

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Сибирский государственный университет
телекоммуникаций и информатики» (СибГУТИ)

Кафедра ВС

Лабораторная работа 1
по дисциплине «Моделирование»

Выполнил: ст. гр.
ЗМП-41 Лёвкин И. А.

Проверил: Родионов А.С.

Новосибирск 2025

Задание

Построить датчик независимых, одинаково распределённых случайных величин по заданной плотности.

Ход работы:

Проверка условия нормировки

Плотность распределения:

$$f(x) = \begin{cases} 0.25e^{-x+0.25}, & \text{при } x \in (0.25, 1) \\ k(x - 1.5), & \text{при } x \in (1, 1.5) \end{cases}$$

Условие нормировки:

$$\int_{-\infty}^{\infty} f(x) dx = 1$$

Вычисляем интегралы:

1. Для $x \in (-0.25, 1)$

$$\int_{-0.25}^1 0.25e^{-(x+0.25)} dx = 0.25(1 - e^{-1.25})$$

2. Для $x \in (1, 1.5)$

$$\int_1^{1.5} k(x - 1.5) dx = -\frac{k}{8}$$

Решение:

$$0.25(1 - e^{-1.25}) - \frac{k}{8} = 1 \Rightarrow k = 8(0.25(1 - e^{-1.25}) - 1)$$

Функция распределения F(x)

1. Для $x \in (-0.25, 1)$:

$$F(x) = 0.25(1 - e^{-(x+0.25)})$$

2. Для $x \in (1, 1.5)$:

$$F(x) = 0.25(1 - e^{-1.25}) + k \left(\frac{(x - 1.5)^2}{2} - \frac{0.25}{2} \right)$$

Метод обратного преобразования

Генерируем $u \sim U(0, 1)$ и находим $x = F^{-1}(u)$

1. Если $u \leq F(1) = 0.25(1 - e^{-1.25})$:

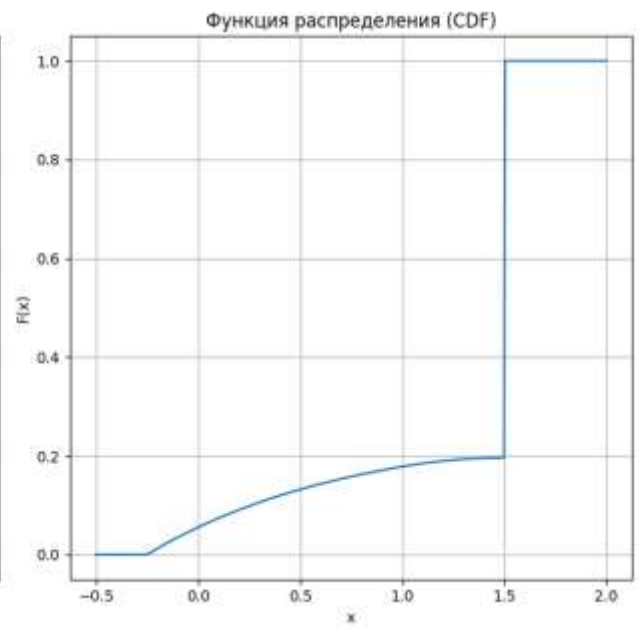
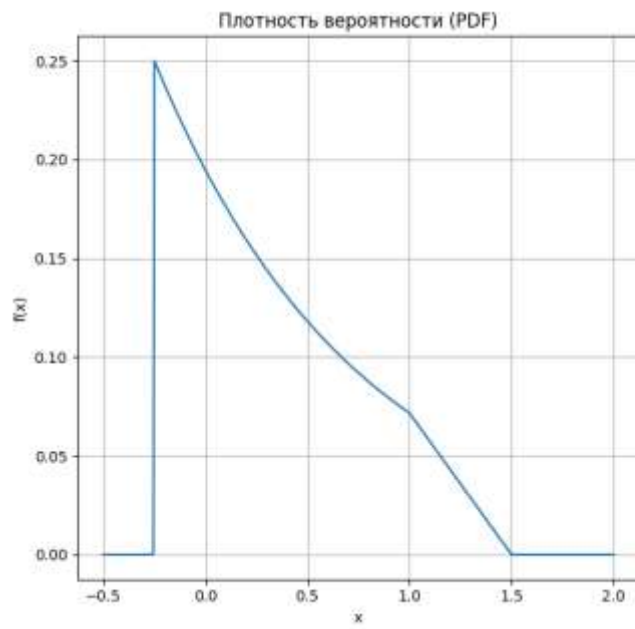
$$x = -\ln(1 - 4u) - 0.25$$

2. Если $u > F(1)$:

$$x = 1.5 - \sqrt{\frac{2}{k} \left(u - 0.25(1 - e^{-1.25}) + \frac{k}{8} \right)}$$

Результаты:

Задание 1



Листинг

```
import numpy as np
import matplotlib.pyplot as plt

# Константа для экспоненциальной части
def exp_part(x):
    return 0.25 * np.exp(-(x + 0.25))

# Находим значение f(1) из экспоненциальной части
f1 = exp_part(1)

# Находим коэффициент k так, чтобы f(x) была непрерывна в точке x = 1
# f1 = k * (1 - 1.5) => k = -f1 / 0.5
k = -2 * f1

def pdf(x):
    if -0.25 < x < 1:
        return exp_part(x)
    elif 1 < x < 1.5:
        return k * (x - 1.5)
    else:
        return 0.0

def cdf(x):
    if x <= -0.25:
```

```

        return 0.0
    elif x >= 1.5:
        return 1.0
    else:
        xs = np.linspace(-0.25, x, 1000)
        dx = xs[1] - xs[0]
        return np.sum([pdf(xi) for xi in xs]) * dx

# Векторизация
pdf_vec = np.vectorize(pdf)
cdf_vec = np.vectorize(cdf)

# Графики
x_vals = np.linspace(-0.5, 2, 500)
pdf_vals = pdf_vec(x_vals)
cdf_vals = cdf_vec(x_vals)

plt.figure(figsize=(12, 6))

# PDF
plt.subplot(1, 2, 1)
plt.plot(x_vals, pdf_vals)
plt.title("Плотность вероятности (PDF)")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.grid(True)

plt.subplot(1, 2, 2)

```

```
plt.plot(x_vals, cdf_vals)
plt.title("Функция распределения (CDF)")
plt.xlabel("x")
plt.ylabel("F(x)")
plt.grid(True)

plt.tight_layout()
plt.show()

print(k)
```