

Numerické metody

Doc. RNDr. Libor Čermák, CSc. RNDr. Rudolf Hlavička, CSc.

Ústav matematiky
Fakulta strojního inženýrství
Vysoké učení technické v Brně

1. listopadu 2006

Obsah

- 3 Aproximace funkcí
 - Aproximace a interpolace
 - Interpolace
 - Interpolace polynomem
 - Interpoláčn  splajny
 - Interpolace funkc  v ce prom nn ch
 - Metoda nejmen  ch  tverc 
 - Literatura

Aproximovat funkci $f(x)$ znamená nahradit ji funkcí $\varphi(x)$, která je k $f(x)$ v jistém smyslu blízka. Píšeme $\varphi(x) \approx f(x)$. Budeme se zabývat dvěma základními typy aproximace, a to interpolací a metodou nejmenších čtverců.

Interpolace je taková aproximace, při níž $\varphi(x)$ nabývá v zadaných bodech x_i předepsaných hodnot $y_i = f(x_i)$. Někdy navíc žádáme, aby funkce φ a f měly v bodech x_i také stejné derivace. Interpolaci je věnován odstavec 5.

Metoda nejmenších čtverců je taková aproximace, při níž $\varphi(x)$ „prokládáme“ mezi zadanými body $[x_i, y_i]$ tak, aby „vzdálenost“ funkcí f a φ byla v jistém smyslu minimální. Je přitom charakteristické, že funkce φ body $[x_i, y_i]$ neprochází. Metoda nejmenších čtverců je vyložena v odstavci 50.

Aproximaci $\varphi(x)$ použijeme k přibližnému výpočtu hodnot funkce $f(x)$, třeba při vykreslování $\varphi \approx f$. Je žádoucí, aby výpočet $\varphi(x)$ byl „jednoduchý“. Proto se φ často hledá ve tvaru polynomu.

Obecně, $\varphi(x)$ se používá k řešení úloh, v nichž vystupuje funkce f , kterou je účelné nebo dokonce nezbytné nahradit její vhodnou aproximací φ . Jako příklad uveďme výpočet derivace nebo určitého integrálu: $f'(x)$ nahradíme pomocí $\varphi'(x)$ a $\int_a^b f(x) dx$ nahradíme pomocí $\int_a^b \varphi(x) dx$.

Obsah

- 3 Aproximace funkcí
 - Aproximace a interpolace
 - Interpolace
 - Interpolace polynomem
 - Interpolační splajny
 - Interpolace funkcí více proměnných
 - Metoda nejmenších čtverců
 - Literatura

Interpolační funkci $\varphi(x)$ vybíráme z vhodné třídy funkcí. Omezíme se na dva nejběžnější případy:

- a) $\varphi(x)$ je polynom;
- b) $\varphi(x)$ je po částech polynom, na každém subintervalu obecně jiný.

Interpolace polynomem

Předpokládejme, že jsou dány navzájem různé body

$$x_0, x_1, \dots, x_n, \quad x_i \neq x_j \quad \text{pro} \quad i \neq j,$$

říkáme jim také **uzly interpolace**, a v každém z nich je předepsána hodnota y_i .

Hledáme **interpoláčn**í **polynom** $P_n(x)$ stupně nejvýše n , který splňuje **interpoláčn**í **podm**ínky

$$P_n(x_i) = y_i, \quad i = 0, 1, \dots, n. \quad (3.1)$$

Existenci interpolačního polynomu dokážeme tak, že ho zkonstruujeme.

Lagrangeův tvar interpolačního polynomu má vyjádření

$$P_n(x) = y_0 \ell_0(x) + y_1 \ell_1(x) + \cdots + y_n \ell_n(x) = \sum_{i=0}^n y_i \ell_i(x), \quad (3.2)$$

kde $\ell_i(x)$ jsou tzv. **fundamentální polynomy** definované předpisem

$$\ell_i(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}. \quad (3.3)$$

Snadno nahlédneme, že

$$\ell_i(x_k) = \begin{cases} 1 & \text{pro } k = i, \\ 0 & \text{pro } k \neq i, \end{cases} \quad i, k = 0, 1, \dots, n, \quad (3.4)$$

takže interpolační podmínky $P_n(x_k) = \sum_{i=0}^n y_i \ell_i(x_k) = y_k$, $k = 0, 1, \dots, n$, jsou splněny.

Interpolační polynom je daty $[x_i, y_i]$, $i = 0, 1, \dots, n$, určen jednoznačně. Skutečně, jsou-li P a Q interpolační polynomy splňující interpolační podmínky $P(x_i) = Q(x_i) = y_i$, pak polynom $P - Q$ je roven nule v uzlech x_0, x_1, \dots, x_n . Avšak polynom stupně nejvýše n nemůže mít více než n kořenů, pokud se nerovná identicky nule. V našem případě proto nutně $P - Q = 0$ a tedy $P = Q$. Tím je jednoznačnost interpolačního polynomu dokázána.

Příklad 3.1. Určíme interpolační polynom pro data předepsaná tabulkou

x_i	-1	1	2	3
y_i	-6	-2	-3	2

Nejdříve získáme fundamentální polynomy

$$\ell_0(x) = \frac{(x-1)(x-2)(x-3)}{(-1-1)(-1-2)(-1-3)} = -\frac{1}{24}(x^3 - 6x^2 + 11x - 6),$$

$$\ell_1(x) = \frac{(x+1)(x-2)(x-3)}{(1+1)(1-2)(1-3)} = \frac{1}{4}(x^3 - 4x^2 + x + 6),$$

$$\ell_2(x) = \frac{(x+1)(x-1)(x-3)}{(2+1)(2-1)(2-3)} = -\frac{1}{3}(x^3 - 3x^2 - x + 3),$$

$$\ell_3(x) = \frac{(x+1)(x-1)(x-2)}{(3+1)(3-1)(3-2)} = \frac{1}{8}(x^3 - 2x^2 - x + 2)$$

a pak sestavíme interpolační polynom

$$P_3(x) = -6 \cdot \ell_0(x) - 2 \cdot \ell_1(x) - 3 \cdot \ell_2(x) + 2 \cdot \ell_3(x) = x^3 - 3x^2 + x - 1. \quad \square$$

Hlavní předností Lagrangeova tvaru interpolačního polynomu je jeho elegantní forma. Používá se proto zejména v teoretických úvahách. Pro praktické použití však ideální není. Upozorníme na dva jeho hlavní nedostatky.

- (a) Přidáme-li další uzel x_{n+1} , musíme přepočítat všechny fundamentální polynomy.
- (b) Počet operací potřebných k výpočtu hodnoty $P_n(\bar{x})$ je poměrně značný, vyžaduje $2n^2 + 2n$ násobících operací a $2n^2 + 3n$ operací sčítacích.

Oba tyto nedostatky odstraňuje

Newtonův tvar interpolačního polynomu, ve kterém

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}). \quad (3.5)$$

Přidání dalšího uzlu x_{n+1} je snadné, k $P_n(x)$ stačí přičíst jeden další člen, neboť

$$P_{n+1}(x) = P_n(x) + a_{n+1}(x - x_0)(x - x_1) \cdots (x - x_n).$$

Nedostatek (a) Lagrangeova tvaru interpolačního polynomu jsme tedy překonali.

Hodnotu $z = P_n(\bar{x})$ určíme podobně jako v **Hornerově schématu**, tj. postupem:

$$z := a_n \text{ a pak pro } i = n-1, n-2, \dots, 0 \text{ počítej } z := z(\bar{x} - x_i) + a_i. \quad (3.6)$$

Koeficienty a_i lze vypočítat přímo z interpolačních podmínek (3.1) a dosáhnout tak významné úspory v počtu operací. Existuje však ještě lepší způsob a ten si teď uvedeme.

Nejdříve definujeme **poměrné difference**:

$$P[x_i] := y_i,$$

$$P[x_i, x_{i+1}] := (P[x_{i+1}] - P[x_i]) / (x_{i+1} - x_i),$$

$$P[x_i, x_{i+1}, x_{i+2}] := (P[x_{i+1}, x_{i+2}] - P[x_i, x_{i+1}]) / (x_{i+2} - x_i),$$

a dále, pro $3 \leq k \leq n$:

$$P[x_i, x_{i+1}, \dots, x_{i+k-1}, x_{i+k}] := (P[x_{i+1}, \dots, x_{i+k}] - P[x_i, \dots, x_{i+k-1}]) / (x_{i+k} - x_i). \quad (3.7)$$

Dá se ukázat, že

$$a_i = P[x_0, x_1, \dots, x_i], \quad (3.8)$$

takže **Newtonův tvar interpolačního polynomu** je

$$\begin{aligned} P_n(x) = & P[x_0] + P[x_0, x_1](x - x_0) + P[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots \\ & + P[x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \dots (x - x_{n-1}). \end{aligned} \quad (3.9)$$

Označíme-li $P_{ik} = P[x_{i-k}, \dots, x_i]$, pak $a_i = P_{ii}$. Následuje

algoritmus výpočtu poměrných diferencí

Pro $i = 0, 1, \dots, n$ proved' $P_{i0} := y_i$.

Pro $k = 1, 2, \dots, n$ prováděj :

pro $i = k, k + 1, \dots, n$ prováděj :

$$P_{ik} := (P_{i,k-1} - P_{i-1,k-1}) / (x_i - x_{i-k})$$

konec cyklu i ,

konec cyklu k .

x_0	P_{00}					
x_1	P_{10}	P_{11}				
x_2	P_{20}	P_{21}	P_{22}			
x_3	P_{30}	P_{31}	P_{32}	P_{33}		
\vdots	\vdots	\vdots	\vdots		\ddots	
x_n	P_{n0}	P_{n1}	P_{n2}	\dots	$P_{n,n-1}$	P_{nn}

Výpočet lze přehledně zaznamenat do tabulky, kterou vyplňujeme po sloupcích. Výpočet koeficientů $a_i = P_{ii}$ a následný výpočet $P_n(\bar{x})$ podle (3.6) vyžaduje $\frac{1}{2}n^2 + \frac{3}{2}n$ násobících operací a $n^2 + 3n$ operací sčítacích. Dosáhli jsme tedy významně nižšího počtu operací, než kolik je jich třeba pro výpočet $P_n(\bar{x})$ z Lagrangeova interpolačního polynomu. To znamená, že také nedostatek (b) Lagrangeova interpolačního polynomu je uspokojivě vyřešen.

Příklad 3.2. Sestavíme Newtonův interpolační polynom pro data z příkladu 3.1. Průběh výpočtu zaznamenáváme do tabulky. Dostaneme

x_i	P_{i0}	P_{i1}	P_{i2}	P_{i3}	
-1	-6				$\Rightarrow a_0 = -6$
1	-2	2			$\Rightarrow a_1 = 2$
2	-3	-1	-1		$\Rightarrow a_2 = -1$
3	2	5	3	1	$\Rightarrow a_3 = 1,$

takže $P_3(x) = -6 + 2 \cdot (x + 1) + (-1) \cdot (x + 1)(x - 1) + 1 \cdot (x + 1)(x - 1)(x - 2)$.
V bodě $\bar{x} = 0,5$ podle (3.6) vypočteme

$$P_3(0,5) = ((1 \cdot (0,5 - 2) - 1) \cdot (0,5 - 1) + 2) \cdot (0,5 + 1) - 6 = -1,125.$$

Když přidáme další uzel $x_4 = 0$ a v něm předepíšeme hodnotu $y_4 = 2$, stačí dopočítat jeden řádek tabulky. Dostaneme

x_4	P_{40}	P_{41}	P_{42}	P_{43}	P_{44}	
0	2	0	2,5	0,5	-0,5	$\Rightarrow a_4 = -0,5,$

a tedy $P_4(x) = P_3(x) + (-0,5) \cdot (x+1)(x-1)(x-2)(x-3)$. \square

Chyba aproximace interpolačním polynomem. Nejdříve zavedeme následující

Označení. Symbolem $C\langle a, b \rangle$ budeme značit množinu všech funkcí, které jsou v intervalu $\langle a, b \rangle$ spojité, a symbolem $C^k\langle a, b \rangle$ pak množinu všech funkcí, které jsou v intervalu $\langle a, b \rangle$ spojité spolu se svými derivacemi až do řádu k včetně. Pro $k = 0$ zřejmě $C^0\langle a, b \rangle \equiv C\langle a, b \rangle$.

Předpokládejme, že čísla y_i nejsou libovolná, ale že $y_i = f(x_i)$ jsou hodnoty funkce f v uzlech interpolace. Pak nás jistě bude zajímat chyba

$$E_n(\bar{x}) := f(\bar{x}) - P_n(\bar{x})$$

ve zvoleném bodě \bar{x} . Pro $\bar{x} = x_i$ je $E_n(x_i) = 0$. Jaká je ale chyba mimo uzly interpolace?

Nechť tedy \bar{x} je libovolný bod, $\langle a, b \rangle$ je nějaký interval obsahující všechny uzly x_i interpolace a také zkoumaný bod \bar{x} , a necht' $f \in C^{n+1}\langle a, b \rangle$. Pak pro chybu $E_n(\bar{x})$ platí

$$E_n(\bar{x}) = f(\bar{x}) - P_n(\bar{x}) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(\bar{x} - x_0)(\bar{x} - x_1) \dots (\bar{x} - x_n), \quad (3.10)$$

kde $\xi = \xi(\bar{x})$ je (blíže určený) bod z intervalu $\langle a, b \rangle$. Zápisem $\xi = \xi(\bar{x})$ přitom chceme zdůraznit, že poloha bodu ξ závisí nejen na funkci f a na interpolantu P_n , ale také na zvoleném bodu \bar{x} .

Poznámky. Abychom zjednodušili výklad, budeme předpokládat, že

$$x_0 < x_1 < \dots < x_n.$$

- 1) Jestliže M_{n+1} je taková konstanta, že $|f^{(n+1)}(x)| \leq M_{n+1}$ pro každé $x \in \langle a, b \rangle$, pak

$$|E_n(\bar{x})| \leq \frac{M_{n+1}}{(n+1)!} \max_{x \in \langle a, b \rangle} |\omega_{n+1}(x)|, \quad (3.11)$$

kde $\omega_{n+1}(x) = (x - x_0)(x - x_1) \dots (x - x_n)$. Odhad (3.11) je však obvykle příliš pesimistický.

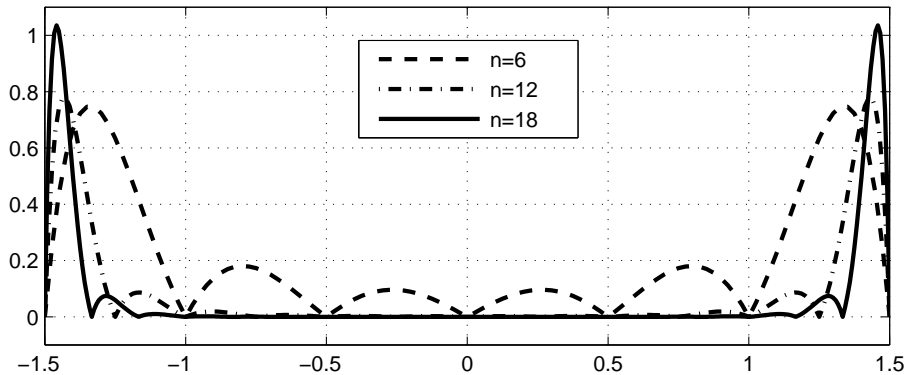
- 2) Jestliže má funkce $f(x)$ derivace všech řádů ohraničené stejnou konstantou, pak pro dostatečně velké n je chyba libovolně malá.

Příklad 3.3. Pro $f(x) = \sin x$ lze vzít $M_{n+1} = 1$, proto

$$|E_n(x)| \leq \frac{(b-a)^{n+1}}{(n+1)!}. \quad \text{Dá se dokázat, že } \frac{(b-a)^{n+1}}{(n+1)!} \rightarrow 0 \quad \text{pro } n \rightarrow \infty,$$

takže $P_n(x) \rightarrow f(x)$ pro každé x z libovolného konečného intervalu $\langle a, b \rangle$.

□



Obr. 3.1: Graf funkce $|\omega_{n+1}(x)|$

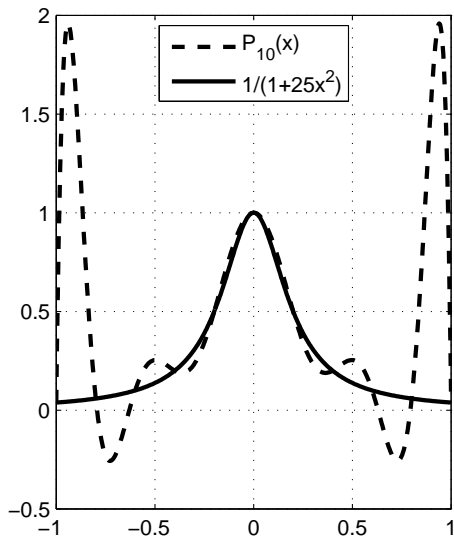
- 3) Jestliže interpolační polynom používáme k výpočtu hodnot interpolované funkce vně intervalu $\langle x_0, x_n \rangle$, říkáme, že provádíme **extrapolaci**. V tomto případě může být chyba aproximace velká, neboť hodnota $|\omega_{n+1}(x)|$ rychle roste, když se x vzdaluje od x_0 doleva nebo od x_n doprava.
- 4) $\omega_{n+1}(x)$ může nabývat velkých hodnot také uvnitř intervalu $\langle x_0, x_n \rangle$, zejména když jsou uzly x_i rozmístěny rovnoměrně, tj. když $x_i = x_0 + ih$, kde h je pevně zvolený krok.

Na obr. 3.1 vidíme, že ve středu intervalu $\langle x_0, x_n \rangle$ nabývá $|\omega_{n+1}(x)|$ nejmenších hodnot, v blízkosti středů krajních intervalů, zejména intervalů $\langle x_0, x_1 \rangle$ a $\langle x_{n-1}, x_n \rangle$, je však hodnota $|\omega_{n+1}(x)|$ značná. Toto chování polynomu $\omega_{n+1}(x)$ se promítne i do průběhu interpolantu $P_n(x)$.

Příklad 3.4. Sestrojíme interpolační polynom Rungeovy funkce

$$f(x) = \frac{1}{1 + 25x^2}$$

na rovnoměrném dělení intervalu $\langle -1, 1 \rangle$.



Obr. 3.2: Interpolační polynom Rungeovy funkce

Jde o známý příklad, na kterém se demonstruje, že pro rostoucí počet dílků chyba interpolace neomezeně roste. \square

Vhodným rozmístěním uzlů lze chybu $|\omega_{n+1}(x)|$ minimalizovat, viz cvičení. Tuto možnost však obvykle nemáme, neboť uzly jsou pevně dány. Proto používání interpolačních polynomů vysokých stupňů obecně nelze doporučit.

Hermitova interpolace. Doposud jsme se zabývali **Lagrangeovou interpolací**.

Její charakteristickým rysem je to, že interpolační polynom $P_n(x)$ je určen zadanými hodnotami $P_n(x_i) = y_i$ v uzlech x_i . Pokud interpolační polynom určují navíc také předepsané derivace, hovoříme o **Hermitově interpolaci**.

Předpokládejme tedy, že v každém uzlu x_i je zadáno $\alpha_i + 1$ čísel $y_i^{(0)}$,

$y_i^{(1)}, \dots, y_i^{(\alpha_i)}$. Označme $\alpha = n + \sum_{i=0}^n \alpha_i$. Pak **Hermitovým interpolačním polynomem** $P_\alpha(x)$ nazveme polynom stupně nejvýše α , který splňuje interpolační podmínky

$$\frac{d^j}{dx^j} P_\alpha(x_i) = y_i^{(j)}, \quad j = 0, 1, \dots, \alpha_i, \quad i = 0, 1, \dots, n. \quad (3.12)$$

Nultou derivací přitom rozumíme funkční hodnotu. Je dokázáno, že existuje jediný takový polynom. Jestliže

$$y_i^{(j)} = \frac{d^j}{dx^j} f(x_i), \quad j = 0, 1, \dots, \alpha_i, \quad i = 0, 1, \dots, n,$$

říkáme, že $P_\alpha(x)$ je Hermitův interpolační polynom funkce $f(x)$.

Nechť $\langle a, b \rangle$ je interval obsahující uzly interpolace. Jestliže $f \in C^{\alpha+1}\langle a, b \rangle$, pak pro chybu Hermitovy interpolace v bodě $\bar{x} \in \langle a, b \rangle$ platí

$$f(\bar{x}) - P_{\alpha}(\bar{x}) = \frac{f^{(\alpha+1)}(\xi)}{(\alpha+1)!} (\bar{x} - x_0)^{\alpha_0+1} (\bar{x} - x_1)^{\alpha_1+1} \dots (\bar{x} - x_n)^{\alpha_n+1}, \quad (3.13)$$

kde $\xi = \xi(\bar{x})$ je (nějaký blíže neurčený) bod z intervalu $\langle a, b \rangle$.

Použití Hermitova polynomu vysokého stupně obecně nelze doporučit, protože (stejně jako u Lagrangeovy interpolace) může být chyba interpolace mezi uzly značná.

Vzorec pro určení koeficientů Hermitova interpolačního polynomu je poměrně komplikovaný, lze ho najít např. v [Stoer, Bulirsch], zde ho neuvádíme. Místo toho si na příkladu ukážeme, jak lze Hermitův polynom určit přímo z interpolačních podmínek.

Příklad 3.5. Sestrojíme Hermitův interpolační polynom pro data podle tabulky

x_i	y_i	y_i'	y_i''
-1	2	-4	12
1	2	4	

Tedy $x_0 = -1$, $\alpha_0 = 2$, $y_0^{(0)} = 2$, $y_0^{(1)} = -4$, $y_0^{(2)} = 12$,

$x_1 = 1$, $\alpha_1 = 1$, $y_1^{(0)} = 2$, $y_1^{(1)} = 4$.

Protože je předepsáno celkem 5 podmínek, Hermitův polynom navrhne jako polynom stupně $\alpha = 4$. Abychom si ušetřili práci, zapíšeme ho ve tvaru mocninného rozvoje okolo toho bodu, v němž je předepsán největší počet podmínek, v našem případě tedy okolo $x_0 = -1$. Pak

$$P_4(x) = a + b(x + 1) + c(x + 1)^2 + d(x + 1)^3 + e(x + 1)^4.$$

Koeficienty a , b , c získáme snadno. Z podmínky $P_4(-1) = 2$ okamžitě dostaneme $a = 2$. Podobně z podmínky $P_4'(-1) = -4$ obdržíme $b = 4$ a protože $P_4''(-1) = 2c$, z podmínky $P_4''(-1) = 12$ dostaneme $c = 6$. Dále

$$P_4(1) = 2 - 4 \cdot 2 + 6 \cdot 2^2 + d \cdot 2^3 + e \cdot 2^4 = 2 \quad \implies \quad 8d + 16e = -16,$$

$$P_4'(1) = -4 + 2 \cdot 6 \cdot 2 + 3 \cdot d \cdot 2^2 + 4 \cdot e \cdot 2^3 = 4 \quad \implies \quad 12d + 32e = -16.$$

Tuto soustavu vyřešíme a dostaneme $d = -4$, $e = 1$. Tedy

$$P_4(x) = 2 - 4(x+1) + 6(x+1)^2 - 4(x+1)^3 + (x+1)^4 = x^4 - 1. \quad \square$$

Interpolační splajny

Jestliže chceme interpolovat funkci $f(x)$ na poměrně dlouhém intervalu $\langle a, b \rangle$, musíme žádat splnění interpolačních podmínek v dostatečně velkém počtu bodů. Pokud bude interpolantem polynom, musí být vysokého stupně a to, jak víme, obvykle vede k velkým chybám mezi uzly. Tudy proto cesta nevede. Lepší je rozdělit interval $\langle a, b \rangle$ na řadu menších subintervalů a na každém z nich sestavit interpolační polynom nižšího stupně.

Předpokládejme, že

$$a = x_0 < x_1 < \cdots < x_{i-1} < x_i < x_{i+1} < \cdots < x_{n-1} < x_n = b \quad (3.14)$$

je **dělení** intervalu $\langle a, b \rangle$. V každém uzlu x_i je předepsána hodnota y_i interpolantu. Délku i -tého intervalu $\langle x_{i-1}, x_i \rangle$ označíme h_i a délku nejdelšího intervalu h , tj.

$$h_i = x_i - x_{i-1}, \quad i = 1, 2, \dots, n, \quad h = \max_{1 \leq i \leq n} h_i. \quad (3.15)$$

Hledaný po částech polynomický interpolant budeme značit $S(x)$ a nazveme ho **interpolačním splajnem**. Na každém intervalu $\langle x_{i-1}, x_i \rangle$ je $S(x)$ polynom, jehož příslušnost k i -tému intervalu vyznačíme indexem i , tj.

$S(x)$ je na intervalu $\langle x_{i-1}, x_i \rangle$ polynom $S_i(x)$.

K vyjádření polynomu $S_i(x)$ s výhodou použijeme **lokální proměnnou**

$$s = x - x_{i-1}.$$

Budeme také používat první poměrnou diferenci

$$\delta_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}} = \frac{y_i - y_{i-1}}{h_i}.$$

Lineární interpolační splajn (dále jen lineární splajn) je to nejjednodušší, co nás napadne: každé dva sousední body $[x_{i-1}, y_{i-1}]$ a $[x_i, y_i]$ spojíme úsečkou. Zřejmě

$$S_i(x) = y_{i-1} + \frac{y_i - y_{i-1}}{x_i - x_{i-1}}(x - x_{i-1}) = y_{i-1} + s\delta_i \quad (3.16)$$

je lineární interpolační polynom procházející body $[x_{i-1}, y_{i-1}]$ a $[x_i, y_i]$. Lineární splajn $S(x)$ je spojitá funkce, derivace $S'(x)$ je však ve vnitřních uzlech obecně nespojitá.

Jestliže $y_i = f(x_i)$, $i = 0, 1, \dots, n$, a $f \in C^2\langle a, b \rangle$, pak pro chybu interpolace platí

$$|f(x) - S(x)| \leq Ch^2, \quad (3.17)$$

kde $x \in \langle a, b \rangle$ je libovolné a C je konstanta nezávislá na h .

Pro dostatečně mnoho uzlů lze učinit chybu libovolně malou. Například při vykreslování na obrazovku monitoru s rozlišením 1024×768 bodů jistě stačí použít 1024 interpolačních uzlů k získání kvalitního grafu interpolované funkce. Možná bychom byli spokojeni, i kdybychom zvolili méně uzlů, avšak při postupném snižování počtu uzlů by nutně nastal okamžik, kdy by nás již začaly rušit ostré hrany grafu $S(x)$ v interpolačních uzlech. Pokud bychom současně vykreslovali také funkci $f(x)$, pak by nám začaly vadit také viditelné odchylky interpolantu $S(x)$ od interpolované funkce $f(x)$ mezi uzly interpolace.

Přesnější interpolant bychom mohli sestavit tak, že bychom na intervalech $\langle x_0, x_k \rangle, \langle x_k, x_{2k} \rangle, \dots$ aproximovali $f(x)$ pomocí interpolačních polynomů stupně (nejvýše) $k > 1$. Chyba interpolace by v tom případě byla úměrná h^{k+1} , derivace v uzlech x_k, x_{2k}, \dots by však zůstaly nespojité. Velké k ale nemá smysl používat, jinak bychom zase mohli dostat velké chyby mezi uzly interpolace a byli bychom zpět v situaci, které jsme se právě interpolací po částech chtěli vyhnout.

Velmi populární je aproximace po částech kubickým polynomem, která je nejen spojitá, ale má také spojité první nebo dokonce i druhé derivace. Popisu takových aproximací se budeme věnovat v následujících odstavcích.

Hermitův kubický interpolační splajn (dále jen Hermitův kubický splajn) hledáme jako funkci $S(x)$, která

- a) je v intervalu $\langle a, b \rangle$ spojitá spolu se svou první derivací, tj. $S \in C^1\langle a, b \rangle$,
- b) splňuje interpolační podmínky

$$S(x_i) = y_i, \quad S'(x_i) = d_i, \quad i = 0, 1, \dots, n, \quad (3.18)$$

kde y_i , d_i jsou předepsané funkční hodnoty a derivace,

- c) je na každém intervalu $\langle x_{i-1}, x_i \rangle$, $i = 1, 2, \dots, n$, polynom (nejvýše) třetího stupně.

$S_i(x)$ je tedy kubický Hermitův polynom jednoznačně určený podmínkami

$$\begin{aligned} S_i(x_{i-1}) &= y_{i-1}, & S'_i(x_{i-1}) &= d_{i-1}, \\ S_i(x_i) &= y_i, & S'_i(x_i) &= d_i. \end{aligned}$$

Snadno ověříme, že tyto podmínky jsou splněny pro

$$S_i(x) = y_{i-1} + s d_{i-1} + s^2 \frac{3\delta_i - 2d_{i-1} - d_i}{h_i} + s^3 \frac{d_{i-1} - 2\delta_i + d_i}{h_i^2}. \quad (3.19)$$

Funkce $S(x)$ je spojitá spolu se svou první derivací, druhá derivace už obecně spojitá není.

Jestliže $y_i = f(x_i)$, $d_i = f'(x_i)$, $i = 0, 1, \dots, n$, a $f \in C^4\langle a, b \rangle$, pak pro chybu interpolace platí

$$|f(x) - S(x)| \leq Ch^4, \quad (3.20)$$

kde $x \in \langle a, b \rangle$ je libovolné a C je konstanta nezávislá na h .

Pokud derivace d_i nejsou k dispozici, musíme je vypočítat pomocí vhodně zvolených dodatečných podmínek.

Hermitův kubický interpolační splajn zachovávající tvar je jednou z možností. Derivace d_i se vybírají tak, aby průběh $S(x)$ kopíroval průběh lineárního splajnu procházejícího body $[x_i, y_i]$. Konkrétně, je-li $L(x)$ lineární splajn, požadujeme:

- a) když má $L(x)$ ve vnitřním uzlu lokální extrém, pak ho tam má také $S(x)$;
- b) když $L(x)$ mezi dvěma sousedními uzly neklesá, pak tam také $S(x)$ neklesá, a podobně, když $L(x)$ mezi dvěma sousedními uzly neroste, pak tam neroste ani $S(x)$.

Jednu zdařilou implementaci lze najít v MATLABu jako funkci `pchip`, viz [Matlab], [Moler]. Výpočet směrnic d_i probíhá podle následujících pravidel:

- a) **Vnitřní uzly.** Jestliže směrnice δ_i a δ_{i+1} mají opačná znaménka, nebo je-li některá z nich rovna nule, tj. když platí

$$\delta_i \delta_{i+1} \leq 0, \quad \text{položíme} \quad d_i = 0.$$

V opačném případě určíme d_i jako zobecněný harmonický průměr směrnic δ_i a δ_{i+1} ze vztahu

$$\frac{w_1 + w_2}{d_i} = \frac{w_1}{\delta_i} + \frac{w_2}{\delta_{i+1}}, \quad \text{kde} \quad w_1 = h_i + 2h_{i+1}, \quad w_2 = 2h_i + h_{i+1}.$$

- b) **Okrajové uzly x_0 a x_n .** Nejjednodušší je položit $d_0 = \delta_1$, $d_n = \delta_n$. V algoritmu `pchip` se ale používá kvalitnější aproximace vycházející z kvadratické interpolace, viz [Moler].

Kubický interpolační splajn (dále jen kubický splajn). Směrnice d_i ve vnitřních uzlech můžeme určit také tak, že požadujeme, aby splajn $S(x)$ měl navíc spojitou také druhou derivaci, tj. aby platilo

$$S_i''(x_i) = S_{i+1}''(x_i), \quad i = 1, 2, \dots, n-1. \quad (3.21)$$

Derivováním (3.19) dostaneme

$$S_i''(x) = \frac{(6h_i - 12s)\delta_i + (6s - 4h_i)d_{i-1} + (6s - 2h_i)d_i}{h_i^2}.$$

Pro $x = x_i$ je $s = h_i$, takže

$$S_i''(x_i) = \frac{-6\delta_i + 2d_{i-1} + 4d_i}{h_i}.$$

Pro $x = x_{i-1}$ je $s = 0$ a

$$S_i''(x_{i-1}) = \frac{6\delta_i - 4d_{i-1} - 2d_i}{h_i}.$$

Když v posledním vzorci zvětšíme index i o jedničku, dostaneme

$$S''_{i+1}(x_i) = \frac{6\delta_{i+1} - 4d_i - 2d_{i+1}}{h_{i+1}}.$$

Dosazením do (3.21) tak dostaneme rovnice

$$h_{i+1}d_{i-1} + 2(h_{i+1} + h_i)d_i + h_id_{i+1} = 3(h_{i+1}\delta_i + h_i\delta_{i+1}), \quad i = 1, 2, \dots, n-1. \quad (3.22)$$

Jestliže předepíšeme **okrajové podmínky**

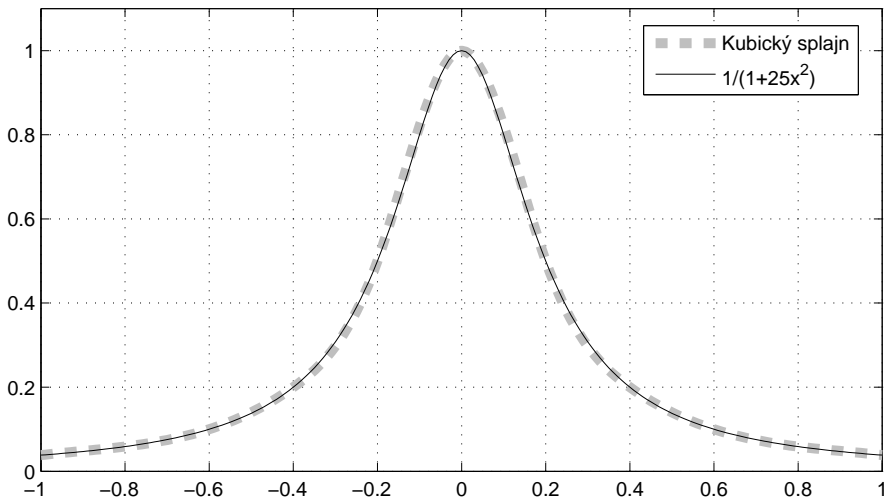
$$S'(a) = d_a, \quad S'(b) = d_b, \quad (3.23)$$

pak v soustavě (3.22) dosadíme v první rovnici $d_0 := d_a$ a člen h_2d_a převedeme na pravou stranu, a v poslední rovnici dosadíme $d_n := d_b$ a člen $h_{n-1}d_b$ převedeme na pravou stranu. Soustavu pak řešíme a získáme zbývající směrnice $d_i, i = 1, 2, \dots, n-1$.

Matice soustavy je třídiagonální, diagonálně dominantní, takže soustavu lze snadno vyřešit GEM upravenou pro soustavy s třídiagonální maticí.

Jestliže $y_i = f(x_i)$, $i = 0, 1, \dots, n$, $d_0 = f'(x_0)$, $d_n = f'(x_n)$, a když $f \in C^4\langle a, b \rangle$, pak pro chybu interpolace opět platí (3.20).

Obrázek 3.3 potvrzuje, že pomocí kubického splajnu lze pro data stejná jako v příkladu 3.4 dostat zcela vyhovující aproximaci Rungeovy funkce.



Obr. 3.3: Aproximace Rungeovy funkce kubickým splajnem

Kubický splajn má pozoruhodnou **extrémální vlastnost**, kterou si teď popíšeme. Označíme

$$V = \{v \in C^2\langle a, b \rangle \mid v(x_i) = y_i, i = 0, 1, \dots, n, v'(x_0) = d_0, v'(x_n) = d_n\}$$

množinu všech funkcí, které mají v intervalu $\langle a, b \rangle$ spojitou druhou derivaci, procházejí zadanými body $[x_i, y_i]$, $i = 0, 1, \dots, n$, a v krajních bodech $a = x_0$ a $x_n = b$ jejich derivace nabývají předepsaných hodnot d_0 a d_n . Pak $\int_a^b [v''(x)]^2 dx$ nabývá na množině funkcí V své nejmenší hodnoty pro kubický splajn $S(x)$, tj. platí

$$\int_a^b [S''(x)]^2 dx = \min_{v \in V} \int_a^b [v''(x)]^2 dx.$$

Tato vlastnost má zajímavou interpretaci v mechanice. Je totiž známo, že ohybová energie homogenního izotropního prutu, jehož střednicová čára má rovnici $y = v(x)$, $x \in \langle a, b \rangle$, má přibližně hodnotu $E(v) = c \int_a^b [v''(x)]^2 dx$, kde c je vhodná konstanta. A dále je také známo, že prut, který je donucen procházet pevnými interpolačními body $[x_i, y_i]$ takovým způsobem, že je namáhán pouze silami působícími kolmo k prutu, zaujme pozici s minimální energií. Extrémální vlastnost tedy tvrdí, že kubický splajn aproximuje střednicovou čáru takového prutu.

Jestliže směrnice d_a a d_b v krajních bodech intervalu $\langle a, b \rangle$ neznáme, osvědčil se postup, označovaný v anglicky psané literatuře jako **not a knot**. Myšlenka je jednoduchá: požadujeme, aby splajn byl jednoduchým polynomem třetího stupně na prvních dvou intervalech, tj. pro $x_0 \leq x \leq x_2$, a na posledních dvou intervalech, tj. pro $x_{n-2} \leq x \leq x_n$. V uzlech x_1 a x_{n-1} tedy už nedochází k napojování dvou různých polynomů, tj. uzel x_1 a x_{n-1} už není „knot“, česky **uzel splajnu**, odtud název postupu „not a knot“.

Polynomy $S_1(x)$ a $S_2(x)$ mají v bodě x_1 společnou funkční hodnotu y_1 , stejnou první derivaci d_1 a podle (3.21) také stejnou druhou derivaci. Aby oba polynomy byly totožné stačí, když budou mít v bodě x_1 také stejnou třetí derivaci. Stejnou úvahu lze provést v bodě x_{n-1} . Dostáváme tak okrajové podmínky

$$S_1'''(x_1) = S_2'''(x_1), \quad S_{n-1}'''(x_{n-1}) = S_n'''(x_{n-1}). \quad (3.24)$$

Když pomocí (3.19) vyjádříme podmínku $S_1'''(x_1) = S_2'''(x_1)$ a upravíme ji pomocí první rovnice soustavy (3.22), dostaneme rovnici

$$h_2 d_0 + (h_2 + h_1) d_1 = [(3h_1 + 2h_2)h_2 \delta_1 + h_1^2 \delta_2] / (h_1 + h_2). \quad (3.25)$$

Podobně zpracujeme také podmínku $S_{n-1}'''(x_{n-1}) = S_n'''(x_{n-1})$ a dostaneme rovnici

$$(h_n + h_{n-1}) d_{n-1} + h_{n-1} d_n = [h_n^2 \delta_{n-1} + (2h_{n-1} + 3h_n)h_{n-1} \delta_n] / (h_{n-1} + h_n). \quad (3.26)$$

Neznámé d_0, d_1, \dots, d_n pak dostaneme jako řešení soustavy rovnic, z nichž první je rovnice (3.25), pak následuje $n - 1$ rovnic (3.22) a nakonec přijde ještě rovnice (3.26). Použijeme opět GEM upravenou pro soustavy s třídiagonální maticí.

Shrnutí. Kubický interpolační splajn $S(x)$ je funkce, která

- a) je v intervalu $\langle a, b \rangle$ spojitá spolu se svou první a druhou derivací, tj.
 $S \in C^2\langle a, b \rangle$,
- b) splňuje interpolační podmínky $S(x_i) = y_i$, $i = 0, 1, \dots, n$, kde y_i jsou předepsané funkční hodnoty,
- c) je na každém intervalu $\langle x_{i-1}, x_i \rangle$ polynom (nejvýše) třetího stupně,
- d) splňuje okrajové podmínky (3.23) nebo (3.24).

Příklad 3.6. Oblouk $[x(t), y(t)] \equiv [\cos t, \sin t]$, $t \in \langle 0, \pi/2 \rangle$, budeme aproximovat křivkou $[S^x(t), S^y(t)]$, kde $S^x(t)$ resp. $S^y(t)$ jsou kubické splajny funkcí $\cos t$ resp. $\sin t$ na intervalu $\langle 0, \pi/2 \rangle$. Interval $\langle 0, \pi/2 \rangle$ rozdělíme na n stejných dílků, takže $t_i = \pi i / (2n)$. Označíme $d_i^x = [S^x]'(t_i)$, $d_i^y = [S^y]'(t_i)$. Protože $x'(t) = -\sin t$, $y'(t) = \cos t$, položíme

$$d_0^x = -\sin 0 = 0, \quad d_n^x = -\sin(\pi/2) = -1, \quad d_0^y = \cos 0 = 1, \quad d_n^y = \cos(\pi/2) = 0.$$

V dalším budeme uvažovat $n = 3$. Soustavu rovnic (3.22) sestavíme zvlášť pro x -ovou a zvlášť pro y -ovou složku. Matice obou soustav je stejná, pravé strany jsou však různé. Na dalším řádku je uvedena soustava rovnic se dvěma pravými stranami a její řešení:

$$\frac{1}{6}\pi \begin{pmatrix} 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \end{pmatrix} \begin{pmatrix} d_1^x & d_1^y \\ d_2^x & d_2^y \\ -1 & 0 \end{pmatrix} = 3 \begin{pmatrix} -\frac{1}{2} & \frac{1}{2}\sqrt{3} \\ -\frac{1}{2}\sqrt{3} & \frac{1}{2} \end{pmatrix} \quad \begin{aligned} d_1^y &= -d_2^x \doteq 0,865537, \\ d_2^y &= -d_1^x \doteq 0,499813. \end{aligned}$$

$S_i^x(t)$ a $S_i^y(t)$ určíme podle (3.19). Například pro $t \in \langle \pi/6, \pi/3 \rangle$ dostaneme

$$S_2^x(t) \doteq 0,8660 - 0,4998(t - \frac{1}{6}\pi) - 0,4431(t - \frac{1}{6}\pi)^2 + 0,1195(t - \frac{1}{6}\pi)^3,$$

$$S_2^y(t) \doteq 0,5000 + 0,8655(t - \frac{1}{6}\pi) - 0,2554(t - \frac{1}{6}\pi)^2 - 0,1195(t - \frac{1}{6}\pi)^3. \quad \square$$

Konstrukce kubického interpolačního splajnu užitím druhých derivací.

Snadno ověříme, že kubický polynom

$$S_i(x) = y_{i-1} + s \frac{6\delta_i - 2h_i M_{i-1} - h_i M_i}{6} + s^2 \frac{M_{i-1}}{2} + s^3 \frac{M_i - M_{i-1}}{6h_i}. \quad (3.27)$$

splňuje podmínky

$$\begin{aligned} S_i(x_{i-1}) &= y_{i-1}, & S_i''(x_{i-1}) &= M_{i-1}, \\ S_i(x_i) &= y_i, & S_i''(x_i) &= M_i. \end{aligned}$$

Funkce $S(x)$, která je na každém intervalu $\langle x_{i-1}, x_i \rangle$ definována předpisem (3.27), proto zřejmě splňuje podmínky $S(x_i) = y_i$, $S''(x_i) = M_i$, $i = 0, 1, \dots, n$. $S(x)$ je tedy na intervalu $\langle a, b \rangle$ spojitá a má v něm spojitou druhou derivaci. Abychom dostali kubický splajn, musí mít $S(x)$ v $\langle a, b \rangle$ spojitou také první derivaci. Protože nespojitost $S'(x)$ může nastat jedině ve vnitřních uzlech, stačí požadovat

$$S'_i(x_i) = S'_{i+1}(x_i), \quad i = 1, 2, \dots, n-1. \quad (3.28)$$

Vyjádříme-li (3.28) pomocí (3.27), dostaneme rovnice

$$h_i M_{i-1} + 2(h_i + h_{i+1})M_i + h_{i+1}M_{i+1} = 6(\delta_{i+1} - \delta_i), \quad i = 1, 2, \dots, n-1. \quad (3.29)$$

Když zvolíme okrajové podmínky

$$S''(a) = M_a, \quad S''(b) = M_b, \quad (3.30)$$

dosadíme je do (3.29), soustavu rovnic vyřešíme a získáme M_i , $i = 1, 2, \dots, n-1$.

Všimněte si, že matice soustavy (3.29) je symetrická. Je samozřejmě možné uvažovat také jiné typy okrajových podmínek, např. (3.23) nebo (3.24).

Kubický splajn s vlastností $S''(a) = S''(b) = 0$ se nazývá **přirozený kubický splajn**. Je známo, že přirozený kubický splajn aproximuje průhyb prostě podepřeného (homogenního izotropního) nosníku namáhaného (silami působícími kolmo k nosníku) tak, že tento nosník prochází body $[x_i, y_i]$. M_i mají význam ohybových momentů v $[x_i, y_i]$.

Interpolace funkcí více proměnných

Omezíme se na případ, kdy f je funkce dvou proměnných definovaná v oblasti Ω .

Interpolace po částech lineární. Předpokládejme, že Ω je mnohoúhelník. Oblast Ω **triangulujeme**, tj. vyjádříme ji jako sjednocení trojúhelníků T_1, T_2, \dots, T_m , z nichž každé dva různé buďto nemají žádný společný bod nebo mají společný vrchol popřípadě mají společnou stranu. Množinu $\mathcal{T} = \{T_k\}_{k=1}^m$ všech takových trojúhelníků nazýváme **triangulací** oblasti Ω . Vrcholy trojúhelníků triangulace označíme $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$, \dots , $P_n = [x_n, y_n]$ a nazveme je **uzly triangulace**. Předpokládejme, že v každém uzlu P_i je předepsána hodnota $f_i = f(x_i, y_i)$ interpolované funkce f .

Po částech lineárním interpolantem funkce f na oblasti Ω rozumíme funkci S , která je v Ω spojitá, splňuje interpolační podmínky $S(x_i, y_i) = f_i$, $i = 1, 2, \dots, n$, a která je na každém trojúhelníku $T_k \in \mathcal{T}$ lineární. Na T_k je tedy $z = S(x, y) \equiv S_k(x, y)$ rovnice roviny určené hodnotami funkce f ve vrcholech T_k . Připomeňme, že rovnice roviny procházející body $[x_a, y_a, z_a]$, $[x_b, y_b, z_b]$ a $[x_c, y_c, z_c]$ může být vyjádřena ve tvaru

$$\begin{vmatrix} x - x_a & y - y_a & z - z_a \\ x_b - x_a & y_b - y_a & z_b - z_a \\ x_c - x_a & y_c - y_a & z_c - z_a \end{vmatrix} = 0.$$

Vypočítat hodnotu $z = S(x, y)$ pro $(x, y) \in \Omega$ je snadné: určíme trojúhelník T_k , v němž bod $[x, y]$ leží, a vypočteme $z = S_k(x, y)$.

Chyba interpolace je tím menší, čím jemnější triangulaci zvolíme. Když $f \in C^2(\Omega)$ (tj. když f je v Ω spojitá spolu se svými prvními a druhými parciálními derivacemi), pak

$$|f(x, y) - S(x, y)| \leq Ch^2,$$

kde h je nejdelší strana trojúhelníků triangulace a C je konstanta nezávislá na h .

Interpolace po částech bilineární. Předpokládejme, že $\Omega = \langle a, b \rangle \times \langle c, d \rangle$ je obdélník. Pomocí dělení $a = x_0 < x_1 < \dots < x_n = b$, $c = y_0 < y_1 < \dots < y_m = d$ rozložíme výchozí obdélník Ω na menší obdélníky

$$R_{ij} = \{(x, y) \mid x_{i-1} \leq x \leq x_i, y_{j-1} \leq y \leq y_j\}, \quad i = 1, 2, \dots, n, j = 1, 2, \dots, m.$$

Předpokládejme, že v uzlech $[x_i, y_j]$ jsou předepsány hodnoty $f_{ij} = f(x_i, y_j)$ funkce f .

Na obdélníku R_{ij} definujeme funkci

$$\begin{aligned} S_{ij}(x, y) = & f_{i-1,j-1} \frac{x_i - x}{x_i - x_{i-1}} \frac{y_j - y}{y_j - y_{j-1}} + f_{i,j-1} \frac{x - x_{i-1}}{x_i - x_{i-1}} \frac{y_j - y}{y_j - y_{j-1}} + \\ & + f_{i-1,j} \frac{x_i - x}{x_i - x_{i-1}} \frac{y - y_{j-1}}{y_j - y_{j-1}} + f_{ij} \frac{x - x_{i-1}}{x_i - x_{i-1}} \frac{y - y_{j-1}}{y_j - y_{j-1}}. \end{aligned}$$

Funkce S_{ij} je bilineární, tj. pro pevné $x = C$ je $S_{ij}(C, y)$ lineární funkce proměnné y a pro pevné $y = D$ je $S_{ij}(x, D)$ lineární funkce proměnné x . Funkce S_{ij} je interpolant funkce f na obdélníku R_{ij} , tj. S_{ij} nabývá ve vrcholech $[x_{i-1}, y_{j-1}]$, $[x_i, y_{j-1}]$, $[x_{i-1}, y_j]$ a $[x_i, y_j]$ stejných hodnot jako funkce f , jak se snadno přesvědčíme.

Na Ω definujeme funkci S předpisem $S(x, y) = S_{ij}(x, y)$ pro $(x, y) \in R_{ij}$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$. Funkce S je po částech (= po obdélnících R_{ij}) bilineární. Protože $S(x_i, y_j) = f_{ij}$, $i = 0, 1, \dots, n$, $j = 0, 1, \dots, m$, řekneme, že S je **po částech bilineární interpolant** funkce f na obdélníku Ω . Snadno ověříme, že S je v Ω spojitá (stačí si uvědomit, že S_{ij} , a tedy také S , je na každé straně obdélníka R_{ij} jednoznačně určena pomocí hodnot funkce f v koncových bodech této strany). Když $f \in C^2(\Omega)$, pak pro chybu interpolace platí odhad

$$|S(x, y) - f(x, y)| \leq C(h^2 + k^2), \quad \text{kde } h = \max_{1 \leq i \leq n} (x_i - x_{i-1}), \quad k = \max_{1 \leq j \leq m} (y_j - y_{j-1})$$

a C je konstanta nezávislá na h , k .

Obsah

- 3 Aproximace funkcí
 - Aproximace a interpolace
 - Interpolace
 - Interpolace polynomem
 - Interpoláčn  splajny
 - Interpolace funkc  v ce prom nn ch
 - Metoda nejmen  ch  tverc 
 - Literatura

označuje postup pro přibližné řešení přeuročených nebo nepřesně zadaných soustav rovnic, založený na minimalizaci kvadrátů jejich reziduí.

Prokládání dat křivkami je významná skupina úloh, které lze metodou nejmenších čtverců řešit (v anglicky psané literatuře se pro tyto aplikace používá označení **curve fitting**). Popišme si, o co v takových úlohách jde.

Nechť tedy t je nezávisle proměnná, například čas, a $y(t)$ je neznámá funkce proměnné t , kterou chceme aproximovat. Předpokládejme, že jsme provedli m **pozorování**, tj. hodnoty y byly změřeny pro určité (navzájem různé) hodnoty t :

$$y_i = y(t_i), \quad i = 1, 2, \dots, m.$$

Naším záměrem je modelovat $y(t)$ lineární kombinací n **bázových funkcí** pro nějaké $n \leq m$:

$$y(t) \approx x_1\varphi_1(t) + x_2\varphi_2(t) + \dots + x_n\varphi_n(t) := R_n(t).$$

Bázové funkce navrhujeme podle očekávaného průběhu neznámé funkce $y(t)$, určit se mají **parametry** x_1, x_2, \dots, x_n . Funkce $R_n(t)$ se ve statistice nazývá **lineární regresní funkce**.

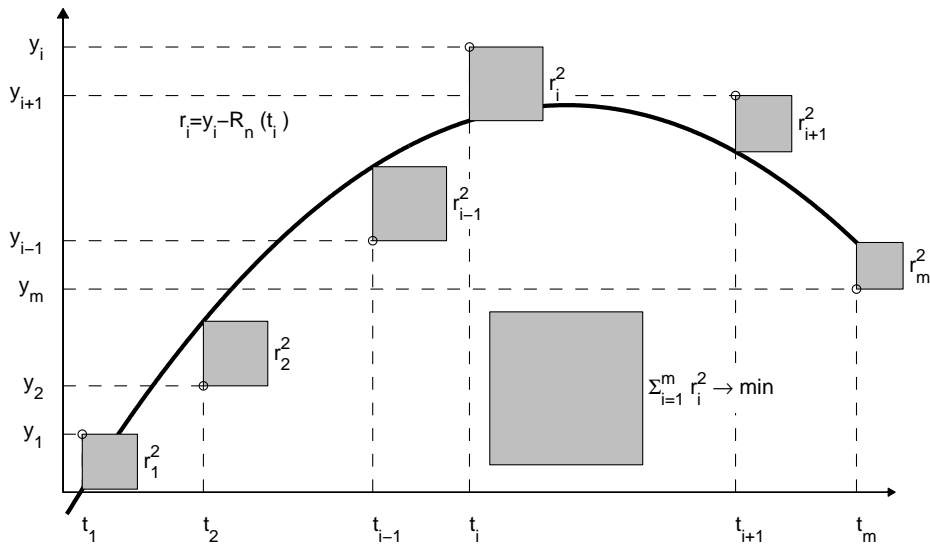
Návrhová matice \mathbf{A} modelu je obdélníková matice, která má m řádků a n sloupců:

$$\mathbf{A} = \begin{pmatrix} \varphi_1(t_1) & \varphi_2(t_1) & \dots & \varphi_n(t_1) \\ \varphi_1(t_2) & \varphi_2(t_2) & \dots & \varphi_n(t_2) \\ \vdots & \vdots & & \vdots \\ \varphi_1(t_m) & \varphi_2(t_m) & \dots & \varphi_n(t_m) \end{pmatrix} \equiv (\varphi_1, \varphi_2, \dots, \varphi_n),$$

kde $\varphi_i = (\varphi_i(t_1), \varphi_i(t_2), \dots, \varphi_i(t_m))^T$ je i -tý sloupec \mathbf{A} . Maticová formulace modelu je

$$\mathbf{y} \approx \mathbf{Ax},$$

kde $\mathbf{y} = (y_1, y_2, \dots, y_m)^T$ jsou naměřená data a $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ je vektor neznámých parametrů. Symbol \approx vyjadřuje přibližnou rovnost.



Obr. 3.4: Princip metody nejmenších čtverců

Rezidua jsou rozdíly mezi pozorováními a modelem:

$$r_i = y_i - R_n(t_i) = y_i - \sum_{j=1}^n \varphi_j(t_i) x_j \equiv y_i - \sum_{j=1}^n a_{ij} x_j, \quad i = 1, 2, \dots, n,$$

kde $a_{ij} = \varphi_j(t_i)$. V maticovém zápisu

$$\mathbf{r} = \mathbf{y} - \mathbf{A}\mathbf{x}. \quad (3.31)$$

Parametry x_i chceme určit tak, aby rezidua byla co nejmenší. Metodu nejmenších čtverců dostaneme, když minimalizujeme součet čtverců reziduí:

$$\|\mathbf{r}\|^2 := \sum_{i=1}^m r_i^2 \rightarrow \min. \quad (3.32)$$

Někdy se používá také **vážená metoda nejmenších čtverců**: když jsou některá pozorování důležitější nebo přesnější než ostatní, pak můžeme přisoudit jednotlivým pozorováním různé váhy $w_i > 0$ a minimalizovat součet vážených čtverců

$$\|\mathbf{r}\|_w^2 := \sum_{i=1}^m w_i r_i^2 \rightarrow \min .$$

Je-li například chyba i -tého pozorování přibližně rovna e_i , zvolíme $w_i = 1/e_i$. Je dobré si uvědomit, že každou metodu pro řešení nevážené metody nejmenších čtverců lze použít pro řešení metody vážené: stačí pronásobit y_i a i -tý řádek **A** součinitelem $\sqrt{w_i}$.

Přibližné řešení přeúřčené soustavy rovnic $\mathbf{Ax} = \mathbf{y}$ (tj. rovnic je víc než neznámých), které minimalizuje délku rezidua $\mathbf{r} = \mathbf{y} - \mathbf{Ax}$, se nazývá **řešení soustavy lineárních rovnic metodou nejmenších čtverců**.

Normální soustava rovnic. Řešení minimalizační úlohy (3.32) musí splňovat nutnou podmínku pro extrém:

$$\frac{\partial \|\mathbf{r}\|^2}{\partial x_k} = \frac{\partial}{\partial x_k} \sum_{i=1}^m \left(y_i - \sum_{j=1}^n a_{ij} x_j \right)^2 = 0, \quad k = 1, 2, \dots, n.$$

Když provedeme naznačené derivování, dostaneme

$$\frac{\partial \|\mathbf{r}\|^2}{\partial x_k} = 2 \sum_{i=1}^m \left(y_i - \sum_{j=1}^n a_{ij} x_j \right) (-a_{ik}) = 0$$

a odtud

$$\sum_{j=1}^n \left(\sum_{i=1}^m a_{ik} a_{ij} \right) x_j = \sum_{i=1}^m a_{ik} y_i, \quad k = 1, 2, \dots, n,$$

což lze zapsat maticově jako

$$\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{y}. \quad (3.33)$$

Soustava lineárních rovnic (3.33) je známa jako **normální soustava rovnic**. Když jsou sloupce matice \mathbf{A} lineárně nezávislé, je matice $\mathbf{G} := \mathbf{A}^T \mathbf{A}$ pozitivně definitní a řešení \mathbf{x}^* normální soustavy rovnic je jediné řešení úlohy (3.32), tj. platí

$$\|\mathbf{y} - \mathbf{A}\mathbf{x}^*\|^2 = \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2.$$

Vyjádříme-li normální soustavu rovnic pomocí vektorů φ_i , dostaneme

$$\begin{pmatrix} (\varphi_1, \varphi_1) & (\varphi_1, \varphi_2) & \cdots & (\varphi_1, \varphi_n) \\ (\varphi_2, \varphi_1) & (\varphi_2, \varphi_2) & \cdots & (\varphi_2, \varphi_n) \\ \vdots & \vdots & \cdots & \vdots \\ (\varphi_n, \varphi_1) & (\varphi_n, \varphi_2) & \cdots & (\varphi_n, \varphi_n) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} (\varphi_1, \mathbf{y}) \\ (\varphi_2, \mathbf{y}) \\ \vdots \\ (\varphi_n, \mathbf{y}) \end{pmatrix}, \quad (3.34)$$

kde

$$(\varphi_k, \varphi_j) = \sum_{i=1}^m \varphi_k(t_i) \varphi_j(t_i) \quad \text{a} \quad (\varphi_k, \mathbf{y}) = \sum_{i=1}^m \varphi_k(t_i) y_i$$

jsou skalární součiny vektorů φ_k, φ_j a φ_k, \mathbf{y} . Matice \mathbf{G} soustavy (3.34) se nazývá **Gramova matice** soustavy vektorů $\varphi_j, j = 1, 2, \dots, n$.

Při návrhu aproximace $R_n(t)$ bychom měli vybírat funkce $\varphi_i(t)$ tak, aby sloupce φ_i matice \mathbf{A} byly lineárně nezávislé. V opačném případě, jak se dá ukázat, má úloha (3.32) nekonečně mnoho řešení, což je zřejmě nežádoucí.

Uveďme si dva významné speciální případy, pro které jsou sloupce matice \mathbf{A} lineárně nezávislé (důkaz lze najít např. v [Berezin]):

- a) $\varphi_j(t)$ je polynom stupně $j - 1$, např. $\varphi_j(t) = t^{j-1}$, $j = 1, 2, \dots, n$;
- b) pro $n = 2N + 1$, kde N je celé nezáporné číslo, volíme

$$\varphi_1(t) = 1, \quad \varphi_{2k}(t) = \cos kt, \quad \varphi_{2k+1}(t) = \sin kt, \quad k = 1, 2, \dots, N,$$

a „časy pozorování“ t_i vybíráme z intervalu $\langle c, c + 2\pi \rangle$, kde c je libovolné číslo.

Aproximace $R_n(t)$ je v případě a) algebraický polynom a v případě b) trigonometrický polynom.

Když je $m = n$ a matice \mathbf{A} je regulární, pak $\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{y}$ a $\mathbf{r} = \mathbf{0}$, tj. $R_n(t_i) = y_i$, $i = 1, 2, \dots, m$. Pokud jsou však naměřená data y_i zatížena chybami, pak není účelné, aby funkce $R_n(t)$ tyto chyby kopírovala. Naopak, aby $R_n(t)$ věrohodně vystihovala (rekonstruovala) neznámou funkci $y(t)$, je žádoucí, aby $R_n(t)$ naměřená data **vyrovnávala (vyhlazovala)**. To je ale možné jen tehdy, když počet pozorování m je výrazně větší než počet n návrhových parametrů, tj. pro $m \gg n$.

Příklad 3.7. Pro data předepsaná tabulkou

t_i	0	0,5	1	1,5	2	2,5	3
y_i	3,57	2,99	2,62	2,33	2,22	2,10	2,05

určíme aproximaci $R_2(t) = x_1 + x_2 e^{-t}$ metodou nejmenších čtverců. Zřejmě $\varphi_1(t) = 1$ a $\varphi_2(t) = e^{-t}$. Normální soustava rovnic je tvaru

$$\begin{pmatrix} \sum_{i=1}^7 1 \cdot 1 & \sum_{i=1}^7 1 \cdot e^{-t_i} \\ \sum_{i=1}^7 e^{-t_i} \cdot 1 & \sum_{i=1}^7 e^{-t_i} \cdot e^{-t_i} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^7 1 \cdot y_i \\ \sum_{i=1}^7 e^{-t_i} \cdot y_i \end{pmatrix}.$$

Vypočteme-li příslušné sumy, dostaneme soustavu (zobrazujeme nejvýše 4 desetinná místa)

$$\begin{pmatrix} 7 & 2,4647 \\ 2,4647 & 1,5805 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 17,88 \\ 7,4422 \end{pmatrix},$$

jejíž řešení je $x_1 \doteq 1,9879$ a $x_2 \doteq 1,6087$. Hledaná aproximace $R_2(t) \doteq 1,99 + 1,61e^{-t}$ a $\|\mathbf{r}\| \doteq 0,0651$. \square

Příklad 3.8. Daty z předchozího příkladu proložíme postupně polynomy prvního, druhého a třetího stupně. Za bázové volíme funkce $\varphi_j(t) = t^{j-1}$, kde $j = 1, 2, \dots, n$ a $n = 2, 3, 4$.

- a) **Polynom prvního stupně.** Pro $\varphi_1(t) = 1$ a $\varphi_2(t) = t$ dostaneme normální soustavu rovnic

$$\begin{pmatrix} 7 & 10,5 \\ 10,5 & 22,75 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 17,88 \\ 23,45 \end{pmatrix},$$

která má řešení $x_1 \doteq 3,28$ a $x_2 \doteq -0,48$, takže $R_2(t) \doteq 3,28 - 0,48t$ a $\|\mathbf{r}\| \doteq 0,4756$. Aproximace lineárním polynomem tedy není vhodná, neboť je málo přesná.

- b) **Polynom druhého stupně.** Normální soustava rovnic

$$\begin{pmatrix} 7 & 10,5 & 22,75 \\ 10,5 & 22,75 & 55,125 \\ 22,75 & 55,125 & 142,1875 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 17,88 \\ 23,45 \\ 49,0625 \end{pmatrix}$$

má řešení $x_1 \doteq 3,53$, $x_2 \doteq -1,09$ a $x_3 \doteq 0,20$, takže $R_3(t) \doteq 3,53 - 1,09t + 0,2t^2$ a $\|\mathbf{r}\| \doteq 0,1006$. Velikost rezidua se zmenšila, je však stále větší než v příkladu 3.7.

- c) **Polynom třetího stupně.** Normální soustava rovnic (zobrazujeme nejvýše 4 desetinná místa)

$$\begin{pmatrix} 7 & 10,5 & 22,75 & 55,125 \\ 10,5 & 22,75 & 55,125 & 142,1875 \\ 22,75 & 55,125 & 142,1875 & 381,2813 \\ 55,125 & 142,1875 & 381,2813 & 1049,5469 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 17,88 \\ 23,45 \\ 49,0625 \\ 116,78 \end{pmatrix}$$

má řešení $x_1 \doteq 3,57$, $x_2 \doteq -1,35$, $x_3 \doteq 0,43$ a $x_4 \doteq -0,05$, takže $R_4(t) \doteq 3,57 - 1,35t + 0,43t^2 - 0,05t^3$ a $\|\mathbf{r}\| \doteq 0,0360$.

Pokud bychom stupeň polynomu dále zvyšovali, zjistili bychom, že polynom $R_7(t)$ šestého stupně prochází všemi body $[t_i, y_i]$, takže je to interpolační polynom.

Všimněte si ještě prvků Gramových matic: s rostoucím řádem největší koeficient prudce roste. Spolu s ním prudce roste také číslo podmíněnosti Gramových matic. Do následující tabulky jsme zaznamenali čísla podmíněnosti $\kappa_2(\mathbf{G})$ pro $n = 2, 3, \dots, 7$ (spočtená v MATLABu pomocí maticové $\|\cdot\|_2$ normy)










n	2	3	4	5	6	7
$\kappa_2(\mathbf{G})$	16	$4,27 \cdot 10^2$	$1,91 \cdot 10^4$	$1,20 \cdot 10^6$	$1,17 \cdot 10^8$	$2,31 \cdot 10^{10}$









Pokud bychom tabulku dat z příkladu 3.7 rozšířili o další sloupce, mohli bychom teoreticky v regresní funkci $\sum_{j=1}^n x_j t^{j-1}$ zvětšovat n (až do počtu m sloupců). Prakticky to však možné není, neboť pro velké n by číslo podmíněnosti Gramovy matice bylo neúnosně velké. Regresní funkce $R_n(t)$ s báзовými funkcemi $\varphi_j(t) = t^{j-1}$, $j = 1, 2, \dots, n$, se proto pro větší n nepoužívají. Pokud potřebujeme polynomickou regresní funkci vyššího stupně, měli bychom použít tzv. **ortogonální polynomy**, viz např. [Stoer, Bulirsch]. \square







Řešení přeürčených soustav rovnic. Číslo podmíněnosti $\kappa_2(\mathbf{A})$ je definováno také pro obdélníkovou matici \mathbf{A} a platí $\kappa_2(\mathbf{A}^T \mathbf{A}) = [\kappa_2(\mathbf{A})]^2$. To znamená, že číslo podmíněnosti Gramovy matice $\mathbf{A}^T \mathbf{A}$ je výrazně větší než číslo podmíněnosti návrhové matice \mathbf{A} . Normální rovnice se proto k řešení rozsáhlých soustav přeürčených rovnic nehodí. Existují lepší postupy využívající tzv. **QR rozklad** matice \mathbf{A} nebo tzv. **pseudoinverzi** matice \mathbf{A} , viz např. [Stoer, Bulirsch], [Moler]. Obě tyto metody jsou stabilní a Gramovu matici nepoužívají. Kvalitní implementace jsou dostupné např. v MATLABu, viz [Matlab]. Tyto metody se také bez potíží vypořádkají s případem, když jsou sloupce matice \mathbf{A} lineárně závislé. Dá se ukázat, že v tom případě má přeürčená soustava rovnic nekonečně mnoho řešení (ve smyslu metody nejmenších čtverců, tj. má nekonečně mnoho řešení, která mají stejnou délku rezidua). Jestliže \mathbf{A}^+ je pseudoinverze matice \mathbf{A} , pak $\mathbf{A}^+ \mathbf{y}$ je jednoznačně definované řešení, které minimalizuje délku rezidua, a pokud je takových řešení nekonečně mnoho, pak $\mathbf{A}^+ \mathbf{y}$ je to z nich, které má nejmenší délku.

Obsah

- 3 Aproximace funkcí
 - Aproximace a interpolace
 - Interpolace
 - Interpolace polynomem
 - Interpoláčn  splajny
 - Interpolace funkc  v ice prom enn ch
 - Metoda nejmen  ch  tverc 
 - Literatura

-  I.S. Berezin, N.P. Židkov: *Číslennyje metody I,II*, Nauka, Moskva, 1962.
-  G. Dahlquist, G. Å Björk: *Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
-  M. Fiedler: *Speciální matice a jejich použití v numerické matematice*, SNTL, Praha, 1981.
-  D. Hanselman, B. Littlefield: *Mastering MATLAB 7*, Pearson Prentice Hall, Upper Saddle River, NJ, 2005.
-  G. Hämmerlin, K. H. Hoffmann: *Numerical Mathematics*, Springer-Verlag, Berlin, 1991.
-  M. T. Heath: *Scientific Computing. An Introductory Survey*, McGraw-Hill, New York, 2002.
-  I. Horová, J. Zelinka: *Numerické metody*, učební text Masarykovy univerzity, Brno, 2004.
-  J. Kobza: *Splajny*, učební text Palackého univerzity, Olomouc, 1993.
-  J. Klapka, J. Dvořák, P. Popela: *Metody operačního výzkumu*, učební text, FSI VUT Brno, 2001.

-  J. H. Mathews, K. D. Fink: *Numerical Methods Using MATLAB*, Pearson Prentice Hall, New Jersey, 2004.
-  MATLAB: *Mathematics*, Version 7, The MathWorks, Inc., Natick, 2004.
-  G. Meurant: *Computer Solution of Large Linear Systems*, Elsevier, Amsterdam, 1999.
-  S. Míka: *Numerické metody algebry*, SNTL, Praha, 1985..
-  C. B. Moler: *Numerical Computing with MATLAB*, Siam, Philadelphia, 2004.
<http://www.mathworks.com/moler>.
-  J. Nocedal, S. J. Wright: *Numerical Optimization, Springer Series in Operations Research*, Springer, Berlin, 1999.
-  A. Quarteroni, R. Sacco, F. Saleri: *Numerical Mathematics*, Springer, Berlin, 2000.
-  W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling: *Numerical Recipes in Pascal, The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1990.

-  P. Přikryl: *Numerické metody matematické analýzy*, SNTL, Praha, 1985.
-  A. R. Ralston: *Základy numerické matematiky*, Academia, Praha, 1973.
-  K. Rektorys: *Přehled užití matematiky I,II*, Prometheus, Praha, 1995.
-  J. Stoer, R. Bulirsch: *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1993.
-  E. Vitásek: *Numerické metody*, SNTL, Praha, 1987.
-  W. Y. Yang, W. Cao, T. S. Chung, J. Morris: *Applied Numerical Methods Using Matlab*, John Willey & Sons, New Jersey, 2005.