G05AAFP
NAG Parallel Library Routine Document

Note: Before using this routine, please read the Users' Note for your implementation to check for implementation-dependent details. You are advised to enclose any calls to NAG Parallel Library routines between calls to Z01AAFP and Z01ABFP.

1 Description

G05AAFP generates a random number from a uniform distribution in the interval $(0, 1)$. A total of 273 statistically independent generators are available; it is possible to select a particular generator and initialize the seeds for the generator by preceding a call to G05AAFP with a call to G05ABFP. If G05ABFP is not used to select the generator and set the seeds, default values for the generator and seeds are assumed.

2 Specification

```
    DOUBLE PRECISION FUNCTION G05AAFP()
```

3 Data Distribution

3.1 Definitions

Not applicable.

3.2 Global and Local Arguments

Not applicable.

4 Arguments

None.

5 Errors and Warnings

None.

6 Further Comments

Repeatable sequences of random numbers can be generated by calling G05ABFP with the same seeds and generator number before starting each sequence.

Routine G05AAFP may be called without a prior call to Z01AAFP. See the Tutorial for an example.

6.1 Algorithmic Detail

Each basic generator uses a Wichmann–Hill type generator, which is a variant of a multiplicative congruential algorithm to produce random numbers U_i:

$$\begin{aligned} n_{1,i} &= (a_1 \times n_{1,i-1}) \bmod m_1 \\ n_{2,i} &= (a_2 \times n_{2,i-1}) \bmod m_2 \\ n_{3,i} &= (a_3 \times n_{3,i-1}) \bmod m_3 \\ n_{4,i} &= (a_4 \times n_{4,i-1}) \bmod m_4 \end{aligned} \qquad (1)$$

$$U_i = \left(\frac{n_{1,i}}{m_1} + \frac{n_{2,i}}{m_2} + \frac{n_{3,i}}{m_3} + \frac{n_{4,i}}{m_4} \right) \bmod 1.0$$

where $a_1, a_2, a_3, a_4, m_1, m_2, m_3, m_4$ are constant integers for each generator. The constant a_1, a_2, a_3 and a_4 are in the range 112 to 127 and the constants m_1, m_2, m_3 and m_4 are prime numbers in the range 16718909 to 16776971, i.e., close to $2^{24} = 16777216$. The constants have been chosen so that they give good results with the spectral test, see Knuth [1] and Maclaren [2]. The period of each generator would be at least 2^{92} if it were not for common factors between $(m_1 - 1), (m_2 - 1), (m_3 - 1)$ and $(m_4 - 1)$, however, each should still have a period of at least 2^{80}. Further details of the generators can be obtained from NAG and further discussion of the properties of these generators is given in Maclaren [2] who shows that the sequences are essentially independent of one another in the sense used by the spectral test.

7 References

[1] Knuth D E (1981) *The Art of Computer Programming (Volume 2)* Addison-Wesley (2nd Edition)

[2] Maclaren N M (1989) The generation of multiple independent sequences of pseudorandom numbers *Appl. Statist.* **38** 351–359

8 Example

This example generates a series of random numbers on each processor in a 2 by 2 logical grid of processors. **Note:** the default values for the seeds and generator are used on each processor (i.e., there is no call to G05ABFP), so the sequences are identical on each processor.

8.1 Example Text

```
*     G05AAFP Example Program Text
*     NAG Parallel Library Release 2. NAG Copyright 1996.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      INTEGER          M
      PARAMETER        (M=5)
*     .. Local Scalars ..
      INTEGER          I, ICNTXT, IFAIL, IFLAG, J, K, MP, NCR, NP, NRR
      LOGICAL          ROOT
*     .. Local Arrays ..
      DOUBLE PRECISION RVEC(M)
*     .. External Functions ..
      DOUBLE PRECISION G05AAFP
      LOGICAL          Z01ACFP
      EXTERNAL         G05AAFP, Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         DGERV2D, DGESD2D, Z01AAFP, Z01ABFP, Z01BAFP
*     .. Executable Statements ..
      ROOT = Z01ACFP()
      IF (ROOT) WRITE (NOUT,*) 'G05AAFP Example Program Results'
*
      MP = 2
      NP = 2
*
*     Declare the processor grid
*
      IFAIL = 0
      CALL Z01AAFP(ICNTXT,MP,NP,IFAIL)
*
*     Now fill the vector with random numbers
*
      DO 20 I = 1, M
```

```
            RVEC(I) = G05AAFP()
   20    CONTINUE
*
*        Print the local vector
*
         IF (ROOT) THEN
            WRITE (NOUT,99999) 0, 0
            WRITE (NOUT,99998) (RVEC(K),K=1,M)
            WRITE (NOUT,*)
         END IF
*
*        Print the vectors from the other processors
*        First find out ROOT coordinate position
*
         IFAIL = 1
         CALL Z01BAFP(ICNTXT,NRR,NCR,IFLAG)

         IF (ROOT) THEN
*
*                    Receive the vector from processor I,J and print
*
            DO 60 J = 0, NP - 1
               DO 40 I = 0, MP - 1
                  IF (I.NE.NRR .OR. J.NE.NCR) THEN
                     WRITE (NOUT,99999) I, J
                     CALL DGERV2D(ICNTXT,M,1,RVEC,M,I,J)
                     WRITE (NOUT,99998) (RVEC(K),K=1,M)
                     WRITE (NOUT,*)
                  END IF
   40          CONTINUE
   60       CONTINUE
         ELSE
*
*                       Send the local vector to the root processor
*
            CALL DGESD2D(ICNTXT,M,1,RVEC,M,NRR,NCR)
         END IF
*
         IFAIL = 0
         CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
         STOP
*
99999 FORMAT ('Results from processor (',I2,',',I2,' )')
99998 FORMAT (1X,F10.4)
         END
```

8.2 Example Data

None.

8.3 Example Results

```
 G05AAFP Example Program Results
Results from processor ( 0, 0 )
     0.9760
     0.3585
     0.7685
```

```
          0.9150
          0.6040

     Results from processor ( 1, 0 )
          0.9760
          0.3585
          0.7685
          0.9150
          0.6040

     Results from processor ( 0, 1 )
          0.9760
          0.3585
          0.7685
          0.9150
          0.6040

     Results from processor ( 1, 1 )
          0.9760
          0.3585
          0.7685
          0.9150
          0.6040
```