



马特赛旋转 (Mersenne Twister) 随机数生成器中的 Skipping Ahead 技术

如何能够多流 (stream) 的产生伪随机数 (pseudorandom) 是相当重要的一个课题，因为若其可行的话，程序员就能开发并行的应用程序。两个最显而易见与直觉的方法就是：使用不同的随机数默认 seed，或者是在不同 stream 中使用不同的随机数生成器。然而这么做都不具任何的统计意义。原因是，第一种方法只能保证在个别流中随机数的统计特性，并无法确保各个流彼此间的统计性；第二个方法则是当在使用不同数量的流时，必须要仰赖足够数量的产生器，而且其无法取得重复的结果。

有两种替代技术可行。就是在多流中让每个流分别自传统产生器所产生的序列中，取得各个不同的子集合，这个技术在 NAG 算法库中被称作 leap-frog (蛙跳式) 与 skip-ahead (跳前) 方法。

Leap-frogging 方法在开始时透过产生器自序列中回传自 k 位置起始的所有 n 个数量的值。然而，对许多的产生器而言，这个方法有可能会产生不佳的统计特性，而且会增加产生每一个变量所花的时间。

我们建议能有效率的建构多流且可得到较佳统计特性的方法，便是使用 NAG 算法库的 skip-ahead 技术。此方法简单来说就是，将单一的序列分成 k 个不重迭的流，每个以 n 变数距离分隔 (n 是每个流所需变数数量的上限)。

在 NAG 算法库 23 版之前，我们并无法在马特赛旋转 (Mersenne Twister) 随机数生成器中使用 skip-ahead 技术。Mersenne Twister 随机数生成器是最适合的随机数生成器了，因为它具有非常长的重复周期 (大约是 2^{19937})、能够快速实现，且具有很好的统计特性，如今我们增加这样的功能。由于具有很长周期的 G05KKF 新函数已加入到我们的算法库中，可以让使用者以 2 的 n 次方的 skip-ahead 方式取得全部序列中的位置，也因此可以让使用者在采用多流进行开发时，可以确保每个流中都能产生不会重迭的大量随机数资料。