1 SUMMARY

Generates uniformly distributed pseudo-random numbers. Random numbers are generated in the ranges $0 < \xi < 1$, $-1 < \eta < 1$ and random integers in $1 \leq k \leq N$ where $N$ is specified by the user.

A multiplicative congruent method is used where a 32 bit generator word $g$ is maintained. On each call to the subroutine $g_{n+1}$ is updated to $9228907 \times g_{n} \mod (2^{32})$; the initial 32 bit value of $g$ is '01010101...01'. Depending on the type of random number required the following are computed, $\xi = 2^{-32} g_{n+1}$; $\eta = 2^{-31} g_{n+1} - 1$ or $k = \text{int.part}(\xi N) + 1$.

The subroutine also provides a facility for saving the current value of the generator word and for re-starting with any specified value.

ATTRIBUTES — Version: 1.0.0. Types: FA01A; FA01AD. Original date: August 1967. Origin: R.C.F.McLatchie, Harwell. Remark: FA01A was formerly called FA01AS.

2 HOW TO USE THE PACKAGE

2.1 Argument lists and calling sequences

There are four entries.

(i) to obtain random floating point numbers

The single precision version

R = FA01A(I)

The double precision version

R = FA01AD(I)

$\xi$ is an INTEGER variable which must be set by the user to chose one of two ranges for the random number. Only its sign is significant. If $I \geq 0$ the result is a real value in the range $0 < \xi < 1$ and if $I < 0$ the result is a real value in the range $-1 < \eta < 1$.

FA01A is a REAL valued function subprogram returned set to the random number.

FA01AD is a DOUBLE PRECISION valued function subprogram returned set to the random number.

(ii) to obtain random integers

The single precision version

CALL FA01B(N,K)

The double precision version

CALL FA01BD(N,K)

$N$ is an INTEGER variable which must be set by the user to specify the upper limit $N$ of the range of integers from which the random number is to be taken, i.e. the random number $k$ is chosen from $1 \leq k \leq N$. Restriction: $N$ must be positive.

$K$ is an INTEGER variable which will be set by the subroutine to the random integer $k$. 

All use is subject to licence.

FA01 v 1.0.0
http://www.hsl.rl.ac.uk/ 1 Documentation date: 8th February 2011
(iii) to save the current generator word
   The single precision version
   CALL FA01C(IL, IR)
   The single precision version
   CALL FA01CD(IL, IR)

   IL and IR are INTEGER variables which will be set by the subroutine to the current value of the generator word 
   \[ g = IL \times 16^4 + IR. \]

(iv) to reset the current value of the generator word
   The single precision version
   CALL FA01D(IL, IR)
   The double precision version
   CALL FA01DD(IL, IR)

   IL and IR are INTEGER variables which must be set by the user to the new generator word and should only be set 
   to values previously obtained from FA01C/CD.

2.2 Common
   One common block is used. The common block is
   The single precision version
   COMMON / FA01E/GL, GR
   The double precision version
   COMMON / FA01ED/GL, GR

   where the parameters are given default values by a block data subprogram FA01F/FD.

   GL, GR are DOUBLE PRECISION variables which contain current value of the generator word.

3 GENERAL INFORMATION

   Use of common: The common block FA01E/ED is used, see § 2.2.
   Workspace: none.
   Other subprograms: none.
   Input/Output: none.

4 METHOD

   The generator integer \( g \) is held as a double precision variable \( G \) and revised on every call of FA01A/AD or 
   FA01B/BD. It is initialized as if FA01D/DD is called with

   \[ IL = IR = 21845 = (16^4 - 1)/3 \]

   and on each call is revised as if the statement

   \[ G = DMOD(G \times 9228907., 16. ** 8) \]

   were executed. The output from FA01A/B uses the revised \( G \) as follows

   \[ FA01A(1) = G / 16. ** 8 \]
FA01A(-1)=2*G/16**8-1.
K=INT(FA01A(1)*FLOAT(N))+1

5  EXAMPLE OF USE

We illustrate the use of the package by saving and printing the generator word, printing a random positive real number, printing a random integer in the range 1 to 9, and resetting the generator word.

Suitable code is as follows:

C  SIMPLE  EXAMPLE  OF  USE  OF  FA01  PACKAGE
C  SAVE  AND  PRINT  THE  GENERATOR  WORD
   CALL  FA01C(IL,IR)
   WRITE(6,10)IL,IR
10   FORMAT('  GENERATOR  = ',2I6)
C  PRINT  A  RANDOM  POSITIVE  REAL  NUMBER
   WRITE(6,20)FA01A(1)
20   FORMAT('  RANDOM  POSITIVE  NUMBER  = ',F8.6)
C  PRINT  A  RANDOM  INTEGER  IN  THE  RANGE  1  TO  9
   CALL  FA01B(9,I)
   WRITE(6,30)I
30   FORMAT('  RANDOM  INTEGER  IN  RANGE  1  TO  9  = ',I1)
C  RESET  THE  GENERATOR  WORD
   CALL  FA01D(IL,IR)
C
STOP
END

This produces the following output

GENERATOR  =  21845  21845
RANDOM  POSITIVE  NUMBER  =  0.332617
RANDOM  INTEGER  IN  RANGE  1  TO  9  =  1