

Interpolace a extrapolace II/III
Kubický spline

Jan Schee

ÚF FPF SU Opava 2012

Máme tabulovanou funkci $y_i = y(x_i)$

Na intervalu $[x_i, x_{i+1}]$ bude lineární interpolace daná vztahem

$$y = A y_i + B y_{i+1} \quad (1)$$

kde je

$$A \equiv \frac{x_{i+1} - x}{x_{i+1} - x_i}, \quad B \equiv 1 - A = \frac{x - x_i}{x_{i+1} - x_i}$$

Linearita implikuje, že vztah (1) má:

- nulové druhé derivace na každém intervalu,
- nedefinované nebo divergující první derivace v abcisách x_i .

Cílem konstrukce kubického splinu je získání interpolační formule, která má:

- hladké první derivace,
- spojité druhé derivace,

uvniř intervalu i v jeho hraničních bodech.

Předpokládáme, že máme, kromě funkčních hodnot, tabulované druhé derivace $y''_i = y''(x_i)$. Pak, na každém intervalu, můžeme přičíst k pravé straně rovnice (1) kubický polynom jehož druhá derivace se bude měnit lineárně od hodnoty y''_i (vlevo) k hodnotě y''_{i+1} (vpravo). Takto získáme požadovanou spojitou druhou derivaci splinu.

Za výše uvedených podmínek, přejde rovnice (1) na tvar

$$y = A y_i + B y_{i+1} + C y''_i + D y''_{i+1} \quad (2)$$

kde jsou A a B stejné jako v (1) a

$$C \equiv \frac{1}{6} (A^3 - A) (x_{i+1} - x_i)^2, \quad D \equiv \frac{1}{6} (B^3 - B) (x_{i+1} - x_i)^2 \quad (3)$$

První derivace tohoto polynomu je

$$\frac{dy}{dx} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{3A^2 - 1}{6} (x_{i+1} - x_i) y''_i + \frac{3B^2 - 1}{6} (x_{i+1} - x_i) y''_{i+1} \quad (4)$$

Druhá derivace tohoto polynomu je

$$\frac{d^2 y}{dx^2} = A y''_i + B y''_{i+1} \quad (5)$$

Zbývá se vypořádat se skutečností, že y''_i vlastně neznáme.

Využijeme požadavek, že první derivace (rce (3)) mají být spojité na hranici dvou intervalů.

Hledanou rovnici sestavíme tak, že hodnotu funkce (3) v $x=x_i$ na intervalu $[x_{i-1}, x_i]$ položíme rovnu hodnotě funkce (3) v bodě $x=x_i$ ale na intervalu $[x_i, x_{i+1}]$. Dostáváme tak $N-2$ rovnic

$$\frac{x_i - x_{i-1}}{6} y''_{i-1} + \frac{x_{i+1} - x_{i-1}}{3} y''_i + \frac{x_{i+1} - x_i}{6} y''_{i+1} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$$

pro N neznámých y''_i .

Abychom našli jednoznačné řešení, musíme určit další dvě podmínky. Nejčastěji se používají následující:

- nastavit jednu nebo obě hodnoty y''_1 a y''_N rovny nule – *přirozený kubický spline*
- nastavit jednu nebo obě hodnoty y''_1 a y''_N rovny hodnotě určené z rovnice (4) tak, že první derivace interpolující funkce bude mít v krajních bodech předem stanovenou hodnotu.

```

void spline(double x[], double y[], int n, double yp1,
            double ypn, double y2[])
{
    int i,k;
    double p,qn,sig,un,*u;

    u=dvector(1,n-1); //alokace pomocneho pole
    if (yp1 > 0.99e30) //okrajove podminky splinu
        y2[1]=u[1]=0.0;
    else {
        y2[1] = -0.5;
        u[1]=(3.0/(x[2]-x[1]))*((y[2]-y[1])/(x[2]-x[1])-yp1);
    }
    for (i=2;i<=n-1;i++) { //vyp. pomoc. tabulky uvnitr intervalu
        sig=(x[i]-x[i-1])/(x[i+1]-x[i-1]);
        p=sig*y2[i-1]+2.0;
        y2[i]=(sig-1.0)/p;
        u[i]=(y[i+1]-y[i])/(x[i+1]-x[i]) - (y[i]-y[i-1])/(x[i]-
x[i-1]));
        u[i]=(6.0*u[i]/(x[i+1]-x[i-1])-sig*u[i-1])/p;
    }
    /* funkce pokračuje na dalsim slidu */
}

```

```
if (ypn > 0.99e30) //okrajove podminky splinu
    qn=un=0.0;
else {
    qn=0.5;
    un=(3.0/(x[n]-x[n-1]))*(ypn-(y[n]-y[n-1])/(x[n]-x[n-1]));
}
y2[n]=(un-qn*u[n-1])/(qn*y2[n-1]+1.0);
for (k=n-1;k>=1;k--) //vypocet tabulky druhych derivaci
    y2[k]=y2[k]*y2[k+1]+u[k];
free_dvector(u,1,n-1);
} //konec funkce spline()
```

```

void splint(double xa[], double ya[], double y2a[], int n,
double x, double *y)
{
    int klo,khi,k;
    double h,b,a;

    klo=1;
    khi=n;
    while (khi-klo > 1) { //hledani intervalu do ktereho padne x
        k=(khi+klo) >> 1; //puleni intervalu
        if (xa[k] > x) khi=k;
        else klo=k;
    }
    h=xa[khi]-xa[klo];
    if (h == 0.0) {
        fprintf(stderr, "Spatny vstup xa ve funkci splint.\n");
        return ;
    }
    //vypocet interpolacniho polynomu podle vztahu (2)
    a=(xa[khi]-x)/h;
    b=(x-xa[klo])/h;
    *y=a*ya[klo]+b*ya[khi]+((a*a*a-a)*y2a[klo]+(b*b*b-
b)*y2a[khi])*(h*h)/6.0;
}

```