Министерство науки и высшего образования Российской Федерации  
Рязанский государственный радиотехнический университет  
Кафедра вычислительной и прикладной математики

# ЛАБОРАТОРНАЯ РАБОТА №5 ГЕНЕРИРОВАНИЕ СЛУЧАЙНЫХ ВЕЛИЧИН С ЧАСТО ИСПОЛЬЗУЕМЫМИ ЗАКОНАМИ РАСПРЕДЕЛЕНИЯ

**Выполнил:**  
студент 4 курса группы 843  
Попов М.С.

**Проверил:** д-р техн. наук,

профессор каф. ВПМОвечкин Г.В.

Рязань, 2022 год

**Задание:**

СОСТАВИТЬ ПОДПРОГРАММЫ ГЕНЕРИРОВАНИЯ СЛУЧАЙНЫХ ВЕЛИЧИН, ПОДЧИНЕННЫХ РАСПРЕДЕЛЕНИЮ, УКАЗАННОМУ В ВАРИАНТЕ ЗАДАНИЯ (ТАБ. 5)  
1) ПО ПОЛУЧЕННОЙ С ПОМОЩЬЮ ПОДПРОГРАММЫ ВЫБОРКЕ ПОСТРОИТЬ И ПРОАНАЛИЗИРОВАТЬ ГИСТОГРАММУ ЧАСТОТ И СТАТИСТИЧЕСКУЮ ФУНКЦИЮ РАСПРЕДЕЛЕНИЯ  
2) ОЦЕНИТЬ МАТОЖИДАНИЕ И ДИСПЕРСИЮ СЛУЧАЙНОЙ ВЕЛИЧИНЫ.  
3) СООТВЕТСТВИЕ ЭМПИРИЧЕСКИХ ДАННЫХ ТЕОРЕТИЧЕСКОМУ РАСПРЕДЕЛЕНИЮ ПРОВЕРИТЬ С ПОМОЩЬЮ КРИТЕРИЯ ПИРСОНА ИЛИ КРИТЕРИЯ КОЛМОГОРОВА.  
ОБЪЕМ ВЫБОРКИ СЛУЧАЙНЫХ ВЕЛИЧИН НЕ МЕНЕЕ 1000.  
КОЛИЧЕСТВО ИНТЕРВАЛОВ РАЗБИЕНИЯ K = 15 ИЛИ K = 25.  
ТЕОРЕТИЧЕСКАЯ ЧАСТЬ ДЛЯ ДАННОЙ ЛАБОРАТОРНОЙ РАБОТЫ ПРЕДСТАВЛЕНА В УЧЕБНИКЕ [1] НА СТР. 76–83.

C:\Users\matyh\Desktop\КМ\Лабараторные работы КМ\assets\img\varLabFifth.png

/\* Переменные \*/

let arrayNormalRasprX = []; // Массив с нормальным законом распределнения X

let arrayNormalRasprY = []; // Массив с нормальным законом распределнения Y

let arrayNormal = [];

let randomAmountArray = [];

let randomAmountArrayNorml = [];

let statFunction = []; // Стат функция

let sumStatFunction = []; // Стат функция

let sumNormlRulArray = [];

let normlSecondArry = [];

let amountArray = []; // Массив с количеством чисел на каждом интервале

let verPopSluchVal = []; // Массив вероятностей попадания случайных чисел на каждом интервале

let waitAmountVal = []; // Массив ожидаемых числовых величин

let c = []; // Массив С

let mearDifr = 0; // Величина расхождения между теоретической и эксперементальной плотностями вероятности

let XiCv = 0; // "хи-квадрат"

let randomRightArray = []; //Массив правых границ интервалов

let randomLeftArray = []; //Массивы левых границ интервалов

let sizeInterval = 0;

let firstHistogrammData = {}; // Данные для гистограммы

// Параметры выборки

const DELTA = 67; // Дельта

const SAMPLE\_SIZE = 1000; // Размер выборки

const AMOUNT\_INTERVALS = 15; // Количество интервалов

const MATH\_WAITING = -1.5; // Мат. ожидание

const DISPERTION = 1.7; // Дисперсия

// Переменные для критерия Колмогорова

let iOnN = []; // i/n

let iSubOneOnN = []; // (i-1)/n

let dPlus = []; // D+

let lastElementDPlus = 0; // Последний элемент D+

let dSub = []; // D-

let lasrElementDSub = 0; // Последний элемент D-

let maxD = 0; // Максимальное D

let dTabl = 0 // D табл.

let lyambda = 0; // Лямбда

window.addEventListener("load", Diagram);

Министерство науки и высшего образования Российской Федерации  
Рязанский государственный радиотехнический университет  
Кафедра вычислительной и прикладной математики

# ЛАБОРАТОРНАЯ РАБОТА №1 ИЗУЧЕНИЕ БАЗОВЫХ ГЕНЕРАТОРОВ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

**Выполнил:**  
студент 4 курса группы 843  
Володин В.А.

**Проверил:** д-р техн. наук,

профессор каф. ВПМОвечкин Г.В.

Рязань, 2022 г.

Министерство науки и высшего образования Российской Федерации  
Рязанский государственный радиотехнический университет  
Кафедра вычислительной и прикладной математики

# ЛАБОРАТОРНАЯ РАБОТА №4

Генерирование случайных

величин с нормальным законом распределения

**Выполнил:**  
студент 4 курса группы 843  
Гиголо Мпена Ж.В.

**Проверил:** д-р техн. наук,

профессор каф. ВПМОвечкин Г.В.

Рязань, 2022 год

Министерство науки и высшего образования Российской Федерации  
Рязанский государственный радиотехнический университет  
Кафедра вычислительной и прикладной математики

# ЛАБОРАТОРНАЯ РАБОТА №6 Моделирование методом Монте-Карло

**Выполнил:**  
студент 4 курса группы 843  
Попов М.С.

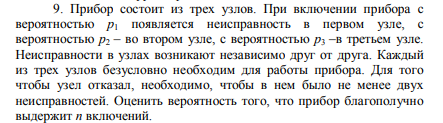
**Проверил:** д-р техн. наук,

профессор каф. ВПМОвечкин Г.В.

Рязань, 2022 год

**Задание**

Составить программу решения задачи, определенной в соответствии с вариантом задания, с помощью машинного моделирования (метод Монте-Карло). Построить доверительный интервал для полученных оценок, накрывающий точное значение оцениваемых вероятностей с надежностью β=0,95. Правильность результатов проверить аналитическим решением задачи. Теоретическая часть для данной лабораторной работы представлена в учебнике [1] на стр. 94–108 и 110–117

****

/\* Переменные \*/

let arrayNormalRasprX = []; // Массив с нормальным законом распределнения X

let arrayNormalRasprY = []; // Массив с нормальным законом распределнения Y

let arrayNormal = [];

let randomAmountArray = [];

let randomAmountArrayNorml = [];

let statFunction = []; // Стат функция

let sumStatFunction = []; // Стат функция

let sumNormlRulArray = [];

let normlSecondArry = [];

let amountArray = []; // Массив с количеством чисел на каждом интервале

let verPopSluchVal = []; // Массив вероятностей попадания случайных чисел на каждом интервале

let waitAmountVal = []; // Массив ожидаемых числовых величин

let c = []; // Массив С

let mearDifr = 0; // Величина расхождения между теоретической и эксперементальной плотностями вероятности

let XiCv = 0; // "хи-квадрат"

let randomRightArray = []; //Массив правых границ интервалов

let randomLeftArray = []; //Массивы левых границ интервалов

let sizeInterval = 0;

let firstHistogrammData = {}; // Данные для гистограммы

// Параметры выборки

const DELTA = 67; // Дельта

const SAMPLE\_SIZE = 1000; // Размер выборки

const AMOUNT\_INTERVALS = 15; // Количество интервалов

const MATH\_WAITING = -1.5; // Мат. ожидание

const DISPERTION = 1.7; // Дисперсия

// Переменные для критерия Колмогорова

let iOnN = []; // i/n

let iSubOneOnN = []; // (i-1)/n

let dPlus = []; // D+

let lastElementDPlus = 0; // Последний элемент D+

let dSub = []; // D-

let lasrElementDSub = 0; // Последний элемент D-

let maxD = 0; // Максимальное D

let dTabl = 0 // D табл.

let lyambda = 0; // Лямбда

// Генератор для бета-распределения

function getRandomValue(arrayFirst, arraySecond, mathWait, middleSqwOtkl, n) {

let p = 2;

let m = 5;

for (let i = 0; i < n; i++) {

s = 0.0;

for (let j = 0; j < m; j++) {

r = Math.random();

s += Math.log(r) / (p + j);

}

arrayFirst[i] = Math.exp(s);

}

}

// Генератор для распределения Вейбула

function getRandomValueSecond(arrayFirst, mathWait, middleSqwOtkl, n) {

let c = 2;

let b = 5;

for (let i = 0; i < n; i++) {

r = Math.random();

// Знаменатель такой, поскольку нам

// необходимо смаштабировать выборку, для

// получения корректных результатов

arrayFirst[i] = (b \* Math.pow(-Math.log(r), 1 / c)) / (b \* 3);

}

}