Digitální učební materiál

Číslo projektu:	CZ.1.07/1.5.00/34.0548
Název školy:	Gymnázium, Trutnov, Jiráskovo náměstí 325
Název materiálu:	VY_32_INOVACE_158_IVT
Autor:	Ing. Pavel Bezděk
Tematický okruh:	Algoritmy
Datum tvorby:	září 2013
Ročník:	4. ročník a oktáva
Anotace:	Algoritmus XVIII. – Algoritmus interpolačních polynomů I. Lagrangeův polynom
Metodický pokyn:	Při výuce nutno postupovat individuálně.
Pokud není uvedeno jinak, je použitý materiál z vlastních zdrojů autora DUM.	

evropský sociální fond v ČR
EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Autor	Ing. Pavel Bezděk		
Vytvořeno dne	18. 9. 2013		
Odpilotováno dne	3. 4. 2014	ve třídě	4.A
Vzdělávací oblast	Informatika a informační a komunikační technologie		
Vzdělávací obor	Informatika a výpočetní technika		
Tematický okruh	Algoritmus		
Téma	Algoritmus XVIII. - Algoritmus interpolačních polynomů I. - Lagrangeův polynom		
Klíčová slova	Algoritmus, Lagrangeův interpolační polynom		

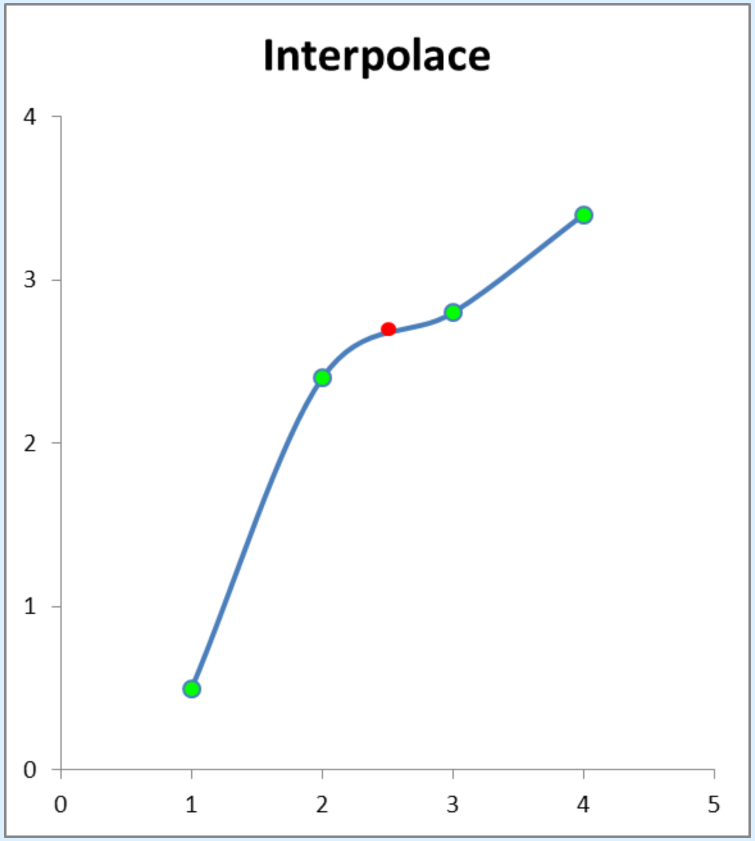
Lagrangeův interpolační polynom

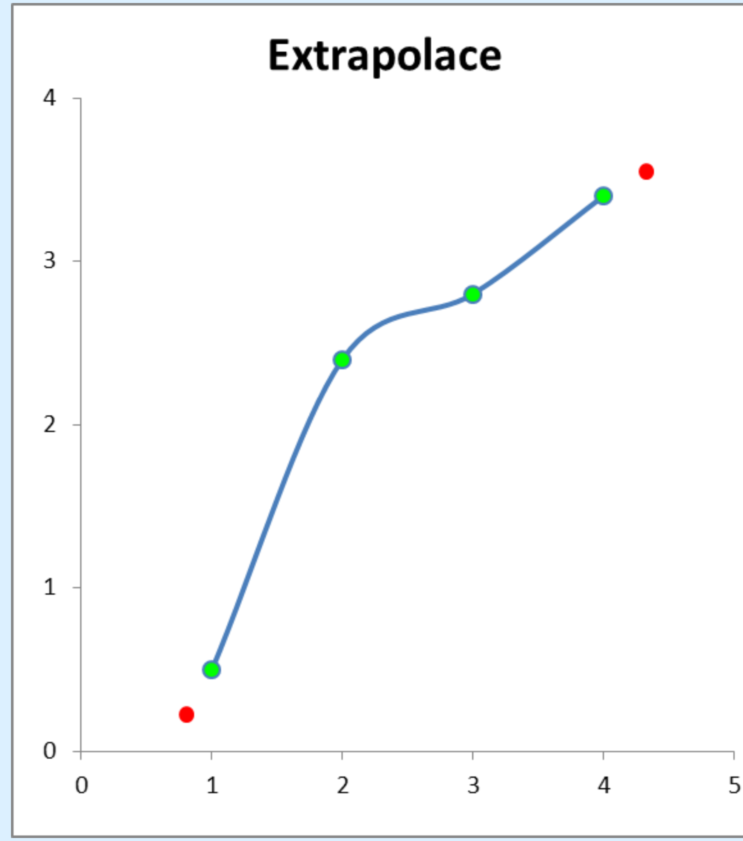
Interpolace x extrapolace

Interpolace je přibližný výpočet hodnot funkce v bodě, které leží uvnitř daného intervalu **int($\mathbf{x_0,x_1,x_2, \dots x_n}$)**.

Extrapolace je přibližný výpočet hodnot funkce v bodě, které leží vně intervalu **int($\mathbf{x_0,x_1,x_2, \dots x_n}$)** z hodnot funkce v krajních bodech $\mathbf{x_0}$ a $\mathbf{x_n}$, popř. i z hodnot funkce v některých vnitřních bodech intervalu int($x_0,x_1,x_2, \dots x_n$). Chyba výpočtu může být velká. Ale pro x velmi blízká koncovým bodům intervalu int($x_0,x_1,x_2, \dots x_n$), lze dostat dobré výsledky.

Interpolace x extrapolace





Vypočteme bod funkce ● mezi známými body funkce ●

Vypočteme bod funkce ● mimo známé body funkce ●

Interpolační podmínky

Předpokládejme nyní, že funkce ***f*** je dána pouze svou ***hodnotou v n+1 bodech*** $x_0, x_1, x_2, \ldots x_n$ a že chceme najít aproximaci φ takovou, která napodobuje chování funkce ***f*** v okolí všech těchto bodů. Např. funkce zadané tabulkou.

O tabulkových bodech x_i $i=0, 1, 2, \ldots n$, v nichž jsou zadané hodnoty $f(x_i)$, budeme předpokládat, že jsou navzájem různé.

Budeme se při interpolaci snažit volit funkci φ tak, aby platilo

$$\varphi(x_i)= f(x_i) \quad i= 0,1,2\ldots n \qquad \textit{(interpolační podmínky)}$$

tj. aby přesně nabývala zadaných funkčních hodnot.

Existuje takový interpolační polynom pro funkci ***f*** v bodech $x_0, x_1, x_2, \ldots x_n$ a budeme ho značit L_n a nazývat ***Lagrangeův interpolační polynom.*** Tabulkovým bodům $x_0, x_1, x_2, \ldots x_n$ budeme říkat ***uzly interpolace***.

Lagrangeův interpolační polynom

$$L_n(x) = \sum_{i=0}^{n} f(x_i)\, l_i(x)$$

Lagrangeův interpolační polynom

(Polynom n-tého stupně: $P_n(x)=p_0+p_1x+p_2x^2+p_3x^3 \ldots\ldots\ldots + p_nx^n$
nejvyšší mocnina x je n-tá – x^n a $p_n \neq 0$)

$$L_n(x)=\sum_{i=0}^{n} f(x_i)\,l_i(x)$$

kde l_i jsou polynomy n-tého stupně takové, že platí $l_i(x_k) = 0$ pro $k \neq i$

$l_i(x_k) = 1$ pro $k=i$

(x_i jsou navzájem různé, i=0,1,2,…n)

i, k = 0,1,2, …n

Polynom l_i má tedy kořeny $x_0, x_1, \ldots x_{i-1}, x_{i+1}, \ldots x_n$
a můžeme jej zapsat ve tvaru:

$$l_i(x)=\frac{(x-x_0)(x-x_1)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n)}{(x_i-x_0)(x_i-x_1)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_n)}$$

Výpočet Lagrangeova polynomu

Funkce *f* dána svými hodnotami v bodech $x_0=0$, $x_1=1$, $x_2=-1$, $x_3=3$,
$f(0)=1$, $f(1)=2$, $f(-1)=2$, $f(3)=0$.

Vypočítáme přibližnou hodnotu f(2) pomocí Lagrangeova interpolačního polynomu

$l_o(x) = ((x-1)(x+1)(x-3))/((0-1)(0+1)(0-3)) = 1/3(x-1)(x+1)(x-3)$ $\quad l_o(2) =-1$

$l_1(x) = ((x-0)(x+1)(x-3))/((1-0)(1+1)(1-3)) = (1/(-4))x(x+1)(x-3)$ $\quad l_1(2) = 3/2$

$l_2(x) = ((x-0)(x-1)(x-3))/((-1-0)(-1-1)(-1-3)) = (1/(-8))x(x-1)(x-3)$ $\quad l_2(2) = 1/4$

$l_3(x) = ((x-0)(x-1)(x+1))/((3-0)(3-1)(3+1)) = (1/24)x(x+1)(x-3)$ $\quad l_3(2) = 1/4$

$L_3(2) = 1\cdot(-1)+2\cdot 3/2+2\cdot 1/4+0\cdot 1/4 = 5/2 \approx f(2)$

Lagrangeův polynom:

$L_3 = 1\cdot l_o(x) +2\cdot l_1(x) +2\cdot l_2(x) +0\cdot l_3(x)$

$L_3 = (1/3)(x-1)(x+1)(x-3)-(1/2)x(x+1)(x-3) –(1/4)x(x-1)(x-3)$

$$L_3(2) = -(5/12)x^3 + x^2 + (5/12)x +1$$

Lineární a kvadratická interpolace

Lineární interpolace n=1, máme známé 2 body funkce, aproximujeme polynomem 1.stupně $P_1 = p_0 + p_1 x$ jeho grafem je přímka spojující 2 známé body funkce, vypočtený bod tedy leží na této přímce. Používáme třeba při práci s tabulkami.

$$L_1 = \frac{(x - x_1)}{(x_0 - x_1)} f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1)$$

Kvadratická interpolace n=2, máme známé 3 body funkce, aproximujeme polynomem 2.stupně $P_2 = p_0 + p_1 x + p_2 x^2$ jeho grafem je parabola spojující 3 známé body funkce, vypočtený bod tedy leží na této parabole. Kvadratická interpolace je přesnější než lineární interpolace (kromě případů, když funkce ***f*** je lineární).

$$L_2 = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2)$$

Výpočet lineární a kvadratické interpolace

Tabulka hodnot funkce f(x)=sin(x), máme dopočítat hodnotu sin 20°18’ z hodnot v bodech x_0=20°, x_1=21°,x_2=22°.

x	20°	21°	22°
sin (x)	0,34202	0,35837	0,37461

20°18’=20,3°

Lineární interpolace:

$$L_1(x) = f(x_0)\,(x-x_1)/(x_0-x_1) \; + \; f(x_1)\,(x-x_0)/(x_1-x_0)$$

sin 20°18’≈ 0,34202(20,3-21)/(20-21) + 0,35837 (20,3-20)/(21-20)=
=0,346925⍰0,34693

Kvadratická interpolace:

$$L_2(x) = f(x_0)((x-x_1)(x-x_2))/((x_0-x_1)\,(x_0-x_2) + f(x_1)((x-x_0)(x-x_2))/((x_1-x_0)(x_1-x_2))$$
$$+ f(x_2)\,((x-x_0)(x-x_1))/((x_2-x_0)(x_2-x_1))$$

sin 20°18’ ≈ 0,34202 ((20,3-21)(20,3-22))/((20-21)(20-22))
+0,35837((20,3-20) (20,3-22))/((21-20)(21-22))
+0,37461 ((20,3-20)(20,3-21))/((22-20)(22-21)) = 0,346936⍰0,34694

Přesná hodnota sin 20°18’= 0,34693655 ⍰ 0,34694. Vidíme, že kvadratická interpolace je přesnější, ale v tomto případě přesnost lineární interpolace byla dostačující.

Lagrangeova interpolace Pascal

```
program Lagrange;
uses crt;
 const n=10;
type pole=array[1..n] of real;
var xx,yy: pole;  x,y: real;   i,k,l,mm:byte;
 procedure nacti(m:byte;var aa,bb:pole);
  var d:byte; g,gg:real;
  begin
      for d:=1 to m do begin
          writeln('   Zadej souradnice  ',d,'. bodu:');
          write('   x=');     readln(g);      aa[d]:=g;
          write('   y=');     readln(gg);    bb[d]:=gg;  end;
  end;
procedure tisk(m:byte;aa,bb:pole);
 var t:byte;
 begin  writeln;  for t:=1 to m do write('
f(',aa[t]:5:3,')=',bb[t]:5:3,'    ');
 writeln; end;
```

```
procedure lagran(m:byte; a:real; aa,bb:pole; var ss:real);
 var q,qq:byte; s,j,z:real;
   begin       ss:=0;
    for q:=1 to m do begin     s:=1; j:=1;   z:=0;
                  if q=1 then for qq:=2 to m do begin
                                      s:=s*(a-aa[qq]); j:=j*(aa[q]-aa[qq])
                                              end
                        else if q<m then begin
                                        for qq:=1 to q-1 do begin
                                       s:=s*(a-aa[qq]); j:=j*(aa[q]-aa[qq])
                                                          end;
                                        for qq:=q+1 to m do begin
                                       s:=s*(a-aa[qq]); j:=j*(aa[q]-aa[qq])
                                                          end
                                        end
                                   else for qq:=1 to m-1 do begin
                                       s:=s*(a-aa[qq]); j:=j*(aa[q]-aa[qq])
                                                          end;
     z:=((s/j)*bb[q]);   write('   f(x',q,')*l',q,'=',z:5:3,'   '); ss:=ss+z;
                  end;  writeln;
   end;
```

```
begin
clrscr;
  writeln('   Zadej pocet bodu:      Maximum = ',n);  read(mm);
  writeln('   Zadej x-souradnici hledaneho bodu: ');  readln(x);
  writeln;
  nacti(mm,xx,yy);   writeln;

  lagran(mm,x,xx,yy,y);

  writeln;   writeln('   Zadane body : ');
  tisk(mm,xx,yy);

  writeln;
writeln( '   Hodnota Lagrangeova polynomu L',mm,'(',x:3:2,')  =',y:10:3);

  repeat until keypressed;
end.
```

Lagrangeuv interpolační polynom C++

```
// Lagrangeuv interpolacni polynom
#include <iostream>
#include <stdlib.h>
//#include <conio.h>
const int N=50;
using namespace std;

int  main()
{               double x[N],y[N];    // tabulka funkce
                double x0,fx0,cit,jm;   // x0 -> bod, v nemz urcujeme funkcni hodnotu
                int i,n,p;
                cout<<endl<<"      Lagrangeuv interpolacni polynom."<<endl;
                cout <<endl<< "   Zadejte pocet vstupnich bodu (max. 50):"<<endl;
                cout<<"   Pocet bodu:  "; cin >> n; cout<<endl;

                cout << "   Zadejte tabulku:"<<endl;
                for(i=0;i<n;i++)
      {cout<<"   x["<<i<<"]= "; cin >>x[i];
       cout<<"   y["<<i<<"]= "; cin >>y[i];}
```

```
cout << endl<<"   Zadejte hodnotu x0, pro niz chceme urcit y=f(x0)"<<endl;
                cout<<"   x0 = ";    cin >> x0;   cout<<endl;
               for (i=0;i<n;i++)   {cout<<"  x["<<i<<"]="<<x[i]<<"   y["<<i<<"]="<<y[i]<<endl;}
                fx0=0;  for( int i=0; i<n;  i++)      { cit=1; jm= 1; if (i==0) {for ( p=1; p<n;p++)
                         {cit *=(x0-x[p]); jm *=(x[i] - x[p]);     }
                                                      }
                    else  if (i<(n-1))
                  { for ( p=0; p<(i);p++)    {cit *=(x0-x[p]); jm *=(x[i] - x[p]);    }
                         for(p=(i+1); p<n;p++)      {cit *=(x0-x[p]); jm *=(x[i] - x[p]);   }
                  }
                   else     { for(p=0; p<(n-1);p++)
                          {cit *=(x0-x[p]); jm *=(x[i] - x[p]);      }
                           }
                           fx0 +=((cit/jm)*y[i]);
    }
   cout<<endl<<"  f("<<x0<<") = "<<fx0<<endl;
               return 0;
}
```

Použité zdroje

PŘIKRYL, Petr. *Numerické metody analýzy*. Praha: Státní nakladatelství technické literatury, 1988. MATEMATIKA PRO VYSOKÉ ŠKOLY TECHNICKÉ, sešit XXIV.