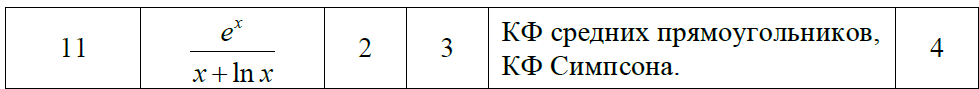
**Лабораторная работа №1**

**«Приближенное вычисление интегралов»**

Чеботаревский Никита, 3 курс, 8 группа

**Постановка задачи**

Вычислить интеграл с точностью ε = , используя составные квадратурные формулы (КФ), указанные в варианте задания, и правило Рунге оценки погрешности.



В содержание отчета должна быть включена следующая информация:

1) Применяемые составные квадратурные формулы

2) Правило Рунге оценки погрешности

3) Результаты вычислительного эксперимента, оформленного в виде таблицы

4) Выводы

5) Листинг программы с комментариями

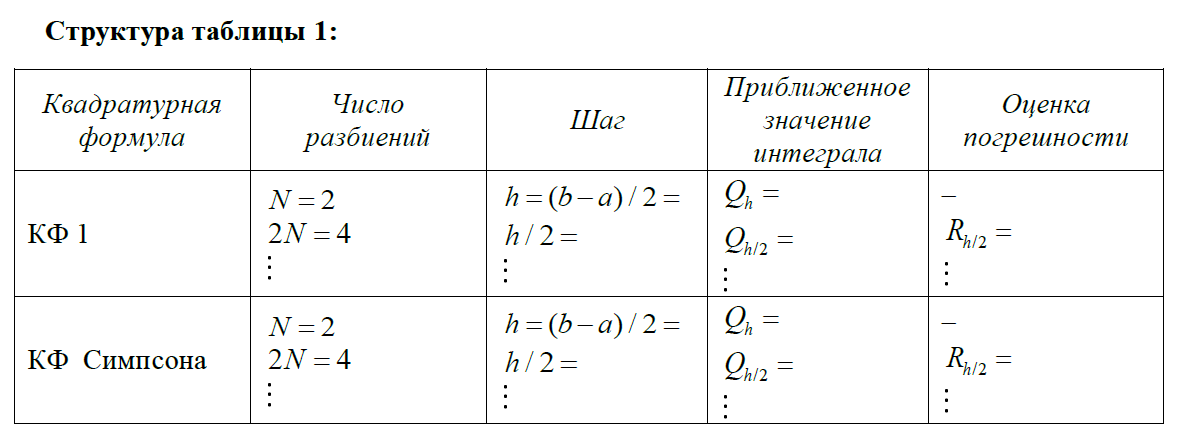
Вычислить приближенное значение интеграла из задания 1, используя квадратурную формулу наивысшей алгебраической степени точности (НАСТ) с *k* узлами.

В содержание отчёта должна быть включена следующая информация:

1) КФ НАСТ с k узлами

2) Приближённое значение интеграла, вычисленное с помощью КФ НАСТ

3) Листинг программы с комментариями



**Теоретические сведения**

Составная квадратурная формула средних прямоугольников:

Составная квадратурная формула Симпсона:

Практическое правило двойного пересчёта

1) Пусть – число разбиений, – заданная точность.

2) Вычисляем – значения интеграла, вычисленные по одной из составных квадратурных формул с числом разбиений соответственно.

3) Посчитаем погрешность для числа разбиений по следующей формуле:

m – порядок точности квадратурной формулы:

m для составной квадратурной формулы средних прямоугольников равна 2,

m для составной квадратурной формулы Симпсона равна 4.

4)

Если данное условие не выполняется, то переходим к следующему пункту

5)

Задание 2

*.*

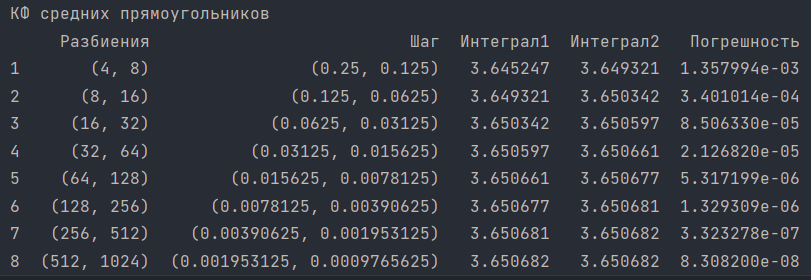
Так как у нас весовая функция p(x) = 1, то для k = 4 узлов, наше t является узлом Лежандра 3 степени t = .

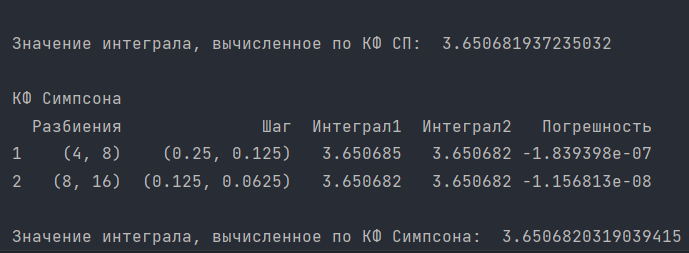
**Листинг программы**

*import* numpy *as* np  
*import* pandas *as* pd  
  
  
*def* f(x):  
 *return* np.exp(x) / (x + np.log(x))  
  
  
*def* count\_h(n, left, right):  
 *return* (right - left) / n  
  
  
*def* count\_r(m, integral1, h1, integral2, h2):  
 result = (integral2 - integral1) / (h1 \*\* m - h2 \*\* m)  
 *return* result \* h2 \*\* m  
  
  
*def* count\_q\_with\_middle\_rectangles(n, left, right):  
 h = count\_h(n, left, right)  
 q = 0  
  
 *for* k *in* range(n):  
 q += f(left + k \* h + h / 2)  
  
 *return* h \* q  
  
  
*def* count\_q\_with\_simpson(n, left, right):  
 h = count\_h(n, left, right)  
 result = f(left) + f(right)  
  
 sum1 = 0  
 *for* k *in* range(1, n):  
 sum1 += f(left + k \* h)  
  
 result += (2 \* sum1)  
  
 sum2 = 0  
 *for* k *in* range(n):  
 sum2 += f(left + k \* h + h / 2)  
  
 result += (4 \* sum2)  
 *return* (h / 6) \* result  
  
  
*def* count\_integral(n1, n2, e, left, right, func, m):  
 q1 = func(n1, left, right)  
 h1 = count\_h(n1, left, right)  
 q2 = func(n2, left, right)  
 h2 = count\_h(n2, left, right)  
 r = count\_r(m, q1, h1, q2, h2)  
 data = []  
 k = 1  
  
 *while True*:  
 *if* abs(r) <= e:  
 *return* data, np.arange(1, k)  
 *else*:  
 k += 1  
 n1, n2 = n2, n2 \* 2  
 q1 = q2  
 h1 = count\_h(n1, left, right)  
 q2 = func(n2, left, right)  
 h2 = count\_h(n2, left, right)  
 r = count\_r(m, q1, h1, q2, h2)  
 data.append([(n1, n2), (h1, h2), q1, q2, r])  
  
  
*def* integral(left, right):  
 t = [-(3 / 5) \*\* (1 / 2), 0, (3 / 5) \*\* (1 / 2)]  
 y = [5 / 9, 8 / 9, 5 / 9]  
 result = 0  
  
 *for* i *in* range(len(t)):  
 result += f((left + right) / 2 + t[i] \* (right - left) / 2) \* y[i]  
  
 *return* result \* ((right - left) / 2)  
  
  
*if* \_\_name\_\_ == '\_\_main\_\_':  
 name\_columns = ['Разбиения', 'Шаг', 'Интеграл1', 'Интеграл2', 'Погрешность']  
 a, b = 2, 3  
 eps = 10 \*\* (-7)  
 m\_s = [2, 4]  
 functions = [count\_q\_with\_middle\_rectangles, count\_q\_with\_simpson]  
  
 info, index = count\_integral(2, 4, eps, 2, 3, functions[0], m\_s[0])  
 info1, index1 = count\_integral(2, 4, eps, 2, 3, functions[1], m\_s[1])  
  
 print('КФ средних прямоугольников')  
 print(pd.DataFrame(info, index, name\_columns), end='\n' \* 2)  
 print(f'Значение интеграла, вычисленное по КФ СП: {info[-1][3]}', end='\n' \* 2)  
  
 print('КФ Симпсона')  
 print(pd.DataFrame(info1, index1, name\_columns), end='\n' \* 2)  
 print(f'Значение интеграла, вычисленное по КФ Симпсона: {info1[-1][3]}', end='\n' \* 2)  
  
 print(f"Значение интеграла, вычисленное с помощью КФ НАСТ: {integral(a, b)}")

**Результаты вычислительного эксперимента:**

**Задание 1:**





**Задание 2:**

****

**Вывод:**

С помощью квадратичных формул, рассмотренных выше можно вычислить приближённое значение интеграла с заданной точностью, однако сделать это можно с разной скоростью. По формуле Симпсона вычисление происходит быстрее, так как алгебраическая степень точности у формулы Симпсона 4, а у формулы средних прямоугольников 2.