERNAL         E04YAF, E04YBF, LSQFUN, LSQHES
*     .. Common blocks ..
      COMMON           Y, T
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E04YBF Example Program Results'
```

```
*     Skip heading in data file
      READ (NIN,*)
      M = MDEC
      N = NDEC
*     Observations of TJ (J = 1, 2, 3) are held in T(I, J)
*     (I = 1, 2, . . . , 15)
      DO 20 I = 1, M
         READ (NIN,*) Y(I), (T(I,J),J=1,N)
   20 CONTINUE
*     Set up an arbitrary point at which to check the derivatives
      X(1) = 0.19e0
      X(2) = -1.34e0
      X(3) = 0.88e0
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'The test point is'
      WRITE (NOUT,99999) (X(J),J=1,N)
*     Check the 1st derivatives
      IFAIL = 0
*
      CALL E04YAF(M,N,LSQFUN,X,FVEC,FJAC,LJ,IW,LIW,W,LW,IFAIL)
*
*     Check the evaluation of B
      IFAIL = 1
*
      CALL E04YBF(M,N,LSQFUN,LSQHES,X,FVEC,FJAC,LJ,B,LB,IW,LIW,W,LW,
     +            IFAIL)
*
      WRITE (NOUT,*)
      IF (IFAIL.LT.0) THEN
         WRITE (NOUT,99998) 'IFLAG was set to ', IFAIL,
     +     'in LSQFUN or LSQHES'
      ELSE IF (IFAIL.EQ.1) THEN
         WRITE (NOUT,*) 'A parameter is outside its expected range'
      ELSE
         IF (IFAIL.EQ.0) THEN
            WRITE (NOUT,*)
     +        'The matrix B is consistent with 1st derivatives'
         ELSE IF (IFAIL.EQ.2) THEN
            WRITE (NOUT,*)
     +        'Probable error in calculation of the matrix B'
         END IF
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'At the test point, LSQFUN gives'
         WRITE (NOUT,*)
         WRITE (NOUT,*)
     +     '      Residuals                    1st derivatives'
         WRITE (NOUT,99997) (FVEC(I),(FJAC(I,J),J=1,N),I=1,M)
         WRITE (NOUT,*)
         WRITE (NOUT,*)
     +     'and LSQHES gives the lower triangle of the matrix B'
         WRITE (NOUT,*)
         K = 1
         DO 40 I = 1, N
            WRITE (NOUT,99997) (B(J),J=K,K+I-1)
            K = K + I
   40    CONTINUE
      END IF
      STOP
*
99999 FORMAT (1X,4F10.5)
99998 FORMAT (1X,A,I3,A)
99997 FORMAT (1X,1P,4e15.3)
      END
*
      SUBROUTINE LSQFUN(IFLAG,M,N,XC,FVECC,FJACC,LJC,IW,LIW,W,LW)
*     Routine to evaluate the residuals and their 1st derivatives
*     .. Parameters ..
      INTEGER           MDEC, NDEC
      PARAMETER         (MDEC=15,NDEC=3)
*     .. Scalar Arguments ..
      INTEGER           IFLAG, LIW, LJC, LW, M, N
```

```
*     .. Array Arguments ..
      real              FJACC(LJC,N), FVECC(M), W(LW), XC(N)
      INTEGER           IW(LIW)
*     .. Arrays in Common ..
      real              T(MDEC,NDEC), Y(MDEC)
*     .. Local Scalars ..
      real              DENOM, DUMMY
      INTEGER           I
*     .. Common blocks ..
      COMMON            Y, T
*     .. Executable Statements ..
      DO 20 I = 1, M
         DENOM = XC(2)*T(I,2) + XC(3)*T(I,3)
         FVECC(I) = XC(1) + T(I,1)/DENOM - Y(I)
         FJACC(I,1) = 1.0e0
         DUMMY = -1.0e0/(DENOM*DENOM)
         FJACC(I,2) = T(I,1)*T(I,2)*DUMMY
         FJACC(I,3) = T(I,1)*T(I,3)*DUMMY
   20 CONTINUE
      RETURN
      END
*
      SUBROUTINE LSQHES(IFLAG,M,N,FVECC,XC,B,LB,IW,LIW,W,LW)
*     Routine to compute the lower triangle of the matrix B
*     (stored by rows in the array B)
*     .. Parameters ..
      INTEGER           MDEC, NDEC
      PARAMETER         (MDEC=15,NDEC=3)
*     .. Scalar Arguments ..
      INTEGER           IFLAG, LB, LIW, LW, M, N
*     .. Array Arguments ..
      real              B(LB), FVECC(M), W(LW), XC(N)
      INTEGER           IW(LIW)
*     .. Arrays in Common ..
      real              T(MDEC,NDEC), Y(MDEC)
*     .. Local Scalars ..
      real              DUMMY, SUM22, SUM32, SUM33
      INTEGER           I
*     .. Common blocks ..
      COMMON            Y, T
*     .. Executable Statements ..
      B(1) = 0.0e0
      B(2) = 0.0e0
      SUM22 = 0.0e0
      SUM32 = 0.0e0
      SUM33 = 0.0e0
      DO 20 I = 1, M
         DUMMY = 2.0e0*T(I,1)/(XC(2)*T(I,2)+XC(3)*T(I,3))**3
         SUM22 = SUM22 + FVECC(I)*DUMMY*T(I,2)**2
         SUM32 = SUM32 + FVECC(I)*DUMMY*T(I,2)*T(I,3)
         SUM33 = SUM33 + FVECC(I)*DUMMY*T(I,3)**2
   20 CONTINUE
      B(3) = SUM22
      B(4) = 0.0e0
      B(5) = SUM32
      B(6) = SUM33
      RETURN
      END
```

9.2 Program Data

```
E04YBF Example Program Data
 0.14  1.0 15.0  1.0
 0.18  2.0 14.0  2.0
 0.22  3.0 13.0  3.0
 0.25  4.0 12.0  4.0
 0.29  5.0 11.0  5.0
 0.32  6.0 10.0  6.0
 0.35  7.0  9.0  7.0
 0.39  8.0  8.0  8.0
 0.37  9.0  7.0  7.0
 0.58 10.0  6.0  6.0
 0.73 11.0  5.0  5.0
 0.96 12.0  4.0  4.0
 1.34 13.0  3.0  3.0
 2.10 14.0  2.0  2.0
 4.39 15.0  1.0  1.0
```

9.3 Program Results

```
 E04YBF Example Program Results

 The test point is
    0.19000  -1.34000   0.88000

 The matrix B is consistent with 1st derivatives

 At the test point, LSQFUN gives

       Residuals                   1st derivatives
      -2.029E-03      1.000E+00     -4.061E-02     -2.707E-03
      -1.076E-01      1.000E+00     -9.689E-02     -1.384E-02
      -2.330E-01      1.000E+00     -1.785E-01     -4.120E-02
      -3.785E-01      1.000E+00     -3.043E-01     -1.014E-01
      -5.836E-01      1.000E+00     -5.144E-01     -2.338E-01
      -8.689E-01      1.000E+00     -9.100E-01     -5.460E-01
      -1.346E+00      1.000E+00     -1.810E+00     -1.408E+00
      -2.374E+00      1.000E+00     -4.726E+00     -4.726E+00
      -2.975E+00      1.000E+00     -6.076E+00     -6.076E+00
      -4.013E+00      1.000E+00     -7.876E+00     -7.876E+00
      -5.323E+00      1.000E+00     -1.040E+01     -1.040E+01
      -7.292E+00      1.000E+00     -1.418E+01     -1.418E+01
      -1.057E+01      1.000E+00     -2.048E+01     -2.048E+01
      -1.713E+01      1.000E+00     -3.308E+01     -3.308E+01
      -3.681E+01      1.000E+00     -7.089E+01     -7.089E+01

 and LSQHES gives the lower triangle of the matrix B

       0.000E+00
       0.000E+00      1.571E+04
       0.000E+00      1.571E+04      1.571E+04
```