|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № 1**

|  |  |
| --- | --- |
| **Тема:** Линейная интерполяция с помощью полинома Ньютона  **Студент:** Нгуен Ань Тхы  **Группа:** ИУ7 – И46Б  **Оценка (баллы): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель:** Градов В. М. |  |

Москва.

2020 г.

**Входные данные:**

- Таблица с данными (x, y)

- Степень полинома n

- значение x

**Выходные данные:**

- Значение y(x) функции в точке x

- Корень функции заданной таблицы (Найденный методом половинного деления)

- Корень функции (Найденный обратным интерполированием).

**Алгоритм:**

Интерполяционный многочлен Ньютона с разделенными разностями. Пусть функция  задана с произвольным шагом, и точки таблицы значений пронумерованы в произвольном порядке.

Разделенные разности: нулевого порядка совпадают со значениями функции в узлах. Разделенные разности первого порядка определяются через разделенные разности нулевого порядка:



Разделенные разности второго порядка определяются через разделенные разности первого порядка:



Разделенные разности *k*-го порядка определяются через разделенные разности порядка :



Используя понятие разделенной разности интерполяционный многочлен Ньютона можно записать в следующем виде:



За точностью расчета можно следить по убыванию членов суммы *Pn(x)*. Если функция достаточно гладкая, то справедливо приближенное равенство . Это приближенное равенство можно использовать для практической оценки погрешности интерполяции: .

**Тесты:**

Input n: 5

Input x: 2.6

Таблица интерполяции:

0.000 -10.000 -0.000 -0.000 3.000 1.000 -0.000

1.000 -10.000 -0.000 9.000 7.000 1.000

2.000 -10.000 18.000 30.000 11.000

3.000 8.000 78.000 63.000

4.000 86.000 204.000

5.000 290.000

Interpolate: y(2.600) = -3.510

f(2.600) = -3.510

Error: 0.000

root = 2.753

**Код программы:**

from math import sin, cos, ceil, radians

def f(x):

return x\*\*4 - 3 \* x\*\*3 + 2 \* x\*\*2 - 10

def creat\_table(n, x):

step = ceil(n / 2)

table = []

X = []

Y = []

for i in range(-step, n - step):

X.append(x + i)

Y.append(f(x + i))

table.append(X)

table.append(Y)

for i in range(n - 1):

tab = []

Y = table[i + 1]

for k in range(len(Y) - 1):

y\_value = (Y[k] - Y[k + 1]) / (X[k] - X[k + i + 1])

tab.append(y\_value)

table.append(tab)

return table

def print\_table(table):

for index in range(len(table)):

for tab in table:

if len(tab) > index:

print("{:4.3f}".format(tab[index]), end = ' ')

print()

def interpolation(table, x):

yx = table[1][0]

X = table[0]

for i in range(2, len(table)):

Y = table[i]

dx = 1

for k in range(i - 1):

dx \*= (x - X[k])

yx += Y[0] \* dx

return yx

eps = 1e-8

def root(xa, ya, xb, yb):

if abs(xa - xb) < eps:

return xa

xc = (xb + xa) / 2

# yc = (yb - ya) / (xb - xa)

yc = interpolation(table, xc)

if abs(yc) < eps:

return xc

if (yc \* ya < 0):

return root(xa, ya, xc, yc)

return root(xc, yc, xb, yb)

def reverse\_inter(X, Y):

for i in range(len(Y) - 1):

if (Y[i] \* Y[i + 1] < 0):

print("root = {:4.3f}".format(root(X[i], Y[i], X[i + 1], Y[i + 1])))

def input\_n():

try:

n = int(input("Input n: "))

if (n < 0):

print("Error n < 0!")

return input\_n()

else:

return n

except ValueError:

print("Input Error!")

input\_n()

n = input\_n()

x = float(input("Input x: "))

table = creat\_table(n + 1, ceil(x))

print()

print\_table(table)

yx = interpolation(table, x)

print("Interpolate: y({:4.3f}) = {:4.3f}".format(x, yx))

print("f({:4.3f}) = {:4.3f}".format(x, f(x)))

print("Error: {:4.3f}".format(round(yx - f(x)), 3))

reverse\_inter(table[0], table[1])