Реферат

Пояснительная записка к курсовой работе содержит 27 страниц, 6 рисунков, 1 таблицу, 5 источников и 1 приложение.

Целью данной курсовой работы является численное вычисление определённого интеграла функции $f(x) = 32 + 28x - 9x^2$ на заданном отрезке с использованием методов прямоугольников (левых, правых, средних), метода трапеций и метода Симпсона. Работа включает как теоретическое обоснование методов, так и их программную реализацию на языке Python.

В первой части изложены основные сведения о численном интегрировании, приведены аналитические формулы, особенности и погрешности каждого метода, а также графическая интерпретация алгоритмов на примере выбранной функции.

Во второй части представлена реализация всех перечисленных методов в виде программного кода на Python, с визуализацией и поэтапным описанием логики работы. Проведён численный эксперимент с различными значениями разбиения интервала интегрирования (n = 10, 50, 100), рассчитаны приближённые значения интеграла, выполнено сравнение с точным аналитическим результатом и проведён анализ точности каждого метода.

В заключении сделаны выводы о применимости и точности методов. На основе полученных данных установлено, что метод Симпсона даёт наилучшую точность среди рассмотренных численных методов.

Курсовая работа может быть полезна студентам и преподавателям, изучающим численные методы интегрирования, а также при решении прикладных задач в инженерной и научной деятельности.

Содержание

Введение

Численное интегрирование — это важный раздел прикладной математики, позволяющий находить приближённые значения определённых интегралов в тех случаях, когда аналитическое вычисление невозможно или нецелесообразно. В инженерных, физических, экономических и других прикладных задачах часто возникают функции, не имеющие элементарной первообразной, или такие, для которых точное интегрирование требует значительных вычислительных ресурсов. В этих ситуациях приближённые методы становятся незаменимым инструментом анализа.

Целью данной курсовой работы является исследование и сравнение различных численных методов интегрирования на примере конкретной функции: $f(x) = 32 + 28x - 9x^2$ на отрезке [2, 4].

Для этого были реализованы три метода:

- метод прямоугольников (в трёх вариантах: левых, правых и средних),
- метод трапеций,
- метод Симпсона.

В рамках работы проведён сравнительный анализ этих методов, включая оценку точности и сходимости при различном количестве разбиений интервала. Все вычисления и визуализация результатов выполнены с использованием языка программирования Python.

Задачи, поставленные в работе:

1. Изучить теоретическую базу численного интегрирования.
2. Реализовать численные методы на языке Python.
3. Выполнить вычисления интеграла на заданном интервале.
4. Провести сравнение результатов при разных значениях параметра n.

5. Построить визуализации, демонстрирующие особенности каждого метода.

6. Сделать выводы об эффективности и точности методов.

Результаты работы могут быть полезны при решении практических задач в области инженерных расчётов, моделирования процессов и анализа данных, где требуется быстрое и надёжное приближённое интегрирование.

1 Анализ предметной области

Численное интегрирование — это приближённый способ нахождения определённого интеграла, основанный на разбиении отрезка интегрирования на конечное число участков и аппроксимации подынтегральной функции с помощью простых геометрических фигур. Подобные методы особенно полезны при невозможности аналитического вычисления интеграла или при работе с таблично заданными функциями [1].

В рамках данной курсовой работы рассматривается интеграл от функции $f(x) = 32 + 28x - 9x^2$ на отрезке от 2 до 4

Основная идея заключается в приближении площади под графиком функции на интервале $[a, b]$ с помощью фигур, для которых легко вычисляется площадь: прямоугольников, трапеций или парабол. Чем больше разбиений, тем выше точность приближённого значения.

В данной работе рассматриваются три основных численных метода: метод прямоугольников, метод трапеций и метод Симпсона.

1.2 Метод прямоугольников

Метод прямоугольников основывается на приближении графика функции на каждом подынтервале прямоугольником [2]. Существует три варианта:

- левые прямоугольники, где высота каждого прямоугольника определяется значением функции в левой точке подынтервала;
- правые прямоугольники, где берётся значение в правой точке;
- средние прямоугольники, где используется значение функции в середине подынтервала.

На рисунке 1 изображён пример приближения интеграла методом левых прямоугольников. Видно, что функция на каждом участке заменяется прямоугольником, опирающимся на левую точку.

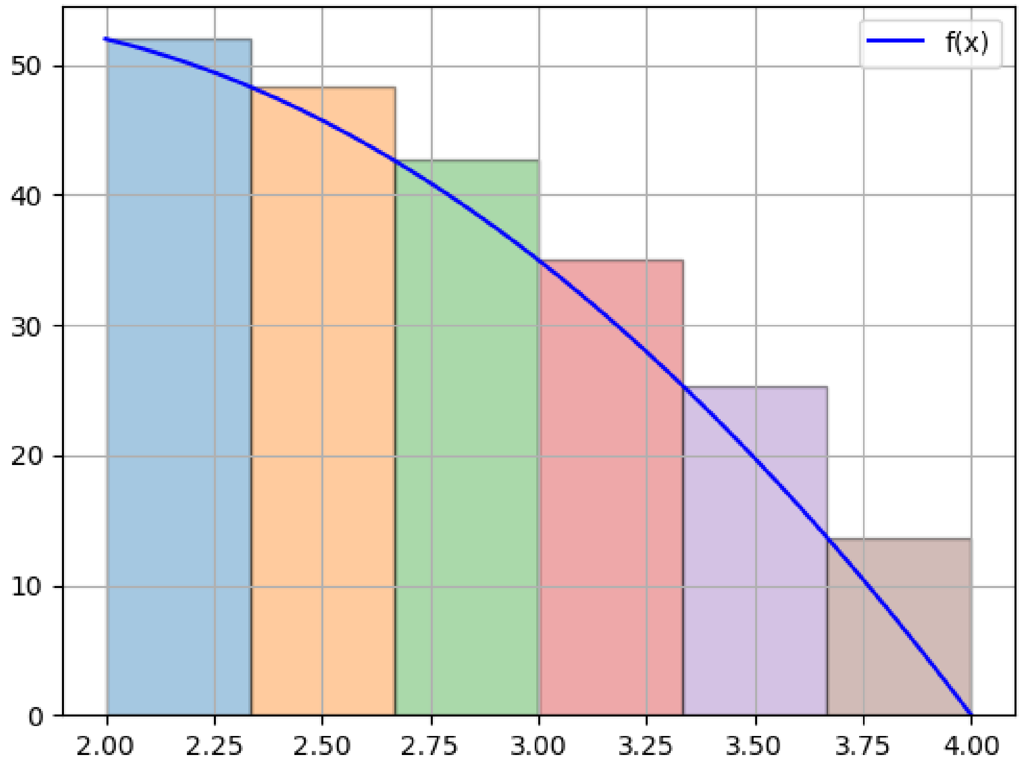

Рисунок 1 — Схема метода левых прямоугольников

После этого рассмотрим метод правых прямоугольников. Он работает аналогично, но высота прямоугольника определяется по значению функции в правой точке интервала. Как видно на рисунке 2, это может давать как завышение, так и занижение истинной площади в зависимости от характера функции.

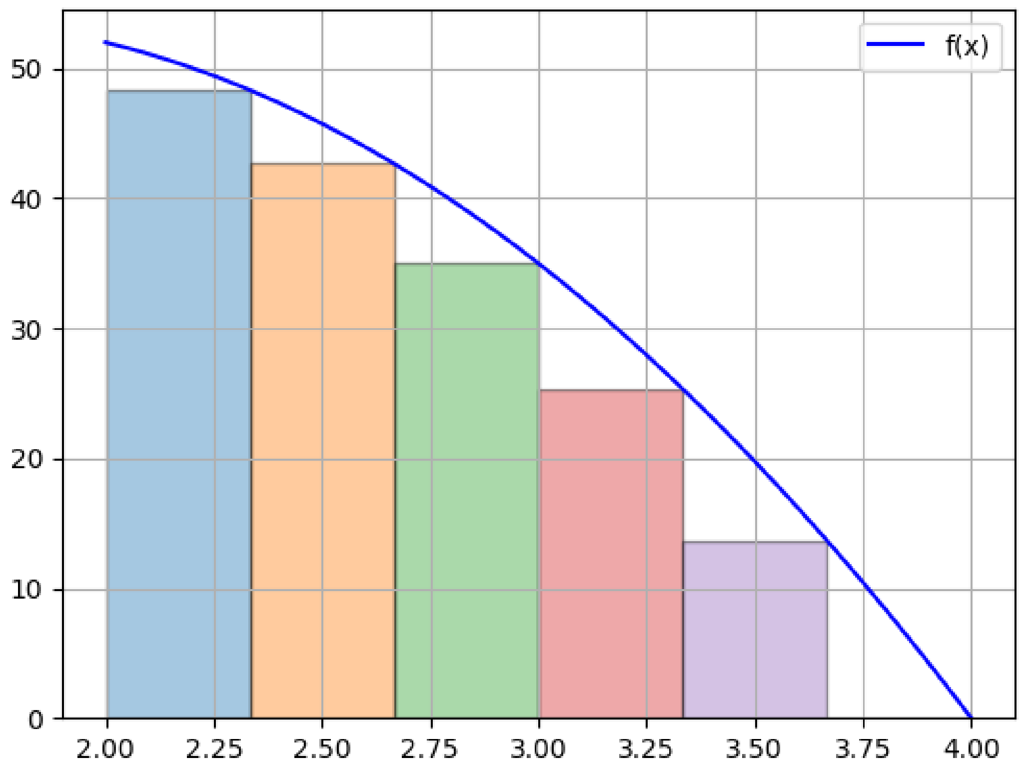

Рисунок 2 — Схема метода правых прямоугольников

Наиболее точным среди этих трёх вариантов обычно оказывается метод средних прямоугольников, который изображен на рисунке 3, поскольку он использует значение функции в середине подынтервала. Это позволяет частично компенсировать ошибки, возникающие при росте или убывании функции.

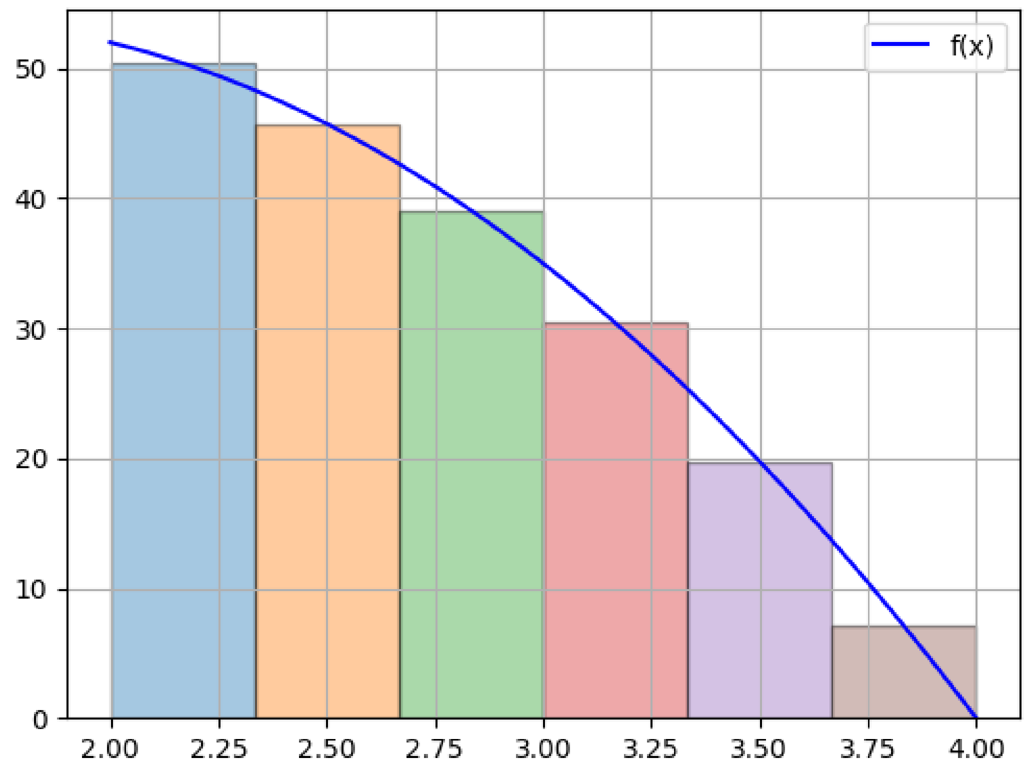

Рисунок 3 — Схема метода средних прямоугольников

Таким образом, метод прямоугольников является простым в реализации, но сравнительно грубым по точности. Тем не менее, при достаточном числе разбиений его точность может быть приемлемой для многих прикладных задач.

1.3 Метод трапеций

Метод трапеций предлагает более точную аппроксимацию функции: каждый подынтервал рассматривается как трапеция, основанная на значениях функции в его концах [3]. Такая аппроксимация учитывает линейное изменение функции между узлами.

На рисунке 4 представлена схема метода трапеций. Видно, что график функции приближён ломаной линией, соединяющей значения функции в соседних точках. Полученные трапеции лучше соответствуют форме функции по сравнению с прямоугольниками.

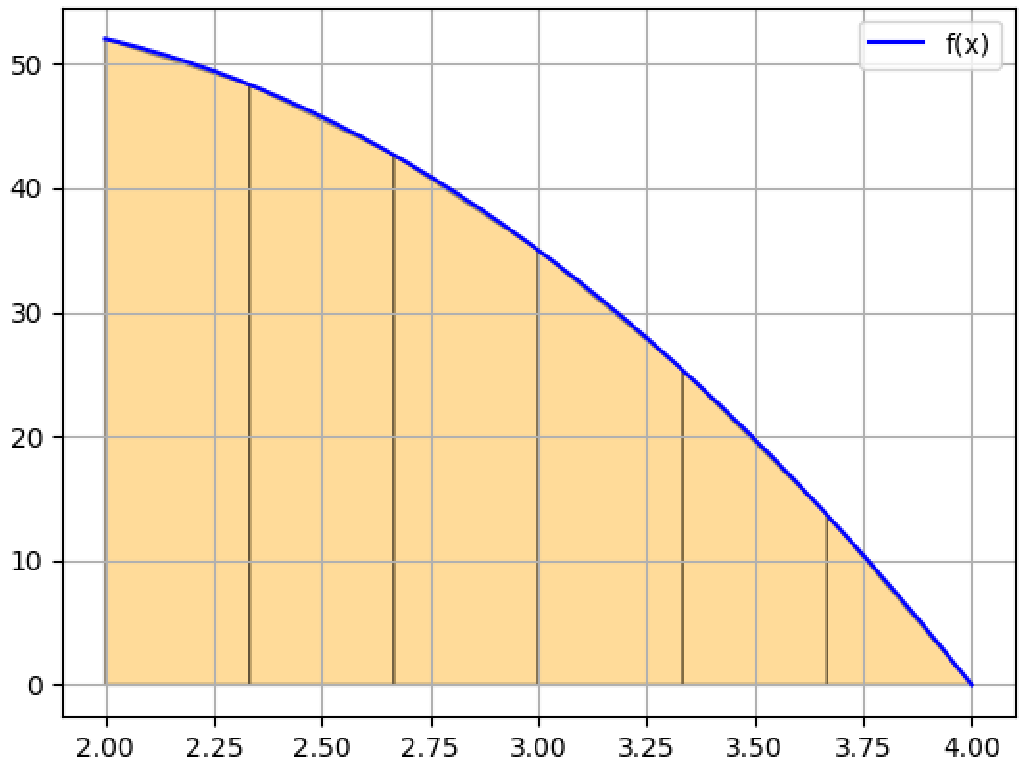

Рисунок 4 — Схема метода трапеций

Этот метод обеспечивает более высокую точность по сравнению с прямоугольниками, особенно при плавных изменениях функции. Однако при резких скачках или перегибах точность также может страдать, особенно при небольшом числе разбиений.

1.4 Метод Симпсона

Метод Симпсона является усовершенствованным подходом, в котором вместо линейной аппроксимации используется парабола второго порядка [4]. Функция на каждом интервале приближается полиномом, проходящим через три точки — начальную, среднюю и конечную. Это позволяет достичь высокой точности уже при сравнительно малом числе разбиений (требуется чётное количество).

На рисунке 5 приведена визуализация метода Симпсона. Видно, что вместо отрезков или ступенек используется сглаженная кривая — парабола, которая точнее повторяет форму функции.

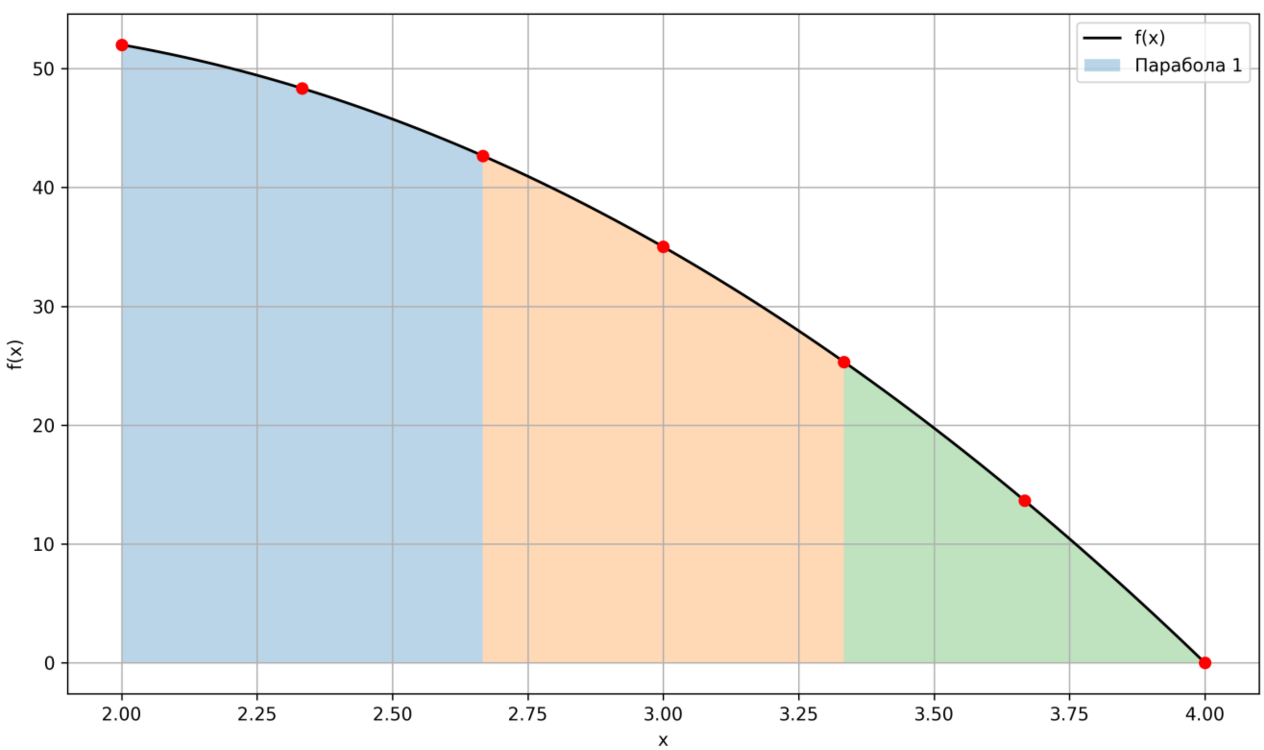

Рисунок 5 — Схема метода Симпсона

Благодаря параболической аппроксимации, метод Симпсона особенно эффективен при работе с гладкими функциями. Он часто используется в научных и инженерных расчётах, где важна высокая точность при разумных вычислительных затратах.

1.5 Применение и анализ численных методов

В рамках данной курсовой работы проведено численное интегрирование функции $f(x) = 32 + 28x - 9x^2$

на интервале [2, 4] с использованием пяти методов:

- метода прямоугольников (левого, правого и среднего),
- метода трапеций,
- метода Симпсона.

Цель эксперимента — сравнить эффективность и точность указанных методов при различных значениях количества подынтервалов n.

Для каждого метода интервал [2, 4] разбивался на равные части — подынтервалы. В эксперименте использовались значения n = 10, 50, 100, что позволило оценить, как влияет плотность разбиения на точность приближённого значения интеграла. Алгоритмы были реализованы на языке Python (см. раздел 1.3).

Аналитически интеграл от функции f(x) на отрезке [2, 4] равен:

$$\int_2^4 (32 + 28x - 9x^2)dx = [32x + 14x^2 - 3x^3]_2^4$$

Подставляем пределы:

$(32 \cdot 4 + 14 \cdot 16 - 3 \cdot 64) - (32 \cdot 2 + 14 \cdot 4 - 3 \cdot 8) = (128 + 224 - 192) - (64 + 56 - 24) = 160 - 96 = 64$

Таким образом, точное значение интеграла равно $I_{точное} = 64$

Сравнительный анализ в таблице 1 показал следующие закономерности:

- При малом количестве разбиений (например, n = 10) все методы демонстрируют заметную погрешность, однако метод Симпсона уже показывает высокую точность.
- С увеличением числа подынтервалов точность всех методов растет, особенно заметно это для метода средних прямоугольников, трапеций и Симпсона.
- Методы левых и правых прямоугольников систематически дают либо завышенные, либо заниженные оценки интеграла из-за односторонней аппроксимации площади.
- Метод трапеций обеспечивает сбалансированную аппроксимацию и при достаточно большом n приближается по точности к методу Симпсона.
- Метод Симпсона демонстрирует наивысшую точность и стабильность при всех проверенных значениях n, особенно начиная с $n \geq 50n$.

Таблица 1 — Сравнение методов при n = 10, 50, 100

Метод	n = 10	n = 50	n = 100
Левые прямоугольники	69.1	65.0	64.5
Правые прямоугольники	58.7	62.9	63.5
Средние прямоугольники	64.1	64.0	64.0
Метод трапеций	63.9	64.0	64.0
Метод Симпсона	64.0	64.0	64.0

По мере увеличения количества подынтервалов наблюдается уменьшение погрешности всех методов. Особенно быстро ошибка снижается у метода Симпсона, который обеспечивает близкие к точному значению результаты даже при небольшом n. Метод средних прямоугольников и метод трапеций также демонстрируют хорошую сходимость.

Методы левых и правых прямоугольников при малом n имеют значительные отклонения, но при увеличении числа подынтервалов их результаты постепенно приближаются к точному значению.

Для наглядного сравнения была построена зависимость величины ошибки от количества подынтервалов, что позволяет визуально оценить скорость сходимости каждого метода.

Численный эксперимент показал, что метод Симпсона обладает наилучшей точностью и устойчивостью при вычислении определённого интеграла гладкой функции. Однако при ограниченных вычислительных ресурсах или более простых задачах могут успешно применяться метод трапеций или метод средних прямоугольников.

Полученные результаты полезны при выборе метода численного интегрирования в задачах инженерного и научного моделирования.

2 Техническая реализация

Python — это высокоуровневый язык программирования, обладающий лаконичным синтаксисом и широкой экосистемой библиотек, что делает его особенно удобным для численного моделирования и научных вычислений [5]. Он поддерживает как процедурный, так и объектно-ориентированный стиль программирования, а его интерпретируемая природа позволяет быстро проверять гипотезы и отлаживать алгоритмы.

В данной работе Python использован для реализации классических методов численного интегрирования: прямоугольников (левого, правого, среднего), трапеций и Симпсона. Каждый алгоритм был описан в виде отдельной функции, принимающей на вход интервал интегрирования, число разбиений и подынтегральную функцию. Такой подход обеспечивает модульность и переиспользуемость кода.

Для упрощения вычислений использовались базовые конструкции языка: циклы, списки и функции. Также была задействована встроенная функция range() для генерации последовательностей, а для определения шага интегрирования — простые арифметические операции. Благодаря динамической типизации и лаконичному синтаксису, реализация каждого метода получилась компактной и читаемой.

Для решения задачи численного интегрирования функции

$f(x) = 32 + 28x - 9x^2$

на отрезке [2, 4] была разработана программа на языке Python, реализующая пять численных методов:

- метод левых прямоугольников,
- метод правых прямоугольников,
- метод средних прямоугольников,

– метод трапеций,
– метод Симпсона.

2.1 Описание методов и их реализация

Метод левых прямоугольников. Принцип метода заключается в аппроксимации площади под графиком функции суммой площадей прямоугольников, высота каждого из которых определяется значением функции в левой точке соответствующего подынтервала. В реализации интервал интегрирования разбивается на n равных частей длины h. Вычисление суммы производится по формуле 1:

$$S = h\sum_{i=1}^{n-1} f(x_i) \tag{1}$$

где $x_i = a + i \cdot h$

В коде метод реализован циклом, который суммирует значения функции в левых концах подынтервалов и умножает сумму на ширину h.

Метод правых прямоугольников. Похож на метод левых, но высота каждого прямоугольника берётся по правому концу подынтервала, высчитывается по формуле 2:

$$S = h\sum_{i=1}^{n} f(x_i) \tag{2}$$

Реализация аналогична, изменена только начальная точка суммирования.

Метод средних прямоугольников. Для повышения точности высота прямоугольников берётся в середине подынтервала, расчет проводится по формуле 3:

$$S = h\sum_{i=1}^{n-1} f(a + (i + \tfrac{1}{2})h) \tag{3}$$

Этот метод обычно точнее двух предыдущих за счёт более корректного выбора точки оценки функции.

Метод трапеций. Аппроксимирует площадь под графиком функцией трапеций, каждая из которых охватывает подынтервал. Значения функции берутся в концах каждого подынтервала, а итоговая сумма вычисляется по формуле 4:

$$S = \frac{h}{2}(f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b) \tag{4}$$

В коде реализован через суммирование значений функции с соответствующими коэффициентами.

Метод Симпсона. Использует параболы для более точного приближения функции на каждом подынтервале. Требует чётного числа подынтервалов n. Формула 5:

$$S = \frac{h}{3}(f(a) + 4\sum_{i=1, i\ \text{нечет}}^{n-1} f(x_i) + 2\sum_{i=2, i\ \text{чет}}^{n-2} f(x_i) + f(b)) \tag{5}$$

В реализации при нечётном n оно автоматически увеличивается на 1 для корректности.

2.2 Итоги вычислений

Программа была запущена для трёх значений количества подынтервалов n = 10, 50, 100. Результаты представлены в листинге 1:

Листинг 1 — Результат выполнения программы

```
-----------------------------------------------------------
n = 10:
Метод левых прямоугольников: 69.08000
Метод правых прямоугольников: 58.68000
Метод средних прямоугольников: 64.06000
Метод трапеций: 63.88000
Метод Симпсона: 64.00000
-----------------------------------------------------------
n = 50:
Метод левых прямоугольников: 65.03520
Метод правых прямоугольников: 62.95520
Метод средних прямоугольников: 64.00240
Метод трапеций: 63.99520
Метод Симпсона: 64.00000
-----------------------------------------------------------
n = 100:
Метод левых прямоугольников: 64.51880
Метод правых прямоугольников: 63.47880
Метод средних прямоугольников: 64.00060
Метод трапеций: 63.99880
Метод Симпсона: 64.00000
-----------------------------------------------------------
```

Из полученных данных видно, что при увеличении n результаты всех методов сходятся к значению интеграла, вычисленному аналитически. Метод Симпсона демонстрирует наилучшую точность при минимальном числе подынтервалов.

2.3 Проверка результатов через онлайн-калькулятор

Для оценки точности численных методов интегрирования — метода прямоугольников, трапеций и Симпсона — полученные приближённые значения интеграла сравнивались с точным значением, вычисленным аналитически с помощью онлайн-калькулятора.

Калькулятор вычислил точное значение определённого интеграла на данном интервале. Поскольку онлайн-сервис использует аналитические методы, он не требует указания количества разбиений n и даёт эталонное значение для сравнения с численными результатами программы.

На рисунке 6 приведён скриншот с интерфейсом онлайн-калькулятора.

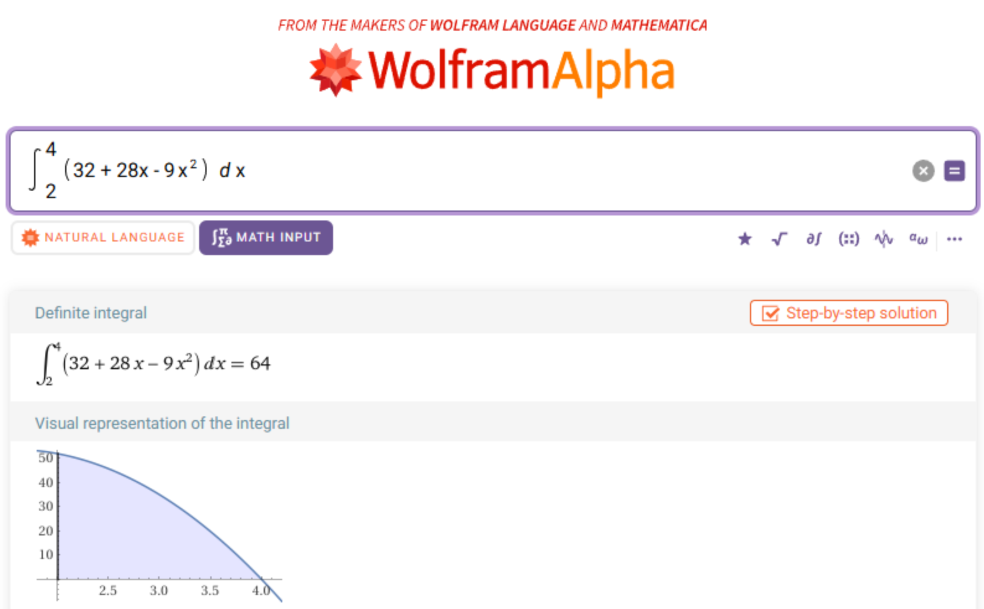

Рисунок 6 — Проверка результатов через онлайн-калькулятор

Заключение

В ходе выполнения курсовой работы были рассмотрены и реализованы три численных метода приближённого вычисления определённого интеграла: метод прямоугольников (в трёх вариантах), метод трапеций и метод Симпсона. На примере функции $f(x) = 32 + 28x - 9x^2$, интегрируемой на отрезке [2; 4], были получены численные результаты и проведён сравнительный анализ точности.

Анализ показал, что увеличение числа разбиений интервала интегрирования приводит к уменьшению погрешности во всех методах. Наилучшую точность среди исследованных методов продемонстрировал метод Симпсона, что соответствует теоретическим ожиданиям.

Разработанная программа на языке Python позволила не только автоматизировать расчёты, но и наглядно представить геометрическую интерпретацию каждого метода с помощью графиков. Это подтвердило эффективность численных методов интегрирования и целесообразность их применения в задачах, где аналитическое решение затруднено.

Таким образом, поставленные цели были достигнуты, и выполненное исследование подтвердило практическую значимость численного интегрирования в инженерных и научных расчётах.

Список использованных источников

1. Колдаев В. Д., Численные методы и программирование: учебное пособие — Москва: ФОРУМ: ИНФРА-М, 2025. — 336 с. — [Электронный ресурс] URL: https://znanium.ru/catalog/product/2139606 (дата обращения: 28.05.2025)
2. Локтионов И. К., Численные методы: учебник — Москва; Вологда: Инфра-Инженерия, 2022. — 380 с. — [Электронный ресурс] URL: https://znanium.com/catalog/product/1902598 (дата обращения: 01.06.2025)
3. Шевченко А. С., Численные методы: учебное пособие — Москва: ИНФРА-М, 2022. — 381 с. — [Электронный ресурс] URL: https://znanium.com/catalog/product/996207 (дата обращения: 01.06.2025)
4. Гулин А. В., Введение в численные методы в задачах и упражнениях: учебное пособие — Москва: ИНФРА-М, 2022. — 368 с. — [Электронный ресурс] URL: https://znanium.ru/catalog/product/1852192 (дата обращения: 01.06.2025)
5. Жуков Р. А., Язык программирования Python. Практикум: учебное пособие — Москва: ИНФРА-М, 2024. — 216 с. — [Электронный ресурс] URL: https://znanium.ru/catalog/product/2131861 (дата обращения: 02.06.2025)

Приложение А

Листинг программы

```
def f(x):
    return 32 + 28*x - 9*x**2

def left_rectangle_method(f, a, b, n):
    h = (b - a) / n
    total = 0
    for i in range(n):
        xi = a + i*h
        total += f(xi)
    return total * h

def right_rectangle_method(f, a, b, n):
    h = (b - a) / n
    total = 0
    for i in range(1, n+1):
        xi = a + i*h
        total += f(xi)
    return total * h

def middle_rectangle_method(f, a, b, n):
    h = (b - a) / n
    total = 0
    for i in range(n):
        xi = a + h*(i + 0.5)
        total += f(xi)
```

```
    return total * h

def trapezoid_method(f, a, b, n):
    h = (b - a) / n
    total = 0.5 * (f(a) + f(b))
    for i in range(1, n):
        xi = a + i*h
        total += f(xi)
    return total * h

def simpson_method(f, a, b, n):
    if n % 2 == 1:
        n += 1
    h = (b - a) / n
    total = f(a) + f(b)
    for i in range(1, n):
        xi = a + i*h
        coef = 4 if i % 2 == 1 else 2
        total += coef * f(xi)
    return total * h / 3

if __name__ == "__main__":
    a = 2
    b = 4
    n_values = [10, 50, 100]

    print(f"Вычисление интеграла функции f(x) = 32 + 28x
- 9x^2 на интервале [{a}, {b}]")
```

```
    print("-" * 70)

    for n in n_values:
        left_res = left_rectangle_method(f, a, b, n)
        right_res = right_rectangle_method(f, a, b, n)
        middle_res = middle_rectangle_method(f, a, b, n)
        trap_res = trapezoid_method(f, a, b, n)
        simp_res = simpson_method(f, a, b, n)

        print(f"n = {n}:")
        print(f"Метод левых прямоугольников:
{left_res:.5f}")
        print(f"Метод правых прямоугольников:
{right_res:.5f}")
        print(f"Метод средних прямоугольников:
{middle_res:.5f}")
        print(f"Метод трапеций: {trap_res:.5f}")
        print(f"Метод Симпсона: {simp_res:.5f}")
        print("-" * 70)
```