NAG Library Routine Document

G05YMF

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

G05YMF generates a uniformly distributed low-discrepancy sequence as proposed by Sobol, Faure or Niederreiter. It must be preceded by a call to one of the initialization routines G05YLF or G05YNF.

2 Specification

```
SUBROUTINE G05YMF (N, RCORD, QUAS, LDQUAS, IREF, IFAIL)
INTEGER            N, RCORD, LDQUAS, IREF(liref), IFAIL
REAL (KIND=nag_wp) QUAS(LDQUAS,tdquas)
```

3 Description

Low discrepancy (quasi-random) sequences are used in numerical integration, simulation and optimization. Like pseudorandom numbers they are uniformly distributed but they are not statistically independent, rather they are designed to give more even distribution in multidimensional space (uniformity). Therefore they are often more efficient than pseudorandom numbers in multidimensional Monte–Carlo methods.

G05YMF generates a set of points $x^1, x^2, \ldots, x^N$ with high uniformity in the S-dimensional unit cube $I^S = [0,1]^S$.

Let G be a subset of I^S and define the counting function $S_N(G)$ as the number of points $x^i \in G$. For each $x = (x_1, x_2, \ldots, x_S) \in I^S$, let G_x be the rectangular S-dimensional region

$$G_x = [0, x_1) \times [0, x_2) \times \cdots \times [0, x_S)$$

with volume $x_1, x_2, \ldots, x_S$. Then one measure of the uniformity of the points $x^1, x^2, \ldots, x^N$ is the discrepancy:

$$D_N^*\left(x^1, x^2, \ldots, x^N\right) = \sup_{x \in I^S} |S_N(G_x) - N x_1, x_2, \ldots, x_S|.$$

which has the form

$$D_N^*\left(x^1, x^2, \ldots, x^N\right) \leq C_S (\log N)^S + O\left((\log N)^{S-1}\right) \quad \text{for all} \quad N \geq 2.$$

The principal aim in the construction of low-discrepancy sequences is to find sequences of points in I^S with a bound of this form where the constant C_S is as small as possible.

The type of low-discrepancy sequence generated by G05YMF depends on the initialization routine called and can include those proposed by Sobol, Faure or Niederreiter. If the initialization routine G05YNF was used then the sequence will be scrambled (see Section 3 in G05YNF for details).

4 References

Bratley P and Fox B L (1988) Algorithm 659: implementing Sobol's quasirandom sequence generator *ACM Trans. Math. Software* **14(1)** 88–100

Fox B L (1986) Algorithm 647: implementation and relative efficiency of quasirandom sequence generators *ACM Trans. Math. Software* **12(4)** 362–376

5 Arguments

Note: the following variables are used in the parameter descriptions:

$idim$ = IDIM, the number of dimensions required, see G05YLF or G05YNF

$liref$ = LIREF, the length of IREF as supplied to the initialization routine G05YLF or G05YNF

$tdquas$ = N if RCORD = 1; otherwise $tdquas = idim$.

1: N – INTEGER *Input*

On entry: the number of quasi-random numbers required.

Constraint: N ≥ 0 and N + previous number of generated values $\leq 2^{31} - 1$.

2: RCORD – INTEGER *Input*

On entry: the order in which the generated values are returned.

Constraint: RCORD = 1 or 2.

3: QUAS(LDQUAS, $tdquas$) – REAL (KIND=nag_wp) array *Output*

On exit: contains the N quasi-random numbers of dimension $idim$.

If RCORD = 1, QUAS(i, j) holds the jth value for the ith dimension.

If RCORD = 2, QUAS(i, j) holds the ith value for the jth dimension.

4: LDQUAS – INTEGER *Input*

On entry: the first dimension of the array QUAS as declared in the (sub)program from which G05YMF is called.

Constraints:

if RCORD = 1, LDQUAS $\geq idim$;
if RCORD = 2, LDQUAS $\geq$ N.

5: IREF($liref$) – INTEGER array *Communication Array*

On entry: contains information on the current state of the sequence.

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

6: 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, N $= \langle value \rangle$.
Constraint: N ≥ 0.

On entry, value of N would result in too many calls to the generator: N $= \langle value \rangle$, generator has previously been called $\langle value \rangle$ times.

IFAIL $= 2$

On entry, RCORD $= \langle value \rangle$.
Constraint: RCORD $= 1$ or 2.

IFAIL $= 4$

On entry, LDQUAS $= \langle value \rangle$, $idim = \langle value \rangle$.
Constraint: if RCORD $= 1$, LDQUAS $\geq idim$.

On entry, LDQUAS $= \langle value \rangle$ and N $= \langle value \rangle$.
Constraint: if RCORD $= 2$, LDQUAS $\geq$ N.

IFAIL $= 5$

On entry, IREF has either not been initialized or has been corrupted.

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

G05YMF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

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.

The Sobol, Sobol (A659) and Niederreiter quasi-random number generators in G05YMF have been parallelized, but require quite large problem sizes to see any significant performance gain. Parallelism is only enabled when RCORD $= 2$. The Faure generator is serial.

9 Further Comments

None.

10 Example

This example calls G05YLF and G05YMF to estimate the value of the integral

$$\int_0^1 \cdots \int_0^1 \prod_{i=1}^{s} |4x_i - 2| dx_1, \, dx_2, \ldots, \, dx_s = 1.$$

In this example the number of dimensions S is set to 8.

10.1 Program Text

```
!   G05YMF Example Program Text
!   Mark 26 Release. NAG Copyright 2016.

    Module g05ymfe_mod

!     G05YMF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Accessibility Statements ..
      Private
      Public                           :: ifun
!     .. Parameters ..
      Integer, Parameter, Public       :: nin = 5, nout = 6
    Contains
      Function ifun(x,lx)
!       Function being integrated, in this example
!       ABS(4.0 X - 2)

!       .. Function Return Value ..
        Real (Kind=nag_wp)             :: ifun
!       .. Scalar Arguments ..
        Integer, Intent (In)           :: lx
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (In) :: x(lx)
!       .. Local Scalars ..
        Integer                        :: d
!       .. Intrinsic Procedures ..
        Intrinsic                      :: abs
!       .. Executable Statements ..
        ifun = 1.0E0_nag_wp
        Do d = 1, lx
          ifun = ifun*abs(4.0E0_nag_wp*x(d)-2.0E0_nag_wp)
        End Do
      End Function ifun
    End Module g05ymfe_mod
    Program g05ymfe

!     G05YMF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: g05ylf, g05ymf, nag_wp, x04caf
      Use g05ymfe_mod, Only: ifun, nin, nout
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: sum, vsbl
      Integer                          :: dn, genid, i, idim, ifail, iskip,    &
                                          ldquas, liref, n, rcord
      Character (80)                   :: title
```

```
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: quas(:,:)
      Integer, Allocatable             :: iref(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: real
!     .. Executable Statements ..
      Write (nout,*) 'G05YMF Example Program Results'
      Write (nout,*)

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

!     Fix the RCORD = 1, so QUAS(IDIM,N). As we
!     are accessing each dimension in turn for a given variate
!     when evaluating the function, this is more efficient
      rcord = 1

!     Read in the generator to use
      Read (nin,*) genid

!     Read in the problem size
      Read (nin,*) n, idim, iskip

      If (genid==4) Then
        liref = 407
      Else
        liref = 32*idim + 7
      End If
      ldquas = idim
      Allocate (quas(ldquas,n),iref(liref))

!     Initialize the generator
      ifail = 0
      Call g05ylf(genid,idim,iref,liref,iskip,ifail)

!     Generate N quasi-random variates
      ifail = 0
      Call g05ymf(n,rcord,quas,ldquas,iref,ifail)

!     Evaluate the function, and sum
      sum = 0.0E0_nag_wp
      Do i = 1, n
        sum = sum + ifun(quas(1:idim,i),idim)
      End Do

!     Convert sum to mean value
      vsbl = sum/real(n,kind=nag_wp)
      Write (nout,*)
      Write (nout,99999) 'Value of integral = ', vsbl

!     Read in number of variates to display
      Read (nin,*) dn

!     Display the first DN variates
      Write (nout,*)
      Write (title,99998) 'First ', dn, ' variates for all ', idim,           &
        ' dimensions'
      Flush (nout)
      ifail = 0
      Call x04caf('General',' ',idim,dn,quas,ldquas,title,ifail)

99999 Format (1X,A,F8.4)
99998 Format (A,I0,A,I0,A)
    End Program g05ymfe
```

10.2 Program Data

```
G05YMF Example Program Data
1                 :: GENID
200  8  1000      :: N,IDIM,ISKIP
5                 :: DN
```

10.3 Program Results

```
 G05YMF Example Program Results


 Value of integral =   1.0410

 First 5 variates for all 8 dimensions
          1       2       3       4       5
 1   0.7197  0.9697  0.4697  0.3447  0.8447
 2   0.5967  0.3467  0.8467  0.4717  0.9717
 3   0.0186  0.7686  0.2686  0.1436  0.6436
 4   0.1768  0.9268  0.4268  0.3018  0.8018
 5   0.7803  0.5303  0.0303  0.1553  0.6553
 6   0.4072  0.1572  0.6572  0.7822  0.2822
 7   0.5459  0.2959  0.7959  0.4209  0.9209
 8   0.3994  0.1494  0.6494  0.0244  0.5244
```