# NAG Library Routine Document

# G05SQF

**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

G05SQF generates a vector of pseudorandom numbers uniformly distributed over the interval $[a, b]$.

## 2 Specification

```
SUBROUTINE G05SQF (N, A, B, STATE, X, IFAIL)

INTEGER            N, STATE(*), IFAIL
REAL (KIND=nag_wp) A, B, X(N)
```

## 3 Description

If $a = 0$ and $b = 1$, G05SQF returns the next $n$ values $y_i$ from a uniform $(0, 1]$ generator (see G05SAF for details).

For other values of $a$ and $b$, G05SQF applies the transformation

$$x_i = a + (b - a)y_i.$$

The routine ensures that the values $x_i$ lie in the closed interval $[a, b]$.

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 G05SQF.

## 4 References

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

## 5 Parameters

1: N – INTEGER *Input*

*On entry*: $n$, the number of pseudorandom numbers to be generated.

*Constraint*: $\mathrm{N} \geq 0$.

2: A – REAL (KIND=nag_wp) *Input*
3: B – REAL (KIND=nag_wp) *Input*

*On entry*: the end points $a$ and $b$ of the uniform distribution.

*Constraint*: $\mathrm{A} \leq \mathrm{B}$.

4: 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.

5: X(N) – REAL (KIND=nag_wp) array *Output*

*On exit*: the $n$ pseudorandom numbers from the specified uniform distribution.

6: 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, N $< 0$.

IFAIL $= 3$

On entry, B $<$ A.

IFAIL $= 4$

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

## 7 Accuracy

Not applicable.

## 8 Further Comments

Although $y_i$ takes a value from the half closed interval $(0, 1]$ and $x_i = a + (b - a)y_i$, $x_i$ is documented as taking values from the closed interval $[a, b]$. This is because for some values of $a$ and $b$, G05SQF may return a value of $a$ due to numerical rounding.

## 9 Example

This example prints five pseudorandom numbers from a uniform distribution between $-1.0$ and 1.0, generated by a single call to G05SQF, after initialization by G05KFF.

### 9.1 Program Text

```
    Program g05sqfe

!     G05SQF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: g05kff, g05sqf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
```

```
!     .. Parameters ..
      Integer, Parameter                :: lseed = 1, nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)                :: a, b
      Integer                           :: genid, ifail, lstate, n, subid
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: x(:)
      Integer                           :: seed(lseed)
      Integer, Allocatable              :: state(:)
!     .. Executable Statements ..
      Write (nout,*) 'G05SQF Example Program Results'
      Write (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
      Read (nin,*) n

      Allocate (x(n))

!     Read in the distribution parameters
      Read (nin,*) a, b

!     Generate the variates
      ifail = 0
      Call g05sqf(n,a,b,state,x,ifail)

!     Display the variates
      Write (nout,99999) x(1:n)

99999 Format (1X,F10.4)
    End Program g05sqfe
```

## 9.2 Program Data

```
G05SQF Example Program Data
1  1  1762543     :: GENID,SUBID,SEED(1)
5 3               :: N,NMIX
-1.0 1.0          :: A,B
```

## 9.3 Program Results

```
 G05SQF Example Program Results

     0.2727
    -0.7870
     0.4921
     0.5965
    -0.7908
```