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

ФАКУЛЬТЕТ АЭРОФИЗИКИ И КОСМИЧЕСКИХ ТЕХНОЛОГИЙ

Группа
Б03-908

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ - 3

Численное вычисление интегралов

Выполнил:

Агеев Рамиль Наильевич

/ / /
(подпись) *(дата)*

Долгопрудный
2021г.

Содержание

1 Условие задачи

Требуется найти значение интеграла вида:

$$\ln{(1 + x^{\frac{2}{3}})}/x \tag{1}$$

, на отрезке $[a, b]$, где $a = 0, b = 1$. Заданная точность $\epsilon = 10^{-4}$.

Решить требуется 4-мя методами:

1. Прямоугольный Средний;
2. Прямоугольный Правый;
3. Метод Трапеций;
4. Метод Симпсона.

2 Теоретические сведения

2.1 Подготовка к решению

Для начала стоит заметить, что в интеграле есть особая точка в $a = 0$. Для того, чтобы избавиться от этой "неприятности"следует разделить интеграл на две части:

$$\int_0^\delta \ln{(1 + x^{\frac{2}{3}})}/x dx \text{ и } \int_\delta^1 \ln{(1 + x^{\frac{2}{3}})}/x dx$$

Первый интеграл с помощью разложения по Маклорену приводится упрощенному и там же находится нужный $\delta = 10^{-6}/4$. И найденный интеграл равен 0.000059

Второй интеграл легко находится с помощью методов численного интегрирования, приведенными ниже.

2.2 Общая Теория

Для начала опишем общие переменные и концепции, которые будут использоваться в каждом методе.

$$h_n - \text{шаг интегрирования, который в рамках нашей задачи будет постоянен.}$$

,

$$f_{n+\frac{1}{2}} = f(x + \frac{h_n}{2})$$

$$f_n = f(x_n)$$

$$f_{n+1} = f(x_{n+1})$$

$$I = \int_a^b x^2 dx$$

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

В концепции каждого метода применяется идея о том, чтобы разделить график функции на N частей, которые будут суммироваться с шагом h.

Во всех методах я пользовался шагом, определенным для метода средних прямоугольников $h = 10^{-5}$, так как данный шаг по точности будет точно подходить и для других методов. Определил я его методом Рунге, который будет описан ниже. Причиной явился тот факт, что численно точно оказалось невозможным посчитать максимальные значения функции на данном отрезке. В связи с этим формулы для определения погрешностей, пройденные на занятиях, не удалось применить(но общий принцип также будет описан ниже).

Метод Рунге

Выбираем h и $2h$, потом смотрим, чтобы разница оказалась меньше заданной точности.

$$I_{\text{точный}} = \tilde{I}_h + 2^p * ch^4, \text{обозначим за B}$$

$$I_{\text{точный}} = \tilde{I}_{2h} + 2^p ch^4$$

$$|\tilde{I}_{2h} - \tilde{I}_h| \approx (2^p - 1)ch^4$$

$$I_T - \tilde{I}_h \approx \frac{B}{2^p - 1}$$

Формула оценки погрешностей для разных методов

$$|\tilde{I} - I| \leq \frac{M_i * (b - a) * h}{2^p}$$

Где $M_i = \max |f'|$ на $[a, b]$.

2.3 Метод Средних Прямоугольников

Формула и ответ

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

```
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
```

С помощью этого метода получили значение *1.23362*

2.4 Метод Правых Прямоугольников

Формула и ответ

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

```
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
```

С помощью этого метода получили значение *1.23353*

2.5 Метод Трапеций

Формула и ответ

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

```
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
```

С помощью этого метода получили значение *1.23366*

2.6 Метод Симпсона

Формула и ответ

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

```
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
```

С помощью этого метода получили значение *1.23345*

3 Ответ

В итоге, наиболее точным методом оказался Метод Симпсона (точное значение равно 1.2334).

4 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 успехом!