Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина «Методы численного анализа»

ОТЧЕТ

к лабораторной работе №6

на тему:

«ИНТРЕПОЛЯЦИОННЫЕ МНОГОЧЛЕНЫ»

БГУИР 1-40 04 01

Выполнил студент группы 253505
БЕКАРЕВ Станислав Сергеевич

____
(дата, подпись студента)

Проверил доцент кафедры информатики
АНИСИМОВ Владимир Яковлевич

____
(дата, подпись преподавателя)

Минск 2023

Содержание

Цель работы

- изучить интерполяцию функций с помощью интерполяционных многочленов Лагранжа и Ньютона;
- составить программу нахождения интерполяционного многочлена методом Лагранжа и методом Ньютона;
- выполнить тестовые примеры и проверить правильность работы программы

ЗАДАНИЕ. Построить интерполяционные многочлены в форме Лагранжа и Ньютона, используя номер варианта к, соответствующие значения параметров m и p_i и значения x_i, y_i из таблиц:

$\mathbf{x_i}$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\mathbf{p_i}$	0.0	0.41	0.79	1.13	1.46	1.76	2.04	2.3	2.55	2.79	3.01

$$y_i=p_i+(-1)^k m$$

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14
m	0	1	1.5	2	2.5	3	3.5	4	4.5	1.8	2.53	3.96	5.33	1.96

Оценить погрешность. Вычислить значение ф-ии в точке 0.47 с помощью интерполяционного многочлена и многочлена наилучшего приближения. Сравнить значения.

Вариант 3

Программная реализация

Исходные данные

```
Xi:  [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
Yi:  [ -1.50 -1.09 -0.71 -0.37 -0.04 0.26 0.54 0.80 1.05 1.29 1.51 ]
```

Код метода(многочлена) Лагранжа:

```
def lagrange_method(X, Y):
    f_x = 1
    L_x = 0
    for i in range(len(X)):
        f_x *= (x - X[i])
    for i in range(len(Y)):
        f_xj = f_x / (x - X[i])
        L_x += (f_xj / f_xj.subs(x, X[i])) * Y[i]
    return L_x
```

Код метода(многочлена) Ньютона:

```
def newton_method(X, Y):
    N_x = Y[0]
    for i in range(1, len(X)):
        f_i = add_newton_f(X[:i + 1:], Y[:i + 1:])
        for j in range(i):
            f_i *= (x - X[j])
        N_x += f_i
    return N_x


def add_newton_f(X, Y):
    if(len(X) == 2):
        return (Y[1] - Y[0]) / (X[1] - X[0])
    a = add_newton_f(X[1::], Y[1::])
    b = add_newton_f(X[:-1:], Y[:-1:])
    return (a - b) / (X[-1] - X[0])
```

Полученные результаты

```
Изначальные данные:
Xi:  [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
Yi:  [ -1.50 -1.09 -0.71 -0.37 -0.04 0.26 0.54 0.80 1.05 1.29 1.51 ]
Многочлен Лагранжа:
Значения в точках Xi:
Y[i]:  [ -1.50 -1.09 -0.71 -0.37 -0.04 0.26 0.54 0.80 1.05 1.29 1.51 ]
Значения в точке 0.47:
0.172652079713345
Отклонение от изначальных значений:
[ 2.2e-16 -9.1e-15 -9.2e-14 -5.1e-13 -1.9e-12
-6.6e-12 -2.0e-11 -5.4e-11 -1.4e-10 -3.4e-10 -7.7e-10 ]
Многочлен Ньютона:
Значения в точках Xi:
Y[i]:  [ -1.50 -1.09 -0.71 -0.37 -0.04 0.26 0.54 0.80 1.05 1.29 1.51 ]
Значения в точке 0.47:
0.172652079708683
Отклонение от изначальных значений:
[ 0.0e+00 0.0e+00 2.2e-15 -8.0e-15 1.3e-13
1.8e-13 9.5e-13 1.2e-12 2.5e-12 2.8e-12 -2.0e-12 ]
```

Для того чтобы проверить точность применим интерполяцию к функции log(x) с различным количеством узлов (т.е. точек Xi) и посмотрим на максимальную разницу между функциями на отрезке [1, 4].

Исходя из формулы погрешности интерполяции, погрешность уменьшается при росте кол-ва узлов интерполяции:

$$\omega(x) = (x - x_0)(x - x_1) \cdot \ldots \cdot (x - x_n),$$

$$R_n(x) = f(x) - P_n(x) \quad . \quad \text{Пусть} \quad f(x) \in C^{n+1}[a,b].$$

$$|R_n(x)| \le \frac{M_{n+1}}{(n+1)!}|\omega(x)|, \quad \text{где} \quad M_{n+1} = \max_{a \le x \le b}\left|f^{(n+1)}(x)\right|.$$

Тестовый пример 1. Четыре узла

```
Изначальная функция:
log(x)
Изначальные данные:
Xi:  [ 1.0000 2.0000 3.0000 4.0000 ]
Yi:  [ 0.0000 0.6931 1.0986 1.3863 ]
Многочлен Лагранжа:
Максимальное отклонение от log(x):
0.0141816180247324
Многочлен Ньютона:
Максимальное отклонение от log(x):
0.0141816180247319
```

Тестовый пример 2. Семь узлов

```
Изначальная функция:
log(x)
Изначальные данные:
Xi:  [ 1.0000 1.5000 2.0000 2.5000 3.0000 3.5000 4.0000 ]
Yi:  [ 0.0000 0.4055 0.6931 0.9163 1.0986 1.2528 1.3863 ]
Многочлен Лагранжа:
Максимальное отклонение от log(x):
0.000561669867694792
Многочлен Ньютона:
Максимальное отклонение от log(x):
0.000561669867690545
```

Тестовый пример 3. Десять узлов

```
Изначальная функция:
log(x)
Изначальные данные:
Xi:  [ 1.0000 1.3333 1.6667 2.0000 2.3333 2.6667 3.0000 3.3333 3.6667 4.0000 ]
Yi:  [ 0.0000 0.2877 0.5108 0.6931 0.8473 0.9808 1.0986 1.2040 1.2993 1.3863 ]
Многочлен Лагранжа:
Максимальное отклонение от log(x):
3.20621016597494e-5
Многочлен Ньютона:
Максимальное отклонение от log(x):
3.20621045546282e-5
```

Вывод

В ходе выполнения лабораторной работы я изучил метод Лагранжа и метод Ньютона построения интерполяционных многочленов, написал программу их реализации на языке Python, правильность работы программы проверил на тестовых примерах.

На основании тестов можно сделать следующие выводы:

- Достоинство интерполяционного многочлена Ньютона является удобство в расширении интерполяции и добавлении узлов
- Недостатком интерполяционного многочлена Ньютона является сложность в подсчете конечных разностей по сравнению с многочленом Лагранжа
- Точность интерполяции функции увеличивается с ростом количества узлов (точек Xi) отобранных для построения интерполяционного многочлена Лагранжа или Ньютона