

ПОЛУЧЕНИЕ ПСЕВДОСЛУЧАЙНЫХ ВЕЛИЧИН МЕТОДАМИ НЕЙМАНА

Волхонский А.Н.¹

¹ФГБОУ ВО «Самарский государственный технический университет», Самара, e-mail: avolhonskij34@gmail.com

Аннотация. В современной промышленности активно развивается тенденция замены ручного труда автоматическим, осуществляемым без участия человека или с его минимальным участие. Автоматизация производств включает в себя множество различных этапов и соответствующих. Данные этапы включают в себя множество различных операций, начиная от определения участка технологического процесса, автоматизацию которого предполагается произвести, заканчивая выбором наиболее эффективного в данных условиях промышленного оборудования и установкой его в соответствии с разработанной схемой автоматизации. Одним из основных этапов является построение модели процесса. Данный этап позволяет выявить возможные реакции системы на различные производимые над ней операции. При помощи полученной модели производят расчет настроек регуляторов, устанавливаемых на линиях подачи управляющего воздействия на объекты управления. Для наибольшей эффективности модель должна отражать как собственные свойства объекта, так и возможные возмущающие воздействия. Для проектирования возмущающих воздействий используют генераторы белого шума, случайной величины с определенным распределением вероятностей. Существует множество алгоритмов получения псевдослучайных величин, отвечающих данному условию. В данной статье будут рассмотрены алгоритмы получения псевдослучайных величин по методу Неймана, построены блок-схемы алгоритмов и представлена программная реализация метода на языке программирования java script.

Ключевые слова: автоматизация, белый шум, псевдослучайные числа, блок-схема, java script.

OBTAINING PSEUDORANDOM VALUES BY NEUMANN METHODS

Volkhinskij A.N.¹

¹Samara State Technical University, Samara, e-mail: avolhonskij34@gmail.com

Annotation. In modern industry, the trend of replacing manual labor with automatic labor, carried out without human participation or with minimal participation, is actively developing. Production automation involves many different stages and processes. These stages include many different operations, ranging from determining the section of the technological process that is supposed to be automated, to choosing the most efficient industrial equipment in these conditions and installing it in accordance with the developed automation scheme. One of the main steps is to build a process model. This stage allows you to identify possible reactions of the system to various operations performed on it. With the help of the obtained model, the settings of the regulators installed on the supply lines of the control action to the control objects are calculated. For maximum efficiency, the model should reflect both the object's own properties and possible disturbing effects. For the design of disturbing effects, white noise generators are used, a random variable with a certain probability distribution. There are many algorithms for obtaining pseudo-random variables that meet this condition. In this article, algorithms for obtaining pseudo-random variables using the Neumann method are considered, flowcharts of algorithms are constructed, and a software implementation of the method in the java script programming language is presented.

Keywords: automation, white noise, pseudorandom numbers, flowchart, java script.

При исследовании различных динамических систем часто используются имитационные модели. Согласно одному из определений, имитационная модель это определенный алгоритм, описывающий функционирование системы во времени, реализация которого осуществляется с помощью соответствующей машинной программы.

При невозможности описать весь спектр воздействующих на систему факторов в модель вводится элемент случайности. Таким образом, для работы моделирующей программы необходима генерация случайных чисел.

Компьютерная программа, работающая по определенному алгоритму, может генерировать только псевдослучайные числа. Результаты работы различных алгоритмов проверяются статистическими тестами. Для генерации случайных чисел в программу подставляются случайным образом начальные условия. Для моделирования могут быть необходимы случайные величины с самыми разными характеристиками, но обычно, сначала генерируется базовая последовательность случайных чисел. Совокупность независимых, равномерно распределенных на отрезке $[0,1]$ случайных величин R_i , ($i=0,1\dots$) называется последовательностью базовых случайных чисел. Гистограмма, показывающее равномерное распределение величин представлена на рисунке 1.

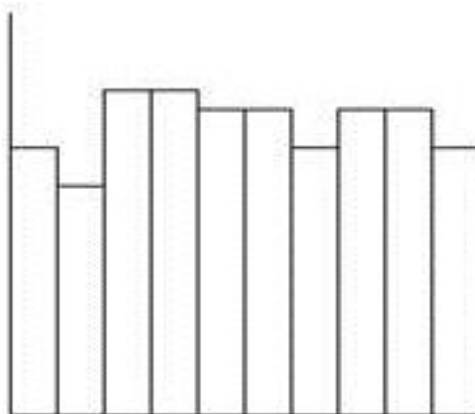


Рисунок 1 – Гистограмма равномерного распределения величин

Алгоритм середины квадратов Неймана

Первый алгоритм для получения псевдослучайных чисел был предложен Дж. Фон Нейманом в 1951 г. Он называется методом середины квадратов.

Пусть задано случайно выбранное 4-значное число $R_0 = 0.9876$.

Возведя его в квадрат, получим 8-значное число $R_0^2 = 0.97535376$.

Выбрав четыре средних цифры этого числа, получим следующее псевдослучайное число: $R_1 = 0.5353$.

Повторив операцию возведения в квадрат $R_1^2 = 0,28654609$ и процедуру выбора средних цифр числа R_1^2 , получим очередное псевдослучайное число: $R_2 = 0.6546$.

Таким образом, получен очень простой рекуррентный алгоритм генерирования псевдослучайных точек, который называется методом Неймана или методом серединных квадратов. Для запуска алгоритма достаточно задать некоторое произвольное начальное число

R_0 . Разным стартовым числом, вообще говоря, будут соответствовать разные последовательности чисел.

Блок-схема данного алгоритма представлена на рисунке 2.

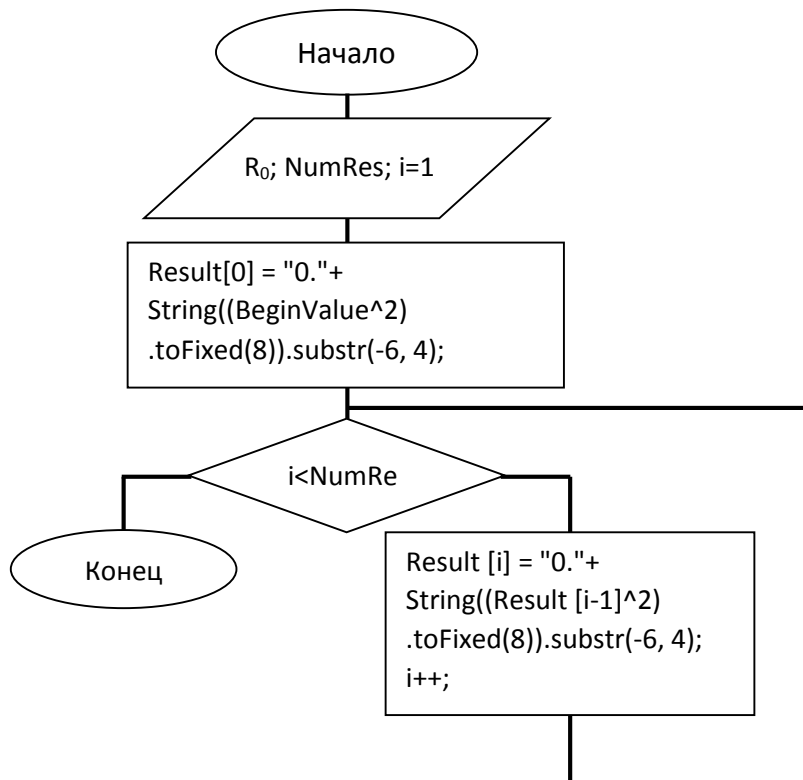


Рисунок 2 – Блок-схема реализации метода середины квадратов Неймана

Программная реализация данного метода на языке программирования java script представлена на рисунке 3.

```
<html>
<head>
  <script type="text/javascript">
    var BeginValue = 0.9876;
    var numResults= 100;
    var ResultTable=[];
    ResultTable[0] = "0."+String((BeginValue*BeginValue).toFixed(8)).substr(-6, 4);
    console.log(ResultTable[0]);
    for(var i=1;i<numResults;i++){
      ResultTable[i] = "0."+String((ResultTable[i-1]*ResultTable[i-1]).toFixed(8)).substr(-6, 4);
    }
    console.log(ResultTable);
  </script>
</head>
</html>
```

Рисунок 3 – Реализация метода на языке программирования java script

Этот алгоритм не оправдал себя: получалось больше чем нужно для равномерного распределения малых значений R_i . Кроме того, весьма часто последовательность случайных чисел оказывается слишком короткой, т.е. после некоторого R_i числа начинают повторяться. Также в последовательности может вообще отсутствовать случайность, как если в качестве начального числа выбрано $R_0=0.4500$.

Модифицированный метод Неймана

Значительно лучшие результаты дает определенная модификация алгоритма Неймана, состоящая в том, что вначале произвольно выбирается пара чисел R_0 и R_1 . Составляется их произведение $R_0 * R_1$, и его средние цифры используются в качестве числа R_2 . Процесс повторяется для R_1, R_2 с получением R_3 и т.д.

Блок-схема данного алгоритма представлена на рисунке 4.

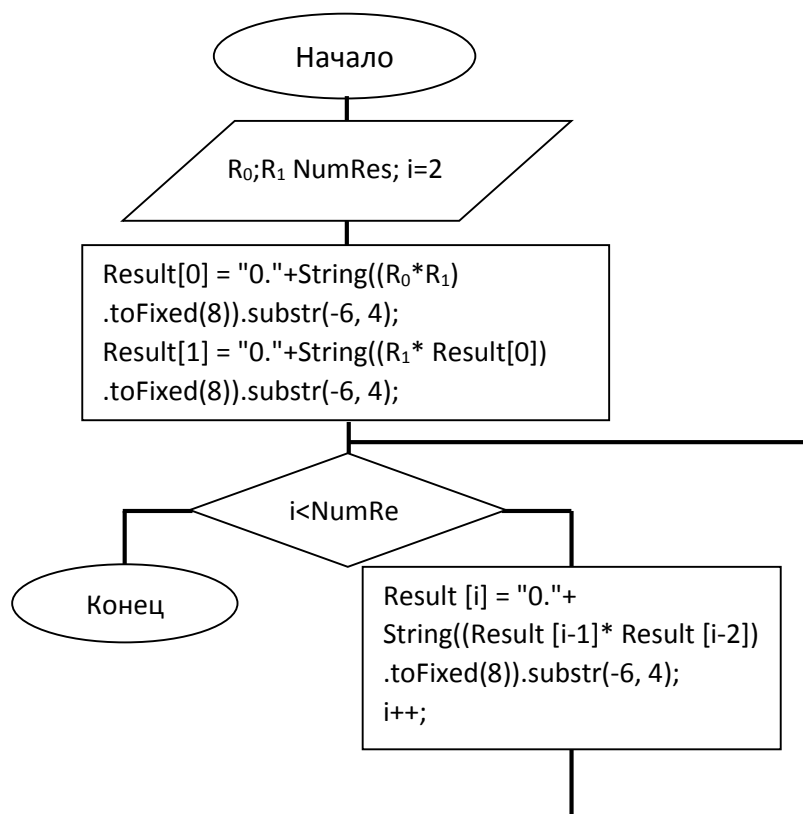


Рисунок 4 – Блок-схема реализации модифицированного метода Неймана

Программная реализация данного метода на языке программирования java script представлена на рисунке 5.

```
<html>
<head>
  <script type="text/javascript">
    var BeginValue = 0.5836;
    var BeginValue2 = 0.2176;
    var numResults= 100;
    var ResultTable=[];
    ResultTable[0] = "0."+String((BeginValue*BeginValue2).toFixed(8)).substr(-6, 4);
    ResultTable[1] = "0."+String((BeginValue2*ResultTable[0]).toFixed(8)).substr(-6, 4);
    console.log(ResultTable[0]);
    for(var i=2;i<numResults;i++){
      ResultTable[i] = "0."+String((ResultTable[i-1]*ResultTable[i-2]).toFixed(8)).substr(-6, 4);
    }
    console.log(ResultTable);
  </script>
</head>
</html>
```

Рисунок 5 – Реализация модифицированного метода Неймана на языке программирования java script

Такой рекуррентный метод дает меньшее отклонение образуемых псевдослучайных чисел от равномерного распределения, нежели метод Неймана.

Список литературы:

1. Метод Неймана// Студопедия [Электронный ресурс] URL: https://studopedia.ru/9_219175_lektsiya-.html
2. В.А. РУМЯНЦЕВА ПРАКТИКУМ ПО МОДЕЛИРОВАНИЮ СИСТЕМ// Страницка Валентины Румянцевой [Электронный ресурс] URL: <http://valaru.narod.ru/work/metodichka.pdf>
3. Метод Неймана // Научная библиотека избранных естественно-научных изданий [Электронный ресурс] URL: https://scask.ru/a_book_dm.php?id=7