**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

Факультет физико-математических и естественных наук

ОТЧЕТ

по лабораторной работе № 1

дисциплина: Вычислительные методы

Студент:

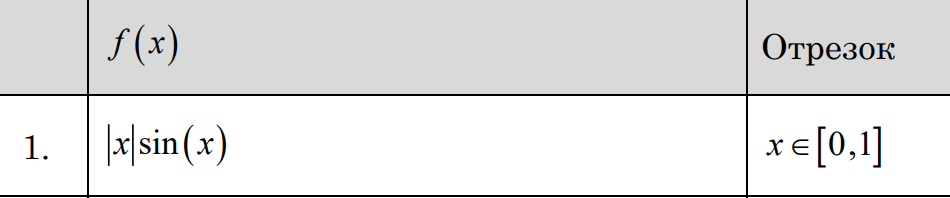
Группа:

**МОСКВА**

2022 г.

**Цел работу:** чтобыузнать об интерполяционном методе Ларанжа, научиться использовать метод и программировать алгоритм на любом языке программирования.

**Мой вариант:**

****

**Ход работы:**

mathдля расчета cos, sin и других значений.

numpy для правильного вычисления абсолютного значения.

import math

import numpy as np

затем я поставил диапазон x в массиве, чтобы вычислить его длину и сэкономить место.

# change range here:

ranges = [0,1];

# lenght of data:

data\_len = len(ranges);

вычисляя значение f (x), которое является точками y, я использовал цикл for для каждого диапазона x, который нужно вычислить по заданной нам функции, затем я добавляю его к созданному мной массиву с именем y.

# f(x) value:

y = [];

# find f(x) value:

for i in range(data\_len):

    number = np.abs(ranges[i])\*math.sin(ranges[i]);

    y.append(number);

для получения точки интерполяции использовалась формула получения точки интерполяции, как показано ниже. Затем верните его в переменную с именем ix\_point, чтобы использовать его дальше в алгоритме.

# get point of interpolation:

*def* get\_interpolation\_point(*x*,*x2*,*y*,*y2*):

    int\_point = *y* + ((*x*-*x2*)/(*x2*-*x*)) \* (*y2* - *y*);

    return int\_point;

# set point of interpolation:

ix\_point = get\_interpolation\_point(ranges[0],ranges[1],y[0],y[1]);

установка интерполированного значения 0 по умолчанию.

# interpolated value:

iy\_point = 0;

для алгоритма, который я использовал 2 для цикла, один для вычисления интерполированного значения и один для вычисления точки интерполяции, интерполированное значение вычисляется после того, как каждый диапазон x вычисляется внутренним циклом for.

# lagrange algorith:

for i in range(data\_len):

    p = 1;

    for j in range(data\_len):

        if i != j:

            p = p \* (ix\_point - ranges[j])/(ranges[i] - x[j]);

    iy\_point = iy\_point + p \* y[i];

наконец, я распечатываю данные, которые я нашел на консоли.

# printout the point value:

print(*f*"Interpolated value at {ix\_point} is {iy\_point}");

**Вывод:**

**Interpolated value at -0.8414709848078965 is -0.7080734182735712**