E04YCF – 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

E04YCF returns estimates of elements of the variance-covariance matrix of the estimated regression coefficients for a nonlinear least-squares problem. The estimates are derived from the Jacobian of the function $f(x)$ at the solution.

This routine may be used following any one of the nonlinear least-squares routines E04FCF, E04FYF, E04GBF, E04GYF, E04GDF, E04GZF, E04HEF or E04HYF.

2 Specification

```
SUBROUTINE E04YCF(JOB, M, N, FSUMSQ, S, V, LV, CJ, WORK, IFAIL)
INTEGER           JOB, M, N, LV, IFAIL
real              FSUMSQ, S(N), V(LV,N), CJ(N), WORK(N)
```

3 Description

E04YCF is intended for use when the nonlinear least-squares function, $F(x) = f^T(x)f(x)$, represents the goodness of fit of a nonlinear model to observed data. The routine assumes that the Hessian of $F(x)$, at the solution, can be adequately approximated by $2J^TJ$, where J is the Jacobian of $f(x)$ at the solution. The estimated variance-covariance matrix C is then given by

$$C = \sigma^2(J^TJ)^{-1} \quad J^TJ \text{ non-singular},$$

where σ^2 is the estimated variance of the residual at the solution, $\bar{x}$, given by

$$\sigma^2 = \frac{F(\bar{x})}{m-n},$$

m being the number of observations and n the number of variables.

The diagonal elements of C are estimates of the variances of the estimated regression coefficients. See the Chapter Introduction and Bard [1] and Wolberg [2] for further information on the use of C.

When J^TJ is singular then C is taken to be

$$C = \sigma^2(J^TJ)^{\dagger},$$

where $(J^TJ)^{\dagger}$ is the pseudo-inverse of J^TJ, and

$$\sigma^2 = \frac{F(\bar{x})}{m-k}, \quad k = \text{rank}(J)$$

but in this case the parameter IFAIL is returned as non-zero as a warning to the user that J has linear dependencies in its columns. The assumed rank of J can be obtained from IFAIL.

The routine can be used to find either the diagonal elements of C, or the elements of the jth column of C, or the whole of C.

E04YCF must be preceded by one of the nonlinear least-squares routines mentioned in Section 1, and requires the parameters FSUMSQ, S and V to be supplied by those routines. FSUMSQ is the residual sum of squares $F(\bar{x})$, and S and V contain the singular values and right singular vectors respectively in the singular value decomposition of J. S and V are returned directly by the comprehensive routines E04FCF, E04GBF, E04GDF and E04HEF, but are returned as part of the workspace parameter W from the easy-to-use routines E04FYF, E04GYF, E04GZF and E04HYF. In the case of E04FYF, S starts at W(NS), where

NS = 6×N + 2×M + M×N + 1 + max(1,N×(N−1)/2) and in the cases of the remaining easy-to-use routines, S starts at W(NS), where

NS = 7×N + 2×M + M×N + N×(N+1)/2 + 1 + max(1,N×(N−1)/2).

The parameter V starts immediately following the elements of S, so that V starts at W(NV), where

NV = NS + N.

For all the easy-to-use routines the parameter LV must be supplied as N. Thus a call to E04YCF following E04FYF can be illustrated as

```
.
.
.
      CALL E04FYF (M, N, LSFUN1, X, FSUMSQ, W, LW, IUSER, USER, IFAIL)
.
.
.
      NS = 6*N + 2*M + M*N + 1 + MAX((1,(N*(N-1))/2)
      NV = NS + N
      CALL E04YCF (JOB, M, N, FSUMSQ, W(NS), W(NV), N, CJ, WORK, IFAIL)
```

where the parameters M, N, FSUMSQ and the $(n+n^2)$ elements W(NS), WS(NS+1),. . .,W(NV+N^2−1) must not be altered between the calls to E04FYF and E04YCF. The above illustration also holds for a call to E04YCF following a call to one of E04GYF, E04GZF or E04HYF, except that NS must be computed as

```
      NS = 7*N + 2*M + M*N + (N*(N+1))/2 + 1 + MAX((1,N*(N-1))/2)
```

4 References

[1] Bard Y (1974) *Nonlinear Parameter Estimation* Academic Press

[2] Wolberg J R (1967) *Prediction Analysis* Van Nostrand

5 Parameters

1: JOB — INTEGER *Input*

On entry: which elements of C are returned as follows:

JOB = −1

The n by n symmetric matrix C is returned.

JOB = 0

The diagonal elements of C are returned.

JOB > 0

The elements of column JOB of C are returned.

Constraint: $-1 \le$ JOB $\le$ N.

2: M — INTEGER *Input*

On entry: the number m of observations (residuals $f_i(x)$).

Constraint: M $\ge$ N.

3: N — INTEGER *Input*

On entry: the number n of variables (x_j).

Constraint: $1 \leq$ N $\leq$ M.

4: FSUMSQ — ***real*** *Input*

On entry: the sum of squares of the residuals, $F(\bar{x})$, at the solution $\bar{x}$, as returned by the nonlinear least-squares routine.

Constraint: FSUMSQ ≥ 0.0.

5: S(N) — ***real*** array *Input*

On entry: the n singular values of the Jacobian as returned by the nonlinear least-squares routine. See Section 3 for information on supplying S following one of the easy-to-use routines.

6: V(LV,N) — ***real*** array *Input/Output*

On entry: the n by n right-hand orthogonal matrix (the right singular vectors) of J as returned by the nonlinear least-squares routine. See Section 3 for information on supplying V following one of the easy-to-use routines.

On exit: when JOB ≥ 0 then V is unchanged.

When JOB $= -1$ then the leading n by n part of V is overwritten by the n by n matrix C. When E04YCF is called with JOB $= -1$ following an easy-to-use routine this means that C is returned, column by column, in the n^2 elements of W given by W(NV),W(NV+1),...,W(NV+N$^2-1$). (See Section 3 for the definition of NV).

7: LV — INTEGER *Input*

On entry: the first dimension of the array V as declared in the (sub)program from which E04YCF is called. When V is passed in the workspace parameter W following one of the easy-to-use least-square routines, LV must be the value N.

Constraint: LV $\geq$ N if JOB$= -1$.

8: CJ(N) — ***real*** array *Output*

On exit: with JOB $= 0$, CJ returns the n diagonal elements of C.

With JOB $= j > 0$, CJ returns the n elements of the jth column of C.

When JOB $= -1$, CJ is not referenced.

9: WORK(N) — ***real*** array *Workspace*

When JOB $= -1$ or 0 then WORK is used as internal workspace.

When JOB > 0, WORK is not referenced.

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 = 1

On entry, JOB < -1,
or JOB $>$ N,
or N < 1,
or M $<$ N,
or FSUMSQ < 0.0,
or LV $<$ N.

IFAIL = 2

The singular values are all zero, so that at the solution the Jacobian matrix J has rank 0.

IFAIL > 2

At the solution the Jacobian matrix contains linear, or near linear, dependencies amongst its columns. In this case the required elements of C have still been computed based upon J having an assumed rank given by (IFAIL$-$2). The rank is computed by regarding singular values SV(j) that are not larger than $10\times\epsilon\times$SV(1) as zero, where ϵ is the ***machine precision*** (see X02AJF). Users who expect near linear dependencies at the solution and are happy with this tolerance in determining rank should call E04YCF with IFAIL = 1 in order to prevent termination by P01ABF. It is then essential to test the value of IFAIL on exit from E04YCF.

Overflow

If overflow occurs then either an element of C is very large, or the singular values or singular vectors have been incorrectly supplied.

7 Accuracy

The computed elements of C will be the exact covariances corresponding to a closely neighbouring Jacobian matrix J.

8 Further Comments

When JOB $= -1$ the time taken by the routine is approximately proportional to n^3. When JOB ≥ 0 the time taken by the routine is approximately proportional to n^2.

9 Example

To estimate the variance-covariance matrix C for the 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 program uses (0.5,1.0,1.5) as the initial guess at the position of the minimum and computes the least-squares solution using E04FYF. See the routine document E04FYF for further information.

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.

```
*     E04YCF Example Program Text.
*     Mark 19 Revised.  NAG Copyright 1999.
*     .. Parameters ..
      INTEGER          MDEC, NDEC, LUSER, LW
      PARAMETER        (MDEC=15,NDEC=3,LUSER=MDEC*(NDEC+1),
     +                 LW=7*NDEC+NDEC*NDEC+2*MDEC*NDEC+3*MDEC+NDEC*
     +                 (NDEC-1)/2)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             FSUMSQ
      INTEGER          I, IFAIL, J, M, N, NS, NV
*     .. Local Arrays ..
      real             CJ(NDEC), USER(LUSER), W(LW), X(NDEC)
      INTEGER          IUSER(1)
*     .. External Subroutines ..
      EXTERNAL         E04FYF, E04YCF, LSFUN1
*     .. Intrinsic Functions ..
      INTRINSIC        MAX
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E04YCF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      M = MDEC
      N = NDEC
*
*     For I = 1, 2, ..., M
*     The measurements Y(I) are stored in USER (I)
*     Observations T(I,1) are stored in USER(1*M+I)
*     Observations T(I,2) are stored in USER(2*M+I)
*     Observations T(I,3) are stored in USER(3*M+I)
*
```

```
      DO 20 I = 1, M
         READ (NIN,*) USER(I), (USER(J*M+I),J=1,N)
   20 CONTINUE
      X(1) = 0.5e0
      X(2) = 1.0e0
      X(3) = 1.5e0
      IFAIL = 1
*
      CALL E04FYF(M,N,LSFUN1,X,FSUMSQ,W,LW,IUSER,USER,IFAIL)
*
      IF (IFAIL.NE.0) THEN
         WRITE (NOUT,*)
         WRITE (NOUT,99999) 'Error exit from E04FDF. IFAIL = ', IFAIL
         WRITE (NOUT,*) '- see routine document'
      END IF
      IF (IFAIL.NE.1) THEN
         WRITE (NOUT,*)
         WRITE (NOUT,99998) 'On exit, the sum of squares is', FSUMSQ
         WRITE (NOUT,*) 'at the point'
         WRITE (NOUT,99997) (X(J),J=1,N)
*
*        Compute estimates of the variances of the sample regression
*        coefficients at the final point.
*        Since NS is greater than N we can use the first N elements
*        of W for the parameter WORK.
*
         NS = 6*N + 2*M + M*N + 1 + MAX(1,(N*(N-1))/2)
         NV = NS + N
         IFAIL = 1
*
         CALL E04YCF(0,M,N,FSUMSQ,W(NS),W(NV),N,CJ,W,IFAIL)
*
         IF (IFAIL.NE.0) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99999) 'Error exit from E04YCF.  IFAIL = ',
     +        IFAIL
            WRITE (NOUT,*) '- see routine document'
         END IF
         IF ((IFAIL.NE.1) .AND. (IFAIL.NE.2)) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,*)
     +        'and estimates of the variances of the sample'
            WRITE (NOUT,*) 'regression coefficients are'
            WRITE (NOUT,99997) (CJ(J),J=1,N)
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,I3)
99998 FORMAT (1X,A,F12.4)
99997 FORMAT (1X,3F12.4)
      END
*
      SUBROUTINE LSFUN1(M,N,XC,FVECC,IUSER,USER)
*     Routine to evaluate the residuals
*     .. Scalar Arguments ..
      INTEGER           M, N
*     .. Array Arguments ..
```

```
      real              FVECC(M), USER(M*(N+1)), XC(N)
      INTEGER           IUSER(1)
*     .. Local Scalars ..
      INTEGER           I
*     .. Executable Statements ..
      DO 20 I = 1, M
         FVECC(I) = XC(1) + USER(I+M)/(XC(2)*USER(I+M*2)+XC(3)
     +              *USER(I+M*3)) - USER(I)
   20 CONTINUE
      RETURN
      END
```

9.2 Program Data

```
E04YCF 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

```
 E04YCF Example Program Results

 On exit, the sum of squares is      0.0082
 at the point
       0.0824      1.1330      2.3437

 and estimates of the variances of the sample
 regression coefficients are
       0.0002      0.0948      0.0878
```