Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1
по дисциплине “Вычислительная математика”

Выполнила:
Шевченко Д. П.,
группа P3230

Преподаватель:
Перл О.В.

Санкт-Петербург
2024

Оглавление

Задание

Метод Ньютона с использованием полиномов Чебышева

Дан набор точек, по которым необходимо построить интерполяционный полином Ньютона с использованием полиномов Чебышёва для заданного интервала. Также задана координата x, для которой необходимо найти значение интерполяционного полинома.

Формат входных данных:
f
a
b
x

где f - номер интерполируемой функции, a и b - границы интерполируемого интервала и x - значение аргумента для интерполяционного полинома.

Степень полинома Чебышева (количество узлов интерполяции) следует увеличивать до тех пор, пока модуль разницы между значениями интерполирующей функции в искомой точке x не будет меньше, чем 0.01.

Формат выходных значений: вещественное число, являющееся значением интерполяционного полинома в точке x.

Описание численного метода

Для применения метода Ньютона нужно выбрать функцию, которую будем интерполировать. Обозначаем интервал интерполирования и точку, в которой хотим вычислить значение интерполяционного полинома.
Для начала определим количество узлов Чебышева. Возьмем сначала 1 узел. Будем их увеличивать на 1, пока разница между вычисленными значениями полинома в точке будет не меньше, чем 0.01
Вычисляем каждый x_k по формуле:

$$\frac{a+b}{2} + \frac{b-a}{2} * \cos\left(\frac{2k-1}{2n}\pi\right)$$

a, b – границы интервала
k – номер узла
n – количество узлов

После, вычисляем значение функции в каждом узле, подставляя x_k в функцию.
Далее нужно вычислить разделенные разности, по формуле:

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

$$\dots$$

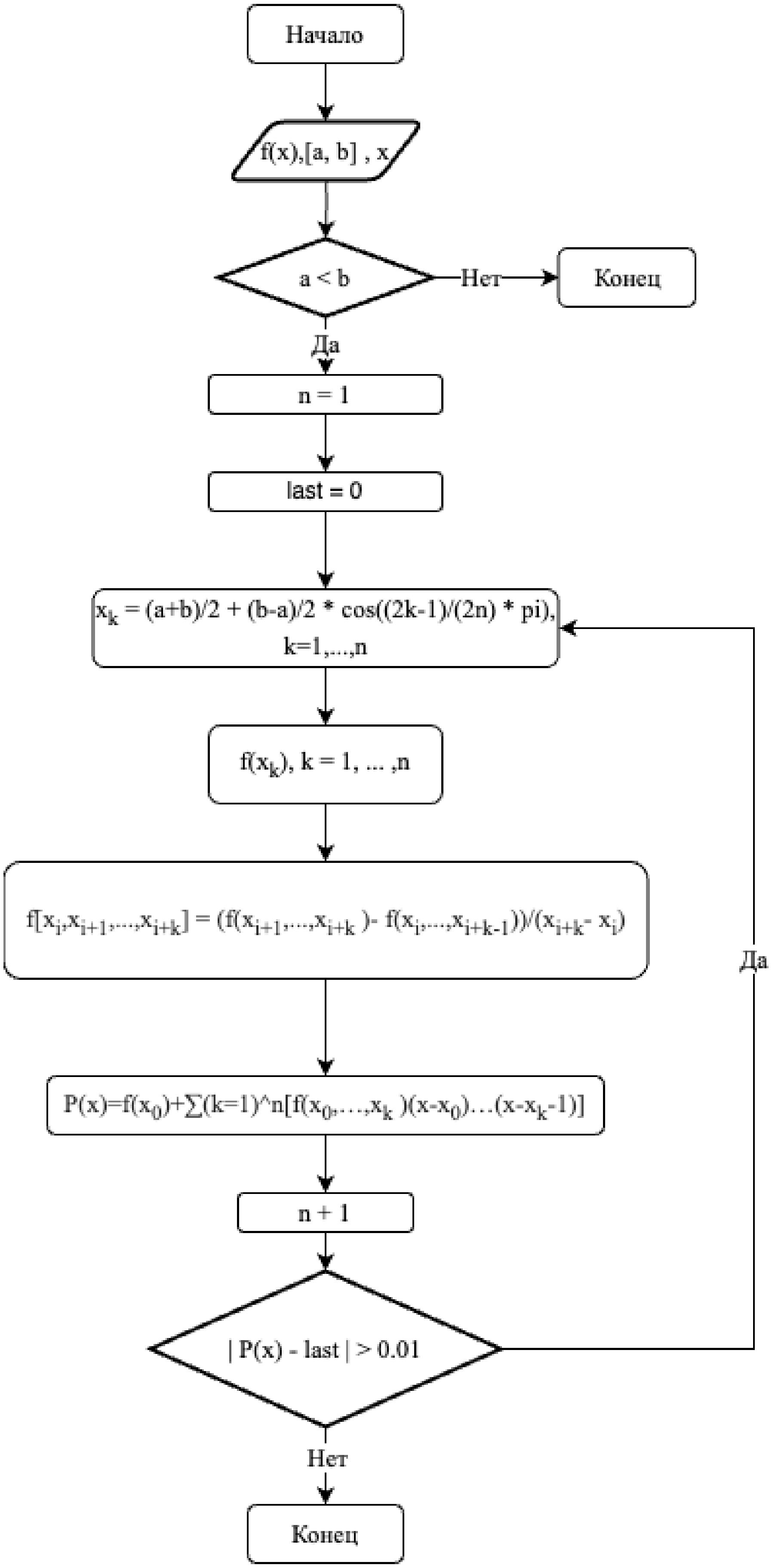
$$f[x_i, x_{i+1}, \dots, x_{i+k}] = \frac{f(x_{i+1}, \dots, x_{i+k}) - f(x_i, \dots, x_{i+k-1})}{x_{i+k} - x_i}$$

Далее строим интерполяционный полином Ньютона, для этого используем формулу:

$$P(x) = f(x_0) + \sum_{k=1}^{n} f(x_0, \dots, x_k)(x - x_0) \dots (x - x_{k-1})$$

x – точка, в которой хотим вычислить значение полинома
x_k – узлы
$f(x_0, \dots, x_k)$ – разделенные разности

Блок схема


Начало
f(x),[a, b] , x
a < b
Нет
Конец
Да
n = 1
last = 0
x_k = (a+b)/2 + (b-a)/2 * cos((2k-1)/(2n) * pi),
k=1,...,n
f(x_k), k = 1, ... ,n
f[x_i,x_{i+1},...,x_{i+k}] = (f(x_{i+1},...,x_{i+k})- f(x_i,...,x_{i+k-1}))/(x_{i+k}- x_i)
P(x)=f(x_0)+∑(k=1)^n[f(x_0,…,x_k)(x-x_0)…(x-x_k-1)]
n + 1
| P(x) - last | > 0.01
Да
Нет
Конец

Код

```
def newton_polinom(x, nodes, func):
    func_in_node = [func(node) for node in nodes]  # значение функции в 
конкретном узле
    for i in range(1, len(func_in_node)):
        for j in range(len(func_in_node) - 1, i - 1, -1):
            func_in_node[j] = (func_in_node[j] - func_in_node[j - 1]) / 
(nodes[j] - nodes[j - i])
    value = func_in_node[
        -1]  # В начале устанавливается начальное значение value равное 
последнему коэффициенту func_in_node[-1]
    for i in range(len(func_in_node) - 2, -1, -1):
        value = func_in_node[i] + (x - nodes[i]) * value
    return value


def interpolate_by_newton(f, a, b, x):
    node_count = 1
    result = 1000000000
    last_value = 0
    if a > b:
        raise NotImplementedError(f"a must be smaller then b")
    if f == 2 and x == 0:
        return math.nan
    while abs(result - last_value) > 0.01:
        node_count += 1
        nodes = []
        for i in range(1, node_count + 1):
            node = (a + b) / 2 + ((b - a) / 2) * math.cos(math.pi * (2 * i - 
1) / 2 / node_count)
            nodes.append(node)
        func_value_in_x = newton_polinom(x, nodes, 
FunctionSet.get_function(f))
        last_value = result
        result = func_value_in_x
        if math.isnan(result) or math.isinf(result):
            return math.nan
    return result
```

Вывод

Результаты запуска метода на различных данных:

1) Корректный вывод для функции Вейерштрасса

```
1
1
2
2
0.7811783552056129
```

2) Корректный вывод для Гамма-функции

```
2
1
2
1.5
0.8849702054967064
```

3) Интервал введен неверно

```
2
2
1
1
Traceback (most recent call last):
  File "/Users/dasha/PycharmProjects/metops/test.py", line 138, in <module>
    result = interpolate_by_newton(f, a, b, x)
  File "/Users/dasha/PycharmProjects/metops/test.py", line 115, in interpolate_by_newton
    raise NotImplementedError(f"a must be smaller then b")
NotImplementedError: a must be smaller then b
```

4) Будет бесконечное количество узлов Чебышева

```
1
1
2
7
nan
```

5) x не может быть равен нулю, потому что у гамма функции в одном из слагаемых знаменатель выглядит так: x + 0.0

```
2
0
2
0
nan
```

Сравнение с другими методами:

По сравнению с методом Лагранжа, в методе Ньютона при добавлении нового узла не нужно считать заново весь полином, а только 1 новое слагаемое, что значительно легче и быстрее.

Интерполяция по узлам Чебышева является более точной, чем по равноотстоящим узлам. С помощью использования полинома Чебышева максимумы величины ошибки можно уравнять если распределить узлы, так как при равномерном распределении узлов погрешность растет на границах интервала, тем более с увеличением порядка интерполяции.

Анализ применимости метода:

Если у функции есть особые точки, то результаты в них могут быть не точными.
Узлы Чебышева позволяют уменьшить ошибку приближения, так как узлы равномерно распределены на интервале.

Анализ алгоритмической сложности метода:

Для вычисления x в узлах, используется цикл for, у которого сложность O(n), n – количество узлов

Для вычисления разделенных разностей используется два цикла for, у которых сложность $O(n^2)$, так как len(func_in_node) равно количеству узлов.

Для вычисления значения полинома в точке тоже используется цикл for, который проходится n – 2 раз, соответственно сложность O(n-2) =O(n)

Из этого следует, что сложность метода равна $O(n^2 + 2n) = O(n^2 + n)$

Анализ численной ошибки самого численного метода

Численная ошибка определяется убыванием членов суммы полинома.

Согласно эффекту Рунге, полиномиальные интерполяции плохо ведут себя по поточечному супремуму погрешности. Но узлы Чебышева гарантируют равномерно стремящуюся ошибку интерполяции.

Соответственно, можно сделать вывод, что с помощью метода Ньютона легче добавлять узлы, при этом не пересчитывая все заново. А с помощью узлов Чебышева можно снизить влияние феномена Рунге.