Module 21.1: nag_rand_util
Utilities for Random Number Generation

nag_rand_util provides a procedure for setting the seed that is used by random number generating procedures.

Contents

Introduction

1 Setting and Using the Seed

This module is concerned with the setting of the argument `seed`, used by the random number generating procedures of this chapter, to give repeatable or non-repeatable sequences of random numbers from a specified statistical distribution.

All the procedures which generate random numbers rely heavily on a stream of random numbers from the uniform (0,1) distribution which uses a *multiplicative congruential* algorithm:

$$n_i = (13^{13} \times n_{i-1}) \mod(2^{59})$$

(see Knuth [1]). The current state of this stream is stored in an argument `seed`, which is a structure of the derived type `nag_seed_wp` and it supplies the initial value n_0.

Before any of the generating procedures are called, the `seed` must be initialized by a call to `nag_rand_seed_set`. It is then updated by each subsequent call to a generating procedure. For example, the following code fragments show how to generate a non-repeatable sequence of random numbers from a uniform distribution:

```
call nag_rand_seed_set( seed )
 . . .
do
   . . .
  x = nag_rand_uniform( seed )
   . . .
end do
```

You should *not* call `nag_rand_seed_set` again to re-initialize the `seed` unless you wish to start a new stream of random numbers.

The procedure `nag_rand_seed_set` has one optional argument which may be used to ensure that the seed is initialized in a repeatable way, so that the same stream of random numbers is generated each time your program is run. This is especially useful during the development of a program. If the optional argument is not supplied to `nag_rand_seed_set`, the initialization depends on the system clock and is not repeatable; this is recommended for proper simulations.

Procedure: nag_rand_seed_set

1 Description

`nag_rand_seed_set` sets the seed that is used by the procedures in modules `nag_rand_contin` (21.2) amd `nag_rand_discrete` (21.3) to give repeatable or non-repeatable sequences of random numbers. This, however, depends on whether `k` is present or not in the argument list. If `k` is present, subsequent use of the seed will result in a repeatable sequence of random numbers; otherwise, a non-repeatable sequence of random numbers is returned.

It should be noted that this procedure *must* be invoked before calling any procedures in the module `nag_rand_contin` (21.2) or `nag_rand_discrete` (21.3) .

2 Usage

```
USE nag_rand_util

CALL nag_rand_seed_set(seed  [, optional arguments])
```

3 Arguments

3.1 Mandatory Argument

seed — type(nag_seed_*wp*), intent(out)
> *Output:* an initial value of the seed (see the Module Introduction).
> *Note:* the components of the `seed` are private and are not accessible.

3.2 Optional Argument

k — integer, intent(in), optional
> *Input:* determines the initial value of `seed`, resulting in a repeatable sequence of random numbers.
> *Default:* the initial value of `seed` is calculated from the system clock, resulting in a non-repeatable sequence of random numbers.

4 Error Codes

None.

5 Examples of Usage

Illustrations of the use of this procedure appear in the examples given in the module documents for `nag_rand_contin` (21.2) and `nag_rand_discrete` (21.3).

Derived Type: **nag_seed_**wp

Note. The names of derived types containing real/complex components are precision dependent. For double precision the name of this type is `nag_seed_dp`. For single precision the name is `nag_seed_sp`. Please read the Users' Note for your implementation to check which precisions are available.

1   Description

The derived type `nag_seed_`wp stores the seed which is used to generate a stream of random numbers from the basic uniform (0,1) distribution; it also stores additional information required by the procedures that generate random numbers from a Normal distribution.

The components of this type are private.

Structures of this type *must be initialized* by a call to `nag_rand_seed_set` before being passed to any other procedure.

Distinct structures of this type may be used to generate distinct independent streams of random numbers.

2   Type Definition

```
type nag_seed_wp
  private
  .
  .
  .
end type nag_seed_wp
```

3   Components

In order to reduce the risk of accidental data corruption the components of this type are private and may not be accessed directly.

References

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