Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина: Методы численного анализа

ОТЧЁТ
к лабораторной работе
на тему

Интерполяционные многочлены

Выполнил: студент группы 253505
Снежко Максим Андреевич

Проверил: Анисимов Владимир Яковлевич

Минск 2023

Содержание

1. ЦЕЛЬ РАБОТЫ

1) Изучить интерполяцию функций с помощью интерполяционных многочленов Лагранжа и Ньютона, аппроксимацию функций методом наименьших квадратов.
2) Составить алгоритм и программу нахождения многочленов соответствующими методами.
3) Проверить правильность работы программы на тестовых примерах.
4) Решить задание заданного варианта.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

7. АППРОКСИМАЦИЯ И ИНТЕРПОЛЯЦИЯ ФУНКЦИЙ

Из математического анализа известно, что в окрестности точки x_0 любую n раз непрерывно дифференцируемую функцию можно аппроксимировать (приблизить) ее многочленом Тейлора:

$$P_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)(x - x_0)^k}{k!},$$

причем

$$f(x_0) = P_n(x_0),$$
$$f'(x_0) = P'_n(x_0),$$
$$\dots\dots\dots\dots\dots\dots$$
$$f^{(n)}(x_0) = P_n^{(n)}(x_0).$$

Очевидно, такая аппроксимация во многих отношениях является очень хорошей, но она имеет локальный характер, т.е. хорошо аппроксимирует функцию только вблизи точки x_0. Это главный недостаток аппроксимации с помощью многочлена Тейлора.

Если речь идет об аппроксимации функции на отрезке, применяются другие методы.

Пусть $f(x) \in C[a,b]$ – непрерывная функция. Рассмотрим задачу аппроксимации (приближения) ее более простой функцией (обычно многочленом).

Известно из математического анализа, что в силу теоремы Вейерштрасса, любую функцию можно с какой угодно точностью приблизить многочленом по норме $\|f(x)\| = \max\limits_{a \le x \le b} |f(x)|$ пространства C[a, b], т. е. в смысле равномерной сходимости. Но существуют и другие нормы:

$$\|f(x)\| = \int_a^b |f(x)|dx \quad \text{или} \quad \|f(x)\| = \sqrt{\int_b^a |f(x)|^2 dx}.$$

Тогда $\|f(x) - P(x)\| < \varepsilon$ означает, что площадь или усредненная площади фигуры, заключенной между графиками функции $f(x)$ и многочлена $P(x)$, должна быть меньше ε (заданной точности).

Возможен и другой подход, когда в качестве аппроксимирующей функции берут многочлен или другую достаточно простую функцию, значения которых совпадают со значениями исходной функции в заданных заранее точках, так называемых узлах. Такого рода приближение функций имеет свое собственное название - интерполяция.

7.1. Интерполяционный многочлен

Пусть $f(x)$ – функция, непрерывная на отрезке $[a,b]$.

Выберем на этом отрезке точки, называемые *узлами интерполяции*:

$a \le x_0 < x_1 < ... < x_n \le b$.

Предположим, что известны значения функции в узлах интерполяции: $f(x_k) = y_k, \quad k = 0,1,...,n$.

Ставится задача найти многочлен $P_n(x)$ такой, что

$$P_n(x_k) = y_k, \qquad \forall k = 0,1,...,n. \tag{7.1}$$

Такой многочлен $P_n(x)$ называется *интерполяционным многочленом*, а задача его нахождения – *задачей интерполяции.*

Покажем, что задача интерполяции имеет решение, причем единственное.

Пусть $P_n(x) = \sum_{k=0}^{n} a_k x^{n-k}$.

Тогда для определения коэффициентов многочлена из условия (7.1) получаем систему:

$$\begin{cases} a_0 x_0^n + a_1 x_0^{n-1} + ... + a_n = y_0 \\ a_0 x_1^n + a_1 x_1^{n-1} + ... + a_n = y_1 \\ .... \\ a_0 x_n^n + a_1 x_n^{n-1} + ... + a_n = y_n . \end{cases}$$

Ее определитель Δ с точностью до знака совпадает с так называемым определителем Вандермонда.

$$W(x_0,...,x_n) = \begin{vmatrix} 1 & 1 & ... & 1 \\ x_0 & x_1 & ... & x_n \\ x_0^2 & x_1^2 & ... & x_n^2 \\ ... & ... & ... & ... \\ x_0^n & x_1^n & ... & x_n^n \end{vmatrix} = \prod_{i<j}(x_j - x_i) \ne 0 .$$

Поскольку все x_i различны, определитель Δ отличен от нуля, и, следовательно, система имеет единственное решение. Отсюда вытекает существование и единственность интерполяционного многочлена.

Погрешность интерполяции.

Обозначим

$R_n(x) = f(x) - P_n(x)$ и будем искать ее оценку.

Пусть $f(x) \in C^{n+1}[a,b]$. Положим $R_n(x) = \omega(x) r(x)$,

где $\omega(x) = (x - x_0)(x - x_1) \cdot ... \cdot (x - x_n)$.

Зафиксируем произвольную точку x, отличную от узлов интерполяции x_i, $i = \overline{0,n}$, и построим вспомогательную функцию:

$$F(t) = P_n(t) + \omega(t) r(x) - f(t), \qquad a \le t \le b . \tag{7.2}$$

Очевидно, $F(x) = 0$ и, кроме того $F(x_k) = 0, \qquad k = \overline{0,n}$.

Таким образом, функция $F(t)$ имеет по крайней мере $(n+2)$ нуля на отрезке $[a,b]$. Применим теорему Ролля, по которой между каждой парой нулей функции находится по крайней мере один нуль производной этой функции.

Легко видеть, что $L_n(x_i) = l_i(x_i) y_i = 1 \cdot y_i = y_i$, $i = \overline{0,n}$, т.е. это интерполяционный многочлен. Его называют интерполяционным многочленом Лагранжа.

Пример. Рассмотрим задачу интерполяции для функции

$f(x) = \sin\frac{\pi}{2}x$, на $[0,1]$.

Выберем в качестве узлов точки $x_0 = 0$, $x_1 = 1/3$, $x_2 = 1$. Тогда значения функции: $y_0 = 0$, $y_1 = 1/2$, $y_2 = 1$.

Получим

$$L_2(x) = \frac{(x-1/3)\cdot(x-1)}{(-1/3)\cdot(-1)} + \frac{x\cdot(x-1)\cdot\frac{1}{2}}{1/3\cdot(-2/3)} + \frac{(x-1/3)\cdot x}{2/3\cdot 1} = -3/4\cdot x^2 + 7/4\cdot x.$$

Оценим погрешность. Поскольку можно показать, что $|\omega(x)| \le 0{,}079$, то

$$R_2(x) \le \frac{\pi^3}{3!\cdot 8}\max_{0<x<1}|\varpi(x)| \le \frac{\pi^3}{3!\cdot 8}\cdot 0{,}079.$$

Линейная интерполяция

Пусть $n=1$, т. е. даны два узла x_0, x_1 справа и слева от точки x:

$x_0 \le x \le x_1$.

Построим интерполяционный многочлен первой степени по этим узлам.

Значения функции $f(x)$ в этих узлах y_0, y_1.

Получаем:

$$L_1(x) = \frac{x-x_1}{x_0-x_1}\cdot y_0 + \frac{x-x_0}{x_1-x_0}\cdot y_1 = y_0 + \frac{y_1-y_0}{x_1-x_0}\cdot(x-x_0).$$

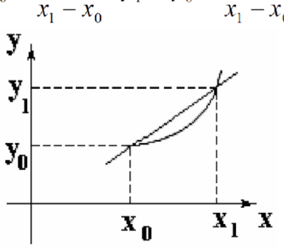


Рис. 7.1.

т. е. графически интерполяционный многочлен представляет собой хорду, соединяющую точки (x_0, y_0) и (x_1, y_1) (рис. 7.1).

Оценим погрешность линейной интерполяции.

Пусть $h = x_1 - x_0$.

Тогда $\max\limits_{x_0 \le x \le x_1}|\omega(x)| = \max|(x-x_0)\cdot(x-x_1)| = \frac{h^2}{4}$,

так как функция $|\omega(x)|$ достигает максимума на $[x_0, x_1]$ в точке $x_m = \frac{x_0+x_1}{2}$.

(рис. 7.2).

69

Тогда производная $F'(t)$ имеет по крайней мере $(n+1)$ нулей на данном интервале (a,b). Продолжая рассуждение, получим в итоге, что $F^{(n)}(t)$ имеет, по крайней мере, два нуля, а $F^{(n+1)}(t)$ – один нуль в некоторой точке ξ на (a,b).

Продифференцируем равенство (7.2) $(n+1)$ раз и подставим $t = \xi$. Получим

$$F^{(n+1)}(\xi) = (n+1)! \cdot r(x) - f^{(n+1)}(\xi) = 0.$$

Откуда $r(x) = \dfrac{f^{(n+1)}(\xi)}{(n+1)!}$.

Тогда

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega(x),$$

где $\xi \in [a,b]$ (очевидно формула напоминает остаток формулы Тейлора в форме Лагранжа). В итоге имеем оценку погрешности интерполяции:

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

Интерполяционный многочлен Лагранжа

Пусть даны узлы на отрезке $[a,b]$, $a \le x_0 < x_1 < \ldots < x_n \le b$, и значения функции *F(x)* в узлах

$$f(x_i) = y_i, \qquad i = \overline{0,n}.$$

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

$$\omega_j(x) = (x-x_0)\cdot\ldots\cdot(x-x_{j-1})(x-x_{j+1})\cdot\ldots\cdot(x-x_n),$$

т. е. $\omega_j(x) = \dfrac{\omega(x)}{x-x_j}$.

Положим $l_j(x) = \dfrac{\omega_j(x)}{\omega_j(x_j)}$,

т. е. $$l_j(x) = \frac{(x-x_0)\cdot\ldots\cdot(x-x_{j-1})(x-x_{j+1})\cdot\ldots\cdot(x-x_n)}{(x_j-x_0)\cdot\ldots\cdot(x_j-x_{j-1})(x_j-x_{j+1})\cdot\ldots\cdot(x_j-x_n)}.$$

Очевидно $l_j(x_i) = \begin{cases} 0, & \text{при } i \ne j \\ 1, & \text{при } i = j. \end{cases}$

Построим многочлен $L_n(x) = \sum_{j=0}^{n} l_j(x) y_j$.

$$f_1(x_k, x_{k+1}) = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} = \frac{\Delta y_k}{\Delta x_k}.$$

Разделенной разностью второго порядка называется выражение

$$f_2(x_k, x_{k+1}, x_{k+2}) = \frac{f_1(x_{k+1}, x_{k+2}) - f_1(x_k, x_{k+1})}{x_{k+2} - x_k}$$ и т. д.

Пусть x – любая точка отрезка, не совпадающая с узлами. Тогда

$$f_1(x, x_0) = \frac{y_0 - f(x)}{x_0 - x},$$

откуда $f(x) = y_0 + f_1(x, x_0)(x - x_0)$. (7.3)

Далее $f_2(x, x_0, x_1) = \dfrac{f_1(x_0, x_1) - f_1(x, x_0)}{x_1 - x}$,

откуда $f_1(x, x_0) = f_1(x_0, x_1) + f_2(x, x_0, x_1)(x - x_1)$.

Подставляя в (7.3), получаем

$$f(x) = y_0 + f_1(x_0, x_1)(x - x_0) + f_2(x, x_0, x_1)(x - x_0)(x - x_1). \quad (7.4)$$

Далее $f_3(x, x_0, x_1, x_2) = \dfrac{f_2(x_0, x_1, x_2) - f_2(x, x_0, x_1)}{x_2 - x}$,

откуда $f_2(x, x_0, x_1) = f_2(x_0, x_1, x_2) + f_3(x, x_0, x_1, x_2)(x - x_2)$.

Подставляя в (4), имеем:

$$f(x) = y_0 + f_1(x_0, x_1)(x - x_0) + f_2(x, x_0, x_2)(x - x_0)(x - x_1) + \\ + f_3(x, x_0, x_1, x_2)(x - x_0)(x - x_1)(x - x_2). \quad (7.5)$$

Продолжая процесс, получим:

$$f(x) = N_n(x) + f_{n+1}(x, x_0, ..., x_n)(x - x_0)...(x - x_n),$$

где $N_n(x) = y_0 + f_1(x_0, x_1)(x - x_0) + ... + f_n(x_0, ..., x_n)(x - x_0)...(x - x_{n-1})$.

Очевидно, при $x = x_i$, $\forall i = \overline{0, n}$, $f(x_i) = N_n(x_i)$, $i = \overline{0, n}$,

т. е. $N_n(x)$ − интерполяционный многочлен. Его называют интерполяционным многочленом Ньютона.

Достоинство интерполяционного многочлена Ньютона: он удобен при расширении интерполяции и добавлении узлов.

Недостаток: в какой-то степени он сложнее в подсчете конечных разностей по сравнению с многочленом Лагранжа.

Интерполяционный многочлен Ньютона - Грегори

Рассмотрим случай задачи интерполяции с равноотстоящими узлами, т. е. пусть

$h = x_{i+1} - x_i$, для всех $i = \overline{0, n}$.

Будем искать интерполяционный многочлен Ньютона в форме

$$N(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + ... + a_n(x - x_0) \cdot ... \cdot (x - x_{n-1}),$$

часть информации. Поэтому вместо интерполяционного многочлена будем искать многочлен $P_m(x)$ меньшей степени $(m<n)$, такой что сумма

$$\sum_{i=0}^{n}[f(x_i)-P_m(x_i)]^2$$

принимает наименьшее значение. Данный многочлен называется многочленом наилучшего приближения по методу наименьших квадратов.

Положим

$$P_m(x)=a_0x^m+\ldots+a_m$$

и будем искать решение задачи

$$S(a_0,\ldots,a_m)=\sum_{i=0}^{n}[a_0x_i^m+\ldots+a_{m-1}x_i+a_m-y_i]^2\to\min.$$

Приравнивая к нулю производные S, получим систему линейных уравнений для определения коэффициентов a_i:

$$\frac{\partial S}{\partial a_0}=2\sum_{i=0}^{n}[a_0x_i^m+\ldots+a_m-y_i]\cdot x_i^m=0$$

$$\frac{\partial S}{\partial a_1}=2\sum_{i=0}^{n}[a_0x_i^m+\ldots+a_m-y_i]\cdot x_i^{m-1}=0$$

....

$$\frac{\partial S}{\partial a_{m-1}}=2\sum_{i=0}^{n}[a_0x_i^m+\ldots+a_m-y_i]\cdot x_i=0$$

$$\frac{\partial S}{\partial a_m}=2\sum_{i=0}^{n}[a_0x_i^m+\ldots+a_m-y_i]\cdot 1=0\ .$$

Отсюда получается

$$\begin{cases} a_0(\sum_{i=0}^{n}x_i^{2m})+a_1\sum_{i=0}^{n}x_i^{2m-1}+\ldots+a_m\sum_{i=0}^{n}x_i^m=\sum_{i=0}^{n}y_ix_i^m \\ a_0(\sum_{i=0}^{n}x_i^{2m-1})+\ldots+a_m\sum_{i=0}^{n}x_i^{m-1}=\sum_{i=0}^{n}y_ix_i^{m-1} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_0(\sum_{i=0}^{n}x_i^{n})+\ldots+a_m\sum_{i=0}^{n}1=\sum_{i=0}^{n}y_i \end{cases}$$

– нормальная система для определения коэффициентов $a_0,a_1,\ldots,a_n$.

Когда $m\le n$, можно показать, что нормальная система имеет единственное решение, которое действительно дает минимальное значение для функции S. Получив решения нормальной системы $a_0,\ldots,a_n$, строим многочлен наилучшего приближения по методу наименьших квадратов.

В частном случае, когда $m-n$, многочлен $P_n(x)$ переходит в интерполяционный многочлен.

Для решения нормальной системы обычно используется следующая таблица:

причем, поскольку функции $f_1,...,f_n$ линейно независимы, квадратичная форма равна нулю только тогда, когда все $\alpha_1,...,\alpha_n$ нулевые.

Следовательно, решение нормальной системы доставляет минимум функции $Q(\alpha_1,...,\alpha_n)$.

Теорема доказана.

Следствие. Чтобы численно решить задачу построения среднеквадратичного многочлена, надо составить и решить нормальную систему, а ее решение взять в качестве коэффициентов обобщенного многочлена.

Пример. Пусть $f(x)=\sqrt{x},\ x\in[0,\ 1]$. Построим многочлен наилучшего средне квадратичного отклонения по системе линейно независимых функций: *1, x*. Обозначим его $T_2(x)=a+b\cdot x$.

Получаем:

$$[\Gamma(1,x)]=\begin{pmatrix}(1,1) & (1,x)\\ (x,1) & (x,x)\end{pmatrix}=\begin{pmatrix}1 & 1/2\\ 1/2 & 1/3\end{pmatrix},$$

$$(\sqrt{x},1)=\int_0^1\sqrt{x}dx=\frac{2}{3}x^{\frac{3}{2}}\Big|_0^1=\frac{2}{3},$$

$$(\sqrt{x},x)=\int_0^1\sqrt{x}\cdot xdx=\frac{2}{5}\ .$$

Записываем нормальную систему:

$$\begin{cases}a+\frac{1}{2}b=\frac{2}{3}\\ \frac{1}{2}a+\frac{1}{3}b=\frac{2}{5},\end{cases}$$

решая ее, находим:

$$a=\frac{4}{15},\qquad b=\frac{4}{5},\qquad T_2(x)=\frac{4}{15}+\frac{4}{5}x\ .$$

7.3. Аппроксимация методом наименьших квадратов

Пусть дана функция $f(x)$ на отрезке $[a,\ b]$.

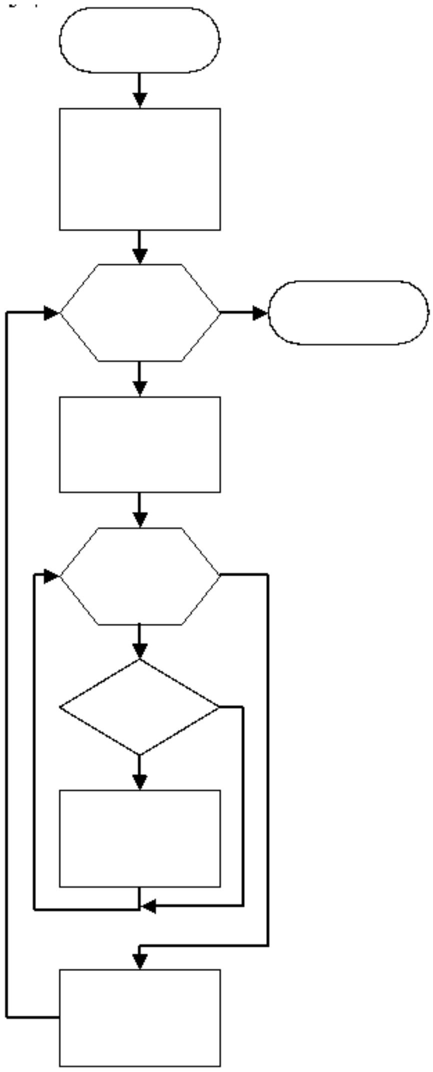
Разобьем отрезок с помощью узлов

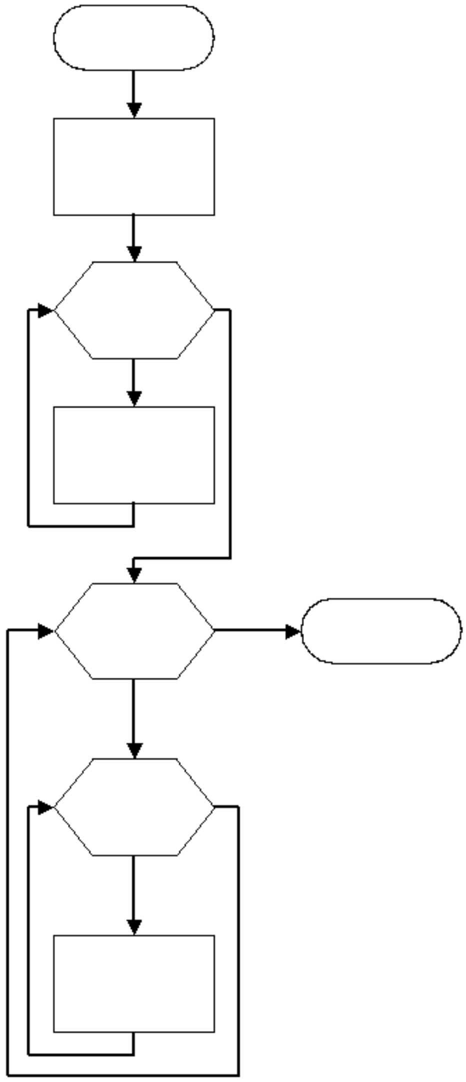
$a\le x_0<x_1<...<x_n\le b$.

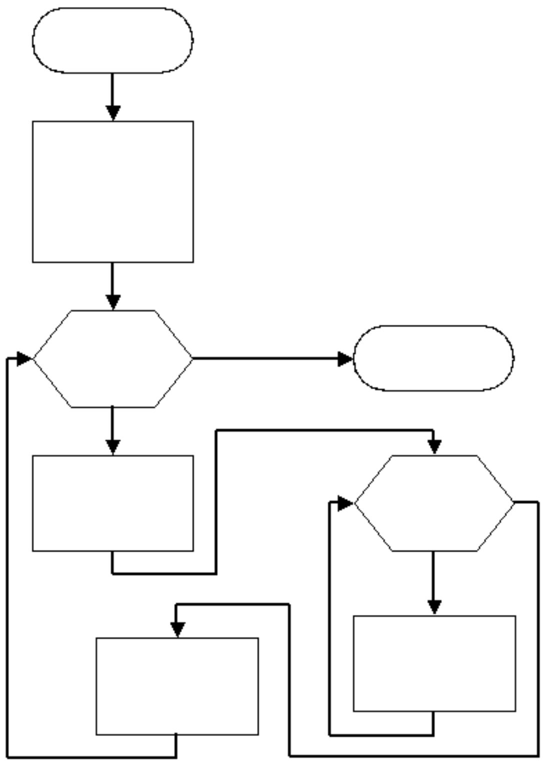
Пусть $y_0,y_1,...,y_n$ – значение функции *f(x)* в узлах.

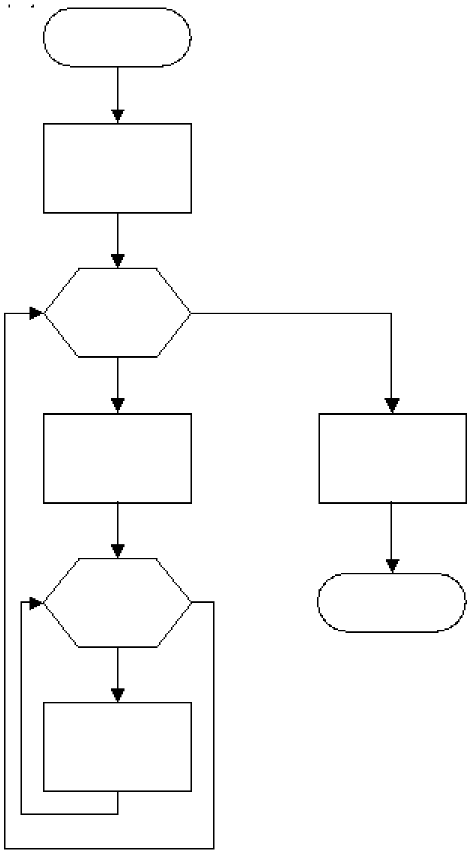
Если n – большое число, то интерполяционный $L_n(x)$ – многочлен высокой степени. Зачастую неудобно использовать многочлены очень высокой степени. Очевидно, мы можем отказаться от использования части узлов и тем самым понизить степень интерполяционного многочлена, но тогда теряется

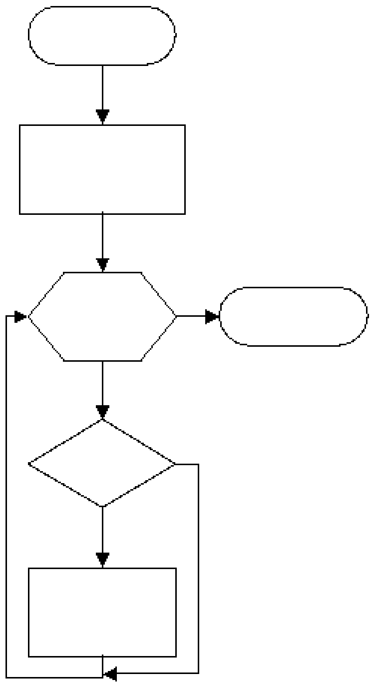
3. АЛГОРИТМ РЕШЕНИЯ

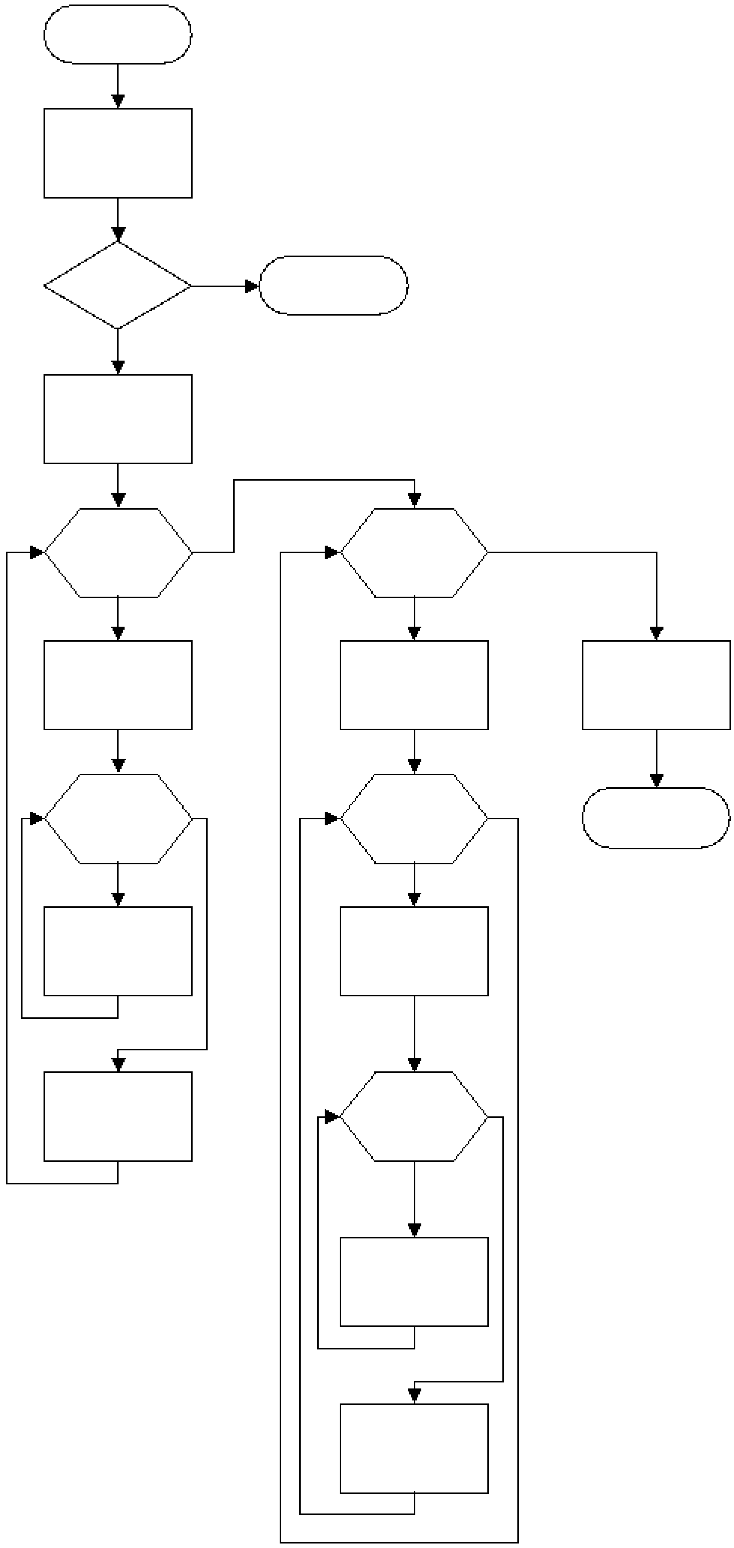












4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.core.multiarray import dot


def input():
    x = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
    p = [0.0, 0.41, 0.79, 1.13, 1.46, 1.76, 2.04, 2.3, 2.55, 2.79, 3.01]
    k = 11
    m = 2.53
    y = [ (p[i] + ((-1) ** k) * m) for i in range(len(x))]
    dots = list(zip(x, y))
    return dots

# Многочлен Лагранжа
def Lagrange(dots):
    n = len(dots)                     # Число точек
    (x, y) = map(list, zip(*dots)) # Списки X и Y отдельно
    polynom = np.poly1d([0])        # polynom = 0
    for i in range(n):
        p = np.poly1d([1]) # p = 1
        for j in range(n):
            if j != i:     # пропускаем j-ое
                p *= np.poly1d([ 1, -x[j] ]) / (x[i] - x[j]) # p *= (X-
Xj)/(Xi-Xj)
        polynom += y[i] * p                                # polynom +=
P*Yi
    return polynom



# Разделенные разности
def DividedDifferences(x):
    n = len(x)
    diffs = [[None for j in range(n - i)] for i in range(n)]

    for i in range(n):
        diffs[i][0] = y[i]

    for j in range(1, n):
        for i in range(n - j):
            diffs[i][j] = ((diffs[i + 1][j - 1] - diffs[i][j - 1]) / (x[i +
j] - x[i]))

    return diffs


# Многочлен Ньютона
def Newton(dots):
    n = len(dots)                     # Число точек
    (x, y) = map(list, zip(*dots)) # Списки X и Y отдельно

    diffs = DividedDifferences(x)   # Разделенные разности
    polynom = np.poly1d([0]) # polynom = 0
    for i in range(n):
        p = np.poly1d([1])   # p = 1
        for j in range(i):
            p *= np.poly1d([ 1, -x[j] ]) # p *= (X - Xj)
        polynom += p * diffs[0][i]       # polynom += p * fn(x0,..,xi)

    return polyno
```

```
# МНК при m == n
def Simple(dots):
    n = len(dots)
    (x, y) = map(list, zip(*dots))
    A = []
    for i in range(n):
        A.append([])
        for j in range(n):
            A[i].append(x[i] ** j)
    polynom = np.poly1d(np.linalg.solve(A, y)[::-1])
    return polynom

# Аппроксимация методом наименьших квадратов
def Squares(dots, m = None):
    n = len(dots) - 1
    if m is None:
        m = n
    assert 0 <= m <= n
    if m == n:
        return Simple(dots)

    (x, y) = map(list, zip(*dots))

    b = []
    for k in range(m + 1):
        s = 0
        for i in range(n + 1):
            s += y[i] * (x[i] ** (m - k))
        b.append(s)

    A = []
    for k in range(m + 1):
        A.append([])

        for j in range(m + 1):
            s = 0
            for i in range(n + 1):
                s += x[i] ** (2 * m - k - j)
            A[k].append(s)

    polynom = np.poly1d(np.linalg.solve(A, b))
    return polynom

# Общая погрешность
def get_error(method, dots):
    func = method(dots) - Squares(dots, 10)
    der = np.polyder(func)

    max_error = 0.0
    for root in np.roots(der):
        if x[0] <= root <= x[-1]:
            max_error = max(max_error, abs(np.polyval(func, root)))

    return max_error
```

```
dots = input()
(x, y) = map(list, zip(*dots))
print("(x,y) =", dots, '\n')

lagrange = Lagrange(dots)
print("Полином Лагранжа =")
print(lagrange, '\n')

newton = Newton(dots)
print("Полином Ньютона =")
print(newton, '\n')

squares = Squares(dots)
print("Полином МНК =")
print(squares, '\n')

xdot = 0.47
width = 25

print(f"Полином Лагранжа ({xdot}) =".ljust(width), lagrange(xdot))
print(f"Полином Ньютона ({xdot}) =".ljust(width), newton(xdot))
print(f"Молином МНК ({xdot}) =".ljust(width), squares(xdot))
#print(f"Молином МНК ({xdot}) =", "{:.4f}".format(squares(xdot)))

print("|Лагранж - Ньютон| =".ljust(width), abs(lagrange(xdot) -
newton(xdot)))
print("|Лагранж - МНК| =".ljust(width), abs(lagrange(xdot) - squares(xdot)))
print("|Ньютон - МНК| =".ljust(width), abs(newton(xdot) - squares(xdot)))

print("Погрешность Лагранж: ".ljust(width), get_error(Lagrange, dots))
print("Погрешность Ньютон: ".ljust(width), get_error(Newton, dots))
#print(f"Inaccuracy({xdot}) = ", Inaccuracy(x, xdot))


plotdots = 10**4
plt.plot(x, y, 'og',linewidth=5)

xplot = np.linspace(min(x), max(x), plotdots)

yplot = [squares(xdot) for xdot in xplot]
plt.plot(xplot, yplot, 'r',linewidth=4) # Красный

yplot = [lagrange(xdot) for xdot in xplot]
plt.plot(xplot, yplot, 'b--',linewidth=4) # Пунктир голубой

yplot = [newton(xdot) for xdot in xplot]
plt.plot(xplot, yplot, 'm:', linewidth=4) # Фиолетовый точками


plt.show()
```

5. ТЕСТОВЫЕ ПРИМЕР

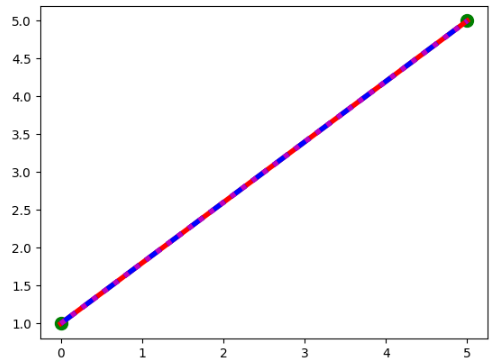
Построить интерполяционные многочлены в формах Лагранжа и Ньютона, построить многочлен наилучшего приближения по методу наименьших квадратов.

Цвета: 1 – голубой, 2 – фиолетовый, 3 – красный.

Тестовый пример 1

x	*0*	*5*
y	*1*	*5*

Вывод программы:

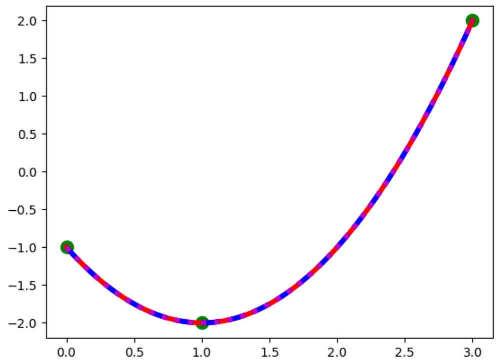


$$y = 0.8x + 1$$

Тестовый пример 2

x	*0*	*1*	*3*
y	*-1*	*-2*	*2*

Вывод программы:

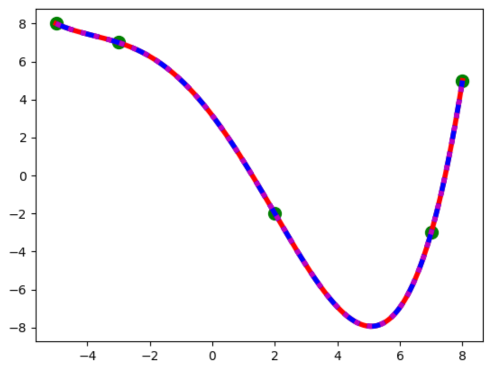


$$y = x^2 - 2x - 1$$

Тестовый пример 3

x	*-5*	*-3*	*2*	*7*	*8*
y	*8*	*7*	*-2*	*-3*	*5*

Вывод программы:

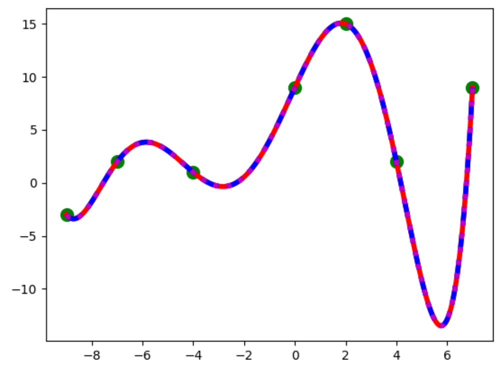


$$y = 0.006222x^4 + 0.02259x^3 -$$
$$-0.2804x^2 - 2.158x + 3.157$$

Тестовый пример 4

x	*-9*	*-7*	*-4*	*0*	*2*	*7*	*4*
y	*-3*	*2*	*1*	*9*	*15*	*9*	*2*

Вывод программы:



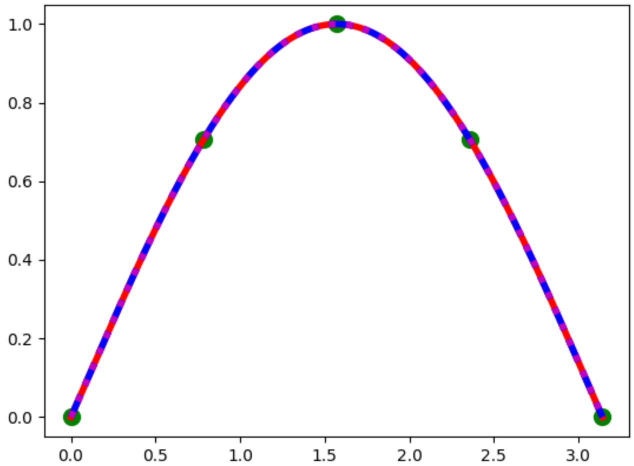
$$y = 0.0005695x^6 + 0.006709x^5 -$$
$$-0.02497x^4 - 0.4247x^3 - 0.2149x^2 +$$
$$+5.203x + 9$$

Тестовый пример 5

x	0	$\pi/4$	$\pi/2$	$3\pi/4$	π
$y = sin\ sin\ x$					

Построить многочлены в формах Лагранжа и Ньютона, многочлен наилучшего приближения. Найти значение многочлена в точке $x = \frac{3\pi}{8}$.

Вывод программы:




```
Полином Лагранжа =
         4          3           2
0.03758 x - 0.2361 x + 0.05829 x + 0.982 x

Полином Ньютона =
         4          3           2
0.03758 x - 0.2361 x + 0.05829 x + 0.982 x

Полином МНК =
         4          3           2
0.03758 x - 0.2361 x + 0.05829 x + 0.982 x

Полином Лагранжа (1.1780972450961724) =  0.9240958691207962
Полином Ньютона (1.1780972450961724) =   0.9240958691207961
Полином МНК (1.1780972450961724) =       0.9240958691207964
|Лагранж - Ньютон| =                     1.1102230246251565e-16
|Лагранж - МНК| =                        2.220446049250313e-16
|Ньютон - МНК| =                         3.3306690738754696e-16
Погрешность Лагранж (отн. МНК):          2.228693453924558e-16
Погрешность Ньютон (отн. МНК):           6.397903718862319e-16
```

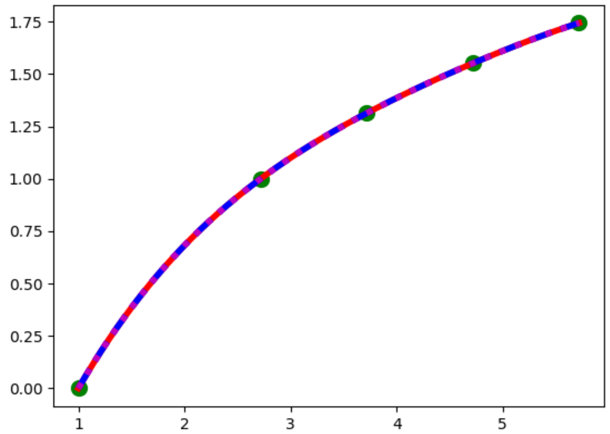
$$sin\ \boldsymbol{sin}\left(\frac{3\pi}{8}\right) = \frac{\sqrt{\mathbf{2} + \sqrt{\mathbf{2}}}}{\mathbf{2}} \approx \mathbf{0.92388}$$

Тестовый пример 6

x	1	e	$e+1$	$e+2$	$e+3$
$y = \ln \ln x$					

Построить многочлены в формах Лагранжа и Ньютона, многочлен наилучшего приближения. Найти значение многочлена в точке $x = e + 1/2$.

Вывод программы:




```
Полином Лагранжа =
           4           3          2
-0.002466 x + 0.04647 x - 0.3488 x + 1.44 x - 1.135

Полином Ньютона =
           4           3          2
-0.002466 x + 0.04647 x - 0.3488 x + 1.44 x - 1.135

Полином МНК =
           4           3          2
-0.002466 x + 0.04647 x - 0.3488 x + 1.44 x - 1.135

Полином Лагранжа (3.218281828459045) =    1.170147855669581
Полином Ньютона (3.218281828459045) =    1.1701478556695735
Полином МНК (3.218281828459045) =        1.1701478556695748
|Лагранж - Ньютон| =                     7.549516567451064e-15
|Лагранж - МНК| =                        6.217248937900877e-15
|Ньютон - МНК| =                         1.3322676295501878e-15
Погрешность Лагранж (отн. МНК):          6.995285359459231e-15
Погрешность Ньютон (отн. МНК):           2.9392912740145552e-15
```

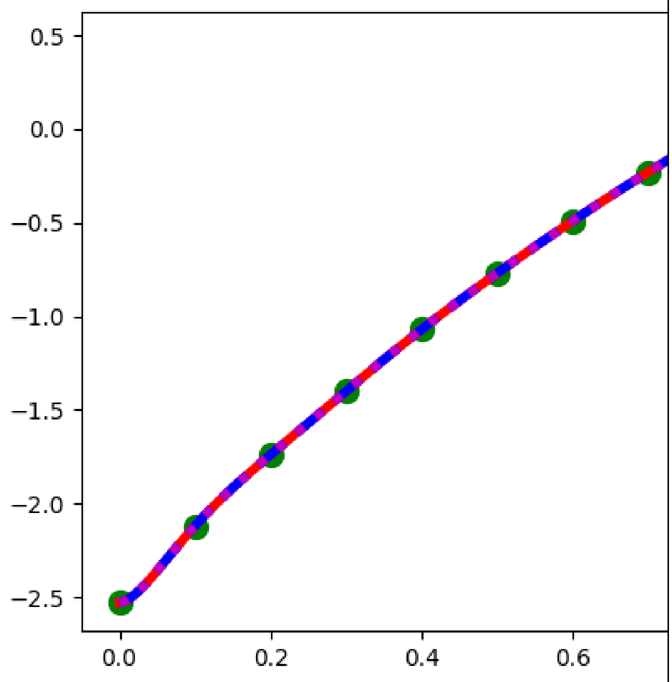
$$\ln \ln \left(e + \frac{1}{2}\right) \approx 1.16885$$

6. ЗАДАНИЕ

Вариант 11

Построить интерполяционные многочлены в форме Лагранжа[1] и Ньютона[2]. Оценить погрешность. Вычислить значение функции в точке 0.47 с помощью интерполяционного многочлена и многочлена наилучшего приближения[3]. Сравнить значения.

<table>
<tr><td>x</td><td>0.0</td><td>0.1</td><td>0.2</td><td>0.3</td><td>0.4</td><td>0.5</td><td>0.6</td><td>0.7</td><td>0.8</td><td>0.9</td><td>1.0</td></tr>
<tr><td>y</td><td>-2.53</td><td>-2.12</td><td>-1.74</td><td>-1.40</td><td>-1.07</td><td>-0.77</td><td>-0.49</td><td>-0.23</td><td>0.02</td><td>0.26</td><td>0.48</td></tr>
<tr><td colspan="8">Значение в точке x = 0.47</td></tr>
<tr><td colspan="4">Полином Лагранжа</td><td colspan="4">Полином Ньютона</td></tr>
<tr><td colspan="4">≈ - 0.8573</td><td colspan="4">≈ - 0.8573</td></tr>
<tr><td colspan="8">Многочлен наилучшего приближения</td></tr>
<tr><td>Степень</td><td>1</td><td>2</td><td colspan="2">3</td><td>5</td><td>8</td><td>10</td></tr>
<tr><td>Значение ≈</td><td>-0.9612</td><td>-0.8611</td><td colspan="2">-0.8587</td><td>-0.8596</td><td>-0.8570</td><td>-0.8573</td></tr>
<tr><td>|МНП – Лагранж| ≈</td><td>0.1039</td><td>0.0038</td><td colspan="2">0.0014</td><td>0.0023</td><td>0.0003</td><td>0.0000</td></tr>
<tr><td>|МНП – Ньютон| ≈</td><td>0.1039</td><td>0.0038</td><td colspan="2">0.0014</td><td>0.0023</td><td>0.0003</td><td>0.0000</td></tr>
</table>



Цвета: 1 – голубой, 2 – фиолетовый, 3 – красный

Вывод программы:

```
Полином Лагранжа =
      10             9              8             7             6
3279 x  - 1.682e+04 x + 3.714e+04 x - 4.611e+04 x + 3.532e+04 x
              5         4          3          2
 - 1.719e+04 x + 5268 x - 966.3 x + 92.75 x + 0.6282 x - 2.53

Полином Ньютона =
      10             9              8             7             6
3279 x  - 1.682e+04 x + 3.714e+04 x - 4.611e+04 x + 3.532e+04 x
              5         4          3          2
 - 1.719e+04 x + 5268 x - 966.3 x + 92.75 x + 0.6282 x - 2.53

Полином МНК =
      10             9              8             7             6
3279 x  - 1.682e+04 x + 3.714e+04 x - 4.611e+04 x + 3.532e+04 x
              5         4          3          2
 - 1.719e+04 x + 5268 x - 966.3 x + 92.75 x + 0.6282 x - 2.53

Полином Лагранжа (0.47) =                -0.8573479203111825
Полином Ньютона (0.47) =                 -0.8573479202912302
Полином МНК (0.47) =                     -0.8573479202915222
|Лагранж - Ньютон| =                     1.9952262064748538e-11
|Лагранж - МНК| =                        1.9660273409272122e-11
|Ньютон - МНК| =                         2.9198865547641617e-13
Погрешность Лагранж (отн. МНК):          5.059606938803385e-11
Погрешность Ньютон (отн. МНК):           4.382271308985439e-12
```

7. ВЫВОД

Таким образом, в ходе выполнения лабораторной работы была изучена интерполяция функций с помощью интерполяционных многочленов Лагранжа и Ньютона, аппроксимация методом наименьших квадратов. Составлен алгоритм и программа нахождения многочленов соответствующими методами, проверена правильность работы на тестовых примерах. Согласно заданному варианту построены интерполяционные многочлены в форме Лагранжа и Ньютона, построен многочлен наилучшего приближения по методу наименьших квадратов, вычислено значение функции в точке.