

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. (Р3221)
Преподаватель:
Перл О.В.

Санкт-Петербург
2024

Описание численного метода:

Метод Ньютона - это итерационный численный метод нахождения приближенного значения корня заданной функции. Он основан на идее линейной аппроксимации функции в окрестности текущего приближения к корню.

Интерполяционный полином Ньютона - это многочлен, который используется для интерполяции набора данных.

Формула:

$$P(x) = f(x_0) + \sum_{i=1}^n (x - x_0)(x - x_1) \dots (x - x_{i-1}) f[x_0, x_1, \dots, x_i]$$

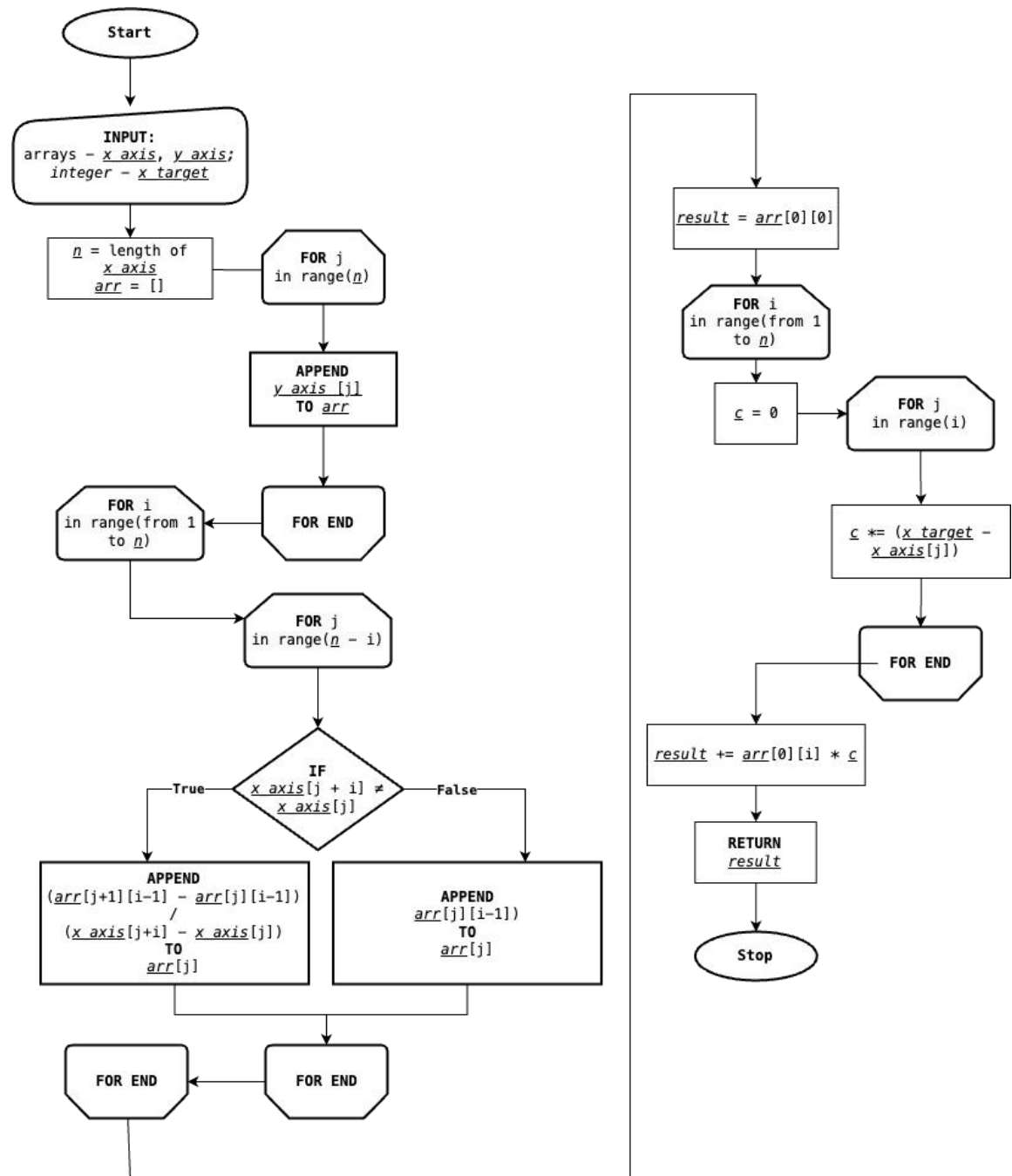
$$f[x_i] = f(x_i)$$

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$$

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}$$

Метод Ньютона может быть более эффективным, чем метод Лагранжа, в том случае если нам известен набор значений функции в некоторых точках, неравномерно распределенных на всём интервале.

Блок-схема:



Метод реализованный на языке 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:

```
31 def run_tests ():
32     if round(interpolate_by_newton([0, 1, 2, 3, 4], [1, 2, 3, 4, 5], 2.5), 2) == 3.5: print("Test 1 passed")
33     else: print("Test 1 failed")
34
35     if round(interpolate_by_newton([0, 1, 2, 3, 4], [0, 1, 4, 9, 16], 2.5), 2) == 6.25: print("Test 2 passed")
36     else: print("Test 2 failed")
37
38     if round(interpolate_by_newton([-2, -1, 0, 1, 2], [4, 1, 0, 1, 4], -2.5), 2) == 6.25: print("Test 3 passed")
39     else: print("Test 3 failed")
40
41     if round(interpolate_by_newton([0, 1, 2, 3, 4], [0, 1, 8, 27, 64], 2.5), 2) == 15.62: print("Test 4 passed")
42     else: print("Test 4 failed")
43
44     if round(interpolate_by_newton([200, 400, 600, 800, 1000], [5, 10, 15, 20, 25], 750), 2) == 18.75: print("Test 5 passed")
45     else: print("Test 5 failed")
46
47
48
49
50 if __name__ == '__main__':
51     run_tests()
52
```

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 результате выполнения лабораторной работы я ознакомился с методом Ньютона для интерполяции.

Метод Ньютона - это эффективный метод интерполяции, который использует конечные разности для вычисления полинома. Он превосходит метод Лагранжа при неравномерном распределении узлов и позволяет добавлять новые узлы без пересчета всей таблицы разностей.

Однако, его точность и эффективность зависят от количества узлов, а алгоритмическая сложность составляет $O(n^2)$ для построения таблицы разностей и $O(n)$ для вычисления интерполяционного полинома.