NAG Toolbox for MATLAB

g05lx

1 Purpose

g05lx sets up a reference vector and generates an array of pseudo-random numbers from a multivariate Student's t distribution with ν degrees of freedom, mean vector a and covariance matrix $\frac{\nu}{\nu-2}C$.

2 Syntax

```
[x, iseed, r, ifail] = g05lx(mode, df, xmu, c, n, igen, iseed, r, 'm',
m, 'lr', lr)
```

3 Description

When the covariance matrix is nonsingular (i.e., strictly positive-definite), the distribution has probability density function

$$f(x) = \frac{\Gamma\left(\frac{(\nu+m)}{2}\right)}{(\pi\nu)^{m/2}\Gamma(\nu/2)|C|^{\frac{1}{2}}}\left[1 + \frac{(x-a)^{\mathrm{T}}C^{-1}(x-a)}{\nu}\right]^{\frac{-(\nu+m)}{2}}$$

where m is the number of dimensions, ν is the degrees of freedom, a is the vector of means, x is the vector of positions and $\frac{\nu}{\nu-2}C$ is the covariance matrix.

The function returns the value

$$x = a + \sqrt{\frac{\nu}{s}}z$$

where z is generated by g05ly from a Normal distribution with mean zero and covariance matrix C and s is generated by g05lc from a χ^2-distribution with ν degrees of freedom.

One of the initialization functions g05kb (for a repeatable sequence if computed sequentially) or g05kc (for a non-repeatable sequence) must be called prior to the first call to g05lx.

4 References

Knuth D E 1981 *The Art of Computer Programming (Volume 2)* (2nd Edition) Addison–Wesley

Wilkinson J H 1965 *The Algebraic Eigenvalue Problem* Oxford University Press, Oxford

5 Parameters

5.1 Compulsory Input Parameters

1: **mode – int32 scalar**

Selects the operation to be performed:

mode $= 0$

Initialize and generate random numbers.

mode $= 1$

Initialize only (i.e., set up reference vector).

mode $= 2$

Generate random numbers using previously set up reference vector.

Constraint: $0 \leq$ **mode** ≤ 2.

2: **df – int32 scalar**

ν, the number of degrees of freedom of the distribution.

Constraint: **df** ≥ 3.

3: **xmu(m) – double array**

a, the vector of means of the distribution.

4: **c(ldc,m) – double array**

ldc, the first dimension of the array, must be at least **m**.

Matrix which, along with **df** defines the covariance of the distribution. Only the upper triangle need be set.

Constraint: **c** must be positive semi-definite to ***machine precision***

5: **n – int32 scalar**

n, the number of random variates required.

Constraint: **n** ≥ 1.

6: **igen – int32 scalar**

Must contain the identification number for the generator to be used to return a pseudo-random number and should remain unchanged following initialization by a prior call to g05kb or g05kc.

7: **iseed(4) – int32 array**

Contains values which define the current state of the selected generator.

8: **r(lr) – double array**

If **mode** $= 2$, the reference vector as set up by g05lx in a previous call with **mode** $= 0$ or 1.

5.2 Optional Input Parameters

1: **m – int32 scalar**

Default: The dimension of the arrays **c**, **x**. (An error is raised if these dimensions are not equal.)

m, the number of dimensions of the distribution.

Constraint: **m** > 0.

2: **lr – int32 scalar**

Default: The dimension of the array **r**.

If **mode** $= 2$, it must be the same as the value of **lr** specified in the prior call to g05lx with **mode** $= 0$ or 1.

Constraint: **lr** $>$ **m**$($**m**$+1)+1$.

5.3 Input Parameters Omitted from the MATLAB Interface

ldc, ldx

5.4 Output Parameters

1: **x(ldx,m) – double array**

The array of pseudo-random multivariate Student's t vectors generated by the function, with $X(i,j)$ holding the jth dimension for the ith variate.

2: **iseed(4) – int32 array**

Contains updated values defining the new state of the selected generator.

3: **r(lr) – double array**

If **mode** $= 0$ or 1, the reference vector that can be used in subsequent calls to g05lx with **mode** $= 2$.

4: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **mode** $\neq 0$, 1 or 2.

ifail $= 2$

On entry, **m** < 1.

ifail $= 3$

On entry, **df** ≤ 2.

ifail $= 5$

The covariance matrix **c** is not positive semi-definite to ***machine precision***.

ifail $= 6$

On entry, **ldc** $<$ **m**.

ifail $= 7$

On entry, **n** < 1.

ifail $= 9$

On entry, **ldx** $<$ **n**.

ifail $= 10$

On entry, invalid value for **igen**. **igen** must be the same as the value as specified in the prior call to g05lx with **mode** $= 0$ or 1.

ifail $= 12$

The reference vector **r** has been corrupted or **m** has changed since **r** was set up in a previous call with **mode** $= 0$ or 1.

ifail $= 13$

On entry, **lr** $<$ **m**$($**m** $+ 1)$.

7 Accuracy

The maximum absolute error in LL^{T}, and hence in the covariance matrix of the resulting vectors, is less than $(m\epsilon + (m+3)\epsilon/2)$ times the maximum element of C, where ϵ is the ***machine precision***. Under normal circumstances, the above will be small compared to sampling error.

8 Further Comments

The time taken by g05lx is of order nm^3.

It is recommended that the diagonal elements of C should not differ too widely in order of magnitude. This may be achieved by scaling the variables if necessary. The actual matrix decomposed is $C + E = LL^{\mathrm{T}}$, where E is a diagonal matrix with small positive diagonal elements. This ensures that, even when C is singular, or nearly singular, the Cholesky Factor L corresponds to a positive-definite covariance matrix that agrees with C within ***machine precision***.

9 Example

```
mode = int32(0);
df = int32(10);
xmu = [1;
     2;
     -3;
     0];
c = [1.69, 0.39, -1.86, 0.07000000000000001;
     0, 98.01000000000001, -7.07, -0.71;
     0, 0, 11.56, 0.03;
     0, 0, 0, 0.01];
n = int32(10);
igen = int32(1);
iseed = [int32(1762543);
     int32(9324783);
     int32(42344);
     int32(742355)];
r = zeros(22, 1);
[igen, iseed] = g05kb(igen, iseed);
[x, iseedOut, rOut, ifail] = g05lx(mode, df, xmu, c, n, igen, iseed, r)
```

```
x =
    3.0999   -5.3200   -6.8459    0.1218
    0.1668    7.0595   -2.7861   -0.1162
    0.9310   11.5035    0.5182   -0.0219
   -0.7092    1.2452   -1.0941   -0.0633
    0.7267  -10.2979    0.0582    0.0576
    1.1014   13.6137   -3.3920   -0.0590
    0.0400   -1.6104    2.3485    0.0032
    2.1146    7.9958  -12.5358   -0.0471
    2.7721  -13.4943   -2.4307    0.1480
    0.0397  -18.3675    0.9874    0.1178
iseedOut =
     4474539
     9882971
     9948242
     5842803
rOut =
    4.5000
    1.3000
    0.3000
   -1.4308
    0.0538
         0
    9.8955
   -0.6711
   -0.0734
         0
         0
    3.0104
    0.0192
         0
         0
         0
    0.0367
```

```
    1.0000
    2.0000
   -3.0000
         0
   10.0000
ifail =
           0
```