МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное Государственное Автономное Образовательное Учреждение Высшего Образования "Национальный Исследовательский Университет ИТМО"

##### ФАКУЛЬТЕТ ПИиКТ

## ЛАБОРАТОРНАЯ РАБОТА №5

### по дисциплине

### «ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА»

### Вариант № 7

##### ***Выполнил:*** Студент группы P3219 Зайцев Артём Михайлович

#### Преподаватель:

##### Бострикова Дарья

##### Константиновна

Санкт-Петербург, 2024

Содержание

[Цель 3](#_Toc28668)

[Задание 4](#_Toc11805)

[Вычислительная реализация задачи 5](#_Toc23730)

[Исходный код программы и пример работы 7](#_Toc14197)

[Вывод: 8](#_Toc20308)

# Цель

Решить задачу интерполяции, найти значения функции при заданных значениях аргумента, отличных от узловых точек.

# Задание

## Обязательное задание

### Вычислительная реализация задачи:

1. Выбрать из табл. 1 заданную по варианту таблицу y = f(x) (таблица 1.1 – таблица 1.5);

2. Построить таблицу конечных разностей для заданной таблицы. Таблицу отразить в отчете;

3. Вычислить значения функции для аргумента X1 (см. табл.1), используя

первую или вторую интерполяционную формулу Ньютона. Обратить внимание какой конкретно формулой необходимо воспользоваться;

4. Вычислить значения функции для аргумента X2 (см. табл. 1), используя первую или вторую интерполяционную формулу Гаусса. Обратить внимание какой конкретно формулой необходимо воспользоваться;

### Программная реализация задачи

1. Исходные данные задаются тремя способами:

a) в виде набора данных (таблицы x,y), пользователь вводит значения с клавиатуры;

b) в виде сформированных в файле данных (подготовить не менее трех тестовых вариантов);

c) на основе выбранной функции, из тех, которые предлагает программа, например, sin. Пользователь выбирает уравнение, исследуемый интервал и количество точек на интервале (не менее двух функций).

2. Сформировать и вывести таблицу конечных разностей;

3. Вычислить приближенное значение функции для заданного значения аргумента, введенного с клавиатуры, указанными методами (см. табл. 2). Сравнить полученные значения;

4. Построить графики заданной функции с отмеченными узлами интерполяции и интерполяционного многочлена Ньютона/Гаусса (разными цветами);

# Вычислительная реализация задачи

По моему варианту предлагается построить интерполяционные многочлены по следующим данным

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x | 0,5 | 0,55 | 0,6 | 0,65 | 0,7 | 0,75 | 0,8 |
| y | 1,532 | 2,5356 | 3,5406 | 4,5462 | 5,5504 | 6,5559 | 7,5594 |

Затем посчитаем конечные разности и построим талибцу

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| xi | d0yi | d1yi | d2yi | d3yi | d4yi | d5yi | d6yi | d7yi |
| x1 | 1 | 0.250 | 0.500 | 0 | 0 | 0 | 0 |  |
| x2 | 1.250 | 0.750 | 0.500 | 0 | 0 | 0 |  |  |
| x3 | 2 | 1.250 | 0.500 | 0 | 0 |  |  |  |
| x4 | 3.250 | 1.750 | 0.500 | 0 |  |  |  |  |
| x5 | 5 | 2.250 | 0.500 |  |  |  |  |  |
| x6 | 7.250 | 2.750 |  |  |  |  |  |  |
| x7 | 10 |  |  |  |  |  |  |  |

Найдём приближенное значение функции при заданных X

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| X1 | X2 | X3 | X4 | X5 | X6 | X7 |
| 0,502 | 0,751 | 0,523 | 0,761 | 0,545 | 0,783 | 0,557 |

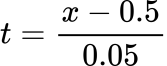
Видно, что все они находятся в левой половине отрезков -> воспользуется 1 формулой Ньютона

Вычислим интерполяционный многочлен

wps

wps

wps

При 

Получается

wps

wps

wps

Теперь посчитаем заданные X

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| X | 0,502 | 0,751 | 0,523 | 0,761 | 0,545 | 0,783 | 0,557 |
| Y | 1.572 | 6.576 | 1.994 | 6.777 | 2.435 | 7.219 | 2.6761 |

Затем найдём приближённые значения X с помощью формулы Гаусса

Напишем формулу интерполяционного многочлена Гаусса

Давайте за A возьмём 0.7, чтобы все значения были меньше А.

Тогда C:/Users/zam12/AppData/Local/Temp/wps.KmPzZAwps

Значит сам многочлен будет иметь следующий вид

C:/Users/zam12/AppData/Local/Temp/wps.abMBYpwps

wps

wps

wps

wps

Подсчитаем его значение в каждой заданной точке:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| X | 0,645 | 0,651 | 0,639 | 0,661 | 0,627 | 0,683 | 0,641 |
| Y | 3.44 | 3.56 | 3.32 | 3.76 | 3.07 | 4.2 | 3.35 |

# Исходный код программы

def do\_linear(x, y, pow=1):  
 matrix = list()  
 b = list()  
  
 for i in range(pow + 1):  
 matrix.append(  
 [pow\_sum(x, i + j) for j in range(pow + 1)]) *# [pow\_sum(x, i), pow\_sum(x, i + 1), pow\_sum(x, i + 2)]* b.append(sum([x\_i \*\* i \* y\_i for x\_i, y\_i in zip(x, y)]))  
  
 A = slae\_solve(np.array(matrix), np.array(b))  
  
 func = lambda x: count\_polinom(x, A)  
  
 new\_y = [func(i) for i in x]  
 S = sum([(res - target) \*\* 2 for res, target in zip(new\_y, y)])  
 return [x, new\_y], S, A  
  
  
def do\_tetra(x, y):  
 print('tetra')  
 table, s, a = do\_linear(x, y, 3)  
 return table, s, sum([X \*\* i \* a[i] for i in range(4)])  
  
  
def do\_quadratic(x, y):  
 print('quadratic')  
 table, s, a = do\_linear(x, y, 2)  
 return table, s, sum([X \*\* i \* a[i] for i in range(3)])  
  
  
def do\_line(x, y):  
 print('line')  
 table, s, a = do\_linear(x, y)  
 return table, s, sum([X \*\* i \* a[i] for i in range(2)]), pirson\_coef(x, y)  
  
  
def do\_exp(x, y): *# a\*e^(bx)* print('exp')  
 y = np.log(y)  
 table, s, a = do\_linear(x, y)  
 a[0] = np.exp(a[0])  
 return table, s, a[0] \* exp(X \* a[1])  
  
  
def do\_pow(x, y): *# a\*x^b* print('pow')  
 x = np.log(x)  
 y = np.log(y)  
 table, s, a = do\_linear(x, y)  
 a[0] = np.exp(a[0])  
 return table, s, a[0] \* X \*\* a[1]  
  
  
def do\_log(x, y): *# a\*ln(x) + b* print('log')  
 x = np.log(x)  
 table, s, a = do\_linear(x, y)  
  
 return table, s, a[1] \* log(X) + a[0]

# Вывод:

После выполнения данной лабораторной работы мы изучили численные методы аппроксимации. Идея приближения заключается в нахождении коэффициентов заданной функции так, чтобы она наилучшим образом описывала входные данные.

Мы можем либо сами из графического представления точек понять, какой класс функций использовать, либо исследовать все имеющиеся классы функций и выбрать ту, у которой меньше СКО.

Сравнение функций без данных не имеет смысла, тк каждая из функций хорошо описывает разные данные.