



Skipping Ahead the Mersenne Twister Random Number Generator

The ability to generate multiple streams of pseudorandom numbers is important since it allows programmers to develop parallel applications. The two most obvious and straightforward ways to achieve this, by using different initial seeds or by using different underlying generators for the streams, are unattractive. This is because in the first case the statistically random properties can be guaranteed only *within* each stream and not *between* them and the second case relies on a sufficient number of different generators being available and causes non-repeatable results when using a different number of streams.

Two alternative techniques for creating multiple streams are based on taking subsets of values from a single sequence that would be produced by conventional use of a generator; these are called leap-frog and skip-ahead in the NAG Library.

Leap-frogging involves priming a generator to return every n^{th} value starting from the k^{th} position in the sequence. However, with some generators, this technique can lead to streams with poor statistical properties and can also increase the time taken to generate each variate.

The recommended way to create multiple streams efficiently and with good statistical properties using the NAG Library is to use the skip-ahead technique. With this a single sequence is partitioned into k non-overlapping streams, each a distance of n variates apart (where n is an upper bound on the number of variates required from each stream).

Before Mark 23 it was not possible to skip-ahead using the recommended base generator, the Mersenne Twister, which is favoured because of its very long period of repetition (approximately 2^{19937}), fast implementation and good statistical properties; at Mark 23 this is now possible. In addition, because of this generator's very long period a new routine, G05KKF, has been added which provides access to an entire sequence by allowing users to specify a power of 2 places to skip-ahead, thus providing the possibility of using multiple streams each of which can generate huge numbers of variates without overlapping.