**Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования**

**Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика и системы управленияКафедра ИУ6 Компьютерные системы и сети Группа ИУ6-63Б

**ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА**

**Отчет по домашнему заданию N4: Численное интегрирование**

**Вариант 10**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент: |  |  | В.К. Залыгин |
| Преподаватель: | дата, подпись |  | Ф.И.О.  Я.Ю. Павловский |
|  | дата, подпись |  | Ф.И.О. |

*Москва, 2025*

**Задание**

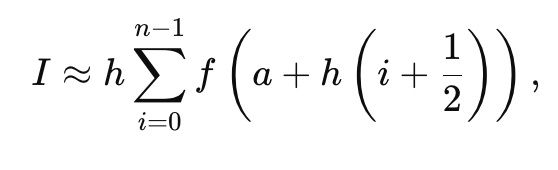
Цель домашней работы: изучения методов численного интегрирования, изучения метода Рунге.

Для достижения цели необходимо решить следующие задачи:

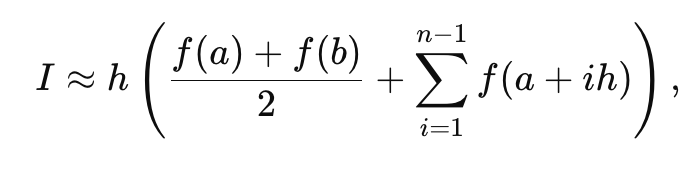
1. Найти по формуле Ньютона-Лейбница точное значение заданного определенного интеграла и внести полученное значение в отчет с 4-мя верными знаками после запятой.
2. Реализовать в среде MatLab или на языке Python вычисление данного определенного интеграла по формуле центральных прямоугольников и трапеции и найти его значения с выбором шага (вторая производная) для удолетворении заданной точности .
3. Реализовать в среде MatLab или на языке Python вычисление данного определенного интеграла по трапеции и Симпсона и найти его значения с автоматическим выбором шага по правилу Рунге для удовлетворения заданной точности.
4. Сравнить полученные значения с точным значением определенного интеграла, убедится в том, что результаты удовлетворяют заданной точности.

**Методы численного интегрирования**

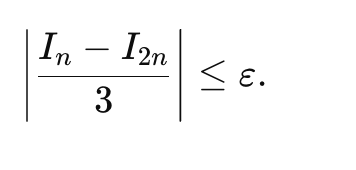
**Метод центральных прямоугольников**  
Приближает интеграл суммой площадей прямоугольников, высоты которых равны значению функции в середине каждого подотрезка.  
Формула:



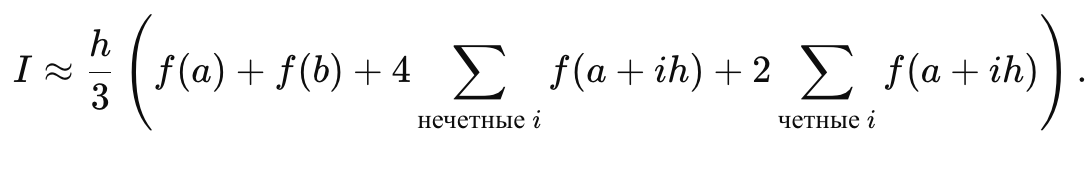
**Метод трапеций**  
Аппроксимирует функцию на каждом подотрезке отрезком прямой и вычисляет сумму площадей трапеций:

****

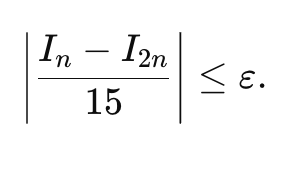
**Правило Рунге для метода трапеций**  
Позволяет автоматически подобрать число разбиений nnn для достижения заданной точности ε. Погрешность оценивается по разности интегралов на сетках с n и 2n узлами:

****

**Метод Симпсона (параболический метод)**  
Приближает интеграл суммой площадей парабол, проходящих через три последовательные точки. Требует чётное число разбиений nnn. Формула:

****

**Правило Рунге для метода Симпсона**  
Для метода Симпсона точность оценивается по правилу Рунге с другим коэффициентом:

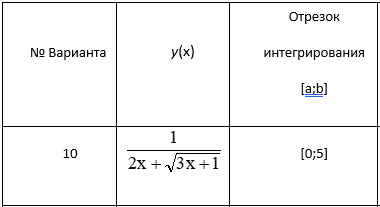
****

**Погрешность**

Для каждого метода дополнительно рассчитывается абсолютная ошибка как разность между численным и точным значениями интеграла.

Значения для индивидуального варианта указаны в таблице 1.

Таблица 1 – Значения индивидуального варианта



Решением интеграла является семейство функций .

Листинг 1 – Код программы main.py

|  |
| --- |
| from data\_input import get\_parameters, exact\_integral  from methods import (  central\_rectangles,  trapezoidal,  runge\_trapezoidal,  runge\_simpson,  )  def main():  a, b, eps = get\_parameters()  exact = exact\_integral()  print(f"Точное значение интеграла: {exact:.6f}\n")  # Центральные прямоугольники  n = 2  while True:  integral = central\_rectangles(a, b, n)  next\_integral = central\_rectangles(a, b, n \* 2)  if abs(integral - next\_integral) / 3 <= eps:  break  n \*= 2  print(f"Метод центральных прямоугольников: {next\_integral:.6f} при n = {n \* 2}")  # Метод трапеций с шагом по второй производной  n = 2  while True:  integral = trapezoidal(a, b, n)  next\_integral = trapezoidal(a, b, n \* 2)  if abs(integral - next\_integral) / 3 <= eps:  break  n \*= 2  print(f"Метод трапеций: {next\_integral:.6f} при n = {n \* 2}")  # Метод трапеций с Рунге  integral\_runge, n\_runge = runge\_trapezoidal(a, b, eps)  print(f"Метод трапеций с правилом Рунге: {integral\_runge:.6f} при n = {n\_runge}")  # Метод Симпсона с Рунге  integral\_simpson, n\_simpson = runge\_simpson(a, b, eps)  print(f"Метод Симпсона с правилом Рунге: {integral\_simpson:.6f} при n = {n\_simpson}")  print("\nСравнение:")  print(f"Абсолютная ошибка центральных прямоугольников: {abs(next\_integral - exact):.2e}")  print(f"Абсолютная ошибка трапеций: {abs(next\_integral - exact):.2e}")  print(f"Абсолютная ошибка трапеций с Рунге: {abs(integral\_runge - exact):.2e}")  print(f"Абсолютная ошибка Симпсона с Рунге: {abs(integral\_simpson - exact):.2e}")  if \_\_name\_\_ == "\_\_main\_\_":  main() |

Листинг 2 – Код программы methods.py

|  |
| --- |
| from data\_input import f  def central\_rectangles(a, b, n):  """ Метод центральных прямоугольников. """  h = (b - a) / n  result = 0  for i in range(n):  xi = a + h \* (i + 0.5)  result += f(xi)  return result \* h  def trapezoidal(a, b, n):  """ Метод трапеций. """  h = (b - a) / n  result = (f(a) + f(b)) / 2  for i in range(1, n):  xi = a + h \* i  result += f(xi)  return result \* h  def runge\_trapezoidal(a, b, eps):  """ Метод трапеций с автоматическим выбором шага по правилу Рунге. """  n = 2  I\_n = trapezoidal(a, b, n)  n \*= 2  I\_2n = trapezoidal(a, b, n)  while abs(I\_n - I\_2n) / 3 > eps:  n \*= 2  I\_n = I\_2n  I\_2n = trapezoidal(a, b, n)  return I\_2n, n  def simpson(a, b, n):  """ Метод Симпсона. """  if n % 2 != 0:  n += 1 # Симпсон требует чётное число отрезков  h = (b - a) / n  result = f(a) + f(b)  for i in range(1, n, 2):  result += 4 \* f(a + i \* h)  for i in range(2, n-1, 2):  result += 2 \* f(a + i \* h)  return result \* h / 3  def runge\_simpson(a, b, eps):  """ Метод Симпсона с автоматическим выбором шага по правилу Рунге. """  n = 2  I\_n = simpson(a, b, n)  n \*= 2  I\_2n = simpson(a, b, n)  while abs(I\_n - I\_2n) / 15 > eps:  n \*= 2  I\_n = I\_2n  I\_2n = simpson(a, b, n)  return I\_2n, n |

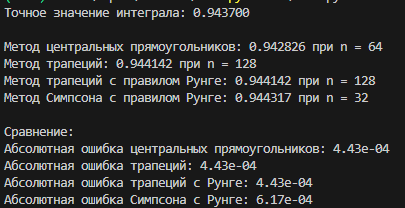
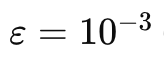


Рисунок 1 – Результат работы программы

**Вывод**

В ходе работы были изучены и реализованы численные методы вычисления определённых интегралов: метод центральных прямоугольников, метод трапеций, метод трапеций с использованием правила Рунге и метод Симпсона с правилом Рунге. Для интегрирования заданной функции на отрезке [0,5] с точностью  были получены численные результаты, а также рассчитаны абсолютные ошибки относительно точного значения интеграла. Анализ показал, что методы с использованием правила Рунге (особенно метод Симпсона) обеспечивают более высокую точность при меньшем числе разбиений по сравнению с базовыми методами. Метод центральных прямоугольников и метод трапеций сходятся медленнее, но тоже достигают требуемой точности при достаточно большом числе разбиений. Таким образом, использование адаптивных методов (с контролем погрешности) позволяет оптимизировать расчёты и получать высокоточные результаты при меньших вычислительных затратах.