APROXIMACE FUNKCÍ

Jedním ze základních úkolů numerických metod matematické analýzy je studium aproximací funkcí. Při numerickém řešení úloh matematické analýzy totiž 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.

Poznamenejme zde, že jsme již aproximaci funkce 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, $\ln x$, ...) 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.

Typickým příkladem užití aproximace funkcí jsou numerické metody pro výpočet určitého integrálu. Další oblastí numerické matematiky založenou na užití aproximací je zpracování výsledků měření. Hledáme zde zpravidla jednoduchý analytický výraz vyjadřující přibližnou funkční závislost pro tabulkou zadané hodnoty.

Při výběru vhodné aproximace postupujeme tak, že předem zvolíme tvar aproximující funkce, ve které vystupují nějaké proměnné parametry, a hodnoty těchto parametrů se pak snažíme určit tak, aby získaná aproximace vyhovovala našim požadavkům.

Základní otázkou zůstává, v jakém tvaru budeme hledat aproximaci φ funkce f. Jednou z možností je aproximaci volit např. ve tvaru polynomu $\varphi(x) = a_3x^3 + a_2x^2 + a_1x + a_0$.

Obecně tedy nejprve určíme systém jednoduchých **základních (bázových) funkcí** (ne nutně polynomů) φ_0, φ_1, φ_2, ..., φ_n a funkci f aproximujeme lineární kombinací základních funkcí, tj.

$$\varphi(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \ldots + c_n\varphi_n(x). \tag{1}$$

Otázka výběru aproximace se tedy převede na určení hodnot parametrů c_0, c_1, ..., c_n podle nějakého kritéria vhodného pro konkrétní úlohu.

Poznámka: Velmi často budeme za základní funkce volit funkce 1, x, x^2, ..., x^n, tj. aproximaci φ budeme hledat ve třídě polynomů nejvýše n-tého stupně.

Nyní se zamyslíme nad kritérii, podle kterých budeme určovat parametry z lineární kombinace (1). Ty samozřejmě vyplývají z charakteru konkrétní úlohy a zhruba je můžeme rozdělit do tří skupin

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.

Aproximace na okolí bodu

Nyní se budeme věnovat jednotlivým úlohám. Začneme aproximací na okolí bodu, tj. mluvíme o **aproximaci Taylorovým polynomem**.

Předpokládáme, že daná funkce f má v daném bodě x_0 aspoň n derivací, a že hodnoty těchto derivací v bodě x_0 známe. Podmínky pro funkci, která co nejlépe napodobuje chování funkce f matematicky zapíšeme takto:

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

tj. hodnoty derivací funkcí f a φ v bodě x_0 jsou stejné až do řádu n. Tuto podmínku samozžejmě splňuje Taylorův polynom

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

Pro chybu aproximace Taylorovým polynomem platí

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

a umíme-li odhadnout $n + 1$ derivaci funkce f na daném okolí bodu x_0, můžeme provést následující odhad chyby aproximace.

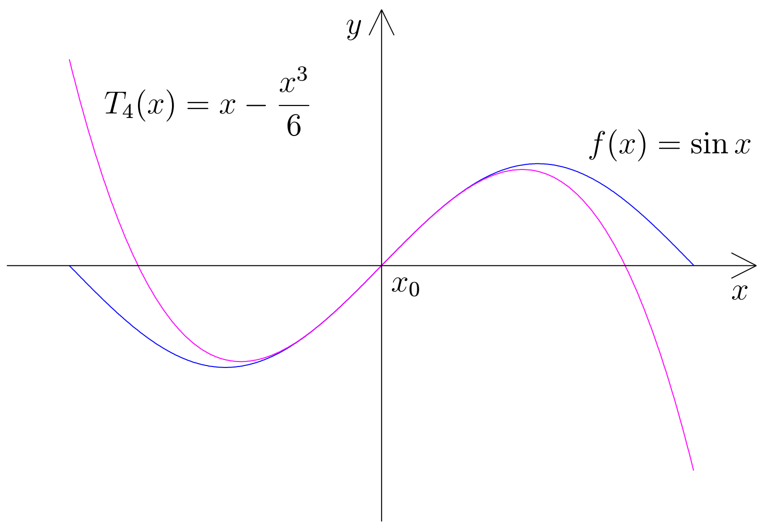
$$\text{Platí-li } |f^{(n+1)}(x)| \le M \quad \forall x \in \mathcal{U}(x_0), \quad \text{potom } |e(x)| \le \frac{M}{(n+1)!}|x - x_0|^{n+1}.$$

Příklad: Stanovte Taylorův polynom 4.stupně, který aproximuje funkci $f(x) = \sin x$ v bodě $x_0 = 0$.

Řešení:

$$T_4(x) = \sin 0 + x.\cos 0 - \frac{x^2}{2}\sin 0 - \frac{x^3}{6}\cos 0 + \frac{x^4}{24}\sin 0 \;=\; x - \frac{x^3}{6}.$$

Vidíme, že v Taylorově polynomu 4.stupně vypadly členy s x^2 a x^4, protože obsahovaly $\sin 0$. Takže v našem případě platí, že $T_4(x) = T_3(x)$.



Odpovězme na otázku, pro jaké x lze psát $\sin x \approx x - \frac{x^3}{6}$, aby chyba nebyla větší než 10^{-6}?

Výraz pro chybu má tvar

$$e(x) = \frac{x^5}{5!}(-\cos\xi)$$

Dále víme, že platí

$$|\cos\xi| \leq 1$$

a proto

$$\underbrace{|e(x)|}_{\leq 10^{-6}} \leq \frac{|x^5|}{5!} = \frac{|x^5|}{120}$$

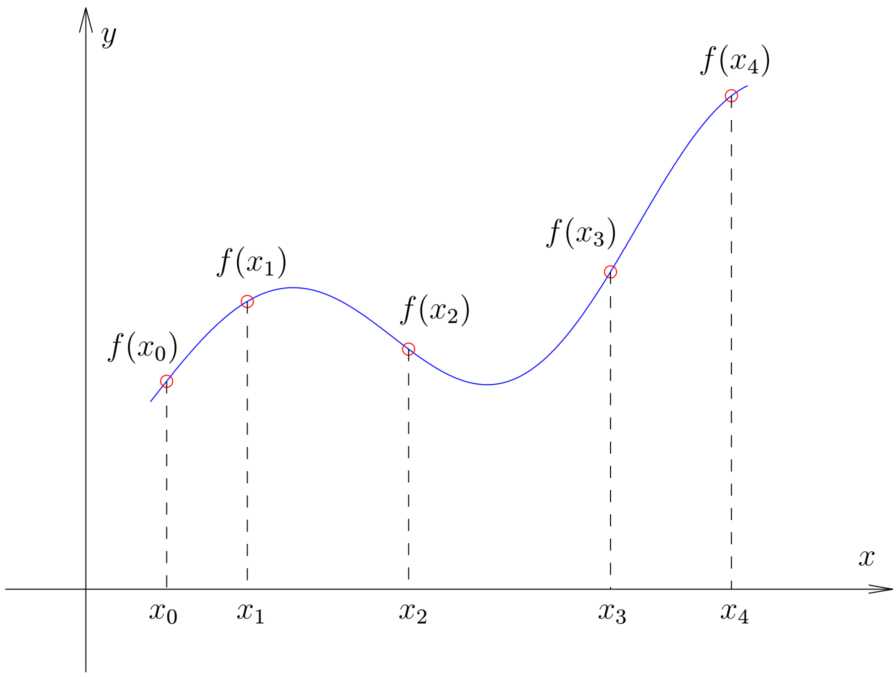
tj. chceme, aby

$$\frac{|x^5|}{120} \leq 10^{-6} \quad\Rightarrow\quad |x^5| \leq 120.10^{-6} \quad\Rightarrow\quad |x| \leq \underbrace{0.164375}_{\approx 9^\circ 25'} \text{ (rad)}$$

$\square$

Aproximace interpolačním polynomem

Chceme-li aproximovat funkci, která je dána svými hodnotami v $n+1$ bodech x_i, $i = 0, 1, \dots, n$ (body x_i nazýváme uzly interpolace), a požadujeme-li, aby aproximace procházela zadanými body, použijeme aproximaci interpolačním polynomem. Aproximace nám potom poslouží k získání přibližné hodnoty zadané funkce v libovolném bodě intervalu $\langle x_0, x_n \rangle$.



Máme-li zadány hodnoty funkce f v $n+1$ různých bodech, tzn. máme zadáno $n+1$ tzv. **interpolačních podmínek** pro polynom φ, je zřejmé, že stupeň hledaného polynomu bude n (polynom n-tého stupně má $n+1$ koeficientů). Lze ukázat, že mezi všemi polynomy nejvýše n-tého stupně existuje právě jeden, který je interpolačním polynomem pro zadanou funkci. Pro určení interpolačního polynomu existuje několik postupů, ale je třeba si uvědomit, že pro zadanou funkci všechny postupy určí stejný polynom. Ukážeme si dvě možnosti určení interpolačního polynomu, tzv. **Lagrangeův** a **Newtonův interpolační polynom**.

Lagrangeův interpolační polynom

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

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

Lagrangeův interpolační polynom hledáme ve tvaru

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

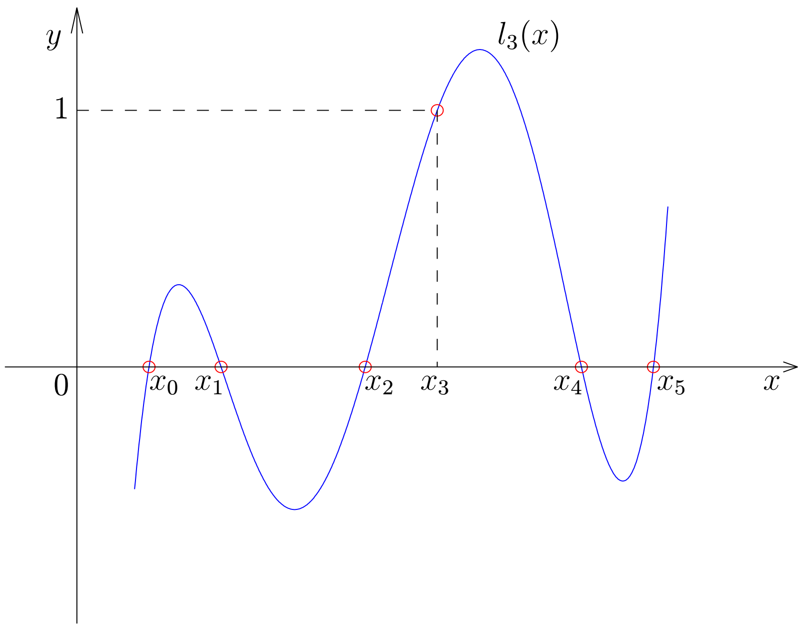
kde $l_i(x)$ jsou polynomy n-tého stupně takové, že platí

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

(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 $x_0, x_1, x_{i-1}, x_{i+1}, \dots x_n$ a nabývá hodnoty 1 v bodě x_i. Můžeme jej tedy zapsat ve tvaru

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

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



Příklad: Stanovte Lagrangeův interpolační polynom pro funkci f, která je dána tabulkou a určete přibližnou hodnotu $f(2)$.

i	0	1	2
x_i	0	1	3
$f(x_i)$	1	2	0

Řešení:

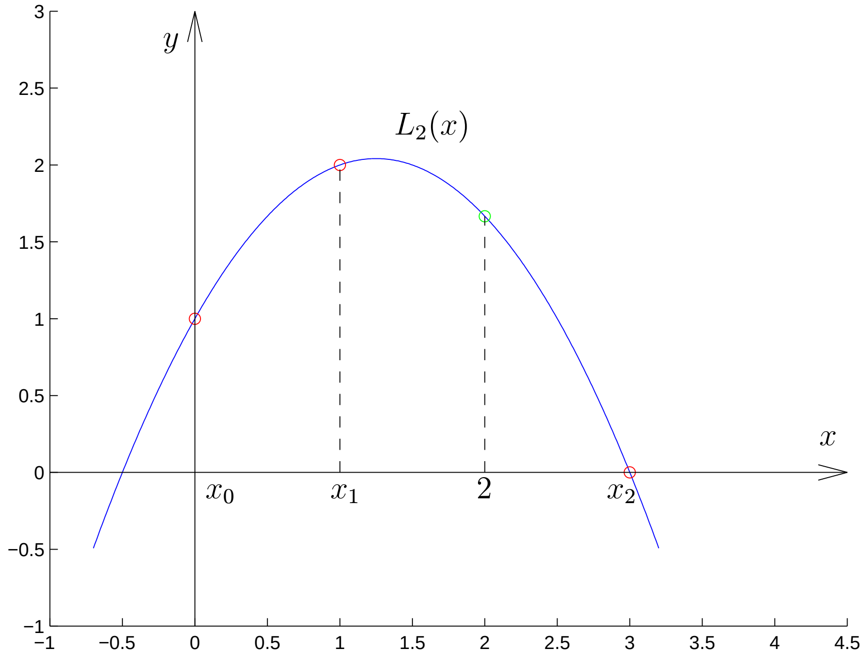
$$L_2(x) = f(x_0)\cdot l_0(x) + f(x_1)\cdot l_1(x) + f(x_2)\cdot l_2(x),$$

kde

$$\begin{aligned} l_0(x) &= \frac{(x-1)(x-3)}{(0-1)(0-3)} = \frac{1}{3}(x-1)(x-3) \\ l_1(x) &= \frac{(x-0)(x-3)}{(1-0)(1-3)} = -\frac{1}{2}x(x-3) \\ l_2(x) &= \frac{(x-0)(x-1)}{(3-0)(3-1)} = \frac{1}{6}x(x-1) \end{aligned}$$

Dosadíme

$$\begin{aligned} L_2(x) &= 1\cdot\left(\frac{1}{3}(x-1)(x-3)\right) + 2\cdot\left(-\frac{1}{2}x(x-3)\right) + 0\cdot\left(\frac{1}{6}x(x-1)\right) = \\ &= \frac{1}{3}(x^2-4x+3) - (x^2-3x) = \underline{\underline{-\frac{2}{3}x^2 + \frac{5}{3}x + 1}} \end{aligned}$$



$L_2(2) = ?$

$$L_2(2) = -\frac{2}{3}\cdot 2^2 + \frac{5}{3}\cdot 2 + 1 = \frac{-8+10+3}{3} = \frac{5}{3}$$

□

Newtonův interpolační polynom

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

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

Opět požadujeme splnění interpolačních podmínek

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

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). U Lagrangeova polynomu bychom museli celý výpočet provést znovu.

Ukažme si co dostaneme dosazováním interpolačních podmínek do předpisu polynomu:

$$N_n(x_0) = a_0 = f(x_0)$$

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

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

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

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

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

$$a_2 = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

Poznámka: Počítat koeficienty a_i přímo ze soustavy není praktické. Koeficienty budeme počítat pomocí tzv. **poměrných diferencí**. Kostrukci Newtonova polynomu si ukážeme v následujícím příkladě.

Příklad: Stanovte Newtonův interpolační polynom pro funkci f, která je dána tabulkou:

i	0	1	2	3
x_i	0	1	-1	3
$f(x_i)$	1	2	2	0

Řešení: Stupeň polynomu je $n = 3$ (jsou zadány 4 funkční hodnoty). Celý postup můžeme přehledně zapsat do tabulky:

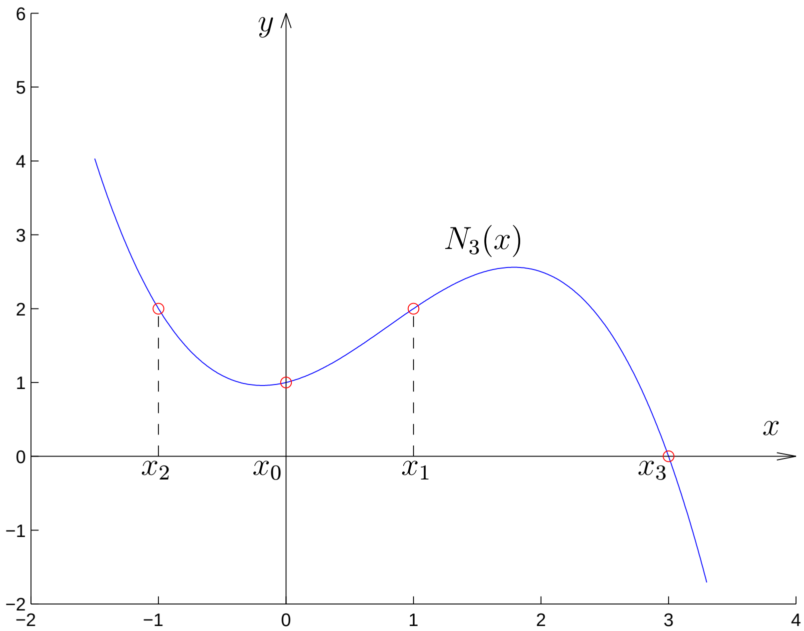
i	x_i	$f(x_i)$	$\dfrac{f(x_i)-f(x_{i-1})}{x_i-x_{i-1}} \overset{ozn.}{=} f^I(x_i)$	$\dfrac{f^I(x_i)-f^I(x_{i-1})}{x_i-x_{i-2}} \overset{ozn.}{=} f^{II}(x_i)$	$\dfrac{f^{II}(x_i)-f^{II}(x_{i-1})}{x_i-x_{i-3}} \overset{ozn.}{=} f^{III}(x_i)$
0	0	**1**			
1	1	2	$\dfrac{2-1}{1-0} = \mathbf{1}$		
2	-1	2	$\dfrac{2-2}{-1-1} = 0$	$\dfrac{0-1}{-1-0} = \mathbf{1}$	
3	3	0	$\dfrac{0-2}{3-(-1)} = -\dfrac{1}{2}$	$\dfrac{-\frac{1}{2}-0}{3-1} = -\dfrac{1}{4}$	$\dfrac{-\frac{1}{4}-1}{3-0} = -\mathbf{\dfrac{5}{12}}$

V tabulce jsou na diagonále postupně uvedeny koeficienty a_0, a_1, a_2 a a_3 Newtonova interpolačního polynomu

$$N_3(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + a_3(x-x_0)(x-x_1)(x-x_2).$$

Po dosazení

$$N_3(x) = 1 + 1(x-0) + 1(x-0)(x-1) - \frac{5}{12}(x-0)(x-1)(x+1) = \ldots =$$
$$= \underline{\underline{-\frac{5}{12}x^3 + x^2 + \frac{5}{12}x + 1}}$$



□

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

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

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

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

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

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.

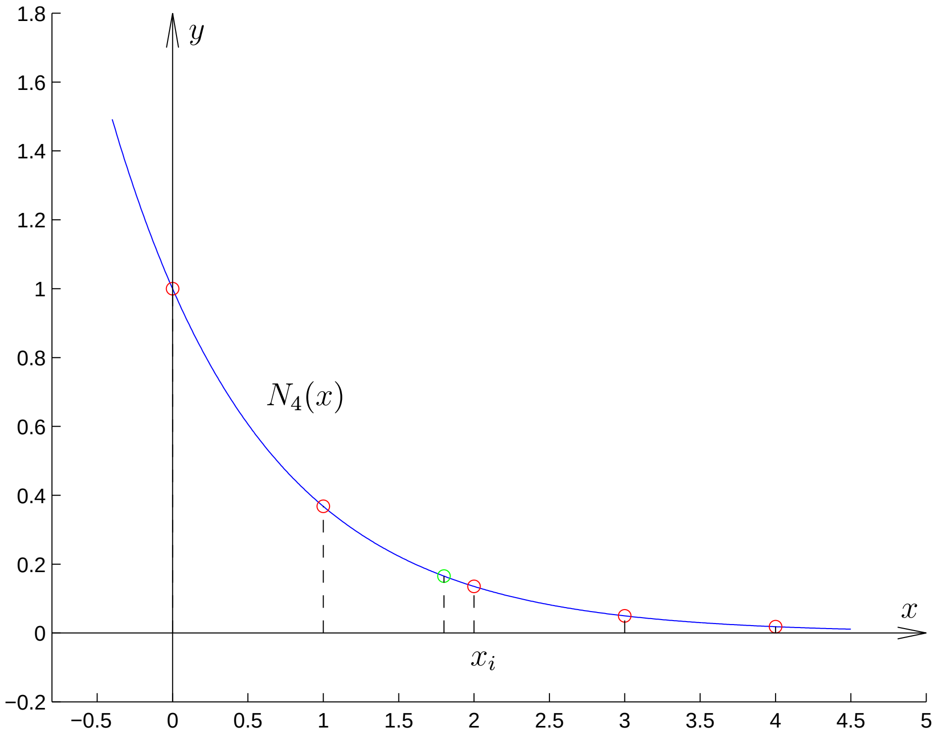
Příklad: Vypočtěte $f(1{,}8)$, kde funkce $f(x)$ je dána tabulkou:

x_i	0	1	2	3	4
$f(x_i)$	1,0000	0,36788	0,13534	0,04979	0,01832

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 Nevillova algoritmu, popř. o vhodnosti interpolace pomocí $N_n(x)$.

Nevillovo schéma:

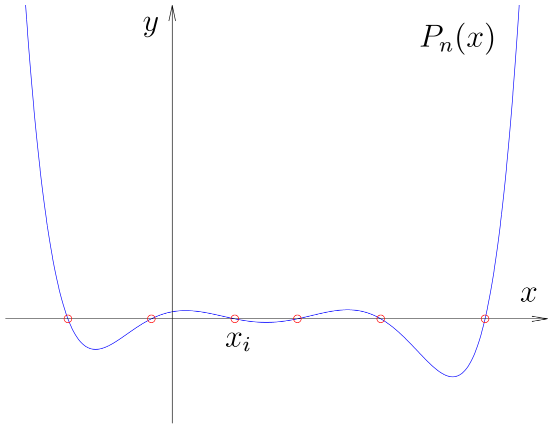
$\lvert\alpha - x_i\rvert$	x_i	$f(x_i)$				
0,2	2	**0,13534**				
0,8	1	0,36788	**0,18185**			
1,2	3	0,04979	0,24064	**0,17009**		
1,8	0	1,00000	0,42987	0,08926	**0,16201**	
2,2	4	0,01832	0,55824	0,27583	0,13901	**0,16431**



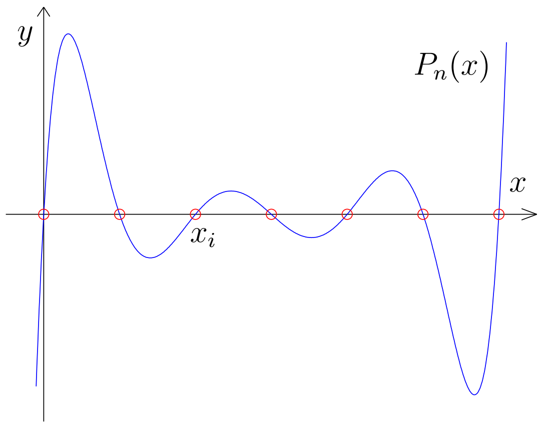
□

Poznámky:

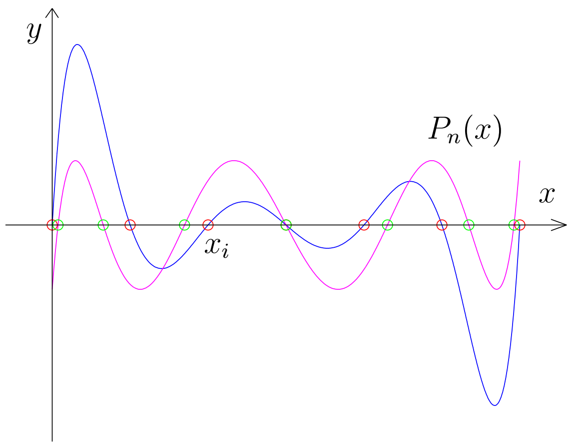
1) Interpolační polynomy vyšších řádů není vhodné užívat pro aproximaci hodnot funkce mimo interval obsahující uzly interpolace (tzv. extrapolaci), protože absolutní hodnota polynomu nabývá velkých hodnot.



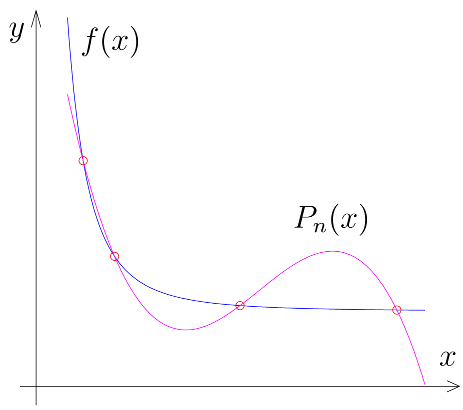
2) Dále není obecně vhodné interpolovat polynomem funkci, která je dána velkým počtem svých hodnot. Stupeň interpolačního polynomu by potom byl velký a vlastnosti polynomu vysokého stupně nejsou dobré nejen mimo interval obsahující uzly interpolace, ale i uvnitř. Interpolační polynom směrem ke krajním bodům intervalu může vlivem nepřesností ve vstupních datech výrazně oscilovat, zejména při použití ekvidistantních uzlů.



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

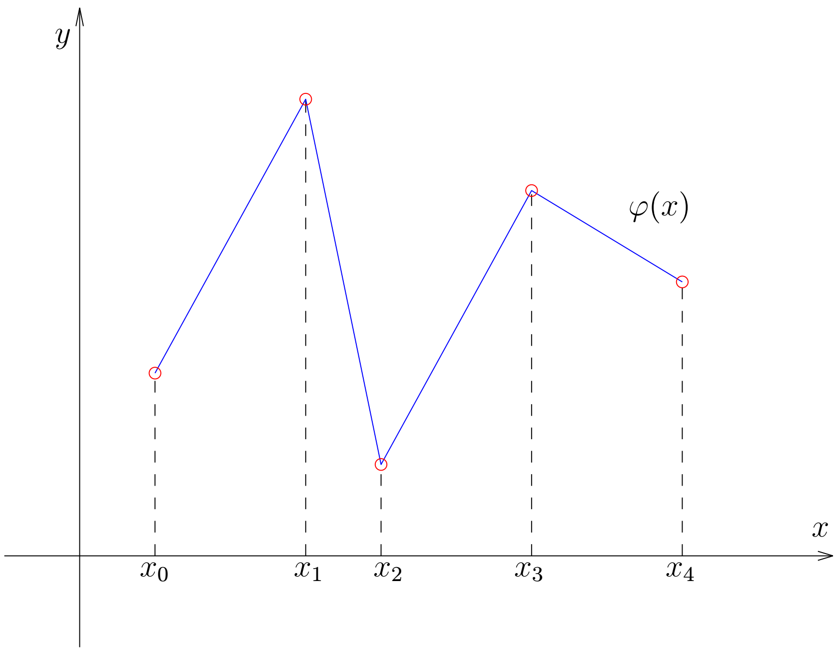


4) Interpolace polynomem není obecně vhodná např. pro funkce, které mají asymptotu. Toto je jedním z důvodů, proč zavádíme tzv. **interpolaci spline funkcemi**.



Interpolace spline funkcemi

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



Nejvíce používanou je tzv. **kubická spline interpolace** vycházející z myšlenky, že danou funkci f můžeme na $\langle a, b\rangle$ aproximovat funkcemi, které jsou po částech polynomy 3.stupně. Poznamenejme zde, že ostatní volby stupně polynomů nepřináší lepší výsledky a výpočty jsou v případě větších stupňů složitější.

Interval $\langle a, b\rangle$ rozdělíme na n dílčích intervalů

$$a = x_0 < x_1 < x_2 < \ldots < x_n = b.$$

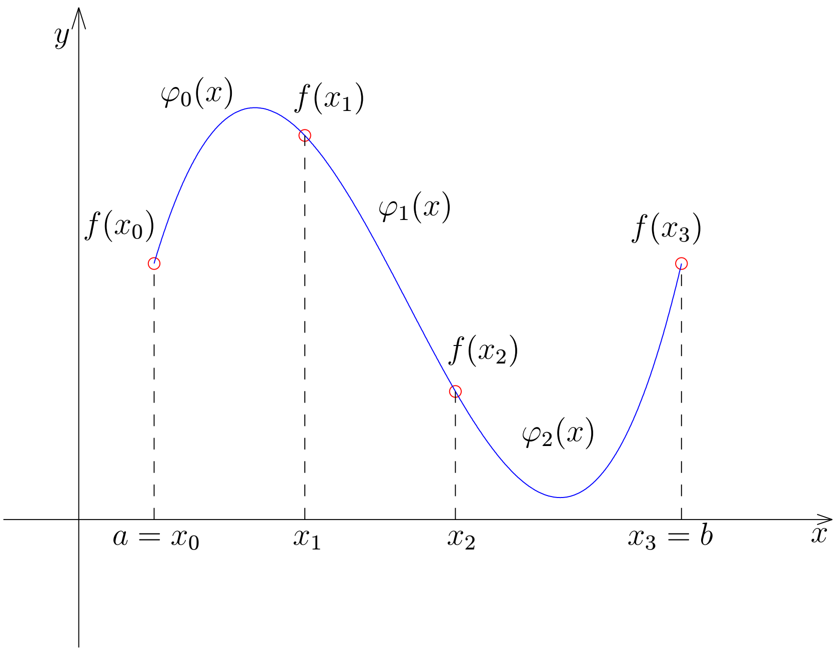
Funkci f aproximujeme funkcí φ, pro kterou platí:

a) $\varphi \in \mathbb{C}^2(\langle a, b\rangle)$ (funkce φ je spojitá a má spojité první i druhé derivace)

b) na každém intervalu $\langle x_i, x_{i+1}\rangle$ je φ polynom 3. stupně

V následujícím příkladu jsou rozepsány podmínky pro konstrukci kubické spline funkce.

Příklad :



Jednotlivé funkce $\varphi_i(x)$ (na každém intervalu $\langle x_i, x_{i+1}\rangle$ jde o jinou funkci) mají tvar:

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

Musí platit: $\quad (\varphi \in \mathbb{C}^2(\langle a, b\rangle)$ a podmínky interpolace)

1) $\quad \left.\begin{aligned} \varphi_0(x_1) &= \varphi_1(x_1) \\ \varphi_1(x_2) &= \varphi_2(x_2) \end{aligned}\right\}$ spojitost funkce φ

2) $\quad \left.\begin{aligned} \varphi_0'(x_1) &= \varphi_1'(x_1) \\ \varphi_1'(x_2) &= \varphi_2'(x_2) \end{aligned}\right\}$ spojitost 1. derivace funkce φ

3) $\quad \left.\begin{aligned} \varphi_0''(x_1) &= \varphi_1''(x_1) \\ \varphi_1''(x_2) &= \varphi_2''(x_2) \end{aligned}\right\}$ spojitost 2. derivace funkce φ

4) $\quad \left.\begin{aligned} \varphi_0(x_0) &= f(x_0) \\ \varphi_1(x_1) &= f(x_1) \\ \varphi_2(x_2) &= f(x_2) \\ \varphi_2(x_3) &= f(x_3) \end{aligned}\right\}$ interpolační podmínky

Poznámka

Dostali jsme 10 podmínek na funkci φ, která je však určena 3×4 parametry a_i, b_i, c_i, d_i (tj. 12 podmínek) $\Rightarrow$ chybí ještě 2 podmínky.

Je vhodné doplnit některou z následujících dvojic podmínek:

5) a) $\varphi'(a) = f'(a),\ \varphi'(b) = f'(b)$ ... podmínky tečen

b) $\varphi'(a) = \varphi'(b),\ \varphi''(a) = \varphi''(b)$ ... podmínky periodicity

c) $\varphi''(a) = 0, \quad \varphi''(b) = 0$ ... tzv. přirozené podmínky

Poznámka

Uvědomme si, že

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

Závěr

Podmínky 1) - 5) představují soustavu lineárních algebraických rovnic s řídkou maticí. Jejím vyřešením určíme koeficienty a_i, b_i, c_i, d_i a tím i funkce φ_i, tj. φ.

Diskrétní L$_2$-aproximace

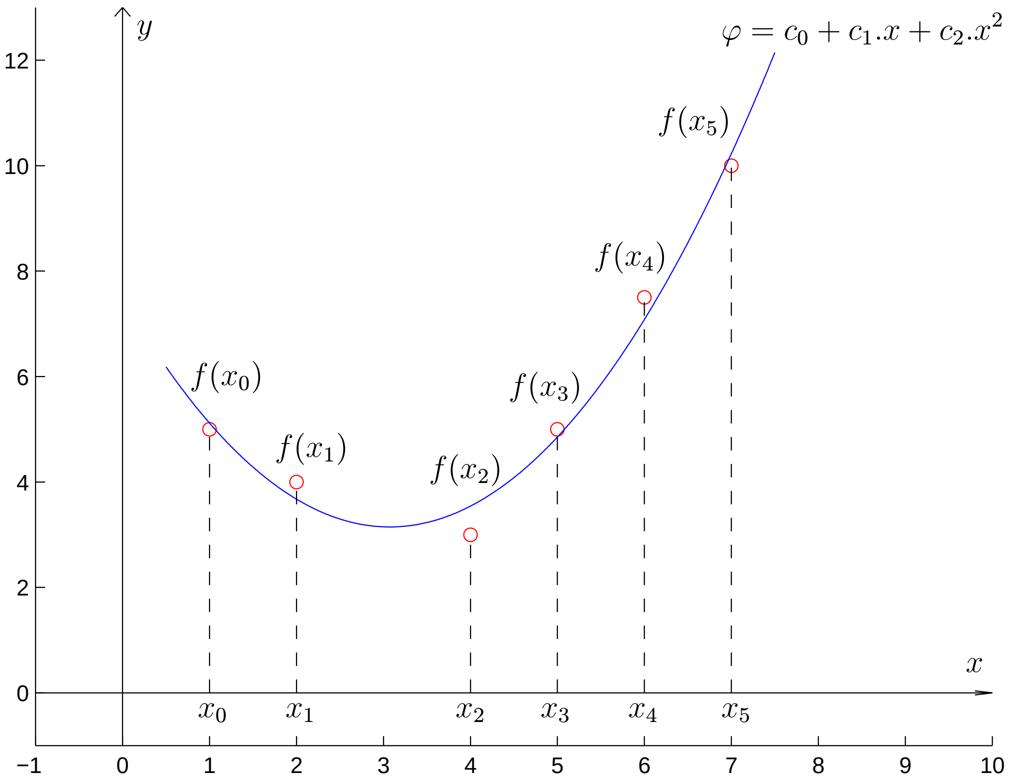
Myšlenka:

Chceme aproximovat funkci, která je dána tabulkou $\{(x_i, f(x_i))\}$. V případě, kdy jsou $f(x_i)$ zatíženy chybou (např. výsledky měření), není vhodné provádět interpolaci. Aproximaci φ hledáme ve tvaru

$$\varphi(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \ldots + c_n\varphi_n(x),$$

kde φ_i jsou zadané funkce a c_i hledané parametry (počet bázových funkcí φ_i je menší než počet zadaných bodů, v případě rovnosti se již jedná o interpolaci). Naším cílem je minimalizovat „souhrnou odchylku“ funkce φ od zadaných dat.

Ilustrační obrázek:



Metodu diskrétní L_2-aproximace popíšeme na následujícím příkladu.

Příklad

Je dána funkce f tabulkou

x_i	$0,5$	$0,8$	$0,9$	$1,1$	$1,2$
$f(x_i)$	$2,25$	$0,72$	$0,33$	$-0,27$	$-0,48$

Tuto funkci chceme aproximovat lineární funkcí metodou nejmenších čtverců.

Lineární funkce $$\varphi = c_0 . \underbrace{1}_{\varphi_0(x)} + c_1 . \underbrace{x}_{\varphi_1(x)}$$

Minimalizujeme funkci

$$r(c_0, c_1) = \sum_{i=1}^{5} |f(x_i) - \varphi(x_i)|^2 = \sum_{i=1}^{5} |f(x_i) - c_0 - c_1 x_i|^2$$

Podmínky minima

$$\left.\begin{array}{ll} (1) & \dfrac{\partial r}{\partial c_1} = -2 \sum\limits_{i=1}^{5} (f(x_i) - c_0 - c_1 x_i) x_i = 0 \\ (2) & \dfrac{\partial r}{\partial c_0} = -2 \sum\limits_{i=1}^{5} (f(x_i) - c_0 - c_1 x_i) = 0 \end{array}\right\} \text{ 2 rovnice pro neznámé } c_0, c_1.$$

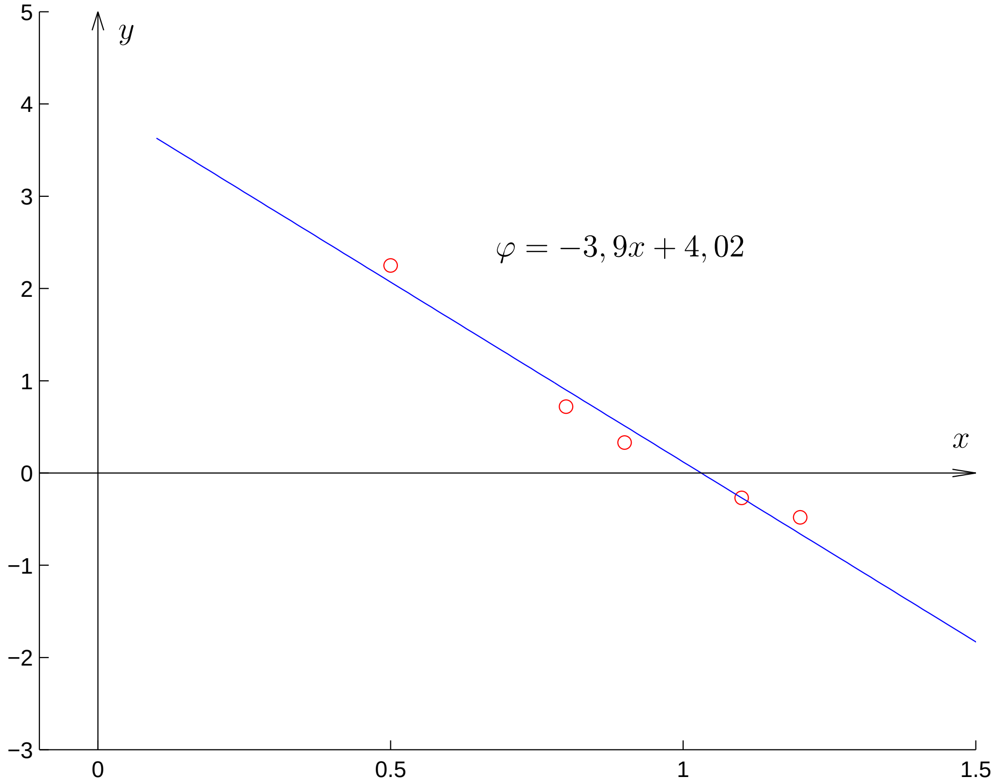
Konkrétně

$$\begin{array}{ll} (1) & \quad (2,25 - c_0 - 0,5c_1)0,5 + (0,72 - c_0 - 0,8c_1)0,8 + \\ & + (0,33 - c_0 - 0,9c_1)0,9 + (-0,27 - c_0 - 1,1c_1)1,1 + \\ & + (-0,48 - c_0 - 1,2c_1)1,1 = 0 \\ (2) & \quad (2,25 - c_0 - 0,5c_1) + (0,72 - c_0 - 0,8c_1) + \\ & + (0,33 - c_0 - 0,9c_1) + (-0,27 - c_0 - 1,1c_1) + \\ & + (-0,48 - c_0 - 1,2c_1) = 0 \end{array}$$

Po sečtení

$$\left.\begin{array}{lrcrcr} (1) & -4,35c_1 & - & 4,5c_0 & = & -1,125 \\ (2) & -4,5c_1 & - & 5c_0 & = & -2,55 \end{array}\right\} \Rightarrow \quad \begin{array}{l} c_1 = -3,9 \\ c_0 = 4,02 \end{array}$$

$$\underline{\varphi = -3,9x + 4,02}$$



Jiný přístup:

Zapišme interpolační podmínky

$$c_1 x_i + c_0 = f(x_i).$$

Jejich přesné splnění samozřejmě obecně nelze zaručit (pouze ve speciálních případech), protože získáme tzv. přeurčenou soustavu (více rovnic než neznámých):

$$\begin{bmatrix} 0,5 & 1 \\ 0,8 & 1 \\ 0,9 & 1 \\ 1,1 & 1 \\ 1,2 & 1 \end{bmatrix} . \begin{bmatrix} c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} 2,25 \\ 0,72 \\ 0,33 \\ -0,27 \\ -0,48 \end{bmatrix} \qquad \Leftrightarrow \qquad Q.c = f$$

Získanou přeurčenou soustavu řešíme ve smyslu nejmenších čtverců, tj.

$$Q^T Q c = Q^T f$$

(jedná se o stejnou soustavu jako v prvním přístupu).

Poznámka

V případě, že některé hodnoty chceme eliminovat, například důsledkem špatného měření, nebo když jsou hodnoty pro větší x_i zatíženy větší chybou, je vhodné použít váhy, tj. minimalizujeme

$$r(c_i) = \sum_{i=1}^{n} |f(x_i) - \varphi(x_i)|^2 \, w_i$$

Poznámka

Otázkou ještě zůstává volba tvaru $\varphi(x)$.
První možností je volit

$$\varphi_0(x) = 1, \quad \varphi_1(x) = x, \quad \varphi_2(x) = x^2, \; \dots \;, \quad \varphi_n(x) = x^n.$$

Opět je třeba určit ještě stupeň polynomu (a to např. ze znalosti chování funkce f nebo pomocí statistických metod).

Ukazuje se, že takto volené $\varphi_i(x)$ nejsou nejlepší pro výpočty.

→ Soustava normálních rovnic je pro větší n špatně podmíněná.

Za bázové funkce φ_i je vhodné volit ortogonální polynomy, pro které platí

$$(\varphi_i, \varphi_j) = \delta_{ij} = \begin{matrix} 1, & i = j \\ 0, & i \neq j \end{matrix}$$

Symbol (f, g) představuje skalární součin funkcí, tj.

$$\sum_{i=1}^{n} f(x_i) \, g(x_i) \text{ resp. } \int_a^b f g \, dx \text{ ve spojitém případě.}$$

Příklady ortogonálních polynomů:

Čebyševovy, Legendrovy, Laguerrovy, Gramovy ... viz literatura

→ Soustava normálních rovnic má pro ortogonální polynomy diagonální matici. (rozmyslet!)

Poznámka

Analogické úvahy jako v případě diskrétní L_2-aproximace pro funkci zadanou tabulkou můžeme provést i pro funkci zadanou na celém intervalu $\langle a, b \rangle$. Získáme tzv. **spojitou $\mathbf{L_2}$-aproximaci** funkce. (Součty přejdou formálně na integrály přes daný interval.) Princip výpočtu demostruje následující příklad.

Spojitá L$_2$-aproximace

Příklad

Stanovte spojitou L_2-aproximaci funkce $f(x) = \sqrt{x}$, $x \in \langle 0, 1\rangle$ lineární funkcí $\varphi(x) = c_1 x + c_0$.

Minimalizujeme funkci

$$r(c_0, c_1) = \int_0^1 |f(x) - \varphi(x)|^2 \, dx = \int_0^1 (\sqrt{x} - c_0 - c_1 x)^2 \, dx.$$

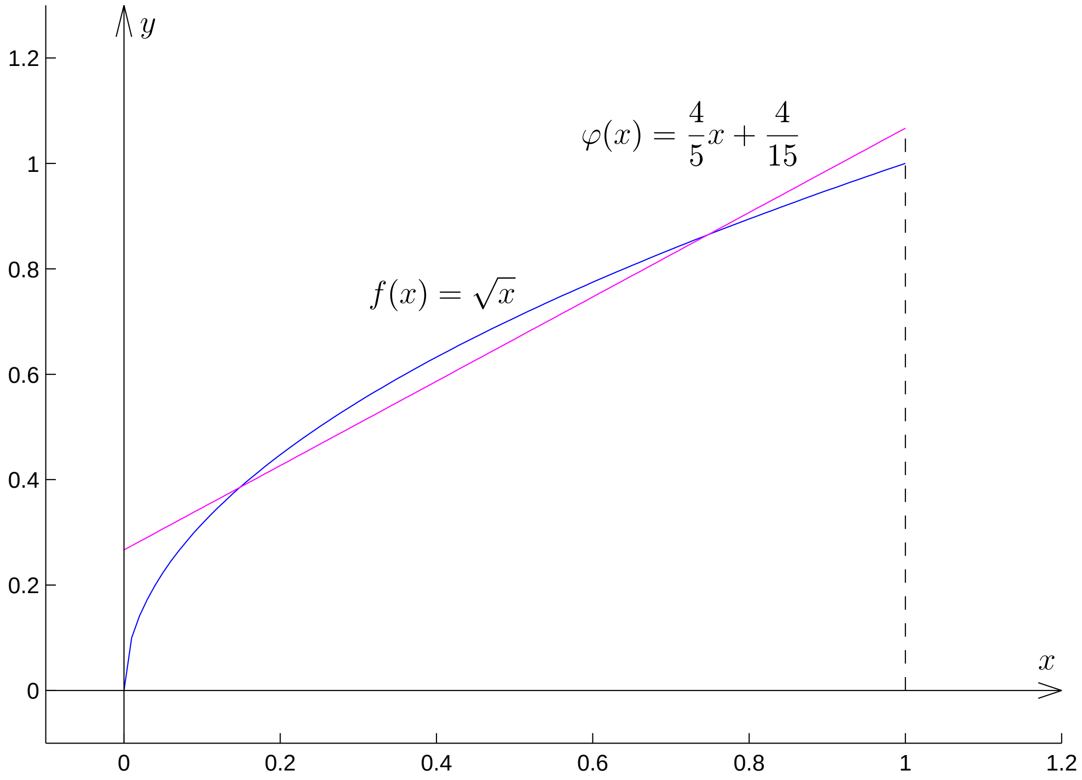
Podmínky minima

$$\left.\begin{aligned} &(1) \quad \frac{\partial r}{\partial c_1} = -2 \int_0^1 (\sqrt{x} - c_0 - c_1 x) x \, dx = 0 \\ &(2) \quad \frac{\partial r}{\partial c_0} = -2 \int_0^1 (\sqrt{x} - c_0 - c_1 x) \, dx = 0 \end{aligned}\right\} \text{ 2 rovnice pro neznámé } c_0, c_1.$$

$$\begin{aligned} &(1) \quad -2\left[\frac{2}{5} x^{\frac{5}{2}} - c_0 \frac{x^2}{2} - c_1 \frac{x^3}{3}\right]_0^1 = 0 \\ &(2) \quad -2\left[\frac{2}{3} x^{\frac{3}{2}} - c_0 x - c_1 \frac{x^2}{2}\right]_0^1 = 0 \end{aligned}$$

$$\left.\begin{aligned} &(1) \quad \frac{2}{5} - \frac{1}{2} c_0 - \frac{1}{3} c_1 = 0 \\ &(2) \quad \frac{2}{3} - c_0 - \frac{1}{2} c_1 = 0 \end{aligned}\right\} \Rightarrow \left.\begin{aligned} \frac{1}{2} c_0 + \frac{1}{3} c_1 &= \frac{2}{5} \\ c_0 + \frac{1}{2} c_1 &= \frac{2}{3} \end{aligned}\right\} \Rightarrow \begin{aligned} c_1 &= \frac{4}{5} \\ c_0 &= \frac{4}{15} \end{aligned}$$

$$\varphi(x) = \frac{4}{5} x + \frac{4}{15}$$



$\square$

Poznámka

Obecně lze opět zavést váhovou funkci $w = w(x)$ a minimalizovat

$$r(c_i) = \int_a^b |f(x) - \varphi(x)|^2 w(x)\, dx$$

Fourierova analýza

Do této chvíle jsme se zabývali aproximacemi funkce pouze pomocí polynomů. V úvodu jsme uvedli, že za bázové funkce můžeme volit libovolné funkce. Například pro aproximaci periodických funkcí není vhodné použít polynomy (a to jak ve smyslu interpolace tak ve smyslu L_2-aproximace). Pro aproximaci periodických funkcí je vhodné použít nějaký systém periodických bázových funkcí, např. systém tzv. **trigonometrických polynomů**:

$$\begin{aligned}
\varphi_0(x) &= 1 \\
\varphi_{2k-1}(x) &= \cos\frac{2\pi kx}{T} \qquad k = 1, 2, \ldots \\
\varphi_{2k}(x) &= \sin\frac{2\pi kx}{T} \qquad k = 1, 2, \ldots,
\end{aligned}$$

kde T představuje periodu zadané funkce (vzdálenost prvního a posledního uzlu v diskrétním případě, resp. délku zadaného intervalu ve spojitém případě). Pro jednoduchost uvažujeme ekvidistantní uzly (v diskrétním případě). Počet uvažovaných bázových funkcí volíme buď menší než je počet zadaných bodů (ve smyslu L_2-aproximace), nebo roven počtu zadaných bodů (ve smyslu interpolace).

Jednoduchým cvičením je ukázat, že systém trigonometrických polynomů je ortogonální (jak v diskrétním tak ve spojitém případě). Ověřte!

Úlohu najít koeficienty c_i u bázových funkcí φ_i z vyjádření

$$\varphi(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \ldots + c_n\varphi_n(x).$$

nazýváme v tomto případě **Fourierovou analýzou**.

Formálně pouze přeznačíme koeficienty c_i, tj. u bázové funkce $\varphi_0(x) = 1$ použijeme koeficient A_0, u bázových funkcí $\varphi_{2k-1}(x) = \cos(2\pi kx)/T$ použijeme koeficienty A_k a u bázových funkcí $\varphi_{2k}(x) = \sin(2\pi kx)/T$ použijeme koeficienty B_k.

Následující jednoduchý příklad naznačí princip Fourierovy analýzy.

Příklad

Aproximujte 2π-periodickou funkci zadanou tabulkou za použití maximálního počtu bázových funkcí (tj. ve smyslu interpolace).

x_i	0	$\pi/2$	π	$3\pi/2$
$f(x_i)$	12	-4	0	4

Řešení

Ze zadání je zřejmé, že perioda zadané funkce je 2π. Aproximující trigonometrický polynom budeme tedy volit ve tvaru

$$\varphi(x) = A_0 + A_1\cos x + B_1\sin x + A_2\cos 2x.$$

Zapíšeme interpolační podmínky

$$\varphi(x_j) = f(x_j), \quad j = 0, 1, 2, 3,$$

tj.

$$\begin{bmatrix} 1 & \cos x_0 & \sin x_0 & \cos 2x_0 \\ 1 & \cos x_1 & \sin x_1 & \cos 2x_1 \\ 1 & \cos x_2 & \sin x_2 & \cos 2x_2 \\ 1 & \cos x_3 & \sin x_3 & \cos 2x_3 \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ B_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ f(x_3) \end{bmatrix},$$

tj.

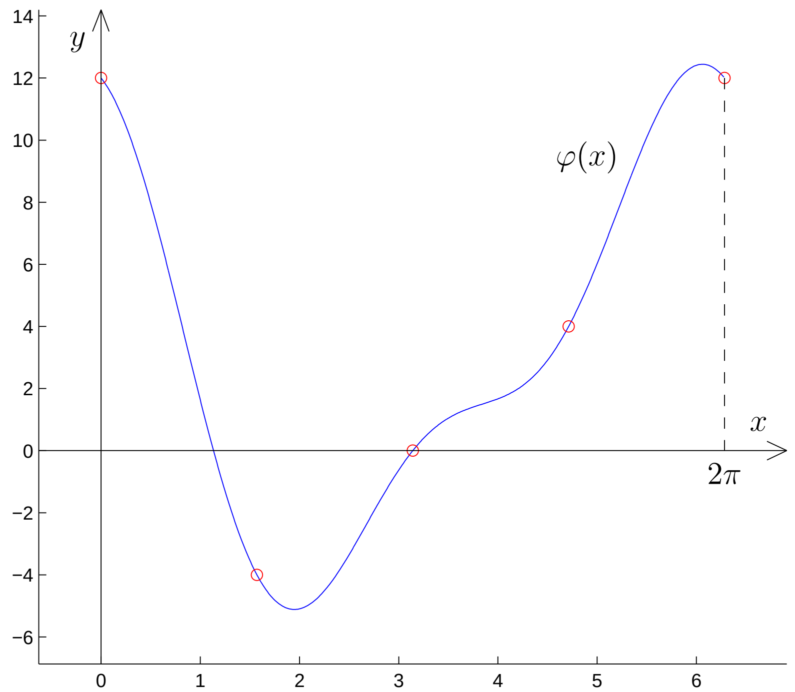
$$\begin{bmatrix} 1 & \cos 0 & \sin 0 & \cos 0 \\ 1 & \cos \pi/2 & \sin \pi/2 & \cos \pi \\ 1 & \cos \pi & \sin \pi & \cos 2\pi \\ 1 & \cos 3\pi/2 & \sin 3\pi/2 & \cos 3\pi \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ B_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} 12 \\ -4 \\ 0 \\ 4 \end{bmatrix},$$

tj.

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & -1 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ B_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} 12 \\ -4 \\ 0 \\ 4 \end{bmatrix}.$$

Vyřešením soustavy získáme hledané koeficienty $A_0 = 3$, $A_1 = 6$, $B_1 = -4$, $A_2 = 3$ a tím i aproximující trigonometrický polynom

$$\underline{\varphi(x) = 3 + 6\cos x - 4\sin x + 3\cos 2x.}$$



$\square$

Úlohu a řešení Fourierovy analýzy lze formulovat elegantně použitím komplexní proměnné. Uvažujme pro jednoduchost lichý počet bázových funkcí ($N = 2L + 1$) a periodu dané funkce 2π. Potom má aproximující funkce tvar

$$\varphi(x) = A_0 + \sum_{k=1}^{L} (A_k \cos kx + B_k \sin kx). \qquad (*)$$

Pro funkce $\sin x$ a $\cos x$ platí vztahy

$$\cos x = \frac{e^{ix} + e^{-ix}}{2} \ , \quad \sin x = \frac{e^{ix} - e^{-ix}}{2i} = -\frac{1}{2}\, i\, (e^{ix} - e^{-ix})$$

a tedy

$$\varphi(x) = A_0 + \sum_{k=1}^{L} \left(\frac{1}{2}\, A_k\, (e^{ikx} + e^{-ikx}) - \frac{1}{2}\, i\, B_k\, (e^{ikx} - e^{-ikx}) \right) =$$

$$= A_0 + \sum_{k=1}^{L} \left(\frac{1}{2}\, (A_k - i\, B_k)\, e^{ikx} + \frac{1}{2}\, (A_k + i\, B_k)\, e^{-ikx} \right).$$

Označíme-li

$$C_0 = A_0 \ , \quad C_k = \frac{1}{2}\, (A_k - i\, B_k) \ , \quad C_{-k} = \frac{1}{2}\, (A_k + i\, B_k)$$

dostaneme

$$\underline{\varphi(x) = \sum_{k=-L}^{L} C_k\, e^{ikx}.}$$

Pro koeficienty dostaneme vynásobením $(*)$ jednotlivými bázovými funkcemi, využitím jejich ortogonality a interpolačních podmínek předpisy:

$$A_0 = \frac{1}{N} \sum_{j=0}^{N-1} f(x_j)$$

$$A_k = \frac{2}{N} \sum_{j=0}^{N-1} f(x_j)\, \cos kx_j$$

$$B_k = \frac{2}{N} \sum_{j=0}^{N-1} f(x_j)\, \sin kx_j$$

$$C_k = \frac{1}{N} \sum_{j=0}^{N-1} f(x_j)\, e^{-ikx_j}$$

Poznámka

Vezmeme-li aproximující polynom o menším počtu bázových funkcí než je počet zadaných bodů, jedná se o aproximaci ve smyslu metody nejmenších čtverců, tj. diskrétní L_2-aproximaci. Potom obecně nemohou být splněny interpolační podmínky přesně (pouze ve speciálních případech).

Poznámka

Výpočet koeficientů C_k představuje sčítání konečné řady. Uvažujeme-li počet aproximujících bázových funkcí N jako mocninu čísla 2 (tj. $N = 2^M$), lze odvodit velmi rychlý a efektivní algoritmus pro výpočet koeficientů C_k. Tento algoritmus se potom nazývá **rychlá Fourierova analýza**. Podrobněji viz literatura.