ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 137 за 2019 г.

ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Белов А.А., Калиткин Н.Н., Тинтул М.А.

Визуальная верификация генераторов псевдослучайных чисел

Рекомендуемая форма библиографической ссылки: Белов А.А., Калиткин Н.Н., Тинтул М.А. Визуальная верификация генераторов псевдослучайных чисел // Препринты ИПМ им. М.В.Келдыша. 2019. № 137. 28 с. http://doi.org/10.20948/prepr-2019-137
URL: http://library.keldysh.ru/preprint.asp?id=2019-137

Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук

А. А. Белов, Н. Н. Калиткин, М. А. Тинтул

ВИЗУАЛЬНАЯ ВЕРИФИКАЦИЯ ГЕНЕРАТОРОВ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

Москва — 2019

Белов А. А. , Калиткин Н. Н. , Тинтул М. А.

ВИЗУАЛЬНАЯ ВЕРИФИКАЦИЯ ГЕНЕРАТОРОВ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

Рассмотрена проблема построения последовательностей равномерно распределенных псевдослучайных чисел. Использован простой визуальный критерий для оценки случайности чисел последовательности. Этот тест показал, что наиболее распространенные современные генераторы случайных чисел, основанные на методе вихря Мерсенна, линейной конгруэнтной последовательности и ряде других принципов, дают неудовлетворительные результаты. Поэтому проблема построения хороших генераторов остается нерешенной, а к результатам расчета случайных процессов (метод молекулярной динамики и др.) следует относиться с осторожностью.

Ключевые слова: методы Монте-Карло, псевдослучайные числа, тестирование

Aleksandr Aleksandrovich Belov, Nikolai Nikolaevich Kalitkin, Maksim Aleksandrovich Tintul

Unreliability of pseudo-random number generators

We consider construction of uniformly distributed pseudo-random number sequences. To estimate the sequence randomness, simple visual criterion is implied. This test shows the most widespread generators based on the Mersenne twister, linear congruent sequence and some other principles do not provide satisfactory results. Therefore, construction of good generators is still an unsolved problem. Also, results of stochastic process calculations (e.g., molecular dynamics etc.) should be treated with caution.

Key words: Monte-Carlo methods, pseudo-random numbers, testing

Работа поддержана грантами РФФИ № 18-01-00175 и МК-1780.2019.1

1. Проблема

Методы Монте-Карло широко используются в различных областях прикладной математики: вычисление многомерных интегралов, разыгрывание столкновений в методах молекулярной динамики и задачах реакторной защиты, оценка отказов в сложных конструкциях и многие другие. Все эти методы используют некоторые последовательности случайных чисел с заданной плотностью распределения. Чаще всего такие задачи сводятся к использованию последовательности случайных чисел γ_n, равномерно распределенных на отрезке [0,1]. Эти последовательности имитируют наборами чисел, получаемых с помощью некоторых математических алгоритмов. Такие наборы называют ***псевдослучайными.*** Все эти алгоритмы являются по сути эвристическими.

Нередко в приложениях требуются случайные числа с другими плотностями распределения $\rho(x)$ – гауссовым, пуассоновым и т.п. Разыгрывание такой случайной величины можно формально свести к разыгрыванию переменной γ_n:

$$\gamma_n = \int_{-\infty}^{x_n} \rho(x)dx \qquad (1)$$

Разыгрывая γ_n и решая уравнение (1), находим значение x_n. Этот способ удобен, если интеграл берется в элементарных функциях, и полученное уравнение легко решается. В противном случае могут использовать аппроксимацию интеграла по формуле трапеций или прямоугольников. Например, так реализован способ получения гауссовых чисел в среде Matlab. Очевидно, качество полученной последовательности значений x_n не может быть лучше, чем качество последовательности γ_n.

Любой предлагаемый алгоритм обычно тестируют – проверяют, насколько выполняются те или иные свойства, характерные для чисто случайных последовательностей. Например, близко ли выборочное среднее от γ_n к 1/2; существенна ли корреляция между γ_n и γ_{n+1}, равномерность распределения точек в кубах разной размерности и т.д. В [1] тестировалось 13 популярных генераторов псевдослучайных чисел. При этом использовалась стандартная библиотека тестов Diehard, включавшая проверку равномерности заполнения единичного куба с числом измерений до 15. Автор работы констатирует, что

последовательности удовлетворяют тестам, однако он считает, что это не гарантирует хорошего качества последовательностей. Поэтому систему формальных математических тестов нельзя считать достаточно надежной.

Обратим внимание на то, что авторы, разрабатывающие генераторы псевдослучайных чисел, обычно стремятся получить как можно более длинные периоды последовательностей. Однако при огромной длине последовательности использовать целую замкнутую цепочку чисел практически невозможно. В реальных расчетах каждый раз используется небольшая доля полной последовательности. Поэтому даже если полная последовательность имеет хорошие свойства равномерности, то таковые свойства не гарантированы для произвольно выбранной малой доли последовательности. При теоретическом исследовании этот аспект обычно упускают из виду, а свойства отдельных случайно выбранных участков последовательности можно проверить только эвристически.

Математики дают путевку в жизнь только тем генераторам, которые проходят подобные тесты. Затем физики, которые не всегда хорошо знают математику, но очень часто ее обожествляют, доверчиво используют эти последовательности в своих расчетах. Рассмотрим критически эту проблему. Любой математический алгоритм является детерминированной процедурой, поэтому любая псевдослучайная последовательность не может быть истинно случайной и должна содержать какие-то закономерности. Вопрос лишь в том – насколько они существенны. Мы протестировали ряд широко употребительных алгоритмов и убедились, что их вряд ли можно считать надежными. Данная работа посвящена иллюстрации этих выводов.

2. Простейший визуальный тест

Тесты псевдослучайных последовательностей можно условно разбить на две группы: статистические и визуальные. В статистических тестах проверяются важнейшие соотношения между членами последовательности g_n. Например, можно вычислять выборочные моменты и корреляции исследуемой последовательности и сравнивать их с теоретически ожидаемыми для данного распределения. Однако для тщательного исследования требуется вычислять моменты и корреляции высоких порядков. Это слишком трудоемко. Обычно

ограничиваются несколькими первыми моментами и парными корреляциями, что не обеспечивает надежность тестирования.

Визуальные тесты не являются строгими, зато они очень просты и наглядны. Например, очень популярен следующий тест. В единичном кубе строят псевдослучайные точки по традиционному правилу: первые три числа g_1, g_2, g_3 являются координатами x, y, z первой случайной точки, вторые три числа – координатами второй случайной точки и т.д. Затем используют программу проекции куба на экран компьютера и вращают куб в пространстве. Как правило, при некоторых углах обзора удается наблюдать на экране неравномерность распределения точек, причем значительную. Обычно точки группируются вблизи некоторых параллельных плоскостей в трехмерном пространстве, а в промежутках между этими плоскостями плотность распределения точек существенно меньше. Эта процедура требует хорошей квалификации тестирующего, иначе можно прозевать соответствующий угол поворота.

Заметим, что в этом сложном способе мы наблюдаем не всю трехмерную картину, а лишь ее проекцию на плоскость. Поэтому гораздо проще с самого начала ограничиться плоским случаем – исследованием равномерности распределения точек в единичном квадрате. Получаемую картину легко оценить визуально, причем для этого не нужно выполнять вручную какие-либо дополнительные операции, например, вращение.

При этом отметим одну существенную техническую деталь. Размер маркеров на экране или при выводе на печать надо выбирать так, чтобы в местах максимального сгущения точек они почти перекрывались бы. Для этого суммарная площадь всех маркеров должна составлять 15-25% от площади единичного квадрата (таким образом, размер маркера зависит от числа выведенных точек).

Нередко визуальный контроль объявляют субъективным. Однако известно, что глаз зачастую лучше выявляет закономерности, чем формальные математические методы. Например, глаз легко оценивает выход последовательности точек на асимптоту. Формальные математические критерии требуют гораздо большего числа точек для надежного установления такого выхода. Это хорошо известно тем, кто разрабатывает методики автоматического контроля точности расчетов.

Другой пример, знакомый не только математикам – парадокс телеграфного провода. Среднее разрешение человеческого глаза составляет 1 угловую минуту. Однако человек видит телеграфный провод даже тогда, когда наблюдаемый диаметр провода составляет 7 угловых секунд!

Рассмотренные генераторы. Мы исследовали 15 генераторов псевдослучайных чисел, широко распространенных в отечественной и зарубежной литературе, а также в коммерческих пакетах:

· вихрь Мерсенна (Mersenne twister) [2];
· быстрый вихрь Мерсенна (SIMD-oriented fast Mersenne twister) [3];
· мультипликативный конгруэнтный генератор (Multiplicative congruential generator) [4];
· мультипликативный генератор Фибоначчи с запаздыванием (64-bit multiplicative lagged Fibonacci generator) [5],
· комбинированный множественный рекурсивный генератор (combined multiple recursive generator) [6];
· генератор Philo4x32 [7];
· генератор Threefry4x64 [7];
· генератор Марсалья (Marsaglia's SHR3 shift-register generator) [8];
· модифицированный генератор Subtract-with-Borrow (modified Subtract-with-Borrow generator) [9];
· rand на языке C/C++;
· модифицированная последовательность Лемера [10];
· последовательность чисел из сборника [11];
· одинарные точки Бахвалова [12]
· симметричные точки Бахвалова [12]
· последовательности Соболя [13,14];

Этот список частично совпадает с последовательностями, тестированными в [1]. Дадим краткие характеристики этих алгоритмов.

1⁰ Вихрь Мерсенна – это весьма сложный алгоритм. В нем генерируются две различные независимые последовательности 32-битовых целых чисел, из которых составляется одно 64-битовое число. Поэтому окончательный результат соответствует точности double precision. Такая комбинация двух последовательностей обеспечивает период 2^{19937}-1, далеко превосходящий все

ожидания современных вычислителей. Этот алгоритм считается одним из наиболее совершенных.

2⁰ Быстрый вихрь Мерсенна считается более быстрой реализацией предыдущего алгоритма. Однако проверить, приводят ли оба алгоритма к строго одинаковым результатам, непросто. Формально пользователь может задавать начало последовательности. Однако алгоритмы не являются идентичными, и фактически начала последовательностей будут различными.

3⁰ Мультипликативный конгруэнтный генератор – это алгоритм Лемера с некоторыми конкретными константами:

$$m_{n+1} = (Am_n + C) \bmod M, \quad A = 7^5, \quad C = 0, \quad M = 2^{32} - 1. \qquad (2)$$

Этот алгоритм прост и легко программируется. Он использует одну 32-битовую последовательность целых чисел, и его период не может превышать M. Это несравненно меньше, чем для вихря Мерсенна. Однако вряд ли это является серьезным недостатком, учитывая замечания, сформулированные в разделе 1.

4⁰ 64-битовый мультипликативный генератор Фибоначчи с запаздыванием использует одну 64-битовую последовательность целых чисел. Его период составляет приблизительно 2^{124}. Это много больше, чем нужно для современных приложений. Константы запаздывания равны $l = 63,\ k = 31.$

5⁰ Комбинированный множественный рекурсивный генератор использует две 32-битовые последовательности целых чисел. Рекурсивная процедура обеспечивает длину периода 2^{191}.

6⁰ Philox 4x32 – генератор с использованием четырех независимых последовательностей 32-битовых чисел. По-видимому, этот генератор ориентирован на задачи криптографии. В каждой последовательности нахождение m_{n+1} через m_n осуществляется с помощью сети Фейстеля с некоторыми наборами ключей. Период составной последовательности равен 2^{193}.

7⁰ Threefry 4x64 – это адаптация блочного криптографического алгоритма Threefish из хэш-функции Skein. Она использует четыре 64-битовых последовательности и обеспечивает период 2^{514}.

8⁰ Генератор Марсалья использует линейный конгруэнтный алгоритм (2) с параметрами $A = 69069,\ M = 2^{32},\ C = 1234567$. Вычисления производятся с 32-битовыми числами. Каждое полученное по формуле (2) число подвергается следующей процедуре. Сначала оно сдвигается влево на 5 регистров и

суммируется с исходным числом. Затем полученное число сдвигается на 17 регистров вправо и оба этих числа суммируются. Новое число сдвигается на 13 регистров влево, и также выполняется суммирование. Все указанные суммирования выполнялись с 64 разрядами. Период полученной последовательности оценивался как 2^{64}. Использование в качестве $M = 2^{32}$ степени числа 2 удешевляет вычисления, но такой выбор может ухудшить качество полученных чисел.

9⁰ Модифицированный генератор Subtract-with-Borrow аналогичен генератору Фибоначчи с задержкой (константы задержки 27 и 12). Модификация, описанная в [10], позволила увеличить период до 2^{1492}.

10⁰ Генератор rand языка C/C++ использует алгоритм Лемера и 16-битовые целые числа. Период такой последовательности не может превышать 2^{16}, то есть он очень невелик и годится лишь для учебных расчетов. Несмотря на это, генератор очень популярен (быть может, потому, что он является стандартным генератором в C/C++).

11⁰ Модифицированный алгоритм Лемера использует три независимых последовательности 16-битовых целых чисел. Он был распространен около 1990-го года, когда широко использовались 16-разрядные PC XT. Период такого генератора не может превышать 2^{48}.

Во всех указанных выше программах начало последовательности либо детерминировано, либо случайно (и нередко привязывается к часам компьютера).

12⁰ В 1955 году американской корпорацией RAND была опубликована книга [11], содержащая миллион случайных цифр и сто тысяч стандартных нормальных отклонений. Эти случайные цифры были получены с помощью электронной рулетки (электронного источника шумов), то есть физического прибора, а не математического алгоритма. Поэтому довольно распространено мнение, что такие числа являются действительно случайными, то есть имеют преимущества перед математическими псевдослучайными числами.

Однако один из авторов данной работы наблюдал в 1950-е годы, как выполнялись расчеты методом Монте-Карло на ЭВМ, содержащих встроенную электронную рулетку. Сотрудница, которая вела эти расчеты, начинала с того, что лично проверяла регулировку поданного на ЭВМ напряжения. От этого напряжения ощутимо зависели результаты расчетов. Вспомним, что ни один

источник напряжения не является абсолютно стабильным. Если к этой сети подключаются новые нагрузки, то происходит небольшой скачок напряжения, релаксирующий к заданной норме. Такие подключения происходят постоянно. Поэтому электронную рулетку нельзя считать надежным источником случайных чисел.

13⁰ Одинарные точки Бахвалова строятся следующим образом. Сторона p-мерного куба разбивается на n равных частей. Соответственно куб разбивается на n^p одинаковых кубиков. В каждом кубике ставится одна случайная точка. В данной работе для постановки этих случайных точек был использован генератор rand языка C/C++.

Доказано, что многомерные кубатуры с использованием одинарных бахваловских точек имеют погрешность не $O(N^{-1/2})$ как для обычных псевдослучайных точек, а $O(N^{-1/2-1/p})$. Например, для $n=2$ это составляет $O(N^{-1})$. При размерности пространства $p<6$ это дает существенный выигрыш в точности.

14⁰ Чтобы получить симметричные точки Бахвалова, в каждом кубике выбирается одна случайная точка. Затем строится еще одна точка, симметричная первой относительно центра кубика. Естественно ожидать, что при использовании одного и того же генератора симметричные и ординарные точки будут заполнять многомерный куб более равномерно, чем псевдослучайные. Однако в первом случае увеличится корреляция, так как генератор работает в значительно более узком диапазоне, что приводит к ухудшению статистических свойств получаемой последовательности.

Для симметричных точек Бахвалова точность многомерных кубатур возрастает до $O(N^{-1/2-2/p})$. Такие кубатуры сохраняют выигрыш в точности до $p \approx 8$.

15⁰ Особняком стоят последовательности Соболя [13,14]. Это название широко распространено в зарубежной литературе, а сам автор назвал их $ЛП_t$ - последовательностями. Эти числа следует называть не псевдослучайными, а квазислучайными. Они строятся так, что при магических числах точек $N=2^k$ проекции всех точек на любую ось единичного куба различаются и образуют равномерную сетку с шагом $2^{-k}=1/N$. При промежуточных N часть узлов этой сетки остаются незанятыми. Период такой

последовательности неограничен. В [14] приведены таблицы для построения этих последовательностей при размерности пространства $p \pounds 13$; но в настоящее время такие таблицы разработаны для $p \sim 4000$.

Для широкого круга многомерных кубатур числа Соболя имеют хорошую точность при $p \pounds 13$. Однако есть ряд задач, где преимущества этих последовательностей сказываются даже при гораздо б*о*льших размерностях.

3. Результаты тестирования

Зависимость от N. Естественно ожидать, что при увеличении N распределение точек будет все более равномерным. Подробные расчеты были проведены для генератора Мерсенна для $N = 100, 300, 1000, 3000, 10000$; при дальнейшем увеличении N точки сливались, и визуальное исследование становилось проблематичным. Начало последовательности было одинаковым во всех этих расчетах. Типичные примеры для $N = 1000$ и $N = 3000$ приведены на рис. 1 и 2 соответственно. На рис. 1 диаметр точек в $\sqrt{3}$ раз больше, чем на рис. 2, чтобы суммарная площадь точек была одинаковой.

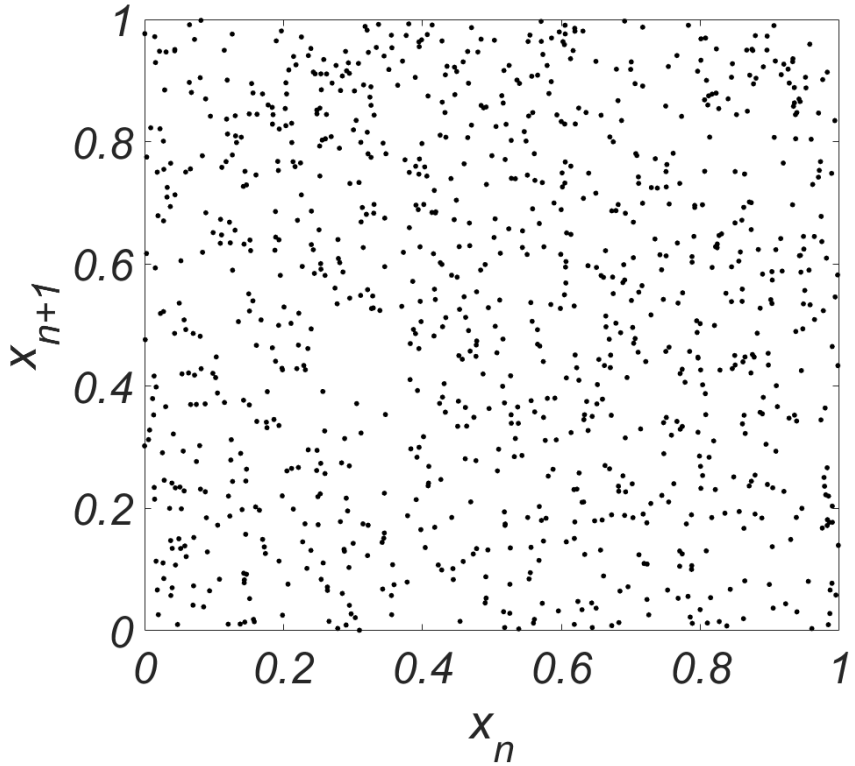

Рис.1. Вихрь Мерсенна, $N = 1000$.

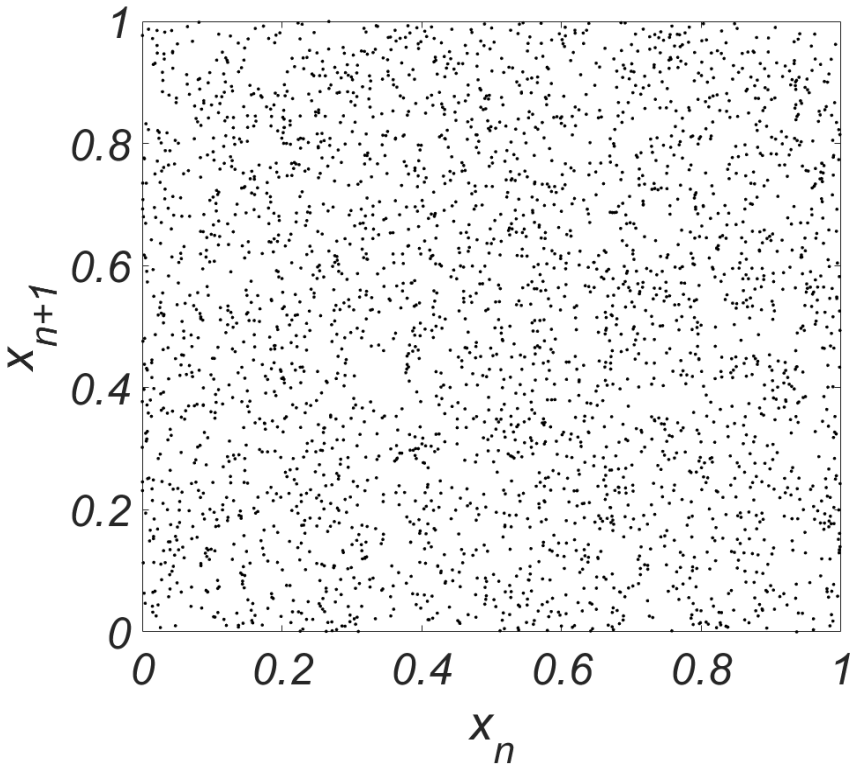

Рис.2. Вихрь Мерсенна, $N = 3000$.

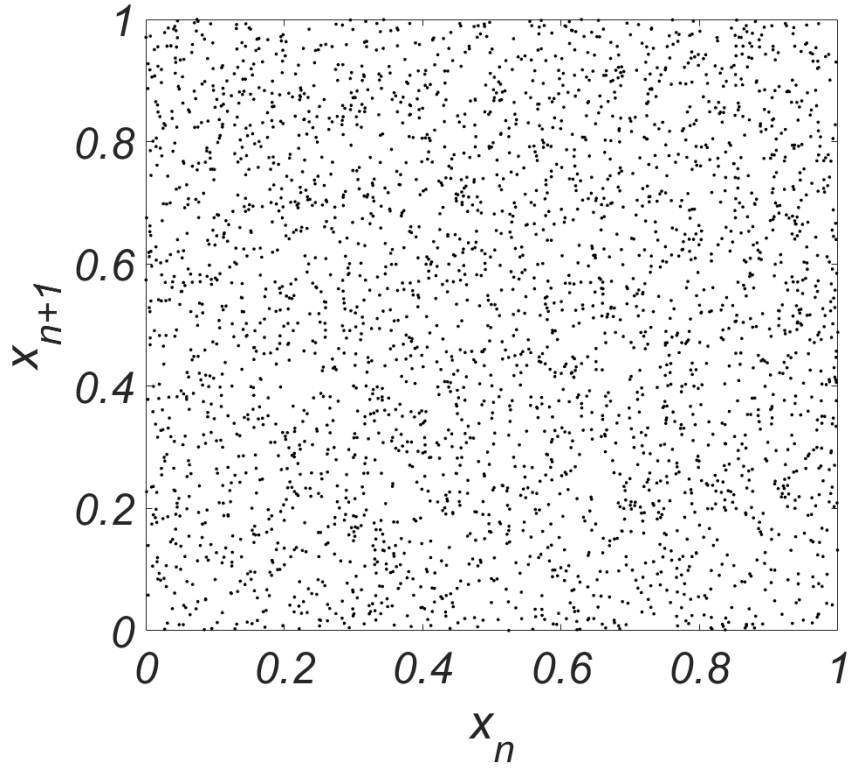

Рис.3. Вихрь Мерсенна, $N = 3000$, случайное начало последовательности.

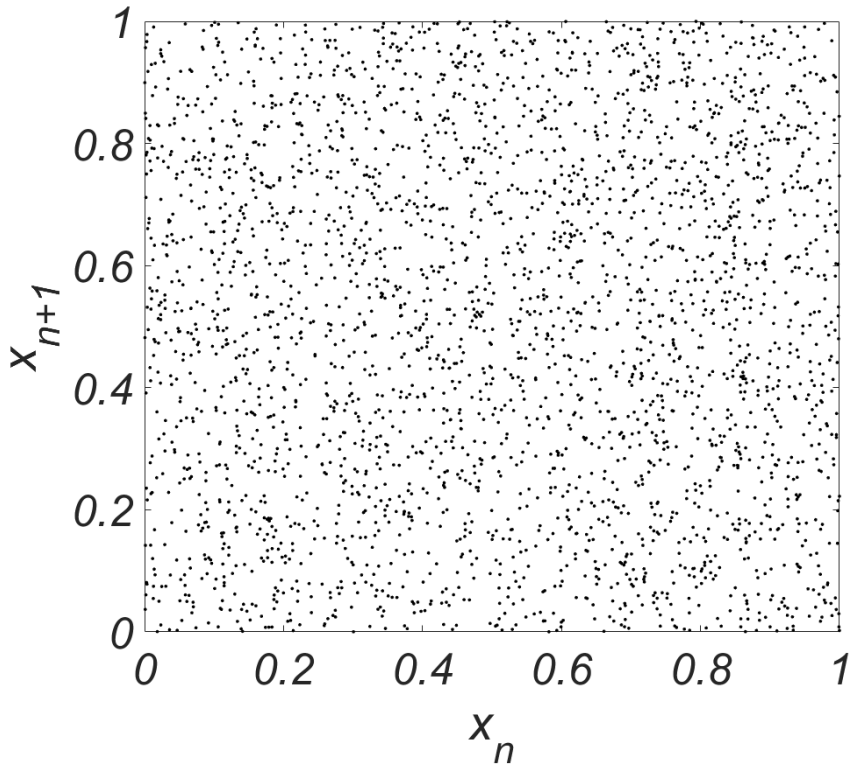

Рис.4. Быстрый вихрь Мерсенна, $N = 3000$.

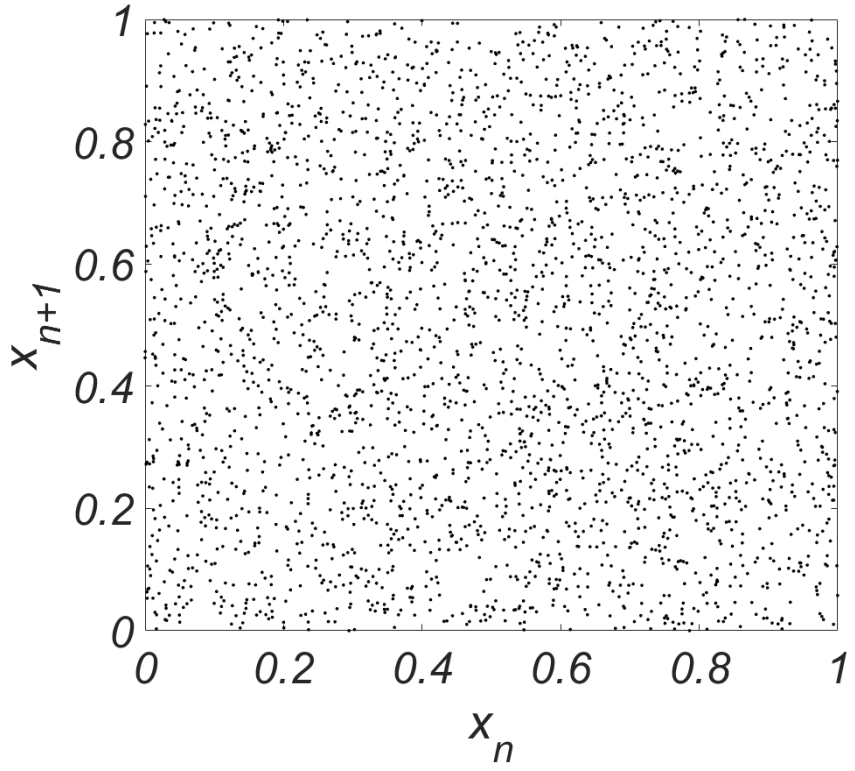

Рис.5. Мультипликативный конгруэнтный генератор, $N = 3000$.

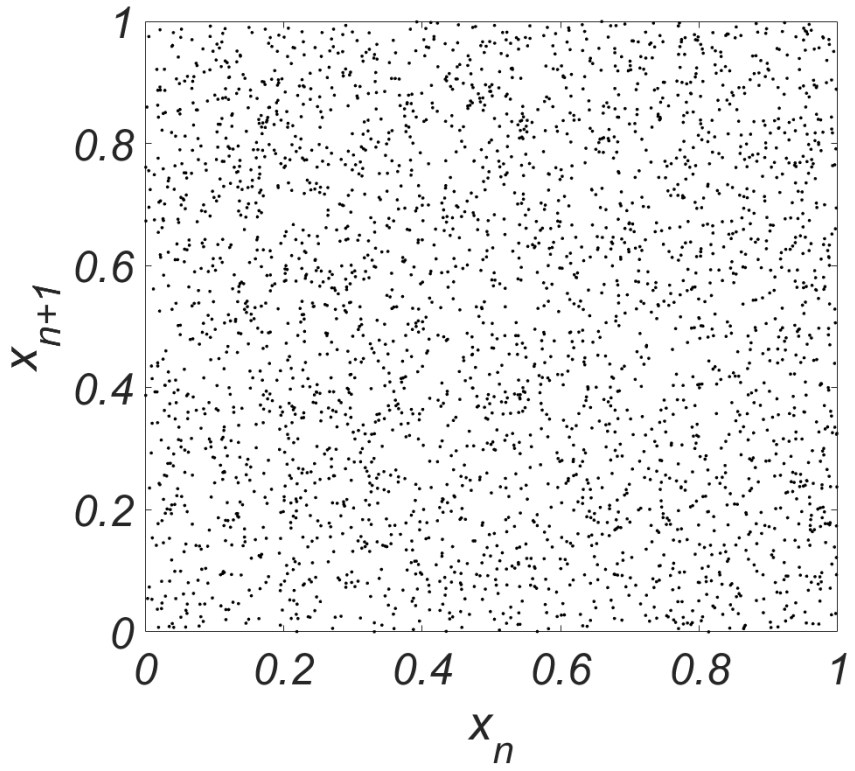

Рис.6. Мультипликативный генератор Фибоначчи с запаздыванием, $N = 3000$.

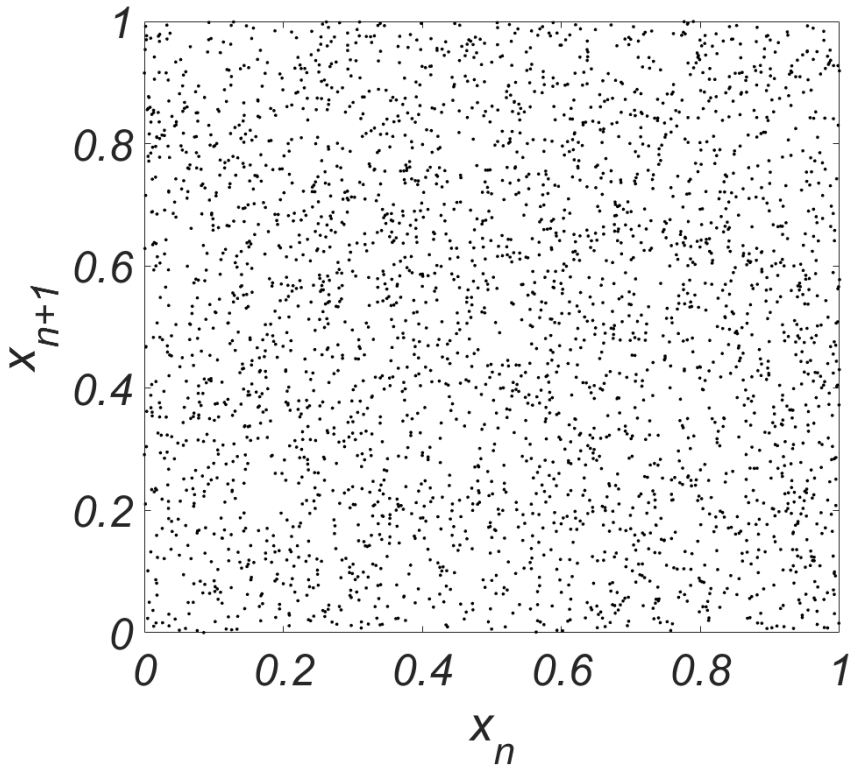

Рис.7. Комбинированный множественный рекурсивный генератор, $N = 3000$.

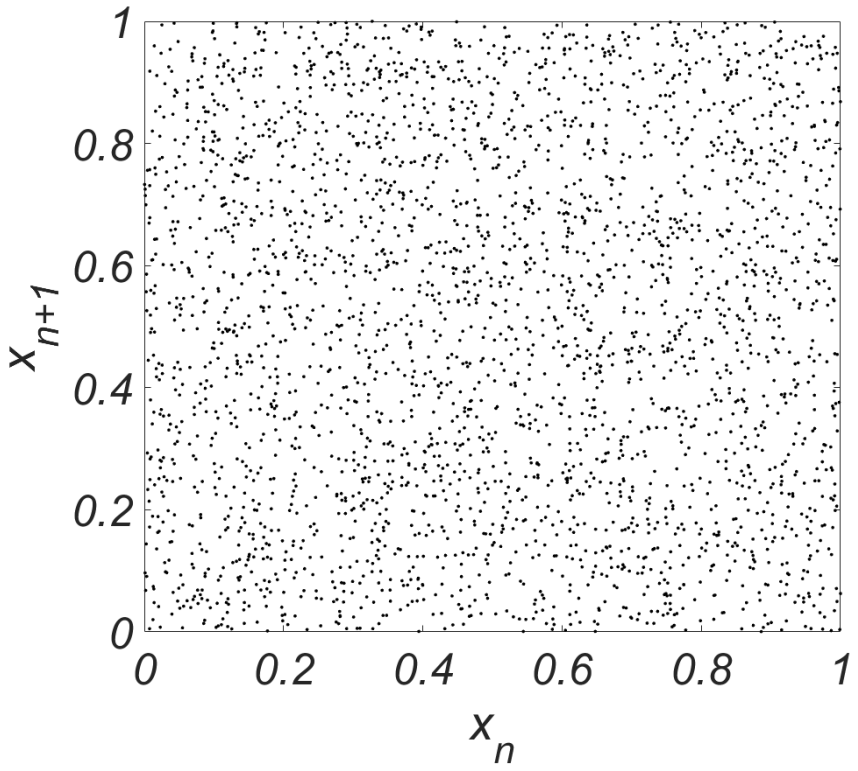

Рис.8. Генератор Philox 4x32, $N = 3000$.

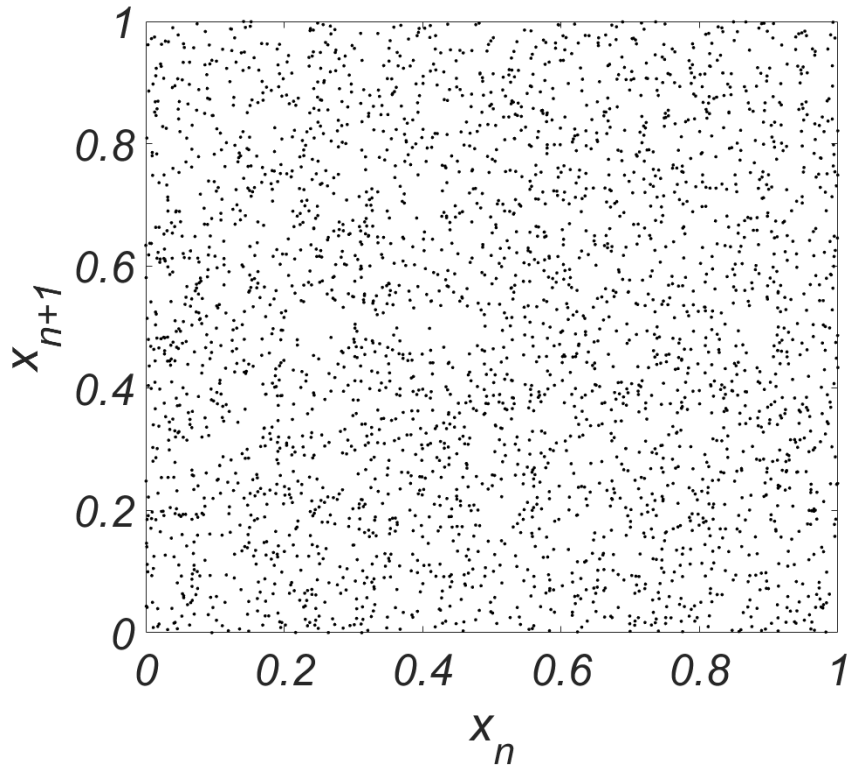

Рис.9. Генератор Threefry 4x64, $N = 3000$.

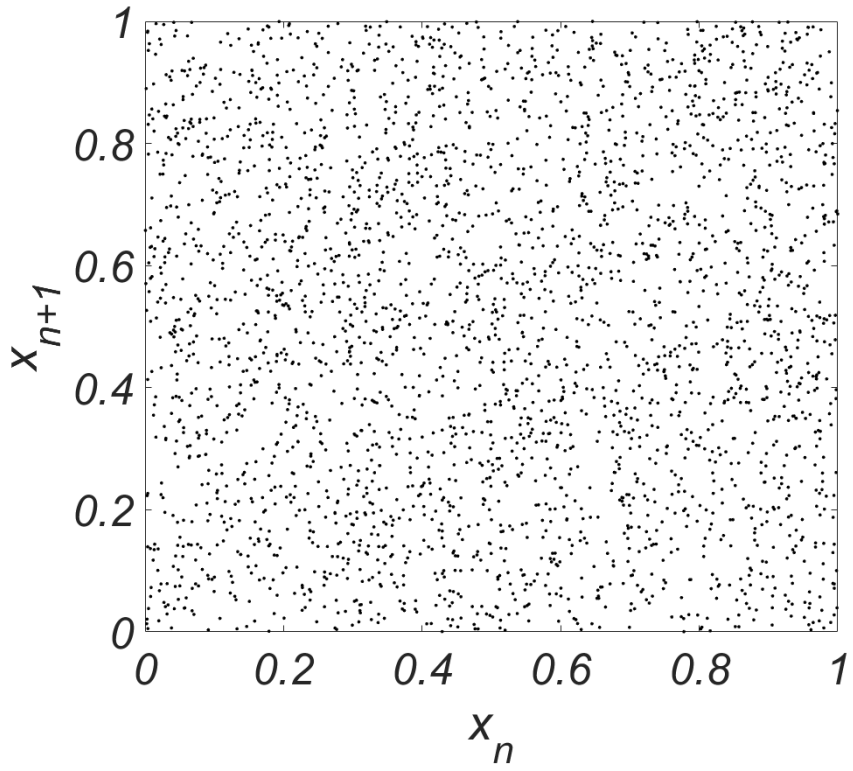

Рис.10. Генератор Марсалья, $N = 3000$.

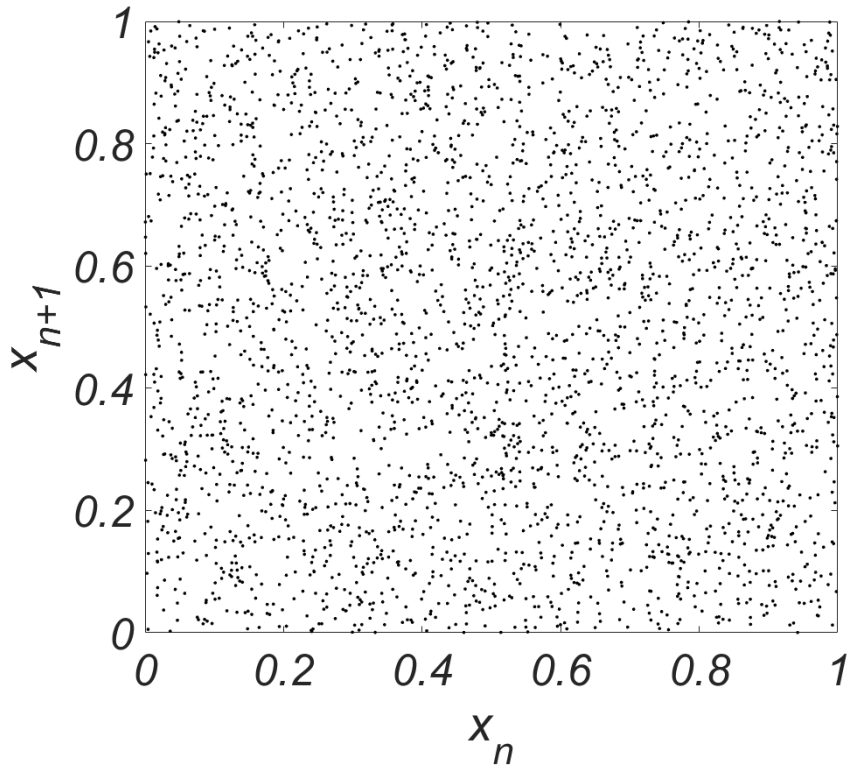

Рис.11. Модифицированный генератор Subtract-with-Borrow, $N = 3000$.

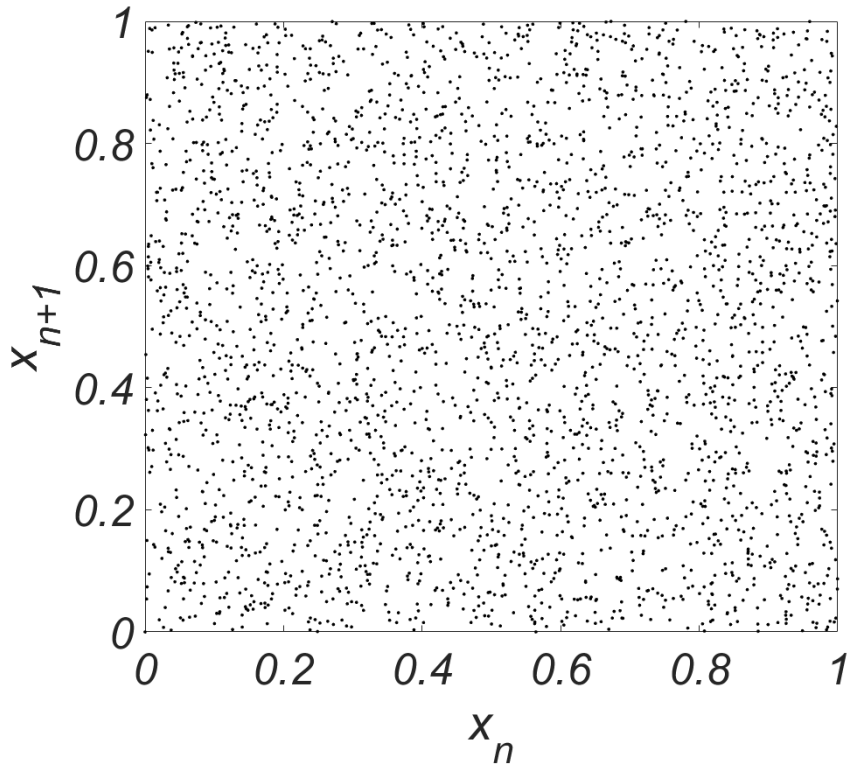

Рис.12. Генератор rand языка C/C++, $N = 3000$.

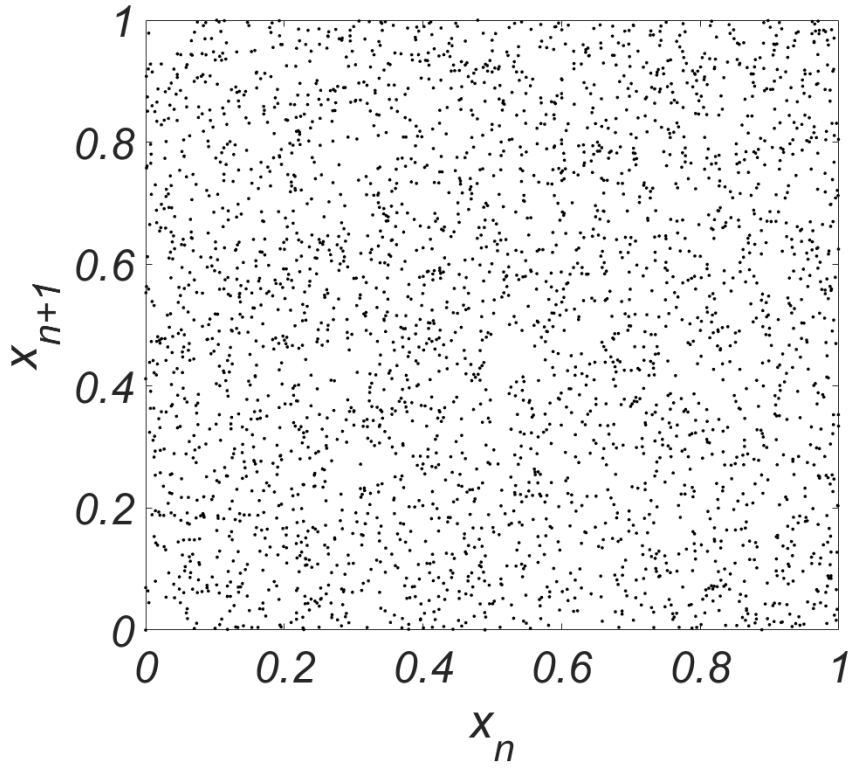

Рис.13. Генератор rand языка C/C++, $N = 3000$, другое начало последовательности

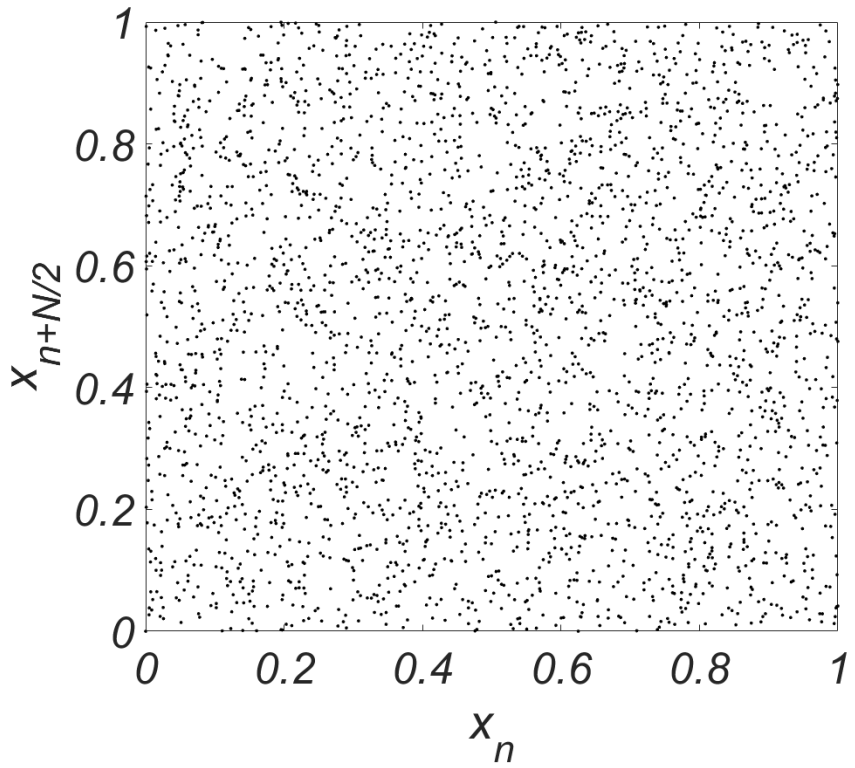

Рис.14. Генератор rand языка C/C++, $N = 3000$, другое определение координат точек.

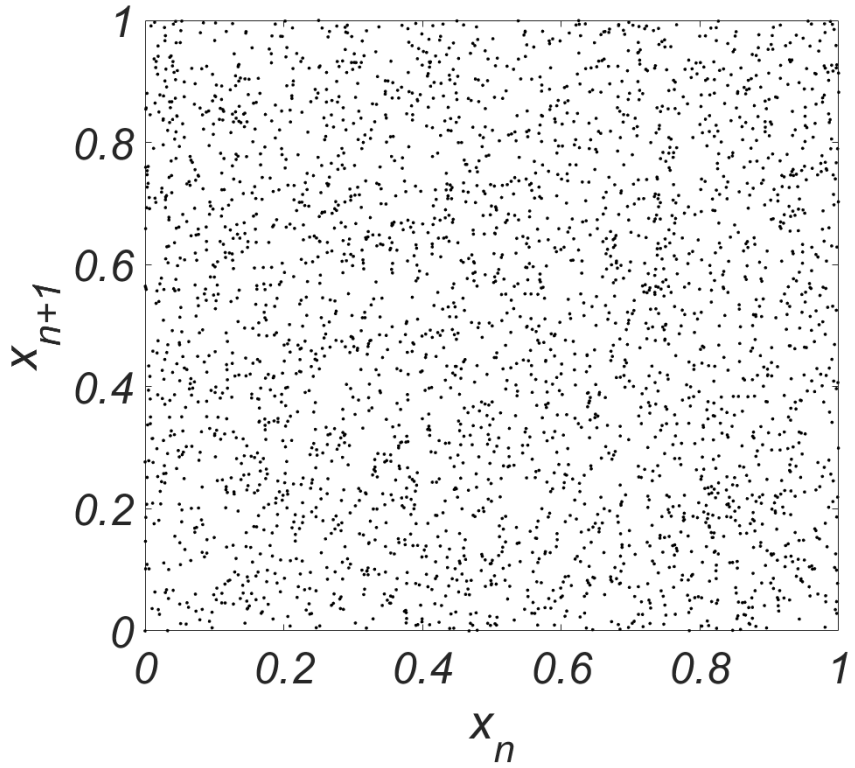

Рис.15. Модифицированная последовательность Лемера, $N = 3000$.

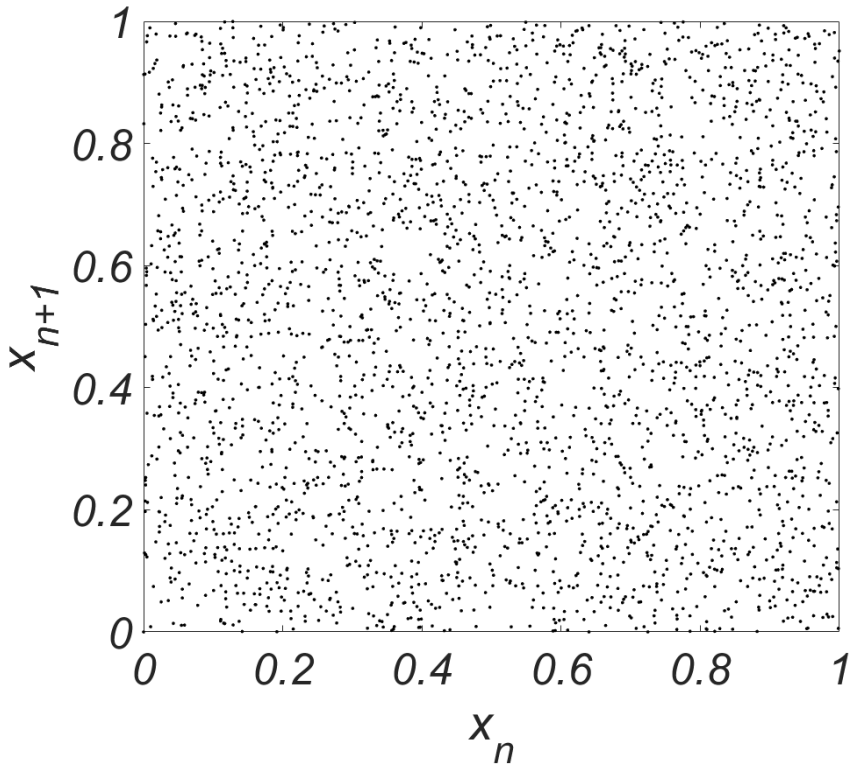

Рис.16. Последовательность чисел из сборника [11], $N = 3000$.

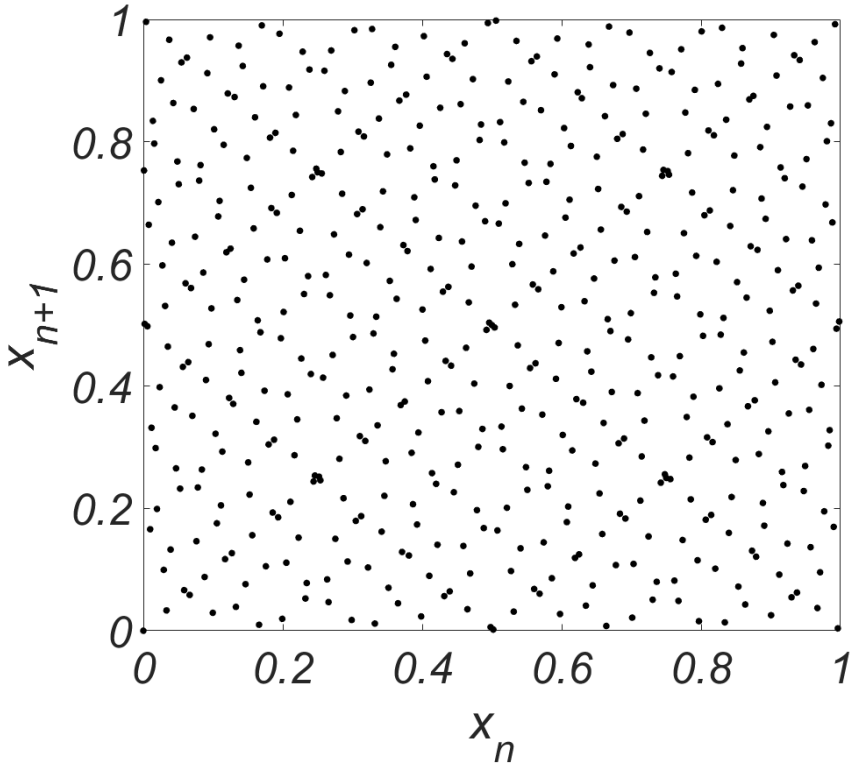

Рис.17. Последовательность Соболя, $N = 512$.

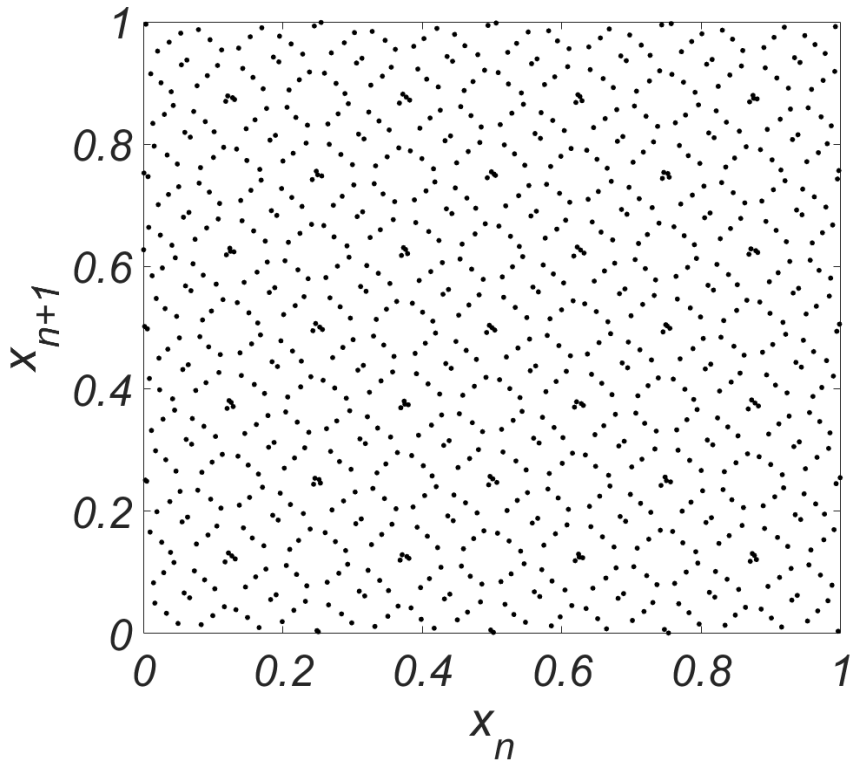

Рис.18. Последовательность Соболя, $N = 1024$.

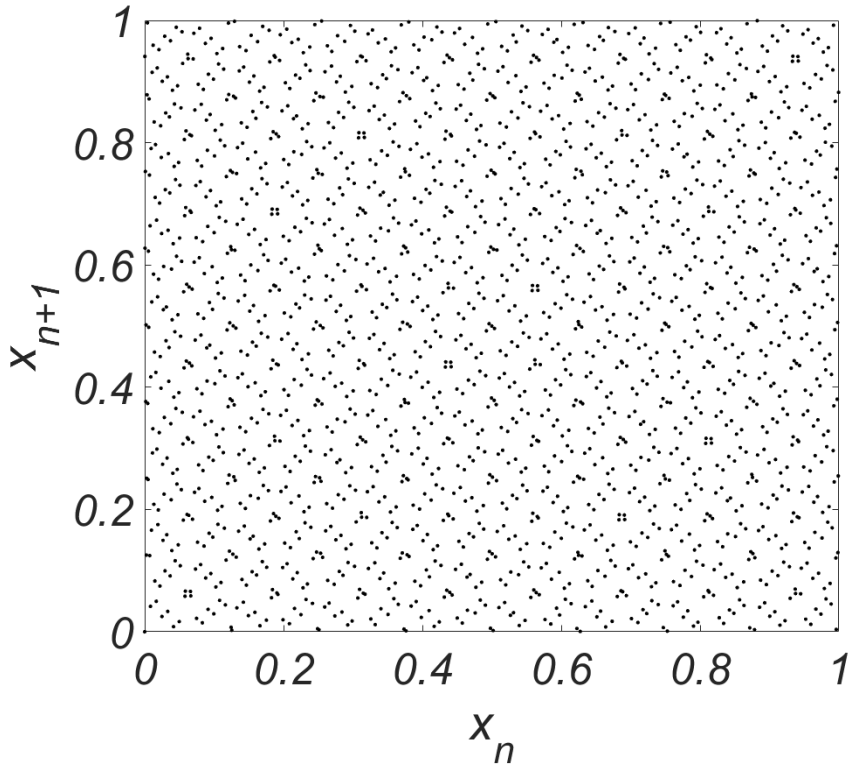

Рис.19. Последовательность Соболя, $N = 2048$.

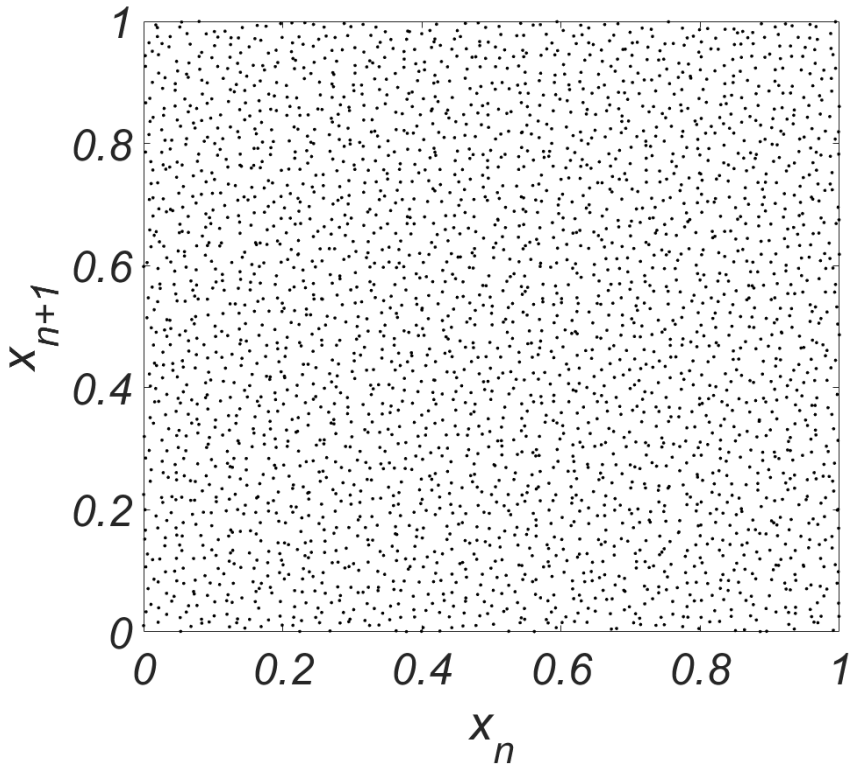

Рис.20. Одинарные точки Бахвалова, $N = 3025$.

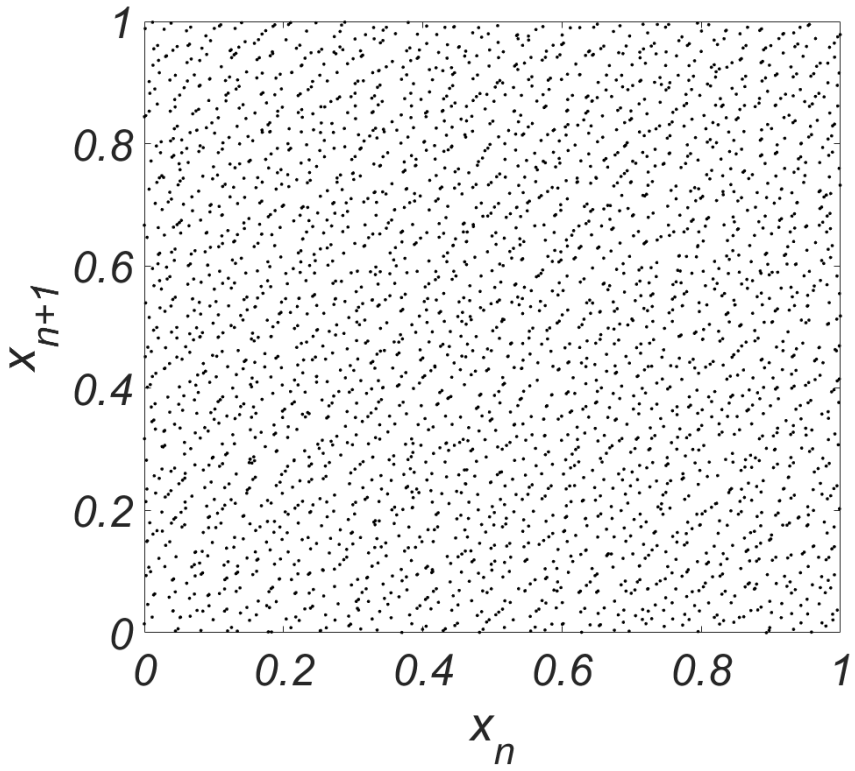

Рис.21. Симметричные точки Бахвалова, $N = 3042$.

На обоих рисунках хорошо видны области сильного сгущения и сильного разрежения точек. Важно то, что рис. 2 строится по той же последовательности с тем же началом, то есть он получен наложением на рис. 1 следующего отрезка последовательности длиной 2000. Тем не менее области сгущения и разрежения точек оказываются в других участках квадрата. При дальнейшем увеличении числа точек места сгущения и разрежения также смещаются. Но даже при очень больших $N = 10000$ неравномерность заполнения остается значительной. Это свидетельствует о существенной неравномерности заполнения квадрата и о не слишком высоком качестве участков последовательности с заметно различающимися длинами.

Начало последовательности. На рис. 3 приведен расчет для генератора Мерсенна с $N = 3000$, но со случайно выбранным началом последовательности. Сравнение с рис. 2 показывает, что неравномерность заполнения по-прежнему достаточно велика, а области наибольшей и наименьшей плотности занимают другие места. Аналогичная картина была при других начальных точках последовательности. Таким образом, все участки последовательности характеризуются значительной неравномерностью распределения точек.

Генератор rand() языка C/C++. Этому генератору было уделено особое внимание благодаря его большой популярности. Для него были проведены аналогичные эксперименты. Результаты оказались качественно такими же, как для генератора Мерсенна, хотя генератор rand() несоизмеримо проще последнего и использует только 16-битовые числа. В частности, на рис. 12 и рис. 13 приведены результаты для $N = 3000$ при двух разных началах последовательности. Видно, что площади разрежений примерно одинаковы, но разрежения находятся в разных местах квадрата.

На этом генераторе был проведен еще один интересный эксперимент. В линейной конгруэнтной последовательности каждое очередное число вычисляется по предыдущему. Поэтому потенциальной проблемой может быть значительная корреляция пары соседних чисел. Чтобы проверить это, изменим постановку вычислительного эксперимента: возьмем первые $N/2$ чисел за координаты по оси абсцисс, а вторые $N/2$ чисел – за координаты по оси ординат. Результаты показаны на рис. 14. Видно, что качественная картина осталась той же, и средняя площадь разрежений также практически не изменилась. Это говорит о том, что параметры последовательности выбраны

достаточно удачно, и коррелированность пары чисел не зависит от того, являются они соседними или далеко отстоят друг от друга.

Другие генераторы. На рис. 4-11 и рис. 15, 16 приведены результаты тестирования других 10 генераторов псевдослучайных чисел. Для всех выбрано $N = 3000$. Видно, что картина оказывается такой же, как для генераторов Мерсенна и rand(). При этом отметим, что одна из этих последовательностей RAND (рис. 16) использует не математические псевдослучайные числа, а электронную рулетку (электронный источник шумов), от которого обычно ожидают истинной случайности. Тем не менее, конструктивные особенности датчика внесли, по-видимому, неслучайность в результаты.

Эти иллюстрации показывают, что различные генераторы псевдослучайных чисел дают качественно сходные результаты. Отчетливо наблюдается неравномерность заполнения квадрата, а места сгущения и разрежения точек зависят от числа точек и выбранного участка последовательности. Нами исследована только часть опубликованных генераторов. Постоянно продолжают строиться новые генераторы псевдослучайных чисел. Тем не менее, принципиальных прорывов пока что не видно.

Точки Бахвалова. Эти точки не являются псевдослучайными. Скорее, их можно называть псевдоравномерными, поскольку содержащие их p-мерные кубики образуют равномерную p-мерную сетку. Для сравнения с описанными выше генераторами псевдослучайных чисел выбиралось такое же количество точек Бахвалова $N \sim 3000$. Для одинарных точек Бахвалова было взято разбиение стороны квадрата на 55 отрезков, то есть $N = 3025$. Для симметричных точек Бахвалова сторона квадрата разбивалась на 39 отрезков, то есть $N = 3042$.

Результаты представлены на рис. 20, 21. Видно, что по сравнению со всеми предыдущими рисунками заполнение единичного квадрата выглядит гораздо более равномерным, причем для симметричных точек Бахвалова распределение более равномерно. Это объясняет повышение точности квадратурных формул.

Однако вопрос о возможности использования подобных точек для описания динамических процессов остается открытым.

Последовательность Соболя. На рис. 17 – 19 представлены последовательности Соболя для трех соседних магических чисел

$N = 512, 1024, 2048$. Наблюдаемая картина качественно отличается от всех предыдущих рисунков. Видно, что заполнение имеет характер регулярного кружева из ~256 ячеек. На всех трех сетках просматривается одна и та же структура – 16 на 16 ячеек. При $N = 512$ на ячейку в среднем приходится 2 точки, поэтому структура очень слабо выражена; но зато заметна более грубая структура 3 на 3 ячейки. При $N = 1024$ грубая структура практически неразличима, а более тонкая структура просматривается отчетливее. При $N = 2048$ эта структура видна еще отчетливее (больше точек появляется на границе ячеек), а более подробной структуры пока не наблюдается. При большем N визуальный анализ затруднителен. В целом структура кружева напоминает фрактал.

Точки в среднем распределены в квадрате гораздо равномернее, чем для псевдослучайных последовательностей, хотя внутри одной ячейки распределение точек существенно неравномерное.

Поэтому такую последовательность называют не псевдослучайной, а квазислучайной. Интуитивно ясно, что такая последовательность должна дать гораздо лучшие результаты для многомерного численного интегрирования. Однако для других приложений, где требуется существенная случайность (например, криптография) такая последовательность может оказаться неподходящей.

Сетка точек при $N \neq 2^k$ получается из магической сетки выбрасыванием части точек. Интуитивно следует ожидать, что результаты вычисления многомерного интеграла при немагическом числе точек окажутся хуже, чем при магическом.

Заметим, что последовательность Соболя строится неоднозначно. В ней возможен различный выбор так называемых направляющих чисел. От их выбора заметно зависит визуальное качество построенной сетки. Поэтому выбор направляющих чисел является важным аспектом. Мы брали направляющие числа непосредственно из [13].

4. Заключение

В настоящее время вряд ли существует какой-то алгоритм построения псевдослучайных чисел, который можно с полной уверенностью использовать для расчета любых случайных процессов. В литературе встречались

высказывания о том, что хорошие случайные свойства проявляются только на отрезках очень большой длины $N \sim 10^9$. Вероятно, при меньших N известные алгоритмы можно применять для тех или иных конкретных задач, но такие ситуации нужно дополнительно обосновывать. Например, для многомерных кубатур можно уверенно рекомендовать последовательности Соболя с магическими числами точек.

Поэтому проблема построения хороших генераторов остается нерешенной, а к результатам расчета случайных процессов (метод молекулярной динамики и др.) следует относиться с осторожностью.

Авторы искренне благодарны И.М. Соболю, И.А. Козлитину, Д.Д. Соколову за полезные обсуждения.

Работа поддержана грантами РФФИ 18-01-00175, МК-1780.2019.1.

СПИСОК ЛИТЕРАТУРЫ

1. *Цветков Е.А.* Эмпирическое исследование статистических свойств некоторых генераторов псевдослучайных чисел // Матем. Моделирование. 2011. Т. 23, № 5, С. 81–94.
2. *Matsumoto, M., Nishimura T.* Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudorandom Number Generator // ACM Transactions on Modeling and Computer Simulation. 1998. Vol. 8, No. 1. P. 3–30.
3. *Matsumoto M., Saito M.* A PRNG Specialized in Double Precision Floating Point Numbers Using an Affine Transition // Monte Carlo and Quasi-Monte Carlo Methods. 2008. DOI: 10.1007/978-3-642-04107-5_38. 2009.
4. *Park S.K., Miller K.W.* Random Number Generators: Good Ones Are Hard to Find // Communications of the ACM. 1998. Vol. 31, No. 10. P. 1192–1201.
5. *Mascagni, M., Srinivasan A.* Parameterizing Parallel Multiplicative Lagged-Fibonacci Generators // Parallel Computing. 2004. Vol. 30. P. 899–916.
6. *L'Ecuyer P.* Good Parameter Sets for Combined Multiple Recursive Random Number Generators // Operations Research. 1999. Vol. 47, No. 1. P. 159–164.
7. *Salmon, J.K., Moraes M.A., Dror R. O., Shaw D. E.* Parallel Random Numbers: As Easy as 1, 2, 3 // Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC11). New York, NY: ACM, 2011.

8. *Marsaglia G., Tsang W.W.* The ziggurat method for generating random variables // Journal of Statistical Software. 2000. Vol. 5, P. 1–7.
9. *Marsaglia, G., Zaman A.* A new class of random number generators // Annals of Applied Probability. 1991. Vol. 1, No. 3. P. 462–480.
10. *Wichmann B.A., Hill I.D.* An efficient and portable pseudo-random number generator // Applied Statistics. 1982. Vol. 31, No. 2. P. 188–190.
11. RAND Corporation. A million random digits with 100 000 normal deviates. The Free Press, 1955.
12. *Бахвалов Н.С., Жидков Н.П., Кобельков Г.М.* Численные методы. М.: Бином, 2008.
13. *Соболь И.М.* О распределении точек в кубе и сетках интегрирования.// Успехи матем. наук. 1966. Т. 21. № 5. С. 271-272.
14. *Соболь И.М.* Численные методы Монте-Карло. М.: Наука, 1973.

Оглавление