



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н. Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

---

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

## Лабораторная работа № 1

**Дисциплина:** Моделирование

**Тема:** Изучение функций распределения и функций плотности  
распределения случайных чисел

**Вариант:** 9

**Студент:** Платонова О. С.

**Группа:** ИУ7-75Б

**Оценка (баллы)** \_\_\_\_\_

**Преподаватель:** Рудаков И. В.

Москва, 2021 г.

**Цель работы:** построение графиков функции распределения и функции плотности равномерного распределения и распределения Пуассона.

### Равномерное распределение $\mathcal{U}(a, b)$

Функция распределения:

$$F_X(x) \equiv P(X \leq x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x < b \\ 1, & x \geq b \end{cases}$$

Функция плотности:

$$f_X(x) = \begin{cases} \frac{1}{b-a}, & x \in [a, b] \\ 0, & x \notin [a, b] \end{cases}$$

### Распределение Пуассона $P(\lambda)$

Функция распределения:

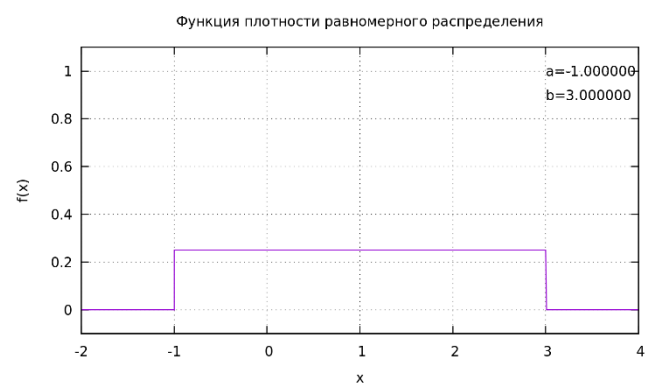
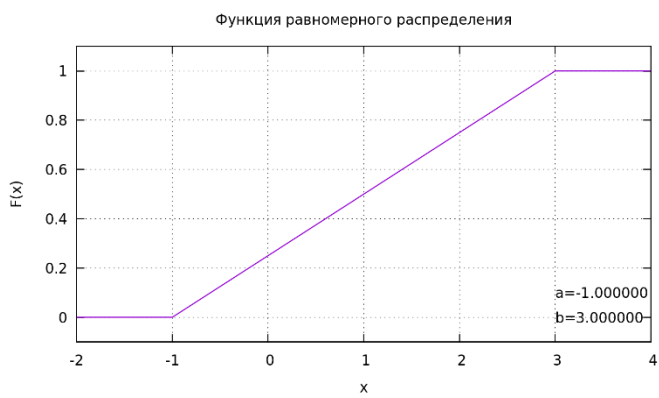
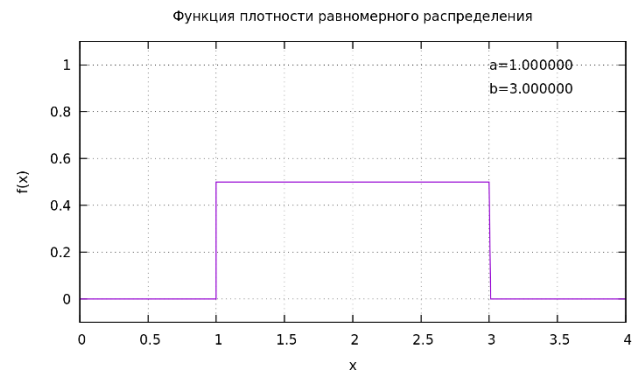
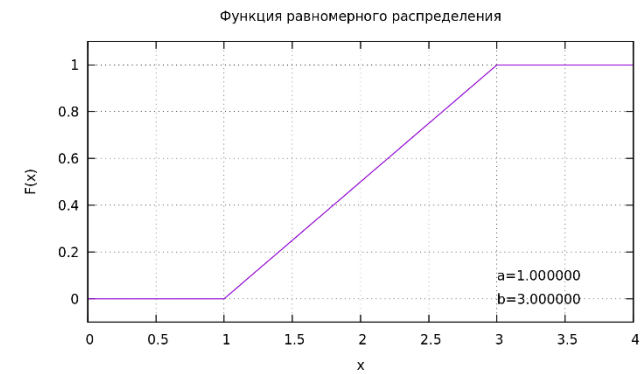
$$F_X(x) \equiv P(X = x) = \sum_{k=0}^x \frac{\lambda^k e^{-\lambda}}{k!}$$

Функция плотности:

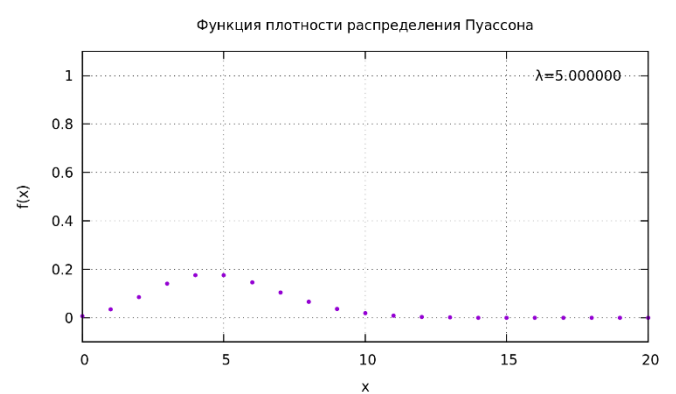
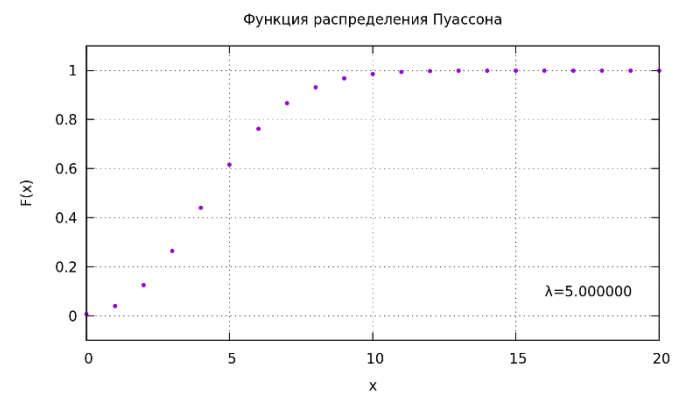
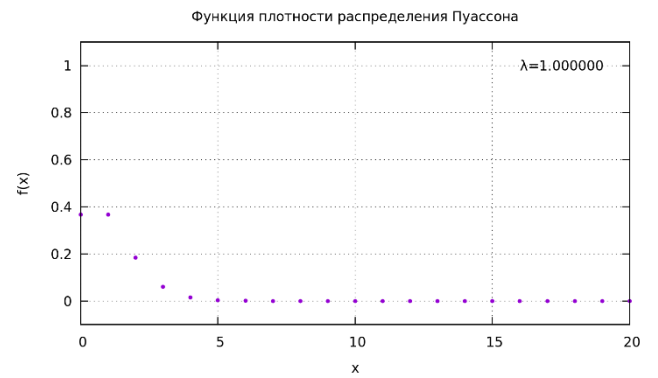
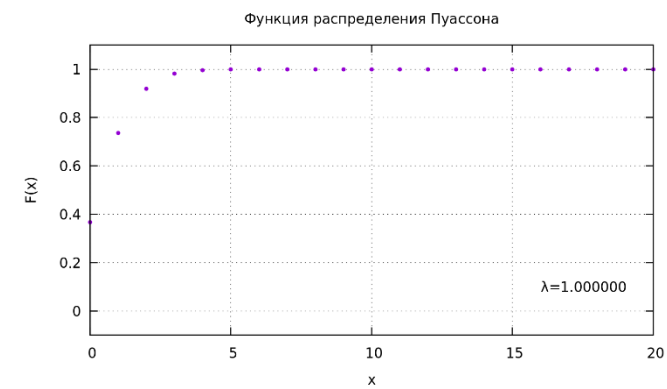
$$f_X(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

# Результаты

## Равномерное распределение



## Распределение Пуассона



## Листинг

### Файл uniform.cpp

```
1  #include "uniform.h"
2  #include "gnuplot.h"
3
4  void FUniform(Gnuplot *plot, const float a, const float b)
5  {
6      for (float x = a - 1; x < a; x += 0.01) {
7          (*plot)(x, 0);
8      }
9
10     for (float x = a; x < b; x += 0.01) {
11         float y = (x - a) / (b - a);
12         (*plot)(x, y);
13     }
14
15     for (float x = b; x < b + 1; x += 0.01) {
16         (*plot)(x, 1);
17     }
18 }
19
20 void fUniform(Gnuplot *plot, const float a, const float b)
21 {
22     for (float i = a - 1; i < a; i += 0.01) {
23         (*plot)(i, 0);
24     }
25
26     for (float i = a; i <= b; i += 0.01) {
27         float x = i;
28         float y = 1.0 / (b - a);
29         (*plot)(x, y);
30     }
31
32     for (float i = b + 0.01; i < b + 1; i += 0.01) {
33         (*plot)(i, 0);
34     }
35 }
36
37 void plotFUniform(const float a, const float b)
38 {
39     string aStr = "a=" + std::to_string(a);
40     string bStr = "b=" + std::to_string(b);
41
42     Gnuplot plot;
43     plot.initF("Функция равномерного распределения", aStr, bStr);
44
45     FUniform(&plot, a, b);
46 }
47
48 void plotfUniform(const float a, const float b)
49 {
50     string aStr = "a=" + std::to_string(a);
51     string bStr = "b=" + std::to_string(b);
52
53     Gnuplot plot;
54     plot.initf("Функция плотности равномерного распределения", aStr, bStr);
55
56     fUniform(&plot, a, b);
57 }
58
```

## Файл poisson.cpp

```
1  #include <cmath>
2
3  #include "poisson.h"
4  #include "gnuplot.h"
5
6  void FPoisson(Gnuplot *plot, const float lambda)
7  {
8      for (int x = 0; x <= 20; x++) {
9          float y = 0;
10         for (int i = 0; i <= x; i++) {
11             y += exp(-lambda) * pow(lambda, i) / tgamma(i + 1);
12         }
13         (*plot)(x, y);
14     }
15 }
16
17 void fPoisson(Gnuplot *plot, const float lambda)
18 {
19     for (int x = 0; x <= 20; x++) {
20         float y = exp(-lambda) * pow(lambda, x) / tgamma(x + 1);
21         (*plot)(x, y);
22     }
23 }
24
25 void plotFPoisson(const float lambda)
26 {
27     string lmbdStr = "{/Symbol l}=" + std::to_string(lambda);
28
29     Gnuplot plot;
30     plot.initF("Функция распределения Пуассона", lmbdStr);
31
32     FPoisson(&plot, lambda);
33 }
34
35 void plotfPoisson(const float lambda)
36 {
37     string lmbdStr = "{/Symbol l}=" + std::to_string(lambda);
38
39     Gnuplot plot;
40     plot.initf("Функция плотности распределения Пуассона", lmbdStr);
41
42     fPoisson(&plot, lambda);
43 }
44
```