G05ZAF – NAG Fortran Library Routine Document

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

This routine is used to select the underlying mechanism for generating pseudo-random numbers in subsequent calls to other routines in this chapter.

2 Specification

```
SUBROUTINE G05ZAF(CGEN)
CHARACTER*1       CGEN
```

3 Description

There are two underlying mechanisms that can be used by the routines in this chapter for generating pseudo-random numbers. The first mechanism can be selected by calling G05ZAF with its parameter set to 'O' while the second mechanism can be selected by calling G05ZAF with its parameter set to 'W'. Any other parameter setting results in the default mechanism being selected. The statistical properties of the two mechanisms are different; it is therefore NOT recommended that you select different mechanisms from within the same program.

The first mechanism is that used in the NAG Fortran Library (up to Mark 19) and in Release 1 of the NAG Fortran SMP Library. It employs a single generator of cycle length 2^{57} that uses the multiplicative congruential algorithm (see Knuth [1]):

$$b_{i+1} = 13^{13} \times b_i \mathrm{mod} 2^{59},$$

where the integer b_{i+1} is divided by 2^{59} to yield a real value y, which is guaranteed to satisfy

$$0 < y < 1.$$

It is important to note that only one process (or thread) can use routines in this chapter to generate pseudo-random numbers at any one time when this first mechanism is selected. For example, it is not safe to select the first mechanism and then call routines in this chapter from within your own defined 'parallel region'.

The second mechanism uses a variant of the multiplicative congruential algorithm known as the Wichmann–Hill algorithm (see Maclaren [2]) (see the Chapter Introduction) and contains 273 independent generators with cycle length of around 2^{80}. This mechanism is the recommended choice for selection prior to calling other routines in this chapter.

Consult the Users' Note for information on the default mechanism used for your implementation. It may be that you wish to select only the default mechanism and so do not require to call G05ZAF prior to calling other routines in this chapter.

4 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

5 Parameters

1: CGEN — CHARACTER*1 *Input*

On entry: indicates which underlying mechanism is to be used for generating pseudo-random numbers as follows:

CGEN = 'O'

use first (Original) mechanism as described in Section 3,

CGEN = 'W'

use second (Wichmann–Hill) mechanism as described in Section 3.

6 Error Indicators and Warnings

None.

7 Accuracy

Not applicable.

8 Further Comments

The first mechanism for generating pseudo-random numbers is to be used primarily for those wishing to replicate results from code (calling routines in this chapter) previously linked to the NAG Fortran Library (Mark 19 or earlier) or to Release 1 of the NAG Fortran SMP Library. It is only safe to use this mechanism when the executable is run as a single process (thread).

For all other purposes it is recommended that the second mechanism (Wichmann–Hill generators) be selected at the start of any program making calls to routines in this chapter.

9 Example

The example program selects the Wichmann–Hill generators then performs a runs up test for randomness (using G08EAF) on 10000 pseudo-random numbers taken from a uniform distribution between 0 and 1 generated by G05FAF. G08EAF is called 10 times with 1000 observations each time.

9.1 Program Text

```
*     G05ZAF Example Program Text
*     NAG Fortran SMP Library, Release 2.  NAG Copyright 2000.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      INTEGER          M, N, MAXR, LDC, LWRK
      PARAMETER        (M=0,N=10000,MAXR=6,LDC=10,LWRK=34)
      DOUBLE PRECISION TOL
      PARAMETER        (TOL=0.05D0)
*     .. Local Scalars ..
      DOUBLE PRECISION CHI, DF, P
      INTEGER          I, IFAIL, NRUNS
      CHARACTER        CL
*     .. Local Arrays ..
      DOUBLE PRECISION C(LDC,MAXR), EXPECT(MAXR), WRK(LWRK), X(N)
      INTEGER          NCOUNT(MAXR)
*     .. External Subroutines ..
      EXTERNAL         G05CBF, G05FAF, G05ZAF, G08EAF
```

```
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G05ZAF Example Program Results'
      CALL G05ZAF('W')
      CALL G05CBF(6*273+111)
      DO 20 I = 1, 10
         IF (I.EQ.1) THEN
            CL = 'First'
         ELSE IF (I.EQ.10) THEN
            CL = 'Last'
         ELSE
            CL = 'Intermediate'
         END IF
         CALL G05FAF(0.0D0,1.0D0,N,X)
         IFAIL = -1
*
         CALL G08EAF(CL,N,X,M,MAXR,NRUNS,NCOUNT,EXPECT,C,LDC,CHI,DF,P,
     +               WRK,LWRK,IFAIL)
*
         IF (CL.NE.'L' .AND. CL.NE.'l' .AND. IFAIL.NE.0) GO TO 40
*
   20 CONTINUE
*
      IF (IFAIL.EQ.0 .OR. IFAIL.EQ.10) THEN
         WRITE (NOUT,*)
         IF (P.GE.TOL) THEN
            WRITE (NOUT,*) 'Generated sequence passes runs up test'
         ELSE
            WRITE (NOUT,*) 'Generated sequence fails runs up test'
         END IF
      END IF
   40 CONTINUE
      STOP
*
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 G05ZAF Example Program Results

 Generated sequence passes runs up test
```