Міністерство освіти і науки України  
НТУУ «КПІ ім. Ігоря Сікорського»  
Кафедра диференціальних рівнянь

Лабораторна робота №10  
Тема: “Числове інтегрування”  
Варіант № 6

Виконав студент 2-го курсу  
ТЕФ, групи ТР-71  
Зуєв Михайло Олександрович

Київ – 2018

# Код програми:

""" Лабораторна работа номер 10

з курсу Чисельні методи, варіант 6

Завдання: Наближенно обчислити значення визначеного інтеграла з точністю ε = 0.001

за допомогою подвійного перерахунку, узявше початкове значення L = 2.

Використовувати метод Ньютона-Котеса з n = 7. Підінтегральна функція:

f(x) = x / [(3\*x^2 + x + 1) \* sqrt(3\*x^2 + x + 1)], на відрізку[0;2]

Виконав студент 2 курсу: Зуєв Михайло Олександрович

"""

import math

import numpy as np

import matplotlib.pyplot as plt

def function(arg):

"""

Считает значение подинтегральной функции f(x)

:param arg: аргумент функции

:return: значение подинтегральной функции в точке arg

"""

tmp = 3 \* arg \* arg + arg + 1

return arg / (tmp \* math.sqrt(tmp))

def original\_function(arg):

"""

Счиатет значение первоначальной функции F(x)

:param arg: аргумент функции

:return: значение первоначальной в точке arf

"""

return -(2 \* arg + 4) / (11 \* math.sqrt(3 \* arg \* arg + arg + 1))

def approximation\_integrate(a, b, l=2):

"""

Вычисление определенного интеграла с помощью метода Ньютона-Котеса

:param a: левая граница интегрирования

:param b: правая граница интегрирования

:param l: количество интервалов

:return: значение определеного интеграла на отрезке [a; b]

"""

# Степень полинома

n = 7

# Табличные значение

r = 7.0 / 17280

p = np.array([751, 3577, 1323, 2989, 2989, 1323, 3577, 751], dtype=np.float64)

# Шаг

dl = float(b - a) / l

# Расстояние между узлами интерполяции

h = dl / n

# Значение определенного интеграла (сума)

integral\_value = 0

# Границы частичного интеграла

left = a

while left <= b:

sum = 0

for i in range(n + 1):

y = function(left + h \* i)

sum += p[i] \* y

integral\_value += r \* h \* sum

left += dl

return integral\_value

def integrate(a, b, eps):

"""

Считает интеграл с помощью двойного перерасчета, что бы |I1 - I2| < eps

:param a: левая граница интегрирования

:param b: правая граница интегрирования

:param eps: точность интегрирования

:return: значение определеного интеграла на отрезке [a; b]

"""

l = 2

cur\_value = approximation\_integrate(a, b, l)

prev\_value = 0

while math.fabs(cur\_value - prev\_value) >= eps:

l \*= 2

prev\_value = cur\_value

cur\_value = approximation\_integrate(a, b, l)

return cur\_value

# Границы интеграрованиия

a = 0

b = 2

# Точность интегрирования

eps = 1e-3

# Точное значение определеного интеграла

exact\_integral\_value = original\_function(b) - original\_function(a)

# Приближенное значение определеного интеграла

integral\_value = integrate(a, b, 0.001)

# Таблица значений функции

print("-------------------")

print("| x | f(x) |")

print("-------------------")

h = (b - a) / 10

x = a

while x < b:

print("|{0:6.2f} |{1:8.3f} |".format(x, function(x)))

x += h

print("-------------------\n")

# Ответы

print("With eps = {0}:".format(eps))

print("\tExact value = {0:6.4f}".format(exact\_integral\_value))

print("\tApproximation value = {0:6.4f}".format(integral\_value))

# Дополнительное задание

#

# Нарисовать графи E(L) = (I\_точ - I\_набл) / I\_набл, для eps = 1e-5

eps = 1e-5

l = 2

cur\_value = approximation\_integrate(a, b, l)

prev\_value = 0

xData = [l]

yData = [abs((exact\_integral\_value - cur\_value) / cur\_value)]

while math.fabs(cur\_value - prev\_value) >= eps:

l \*= 2

prev\_value = cur\_value

cur\_value = approximation\_integrate(a, b, l)

xData.append(l)

yData.append(abs((exact\_integral\_value - cur\_value) / cur\_value))

plt.plot(xData, yData, label='E(L)')

plt.xlabel(r'$L$')

plt.ylabel(r'$E$')

plt.title(r'$E(L) = (I\_точ - I\_набл) / I\_набл$')

plt.legend()

plt.show()

# Результати роботи:

-------------------

| x | f(x) |

-------------------

| 0.00 | 0.000 |

| 0.20 | 0.132 |

| 0.40 | 0.155 |

| 0.60 | 0.137 |

| 0.80 | 0.112 |

| 1.00 | 0.089 |

| 1.20 | 0.072 |

| 1.40 | 0.059 |

| 1.60 | 0.049 |

| 1.80 | 0.041 |

| 2.00 | 0.034 |

-------------------

With eps = 0.001:

Exact value = 0.1759

Approximation value = 0.1764

