**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

Крагель Алина Олеговна

**Приближение функции с помощью интерполяционного полинома Лагранжа**

Отчет по лабораторной работе №1

(«Методы численного анализа»)

Студента 2 курса 10 группы

Работа сдана \_\_\_марта 2021 г. Преподаватель:

Зачтена \_\_\_\_\_\_\_\_\_ 2020 г. Никифоров Иван Васильевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Доцент кафедры

(Подпись преподавателя) вычислительной математики

Минск 2020

**Постановка задачи**

Для функции з) на интервале [-3, 0] с шагом 0.25 построить интерполяционный многочлен Лагранжа.

**Теория**

Пусть ­– набор различных точек (узлов) на отрезке *[a, b]*, в которых заданы значения функции *f(x)* так, что . Требуется построить многочлен, принимающий в точках значения , и оценить погрешность приближения достаточно гладкой функции этим многочленом на всем отрезке *[a, b].*

Приведем в явном виде вспомогательные многочлены степени *n – 1*, удовлетворяющие условиям , при :

.

Далее с их помощью запишем формулу для многочлена Лагранжа:

.

Существование и единственность многочлена *n – 1*, принимающего в n различных точках заданные значения, следует из отличия от нуля соответствующего определителя Вандермонда; поэтому указанный многочлен есть решение поставленной задачи.

**Листинг**

import numpy as np  
import matplotlib.pyplot as plt  
import math  
import functools  
  
  
a = -3  
b = 0  
h = 0.25  
n = int((b - a) / h - 1)  
fun = lambda x\_for\_fun: 2 \*\* (x\_for\_fun - 1)  
x = [i for i in np.arange(a, b + h, h)]  
y = [fun(i) for i in [i for i in np.arange(a, b + h, h)]]  
interpolate\_fun = lambda value:sum([y[j] \* functools.reduce(lambda \_x, \_y: \_x \* \_y, [(value - x[i]) / (x[j] - x[i]) if i != j else 1 for i in range(len(x))]) for j in range(len(x))])  
norm1 = max([abs(fun(i) - interpolate\_fun(i)) for i in np.arange(a, b, 0.001)])  
norm2 = 1 / 4 \* h \*\* (n + 1) \* math.factorial(n)  
  
sp = plt.subplot()  
sp.plot(np.linspace(-4, 10, 100), np.vectorize(fun)(np.linspace(-4, 10, 100)), label="f(x)", color="blue")  
sp.plot(np.linspace(-4, 10, 100), np.vectorize(interpolate\_fun)(np.linspace(-4, 10, 100)), label="Ln(x)", color="purple")  
sp.plot([i for i in np.arange(a, b + h, h)], np.vectorize(fun)([i for i in np.arange(a, b + h, h)]), 'go', label="Узлы сетки")  
plt.title("f(x)=2^(x-1)\nПогрешность оценки: " + str(norm2) + "\nПогрешность на [-3,0]:" + str(norm1))  
plt.grid()  
sp.legend()  
plt.show()

**Результаты**

****

**Вывод**

Таким образом, в результате поставленного эксперимента удалось интерполировать данную на интервале с детермированным шагом функцию, совпадающую на этом интервале с точным представлением функции с некоторой погрешностью на интервале.