



Федеральное государственное бюджетное образовательное учреждение
высшего образования
"МИРЭА – Российский технологический университет"
РТУ МИРЭА

Институт кибербезопасности и цифровых технологий

ОТЧЕТ
О ВЫПОЛНЕНИИ ПРАКТИЧЕСКОЙ РАБОТЫ №3
«Основные понятия статистического эксперимента»
по дисциплине
«Моделирование систем»

Выполнил: студент 3 курса
группы БСБО-11-21
Смирнов И.А.

Проверил: Хорсик И.А.

Задание №1.

Создать функцию, описывающую распределение плотности вероятности для нормального закона распределения случайной величины. Построить графики плотности вероятности нормального закона с заданными значениями математического ожидания M и среднеквадратического отклонения σ : $M=10$ и $\sigma=2$, $M=10$ и $\sigma=1$, $M=10$ и $\sigma=\frac{1}{2}$, $M=12$ и $\sigma=1$.

Решение:

- param x: Значение случайной величины (может быть массивом)
- param mu: Среднее значение (математическое ожидание)
- param sigma: Стандартное отклонение
- return: Значение функции плотности вероятности для заданного x, mu и sigma

```
# Task 1
def norm_distr(x, mu, sigma):
    coeff = 1 / (sigma * np.sqrt(2 * np.pi))
    exp = np.exp(-(np.power(x - mu, 2) / (2 * np.power(sigma, 2))))
    prob_dens = coeff * exp
    return prob_dens

params = [(10, 2), (10, 1), (10, 0.5), (12, 1)]

x = np.linspace(0, 20, 1000)

plt.figure(figsize=(12, 8))
for mu, sigma in params:
    y = norm_distr(x, mu, sigma)
    plt.plot(x, y, label=f'μ={mu}, σ={sigma}')

plt.title('Плотность вероятности нормального распределения')
plt.xlabel('Значение случайной величины')
plt.ylabel('Плотность вероятности')
plt.legend()
plt.grid(True)
plt.show()
```

Листинг 1. Код программы задания 1.

Моделирование нормального распределения методом обратной функции

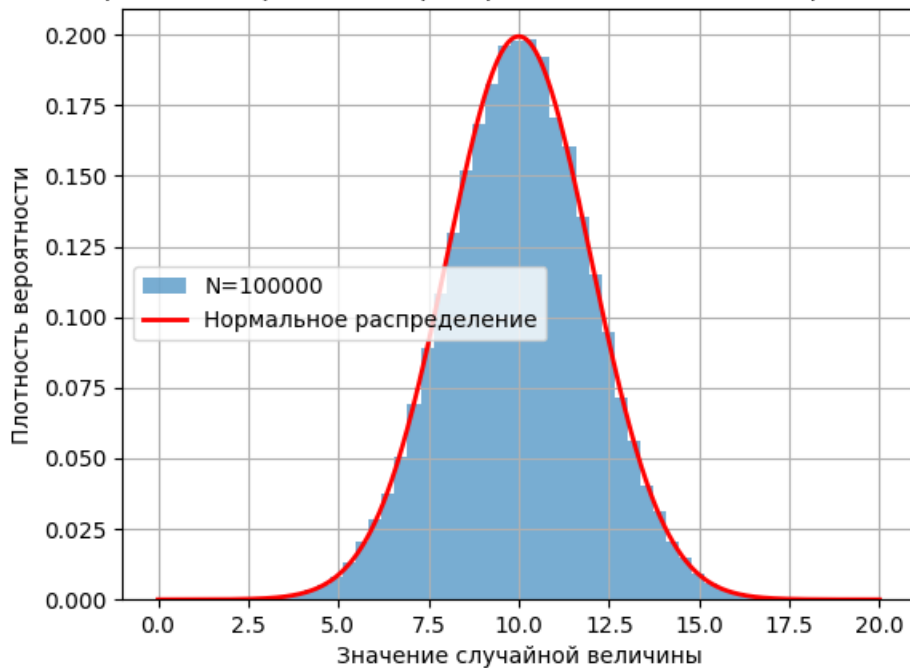


Рис. 2. Результат задания 1.

Задание №2.

Описать функцию распределения для нормального закона с параметрами $M = 10$ и $\sigma = 2$ и построить ее график на интервале от 0 до 20.

Решение:

```
# Task 2

cumul_distr = norm.cdf(x, 10, 2)

plt.figure(figsize=(8, 6))
plt.plot(x, cumul_distr, color='b')
plt.title('Функция распределения нормального закона\n$\mu=10$, $\sigma=2$')
plt.xlabel('Значение случайной величины')
plt.ylabel('Значение функции распределения')
plt.grid(True)
plt.show()
```

Листинг 3. Код программы задания 2.

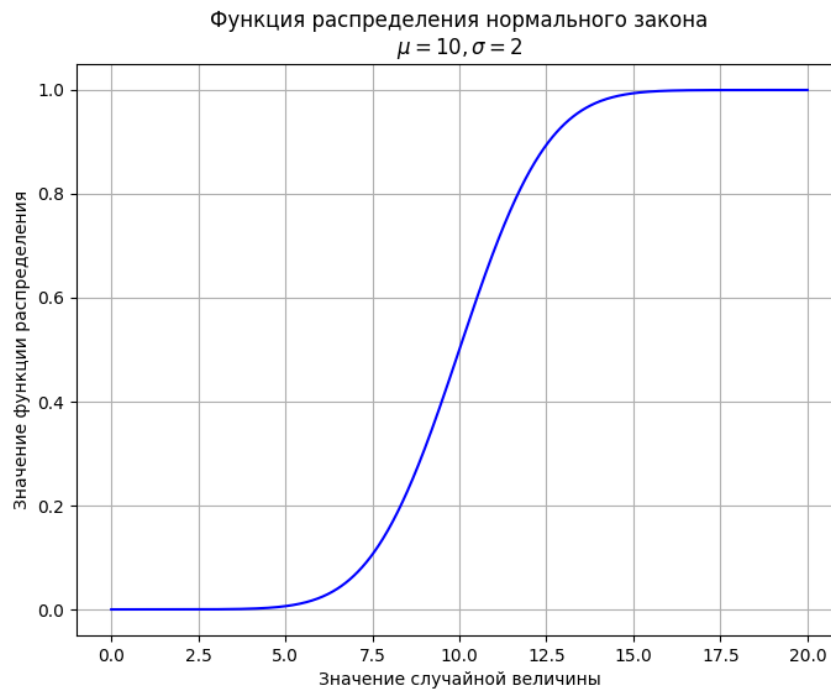


Рис. 4. Результат задания 2.

Задание №3.

Для полученной функции распределения (см. задание 2) провести моделирование нормального закона распределения с помощью метода обратной функции. Для моделирования равномерно распределенной случайной величины использовать встроенный генератор случайных чисел MathCAD (функция *rnd*). Для представления обратной функции использовать кусочно-линейную аппроксимацию для 100 равномерных участков на интервале от 0 до 20. Число экспериментов для моделирования принять равным $N = 10^3, 10^4, 10^5, 10^6$.

Решение:

```
# Task 3

N_values = [10 ** 3, 10 ** 4, 10 ** 5, 10 ** 6]

mu = 10
sigma = 2

for N in N_values:
    random_numbers = [random.uniform(0, 1) for _ in range(N)]
    normal_random_numbers = [norm.ppf(u, loc=mu, scale=sigma) for u in
                             random_numbers]

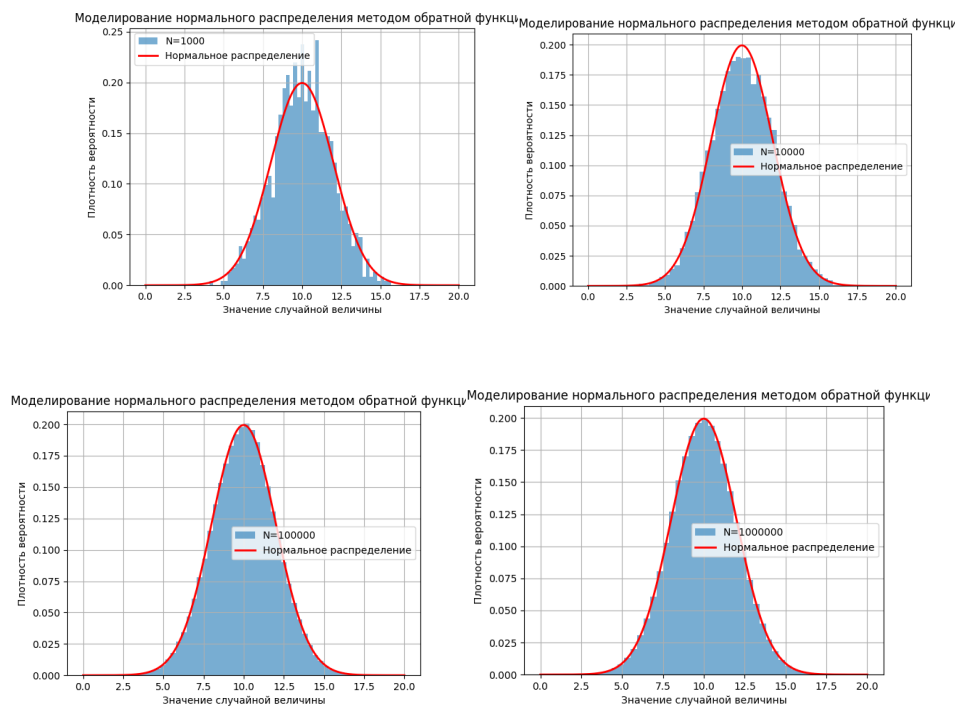
    plt.hist(normal_random_numbers, bins=50, density=True, alpha=0.6,
             label=f'N={N}')

    x = np.linspace(0, 20, 1000)
    norm_distr = norm.pdf(x, loc=mu, scale=sigma)
    plt.plot(x, norm_distr, color='red', linewidth=2, label='Нормальное
распределение')

    plt.title('Моделирование нормального распределения методом обратной функции')
    plt.xlabel('Значение случайной величины')
    plt.ylabel('Плотность вероятности')
    plt.legend()
    plt.grid(True)
    plt.show()
```

Листинг 5. Моделирование нормального закона.

Рис. 6. Результат задания 3.



Задание №4.

Построить гистограммы относительных частот для полученной последовательности случайных чисел на 100 интервалах и для числа экспериментов $N = 10^2, 10^3, 10^4, 10^5$.

Решение:

```
# Task 4

num_bins = 100

N_values = [10 ** 2, 10 ** 3, 10 ** 4, 10 ** 5]

for N in N_values:
    random_numbers = [random.uniform(0, 1) for _ in range(N)]
    normal_random_numbers = [norm.ppf(u, loc=mu, scale=sigma) for u in
random_numbers]

    plt.hist(normal_random_numbers, bins=num_bins, density=True, alpha=0.6,
label=f'N={N} ')

    x = np.linspace(0, 20, 1000)
    norm_distr = norm.pdf(x, loc=mu, scale=sigma)
    plt.plot(x, norm_distr, color='red', linewidth=2, label='Нормальное
распределение')

plt.title('Гистограммы относительных частот для разных чисел экспериментов')
plt.xlabel('Значение случайной величины')
plt.ylabel('Относительная частота')
plt.legend()
plt.grid(True)
plt.show()
```

Листинг 7. Код решения задачи определения периода.

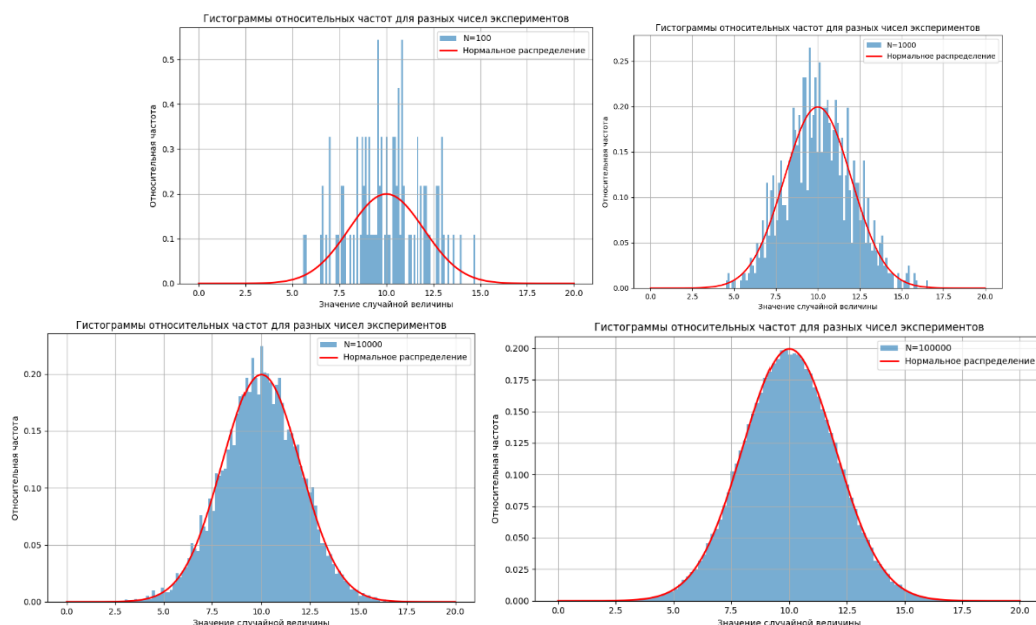


Рис. 8. Гистограммы относительных частот.

Задание №5.

Построить гистограммы относительных частот для полученной последовательности случайных чисел на 100 интервалах и для числа экспериментов $N = 10^2, 10^3, 10^4, 10^5$.

Решение:

```
# Task 5

x_theor = np.linspace(0, 20, 1000)
theor_distr = norm.pdf(x_theor, loc=mu, scale=sigma)

mean_squared_errors = []
for N in N_values:
    random_numbers = [random.random() for _ in range(N)]
    normal_random_numbers = [norm.ppf(u, loc=mu, scale=sigma) for u in
random_numbers]
    hist, bins = np.histogram(normal_random_numbers, bins=1000, density=True)
    exp_distr = hist * np.diff(bins)
    mean_squared_error = np.mean((theor_distr[:1000] - exp_distr) ** 2)
    mean_squared_errors.append(mean_squared_error)

plt.figure(figsize=(8, 6))
plt.semilogx(N_values, mean_squared_errors, marker='o', linestyle='-', color='b')
plt.title('Зависимость средней квадратичной погрешности от числа экспериментов')
plt.xlabel('Число экспериментов (логарифмическая шкала)')
plt.ylabel('Средняя квадратичная погрешность')
plt.grid(True)
plt.show()
```

Листинг 9. Код программы задания 5.

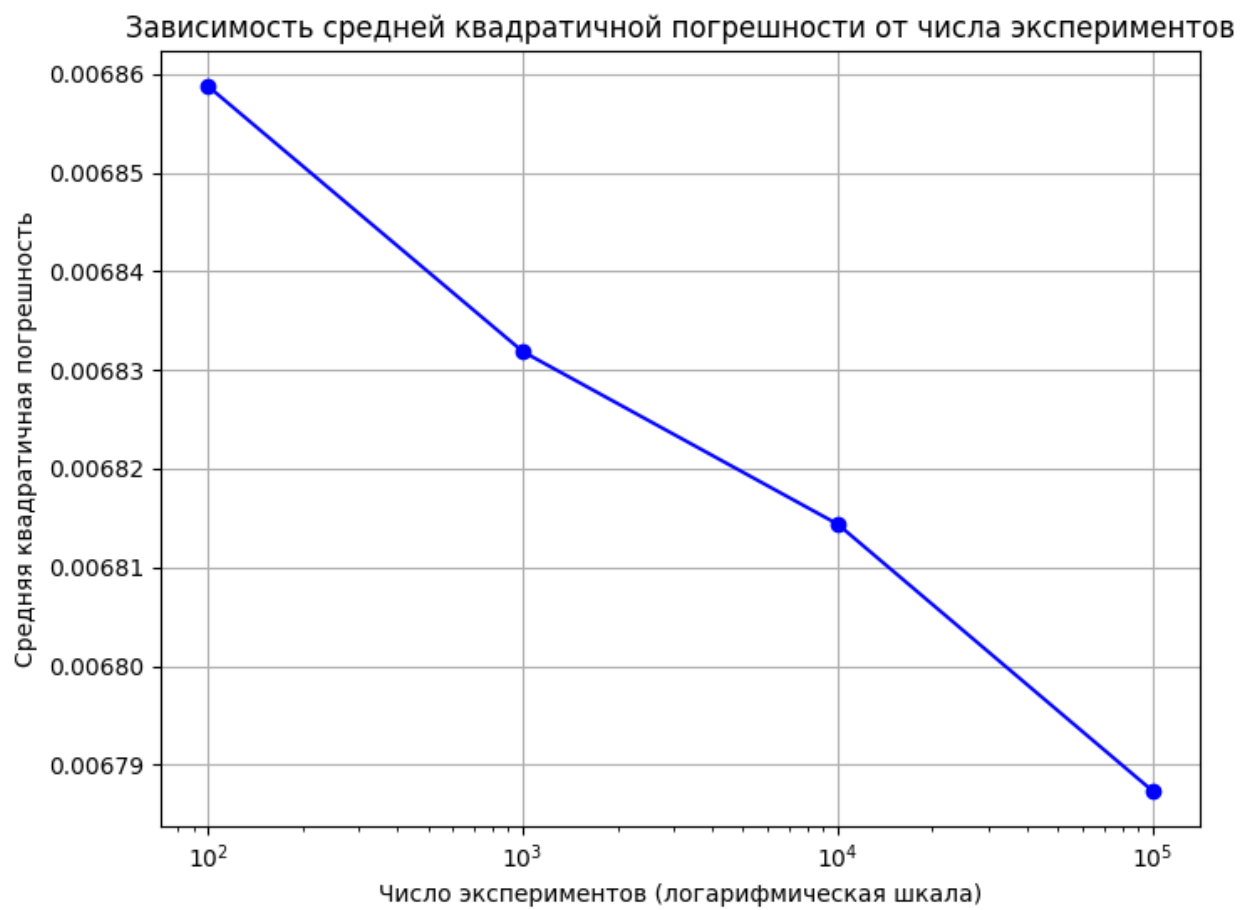


Рис. 10. Результат задания