Numerické metody — aproximace funkcí

Mirko Navara
http://cmp.felk.cvut.cz/~navara/
katedra kybernetiky FEL ČVUT
Karlovo náměstí, budova G, místnost 104a

https://moodle.fel.cvut.cz/courses/B4B01NUM

25. září 2023

Obsah

1 APROXIMACE FUNKCÍ **9**
1.1 Typické úlohy . . . 9
1.1.1 Aproximace funkcí v ekonomii . . . 9
1.1.2 Aproximace funkcí v teorii pravděpodobnosti a matematické analýze . . . 9
1.1.3 Aproximace funkcí v elektrotechnice . . . 9
1.1.4 Základní úloha aproximace . . . 10
1.2 Interpolace . . . 11
1.2.1 Prostá interpolace . . . 11
1.3 Interpolace polynomem . . . 12
1.3.1 Lagrangeova konstrukce interpolačního polynomu . . . 13
1.3.2 Newtonova konstrukce interpolačního polynomu . . . 15
1.3.3 Nevillův algoritmus . . . 16
1.3.4 Chyba aproximace interpolačním polynomem . . . 17
1.3.5 Čebyševovy polynomy . . . 20
1.3.6 Hermitův interpolační polynom . . . 23
1.3.7 Aproximace Taylorovou řadou . . . 24
1.4 Interpolace spliny . . . 26
1.4.1 Kubický spline . . . 26
1.5 Metoda nejmenších čtverců . . . 30
1.5.1 Řešení aproximace podle kritéria nejmenších čtverců . . . 30
1.5.2 Ortogonalizace . . . 31
1.5.3 Aproximace goniometrickým polynomem . . . 33
1.5.4 Čebyševova aproximace polynomem . . . 33
1.6 Dodatek: Přehled značení . . . 35

2 NUMERICKÁ DERIVACE **28**
2.1 Formulace problému . . . 28
2.2 Chyba numerické derivace . . . 29
2.3 Odhady chyb numerické derivace . . . 29
2.4 Doporučená délka kroku . . . 30

3 RICHARDSONOVA EXTRAPOLACE **32**
3.1 Obecný princip Richardsonovy extrapolace . . . 32
3.2 Využití Richardsonovy extrapolace v numerické derivaci . . . 32
3.3 Dodatek: Přehled značení . . . 34

6 NUMERICKÁ INTEGRACE **65**

7 Newtonovy-Cotesovy vzorce **66**
7.1 Metoda levých obdélníků . . . 66
7.2 Obdélníková metoda . . . 66
7.3 Lichoběžníková metoda . . . 67
7.4 Simpsonova metoda . . . 67
7.5 Obecné Newtonovy-Cotesovy vzorce . . . 68

8 Odhad chyby numerické integrace **69**
8.1 Řád metod integrace . . . 69

9 Gaussova metoda integrace **70**

10 Richardsonova extrapolace **72**

11 Richardsonova extrapolace při integraci **72**

12 Rombergova metoda **75**

13 Praktické stanovení počtu intervalů **76**

14 Řešení obtížnějších úloh úpravou zadání **77**
14.1 Integrace přes nekonečný interval . . . 77
14.2 Omezení intervalu . . . 77
14.3 Pomalu konvergentní integrály . . . 77
14.4 Dodatek: Přehled značení . . . 80

4 NUMERICKÉ ŘEŠENÍ NELINEÁRNÍCH ROVNIC **35**
4.1 Formulace problému . . . 35
4.2 Metoda půlení intervalu neboli bisekce . . . 35
4.3 Metoda regula falsi . . . 35
4.4 Univerzální odhad chyby . . . 36
4.5 Metoda sečen . . . 37
4.6 Newtonova metoda (metoda tečen) . . . 37
4.6.1 Odhad chyby Newtonovy metody . . . 38
4.6.2 Konvergence Newtonovy metody . . . 39
4.6.3 Náhrada derivace numerickým odhadem . . . 39
4.7 Rychlost konvergence (řád metody) . . . 40
4.7.1 Řád Newtonovy metody . . . 42
4.7.2 Řád metody regula falsi . . . 42
4.8 Kombinace startovacích a zpřesňujících metod . . . 44
4.9 Metoda prosté iterace (MPI) . . . 45
4.9.1 Kontraktivní funkce . . . 46
4.9.2 Věta o pevném bodě . . . 46
4.9.3 Optimalizace MPI . . . 46
4.9.4 Řád metody prosté iterace . . . 47
4.9.5 Kritéria pro výběr metody řešení rovnic . . . 48
4.10 Podobné úlohy . . . 48
4.10.1 Hledání násobných kořenů . . . 48
4.10.2 Řešení algebraických rovnic neboli hledání kořenů polynomů . . . 49
4.10.3 Řešení rovnic v komplexním oboru . . . 49
4.10.4 Řešení soustav rovnic . . . 49
4.11 Dodatek: Přehled značení . . . 50

5 NUMERICKÉ ŘEŠENÍ SOUSTAV LINEÁRNÍCH ROVNIC **51**
5.1 Formulace úlohy a její obtíže . . . 51
5.1.1 Druhy problémů . . . 51
5.1.2 Špatná podmíněnost . . . 51
5.1.3 Zdroje chyb . . . 52
5.2 Přímé metody . . . 52
5.2.1 Gaussova eliminace (GEM) . . . 52
5.2.2 Výběr hlavního prvku . . . 52
5.2.3 Gaussova-Jordanova redukce . . . 53
5.2.4 LU-rozklad . . . 53
5.2.5 Výpočet inverzní matice . . . 54
5.2.6 Výpočet determinantu . . . 54
5.2.7 Zpřesnění výsledků pomocí rezidua . . . 55
5.3 Iterační metody . . . 55
5.3.1 Normy vektorů a matic . . . 55
5.3.2 Vlastní čísla a spektrální poloměr . . . 59
5.3.3 Výpočet vlastních čísel . . . 59
5.3.4 Maticové iterační metody . . . 60
5.3.5 Jacobiova iterační metoda (JIM) . . . 61
5.3.6 Gaussova-Seidelova iterační metoda (GSM) . . . 61
5.3.7 Superrelaxační metoda (SOR – Successive OverRelaxation method) . . . 62
5.4 Jaký postup volit? . . . 63
5.5 Dodatek: Přehled značení . . . 64

1 APROXIMACE FUNKCÍ

1.1 Typické úlohy

1.1.1 Aproximace funkcí v ekonomii

Úloha: Odhadněte rychlost růstu průmyslové výroby v posledním období a nejlépe i v blízké budoucnosti.
Úloha: Ze známých hodnot burzovních indexů do dnešního dne odhadněte jejich zítřejší hodnoty.

1.1.2 Aproximace funkcí v teorii pravděpodobnosti a matematické analýze

Úloha: Náhodná veličina s normovaným normálním rozdělením je popsána distribuční funkcí

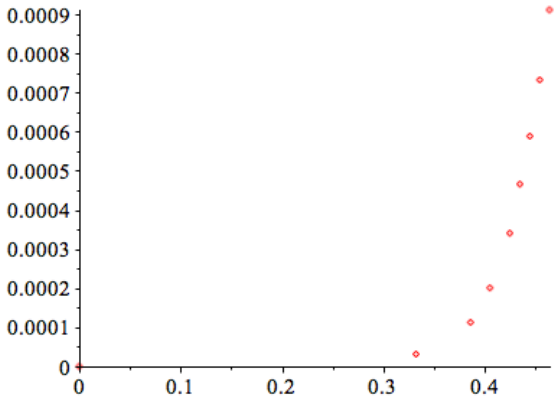
$$\Phi(u) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{u} \exp\left(\frac{-t^2}{2}\right) \mathrm{d}t\,.$$

(Čistý) matematik: To je transcendentní funkce.
Numerický matematik: Numerická integrace dá přibližný výsledek s požadovanou přesností.
(Ve skutečnosti i exponenciální funkce je počítána jen numericky, procesor sám umí jen 4 základní početní úkony.)
Pro rychlejší opakovaný výpočet si připravíme tabulku Gaussova integrálu.

1.1.3 Aproximace funkcí v elektrotechnice

Úloha: Ze známého napětí baterie (v mobilu, v počítači) odhadněte zbývající kapacitu.
Vycházíme z konečně mnoha hodnot, ale aproximaci chceme použít na celém intervalu.
Zde navíc chceme, aby byla monotónní.

Úloha: Máme nakreslit V-A charakteristiku diody na základě naměřených dat:



Úloha: Digitální obrázek zvětšíme, popř. otočíme. Změní se počet pixelů, popř. i jejich orientace.

Motivační úlohy na aproximaci

- **Úloha 1**: Závislost směnného kursu na čase na základě údajů z burzy.
- **Úloha 2**: Rychlý odhad Gaussova integrálu (distribuční funkce normálního rozdělení) s využitím tabulkových hodnot.
- **Úloha 3**: V-A charakteristika diody na základě naměřených hodnot.
- **Úloha 4**: Teploty naměřené na meteorologické stanici.

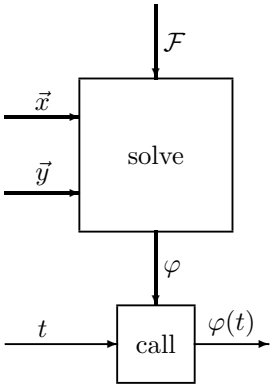
Data jsou vždy diskrétní (konečná).
Rozlišujeme:

1. aproximaci spojité funkce pomocí konečných dat (1. zápočtová úloha),
2. nalezení funkce s malým počtem parametrů, která přibližně odpovídá datům (2. zápočtová úloha).

1.1.4 Základní úloha aproximace

Dáno:
vektor $\vec{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{R}^n$ určující n (různých) **uzlových bodů**,
vektor $\vec{y} = (y_0, \ldots, y_{n-1}) \in \mathbb{R}^n$ požadovaných hodnot v uzlových bodech,
množina funkcí $\mathcal{F}$, které smíme použít k aproximaci (definovaných alespoň v bodech $x_0, \ldots, x_{n-1}$).
Úkol: vybrat funkci $\varphi \in \mathcal{F}$ tak, aby vektor $\varphi(\vec{x}) = \big(\varphi(x_0), \ldots, \varphi(x_{n-1})\big) \in \mathbb{R}^n$ byl „co nejbližší“ danému vektoru $\vec{y}$.

Blokové schema obecné aproximace



Předpoklad: $\mathcal{F}$ je lineární obal k *známých*, tzv. **aproximačních funkcí** $\varphi_0, \ldots, \varphi_{k-1}$, $k \leq n$:

$$\mathcal{F} = \mathrm{Lin}\{\varphi_0, \ldots, \varphi_{k-1}\} = \left\{ \sum_{j<k} c_j \, \varphi_j : c_j \in \mathbb{R} \right\}.$$

Zbývá určit vektor reálných koeficientů $\vec{c} = (c_0, \ldots, c_{k-1})$ lineární kombinace

$$\varphi = \sum_{j<k} c_j \, \varphi_j \, .$$

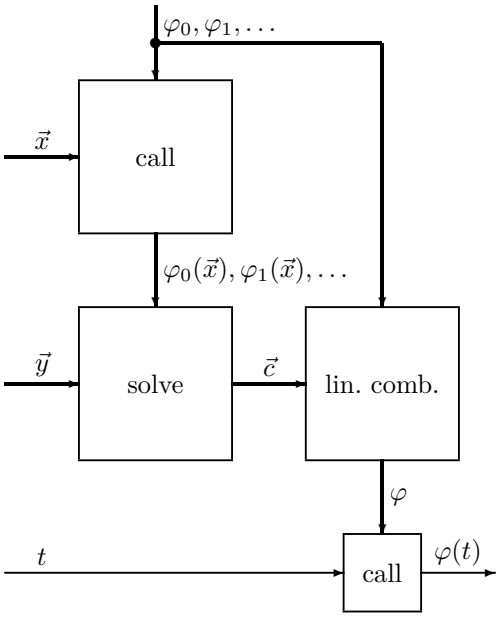
Vektorová formulace aproximační úlohy

Aproximační funkce jsou reprezentovány vektory $\varphi_j(\vec{x}) = (\varphi_j(x_0), \ldots, \varphi_j(x_{n-1}))$, $j = 0, \ldots, k-1$. (Žádné jiné jejich hodnoty do aproximační úlohy nevstupují.)
Pro výslednou aproximaci platí

$$\varphi(\vec{x}) = \sum_{j<k} c_j \, \varphi_j(\vec{x}) \, ,$$

neboli k $\vec{y}$ hledáme „co nejbližší“ vektor $\varphi(\vec{x})$ z lineárního obalu **známých** vektorů $\varphi_0(\vec{x}), \ldots, \varphi_{k-1}(\vec{x})$.

Blokové schema obecné aproximace lineární kombinací



1.2 Interpolace

Speciální případ aproximace, kdy požadujeme **přesnou shodu** v uzlových bodech.

Dáno:
vektor $\vec{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{R}^n$ určující n (různých) uzlových bodů,
vektor $\vec{y} = (y_0, \ldots, y_{n-1}) \in \mathbb{R}^n$ požadovaných hodnot v uzlových bodech,
množina funkcí $\mathcal{F}$, které smíme použít k aproximaci (definovaných alespoň v bodech $x_0, \ldots, x_{n-1}$).

Hledáme: $\varphi \in \mathcal{F}$ splňující $\varphi(\vec{x}) = \vec{y}$, tj.

$$\varphi(x_i) = y_i\,, \qquad i = 0, \ldots, n-1\,.$$

Předpoklad: $\mathcal{F} = \mathrm{Lin}\{\varphi_0, \ldots, \varphi_{k-1}\} = \left\{ \sum_{j<k} c_j\, \varphi_j : c_j \in \mathbb{R} \right\}.$

Ve vektorové formulaci: $\varphi(\vec{x}) = \sum_{j<k} c_j\, \varphi_j(\vec{x}) = \vec{y}\,.$

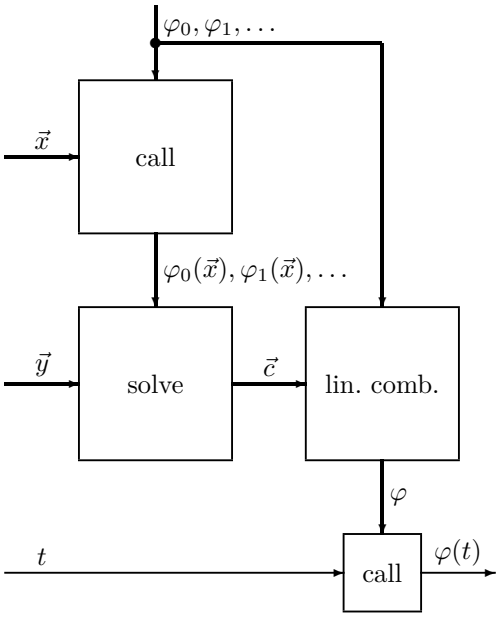
1.2.1 Prostá interpolace

Dosazením dostáváme

$$\sum_{j<k} c_j\, \varphi_j(x_i) = y_i, \qquad i = 0, \ldots, n-1\,,$$

což je soustava k lineárních rovnic pro neznámé $c_0, \ldots, c_{n-1}$.
Pro jednoznačnost řešení potřebujeme $k = n$ a navíc, aby aritmetické **vektory** $\varphi_j(\vec{x}) = (\varphi_j(x_0), \ldots, \varphi_j(x_{n-1}))$, $j = 0, \ldots, n-1$, byly lineárně nezávislé. (To lze pokazit, i když **funkce** $\varphi_0, \ldots, \varphi_{n-1}$ jsou lineárně nezávislé. Záleží i na volbě uzlových bodů.)
Složitost výpočtu $\propto n^3$, u speciálních úloh dosáhneme menší.

Blokové schema obecné prosté interpolace



1.3 Interpolace polynomem

Interpolujeme polynomem stupně $< n$, neboli $\leq n-1$.
Má právě n koeficientů, což potřebujeme pro existenci a jednoznačnost řešení.
Za bázi prostoru těchto polynomů můžeme volit $\varphi_j(t) = t^j, \quad j = 0, \ldots, n-1$.
Jiná báze může být výhodnější.

Výhody interpolace polynomem

- Jediné, co počítač umí rychle, jsou 4 základní aritmetické operace, tedy kromě polynomů už jen racionální funkce.
- Polynomy lze snadno integrovat, derivovat...
- Řešitelnost:

Věta 1.1 *Interpolační úlohu s n různými uzlovými body řeší právě jeden polynom stupně $< n$.*

- Univerzálnost:

Věta 1.2 (Weierstrassova) *Nechť f je spojitá funkce na uzavřeném intervalu I a nechť je dáno číslo $\varepsilon > 0$. Pak existuje polynom φ takový, že*

$$\forall t \in I : |f(t) - \varphi(t)| \leq \varepsilon\,.$$

Nevýhody interpolace polynomem

- Weierstrassova věta neříká nic o potřebném stupni polynomu, takže výsledek nemusí být použitelný.
- Velmi málo skutečných závislostí je polynomiálních.

Lze řešit prostou interpolací, ale se složitostí $\propto n^3$; jde to lépe.

x_i	g_i
-2	4
0	2
2	3
4	3

$\varphi(t) = c_3 t^3 + c_2 t^2 + c_1 t + c_0$

$-8c_3 + 4c_2 - 2c_1 + c_0 = 4$

$c_0 = 2$

$8c_3 + 4c_2 + 2c_1 + c_0 = 1$

$64c_3 + 16c_2 + 4c_1 + c_0 = 1$

„Vodorovná linearita“ – interpolace lineárním polynomem

Dvěma body $(x_i, y_i), (x_j, y_j)$ prokládáme lineární funkci φ. Její hodnota v bodě t je váženým průměrem y_i, y_j, kde váhy jsou nepřímo úměrné vzdálenostem $t - x_i$, $x_j - t$:

$$x_i \mapsto \varphi(x_i) = y_i\,,$$
$$x_j \mapsto \varphi(x_j) = y_j\,,$$
$$t = \frac{\frac{x_i}{t-x_i} + \frac{x_j}{x_j-t}}{\frac{1}{t-x_i} + \frac{1}{x_j-t}} \mapsto \varphi(t) = \frac{\frac{y_i}{t-x_i} + \frac{y_j}{x_j-t}}{\frac{1}{t-x_i} + \frac{1}{x_j-t}}\,,$$
$$t = \frac{(x_j - t)\,x_i + (t - x_i)\,x_j}{x_j - x_i} \mapsto \varphi(t) = \frac{(x_j - t)\,y_i + (t - x_i)\,y_j}{x_j - x_i}\,. \tag{1}$$

Odvození vyžadovalo $t \notin \{x_i, x_j\}$, ale výsledek platí vždy.

1.3.1 Lagrangeova konstrukce interpolačního polynomu

1. Pro $j = 0, \ldots, n-1$ vyřešíme speciální případ, kdy j-tá složka vektoru $\vec{y}$ je jednotková, ostatní nulové; výsledkem je polynom ϱ_j,

$$\varrho_j(x_i) = \delta_{ij} = \begin{cases} 1 & \text{pro } i = j, \\ 0 & \text{pro } i \neq j. \end{cases}$$

(δ_{ij} je tzv. **Kroneckerovo delta**)

Polynom ϱ_j stupně $\leq n-1$ má $n-1$ kořenů $x_0, \ldots, x_{j-1}, x_{j+1}, \ldots, x_{n-1}$;

$$\begin{aligned} \varrho_j(t) &= e_j (t - x_0) \ldots (t - x_{j-1})(t - x_{j+1}) \ldots (t - x_{n-1}) \\ &= e_j \prod_{i<n, i\neq j} (t - x_i)\,, \end{aligned}$$

kde e_j určíme ze vztahu $\varrho_j(x_j) = 1$:

$$e_j = \frac{1}{\prod\limits_{i<n, i\neq j} (x_j - x_i)}\,, \qquad \varrho_j(t) = \frac{\prod\limits_{i<n, i\neq j} (t - x_i)}{\prod\limits_{i<n, i\neq j} (x_j - x_i)}\,.$$

2. Obecné řešení úlohy interpolace polynomem je lineární kombinace

$$\varphi = \sum_{j<n} y_j\, \varrho_j\,,$$

$$\varphi(t) = \sum_{j<n} y_j\, \varrho_j(t) = \sum_{j<n} y_j \frac{\prod\limits_{i<n, i\neq j} (t - x_i)}{\prod\limits_{i<n, i\neq j} (x_j - x_i)}\,.$$

Kontrola:

$$\varphi(x_i) = \sum_{j<n} y_j\, \varrho_j(x_i) = \sum_{j<n} y_j\, \delta_{ij} = y_i\,.$$

Myšlenka Lagrangeovy konstrukce interpolačního polynomu je použitelná i pro obecnější úlohy.

$\vec{e_0} = (1,0,0,\dots) \mapsto f_0(t)$

$\vec{e_1} = (0,1,0,\dots) \mapsto f_1$

$\vdots$

$\vec{e_{n-1}} = (0,0,\dots 0,1) \mapsto f_{n-1}$

$y = (y_0, \dots y_{n-1}) =$

$= y_0\vec{e_0} + \dots + y_{n-1}\vec{e_{n-1}} \to \varphi = y_0 f_0 + \dots + y_0 f_{n-1}$

x_i	-2	0	2	4
y_i	4	2	3	3
$f_0(x_i)$	1	0	0	0

$f_0(t) = e_0 (t-0)(t-2)(t-4)$

↳ konstanta (ne vektory jako takové)

$e_0 (x_0-0)(x_0-2)(x_0-4) = 1$

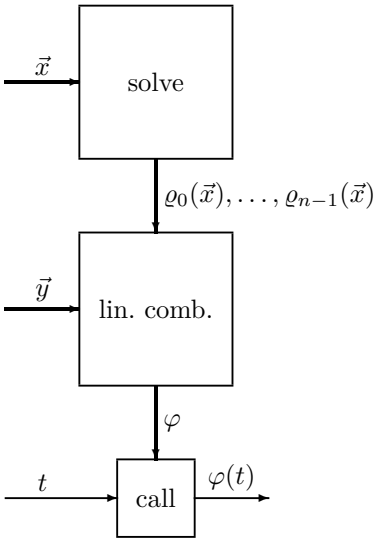
$$\frac{+(t-2)(t-4)}{(x_0)(x_0-2)(x_0-4)} = \frac{\dots}{-48}$$

$-2 \quad -2 \quad -2$

$$\varphi(t) = \frac{-1}{12}t^3 + \frac{3}{8}t^2 + \frac{1}{12}t + 2 =$$

$$((\left(\frac{-1}{12} + \frac{3}{8}\right)t + \frac{1}{12})t + 2$$

Blokové schema Lagrangeovy konstrukce interpolačního polynomu



„Svislá linearita“ aproximační úlohy (v nezávislé proměnné)

Většinou předpokládáme linearitu výstupu (princip superpozice), což zajistí nezávislost řešení na zvoleném lineárním měřítku.
Pak jakékoli řešení aproximační úlohy musí **lineárně** záviset na složkách aritmetického vektoru $\vec{y} = (y_0, \ldots, y_{n-1})$.
Vstupní vektor $\vec{y} = (y_0, \ldots, y_{n-1})$ je lineární kombinací vektorů **standardní báze**,

$$\vec{y} = y_0 \underbrace{(1, 0, \ldots, 0)}_{\vec{e}_0} + y_1 \underbrace{(0, 1, 0, \ldots, 0)}_{\vec{e}_1} + \ldots + y_{n-1} \underbrace{(0, \ldots, 0, 1)}_{\vec{e}_{n-1}} = \sum_{j<n} y_j \, \vec{e}_j \,.$$

Výsledná aproximace bude lineární kombinací (se stejnými koeficienty $y_0, \ldots, y_{n-1}$) funkcí ϱ_j, které řeší aproximační úlohu se vstupy $\vec{e}_j$,

$$\begin{aligned}
\vec{e}_0 &= (1, 0, 0, \ldots, 0) && \mapsto \varrho_0 \,, \\
\vec{e}_1 &= (0, 1, 0, \ldots, 0) && \mapsto \varrho_1 \,, \\
& && \ldots \\
\vec{e}_{n-1} &= (0, 0, \ldots, 0, 1) && \mapsto \varrho_{n-1} \,, \\
\vec{y} &= \sum_{j<n} y_j \, \vec{e}_j && \mapsto \sum_{j<n} y_j \, \varrho_j = \varphi \,.
\end{aligned}$$

Funkce $\varrho_0, \varrho_1, \ldots, \varrho_{n-1}$ nám poskytují plnou informaci o řešení úlohy pro libovolné $\vec{y} = (y_0, \ldots, y_{n-1})$.
Poznámka: Pro interpolaci jsou $\vec{e}_0, \ldots, \vec{e}_{n-1}$ vektory hodnot funkcí $\varrho_0, \ldots, \varrho_{n-1}$,

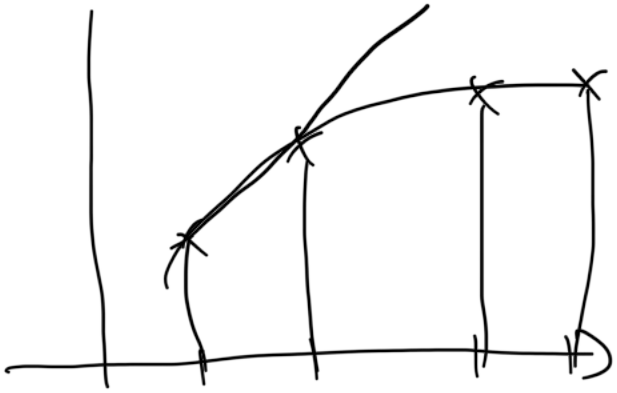
$$\vec{e}_j = (\varrho_j(x_0), \ldots, \varrho_j(x_{n-1})) \,.$$

Pro obecnou aproximaci to tak být nemusí, ale uvedené důsledky linearity zůstávají v platnosti.

Nedostatky Lagrangeovy konstrukce interpolačního polynomu

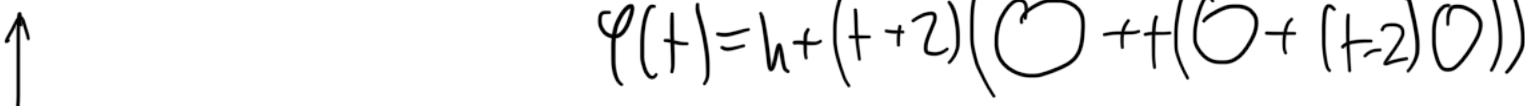
Složitost $\propto n^2$, a to i pro výpočet jedné hodnoty, pokud výsledek neupravíme.
Při aproximaci vývoje na burze dostáváme průběžně nová data.
Musíme kvůli tomu počítat vše znova?
(Některé mezivýsledky lze použít, pokud jsme si je nezapomněli zaznamenat.)
Lze však vyjít z předchozího výsledku a ten jen opravit o určitý polynom:

x_i	-2	0	2	4
y_i	4	2	2	3

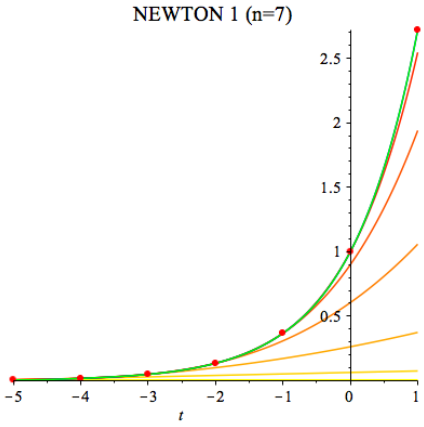


$$\varphi(t) = 4 + (-1)\cdot(t-2) + \left(\tfrac{3}{8}\right)(t+2)(t-2) + \left(-\tfrac{1}{12}\right)(t+2)(t-2)$$

$$= 4 + (t+2)\left(-1 + \left(\tfrac{3}{8} - \tfrac{1}{12}(t-2)\right)\right)$$



1.3.2 Newtonova konstrukce interpolačního polynomu



Jedná se stále o stejný polynom.
Konstantní polynom $d_0 = y_0$ má správnou hodnotu v uzlovém bodě x_0.
Hledaný polynom φ dostaneme přičtením vhodného polynomu ω_0 nulového v x_0: $t \mapsto (t - x_0)\,\omega_0(t)$, kde ω_0 je polynom stupně $\leq n - 2$:

$$
\begin{array}{llll}
\varphi(t) = d_0 + (t - x_0)\,\omega_0(t)\,, & d_0 = y_0\,, & \omega_0(x_i) = \dfrac{y_i - d_0}{x_i - x_0}\,, & i > 0\,,\\
\omega_0(t) = d_1 + (t - x_1)\,\omega_1(t)\,, & d_1 = \omega_0(x_1)\,, & \omega_1(x_i) = \dfrac{\omega_0(x_i) - d_1}{x_i - x_1}\,, & i > 1\,,\\
\omega_1(t) = d_2 + (t - x_2)\,\omega_2(t)\,, & d_2 = \omega_1(x_2)\,, & \omega_2(x_i) = \dfrac{\omega_1(x_i) - d_2}{x_i - x_2}\,, & i > 2\,,\\
\omega_2(t) = d_3 + (t - x_3)\,\omega_3(t)\,, & d_3 = \omega_2(x_3)\,, & \omega_3(x_i) = \dfrac{\omega_2(x_i) - d_3}{x_i - x_3}\,, & i > 3\,,\\
\ldots & & & \\
\omega_{n-3}(t) = d_{n-2} + (t - x_{n-2})\,\omega_{n-2}(t)\,, & d_{n-2} = \omega_{n-3}(x_{n-2})\,, & \omega_{n-2}(x_i) = \dfrac{\omega_{n-3}(x_i) - d_{n-2}}{x_i - x_{n-2}}\,, & i > n-2\,,\\
\omega_{n-2}(t) = d_{n-1} = \omega_{n-2}(x_{n-1})\,. & & &
\end{array}
$$

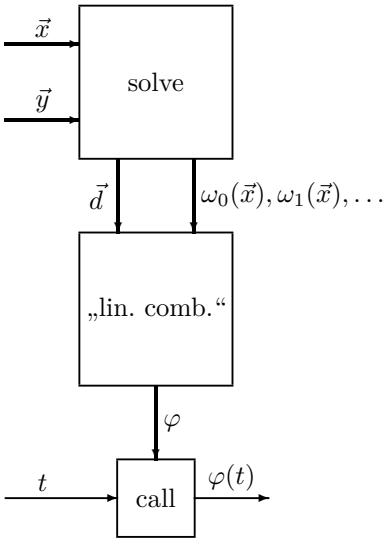
Zpětným dosazením dostaneme

$$
\begin{aligned}
\varphi(t) \quad = \quad & d_0 + (t - x_0) \cdot [d_1 + (t - x_1) \cdot [d_2 + \\
& + \ldots (t - x_{n-3}) \cdot [d_{n-2} + (t - x_{n-2}) \cdot d_{n-1}] \ldots]]\,.
\end{aligned}
$$

$$
\begin{aligned}
\varphi(t) \quad &= \quad d_0 + (t - x_0)\,d_1 + (t - x_0)\,(t - x_1)\,d_2 + \ldots + d_{n-1} \prod_{i<n-1} (t - x_i)\\
&= \quad \sum_{j<n} d_j \prod_{i<j} (t - x_i)\,.
\end{aligned}
$$

Složitost výpočtu koeficientů $\propto n^2$, výpočtu jedné funkční hodnoty *podle prvního, neroznásobeného vzorce* $\propto n$.

Blokové schema Newtonovy konstrukce interpolačního polynomu



1.3.3 Nevillův algoritmus

Nezajímají nás koeficienty, pouze hodnota interpolačního polynomu pro jediný argument t.

Jedním bodem (x_i, y_i) proložíme konstantní polynom y_i . Dva body (x_i, y_i), (x_{i+1}, y_{i+1}) lineárně interpolujeme, hodnota v t je dle (1):

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

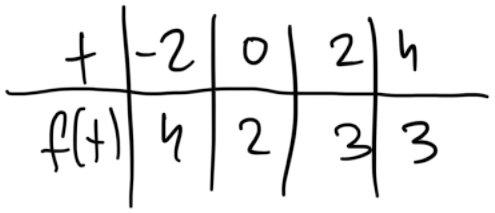
což je správně pro $t \in \{x_i, x_{i+1}\}$.
Rekurentní postup: Obecněji pro $j < k$ označme $y_{j..k}$ hodnotu v bodě t interpolačního polynomu proloženého body $(x_j, y_j), \ldots, (x_k, y_k)$. Závisí na počáteční volbě t; pro $t \in \{x_j, \ldots, x_k\}$ by byl $y_{j..k}$ správný výsledek.
Speciálně:

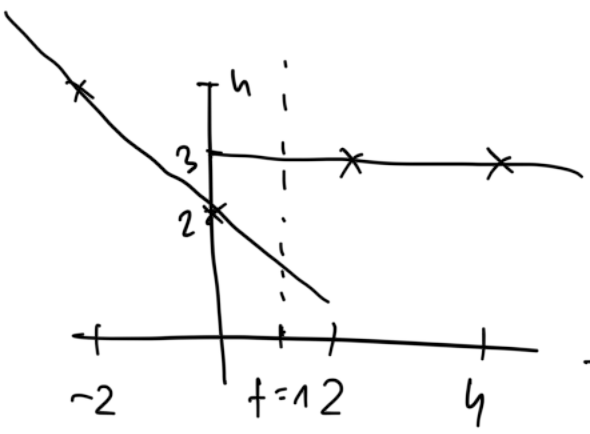
$y_{i+1..i+m}$ by vyšlo správně pro $t \in \{x_{i+1}, \ldots, x_{i+m-1}\}$ a pro $t = x_{i+m}$,
$y_{i..i+m-1}$ by vyšlo správně pro $t \in \{x_{i+1}, \ldots, x_{i+m-1}\}$ a pro $t = x_i$,
$y_{i..i+m}$ by vyšlo správně pro $t \in \{x_{i+1}, \ldots, x_{i+m-1}\}$ a pro $t = x_i$, $t = x_{i+m}$;

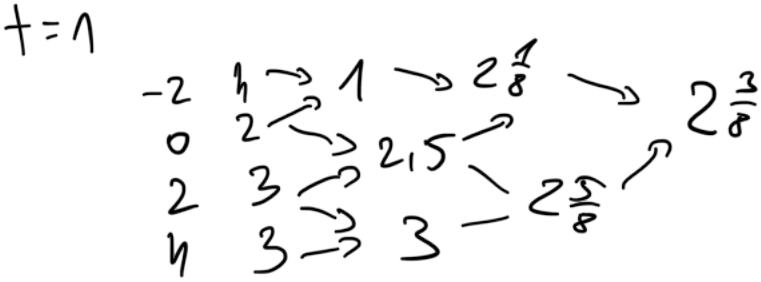
dostaneme ho lineární interpolací mezi x_i a x_{i+m} dle (1):

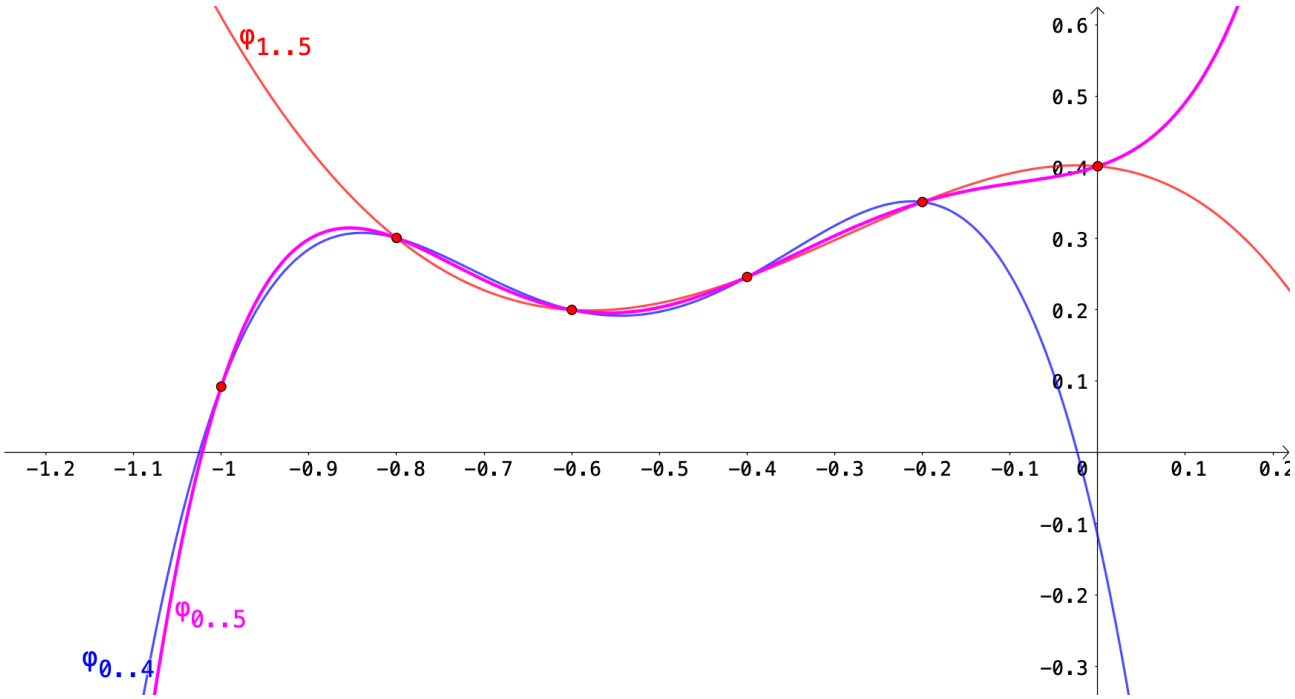
$$y_{i..i+m} = \frac{(x_{i+m} - t)\, y_{i..i+m-1} + (t - x_i)\, y_{i+1..i+m}}{x_{i+m} - x_i} .$$

Počítáme pro všechna $i = 0, \ldots, n-1-m$ (vnitřní cyklus); $m = 1, \ldots, n-1$ (vnější cyklus);
$y_{0..n-1}$ je výsledek. Složitost $\propto n^2$.

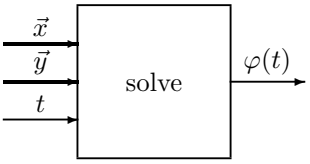








Blokové schema Nevillova algoritmu



Numerické chyby Nevillova algoritmu lze dále omezit, budeme-li místo hodnot interpolačních polynomů počítat s jejich rozdíly,

$$\begin{aligned} C_{i,m} &= y_{i..i+m} - y_{i..i+m-1}\,, \\ D_{i,m} &= y_{i..i+m} - y_{i+1..i+m}\,, \end{aligned}$$

které lze počítat podle rekurentních vzorců

$$\begin{aligned} C_{i,m} &= \frac{(t - x_i)\,(C_{i+1,m-1} - D_{i,m-1})}{x_{i+m} - x_i}\,, \\ D_{i,m} &= \frac{(t - x_{i+m})\,(C_{i+1,m-1} - D_{i,m-1})}{x_{i+m} - x_i} \end{aligned}$$

a výsledek např. jako

$$y_{0..n-1} = y_0 + \sum_{m=1}^{n-1} C_{0,m}\,.$$

1.3.4 Chyba aproximace interpolačním polynomem

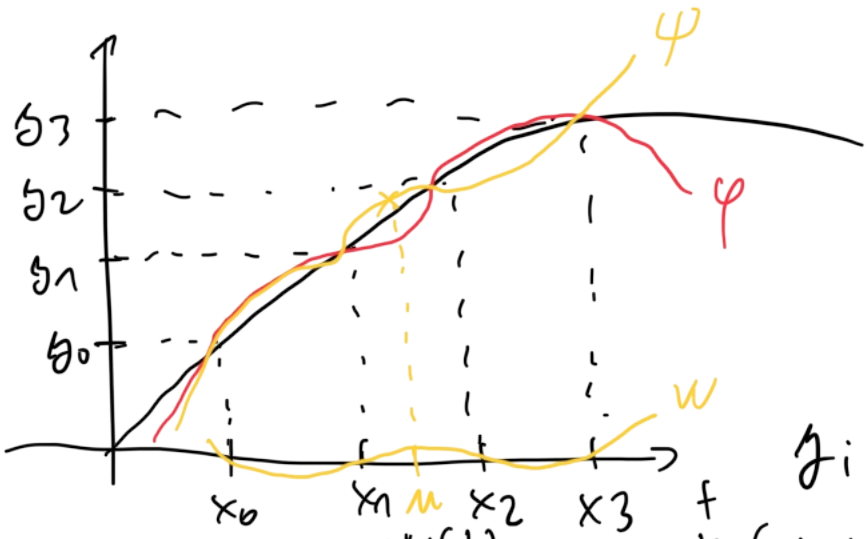
V uzlových bodech chyba metody nulová, pouze chyba zaokrouhlovací.

Chyba metody v ostatních bodech, nahrazujeme-li funkci f interpolačním polynomem.

Odvození chyby aproximace interpolačním polynomem

Chyba metody $f - \varphi$ je v uzlových bodech nulová, zkoumáme její hodnotu $f(u) - \varphi(u)$ v bodě $u \notin \{x_0, \ldots, x_{n-1}\}$. V duchu Newtonovy konstrukce najdeme interpolační polynom ψ_u stupně n, který správně interpoluje i v bodě u:

chyba interpolace



$y_i = f(x_i)$

$$\psi_u(t) = \varphi(t) + P_u \overbrace{\prod_{i<n}(t-x_i)}^{w(t)} = t^n + (\text{pol. st.} < n)$$

$$(t^n)^{(n)} = (n t^{n-1})^{(n-1)} = n(n-1)\ldots = n!$$

$$\psi_u^{(n)}(t) = \underbrace{\varphi^{(n)}(t)}_{0} + P_u \cdot n!$$

$$f(u) = \psi(u) = \varphi(u) + P_u W(u) \Rightarrow P_u = \frac{\boxed{f(u)-\varphi(u)}}{W(u)}$$

$f-\psi$ má kořeny $x_0, \ldots, x_{n-1}, u \ldots (n+1)$ kořenů

$(f-\psi)'$ má $> n$ kořenů v int. obs.

$(f-\psi)''$ má $> (n-1)$

$\vdots$

$(f-\psi)^{(n)}$ má > 1 kořen ξ_u

$$0 = f^{(n)}(\xi_u) - \psi^{(n)}(\xi_u) = f^{(n)}(\xi_u) - P_u n! =$$

$$= f^{(n)}(\xi_u) - \frac{f(u)-\varphi(u)}{W(u)} n! =$$

$$= |f(u)-\varphi(u)| = \frac{|f^{(n)}\xi_u|}{n!} |W(u)| \leq \frac{M_n}{n!} |W(u)|$$

k φ přičteme „vhodný násobek“ polynomu W stupně n s uzlovými body $x_0, \dots, x_{n-1}$,

$$\psi_u = \varphi + P_u \underbrace{\prod_{i<n} (t - x_i)}_{W(t)} .$$

(Polynom ψ_u závisí na volbě bodu u.)
„Vhodný násobek“ P_u splňuje $\psi_u(u) = f(u)$, takže provedená oprava v bodě u je právě chyba interpolace polynomem φ:

$$f(u) - \varphi(u) = \psi_u(u) - \varphi(u) = P_u\, W(u)\,,$$
$$P_u = \frac{f(u) - \varphi(u)}{W(u)}\,.$$

Funkce f, ψ_u jsou si „blízké“, máme
$\geq n+1$ kořenů $u, x_0, \dots, x_{n-1}$ funkce $f - \psi_u$, mezi každými dvěma (aspoň jeden) kořen derivace,
$\geq n$ kořenů funkce $(f - \psi_u)'$,
$\geq n-1$ kořenů funkce $(f - \psi_u)''$,
...
≥ 1 kořen funkce $(f - \psi_u)^{(n)} = f^{(n)} - \psi_u^{(n)}$, označme jej ξ_u (závisí na u).
(Potřebovali jsme spojité derivace do řádu n.) Budeme potřebovat n-tou derivaci polynomu

$$\psi_u(t) = \varphi(t) + P_u \prod_{i<n} (t - x_i) = \underbrace{\varphi(t)}_{\text{polynom stupně}<n} + P_u \big(t^n + \underbrace{\cdots}_{\text{polynom stupně}<n}\big)\,,$$
$$\psi_u^{(n)}(t) = P_u\, (t^n)^{(n)} = P_u\, n!\,.$$

Využijeme

$$0 = f^{(n)}(\xi_u) - \psi_u^{(n)}(\xi_u) = f^{(n)}(\xi_u) - \overbrace{\frac{f(u) - \varphi(u)}{W(u)}}^{P_u} n!\,,$$
$$f(u) - \varphi(u) = \frac{f^{(n)}(\xi_u)}{n!}\, W(u)\,,$$
$$|f(u) - \varphi(u)| = \frac{|f^{(n)}(\xi_u)|}{n!}\, |W(u)|\,,$$

kde $|f^{(n)}(\xi_u)|$ nahradíme horním odhadem na uzavřeném intervalu $I \supseteq \{u, x_0, \dots, x_{n-1}\}$: $\quad M_n \geq \max_{t \in I} |f^{(n)}(t)|$,

$$|f(u) - \varphi(u)| \leq \frac{M_n}{n!}\, |W(u)|\,.$$

Pomocí horního odhadu $\overline{w} \geq \max_{t \in I} |W(t)|$ dostaneme odhad chyby nezávislý na u:

$$|f(u) - \varphi(u)| \leq \frac{M_n}{n!}\, \overline{w}\,.$$

Předpoklad: f má spojité derivace do řádu n na *uzavřeném* intervalu $I \supseteq \{u, x_0, \dots, x_{n-1}\}$.

Co můžeme ovlivnit

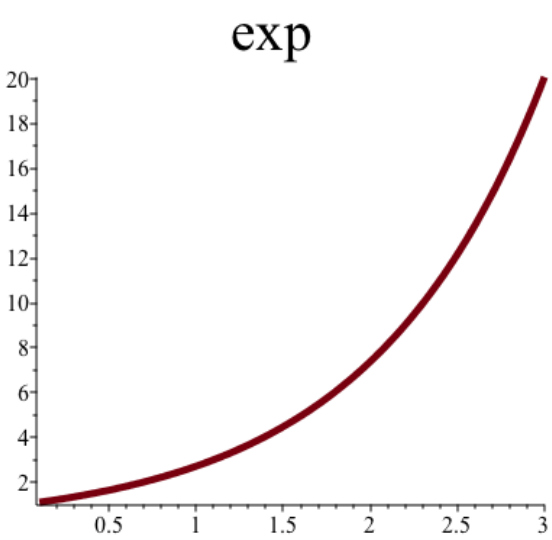
1. Počet uzlových bodů $\implies$ koeficient $\dfrac{1}{n!}$ ☺, ale i M_n ☹.

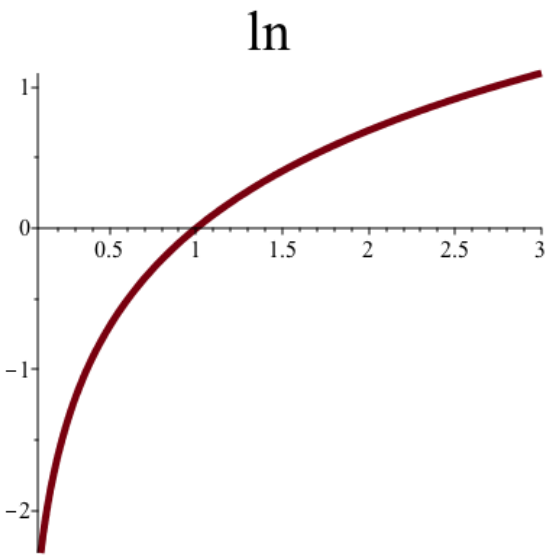
2. Interval I, ale musí obsahovat $\{u, x_0, \dots, x_{n-1}\}$, to lze jen pokazit, když u bude „daleko“ ☹; rozlišujeme:

   - **extrapolaci**, kdy aproximujeme mimo interval $\langle \min_i x_i, \max_i x_i \rangle$,
   - **interpolaci (v užším smyslu)**, kdy aproximujeme na intervalu $\langle \min_i x_i, \max_i x_i \rangle$.

3. Rozložení uzlových bodů $\implies \overline{w}$ ☺.

Příklad:





Ad 1. Počet uzlových bodů

$I = \langle 1/2, 2 \rangle$

	f	exp	ln
	$\lvert f^{(n)}(u)\rvert$	$\exp(u)$	$\frac{(n-1)!}{u^n}$
	argmax	2	$1/2$
n	$\frac{M_n}{n!}$	$\frac{\exp(2)}{n!}$	$\frac{2^n}{n}$
2	$\frac{M_2}{2!}$	3.7	2
4	$\frac{M_4}{4!}$	0.31	4
8	$\frac{M_8}{8!}$	$1.8 \cdot 10^{-4}$	32
16	$\frac{M_{16}}{16!}$	$3.5 \cdot 10^{-13}$	4096
		☺	☹

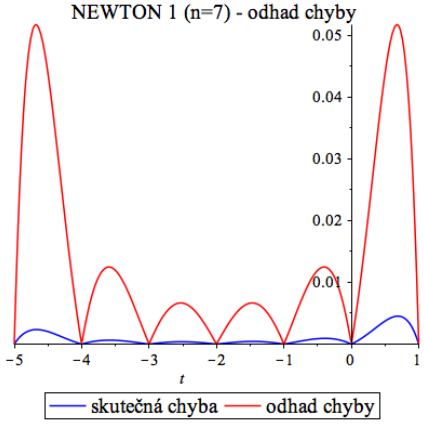
Ad 2. Extrapolace

$I = \langle 1/10, 3 \rangle$

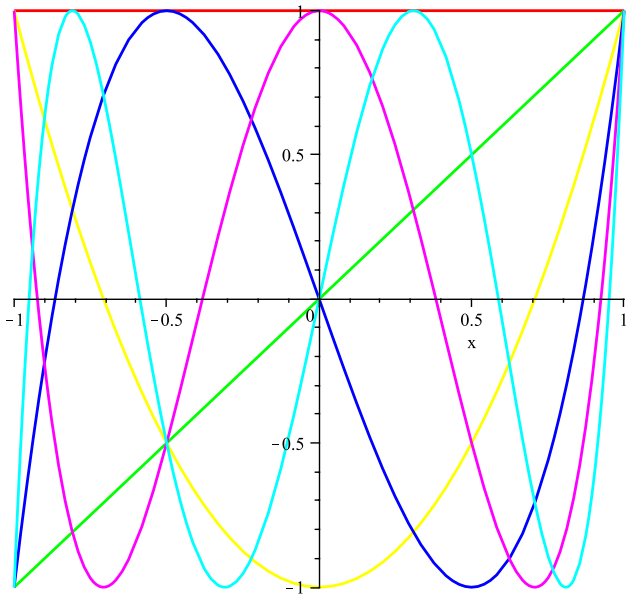
	f	exp	ln
	$\lvert f^{(n)}(u) \rvert$	$\exp(u)$	$\frac{(n-1)!}{u^n}$
	argmax	3	$1/10$
n	$\frac{M_n}{n!}$	$\frac{\exp(3)}{n!}$	$\frac{10^n}{n}$
2	$\frac{M_2}{2!}$	10	50
4	$\frac{M_4}{4!}$	0.84	2500
8	$\frac{M_8}{8!}$	$5 \cdot 10^{-4}$	$1.25 \cdot 10^7$
16	$\frac{M_{16}}{16!}$	10^{-12}	$6.25 \cdot 10^{14}$
		☺	☹☹

Zbývá:
Ad 3. Rozložení uzlových bodů

Chyba aproximace interpolačním polynomem



1.3.5 Čebyševovy polynomy



$$\gamma_j(t) = \cos(j \arccos t)\,, \qquad j = 0, 1, 2, \ldots$$

Počítají se z rekurentního vztahu

$$\begin{aligned}
\gamma_0(t) &= 1 = \cos(0 \arccos t), \\
\gamma_1(t) &= t = \cos(1 \arccos t), \\
\gamma_j(t) &= 2\,t\,\gamma_{j-1}(t) - \gamma_{j-2}(t), \qquad j \geq 2\,,
\end{aligned}$$

Dh:

$$\begin{aligned}
\cos\alpha + \cos\beta &= 2\,\cos\frac{\alpha-\beta}{2}\,\cos\frac{\alpha+\beta}{2}\,, \quad \alpha := j\theta\,, \quad \beta := (j-2)\theta\,, \\
\cos j\theta + \cos(j-2)\theta &= 2\,\cos\theta\,\cos(j-1)\theta\,, \\
\cos j\theta &= 2\,\underbrace{\cos\theta}_{t}\,\cos(j-1)\theta - \cos(j-2)\theta\,, \quad \theta := \arccos t\,, \\
\underbrace{\cos(j \arccos t)}_{\gamma_j(t)} &= 2t\,\underbrace{(\cos(j-1)\arccos t)}_{\gamma_{j-1}(t)} - \underbrace{\cos((j-2)\arccos t)}_{\gamma_{j-2}(t)}\,.
\end{aligned}$$

Obor hodnot funkcí γ_j, $j \geq 1$, na intervalu $\langle -1, 1\rangle$ je $\langle -1, 1\rangle$.
Kořeny $z_0, \ldots, z_{j-1}$ jsou řešení rovnice

$$\begin{aligned}
\cos(j \arccos z_k) &= 0\,, \\
j \arccos z_k &= \frac{\pi}{2} + k\pi\,, \\
\arccos z_k &= \frac{1}{j}\left(\frac{\pi}{2} + k\pi\right) \in \langle 0, \pi\rangle\,, \\
z_k &= \cos\left(\frac{\pi}{j}\left(k + \frac{1}{2}\right)\right)\,, \qquad k = 0, 1, \ldots, j-1\,.
\end{aligned}$$

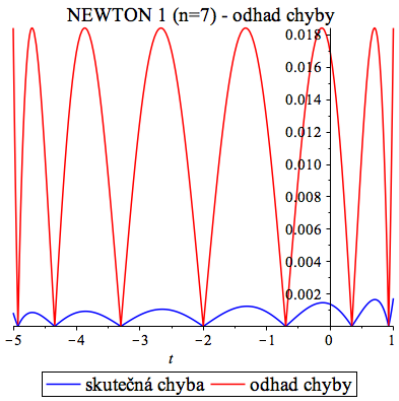
Pro $j = n$ dostáváme doporučené **kosinové** rozdělení uzlových bodů na intervalu $\langle -1, 1\rangle$; obecně na $\langle a, b\rangle$

$$x_k = \frac{b+a}{2} + \frac{b-a}{2}\,z_k = \frac{b+a}{2} + \frac{b-a}{2}\,\cos\frac{\pi(k+\frac{1}{2})}{n}\,, \quad k = 0, \ldots, n-1\,.$$

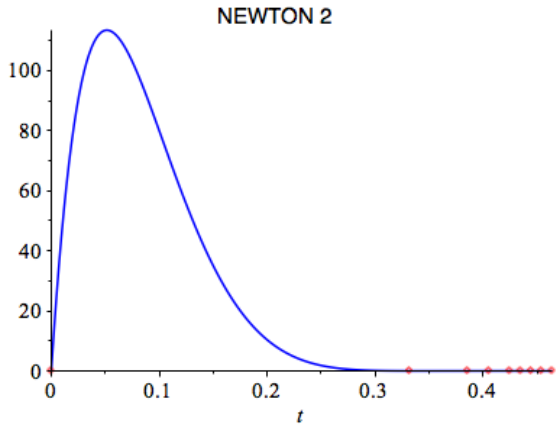
Pokud požadujeme, aby byly uzlové body v krajních bodech intervalu, můžeme vzorec modifikovat:

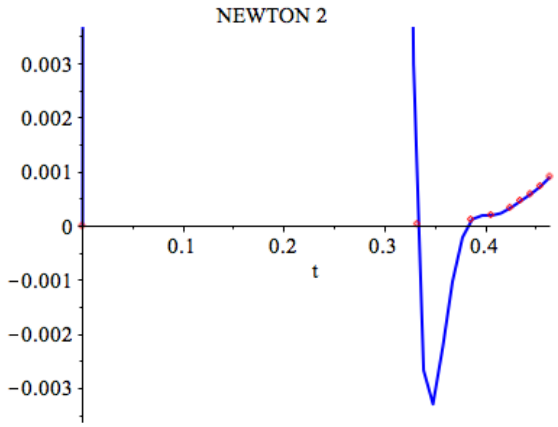
$$x_k = \frac{b+a}{2} + \frac{b-a}{2}\,\cos\frac{\pi\,k}{n-1}, \quad k = 0, \ldots, n-1\,.$$

Chyba aproximace při kosinovém rozdělení uzlových bodů

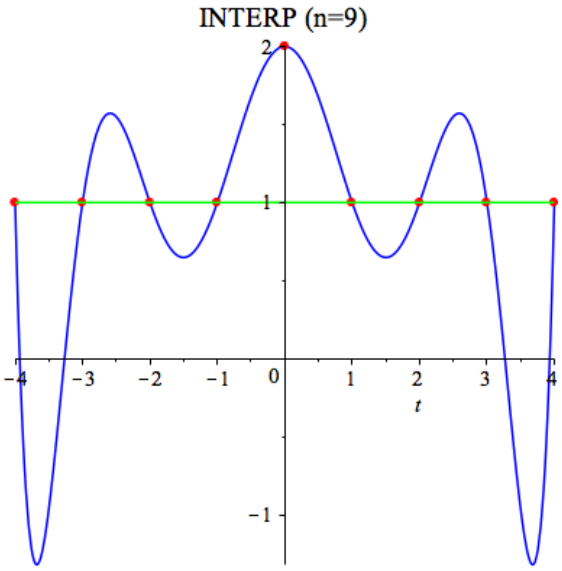


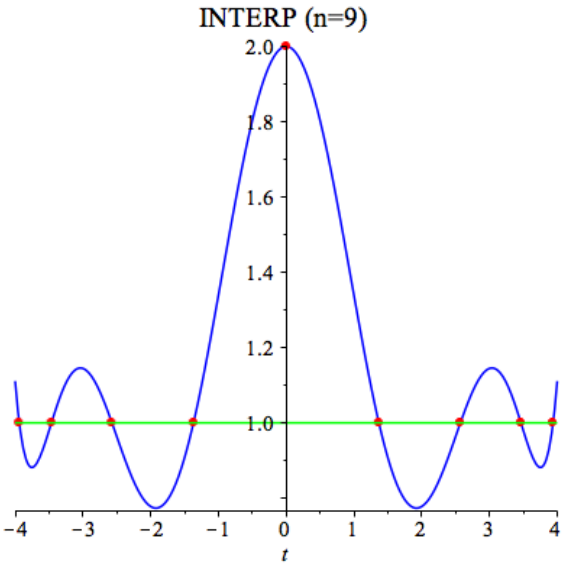
Motivační úloha - interpolační polynom



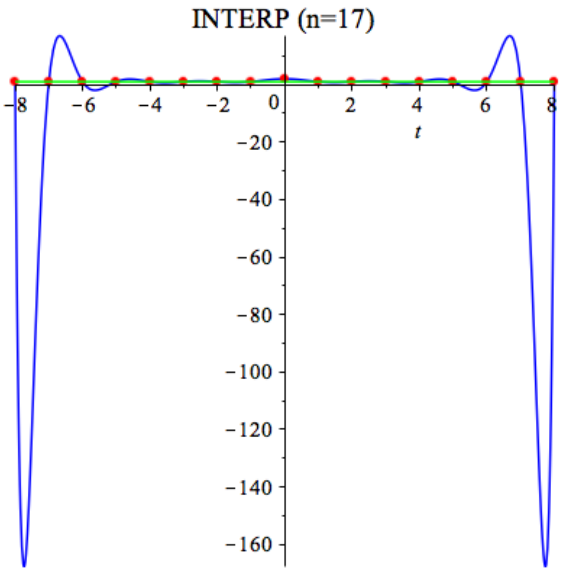


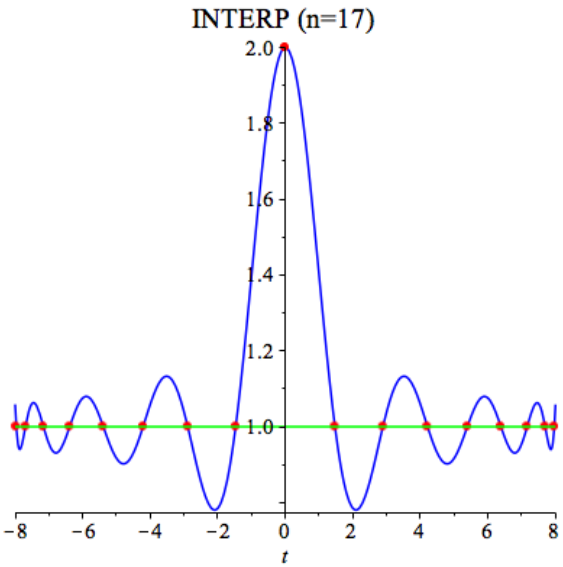
Vliv lokálních změn na interpolační polynom





Vliv lokálních změn na interpolační polynom



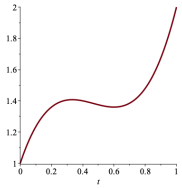


1.3.6 Hermitův interpolační polynom

Příklad: **Dáno:** $x_0 = 0, \quad x_1 = 1; \qquad y_{0,0} = 1, \quad y_{1,0} = 2, \quad y_{0,1} = 3, \quad y_{1,1} = 4.$
Hledáme: polynom $\varphi(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3$, splňující

$$\begin{array}{llll} \varphi(x_0) = y_{0,0}\,, & \varphi(x_1) = y_{1,0}\,, & \varphi'(x_0) = y_{0,1}\,, & \varphi'(x_1) = y_{1,1}\,, \\ \varphi(0) = 1\,, & \varphi(1) = 2\,, & \varphi'(0) = 3\,, & \varphi'(1) = 4\,. \end{array}$$

$$\varphi(t) = 1 + 3t - 7t^2 + 5t^3\,.$$



Jiný postup: Stejně jako u Lagrangeovy konstrukce interpolačního polynomu můžeme sestrojit nejdříve polynomy $\eta, \varrho, \sigma, \tau$ stupně 3, splňující

ψ	$\psi(0)$	$\psi(1)$	$\psi'(0)$	$\psi'(1)$
η	1	0	0	0
ϱ	0	1	0	0
σ	0	0	1	0
τ	0	0	0	1

Výsledek je jejich lineární kombinace se známými koeficienty:

$$\begin{aligned} \varphi &= y_{0,0}\,\eta + y_{1,0}\,\varrho + y_{0,1}\,\sigma + y_{1,1}\,\tau \\ &= 1\eta + 2\varrho + 3\sigma + 4\tau\,. \end{aligned}$$

Polynomy řešící dílčí úlohy:

$\eta(0)$	$\eta(1)$	$\eta'(0)$	$\eta'(1)$
1	0	0	0

η má dvojnásobný kořen 1, je tedy tvaru

$$\eta(t) = (at + b)\,(t-1)^2\,,$$

kde a, b určíme z hodnot v 0:

$$\begin{aligned} \eta(0) = b &= 1 \\ \eta'(0) = a - 2b &= 0 \end{aligned}$$

$$a = 2, \qquad b = 1, \qquad \eta(t) = (2t+1)\,(t-1)^2$$

$\varrho(0)$	$\varrho(1)$	$\varrho'(0)$	$\varrho'(1)$
0	1	0	0

ϱ má dvojnásobný kořen 0, je tedy tvaru

$$\varrho(t) = (a^* t + b^*)\,t^2\,,$$

kde a^*, b^* určíme z hodnot v 1:

$$\begin{aligned} \varrho(1) = a^* + b^* &= 1 \\ \varrho'(1) = 3a^* + 2b^* &= 0 \end{aligned}$$

$$a^* = -2, \qquad b^* = 3, \qquad \varrho(t) = (-2t+3)\,t^2$$

polynomy vedlejší úlohy

$\eta(0)$	$\eta(1)$	$\eta'(0)$	$\eta'(1)$
1	0	0	0

$\eta(t) = (at+b)(t-1)^2$ $\quad \eta'(t) = a(t-1)^2 + (at+b)2(t-1)$

$\eta(0) = b = 1$

$\eta'(0) = a + b\cdot 2(-1) = a - 2b = 0$

$a = 2b = 2$

$\sigma(0)$	$\sigma(1)$	$\sigma'(0)$	$\sigma(1)$
0	0	1	0

$\sigma(t) = c\cdot t(t-1)^2$

$\sigma'(t) = c\cdot\left((t-1)^2 + t\cdot 2(t-1)\right)$

$\sigma'(0) = c = 1$

$\sigma(0)$	$\sigma(1)$	$\sigma'(0)$	$\sigma'(1)$
0	0	1	0

σ má dvojnásobný kořen 1 a jednoduchý kořen 0, je tedy tvaru

$$\sigma(t) = c\,t\,(t-1)^2\,,$$

kde c určíme z derivace v 0:

$$\sigma'(0) = c = 1$$
$$\sigma(t) = t\,(t-1)^2$$

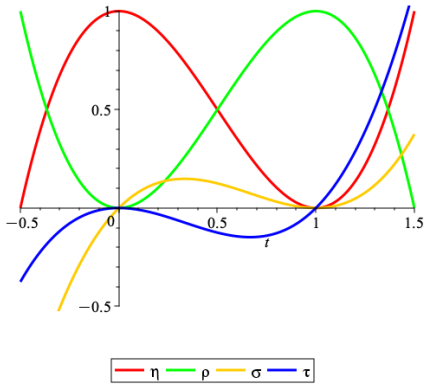
$\tau(0)$	$\tau(1)$	$\tau'(0)$	$\tau'(1)$
0	0	0	1

τ má dvojnásobný kořen 0 a jednoduchý kořen 1, je tedy tvaru

$$\tau(t) = c^*\,t^2\,(t-1)\,,$$

kde c^* určíme z derivace v 1:

$$\tau'(1) = c^* = 1$$
$$\tau(t) = t^2\,(t-1)$$



φ dostaneme jako lineární kombinaci $\eta, \varrho, \sigma, \tau$,

$$\varphi = y_{0,0}\,\eta + y_{1,0}\,\varrho + y_{0,1}\,\sigma + y_{1,1}\,\tau$$

1.3.7 Aproximace Taylorovou řadou

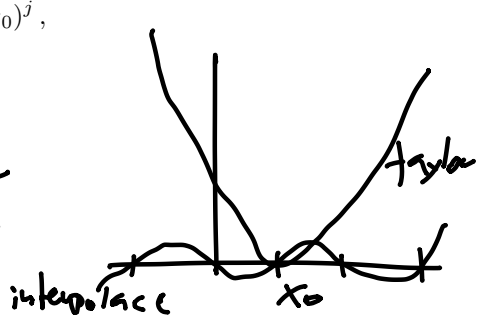
(přesněji Taylorovým polynomem)
Speciální případ Hermitova interpolačního polynomu s jediným uzlovým bodem, v němž je zadáno prvních n derivací (včetně nulté).
Úloha: Dáno:
Uzlový bod x_0 ,
n hodnot $y_{0,0}, y_{0,1}, \ldots, y_{0,n-1} \in \mathbb{R}$.
Hledáme: polynom φ stupně $< n$ takový, že

$$\varphi^{(j)}(x_0) = y_{0,j}\,, \qquad j = 0, \ldots, n-1\,.$$

Řešení je ve tvaru

$$\varphi = \sum_{j<n} y_{0,j}\,\varrho_j\,, \qquad \varrho_j^{(i)}(x_0) = \delta_{ij}\,, \qquad \varrho_j(t) = \frac{1}{j!}\,(t-x_0)^j\,,$$

$e^t = 1 + t + \frac{t^2}{2} + \frac{t^3}{6} + \ldots = \sum_{i=0}^{\infty} \frac{t^i}{i!}$



$$\varphi(t) = \sum_{j<n} \frac{y_{0,j}}{j!} (t - x_0)^j .$$

Pokud jsou $y_{0,0}, y_{0,1}, \ldots, y_{0,n-1}$ hodnoty derivací nějaké funkce f v bodě x_0, tj. $y_{0,j} = f^{(j)}(x_0)$, pak φ je konečná Taylorova řada se středem x_0:

$$\varphi(t) = \sum_{j<n} \frac{f^{(j)}(x_0)}{j!} (t - x_0)^j .$$

Pokud f má na intervalu $I(u, x_0)$ spojitou derivaci řádu n, pak

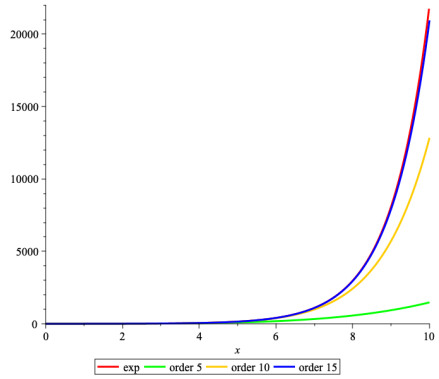
$$f(u) - \varphi(u) = \frac{f^{(n)}(\xi_u)}{n!} (u - x_0)^n ,$$

kde $\xi_u \in I(u, x_0)$,

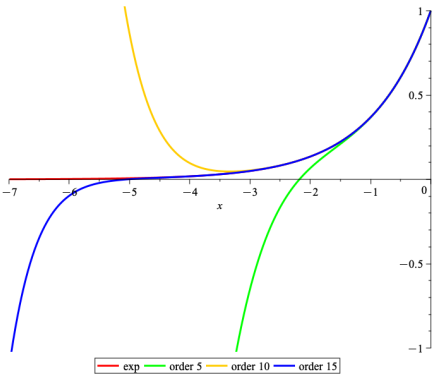
$$|f(u) - \varphi(u)| \leq \frac{M_n}{n!} |(u - x_0)^n| .$$

Jediný rozdíl od chyby interpolačního polynomu je, že polynom $W(u)$ s kořeny $x_0, \ldots, x_{n-1}$ je nahrazen polynomem $(u - x_0)^n$ (stejného stupně) s n-násobným kořenem x_0. Aproximace Taylorovou řadou je velmi přesná v okolí x_0, na úkor chyby ve vzdálenějších bodech.

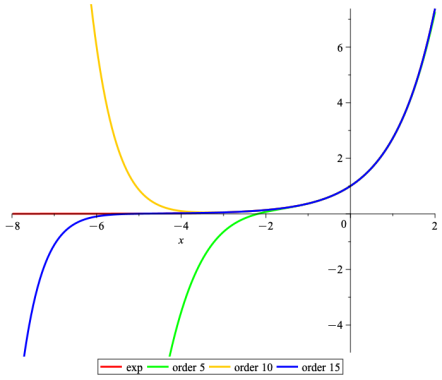
Aproximace Taylorovým polynomem



Aproximace exponenciály (červeně) Taylorovými polynomy stupňů 5, 10, 15 pro kladné argumenty.



Aproximace exponenciály (červeně) Taylorovými polynomy stupňů 5, 10, 15 pro záporné argumenty.



Aproximace exponenciály (červeně) Taylorovými polynomy stupňů 5, 10, 15 pro kladné i záporné argumenty.

1.4 Interpolace spliny

Nevýhoda interpolace polynomem: malá změna vstupní hodnoty v jednom uzlovém bodě může zásadně ovlivnit výsledné hodnoty v místech značně vzdálených.

Spline je funkce po částech polynomiální. (Je dána různými polynomy nízkého stupně na jednotlivých intervalech.)

Nejjednodušším případem je náhrada po částech lineární funkcí, „lomenou čarou“, **lineární spline**.

1.4.1 Kubický spline

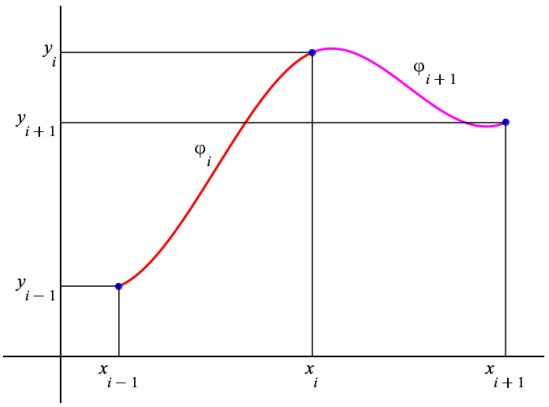
Úloha: Dáno:
vektor $\vec{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{R}^n$ určující n uzlových bodů **vzestupně uspořádaných**,
vektor $\vec{y} = (y_0, \ldots, y_{n-1}) \in \mathbb{R}^n$ požadovaných hodnot v uzlových bodech,
Hledáme: funkci φ, definovanou na intervalu $\langle x_0, x_{n-1}\rangle$, splňující:

- $\varphi(x_i) = y_i, \qquad i = 0, \ldots, n-1$,
- φ se na intervalu $\langle x_{i-1}, x_i\rangle$ shoduje s nějakým polynomem φ_i stupně nejvýše 3, $\qquad i = 1, \ldots, n-1$,
- φ má na intervalu $\langle x_0, x_{n-1}\rangle$ spojitou první a druhou derivaci.

(Toto zadání bude nutné ještě upřesnit.) Spojitost derivací stačí zajistit v bodech $x_1, \ldots, x_{n-2}$:

$$\begin{aligned} \varphi_i'(x_i) &= \varphi_{i+1}'(x_i), \qquad i = 1, \ldots, n-2\,, \\ \varphi_i''(x_i) &= \varphi_{i+1}''(x_i), \qquad i = 1, \ldots, n-2\,. \end{aligned}$$

Předpokládejme, že známe hodnoty $c_i = \varphi'(x_i) = \varphi_i'(x_i) = \varphi_{i+1}'(x_i)$, $i = 0, \ldots, n-1$. Tím bude zajištěna spojitost φ', zbývá zajistit spojitost φ''.



Polynomy φ_i, $i = 1, \dots, n-1$, lze najít stejně jako Hermitův interpolační polynom v předchozí úloze, pouze uzlové body jsou x_{i-1}, x_i místo $0, 1$.

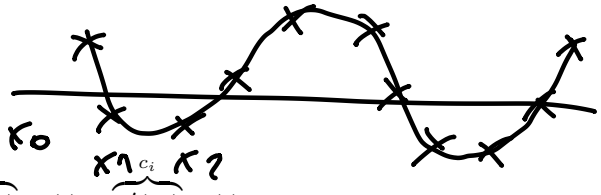
Obecný případ dostaneme lineární transformací $u \in \langle 0, 1\rangle$ na $t \in \langle x_{i-1}, x_i\rangle$:

$$t = x_{i-1} + (x_i - x_{i-1})\, u\,, \qquad u = \frac{t - x_{i-1}}{x_i - x_{i-1}}\,.$$

Na druhých derivacích se to projeví např.

$$\begin{aligned} \eta_i(t) &= \eta(u)\,, \\ \frac{\mathrm{d}\eta_i}{\mathrm{d}t} &= \frac{\mathrm{d}\eta}{\mathrm{d}u}\cdot\frac{\mathrm{d}u}{\mathrm{d}t}\,, \\ \eta_i'(t) &= \frac{\eta'(u)}{x_i - x_{i=1}}\,, \\ \eta_i''(t) &= \frac{\eta''(u)}{(x_i - x_{i=1})^2}\,. \end{aligned}$$

(nekorektní Leibnizova notace)



Na intervalu $\langle x_{i-1}, x_i\rangle$, $i = 1, \dots, n-1$, dostáváme

$$\varphi_i(t) = y_{i-1}\,\eta_i(t) + y_i\,\varrho_i(t) + \overbrace{\varphi'(x_{i-1})}^{c_{i-1}}\,\sigma_i(t) + \overbrace{\varphi'(x_i)}^{c_i}\,\tau_i(t)\,,$$

kde $\eta_i, \varrho_i, \sigma_i, \tau_i$ jsou polynomy stupně nejvýše 3 (určené stejně jako polynomy $\eta, \varrho, \sigma, \tau$ v úloze na Hermitův interpolační polynom).
Můžeme vyčíslit jejich druhé derivace:

$$\begin{aligned} \eta''(0) &= -6\,, & \varrho''(0) &= 6\,, & \sigma''(0) &= -4\,, & \tau''(0) &= -2\,, \\ \eta_{i+1}''(x_i) &= \frac{-6}{(x_{i+1}-x_i)^2}\,, & \varrho_{i+1}''(x_i) &= \frac{6}{(x_{i+1}-x_i)^2}\,, & \sigma_{i+1}''(x_i) &= \frac{-4}{(x_{i+1}-x_i)^2}\,, & \tau_{i+1}''(x_i) &= \frac{-2}{(x_{i+1}-x_i)^2}\,, \\ \eta''(1) &= 6\,, & \varrho''(1) &= -6\,, & \sigma''(1) &= 2\,, & \tau''(1) &= 4\,, \\ \eta_i''(x_i) &= \frac{6}{(x_i-x_{i-1})^2}\,, & \varrho_i''(x_i) &= \frac{-6}{(x_i-x_{i-1})^2}\,, & \sigma_i''(x_i) &= \frac{2}{(x_i-x_{i-1})^2}\,, & \tau_i''(x_i) &= \frac{4}{(x_i-x_{i-1})^2}\,. \end{aligned}$$

Zbývá určit $c_i = \varphi'(x_i)$; pro $i = 1, \dots, n-2$ máme

$$\begin{aligned} & y_{i-1}\,\eta_i''(x_i) + y_i\,\varrho_i''(x_i) + \overbrace{\varphi'(x_{i-1})}^{c_{i-1}}\,\sigma_i''(x_i) + \overbrace{\varphi'(x_i)}^{c_i}\,\tau_i''(x_i) = \\ = \;& y_i\,\eta_{i+1}''(x_i) + y_{i+1}\,\varrho_{i+1}''(x_i) + \overbrace{\varphi'(x_i)}^{c_i}\,\sigma_{i+1}''(x_i) + \overbrace{\varphi'(x_{i+1})}^{c_{i+1}}\,\tau_{i+1}''(x_i)\,, \end{aligned} \tag{2}$$

$$\frac{6y_{i-1} - 6y_i + 2\overbrace{\varphi'(x_{i-1})}^{c_{i-1}} + 4\overbrace{\varphi'(x_i)}^{c_i}}{(x_i - x_{i-1})^2} = \frac{-6y_i + 6y_{i+1} - 4\overbrace{\varphi'(x_i)}^{c_i} - 2\overbrace{\varphi'(x_{i+1})}^{c_{i+1}}}{(x_{i+1} - x_i)^2}\,,$$

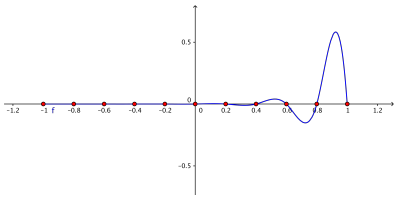
$$\begin{aligned} & \frac{2}{(x_i - x_{i-1})^2}\overbrace{\varphi'(x_{i-1})}^{c_{i-1}} + \frac{4}{(x_i - x_{i-1})^2}\overbrace{\varphi'(x_i)}^{c_i} + \\ & + \frac{4}{(x_{i+1} - x_i)^2}\overbrace{\varphi'(x_i)}^{c_i} + \frac{2}{(x_{i+1} - x_i)^2}\overbrace{\varphi'(x_{i+1})}^{c_{i+1}} = \frac{-6y_{i-1} + 6y_i}{(x_i - x_{i-1})^2} + \frac{-6y_i + 6y_{i+1}}{(x_{i+1} - x_i)^2}\,, \end{aligned}$$

to je $n-2$ lineárních rovnic o n neznámých; matice soustavy má největší koeficienty na diagonále.
Zbývají 2 volitelné parametry. Obvykle se volí $\varphi''(x_0) = \varphi''(x_{n-1}) = 0$, tzv. **přirozený spline** (angl. *natural spline*). To znamená, že v (2) pro $i = 0$, resp. $n-1$, nahradíme (nedefinovanou) levou, resp. pravou, stranu

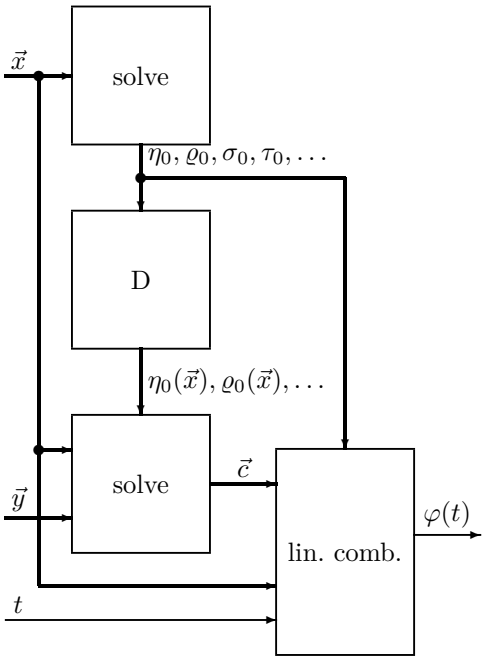
nulou.
Další možnosti, které v současnosti podporuje Maple:

$$\varphi'(x_0) = \varphi'(x_{n-1})\,, \qquad \varphi''(x_0) = \varphi''(x_{n-1})\,, \qquad (\texttt{endpoints='periodic'}),$$

spojitost 3. derivace v x_1, x_{n-2} (`endpoints='notaknot'`),
nebo explicitní zadání první derivace v krajních bodech, popř. jakýchkoli podobných podmínek v požadovaném počtu.
Tyto volby se projeví pouze na okrajích intervalu.



Blokové schema interpolace spiny



Výpočet má dvě části:
1. Výpočet koeficientů $c_i = \varphi'(x_i)$, $i = 0, \ldots, n-1$ (složitost nejvýše $\propto n^3$).
2. Výpočet funkční hodnoty (složitost $\propto \ln n$).

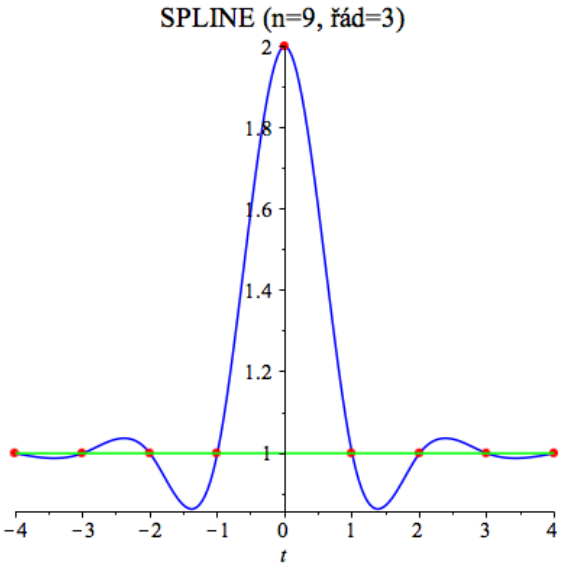
Matice soustavy je **třídiagonální** a koeficienty na diagonále jsou největší z řádku (využijeme pro efektivnější řešení).

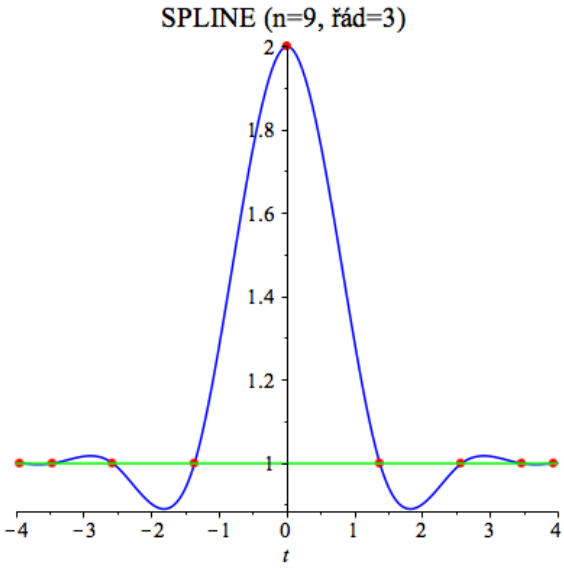
Spliny nelze extrapolovat!

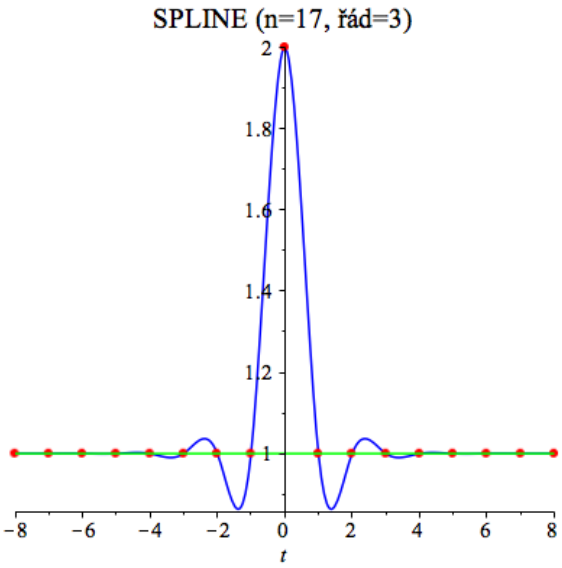
Přesto to většina implementací dovoluje.

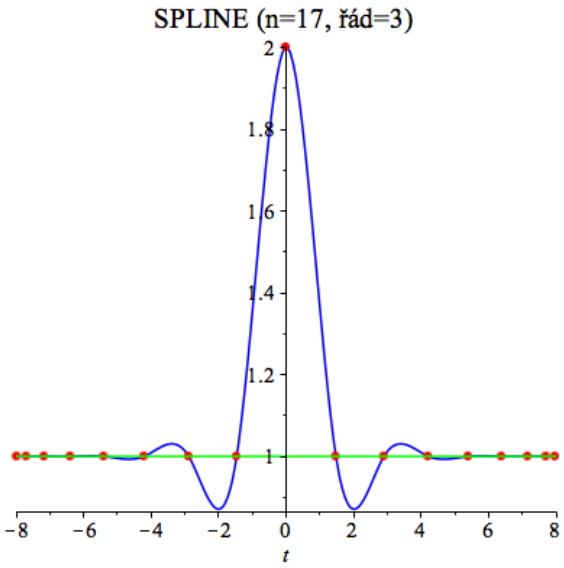
Poznámka: Volba rozdělení má v případě interpolace splinem výrazně menší vliv.

Vliv lokálních změn na spline

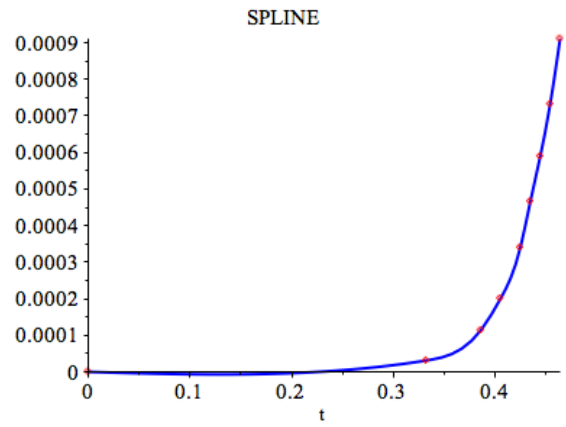


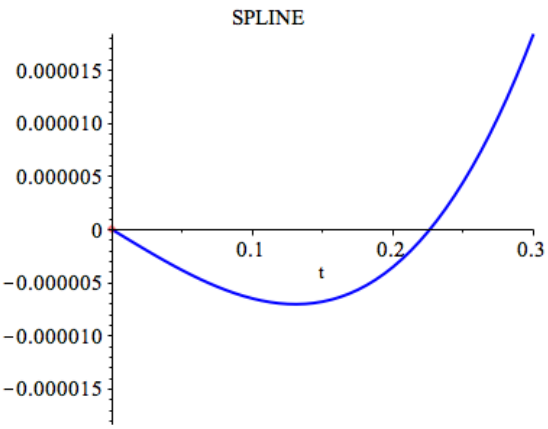






Motivační úloha - spline





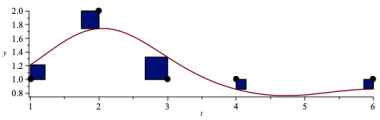
1.5 Metoda nejmenších čtverců

Úloha: Dáno:
vektor $\vec{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{R}^n$ určující n uzlových bodů (**ne nutně různých**),
vektor $\vec{y} = (y_0, \ldots, y_{n-1}) \in \mathbb{R}^n$ **žádoucích** hodnot v uzlových bodech,
k funkcí $\varphi_0, \ldots, \varphi_{k-1}$, $k \leq n$, definovaných alespoň ve všech uzlových bodech.
Hledáme: koeficienty $c_0, \ldots, c_{k-1} \in \mathbb{R}$ lineární kombinace funkcí φ_j

$$\varphi = \sum_{j<k} c_j \, \varphi_j$$

takové, abychom minimalizovali výraz

$$H_2 = \sum_{i<n} (\varphi(x_i) - y_i)^2 = \sum_{i<n} \left(\sum_{j<k} c_j \, \varphi_j(x_i) - y_i \right)^2$$



Modifikovaná kritéria

$$H_{2w} = \sum_{i<n} w_i \, (\varphi(x_i) - y_i)^2$$

řeší se analogicky (Maple podporuje),

$$H_1 = \sum_{i<n} |\varphi(x_i) - y_i|$$

nevede na jednoznačné řešení, neužívá se,

$$H_0 = \max_{i<n} |\varphi(x_i) - y_i|$$

řeší se (tzv. Čebyševova aproximace), ale je obtížnější.

1.5.1 Řešení aproximace podle kritéria nejmenších čtverců

V $\mathbb{R}^n$ zavedeme skalární součin vektorů
$\vec{u} = (u_0, \ldots, u_{n-1})$, $\vec{v} = (v_0, \ldots, v_{n-1})$:

$$\vec{u} \cdot \vec{v} = \sum_{i<n} u_i \cdot v_i \, .$$

Máme aproximovat vektor $\vec{y}$ lineární kombinací $\varphi(\vec{x}) = \sum_{j<k} c_j \, \varphi_j(\vec{x})$,
kritérium je $H_2 = (\varphi(\vec{x}) - \vec{y}) \cdot (\varphi(\vec{x}) - \vec{y}) = \|\varphi(\vec{x}) - \vec{y}\|^2$.
Řešení: Kolmý průmět splňuje soustavu podmínek (pro $m = 0, \ldots, k-1$)

$$\begin{aligned} (\varphi(\vec{x}) - \vec{y}) &\perp \varphi_m(\vec{x}) \, , \\ (\varphi(\vec{x}) - \vec{y}) \cdot \varphi_m(\vec{x}) &= 0 \, , \\ \varphi(\vec{x}) \cdot \varphi_m(\vec{x}) &= \vec{y} \cdot \varphi_m(\vec{x}) \, . \end{aligned}$$

Tedy $\varphi(\vec{x})$ se vzhledem ke skalárním součinům s vektory $\varphi_m(\vec{x})$ chová stejně jako $\vec{y}$.

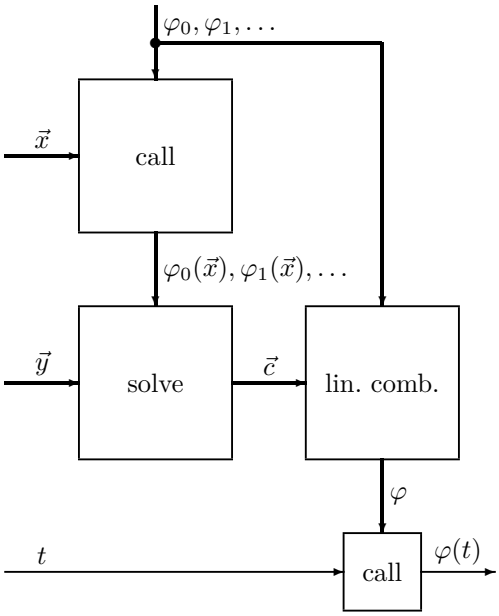
$$\begin{aligned}\left(\sum_{j<k} c_j\,\varphi_j(\vec{x})\right)\cdot\varphi_m(\vec{x}) &= \vec{y}\cdot\varphi_m(\vec{x})\,,\\ \sum_{j<k} c_j(\varphi_j(\vec{x})\cdot\varphi_m(\vec{x})) &= \vec{y}\cdot\varphi_m(\vec{x})\,, \qquad m=0,\ldots,k-1\,;\end{aligned}$$

soustava lineárních rovnic pro neznámé $c_0,\ldots,c_{k-1}$ (**soustava normálních rovnic**).

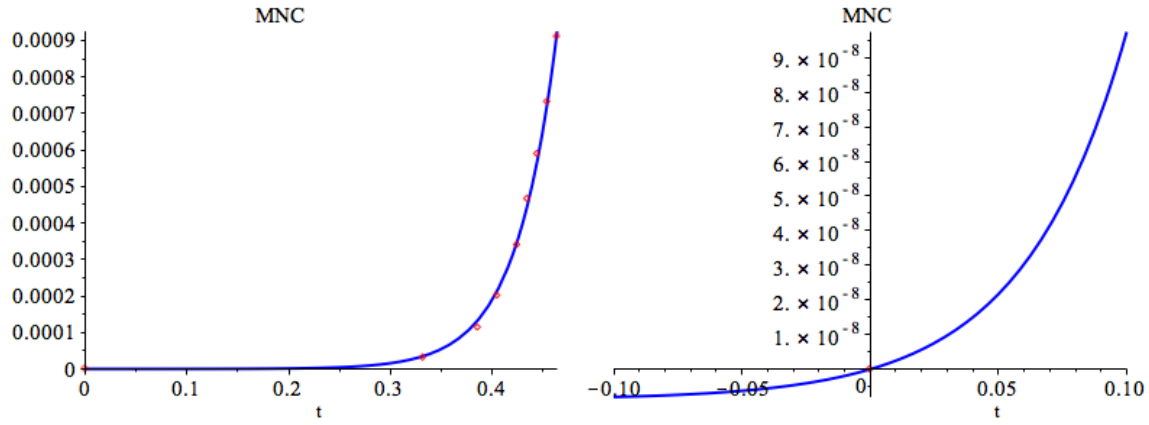
Speciální případ: aproximujeme polynomem stupně $< k$, můžeme volit $\varphi_j(t) = t^j$,

$$\varphi_j(\vec{x})\cdot\varphi_m(\vec{x}) = \sum_{i<n} x_i^j\cdot x_i^m = \sum_{i<n} x_i^{j+m}\,.$$

Blokové schema obecné aproximace metodou nejmenších čtverců



Motivační úloha - metoda nejmenších čtverců



Aproximace V-A charakteristiky diody lineární kombinací konstanty a dvou exponenciál s vhodnými základy.

1.5.2 Ortogonalizace

V podprostoru $P = \mathrm{Lin}\{\varphi_0(\vec{x}),\ldots,\varphi_{k-1}(\vec{x})\}$ najdeme ortogonální bázi $(\psi_0(\vec{x}),\ldots,\psi_{k-1}(\vec{x}))$,

$$\psi_j(\vec{x})\cdot\psi_m(\vec{x}) = 0 \text{ pro } j\neq m\,.$$

Ortogonalita závisí nejen na funkcích $\varphi_0, \ldots, \varphi_{k-1}$, ale i na volbě uzlových bodů!
Hledáme řešení ve tvaru $\varphi = \sum_{j<k} d_j \psi_j$, kde d_j, $j = 0, \ldots, k-1$ jsou souřadnice vzhledem k nové bázi. Matice soustavy normálních rovnic je diagonální:

$$\begin{aligned} d_j \left(\psi_j(\vec{x}) \cdot \psi_j(\vec{x})\right) &= \vec{y} \cdot \psi_j(\vec{x}), \qquad j = 0, \ldots, k-1, \\ d_j = \frac{\vec{y} \cdot \psi_j(\vec{x})}{\psi_j(\vec{x}) \cdot \psi_j(\vec{x})} &= \frac{\vec{y} \cdot \psi_j(\vec{x})}{\|\psi_j(\vec{x})\|^2}, \qquad j = 0, \ldots, k-1. \end{aligned}$$

Navíc lze volit vektory $\psi_j(\vec{x})$ jednotkové, pak vyjde jednotkový i jmenovatel.

Gramova-Schmidtova ortogonalizace

Z báze $(\vec{\varphi}_0, \ldots, \vec{\varphi}_{k-1})$ vytvoříme ortogonální bázi $(\psi_0(\vec{x}), \ldots, \psi_{k-1}(\vec{x}))$:

$$\begin{aligned} \psi_0(\vec{x}) &= \varphi_0(\vec{x}), \\ \psi_1(\vec{x}) &= \varphi_1(\vec{x}) + \alpha_{1,0}\psi_0(\vec{x}), \end{aligned}$$

$$\begin{aligned} \psi_1(\vec{x}) \cdot \psi_0(\vec{x}) = 0 \quad &\Rightarrow \quad \varphi_1(\vec{x}) \cdot \psi_0(\vec{x}) + \alpha_{1,0}\psi_0(\vec{x}) \cdot \psi_0(\vec{x}) = 0 \\ &\Rightarrow \quad \alpha_{1,0} = \frac{-\varphi_1(\vec{x}) \cdot \psi_0(\vec{x})}{\psi_0(\vec{x}) \cdot \psi_0(\vec{x})} \end{aligned}$$

$$\psi_2(\vec{x}) = \varphi_2(\vec{x}) + \alpha_{2,0}\psi_0(\vec{x}) + \alpha_{2,1}\psi_1(\vec{x})$$

$$\begin{aligned} \psi_2(\vec{x}) \cdot \psi_0(\vec{x}) = 0 \quad &\Rightarrow \quad \varphi_2(\vec{x}) \cdot \psi_0(\vec{x}) + \alpha_{2,0}\psi_0(\vec{x}) \cdot \psi_0(\vec{x}) + \alpha_{2,1} \underbrace{\psi_1(\vec{x}) \cdot \psi_0(\vec{x})}_{0} = 0 \\ &\Rightarrow \quad \alpha_{2,0} = \frac{-\varphi_2(\vec{x}) \cdot \psi_0(\vec{x})}{\psi_0(\vec{x}) \cdot \psi_0(\vec{x})} \end{aligned}$$

$$\begin{aligned} \psi_2(\vec{x}) \cdot \psi_1(\vec{x}) = 0 \quad &\Rightarrow \quad \varphi_2(\vec{x}) \cdot \psi_1(\vec{x}) + \alpha_{2,0} \underbrace{\psi_0(\vec{x}) \cdot \psi_1(\vec{x})}_{0} + \alpha_{2,1}\psi_1(\vec{x}) \cdot \psi_1(\vec{x}) = 0 \\ &\Rightarrow \quad \alpha_{2,1} = \frac{-\varphi_2(\vec{x}) \cdot \psi_1(\vec{x})}{\psi_1(\vec{x}) \cdot \psi_1(\vec{x})} \end{aligned}$$

$$\cdots$$

$$\psi_j(\vec{x}) = \varphi_j(\vec{x}) + \sum_{m<j} \alpha_{j,m}\psi_m(\vec{x})$$

$$\begin{aligned} \forall p, p < j: \quad \psi_j(\vec{x}) \cdot \psi_p(\vec{x}) &= 0 = \varphi_j(\vec{x}) \cdot \psi_p(\vec{x}) + \sum_{m<j} \alpha_{j,m}\psi_m(\vec{x}) \cdot \psi_p(\vec{x}) \\ &= \varphi_j(\vec{x}) \cdot \psi_p(\vec{x}) + \alpha_{j,p}\psi_p(\vec{x}) \cdot \psi_p(\vec{x}) \\ \Rightarrow \alpha_{j,p} &= \frac{-\varphi_j(\vec{x}) \cdot \psi_p(\vec{x})}{\psi_p(\vec{x}) \cdot \psi_p(\vec{x})} \end{aligned}$$

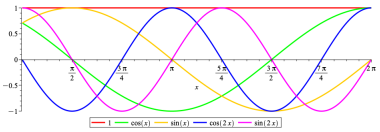
$$\begin{pmatrix} \psi_0(\vec{x}) \\ \psi_1(\vec{x}) \\ \psi_2(\vec{x}) \\ \psi_3(\vec{x}) \\ \vdots \\ \psi_{k-1}(\vec{x}) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ \alpha_{1,0} & 1 & 0 & 0 & \cdots & 0 \\ \alpha_{2,0} & \alpha_{2,1} & 1 & 0 & \cdots & 0 \\ \alpha_{3,0} & \alpha_{3,1} & \alpha_{3,2} & 1 & & 0 \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ \alpha_{k-1,0} & \alpha_{k-1,1} & \alpha_{k-1,2} & \cdots & \cdots & 1 \end{pmatrix} \begin{pmatrix} \vec{\varphi}_0 \\ \vec{\varphi}_1 \\ \vec{\varphi}_2 \\ \vec{\varphi}_3 \\ \vdots \\ \vec{\varphi}_{k-1} \end{pmatrix}$$

(Komu vadí maticový zápis pro vektory, může je nahradit řádky matice.)

1.5.3 Aproximace goniometrickým polynomem

(konečnou Fourierovou řadou)
Aproximace metodou nejmenších čtverců, přičemž aproximační funkce jsou

$$1\,, \quad \cos 2\pi\frac{t}{T}\,, \quad \sin 2\pi\frac{t}{T}\,, \quad \cos 4\pi\frac{t}{T}\,, \quad \sin 4\pi\frac{t}{T}\,, \quad \dots$$



Pro ekvidistantní uzlové body na intervalu délky T,

$$x_i = a + i\,\frac{T}{n}, \qquad i = 0, \dots, n-1$$

jsou vektory $\varphi_j(\vec{x})$ (kterých smí být nejvýše n) ortogonální.
Pro ortogonální funkce je složitost $\propto k\,n$.
Pro $k = n$ dostáváme $\propto n^2$.

Rychlá Fourierova transformace (Fast Fourier Transform, FFT) dále snižuje složitost na $\propto n \ln n$.

K tomu vyžaduje navíc $n = 2^m$, jinak efektivita klesá.

Použití: Aproximace periodických a „téměř periodických“ průběhů, zejména akustických, ale např. i obrazových; komprese mp3 a jpeg.
Rozklad na frekvence dovoluje další zpracování, digitální filtraci, rozpoznávání atd.

1.5.4 Čebyševova aproximace polynomem

Úloha: Dáno:
omezený interval I,
spojitá funkce f na I,
$k \in \mathbb{N}$.
Hledáme: polynom φ stupně menšího než k takový, abychom minimalizovali výraz

$$H_0 = \max_{t \in I} |\varphi(t) - f(t)|\,.$$

To se také dělá, ale je to mnohem pracnější. Častěji se používá modifikovaná aproximace metodou nejmenších čtverců:

Čebyševova aproximace polynomem

Pro jednoduchost na intervalu $I = \langle -1, 1\rangle$; zobecnění na interval $\langle a, b\rangle$ dostaneme lineární transformací

$$x = \frac{b+a}{2} + \frac{b-a}{2}z\,,$$

inverzní transformace je

$$z = \frac{x - \frac{b+a}{2}}{\frac{b-a}{2}}\,.$$

Za bázi prostoru všech polynomů stupně $< k$ volíme **Čebyševovy polynomy**.
Pokud můžeme, volíme $n \geq k$ uzlových bodů $x_0, \dots, x_{n-1} \in \langle -1, 1\rangle$ jako kořeny Čebyševova polynomu stupně n, tj. s kosinovým rozdělením:

$$x_i = \cos\Big(\frac{\pi}{n}\Big(i + \frac{1}{2}\Big)\Big)\,, \qquad i = 0, \dots, n-1.$$

Pro $k = n$ dostaneme interpolační polynom (s doporučeným kosinovým rozdělením uzlových bodů). Řešení pro $k < n$ se od interpolačního polynomu liší zanedbáním členů vyššího řádu. Koeficienty $c_k, \ldots, c_{n-1}$ bývají malé. (Závisí ovšem na vyšších derivacích aproximované funkce!) Chyba v uzlových bodech je proto omezena výrazem

$$|\varphi(x_i) - f(x_i)| \leq \sum_{j=k}^{n-1} |c_j|.$$

Poznámky o Čebyševově aproximaci

- Neoptimalizujeme přesně kritérium H_0, ale výsledek se od optimálního řešení příliš neliší.
- O chybě mimo uzlové body nelze říci mnoho, přesto lze postup doporučit.
- Rekurentní vzorec lze použít nejen ke stanovení Čebyševových polynomů, ale i přímo k výpočtu jejich hodnot v daném bodě.
- Nedoporučuje se výsledek roznásobovat do standardního tvaru $\varphi(t) = \sum_{j<k} b_j\, t^j$.
- Metodu lze zobecnit i na případ, kdy hledáme aproximaci ve tvaru součinu známé funkce a neznámého polynomu.

1.6 Dodatek: Přehled značení

Popis je zjednodušený a nemusí být přesný, podrobnosti jsou v textu. Značení použité jen lokálně zde není uvedeno.

Značení specifické pro tuto kapitolu

n ... počet uzlových bodů
$\vec{x} = (x_0, \ldots, x_{n-1})$ ... vektor uzlových bodů
$\vec{y} = (y_0, \ldots, y_{n-1})$ ... vektor požadovaných hodnot v uzlových bodech
f ... aproximovaná funkce, $f(x_i) = y_i$
φ ... aproximující funkce, $\varphi \approx f$, $\varphi(x_i) \doteq y_i$
$\mathcal{F}$ ... množina povolených aproximujících funkcí, $\varphi \in \mathcal{F}$
$(\varphi_0, \ldots, \varphi_{k-1})$ ... báze prostoru povolených aproximujících funkcí, $\mathcal{F} \in \mathrm{Lin}\{\varphi_0, \ldots, \varphi_{k-1}\}$
k ... dimenze prostoru povolených aproximujících funkcí, $k = \dim \mathcal{F}$
$\vec{c} = (c_0, \ldots, c_{k-1})$ ... vektor koeficientů lineární kombinace, souřadnice aproximace φ vzhledem k bázi $(\varphi_0, \ldots, \varphi_{k-1})$, $\varphi = \sum_{j<k} c_j \, \varphi_j$
$\varphi(\vec{x}), \varphi_0(\vec{x}), \ldots, \varphi_{k-1}(\vec{x}), \psi_0(\vec{x})$ apod. ... vektory hodnot funkcí v uzlových bodech $x_0, \ldots, x_{n-1}$, $\varphi(\vec{x}) = (\varphi(x_0), \ldots, \varphi(x_{n-1}))$ apod.
W ... polynom použitý v odhadech chyby, $W(t) = \prod_{i<n}(t - x_i)$
$\overline{w}$ ... horní odhad absolutní hodnoty polynomu W na použitém intervalu
γ_j ... Čebyševův polynom řádu j, $\gamma_j(t) = \cos(j \arccos t)$
$y_{i,j}$ ... požadovaná hodnota j-té derivace v i-tém uzlovém bodě, $y_{i,j} = f^{(j)}(x_i) \doteq \varphi^{(j)}(x_i)$
T ... perioda aproximované periodické funkce

Značení používané podobně v celém předmětu

δ_{ij} ... Kroneckerovo delta, $\delta_{ii} = 1$, $\delta_{ij} = 0$ pro $i \neq j$
$\cdot$ ... skalární součin, $\vec{u} \cdot \vec{v} = \sum_{i<n} u_i \cdot v_i$
$\|.\|$ ... norma, obvykle euklidovská, $\|\vec{u}\| = \sqrt{\vec{u} \cdot \vec{u}}$
$I(\ldots)$ ... nejmenší interval obsahující čísla (body) v závorce, např. $I(x_0, \ldots, x_{n-1}) = \langle \min_i x_i, \max_i x_i \rangle$
M_j ... horní odhad absolutní hodnoty j-té derivace aproximované funkce, $|f^{(j)}| \leq M_j$ na použitém intervalu

Literatura

[Navara, Němeček] Navara, M., Němeček, A.: *Numerické metody.* ČVUT, Praha, dotisk 2005.

[Knuth] Knuth, D.E.: *Fundamental Algorithms.* Vol. 1 of *The Art of Computer Programming,* 3rd ed., Addison-Wesley, Reading, MA, 1997.

[Num. Recipes] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical Recipes (The Art of Scientific Computing).* 3rd edition, Cambridge University Press, Cambridge, 2007. `http://www.nrbook.com/a/bookcpdf.php`

[Handbook Lin. Alg.] Hogben, L. (ed.): *Handbook of Linear Algebra.* Chapman & Hall/CRC, Boca Raton/London/New York, 2007.