Санкт-Петербургский национальный
исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники
Направление подготовки 09.03.04 «Программная инженерия»
Дисциплина «Вычислительная математика»

Отчет
По лабораторной работе №1
«Метод Ньютона»

Выполнил студент:
Бабушкин А.М. (P3221)
Преподаватель:
Перл О.В.

Санкт-Петербург
2024

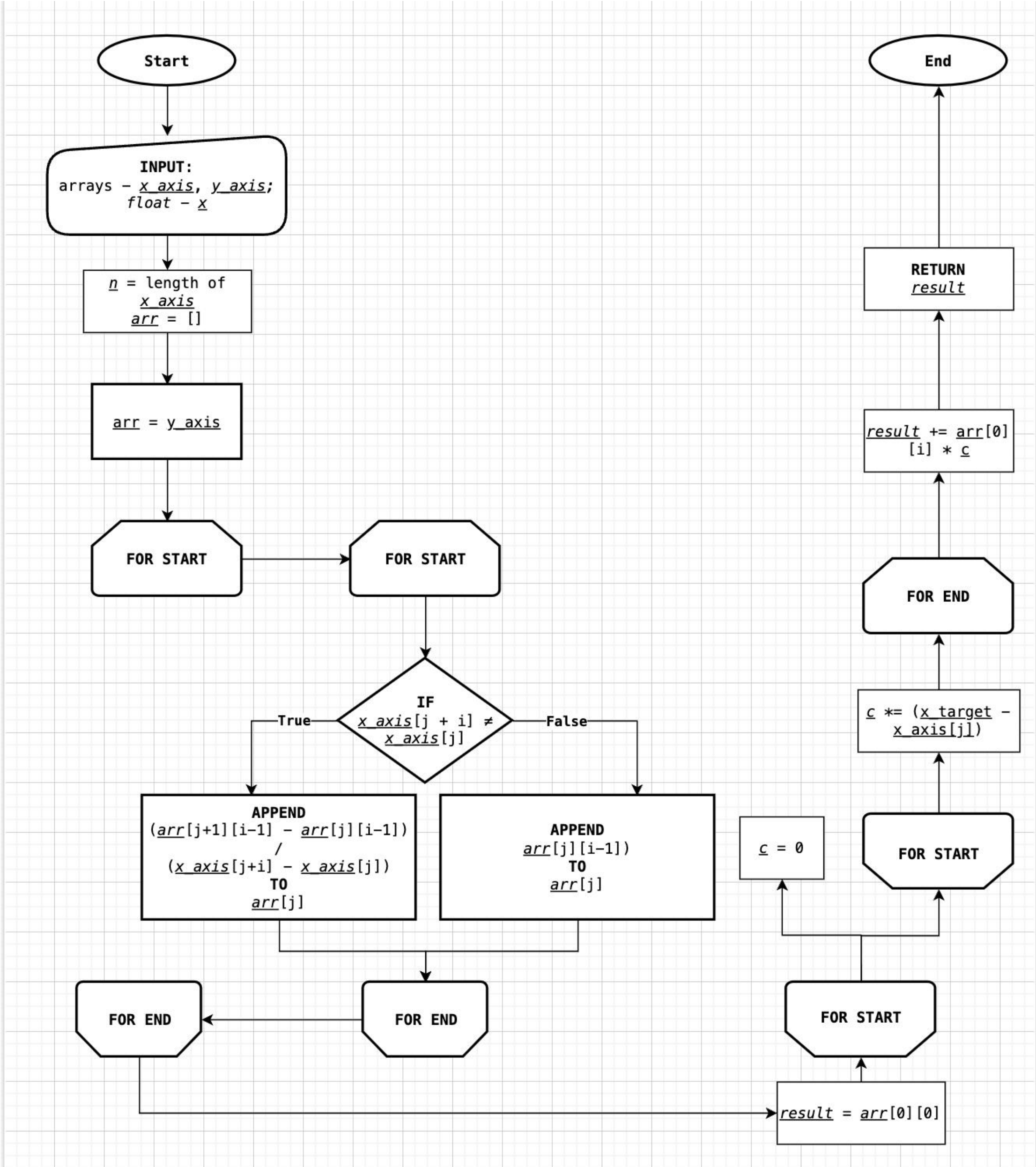
Описание численного метода:

Метод Ньютона- это численный метод нахождения приближенных значений функции ***y*** в точках ***x*** на заданном промежутке (от x_0 до x_n).
Интерполяционный полином Ньютона - это многочлен, служащий для построения полиномов **n**-ой степени, которые совпадают в точках (**n+1**) со значениями неизвестной **y** функции.

Формула :

$P_n(x) = f(x_0) + (x-x_0)f(x_0;x_1) + (x - x_0)(x - x_1)f(x_0;x_1;x_2)+\ldots+(x - x_0)\ldots(x - x_{n-1})f(x_0;..;x_n)$

Блок-схема:



Start
INPUT:
arrays – x_axis, y_axis;
float – x
n = length of
x_axis
arr = []
arr = y_axis
FOR START
FOR START
IF
x_axis[j + i] ≠
x_axis[j]
True
False
APPEND
(arr[j+1][i–1] – arr[j][i–1])
/
(x_axis[j+i] – x_axis[j])
TO
arr[j]
APPEND
arr[j][i–1])
TO
arr[j]
FOR END
FOR END
result = arr[0][0]
FOR START
c = 0
FOR START
c *= (x_target –
x_axis[j])
FOR END
result += arr[0]
[i] * c
RETURN
result
End

Метод реализованный на языкеPython:

```python
def interpolate_by_newton(x_axis, y_axis, x_target):
    n = len(x_axis)
    arr = [[y_axis[j]] for j in range(n)]
    for i in range(1, n):
        for j in range(n - i):
            if x_axis[j + i] != x_axis[j]:
                arr[j].append((arr[j + 1][i - 1] - arr[j][i - 1]) / (x_axis[j + i] - x_axis[j]))
            else:
                arr[j].append(arr[j][i - 1])

    result = arr[0][0]
    for i in range(1, n):
        c = 1
        for j in range(i):
            c *= (x_target - x_axis[j])
        result += arr[0][i] * c

    return result
```

Тесты:

```python
def run_tests ():
    if round(interpolate_by_newton([0, 1, 2, 3, 4], [1, 2, 3, 4, 5], 2.5), 2) == 3.5: print("Test 1 passed")
    else: print("Test 1 failed")

    if round(interpolate_by_newton([0, 1, 2, 3, 4], [0, 1, 4, 9, 16], 2.5), 2) == 6.25: print("Test 2 passed")
    else: print("Test 2 failed")

    if round(interpolate_by_newton([-2, -1, 0, 1, 2], [4, 1, 0, 1, 4], -2.5), 2) == 6.25: print("Test 3 passed")
    else: print("Test 3 failed")

    if round(interpolate_by_newton([0, 1, 2, 3, 4], [0, 1, 8, 27, 64], 2.5), 2) == 15.62: print("Test 4 passed")
    else: print("Test 4 failed")

    if round(interpolate_by_newton([200, 400, 600, 800, 1000], [5, 10, 15, 20, 25], 750), 2) == 18.75: print("Test 5 passed")
    else: print("Test 5 failed")




if __name__ == '__main__':
    run_tests()
```

ПРОБЛЕМЫ 10 ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

```
aleksandrbabushkin@Air-Aleksandr s4-compmath-lab1 % /usr/bin/python3 /Users/aleksandrbabushkin/ITMO/4th-semester/computational-math/s4-compmath-lab1/main.py
Test 1 passed
Test 2 passed
Test 3 passed
Test 4 passed
Test 5 passed
aleksandrbabushkin@Air-Aleksandr s4-compmath-lab1 %
```

Вывод:

В ходе выполнения лабораторной работы мной был осовен метод интерполяции полиномом Ньютона. Я изучил чем он отличается от других методов интеорполяции, какими преимуществами и недостатками он обладает.

Сложность алгоритма – *O(n)*.

Точность итоговых значений после работы алгоритма зависит от количества входных значений, то есть чем больше узлов нам известно, тем приближеннее будет полином к действительной функции.

Это особенно заметно когда мы берем набор из небольшого количества узлов в котором есть пара точек, значения х и у которых сильно отличаются. Так, при построении графика на интервале между этими точками, линия графика может немного сильнее отклоняться вниз или вверх, в зависимости от расположения самих точек, чем если бы мы использовали к примеру метод кубических сплайнов.

Полином Ньютона отличается от другого, крайне распространенного, метода - полинома Лагранжа способом задания.

Полином Ньютона использует конечные разности, вычисляемые для каждой точки набора, по которым уже и строится полином.

Полином Лагранжа использует базисные полиномы, вычисляемые для каждой точки набора, которые объединяются в единый при помощи коэффициентов Лагранжа.

Преимущество полинома Ньютона над полиномом Лагранжа в том, что при добавлении нового узла в набор нам не нужно полностью пересчитывать полином, а лишь добавить одно новое слагаемое.