NAG Fortran Library Routine Document

G01NBF

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
G01NBF computes the moments of ratios of quadratic forms in Normal variables and related statistics.

2 Specification

SUBROUTINE G01NBF(CASE, MEAN, N, A, LDA, B, LDB, C, LDC, ELA, EMU, SIGMA, LDSIG, L1, L2, LMAX, RMOM, ABSERR, EPS, WK, IFAIL)
INTEGER N, LDA, LDB, LDC, LDSIG, L1, L2, LMAX, IFAIL
REAL A(LDA,N), B(LDB,N), C(LDC,*), ELA(*), EMU(*), SIGMA(LDSIG,N), RMOM(L2-L1+1), ABSERR, EPS, WK(3*N*N+(8+L2)*N)
CHARACTER*1 CASE, MEAN

3 Description
Let $x$ have an $n$-dimensional multivariate Normal distribution with mean $\mu$ and variance-covariance matrix $\Sigma$. Then for a symmetric matrix $A$ and symmetric positive semi-definite matrix $B$, G01NBF computes a subset, $l_1$ to $l_2$, of the first 12 moments of the ratio of quadratic forms

$$R = \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T B \mathbf{x}}.$$ 

The $s$th moment (about the origin) is defined as

$$E(R^s),$$

where $E$ denotes the expectation. Alternatively, G01NBF will compute the following expectations:

$$E(R^s(a^T x))$$

and

$$E(R^s(x^T C x)),$$

where $a$ is a vector of length $n$ and $C$ is a $n$ by $n$ symmetric matrix, if they exist. In the case of (2) the moments are zero if $\mu = 0$.

The conditions of theorems 1, 2 and 3 of Magnus (1986) and Magnus (1990) are used to check for the existence of the moments. If all the requested moments do not exist, the computations are carried out for those moments that are requested up to the maximum that exist, $l_{\text{MAX}}$.

G01NBF is based on the routine QRMOM written by Magnus and Pesaran (1993a) and based on the theory given by Magnus (1986) and Magnus (1990). The computation of the moments requires first the computation of the eigenvectors of the matrix $L^T B L$, where $LL^T = \Sigma$. The matrix $L^T B L$ must be positive semi-definite and not null. Given the eigenvectors of this matrix, a function which has to be integrated over the range zero to infinity can be computed. This integration is performed using D01AMF.

4 References


5 Parameters

1: CASE – CHARACTER*1

*Input*

*On entry:* indicates the moments of which function are to be computed.

If CASE = 'R' (Ratio), $E(R^t)$ is computed.

If CASE = 'L' (Linear with ratio), $E(R^t(a^T x))$ is computed.

If CASE = 'Q' (Quadratic with ratio), $E(R^t(x^T C x))$ is computed.

*Constraint:* CASE = 'R', 'L' or 'Q'.

2: MEAN – CHARACTER*1

*Input*

*On entry:* indicates if the mean, $\mu$, is zero.

If MEAN = 'Z', $\mu$ is zero.

If MEAN = 'M', the value of $\mu$ 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 G01NBF is called.

*Constraint:* $LDA \geq N$.

6: B(LDB,N) – real array

*Input*

*On entry:* the $n$ by $n$ positive semi-definite symmetric matrix $B$. Only the lower triangle is referenced.

*Constraint:* the matrix $B$ must be positive semi-definite.

7: LDB – INTEGER

*Input*

*On entry:* the first dimension of the array B as declared in the (sub)program from which G01NBF is called.

*Constraint:* $LDB \geq N$.

8: C(LDC,*) – real array

*Input*

*Note:* the second dimension of the array C must be at least $N$ if CASE = 'Q', and at least 1 otherwise.

*On entry:* if CASE = 'Q', C must contain the $n$ by $n$ symmetric matrix $C$; only the lower triangle is referenced. If CASE $\neq 'Q'$, C is not referenced.
9: LDC – INTEGER  
   On entry: the first dimension of the array C as declared in the (sub)program from which G01NBF is called.
   Constraint: if CASE = 'Q', LDC ≥ N, otherwise LDC ≥ 1.

10: ELA(*) – real array  
   Note: the dimension of the array ELA must be at least N if CASE = 'L', and at least 1 otherwise.
   On entry: if CASE = 'L', ELA must contain the vector a of length n, otherwise A is not referenced.

11: EMU(*) – real array  
   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.

12: SIGMA(LDSIG,N) – real array  
   On entry: the n by n variance-covariance matrix Σ. Only the lower triangle is referenced.
   Constraint: the matrix Σ must be positive-definite.

13: LDSIG – INTEGER  
   On entry: the first dimension of the array SIGMA as declared in the (sub)program from which G01NBF is called.
   Constraint: LDSIG ≥ N.

14: L1 – INTEGER  
   On entry: the first moment to be computed, l₁.
   Constraint: 0 < L1 ≤ L2.

15: L2 – INTEGER  
   On entry: the last moment to be computed, l₂.
   Constraint: L1 ≤ L2 ≤ 12.

16: LMAX – INTEGER  
   On exit: the highest moment computed, l_{MAX}. This will be l₂ if IFAIL = 0 on exit.

17: RMOM(L2−L1+1) – real array  
   On exit: the l₁ to l_{MAX} moments.

18: ABSERR – real  
   On exit: the estimated maximum absolute error in any computed moment.

19: EPS – real  
   On entry: the relative accuracy required for the moments, this value is also used in the checks for the existence of the moments. If EPS = 0.0, a value of $\sqrt{\epsilon}$ where $\epsilon$ is the machine precision used.
   Constraint: EPS = 0.0 or EPS ≥ machine precision.
WK(3*N*N+(8+L2)*N) – real array

Workspace

21: 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, because for this routine the values of the output parameters may be useful even if IFAIL ≠ 0 on exit, the recommended value is −1. 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 LDA < N,

or LDB < N,

or LDSIG < N,

or CASE = 'Q' and LDC < N,

or CASE ≠ 'Q' and LDC < 1,

or L1 < 1,

or L1 > L2,

or L2 > 12,

or CASE ≠ 'R', 'L' or 'Q',

or MEAN ≠ 'M' or 'Z',

or EPS ≠ 0.0 and EPS < machine precision.

IFAIL = 2

On entry, Σ is not positive-definite,

or B is not positive semi-definite or is null.

IFAIL = 3

None of the required moments can be computed.

IFAIL = 4

The matrix \( L^TBL \) is not positive semi-definite or is null.

IFAIL = 5

The computation to compute the eigenvalues required in the calculation of moments has failed to converge: this is an unlikely error exit.

IFAIL = 6

Only some of the required moments have been computed, the highest is given by LMAX.

IFAIL = 7

The required accuracy has not been achieved in the integration. An estimate of the accuracy is returned in ABSERR.
7  Accuracy
The relative accuracy is specified by EPS and an estimate of the maximum absolute error for all computed
moments is returned in ABSERR.

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 least-squares estimate of \( \beta, \hat{\beta} \), is given by
\[ \hat{\beta} = \frac{\sum_{t=2}^{n} y_ty_{t-1}}{\sum_{t=2}^{n} y_t^2}. \]
Thus \( \hat{\beta} \) can be written as a ratio of quadratic forms and its moments computed using G01NBF. The matrix
\( A \) is given by
\[ A(i+1,i) = \frac{1}{2}, \quad i = 1, 2, \ldots, n-1; \]
\[ A(i,j) = 0, \quad \text{otherwise}, \]
and the matrix \( B \) is given by
\[ B(i,i) = 1, \quad i = 1, 2, \ldots, n-1; \]
\[ B(i,j) = 0, \quad \text{otherwise}. \]
The value of \( \Sigma \) 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 \( \beta, y_0, n \), and the number of moments required are read in and the moments computed and
printed.

9.1 Program Text
Note: the listing of the example program presented below uses bold italics 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.

*  G01NBF Example Program Text
*  .. Parameters ..
INTEGER NDIM
PARAMETER (NDIM=10)
INTEGER NIN, NOUT
PARAMETER (NIN=5, NOUT=6)
*  .. Local Scalars ..
real ABSERR, BETA, Y0
INTEGER I, IFAIL, J, L1, L2, LMAX, N
*  .. Local Arrays ..
real A(NDIM,NDIM), B(NDIM,NDIM), C(NDIM,NDIM),
+ ELA(NDIM), EMU(NDIM), RMOM(12), SIGMA(NDIM,NDIM),
+ WK(3*NDIM+NDIM+20*NDIM)
* .. External Subroutines ..
EXTERNAL G01NBF
* .. Executable Statements ..
WRITE (NOUT,*) 'G01NBF Example Program Results'
READ (NIN,*)
READ (NIN,*) BETA, Y0
READ (NIN,*) N, L1, L2
IF (N.LE.NDIM .AND. L2.LE.12) THEN
* Compute A, EMU, and SIGMA for simple autoregression
* DO 40 I = 1, N
 DO 20 J = I, N
   A(J,I) = 0.0
   B(J,I) = 0.0
  20 CONTINUE
40 CONTINUE
 DO 60 I = 1, N - 1
   A(I+1,I) = 0.5
   B(I,I) = 1.0
  60 CONTINUE
EMU(1) = Y0*BETA
 DO 80 I = 1, N - 1
   EMU(I+1) = BETA*EMU(I)
  80 CONTINUE
SIGMA(1,1) = 1.0
 DO 100 I = 2, N
   SIGMA(I,I) = BETA*BETA*SIGMA(I-1,I-1) + 1.0
 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 = -1
 IF (IFAIL.EQ.0 .OR. IFAIL.GE.6) THEN
   WRITE (NOUT,*)
   WRITE (NOUT,99999) ' N = ', N, ' BETA = ', BETA, ' Y0 = ', Y0
   WRITE (NOUT,*) ' Moments'
   J = 0
   DO 160 I = L1, LMAX
     J = J + 1
     WRITE (NOUT,99998) I, RMOM(J)
 160 CONTINUE
 END IF
END IF
* CALL G01NBF('Ratio','Mean',N,A,NDIM,B,NDIM,C,NDIM,ELA,EMU,+
   SIGMA,NDIM,L1,L2,LMAX,RMOM,ABSERR,0.0,WK,IFAIL)
* IF (IFAIL.EQ.0 .OR. IFAIL.GE.6) THEN
  WRITE (NOUT,99999) ' N = ', N, ' BETA = ', BETA, ' Y0 = ', Y0
  WRITE (NOUT,*) ' Moments'
  WRITE (NOUT,99998) ' I, RMOM(J)
 99999 FORMAT (A,I3,2(A,F6.3))
 99998 FORMAT (I3,E12.3)
END

9.2 Program Data
G01NBF Example Program Data
0.8 1.0 : Beta Y0
10 1 3 : N L1 L2

G01NBF.6
9.3 Program Results

G01NBF Example Program Results

N = 10 BETA = 0.800 Y0 = 1.000

Moments

1  0.682E+00
2  0.536E+00
3  0.443E+00