Федеральное государственное автономное образовательное учреждение высшего образования **«Национальный исследовательский университет ИТМО»**

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа **№4**

**«**АППРОКСИМАЦИЯ ФУНКЦИИ МЕТОДОМ

НАИМЕНЬШИХ КВАДРАТОВ**»**

по дисциплине «Вычислительная математика**»**

Вариант: **18**

**Преподаватель:**

**Выполнил:**

Шаматульский Роман Константинович

**Группа:** Р3212

Санкт-Петербург, 2025 г.

Оглавление

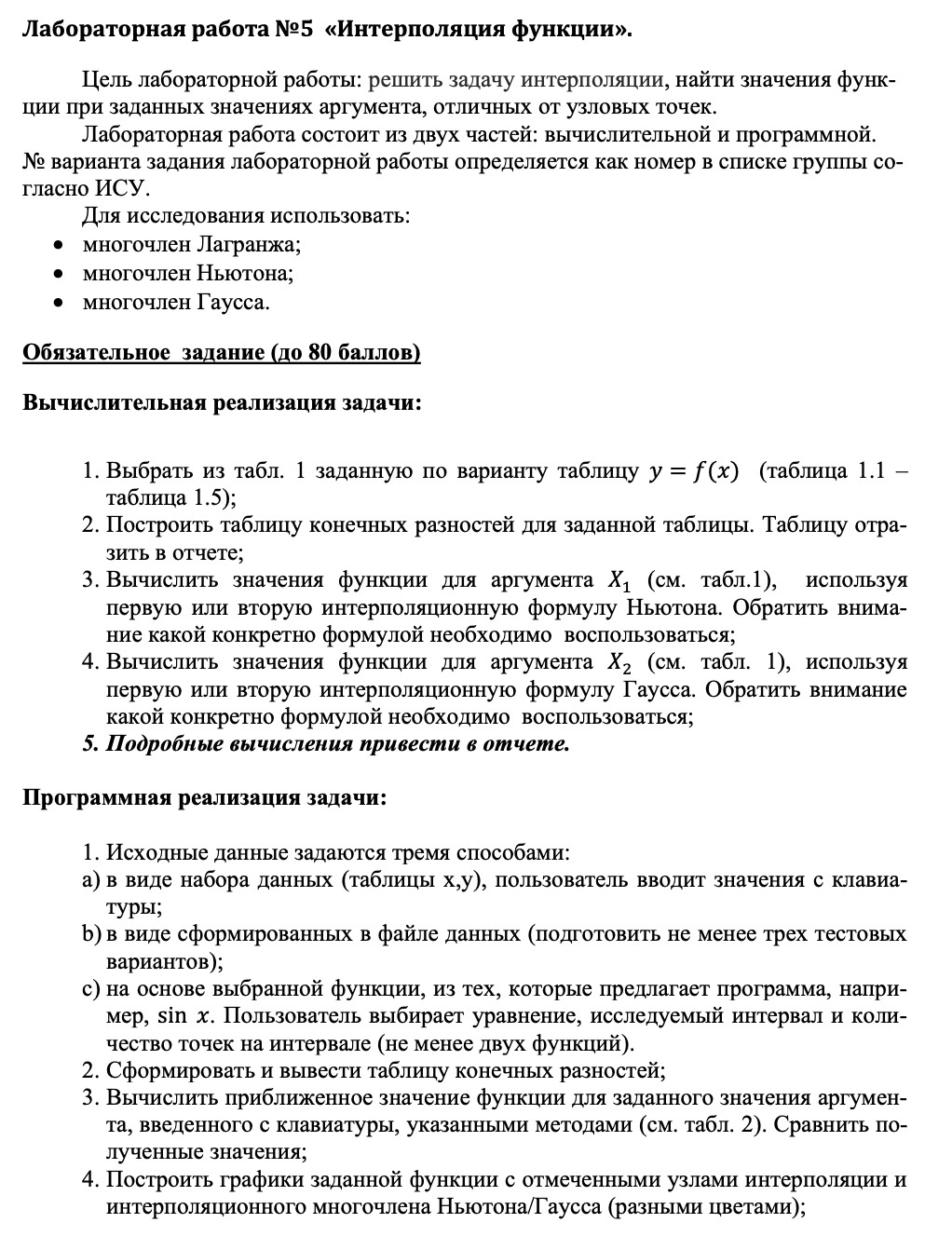
[Текст задания 3](#_Toc196928063)

[Вычислительная часть 4](#_Toc196928064)

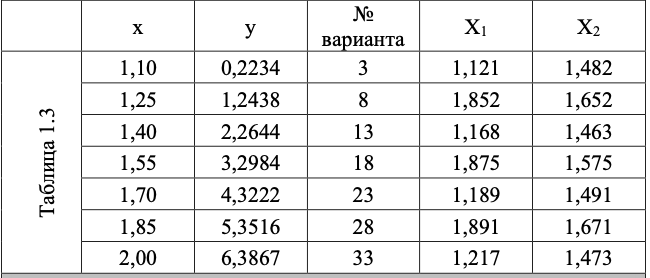
[Программная часть 6](#_Toc196928065)

[Вывод 10](#_Toc196928066)

# Текст задания



# Вычислительная часть



**Таблица конечных разностей**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x | y |  |  |  |  |  |  |
| 1,10 | 0,2234 | 1,0204 | 0,0002 | 0.0132 | -0.0368 | 0.0762 | -0.1313 |
| 1,25 | 1,2438 | 1,0206 | 0,0134 | -0.0236 | 0.0394 | -0.0551 |  |
| 1,40 | 2,2644 | 1,0340 | -0.0102 | 0.0158 | -0.0157 |  |  |
| 1,55 | 3,2984 | 1,0238 | 0.0056 | 0.0001 |  |  |  |
| 1,70 | 4,3222 | 1,0294 | 0.0057 |  |  |  |  |
| 1,85 | 5,3516 | 1,0351 |  |  |  |  |  |
| 2,00 | 6,3867 |  |  |  |  |  |  |

**Вычисление значения функции для X1:**

Так как h=0,15

И , что ближе к , чем к , так как

*,* и

Мы берем вторую интреполяционную формулу, так как она считает с конца, а не с начала.

И строим полином:

**Вычисление значения функции для X2:**

Нам нужно оценить насколько близко к центру таблицы () находится :

Поскольку это маленькое по модулю число, то наше лежит правее центрального элемента и мы будем использовать первую формулу Гаусса.

# Программная часть

import sys, math, numpy as np  
import matplotlib.pyplot as plt  
  
def finite\_diffs(y):  
 table=[y.copy()]  
 while len(table[-1])>1:  
 table.append([table[-1][i+1]-table[-1][i] for i in range(len(table[-1])-1)])  
 return table  
  
def lagrange(xv,yv,x):  
 s=0.0  
 for i in range(len(xv)):  
 p=1.0  
 for j in range(len(xv)):  
 if i!=j:  
 p\*=(x-xv[j])/(xv[i]-xv[j])  
 s+=yv[i]\*p  
 return s  
  
def newton\_divided(xv,yv,x):  
 div=yv.copy()  
 n=len(xv)  
 for k in range(1,n):  
 for i in range(n-1,k-1,-1):  
 div[i]=(div[i]-div[i-1])/(xv[i]-xv[i-k])  
 s=div[-1]  
 for k in range(n-2,-1,-1):  
 s=s\*(x-xv[k])+div[k]  
 return s  
  
def newton\_fd(xv,yv,x):  
 h=xv[1]-xv[0]  
 table=finite\_diffs(yv)  
 n=len(xv)  
 if abs(x-xv[0])<abs(x-xv[-1]):  
 t=(x-xv[0])/h  
 s,fact=table[0][0],1.0  
 for k in range(1,n):  
 fact\*=(t-(k-1))/k  
 s+=fact\*table[k][0]  
 else:  
 t=(x-xv[-1])/h  
 s,fact=table[0][-1],1.0  
 for k in range(1,n):  
 fact\*=(t+(k-1))/k  
 s+=fact\*table[k][-1]  
 return s  
  
def read\_points():  
 m=int(input("Количество точек: "))  
 pts=[tuple(map(float,input("x y: ").split())) for \_ in range(m)]  
 pts.sort()  
 xv,yv=zip(\*pts)  
 return list(xv),list(yv)  
  
def func\_choice():  
 print("Доступные функции:\ny = sin(x) -- sin\ny = cos(x) -- cos\ny = exp(x) -- exp")  
 fx=input("Выберите функцию: ").strip()  
 a,b=map(float,input("Интервал a b: ").split())  
 n=int(input("Число точек: "))  
 xv=[a+i\*(b-a)/(n-1) for i in range(n)]  
 if fx=="sin":  
 yv=[math.sin(v) for v in xv]  
 elif fx=="cos":  
 yv=[math.cos(v) for v in xv]  
 else:  
 yv=[math.exp(v) for v in xv]  
 return xv,yv  
  
def load\_file(path):  
 with open(path,"r",encoding="utf-8") as f:  
 lines=[l.strip() for l in f if l.strip()]  
 try:  
 count=int(lines[0])  
 data=lines[1:]  
 except ValueError:  
 data=lines  
 xv,yv=[],[]  
 for l in data:  
 parts=l.split()  
 if len(parts)>=2:  
 a,b=map(float,parts[:2])  
 xv.append(a)  
 yv.append(b)  
 return xv,yv  
  
def plot\_method(title,xv,yv,ys\_func):  
 xs=np.linspace(min(xv),max(xv),400)  
 plt.figure()  
 plt.plot(xs,[ys\_func(xx) for xx in xs])  
 plt.scatter(xv,yv,color="black",zorder=5)  
 plt.title(title)  
 plt.xlabel("x")  
 plt.ylabel("y")  
 plt.grid(True)  
  
  
def main():  
 if input("Ввод из файла? (y/n): ").lower().startswith("y"):  
 xv,yv=load\_file(input("Файл: "))  
 else:  
 if input("Сгенерировать по функции? (y/n): ").lower().startswith("y"):  
 xv,yv=func\_choice()  
 else:  
 xv,yv=read\_points()  
 steps=[round(xv[i+1]-xv[i],10) for i in range(len(xv)-1)]  
 if len(set(steps))!=1:  
 print("Шаг по x должен быть равномерным")  
 sys.exit()  
 table=finite\_diffs(yv)  
 print("Таблица конечных разностей:")  
 for i in range(len(xv)):  
 row=[f"{xv[i]:g}",f"{yv[i]:g}"]  
 for k in range(1,len(table)):  
 if i<len(table[k]):  
 row.append(f"{table[k][i]:g}")  
 print("\t".join(row))  
 xq=float(input("Введите X для интерполяции: "))  
 print("Лагранж:",lagrange(xv,yv,xq))  
 print("Ньютон разделённые:",newton\_divided(xv,yv,xq))  
 print("Ньютон конечные:",newton\_fd(xv,yv,xq))  
  
 plot\_method("Lagrange interpolation",xv,yv,lambda x: lagrange(xv,yv,x))  
 plot\_method("Newton (divided) interpolation",xv,yv,lambda x: newton\_divided(xv,yv,x))  
 plot\_method("Newton (finite diffs) interpolation",xv,yv,lambda x: newton\_fd(xv,yv,x))  
 plt.show()  
  
if \_\_name\_\_=="\_\_main\_\_":  
 main()

**Результат выполнения**

Ввод из файла? (y/n): n

Сгенерировать по функции? (y/n): y

Доступные функции:

y = sin(x) -- sin

y = cos(x) -- cos

y = exp(x) -- exp

Выберите функцию: cos

Интервал a b: 5 7

Число точек: 5

Таблица конечных разностей:

5 0.283662 0.425008 -0.173507 -0.0615761 0.0575566

5.5 0.70867 0.251501 -0.235083 -0.00401954

6 0.96017 0.0164173 -0.239103

6.5 0.976588 -0.222685

7 0.753902

Введите X для интерполяции: 5.2

Лагранж: 0.4681508470940356

Ньютон разделённые: 0.46815084709403565

Ньютон конечные: 0.4681508470940356

# Вывод

В лабораторной работе были реализованы три метода интерполяции: Лагранжа, Ньютона с разделёнными и с конечными разностями. Полученные знания помогут мне в изучении математики в будущем.