МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский Нижегородский государственный
университет им. Н.И. Лобачевского»

Радиофизический факультет

ИТОГОВАЯ РАБОТА ПО ДИСЦИПЛИНЕ "МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ"

Отчет по (учебной) практике
Студента группа 427(0424С1ИБг1)
2 курса специалитета
Скороходов С.А.

Основная образовательная программа
подготовки по направлению
10.05.02 «Информационная
безопасность
телекоммуникационных систем»
(направленность «Системы
подвижной цифровой
защищенной связи»)

Нижний Новгород 2025

Содержание

I. ВВЕДЕНИЕ

Цель

Методом Монте-Карло вычислить объем V тела, ограниченного поверхностью

$$x^2 + y^2 - 1 \leqslant z^2 \leqslant \frac{3}{5}(x^2 + y^2 + 1) \tag{1}$$

Задачи

1. Написать функцию для нахождения объема функции в пространстве $\mathbb{R}^3$;
2. Найти объем тела, ограниченного функцией (1);
3. Оценить время и точность вычислений.

II. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1. Нахождение теоретического объема тела

Перед написанием метода Монте-Карло необходимо сначала найти теоретическое значение объема тела. Представляет собой зажатое между двумя однополостными гиперболоидами фигуру. Произведем замену переменных для цилиндрических координат:

$$\begin{cases} x = r\cos\varphi \\ y = r\sin\varphi \\ z = z(x,y) \end{cases} \Rightarrow \boxed{r^2 - 1 \leqslant z^2 \leqslant \frac{3}{5}(r^2+1)} \tag{2}$$

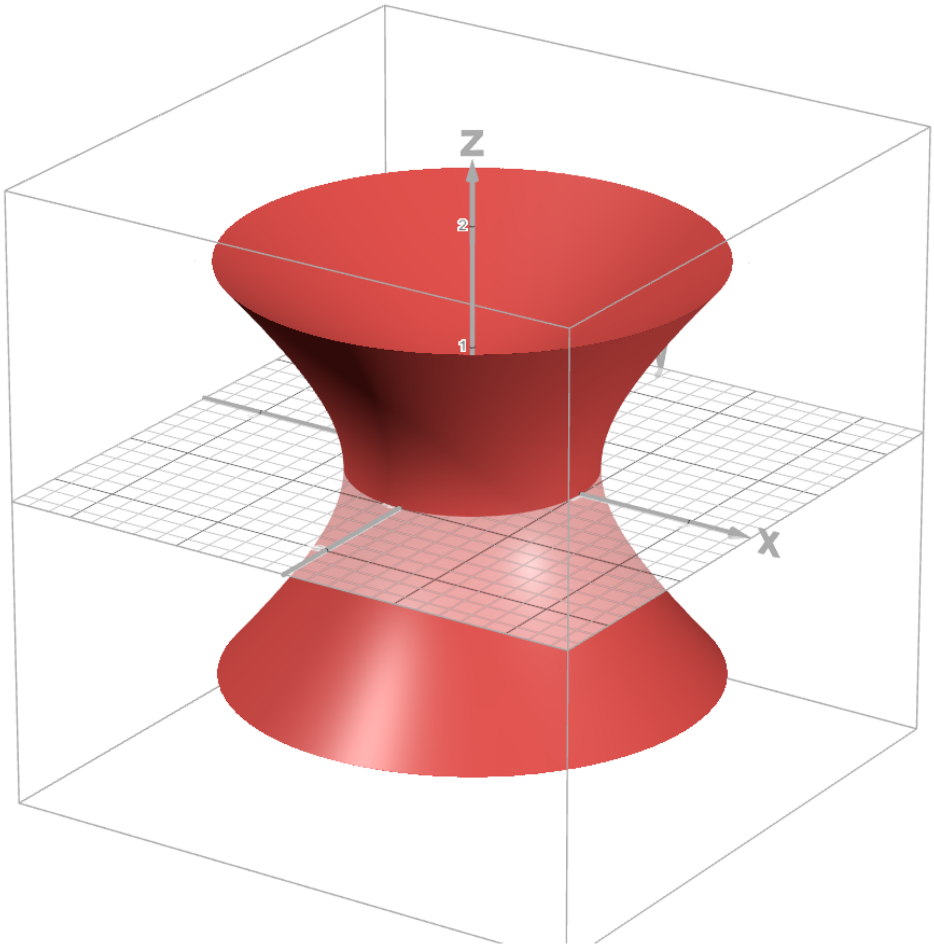

Рис. 1. Визуализация функции (1) в трехмерном пространстве (выполнена в Desmos)

Необходимо, что выполнялось условие

$$r^2 - 1 \leqslant \frac{3}{5}(r^2+1)$$

$$\frac{2}{5}r^2 \leqslant \frac{8}{5}$$

$$r^2 \leqslant 4 \rightarrow \boxed{0 \leqslant r \leqslant 2} \tag{3}$$

Рассмотрим два случая, когда нижняя граница $z_{\min}(r)$ меняет свое выражение:

1. При $0 \leqslant r \leqslant 1$: $r^2 - 1 \leqslant 0$, поэтому условие $z^2 \geqslant r^2 - 1$ выполняется всегда. Тогда:

$$z_{min}(r) = 0,\ z_{max}(r) = \sqrt{\frac{3}{5}(r^2+1)}$$

2. При $1 \leqslant r \leqslant 2$: $r^2 - 1 \geqslant 0$, тогда:

$$z_{min}(r) = \sqrt{r^2-1},\ z_{max}(r) = \sqrt{\frac{3}{5}(r^2+1)}$$

С учетом симметрии тела относительно плоскости $z = 0$, объем в цилиндрических координатах:

$$V = 2\pi \int_0^2 \left(2\left(z_{\max}(r) - z_{\min}(r)\right)\right) r, dr = 4\pi \int_0^2 r\left(z_{\max}(r) - z_{\min}(r)\right) dr. \tag{4}$$

Разбиваем интеграл на два:

$$V = 4\pi \left(\int_0^1 r\sqrt{\frac{3}{5}(r^2+1)}\, dr + \int_1^2 r\left(\sqrt{\frac{3}{5}(r^2+1)} - \sqrt{r^2-1}\right) dr \right) \tag{5}$$

Вычислим первый интеграл:

$$\int_0^1 r\sqrt{\frac{3}{5}(r^2+1)}\, dr = \frac{1}{2}\sqrt{\frac{3}{5}} \int_0^1 \sqrt{r^2+1}\, d(r^2+1) = \frac{1}{2}\sqrt{\frac{3}{5}}\frac{2}{3}\left(r^2+1\right)^{3/2}\Big|_0^1 = $$

$$= \boxed{\frac{1}{3}\sqrt{\frac{3}{5}}(2\sqrt{2}-1)} \tag{6}$$

Вычислим второй интеграл, разделив его тоже на две части:

$$\int_1^2 r\sqrt{\frac{3}{5}(r^2+1)}\, dr = \sqrt{\frac{3}{5}}\frac{1}{3}\left(r^2+1\right)^{3/2}\Big|_1^2 = \boxed{\frac{1}{3}\sqrt{\frac{3}{5}}(5\sqrt{5}-2\sqrt{2})} \tag{7}$$

$$\int_1^2 r\sqrt{r^2-1}\, dr = \frac{1}{2}\int_1^2 \sqrt{r^2-1}\, d(r^2-1) = \frac{1}{3}\left(r^2-1\right)^{3/2}\Big|_1^2 = \frac{1}{3}3\sqrt{3} = \boxed{\sqrt{3}} \tag{8}$$

Подставим значения интегралов в формулу (5):

$$V = 4\pi\left(\frac{1}{3}\sqrt{\frac{3}{5}}(2\sqrt{2}-1) + \frac{1}{3}\sqrt{\frac{3}{5}}(5\sqrt{5}-2\sqrt{2}) - \sqrt{3}\right) =$$

$$= 4\pi\left(\frac{1}{3}\sqrt{\frac{3}{5}}(5\sqrt{5}-1) - \sqrt{3}\right) = \boxed{\frac{4\pi\sqrt{15}(2\sqrt{5}-1)}{15}} \tag{9}$$

Вычислим примерное значение объема:

$$\boxed{V \approx 11.265771973094854} \tag{10}$$

Теперь, зная теоретическое значение объема тела, мы сможем анализировать точность получаемых данных.

III. ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Реализация метода Монте-Карла

Метод Монте-Карло для нахождения объема тела заключается в следующем алгоритме:

1. Генерируем `N` случайных значений для каждой оси пространства, в котором находится наша фигура, тем самым получая `N` точек;
2. Считаем количество точек, которые попали в область фигуры:
3. Находим по формуле объем нашего тела:

$$V_{\text{тела}} = \frac{M}{N} V,$$

где V — объем области, в которой считаем объем фигуры.

Назовем функцию `monte_carlo_3rd`, то есть для трехмерного пространства. В качестве параметров функции будет вводить:

- Функция, которая принимает координаты точки и возвращает `True`, если точка находится внутри фигуры, и `False` в противном случае;
- Пределы области по трем осям XYZ;
- Количество генерируемых точек;

Листинг 1 — `monte_carlo.py`

```
import numpy as np
from typing import Callable, Tuple

def monte_carlo_3rd(func: Callable[[np.array, np.array, np.array], bool],
                    x_lim: Tuple[float, float],
                    y_lim: Tuple[float, float],
                    z_lim: Tuple[float, float],
                    n: int) -> float:
    """ Вычисление объема функции методом Монте-Карло

    :param func: Функция для расчета объема. Должна принимать списки вещественных значений
    по трём осям XYZ и возвращать булево значение: находится ли точка внутри области функции.
    :type func: Callable[[np.array, np.array, np.array], bool]
    :param x_lim: Минимальный и максимальный пределы по оси X, в которых будет считаться объем
    :type x_lim: Tuple[float, float]
    :param y_lim: Минимальный и максимальный пределы по оси Y, в которых будет считаться объем
    :type y_lim: Tuple[float, float]
    :param z_lim: Минимальный и максимальный пределы по оси Z, в которых будет считаться объем
    :type z_lim: Tuple[float, float]
    :param n: Количество точек
    :type n: int
    :return: Объем функции
```

```
    :rtype: float
    """
    # Создаем N случайных точек
    x_random = np.random.uniform(*x_lim, n)
    y_random = np.random.uniform(*y_lim, n)
    z_random = np.random.uniform(*z_lim, n)

    # Считаем общий объем, в котором находится наша область
    v = (max(x_lim) - min(x_lim)) * (max(y_lim) - min(y_lim)) * (max(z_lim) - min(z_lim))

    # Считаем количество точек, которые находятся внутри функции
    m = np.sum(func(x_random, y_random, z_random))

    # Считаем и возвращаем объем
    return float(m / n * v)
```

2. Объем тела

Протестируем функцию на формуле (1) и на 1 000 000 точек:

Листинг 2 — task_16.py

```
from monte_carlo import monte_carlo_3rd

def check_inside_function(x: float, y: float, z: float) -> bool:
    """Проверяем, находится ли введенная точка внутри функции

    :param x: Координата по оси X
    :type x: float
    :param y: Координата по оси Y
    :type y: float
    :param z: Координата по оси Z
    :type z: float
    :return: Если находится внутри, то True, иначе False
    :rtype: bool
    """
    return (x*x + y*y - 1 <= z*z) & (z*z <= 3/5 * (x*x + y*y + 1))


V = monte_carlo_3rd(
    func=check_inside_function,
    x_lim=(-3, 3),
    y_lim=(-3, 3),
    z_lim=(-3, 3),
    n=1_000_000
)

print(f"{V=}")
```

В результате получается значение:

$$V = 11.248415999999999$$

$$\Delta V = |V_{\text{прак}} - V_{\text{теор}}| = 0.017355973094854704$$

3. Оценка времени и точности вычислений

Для проведения экспериментов на точность получаемых значений напишем программу, которая будет изменять количество случайно сгенерированных точек от 0 до 1 000 000 с шагом 2 000, чтобы уменьшить время вычисления, и вносить их в таблицу.

Чтобы было удобно обрабатывать данные, сразу будем вычислять:

- Среднеквадратичное отклонение (`std`);
- Среднее арифметическое значение (`mean`);
- Минимальное значение (`min`);
- Максимальное значение (`max`);
- Случайную погрешность значений (`random_error`);

Объема и затраченного времени на вычисление его. Случайная погрешность вычисляется по формуле:

$$\Delta V_{\text{сл}} = t_{\alpha,\infty} \cdot \frac{S_V}{3}, \tag{11}$$

где α — доверительный коэффициент, возьмем его равным $\alpha = 0.95$, тогда $t_{0.95,\infty} = 1.96$.

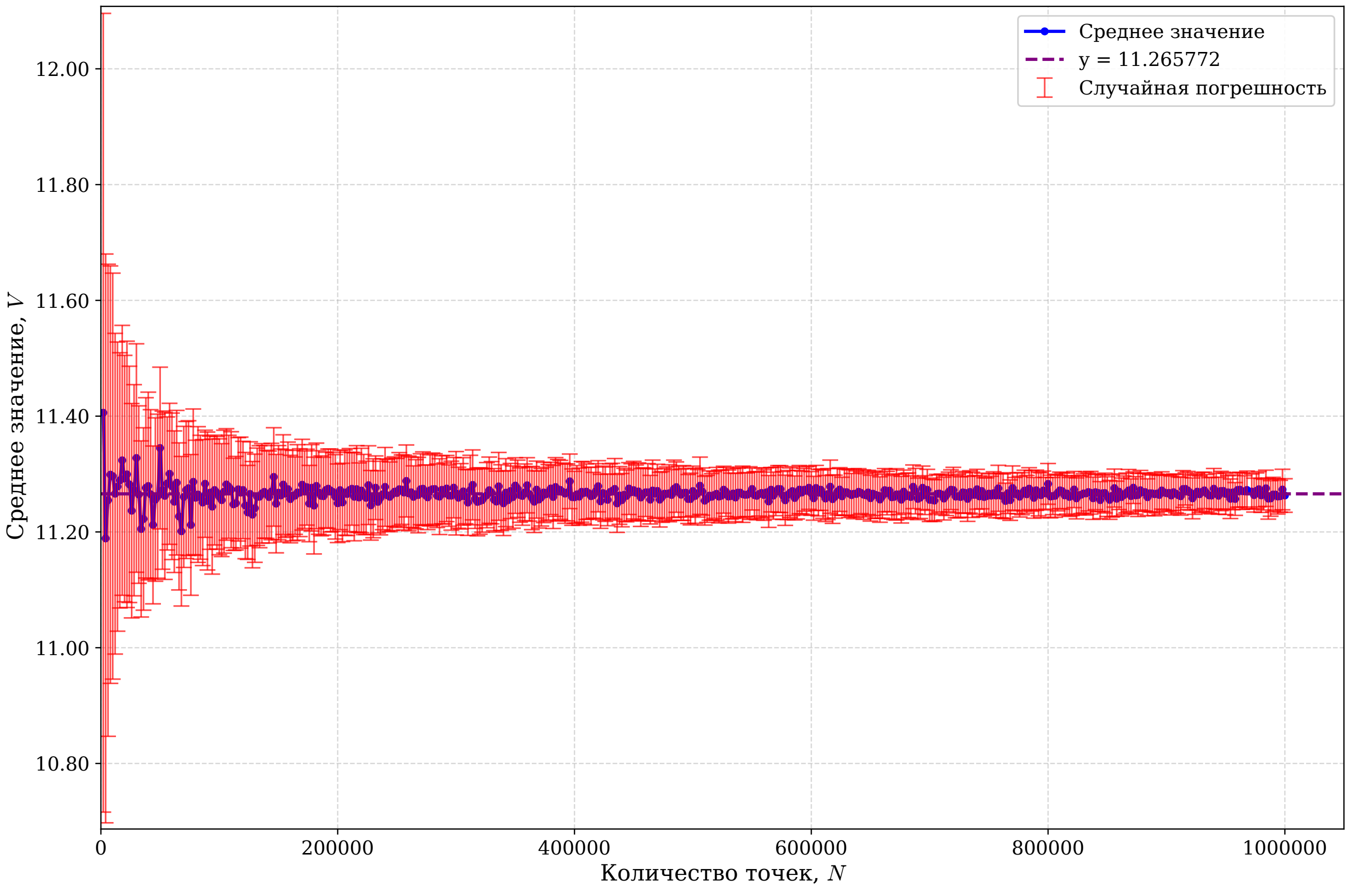

Рис. 2. Зависимость среднего значения объема от количества сгенерированных случайных точек

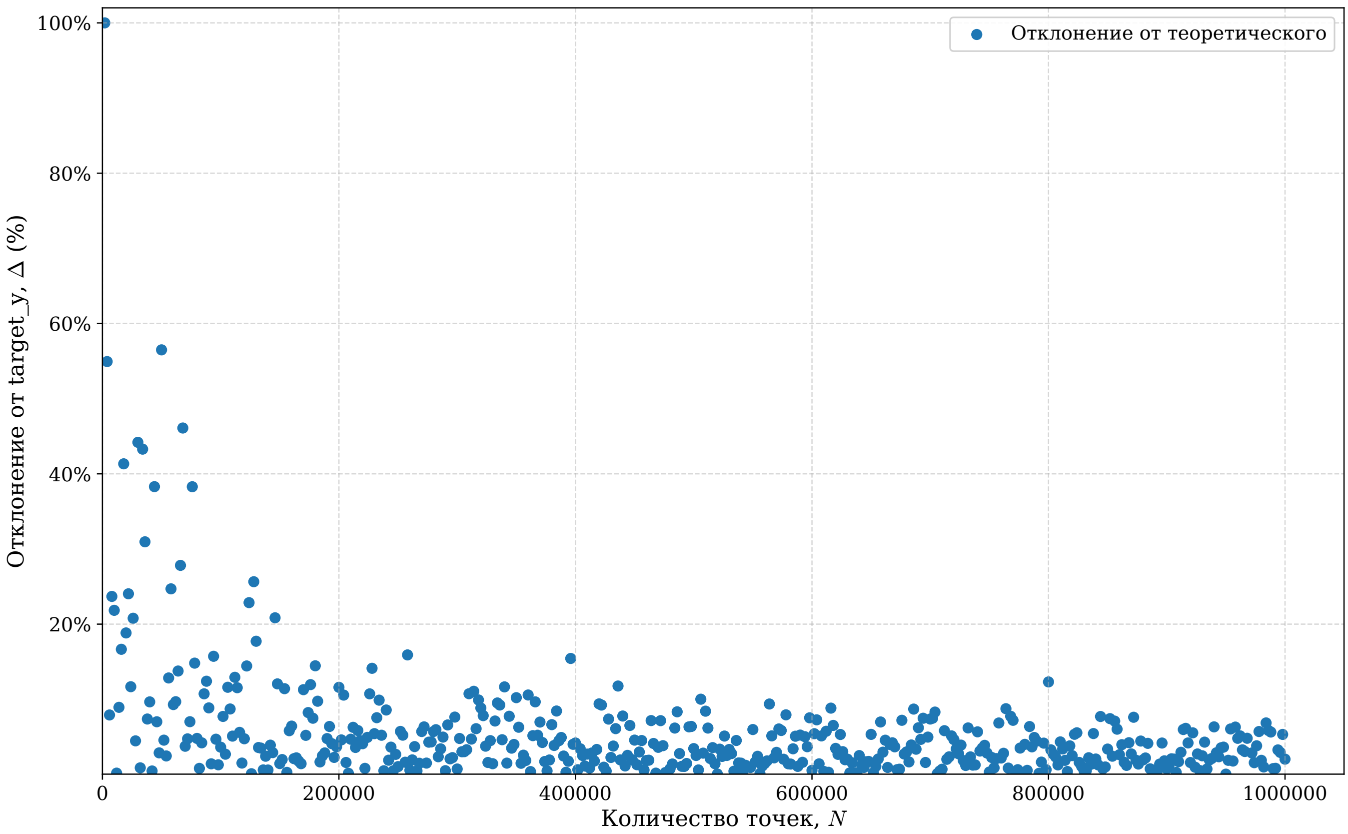

Рис. 3. Зависимость отношения разницы среднего объема тела, полученного методом Монте-Карло, и теоретического от количества сгенерированных случайных точек

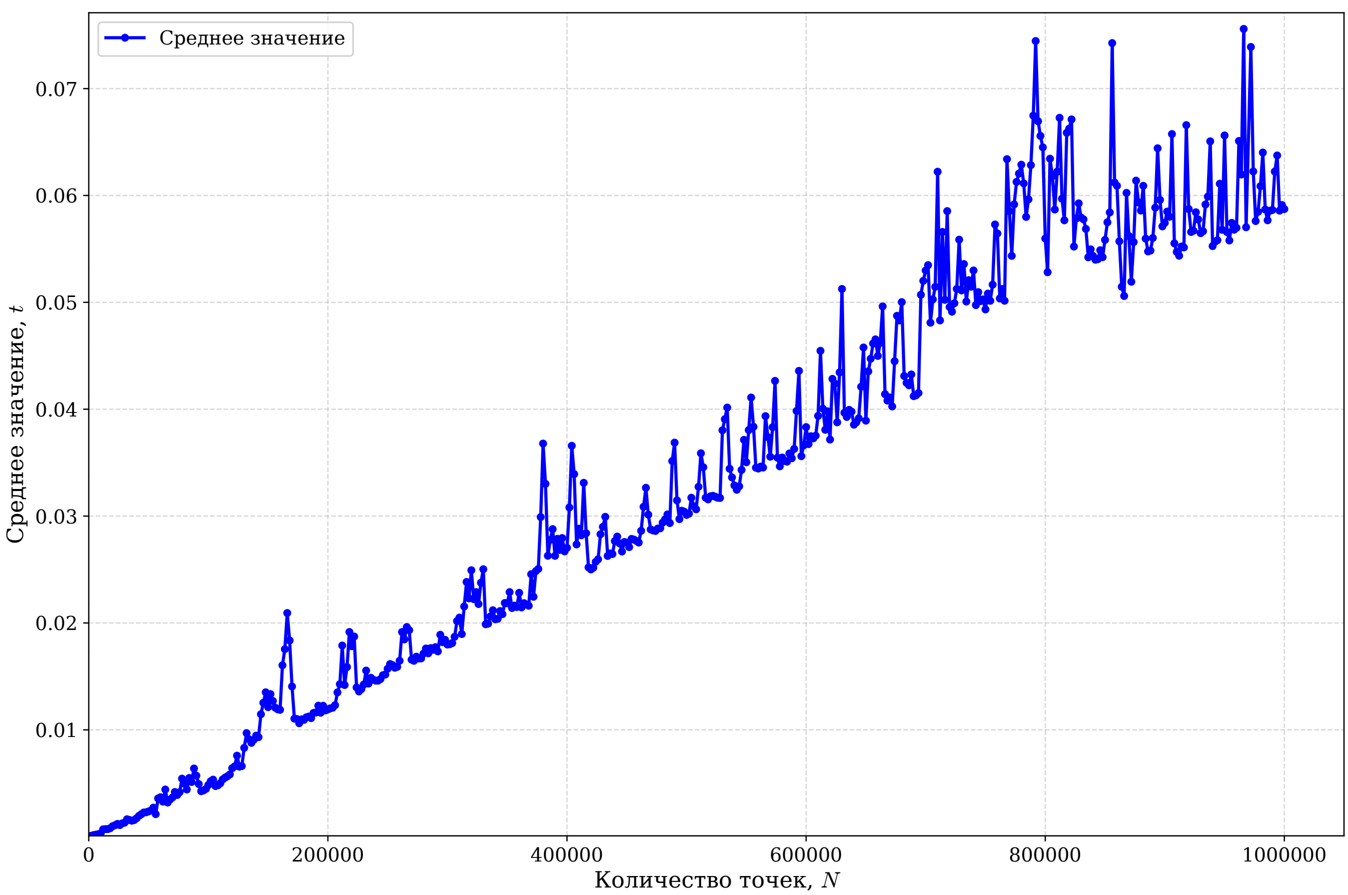

Рис. 4. Зависимость среднего значения времени от количества сгенерированных случайных точек

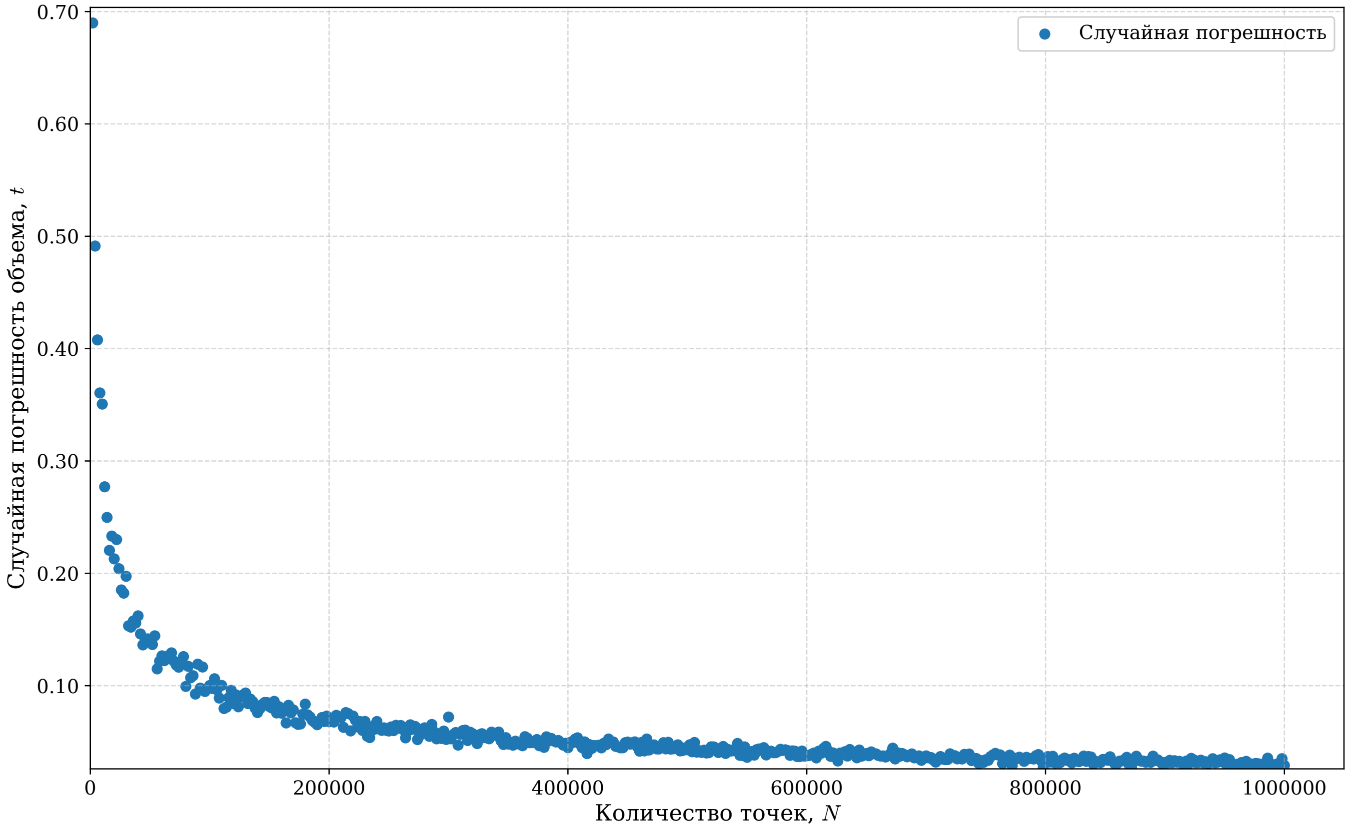

Рис. 5. Зависимость случайной погрешности от количества сгенерированных случайных точек

Из графиков (рис. 2 и 3) видно, что увеличение числа генерируемых точек повышает точность вычислений. Это проявляется в уменьшении разницы между теоретическим значением и средним практическим результатом. Графики демонстрируют обратную зависимость: с ростом числа точек скорость улучшения точности снижается.

Время на вычисление объема (рис. 4 и 5) увеличивается линейно. Это связано со сложностью алгоритма Монте-Карло, которая пропорциональна числу точек ($O(N)$).

4. Исходники и программы

Все программы представлены, таблицы и исходники документов загружены на GitHub:

https://github.com/SerKin0/IBTS-math-modeling-2k3s-monte-carlo-3d-volume

IV. ВЫВОД

На графиках зависимости точности и времени вычислений от числа сгенерированных точек наблюдается ожидаемое поведение метода Монте-Карло:

- С увеличением количества точек **погрешность расчёта объёма уменьшается**, приближаясь к теоретическому значению. При этом скорость снижения погрешности замедляется, что соответствует зависимости $O(1/\sqrt{N})$.
- **Время вычислений растёт линейно** с увеличением числа точек, что отражает вычислительную сложность алгоритма $O(N)$.

Таким образом, в ходе работы были успешно решены все поставленные задачи:

1. Разработана программная реализация метода Монте-Карло для вычисления объёма трёхмерного тела.
2. Вычислен объём заданной области с высокой точностью.
3. Проведён анализ точности и времени работы метода, подтвердивший его эффективность и предсказуемость.