Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования Московский государственный технический университет имени Н.Э. Баумана

Лабораторная работа №3
«Численное интегрирование»
по дисциплине
«Численные методы»

Студент группы ИУ9-62 Белогуров А.А.

Преподаватель Домрачева А.Б.

Москва, 2017

Цель: Сравнительный анализ методов численного интегрирования:

1. Метод средних прямоугольников.
2. Метод квадраторных трапеций.
3. Метод Симпсона.

Постановка задачи:

Дан интеграл:

$$\int_a^b f(x)dx$$

где $a, b, f(x)$ - известны.

Найти интеграл:

$$I \approx I^*$$

Тестовый пример (1):

$$I = \int_0^1 e^x dx = e^x \Big|_0^1 = e - 1 = 1.718282 \tag{1}$$

Теоретические сведения:

Численное интегрирование — вычисление значения определённого интеграла (как правило, приближённое). Под численным интегрированием понимают набор численных методов для нахождения значения определённого интеграла.

Основная идея большинства методов численного интегрирования состоит в замене подынтегральной функции на более простую, интеграл от которой легко вычисляется аналитически. При этом для оценки значения интеграла получаются формулы вида:

$$I \approx \sum_{i=1}^{n} w_i f(x_i)$$

При замене подынтегральной функции на полином нулевой, первой и второй степени получаются соответственно методы прямоугольников, трапеций и парабол (Симпсона).

1. Метод средних прямоугольников

Если рассмотреть график подынтегральной функции, то метод будет заключаться в приближённом вычислении площади под графиком суммированием площадей конечного числа прямоугольников, ширина которых будет определяться расстоянием между соответствующими соседними узлами интегрирования, а высота — значением подынтегральной функции в этих узлах.

Положим, что $h = \frac{a-b}{n}$ тогда получим следующее разбиение отрезка $[a, b]$ на n равных частей (2):

$$x_0 = a, x_1 = x_0 + h, x_2 = x_1 + h, ..., x_n = x_{n-1} + h \tag{2}$$

Тогда приближенное значение интеграла с помощью метода средних прямоугольников будет вычисляться по формуле (3):

$$I^* = \sum_{i=1}^{n} I_i^* = h \sum_{i=1}^{n} f(\frac{x_{i-1} + x_i}{2}) = h \sum_{i=1}^{n} f(a + i * h - \frac{h}{2}) \tag{3}$$

2. 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.

Разбиение производится аналогично методу средних прямоугольников (2).

Квадраторная формула трапеций будет выглядеть следующим образом (4), (5):

$$I_i^* = \frac{f(x_{i-1}) + f(x_i)}{2}h \tag{4}$$

$$I^* = \sum_{i=1}^{n} I_i^* = \frac{h}{2}\sum_{i=1}^{n}(f(x_0)+f(x_1)+...+f(x_n)) = h(\frac{f(a)+f(b)}{2}+\sum_{i=1}^{n-1} f(x_i)) \tag{5}$$

3. Метод Симпсона

Суть метода заключается в приближении подынтегральной функции на отрезке $[a, b]$ интерполяционным многочленом второй степени $P_2(x)$, то есть приближение графика функции на отрезке параболой.

Разбиение производится аналогично методу средних прямоугольников (2).

Парабола Симпсона представлена формулой (6):

$$P_2(x) = f(x_{i-0.5}) + \frac{f(x_i) - f(x_{i-1})}{h}(x - x_{i-0.5}) + \\ + \frac{f(x_i) - 2f(x_{i-0.5}) + f(x_{i-1})}{\frac{h^2}{2}} * (x - x_{i-0.5})^2 \tag{6}$$

Тогда формула Симпсона выглядит следующим образом(7), (8):

$$I_i^* = \int_{x_{i-1}}^{x_i} P_2(x)dx = \frac{h}{6}(f(x_{i-1}) + 4f(x_{i-0.5}) + f(x_i)) \tag{7}$$

$$I* = \sum_{i=1}^{n} I_i^* = \frac{h}{6}(f(a) + f(b) + 4\sum_{i=1}^{n} f(x_{i-0.5}) + 2\sum_{i=1}^{n-1} f(x_i)) \qquad (8)$$

Вычисление интегралов с учетом погрешности

$I = I^* + O(h^k)$, где k - порядок точности метода.

$k = 2$ - для метода средних прямоугольников и трапеций.

$k = 4$ - для метода Симпсона.

$O(h^k) = ch^k$, где h - шаг, c - некоторая константа.

Тогда:

$$I \approx I_h^* + ch^k$$

Считаем, что вычисления проводятся без вычислительной погрешности, тогда можно записать строгое равенство:

$$I = I_h^* + ch^k \qquad (9)$$

$$I = I_{\frac{h}{2}}^* + c(\frac{h}{2})^k \qquad (10)$$

Из равенств (9) и (10) следует равенство (11):

$$I_h^* + ch^k = I_{\frac{h}{2}}^* + c(\frac{h}{2})^k \qquad (11)$$

$$c(\frac{h}{2})^k = \frac{I_h^* - I_{\frac{h}{2}}^*}{1 - 2^k} \qquad (12)$$

Подставим (12) в (9) и получим значение интеграла с погрешностью(13):

$$I = I_{\frac{h}{2}}^* + \frac{I_h^* - I_{\frac{h}{2}}^*}{1 - 2^k} \qquad (13)$$

Где значение R (14) есть уточнение по Ричардсону:

$$R = \frac{I_h^* - I_{\frac{h}{2}}^*}{1 - 2^k} \tag{14}$$

Практическая реализация:

Листинг 1: Численное интегрирование

```python
import math


# Quadrature trapezoidal formula
def method_of_trapezoid(a=0, b=1, n=2):
    h = (b - a) / n
    list_of_x = [a + i * h for i in range(1, n)]

    series_of_sum = 0
    for i in range(0, n - 1):
        series_of_sum += math.exp(list_of_x[i])

    return h * ((math.exp(a) + math.exp(b)) / 2 + series_of_sum)


# Simpson's rule
def method_of_simpson(a=0, b=1, n=2):
    h = (b - a)/n
    list_of_x1 = [a + h * i  - h/2 for i in range(1, n+1)]
    list_of_x2 = [a + h * i for i in range(1, n)]

    series_of_sum = 0
    for i in range(0, n):
        series_of_sum += 4 * math.exp(list_of_x1[i])
        if i < n-1:
            series_of_sum += 2 * math.exp(list_of_x2[i])

    return h/6 * (math.exp(a) + math.exp(b) + series_of_sum)


# Method of medium rectangles
def method_of_rectangles(a=0, b=1, n=2):
    h = (b - a)/n
```

```python

    result = 0
    for i in range(0, n):
        result += h * math.exp(a + h * i + h/2)

    return result


# Clarification on Richardson
def get_richardson(i_n, i_n2, k):
    return (i_n - i_n2) / (2 ** k - 1)


def calc_integral(epsilon, method, k):
    for i in range(len(epsilon)):
        print("\nEpsilon: " + str(epsilon[i]))
        n = 1
        richardson = float('inf')
        iteration = i_n = 0

        while abs(richardson) >= epsilon[i]:
            n *= 2
            i_n2 = i_n
            i_n = method(n=n)

            richardson = get_richardson(i_n, i_n2, k)
            iteration += 1

        print(" Iteration: " + str(iteration))
        print(" Result: " + str(i_n))
        print(" Result with Richardson: " + str(i_n + richardson))


if __name__ == "__main__":
    eps = [10 ** (-i) for i in range(1, 4)]

    print("\nMethod of trapezoid:")
    calc_integral(eps, method_of_trapezoid, 2)

    print("\nMethod of rectangles:")
    calc_integral(eps, method_of_rectangles, 2)

```

```python
    print("\nMethod of Simpson:")
    calc_integral(eps, method_of_simpson, 4)
```

Результаты:

Для тестирования был выбран интеграл (1).

В качестве ε для каждого метода были выбраны следующие значения:

$$\varepsilon = 0.1, \varepsilon = 0.01, \varepsilon = 0.001$$

Ниже приведена таблица результата тестовой программы (Листинг 1) на указанных выше методах:

Метод	Кол-во итераций	Значение без уточнения по Ричардсону	Значение с уточнением по Ричардсону
$\varepsilon = 0.1$			
Метод средних прямоугольников	2	1.713815	1.718249
Метод трапеций	2	1.727222	1.718319
Метод Симпсона	2	1.718284	1.718282
$\varepsilon = 0.01$			
Метод средних прямоугольников	2	1.713815	1.718249
Метод трапеций	2	1.727222	1.718319
Метод Симпсона	2	1.718284	1.718282
$\varepsilon = 0.001$			
Метод средних прямоугольников	4	1.718002	1.718282
Метод трапеций	4	1.718841	1.718282
Метод Симпсона	2	1.718284	1.718282

Выводы:

В ходе выполнения лабораторной работы были рассмотрены различных 3 метода численного интегрирования: метод средних прямоугольников, метод трапеций и метод Симпсона. Была написана реализация данных методов на языке программирования Python.

Как видно выше в таблице вычислений, среди всех методов выигрывает метод Симпсона (он точнее и для вычислений при $\varepsilon = 0.001$ ему понадобилось меньше итераций для получения результата).

Если же сравнивать между собой два остальных метода, то метод средних прямоугольников является точнее метода трапеций. Но в то же время метод трапеций может применяться с произвольным шагом, в отличие от метода средних прямоугольников, который не применим, например, к функциям, заданным в конечном числе точек.