Российский университет дружбы народов им. П. Лумумбы

Факультет физико-математических и естественных наук

Лабораторная работа №3

Дисциплина: Вычислительные методы

Студент: Шуплецов Александр Андреевич

Группа: НФИбд-01-22

Москва

2024 г.

Оглавление

Задание

Интегрирование

1. Реализовать в программе методы левых и правых прямоугольников, метод трапеций и метод Симпсона для приближенного расчета интеграла $\int_a^b f(x)dx$.

   В программной реализации предусмотреть разбиение отрезка $[a;b]$, по которому ведется интегрирование, на M отрезков равной длины.

2. Вычислить аналитически значение интеграла $I = \int_a^b f(x)dx$ (<u>функция $f(x)$ и отрезок даны в индивидуальном задании</u>).

   Сравнить в программе полученное аналитическое значение I с приближенными значениями интеграла $\int_a^b f(x)dx$, вычисленными с помощью методов левых и правых прямоугольников, метода трапеций и метода Симпсона.

   Вывести в программе таблицу следующего вида

N	I	I_L^N	I_R^N	I_T^N	I_S^N
$2N$	I	I_L^{2N}	I_R^{2N}	I_T^{2N}	I_S^{2N}
$5N$	I	I_L^{5N}	I_R^{5N}	I_T^{5N}	I_S^{5N}
$10N$	I	I_L^{10N}	I_R^{10N}	I_T^{10N}	I_S^{10N}

где

$N = 16$,

I – аналитическое значение интеграла $\int_a^b f(x)dx$,

I_L^M – приближенное значение интеграла $\int_a^b f(x)dx$, вычисленное методом левых прямоугольников с помощью разбиения отрезка интегрирования $[a;b]$ на M отрезков равной длины;

I_R^M – приближенное значение интеграла $\int_a^b f(x)dx$, вычисленное методом правых прямоугольников с помощью разбиения отрезка интегрирования $[a;b]$ на M отрезков равной длины;

I_T^M – приближенное значение интеграла $\int_a^b f(x)dx$, вычисленное методом трапеций с помощью разбиения отрезка интегрирования $[a;b]$ на M отрезков равной длины;

I_S^M – приближенное значение интеграла $\int_a^b f(x)dx$, вычисленное методом Симпсона с помощью разбиения отрезка интегрирования $[a;b]$ на M отрезков равной длины.

3. Вычислить <u>для метода левых прямоугольников</u> минимальное значение целого числа M, при котором погрешность интегрирования меньше, чем 10^{-3}.
Вывести в программе таблицу следующего вида:

$M-2$	I	I_L^{M-2}	$\left\|I-I_L^{M-2}\right\|$
$M-1$	I	I_L^{M-1}	$\left\|I-I_L^{M-1}\right\|$
M	I	I_L^{M}	$\left\|I-I_L^{M}\right\|$
$M+1$	I	I_L^{M+1}	$\left\|I-I_L^{M+1}\right\|$

4. Вычислить <u>для метода правых прямоугольников</u> минимальное значение целого числа M, при котором погрешность интегрирования меньше, чем 10^{-3}.
Вывести в программе таблицу следующего вида:

$M-2$	I	I_R^{M-2}	$\left\|I-I_R^{M-2}\right\|$
$M-1$	I	I_R^{M-1}	$\left\|I-I_R^{M-1}\right\|$
M	I	I_R^{M}	$\left\|I-I_R^{M}\right\|$
$M+1$	I	I_R^{M+1}	$\left\|I-I_R^{M+1}\right\|$

5. Вычислить <u>для метода трапеций</u> минимальное значение целого числа M, при котором погрешность интегрирования меньше, чем 10^{-3}.
Вывести в программе таблицу следующего вида:

$M-2$	I	I_T^{M-2}	$\left\|I-I_T^{M-2}\right\|$
$M-1$	I	I_T^{M-1}	$\left\|I-I_T^{M-1}\right\|$
M	I	I_T^{M}	$\left\|I-I_T^{M}\right\|$
$M+1$	I	I_T^{M+1}	$\left\|I-I_T^{M+1}\right\|$

6. Вычислить <u>для метода Симпсона</u> минимальное значение целого <u>четного</u> числа M, при котором погрешность интегрирования меньше, чем 10^{-3}.
Вывести в программе таблицу следующего вида:

$M-4$	I	I_S^{M-4}	$\left\|I-I_S^{M-4}\right\|$
$M-2$	I	I_S^{M-2}	$\left\|I-I_S^{M-2}\right\|$
M	I	I_S^{M}	$\left\|I-I_S^{M}\right\|$
$M+2$	I	I_S^{M+2}	$\left\|I-I_S^{M+2}\right\|$

7. В сравнить полученные в пп.3-6 значения M для каждого из реализованных методов, проанализировать полученные результаты.

Я возьму вариант 19:

19.	$(x-1)\sin(x)$	$[-1,1]$

Теоретическая справка

Данная лабораторная работа представляет собой компьютерную реализацию численного интегрирования.

2.1. Формула Ньютона-Лейбница и численное интегрирование.

Из курса математического анализа вы знакомы с вычислением определенных интегралов с помощью формулы Ньютона-Лейбница:

$$I = \int_a^b f(x)dx = F(b) - F(a), \qquad (2.1)$$

Универсальные алгоритмы вычисления определенных интегралов дают формулы численного интегрирования или, как их обычно называют, *квадратурные формулы* (буквально — формулы вычисления площадей). Квадратурные формулы имеют вид

$$I = \int_a^b f(x)dx = \sum_{i=1}^{n} c_i f(x_i) + R_n. \qquad (2.2)$$

Полный программный код, с подробным описанием функций, реализующих численное интегрирование.

```
import math

def func(x):
    return (x-1)*math.sin(x)

def splitter(a,b,M):
    step = (b-a)/M
    arr_x = [0]*(M+1)
    for i in range (0,M+1):
        arr_x[i] = a + step*i
    return arr_x


def middle(a,b,M):
    arr_m = [0]*(M+1)
    for i in range (0,M+1):
        arr_m[i] = a + (i-1/2)*((b-a)/M)
    return arr_m

def middle_rectangle(a,b,M):
    f = 0
    for i in range (1,M+1):
        ksi = a + (i-1/2)*((b-a)/M)
        f = f + func(ksi)
    I = ((b-a)/M)*f
    return I

def left_rectangle(a,b,M):
    f = 0
    arr_x = splitter(a,b,M)
    arr_y = [0]*(M+1)
    for i in range (M):
        arr_y[i] = func(arr_x[i])
    for i in range (M):
        f = f + arr_y[i]
    I = ((b-a)/M)*f
    return I

def right_rectangle(a,b,M):
    f = 0
    arr_x = splitter(a,b,M)
    arr_y = [0]*(M+1)
    for i in range (1, M+1):
        arr_y[i] = func(arr_x[i])
    for i in range (1,M+1):
        f = f + arr_y[i]
    I = ((b-a)/M)*f
    return I

def trapez(a,b,M):
    arr_x = splitter(a,b,M)
    arr_y = [0]*(M+1)
```

```
    for i in range (0, M+1):
        arr_y[i] = func(arr_x[i])
    summ = 0.5*(func(a) + func(b))
    f_t = 0
    for i in range (1,M):
        f_t = f_t + arr_y[i]
    T = ((b-a)/M)*(summ + f_t)
    return T


def simpson(a,b,M):
    arr_x = splitter(a,b,M)
    arr_y = [0]*(M+1)
    for i in range (0, M+1):
        arr_y[i] = func(arr_x[i])
    summ = func(a) + func(b)
    f_t = 0
    for i in range (1,M):
        if i%2 == 0:
            f_t = f_t + 2*arr_y[i]
        else:
            f_t = f_t + 4*arr_y[i]
    T = ((b-a)/(3*M))*(summ + f_t)
    return T

count_tab = 10
N = 16
I = 0.6023373578795136
for i in range (4):
    if i == 0:
        print ("{:<6d}{:<20}{:<10f}{:<10f}{:<10f}{:<10f}".format(N, I,
left_rectangle(-1, 1, N), right_rectangle(-1, 1, N), trapez(-1, 1, N),
simpson(-1,1,N)))
    if i == 1:
        M = 2*N
        print ("{:<6d}{:<20}{:<10f}{:<10f}{:<10f}{:<10f}".format(M, I,
left_rectangle(-1, 1, M), right_rectangle(-1, 1, M), trapez(-1, 1, M),
simpson(-1,1,M)))
    if i == 2:
        M = 5*N
        print ("{:<6d}{:<20}{:<10f}{:<10f}{:<10f}{:<10f}".format(M, I,
left_rectangle(-1, 1, M), right_rectangle(-1, 1, M), trapez(-1, 1, M),
simpson(-1,1,M)))
    if i == 3:
        M = 10*N
        print ("{:<6d}{:<20}{:<10f}{:<10f}{:<10f}{:<10f}".format(M, I,
left_rectangle(-1, 1, M), right_rectangle(-1, 1, M), trapez(-1, 1, M),
simpson(-1,1,M)))

print('------------------------------')

for i in range (1,10000):
    if (abs(I - left_rectangle(-1, 1, i)) < 10**(-3)):
        print('Минимальное значение M для метода левых прямоугольников: ', i)
        min_M = i
        print('\n')
        break
M = min_M
for i in range (M-2, M+2):
```

```
    print("{:<6d}{:<20}{:<20f}{:.20f}".format(i, I,  left_rectangle(-1, 1, 
i), abs(I - left_rectangle(-1, 1, i))))

print('------------------------------')

for i in range (1,10000):
    if (abs(I - right_rectangle(-1, 1, i)) < 10**(-3)):
        print('Минимальное значение M для метода правых прямоугольников: ', 
i)
        min_M = i
        print('\n')
        break
M = min_M

for i in range (M-2, M+2):
    print("{:<6d}{:<20}{:<20f}{:<20f}".format(i, I,  right_rectangle(-1, 1, 
i), abs(I - right_rectangle(-1, 1, i))))

print('------------------------------')

for i in range (1,10000):
    if (abs(I - trapez(-1, 1, i)) < 10**(-3)):
        print('Минимальное значение M для метода трапеции: ', i)
        min_M = i
        print('\n')
        break
M = min_M

for i in range (M-2, M+2):
    print("{:<6d}{:<20}{:<20f}{:<20f}".format(i, I,  trapez(-1, 1, i), abs(I 
- trapez(-1, 1, i))))

print('------------------------------')

for i in range (1,10000):
    if (abs(I - simpson(-1, 1, i)) <= 10**(-3)):
        print('Минимальное значение M для метода Симпсона: ', i)
        min_M = i
        print('\n')
        break
M = min_M

for i in range (M-2, M+2):
    if i%2 == 0:
        print("{:<6d}{:<20}{:<20f}{:<20f}".format(i, I,  simpson(-1, 1, i), 
abs(I - simpson(-1, 1, i))))
```

Функции

func(x): определяет функцию ((x-1)\sin(x)), которую мы будем интегрировать.

splitter(a, b, M): создает массив точек (узлов) для разбивки интервала ([a, b]) на (M) частей.

middle(a, b, M): создает массив средних значений между узлами для вычисления суммы в методе средних прямоугольников.

middle_rectangle(a, b, M): вычисляет интеграл по методу средних прямоугольников. Здесь выбирается точка в середине каждого подотрезка и суммируется значение функции в этих средних точках.

left_rectangle(a, b, M): вычисляет интеграл по методу левых прямоугольников, используя левую границу каждого подотрезка как точку, в которой вычисляется значение функции.

right_rectangle(a, b, M): вычисляет интеграл по методу правых прямоугольников, используя правую границу каждого подотрезка.

trapez(a, b, M): реализует метод трапеций. Это метод использует среднее значение функции на каждом подотрезке.

simpson(a, b, M): реализует метод Симпсона, который является более сложной техникой, использующей квадратичное приближение и чередующиеся коэффициенты (2) и (4) для внутренних точек по отношению к крайним точкам.

Основной код

N изначально установлено как 16, это начальное количество подотрезков для разбиения интервала.

I — истинное значение интеграла на интервале ([-1, 1]), предварительно вычислено.

Проходят 4 итерации увеличивая количество подотрезков (M) (M = N, M = 2N, M = 5N, M = 10*N) и вычисляют аппроксимированное значение интеграла для каждого из методов вывода на экран результата и сравнения с (I).

Затем, в отдельных циклах for, с силой набора подотрезков ((M)) проверяется, при каком наименьшем (M) каждый из методов дает приближение с точностью (10^{-3}).

Распечатка значений при (M-2, M-1, M, M+1) позволяет увидеть ближайшие значения вокруг найденного минимального (M) для более детального анализа.

Цель кода состоит в сравнении различных методов численного интегрирования по их точности и скорости сходимости к истинному значению интеграла.

Численные расчеты

```
16    0.6023373578795136  0.711122  0.500754  0.605938  0.602329
32    0.6023373578795136  0.655829  0.550645  0.603237  0.602337
80    0.6023373578795136  0.623518  0.581445  0.602481  0.602337
160   0.6023373578795136  0.612892  0.591855  0.602373  0.602337
------------------------------
Минимальное значение M для метода левых прямоугольников:  1684


1682  0.6023373578795136  0.603338            0.0010008856363775686 7
1683  0.6023373578795136  0.603338            0.0010002907397379523 5
1684  0.6023373578795136  0.603337            0.0009996965498528798 1
1685  0.6023373578795136  0.603336            0.0009991030654641353 0
------------------------------
Минимальное значение M для метода правых прямоугольников:  1683


1681  0.6023373578795136  0.601337            0.001001
1682  0.6023373578795136  0.601337            0.001000
1683  0.6023373578795136  0.601338            0.001000
1684  0.6023373578795136  0.601338            0.000999
------------------------------
Минимальное значение M для метода трапеции:  31


29    0.6023373578795136  0.603433            0.001096
30    0.6023373578795136  0.603361            0.001024
31    0.6023373578795136  0.603296            0.000959
32    0.6023373578795136  0.603237            0.000900
------------------------------
Минимальное значение M для метода Симпсона:  6


4     0.6023373578795136  0.600107            0.002230
6     0.6023373578795136  0.601908            0.000429
```

Вывод

Я реализовал численное интегрирование с помощью компьютерной программы на языке Python.