Федеральное государственное автономное образовательное учреждение высшего образования «Московский физико-технический институт (национальный исследовательский университет)»

Физтех-школа аэрокосмических технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ – 1

Численное вычисление интегралов

Выполнил:
Алиев Артем Эльдарович

Группа Б03-907

Долгопрудный

2021 г.

Содержание

1 Условие

Найти значение интеграла вида: **x^5 / (1 – tg(x^7))**, на отрезке [a, b], где **a = -1, b = 0.5** с заданной точностью **E = 10^{-4}**, используя 4 метода:

1 Средних прямоугольников

2 Правых прямоугольников

3 Трапеций

4 Симпсона

2 Теория

Самые широко используемые в практических вычислениях - методы прямоугольников, трапеций, Симпсона .Способ их получения состоит в следующем. Разобьем отрезок интегрирования $[a, b]$ на N элементарных шагов. Точки разбиения $x_n (n = 0, 1, ..., N)$; $h_n = x_{n+1} - x_n$ так что $\sum_{n=0}^{N-1} h_n = b - a$.

(В частном случае шаг интегрирования может быть постоянным $h = (b - a)/N$.)

Искомое значение интеграла представим в виде

$$J = \sum_{n=0}^{N-1} \int_{x_n}^{x_{n+1}} f(x)\, dx = \sum_{n=0}^{N-1} J_n, \qquad (1)$$

Метод трапеций

На элементарном отрезке $[x_n, x_{n+1}]$ заменим подынтегральную функцию интерполяционным полиномом первой степени:

$$f(x) \approx f_n + \frac{f_{n+1} - f_n}{x_{n+1} - x_n}(x - x_n)$$

Выполняя интегрирование по отрезку, приходим к локальной формуле трапеций:

$$\tilde{J}_n = \frac{1}{2}(x_{n+1} - x_n)(f_{n+1} + f_n) = \frac{1}{2} h_n (f_{n+1} + f_n) \qquad (2)$$

Суммируя (2) по всем отрезкам, получаем формулу трапеций для вычисления приближения:

$$\tilde{J} = \frac{1}{2}\sum_{n=0}^{N-1} h_n(f_n + f_{n+1})$$

Погрешность для метода трапеций

Формула трапеций является формулой второго порядка, поэтому формула оценки погрешности имеет вид:

$$|\tilde{J} - J| \leq \frac{1}{12}(b-a)M_2\overline{h}^2$$

Правило Рунге

Для контроля точности используется правило Рунге, для которого не надо вычислять даже производные.

$$J - \tilde{J}^p(h) \approx (\tilde{J}^p(h) - \tilde{J}^p(2h))/(2^p - 1)$$

3 Решение

3.1 Проверка интеграла

Подставив обе точки *a* и *b* в нашу функцию мы обнаружили, что в нашем случае *отсутствуют особые точки* и “неприятности”, поэтому смело приступаем к программному вычислению интеграла разными методами.

3.2 Метод средних прямоугольников

$$I = \sum_{n=0}^{N-1} h_n f_{n+\frac{1}{2}}$$ Ответ: -0.111595650530556

```
def aver_rectangle(f, h, a):


  n = int((b - a) / h)
  res = 0
  for i in range(n):
    res += (h * f.subs(x, (a + h/2) + i*h))
  return res
```

3.3 Метод правых прямоугольников

$$I = \sum_{n=0}^{N-1} h_n f_{n+1}$$

Ответ: -0.111597763090279

```
def right_rectangle(f, h, a):


    n = int((b - a) / h)
    res = 0
    for i in range(n):
        res += (h * f.subs(x, a + i * h))
    return res
```

3.4 Метод трапеций

$$I = \sum_{n=0}^{N-1} \frac{h_n}{2}(f_{n+1} + f_n)$$

Ответ: -0.111597605609942

```
def trapezoidal(f, a, b, h):


    n = int((b - a) / h)
    result = 0.5*f.subs(x, a) + 0.5*f.subs(x, b)
    for i in range(n):
        result += f.subs(x, a + i*h)
    result *= h
    return result
```

3.5 Метод Симпсона

$$I = \sum_{n=0}^{N-1} \frac{h_n}{6}(f_{n+1} + 4f_{n+\frac{1}{2}} + f_n)$$

Ответ: -0.11159825732843703

```
def simpsons(f, a, b, h):


    tmp_sum = float(f.subs({x: a})) + \
            float(f.subs({x: b}))
    n = int((b - a) / h)
    for step in range(n):
        if step % 2 != 0:
            tmp_sum += 4 * float(f.subs({x: a + step * h}))
        else:
            tmp_sum += 2 * float(f.subs({x: a + step * h}))


    return tmp_sum * h / 3
```

4 Результаты

Наиболее точным методом является метод средних квадратов: -0.111595650530556 (точное значение равно -0.1115956505221).

5 Вывод

В данной задаче мы воспользовались формулой Маклорена для нахождения одной из частей интеграла и 4-мя численными методами интегрирования для нахождения второго. В каждом из методов получилось не выйти за рамки заданной точности, что можно считать успешным выполнением данной работы.