NAG Library Routine Document

G05YBF

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

To generate multi-dimensional quasi-random sequences with a Gaussian or log-normal probability distribution.

2 Specification

```
SUBROUTINE G05YBF(FCALL, SEQ, LNORM, MEAN, STD, ISKIP, IDIM, QUASI,
1                 IREF, IFAIL)
INTEGER           ISKIP, IDIM, IREF(2000), IFAIL
double precision  MEAN(IDIM), STD(IDIM), QUASI(IDIM)
LOGICAL           FCALL, LNORM
CHARACTER*1       SEQ
```

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.

G05YBF generates multi-dimensional quasi-random sequences with a Gaussian or log-normal probability distribution. The sequences are generated in pairs using the Box–Muller method. This means that an even number of dimensions are required by this routine. If an odd number of dimensions are required then the extra dimension must be computed, but can then be ignored.

This routine uses the sequences as described in G05YAF.

4 References

Box G E P and Muller M E (1958) A note on the generation of random normal deviates *Ann. Math. Statist.* **29** 610–611

Brately 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 Parameters

1: FCALL – LOGICAL *Input*

On entry: the type of operation to perform.

FCALL = .TRUE.
The first call for initialization and there is no output via array QUASI.

FCALL = .FALSE.
The sequence has already been initialized by a prior call to G05YBF with FCALL = .TRUE.. Random numbers are output via array QUASI.

2: SEQ – CHARACTER*1 *Input*

On entry: the type of sequence to generate.

SEQ = 'S'
A Sobol sequence.

SEQ = 'N'
A Niederreiter sequence.

SEQ = 'F'
A Faure sequence.

Constraint: SEQ = 'S', 'N' or 'F'.

3: LNORM – LOGICAL *Input*

On entry: indicates whether to create Gaussian or log-normal variates.

LNORM = .TRUE.
The variates are log-normal.

LNORM = .FALSE.
The variates are Gaussian.

4: MEAN(IDIM) – ***double precision*** array *Input*

On entry: MEAN(k) is the mean of distribution for the kth dimension.

5: STD(IDIM) – ***double precision*** array *Input*

On entry: STD(k) is the standard deviation of the distribution for the kth dimension.

Constraint: STD(i) > 0.0, for $i = 1, 2, \ldots,$ IDIM.

6: ISKIP – INTEGER *Input*

On entry: the number of terms in the sequence to skip on initialization.

If SEQ = 'F', ISKIP is not referenced.

Constraint: if SEQ = 'N' or 'S' and FCALL = .TRUE., ISKIP ≥ 0.

7: IDIM – INTEGER *Input*

On entry: the number of dimensions required.

Constraint: $2 \leq$ IDIM ≤ 40 and IDIM must be even.

8: QUASI(IDIM) – ***double precision*** array *Output*

On exit: the random numbers, generated in pairs. That is, on the first call with FCALL = .FALSE., QUASI(k) contains the first quasi-random number for the kth dimension. On the next call QUASI(k) contains the second quasi-random number for the kth dimension, etc..

9: IREF(2000) – INTEGER array *Communication Array*

On entry/exit: workspace used to contain information between calls to the routine. The contents of this array should not be changed.

10: 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.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

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, because for this routine the values of the output parameters may be useful even if IFAIL $\neq 0$ on exit, the recommended value is -1. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

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).

Note: G05YBF may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, ISKIP < 0 if SEQ $=$ 'N' or 'S' and FCALL $=$ .TRUE.,
or IDIM < 2,
or IDIM > 40,
or SEQ $\neq$ 'F', 'N' or 'S',
or STD$(i) \leq 0.0$, for some $i = 1, 2, \ldots,$ IDIM.

IFAIL $= 2$

On entry, the array IREF has not been correctly initialized.

IFAIL $= 3$

The value of ISKIP is too large.

IFAIL $= 4$

There have been too many calls in the sequence.

IFAIL $= 5$

An internal error has occurred within the routines. Please contact NAG.

7 Accuracy

Not applicable.

8 Further Comments

The maximum length of the generated sequences is $2^{29} - 1$, this should be adequate for practical purposes. For more information see G05YAF.

9 Example

This example program calculates the sum of the expected values of the kurtosis of 20 independent Gaussian samples. A quasi-random Faure sequence generator is used.

9.1 Program Text

```
*     G05YBF Example Program Text
*     Mark 20 Release. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      DOUBLE PRECISION ZERO
      PARAMETER        (ZERO=0.0D0)
*     .. Local Scalars ..
      DOUBLE PRECISION SUM, VAL1, VAL2
```

```
      INTEGER          I, IDIM, IFAIL, ISKIP, NTIMES
      LOGICAL          FCALL, LNORM
      CHARACTER        SEQ
*     .. Local Arrays ..
      DOUBLE PRECISION MEAN(20), QUASI(20), STD(20)
      INTEGER          IREF(2000)
*     .. External Functions ..
      DOUBLE PRECISION FUN
      EXTERNAL         FUN
*     .. External Subroutines ..
      EXTERNAL         G05YBF
*     .. Intrinsic Functions ..
      INTRINSIC        DBLE
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G05YBF Example Program Results'
      NTIMES = 10000
      IDIM = 20
      FCALL = .TRUE.
      SEQ = 'F'
      LNORM = .FALSE.
      DO 20 I = 1, IDIM
         MEAN(I) = 2.0D0
         STD(I) = 1.0D0
   20 CONTINUE
*
      IFAIL = 1
      CALL G05YBF(FCALL,SEQ,LNORM,MEAN,STD,ISKIP,IDIM,QUASI,IREF,IFAIL)
*
      WRITE (NOUT,*)
      IF (IFAIL.EQ.0) THEN
         SUM = ZERO
         FCALL = .FALSE.
         DO 40 I = 1, NTIMES
            IFAIL = 0
*
            CALL G05YBF(FCALL,SEQ,LNORM,MEAN,STD,ISKIP,IDIM,QUASI,IREF,
     +                  IFAIL)
*
            SUM = SUM + FUN(IDIM,MEAN,STD,QUASI)
   40    CONTINUE
         VAL1 = SUM/DBLE(NTIMES)
         WRITE (NOUT,99999) 'Calculate value of the integral = ', VAL1
         VAL2 = DBLE(IDIM)*3.0D0
         WRITE (NOUT,*)
         WRITE (NOUT,99999) 'Exact value of the integral     = ', VAL2
      ELSE
         WRITE (NOUT,99998) IFAIL
      END IF
*
99999 FORMAT (1X,A,F8.3)
99998 FORMAT (1X,' ** G05YBF returned with IFAIL = ',I5)
      END
*
      DOUBLE PRECISION FUNCTION FUN(IDIM,MEAN,STD,X)
*     .. Parameters ..
      DOUBLE PRECISION ZERO
      PARAMETER        (ZERO=0.0D0)
*     .. Scalar Arguments ..
      INTEGER          IDIM
*     .. Array Arguments ..
      DOUBLE PRECISION MEAN(IDIM), STD(IDIM), X(IDIM)
*     .. Local Scalars ..
      DOUBLE PRECISION TMP1, TMP2
      INTEGER          J
*     .. Executable Statements ..
      TMP1 = ZERO
      DO 20 J = 1, IDIM
         TMP2 = (X(J)-MEAN(J))/STD(J)
         TMP1 = TMP1 + TMP2*TMP2*TMP2*TMP2
```

```
   20 CONTINUE
      FUN = TMP1
      RETURN
*
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 G05YBF Example Program Results

 Calculate value of the integral =   60.068

 Exact value of the integral     =   60.000
```