

NAG の乱数生成器

6 種類の乱数生成器を提供している NAG 数値計算ライブラリ

NAG 数値計算ライブラリでは、現在以下の 6 種類の乱数生成器が提供されています。

- NAG 基本生成器（線形合同法による乱数生成器）
- Wichmann-Hill I 生成器（4 つの線形合同法を組み合わせた乱数生成器）
- Wichmann-Hill II 生成器（4 つの線形合同法を組み合わせた乱数生成器、大きな入力引数に対応）
- Mersenne Twister 生成器（Twisted Generalized Feedback Shift Register 乱数生成器）
- L' Ecuyer MRG32k3a 生成器（二つの多重再帰生成器を組み合わせた乱数生成器）
- ACORN 生成器（特殊な多重再帰生成器）

これらの乱数生成器は、C/C++/Fortran/Java 等様々な環境で使用することができます。また、並列環境や Xeon Phi™ コプロセッサを用いた環境でも使用することができます。

NAG の提供している乱数生成法は様々なアプリケーションでの実利用において十分な質と性能を発揮します。またルーチンのインターフェース（仕様）は今後別のタイプの生成器を追加することのできる柔軟性の高いものとなっています。

乱数生成の分野は発展スピードが速くこれまでに様々な乱数生成法が提案されてきていますが、どの乱数生成器を使用するかは以下に示すいくつかのポイントで判断することができます。

- どれだけ簡単に実装できるか
- 周期（長いほど良い）
- 乱数生成器の統計的性質
- 乱数生成速度
- どれだけ簡単に並列化できるか

これらの点について、NAG の各生成器がどのような特徴をもっているか、その長所と短所をまとめた表を以下に示します。

なお NAG 製品をご利用いただく場合は、「実装」及び「並列化」の 2 点については（NAG にて既に行われているため）ユーザの皆様は気になさる必要がございません。

乱数生成器	長所	短所
NAG 基本生成器	線形合同法の乱数生成器のため乱数生成の数式を理解しやすい。 Wichmann-Hill I 生成器よりも乱数生成速度がやや速く、周期もやや長い。	Diehard テストをクリアしたが TestU01 の Big Crush テストをクリアしていない。 最近の生成器よりも周期が短い。 統計的な性質が悪くなりやすい。 良質な線形合同法の生成器ではあるが、もはや最前線ではない。
Wichmann-Hill I 生成器	273 個の独立した乱数生成器で構成されているため独立したストリームを生成できる。 良質な生成器である。	最近の生成器よりも周期が短い。 統計的な性質が悪くなりやすい。
Wichmann-Hill II 生成器	MRG32k3a と比較して skip-ahead が速く簡単に行える。 シードの設定が容易である。 良質の万能な生成器である。 TestU01 の Big Crush テストをクリア。	特になし。
Mersenne Twister 生成器	周期が長い。 乱数生成速度が速い。 TestU01 の Big Crush テストをクリア。	シードの設定が困難である。 並列化が難しい。
L' Ecuyer MRG32k3a 生成器	乱数生成速度が速い。 並列化が容易である。(特に GPU 環境で) 良質の万能な生成器である。 TestU01 の Big Crush テストをクリア。	特になし。
ACORN 生成器	入力引数の与え方によっては長い周期が実現可能である。 統計的な性質が良い。 TestU01 の Big Crush テストをクリア。	NAG の他の生成器と比べて処理速度が遅い。

※) 乱数生成器の性質をテストするためのいくつかのテストスイートが提案されていますが、このうちで最初に世の中に広く受け入れられたものに Diehard という一連のテストがあります。また最近では TestU01 という一連のテストが広く受け入れられています。この TestU01 には Small Crush と呼ばれる主要なテストのみ含んだ一連のテストと Big Crush と呼ばれる TestU01 の全てのテストを含んでいる一連のテストがあります。

NAG 数値計算ライブラリの乱数生成器の詳細については [Chapter G05](#) を参照下さい。