Модел. и анализ информ. систем. Т. **19**, № **2** (2012) 145–162

УДК 519.246

Генерация случайных чисел и параллельных потоков случайных чисел для расчетов Монте-Карло

Бараш Л. Ю., Щур Л. Н.

Институт теоретической физики им. Л.Д. Ландау Российской академии наук

e-mail: barash@itp.ac.ru

получена 25 ноября 2011 года

Ключевые слова: генераторы случайных чисел, параллельные системы и вычисления, нелинейные динамические системы

Рассматриваются современные методы и пакеты программ генерации псевдослучайных чисел высокого качества, а также генерации параллельных потоков случайных чисел, для использования в расчетах Монте-Карло. Рассмотрено свойство равнораспределения вероятности для генераторов вида Multiple Recursive Generators и параметры, при которых это свойство выполняется на длине до логарифма размера сетки.

1. Введение

Генераторы случайных чисел являются обязательной составной частью программного обеспечения операционных систем. Большинство таких генераторов основано на линейно-конгруэнтном методе. Генераторы случайных чисел также используются в численном моделировании и в устройствах шифрования. Мы не касаемся в нестоящей статье последнего случая. Для задач моделирования используются генераторы, основанные на упоминавшемся линейно-конгруэнтном методе и на алгоритме сдвиговых регистров, а также их модификации и комбинации.

Проблема заключается в том, что на каждом витке качественного развития вычислительных систем обнаруживаются дефекты последовательности псевдослучайных чисел. Так, в 1967 году был выявлен (Coveyou and MacPherson, 1967) существенный дефект линейно-конгруэнтного метода, его не следует использовать в приложениях, имеющих дело со случайными векторами в n-мерном пространстве при $n > 1$ из-за плохой геометрической решетчатой структуры генерируемых векторов, которые все расположены на множестве параллельных гиперплоскостей. В 1992 году был выявлен дефект генератора типа сдвиговый регистр; его использование в алгоритмах, где есть типичный размер геометрической структуры, а ее рост завершается в результате сравнения со случайным числом, приводит к гигантским систематическим ошибкам (Ferrenberg и др., 1992).

При проведении расчетов методом Монте-Карло на суперкомпьютерных системах производительностью более сотни терафлоп ожидается, что в существующих генераторах случайных чисел проявятся дефекты. Необходим поиск новых методов генерации случайных чисел и реализация эффективных алгоритмов и методов для использования более 10^{15} чисел в одном расчете.

Еще одна проблема возникает при проведении расчетов на параллельных вычислительных системах, включая гибридные суперкомпьютерные системы. В таких расчетах необходим метод генерации некоррелированных параллельных потоков случайных чисел, а также реализации таких методов в виде программного обеспечения и библиотек генераторов. Эта задача до сих пор не решена.

В статье мы анализируем в деталях состояние исследований в области генерации случайных чисел и параллельных потоков случайных чисел и анализируем возможные подходы к решению вышеуказанных проблем.

2. Требования к генераторам случайных чисел

Генераторы случайных чисел (RNG) и их реализации в библиотеках подпрограмм должны удовлетворять ряду существенных требований:

(1) *Статистическая устойчивость.* Значения на выходе идеального RNG должны быть равномерно распределены, а корреляции должны отсутствовать. Другими словами, все подпоследовательности фиксированной длины должны иметь одну и ту же вероятность появления в последовательности, выдаваемой генератором. С практической точки зрения, последовательность псевдослучайных чисел должна пройти набор статистических тестов на равномерное распределение и независимость.

(2) *Непредсказуемость.* В основном это свойство важно для криптографических алгоритмов. Требование состоит в невозможности надежно предсказать значение a_{n+1} по $(a_0, \ldots, a_n)$ при помощи какого-либо полиномиального алгоритма. Здесь a_n — значение на выходе генератора.

(3) *Длинный период.* Период генератора должен быть достаточно большим, чтобы не быть исчерпанным за месяцы компьютерного времени. Численный эксперимент на суперкомпьютере может задействовать 10^9 случайных чисел в секунду в течение многих часов (или месяцев в случае, например, вычислений КХД), поэтому $10^{13} - 10^{16}$ случайных чисел могут вносить вклад в результат вычислительного эксперимента. Для большинства генераторов использование небольшой части периода T предпочтительнее с точки зрения статистических свойств, чем использование периода целиком. Распространенным эмпирическим правилом, хотя и недостаточно универсальным, является использование не более $\sqrt{T}$ чисел.

(4) *Эффективность.* Должна существовать эффективная реализация RNG с точки зрения скорости вычислений и использования оперативной памяти.

(5) *Наличие теории.* Свойства генератора, такие как длина периода, часто могут быть найдены точно с помощью аналитических методов. Для RNG чрезвычай-

но желательно понимать свойства генерируемой псевдослучайной последовательности, а не рассчитывать только лишь на эмпирические тесты. По этой причине хороший генератор должен быть основательно проанализирован теоретически.

(6) *Воспроизводимость.* Часто полезно повторить ту же самую последовательность псевдослучайных чисел, которая была использована в предыдущем запуске приложения. Большинство генераторов псевдослучайных чисел выдает воспроизводимые последовательности, в отличие от последовательностей, генерируемых физическими устройствами. Например, случайный бит можно получить, приготовив кубит $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$ и спроектировав его на $\{|0\rangle, |1\rangle\}$, однако последовательность таких случайных битов будет принципиально невоспроизводима.

(7) *Переносимость* — это возможность генерировать одну и ту же последовательность псевдослучайных чисел на разных программно-аппаратных платформах.

(8) *Пропуск кусков.* Для любого большого r должна быть возможность быстрого вычисления значения s_{n+r} напрямую из предыдущего s_n, без генерации промежуточных состояний. Здесь s_n — состояние генератора. Это свойство особо важно для генерации параллельных потоков случайных чисел (см. ниже).

(9) *Правильная инициализация.* Важно, чтобы короткие последовательности, выдаваемые RNG, не имели корреляций; это важная задача, которую необходимо специально решить. Для ряда генераторов добиться этого непросто, особенно если состояния генератора обладают большим объемом информации.

3. Существующие методы генерации случайных чисел

Как отмечалось во введении, наиболее широко используемые методы генерации случайных чисел могут быть поделены на два основных класса: линейно-конгруэнтный метод и метод сдвиговых регистров.

А) Линейно-конгруэнтный метод

Линейно-конгруэнтный метод (ЛКМ) был предложен Лемером (Lemer, 1951). Последовательность псевдослучайных чисел $(x_0, x_1, \ldots, x_n, \ldots)$ вычисляется после задания начального значения x_0 по формуле

$$x_{n+1} = (ax_n + c) \,(\mathrm{mod}\ M).$$

Здесь a — множитель, c — приращение, M — модуль. Очевидно, максимально возможный период такой последовательности не может превосходить модуля M. Справедливо следующее утверждение (Кнут, 2000).

Теорема. *Длина периода линейной конгруэнтной последовательности равна M тогда и только тогда, когда*

- *c и M взаимно просты*

- $p|(a-1)$ *для любого простого* $p|M$
- $4|(a-1)$, *если* $4|M$.

Линейно-конгруэнтный метод имеет два существенных недостатка. Первый, это алгоритмическое ограничение на максимальный период, который не может превосходить машинную длину целого числа (при этом последовательность длины $2^{32} \approx 4 \cdot 10^9$ исчерпывается на современных рабочих станциях в течение нескольких секунд). Второй недостаток заключается в том, что ЛКМ не следует использовать в приложениях, имеющих дело со случайными векторами в многомерном пространстве, поскольку соответствующие точки, получаемые из псевдослучайной последовательности на выходе ЛКМ, будут лежать в пространстве меньшей размерности (Кнут, 2000; Coveyou, MacPherson, 1968; Tezuka, 1995).

Б) Генераторы, основанные на сдвиговом регистре

Рассмотрим последовательность

$$x_n = (a_1 x_{n-1} + \cdots + a_k x_{n-k}) \pmod 2.$$

Характеристическим полиномом этой последовательности является $P(z) = z^k - a_1 z^{k-1} - \cdots - a_k$. Это линейно-рекуррентное соотношение в поле $\mathbb{Z}_2$, состоящем из двух элементов, нуля и единицы. Такое рекуррентное соотношение называется сдвиговым регистром и имеет период длины $p = 2^k - 1$ тогда и только тогда, когда P является примитивным полиномом (см. Golomb, 1967). Генератором на сдвиговом регистре называется генератор с выходной последовательностью

$$u_n = \sum_{i=1}^{L} x_{ns+i-1} 2^{-i},$$

где размер шага s и длина слова L — целые положительные числа.

Отсюда видно, что генераторы, основанные на сдвиговом регистре (ГСР), быстрые и обладают гигантским периодом при условии правильного выбора примитивных триномов, лежащих в основе таких генераторов. Поэтому они особенно хорошо подходят для приложений, требующих большого количества случайных чисел. Однако в генераторах этого класса были обнаружены корреляции, которые могут привести к систематическим ошибкам в вычислениях Монте-Карло (Ferrenberg и др., 1992; Vattulainen и др., 1994; Schmid, Wilding, 1995; Shchur, Bloete, 1997; Grassberger, 1993).

В) Современные модификации и обобщения

Современные модификации и обобщения методов ЛКМ и ГСР имеют намного лучшие статистические свойства. В качестве примеров можно привести генератор Mersenne Twister (Matsumoto, Tishimura, 1998), комбинированные ЛКМ-генераторы (L'Ecuyer, 1999), комбинированные генераторы из сдвиговых регистров (L'Ecuyer, 1996; L'Ecuyer, 1999), а также генераторы, основанные на параллельной эволюции ансамбля преобразований тора (Barash, Shchur, 2006; Barash, Shchur, 2011; Barash, 2011). Сравним основные изученные свойства этих генераторов. В генераторе Mersenne Twister (MT19937) наблюдается точное равнораспределение вероятности в

размерности 623, период генератора порядка 10^{6001}. Для комбинированного ЛКМ-генератора MRG32K3A точное равнораспределение не выполняется, но при помощи изучения так называемых коэффициентов доброкачественности (figures of merit) утверждается, что он ведет себя близко к равнораспределению в размерности 45, его период порядка 10^{57}. Для комбинированного генератора из четырех сдвиговых регистров LFSR113 точное равнораспределение наблюдается в размерности порядка 30, его период порядка 10^{34}. Статистические тесты выявляют дополнительные корреляции в выходных последовательностях MT19937 и LFSR113, но найденные корреляции связаны исключительно с тем, что выходные биты данных генераторов имеют линейную структуру по построению, что не является серьезным недостатком данных генераторов (L'Ecuyer, Simard, 2007). Для генераторов, основанных на параллельной эволюции ансамбля преобразований тора, которые будут подробнее рассмотрены в разделах 4 и 5, точное равнораспределение наблюдается в размерности порядка 30, в то же время, генератор ведет себя близко к равнораспределению в размерности примерно 350, период типичного такого генератора GM55.4 порядка 10^{31}.

4. Метод генерации случайных чисел, основанный на параллельной эволюции ансамбля преобразований тора

В работах (Barash, Shchur, 2006; Barash, Shchur, 2011; Barash, 2011) предложено конструировать генератор случайных чисел на основе параллельной эволюции ансамбля преобразований тора. Состояние генератора состоит из значений $x_i^{(n-1)}, x_i^{(n-2)} \in \{0, 1, \ldots, g-1\}$, $i = 0, 1, \ldots, s-1$. Функция перехода генератора определяется рекуррентным соотношением

$$x_i^{(n)} = kx_i^{(n-1)} - qx_i^{(n-2)} \pmod{g}, \quad i = 0, 1, \ldots, s-1. \tag{1}$$

Значения $x_i^{(n)}$, $i = 0, 1, \ldots, s-1$ можно рассматривать как абсциссы s точек $\left(x_i^{(n)}, y_i^{(n)}\right)^T$, $i = 0, 1, \ldots, s-1$, лежащих на решетке $g \times g$ на двумерном торе. Тогда каждое рекуррентное соотношение описывает динамику x-координаты точки двумерного тора:

$$\begin{pmatrix} x_i^{(n)} \\ y_i^{(n)} \end{pmatrix} = M \begin{pmatrix} x_i^{(n-1)} \\ y_i^{(n-1)} \end{pmatrix} \pmod{g}, \tag{2}$$

где матрица $M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix}$ — матрица с целыми элементами, при этом $k = \operatorname{Tr} M$, $q = \det M$, где $\operatorname{Tr} M$ и $\det M$ обозначают след матрицы M и определитель матрицы M соответственно (Barash, Shchur, 2006; Grothe, 1987; Niederreiter, 1995). Действительно, из (2) следует, что

$$\begin{aligned} kx_i^{(n-1)} - qx_i^{(n-2)} &= (m_1 + m_4)x_i^{(n-1)} - (m_1m_4 - m_2m_3)x_i^{(n-2)} = \left(x_i^{(n)} - m_2y_i^{(n-1)}\right) + \\ &+ m_4x_i^{(n-1)} - m_1m_4x_i^{(n-2)} + m_2m_3x_i^{(n-2)} = x_i^{(n)} - m_2\left(y_i^{(n-1)} - m_3x_i^{(n-2)}\right) + \\ &+ m_4\left(x_i^{(n-1)} - m_1x_i^{(n-2)}\right) = x_i^{(n)} - m_2m_4y_i^{(n-2)} + m_2m_4y_i^{(n-2)} = x_i^{(n)} \pmod{g}. \end{aligned}$$

Итак, рекуррентное соотношение (1) тесно связано с так называемым матричным генератором псевдослучайных чисел, изученным в (Кнут, 2000; Grothe, 1987; Niederreiter, 1995).

Осталось определить выходную функцию генератора G: $L^s \to \{0, 1, \ldots, 2^s - 1\}$, при помощи которой вычисляется каждый элемент последовательности $\{a^{(n)}\}$, которая будет на выходе генератора. Пусть $\alpha_i^{(n)}$ обозначает 0 или 1 в зависимости от того, $x_i^{(n)} < g/2$ или $x_i^{(n)} \geq g/2$, т.е. $\alpha_i^{(n)} = \lfloor 2x_i^{(n)}/g \rfloor$. Тогда искомое число на выходе генератора будет следующим: $a^{(n)} = \sum_{i=0}^{s-1} \alpha_i^{(n)} \cdot 2^i$. Другими словами, $a^{(n)}$ — это s-битовое целое число, которое состоит из первых битов чисел $x_0^{(n)}, x_1^{(n)}, \ldots, x_{s-1}^{(n)}$. Мы видим, что построенный таким образом RNG содержит много скрытой информации. Например, для $g = 2^m$, $s(m-1)$ битов вектора $\begin{pmatrix} x_i^{(n)} \\ y_i^{(n)} \end{pmatrix}$ являются скрытыми переменными; это именно те биты, которые не участвуют в построении выходного значения $a^{(n)}$. В более общем случае выходной функцией генератора является

$$a^{(n)} = \sum_{i=0}^{s-1} \left\lfloor 2^v x_i^{(n)}/g \right\rfloor \cdot 2^{iv}, \tag{3}$$

где v бит берется из каждого рекуррентного соотношения. Для того чтобы можно было генерировать 32-битные псевдослучайные числа, необходимо, чтобы выполнялось $sv \geq 32$. Последовательность битов $\left\{ \left\lfloor 2^v x_i^{(n)}/g \right\rfloor \right\}$, где i фиксировано и $\{x_i^{(n)}\}$ генерируется при помощи соотношения (2), будем называть потоком v-битовых блоков, генерируемым матрицей M. Пары чисел $x_i^{(0)}, x_i^{(1)} \in \mathbb{Z}_g$ для соотношения (1) и $x_i^{(0)}, y_i^{(0)} \in \mathbb{Z}_g$ для соотношения (2) представляют собой начальные пары чисел для потоков из v-битовых блоков, генерируемых при помощи (1) и (2) соответственно. Рассмотрим множество допустимых начальных пар чисел, содержащее все пары, такие что хотя бы одно из двух чисел не делится на p. Выбор начальной пары случайным образом из равномерного распределения по множеству допустимых начальных пар определяет меру вероятности для выходных последовательностей потока из v-битовых блоков. Такие вероятности будут рассматриваться ниже в следующем разделе.

Если $g = p \cdot 2^t$, где p — простое число, то параметры k и q выбираются так, чтобы характеристический полином $f(x) = x^2 - kx + q$ был примитивен в поле $\mathbb{Z}_p$. Примитивность характеристического полинома гарантирует максимальный возможный период $p^2 - 1$ выходной последовательности для $g = p$. Легко видеть, что использование $g = p \cdot 2^t$ вместо $g = p$ не уменьшает значение периода.

Существует сравнительно простой алгоритм «пропуска кусков» для данного генератора, т.е. быстрого вычисления $x^{(n)}$ в (1), используя только значения $x^{(0)}$ и $x^{(1)}$, для любого большого n. Действительно, если $x^{(2n)} = k_n x^{(n)} - q_n x^{(0)} \pmod g$, то $x^{(4n)} = (k_n^2 - 2q_n)x^{(2n)} - q_n^2 x^{(0)} \pmod g$. Как было уже упомянуто в (Barash, Shchur, 2006), это помогает инициализировать генератор. Чтобы инициализировать все s рекуррентных соотношений, используются следующие начальные условия: $x_i^{(0)} = x^{(iA)}$, $x_i^{(1)} = x^{(iA+1)}$, $i = 0, 1, \ldots, s-1$. Здесь A — величина порядка $(p^2-1)/s$. Мы протестировали реализации с различными значениями величины A и нашли во всех случаях, что конкретный выбор A не был важен для статистических свойств. Хотя бы одно из чисел $x_i^{(0)}$ и $x_i^{(1)}$ не должно делиться на p, чтобы избежать коротких циклов и, в частности, цикла, состоящего из одних нулей. В результате такой инициализации

все s начальных точек принадлежат одной и той же орбите тора периода p^2-1, в то время как минимальное расстояние A между начальными точками вдоль орбиты выбрано очень большим.

В следующей таблице указаны параметры генераторов, основанных на автоморфизме двумерного тора, предложенных в (Barash, Shchur, 2011; Barash, 2011). Параметры подбирались таким образом, чтобы для генератора выполнялось свойство равнораспределения вероятностей в размерности порядка логарифма величины g, а также чтобы период генератора был максимально возможным (Barash, 2011).

Генератор	K	q	g	v	Период
GM19	15	28	$2^{19}-1$	1	$2.7\cdot 10^{11}$
GM31	11	14	$2^{31}-1$	1	$4.6\cdot 10^{18}$
GM61	24	74	$2^{61}-1$	1	$5.3\cdot 10^{36}$
GM29.1	4	2	$2^{29}-3$	1	$2.8\cdot 10^{17}$
GM55.4	256	176	$16(2^{51}-129)$	4	$\geq 5.1\cdot 10^{30}$
GM58.1	8	48	$2^{29}(2^{29}-3)$	1	$\geq 2.8\cdot 10^{17}$
GM58.3	8	48	$2^{29}(2^{29}-3)$	3	$\geq 2.8\cdot 10^{17}$
GM58.4	8	48	$2^{29}(2^{29}-3)$	4	$\geq 2.8\cdot 10^{17}$

5. Геометрические и статистические свойства

В работе (Barash, Shchur, 2006) найдена связь между статистическими свойствами, результатами теста на случайное блуждание и геометрическими свойствами гиперболических автоморфизмов двумерного тора. Таким образом, рассматривались преобразования (2) для случая $q=\det M=1$. В частности, в (Barash, Shchur, 2006) показано, что вероятность, что 0000 — последовательность длины четыре из старших битов, генерируемых таким преобразованием, зависит только от следа k матрицы и равна $P=P_0k^2/(k^2-1)$ для четного k, где $P_0=1/16$. Если же k нечетно, то все такие последовательности из четырех битов равновероятны. В этом случае вероятность того, что 00000 — последовательность длины пять из старших битов, генерируемых таким преобразованием, равна $P=P_0(1+1/(3k^2-6))$ для нечетного k, где $P_0=1/32$. Условие $P>P_0$ означает, что 5-мерное равнораспределение вероятностей никогда не выполняется для $q=1$, т.е. для консервативных гиперболических автоморфизмов тора. Ниже будет рассмотрен более общий случай $q\neq 1$ (Barash, 2011).

Пусть $X_i=\left\{(x,y)^T \,|\, i/2^v\leq x/g<(i+1)/2^v, 0\leq y/g<1\right\}$, т.е. тор поделен на 2^v вертикальных полосок $X_0,X_1,\ldots,X_{2^v-1}$. Пусть g делится на 2^v . Рассмотрим сдвиг $S:(x,y)^T\to(x+g/2^v,y)^T \pmod g$ т.е. $S(X_i)=X_{(i+1)(\mathrm{mod}\ 2^v)}$. Сдвиг S является суперпозицией двух поворотов: $S=R_1R_2$, где R_1 — поворот на 180 градусов относительно точки $(1/2^{v+1},1/2)^T$, а R_2 — поворот на 180 градусов относительно точки $(1/2^v,1/2)^T$.

Утверждение 1. *Если*

а) $M=\begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix}$ *— матрица с целыми элементами;*

b) m_1, $q = \det M$ *и* g *делятся на* 2^v*;*

c) образ сетки $g \times g$ *при преобразовании* M^j *инвариантен по отношению к сдвигу* S *для* $j = 0, 1, \ldots, n$*;*

то все последовательности длины n *в потоке* v*-битовых блоков, генерируемых при помощи матрицы* M*, равновероятны.*

Доказательство. В этом случае элемент $m_1^{(n)}$ матрицы $M^n = \begin{pmatrix} m_1^{(n)} & m_2^{(n)} \\ m_3^{(n)} & m_4^{(n)} \end{pmatrix} \pmod g$ удовлетворяет рекуррентному соотношению $m_1^{(n)} = km_1^{(n-1)} - qm_1^{(n-2)} \pmod g$. Следовательно, $m_1^{(n)}$ делится на 2^v для любого целого $n \geq 1$. Поскольку $m_1^{(n)}$ делится на 2^v, имеем $M^n S(x,y)^T = M^n(x+g/2^v \pmod g, y)^T = M^n(x,y)^T + (0, m_3^{(n)} g/2^v)^T$. Следовательно, множество точек A таких, что $A \in X_i$ и $M^n(A) \in X_j$, переходит при сдвиге S в множество точек A таких, что $A \in X_{(i+1) \pmod{2^v}}$ и $M^n(A) \in X_j$.

Докажем теперь по индукции, что все последовательности длины n равновероятны. Очевидно, поскольку g делится на 2^v, последовательности длины 1 равновероятны: $P(0) = P(1) = \cdots = P(2^v - 1) = 1/2^v$. Предположим, что все последовательности длины $n-1$ равновероятны. Пусть $\alpha_i = P\big(\{ix_1 \ldots x_{n-1}\}\big)$, $i = 0, 1, \ldots, 2^v - 1$, т.е. α_i обозначают вероятности последовательностей длины n. Тогда $\alpha_i = \alpha_{i+1}$, $i = 0, 1, \ldots, 2^v - 2$, потому что множество точек A сетки $g \times g$ таких, что $A \in X_i$, $M(A) \in X_{x_1}, \ldots, M^{n-1}(A) \in X_{x_{n-1}}$ переходит со сдвигом S в множество точек A сетки $g \times g$ таких, что $A \in X_{(i+1) \bmod 2^v}$, $M(A) \in X_{x_1}, \ldots, M^{n-1}(A) \in X_{x_{n-1}}$. С другой стороны, величина $\sum_{i=0}^{2^v-1} \alpha_i$ есть вероятность последовательности $x_1, \ldots, x_{n-1}$ длины $n-1$ и равна $1/2^{v(n-1)}$. Следовательно, $\alpha_i = 1/2^{vn}$, $i = 0, 1, \ldots, 2^v - 1$, и все последовательности длины n равновероятны. Утверждение 1 доказано.

Пример. *Для* $M = \begin{pmatrix} 2 & 2 \\ 1 & 2 \end{pmatrix}$, $M = \begin{pmatrix} 10 & 17 \\ -4 & -2 \end{pmatrix}$ *и* $M = \begin{pmatrix} 244 & 43 \\ 32 & 12 \end{pmatrix}$ *подпоследовательности длины* $1, 2, \ldots, \ell$ *потока битов, генерируемого при помощи матрицы* M*, равновероятны, где* $\ell = 2t - 1$, $\ell = (t-1)/2$ *и* $\ell = (t-1)/2$ *соответственно. Здесь* $g = p \cdot 2^t$*, где* p *— нечетное простое число, а матрицы отвечают реализациям GM29, GM58 и GM55 соответственно.*

Действительно, докажем это утверждение, т.е. докажем, что образ сетки $g \times g$ при преобразовании M^j инвариантен относительно сдвига S для $j = 0, 1, \ldots, n$ и $n \leq \ell$. В частности, инвариантность имеет место, если имеются целые числа $r, l < t$ такие, что расстояние между целочисленными векторами $(x + g/2^{r+1}, y + g/2^{l+1})^T$ и $(x,y)^T$ после применения преобразования M^j равно $(g/2, 0)^T$ по модулю g. Отсюда имеем $\big(m_1^{(j)}/2^r + m_2^{(j)}/2^l, m_3^{(j)}/2^r + m_4^{(j)}/2^l\big)^T \equiv (1,0)^T \mod 2$. Для матрицы $M = \begin{pmatrix} 2 & 2 \\ 1 & 2 \end{pmatrix}$ условие выполняется при $r = j/2$, $l = j/2 - 1$ для четных j и $r = (j-1)/2$, $l = (j+1)/2$ для нечетных j. Таким образом, $\ell = j_{max} + 1 = 2t - 1$. Аналогично, для каждой из матриц $M = \begin{pmatrix} 10 & 17 \\ -4 & -2 \end{pmatrix}$ и $M = \begin{pmatrix} 244 & 43 \\ 32 & 12 \end{pmatrix}$ условие выполнено при $\ell = (t-1)/2$.

Утверждение 2. *Рассмотрим матрицу M с целыми элементами и следующие целые величины: $g = p \cdot 2^t$, $q = \det M = 2^u w (\mathrm{mod}\ g)$, $k = \mathrm{Tr}\, M = 2^m r (\mathrm{mod}\ g)$, $u \geq 1$, $t \geq v$, $m \geq 0$. Здесь w, r — нечетные целые числа, а p — нечетное простое число. Тогда*

a) все 2^{vj} последовательности длины j в потоке v-битовых блоков, генерируемом рекуррентным соотношением (1), равновероятны для $j = 1, 2, \ldots, \ell$. Здесь $\ell = \lceil (t-v)/\lceil u/2 \rceil \rceil$ для $u \leq 2m$ и $\ell = \lceil (t-v)/(u-m) \rceil$ для $u > 2m$;

b) если k четно, то образ сетки $g \times g$ при преобразовании M^{2t} является сеткой $p \times p$ на торе;

c) если k нечетно, то образ сетки $g \times g$ при преобразовании $M^{\lceil t/u \rceil}$ не инвариантен по отношению к сдвигу S.

Доказательство.

a) Пусть $X'_i = \left\{ x | ig/2^v \leq x < (i+1)g/2^v \right\}$, $i = 0, 1, \ldots, 2^v - 1$. Пусть $k_0 = 1$, $k_1 = k (\mathrm{mod}\ g)$, $k_{i+1} = kk_i - qk_{i-1} (\mathrm{mod}\ g)$, $i \in \mathbb{N}$. Рассмотрим выражения

$$\xi^{(h-i)} = p \cdot 2^{t-iu-v} w^{h-i} k_i (\mathrm{mod}\ g), \tag{4}$$

которые определяют целые величины $\xi^{(h-i)}$ для $i = 0, 1, \ldots, i_{max}$ для некоторого i_{max}. Легко убедиться, что следующие соотношения выполняются: $\xi^{(j)} = k\xi^{(j-1)} - q\xi^{(j-2)} (\mathrm{mod}\ g)$, $j = h, h-1, \ldots, h - i_{max} + 2$. Кроме того, легко проверить, что $\xi^{(h+i)} = 0 (\mathrm{mod}\ g)$, для $i \in \mathbb{N}$, где $\xi^{(h+i)}$ определяется в этом случае как $k\xi^{(h+i-1)} - q\xi^{(h+i-2)} (\mathrm{mod}\ g)$.

Легко показать по индукции следующее: если $u \leq 2m$, то k_i делится на $2^{\min(fi,t)}$, где $f = \lfloor u/2 \rfloor$; если $u > 2m$, то k_i делится на $2^{\min(mi,t)}$. Следовательно, выражения (4) определяют целые значения $\xi^{(h-i)}$ для $i = 0, 1, \ldots, \ell - 1$, где $\ell = \lceil (t-v)/\lceil u/2 \rceil \rceil$ для $u \leq 2m$ и $\ell = \lceil (t-v)/(u-m) \rceil$ для $u > 2m$.

Докажем теперь, что каждая последовательность длины $n \leq l$ имеет одну и ту же вероятность $1/2^{vn}$. Очевидно, поскольку g делится на 2^v, последовательности длины 1 равновероятны и $P(i) = 1/2^v$ для $i = 0, 1, \ldots, 2^v - 1$. Пусть $P(x_h \ldots x_{n-1})$ обозначает вероятность того, что последние $n - h$ элементов последовательности длины n есть $x_h, \ldots, x_{n-1}$, где $x_i \in \{0, 1, \ldots, 2^v - 1\}$, $h < n$, $i = h, \ldots, n-1$. Тогда $P(x_h \ldots x_{n-1}) = |B|/g^2$, где $B = \left\{ (x^{(0)}, x^{(1)})^T \,|\, x^{(h)} \in X'_{x_h}, \ldots, x^{(n-1)} \in X'_{x_{n-1}} \right\}$, $x^{(i)}$ определяется как $kx^{(i-1)} - qx^{(i-2)} (\mathrm{mod}\ g)$ для $i \geq 2$ и $|B|$ обозначает число элементов множества B.

Следовательно, если $h \leq \ell - 1$ то $P(x_h x_{h+1} \ldots x_{n-1}) = P(x'_h x_{h+1} \ldots x_{n-1})$, где $x'_h = x_h + w^h (\mathrm{mod}\ g)$. Действительно,
$\left\{ \left(x^{(0)} + \xi^{(0)}, x^{(1)} + \xi^{(1)}\right)^T \,\middle|\, x^{(h)} \in X'_{x_h}, x^{(h+1)} \in X'_{x_{h+1}}, \ldots, x^{(n-1)} \in X'_{x_{n-1}} \right\} =$
$\left\{ \left(x^{(0)}, x^{(1)}\right)^T \,\middle|\, x^{(h)} \in X'_{x'_h}, x^{(h+1)} \in X'_{x_{h+1}}, \ldots, x^{(n-1)} \in X'_{x_{n-1}} \right\}$, где $\xi^{(0)}$ и $\xi^{(1)}$ определены при помощи (4) и являются целыми числами при $h \leq \ell$. Поскольку w^h

является нечетным целым числом, получаем отсюда $\beta_0 = \beta_1 = \cdots = \beta_{2^v-1}$, где $\beta_i = P(ix_{h+1} \dots x_{n-1})$, $i = 0, 1, \dots, 2^v - 1$.

В частности, для $h = n - 1$ и $n \le \ell$ имеем $P(i) = 1/2^v$, $i = 0, 1, \dots, 2^v - 1$ и по индукции получаем для $h \le \ell - 1$, что $P(ix_{h+1} \dots x_{n-1}) = 1/2^{v(n-h)}$, $i = 0, 1, \dots, 2^v - 1$. В частности, если $n \le \ell$, то $P(ix_1 \dots x_{n-1}) = 1/2^{vn}$, $i = 0, 1, \dots, 2^v - 1$, и, следовательно, $P(x_0 x_1 \dots x_{n-1}) = 1/2^{vn}$.

b) В этом случае $x^{(n+2)} = kx^{(n+1)} - qx^{(n)} (\mathrm{mod}\ g)$, $n = 0, 1, 2, \dots$ Следовательно, если $(x^{(0)}, y^{(0)})^T$ принадлежит сетке $g \times g$, то $x^{(2)}, y^{(2)}$ — четные целые числа, $x^{(4)}, y^{(4)}$ делятся на 4 и т.д. $x^{(2t)}$ и $y^{(2t)}$ делятся на 2^t и, следовательно, $(x^{(2t)}, y^{(2t)})^T$ принадлежит сетке $p \times p$ на торе.

c) Пусть L — образ сетки $g \times g$ при преобразовании M^n, где $n = \lceil t/u \rceil$. Тогда $(0, 0)^T \in L$, поскольку $M^n(0, 0)^T = (0, 0)^T$. Если L инвариантно по отношению к сдвигу, то $(g/2^v, 0)^T \in L$. Следовательно, существует точка $(x, y)^T$ сетки $g \times g$ такая, что $M^n(x, y)^T = (g/2^v, 0)^T (\mathrm{mod}\ g)$. Поскольку $m_1^{(n)}$ делится на 2^v, имеем $M^n(g/2^v, 0)^T = \left(0, m_3^{(n)} g/2^v\right)^T (\mathrm{mod}\ g)$. Следовательно, $0 = k'_n g/2^v - q^n x \pmod g$, где $k'_n = \operatorname{Tr} M^n$ является нечетным целым числом для $n \ge 1$. Здесь мы приходим к противоречию, поскольку $q^n = 0 (\mathrm{mod}\ 2^t)$, $k'_n g/2^v \ne 0 (\mathrm{mod}\ 2^t)$.

Утверждение 2 доказано.

Хотя точное равнораспределение перестает работать, когда расстояния между точками последовательности начинают превышать $2t$, численные расчеты показывают, что равнораспределение работает приближенно с высокой точностью для последовательностей битов длины n, где $n < 6.8 \log p$. Кроме того, можно взять n точек с произвольными расстояниями между ними вдоль орбиты (не превышающими $p^2 - 1$), где $n < 6.8 \log p$, и все еще приблизительное равнораспределение будет работать с высокой точностью. Если $v = 1$, то выходное значение $a^{(n)}$ в (3) состоит из старших битов s последовательных точек вдоль орбиты матрицы M^A, где A определено в разделе 4, поэтому, согласно численным результатам, выходное значение имеет равномерное распределение с очень высокой точностью.

В большинстве случаев образ сетки $g \times g$ на торе при преобразовании M^j, где $j \ge 2t$, является сеткой $p \times p$, поэтому интересны для изучения отклонения от равнораспределения для сетки $p \times p$. Мы посчитали точные площади областей на торе, которые соответствуют каждой из последовательностей для $M = \begin{pmatrix} 1 & 1 \\ 1 & 3 \end{pmatrix}$. Вычисления были проведены на рабочей станции с использованием Class Library for Numbers для точной арифметики произвольных рациональных чисел. Для каждой из 2^n последовательностей длины $n \in \mathbb{N}$ соответствующее множество точек единичного двумерного тора состоит из заполненных многоугольников. Были найдены точные рациональные координаты всех вершин каждого многоугольника. Также было вычислено точное число точек сетки $p \times p$ внутри каждого многоугольника. Как выяснилось, общая площадь многоугольников, соответствующих каждой из 2^n последовательностей длины n, равняется $1/2^n$. Как показывают результаты расчетов, такое равенство площадей, соответствующих разным последовательностям одной

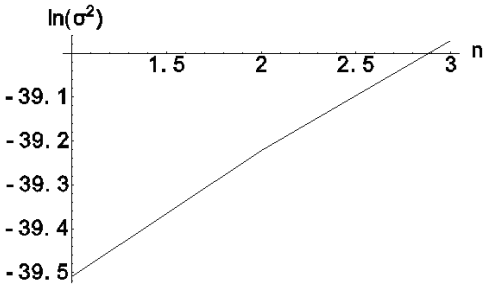

Рис. 1. Дисперсия числа точек сетки $p \times p$, соответствующих последовательностям длины n, в зависимости от n. Значения нормированы так, чтобы $\langle A_n \rangle = 1$

и той же длины, выполняется для матриц с четным определителем и не выполняется для матриц с нечетным определителем. Пусть $A_{n,0}, A_{n,1}, \ldots, A_{n,2^n-1}$ — числа точек сетки $p \times p$ внутри множеств заполненных многоугольников, которые соответствуют последовательностям длины n. Тогда $\sum_{i=0}^{2^n-1} A_{n,i} = p^2$. Следовательно, если множество чисел A_n определить как $A_n = \left\{2^n A_{n,0}/p^2, 2^n A_{n,1}/p^2, \ldots, 2^n A_{n,2^n-1}/p^2\right\}$, то $\langle A_n \rangle = 1$, где $\langle A_n \rangle$ — среднее значение чисел из A_n. Зависимость логарифма дисперсии A_n от n показана на рисунке 1 для $p = 2^{29} - 3$. Вычисления для меньших значений p и больших значений n показывают, что зависимость $\log(\sigma^2)$ от n практически линейна. Вычисления показывают, что отклонения от равнораспределения пренебрежимо малы в том смысле, что $\sigma(A_n)$ намного меньше, чем $\langle A_n \rangle = 1$, если $n < 6.8 \log p$. В частности, для $p = 2^{29} - 3$ отклонения малы при $n < 130$.

Также вычисления показывают, что дисперсия для нескольких точек орбиты матрицы M на сетке $p \times p$ на торе существенно зависит от числа точек и от значения p и очень слабо зависит (в пределах нескольких процентов) от расстояний между точками вдоль орбиты.

6. Методы генерации параллельных потоков случайных чисел

Основными используемыми методами генерации параллельных потоков случайных чисел являются следующие (согласно классификации предложенной в (Bauke, Mertens, 2007)):

А) Случайный выбор начальных величин

Все процессы используют один и тот же генератор случайных чисел, но с разными «случайными» начальными значениями. Надежда на то, что они будут генерировать статистически независимые и непересекающиеся подпоследовательности, не имеет теоретического обоснования. Применение такого «оптимистического» метода следует избегать.

Б) Параметризация

Все процессы используют генератор одного и того же типа, но для каждого процесса используется свой набор параметров. Например, используются линейно-конгруэнтные генераторы, отличающиеся друг от друга приращением c; в качестве приращения для разных процессов берутся различные простые числа (Percus, Kalus, 1989). В другом варианте используются разные множители a для разных потоков

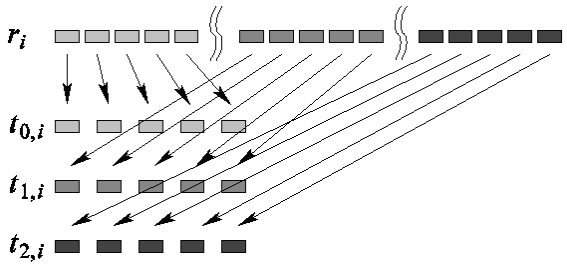

Рис. 2. Параллелизация расщеплением блока (иллюстрация из (Bauke, Mertens, 2007))

(Mascagni, 1998). Эти методы имеют слабое теоретическое обоснование и эмпирические тесты выявили серьезные корреляции между потоками (Matteis, Pagnutti, 1990). В то же время, использование различных параметров в генераторах типа ГСР не выявило заметных корреляций между потоками (Bloete, Shchur, Talapov, 1999). Однако число возможных потоков в таком методе органичено наборами известных параметров (см. обзор методов поиска параметров для генераторов типа ГСР в (Бараш, 2005)).

В) Расщепление блока

Пусть M — максимальное число вызовов генератора случайных чисел у одиночного процесса, p — число процессов. Тогда мы можем разделить выходную последовательность генератора случайных чисел на последовательные блоки длины M (см. рис. 2). Для того, чтобы этот метод можно было применить, требуется следующее:

а) должна быть возможность до начала вычислений сделать оценку сверху для величины M;

б) у генератора случайных чисел должен быть эффективный алгоритм для «пропуска кусков»;

в) должно быть теоретическое обоснование того, что такое использование генератора не приведет к тому, что сыграют негативную роль возможные корреляции между элементами выходной последовательности, находящимися в обычной выходной последовательности на большом расстоянии между собой.

Для некоторых генераторов, например, для генератора, основанного на ансамбле преобразований тора, последствия метода расщепления блока в зависимости от параметров генератора можно теоретически предсказать, поскольку пропуск M чисел в выходной последовательности приводит к генератору того же типа.

Г) Чехарда

Метод параллелизации чехардой основан на том, что каждый из процессов, используя одно из значений генератора, каждый раз пропускает следующие $p - 1$ значений, где p — число процессов (см. рис. 3). Это достаточно универсальный и надежный метод, не требующий заранее априори оценивать максимальное число вызовов генератора случайных чисел у одиночного процесса. Конечно, у генератора случайных чисел должен быть эффективный алгоритм для «пропуска кусков».

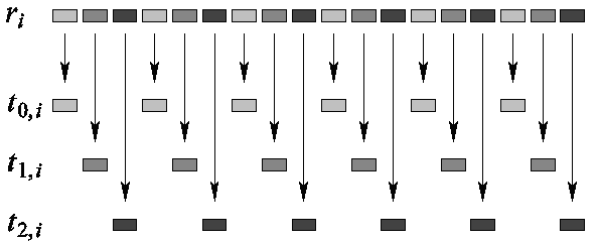

Рис. 3. Параллелизация чехардой (иллюстрация из (Bauke, Mertens, 2007))

7. Существующие пакеты программ

Ниже мы приводим сравнительный анализ существующих библиотек по генерации случайных чисел и параллельных потоков случайных чисел.

А) GNU Scientific Library

Библиотека GNU Scientific Library включает в себя реализации следующих генераторов псевдослучайных чисел: borosh13, coveyou, cmrg, fishman18, fishman20, fishman2x, gfsr4, knuthran, knuthran2, lecuyer21, minstd, mrg, mt19937, mt19937_1999, mt19937_1998, r250, ran0, ran1, ran2, ran3, rand, rand48, random_bsd, random_glibc2, random_libc5, random_libc5, random8_bsd, random8_glibc2, random8_libc5, random32_bsd, random32_glibc2, random32_libc5, random64_bsd, random64_glibc2, random64_libc5, random128_bsd, random128_glibc2, random128_libc5, random256_bsd, random256_glibc2, random256_libc5, randu, ranf, ranlux, ranlux389, ranlxd1, ranlxd2, ranlxs0, ranlxs1, ranlxs2, ranmar, slatec, taus, taus2, transputer, tt800, uni, uni32, vax, waterman14, zuf.

Это старые генераторы случайных чисел в их классическом виде (линейно-конгруэнтные, сдвиговые регистры, lagged Fibonacci и т.д. в их классическом виде), имеющие упомянутые выше недостатки. Такие генераторы нельзя применять для вычислений методом Монте-Карло на современных суперкомпьютерных системах.

Из современных генераторов добавлен Mersenne Twister и некоторые его предшественники. Генераторы реализованы для CPU стандартным образом на языке Си. Ускорение при помощи технологии SIMD, как и генерация параллельных потоков случайных чисел, в библиотеке не предусмотрены.

Б) Intel MKL Library

Библиотека Intel MKL Library включает в себя составную часть, которая называется Vector Statistical Library. В ней реализованы следующие генераторы псевдослучайных чисел: MCG31m1 (линейно-конгруэнтный генератор), R250 (сдвиговый регистр), MRG32K3A (комбинированный MRG), MCG59 (линейно-конгруэнтный), WH (комбинированный линейно-конгруэнтный, 273 разных параметра), MT19937 (Mersenne Twister), MT2203 (6024 генератора такого же типа как Mersenne Twister с разными параметрами), SFMT19937 (генератор аналогичный Mersenne Twister, параметры специально подобраны для ускорения при помощи технологии SIMD).

Библиотека содержит 6 генераторов и их разные версии. Все генераторы реализованы для CPU, имеется увеличение эффективности в несколько раз по сравнению с обычными реализациями, которое достигается при помощи технологии SIMD, т.е.

при помощи SSE-команд и 128-битных XMM-регистров. Хотя библиотека содержит наборы из реализаций с разными параметрами для двух генераторов (273 версии генератора WH и 6024 версии генератора MT2203), ее не следует использовать для генерации параллельных потоков псевдослучайных чисел, поскольку отсутствие корреляций между разными потоками не исследовано и не гарантируется.

В) RNGSSELIB

В программный пакет включены известные современные и наиболее надежные генераторы: MT19937, MRG32K3A, LFSR113, а также генераторы GM19, GM31 и GM61, основанные на параллельной эволюции автоморфизмов тора. Библиотека содержит как обычные реализации, так и новые реализации, в которых использована технология SIMD, т.е. команды SSE и регистры XMM для существенного ускорения работы на современных процессорах. Для известных ранее генераторов эффективность работы новых реализаций превосходит эффективность реализаций других авторов, в частности, превосходит эффективность реализаций из библиотеки Intel Math Kernel Library.

В работе (Barash, Shchur, 2011) представлены детальное сравнение эффективности работы и детальные результаты статистических тестов для всех представленных в библиотеке генераторов. Генерация параллельных потоков случайных чисел в библиотеке не предусмотрена.

Г) The Scalable Parallel Random Number Generators Library (SPRNG)

Библиотека SPRNG включает в себя реализации следующих генераторов псевдослучайных чисел: LCG48 (линейно-конгруэнтный), LFG (Lagged Fibonacci Generator), LCG64 (линейно-конгруэнтный), CMRG (комбинированный MRG-генератор), MLFG (мультипликативный Lagged Fibonacci Generator), PMLCG (линейно-конгруэнтный). Это старые генераторы случайных чисел, почти все из них в классическом виде.

Библиотека была разработана в середине 1990-х годов, используемые алгоритмы генерации требуют дополнительной проверки современными библиотеками статистических тестов. Генераторы реализованы для CPU стандартным образом на Си. Ускорение при помощи технологии SIMD не предусмотрено.

Хотя библиотека включает в себя подпрограммы для генерации параллельных потоков случайных чисел, генераторы распараллелены методом параметризации, без особой теории, поэтому отсутствие корреляций между разными потоками не гарантировано.

Д) Tina's Random number generator library (TRNG)

Библиотека TRNG включает в себя реализации следующих генераторов псевдослучайных чисел: lcg64, lcg64_shift, mrg_, mrg_s, yarn_, yarn_s, lagfib_xor, lagfib_plus, mt19937, mt19937_64. Таким образом, это классические линейно-конгруэнтные генераторы, MRG-генераторы, Lagged Fibonacci-генераторы, а также Mersenne Twister. Эти генераторы реализованы как для CPU, так и предложены алгоритмы с ускорением вычислений при помощи GPU и технологии CUDA.

Для первых шести из перечисленных выше генераторов, реализованных в библиотеке TRNG, предложены методы параллелизации и генерации параллельных потоков случайных чисел.

8. Заключение

В работе мы привели анализ современных методов генерации случайных чисел и библиотек программ. Анализ проведен с точки зрения применимости методов генерации параллельных потоков случайных чисел или возможности их расширения на этот случай. Можно сделать вывод, что одним из наиболее перспективных, не имеющих принципиальных ограничений на число параллельных потоков является предложенный ранее авторами метод генерации, основанный на преобразованиях тора.

Для этого метода нами ранее развита теория, которая позволяет уверенно прогнозировать возможность создания генераторов и библиотек генераторов для использования при моделировании на параллельных вычислительных системах и суперкомпьютерных системах гибридного типа. Работа в этом направлении уже ведется.

9. Благодарности

Авторы благодарны С.А. Крашакову и А.Ю. Меньшутину за полезные обсуждения.

Работа выполнена при поддержке Министерства образования и науки Российской Федерации, ГК № 07.514.11.4032.

Литература

1. Бараш Л.Ю. Алгоритм AKS проверки чисел на простоту и поиск констант генераторов псевдослучайных чисел // Безопасность информационных технологий. 2005. 2. С. 27–38.

2. Кнут Д.Э. Искусство программирования. Том 2: Получисленные алгоритмы. 3-е изд. Вильямс, 2000.

3. Лихтенберг А., Либерман М. Регулярная и стохастическая динамика. М.: Мир, 1984.

4. Шустер Г. Детерминированный хаос, введение. М.: Мир, 1988.

5. Arnol'd V. I., Avez A. Ergodic Problems of Classical Mechanics. Nenjamin, New York, 1968.

6. Barash L.Yu. // Europhysics Letters. 2011. 95, 10003.

7. Barash L., Shchur L.N. // Phys.Rev. 2006. E 73, 036701.

8. Barash L.Yu., Shchur L.N. // Comput. Phys. Commun. 2011. 182. P. 1518–1527.

9. Barash L.Yu. // Springer Proceedings in Mathematics and Statistics. Springer-Verlag, Berlin, Heidelberg, 2012. Vol. 23. P. 245–260.

10. Bauke H. Tina's Random Number Generator Library. 2011 http://numbercrunch.de/trng/

11. Bauke H., Mertens S. // Phys. Rev. 2007. E 75, 066701 .

12. Beach K.S.D., Lee P.A., Monthoux P. // Phys. Rev. Lett. 2004. 92, 026401.

13. Binder K., Heermann D. W. Monte Carlo Simulation in Statistical Physics. Berlin: Springer-Verlag, 1992.

14. Bizzarri A.R. // J. Phys.: Cond. Mat. 2004. 16, R83.

15. Bloete H.W.J., Shchur L.N. and Talapov A.L. // Int. J. Mod. Phys. (1999. C 10. P. 1137–1148.

16. Blum L., Blum M., Shub M. // SIAM J. of Comp. 1986. 15. 364.

17. Chapman R. Notes on Algebraic Numbers.
http://www.secamlocal.ex.ac.uk/people/staff/rjchapma/notes/algn.pdf
(1995, 2002)

18. Cohn H. A Second Course in Number Theory. New York: Wiley, 1962. [Reprinted by Dover, New York with the title Advanced Number Theory (1980).]

19. Coveyou R.R. and MacPherson R.D. // J. ACM. 1967. 14. 100; Marsaglia G. // Proc. Nat. Acad. Sci. USA. 1968. 61. 25.

20. Ferrenberg A. M., Landau D. P. , Wong Y. // J. Phys.Rev.Lett. 1992. 69, 3382 .

21. Galassi M. et al. GNU Scientific Library Reference Manual. Third Edition. Network Theory Ltd., 2009.

22. Golomb S. W. Shift Register Sequences. Holden-Day, San Francisco, 1967.

23. Grassberger P. // Phys. Lett. 1993. 181, 43.

24. Grothe H. // Statistical Papers. 1987. 28, 233.

25. Intel® Math Kernel Library. Reference Manual, September 2007
http://www.intel.com/cd/software/products/emea/rus/358888.htm

26. Keating J. P. Asymptotic properties of the periodic orbits of the cat maps // Nonlinearity. 1991. 4. P. 277–307 .

27. Landau D.P. and Binder K. A Guide to Monte Carlo Simulations in Statistical Physics. Cambridge: Cambridge University Press, 2000.

28. L'Ecuyer P. // Ann. Oper. Res. 1994. 53, 77.

29. L'Ecuyer P. // Math. of Comp. 1996. 65, 203.

30. L'Ecuyer P. // Oper. Res. 1999. 47, 159.

31. L'Ecuyer P. // Math. of Comp. 1999. 68, 261.

32. L'Ecuyer P., Simard R. TestU01: A Software Library in ANSI C for Empirical Testing of Random Number Generators, 2002. Software user's guide http://www.iro.umontreal.ca/ simardr/testu01/tu01.html.

33. L'Ecuyer P., Simard R. // ACM TOMS. 2007. 33(4). Article 22.

34. Lemer D.H. Proceedings of the 2nd Symposium on Large-Scale Digital Calculating Machinery. Cambridge, MA, 1951. P. 141–146.

35. Luchow A. // Ann. Rev. Phys. Chem. 2000. 51, 501.

36. Marsaglia G. Die Hard: A battery of tests for random number generators http://stat.fsu.edu/pub/diehard

37. Mascagni M. // Parallel Computations. 1998. 24, 923.

38. Mascagni M. and Srinivasan A. Algorithm 806: SPRNG: A Scalable Library for Pseudorandom Number Generation, ACM Transactions on Mathematical Software. 2000. 26. P. 436–461 .

39. Matsumoto M. and Tishimura T. // ACM Trans. on Mod. and Comp. Sim. 1998. 8, 3.

40. Matteis A.D. and Pagnutti S. // Parallel Comput. 1990. 13, 193.

41. Niederreiter H. Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing / ed. H. Niederreiter and P. J.-S. Shiue // Lecture Notes in Statistics. Springer-Verlag, 1995. Vol. 106.

42. Percival I. C., Vivaldi F. Arithmetical Properties of Strongly Chaotic Motions // Physica. 1987. 25D. P. 105–130.

43. Percus O.E. and Calos M.H. // J. Parallel. Distrib. Comput. 1989. 6, 477.

44. Pieper S.C. and Wiring R.B. // Ann. Rev. Nucl. Part. Sci. 2001. 51, 53.

45. Schmid F., Wilding N. B. // Int.J.Mod.Phys. 1995. C 6, 781.

46. Shchur L.N., Heringa J. R., Bloete H. W. J. // Physica. 1997. A241, 579.

47. Shchur L. N., Bloete H. W. J. // Phys.Rev. 1997. E 55, R4905.

48. Tezuka S. Uniform Random Numbers: Theory and Practice. Kluwer, Boston et al., 1995.

49. Vattulainen I., Ala-Nissila T., Kankaala K. // Phys. Rev. Lett. 1994. 73, 2513.

Generation of Random Numbers and Parallel Random Number Streams for Monte Carlo Simulations

Barash L. Yu., Shchur L. N.

Keywords: random number generators, parallel computing, nonlinear dynamical systems

Modern methods and libraries for high quality pseudorandom number generation and for generation of parallel random number streams for Monte Carlo simulations are considered. The probability equidistribution property and the parameters when the property holds at dimensions up to logarithm of mesh size are considered for Multiple Recursive Generators.

Сведения об авторах:
Бараш Лев Юрьевич,
Институт теоретической физики им. Л.Д. Ландау Российской академии наук,
канд. физ.-мат. наук, младший научный сотрудник.
Щур Лев Николаевич,
Институт теоретической физики им. Л.Д. Ландау Российской академии наук,
д-р физ.-мат. наук, профессор, ведущий научный сотрудник.