

# NAG Library Routine Document

## G05RZF

**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

G05RZF sets up a reference vector and generates an array of pseudorandom numbers from a multivariate Normal distribution with mean vector  $a$  and covariance matrix  $C$ .

### 2 Specification

```
SUBROUTINE G05RZF (MODE, N, M, XMU, C, LDC, R, LR, STATE, X, LDX, IFAIL)
INTEGER          MODE, N, M, LDC, LR, STATE(*), LDX, IFAIL
REAL (KIND=nag_wp) XMU(M), C(LDC,M), R(LR), X(LDX,*)
```

### 3 Description

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

$$f(x) = \sqrt{\frac{|C^{-1}|}{(2\pi)^m}} \exp\left(-\frac{1}{2}(x-a)^T C^{-1}(x-a)\right)$$

where  $m$  is the number of dimensions,  $C$  is the covariance matrix,  $a$  is the vector of means and  $x$  is the vector of positions.

Covariance matrices are symmetric and positive semidefinite. Given such a matrix  $C$ , there exists a lower triangular matrix  $L$  such that  $LL^T = C$ .  $L$  is not unique, if  $C$  is singular.

G05RZF decomposes  $C$  to find such an  $L$ . It then stores  $m$ ,  $a$  and  $L$  in the reference vector  $r$  which is used to generate a vector  $x$  of independent standard Normal pseudorandom numbers. It then returns the vector  $a + Lx$ , which has the required multivariate Normal distribution.

It should be noted that this routine will work with a singular covariance matrix  $C$ , provided  $C$  is positive semidefinite, despite the fact that the above formula for the probability density function is not valid in that case. Wilkinson (1965) should be consulted if further information is required.

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 G05RZF.

### 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 Arguments

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 or 3

Generate variates using reference vector set up in a prior call to G05RZF.

MODE = 2 or 4

Set up reference vector and generate variates.

The variates are stored differently in X for MODE = 1 or 2 compared with MODE = 3 or 4.

*Constraint:* MODE = 0, 1, 2, 3 or 4.

- 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: XMU(M) – REAL (KIND=nag\_wp) array *Input*  
*On entry:*  $a$ , the vector of means of the distribution.
- 5: 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*.
- 6: LDC – INTEGER *Input*  
*On entry:* the first dimension of the array  $C$  as declared in the (sub)program from which G05RZF is called.  
*Constraint:*  $LDC \geq M$ .
- 7: R(LR) – REAL (KIND=nag\_wp) array *Input/Output*  
*On entry:* if MODE = 1 or 3, the reference vector as set up by G05RZF in a previous call with MODE = 0, 2 or 4.  
*On exit:* if MODE = 0 or 2, the reference vector that can be used in subsequent calls to G05RZF with MODE = 1 or 3.
- 8: LR – INTEGER *Input*  
*On entry:* the dimension of the array  $R$  as declared in the (sub)program from which G05RZF is called. If MODE = 1 or 3, it must be the same as the value of LR specified in the prior call to G05RZF with MODE = 0, 2 or 4.  
*Constraint:*  $LR \geq M \times (M + 1) + 1$ .
- 9: 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.
- 10: X(LDX, \*) – REAL (KIND=nag\_wp) array *Output*  
**Note:** the second dimension of the array  $X$  must be at least  $M$  if MODE = 1 or 2 and at least  $N$  if MODE = 3 or 4.  
*On exit:* the array of pseudorandom multivariate Normal vectors generated by the routine.

Two possible storage orders are available. If  $\text{MODE} = 1$  or  $2$  then  $X(i, j)$  holds the  $j$ th dimension for the  $i$ th variate. If  $\text{MODE} = 3$  or  $4$  this ordering is reversed and  $X(j, i)$  holds the  $j$ th dimension for the  $i$ th variate.

11: LDX – INTEGER *Input*

*On entry:* the first dimension of the array X as declared in the (sub)program from which G05RZF is called.

*Constraints:*

if  $\text{MODE} = 1$  or  $2$ ,  $\text{LDX} \geq N$ ;  
if  $\text{MODE} = 3$  or  $4$ ,  $\text{LDX} \geq M$ .

12: IFAIL – INTEGER *Input/Output*

*On entry:* IFAIL must be set to 0,  $-1$  or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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, invalid value for MODE.  
Constraint:  $\text{MODE} = 0, 1, 2, 3$  or  $4$ .

IFAIL = 2

On entry,  $N = \langle \text{value} \rangle$ .  
Constraint:  $N \geq 0$ .

IFAIL = 3

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

IFAIL = 5

On entry, the covariance matrix  $C$  is not positive semidefinite to *machine precision*.

IFAIL = 6

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

IFAIL = 7

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

IFAIL = 8

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

IFAIL = 9

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

IFAIL = 11

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

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 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL = -399

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL = -999

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

## 7 Accuracy

Not applicable.

## 8 Parallelism and Performance

G05RZF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G05RZF 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 G05RZF 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  $C + E = LL^T$ , 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 multivariate Normal distribution with means vector

$$\begin{bmatrix} 1.0 \\ 2.0 \\ -3.0 \\ 0.0 \end{bmatrix}$$

and 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 G05RZF. All ten observations are generated by a single call to G05RZF with `MODE = 2`. The random number generator is initialized by G05KFF.

### 10.1 Program Text

```

Program g05rzfe

!      G05RZF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
Use nag_library, Only: g05kff, g05rzf, 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(:,,:), xmu(:)
Integer                     :: seed(lseed)
Integer, Allocatable         :: state(:)
!      .. Executable Statements ..
Write (nout,*) 'G05RZF 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 initializer 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 (x(ldx,m),c(ldc,m),r(lr),xmu(m))

!      Read in vector of means
      Read (nin,*) xmu(1:m)

!      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 G05RZF, so set up reference vector
!      and generate values in one go
      mode = 2

!      Generate variates
      ifail = 0
      Call g05rzf(mode,n,m,xmu,c,ldc,r,lr,state,x,ldx,ifail)

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

      End Program g05rzfe

```

## 10.2 Program Data

G05RZF Example Program Data

```

1 1 1762543      :: GENID,SUBID,SEED(1)
10 4           :: N,M
1.0 2.0 -3.0 0.0 :: XMU
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

G05RZF Example Program Results

Variates	1	2	3	4
1	1.4534	-14.1206	-3.7410	0.1184
2	-0.6191	-4.8000	-0.1473	-0.0304
3	1.8607	5.3206	-5.0753	0.0106
4	2.0861	-13.6996	-1.3451	0.1428
5	-0.6326	3.9729	0.5721	-0.0770
6	0.9754	-3.8162	-4.2978	0.0040
7	0.6174	-5.1573	2.5037	0.0772
8	2.0352	26.9359	2.2939	-0.0826
9	0.9941	14.7700	-1.0421	-0.0549
10	1.5780	2.8916	-2.1725	-0.0129

---