МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**Кафедра: Алгебры Геометрии и Дискретной Математики**

**ОТЧЕТ**

на тему:

**«Исследование и тестирование ГПСЧ»**

**Выполнил:** студент группы 081503-1

Улитин Александр Владимирович

Подпись

**Научный руководитель:**

доцент кафедры АГДМ, кандидат физико-математических наук

Веселов Сергей Иванович

Подпись

Нижний Новгород  
2017

**СОДЕРЖАНИЕ**

[Введение. 3](#_Toc501382388)

[ГПСЧ. 4](#_Toc501382389)

[Тестирование ГПСЧ. 5](#_Toc501382390)

[Энтропия. 5](#_Toc501382391)

[Среднее арифметическое. 5](#_Toc501382392)

[Значение Монте-Карло для π. 6](#_Toc501382393)

[Коэффициент последовательной корреляции. 6](#_Toc501382394)

[Вывод. 7](#_Toc501382395)

[Список литературы и ссылки. 8](#_Toc501382396)

[Приложений 1. 9](#_Toc501382397)

## Введение.

В криптографии одной из главной составляющей является генерация псевдослучайных чисел (ГПСЧ). Моя же научная работа направлена на исследование ГПСЧ и их тестирование. В данном отчете будет рассмотрен ГПСЧ созданный Виктором Данко (Danko, 2012)[1].

## ГПСЧ.

По статье Виктора Данко был написан генератор на языке C++[2]. Код представляет собой файлы “rnd.h”, “rnd.cpp”, в которых реализована функция для генерации чисел, и “main.cpp” для получения txt файла со сгенерированной последовательностью. Для работы генератора нужен входной параметр длиной 8-бит. Дальше генератор создает последовательность заданной длины.

## Тестирование ГПСЧ.

Для тестирования генератора были созданы последовательности размером 128 КБайт и 32 КБайта со всеми возможными входными параметрами, но результаты теста оказались идентичными, поэтому в данном отчете рассматривается последовательность размером 32 КБайта, кроме теста на значение Монте-Карло для π.

Для проверки алгоритма я использовал готовую библиотеку [3].

### Энтропия.

Для исследования энтропии в Приложении 1 приводиться график на рисунке 1. Он показывает вычисленную энтропию в зависимости от входного параметра. Если приглядеться, то можно увидеть 5 уровней, это точки со значениями энтропии 0, 0.54, 0.81, 0.95 и 1. Входные параметры 0 и 255 являются неуместными для этого генератора, т.к. для них последовательность вся состоит из 0 или 1 соответственно. Следующие слабое место это числа, которые имеют либо одну 1 в числе, либо один 0. При таких входных параметрах получаем энтропию 0,543564. Дальше энтропия имеет приемлемые значения.

### Среднее арифметическое.

Т.к. последовательность подавалась, как поток бит, то наилучший вариант, когда среднее арифметическое равно 0.5. График этого теста представлен на рисунке 2. Лучше всего анализировать этот тест по таблице результатов тестирования[[1]](#footnote-1)[4]. Если отсортировать по убыванию или возрастанию среднего арифметического, то можно увидеть, что увеличение единиц в числе[[2]](#footnote-2), которые подается на вход ГПСЧ, увеличивает среднее арифметическое на 0,125. Это свойство критическое для генератора.

### Значение Монте-Карло для π.

Для сравнения результатов тестирования с генератором основанном на радиоактивном распаде, нужна слишком большая сгенерированная последовательность. Пока же представлены результаты для 128 КБайт на рисунке 3. Из него видно, что большинство последовательностей выдают результат, приближенный к 4. Последовательность, которая больше всего удовлетворяет этому тесту, получается числом 27 и имеет точность до сотых.

### Коэффициент последовательной корреляции.

Этот тест лучше всего анализировать через таблицу результатов тестирования[[3]](#footnote-3)[5]. Через него видно, что всего лишь 16 последовательностей имеют значение 0. Все остальные последовательности имеют некие зависимости относительно своих бит.

## Вывод.

Данный генератор нуждается в доработке. Пути решения могут быть разные. Можно ограничить ввод параметров, которые дают плохие результаты в тестах, путем переопределения этих параметров на другие. Так же можно выбрать одно число, которое лучше всего проходит тесты, и генерировать числа только через него, тогда в качестве другого входного параметра можно брать число, с которого начинать считывать последовательность, или с каким шагом считываются числа из последовательности.

## Список литературы и ссылки.

1. Dańko W., AN ALGORITHM FOR GENERATING BINARY PSEUDO-RANDOM SEQUENCES 2012 - <https://github.com/Ulitin/Research_work>.
2. ГПСЧ на языке C++ - <https://github.com/Ulitin/Research_work>.
3. Библиотека с тестами и их описанием - <http://www.fourmilab.ch/random/>.
4. Таблица с результатами тестированияc последовательности размером 32КБайта full\_report\_32KB - <https://github.com/Ulitin/Research_work>.
5. Таблица с результатами тестирования последовательности размером 128КБайта full\_report\_128KB - <https://github.com/Ulitin/Research_work>.

## Приложений 1.

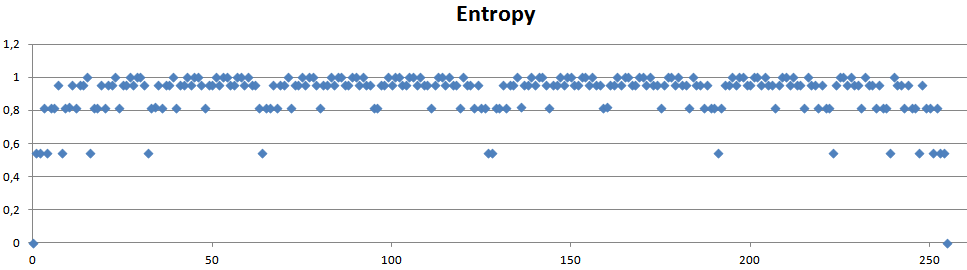


рис.1 График зависимости энтропии от входного параметра в ГПСЧ.

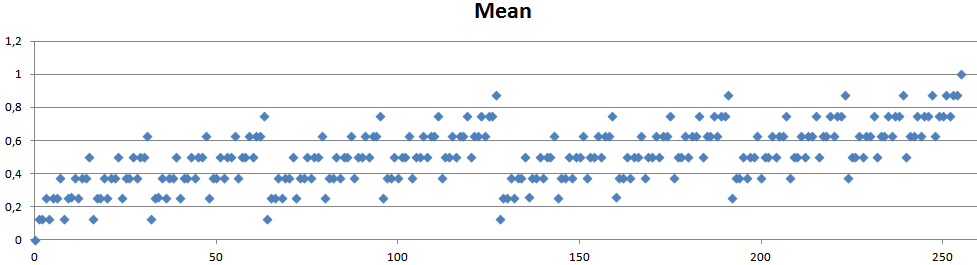


рис.2 График зависимости среднего арифметического от входного параметра в ГПСЧ.

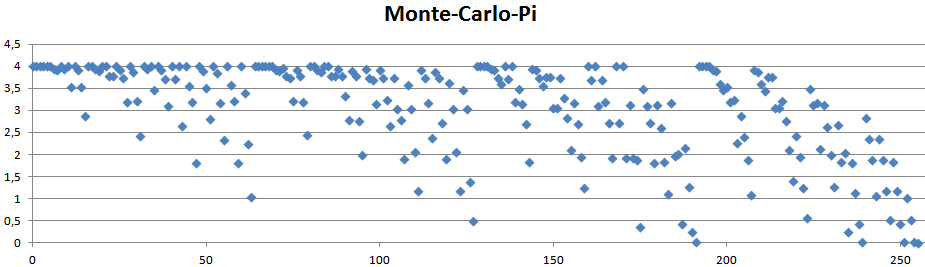


рис.3 График зависимости выходных данных теста на значение Монте-Карло для π от входного параметра в ГПСЧ.

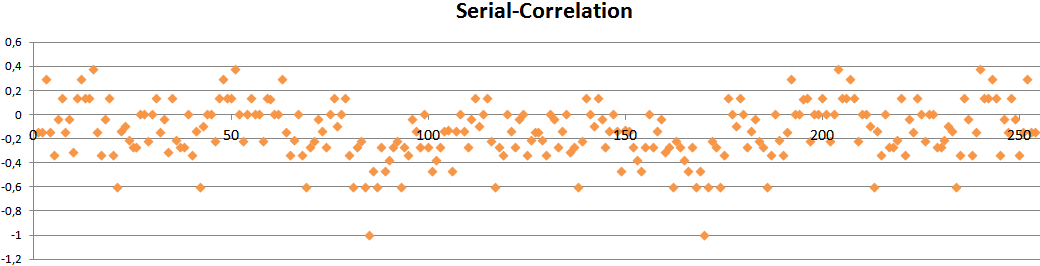


рис.4 График зависимости коэффициента последовательной корреляции от входного параметра в ГПСЧ.

1. Таблица результатов тестирования последовательностей размером 32 КБайт [↑](#footnote-ref-1)
2. Рассматриваем число в двоичной системе отсчета [↑](#footnote-ref-2)
3. Таблица результатов тестирования последовательностей размером 128 КБайт [↑](#footnote-ref-3)