Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра кібербезпеки та програмного забезпечення

**Звіт**

Дисципліна: Алгоритми та методи обчислень

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2018

Лабораторная работа № 3.

**Тема**: Интерполяции функций с помощью формулы Лагранжа и многочлену Ньютона

Задание:

1. Найти приближенное значение функции при этом значении аргумента с помощью интерполяционного многочлена Лагранжа, если функция задана:

1.1. в неравноудаленных узлах таблицы;

1.2. в равноудаленных узлах таблицы.

Ход работы

Задание 1:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Y | 1,80866 | 0,89492 | 1,02964 | 2,20966 | 1,34087 | 1,52368 |
| X | 0,43 | 0,48 | 0,55 | 0,62 | 0,70 | 0,75 |

x = 0,555

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Задачa №1

def LagrangeFunction(list\_x, list\_y, value):

z=0

for j in range(len(list\_y)):

p1=1; p2=1

for i in range(len(list\_x)):

if i==j:

p1=p1\*1; p2=p2\*1

else:

p1=p1\*(value-list\_x[i])

p2=p2\*(list\_x[j]-list\_x[i])

z=z+list\_y[j]\*p1/p2

return z

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Вариан выполнения

variant\_x = [0.43, 0.48, 0.55, 0.62, 0.70, 0.75]

variant\_y = [1.80866, 0.89492, 1.02964, 2.20966, 1.34087, 1.52368]

variant\_value = 0.555

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Образец выполнения

example\_x = [0.05, 0.10, 0.17, 0.25, 0.30, 0.36]

example\_y = [0.050042, 0.100335, 0.171657, 0.255342, 0.309336, 0.376403]

example\_value = 0.263

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

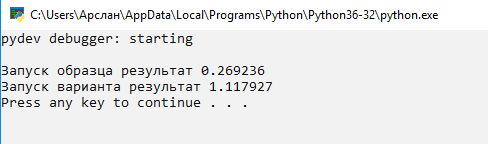
#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Вызов функций

print("Запуск образца результат %f" % (LagrangeFunction(example\_x, example\_y, example\_value)))

print("Запуск варианта результат %f" % (LagrangeFunction(variant\_x, variant\_y, variant\_value)))

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Y | 5,04192 | 5,17744 | 5,32016 | 5,47069 | 5,62968 | 5,79788 |
| X | 1,375 | 1,380 | 1,385 | 1.390 | 1,395 | 1,400 |

x = 1,3934

from math import \*

from functools import reduce

from operator import mul

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Задача №2

def LagrangeFunction(list\_x, list\_y, value):

lenx = len(list\_x)

leny = len(list\_y)

if lenx != leny and lenx < 2:

return -1

temp = (value - list\_x[0]) / (list\_x[1] - list\_x[0])

koefficient = (reduce(mul, [temp - i for i in range(lenx)]))

total = 0

for i in range(lenx):

result = (pow(-1, lenx-1 - i) \* factorial(i) \* factorial(lenx-1 - i))

total += list\_y[i] / ((temp - i) \* result)

return koefficient \* total

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Вариан выполнения

variant\_x = [1.375, 1.380, 1.385, 1.390, 1.395, 1.400]

variant\_y= [5.04192, 5.17744, 5.32016, 5.47069, 5.62968, 5.79788]

variant\_value = 1.3934

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Образец выполнения

example\_x = [0.101, 0.106, 0.111, 0.116, 0.121, 0.126]

example\_y = [1.26183, 1.27644, 1.29122, 1.30617, 1.32130, 1.32660]

example\_value = 0.1157

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

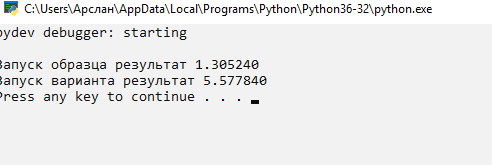
#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Вызов функций

print("Запуск образца результат %f" % (LagrangeFunction(example\_x, example\_y, example\_value)))

print("Запуск варианта результат %f" % (LagrangeFunction(variant\_x, variant\_y, variant\_value)))

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



Задание 2

Вычислить значение функции при заданных значениях аргумента, используя интерполяционную формулу Ньютона для неравноудаленных узлов. При вычислениях учитывать только разделены разницы первого и второго порядков. Вычисление о вести дважды используя, если это возможно, различные узлы

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Y | 3,25578 | 3,17639 | 3,12180 | 3,04819 | 2,98755 | 2,91950 | 2,83598 |
| X | 0,298 | 0,303 | 0,310 | 0,317 | 0,323 | 0,330 | 0,339 |

x1 = 0,312

x2 = 0,304

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Разница первого порядка

def firstLevel(x0, x1, y0, y1):

return (y1 - y0)/(x1 - x0)

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Разница второго порядка

def secondLevel(x0, x1, x2, y0, y1, y2):

return (firstLevel(x1, x2, y1, y2) - firstLevel(x0, x1, y0, y1)) / (x2 - x0)

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Формула Ньютона

def NewtonFunction(x, list\_x, list\_y):

lenx = len(list\_x)

leny = len(list\_y)

if lenx != leny and lenx < 3:

return None

return [list\_y[i] +

(firstLevel(list\_x[i], list\_x[i+1], list\_y[i], list\_y[i+1])\* (x-list\_x[i]))+

(secondLevel(list\_x[i], list\_x[i+1], list\_x[i+2], list\_y[i], list\_y[i+1], list\_y[i+2])\*(x-list\_x[i]) \* (x-list\_x[i+1])) for i in range(int(lenx/2))]

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*# Вариант выполнения

variant\_x = [0.298, 0.303, 0.310, 0.317, 0.323, 0.330, 0.339]

variant\_y = [3.25578, 3.17639, 3.12180, 3.04819, 2.98755, 2.91950, 2.83598]

value1 = 0.312

value2 = 0.304

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

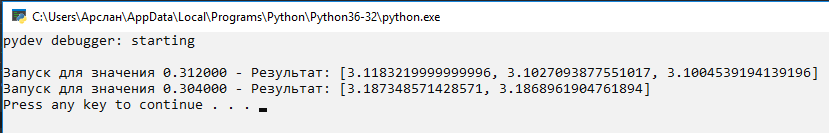
# Вызов функций

result1 = NewtonFunction(value1, variant\_x, variant\_y)

result2 = NewtonFunction(value2, variant\_x[2:], variant\_y[2:])

print("Запуск для значения %f - Результат: %s" % (value1, result1))

print("Запуск для значения %f - Результат: %s" % (value2, result2))

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*