

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина» (УрФУ)

Физико-технологический институт

Кафедра «Технической физики»

Оценка

Преподаватель

Кашин И.В.

**ЧИСЛЕННЫЙ РАСЧЕТ ОБЪЕМА СЛОЖНОГО ТЕЛА**

ОТЧЕТ

по лабораторной работе №2

|  |  |  |
| --- | --- | --- |
| Студент | Черняков Матвей Сергеевич | ФИО студента |

|  |
| --- |
| Специальность (направление подготовки) |
| 09.03.02 Информационные системы и технологии | |

|  |  |
| --- | --- |
| Группа | Фт-420008 |

Екатеринбург

2025

**СОДЕРЖАНИЕ**

[ОПИСАНИЕ ЗАДАЧИ 3](#_Toc208878197)

[ПРИНЦИП ПАРАЛЛЕЛИЗАЦИИ 6](#_Toc208878198)

[РЕШЕНИЕ ЗАДАЧИ 8](#_Toc208878199)

[ОЦЕНКА ЭФФЕКТИВНОСТИ ПАРАЛЛЕЛИЗАЦИИ 10](#_Toc208878200)

[ПРИЛОЖЕНИЕ А – ЛИСТИНГ КОДА 12](#_Toc208878201)

[ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А – ЛИСТИНГ КОДА 13](#_Toc208878202)

[ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А – ЛИСТИНГ КОДА 14](#_Toc208878203)

[ПРИЛОЖЕНИЕ Б – ТАБЛИЦА 2 15](#_Toc208878204)

[ПРИЛОЖЕНИЕ В – ТАБЛИЦА 3 16](#_Toc208878205)

ОПИСАНИЕ ЗАДАЧИ

В задаче рассматривается численный расчет объема сложного тела. Целью является сравнение точности и производительности двух способов вычислений — однопоточного и многопоточного. Для примера была задана функция в цилиндрических координатах, описывающая трехмерное тело:

Параметры были подобраны так, чтобы изображение было похоже на объемную звезду с полостью внутри (рисунок 1). Такой эффект достигается при параметрах:

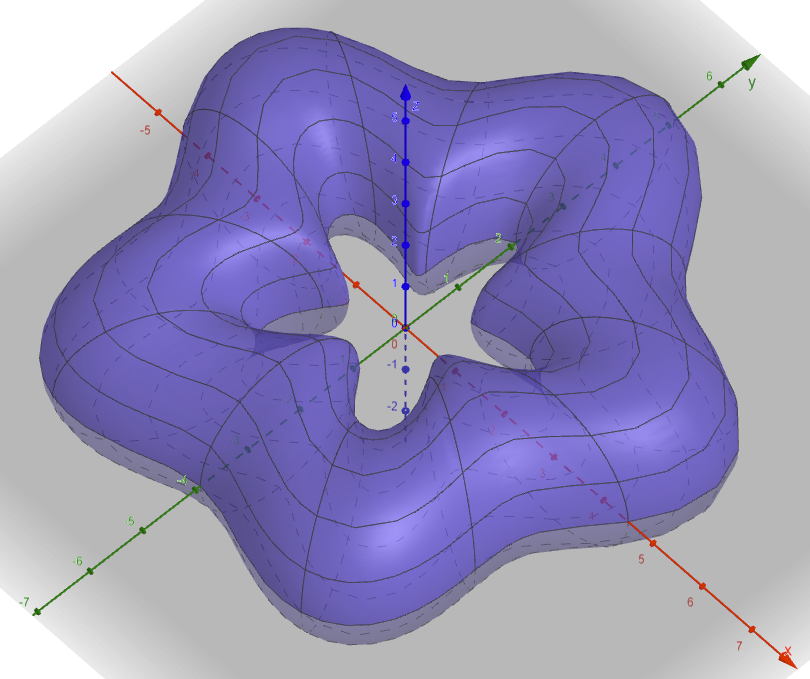


Рисунок 1 – Исходное трехмерное тело

При переводе в сферическую систему координат получим:

И если выразить :

Для проверки была построена фигура по формулам сферической системы координат на языке Python (рисунок 2).

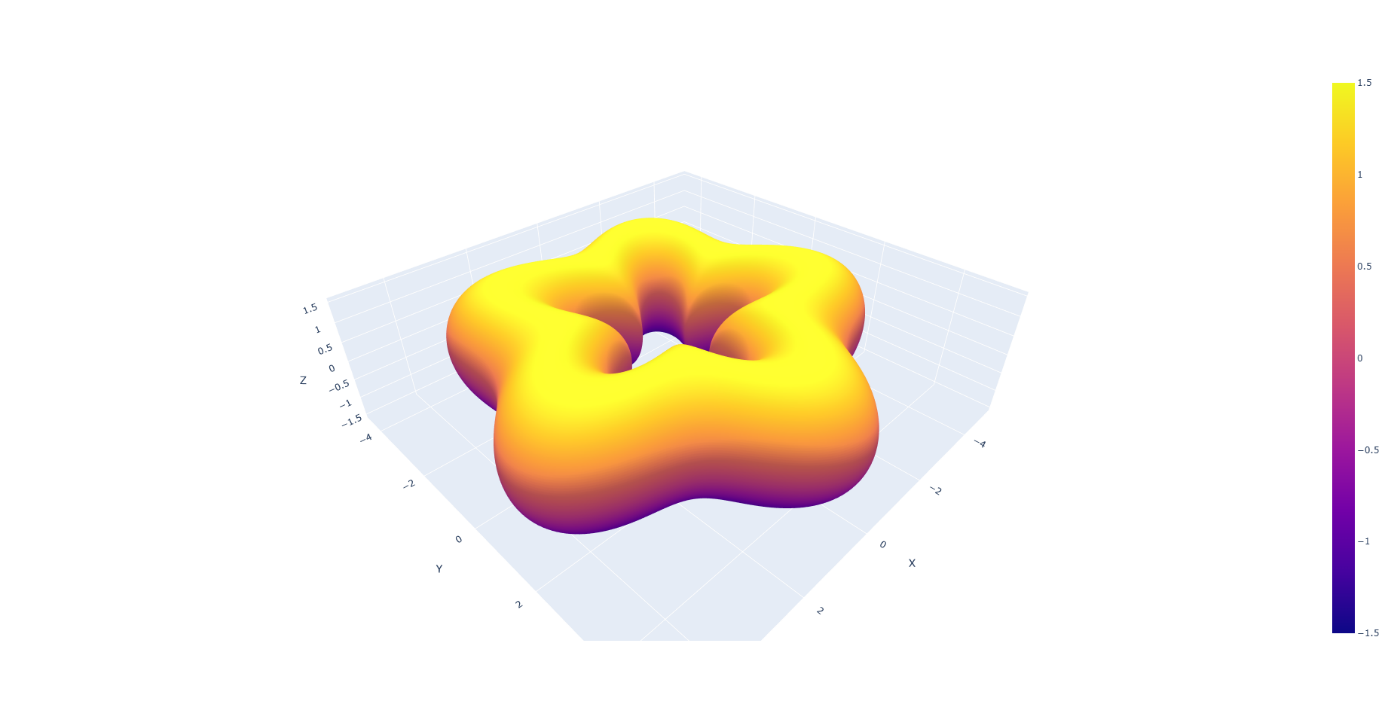


Рисунок 2 – Исходное тело, построенное в сферической системе координат

Объем можно найти аналитическими методами. Одним из примеров является метод вычисления через тройной интеграл. В сферической системе координат тройной элемент объёма выражается как:

где

Отсюда тройной интеграл для объёма записывается как:

где ρ′ — переменная интегрирования по радиусу

После ряда сокращений и подстановки радиуса получаем:

Решение данного интеграла можно получить аналитически или численно с помощью сторонних калькуляторов или Python-библиотек. Численное значение объема составляет 148.157509543295 (c точностью до 12 знаков после запятой).

ПРИНЦИП ПАРАЛЛЕЛИЗАЦИИ

Параллельные вычисления позволяют ускорять выполнение программы за счёт одновременной работы нескольких вычислительных единиц. Вместо того чтобы выполнять задачи последовательно, они делятся на части и обрабатываются одновременно, что особенно важно при больших объёмах вычислений.

Процессы — это независимые программы, каждая со своей областью памяти. Потоки же являются частью одного процесса и используют общую память (рисунок 3). Это делает потоки «легче», но сложнее в управлении, так как они могут мешать друг другу при доступе к одним и тем же данным.

Процесс обладает собственной выделенной памятью и ресурсами. Он изолирован от других процессов, что повышает стабильность, но делает переключение между процессами более затратным по ресурсам.

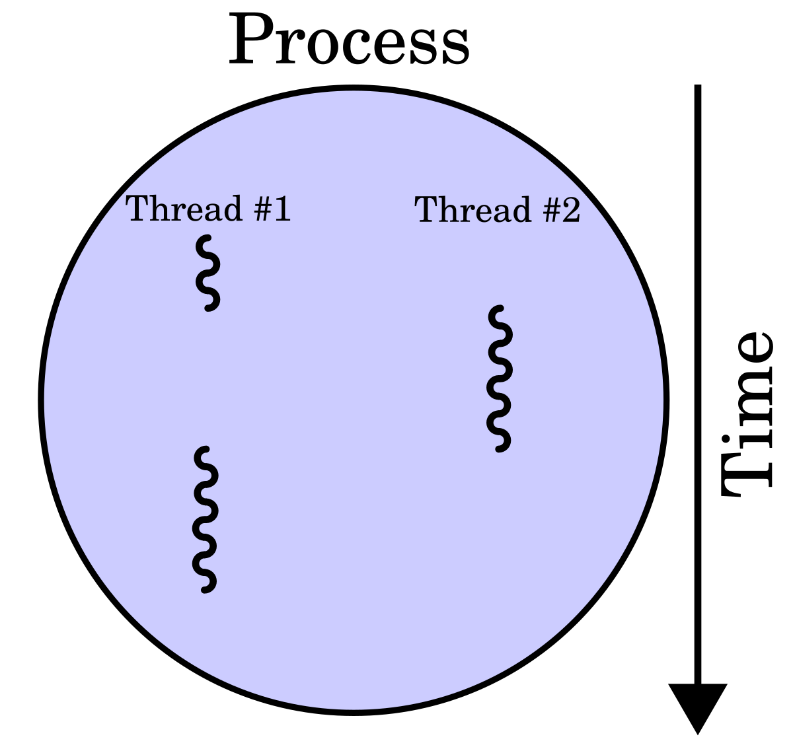


Рисунок 3 – Потоки внутри процессов

Потоки внутри одного процесса разделяют память, что ускоряет обмен данными, но в Python существует глобальная блокировка интерпретатора (GIL), которая позволяет одновременно выполняться только одному потоку на уровне Python-кода, что ограничивает возможности многопоточности.

Чтобы обойти ограничение GIL, в Python используется модуль multiprocessing, который создаёт отдельные процессы. Каждый процесс выполняется независимо и имеет собственную память, что позволяет полностью задействовать все ядра процессора.

Задача разбивается на части, которые распределяются между несколькими процессами. Каждый процесс выполняет свой участок работы, а затем результаты собираются и объединяются в итоговый ответ.

В рамках поставленной задачи это сводится к делению отрезка на равные части, где каждый поток забирает часть отрезков и вычисляет площадь под функцией на этих отрезках. Затем эти площади суммируются и получается единый результат.

РЕШЕНИЕ ЗАДАЧИ

В коде реализуются три основные функции и ключевой стартовый блок кода.

Функция function(x) – возвращает значение заданной функции в точке x.

def function(x):  
 return math.sin(x / 2) + 0.25 \* x

Функция get\_plank\_area(x, delta\_x) – возвращает площадь под функцией на заданном отрезке от точки x до x + .

def get\_plank\_area(x, delta\_x):  
 y\_n = function(x)  
 y\_n2 = function(x + delta\_x)  
 return delta\_x \* min(y\_n, y\_n2) + delta\_x \* 0.5 \* (max(y\_n, y\_n2) - min(y\_n, y\_n2))

Функция worker(args) – считает площадь для всех данных отрезков и возвращает сумму найденных площадей, а также время выполнения.

def worker(args):x, count, delta\_x = args  
 start = time.perf\_counter()  
 local\_sum = 0  
 for i in range(count):  
 local\_sum += get\_plank\_area(x, delta\_x)  
 x += delta\_x  
 end = time.perf\_counter()  
 return local\_sum, (end - start)

Далее для экспериментов используется вызов этих функций с помощью pool из библиотеки multiprocessing.

chunk\_size = N // M  
tasks = []  
for i in range(M):  
 start\_idx = LEFT\_BORDER + i \* chunk\_size \* delta\_x  
 tasks.append((start\_idx, chunk\_size, delta\_x))  
  
with multiprocessing.Pool(processes=M) as pool:  
 results = pool.map(worker, tasks)

ОЦЕНКА ЭФФЕКТИВНОСТИ ПАРАЛЛЕЛИЗАЦИИ

Используемый в тестированиях процессор AMD Ryzen 9 5900X имеет следующие характеристики, представленные в таблице 1.

Таблица 1. Характеристики AMD Ryzen 9 5900X.

|  |  |
| --- | --- |
| Released | 2020 |
| Segment | Desktop |
| Socket | Socket AM4 |
| **Number of cores** | **12** |
| **Number of threads** | **24** |
| Base frequency | 3700 MHz |
| Turbo Core | 4800 MHz |
| Unlocked multiplier | yes |
| Architecture (core) | Vermeer |
| Lithography | 7 nm |
| Transistors, mil. | 24 |
| TDP | 105 W |
| Max. temperature | 95° C |

Ключевыми параметрами является количество ядер (12) и количество потоков (24). Для данных характеристик было проведено два тестирования:

1. Сравнение идеального и условного времени вычисления при M от 1 до 24. Путем перебора значения N было выявлено, что результат превышает 20 секунд при значениях N>32000000. Результаты тестирования при N=32000000 представлены на верхнем графике (рисунок 4). Точные значения представлены в таблице 2 (приложение Б);
2. Зависимость ошибки от N при M=24. Перебор N осуществлялся от 24 до 576 с шагом 24, чтобы достичь равномерного распределения объема вычислений площадей. Результаты представлены на нижнем графике (рисунок 4). Точные значения представлены в таблице 3 (приложение В).

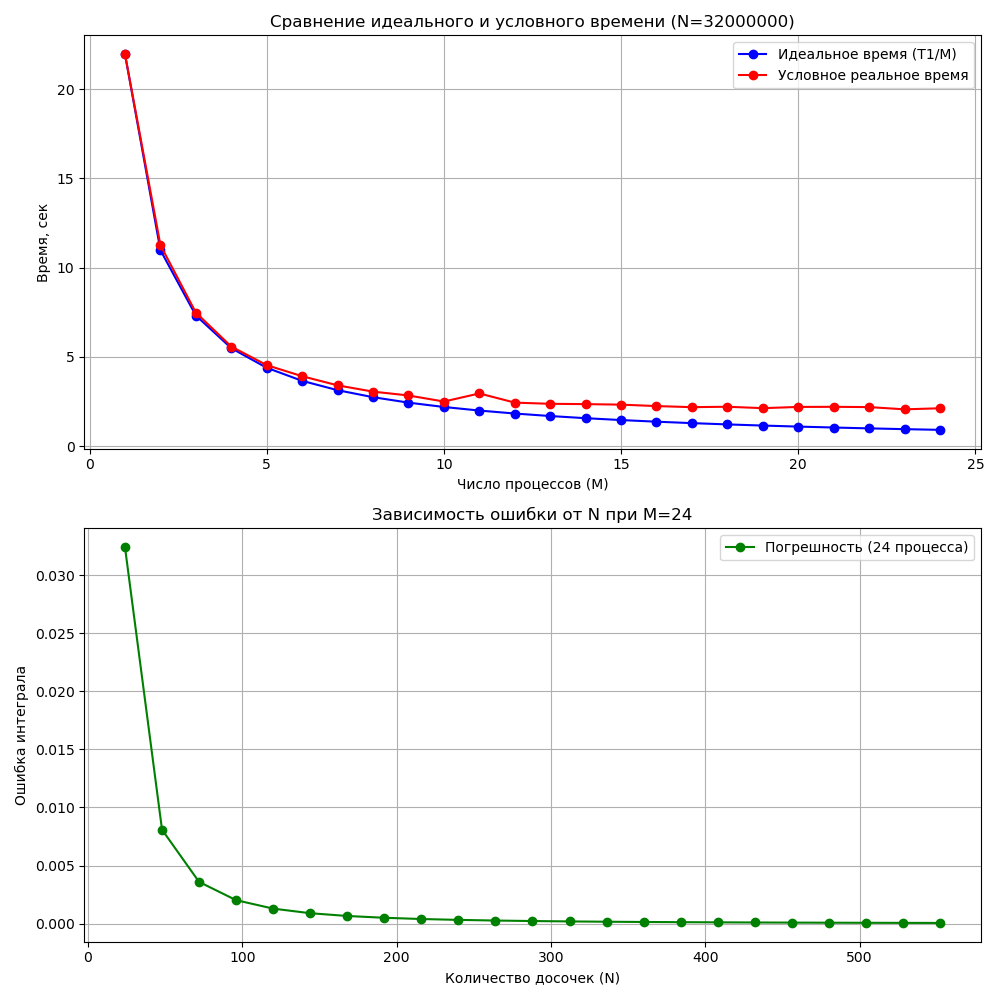


Рисунок 4 – Графики экспериментов

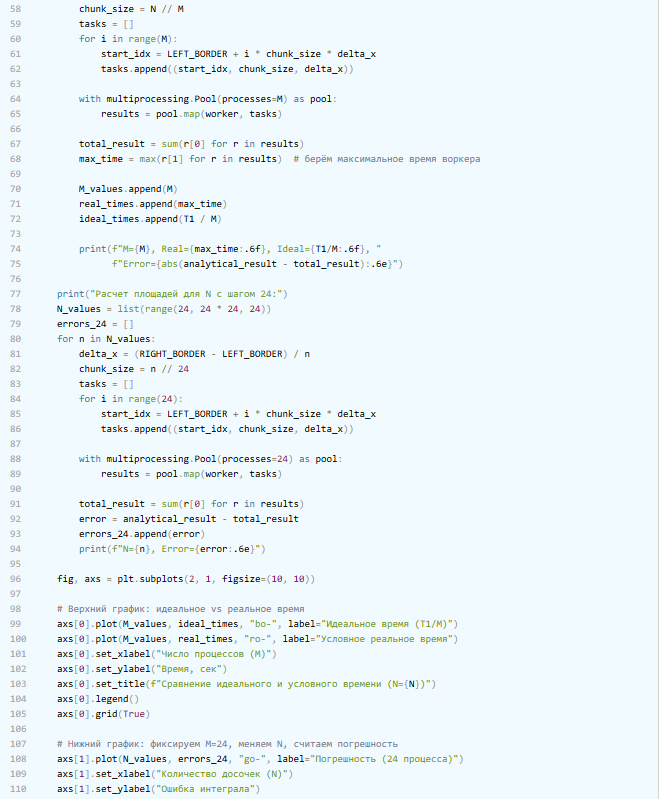
По графику для эксперимента 1 видно, что реальное время достаточно близко к идеальному и отклоняется на большом количестве потоков примерно на 1 секунду.

Второй график демонстрирует зависимость погрешности от числа N – чем больше делений участка функции, тем точнее результат.

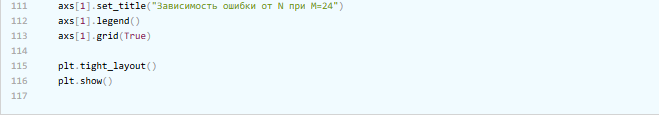
ПРИЛОЖЕНИЕ А – ЛИСТИНГ КОДА



ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А – ЛИСТИНГ КОДА



ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А – ЛИСТИНГ КОДА



ПРИЛОЖЕНИЕ Б – ТАБЛИЦА 2

Таблица 2. Время и ошибки для разного числа процессов M.

|  |  |  |  |
| --- | --- | --- | --- |
| **M** | **Real (сек)** | **Ideal (сек)** | **Error** |
| 1 | 21.953083 | 21.953088 | 1.828761e-08 |
| 2 | 11.261112 | 10.976544 | 8.353204e-09 |
| 3 | 7.480686 | 7.317696 | 5.002950e-06 |
| 4 | 5.578242 | 5.488272 | 8.870394e-09 |
| 5 | 4.538055 | 4.390618 | 7.203745e-09 |
| 6 | 3.917484 | 3.658848 | 5.007518e-06 |
| 7 | 3.407021 | 3.136155 | 1.002153e-05 |
| 8 | 3.054618 | 2.744136 | 3.855924e-09 |
| 9 | 2.840623 | 2.439232 | 1.252888e-05 |
| 10 | 2.500210 | 2.195309 | 3.230234e-09 |
| 11 | 2.953146 | 1.995735 | 2.506200e-05 |
| 12 | 2.440295 | 1.829424 | 2.004931e-05 |
| 13 | 2.373145 | 1.688699 | 1.503654e-05 |
| 14 | 2.355782 | 1.568078 | 1.002371e-05 |
| 15 | 2.326788 | 1.463539 | 1.253035e-05 |
| 16 | 2.245960 | 1.372068 | 1.933998e-09 |
| 17 | 2.185620 | 1.291358 | 4.010202e-05 |
| 18 | 2.208825 | 1.219616 | 3.508915e-05 |
| 19 | 2.129643 | 1.155426 | 2.506326e-05 |
| 20 | 2.198139 | 1.097654 | 1.542510e-09 |
| 21 | 2.204646 | 1.045385 | 2.756992e-05 |
| 22 | 2.189144 | 0.997868 | 2.506351e-05 |
| 23 | 2.064238 | 0.954482 | 2.005058e-05 |
| 24 | 2.123062 | 0.914712 | 2.005063e-05 |

ПРИЛОЖЕНИЕ В – ТАБЛИЦА 3

Таблица 3. Погрешности для 24 процессов, при разных N.

|  |  |
| --- | --- |
| **N** | **Error** |
| 24 | 3.240510e-02 |
| 48 | 8.087008e-03 |
| 72 | 3.593055e-03 |
| 96 | 2.020863e-03 |
| 120 | 1.293284e-03 |
| 144 | 8.980883e-04 |
| 168 | 6.598086e-04 |
| 192 | 5.051603e-04 |
| 216 | 3.991359e-04 |
| 240 | 3.232983e-04 |
| 264 | 2.671876e-04 |
| 288 | 2.245111e-04 |
| 312 | 1.912990e-04 |
| 336 | 1.649462e-04 |
| 360 | 1.436863e-04 |
| 384 | 1.262866e-04 |
| 408 | 1.118662e-04 |
| 432 | 9.978182e-05 |
| 456 | 8.955480e-05 |
| 480 | 8.082316e-05 |
| 504 | 7.330895e-05 |
| 528 | 6.679593e-05 |
| 552 | 6.111383e-05 |