NAG Library Routine Document

G01LBF

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

G01LBF returns a number of values of the probability density function (PDF), or its logarithm, for the multivariate Normal (Gaussian) distribution.

2 Specification

```
SUBROUTINE G01LBF (ILOG, K, N, X, LDX, XMU, IULD, SIG, LDSIG, PDF, RANK,        &
                   IFAIL)
INTEGER            ILOG, K, N, LDX, IULD, LDSIG, RANK, IFAIL
REAL (KIND=nag_wp) X(LDX,*), XMU(N), SIG(LDSIG,*), PDF(K)
```

3 Description

The probability density function, $f(X : \mu, \Sigma)$ of an n-dimensional multivariate Normal distribution with mean vector μ and n by n variance-covariance matrix Σ, is given by

$$f(X : \mu, \Sigma) = ((2\pi)^n |\Sigma|)^{-1/2} \exp\left(-\tfrac{1}{2}(X-\mu)^{\mathrm{T}}\Sigma^{-1}(X-\mu)\right).$$

If the variance-covariance matrix, Σ, is not of full rank then the probability density function, is calculated as

$$f(X : \mu, \Sigma) = ((2\pi)^r \mathrm{pdet}(\Sigma))^{-1/2} \exp\left(-\tfrac{1}{2}(X-\mu)^{\mathrm{T}}\Sigma^{-}(X-\mu)\right)$$

where $\mathrm{pdet}(\Sigma)$ is the pseudo-determinant, Σ^{-} a generalized inverse of Σ and r its rank.

G01LBF evaluates the PDF at k points with a single call.

4 References

None.

5 Parameters

1: ILOG – INTEGER *Input*

On entry: the value of ILOG determines whether the logarithmic value is returned in PDF.

ILOG $= 0$
$f(X : \mu, \Sigma)$, the probability density function is returned.

ILOG $= 1$
$\log(f(X : \mu, \Sigma))$, the logarithm of the probability density function is returned.

Constraint: ILOG $= 0$ or 1.

2: K – INTEGER *Input*

On entry: k, the number of points the PDF is to be evaluated at.

Constraint: K ≥ 0.

3: N – INTEGER *Input*

On entry: n, the number of dimensions.

Constraint: N $\geq$ 2.

4: X(LDX,$*$) – REAL (KIND=nag_wp) array *Input*

Note: the second dimension of the array X must be at least K.

On entry: X, the matrix of k points at which to evaluate the probability density function, with the ith dimension for the jth point held in X(i,j).

5: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G01LBF is called.

Constraint: LDX $\geq$ N.

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

On entry: μ, the mean vector of the multivariate Normal distribution.

7: IULD – INTEGER *Input*

On entry: indicates the form of Σ and how it is stored in SIG.

IULD $= 1$
SIG holds the lower triangular portion of Σ.

IULD $= 2$
SIG holds the upper triangular portion of Σ.

IULD $= 3$
Σ is a diagonal matrix and SIG only holds the diagonal elements.

IULD $= 4$
SIG holds the lower Cholesky decomposition, L such that $LL^{\mathrm{T}} = \Sigma$.

IULD $= 5$
SIG holds the upper Cholesky decomposition, U such that $U^{\mathrm{T}}U = \Sigma$.

Constraint: IULD $= 1$, 2, 3, 4 or 5.

8: SIG(LDSIG,$*$) – REAL (KIND=nag_wp) array *Input*

Note: the second dimension of the array SIG must be at least N.

On entry: information defining the variance-covariance matrix, Σ.

IULD $= 1$ or 2
SIG must hold the lower or upper portion of Σ, with Σ_{ij} held in SIG(i,j). The supplied variance-covariance matrix must be positive semidefinite.

IULD $= 3$
Σ is a diagonal matrix and the ith diagonal element, Σ_{ii}, must be held in SIG$(1,i)$

IULD $= 4$ or 5
SIG must hold L or U, the lower or upper Cholesky decomposition of Σ, with L_{ij} or U_{ij} held in SIG(i,j), depending on the value of IULD. No check is made that LL^{T} or $U^{\mathrm{T}}U$ is a valid variance-covariance matrix. The diagonal elements of the supplied L or U must be greater than zero

9: LDSIG – INTEGER *Input*

On entry: the first dimension of the array SIG as declared in the (sub)program from which G01LBF is called.

Constraints:

if IULD $= 3$, LDSIG ≥ 1;
otherwise LDSIG $\geq$ N.

10: PDF(K) – REAL (KIND=nag_wp) array *Output*

On exit: $f(X : \mu, \Sigma)$ or $\log(f(X : \mu, \Sigma))$ depending on the value of ILOG.

11: RANK – INTEGER *Output*

On exit: r, rank of Σ.

12: 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 $= 11$

On entry, ILOG $= \langle value \rangle$.
Constraint: ILOG $= 0$ or 1.

IFAIL $= 21$

On entry, K $= \langle value \rangle$.
Constraint: K ≥ 0.

IFAIL $= 31$

On entry, N $= \langle value \rangle$.
Constraint: N ≥ 2.

IFAIL $= 51$

On entry, LDX $= \langle value \rangle$ and N $= \langle value \rangle$.
Constraint: LDX $\geq$ N.

IFAIL $= 71$

On entry, IULD $= \langle value \rangle$.
Constraint: IULD $= 1, 2, 3, 4$ or 5.

IFAIL $= 81$

On entry, Σ is not positive semidefinite.

IFAIL = 82

On entry, at least one diagonal element of Σ is less than or equal to 0.

IFAIL = 83

On entry, Σ is not positive definite and eigenvalue decomposition failed.

IFAIL = 91

On entry, LDSIG = ⟨*value*⟩.
Constraint: if IULD = 3, LDSIG $\geq$ 1.

IFAIL = 92

On entry, LDSIG = ⟨*value*⟩.
Constraint: if IULD $\neq$ 3, LDSIG $\geq$ N.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

This example prints the value of the multivariate Normal PDF at a number of different points.

9.1 Program Text

```
    Program g01lbfe

!     G01LBF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: g01lbf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter              :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                         :: i, ifail, ilog, iuld, k, ldsig, ldx, &
                                         n, rank
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: pdf(:), sig(:,:), x(:,:), xmu(:)
!     .. Intrinsic Procedures ..
      Intrinsic                       :: repeat
!     .. Executable Statements ..
      Write (nout,*) 'G01LBF Example Program Results'
      Write (nout,*)

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

!     Read in the problem size and how the covariance matrix is stored
!     and whether the log PDF is required
      Read (nin,*) k, n, iuld, ilog

!     Allocate arrays
      ldx = n
      Allocate (x(ldx,k),xmu(n),pdf(k))
```

```
!     Read in and echo the vector of means
      Read (nin,*) xmu(1:n)
      Write (nout,*) 'Vector of Means: '
      Write (nout,99999) xmu(1:n)
      Write (nout,*)

!     Read in and echo the covariance matrix
      If (iuld==3) Then
!       Covariance matrix is diagonal
        ldsig = 1
        Allocate (sig(ldsig,n))
        Read (nin,*) sig(1,1:n)

        Write (nout,*) 'Diagonal Elements of Covariance Matrix: '
        Write (nout,99999) sig(1,1:n)

      Else
!       Read in an upper or lower triangular matrix
        ldsig = n
        Allocate (sig(ldsig,n))
        If (iuld==1 .Or. iuld==4) Then
!         Lower triangular matrix
          Read (nin,*)(sig(i,1:i),i=1,n)

          If (iuld==1) Then
            Call x04caf('Lower','Nonunit',n,n,sig,ldsig,'Covariance Matrix:', &
              ifail)
          Else
            Call x04caf('Lower','Nonunit',n,n,sig,ldsig, &
              'Lower Triangular Cholesky Factor of Covariance Matrix:',ifail)
          End If
        Else
!         Upper triangular matrix
          Read (nin,*)(sig(i,i:n),i=1,n)

          If (iuld==2) Then
            Call x04caf('Upper','Nonunit',n,n,sig,ldsig,'Covariance Matrix:', &
              ifail)
          Else
            Call x04caf('Upper','Nonunit',n,n,sig,ldsig, &
              'Upper Triangular Cholesky Factor of Covariance Matrix:',ifail)
          End If
        End If
      End If

!     Read in the points at which to evaluate the PDF
      Read (nin,*)(x(1:n,i),i=1,k)

!     Evaluate the PDF
      ifail = 0
      Call g01lbf(ilog,k,n,x,ldx,xmu,iuld,sig,ldsig,pdf,rank,ifail)

!     Display results
      Write (nout,*)
      Write (nout,*) 'Rank of the covariance matrix: ', rank
      Write (nout,*)
      If (ilog==1) Then
        Write (nout,*) '     log(PDF)                  X'
      Else
        Write (nout,*) '       PDF                     X'
      End If
      Write (nout,*) '  ', repeat('-',48)
      Do i = 1, k
        Write (nout,99998) pdf(i), x(1:n,i)
      End Do

99999 Format (1X,100(F8.4,1X))
99998 Format (1X,1P,E13.4,0P,10(1X,F8.4))
    End Program g01lbfe
```

9.2 Program Data

```
G01LBF Example Program Data
  2  4  1  0                  : K,N,IULD,ILOG
  0.10   0.20   0.30   0.40   : End XMU
  4.16
 -3.12   5.03
  0.56  -0.83   0.76
 -0.10   1.18   0.34   1.18   : End SIG
  1.00   1.00   1.00   1.00
  1.00   2.00   3.00   4.00   : End of X
```

9.3 Program Results

```
 G01LBF Example Program Results

 Vector of Means:
   0.1000   0.2000   0.3000   0.4000

 Covariance Matrix:
             1          2          3          4
1      4.1600
2     -3.1200     5.0300
3      0.5600    -0.8300     0.7600
4     -0.1000     1.1800     0.3400     1.1800

 Rank of the covariance matrix:  4

        PDF                      X
   -------------------------------------------------
    3.0307E-03   1.0000   1.0000   1.0000   1.0000
    4.5232E-06   1.0000   2.0000   3.0000   4.0000
```