Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”

Факультет ПИиКТ

ITMO UNIVERSITY

ОТЧЁТ

По лабораторной работе № 4

По предмету: Вычислительная математика

Вариант: Метод Ньютона

Студент:

Андрейченко Леонид Вадимович

Группа P3230

Преподаватель:

Перл Ольга Вячеславовна

Санкт – Петербург

2022

Описание метода

Интерполирование - способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных значений. Один из методов интерполяции - интерполяционный полином Ньютона – полином при построении которого используются разделённые разности. Разделенная разность – обобщение понятия производной для дискретного набора точек.

$$f(x_j;x_{j+1};\dots;x_{j+k-1};x_{j+k})=\frac{f(x_{j+1};\dots;x_{j+k-1};x_{j+k})-f(x_j;x_{j+1};\dots;x_{j+k-1};x_{j+k-1})}{x_{j+k}-x_j}$$

В случае нахождения разделенной разности от точки x_0 мы можем упростить подсчет. Так например для подсчета разделенной разности мы имеем n точек.

$$f(x_0;x_1;\dots;x_{n-1};x_n)=\frac{f(x_1;\dots;x_{n-1};x_n)-f(x_0;x_1;\dots;x_{n-1};)}{x_n-x_0}$$

Тогда, для вычисления разделенной разности мы можем использовать формулу:

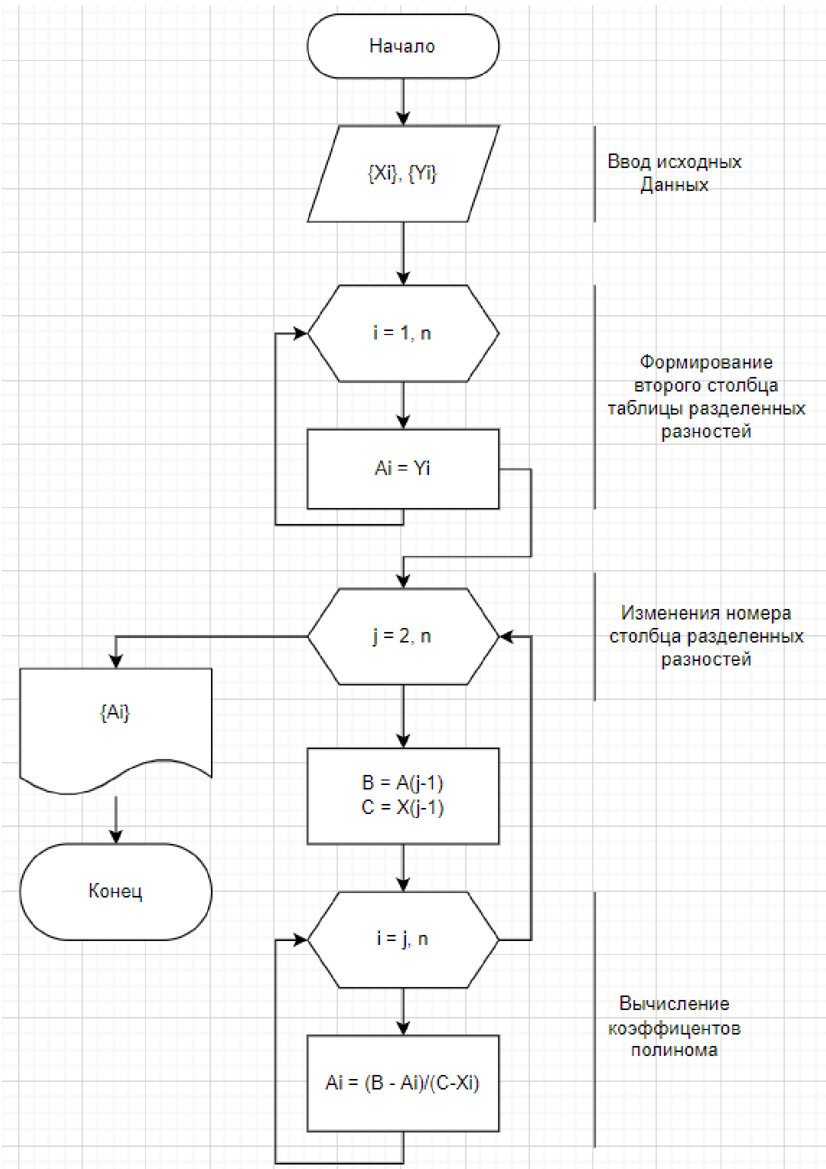
$$f(x_0;x_1;\dots;x_{n-1};x_n)=\sum_{j=0}^{n}\frac{f(x_j)}{\prod_{\substack{i=0\\ i\neq j}}^{n} x_j-x_i}$$

Используя данную форму полином Ньютона определяется следующим образом:

$$P_n(x)=f(x_0)+\sum_{k=1}^{n} f(x_0;\dots;x_k)\cdot\prod_{i=0}^{k-1}(x-x_i)$$

Таким образом мы получаем интерполяционным многочлен Ньютона для заданной функции.

Блок схема метода

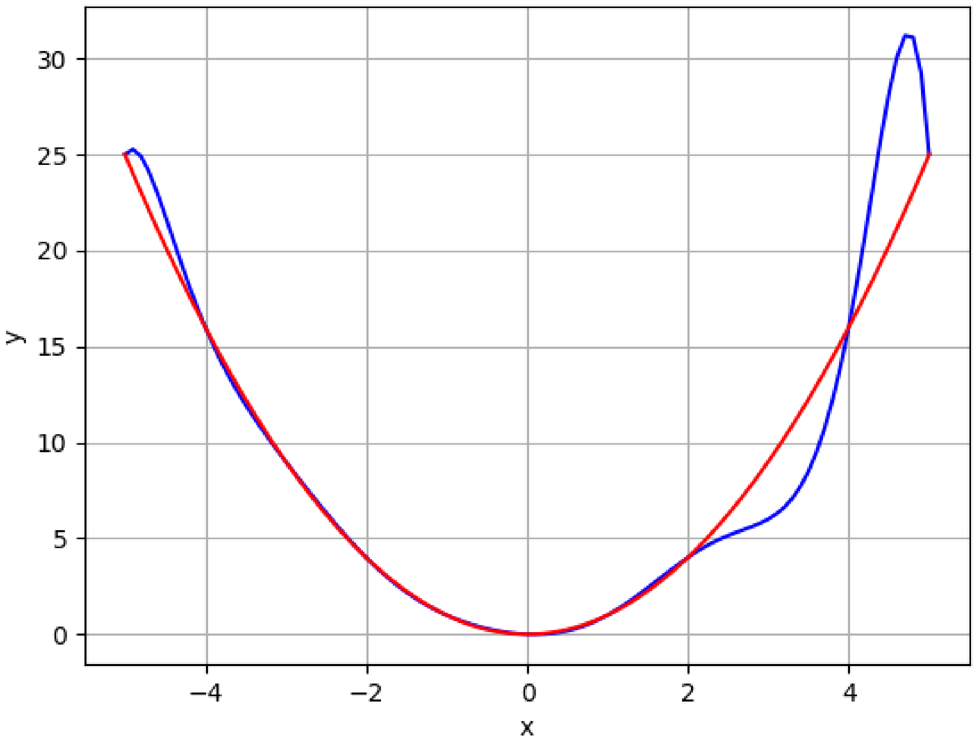


Листинг программы

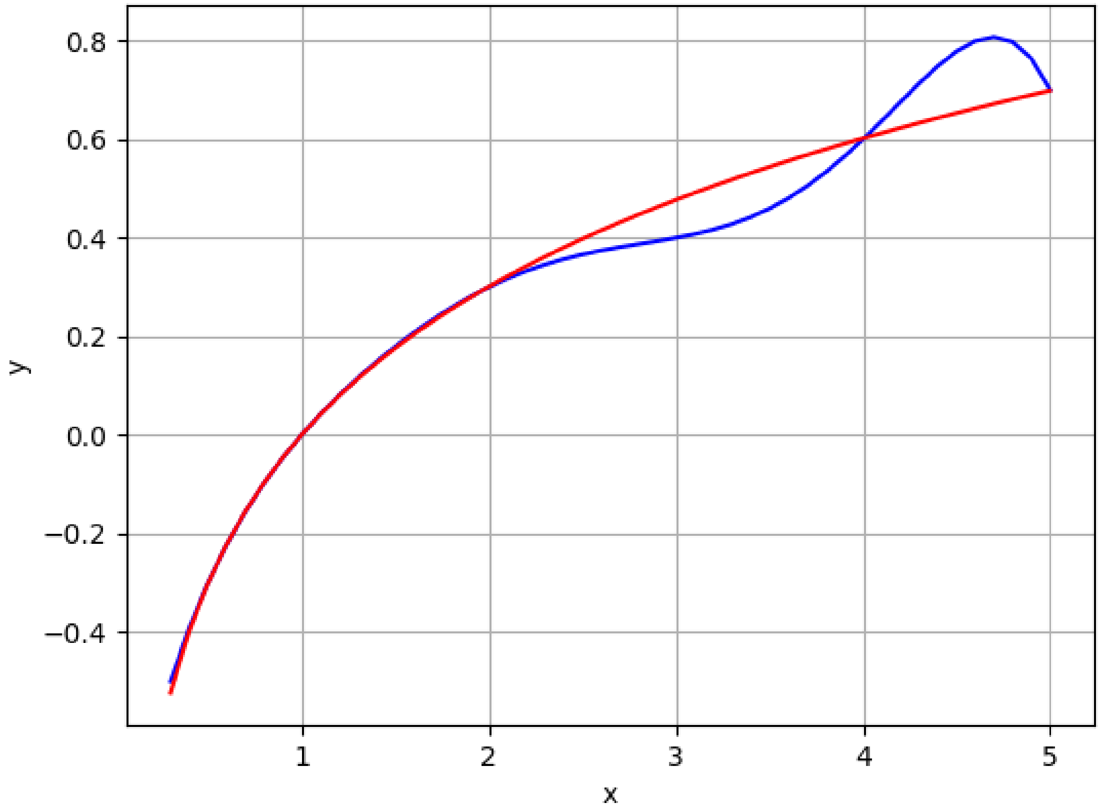
```
def divided_differences(x_values, y_values, k):
    result = 0
    for j in range(k + 1):
        mul = 1
        for i in range(k + 1):
            if i != j:
                mul *= x_values[j] - x_values[i]
        result += y_values[j] / mul
    return result


def create_newton_polynomial(x_values, y_values):
    div_diff = []
    for i in range(1, len(x_values)):
        div_diff.append(divided_differences(x_values, y_values, i))

    def newton_polynomial(x):
        result = y_values[0]
        for k in range(1, len(y_values)):
            mul = 1
            for j in range(k):
                mul *= (x - x_values[j])
            result += div_diff[k - 1] * mul
        return result

    return newton_polynomial
```

Пример работы

```
Здравствуйте, пожалуйсла выберите функцию:
(1) y = x
(2) y = x^2
(3) y = lg(x)
(4) y = sin(x)
>> 2
```



```
Здравствуйте, пожалуйсла выберите функцию:
(1) y = x
(2) y = x^2
(3) y = lg(x)
(4) y = sin(x)
>> 3
```



Вывод

В ходе выполнения данной лабораторной работы я познакомился с тем, чем отличается интерполяция от аппроксимации, узнал про способы интерполяции, и реализовал один из них. Так, например интерполяционный полином Лагранжа содержит значения исходной ф-и в явном виде, полином Ньютона содержит ети значения неявно (через конечные разности). В свою очередь метод сплайнов использует прогонку для вычисления СЛАУ. Поэтому его сложность равна O(n). Для методов Ньютона и Лагранжа погрешность (остаточный член) буден больше, чем больше точек построения, а для метода Сплайнов нельзя сказать тоже так как мы постоянно используем многочлены 3-й степени, и чем больше точек мы будем использовать – тем точнее будем приближать.