NAG Fortran Library Routine Document

G05YFF

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

G05YFF generates a sequence of quasi-random numbers using Sobol's method. It must be preceded by a call to G05YEF to initialize the generator for IDIM dimensions.

2 Specification

```
SUBROUTINE G05YFF (N, QUASI, IREF, IFAIL)
INTEGER            N, IREF(2000), IFAIL
double precision   QUASI(N,*)
```

3 Description

G05YFF generates a sequence of N quasi-random numbers of dimension IDIM using Sobol's method.

4 References

None.

5 Parameters

1: N – INTEGER *Input*

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

Constraint: $N \geq 1$.

2: QUASI(N,$*$) – ***double precision*** array *Output*

Note: the second dimension of the array QUASI must be at least IDIM + 1 if IDIM must be odd and at least IDIM otherwise.

On exit: contains N quasi-random numbers of dimension IDIM.

3: IREF(2000) – INTEGER array *Input/Output*

On entry: contains vital information for the generator.

On exit: updated information for the generation of a further set of quasi-random numbers.

IREF must not be changed between calls of G05YFF.

4: 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 Chapter P01 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, 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.**

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 $= 1$

Incorrect initialization. G05YEF must be called prior to G05YFF and IREF must remain unaltered after this call.

IFAIL $= 2$

There have been too many calls to the generator. It is not able to deliver any more random numbers.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

This examples calls G05YEF and G05YFF to estimate the value of an integral.

9.1 Program Text

```
*     G05YFF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      DOUBLE PRECISION SUM, VSBL
      INTEGER          I, IDIM, IFAIL, ISKIP, NTIMES
*     .. Local Arrays ..
      DOUBLE PRECISION QUASI(2,16)
      INTEGER          IREF(2000)
*     .. External Functions ..
      DOUBLE PRECISION FUN
      EXTERNAL         FUN
*     .. External Subroutines ..
      EXTERNAL         G05YEF, G05YFF
*     .. Intrinsic Functions ..
      INTRINSIC        DBLE
*     .. Executable Statements ..
      WRITE (NOUT,99999) 'G05YFF Example Program Results'
      IDIM = 15
      NTIMES = 5000
      IFAIL = 0
      ISKIP = 20
*
      CALL G05YEF(IDIM,IREF,ISKIP,IFAIL)
*
      SUM = 0.0D0
      DO 20 I = 1, NTIMES
*
         CALL G05YFF(2,QUASI,IREF,IFAIL)
*
         SUM = SUM + FUN(IDIM,QUASI,2)
   20 CONTINUE
      VSBL = SUM/DBLE(2*NTIMES)
```

```
      WRITE (NOUT,99999)
      WRITE (NOUT,99999) 'Value of integral = ', VSBL
      STOP
*
99999 FORMAT (1X,A,F8.4)
      END
*
      DOUBLE PRECISION FUNCTION FUN(IDIM,X,N)
*     .. Scalar Arguments ..
      INTEGER                       IDIM, N
*     .. Array Arguments ..
      DOUBLE PRECISION              X(N,IDIM)
*     .. Local Scalars ..
      DOUBLE PRECISION              TMP, TMP1
      INTEGER                       J
*     .. Intrinsic Functions ..
      INTRINSIC                     ABS
*     .. Executable Statements ..
      TMP = 1.0D0
      TMP1 = 1.0D0
      DO 20 J = 1, IDIM
         TMP = TMP*ABS(4.0D0*X(1,J)-2.0D0)
         TMP1 = TMP1*ABS(4.0D0*X(2,J)-2.0D0)
   20 CONTINUE
      FUN = TMP + TMP1
      RETURN
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 G05YFF Example Program Results

 Value of integral =   1.0273
```