FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Interpolace funkcí

SEM2 – LS 2023/2024

Pavel Exner, Petr Rálek
NTI, FM TUL

pavel.exner@tul.cz, petr.ralek@tul.cz

[edit: 3. dubna 2024]

Základní pojmy

Cílem bude najít funkci, která bude určitým způsobem popisovat sadu dat.

Rozlišujeme:

- **Interpolace** je přibližné určení hodnoty funkce v bodě, který leží uvnitř intervalu (nebo obecněji množiny), na němž jsou zadány hodnoty.
- **Extrapolace** znamená určení hodnoty funkce vně intervalu (množiny), na kterém jsou zadány hodnoty.
- **Aproximací** se rozumí nalezení funkce, která je v nějakém smyslu blízká zadaným bodům, ale nemusí těmito body procházet.

Interpolace

Nechť jsou dány body $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, například:

	x_0	x_1	x_2	x_3
x	1.0	1.5	2.5	3.5
$f(x)$	0.5	2.25	0.75	1.5
	y_0	y_1	y_2	y_3

Body x_i, $i = 0, \ldots, n$ nemusí být rozmístěny rovnoměrně, budeme však předpokládat, že jsou uspořádány vzestupně, tj. $x_i < x_{i+1}$. Naším cílem je nalézt vhodnou funkci f tak, aby splňovala

$$f(x_0) = y_0, \quad f(x_1) = y_1, \quad \ldots \quad f(x_n) = y_n.$$

Podle toho, jaký zvolíme druh funkce, mluvíme o různých typech interpolace.

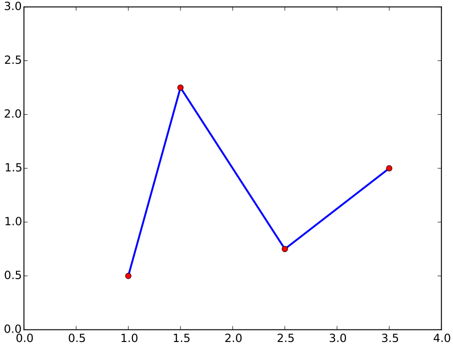
Po částech lineární interpolace I

Funkci f můžeme zvolit tak, aby byla lineární na každém úseku $[x_i, x_{i+1}]$, $i = 0, \ldots, n-1$. Pro $x \in [x_i, x_{i+1}]$ tedy definujeme

$$f(x) = \frac{x - x_i}{x_{i+1} - x_i} y_{i+1} + \frac{x_{i+1} - x}{x_{i+1} - x_i} y_i,$$

Tato funkce je velice jednoduchá, má však nevýhodu, že v bodech x_i není hladká (nelze v nich sestrojit tečnu).

Po částech lineární interpolace II



	x_0	x_1	x_2	x_3
x	1.0	1.5	2.5	3.5
$f(x)$	0.5	2.25	0.75	1.5
	y_0	y_1	y_2	y_3

Příklad po částech lineární interpolace.

Vandermondeova matice

Mějme polynom $p_n(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \cdots + a_1 x + a_0$, kterým chceme interpolovat body $[x_i, y_i]$.
Naivním přístupem je dosazení do interpolačních podmínek

$$p_n(x_i) = y_i \qquad \forall i = 0, \ldots, n$$

Dostaneme matici

$$\begin{pmatrix} x_0^n & x_0^{n-1} & \cdots & x_0 & 1 \\ x_1^n & x_1^{n-1} & \cdots & x_1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_n^n & x_n^{n-1} & \cdots & x_n & 1 \end{pmatrix} \begin{pmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Matice soustavy, nazývaná Vandermondeova, může být velmi špatně podmíněná. To klade velké nároky na numerické metody. Řešení navíc vykazuje výrazné zákmity pro větší počet uzlů (vyšší řád polynomu).

Lagrangeova interpolace I

Pro určení interpolačního polynomu $p_n(x)$ existuje mnoho technik. Lagrange odvodil postup pro výpočet tohoto polynomu:

$$P_n(x) = y_0 L_0(x) + y_1 L_1(x) + \ldots + y_n L_n(x) = \sum_{i=0}^{n} y_i L_i(x),$$

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

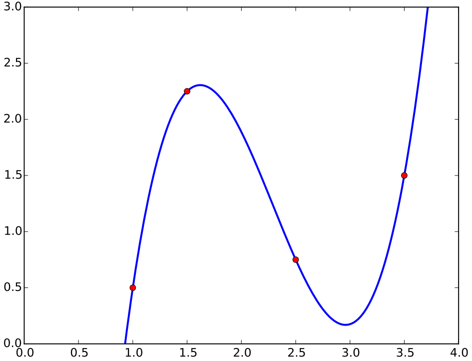
Lagrangeova interpolace II

Příklad výpočtu Lagrangeova interpolačního polynomu:

	x_0	x_1	x_2	x_3
x	1.0	1.5	2.5	3.5
$f(x)$	0.5	2.25	0.75	1.5
	y_0	y_1	y_2	y_3

$$
\begin{aligned}
P_3(x) =& 0.5\frac{(x-1.5)(x-2.5)(x-3.5)}{(1-1.5)(1-2.5)(1-3.5)} \\
&+ 2.25\frac{(x-1)(x-2.5)(x-3.5)}{(1.5-1)(1.5-2.5)(1.5-3.5)} \\
&+ 0.75\frac{(x-1)(x-1.5)(x-3.5)}{(2.5-1)(2.5-1.5)(2.5-3.5)} \\
&+ 1.5\frac{(x-1)(x-1.5)(x-2.5)}{(3.5-1)(3.5-1.5)(3.5-2.5)} = \frac{107}{60}x^3 - \frac{49}{4}x^2 + \frac{6157}{240}x - \frac{235}{16}.
\end{aligned}
$$

Lagrangeova interpolace III



	x_0	x_1	x_2	x_3
x	1.0	1.5	2.5	3.5
$f(x)$	0.5	2.25	0.75	1.5
	y_0	y_1	y_2	y_3

Lagrangeův interpolační polynom pro zadaná data.

Hermiteova polynomiální interpolace I

Pro případy, kdy jsou k dispozici nejen hodnoty, ale i derivace ve stejných bodech, může být vhodná Hermiteova interpolace. Předpokládejme tedy, že pro $n+1$ bodů jsou zadána data (x_i, y_i, y_i'), $i = 0, \ldots, n$, např.:

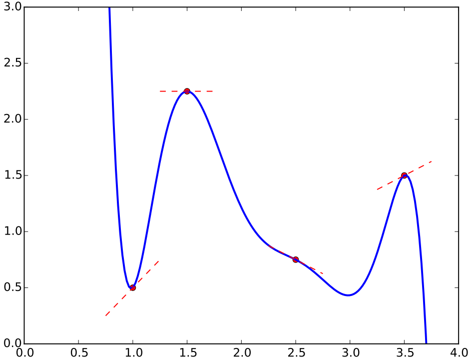
i	0	1	2	3
x_i	1.0	1.5	2.5	3.5
y_i	0.5	2.25	0.75	1.5
y_i'	1	0	-0.5	0.5

Hermiteův interpolační polynom je polynom P_{2n+1} stupně $2n+1$, který splňuje

$$P_{2n+1}(x_i) = y_i, \quad P_{2n+1}'(x_i) = y_i', \quad i = 0, \ldots, n.$$

Těmito $2n+2$ podmínkami je P_{2n+1} určen jednoznačně. Existuje obecný postup, jak tento polynom vypočítat, nebudeme jej zde však uvádět.

Hermiteova polynomiální interpolace II



i	0	1	2	3
x_i	1.0	1.5	2.5	3.5
y_i	0.5	2.25	0.75	1.5
y_i'	1	0	-0.5	0.5

Hermiteův interpolační polynom pro zadaná data.

Interpolace kubickým splinem I

Nevýhody Lagrangeovy a Hermiteovy interpolace:

- změna 1 hodnoty znamená přepočítání celého interpolačního polynomu.
- pro velká n roste stupeň polynomu $\rightarrow$ nárůst zaokrouhlovacích chyb a **oscilace**

Tyto nevýhody odstraňují tzv. **kubické spliny**.

Kubický spline je funkce $f(x)$ s těmito vlastnostmi:

- prochází zadanými body (x_i, y_i), $i = 0, \ldots, n$
- na každém intervalu (x_i, x_{i+1}) je y polynom 3. stupně (kubická funkce)
- v každém vnitřním bodě x_i mají obě kubické funkce stejnou derivaci (tečnu) a druhou derivaci (křivost)

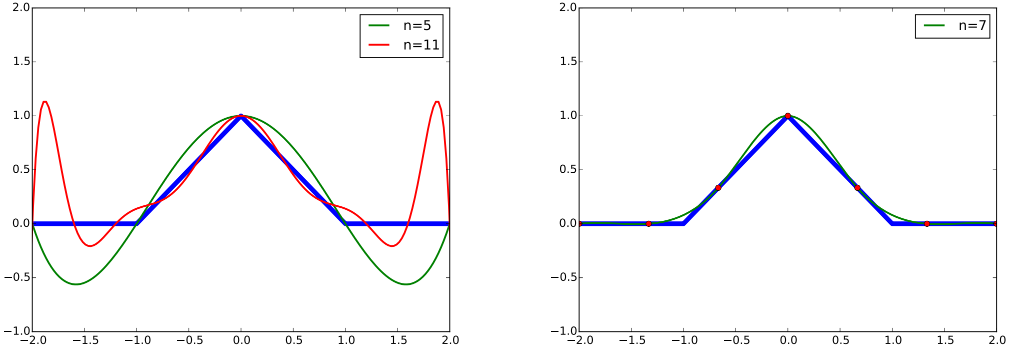
Interpolace kubickým splinem II

Tyto vlastnosti představují $4n-2$ podmínek pro $4n$ koeficientů. Pro jednoznačné určení kubického splinu je třeba přidat 2 dodatečné podmínky v krajních bodech. Typicky se volí např. následující podmínky:

a) nulová křivost
b) konstantní křivost na $[x_0, x_1]$ a $[x_{n-1}, x_n]$
c) lineární extrapolace křivosti

Výpočet koeficientů kubického splinu pak lze realizovat poměrně efektivně. Interpolace splinem je vhodná pro širokou škálu úloh, včetně nespojitých nebo nehladkých dat.

Interpolace kubickým splinem III



Interpolace nehladkých dat. Vlevo: Lagrangeova interpolace, vpravo: kubický spline.

Matlab a interpolace

`A = vander(x)` – Vandermondeova matice pro body ve vektoru `x`

Interpolační funkce mají všechny obdobné rozhraní:

```
yq = interpolacni_funkce(x,y,xq)
```

Funkce nalezne interpolační hodnoty `yq` v zadaných bodech `xq` pro data `[x,y]`.
Následující funkce počítají kubický hermitovský spline:

- `spline` – prokládá kubický spline (drží spojitost 2. derivací v x_i)
- `pchip` – nemá překmity, omezuje oscilace
- `makima` – může mít překmity, omezuje oscilace

Všechny se také volat pomocí funkce `interp1(x,y,xq,method)`, kde `method` specifikuje metodu.

`polyfit(x,y,n)` – aproximace pomocí nejmenších čtverců (viz příští prezentace), pokud `n` odpovídá délce `x` a `y` $\Rightarrow$ interpolační polynom řádu `n` (vyzkoušejte!)