NAG Library Routine Document

G05RDF

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

G05RDF sets up a reference vector and generates an array of pseudorandom numbers from a Normal (Gaussian) copula with covariance matrix C.

2   Specification

```
SUBROUTINE G05RDF (MODE, N, M, C, LDC, R, LR, STATE, X, LDX, IFAIL)
INTEGER           MODE, N, M, LDC, LR, STATE(*), LDX, IFAIL
REAL (KIND=nag_wp) C(LDC,M), R(LR), X(LDX,M)
```

3   Description

The Gaussian copula, G, is defined by

$$G(u_1, u_2, \ldots, u_m; C) = \Phi_C\left(\phi_{C_{11}}^{-1}(u_1), \phi_{C_{22}}^{-1}(u_2), \ldots, \phi_{C_{mm}}^{-1}(u_m)\right)$$

where m is the number of dimensions, Φ_C is the multivariate Normal density function with mean zero and covariance matrix C and $\phi_{C_{ii}}^{-1}$ is the inverse of the univariate Normal density function with mean zero and variance C_{ii}.

G05RZF is used to generate a vector from a multivariate Normal distribution and G01EAF is used to convert each element of that vector into a uniformly distributed value between zero and one.

One of the initialization routines G05KFF (for a repeatable sequence if computed sequentially) or G05KGF (for a non-repeatable sequence) must be called prior to the first call to G05RDF.

4   References

Nelsen R B (1998) *An Introduction to Copulas. Lecture Notes in Statistics 139* Springer

Sklar A (1973) Random variables: joint distribution functions and copulas *Kybernetika* **9** 499–460

5   Parameters

1:    MODE – INTEGER   *Input*

On entry: a code for selecting the operation to be performed by the routine.

MODE $= 0$
    Set up reference vector only.

MODE $= 1$
    Generate variates using reference vector set up in a prior call to G05RDF.

MODE $= 2$
    Set up reference vector and generate variates.

Constraint: MODE $= 0$, 1 or 2.

2:    N – INTEGER   *Input*

      On entry: n, the number of random variates required.

      Constraint: $N \geq 0$.

3:    M – INTEGER   *Input*

      On entry: m, the number of dimensions of the distribution.

      Constraint: $M > 0$.

4:    C(LDC, M) – REAL (KIND=nag_wp) array   *Input*

      On entry: the covariance matrix of the distribution. Only the upper triangle need be set.

      Constraint: C must be positive semidefinite to **machine precision**.

5:    LDC – INTEGER   *Input*

      On entry: the first dimension of the array C as declared in the (sub)program from which G05RDF is called.

      Constraint: $LDC \geq M$.

6:    R(LR) – REAL (KIND=nag_wp) array   *Communication Array*

      On entry: if MODE = 1, the reference vector as set up by G05RDF in a previous call with MODE = 0 or 2.

      On exit: if MODE = 0 or 2, the reference vector that can be used in subsequent calls to G05RDF with MODE = 1.

7:    LR – INTEGER   *Input*

      On entry: the dimension of the array R as declared in the (sub)program from which G05RDF is called. If MODE = 1, it must be the same as the value of LR specified in the prior call to G05RDF with MODE = 0 or 2.

      Constraint: $LR \geq M \times (M + 1) + 1$.

8:    STATE($*$) – INTEGER array   *Communication Array*

      Note: the actual argument supplied **must** be the array STATE supplied to the initialization routines G05KFF or G05KGF.

      On entry: contains information on the selected base generator and its current state.

      On exit: contains updated information on the state of the generator.

9:    X(LDX, M) – REAL (KIND=nag_wp) array   *Output*

      On exit: the array of values from a multivariate Gaussian copula, with $X(i, j)$ holding the jth dimension for the ith variate.

10:   LDX – INTEGER   *Input*

      On entry: the first dimension of the array X as declared in the (sub)program from which G05RDF is called.

      Constraint: $LDX \geq N$.

11:   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, MODE $= \langle value \rangle$.
> Constraint: MODE $= 0$, 1 or 2.

IFAIL $= 2$

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

IFAIL $= 3$

> On entry, M $= \langle value \rangle$.
> Constraint: M > 0.

IFAIL $= 4$

> On entry, the covariance matrix C is not positive semidefinite to *__machine precision__*.

IFAIL $= 5$

> On entry, LDC $= \langle value \rangle$ and M $= \langle value \rangle$.
> Constraint: LDC $\geq$ M.

IFAIL $= 6$

> M is not the same as when R was set up in a previous call.
> Previous value of M $= \langle value \rangle$ and M $= \langle value \rangle$.

IFAIL $= 7$

> On entry, LR is not large enough, LR $= \langle value \rangle$: minimum length required $= \langle value \rangle$.

IFAIL $= 8$

> On entry, STATE vector has been corrupted or not initialized.

IFAIL $= 10$

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

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

See Section 7 in G05RZF for an indication of the accuracy of the underlying multivariate Normal distribution.

8 Parallelism and Performance

G05RDF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G05RDF 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

The time taken by G05RDF 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*.

10 Example

This example prints ten pseudorandom observations from a Normal copula with covariance matrix

$$
\begin{bmatrix}
1.69 & 0.39 & -1.86 & 0.07 \\
0.39 & 98.01 & -7.07 & -0.71 \\
-1.86 & -7.07 & 11.56 & 0.03 \\
0.07 & -0.71 & 0.03 & 0.01
\end{bmatrix},
$$

generated by G05RDF. All ten observations are generated by a single call to G05RDF with MODE $= 2$. The random number generator is initialized by G05KFF.

10.1 Program Text

```
    Program g05rdfe

!     G05RDF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: g05kff, g05rdf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
```

```
!       .. Parameters ..
        Integer, Parameter                :: lseed = 1, nin = 5, nout = 6
!       .. Local Scalars ..
        Integer                           :: genid, i, ifail, ldc, ldx, lr,      &
                                             lstate, m, mode, n, subid
!       .. Local Arrays ..
        Real (Kind=nag_wp), Allocatable  :: c(:,:), r(:), x(:,:)
        Integer                           :: seed(lseed)
        Integer, Allocatable              :: state(:)
!       .. Executable Statements ..
        Write (nout,*) 'G05RDF Example Program Results'
        Write (nout,*)
        Flush (nout)

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

!       Read in the base generator information and seed
        Read (nin,*) genid, subid, seed(1)

!       Initial call to initialiser to get size of STATE array
        lstate = 0
        Allocate (state(lstate))
        ifail = 0
        Call g05kff(genid,subid,seed,lseed,state,lstate,ifail)

!       Reallocate STATE
        Deallocate (state)
        Allocate (state(lstate))

!       Initialize the generator to a repeatable sequence
        ifail = 0
        Call g05kff(genid,subid,seed,lseed,state,lstate,ifail)

!       Read in sample size and number of dimensions
        Read (nin,*) n, m

        ldc = m
        ldx = n
        lr = m*(m+1) + 1
        Allocate (c(ldc,m),x(ldx,m),r(lr))

!       Read in upper triangle portion of the covariance matrix
        Do i = 1, m
          Read (nin,*) c(i,i:m)
        End Do

!       Using a single call to G05RDF, so set up reference vector
!       and generate values in one go
        mode = 2

!       Generate variates
        ifail = 0
        Call g05rdf(mode,n,m,c,ldc,r,lr,state,x,ldx,ifail)

!       Display variates
        ifail = 0
        Call x04caf('General',' ',n,m,x,ldx,'Variates',ifail)

      End Program g05rdfe
```

10.2 Program Data

```
G05RDF Example Program Data
1  1  1762543          :: GENID,SUBID,SEED(1)
10   4                 :: N,M
1.69  0.39 -1.86  0.07
      98.01 -7.07 -0.71
            11.56  0.03
                   0.01 :: End of C (upper triangular part)
```

10.3 Program Results

```
G05RDF Example Program Results

Variates
           1        2        3        4
  1    0.6364   0.0517   0.4137   0.8817
  2    0.1065   0.2461   0.7993   0.3806
  3    0.7460   0.6313   0.2708   0.5421
  4    0.7983   0.0564   0.6868   0.9234
  5    0.1046   0.5790   0.8533   0.2208
  6    0.4925   0.2784   0.3513   0.5158
  7    0.3843   0.2349   0.9472   0.7801
  8    0.7871   0.9941   0.9403   0.2044
  9    0.4982   0.9015   0.7176   0.2914
 10    0.6717   0.5359   0.5961   0.4487
```