

Использование Multi-GPU для многомерного численного интегрирования методом Монте-Карло.

Л.Ю. Бараш, Л.Н. Щур

ИТФ им. Л.Д. Ландау РАН

НЦЧ РАН

Таруса, ИКИ РАН, 2 октября 2014 года

Генерация параллельных потоков случайных чисел для расчетов Монте-Карло с использованием GPU.

1. Разработаны библиотека PRAND и библиотека RNGSSELIB по параллельной генерации псевдослучайных чисел, которая включает генераторы, основанные на параллельной эволюции автоморфизмов тора (GM19, GM31, GM61, GM29.1, GM55.4, GQ58.1, GQ58.3, GQ58.4), генератор MRG32K3A, генератор LFSR113, генератор MT19937.
2. Для каждого из генераторов реализованы:
 - инициализация параллельных потоков методом расщепления блока; имеется возможность инициализировать до 10^{19} независимых потоков
 - эффективные версии для CPU с использованием SIMD-параллелизма и SSE-команд;
 - однопоточные версии, которые можно использовать в вычислениях Монте-Карло на графических процессорах, распределенных по нитям и вычислительным узлам произвольным образом;
 - параллельные версии с использованием множества нитей графического процессора для ускорения вычислений.
 - совместимость с фортраном; имеются примеры использования на Фортране и на Си.

Линейно-конгруэнтные: $x_{n+1} = (ax_n + c)(\text{mod } M).$

Сдвиговые регистры: $x_n = (a_1x_{n-1} + \dots + a_kx_{n-k})(\text{mod } 2),$

$$u_n = \sum_{i=1}^L x_{ns+i-1} 2^{-i}, \quad P(z) = z^k - a_1z^{k-1} - \dots - a_k.$$

Генератор Mersenne Twister: $\mathbf{x}_{k+n} := \mathbf{x}_{k+m} \oplus (\mathbf{x}_k^u | \mathbf{x}_{k+1}^l)A.$

Комбинированный генератор MRG32K3A: $x_n = (ax_{n-2} + bx_{n-3})(\text{mod } m_1),$

$$a = 1403580, \quad b = -810728, \quad c = 527612, \quad d = -1370589, \quad y_n = (cy_{n-1} + dy_{n-3})(\text{mod } m_2),$$

$$m_1 = 2^{32} - 209, \quad m_2 = 2^{32} - 22853, \quad z_n = (x_n + y_n)(\text{mod } m_1).$$

Комбинированный
Tausworthe-генератор
LFSR113 является
комбинацией четырех
сдвиговых регистров:

- $x_n = x_{n-31} \oplus x_{n-25}, \quad s = 18,$
- $x_n = x_{n-29} \oplus x_{n-27}, \quad s = 2,$
- $x_n = x_{n-28} \oplus x_{n-15}, \quad s = 7,$
- $x_n = x_{n-25} \oplus x_{n-22}, \quad s = 13.$

PRNGs based on ensemble of dynamical systems and their properties:

The state of the generator consists of the values $x_i^{(n-1)}, x_i^{(n-2)} \in \{0, 1, \dots, g-1\}$, $i = 0, 1, \dots, s-1$. The transition function of the generator is defined by the recurrence relation

$$x_i^{(n)} = kx_i^{(n-1)} - qx_i^{(n-2)} \pmod{g}, \quad (1)$$

where $i = 0, 1, \dots, s-1$. The values $x_i^{(n)}$, $i = 0, 1, \dots, s-1$ can be considered as x -coordinates of s points $(x_i^{(n)}, y_i^{(n)})^T$, $i = 0, 1, \dots, s-1$ of the $g \times g$ lattice on the two-dimensional torus, then each recurrence relation describes the dynamics of x -coordinate of a point on the two-dimensional torus:

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

where matrix $M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix}$ is a matrix with integer elements, $k = \text{Tr } M$, $q = \det M$ and $\text{Tr } M$ is a trace of matrix M .

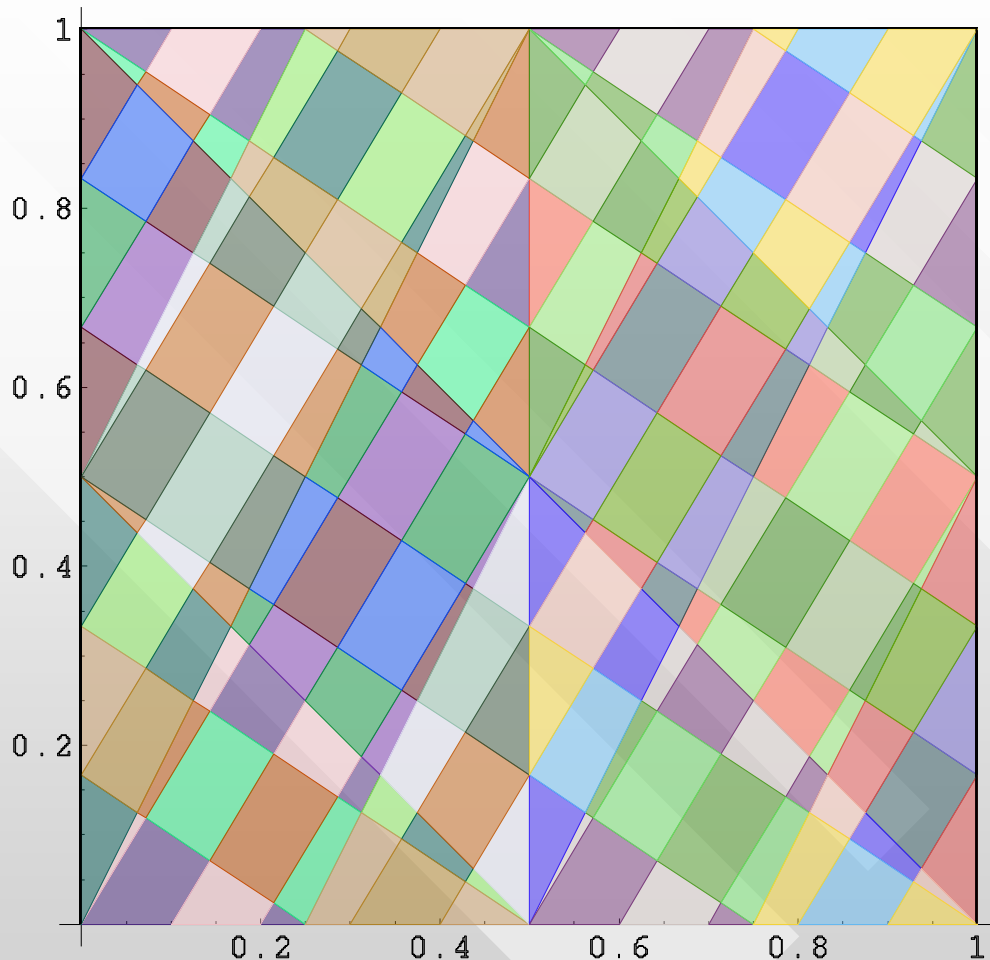
The output function:

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

where v bits are taken from each recurrence and $i = 0, 1, \dots, s-1$.

The sequence of bits $\{\lfloor 2^v x_i^{(n)} / g \rfloor\}$, where i is fixed and $\{x_i^{(n)}\}$ is generated with relation (2) is designated as a stream of v -bit blocks generated with matrix M .

Области на торе и пятибитовые последовательности генерируемые гиперболическим автоморфизмом



L.Yu. B., Europhysics Letters (EPL) 95, 10003 (2011).
L.Yu. B., L.N. Shchur, Phys.Rev. E 73 , 036701 (2006).

Equidistribution properties of the PRNGs based on ensemble of dynamical systems:

Proposition 1. If (i) $M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix}$ is a matrix with integer values m_1, m_2, m_3, m_4 , (ii) $m_1, q = \det M$ and g are divisible by 2^v , (iii) the image of the lattice $g \times g$ with the transformation M^j is invariant with respect to the shift S for $j = 0, 1, \dots, n$, then all the sequences of length n in a stream of v -bit blocks generated with matrix M are equiprobable.

Proposition 2. Consider a matrix M with integer elements and the following integer quantities: $g = p \cdot 2^t$, $q = \det M = 2^u w \pmod{g}$, $k = \text{Tr } M = 2^m r \pmod{g}$, $u \geq 1$, $t \geq v$, $m \geq 0$. Here w, r are odd integers and p is an odd prime. Then all 2^j sequences of length j in a stream of v -bit blocks generated with recurrence relation (1) are equiprobable for $j = 1, 2, \dots, \ell$. Here $\ell = \lceil (t - v) / \lceil u/2 \rceil \rceil$ for $u \leq 2m$ and $\ell = \lceil (t - v) / (u - m) \rceil$ for $u > 2m$;

L.Yu. B., *Applying dissipative dynamical systems to pseudorandom number generation: Equidistribution property and statistical independence of bits at distances up to logarithm of mesh size*, Europhysics Letters (EPL) 95, 10003 (2011).

L.Yu. B., *Geometric and statistical properties of pseudorandom number generators based on multiple recursive transformations*, Springer Proceedings in Mathematics and Statistics, Volume 23, p. 265-280 (2012).

[book chapter in “Monte Carlo and Quasi-Monte Carlo Methods 2010”, Ed. by Henryk Wozniakowski and Leszek Plaskota, Springer-Verlag, 2012. xii,732 pp, ISBN 978-3-642-27439-8].

Table 2: Initialization of pseudorandom streams for RNGs in PRAND library

Function initializing sequence	Number of sequences	Maximal length
gm19_init_sequence_	1000	$6 \cdot 10^6$
gm29_init_short_sequence_	10^8	$8 \cdot 10^7$
gm29_init_medium_sequence_	10^6	$8 \cdot 10^9$
gm29_init_long_sequence_	10^4	$8 \cdot 10^{11}$
gm31_init_short_sequence_	10^9	$8 \cdot 10^7$
gm31_init_medium_sequence_	10^7	$8 \cdot 10^9$
gm31_init_long_sequence_	10^5	$8 \cdot 10^{11}$
gm55_init_short_sequence_	10^{18}	10^{10}
gm55_init_long_sequence_	$4 \cdot 10^9$	10^{20}
gq58x1_init_short_sequence_	10^8	$8 \cdot 10^7$
gq58x1_init_medium_sequence_	10^6	$8 \cdot 10^9$
gq58x1_init_long_sequence_	10^4	$8 \cdot 10^{11}$
gq58x3_init_short_sequence_	$2 \cdot 10^8$	$8 \cdot 10^7$
gq58x3_init_medium_sequence_	$2 \cdot 10^6$	$8 \cdot 10^9$
gq58x3_init_long_sequence_	$2 \cdot 10^4$	$8 \cdot 10^{11}$
gq58x4_init_short_sequence_	$3 \cdot 10^8$	$8 \cdot 10^7$
gq58x4_init_medium_sequence_	$3 \cdot 10^6$	$8 \cdot 10^9$
gq58x4_init_long_sequence_	$3 \cdot 10^4$	$8 \cdot 10^{11}$
gm61_init_sequence_	$1.8 \cdot 10^{19}$	10^{10}
gm61_init_long_sequence_	$4 \cdot 10^9$	$3 \cdot 10^{25}$
lfsr113_init_sequence_	$3.8 \cdot 10^{18}$	10^{10}
lfsr113_init_long_sequence_	$4 \cdot 10^9$	10^{24}
mrg32k3a_init_sequence_	10^{19}	10^{38}
mt19937_init_sequence_	10^{19}	10^{130}

Ошибка вычисления интеграла для простого метода Монте-Карло: $\frac{\sigma}{\sqrt{n}}$

$$\sigma^2 = \langle f^2 \rangle - \langle f \rangle^2, \quad \text{где} \quad \langle f^2 \rangle = \frac{1}{n} \sum_{i=1}^n f(x_i)^2.$$

Методы уменьшения дисперсии:

$$p(x) = \frac{|f(x)|}{\int |f(x)| dx}, \quad F = \int_R \left(\frac{f(x)}{p(x)} \right) p(x) dx,$$

1. Выборка по значимости:

$$p(u_1, \dots, u_d) = p_1(u_1)p_2(u_2) \dots p_d(u_d).$$

2. Стратифицированная выборка: итерационный метод с разбиением на гиперкубы и с количеством точек пропорциональным дисперсии в гиперкубе для уменьшения дисперсии полного интеграла от итерации к итерации.

Схема вычислений

1. Инициализация параметров
2. Генерация N пробных точек, каждая из которых является вектором из k псевдослучайных чисел, если размерность равна k .
3. Вычисление подынтегральной функции на множестве пробных точек.
4. Суммирование значений подынтегральной функции и их квадратов для всей области интегрирования, а также в каждом гиперкубе. Вычисление средних величин и дисперсий.
5. Оптимизация сетки и других параметров после вычисления значений.
6. Повторение шагов 2–5 вплоть до M итераций

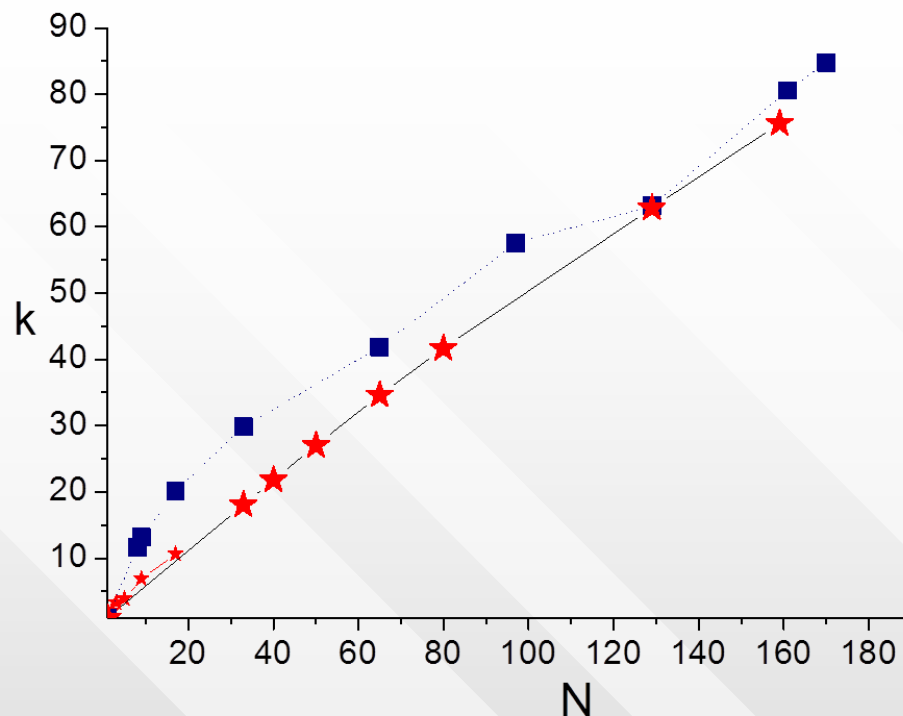


Рисунок 1. Зависимость эффективности параллелизации $k = T(1)/T(N)$ от N , где $T(N)$ – время вычисления многомерного интеграла с заданной точностью (не зависящей от N) при запуске вычисления в режиме N `tri`-узлов. Звезды обозначают вычисления на суперкомпьютере «Ломоносов» Московского государственного университета. Квадраты обозначают вычисления на суперкомпьютере «К-100» Института прикладной математики им. М.В. Келдыша Российской академии наук. Программное обеспечение было разработано при помощи технологий MPI и CUDA.

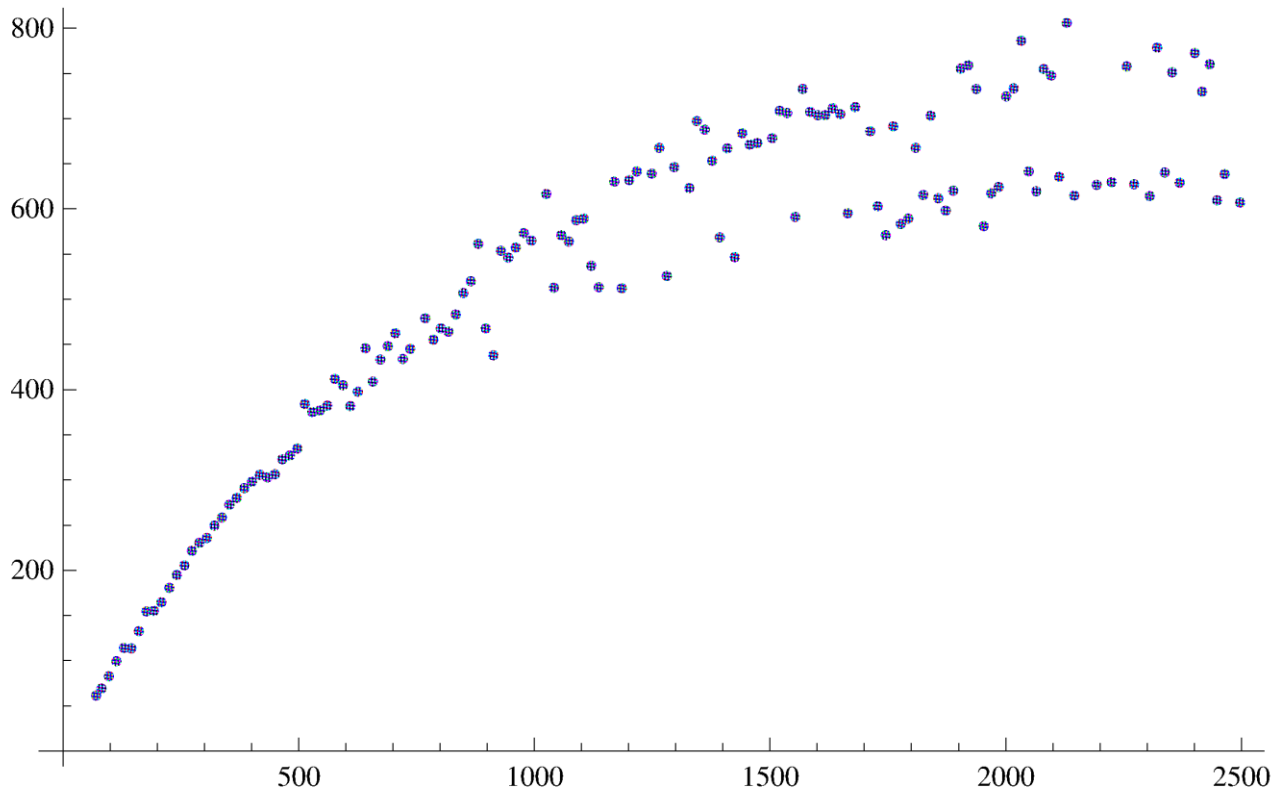


Рисунок 2. Зависимость эффективности параллелизации $k = T(1)/T(N)$ от N , где $T(N)$ – время вычисления многомерного интеграла с заданной точностью (не зависящей от N) при запуске вычисления в режиме N три-узлов. Вычисления были проведены на суперкомпьютере «Ломоносов» Московского государственного университета. Программное обеспечение было разработано при помощи технологий MPI и CUDA.

Спасибо за внимание!

Литература

1. T. Hahn, *CUBA — a library for multidimensional numerical integration*, Computer Physics Communications 176 (2007) 712–713.
2. J. Kanzaki, *Monte Carlo Integration in GPU*, Eur. Phys. J. C (2011)71:1559.
3. L.Yu. Barash, L.N. Shchur, *RNGSSELIB: Program library for random number generation. More generators, parallel streams of random numbers and Fortran compatibility*, Computer Physics Communications 184 (2013) 2367.
4. L.Yu. Barash, L.N. Shchur, *PRAND: GPU accelerated parallel random number generation library*, Computer Physics Communications 185 (2014) 1343.
5. Л.Ю. Бараш, Л.Н. Щур, Использование гибридных суперкомпьютеров для многомерного численного интегрирования методом Монте-Карло, готовится к печати