NAG Fortran Library Routine Document

G01NAF

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

G01NAF computes the cumulants and moments of quadratic forms in Normal variates.

2 Specification

```
 SUBROUTINE G01NAF(MOM, MEAN, N, A, LDA, EMU, SIGMA, LDSIG, L, RKUM,
1                  RMOM, WK, IFAIL)
 INTEGER           N, LDA, LDSIG, L, IFAIL
 real              A(LDA,N), EMU(*), SIGMA(LDSIG,N), RKUM(L), RMOM(*),
1                  WK(3*N*(N+1)/2+N)
 CHARACTER*1       MOM, MEAN
```

3 Description

Let x have an n-dimensional multivariate Normal distribution with mean μ and variance-covariance matrix Σ. Then for a symmetric matrix A, G01NAF computes up to the first 12 moments and cumulants of the quadratic form $Q = x^T Ax$. The sth moment (about the origin) is defined as

$$E(Q^s),$$

where E denotes expectation. The sth moment of Q can also be found as the coefficient of $t^s/s!$ in the expansion of $E(e^{Qt})$. The sth cumulant is defined as the coefficient of $t^s/s!$ in the expansion of $\log(E(e^{Qt}))$.

The routine is based on the routine CUM written by Magnus and Pesaran (1993a) and based on the theory given by Magnus (1978), Magnus (1979) and Magnus (1986).

4 References

Magnus J R (1978) The moments of products of quadratic forms in Normal variables *Statist. Neerlandica* **32** 201–210

Magnus J R (1979) The expectation of products of quadratic forms in Normal variables: the practice *Statist. Neerlandica* **33** 131–136

Magnus J R (1986) The exact moments of a ratio of quadratic forms in Normal variables *Ann. Économ. Statist.* **4** 95–109

Magnus J R and Pesaran B (1993a) The evaluation of cumulants and moments of quadratic forms in Normal variables (CUM): Technical description *Comput. Statist.* **8** 39–45

Magnus J R and Pesaran B (1993b) The evaluation of moments of quadratic forms and ratios of quadratic forms in Normal variables: Background, motivation and examples *Comput. Statist.* **8** 47–55

5 Parameters

1: MOM – CHARACTER*1 *Input*

On entry: indicates if moments are computed in addition to cumulants.

If MOM = 'C', only cumulants are computed.

If MOM = 'M', moments are computed in addition to cumulants.

Constraint: MOM = 'C' or 'M'.

2: MEAN – CHARACTER*1 *Input*

On entry: indicates if the mean, μ, is zero.

If MEAN = 'Z', μ is zero.

If MEAN = 'M', the value of μ is supplied in EMU.

Constraint: MEAN = 'Z' or 'M'.

3: N – INTEGER *Input*

On entry: the dimension of the quadratic form, n.

Constraint: N > 1.

4: A(LDA,N) – ***real*** array *Input*

On entry: the n by n symmetric matrix A. Only the lower triangle is referenced.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which G01NAF is called.

Constraint: LDA ≥ N.

6: EMU(*) – ***real*** array *Input*

Note: the dimension of the array EMU must be at least at least N if MEAN=M, and at least 1 otherwise.

On entry: if MEAN = 'M'EMU must contain the n elements of the vector μ. If MEAN = 'Z', EMU is not referenced.

7: SIGMA(LDSIG,N) – ***real*** array *Input*

On entry: the n by n variance-covariance matrix Σ. Only the lower triangle is referenced.

Constraint: the matrix Σ must be positive-definite.

8: LDSIG – INTEGER *Input*

On entry: the first dimension of the array SIGMA as declared in the (sub)program from which G01NAF is called.

Constraint: LDSIG ≥ N.

9: L – INTEGER *Input*

On entry: the required number of cumulants, and moments if specified.

Constraint: 1 ≤ L ≤ 12.

10: RKUM(L) – ***real*** array *Output*

On exit: the L cumulants of the quadratic form.

11: RMOM(*) – ***real*** array *Output*

Note: the dimension of the array RMOM must be at least at least L if MOM=M, and at least 1 otherwise.

On exit: if MOM = 'M', the L moments of the quadratic form.

12: WK(3*N*(N+1)/2+N) – ***real*** array *Workspace*

13: 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, for users not familiar with this parameter the recommended value is 0. **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 $= 1$

On entry, N ≤ 1,
or L < 1,
or L > 12,
or LDA $<$ N,
or LDSIG $<$ N,
or MOM $\neq$ 'C' or 'M',
or MEAN $\neq$ 'M' or 'Z'.

IFAIL $= 2$

On entry, the matrix Σ is not positive-definite.

7 Accuracy

In a range of tests the accuracy was found to be a modest multiple of ***machine precision***. See Magnus and Pesaran (1993b).

8 Further Comments

None.

9 Example

The example is given by Magnus and Pesaran (1993b) and considers the simple autoregression

$$y_t = \beta y_{t-1} + u_t, \quad t = 1, 2, \ldots n,$$

where $\{u_t\}$ is a sequence of independent Normal variables with mean zero and variance one, and y_0 is known. The moments of the quadratic form

$$Q = \sum_{t=2}^{n} y_t y_{t-1}$$

are computed using G01NAF. The matrix A is given by:

$$\begin{aligned} A(i+1,i) &= \tfrac{1}{2}, \quad i = 1, 2, \ldots n-1; \\ A(i,j) &= 0, \quad \text{otherwise.} \end{aligned}$$

The value of Σ can be computed using the relationships

$$\text{var}(y_t) = \beta^2 \, \text{var}(y_{t-1}) + 1$$

and

$$\text{cov}(y_t y_{t+k}) = \beta \, \text{cov}(y_t y_{t+k-1})$$

for $k \geq 0$ and $\text{var}(y_1) = 1$.

The values of β, y_0, n, and the number of moments required are read in and the moments and cumulants printed.

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.

```
*     G01NAF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NDIM
      PARAMETER        (NDIM=10)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             BETA, CON
      INTEGER          I, IFAIL, J, L, N
*     .. Local Arrays ..
      real             A(NDIM,NDIM), EMU(NDIM), RKUM(12), RMOM(12),
     +                 SIGMA(NDIM,NDIM), WK(3*NDIM*(NDIM+1)/2+NDIM)
*     .. External Subroutines ..
      EXTERNAL         G01NAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G01NAF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) BETA, CON
      READ (NIN,*) N, L
      IF (N.LE.NDIM .AND. L.LE.12) THEN
*
*        Compute A, EMU, and SIGMA for simple autoregression
*
         DO 40 I = 1, N
            DO 20 J = I, N
               A(J,I) = 0.0e0
   20       CONTINUE
   40    CONTINUE
         DO 60 I = 1, N - 1
            A(I+1,I) = 0.5e0
   60    CONTINUE
         EMU(1) = CON*BETA
         DO 80 I = 1, N - 1
            EMU(I+1) = BETA*EMU(I)
   80    CONTINUE
         SIGMA(1,1) = 1.0e0
         DO 100 I = 2, N
            SIGMA(I,I) = BETA*BETA*SIGMA(I-1,I-1) + 1.0e0
  100    CONTINUE
         DO 140 I = 1, N
            DO 120 J = I + 1, N
               SIGMA(J,I) = BETA*SIGMA(J-1,I)
  120       CONTINUE
  140    CONTINUE
         IFAIL = 0
*
         CALL G01NAF('M','M',N,A,NDIM,EMU,SIGMA,NDIM,L,RKUM,RMOM,WK,
     +               IFAIL)
*
         WRITE (NOUT,*)
```

```
         WRITE (NOUT,99999) ' N = ', N, ' BETA = ', BETA, ' CON = ', CON
         WRITE (NOUT,*)
         WRITE (NOUT,*) '    Cumulants       Moments'
         WRITE (NOUT,*)
         DO 160 I = 1, L
            WRITE (NOUT,99998) I, RKUM(I), RMOM(I)
  160    CONTINUE
      END IF
      STOP
*
99999 FORMAT (A,I3,2(A,F6.3))
99998 FORMAT (I3,e12.4,4X,e12.4)
      END
```

9.2 Program Data

```
G01NAF Example Program Data
0.8  1.0   : BETA, CON
10   4     : N, L
```

9.3 Program Results

```
 G01NAF Example Program Results

 N =  10 BETA =  0.800 CON =  1.000

    Cumulants       Moments

  1  0.1752E+02      0.1752E+02
  2  0.3501E+03      0.6569E+03
  3  0.1609E+05      0.3986E+05
  4  0.1170E+07      0.3404E+07
```