

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

Специальность 1-40 04 01 «Информатика и технологии программирования»

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
по дисциплине «Операционные системы и среды»**

**на тему: «Распределенное компонентное приложение подготовки данных
для обработки и отображения поверхности»**

Исполнитель: студент гр. ИП-31
Федосов О.В.

Руководитель: преподаватель
Косинов Г.П.

Дата проверки: _____

Дата допуска к защите: _____

Дата защиты: _____

Оценка работы: _____

Подписи членов комиссии
по защите курсовой работы: _____

Гомель 2020

СОДЕРЖАНИЕ

Введение	3
1 Обзор существующих методов решения задачи	4
1.1 Решение интегралов	4
1.2 Выбор языка программирования для разработки приложения	8
1.3 Построение поверхности	9
1.4 Инструменты для распараллеливания вычислений на потоки	11
2 Алгоритмический анализ	13
2.1 Постановка задачи	13
2.2 Анализ исходных данных	13
2.3 Алгоритм обработки данных	13
2.4 Структура приложения	14
3 Разработка программного кода	17
3.1 Описания классов	17
3.2 Разработка графического интерфейса	22
4 Тестирование и верификация	28
4.1 Тестирование	28
4.2 Верификация и анализ полученных данных	35
Заключение	37
Список использованных источников	38
Приложения А - Листинг програмного кода	39

ВВЕДЕНИЕ

В курсовом проекте необходимо найти площадь поверхности численным методом трапеции.

При нахождении поверхности необходимо использовать технологию распараллеливания выбранного языка программирования. Количество поток указывается пользователем. Провести анализ погрешности и анализ зависимости времени выполнения от количества потоков.

Основное задачей курсового проекта является разработка приложения, которое позволит найти площадь поверхности. Приложение должно реализовывать построение поверхности и проведение анализа. Язык программирования и тип приложения выбирается произвольный.

					Курсовая проект	Лист
						3
Изм.	Лист	№ докум.	Подпись	Дата		

1 ОБЗОР МЕТОДОВ РЕШЕНИЯ ЗАДАЧИ

1.1 Решение интегралов

1.1.1 Интеграл

Интеграл — одно из важнейших понятий математического анализа, которое возникает при решении задач о нахождении площади под кривой, пройденного пути при неравномерном движении, массы неоднородного тела, и тому подобных, а также в задаче о восстановлении функции по её производной (неопределённый интеграл).

Двойные интегралы – это обобщение понятия определённого интеграла для функции двух переменных, заданной как $z = f(x, y)$.

Двойной интеграл записывается как показано на формуле (1.1):

$$\iint_D f(x, y) dx dy \quad (1.1)$$

Здесь D – плоская фигура, ограниченная линиями, выражения которых (равенства) даны в задании вычисления двойного интеграла. Слева и справа – равенствами, в которых слева переменная x , а сверху и снизу – равенствами, в которых слева переменная y . Это место и далее – одно из важнейших для понимания техники вычисления двойного интеграла.

Вычислить двойной интеграл – значит найти число, равное площади упомянутой фигуры D .

1.1.2 Метод трапеции

Метод трапеций – метод численного интегрирования функции одной переменной, заключающийся в замене на каждом элементарном отрезке подынтегральной функции на многочлен первой степени, то есть линейную функцию. Площадь под графиком функции аппроксимируется прямоугольными трапециями. Алгебраический порядок точности равен 1.

Если отрезок $[a, b]$ является элементарным и не подвергается дальнейшему разбиению, значение интеграла можно найти по формуле (1.2):

$$\int_a^b f(x) dx \approx \frac{f(a) + f(b)}{2} (b - a) + E(f), \quad E(f) = -\frac{f''(\xi)}{12} (b - a)^3 \quad (1.2)$$

Это простое применение формулы для площади трапеции — произведение полусуммы оснований, которыми в данном случае являются значения функции в крайних точках отрезка, на высоту (длину отрезка интегрирования).

Погрешность аппроксимации можно оценить через максимум второй

					Курсовая проект	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

производной (1.3):

$$|E(f)| \leq \frac{(b-a)^3}{12n^2} \max_{x \in [a,b]} |f''(x)|, \quad \frac{(b-a)^3}{12n^2} = \frac{nh^3}{12} \quad (1.3)$$

Аппроксимация функции линейной зависимостью при интегрировании методом трапеций показана на рисунке 1.1.

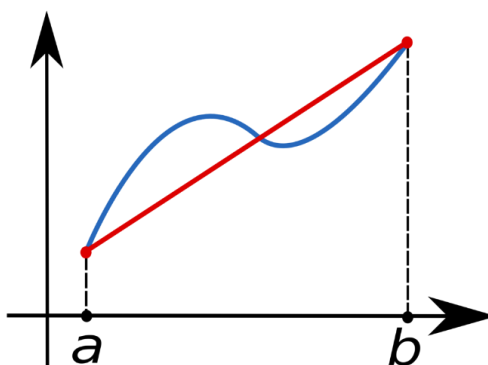


Рисунок 1.1 – Аппроксимация функции линейной зависимостью при интегрировании методом трапеций

Рассмотрим вопросы численного нахождения двойного интеграла по прямоугольной области $[a, b] \times [c, d]$ (1.4):

$$\int_a^b \int_c^d f(x, y) dy dx \quad (1.4)$$

Рассмотрим способа вывода аналога формулы трапеции для численного решения данного интеграла.

Вывод с помощью одномерных интегралов. Так как мы уже знаем формулу трапеции для интегралов от функций одной переменной, то удобно представить двойной интеграл через два интеграла, каждый из которых будет вычисляться от функции одной переменной и может быть численно найден с помощью уже известной нам одномерной формулы трапеции. С этой целью введем вспомогательную функцию $g(x)$ (1.5):

$$\int_a^b \int_c^d f(x, y) dy dx = \int_a^b g(x) dx, \quad g(x, y) = \int_c^d f(x, y) dy \quad (1.5)$$

Каждый из интегралов (1.6):

$$\int_a^b g(x, y) dx, \quad \int_c^d f(x, y) dy \quad (1.6)$$

можно вычислить с помощью формул численного интегрирования для одномерных интегралов.

Воспользуемся формулой трапеции и начнем с $g(x, y) = \int_c^d f(x, y) dy$. На отрезке $[c, d]$ введем равномерную сетку с шагом (1.7):

$$\begin{aligned} wh_y &= \{y_j = c + 0.5h_y + jh_y, \\ j &= 1, 2, \dots, N_y - 1, h_y N_y = d - c\} \end{aligned} \quad (1.7)$$

Тогда для рассматриваемого интеграла составная формула трапеции будет иметь вид (1.8):

$$g(x, y) = \int_c^d f(x, y) dy \approx \sum_{j=1}^{N_y - 1} f(x, y_j), \quad y_j \in wh_y \quad (1.8)$$

Для численного нахождения интеграла по переменной x на отрезке $[a, b]$ введем равномерную сетку с шагом h_x (1.9):

$$\begin{aligned} wh_x &= \{x_i = a + 0.5h_x + ih_x, \\ x &= 1, 2, \dots, N_x - 1, h_x N_x = b - a\}. \end{aligned} \quad (1.9)$$

на которой для интеграла $\int_a^b g(x, y) dx$ получим формулу трапеции (1.10):

$$\int_c^d g(x, y) dx \approx \sum_{i=1}^{N_x - 1} f(x_i, y), \quad x_i \in wh_x \quad (1.10)$$

Объединяя формулы по каждому направлению, получим составную формулу трапеции для двойного интеграла (1.11):

$$\int_a^b \int_c^d f(x, y) dy dx \approx \sum_{i=1}^{N_x-1} h_x \sum_{j=1}^{N_y-1} h_y f(x_i, y_j) \quad (1.11)$$

Вычислив данный интеграл, мы получим площадь данной нам поверхности. Составим интеграл для вычисления площади поверхности (1.12):

$$S_{\text{повер}} = \iint_D \sqrt{1 + \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2} \quad (1.12)$$

В данном курсовом проекте мы имеем следующую поверхность для интегрирования (1.13):

$$z = Ax^2 + By^2 + Cx + Dy \quad (1.13)$$

Вычислим производные для нашего интеграла (1.14):

$$\left(\frac{\partial z}{\partial x}\right)^2 = (2Ax + C)^2, \quad \left(\frac{\partial z}{\partial y}\right)^2 = (2By + D)^2 \quad (1.14)$$

Полученные производные подставим в наш интеграл (1.15):

$$S_{\text{повер}} = \iint_D \sqrt{1 + (2Ax + C)^2 + (2By + D)^2} \quad (1.15)$$

Вычислив данный интеграл методом трапеции, который был описан выше, мы получим площадь поверхности.

Применение составной формулы трапеции показано на рисунке 1.2.

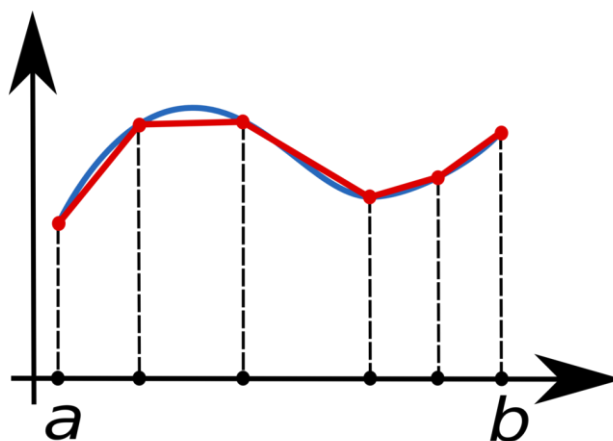


Рисунок 1.2 – Применение составной формулы трапеции

1.2 Выбор языка программирования для разработки приложения

1.2.1 Выбор язык программирования

В процессе анализа поставленной задачи было решено разработать приложение, реализующее все требования к курсовому проекту. Для разработки приложения использовался язык программирования Python. Выбор пал на Python потому что в нем легко можно строить различные графики, поверхности и диаграммы. Кроме того, в Python можно достаточно просто найти интеграл и провести сравнительный анализ, не прибегая к математическим пакетам типа MathCad. Пользовательский интерфейс будет создан с использованием фреймворка Django. Django позволяет создать WEB-приложение достаточно просто.

1.2.2 Python

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объем полезных функций.

Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

Django — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation.

Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из

существенных архитектурных отличий этого фреймворка от некоторых других. Один из основных принципов фреймворка — DRY

Также, в отличие от других фреймворков, обработчики URL в Django конфигурируются явно при помощи регулярных выражений.

Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.

1.3 Построение поверхности

1.3.1 Поверхность

Поверхность в геометрии и топологии — двумерное топологическое многообразие. Наиболее известными примерами поверхностей являются границы геометрических тел в обычном трёхмерном евклидовом пространстве. С другой стороны, существуют поверхности (например, бутылка Клейна), которые нельзя вложить в трёхмерное евклидово пространство без привлечения сингулярности или самопересечения.

Концепция поверхности применяется в физике, инженерном деле, компьютерной графике и прочих областях при изучении физических объектов. Например, анализ аэродинамических качеств самолёта базируется на обтекании потоком воздуха его поверхности.

Поверхность определяется как множество точек, координаты которых удовлетворяют определённому виду уравнений (1.16):

$$F(x, y, z) = 0 \quad (1.16)$$

Если функция $F(x, y, z)$ непрерывна в некоторой точке и имеет в ней непрерывные частные производные, по крайней мере одна из которых не обращается в нуль, то в окрестности этой точки поверхность, заданная уравнением (1.16), будет правильной поверхностью.

Помимо указанного выше неявного способа задания, поверхность может быть определена явно, если одну из переменных, например, z , можно выразить через остальные (1.17):

$$z = f(x, y) \quad (1.17)$$

Также существует параметрический способ задания. В этом случае поверхность определяется системой уравнений (1.18):

$$\begin{cases} x = x(u, v) \\ y = y(u, v) \\ z = z(u, v) \end{cases} \quad (1.18)$$

1.3.2 Построение графиков в Python при помощи Matplotlib

Визуализация данных — это большая часть работы специалистов в области data science. На ранних стадиях развития проекта часто необходимо выполнять разведочный анализ данных (РАД, Exploratory data analysis (EDA)), чтобы выявить закономерности, которые обнаруживают данные. Визуализация данных помогает представить большие и сложные наборы данных в простом и наглядном виде. На этапе окончания проекта важно суметь отчитаться о его результатах так, чтобы даже непрофессионалам, не обладающим техническими знаниями, всё стало ясно и понятно.

Matplotlib — это популярная библиотека для визуализации данных, написанная на языке Python. Хотя пользоваться ей очень просто, настройка данных, параметров, графиков и отрисовки для каждого нового проекта — занятие нудное и утомительное. В курсовом проекте необходимо будет построить поверхность и диаграмму.

Для построения поверхности используйте функцию `plot_surface()`. Пример поверхности показан на рисунке 1.3.

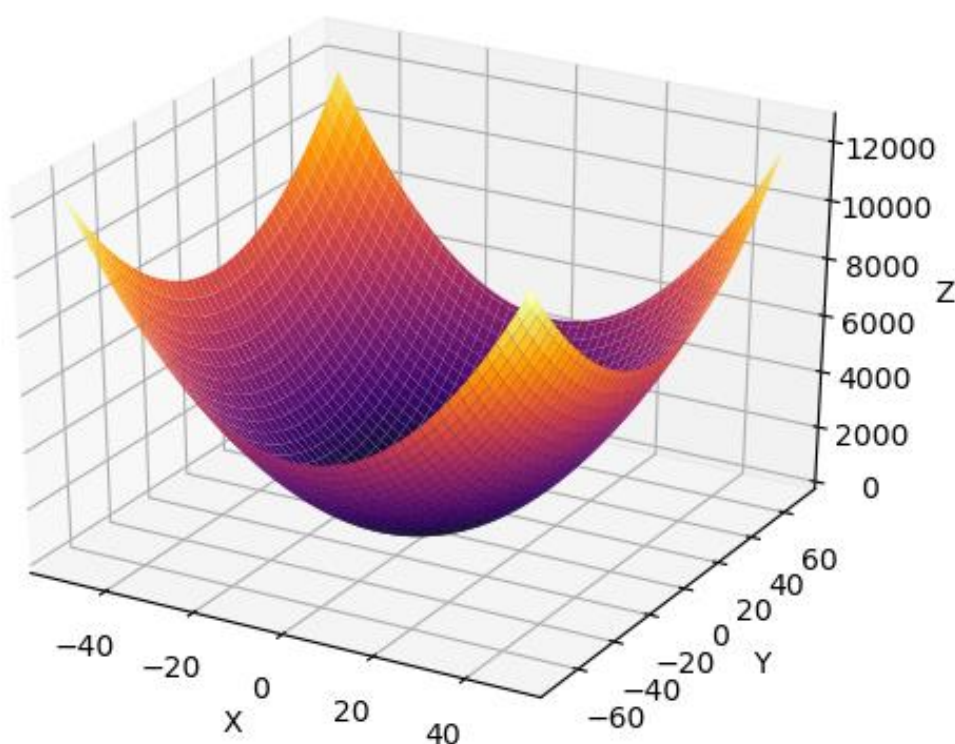


Рисунок 1.3 – Поверхность с использованием библиотеки Matplotlib

Для построения диаграммы используйте функцию `bar()`. Пример поверхности показан на рисунке 1.4.

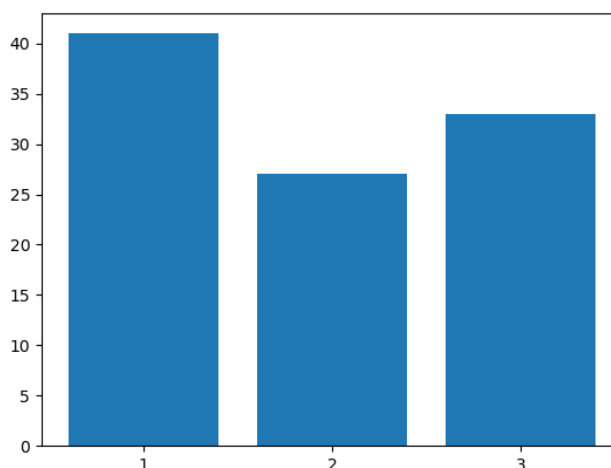


Рисунок 1.3 – Диаграмма с использованием библиотеки Matplotlib

1.4 Инструменты для распараллеливания вычислений на потоки

1.4.1 Параллельное вычисление в Python

Многопроцессорная обработка – это использование двух или более процессорных блоков в одной компьютерной системе. Это лучший способ получить полный потенциал от нашего оборудования, используя полное количество процессорных ядер, доступных в нашей компьютерной системе.

Многопоточная обработка – это способность ЦП управлять использованием операционной системы путем одновременного выполнения нескольких потоков. Основная идея многопоточности заключается в достижении параллелизма путем разделения процесса на несколько потоков.

При работе с параллельными приложениями в Python есть ограничение, называемое GIL (Global Interpreter Lock). GIL никогда не позволяет нам использовать несколько ядер CPU, и поэтому мы можем сказать, что в Python нет настоящих потоков. GIL – мьютекс – блокировка взаимного исключения, которая делает вещи безопасными. Другими словами, мы можем сказать, что GIL препятствует параллельному выполнению кода Python несколькими потоками. Блокировка может удерживаться только одним потоком за раз, и если мы хотим выполнить поток, он должен сначала получить блокировку.

Используя многопроцессорность, мы можем эффективно обойти ограничение, вызванное GIL:

- 1) Используя многопроцессорность, мы используем возможности нескольких процессов и, следовательно, мы используем несколько экземпляров GIL.
- 2) В связи с этим нет ограничений на выполнение байт-кода одного потока в наших программах одновременно.

Используя многопроцессорность, мы используем возможности нескольких процессов и, следовательно, мы используем несколько экземпляров GIL. В связи с этим нет ограничений на выполнение байт-кода одного потока в наших программах одновременно.

Таблица 1.1 показывает некоторые важные различия между многопроцессорной и мультипрограммной обработки.

Таблица 1.1 – Сравнение многопроцессорной и мультипрограммной обработки

Многопроцессорная обработка	Мультипрограммирование
Под многопроцессорной обработкой понимается обработка нескольких процессов одновременно несколькими процессорами.	Мультипрограммирование хранит несколько программ в основной памяти одновременно и выполняет их одновременно, используя один процессор.
Он использует несколько процессоров.	Он использует один процессор.
Это позволяет параллельную обработку.	Переключение контекста происходит.
Меньше времени уходит на обработку работ.	Больше времени уходит на обработку заданий.
Это способствует гораздо более эффективному использованию устройств компьютерной системы.	Менее эффективен, чем многопроцессорная.

1.4.2. Применение параллельного вычисления в курсовом проекте

Для распараллеливания в курсовом проекте будет использоваться библиотека multiprocessing. Библиотека предоставляет возможность создание нескольких процессов одновременно, что позволяет обойти GIL и выполнять вычисление параллельно. Для реализации многопроцессорности библиотека multiprocessing предоставляет различное количество классов, которые имеют свои особенности. В курсовом проекте будет использоваться класс Process, который позволяет создавать новые процессы. Для общения между процессами будет использоваться класс Value, которая создается в общей памяти и позволяет создать переменную определенного типа и передавать ее процессам.

Для создания нового процесса, необходимо передать в конструктор класс Process целевой метод, для которого будет создан процесс, и параметры метода. После этого необходимо запустить, используя метод *start()*. Что бы дождаться выполнения процесса, необходимо вызвать метод *join()*. Количество созданных процессов неограниченно. Для создания несколько процессов одновременно, можно использовать массив. Каждый созданный процесс мы сохраним как элемент массив, а для ожидания выполнения каждого процесса используем перебор элементов массива.

Для создания общей переменной для всех процессов необходимо передать в конструктор класса Value тип переменной и начальное значение. После этого передать переменную как один из параметров функции, для которой создан процесс. Чтобы присвоить или получить значение необходимо использовать свойство *value*.

2 АЛГОРИТМИЧЕСКИЙ АНАЛИЗ

2.1 Постановка задачи

Были поставлены следующие задачи, которые необходимо было решить:

1. Провести анализ заданной поверхности (согласно варианту) в математическом пакете. Анализ включает: построение графика поверхности, вычисления площади поверхности, используя аналитическую формулу (1.12);
2. Провести численный анализ заданной поверхности, используя формулы численного интегрирования (согласно варианту). Программный код оформить в виде компонента, позволяющий принимать в качестве параметров границы области интегрирования, вычислять площадь заданной поверхности в пределах интегрирования, готовить данные для построения графика в математическом пакете;
3. Провести анализ погрешности численных методов, передав данные, полученные в пункте 2 в математический пакет. Программный код оформить в виде компонента;
4. Провести анализ скорости от количества потоков, используемых для расчетов.

2.2 Анализ исходных данных

Необходимо разработать распределенное компонентное приложение подготовки данных для обработки и отображения поверхности.

Методом численного интегрирования выбрать метод Трапеции.

Вид поверхности: $z = Ax^2 + By^2 + Cx + Dy$

Для данной поверхности X_n , X_k , Y_n , Y_k принять -50, 50, -70, 70 соответственно.

Значения коэффициентов:

$A = 3$, $B = 1$, $C = 5$, $D = -5$

В программе предусмотреть возможность отображения поверхности, расчет площади заданной поверхности в заданных пределах, анализ влияния распараллеливания выполнения расчета площади на время.

Проверить результат нахождения площади и построения поверхности в математическом пакете, используя стандартные функции, и сравнить результаты.

2.3 Алгоритм обработки данных

Для нахождения площади поверхности необходимо указать параметры функции, начало и конец интервалов X , Y и количество разбиений. Далее происходит вычисление площади поверхности.

Для построения поверхности необходимо указать файлы с данными по осям X , Y , Z .

					Курсовая проект	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

Вычисления погрешностей.

Полученное из опыта значение измеряемой величины может отличаться от ее действительного (истинного) значения.

Погрешность измерения – отклонение результата измерения от истинного (действительного) значения измеряемой величины.

Бывает полезно узнать, как сильно приближенное значение числа отличается от его точного значения. Понятно, что чем это различие меньше, тем лучше, тем точнее выполнено измерение или вычисление.

Для определения точности измерений (вычислений) вводят такое понятие как погрешность приближения.

Существует множество видов погрешностей. В данном курсовом проекте используются погрешности двух видов:

1. абсолютная погрешность;
2. относительная погрешность.

Абсолютной погрешностью числа называют разницу между этим числом и его точным значением.

Абсолютной погрешностью ΔX называется разность между измеренным и действительным значениями. Формула погрешности (2.1):

$$\Delta X = X_1 - X_D \quad (2.1)$$

где X_I – измеренное значение, X_D – действительное значение измеряемой величины.

Абсолютная погрешность и предельная абсолютная погрешность не достаточны для характеристики точности измерения или вычисления. Качественно более существенна величина относительной погрешности.

Различают систематические и случайные погрешности.

Систематическая погрешность – это составляющая погрешности измерения, остающаяся постоянной или закономерно изменяющаяся при повторных измерениях одной и той же величины.

Случайная погрешность – это составляющая погрешности измерения, изменяющаяся случайным образом при повторных измерениях одной и той же величины. Наличие случайных погрешностей выявляется при проведении ряда измерений постоянной физической величины, когда оказывается, что результаты измерений не совпадают друг с другом.

2.4 Структура приложения

Структура приложения показана на рисунке 2.1.

					Курсовая проект	Лист
						14
Изм.	Лист	№ докум.	Подпись	Дата		

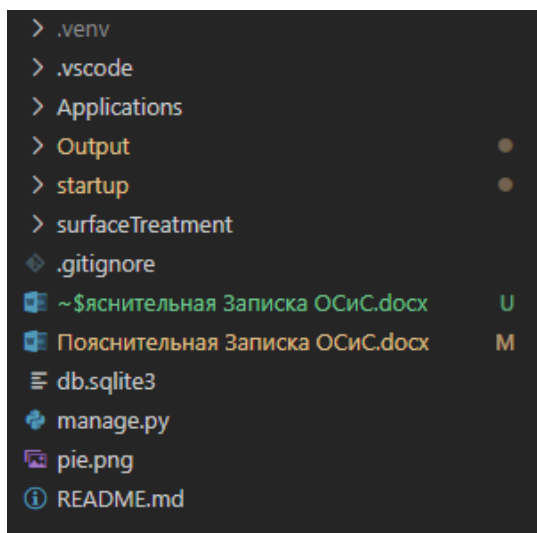


Рисунок 2.1 – Структура приложения

Структура приложения состоит из следующих основных элементов:

1. Applications – в этом разделе находится дополнительная информация для курсового проекта;
2. Output – в этом разделе хранятся файлы с выходными данными;
3. Startup – в этом разделе находятся все *templates* (страницы), стили, изображения и библиотеки;
4. SurfaceTreatment – в этом разделе находится основное приложение.

Для выполнения поставленной задачи, приложение должно реализовывать следующий функционал:

- 1) расчет площади заданной поверхности заданным численным методом, согласно варианту;
- 2) построение графика поверхности;
- 3) анализ влияния распараллеливания вычисления площади поверхности заданным численным методом на время вычисления и относительную погрешность относительно аналитического значения.

Структурная схема приложения показана на рисунке 2.2.

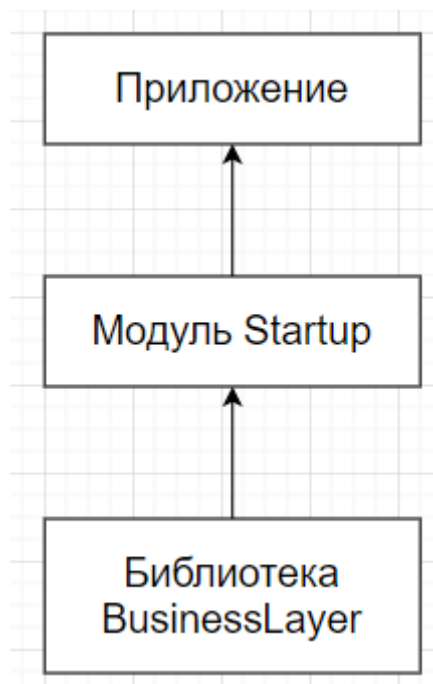


Рисунок 2.2 – Структурная схема приложения

Модули комплексного приложения для решения задачи:

- 1) главное приложение (SurfaceTreatment);
- 2) библиотека (BusinessLayer);
- 3) модуль (Startup).

Библиотека BusinessLayer выполняет основную бизнес-логику. Она состоит из двух классов:

- 1) TrapezoidMethod: используется для вычисления двойного интеграла, распараллеливания и построения поверхности и диаграммы;
- 2) FileHelper: используется для записи данных в файл и чтения из файла.

Модуль Startup реализует весь пользовательский интерфейс и основной функционал нахождения и построения поверхности. В Startup подключен модуль BusinessLayer для работы с поверхностью.

Фреймворк Django позволяет создать приложение из одно или несколько вспомогательных приложений (модулей). Таким образом можно легко изменить приложение, подключив другой модуль или убрать один из модулей приложения (если их несколько).

Приложение будет использовать один модуль – Startup.

3 РАЗРАБОТКА ПРОГРАММНОГО КОДА

3.1 Описания классов

Методы класса TrapezoidMethod показаны в таблице 3.1. Так как Python не является строго типизированным язык программирования, тип входящих параметров и тип возвращаемых значений не указывается.

Таблица 3.1 – Описание TrapezoidMethod

Наименование методов	Описание
<i>setParams</i>	Установка значений для параметров A, B, C, D
<i>setIntervals</i>	Установка значений для интервалов X_n , X_k , Y_n , Y_k
<i>execute</i>	Нахождение площади поверхности
<i>executeAnalysis</i>	Нахождения площади поверхности для анализа
<i>calcFormY</i>	Нахождения интеграла по Y
<i>calcFormX</i>	Нахождение интеграла по X
<i>getFunc</i>	Получение результат функции площади поверхности
<i>getFuncForY</i>	Получение производной по Y
<i>getFuncForX</i>	Получение производной по X
<i>getFuncForWrite</i>	Получение функции для записи в файл
<i>getMatrix</i>	Получение матрицы Z
<i>draw</i>	Отрисовка поверхности
<i>drawAnalysis</i>	Отрисовка диаграммы
<i>writeFile</i>	Запись в файл частичного результата (при использовании нескольких процессов)

Параметры метода *setParams* показаны в таблице 3.2.

Таблица 3.2 – Описание параметров метода *setParams*

Наименование методов	Описание
<i>a</i>	Параметр функции А
<i>b</i>	Параметр функции В
<i>c</i>	Параметр функции С
<i>d</i>	Параметр функции D

Параметры метода *setIntervals* показаны в таблице 3.3.

Таблица 3.3 – Описание параметров метода *setIntervals*

Наименование методов	Описание
<i>xs</i>	Начальное значение интервала X
<i>xf</i>	Конечное значение интервала X
<i>ys</i>	Начальное значение интервала Y
<i>yf</i>	Конечное значение интервала Y

Параметры метода *execute* показаны в таблице 3.4.

Таблица 3.4 – Описание параметров метода *execute*

Наименование методов	Описание
<i>n</i>	Количество делений
<i>processNumber</i>	Количество процессов

Параметры метода *executeAnalysis* показаны в таблице 3.5.

Таблица 3.5 – Описание параметров метода *executeAnalysis*

Наименование методов	Описание
<i>n</i>	Количество делений
<i>processNumber</i>	Количество процессов

Параметры метода *calcFormY* показаны в таблице 3.6.

Таблица 3.6 – Описание параметров метода *calcFormY*

Наименование методов	Описание
x	Список точек для X
y_s	Начальное значение Ys
h_y	h для интеграла по Y
h_x	h для интеграла по X
n	Количество делений
num	Переменная, в которой будет сохранен результат вычисления

Параметры метода *calcFormX* показаны в таблице 3.7.

Таблица 3.7 – Описание параметров метода *calcFormX*

Наименование методов	Описание
x	Список точек для X
y	Значение Y
h_x	h для интеграла по X

Параметры метода *getFunc* показаны в таблице 3.8.

Таблица 3.8 – Описание параметров метода *getFunc*

Наименование методов	Описание
x	Список точек для X
y	Значение Y

Параметры метода *getFuncForY* показаны в таблице 3.9.

Таблица 3.9 – Описание параметров метода *getFuncForY*

Наименование методов	Описание
y	Значение Y

Параметры метода *getFuncForX* показаны в таблице 3.10.

Таблица 3.10 – Описание параметров метода *getFuncForX*

Наименование методов	Описание
<i>x</i>	Значение X

Параметры метода *getFuncForWrite* показаны в таблице 3.11.

Таблица 3.11 – Описание параметров метода *getFuncForWrite*

Наименование методов	Описание
<i>x</i>	Значение X
<i>y</i>	Значение Y

Параметры метода *draw* показаны в таблице 3.12.

Таблица 3.12 – Описание параметров метода *draw*

Наименование методов	Описание
<i>x</i>	Значение X
<i>y</i>	Значение Y
<i>z</i>	Значение Z

Параметры метода *drawAnalysis* показаны в таблице 3.13.

Таблица 3.13 – Описание параметров метода *drawAnalysis*

Наименование методов	Описание
<i>times</i>	Время
<i>procNumbers</i>	Количество процессов

Параметры метода *writeFile* показаны в таблице 3.14.

Таблица 3.14 – Описание параметров метода *writeFile*

Наименование методов	Описание
<i>result</i>	Результат для записи в файл

Методы класса FileHelper показаны в таблице 3.15.

Таблица 3.15 – Описание FileHelper

Наименование методов	Описание
<i>writeToFiles</i>	Запись данных для поверхности в файлы
<i>writeToFile</i>	Запись частичного результат в файл
<i>readOfFiles</i>	Чтение из файлов

Параметры метода *writeToFiles* показаны в таблице 3.16.

Таблица 3.16 – Описание параметров метода *writeToFiles*

Наименование методов	Описание
<i>xArraysPath</i>	Путь до файла с данными по X
<i>yArraysPath</i>	Путь до файла с данными по Y
<i>zArraysPath</i>	Путь до файла с данными по Z

Параметры метода *writeToFile* показаны в таблице 3.17.

Таблица 3.17 – Описание параметров метода *writeToFile*

Наименование методов	Описание
<i>result</i>	Данные для сохранения в файл
<i>squarePath</i>	Путь до файла с данными площади

Параметры метода *readOfFiles* показаны в таблице 3.17.

Таблица 3.17 – Описание параметров метода *readOfFiles*

Наименование методов	Описание
<i>xArraysPath</i>	Путь до файла с данными по X
<i>yArraysPath</i>	Путь до файла с данными по Y
<i>zArraysPath</i>	Путь до файла с данными по Z

3.2 Разработка графического интерфейса

Стартовая страница приложения показан на рисунке 3.1.

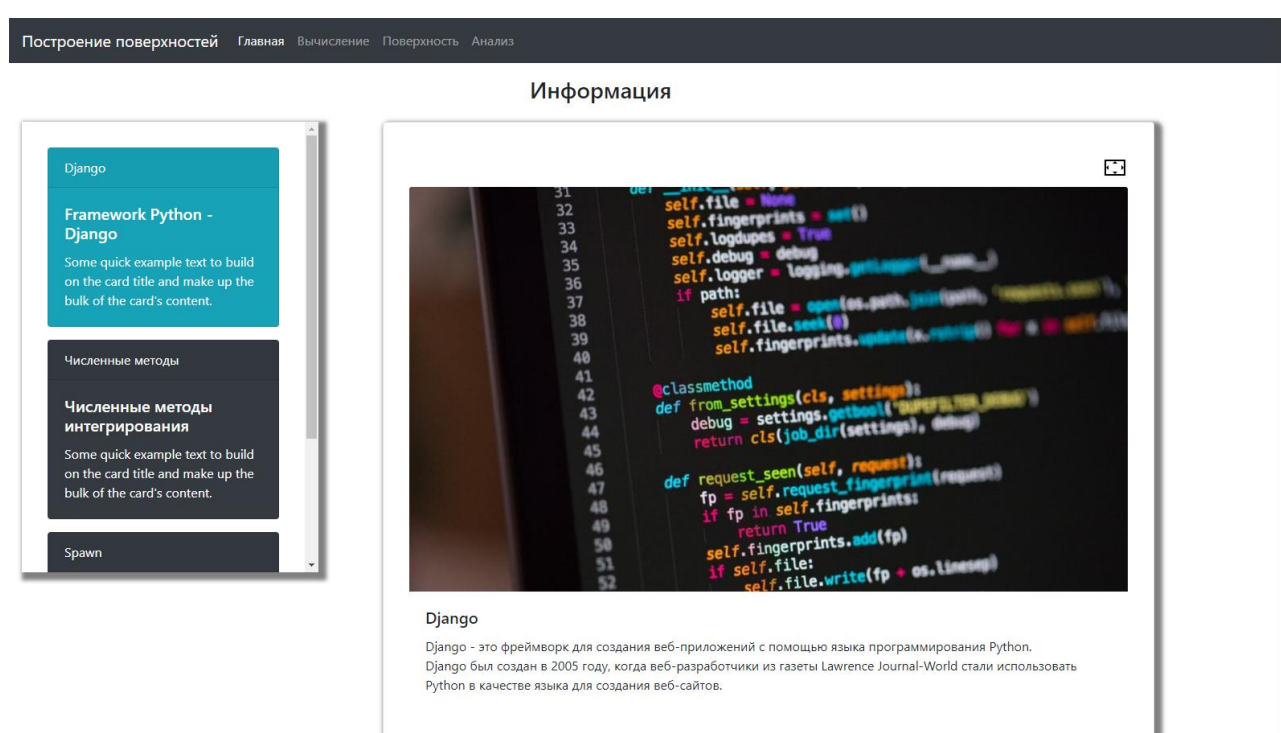


Рисунок 3.1 – Стартовая страница приложения

На стартовой странице отображена вся необходимая информация о курсовом проекте: описание фреймворка Django, численного метода, распараллеливания и др.

Кроме того, всю предоставленную информацию можно просматривать в полноэкранном режиме. Полноэкранный режим показан на рисунке 3.2.

- 2) Интервалы: в данном блоке указываются интервалы для интеграла: начальное значение X , Y и конечное значение X , Y ;
- 3) Процессы: в данном блоке указывается количество процессов (для параллельного вычисления) и количество делений;
- 4) Дополнительные настройки: в данном блоке указывается, необходимо ли сохранять данные, можно также указать имена файлов для сохранения и отобразить точное вычисление.

Результат вычисления отображается в разделе Очередь. Пример результата вычисления показан на рисунке 3.3.

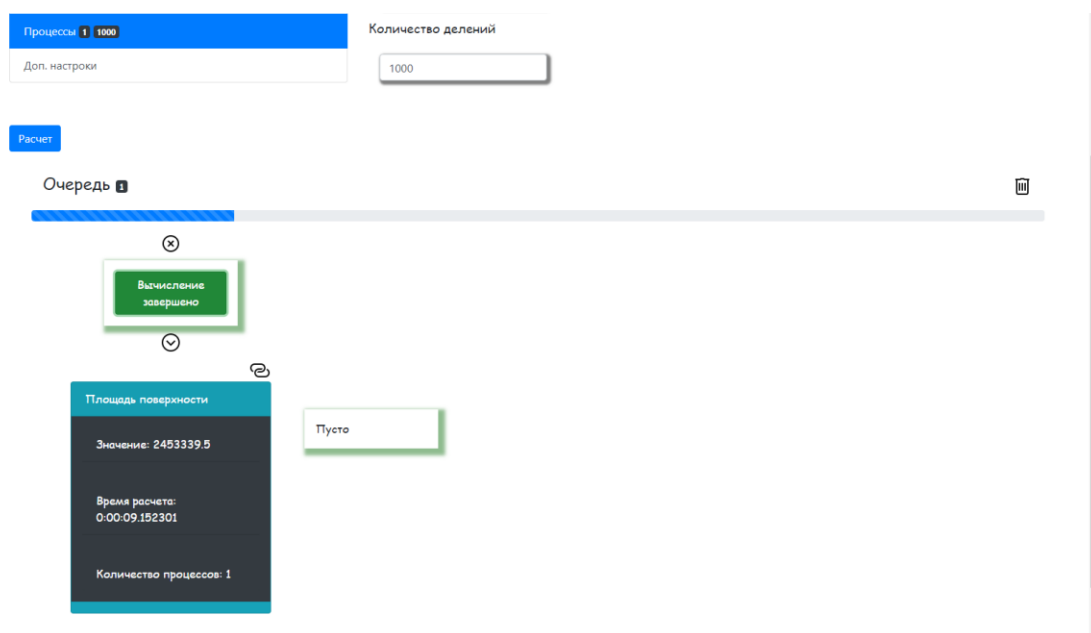


Рисунок 3.3 – Вывод результат вычисления

В качестве дополнительного функционала была реализована возможность закреплять найденное решение. Это позволит всегда видеть результат, который был закреплен, и сравнивать с другими результатами. Одновременно можно закрепить лишь один результат. Закрепленный результат можно удалить в любое время. Пример показан на рисунке 3.4.

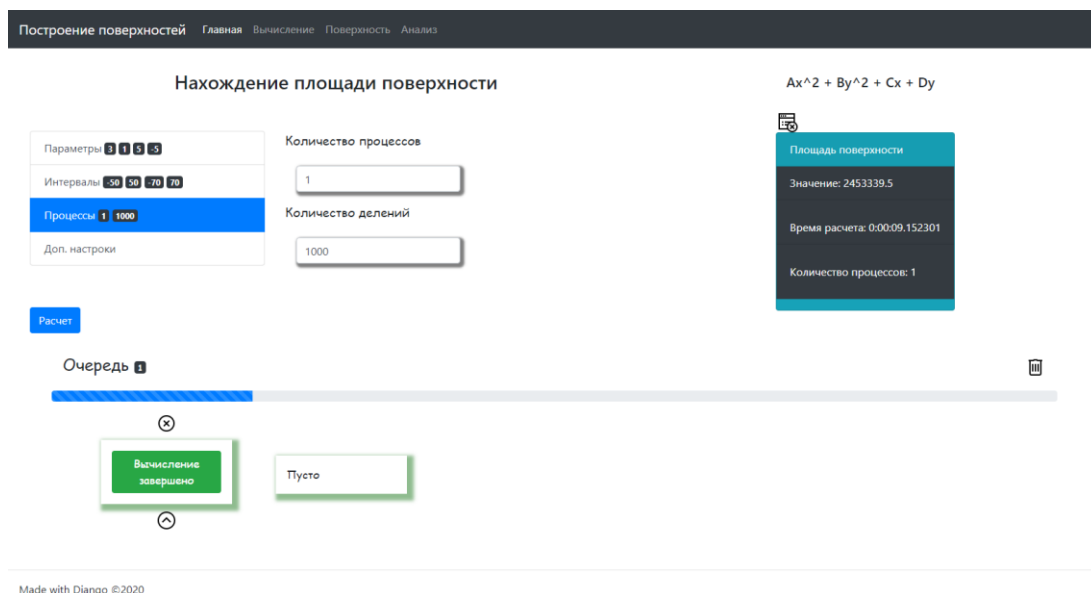


Рисунок 3.4 – Пример закрепления результата

На странице Поверхность можно построить поверхность в Python. Построенная поверхность также будет отображаться на самой странице, в виде изображения. Страница показана на рисунке 3.5.

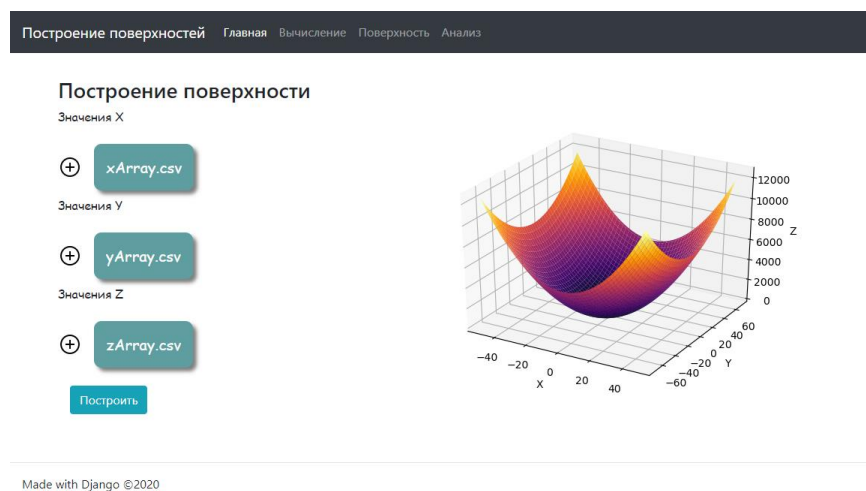


Рисунок 3.5 – Страница построения поверхности

После построения поверхность отображается в 3D-пространстве, а после – в виде рисунка на странице. Для выбора файла используется диалоговое окно выбора. Диалоговое окно показано на рисунке 3.6.

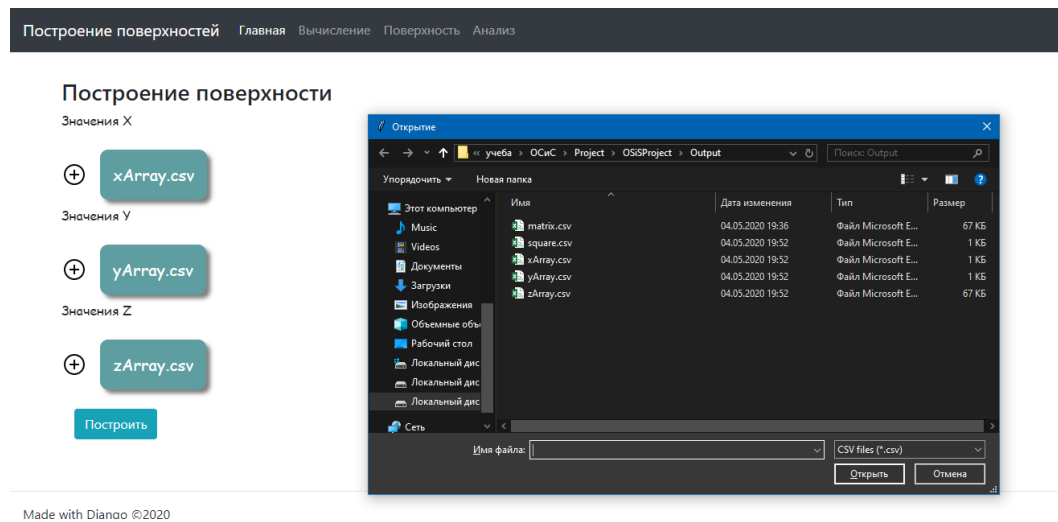


Рисунок 3.6. – Диалоговое окно выбора файла

Поверхность показана на рисунке 3.7.

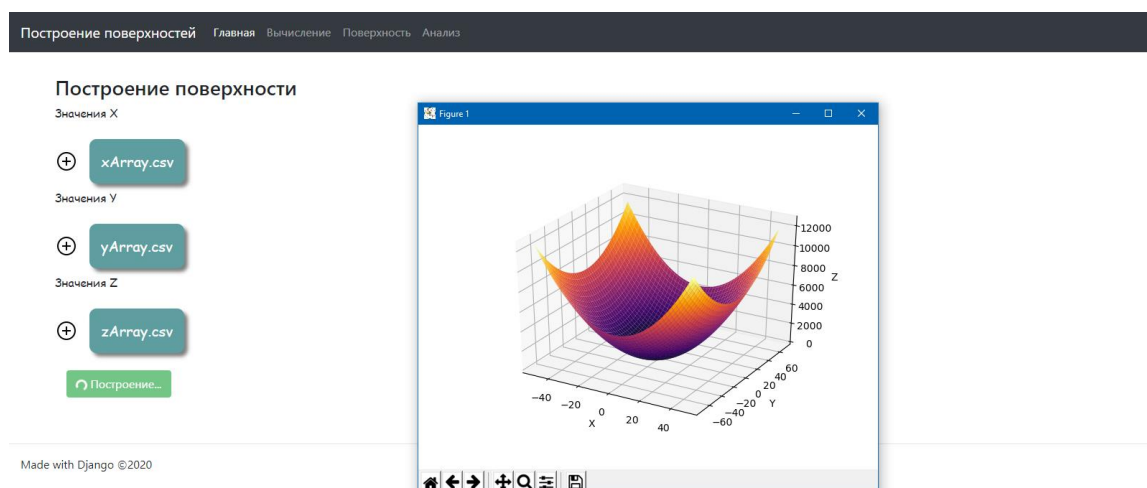


Рисунок 3.7 – Поверхность

На странице Анализ можно провести анализ зависимости время выполнения от количества запущенных процессов для вычисления. После выполнение анализа отображается результат при разных количествах процессов. Так при указании 5 процессов отобразится 5 результатов: начиная от 1 процесса заканчивая 5. Страница показана на рисунке 3.8.

Анализ данных

Параметры **3 1 5 5**

Интервалы **-50 50 -70 70**

Процессы **5 1000**

Количество процессов

Количество точек

Анализ

Площадь поверхности	Площадь поверхности	Площадь поверхности	Площадь поверхности	Площадь поверхности
Значение: 2452842.0	Значение: 2453052.375	Значение: 2453044.3125	Значение: 2307985.25	Значение: 2373699.1875
Время расчета: 0:00:38.372782	Время расчета: 0:00:35.107447	Время расчета: 0:00:33.283962	Время расчета: 0:00:38.095508	Время расчета: 0:00:45.162672
Количество процессов: 1	Количество процессов: 2	Количество процессов: 3	Количество процессов: 4	Количество процессов: 5

Made with Django ©2020

Рисунок 3.8 – Страница анализа

После проведения анализа отображается диаграмма зависимости время выполнения от количества процессов. Диаграмма строится с использованием библиотеки Python matplotlib. Диаграмма показана на рисунке 3.9.

Анализ данных

Параметры **3 1 5 5**

Интервалы **-50 50 -70 70**

Процессы **5 1000**

Количество процессов

Количество точек

Выполнение...

Made with Django ©2020

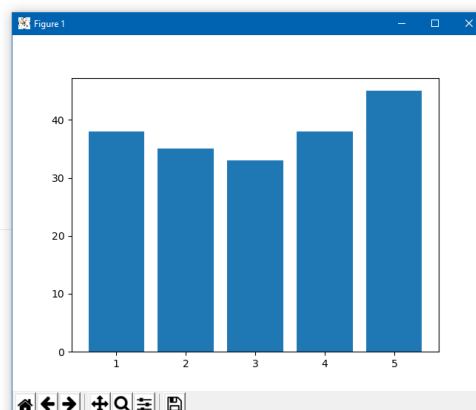


Рисунок 3.9 – Диаграмма

4 ТЕСТИРОВАНИЕ И ВЕРИФИКАЦИЯ

4.1 Тестирование

Для тестирования графического интерфейса необходимо составить чек-лист. В таблице 4.1 приведен список тестов по проверке графического интерфейса.

Таблица 4.1 – Чек-лист тестирования графического интерфейса

Номер теста	Описание	Ожидаемый результат
1.1	Увеличить браузер на весь экран	Страница отобразилась в увеличенном размере
1.2	Уменьшить браузер до минимального допустимого размера	Страница отобразилась в уменьшенном виде
1.3	Поле с указанием количество процессов оставить пустым	Вычисление площади не возможно
1.4	Ввести отрицательное число для поле ввода количества деления	Вычисление площади не возможно
1.5	Указать начальное значение для X больше конечного	Вычисление площади возможно
1.6	Указать конечное значение для X меньше начального	Построение поверхности не возможно
1.7	Не указывать файл с данными при построении поверхности	Построение поверхности не возможно
1.8	Поле для ввода имени файла оставить пустым	Будет использоваться название по умолчанию
1.9	Оставить поля пустыми при указании параметров	Будут использоваться значения по умолчанию

Проверим масштабируемость приложения, уменьшив окно браузера до минимального допустимого размера. На рисунке 4.1 изображен результат выполнения теста.

Построение поверхностей

Нахождение площади поверхности

$Ax^2 + By^2 + Cx + Dy$

Параметры 3 1 5 -5

Интервалы -50 50 -70 70

Процессы 1 1000

Доп. настройки

Значение A

3

Значение B

1

Значение C

5

Значение D

-5

Расчет

Очередь

Пусто

Рисунок 4.1 – Результат выполнения теста 1.1

Теперь проверим масштабируемость приложения увеличив окно браузера до максимального размера. На рисунке 4.2 изображен результат выполнения теста.

Построение поверхностей
Главная
Вычисление
Поверхность
Анализ

Нахождение площади поверхности

$Ax^2 + By^2 + Cx + Dy$

Параметры
1
1
5
-5

Интервалы
-50
50
170
10

Процессы
1
1000

Доп. настройки

Значение A

3

Значение B

1

Значение C

5

Значение D

-5

Расчет

Очередь

Пусто

Рисунок 4.2 – Результат выполнения теста 1.2

Если поле для ввода количества процессов оставить пустым, то вычисление будет невозможно и появится предупреждение. На рисунке 4.3 показано, что кнопка вычисления заблокирована если поле оставить пустым.

Построение поверхностей
Главная
Вычисление
Поверхность
Анализ

Нахождение площади поверхности

$Ax^2 + By^2 + Cx + Dy$

Параметры
1
1
5
-5

Интервалы
-50
50
170
10

Процессы
1000

Доп. настройки

Количество процессов

Значение должно быть больше 0

Количество точек

1000

Расчет

Очередь

Пусто

Made with Django ©2020

Рисунок 4.3 – Результат выполнения теста 1.3

На рисунке 4.4 показано, что кнопка для вычисления заблокируется, если указать отрицательное значение для поля количество делений.

Построение поверхностей
Главная
Вычисление
Поверхность
Анализ

Нахождение площади поверхности
 $Ax^2 + By^2 + Cx + Dy$

Параметры 1 2 3 4 5

Интервалы -50 50 -70 70

Процессы 1 -10

Доп. настройки

Количество процессов
1

Количество точек
-10

Значение должно быть больше 0

Расчет

Очередь

Пусто

Made with Django ©2020

Рисунок 4.4 – Результат проверки теста 1.4

На рисунке 4.5 показано, что, если начальное значение для интервала по X указать больше, чем конечное, отобразится предупреждение. Вычисление при этом возможно.

Построение поверхностей
Главная
Вычисление
Поверхность
Анализ

Нахождение площади поверхности
 $Ax^2 + By^2 + Cx + Dy$

Параметры 1 2 3 4 5

Интервалы 60 50 -70 70

Процессы 1 1000

Доп. настройки

Начальное значение X
60

Должно быть меньше конечного

Конечное значение X
50

Начальное значение Y
-70

Конечное значение Y
70

Расчет

Очередь

Рисунок 4.5 – Результат теста 1.5

Если указать конечное значение меньше начального, отобразится предупреждение. При этом вычисление возможно. Результат выполнения теста 1.6 показано на рисунке 4.6.

Построение поверхностей
Главная
Вычисление
Поверхность
Анализ

Чтобы выйти из полноэкранного режима, нажмите F11

Нахождение площади поверхности

Параметры
3
1
5
-5

Интервалы
-50
-60
-70
70

Процессы
1
1000

Доп. настройки

Начальное значение X

-50

Конечное значение X

-60

Должно быть больше начального

Начальное значение Y

-70

Конечное значение Y

70

Расчет

Очередь

$Ax^2 + By^2 + Cx + Dy$

Рисунок 4.6 – Результат теста 1.6

На рисунке 4.7 показано, что при попытке построить поверхность, не указав хотя бы один из файлов с данными, кнопка построения заблокируется и появится предупреждение.

Построение поверхностей
Главная
Вычисление
Поверхность
Анализ

Построение поверхности

Значения X

+

Необходимо выбрать файл

Значения Y

+

yArray.csv

Значения Z

+

zArray.csv

Построить

Made with Django ©2020

Рисунок 4.7 – Результат выполнения теста 1.7

При расчете площади поверхности пользователь может сохранить результат данных в файлы. Если пользователь оставит поле, для указания названия файла, пустым, то будут использоваться название файлов по умолчанию и появится предупреждение. Результат теста показан на рисунке 4.8.

					Курсовая проект	Лист
						32
Изм.	Лист	№ докум.	Подпись	Дата		

Построение поверхностей

Главная

Вычисление

Поверхность

Анализ

Нахождение площади поверхности

$Ax^2 + By^2 + Cx + Dy$

Параметры: 3 1 5 5

Интервалы: -50 50 -70 70

Процессы: 1 1000

Доп. настройки

☒ Показать точное решение

☒ Сохранить результат в файл

☐ Сохранить как...

Сохранение данных X

Значение по умолчанию

Сохранение данных Y

Значение по умолчанию

Сохранение данных Z

Значение по умолчанию

Рисунок 4.8 – Результат проверки теста 1.8

Если не указывать никакие параметры (поля оставить пустыми), будут использоваться значения по умолчанию. Результат выполнения теста 1.9 показан на рисунках 4.9 и 4.10.

Построение поверхностей

Главная

Вычисление

Поверхность

Анализ

Нахождение площади поверхности

$Ax^2 + By^2 + Cx + Dy$

Параметры

Интервалы: -50 50 -70 70

Процессы: 1 1000

Доп. настройки

Значение A

Значение B

Значение C

Значение D

Расчет

Очередь

Пусто

Рисунок 4.9 – Не заполнение полей

Построение поверхностей
Главная
Вычисление
Поверхность
Анализ

Нахождение площади поверхности

$Ax^2 + By^2 + Cx + Dy$

Параметры
1
1
1
1
По умолчанию

Интервалы
-50
50
-70
70

Процессы
1
1000

Доп. настройки

Значение A

1

Значение B

1

Значение C

1

Значение D

1

Расчет

Очередь

Вычисление...

Пусто

Рисунок 4.10 – Результат теста 1.9

В таблице 4.2 приведен результат тестирования.

Таблица 4.2 – Результат выполнения тестов из чек-листа

Результат выполнения	Номер теста
Выполнено успешно	1.1
Выполнено успешно	1.2
Выполнено успешно	1.3
Выполнено успешно	1.4
Выполнено успешно	1.5
Выполнено успешно	1.6
Выполнено успешно	1.7
Выполнено успешно	1.8
Выполнено успешно	1.9

4.2 Верификация и анализ полученных данных

Приложение, написанное в результате выполнения курсового проекта, находит площадь поверхности, строит поверхность и проводит анализ.

Анализ позволил в виде диаграммы показать, что параллельное вычисление выполняет расчет более быстрее, чем в один поток. Однако это не всегда истинно: распараллеливание не всегда показывает лучший результат. Если вычисление является простым, то параллельное вычисление может только замедлить работу. Поэтому необходимо сначала определить: нужно ли параллельное вычисление.

Для проверки найденной площади используется библиотека Python *scipy*. Данная библиотека позволяет вычислять интегралы, в том числе двойные интегралы. Результатом вычисления двойного интеграла является найденная площадь и погрешность вычисления. Сравнение найденной площади с помощью приложения, и площади, найденной при помощи библиотеки *scipy* показано на рисунке 4.6.

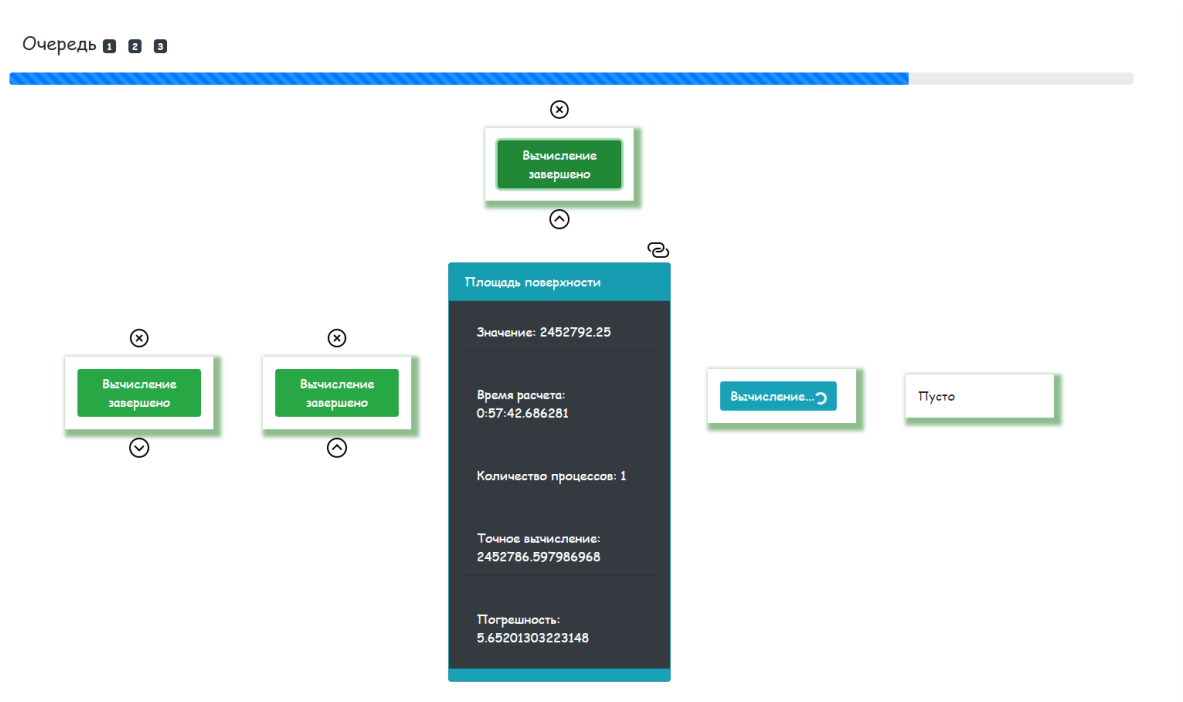


Рисунок 4.6 – Сравнение вычисления с точным вычислением

Для сравнение точного решения средствами Python был использован математический пакет MathCad. Результат вычисление с использованием математического пакета показан на рисунке 4.7.

$$\begin{aligned}
 a &:= 3 \\
 b &:= 1 \\
 c &:= 5 \\
 d &:= -5 \\
 \\
 func(x, y) &:= a \cdot x^2 + b \cdot y^2 + c \cdot x + d \cdot y \\
 \\
 res &:= \int_{-70}^{70} \int_{-50}^{50} \sqrt{1 + \left(\frac{d}{dy} func(x, y) \right)^2 + \left(\frac{d}{dx} func(x, y) \right)^2} dx dy = 2.453 \cdot 10^6
 \end{aligned}$$

Рисунок 4.7 – Нахождение площади с использованием MathCad

Как видно на рисунке 4.6 и рисунке 4.7 результат точного вычисления площади почти одинаковы.

При необходимости отображение точного вычисления и погрешности можно скрыть. Вычисление площади без указания точного вычисления показано на рисунке 4.8.

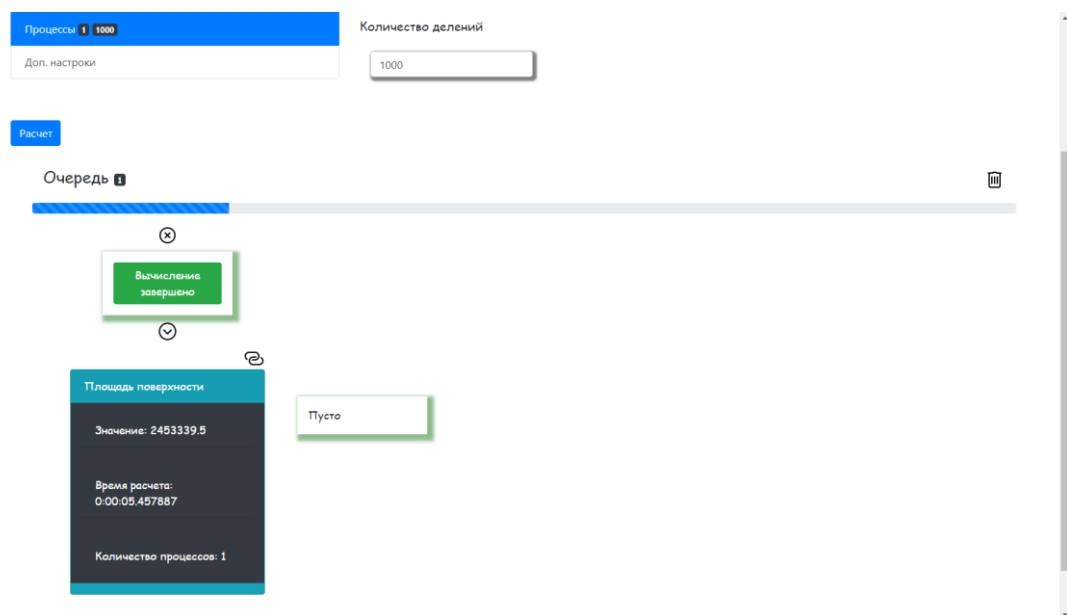


Рисунок 4.8 – Вывод результат без точного решения

ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта были решены все поставленные цели. Было реализовано приложения для нахождения площади поверхности и построение площади. Были получены навыки по работе с графикой в языке программирования Python

Также были изучены принципы параллельного вычисления на Python и получены навыки разработки с использованием фреймворка Django.

Для приложения были созданы независимые библиотеки.

					Курсовая проект	Лист
						37
Изм.	Лист	№ докум.	Подпись	Дата		

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Калиткин, Н.Н. Численные методы / Н.Н. Калиткин. – СПб.:BHV, 2014. – 202 с.
2. Холл, Дж. Современные численные методы решения интегралов/ Дж. Холл, Дж. Уамл. — М.: Мир, 2007. – 312 с.
3. Кудрявцев, Е. М. Справочник по Mathcad / Е. М. Кудрявцев. – М.:ДМК Пресс. – 2005. – 111 с.
4. Бизли. Python. Подробный справочник / В.И.Юров. – Спб: Питер, 2014.– 624 с.
5. Тестирование ПО [Электронный ресурс] — Режим доступа: <https://doitsmartly.ru/all-articles/sw-testing.html> — Дата доступа: 25.11.2019
6. Multiprocessing [Электронный ресурс] — Режим доступа: <https://docs.python.org/2/library/multiprocessing.html#managers>— Дата доступа: 25.11.2019