Kapitola 7. Aproximace funkcí - I

Motivace

Při numerickém řešení úloh matematické analýzy často nahrazujeme danou funkci f, vystupující v řešené matematické úloze, jinou funkcí φ, která v nějakém vhodném smyslu napodobuje funkci f a snadno se přitom matematicky zpracovává či modeluje na počítači. Tuto funkci φ nazýváme **aproximací funkce** f.

Poznámka: Aproximaci funkce jsme již používali u řešení nelineární rovnice. Například u Newtonovy metody jsme danou funkci f z řešené rovnice $f(x) = 0$ aproximovali lineární funkcí (tečnou ke grafu funkce f); podobně tak tomu bylo u metody sečen.

Poznámka: Již pouhý výpočet funkčních hodnot některých základních funkcí ($\sin x$, e^x, $\ln x$, $\ldots$) v počítači či na kalkulačce se provádí užitím aproximace těchto funkcí. Tyto aproximace jsou ovšem zabudovány do výpočetního systému a uživatel si často ani neuvědomuje, že píše-li v programu např. `y=sin(x)`, nahrazuje výpočet hodnoty funkce $\sin x$ výpočtem hodnoty jistého polynomu.

Příklady užití:

- numerické metody pro výpočet určitého integrálu
- zpracování výsledků měření

Formulace (základní aproximační úloha)

Je dána funkce $f = f(x)$, $x \in \langle a, b\rangle$. Zvolíme $(n+1)$ lineárně nezávislých funkcí $\varphi_0, \varphi_1, \ldots \varphi_n$ a hledáme funkci φ definovanou na $\langle a, b\rangle$, kterou lze vyjádřit ve tvaru lineární kombinace

$$\boxed{\varphi(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \ldots + c_n\varphi_n(x)}$$

a která je v nějakém smyslu blízká funkci f.

- Tento typ aproximace se nazývá **lineární aproximace**
- Pokud za funkce $\varphi_i(x)$ volíme polynomy, mluvíme o **polynomiální aproximaci**
- Příkladem **nelineární aproximace** je funkce $\varphi(x)$:

$$\boxed{\varphi(x) = \frac{a_0 + a_1x + \ldots + a_mx^m}{1 + b_1x + \ldots + b_nx^n}}$$

V tomto případě mluvíme o **racionální aproximaci**

Přístupy k aproximaci

Aproximace na okolí bodu - Použijeme, chceme-li aproximovat chování funkce v malém okolí bodu. Příkladem může být např. vyčíslení hodnoty $\sin\frac{\pi}{4}$ na kalkulačce.

Interpolace - Použijeme, chceme-li tabulkou danými body proložit polynom, tj. požadujeme-li, aby aproximace přesně procházela zadanými body.

L_2-aproximace - Použijeme, hledáme-li funkční závislost mezi tabulkou danými body (získaných například měřením), kde nutně nevyžadujeme, aby aproximace danými body procházela. Důvodem můžou být např. chyby, se kterými jsme hodnoty naměřili.

Věta (Weierstrassova)

Nechť funkce $f(x)$ je spojitá na $\langle a, b\rangle$. Potom ke každému $\varepsilon > 0$ existuje přirozené číslo n a polynom $p_n(x)$ stupně nejvýše n tak, že

$$\boxed{\max_{x\in\langle a,b\rangle} |f(x) - p_n(x)| < \varepsilon}$$

Polynom $p_n(x)$ se nazývá **nejlepší aproximací** spojité funkce f na $\langle a, b\rangle$.

Aproximace na okolí bodu

– mluvíme o **aproximaci Taylorovým polynomem**

Předpokládáme, že daná funkce f má v daném bodě x_0 a jeho okolí spojité derivace až do řádu n. Podmínky pro funkci, která co nejlépe napodobuje chování funkce f matematicky zapíšeme takto:

$$\varphi^{(j)}(x_0) = f^{(j)}(x_0), \quad j = 0, 1, \ldots, n$$

(Hodnoty derivací funkcí f, φ v bodě x_0 jsou stejné až do řádu n.)

Tuto podmínku samozřejmě splňuje Taylorův polynom

$$T_n(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$$

Pro chybu aproximace Taylorovým polynomem platí

$$e(x) = f(x) - T_n(x) = f^{(n+1)}(\xi)\frac{(x - x_0)^{n+1}}{(n+1)!}, \quad \xi \in \mathcal{U}(x_0)$$

umíme-li odhadnout $(n + 1)$-ní derivaci funkce f na daném okolí bodu x_0, můžeme provést následující odhad chyby aproximace:

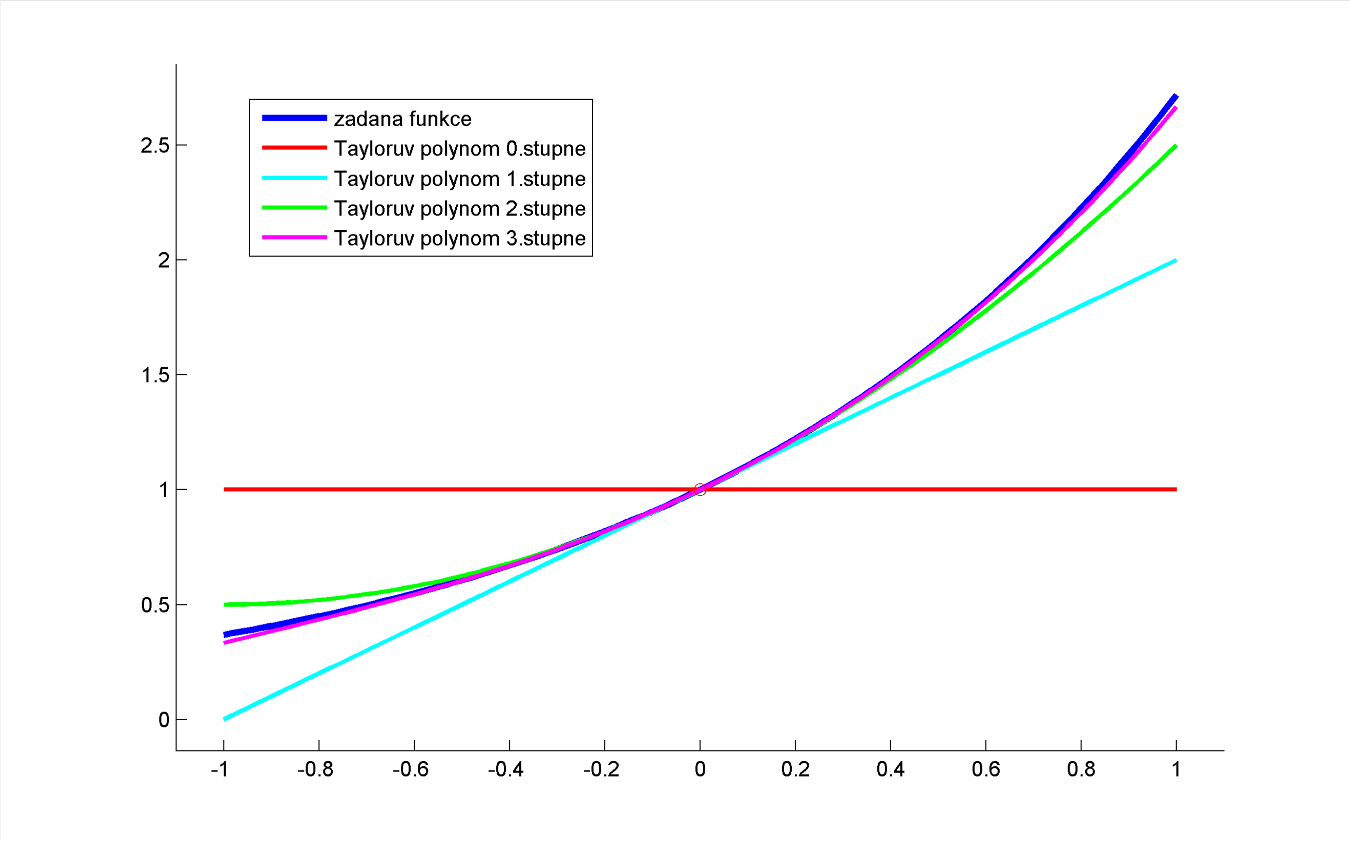
Platí-li $|f^{(n+1)}(x)| \le M \quad \forall x \in \mathcal{U}(x_0)$, potom $|e(x)| \le \dfrac{M}{(n+1)!}|x - x_0|^{n+1}$.

Příklad Stanovte odhad chyby aproximace Taylorovým polynomem 10. stupně funkce $f(x) = \mathrm{e}^x$ v bodě $x_0 = 0$ na intervalu $\langle -1, 1\rangle$.

$$f^{(j)}(x) = \mathrm{e}^x,\ j = 0, 1, \ldots; \quad f^{(j)}(0) = 1,\ j = 0, 1, \ldots$$

$$\boxed{T_n(x) = 1 + x + \frac{x^2}{2} + \ldots + \frac{x^{10}}{10!}}$$

$$e(x) = \mathrm{e}^x - T^{10}(x) = \frac{\mathrm{e}^{\xi}}{11!}x^{11} \quad \Rightarrow \quad \boxed{|e(x)| \leq \frac{\mathrm{e}}{11!} \leq 7.10^{-8}}$$



Příklad Určete stupeň Taylorova polynomu funkce $f(x) = \sin x$ v bodě $x_0 = 0$ tak, aby jeho chyba na intervalu $\langle -\pi, \pi\rangle$ byla nejvýše 10^{-5}, resp. (10^{-12}).

Pro chybu platí

$$\boxed{e(x) = f(x) - T_n(x) = \frac{(x - x_0)^{n+1}}{(n+1)!} f^{(n+1)}(\xi)}.$$

Zajímá nás chyba na intervalu délky 2π $\Rightarrow$ odhad pro $f^{(n+1)}(x)$ je $\Big|\underbrace{f^{(n+1)}(x)}_{(*)}\Big| \leq 1 \quad \forall x \in \langle -\pi, \pi\rangle$

$(*)$ buď $\pm \sin x$ nebo $\pm \cos x$ - tento odhad je vždy nejmenší možný

$$\left|\frac{x^{n+1}}{(n+1)!}\right| \leq 10^{-5} \quad \forall x \in \langle -\pi, \pi \rangle$$

$$\frac{\pi^{n+1}}{(n+1)!} \leq 10^{-5} \quad (\text{resp. } 10^{-12})$$

n	$\frac{\pi^{n+1}}{(n+1)!}$
0	3.141592653589793
1	4.934802200544679
2	5.167712780049969
3	4.058712126416768
4	2.550164039877345
5	1.335262768854589
6	0.599264529320792
7	0.235330630358893
8	0.082145886611128
9	0.025806891390014
10	0.007370430945714
11	0.001929574309404
12	0.000466302805768
13	0.000104638104925
14	0.000021915353448
15	**0.000004303069587** $< 10^{-5}$
16	0.000000795205400
17	0.000000138789525
18	0.000000022948429
19	0.000000003604731
20	0.000000000539266
21	0.000000000077007
22	0.000000000010518
23	0.000000000001377
24	**0.000000000000173** $< 10^{-12}$

Jak vypadá Taylorův polynom $T_n(x)$?

$$\begin{array}{lll|lll}
f(x) & = \sin x & & f(0) & = 0 \\
f'(x) & = \cos x & & f'(0) & = 1 \\
f''(x) & = -\sin x & & f''(0) & = 0 \\
f'''(x) & = -\cos x & & f'''(0) & = -1 \\
\hline
f^{(4)}(x) & = \sin x & & f^{(4)}(0) & = 0 \\
& \vdots & & & \vdots
\end{array}$$

$$T_n(x) = \underbrace{f(x_0)}_{\to 0} + \underbrace{f'(x_0)}_{\to 1}(x - \underbrace{x_0}_{\to 0}) + \frac{1}{2!}\underbrace{f''(x_0)}_{\to 0}(x - \underbrace{x_0}_{\to 0})^2 + \ldots + \frac{1}{n!} f^{(n)}(x_0)(x - \underbrace{x_0}_{\to 0})^n$$

$$\boxed{T_{15}(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \frac{x^{13}}{13!} - \frac{x^{15}}{15!}}$$

Má-li kalkulačka vypočítat např. $\sin 0{,}4$ a má povolenou chybu 10^{-5}, určí $\boxed{\sin 0{,}4 \doteq T_{15}(0{,}4)}$.

Pro dodržení chyby 10^{-12} stačí určit $\boxed{\sin 0{,}4 \doteq T_{24}(0{,}4)}$, tj. přičíst další 4 členy.

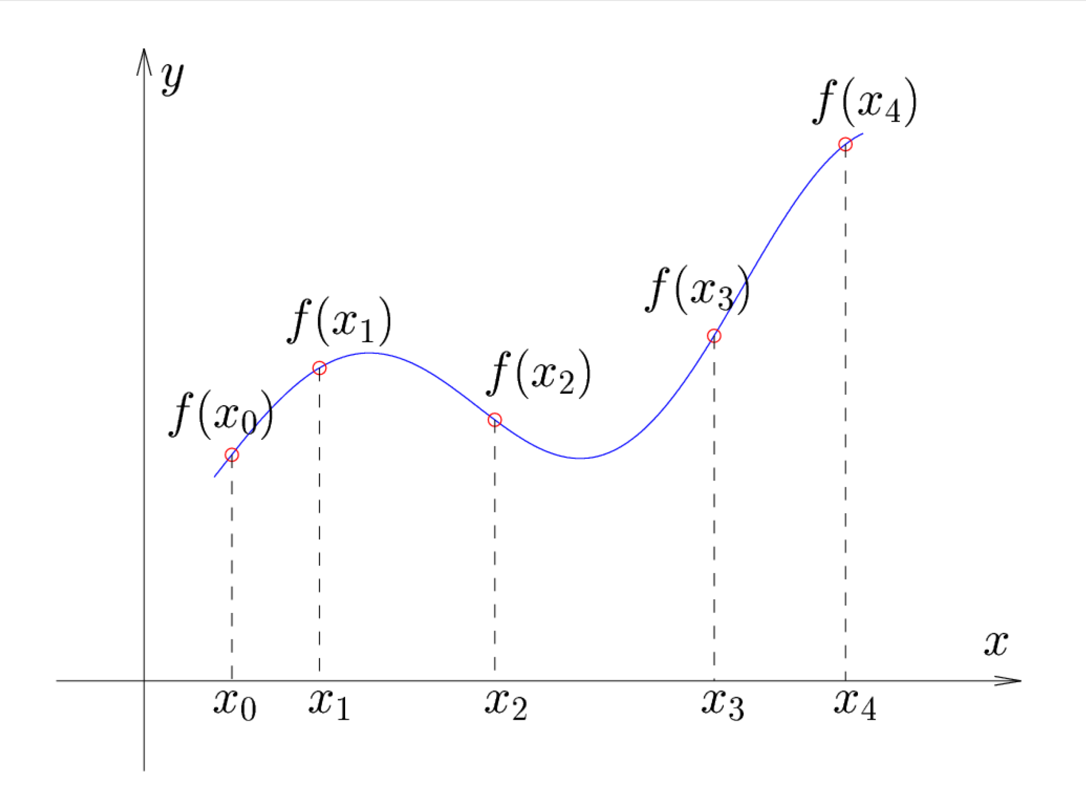
Poznámka:

Stejný postup lze užít i pro výpočet v bodech mimo interval $\langle -\pi, \pi \rangle$. Stačí využít periodičnosti funkce $\sin x$.

Aproximace interpolačním polynomem

Aproximujeme funkci, která je dána svými hodnotami v $n+1$ bodech x_i, $i = 0, 1, \ldots, n$ (body x_i nazýváme uzly interpolace), a požadujeme, aby aproximace (interpolačním polynomem) procházela zadanými body.

Aproximace nám potom poslouží k získání přibližné hodnoty zadané funkce v libovolném bodě intervalu $\langle x_0, x_n \rangle$.



Interpolační podmínky

$$\boxed{P_n(x_i) = f(x_i), \quad i = 0, 1, \ldots, n}$$

Chyba $e(x) = f(x) - P_n(x)$ pak nabývá nulových hodnot v uzlech interpolace.

Věta Interpolační úloha má jediné řešení, pokud jsou uzly $x_0, x_1, \ldots, x_n$ navzájem různé.

Důkaz: Interpolační polynom uvažujeme ve tvaru

$$\boxed{P_n(x) = \sum_{k=0}^{n} a_k x^k}.$$

Dosazením do interpolačních podmínek získáme soustavu $(n+1)$ lineárních rovnic pro koeficienty $a_0, a_1, \ldots, a_n$.

$$\boxed{\sum_{k=0}^{n} a_k x_i^k = f(x_i) \quad i = 0, 1, \ldots, n}$$

Matice soustavy (Vandermondova matice):

$$\mathbf{V} = \begin{bmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{bmatrix}$$

Soustava má právě jedno řešení, pokud $\boxed{\det \mathbf{V} \neq 0}$.

Matici soustavy převedeme na trojúhelníkový tvar

- přičtením násobku řádku k jinému řádku se determinant nezmění
- vynásobíme-li řádek číslem α, pak je determinant α násobkem původního

Od 2. až $(n+1)$-ního řádku odečteme 1. řádek.

$$\mathbf{V} \sim \begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 & \ldots & x_0^n \\ 0 & x_1 - x_0 & x_1^2 - x_0^2 & x_1^3 - x_0^3 & \ldots & x_1^n - x_0^n \\ 0 & x_2 - x_0 & x_2^2 - x_0^2 & x_2^3 - x_0^3 & \ldots & x_2^n - x_0^n \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & x_n - x_0 & x_n^2 - x_0^2 & x_n^3 - x_0^3 & \ldots & x_n^n - x_0^n \end{bmatrix}$$

Vynormujeme – $(j+1)$-ní řádek vydělíme $(x_j - x_0)$, $j = 1, 2, \ldots, n$.

$$\mathbf{V} \sim \begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 & \dots & x_0^n \\ 0 & 1 & \dfrac{x_1^2 - x_0^2}{x_1 - x_0} & \dfrac{x_1^3 - x_0^3}{x_1 - x_0} & \dots & \dfrac{x_1^n - x_0^n}{x_1 - x_0} \\ 0 & 1 & \dfrac{x_2^2 - x_0^2}{x_2 - x_0} & \dfrac{x_2^3 - x_0^3}{x_2 - x_0} & \dots & \dfrac{x_2^n - x_0^n}{x_2 - x_0} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 1 & \dfrac{x_n^2 - x_0^2}{x_n - x_0} & \dfrac{x_n^3 - x_0^3}{x_n - x_0} & \dots & \dfrac{x_n^n - x_0^n}{x_n - x_0} \end{bmatrix}$$

<u>od 3. až $(n+1)$-ního řádku odečteme 2. řádek</u>

$$\mathbf{V} \sim \begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 & \dots & x_0^n \\ 0 & 1 & x_1 + x_0 & x_1^2 + x_1 x_0 + x_0^2 & \dots & \dots \\ 0 & 0 & x_2 - x_1 & (*) & \dots & \dots \\ 0 & 0 & x_3 - x_1 & (**) & \dots & \dots \\ \vdots & \vdots & \vdots & & & \\ 0 & 0 & x_n - x_1 & \dots & \dots & \dots \end{bmatrix}$$

$$(*) = \frac{x_2^3 - x_0^3}{x_2 - x_0} - \frac{x_1^3 - x_0^3}{x_1 - x_0} = (x_2^2 + x_2 x_0 + x_0^2) - (x_1^2 + x_1 x_0 + x_0^2) = x_2^2 - x_1^2 + x_0(x_2 - x_1) = (x_2 - x_1)(x_2 + x_1 + x_0)$$

$$(**) = \frac{x_3^3 - x_0^3}{x_3 - x_0} - \frac{x_1^3 - x_0^3}{x_1 - x_0} = (x_3^2 + x_3 x_0 + x_0^2) - (x_1^2 + x_1 x_0 + x_0^2) = x_3^2 - x_1^2 + x_0(x_3 - x_1) = (x_3 - x_1)(x_3 + x_1 + x_0)$$

<u>Vynormujeme</u> – $(j+1)$-ní řádek vydělíme $(x_j - x_1)$, $j = 2, 3, \dots, n$.

$$\mathbf{V} \sim \begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 & \dots & x_0^n \\ 0 & 1 & x_1 + x_0 & x_1^2 + x_1 x_0 + x_0^2 & \dots & \dots \\ 0 & 0 & 1 & x_2 + x_1 + x_0 & \dots & \dots \\ 0 & 0 & 1 & x_3 + x_1 + x_0 & \dots & \dots \\ \vdots & \vdots & \vdots & & & \\ 0 & 0 & 1 & x_n + x_1 + x_0 & \dots & \dots \end{bmatrix}$$

Získáme trojúhelníkovou matici s jedničkami na diagonále, tj. výsledná matice má determinant roven 1.

Při úpravách jsme dělili $(x_j - x_i)$, $j > i$

$$\Rightarrow \quad \boxed{\det \mathbf{V} = \prod_{j>i}^{n} (x_j - x_i)}$$

$$\det \mathbf{V} \neq 0 \quad \Leftrightarrow \quad x_i \neq x_j \quad \forall i \neq j$$

□

Lagrangeův interpolační polynom

Označme si hledaný polynom n-tého stupně $L_n(x)$.
Víme, že musí být splněny interpolační podmínky

$$L_n(x_i) = f(x_i), \quad i = 0, 1, \ldots, n.$$

Lagrangeův interpolační polynom hledáme ve tvaru

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

kde $l_i(x)$ jsou polynomy n-tého stupně takové, že platí

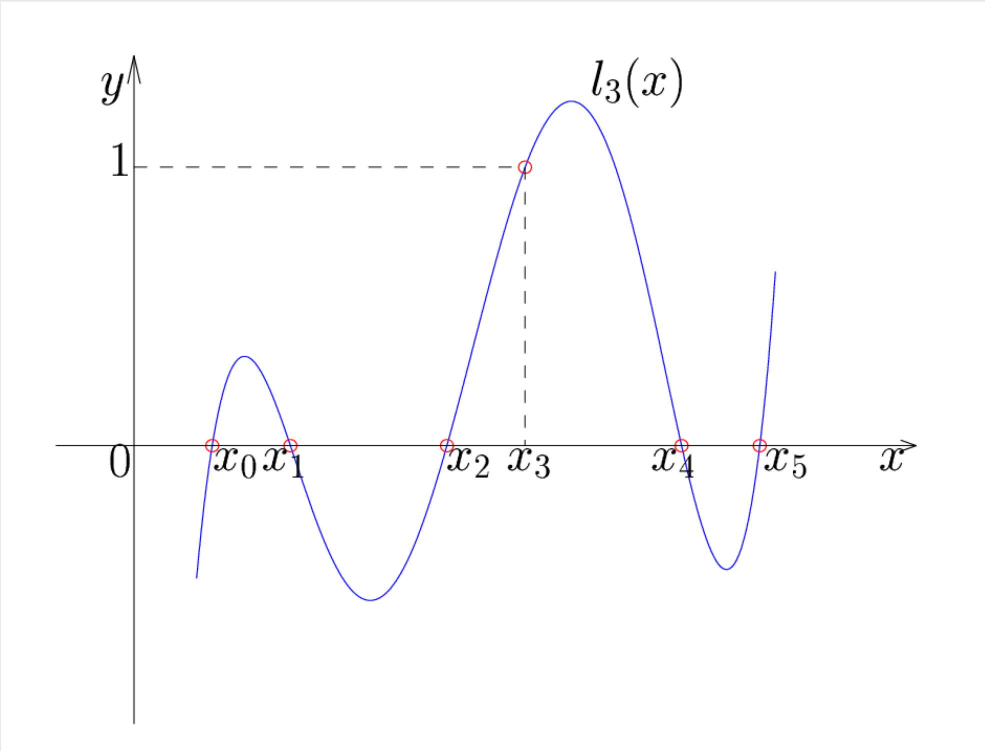
$$l_i(x_j) = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

(snadno se přesvědčíte, že dosadíte-li do předpisu pro $L_n(x)$ uzly interpolace, získáte zadané interpolační podmínky).

Konkretizujme nyní dílčí polynomy $l_i(x)$.
Víme, že $l_i(x)$ má kořeny $x_0, x_1, x_{i-1}, x_{i+1}, \ldots, x_n$ a nabývá hodnoty 1 v bodě x_i.
Můžeme jej tedy zapsat ve tvaru

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

Na obrázku je ukázán příklad dílčího polynomu $l_3(x)$:



Newtonův interpolační polynom

Označme si hledaný polynom n-tého stupně $N_n(x)$.
Pro jeho odvození použijeme jinou kontrukci. Polynom volíme ve tvaru

$$\boxed{N_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \cdots + a_n(x-x_0)(x-x_1)\dots(x-x_{n-1})}.$$

Opět požadujeme splnění interpolačních podmínek

$$N_n(x_i) = f(x_i), \quad i = 0, 1, \dots, n.$$

Poznámka: Výhodou volby tohoto zdánlivě složitého předpisu je fakt, že přidáme-li další bod interpolace $[x_{n+1}, f(x_{n+1})]$, nemusíme celý výpočet opakovat, ale stačí dopočítat příslušný koeficient a_{n+1} (ostatní koeficienty a_i zůstávají beze změny). Při použití Lagrangeova polynomu bychom museli celý výpočet provést znovu.

Ukažme si, co dostaneme dosazováním interpolačních podmínek do předpisu polynomu:

$$N_n(x_0) = \boxed{a_0 = f(x_0)}$$

$$N_n(x_1) = a_0 + a_1(x_1 - x_0) = f(x_1) \quad \Rightarrow \quad \boxed{a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}}$$

$$N_n(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) = f(x_2)$$

$$\Rightarrow \quad a_2 = \frac{f(x_2) - f(x_0) - a_1 \overbrace{(x_2 - x_0)}^{x_2 - x_1 + x_1 - x_0}}{(x_2 - x_0)(x_2 - x_1)} =$$

$$= \frac{f(x_2) - f(x_0) - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_1) - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}(x_1 - x_0)}{(x_2 - x_0)(x_2 - x_1)} =$$

$$= \frac{f(x_2) - f(x_1) - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

$$\Rightarrow \quad \boxed{a_2 = \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}}$$

Poznámka:

Počítat koeficienty a_i přímo ze soustavy není praktické, budeme je počítat pomocí tzv. **poměrných diferencí**.

Algoritmus (koeficienty Newtonova polynomu)

Pro $i = 0, 1, \ldots, n$

$$\boxed{\mathbf{A}_{i0} = f(x_i)}$$

Pro $k = 1, 2, \ldots, i$

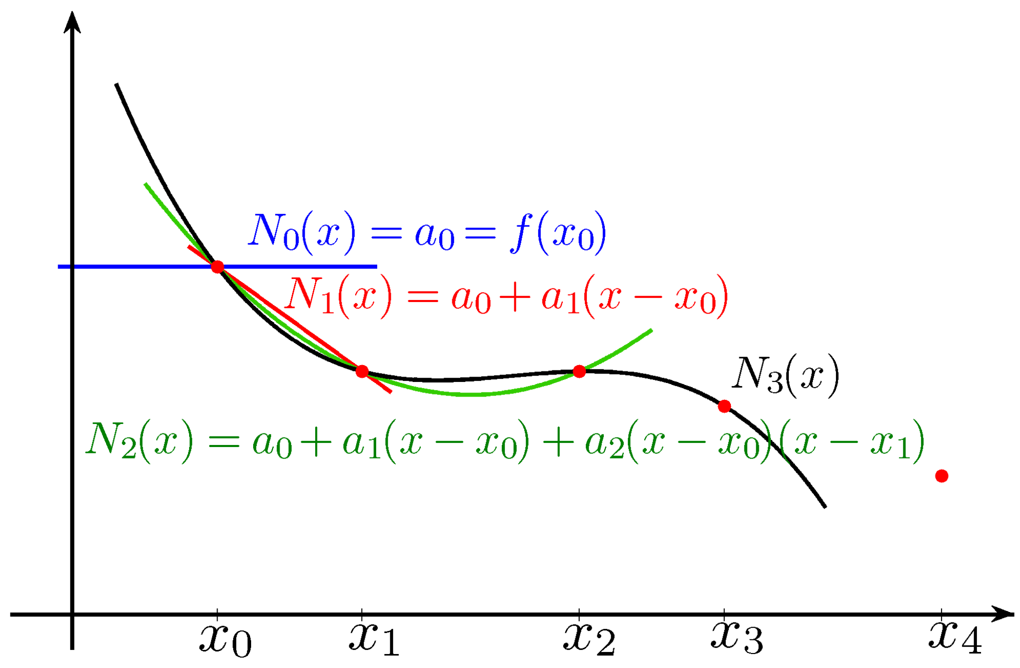
$$\boxed{\mathbf{A}_{ik} = \frac{\mathbf{A}_{i,k-1} - \mathbf{A}_{i-1,k-1}}{x_i - x_{i-k}}}$$

Výstup: $\boxed{a_i = \mathbf{A}_{ii}}$

Schéma algoritmu

$$\begin{array}{c|lllll}
x_0 & f(x_0) = \mathbf{A}_{00} = a_0 & & & & \\
x_1 & f(x_1) = \mathbf{A}_{10} & \mathbf{A}_{11} = a_1 & & & \\
x_2 & f(x_2) = \mathbf{A}_{20} & \mathbf{A}_{21} & \mathbf{A}_{22} = a_2 & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \\
x_n & f(x_n) = \mathbf{A}_{n0} & \mathbf{A}_{n1} & \mathbf{A}_{n2} & \ldots & \mathbf{A}_{nn} = a_n
\end{array}$$

Poznámka: Vezmu-li z matice $\mathbf{A}$ pouze prvních M řádků znamená to, že jsem sestavil Newtonův polynom pro pouze prvních M zadaných tabulkových bodů.



$N_0(x) = a_0$ prochází bodem $[x_0, f(x_0)]$

$N_1(x) = a_0 + a_1(x - x_0)$
prochází bodem $[x_0, f(x_0)]$ a směrnici má takovou, aby procházel také bodem $[x_1, f(x_1)]$

$N_2(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1)$
prochází bodem $[x_0, f(x_0)]$, dále jelikož platí $N_2(x_1) = N_1(x_1)$, prochází i bodem $[x_1, f(x_1)]$ a navíc
prochází bodem $[x_2, f(x_2)]$.

$$N_2(x) = \overbrace{\underbrace{a_0}_{(*)} + \underbrace{a_1(x - x_0)}_{(**)}}^{N_1(x)} + \underbrace{a_2(x - x_0)(x - x_1)}_{(***)}$$

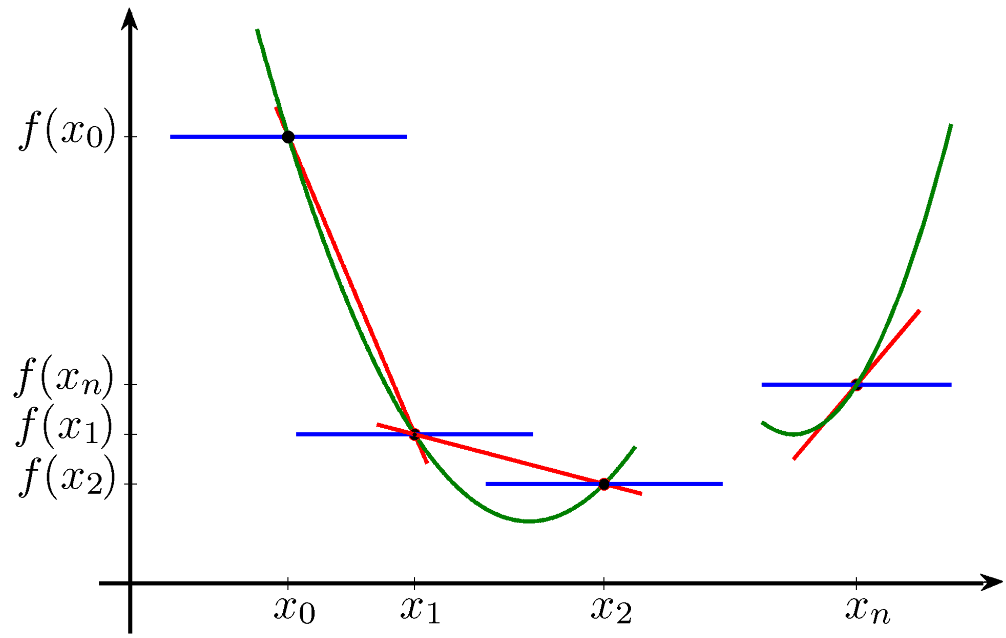
$(*)$ prochází $[x_0, f(x_0)]$

$(**)$ přidáme tento člen tak, aby se zachoval průchod $[x_0, f(x_0)]$ (hodnota pro $x = x_0$ je 0), a_1 se určí
tak, aby procházel $[x_1, f(x_1)]$

$(***)$ přidáme tento člen tak, aby se zachoval průchod $[x_0, f(x_0)]$ a $[x_1, f(x_1)]$ (hodnota členu pro
$x = x_0$ a $x = x_1$ je nulová), a_2 se určí tak, aby procházel bodem $[x_2, f(x_2)]$

Co znamenají jednotlivá čísla v tabulce?

$$
\begin{array}{c|llll}
x_0 & f(x_0) = \mathbf{A}_{00} & & & \\
x_1 & f(x_1) = \mathbf{A}_{10} & \mathbf{A}_{11} & & \\
x_2 & f(x_2) = \mathbf{A}_{20} & \mathbf{A}_{21} & \mathbf{A}_{22} & \\
\vdots & \quad\vdots & \vdots & \vdots & \\
x_n & f(x_n) = \mathbf{A}_{n0} & \mathbf{A}_{n1} & \mathbf{A}_{n2} &
\end{array}
$$



Věta Má-li funkce f, k níž přísluší data $f(x_i)$, $i = 0, 1, \ldots, n$, spojité derivace v nějakém intervalu $\langle a, b\rangle \supset$ int $(x_0, x_1, \ldots, x_n)$ do řádu n a derivaci $f^{(n+1)}(x)$ v (a, b), potom $\forall x \in \langle a, b\rangle\ \exists \xi = \xi(x) \in (a, b)$ tak, že pro chybu interpolačního polynomu $P_n(x)$ platí:

$$\boxed{e(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\underbrace{(x - x_0)(x - x_1)\ldots(x - x_n)}_{\text{polynom } (n+1) \text{ stupně}}} \qquad \circledast$$

(Důkaz pomocí věty o střední hodnotě, viz dále.)

Odhad chyby interpolace

Umíme-li stanovit číslo M takové, že $\forall x \in \langle a,b\rangle$ je $\left|f^{(n+1)}(x)\right| \leq M$, pak

$$\boxed{|e(x)| \leq \frac{M}{(n+1)!}|w(x)| \leq \frac{M}{(n+1)!}\max_{x\in\langle a,b\rangle}|w(x)|},$$

kde jsme označili $w(x) = (x-x_0)(x-x_1)\dots(x-x_n)$.

$\Rightarrow$ Průběh chyby nezávisí jen na $f^{(n+1)}(x)$, ale i na $w(x)$!

Důkaz: Zvolme bod $x \neq x_i$, $i = 0,1,\dots,n$ libovolně. Definujme funkci

$$\boxed{F(t) = f(t) - P_n(t) - \frac{f(x)-P_n(x)}{w_{n+1}(x)}w_{n+1}(t)}. \qquad ⊛⊛$$

Tato funkce proměnné t má $n+2$ nulových bodů:

- $x_i,\ \ i = 0,1,\dots,n$:

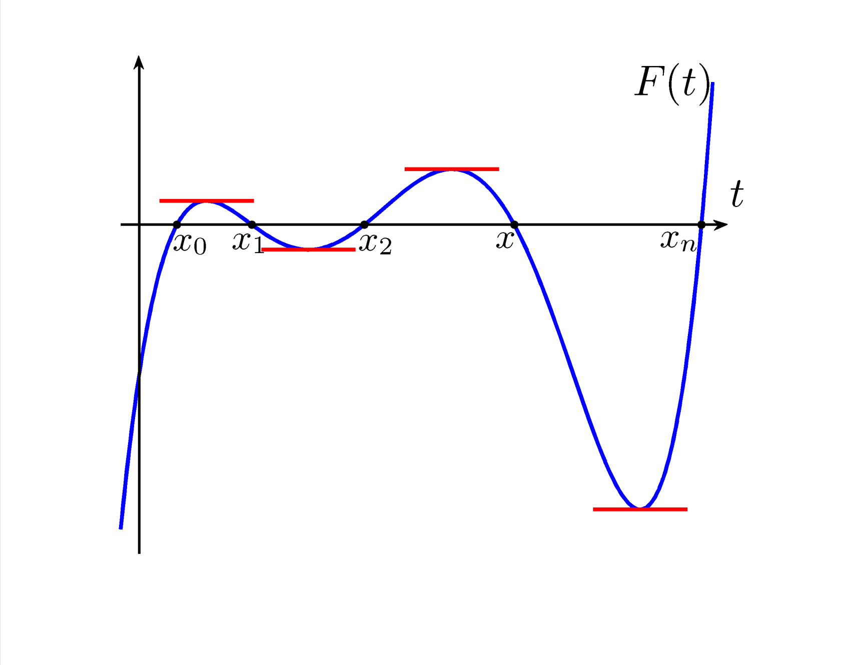
$$F(x_i) = \underbrace{f(x_i) - P_n(x_i)}_{=0} - \frac{f(x)-P_n(x)}{w_{n+1}(x)}\underbrace{w_{n+1}(x_i)}_{=0} = 0$$

- x:

$$F(x) = f(x) - P_n(x) - \frac{f(x)-P_n(x)}{w_{n+1}(x)}w_{n+1}(x) = 0$$

Použijeme-li $(n+1)$krát **Rollovu větu** zjistíme, že první derivace $F'(t)$ má v (a,b) alespoň $n+1$ nulových bodů.

Rollova věta: Nechť $f(x)$ je v $\langle a,b\rangle$ spojitá a má v (a,b) derivaci. Nechť $f(a) = f(b)$. Pak existuje alespoň 1 bod $c \in (a,b)$ tak, že $f'(c) = 0$.



Pokračujeme dál a aplikujeme nyní **Rollovu větu** na funkci $F'(t)$ a zjistíme, že $F''(t)$ má v (a,b) alespoň n nulových bodů, atd.

$\vdots$

$F^{(n+1)}(t)$ má v (a,b) alespoň jeden nulový bod a ten označíme $\xi = \xi(x)$.

Vztah ⊛⊛ zderivujeme $(n+1)$ krát podle t:

$$\underbrace{F^{(n+1)}(t)}_{(\bullet)} = f^{(n+1)}(t) - \underbrace{P_n^{(n+1)}(t)}_{=0\ (\bullet\bullet)} - \frac{f(x)-P_n(x)}{w_{n+1}(x)} \underbrace{w_{n+1}^{(n+1)}(t)}_{=(n+1)!\ (\bullet\bullet\bullet)}$$

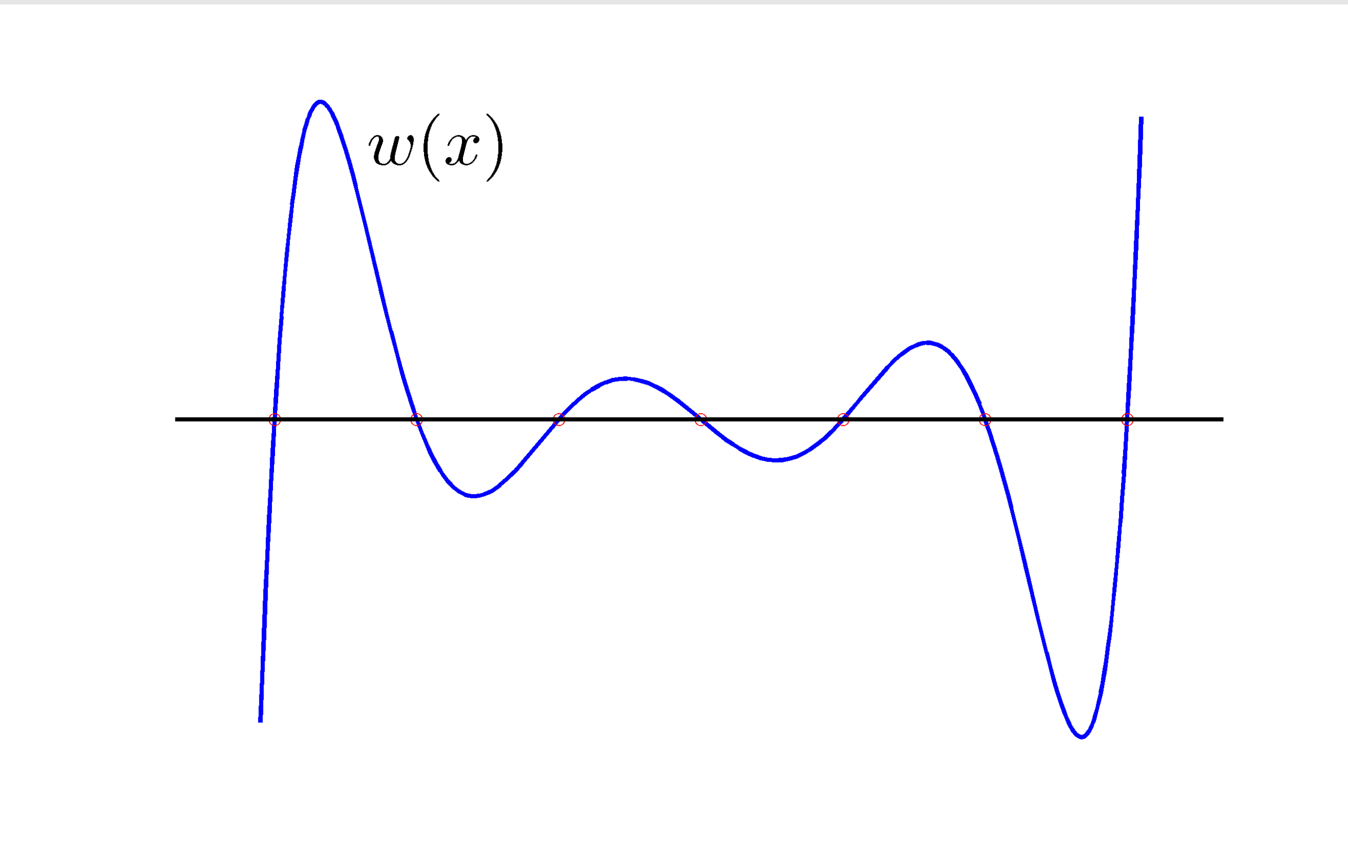
($\bullet$) Víme, že existuje ξ tak, že $F^{(n+1)}(\xi) = 0$
($\bullet\bullet$) P_n ... polynom n-tého stupně
($\bullet\bullet\bullet$) $w_{n+1}(t)$ je polynom $(n+1)$ stupně a u t^{n+1} je koeficient 1

$$0 = f^{(n+1)}(\xi) - \frac{f(x)-P_n(x)}{w_{n+1}(x)}(n+1)! \quad \Rightarrow \quad \boxed{f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\, w_{n+1}(x)}$$

$\square$

- Pokud interpolujeme funkci zadanou v ekvidistantních uzlech, mohou mít nepřesnosti ve vstupních datech silný vliv na hodnotu výsledku → úloha je špatně podmíněná

  Poznámka: Špatná podmíněnost platí tím více i pro extrapolaci.



- Dobrou strategií je volit x_i tak, aby byly rozloženy stejně jako kořeny Čebyševových polynomů $\rightarrow$ minimalizuje se tak hodnota $\max|w(x)|$.

Definice (**Čebyševova aproximace**)

K dané spojité funkci $f(x)$, $x \in \langle a, b\rangle$, chceme najít mezi všemi polynomy $P_n(x)$ stupně nejvýše n takový polynom $P_n^*(x)$, který splňuje:

$$\boxed{\max_{x \in \langle a,b\rangle} |f(x) - P_n^*(x)| = \min_{P_n(x)} \max_{x \in \langle a,b\rangle} |f(x) - P_n(x)|}.$$

Poznámka:

Při volbě ekvidistantních uzlů byla úloha špatně podmíněná.

Při vhodné volbě uzlů (kořeny tzv. **Čebyševových polynomů** - viz dále $\heartsuit$) má interpolační proces pro $n \to \infty$ tu vlastnost, že interpolační polynomy **konvergují** na $\langle a, b\rangle$ **stejnoměrně** k aproximované funkci např. v případě, když existuje spojitá první derivace f' na $\langle a, b\rangle$.

Stejnoměrná konvergence funkce f_n definované na $\langle a, b\rangle$

- varianta 1: posloupnost $\{f_n\}$ je na $\langle a, b\rangle$ stejnoměrně konvergentní

  $\Leftrightarrow$ $\forall \varepsilon > 0$ $\exists n_0 \in \mathbb{N}$: stejné $\forall x \in \langle a, b\rangle$ tak, že

$$\forall n > n_0 \text{ a } \forall x \in \langle a, b\rangle : |f_n(x) - f(x)| < \varepsilon$$

- varianta 2: posloupnost $\{f_n\}$ je na $\langle a, b\rangle$ stejnoměrně konvergentní

  $\Leftrightarrow$ $\lim\limits_{n\to\infty} \sup\limits_{x\in\langle a,b\rangle} |f_n(x) - f(x)| = 0$

Čebyševovy polynomy

$$\boxed{\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_{n+1} &= 2\,x\,T_n(x) - T_{n-1}(x) \qquad \spadesuit \end{aligned}}$$

Užijeme-li substituci $x = \cos\alpha$, $\alpha = \arccos x$ a goniometrické vzorce, dostaneme

$$\boxed{T_n(x) = \cos(n \arccos x) \quad x \in \langle -1, 1\rangle \qquad \clubsuit}$$

Důkaz:

$$T_0(x) = \cos(0 \arccos x) = 1 \quad \checkmark$$
$$T_1(x) = \cos(1 \arccos x) = x \quad \checkmark$$
$$T_n = \cos(n \arccos x) \quad \ldots \quad \text{dosadíme do vztahu } \spadesuit$$

$$\cos\big(\underbrace{(n+1)\arccos x}_{\alpha}\big) = 2x\cos(n\arccos x) - \cos\big(\underbrace{(n-1)\arccos x}_{\beta}\big)$$

Platí:

$$\boxed{\cos\alpha + \cos\beta = 2\cos\frac{\alpha+\beta}{2}\cos\frac{\alpha-\beta}{2}} \quad \checkmark$$

Kořeny polynomů jsou

$$\cos(n\arccos x) = 0$$

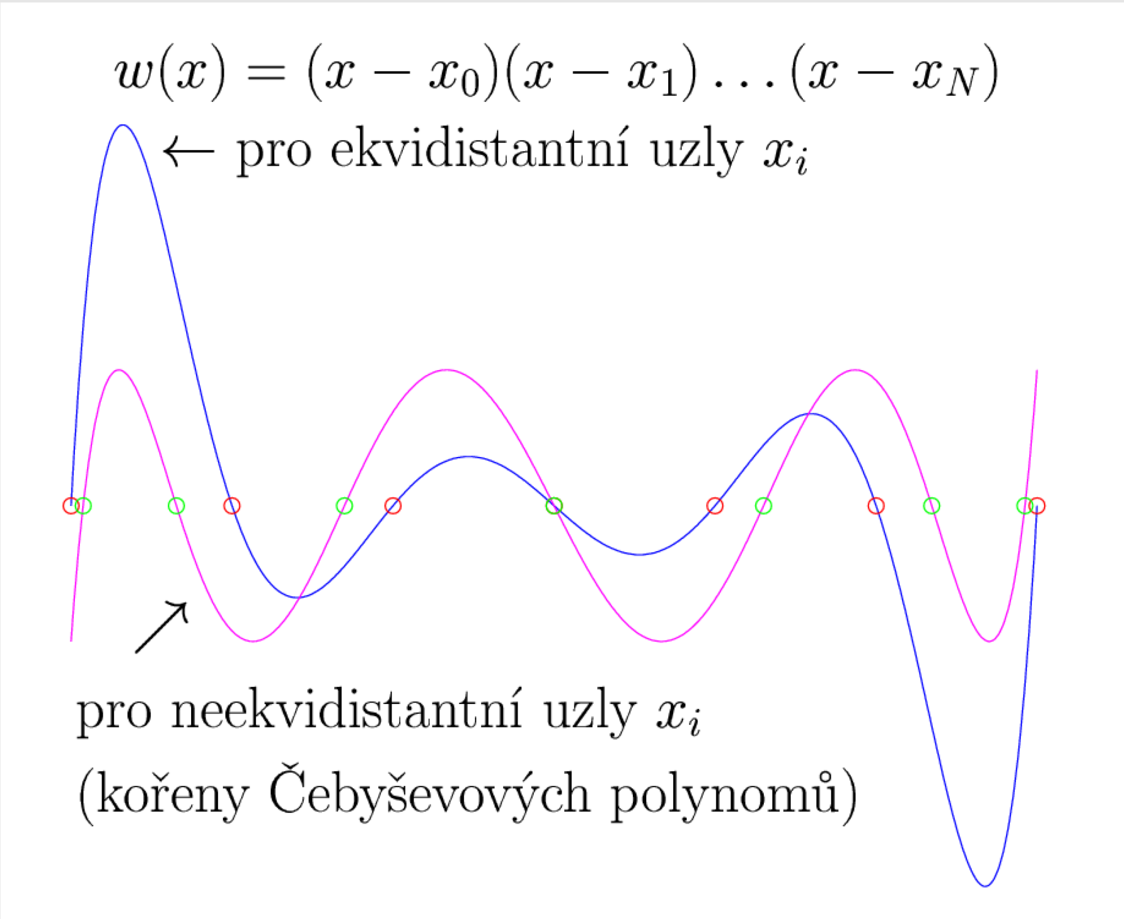
$$n\arccos x_k = \frac{\pi}{2} + k\pi, \quad k \in \mathbb{Z}$$

$$\arccos x_k = \frac{2k+1}{n}\cdot\frac{\pi}{2}$$

$$\boxed{x_k = \cos\left(\frac{2k+1}{n}\cdot\frac{\pi}{2}\right), \quad k = 0, 1, \dots, n-1} \quad \heartsuit$$

Poznámka: Pro obecný interval $\langle a, b\rangle$ použijeme transformaci

$$r_k = \frac{b-a}{2}x_k + \frac{a+b}{2}.$$



Příklad

Uvažujme funkci $f(x) = \ln(x+1)$ zadanou na intervalu $\langle 0, 5\rangle$.

Pro zvolený počet n uzlových bodů proveďte interpolaci zadané funkce pro uzlové body, které jsou

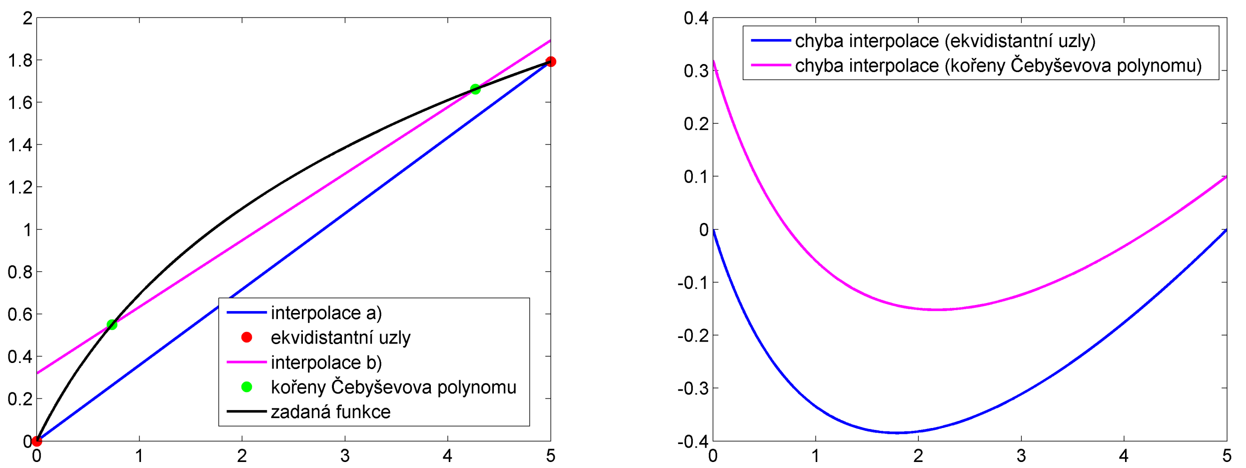
a) ekvidistantní,

b) kořeny Čebyševových polynomů.

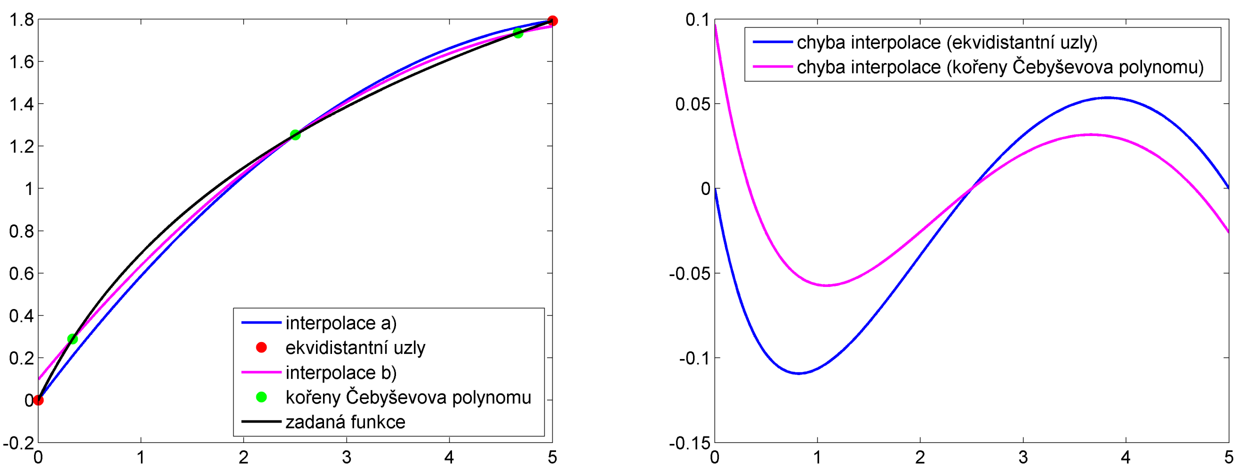
Porovnejte chybu získaných interpolačních polynomů.

---

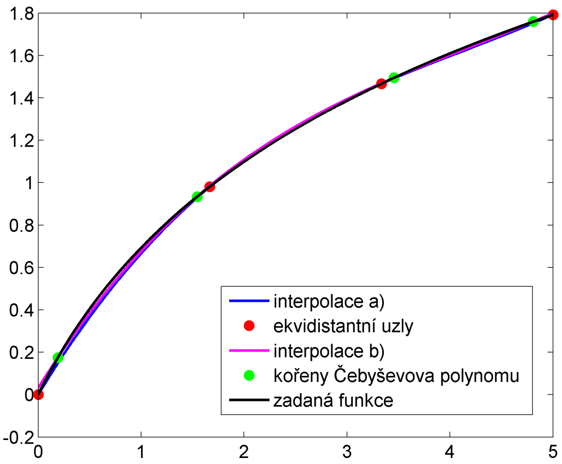
1) $n = 2$

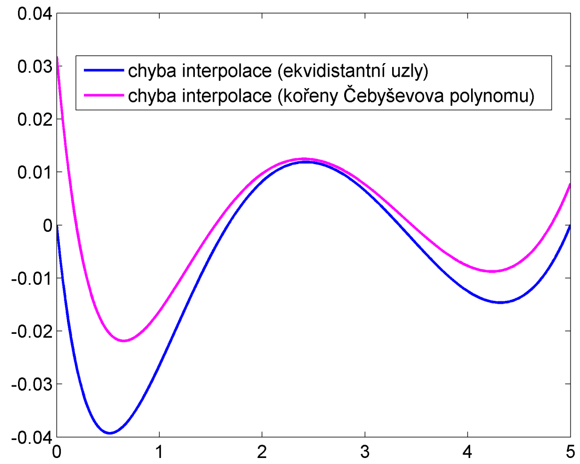


2) $n = 3$

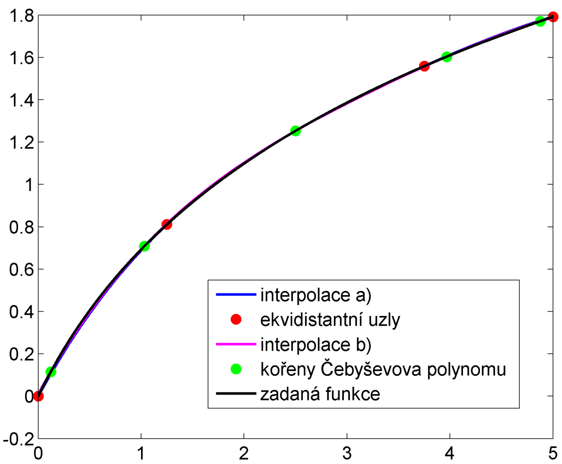


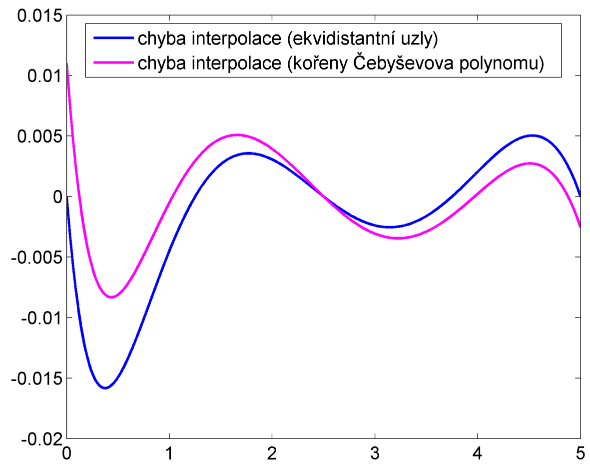
3) $n = 4$


interpolace a)
ekvidistantní uzly
interpolace b)
kořeny Čebyševova polynomu
zadaná funkce

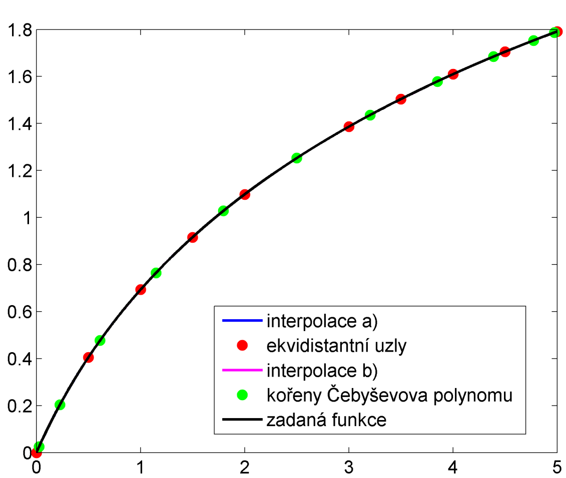

chyba interpolace (ekvidistantní uzly)
chyba interpolace (kořeny Čebyševova polynomu)

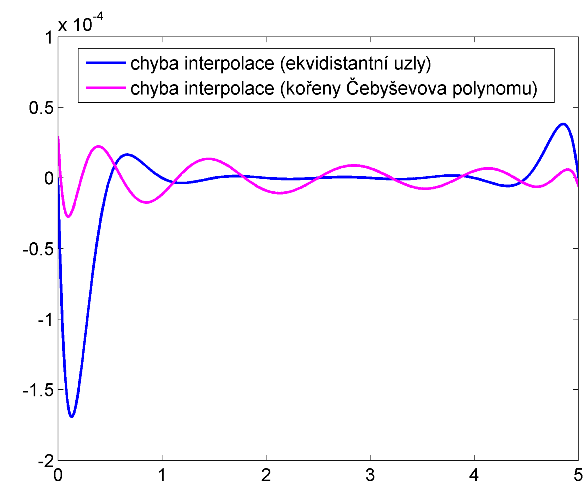
4) $n = 5$


interpolace a)
ekvidistantní uzly
interpolace b)
kořeny Čebyševova polynomu
zadaná funkce

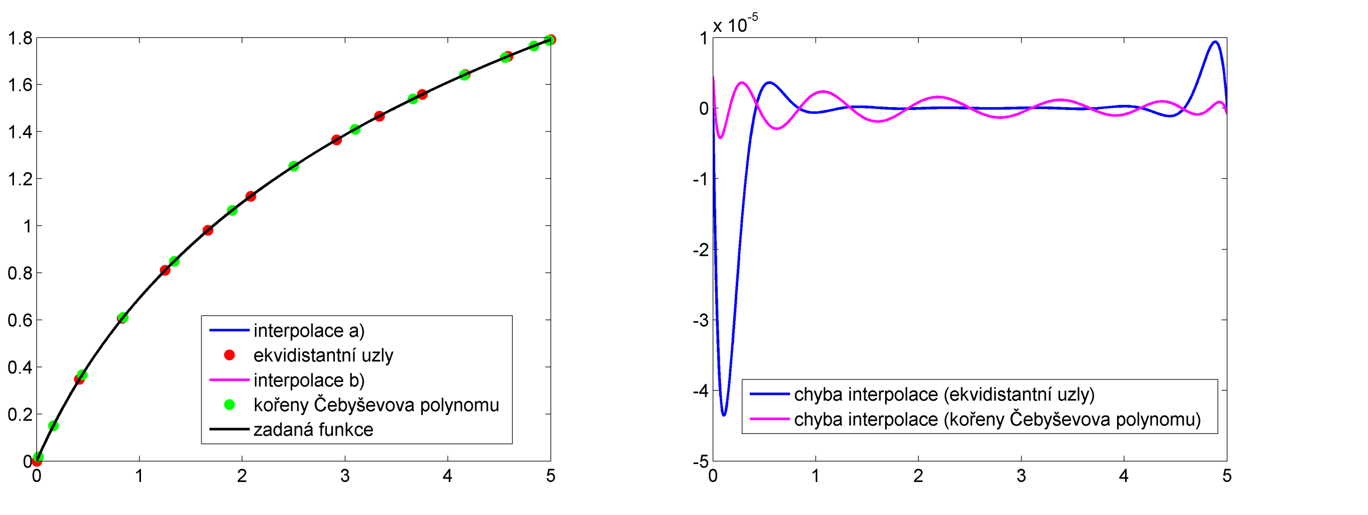

chyba interpolace (ekvidistantní uzly)
chyba interpolace (kořeny Čebyševova polynomu)

5) $n = 11$

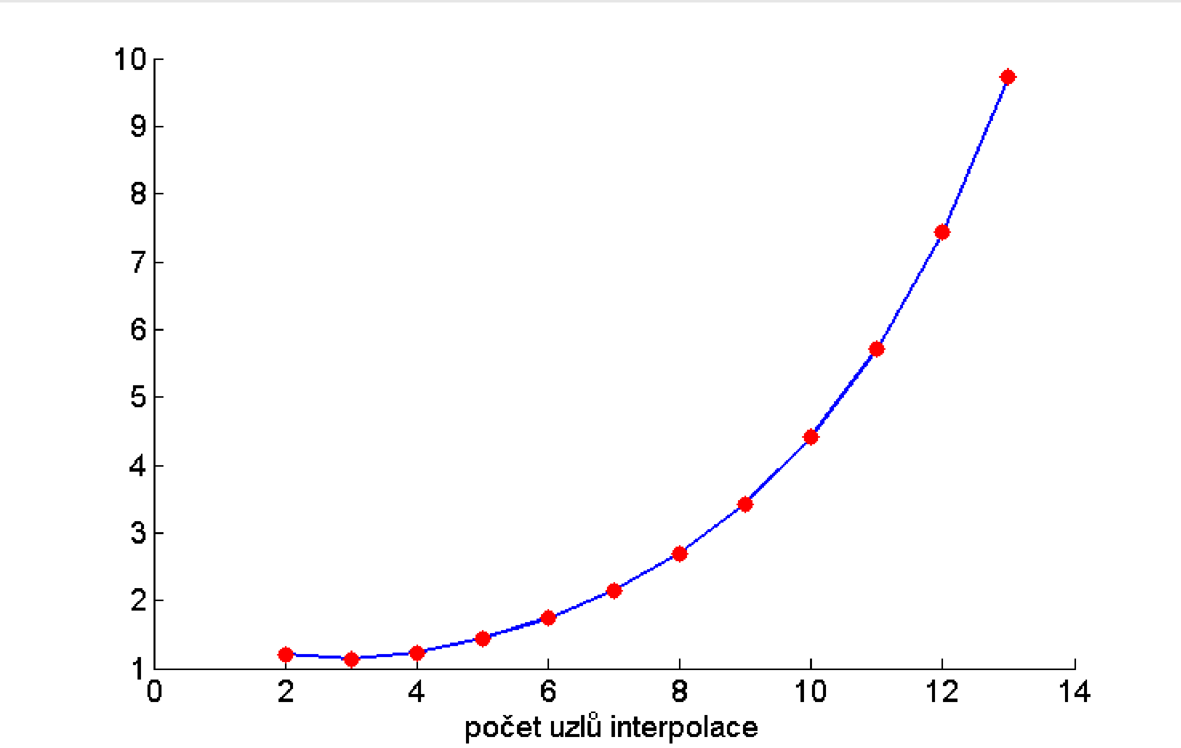

interpolace a)
ekvidistantní uzly
interpolace b)
kořeny Čebyševova polynomu
zadaná funkce


x 10^-4
chyba interpolace (ekvidistantní uzly)
chyba interpolace (kořeny Čebyševova polynomu)

6) $n = 13$

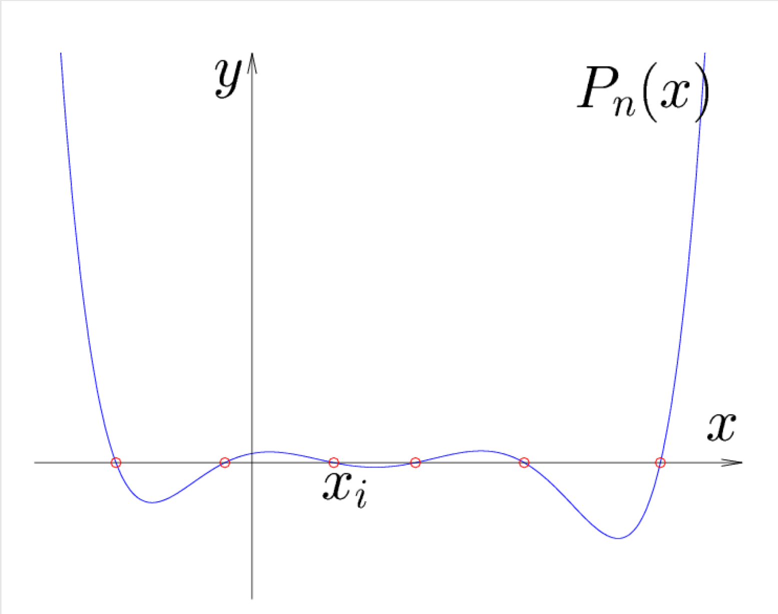


Následující obrázek ukazuje poměr maximové normy chyby interpolačního polynomu vypočteného pro hodnoty v ekvidistantních uzlech ku maximové normě chyby interpolačního polynomu vypočteného pro hodnoty v uzlech určených jakožto kořeny Čebyševových polynomů.



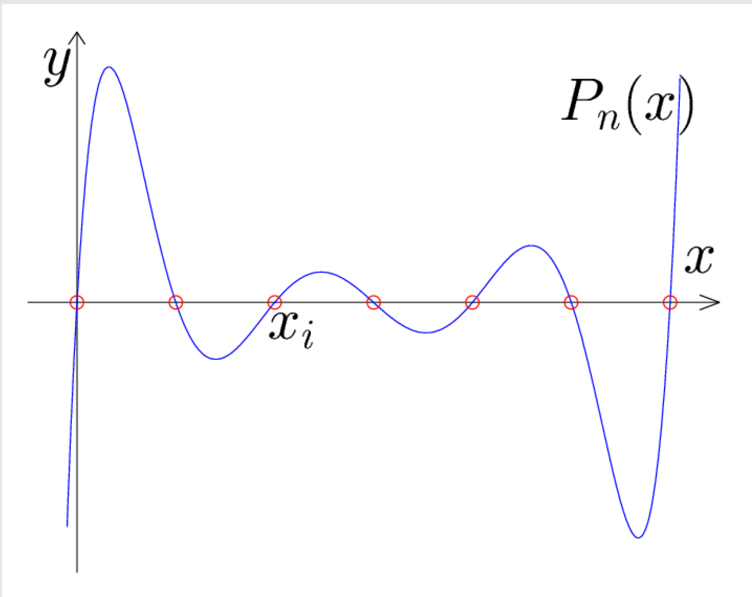
Poznámka:

Interpolační polynomy vyšších řádů není vhodné užívat pro aproximaci hodnot funkce mimo interval obsahující uzly interpolace (tzv. extrapolaci), protože absolutní hodnota polynomu nabývá velkých hodnot.



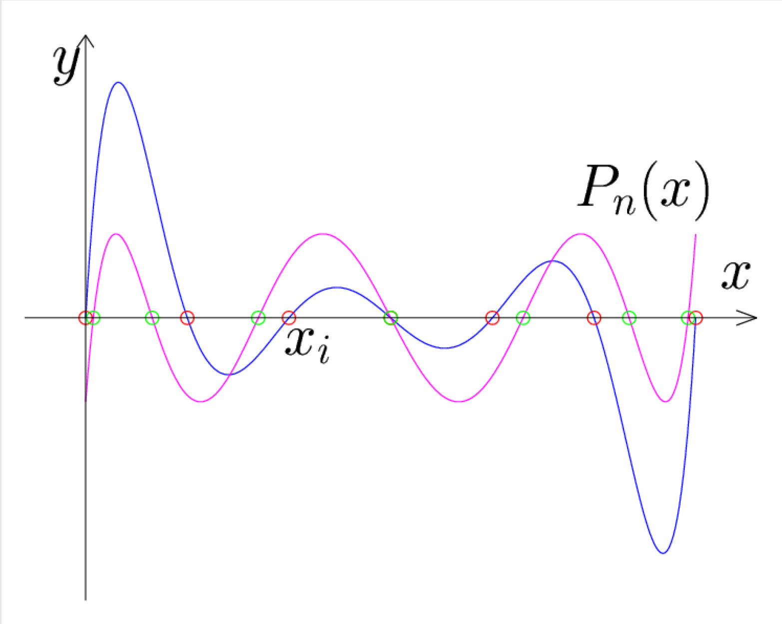
Poznámka:

Není obecně vhodné interpolovat polynomem funkci, která je dána velkým počtem svých hodnot. Stupeň interpolačního polynomu by potom byl velký.



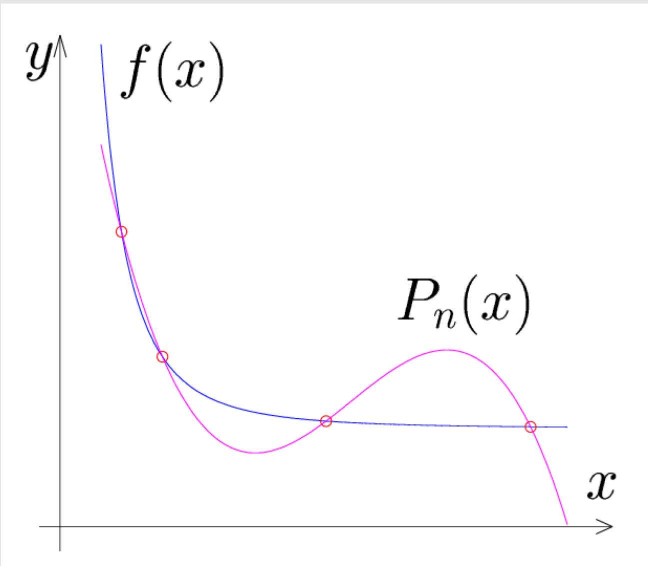
Poznámka:

Použijeme-li vhodně zvolené neekvidistantní uzly, můžeme amplitudy chyby minimalizovat. (Vhodnou volbou jsou uzly zvolené jako kořeny tzv. Čebyševových polynomů.)



Poznámka:

Interpolace polynomem není obecně vhodná např. pro funkce, které mají asymptotu.



Pokud chceme aproximovat funkci, která má asymptotu, je vhodné místo lineární aproximace (polynomiální) použít nelineární aproximaci

$$\boxed{R_{M,N}(x) = \frac{P_M(x)}{Q_N(x)} = \frac{p_0 + p_1 x + \cdots + p_M x^M}{1 + q_1 x + \cdots + q_N x^N}}$$

Poznámka:

Pokud máme další informace např. o derivaci dané funkce v uzlových bodech, můžeme použít tzv. **Hermitovu interpolaci**.

Poznámka:

Všimněme si, že v konstrukci interpolačního polynomu nezáleží na pořadí zadaných tabulkových bodů.

V řadě případů potřebujeme kromě a_i vypočítat hodnotu polynomu v daném bodě α, tj.

$$N_n(\alpha) = a_0 + a_1(\alpha - x_0) + a_2(\alpha - x_0)(\alpha - x_1) + \cdots + a_n(\alpha - x_0)(\alpha - x_1)\ldots(\alpha - x_{n-1}).$$

Při vhodném uzávorkování můžeme výpočet zefektivnit (zmenšíme počet operací sčítání a násobení):

$$N_n(\alpha) = a_0 + (\alpha - x_0)\Big[a_1 + (\alpha - x_1)\Big[a_2 + (\alpha - x_2)[a_3 + \ldots [a_n]]\Big]\Big].$$

Tento postup můžeme samozřejmě použít jen tehdy, když už známe koeficienty a_i.

Chceme-li vypočítat pouze hodnotu polynomu $N_n(\alpha)$ v bodě α za co nejmenšího počtu operací a nepotřebujeme-li koeficienty a_i, použijeme tzv. **Nevilleův algoritmus**.

Princip je podobný jako v algoritmu pro určení koeficientů Newtonova polynomu.

Nevilleův algoritmus

1. $P_{i,0} = f(x_i), \qquad i = 0, 1, \ldots, n$
2. $P_{i,k} = P_{i,k-1} + (\alpha - x_i)\,\dfrac{P_{i,k-1} - P_{i-1,k-1}}{x_i - x_{i-k}};$
3. $N_n(\alpha) = P_{nn}$

Princip Nevilleova algoritmu je ukázán v následujícím příkladu.

Příklad Vypočtěte $f(3.5)$, kde funkce $f(x)$ je dána tabulkou:

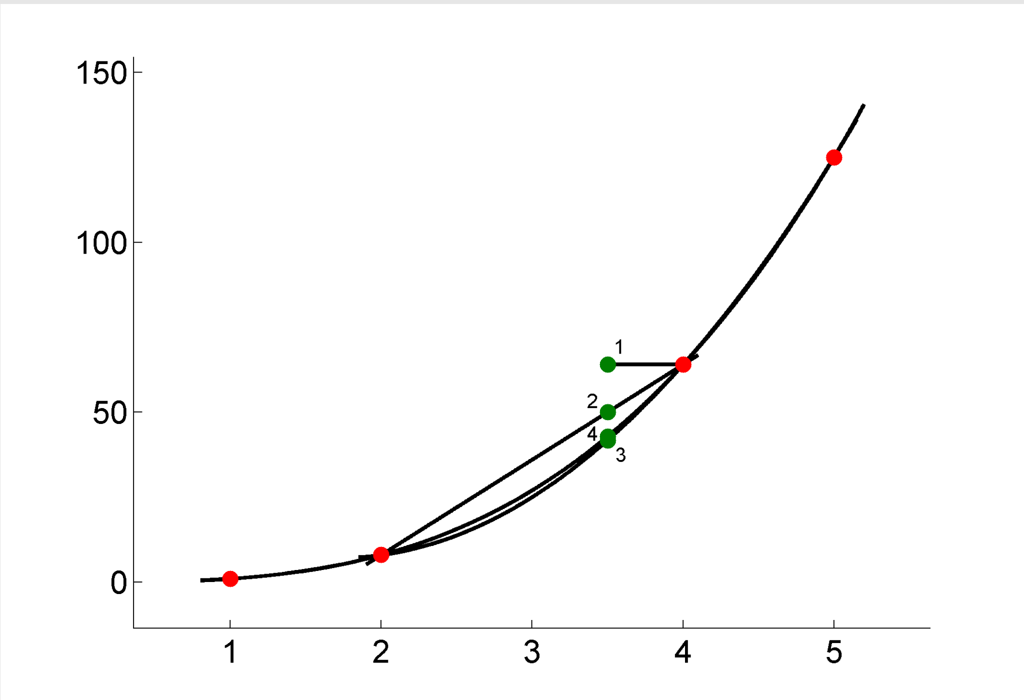
x_i	1	2	4	5
$f(x_i)$	1	8	64	125

Řešení:

Uzly x_i je výhodné uspořádat podle rostoucí vzdálenosti od bodu α, v němž chceme stanovit přibližnou hodnotu funce $f(x)$. Podle rozdílu hodnot $P_{i,i}$ a $P_{i-1,i-1}$ $(i = 1, \ldots, n)$ lze rozhodnout o předčasném ukončení Nevilleova algoritmu, popř. o vhodnosti interpolace pomocí $N_n(x)$.

$\alpha - x_i$	x_i	$f(x_i)$			
-0,5	4	**64**			
1,5	2	8	8+1,5 $\dfrac{8-64}{2-4}$ $= $ **50**		
-1,5	5	125	125–1,5 $\dfrac{125-8}{5-2}$ $=$ 66,5	66,5–1,5 $\dfrac{66,5-50}{5-4}$ $=$ **41,75**	
2,5	1	1	1+2,5 $\dfrac{1-125}{1-5}$ $=$ 78,5	78,5+2,5 $\dfrac{78,5-66,5}{1-2}$ $=$ 48,5	48,5+2,5 $\dfrac{48,5-78,5}{1-4}$ $=$ **42,875**

Z následujících obrázků je patrný význam hodnot, které dostáváme na diagonále.



Poznámka:

Vyjděme z předpokladu, že máme přibližně interpolovat hodnotu tabulkou dané funkce v libovolném bodě. Pokud nutně netrváme na tom, že v tomto bodě chceme přesně určit hodnotu interpolačního polynomu procházejícími všemi tabulkovými body, přistupujeme k Nevilleovu algoritmu iteračně, tj. pokud bude rozdíl po sobě jdoucích diagonálních prvků dostatečně malý, ukončíme výpočet.

V tomto případě je ovšem rozumné seřadit uzlové body podle rostoucí vzdálenosti od zadaného bodu, ve kterém interpolujeme hodnotu funkce.

V následujích příkladech jsou demonstrovány výsledky pro stejné zadání, ovšem při použití různých seřazení uzlových bodů:

1) od nejbližšího po nejvzdálenější,

2) od nejvzdálenějšího po nejbližší.

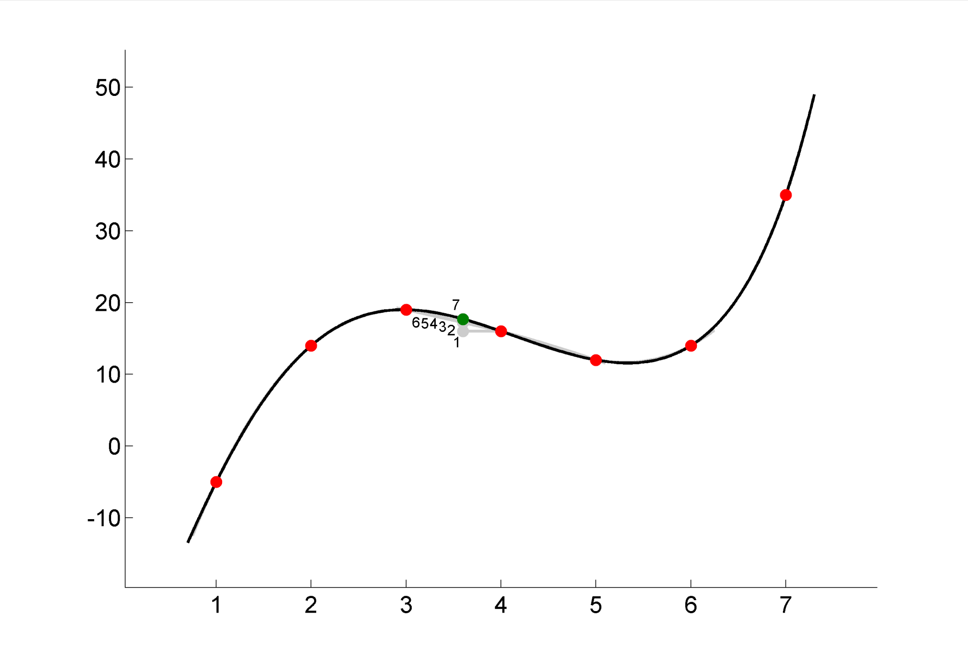
Příklad 1 Interpolujte hodnotu zadané funkce f v bodě $\alpha = 3{,}6$.

x_i	1	2	3	4	5	6	7
$f(x_i)$	-5	14	19	16	12	14	35

Výsledky Nevilleova algoritmu, když uvažujeme uzly seřazené podle vzdálenosti od α **vzestupně**:

výsledky získané v MATLABu

```
   x(k)    |   f(k)    |   Aproximace f(alfa)
=================================================
   4.0000 |  16.0000
   3.0000 |  19.0000 |  17.2000
   5.0000 |  12.0000 |  16.9000 |  17.3200
   2.0000 |  14.0000 |  12.9333 |  19.2800 |  17.7120
   6.0000 |  14.0000 |  14.0000 |  11.4400 |  17.7120 |  17.7120
   1.0000 |  -5.0000 |   4.8800 |  28.5920 |  17.4432 |  17.7926 |  17.7228
   7.0000 |  35.0000 |  12.3333 | -13.0080 |  15.2800 |  18.9574 |  17.9674 |  17.6901
```



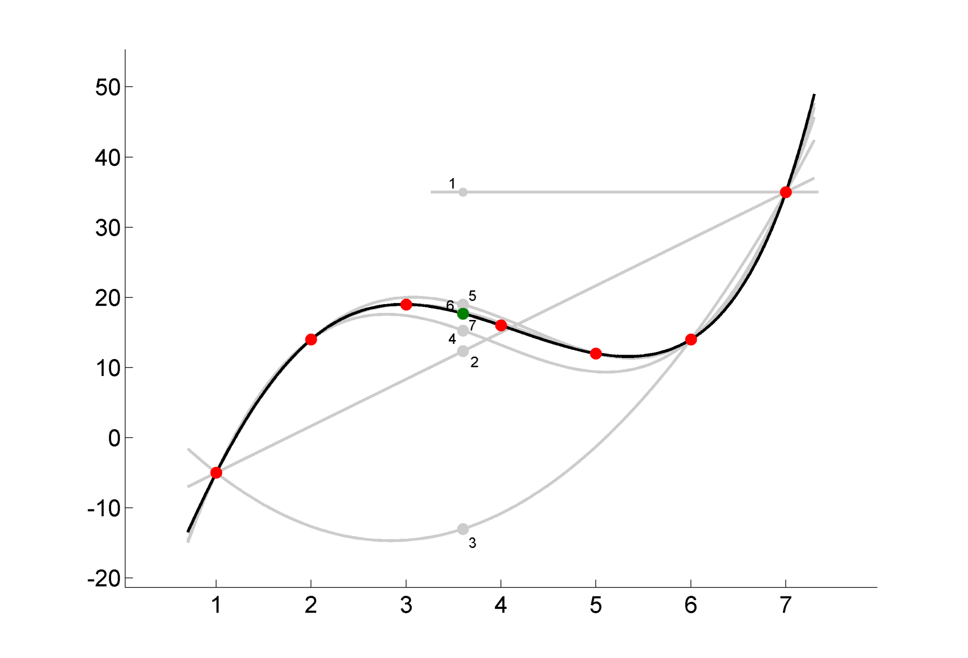
Příklad 2 Interpolujte hodnotu zadané funkce f v bodě $\alpha = 3{,}6$.

x_i	1	2	3	4	5	6	7
$f(x_i)$	-5	14	19	16	12	14	35

Výsledky Nevilleova algoritmu, když uvažujeme uzly seřazené podle vzdálenosti od α **sestupně**:

výsledky získané v MATLABu

```
   x(k)   |   f(k)   |   Aproximace f(alfa)
================================================
   7.0000 |  35.0000
   1.0000 |  -5.0000 |  12.3333
   6.0000 |  14.0000 |   4.8800 | -13.0080
   2.0000 |  14.0000 |  14.0000 |  28.5920 |  15.2800
   5.0000 |  12.0000 |  12.9333 |  11.4400 |  17.4432 |  18.9574
   3.0000 |  19.0000 |  16.9000 |  19.2800 |  17.7120 |  17.7926 |  17.9674
   4.0000 |  16.0000 |  17.2000 |  17.3200 |  17.7120 |  17.7120 |  17.7228 |  17.6901
```

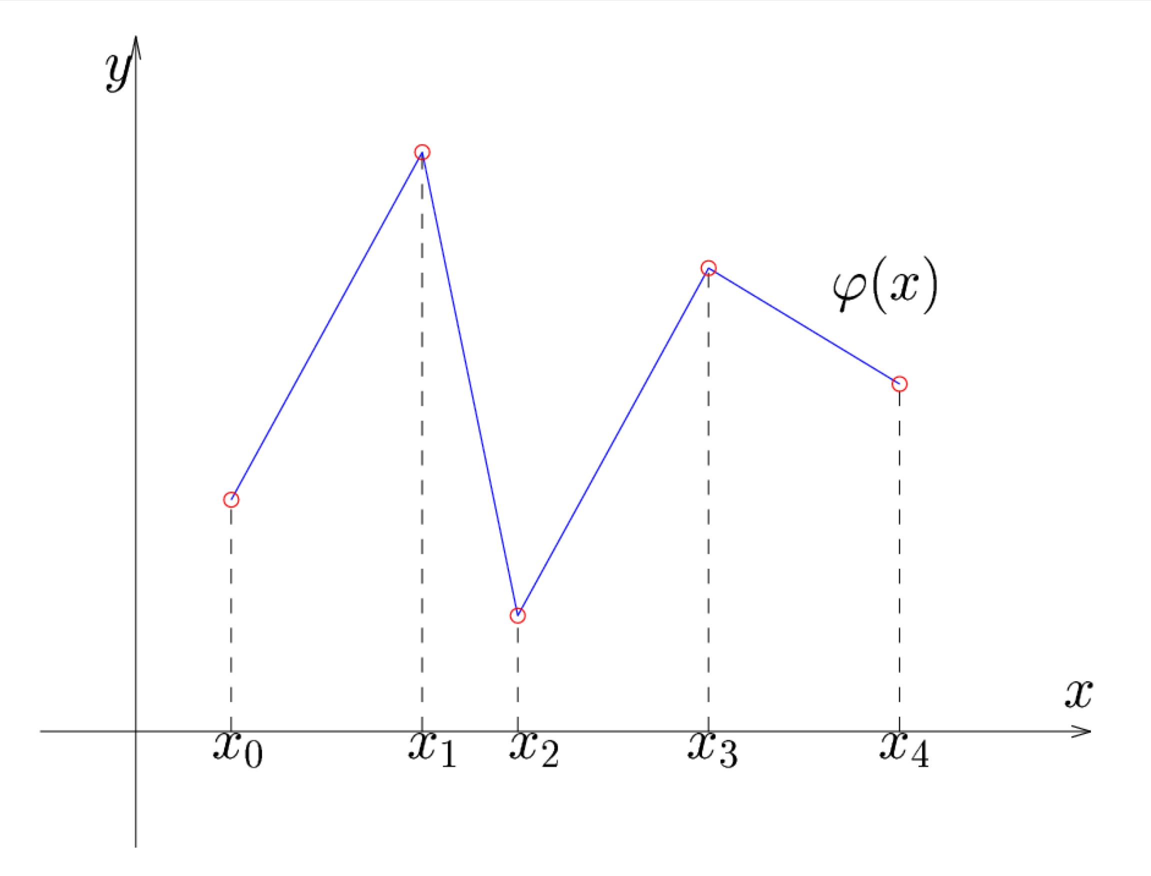


Poznámka

Jak se dá interpretovat libovolná (tj. i nediagonální) hodnota v trojúhelníkové matici, kterou získáváme Nevilleovým algoritmem?

Interpolace spline funkcemi

Nejjednodušší spline funkcí je tzv. **lineární spline funkce**; jde vlastně o lomenou čáru spojující zadané interpolované body.



Máme dánu funkci f tabulkou hodnot $\{x_i, f_i\}$ $i = 0, 1, \ldots, n$.

Funkci $s(x)$ definovanou na intervalu $\langle x_0, x_n\rangle$ nazýváme **lineární spline interpolací** funkce $f(x)$, má-li následující vlastnosti:

(i) na každém intervalu $\langle x_i, x_{i+1}\rangle$, $i = 0, 1, \ldots, n-1$ je polynom prvního stupně, tj.

$$s(x) = s_i(x),\ x \in \langle x_i, x_{i+1}\rangle,\ \text{kde } \boxed{s_i(x) = a_i + b_i(x - x_i)}$$

(ii) splňuje interpolační podmínky $s(x_i) = f(x_i)$, tj.

$$\boxed{\begin{aligned} &s_i(x_i) = f_i, \quad i = 0, 1, \ldots, n-1 \\ &s_{n-1}(x_n) = f_n \end{aligned}}$$

(iii) je spojitá na $\langle x_0, x_n\rangle$, tj. i v uzlech x_i

$$\boxed{s_i(x_{i+1}) = s_{i+1}(x_{i+1}), \quad i = 0, 1, \ldots, n-2}$$

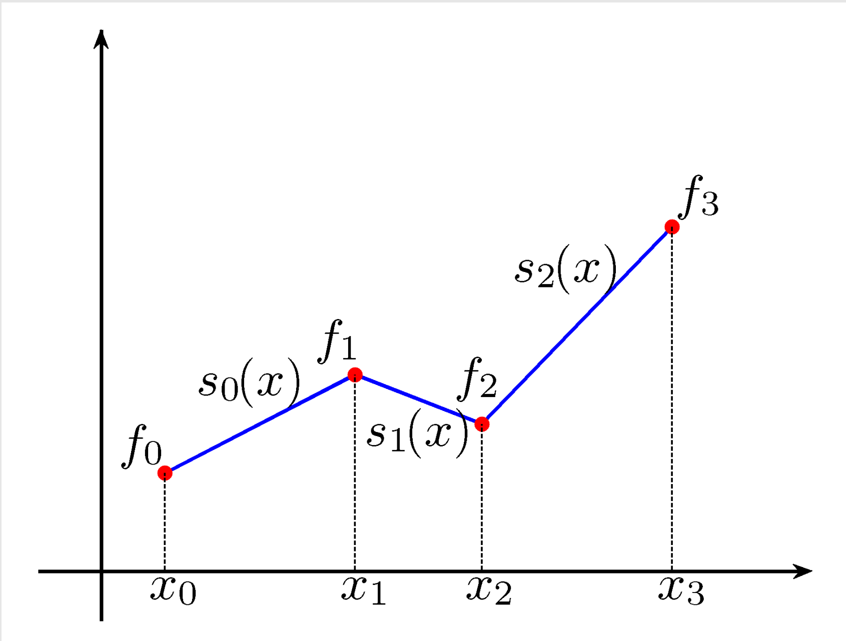
Poznámka: Těmito požadavky je funkce $s(x)$ určena jednoznačně.

(i) ... hledáme $2n$ koeficientů a_i a b_i

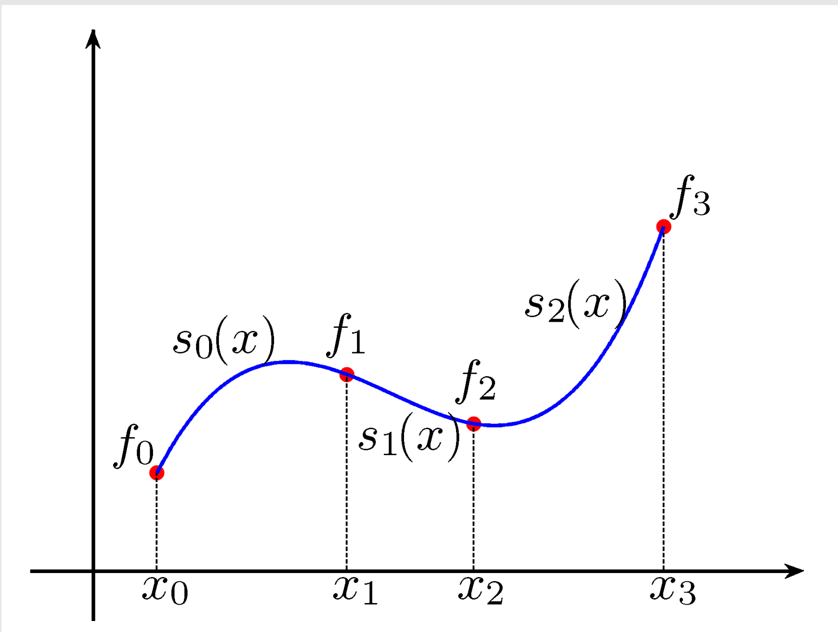
(ii) představuje $(n+1)$ podmínek

(iii) představuje $(n-1)$ podmínek

Platí: $\boxed{s_i(x) = f_i + \frac{f_{i+1} - f_i}{h_i}(x - x_i), \quad h_i = x_{i+1} - x_i, \quad i = 0, 1, \ldots, n-1}$



Pokud bychom chtěli, aby byla aproximace spline funkcí hladká, musíme použít polynomy vyššího stupně než 1. Nejvíce používanou je tzv. **kubická spline interpolace**, která používá polynomy 3 stupně.



Poznamenejme zde, že ostatní volby stupně polynomů nepřináší lepší výsledky a výpočty jsou v případě vyšších stupňů složitější.

Kubická spline interpolace

Funkce f je dána tabulkou $\{x_i, f_i\}$, $i = 0, 1, \ldots, n$

Funkci $s(x)$ definovanou na intervalu $\langle x_0, x_n\rangle$ nazýváme **kubickou spline interpolací** funkce f, má-li následující vlastnosti:

(i) je na každém intervalu $\langle x_i, x_{i+1}\rangle$, $i = 0, 1, \ldots, n-1$ polynomem 3. stupně ve tvaru

$$\boxed{s_i(x) = a_i + b_i(x - x_i) + \frac{c_i}{2}(x - x_i)^2 + \frac{d_i}{6}(x - x_i)^3}$$

(ii) splňuje interpolační podmínky $s(x_i) = f(x_i)$, tj.

$$\boxed{\begin{aligned} &s_i(x_i) = f(x_i), \quad i = 0, 1, \ldots, n-1 \\ &s_{n-1}(x_n) = f_n \end{aligned}}$$

(iii) je spojitá na $\langle x_0, x_n\rangle$, tj. v uzlech x_i platí

$$\boxed{s_i(x_{i+1}) = s_{i+1}(x_{i+1}), \quad i = 0, 1, \ldots, n-2}$$

(iv) má spojitou první derivaci na $\langle x_0, x_n\rangle$

$$\boxed{s_i'(x_{i+1}) = s_{i+1}'(x_{i+1}), \quad i = 0, 1, \ldots, n-2}$$

(v) má spojitou druhou derivaci na $\langle x_0, x_n\rangle$

$$\boxed{s_i''(x_{i+1}) = s_{i+1}''(x_{i+1}), \quad i = 0, 1, \ldots, n-2}$$

Funkce $s(x)$ není podmínkami (ii) – (v) určena jednoznačně:

(ii)	...	$(n+1)$ podmínek
(iii)	...	$(n-1)$ podmínek
(iv)	...	$(n-1)$ podmínek
(v)	...	$(n-1)$ podmínek
celkem	...	$4n-2$ podmínek; (počet koeficientů je ale $4n$)

2 podmínky je nutno dodat:

(A) přirozené podmínky

$$\boxed{s''(x_0) = s''(x_n) = 0}$$

(B) podmínky periodicity (s periodou $T = x_n - x_0$)

$$s(x_0) = s(x_n) \quad \ldots \text{ splněna automaticky } (f_0 = f_n)$$

$$\boxed{s'(x_0) = s'(x_n), \quad s''(x_0) = s''(x_n)}$$

(C) podmínky tečen

$$\boxed{s'(x_0) = y_0', \quad s'(x_n) = y_n'}, \qquad \text{kde } y_0', y_n' \text{ jsou daná čísla}$$

(D) viz MATLAB

Konstrukce kubické spline funkce

Funkci $s(x)$ lze psát ve tvaru

$$\boxed{s(x) = \eta_0 s_0(x) + \eta_1 s_1(x) + \cdots + \eta_{n-1} s_{n-1}(x)}$$

kde $\eta_i = \eta_i(x)$ jsou charakteristické funkce intervalu, tj.

$$\eta_i = 1 \text{ na } \langle x_i, x_{i+1}), \qquad \eta_{n-1} = 1 \text{ na } \langle x_{n-1}, x_n\rangle$$
$$\eta_i = 0 \text{ jinde}$$

Snadno lze odvodit:

$$\boxed{s(x_i) = a_i, \quad s'(x_i) = b_i, \quad s''(x_i) = c_i, \quad s'''(x_i+) = d_i, \quad s'''(x_i-) = d_{i-1}}$$

Pomocí těchto vztahů přepíšeme podmínky (ii) až (v) (označíme $h_i = x_{i+1} - x_i, \quad i = 0, 1, \ldots, n-1$)

(ii) interpolační podmínky

$$f_i = a_i, \quad i = 0, 1, \ldots, n-1$$

$$f_n = a_{n-1} + b_{n-1}h_{n-1} + \frac{c_{n-1}}{2}h_{n-1}^2 + \frac{d_{n-1}}{6}h_{n-1}^3$$

(iii) spojitost

$$f_{i+1} = a_i + b_i h_i + \frac{c_i}{2}h_i^2 + \frac{d_i}{6}h_i^3, \quad i = 0, 1, \ldots, n-2$$

$$\left[s_{i+1}(x_{i+1}) = s_i(x_{i+1})\right]$$

(iv) spojitost derivace

$$b_i + c_i h_i + \frac{d_i}{2}h_i^2 = b_{i+1}, \quad i = 0, 1, \ldots, n-2$$

$$\left[s_i'(x_{i+1}) = s_{i+1}'(x_{i+1})\right]$$

(v) spojitost 2. derivace

$$c_i + d_i h_i = c_{i+1}, \quad = 0, 1, \dots, n-2$$
$$\left[s_i''(x_{i+1}) = s_{i+1}''(x_{i+1}) \right]$$

Z těchto podmínek lze sestavit soustavu lineárních algebraických rovnic pro neznámé $c_0, \dots, c_n$ a jejich prostřednictvím vypočítat $b_0, \dots, b_n$ a $d_0, \dots, d_n$. Po úpravách dostaneme:

$$\boxed{\alpha_k c_{k-1} + 2c_k + \beta_k c_{k+1} = g_k, \quad k = 1, 2, \dots, n-1}$$

kde

$$\boxed{\begin{aligned} \beta_k &= \frac{h_k}{h_k + h_{k-1}} \\ \alpha_k &= 1 - \beta_k \\ g_k &= \frac{6}{h_k + h_{k-1}} \left[\frac{f_{k+1} - f_k}{h_k} - \frac{f_k - f_{k-1}}{h_{k-1}} \right] \end{aligned}}$$

Vztahy představují soustavu $n-1$ rovnic pro $n+1$ neznámých. Pro jednoznačnost je třeba přidat jednu z podmínek $(A), (B), (C)$. Soustava bude mít tvar:

$$\begin{bmatrix} 2 & \beta_0 & & & & & \\ \alpha_1 & 2 & \beta_1 & & & & \\ & \alpha_2 & 2 & \beta_2 & & & \\ & & & \ddots & & & \\ & & & & \alpha_{n-1} & 2 & \beta_{n-1} \\ & & & & & \alpha_n & 2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \\ \\ \\ c_n \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ \\ \\ \\ g_n \end{bmatrix} \qquad (*)$$

$(*)$ ... první a poslední řádek soustavy představuje podmínka (A) nebo (B),
... použití podmínky (C) znamená vyškrtnout 1. a poslední řádek a sloupec

Poznámka:

Matice soustavy je třídiagonální a ostře diagonálně dominantní $(\alpha_k + \beta_k = 1)$. $\Rightarrow$ je regulární $\Rightarrow \exists!$ řešení.

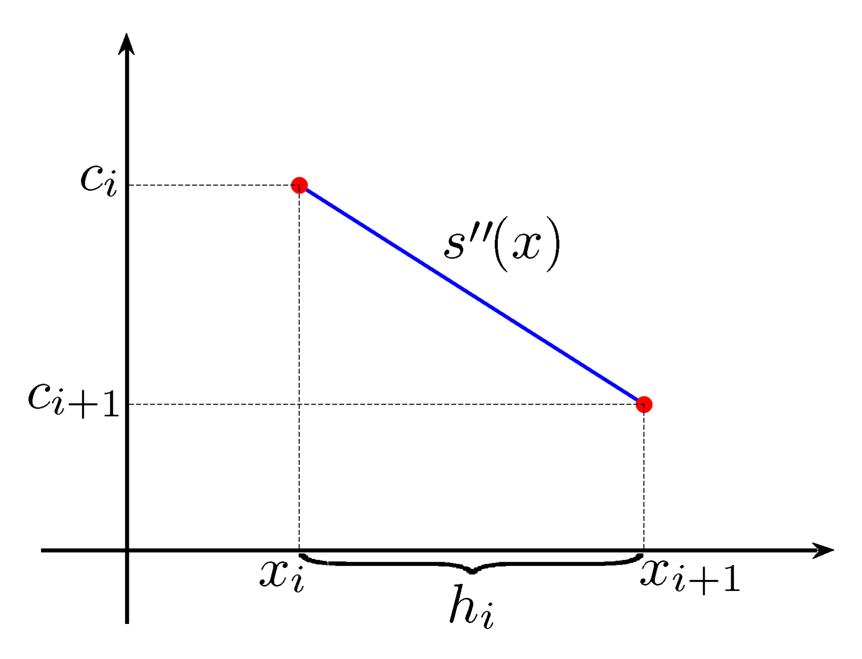
Soustavu řešíme GEM pro třídiagonální matice.

Odvození třídiagonální matice pro výpočet koeficientů c_i

Druhá derivace $\boxed{s_i''(x) = c_i + d_i(x - x_i)}$ je lineární funkcí, tj. známe-li $s''(x_i) = c_i$, pak na $\langle x_i, x_{i+1} \rangle$ můžeme psát

$$\boxed{s_i''(x) = c_i \frac{x_{i+1} - x}{h_i} + c_{i+1} \frac{x - x_i}{h_i}}. \qquad \circledast$$

(Ověření: $s_i''(x_i) = c_i$, $s_i''(x_{i+1}) = c_{i+1}$, $s_i''(x)$ je lineární.)



Integrací ⊛ dostaneme

$$\boxed{s_i'(x) = -c_i \frac{(x_{i+1}-x)^2}{2h_i} + c_{i+1}\frac{(x-x_i)^2}{2h_i} + A_i} \qquad ⊛⊛$$

$$s_i(x) = c_i \frac{(x_{i+1}-x)^3}{6h_i} + c_{i+1}\frac{(x-x_i)^3}{6h_i} + A_i(x-x_i) + B_i$$

Z interpolačních podmínek plyne ($s(x_i) = f_i$):

$$\boxed{B_i = f_i - \frac{c_i h_i^2}{6}}$$

Pro $x = x_{i+1}$

$$f_{i+1} = c_{i+1}\frac{h_i^2}{6} + A_i h_i + \underbrace{f_i - \frac{c_i h_i^2}{6}}_{B_i}$$

$$\Rightarrow \quad \boxed{A_i = \frac{f_{i+1}-f_i}{h_i} - \frac{(c_{i+1}-c_i)h_i}{6}} \qquad ⊛⊛⊛$$

Ze spojitosti první derivace $\boxed{s_i'(x_{i+1}^-) = s_{i+1}'(x_{i+1}^+),\ i = 0,1,\dots,n-2}$ s užitím ⊛⊛ a ⊛⊛⊛ plyne

$$\overbrace{c_{i+1}\frac{h_i}{2} + \underbrace{\frac{f_{i+1}-f_i}{h_i} - \frac{(c_{i+1}-c_i)h_i}{6}}_{A_i}}^{s_i'(x_{i+1})} = \underbrace{-c_{i+1}\frac{h_{i+1}^2}{2h_{i+1}} + \underbrace{\frac{f_{i+2}-f_{i+1}}{h_{i+1}} - \frac{(c_{i+2}-c_{i+1})h_{i+1}}{6}}_{A_{i+1}}}_{s_{i+1}'(x_{i+1})}$$

$$c_i \frac{h_i}{6} + c_{i+1}\Big(\underbrace{\frac{h_i}{2} - \frac{h_i}{6}}_{=\frac{h_i}{3}} + \underbrace{\frac{h_{i+1}}{2} - \frac{h_{i+1}}{6}}_{=\frac{h_{i+1}}{3}}\Big) + c_{i+2}\frac{h_{i+1}}{6} = \frac{f_{i+2}-f_{i+1}}{h_{i+1}} - \frac{f_{i+1}-f_i}{h_i}$$

Vynásobíme $\dfrac{6}{h_i + h_{i+1}}$ a dostaneme

$$\boxed{\underbrace{\frac{h_i}{h_i + h_{i+1}}}_{=\alpha_i} c_i \; + \; 2\,c_{i+1} \; + \; \underbrace{\frac{h_{i+1}}{h_i + h_{i+1}}}_{=\beta_i} c_{i+2} = \underbrace{\frac{6}{h_i + h_{i+1}}\left(\frac{f_{i+2}-f_{i+1}}{h_{i+1}} - \frac{f_{i+1}-f_i}{h_i}\right)}_{=g_i}}$$

Minimální vlastnost a odhad chyby

Označme $S_1(\langle a, b\rangle)$ množinu funkcí f, které splňují podmínky (ii) až (v) a podmínku (A) a jsou navíc na $\langle a, b\rangle$ integrovatelné s kvadrátem. Mezi všemi funkcemi $f \in S_1(\langle a, b\rangle)$ právě **přirozený kubický spline** udílí nejmenší hodnotu integrálu

$$\boxed{J(f) = \int_a^b \left|f''(x)\right|^2 \, dx}.$$

$J(f)$ ... míra celkové křivosti křivky $y = f(x)$.

Věta

Nechť funkce f má spojité derivace až do řádu 4 a má omezenou 4. derivaci pro $x \in \langle a, b\rangle$. Nechť dále platí:

$$\frac{h}{h_i} \le K, \quad i = 0, 1, \dots, n-1, \quad h = \max_i |x_{i+1} - x_i|$$

Když $s(x)$ je spline interpolace funkce f v bodech x_i a splňuje podmínky $s'(a) = f'(a), s'(b) = f'(b)$, potom pro $x \in \langle a, b\rangle$ platí:

$$|f(x) - s(x)| \le c_1 K h^4$$
$$|f'(x) - s'(x)| \le c_2 K h^3$$
$$|f''(x) - s''(x)| \le c_3 K h^2.$$

Příklad

Určete kubický spline pro funkci zadanou tabulkou

x_i	1	2	3	4	5	6	7
$f(x_i)$	3	8	1	7	2	4	3

Použijte různé dodatečné podmínky. Pro spline s podmínkami tečen použijte $f'(1) = 0$ a $f'(7) = 0$.