|  |  |
| --- | --- |
|  | **Министерство образования и науки 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 (ИУ)

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии (ИУ7)

**Лабораторная работа №1**

**Тема:** Алгоритм и программа построения интерполяционных полиномов

Ньютона и Эрмита

**Студент:** АртемьевИ.О.

**Группа:** ИУ7-43Б

**Оценка (баллы): \_\_\_\_\_\_\_\_\_**

**Преподаватель:** ГрадовВ.М.

Москва, 2021 г *.*

# **Цель работы**

Изучить метод нахождения значения функции в заданной точке с помощью интерполяционных полиномов Ньютона и Эрмита.

# **Задание**

1. Найти Pn(x) и Hn(x)
2. Сравнить результаты вычисления значения функции обоими методами
3. Найти корень функции методом обратной интерполяции

# **Входные данные**

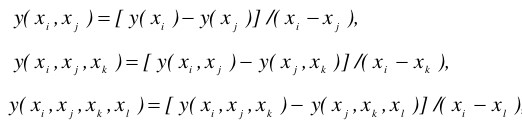
1. Таблица координат
2. Координата точки по оси абсцисс
3. Степень искомых полиномов

# **Выходные данные**

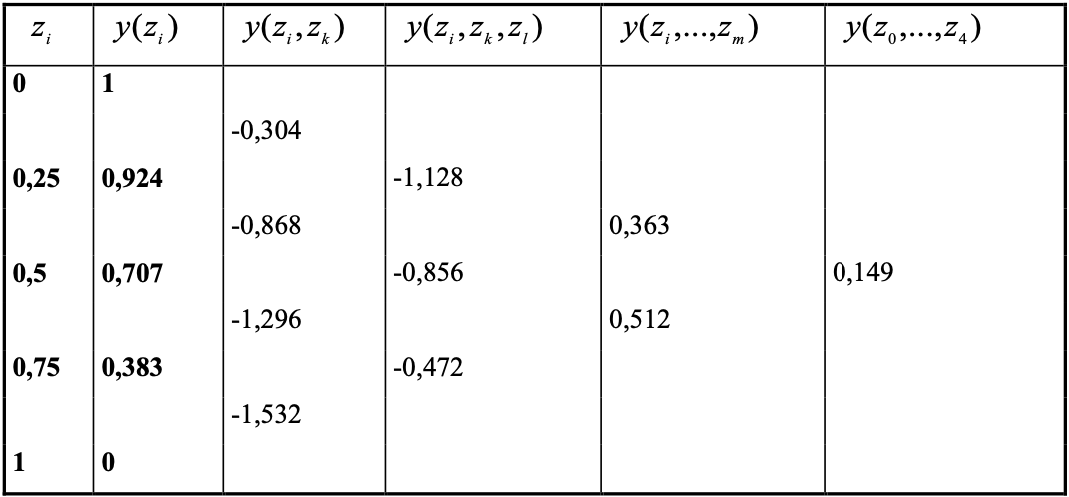
1. Значение функции в точке Х, найденное двумя методами
2. Корень функции, найденный с помощью метода обратной интерполяции

# **Анализ алгоритма**

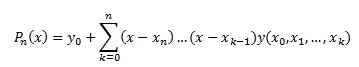
В алгоритме подсчитываются разделенные разности по следующим формулам:



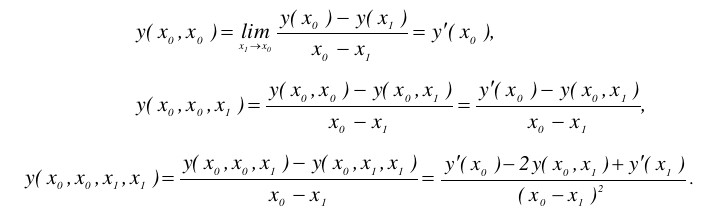
Строится таблица следующего формата:



Далее с помощью этой таблицы подсчитывается полином Ньютона, имеющий формулу:



При нахождении полинома Эрмита, производные функции в точке рассматриваются в качестве предельного перехода от разделенных разностей, следовательно:



Аналогично полиному Ньютона строим таблицу разделенных разностей и учитываем в нужных местах замену 0/0 на производную.

При поиске корня обратной интерполяцией, столбцы меняются местами, а аргумент задается равным 0

# **Исходные данные**

|  |  |  |
| --- | --- | --- |
| x | y | y’ |
| 0.00 | 1.000000 | -1.000000 |
| 0.15 | 0.838771 | -1.14944 |
| 0.30 | 0.655336 | -1.29552 |
| 0.45 | 0.450447 | -1.43497 |
| 0.60 | 0.225336 | -1.56464 |
| 0.75 | -0.018310 | -1.68164 |
| 0.90 | -0.278390 | -1.78333 |
| 1.05 | -0.552430 | -1.86742 |

# **Результаты вычислений**

|  |  |  |  |
| --- | --- | --- | --- |
| Аргумент  x = 0.525 | Значение функции | | Корень |
| Степень полинома | Ньютон | Эрмит |
| 1 | 0.337891 | 0.337891 | 0.738727 |
| 2 | 0.340208 | 0.340288 | 0.739174 |
| 3 | 0.340314 | 0.340323 | 0.739095 |
| 4 | 0.340324 | 0.340324 | 0.739081 |

**Исходный код**

**// Файл myio.py**

from aux\_funcs import table\_args\_convert\_to\_float

def input\_inf():

filename = input("Input the filename: ")

try:

with open(filename) as file:

table = [row.split() for row in file.readlines()]

degree = int(input("Input the degree of the polynomial: "))

arg\_val = float(input("Input the value of the argument: "))

except:

print("Error, check that the data is correct")

exit()

table\_args\_convert\_to\_float(table)

return table, degree, arg\_val

def print\_results(newton, hermite, root):

print("Newton: {:.6f}".format(newton))

print("Hermite: {:.6f}".format(hermite))

print("Root: {:.6f}".format(root))

**// Файл process\_funcs.py**

def get\_interval(table, degree, arg\_val):

for upper\_arg\_ind in range(len(table)):

if table[upper\_arg\_ind][0] > arg\_val: break

lower\_arg\_ind = upper\_arg\_ind

while upper\_arg\_ind - lower\_arg\_ind < degree:

if lower\_arg\_ind > 0: lower\_arg\_ind -= 1

if upper\_arg\_ind - lower\_arg\_ind >= degree: break

if upper\_arg\_ind < len(table) - 1: upper\_arg\_ind += 1

return lower\_arg\_ind, upper\_arg\_ind

def get\_diffs\_table(table, interval):

"""

interval[0] - the lower index of the argument

interval[1] - the upper index of the argument

"""

diffs\_table = [[], []]

deriv\_table = []

for i in range(interval[0], interval[1] + 1):

diffs\_table[0].append(table[i][0])

diffs\_table[1].append(table[i][1])

deriv\_table.append(table[i][2])

for i in range(1, interval[1] - interval[0] + 1):

row = []

for j in range(interval[1] - interval[0] - i + 1):

if diffs\_table[0][j] - diffs\_table[0][j + i] == 0:

row.append(deriv\_table[j])

continue

row.append((diffs\_table[i][j] - diffs\_table[i][j + 1]) / (diffs\_table[0][j] - diffs\_table[0][j + i]))

diffs\_table.append(row)

return diffs\_table

def get\_val\_pol(table, degree, arg\_val):

table.sort()

interval = get\_interval(table, degree, arg\_val)

diffs\_table = get\_diffs\_table(table, interval)

mul = 1

val = diffs\_table[1][0]

for i in range(2, len(diffs\_table)):

mul \*= (arg\_val - diffs\_table[0][i - 2])

val += diffs\_table[i][0] \* mul

return val

**// Файл aux\_funcs.py**

import copy

def table\_args\_convert\_to\_float(table):

for row in table:

for i in range(len(row)):

row[i] = float(row[i])

def swap\_clms(table):

new\_table = copy.deepcopy(table)

for row in new\_table:

row[0], row[1] = row[1], row[0]

return new\_table

def get\_extend\_table(table):

ext\_table = []

for row in table:

ext\_table.append(row)

ext\_table.append(row)

return ext\_table

**// Файл main.py**

from myio import \*

from process\_funcs import \*

from aux\_funcs import get\_extend\_table, swap\_clms

def main():

"""

new\_table - table with rearranged columns for reverse interpolation

"""

inform = input\_inf()

table, degree, arg\_val = inform[0], inform[1], inform[2]

ext\_table = get\_extend\_table(table)

new\_table = swap\_clms(table)

newton = get\_val\_pol(table, degree, arg\_val)

hermite = get\_val\_pol(ext\_table, degree, arg\_val)

root = get\_val\_pol(new\_table, degree, 0)

print\_results(newton, hermite, root)

if \_\_name\_\_ == "\_\_main\_\_":

main()

# **Контрольные в опросы**

# 

1. Будет ли работать программа при степени полинома n = 0?

Будет, так как в качестве полинома будет использоваться свободный член.

1. Как практически оценить погрешность интерполяции?

Почему сложно применить для этих целей теоретическую оценку?

С помощью первого отброшенного члена полинома.

Теоретическую оценку провести тяжело, т.к. для нее нужна производная, которую, часто, невозможно установить

1. Если в двух точках заданы значения функции и ее первых производных, то полином какой минимальной степени может быть построен на этих точках?

Минимальная степень - 0.

Максимальная степень - 3.

1. В каком месте алгоритма построения полинома существенна информация об упорядоченности аргумента функции (возрастает, убывает)?

При построении таблицы разделенных разностей.

1. Что такое выравнивающие переменные и как их применить для повышения точности интерполяции?

Выравнивающие переменные - переменные, в которых график функций близок к прямолинейному.

При повышении точности строят полином Ньютона для этих переменных, а потом вместо них используют x и y.