Chapter g05 – Random Number Generators

1. Scope of the Chapter

This chapter covers the generation of sequences of independent pseudo-random numbers from various distributions.

2. Background

A sequence of *pseudo-random numbers* is a sequence of numbers generated in some systematic way such that its statistical properties are as close as possible to those of true random numbers: for example negligible correlation between consecutive numbers. The most common method used is a *multiplicative congruential* algorithm defined as:

$$n_i = (an_{i-1}) \bmod m. \quad (1)$$

The integers n_i are then divided by m to give uniformly distributed random numbers in the range (0,1).

The NAG generator uses the values $a = 13^{13}$ and $m = 2^{59}$; for further details see nag_random_continuous_uniform (g05cac). This generator gives a *cycle length* (i.e., the number of random numbers before the sequence starts repeating itself) of 2^{57}. A good rule of thumb is never to use more numbers than the square root of the cycle length in any one experiment as the statistical properties are impaired. For closely related reasons breaking numbers down into their bit patterns and using individual bits may cause trouble.

The sequence given in (1) needs an initial value n_0, known as the *seed*. The use of the same seed will lead to the same sequence of numbers. One method of obtaining the seed is to use the real-time clock, this will give a non-repeatable sequence. It is important to note that the statistical properties of the random numbers are only guaranteed within sequences and not between sequences. Repeated initialization will thus render the numbers obtained less rather than more independent.

It may also be useful to store the state of the generator at a particular point and restart from that point later; facilities for this are provided.

Random numbers from other distributions may be obtained from the uniform random numbers by the use of transformations, rejection techniques and for discrete distributions table based methods.

Transformations can be based on the fact that if a continuous random variable x has cumulative distribution function (CDF) $F(x)$ then a random observation can be generated by $F^{-1}(u)$, where u is a uniform (0,1) variate. For example, the CDF of the *logistic* distribution is: $1 - [1 + \exp((x-a)/k)]^{-1}$, so variates can be generated using $a + k\log(u/(1-u))$. Another example is the *Weibull* distribution which can be generated by $b(-\log u)^{1/a}$. Alternatively the relationship between distributions can be used, for example an F variate can easily be obtained from a beta variate and a Student's t distribution with n degrees of freedom can be generated as $v\sqrt{(n/w)}$ where v has a Normal distribution and w has a gamma distribution. In some cases the distribution is just a special case of another distribution, for example the χ^2 distribution is a special case of the gamma distribution.

Rejection techniques are based on the ability to easily generate random numbers from a distribution (called the envelope) similar to the distribution required. The value from the envelope distribution is then accepted as a random number from the required distribution with a certain probability; otherwise it is rejected and a new number generated from the envelope distribution.

For discrete distributions the cumulative probabilities, $P_i = Prob(x \le i)$ can be stored in a table then, given a uniform (0,1) random variate, u, the table is searched for i such that $P_{i-1} < u \le P_i$. The returned value i will have the required distribution. The table searching can be made more efficient by using an index, see Ripley (1987). The table and its index are stored in a reference vector, the effort to construct the reference may be considerable but the methods are very efficient when many values can be generated from a single reference vector.

Whilst more efficient methods are provided for low-dimensional quadrature in Chapter d01, it should be observed that the uniform generator provides the basic tool for Monte Carlo integration.

3. References

Dagpunar J (1988) *Principles of Random Variate Generation* Oxford University Press.
Morgan B J T (1984) *Elements of Simulation* Chapman and Hall.
Ripley B D (1987) *Stochastic Simulation* Wiley.

4. Available Functions

4.1. Continuous Distributions

Exponential distribution	`g05dbc`
Normal distribution	`g05ddc`
Uniform distribution (0,1)	`g05cac`
Uniform distribution (a,b)	`g05dac`
Beta distribution	`g05fec`
Gamma distribution	`g05ffc`
Multivariate Normal distribution	`g05eac + g05ezc`

4.2. Discrete distributions

Uniform distribution (m,n)	`g05dyc`
Generate discrete variate from reference vector	`g05eyc`
The following functions set up a reference vector for use by g05eyc:	
Poisson distribution	`g05ecc`
Binomial distribution	`g05edc`
General discrete distribution	`g05exc`

4.3. Other random structures

Random permutation	`g05ehc`
Random sample	`g05ejc`

4.4. Time series simulation

ARMA model	`g05hac`

4.5. Utility functions

Initialise generator, repeatable value	`g05cbc`
Initialise generator, non-repeatable value	`g05ccc`
Restore generator	`g05cgc`
Save generator	`g05cfc`