NAG Toolbox for MATLAB

e04yc

1 Purpose

e04yc 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 function may be used following any one of the nonlinear least-squares functions e04fc, e04fy, e04gb, e04gy, e04gd, e04gz, e04he or e04hy.

2 Syntax

```
[v, cj, ifail] = e04yc(job, m, fsumsq, s, v, 'n', n)
```

3 Description

e04yc is intended for use when the nonlinear least-squares function, $F(x) = f^{\mathrm{T}}(x)f(x)$, represents the goodness-of-fit of a nonlinear model to observed data. The function assumes that the Hessian of $F(x)$, at the solution, can be adequately approximated by $2J^{\mathrm{T}}J$, where J is the Jacobian of $f(x)$ at the solution. The estimated variance-covariance matrix C is then given by

$$C = \sigma^2 \left(J^{\mathrm{T}}J\right)^{-1}, \qquad J^{\mathrm{T}}J \qquad \text{nonsingular,}$$

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 E04 Chapter Introduction, Bard 1974 and Wolberg 1967 for further information on the use of C.

When $J^{\mathrm{T}}J$ is singular then C is taken to be

$$C = \sigma^2 \left(J^{\mathrm{T}}J\right)^{\dagger},$$

where $\left(J^{\mathrm{T}}J\right)^{\dagger}$ is the pseudo-inverse of $J^{\mathrm{T}}J$, and

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

but in this case the parameter **ifail** is returned as nonzero as a warning to you that J has linear dependencies in its columns. The assumed rank of J can be obtained from **ifail**.

The function 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.

e04yc must be preceded by one of the nonlinear least-squares functions mentioned in Section 1, and requires the parameters **fsumsq**, **s** and **v** to be supplied by those functions (e.g., see e04fc). **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 functions e04fc, e04gb, e04gd and e04he, but are returned as part of the workspace parameter **w** (from one of the easy-to-use functions). In the case of e04fy, **s** starts at $\mathbf{w}(NS)$, where

$$NS = 6 \times \mathbf{n} + 2 \times \mathbf{m} + \mathbf{m} \times \mathbf{n} + 1 + \max(1, \mathbf{n} \times (\mathbf{n} - 1)/2)$$

and in the cases of the remaining easy-to-use functions, **s** starts at $\mathbf{w}(NS)$, where

$$NS = 7 \times \mathbf{n} + 2 \times \mathbf{m} + \mathbf{m} \times \mathbf{n} + \mathbf{n} \times (\mathbf{n} + 1)/2 + 1 + \max(1, \mathbf{n} \times (\mathbf{n} - 1)/2).$$

The parameter **v** starts immediately following the elements of **s**, so that **v** starts at $\mathbf{w}(NV)$, where

$$NV = NS + \mathbf{n}.$$

For all the easy-to-use functions the parameter **ldv** must be supplied as **n**. Thus a call to e04yc following e04fy can be illustrated as

```
.
.
.
[x, fsumsq, user, ifail] = e04fy(m, lsfun1, x);
.
.
.
ns = 6*n + 2*m + m*n + 1 + max((1,(n*(n-1))/2);
nv = ns + n;
[v, cj, ifail] = e04yc(job, m, fsumsq, w(ns:nv-1), w(nv:numel(w)));
```

where the parameters **m**, **n**, **fsumsq** and the $(n + n^2)$ elements $\mathbf{w}(NS), \mathbf{w}(NS + 1), \ldots, \mathbf{w}(NV + \mathbf{n}^2 - 1)$ must not be altered between the calls to e04fy and e04yc. The above illustration also holds for a call to e04yc following a call to one of e04gy, e04gz or e04hy, except that NS must be computed as

$$NS = 7 \times \mathbf{n} + 2 \times \mathbf{m} + \mathbf{m} \times \mathbf{n} + (\mathbf{n} \times (\mathbf{n} + 1))/2 + 1 + \max((1, \mathbf{n} \times (\mathbf{n} - 1))/2).$$

4 References

Bard Y 1974 *Nonlinear Parameter Estimation* Academic Press

Wolberg J R 1967 *Prediction Analysis* Van Nostrand

5 Parameters

5.1 Compulsory Input Parameters

1: **job – int32 scalar**

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: $-1 \leq$ **job** $\leq$ **n**.

2: **m – int32 scalar**

The number m of observations (residuals $f_i(x)$).

Constraint: **m** $\geq$ **n**.

3: **fsumsq – double scalar**

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

Constraint: **fsumsq** ≥ 0.0.

4: **s(n) – double array**

The n singular values of the Jacobian as returned by the nonlinear least-squares function. See Section 3 for information on supplying **s** following one of the easy-to-use functions.

5: **v(ldv,n) – double array**

ldv, the first dimension of the array, must be at least if **job** $= -1$, **ldv** $\geq$ **n**.

The n by n right-hand orthogonal matrix (the right singular vectors) of J as returned by the nonlinear least-squares function. See Section 3 for information on supplying **v** following one of the easy-to-use functions.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The dimension of the arrays **s**, **v**, **cj**. (An error is raised if these dimensions are not equal.)

the number n of variables (x_j).

Constraint: $1 \leq$ **n** $\leq$ **m**.

5.3 Input Parameters Omitted from the MATLAB Interface

ldv, work

5.4 Output Parameters

1: **v(ldv,n) – double array**

If **job** ≥ 0, **v** is unchanged.

If **job** $= -1$, the leading n by n part of **v** contains the n by n matrix C. When e04yc is called with **job** $= -1$ following an easy-to-use function this means that C is returned, column by column, in the n^2 elements of **w** given by $\mathbf{w}(NV), \mathbf{w}(NV+1), \ldots, \mathbf{w}(NV+\mathbf{n}^2-1)$. (See Section 3 for the definition of NV.)

2: **cj(n) – double array**

If **job** $= 0$, **cj** returns the n diagonal elements of C.

If **job** $= j > 0$, **cj** returns the n elements of the jth column of C.

If **job** $= -1$, **cj** is not referenced.

3: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Note: e04yc may return useful information for one or more of the following detected errors or warnings.

ifail $= 1$

On entry, **job** < -1,
or **job** $>$ **n**,
or **n** < 1,
or **m** $<$ **n**,
or **fsumsq** < 0.0,
or **ldv** $<$ **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 $10\epsilon \times SV(1)$ as zero, where ϵ is the ***machine precision*** (see x02aj). If you expect near linear dependencies at the solution and are happy with this tolerance in determining rank you should call e04yc with **ifail** $= 1$ in order to prevent termination. It is then essential to test the value of **ifail** on exit from e04yc.

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 e04yc is approximately proportional to n^3. When **job** ≥ 0 the time taken by the function is approximately proportional to n^2.

9 Example

```
job = int32(0);
m = int32(15);
fsumsq = 0.00821487730657898;
s = [4.096503460740998;
     1.59495793805472;
     0.06125849312174952];
v = [0.9353959086918022, 0.3529512209498857, -0.02144597007884219;
     -0.2592284256717189, 0.6432345920936757, -0.7204511661853596;
     -0.2404893289241745, 0.6794664783225641, 0.6931739951192144];
[vOut, cj, ifail] = e04yc(job, m, fsumsq, s, v)
```

```
vOut =
    0.9354    0.3530   -0.0214
   -0.2592    0.6432   -0.7205
   -0.2405    0.6795    0.6932
cj =
    0.0002
    0.0948
    0.0878
ifail =
           0
```