Міністерство освіти та Науки України

Національний технічний університет “Харківський політехнічний інститут”

Кафедра «Програмна інженерія і інформаційні технології управління»

Індивідуальне завдання №4

з дисципліни

«Чисельні методи»

Виконали:

студенти групи КН-34б

Рузняєв Д.А.

Костюк I.Ю.

Перевірив:

Проф. Гужва В.О.

Харків

2016

**ПОЛИНОМ ЛАГРАНЖА**

Дано : n + 1 значение функции дейcтвительного переменного

y = f(x) , x0 ... xn

y0 = f(x0)…yn=f(yn)

Выберем в качестве функции полином Pn(x) степени не выше n, значения которого совпадает со значением функции f(x) в узлах интерполяции, т.е.

Pn(x0) = y0…Pn(xn)=yn (1)

Покажем ,что существует единственное решение

Пусть большое Pn(x) = а0 + а1х + а2х^2 + ...+ anxn

Коэффициенты ai можно определить из системы уравнений.

A0 +a1x0 + a2x0^2 + …anx0^n = y0

A0 + a1x1 + a2x1^2 + … anx1^n = y0 (2)

A0 + a1xn + a2xn^2 +… anxn^n = y0

Из системы (2) необходимо определить коэффициенты а0...аn

Известно ,что если ранг матрицы системы равен числу неизвестных ,следовательно система имеет единственное решение. Предполагают :

Все хi (i от 0 до n) различны ,следовательно определитель системы W:

W = 1 x0 x0^2 …x0^n

1 x1 x1^2 …x1^n

1 xn xn^2 …xn^n

W не равно 0, следовательно у системы 1 решение.

Полином построить можно в общем виде :

Pn(x) = y0((x-1)(x-2)…(x-xn))/((x0-x1)(x0-x2)…(x0-xn)) + y1((x-x0)(x-x2)…(x-xn))/((x1-x0)(x1-x2)…(x1-xn)) + …+ yn((x-x0)(x-x1)…(x-xn-1))/((xn-x0)(xn-x1)…(xn – xn-1))

Условие 1 для этого полинома выполняется .

При определении полинома Pn(x) использует базис. Лагранжевы коэффициенты Lj(x), x = 1,n , степени n таких, что

Lj(xi) = 1,если I = j 0,

если I не равно j

Тогда :

Pn(x) = y0L0(x) + y1L1(x) +…+ ynLn(x) = ∑yjLj(x), j= от 0 до n

Определим Lj(x) ,как многочлен степени L1 обращающихся в 0 в точках

х0,х1...хi-1

Xi+1…xn

И равный 1 в точке xi.

Полагая, что Lj(х) = а(х – х0)(х-х1)...(х-хi)(x-xi+1)…(x – xn), которые при x = xi

Li(x) = A(xi-x0)…(xi-xi-1)(xi-x+1)…(xi-xn) =1

A = 1/(xi-x0)(xi-x1)…(xi-xi-1)(xi-xi+1)…(xi-xn)

Lj(x)=((x-x0)(x-x1)…(x-xi-1)(x-xi+1)…(x-xn))/((xi-x0)(xi-x1)…(xi-xi-1)(xi-xi+1)…(xi-xn)) (5)

Pn(x) = принимает значение в точках xi и равно 0 во всех остальных точках.

Xj (где xj не равно xi)

Введём :

W(x) = (x-x0)(x-x1)…(x-xi)…(x-xn)

Возьмем производную по х

W’(x) = (x-x1)(x-x2)…(x-xn)+ (x-x0)(x-x2)…(x-xn)+…+(x-x0)(x-x1)…(x-xn-1)

Отсюда можно получить

W’(xi) = (xi-x0)(xi-x1)…(xi-xi-1)(xi-xi+1)…(xi-xn)

Поэтому выражение (5):

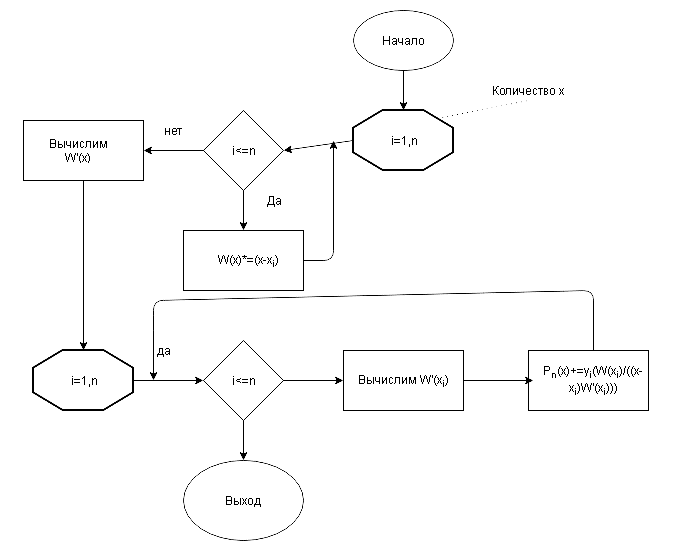
Li(x) = w(x)/((x-xi)w’(xi))

Тогда Полином Лагранжа

Pn(x) = ∑yiLi(x) = ∑yi (w(x)/((x-xi)w’(xi))

Интерполяционный полином Лагранжа удобно применять ,когда ведется интерполирование по одним и тем же узлом. Для них заранее можно составить коэффициенты ,т.к. они не зависят от функции f(x). Но если полином не достаточно хорошо аппроксимирует заданную функцию и для улучшения приближения требуется повысить его степень (увеличить число узлов интерполяций).Следовательно поленом Лагранжа прийдётся перевычеслить .

БЛОК-СХЕМА ПОЛИНОМА ЛАГРАНЖА



ИСХОДНЫЙ КОД МЕТОДА ПОЛИНОМ ЛАГРАНЖА

public class Lagrange {  
 StringBuilder stroke=new StringBuilder("Интерполяционный полином Лагранжа");  
 double[]xi,yi,dw;  
 double w,L;  
 public StringBuilder getStroke() {  
 return stroke;  
 }  
 public Lagrange (double []xi,double [] yi){  
 this.xi=xi;  
 this.yi=yi;  
 }  
 public double[] mainMehod(){  
 dw=new double[xi.length];  
 double a=1;  
 for (int i=0;i<xi.length;i++){  
 a=1;  
 for(int j=0;j<xi.length;j++){  
 if(j!=i){  
 a\*=xi[i]-xi[j];  
 }  
 }  
 dw[i]=a;  
 }  
 return dw;  
 }  
 public void printPolynom(){  
 //System.out.print("Pn(x) = ");  
 stroke.append("Pn(x) = ");  
 for (int i=0;i<xi.length;i++) {  
 if(i==0) {  
 //System.out.print(yi[i] + "\*(");  
 stroke.append(yi[i] + "\*(");  
 }  
 else{  
 if(yi[i]<0){  
 //System.out.print(yi[i] + "\*(");  
 stroke.append(yi[i] + "\*(");  
 }  
 else{  
 //System.out.print(" + "+yi[i] + "\*(");  
 stroke.append(" + "+yi[i] + "\*(");  
 }  
 }  
 for (int j = 0; j < xi.length; j++) {  
 if(j!=i){  
 if(xi[j]<0) {  
 // System.out.print("(x + " + xi[j] \* (-1) + ")");  
 stroke.append("(x + " + xi[j] \* (-1) + ")");  
 }  
 else {  
 // System.out.print("(x - " + xi[j] + ")");  
 stroke.append("(x - " + xi[j] + ")");  
 }  
 }  
 }  
 if(dw[i]<0) {  
 stroke.append(")/(" + dw[i]+")");  
 }  
 else  
 stroke.append(")/" + dw[i]);  
 }  
 }  
 public static void main(String[] args) {  
 double[]xi,yi;  
 int length;  
 Scanner s = new Scanner(System.in);  
 System.out.print("Введите количество узлов: ");length=s.nextInt();  
 xi=new double [length];  
 yi=new double [length];  
 for (int i=0;i<length;i++) {  
 System.out.print("x" + i + " = ");xi[i]=s.nextDouble();  
 System.out.print("y" + i + " = " );yi[i]=s.nextDouble();  
 }  
 Lagrange lag = new Lagrange(xi,yi);  
 lag.getStroke().append("\nИсходные данные:\n");  
 for (int i=0;i<length;i++) {  
 lag.getStroke().append("x" + i + " = "+xi[i]+"\t y" + i + " = "+yi[i]+"\n");  
 }  
 lag.mainMehod();  
 lag.printPolynom();  
 System.out.println(lag.getStroke());  
 }  
}

РУЧНОЙ ПРОСЧЕТ МЕТОДОМ ПОЛИНОМА ЛАГРАНЖА

*Матрица студента Рузняева Д.А*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X | 7 | 10 | 13 | 16 | 19 |
| Y | 2 | 4 | 14 | 20 | 10 |

W(x) = (x - 7) \* (x - 10) \* (x - 13) \* (x - 16) \* (x - 19)

W'(x) = (x - 10) \* (x - 13) \* (x - 16) \* (x - 19) + (x - 7) \* (x - 13) \* (x - 16) \* (x - 19) + (x -7) \* (x - 10) \* (x - 16) \* (x - 19) + (x -7) \* (x - 10) \* (x - 13) \* (x - 19) + (x -7) \* (x - 10) \* (x - 13) \* (x - 16)

W'(7) = 1944

W'(10) = -486

W'(13) = 324

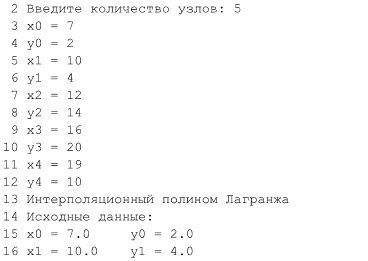
W'(16) = -486

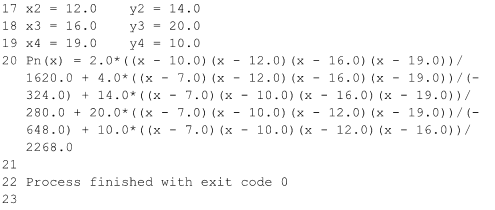
W'(19) = 1944

Найдем полином P5

P5(x) = (2/1944)\*(x - 10)(x - 13)(x - 16)(x - 19) + (4/-486)\*(x -7)(x - 13)(x - 16)(x - 19) + (14/324)\*(x -7)(x - 10)(x - 16)(x - 19) + (20/-486)\*(x -7)(x - 10)(x - 13)(x - 19) + (10/1944)\*(x -7)(x - 10)(x - 13)(x - 16)= (1/27)(x3 + x2 + 8\*x + 33)

ПРОСЧЕТ С ПОМОЩЬЮ ПРОГРАММЫ МЕТОДОМ ПОЛИНОМ ЛАГРАНЖА





МЕТОД НЬЮТОНА ДЛЯ РАЗДЕЛЬНЫХ РАЗНОСТЕЙ ПОРЯДКА

Этот полином допускает уточнение результатов интерполяции последовательным прибавлением новых узлов. Пусть для функции f(x) заданы необязательно равностоящие узлы интерполяции x0…xn и соответственное значение функции f(x0)…F(xn).

Разделенными разностями 1 порядка называют величины имеющие смысл средних скоростей изменения функции и вычисляемые по формулам.

F(x0x1) = (f(x1) – f(x0))/(x1-x0)

F(xn-1xn) = (f(xn) – f(xn-1))/xn – xn-1

Это разделенные разности 1 порядка .

Aнологично 2 порядка.

F(x0x1x2) = (f(x1x2) – f(x0x1))/(x2 – x0)

F(x1x2x3) = (f(x2x3) – f(x1x2))/(x2-x0)

F(xn-2xn-1xn) = (f(x1…xn) – f(xn-2xn-1))/(xn – xn-2)

Так же определяем разделенные разности 4, 5... порядка

F(x0x1…xn) = (f(x1…xn) – f(x0 + …xn-1))/(xn – xo)

Разделенные разности удобно записать в виде таблицы.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X | y | Разделенные разности порядка : | | | |
| 1 | 2 | 3 | 4 |
| X0 | F(x0) |  |  |  |  |
|  |  | F(x0,x1) |  |  |  |
| X1 | F(x1) |  | F(x0x1x2) |  |  |
|  |  | F(x1,x2) |  | F(x0x1x2x3) |  |
| X2 | F(x2) |  | F(x1x2x3) |  | F(x0x1x2x3x4) |
|  |  | F(x2,x3) |  | F(x1x2x3x4) |  |
| X3 | F(x3) |  | F(x2x3x4) |  |  |
|  |  | F(x3,x4) |  |  |  |
| X4 | F(x4) |  |  |  |  |

Отметим свойства разделенных разностей :

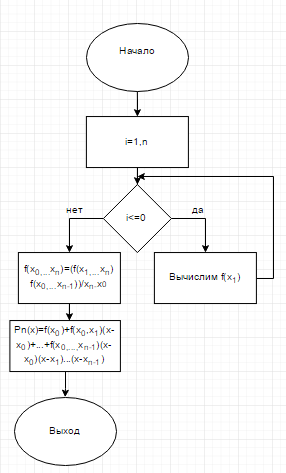
1 Если функция F(x) n раз неприрывна дифференцируется ,следовательно выполняется соотношение

F(x0x1..xn) = 1/n! Fn(Ѯ), где x0 < Ѯ < xn

2 Разделенные разности n-го порядка от алгоритмов полинома n-ой степени постоянны,а разности более высоких порядков = 0.

Это свойство позволяет выбрать степень интерполяции полинома такой ,чтобы они совпадали с порядком практически постоянных разделенных разностей.

БЛОК-СХЕМА МЕТОД НЬЮТОНА ДЛЯ РАЗДЕЛЬНЫХ РАЗНОСТЕЙ ПОРЯДКА



ИСХОДНЫЙ КОД МЕТОД НЬЮТОНА ДЛЯ РАЗДЕЛЬНЫХ РАЗНОСТЕЙ ПОРЯДКА

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Newtone

{

class Program

{

static void Main(string[] args)

{

/\*double [][] a = new double [][]{

{2,1,0,0,0,0},

{4,3.5,0,0,0,0},

{6,11,0,0,0,0},

{8,26.5,0,0,0,0},

{10,53,0,0,0,0} };\*/

double[][] a = new double[5][];

for (int i = 0; i < 5; i++)

a[i] = new double[6];

a[0] = new double[] { 2.5, 3, 0, 0, 0, 0 };

a[1] = new double[] { 3.5, 6, 0, 0, 0, 0 };

a[2] = new double[] { 4.5, 21, 0, 0, 0, 0 };

a[3] = new double[] { 5.5, 30, 0, 0, 0, 0 };

a[4] = new double[] { 6.5, 15, 0, 0, 0, 0 };

for( int j = 2;j<6;j++){

for(int i =0; i < 4; i++) {

if(a[i+1][j-1]!= 0){

a[i][j] = a[i+1][j-1] - a[i][j-1];

Console.WriteLine(a[i+1][j-1] + " - " + a[i][j-1] + " = " + a[i][j]);}

}}

for( int i = 0;i<5;i++){

for(int j =0; j < 6; j++){

Console.Write(a[i][j] + " ");

}

Console.WriteLine();

}

Console.Write("Pn(x)=");

for(int i = 1; i <5;i++){

if(i!=4)

Console.Write(a[0][i]+ "/("+ fact(i-1)+" \* " + ((int)(Math.Pow(3,( i-1)))+ ")"+x(i,a)+" + "));

else

Console.Write(a[0][i]+ "/("+ fact(i-1)+" \* " + ((int)(Math.Pow(3,( i-1)))+ ")"+x(i,a)+""));

}

Console.WriteLine();

}

public static int fact(int n){

int sum = 1;

for(int i = 1; i <=n;i++)

sum \*= i;return sum;}

public static String x (int i,double[][] a){

String x="";

if (i == 2) {

x += "(x - " + a[0][0] + ")";

return x;

}if (i == 3){

for(int j = 0; j < i; j++)

x += "(x - " + a[j][0]+")";

return x;}

if (i == 4){

for(int j = 0; j < i; j++)

x += "(x - " + a[j][0]+")";

return x;}

if (i == 5){

for(int j = 0; j < i; j++)

x += "(x - " + a[j][0]+")";

return x; else return "";}}

РУЧНОЙ ПОДСЧЕТ МЕТОДА НЬЮТОНА ДЛЯ РАЗДЕЛЬНЫХ РАЗНОСТЕЙ ПОРЯДКА

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X | 1 | 2 | 3 | 4 | 5 |
| Y | 1.5 | 3,5 | 9,5 | 22,5 | 45,5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Xі | F(xі) | Первого | Второго | Третьего | четвертого |
| X0  X1  X2  X3  X4 | 1  2  3  4  5 | 1,5  3,5  9,5  22,5  45,5 | 3,5 - 1.5 = 2  9.5 - 3.5 = 6  22.5 - 9.5 = 13  45.5 - 22.5 = 23 | 6 – 2 = 4  13 -6 = 7  23 – 13 = 10 | 7 – 4 = 3  10 – 7 = 3 | 3 – 3 = 0 |

P(x)= 1,5/(1 \* 1) + 2/(1 \* 3)(x - -6) + 4/(2 \* 9)(x - -6)(x - -4)(x - -2) + 3/(6 \* 27)(x - -6)(x - -4)(x - -2)(x - 0)

ПРОСЧЕТ С ПОМОЩЬЮ ПРОГРАММЫ МЕТОДА НЬЮТОНА ДЛЯ РАЗДЕЛЬНЫХ РАЗНОСТЕЙ ПОРЯДКА

