G04EAF – 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

G04EAF computes orthogonal polynomial or dummy variables for a factor or classification variable.

2 Specification

```
SUBROUTINE G04EAF(TYPE, N, LEVELS, IFACT, X, LDX, V, REP, IFAIL)
INTEGER            N, LEVELS, IFACT(N), LDX, IFAIL
real               X(LDX,*), V(*), REP(LEVELS)
CHARACTER*1        TYPE
```

3 Description

In the analysis of an experimental design using a general linear model the factors or classification variables that specify the design have to be coded as dummy variables. G04EAF computes dummy variables that can then be used in the fitting of the general linear model using G02DAF.

If the factor of length n has k levels then the simplest representation is to define k dummy variables, X_j such that $X_j = 1$ if the factor is at level j and 0 otherwise for j = 1,2,...,k. However, there is usually a mean included in the model and the sum of the dummy variables will be aliased with the mean. To avoid the extra redundant parameter $k-1$ dummy variables can be defined as the contrasts between one level of the factor, the reference level and the remaining levels. If the reference level is the first level then the dummy variables can be defined as $X_j = 1$ if the factor is at level j and 0 otherwise for $j = 2, 3, \ldots, k$. Alternatively, the last level can be used as the reference level.

A second way of defining the $k-1$ dummy variables is to use a Helmert matrix in which levels $2, 3, \ldots, k$ are compared with the average effect of the previous levels. For example if $k = 4$ then the contrasts would be:

1	−1	−1	−1
2	1	−1	−1
3	0	2	−1
4	0	0	3

Thus variable j, for $j = 1, 2, \ldots, k-1$ is given by

$X_j = -1$ if factor is at level less than $j+1$

$$X_j = \sum_{i=1}^{j} r_i / r_{j+1} \text{ if factor is at level } j+1$$

$X_j = 0$ if factor is at level greater than $j+1$

where r_j is the number of replicates of level j.

If the factor can be considered as a set of values from an underlying continuous variable then the factor can be represented by a set of $k-1$ orthogonal polynomials representing the linear, quadratic etc. effects of the underlying variable. The orthogonal polynomial is computed using Forsythe's algorithm [2], see Cooper [1]. The values of the underlying continuous variable represented by the factor levels have to be supplied to the routine.

The orthogonal polynomials are standardized so that the sum of squares for each dummy variable is one. For the other methods integer (± 1) representations are retained except that in the Helmert representation the code of level $j+1$ in dummy variable j will be a fraction.

4 References

[1] Cooper B E (1968) Algorithm AS 10. The use of orthogonal polynomials *Appl. Statist.* **17** 283–287

[2] Forsythe G E (1957) Generation and use of orthogonal polynomials for data fitting with a digital computer *J. Soc. Indust. Appl. Math.* **5** 74–88

5 Parameters

1: TYPE — CHARACTER*1 *Input*

On entry: the type of dummy variable to be computed.

If TYPE = 'P', an orthogonal Polynomial representation is computed.
If TYPE = 'H', a Helmert matrix representation is computed.
If TYPE = 'F', the contrasts relative to the First level are computed.
If TYPE = 'L', the contrasts relative to the Last level are computed.
If TYPE = 'C', a Complete set of dummy variables is computed.

Constraint: TYPE = 'P', 'H', 'F', 'L' or 'C'.

2: N — INTEGER *Input*

On entry: the number of observations for which the dummy variables are to be computed, n.

Constraint: N $\geq$ LEVELS.

3: LEVELS — INTEGER *Input*

On entry: the number of levels of the factor, k.

Constraint: LEVELS ≥ 2.

4: IFACT(N) — INTEGER array *Input*

On entry: the n values of the factor.

Constraint: $1 \leq$ IFACT(i) $\leq$ LEVELS, for $i = 1, 2, \ldots, n$.

5: X(LDX,*) — ***real*** array *Output*

Note: the second dimension of the array X must be at least LEVELS−1 if TYPE = 'P', 'H', 'F' or 'L' and LEVELS if TYPE = 'C' .

On exit: the n by k^* matrix of dummy variables, where $k^* = k - 1$ if TYPE = 'P', 'H', 'F' or 'L' and $k^* = k$ if TYPE = 'C'.

6: LDX — INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G04EAF is called.

Constraint: LDX $\geq$ N.

7: V(*) — ***real*** array *Input*

Note: the dimension of the array V must be at least LEVELS if TYPE = 'P' and 1 otherwise.

On entry: if TYPE = 'P' the k distinct values of the underlying variable for which the orthogonal polynomial is to be computed. If TYPE $\neq$ 'P' V is not referenced.

Constraint: if TYPE = 'P' then the k values of V must be distinct.

8: REP(LEVELS) — ***real*** array *Output*

On exit: the number of replications for each level of the factor, r_i, $i = 1, 2, \ldots, k$.

9: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

6 Errors 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 detected by the routine:

IFAIL = 1

On entry, LEVELS < 2,
or N < LEVELS,
or LDX < N,
or TYPE ≠ 'P', 'H', 'F', 'L' or 'C'.

IFAIL = 2

On entry, a value of IFACT is not in the range $1 \leq \text{IFACT}(i) \leq \text{LEVELS}$, for $i = 1, 2, \ldots, n$,
or TYPE = 'P' and not all values of V are distinct,
or not all levels are represented in IFACT.

IFAIL = 3

An orthogonal polynomial has all values zero. This will be due to some values of V being very close together. Note this can only occur if TYPE = 'P'.

7 Accuracy

The computations are stable.

8 Further Comments

Other routines for fitting polynomials can be found in Chapter E02.

9 Example

Data are read in from an experiment with four treatments and three observations per treatment with the treatment coded as a factor. G04EAF is used to compute the required dummy variables and the model is then fitted by G02DAF.

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.

```
*     G04EAF Example Program Text
*     Mark 17 Release. NAG Copyright 1995.
*     .. Parameters ..
      INTEGER          MMAX, NMAX
      PARAMETER        (MMAX=5,NMAX=12)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             RSS, TOL
      INTEGER          I, IDF, IFAIL, IP, IRANK, J, LDX, LEVELS, M, N
      LOGICAL          SVD
      CHARACTER        MEAN, TYPE, WEIGHT
*     .. Local Arrays ..
      real             B(MMAX), COV((MMAX*MMAX+MMAX)/2), H(NMAX),
```

```
     +                 P(MMAX*(MMAX+2)), Q(NMAX,MMAX+1), REP(MMAX),
     +                 RES(NMAX), SE(MMAX), V(MMAX),
     +                 WK(MMAX*MMAX+5*(MMAX-1)), WT(NMAX), X(NMAX,MMAX),
     +                 Y(NMAX)
      INTEGER          IFACT(NMAX), ISX(MMAX)
*     .. External Subroutines ..
      EXTERNAL         G02DAF, G04EAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G04EAF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, LEVELS, TYPE, WEIGHT, MEAN
      WRITE (NOUT,*)
      IF (N.LE.NMAX .AND. LEVELS.LE.MMAX) THEN
         IF (WEIGHT.EQ.'W' .OR. WEIGHT.EQ.'w') THEN
            DO 20 I = 1, N
               READ (NIN,*) IFACT(I), Y(I), WT(I)
   20       CONTINUE
         ELSE
            DO 40 I = 1, N
               READ (NIN,*) IFACT(I), Y(I)
   40       CONTINUE
         END IF
         IF (TYPE.EQ.'P' .OR. TYPE.EQ.'p') THEN
            READ (NIN,*) (V(J),J=1,LEVELS)
         END IF
*
*        Calculate dummy variables
*
         LDX = NMAX
         IFAIL = 0
*
         CALL G04EAF(TYPE,N,LEVELS,IFACT,X,LDX,V,REP,IFAIL)
*
         IF (TYPE.EQ.'C' .OR. TYPE.EQ.'c') THEN
            M = LEVELS
         ELSE
            M = LEVELS - 1
         END IF
         DO 60 J = 1, M
            ISX(J) = 1
   60    CONTINUE
         IP = M
         IF (MEAN.EQ.'M' .OR. MEAN.EQ.'m') IP = IP + 1
*        Set tolerance
         TOL = 0.00001e0
         IFAIL = 0
*
         CALL G02DAF(MEAN,WEIGHT,N,X,LDX,M,ISX,IP,Y,WT,RSS,IDF,B,SE,COV,
     +               RES,H,Q,NMAX,SVD,IRANK,P,TOL,WK,IFAIL)
*
         IF (SVD) THEN
            WRITE (NOUT,99999) 'Model not of full rank, rank = ', IRANK
            WRITE (NOUT,*)
         END IF
         WRITE (NOUT,99998) 'Residual sum of squares = ', RSS
         WRITE (NOUT,99999) 'Degrees of freedom = ', IDF
         WRITE (NOUT,*)
```

```
         WRITE (NOUT,*) 'Variable   Parameter estimate   Standard error'
         WRITE (NOUT,*)
         DO 80 J = 1, IP
            WRITE (NOUT,99997) J, B(J), SE(J)
   80    CONTINUE
      END IF
      STOP
*
99999 FORMAT (1X,A,I4)
99998 FORMAT (1X,A,e12.4)
99997 FORMAT (1X,I6,2e20.4)
      END
```

9.2 Program Data

```
G04EAF Example Program Data
 12 4 'C' 'U' 'M'
1 33.63
4 39.62
2 38.18
3 41.46
4 38.02
2 35.83
4 35.99
1 36.58
3 42.92
1 37.80
3 40.43
2 37.89
```

9.3 Program Results

```
 G04EAF Example Program Results

 Model not of full rank, rank =    4

 Residual sum of squares =   0.2223E+02
 Degrees of freedom =    8

 Variable   Parameter estimate   Standard error

      1          0.3056E+02          0.3849E+00
      2          0.5447E+01          0.8390E+00
      3          0.6743E+01          0.8390E+00
      4          0.1105E+02          0.8390E+00
      5          0.7320E+01          0.8390E+00
```