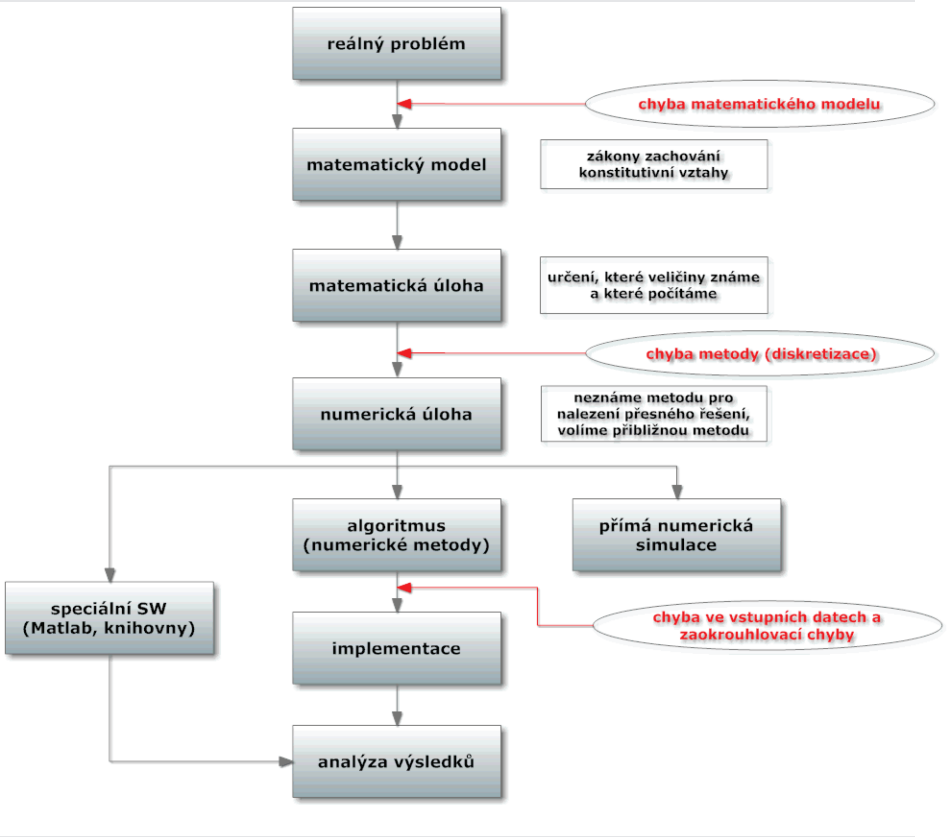
Kapitola 1. Úvod do numerické matematiky

Numerická matematika = věda, která se zabývá řešením matematicky formulovaných úloh pomocí logických operací a aritmetických operací s čísly o konečné délce.



Příklad

Reálný problém ... intravenózní dávkování léku

Matematický model

- nezávisle proměnná je pouze čas t
- šíření látky není závislé na prostorových proměnných

Numerické metody
Josef Daněk
KMA/NM
13.2.2013

- popis pomocí diferenciální rovnice

$$\boxed{\frac{dC}{dt} = -k \cdot C}$$

kde C je koncentrace látky v krvi a $k > 0$ je absorpční koeficient

- počáteční podmínka

$$\boxed{C(0) = C_0}$$

chyba matematického modelu odpovídá zjednodušujícím předpokladům

Matematická úloha

- chceme vypočítat hodnotu koncentrace látky v čase $t \in < 0, T >$

Numerická úloha

- řešení hledáme pouze v konečně mnoha bodech
  (diskretizujeme čas, $t_0 = 0$, $t_n = n \cdot \frac{T}{N}$, $t_N = T$)
  N je počet dělení intervalu $< 0, T >$

chyba diskretizace (metody)

Numerická metoda

- derivaci $\frac{dC}{dt}$ aproximujeme poměrnou diferencí

$$\boxed{\frac{C_{n+1} - C_n}{\frac{T}{N}} = -k \cdot C_n}$$

chyba diskretizace (metody)

Výpočet

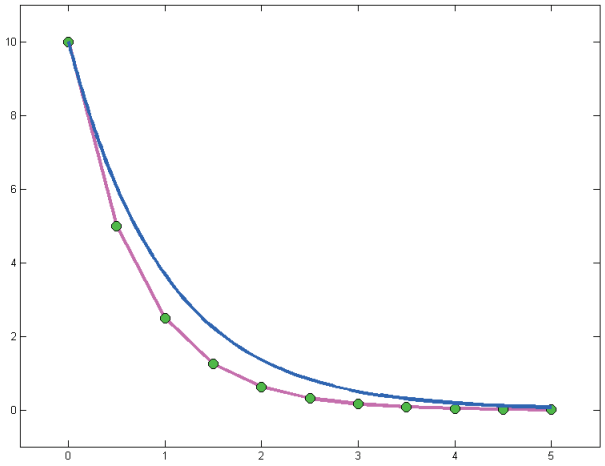
$$C_{n+1} = (1 - \frac{T}{N} \cdot k) \cdot C_n, \quad C_0 \text{ dáno}$$

zaokrouhlovací chyby

Analytické řešení

$$C(t) = C_0 \cdot e^{-kT}$$

např: $C(0) = 10$, $k = 1$, $T = 5$, $N = 10$



Reálné hodnoty ???

CHYBY

x ... přesná hodnota
$\tilde{x}$ ... přibližná hodnota

absolutní chyba ... $A(x) = |x - \tilde{x}| \le \underbrace{a(x)}_{\text{odhad}}$

relativní chyba ... $R(x) = \dfrac{A(x)}{|x|} \le \underbrace{r(x)}_{\text{odhad}}$

Pozn.: Při odečítání „blízkých" čísel roste relativní chyba (ztráta platných číslic)

$$\boxed{a(x \pm y) = a(x) + a(y)}$$

$$|(x \pm y) - (\tilde{x} \pm \tilde{y})| \le |x - \tilde{x}| + |\tilde{y} - y|$$

$$\boxed{r(x \pm y) = \frac{a(x) + a(y)}{|x \pm y|}} \qquad |x \pm y| \to 0_+ \quad \text{!!!}$$

Pozn.: Násobení a dělení nemohou podstatně zvětšit relativní chybu

$$\boxed{a(x \cdot y) = |x| \cdot a(y) + |y| \cdot a(x)}$$

$$|xy - \tilde{x}\tilde{y}| = |xy - \tilde{x}y + \tilde{x}y - \tilde{x}\tilde{y}| = |y(x - \tilde{x}) + \underbrace{\tilde{x}}_{\approx x}(y - \tilde{y})| \le |y| \cdot |x - \tilde{x}| + |x| \cdot |y - \tilde{y}|$$

$$\boxed{r(x \cdot y) = r(x) + r(y)}$$

Numerické metody KMA/NM
Josef Daněk 13.2.2o13

$$\frac{|x|a(y) + |y|a(x)}{|xy|} = \frac{a(y)}{|y|} + \frac{a(x)}{|x|}$$

$$\boxed{a\left(\frac{x}{y}\right) = \frac{|x| \cdot a(y) + |y| \cdot a(x)}{y^2}}$$

$$\left|\frac{x}{y} - \frac{\tilde{x}}{\tilde{y}}\right| = \left|\frac{1}{y\tilde{y}}(x\tilde{y} - \tilde{x}y)\right| = \left|\frac{1}{y\underbrace{\tilde{y}}_{\approx y}}(x\tilde{y} - xy + xy - \tilde{x}y)\right| =$$

$$= \left|\frac{1}{y\tilde{y}}(x(\tilde{y} - y) + y(x - \tilde{x}))\right| \le \frac{1}{y^2}(|x| \cdot |y - \tilde{y}| + |y| \cdot |x - \tilde{x}|)$$

$$\boxed{r\left(\frac{x}{y}\right) = r(x) + r(y)}$$

$$\frac{\dfrac{|x|a(y) + |y|a(x)}{y^2}}{\left|\dfrac{x}{y}\right|} = \frac{a(y)}{|y|} + \frac{a(x)}{|x|}$$

Definice: Mějme dány dvě množiny X(vstupní data) a Y(výstupní data). Předpokládejme, že X, Y jsou Banachovy prostory. **Úlohou** rozumíme relaci

$$y = U(x),\ x \in X,\ y \in Y.$$

Definice: Řekneme, že úloha je **korektní** na dvojici prostorů (X, Y), když

- $\forall x \in X\ \exists! y \in Y:\ y = U(x)$ (zobrazení),
- řešení y spojitě závisí na vstupních datech

$$\forall \{x_n\}:\ x_n \to x,\ U(x_n) = y_n:\ y_n \to y = U(x).$$

Poznámka: Banachův prostor = úplný + normovaný

úplný prostor: metrický prostor, kde $\forall$ Cauchyovská posl. $u_n \subset X$ má limitu $u \in X$

normovaný prostor = množina X:

a) X je lineární;

b) $\forall u \in X \to ||u||$:

$$||u|| \ge 0,\ \ ||u|| = 0 \Leftrightarrow u = 0;$$
$$||au|| = |a| \cdot ||u|| \quad \forall a \in \mathbb{R};$$
$$||u + v|| \le ||u|| + ||v||;$$

c) $d(u, v) = ||u - v||$

Poznámka: Protože X, Y jsou Banachovy prostory, lze spojitost zaručit podmínkou

$$||y_n - y||_Y \le L||x_n - x||_X.$$

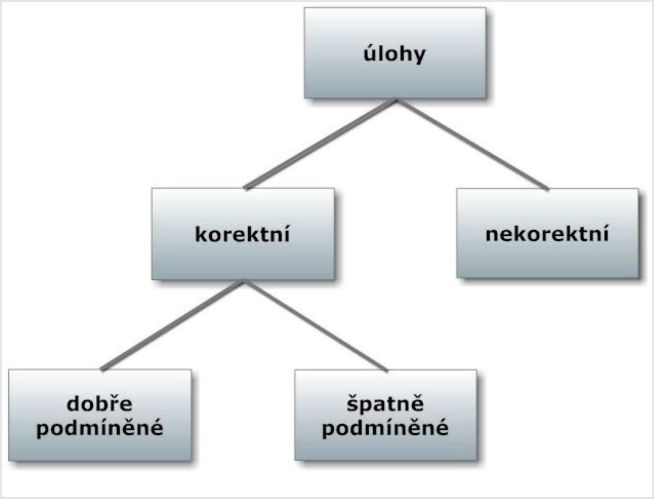
Poznámka: **Nekorektní** úlohy jsou úlohy, které nejsou korektní.
Někdy je nekorektnost způsobena pouze nevhodnou formulací.

Definice: Úloha je **dobře podmíněná**, jestliže malá relativní změna ve vstupních datech vyvolá malou relativní změnu řešení.

Číslo podmíněnosti úlohy $y = U(x)$

$$C_p = \frac{\frac{||\Delta y||}{||y||}}{\frac{||\Delta x||}{||x||}}$$

Poznámka: Je-li $C_p \approx 1$ je úloha velmi dobře podmíněná.
V praxi hovoříme o špatně podmíněné úloze pro $C_p \gtrapprox 100$.



Příklad 1

Posuďte podmíněnost úlohy určit hodnotu funkce $y = \sin(x)$

a) v bodě 3,14;
b) v bodě -0,01.

---

a) Volíme $x = 3,14,\ \Delta x = 0,01 \quad \leftarrow \quad$ *malá změna na vstupu*

$(y =)\ \sin x = \sin 3,14 = 0,0015926$

$(y + \Delta y =)\ \sin(x + \Delta x) = \sin 3,15 = -0,0084072$

$\Delta y = \sin(x + \Delta x) - \sin x = -0,0099998 \quad \leftarrow \quad$ *změna na výstupu*

Relativní chyba na vstupu: $\dfrac{|\Delta x|}{|x|} \doteq 0,0031847$

Numerické metody KMA/NM
Josef Daněk 13.2.2o13

Relativní chyba na výstupu: $\dfrac{|\Delta y|}{|y|} \doteq 6,2789149$

$C_p \doteq 1971,6 \quad \rightarrow \quad$ špatně podmíněná úloha

---

b) Volíme $x = -0,01,\ \Delta x = 0,01$

$\sin x = -0,0099998$

$\sin(x + \Delta x) = \sin 0 = 0$

$\Delta y = 0,099998$

Relativní chyba na vstupu: $\dfrac{|\Delta x|}{|x|} \doteq 1$

Relativní chyba na výstupu: $\dfrac{|\Delta y|}{|y|} \doteq 1$

$C_p \doteq 1 \quad \rightarrow \quad$ velmi dobře podmíněná úloha

Poznámka: Podívejme se na předchozí příklad obecněji. Úloha má tvar $y = f(x)$.
Podle věty o střední hodnotě platí:

$$|\Delta y| \approx |f'(x)| \cdot |\Delta x|$$

odtud:

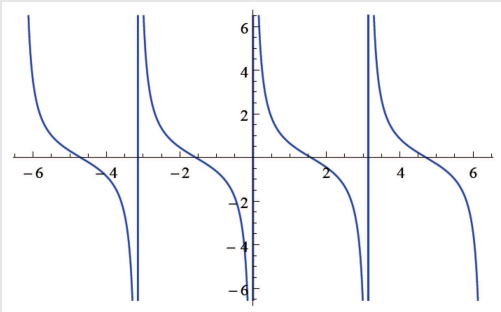
$$\left|\frac{\Delta y}{y}\right| \approx \frac{|f'(x)| \cdot |\Delta x|}{|f(x)|} = \boxed{\left|\frac{x \cdot f'(x)}{f(x)}\right|} \cdot \left|\frac{\Delta x}{x}\right|$$

Tedy

$$C_p \approx \left|\frac{x \cdot f'(x)}{f(x)}\right|$$

v našem případě: $y = \sin x \quad \Rightarrow \quad y' = \cos x$

$$C_p \approx \left|\frac{x \cos x}{\sin x}\right| = |x \operatorname{cotg} x|$$



$\lim\limits_{x \to \pi\pm} x \operatorname{cotg} x = \pm\infty$

$\lim\limits_{x \to 0\pm} x \dfrac{\cos x}{\sin x} = \cos 0 \cdot \lim\limits_{x \to 0} \dfrac{x}{\sin x} = 1 \cdot 1 = 1$

Poznámka: Podobné příklady (posuďte podmíněnost úlohy určit hodnotu):

a) $\boxed{f(x)=x^\alpha,\ x\to 0,\ x>0}$

$$C_p=\left|\frac{xf'(x)}{f(x)}\right|=\left|\frac{x\alpha x^{\alpha-1}}{x^\alpha}\right|=\alpha$$

b) $\boxed{f(x)=\arcsin x,\ x\to 1,\ x<1}$

$$C_p=\left|\frac{xf'(x)}{f(x)}\right|=\left|\frac{x\frac{1}{\sqrt{1-x^2}}}{\arcsin x}\right|=\left|\underbrace{\frac{1}{\sqrt{1-x^2}}}_{\to 0+}\overbrace{\left(\frac{x}{\arcsin x}\right)}^{\to 1}\right|\xrightarrow{x\to 1}\infty$$

c) $\boxed{f(x)=x-1,\ x\to 1}$

$$C_p=\left|\frac{x\cdot 1}{x-1}\right|\longrightarrow\infty$$

Příklad 2

Posuďte podmíněnost úlohy řešit soustavu lineárních algebraických rovnic (pro $\alpha\neq\pm1$)

$$\begin{aligned}x+\alpha y&=1\\ \alpha x+y&=0\\ x(1-\alpha^2)&=1\end{aligned}$$

$$\boxed{x=\frac{1}{1-\alpha^2}}$$

$$y=-\frac{\alpha}{1-\alpha^2}$$

Nechť vstup je hodnota $\underline{\alpha}$ a výstup hodnota $\underline{x}$.
Pak

$$C_p=\frac{\frac{|\Delta x|}{|x|}}{\frac{|\Delta\alpha|}{|\alpha|}}\overset{*}{\approx}\boxed{\left|\alpha\frac{dx}{d\alpha}\frac{1}{x}\right|}\underbrace{=}_{**}\left|\alpha\frac{2\alpha}{(1-\alpha^2)^2}\frac{1}{\frac{1}{1-\alpha^2}}\right|=\frac{2\alpha^2}{1-\alpha^2}$$

$\Rightarrow$ pro $\alpha^2\to 1$ je tato úloha špatně podmíněná!

---

* viz předchozí poznámka

** $\frac{dx}{d\alpha}=\frac{d}{d\alpha}\left(\frac{1}{1-\alpha^2}\right)=-\left(\frac{1}{(1-\alpha^2)^2}(-2\alpha)\right)$

Pozn.: Matice výše uvedené soustavy je pro hodnoty α blízké ±1 skoro singulární.

STABILITA (PODMÍNĚNOST) ALGORITMU

Numerické metody KMA/NM
Josef Daněk 13.2.2o13

„U nestabilní metody (algoritmu) se relativně malé chyby v jednotlivých krocích výpočtu postupně akumulují tak, že dojde ke katastrofální ztrátě přesnosti numerického řešení úlohy."

- Při výpočtu dochází k zaokrouhlovacím chybám. Je proto vhodné vybírat algoritmy málo citlivé na zaokrouhlovací chyby.

Stabilní algoritmus

- dobře podmíněný - málo citlivý na poruchy ve vstupních datech
- numericky stabilní - málo citlivý na vliv zaokrouhlovacích chyb

Poznámka:
U stabilních metod roste chyba výsledku s počtem kroků N nejvýše lineárně
(v ideálním případě, kdy je znaménko chyby náhodné, zaokrouhlovací chyba roste $\sim\sqrt{N}$).

U nestabilních metod roste zaokrouhlovací chyba rychleji, např. geometrickou řadou $\sim q^N$, kde $|q|>1$.

Příklad 3

Řešte diferenční rovnici (rekurentní formule, nestabilní rekurze)

$$\boxed{x_{n+1}=\frac{13}{3}x_n-\frac{4}{3}x_{n-1},\quad x_0=1,\ x_1=\frac{1}{3}}$$

Snadno se ukáže, že řešení je $x_n=\frac{1}{3^n}$ (dosazením).

Při numerickém výpočtu dojdeme k problémům (viz obr). Hodnoty x_n začnou velmi rychle klesat.
Pro vysvětlení ukážeme obecné řešení zadané diferenční rovnice.

- charakteristický polynom

$$\lambda^2=\frac{13}{3}\lambda-\frac{4}{3}$$

(předpokládáme řešení λ^n: $\lambda^{n+1}=\frac{13}{3}\lambda^n-\frac{4}{3}\lambda^{n-1}$)

- kořeny

$$\lambda_{1,2}=\frac{\frac{13}{3}\pm\sqrt{(\frac{13}{3})^2-4\cdot\frac{4}{3}}}{2}=\frac{\frac{13}{3}\pm\sqrt{\frac{121}{9}}}{2},\quad\text{tj. }\lambda_1=\frac{1}{3},\lambda_2=4$$

- obecné řešení

$$x_n=A\cdot\left(\frac{1}{3}\right)^n+B\cdot 4^n$$

$$x_0=1=A\cdot\left(\frac{1}{3}\right)^0+B\cdot 4^0=A+B=1$$

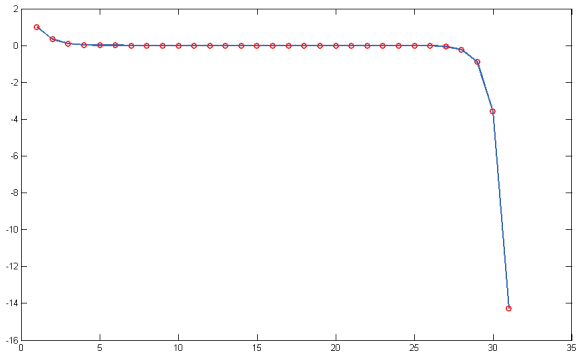
$$x_1=\frac{1}{3}=A\cdot\left(\frac{1}{3}\right)^1+B\cdot 4^1=\frac{1}{3}\cdot A+4\cdot B=\frac{1}{3}$$

$$\Rightarrow\quad A=1,\ B=0$$

Přes počáteční podmínku $B=0$ vzniknou vlivem zaokrouhlovacích chyb malé druhé komponenty řešení

Výsledky z MATLABu, FORMAT SHORT, pevná čárka na 5 číslic

```
clc;
clear;
format short;

n=30;

x(1)=1;
x(2)=1/3;

for i=2:n
 x(i+1)=13/3*x(i)-4/3*x(i-1);
end

plot(1:n+1,x,'b-',1:n+1,x,'ro');
```



Příklad 4

Vypočtěte přibližně hodnotu

$$\boxed{J_n = \int_0^1 \frac{x^n}{x+5}\,dx}$$

Platí:

$$\underbrace{\int_0^1 x^{n-1}\,dx}_{\left[\frac{1}{n}x^n\right]_0^1 = \frac{1}{n}} = \int_0^1 \frac{x^{n-1}(x+5)}{x+5}\,dx = \underbrace{\int_0^1 \frac{x^n}{x+5}\,dx}_{J_n} + 5\underbrace{\int_0^1 \frac{x^{n-1}}{x+5}\,dx}_{J_{n-1}}$$

Dále:

$$J_0 = \int_0^1 \frac{1}{x+5}\,dx = [\ln|x+5|]_0^1 = \ln\frac{6}{5}$$

Rekurentní formule:

$$\boxed{\begin{aligned} J_0 &= \ln\frac{6}{5} \\ J_n &= -5\cdot J_{n-1} + \frac{1}{n} \end{aligned}}$$

Nestabilní algoritmus ! ... vždy $\exists n_0 :\ J_{n_0} < 0$!

Proto je lépe postupovat odzadu:

- dokážeme, že $\boxed{\lim_{n\to\infty} J_n = 0}$

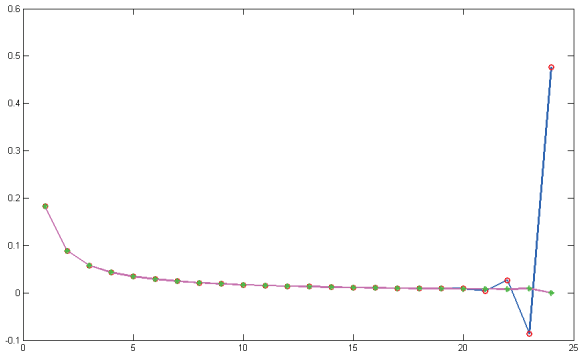
$$|J_n| = \left|\int_0^1 \frac{x^n}{x+5}\,dx\right| \le \int_0^1 \left|\frac{x^n}{x+5}\right| dx \le \frac{1}{5}\int_0^1 x^n dx = \frac{1}{5(n+1)} \to_{n\to\infty} 0$$

- např. zvolíme $J_{100} = 0$ a počítáme $J_{n-1} = -\frac{1}{5}(J_n - \frac{1}{n})$

$$\boxed{\begin{aligned} J_{100} &= 0 \\ J_{n-1} &= -\frac{1}{5}\cdot J_n + \frac{1}{5n} \end{aligned}}$$

Výsledky z MATLABu, FORMAT SHORT E

```
clc;
clear;
format short e;

n=24;

J(1)=log(6/5);
JJ(n)=0;

for i=1:n-1
 J(i+1) = -5*J(i) + 1/i;
end
for i=n-1:-1:1
 JJ(i) = (1/i - JJ(i+1)) / 5;
end

[J' JJ']
plot(1:n,J,'b-',1:n,J,'ro');
hold on
plot(1:n,JJ,'m-',1:n,JJ,'g*');
```



J	JJ
1.8232e-001	1.8232e-001
8.8392e-002	8.8392e-002
5.8039e-002	5.8039e-002
4.3139e-002	4.3139e-002
3.4306e-002	3.4306e-002
2.8468e-002	2.8468e-002
2.4325e-002	2.4325e-002
2.1233e-002	2.1233e-002
1.8837e-002	1.8837e-002
1.6926e-002	1.6926e-002
1.5368e-002	1.5368e-002
1.4071e-002	1.4071e-002
1.2977e-002	1.2977e-002
1.2040e-002	1.2040e-002
1.1229e-002	1.1229e-002
1.0522e-002	1.0521e-002
9.8903e-003	9.8964e-003
9.3719e-003	9.3414e-003
8.6960e-003	8.8485e-003
9.1515e-003	8.3893e-003
4.2426e-003	8.0535e-003
2.6406e-002	7.3518e-003
-8.6575e-002	8.6957e-003
4.7635e-001	0

Zobrazení čísel

Motivace: $\boxed{\sum_{k=1}^{100000} \frac{1}{10} = 9998{,}55664}$

- Lidé používají desítkovou soustavu.
- Počítače dvojkovou.

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Komunikace s počítačem

- Zadání v 10-soustavě.
- Převod do 2-soustavy (počítač).
- Výpočet (počítač).
- Zpětný převod do 10-soustavy (počítač).
- Výsledek v 10-soustavě.

Soustavy

- desítková

  $1563 = (1 \cdot 10^3) + (5 \cdot 10^2) + (6 \cdot 10^1) + (3 \cdot 10^0)$

  obecně

  $N = (a_k \cdot 10^k) + (a_{k-1} \cdot 10^{k-1}) + \cdots + (a_1 \cdot 10^1) + (a_0 \cdot 10^0)$

  $(N \in \mathbb{N}), \quad a_k \in \{0, 1, 2, \ldots, 9\}$

  značení

  $N = a_k a_{k-1} a_{k-2} \ldots a_1 a_0$

- dvojková

  $1563 = (1{\cdot}2^{10}){+}(1{\cdot}2^9){+}(0{\cdot}2^8){+}(0{\cdot}2^7){+}(0{\cdot}2^6){+}(0{\cdot}2^5){+}(1{\cdot}2^4){+}(1{\cdot}2^3){+}(0{\cdot}2^2){+}(1{\cdot}2^1){+}(1{\cdot}2^0)$

  $\boxed{(1563)_{10} \;=\; (11000011011)_2}$

Binární zlomky

lze vyjádřit jako sumu se zápornými mocninami dvou

$R \in \mathbb{R} \qquad 0 < R < 1 \qquad d_j \in \{0, 1\}$

$R = (d_1 \cdot 2^{-1}) + (d_2 \cdot 2^{-2}) + \ldots + (d_n \cdot 2^{-n}) + \ldots$

$R = (0, d_1 d_2 \ldots d_n \ldots)_2$

Zápis čísel

- V desítkové soustavě (vědecká notace)

  $0{,}000747 = 7{,}47 \cdot 10^{-4}$

  $313{,}815 = 3{,}13815 \cdot 10^2$

- Strojová čísla

  normalizovaná pohyblivá řádová čárka (REAL)

  $\boxed{\mathbf{x} = \pm \mathbf{q} \cdot \mathbf{2^n} \qquad \frac{1}{2} \le \mathbf{q} < 1 \ldots \text{ mantisa}, \quad \mathbf{n} \ldots \text{ exponent}}$

Poznámka: Mnoho reálných čísel, které lze v desítkové soustavě zapsat pomocí konečného počtu cifer, pro zápis ve dvojkové soustavě vyžaduje nekonečně mnoho cifer.

$$(0,7)_{10} = (0,\overline{10110})_2 = 1\cdot 2^{-1} + \sum_{k=0}^{\infty} 1\cdot 2^{-(3+4k)} + \sum_{k=0}^{\infty} 1\cdot 2^{-(4+4k)} =$$

$$= 2^{-1} + 2^{-3}\cdot\sum_{k=0}^{\infty}\left(2^{-4}\right)^k + 2^{-4}\cdot\sum_{k=0}^{\infty}\left(2^{-4}\right)^k = \frac{1}{2} + \frac{1}{8}\cdot\underbrace{\frac{1}{1-\frac{1}{16}}}_{=\frac{16}{15}} + \frac{1}{16}\cdot\frac{16}{15} =$$

$$= \frac{1}{2} + \frac{2}{15} + \frac{1}{15} = \frac{15+4+2}{30} = \frac{21}{30} = \frac{7}{10}$$

Příklad:

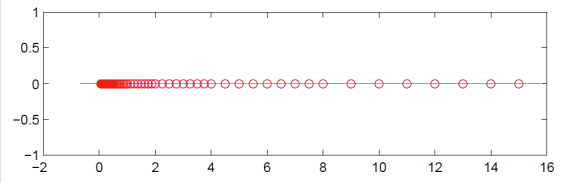
Sestrojte všechna strojová čísla s mantisou délky 4 a exponentem v rozsahu od -3 do 4, tj.

$x = \mathbf{q}\cdot 2^{\mathbf{n}}$, kde $\mathbf{q} = 0,d_1\,d_2\,d_3\,d_4$, $\mathbf{n}\in\{-3,-2,-1,0,1,2,3,4\}$

Abychom si lépe uvědomili jakou mantisou a jakým exponentem je určeno získané číslo, uvedeme si je v následující tabulce.

$\mathbf{q}\setminus\mathbf{n}$	-3	-2	-1	0	1	2	3	4
0.1000_2	0,0625	0,125	0,25	0,5	1	2	4	8
0.1001_2	0.0703125	0,140625	0,28125	0,5625	1,125	2 ,25	4,5	9
0.1010_2	0.078125	0,15625	0,3125	0,625	1,25	2 ,5	5	10
0.1011_2	0.0859375	0,171875	0,34375	0,6875	1,375	2 ,75	5,5	11
0.1100_2	0.09375	0,1875	0,375	0,75	1,5	3	6	12
0.1101_2	0.1015625	0,203125	0,40625	0,8125	1,625	3, 25	6,5	13
0.1110_2	0.109375	0,21875	0,4375	0,875	1,75	3, 5	7	14
0.1111_2	0.1171875	0,234375	0,46875	0,9375	1,875	3, 75	7,5	15

Získaná čísla si je také vhodné vykreslit na číselnou osu, získáme tak přehled o jejich rozložení. Snadno zjistíme, že čísla nejsou rozložena rovnoměrně.



pomocná funkce v MATLABu


```
function [A,P]=stroj_cisla(cisel_mantisy,exponent,zobraz);
%
% [A,P]=stroj_cisla(4,-3:4,1);

for i=1:length(exponent)
 for j=0:2^(cisel_mantisy-1)-1
   zaklad=dec2bin(j);
   zakladstr=num2str(zaklad);
   for k=1:cisel_mantisy-length(zakladstr)-1
     zakladstr=strcat('0',zakladstr);
   end;
   zakladstr=strcat('1',zakladstr);
   zaklad=bin2dec(zakladstr)*2^(-cisel_mantisy);
   A(j+1,i)=zaklad*2^exponent(i);
 end;
end;

[k,l]=size(A);
P=sort(reshape(A,1,k*l));

if zobraz==1
  figure(1);
  plot(P,zeros(size(P)),'ro');
  pr=(P(k*l)-P(1))/20;
  hold on;
  plot([P(1)-pr,P(k*l)+pr],[0 0],'b-');
end;

format short g;
```

Příklad 5:

Uvažujme množinu strojových čísel vygenerovanou v předchozím příkladu (tj. strojová čísla s mantisou délky 4 a exponentem v rozsahu od -3 do 4). Předpokládáme, že počítač zobrazí číslo na nejbližší číslo, které lze zobrazit, v případě shody na větší.

Ukažme si, jak se v tomto stroji sečtou čísla $\frac{1}{10}$ a $\frac{1}{5}$.

Výsledky získané z MATLABu

```
-----------------------------------------------------------------
 Zobrazeni souctu cisel A a B v zadane mnozine strojovych cisel
 s mantisou delky M a exponentem v rozsahu od Exp_min do Exp_max
-----------------------------------------------------------------
 Cislo A = 0.100000
 Cislo B = 0.200000
 Pocet cisel mantisy M = 4
 Rozsah pro exponent:  od -3 do 4

  cislo              zapis obrazu       obraz
-----------------------------------------------------------
A =  0.100000        0.1101 x 2^-3      0.1015625
B =  0.200000        0.1101 x 2^-2      0.203125

obrazA+obrazB=
     0.3046875       0.1010 x 2^-1      0.3125

A+B= 0.300000        0.1010 x 2^-1      0.3125
```

Poznámka:

V tomto příkladě se shodoval obraz přesného výsledku s obrazem součtu obrazů jednotlivých sčítanců.

Příklad 6:

Uvažujme množinu strojových čísel vygenerovanou v předchozím příkladu (tj. strojová čísla s mantisou délky 4 a exponentem v rozsahu od -3 do 4). Předpokládáme, že počítač zobrazí číslo na nejbližší číslo, které lze zobrazit, v případě shody na větší.

Ukažme si, jak se v tomto stroji sečtou čísla $\frac{3}{10}$ a $\frac{1}{6}$.

Výsledky získané z MATLABu

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

```
-----------------------------------------------------------------
 Zobrazeni souctu cisel A a B v zadane mnozine strojovych cisel
 s mantisou delky M a exponentem v rozsahu od Exp_min do Exp_max
-----------------------------------------------------------------
 Cislo A = 0.300000
 Cislo B = 0.166667
 Pocet cisel mantisy M = 4
 Rozsah pro exponent:  od -3 do 4

  cislo              zapis obrazu       obraz
-----------------------------------------------------------
A =  0.300000        0.1010 x 2^-1      0.3125
B =  0.166667        0.1011 x 2^-2      0.171875

obrazA+obrazB=
     0.484375        0.1000 x 2^0       0.5

A+B= 0.466667        0.1011 x 2^0       0.46875
```

Poznámka:

V tomto příkladě se obraz přesného výsledku s obrazem součtu obrazů jednotlivých sčítanců neshodoval !

Chyba výpočtu:

$$\frac{7}{15} - 0,1000_2 \cdot 2^0 = \frac{14-15}{30} = -\frac{1}{30} = -0,0\overline{3}$$

Relativně:

$$\frac{\frac{1}{30}}{\frac{7}{15}} = \frac{1}{14} = 7,14\,\% \qquad !!!$$

Přesnost počítače

- Vymezíme-li pro mantisu 24 bitů, získáme 7 desetinných míst ($2^{24} = 16\,777\,216$).
- Vymezíme-li pro mantisu 32 bitů, získáme 9 desetinných míst ($2^{32} = 4\,294\,967\,296$).

Základní formáty:

Formát	Bytes	Bitů pro mantisu	Bitů pro exponent
Single	4	24	8
Double	8	53	11

Příklad:

- Uvažujme formát SINGLE , tj. 24 bitů pro mantisu.

  $\frac{1}{10} = 0,0\overline{0011}_2 \approx 0,1100\,1100\,1100\,1100\,1100\,1100_2 \cdot 2^{-3}$.

  Chyba zobrazení je $0,\overline{1100}_2 \cdot 2^{-27} (= \frac{1}{10} \cdot 2^{-24}) \approx 5,96 \cdot 10^{-9}$.

- Máme-li počítat $\sum_{k=1}^{100000} \frac{1}{10}$, dostaneme ve formátu SINGLE 9.998,55664.

Chyba musí být větší než $100000 \cdot 5,96 \cdot 10^{-9} = \mathbf{5,96 \cdot 10^{-4}}$.

Ve skutečnosti je chyba ještě větší, neboť se v průběhu výpočtu musí částečně suma zaokrouhlovat dolů nebo nahoru, jak suma roste, později přičítaná čísla $\frac{1}{10}$ jsou oproti sumě menší a jsou tedy počítány s menší přesností (viz následující příklad).

Příklad: Ve formátu SINGLE sečtěte čísla 10000 a 0,1.

```
------------------------------------------------------------------
Prevod cisla 10000 z 10-soustavy do 2-soustavy na 0 desetinnych mist
Cela cast ........... 10000
Desetinna cast ...... 0.000000
------------------------------------------------------------------
prevod_cele_casti =

10000 : 2 = 5000 : 2 = 2500 : 2 = 1250 : 2 = 625 : 2 = 312 : 2 =
    0          0          0          0         1         0


= 156 : 2 = 78 : 2 = 39 : 2 = 19 : 2 = 9 : 2 = 4 : 2 = 2 : 2 = 1
    0        0        1        1       1       0       0

Cislo 10000 v 10-soustave prevedeno do 2-soustavy je 10011100010000.
```

$$(10000)_{10} = (10011100010000)_2 = 0,100111001 \cdot 2^{14}$$

$$(2^{14} = 16384)$$

$$10000 \ \ldots \ 0,1001\,1100\,1000\,0000\,0000\,0000 \cdot 2^{14}$$

$$0,1 \ \ldots \ 0,1100\,1100\,1100\,1100\,1100\,1100 \cdot 2^{-3}$$

$$0,1 \text{ po SHIFTu } \ldots \ 0,0000\,0000\,0000\,0000\,0110\,0110 \cdot 2^{14}$$

$$= (01100110)_2 \cdot 2^{-24} \cdot 2^{14} = (64+32+4+2) \cdot 2^{-10} =$$

$$= \tfrac{102}{1024} = 0,099609375$$

$$10000 + 0,1 \ \ldots \ 0,1001\,1100\,1000\,0000\,0110\,0110 \cdot 2^{14}$$

Číslo 10000 je zobrazeno přesně.
Chyba zobrazení 0,1 po SHIFTU je $\frac{1}{10} - \frac{102}{1024} = 0,1 - 0,099609375 = 3,90625 \cdot 10^{-4}$

Shrnutí:

$$10000 + 0,1 \ \rightarrow \ \text{výsledek s chybou } 3,90625 \cdot 10^{-4}$$

(v sumě z motivačního příkladu jde o jeden krok)

Numerické metody KMA/NM
Josef Daněk 13.2.2o13

skript v MATLABu

```
s=0;
h=single(1/10);

for i=1:100000
  s=s+h;
end;

s
```

Kapitola 2. Nelineární rovnice

Formulace:

Je dána funkce $f : \mathbb{R} \to \mathbb{R}$ definovaná na intervalu $\langle a, b\rangle$. Hledáme $x \in \langle a, b\rangle$ tak, aby $f(x) = 0$.
(x ... kořen rovnice)

Poznámka:

Najít přesné řešení analyticky je možné jen ve velmi jednoduchých případech, např. při řešení lineární rovnice $12x - 3 = 0$, při řešení kvadratické rovnice $4x^2 - 5x + 8 = 0$ nebo např. při řešení rovnice $\sin 5x = \pi$. Proto je nutné pro nalezení kořenů použít nějakou numerickou metodu.

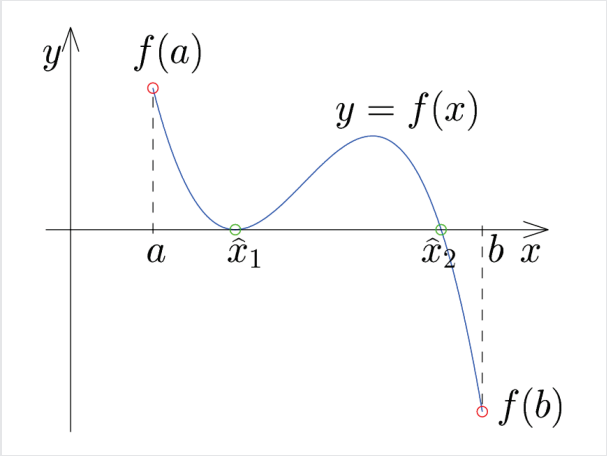
Numerické metody, kterými se budeme zabývat jsou založeny na **iteračních principech**. Pro každou iterační metodu nás budou zajímat odpovědi na dvě otázky:

- Konverguje posloupnost iterací ke hledanému kořenu?
- Jestliže ano, jak rychle?

Věta:

Předpokládejme, že
(i) reálná funkce f je spojitá pro $x \in \langle a, b\rangle$,
(ii) $f(a).f(b) < 0$.
Potom existuje aspoň jedno řešení x rovnice $f(x) = 0$ na $\langle a, b\rangle$.

Větu ilustruje následující obrázek.



Startovací metody

- metoda půlení intervalu
- regula falsi
- metoda prosté iterace

Zpřesňující metody

- Newtonova metoda
- metoda sečen
- Mullerova metoda

Metoda prosté iterace

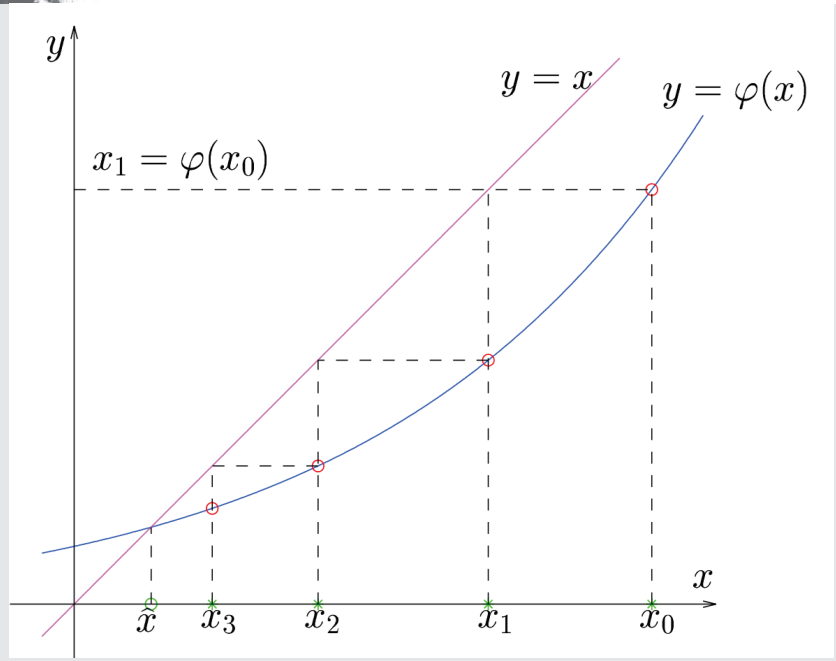
Všechny (jednobodové) iterační metody lze pokládat za speciální případ této metody.

Princip:

- původní rovnici $\boxed{f(x) = 0}$ přepíšeme na tvar $\boxed{x = \varphi(x)}$
- existuje celá řada možností, jak to udělat!
- na konkrétní volbě funkce φ závisí
  konvergence metody
  rychlost konvergence

Algoritnus:

1) Zadáme $x_0 \in \langle a, b\rangle$, $\varepsilon > 0$
2) $x_{k+1} = \varphi(x_k)$
3) Je-li $|x_{k+1} - x_k| < \varepsilon$, pak $x = x_{k+1}$, KONEC
   jinak jdi na 2)



Příklad 1

Metodou prosté iterace najděte na intervalu $\langle 1,4\rangle$ řešení rovnice

$$\boxed{x^2+\ln x-\frac{10}{x}=0.}$$

Za počáteční iteraci volte střed zadaného intervalu, tj. $x_0=2{,}5$.

Řešení

Ukážeme si 4 způsoby přepisu rovnice $f(x)=0$ na tvar $x=\varphi(x)$.

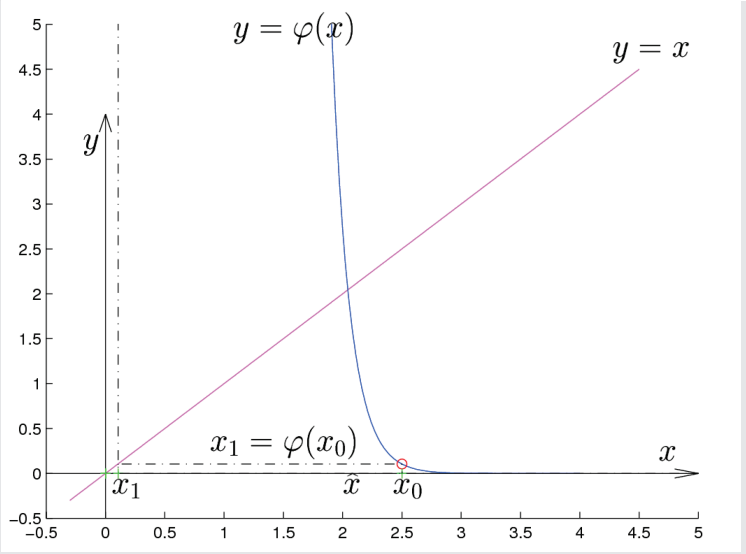
1. způsob:

$$\ln x=\frac{10}{x}-x^2 \quad\Rightarrow\quad \boxed{x=e^{\left(\frac{10}{x}-x^2\right)}} \quad \text{tj.} \quad \varphi(x)=e^{\left(\frac{10}{x}-x^2\right)}$$

k	x_k
0	2.5
1	0.1054
2	$1.5845\cdot 10^{41}$

Již první iterace x_1 je mimo zadaný interval, navíc druhá iterace x_2 je velmi velké číslo a proto metoda prosté iterace nekonverguje.

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

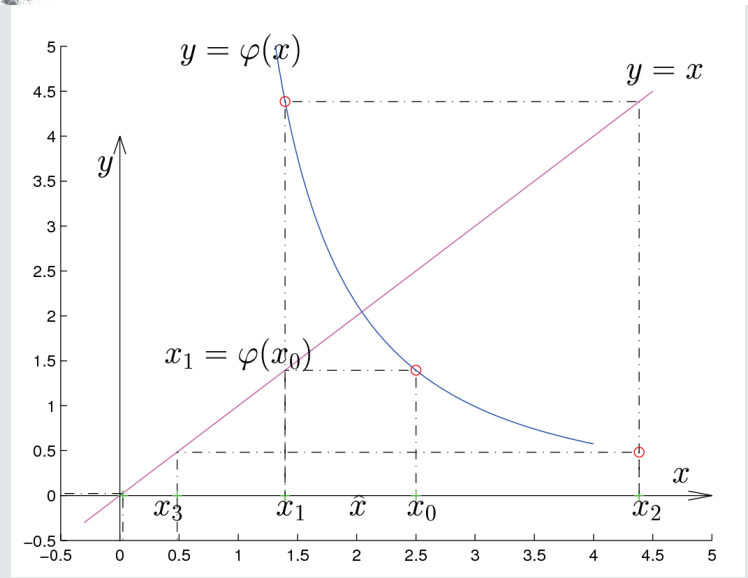


2. způsob:

$$x^2+\ln x=\frac{10}{x} \quad\Rightarrow\quad \boxed{x=\frac{10}{x^2+\ln x}} \quad \text{tj.} \quad \varphi(x)=\frac{10}{x^2+\ln x}$$

k	x_k
0	2.5
1	1.3954
2	4.3852
3	0.4829
4	−20.2122

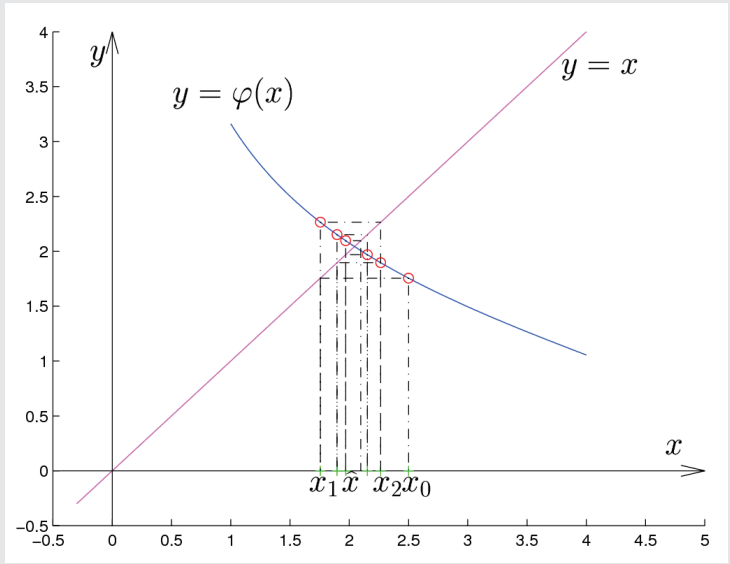
Podobně jako v předchozím případě, zde je 2. iterace x_2 mimo zadaný interval a metoda prosté iterace opět nekonverguje.



3. způsob:

$$x^2 = \frac{10}{x} - \ln x \quad \Rightarrow \quad \boxed{x = \sqrt{\frac{10}{x} - \ln x}} \quad \text{tj.} \quad \varphi(x) = \sqrt{\frac{10}{x} - \ln x}$$

k	x_k
0	2.5
1	1.7560
2	2.2653
3	1.8965
4	2.1524
5	1.9696
6	2.0974
7	2.0067
8	2.0704
9	2.0254
10	2.0571
11	2.0347
12	2.0505
13	2.0393
14	2.0472
15	2.0416
16	2.0455
17	2.0428
18	2.0447
19	2.0434

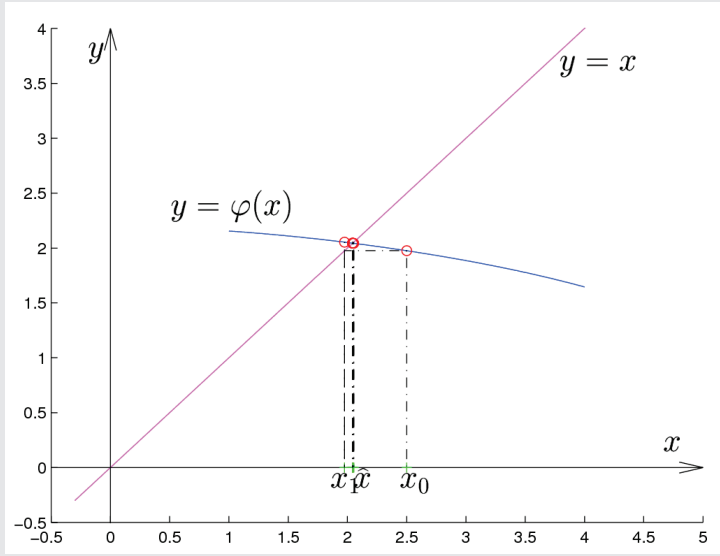


V tomto případě metoda prosté iterace konvergovala k výsledku $\widehat{x}$, rychlost "zahušťování" byla ovšem malá.

4. způsob:

$$x^3 + x\ln x - 10 = 0 \quad \Rightarrow \quad \boxed{x = \sqrt[3]{10 - x\ln x}} \quad \text{tj.} \quad \varphi(x) = \sqrt[3]{10 - x\ln x}$$

k	x_k
0	2.5
1	1.9755
2	2.0532
3	2.0427
4	2.0441
5	2.0439



V tomto posledním případě metoda prosté iterace konvergovala k výsledku $\widehat{x}$ velmi rychle. To dokazuje ten fakt, že kdybychom použili pro zastavení podmínku, aby absolutní hodnota rozdílu dvou po sobě jdoucích iterací byla menší než 10^{-10} potřebovali bychom k tomu pouze 10 iterací.

Poznámka:
Chování metody prosté iterace je závislé na zvoleném předpisu pro funkci $\varphi = \varphi(x)$. Porovnáním grafů z předchozího příkladu lze usoudit, že je vhodné, aby se funkce $\varphi(x)$ co nejvíce blížila konstantní funkci.

Věta (Postačující podmínky konvergence metody prosté iterace.)

Předpokládejme, že je funkce φ na intervalu $I = \langle a, b\rangle$ spojitá a platí:

(a) $\forall x \in I : \varphi(x) \in I$ (**funkce φ zobrazuje I do sebe**),

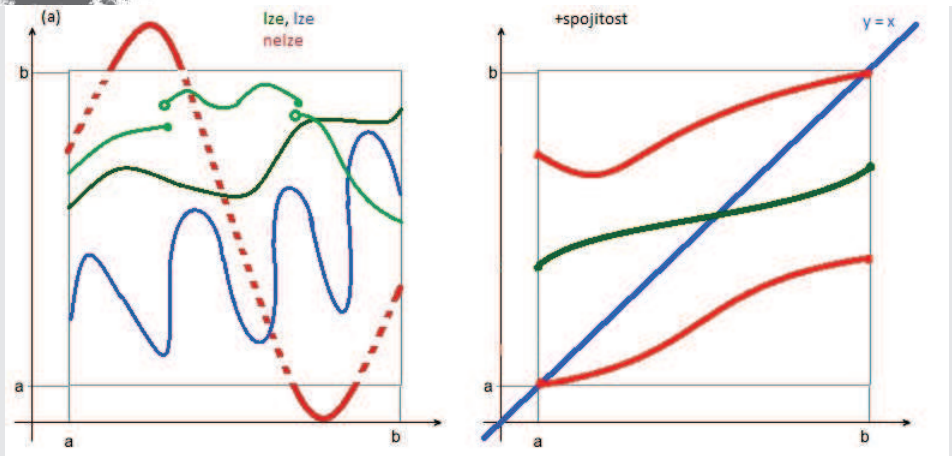
(b) $\exists q \in \langle 0, 1) : \ |\varphi(x) - \varphi(y)| \le q|x - y| \quad \forall x, y \in I$ (**funkce φ je kontrakce**).

Potom

1) v intervalu I existuje právě jeden kořen α rovnice $x = \varphi(x)$,
2) posloupnost $\{x_k\}_{k=1}^{\infty}$ určená formulí $x_k = \varphi(x_{k-1})$ konverguje pro každé $x_0 \in I$ a $\lim_{k\to\infty} x_k = \alpha$.

Důkaz

1) existence je důsledkem podmínky (a) a spojitosti φ



$\Rightarrow \quad \varphi(x) \geq a, \;\; \varphi(x) \leq b \quad \forall x \in \langle a,b \rangle$

$\Rightarrow$ musí platit

$$\varphi(a) \geq a, \; \varphi(b) \leq b$$
$$\varphi(a) \leq b, \; \varphi(b) \geq a$$

jednoznačnost plyne z vlastnosti (b)

DK sporem: Předpokládejme, že existují 2 různé hodnoty $\alpha_1 \neq \alpha_2$ takové, že

$$\alpha_1 = \varphi(\alpha_1), \quad \alpha_2 = \varphi(\alpha_2)$$

Potom platí:

$$|\alpha_2 - \alpha_1| = |\varphi(\alpha_2) - \varphi(\alpha_1)| \underbrace{\leq}_{(b)} q \cdot |\alpha_2 - \alpha_1| \underbrace{<}_{spor} |\alpha_2 - \alpha_1|$$

---

2) Platí:

$$x_k = \varphi(x_{k-1})$$
$$\alpha = \varphi(\alpha)$$

po odečtení:

$$x_k - \alpha = \varphi(x_{k-1}) - \varphi(\alpha)$$

$$|x_k - \alpha| = |\varphi(x_{k-1}) - \varphi(\alpha)| \underbrace{\leq}_{(b)} q \cdot |x_{k-1} - \alpha| \qquad (*)$$

$(*) \Rightarrow$

$$|x_1 - \alpha| \leq q \cdot |x_0 - \alpha|$$
$$|x_2 - \alpha| \leq q \cdot |x_1 - \alpha| \leq q^2 \cdot |x_0 - \alpha|$$
$$\ldots$$

$$\boxed{|x_k - \alpha| \leq \underbrace{q^k}_{(**)} \underbrace{|x_0 - \alpha|}_{konst} \qquad \forall x_0 \in \langle a,b \rangle}$$

$(**)$ $q^k \to 0$ pro $k \to \infty$ $(|q| < 1)$

Poznámka:

Podívejte se na souvislost předpokladů předchozí věty a volby funkce φ v jednotlivých případech příkladu 1.

Poznámka:

Pro diferencovatelnou funkci φ lze podmínku (b) nahradit podmínkou

(b') $\exists q \in \langle 0,1) : \; |\varphi'(x)| \leq q \quad \forall x \in I$

Poznámka:

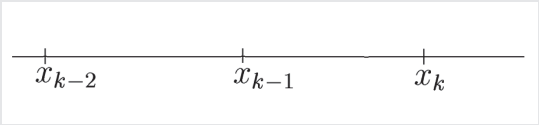
Rychlost konvergence metody prosté iterace je charakterizována $|\varphi'(x_k)|$, jelikož lze psát

$$\varphi'(x_k) \approx \frac{\varphi(x_{k+1}) - \varphi(x_k)}{x_{k+1} - x_k} = \frac{x_{k+2} - x_{k+1}}{x_{k+1} - x_k}.$$

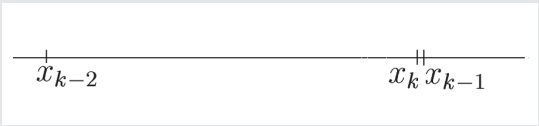
Poznámka:

Souvislost "zahušťování iterací" a hodnotě φ'

a) $q \lessapprox 1 \quad \Rightarrow \quad |x_k - x_{k-1}| \lessapprox |x_{k-1} - x_{k-2}|$

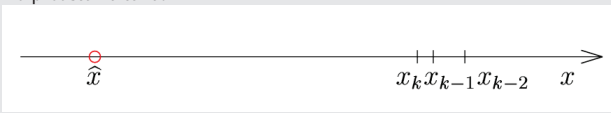


b) $q \approx 0 \quad \Rightarrow \quad |x_k - x_{k-1}| \ll |x_{k-1} - x_{k-2}|$



Poznámka:

Jak bylo řečeno hodnota rozdílu dvou po sobě jdoucích iterací neodpovídá obecně chybě přibližného řešení. Geometricky si to lze představit takto:



Odhad chyby metody prosté iterace

- Máme konvergentní proces $x_k = \varphi(x_{k-1}), \quad k = 1,2,\ldots$
- Přesné řešení α splňuje vztah $\alpha = \varphi(\alpha), \quad \lim_{k \to \infty} x_k = \alpha$

- Po odečtení dostaneme $x_k - \alpha = \varphi(x_{k-1}) - \varphi(\alpha)$

  tj. $|x_k - \alpha| = |\varphi(x_{k-1}) - \varphi(\alpha)|$ $\leftarrow$

- Předpokládáme, že φ je lipchitzovská s konstantou $q \in (0,1)$, tj. musí platit:

  $$|\varphi(x_{k-1}) - \varphi(\alpha)| \leq q \cdot |x_{k-1} - \alpha| \quad \leftarrow$$

- Dále použijeme $\triangle$ nerovnost:

  $$|x_{k-1} - \alpha| = |x_{k-1} - x_k + x_k - \alpha| \leq |x_{k-1} - x_k| + |x_k - \alpha| \quad \leftarrow$$

- Z posledních 3 vztahů dostaneme:

  $$|x_k - \alpha| \leq q \cdot |x_{k-1} - x_k| + q \cdot |x_k - \alpha|$$

  $$(1-q) \cdot |x_k - \alpha| \leq q \cdot |x_{k-1} - x_k| \qquad / \cdot \frac{1}{1-q} > 0$$

  $$\boxed{|x_k - \alpha| \leq \frac{q}{1-q} \cdot |x_{k-1} - x_k|}$$

- Použijeme-li zastavovací podmínku

  $$|x_k - x_{k-1}| < \varepsilon,$$

  potom platí odhad chyby

  $$\boxed{|x_k - \alpha| \leq \frac{q}{1-q}\varepsilon}$$

Příklad 2

Pomocí metody prosté iterace řešte na intervalu $\langle 0,4\rangle$ rovnici

$$x - \sqrt{x+4} = 0.$$

---

přesné řešení:

$$x = \sqrt{x+4} \quad /^2$$
$$x^2 = x + 4$$
$$x^2 - x - 4 = 0$$
$$x_{1,2} = \frac{1 \pm \sqrt{17}}{2} \quad \Rightarrow \quad \underline{x_1 \approx 2,5615}, \quad x_2 \approx -1,5615 \notin \langle 0,4\rangle$$

- Rovnici přepíšeme na tvar: $x = \underbrace{\sqrt{x+4}}_{\varphi(x)}$
- Ověříme splnění předpokladů věty o postačujících podmínkách konvergence metody prosté iterace:

  (a) $\forall x \in \langle 0,4\rangle$: $0 \leq \sqrt{x+4} \leq 4$

  $$0 \leq x + 4 \leq 16$$
  $$-4 \leq x \leq 12$$

(b′) $\forall x \in \langle 0,4\rangle$: $|\varphi'(x)| < 1$

$$\left|\frac{1}{2\sqrt{x+4}}\right| < 1$$
$$\frac{1}{2\sqrt{x+4}} < 1$$
$$1 < 2\sqrt{x+4}$$
$$1 < 4x + 16$$
$$-15 < 4x$$

- Vlastní výpočet:

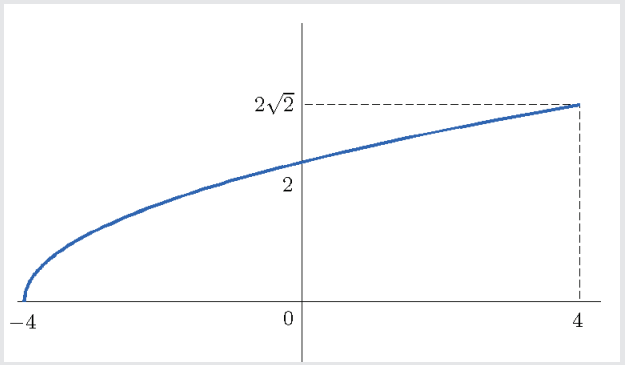
  volíme $x_0 = 2$ a pro zastavovací podmínku hodnotu $\varepsilon = 0.001$.

k	x_k
0	2
1	2.4494
2	2.5395
3	2.5572
4	2.5607
5	2.5613

$\Rightarrow \quad \underline{\tilde{x} = x_5 = 2.5613.}$

- Odhadněme velikost chyby přibližného řešení předchozího příkladu.

  Graf funkce $\varphi(x)$:



Platí:

$\varphi' = \frac{1}{2\sqrt{x+4}}$ ... kladná klesající funkce

$(\varphi'' = (\frac{1}{2}(x+4)^{-\frac{1}{2}})' = -\frac{1}{4}(x+4)^{-\frac{3}{2}} = -\frac{1}{4(x+4)\sqrt{x+4}} < 0)$

$\Downarrow$

$$\max_{0\le x\le 4}|\varphi'(x)| = |\varphi'(0)| = \frac{1}{4} = q \quad \dots \quad \text{podmínka (b')}$$

Zvolili jsme $\varepsilon = 0,001$ a proto platí odhad chyby:

$$\boxed{|x_5 - \alpha| \le \frac{\frac{1}{4}}{1-\frac{1}{4}} . 0,001 = 0,000333}$$

Definice: Říkáme, že posloupnost x_k **konverguje** k číslu α **rychlostí** r, jestliže pro $k \to \infty$

$$|x_{k+1} - \alpha| = c|x_k - \alpha|^r + O(|x_k - \alpha|^{r+1}).$$

Mluvíme o asymptotické rychlosti konvergence ($k \to \infty$).

Poznámka:

$$f(x) = O(g(x)) \text{ pro } x \to a \quad \Leftrightarrow \quad \left|\frac{f(x)}{g(x)}\right| \text{ je omezená (a nenulová) pro } x \to a.$$

Příklady:

a) $\boxed{x^5 \cdot \sin x = O(x^5) \quad \text{pro} \quad x \to \infty} \Leftrightarrow \left|\frac{x^5 \cdot \sin x}{x^5}\right| = |\sin x| \le 1 \quad \text{pro } x \to \infty$

b) $\boxed{x^5 \cdot \sin x = O(x^6) \quad \text{pro} \quad x \to 0} \Leftrightarrow \left|\frac{x^5 \cdot \sin x}{x^6}\right| = \left|\frac{\sin x}{x}\right| \to 1 \quad \text{pro } x \to 0$

Rychlost konvergence metody prosté iterace

Je-li funkce φ dostatečně hladká, můžeme napsat její Taylorův rozvoj v bodě α a potom pro $x = x_{k-1}$ platí:

$$\varphi(x_{k-1}) = \varphi(\alpha) + \varphi'(\alpha)(x_{k-1} - \alpha) + \frac{\varphi''(\alpha)}{2}(x_{k-1} - \alpha)^2 + \frac{\varphi'''(\xi)}{6}(x_{k-1} - \alpha)^3$$

$$x_k - \alpha = \varphi'(\alpha)(x_{k-1} - \alpha) + \frac{\varphi''(\alpha)}{2}(x_{k-1} - \alpha)^2 + \frac{\varphi'''(\xi)}{6}(x_{k-1} - \alpha)^3$$

- je-li $\varphi'(\alpha) \neq 0$, potom
$$x_k - \alpha = \varphi'(\alpha)(x_{k-1} - \alpha)^1 + O((x_{k-1} - \alpha)^2)$$
$\Rightarrow$ rychlost konvergence je řádu 1
- je-li $\varphi'(\alpha) = 0$ a $\varphi''(\alpha) \neq 0$, potom
$$x_k - \alpha = \frac{\varphi''(\alpha)}{2}(x_{k-1} - \alpha)^2 + O((x_{k-1} - \alpha)^3)$$
$\Rightarrow$ rychlost konvergence je řádu 2

Newtonova metoda

Předpoklady:

Nechť v intervalu $I = \langle a, b\rangle$ leží jediný jednoduchý kořen $\hat{x}$ rovnice $f(x) = 0$. Jelikož mluvíme o zpřesňující

metodě, předpokládáme, že máme zadánu nultou iteraci $x_0 \in I$, která je relativně blízko hledanému řešení. Vyjádříme Taylorův rozvoj funkce f v bodě x_0. Přitom předpokládáme, že existují příslušné derivace funkce f.

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(\xi)(x - x_0)^2$$

Rovnici $f(x) = 0$ nahradíme lineární rovnicí

$$f(x_0) + f'(x_0)(x - x_0) = 0$$

Ta má kořen

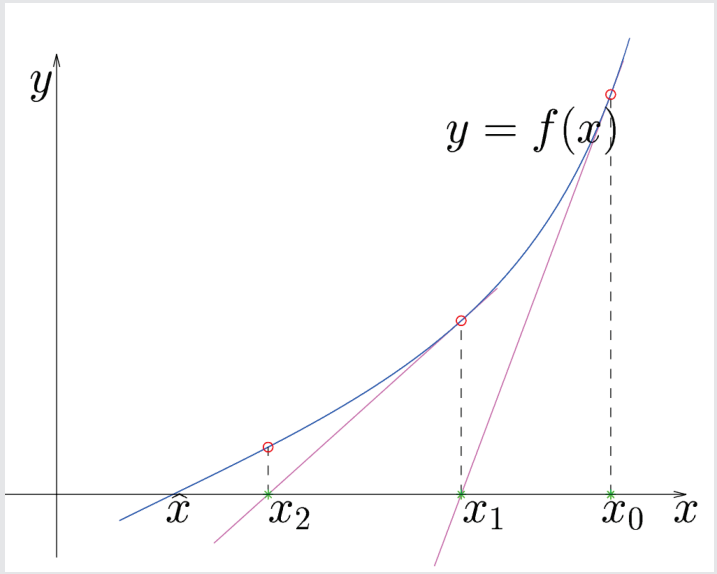
$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Celý postup opakujeme a dostáváme iterační formuli

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Geometrický význam Newtonovy metody:

Křivku $y = f(x)$ nahradíme tečnou ke grafu v bodě x_k a hodnotu x_{k+1} získáme jako průsečík tečny s osou x. Proto se také Newtonova metoda nazývá **metoda tečen** nebo **metoda linearizace**.



Poznámka:

Jako zastavovací podmínku lze např. volit $|x_{k+1} - x_k| < \varepsilon$ nebo $|f(x_k)| < \delta$.

Poznámka:

Algoritmus Newtonovy metody je speciálním případem metody prosté iterace. Za funkci φ jsme volili funkci

$$\varphi(x) = x - \frac{f(x)}{f'(x)}.$$

Rychlost konvergence Newtonovy metody

1. způsob odvození (Newtonova metoda jako speciální případ metody prosté iterace)

Rychlost konvergence závisí na $\varphi'(\alpha)$, resp. $\varphi''(\alpha)$ ... (viz dříve).

Platí:

$$\varphi(x) = x - \frac{f(x)}{f'(x)}$$

$$\varphi' = 1 - \frac{f'.f' - f.f''}{(f')^2} = 1 - 1 + \frac{f.f''}{(f')^2} = \frac{f.f''}{(f')^2}$$

$$\varphi'(\alpha) = 0, \quad \text{protože } f(\alpha) = 0$$

Platí:

$$\varphi'' = \frac{(f'.f'' + f.f'''). (f')^2 - f.f''.2.f'.f''}{(f')^4}$$

$$\varphi''(\alpha) = \frac{(f')^3.f''}{(f')^4} = \frac{f''(\alpha)}{f'(\alpha)}, \quad \text{protože } f(\alpha) = 0$$

Platí tedy $\varphi'(\alpha) = 0$ a obecně $\varphi''(\alpha) \neq 0$ $\Rightarrow$ rychlost konvergence je řádu 2.

2. způsob odvození

Pomocí Taylorova rozvoje funkce f v bodě x_0 (nechť existují $f'(x)$ a $f''(x)$ v I):

$$f(x) = f(x_0) + f'(x_0)\cdot(x - x_0) + \frac{1}{2}\cdot f''(\xi_0)\cdot(x - x_0)^2$$

Dosadíme za x přesné řešení α (tj. $f(\alpha) = 0$)

$$\underbrace{f(\alpha)}_{=0} = f(x_0) + f'(x_0)\cdot(\alpha - x_0) + \frac{1}{2}\cdot f''(\xi_0)\cdot(\alpha - x_0)^2$$

Vydělíme $f'(x_0) \neq 0$:

$$0 = \underbrace{\frac{f(x_0)}{f'(x_0)} + \alpha - x_0}_{= -x_1} + \frac{1}{2}\frac{f''(\xi_0)}{f'(x_0)}(\alpha - x_0)^2$$

$$x_1 - \alpha = \frac{1}{2}\frac{f''(\xi_0)}{f'(x_0)}(\alpha - x_0)^2$$

Proces opakujeme:

$$\underbrace{x_{k+1} - \alpha}_{\text{chyba } k+1 \text{ iterace}} = \frac{1}{2}\frac{f''(\xi_k)}{f'(x_k)} \underbrace{(\alpha - x_k)^2}_{\substack{\text{kvadrát chyby} \\ k\text{-té iterace}}} \qquad (*)$$

$\Rightarrow$ rychlost konvergence $= 2$

Nechť platí

$$\boxed{\frac{1}{2}\left|\frac{f''(\xi)}{f'(\eta)}\right| \leq C \quad \forall \xi, \eta \in I}$$

(Určit C může být obecně problém.)

Označíme-li $\varepsilon_k = x_k - \alpha$ chybu k-té iterace, potom z $(*)$ plyne

$$|\varepsilon_{k+1}| \leq C|\varepsilon_k|^2, \text{ tj. } |C\varepsilon_{k+1}| \leq |C\varepsilon_k|^2$$

$$|C\varepsilon_1| \leq |C\varepsilon_0|^2$$
$$|C\varepsilon_2| \leq |C\varepsilon_1|^2 \leq (|C\varepsilon_0|^2)^2$$
$$|C\varepsilon_3| \leq |C\varepsilon_2|^2 \leq ((|C\varepsilon_0|^2)^2)^2$$
$$\vdots$$
$$|C\varepsilon_k| \leq |C\varepsilon_0|^{2^k}$$

Dostáváme odhad chyby:

$$\boxed{|\varepsilon_k| \leq \frac{1}{C}|C\varepsilon_0|^{2^k}}$$

Postačující podmínka konvergence:

Platí $|\varepsilon_1| \leq C|\varepsilon_0|^2 = \underbrace{C|\varepsilon_0|}_{\#}|\varepsilon_0|$

Pokud bude $\# < 1$, dostaneme kontrakci.

$$|C\varepsilon_0| = \boxed{|C(\alpha - x_0)| < 1}$$

$$|C\underbrace{(\alpha - x_1)}_{\varepsilon_1}| \leq |C\underbrace{(\alpha - x_0)}_{\varepsilon_0}| < 1, \text{ tj. } |C\varepsilon_1| < 1$$

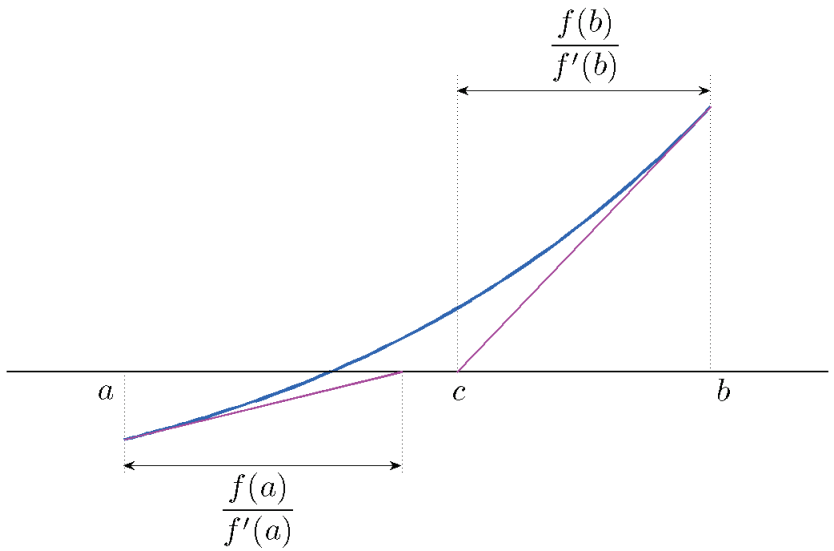
$$\vdots$$

Pro "velkou" hodnotu C musí být počáteční iterace x_0 "velmi přesná".

Věta (Postačující podmínky konvergence Newtonovy metody.)

Je-li $f'(x) \neq 0$, f'' nemění znaménko v $I = \langle a, b\rangle$, platí-li $f(a)\cdot f(b) < 0$ a $\left|\frac{f(a)}{f'(a)}\right| < b - a$, $\left|\frac{f(b)}{f'(b)}\right| < b - a$, potom Newtonova metoda konverguje $\forall x_0 \in I$

Platnost tvrzení lze ověřit pomocí následujícího obrázku.



např. $\dfrac{f(b)-\overbrace{f(c)}^{=0}}{b-c}=f'(b) \quad \Rightarrow \quad b-c=\dfrac{f(b)}{f'(b)}$

Praktické pravidlo pro odhad přesnosti:

Je-li $|\alpha - x_k| < 10^{-d}$ potom $|\alpha - x_{k+1}| < 10^{-2d}$.

(pokud jsou splněny předpoklady pro odvození metody)

Poznámka:

Dosud jsme řešili nelineární rovnici pouze v $\mathbb{R}$. Algoritmus Newtonovy metody můžeme však použít i pro řešení dané rovnice v oboru **komplexních čísel**.

Příklad 1

Newtonovou metodou řešte v komplexním oboru rovnici

$$z^4 + z = 0, \qquad z = x + iy, \quad x, y \in \mathbb{R}$$

Iterační formule bude mít tvar

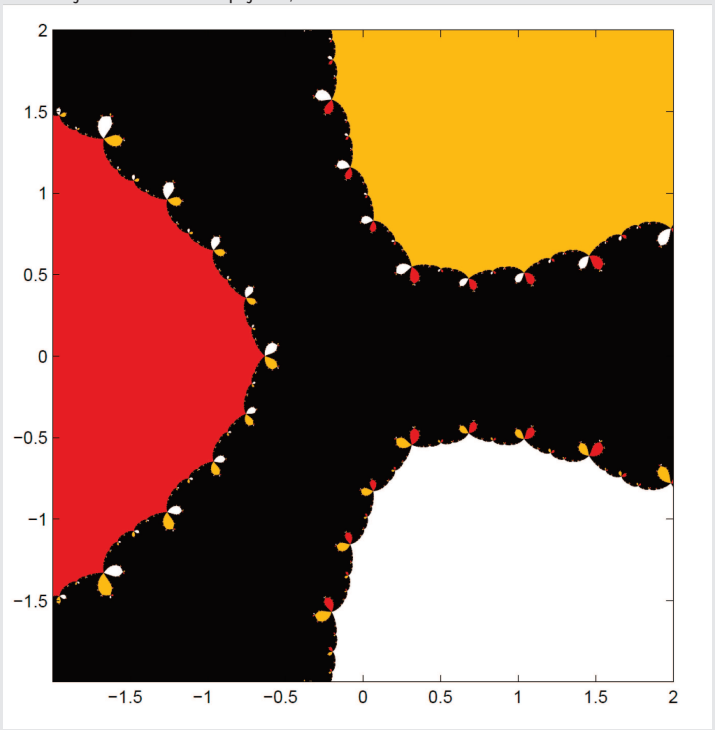
$$z_{k+1} = z_k - \frac{z_k^4 + z_k}{4z_k^3 + 1}$$

Je zřejmé, že daná rovnice bude mít 4 řešení:

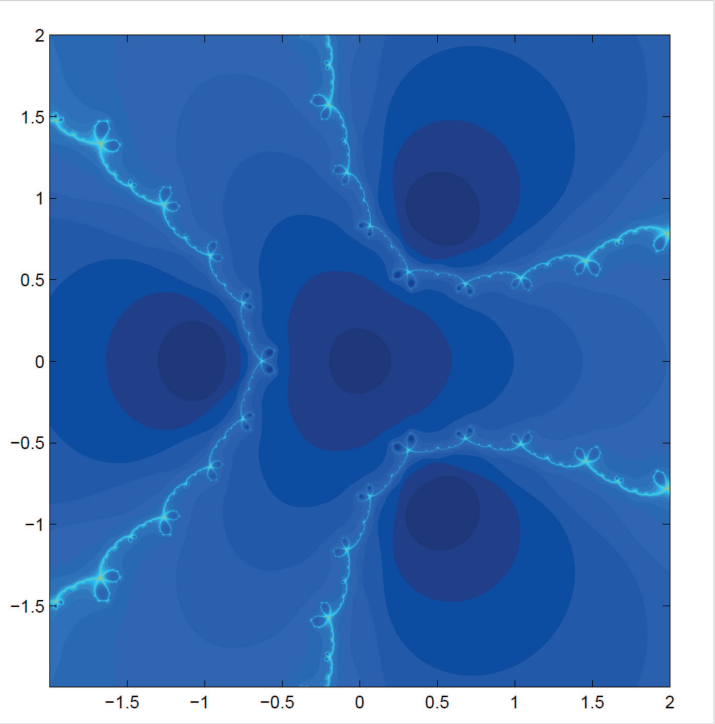
$$0, \quad -1, \quad \frac{1}{2} + \frac{\sqrt{3}}{2}i, \quad \frac{1}{2} - \frac{\sqrt{3}}{2}i$$

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Řešíme-li danou rovnici Newtonovou metodou pro konkrétní počáteční aproximaci ze čtverce $\langle -2; 2\rangle \times \langle -2; 2\rangle$, dostaneme jedno ze čtyř uvedených řešení. Obarvíme-li bod představující počáteční aproximaci různou barvou, podle toho k jakému řešení dospějeme, získáme fraktálovou strukturu.



Pokud vykreslíme pro každý počáteční bod počet iterací nutných k rozhodnutí, ke kterému z možných kořenů metoda konverguje, dostaneme následující obrázek (tmavé odstíny znamenají malý počet iterací, světlé odstíny velký počet iterací).

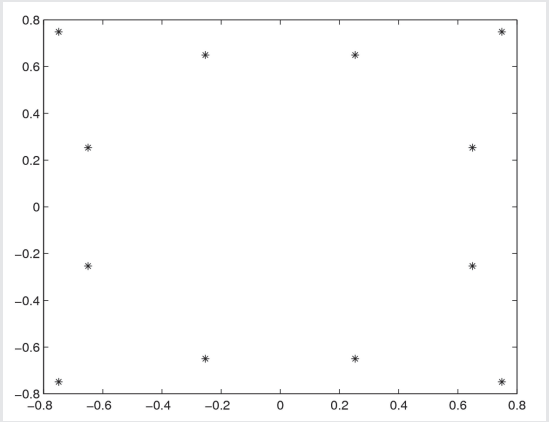


Příklad 2
Newtonovou metodou řešte v komplexním oboru na čtverci $\langle -1.2; 1.2\rangle \times \langle -1.2; 1.2\rangle$ rovnici

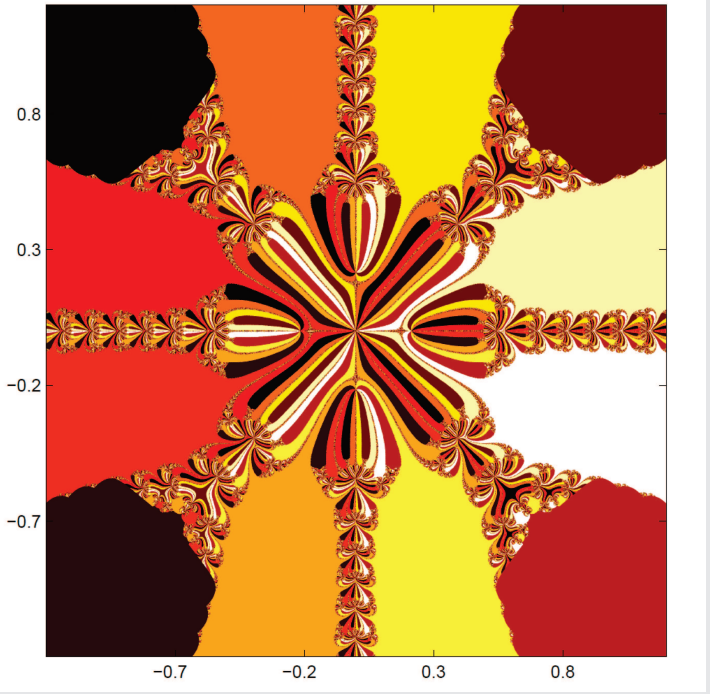
$$z^{12} + 744/611z^8 - 86/16057z^4 + 25/357 = 0$$

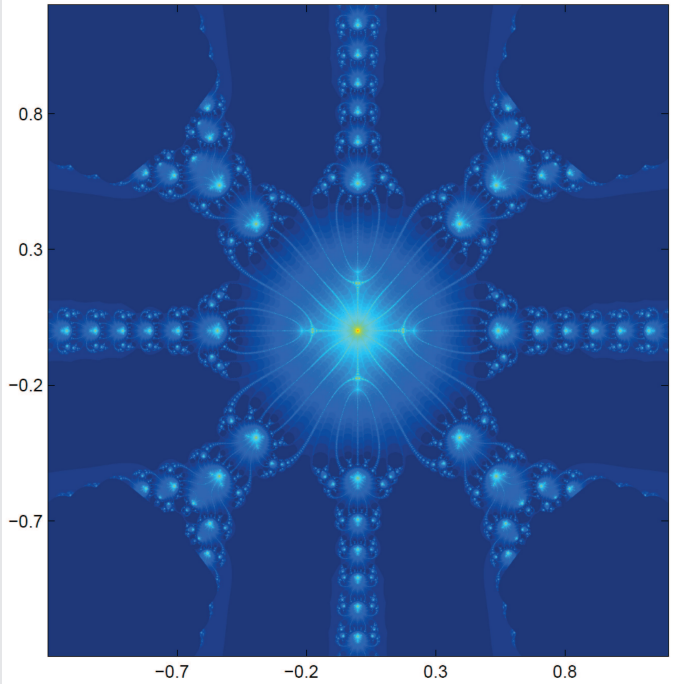
Kořeny:

$-0.75 + 0.75i$
$-0.75 - 0.75i$
$0.75 + 0.75i$
$0.75 - 0.75i$
$-0.65 + 0.25i$
$-0.65 - 0.25i$
$-0.25 + 0.65i$
$-0.25 - 0.65i$
$0.25 + 0.65i$
$0.25 - 0.65i$
$0.65 + 0.25i$
$0.65 - 0.25i$



Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



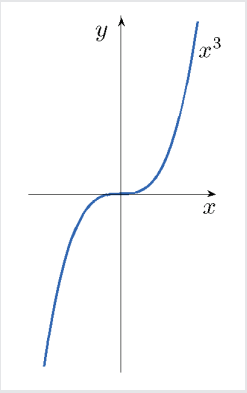


Poznámka:

Při odvozování Newtonovy metody jsme předpokládali, že $f'(\alpha) \neq 0$, tj. α je jednoduchý kořen.

Příklad:

Pomocí Newtonovy metody najděte kořen rovnice $\boxed{x^3 = 0}$.



$\alpha = 0$ ... trojnásobný kořen

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

$$x_{k+1} = x_k - \frac{x_k^3}{3x_k^2}$$

$\boxed{x_{k+1} = \frac{2}{3}x_k}$ $\Rightarrow$ rychlost konvergence je **1** !!!

$$x_{k+1} = \frac{2}{3}x_k \ \wedge \ \alpha = \frac{2}{3}\alpha \quad \Rightarrow \quad (x_{k+1} - \alpha) = \frac{2}{3}(x_k - \alpha)^1$$

Definice: Kořen α rovnice $f(x) = 0$ má **násobnost** s, jestliže $\boxed{0 \neq g(\alpha) < \infty}$, kde $\boxed{g(x) = \frac{f(x)}{(x-\alpha)^s}}$

Modifikovaná iterační formule

$$\boxed{x_{k+1} = x_k - s\frac{f(x_k)}{f'(x_k)}}$$ ... již opět kvadratický iterační proces

pro předchozí příklad: $x_{k+1} = x_k - 3\frac{x_k^3}{3x_k^2} = x_k - x_k = 0$

nevýhoda - musíme znát násobnost s

Jiný přístup pro hledání násobných kořenů

Je-li α s-násobný kořen rovnice $f(x) = 0$, potom je α $(s-1)$-násobným kořenem rovnice $f'(x) = 0$ a tedy jednoduchým kořenem rovnice

$$g(x) = \frac{f(x)}{f'(x)} = 0.$$

D.cv: $g'(x) = ?$

Aitkenův proces

Konverguje-li iterační metoda lineárně, lze pomocí Aitkenova procesu urychlit konvergenci.
Platí:

$$\boxed{\alpha - x_{k+1} = C_{k+1}(\alpha - x_k), \quad |C_k| < 1}$$

kde $|C_k| \to C$ je asymptotická konstanta chyby.

Jsme-li blízko limity, jsou čísla C_k přibližně stejná a lze psát

$$\alpha - x_{k+1} \approx \bar{C}(\alpha - x_k), \ |\bar{C}| = C$$

Pro další iteraci

$$\alpha - x_{k+2} \approx \bar{C}(\alpha - x_{k+1})$$

Po vyloučení $\bar{C}$:

$$\frac{\alpha - x_{k+1}}{\alpha - x_k} \approx \frac{\alpha - x_{k+2}}{\alpha - x_{k+1}}$$

$$(\alpha - x_{k+2})(\alpha - x_k) \approx (\alpha - x_{k+1})^2$$

$$\alpha^2 - \alpha(x_k + x_{k+2}) + x_k x_{k+2} \approx \alpha^2 - 2\alpha x_{k+1} + x_{k+1}^2$$

$$x_k x_{k+2} - x_{k+1}^2 \approx \alpha(x_k - 2x_{k+1} + x_{k+2})$$

$$\boxed{\alpha \approx \frac{x_k x_{k+2} - x_{k+1}^2}{x_k - 2x_{k+1} + x_{k+2}}}$$

Prakticky:

$$x_0, x_1, x_2 \;\to\; \alpha =: x_3$$

$$x_3, x_4, x_5 \;\to\; \alpha =: x_6$$

$$\dots$$

Příklad 3

Pomocí **metody prosté iterace** řešte rovnici $x^2 - x = 0$. Použijte přepis $\boxed{x = \sqrt{x}}$, počáteční iteraci $x_0 = 3$ a zastavovací podmínku $|x_k - x_{k-1}| < 10^{-5}$.

výsledky získané v MATLABu

krok	x(k)	dx(k)=x(k)-x(k-1)	dx(k)/dx(k-1)
0	3.000000		
1	1.732051	-1.267949	
2	1.316074	-0.415977	0.328071
3	1.147203	-0.168871	0.405963
4	1.071075	-0.076127	0.450800
5	1.034928	-0.036148	0.474833
6	1.017314	-0.017614	0.487272
7	1.008620	-0.008694	0.493600
8	1.004301	-0.004319	0.496791
9	1.002148	-0.002153	0.498393
10	1.001073	-0.001075	0.499196
11	1.000537	-0.000537	0.499598
12	1.000268	-0.000268	0.499799
13	1.000134	-0.000134	0.499899
14	1.000067	-0.000067	0.499950
15	1.000034	-0.000034	0.499975
16	1.000017	-0.000017	0.499987
17	1.000008	-0.000008	0.499994

Předchozí výpočet urychlete použitím **Aitkenova procesu**.

výsledky získané v MATLABu

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

krok	x(k)	dx(k)=x(k)-x(k-1)	dx(k)/dx(k-1)
0	3.000000		
1	1.732051	-1.267949	
2	1.316074	-0.415977	0.328071
Zpresneni pomoci Aitkenovy formule			
3	1.112973	-0.203101	0.488251
4	1.054975	-0.057997	0.285559
5	1.027120	-0.027855	0.480285
Zpresneni pomoci Aitkenovy formule			
6	1.001378	-0.025742	0.924133
7	1.000689	-0.000689	0.026771
8	1.000344	-0.000344	0.499742
Zpresneni pomoci Aitkenovy formule			
9	1.000000	-0.000344	0.998968
10	1.000000	-0.000000	0.000344

Příklad 4

Pomocí **Newtonovy metody** řešte rovnici $x^2 - x = 0$. Použijte počáteční iteraci $x_0 = 3$ a zastavovací podmínku $|x_k - x_{k-1}| < 10^{-5}$.

výsledky získané v MATLABu

krok	x(k)	dx(k)=x(k)-x(k-1)	dx(k)/dx(k-1)
0	3.000000		
1	1.800000	-1.200000	
2	1.246154	-0.553846	0.461538
3	1.040603	-0.205551	0.371134
4	1.001525	-0.039078	0.190113
5	1.000002	-0.001522	0.038959
6	1.000000	-0.000002	0.001522

Rychlost konvergence Newtonovy metody je 2, tj. pro urychlení nelze použít Aitkenův proces. Pokud bychom jej použili, výpočet se naopak zpomalí.

výsledky získané v MATLABu

krok	x(k)	dx(k)=x(k)-x(k-1)	dx(k)/dx(k-1)
0	3.000000		
1	1.800000	-1.200000	
2	1.246154	-0.553846	0.461538
Zpresneni pomoci Aitkenovy formule			
3	0.771429	-0.474725	0.857143
4	1.096241	0.324812	-0.684211
5	1.007767	-0.088473	-0.272383
Zpresneni pomoci Aitkenovy formule			
6	1.026707	0.018940	-0.214073
7	1.000677	-0.026030	-1.374350
8	1.000000	-0.000677	0.025995
Zpresneni pomoci Aitkenovy formule			
9	0.999982	-0.000018	0.026688
10	1.000000	0.000018	-0.974664
11	1.000000	-0.000000	-0.000018

Nevýhody Newtonovy metody

- zadaná funkce f musí být diferencovatelná
- derivace se přímo vyskytuje v iterační formuli
- v každé iteraci musíme kromě funkční hodnoty počítat také hodnotu derivace

Pro odbourání poslední vlastnosti můžeme za předpokladu, že se derivace f' na okolí kořene příliž nemění, Newtonovu metodu modifikovat tak, že hodnotu derivace vypočteme pouze jednou, tj. v bodě x_0 a položíme

$$f'(x_k) \approx f'(x_0).$$

Dostaneme iterační formuli **modifikované Newtonovy metody**

$$\boxed{x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)}}.$$

Chceme-li modifikovat Newtonovu metodu pro funkce, které nejsou diferencovatelné, nahradíme v iterační formuli derivaci $f'(x_k)$ diferenčním podílem

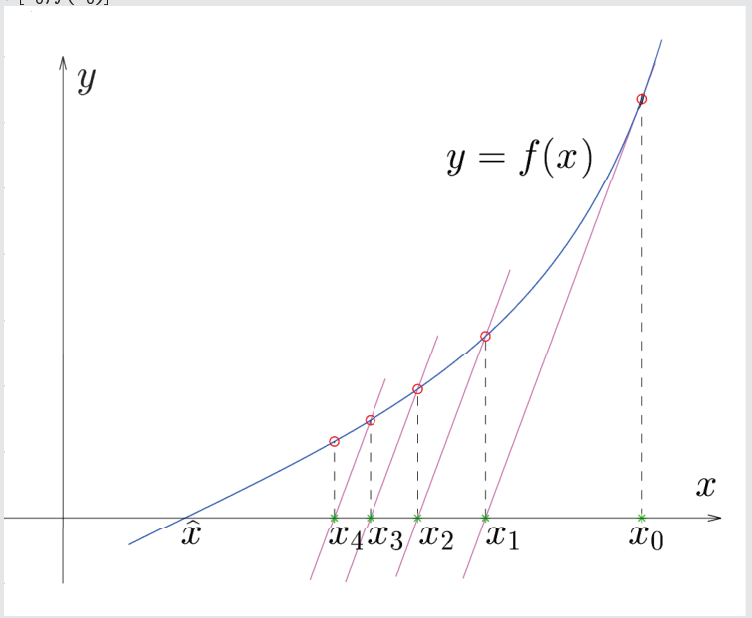
$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}.$$

Dostaneme iterační formuli **metody sečen**

$$\boxed{x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}}.$$

Geometrický význam modifikované Newtonovy metody

Tečny ke grafu v bodech $[x_k, f(x_k)]$ nahrazujeme přímkami rovnoběžnými s tečnou ke grafu funkce $y = f(x)$ v bodě $[x_0, f(x_0)]$.



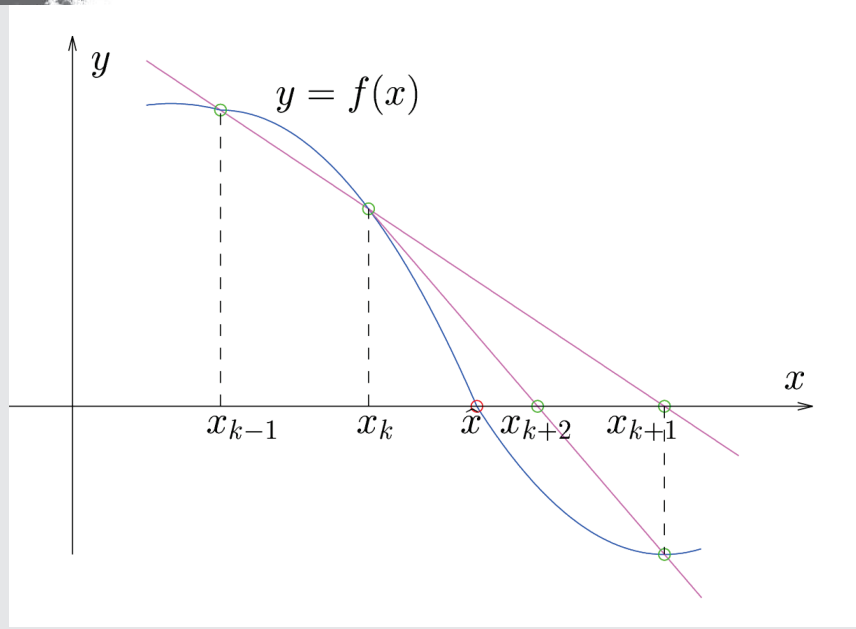
Poznámka:

V této modifikaci počítáme pouze jednu hodnotu derivace $f'(x_0)$, a proto je tento postup vhodný je-li derivace $f'(x)$ složitá. Nemění-li $f'(x)$ a $f''(x)$ znaménko, je možné dokázat konvergenci této metody.

Geometrický význam metody sečen

Mějme dvě dobré aproximace x_{k-1} a x_k kořene $\hat{x}$ rovnice $f(x) = 0$. Křivku $y = f(x)$ nahradíme přímkou (sečnou), která prochází body $[x_{k-1}, f(x_{k-1})]$ a $[x_k, f(x_k)]$.

Další iteraci x_{k+1} získáme jako průsečík sečny s osou x.



Poznámka:

Pro zahájení výpočtu potřebujeme znát **2** počáteční aproximace, ale na rozdíl od Newtonovy metody počítáme v každém kroku pouze jednu novou funkční hodnotu, což je úspora času.

Poznámka:

Metoda sečen má obdobný algoritmus jako metoda regula falsi, nepožadujeme však splnění podmínky $f(x_{k-1}) \cdot f(x_k) < 0$.

Rychlost konvergence metody sečen

Odvozuje se podobně jako u Newtonovy metody. Nechť platí

$$\boxed{\frac{1}{2}\left|\frac{f''(\xi)}{f'(\eta)}\right| \le C \qquad \forall \xi, \eta \in I}.$$

Potom

$$|\varepsilon_{k+1}| \le C \cdot |\varepsilon_k| \cdot |\varepsilon_{k-1}|.$$

Náznak odvození:

Označme $d_k = C|\varepsilon_k|$, potom

$$d_{k+1} \le d_k\, d_{k-1}.$$

Nechť čísla d_0 a d_1 jsou rovna číslu $d < 1$ nebo menší, potom dosazováním dostáváme

$$d_2 \le d_1\, d_0 \le d\, d = d^2$$
$$d_3 \le d_2\, d_1 \le d^2\, d = d^3$$
$$d_4 \le d_3\, d_2 \le d^3\, d^2 = d^5$$
$$d_5 \le d_4\, d_3 \le d^5\, d^3 = d^8$$
$$\vdots$$
$$d_{k+1} \le d^{n_{k+1}}, \quad \text{kde } n_{k+1} = n_k + n_{k-1},\ n_1 = 1,\ n_0 = 1.$$

Rekurentně daná posloupnost n_k definuje tzv. **Fibonacciova čísla**. Odpovídající charakteristická rovnice je

$$\varrho^2 = \varrho + 1.$$

Pro její kořeny platí:

$$\varrho_{1,2} = \frac{1 \pm \sqrt{5}}{2} \doteq \begin{cases} 1{,}618 \\ -0{,}618 \end{cases}.$$

Snadno se ověří, že explicitní vyjádření pro n_k je následující

$$n_k = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{k+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{k+1}\right].$$

Hledaný řád metody r potom získáme ze vztahu

$$d_{k+1} \approx d_k^r,$$

tj. po zlogaritmování

$$r = \frac{\log d_{k+1}}{\log d_k} = \frac{\log d^{n_{k+1}}}{\log d^{n_k}} = \frac{n_{k+1}}{n_k}, \quad \text{pro } k \to \infty.$$

$$\lim_{k\to\infty} \frac{n_{k+1}}{n_k} = \lim_{k\to\infty} \frac{\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{k+2} - \left(\frac{1-\sqrt{5}}{2}\right)^{k+2}\right]}{\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{k+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{k+1}\right]} = \frac{1+\sqrt{5}}{2}$$

Pro rychlost konvergence tedy dostáváme

$$r = \frac{1}{2}(1+\sqrt{5}) \doteq 1{,}618 \qquad (\text{pro } k \to \infty).$$

Poznámka:

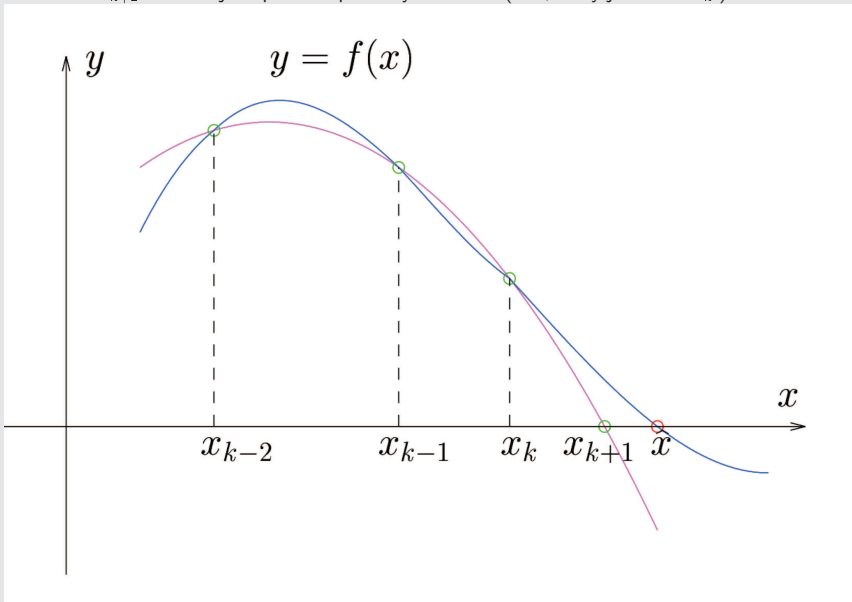
Metoda sečen je tzv. dvoukroková interpolační metoda, analogicky lze odvodit tříkrokovou interpolační metodu, kterou nazýváme **Mullerova metoda**.

Geometrický význam Mullerovy metody

Mějme tři dobré aproximace x_{k-2}, x_{k-1} a x_k kořene x rovnice

$$f(x) = 0.$$

Křivku $y = f(x)$ nahradíme parabolou (kvadratickou funkcí), která prochází body $[x_{k-2}, f(x_{k-2})]$ $[x_{k-1}, f(x_{k-1})]$ a $[x_k, f(x_k)]$.
Další iteraci x_{k+1} získáme jako průsečík paraboly s osou x. (Ten, který je blíže k x_k.)



Prostředky MATLABu pro řešení nelineárních rovnic

- `fzero` pro obecnou nelineární rovnici
- `roots` pro kořeny polynomu

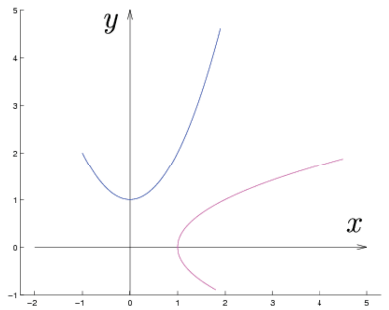
Příklad:

Řešte soustavu dvou rovnic pro dvě neznámé ($a \in \mathbb{R}$ ... parametr)

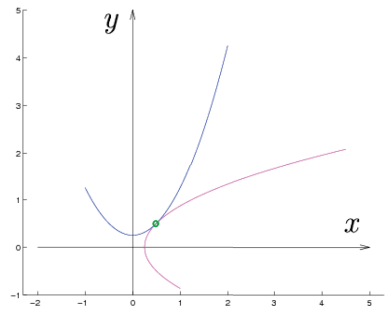
$$x^2 - y + a = 0$$
$$-x + y^2 + a = 0$$

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

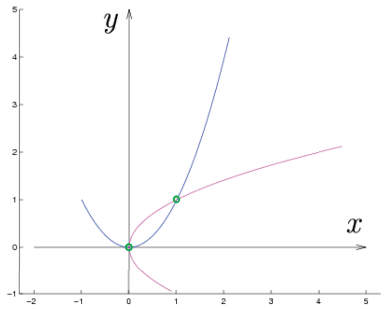
1) $a = 1$



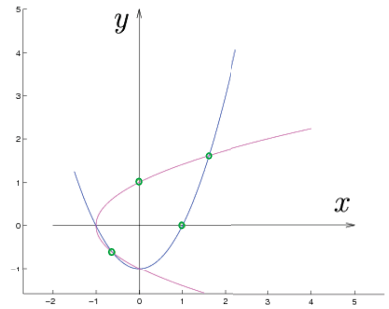
2) $a = \frac{1}{4}$



3) $a = 0$



4) $a = -1$



Formulace:

Jsou dány funkce $F_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, n$ definované na $\langle a_i, b_i \rangle$.
Označme $I = \langle a_1, b_1 \rangle \times \langle a_2, b_2 \rangle \times \cdots \times \langle a_n, b_n \rangle$.
Hledáme $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T \in I$ tak, aby

$$F_1(x_1, x_2, \ldots, x_n) = 0$$
$$F_2(x_1, x_2, \ldots, x_n) = 0$$
$$\vdots$$
$$F_n(x_1, x_2, \ldots, x_n) = 0$$

Vektorově:

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \quad \text{kde } \mathbf{F} = (F_1, F_2, \ldots, F_n)^T.$$

Věta (Postačující podmínky konvergence metody prosté iterace.)

Předpokládejme, že je funkce Φ na I spojitá a platí:

(a) $\forall \mathbf{x} \in I: \ \Phi(\mathbf{x}) \in I$ (funkce Φ zobrazuje I do sebe),

(b) $\exists q \in \langle 0,1): \ ||\Phi(\mathbf{x}) - \Phi(\mathbf{y})|| \leq q||\mathbf{x}-\mathbf{y}|| \quad \forall \mathbf{x},\mathbf{y} \in I$ (funkce Φ je kontrakce).

Potom

1. v množině I existuje právě jedno řešení $\mathbf{x}$ soustavy rovnic $\mathbf{x} = \Phi(\mathbf{x})$,
2. posloupnost $\{\mathbf{x}^k\}_{k=1}^{\infty}$ určená formulí $\mathbf{x}^k = \Phi(\mathbf{x}^{k-1})$ konverguje pro každé $\mathbf{x}^0 \in I$ a $\lim_{k\to\infty} \mathbf{x}^k = \mathbf{x}$.

Metoda prosté iterace

Soustavu rovnic $\boxed{\mathbf{F}(\mathbf{x}) = \mathbf{0}}$ nahradíme soustavou rovnic $\boxed{\mathbf{x} = \Phi(\mathbf{x})}$ (více možností).

Algoritmus:

1) Zadáme $\mathbf{x}^0 \in I$, $\varepsilon > 0$
2) $\mathbf{x}^{k+1} = \Phi(\mathbf{x}^k)$
3) Je-li $||\mathbf{x}^{k+1} - \mathbf{x}^k|| < \varepsilon$, pak $\mathbf{x} = \mathbf{x}^{k+1}$, KONEC
   jinak jdi na 2)

Příklad 5

Řešte metodou prosté iterace soustavu dvou rovnic pro dvě neznámé

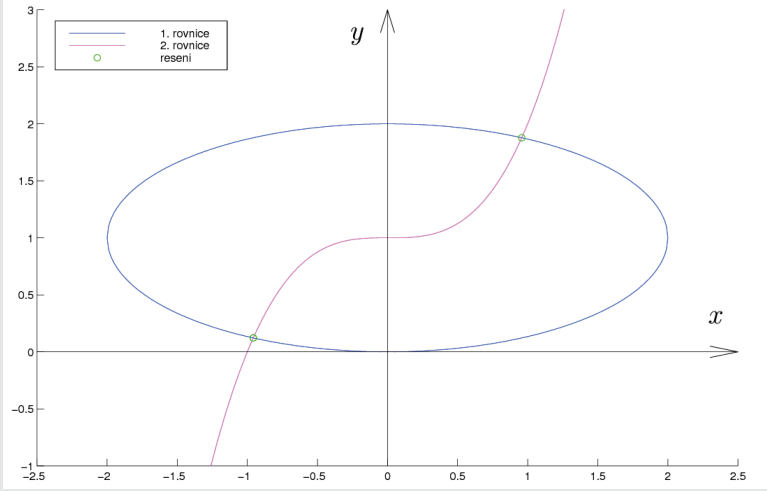
$$x^2 + 4y^2 - 8y = 0$$
$$x^3 - y + 1 = 0$$

---

1. rovnice je rovnicí eplipsy $x^2 + 4(y-1)^2 = 4$

$$\left(\frac{x}{2}\right)^2 + (y-1)^2 = 1$$

2. rovnici upravíme na tvar $y = x^3 + 1$



Z 2.rovnice vyjádříme x:

$x^3 = y - 1$

$x = \sqrt[3]{y-1}$

Z 1.rovnice vyjádříme y:

$4y^2 = 8y - x^2$

$y = \frac{1}{2}\sqrt{8y - x^2}$

Rekurentní formule:

$x_{k+1} = \sqrt[3]{y_k - 1}$

$y_{k+1} = \frac{1}{2}\sqrt{8y_k - x_{k(+1)}^2}$

výsledky získané v MATLABu

krok	x_1(k)	x_2(k)	||x(k)-x(k-1)||
0	1.000000	1.000000	
1	0.000000	1.322876	1.050832
2	0.686033	1.626577	0.750250
3	0.855706	1.770732	0.222643
4	0.916856	1.832595	0.086985
5	0.940758	1.858772	0.035448
6	0.950516	1.869836	0.014752
7	0.954580	1.874514	0.006197
8	0.956288	1.876492	0.002613
9	0.957009	1.877328	0.001104
10	0.957313	1.877682	0.000467

výsledky získané v MATLABu

```
| krok |      x_1(k)      |      x_2(k)      |  ||x(k)-x(k-1)||  |
-------------------------------------------------------------------
|    0 |    -1.000000     |      0.000000    |                   |
|    1 |     0.500000     |      0.000000    |      1.802776     |
|    2 |     0.651123     |      0.677644    |      2.108913     |
|    3 |     0.670383     |      1.241743    |      1.749676     |
|    4 |     0.716783     |      1.573937    |      1.015477     |
|    5 |     0.838816     |      1.743370    |      0.454007     |
|    6 |     0.906819     |      1.820267    |      0.181812     |
|    7 |     0.936234     |      1.853465    |      0.074113     |
|    8 |     0.948577     |      1.867581    |      0.030794     |
|    9 |     0.953759     |      1.873559    |      0.012921     |
|   10 |     0.955941     |      1.876088    |      0.005446     |
|   11 |     0.956862     |      1.877158    |      0.002300     |
|   12 |     0.957251     |      1.877610    |      0.000972     |
```

Poznámka:

Pozor na definiční obory funkcí.
Pozor na zápis funkcí v MATLABu.

```
>> (-8)^(1/3)
ans =
   1.0000 + 1.7321i
```

Poznámka: Pokud chceme v předchozím příkladě najít druhé řešení, je třeba zvolit jiný předpis pro funkci Φ.

výsledky získané v MATLABu


```
Metoda proste iterace pro reseni soustavy nelinearnich rovnic
   F(x)=0    s vyuzitim prepisu na tvar    x=Phi(x)
pro pocatecni aproximaci x0=[1,1] a
zastavovaci podminku ||x(k)-x(k-1)||<0.001

Funkce Phi(x) je zadana takto:

function out=Phi(in);
x=in(1);
y=in(2);
if (y-1)>=0
 out(1)=(y-1)^(1/3);
else
 out(1)=-(1-y)^(1/3);
end;
out(2)=(x^2+4*y^2)/8;
out=out';

| krok |      x_1(k)      |      x_2(k)      |  ||x(k)-x(k-1)||  |
-------------------------------------------------------------------
|    0 |     1.000000     |     1.000000     |                   |
|    1 |     0.000000     |      0.625000    |      1.068000     |
|    2 |    -0.721125     |      0.195312    |      0.839436     |
|    3 |    -0.930127     |      0.084076    |      0.236761     |
|    4 |    -0.971150     |      0.111677    |      0.049444     |
|    5 |    -0.961296     |      0.124127    |      0.015879     |
|    6 |    -0.956783     |      0.123215    |      0.004604     |
|    7 |    -0.957116     |      0.122020    |      0.001240     |
|    8 |    -0.957550     |      0.121953    |      0.000440     |
```

Newtonova metoda

Odvození je opět analogické případu funkce jedné reálné proměnné.
Vyjádříme si Taylorův rozvoj funkce $\mathbf{F}$ v bodě $\mathbf{x}^k$.
(Předpokládáme, že existují derivace !)
Soustavu rovnic $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ nahradíme soustavou lineárních rovnic

$$\mathbf{F}(\mathbf{x}^k) + \mathbf{F}'(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k) = \mathbf{0}$$

Její řešení označíme $\mathbf{x}^{k+1}$, tj.

$$\mathbf{F}(\mathbf{x}^k) + \mathbf{F}'(\mathbf{x}^k)\underbrace{(\mathbf{x}^{k+1} - \mathbf{x}^k)}_{\mathbf{h}^k} = \mathbf{0}$$

Dostáváme soustavu

$$\mathbf{F}'(\mathbf{x}^k)\mathbf{h}^k = -\mathbf{F}(\mathbf{x}^k),$$

která má řešení

$$\mathbf{h}^k = -[\mathbf{F}'(\mathbf{x}^k)]^{-1}\mathbf{F}(\mathbf{x}^k)$$

Novou iteraci $\mathbf{x}^{k+1}$ získáme ze vztahu

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{h}^k$$

Poznámka:

$\mathbf{F}'(\mathbf{x}^k)$ je Jacobiho matice funkce $\mathbf{F}(\mathbf{x})$ v bodě $\mathbf{x}^k$.
Je zřejmé, že musí být regulární (musí existovat matice k ní inverzní).

Příklad 6

Řešte Newtonovou metodou soustavu dvou rovnic pro dvě neznámé

$$x^2 + 4y^2 - 8y = 0$$
$$x^3 - y + 1 = 0$$

$$\mathbf{F}(x,y) = \begin{bmatrix} x^2 + 4y^2 - 8y \\ x^3 - y + 1 \end{bmatrix}$$

$$\frac{\partial F_1(x,y)}{\partial x} = 2x \qquad \frac{\partial F_1(x,y)}{\partial y} = 8y - 8$$

$$\frac{\partial F_2(x,y)}{\partial x} = 3x^2 \qquad \frac{\partial F_2(x,y)}{\partial y} = -1$$

$$\mathbf{F}'(x,y) = \begin{bmatrix} 2x & 8y-8 \\ 3x^2 & -1 \end{bmatrix}$$

---

1. iterace

$$\begin{bmatrix} x^1 \\ y^1 \end{bmatrix} = \begin{bmatrix} x^0 \\ y^0 \end{bmatrix} + \begin{bmatrix} h_1^0 \\ h_2^0 \end{bmatrix}$$

Platí

$$\mathbf{h}^0 = -[\mathbf{F}'(x^0,y^0)]^{-1}\mathbf{F}(x^0,y^0)$$

Abychom nemuseli počítat inverzní matici, vypočteme $\mathbf{h}^0$ jako řešení soustavy

$$\mathbf{F}'(x^0,y^0)\mathbf{h}^0 = -\mathbf{F}(x^0,y^0)$$

$$\mathbf{F}(x^0,y^0) = \begin{bmatrix} 4 \\ 7 \end{bmatrix} \qquad \mathbf{F}'(x^0,y^0) = \begin{bmatrix} 4 & 8 \\ 12 & -1 \end{bmatrix}$$

Dostaneme

$$\mathbf{h}^0 = \begin{bmatrix} -0.6 \\ -0.2 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} x^1 \\ y^1 \end{bmatrix} = \begin{bmatrix} x^0 \\ y^0 \end{bmatrix} + \begin{bmatrix} h_1^0 \\ h_2^0 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} -0.6 \\ -0.2 \end{bmatrix} = \begin{bmatrix} 1,4 \\ 1,8 \end{bmatrix}$$

---

2. iterace

Numerické metody KMA/NM
Josef Daněk 13.2.2o13

$$\begin{bmatrix} x^2 \\ y^2 \end{bmatrix} = \begin{bmatrix} x^1 \\ y^1 \end{bmatrix} + \begin{bmatrix} h_1^1 \\ h_2^1 \end{bmatrix}$$

Platí

$$\mathbf{h}^1 = -[\mathbf{F}'(x^1,y^1)]^{-1}\mathbf{F}(x^1,y^1)$$

Abychom nemuseli počítat inverzní matici, vypočteme $\mathbf{h}^1$ jako řešení soustavy

$$\mathbf{F}'(x^1,y^1)\mathbf{h}^1 = -\mathbf{F}(x^1,y^1)$$

$$\mathbf{F}(x^1,y^1) = \begin{bmatrix} 0,52 \\ 1,944 \end{bmatrix} \qquad \mathbf{F}'(x^1,y^1) = \begin{bmatrix} 2,8 & 6,4 \\ 5,88 & -1 \end{bmatrix}$$

Dostaneme

$$\mathbf{h}^1 = \begin{bmatrix} -0,3206 \\ 0,0590 \end{bmatrix}$$

$$\begin{bmatrix} x^2 \\ y^2 \end{bmatrix} = \begin{bmatrix} x^1 \\ y^1 \end{bmatrix} + \begin{bmatrix} h_1^1 \\ h_2^1 \end{bmatrix} = \begin{bmatrix} 1,4 \\ 1,8 \end{bmatrix} + \begin{bmatrix} -0,3206 \\ 0,0590 \end{bmatrix} = \begin{bmatrix} 1,0794 \\ 1,8590 \end{bmatrix}$$

Další iterace bychom počítali podobně. V následují tabulce jsou shrnuty 4. iterace Newtonovy metody.

k	x_k	y_k	$\|\|\mathbf{x}_k - \mathbf{x}_{k-1}\|\|$
0	2.0000	2.0000	
1	1.4000	1.8000	0.6325
2	1.0794	1.8590	0.3260
3	0.9703	1.8763	0.1105
4	0.9577	1.8779	0.0127

Normy vektorů a matic

Připomenutí:

Norma je zobrazení n lineárního vektorového prostoru $\mathcal{L}$ do $\mathbb{R}_0^+$:

1. $n(x) = 0 \Leftrightarrow x = 0$ (definitnost)
2. $n(\lambda x) = |\lambda| n(x)$ (homogenita)
3. $n(x+y) \leq n(x) + n(y)$ ($\triangle$ nerovnost)

VEKTORY:

$||\mathrm{x}||_p = (\sum_i |x_i|^p)^{\frac{1}{p}}$
... p-tá vektorová norma

$||\mathrm{x}||_1 = \sum_i |x_i|$ ... první vektorová norma

$||\mathrm{x}||_2 = \sqrt{\sum_i x_i^2}$ ... euklidovská vektorová norma

$||\mathrm{x}||_\infty = \max_i |x_i| = \lim_{p\to\infty} (\sum_i |x_i|^p)^{\frac{1}{p}}$
... maximová norma

MATICE:

$\leftrightarrow$ $||\mathbf{A}||_S = \max_k\{\sum_i |a_{ik}|\}$ ... sloupcová maticová norma

$\leftrightarrow$ $||\mathbf{A}||_{SP} = \max_k\{\lambda_k^{\frac{1}{2}}(\mathbf{A}^H\mathbf{A})\}$ ... spektrální norma

$\leftrightarrow$ $||\mathbf{A}||_R = \max_i\{\sum_k |a_{ik}|\}$ ...řádková maticová norma

Poznámky:

$\mathbf{A}^H$ ... hermitovsky transponovaná matice; $\mathbf{A}^H = [a_{ij}^H] = [\bar{a}_{ji}]$; $\bar{a}$ je komplexně sdružené číslo k číslu a

Symbol $\leftrightarrow$ značí vazbu mezi vektorovou a maticovou normou.
Příslušná maticová norma **je generovaná** příslušnou vektorovou normou.

Příklad

Pro zadanou matici $\mathbf{A}$ a vektor $\mathbf{x}$ určete výše uvedené normy.

$$\mathbf{A} = \begin{bmatrix} 2 & -1 \\ 0 & 3 \end{bmatrix}, \qquad \mathrm{x} = \begin{bmatrix} 6 \\ -1 \end{bmatrix}.$$

$||\mathbf{A}||_S = \max\{|2| + |0|; |-1| + |3|\} = \max\{2; 4\} = 4$

$||\mathbf{A}||_R = \max\{|2| + |-1|; |0| + |3|\} = \max\{3; 3\} = 3$

$||\mathbf{A}||_{SP}$:

$$\mathbf{A}^H\mathbf{A} = \begin{bmatrix} 2 & 0 \\ -1 & 3 \end{bmatrix} \cdot \begin{bmatrix} 2 & -1 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 4 & -2 \\ -2 & 10 \end{bmatrix}$$

$$\det(\mathbf{A}^H\mathbf{A} - \lambda\mathbf{I}) = \begin{vmatrix} 4-\lambda & -2 \\ -2 & 10-\lambda \end{vmatrix} =$$
$$= (4-\lambda)(10-\lambda) - 4 = \lambda^2 - 14\lambda + 36$$

$$\lambda_{1,2}(\mathbf{A}^H\mathbf{A}) = \frac{14 \pm \sqrt{14^2 - 4\cdot 36}}{2} =$$
$$= 7 \pm \sqrt{7^2 - 36} = 7 \pm \sqrt{13}$$

$$\max|\lambda_{1,2}| = 7 + \sqrt{13}$$

$||\mathbf{A}||_{SP} = \sqrt{7+\sqrt{13}} \doteq \mathbf{3,2566}$

$||\mathrm{x}||_1 = |6| + |-1| = \mathbf{7}$

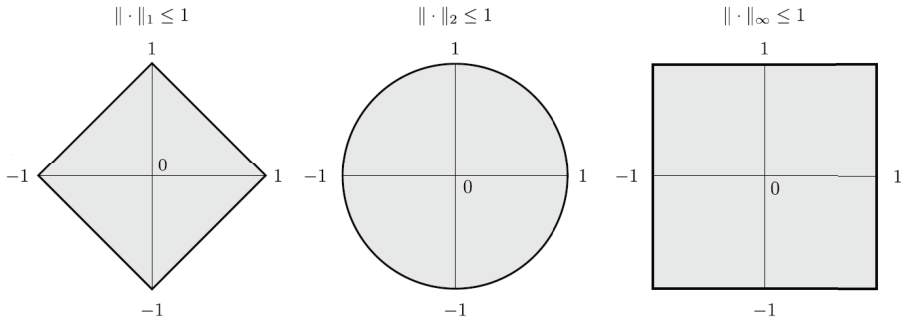
$||\mathrm{x}||_\infty = \max\{|6|, |-1|\} = \mathbf{6}$

$||\mathrm{x}||_2 = \sqrt{6^2 + (-1)^2} = \sqrt{37} \doteq \mathbf{6,0827}$

Numerické metody KMA/NM
Josef Daněk 13.2.2o13

Geometrický význam vektorových norem

jednotkové koule v $\mathbb{R}^2$ ... množina (bodů) prvků s normou ≤ 1:



1. $||\mathrm{x}||_1 = \sum_i |x_i| = |x| + |y| \leq 1$
2. $||\mathrm{x}||_2 = \sqrt{\sum_i x_i^2} = \sqrt{x^2 + y^2} \leq 1$
3. $||\mathrm{x}||_\infty = \max_i |x_i| = \max\{|x|, |y|\} \leq 1$

Kapitola 3. SLAR - přímé metody

Formulace:

Je dána čtvercová matice $\mathbf{A} = [a_{ij}]_{i,j=1}^{n}$ a vektor pravé strany $\mathbf{b} = [b_1, b_2, \ldots, b_n]^T$.

Hledáme vektor $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ tak, aby platilo

$$\mathbf{Ax} = \mathbf{b},$$

rozepsáno po složkách

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

Předpokládáme, že je matice $\mathbf{A}$ regulární (tj. soustava má právě jedno řešení).

Máme dva základní typy soustav:

- soustavy s obecnou maticí
- soustavy se speciální maticí (symetrická, pozitivně definitní, řídká, pásová apod.)

Pro první skupinu se většinou používají přímé metody, pro druhou skupinu metody iterační nebo speciální modifikace přímých metod.

Cramerovo pravidlo

neznámá složka řešení $$x_i = \frac{\det \mathbf{A}_i}{\det \mathbf{A}}$$

počet operací:

Je nutné vypočítat $(n+1)$ determinantů.
Pro výpočet determinantu je třeba $n!$ sčítání a v každém sčítanci je $(n-1)$ násobení.
Dostáváme:

$$(n+1)[(n-1)n! + n!] = n(n+1)!$$

např: pro $n = 30$, 10^6 operací za sekundu $\rightarrow$ výpočet trvá $7,82 \cdot 10^{21}$ let

Idea dalších přímých metod vychází z faktu, že soustavy

$$\mathbf{Ax} = \mathbf{b} \quad \text{a} \quad \mathbf{TAx} = \mathbf{Tb},$$

kde $\mathbf{T}$ je regulární matice, mají totéž řešení, tj. jsou ekvivalentní.

Touto transformací lze získat trojúhelníkovou soustavu

$$\mathbf{Ux} = \mathbf{y}: \qquad \mathbf{U} = \mathbf{TA}, \qquad \mathbf{y} = \mathbf{Tb}$$

Numerické metody
Josef Daněk
KMA/NM
13.2.2013

př:

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Trojúhelníkovou soustavu lze velmi snadno řešit zpětnou substitucí. Realizovaný proces se nazývá zpětný chod.

Gaussova eliminační metoda

$\mathbf{Ax} = \mathbf{b}$ rozepsáno po složkách $\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right]$

Definujeme multiplikátory $\boxed{m_{21} = -\frac{a_{21}}{a_{11}}}$, $\boxed{m_{31} = -\frac{a_{31}}{a_{11}}}$

$$\begin{aligned} \check{\mathrm{r}}_1^{(1)} &= \check{\mathrm{r}}_1 & b_1^{(1)} &= b_1 \\ \check{\mathrm{r}}_2^{(1)} &= \check{\mathrm{r}}_2 + m_{21}\check{\mathrm{r}}_1 & b_2^{(1)} &= b_2 + m_{21}b_1 \\ \check{\mathrm{r}}_3^{(1)} &= \check{\mathrm{r}}_3 + m_{31}\check{\mathrm{r}}_1 & b_3^{(1)} &= b_3 + m_{31}b_1 \end{aligned}$$

Získáme novou soustavu ... **1. fáze eliminace**

$$\mathbf{A}^{(1)}\mathbf{x} = \mathbf{b}^{(1)} \quad \left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & b_2^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & b_3^{(1)} \end{array}\right]$$

Definujeme multiplikátor $\boxed{m_{32} = -\frac{a_{32}^{(1)}}{a_{22}^{(1)}}}$

$$\begin{aligned} \check{\mathrm{r}}_1^{(2)} &= \check{\mathrm{r}}_1^{(1)} & b_1^{(2)} &= b_1^{(1)} \\ \check{\mathrm{r}}_2^{(2)} &= \check{\mathrm{r}}_2^{(1)} & b_2^{(2)} &= b_2^{(1)} \\ \check{\mathrm{r}}_3^{(2)} &= \check{\mathrm{r}}_3^{(1)} + m_{32}\check{\mathrm{r}}_2^{(1)} & b_3^{(2)} &= b_3^{(1)} + m_{32}b_2^{(1)} \end{aligned}$$

Získáme novou soustavu ... **2. fáze eliminace**

$$\mathbf{A}^{(2)}\mathbf{x} = \mathbf{b}^{(2)} \quad \left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & b_2^{(1)} \\ 0 & 0 & a_{33}^{(2)} & b_3^{(2)} \end{array}\right]$$

Celý tento postup nazýváme přímý chod. Trojúhelníkovou soustavu řešíme zpětným chodem.

Efektivnost algoritmu GEM

Bereme v úvahu pouze operace násobení a dělení (počet operací sčítání je přibližně stejný).
(N ... je řád matice $\mathbf{A}$)

- Celkem je $N-1$ fází eliminace. V K-té fázi počítáme $N-K$ multiplikátorů (tj. $N-K$ dělení)

$$\sum_{K=1}^{N-1}(N-K) = (N-1)N - \sum_{K=1}^{N-1}K = (N-1)N - \frac{1}{2}(N-1)N = \underline{\underline{\frac{1}{2}(N-1)N}}$$

- Každým multiplikátorem vynásobíme $(N-K+1)$ prvků rozšířené matice (jeden rozšířený řádek), tj. $(N-K)(N-K+1)$ v K-té fázi

$$\sum_{K=1}^{N-1}(N-K)(N-K+1) = \sum_{K=1}^{N-1}\left[(N^2+N) - K(2N+1) + K^2\right] =$$

$$= (N-1)(N^2+N) - \frac{1}{2}N(N-1)(2N+1) + \frac{1}{6}(N-1)N(2N-1) =$$

$$= N^3 - N^2 + N^2 - N - N^3 + \frac{1}{2}N^2 + \frac{1}{2}N + \frac{1}{3}N^3 - \frac{1}{2}N^2 + \frac{1}{6}N =$$

$$= \underline{\underline{\frac{1}{3}N^3 - \frac{1}{3}N}}$$

- Zpětný chod

$$1 + (\underbrace{1}_{\text{dělení}} + \underbrace{1}_{\text{násobení}}) + (1+2) + \cdots + (1+N-1) = \sum_{K=1}^{N} K = \underline{\underline{\frac{1}{2}N(N+1)}}$$

Celkem

$$\underbrace{\frac{1}{2}N(N-1)}_{\text{výpočet multiplikátorů}} + \underbrace{\frac{1}{3}N^3 - \frac{1}{3}N}_{\text{přímý chod}} + \underbrace{\frac{1}{2}N(N+1)}_{\text{zpětný chod}} = \boxed{\frac{1}{3}N^3 + N^2 - \frac{1}{3}N}$$

Příklad 1

Řešte soustavu rovnic

$$\begin{array}{rcrcrcl} x & + & 2y & + & 3z & = & 14 \\ 2x & + & 4y & + & 5z & = & 25 \\ 7x & + & 8y & + & 9z & = & 50 \end{array}, \qquad \text{tj.} \qquad \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 7 & 8 & 9 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 14 \\ 25 \\ 50 \end{bmatrix}$$

Řešení

Pro zápis budeme používat tvar *matice rozšířené*:

Numerické metody KMA/NM
Josef Daněk 13.2.2o13

$$\left[\begin{array}{ccc|c} \mathbf{1} & 2 & 3 & 14 \\ 2 & 4 & 5 & 25 \\ 7 & 8 & 9 & 50 \end{array}\right] \quad /\cdot(-\tfrac{2}{\mathbf{1}}) \to + \quad /\cdot(-\tfrac{7}{\mathbf{1}}) \to +$$

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 14 \\ 0 & \mathbf{0} & -1 & -3 \\ 0 & -6 & -12 & -48 \end{array}\right] \quad /\cdot(-\tfrac{6}{\mathbf{0}}) \to + \quad !!! \text{ dělíme } \mathbf{0}$$

- Algoritmus Gaussovy eliminační metody pro tento příklad není realizovatelný.
- Snadno se přesvědčíme, že má daná soustava řešení

$$x = 1,\ y = 2 \ \text{ a } \ z = 3.$$

Gaussova eliminační metoda ale selhala.

Otázka 1: Pro jaké matice $\mathbf{A}$ má soustava $\mathbf{Ax} = \mathbf{b}$ právě jedno řešení?

→ matice $\mathbf{A}$ musí být **regulární**, tj.

- všechna vlastní čísla musí být různá od nuly
- jinak řečeno řádky matice $\mathbf{A}$ musí být lineárně nezávislé
- jinak řečeno sloupce matice $\mathbf{A}$ musí být lineárně nezávislé
- jinak řečeno $\det \mathbf{A} \neq 0$

Poznámka: Vlastní číslo matice $\mathbf{A}$ je číslo λ splňující rovnici $\mathbf{Av} = \lambda\mathbf{v}$, kde $\mathbf{v}$ je vlastní vektor odpovídající vlastnímu číslu λ. Číslo λ tedy určitým způsobem charakterizuje matici $\mathbf{A}$.

Otázka 2: Pro jaké matice $\mathbf{A}$ je algoritmus Gaussovy eliminační metody realizovatelný?

→ **Věta:** Je-li matice $\mathbf{A}$ **ostře diagonálně dominantní**, pak je algoritmus GEM realizovatelný.

Poznámka: Matice $\mathbf{A} = [a_{ij}]_{i,j=1,\ldots,n}$ je ostře diagonálně dominantní, platí-li

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|,$$

tj. absolutní hodnota diagonálního prvku je větší než součet absolutních hodnot ostatních prvků v řádku.

Např.:

$$\mathbf{A} = \begin{bmatrix} \mathbf{7} & 1 & 2 \\ 1 & \mathbf{4} & 1 \\ 2 & 4 & \mathbf{9} \end{bmatrix}.$$

→ **Věta:** Je-li matice $\mathbf{A}$ **symetrická** a **pozitivně definitní**, pak je algoritmus GEM realizovatelný.

Poznámka: Matice $\mathbf{A}$ je symetrická, platí-li pro její prvky

$$a_{ij} = a_{ji} \quad \forall i, j = 1, 2, \ldots, n.$$

Poznámka: Matice $\mathbf{A}$ je pozitivně definitní,

má-li všechna vlastní čísla kladná

nebo jinak řečeno $\forall \mathbf{x} \neq \mathbf{o} : \ \mathbf{x}^T \mathbf{A} \mathbf{x} > 0$.

Poznámka: Pro soustavu s maticí, která splňuje předpoklady některé z uvedených vět, je možné dopředu říci, že půjde řešit pomocí Gaussovy eliminační metody. Obráceně to ovšem neplatí, tj. není-li např. matice soustavy ostře diagonálně dominantní, ještě to obecně neznamená, že nepůjde pomocí Gaussovy eliminační metody řešit.

Poznámka: Abychom zaručili, že soustava půjde vyřešit pro libovolnou regulární matici, musíme algoritmus Gaussovy eliminační metody upravit. Zavedeme tzv. **výběr hlavního prvku (pivotaci)**.

Poznámka: **Pivot (hlavní prvek)** ... první nenulový prvek v daném řádku matice.

Příklad 2

Pomocí **GEM se sloupcovou pivotací** vyřešte soustavu rovnic z Příkladu 1, kde selhala klasická GEM, tj. řešíme soustavu $\mathbf{A}\mathbf{x} = \mathbf{b}$, kde

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 7 & 8 & 9 \end{bmatrix} \quad \text{a} \quad \mathbf{b} = \begin{bmatrix} 14 \\ 25 \\ 50 \end{bmatrix}.$$

Řešení

1. sloupec ↓

vyměň →
$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 14 \\ 2 & 4 & 5 & 25 \\ \mathbf{7} & 8 & 9 & 50 \end{array}\right) \rightsquigarrow \left(\begin{array}{ccc|c} 7 & 8 & 9 & 50 \\ 2 & 4 & 5 & 25 \\ 1 & 2 & 3 & 14 \end{array}\right) \quad /\cdot(-\tfrac{2}{7}) \leftarrow + \quad /\cdot(-\tfrac{1}{7}) \leftarrow +$$

2. sloupec ↓

není třeba měnit →
$$\left(\begin{array}{ccc|c} 7 & 8 & 9 & 50 \\ 0 & \frac{12}{7} & \frac{17}{7} & \frac{75}{7} \\ 0 & \frac{6}{7} & \frac{12}{7} & \frac{48}{7} \end{array}\right) \quad /\cdot(-\tfrac{\frac{6}{7}}{\frac{12}{7}}) = -\tfrac{1}{2} \leftarrow +$$

Numerické metody KMA/NM
Josef Daněk 13.2.2o13

$$\left(\begin{array}{ccc|c} 7 & 8 & 9 & 50 \\ 0 & \frac{12}{7} & \frac{17}{7} & \frac{75}{7} \\ 0 & 0 & \frac{1}{2} & \frac{3}{2} \end{array}\right) \quad \Longrightarrow \quad \mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Poznámky:

- Při sloupcové pivotaci jsme postupně v každém sloupci (resp. jeho části pod diagonálou včetně) vybírali číslo, které bylo maximální v absolutní hodnotě a v případě, že toto číslo neleželo na diagonále, vyměnili jsme příslušné 2 rovnice. Dále jsme pokračovali jako v GEM bez pivotace, tj. nulovali jsme koeficienty pod diagonálou.
- Sloupcová pivotace není jediná možnost. Podobně můžeme vybírat i maximální prvek v absolutní hodnotě z příslušného řádku (resp. jeho části) a poté vyměnit příslušné sloupce. Pozor! Je ovšem třeba zaměnit i příslušné složky řešení $\mathbf{x}$. V tomto případě hovoříme o **řádkové pivotaci**.
- Další možností je vybírat maximální prvek v absolutní hodnotě z celé matice $\mathbf{A}$ (resp. příslušné podmatice). V tomto případě hovoříme o **úplné pivotaci**. Opět je třeba mít na paměti, že je třeba zaměnit složky ve vektoru řešení. Nevýhodou úplné pivotace je pomalejší výpočet neboť hlavní prvek vyhledáváme z celé dosud neupravené části.
- Libovolnou pivotací dosáhneme realizovatelnosti GEM pro libovolnou regulární matici $\mathbf{A}$.

Metoda LU-rozkladu

Opět uvažujeme regulární matici $\mathbf{A}$ řádu N. Matici $\mathbf{A}$ lze rozložit na součin $\boxed{\mathbf{A} = \mathbf{L}\mathbf{U}}$, kde $\mathbf{L}$ je dolní trojúhelníková matice řádu N a $\mathbf{U}$ je horní trojúhelníková matice řádu N.

Např:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

Tento rozklad není dán jednoznačně (12 neznámých a 9 podmínek), jednoznačnosti dosáhneme např. tím, že položíme $\boxed{l_{ii} = 1, \ i = 1, 2, \ldots, N}$.

Algoritmus: (viz skripta)

je odvozen z postupného násobení řádků matice $\mathbf{L}$ a sloupců matice $\mathbf{U}$

$(1,1)\ u_{11} = a_{11}$	$(2,1)\ a_{21} = l_{21}u_{11}$	$(3,1)\ a_{31} = l_{31}u_{11}$
$(1,2)\ u_{12} = a_{12}$	$(2,2)\ a_{22} = l_{21}u_{12} + u_{22}$	$(3,2)\ a_{32} = l_{31}u_{12} + l_{32}u_{22}$
$(1,3)\ u_{13} = a_{13}$	$(2,3)\ a_{23} = l_{21}u_{13} + u_{23}$	$(3,3)\ a_{33} = l_{31}u_{13} + l_{32}u_{23} + u_{33}$

Řešení soustavy $\mathbf{A}\mathbf{x} = \mathbf{b}$ metodou LU-rozkladu:

1. Realizace LU-rozkladu: $\mathbf{A} = \mathbf{L}\mathbf{U}$
2. Řešení trojúhelníkové soustavy: $\mathbf{L}\mathbf{y} = \mathbf{b}$
3. Řešení trojúhelníkové soustavy: $\mathbf{U}\mathbf{x} = \mathbf{y}$

$$\boxed{\mathbf{L}\underbrace{\mathbf{U}\mathbf{x}}_{\mathbf{y}} = \mathbf{b}}$$

Souvislost GEM a metody LU-rozkladu

Gaussovu eliminaci lze popsat pomocí násobení regulárními maticemi.

První fázi popíšeme takto ... $\boxed{\mathbf{A}^{(1)} = \mathbf{M}_1\mathbf{A}}$, kde $\mathbf{M}_1 = \begin{bmatrix} 1 & & & & \\ m_{21} & 1 & & & \\ m_{31} & & 1 & & \\ m_{41} & & & 1 & \\ \vdots & & & & \ddots \\ m_{n1} & & & & 1 \end{bmatrix}$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ m_{21}a_{11}+a_{21} & m_{21}a_{12}+a_{22} & m_{21}a_{13}+a_{23} & \dots & m_{21}a_{1n}+a_{2n} \\ m_{31}a_{11}+a_{31} & m_{31}a_{12}+a_{32} & m_{31}a_{13}+a_{33} & \dots & m_{31}a_{1n}+a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ m_{n1}a_{11}+a_{n1} & m_{n1}a_{12}+a_{n2} & m_{n1}a_{13}+a_{n3} & \dots & m_{n1}a_{1n}+a_{nn} \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & & & & \\ m_{21} & 1 & & & \\ m_{31} & & 1 & & \\ m_{41} & & & 1 & \\ \vdots & & & & \ddots \\ m_{n1} & & & & 1 \end{bmatrix} \cdot \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}$$

Druhou fázi popíšeme takto ... $\boxed{\mathbf{A}^{(2)} = \mathbf{M}_2\mathbf{A}^{(1)}}$, kde $\mathbf{M}_2 = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & m_{32} & 1 & & \\ & m_{42} & & 1 & \\ & \vdots & & & \ddots \\ & m_{n2} & & & 1 \end{bmatrix}$

$\vdots$

Nakonec $(n-1)$ fázi popíšeme ... $\boxed{\mathbf{A}^{(n-1)} = \mathbf{M}_{n-1}\mathbf{A}^{(n-2)}}$, kde $\mathbf{M}_{n-1} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & 1 & \\ & & & m_{n,n-1} & 1 \end{bmatrix}$

Dostali jsme horní trojúhelníkovou matici, označíme ji např. $\mathbf{V}$

$$\boxed{\mathbf{V} = \mathbf{A}^{(n-1)} = \underbrace{\mathbf{M}_{n-1}\mathbf{M}_{n-2}\mathbf{M}_{n-3}\dots\mathbf{M}_2\mathbf{M}_1}_{\text{ozn.}\mathbf{M}}\mathbf{A}} \quad \Rightarrow \quad \boxed{\mathbf{A} = \mathbf{M}^{-1}\mathbf{V}}$$

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

$$\boxed{\mathbf{M}^{-1} = \mathbf{M}_1^{-1}\mathbf{M}_2^{-1}\dots\mathbf{M}_{n-1}^{-1}}$$

Jak vypadá např. $\mathbf{M}_2^{-1}$?

$$\mathbf{M}_2 = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & m_{32} & 1 & & \\ & m_{42} & & 1 & \\ & \vdots & & & \ddots \\ & m_{n2} & & & 1 \end{bmatrix} \quad \rightarrow \quad \mathbf{M}_2^{-1} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & -m_{32} & 1 & & \\ & -m_{42} & & 1 & \\ & \vdots & & & \ddots \\ & -m_{n2} & & & 1 \end{bmatrix}$$

protože po vynásobení:

- i-tý řádek $\times$ j-tý sloupec $(j \neq i)$
  - buď $m_{42}\cdot 1 + 1\cdot(-m_{42}) = 0$
  - nebo $m_{42}\cdot 0 + 0\cdot 1 + 1\cdot 0 = 0$
- i-tý řádek $\times$ i-tý sloupec
  - $m_{42}\cdot 0 + 1\cdot 1 = 1$

tj. $\mathbf{M}_2\cdot\mathbf{M}_2^{-1} = \mathbf{I}$

Jak vypadá $\mathbf{M}^{-1}$?

$$\mathbf{M}^{-1} = \begin{bmatrix} 1 & & & & \\ -m_{21} & 1 & & & \\ -m_{31} & -m_{32} & 1 & & \\ \vdots & & & \ddots & \\ -m_{n1} & -m_{n2} & -m_{n3} & \dots & 1 \end{bmatrix}$$

Př.:

$$\begin{bmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & -m_{32} & 1 \end{bmatrix}$$

Platí

$$\boxed{\mathbf{A} = \underbrace{\mathbf{M}^{-1}}_{(*)}\cdot\underbrace{\mathbf{V}}_{(**)}}$$

(*) dolní trojúhelníková s 1 na diagonále

(**) horní trojúhelníková

$\Rightarrow$ jedná se o LU-rozklad (rozklad je jednoznačný) $\boxed{\mathbf{L} = \mathbf{M}^{-1} \text{ a } \mathbf{U} = \mathbf{V}}$

Výpočet determinantů

1. Užití GEM

$$\mathbf{U} = \mathbf{M}_{N-1}\mathbf{M}_{N-2}\dots\mathbf{M}_2\mathbf{M}_1\mathbf{A}$$

$$\det \mathbf{U} = \underbrace{\det \mathbf{M}_{N-1}}_{=1}\underbrace{\det \mathbf{M}_{N-2}}_{=1}\dots\underbrace{\det \mathbf{M}_2}_{=1}\underbrace{\det \mathbf{M}_1}_{=1}\det \mathbf{A}$$

$$\det \mathbf{A} = \det \mathbf{U} = \prod_{i=1}^{N} u_{ii}$$

2. Užití LU-rozkladu

$$\det \mathbf{A} = \det \mathbf{L}\ \det \mathbf{U} = \prod_{i=1}^{N} u_{ii}$$

Výpočet inverzní matice

1. Užití GEM

$$\mathbf{A}\,\mathbf{X} = \mathbf{I} \qquad \text{(maticová soustava)}$$
$$\mathbf{X} = \mathbf{A}^{-1}$$

2. Užití LU-rozkladu

$$\mathbf{A} = \mathbf{L}\,\mathbf{U}$$
$$\mathbf{A}^{-1} = \mathbf{U}^{-1}\,\mathbf{L}^{-1}$$

Numerické aspekty GEM a metody LU-rozkladu

Při numerické realizaci nevypočteme přesně matice $\mathbf{L}$ a $\mathbf{U}$, ale přibližné matice $\widetilde{\mathbf{L}}$ a $\widetilde{\mathbf{U}}$.

Teoreticky platí $\mathbf{A} = \mathbf{L}\mathbf{U}$.

Označíme $\widetilde{\mathbf{A}} = \widetilde{\mathbf{L}}\widetilde{\mathbf{U}}$ ... dopočteno pro získané matice $\widetilde{\mathbf{L}}$ a $\widetilde{\mathbf{U}}$.

Budeme zkoumat rozdíl $\widetilde{\mathbf{A}} - \mathbf{A}$.

Označme $\mathbf{E}$ a $\mathbf{F}$ matice chyb takové, že platí:

$$\widetilde{\mathbf{L}} = \mathbf{L} + \mathbf{E}, \quad \widetilde{\mathbf{U}} = \mathbf{U} + \mathbf{F}$$

Potom:

$$\widetilde{\mathbf{A}} - \mathbf{A} = \widetilde{\mathbf{L}}\widetilde{\mathbf{U}} - \mathbf{L}\mathbf{U} = (\mathbf{L}+\mathbf{E})(\mathbf{U}+\mathbf{F}) - \mathbf{L}\mathbf{U} = \mathbf{E}\mathbf{U} + \mathbf{L}\mathbf{F} + \mathbf{E}\mathbf{F}$$

Numerické metody KMA/NM
Josef Daněk 13.2.2o13

Odtud plyne závěr: Pokud jsou multiplikátory v absolutní hodnotě velké, pak prvky $\mathbf{L}$ jsou v absolutní hodnotě velké $\Rightarrow$ chyba může být velká. Toto je jeden z důvodů realizace pivotace.

Přímé metody pro soustavy se speciální maticí

Uvažujeme matice:

- symetrická
- symetrická a pozitivně definitní
- diagonálně dominantní
- pásová

Platí: Je-li matice $\mathbf{A}$ symetrická a $\mathbf{A}^{(k)}$ jsou matice získané GEM v základní verzi, pak podmatice $\mathbf{A}^{(k)}$ jsou také symetrické.

př:

$$\begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix} \quad / \cdot(-\tfrac{b}{a}) \leftarrow + \quad / \cdot(-\tfrac{c}{a}) \leftarrow +$$

$$\begin{bmatrix} a & b & c \\ 0 & d-\frac{b^2}{a} & e-\frac{bc}{a} \\ 0 & e-\frac{bc}{a} & f-\frac{c^2}{a} \end{bmatrix}$$

Lze pak použít **symetrickou verzi GEM a LU-rozkladu**.

Je-li matice $\mathbf{A}$ navíc pozitivně definitní, pak lze realizovat **Choleského metodu** rozkladu

$$\boxed{\mathbf{A} = \mathbf{U}^T\mathbf{U}}$$

Poznámka: V algoritmu je potřeba realizovat výpočet odmocnin a to lze pouze pro pozitivně definitní matice.

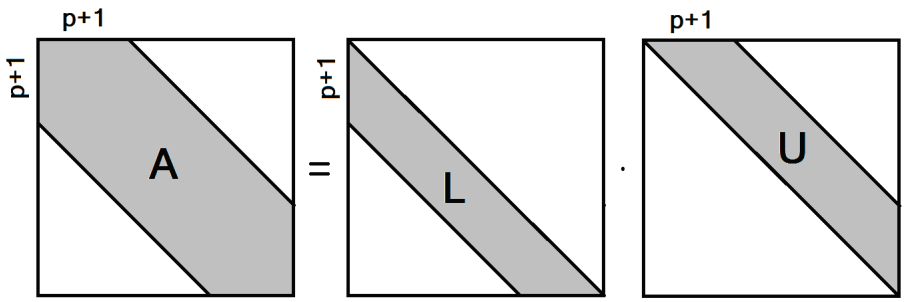
př:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} u_{11} & 0 & 0 \\ u_{21} & u_{22} & 0 \\ u_{31} & u_{32} & u_{33} \end{bmatrix} \cdot \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

$(1,1):\ a_{11} = u_{11}^2$
$(1,2):\ a_{12} = u_{11}\cdot u_{12}$
$(1,3):\ a_{13} = u_{11}\cdot u_{13}$
$(2,1):\ a_{21} = u_{12}\cdot u_{11}$
$(2,2):\ a_{22} = u_{12}^2 + u_{22}^2$
$(2,3):\ a_{23} = u_{12}\cdot u_{13} + u_{22}\cdot u_{23}$
$(3,1):\ a_{31} = u_{13}\cdot u_{11}$
$(3,2):\ a_{32} = u_{13}\cdot u_{12} + u_{23}\cdot u_{22}$
$(3,3):\ a_{33} = u_{13}^2 + u_{23}^2 + u_{33}^2$

Metoda LU-rozkladu pro pásové matice

Uvažujeme matici $\mathbf{A}$ takovou, že $\boxed{a_{ij} = 0}$, když $\boxed{|i-j| > p}$ (šířka pásu je $2p+1$).



Pokud lze realizovat LU-rozklad, pak

$$l_{ij} = 0, \text{ když } \; j > i \;\text{ a }\; j < i - p,$$
$$u_{ij} = 0, \text{ když } \; j < i \;\text{ a }\; j > i + p.$$

Poznámka: V případě obecné matice však nelze čekat, že matice $\mathbf{L}$ a $\mathbf{U}$ bude mít nulový prvek v téže pozici jako jej měla matice $\mathbf{A}$.

Pro pásové, symetrické, pozitivně definitní matice používáme **speciální verzi Choleského rozkladu**.

Metoda faktorizace pro třídiagonální matici

Uvažujeme soustavu $n+1$ lineárních algebraických rovnic $\boxed{\mathbf{A}\cdot\mathbf{Y} = \mathbf{F}}$ ve tvaru:

$$\left(\begin{array}{cccccc|c}
c_0 & -b_0 & & & & & f_0 \\
-a_1 & c_1 & -b_1 & & & & f_1 \\
 & -a_2 & c_2 & -b_2 & & & f_2 \\
 & & -a_3 & c_3 & -b_3 & & f_3 \\
 & & & \ddots & \ddots & \ddots & \vdots \\
 & & & & -a_{n-1} & c_{n+1} \quad -b_{n-1} & f_{n-1} \\
 & & & & & -a_n \quad c_n & f_n
\end{array}\right)$$

Označíme

$$\boxed{\alpha_1 = \frac{b_0}{c_0}}, \quad \boxed{\beta_1 = \frac{f_0}{c_0}}$$

První dvě rovnice (první rovnice je vydělená c_0) lze psát ve tvaru:

$$\begin{array}{rcrcrcl}
y_0 & - & \alpha_1 y_1 & & & = & \beta_1 \quad / \cdot a_1 \\
-a_1 y_0 & + & c_1 y_1 & - & b_1 y_2 & = & f_1
\end{array} \quad \hookleftarrow +$$

$$(c_1 - a_1\alpha_1)\, y_1 - b_1\, y_2 = f_1 + a_1\, \beta_1$$

Přepíšeme (rovnice vydělená koeficientem u y_1) na tvar:

$$y_1 - \alpha_2 y_2 = \beta_2$$

kde

$$\boxed{\alpha_2 = \frac{b_1}{c_1 - a_1\alpha_1}}, \quad \boxed{\beta_2 = \frac{f_1 + a_1\beta_1}{c_1 - a_1\alpha_1}}$$

Po zobecnění:

- PŘÍMÝ CHOD

$$\boxed{\alpha_1 = \frac{b_0}{c_0}} \qquad \boxed{\beta_1 = \frac{f_0}{c_0}}$$

$$\boxed{\alpha_{i+1} = \frac{b_i}{c_i - a_i \cdot \alpha_i}} \;\; i = 1, 2, \ldots, n-1 \qquad \boxed{\beta_{i+1} = \frac{f_i + a_i \cdot \beta_i}{c_i - a_i \cdot \alpha_i}} \;\; i = 1, 2, \ldots, n$$

- ZPĚTNÝ CHOD

Pro poslední dvě rovnice získáme:

$$\begin{array}{rcrcrcl}
y_{n-1} & - & \alpha_n y_n & & & = & \beta_n \quad / \cdot a_n \\
-a_n y_{n-1} & + & c_n y_n & & \clubsuit & = & f_n
\end{array} \quad \hookleftarrow +$$

$$(c_n - a_n\alpha_n)\, y_n = f_n + a_n\, \beta_n$$

♣ ve druhé rovnici již není člen $-b_n y_{n+1}$

Vyjádříme poslední složku řešení:

$$\boxed{y_n = \frac{f_n + a_n\beta_n}{c_n - a_n\alpha_n} =: \beta_{n+1}}$$

Zpětně dosazujeme:

$$\boxed{y_{i-1} = \beta_i + \alpha_i \cdot y_i} \;\; i = n, n-1, \ldots, 1 \qquad (*)$$

Efektivnost algoritmu

dělení		$2n+1$	$(1+n-1+1+n)$
násobení	$\cdot$	$3n$	$(n+n+n)$
sčítání a odčítání	$\pm$	$3n$	$(n+n+n)$
celkem		$(8n+1)$ operací	

Poznámka: Pokud budeme řešit tuto soustavu pro různé pravé strany $\mathbf{F}$, nemusíme již znovu vyjadřovat koeficienty α_i, protože nezávisí na $\mathbf{F}$. Stačí tedy přepočítat β_i.

Zatím jsme neuvedli předpoklady pro metodu faktorizace

- je třeba zajistit, aby jmenovatel $c_i - a_i\alpha_i$ byl nenulový pro $i = 1, 2, \ldots, n$
- y_i se určuje z rekurentní formule $(*)$, přitom může dojít k akumulaci zaokrouhlovacích chyb.

Nechť α_i, β_i jsou dokonce přesně vypočítané a nechť máme $\tilde{y}_n = y_n + \varepsilon_n$ (s chybou ε_n).

Potom postupně vypočítáme podle $(*)$

$$\boxed{\tilde{y}_{i-1} = \beta_i + \alpha_i \tilde{y}_i}, \quad i = n, n-1, \ldots, 1$$

Označíme-li $\boxed{\varepsilon_i = \tilde{y}_i - y_i}$ chybu, bude jistě splňovat homogenní rovnici

$$\boxed{\varepsilon_{i-1} = \alpha_i \varepsilon_i}, \quad i = n, n-1, \ldots, 1$$

(protože přesné hodnoty y_i splňují $y_{i-1} = \beta_i + \alpha_i y_i$)

$\Rightarrow$ Pokud by byly koeficienty $|\alpha_i| > 1$, dojde k velkému nárůstu chyby ε_0 !!!

Pro $\underbrace{\boxed{|\alpha_i| \leq 1, i = 1, 2, \ldots, n}}_{(\bullet)}$ je algoritmus stabilní.

Postačující podmínky pro zajištění $(\bullet)$: Matice soustavy je ostře diagonálně dominantní.

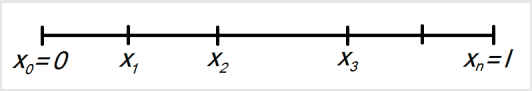
Důkaz viz literatura.

Příklady aplikací, které vedou na soustavu s třídiagonální maticí

1. Řešení okrajové úlohy

$$\boxed{\begin{array}{rcll} (k(x)u'(x))' - q(x)u(x) & = & -f(x) & x \in (0,l) \\ u(0) & = & 0 & \\ u(l) & = & 0 & \\ k(x) & > & 0 & \\ q(x) & \geq & 0 & \end{array}}$$

x na $\langle 0, l \rangle$ diskretizujeme ... síť x_i



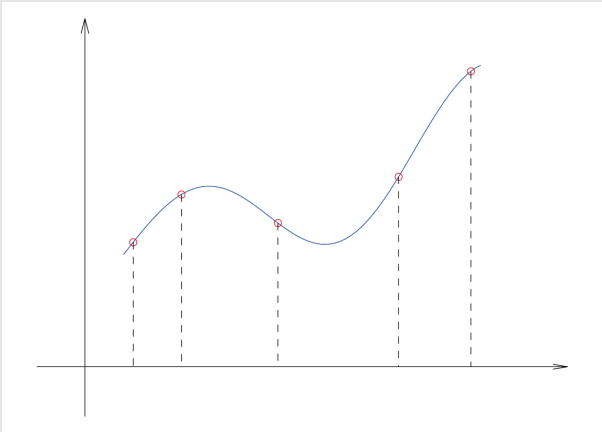
původní úlohu nahradíme úlohou s diferenční rovnicí a použijeme vzorec pro poměrnou diferenci

2. Diferenční schémata pro rovnici vedení tepla

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

$$\boxed{\begin{array}{rcll} \dfrac{\partial u}{\partial t} & = & \dfrac{\partial^2 u}{\partial x^2} & x \in (0,l), \quad t > 0 \\ u(0,t) & = & \mu_1(t) & \\ u(l,t) & = & \mu_2(t) & \\ u(x,0) & = & u_0(x) & \end{array}}$$

3. Soustava pro výpočet koeficientů kubického spline (aproximace funkce) ... budeme probírat



Podmíněnost úlohy řešit SLAR

Uvažujeme opět soustavu $\mathbf{Ax} = \mathbf{b}$, $\mathbf{A}$ ... $n \times n$ regulární, $\mathbf{b} \in \mathbb{R}^n$.

Označení:

$\Delta\mathbf{A}$ ... malá změna matice $\mathbf{A}$

$\Delta\mathbf{b}$ ... malá změna vektoru $\mathbf{b}$

$\Delta\mathbf{x}$ ... odpovídající změna vektoru neznámých

$\mathbf{x}^*$ ... přesné řešení soustavy $\mathbf{Ax} = \mathbf{b}$

Platí:

$$\boxed{(\mathbf{A} + \Delta\mathbf{A})(\mathbf{x}^* + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}}$$

1. Uvažujme situaci $\Delta\mathbf{A} = \mathbf{0}$, tj. $\mathbf{A}$ je zadána přesně

Otázka: Jakou změnu řešení vyvolá změna pravé strany?

$$\mathbf{Ax}^* + \mathbf{A}\Delta\mathbf{x} = \mathbf{b} + \Delta\mathbf{b}$$
$$\mathbf{A}\Delta\mathbf{x} = \Delta\mathbf{b}$$

$$\Delta\mathbf{x} = \mathbf{A}^{-1}\Delta\mathbf{b}$$

Z vlastností maticové normy plyne:

$$\mathbf{Ax}^* = \mathbf{b} \;\Rightarrow\; ||\mathbf{b}|| \le ||\mathbf{A}|| \cdot ||\mathbf{x}^*|| \;\Rightarrow\; \frac{1}{||\mathbf{x}^*||} \le \frac{||\mathbf{A}||}{||\mathbf{b}||}$$

$$\Delta\mathbf{x} = \mathbf{A}^{-1}\Delta\mathbf{b} \;\Rightarrow\; ||\Delta\mathbf{x}|| \le ||\mathbf{A}^{-1}|| \cdot ||\Delta\mathbf{b}||$$

$$\frac{||\Delta\mathbf{x}||}{||\mathbf{x}^*||} \le \frac{||\mathbf{A}^{-1}|| \cdot ||\Delta\mathbf{b}|| \cdot ||\mathbf{A}||}{||\mathbf{b}||}$$

$$C_p = \frac{\frac{||\Delta\mathbf{x}||}{||\mathbf{x}^*||}}{\frac{||\Delta\mathbf{b}||}{||\mathbf{b}||}} \le ||\mathbf{A}^{-1}|| \cdot ||\mathbf{A}||$$

2. Případ, kdy $\underline{\Delta\mathbf{b} = \mathbf{0}}$, tj. $\mathbf{b}$ je zadána přesně

$$(\mathbf{A} + \Delta\mathbf{A})(\mathbf{x}^* + \Delta\mathbf{x}) = \mathbf{b}$$

$$\underline{\mathbf{Ax}^*} + \Delta\mathbf{A}\mathbf{x}^* + \mathbf{A}\Delta\mathbf{x} + \Delta\mathbf{A}\Delta\mathbf{x} = \underline{\mathbf{b}}$$

$$\mathbf{A}\Delta\mathbf{x} = -\Delta\mathbf{A}(\mathbf{x}^* + \Delta\mathbf{x})$$

$$\Delta\mathbf{x} = -\mathbf{A}^{-1}\Delta\mathbf{A}(\mathbf{x}^* + \Delta\mathbf{x})$$

$$||\Delta\mathbf{x}|| \le ||\mathbf{A}^{-1}|| \cdot ||\Delta\mathbf{A}|| \cdot ||\mathbf{x}^* + \Delta\mathbf{x}|| \cdot \frac{||\mathbf{A}||}{||\mathbf{A}||}$$

$$\frac{||\Delta\mathbf{x}||}{||\mathbf{x}^* + \Delta\mathbf{x}||} \le \underbrace{||\mathbf{A}^{-1}|| \cdot ||\mathbf{A}||}_{\ge C_p} \cdot \frac{||\Delta\mathbf{A}||}{||\mathbf{A}||}$$

$$C_p = \frac{\frac{||\Delta\mathbf{x}||}{||\mathbf{x}^* + \Delta\mathbf{x}||}}{\frac{||\Delta\mathbf{A}||}{||\mathbf{A}||}} \le ||\mathbf{A}^{-1}|| \cdot ||\mathbf{A}||$$

3. Rozmyslete obecný případ $\Delta\mathbf{b} \neq \mathbf{0}$, $\Delta\mathbf{A} \neq \mathbf{0}$ viz skripta (D.cv.)

Poznámka: Pro symetrické matice je číslo podmíněnosti podíl největší a nejmenší absolutní hodnoty vlastního čísla.

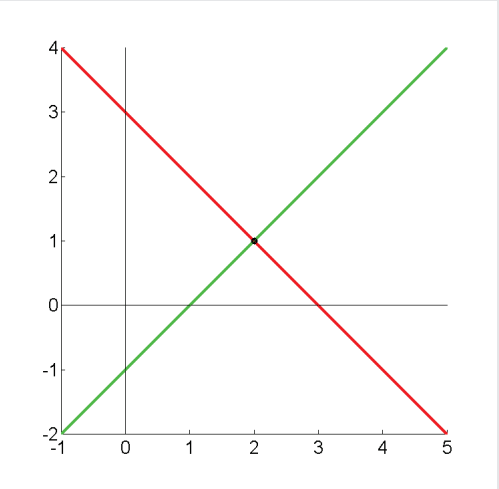
$$C_p = \underbrace{||\mathbf{A}^{-1}||}_{\frac{1}{\lambda_{min}}} \cdot \underbrace{||\mathbf{A}||}_{\lambda_{max}} = \frac{|\lambda_{max}|}{|\lambda_{min}|}$$

(P[illegible])

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Geometrická interpretace - dobře podmíněná úloha (2D)

$$\left.\begin{array}{l} x + y = 3 \\ x - y = 1 \end{array}\right\} \Rightarrow \left.\begin{array}{l} y = 3 - x \\ y = x - 1 \end{array}\right\} \text{ řešení } \begin{array}{l} x^* = 2 \\ y^* = 1 \end{array}$$



$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

Dobře podmíněná úloha - malá relativní změna vstupních dat vyvolá malou relativní změnu výstupních dat.

$$C_p = ||A^{-1}|| \cdot ||A||$$

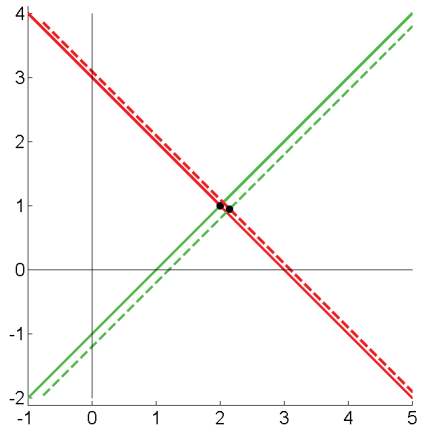
$$\mathbf{A}^{-1} = \begin{bmatrix} 0,5 & 0,5 \\ 0,5 & -0,5 \end{bmatrix} = 0,5\,\mathbf{A}$$

$C_p = 1 \cdot 2 = \mathbf{2}$ (řádková, sloupcová norma); $C_p = \frac{1}{\sqrt{2}}\sqrt{2} = \mathbf{1}$ (spektrální norma)

$$\left(\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad ||\mathbf{A}||_{SP} = \sqrt{2}, \quad ||\mathbf{A}^{-1}||_{SP} = \tfrac{1}{\sqrt{2}}\right)$$

1. změna pravé strany (změna matice $\Delta\mathbf{A} = \mathbf{0}$)

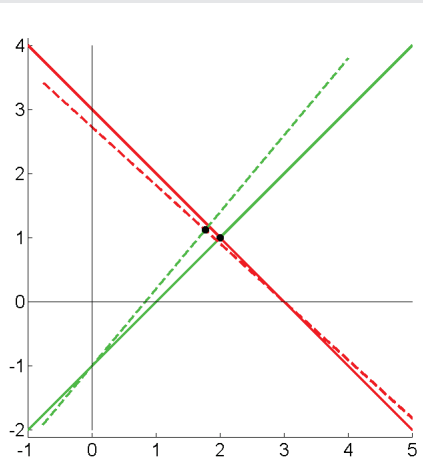
$$\Delta\mathbf{b} = \begin{bmatrix} 0,1 \\ 0,2 \end{bmatrix}, \quad \mathbf{b} + \Delta\mathbf{b} = \begin{bmatrix} 3,1 \\ 1,2 \end{bmatrix}$$



změněná soustava $y = 3,1 - x$; $y = x - 1,2$

2. změna matice soustavy (změna pravé strany $\Delta\mathbf{b} = \mathbf{0}$)

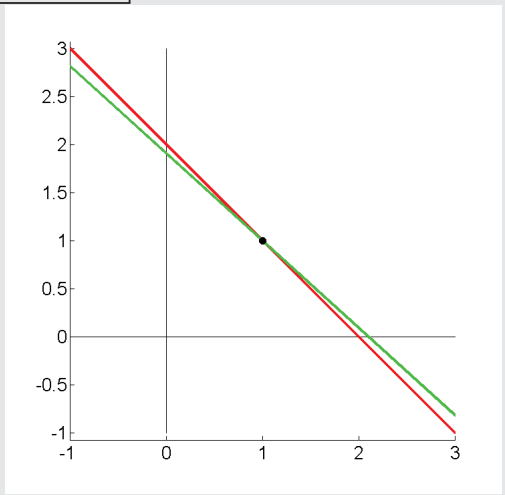
$$\Delta\mathbf{A} = \begin{bmatrix} 0 & 0,1 \\ 0,2 & 0 \end{bmatrix}, \quad \mathbf{A} + \Delta\mathbf{A} = \begin{bmatrix} 1 & 1,1 \\ 1,2 & -1 \end{bmatrix}$$



změněná soustava $y = \frac{1}{1,1}(3 - x)$; $y = 1,2x - 1$

Geometrická interpretace - špatně podmíněná úloha (2D)

$$\left.\begin{array}{l} x + y = 2 \\ x + 1,1\,y = 2,1 \end{array}\right] \Rightarrow \left.\begin{array}{l} y = 2 - x \\ y = \frac{1}{1,1}(2,1 - x) \end{array}\right\} \text{ řešení } \begin{array}{l} x^* = 1 \\ y^* = 1 \end{array}$$



$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 1,1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 2 \\ 2,1 \end{bmatrix}$$

Špatně podmíněná úloha - malá relativní změna vstupních dat vyvolá velkou relativní změnu výstupních dat.

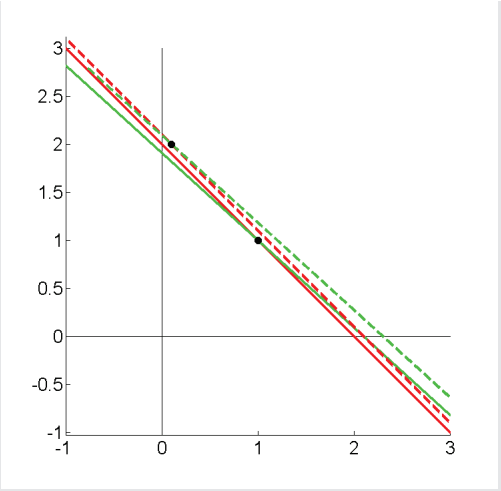
$$C_p = ||A^{-1}|| \cdot ||A||$$

$$\mathbf{A}^{-1} = \begin{bmatrix} 11 & -10 \\ -10 & 10 \end{bmatrix}$$

$C_p = 2,1 \cdot 21 = \mathbf{44,1}$ (řádková, sloupcová norma); $C_p = 2,0512 \cdot 20,5125 = \mathbf{42,07}$ (spektrální norma)

1. změna pravé strany (změna matice $\Delta\mathbf{A} = \mathbf{0}$)

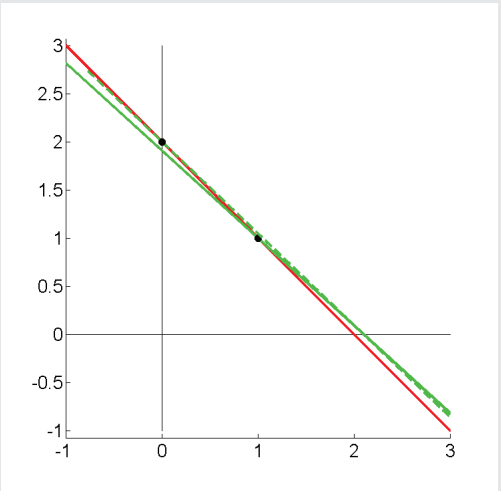
$$\Delta\mathbf{b} = \begin{bmatrix} 0,1 \\ 0,2 \end{bmatrix}, \quad \mathbf{b} + \Delta\mathbf{b} = \begin{bmatrix} 2,1 \\ 2,3 \end{bmatrix}$$



změněná soustava $y = 2,1 - x$; $y = \frac{1}{1,1}(2,3 - x)$

2. změna matice soustavy (změna pravé strany $\Delta\mathbf{b} = \mathbf{0}$)

$$\Delta\mathbf{A} = \begin{bmatrix} 0 & 0 \\ 0 & -0,05 \end{bmatrix}, \quad \mathbf{A} + \Delta\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 1,05 \end{bmatrix}$$



změněná soustava $y = 2 - x$; $y = \frac{1}{1,05}(2,1 - x)$

Poznámka: Jiný výpočet čísla podmíněnosti (prakticky)

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

změna na vstupu ... $\Delta\mathbf{A}$

vyvolá změnu na výstupu ... $\Delta\mathbf{x}$

$$(\mathbf{A} + \Delta\mathbf{A})(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b}$$

$$C_p = \frac{\frac{||\Delta\mathbf{x}||}{||\mathbf{x}||}}{\frac{||\Delta\mathbf{A}||}{||\mathbf{A}||}}$$

Příklad

$$\mathbf{A} = \begin{bmatrix} 50 & -100 \\ 50 & -101 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

změna matice soustavy

$$\Delta\mathbf{A} = \begin{bmatrix} 0 & 0,1 \\ 0,2 & 0 \end{bmatrix} \quad \Rightarrow \quad \widetilde{\mathbf{A}} = \mathbf{A} + \Delta\mathbf{A} = \begin{bmatrix} 50 & -99,9 \\ 50,2 & -101 \end{bmatrix}$$

$$\tilde{\mathbf{x}} = \widetilde{\mathbf{A}}^{-1}\mathbf{b} = \begin{bmatrix} 2,8527 \\ 1,4278 \end{bmatrix}$$

$$\Delta\mathbf{x} = \mathbf{x} - \tilde{\mathbf{x}} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} - \begin{bmatrix} 2,8527 \\ 1,4278 \end{bmatrix} = \begin{bmatrix} -0,8527 \\ -0,4278 \end{bmatrix}$$

$\|.\|$	1.vektorová / sloupcová	maximová / řádková	euklidovská / spektrální
$\Delta\mathbf{x} = \begin{bmatrix} -0,8527 \\ -0,4278 \end{bmatrix}$	1,2805	0,8527	0,9539
$\mathbf{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$	3	2	2,2361
$\Delta\mathbf{A} = \begin{bmatrix} 0 & 0,1 \\ 0,2 & 0 \end{bmatrix}$	0,2	0,2	0,2
$\mathbf{A} = \begin{bmatrix} 50 & -100 \\ 50 & -101 \end{bmatrix}$	201	151	158,7479
C_p	428,97	321,89	338,6

$$||\mathbf{x}||_1 = \sum_i |x_i|, \quad ||\mathbf{x}||_\infty = \max_i |x_i|, \quad ||\mathbf{x}||_2 = \sqrt{\sum_i x_i^2}$$

$$||\mathbf{A}||_S = \max_k \sum_i |a_{ik}|, \quad ||\mathbf{A}||_R = \max_i \sum_k |a_{ik}|, \quad ||\mathbf{A}||_{SP} = \max_k \lambda_k^{\frac{1}{2}}(A^H A)$$

(Při použití různých norem dostáváme obecně různá čísla podmíněnosti.)

$$C_p = ||\mathbf{A}^{-1}|| \cdot ||\mathbf{A}||$$

$$\mathbf{A}^{-1} = \begin{bmatrix} 2,02 & -2 \\ 1 & -1 \end{bmatrix}$$

$C_p = 151 \cdot 4,02 = 607,02$ (řádková $||.||$);

$C_p = 201 \cdot 3,02 = 607,02$ (sloupcová $||.||$);

$C_p = 158,7479 \cdot 3,1750 \doteq 504,03$ (spektrální $||.||$)

Poznámky k podmíněnosti

- Stále předpokládáme, že $\mathbf{A}$ je regulární matice.
  Soustava $\boxed{\mathbf{Ax} = \mathbf{b}}$ má potom právě jedno řešení.
  Předpokládejme, že je matice $\mathbf{A}$ normalizována tak, že její max. prvek v absolutní hodnotě je roven 1.
  Je-li soustava špatně podmíněná, potom matice $\mathbf{A}^{-1}$ musí obsahovat velké prvky.

  např:
  $$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0,99999 \end{bmatrix} \quad \Rightarrow \quad \mathbf{A}^{-1} = \begin{bmatrix} -99999 & 100000 \\ 100000 & -100000 \end{bmatrix}$$

  To odpovídá faktu, že číslo podmíněnosti je velké $\boxed{C_p = ||\mathbf{A}|| \cdot ||\mathbf{A}^{-1}||}$ (norma $||\mathbf{A}^{-1}||$ je velká).

- Rozbor chyb

  $\boxed{\mathbf{Ax} = \mathbf{b}}$, $\mathbf{x}^*$ ... přesné, $\mathbf{x}_c$ ... vypočtené

  $\Rightarrow$ chybu můžeme měřit pomocí rezidua $\boxed{\mathbf{r} = \mathbf{Ax}_c - \mathbf{b}}$ (pro přesné řešení je $\mathbf{r} = \mathbf{Ax}^* - \mathbf{b} = \mathbf{0}$)

  Platí: Je-li $\mathbf{x}_c$ blízko $\mathbf{x}^*$ $\Rightarrow$ reziduum je malé. Bohužel to **neplatí obráceně** !!!

  $$\mathbf{r} = \mathbf{A}(\mathbf{x}_c - \mathbf{x}^*)$$

  $$\boxed{\mathbf{x}_c - \mathbf{x}^* = \mathbf{A}^{-1}\mathbf{r}}$$

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Pro špatně podmíněnou úlohu obsahuje $\mathbf{A}^{-1}$ velké prvky, které i pro malé hodnoty $\mathbf{r}$ mohou znamenat velkou chybu.

př:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0,99999 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 2 \\ 1,99999 \end{bmatrix}$$

přesné řešení: $\mathbf{x}^* = [1,1]^T$
vypočtené může být: $\mathbf{x}_c = [-98,100]^T$

$$\mathbf{r} = \mathbf{Ax}_c - \mathbf{b} = \begin{bmatrix} 2 \\ 1,999 \end{bmatrix} - \begin{bmatrix} 2 \\ 1,99999 \end{bmatrix} = \begin{bmatrix} 0 \\ -0,00099 \end{bmatrix}$$

$\Rightarrow$ je lepší pokoušet se odhadovat $||\mathbf{x}_c - \mathbf{x}^*||$.

Bohužel se v odhadech vždy vyskytuje norma $||\mathbf{A}^{-1}||$!!! Podrobněji viz literatura.

Kapitola 4. SLAR - iterační metody

Metody na řešení SLAR

- přímé (GEM, metoda LU-rozkladu) ✓
- iterační
- gradientní

Iterační metody najdou přesné řešení teoreticky až po nekonečně mnoha krocích.

Pamatujme si, že v numerické praxi používáme pro řešení soustav s plnou maticí **přímé metody**, zatímco pro speciální (řídké) matice používáme **iterační metody**.

Toto rozdělení je dáno **výpočetní složitostí** těchto metod, tj. počtem matematických operací sčítání, odčítání, násobení a dělení nutných k získání výsledku.

Poznámka: V případě plné matice je výpočetní cena v každé iteraci řádu n^2, srovnáme-li toto s celkovou výpočetní cenou přímých metod, tj. řádově $2/3\ n^3$, vidíme, že má-li být výpočetní složitost iterační metody stejná jako u přímé metody, musela by iterační metoda najít řešení (s předem zadanou přesností) řádově po n iteracích. Na druhou stranu v případě speciální (řídké) matice je výhodné použít iterační metodu.

Příklad

Uvažujme rovnici

$$9x = 9$$

Přesné řešení je

$$x^* = 1$$

Rovnici lze přepsat např. na tvar

$$10x - x = 9$$

$$x = \frac{9 + x}{10}$$

- viz metoda prosté iterace pro nelineární rovnice
- nyní uvažujeme lineární rovnice, proto předpis funkce $\varphi(x)$ může být lineární
- řešení hledáme pomocí rekurentní formule (volíme např. $x^{(0)} = 0$)

$$x^{(k+1)} = \frac{9 + x^{(k)}}{10}$$

Dostáváme

$$x^{(1)} = 0.9$$

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

$$x^{(2)} = 0.99$$
$$x^{(3)} = 0.999$$
$$x^{(4)} = 0.9999$$

Zastavíme např. pomocí podmínky na rozdíl dvou po sobě jdoucích iterací $|x^{(4)} - x^{(3)}| < \varepsilon = 0.001$

Uvedený postup realizujeme pro soustavy.

Podobně jako v metodě prosté iterace pro nelineární soustavy přepíšeme soustavu

$$\mathbf{Ax - b = o} \qquad \longleftrightarrow \qquad \mathbf{F(x) = o}$$

na tvar

$$\mathbf{x = Hx + g} \qquad \longleftrightarrow \qquad \mathbf{x = \Phi(x)}$$

Uvažujeme-li soustavu lineárních algebraických rovnic, tj. funkce $\mathbf{F}$ je lineární, můžeme potom najít lineární předpis pro funkci $\mathbf{\Phi}$.

Všechny iterační metody pro řešení soustavy lineárních algebraických rovnic budou používat iterační formuli

$$\boxed{\mathbf{x}^{(k+1)} = \mathbf{H} \cdot \mathbf{x}^{(k)} + \mathbf{g}}$$

samozřejmě s různou iterační maticí $\mathbf{H}$ a vektorem $\mathbf{g}$ a je zřejmé, že o kvalitě metody rozhodují právě vlastnosti matice $\mathbf{H}$.

Počáteční aproximaci $\mathbf{x}^{(0)}$ zvolíme a výpočet ukončíme pomocí zastavovací podmínky

$$||\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}|| < \varepsilon$$

Jacobiova metoda

Princip:

Z i-té rovnice vyjádříme i-tou složku vektoru $\mathbf{x}$

i-tá rovnice: $a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i$

pro $a_{ii} \neq 0$: $x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1,\ j \neq i}^{n} a_{ij} x_j \right)$

Iterační formule:

$$\boxed{x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1,\ j \neq i}^{n} a_{ij} x_j^{(k)} \right)}$$

Gaussova-Seidelova metoda

Princip:

Stejný jako u Jacobiovy metody s tím rozdílem, že jestliže při výpočtu $(k+1)$-iterace již známe $(k+1)$-iteraci některých složek, tak ji použijeme.

Z i-té rovnice vyjádříme i-tou složku vektoru $\mathbf{x}$

i-tá rovnice: $a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i$

pro $a_{ii} \neq 0$: $x_i = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1,\ j\neq i}^{n} a_{ij}x_j\right)$

Iterační formule:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)$$

Relaxační metoda SOR

Princip:

Vyjdeme z Gaussovy-Seidelovy metody jejíž iterační formule je

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right),$$

dále vyjádříme $(k+1)$-ní iteraci pomocí k-té iterace a příslušné změny (tj. přičteme a odečteme $x_i^{(k)}$)

$$x_i^{(k+1)} = x_i^{(k)} + \underbrace{\frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\right)}_{=r_i^{(k)}},$$

k urychlení výpočtu použijeme ideu, že nepřičteme k předchozí iteraci změnu $r_i^{(k)}$, ale její násobek $\omega r_i^{(k)}$

$$x_i^{(k+1)} = x_i^{(k)} + \omega \cdot \frac{1}{a_{ii}}\Big(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} \underbrace{- \sum_{j=i}^{n} a_{ij}x_j^{(k)}}_{-a_{ii}x_i^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}}\Big)$$

Iterační formule:

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \omega \underbrace{\frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)}_{x_i^{(k+1)} \text{ z G-S}}$$

Poznámka: $(k+1)$-iterace metody SOR je lineární kombinací $(k+1)$-iterace získané Gauss-Seidlovou metodou a předchozí k-té iterace metody SOR

$$x_i^{(k+1)} = \omega\, x_{i\,\mathrm{GS}}^{(k+1)} + (1-\omega)\, x_i^{(k)}.$$

Maticový zápis iteračních metod

Nejprve rozložíme matici $\mathbf{A}$:

$$\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U},$$

kde $\mathbf{L}$ je dolní trojúhelníková část matice $\mathbf{A}$ s nulami na diagonále, $\mathbf{D}$ je diagonální matice a $\mathbf{U}$ je horní trojúhelníková část matice $\mathbf{A}$ s nulami na diagonále.

Jacobiova metoda

$$\begin{aligned}
\mathbf{A}\mathbf{x} &= \mathbf{b}\\
(\mathbf{L}+\mathbf{D}+\mathbf{U})\mathbf{x} &= \mathbf{b}\\
\mathbf{D}\mathbf{x} + (\mathbf{L}+\mathbf{U})\mathbf{x} &= \mathbf{b}\\
\mathbf{D}\mathbf{x} &= \mathbf{b} - (\mathbf{L}+\mathbf{U})\mathbf{x}\\
\mathbf{x} &= \underbrace{-\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})}_{\mathbf{H}_J}\mathbf{x} + \underbrace{\mathbf{D}^{-1}\mathbf{b}}_{\mathbf{g}_J}
\end{aligned}$$

Gauss-Seidlova metoda

$$\begin{aligned}
\mathbf{A}\mathbf{x} &= \mathbf{b}\\
(\mathbf{L}+\mathbf{D}+\mathbf{U})\mathbf{x} &= \mathbf{b}\\
(\mathbf{L}+\mathbf{D})\mathbf{x} + \mathbf{U}\mathbf{x} &= \mathbf{b}\\
(\mathbf{L}+\mathbf{D})\mathbf{x} &= \mathbf{b} - \mathbf{U}\mathbf{x}\\
\mathbf{x} &= \underbrace{-(\mathbf{L}+\mathbf{D})^{-1}\mathbf{U}}_{\mathbf{H}_{GS}}\mathbf{x} + \underbrace{(\mathbf{L}+\mathbf{D})^{-1}\mathbf{b}}_{\mathbf{g}_{GS}}
\end{aligned}$$

Relaxační metoda SOR

$$\begin{aligned}
\mathbf{A}\mathbf{x} &= \mathbf{b} \\
\omega\mathbf{A}\mathbf{x} &= \omega\mathbf{b} \qquad /+\mathbf{D}\mathbf{x} \\
(\omega\mathbf{A}+\mathbf{D})\mathbf{x} &= \omega\mathbf{b}+\mathbf{D}\mathbf{x} \\
[\omega(\mathbf{L}+\mathbf{D}+\mathbf{U})+\mathbf{D})]\,\mathbf{x} &= \omega\mathbf{b}+\mathbf{D}\mathbf{x} \\
(\omega\mathbf{L}+\mathbf{D})\mathbf{x} &= \omega\mathbf{b}+\mathbf{D}\mathbf{x}-\omega\mathbf{D}\mathbf{x}-\omega\mathbf{U}\mathbf{x} \\
(\omega\mathbf{L}+\mathbf{D})\mathbf{x} &= [(1-\omega)\mathbf{D}-\omega\mathbf{U}]\,\mathbf{x}+\omega\mathbf{b} \\
\mathbf{x} &= \underbrace{(\omega\mathbf{L}+\mathbf{D})^{-1}\,[(1-\omega)\mathbf{D}-\omega\mathbf{U}]}_{\mathbf{H}_{SOR}}\mathbf{x}+\underbrace{(\omega\mathbf{L}+\mathbf{D})^{-1}\omega\mathbf{b}}_{\mathbf{g}_{SOR}}
\end{aligned}$$

Iterační metoda je dána formulí

$$\boxed{\mathbf{x}^{(k+1)}=\mathbf{H}\mathbf{x}^{(k)}+\mathbf{g}}$$

Pro přesné řešení $\mathbf{x}^*$ musí platit

- $\mathbf{x}^*=\mathbf{H}\mathbf{x}^*+\mathbf{g}$
- $\mathbf{x}^*=\mathbf{A}^{-1}\mathbf{b}$

$$\Rightarrow \qquad \boxed{\mathbf{A}^{-1}\mathbf{b}=\mathbf{H}\mathbf{A}^{-1}\mathbf{b}+\mathbf{g}} \qquad (*)$$

Definice: Iterační metodu $\mathbf{x}^{(k+1)}=\mathbf{H}\mathbf{x}^{(k)}+\mathbf{g}$ nazveme **konzistentní**, pokud platí $(*)$.

Poznámka: Uvedené metody jsou konzistentní.

- např. pro Jacobiovu metodu musí platit:

$$\mathbf{A}^{-1}\mathbf{b}=\underbrace{-\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})}_{\mathbf{H}_J}\mathbf{A}^{-1}\mathbf{b}+\underbrace{\mathbf{D}^{-1}\mathbf{b}}_{\mathbf{g}_J}$$

$$\mathbf{A}^{-1}\mathbf{b}=\underbrace{\mathbf{D}^{-1}\Big(\underbrace{-(\mathbf{L}+\mathbf{U})+\mathbf{A}}_{\mathbf{D}}\Big)}_{\mathbf{I}}\mathbf{A}^{-1}\mathbf{b}$$

- D.cv: Ukažte, že Gauss-Seidelova metoda a metoda SOR jsou konzistentní.

Definice: Iterační metoda $\mathbf{x}^{(k+1)}=\mathbf{H}\mathbf{x}^{(k)}+\mathbf{g}$ se nazývá **konvergentní**, jestliže pro každou počáteční aproximaci $\mathbf{x}^{(0)}\in\mathbb{R}$ platí:

$$\boxed{\lim_{k\to\infty}\mathbf{x}^{(k)}=\mathbf{x}^* \quad (=\mathbf{A}^{-1}\mathbf{b})}.$$

Chyba k-té iterace:

$$\boxed{\mathbf{e}^{(k)}=\mathbf{x}^{(k)}-\mathbf{x}^*}.$$

Nutná a postačující podmínka konvergence metody

iterační předpis	$\mathbf{x}^{(k)}=\mathbf{H}\mathbf{x}^{(k-1)}+\mathbf{g}$
konzistentní metoda	$\mathbf{x}^*=\mathbf{H}\mathbf{x}^*+\mathbf{g}$

Numerické metody KMA/NM
Josef Daněk 13.2.2013

po odečtení dostáváme

$$\mathbf{x}^{(k)}-\mathbf{x}^*=\mathbf{H}(\mathbf{x}^{(k-1)}-\mathbf{x}^*)$$
$$\mathbf{e}^{(k)}=\mathbf{H}\mathbf{e}^{(k-1)}$$

Platí:

$$\mathbf{e}^{(k)}=\mathbf{H}\mathbf{e}^{(k-1)}=\mathbf{H}^2\mathbf{e}^{(k-2)}=\ \ldots\ =\mathbf{H}^k\mathbf{e}^{(0)}$$

$$\lim_{k\to\infty}\mathbf{x}^{(k)}=\mathbf{x}^* \quad\Leftrightarrow\quad \lim_{k\to\infty}\mathbf{e}^{(k)}=\mathbf{0}$$

Věta: Daná konzistentní iterační metoda $\mathbf{x}^{k+1}=\mathbf{H}\mathbf{x}^{(k)}+\mathbf{g}$ konverguje pro libovolné $\mathbf{x}^{(0)}\in\mathbb{R}^n$ právě tehdy, když je **stabilní**, tj.

$$\boxed{\varrho(\mathbf{H})=\max_i|\lambda_i(\mathbf{H})|<1},$$

kde číslo $\varrho(\mathbf{H})$ nazýváme **spektrální poloměr** matice $\mathbf{H}$ a $\lambda_i(\mathbf{H})$ jsou vlastní čísla matice $\mathbf{H}$.

Poznámka: Připomeňme souvislost s metodou prosté iterace pro řešení soustav nelineárních rovnic. Funkce $\mathbf{\Phi}(\mathbf{x})$ z přepisu $\mathbf{x}=\mathbf{\Phi}(\mathbf{x})$ musela splňovat podmínku (b') ... (pokud byla diferencovatelná)

$$\boxed{\exists q\in\langle 0,1):\quad ||\Phi'(x)||\le q\quad\forall x}$$

V našem případě je $\mathbf{\Phi}(\mathbf{x})=\mathbf{H}\mathbf{x}+\mathbf{g}\ \Rightarrow\ \mathbf{\Phi}'(\mathbf{x})=\mathbf{H}$.

Tj.

$$\boxed{||\mathbf{H}||<1}.$$

Spektrální poloměr $\varrho(\mathbf{H})$ je také normou matice $\mathbf{H}$, tj. $\boxed{\varrho(\mathbf{H})<1}$.

Předchozí věta je silnější (kritérium) $\Leftrightarrow$

Věta pro prostou iteraci uváděla postačující podmínky $\Rightarrow$

Poznámka: Určovat $\varrho(\mathbf{H})$ je celkem drahé, proto za chvíli uvedeme větu (postačující podmínky) jejíž předpoklady se ověří snadněji.

Definice:

Maticovou normu $||.||$ nazveme **multiplikativní**, splňuje-li pro všechny čtvercové matice $\mathbf{A}$, $\mathbf{B}$ řádu n vztah

$$\boxed{||\mathbf{A}\cdot\mathbf{B}||\le||\mathbf{A}||\cdot||\mathbf{B}||}$$

Příklad 2

Řešme soustavu $\mathbf{A}\mathbf{x}=\mathbf{b}$ Jacobiovou metodou.

$$\mathbf{A}=\begin{bmatrix}1 & 0,9 & 0,9\\ 0,9 & 1 & 0,9\\ 0,9 & 0,9 & 1\end{bmatrix},\quad \mathbf{b}=\begin{bmatrix}2,8\\ 2,8\\ 2,8\end{bmatrix},\quad \text{přesné řešení } \mathbf{x}^*=\begin{bmatrix}1\\ 1\\ 1\end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 0 & -0,9 & -0,9 \\ -0,9 & 0 & -0,9 \\ -0,9 & -0,9 & 0 \end{bmatrix}$$

$\ldots$ vlastní čísla jsou $-1,8;\ 0,9;\ 0,9 \ \Rightarrow\ \varrho(\mathbf{H}) = 1,8 > 1 \ \Rightarrow$ metoda diverguje !!!

Uvažujme normu $||\mathbf{H}||_M = \max_{i,j} |h_{ij}|$

D.cv. Ukažte, že $||.||_M$ splňuje vlastnosti normy.

$$\boxed{||\mathbf{H}||_M = 0,9 < 1 \quad !!!}$$

$||.||_M$ není multiplikativní:

např:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}, \quad \mathbf{A}\cdot\mathbf{B} = \begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}$$

$$||\mathbf{A}||_M = 1, \quad ||\mathbf{B}||_M = 2, \quad ||\mathbf{A}\cdot\mathbf{B}||_M = 4$$

$$||\mathbf{A}||_M \cdot ||\mathbf{B}||_M = 2 \underbrace{<}_{!!!} ||\mathbf{A}\cdot\mathbf{B}||_M = 4$$

Věta: Pro každou multiplikativní maticovou normu $||.||$ a čtvercovou matici $\mathbf{A}$ platí:

$$\boxed{\varrho(\mathbf{A}) \le ||\mathbf{A}||}.$$

Důkaz:

$$\varrho(\mathbf{A}) = \max_i\{|\lambda_i|\} = |\lambda_p|$$

- nechť číslu λ_p odpovídá normovaný vlastní vektor $\mathbf{v}_p \quad (\nu(\mathbf{v}_p) = 1)$
- potom

$$\varrho(\mathbf{A}) = |\lambda_p| \cdot \nu(\mathbf{v}_p) = \nu(\lambda_p \mathbf{v}_p) = \nu(\mathbf{A}\mathbf{v}_p)$$

vlastnost kompatibilní maticové a vektorové normy:

$$\boxed{\nu(\mathbf{A}\mathbf{v}_p) \le ||\mathbf{A}|| \cdot \nu(\mathbf{v}_p)} = ||\mathbf{A}|| \cdot 1 = ||\mathbf{A}||$$

Pomocná věta: Ke každé multiplikativní maticové normě μ existuje kompatibilní vektorová norma ν.

Důkaz: Je dána maticová norma μ.

Definujeme $\nu(\mathbf{x}) = \mu([\mathbf{x}, \mathbf{0}, \mathbf{0}, \ldots, \mathbf{0}])$ ... splňuje vlastnosti normy

? kompatibilita:

Numerické metody KMA/NM
Josef Daněk 13.2.2o13

$$\nu(\mathbf{A}\mathbf{x}) = \mu([\mathbf{A}\mathbf{x}, \mathbf{0}, \mathbf{0}, \ldots, \mathbf{0}]) = \mu(\mathbf{A}\,[\mathbf{x}, \mathbf{0}, \mathbf{0}, \ldots, \mathbf{0}]) \underbrace{\le}_{(*)} \mu(\mathbf{A})\mu([\mathbf{x}, \mathbf{0}, \mathbf{0}, \ldots, \mathbf{0}]) = \mu(\mathbf{A}) \cdot \nu(\mathbf{x})$$

$(*)$ μ je multiplikativní

Věta (Postačující podmínka konvergence)

Je-li pro multiplikativní normu splněna podmínka $\boxed{||\mathbf{H}|| \le q < 1}$, potom posloupnost $\left\{\mathbf{x}^{(k)}\right\}$ určená konzistentní formulí

$$\mathbf{x}^{(k)} = \mathbf{H}\mathbf{x}^{(k-1)} + \mathbf{g}$$

konverguje při libovolné volbě vektoru $\mathbf{x}^{(0)} \in \mathbb{R}^n$ a platí

$$\boxed{\lim_{k\to\infty} \mathbf{x}^{(k)} = (\mathbf{I} - \mathbf{H})^{-1}\mathbf{g} = \mathbf{x}^*}.$$

Důkaz: Důsledek kritéria a předchozí věty (jiný důkaz viz skripta).

Odhad chyby

Předpokládáme, že je splněna postačující podmínka konvergence

$$||\mathbf{H}|| \le q < 1$$

$$\begin{aligned} \mathbf{x}^{(k)} - \mathbf{x}^* &= \mathbf{H}(\mathbf{x}^{(k-1)} - \mathbf{x}^*) = \\ &= \mathbf{H}(\mathbf{x}^{(k-1)} - \mathbf{x}^{(k)} + \mathbf{x}^{(k)} - \mathbf{x}^*) = \\ &= \mathbf{H}(\mathbf{x}^{(k-1)} - \mathbf{x}^{(k)}) + \mathbf{H}(\mathbf{x}^{(k)} - \mathbf{x}^*) \end{aligned}$$

Odtud dostáváme

$$\left\|\mathbf{x}^{(k)} - \mathbf{x}^*\right\| \ \le\ q\left\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right\| + q\left\|\mathbf{x}^{(k)} - \mathbf{x}^*\right\|$$

tj.

$$\underbrace{(1-q)}_{>0}\left\|\mathbf{x}^{(k)} - \mathbf{x}^*\right\| \ \le\ q\left\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right\|$$

a po vydělení

$$\boxed{\left\|\mathbf{x}^{(k)} - \mathbf{x}^*\right\| \ \le\ \frac{q}{1-q}\left\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right\|}.$$

Jestliže $\left\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right\| < \varepsilon$, potom

$$\boxed{\left\|\mathbf{x}^{(k)} - \mathbf{x}^*\right\| \ \le\ \frac{q}{1-q}\,\varepsilon}.$$

Příklad 3

Jacobiovou metodou řešte soustavu $\mathbf{Ax} = \mathbf{b}$.

$$\mathbf{A} = \begin{bmatrix} 8 & 4 & 2 \\ 1 & 10 & 1 \\ 0 & 0 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 14 \\ 12 \\ 2 \end{bmatrix}, \quad \mathbf{x}^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{A} = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{L}} + \underbrace{\begin{bmatrix} 8 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 2 \end{bmatrix}}_{\mathbf{D}} + \underbrace{\begin{bmatrix} 0 & 4 & 2 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{U}}$$

$$\mathbf{H}_J = -\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U}) = \begin{bmatrix} 0 & -0,5 & -0,25 \\ -0,1 & 0 & -0,1 \\ 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{g}_J = \mathbf{D}^{-1}\mathbf{b} = \begin{bmatrix} 1,75 \\ 1,2 \\ 1 \end{bmatrix}$$

Provedeme 5 iterací:

k	x_1^k	x_2^k	x_3^k
0	0	0	0
1	1,75	1,2	1
2	0,9	0,925	1
3	1,0375	1,01	1
4	0,995	0,99625	1
5	1,001875	1,0005	1

$$\underbrace{\mathbf{x}^{(5)} - \mathbf{x}^{(4)}}_{=\mathbf{r}^{(5)}} = \begin{bmatrix} 0,006875 \\ 0,004250 \\ 0,000000 \end{bmatrix}$$

Odhadněme chybu $\mathbf{x}^{(5)}$, tj.

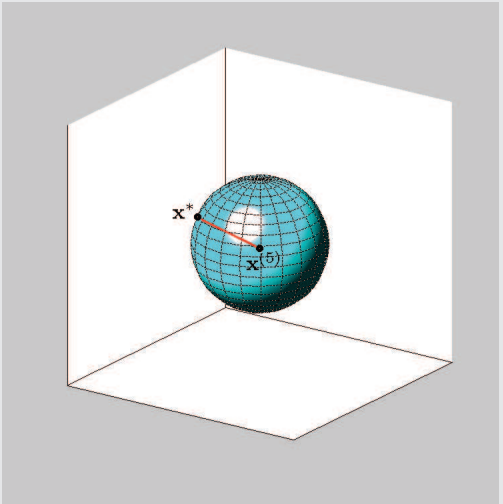
$$||\mathbf{x}^{(5)} - \mathbf{x}^*|| \leq \frac{||\mathbf{H}||}{1 - ||\mathbf{H}||} \cdot ||\mathbf{x}^{(5)} - \mathbf{x}^{(4)}||$$

maticová norma	vektorová norma	odhad chyby
$\|\|\mathbf{H}\|\|_S = \max_k(\sum_i h_{ik}) = 0,5$	$\|\|\mathbf{r}^{(5)}\|\|_1 = \sum_i \|r_i\| = 0,011125$	$\frac{0,5}{1-0,5} \cdot 0,011125 = \mathbf{0,01112}$
$\|\|\mathbf{H}\|\|_R = \max_i(\sum_k h_{ik}) = 0,75$	$\|\|\mathbf{r}^{(5)}\|\|_\infty = \max_i \|r_i\| = 0,006875$	$\frac{0,75}{0,25} \cdot 0,006875 = \mathbf{0,0206}$
$\|\|\mathbf{H}\|\|_{SP} = \max_k \lambda_k^{\frac{1}{2}}(\mathbf{H}^H\mathbf{H}) \doteq 0,56$	$\|\|\mathbf{r}^{(5)}\|\|_2 = \sum_i r_i^2 \doteq 0,0081$	$\frac{0,56}{0,44} \cdot 0,0081 = \mathbf{0,0103}$

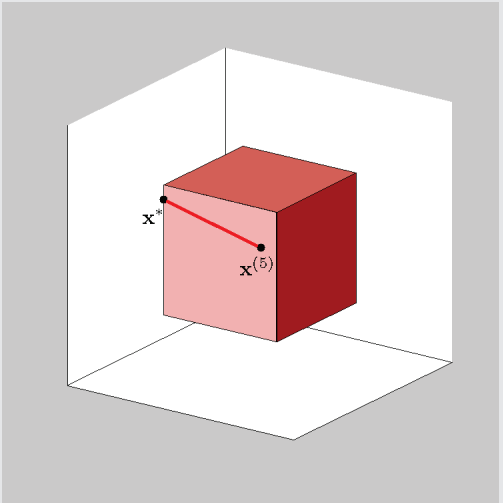
Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Geometrický význam

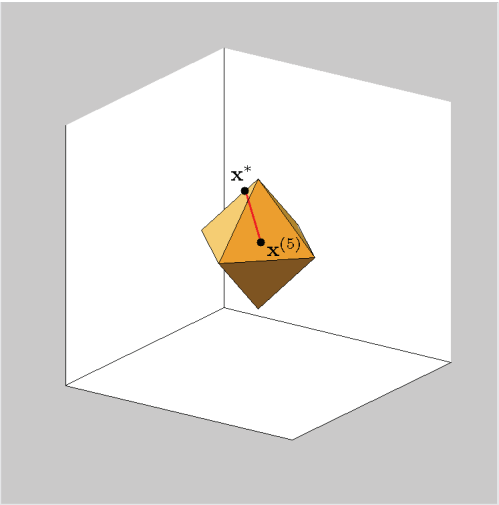
$||\mathbf{x}^{(5)} - \mathbf{x}^*||_2 \leq 0,0103$ vzdálenost $\mathbf{x}^{(5)}$ a $\mathbf{x}^*$ je menší než vypočtená hodnota



$||\mathbf{x}^{(5)} - \mathbf{x}^*||_\infty \leq 0,0206$



$||\mathbf{x}^{(5)} - \mathbf{x}^*||_1 \leq 0,011125$



Rychlost konvergence

- Lineární rychlost konvergence

$$\exists q \in (0,1)\ \exists k_0 \geq 0\ \forall k > k_0 : \quad \boxed{||\mathbf{x}^{(k+1)} - \mathbf{x}^*|| \leq q||\mathbf{x}^{(k)} - \mathbf{x}^*||}$$

- Superlineární rychlost konvergence

$$\boxed{||\mathbf{x}^{(k+1)} - \mathbf{x}^*|| \leq q_k||\mathbf{x}^{(k)} - \mathbf{x}^*||, \quad \text{kde} \quad q_k \to 0\ (k \to \infty)}$$

- Konvergence řádu **r**

$$\boxed{||\mathbf{x}^{(k+1)} - \mathbf{x}^*|| \leq q||\mathbf{x}^{(k)} - \mathbf{x}^*||^r}$$

Poznámka:

- Jacobiova, Gauss-Seidelova i SOR metoda mají lineární rychlost konvergence

$$\mathbf{x}^{(k+1)} - \mathbf{x}^* = \mathbf{H}(\mathbf{x}^{(k)} - \mathbf{x}^*)$$

během výpočtu se nemění iterační matice $\mathbf{H}$, jedná se o **stacionární metody**

$$||\mathbf{H}|| \leq q, \quad ||\mathbf{H}|| \ \ldots \text{ pevné číslo}$$

- Metody se superlineární rychlostí konvergence patří mezi **nestacionární procesy**

$$\boxed{\mathbf{x}^{(k+1)} = \mathbf{H}_k\mathbf{x}^{(k)} + \mathbf{g}_k}$$

V každém kroku se mohou měnit $\mathbf{H}_k$, $\mathbf{g}_k$.
Potom $||\mathbf{H}_k|| \leq q_k$, platí-li $q_k \to 0$ pak jde o superlineární metodu.

Definujeme asymptotickou rychlost konvergence

$$\boxed{R = -\log\frac{||\mathbf{e}^{(k)}||}{||\mathbf{e}^{(k-1)}||} \geq -\log||\mathbf{H}||},$$

ta určuje počet platných desetinných míst získaných v jednom iteračním kroku.

Prakticky:

$$\frac{||\mathbf{e}^{(k)}||}{||\mathbf{e}^{(k-1)}||} = \frac{||\mathbf{x}^{(k)} - \mathbf{x}^*||}{||\mathbf{x}^{(k-1)} - \mathbf{x}^*||} \approx \frac{\frac{q}{1-q}}{\frac{q}{1-q}}\frac{||\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}||}{||\mathbf{x}^{(k-1)} - \mathbf{x}^{(k-2)}||}$$

Poznámka: Pro metody s lineární rychlostí konvergence (Jacobiova, Gauss-Seidelova, SOR metoda) lze pro urychlení použít **Aitkenovu extrapolační formuli** (viz dříve).

Posloupnost chyb je geometrická

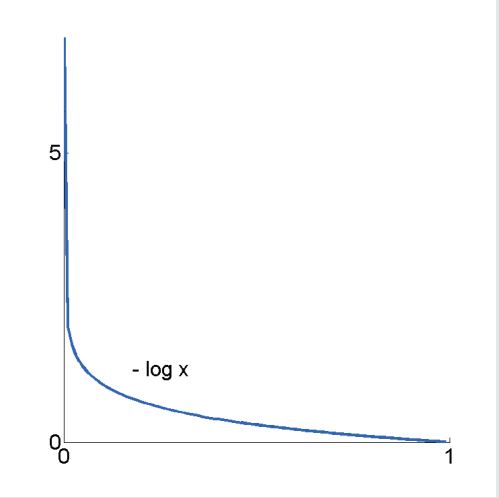
$$\mathbf{e}^{(k)} = \mathbf{H}\mathbf{e}^{(k-1)}$$

$$\frac{\overbrace{x_i^{(k+1)} - x_i^*}^{e_i^{(k+1)}}}{\underbrace{x_i^{(k)} - x_i^*}_{e_i^{(k)}}} \approx \frac{\overbrace{x_i^{(k)} - x_i^*}^{e_i^{(k)}}}{\underbrace{x_i^{(k-1)} - x_i^*}_{e_i^{(k-1)}}}$$

po úpravě:

$$\boxed{x_i^* \approx x_i^{(k+1)} - \frac{(x_i^{(k+1)} - x_i^{(k)})^2}{x_i^{(k+1)} - 2x_i^{(k)} + x_i^{(k-1)}}}$$

Při odvození metody SOR jsme se pokusili urychlit výpočet změnou iterační matice $\mathbf{H}$ tak, aby měla menší spektrální poloměr $\varrho(\mathbf{H})$. Čím menší je $\varrho(\mathbf{H})$, tím je větší asymptotická rychlost konvergence.



Věta: Spektrální poloměr $\varrho(\mathbf{H}_{SOR})$ splňuje podmínku

$$\varrho(\mathbf{H}_{SOR}) \geq |\omega - 1| \quad \forall \omega \in \mathbb{R}$$

Důkaz:

$$\mathbf{H}_{SOR} = (\omega\mathbf{L} + \mathbf{D})^{-1}\left[(1-\omega)\mathbf{D} - \omega\mathbf{U}\right]$$

Je známo, že součin vlastních čísel je roven determinantu

$$\prod_{i=1}^{n} \lambda_i = \det(\mathbf{H}_{SOR})$$

$$\det(\mathbf{H}_{SOR}) = \det\left[(\omega\mathbf{L} + \mathbf{D})^{-1}\left[(1-\omega)\mathbf{D} - \omega\mathbf{U}\right]\right] = \det\left[\left(\mathbf{D}(\omega\mathbf{D}^{-1}\mathbf{L} + \mathbf{I})\right)^{-1}\mathbf{D}\left[(1-\omega)\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{U}\right]\right] =$$

$$= \det\left[(\omega\mathbf{D}^{-1}\mathbf{L}+\mathbf{I})^{-1}\left[(1-\omega)\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{U}\right]\right] = \det\underbrace{(\omega\mathbf{D}^{-1}\mathbf{L} + \mathbf{I})}_{(*)}{}^{-1} \cdot \det\underbrace{\left[(1-\omega)\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{U}\right]}_{(**)} = (1-\omega)^n$$

$(*)$ dolní trojúhelníková matice s 1 na diagonále
$(**)$ horní trojúhelníková matice a prvky $(1-\omega)$ na diagonále

---

$$\prod_{i=1}^{n} \lambda_i = (1-\omega)^n \underbrace{\Rightarrow}_{(***)} \boxed{\underbrace{\max_i |\lambda_i|}_{\varrho(\mathbf{H}_{SOR})} \geq |1-\omega|}$$

$(***)$ Důkaz sporem: $\forall \lambda_i$: $|\lambda_i| < |1-\omega| \quad / \prod_{i=1}^{n}$

$$\prod_{i=1}^{n} |\lambda_i| < |1-\omega|^n$$

Důsledek: Aby SOR konvergovala, musí platit:

$$|\omega - 1| \leq \boxed{\varrho(\mathbf{H}_{SOR}) < 1}$$

$$|\omega - 1| < 1 \Rightarrow \boxed{\omega \in (0,2)}$$

Poznámky:

Parametr ω v relaxační metodě SOR volíme z intervalu $(0,2)$.

Pro $\omega = 1$ přejde relaxační metoda na Gauss-Seidlovu metodu.

Volba parametru ω samozřejmě ovlivní rychlost konvergence iteračního procesu metody SOR.
Lze ukázat, že existuje optimální hodnota parametru omega

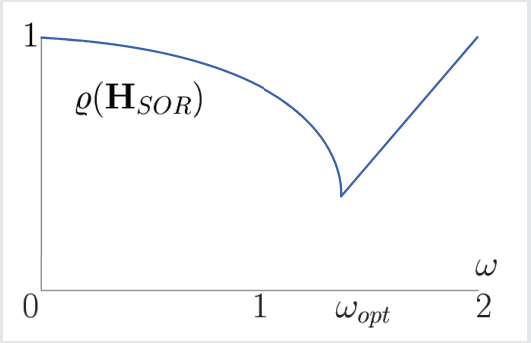
$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \varrho^2(\mathbf{H}_J)}},$$

Numerické metody KMA/NM
Josef Daněk 13.2.2013

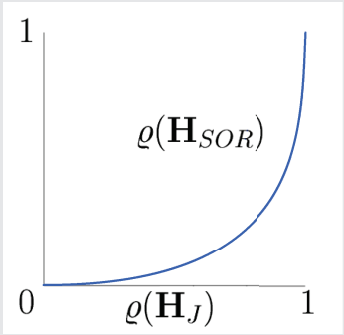
kde $\varrho(\mathbf{H}_J)$ je spektrální poloměr Jacobiovy iterační matice $\mathbf{H}_J$.

Pro spektrální poloměr iterační matice $\mathbf{H}_{SOR}$ relaxační metody lze odvodit následující závislosti:

- závislost spektrálního poloměru iterační matice metody SOR na relaxačním parametru ω



- závislost spektrálního poloměru iterační matice metody SOR na spektrálním poloměru iterační matice Jacobiovy metody



Konvergenční věty

Dosud jsme udávali podmínky pro iterační matici $\mathbf{H}$. To je ovšem nepraktické. Uvedeme několik snadněji ověřitelných podmínek.

Věta 1 Je-li matice $\mathbf{A}$ ostře diagonálně-dominantní, potom konverguje Jacobiova i Gauss-Seidelova metoda pro libovolnou volbu $\mathbf{x}^{(0)}$.

Věta 2 Je-li matice $\mathbf{A}$ symetrická a pozitivně-definitní, potom Gauss-Seidelova metoda konverguje pro libovolnou volbu $\mathbf{x}^{(0)}$.

Věta 3 Nutnou podmínkou konvergence metody SOR je $0 < \omega < 2$. Přidáme-li symetrii a pozitivní definitnost matice $\mathbf{A}$, dostaneme postačující podmínky konvergence.

Důkaz Věty 1 pro Jacobiovu metodu:

$$\mathbf{Ax} = \mathbf{b}, \qquad \text{rozklad matice } \mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$$

$$\text{označíme-li } \mathbf{C} = \mathbf{L} + \mathbf{U}, \text{ potom } \mathbf{A} = \mathbf{C} + \mathbf{D}$$

- Jacobiova metoda

$$\mathbf{x}^{(k+1)} = \mathbf{H}_J \mathbf{x}^{(k)} + \mathbf{g}_J$$
$$\mathbf{H}_J = -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U}) = -\mathbf{D}^{-1}\mathbf{C} \quad \text{a} \quad \mathbf{g}_J = \mathbf{D}^{-1}\mathbf{b}$$

- Matice $\mathbf{A}$ je ostře diagonálně-dominantní, tj. platí

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}| \qquad i = 1, 2, \ldots, n$$

- Pro náš rozklad $\mathbf{A} = \mathbf{C} + \mathbf{D}$ tedy platí:

$$|d_{ii}| > \sum_{j=1}^{n} |c_{ij}| \qquad i = 1, 2, \ldots, n \qquad / : |d_{ii}| \neq 0$$

(kdyby $|d_{ii}| = 0$, potom by byl celý řádek nulový ... $\mathbf{A}$ je ale regulární)

- Platí:

$$\boxed{\sum_{j=1}^{n} \frac{|c_{ij}|}{|d_{ii}|} < 1} \qquad \text{pro } i = 1, 2, \ldots, n \qquad (*)$$

- Pro iterační matici platí:

$$\mathbf{H}_J = -\mathbf{D}^{-1}\mathbf{C} = -\begin{bmatrix} \frac{1}{d_{11}} & & \\ & \frac{1}{d_{22}} & \\ & & \ddots \\ & & & \frac{1}{d_{nn}} \end{bmatrix} \cdot \begin{bmatrix} \\ c_{ij} \\ \\ \end{bmatrix} = -\begin{bmatrix} \\ \frac{c_{ij}}{d_{ii}} \\ \\ \end{bmatrix}$$

- Řádková norma matice $\mathbf{H}_J$:

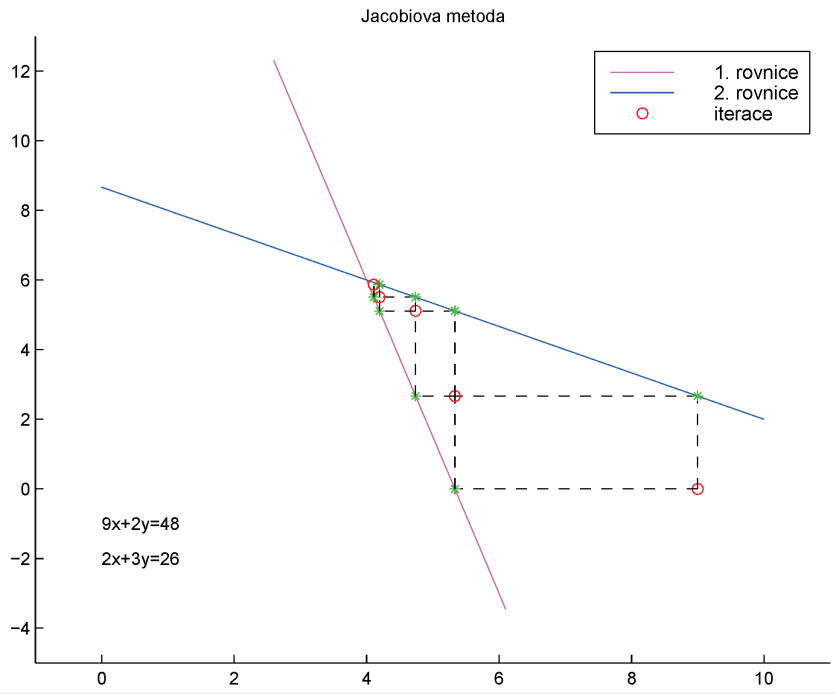
$$\|\mathbf{H}_J\| = \max_i \sum_{j=1}^{n} \left|\frac{c_{ij}}{d_{ii}}\right| < 1 \qquad \text{plyne z } (*)$$

Geometrický význam Jacobiovy metody

$$\begin{aligned} 9x &+ 2y = 48 \qquad & x^{(k+1)} &= \tfrac{1}{9}(48 - 2y^{(k)}) \\ 2x &+ 3y = 26 \qquad & y^{(k+1)} &= \tfrac{1}{3}(26 - 2x^{(k+1)}) \end{aligned}$$

k	$x^{(k)}$	$y^{(k)}$
0	9.0000	0
1	5.3333	2.6667
2	4.7407	5.1111
3	4.1975	5.5062
4	4.1097	5.8683
5	4.0293	5.9268

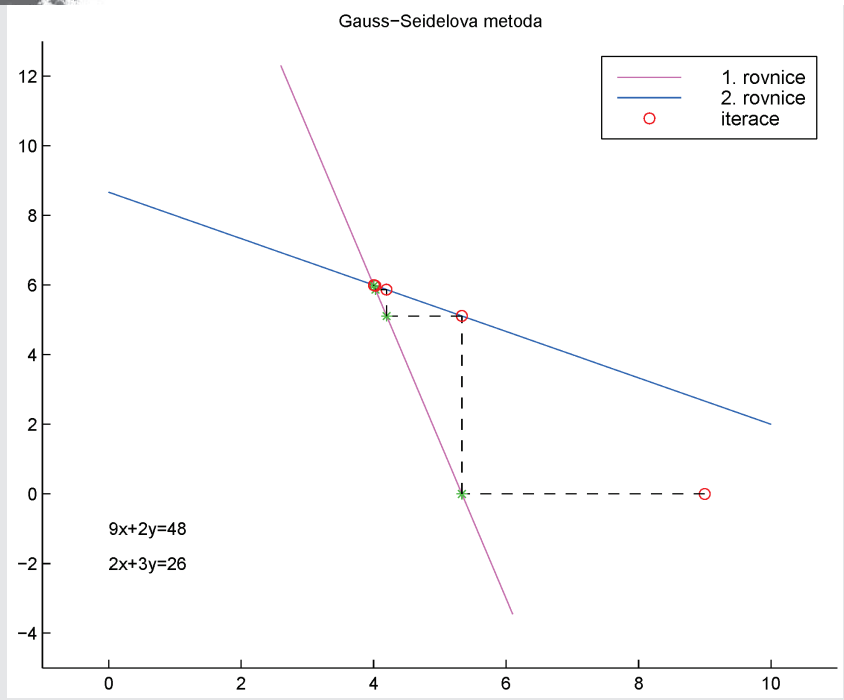
Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



Geometrický význam Gauss-Seidelovy metody

$$\begin{aligned} 9x &+ 2y = 48 \qquad & x^{(k+1)} &= \tfrac{1}{9}(48 - 2y^{(k)}) \\ 2x &+ 3y = 26 \qquad & y^{(k+1)} &= \tfrac{1}{3}(26 - 2x^{(k+1)}) \end{aligned}$$

k	$x^{(k)}$	$y^{(k)}$
0	9.0000	0
1	5.3333	5.1111
2	4.1975	5.8683
3	4.0293	5.9805
4	4.0043	5.9971
5	4.0006	5.9996



Geometrický význam metody SOR ($\omega < 1$)

$$9x + 2y = 48 \qquad x_{GS}^{(k+1)} = \tfrac{1}{9}(48 - 2y^{(k)})$$

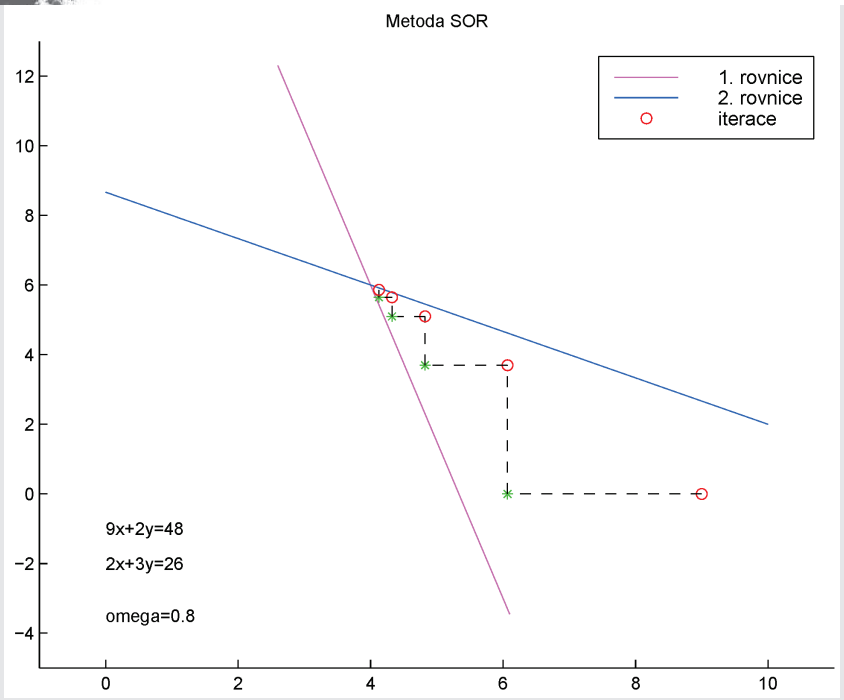
$$2x + 3y = 26 \qquad y_{GS}^{(k+1)} = \tfrac{1}{3}(26 - 2x^{(k+1)})$$

$$x^{(k+1)} = \omega\tfrac{1}{9}(48 - 2y^{(k)}) + (1-\omega)x^{(k)}$$

$$y^{(k+1)} = \omega\tfrac{1}{3}(26 - 2x^{(k+1)}) + (1-\omega)y^{(k)}$$

k	$x^{(k)}$	$y^{(k)}$
0	9.0000	0
1	6.0667	3.6978
2	4.8226	5.1008
3	4.3244	5.6472
4	4.1276	5.8614
5	4.0502	5.9455

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



Geometrický význam metody SOR ($\omega > 1$)

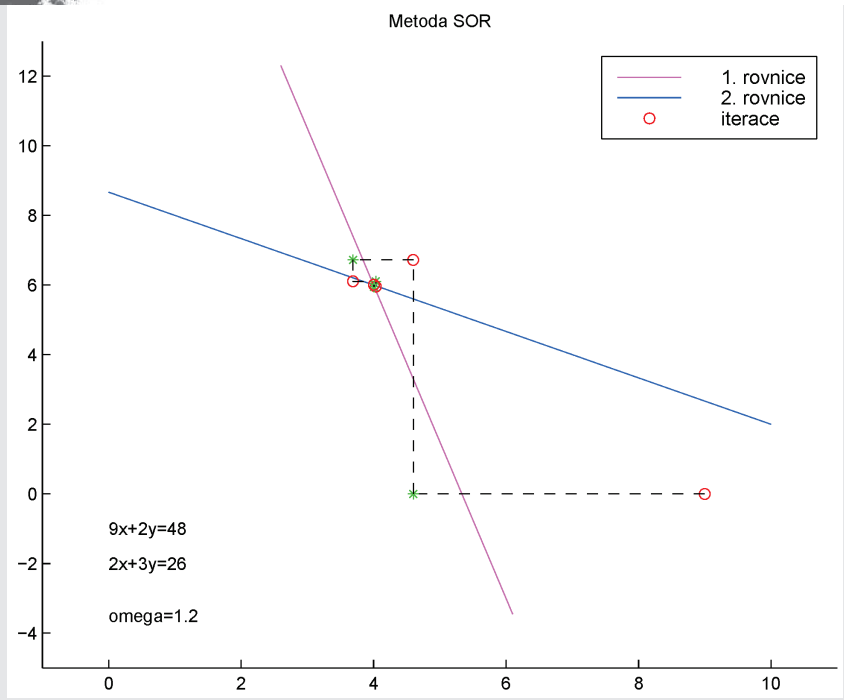
$$9x + 2y = 48 \qquad x_{GS}^{(k+1)} = \tfrac{1}{9}(48 - 2y^{(k)})$$

$$2x + 3y = 26 \qquad y_{GS}^{(k+1)} = \tfrac{1}{3}(26 - 2x^{(k+1)})$$

$$x^{(k+1)} = \omega\tfrac{1}{9}(48 - 2y^{(k)}) + (1-\omega)x^{(k)}$$

$$y^{(k+1)} = \omega\tfrac{1}{3}(26 - 2x^{(k+1)}) + (1-\omega)y^{(k)}$$

k	$x^{(k)}$	$y^{(k)}$
0	9.0000	0
1	4.6000	6.7200
2	3.6880	6.1056
3	4.0342	5.9515
4	4.0061	6.0048
5	3.9975	6.0010



Numerické metody
Josef Daněk
KMA/NM
13.2.2013

Kapitola 5. SLAR - gradientní metody

Metody na řešení SLAR

- přímé (GEM, metoda LU-rozkladu) ✓
- iterační (Jacobiova m., Gauss-Seidelova m., metoda SOR) ✓
- gradientní

Motivace

Uvažujme kvadratickou funkci reálné proměnné x:

$$f(x) = \frac{1}{2}ax^2 - bx + c, \quad a > 0.$$

Nutná a postačující podmínka minima funkce $\left(f'(x) = 0\right)$ má tvar

$$ax = b.$$

To znamená, že místo řešení lineární rovnice můžeme řešit úlohu najít minimum konvexní kvadratické funkce $f(x)$ (obě úlohy mají stejná řešení).

Uvědomme si, že v případě funkce více proměnných je třeba splnit další podmínky kladené na matici soustavy $\mathbf{A}$, abychom zaručili konvexnost příslušné kvadratické funkce.

Uvažujeme soustavu (kde matice $\mathbf{A}$ je symetrická, pozitivně definitní)

$$\mathbf{Ax} = \mathbf{b}$$

Dále uvažujeme kvadratickou formu, tzv. **energetický funkcionál**

$$F(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Ax} - \mathbf{b}^T\mathbf{x}.$$

Platí

$$\operatorname{grad} F(\mathbf{x}) = \mathbf{A}\,\mathbf{x} - \mathbf{b}.$$

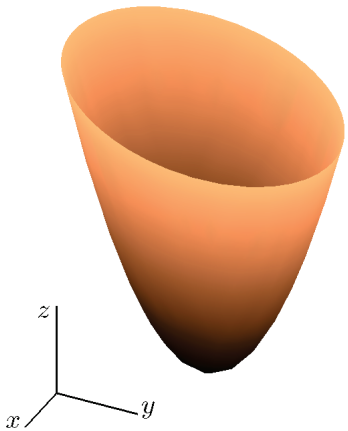
Funkce $F(\mathbf{x})$ je konvexní a kvadratická $\Rightarrow$ $F(\mathbf{x})$ má globální minimum a pro bod minima $\tilde{\mathbf{x}}$ platí

$$\operatorname{grad} F(\tilde{\mathbf{x}}) = \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b} = \mathbf{0}.$$

Bod minima $\tilde{\mathbf{x}}$ je tedy řešením soustavy $\mathbf{Ax} = \mathbf{b}$.

Poznámka: Úlohy najít bod minima funkce F a řešit soustavu $\mathbf{Ax} = \mathbf{b}$ jsou ekvivalentní.

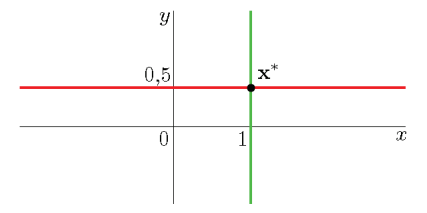
Poznámka: V případě soustavy 2 rovnic si lze udělat geometrickou představu, neboť pro $\mathbf{x} \in \mathbb{R}^2$ je grafem funkce $F(\mathbf{x})$ eliptický paraboloid, jehož vrstevnice jsou elipsy. Minima $F(\mathbf{x})$ se nabývá ve vrcholu paraboloidu.



Příklad 1

Uvažujme velmi jednoduchou soustavu $\mathbf{Ax} = \mathbf{b}$, kde

$$\mathbf{A} = \begin{bmatrix} 25 & 0 \\ 0 & 16 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 25 \\ 8 \end{bmatrix}, \quad \mathbf{x}^* = \begin{bmatrix} 1 \\ 0{,}5 \end{bmatrix}.$$



Odpovídající kvadratická funkce je

$$F(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} = \frac{1}{2}\begin{bmatrix} x & y \end{bmatrix}\begin{bmatrix} 25 & 0 \\ 0 & 16 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} 25 & 8 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{2}(25x^2 + 16y^2) - 25x - 8y.$$

Vrstevnice (hladiny):

$$\boxed{F(\mathbf{x}) = c}$$

$$\frac{1}{2}(25x^2 + 16y^2) - 25x - 8y = c$$

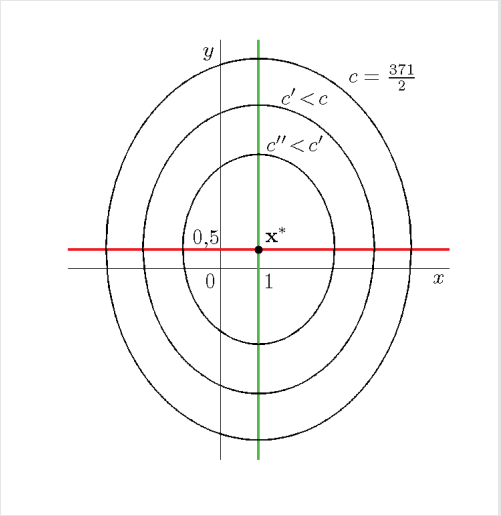
$$25x^2 + 16y^2 - 50x - 16y = 2c$$

$$25(x-1)^2 - 25 + 16(y - \frac{1}{2})^2 - 4 = 2c$$

$$25(x-1)^2 + 16(y - \frac{1}{2})^2 = 2c + 29$$

např. pro $c = \frac{371}{2}$:

$$\boxed{\frac{(x-1)^2}{4^2} + \frac{(y-\frac{1}{2})^2}{5^2} = \frac{2c+29}{400} = 1}$$

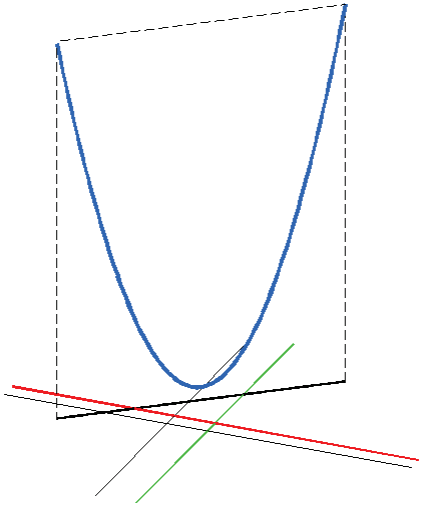


Řezy svislou rovinou $y = px + q$

$$F(x) = \frac{1}{2}(25x^2 + 16y^2) - 25x - 8y = \frac{1}{2}(25x^2 + 16(px+q)^2) - 25x - 8(px+q) =$$

$$= \frac{1}{2}(25x^2 + 16(p^2x^2 + 2pqx + q^2)) - 25x - 8(px+q) =$$

$$= \underbrace{(\frac{25}{2} + \frac{16}{2}p^2)}_{>0}x^2 + (16pq - 25 - 8p)x + 8q^2 - 8q$$



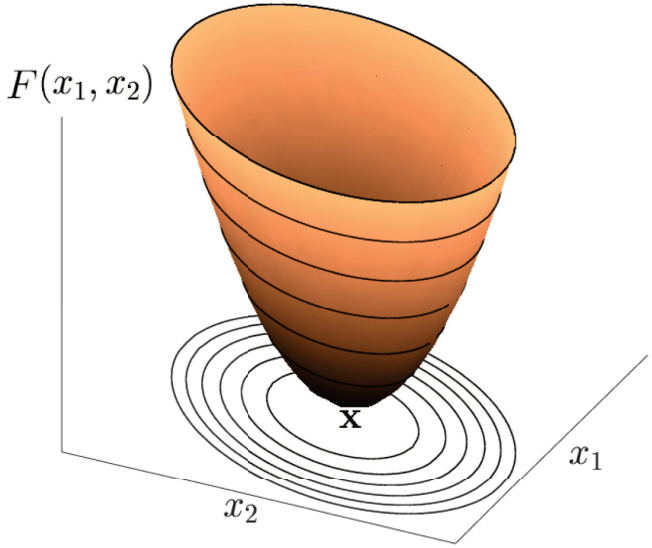
Princip

Stejně jako u každé iterační metody nejprve zvolíme počáteční aproximaci řešení $\mathbf{x}^{(0)}$.

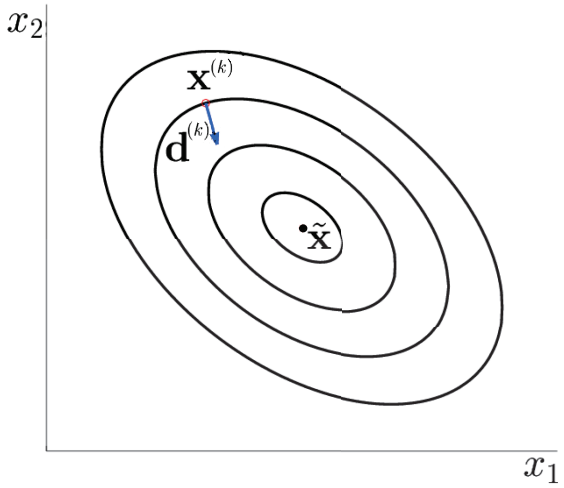
Princip gradientních metod spočívá v tom, že zvolíme směr a v tomto směru se budeme chtít co nejvíce přiblížit k přesnému řešení. Gradientní metoda je tedy určena volbou směrů, ve kterých minimalizujeme funkci F.

Během jedné iterace se pohybujeme po povrchu grafu funkce $F(\mathbf{x})$ tak, abychom se dostali na nižší vrstevnici.

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



V případě soustavy dvou rovnic získáme promítnutím grafu funkce $F(\mathbf{x})$ do roviny proměnných x_1, x_2 systém soustředných elips - hladin (vrstevnic).



Metoda největšího spádu

Metodu největšího spádu získáme, pokud budeme za směrové vektory volit směry největšího spádu, tj. vektory

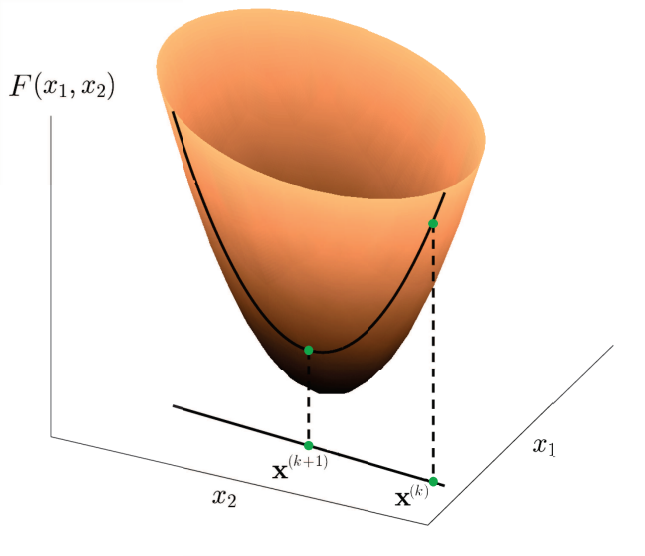
$$\boxed{\mathbf{d}^{(k)} = -\operatorname{grad} F(\mathbf{x}^{(k)}) = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}}.$$

Iterační formuli volíme ve tvaru

$$\boxed{\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)} \cdot \mathbf{d}^{(k)}},$$

v každém kroku metody určíme směr největšího spádu $\mathbf{d}^{(k)}$ a provedeme jednorozměrnou minimalizaci v tomto směru, tj.

$$\min_{t>0} F(\mathbf{x}^{(k)} + t\,\mathbf{d}^{(k)}).$$



Minimalizovanou funkci proměnné t označíme $\Psi(t)$.

Potom platí:

$$\underbrace{F(\mathbf{x}^{(k)} + t\,\mathbf{d}^{(k)})}_{\Psi(t)} = \frac{1}{2}(\mathbf{x}^{(k)} + t\,\mathbf{d}^{(k)})^T\mathbf{A}(\mathbf{x}^{(k)} + t\,\mathbf{d}^{(k)}) - \mathbf{b}^T(\mathbf{x}^{(k)} + t\,\mathbf{d}^{(k)}) =$$

$$= \frac{1}{2}\mathbf{x}^{(k)T}\mathbf{A}\mathbf{x}^{(k)} + \frac{1}{2}t\,\mathbf{x}^{(k)T}\mathbf{A}\mathbf{d}^{(k)} + \frac{1}{2}t\,\mathbf{d}^{(k)T}\mathbf{A}\mathbf{x}^{(k)} + \frac{1}{2}t^2\,\mathbf{d}^{(k)T}\mathbf{A}\mathbf{d}^{(k)} - \mathbf{b}^T\mathbf{x}^{(k)} - t\,\mathbf{b}^T\mathbf{d}^{(k)}$$

$$\frac{d\Psi(t)}{dt} = \frac{1}{2}\mathbf{x}^{(k)T}\mathbf{A}\mathbf{d}^{(k)} + \frac{1}{2}\mathbf{d}^{(k)T}\mathbf{A}\mathbf{x}^{(k)} + t\,\mathbf{d}^{(k)T}\mathbf{A}\mathbf{d}^{(k)} - \mathbf{b}^T\mathbf{d}^{(k)}$$

Poznámka: První 2 členy, tj. $\mathbf{x}^{(k)T}\mathbf{A}\mathbf{d}^{(k)}$ a $\mathbf{d}^{(k)T}\mathbf{A}\mathbf{x}^{(k)}$ jsou skaláry a jsou si pro symetrickou matici $\mathbf{A}$ rovny.

$$\mathbf{d}^{(k)T}\mathbf{A}\mathbf{x}^{(k)} = (\mathbf{d}^{(k)T}\mathbf{A}\mathbf{x}^{(k)})^T = (\mathbf{A}\mathbf{x}^{(k)})^T\mathbf{d}^{(k)} = \mathbf{x}^{(k)T}\mathbf{A}^T\mathbf{d}^{(k)}$$

$$\frac{d\Psi(t)}{dt} = t\,\mathbf{d}^{(k)T}\mathbf{A}\mathbf{d}^{(k)} + \underbrace{\underbrace{\mathbf{x}^{(k)T}\mathbf{A}\mathbf{d}^{(k)} - \mathbf{b}^T\mathbf{d}^{(k)}}_{(\mathbf{x}^{(k)T}\mathbf{A} - \mathbf{b}^T)\,\mathbf{d}^{(k)}}}_{-\mathbf{d}^{(k)T}}$$

$$\frac{d\Psi(t)}{dt} = t\,\mathbf{d}^{(k)T}\mathbf{A}\mathbf{d}^{(k)} + \underbrace{\underbrace{\mathbf{x}^{(k)T}\mathbf{A}\mathbf{d}^{(k)} - \mathbf{b}^T\mathbf{d}^{(k)}}_{(\mathbf{x}^{(k)T}\mathbf{A} - \mathbf{b}^T)\,\mathbf{d}^{(k)}}}_{-\mathbf{d}^{(k)T}}$$

$$\frac{d\Psi(t)}{dt} = t\,\mathbf{d}^{(k)T}\mathbf{A}\mathbf{d}^{(k)} - \mathbf{d}^{(k)T}\,\mathbf{d}^{(k)} = 0$$

$$\boxed{t^{(k)} = \frac{\mathbf{d}^{(k)T}\,\mathbf{d}^{(k)}}{\mathbf{d}^{(k)T}\mathbf{A}\,\mathbf{d}^{(k)}}}$$

Poznámka:

Pokud by matice $\mathbf{A}$ nesplňovala podmínku symetrie, jaký výsledek by nám dala metoda největšího spádu?

$$\operatorname{grad} F(\mathbf{x}) = \mathbf{0}$$

$$\operatorname{grad} F(\mathbf{x}) = \operatorname{grad}\left(\frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x}\right) = \frac{1}{2}(\mathbf{x}^T\mathbf{A})^T + \frac{1}{2}\mathbf{A}\mathbf{x} - \mathbf{b} = \frac{1}{2}\mathbf{A}^T\mathbf{x} + \frac{1}{2}\mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}$$

$$\Rightarrow \quad \boxed{\frac{1}{2}(\mathbf{A}^T + \mathbf{A})\mathbf{x} = \mathbf{b}}$$

Algoritmus **metody největšího spádu**

1. volba $\mathbf{x}^{(0)}$, ε
2. výpočet směru spádu $\mathbf{d}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}$
3. výpočet koeficientu $t^{(k)} = \dfrac{\mathbf{d}^{(k)T}\,\mathbf{d}^{(k)}}{\mathbf{d}^{(k)T}\mathbf{A}\,\mathbf{d}^{(k)}}$
4. výpočet nové iterace $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\mathbf{d}^{(k)}$
5. $k = k + 1$ a zpět na 2) pokud $||\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}|| > \varepsilon$

Poznámka: Abychom ušetřili operace násobení matice a vektoru, určíme $\mathbf{d}^{(k+1)}$ takto:

$$\mathbf{d}^{(k+1)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k+1)} = \mathbf{b} - \mathbf{A}(\mathbf{x}^{(k)} + t^{(k)}\mathbf{d}^{(k)}) = \mathbf{d}^{(k)} - t^{(k)}\underbrace{\mathbf{A}\mathbf{d}^{(k)}}_{(*)}$$

$(*)$ toto se počítalo v kroku 3 v předchozí iteraci

Věta: Metoda největšího spádu konverguje (pro symetrickou, pozitivně definitní matici $\mathbf{A}$) pro libovolnou volbu počáteční aproximace $\mathbf{x}^{(0)}$ k přesnému řešení soustavy $\mathbf{Ax} = \mathbf{b}$.

Důkaz:

Konvergenci dokážeme v normě $||.||_{\mathbf{A}} = \sqrt{\mathbf{x}^T\mathbf{Ax}}$ (tzv. energetická norma).

$||.||_{\mathbf{A}}$ je s euklidovskou normou $||.||_2$ ekvivalentní, tj. z toho již plyne i konvergence v $||.||_2 = \sqrt{\mathbf{x}^T\mathbf{x}}$

(Definice: X ... lineární prostor, $||.||_1$ a $||.||_2$ ... normy na X;
$||.||_1$ a $||.||_2$ jsou ekvivalentní, existují-li čísla $c, C > 0$: $\forall \mathbf{x} \in X \quad c||\mathbf{x}||_1 \leq ||\mathbf{x}||_2 \leq C||\mathbf{x}||_1$)

tj. má platit

$$c^2\mathbf{x}^T\mathbf{x} \quad \leq \quad \mathbf{x}^T\mathbf{Ax} \quad \leq \quad C^2\mathbf{x}^T\mathbf{x}$$

$$\mathbf{x}^T(c^2\mathbf{I})\mathbf{x} \quad \leq \quad \mathbf{x}^T\mathbf{Ax} \quad \leq \quad \mathbf{x}^T(C^2\mathbf{I})\mathbf{x}$$

platí pro $c = |\lambda_{min}|$, $C = |\lambda_{max}|$

$\mathbf{x}^*$ ... přesné řešení $\mathbf{Ax} = \mathbf{b}$

$\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$ ... chyba k-té iterace

Odvoďme nejprve vztah pro energetickou normu chyby k-té iterace.

$$F(\mathbf{x}^{(k)}) - F(\mathbf{x}^*) = F(\mathbf{x}^* + \mathbf{e}^{(k)}) - F(\mathbf{x}^*) = \ \ldots \qquad (*)$$

Obecně pro 2 body $\mathbf{x}$, $\mathbf{x} + t\mathbf{d}$ platí:

$$\mathbf{F(x + td) - F(x)} = \frac{1}{2}(\mathbf{x} + t\mathbf{d})^T\mathbf{A}(\mathbf{x} + t\mathbf{d}) - \mathbf{b}^T(\mathbf{x} + t\mathbf{d}) - \frac{1}{2}\mathbf{x}^T\mathbf{Ax} + \mathbf{b}^T\mathbf{x} =$$

$$= \underline{t\mathbf{x}^T\mathbf{Ad}} + \frac{1}{2}t^2\mathbf{d}^T\mathbf{Ad} \underline{- t\mathbf{b}^T\mathbf{d}} =$$

$$= \underline{\mathbf{td^T(Ax - b)}} + \frac{\mathbf{1}}{\mathbf{2}}\mathbf{t^2d^TAd}$$

Pro náš případ $\mathbf{x} = \mathbf{x}^*$, $t = 1$, $\mathbf{d} = \mathbf{e}^{(k)}$

$$\ldots = F(\mathbf{x}^* + \mathbf{e}^{(k)}) - F(\mathbf{x}^*) = \frac{1}{2}\mathbf{e}^{(k)^T}\mathbf{Ae}^{(k)} = ||\mathbf{e}^{(k)}||^2_{\mathbf{A}} \qquad (**)$$

$(*) + (**) \quad \Rightarrow \quad \boxed{F(\mathbf{x}^{(k)}) - F(\mathbf{x}^*) = \frac{1}{2}\mathbf{e}^{(k)^T}\mathbf{Ae}^{(k)}} \qquad (***)$

$(***) \quad \Rightarrow \quad F(\mathbf{x}^{(k+1)}) - F(\mathbf{x}^*) = \frac{1}{2}\mathbf{e}^{(k+1)^T}\mathbf{Ae}^{(k+1)} \qquad (****)$

Odečtením $(****)$ a $(***)$ dostaneme

$$\boxed{F(\mathbf{x}^{(k+1)}) - F(\mathbf{x}^{(k)}) = \frac{1}{2}\mathbf{e}^{(k+1)^T}\mathbf{Ae}^{(k+1)} - \frac{1}{2}\mathbf{e}^{(k)^T}\mathbf{Ae}^{(k)}}, \qquad (\clubsuit)$$

kde iterace $\mathbf{x}^{(k+1)}$ je vypočtena metodou největšího spádu, tj.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\mathbf{r}^{(k)}.$$

Pro výraz na levé straně opět použijeme **zvýrazněný vztah** pro hodnoty $\mathbf{x} = \mathbf{x}^{(k)}$, $t = t^{(k)}$, $\mathbf{d} = \mathbf{r}^{(k)}$

$$F(\mathbf{x}^{(k)} + t^{(k)}\mathbf{r}^{(k)}) - F(\mathbf{x}^{(k)}) = t^{(k)}\mathbf{r}^{(k)^T}\big(\underbrace{\mathbf{Ax}^{(k)} - \mathbf{b}}_{-\mathbf{r}^{(k)}}\big) + \frac{1}{2}t^{(k)^2}\mathbf{r}^{(k)^T}\mathbf{Ar}^{(k)} =$$

$\Big($ $t^{(k)}$ jsme počítali podle vztahu $\boxed{t^{(k)} = \frac{\mathbf{r}^{(k)^T}\mathbf{r}^{(k)}}{\mathbf{r}^{(k)^T}\mathbf{Ar}^{(k)}}}$ $\Big)$

$$= -\frac{(\mathbf{r}^{(k)^T}\mathbf{r}^{(k)})^2}{\mathbf{r}^{(k)^T}\mathbf{Ar}^{(k)}} + \frac{1}{2}\frac{(\mathbf{r}^{(k)^T}\mathbf{r}^{(k)})^2}{(\mathbf{r}^{(k)^T}\mathbf{Ar}^{(k)})^2}\mathbf{r}^{(k)^T}\mathbf{Ar}^{(k)} = -\frac{1}{2}\frac{(\mathbf{r}^{(k)^T}\mathbf{r}^{(k)})^2}{\mathbf{r}^{(k)^T}\mathbf{Ar}^{(k)}} \qquad (\spadesuit)$$

Porovnáním $(\spadesuit)$ a $(\clubsuit)$ dostaneme

$$\boxed{-\frac{1}{2}\frac{(\mathbf{r}^{(k)^T}\mathbf{r}^{(k)})^2}{\mathbf{r}^{(k)^T}\mathbf{Ar}^{(k)}} = \frac{1}{2}\mathbf{e}^{(k+1)^T}\mathbf{Ae}^{(k+1)} - \frac{1}{2}\mathbf{e}^{(k)^T}\mathbf{Ae}^{(k)}}. \qquad (\heartsuit)$$

Nyní poslední rovnici

a) vynásobíme 2
b) poslední člen převedeme na druhou stranu
c) a vydělíme jím rovnici

Dostaneme

$$\boxed{\frac{\mathbf{e}^{(k+1)^T}\mathbf{Ae}^{(k+1)}}{\mathbf{e}^{(k)^T}\mathbf{Ae}^{(k)}} = 1 - \frac{(\mathbf{r}^{(k)^T}\mathbf{r}^{(k)})^2}{\mathbf{r}^{(k)^T}\mathbf{Ar}^{(k)}\ \mathbf{e}^{(k)^T}\underbrace{\mathbf{Ae}^{(k)}}_{=\mathbf{r}^{(k)}}}} \qquad (\diamondsuit)$$

Platí $\boxed{\mathbf{Ae}^{(k)} = \mathbf{r}^{(k)}}$, protože $\mathbf{Ax}^* - \mathbf{b} = \mathbf{0}$ a $\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{Ax}^{(k)}$

$$\mathbf{r}^{(k)} = \mathbf{b}\ \underbrace{\underbrace{-\mathbf{Ax}^{(k)} + \mathbf{Ax}^*}_{\mathbf{A}(\underbrace{\mathbf{x}^* - \mathbf{x}^{(k)}}_{\mathbf{e}^{(k)}})} - \mathbf{b}}$$

Dále z $\boxed{\mathbf{Ae}^{(k)} = \mathbf{r}^{(k)}}$ plyne $\mathbf{e}^{(k)} = \mathbf{A}^{-1}\mathbf{r}^{(k)}$ a tedy odhad

$$||\mathbf{e}^{(k)}|| \leq ||\mathbf{A}^{-1}|| \cdot ||\mathbf{r}^{(k)}||$$

Pro $(\diamondsuit)$ dostáváme odhad:

$$\frac{\mathbf{e}^{(k+1)^T}\mathbf{Ae}^{(k+1)}}{\mathbf{e}^{(k)^T}\mathbf{Ae}^{(k)}} \leq 1 - \frac{||\mathbf{r}^{(k)}||^4}{||\mathbf{A}|| \cdot ||\mathbf{r}^{(k)}||^2 \cdot ||\mathbf{A}^{-1}|| \cdot ||\mathbf{r}^{(k)}||^2} = 1 - \frac{1}{||\mathbf{A}|| \cdot ||\mathbf{A}^{-1}||} = \mathbf{q} < 1$$

Tj.

$$\boxed{\mathbf{e}^{(k+1)^T}\mathbf{Ae}^{(k+1)} \leq \mathbf{q}\ \mathbf{e}^{(k)^T}\mathbf{Ae}^{(k)} \quad \forall k}. \qquad (\blacksquare)$$

$$\mathbf{e}^{(k+1)^T}\mathbf{Ae}^{(k+1)} \leq \mathbf{q}\ \mathbf{e}^{(k)^T}\mathbf{Ae}^{(k)} \quad \forall k$$

Důkaz:

Jde o to odhadnout q v (■).

$$||e^{(k+1)}||_A^2 \le q||e^{(k)}||_A^2 \quad \Rightarrow \quad ||e^{(k)}||_A^2 \le q^k||e^{(0)}||_A^2$$

$$q = 1 - \frac{1}{\underbrace{||A||}_{=\lambda_{max}} \cdot \underbrace{||A^{-1}||}_{=\frac{1}{\lambda_{min}}}} = 1 - \frac{1}{\varkappa(A)} = \boxed{\frac{\varkappa(A)-1}{\varkappa(A)} \le \sqrt{\frac{\varkappa(A)-1}{\varkappa(A)+1}}}$$

Důkaz poslední nerovnosti:

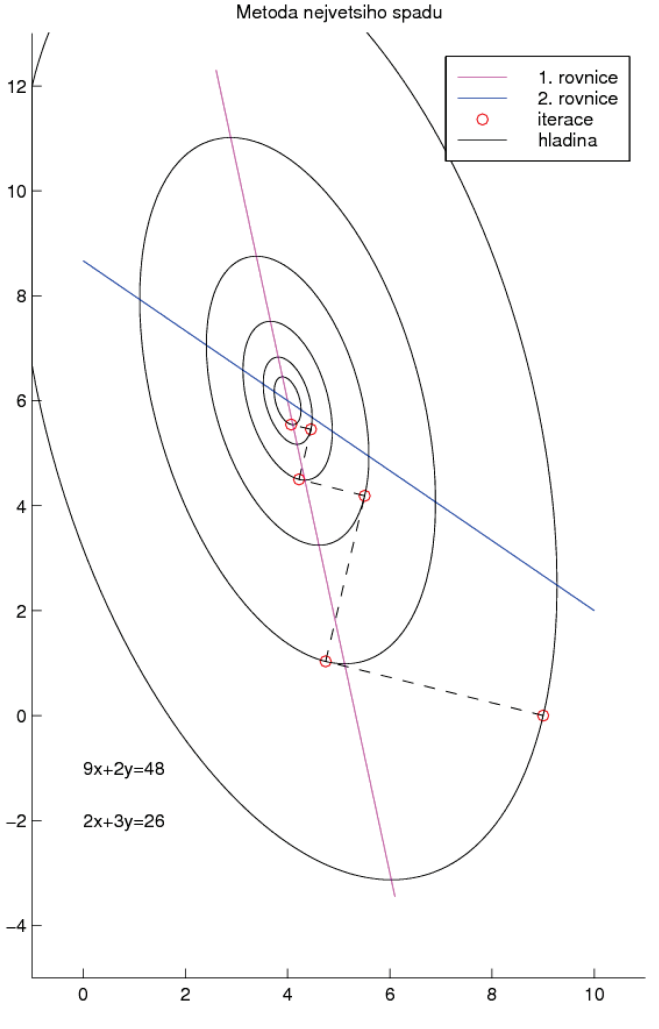
$$\frac{(\varkappa(A)-1)^2}{\varkappa^2(A)} \le \frac{\varkappa(A)-1}{\varkappa(A)+1} \quad / : (\varkappa(A)-1)$$

$$\frac{\varkappa(A)-1}{\varkappa^2(A)} \le \frac{1}{\varkappa(A)+1} \quad / \cdot \varkappa^2(A)(\varkappa(A)+1)$$

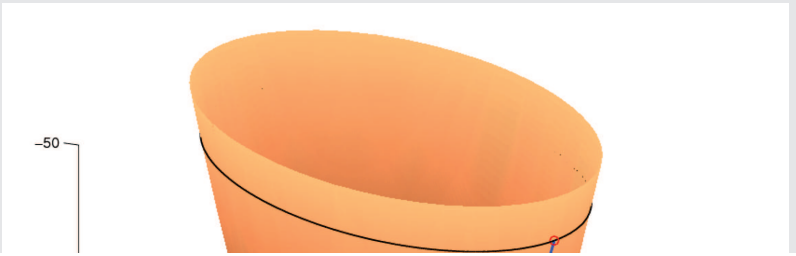
$$\varkappa^2(A) - 1 \le \varkappa^2(A) \quad \dots OK$$

Geometrický význam metody největšího spádu

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



k	$x^{(k)}$	$y^{(k)}$
0	9.0000	0
1	4.7425	1.0321
2	5.5081	4.1902
3	4.2240	4.5015
4	4.4549	5.4541
5	4.0676	5.5480



Vlastnost reziduí

Všimněme si faktu, že vždy po sobě jdoucí iterace směru spádu, tj. $\mathbf{d}^{(k)}$ a $\mathbf{d}^{(k+1)}$ jsou na sebe kolmé.

Cvičení: Ukažte, že platí $\boxed{\mathbf{d}^{(k)T}\mathbf{d}^{(k+1)} = 0}$.

$$\begin{aligned}
\mathbf{d}^{(k)T}\mathbf{d}^{(k+1)} &= \mathbf{d}^{(k)T}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(k+1)}) = \\
&= \mathbf{d}^{(k)T}(\mathbf{b} - \mathbf{A}(\mathbf{x}^{(k)} + t^{(k)}\mathbf{d}^{(k)})) = \\
&= \mathbf{d}^{(k)T}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} - t^{(k)}\mathbf{A}\,\mathbf{d}^{(k)}) = \\
&= \mathbf{d}^{(k)T}(\mathbf{d}^{(k)} - t^{(k)}\mathbf{A}\,\mathbf{d}^{(k)}) = \\
&= \mathbf{d}^{(k)T}\mathbf{d}^{(k)} - t^{(k)}\mathbf{d}^{(k)T}\mathbf{A}\,\mathbf{d}^{(k)} = \\
&= \mathbf{d}^{(k)T}\mathbf{d}^{(k)} - \frac{\mathbf{d}^{(k)T}\mathbf{d}^{(k)}}{\mathbf{d}^{(k)T}\mathbf{A}\,\mathbf{d}^{(k)}}\mathbf{d}^{(k)T}\mathbf{A}\,\mathbf{d}^{(k)} = \\
&= \mathbf{d}^{(k)T}\mathbf{d}^{(k)} - \mathbf{d}^{(k)T}\mathbf{d}^{(k)} = 0
\end{aligned}$$

Poznámka:

V případě, že budou hladiny (elipsy) „velmi protáhlé“, bude obecně metoda největšího spádu konvergovat velmi pomalu, nastane tzv. **cik-cak efekt**.

Na druhou stranu, pokud budou hladiny (elipsy) „skoro kružnice“, bude metoda největšího spádu konvergovat velmi rychle.

Nevýhodu cik-cak efektu odstraní nová metoda, tzv. **metoda sdružených gradientů**, která využívá důmyslnější volby směrů minimalizace, a sice tak, aby se neopakovali, jak k tomu docházelo u metody největšího spádu.

Příklad 1 - pokračování

Uvažovali jsme jednoduchou soustavu $\mathbf{A}\mathbf{x} = \mathbf{b}$, kde

$$\mathbf{A} = \begin{bmatrix} 25 & 0 \\ 0 & 16 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 25 \\ 8 \end{bmatrix}, \quad \mathbf{x}^* = \begin{bmatrix} 1 \\ 0{,}5 \end{bmatrix}.$$

Jedna z vrstevnic měla tvar

$$\boxed{\frac{(x-1)^2}{4^2} + \frac{(y-\frac{1}{2})^2}{5^2} = 1}$$

poměr poloos:

$$\sqrt{\lambda_2} = \sqrt{16} \to 4 : 5 \leftarrow \sqrt{25} = \sqrt{\lambda_1}$$

$$\boxed{\sqrt{\lambda_2} : \sqrt{\lambda_1}}$$

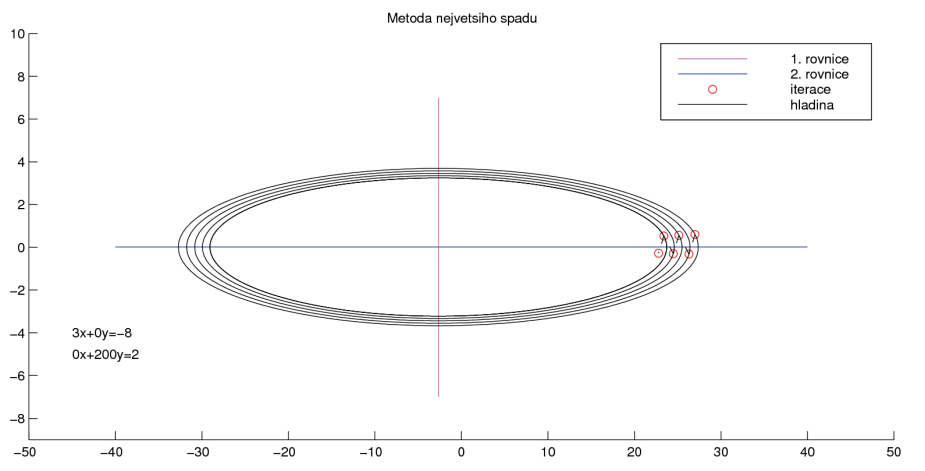
Poznámka:

- Pro případ $\lambda_2 \gg \lambda_1$ získáme protáhlé elipsy
- Pro případ $\lambda_2 \approx \lambda_1$ získáme skoro kružnice

Příklad 2

Pomocí metody největšího spádu řešte soustavu $\mathbf{A}\mathbf{x} = \mathbf{b}$, kde

$$\mathbf{A} = \begin{bmatrix} 3 & 0 \\ 0 & 200 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -8 \\ 2 \end{bmatrix}, \quad \text{počáteční iterace } \mathbf{x}^{(0)} = \begin{bmatrix} 27 \\ 0{,}6 \end{bmatrix}.$$



k	$x^{(k)}$	$y^{(k)}$
0	27.000000	0.600000
1	26.307758	-0.317804
2	25.139940	0.563008
3	24.491101	-0.297251
4	23.396504	0.528335
5	22.788346	-0.277987
⋮	⋮	⋮
250	-2.657606	0.010180

vlastní čísla matice $\mathbf{A}$:

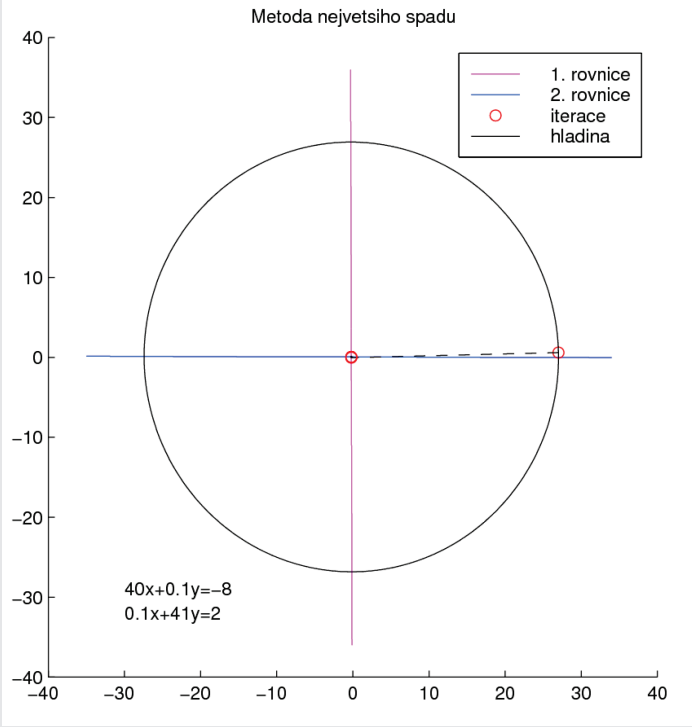
$$\lambda_1 = 3 \quad \text{a} \quad \lambda_2 = 200$$

přesné řešení soustavy je $\mathbf{x}^* = \begin{bmatrix} -\frac{8}{3} \\ \frac{1}{100} \end{bmatrix}$

Příklad 3

Pomocí metody největšího spádu řešte soustavu $\mathbf{A}\mathbf{x} = \mathbf{b}$, kde

$$\mathbf{A} = \begin{bmatrix} 40 & 0{,}1 \\ 0{,}1 & 41 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -8 \\ 2 \end{bmatrix}, \quad \text{počáteční iterace } \mathbf{x}^{(0)} = \begin{bmatrix} 27 \\ 0{,}6 \end{bmatrix}.$$



k	$x^{(k)}$	$y^{(k)}$
0	27.000000	0.600000
1	-0.197972	-0.032418
2	-0.199872	0.049274
3	-0.200123	0.049268

vlastní čísla matice $\mathbf{A}$:

$$\lambda_1 \doteq 39{,}99 \quad \text{a} \quad \lambda_2 \doteq 41{,}01$$

přesné řešení soustavy $\mathbf{x}^*$ se s $\mathbf{x}^{(3)}$ shoduje na 6 desetinných míst

Poznámky k rychlosti konvergence:

$$\boxed{||\mathbf{x}^{(k)} - \mathbf{x}^*||_{\mathbf{A}} = \left(\frac{\varkappa(\mathbf{A})-1}{\varkappa(\mathbf{A})+1}\right)^k ||\mathbf{x}^{(0)} - \mathbf{x}^*||_{\mathbf{A}}}$$

- Je-li $\varkappa(\mathbf{A}) \gg 1$, tj. $\lambda_{max} \gg \lambda_{min}$, pak metoda největšího spádu konverguje pomalu

$$\frac{\varkappa(\mathbf{A})-1}{\varkappa(\mathbf{A})+1} = \frac{\varkappa(A)-1+1-1}{\varkappa(A)+1} = 1 - \underbrace{\frac{2}{\varkappa(A)+1}}_{\to\infty \text{ pro } \varkappa(A)\to\infty} \lessapprox 1$$

- Je-li $\varkappa(\mathbf{A}) \gtrapprox 1$, tj. $\lambda_{max} \approx \lambda_{min}$, pak metoda největšího spádu konverguje rychle

$$\frac{\varkappa(\mathbf{A})-1}{\varkappa(\mathbf{A})+1} = 1 - \underbrace{\frac{2}{\varkappa(\mathbf{A})+1}}_{\approx 2} \approx 0$$

- Pokud jsou vrstevnice sféry (v $\mathbb{R}^2$ kružnice), potom metoda největšího spádu nalezne řešení (přesné) v jednom kroku.

Poznámka:

Směr, ve kterém provádíme minimalizaci v rámci jednoho kroku metody, můžeme volit i jinak než směr největšího spádu.

Obecně označme používané směry $\mathbf{s}^{(k)}$.

Novou iteraci hledáme ve tvaru

$$\boxed{\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t\,\mathbf{s}^{(k)}}$$

Koeficient t získáme z jednorozměrné minimalizace

$$\boxed{\min_{t>0} \underbrace{F(\mathbf{x}^{(k)} + t\,\mathbf{s}^{(k)})}_{\Phi(t)}}$$

$$\Phi(t) = \frac{1}{2}(\mathbf{x}^{(k)} + t\,\mathbf{s}^{(k)})^T\mathbf{A}(\mathbf{x}^{(k)} + t\,\mathbf{s}^{(k)}) - \mathbf{b}^T(\mathbf{x}^{(k)} + t\,\mathbf{s}^{(k)})$$

$$\frac{d\Phi(t)}{dt} = t\mathbf{s}^{(k)T}\mathbf{A}\mathbf{s}^{(k)} + \underbrace{\mathbf{x}^{(k)T}\mathbf{A}\mathbf{s}^{(k)} - \mathbf{b}^T\mathbf{s}^{(k)}}_{\underbrace{(\mathbf{x}^{(k)T}\mathbf{A} - \mathbf{b}^T)}_{\underbrace{(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b})^T}_{-\mathbf{r}^{(k)} \text{ (reziduum)}}}\ \mathbf{s}^{(k)}} = 0$$

$$\boxed{t^{(k)} = \frac{\mathbf{r}^{(k)T}\mathbf{s}^{(k)}}{\mathbf{s}^{(k)T}\mathbf{A}\mathbf{s}^{(k)}}}$$

Volíme-li za vektory $\mathbf{s}^{(k)}$ postupně jednotkové vektory souřadných os, získáme Gauss-Seidelovu metodu !!!

Pokud na vektor $\mathbf{x}$ aplikujeme 1 iteraci gradientní metody se směrovým vektorem $\mathbf{e}_i = [0,\ldots,0,1,0,\ldots 0]^T$ (na i-té pozici 1, jinak 0), dostaneme:

$$\mathbf{x} := \mathbf{x} + \frac{\mathbf{r}^T\mathbf{e}_i}{\mathbf{e}_i^T\mathbf{A}\mathbf{e}_i}\mathbf{e}_i. \qquad (\bullet)$$

Platí: $\mathbf{e}_i^T\mathbf{A}$ ... i-tý řádek matice $\mathbf{A}$

$\mathbf{e}_i^T\mathbf{A}\mathbf{e}_i$ ... diagonální prvek a_{ii} matice $\mathbf{A}$

$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$ ... reziduum

$\mathbf{r}^T\mathbf{e}_i = r_i$ ... i-tá složka vektoru $\mathbf{r}$

$r_i = b_i - \sum_{j=1}^n a_{ij}x_j$

Vztah $(\bullet)$ zvětší i-tou složku vektoru $\mathbf{x}$ o hodnotu r_i, tj.

$$x_i := x_i + \underbrace{\frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{n} a_{ij}x_j\right)}_{=r_i},$$

$$x_i := \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}x_j\right).$$

Script v MATLABu

```
function [vysledky_gs,vysledky_gm]=gs_gm(A,b,x0,iteraci);

%************************************************************
%           Porovnani Gauss-Seidelovy metody a
%         gradientni metody, kde za smery volime
%            jednotkove vektory souradnych os
%************************************************************

n=size(A,1);

%************************************************************
%  Gauss-Seidelova metoda
%************************************************************

x=x0;
vysledky_gs=x';
D=diag(diag(A));  L=tril(A)-D;  U=triu(A)-D;
H=-(L+D)\U;
g=(L+D)\b;

for i=1:iteraci
  x=H*x+g;
  vysledky_gs=[vysledky_gs;x'];
end

%************************************************************
%  Gradientni metoda
%************************************************************

x=x0;
vysledky_gm=x';

for i=1:iteraci
   for j=1:n
     s=zeros(n,1);
     s(j)=1;
     r=-A*x+b;
     t=(r'*s)/(s'*A*s);
     x=x+t*s;
   end;
   vysledky_gm=[vysledky_gm;x'];
end
```


Úvaha

Při vhodné volbě směrových vektorů $\mathbf{s}^{(k)}$ je možné dojít do přesného řešení za konečný počet kroků $\leq n$. Musí existovat n vektorů $\mathbf{s}^{(k)}$ tak, že

$$\boxed{\mathbf{x}^* - \mathbf{x}^{(0)} = \sum_{k=1}^{n} t^{(k)}\mathbf{s}^{(k)}} \qquad (*)$$

Jak volit směry $\mathbf{s}^{(k)}$?

- Zkusíme takto: nechť $\mathbf{s}^{(k)}$ tvoří bázi (ortogonální) n-rozměrného euklidovského prostoru, potom vynásobením $(*)$ skalárně s $\mathbf{s}^{(k)}$ a úpravou získáme

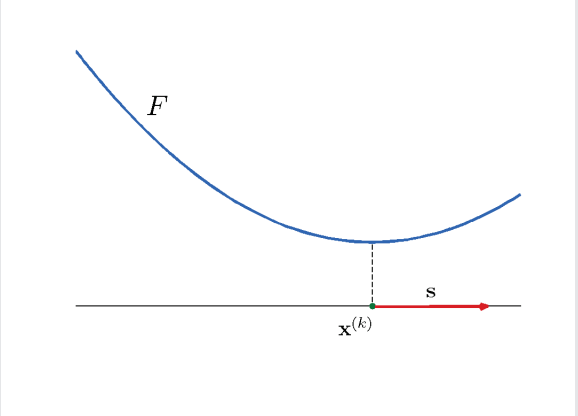
$$\mathbf{s}^{(k)T}(\mathbf{x}^* - \mathbf{x}^{(0)}) = t^{(k)}\mathbf{s}^{(k)T}\mathbf{s}^{(k)}$$

$$t^{(k)} = \frac{\mathbf{s}^{(k)T}(\mathbf{x}^* - \mathbf{x}^{(0)})}{\mathbf{s}^{(k)T}\mathbf{s}^{(k)}} \qquad \ldots \text{ nešikovné !, obsahuje přesné řešení}$$

- je třeba zvolit lepší strategii volby vektorů $\mathbf{s}^{(k)}$

Definice $\mathbf{x}^{(k)}$ je **optimální vzhledem ke směru** $\mathbf{s} \neq \mathbf{0}$, jestliže

$$\boxed{F(\mathbf{x}^{(k)}) \leq F(\mathbf{x}^{(k)} + t\mathbf{s}) \quad \forall t \in \mathbb{R}} \qquad (\star)$$



Poznámka: Je-li $\mathbf{x}^{(k)}$ optimální vzhledem k libovolnému směru z vektorového prostoru V, říkáme, že je $\mathbf{x}^{(k)}$ **optimální vzhledem k** V.

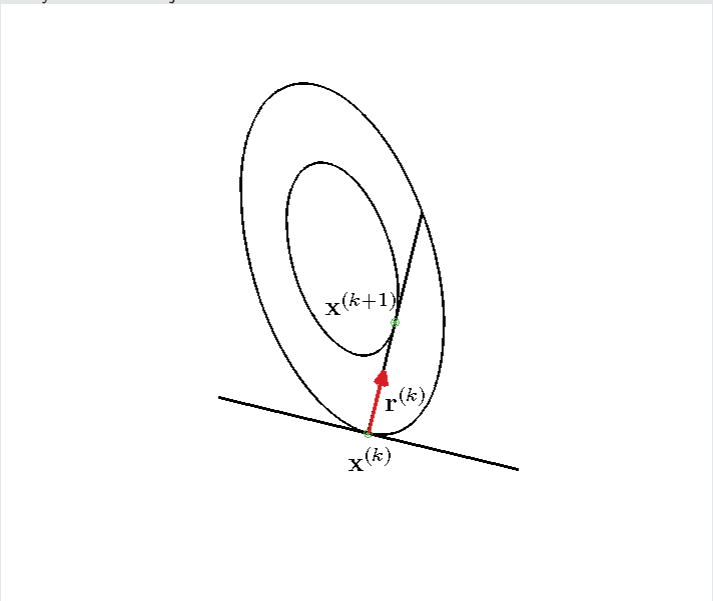
Podle $(\star)$ se minima nabývá pro $t = 0$, tzn. že derivace F podle t je v minimu $(t = 0)$ rovna 0:

$$\frac{\partial F}{\partial t}\left(\mathbf{x}^{(k)} + t\,\mathbf{s}\right) = t\,\mathbf{s}^T\mathbf{A}\mathbf{s} + \mathbf{s}^T\left(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\right)$$

$$\frac{\partial F}{\partial t}\left(\mathbf{x}^{(k)}\right) = \mathbf{s}^T \underbrace{\left(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\right)}_{=\mathbf{r}^{(k)}} = 0$$

$$\boxed{\mathbf{s} \perp \mathbf{r}^{(k)}}$$

Poznámka: Iterace $\mathbf{x}^{(k+1)}$ metody největšího spádu je optimální vzhledem k reziduu $\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}$ ... směry, ve kterých minimalizujeme.



Naším cílem je, aby se i v dalších iteracích zachovávala optimalita k již použitým směrům.

To pro metodu největšího spádu bohužel neplatí.

Např. pro soustavu ve $2D$ jsme ukazovali, že směry největšího spádu (reziduí) jsou na sebe kolmé,

tj. $\mathbf{r}^{(k)} \perp \mathbf{r}^{(k+1)}$ a $\mathbf{r}^{(k+1)} \perp \mathbf{r}^{(k+2)}$ $\Rightarrow$ $\boxed{\mathbf{r}^{(k)} \parallel \mathbf{r}^{(k+2)}}$!!!

$\Rightarrow$ $\mathbf{x}^{(k+2)}$ je optimální vzhledem k $\mathbf{r}^{(k+1)}$, ale již není optimální vzhledem k $\mathbf{r}^{(k)}$

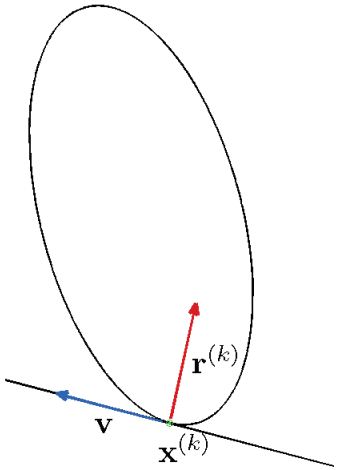
Existují směry, které udržují optimalitu k předchozím?

Nechť

$$\boxed{\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}}.$$

Předpokládejme, že $\mathbf{x}^{(k)}$ ke optimální vzhledem k $\mathbf{v}$ (tj. $\mathbf{r}^{(k)} \perp \mathbf{v}$).

Numerické metody
Josef Daněk
KMA/NM
13.2.2013



Chceme-li, aby bylo i $\mathbf{x}^{(k+1)}$ optimální vzhledem k $\mathbf{v}$, (tj. $\mathbf{r}^{(k+1)} \perp \mathbf{v}$), musí platit:

$$0 = \mathbf{v}^T\mathbf{r}^{(k+1)} = \mathbf{v}^T(\mathbf{b}-\mathbf{A}\mathbf{x}^{(k+1)}) = \mathbf{v}^T\left(\mathbf{b} - \mathbf{A}(\mathbf{x}^{(k)} + \mathbf{s})\right) = \mathbf{v}^T\left(\underbrace{\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}}_{\mathbf{r}^{(k)}} - \mathbf{A}\mathbf{s}\right) = \mathbf{v}^T\left(\mathbf{r}^{(k)} - \mathbf{A}\mathbf{s}\right) = -\mathbf{v}^T\mathbf{A}\mathbf{s}$$

Závěr

Chceme-li zachovat optimalitu vzhledem ke všem použitým směrům, musí tyto směry splňovat podmínky tzv. A-**ortogonality**, tj. pro 2 různé směry $\mathbf{s}$ a $\mathbf{v}$ musí platit:

$$\boxed{\mathbf{v}^T\mathbf{A}\mathbf{s} = 0}.$$

Poznámka: Vektorům které jsou A-**ortogonální** se také říká A-**sdružené**.

Metoda sdružených gradientů

Za směry, ve kterých minimalizujeme, budeme brát A-ortogonální vektory $\mathbf{s}^{(k)}$.

Platí tedy:

$$\boxed{{\mathbf{s}^{(k)}}^T \mathbf{A}\, \mathbf{s}^{(l)} = 0, \quad k \neq l}.$$

Chceme, aby platilo:

$${\mathbf{s}^{(k)}}^T\mathbf{A} \cdot / \qquad \boxed{\mathbf{x}^* - \mathbf{x}^{(0)} = \sum_{k=1}^{n} t^{(k)}\mathbf{s}^{(k)}} \qquad (*)$$

$$\mathbf{s}^{(k)T}\underbrace{\mathbf{A}(\mathbf{x}^* - \mathbf{x}^{(0)})}_{\mathbf{A}\mathbf{x}^* - \mathbf{A}\mathbf{x}^{(0)} = \underbrace{\mathbf{A}\mathbf{x}^* - \mathbf{b}}_{=\mathbf{0}} \underbrace{- \mathbf{A}\mathbf{x}^{(0)} + \mathbf{b}}_{\mathbf{r}^{(0)}}} = t^{(k)}\,\mathbf{s}^{(k)T}\mathbf{A}\,\mathbf{s}^{(k)}$$

$$\boxed{t^{(k)} = \frac{\mathbf{s}^{(k)T}\mathbf{r}^{(0)}}{\mathbf{s}^{(k)T}\mathbf{A}\mathbf{s}^{(k)}}}$$

Strategie volby směrů

- Máme-li ortogonální bázi $\mathbb{R}^n$, lze z ní procesem A-ortogonalizace získat A-ortogonální bázi.
- Za ortogonální bázi budeme volit reziduové vektory.

  Aby proces ortogonalizace vedl k cíli, musíme zaručit, že reziduové vektory tvoří bázi.

  Ortogonalitu ukážeme vzápětí; může se stát, že se některé reziduum anuluje.

  Potom ovšem iterační proces končí - dosáhli jsme přesného řešení.
- Provádíme tedy současně 2 procesy!
  - iterační proces
  - proces A-ortogonalizace
- Vektory reziduí budeme značit $\mathbf{r}^{(k)}$, získané sdružené směry označíme $\mathbf{s}^{(k)}$
  - pro zadané $\mathbf{x}^{(0)}$ určíme $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$
  - $\mathbf{s}^{(0)}$ položíme rovno $\mathbf{r}^{(0)}$
  - určíme $\mathbf{x}^{(1)}$ optimální vzhledem k $\mathbf{s}^{(0)}$
  - určíme $\mathbf{r}^{(1)}$
  - $\mathbf{s}^{(1)}$ určujeme z $\mathbf{r}^{(1)}$ tak, aby $\mathbf{s}^{(1)T}\mathbf{A}\mathbf{s}^{(0)} = 0$
  - atd.

Proces A-ortogonalizace

$$\boxed{\mathbf{s}^{(k)} = \mathbf{r}^{(k)} + \sum_{i=1}^{k-1} \beta_{ki}\,\mathbf{s}^{(i)}} \qquad (\bullet\bullet)$$

(Při určení $\mathbf{s}^{(k)}$ vyjdeme z $\mathbf{r}^{(k)}$. Přičítáme násobky předchozích $\mathbf{s}^{(i)}$ tak, abychom zaručili A-ortogonalitu.)

Koeficienty β_{ki} volíme tak, aby

$$\boxed{\mathbf{s}^{(k)}\mathbf{A}\mathbf{s}^{(i)} = 0, \quad (i < k)}$$

$(\bullet\bullet)$ vynásobíme $\mathbf{s}^{(i)T}\mathbf{A}\cdot/$

$$\underbrace{\mathbf{s}^{(i)T}\mathbf{A}\mathbf{s}^{(k)}}_{=0} = \mathbf{s}^{(i)T}\mathbf{A}\mathbf{r}^{(k)} + \beta_{ki}\,\mathbf{s}^{(i)T}\mathbf{A}\mathbf{s}^{(i)}$$

$$\Rightarrow \quad \boxed{\beta_{ki} = -\frac{\mathbf{s}^{(i)T}\mathbf{A}\mathbf{r}^{(k)}}{\mathbf{s}^{(i)T}\mathbf{A}\mathbf{s}^{(i)}}}$$

Numerické metody KMA/NM
Josef Daněk 13.2.2o13

Z vlastností A-ortogonality vyplývá řada skutečností.

Věta 1 Platí

$$\boxed{\begin{aligned} \mathbf{r}^{(k)T}\cdot\mathbf{s}^{(j)} &= \mathbf{r}^{(0)T}\cdot\mathbf{s}^{(j)} && k \le j \\ \mathbf{r}^{(k)T}\cdot\mathbf{s}^{(j)} &= 0 && k > j \end{aligned}}$$

Důkaz:

$$\begin{aligned} -\mathbf{b}\cdot/ \quad \mathbf{A}\cdot/ \qquad & \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\mathbf{s}^{(k)} \\ \Rightarrow \qquad & -\mathbf{r}^{(k+1)} = -\mathbf{r}^{(k)} + t^{(k)}\mathbf{A}\,\mathbf{s}^{(k)} \\ \Rightarrow \qquad & \mathbf{r}^{(k)} = \mathbf{r}^{(0)} - \sum_{j=1}^{k-1} t^{(j)}\mathbf{A}\,\mathbf{s}^{(j)} \end{aligned}$$

vynásobíme skalárně s $\mathbf{s}^{(j)}$

$$\boxed{\mathbf{r}^{(k)T}\mathbf{s}^{(j)} = \mathbf{r}^{(0)T}\mathbf{s}^{(j)} - \underbrace{t^{(j)}}_{(*)}\,\mathbf{s}^{(j)T}\mathbf{A}\,\mathbf{s}^{(j)}}$$

$(*)$ počítáme (viz dříve) takto

$$t^{(j)} = \frac{\mathbf{r}^{(0)T}\mathbf{s}^{(j)}}{\mathbf{s}^{(j)T}\mathbf{A}\mathbf{s}^{(j)}}$$

□

Důkaz:

vztah $(\bullet\bullet)$ vynásobíme skalárně s $\mathbf{A}\mathbf{s}^{(j)}$

$$\mathbf{s}^{(k)} = \mathbf{r}^{(k)} + \sum_{i=1}^{k-1} \beta_{ki}\,\mathbf{s}^{(i)} \qquad (\bullet\bullet)$$

$$\underbrace{\mathbf{s}^{(j)T}\mathbf{A}\,\mathbf{s}^{(k)}}_{\begin{array}{l}=0\ (k<j)\\ \neq 0\ (k=j)\end{array}} = \mathbf{s}^{(j)T}\mathbf{A}\,\mathbf{r}^{(k)} + \sum_{i=1}^{k-1}\beta_{ki}\ \underbrace{\mathbf{s}^{(j)T}\mathbf{A}\,\mathbf{s}^{(i)}}_{=0\ (k\le j)}$$

□

Důkaz:

Úplnou matematickou indukcí ukážeme, že $\boxed{\mathbf{r}^{(j)T}\mathbf{r}^{(k)} = 0 \quad \text{pro} \quad j > k}$.

Platí

$$\mathbf{r}^{(k+1)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k+1)} = \mathbf{b} - \mathbf{A}(\mathbf{x}^{(k)} + t^{(k)}\mathbf{r}^{(k)}) = \mathbf{r}^{(k)} - t^{(k)}\mathbf{A}\mathbf{r}^{(k)}$$

1. $\underline{\underline{j = 1}} \Rightarrow \ k = 0$

$$\mathbf{r}^{(1)T}\mathbf{r}^{(0)} = \left(\mathbf{r}^{(0)} + t^{(0)}\mathbf{A}\,\mathbf{s}^{(0)}\right)^T\mathbf{r}^{(0)} = \mathbf{r}^{(0)T}\mathbf{r}^{(0)} + t^{(0)}\mathbf{s}^{(0)T}\mathbf{A}\,\mathbf{r}^{(0)} = \mathbf{r}^{(0)T}\mathbf{r}^{(0)} - \frac{\mathbf{r}^{(0)T}\mathbf{s}^{(0)}}{\mathbf{s}^{(0)T}\mathbf{A}\,\mathbf{s}^{(0)}}\,\mathbf{s}^{(0)T}\mathbf{A}\,\mathbf{r}^{(0)} = 0$$

$(\mathbf{r}^{(0)} = \mathbf{s}^{(0)})$

2. a) $\underline{\forall k < j}$ platí: $\boxed{\mathbf{r}^{(j+1)^T}\mathbf{r}^{(k)} = 0}$

$$\mathbf{r}^{(j+1)^T}\mathbf{r}^{(k)} = \left(\mathbf{r}^{(j)} + t^{(j)}\mathbf{A}\,\mathbf{s}^{(j)}\right)^T \mathbf{r}^{(k)} = \underbrace{\mathbf{r}^{(j)^T}\mathbf{r}^{(k)}}_{=0 \text{ (předpoklad)}} + t^{(j)}\underbrace{\mathbf{s}^{(j)^T}\mathbf{A}\,\mathbf{r}^{(k)}}_{=0 \text{ (Věta 2)}}$$

b) $\boxed{\mathbf{r}^{(j+1)^T}\mathbf{r}^{(j)} = 0}$

$$\mathbf{r}^{(j+1)^T}\mathbf{r}^{(j)} = \left(\mathbf{r}^{(j)} + t^{(j)}\mathbf{A}\,\mathbf{s}^{(j)}\right)^T \mathbf{r}^{(j)} = \mathbf{r}^{(j)^T}\mathbf{r}^{(j)} + t^{(j)}\mathbf{s}^{(j)^T}\mathbf{A}\,\mathbf{r}^{(j)} = \underbrace{\mathbf{r}^{(j)^T}\mathbf{r}^{(j)} - \frac{\mathbf{r}^{(0)^T}\mathbf{s}^{(j)}}{\mathbf{s}^{(j)^T}\mathbf{A}\,\mathbf{s}^{(j)}}\,\mathbf{s}^{(j)^T}\mathbf{A}\,\mathbf{r}^{(j)}}$$

$\mathbf{s}^{(j)^T}\mathbf{A}\,\mathbf{s}^{(j)} = \mathbf{s}^{(j)^T}\mathbf{A}\,\mathbf{r}^{(j)}$ **(Věta 2**

$\mathbf{r}^{(j)^T}\mathbf{r}^{(j)} = \mathbf{r}^{(0)^T}\mathbf{s}^{(j)}$ **(Věta 1**

□

Důkaz: Platí:

$$\beta_{ki} = -\frac{\mathbf{s}^{(i)^T}\mathbf{A}\mathbf{r}^{(k)}}{\mathbf{s}^{(i)^T}\mathbf{A}\,\mathbf{s}^{(i)}}$$

Pro čitatel platí:

$$\mathbf{s}^{(i)^T}\mathbf{A}\,\mathbf{r}^{(k)} = \mathbf{r}^{(k)^T}\mathbf{A}\,\mathbf{s}^{(i)} = \mathbf{r}^{(k)^T}\frac{\mathbf{1}}{\mathbf{t}^{(i)}}\left(\mathbf{r}^{(i+1)} - \mathbf{r}^{(i)}\right),$$

kde se použil vztah

$$\mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} + \mathbf{t}^{(i)}\mathbf{A}\,\mathbf{s}^{(i)}, \quad t^{(i)} \neq 0 \text{ pro } \mathbf{r}^{(i)} \neq \mathbf{0}$$

Platí

- $\underline{\text{pro } i < k-1}$: čitatel $\beta_{ki} = \mathbf{r}^{(k)^T}\left(\mathbf{r}^{(i+1)} - \mathbf{r}^{(i)}\right)\frac{1}{t^{(i)}} = 0$ **(Tvrzení)**
- $\underline{\text{pro } i = k-1}$: čitatel $\beta_{k,k-1} = \mathbf{r}^{(k)^T}\left(\mathbf{r}^{(k)} - \mathbf{r}^{(k-1)}\right)\frac{1}{t^{(k-1)}} = \mathbf{r}^{(k)^T}\mathbf{r}^{(k)}\frac{1}{t^{(k-1)}} \neq 0$ (pro $\mathbf{r}^{(k)} \neq \mathbf{0}$)

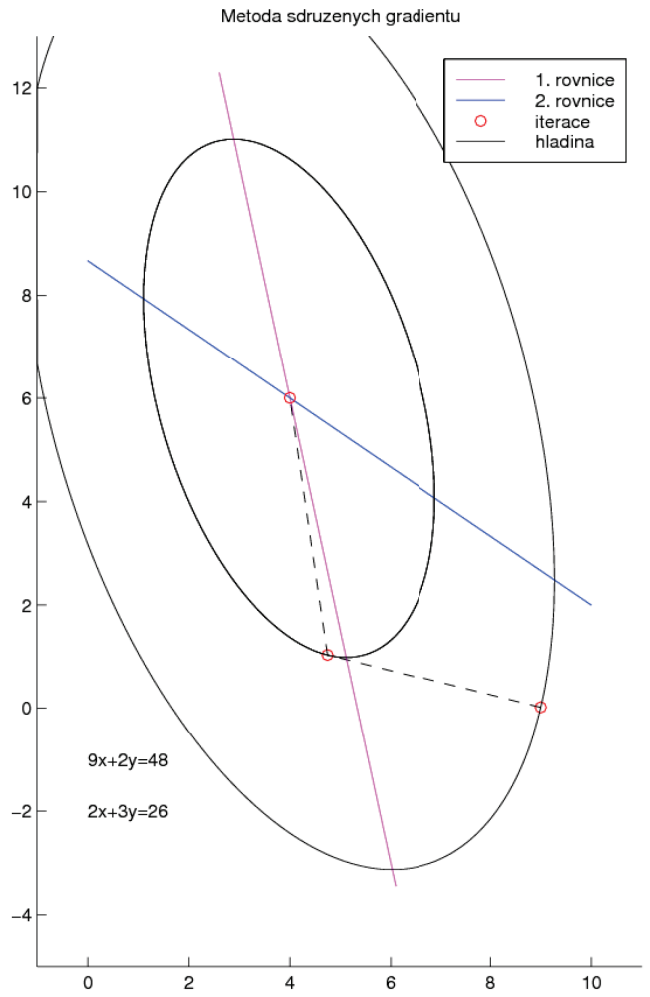
□

Algoritmus (**Metoda sdružených gradientů**)

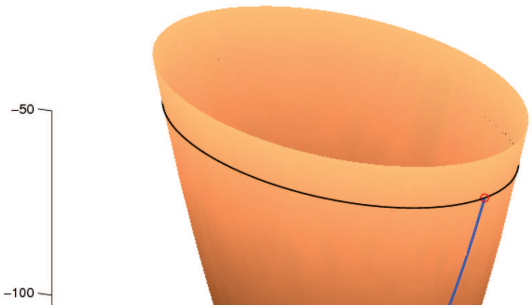
1. $\mathbf{x}^{(0)}$, ε
2. $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\,\mathbf{x}^{(0)}$, $\mathbf{s}^{(0)} = \mathbf{r}^{(0)}$
3. $t^{(k)} = \dfrac{\mathbf{s}^{(k)^T}\mathbf{r}^{(k)}}{\mathbf{s}^{(k)^T}\mathbf{A}\,\mathbf{s}^{(k)}}$
4. $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\mathbf{s}^{(k)}$
5. $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} + t^{(k)}\mathbf{A}\,\mathbf{s}^{(k)}$
6. $\beta_k = -\dfrac{\mathbf{s}^{(k)^T}\mathbf{A}\,\mathbf{r}^{(k+1)}}{\mathbf{s}^{(k)^T}\mathbf{A}\,\mathbf{s}^{(k)}}$
7. $\mathbf{s}^{(k+1)} = \mathbf{r}^{(k+1)} + \beta_k\mathbf{s}^{(k)}$
8. If $||\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}|| < \varepsilon$ then konec, else $\rightarrow$ add 3

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Geometrický význam metody sdružených gradientů



k	$x^{(k)}$	$y^{(k)}$
0	9.0000	0
1	4.7425	1.0321
2	4.0000	6.0000



Poznámka: Gradientní metody patří mezi **nestacionární metody**.

např. pro metodu největšího spádu platí

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\mathbf{d}^{(k)} = \mathbf{x}^{(k)} + t^{(k)}\left(\mathbf{b} - \mathbf{A}\,\mathbf{x}^{(k)}\right) = \underbrace{\left(\mathbf{I} - t^{(k)}\mathbf{A}\right)}_{\mathbf{H}^{(k)}}\mathbf{x}^{(k)} + \underbrace{t^{(k)}\mathbf{b}}_{\mathbf{g}^{(k)}}$$

V každém kroku se mění matice $\mathbf{H}^{(k)}$.

Platí-li $||\mathbf{H}^{(k)}|| \to 0$ (pro $k \to \infty$), dostaneme metody se superlineární rychlostí konvergence.

Věta Nechť $\mathbf{A}$ je symetrická pozitivně definitní. Potom metoda sdružených gradientů konverguje nejvýše po n krocích. Navíc chyba k-té iterace $(k < n)$ je ortogonální na směry $\mathbf{s}^{(j)}$, $j = 0, 1, \ldots, k-1$ a platí:

$$\boxed{||\mathbf{x}^{(k)} - \mathbf{x}^*||_{\mathbf{A}} \leq \frac{2C^k}{1 + C^{2k}}||\mathbf{x}^{(0)} - \mathbf{x}^*||_{\mathbf{A}}},$$

kde $C = \dfrac{\sqrt{\varkappa(\mathbf{A})} - 1}{\sqrt{\varkappa(\mathbf{A})} + 1}$, $\varkappa(\mathbf{A}) = \dfrac{\lambda_{max}}{\lambda_{min}}$.

Poznámka: V metodě největšího spádu vystupuje ve vztahu pro chybu k-té iterace koeficient

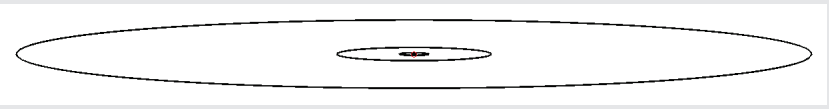
$$\left(\frac{\varkappa(\mathbf{A}) - 1}{\varkappa(\mathbf{A}) + 1}\right)^k.$$

Je zřejmé, že na rychlost konvergence má vliv číslo $\varkappa(\mathbf{A})$, tj. λ_{max} a λ_{min}. Čím blíže je λ_{max} a λ_{min}, tím rychleji metody konvergují.

Příklad 4

Řešte soustavu $\boxed{\mathbf{A}\mathbf{x} = \mathbf{b}}$, kde

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 10000 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 10000 \end{bmatrix}, \quad \text{přesné řešení } \mathbf{x}^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$



- poměr poloos elips je $\sqrt{10000} : \sqrt{1} = 100 : 1$!!!
- $\varkappa(\mathbf{A}) = \frac{10000}{1} = 10000 \quad \Rightarrow$ pomalá konvergence!

Vezměme si matici $\mathbf{P}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{10000} \end{bmatrix}$ $(\det(\mathbf{P}^{-1}) \neq 0)$ a řešme soustavu

$$\boxed{\mathbf{P}^{-1}\mathbf{A}\mathbf{x} = \mathbf{P}^{-1}\mathbf{b}}$$

$$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}1\\ 1\end{bmatrix}$$

- $\varkappa(\mathbf{P}^{-1}\mathbf{A})=\frac{1}{1}=1 \quad\Rightarrow\quad$ rychlá konvergence (1. iterace). Mluvíme o tzv. **předpodmiňování**.

Poznámka:

Chceme-li i novou (předpodmíněnou) soustavu řešit metodou sdružených gradientů, musí být její matice symetrická pozitivně definitní.

Místo matice $\mathbf{P}^{-1}\mathbf{A}$ vezmeme matici (podobnou $\mathbf{A}$) $\mathbf{P}^{-\frac{1}{2}}\mathbf{A}\mathbf{P}^{-\frac{1}{2}}$ ($\mathbf{P}$ ... symetrická pozitivně definitní) a řešíme soustavu

$$\boxed{\widetilde{\mathbf{A}}\tilde{\mathbf{x}}=\tilde{\mathbf{b}}}$$
$$\widetilde{\mathbf{A}}=\mathbf{P}^{-\frac{1}{2}}\mathbf{A}\mathbf{P}^{-\frac{1}{2}},\quad \tilde{\mathbf{x}}=\mathbf{P}^{\frac{1}{2}}\mathbf{x},\quad \tilde{\mathbf{b}}=\mathbf{P}^{-\frac{1}{2}}\mathbf{b}$$

Jak volit matice předpodmínění $\mathbf{P}$?

... řada možností, např. $\mathbf{P}=\operatorname{diag}(\mathbf{A})$

Příklad 5

Porovnejte vlastní čísla zadané matice $\mathbf{A}$ a matice získané pomocí diagonálního předpodmínění.

$$\mathbf{A}=\begin{bmatrix}1000000 & 200 & 30 & 0\\ 200 & 10000 & 40 & 0\\ 30 & 40 & 100 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$\mathbf{P}=\begin{bmatrix}1000000 & 0 & 0 & 0\\ 0 & 10000 & 0 & 0\\ 0 & 0 & 100 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$\mathbf{P}^{-\frac{1}{2}}=\begin{bmatrix}\frac{1}{\sqrt{1000000}} & 0 & 0 & 0\\ 0 & \frac{1}{\sqrt{10000}} & 0 & 0\\ 0 & 0 & \frac{1}{\sqrt{100}} & 0\\ 0 & 0 & 0 & \frac{1}{\sqrt{1}}\end{bmatrix}=\begin{bmatrix}\frac{1}{1000} & 0 & 0 & 0\\ 0 & \frac{1}{100} & 0 & 0\\ 0 & 0 & \frac{1}{10} & 0\\ 0 & 0 & 0 & \frac{1}{1}\end{bmatrix}$$

$$\widetilde{\mathbf{A}}=\mathbf{P}^{-\frac{1}{2}}\mathbf{A}\mathbf{P}^{-\frac{1}{2}}=\begin{bmatrix}\frac{1}{1000} & 0 & 0 & 0\\ 0 & \frac{1}{100} & 0 & 0\\ 0 & 0 & \frac{1}{10} & 0\\ 0 & 0 & 0 & \frac{1}{1}\end{bmatrix}\begin{bmatrix}1000000 & 200 & 30 & 0\\ 200 & 10000 & 40 & 0\\ 30 & 40 & 100 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\frac{1}{1000} & 0 & 0 & 0\\ 0 & \frac{1}{100} & 0 & 0\\ 0 & 0 & \frac{1}{10} & 0\\ 0 & 0 & 0 & \frac{1}{1}\end{bmatrix}=$$
$$=\begin{bmatrix}1 & 0.002 & 0.003 & 0\\ 0.002 & 1 & 0.04 & 0\\ 0.003 & 0.04 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

---

vlastní čísla matice $\mathbf{A}$:

Numerické metody KMA/NM
Josef Daněk 13.2.2o13

$\lambda_1=1;\quad \lambda_2\doteq 99{,}837534233;\quad \lambda_3\doteq 10000{,}121161153;\quad \lambda_4\doteq 1000000{,}041304614$

$$\varkappa(\mathbf{A})\doteq\frac{1000000{,}041304614}{1}=1000000{,}041304614$$

vlastní čísla matice $\widetilde{\mathbf{A}}=\mathbf{P}^{-\frac{1}{2}}\mathbf{A}\mathbf{P}^{-\frac{1}{2}}$:

$\tilde{\lambda}_1\doteq 0{,}959987454937999;\quad \tilde{\lambda}_2\doteq 0{,}999702401514713;\quad \tilde{\lambda}_3=1;\quad \tilde{\lambda}_4\doteq 1{,}040310143547287$

$$\varkappa(\widetilde{\mathbf{A}})\doteq\frac{1{,}040310143547287}{0{,}959987454937999}\doteq 1{,}083670560689229$$

Kapitola 6. Vlastní čísla a vlastní vektory

Výpočet vlastních čísel a vlastních vektorů

S pojmem **vlastního čísla** jsme se již setkali například u iteračních metod pro řešení soustav lineárních algebraických rovnic. Velikosti vlastních čísel iterační matice rozhodovaly o konvergenci příslušné iterační metody. S úlohou na vlastní čísla se setkáme i v aplikacích při řešení řady technických a fyzikálních problémů.

Definice: Je dána čtvercová matice $\mathbf{A}$ řádu n. Číslo λ, pro které má soustava

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v} \quad \text{resp.} \quad (\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$$

nenulové řešení, se nazývá **vlastní číslo** matice $\mathbf{A}$, jemu odpovídající nenulové řešení $\mathbf{v}$ **vlastní vektor** matice $\mathbf{A}$.

Homogenní soustava má nenulové řešení $\Leftrightarrow$ matice soustavy je singulární, tj. její determinant je nulový.

Vlastní čísla $\lambda_1, \lambda_2, \ldots, \lambda_n$ jsou kořeny **charakteristické rovnice**

$$p_{\mathbf{A}}(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I}) = 0.$$

Ke každému vlastnímu číslu λ_i existuje alespoň jeden vlastní vektor $\mathbf{v}_i$.

Poznámka: Charakteristický polynom je stupně n $\Rightarrow$ $\exists\, n$ vlastních čísel.

Definice: Matici $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ nazýváme **spektrální maticí** matice $\mathbf{A}$.

Úlohy na nalezení vlastních čísel rozdělíme do dvou skupin:

- **Úplný problém** – úloha najít všechna vlastní čísla
- **Částečný problém** – úloha najít pouze některá vl. čísla (obvykle s největší absolutní hodnotou)

Úlohu na vlastní čísla si připomeneme na příkladu.

Příklad 1

Stanovte taková čísla λ, pro která má homogenní soustava $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ nenulové řešení, dále určete toto řešení, pro matici

$$\mathbf{A} = \begin{bmatrix} 2 & 0 & 0 \\ 2 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}.$$

Řešíme tedy soustavu

$$(\mathbf{A} - \lambda\mathbf{I})\,\mathbf{v} = \begin{bmatrix} 2-\lambda & 0 & 0 \\ 2 & 2-\lambda & 1 \\ 1 & 1 & 2-\lambda \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Aby homogenní soustava měla nenulové řešení, musí být determinant soustavy nulový. Hledáme proto taková λ, aby

$$\det(\mathbf{A} - \lambda\mathbf{I}) = (2-\lambda)^3 - (2-\lambda) = (2-\lambda)\left[(2-\lambda)^2 - 1\right] = (2-\lambda)(1-\lambda)(3-\lambda) = 0.$$

Dostali jsme algebraickou rovnici stupně 3 a pouze pro její kořeny

$$\lambda_1 = 3, \qquad \lambda_2 = 2, \qquad \lambda_3 = 1$$

bude mít uvažovaná soustava nenulové řešení.

Ke každému vlastnímu číslu λ_i můžeme najít nenulové řešení homogenní soustavy

$$(\mathbf{A} - \lambda_i\mathbf{I})\,\mathbf{v} = \mathbf{0}.$$

Např. pro $\lambda_1 = 3$ řešíme soustavu

$$\begin{bmatrix} -1 & 0 & 0 \\ 2 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Matice soustavy je samozřejmě singulární a proto bude existovat celý systém řešení v závislosti na parametru $r \in \mathbb{R}$. Každý vektor $[0, r, r]^T$ řeší danou soustavu. Ze systému vybereme jednoho zástupce, např. $\mathbf{v}^{(1)} = [0, 1, 1]^T$, a říkáme, že $\mathbf{v}^{(1)}$ je **vlastní vektor** odpovídající vlastnímu číslu λ_1. Podobně bychom nalezli vlastní vektory odpovídající vlastním číslům λ_2 a λ_3.

Poznámka:

Vlastní čísla (horní) trojúhelníkové matice jsou rovna jejím diagonálním prvkům, neboť charakteristický polynom má tvar:

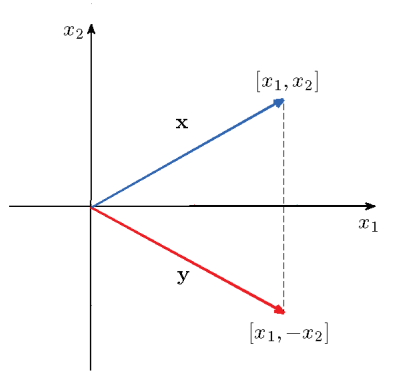
$$p_{\mathbf{A}}(\lambda) = (a_{11} - \lambda)(a_{22} - \lambda)\ldots(a_{nn} - \lambda).$$

Motivace

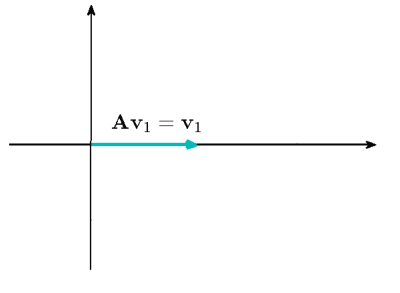
Vlastní vektor je takový vektor, pro který platí, že vynásobíme-li matici $\mathbf{A}$ s tímto vektorem, získáme násobek původního vektoru. Mluvíme o **samodružných prvcích**.

Příklad: Osová souměrnost = zobrazení $\mathbf{y} = \mathbf{A}\mathbf{x}$.

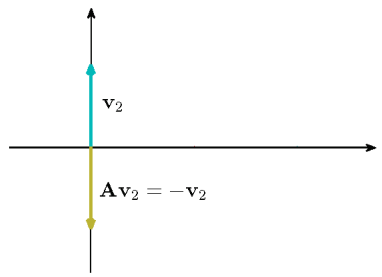
$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



$$\underbrace{\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}}_{\mathbf{A}}\underbrace{\begin{bmatrix}1\\ 0\end{bmatrix}}_{\mathbf{v}_1} = \underbrace{1}_{\lambda_1}\underbrace{\begin{bmatrix}1\\ 0\end{bmatrix}}_{\mathbf{v}_1}$$



$$\underbrace{\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}}_{\mathbf{A}}\underbrace{\begin{bmatrix}0\\ 1\end{bmatrix}}_{\mathbf{v}_2} = \underbrace{-1}_{\lambda_1}\underbrace{\begin{bmatrix}0\\ 1\end{bmatrix}}_{\mathbf{v}_2}$$



Příklad: Určete vlastní čísla a vlastní vektory těchto matic:

$$\mathbf{A} = \left[\begin{array}{c|c|c} 2 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 2 \end{array}\right], \quad \mathbf{B} = \left[\begin{array}{cc|c} 2 & 1 & 0 \\ 0 & 2 & 0 \\ \hline 0 & 0 & 2 \end{array}\right],$$

$$\mathbf{C} = \left[\begin{array}{c|cc} 2 & 0 & 0 \\ \hline 0 & 2 & 1 \\ 0 & 0 & 2 \end{array}\right], \quad \mathbf{D} = \left[\begin{array}{ccc} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{array}\right].$$

Řešení: Všechny zadané matice mají stejný charakteristický polynom

$$p_{\mathrm{A}}(\lambda) = p_{\mathrm{B}}(\lambda) = p_{\mathrm{C}}(\lambda) = p_{\mathrm{D}}(\lambda) = (2-\lambda)^3,$$

Vidíme, že $\lambda = 2$ je trojnásobné vl. číslo všech čtyř matic.

Vlastní vektory:

$\mathbf{A}$: $v^{(1)} = [1,0,0]^T$
$v^{(2)} = [0,1,0]^T$
$v^{(3)} = [0,0,1]^T$

Pozn.: matice $\mathbf{A}-\lambda\mathbf{I}$ je nulová, tj. systém všech řešení rovnice $\mathbf{A} - \lambda\mathbf{I} = 0$ je lin. kombinací $v^{(1)}$, $v^{(2)}$, $v^{(3)}$.

$\mathbf{B}$: $v^{(1)} = [1,0,0]^T$
$v^{(3)} = [0,0,1]^T$

Pozn.: $\mathbf{B} - \lambda\mathbf{I} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$

$\mathbf{C}$: $v^{(1)} = [1,0,0]^T$
$v^{(2)} = [0,1,0]^T$

Pozn.: $\mathbf{C} - \lambda\mathbf{I} = \begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}$

$\mathbf{D}$: $v^{(1)} = [1,0,0]^T$

Pozn.: $\mathbf{D} - \lambda\mathbf{I} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}$

Poznámka: Počet lineárně nezávislých vlastních vektorů může být menší než je řád matice.

Připomeňme si některé poznatky z lineární algebry.

Definice: Říkáme, že matice $\mathbf{A}$ a $\mathbf{B}$ jsou podobné, existuje-li regulární matice $\mathbf{P}$ taková, že
$\boxed{\mathbf{P^{-1}AP = B}}$, resp. $\boxed{\mathbf{A = PBP^{-1}}}$.

Věta Podobné matice mají stejná vlastní čísla.

Věta: Nechť $\mathbf{A}$ je reálná symetrická matice. Potom existuje ortogonální matice $\mathbf{Q}$ taková, že pro spektrální matici platí

$$\boxed{\Lambda = \mathbf{Q}^T\mathbf{AQ}}.$$

Důkaz:

$$\det(\mathbf{A}-\lambda\mathbf{I}) = \det(\mathbf{A}-\lambda\mathbf{I})\cdot\frac{\det(\mathbf{P})}{\det(\mathbf{P})} = \det(\mathbf{P^{-1}})\cdot\det(\mathbf{A}-\lambda\mathbf{I})\cdot\det(\mathbf{P}) = \det\left[\mathbf{P^{-1}}(\mathbf{A}-\lambda\mathbf{I})\mathbf{P}\right] =$$

$$= \det\left(\underbrace{\mathbf{P^{-1}AP}}_{\mathbf{B}} - \lambda\mathbf{I}\right)$$

□

Věta Je-li $\mathbf{v}$ vlastní vektor matice $\mathbf{A}$, potom $\mathbf{P^{-1}v}$ je vlastní vektor matice $\mathbf{B} = \mathbf{P^{-1}AP}$.

Důkaz:

$$\mathbf{Av} = \lambda\mathbf{v}$$
$$\mathbf{P^{-1}}\cdot/ \quad \mathbf{PBP^{-1}v} = \lambda\mathbf{v}$$
$$\mathbf{B}\underbrace{\mathbf{P^{-1}v}}_{\mathbf{w}} = \lambda\underbrace{\mathbf{P^{-1}v}}_{\mathbf{w}}$$

□

Poznámka: Pokud jsou vlastní vektory $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ lineárně nezávislé, potom platí:

$$\mathbf{X^{-1}AX} = \Lambda \qquad (\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \ \ldots \text{ spektrální matice})$$

Matice $\mathbf{A}$ je tedy podobná diagonální matici. Matice $\mathbf{X}$ je matice, jejíž sloupce tvoří vlastní vektory

$$\boxed{\mathbf{AX} = \mathbf{X}\Lambda}$$

$$\mathbf{A}\cdot\underbrace{\begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_n\end{bmatrix}}_{\mathbf{X}} = \underbrace{\begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_n\end{bmatrix}}_{\mathbf{X}}\cdot\underbrace{\begin{bmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n\end{bmatrix}}_{\Lambda}$$

$$= \begin{bmatrix}\lambda_1\mathbf{v}_1 & \lambda_2\mathbf{v}_2 & \ldots & \lambda_n\mathbf{v}_n\end{bmatrix}$$

Věta Nechť $\mathbf{A}$ je čtvercová matice řádu n, λ její vlastní číslo a $\mathbf{v}$ její vlastní vektor, tj. $\mathbf{Av} = \lambda\mathbf{v}$. Potom platí:

(i) $k \in \mathbb{N} \quad \lambda(\mathbf{A}^k) = [\lambda(\mathbf{A})]^k$

(ii) $\mathbf{A}$ ... regulární $\Rightarrow \quad \lambda(\mathbf{A^{-1}}) = [\lambda(\mathbf{A})]^{-1}$

(iii) $\lambda(\mathbf{A}^H) = \overline{\lambda(\mathbf{A})}$

(iv) vlastní čísla symetrické (hermitovské) matice jsou reálná

(v) vlastní vektory symetrické matice odpovídající různým vlastním číslům jsou ortogonální

(vi) symetrická pozitivně definitní matice má všechna vlastní čísla kladná

Důkaz:

(i)

$$\mathbf{A}\cdot/ \quad \mathbf{Av} = \lambda\mathbf{v} \quad \Rightarrow \quad \mathbf{A^2v} = \lambda\underbrace{\mathbf{Av}}_{=\lambda\mathbf{v}} = \lambda^2\mathbf{v}$$

$$\mathbf{A}\cdot/ \quad \mathbf{A}^k\mathbf{v} = \lambda^k\mathbf{v} \quad \Rightarrow \quad \mathbf{A}^{k+1}\mathbf{v} = \lambda^k\underbrace{\mathbf{Av}}_{=\lambda\mathbf{v}} = \lambda^{k+1}\mathbf{v}$$

(ii)

$$\mathbf{Av} = \lambda\mathbf{v} \quad \Rightarrow \quad \mathbf{A^{-1}Av} = \lambda\mathbf{A^{-1}v}$$
$$\mathbf{v} = \lambda\mathbf{A^{-1}v} \qquad /\cdot\frac{1}{\lambda}$$
$$\frac{1}{\lambda}\mathbf{v} = \mathbf{A^{-1}v}$$

(iii) Označme $\mathbf{B} = \mathbf{A} - \lambda\mathbf{I}$. Platí

$$\det\mathbf{B}^H = \overline{\det\mathbf{B}^T} = \overline{\det\mathbf{B}}$$

$$\det(\mathbf{A}^H - \overline{\lambda}\mathbf{I}) = \overline{\det(\mathbf{A}-\lambda\mathbf{I})} = 0$$

(iv) $\mathbf{A}=\mathbf{A}^H, \quad \mathbf{Av}=\lambda\mathbf{v}$

$$\lambda\mathbf{v}^H\mathbf{v}=\mathbf{v}^H(\lambda\mathbf{v})=\mathbf{v}^H\mathbf{Av}=\underbrace{\mathbf{v}^H\mathbf{A}^H\mathbf{v}}_{\text{číslo}}=\overline{(\mathbf{v}^H\mathbf{A}^H\mathbf{v})^H}=\overline{\mathbf{v}^H\underbrace{\mathbf{Av}}_{\lambda\mathbf{v}}}=\overline{\lambda\mathbf{v}^H\mathbf{v}}\underbrace{=}_{(*)}\bar{\lambda}\mathbf{v}^H\mathbf{v}$$

přidáme ^{H} a $\overline{}$

$$(*)\qquad \overline{(\mathbf{v}^H\mathbf{v})^H}=\mathbf{v}^H\mathbf{v}\qquad\Rightarrow\qquad \overline{\mathbf{v}^H\mathbf{v}}=\mathbf{v}^H\mathbf{v}$$

(v)

$$\left.\begin{array}{ll}\mathbf{u}^H\cdot\,/ & \mathbf{Av}=\lambda\mathbf{v}\\ \mathbf{v}^H\cdot\,/ & \mathbf{Au}=\mu\mathbf{u}\end{array}\right\}\qquad \lambda\neq\mu;\ \lambda=\bar{\lambda},\ \mu=\bar{\mu},\ \mathbf{A}=\mathbf{A}^H$$

$$\begin{array}{l}\mathbf{u}^H\mathbf{Av}=\lambda\mathbf{u}^H\mathbf{v}\\ \mathbf{v}^H\mathbf{Au}=\mu\mathbf{v}^H\mathbf{u}\qquad /^H\\ \mathbf{u}^H\mathbf{A}^H\mathbf{v}=\mu\mathbf{u}^H\mathbf{v}\end{array}$$

$$0=\underbrace{(\lambda-\mu)}_{\neq 0}\mathbf{u}^H\mathbf{v}\qquad\Rightarrow\qquad \boxed{\mathbf{u}^H\mathbf{v}=0}$$

(vi)

$$\begin{array}{l}\mathbf{Av}=\lambda\mathbf{v}\\ \mathbf{v}^T\mathbf{Av}=\lambda\mathbf{v}^T\mathbf{v}\end{array}$$

Platí (pro pozitivně definitní matici $\mathbf{A}$):

$$\forall\mathbf{v}\neq\mathbf{o}:\qquad \mathbf{v}^T\mathbf{Av}>0\qquad\Rightarrow\qquad \lambda\underbrace{\mathbf{v}^T\mathbf{v}}_{>0}>0\qquad\Rightarrow\qquad \boxed{\lambda>0}$$

□

Poznámka: Ortogonální matice $\mathbf{Q}$: $\boxed{\mathbf{Q}^T\mathbf{Q}=\mathbf{I}}$ $/\cdot\mathbf{Q}^{-1}$ $\boxed{\mathbf{Q}^T=\mathbf{Q}^{-1}}$

Podmíněnost úlohy na vlastní čísla

Omezíme se na případ, kdy matice $\mathbf{A}$ má n lineárně nezávislých vlastních vektorů $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}$ odpovídajících vlastním číslům $\lambda_1, \lambda_2, \ldots, \lambda_n$.

- Δa_{ij} ... malé změny v prvcích a_{ij} $\quad |\Delta a_{ij}|\leq\varepsilon$
- porušená matice $\mathbf{A}(\varepsilon)=\mathbf{A}+\Delta\mathbf{A}$ má vlastní čísla $\lambda_k(\varepsilon)=\lambda_k+\Delta\lambda_k$
- dále platí (viz literatura):

$$\boxed{|\lambda_k(\varepsilon)-\lambda_k|\lesssim\varkappa_k\varepsilon,\qquad \text{kde}\quad \varkappa_k=\frac{1}{|\cos\alpha_k|}}$$

kde α_k je úhel $\mathbf{v}_k$ a vlastního vektoru $\mathbf{A}^H$ odpovídajícímu vlastnímu číslu $\overline{\lambda_k}$

- Pro symetrickou matici je

$$\begin{array}{ll}\alpha_k=0 & \Rightarrow\quad \varkappa_k=1\\ |\lambda_k(\varepsilon)-\lambda_k|\leq\varepsilon & \ldots\quad \textbf{dobře podmíněná úloha}\end{array}$$

- Pro nesymetrickou matici je

$$\alpha_k\neq 0\qquad\Rightarrow\qquad \varkappa_k \text{ může být velmi velké}$$

$$\ldots\qquad \textbf{špatně podmíněná úloha}$$

Příklad

script v MATLABu

```
A=[-1 5 0;  0 3 1;  0 0 2]
AH=ctranspose(A)

[v,c]=eig(A,'nobalance')
[vH,cH]=eig(AH,'nobalance')

 disp('------------------------------------------------------------')
 disp(' Vlastni vektory A a AH odpovidajici vlastnimu cislu lambda,')
 disp(' cos uhlu, ktery sviraji a tento uhel')
 for j=1:length(A)
  disp('------------------------------------------------------------')
  lambda=c(j,j)
  vlastni_vektor_A=v(:,j)'
  vlastni_vektor_AH=vH(:,j)'
  cosinus_uhlu=vlastni_vektor_A*vlastni_vektor_AH'...
                  /norm(vlastni_vektor_A)/norm(vlastni_vektor_AH)
  uhel=acos(cosinus_uhlu);
  uhel=uhel*180/pi
  pause
end;
```

výsledky v MATLABu

```
A =
    -1     5     0
     0     3     1
     0     0     2
AH =
    -1     0     0
     5     3     0
     0     1     2
v =
    1.0000    1.0000   -1.0000
         0    0.8000   -0.6000
         0         0    0.6000
c =
    -1     0     0
     0     3     0
     0     0     2
vH =
   -0.8000    0.0000   -0.0000
    1.0000   -1.0000    0.0000
   -0.3333   -1.0000    1.0000
cH =
   -1.0000         0         0
         0    3.0000         0
         0         0    2.0000
------------------------------------------------------------
 Vlastni vektory A a AH odpovidajici vlastnimu cislu lambda,
 cos uhlu, ktery sviraji a tento uhel
------------------------------------------------------------
lambda =
    -1
vlastni_vektor_A =
     1     0     0
vlastni_vektor_AH =
   -0.8000    1.0000   -0.3333
cosinus_uhlu =
   -0.6046
uhel =
  127.1966
------------------------------------------------------------
lambda =
     3
vlastni_vektor_A =
    1.0000    0.8000         0
vlastni_vektor_AH =
    0.0000   -1.0000   -1.0000
cosinus_uhlu =
   -0.4417
uhel =
  116.2141
------------------------------------------------------------
lambda =
     2
vlastni_vektor_A =
   -1.0000   -0.6000    0.6000
```


Příklad 2

$$\mathbf{A} = \begin{bmatrix} 20 & 20 & & & & & \\ & 19 & 20 & & & & \\ & & 18 & 20 & & & \\ & & & \ddots & \ddots & & \\ & & & & \ddots & \ddots & \\ & & & & & 2 & 20 \\ \varepsilon & & & & & & 1 \end{bmatrix}$$

Výpočet determinantu $(\mathbf{A} - \lambda\mathbf{I})$ pomocí rozvoje podle posledního řádku:

$$p_{\mathbf{A}}(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I}) = (20-\lambda)(19-\lambda)\dots(1-\lambda) - 20^{19}\varepsilon$$

- pro $\varepsilon = 0$ $\Rightarrow$ $\boxed{\lambda_{min} = 1}$
- pro $\varepsilon = 20!\,20^{-19} \doteq 4{,}64 \cdot 10^{-7}$

$$p_{\mathbf{A}}(\lambda) = \underbrace{(20-\lambda)(19-\lambda)\dots(1-\lambda)}_{\text{konst.člen } = 20!} - 20^{19}\,20!\,20^{-19} = \lambda \cdot (\dots) = 0 \quad \Rightarrow \quad \boxed{\lambda_{min} = 0}$$

- Malé změně ε odpovídá velká změna vlastního čísla λ_{min}.
- Vlastní čísla nesymetrických matic jsou citlivá na změnu prvků (citlivost roste s rostoucí vzdáleností od diagonály).

Mocninná metoda

Chceme určit vlastní číslo matice $\mathbf{A}$ s největší absolutní hodnotou (dominantní vlastní číslo).

Předpoklady:

1. $\mathbf{A}$ má n-lineárně nezávislých vlastních vektorů
2. existuje jediné dominantní vlastní číslo
3. vlastní čísla lze seřadit: $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$.

Odvození:

1. Zvolíme $\mathbf{y}^{(0)}$ jako lineární kombinaci vlastních vektorů

$$\mathbf{y}^{(0)} = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \dots + \alpha_n\mathbf{v}_n.$$

2. Sestrojíme posloupnost

$$\mathbf{y}^{(k)} = \mathbf{A}\mathbf{y}^{(k-1)}, \quad \text{tj.} \quad \mathbf{y}^{(k)} = \mathbf{A}^k\mathbf{y}^{(0)}.$$
$$\mathbf{y}^{(k)} = \alpha_1\mathbf{A}^k\mathbf{v}_1 + \alpha_2\mathbf{A}^k\mathbf{v}_2 + \dots + \alpha_n\mathbf{A}^k\mathbf{v}_n.$$

3. Platí $\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i$, proto

$$\mathbf{y}^{(k)} = \alpha_1 \underbrace{\lambda_1^k}_{*} \mathbf{v}_1 + \alpha_2 \lambda_2^k \mathbf{v}_2 + \cdots + \alpha_n \lambda_n^k \mathbf{v}_n.$$

4. Vytkneme dominantní vlastní číslo (viz $*$)

$$\mathbf{y}^{(k)} = \lambda_1^k \Big[\alpha_1 \mathbf{v}_1 + \underbrace{\sum_{i=2}^{n} \alpha_i \overbrace{\left(\frac{\lambda_i}{\lambda_1}\right)^k}^{\to 0} \mathbf{v}_i}_{\varepsilon_k \to \mathbf{0}} \Big].$$

5. Analogicky vyjádříme $\mathbf{y}^{(k+1)}$.

6. Vybereme j-tou složku $\mathbf{y}^{(k+1)}$ a $\mathbf{y}^{(k)}$, vydělíme je a provedeme limitní přechod

$$\lim_{k\to\infty} \frac{y_j^{(k+1)}}{y_j^{(k)}} = \lim_{k\to\infty} \frac{\lambda_1^{k+1}(\alpha_1 v_{1,j} + \overbrace{\varepsilon_{k+1,j}}^{\to 0})}{\lambda_1^k(\alpha_1 v_{1,j} + \underbrace{\varepsilon_{k,j}}_{\to 0})} = \lambda_1$$

Příklad

Mocninnou metodou stanovte dominantní vlastní číslo matice $\mathbf{A}$, kde

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{a} \quad \mathbf{y}^{(0)} = [1,1,1]^T.$$

Řešení: Použijeme iterační formuli

$$\mathbf{y}^{(k+1)} = \mathbf{A}\mathbf{y}^{(k)}, \quad \text{pro } k = 0, 1, \ldots$$

$$\mathbf{y}^{(1)} = [2;3;2]^T \qquad \lambda_1^{(1)} = \frac{y_2^{(1)}}{y_2^{(0)}} = 3,$$

$$\mathbf{y}^{(2)} = [5;7;5]^T \qquad \lambda_1^{(2)} = \tfrac{7}{3} \approx 2,3333,$$

$$\mathbf{y}^{(3)} = [12;17;12]^T \quad \lambda_1^{(3)} = \tfrac{17}{7} \approx 2,4285,$$

$$\mathbf{y}^{(4)} = [29;41;29]^T \quad \lambda_1^{(4)} = \tfrac{41}{17} \approx 2,4117,$$

$$\mathbf{y}^{(5)} = [70;99;70]^T \quad \lambda_1^{(5)} = \tfrac{99}{41} \approx \underline{\underline{2,4146}}.$$

Poznámka:

Abychom zamezili přetečení, resp. podtečení při zobrazení čísel v počítači je vhodné v každém kroku normovat vektor $\mathbf{y}^{(k)}$ (norma $\mathbf{y}^{(k)}$ roste, resp. klesá pro vlastní číslo v absolutní hodnotě větší, resp. menší než 1).

$$\mathbf{y}^{(k)} := \frac{\mathbf{y}^{(k)}}{||\mathbf{y}^{(k)}||}$$

výsledky v MATLABu

```
Mocninna metoda pro vypocet dominantniho vlastniho cisla matice A

A =
   900    20     1
    20   500    30
     1    30   100

| k |    y(1)_k     |    y(2)_k     |    y(3)_k     ||  lambda_k   |
-------------------------------------------------------------------
| 0 | 1.000000e+000 | 1.000000e+000 | 1.000000e+000 ||             |
| 1 | 9.210000e+002 | 5.500000e+002 | 1.310000e+002 || 921.0000000 |
| 2 | 8.400310e+005 | 2.973500e+005 | 3.052100e+004 || 912.0857763 |
| 3 | 7.620054e+008 | 1.663913e+008 | 1.281263e+007 || 907.1158338 |
| 4 | 6.891455e+011 | 9.882011e+010 | 7.035006e+009 || 904.3840077 |
| 5 | 6.222144e+014 | 6.340402e+013 | 4.357249e+012 || 902.8781109 |
| 6 | 5.612654e+017 | 4.427701e+016 | 2.960060e+015 || 902.0450147 |
| 7 | 5.060274e+020 | 3.345262e+019 | 2.185582e+018 || 901.5830307 |
| 8 | 4.560959e+023 | 2.691242e+022 | 1.728164e+021 || 901.3264854 |
| 9 | 4.110263e+026 | 2.262997e+025 | 1.436285e+024 || 901.1839104 |
|10 | 3.703777e+029 | 1.957860e+028 | 1.233554e+027 || 901.1046393 |

>> eig(A)
ans =
  1.0e+002 *
   0.977622158203950
   5.012324900167472
   9.010052941628578
```

výsledky v MATLABu

```
Mocninna metoda pro vypocet dominantniho vlastniho cisla matice A
s normovanim vlastniho vektoru v kazde iteraci

A =
   900    20     1
    20   500    30
     1    30   100

| k |     y(1)_k    |     y(2)_k    |     y(3)_k    || lambda_k  |
-----------------------------------------------------------------
| 0 |    1.0000000 |    1.0000000 |    1.0000000 ||           |
| 1 |  921.0000000 |  550.0000000 |  131.0000000 ||921.000000 |
|   |    1.0000000 |    0.5971770 |    0.1422367 ||           |
| 2 |  912.0857763 |  322.8555917 |   33.1389794 ||912.085776 |
|   |    1.0000000 |    0.3539750 |    0.0363332 ||           |
| 3 |  907.1158338 |  198.0775114 |   15.2525693 ||907.115834 |
|   |    1.0000000 |    0.2183597 |    0.0168144 ||           |
| 4 |  904.3840077 |  129.6842642 |    9.2322257 ||904.384008 |
|   |    1.0000000 |    0.1433951 |    0.0102083 ||           |
| 5 |  902.8781109 |   92.0038151 |    6.3226842 ||902.878111 |
|   |    1.0000000 |    0.1019006 |    0.0070028 ||           |
| 6 |  902.0450147 |   71.1603809 |    4.7572988 ||902.045015 |
|   |    1.0000000 |    0.0788878 |    0.0052739 ||           |
| 7 |  901.5830307 |   59.6021361 |    3.8940255 ||901.583031 |
|   |    1.0000000 |    0.0661083 |    0.0043191 ||           |
| 8 |  901.3264854 |   53.1837310 |    3.4151593 ||901.326485 |
|   |    1.0000000 |    0.0590061 |    0.0037890 ||           |
| 9 |  901.1839104 |   49.6167046 |    3.1490857 ||901.183910 |
|   |    1.0000000 |    0.0550572 |    0.0034944 ||           |
|10 |  901.1046393 |   47.6334548 |    3.0011561 ||901.104639 |
|   |    1.0000000 |    0.0528612 |    0.0033305 ||           |
```

výsledky v MATLABu

Numerické metody KMA/NM
Josef Daněk 13.2.2o13

```
Mocninna metoda pro vypocet dominantniho vlastniho cisla matice A

A =
       1/1000        -3/10000       -1/2000
       1/5000         1/200         -1/10000
      -1/1000         1/500          3/1000

| k |     y(1)_k     |    y(2)_k     |    y(3)_k     ||  lambda_k  |
-------------------------------------------------------------------
| 0 |  1.000000e+000 | 1.000000e+000 | 1.000000e+000 ||            |
| 1 |  2.000000e-004 | 5.100000e-003 | 4.000000e-003 ||  0.0051000 |
| 2 | -3.330000e-006 | 2.514000e-005 | 2.200000e-005 ||  0.0049294 |
| 3 | -2.187200e-008 | 1.228340e-007 | 1.196100e-007 ||  0.0048860 |
| 4 | -1.185272e-010 | 5.978346e-010 | 6.263700e-010 ||  0.0052368 |
| 5 | -6.110626e-013 | 2.902831e-012 | 3.193306e-012 ||  0.0050981 |
| 6 | -3.078565e-015 | 1.407261e-014 | 1.599664e-014 ||  0.0050094 |
| 7 | -1.529867e-017 | 6.814767e-017 | 7.921371e-017 ||  0.0049519 |
| 8 | -7.534983e-020 | 3.297572e-019 | 3.892351e-019 ||  0.0049137 |
| 9 | -3.688946e-022 | 1.594793e-021 | 1.902570e-021 ||  0.0048880 |
|10 | -1.798617e-024 | 7.709928e-024 | 9.266189e-024 ||  0.0048704 |

>> eig(A)
ans =
   0.000770409049726
   0.003399468175334
   0.004830122774940
```

výsledky v MATLABu

```
Mocninna metoda pro vypocet dominantniho vlastniho cisla matice A
s normovanim vlastniho vektoru v kazde iteraci

A =
       1/1000        -3/10000       -1/2000
       1/5000         1/200         -1/10000
      -1/1000         1/500          3/1000

| k |    y(1)_k   |   y(2)_k   |   y(3)_k   ||  lambda_k  |
-------------------------------------------------------------
| 0 |   1.0000000 |  1.0000000 |  1.0000000 ||            |
| 1 |   0.0002000 |  0.0051000 |  0.0040000 ||  0.005100  |
|   |   0.0392157 |  1.0000000 |  0.7843137 ||            |
| 2 |  -0.0006529 |  0.0049294 |  0.0043137 ||  0.004929  |
|   |  -0.1324582 |  1.0000000 |  0.8750994 ||            |
| 3 |  -0.0008700 |  0.0048860 |  0.0047578 ||  0.004886  |
|   |  -0.1780614 |  1.0000000 |  0.9737532 ||            |
| 4 |  -0.0009649 |  0.0048670 |  0.0050993 ||  0.005237  |
|   |  -0.1892287 |  0.9544432 |  1.0000000 ||            |
| 5 |  -0.0009756 |  0.0046344 |  0.0050981 ||  0.005098  |
|   |  -0.1913573 |  0.9090360 |  1.0000000 ||            |
| 6 |  -0.0009641 |  0.0044069 |  0.0050094 ||  0.005009  |
|   |  -0.1924507 |  0.8797227 |  1.0000000 ||            |
| 7 |  -0.0009564 |  0.0042601 |  0.0049519 ||  0.004952  |
|   |  -0.1931316 |  0.8603014 |  1.0000000 ||            |
| 8 |  -0.0009512 |  0.0041629 |  0.0049137 ||  0.004914  |
|   |  -0.1935843 |  0.8471929 |  1.0000000 ||            |
| 9 |  -0.0009477 |  0.0040972 |  0.0048880 ||  0.004888  |
|   |  -0.1938928 |  0.8382309 |  1.0000000 ||            |
|10 |  -0.0009454 |  0.0040524 |  0.0048704 ||  0.004870  |
|   |  -0.1941054 |  0.8320495 |  1.0000000 ||            |
```

Poznámka:

Nejlepší aproximaci dostaneme, dělíme-li složky, které mají největší absolutní hodnotu.
Obecně nelze použít libovolnou složku vektoru $\mathbf{y}^{(k)}$ neboť odpovídající vlastní vektor ji může mít nulovou.

výsledky v MATLABu


```
A =
     1     1     0
     0     2     0
     0     0     3

>> [v,c]=eig(A,'nobalance')

v =
     1     1     0
     0     1     0
     0     0     1

c =
     1     0     0
     0     2     0
     0     0     3

Mocninna metoda pro vypocet dominantniho vlastniho cisla matice A

| k |    y(1)_k    |    y(2)_k    |    y(3)_k    ||lambda_k 1s || 2s || 3s |
-------------------------------------------------------------------------------
| 0 | 2.000000e+00 | 1.000000e+00 | 1.000000e+00 ||            ||    ||    |
| 1 | 3.000000e+00 | 2.000000e+00 | 3.000000e+00 ||  1.5000000 ||  2 ||  3 |
| 2 | 5.000000e+00 | 4.000000e+00 | 9.000000e+00 ||  1.6666667 ||  2 ||  3 |
| 3 | 9.000000e+00 | 8.000000e+00 | 2.700000e+01 ||  1.8000000 ||  2 ||  3 |
| 4 | 1.700000e+01 | 1.600000e+01 | 8.100000e+01 ||  1.8888889 ||  2 ||  3 |
| 5 | 3.300000e+01 | 3.200000e+01 | 2.430000e+02 ||  1.9411765 ||  2 ||  3 |
| 6 | 6.500000e+01 | 6.400000e+01 | 7.290000e+02 ||  1.9696970 ||  2 ||  3 |
| 7 | 1.290000e+02 | 1.280000e+02 | 2.187000e+03 ||  1.9846154 ||  2 ||  3 |
| 8 | 2.570000e+02 | 2.560000e+02 | 6.561000e+03 ||  1.9922481 ||  2 ||  3 |
| 9 | 5.130000e+02 | 5.120000e+02 | 1.968300e+04 ||  1.9961089 ||  2 ||  3 |
|10 | 1.025000e+03 | 1.024000e+03 | 5.904900e+04 ||  1.9980507 ||  2 ||  3 |
```

Poznámka:

Při praktickém použití mocninné metody neověřujeme, zda jsou splněny předpoklady odvození.

Zadaná matice nemusí mít jediné dominantní vlastní číslo nebo
počet lineárně nezávislých vlastních vektorů může být menší než řád matice.

Při nesplněných předpokladech odvození může být konvergence pomalá.
Další nevýhodou mocninné metody je potom odhad chyby získané aproximace.

výsledky v MATLABu

```
A =
     3     1     1
    -1     1     0
     0     0     1

>> [v,c]=eig(A,'nobalance')

v =
    1.0000   -1.0000    0.0000
   -1.0000    1.0000   -1.0000
         0         0    1.0000
c =
    2.0000         0         0
         0    2.0000         0
         0         0    1.0000

Mocninna metoda pro vypocet dominantniho vlastniho cisla matice A

| k |     y(1)_k    |      y(2)_k    |     y(3)_k    ||  lambda_k  |
-------------------------------------------------------------------
| 0 | 1.000000e+000 |  1.000000e+000 | 1.000000e+000 ||            |
| 1 | 5.000000e+000 |  0.000000e+000 | 1.000000e+000 ||  5.0000000 |
| 2 | 1.600000e+001 | -5.000000e+000 | 1.000000e+000 ||  3.2000000 |
| 3 | 4.400000e+001 | -2.100000e+001 | 1.000000e+000 ||  2.7500000 |
| 4 | 1.120000e+002 | -6.500000e+001 | 1.000000e+000 ||  2.5454545 |
| 5 | 2.720000e+002 | -1.770000e+002 | 1.000000e+000 ||  2.4285714 |
| 6 | 6.400000e+002 | -4.490000e+002 | 1.000000e+000 ||  2.3529412 |
| 7 | 1.472000e+003 | -1.089000e+003 | 1.000000e+000 ||  2.3000000 |
| 8 | 3.328000e+003 | -2.561000e+003 | 1.000000e+000 ||  2.2608696 |
| 9 | 7.424000e+003 | -5.889000e+003 | 1.000000e+000 ||  2.2307692 |
|10 | 1.638400e+004 | -1.331300e+004 | 1.000000e+000 ||  2.2068966 |

|50 | 8.556839e+016 | -8.219069e+016 | 1.000000e+000 ||  2.0402685 |

|100| 1.914152e+032 | -1.876123e+032 | 1.000000e+000 ||  2.0200669 |

|150| 3.225580e+047 | -3.182762e+047 | 1.000000e+000 ||  2.0133630 |

|200| 4.836884e+062 | -4.788675e+062 | 1.000000e+000 ||  2.0100167 |

|250| 6.802785e+077 | -6.748508e+077 | 1.000000e+000 ||  2.0080107 |

|300| 9.187032e+092 | -9.125921e+092 | 1.000000e+000 ||  2.0066741 |

|1000|1.608334e+304 | -1.605120e+304 | 1.000000e+000 ||  2.0020007 |


-------------------------------------------------------------------

|1013|      Inf     | -1.332031e+308 | 1.000000e+000 ||        Inf |
```


Z uvedeného příkladu je dále vidět, že je opravdu vhodné normovat vektor $\mathbf{y}^{(k)}$ a zabránit tak přetečení.

Poznámka:

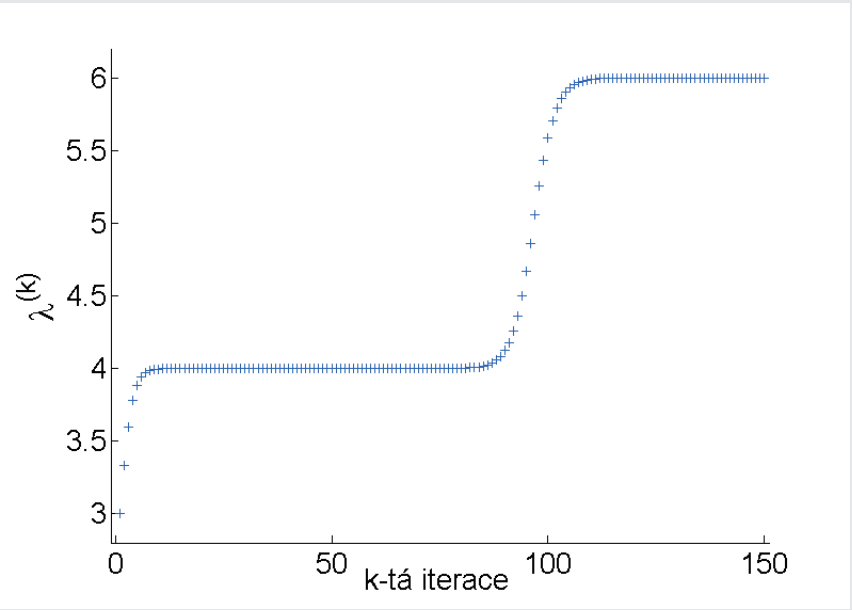
Při praktickém použití mocninné metody použijeme například počáteční volbu vektoru $\mathbf{y}^{(0)} = [1, 1, 1, \ldots, 1]^T$.

Je-li ovšem vektor $\mathbf{y}^{(0)}$ takovou lineární kombinací vlastních vektorů, že koeficient u vlastního vektoru odpovídajícího dominantnímu vlastnímu číslu bude roven 0, potom mocninná metoda nevypočte dominantní vlastní číslo, ale nejbližší nižší, u kterého u odpovídajícího vlastního vektoru bude nenulový koeficient.

Pokud bychom prováděli výpočet dostatečně dlouho, dojde vlivem zaokrouhlovacích chyb k tomu, že u příslušné iterace $\mathbf{y}^{(k)}$ bude již zmiňovaný koeficient u vlastního vektoru odpovídajícího dominantnímu vlastnímu číslu již nenulový a metoda nakonec dominantní vlastní číslo nalezne.

výsledky v MATLABu

```
A =
     4    -3     2
     0     2     0
     0    -4     6

>> [v,c]=eig(A,'nobalance')

v =
    1.0000    1.0000   -0.5000
         0         0   -1.0000
         0    1.0000   -1.0000
c =
     4     0     0
     0     6     0
     0     0     2

>> y=[1 1 1]'
>> alpha=(v\y)'

alpha =
    0.5000         0   -1.0000

Mocninna metoda pro vypocet dominantniho vlastniho cisla matice A
s normovanim vlastniho vektoru v kazde iteraci

| k |   y(1)_k   |   y(2)_k   |   y(3)_k   ||  lambda_k  |
-------------------------------------------------------------
| 0 |  1.0000000 |  1.0000000 |  1.0000000 ||            |
| 1 |  3.0000000 |  2.0000000 |  2.0000000 ||  3.000000  |
|   |  1.0000000 |  0.6666667 |  0.6666667 ||            |
| 2 |  3.3333333 |  1.3333333 |  1.3333333 ||  3.333333  |
|   |  1.0000000 |  0.4000000 |  0.4000000 ||            |
| 3 |  3.6000000 |  0.8000000 |  0.8000000 ||  3.600000  |
|   |  1.0000000 |  0.2222222 |  0.2222222 ||            |
| 4 |  3.7777778 |  0.4444444 |  0.4444444 ||  3.777778  |
|   |  1.0000000 |  0.1176471 |  0.1176471 ||            |
| 5 |  3.8823529 |  0.2352941 |  0.2352941 ||  3.882353  |
|   |  1.0000000 |  0.0606061 |  0.0606061 ||            |
| 6 |  3.9393939 |  0.1212121 |  0.1212121 ||  3.939394  |
|   |  1.0000000 |  0.0307692 |  0.0307692 ||            |
| 7 |  3.9692308 |  0.0615385 |  0.0615385 ||  3.969231  |
|   |  1.0000000 |  0.0155039 |  0.0155039 ||            |
| 8 |  3.9844961 |  0.0310078 |  0.0310078 ||  3.984496  |
|   |  1.0000000 |  0.0077821 |  0.0077821 ||            |
| 9 |  3.9922179 |  0.0155642 |  0.0155642 ||  3.992218  |
|   |  1.0000000 |  0.0038986 |  0.0038986 ||            |
|10 |  3.9961014 |  0.0077973 |  0.0077973 ||  3.996101  |
|   |  1.0000000 |  0.0019512 |  0.0019512 ||            |
|11 |  3.9980488 |  0.0039024 |  0.0039024 ||  3.998049  |
|   |  1.0000000 |  0.0009761 |  0.0009761 ||            |
|12 |  3.9990239 |  0.0019522 |  0.0019522 ||  3.999024  |
|   |  1.0000000 |  0.0004882 |  0.0004882 ||            |
|13 |  3.9995118 |  0.0009763 |  0.0009763 ||  3.999512  |
|   |  1.0000000 |  0.0002441 |  0.0002441 ||            |
```

Poznámka:

Iterační proces ukončujeme použitím zastavovací podmínky ve tvaru

$$|\lambda_1^{(k+1)} - \lambda_1^{(k)}| < \varepsilon .$$

Posuďte výsledky získané pro následující příklad. Kde je problém ?

výsledky v MATLABu

```
A =
     3     3     0
     0     4     2
     0     0     1

>> [v,c]=eig(A,'nobalance')

v =
    1.0000    1.0000    1.0000
         0    0.3333   -0.6667
         0         0    1.0000
c =
     3     0     0
     0     4     0
     0     0     1

>> y=[1 1 1]'
>> alpha=(v\y)'

alpha =
    -5     5     1

Mocninna metoda pro vypocet dominantniho vlastniho cisla matice A

| k |     y(1)_k    |     y(2)_k    |     y(3)_k    ||  lambda_k  |
-------------------------------------------------------------------
| 0 | 1.000000e+000 | 1.000000e+000 | 1.000000e+000 ||            |
| 1 | 6.000000e+000 | 6.000000e+000 | 1.000000e+000 ||  6.0000000 |
| 2 | 3.600000e+001 | 2.600000e+001 | 1.000000e+000 ||  6.0000000 |
```

```
>> y=[1 0 1]'
>> alpha=(v\y)'

alpha =
    -2     2     1

Mocninna metoda pro vypocet dominantniho vlastniho cisla matice A

| k |     y(1)_k    |     y(2)_k    |     y(3)_k    ||  lambda_k  |
-------------------------------------------------------------------
| 0 | 1.000000e+000 | 0.000000e+000 | 1.000000e+000 ||            |
| 1 | 3.000000e+000 | 2.000000e+000 | 1.000000e+000 ||  3.0000000 |
| 2 | 1.500000e+001 | 1.000000e+001 | 1.000000e+000 ||  5.0000000 |
| 3 | 7.500000e+001 | 4.200000e+001 | 1.000000e+000 ||  5.0000000 |
```

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

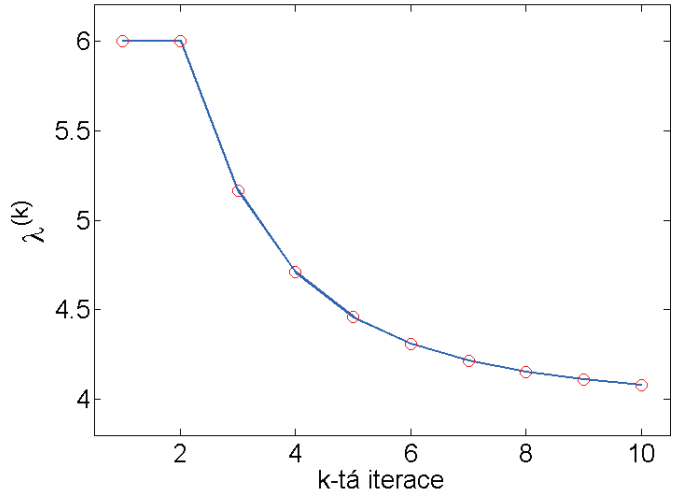
```
>> y=[3 2 1]'
>> alpha=(v\y)'

alpha =
    -6     8     1

Mocninna metoda pro vypocet dominantniho vlastniho cisla matice A

| k |     y(1)_k    |     y(2)_k    |     y(3)_k    ||  lambda_k  |
-------------------------------------------------------------------
| 0 | 3.000000e+000 | 2.000000e+000 | 1.000000e+000 ||            |
| 1 | 1.500000e+001 | 1.000000e+001 | 1.000000e+000 ||  5.0000000 |
| 2 | 7.500000e+001 | 4.200000e+001 | 1.000000e+000 ||  5.0000000 |
```

Všechny předpoklady byly splněny, byla použita i vhodná počáteční volba vektoru $\mathbf{y}^{(0)}$.

Jediné, co se stalo je skutečnost, že v posloupnosti přibližných řešení generovaných mocninnou metodou se objevily dva po sobě jdoucí stejné členy, které zdaleka nebyly limitou této posloupnosti.

výsledky v MATLABu

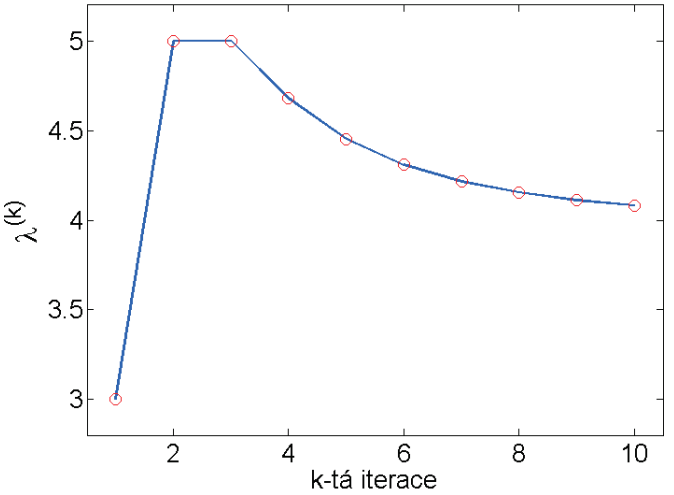
```
Mocninna metoda pro vypocet dominantniho vlastniho cisla matice A

| k |     y(1)_k    |     y(2)_k    |     y(3)_k    ||  lambda_k  |
-------------------------------------------------------------------
| 0 | 1.000000e+000 | 1.000000e+000 | 1.000000e+000 ||            |
| 1 | 6.000000e+000 | 6.000000e+000 | 1.000000e+000 ||  6.0000000 |
| 2 | 3.600000e+001 | 2.600000e+001 | 1.000000e+000 ||  6.0000000 |

| 3 | 1.860000e+002 | 1.060000e+002 | 1.000000e+000 ||  5.1666667 |
| 4 | 8.760000e+002 | 4.260000e+002 | 1.000000e+000 ||  4.7096774 |
| 5 | 3.906000e+003 | 1.706000e+003 | 1.000000e+000 ||  4.4589041 |
| 6 | 1.683600e+004 | 6.826000e+003 | 1.000000e+000 ||  4.3102919 |
| 7 | 7.098600e+004 | 2.730600e+004 | 1.000000e+000 ||  4.2163222 |
| 8 | 2.948760e+005 | 1.092260e+005 | 1.000000e+000 ||  4.1540022 |
| 9 | 1.212306e+006 | 4.369060e+005 | 1.000000e+000 ||  4.1112400 |
|10 | 4.947636e+006 | 1.747626e+006 | 1.000000e+000 ||  4.0811775 |
```



Mocninna metoda pro vypocet dominantniho vlastniho cisla matice A

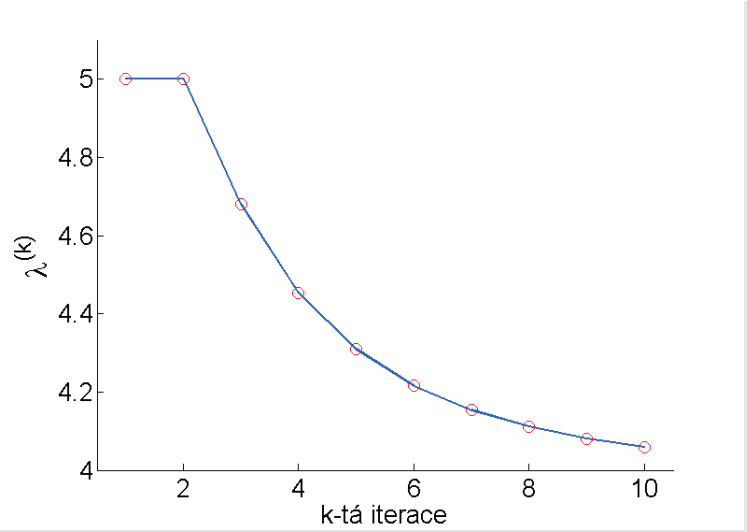
k	y(1)_k	y(2)_k	y(3)_k	lambda_k
0	1.000000e+000	0.000000e+000	1.000000e+000	
1	3.000000e+000	2.000000e+000	1.000000e+000	3.0000000
2	1.500000e+001	1.000000e+001	1.000000e+000	5.0000000
3	7.500000e+001	4.200000e+001	1.000000e+000	5.0000000
4	3.510000e+002	1.700000e+002	1.000000e+000	4.6800000
5	1.563000e+003	6.820000e+002	1.000000e+000	4.4529915
6	6.735000e+003	2.730000e+003	1.000000e+000	4.3090211
7	2.839500e+004	1.092200e+004	1.000000e+000	4.2160356
8	1.179510e+005	4.369000e+004	1.000000e+000	4.1539356
9	4.849230e+005	1.747620e+005	1.000000e+000	4.1112242
10	1.979055e+006	6.990500e+005	1.000000e+000	4.0811737

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



Mocninna metoda pro vypocet dominantniho vlastniho cisla matice A

k	y(1)_k	y(2)_k	y(3)_k	lambda_k
0	3.000000e+000	2.000000e+000	1.000000e+000	
1	1.500000e+001	1.000000e+001	1.000000e+000	5.0000000
2	7.500000e+001	4.200000e+001	1.000000e+000	5.0000000
3	3.510000e+002	1.700000e+002	1.000000e+000	4.6800000
4	1.563000e+003	6.820000e+002	1.000000e+000	4.4529915
5	6.735000e+003	2.730000e+003	1.000000e+000	4.3090211
6	2.839500e+004	1.092200e+004	1.000000e+000	4.2160356
7	1.179510e+005	4.369000e+004	1.000000e+000	4.1539356
8	4.849230e+005	1.747620e+005	1.000000e+000	4.1112242
9	1.979055e+006	6.990500e+005	1.000000e+000	4.0811737
10	8.034315e+006	2.796202e+006	1.000000e+000	4.0596724



Poznámka:

Pro **urychlování konvergence metody**

- lze použít např. Aitkenův proces,
- pokud platí, že λ_1 a λ_2 jsou si velmi blízká, rychlost konvergence mocninné metody bude malá; předpokládáme-li např., že jsou všechna vlastní čísla reálná, lze použít Wilkinsonovu metodu:

$\mathbf{A}$ má vlastní čísla $\lambda_1, \lambda_2, \ldots, \lambda_n$

$\widehat{\mathbf{A}} = \mathbf{A} - p\mathbf{I}$ má vlastní čísla $\lambda_1 - p, \lambda_2 - p, \ldots, \lambda_n - p$

Uvažujme pro jednoduchost, že jsou všechna $\lambda_i > 0$.

Pomalou konvergenci způsobuje podíl $\left|\frac{\lambda_2}{\lambda_1}\right| \lessapprox 1$.

Chceme tento podíl co nejvíce zmenšit: $\frac{\lambda_2 - p}{\lambda_1 - p} < \frac{\lambda_2}{\lambda_1}$.

Jak musíme volit p? $p_{opt} = \frac{\lambda_2 + \lambda_n}{2}$ ... představuje posunutý počátek

Příklad:

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

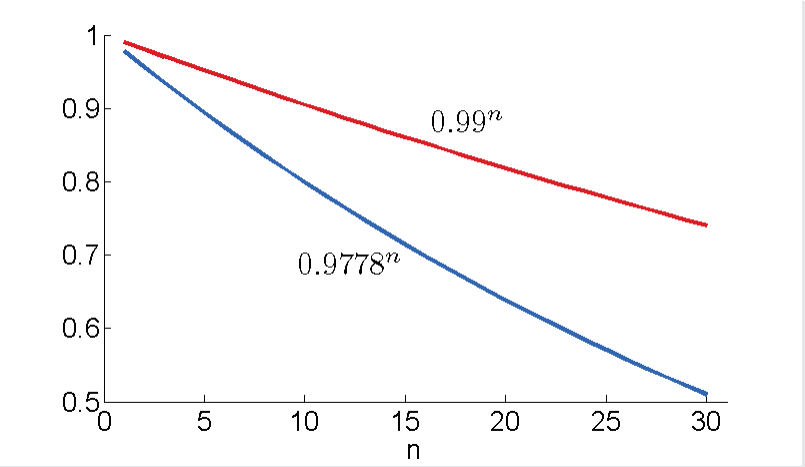
$\mathbf{A} = \begin{bmatrix} 100 & & \\ & 99 & \\ & & 11 \end{bmatrix}$ ... vlastní čísla $\lambda_1 = 100$, $\lambda_2 = 99$, $\lambda_3 = 11$ $\Rightarrow$ $p_{opt} = \frac{99 + 11}{2} = 55$

$\widehat{\mathbf{A}} = \mathbf{A} - 55\mathbf{I} = \begin{bmatrix} 45 & & \\ & 44 & \\ & & -44 \end{bmatrix}$ ... vlastní čísla $\widehat{\lambda}_1 = 45$, $\widehat{\lambda}_2 = 44$, $\widehat{\lambda}_3 = -44$

$$\frac{\lambda_2}{\lambda_1} = \frac{99}{100} = 0,99 \qquad \frac{\widehat{\lambda}_2}{\widehat{\lambda}_1} = \frac{44}{45} \doteq 0,9778$$

výsledky v MATLABu

n	0.99^n	0.9778^n
1	0.9900	0.9778
2	0.9801	0.9561
3	0.9703	0.9349
4	0.9606	0.9141
5	0.9510	0.8938
6	0.9415	0.8740
7	0.9321	0.8546
8	0.9227	0.8356
9	0.9135	0.8171
10	0.9044	0.7989
11	0.8953	0.7812
12	0.8864	0.7638
13	0.8775	0.7469
14	0.8687	0.7303
15	0.8601	0.7141
16	0.8515	0.6982
17	0.8429	0.6827
18	0.8345	0.6676
19	0.8262	0.6528
20	0.8179	0.6383
21	0.8097	0.6241
22	0.8016	0.6102
23	0.7936	0.5967
24	0.7857	0.5834
25	0.7778	0.5705
26	0.7700	0.5578
27	0.7623	0.5454
28	0.7547	0.5333
29	0.7472	0.5215
30	0.7397	0.5099



Metoda Rayleighova podílu

Chceme určit vlastní číslo matice $\mathbf{A}$ s největší absolutní hodnotou (dominantní vlastní číslo).

Při odvození metody Rayleighova podílu budeme navíc (oproti mocninné metodě) předpokládat, že matice $\mathbf{A}$ je symetrická (reálná). Potom musí být vlastní vektory ortonormální $\left(\mathbf{v}_i^T\mathbf{v}_j = 0 \text{ pro } i \neq j \text{ a } \mathbf{v}_i^T\mathbf{v}_i = 1\right)$.

Odvození:

6. krok z odvození mocninné metody nahradíme vyjádřením součinu $\mathbf{y}^{(k)T}\mathbf{y}^{(k)}$

$$\mathbf{y}^{(k)T}\mathbf{y}^{(k)} = \lambda_1^k\Big[\alpha_1\mathbf{v}_1^T + \overbrace{\sum_{i=2}^{n}\alpha_i\left(\frac{\lambda_i}{\lambda_1}\right)^k\mathbf{v}_i^T}^{\varepsilon_k^T}\Big] \cdot \lambda_1^k\Big[\alpha_1\mathbf{v}_1 + \overbrace{\sum_{i=2}^{n}\alpha_i\left(\frac{\lambda_i}{\lambda_1}\right)^k\mathbf{v}_i}^{\varepsilon_k}\Big] =$$

$$= \lambda_1^{2k}\Big[\alpha_1^2 + \underbrace{\sum_{i=2}^{n}\alpha_i^2\left(\frac{\lambda_i}{\lambda_1}\right)^{2k}}_{\varepsilon_k^T\varepsilon_k}\Big]$$

a součinu $\mathbf{y}^{(k)T}\mathbf{y}^{(k+1)T}$

$$\mathbf{y}^{(k)T}\mathbf{y}^{(k+1)} = \lambda_1^k\Big[\alpha_1\mathbf{v}_1^T + \overbrace{\sum_{i=2}^{n}\alpha_i\left(\frac{\lambda_i}{\lambda_1}\right)^k\mathbf{v}_i^T}^{\varepsilon_k^T}\Big] \cdot \lambda_1^{k+1}\Big[\alpha_1\mathbf{v}_1 + \overbrace{\sum_{i=2}^{n}\alpha_i\left(\frac{\lambda_i}{\lambda_1}\right)^{k+1}\mathbf{v}_i}^{\varepsilon_{k+1}}\Big] =$$

$$= \lambda_1^{2k+1}\Big[\alpha_1^2 + \underbrace{\sum_{i=2}^{n}\alpha_i^2\left(\frac{\lambda_i}{\lambda_1}\right)^{2k+1}}_{\varepsilon_k^T\varepsilon_{k+1}}\Big]$$

Dostáváme:

$$\lim_{k\to\infty}\frac{\mathbf{y}^{(k)\mathsf{T}}\mathbf{A}\mathbf{y}^{(k)}}{\mathbf{y}^{(k)\mathsf{T}}\mathbf{y}^{(k)}} = \lim_{k\to\infty}\frac{\mathbf{y}^{(k)\mathsf{T}}\mathbf{y}^{(k+1)}}{\mathbf{y}^{(k)T}\mathbf{y}^{(k)}} = \frac{\lambda_1^{2k+1}(\alpha_1^2 + \overbrace{\varepsilon_k^T\varepsilon_{k+1}}^{\to 0})}{\lambda_1^{2k}(\alpha_1^2 + \underbrace{\varepsilon_k^T\varepsilon_k}_{\to 0})} = \lambda_1.$$

Poznámka: Součin $\varepsilon_k^T\varepsilon_k$ konverguje k nule (pro $k \to \infty$) zhruba dvakrát rychleji než ε_k k nulovému vektoru

$\Rightarrow$ metoda Rayleighova podílu bude rychlejší než mocninná metoda.

Příklad

Metodou Rayleighova podílu určete dominantní vlastní číslo matice $\mathbf{A}$, kde

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{a} \quad \mathbf{y}^{(0)} = [1;1;1]^T.$$

Řešení:

$$\mathbf{y}^{(1)} = [2;3;2]^T \qquad \lambda_1^{(1)} = \frac{\mathbf{y}^{(0)T}\mathbf{y}^{(1)}}{\mathbf{y}^{(0)T}\mathbf{y}^{(0)}} = \tfrac{7}{3} \approx 2,3333,$$

$$\mathbf{y}^{(2)} = [5;7;5]^T \qquad \lambda_1^{(2)} = \tfrac{41}{17} \approx 2,4117,$$

$$\mathbf{y}^{(3)} = [12;17;12]^T \quad \lambda_1^{(3)} = \tfrac{60+119+60}{25+49+25} = \tfrac{239}{99} \approx \underline{\underline{2,41417}}.$$

Příklad 3

Pro stejné zadání symetrické matice $\mathbf{A}$ porovnejme rychlost konvergence mocniné metody a metody Rayleighova podílu.

$$\mathbf{A} = \begin{bmatrix} 60 & 20 & 10 & 1 \\ 20 & 50 & 10 & 2 \\ 10 & 10 & 30 & 5 \\ 1 & 2 & 5 & 10 \end{bmatrix}, \quad \mathbf{y}^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \varepsilon = 10^{-5}.$$

výsledky v MATLABu

```
A =
    60    20    10     1
    20    50    10     2
    10    10    30     5
     1     2     5    10

>> [v,c]=eig(A,'nobalance')

v =
   0.029201136324116   0.070393944798935  -0.657594927428575  -0.749507103097250
  -0.002755406652908   0.347503151150767   0.720730957555407  -0.599817350945878
  -0.241058474917619  -0.907169537175591   0.206770509289230  -0.276007606741715
   0.970067272431118  -0.226560551059880   0.073224003781179  -0.047728910858600

c =
   8.781938031360916                   0                   0                   0
                   0  26.642118061325501                   0                   0
                   0                   0  34.824093743363321                   0
                   0                   0                   0  79.751850163950266

>> y=[1 1 1 1]'
>> alpha=(v\y)'

alpha =
   0.755454527184707  -0.715832992285768   0.343130543197241  -1.673060971643443
```

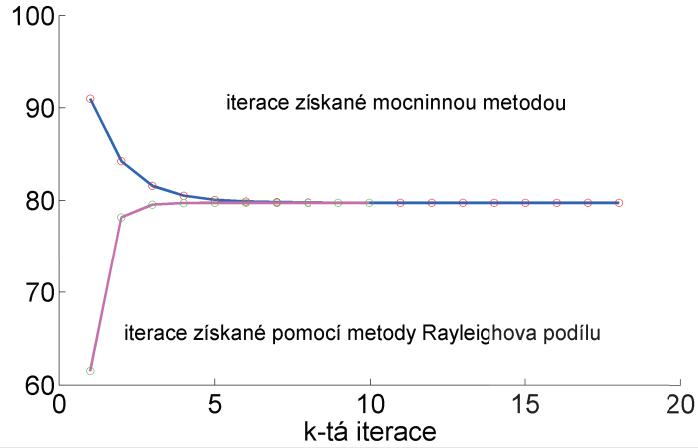
Mocninna metoda pro vypocet dominantniho vlastniho cisla matice A

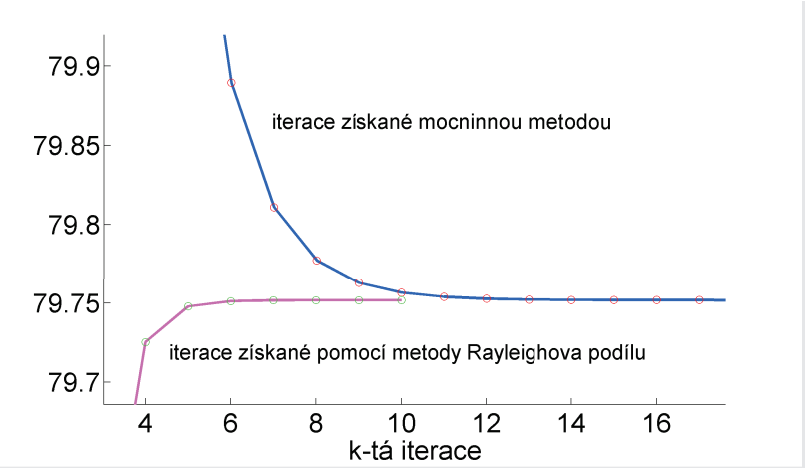
k	y(1)_k	y(2)_k	y(3)_k	y(4)_k	lambda_k
0	1.000000e+000	1.000000e+000	1.000000e+000	1.000000e+000	
1	9.100000e+001	8.200000e+001	5.500000e+001	1.800000e+001	91.0000000
2	7.668000e+003	6.506000e+003	3.470000e+003	7.100000e+002	84.2637363
3	6.256100e+005	5.147800e+005	2.493900e+005	4.513000e+004	81.5871153
4	5.037123e+007	4.083536e+007	1.911125e+007	3.353420e+006	80.5153850
5	4.033447e+009	3.247012e+009	1.502171e+009	2.611324e+008	80.0744179
6	3.222299e+011	2.585635e+011	1.191754e+011	2.064965e+010	79.8894588
7	2.571747e+013	2.060583e+013	9.486443e+012	1.641730e+012	79.8109287
8	2.051671e+015	1.642789e+015	7.560349e+014	1.307786e+014	79.7773243
9	1.636471e+017	1.309947e+017	6.027953e+016	1.042521e+016	79.7628684
10	1.305194e+019	1.044633e+019	4.806931e+018	8.312864e+017	79.7566267
11	1.040944e+021	8.330871e+020	3.833471e+020	6.629211e+019	79.7539244
12	8.301813e+022	6.643928e+022	3.057218e+022	5.286774e+021	79.7527521
13	6.620882e+024	5.298622e+024	2.438173e+024	4.216253e+023	79.7522427
14	5.280287e+026	4.225737e+026	1.944484e+026	3.362525e+025	79.7520212
15	4.211131e+028	3.370099e+028	1.550760e+028	2.681670e+027	79.7519247
16	3.358456e+030	2.687715e+030	1.236759e+030	2.138680e+029	79.7518827
17	2.678431e+032	2.143502e+032	9.863383e+031	1.705636e+031	79.7518643
18	2.136099e+034	1.709482e+034	7.866230e+033	1.360276e+033	79.7518563

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Metoda Rayleighova podilu pro vypocet dominantniho vlastniho cisla matice A

k	y(1)_k	y(2)_k	y(3)_k	y(4)_k	lambda_k
0	1.000000e+000	1.000000e+000	1.000000e+000	1.000000e+000	
1	9.100000e+001	8.200000e+001	5.500000e+001	1.800000e+001	61.5000000
2	7.668000e+003	6.506000e+003	3.470000e+003	7.100000e+002	78.1796884
3	6.256100e+005	5.147800e+005	2.493900e+005	4.513000e+004	79.5606714
4	5.037123e+007	4.083536e+007	1.911125e+007	3.353420e+006	79.7252270
5	4.033447e+009	3.247012e+009	1.502171e+009	2.611324e+008	79.7478446
6	3.222299e+011	2.585635e+011	1.191754e+011	2.064965e+010	79.7512057
7	2.571747e+013	2.060583e+013	9.486443e+012	1.641730e+012	79.7517406
8	2.051671e+015	1.642789e+015	7.560349e+014	1.307786e+014	79.7518308
9	1.636471e+017	1.309947e+017	6.027953e+016	1.042521e+016	79.7518466
10	1.305194e+019	1.044633e+019	4.806931e+018	8.312864e+017	79.7518495





Poznámka:

Pokud jsme vypočítali λ_1, $\mathbf{v}_1$ a chceme určit další vlastní čísla, resp. vlastní vektory λ_2, $\mathbf{v}_2$, λ_3, $\mathbf{v}_3$, ... (ovšem ne všechny), můžeme použít metody využívající znalosti λ_1, $\mathbf{v}_1$ atd.

- **Maticová redukce**

Věta: Nechť λ_1 je vlastní číslo matice $\mathbf{A}$ a $\mathbf{v}_1$ jemu odpovídající vlastní vektor. Nechť $\mathbf{w}$ je libovolný vektor, pro který $\mathbf{w}^T\mathbf{v}_1 = 1$. Pak matice

$$W_1 = \mathbf{A} - \lambda_1 \mathbf{v}_1 \mathbf{w}^T$$

má stejná vlastní čísla jako matice $\mathbf{A}$, s výjimkou vlastního čísla λ_1, které je nahrazeno číslem 0 ($\mathbf{W}_1$ ... redukovaná matice).

Otázka: Jak volit vektor $\mathbf{w}$?

1. Hotellingova redukce

   $\mathbf{w}$ ... levý vlastní vektor vlastního čísla λ_1 (je normalizován: $\mathbf{w}^T\mathbf{v}_1 = 1$)

   obvykle levý vlastní vektor neznáme a může být $\mathbf{w}^T\mathbf{v}_1 = 0$

   užijeme tuto metodu pro symetrické matice, protože potom $\mathbf{w} = \mathbf{v}_1$

   (tj. pravý a levý vlastní vektor odpovídající stejnému vlastnímu číslu je stejný)

2. Wielandtova redukce

   (viz literatura)

3. podobnostní redukce

   (viz literatura)

- **Anihilační postupy**

Je-li $\mathbf{w}$ libovolný vektor a λ_1, $\mathbf{v}_1$ vlastní číslo a vektor matice $\mathbf{A}$, pak vektor

Numerické metody
Josef Daněk
KMA/NM
13.2.2013

$$\mathbf{u} = (\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{w}$$

nemá složku ve směru vektoru $\mathbf{v}_1$

Cv. vyjádříme vektor $\mathbf{w}$ jako lineární kombinaci vlastních vektorů $\mathbf{v}_i$ a ověříme

$$\mathbf{w} = \sum_{i=1}^{n} \beta_i \mathbf{v}_i$$

$$\mathbf{u} = (\mathbf{A} - \lambda_1\mathbf{I})(\sum_{i=1}^{n} \beta_i \mathbf{v}_i) = \sum_{i=1}^{n} (\beta_i \underbrace{\mathbf{A}\mathbf{v}_i}_{\lambda_i \mathbf{v}_i} - \beta_i \lambda_1 \mathbf{v}_i) = \beta_1\lambda_1\mathbf{v}_1 - \beta_1\lambda_1\mathbf{v}_1 + \sum_{i=2}^{n} (\beta_i\lambda_i\mathbf{v}_i - \lambda_1\beta_i\mathbf{v}_i) =$$

$$= 0 \cdot \mathbf{v}_1 + \sum_{i=2}^{n} \beta_i(\lambda_i - \lambda_1)\mathbf{v}_i$$

□

- Použijeme-li $\mathbf{u}$ jako vstup do mocninné metody, získáme λ_2, v_2 (pozor na problém se zaokrouhlovacími chybami).
- Abychom odstranili tento problém, odbouráváme stále složku ve směru $\mathbf{v}_1$

$$\mathbf{u} = (\mathbf{A} - \lambda_1\mathbf{I})\mathbf{u}$$

Charakteristika metod na řešení úplného problému:

1) metody založené na výpočtech vlastních čísel **pomocí charakteristického polynomu**

   Nevýhodné pro velká n (řád matice $\mathbf{A}$), protože je obtížné vypočítat $p_{\mathbf{A}}(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I})$ z definice determinantu.

2) **metody využívající podobnosti matic**

   Tato kategorie metod využívá faktu, že podobné matice mají stejná vlastní čísla.

   Princip: konstruujeme posloupnost navzájem podobných matic, která konverguje k matici, jejíž vlastní čísla se dají jednoduchým způsobem určit.

3) **smíšené metody**

   založené na převodu obecné matice na matici třídiagonální (např. Givensova, Householderova a Lanczosova metoda) a následný efektivní výpočet kořenů charakteristického polynomu této upravené matice.

Metoda LU-rozkladu (**LR-transformace**, **LR-algoritmus**)

(Lower-Upper, Left-Right)

$\mathbf{A} = \mathbf{LU}$ ... rozklad matice $\mathbf{A}$ na dolní trojúhelníkovou matici $\mathbf{L}$ a horní trojúhelníkovou matici $\mathbf{U}$, kde na diagonále matice $\mathbf{L}$ jsou pro jednoznačnost rokladu jednotky.

Sestrojíme matici $\mathbf{B}$, která bude podobná matici $\mathbf{A}$.

$$\mathbf{B} = \mathbf{UL} \qquad (\mathbf{U} = \mathbf{L}^{-1}\mathbf{A} \quad \Rightarrow \quad \mathbf{B} = \mathbf{UL} = \mathbf{L}^{-1}\mathbf{AL}).$$

Postup:

Sestrojíme posloupnost matic $\mathbf{A}_k$:

(i) $\mathbf{A}_0 = \mathbf{A}, \quad k = 0$

(ii) provedeme LU rozklad matice $\mathbf{A}_k = \mathbf{L}_k\mathbf{U}_k$

(iii) sestrojíme matici $\mathbf{A}_{k+1} = \mathbf{U}_k\mathbf{L}_k$

(iv) je-li matice $\mathbf{A}_{k+1}$ horní trojúhelníková $\Rightarrow$ konec,
jinak $k = k+1$ a jdi na (ii)

Poznámka:

Dá se ukázat, že když matice $\mathbf{B}_k = \mathbf{L}_0\mathbf{L}_1\ldots\mathbf{L}_k$ konvergují k regulární matici, potom matice $\mathbf{A}_k$ také konvergují, a to k horní trojúhelníkové matici s vlastními čísly na diagonále. Platí

$$\mathbf{A}_{k+1} = \underbrace{\mathbf{L}_k^{-1}\mathbf{A}_k}_{\mathbf{U}_k}\mathbf{L}_k$$

a tedy

$$\mathbf{A}_{k+1} = \underbrace{\mathbf{L}_k^{-1}\mathbf{L}_{k-1}^{-1}\ldots\mathbf{L}_0^{-1}}_{\mathbf{B}_{k+1}^{-1}}\mathbf{A}_0\underbrace{\mathbf{L}_0\mathbf{L}_1\ldots\mathbf{L}_k}_{\mathbf{B}_{k+1}}$$

Poznámka:

Matice $\mathbf{B}_k$ konvergují k matici, jejíž sloupce tvoří vlastní vektory matice $\mathbf{A}$.
Pro symetrickou matici $\mathbf{A}$ je důkaz zřejmý

$$\mathbf{B}_{k+1}\underbrace{\mathbf{A}_{k+1}}_{\to\,\mathbf{\Lambda}} = \mathbf{A}\mathbf{B}_{k+1}.$$

Poznámka:

Je-li matice $\mathbf{A}$ symetrická a pozitivně definitní, provádíme LU-rozklad ve smyslu Choleského rozkladu ($\mathbf{A} = \mathbf{L}\mathbf{L}^T$). Potom lze ukázat, že $\mathbf{A}_k$ konverguje k diagonální matici.

Nevýhody:

- pomalá konvergence posloupnosti $\mathbf{A}_k$
- velký počet operací pro matice větších řádů
- nelze realizovat pro obecné matice $\mathbf{A}$

Metody ortogonálních transformací

Použijeme podobný princip jako v předchozím případě, tj. sestrojíme posloupnost navzájem podobných matic $\mathbf{A}_0$, $\mathbf{A}_1$, $\mathbf{A}_2$, ... tak, že

$$\boxed{\mathbf{A}_{k+1} = \mathbf{Q}_k^T\mathbf{A}_k\mathbf{Q}_k, \quad k = 0, 1, 2\ldots}$$

Požadujeme, aby posloupnost $\mathbf{A}_k$ konvergovala k matici, jejíž vlastní čísla lehce určíme. Ortogonální matici $\mathbf{Q}_k$ vybíráme speciálním postupem. Výhodou tohoto algoritmu je numerická stabilita.

Poznámka: Pro obecnou matici používáme metodu **QU-rozkladu** (**QR-transformace**).

$\mathbf{A} = \mathbf{Q}\mathbf{U}$ $\quad \mathbf{Q}\ldots$ ortogonální matice ($\mathbf{Q}\mathbf{Q}^\mathsf{T} = \mathbf{I}$, tj. $\mathbf{Q}^\mathsf{T} = \mathbf{Q}^{-1}$)
$\quad \mathbf{U}\ldots$ horní trojúhelníková matice

$\mathbf{B} = \mathbf{U}\mathbf{Q}$ $\quad (\mathbf{U} = \mathbf{Q}^{-1}\mathbf{A} \;\Rightarrow\; \mathbf{B} = \mathbf{Q}^{-1}\mathbf{A}\mathbf{Q} \;\Rightarrow\; \mathbf{B} = \mathbf{Q}^T\mathbf{A}\mathbf{Q})$.

Motivační příklad:

Příkladem ortogonální matice je matice rovinné rotace o úhel α:

$$\mathbf{Q}(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \overset{\text{označ}}{=} \begin{bmatrix} c & -s \\ s & c \end{bmatrix}.$$

Pro matici

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$$

stanovte matici $\mathbf{B} = \mathbf{Q}^T(\alpha)\mathbf{A}\mathbf{Q}(\alpha)$ tak, aby $b_{12} = 0$.

Řešení:

Rozepíšeme si prvky matice $\mathbf{B}$:

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}\cdot\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}\cdot\begin{bmatrix} c & -s \\ s & c \end{bmatrix} =$$

$$= \begin{bmatrix} 2c+s & c+3s \\ -2s+c & -s+3c \end{bmatrix}\cdot\begin{bmatrix} c & -s \\ s & c \end{bmatrix} =$$

$$= \begin{bmatrix} 2c^2+cs+cs+3s^2 & -2cs-s^2+c^2+3cs \\ -2cs+c^2-s^2+3cs & 2s^2-cs-cs+3c^2 \end{bmatrix}.$$

Pro splnění podmínky $b_{12} = 0$ musí platit

$$-2cs - s^2 + c^2 + 3cs = cs - s^2 + c^2 = 0,$$

tj.

$$\underbrace{\cos\alpha\sin\alpha}_{\frac{1}{2}\sin 2\alpha}\underbrace{-\sin^2\alpha+\cos^2\alpha}_{+\cos 2\alpha} = 0.$$

$$\cos 2\alpha = -\frac{1}{2}\sin 2\alpha$$

$$-2 = \tan 2\alpha$$

$$\underline{\alpha \doteq -0,5535}$$

Po dosazení dostaneme, že

$$\mathbf{B} = \begin{bmatrix} 3,6180 & 0 \\ 0 & 1,3819 \end{bmatrix}.$$

$\mathbf{B}$ je diagonální matice s vlastními čísly na diagonále a stejná vlastní čísla má i matice $\mathbf{A}$. □

Poznámka:

Podobně jako v předchozí metodě, pro dostatečně velké k je $\mathbf{A}_k$ horní trojúhelníková matice a vlastní vektory jsou (přibližně) sloupce matice $\mathbf{Q}_0\mathbf{Q}_1\dots\mathbf{Q}_{k-1}$.

Poznámka:

Pro symetrickou matici $\mathbf{A}$ vede uvedený postup na tzv. **metodu Jacobiovy diagonalizace**.

Jacobiova diagonalizace (speciální případ QR-transformace)

Věta: Je-li $\mathbf{A}$ reálná symetrická matice, potom existuje ortogonální matice $\mathbf{Q}$ tak, že

$$\boxed{\mathbf{Q}^T\mathbf{A}\mathbf{Q} = \mathbf{\Lambda}}$$

($\mathbf{\Lambda}$ ... spektrální matice = diagonální matice s vlastními čísly na diagonále).

Princip: Matici $\mathbf{Q}$ získáme součinem matic $\mathbf{Q}_{p,q}(\alpha)$, kde

$$\mathbf{Q}_{p,q}(\alpha) = \begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & \cos\alpha & \dots & -\sin\alpha & & & \\ & & & \vdots & 1 & \vdots & & & \\ & & & \vdots & & \vdots & & & \\ & & & \sin\alpha & \dots & \cos\alpha & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow p\text{-tý řádek} \\ \\ \\ \leftarrow q\text{-tý řádek} \\ \\ \\ \end{matrix}$$

$$\begin{matrix} \uparrow & & \uparrow \\ p\text{-tý sloupec} & & q\text{-tý sloupec} \end{matrix}$$

a parametr α volíme tak, abychom vynulovali prvek v pozici (p,q) a tedy i v pozici (q,p).

(D.cv. $\mathbf{Q}_{pq}^T(\alpha)\mathbf{Q}_{pq}(\alpha) = \mathbf{I}$.)

$$\mathbf{B} \overset{ozn.}{=} \boxed{\mathbf{A}_1 = \mathbf{Q}_{pq}^T(\alpha)\mathbf{A}\mathbf{Q}_{pq}(\alpha)}$$

$$b_{p,q} = \underbrace{\begin{bmatrix}\dots & \cos\alpha & \dots & \sin\alpha & \dots\end{bmatrix}}_{p\text{-tý řádek } \mathbf{Q}_{pq}^T} \mathbf{A} \underbrace{\begin{bmatrix}\dots & -\sin\alpha & \dots & \cos\alpha & \dots\end{bmatrix}^T}_{q\text{-tý sloupec } \mathbf{Q}_{pq}}$$

$$b_{p,q} = a_{pq}(\cos\alpha^2 - \sin\alpha^2) + (a_{qq} - a_{pp})\cos\alpha\sin\alpha$$

požadujeme, aby $b_{p,q} = 0$:

$$a_{pq}\cos 2\alpha + (a_{qq} - a_{pp})\frac{1}{2}\sin 2\alpha = 0 \qquad / \cdot 2$$

$$2a_{pq}\cos 2\alpha = -(a_{qq} - a_{pp})\sin 2\alpha \qquad / : \cos 2\alpha \qquad / : (a_{qq} - a_{pp})$$

$$\boxed{-\frac{2a_{pq}}{a_{qq} - a_{pp}} = \tan 2\alpha \qquad \Rightarrow \qquad \alpha = \dots}$$

Poznámka:

Při výpočtech nemusíme určovat úhel α, ale stačí nám vyjádřit $\sin\alpha$ a $\cos\alpha$. Lze odvodit vzorce pro

$$\sin\alpha = \dots \qquad \cos\alpha = \dots$$

Celkovou matici získáme takto $\boxed{\mathbf{Q} = \prod_{p,q}\mathbf{Q}_{p,q}(\alpha)}$ - postupně nulujeme všechny nediagonální prvky.

Poznámka:

Zbývá zvolit strategii na volbu indexů p a q. Nejjednodušší je postupně nulovat všechny mimodiagonální prvky (podobně jako v Gaussově eliminační metodě pro řešení soustavy lineárních rovnic). Uvědomme si ale, že se získané nuly z předchozího kroku obecně nezachovají. Další možností je nulovat vždy mimodiagonální prvek, který je největší v absolutní hodnotě (zde je třeba v každé iteraci vyhledat tento prvek, což zpomalí výpočet). Iterační proces zastavíme, je-li norma trojúhelníkové matice pod diagonálou menší než zadaná tolerance.

1. varianta - postupné nulování

2. varianta - nulování největšího prvku (v abs. hodnotě)

Vlastní vektory:

$$\mathbf{A}_1 = \mathbf{Q}_1^T\mathbf{A}\mathbf{Q}_1$$
$$\mathbf{A}_2 = \mathbf{Q}_2^T\mathbf{A}_1\mathbf{Q}_2$$
$$\vdots$$
$$\mathbf{A}_k = \mathbf{Q}_k^T\mathbf{A}_{k-1}\mathbf{Q}_k$$

$$\Rightarrow \qquad \mathbf{A}_k = \underbrace{\mathbf{Q}_k^T\mathbf{Q}_{k-1}^T\dots\mathbf{Q}_1^T}_{\mathbf{P}_k^T}\,\mathbf{A}\,\underbrace{\mathbf{Q}_1\mathbf{Q}_2\dots\mathbf{Q}_k}_{\mathbf{P}_k}$$

$$\mathbf{P}_k\underbrace{\mathbf{A}_k}_{(*)} = \mathbf{A}\underbrace{\mathbf{P}_k}_{(**)}$$

$(*)$ $\mathbf{A}_k \to \mathbf{\Lambda}$ $(k \to \infty)$ $\qquad$ $(**)$ $\mathbf{P}_k \to \mathbf{X}$ ... jejíž sloupce jsou vlastní vektory matice $\mathbf{A}$

Givensova transformace

- Slouží pro převod matice $\mathbf{A}$ na třídiagonální tvar (předpokládáme, že $\mathbf{A}$ je symetrická)
- Opět používáme podobnostní transformace

$$\mathbf{A}_{k+1} = \mathbf{Q}_k^T \mathbf{A}_k \mathbf{Q}_k$$

matice $\mathbf{Q}_k$ jsou opět maticemi rovinné rotace, ovšem jsou voleny tak, abychom zachovali již anulované prvky

$$\mathbf{Q}_1 = \begin{bmatrix} 1 & & & & & \\ & \frac{a_{21}}{d} & -\frac{a_{31}}{d} & & & \\ & \frac{a_{31}}{d} & \frac{a_{21}}{d} & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix} \qquad d = \sqrt{a_{21}^2 + a_{31}^2}$$

$\mathbf{A}_2 = \mathbf{Q}_1^T \mathbf{A} \mathbf{Q}_1$ bude mít 0 v pozici (3,1) a (1,3)

$$\mathbf{Q}_2 = \begin{bmatrix} 1 & & & & & \\ & \frac{a_{21}}{d} & & -\frac{a_{41}}{d} & & \\ & & 1 & & & \\ & \frac{a_{41}}{d} & & \frac{a_{21}}{d} & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix} \qquad d = \sqrt{a_{21}^2 + a_{41}^2}$$

$\mathbf{A}_3 = \mathbf{Q}_2^T \mathbf{A}_2 \mathbf{Q}_2$ bude mít 0 v pozici (4,1) a (1,4)

---

atd.

Příklad: Převeďte matici $\mathbf{A}$ na třídiagonální tvar.

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 3 \\ 4 & 2 & 6 \\ 3 & 6 & 5 \end{bmatrix}$$

$a_{21} = 4, \quad a_{31} = 3, \quad d = \sqrt{4^2 + 3^2} = 5$

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

$$\mathbf{Q}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{4}{5} & -\frac{3}{5} \\ 0 & \frac{3}{5} & \frac{4}{5} \end{bmatrix}$$

$\mathbf{Q}_1^T \mathbf{Q}_1 = \mathbf{I}$?

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{4}{5} & -\frac{3}{5} \\ 0 & \frac{3}{5} & \frac{4}{5} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{4}{5} & \frac{3}{5} \\ 0 & -\frac{3}{5} & \frac{4}{5} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Q}_1^T \mathbf{A} \mathbf{Q}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{4}{5} & -\frac{3}{5} \\ 0 & \frac{3}{5} & \frac{4}{5} \end{bmatrix} \begin{bmatrix} 1 & 4 & 3 \\ 4 & 2 & 6 \\ 3 & 6 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{4}{5} & \frac{3}{5} \\ 0 & -\frac{3}{5} & \frac{4}{5} \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 4 & 3 \\ 5 & \frac{26}{5} & \frac{39}{5} \\ 0 & \frac{18}{5} & \frac{2}{5} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{4}{5} & \frac{3}{5} \\ 0 & -\frac{3}{5} & \frac{4}{5} \end{bmatrix} = \begin{bmatrix} 1 & 5 & 0 \\ 5 & \frac{221}{25} & \frac{78}{25} \\ 0 & \frac{78}{25} & -\frac{46}{25} \end{bmatrix}$$

Efektivní výpočet hodnoty charakteristického polynomu pro třídiagonální matici

$$\mathbf{A} = \begin{bmatrix} a_1 & c_1 & & & & \\ b_2 & a_2 & c_2 & & & \\ & b_3 & a_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & c_{n-1} \\ & & & & b_n & a_n \end{bmatrix}$$

$f_{-1}(\lambda) = 0$

$f_0(\lambda) = 1$

$f_k(\lambda) = (a_k - \lambda)\, f_{k-1}(\lambda) \; - \; b_k c_{k-1}\, f_{k-2}(\lambda) \qquad k = 1, 2, \ldots n$

$f_n(\lambda) = p_{\mathbf{A}}(\lambda)$

$$\underbrace{\mathbf{A}-\lambda\mathbf{I}}_{=\mathbf{M}} = \begin{bmatrix} a_1-\lambda & c_1 & & & & \\ b_2 & a_2-\lambda & c_2 & & & \\ & b_3 & a_3-\lambda & c_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & b_{n-1} & a_{n-1}-\lambda & c_{n-1} \\ & & & & b_n & a_n-\lambda \end{bmatrix}$$

rozvoj podle posledního řádku:

$$\det \mathbf{M} = (a_n-\lambda)\det(\mathbf{M}_{n-1}) - b_n c_{n-1}\det(\mathbf{M}_{n-2})$$

($\mathbf{M}_{n-1}$ ... prvních $n-1$ řádků a sloupců z $\mathbf{M}$)

$$\mathbf{M}_1 = a_1-\lambda = \det(\mathbf{M}_1)$$

$$\mathbf{M}_0 = 1$$

$$\mathbf{M}_{-1} = 0$$

Podstata výpočtu vlastních čísel třídiagonální matice pomocí jednoduchého vyjadřování hodnoty charakteristického polynomu metodou bisekce:

Příklad:

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

Zvolíme interval $\langle 0,5\rangle$ ... v něm očekáváme všechna vlastní čísla

Výpočtem snadno určíme

$\boxed{f_3(0) = p_{\mathbf{A}}(0) = 4}$

$f_{-1}(0) = 0$

$f_0(0) = 1$

$f_1(0) = (\underbrace{a_1}_{=2} - 0)\underbrace{f_0(0)}_{=1} - b_1 c_0 \underbrace{f_{-1}(0)}_{=0} = 2$

$f_2(0) = (\underbrace{a_2}_{=2} - 0)\underbrace{f_1(0)}_{=2} - \underbrace{b_2}_{=-1}\underbrace{c_1}_{=-1}\underbrace{f_0(0)}_{=1} = 3$

$f_3(0) = (\underbrace{a_3}_{=2} - 0)\underbrace{f_2(0)}_{=3} - \underbrace{b_3}_{=-1}\underbrace{c_2}_{=-1}\underbrace{f_1(0)}_{=2} = 4$

$\boxed{f_3(5) = p_{\mathbf{A}}(5) = -21}$

$f_{-1}(5) = 0$

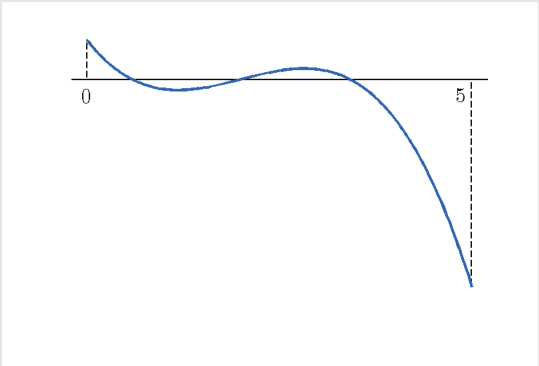
$f_0(5) = 1$

$f_1(5) = (\underbrace{a_1}_{=2} - 5)\underbrace{f_0(5)}_{=1} - b_1 c_0 \underbrace{f_{-1}(5)}_{=0} = -3$

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

$f_2(5) = (\underbrace{a_2}_{=2} - 5)\underbrace{f_1(5)}_{=-3} - \underbrace{b_2}_{=-1}\underbrace{c_1}_{=-1}\underbrace{f_0(5)}_{=1} = 8$

$f_3(5) = (\underbrace{a_3}_{=2} - 5)\underbrace{f_2(5)}_{=8} - \underbrace{b_3}_{=-1}\underbrace{c_2}_{=-1}\underbrace{f_1(5)}_{=-3} = -21$



Vypočteme střed intervalu, $s = \frac{0+5}{2} = 2{,}5$ a určíme $f_3(2{,}5)$ ...

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, $\ln x$, ...) 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 $\varphi_0,\varphi_1,\dots\varphi_n$ 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.

Numerické metody KMA/NM
Josef Daněk 13.2.2013

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:

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

(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:

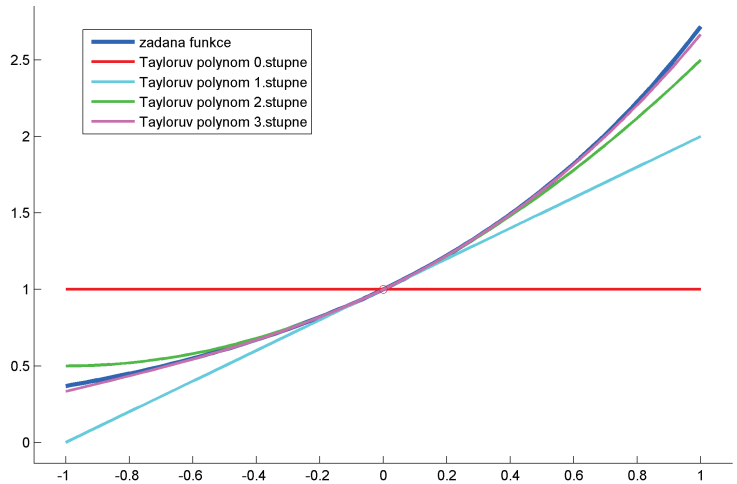
$$\text{Platí-li } |f^{(n+1)}(x)|\le M\quad\forall x\in\mathcal{U}(x_0),\quad \text{potom } |e(x)|\le\frac{M}{(n+1)!}|x-x_0|^{n+1}.$$

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

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

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

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



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 $2\pi \quad \Rightarrow \quad$ odhad pro $f^{(n+1)}(x)$ je $\underbrace{\left|f^{(n+1)}(x)\right|}_{(*)} \leq 1 \quad \forall x \in \langle -\pi, \pi \rangle$

$(*)$ 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 $T_n(x)$?

$f(x)$	$= \sin x$	$f(0)$	$= 0$
$f'(x)$	$= \cos x$	$f'(0)$	$= 1$
$f''(x)$	$= -\sin x$	$f''(0)$	$= 0$
$f'''(x)$	$= -\cos x$	$f'''(0)$	$= -1$
$f^{(4)}(x)$	$= \sin x$	$f^{(4)}(0)$	$= 0$
$\vdots$		$\vdots$	

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

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

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.

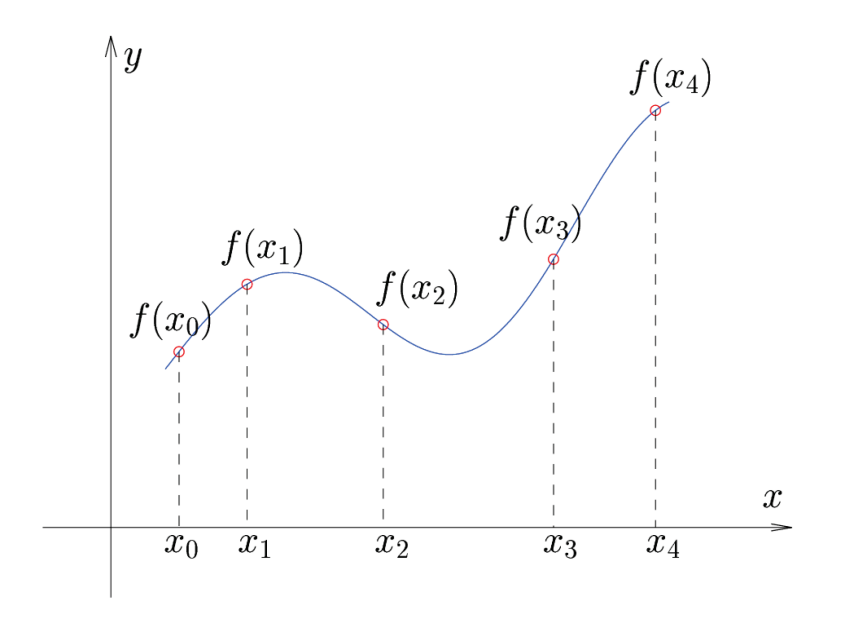
Poznámka:

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

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

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

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

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

Numerické metody
Josef Daněk
KMA/NM
13.2.2013

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

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 $\det \mathbf{V} \neq 0$.

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 $(x_j - x_0)$, $j = 1,2,\ldots,n$.

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

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

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

$$(*) = \frac{x_2^3 - x_0^3}{x_2 - x_0} - \frac{x_1^3 - x_0^3}{x_1 - x_0} = (x_2^2 + x_2x_0 + x_0^2) - (x_1^2 + x_1x_0 + x_0^2) = x_2^2 - x_1^2 + x_0(x_2 - x_1) = (x_2 - x_1)(x_2 + x_1 + x_0)$$

$$(**) = \frac{x_3^3 - x_0^3}{x_3 - x_0} - \frac{x_1^3 - x_0^3}{x_1 - x_0} = (x_3^2 + x_3x_0 + x_0^2) - (x_1^2 + x_1x_0 + x_0^2) = x_3^2 - x_1^2 + x_0(x_3 - x_1) = (x_3 - x_1)(x_3 + x_1 + x_0)$$

$$\vdots$$

Vynormujeme – $(j+1)$-ní řádek vydělíme $(x_j - x_1)$, $j = 2, 3, \ldots, n$.

$$\mathbf{V} \sim \begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 & \ldots & x_0^n \\ 0 & 1 & x_1 + x_0 & x_1^2 + x_1x_0 + x_0^2 & \ldots & \ldots \\ 0 & 0 & 1 & x_2 + x_1 + x_0 & \ldots & \ldots \\ 0 & 0 & 1 & x_3 + x_1 + x_0 & \ldots & \ldots \\ \vdots & \vdots & \vdots & & & \\ 0 & 0 & 1 & x_n + x_1 + x_0 & \ldots & \ldots \end{bmatrix}$$

$$\vdots$$

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 $l_i(x)$.

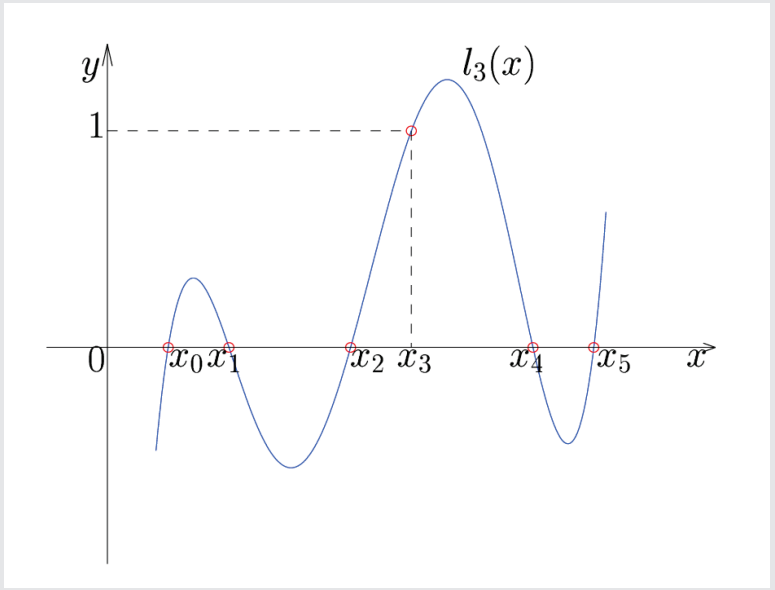
Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

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

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

Opět požadujeme splnění interpolačních podmínek

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

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

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

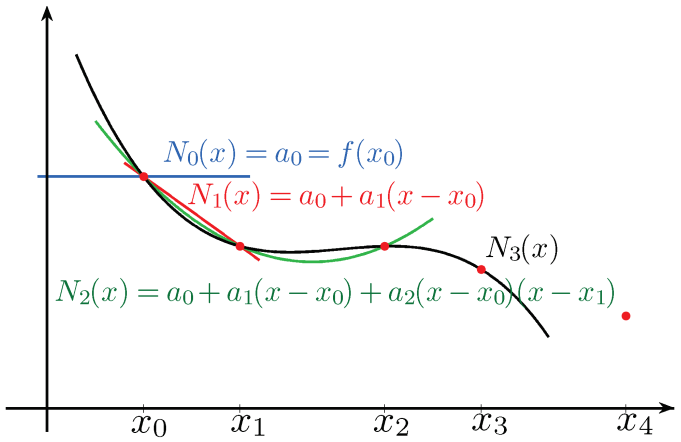
Výstup: $\boxed{a_i = \mathbf{A}_{ii}}$

Schéma algoritmu

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

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.

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



$N_0(x) = a_0$ 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 $[x_2, f(x_2)]$.

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

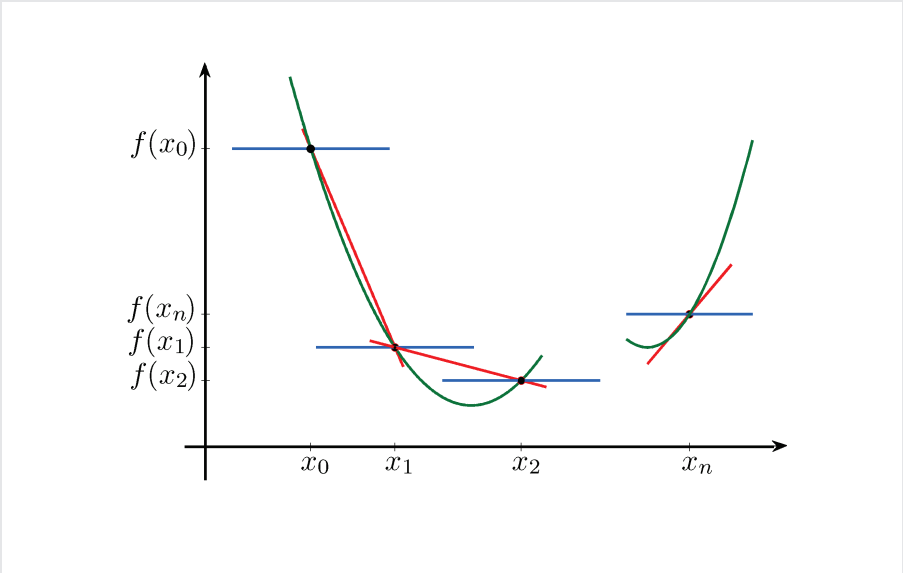
$(*)$ prochází $[x_0, f(x_0)]$

$(**)$ 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?

x_0	$f(x_0) = A_{00}$			
x_1	$f(x_1) = A_{10}$	A_{11}		
x_2	$f(x_2) = A_{20}$	A_{21}	A_{22}	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	
x_n	$f(x_n) = A_{n0}$	A_{n1}	A_{n2}	



Věta Má-li funkce f, k níž přísluší data $f(x_i)$, $i = 0, 1, \ldots, n$, spojité derivace v nějakém intervalu $\langle a, b\rangle \supset \operatorname{int}(x_0, x_1, \ldots, x_n)$ 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:

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

(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 $\left|f^{(n+1)}(x)\right| \le M$, pak

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

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

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

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

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

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

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

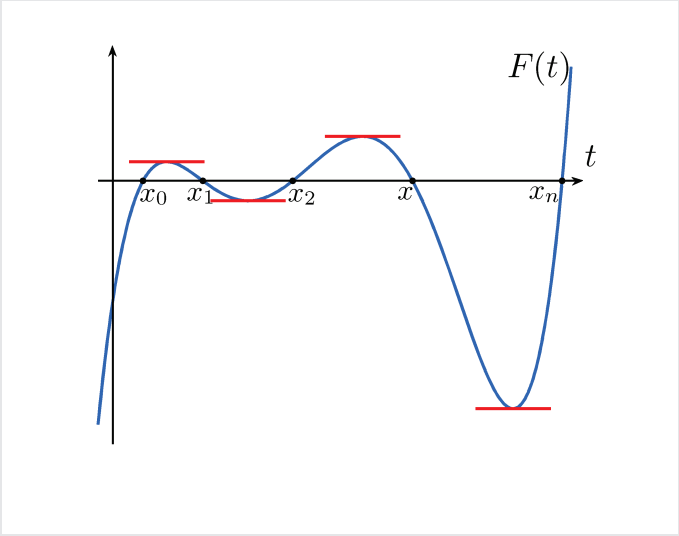
$$F(x_i) = \underbrace{f(x_i) - P_n(x_i)}_{=0} - \frac{f(x) - P_n(x)}{w_{n+1}(x)} \underbrace{w_{n+1}(x_i)}_{=0} = 0$$

- 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:

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

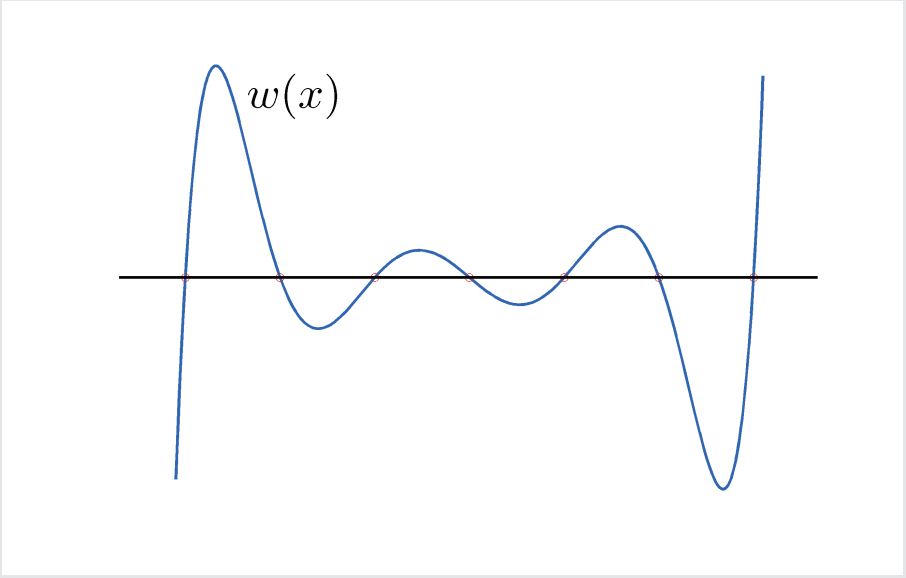
(•) Víme, že existuje ξ tak, že $F^{(n+1)}(\xi)=0$
(••) P_n ... polynom n-tého stupně
(•••) $w_{n+1}(t)$ je polynom $(n+1)$ stupně a u t^{n+1} je koeficient 1

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

□

- 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 → 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 ♡) 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
  $\Leftrightarrow$ $\lim_{n\to\infty} \sup_{x\in\langle a,b\rangle} |f_n(x)-f(x)| = 0$

Čebyševovy polynomy

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

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

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

Důkaz:

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

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

Platí:

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

Kořeny polynomů jsou

$$\cos(n\arccos x) = 0$$
$$n\arccos x_k = \frac{\pi}{2}+k\pi, \quad k\in\mathbb{Z}$$

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

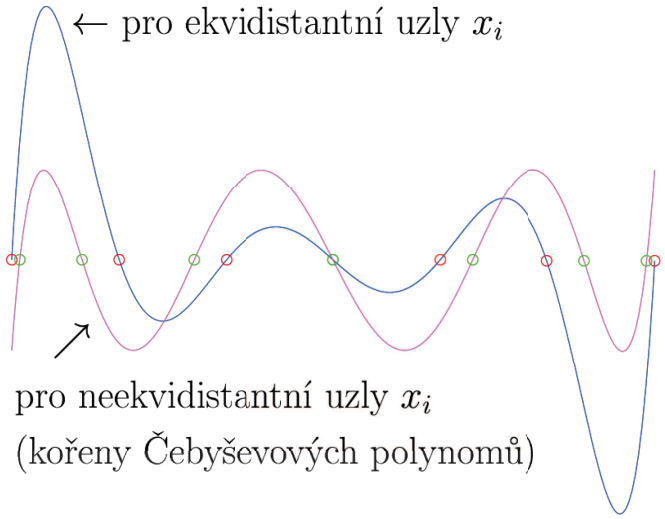
$$\boxed{x_k = \cos\left(\frac{2k+1}{n}\cdot\frac{\pi}{2}\right),\quad k=0,1,\ldots,n-1} \qquad \heartsuit$$

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

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

$$w(x) = (x-x_0)(x-x_1)\ldots(x-x_N)$$

← pro ekvidistantní uzly x_i



pro neekvidistantní uzly x_i
(kořeny Čebyševových polynomů)

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,

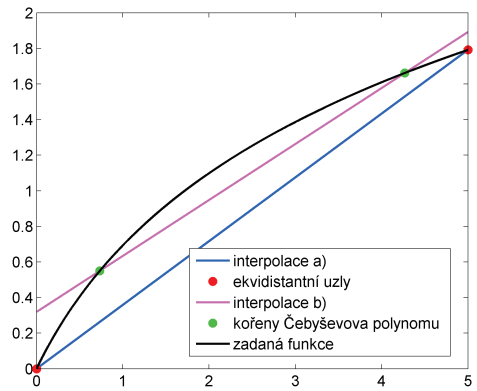
b) kořeny Čebyševových polynomů.

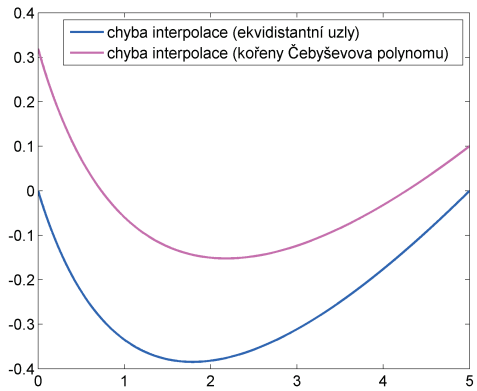
Porovnejte chybu získaných interpolačních polynomů.

---

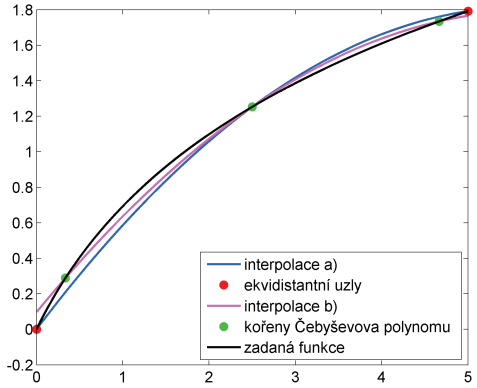
1) $n=2$

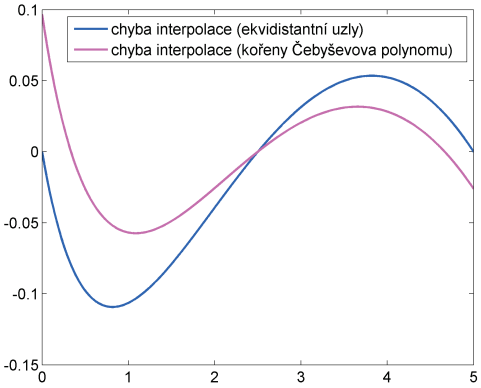
Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



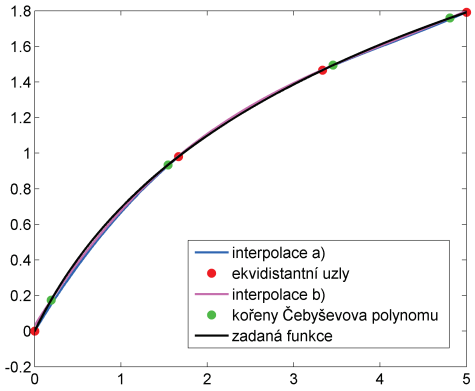


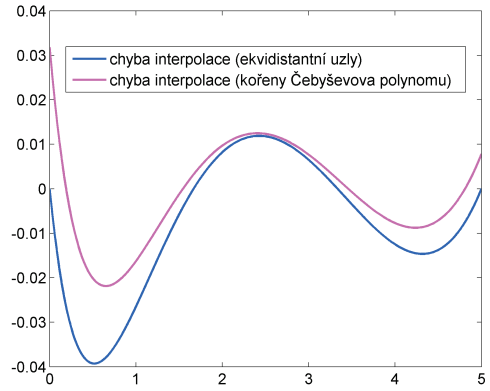
2) $n=3$



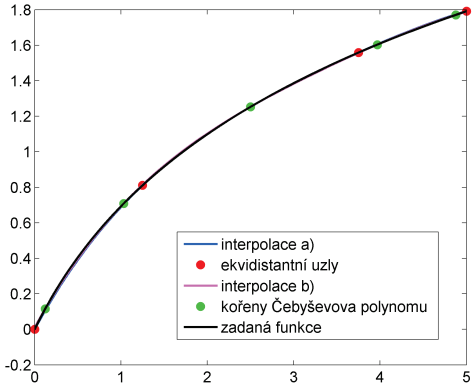


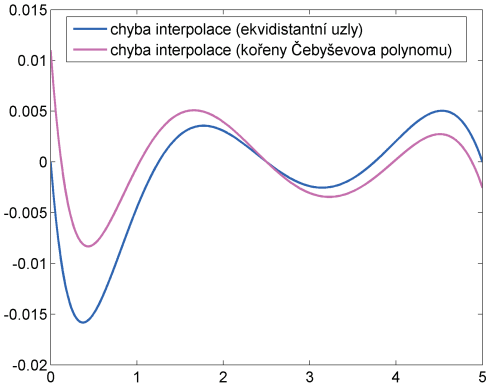
3) $n=4$



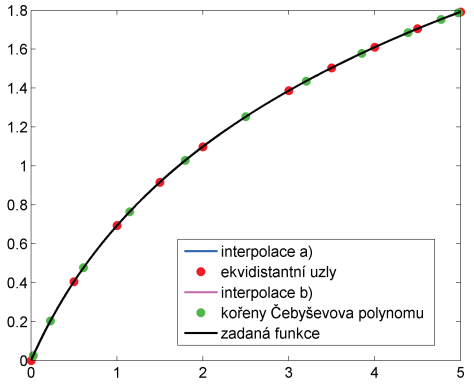


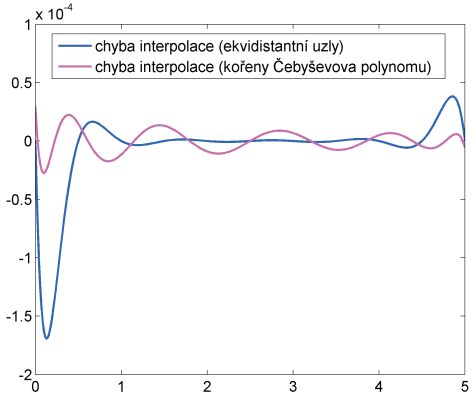
4) $n = 5$



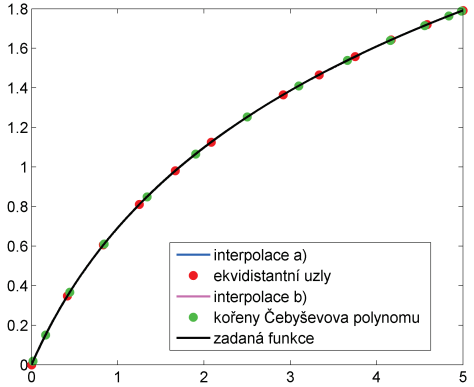


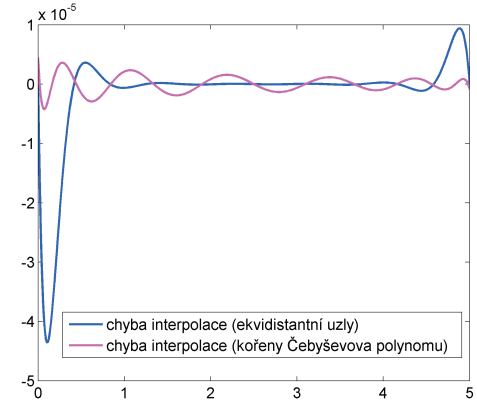
5) $n = 11$



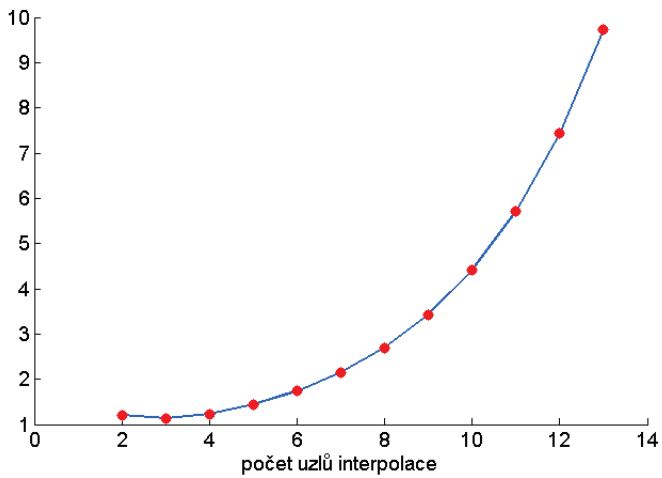


6) $n = 13$



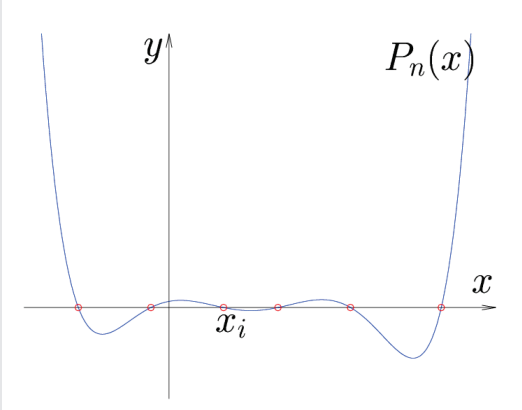


Následující obrázek ukazuje poměr maximové normy chyby interpolačního polynomu vypočteného pro hodnoty v ekvidistantních uzlech ku maximové normě chyby interpolačního polynomu vypočteného pro hodnoty v uzlech určených jakožto kořeny Čebyševových polynomů.



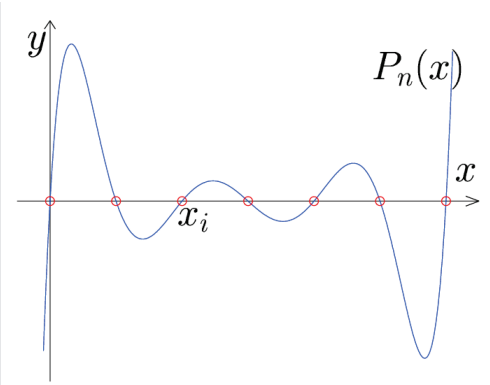
Poznámka:

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.



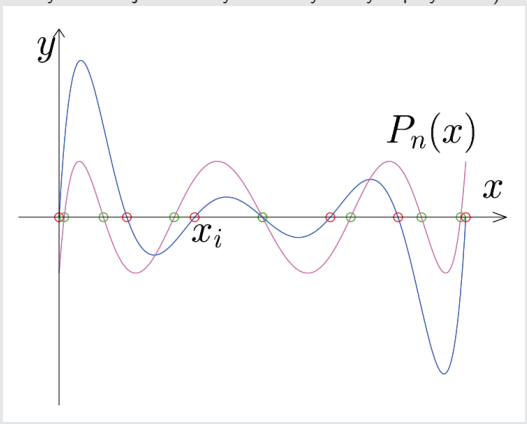
Poznámka:

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



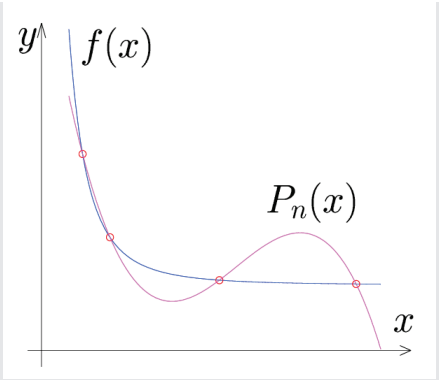
Poznámka:

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

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

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	$8+1{,}5\,\dfrac{8-64}{2-4} = \mathbf{50}$		
-1,5	5	125	$125-1{,}5\,\dfrac{125-8}{5-2} = 66{,}5$	$66{,}5-1{,}5\,\dfrac{66{,}5-50}{5-4} = \mathbf{41{,}75}$	
2,5	1	1	$1+2{,}5\,\dfrac{1-125}{1-5} = 78{,}5$	$78{,}5+2{,}5\,\dfrac{78{,}5-66{,}5}{1-2} = 48{,}5$	$48{,}5+2{,}5\,\dfrac{48{,}5-78{,}5}{1-4} = \mathbf{42{,}875}$

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 následujících 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
```

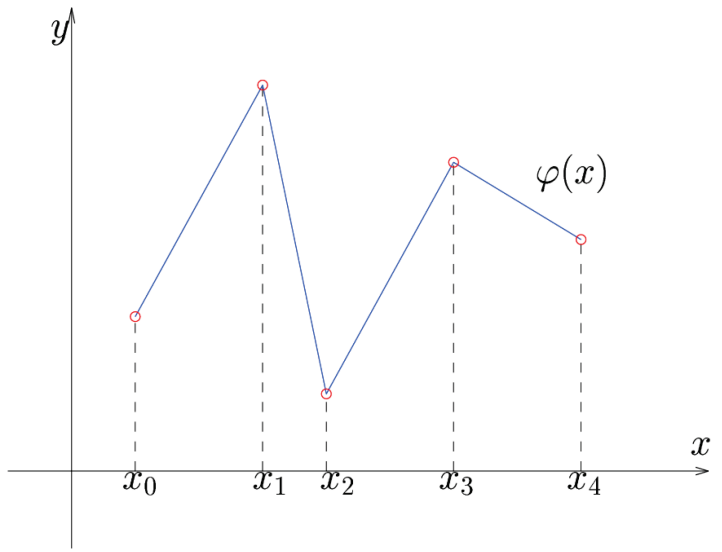
Poznámka

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.

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



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.

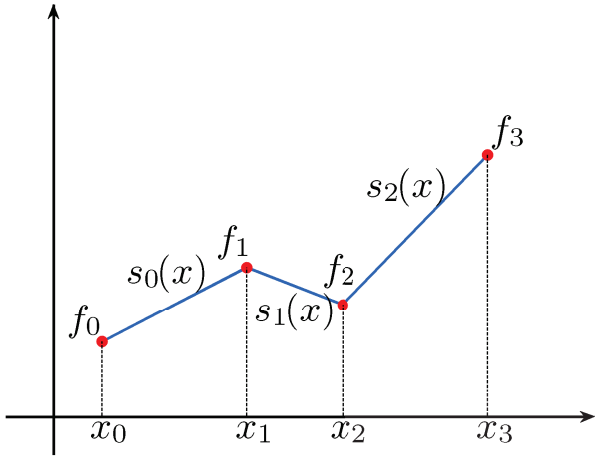
$$s(x) = s_i(x),\ x \in \langle x_i, x_{i+1} \rangle, \text{ kde } \boxed{s_i(x) = a_i + b_i(x - x_i)}$$

(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

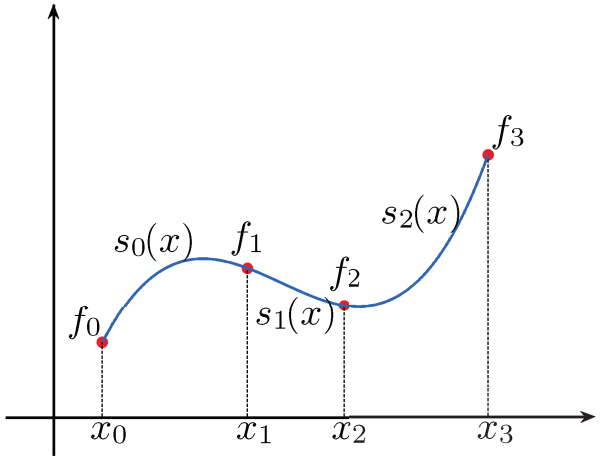
(iii) představuje $(n-1)$ podmínek

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

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.

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



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

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

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

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

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

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

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

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

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

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

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 $T=x_n-x_0$)

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

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

(C) podmínky tečen

$$\boxed{s'(x_0)=y'_0,\quad s'(x_n)=y'_n},\quad \text{kde } y'_0, y'_n \text{ jsou daná čísla}$$

(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.

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

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 $h_i=x_{i+1}-x_i,\quad i=0,1,\dots,n-1$)

(ii) interpolační podmínky

$$f_i=a_i,\quad i=0,1,\dots,n-1$$
$$f_n=a_{n-1}+b_{n-1}h_{n-1}+\frac{c_{n-1}}{2}h_{n-1}^2+\frac{d_{n-1}}{6}h_{n-1}^3$$

(iii) spojitost

$$f_{i+1}=a_i+b_ih_i+\frac{c_i}{2}h_i^2+\frac{d_i}{6}h_i^3,\quad i=0,1,\dots,n-2$$
$$\left[s_{i+1}(x_{i+1})=s_i(x_{i+1})\right]$$

(iv) spojitost derivace

$$b_i+c_ih_i+\frac{d_i}{2}h_i^2=b_{i+1},\quad i=0,1,\dots,n-2$$
$$\left[s'_i(x_{i+1})=s'_{i+1}(x_{i+1})\right]$$

Numerické metody KMA/NM
Josef Daněk 13.2.2o13

(v) spojitost 2. derivace

$$c_i+d_ih_i=c_{i+1},\quad =0,1,\dots,n-2$$
$$\left[s''_i(x_{i+1})=s''_{i+1}(x_{i+1})\right]$$

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.

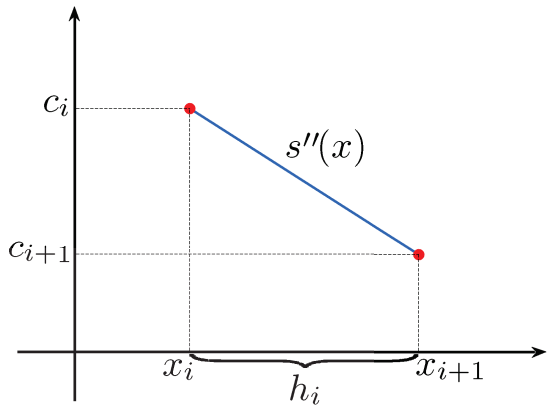
Soustavu řešíme GEM pro třídiagonální matice.

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

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

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

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

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

Pro $x = x_{i+1}$

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

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

Ze spojitosti první derivace $\boxed{s_i'(x_{i+1}^-) = s_{i+1}'(x_{i+1}^+),\ i = 0,1,\ldots,n-2}$ 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

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

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

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

$J(f)$ ... 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:

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

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:

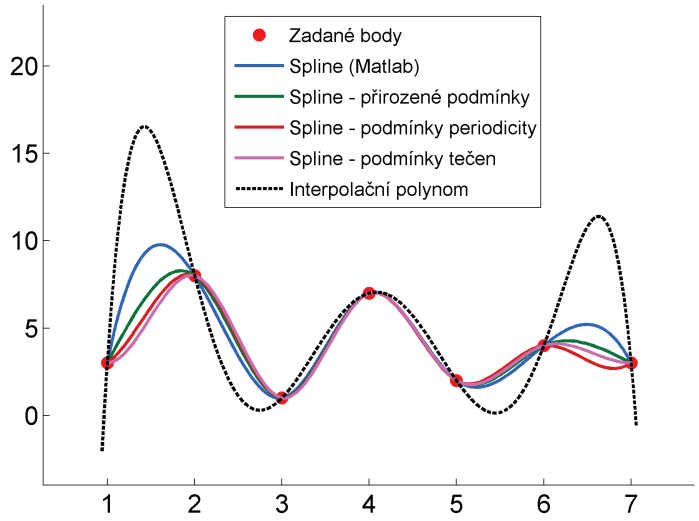
$$|f(x) - s(x)| \le c_1 K h^4$$
$$|f'(x) - s'(x)| \le c_2 K h^3$$
$$|f''(x) - s''(x)| \le c_3 K h^2.$$

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 $f'(7) = 0$.



Numerické metody
Josef Daněk
KMA/NM
13.2.2013

Kapitola 8. Aproximace funkcí - II

Aproximace funkcí

- Aproximace na okolí bodu - aproximujeme chování funkce „v malém okolí bodu“ ✓
- Interpolace - tabulkou danými body prokládáme 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.

- určíme systém jednoduchých **základních (bázových) funkcí** (ne nutně polynomů) $\varphi_0, \varphi_1, \varphi_2, \ldots, \varphi_n$ a funkci f aproximujeme lineární kombinací základních funkcí

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

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

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

Úvod d diskrétní L_2-aproximace

Myšlenka

Chceme aproximovat funkci, která je dána tabulkou $\{[x_i, f(x_i)], i = 0, 1, \ldots, n\}$.

V případě, kdy jsou $f(x_i)$ zatíženy chybou (např. výsledky měření) nebo pokud je bodů „mnoho“, není vhodné provádět interpolaci.

Aproximaci φ hledáme ve tvaru

$$\varphi(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \cdots + c_m\varphi_m(x),$$

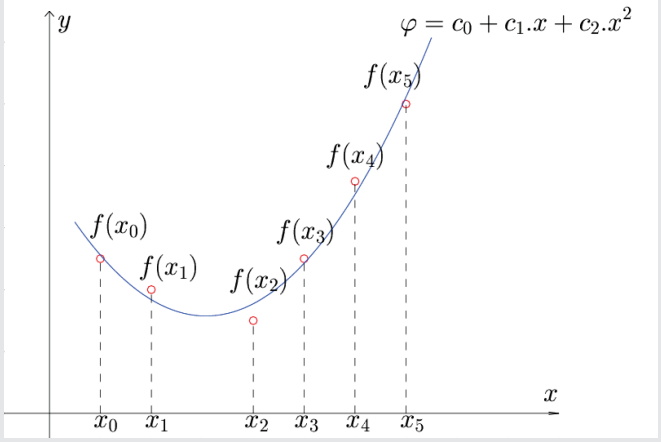
kde φ_i jsou zadané funkce a c_i hledané parametry.

– počet bázových funkcí φ_i je menší než počet zadaných bodů $(m < n)$

– v případě rovnosti se může jednat o interpolaci (záleží na zvolených bázových funkcích)

– o interpolaci se může jednat i pokud je $m < n$

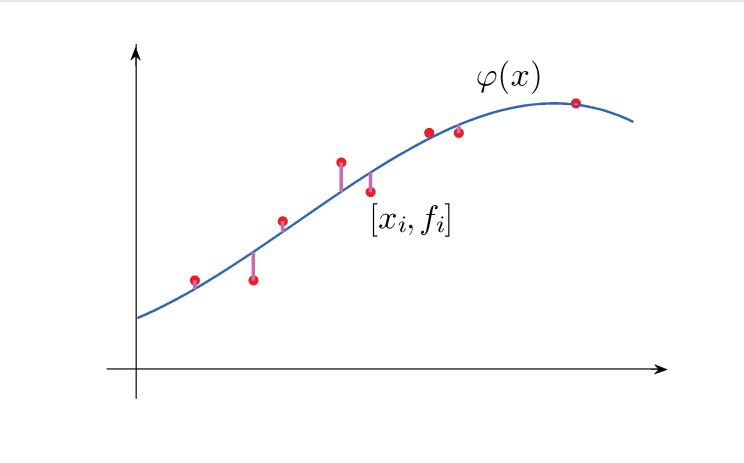
Naším cílem je minimalizovat „odchylku“ funkce φ od zadaných dat.



U interpolace jsme požadovali, aby aproximace přímo procházela zadanými body, tj. chyba

$$\boxed{e_i = f(x_i) - \varphi(x_i) = 0}$$

Nyní na tomto netrváme, pouze chceme tuto chybu v nějakém smyslu minimalizovat.



Jakou použít normu pro měření chyby e ?

- $\max\limits_{0\le i\le n} \{|f(x_i) - \varphi(x_i)|\}$
- $\dfrac{1}{n+1}\sum\limits_{i=0}^{n} |f(x_i) - \varphi(x_i)|$
- $\sqrt{\dfrac{1}{n+1}\sum\limits_{i=0}^{n} |f(x_i) - \varphi(x_i)|^2}$

Cílem je chybu minimalizovat $\Rightarrow$ vybereme tu normu, která umožní nejsnažší postup.

Uvažujme příklad:

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

; $\varphi(x) = c_0 + c_1 x$

Jak by vypadala minimalizace s užitím předchozích norem?

- $\min\limits_{c_0,c_1\in\mathbb{R}} \max\{|1 - c_0 - c_1|, |2 - c_0 - 2c_1|, |2 - c_0 - 3c_1|\}$ ... pro počítání nevhodné
- $\min\limits_{c_0,c_1\in\mathbb{R}} \frac{1}{3}(|1 - c_0 - c_1| + |2 - c_0 - 2c_1| + |2 - c_0 - 3c_1|)$ ... opět nevhodné
- $\min\limits_{c_0,c_1\in\mathbb{R}} \sqrt{\frac{1}{3}[(1 - c_0 - c_1)^2 + (2 - c_0 - 2c_1)^2 + (2 - c_0 - 3c_1)^2]}$ zjednodušíme

(Nezáporná funkce f nabývá svého minima ve stejném bodě jako nabývá minima funkce $\sqrt{f}$)

$$\min_{c_0,c_1\in\mathbb{R}} \underbrace{\left[(1 - c_0 - c_1)^2 + (2 - c_0 - 2c_1)^2 + (2 - c_0 - 3c_1)^2\right]}_{(*)}$$

$(*)$ ... kvadratická funkce proměnných c_0, c_1 (konvexní) $\Rightarrow$ je hladká, snadno se derivuje

Formulace

Je dána funkce f tabulkou hodnot v $n+1$ bodech $x_0, x_1, \ldots, x_n$

x_i	...
$f(x_i)$	...

.

Zvolíme tvar aproximující funkce

$$\boxed{\varphi(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \cdots + c_m\varphi_m(x)}$$

s počtem parametrů c_i nejvýše rovným $n+1$.

Diskrétní L_2-aproximací funkce f je potom taková lineární kombinace bázových funkcí $\varphi_i(x)$, jejíž koeficienty splňují podmínku, že L_2 norma chyby je minimální, tj.

$$\boxed{R(f,\varphi) = \sum_{i=0}^{n} [f(x_i) - \varphi(x_i)]^2 = \sum_{i=0}^{n} \left[f(x_i) - \sum_{j=0}^{m} c_j\varphi_j(x_i)\right]^2}$$

je minimální.

Poznámka:

Tato nejlepší aproximace má velmi dobré statistické vlastnosti a vyrovnává vliv náhodných chyb v zadaných (naměřených) funkčních hodnotách.

Diskrétní L_2-aproximace lineárním polynomem

Úkolem je stanovit diskrétní L_2-aproximaci funkce f dané tabulkou $\{x_i, f_i\}$, $i = 0, 1, \ldots, n$ lineárním polynomem, tj. zvolíme např. $\varphi_0(x) = 1$, $\varphi_1(x) = x$.

Tedy

$$\boxed{\varphi(x) = c_0 + c_1 x}.$$

Minimalizujeme funkci

$$\boxed{R = \sum_{i=0}^{n} [f_i - c_0 - c_1 x_i]^2}.$$

Nutná a postačující podmínka minima je

$$\frac{\partial R}{\partial c_0} = -2\sum_{i=0}^{n}[f_i - c_0 - c_1 x_i] = 0$$

$$\frac{\partial R}{\partial c_1} = -2\sum_{i=0}^{n}[f_i - c_0 - c_1 x_i]\, x_i = 0$$

Koeficienty c_0 a c_1 nalezneme jako řešení soustavy

$$(n+1)\, c_0 + \Big(\sum_{i=0}^{n} x_i\Big) c_1 = \sum_{i=0}^{n} f_i$$

$$\Big(\sum_{i=0}^{n} x_i\Big) c_0 + \Big(\sum_{i=0}^{n} x_i^2\Big) c_1 = \sum_{i=0}^{n} f_i x_i$$

Maticově zapsáno

$\boxed{\mathbf{Pc} = \mathbf{q}}$, kde $\mathbf{P}$ je symetrická, pozitivně definitní.

Jiný postup:

Užijeme metodu pro řešení „neřešitelných soustav".
Platí (mělo by):

$$\boxed{c_0 + c_1 x_i = f_i, \quad i = 0, 1, \ldots, n}$$

$$\begin{bmatrix} 1 & x_0 \\ 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}$$

Maticově zapsáno

$$\boxed{\mathbf{Qc} = \mathbf{F}}.$$

Soustava $\mathbf{Qc} = \mathbf{F}$ je neřešitelná. Provádíme minimalizaci $\underline{\mathbf{r}^T\mathbf{r}}$, kde $\underline{\mathbf{r} = \mathbf{F} - \mathbf{Qc}}$ je reziduum soustavy.
Dosadíme-li, pak platí:

$$\mathbf{r}^T\mathbf{r} = (\mathbf{F} - \mathbf{Qc})^T(\mathbf{F} - \mathbf{Qc}) = \mathbf{F}^T\mathbf{F} - \mathbf{c}^T\mathbf{Q}^T\mathbf{F} - \mathbf{F}^T\mathbf{Qc} + \mathbf{c}^T\mathbf{Q}^T\mathbf{Qc} = \mathbf{F}^T\mathbf{F} - 2\mathbf{c}^T\mathbf{Q}^T\mathbf{F} + \mathbf{c}^T\mathbf{Q}^T\mathbf{Qc},$$

protože $\underbrace{(\mathbf{c}^T\mathbf{Q}^T\mathbf{F})^T}_{\text{číslo}} = \underbrace{(\mathbf{F}^T\mathbf{Qc})}_{\text{číslo}}$.

Matice $\mathbf{Q}^T\mathbf{Q}$ je symetrická, pozitivně definitní. Nutná a postačující podmínka minima:

$$\mathbf{Q}^T\mathbf{Qc} - \mathbf{Q}^T\mathbf{F} = 0 \quad \Rightarrow \quad \boxed{\mathbf{Q}^T\mathbf{Qc} = \mathbf{Q}^T\mathbf{F}}$$

tzv. **soustava normálních rovnic**.
Platí:

$\boxed{\mathbf{P} = \mathbf{Q}^T\mathbf{Q}}$ a $\boxed{\mathbf{q} = \mathbf{Q}^T\mathbf{F}}$.

Diskrétní L_2-aproximace kvadratickým polynomem

Funkci f aproximujeme kvadratickým polynomem

$$\boxed{\varphi(x) = c_0 + c_1 x + c_2 x^2}.$$

Minimalizujeme veličinu

$$\boxed{R = \sum_{i=0}^{n}\left[f_i - c_0 - c_1 x_i - c_2 x_i^2\right]^2}.$$

Z nutných a postačujících podmínek minima

$$\boxed{\frac{\partial R}{\partial c_0} = 0, \quad \frac{\partial R}{\partial c_1} = 0, \quad \frac{\partial R}{\partial c_2} = 0}$$

dostaneme soustavu ve tvaru

$$(n+1)c_0 + \Big(\sum x_i\Big) c_1 + \Big(\sum x_i^2\Big) c_2 = \sum f_i$$

$$\Big(\sum x_i\Big) c_0 + \Big(\sum x_i^2\Big) c_1 + \Big(\sum x_i^3\Big) c_2 = \sum f_i x_i$$

$$\Big(\sum x_i^2\Big) c_0 + \Big(\sum x_i^3\Big) c_1 + \Big(\sum x_i^4\Big) c_2 = \sum f_i x_i^2$$

Stejnou soustavu dostaneme i postupem, kdy řešíme neřešitelnou soustavu pomocí minimalizace kvadrátu rezidua.

Poznámka: V případě, že některé hodnoty chceme eliminovat, např. důsledkem špatného měření, je vhodné použít váhy, tj. minimalizujeme

$$R(f, \varphi, w_i) = \sum_{i=0}^{n}[f(x_i) - \varphi(x_i)]^2 w_i,$$

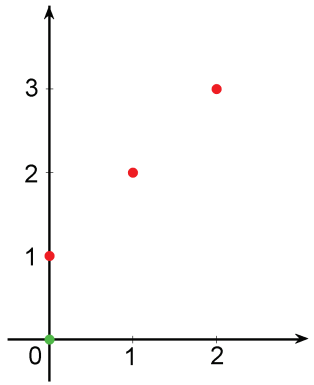
kde w_i ... váha uzlu x_i.

Příklad

Aproximujte funkci f, která je dána tabulkou

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

pomocí funkce $\boxed{\varphi(x) = c_0 x + c_1 x^2 + c_2 x^3}$.



$$\mathbf{Qc} = \mathbf{F}, \qquad \mathbf{Q} \ \ldots \text{ singulární matice}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 4 & 8 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

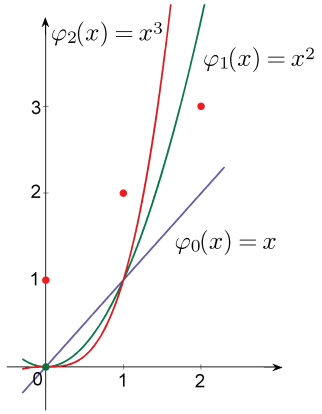
- první rovnici nelze splnit $\Rightarrow$ soustava nemá řešení
- řešíme metodou nejmenších čtverců

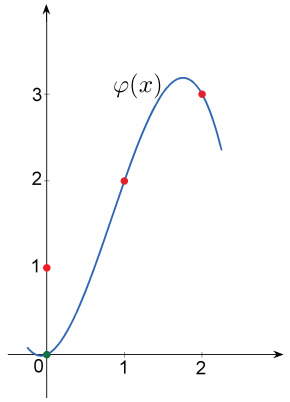
$$\mathbf{Q}^T./ \qquad \mathbf{Qc} = \mathbf{F} \quad \rightarrow \quad \mathbf{Q}^T\mathbf{Qc} = \mathbf{Q}^T\mathbf{F}$$

$$\begin{bmatrix} 5 & 9 & 17 \\ 9 & 17 & 33 \\ 17 & 33 & 65 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 8 \\ 14 \\ 26 \end{bmatrix}$$

$$\Rightarrow \qquad c_0 = 0{,}4, \quad c_1 = 2{,}65, \quad c_2 = -1{,}05$$

$$\boxed{\varphi(x) = 0{,}4\,x + 2{,}65\,x^2 - 1{,}05\,x^3}$$





$$\varphi_0(0) = \varphi_1(0) = \varphi_2(0) = 0 \qquad \Rightarrow \qquad \varphi(0) = 0$$

Řešitelnost úlohy diskrétní L_2-aproximace

Definice: Řekneme, že systém funkcí $\varphi_j(x)$, $j = 0, 1, \ldots, m$, definovaných na $\langle a, b\rangle \supset \text{int}\,(x_0, x_1, \ldots, x_n)$ je **diskrétně lineárně nezávislý**, jsou-li vektory

$$[\varphi_0(x_0), \varphi_0(x_1), \ldots, \varphi_0(x_n)]^T$$
$$[\varphi_1(x_0), \varphi_1(x_1), \ldots, \varphi_1(x_n)]^T$$
$$\vdots$$
$$[\varphi_m(x_0), \varphi_m(x_1), \ldots, \varphi_m(x_n)]^T$$

lineárně nezávislé.

Poznámka:

Tato definice říká, že hodnost matice $\Phi = [\varphi_j(x_i)]_{j=0,1,\ldots,m}^{i=0,1,\ldots,n}$ je rovna $(m+1)$.

Platí, že $m \leq n$.

Podmíněnost úlohy diskrétní L_2-aproximace

Budeme-li aproximovat na intervalu $\langle 0, 1\rangle$ funkci f a zvolíme-li ekvidistantní dělení a bázové funkce budeme volit $\varphi_j = x^j$, bude matice $\mathbf{P} = \mathbf{Q}^T\mathbf{Q}$ soustavy normálních rovnic blízká Hilbertově matici, která je velmi špatně podmíněná.

Řešení: Za funkce $\varphi_j(x)$ volíme ortogonální polynomy (např. Gramovy polynomy).

Poznámka: Ze systému n-lineárně nezávislých funkcí g_i lze pomocí Gram-Schmidtova ortogonalizačního

procesu zkonstruovat systém ortogonálních funkcí.

Spojitá L_2-aproximace

Definice:

Mějme funkci $w = w(x)$, která je definována na $\langle a, b\rangle$ a je kladná a omezená. Množinu reálných funkcí $f = f(x)$ definovaných na (a, b) takových, že

$$\int_a^b w(x)\big[f(x)\big]^2\,dx < \infty$$

označíme $L_2(a, b)$. Skalárním součinem dvou funkcí $f, g \in L_2(a, b)$ nazýváme číslo

$$(f, g) = \int_a^b w(x) f(x) g(x)\,dx.$$

Číslo

$$||f|| = \sqrt{(f, f)} = \left(\int_a^b w(x)\big[f(x)\big]^2\,dx\right)^{\frac{1}{2}}$$

nazýváme normou funkce f v $L_2(a, b)$.

Funkce f, g se nazývají ortogonální, platí-li

$$(f, g) = 0.$$

Chceme tedy minimalizovat veličinu

$$R(f, \varphi) = \left(f - \sum_{j=0}^{m} c_j\varphi_j, f - \sum_{j=0}^{m} c_j\varphi_j\right).$$

Nutné a postačující podmínky minima mají tvar

$$\frac{\partial R}{\partial c_k} = 0, \quad k = 0, 1, \ldots, m.$$

Derivováním a jednoduchými úpravami dostaneme

$$R(f, \varphi) = (f, f) - 2(f, \sum_{j=0}^{m} c_j\varphi_j) + (\sum_{i=0}^{m} c_i\varphi_i, \sum_{j=0}^{m} c_j\varphi_j)$$

$$\frac{\partial R}{\partial c_k} = 0 - 2(f, \varphi_k) + 2(\varphi_k, \sum_{j=0}^{m} c_j\varphi_j) = 0$$

$$\sum_{j=0}^{m} (\varphi_k, c_j\varphi_j) = (f, \varphi_k)$$

$$\sum_{j=0}^{m} c_j(\varphi_k, \varphi_j) = (f, \varphi_k)$$

Zapsáním všech podmínek dostaneme soustavu

$$(\varphi_0, \varphi_0)c_0 + (\varphi_0, \varphi_1)c_1 + \cdots + (\varphi_0, \varphi_m)c_m = (\varphi_0, f)$$
$$(\varphi_1, \varphi_0)c_0 + (\varphi_1, \varphi_1)c_1 + \cdots + (\varphi_1, \varphi_m)c_m = (\varphi_1, f)$$
$$\vdots$$
$$(\varphi_m, \varphi_0)c_0 + (\varphi_m, \varphi_1)c_1 + \cdots + (\varphi_m, \varphi_m)c_m = (\varphi_m, f)$$

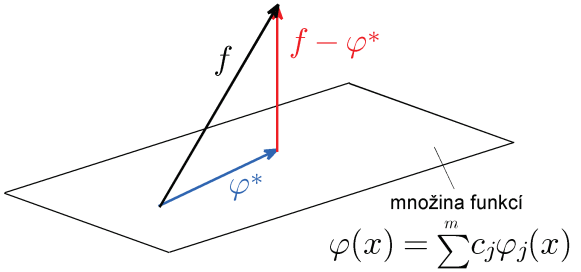
Věta

Jsou-li funkce $\varphi_0, \varphi_1, \ldots, \varphi_m$ lineárně nezávislé, má úloha spojité L_2-aproximace jediné řešení. Koeficienty c_j^* jsou řešením normální soustavy a platí:

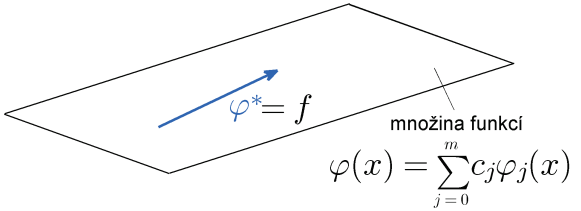
$$(f - \varphi^*, \varphi_j) = 0, \quad j = 0, 1, \ldots, m,$$

tj. funkce $f - \varphi^*$ je ortogonální ke všem funkcím φ_j.

Geometrická interpretace



Pokud leží funkce f v množině funkcí $\varphi(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \cdots + c_m\varphi_m(x)$, potom



Příklad

Stanovte spojitou L_2-aproximaci funkce $f(x) = \ln x$ na $\langle 1, \mathrm{e}\rangle$ lineární funkcí $\varphi(x) = c_1x + c_0$.

Minimalizujeme funkci

$$r(c_0,c_1)=\int_1^{\mathrm{e}}|f(x)-\varphi(x)|^2\,dx=\int_1^{\mathrm{e}}(\ln x-c_0-c_1x)^2\,dx$$

Podmínky minima

$$\frac{\partial r}{\partial c_0}=-2\int_1^{\mathrm{e}}(\ln x-c_0-c_1x)\,dx=0$$

$$\frac{\partial r}{\partial c_1}=-2\int_1^{\mathrm{e}}(\ln x-c_0-c_1x)x\,dx=0$$

$$c_0\int_1^{\mathrm{e}}1\,dx+c_1\int_1^{\mathrm{e}}x\,dx=\int_1^{\mathrm{e}}\ln x\,dx$$

$$c_0\int_1^{\mathrm{e}}x\,dx+c_1\int_1^{\mathrm{e}}x^2\,dx=\int_1^{\mathrm{e}}x\ln x\,dx$$

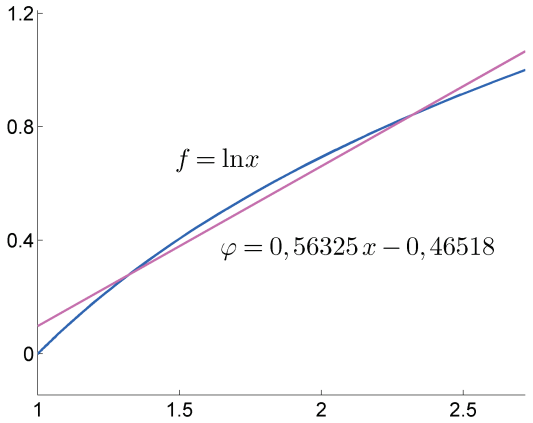
$$\begin{bmatrix}\mathrm{e}-1 & \frac{1}{2}(\mathrm{e}^2-1)\\ \frac{1}{2}(\mathrm{e}^2-1) & \frac{1}{3}(\mathrm{e}^3-1)\end{bmatrix}\begin{bmatrix}c_0\\ c_1\end{bmatrix}=\begin{bmatrix}1\\ \frac{\mathrm{e}^2+1}{4}\end{bmatrix}$$

$$\int_1^{\mathrm{e}}\underbrace{\ln x}_{v}\underbrace{1}_{u'}\,dx=\left|\begin{matrix}v=\ln x & v'=\frac{1}{x}\\ u'=1 & u=x\end{matrix}\right|=\left[x\ln x\right]_1^{\mathrm{e}}-\int_1^{\mathrm{e}}1\,dx=\mathrm{e}-\mathrm{e}+1=1$$

$$\int_1^{\mathrm{e}}\underbrace{x}_{u'}\underbrace{\ln x}_{v}\,dx=\left|\begin{matrix}v=\ln x & v'=\frac{1}{x}\\ u'=x & u=\frac{x^2}{2}\end{matrix}\right|=\left[\frac{x^2}{2}\ln x\right]_1^{e}-\int_1^{\mathrm{e}}\frac{x^2}{2}\frac{1}{x}\,dx=\frac{e^2}{2}-\frac{1}{4}(e^2-1)=\frac{e^2}{4}+\frac{1}{4}$$

Řešení: $c_0\doteq-0,46518,\ c_1\doteq0,56325$

$$\boxed{\varphi(x)=0,56325\,x-0,46518}.$$



Podmíněnost úlohy spojité L_2-aproximace

Příklad: Volíme-li váhu $w(x)\equiv1$ a aproximujeme-li funkci $f=f(x)$ na intervalu $\langle0,1\rangle$ funkcí $\varphi=\varphi(x)$ ve tvaru

$$\varphi(x)=c_0+c_1x+c_2x^2+\cdots+c_mx^m,$$

tj. $\boxed{\varphi_j(x)=x^j}$, platí

$$(\varphi_i,\varphi_j)=\int_0^1x^ix^j\,dx=\frac{1}{i+j+1}.$$

Soustava normálních rovnic $\mathbf{Pc}=\mathbf{g}$ je opět špatně podmíněná, protože $\mathbf{P}$ je Hilbertova matice.

Řešení problému: Volíme funkce $\varphi_j(x),\quad j=0,1,\ldots,m$, ortogonální ve smyslu skalárního součinu

$$(\varphi_i,\varphi_j)=\int_a^bw(x)\varphi_i(x)\varphi_j(x)\,dx.$$

Potom platí: $(\varphi_i,\varphi_j)=0$ pro $i\neq j$ a soustava normálních rovnic má diagonální matici. Pak lze psát:

$$\boxed{c_j^*=\frac{(f,\varphi_j)}{(\varphi_j,\varphi_j)},\quad j=0,1,\ldots,m}$$

$c_j^*\ldots$ **Fourierovy koeficienty**.

Gram-Schmidtův ortogonalizační proces

Jsou dány lineárně nezávislé funkce $g_1,g_2,\ldots,g_n$ (prvky jistého prostoru).

Hledáme funkce (prvky téhož prostoru), které jsou navzájem po dvou ortogonální.

$\boxed{f_1=g_1}$

f_2 hledáme ve tvaru $\boxed{f_2 = g_2 + \kappa_{21} f_1}$ a použijeme $(f_1, f_2) = 0$

$$\underbrace{(f_2, f_1)}_{=0} = (g_2, f_1) + \kappa_{21}(f_1, f_1) \quad \Rightarrow \quad \boxed{\kappa_{21} = -\frac{(g_2, f_1)}{(f_1, f_1)}}$$

f_3 hledáme ve tvaru $\boxed{f_3 = g_3 + \kappa_{31} f_1 + \kappa_{32} f_2}$ a použijeme $(f_3, f_1) = 0$ a $(f_3, f_2) = 0$

$$\underbrace{(f_3, f_1)}_{=0} = (g_3, f_1) + \kappa_{31}(f_1, f_1) + \kappa_{32}\underbrace{(f_2, f_1)}_{=0}$$

$$\Rightarrow \quad \boxed{\kappa_{31} = -\frac{(g_3, f_1)}{(f_1, f_1)}}$$

$$\underbrace{(f_3, f_2)}_{=0} = (g_3, f_2) + \kappa_{31}\underbrace{(f_1, f_2)}_{=0} + \kappa_{32}(f_2, f_2)$$

$$\Rightarrow \quad \boxed{\kappa_{32} = -\frac{(g_3, f_2)}{(f_2, f_2)}}$$

Obecně f_k hledáme ve tvaru $\boxed{f_k = g_k + \kappa_{k1} f_1 + \kappa_{k2} f_2 + \cdots + \kappa_{k,k-1} f_{k-1}}$

a $\boxed{\kappa_{kj} = -\frac{(g_k, f_j)}{(f_j, f_j)}}$

$j = 1, 2, \ldots, k-1$

Příklad

Najděte ortogonální bázi prostoru polynomů do stupně 2 na $\langle 0, 10 \rangle$.

Vyjdeme z báze $g_0 = 1$, $g_1 = x$, $g_2 = x^2$.

$\boxed{f_0 = 1}$

$f_1 = x + \varkappa_{10} f_0$

$$\varkappa_{10} = -\frac{(x,1)}{(1,1)} = -\frac{\int_0^{10} x\,dx}{\int_0^{10} 1\,dx} = -\frac{50}{10}$$

$\boxed{f_1 = x - 5}$

$f_2 = x^2 + \varkappa_{20} 1 + \varkappa_{21}(x-5)$

$$\varkappa_{20} = -\frac{(x^2,1)}{(1,1)} = -\frac{\int_0^{10} x^2\,dx}{\int_0^{10} 1\,dx} = -\frac{\frac{1000}{3}}{10} = -33{,}\overline{3}$$

$$\varkappa_{21} = -\frac{(x^2, x-5)}{(x-5, x-5)} = -\frac{\int_0^{10} x^3 - 5x^2\,dx}{\int_0^{10} (x-5)^2\,dx} = -\frac{\frac{10000}{4} - 5\cdot\frac{1000}{3}}{\frac{5^3}{3} + \frac{5^3}{3}} = -\frac{30000 - 20000}{250 \cdot 4} = -10$$

$f_2 = x^2 - 33{,}\overline{3} - 10(x-5)$

$\boxed{f_2 = x^2 - 10x + 16{,}\overline{6}}$

Příklad

Určete ortogonální bázi prostoru polynomů do stupně 2 pro uzlové body
$x_0 = 0$; $x_1 = 0{,}2$; $x_2 = 0{,}4$; $x_3 = 0{,}6$; $x_4 = 0{,}8$; $x_5 = 1$

Opět použijeme jako výchozí bázi

$g_0 = 1$ tj. $[1; 1; 1; 1; 1; 1]$
$g_1 = x$ tj. $[0; 0{,}2; 0{,}4; 0{,}6; 0{,}8; 1]$
$g_2 = x^2$ tj. $[0; 0{,}04; 0{,}16; 0{,}36; 0{,}64; 1]$

$\boxed{f_0 = g_0 = 1}$

$f_1 = g_1 + \varkappa_{10} f_0$

$$\varkappa_{10} = -\frac{(g_1, f_0)}{(f_0, f_0)} = -\frac{0 + 0{,}2 + 0{,}4 + 0{,}6 + 0{,}8 + 1}{6} = -\frac{3}{6} = -\frac{1}{2}$$

$\boxed{f_1 = x - \frac{1}{2}}$

$f_2 = g_2 + \varkappa_{20} f_0 + \varkappa_{21} f_1$

$$\varkappa_{20} = -\frac{(g_2, f_0)}{(f_0, f_0)} = -\frac{0 + 0{,}04 + 0{,}16 + 0{,}36 + 0{,}64 + 1}{6} = -\frac{2{,}2}{6} = -\frac{1{,}1}{3}$$

$$\varkappa_{21} = -\frac{(g_2, f_1)}{(f_1, f_1)} = -\frac{[0; 0{,}04; 0{,}16; 0{,}36; 0{,}64; 1]^T\,[-0{,}5; -0{,}3; -0{,}1; 0{,}1; 0{,}3; 0{,}5]}{0{,}25 + 0{,}09 + 0{,}01 + 0{,}01 + 0{,}09 + 0{,}25} =$$
$$= -\frac{0 - 0{,}012 - 0{,}016 + 0{,}036 + 0{,}192 + 0{,}5}{0{,}7} = -\frac{0{,}7}{0{,}7} = -1$$

$f_2 = x^2 - \frac{1{,}1}{3} - x + \frac{1}{2}$

$\boxed{f_2 = x^2 - x + \frac{2}{15}} = x^2 - x + 0{,}1\overline{3}$

Poznámka

Pokud bychom zvolili jiné uzlové body x_i, dostali bychom i obecně jiný systém ortogonálních bázových funkcí.

Např. pro $x_0 = 0$; $x_1 = 0{,}125$; $x_2 = 0{,}25$; $x_3 = 0{,}375$; $x_4 = 0{,}5$; $x_5 = 0{,}625$; $x_6 = 0{,}75$; $x_7 = 0{,}875$; $x_8 = 1$

bychom získali následující ortogonální bázi prostoru polynomů do stupně 2:

$\boxed{f_0 = 1}$, $\boxed{f_1 = x - \frac{1}{2}}$, $\boxed{f_2 \doteq x^2 - x + 0{,}1458}$ ... Ověřte (D.cv.).

Poznámka:

Uvažujme úlohu (spojité) L_2-aproximace, kde za aproximující funkci volíme polynom stupně n.

Jak máme volit stupeň polynomu ?

Pokud nemáme další informace, je vhodné řešit normální rovnice postupně pro $m = 0, 1, 2, \ldots$ a sledovat hodnotu

$$\boxed{\sigma_m^2 = \frac{\sum_{i=0}^{n} (f(x_i) - \varphi(x_i))^2}{n - m}},$$

kde φ ... polynom stupně m.

Pokud σ_m^2 s rostoucím m významně klesá, pokračujeme, jinak hodnota po níž nenásleduje výrazný pokles σ_m^2 je ze statistických důvodů vhodným stupněm polynomu.

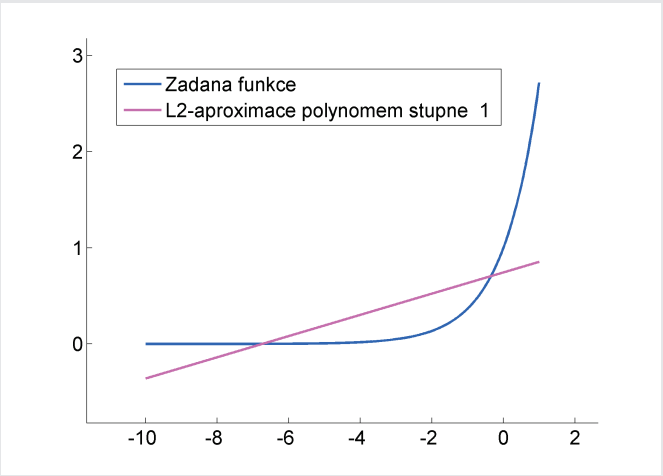
Volíme-li za $\varphi_j = x^j$

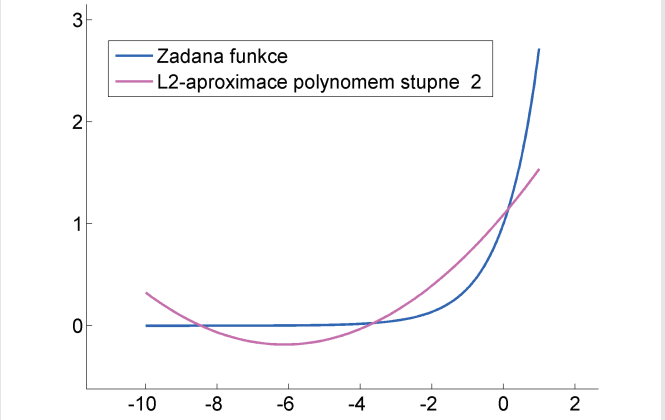
1. musíme řešit soustavu normálních rovnic pro každý stupeň m znovu,
2. špatná podmíněnost

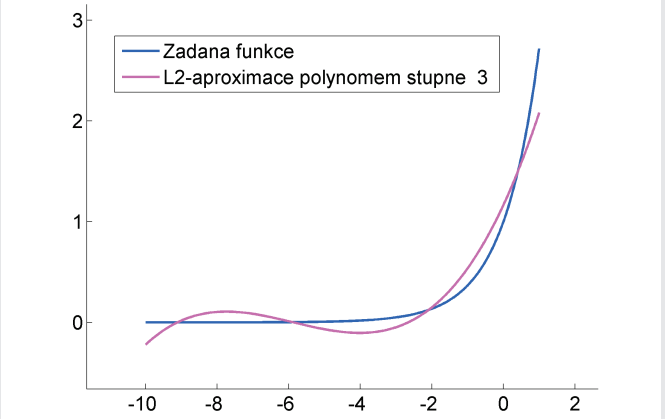
Řešení: Použijeme ortogonální polynomy $\rightarrow$ potom stačí vždy dopočítat pouze 1 koeficient.

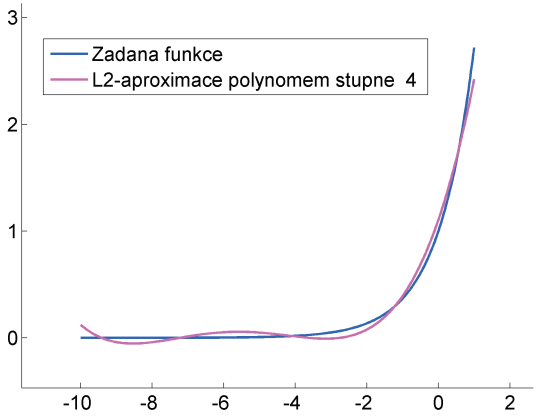
Příklad

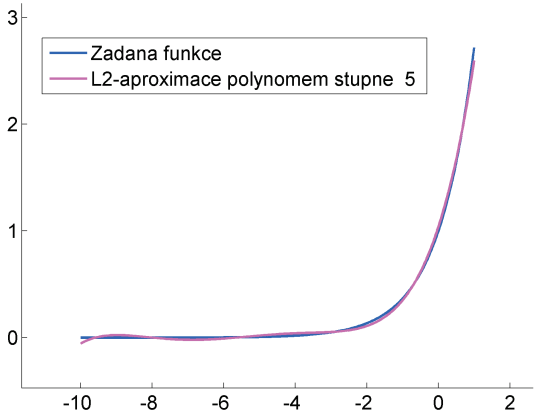
Určete spojitou L_2-aproximaci funkce $f(x) = e^x$ na $\langle -10; 1 \rangle$ pomocí polynomů stupně nejvýše 1, 2, 3, 4 a 5.











Nelineární aproximace metodou nejmenších čtverců

Např. volíme-li $\boxed{\varphi(x) = Ce^{Ax}}$ $(*)$

- 1. přístup je metoda linearizace dat: $(*)$ zlogaritmujeme

$$\underbrace{\ln \varphi}_{\Phi} = \underbrace{\ln C}_{B} + Ax$$

$$\boxed{\Phi = Ax + B}$$

(původní body $[x_i, f_i]$ je třeba transformovat na body $[x_i, \ln f_i]$)

získáme A, B a z B vypočteme $C = e^B$.

- 2. přístup minimalizujeme L_2 normu chyby přímo

$$R(A,C) = \sum_{i=0}^{n} \left(f_i - Ce^{Ax_i}\right)^2$$

parciální derivace:

$$\frac{\partial R}{\partial A} = 2\sum_{i=0}^{n} \left(f_i - Ce^{Ax_i}\right)\left(Cx_i e^{Ax_i}\right) = 0$$

$$\frac{\partial R}{\partial C} = 2\sum_{i=0}^{n} \left(f_i - Ce^{Ax_i}\right)\left(e^{Ax_i}\right) = 0$$

soustava normálních rovnic:

1. rovnice:

$$\sum_{i=0}^{n} \left(f_i - Ce^{Ax_i}\right)\left(Cx_i e^{Ax_i}\right) = 0$$

$$C\sum_{i=0}^{n} f_i x_i e^{Ax_i} - C^2 \sum_{i=0}^{n} x_i e^{2Ax_i} = 0 \qquad / \cdot \frac{1}{C} \neq 0$$

$$\boxed{\sum_{i=0}^{n} f_i x_i e^{Ax_i} - C\sum_{i=0}^{n} x_i e^{2Ax_i} = 0}$$

2. rovnice:

$$\sum_{i=0}^{n} \left(f_i - Ce^{Ax_i}\right)\left(e^{Ax_i}\right) = 0$$

$$\boxed{\sum_{i=0}^{n} f_i e^{Ax_i} - C\sum_{i=0}^{n} e^{2Ax_i} = 0}$$

... soustava nelineárních rovnic, pro řešení lze použít např. Newtonovu metodu.

Příklad

Určete diskétní L_2-aproximaci funkce f zadané tabulkou

x_i	0	1	2	3	4
$f(x_i)$	1,5	2,5	3,5	5	7,5

funkcí ve tvaru $\boxed{\varphi(x) = Ce^{Ax}}$.

Pro řešení použijeme oba předchozí přístupy.

---

1. přístup

script v MATLABu

```
x=0:4
f=[1.5 2.5 3.5 5 7.5]
F=log(f)'
Q=[x.^0' x.^1']
P=Q'*Q
g=Q'*F
koef=P\g
A=koef(2)
C=exp(koef(1))
```

výsledky v MATLABu

```
x =
     0     1     2     3     4
f =
    1.5000    2.5000    3.5000    5.0000    7.5000
F =
    0.4055
    0.9163
    1.2528
    1.6094
    2.0149
Q =
     1     0
     1     1
     1     2
     1     3
     1     4
P =
     5    10
    10    30
g =
    6.1989
   16.3097
koef =
    0.4574
    0.3912
A =
    0.3912
C =
    1.5799
```

$$\Rightarrow \quad \varphi_1(x) \doteq 1{,}5799\,\mathrm{e}^{0{,}3912\,x}$$

2. přístup

script v MATLABu

Numerické metody
Josef Daněk
KMA/NM
13.2.2013

```
koef=fminsearch('R',[1 1]);
AA=koef(1)
CC=koef(2)

%-------------------------------------------------------------
function out=R(koef);
 A=koef(1);
 C=koef(2);
 out=(C-1.5).^2+(C.*exp(A)-2.5).^2+(C.*exp(2*A)-3.5).^2+ ...
     (C.*exp(3*A)-5).^2+(C.*exp(4*A)-7.5).^2;
```

výsledky v MATLABu

```
AA =
    0.3836
CC =
    1.6109
```

$$\Rightarrow \quad \varphi_2(x) \doteq 1{,}6109\,\mathrm{e}^{0{,}3836\,x}$$

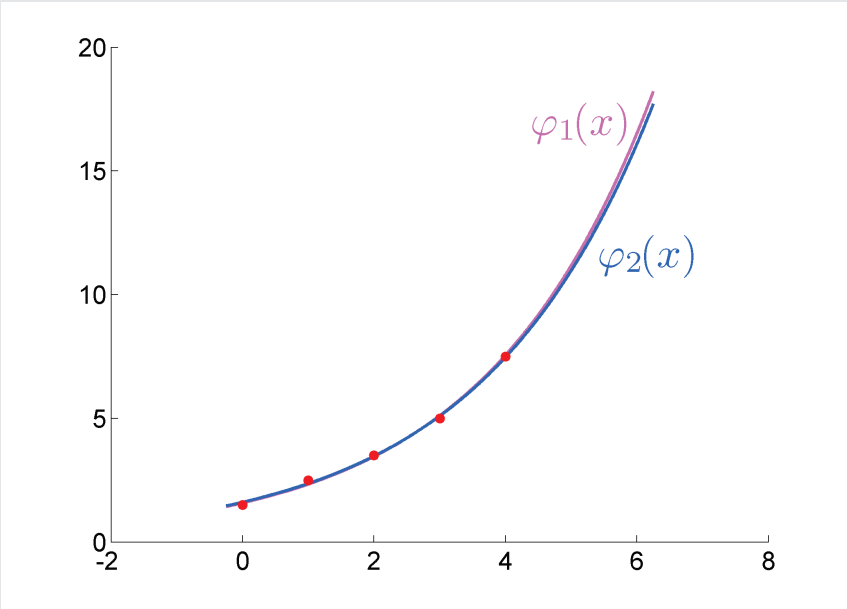


Table 5.6 Change of Variable(s) for Data Linearization

Function, $y = f(x)$	Linearized form, $Y = Ax + B$	Change of variable(s) and constants
$y = \frac{A}{x} + B$	$y = A\frac{1}{x} + B$	$X = \frac{1}{x}, Y = y$
$y = \frac{D}{x + C}$	$y + \frac{-1}{C}(xy) + \frac{D}{C}$	$X = xy, Y = y$
		$C = \frac{-1}{A}, D = \frac{-B}{A}$
$y = \frac{1}{Ax + B}$	$\frac{1}{y} = Ax + B$	$X = x, Y = \frac{1}{y}$
$y = \frac{x}{Ax + B}$	$\frac{1}{y} = A\frac{1}{x} + B$	$X = \frac{1}{x}, Y = \frac{1}{y}$
$y = A\ln(x) + B$	$y = A\ln(x) + B$	$X = \ln(x), Y = y$
$y = Ce^{Ax}$	$\ln(y) = Ax + \ln(C)$	$X = x, Y = \ln(y)$
		$C = e^B$
$y = Cx^A$	$\ln(y) = A\ln(x) + \ln(C)$	$X = \ln(x), Y = \ln(y)$
		$C = e^B$
$y = (Ax + B)^{-2}$	$y^{-1/2} = Ax + B$	$X = x, Y = y^{-1/2}$
$y = Cxe^{-Dx}$	$\ln\left(\frac{y}{x}\right) = -Dx + \ln(C)$	$X = x, Y = \ln\left(\frac{y}{x}\right)$
		$C = e^B, D = -A$
$y = \frac{L}{1 + Ce^{Ax}}$	$\ln\left(\frac{L}{y} - 1\right) = Ax + \ln(C)$	$X = x, Y = \ln\left(\frac{L}{y} - 1\right)$
		$C = e^B$ and L is a constant that must be given

Fourierova analýza

Do této chvíle jsme se zabývali aproximacemi funkce hlavně 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:

$$\begin{aligned}
\varphi_0(x) &= 1 \quad \left(\text{nebo } \frac{1}{2}\right) \\
\varphi_{2k-1}(x) &= \cos\frac{2\pi kx}{T} \qquad k = 1, 2, \dots \\
\varphi_{2k}(x) &= \sin\frac{2\pi kx}{T} \qquad k = 1, 2, \dots,
\end{aligned}$$

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 (pozor na počet) tak ve spojitém případě. Ověřte!

Úlohu najít koeficienty c_i u bázových funkcí φ_i z vyjádření

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

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 ... A_k
u bázových funkcí $\varphi_{2k}(x) = \sin(2\pi kx)/T$ použijeme ... B_k

Následující jednoduchý příklad ukáže 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.

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & -1 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -1 \end{bmatrix}}_{\mathbf{Q}} \underbrace{\begin{bmatrix} A_0 \\ A_1 \\ B_1 \\ A_2 \end{bmatrix}}_{\mathbf{c}} = \underbrace{\begin{bmatrix} 12 \\ -4 \\ 0 \\ 4 \end{bmatrix}}_{\mathbf{F}}.$$

tj.

$$\underbrace{\begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}}_{\mathbf{Q}^T\mathbf{Q}} \underbrace{\begin{bmatrix} A_0 \\ A_1 \\ B_1 \\ A_2 \end{bmatrix}}_{\mathbf{c}} = \underbrace{\begin{bmatrix} 12 \\ 12 \\ -8 \\ 12 \end{bmatrix}}_{\mathbf{Q}^T\mathbf{F}}.$$

$\mathbf{Q}^T\mathbf{Q}$ je diagonální, protože funkce $\varphi_0(x) = \frac{1}{2}$, $\varphi_1(x) = \cos x$, $\varphi_2(x) = \sin x$ jsou diskrétně ortogonální ve smyslu skalárního součinu

$$(\varphi, \psi) = \sum_{j=0}^{2} \varphi(x_j)\psi(x_i) \qquad x_j = \frac{2\pi j}{N}, \quad N = 3$$

D.cv:

$$\left(\frac{1}{2}, \cos x\right) = \cdots = 0$$

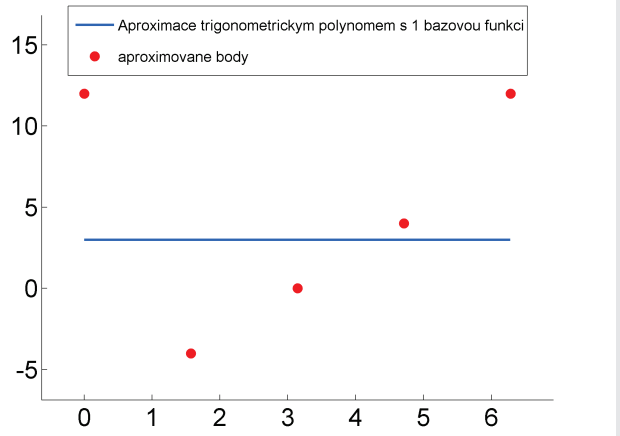
$$\left(\frac{1}{2}, \sin x\right) = \cdots = 0$$

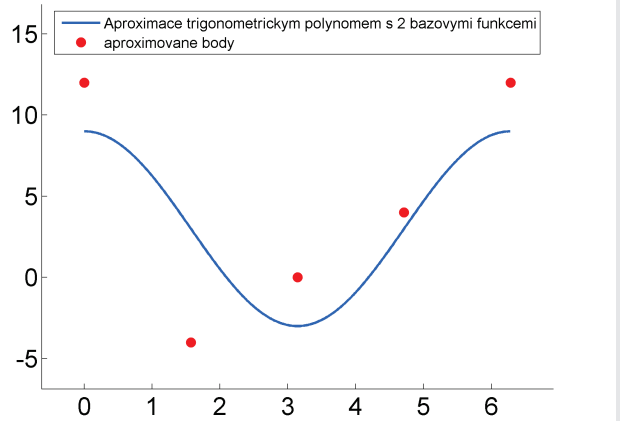
$$\left(\cos x, \sin x\right) = \cdots = 0$$

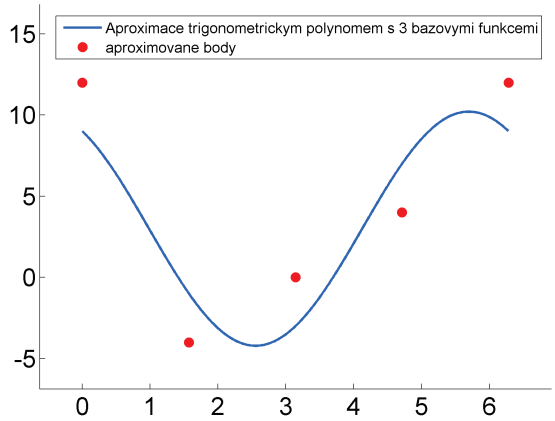
Vyřešením soustavy získáme hledané koeficienty $A_0 = 3$, $A_1 = 6$, $B_1 = -4$, $A_2 = 3$ a tím i aproximující trigonometrický polynom

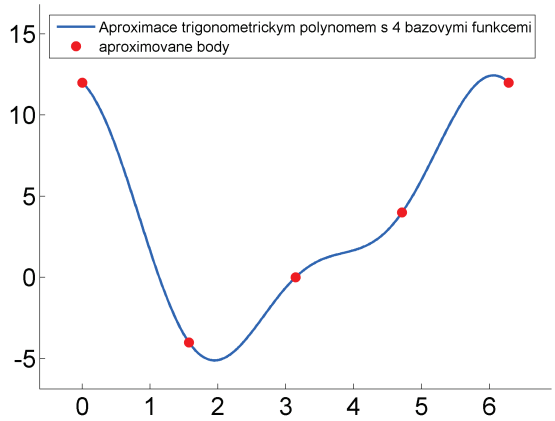
$$\varphi(x) = 3 + 6\cos x - 4\sin x + 3\cos 2x.$$

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13









Úloha diskrétní Fourierovy analýzy

Řadu T periodických funkcí (integrovatelných) lze vyjádřit ve tvaru **Fourierovy řady**

$$f(x) = \sum_{k=0}^{\infty} \left(a_k \cos \frac{2\pi k x}{T} + b_k \sin \frac{2\pi k x}{T} \right)$$

nebo

$$f(x) = \sum_{k=0}^{\infty} r_k \sin \left(\frac{2\pi k x}{T} + v_k \right), \quad \text{kde} \quad r_k^2 = a_k^2 + b_k^2, \quad v_k = \arctan \frac{a_k}{b_k}$$

Položili jsme $a_k = r_k \sin v_k, \quad b_k = r_k \cos v_k,$

$$a \cos \alpha + b \sin \alpha = r \sin v \cos \alpha + r \cos v \sin \alpha = r(\sin v \cos \alpha + \cos v \sin \alpha) = r \sin(\alpha + v).$$

Fourierovou (harmonickou) analýzou rozumíme úlohu určit amplitudy r_k a fáze v_k tzv. harmonických složek $r_k \sin \frac{2\pi k r}{T} + v_k$, je-li dána funkce $f(x)$.

Fourierovou (harmonickou) syntézou rozumíme úlohu určit funkci f, jsou-li dány fáze v_k a amplitudy r_k.

Tuto úlohu můžeme řešit několika způsoby:

(i) Vyjdeme z úlohy spojité Fourierovy analýzy a numericky vypočteme Fourierovy koeficienty, které jsou dány:

$$a_k = \frac{2}{T} \int_0^T f(x) \cos \frac{2\pi k x}{T} \, dx$$

$$b_k = \frac{2}{T} \int_0^T f(x) \sin \frac{2\pi k x}{T} \, dx$$

Užijeme např. lichoběžníkové pravidlo (viz další přednáška). Pozor, pro velká k integrandy oscilují!

(ii) Funkci f aproximujeme (interpolace, diskrétní L_2-aproximace) přímo vhodnou funkcí φ, která má tvar trigonometrického polynomu.

Použijeme přístup (ii) realizovaný v příkladu.

Diskrétní Fourierova analýza - ve smyslu interpolace

Věta Trigonometrický polynom

$$\varphi(x) = \frac{A_0}{2} + \sum_{k=1}^{L} (A_k \cos kx + B_k \sin kx), \qquad N = 2L + 1 \quad (N \text{ liché})$$

resp.

$$\varphi(x) = \frac{A_0}{2} + \sum_{k=1}^{L-1} (A_k \cos kx + B_k \sin kx) + \frac{A_L}{2} \cos Lx, \qquad N = 2L \quad (N \text{ sudé})$$

splňuje interpolační podmínky

$$\varphi(x_j) = f(x_j), \quad x_j = \frac{2\pi j}{N}, \qquad j = 0, 1, \ldots, N-1,$$

právě když koeficienty polynomu $\varphi(x)$ jsou dány pomocí vzorců

$$A_k = \frac{2}{N} \sum_{j=0}^{N-1} f_j \cos kx_j, \qquad k = 0, 1, \ldots, L$$

$$B_k = \frac{2}{N} \sum_{j=0}^{N-1} f_j \sin kx_j, \qquad k = (0,)1, 2, \ldots, L$$

$$\left(A_0 = \frac{2}{N}\sum_{j=0}^{N-1} f_j\right)$$

Důkaz:

$$(\varphi(x), \cos kx) \to A_k, \qquad (\varphi(x), \sin kx) \to B_k \quad \text{a využijeme ortogonality.}$$

Z těchto vzorců vychází algoritmus diskrétní Fourierovy analýzy.

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

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

Pomocí Eulerova vzorce

$$e^{ix} = \cos x + i \sin x$$

lze pro funkce $\sin x$ a $\cos x$ odvodit 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

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

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

Označíme-li

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

dostaneme

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

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

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

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

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

$$C_{\pm k} = \frac{1}{2}(A_k \mp i B_k) = \frac{1}{N}\sum_{j=0}^{N-1} f(x_j) \cos kx_j \mp i \frac{1}{N}\sum_{j=0}^{N-1} f(x_j) \sin kx_j =$$

$$= \frac{1}{N}\sum_{j=0}^{N-1} f(x_j) \underbrace{(\cos kx_j \mp i \sin kx_j)}_{e^{\mp i k x_j}}$$

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

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 (jen ve speciálních případech).

Výpočet koeficientů C_k představuje sčítání konečné řady.

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

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 transformace (Fast Fourier transform - FFT)**.

Princip metody si ukážeme na následujícím příkladě.

Příklad

Uvažujme následující zadání funkce f pro $N = 2^2 = 4$ ekvidistantní uzlové body.

x_i	$x_0 = 0$	$x_1 = \frac{\pi}{2}$	$x_2 = \pi$	$x_3 = \frac{3\pi}{2}$
$f(x_i)$	f_0	f_1	f_2	f_3

Počítáme koeficienty C_k.

Platí:

$$\boxed{C_k = \frac{1}{4}\left(f_0 + f_1 e^{-ik\frac{\pi}{2}} + f_2 e^{-ik\pi} + f_3 e^{-ik\frac{3\pi}{2}}\right), \quad k = 0,1,2,3}.$$

Označme

$$\boxed{w = e^{-i\frac{\pi}{2}}}, \qquad \boxed{F_k = \frac{1}{4} f_k \quad k = 0,1,2,3}.$$

Potom

$$\boxed{C_k = F_0 + F_1 w^k + F_2 w^{2k} + F_3 w^{3k}, \qquad k = 0,1,2,3}$$

Uvědomme si, že platí

$$w^4 = 1 \qquad (\text{obecně } w^N = 1).$$

$$C_0 = \underbrace{(F_0 + F_2)}_{R_0} + \underbrace{(F_1 + F_3)}_{S_0}$$

$$C_1 = \underbrace{(F_0 + w^2 F_2)}_{R_1} + w\underbrace{(F_1 + w^2 F_3)}_{S_1}$$

$$C_2 = (F_0 + F_2) + w^2(F_1 + F_3)$$

$$C_3 = (F_0 + w^2 F_2) + w^3(F_1 + w^2 F_3)$$

Výpočetní náročnost:

na R_k, S_k ... $4S + 2N$

na C_k ... $4S + 3N$

$\sum$ ... $8S + 5N$

Příklad

Aproximujte periodickou funkci f (perioda $T = 31$) zadanou tabulkou pomocí trigonometrického polynomu (ve smyslu L_2-aproximace, až při použití plného počtu bázových funkcí ve smyslu interpolace).

$$\begin{aligned} x_k &= 1, 2, \ldots, 32 \\ f(x_k) &= 1 \quad \text{pro } k = 1, 2, \ldots, 15 \\ &= 0 \quad \text{pro } k = 16 \\ &= -1 \quad \text{pro } k = 17, 18, \ldots, 31 \\ &= 1 \quad \text{pro } k = 32 \end{aligned}$$

Řešení

výsledky v MATLABu


```
koeficient A(0)  = 0.000000    u bazove funkce phi(0)  = 1
koeficient A(1)  = 0.128701    u bazove funkce phi(1)  = cos(2*pi*1*x/31-1)
koeficient B(1)  = 1.265623    u bazove funkce phi(2)  = sin(2*pi*1*x/31-1)
koeficient A(2)  = 0.001321    u bazove funkce phi(3)  = cos(2*pi*2*x/31-1)
koeficient B(2)  = 0.006426    u bazove funkce phi(4)  = sin(2*pi*2*x/31-1)
koeficient A(3)  = 0.126074    u bazove funkce phi(5)  = cos(2*pi*3*x/31-1)
koeficient B(3)  = 0.401825    u bazove funkce phi(6)  = sin(2*pi*3*x/31-1)
koeficient A(4)  = 0.005229    u bazove funkce phi(7)  = cos(2*pi*4*x/31-1)
koeficient B(4)  = 0.012184    u bazove funkce phi(8)  = sin(2*pi*4*x/31-1)
koeficient A(5)  = 0.120926    u bazove funkce phi(9)  = cos(2*pi*5*x/31-1)
koeficient B(5)  = 0.217866    u bazove funkce phi(10) = sin(2*pi*5*x/31-1)
koeficient A(6)  = 0.011564    u bazove funkce phi(11) = cos(2*pi*6*x/31-1)
koeficient B(6)  = 0.016614    u bazove funkce phi(12) = sin(2*pi*6*x/31-1)
koeficient A(7)  = 0.113468    u bazove funkce phi(13) = cos(2*pi*7*x/31-1)
koeficient B(7)  = 0.132175    u bazove funkce phi(14) = sin(2*pi*7*x/31-1)
koeficient A(8)  = 0.020067    u bazove funkce phi(15) = cos(2*pi*8*x/31-1)
koeficient B(8)  = 0.019075    u bazove funkce phi(16) = sin(2*pi*8*x/31-1)
koeficient A(9)  = 0.104007    u bazove funkce phi(17) = cos(2*pi*9*x/31-1)
koeficient B(9)  = 0.080507    u bazove funkce phi(18) = sin(2*pi*9*x/31-1)
koeficient A(10) = 0.030389    u bazove funkce phi(19) = cos(2*pi*10*x/31-1)
koeficient B(10) = 0.018942    u bazove funkce phi(20) = sin(2*pi*10*x/31-1)
koeficient A(11) = 0.092929    u bazove funkce phi(21) = cos(2*pi*11*x/31-1)
koeficient B(11) = 0.045584    u bazove funkce phi(22) = sin(2*pi*11*x/31-1)
koeficient A(12) = 0.042109    u bazove funkce phi(23) = cos(2*pi*12*x/31-1)
koeficient B(12) = 0.015596    u bazove funkce phi(24) = sin(2*pi*12*x/31-1)
koeficient A(13) = 0.080687    u bazove funkce phi(25) = cos(2*pi*13*x/31-1)
koeficient B(13) = 0.020891    u bazove funkce phi(26) = sin(2*pi*13*x/31-1)
koeficient A(14) = 0.054747    u bazove funkce phi(27) = cos(2*pi*14*x/31-1)
koeficient B(14) = 0.008387    u bazove funkce phi(28) = sin(2*pi*14*x/31-1)
koeficient A(15) = 0.067784    u bazove funkce phi(29) = cos(2*pi*15*x/31-1)
koeficient B(15) = 0.003438    u bazove funkce phi(30) = sin(2*pi*15*x/31-1)

 Aproximace je dana predpisem :

-------------------------------------------------------------------------------
                 L
  phi = A(0) + suma [ A(k)*phi(2k-1) + B(k)*phi(2k) ]
                k=1
                                    pro pocet bazovych funkci N=2L+1
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
                L-1
  phi = A(0) + suma [ A(k)*phi(2k-1) + B(k)*phi(2k) ] + A(L)*phi(2L-1)
                k=1
                                    pro pocet bazovych funkci N=2L
-------------------------------------------------------------------------------
```

Kapitola 9. Numerické derivování

Definice: Existuje-li pro danou funkci $f : \mathbb{R} \to \mathbb{R}$ vlastní (tj. konečná) limita

$$\lim_{h \to 0} \frac{f(a+h) - f(a)}{h}$$

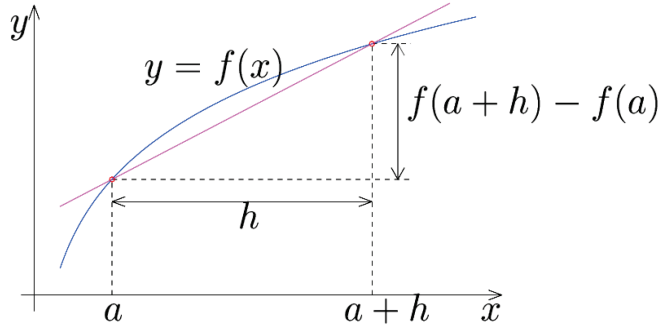
říkáme, že funkce $f(x)$ **má v bodě a derivaci**.

Příslušnou limitu značíme $f'(a)$.

Poznámka:

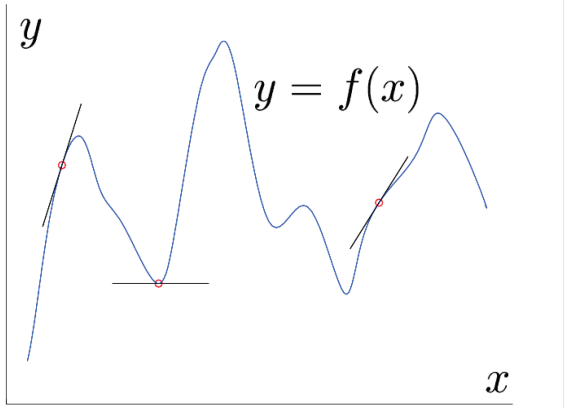
Geometrický význam derivace $f'(a)$ je směrnice tečny křivky dané rovnicí $y = f(x)$ v bodě a (neboť tečna v bodě a je limitní polohou sečny pro $h \to 0$).

Fyzikálně značí derivace funkce $y = f(x)$, kde x je čas a y dráha pohybu, limitu z průměrné rychlosti, tedy okamžitou rychlost v čase a.



Poznámka:

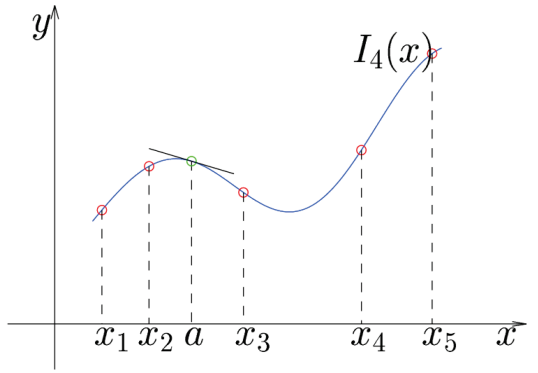
Pro danou funkci $f(x)$ vyjadřuje derivace $f'(x_0)$ míru „stoupání“, resp. „klesání“ v bodě x_0 .



Způsoby odvození vzorců pro výpočet derivace

1. Odvození pomocí interpolačního polynomu

Pro funkci f, která je zadána tabulkou, sestrojíme interpolační polynom a derivaci funkce f v bodě a ztotožníme s derivací tohoto interpolačního polynomu v bodě a.



$$f'(a) \approx I_n'(a)$$
$$f^{(k)}(a) \approx I_n^{(k)}(a)$$

Poznámky:

- Stupeň polynomu nemůže být nižší než řád počítané derivace.
- Pro jednoduchost hledáme hodnotu derivace v uzlovém bodě a navíc uvažujeme ekvidistantní uzly s krokem h.

2. Odvození pomocí Taylorova rozvoje

Pro dostatečně hladkou funkci f platí (pro $h > 0$):

$$f(x_0+h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(\xi_1), \quad \xi_1 \in (x_0, x_0+h)$$
$$f(x_0-h) = f(x_0) - hf'(x_0) + \frac{h^2}{2}f''(\xi_2), \quad \xi_2 \in (x_0-h, x_0)$$

Z první rovnice potom plyne vztah

$$f'(x_0) = \underbrace{\frac{f(x_0+h)-f(x_0)}{h}}_{= D_P f(x_0,h)} - \frac{1}{2}hf''(\xi_1)$$

Podobně ze druhé rovnice

$$f'(x_0) = \underbrace{\frac{f(x_0)-f(x_0-h)}{h}}_{= D_L f(x_0,h)} + \frac{1}{2}hf''(\xi_2)$$

Obdrželi jsme dva základní **dvoubodové** vzorce $D_P f(x_0,h)$ a $D_L f(x_0,h)$, tzv. pravou a levou poměrnou diferenci.

Podobně odvodíme další vzorce pomocí Taylorova rozvoje vyšších řádů. Platí:

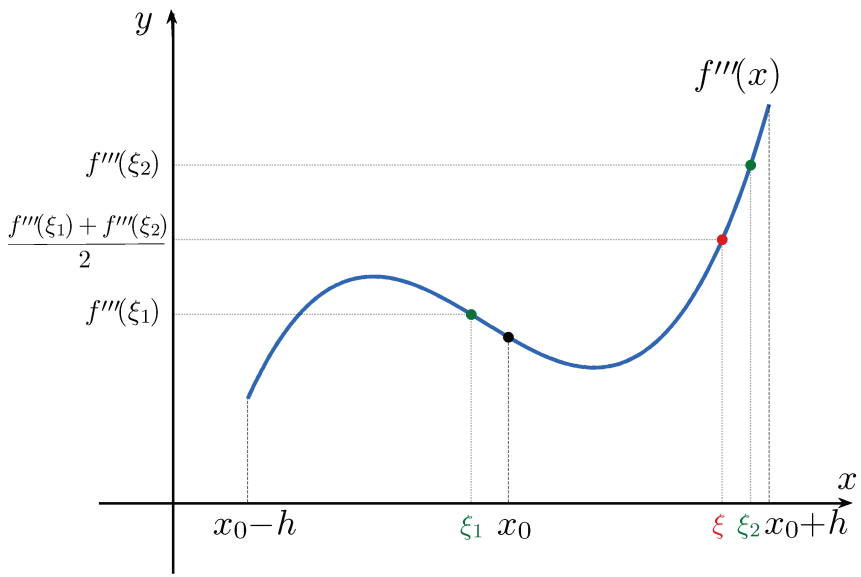
$$f(x_0+h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \frac{h^3}{6}f'''(\xi_1), \quad \xi_1 \in (x_0, x_0+h)$$
$$f(x_0-h) = f(x_0) - hf'(x_0) + \frac{h^2}{2}f''(x_0) - \frac{h^3}{6}f'''(\xi_2), \quad \xi_2 \in (x_0-h, x_0)$$

Po odečtení obdržíme:

$$f(x_0+h) - f(x_0-h) = 2hf'(x_0) + \frac{h^3}{6}(f'''(\xi_1) + f'''(\xi_2))$$

Odtud vyjádříme první derivaci a získáme **tříbodový** vzorec $D_C f(x_0,h)$, tzv. centrální poměrnou diferenci

$$f'(x_0) = \underbrace{\frac{f(x_0+h)-f(x_0-h)}{2h}}_{D_C f(x_0,h)} - \underbrace{\frac{h^2}{12}(f'''(\xi_1)+f'''(\xi_2))}_{\frac{h^2}{6}f'''(\xi)}$$



Uvedené vzorce jsou pro výpočet první derivace $f'(x_0)$.

Pro výpočet druhé derivace $f''(x_0)$ lze použít například vzorec, který dostaneme po sečtení vztahů:

$$f(x_0+h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \frac{h^3}{6}f'''(x_0) + \frac{h^4}{24}f^{(4)}(\xi_1), \qquad \xi_1 \in (x_0, x_0+h)$$
$$f(x_0-h) = f(x_0) - hf'(x_0) + \frac{h^2}{2}f''(x_0) - \frac{h^3}{6}f'''(x_0) + \frac{h^4}{24}f^{(4)}(\xi_2), \qquad \xi_2 \in (x_0-h, x_0)$$

$$f(x_0+h) + f(x_0-h) = 2f(x_0) + \frac{h^2}{2}f''(x_0) + \frac{h^4}{24}(f^{(4)}(\xi_1) + f^{(4)}(\xi_2))$$

Odtud vyjádříme druhou derivaci a získáme **tříbodový** vzorec pro druhou derivaci

$$f''(x_0) = \frac{f(x_0+h) - 2f(x_0) + f(x_0-h)}{h^2} - \underbrace{\frac{h^2}{24}(f^{(4)}(\xi_1) + f^{(4)}(\xi_2))}_{\frac{h^2}{12}f^{(4)}(\xi)}$$

Poznámka:

Samozřejmě lze odvodit řadu dalších vzorců, přičemž platí, že čím více bodů použijeme, tím bude řád chyby vyšší.

Příklad: Pomocí uvedených tří vzorců vypočtěte přibližnou hodnotu první derivace funkce $f(x) = e^x(1-x)$ v bodě $x_0 = 1$. Použijte krok $h = 0{,}1$.

Řešení:

Nejprve si pro kontrolu analyticky zjistíme přesnou hodnotu první derivace funkce f bodě x_0.

$$f'(x) = e^x(1-x) + e^x(-1) = -xe^x, \quad \text{tj.} \quad f'(1) = -1e^1 = -e \approx -2,7182$$

Nyní použijeme pravou, levou a centrální poměrnou diferenci:

1.
$$D_P f(x_0,h) = \frac{f(x_0+h)-f(x_0)}{h} = \frac{e^{1,1}(1-1,1) - e^1(1-1)}{0,1} = \\ = \frac{-0,1e^{1,1}}{0,1} = -e^{1,1} \approx -3,0041 \qquad \text{(chyba } 0,2858\text{)}$$

2.
$$D_L f(x_0,h) = \frac{f(x_0)-f(x_0-h)}{h} = \frac{e^1(1-1) - e^{0,9}(1-0,9)}{0,1} = \\ = \frac{-0,1e^{0,9}}{0,1} = -e^{0,9} \approx -2,4596 \qquad \text{(chyba } 0,2586\text{)}$$

3.
$$D_C f(x_0,h) = \frac{f(x_0+h)-f(x_0-h)}{2h} = \frac{e^{1,1}(1-1,1) - e^{0,9}(1-0,9)}{0,2} = \\ = \frac{-0,1e^{1,1} - 0,1e^{0,9}}{0,2} = -\frac{e^{1,1}+e^{0,9}}{2} \approx -2,7318 \\ \text{(chyba } 0,0136\text{)}$$

Všimněme si velikosti chyb v jednotlivých případech. Potvrzuje se fakt, že chyba prvních dvou (dvoubodových) vzorců je řádu h, tj. v řádu desetin a chyba posledního (tříbodového) vzorce je řádu h^2, tj. v řádu setin.

Podmíněnost úlohy numerického derivování

Uvažujme nyní např. vzorec s pravou diferencí $D_P f(x_0,h)$, tj. platí

$$f'(x_0) = \underbrace{\frac{f(x_0+h)-f(x_0)}{h}}_{D_P f(x_0,h)} - \underbrace{\frac{1}{2}hf''(\xi)}_{\text{chyba metody}}$$

Chybu metody označme r_1.

Platí-li $|f''(x)| < M$ pro $x \in (x_0, x_0+h)$, potom $|r_1| \le \frac{M}{2}h$.

Musíme uvážit chyby měření (zaokrouhlovací chyby) - označíme r_2.

Označíme-li

$f(x_0), f(x_0+h)$ ..... přesné hodnoty

$f^\star(x_0), f^\star(x_0+h)$ ..... vstupní hodnoty

Potom pro r_2 platí

$$r_2 = \underbrace{\frac{f(x_0+h)-f(x_0)}{h}}_{\text{přesná hodnota vzorce}} - \underbrace{\frac{f^\star(x_0+h)-f^\star(x_0)}{h}}_{\text{vypočtená hodnota vzorce}}$$

A dále

$$|r_2| = \left|\frac{f(x_0+h)-f^\star(x_0+h)}{h} + \frac{f^\star(x_0)-f(x_0)}{h}\right| \le \\ \le \frac{|f(x_0+h)-f^\star(x_0+h)|}{h} + \frac{|f^\star(x_0)-f(x_0)|}{h} \le \\ < \frac{\varepsilon}{h} + \frac{\varepsilon}{h} = \frac{2\varepsilon}{h}$$

Využili jsme zde odhady

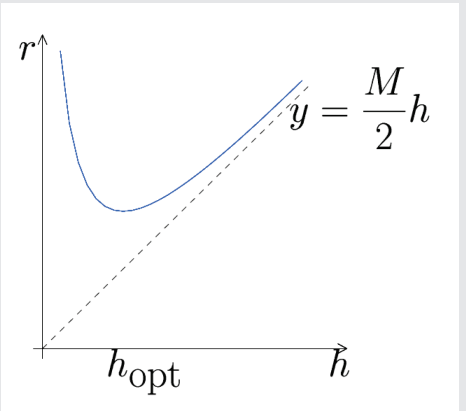
$$|f^\star(x_0+h) - f(x_0+h)| \le \varepsilon$$

$$|f^\star(x_0) - f(x_0)| \le \varepsilon$$

číslo ε může představovat např. strojovou přesnost.

Pro celkovou chybu r potom platí

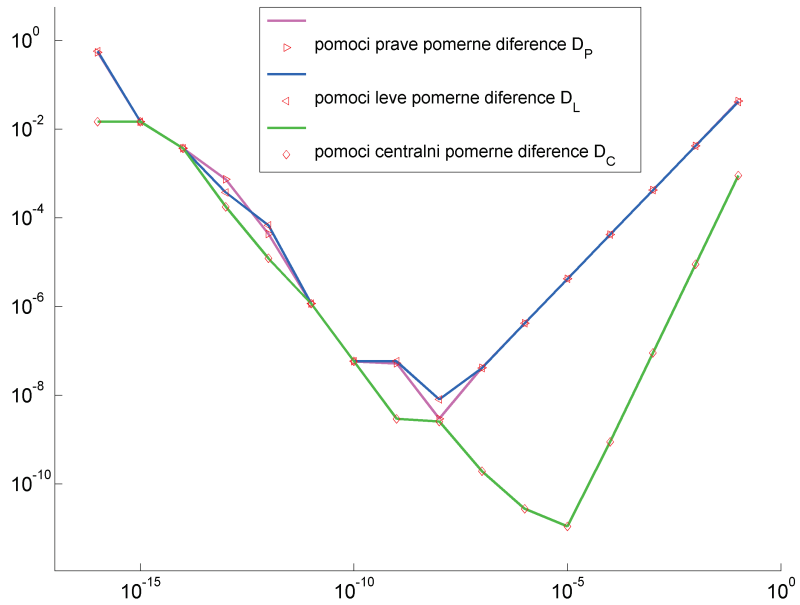
$$|r| \le |r_1| + |r_2| \le \frac{M}{2}h + \frac{2\varepsilon}{h}$$



- Úloha numerického derivování je špatně podmíněná ! (pro zmenšující se h roste chyba)
- Lze najít optimální krok h_{opt}

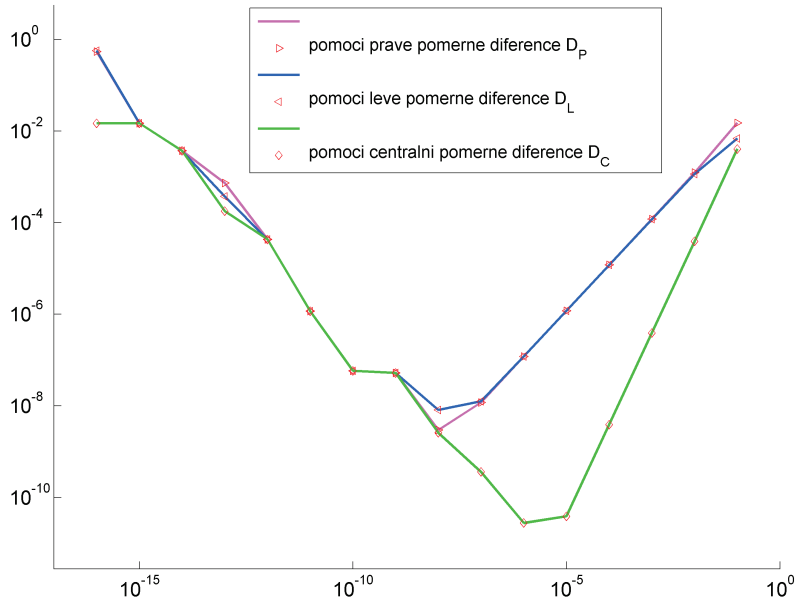
V následujících příkladech ukažte chybu 3 odvozených vzorců pro výpočet první derivace funkce $f(x)$ v bodě x_0 při použití kroků $h = 10^{-16}, 10^{-15}, \ldots, 10^{-1}$.

Příklad 1 $f(x) = \sin x$, $x_0 = 1$.

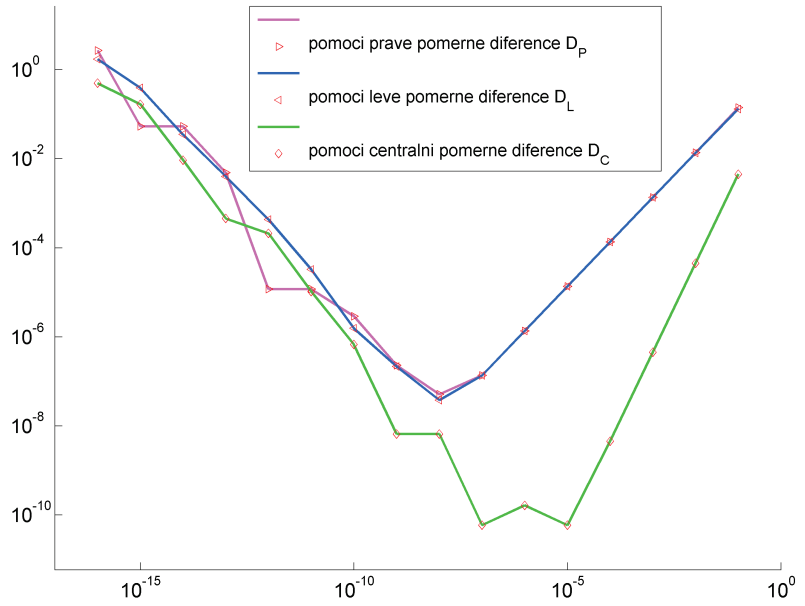


Příklad 2 $f(x) = \sin \frac{1}{x}$, $x_0 = 1$.

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

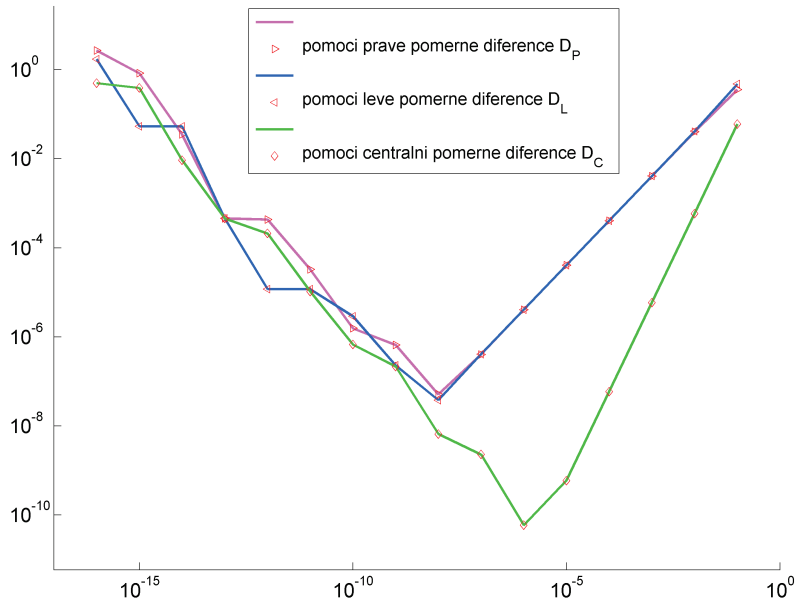


Příklad 3 $f(x) = \mathrm{e}^x$, $x_0 = 1$.



Příklad 4 $f(x) = \mathrm{e}^{\frac{1}{x}}$, $x_0 = 1$.

Numerické metody
Josef Daněk
KMA/NM
13.2.2013



Poznámka:

Na základě špatné podmíněnosti se zdá, že nebude možné při výpočtu derivace dosáhnout libovolné přesnosti.

Zvýšení přesnosti ale můžeme dosáhnout

1) použitím vzorce s chybou vyššího řádu
2) použitím tzv. Richardsonovy extrapolace

Richardsonova extrapolace

Jde o obecný princip, který se používá nejen u numerického derivování.

Myšlenka vychází z toho, že na základě znalosti výrazu pro rozvoj chyby využijeme dvou přibližných výsledků k získání třetího, který bude přesnější.

Tento proces eliminace chyb budeme demonstrovat např. na poměrné centrální diferenci $D_C f(x_0, h)$.

Vyjdeme z Taylorova rozvoje:

$$f(x_0+h) = f(x_0)+hf'(x_0)+\frac{h^2}{2}f''(x_0)+\frac{h^3}{6}f'''(x_0)+\frac{h^4}{24}f^{(4)}(x_0)+\frac{h^5}{5!}f^{(5)}(\xi_1) \quad (1)$$

$$f(x_0-h) = f(x_0)-hf'(x_0)+\frac{h^2}{2}f''(x_0)-\frac{h^3}{6}f'''(x_0)+\frac{h^4}{24}f^{(4)}(x_0)-\frac{h^5}{5!}f^{(5)}(\xi_2) \quad (2)$$

(1)−(2) $$f(x_0+h)-f(x_0-h)=2hf'(x_0)+\frac{h^3}{3}f'''(x_0)+\underbrace{\frac{h^5}{5!}\left(f^{(5)}(\xi_1)+f^{(5)}(\xi_2)\right)}_{O(h^5)}$$

$$\underbrace{\frac{f(x_0+h)-f(x_0-h)}{2h}}_{D_Cf(x_0,h)}=f'(x_0)+\underline{\frac{h^2}{6}f'''(x_0)}+O(h^4) \qquad \circledast$$

Stejný vzorec použijeme pro výpočet s krokem $\bar{h}=2h$.

$$D_Cf(x_0,\bar{h})=f'(x_0)+\frac{\bar{h}^2}{6}f'''(x_0)+O(\bar{h}^4)$$

$$D_Cf(x_0,2h)=f'(x_0)+4\underline{\frac{h^2}{6}f'''(x_0)}+O(h^4) \qquad \circledast\circledast$$

Podtržené členy chceme eliminovat – rovnici $\circledast$ vynásobíme 4

$$4D_Cf(x_0,h)=4f'(x_0)+4\frac{h^2}{6}f'''(x_0)+O(h^4)$$

Odečteme $\circledast\circledast$

$$D_Cf(x_0,2h)=f'(x_0)+4\frac{h^2}{6}f'''(x_0)+O(h^4)$$

$$4D_Cf(x_0,h)-D_Cf(x_0,2h)=3f'(x_0)+O(h^4)$$

$$f'(x_0)=\frac{4D_Cf(x_0,h)-D_Cf(x_0,2h)}{3}+O(h^4)$$

nebo jinak zapsáno

$$f'(x_0)=D_Cf(x_0,h)+\frac{D_Cf(x_0,h)-D_Cf(x_0,2h)}{3}+O(h^4)$$

Poznámka:

Tímto způsobem jsme eliminovali chybu řádu h^2. Algoritmus Richardsonovy extrapolace lze samozřejmě použít opakovaně pro eliminaci chyb vyšších řádů. Tato metoda je potom velmi efektivní.

Pokud bychom chtěli stejným způsobem eliminovat chybu řádu např. 4, potom bychom dostali:

$$D_Cf(x_0,h)=f'(x_0)+Kh^4 \qquad \big/\cdot 2^4 \text{ a odečteme 2. rovnici}$$

$$D_Cf(x_0,2h)=f'(x_0)+K2^4h^4$$

$$f'(x_0)=\frac{2^4D_Cf(x_0,h)-D_Cf(x_0,2h)}{2^4-1}$$

$$f'(x_0)=D_Cf(x_0,h)+\frac{1}{2^4-1}\left(D_Cf(x_0,h)-D_Cf(x_0,2h)\right)$$

Pro eliminaci chyb vyšších řádů postupujeme analogicky, tj. místo **4** použijeme příslušný řád.

Poznámka:

V názvu metody se objevuje slovo extrapolace. Je to proto, že nová hodnota derivace je lineární kombinací

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Při určování rozvoje chyb jednotlivých vzorců vycházíme z Taylorova rozvoje funkce f.

Rozvoj chyby pro $D_Pf(x_0,h)$ a $D_Lf(x_0,h)$

(1) $$f(x_0+h)=f(x_0)+hf'(x_0)+\frac{h^2}{2}f''(x_0)+\frac{h^3}{6}f'''(x_0)+\frac{h^4}{24}f^{(4)}(x_0)+\frac{h^5}{120}f^{(5)}(\xi_1), \qquad \xi_1\in(x_0,x_0+h)$$

(2) $$f(x_0-h)=f(x_0)-hf'(x_0)+\frac{h^2}{2}f''(x_0)-\frac{h^3}{6}f'''(x_0)+\frac{h^4}{24}f^{(4)}(x_0)-\frac{h^5}{120}f^{(5)}(\xi_2), \qquad \xi_2\in(x_0-h,x_0)$$

---

(1) $\Rightarrow$ $$f'(x_0)=\underbrace{\frac{f(x_0+h)-f(x_0)}{h}}_{D_Pf(x_0,h)}-\underbrace{\frac{h}{2}f''(x_0)}_{c_1h}-\underbrace{\frac{h^2}{6}f'''(x_0)}_{c_2h^2}-\underbrace{\frac{h^3}{24}f^{(4)}(x_0)}_{c_3h^3}-\underbrace{\frac{h^4}{120}f^{(5)}(\xi_1)}_{O(h^4)}$$

(2) $\Rightarrow$ $$f'(x_0)=\underbrace{\frac{f(x_0)-f(x_0-h)}{h}}_{D_Lf(x_0,h)}+\underbrace{\frac{h}{2}f''(x_0)}_{c_1h}-\underbrace{\frac{h^2}{6}f'''(x_0)}_{c_2h^2}+\underbrace{\frac{h^3}{24}f^{(4)}(x_0)}_{c_3h^3}-\underbrace{\frac{h^4}{120}f^{(5)}(\xi_2)}_{O(h^4)}$$

Rozvoj chyby pro $D_Cf(x_0,h)$

$$f(x_0+h)=f(x_0)+hf'(x_0)+\frac{h^2}{2}f''(x_0)+\frac{h^3}{6}f'''(x_0)+\frac{h^4}{24}f^{(4)}(x_0)+\frac{h^5}{120}f^{(5)}(x_0)+\frac{h^6}{720}f^{(6)}(x_0)+\frac{h^7}{7!}f^{(7)}(\xi_1),$$
$$\xi_1\in(x_0,x_0+h)$$

$$f(x_0-h)=f(x_0)-hf'(x_0)+\frac{h^2}{2}f''(x_0)-\frac{h^3}{6}f'''(x_0)+\frac{h^4}{24}f^{(4)}(x_0)-\frac{h^5}{120}f^{(5)}(x_0)+\frac{h^6}{720}f^{(6)}(x_0)-\frac{h^7}{7!}f^{(7)}(\xi_2),$$
$$\xi_2\in(x_0-h,x_0)$$

Po odečtení:

$$f(x_0+h)-f(x_0-h)=2hf'(x_0)+\frac{1}{3}h^3f'''(x_0)+\frac{1}{60}h^5f^{(5)}(x_0)+\underbrace{\frac{h^7}{7!}\left(f^{(7)}(\xi_1)+f^{(7)}(\xi_2)\right)}_{=O(h^7)}$$

$$f'(x_0)=\underbrace{\frac{f(x_0+h)-f(x_0-h)}{2h}}_{D_Cf(x_0,h)}-\underbrace{\frac{1}{6}h^2f'''(x_0)}_{c_1h^2}-\underbrace{\frac{1}{120}h^4f^{(5)}(x_0)}_{c_2h^4}-\underbrace{\frac{h^6}{7!}2f^{(7)}(\xi)}_{O(h^6)}$$

Algoritmus Richardsonovy extrapolace

Na základě znalosti rozvoje chyby příslušného vzorce můžeme pro zpřesňování hodnoty vypočtené derivace použít následující algoritmus.

Algoritmus (např. pro vzorec $D_C f$)

Pro $s = 0, 1, 2, \ldots, S$

$T_{s,0} = D_c f(x_0, 2^{-s} h)$

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

$$T_{s,k} = T_{s,k-1} + \frac{T_{s,k-1} - T_{s-1,k-1}}{\underbrace{4^k}_{(*)} - 1}$$

$(*)$ $4^k = 2^{2k}$, protože v rozvoji chyby tohoto vzorce jsou pouze sudé mocniny h.

Schéma

$$\begin{array}{llll} h & T_{00} & & \\ \frac{h}{2} & T_{10} & T_{11} & \\ \frac{h}{4} & T_{20} & T_{21} & T_{22} \\ \frac{h}{8} & T_{30} & T_{31} & T_{32} \quad T_{33} \end{array}$$

Poznámka: Pokud se např. rovnají T_{22} a T_{32}, nemusíme počítat T_{33}, protože vyjde stejně.

Příklad:

Použijte opakovanou Richardsonovu extrapolaci pro výpočet derivace funkce $f(x) = \ln x$ v bodě $x_0 = 3$ pomocí centrální poměrné diference s kroky $h = 0,8$; $0,4$; $0,2$ a $0,1$.

Řešení:

Ukázali jsme, že pro dostatečně hladkou funkci f platí vztah

$$f'(x_0) = \underbrace{\frac{f(x_0+h) - f(x_0-h)}{2h}}_{D_C f(x_0, h)} + \underbrace{c_1 h^2 + c_2 h^4 + c_3 h^6 + \ldots}_{\text{rozvoj chyby}},$$

kde čísla c_1, c_2, c_3 představují kontanty obsahující příslušné derivace.

Výsledky zapíšeme přehledně do tabulky:

h	$f'(x_0, h)$	1. korekce $[\frac{4}{3}; -\frac{1}{3}]$	2. korekce $[\frac{16}{15}; -\frac{1}{15}]$
$0,8$	$0,341589$		
$0,4$	$0,335329$	$\frac{4}{3}\,0,335329 - \frac{1}{3}\,0,341589 = 0,333242$	
$0,2$	$0,333828$	$\frac{4}{3}\,0,333828 - \frac{1}{3}\,0,335329 = 0,333327$	$\frac{16}{15}\,0,333327 - \frac{1}{15}\,0,333242 = \mathbf{0,333332}$
$0,1$	$0,333456$	$\frac{4}{3}\,0,333456 - \frac{1}{3}\,0,333828 = 0,333332$	$\frac{16}{15}\,0,333332 - \frac{1}{15}\,0,333327 = \mathbf{0,333332}$

Ve výpočtu jsme použili jednak 1. korekci pro eliminaci chyby řádu h^2, ale dále také 2. korekci, která eliminovala chybu řádu h^4.

V tabulce chybí sloupec pro 3. korekci. Důvod je ten, že se hodnoty, ze kterých by se extrapolovala nová hodnota, rovnají (dostali bychom to samé číslo).

Pro úplnost dodejme, že přesná hodnota derivace je $f'(x) = \frac{1}{x}$, tj. $f'(3) = \frac{1}{3}$.

Pomocí následujících výsledků lze porovnat efektivitu při použití různých vzorců.

výsledky v MATLABu

```
>> derivace_richardson('D_P', '-sin(exp(x))', 1, 0.4, 3);
-----------------------------------------------------------------
  Vypocte hodnotu prvni derivace zadane funkce
  f=-sin(exp(x)) v bode x0=1.000000 s kroky
  h=[0.4, 0.2, 0.1, 0.05]

  Pro vypocet se pouzije vzorec prave pomerne diference D_P.
  Ke zpresneni se pouzije Richardsonova extrapolace.
-----------------------------------------------------------------

 !     h      |  D_P(f,x0,h)  |   1.korekce   |   2.korekce   |   3.korekce   |
 ============================================================================
 |  0.40000   |  3.006234654  |
 |  0.20000   |  2.941793905  |  2.877353156  |
 |  0.10000   |  2.737868276  |  2.533942647  |  2.419472477  |
 |  0.05000   |  2.612795286  |  2.487722295  |  2.472315512  |  2.479864517  |

  Presna hodnota derivace funkce f v bode x0 je   2.478349732955
```

výsledky v MATLABu

```
>> derivace_richardson('D_L', '-sin(exp(x))', 1, 0.4, 3);
------------------------------------------------------------------
  Vypocte hodnotu prvni derivace zadane funkce
  f=-sin(exp(x)) v bode x0=1.000000 s kroky
  h=[0.4, 0.2, 0.1, 0.05]

  Pro vypocet se pouzije vzorec leve pomerne diference D_L.
  Ke zpresneni se pouzije Richardsonova extrapolace.
------------------------------------------------------------------

 !     h     |  D_L(f,x0,h)  |   1.korekce   |   2.korekce   |   3.korekce   |
 ============================================================================
 |  0.40000  |  1.394507747  |
 |  0.20000  |  1.912110950  |  2.429714153  |
 |  0.10000  |  2.195575019  |  2.479039089  |  2.495480735  |
 |  0.05000  |  2.338245228  |  2.480915437  |  2.481540886  |  2.479549479  |

  Presna hodnota derivace funkce f v bode x0 je   2.478349732955
```

výsledky v MATLABu

```
>> derivace_richardson('D_C', '-sin(exp(x))', 1, 0.4, 3);
------------------------------------------------------------------
  Vypocte hodnotu prvni derivace zadane funkce
  f=-sin(exp(x)) v bode x0=1.000000 s kroky
  h=[0.4, 0.2, 0.1, 0.05]

  Pro vypocet se pouzije vzorec centralni pomerne diference D_C.
  Ke zpresneni se pouzije Richardsonova extrapolace.
------------------------------------------------------------------

 !     h     |  D_C(f,x0,h)  |   1.korekce   |   2.korekce   |   3.korekce   |
 ============================================================================
 |  0.40000  |  2.200371201  |
 |  0.20000  |  2.426952427  |  2.502479503  |
 |  0.10000  |  2.466721648  |  2.479978054  |  2.478477958  |
 |  0.05000  |  2.475520257  |  2.478453127  |  2.478351465  |  2.478349457  |

  Presna hodnota derivace funkce f v bode x0 je   2.478349732955
```

Aproximaci derivace funkce jsme již použili např. u odvození metody sečen z Newtonovy metody.

Newtonova metoda

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

metoda sečen

Numerické metody
Josef Daněk
KMA/NM
13.2.2013

$$x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

Metoda konečných diferencí

... jeden ze způsobů řešení okrajových úloh.

Řešme okrajovou úlohu (ODR 2.řádu)

$$\boxed{\begin{aligned} -u'' + q(x)u &= f(x) \qquad x \in (0,1) \\ u(0) &= g_0 \\ u(1) &= g_1 \end{aligned}} \qquad \circledast$$

q, f ... dané funkce definované a spojité na $\langle 0,1\rangle$, $q(x) \geq 0$
g_0, g_1 ... daná čísla

($\Rightarrow$ úloha má právě 1 klasické řešení)

Interval $\langle 0,1\rangle$ rozdělíme na N stejných podintervalů (ekvidistantní síť)

$S = \{x_i = ih; \quad i = 0, 1, \ldots, N\}$ ... síť; h ... krok sítě.

Přibližné řešení konstruujeme jako funkci diskrétního argumentu x_i.

Přibližné řešení je určeno vektorem $\mathbf{U} = [U_0, U_1, U_2, \ldots, U_{N-1}, U_N]$, kde složky U_i aproximují hodnoty $u(x_i)$ přesného řešení v uzlech sítě.

Klasické řešení splňuje rovnici $\circledast$ v každém bodě $x \in (0,1)$, tj.

$$-u''(x) + q(x)u(x) = f(x)$$

Na $\langle 0,1\rangle$ jsme zvolili rovnoměrnou síť S a v každém vnitřním bodě této sítě musí platit:

$$\boxed{-u''(x_i) + q(x_i)u(x_i) = f(x_i), \quad i = 1, 2, \ldots, N-1} \qquad \circledast\circledast$$

druhou derivaci aproximujeme druhou poměrnou diferencí

$$\boxed{u''(x_i) \approx \frac{1}{h}\left[\frac{u(x_i+h) - u(x_i)}{h} - \frac{u(x_i) - u(x_i-h)}{h}\right] = \frac{1}{h^2}\big[u(x_i-h) - 2u(x_i) + u(x_i+h)\big]}$$

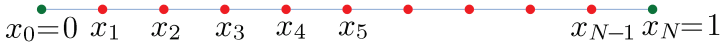
soustavu $\circledast\circledast$ nahradíme soustavou přibližných rovností

$$\boxed{-\frac{1}{h^2}\left[u(x_i-h) - 2u(x_i) + u(x_i+h)\right] + q(x_i)u(x_i) \approx f(x_i)}$$

tj.

$$\begin{aligned} -\frac{1}{h^2}(U_{i-1}-2U_i+U_{i+1})+q(x_i)U_i &= f(x_i) \qquad i=1,2,\ldots,N-1 \\ U_0 &= g_0 \\ U_N &= g_1 \end{aligned}$$

... soustava diferenčních rovnic



Neznáme hodnoty uvnitř intervalu, tj. v $x_1, x_2, \ldots, x_{N-1}$.
V první rovnici figuruje hodnota U_0, ta je ale rovna g_0 a tento člen převedeme na pravou stranu.
Obdobně pro poslední rovnici, za U_N dosadíme g_N a převedeme na pravou stranu.

Získaná soustava:

$$\underbrace{\frac{1}{h^2}\begin{bmatrix} 2+h^2q_1 & -1 & 0 & & \ldots & 0 \\ -1 & 2+h^2q_2 & -1 & 0 & \ldots & 0 \\ 0 & -1 & 2+h^2q_3 & -1 & & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & 0 & -1 & 2+h^2q_{N-2} & -1 \\ 0 & 0 & \ldots & 0 & -1 & 2+h^2q_{N-1} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ \vdots \\ U_{N-1} \end{bmatrix}}_{\mathbf{U}} = \underbrace{\begin{bmatrix} f_1+\frac{g_0}{h^2} \\ f_2 \\ f_3 \\ \vdots \\ \vdots \\ f_{N-1}-\frac{g_1}{h^2} \end{bmatrix}}_{\mathbf{F}}$$

Soustava lineárních algebraických rovnic $\boxed{\mathbf{AU}=\mathbf{F}}$

$\mathbf{A}$ ... symetrická, třídiagonální, pozitivně definitní ($\Rightarrow$ regulární)

- symetrie matice je důsledek symetrie použité diferenční aproximace u'' a očíslování uzlů (zde je vše přirozené, význam u parciálních diferenciálních rovnic)

- pro regulární matici soustavy $\exists!$ řešení

Otázky:

S jakou přesností přibližné řešení aproximuje přesné řešení?

Zmenšuje se chyba přibližného řešení, zjemňujeme-li síť, tj. $h \to 0$?

Poznámka: Pro aproximaci derivací lze samozřejmě použít i jiné vzorce.
Použijeme-li např. vzorec 7, bude výsledná matice soustavy opět pásová, šíře pásu bude nyní 5.

	Derivace	Aproximace	Odhad chyby	Schematický zápis
1	$u'(x_i)$	$\frac{1}{h}(u_{i+1}-u_i)$	$\frac{1}{2}h \max_{(x_i,x_{i+1})} \lvert u''(x)\rvert,\ u\in C^2$	$\frac{1}{h}\{(-1)_i\ (1)_{i+1}\}$
2	$u'(x_i)$	$\frac{1}{h}(u_i-u_{i-1})$	$\frac{1}{2}h \max_{(x_{i-1},x_i)} \lvert u''(x)\rvert,\ u\in C^2$	$\frac{1}{h}\{(-1)_{i-1}\ (1)_i\}$
3	$u'(x_i)$	$\frac{1}{2h}(u_{i+1}-u_{i-1})$	$\frac{1}{6}h^2 \max_{(x_{i-1},x_{i+1})} \lvert u^{(3)}(x)\rvert,\ u\in C^3$	$\frac{1}{2h}\{(-1)_{i-1}\ (0)_i\ (1)_{i+1}\}$
4	$u'(x_i)$	$\frac{1}{2h}(-u_{i+2}+4u_{i+1}-3u_i)$	$\frac{1}{3}h^2 \max_{(x_i,x_{i+2})} \lvert u^{(3)}(x)\rvert,\ u\in C^3$	$\frac{1}{2h}\{(-3)_i\ (4)_{i+1}\ (-1)_{i+2}\}$
5	$u'(x_i)$	$\frac{1}{2h}(3u_i-4u_{i-1}+u_{i-2})$	$\frac{1}{3}h^2 \max_{(x_{i-2},x_i)} \lvert u^{(3)}(x)\rvert,\ u\in C^3$	$\frac{1}{2h}\{(1)_{i-2}\ (-4)_{i-1}\ (3)_i\}$
6	$u''(x_i)$	$\frac{1}{h^2}(u_{i+1}-2u_i+u_{i-1})$	$\frac{1}{12}h^2 \max_{(x_{i-1},x_{i+1})} \lvert u^{(4)}(x)\rvert,\ u\in C^4$	$\frac{1}{h^2}\{(1)_{i-1}\ (-2)_i\ (1)_{i+1}\}$
7	$u''(x_i)$	$\frac{1}{12h^2}(-u_{i-2}+16u_{i-1}-30u_i+16u_{i+1}-u_{i+2})$	$0(h^4),\ u\in C^6$	$\frac{1}{12h^2}\{(-1)_{i-2}\ (16)_{i-1}\ (-30)_i\ (16)_{i+1}\ (-1)_{i+2}\}$

Ukažme si jak postupovat při diskretizaci úlohy s derivací v okrajové podmínce.

Je-li na některém z konců intervalu zadána Neumannova nebo Newtonova okrajová podmínka, musíme předchozí postup modifikovat, neboť neznáme hodnotu $u(0)$ nebo $u(1)$.

V takovém případě musíme sestavit také diferenční aproximaci příslušné okrajové podmínky a připojit ji k soustavě diferenčních rovnic, jež ve vnitřních uzlech aproximují diferenciální rovnici.

Ukážeme tři takové možné aproximace pro řešení následující úlohy.

$$\begin{aligned} -u''+q(x)u &= f(x) && x\in(0,1) \\ \alpha_0 u(0)-\beta_0 u'(0) &= g_0 && \beta_0>0 \\ u(1) &= g_1 \end{aligned} \qquad (\spadesuit)$$

Ve vnitřních uzlech sítě aproximujeme diferenciální rovnici ($\spadesuit$) stejnými diferenčními rovnicemi jako v předchozím případě

$$-\frac{1}{h^2}(U_{i-1}-2U_i+U_{i+1})+q(x_i)U_i = f(x_i) \qquad i=1,2,\ldots,N-1$$

V těchto rovnicích vystupují neznámé $U_0, U_1, \ldots, U_{N-1}, U_N$.

Abychom dostali stejný počet rovnic jako neznámých, musíme tedy (na základě okrajových podmínek) k získaným $N-1$ rovnicím ještě 2 rovnice připojit. Jednu z nich dostaneme z Dirichletovy okrajové podmínky v bodě $x=1$ tak, že položíme $\boxed{U_N=g_1}$. Levou okrajovou podmínku (v bodě $x=0$) můžeme zužitkovat různými způsoby.

1.způsob

Aproximujeme-li $u(0)$ pomocí vzorce 1 (tj. pravé poměrné diference), dojdeme k diferenční rovnici

$$\alpha_0 U_0 - \beta_0 \frac{U_1-U_0}{h} = g_0 \quad (\beta_0 \neq 0),$$

kterou klasické řešení splňuje s chybou velikosti $O(h)$. Rovnici vynásobíme číslem $1/(h\beta_0)$ a upravíme na tvar

$$\boxed{\frac{1}{h^2}\big[(1-\frac{\alpha_0}{\beta_0}h)U_0-U_1\big]=\frac{g_0}{h\beta_0}};$$

tuto úpravu děláme proto, aby výsledná matice soustavy síťových rovnic byla symetrická.

Předchozí rovnici přidáme k získaným diferenčním rovnicím a dosadíme g_1 za U_N. Dostaneme opět soustavu lineárních algebraických rovnic se symetrickou třídiagonální maticí (pozor: $\mathbf{U}=[U_0,U_1,\dots,U_{N-1}]^T$).

$$\frac{1}{h^2}\begin{bmatrix} 1+\frac{\alpha_0 h}{\beta_0} & -1 & 0 & & \dots & 0 \\ -1 & 2+h^2q_1 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2+h^2q_2 & -1 & & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & 0 & -1 & 2+h^2q_{N-2} & -1 \\ 0 & 0 & \dots & 0 & -1 & 2+h^2q_{N-1} \end{bmatrix}\begin{bmatrix} U_0 \\ U_1 \\ U_2 \\ \vdots \\ \vdots \\ U_{N-1} \end{bmatrix}=\begin{bmatrix} \frac{g_0}{h\beta_0} \\ f_1 \\ f_2 \\ \vdots \\ f_{N-2} \\ f_{N-1}-\frac{g_1}{h^2} \end{bmatrix}$$

Matice soustavy je řádu N a lze dokázat, že je regulární. Přibližné řešení lze proto jednoznačně stanovit a dá se ukázat, že jeho **chyba je velikosti** $O(h)$.

Tato skutečnost, tj. snížení řádu chyby metody, možná překvapí, neboť síťová rovnice pro všechny uzly s výjimkou x_0 aproximuje diferenciální rovnici s diskretizační chybou $O(h^2)$. Přesto však okolnost, že jsme se v jediné rovnici dopustili diskretizační chyby velikosti $O(h)$, ovlivní velikost chyby metody nejen v blízkosti bodu x_0, ale ve všech bodech sítě. Jde tu o jev, který je pro užití diferenční metody typický a ukazuje, že chceme-li využít přesnosti, s níž jsme aproximovali diferenciální rovnici samotnou, musíme stejně přesně aproximovat i okrajové podmínky.

2.způsob

Bezprostřední možnost přesnější náhrady hodnoty derivace v okrajové podmínce spočívá v užití vzorců 4 a 5 z tabulky. Tyto aproximace mají pro $u\in C^3$ diskretizační chybu $O(h^2)$.

Vzniklá soustava diferenčních rovnic však již není třídiagonální, není symetrická a navíc se při jejím řešení standardními metodami mohou vyskytnout numerické problémy. Proto tento postup nedoporučujeme.

3.způsob

Hodnotu $u'(0)$ aproximujeme pomocí vzorce 3 (centrální poměrná diference).

$$u'(0)\approx\frac{1}{2h}\big[u(x_1)-u(x_{-1})\big],$$

kde $x_{-1}=-h$.

Chyba aproximace je druhého řádu. Okrajovou podmínku $\alpha_0u(0)-\beta_0u'(0)=g_0$ tak nahradíme diferenční rovnicí

$$\alpha_0U_0-\beta_0\frac{U_1-U_{-1}}{2h}=g_0\quad(\beta_0\neq0). \tag{♡}$$

Je zde však navíc další neznámá U_{-1} a potřebujeme proto připojit ještě jednu rovnici.

Nejjednodušeji to provedeme tak, že žádáme platnost rovnice pro vnitřní uzly i pro hraniční uzel x_0, tj. platnost rovnice

$$-\frac{1}{h^2}(U_{-1}-2U_0+U_1)+q(x_0)U_0=f(x_0). \tag{♢}$$

(Jde vlastně o aproximaci diferenciální rovnice v bodě x0 = 0).

Fiktivní hodnotu U_{-1},která nemá význam aproximace přesného řešení naší okrajové úlohy (neboť toto řešení uvažujeme pouze na intervalu $\langle0,1\rangle$), z rovnic (♡), (♢) vyloučíme

$$U_{-1}=h^2\big(q_0U_0-f_0\big)+2U_0-U_1,$$

dosadíme

$$\alpha_0U_0-\beta_0\frac{U_1-h^2(q_0U_0-f_0)-2U_0+U_1}{2h}=g_0$$

a dospějeme k diferenční rovnici

$$\big(\alpha_0+\frac{1}{2}\beta_0hq_0\big)U_0-\beta_0\frac{U_1-U_0}{h}=g_0+\frac{1}{2}\beta_0hf_0,$$

která aproximuje okrajovou podmínku v bodě $x_0=0$ a kterou dostatečně hladké přesné řešení naší okrajové úlohy splňuje s chybou řádové velikosti $O(h^2)$. Rovnici upravíme na tvar

$$\boxed{\frac{1}{h^2}\big[(1+\frac{\alpha_0h}{\beta_0}+\frac{1}{2}h^2q_0)U_0-U_1\big]=\frac{g_0}{h\beta_0}+\frac{1}{2}f_0}$$

a připojíme ji k diferenčním rovnicím pro vnitřní uzly.

Dá se ukázat, že **chyba přibližného řešení je** v tomto případě velikosti $O(h^2)$.

Kapitola 10. Numerické integrování

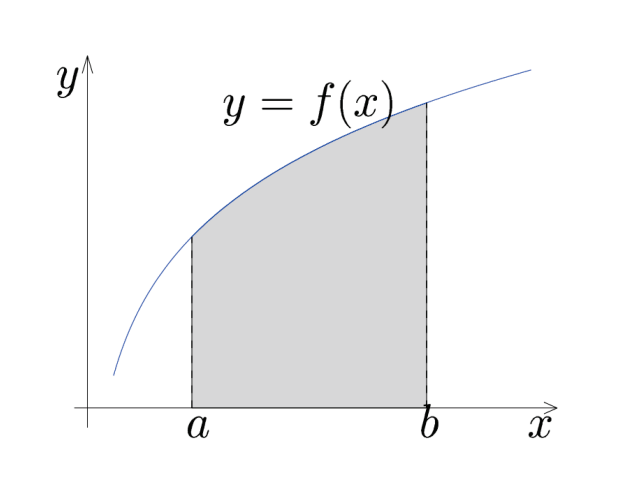
Numerický výpočet hodnoty určitého integrálu

Formulace: Mějme na $\langle a,b\rangle$ dánu integrovatelnou funkci $f = f(x)$. Naším cílem je určit přibližnou hodnotu určitého integrálu

$$I(f) = \int_a^b f(x)\,dx.$$

Poznámka:

Geometrický význam integrálu $I(f)$ (viz obrázek) je obsah plochy mezi grafem funkce f a osou x na intervalu $\langle a,b\rangle$.



Numerické metody výpočtu integrálu užíváme zejména tehdy, když $I(f)$ není možno spočítat analyticky (velmi častý případ) nebo je sice analytické řešení možné, ale je velmi pracné. V případě, že máme zadánu funkci f tabulkou, není ani jiný přístup možný.

Přirozený princip numerických metod pro výpočet integrálu vychází z aproximace funkce. Danou funkci f nahradíme její vhodnou aproximací φ a jako aproximaci integrálu $I(f)$ prohlásíme hodnotu integrálu $I(\varphi)$, tj.

$$I(f) \approx I(\varphi) = \int_a^b \varphi(x)\,dx.$$

Poznámka:

Narozdíl od výpočtu derivace je výpočet integrálu stabilní, protože je-li φ dobrou aproximací funkce f na intervalu $\langle a,b\rangle$, je integrál $I(\varphi)$ dobrou aproximací $I(f)$.

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

$$\left|\int_a^b f(x)\,dx - \int_a^b \varphi(x)\,dx\right| \le \int_a^b |f(x)-\varphi(x)|\,dx \le (b-a)\underbrace{\sup_{x\in\langle a,b\rangle} |f(x)-\varphi(x)|}_{\varepsilon}$$

Princip většiny metod na výpočet určitého integrálu

$$\int_a^b f(x)\,dx$$

je založen na tom, že interval $\langle a,b\rangle$ rozdělíme na N podintervalů $\langle x_k, x_{k+1}\rangle$ tak, že

$$a = x_0 < x_1 < x_2 < \cdots < x_{N-1} < x_N = b.$$

Na těchto podintervalech nahradíme funkci f polynomem a integrujeme tento polynom.

Vzorce pro výpočet určitého integrálu (tzv. **kvadraturní vzorce**) dělíme na:

na intervalech $\langle x_k, x_{k+1}\rangle$ ... **základní**

přes celý interval $\langle a,b\rangle$ ... **složený** (složený kv. vzorec je součtem základních kv. vzorců)

Pro jednoduchost předpokládáme, že jsou všechny podintervaly $\langle x_k, x_{k+1}\rangle$ stejně velké.

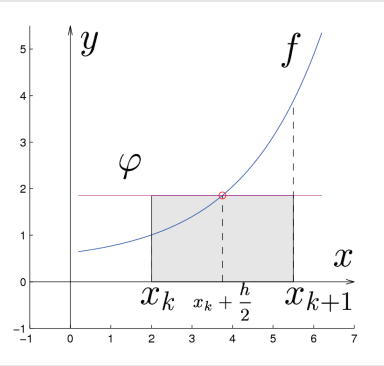
Ekvidistantní uzly potom vyjádříme takto

$$x_k = x_0 + kh, \quad \text{kde} \quad k = 0,1,\ldots,N-1 \quad \text{a} \quad h = \frac{b-a}{N}.$$

Newtonovy-Cotesovy základní kvadraturní vzorce

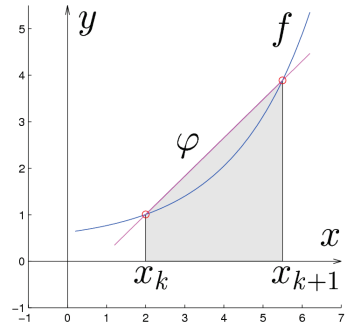
1) **Obdélníkové pravidlo** (f nahrazujeme konstantní funkcí φ)

$$\int_{x_k}^{x_{k+1}} f(x)\,dx \approx \underbrace{h\cdot f(x_k+\frac{h}{2})}_{\equiv R_Z(f,h)}$$



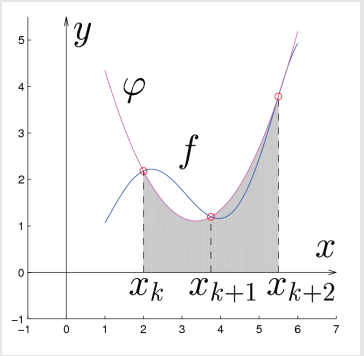
2) **Lichoběžníkové pravidlo** (f nahrazujeme lineární funkcí φ)

$$\int_{x_k}^{x_{k+1}} f(x)\,\mathrm{d}x \approx \underbrace{\frac{h}{2}\left[f(x_k)+f(x_{k+1})\right]}_{\equiv\, T_Z(f,h)}$$



3) **Simpsonovo pravidlo** (f nahrazujeme kvadratickou funkcí φ)

$$\int_{x_k}^{x_{k+2}} f(x)\,\mathrm{d}x \approx \underbrace{\frac{h}{3}\left[f(x_k)+4f(x_{k+1})+f(x_{k+2})\right]}_{\equiv\, S_Z(f,h)}$$

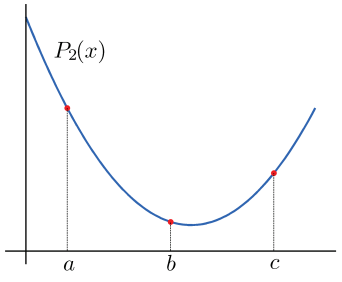


Odvození Simpsonova pravidla

Např. pomocí Lagrangeova interpolačního polynomu:

$$P_2(x) = f(a)l_a(x) + f(b)l_b(x) + f(c)l_c(x)$$

$$l_a(x) = \frac{(x-b)(x-c)}{(a-b)(a-c)} = \frac{(x-b)(x-c)}{2h^2}$$
$$l_b(x) = \frac{(x-a)(x-c)}{(b-a)(b-c)} = \frac{(x-a)(x-c)}{-h^2}$$
$$l_c(x) = \frac{(x-a)(x-b)}{(c-a)(c-b)} = \frac{(x-a)(x-b)}{2h^2}$$



Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

$$\int_a^c P_2(x)\,dx = \frac{f(a)}{2h^2}\int_a^c (x-b)(x-c)\,dx - \frac{f(b)}{h^2}\int_a^c (x-a)(x-c)\,dx + \frac{f(c)}{2h^2}\int_a^c (x-a)(x-b)\,dx =$$

$$= \frac{f(a)}{2h^2}\left[\frac{x^3}{3} - \frac{x^2}{2}(b+c) + xbc\right]_a^c - \frac{f(b)}{h^2}\left[\frac{x^3}{3} - \frac{x^2}{2}(a+c) + xac\right]_a^c + \frac{f(c)}{2h^2}\left[\frac{x^3}{3} - \frac{x^2}{2}(a+b) + xab\right]_a^c =$$

$$= \frac{f(a)}{2h^2}\underbrace{\left[\frac{c^3}{3} - \frac{a^3}{3} - \left(\frac{c^2}{2} - \frac{a^2}{2}\right)(b+c) + (c-a)bc\right]}_{(*)} - \frac{f(b)}{h^2}\underbrace{\left[\frac{c^3}{3} - \frac{a^3}{3} - \left(\frac{c^2}{2} - \frac{a^2}{2}\right)(a+c) + (c-a)ac\right]}_{(**)} +$$
$$+ \frac{f(c)}{2h^2}\underbrace{\left[\frac{c^3}{3} - \frac{a^3}{3} - \left(\frac{c^2}{2} - \frac{a^2}{2}\right)(a+b) + (c-a)ab\right]}_{(***)}$$

$$(*)\quad \frac{1}{6}(c-a)\left[2c^2 + ac + 2a^2 - 3(a+c)(b+c) + 6bc\right] =$$
$$= \frac{2h}{6}\left[2c^2 + 2ac + 2a^2 - 3(a+b)c - 3c^2 - 3ab + 6bc\right] =$$
$$= \frac{2h}{6}\left[-c^2 - ac + 3bc + 2a^2 - 3ab\right] =$$
$$= \frac{2h}{6}\Big[\underbrace{\underbrace{a^2-c^2}_{(a-c)(a+c)}}_{-2h} + 3b\underbrace{(c-a)}_{2h} + a\underbrace{(a-c)}_{-2h}\Big] =$$
$$= \frac{2h}{6}\Big[-(a+c)2h + 2h3b - 2ha\Big] =$$
$$= -\frac{4h^2}{6}\Big[a + c - 3b + a\Big] =$$
$$= -\frac{4h^2}{6}\Big(\underbrace{2a-2b}_{-2h} + \underbrace{c-b}_{h}\Big) =$$
$$= \frac{4h^3}{6} = \frac{2}{3}h^3$$

$(**)$ $\ldots = -\frac{1}{6}(2h)^3$ viz pomocný výpočet pro odvození lichoběžníkového pravidla (slide 10.5.)

$(***)$ $\ldots = \frac{2}{3}h^3$ stejně jako $(*)$ – plyne ze symetrie

$$\int_a^c P_2(x)\,dx = \frac{f(a)}{2h^2}\frac{2}{3}h^3 + \frac{f(b)}{h^2}\frac{1}{6}(2h)^3 + \frac{f(c)}{2h^2}\frac{2}{3}h^3 = \frac{h}{3}\Big[f(a) + 4f(b) + f(c)\Big] = T_Z(f,h)$$

Příklad:

Pomocí základních Newtonových-Cotesových vzorců vypočtěte integrál

$$\int_1^{1,2} \mathrm{e}^x \, dx.$$

Řešení:

(Přesné řešení je $[\mathrm{e}^x]_1^{1,2} = \mathrm{e}^{1,2} - \mathrm{e}^1 \doteq 0{,}601835$.)

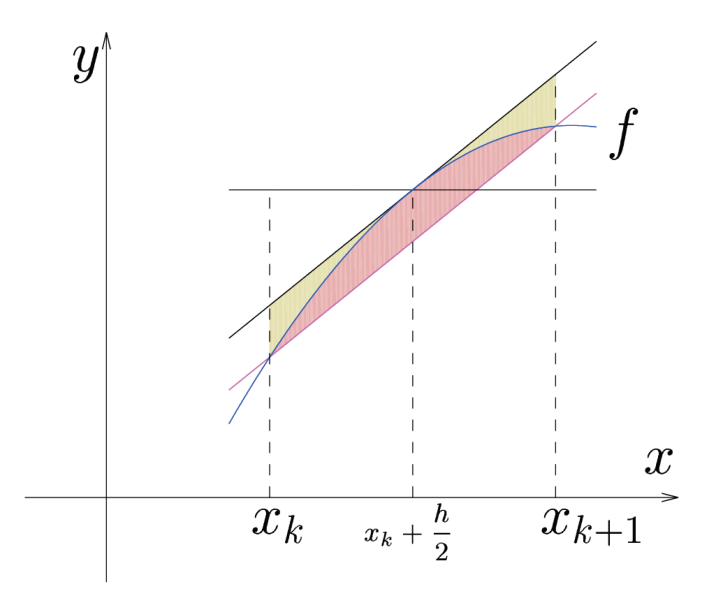
$$R_Z(\mathrm{e}^x;\, 0,2) = 0,2\mathrm{e}^{1,1} \doteq 0{,}600833 \qquad \text{chyba: } 0{,}001002$$

$$T_Z(\mathrm{e}^x;\, 0,2) = \tfrac{0,2}{2}(\mathrm{e}^{1,0} + \mathrm{e}^{1,2}) \doteq 0{,}603839 \qquad \text{chyba: } 0{,}002003$$

$$S_Z(\mathrm{e}^x;\, 0,1) = \tfrac{0,1}{3}(\mathrm{e} + 4\mathrm{e}^{1,1} + \mathrm{e}^{1,2}) \doteq 0{,}601835 \qquad \text{chyba: } 0{,}000000$$

Poznámka:

Všimněme si chyb. U obdélníkového pravidla vyšla chyba menší než u lichoběžníkového, přestože u lichoběžníkového pravidla jsme funkci f aproximovali „lepší“ funkcí φ (lineární). Chyba u Simpsonova pravidla vyšla menší než u ostatních. Tyto výsledky potvrzují vztahy pro chyby jednotlivých vzorců. Fakt, že obdélníkové pravidlo je přesnější než lichoběžníkové můžeme demonstrovat na obrázku:



Základní vzorce se odvodí snadno na základě geometrické interpretace.

Pokud chceme vyjádřit současně i vztahy pro chyby těchto vzorců, musíme použít k odvození Taylorův rozvoj.

$$\int_{x_k}^{x_{k+1}} f(x)\,dx = R_Z(f,h) + \frac{h^3}{24} f''(\xi)$$

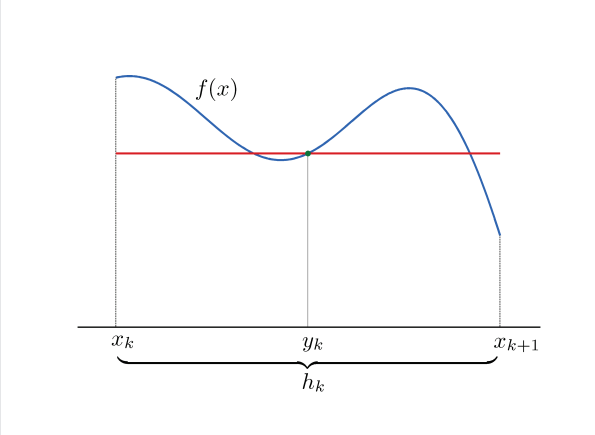
Numerické metody KMA/NM
Josef Daněk 13.2.2o13

$$\int_{x_k}^{x_{k+1}} f(x)\,dx = T_Z(f,h) - \frac{h^3}{12} f''(\xi)$$

$$\int_{x_k}^{x_{k+2}} f(x)\,dx = S_Z(f,h) - \frac{h^5}{90} f^{(4)}(\xi)$$

Odvození pro obdélníkové pravidlo

Předpokládejme, že je integrovaná funkce f dostatečně hladká a použijeme Taylorův polynom.



Označíme

$$h_k = x_{k+1} - x_k, \quad y_k = \frac{x_k + x_{k+1}}{2}$$

$$f(x) = f(y_k) + (x - y_k) f'(y_k) + \frac{1}{2}(x - y_k)^2 f''(\xi_k), \quad \xi_k \in \operatorname{int}\{y_k, x\}$$

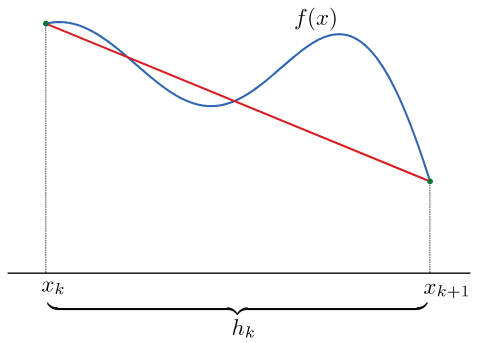
Potom platí:

$$\int_{x_k}^{x_{k+1}} f(x)\,dx = h_k f\Big(\frac{x_k + x_{k+1}}{2}\Big) + \frac{1}{2}\Big[\overbrace{(x_{k+1} - y_k)^2}^{\frac{h_k}{2}} - \overbrace{(x_k - y_k)^2}^{-\frac{h_k}{2}}\Big] f'\Big(\frac{x_k + x_{k+1}}{2}\Big) + f''(\xi_k)\frac{1}{6}\Big[\overbrace{(x_{k+1} - y_k)^3}^{\frac{h_k^3}{8}} - \overbrace{(x_k - y_k)^3}^{-\frac{h_k^3}{8}}\Big]$$

$$\int_{x_k}^{x_{k+1}} f(x)\,dx = \underbrace{h_k f\Big(\frac{x_k + x_{k+1}}{2}\Big)}_{R_Z(f,h_k)} + \underbrace{\frac{h_k^3}{24} f''(\xi_k)}_{\text{chyba metody}}$$

Odvození pro lichoběžníkové pravidlo

Funkci f aproximujeme na $\langle x_k, x_{k+1}\rangle$ lineární funkcí, tj. interpolačním polynomem 1. stupně.



Z aproximací funkce známe:

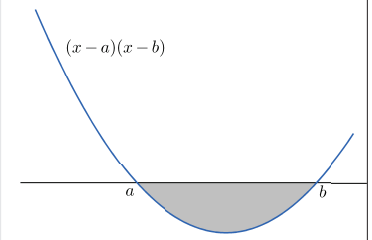
$$f(x) = P_1(x) + \frac{f''(\xi_k)}{2}(x-x_k)(x-x_{k+1}), \quad \xi_k \in (x_k, x_{k+1})$$

Potom platí:

$$\int_{x_k}^{x_{k+1}} f(x)\,dx = \frac{h_k}{2}\big(f(x_k)+f(x_{k+1})\big) + \frac{f''(\xi_k)}{2}\int_{x_k}^{x_{k+1}} (x-x_k)(x-x_{k+1})\,dx$$

pomocný výpočet

$$\begin{aligned}
&\int_a^b (x-a)(x-b)\,dx = \int_a^b \big(x^2 - x(a+b) + ab\big)\,dx = \\
&= \frac{b^3}{3} - \frac{a^3}{3} - \left(\frac{b^2}{2} - \frac{a^2}{2}\right)(a+b) + (b-a)ab = \\
&= \frac{1}{3}(b-a)(b^2+ab+a^2) - \frac{1}{2}(b-a)(a+b)^2 + (b-a)ab = \\
&= \frac{1}{6}(b-a)\left[2b^2+2ab+2a^2-3a^2-6ab-3b^2+6ab\right] = \\
&= \frac{1}{6}(b-a)\left[-a^2+2ab-b^2\right] = -\frac{1}{6}(b-a)^3
\end{aligned}$$



$$\int_{x_k}^{x_{k+1}} f(x)\,dx = \underbrace{\frac{h_k}{2}\big(f(x_k)+f(x_{k+1})\big)}_{T_Z(f,h_k)} - \underbrace{\frac{h_k^3}{12}f''(\xi_k)}_{\text{chyba metody}}$$

Komentář pro Simpsonovo pravidlo

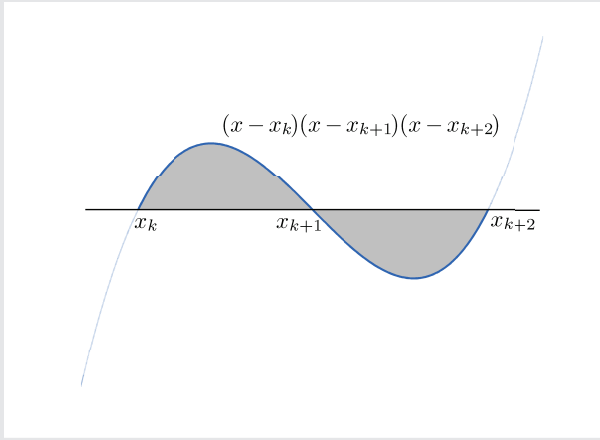
Funkci f aproximujeme na $\langle x_k, x_{k+2}\rangle$ kvadratickou funkcí, tj. interpolačním polynomem 2. stupně.

$$f(x) = P_2(x) + \frac{f'''(\xi)}{6}(x-x_k)(x-x_{k+1})(x-x_{k+2})$$

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

$$\int_{x_k}^{x_{k+2}} f(x)\,dx = S_Z(f,h) + \frac{f'''(\xi)}{6}\int_{x_k}^{x_{k+2}} (x-x_k)(x-x_{k+1})(x-x_{k+2})\,dx \ \ldots$$

Ačkoliv z uvedeného vychází, že chyba by řádově měla být h^4, je chyba o jeden řád vyšší. Důvod je podobný jako u odvození chyby obdélníkového pravidla (integrujeme funkci symetrickou podle středu intervalu).



Jelikož výraz pro chybu základního Simpsonova pravidla obsahuje 4-tou derivaci, je zřejmé, že Simpsonovo pravidlo bude přesně integrovat polynomy až do stupně 3, protože pro ně je 4-tá derivace identicky nulová.

Newton-Cotesovy složené kvadraturní vzorce

Složené kvadraturní vzorce získáme sečtením základních kvadraturních vzorců:

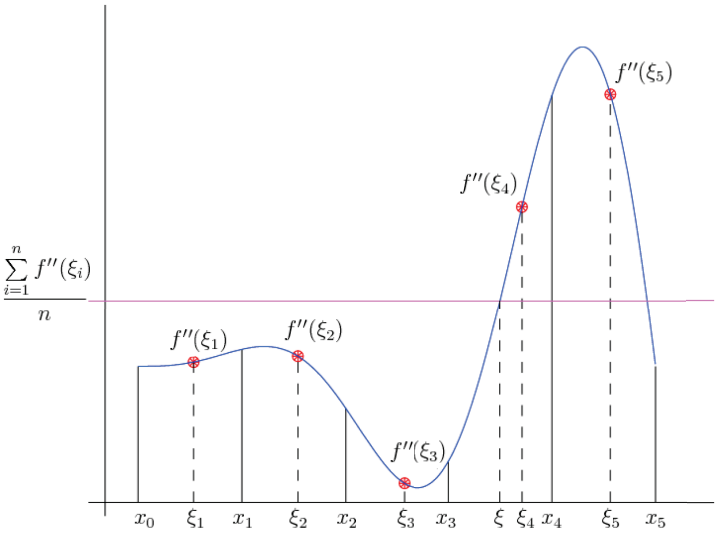
$$\int_a^b f(x)\,dx \;=\; \sum_{k=0}^{N-1}\int_{x_k}^{x_{k+1}} f(x)\,dx \;\approx\; \sum_{k=0}^{N-1}\int_{x_k}^{x_{k+1}} \varphi(x)\,dx$$

$$\begin{aligned}
R(f,h) &\equiv h\cdot\sum_{k=0}^{N-1} f\Big(x_k+\frac{h}{2}\Big) \\
T(f,h) &\equiv \frac{h}{2}\big[f(x_0)+2f(x_1)+2f(x_2)+\cdots+2f(x_{N-1})+f(x_N)\big] = \\
&\quad = h\cdot\Big[\frac{1}{2}f(x_0)+\sum_{k=1}^{N-1} f(x_k)+\frac{1}{2}f(x_N)\Big] \\
S(f,h) &\equiv \frac{h}{3}\big[f(x_0)+4f(x_1)+2f(x_2)+4f(x_3)+ \\
&\quad +\cdots+2f(x_{N-2})+4f(x_{N-1})+f(x_N)\big]
\end{aligned}$$

Pro chyby složených vzorců potom platí:

$$\boxed{\begin{aligned} I(f) &= R(f,h) + (b-a)\,\frac{h^2}{24}\,f''(\xi) \\ I(f) &= T(f,h) - (b-a)\,\frac{h^2}{12}\,f''(\xi) \\ I(f) &= S(f,h) - (b-a)\,\frac{h^4}{180}\,f^{(4)}(\xi) \end{aligned}}$$

$$\sum_{k=1}^{N} \underbrace{\frac{h^3}{24}\,f''(\xi_k)}_{\text{chyba základního vzorce na } \langle x_{k-1}, x_k\rangle} = \frac{h^3}{24}\sum_{k=1}^{N} f''(\xi_k) = \frac{h^3}{24}\,N\,f''(\xi) = \frac{h^3}{24}\,\frac{b-a}{h}\,f''(\xi)$$



Průměr hodnot leží mezi minimální a maximální hodnotou:

$$\min_k f''(\xi_k) \;\le\; \frac{1}{N}\sum_k f''(\xi_k) \;\le\; \max_k f''(\xi_k)$$

Ze spojitosti funkce $f''(x)$ vyplyne:

$$\exists \xi \in (x_0, x_N): \quad f''(\xi) = \frac{1}{N}\sum_k f''(\xi_k)$$

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Jak dosáhnout požadovanou přesnost ?

Ze vzorců lze odhadnout velikost chyby, případně určit krok h tak, aby chyba byla menší než předem zadaná tolerance.

Příklad Určete h tak, aby chyba složeného lichoběžníkového pravidla pro výpočet

$$\boxed{I = \int_2^3 \frac{1}{(x-1)}\,dx}$$

byla nejvýše 10^{-3}.

Musí platit:

$$\boxed{\frac{(b-a)}{12}h^2 \max_{x\in\langle 2,3\rangle} |f''(x)| \le 10^{-3}}$$

$\Rightarrow$ je nutné odhadnout f'':

$$f' = -\frac{1}{(x-1)^2}$$

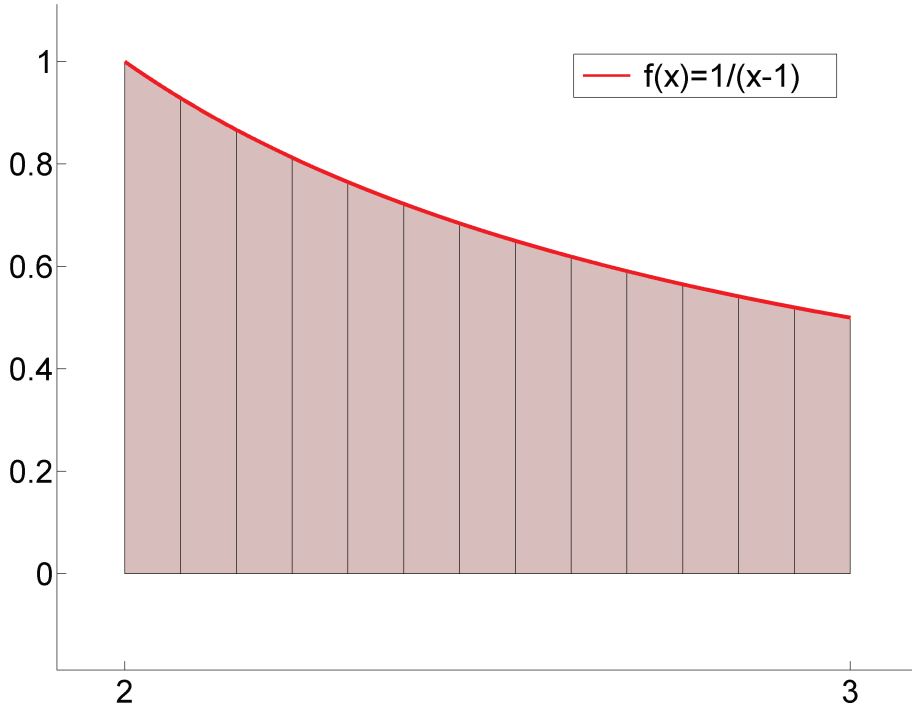
$$f'' = \frac{2}{(x-1)^3} \quad \text{na } \langle 2,3\rangle \text{ je } f'' > 0 \text{ (kladná)}$$

$$f''' = -\frac{3}{(x-1)^4} < 0 \quad \Rightarrow \quad f'' \text{ je klesající}$$

$$\Rightarrow \quad \max_{x\in\langle 2,3\rangle} |f''(x)| = f''(2) = \frac{2}{(2-1)^3} = 2$$

$$\boxed{\frac{1}{12}h^2 \cdot 2 \le 10^{-3}} \quad \Rightarrow \quad h^2 \le 6\cdot 10^{-3}$$

$$\Rightarrow \quad N = \frac{b-a}{h} = \frac{1}{\sqrt{6\cdot 10^{-3}}} \doteq 12{,}9 \quad \Rightarrow \quad \text{nejbližší vyšší } \boxed{N = 13} \quad \Rightarrow \quad \boxed{h = \frac{1}{13}}$$



$$T\left(\frac{1}{(x-1)}, \frac{1}{13}\right) = \ldots \doteq 0,69352$$

Přesná hodnota: $I = \left[\ln|x-1|\right]_2^3 = \ln 2 - \ln 1 = \ln 2 \doteq 0,69315$

Skutečná chyba: $3,7 \cdot 10^{-4} \leq 10^{-3}$

Nevýhody tohoto postupu:

- výrazy pro chybu obsahují derivace (často vysokého řádu), které není lehké odhadnout
- výsledné odhady jsou většinou velmi pesimistické
- Newton-Cotesovy vzorce nejsou konvergentní
  (zvyšujeme-li řád vzorce, nemusí konvergovat aproximace integrálů k teoretické hodnotě)
- pro odhad chyby je vhodné užít metodu poloviního kroku (Richardsonova extrapolace)

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Richardsonova extrapolace

Stručně si připomeňme princip Richardsonovy extrapolace, kterou jsme již používali pro zpřesňování při výpočtu hodnoty derivace funkce.

Předpokládejme, že výraz pro chybu má tvar

$$e(f) = h^k M, \quad h = \frac{b-a}{N}$$

Přesná hodnota integrálu je potom

$$I = K(h) + h^k M. \qquad (\star)$$

Integrál vypočteme stejným vzorcem, ale s krokem $\frac{h}{2}$. Dostaneme

$$I = K\left(\frac{h}{2}\right) + \underbrace{\left(\frac{h}{2}\right)^k M_1}_{\text{ozn. } \varepsilon} \qquad \Rightarrow \qquad h^k = \frac{\varepsilon\, 2^k}{M_1} \qquad (\star\star)$$

Dosadíme-li h^k do $(\star)$, získáme

$$I = K(h) + \frac{\varepsilon\, 2^k M}{M_1} \qquad (\star\star\star)$$

Předpokládáme-li, že se hodnota derivace ve výrazu $e(f)$ pro chybu příliš nemění (tj. $M \approx M_1$), potom $\frac{M}{M_1} \approx 1$ a pro $(\star\star)$ a $(\star\star\star)$ musí platit

$$K\left(\frac{h}{2}\right) + \varepsilon \;\approx\; K(h) + 2^k \varepsilon$$

Odtud plyne odhad chyby ε

$$\varepsilon \;\approx\; \frac{1}{2^k - 1}\left[K\left(\frac{h}{2}\right) - K(h)\right]$$

a přesnější hodnota integrálu je potom

$$\boxed{I = K\left(\frac{h}{2}\right) + \frac{1}{2^k - 1}\left[K\left(\frac{h}{2}\right) - K(h)\right]}$$

k ... řád eliminované chyby

Algoritmus (Pro složené lichoběžníkové pravidlo)

Pro $s = 0, 1, 2, \ldots, S$
  $T_{s,0} = T(f, h_s)$
  Pro $k = 1, 2, \ldots, s$
    $T_{s,k} = T_{s,k-1} + \frac{T_{s,k-1} - T_{s-1,k-1}}{4^k - 1}$

$h_0 = b - a$
$h_1 = \frac{1}{2} h_0$
$\vdots$
$h_s = \frac{1}{2^s} h_0$

Schéma

h T_{00}

$\frac{h}{2}$ T_{10} T_{11}

$\frac{h}{4}$ T_{20} T_{21} T_{22}

$\frac{h}{8}$ T_{30} T_{31} T_{32} T_{33}

Všechny hodnoty $T_{s,k}$ jsou aproximacemi původního integrálu.

Pro funkci f integrovatelnou v Riemannově smyslu platí

$$T_{sk} \to I(f) \quad \text{pro } s \to \infty,\ k = 0,1,\dots$$

a také

$$T_{kk} \to I(f) \quad \text{pro } k \to \infty.$$

Dále se dá ukázat, že celá procedura je numericky stabilní.

Příklad

Pomocí lichoběžníkového pravidla vypočtěte $\int_1^5 \ln x\,dx$. Ke zpřesnění použijte Richardsonovu extrapolaci.

Řešení: Pro rozvoj chyby lichoběžníkového pravidla platí

$$I = T(f,h) + \underbrace{a_1h^2}_{\text{tab. k=2}} + \underbrace{a_2h^4}_{\text{tab. k=4}} + a_3h^6 + \dots$$

Výsledky opět zapíšeme do tabulky

h	$T(f,h)$	1. zpřesnění $(k=2)$	2. zpřesnění $(k=4)$
4	$\frac{4}{2}(\ln 1 + \ln 5) = 3,2188$		
2	$\frac{2}{2}(\ln 1 + 2\ln 3 + \ln 5) = 3,8066$	$\frac{3,8066 - 3,2188}{3} + 3,8066 = \underline{4,0025}$	
1	$\frac{1}{2}(\ln 1 + 2\ln 2 + 2\ln 3 + 2\ln 4 + \ln 5) = 3,9827$	$\frac{3,9827 - 3,8066}{3} + 3,9827 = \underline{4,0414}$	$\frac{4,0414 - 4,0025}{15} + 4,0414 = \underline{\underline{4,04399}}$

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Pro kontrolu uveďme přesnou hodnotu integrálu:

$$\int_1^5 \ln x\,dx = \left|\begin{matrix} u = \ln x & v' = 1 \\ u' = \frac{1}{x} & v = x \end{matrix}\right| = [x\ln x]_1^5 - \int_1^5 dx = 5\ln 5 - 4 \doteq \underline{\underline{4,04719}}$$

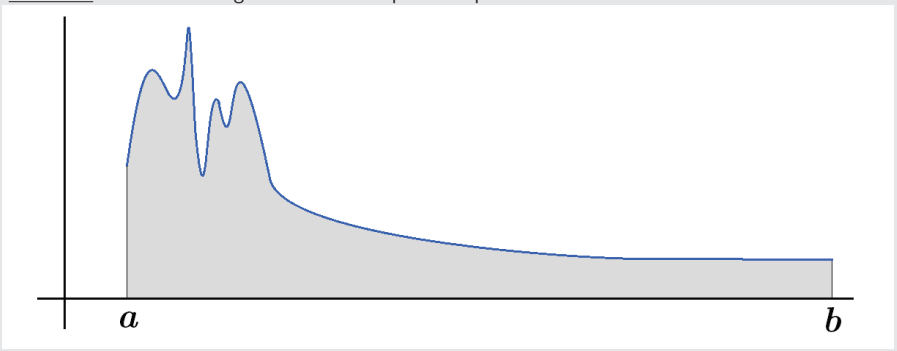
Poznámka

Metoda Richardsonovy extrapolace pro lichoběžníkové pravidlo se nazývá **Rombergova metoda**.

Adaptivní integrování

- intervaly integrace nejsou dány dopředu
- určují se na základě splnění testu chyby založeném na odhadu pomocí metody poloviničního kroku

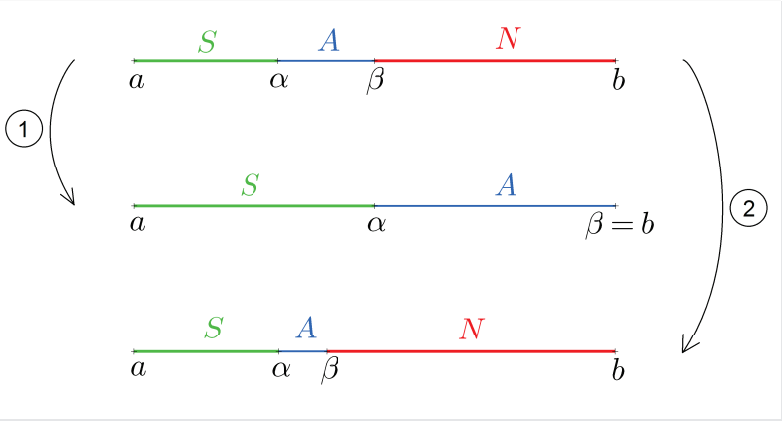
Motivace: Pokud má integrovaná funkce např. tento průběh



je zřejmé, že na druhé části intervalu stačí pro splnění zadané tolerance uvažovat větší kroky, než v první části.

Stavy

S ... interval, na kterém je zajištěno splnění chybového testu
A ... aktivní interval integrace
N ... interval, přes který se ještě nezapočítal dílčí integrál



Změny stavu:

(1) Je splněna podmínka na velikost chyby na intervalu A

(2) Není splněna podmínka na velikost chyby na intervalu A

Test chyby: (pomocí metody poloviního kroku) – interval $\langle\alpha,\beta\rangle$ rozpůlíme a použijeme stejný vzorec

$$\varepsilon_f(\alpha,\beta) \approx \frac{1}{2^k-1}\left[I_{\langle\alpha,\beta\rangle}\left(\frac{h}{2}\right) - I_{\langle\alpha,\beta\rangle}(h)\right], \qquad k - \text{řád chyby vzorce}$$

$$\boxed{\varepsilon_f(\alpha,\beta) \le \varepsilon\,\frac{\beta-\alpha}{b-a}}$$

ε – celková požadovaná přesnost

Algoritmus

na začátku:

$A = \langle a,b\rangle$
$N = \emptyset$
$S = \emptyset$
$I_S = 0 \qquad (I_S \approx \int_a^\alpha f(x)\,dx)$

(1) je splněn TEST CHYBY:

(i) $I_S := I_S + I_A$

(ii) $S = S \cup A; \quad A = N$

(2) není splněn TEST CHYBY:

(i) A rozpůlíme, tj. $A = \langle\alpha, \frac{\alpha+\beta}{2}\rangle$

(ii) $N := N \cup \langle\frac{\alpha+\beta}{2}, \beta\rangle$

(iii) nový TEST CHYBY

(1) a (2) opakujeme dokud $S \neq \langle a,b\rangle$

Příklad

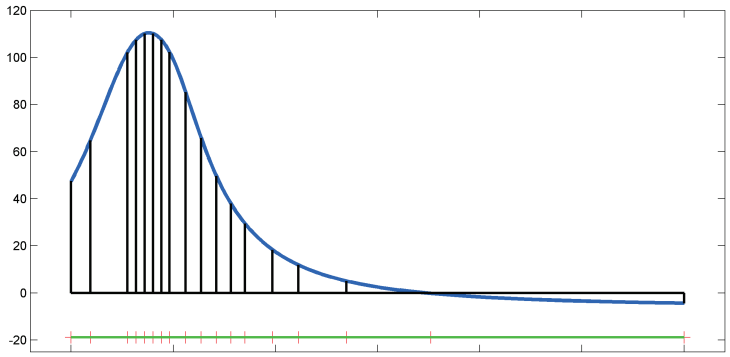
Použijte adaptivní přístup pro výpočet

$$\int_0^3 \frac{1}{(0{,}3x-0{,}1)^2+0{,}01} + \frac{1}{(x-0{,}5)^2+0{,}04} - 6\,dx$$

tak, aby výsledná chyba aproximace integrálu byla menší než 0,25.

Pro výpočet použijte obdélníkové, lichoběžníkové i Simpsonovo pravidlo.

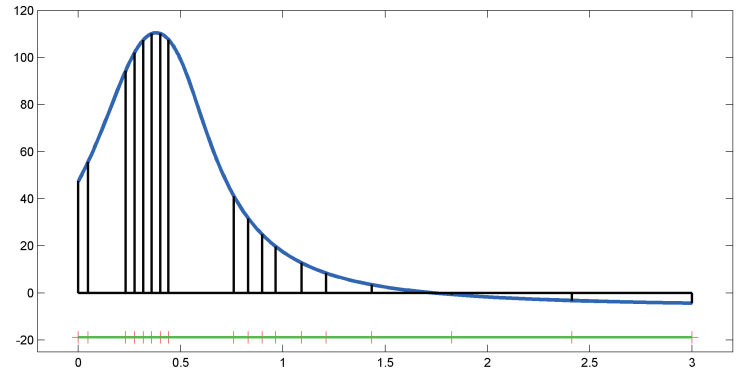
Obdélníkové pravidlo



výsledky v MATLABu

```
------------------------------------------------------------------
  Adaptivni numericky vypocet urciteho integralu funkce
  f(x)=1 /((0.3*x-.1)^2+.01)+1/((x-.5)^2+.04)-6
  na intervalu <0.000000,3.000000>
  se zadanou presnosti 0.250000
  Pro vypocet se pouzije obdelnikove pravidlo I_O.
------------------------------------------------------------------
  Presna hodnota integralu ....... 69.800931
  Vypoctena hodnota integralu .... 69.784747
  Skutecna chyba ................. 0.016184
  Odhadnuta chyba ................ 0.110713
  Pocet podintervalu ............. 17
  Celkovy pocet deleni intervalu
    pro dodrzeni odhadu chyby .... 94
------------------------------------------------------------------
```

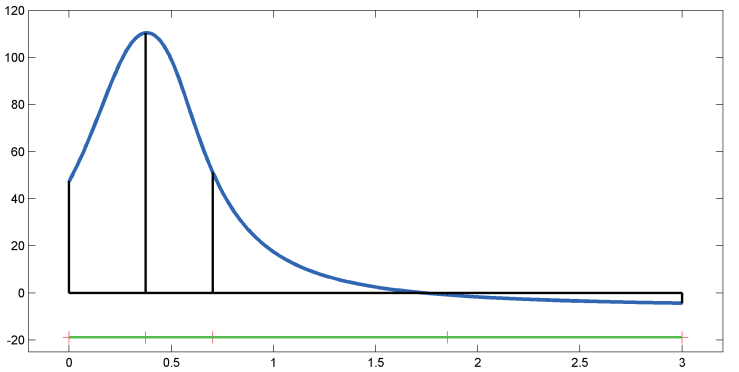
Lichoběžníkové pravidlo



výsledky v MATLABu


```
------------------------------------------------------------------
  Adaptivni numericky vypocet urciteho integralu funkce
  f(x)=1 /((0.3*x-.1)^2+.01)+1/((x-.5)^2+.04)-6
  na intervalu <0.000000,3.000000>
  se zadanou presnosti 0.250000
  Pro vypocet se pouzije lichobeznikove pravidlo I_L.
------------------------------------------------------------------
  Presna hodnota integralu ....... 69.800931
  Vypoctena hodnota integralu .... 69.686611
  Skutecna chyba ................. 0.114320
  Odhadnuta chyba ................ -0.084305
  Pocet podintervalu ............. 17
  Celkovy pocet deleni intervalu
    pro dodrzeni odhadu chyby .... 89
------------------------------------------------------------------
```

Simpsonovo pravidlo



výsledky v MATLABu

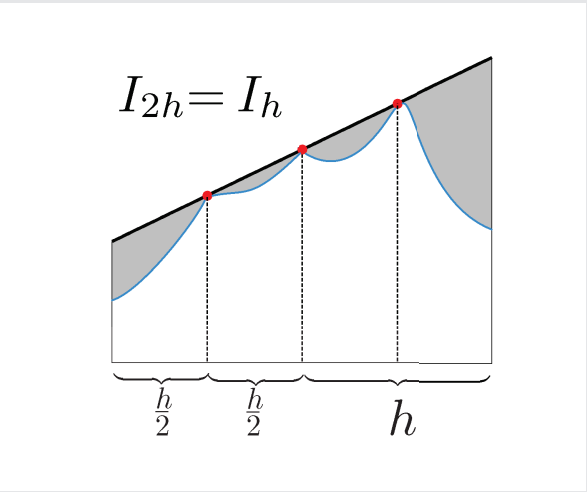
```
-----------------------------------------------------------------
  Adaptivni numericky vypocet urciteho integralu funkce
  f(x)=1 /((0.3*x-.1)^2+.01)+1/((x-.5)^2+.04)-6
  na intervalu <0.000000,3.000000>
  se zadanou presnosti 0.250000
  Pro vypocet se pouzije Simpsonovo pravidlo I_S.
-----------------------------------------------------------------
  Presna hodnota integralu ....... 69.800931
  Vypoctena hodnota integralu .... 69.849993
  Skutecna chyba ................. -0.049061
  Odhadnuta chyba ................ -0.073144
  Pocet podintervalu ............. 4
  Celkovy pocet deleni intervalu
    pro dodrzeni odhadu chyby .... 11
-----------------------------------------------------------------
```

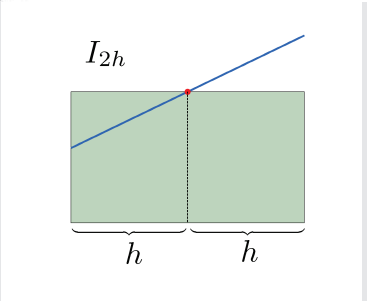
Poznámka

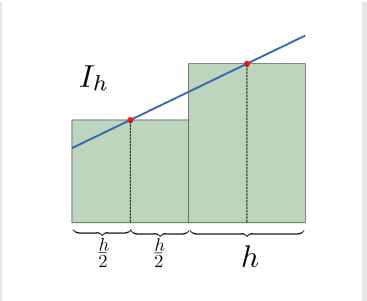
Odhadujeme-li chybu pomocí metody poloviního kroku, nemusí být skutečná chyba menší než zadaná tolerance.

Příklady v nichž je splněn TEST CHYBY, ale chyba je ve skutečnosti větší než zadaná tolerance

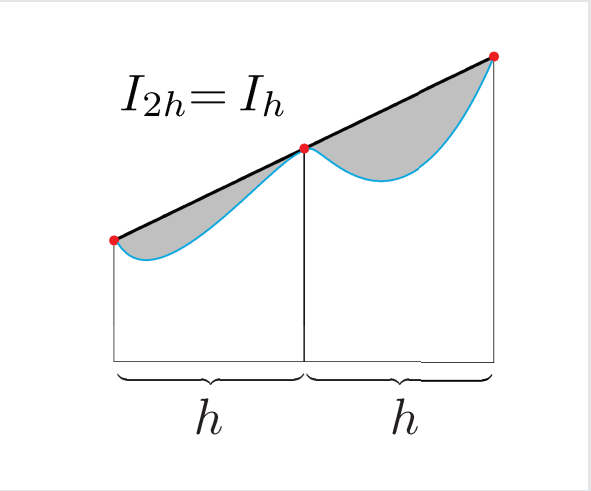
- Obdélníkové pravidlo:





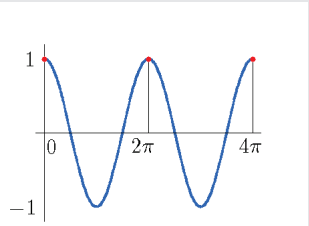


- Lichoběžníkové pravidlo:



Např:

$$\int_0^{4\pi} \cos x \, dx, \qquad \text{tolerance} \quad \varepsilon = 10^{-5}$$



$$I_{4\pi} = 1 \cdot 4\pi \qquad I_{2\pi} = 1 \cdot 2\pi + 1 \cdot 2\pi = 4\pi$$

Odhad chyby je

$$\left(I_{2\pi} - I_{4\pi}\right) \cdot \frac{1}{3} = 0$$

Přesná hodnota je

$$\int_0^{4\pi} \cos x \, dx = 0$$

Chyba skutečná je

$$4\pi$$

Poznámka:

Newtonovy-Cotesovy vzorce používají $(m+1)$ ekvidistantních uzlů a integrují přesně polynomy až do m-tého, případně. $(m+1)$-ního stupně (máme na mysli základní vzorce na intervalu (x_k, x_{k+m})).

Pro zvýšení přesnosti by se mohlo zdát výhodné použít více uzlů a funkci f aproximovat polynomem vyššího řádu. Ze zkušeností z aproximace funkce polynomem ovšem víme, že limitní případ polynomu stupně $m \to \infty$ nemusí odpovídat původní funkci (říkáme, že Newton-Cotesovy vzorce nejsou konvergentní).

Gaussovy kvadraturní vzorce

Princip: Snažíme se , aby kvadraturní vzorec integroval přesně polynomy co možná nejvyššího řádu.

Obecně kvadraturní vzorec (základní) uvažujeme ve tvaru

$$K(f) = \sum_{i=0}^{m} w_i f(x_i),$$

kde w_i jsou tzv. **váhy** a x_i jsou **uzly**.

Máme-li na základním intervalu $m+1$ bodů, potom nejvyšší možný stupeň polynomu, který ještě kvadraturní vzorec integruje přesně, je $2m+1$ (mluvíme o tzv. **algebraickém řádu přesnosti**).

Počet parametrů kvadraturního vzorce je $2m+2$

– polovina pro váhy w_i

– polovina pro uzly x_i

(Newton-Cotesovy vzorce integrovaly přesně polynomy do stupně $\sim m$.)

Cenou za vyšší přesnost budou ovšem neekvidistantní uzly.

Příklad: Odvoďte pro interval $\langle -1, 1\rangle$ základní Gaussův kvadraturní vzorec pro $m = 0$ (tj. v intervalu uvažujeme pouze jeden uzel).

Řešení:

Kvadraturní vzorec pro $m = 0$ má tvar

$$K(f) = w_0 f(x_0),$$

kde vystupují 2 neznámé w_0 a x_0.

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Vzorec musí přesně integrovat:

1) konstantu

$$\int_{-1}^{1} b \, dx = 2b \overset{\text{pož.}}{=} w_0 \cdot \overbrace{f(x_0)}^{b} \quad \Rightarrow \quad w_0 = 2.$$

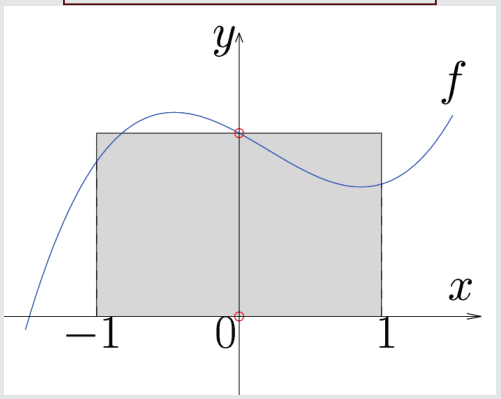
2) lineární funkci

$$\int_{-1}^{1} (ax+b)\, dx = \left[a\frac{x^2}{2} + bx\right]_{-1}^{1} = \underbrace{\frac{a}{2} - \frac{a}{2}}_{=0} + 2b \overset{\text{pož.}}{=} w_0 \cdot \overbrace{f(x_0)}^{ax_0+b}$$

$$\Rightarrow \quad 2b = 2(ax_0 + b) \Rightarrow x_0 = 0.$$

Jednobodový základní Gaussův kvadraturní vzorec je

$$K(f) = \int_{-1}^{1} f(x)\,\mathrm{d}x = 2f(0) + \underbrace{\frac{1}{3} f''(\xi)}_{\text{chyba}}$$



Příklad: Odvoďte pro interval $\langle -1, 1\rangle$ základní Gaussův kvadraturní vzorec pro $m = 1$ (tj. v intervalu uvažujeme 2 uzly).

Řešení:

Kvadraturní vzorec pro $m = 1$ má tvar

$$K(f) = w_0 f(x_0) + w_1 f(x_1),$$

kde vystupují 4 neznámé w_0, w_1, x_0 a x_1.

Vzorec musí přesně integrovat polynom až 3 stupně:

$$\int_{-1}^{1}\left(ax^3+bx^2+cx+d\right)dx=\left[a\frac{x^4}{4}+b\frac{x^3}{3}+c\frac{x^2}{2}+dx\right]_{-1}^{1}=0\cdot a+\frac{2}{3}\cdot b+0\cdot c+2\cdot d \overset{\text{pož.}}{=}$$

$$\overset{\text{pož.}}{=} w_0\underbrace{\left(ax_0^3+bx_0^2+cx_0+d\right)}_{f(x_0)}+w_1\underbrace{\left(ax_1^3+bx_1^2+cx_1+d\right)}_{f(x_1)}=K(f).$$

Soustava nelineárních rovnic pro 4 neznámé:

$$\begin{aligned}
a:&\quad w_0x_0^3+w_1x_1^3 &=& \ 0 &\quad (1)\\
b:&\quad w_0x_0^2+w_1x_1^2 &=& \ \tfrac{2}{3} &\quad (2)\\
c:&\quad w_0x_0+w_1x_1 &=& \ 0 &\quad (3)\\
d:&\quad w_0+w_1 &=& \ 2 &\quad (4)
\end{aligned}$$

---

(1) − (3): $w_0x_0(x_0^2-1)+w_1x_1(x_1^2-1)=0.$

(2) − (4): $w_0(x_0^2-1)+w_1(x_1^2-1)=-\frac{4}{3}$ $\quad/\cdot(-x_1)^{\dagger}$ $\quad/\cdot(-x_0)^{\ddagger}$

$$\left.\begin{array}{lll}
\dagger & \underbrace{w_0(x_0-x_1)}(x_0^2-1) &= \tfrac{4}{3}x_1\\
\ddagger & \underbrace{w_1(x_1-x_0)}(x_1^2-1) &= \tfrac{4}{3}x_0\\
\text{(3) a (4)}\Rightarrow & w_1 &= 2-w_0\\
& w_0x_0+(2-w_0)x_1 &= 0\\
& \underbrace{w_0(x_0-x_1)} &= -2x_1
\end{array}\right\}\Rightarrow$$

$$-2x_1(x_0^2-1)=\frac{4}{3}x_1\ \Rightarrow\ -2(x_0^2-1)=\frac{4}{3}\ \Rightarrow\ x_0^2-1=-\frac{2}{3}\ \Rightarrow\ x_0^2=\frac{1}{3}$$

$$\Rightarrow x_0=-\sqrt{\frac{1}{3}}$$

analogicky:

$$\left.\begin{array}{lll}
\text{(3) a (4)}\Rightarrow & w_0 &= 2-w_1\\
& (2-w_1)x_0+w_1x_1 &= 0\\
& \underbrace{w_1(x_1-x_0)} &= -2x_0
\end{array}\right\}\Rightarrow$$

$$-2x_0(x_1^2-1)=\frac{4}{3}x_0\ \Rightarrow\ -2(x_1^2-1)=\frac{4}{3}\ \Rightarrow\ x_1^2-1=-\frac{2}{3}\ \Rightarrow\ x_1^2=\frac{1}{3}$$

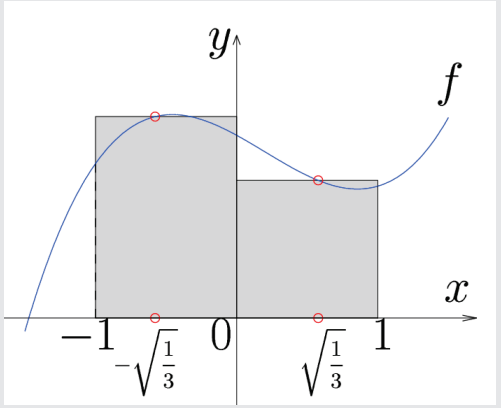
$$\Rightarrow x_1=\sqrt{\frac{1}{3}}$$

$$\begin{array}{lll}
\text{(3) a (4):} & w_0+w_1 &= 2\\
& \sqrt{\frac{1}{3}}w_0-\sqrt{\frac{1}{3}}w_1 &= 0\Rightarrow w_0=w_1\Rightarrow w_0=w_1=1
\end{array}$$

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Dvoubodový základní Gaussův kvadraturní vzorec je

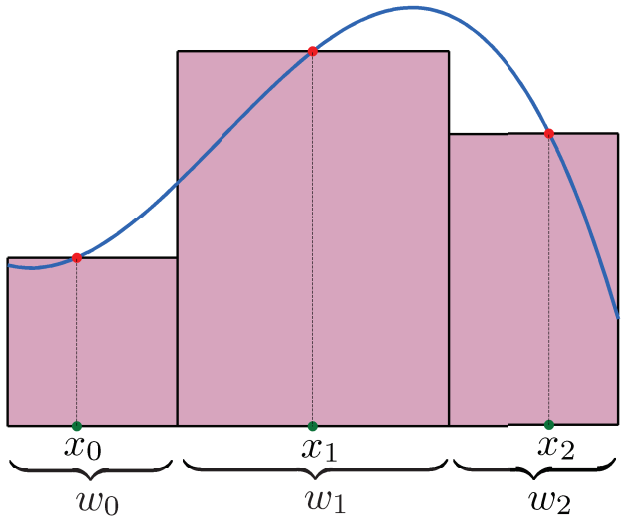
$$K(f)=\int_{-1}^{1}f(x)\,dx=f\left(-\frac{\sqrt{3}}{3}\right)+f\left(\frac{\sqrt{3}}{3}\right)+\underbrace{\frac{1}{135}f^{(4)}(\xi)}_{\text{chyba}}$$



Poznámka:

Další základní Gaussův kvadraturní vzorec (tříbodový, tj. pro $m=2$) vypadá na intervalu $\langle -1,1\rangle$ takto:

$$K(f)=\int_{-1}^{1}f(x)\,dx=\frac{5}{9}f\left(-\sqrt{\frac{3}{5}}\right)+\frac{8}{9}f(0)+\frac{5}{9}f\left(\sqrt{\frac{3}{5}}\right)+\underbrace{\frac{1}{15750}f^{(6)}(\xi)}_{\text{chyba}}.$$



Poznámka: Koeficienty a uzly vzorců vyšších řádů jsou uvedeny v tabulkách.

Poznámka: To, že jsme vyjádřili $\int_{-1}^{1} f(x)\,dx$ neubírá nic na obecnosti, můžeme totiž libovolný interval $\langle a,b\rangle$ transformovat na $\langle -1,1\rangle$ a použít odvozené vztahy.

Poznámka: Gaussovy kvadraturní vzorce jsou konvergentní.

Příklad

Vypočtěte $\int_1^{1,2} \mathrm{e}^x\,dx$ použitím jedno- a dvoubodového základního Gaussova kvadraturního vzorce.

Řešení:

$x_i = 1,1 + 0,1 \cdot x_i^{(0)}$,
$w_i = \frac{1}{2}(1,2-1)w_i^{(0)} = 0,1w_i^{(0)}$.

$n = 0$

$\int_1^{1,2} f(x)\,dx \approx 0,2 \cdot f(1,1) = 0,2 \cdot \mathrm{e}^{1,1} = \underline{0,600833}$.

$x_0 = 1,1 + 0,1 \cdot 0 = 1,1$
$w_0 = 0,1 \cdot 2 = 0,2$

$n = 1$

$\int_1^{1,2} f(x)\,dx \approx$
$\approx 0,1\left[f(1,1-0,1\frac{1}{\sqrt{3}}) + f(1,1+0,1\frac{1}{\sqrt{3}})\right] \doteq$
$\doteq 0,1[2,835632 + 3,182716] =$
$= \underline{0,601834}$.

$x_0 = 1,1 + 0,1 \cdot \left(-\sqrt{\frac{1}{3}}\right) =$
$= 1,1 - 0,1\sqrt{\frac{1}{3}}$,
$x_1 = 1,1 + 0,1\sqrt{\frac{1}{3}}$,
$w_0 = 0,1 \cdot 1 = 0,1$,
$w_1 = 0,1$.

Přesný výsledek: $\mathrm{e}^{1,2} - \mathrm{e} \doteq \underline{\underline{0,601835}}$.

Poznámka:

Podobně jako u Newton-Cotesových vzorců můžeme definovat **složené Gaussovy kvadraturní vzorce**

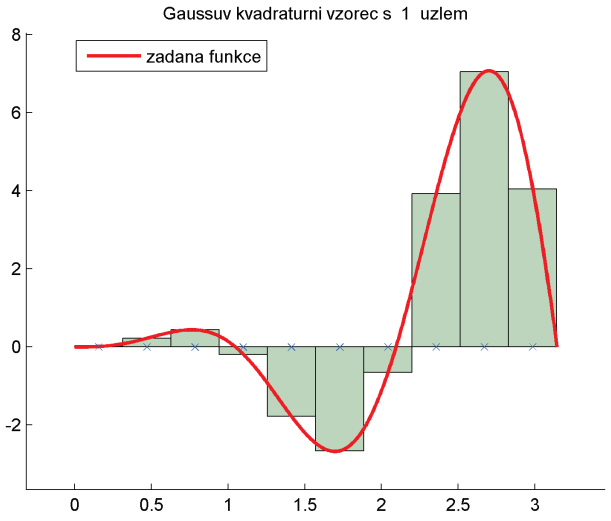
Příklad

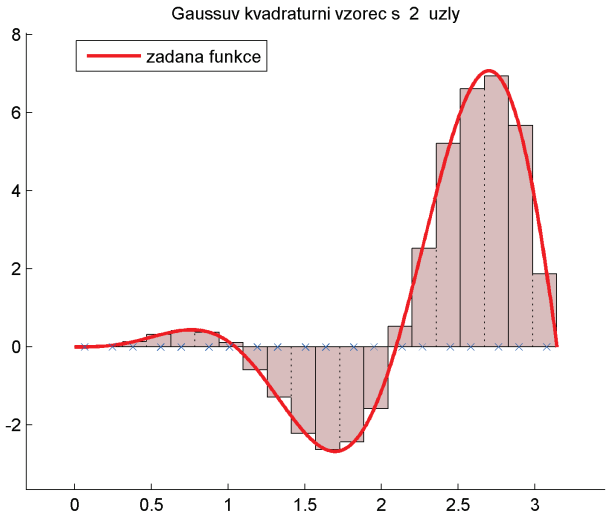
Vypočtěte $\int_0^{\pi} x^2 \sin 3x\,dx$ použitím jedno-, dvou- a tříbodového složeného Gaussova kvadraturního vzorce. Počet dělení intervalu $\langle 0,\pi\rangle$ volte $N = 10$, resp. $N = 20$.

Řešení:

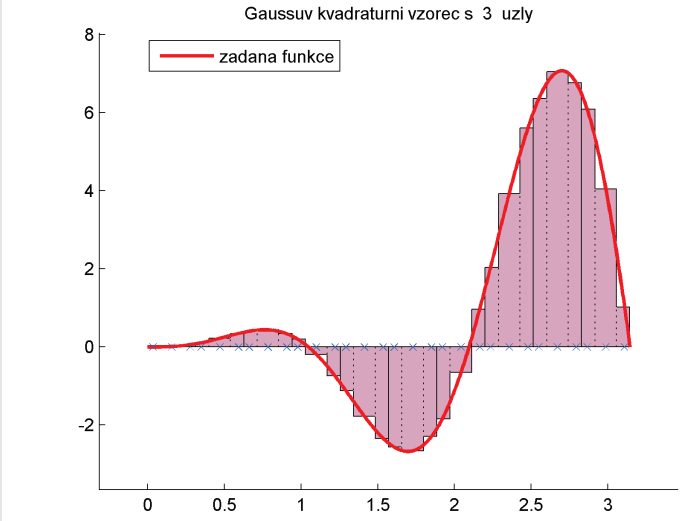
výsledky v MATLABu

```
-------------------------------------------------------------------
  Numericky vypocet urciteho integralu funkce f(x)=x^2*sin(3*x)
  na intervalu <0.000000,3.141593> s poctem deleni N=10
-------------------------------------------------------------------

 Presna hodnota integralu je  3.141720

 Priblizna hodnota integralu                                  chyba
    - pomoci Gausova vzorce s 1 uzlem       3.266250       0.124530
    - pomoci Gausova vzorce s 2 uzly        3.141191      -0.000529
    - pomoci Gausova vzorce s 3 uzly        3.141721       0.000001
```



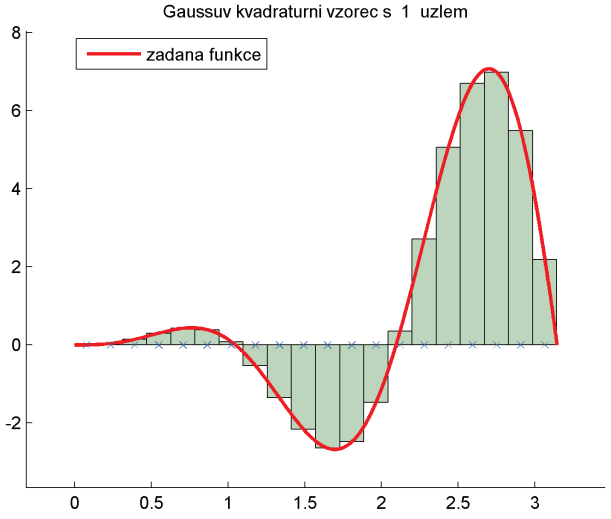


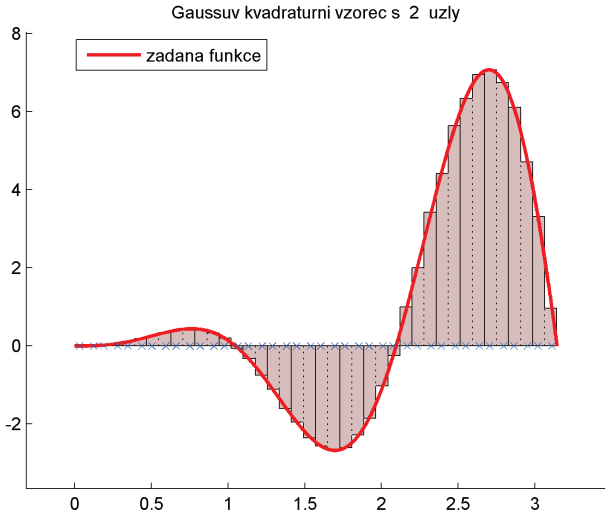
Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



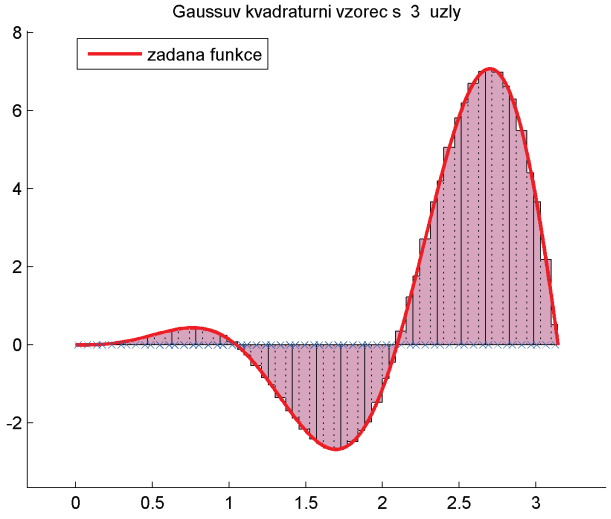
výsledky v MATLABu

```
-----------------------------------------------------------------
 Numericky vypocet urciteho integralu funkce f(x)=x^2*sin(3*x)
 na intervalu <0.000000,3.141593> s poctem deleni N=20
-----------------------------------------------------------------

Presna hodnota integralu je  3.141720

Priblizna hodnota integralu                              chyba
    - pomoci Gausova vzorce s 1 uzlem       3.172331     0.030612
    - pomoci Gausova vzorce s 2 uzly        3.141687    -0.000033
    - pomoci Gausova vzorce s 3 uzly        3.141720     0.000000
```





Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



Integrování periodické funkce přes periodu

Pro lichoběžníkové pravidlo platí:

$$T(f,h) = \int_a^b f(x)\,dx + \frac{h^2}{12}\left[f'(b) - f'(a)\right] - \frac{h^4}{720}\left[f^{(3)}(b) - f^{(3)}(a)\right] + \dots$$

(tzv. **Eulerův - Maclaurinův vzorec**)

Souvislost s rozvojem chyby:

$$T(f,h) = \int_a^b f(x)\,dx + \frac{h^2}{12}\underbrace{f''(x)}_{\frac{f'(b)-f'(a)}{b-a}}(b-a) - \frac{h^4}{720}\underbrace{f^{(4)}(x)}_{\frac{f'''(b)-f'''(a)}{b-a}}(b-a) + \dots$$

Pro kladnou periodickou funkci s periodou na intervalu $\langle a,b\rangle$ platí:

$$f'(a) = f'(b)$$
$$f^{(3)}(a) = f^{(3)}(b)$$
$$\vdots$$

Pozor: Obecně nemusí platit, že $T(f,h)$ je přesná hodnota integrálu $\int_a^b f(x)\,dx$, protože zbytek má tvar

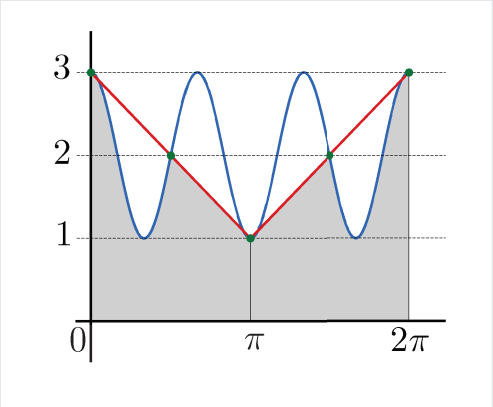
$$(b-a)\,c_{2m}\,h^{2m}\,f^{(2m)}(\xi)$$ a ξ neznáme.

Platí však, že chyba je velikosti $O(h^{2m})$ pro libovolné m takové, že f má spojitou $2m$-tou derivaci. Proto není nutné používat Rombergovu metodu.

Příklad: Vypočtěte složeným lichoběžníkovým pravidlem

$$\int_0^{2\pi} (2+\cos 3x)\,dx.$$

(Přesná hodnota je 4π.)



Zvolíme krok $h=\pi$, tj. $N=2$:

$$T(f,\pi)=\frac{\pi}{2}\big[f(0)+2f(\pi)+f(2\pi)\big]=\frac{\pi}{2}(3+2\cdot 1+3)=4\pi.$$

Platí: Složené lichoběžníkové pravidlo s $(k+2)$ uzly integruje přesně trigonometrické polynomy k-tého stupně a stupňů menších (tj. obsahující členy $\cos kx, \sin kx$) přes periodu 2π.

Nevlastní integrály

Při výpočtu integrálu

$$\boxed{\int_{-\infty}^{\infty} f(x)\,dx}$$

lze většinou předpokládat, že hodnoty funkce f a nižší derivace f jsou vně nějakého intervalu $\langle -R,R\rangle$ dostatečně malé. Proto je vhodné použít lichoběžníkové pravidlo pro integrál

$$\boxed{\int_{-R}^{R} f(x)\,dx}.$$

Příklad

Vypočtěte integrál $I=\int_{-\infty}^{\infty} \mathrm{e}^{-x^2}\,dx$.

Pro $|x|\geq 4$ je integrand menší než $0{,}5\cdot 10^{-6}$ a jeho první derivace menší než 10^{-6}. Použijeme-li lichoběžníkové pravidlo pro $\langle -4,4\rangle$, dostaneme:

$$T(f,1)=1{,}772636$$

$$T(f;0{,}5)=1{,}772453$$

Přesná hodnota

$$I=\sqrt{\pi}\doteq 1{,}7724538$$

Poznamenejme, že

$$\left|\int_{-\infty}^{\infty} \mathrm{e}^{-x^2}\,dx-\int_{-4}^{4} \mathrm{e}^{-x^2}\,dx\right|<10^{-7}.$$

Při výpočtu integrálu $\boxed{\int_0^{\infty} f(x)\,dx}$ můžeme použít transformaci $x=p(t)$.

a) $\underline{x=-\ln t}$, $dx=-\dfrac{dt}{t}$, $\begin{array}{c|c|c} x & 0 & \infty \\ \hline t & 1 & 0 \end{array}$

$$\int_0^{\infty} f(x)\,dx=-\int_1^0 f(-\ln t)\,\frac{dt}{t}=\int_0^1 \frac{f(-\ln t)}{t}\,dt$$

b) $\underline{x=\dfrac{t}{1-t}}$, $dx=(1-t)^{-2}\,dt$, $\begin{array}{c|c|c} x & 0 & \infty \\ \hline t & 0 & 1 \end{array}$

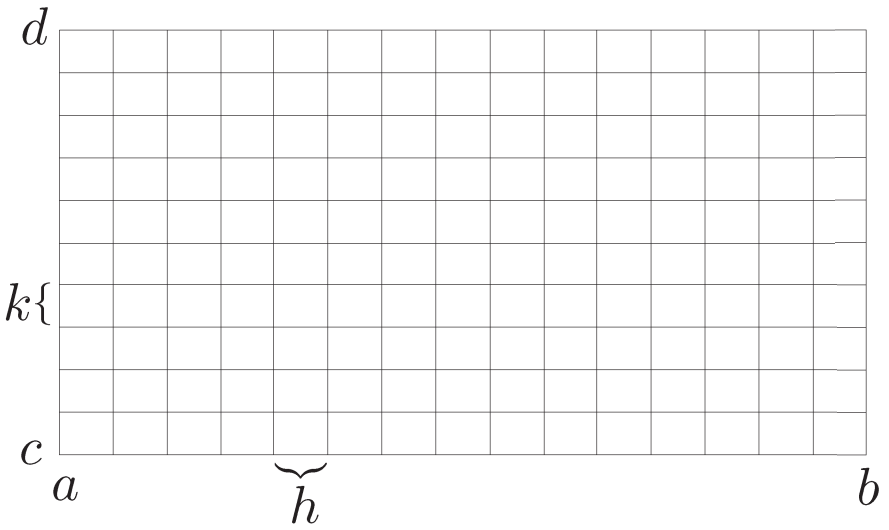
$$\int_0^{\infty} f(x)\,dx=\int_0^1 f\left(\frac{t}{1-t}\right)\frac{dt}{(1-t)^2}$$

Integrování funkce 2 proměnných

Odvoďte obdélníkové a lichoběžníkové pravidlo pro integrování funkce 2 proměnných na obdélníku $\langle a,b\rangle\times\langle c,d\rangle$, tj.

$$\int_a^b \Big(\int_c^d f(x,y)\,dy\Big)\,dx$$

Řešení:

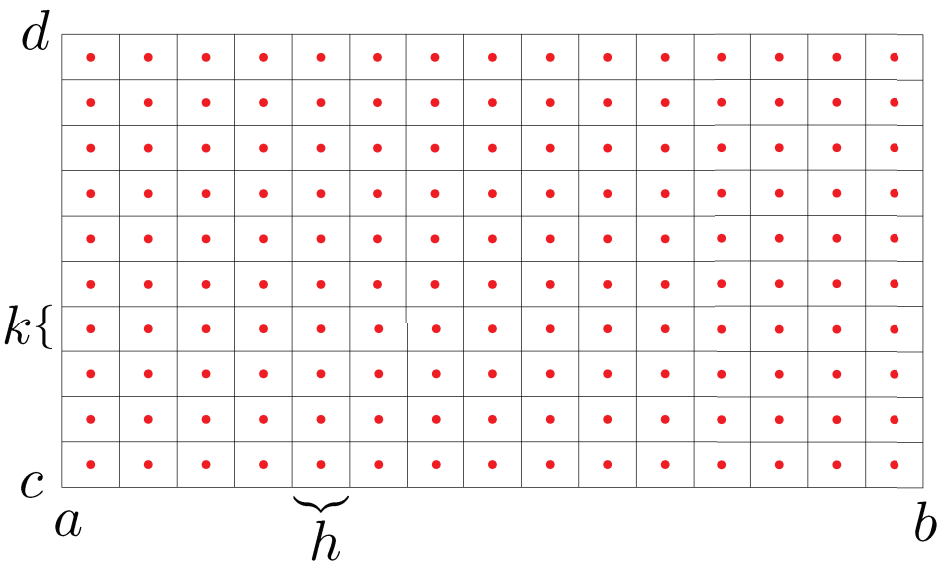


$$h = \frac{b-a}{N}, \quad k = \frac{d-c}{M}, \quad x_i = a + i \cdot h, \quad y_j = c + j \cdot k$$

$$\int_a^b \Big(\int_c^d f(x,y)\,dy \Big)\,dx = \dots$$

- obdélníkové pravidlo:

$$\dots = h \sum_{i=0}^{N-1} \left(\int_c^d f\left(x_i + \frac{h}{2}, y\right) dy \right) = h \sum_{i=0}^{N-1} \left(k \sum_{j=0}^{M-1} f\left(x_i + \frac{h}{2}, y_j + \frac{k}{2}\right) \right) = \underline{hk \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} f\left(x_i + \frac{h}{2}, y_j + \frac{k}{2}\right)}$$

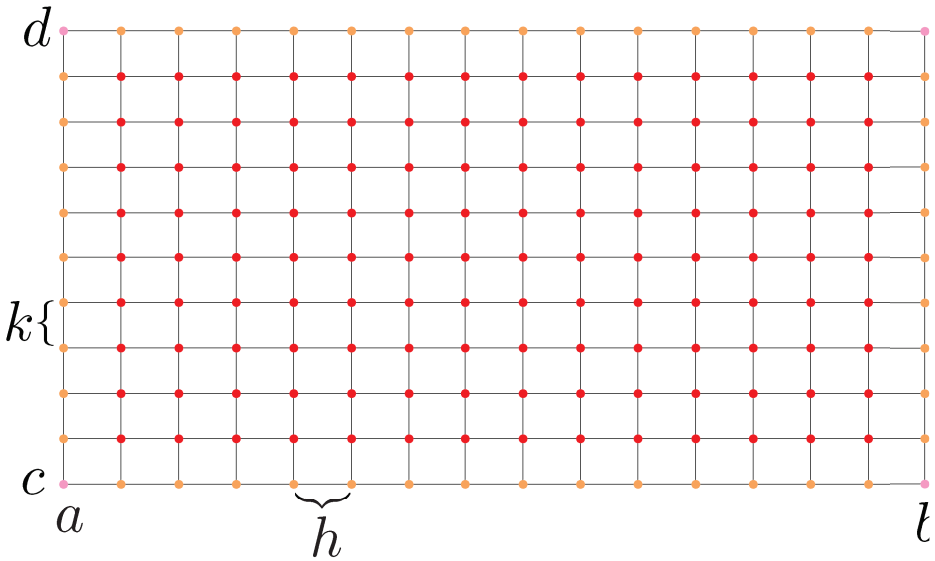


- lichoběžníkové pravidlo:

$$\dots = \frac{h}{2} \sum_{i=0}^{N-1} \left(\int_c^d \Big[f(x_i, y) + f(x_{i+1}, y) \Big] dy \right) =$$

$$= \frac{h}{2} \sum_{i=0}^{N-1} \left(\frac{k}{2} \sum_{j=0}^{M-1} \Big[f(x_i, y_j) + f(x_i, y_{j+1}) \Big] + \frac{k}{2} \sum_{j=0}^{M-1} \Big[f(x_{i+1}, y_j) + f(x_{i+1}, y_{j+1}) \Big] \right) =$$

$$= \underline{\frac{h}{2} \frac{k}{2} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \Big[f(x_i, y_j) + f(x_i, y_{j+1}) + f(x_{i+1}, y_j) + f(x_{i+1}, y_{j+1}) \Big]}$$



Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Kapitola 11. Počáteční úlohy pro ODR - I

Počáteční úlohy pro obyčejné diferenciální rovnice 1. řádu

Formulace:

Je dána funkce dvou proměnných $f = f(x, y)$, $y \in \mathbb{R}$, $x \in \langle a, b\rangle$ a čísla $x_0 \in \langle a, b\rangle$ a $y_0 \in \mathbb{R}$.
Chceme najít takovou funkci $y = y(x)$, $x \in \langle x_0, b\rangle$, která na intervalu (x_0, b) vyhovuje rovnici

$$\boxed{y' = f(x, y)}$$

a splňuje počáteční podmínku

$$\boxed{y(x_0) = y_0}.$$

Funkci $y = y(x)$, která splňuje počáteční podmínku a rovnost $y' = f(x, y(x))$ na příslušném intervalu, nazýváme řešením úlohy.

V některých případech budeme uvažovat speciální úlohu s rovnicí

$$\boxed{y' = a(x)y + b(x)}$$

nebo s rovnicí

$$\boxed{y' = \lambda y}.$$

Velmi podstatnou úlohu v našich dalších úvahách bude hrát předpoklad, že funkce f je na nějakém intervalu **lipschitzovsky spojitá** (v druhé proměnné), tj. platí:

$$\boxed{|f(x, y_1) - f(x, y_2)| \le L\,|y_1 - y_2| \quad \forall x \in \langle a, b\rangle \ \ \forall y_1, y_2 \in \mathbb{R}}$$

Příklad 1

Uvažujme úlohu

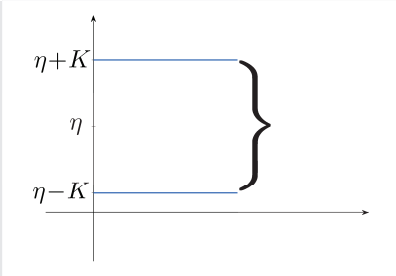
$$\boxed{\begin{array}{c} y' = y^2 \\ y(0) = \eta > 0 \end{array}}$$

Funkce $f(x, y) = y^2$ je lipschitzovsky spojitá na libovolném konečném intervalu $\langle \eta - K, \eta + K\rangle$.
Konstanta $L = 2(\eta + K)$.

$$\left|y_1^2 - y_2^2\right| \le 2(\eta + K)\,|y_1 - y_2|$$

$$|y_1 + y_2| \le 2(\eta + K) \quad \forall x \in \langle a, b\rangle, \quad \forall y_1, y_2 \in \langle \eta - K, \eta + K\rangle$$

$$|y_1 + y_2| \le |y_1| + |y_2| \le \eta + K + \eta + K = 2(\eta + K) \quad \forall x \in \langle a, b\rangle, \quad \forall y_1, y_2 \in \langle \eta - K, \eta + K\rangle$$



Řešíme pomocí separace proměnných

$$\frac{dy}{dx} = y^2$$
$$\frac{dy}{y^2} = dx$$
$$-\frac{1}{y} = x + c$$
$$y = -\frac{1}{x+c}$$
$$y(0) = \eta$$
$$-\frac{1}{c} = \eta$$
$$c = -\frac{1}{\eta}$$

Řešení této úlohy má tvar

$$\boxed{\bar{y}(x) = \frac{1}{\frac{1}{\eta} - x}}$$

Když $x \to \left(\frac{1}{\eta}\right)_-$, pak $\bar{y} \to \infty$.

$\Rightarrow$ Pro všechna $y_1, y_2 \in \mathbb{R}$ nelze najít jednu konstantu L a
řešení neexistuje pro libovolné x (řešení existuje pouze do určitého času).

Příklad 2

Uvažujme úlohu

$$\boxed{\begin{aligned} y' &= \sqrt{y} \\ y(0) &= 0 \end{aligned}}$$

Funkce $f(x,y) = \sqrt{y}$ není lipschitzovsky spojitá v okolí 0, protože $\frac{\partial f(x,y)}{\partial y} = \frac{1}{2\sqrt{y}} \to \infty$ pro $y \to 0_+$.

Z věty o střední hodnotě plyne:

$$\boxed{f(x,y_1) - f(x,y_2) = \frac{\partial f(x,\xi)}{\partial y}(y_1 - y_2)}$$

Řešení této úlohy není jednoznačné:

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

$$\boxed{\begin{aligned} \bar{y}_1(x) &= 0 \\ \bar{y}_2(x) &= \frac{1}{4}x^2 \end{aligned}}$$

$$\frac{dy}{dx} = \sqrt{y}$$

1) $y \equiv 0$ OK 2) $y \neq 0$

$$\frac{dy}{\sqrt{y}} = dx$$
$$2\sqrt{y} = x + c$$
$$\sqrt{y} = \frac{x}{2} + c$$
$$y = \left(\frac{x}{2} + c\right)^2$$
$$y(0) = 0$$
$$c^2 = 0$$
$$c = 0$$
$$y = \frac{x^2}{4}$$

Příklad 3

Uvažujme úlohu

$$\boxed{\begin{aligned} y' &= \lambda y \\ y(0) &= 1 \end{aligned}}$$

Funkce $f(x,y) = \lambda y$ je lipschitzovsky spojitá pro všechna $y \in \mathbb{R}$ s konstantou $L = \lambda$.

$$|\lambda y_1 - \lambda y_2| = |\lambda|\,|y_1 - y_2| \leq |\lambda|\,|y_1 - y_2|$$

Tato úloha má právě jedno řešení pro všechna $x \in \langle 0, \infty)$ ve tvaru

$$\boxed{\bar{y}(x) = \mathrm{e}^{\lambda x}}$$

$$\frac{dy}{dx} = \lambda y$$
$$\frac{dy}{y} = \lambda dx$$
$$\ln|y| = \lambda x + c$$
$$\ln|y| = \ln \mathrm{e}^{\lambda x} + \ln c$$
$$y = c\,\mathrm{e}^{\lambda x}$$
$$y(0) = 1$$
$$c = 1$$
$$y = \mathrm{e}^{\lambda x}$$

Následující příklady ukazují význam Lipschitzovy konstanty.

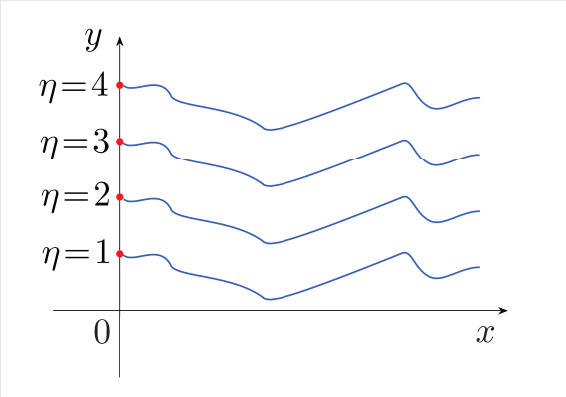
Příklad 4

Uvažujme úlohu ve tvaru

$$\boxed{\begin{aligned} y' &= g(x) \\ y(0) &= \eta \end{aligned}}$$

Řešení má tvar $\bar{y}(x) = \eta + \int_0^x g(\xi)\,d\xi$ a Lipschnitzova konstanta $L = 0$.

Křivky řešení mohou vypadat třeba takto:



Změníme-li počáteční podmínku η, potom nové řešení je pouhé posunutí původního do hodnoty η.

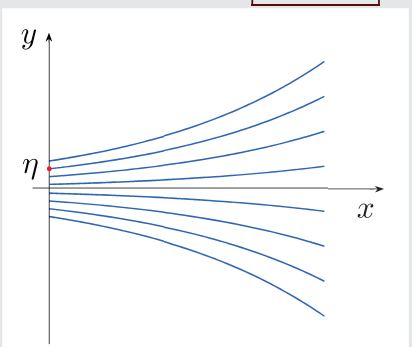
Příklad 5

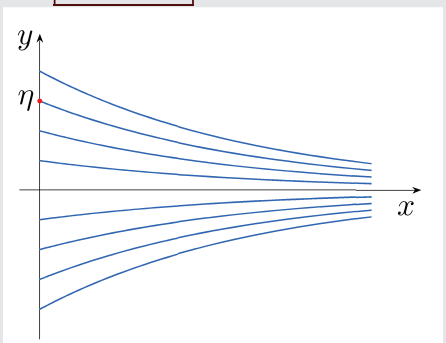
Uvažujme úlohy

$$\boxed{\begin{aligned} y' &= 3y \\ y(0) &= \eta \end{aligned}} \quad \text{a} \quad \boxed{\begin{aligned} y' &= -3y \\ y(0) &= \eta \end{aligned}}$$

Dostaneme řešení

$$\boxed{\bar{y}(x) = \eta e^{3x}} \quad \text{a} \quad \boxed{\bar{y}(x) = \eta e^{-3x}}$$





V obou případech je Lipschitzova konstanta $L = 3$. Její velikost však může ovlivňovat chování konkrétní numerické metody pro konkrétní úlohu.

Věta Nechť funkce $f(x,y)$ má následující vlastnosti:
(i) je definována v pásu $S = \langle a,b\rangle \times \mathbb{R}$ $(a, b$ ... konečné),

(ii) je spojitá v proměnné $x \in \langle a,b\rangle$ pro každé $y \in \mathbb{R}$,

(iii) splňuje Lipschitzovu podmínku v proměnné y, tj. existuje číslo L takové, že platí nerovnost

$$|f(x,y_1) - f(x,y_2)| \le L\,|y_1 - y_2| \quad \forall x \in \langle a,b\rangle \quad \forall y_1, y_2 \in \mathbb{R}$$

Potom pro každé $x_0 \in \langle a,b\rangle$ a libovolné $y_0 \in \mathbb{R}$ existuje právě jedna funkce $y = y(x)$ s vlastnostmi:

a) $y(x)$ je spojitá a spojitě diferencovatelná pro $x \in \langle a,b\rangle$,

b) platí rovnost $y'(x) = f(x, y(x))$ pro $\forall x \in \langle a,b\rangle$,

c) $y(x_0) = y_0$.

Numerické metody lze dělit podle různých kritérií:

A) metody založené na numerické derivaci **X** na numerické integraci

B) jednokrokové metody **X** vícekrokové metody

C) explicitní metody **X** implicitní metody

D) metody s konstantním krokem **X** metody s proměnným krokem

Princip:
Základem metod je diskretizace proměnných.
Přibližné řešení nehledáme jako spojitou funkci, ale nagenerujeme body $x_0, x_1, x_2, \ldots$
a určujeme čísla $y_0, y_1, y_2, \ldots$, která aproximují $y(x_0), y(x_1), y(x_2), \ldots$.
Pro jednoduchost můžeme uvažovat ekvidistantní dělení, tj. $h = x_{k+1} - x_k$, $\forall k$.

Eulerova metoda

- nejjednodušší jednokroková explicitní metoda; lze odvodit řadou postupů
- 1.odvození

  y_0 ... dáno

  y_1 ... počítáme extrapolací z hodnoty y_0, přičemž se na intervalu $\langle x_0, x_1\rangle$ řešení aproximuje přímkou,

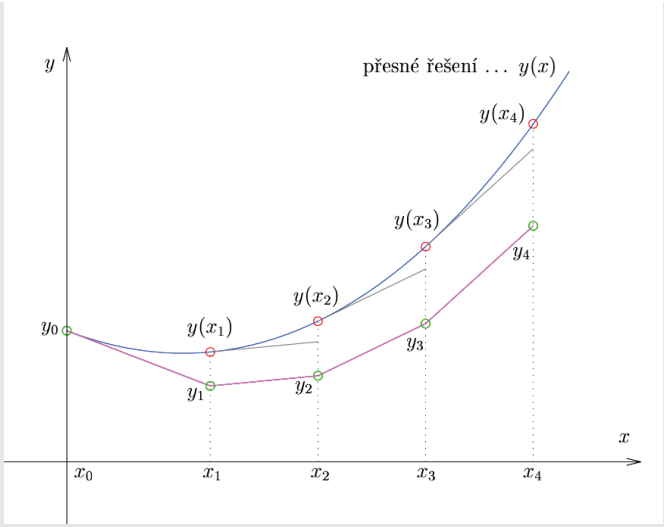
  která prochází bodem $[x_0, y_0]$ a má směrnici $y' = f(x_0, y_0)$.

  Ta má rovnici $y - y_0 = (x - x_0) f(x_0, y_0)$. Tj. pro x_1 dostáváme:

$$y_1 = y_0 + \underbrace{(x_1 - x_0)}_{h} f(x_0, y_0)$$

  Obecně dostaneme rekurentní vztah

$$\boxed{y_{k+1} = y_k + h_k f(x_k, y_k), \quad k = 0, 1, 2, \ldots}$$



- 2.odvození Pomocí Taylorova rozvoje.

$$y(x_{k+1}) = y(x_k) + h_k \underbrace{y'(x_k)}_{=f(x_k,y(x_k))} + \underbrace{\frac{1}{2}h_k^2 y''(\xi_k)}_{(*)}, \quad \xi_k \in (x_k, x_{k+1})$$

$(*)$ zanedbáme a dostaneme vztah pro přibližné řešení

$$\boxed{y_{k+1} = y_k + h_k f(x_k, y_k), \quad k = 0, 1, 2, \dots}$$

y_0 ... počáteční podmínka

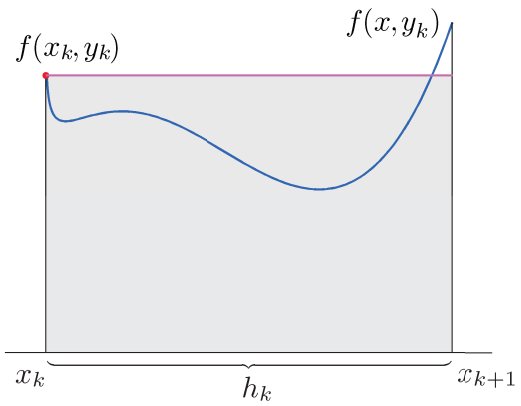
- 3.odvození Původní diferenciální rovnici nahradíme diferenční rovnicí (aproximujeme derivaci).

$$y' = f(x,y) \quad \rightarrow \quad \boxed{\frac{y_{k+1} - y_k}{h_k} = f(x_k, y_k), \quad k = 0, 1, 2, \dots}$$

- 4.odvození Původní diferenciální rovnici zintegrujeme a aproximujeme určitý integrál.

$$y' = f(x,y) \quad \rightarrow \quad y(x_{k+1}) - y(x_k) = \int_{x_k}^{x_{k+1}} f(x, \underbrace{y(x)}_{(*)})\, dx$$

$(*)$ $y(x)$ na $\langle x_k, x_{k+1}\rangle$ aproximujeme konstantou y_k



$$\boxed{y_{k+1} - y_k = h_k f(x_k, y_k), \quad k = 0, 1, 2, \dots}.$$

Poznámka:

Eulerova metoda je

- jednokroková metoda $(y_k \rightarrow y_{k+1})$
  K výpočtu y_{k+1} použijeme pouze předchozí hodnotu y_k.
- explicitní metoda (na pravé straně není y_{k+1})
  V získané formuli je explicitně vyjádřena hodnota y_{k+1}.

Příklad

Pomocí Eulerovy metody řešte následující úlohu na intervalu $\langle 0; 0{,}6\rangle$ s konstantními kroky $h = 0{,}2$ a $h = 0{,}1$.

$$\boxed{\begin{aligned} y' &= x - y \\ y(0) &= 1 \end{aligned}}$$

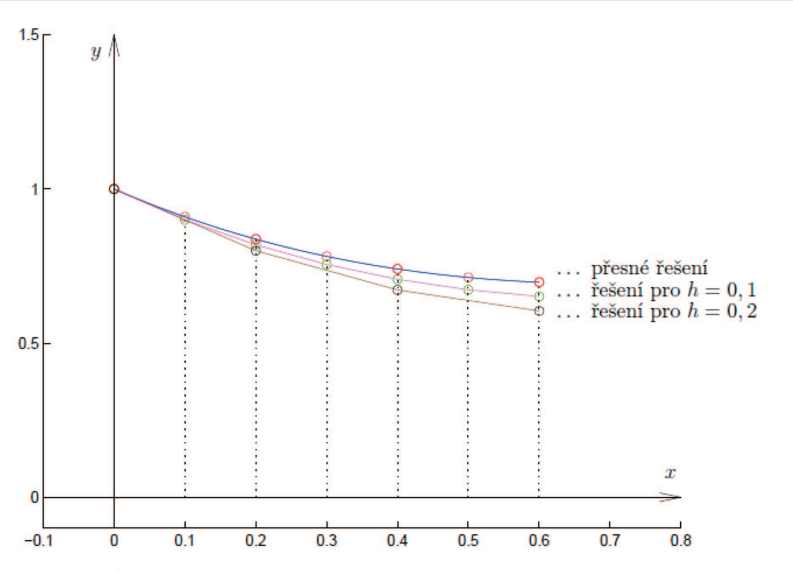
Řešení:

(Přesné řešení: $y(x) = 2e^{-x} + x - 1$).

Eulerova metoda je dána rekurentním vztahem:

$$y_{k+1} = y_k + h \cdot f(x_k, y_k).$$

		$h = 0,2$		$h = 0,1$	
x_k	$y(x_k)$	y_k	e_k	y_k	e_k
0	1,000	1,000	0,000	1,000	0,000
0,1	0,910			0,900	0,010
0,2	0,837	0,800	0,037	0,820	0,017
0,3	0,782			0,758	0,024
0,4	0,741	0,680	0,061	0,712	0,029
0,5	0,713			0,681	0,032
0,6	0,698	0,624	0,074	0,663	0,035



Poznámky:

1) Vidíme, že je chyba úměrná h.
2) Chyba s rostoucím x vzrůstá.

Definice: **Lokální diskretizační chyba** d_k na intervalu $\langle x_k, x_{k+1}\rangle$ je nepřesnost, s níž hodnoty teoretického řešení dané úlohy splňují rekurentní vztah, ze kterého se počítá hodnota y_{k+1}.

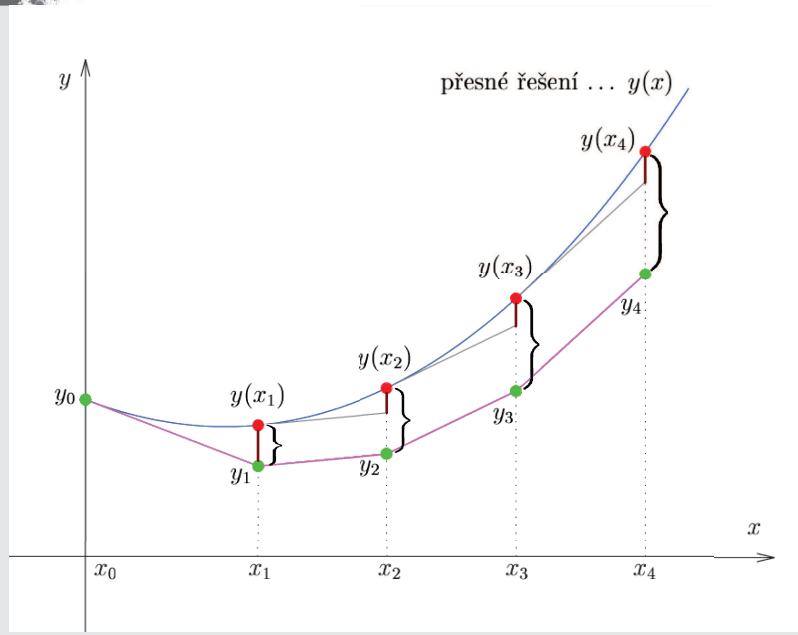
Pro Eulerovu metodu je lokální diskretizační chyba d_k:

$$\boxed{y(x_{k+1}) = y(x_k) + h_k\, f(x_k, y(x_k)) + d_k}$$

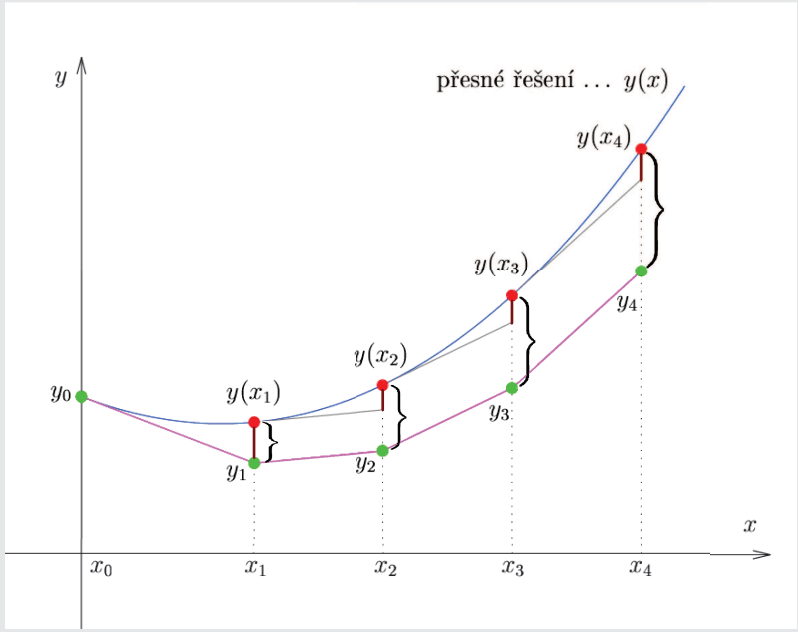
Poznámka:

Lokální diskretizační chyba se nazývá lokální proto, že d_k lze interpretovat také jako chybu jednoho kroku metody (při výpočtu y_{k+1}) za předpokladu, že všechny hodnoty $y_k, y_{k-1}, \ldots$ potřebné k výpočtu y_{k+1} jsou přesné.

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



Definice: **Globální diskretizační chyba** je $e_k = y(x_k) - y_k$, tj. rozdíl teoretické hodnoty řešení a vypočtené hodnoty řešení v daném bodě x_k.



Globální diskretizační chyba Eulerovy metody (pro konstantní krok h)

Přibližné řešení:

$$y_0 = y(x_0)$$
$$y_{k+1} = y_k + h\,f(x_k, y_k) \quad k = 0, 1, \ldots$$

Přesné řešení:

$$y(x_{k+1}) = y(x_k) + h\,f(x_k, y(x_k)) + d_k \quad k = 0, 1, \ldots$$

Po odečtení:

$$\boxed{\begin{aligned} e_0 &= 0 \\ e_{k+1} &= e_k + h\left(f(x_k, y(x_k)) - f(x_k, y_k)\right) + d_k \end{aligned}},$$

tj. v každém kroku se ke globální chybě e_k připočítá lokální chyba d_k a člen $h\cdot(\ldots)$, který představuje nepřesnosti z minulých kroků.

Příklad:

Speciální případ, kdy f nezávisí na y:

$$\boxed{\begin{aligned} y' &= f(x) \\ y(x_0) &= y_0 \end{aligned}} \quad \Rightarrow \quad \boxed{e_{k+1} = \sum_{m=0}^{k} d_m}, \quad \circledast$$

tj. globální chyba je součtem lokálních chyb.

Poznámka:

Lokální chyba Eulerovy metody je $O(h^2)$ (viz další slide).

Protože $\circledast$ má k sčítanců a protože pro pevné x je $k = \dfrac{x-a}{h}$, plyne z $\circledast$

$$\boxed{e(x,h) = \frac{const}{h} \cdot O(h^2) = O(h)}$$

... podobně jako u základních a složených kvadraturních vzorců.

Poznámka:

Lokální i globální diskretizační chyba jsou chyby aproximace, tj. neuvažovali jsme zaokrouhlovací chyby.

Definice: **Řád diferenční metody** je největší přirozené číslo p takové, že pro danou metodu aplikovanou na libovolou počáteční úlohu s dostatečně hladkým řešením platí při každém pevném k a $h_k \to 0$ odhad

$$d_k = O(h_k^{p+1}).$$

Řád Eulerovy metody

Ze vztahu pro lokální diskretizační chybu d_k plyne:

$$d_k = y(x_{k+1}) - y(x_k) - h_k \cdot \underbrace{y'(x_k)}_{=f(x_k, y(x_k))}$$

$y(x_{k+1})$ vyjádříme pomocí Taylorova rozvoje (předpokládáme, že y má 2. derivaci)

$$y(x_{k+1}) = y(x_k) + h_k y'(x_k) + \frac{1}{2} h_k^2 y''(\xi) \quad \xi \in (x_k, x_{k+1})$$

Po dosazení:

$$\boxed{d_k = \frac{1}{2} h_k^2 y''(\xi)} = O(h_k^2)$$

$2 = p + 1 \quad \Rightarrow$ **řád Eulerovy metody** je $p = 1$.

Obecná jednokroková metoda

Eulerova metoda je sice velmi jednoduchá (řád je 1), ale k dosažení určité přesnosti musíme používat velmi malé kroky h_k. Chceme-li jednokrokovou metodu vyššího řádu, musíme se zříci linearity

$$y_{k+1} = y_k + h_n\, f(x_k, y_k) \qquad k = 0, 1, 2, \dots$$

$$\boxed{y_{k+1} = y_k + \Phi(x_k, y_k, h_k, f) \qquad k = 0, 1, 2, \dots}$$

Metody Taylorova typu

Hodnotu $y(x_{k+1})$ budeme aproximovat pomocí Taylorova rozvoje vyššího řádu p (v Eulerově metodě byl použit řád 1), tj.

$$y(x_{k+1}) = y(x_k + h_k) = y(x_k) + h_k\, y'(x_k) + \frac{h_k^2}{2!}\, y''(x_k) + \dots + \frac{h_k^p}{p!}\, y^{(p)}(x_k) + \frac{h_k^{p+1}}{(p+1)!}\, y^{(p+1)}(\xi_k) \quad \xi_k \in (x_k, x_{k+1})$$

Je třeba dosadit za derivace y v bodě x_k. Derivace určíme postupným derivováním funkce f.

$$y' = f(x, y(x))$$

$$y'' = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \cdot \frac{dy}{dx} = f_x + f_y \cdot \underbrace{f}_{=y'} \overset{\text{ozn.}}{=} f^{[1]}(x, y)$$

$$y''' = \frac{\partial f^{[1]}}{\partial x} + \frac{\partial f^{[1]}}{\partial y} \cdot \frac{dy}{dx} = f_x^{[1]} + f_y^{[1]} \cdot \underbrace{f}_{=y'} \overset{\text{ozn.}}{=} f^{[2]}(x, y)$$

$$\vdots$$

Obecně lze odvodit rekurenci

$$y^{(r+1)} = f^{[r]}(x, y(x)) = f_x^{[r-1]}(x, y(x)) + f_y^{[r-1]}(x, y(x)) \cdot f(x, y(x)) \quad r = 1, 2, \dots$$

Po dosazení (uvažujme konstantní krok h) dostáváme

$$\boxed{y_{k+1} = y_k + h\, f(x_k, y_k) + \frac{h^2}{2}\, f^{[1]}(x_k, y_k) + \dots + \frac{h^p}{p!}\, f^{[p-1]}(x_k, y_k)}$$

Poznámka:

Metody Taylorova typu se v praxi nepoužívají právě z důvodu nutnosti vyjadřovat derivace y'', y''', ...

Příklad Odvoďte metodu Taylorova typu 2.řádu pro řešení následující úlohy na intervalu $\langle 0; 0,6\rangle$ s konstantním krokem $h = 0,2$

$$\boxed{\begin{aligned} y' &= x - y, \\ y(0) &= 1 \end{aligned}}.$$

Řešení:

(Přesné řešení: $y(x) = 2e^{-x} + x - 1$).

$$f(x, y) = x - y$$

$$f^{[1]}(x, y) = f_x + f_y \cdot f = 1 + (-1) \cdot f(x, y) = 1 - x + y.$$

Dostáváme rekurentní vztah:

$$y_{k+1} = y_k + h\,(x_k - y_k) + \frac{1}{2}\, h^2\,(1 - x_k + y_k)$$

x_k	$y(x_k)$	y_k	$h(x_k - y_k)$	$\frac{h^2}{2}(1 - x_k + y_k)$	e_k
0	1,000	1,000	-0,200	0,040	0,000
0,2	0,837	0,840	-0,128	0,033	-0,003
0,4	0,741	0,745	-0,069	0,027	-0,004
0,6	0,698	0,703			-0,005

Poznámka:

Vidíme, že metoda Taylorova typu 2. řádu pro $h = 0,2$ dává přesnější výsledky než Eulerova metoda s $h = 0,1$.

Metody Runge-Kuttova typu

- Univerzálnější metody než metody Taylorova typu.
- Vychází také z Taylorova polynomu, ale nepoužívá se ho přímo, aby nebylo nutné explicitně vyjadřovat derivace funkce $f = f(x, y(x))$ a počítat jejich hodnoty. Hledaná aproximace je kombinací několika hodnot funkce f vypočítaných v několika strategicky volených bodech (x, y) na intervalu $\langle x_k, x_{k+1}\rangle$.

Poznámka: Těchto metod je velké množství!

Heunova metoda (Runge-Kuttova metoda 2. řádu)

– vztah $y' = f(x, y(x))$ zintegrujeme přes interval $\langle x_k, x_{k+1}\rangle$

$$\int_{x_k}^{x_{k+1}} y'(x)\, dx = \int_{x_k}^{x_{k+1}} f(x, y(x))\, dx$$

$$y(x_{k+1}) - y(x_k) = \int_{x_k}^{x_{k+1}} f(x, y(x))\, dx$$

– použijeme lichoběžníkové pravidlo

$$y(x_{k+1}) - y(x_k) = \frac{h}{2}\left[f(x_k, y(x_k)) + f(x_{k+1}, \underline{y(x_{k+1})})\right] + \underbrace{O(h^3)}_{\substack{\text{viz chyba} \\ \text{lich. pr.}}}$$

– na pravé straně vystupuje hodnota $y(x_{k+1})$, její aproximaci určíme pomocí Eulerovy metody

$$\overline{y}(x_{k+1}) = y(x_k) + h\, f(x_k, y(x_k)) + O(h^2)$$

– dostáváme metodu ve tvaru

$$\boxed{\begin{aligned}\overline{y}_{k+1} &= y_k + h\, f(x_k, y_k)\\ y_{k+1} &= y_k + \frac{h}{2}\left[f(x_k, y_k) + f(x_{k+1}, \overline{y}_{k+1})\right]\end{aligned}}$$

Poznámka: Lokální diskretizační chyba, tj. chyba jednoho kroku metody, je $d_k = O(h^3)$. Globální chyba je potom o řád nižší, tj. $e_k = O(h^2)$, protože chyba metody se zvětšuje lineárně s počtem kroků $k \sim \frac{1}{h}$.

Modifikovaná Eulerova metoda (Runge-Kuttova metoda 2. řádu)

- vztah $y' = f(x, y(x))$ opět zintegrujeme přes interval $\langle x_k, x_{k+1}\rangle$

$$\int_{x_k}^{x_{k+1}} y'(x)\,dx = \int_{x_k}^{x_{k+1}} f(x, y(x))\,dx$$

$$y(x_{k+1}) - y(x_k) = \int_{x_k}^{x_{k+1}} f(x, y(x))\,dx$$

- použijeme obdélníkové pravidlo

$$y(x_{k+1}) - y(x_k) = h \cdot f\big(x_k + \frac{h}{2}, y(x_k + \frac{h}{2})\big) + \underbrace{O(h^3)}_{\text{viz chyba obd. pr.}}$$

- hodnotu $y(x_k + \frac{h}{2})$ určíme pomocí Eulerovy metody

$$y(x_k + \frac{h}{2}) = y(x_k) + \frac{h}{2} f(x_k, y(x_k)) + O(h^2)$$

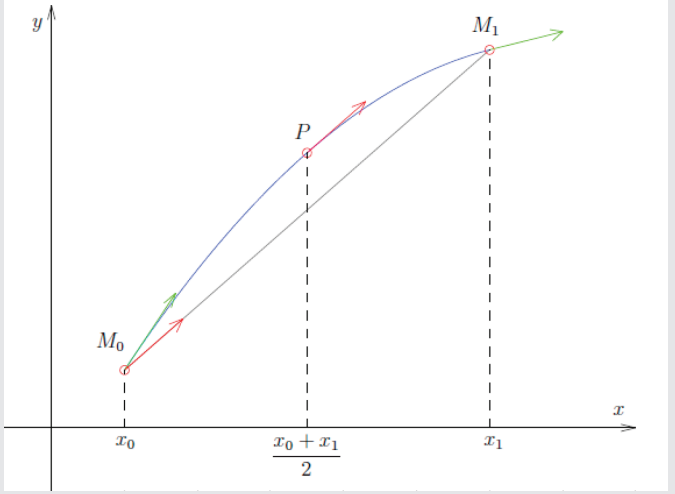
- dostáváme metodu ve tvaru

$$\boxed{\begin{aligned}y_{k+\frac{1}{2}} &= y_k + \frac{h}{2} f(x_k, y_k)\\ y_{k+1} &= y_k + h\, f(x_k + \frac{h}{2}, y_{k+\frac{1}{2}})\end{aligned}}$$

Poznámka: Lokální diskretizační chyba je opět $d_k = O(h^3)$. Globální chyba je potom o řád nižší, tj. $e_k = O(h^2)$, protože chyba metody se zvětšuje lineárně s počtem kroků $k \sim \frac{1}{h}$.

Ukažme si jiné odvození předchozích dvou Runge-Kuttových metod 2. řádu.
Odvození vychází z geometrické interpretace.

Numerické metody KMA/NM
Josef Daněk 13.2.2o13



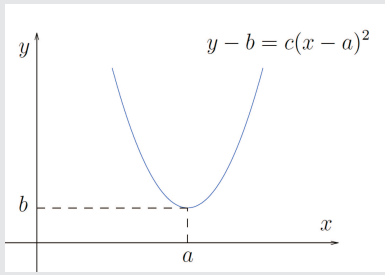
Věta

Nechť oblouk $M_0 M_1$ je částí paraboly. Potom platí:
1. Tečna v bodě P je rovnoběžná s tětivou $M_0 M_1$.
2. Směrnice tětivy $M_0 M_1$ je aritmetickým průměrem směrnic tečen v M_1 a M_2.

Důkaz:

Rovnice paraboly (polynomu 2.stupně): $y - b = c(x-a)^2$

$$y = c(x-a)^2 + b$$
$$\Rightarrow$$
$$y' = 2c(x-a)$$



1. Směrnice tečny v bodě P:

$$y'\big(\frac{x_0 + x_1}{2}\big) = 2c\big(\frac{x_0 + x_1}{2} - a\big) = \underline{\underline{c(x_0 + x_1 - 2a)}}$$

Směrnice tětivy $M_0 M_1$ je:

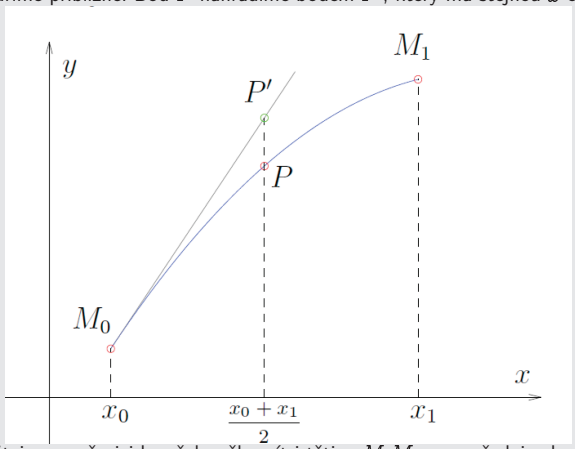
$$\frac{y(x_1)-y(x_0)}{x_1-x_0}=\frac{c(x_1-a)^2+b-c(x_0-a)^2-b}{x_1-x_0}=$$
$$=\frac{cx_1^2-2acx_1+a^2c+b-cx_0^2+2acx_0-a^2c-b}{x_1-x_0}=$$
$$=c\left(\frac{x_1^2-x_0^2-2a(x_1-x_0)}{x_1-x_0}\right)=\underline{\underline{c(x_1+x_0-2a)}}.$$

2. Směrnice tečny v bodě M_0 je:
$$y'(x_0)=2c(x_0-a)$$
Směrnice tečny v bodě M_1 je:
$$y'(x_1)=2c(x_1-a)$$
Jejich aritmetický průměr:
$$\frac{y'(x_0)+y'(x_1)}{2}=\frac{2c(x_0-a)+2c(x_1-a)}{2}=$$
$$=c(x_0-a+x_1-a)=\underline{\underline{c(x_0+x_1-2a)}}.$$

□

Nyní použijeme vlastnost 1.

Známe souřadnice bodu M_0. Jestliže bychom znali y-souřadnici bodu P, pak stačí udělat tečnu a bodem M_0 vést rovnoběžku a dostaneme y-souřadnici bodu M_1. My ale y-souřadnici bodu P neznáme, takže ji vyjádříme přibližně. Bod P nahradíme bodem P', který má stejnou x-ovou souřadnici a leží na tečně k M_0.



P' má souřadnice:
$$\left[x_0+\frac{h_0}{2},y_0+\frac{h_0}{2}\cdot\underbrace{f(x_0,y_0)}_{y_0'(x_0)}\right]$$

Tečna v bodě P' má směrnici:

$y'(x_0+\frac{h_0}{2})$, tj.

$y'(x_0+\frac{h_0}{2})=$

$f(x_0+\frac{h_0}{2},y_0+\frac{h_0}{2}\cdot\overbrace{f(x_0,y_0)}^{k_1})$.

Stejnou směrnici by však měla mít i tětiva $M_0M_1 \Rightarrow$ souřadnice bodu M_1 jsou:

$$x_1=x_0+h_0$$
$$y_1=y_0+h_0\cdot\overbrace{y'(x_0+\frac{h_0}{2})}^{k_2}$$

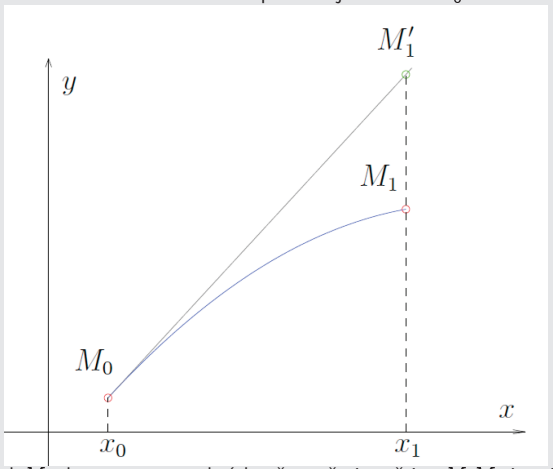
Tyto vztahy lze přepsat do tvaru (obecně)

$$\boxed{\begin{aligned}k_1&=f(x_k,y_k)\\k_2&=f(x_k+\frac{h_k}{2},y_k+\frac{h_k}{2}\cdot k_1)\\y_{k+1}&=y_k+h_k\cdot k_2\end{aligned}}$$

modifikovaná Eulerova metoda

Nyní použijeme vlastnost 2.

Známe souřadnice bodu M_0. Protože neznáme y-souřadnici bodu M_1, nahradíme ho bodem M_1', který má stejnou x-souřadnici a leží na tečně procházející bodem M_0.



M_1' má souřadnice:
$$\Big[\underbrace{x_0+h_0}_{=x_1},y_0+h_0\cdot\overbrace{\underbrace{f(x_0,y_0)}_{=y'(x_0)}}^{\text{ozn.}k_1}\Big]$$

Směrnice tečny v M_1' je:
$$\overbrace{f(x_0+h_0,y_0+h_0\cdot f(x_0,y_0))}^{\text{ozn.}k_2}$$

Bod M_1 dostaneme z podmínky, že směrnice tětivy M_0M_1 je aritmetickým průměrem směrnic tečen v M_0 a M_1', tj.

M_1' má souřadnice:
$$x_1=x_0+h_0$$
$$y_1=y_0+h_0\cdot\frac{1}{2}(k_1+k_2)$$

Obecně:

$$\boxed{\begin{aligned}k_1&=f(x_k,y_k)\\k_2&=f(x_k+h_k,y_k+h_k\cdot k_1)\\y_{k+1}&=y_k+h_k\cdot\frac{(k_1+k_2)}{2}\end{aligned}}$$

Heunova metoda

Poznámka: Obě tyto metody jsou 2.řádu (aproximovali jsme parabolou).

Klasická Runge-Kuttova metodu 4. řádu

- jedna z nejvíce používaných metod tohoto typu
- předpis metody:

$$
\begin{aligned}
k_1 &= f(x_k, y_k) \\
k_2 &= f(x_k + \frac{h}{2}, y_k + \frac{h}{2} \cdot k_1) \\
k_3 &= f(x_k + \frac{h}{2}, y_k + \frac{h}{2} \cdot k_2) \\
k_4 &= f(x_k + h, y_k + h \cdot k_3) \\
y_{k+1} &= y_k + \frac{h}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}
$$

Poznámka: Lokální diskretizační chyba, tj. chyba jednoho kroku metody, je $d_k = O(h^5)$. Globální chyba je potom o řád nižší, tj. $e_k = O(h^4)$, protože chyba metody se zvětšuje lineárně s počtem kroků $k \sim \frac{1}{h}$.

Příklad

Pomocí **Heunovy metody**, **modifikované Eulerovy metody** a **klasické Runge-Kuttovy metody 4. řádu** řešte následující úlohu na intervalu $\langle 0; 0,6\rangle$ s konstantním krokem $h = 0,2$

$$
\begin{aligned}
y' &= x - y, \\
y(0) &= 1
\end{aligned}
$$

Řešení:

(Přesné řešení: $y(x) = 2e^{-x} + x - 1$).

Předpis pro **Heunovu metodu**:

$$
\begin{aligned}
k_1 &= f(x_k, y_k) \\
k_2 &= f(x_k + h, y_k + h \cdot k_1) \\
y_{k+1} &= y_k + h \cdot \frac{k_1 + k_2}{2}
\end{aligned}
$$

x_k	$y(x_k)$	y_k	k_1	k_2	$h \cdot \frac{k_1+k_2}{2}$	e_k
0	1,000	1,000	-1,000	-0,600	-0.160	0,000
0,2	0,837	0,840	-0,640	-0,312	-0.095	-0,003
0,4	0,741	0,745	-0,345	-0,076	-0.042	-0,004
0,6	0,698	0,703				-0,005

Předpis pro **modifikovanou Eulerovu metodu**:

$$
\begin{aligned}
k_1 &= f(x_k, y_k) \\
k_2 &= f(x_k + \frac{h}{2}, y_k + \frac{h}{2} \cdot k_1) \\
y_{k+1} &= y_k + h \cdot k_2
\end{aligned}
$$

x_k	$y(x_k)$	y_k	k_1	k_2	$h \cdot k_2$	e_k
0	1,000	1,000	-1,000	-0,800	-0.160	0,000
0,2	0,837	0,840	-0,640	-0,476	-0.095	-0,003
0,4	0,741	0,745	-0,345	-0,210	-0.042	-0,004
0,6	0,698	0,703				-0,005

Poznámka:

Vidíme, že výsledky Heunovy i modifikované Eulerovy metody odpovídají výsledkům získaným metodou Taylorova typu 2. řádu (uvedené metody jsou 2. řádu).

Předpis pro **klasickou Runge-Kuttovu metodu 4. řádu**:

$$
\begin{aligned}
k_1 &= f(x_k, y_k) \\
k_2 &= f(x_k + \frac{h}{2}, y_k + \frac{h}{2} \cdot k_1) \\
k_3 &= f(x_k + \frac{h}{2}, y_k + \frac{h}{2} \cdot k_2) \\
k_4 &= f(x_k + h, y_k + h \cdot k_3) \\
y_{k+1} &= y_k + \frac{h}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}
$$

x_k	$y(x_k)$	y_k	k_1	k_2	k_3	k_4	e_k
0	1.00000000	1,00000000	-1,000000	-0,800000	-0.820000	-0.636000	0,00000000
0,2	0.83746150	0.83746666	-0.637466	-0.473720	-0.490094	-0.339447	0.00000516
0,4	0.74064009	0.74064854	-0.340648	-0.206583	-0.219990	-0.096650	0.00000845
0,6	0.69762327	0.69763364					0.00001037

Několik otázek k zamyšlení:

1. Uveďte příklad počáteční úlohy pro obyčejnou diferenciální rovnici 1. řádu (s nenulovým řešením), pro kterou budou výsledky **Eulerovy metody** totožné s výsledky **metody Taylorova typu 2. řádu**.

2. Uveďte příklad počáteční úlohy pro obyčejnou diferenciální rovnici 1. řádu (s nenulovým řešením), pro kterou bude **metoda Taylorova typu 2. řádu** totožná s **metodou Taylorova typu 3. řádu**, ale různá od **Eulerovy metody**.

3. Uveďte příklad počáteční úlohy pro obyčejnou diferenciální rovnici 1. řádu (s nenulovým řešením), pro kterou bude **modifikovaná Eulerova metoda** totožná s **Heunovou metodou**, ale různá od **Eulerovy metody**.

Kapitola 12. Počáteční úlohy pro ODR - II

Vícekrokové metody

V případě jednokrokových metod vystupovaly ve formuli pouze hodnoty y_k, y_{k+1}.

V případě vícekrokových metod vypočítáváme hodnotu y_{k+1} pomocí hodnot

$$y_{k-n}, y_{k-n+1}, \ldots, y_{k-1}, y_k, \boxed{y_{k+1}}$$

Poznámka: Pokud nepoužijeme hodnotu y_{k+1}, jedná se o **explicitní metody**, v opačném případě mluvíme o **implicitních** metodách.

Opět vyjdeme z rovnosti

$$\boxed{y' = f(x, y(x))}$$

Musí tedy platit i rovnost integrálů

$$\boxed{\int_{x_k}^{x_{k+1}} y'(x)\,dx = \int_{x_k}^{x_{k+1}} f(x, y(x))\,dx}.$$

Tedy

$$\boxed{y(x_{k+1}) = y(x_k) + \int_{x_k}^{x_{k+1}} \underbrace{f(x, y(x))}_{\overset{\text{ozn}}{=} g(x)}\,dx}. \qquad (\star)$$

Dále postupujeme tak, že funkci $g(x)$ aproximujeme interpolačním polynomem $G_n(x)$, který zintegrujeme přesně.

Poznámka: Připomeňme si odvození jednokrokové Eulerovy metody.

Funkci $g(x)$ ze vztahu $(\star)$ aproximujeme konstantní funkcí $G_0(x)$

a)

$\boxed{G_0(x) = g(x_k)}$, $x \in (x_k, x_{k+1}\rangle$

Dostáváme:

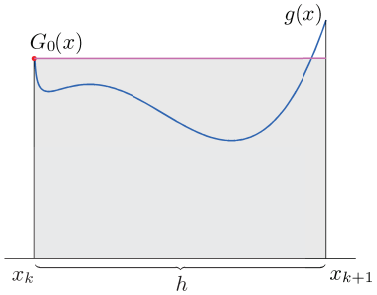
$$y_{k+1} = y_k + \int_{x_k}^{x_{k+1}} G_0(x)\,dx$$

$$y_{k+1} = y_k + hg(x_k)$$

$$\boxed{y_{k+1} = y_k + hf(x_k, y_k)}$$

Eulerova metoda

... explicitní jednokroková metoda, řád 1



b)

$\boxed{G_0(x) = g(x_{k+1})}$, $x \in (x_k, x_{k+1}\rangle$

Dostáváme:

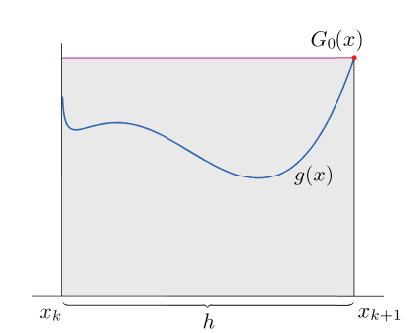
$$y_{k+1} = y_k + \int_{x_k}^{x_{k+1}} G_0(x)\,dx$$

$$y_{k+1} = y_k + hg(x_{k+1})$$

$$\boxed{y_{k+1} = y_k + hf(x_{k+1}, y_{k+1})}$$

Implicitní Eulerova metoda

... implicitní jednokroková metoda, řád 1



Poznámka:

Při použití implicitní metody je třeba zvolit počáteční aproximaci $y_{k+1}^{[0]}$ a dále realizovat iterační proces

$$\boxed{y_{k+1}^{[l+1]} = y_k + hf(x_{k+1}, y_{k+1}^{[l]})}$$

Odvození dvoukrokových metod

Funkci $g(x)$ ze vztahu $(\star)$ aproximujeme lineární funkcí $G_1(x)$

a)

$$\boxed{G_1(x) = g(x_k) + \frac{g(x_k) - g(x_{k-1})}{h}(x - x_k)},$$

$$x \in (x_k, x_{k+1}\rangle$$

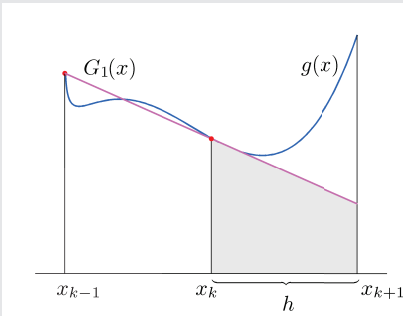
$$\int_{x_k}^{x_{k+1}} G_1(x)\,dx = g(x_k)h + \frac{h}{2}\big[g(x_k) - g(x_{k-1})\big] =$$

$$= \frac{h}{2}\big[3g(x_k) - g(x_{k-1})\big]$$

$$\boxed{y_{k+1} = y_k + \frac{h}{2}\big[3f(x_k, y_k) - f(x_{k-1}, y_{k-1})\big]}$$

Adams-Bashforthova metoda

... explicitní dvoukroková metoda, řád 2

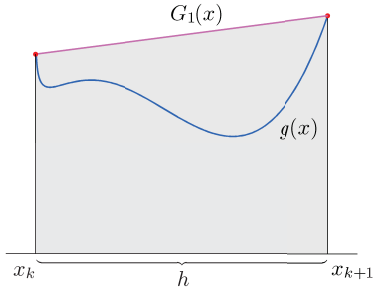


b)

$$G_1(x) = g(x_k) + \frac{g(x_{k+1}) - g(x_k)}{h}(x - x_k), \qquad x \in (x_k, x_{k+1}\rangle$$

$$\int_{x_k}^{x_{k+1}} G_1(x)\,dx = g(x_k)h + \frac{h}{2}\big[g(x_{k+1}) - g(x_k)\big] = \frac{h}{2}\big[g(x_k) + g(x_{k+1})\big]$$

$$y_{k+1} = y_k + \frac{h}{2}\big[f(x_k, y_k) + f(x_{k+1}, y_{k+1})\big]$$



Adams-Moultonova metoda

... implicitní dvoukroková metoda, řád 2

Odvození tříkrokových metod

Funkci $g(x)$ ze vztahu $(\star)$ aproximujeme kvadratickou funkcí $G_2(x)$

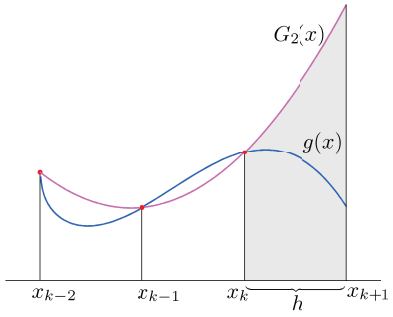
a)

$G_2(x) = \ldots$ polynom procházející body $\big[x_{k-2}, g(x_{k-2})\big]$, $\big[x_{k-1}, g(x_{k-1})\big]$ $\big[x_k, g(x_k)\big]$

$$x \in (x_k, x_{k+1}\rangle$$

$$\int_{x_k}^{x_{k+1}} G_2(x)\,dx = \ldots \text{ D.cv.}$$

$$y_{k+1} = y_k + \frac{h}{12}\big[23f(x_k, y_k) - 16f(x_{k-1}, y_{k-1}) + 5f(x_{k-2}, y_{k-2})\big]$$



Adams-Bashforthova metoda

... explicitní tříkroková metoda, řád 3

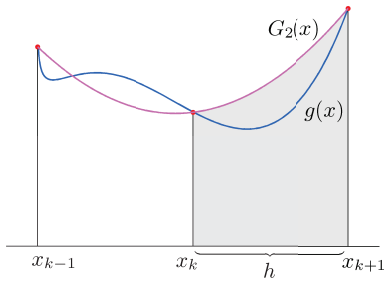
b)

$G_2(x) = \ldots$ polynom procházející body $\big[x_{k-1}, g(x_{k-1})\big]$, $\big[x_k, g(x_k)\big]$ $\big[x_{k+1}, g(x_{k+1})\big]$

$$x \in (x_k, x_{k+1}\rangle$$

$$\int_{x_k}^{x_{k+1}} G_2(x)\,dx = \ldots \text{ D.cv.}$$

$$y_{k+1} = y_k + \frac{h}{12}\big[5f(x_{k+1}, y_{k+1}) + 8f(x_k, y_k) - f(x_{k-1}, y_{k-1})\big]$$



Adams-Moultonova metoda

... implicitní tříkroková metoda, řád 3

Poznámky:

(i) U n-krokových metod je třeba znát n hodnot

$$y_{k-n+1}, y_{k-n}, \ldots, y_k$$

Na začátku výpočtu však tyto hodnoty, tj. $Y_1, \ldots, Y_{n-1}$, nejsou známy.

Pro jejich výpočet je třeba užít explicitní jednokrokové metody odpovídajícího řádu.

(ii) U implicitních metod je třeba určit aproximaci $y_{k+1}^{[0]}$ a realizovat metodu prosté iterace

$$y_{k+1}^{[l+1]} = y_k + \ldots y_{k+1}^{[l]}$$

Obecný zápis metod

Vícekrokovou (i jednokrokovou) metodu lze obecně zapsat ve tvaru

$$\sum_{j=0}^{r} \alpha_j y_{k+j} = h \sum_{j=0}^{r} \beta_j \underbrace{f(x_{k+j}, y_{k+j})}_{\approx y'(x_{k+j})}.$$

- **Explicitní Eulerova metoda** $(\alpha_0 = -1,\ \alpha_1 = 1,\ \beta_0 = 1,\ \beta_1 = 0)$

$$y_{k+1} - y_k = hf(x_k, y_k)$$

- **Implicitní Eulerova metoda** $(\alpha_0 = -1,\ \alpha_1 = 1,\ \beta_0 = 0,\ \beta_1 = 1)$

$$y_{k+1} - y_k = hf(x_{k+1}, y_{k+1})$$

- **Adams-Bashforthova metoda - dvoukroková**

$$(\alpha_0 = 0,\ \alpha_1 = -1,\ \alpha_2 = 1,\ \beta_0 = -\frac{1}{2},\ \beta_1 = \frac{3}{2},\ \beta_2 = 0)$$

$$y_{k+2} - y_{k+1} = \frac{h}{2}\big[3f(x_{k+1}, y_{k+1}) - f(x_k, y_k)\big] \qquad (k := k+1)$$

- **Adams-Moultonova metoda - dvoukroková**

$$(\alpha_0 = -1,\ \alpha_1 = 1,\ \beta_0 = \frac{1}{2},\ \beta_1 = \frac{1}{2})$$

$$y_{k+1} - y_k = \frac{h}{2}\big[f(x_k, y_k) + f(x_{k+1}, y_{k+1})\big]$$

- **Adams-Bashforthova metoda - tříkroková**

$$(\alpha_0 = 0,\ \alpha_1 = 0,\ \alpha_2 = -1,\ \alpha_3 = 1,\ \beta_0 = \frac{5}{12},\ \beta_1 = -\frac{4}{3},\ \beta_2 = \frac{23}{12})$$

$$\boxed{y_{k+3}-y_{k+2}=\frac{h}{12}\big[23f(x_{k+2},y_{k+2})-16f(x_{k+1},y_{k+1})+5f(x_k,y_k)\big]} \qquad (k:=k+2)$$

- **Adams-Moultonova metoda - tříkroková**

$$(\alpha_0=0,\ \alpha_1=-1,\ \alpha_2=1,\ \beta_0=-\frac{1}{12},\ \beta_1=\frac{2}{3},\ \beta_2=\frac{5}{12})$$

$$\boxed{y_{k+2}-y_{k+1}=\frac{h}{12}\big[5f(x_{k+2},y_{k+2})+8f(x_{k+1},y_{k+1})-f(x_k,y_k)\big]} \qquad (k:=k+1)$$

Definice: **Lokální diskretizační chybou** metody rozumíme

$$\boxed{\tau_k=\frac{1}{h}\big[\sum_{j=0}^{r}\alpha_j y(x_{k+j})-h\sum_{j=0}^{r}\beta_j y'(x_{k+j})\big]}.$$

Definice: Řekneme, že metoda je **konzistentní**, je-li splněna podmínka

$$\boxed{\tau_k(h)\to 0 \quad \text{pro } h\to 0}.$$

Použijeme-li Taylorův rozvoj, získáme:

$$\begin{aligned}
y(x_k)&=y(x_k)\\
y(x_{k+1})&=y(x_k)+hy'(x_k)+\frac{1}{2}h^2y''(x_k)+\dots\\
y(x_{k+2})&=y(x_k)+2hy'(x_k)+\frac{1}{2}(2h)^2y''(x_k)+\dots\\
y(x_{k+3})&=y(x_k)+3hy'(x_k)+\frac{1}{2}(3h)^2y''(x_k)+\dots\\
&\vdots\\
y(x_{k+j})&=y(x_k)+jhy'(x_k)+\frac{1}{2}(jh)^2y''(x_k)+\dots
\end{aligned}$$

a analogicky pro derivaci (formálně připíšeme $'$):

$$\begin{aligned}
y'(x_k)&=y'(x_k)\\
y'(x_{k+1})&=y'(x_k)+hy''(x_k)+\frac{1}{2}h^2y'''(x_k)+\dots\\
y'(x_{k+2})&=y'(x_k)+2hy''(x_k)+\frac{1}{2}(2h)^2y'''(x_k)+\dots\\
y'(x_{k+3})&=y'(x_k)+3hy''(x_k)+\frac{1}{2}(3h)^2y'''(x_k)+\dots\\
&\vdots\\
y'(x_{k+j})&=y'(x_k)+jhy''(x_k)+\frac{1}{2}(jh)^2y'''(x_k)+\dots
\end{aligned}$$

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13

Tj. jde vždy polynom v proměnné h ($\Rightarrow$ i τ_k).

Dosazením do vztahu pro lokální diskretizační chybu

$$\tau_k=\frac{1}{h}\big[\sum_{j=0}^{r}\alpha_j y(x_{k+j})-h\sum_{j=0}^{r}\beta_j y'(x_{k+j})\big],$$

tj.

$$\tau_k=\frac{1}{h}\Big[\big(\alpha_0y(x_k)+\alpha_1y(x_{k+1})+\alpha_2y(x_{k+2})+\dots+\alpha_ry(x_{k+r})\big)-h\big(\beta_0y'(x_k)+\beta_1y'(x_{k+1})+\dots+\beta_ry'(x_{k+r})\big)\Big],$$

dostaneme

$$\tau_k=\frac{1}{h}\Big[\underbrace{y(x_k)\big(\sum_{j=0}^{r}\alpha_j\big)}_{\text{konstantní členy}}+\underbrace{hy'(x_k)\big(\sum_{j=0}^{r}(j\alpha_j-\beta_j)\big)}_{\text{lineární členy}}+\underbrace{h^2y''(x_k)\big(\frac{1}{2}\sum_{j=0}^{r}j^2\alpha_j-\sum_{j=0}^{r}j\beta_j\big)}_{\text{kvadratické členy}}+\underbrace{h^3y'''(x_k)(\dots)}_{\text{kubické členy}}+\dots\Big].$$

Po roznásobení:

$$\boxed{\tau_k=\frac{1}{h}y(x_k)\big(\sum_{j=0}^{r}\alpha_j\big)+y'(x_k)\big(\sum_{j=0}^{r}(j\alpha_j-\beta_j)\big)+h\big(\dots\big)+h^2\big(\dots\big)+\dots}$$

D.cv: Ukažte, že dříve odvozené metody jsou konzistentní.

Definice: Polynomy v proměnných v a w ve tvaru

$$\boxed{\rho(v)=\sum_{j=0}^{r}\alpha_jv^j} \quad \text{a} \quad \boxed{\sigma(w)=\sum_{j=0}^{r}\beta_jw^j}$$

nazýváme **charakteristické polynomy** metody.

Poznámka: Metoda je konzistentní, platí-li

$$\boxed{\rho(1)=0} \quad \text{a} \quad \boxed{\rho'(1)=\sigma(1)}$$

Poznámka:

Nestabilita - do výsledku je vnášena chyba, jejíž vliv zesiluje až celý výpočet znehodnotí.

Příčiny - špatná podmíněnost úlohy (nezávisí na volbě metody); nevhodná metoda nebo příliš velký krok.

Definice: Mějme dánu počáteční úlohu s lipschitzovskou funkcí f:

$$\boxed{\begin{aligned}&y'=f(x,y), \quad x\in(0,T\rangle\\&y(0)=\eta\end{aligned}}$$

Řekneme, že metoda je **konvergentní**, když platí:

$$\boxed{\lim_{\substack{h\to 0\\ Nh=T}} y_N=y(T)}$$

a

$$\boxed{\lim_{h\to 0}y_k=\eta \quad \text{pro } k=0,1,\dots,r-1}$$

pro každé pevné T (uvažujeme r-krokovou metodu).

Poznámka: Je zřejmé, že nutnou podmínkou konvergence je konzistence metody. Je i podmínkou postačující?

Věta Uvažujme úlohu

$$\boxed{\begin{aligned} y' &= \lambda y, \quad x \in (0,T\rangle \\ y(0) &= \eta \end{aligned}}.$$

Eulerova metoda je pro tuto úlohu konvergentní.

Důkaz:

$$\begin{array}{lcll} y(x_{k+1}) & = & y(x_k) & + \; h \overbrace{\lambda y(x_k)}^{=y'(x_k)} \quad + \; h\tau_k \\ y_{k+1} & = & y_k & + \; h\,\lambda y_k \\ \hline \underbrace{y(x_{k+1}) - y_{k+1}}_{E_{k+1}} & = & \underbrace{y(x_k) - y_k}_{E_k} & + \; h\,\lambda \underbrace{(y(x_k) - y_k)}_{E_k} \quad + \; h\tau_k \end{array}$$

$$\boxed{E_{k+1} = E_k(1+h\lambda) + h\tau_k}$$

Tj.

$$\begin{aligned} E_1 &= \overbrace{E_0(1+h\lambda)}^{=0} + h\tau_0 \\ E_2 &= E_1(1+h\lambda) + h\tau_1 \\ &\vdots \end{aligned}$$

Obecně lze psát:

$$\boxed{E_k = \underbrace{(1+h\lambda)^k E_0}_{(1+h\lambda)^{k-1}E_1} + h\sum_{m=1}^{k}(1+h\lambda)^{k-m}\tau_{m-1}}$$

Dále platí (z Taylorova rozvoje e^x):

$$|1+h\lambda| \le e^{h|\lambda|}$$

a tedy

$$(1+h\lambda)^{k-m} \le \mathrm{e}^{(k-m)h|\lambda|} \le \mathrm{e}^{kh|\lambda|} \le \mathrm{e}^{T|\lambda|}$$

Potom

$$|E_k| \le \mathrm{e}^{|\lambda|T}\Big[\overbrace{|E_0|}^{=0} + hk \max_{1\le m\le k} \underbrace{|\tau_{m-1}|}_{(**)}\Big]$$
$$(*)$$

$(*)$ $|E_1| = |h\tau_0| \to 0$ pro $h \to 0$.

$(**) \to 0$ pro $h \to 0$, protože je Eulerova metoda konzistentní.

□

Důkaz:

Platí

$$\begin{array}{lcll} y(x_{k+1}) & = & y(x_k) & + \; hf(x_k, y(x_k)) \quad + \; h\tau_k \\ y_{k+1} & = & y_k & + \; hf(x_k, y_k) \\ \hline \underbrace{y(x_{k+1}) - y_{k+1}}_{E_{k+1}} & = & \underbrace{y(x_k) - y_k}_{E_k} & + \; h\Big[f(x_k, y(x_k)) - f(x_k, y_k)\Big] \quad + \; h\tau_k \end{array}$$

Funkce f je lipschitzovsky spojitá:

$$|f(x_k, y(x_k)) - f(x_k, y_k)| \le L \cdot |y(x_k) - y_k| \quad \forall x_k \in \langle 0, T\rangle$$

Pak lze psát:

$$|E_{k+1}| \le |E_k| + h\,L\,|E_k| + h\,|\tau_k|.$$

Dále je důkaz stejný $(|\lambda| = L)$

$$|E_k| \le \mathrm{e}^{LT}\Big[\underbrace{|E_0|}_{=0} + hk \max_{1\le m\le k} \underbrace{|\tau_{m-1}|}_{\to 0\,(h\to 0)}\Big]$$

□

Konvergence vícekrokových metod

Příklad: Uvažujme vícekrokovou metodu ve tvaru

$$\boxed{y_{k+2} = 3y_{k+1} - 2y_k - hf(x_k, y_k)}.$$

Obecný zápis byl

$$\sum_{j=0}^{r} \alpha_j y_{k+j} = h\sum_{j=0}^{r} \beta_j f(x_{k+j}, y_{k+j}).$$

Platí:

$$\alpha_0 = 2, \;\; \alpha_1 = -3, \;\; \alpha_2 = 1, \;\; \sum_{j=0}^{2}\alpha_j = 0$$

$$\beta_0 = -1, \;\; \beta_1 = 0, \;\; \beta_2 = 0, \;\; \sum_{j=0}^{2}\beta_j = -1 \overset{?}{=} \sum_{j=0}^{2} j\alpha_j = 0\cdot\alpha_0 + 1\cdot\alpha_1 + 2\cdot\alpha_2 = 0 + (-3) + 2 = -1$$

$\Rightarrow$ metoda je konzistentní.

Touto metodou budeme řešit počáteční úlohu

$$\boxed{\begin{aligned} y' &= 0, \quad x \in (0,T\rangle \\ y(0) &= 0 \end{aligned}}.$$

Pro tuto úlohu má metoda tvar

$$\boxed{y_{k+2} = 3y_{k+1} - 2y_k}.$$

Obecně lze psát:

$$\boxed{y_k = 2y_0 - y_1 + 2^k(y_1 - y_0)}.$$

Důkaz: (pomocí úplné matematické indukce)

1. $k = 2,\ k = 3$

$$y_2 = 2y_0 - y_1 + 2^2(y_1 - y_0) = 3y_1 - 2y_0 \qquad \checkmark$$

$$y_3 = 2y_0 - y_1 + 2^3(y_1 - y_0) = 7y_1 - 6y_0 \overset{?}{=} 3y_2 - 2y_1 = 3\big(3y_1 - 2y_0\big) - 2y_1 \qquad \checkmark$$

2. $\left.\begin{aligned} y_k &= 2y_0 - y_1 + 2^k(y_1 - y_0) \\ y_{k+1} &= 2y_0 - y_1 + 2^{k+1}(y_1 - y_0) \end{aligned}\right\} \overset{?}{\Rightarrow} \quad y_{k+2} = 2y_0 - y_1 + 2^{k+2}(y_1 - y_0)$

$$y_{k+2} = 3y_{k+1} - 2y_k = 3\big(2y_0 - y_1 + 2^{k+1}(y_1 - y_0)\big) - 2\big(2y_0 - y_1 + 2^k(y_1 - y_0)\big) =$$
$$= 2y_0 - y_1 + \underbrace{(6-2)}_{=2^2} 2^k(y_1 - y_0) \qquad \checkmark$$

□

Problém:

Pokud $y_1 = y_0 = y(0) = 0 \quad \Rightarrow \quad y_k = 0 \ \forall k.$ ✓

Pokud se y_1 bude lišit (i když velmi málo) od 0, pak pro $k \to \infty$: $y_k \to \infty$. ↯

Považujeme-li rovnost $y_{k+2} = 3y_{k+1} - 2y_k$ za diferenční rovnici, můžeme ji řešit.
Předpokládáme, že $y_k = Cv^k$. Pak lze psát

$$Cv^{k+2} = 3Cv^{k+1} - 2Cv^k$$
$$Cv^2 = 3Cv - 2C$$
$$v^2 - 3v + 2 = 0 \qquad (\star)$$
$$v_{1,2} = \frac{3 \pm \sqrt{9-8}}{2}$$
$$v_1 = 2, \quad v_2 = 1$$
$$y_k = C_1 2^k - C_2$$

Dále víme, že pro

$$k = 0: \quad y_0 = C_1 + C_2$$
$$k = 1: \quad y_1 = 2C_1 + C_2$$

$\Rightarrow \quad C_1 = y_1 - y_0;$
$C_2 = y_0 - c_1 = y_0 - y_1 + y_0 = 2y_0 - y_1$

$$\boxed{y_k = (y_1 - y_0)2^k + 2y_0 - y_1}$$

$$y_k = (y_1 - y_0)\underbrace{2}_{=v_1}{}^k + (2y_0 - y_1)\underbrace{1}_{=v_2}{}^k$$

Připomeňme, že vícekrokovou metodu jsme zapisovali ve tvaru

$$\boxed{\sum_{j=0}^{r} \alpha_j y_{k+j} = h \sum_{j=0}^{r} \beta_j f(x_{k+j}, y_{k+j})}$$

a charakteristické polynomy jsme definovali

$$\boxed{\rho(v) = \sum_{j=0}^{r} \alpha_j v^j} \quad \text{a} \quad \boxed{\sigma(w) = \sum_{j=0}^{r} \beta_j w^j}.$$

O stabilitě výpočtu rozhodují kořeny polynomu $\rho(v)$ (viz $(\star)$).

Pro kořeny $\bar{v}_j$ polynomu $\rho(v)$ musí platit $\boxed{|\bar{v}_j| \le 1}$.

TRIAL
Numerické metody KMA/NM
Josef Daněk 13.2.2o13

Definice: Řekneme, že metoda je **D-stabilní**, pokud kořeny charakteristického polynomu $\rho(v)$ splňují podmínky:
(i) $|\bar{v}_j| \le 1$ pro $j = 1, 2, \ldots, r$,
(ii) je-li $\bar{v}_j$ násobný kořen, potom $|\bar{v}_j| < 1$.

Poznámky:

- Je-li metoda D-stabilní, nebude v průběhu výpočtu radikálně zvětšovat jednokrokovou chybu.
- Uvažujme Eulerovu metodu

$$y_{k+1} - y_k = h\, f(x_k, y_k)$$
$$\rho(v) = v - 1 = 0 \ \Rightarrow \ \bar{v} = 1$$

$\Rightarrow$ Eulerova metoda je D-stabilní.

Odhad chyby metodou polovičního kroku

Pro globální chybu metody lze psát

$$\boxed{\underbrace{y(x)}_{\text{přesné}} = \underbrace{y(x,h)}_{(*)} + \underbrace{E_h}_{Ch^p} + \underbrace{F_h}_{O(h^r),\ r>p}} \qquad (\bullet)$$

$(*)$ $\quad y(x,h) = y_k(h), \quad x \in \langle x_k, x_{k+1} \rangle, \quad k = 0, 1, \ldots, N-1$

pro poloviční krok

$$\boxed{y(x) = y\big(x, \frac{h}{2}\big) + E_{\frac{h}{2}} + F_{\frac{h}{2}}} \qquad (\bullet\bullet)$$

Po odečtení $(\bullet) - (\bullet\bullet)$:

$$0 = y(x,h) - y\big(x, \frac{h}{2}\big) + E_h - E_{\frac{h}{2}} + \ldots$$
$$0 \approx y(x,h) - y\big(x, \frac{h}{2}\big) + (2^p - 1)E_{\frac{h}{2}}$$
$$\boxed{E_{\frac{h}{2}} = \frac{y\big(x, \frac{h}{2}\big) - y(x,h)}{2^p - 1}}$$
$$\Rightarrow \quad \boxed{y(x) \approx y\big(x, \frac{h}{2}\big) + E_{\frac{h}{2}}}$$

Poznámka: Opět lze použít **Richardsonovu extrapolaci**

- aktivní extrapolace
  extrapolaci provádíme v každém kroku (extrapolované y_k použijeme pro výpočet y_{k+1})
- pasivní extrapolace
  vypočteme y_k, $k = 0, 1, \ldots, N-1$ s různými parametry h potom provedeme extrapolaci

Algoritmus prediktor-korektor

Poznámka: Jde o obecné schéma výpočtu.

Princip:

Předpokládejme, že máme dostatečně přesně vypočítány hodnoty $y_0, y_1, \dots, y_{k-1}$ nějakou explicitní jednokrokovou metodou.

Nyní chceme počítat y_k.

1) Nejprve nějakou explicitní metodou určíme nultou iteraci $y_k^{[0]}$ jako vstupní hodnotu pro další výpočet (PREDIKTOR).

2) Vypočteme hodnotu pravé strany $F_k^{[s]} = f(x_k, y_k^{[s]})$.

3) Vypočteme lepší aproximaci $y_k^{[s+1]}$ pomocí nějaké implicitní metody s využitím $F_k^{[s]} =: f_k$ (KOREKTOR).

Pomocí kroků **2)** a **3)** určíme N iterací $y_k^{[1]}, y_k^{[2]}, \dots, y_k^{[N]}$ (N – dáno).

Na závěr přiřadíme $y_k = y_k^{[N]}$.

Stejný postup opakujeme pro $y_{k+1}, y_{k+2}, \dots$.

Poznámka: Dané schéma lze použít na různé metody. Je žádoucí použít explicitní a explicitní metodu stejného řádu (pro zachování přesnosti). Volba konkrétních metod je na nás.

Poznámka: Označíme-li operaci:

a) P ... prediktor
b) E ... vyčíslení (*evaluation*)
c) C ... korektor

Můžeme toto schéma zapsat ve tvaru:

$P(EC)^N$ případně $P(EC)^N E$, vyčíslujeme-li ještě $F_k = f(x_k, y_k^{[N]})$ (což je lepší).
Dostaneme pak různé varianty tohoto schématu:

$$\begin{array}{ll} PEC & , \ PECE \\ P(EC)^2 & , \ P(EC)^2E \\ P(EC)^3 & , \ P(EC)^3E \\ \vdots & , \ \vdots \end{array}$$

Příklad: Řešte algoritmem prediktor-korektor založeném na Adamsových metodách druhého řádu na intervalu $\langle 0; 0,6\rangle$ počáteční úlohu:

$$\boxed{\begin{aligned} &y' = y + \mathrm{e}^x, \qquad \text{tj.} \quad f(x, y(x)) = y + \mathrm{e}^x \\ &y(0) = -1 \end{aligned}}$$

Přesné řešení: $y = \mathrm{e}^x(x-1)$.

Použijeme algoritmus typu PEC.
Vzorec prediktoru má tvar:

$$y_{n+1}^{[0]} = y_n + \frac{h}{2}(3F_n - F_{n-1})$$

Korektor:

$$y_{n+1} = y_n + \frac{h}{2}(F_{n+1}^{[0]} + F_n)$$

Volte krok $h = 0,2$.

n	x_n	$\overbrace{y(x_n)}^{\text{přesné}}$	$y_n^{[0]}$	$F_n^{[0]}$	y_n	e_n
0	0	-1		•• 0 $\leftrightarrow$	-1	0
1	0,2	$-0,9771$		•• 0,2425 $\leftrightarrow$ •	$-0,9789$	0,0018
2	0,4	$-0,8950$	$^P -0,9061 \hookrightarrow$	$^E 0,5857 \hookrightarrow$	$^C -0,8960$	0,0010
3	0,6	$-0,7288$	$^P -0,7445 \hookrightarrow$	$^E 1,0776 \hookrightarrow$	$^C -0,7296$	0,0008

•
Pro určení hodnoty y_1 použijeme např.
jednokrokovou modifikovanou Eulerovu metodu (2. řádu):

$$\begin{aligned} k_1 &= f(x_0, y_0) = y_0 + \mathrm{e}^{x_0} = \\ &= -1 + 1 = 0 \\ k_2 &= f(x_0 + h/2, y_0 + h/2 \cdot k_1) = \\ &= -1 + \mathrm{e}^{0,1} \doteq 0,1051 \\ y_1 &= y_0 + h \cdot k_2 \doteq \\ &\doteq -1 + 0,2 \cdot 0,1051 = -0,9789 \end{aligned}$$

••
Určíme hodnoty F_0 a F_1.

Odhad chyby pomocí algoritmu prediktor-korektor

Za předpokladu, že se hodnota derivace $y^{(p+1)}$, kde p je řád metody, příliš nemění, lze odvodit odhad pro lokální chybu algoritmu

$$\boxed{d_k \approx \frac{c_{p+1}^C}{c_{p+1}^P - c_{p+1}^C}(y_{k+1}^C - y_{k+1}^P)}$$

kde

c_{p+1}^P, c_{p+1}^C ... konstanty v lokální chybě metody, tj. $d_k = c_{p+1}\, h^{p+1}\, y^{p+1}(x_k)$

y_{k+1}^C ... vypočteno korektorem

y_{k+1}^P ... vypočteno prediktorem

Podmíněnost úlohy a stabilita metody

Příklad Řešme počáteční úlohu

$$\boxed{\begin{aligned} &y' = y - \frac{x}{3} - \frac{2}{3}, \quad x \in (0, T\rangle \\ &y(0) = 1 \end{aligned}}.$$

řešení této počáteční úlohy má tvar $y = \frac{x}{3} + 1$

obecné řešení dané rovnice je $y = A\mathrm{e}^x + \frac{x}{3} + 1$

$\Rightarrow$ úloha je špatně podmíněná!

$$(y(0) = 1 + \varepsilon \quad \rightarrow \quad y = \varepsilon e^x + \frac{x}{3} + 1)$$

pro řešení je třeba použít metodu vyššího řádu a dostatečně přesnou aritmetiku

Příklad Pomocí Eulerovy metody řešme počáteční úlohu

$$\boxed{\begin{aligned} y' &= \lambda y, \quad x \in (0, T\rangle \\ y(0) &= 1 \end{aligned}} \qquad (\clubsuit)$$

přesné řešení úlohy je $y(x) = e^{\lambda x}$

v tomto případě má Eulerova metoda tvar

$$y_{k+1} = y_k + h\lambda y_k$$

$$y_{k+1} = (1 + \underbrace{h\lambda}_{=\bar{h}})y_k$$

Je-li $|1 + \bar{h}| < 1$, pak je posloupnost y_k omezená a klesající.

Je-li $|1 + \bar{h}| > 1$, pak posloupnost y_k neomezeně roste (osciluje).

$$|1 + \bar{h}| < 1 \quad \Leftrightarrow \quad \bar{h} = h\lambda \in (-2, 0)$$

Pro konkrétní úlohu:

$$\boxed{\begin{aligned} y' &= -5y, \quad x \in (0, 1\rangle \\ y(0) &= 1 \end{aligned}}$$

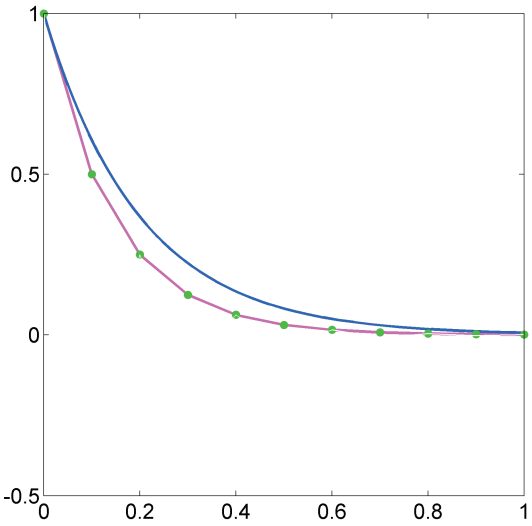
a krok h pomocí Eulerovy metody dostaneme:

1) $h = 0,1$

$$y_{k+1} = \underbrace{(1 - 0,1 \cdot 5)}_{=0,5} y_k$$

x_k	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y_k	1.0000	0.5000	0.2500	0.1250	0.0625	0.0313	0.0156	0.0078	0.0039	0.0020	0.0010

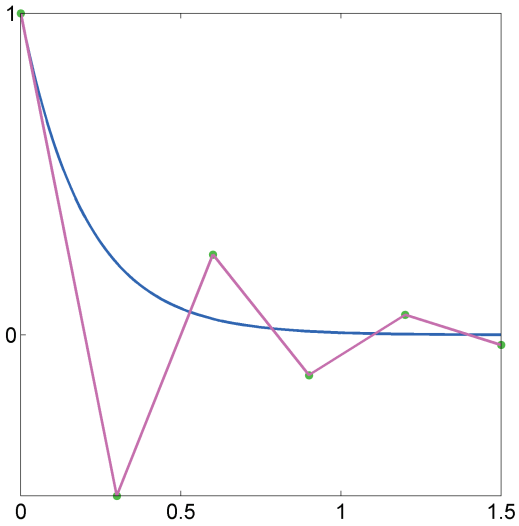
Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



2) $h = 0,3$

$$y_{k+1} = \underbrace{(1 - 0,3 \cdot 5)}_{=-0,5} y_k$$

x_k	0	0.3	0.6	0.9	1.2	1.5
y_k	1.0000	−0.5000	0.2500	−0.1250	0.0625	−0.0313

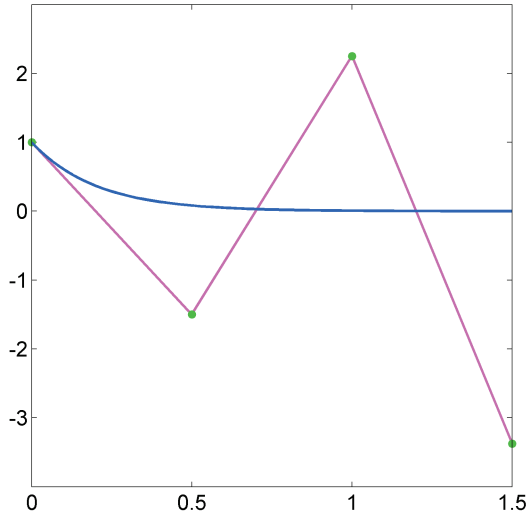


3) $h = 0,5$

$$y_{k+1} = \underbrace{(1 - 0,5 \cdot 5)}_{= -1,5} y_k$$

x_k	0	0.5	1.0	1.5
y_k	1.0000	−1.5000	2.2500	−3.3750

Numerické metody
Josef Daněk
KMA/NM
13.2.2o13



Tj. v případě velké záporné hodnoty $\lambda = -2000 \quad \rightarrow \quad h < \frac{2}{2000} = 0,0001$

řešení $y(x) = \mathrm{e}^{-2000x} \rightarrow 0$ pro $x \rightarrow \infty$

$y_k = (1 + h\lambda)^k y_0 = (1 - 2000h)^k \rightarrow 0$ pro $k \rightarrow \infty$

Poznámka:

Řekneme, že metoda je pro $\bar{h}$ **absolutně stabilní**, jestliže při h a λ: $h\lambda = \bar{h}$, všechna přibližná řešení mají pro $k \rightarrow \infty$ limitu rovnou 0 $(y_k \rightarrow 0)$.

Úlohu (♣) uvažujeme proto, že rovnice $y' = f(x, y)$ po linearizování přejde na tvar $y' = \underbrace{\frac{\partial f}{\partial y}}_{= \lambda} y$.

$\Rightarrow$ Stabilita závisí jak na metodě, tak na úloze.

Připomeňme, že platí:

$$\begin{array}{rcl} y(x_{k+1}) & = & y(x_k) + h\lambda y(x_k) + h\tau_k \\ y_{k+1} & = & y_k + h\lambda y_k \\ \hline E_{k+1} & = & E_k + h\lambda E_k + h\tau_k \end{array}$$

$$|E_{k+1}| \leq |1 + h\lambda| \cdot |E_k| + h|\tau_k|$$

Chceme-li, aby $|E_k| \rightarrow 0$ pro $k \rightarrow \infty$, musíme požadovat

$$\boxed{|1 + h\lambda| < 1}.$$

Tuto úvahu můžeme učinit i pro obecnou vícekrokovou metodu

$$\sum_{j=0}^{r} \alpha_j y_{k+j} = h \sum_{j=0}^{r} \beta_j \lambda y_{k+j}$$

$$\sum_{j=0}^{r} (\alpha_j - \bar{h}\beta_j) y_{k+j} = 0$$

Definujeme **polynom stability**:

$$\Pi(u, \bar{h}) = \sum_{j=0}^{r} (\alpha_j - \bar{h}\beta_j) u^j .$$

Definice: **Oblastí absolutní stability metody** nazýváme množinu

$$\mathcal{A} = \left\{ \bar{h} \in \mathbb{C} : \ |\bar{u}_j| \leq 1 \quad \forall \bar{u}_j : \ \Pi(\bar{u}_j, \bar{h}) = 0 \right\}$$

$|\bar{u}_j| < 1$ pro násobné kořeny

tj. „množina hodnot $\bar{h}$ v komplexní rovině, pro které kořeny polynomu $\Pi(u, \bar{h})$ splňují podmínku $|\bar{u}_j| < 1$“

Příklady

1. (explicitní) Eulerova metoda

$$y_{k+1} - y_k = h f(x_k, y_k)$$

Koeficienty metody

$$\alpha_0 = -1, \ \alpha_1 = 1, \ \beta_0 = 1, \ \beta_1 = 0$$

Polynom stability

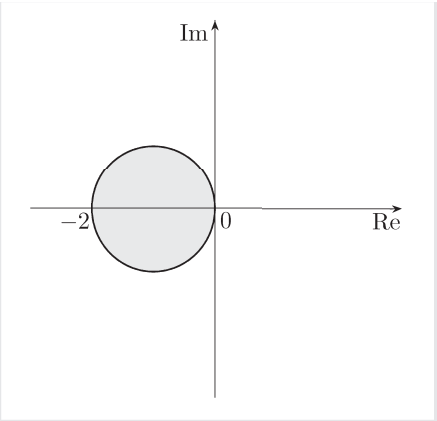
$$\Pi(u, \bar{h}) = \sum_{j=0}^{r} (\alpha_j - \bar{h}\beta_j) u^j$$

$$\Pi(u, \bar{h}) = (-1 - \bar{h}) + u = u - 1 - \bar{h}$$

Kořen

$$\bar{u} = 1 + \bar{h}; \qquad |\bar{u}| = \left|1 + \bar{h}\right| \leq 1$$

Oblast absolutní stability metody



2. implicitní Eulerova metoda

$$y_{k+1} - y_k = h f(x_{k+1}, y_{k+1})$$

Koeficienty metody

$$\alpha_0 = -1, \ \alpha_1 = 1, \ \beta_0 = 0, \ \beta_1 = 1$$

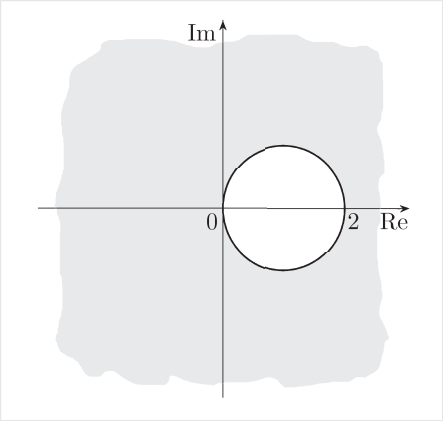
Polynom stability

$$\Pi(u, \bar{h}) = -1 + (1 - \bar{h})u = (1 - \bar{h})u - 1$$

Kořen

$$|\bar{u}| = \frac{1}{\left|1 - \bar{h}\right|} \leq 1; \qquad \left|1 - \bar{h}\right| \geq 1$$

Oblast absolutní stability metody



Intervaly absolutní stability

Eulerova metoda $(-2, 0)$

Implicitní Eulerova metoda $(-\infty, 0) \cup (2, \infty)$

Příklad Stanovte oblast absolutní stability pro tzv. obdélníkové pravidlo, tj. metodu s předpisem

$$y_{k+1} = y_{k-1} + 2hf(x_k, y_k).$$

Koeficienty metody

$$\alpha_0 = -1,\ \alpha_1 = 0,\ \alpha_2 = 1,\ \beta_0 = 0,\ \beta_1 = 2,\ \beta_2 = 0$$

Polynom stability

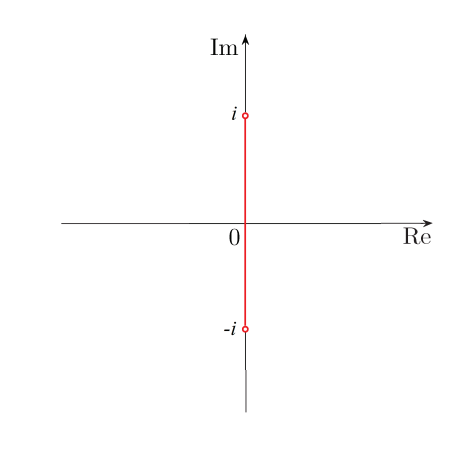
$$\Pi(u, \bar{h}) = -1 - 2\bar{h}u + u^2$$

Kořeny

$$u_{1,2} = \frac{2\bar{h} \pm \sqrt{4\bar{h}^2 + 4}}{2} = \bar{h} \pm \sqrt{\bar{h}^2 + 1}$$

Pro oblast absolutní stability musí platit (D.cv.)

$$|u_1| < 1 \quad \wedge \quad |u_2| < 1$$



Příklad Stanovte oblast absolutní stability pro tzv. lichoběžníkové pravidlo, tj. metodu s předpisem

$$y_{k+1} = y_k + \frac{h}{2}\left[f(x_k, y_k) + f(x_{k+1}, y_{k+1})\right].$$

Koeficienty metody

$$\alpha_0 = -1,\ \alpha_1 = 1,\ \beta_0 = \frac{1}{2},\ \beta_1 = \frac{1}{2}$$

Polynom stability

$$\Pi(u, \bar{h}) = (-1 - \frac{1}{2}h) + (1 - \frac{1}{2}h)u$$

Kořen

$$u = \frac{2 + \bar{h}}{2 - \bar{h}}$$

Platí

$$|u| < 1 \quad \text{pro} \quad \mathrm{Re}\,\bar{h} < 0$$

Neboť

$$\left|\frac{2 + \bar{h}}{2 - \bar{h}}\right| < 1 \ \Leftrightarrow\ \left|2 + \bar{h}\right| < \left|2 - \bar{h}\right| \ \Leftrightarrow\ \left|\bar{h} - (-2)\right| < \left|\bar{h} - 2\right|,$$

tj. vzdálenost $\bar{h}$ od -2 je menší než vzdálenost od 2.

Oblast absolutní stability metody