БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра программного обеспечения информационных технологий

Факультет ФИНО

Специальность ПОИТ

Индивидуальная практическая работа № 1

по дисциплине «Дискретная математика»

Выполнил студент: Засимук Н. С.

Зачетная книжка 3910014

Минск 2017

1. **Задание**

Изучить интерполяцию функции с использованием интерполяционных многочленов.

1. **Многочлен Лагранжа для первых семи точек**

Для точек x1..x7 интерполяционный полином Лагранжа имеет вид:

|  |  |
| --- | --- |
|  | (1) |

Подставляем в полином (1) промежуточные точки 1,5; 2,5; 3,5; 4,5; 5,5; 6,5.

В итоге получаем:

L(1,5)7=2,002; L(2,5)7=1,918; L(3,5)7=1,91; L(4,5)7=1,81; L(5,5)7=1,791; L(6,5)7=1,943.

1. **Программа аппроксимации табличных функций с помощью интерполяционных полиномов Лагранжа и кубическими сплайнами**

unit inter;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Grids, StdCtrls, ExtCtrls;

type

TForm1 = class(TForm)

Button1: TButton;

StringGrid1: TStringGrid;

Edit2: TEdit;

Label1: TLabel;

Label2: TLabel;

Edit3: TEdit;

Label3: TLabel;

RadioButton1: TRadioButton;

RadioButton2: TRadioButton;

PaintBox1: TPaintBox;

Edit1: TEdit;

procedure Button1Click(Sender: TObject);

procedure Edit1Change(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

function LagrangeInterpolation(q:extended):extended;

function CubeSplineInterpolation(q:extended):extended;

procedure DrawAnalyticDependence;

end;

var

Form1: TForm1;

Func:array[1..2,1..20] of extended;

ComX,ComY,maxX,maxY,minX,minY,shX,shY:extended;

NumIntNodes:integer;

implementation

{$R \*.dfm}

function TForm1.LagrangeInterpolation(q:extended):extended;

var

i,j:integer;

l,s:real;

c:real;

begin

l:=0;

for i:=1 to NumIntNodes do

begin

s:=1;

for j:=1 to NumIntNodes do

if j<>i then

s:=s\*(q-Func[1,j])/(Func[1,i]-Func[1,j]);

l:=l+Func[2,i]\*s;

end;

LagrangeInterpolation:=l;

PaintBox1.Canvas.MoveTo((Round((Func[1,1]-minx)\*ComX)),359-Round((Func[2,1]-miny)\*ComY));

PaintBox1.Canvas.Pen.Color:=clRed;

c:=Func[1,1]+0.001;

while(c<Func[1,NumIntNodes]) do

begin

l:=0;

for i:=1 to NumIntNodes do

begin

s:=1;

for j:=1 to NumIntNodes do

if j<>i then

s:=s\*(c-Func[1,j])/(Func[1,i]-Func[1,j]);

l:=l+Func[2,i]\*s;

end;

PaintBox1.Canvas.LineTo((Round((c-minx)\*ComX)),359-Round((l-miny)\*ComY));

c:=c+0.001;

end;

end;

function TForm1.CubeSplineInterpolation(q:extended):extended;

var

S0,SN:real;

count,res:Extended;

A,B,C,D,F:array[1..20]of extended;

h2,h3,p,ctmp,ftmp,z,hi,hi1 : extended; i,j : integer;

alpha,beta:array[1..20] of extended;

begin

for i := 1 to NumIntNodes do

begin

A[i]:=Func[2,i];

end;

c[1]:=0;

c[NumIntNodes]:=0;

alpha[1]:=0;

beta[1]:=0;

for i := 2 to NumIntNodes-1 do

begin

hi:=Func[1,i]-Func[1,i-1];

hi1:=Func[1,i+1]-Func[1,i];

ctmp:=2\*(hi+hi1);

ftmp:=6\*((Func[2,i+1]-Func[2,i])/(Func[1,i+1]-Func[1,i])-(Func[2,i]-Func[2,i-1])/(Func[1,i]-Func[1,i-1]));

z:=hi\*alpha[i-1]+ctmp;

alpha[i]:=-hi1/z;

beta[i]:=(ftmp-hi\*beta[i-1])/z;

end;

for i := NumIntNodes-1 downto 2 do

c[i]:=alpha[i]\*c[i+1]+beta[i];

d[1]:=Func[2,i];

b[1]:=Func[2,i];

for i := NumIntNodes downto 2 do

begin

hi:=Func[1,i]-Func[1,i-1];

d[i]:=(c[i]-c[i-1])/hi;

b[i]:=hi\*c[i]/2-sqr(hi)\*d[i]/6+(Func[2,i]-Func[2,i-1])/hi;

end;

PaintBox1.Canvas.MoveTo((Round((Func[1,1]-minx)\*ComX)),359-Round((Func[2,1]-miny)\*ComY));

PaintBox1.Canvas.Pen.Color:=clBlue;

count:=Func[1,1];

while (count<Func[1,NumIntNodes]) do

begin

for i := 2 to NumIntNodes do

if ((count>=Func[1,i-1])and(count<=Func[1,i])) then

res:=A[i]+B[i]\*(count-Func[1,i])+C[i]\*sqr(count-Func[1,i])/2+D[i]\*sqr(count-Func[1,i])\*(count-Func[1,i])/6;

PaintBox1.Canvas.LineTo((Round((count-minx)\*ComX)),359-Round((res-miny)\*ComY));

count:=count+0.001;

end;

for i := 2 to NumIntNodes do

if ((q>=Func[1,i-1])and(q<=Func[1,i])) then

CubeSplineInterpolation:=A[i]+B[i]\*(q-Func[1,i])+C[i]\*sqr(q-Func[1,i])/2+D[i]\*sqr(q-Func[1,i])\*(q-Func[1,i])/6;

if ((q<Func[1,1])or(q>Func[1,NumIntNodes])) then

ShowMessage('Значение должно быть внутри интервала интерполяции!');

end;

procedure TForm1.Button1Click(Sender: TObject);

var

TestPoint:extended;

i,j:integer;

begin

PaintBox1.Refresh;

PaintBox1.Canvas.Pen.Color:=clBlack;

PaintBox1.Canvas.MoveTo(0,0);

PaintBox1.Canvas.LineTo(0,368);

PaintBox1.Canvas.LineTo(488,368);

PaintBox1.Canvas.LineTo(488,0);

PaintBox1.Canvas.LineTo(0,0);

if(Edit2.Text ='по умолчанию 1') then

Edit2.Text:= '';

if(Edit2.Text <>'') then

TestPoint:=StrToFloat(Edit2.Text)

else

begin

Edit2.Text:= 'по умолчанию 1';

TestPoint:=1;

end;

for i := 1 to NumIntNodes do

for j := 1 to 2 do

Func[j,i]:=StrToFloat(StringGrid1.Cells[j-1,i]);

DrawAnalyticDependence;

if (RadioButton1.Checked) then

Edit3.Text:=FloatToStrF(LagrangeInterpolation(TestPoint),ffFixed,7,4)

else

if (RadioButton2.Checked) then

Edit3.Text:=FloatToStrF(CubeSplineInterpolation(TestPoint),ffFixed,7,4);

end;

procedure TForm1.DrawAnalyticDependence;

var

i,j:Integer;

begin

minX:=Func[1,1];

maxX:=minX;

minY:=Func[2,1];

maxY:=minY;

for i := 2 to NumIntNodes do

begin

if minX>Func[1,i] then minX:=Func[1,i];

if minY>Func[2,i] then minY:=Func[2,i];

if maxX<Func[1,i] then maxX:=Func[1,i];

if maxY<Func[2,i] then maxY:=Func[2,i];

end;

if (maxx<0) then maxx:=0.1;

if (minx>0) then minx:=-0.1;

if (maxy<0) then maxy:=0.1;

if (miny>0) then miny:=-0.1;

ComX:=470/(maxX-minX);

ComY:=350/(maxY-minY);

if minX<0 then

shX:=abs(minX)\*comX else shX:=-minX\*comX;

if minY<0 then

shY:=abs(minY)\*comY else shY:=-minY\*comY;

PaintBox1.Canvas.Pen.Color:=clBlack;

PaintBox1.Canvas.MoveTo((Round((Func[1,1]-minx)\*ComX)),359-Round((Func[2,1]-miny)\*ComY));

for i := 2 to NumIntNodes do

PaintBox1.Canvas.LineTo((Round((Func[1,i]-minx)\*ComX)),359-Round((Func[2,i]-miny)\*ComY));

PaintBox1.Canvas.Pen.Color:=clGreen;

PaintBox1.Canvas.MoveTo(0,359-Round(-miny\*comy));

PaintBox1.Canvas.LineTo(489,359-Round(-miny\*comy));

PaintBox1.Canvas.MoveTo(Round(-minx\*comx),0);

PaintBox1.Canvas.LineTo(Round(-minx\*comx),369);

end;

procedure TForm1.Edit1Change(Sender: TObject);

begin

NumIntNodes:=StrToInt(Edit1.text);

StringGrid1.RowCount:=NumIntNodes+1;

StringGrid1.Cells[0,0]:='x[i]';

StringGrid1.Cells[1,0]:='y[i]';

end;

end.

**Результат работы программы:**



