NAG 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,      &
                   RMOM, WK, IFAIL)

INTEGER            N, LDA, LDSIG, L, IFAIL
REAL (KIND=nag_wp) A(LDA,N), EMU(*), SIGMA(LDSIG,N), RKUM(L), RMOM(*),    &
                   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 $Q = x^{\mathrm{T}}Ax$. 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 $E\left(e^{Qt}\right)$. The sth cumulant is defined as the coefficient of $t^s/s!$ in the expansion of $\log\left(E\left(e^{Qt}\right)\right)$.

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.

MOM = 'C'
  Only cumulants are computed.

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.

MEAN = 'Z'
μ is zero.

MEAN = 'M'
The value of μ is supplied in EMU.

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

3: N – INTEGER *Input*

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

Constraint: $\mathrm{N} > 1$.

4: A(LDA, N) – REAL (KIND=nag_wp) 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: $\mathrm{LDA} \geq \mathrm{N}$.

6: EMU(*) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array EMU must be 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 (KIND=nag_wp) 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: $\mathrm{LDSIG} \geq \mathrm{N}$.

9: L – INTEGER *Input*

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

Constraint: $1 \leq \mathrm{L} \leq 12$.

10: RKUM(L) – REAL (KIND=nag_wp) array *Output*

On exit: the L cumulants of the quadratic form.

11: RMOM($*$) – REAL (KIND=nag_wp) array *Output*

Note: the dimension of the array RMOM must be 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($3 \times N \times (N + 1)/2 + N$) – REAL (KIND=nag_wp) array *Workspace*

13: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

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, if you are 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.**

On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

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, $N \leq 1$,
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.

IFAIL = -99

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL = -399

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL = -999

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

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 Parallelism and Performance

G01NAF is not threaded by NAG in any implementation.

G01NAF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

None.

10 Example

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

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

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:

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

The value of Σ can be computed using the relationships

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

and

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

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

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

10.1 Program Text

```
    Program g01nafe

!     G01NAF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: g01naf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
```

```
      Integer, Parameter              :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)              :: beta, con
      Integer                         :: i, ifail, j, l, lda, ldsig, lwk, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), emu(:), rkum(:), rmom(:),    &
                                         sigma(:,:), wk(:)
!     .. Executable Statements ..
      Write (nout,*) 'G01NAF Example Program Results'
      Write (nout,*)

!     Skip heading in data file
      Read (nin,*)

!     Read in the problem size
      Read (nin,*) beta, con
      Read (nin,*) n, l

      ldsig = n
      lda = n
      lwk = 3*n*(n+1)/2 + n
      Allocate (a(lda,n),emu(n),sigma(ldsig,n),rkum(l),rmom(l),wk(lwk))

!     Compute A, EMU, and SIGMA for simple autoregression
      Do i = 1, n
        Do j = i, n
          a(j,i) = 0.0E0_nag_wp
        End Do
      End Do
      Do i = 1, n - 1
        a(i+1,i) = 0.5E0_nag_wp
      End Do
      emu(1) = con*beta
      Do i = 1, n - 1
        emu(i+1) = beta*emu(i)
      End Do
      sigma(1,1) = 1.0E0_nag_wp
      Do i = 2, n
        sigma(i,i) = beta*beta*sigma(i-1,i-1) + 1.0E0_nag_wp
      End Do
      Do i = 1, n
        Do j = i + 1, n
          sigma(j,i) = beta*sigma(j-1,i)
        End Do
      End Do

!     Compute cumulants
      ifail = 0
      Call g01naf('M','M',n,a,lda,emu,sigma,ldsig,l,rkum,rmom,wk,ifail)

!     Display results
      Write (nout,99999) ' N = ', n, ' BETA = ', beta, ' CON = ', con
      Write (nout,*)
      Write (nout,*) '    Cumulants       Moments'
      Write (nout,*)
      Write (nout,99998)(i,rkum(i),rmom(i),i=1,l)

99999 Format (A,I3,2(A,F6.3))
99998 Format (I3,E12.4,4X,E12.4)
    End Program g01nafe
```

10.2 Program Data

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

10.3 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
```