Министерство образования и науки Российской Федерации Федеральное агентство по образованию

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №3 дисциплины

«Вычислительная математика»

Выполнил студент группы ИВТ-22 /Крючков И. С/ Проверил /Исупов К. С./

Киров 2021

# Задание:

Вариант 10

1. По таблице с неравноотстоящими значениями аргумента выполнить интерполяцию, используя формулу Лагранжа. Точность E<=0,000001.

Задание:

X=0,153

0,10 1,66448

0,20 1,66071

0,29 1,65734

0,40 1,65322

0,49 1,64987

0,55 1,64764

2. По таблице с равноотстоящими значениями аргумента вычислить значения функции для заданных значений аргументов, используя первую и вторую интерполяционные формулы Ньютона. Точность E<=0.000001.

Задание:

X1=0,455; X2=0,5575; X3=0,440; X4=0,5674;

0,45 20,1946

0,46 19,6133

0,47 18,9425

0,48 18,1746

0,49 17,3010

0,50 16,3123

0,51 15,1984

0,52 13,9484

0,53 12,5508

0,54 10,9987

0,55 9,2647

0,56 7,3510

3. По заданным экспериментальным точкам выбрать вид эмпирической зависимости и выполнить среднеквадратичное приближение функции, применив метод наименьших квадратов для оценки параметров выбранной зависимости.

Задание:

3,0 25,0

3,1 29,4

3,2 34,4

3,3 40,4

3,4 47,6

3,5 56,0

4. Проверить результаты с помощью системы Mathcad.

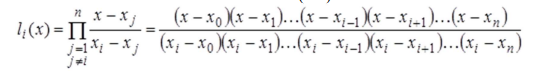
**Теоретические сведения:**

**Формула Лагранжа**

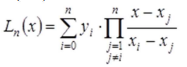
При глобальной интерполяции на всем интервале [a;b] строится единый многочлен. Одной из форм записи интерполяционного многочлена для глобальной интерполяции является многочлен Лагранжа (1.1):



где  – базисные многочлены степени n (1.2):

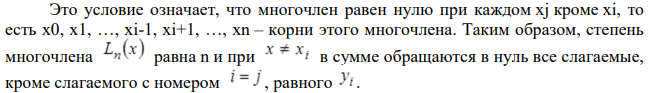


То есть многочлен Лагранжа (1.3):



Многочлен удовлетворяет условию (1.4):





Выражение (1.1) применимо как для равноотстоящих, так и для не равноотстоящих узлов. Погрешность интерполяции методом Лагранжа зависит от свойств функции f(x), от расположения узлов интерполяции и точки x. Полином Лагранжа имеет малую погрешность при небольших значениях n ( n < 20). При больших n погрешность начинает расти, что свидетельствует о том, что метод Лагранжа не сходится (т.е. его погрешность не убывает с ростом n).

Многочлен Лагранжа в явном виде содержит значения функций в узлах интерполяции, поэтому он удобен, когда значения функций меняются, а узлы интерполяции неизменны. Число арифметических операции, необходимых для построения многочлена Лагранжа, пропорционально n2 и является наименьшим для всех форм записи. К недостаткам этой формы записи можно отнести то, что с изменением числа узлов приходится все вычисление проводить заново.

**Формула Ньютона**

Пусть функция f(x) задана с произвольным шагом и точки таблицы значений занумерованы в произвольном порядке. Разделенные разности нулевого порядка совпадают со значениями функции в узлах. Разделенные разности первого порядка определяются через разделенные разности нулевого порядка (2.1):

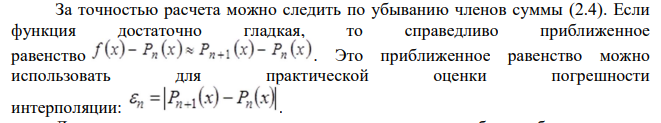


Разделенные разности второго порядка определяются через разделенные разности первого порядка (2.2):

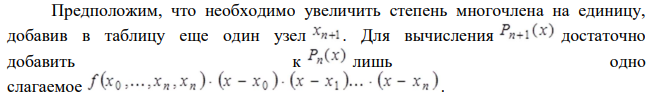


Разделенные разности k-го порядка определяются через разделенную разность порядка k-1 (2.3):

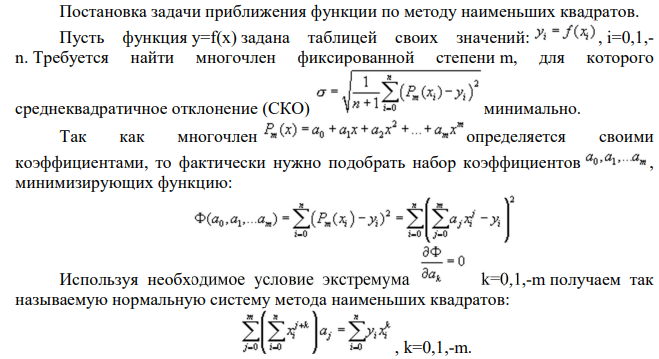




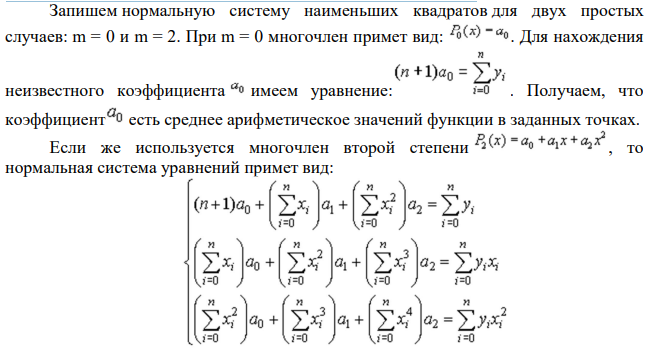
Для повышения точности интерполяции в сумму могут быть добавлены новые члены, что требует подключения дополнительных узлов. При этом для формулы Ньютона безразлично, в каком порядке подключаются новые узлы, в то время как для формулы Лагранжа при добавлении новых узлов все расчеты надо производить заново.



**Среднеквадратичное приближение**



Полученная система есть система алгебраических уравнений относительно неизвестных . Можно показать, что определитель этой системы отличен от нуля, то есть решение существует и единственно. Однако при высоких степенях m система является плохо обусловленной. Поэтому метод наименьших квадратов применяют для нахождения многочленов, степень которых не выше 5. Решение нормальной системы можно найти, например, методом Гаусса.



**Практическая часть**

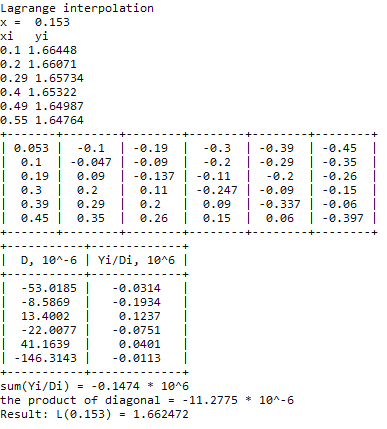
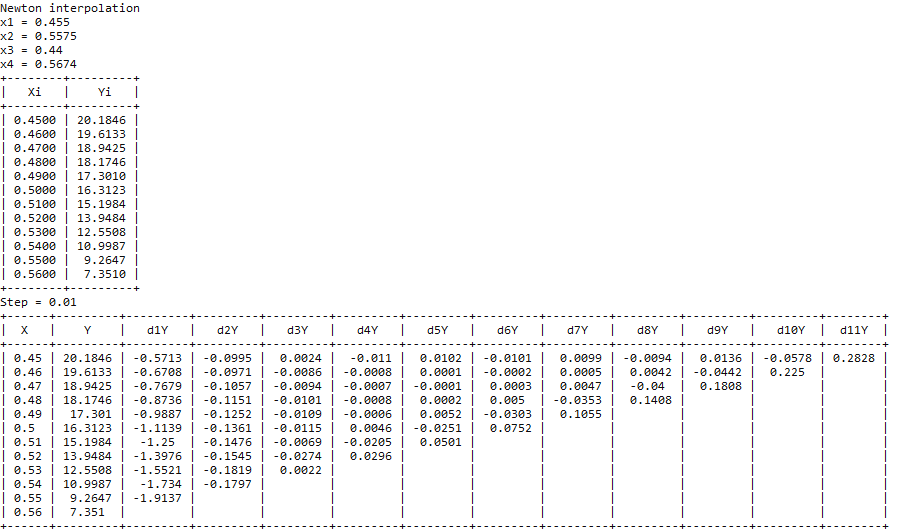
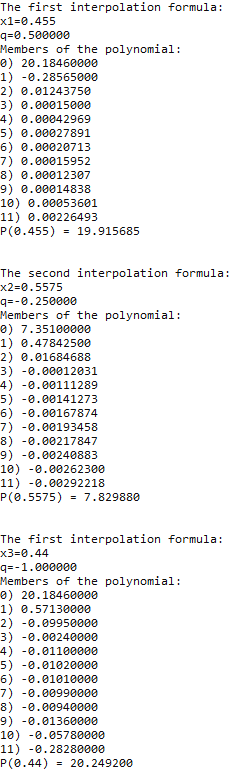


Рис. 1 – Результат выполнения первого задания





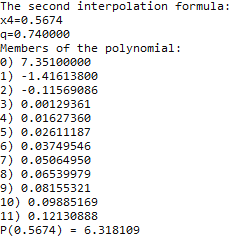


Рис. 2 – Результат выполнения второго задания

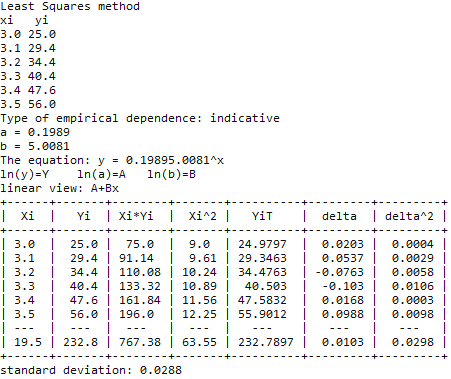


Рис. 3 – Результат выполнения третьего задания

**Листинг кода:**

def lagrange\_interpolation(x, func, eps):

xs, ys = func

n = len(xs)

ls = [reduce(lambda a, b: a \* b, ((x-xs[j])/(xs[i]-xs[j]) for j in range(n) if i != j)) for i in range(n)]

result = sum(l \* y for l, y in zip(ls, ys))

return round(result, int(-log10(eps)))

def getQ(x, x0, step):

return (x-x0)/step

def newton\_member(q, dy, n, y0, t):

# print(q, dy, n, y0, t)

tmp = q

if n > 1:

for i in range(2, n+1):

if t == 1:

tmp \*= (q-i+1)

else:

tmp \*= (q+i-1)

tmp = (tmp/factorial(n))\*dy

elif n == 1:

tmp \*= dy

elif n == 0:

tmp = y0

return tmp

def lin(a, b, x):

return exp(log(a)+log(b)\*x)

def find\_b(x, y):

l = len(x)

sumxlny = 0

sumx = 0

sumlny = 0

sumx2 = 0

for i in range(l):

sumxlny += x[i] \* log(y[i])

sumx += x[i]

sumlny += log(y[i])

sumx2 += x[i]\*x[i]

return exp((l\*sumxlny - sumx\*sumlny)/(l\*sumx2-sumx\*sumx))

def find\_a(x,y, b):

l = len(x)

sumx = 0

sumlny = 0

for i in range(l):

sumx += x[i]

sumlny += log(y[i])

return exp(1/l\*sumlny - (log(b)/l)\*sumx)

def print\_squares(x, y):

tablex\_s = PrettyTable()

tablex\_s.float\_format = '.4'

tablex\_s.field\_names = ["Xi", "Yi", "Xi\*Yi", "Xi^2", "YiT", "delta", "delta^2"]

l = len(x)

b = find\_b(x, y)

a = find\_a(x, y, b)

ar= round(a,4)

br= round(b,4)

print('xi yi')

for i in range(l):

print(f'{x[i]} {y[i]}')

print('Type of empirical dependence: indicative')

print(f'a = {ar}')

print(f'b = {br}')

print(f'The equation: y = {ar}{br}^x')

print('ln(y)=Y ln(a)=A ln(b)=B')

print(f'linear view: A+Bx')

s1 = 0

s2 = 0

s3 = 0

s4 = 0

s5 = 0

s6 = 0

s7 = 0

for i in range(l):

row = []

row.append(x[i])

row.append(y[i])

row.append(x[i]\*y[i])

row.append(round(x[i]\*\*2, 4))

row.append(round(lin(a,b,x[i]),4))

row.append(round(y[i]-lin(a,b,x[i]), 4))

row.append(round((y[i]-lin(a,b,x[i]))\*\*2, 4))

s1 += row[0]

s2 += row[1]

s3 += row[2]

s4 += row[3]

s5 += row[4]

s6 += row[5]

s7 += row[6]

tablex\_s.add\_row(row)

tablex\_s.add\_row(['---']\*7)

tablex\_s.add\_row([round(s1, 4), round(s2, 4), round(s3, 4), round(s4, 4), round(s5, 4), round(s6, 4), round(s7, 4)])

print(tablex\_s)

dev = round(sqrt(s7)/l, 4)

print(f'standard deviation: {dev}')

**Вывод:** в ходе данной лабораторной работы были подробно изучены методы интерполирования, а именно: интерполяция по формулам Лагранжа и Ньютона. Также было изучено среднеквадратичное приближение функции с применением метода наименьших квадратов. После изучения материала по данным темам, была написана программа, в которой реализовано решение заданий, предоставленных на лабораторную работу, при помощи соответствующих методов.