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 управления»

КАФЕДРА «Теоретическая информатика и компьютерные технологии»

Лабораторная работа №2

«Приближенное вычисление определенного интеграла»

по курсу «Численные методы»

Студент: Шемякин В.А.

Группа: ИУ9-62Б

Преподаватель: Домрачева А.Б.

Москва 2025

1 Цель

Целью данной работы является реализация методом приближенного вычисления определенных интегралов:

1. Метод центральных прямоугольников

2. Метод трапеций

3. Метод Симпсона

2 Постановка задачи

Дано: интеграл

$$I = \int_a^b f(x)\,dx, \quad \text{где } f(x) \text{ непрерывна на } [a, b]$$

Найти: значение интеграла

$$I^* \approx I$$

При заданной точности $\varepsilon < 0.001$.
Индивидуальный вариант(№ 5): $f(x) = x\sqrt{x+1}, \quad a = 0, \quad b = 3$

$$\int_0^3 x\sqrt{x+1}\,dx = 7.73333333$$

3 Основные теоретические сведения

3.1 Метод центральных прямоугольников

Метод заключается в вычислении площади под графиком подынтегральной функции с помощью суммирования площадей прямоугольников, ширина которых определяется шагом разбиения (расстояние между узлами интегрирования), а высота – значением подынтегральной функции в узле интегрирования.

Пусть требуется определить значение интеграла функции $f(x)$ на отрезке $[a, b]$. Тогда отрезок разбивается на n равных отрезков длиной $h = \frac{b-a}{n}$. Получаем разбиение данного отрезка точками

$$x_{i-0.5} = a + (i - 0.5)h, \quad i = 1, \ldots, n.$$

Тогда приближённое значение интеграла на всем отрезке будет равно:

$$I^* = h \sum_{i=1}^{n} f\big(a + (i - 0.5)h\big).$$

Этот метод обладает достаточно высокой точностью при малом шаге разбиения, однако его погрешность уменьшается медленнее, чем у метода трапеций и метода Симпсона.

3.2 Метод трапеций

Метод заключается в вычислении площади под графиком подынтегральной функции с помощью суммирования площадей трапеций, высота которых определяется шагом разбиения (расстояние между узлами интегрирования), а высота – значением подынтегральной функции в узле интегрирования.

Пусть требуется определить значение интеграла функции $f(x)$ на отрезке $[a, b]$. Тогда отрезок разбивается на n равных отрезков длиной $h = \frac{b-a}{n}$. Получаем разбиение данного отрезка точками

$$x_i = a + i\,h, \quad i = 1, \ldots, n.$$

Тогда приближённое значение интеграла на всем отрезке будет равно:

$$I^* = h\left(\frac{f(a) + f(b)}{2} + \sum_{i=1}^{n-1} f(x_i)\right) = h\left(\frac{f(a) + f(b)}{2} + \sum_{i=1}^{n-1} f(a + i\,h)\right).$$

3.3 Метод Симпсона

Метод заключается в приближении функции на отрезке $[a, b]$ интерполяционным многочленом второй степени $P_2(x)$.

$$P_2(x) \;=\; f_{i-0.5} \;+\; \frac{f_i - f_{i-1}}{h}\big(x_i - x_{i-0.5}\big) \;+\; \frac{f_i \;-\; 2\,f_{i-0.5} \;+\; f_{i-1}}{\frac{h^2}{2}}\big(x - x_{i-0.5}\big)^2,$$

Тогда приближённое значение интеграла на всем отрезке будет равно:

$$I^* = \frac{h}{6}\big(f(a) + f(b) + 4\sum f(x_{i-0.5}) + 2\sum f(x_i)\big),$$

где $h = \frac{b-a}{n}$.

3.4 Уточнение значения интеграла по Ричардсону

$I \approx I^*_h + O(h^k)$, где k – порядок точности метода, I^*_h – приближённое значение интеграла, вычисленное с помощью метода с шагом h.

Для метода средних прямоугольников и метода трапеций $k = 2$.

Для метода Симпсона $k = 4$.

$O(h^k) \approx c\,h^k, \quad c$ – некоторая константа, h – шаг.

Считаем, что вычисления проводятся без вычислительной погрешности, можно записать строгое равенство $I = I^*_h + c\,h^k$ для шага h и аналогичное равенство для шага $\frac{h}{2}$:

$$I = I^*_{h/2} + c\left(\frac{h}{2}\right)^k.$$

Из этих равенств получаем уточнённое значение интеграла:

$$I = I^*_{h/2} + \frac{I^*_{h/2} - I^*_h}{2^k - 1}.$$

Где значение R – уточнение по Ричардсону:

$$R = \frac{I^*_{h/2} - I^*_h}{2^k - 1}$$

используется для компенсации методологической погрешности численных методов интегрирования.

Чтобы построить процедуру приближённого вычисления интеграла с заданной точностью ε, применяется правило Рунге:

$$|R| < \varepsilon.$$

4 Реализация

Listing 1: Реализация чтения

```
from math import sqrt

eps = 1e-3
a, b = 0, 3
I_exact = 116 / 15
def f(x):
    return x * sqrt(x + 1)

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

def trapezoid(f, n):
    h = (b - a) / n
    return h / 2 * (f(a) + f(b) + 2 * sum(f(a + i * h) for i in range(1, n)))

def simpson(f, n):
    h = (b - a) / n
    return h / 3 * (f(a) + f(b) +
                    4 * sum(f(a + h * i) for i in range(1, n, 2)) +
                    2 * sum(f(a + h * i) for i in range(2, n, 2)))

def richardson_error(I1, I2, k):
    return (I2 - I1) / (2 ** k - 1)

def found_n_for_method(method):
    n = 2
    Ih = 0
    while abs(Ih - I_exact) > eps:
        n *= 2
        Ih = method(f, n)
    return n

def main():

```

```python
n_simpson   = found_n_for_method(simpson)
n_trapezoid = found_n_for_method(trapezoid)
n_rectangle = found_n_for_method(rectangle)

S1 = simpson(f, n_simpson)
S2 = simpson(f, 2 * n_simpson)
R_simpson = richardson_error(S1, S2, 4)

T1 = trapezoid(f, n_trapezoid)
T2 = trapezoid(f, 2 * n_trapezoid)
R_trapezoid = richardson_error(T1, T2, 2)

R1 = rectangle(f, n_rectangle)
R2 = rectangle(f, 2 * n_rectangle)
R_rectangle = richardson_error(R1, R2, 2)

print(f"eps = {eps}, I = {I_exact}")

headers = ["", "Симпсон\t", "Трапеции\t", "Прямоугольники\t"]

col_width = 20
header_row = "".join(f"{h:<{col_width}}" for h in headers)
print(header_row)
print("-" * col_width * len(headers))

rows = []

rows.append((
    "n",
    f"{n_simpson:<{col_width}}\t",
    f"{n_trapezoid:<{col_width}}\t",
    f"{n_rectangle:<{col_width}}\t"
))

rows.append((
    "I*",
    f"{S2:<{col_width}}\t",
    f"{T2:<{col_width}}\t",
    f"{R2:<{col_width}}\t"
))

rows.append((
    "R",
    f"{R_simpson:<{col_width}}\t",
    f"{R_trapezoid:<{col_width}}\t",
    f"{R_rectangle:<{col_width}}\t"
```

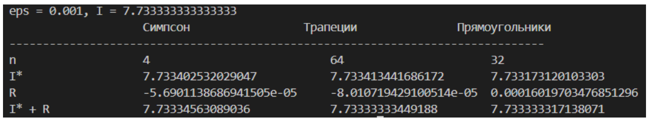
```
    ))

    rows.append((
        "I* + R",
        f"{S2 + R_simpson:<{col_width}}\t",
        f"{T2 + R_trapezoid:<{col_width}}\t",
        f"{R2 + R_rectangle:<{col_width}}\t"
    ))

    for row in rows:
        print(f"{row[0]:<{col_width}}" + "".join(f"{cell:<{col_width}}" for cell in row[1:]))

if __name__ == "__main__":
    main()
```

5 Результаты



```
eps = 0.001, I = 7.733333333333333
                Симпсон                 Трапеции                Прямоугольники
------------------------------------------------------------------------------
n               4                       64                      32
I*              7.733402532029047       7.733413441686172       7.733173120103303
R               -5.6901138686941505e-05 -8.010719429100514e-05  0.00016019703476851296
I* + R          7.73334563089036        7.73333333449188        7.733333317138071
```

Рисунок 1 Результаты вычислений

6 Вывод

В ходе выполнения лабораторной работы были изучены методы численного интегрирования, а также создана реализация на языке python. При достаточно большом разбиении рассматриваемого отрезка самым точным окажется метод Симпсона. При сравнении метода прямоугольников и трапеций первый метод окажется точнее.