4.9 Metoda prosté iterace (MPI)

Rovnici $f(x) = 0$ převedeme na ekvivalentní tvar $\varphi(x) = x$.

Počáteční odhad x_0, $\quad x_i = \varphi(x_{i-1})\,.$

Podmínka ukončení: $\quad |x_i - x_{i-1}| < \eta\,.$

4.9 Metoda prosté iterace (MPI)

Rovnici $f(x) = 0$ převedeme na ekvivalentní tvar $\varphi(x) = x$.

Počáteční odhad x_0, $\quad x_i = \varphi(x_{i-1})\,.$

Podmínka ukončení: $\quad |x_i - x_{i-1}| < \eta\,.$

4.9 Metoda prosté iterace (MPI)

Rovnici $f(x) = 0$ převedeme na ekvivalentní tvar $\varphi(x) = x$.

Počáteční odhad x_0, $\quad x_i = \varphi(x_{i-1})\,.$
Podmínka ukončení: $\quad |x_i - x_{i-1}| < \eta\,.$

Tvrzení 4.1 *Pokud MPI konverguje k $\tilde{x}$ a φ je v $\tilde{x}$ spojitá, pak $\varphi(\tilde{x}) = \tilde{x}$, $f(\tilde{x}) = 0$.*

Důkaz.

4.9 Metoda prosté iterace (MPI)

Rovnici $f(x) = 0$ převedeme na ekvivalentní tvar $\varphi(x) = x$.

Počáteční odhad x_0, $\quad x_i = \varphi(x_{i-1})\,.$
Podmínka ukončení: $\quad |x_i - x_{i-1}| < \eta\,.$

Tvrzení 4.1 *Pokud MPI konverguje k $\tilde{x}$ a φ je v $\tilde{x}$ spojitá, pak $\varphi(\tilde{x}) = \tilde{x}$, $f(\tilde{x}) = 0$.*

Důkaz.

$$\tilde{x} = \lim_{i\to\infty} x_i = \lim_{i\to\infty} \varphi(x_{i-1}) = \varphi\left(\lim_{i\to\infty} x_{i-\cancel{1}}\right) = \varphi(\tilde{x})$$

4.9 Metoda prosté iterace (MPI)

Rovnici $f(x) = 0$ převedeme na ekvivalentní tvar $\varphi(x) = x$.

Počáteční odhad x_0, $x_i = \varphi(x_{i-1})\,.$
Podmínka ukončení: $|x_i - x_{i-1}| < \eta\,.$

Tvrzení 4.1 *Pokud MPI konverguje k $\tilde{x}$ a φ je v $\tilde{x}$ spojitá, pak $\varphi(\tilde{x}) = \tilde{x}$, $f(\tilde{x}) = 0$.*

Důkaz.

$$\varphi(\tilde{x}) = \varphi\left(\lim_{i\to\infty} x_i\right) = \lim_{i\to\infty} \varphi(x_i) = \lim_{i\to\infty} x_{i+1} = \lim_{i\to\infty} x_i = \tilde{x}\,.$$

□

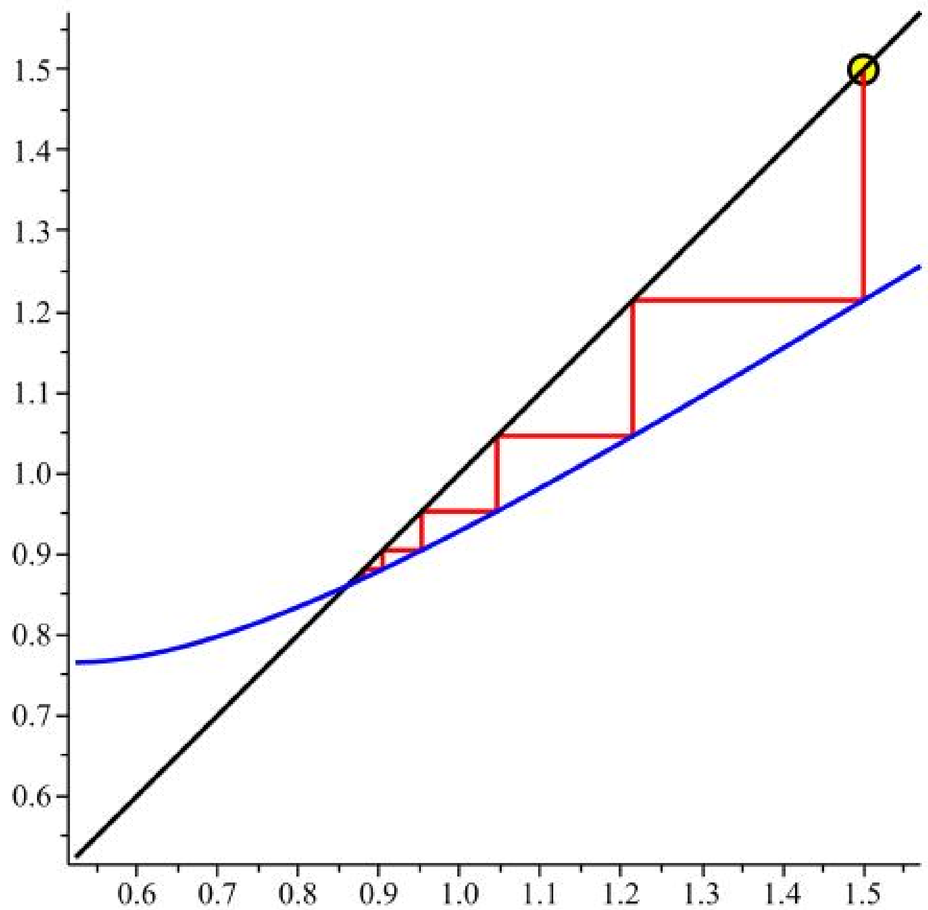
Příklad použití MPI

Příklad 4.1 *Hledáme nejmenší kladné řešení rovnice* $f(x) = 0$*, kde* $f(x) = x - \operatorname{cotg} x$.

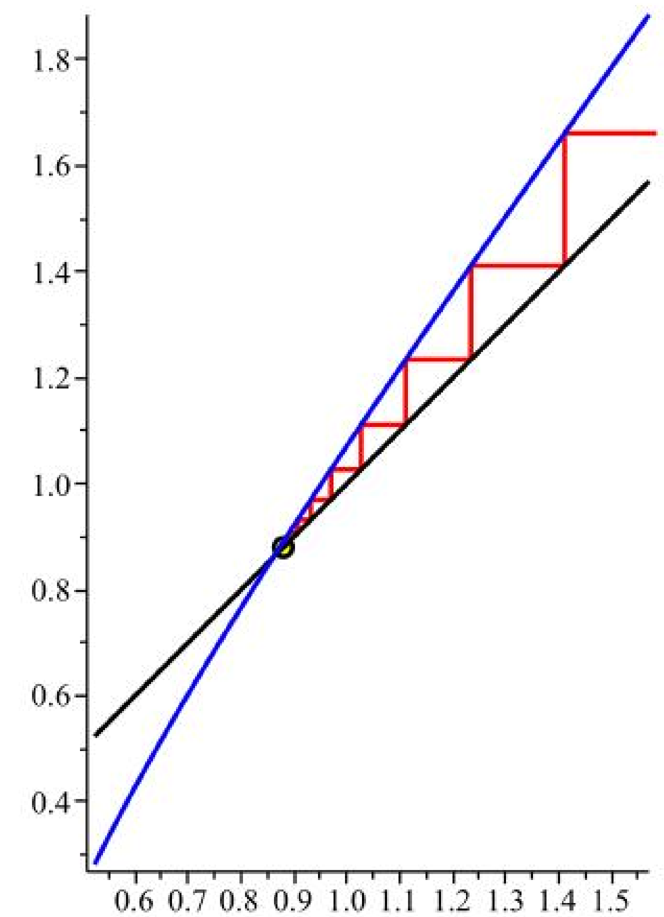
$$f(\tfrac{\pi}{4}) = \tfrac{\pi}{4} - 1 < 0, \quad f(\tfrac{\pi}{2}) = \tfrac{\pi}{2} > 0 \quad \Longrightarrow \quad \overline{x} \in (\tfrac{\pi}{4}, \tfrac{\pi}{2})$$

Zvolíme $\varphi(x) = \lambda f(x) + x$*, kde* $\lambda \neq 0$*; podmínka ukončení pro* $\eta = 0.001$.
Vyzkoušíme $\lambda \in \{-0.2, 0.2, -0.65, -0.8\}$.

Příklad použití MPI

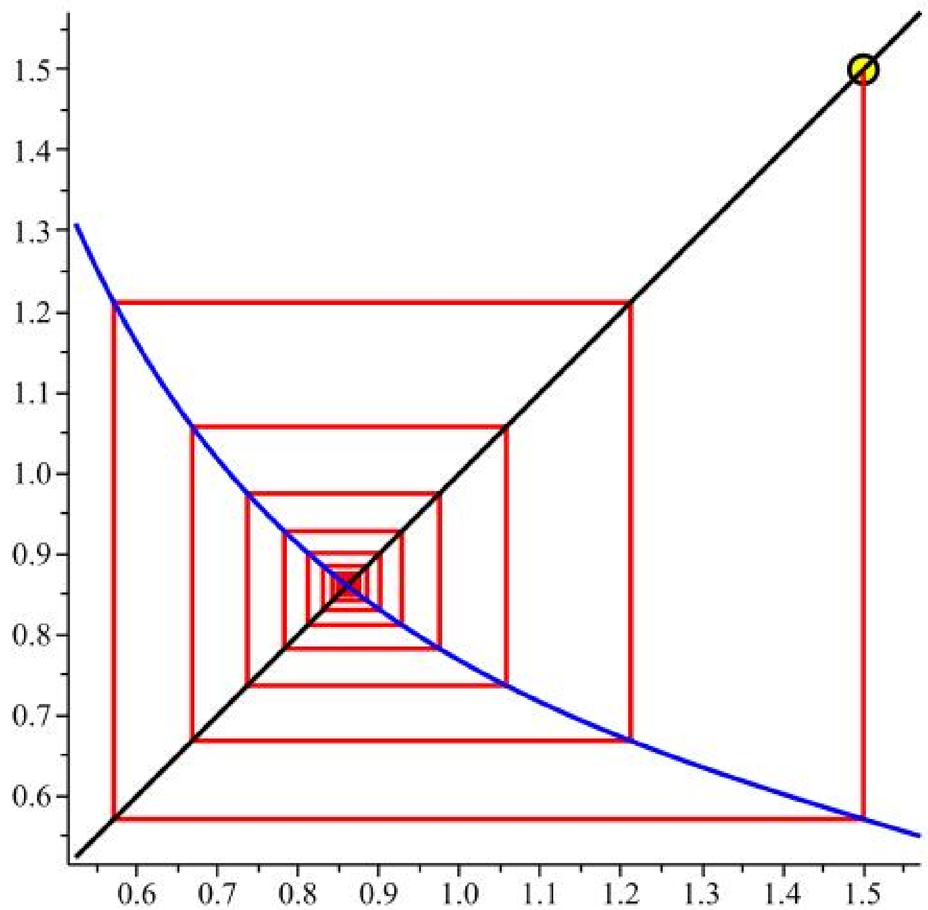


$x_{i+1} = 0.8\, x_i + 0.2 \operatorname{cotg} x_i,$
$x_0 = 1.5,$
konverguje monotónně

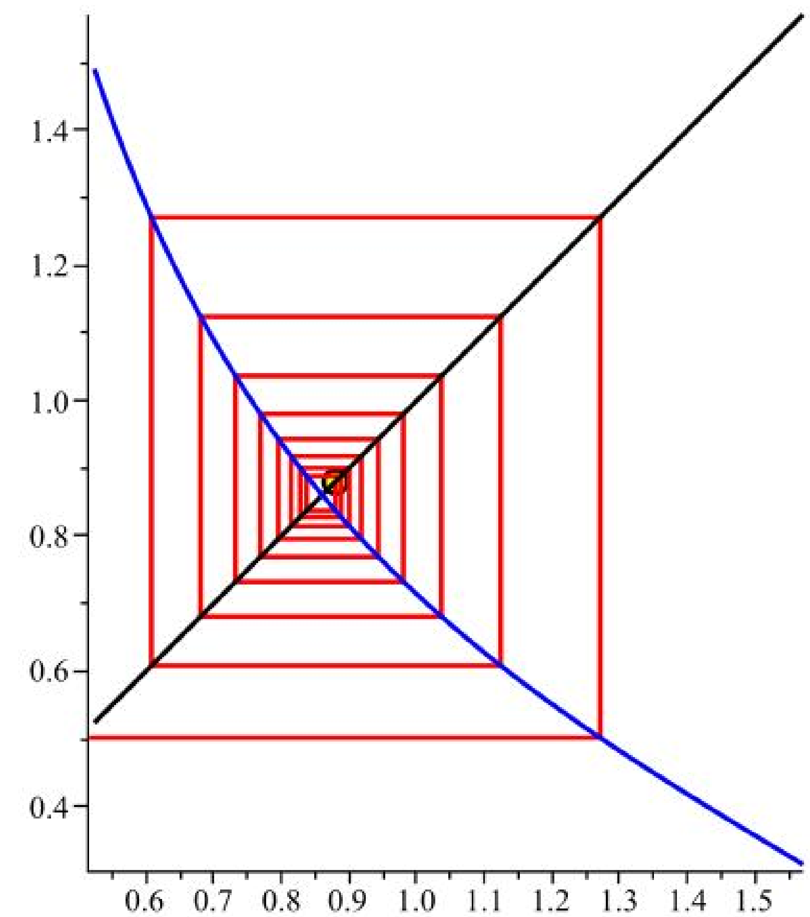


$x_{i+1} = 1.2\, x_i - 0.2 \operatorname{cotg} x_i,$
$x_0 = 0.88$
diverguje monotónně

Příklad použití MPI

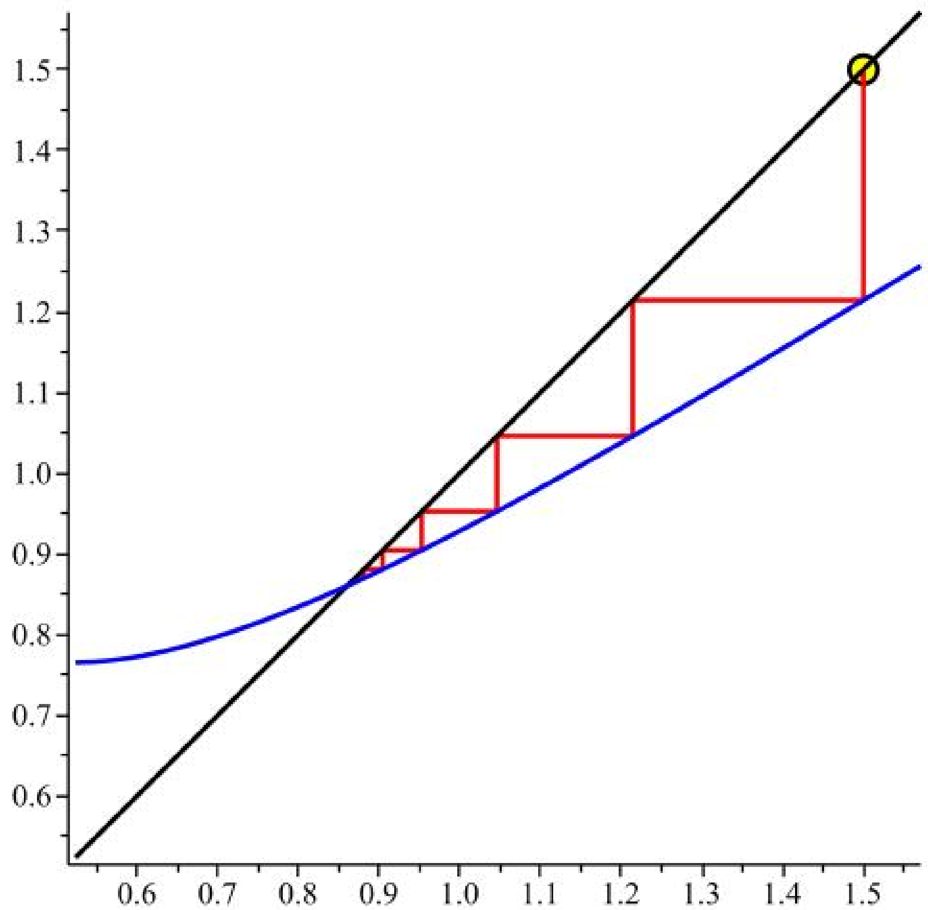


$x_{i+1} = 0.35\, x_i + 0.65 \operatorname{cotg} x_i,$
$x_0 = 1.5,$
konverguje nemonotónně

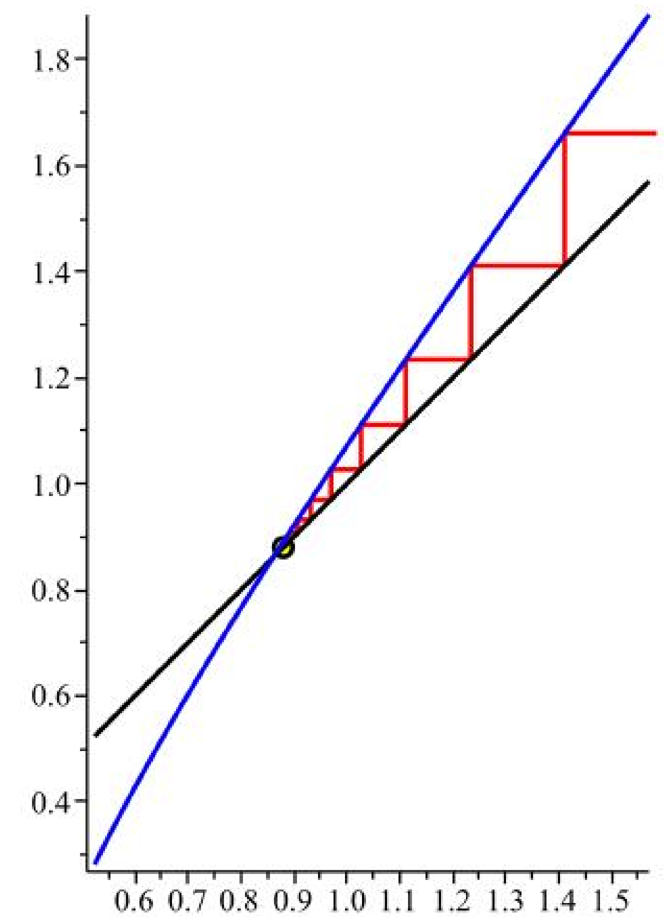


$x_{i+1} = 0.2\, x_i + 0.8 \operatorname{cotg} x_i,$
$x_0 = 0.88,$
diverguje nemonotónně

Příklad použití MPI

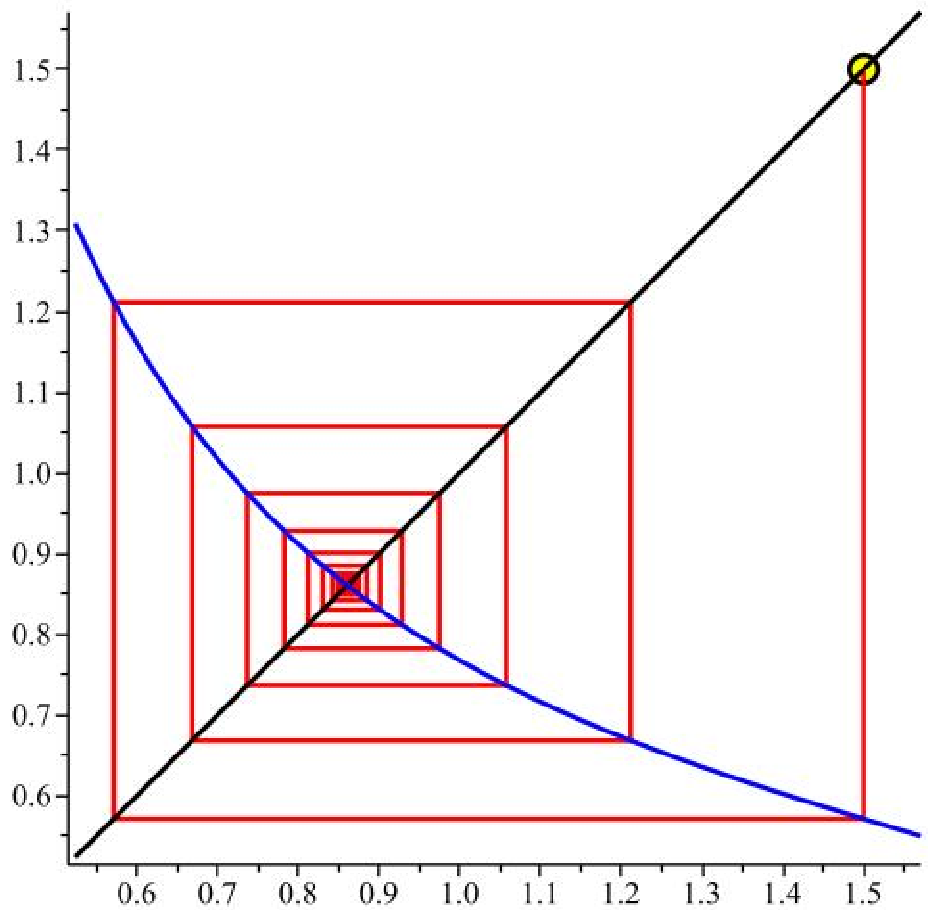


$x_{i+1} = 0.8\, x_i + 0.2 \operatorname{cotg} x_i,$
$x_0 = 1.5,$
konverguje monotónně

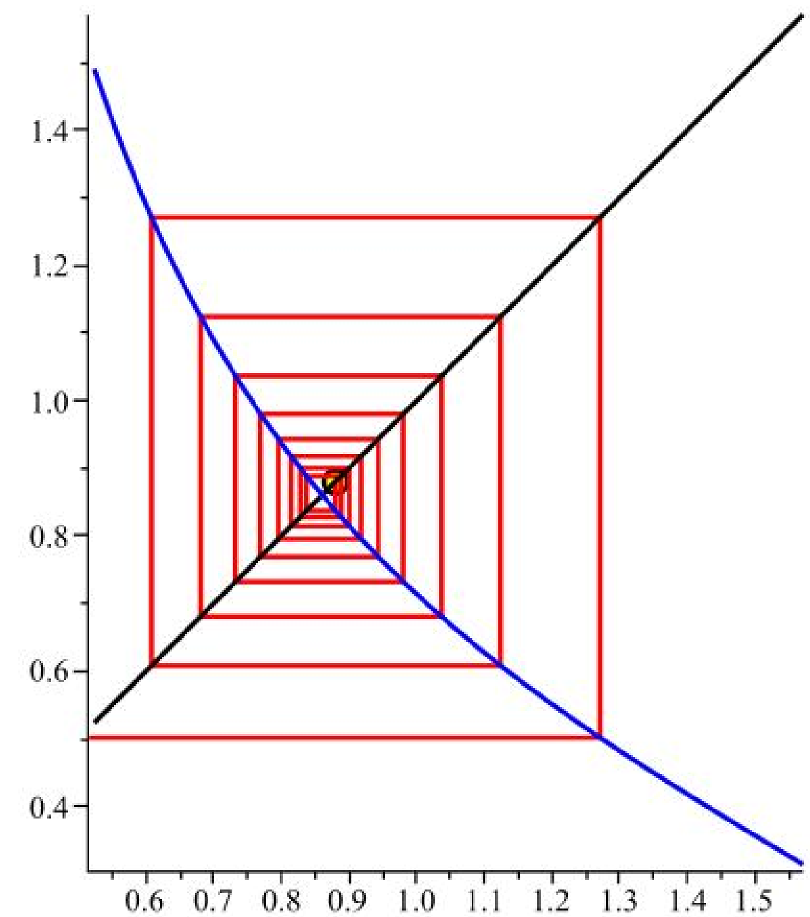


$x_{i+1} = 1.2\, x_i - 0.2 \operatorname{cotg} x_i,$
$x_0 = 0.88$
diverguje monotónně

Příklad použití MPI



$x_{i+1} = 0.35\, x_i + 0.65 \operatorname{cotg} x_i,$
$x_0 = 1.5,$
konverguje nemonotónně



$x_{i+1} = 0.2\, x_i + 0.8 \operatorname{cotg} x_i,$
$x_0 = 0.88,$
diverguje nemonotónně

4.9.1 Kontraktivní funkce

Definice 4.1 *Řekneme, že funkce φ je na intervalu I* **kontraktivní** *(s koeficientem q), jestliže*

$$\exists q < 1 \; \forall u, v \in I : |\varphi(u) - \varphi(v)| \leq q \cdot |u - v| \, .$$

kontraktivita $\Longrightarrow$ spojitost

4.9.1 Kontraktivní funkce

Definice 4.1 *Řekneme, že funkce φ je na intervalu I* **kontraktivní** *(s koeficientem q), jestliže*

$$\exists q < 1 \; \forall u, v \in I : |\varphi(u) - \varphi(v)| \leq q \cdot |u - v| \, .$$

kontraktivita $\implies$ spojitost

4.9.1 Kontraktivní funkce

Definice 4.1 *Řekneme, že funkce φ je na intervalu I* **kontraktivní** *(s koeficientem q), jestliže*

$$\exists q < 1 \; \forall u, v \in I : |\varphi(u) - \varphi(v)| \leq q \cdot |u - v| \, .$$

kontraktivita $\implies$ spojitost

Věta 4.8 (Postačující podmínka pro kontraktivitu) *Nechť funkce φ má na intervalu I spojitou derivaci a existuje $q < 1$ takové, že*

$$\forall x \in I : |\varphi'(x)| \leq q \, .$$

Pak φ je na I kontraktivní s koeficientem q.

4.9.1 Kontraktivní funkce

Definice 4.1 *Řekneme, že funkce φ je na intervalu I* **kontraktivní** *(s koeficientem q), jestliže*

$$\exists q < 1 \ \forall u, v \in I : |\varphi(u) - \varphi(v)| \leq q \cdot |u - v| \,.$$

kontraktivita $\implies$ spojitost

Věta 4.8 (Postačující podmínka pro kontraktivitu) *Nechť funkce φ má na intervalu I spojitou derivaci a existuje $q < 1$ takové, že*

$$\forall x \in I : |\varphi'(x)| \leq q \,.$$

Pak φ je na I kontraktivní s koeficientem q.

Důkaz. $|\varphi(u) - \varphi(v)| = \left| \int_v^u \varphi'(x) \, \mathrm{d}x \right| \leq \int_v^u |\varphi'(x)| \, \mathrm{d}x \leq \int_v^u q \, \mathrm{d}x = q \cdot |u - v| \,.$ $\square$

4.9.2 Věta o pevném bodě

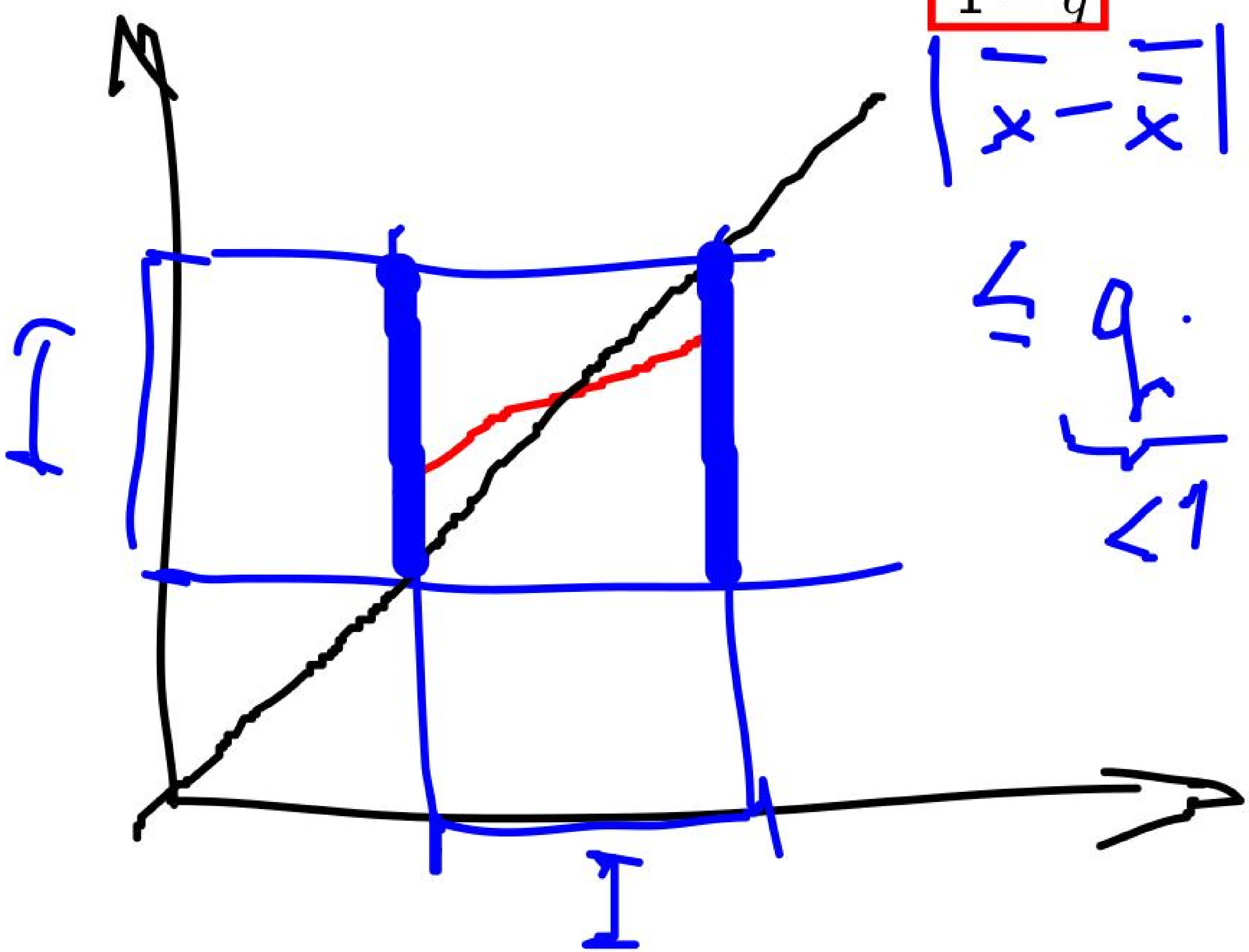
Věta 4.9 (Banachova věta o pevném bodě pro reálné funkce) *Nechť φ je funkce kontraktivní s koeficientem $q < 1$ na nějakém uzavřeném intervalu $I = \langle a, b\rangle$ taková, že zobrazuje I do I. Pak rovnice $\varphi(x) = x$ má v intervalu I právě jedno řešení $\overline{x}$. To dostaneme MPI s libovolnou počáteční hodnotou $x_0 \in I$. Odhad chyby:*

$$|\overline{x} - x_i| \leq \frac{q}{1-q} |x_i - x_{i-1}| .$$

4.9.2 Věta o pevném bodě

Věta 4.9 (Banachova věta o pevném bodě pro reálné funkce) *Nechť φ je funkce kontraktivní s koeficientem $q < 1$ na nějakém uzavřeném intervalu $I = \langle a, b \rangle$ taková, že zobrazuje I do I. Pak rovnice $\varphi(x) = x$ má v intervalu I právě jedno řešení $\overline{x}$. To dostaneme MPI s libovolnou počáteční hodnotou $x_0 \in I$. Odhad chyby:*

$$|\overline{x} - x_i| \le \boxed{\frac{q}{1-q}} |x_i - x_{i-1}| \,.$$



$$|\bar{x} - \bar{\bar{x}}| = |\varphi(\bar{x}) - \varphi(\bar{\bar{x}})| \le$$

$$\le \underbrace{q}_{<1} \cdot \underbrace{|\bar{x} - \bar{\bar{x}}|}_{0}$$

4.9.2 Věta o pevném bodě

Věta 4.9 (Banachova věta o pevném bodě pro reálné funkce) *Nechť φ je funkce kontraktivní s koeficientem $q < 1$ na nějakém uzavřeném intervalu $I = \langle a, b\rangle$ taková, že zobrazuje I do I. Pak rovnice $\varphi(x) = x$ má v intervalu I právě jedno řešení $\overline{x}$. To dostaneme MPI s libovolnou počáteční hodnotou $x_0 \in I$. Odhad chyby:*

$$|\overline{x} - x_i| \leq \frac{q}{1-q} |x_i - x_{i-1}| \, .$$

Důkaz.

- Existence řešení:

φ zobrazuje I do I
$\psi(x) = \varphi(x) - x$ je ψ v a nezáporná a v b nekladná; je spojitá, a tedy má v I nulový bod; ten je řešením rovnice $\varphi(x) = x$.

- Jednoznačnost řešení: Předpokládejme další řešení $\overline{\overline{x}} \in I$. Pak

$$|\overline{x} - \overline{\overline{x}}| = |\varphi(\overline{x}) - \varphi(\overline{\overline{x}})| \leq q \cdot |\overline{x} - \overline{\overline{x}}| \quad \Longrightarrow \quad \overline{\overline{x}} = \overline{x}$$

- Konvergence MPI k řešení:

$$|\overline{x} - x_i| = |\varphi(\overline{x}) - \varphi(x_{i-1})| \leq q \cdot |\overline{x} - x_{i-1}| \leq \ldots \leq q^i \cdot |\overline{x} - x_0| \to 0\,.$$

- Odhad chyby:

$$\begin{aligned} |\overline{x} - x_i| &\leq q \cdot |\overline{x} - x_{i-1}| = q \cdot |(\overline{x} - x_i) + (x_i - x_{i-1})| \\ &\leq q \cdot |\overline{x} - x_i| + q \cdot |x_i - x_{i-1}|\,, \end{aligned}$$

$$|\overline{x} - x_i| \leq \frac{q}{1-q}\,|x_i - x_{i-1}|\,.$$

$\square$

- Konvergence MPI k řešení:

$$|\overline{x} - x_i| = |\varphi(\overline{x}) - \varphi(x_{i-1})| \leq \underbrace{q}_{<1} \cdot |\overline{x} - x_{i-1}| \leq \ldots \leq q^i \cdot |\overline{x} - x_0| \to 0\,.$$

- Odhad chyby:

$$\begin{aligned} \boxed{|\overline{x} - x_i|} &\leq q \cdot |\overline{x} - x_{i-1}| = q \cdot |(\overline{x} - x_i) + (x_i - x_{i-1})| \\ &\leq q \cdot \boxed{|\overline{x} - x_i|} + q \cdot |x_i - x_{i-1}|\,, \end{aligned}$$

$$|\overline{x} - x_i| \leq \frac{q}{1-q}\, \underline{|x_i - x_{i-1}|}\,.$$

□

q	$\frac{q}{1-q}$
0.1	$\frac{0.1}{0.9} = \frac{1}{9}$
0.9	$\frac{0.9}{0.1} = 9$

- Konvergence MPI k řešení:

$$|\overline{x} - x_i| = |\varphi(\overline{x}) - \varphi(x_{i-1})| \leq q \cdot |\overline{x} - x_{i-1}| \leq \ldots \leq q^i \cdot |\overline{x} - x_0| \to 0\,.$$

- Odhad chyby:

$$\begin{aligned} |\overline{x} - x_i| &\leq q \cdot |\overline{x} - x_{i-1}| = q \cdot |(\overline{x} - x_i) + (x_i - x_{i-1})| \\ &\leq q \cdot |\overline{x} - x_i| + q \cdot |x_i - x_{i-1}|\,, \end{aligned}$$

$$|\overline{x} - x_i| \leq \frac{q}{1-q}\,|x_i - x_{i-1}|\,.$$

$\square$

Koeficient kontrakce q je horním odhadem

$$L(1) = \lim_{i\to\infty} \frac{|\overline{x} - x_i|}{|\overline{x} - x_{i-1}|} < 1\,,$$

což je kritérium konvergence použité u předchozích metod.

4.9.3 Optimalizace MPI

Jak rovnici $f(x) = 0$ převést na ekvivalentní tvar $\varphi(x) = x$ takový, že MPI rychle konverguje?

Možné řešení:

$$\varphi(x) = x + \lambda f(x)\,,$$

kde $\lambda \neq 0$ a

$$\varphi'(x) = 1 + \lambda f'(x)$$

je malá.

4.9.3 Optimalizace MPI

Jak rovnici $f(x) = 0$ převést na ekvivalentní tvar $\varphi(x) = x$ takový, že MPI rychle konverguje?

Možné řešení:

$$\varphi(x) = x + \lambda f(x)\,,$$

kde $\lambda \neq 0$ a

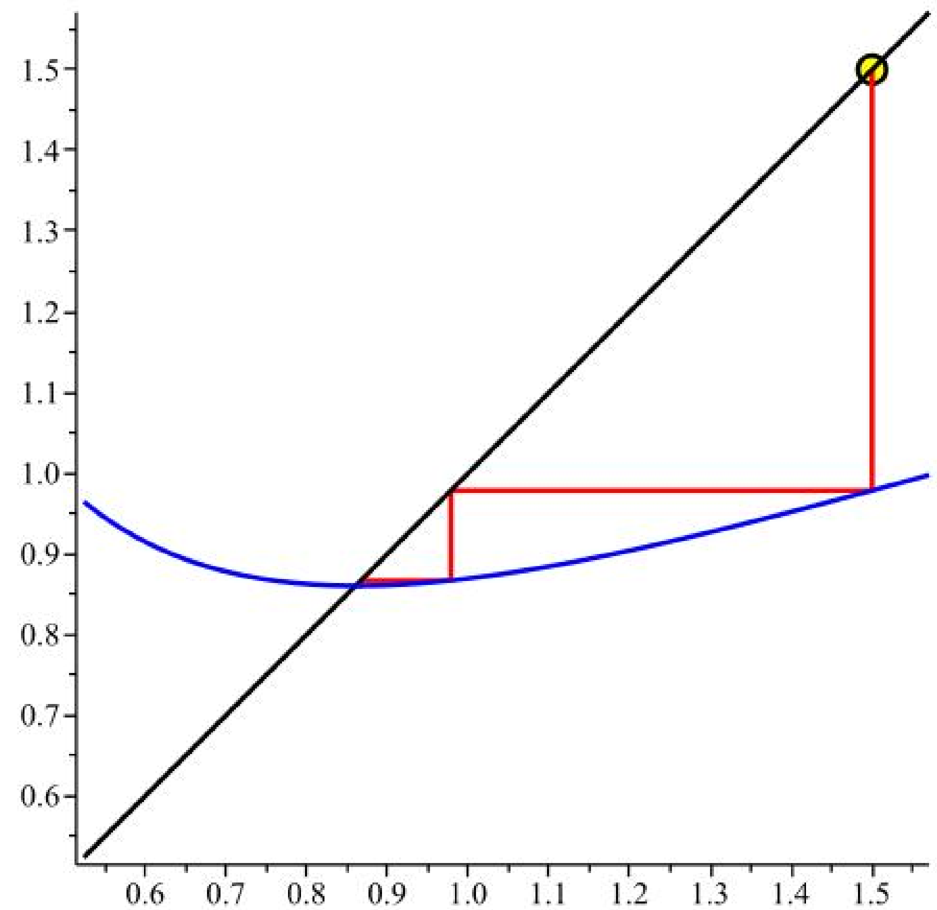
$$\varphi'(x) = 1 + \lambda f'(x)$$

je malá.

Příklad 4.2 (pokračování Př. 4.1) $f'(x) = 2 + \mathrm{cotg}^2\, x \in \langle 2, 3\rangle \implies \lambda \in \langle -\frac{1}{2}, -\frac{1}{3}\rangle$, $-1/f'(0.86) \approx -0.365 \implies \lambda = -0.365$,

$$\varphi(x) = 0.635\, x + 0.365\, \mathrm{cotg}\, x\,.$$

Optimalizace MPI



$$x_{i+1} = 0.635\, x_i + 0.365 \operatorname{cotg} x_i,\ x_0 = 1.5$$

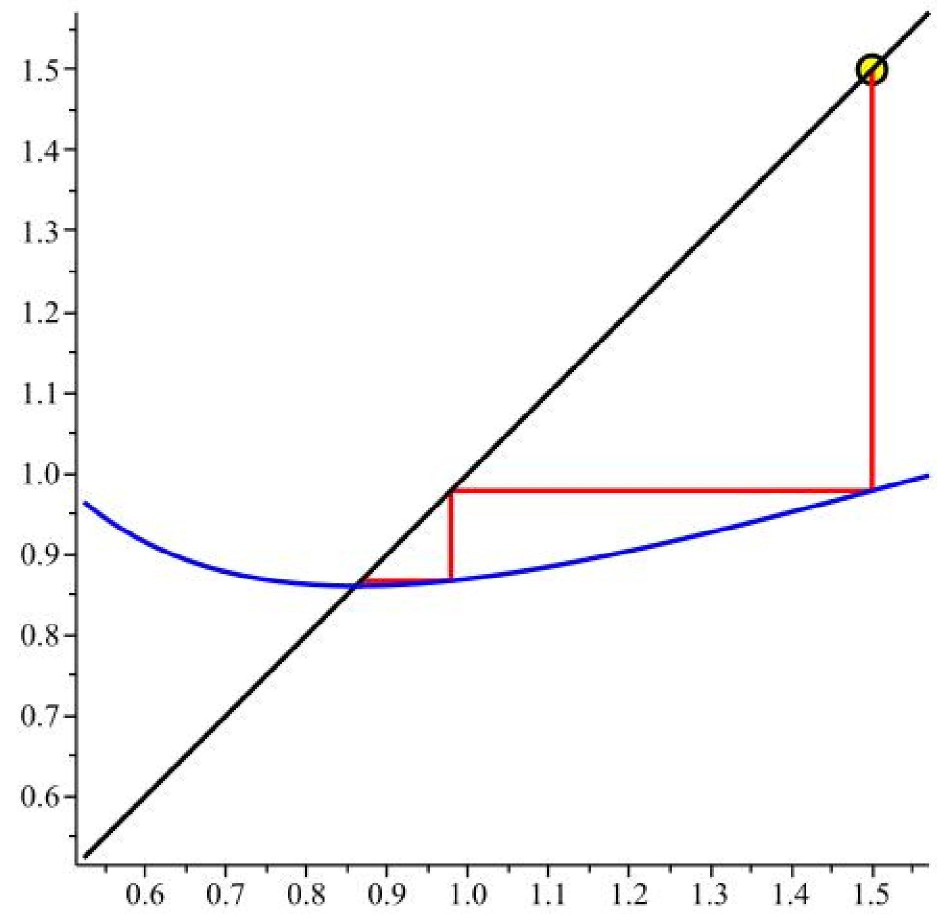
konverguje monotónně a rychle

4.9.4 Řád metody prosté iterace

Věta 4.10 *Nechť MPI konverguje k* $\overline{x}$. *Nechť* p *je nejmenší přirozené číslo, pro které* $\varphi^{(p)}(\overline{x}) \neq 0$, *a* $\varphi^{(p)}$ *je spojitá v nějakém okolí bodu* $\overline{x}$. *Pak řád metody je* p.

Důkaz. Taylorův rozvoj funkce φ se středem $\overline{x}$ vyhodnotíme v x_{i-1}:

Optimalizace MPI



$$x_{i+1} = 0.635\, x_i + 0.365 \operatorname{cotg} x_i, \; x_0 = 1.5$$

konverguje monotónně a rychle

4.9.3 Optimalizace MPI

Jak rovnici $f(x) = 0$ převést na ekvivalentní tvar $\varphi(x) = x$ takový, že MPI rychle konverguje?

Možné řešení:

$$\varphi(x) = x + \lambda f(x)\,,$$

kde $\lambda \neq 0$ a

$$\varphi'(x) = 1 + \lambda f'(x)$$

je malá.

Příklad 4.2 (pokračování Př. 4.1) $f'(x) = 2+\text{cotg}^2\, x \in \langle 2, 3\rangle \implies \lambda \in \langle -\frac{1}{2}, -\frac{1}{3}\rangle$, $-1/f'(0.86) \approx -0.365 \implies \lambda = -0.365$,

$$\varphi(x) = 0.635\, x + 0.365\, \text{cotg}\, x\,.$$

4.9.3 Optimalizace MPI

Jak rovnici $f(x) = 0$ převést na ekvivalentní tvar $\varphi(x) = x$ takový, že MPI rychle konverguje?

Možné řešení:

$$\varphi(x) = x + \lambda f(x)\,,$$

kde $\lambda \neq 0$ a

$$\varphi'(x) = 1 + \lambda f'(x)$$

je malá.

4.9.3 Optimalizace MPI

Jak rovnici $f(x) = 0$ převést na ekvivalentní tvar $\varphi(x) = x$ takový, že MPI rychle konverguje?

Možné řešení:

$$\varphi(x) = x + \lambda f(x)\,,$$

kde $\lambda \neq 0$ a

$$\varphi'(x) = 1 + \lambda f'(x)$$

je malá.

$$1 + \lambda_i f'(x_i) = 0$$

$$\lambda_i = \frac{-1}{f'(x_i)}$$

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

Newton. m.

4.9.3 Optimalizace MPI

Jak rovnici $f(x) = 0$ převést na ekvivalentní tvar $\varphi(x) = x$ takový, že MPI rychle konverguje?

Možné řešení:

$$\varphi(x) = x + \lambda f(x)\,,$$

kde $\lambda \neq 0$ a

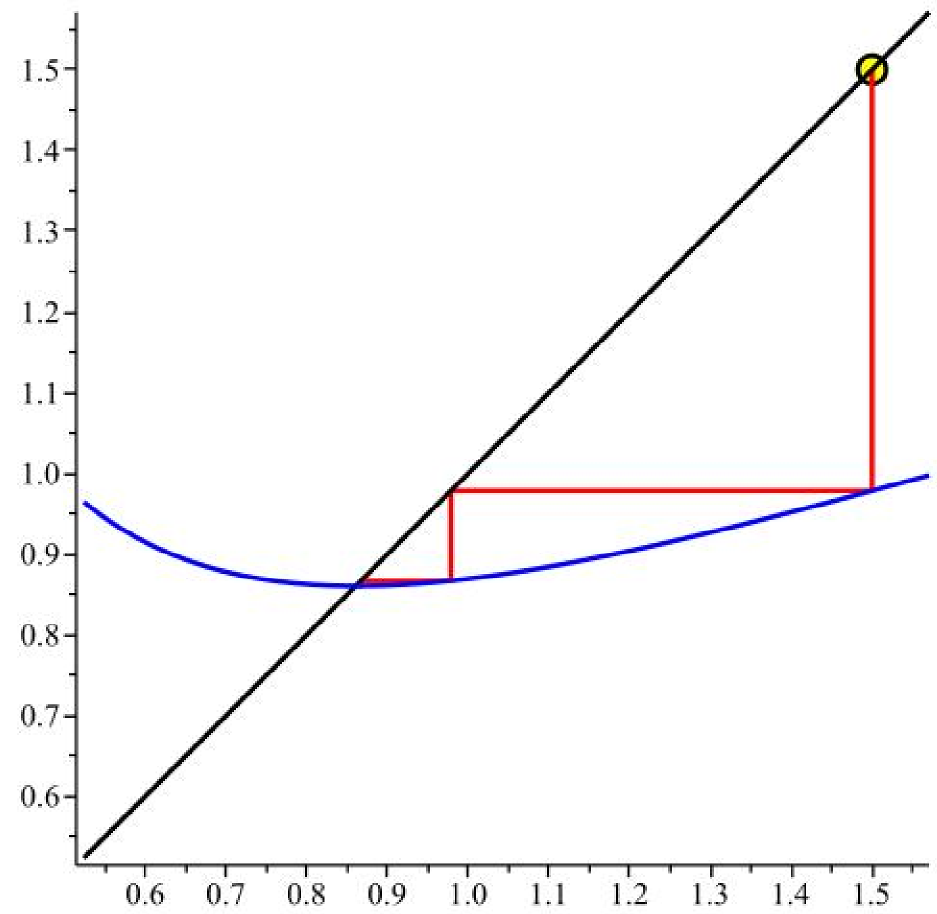
$$\varphi'(x) = 1 + \lambda f'(x)$$

je malá.

Příklad 4.2 (pokračování Př. 4.1) $f'(x) = 2+\mathrm{cotg}^2\, x \in \langle 2, 3\rangle \implies \lambda \in \langle -\frac{1}{2}, -\frac{1}{3}\rangle$, $-1/f'(0.86) \approx -0.365 \implies \lambda = -0.365$,

$$\varphi(x) = 0.635\, x + 0.365\, \mathrm{cotg}\, x\,.$$

Optimalizace MPI



$$x_{i+1} = 0.635\, x_i + 0.365 \operatorname{cotg} x_i,\ x_0 = 1.5$$

konverguje monotónně a rychle

4.9.4 Řád metody prosté iterace

Věta 4.10 *Nechť MPI konverguje k* $\overline{x}$. *Nechť* p *je nejmenší přirozené číslo, pro které* $\varphi^{(p)}(\overline{x}) \neq 0$, *a* $\varphi^{(p)}$ *je spojitá v nějakém okolí bodu* $\overline{x}$. *Pak řád metody je* p.

Důkaz. Taylorův rozvoj funkce φ se středem $\overline{x}$ vyhodnotíme v x_{i-1}:

4.9.4 Řád metody prosté iterace

Věta 4.10 *Nechť MPI konverguje k $\overline{x}$. Nechť p je nejmenší přirozené číslo, pro které $\varphi^{(p)}(\overline{x}) \neq 0$, a $\varphi^{(p)}$ je spojitá v nějakém okolí bodu $\overline{x}$. Pak řád metody je p.*

Důkaz. Taylorův rozvoj funkce φ se středem $\overline{x}$ vyhodnotíme v x_{i-1}:

$$\underbrace{\varphi(x_{i-1})}_{x_i} = \underbrace{\varphi(\overline{x})}_{\overline{x}} + \frac{1}{p!}(x_{i-1} - \overline{x})^p \, \varphi^{(p)}(\xi_{i-1}), \text{ kde } \xi_{i-1} \in I(\overline{x}, x_{i-1}),$$

$$\frac{\ln|\overline{x} - x_i|}{\ln|\overline{x} - x_{i-1}|} = \frac{\ln\left|\frac{1}{p!}(x_{i-1} - \overline{x})^p \, \varphi^{(p)}(\xi_{i-1})\right|}{\ln|\overline{x} - x_{i-1}|} =$$

$$= \frac{p \, \ln|\overline{x} - x_{i-1}|}{\ln|\overline{x} - x_{i-1}|} - \underbrace{\frac{\ln p!}{\ln|\overline{x} - x_{i-1}|}}_{\to 0} + \underbrace{\frac{\ln|\overbrace{\varphi^{(p)}(\xi_{i-1})}^{\to \varphi^{(p)}(\overline{x}) \neq 0}|}{\ln|\overline{x} - x_{i-1}|}}_{\to 0} \to p\,.$$

□

4.9.4 Řád metody prosté iterace

Věta 4.10 *Nechť MPI konverguje k* $\overline{x}$*. Nechť* p *je nejmenší přirozené číslo, pro které* $\varphi^{(p)}(\overline{x}) \neq 0$*, a* $\varphi^{(p)}$ *je spojitá v nějakém okolí bodu* $\overline{x}$*. Pak řád metody je* p*.*

Důkaz. Taylorův rozvoj funkce φ se středem $\overline{x}$ vyhodnotíme v x_{i-1}:

$$\underbrace{\varphi(x_{i-1})}_{x_i} = \underbrace{\varphi(\overline{x})}_{\overline{x}} + \frac{1}{p!}\,(x_{i-1}-\overline{x})^p\,\varphi^{(p)}(\xi_{i-1})\,,\ \text{ kde } \xi_{i-1} \in I(\overline{x}, x_{i-1})\,,$$

$$\frac{\ln|\overline{x}-x_i|}{\ln|\overline{x}-x_{i-1}|} = \frac{\ln\left|\dfrac{1}{p!}\,(x_{i-1}-\overline{x})^p\,\varphi^{(p)}(\xi_{i-1})\right|}{\ln|\overline{x}-x_{i-1}|} =$$

$$= \frac{p\,\cancel{\ln|\overline{x}-x_{i-1}|}}{\cancel{\ln|\overline{x}-x_{i-1}|}} - \underbrace{\frac{\ln p!}{\ln|\overline{x}-x_{i-1}|}}_{\to 0} + \underbrace{\frac{\ln|\overbrace{\varphi^{(p)}(\xi_{i-1})}^{\to\varphi^{(p)}(\overline{x})\neq 0}|}{\ln|\overline{x}-x_{i-1}|}}_{\to 0} \to p\,.$$

$$\to -\infty \qquad\qquad \to -\infty$$

□

Poznámka 4.2 *Nejčastěji je* $\varphi'(\overline{x}) \neq 0$*, takže MPI je řádu* 1*. Nemusí to však být vždy, např. Newtonova metoda je speciálním případem MPI.*

Zrychlení konvergence MPI

Nápad: V každém kroku zvolíme jiný koeficient λ_i tak, aby $\varphi'(x_i) = 0$, tj.

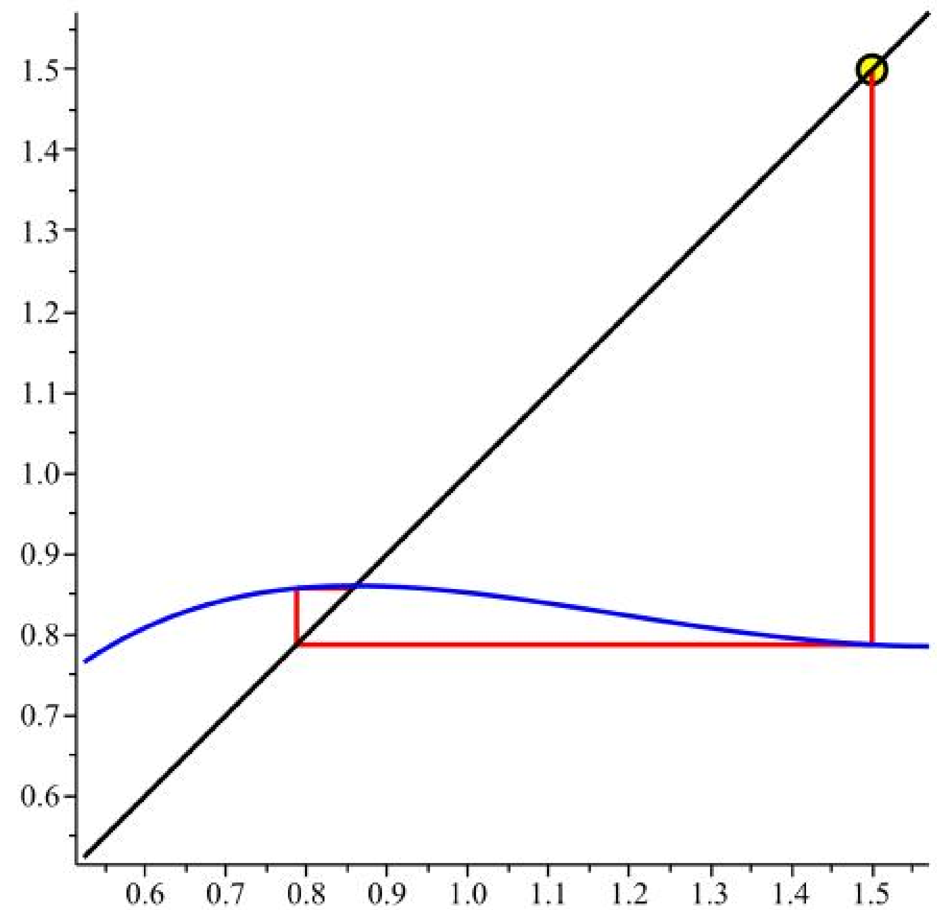
$$\lambda_i = -\frac{1}{f'(x_i)},$$

Dostaneme

$$x_{i+1} = x_i + \lambda_i f(x_i) = x_i - \frac{f(x_i)}{f'(x_i)},$$

což je Newtonova metoda (jako speciální případ MPI); ta je (obvykle) řádu 2, zatímco MPI (obvykle) řádu 1.

Newtonova metoda jako speciální případ MPI



$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)},\ x_0 = 1.5$$

konverguje nemonotónně a rychle

4.9.5 Kritéria pro výběr metody řešení rovnic

- **jednobodové**, např. MPI (která je v jistém smyslu univerzální jednobodovou metodou), Newtonova metoda,
- **dvoubodové**, např. bisekce, regula falsi, metoda sečen,
- **vícebodové**.

4.9.5 Kritéria pro výběr metody řešení rovnic

- **jednobodové**, např. MPI (která je v jistém smyslu univerzální jednobodovou metodou), Newtonova metoda,
- **dvoubodové**, např. bisekce, regula falsi, metoda sečen,
- **vícebodové**.

Z programátorského hlediska:

- **nevyžadující derivaci**, např. bisekce, regula falsi, metoda sečen a *obvykle* MPI (záleží na zvoleném iteračním vzorci),
- **vyžadující znalost první derivace**, např. Newtonova,
- **vyžadující znalost vyšších derivací**.

Kritéria pro výběr metody řešení rovnic 2

Podle konvergence dělíme metody řešení rovnic na

- **vždy konvergentní**, např. bisekce a regula falsi,
- **ostatní**, např. Newtonova, metoda sečen, MPI.

Kritéria pro výběr metody řešení rovnic 2

Podle konvergence dělíme metody řešení rovnic na

- **vždy konvergentní**, např. bisekce a regula falsi,
- **ostatní**, např. Newtonova, metoda sečen, (MPI.)

4.10 Podobné úlohy

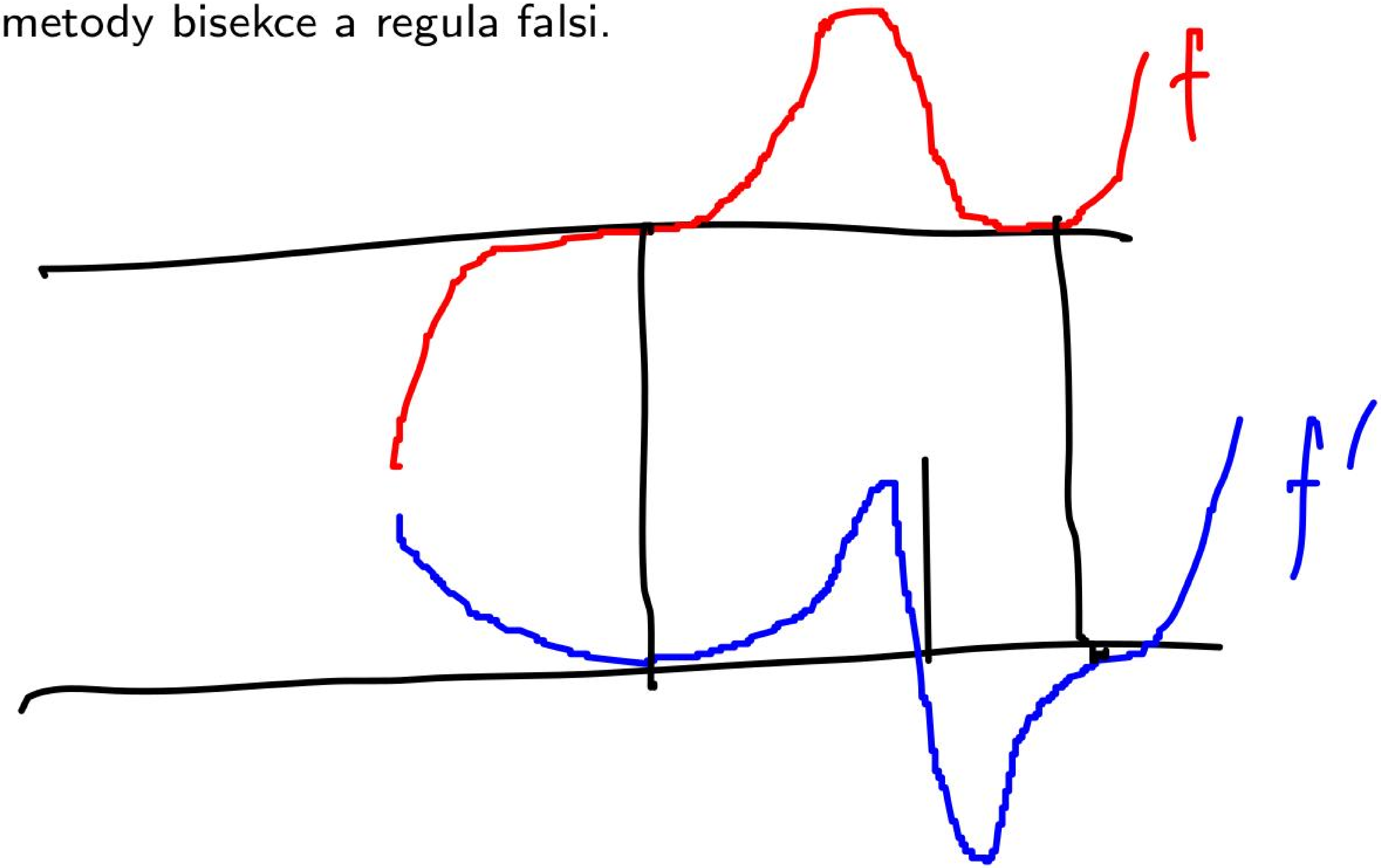
4.10.1 Hledání násobných kořenů

- V okolí kořene *sudé násobnosti* funkce nemění znaménko, takže nelze použít metody bisekce a regula falsi.

4.10 Podobné úlohy

4.10.1 Hledání násobných kořenů

- V okolí kořene *sudé násobnosti* funkce nemění znaménko, takže nelze použít metody bisekce a regula falsi.



4.10 Podobné úlohy

4.10.1 Hledání násobných kořenů

- V okolí kořene *sudé násobnosti* funkce nemění znaménko, takže nelze použít metody bisekce a regula falsi.

- Metoda sečen a Newtonova metoda jsou sice použitelné pro hledání násobných kořenů, ale jejich konvergence je pak prvního řádu.

4.10 Podobné úlohy

4.10.1 Hledání násobných kořenů

- V okolí kořene *sudé násobnosti* funkce nemění znaménko, takže nelze použít metody bisekce a regula falsi.

- Metoda sečen a Newtonova metoda jsou sice použitelné pro hledání násobných kořenů, ale jejich konvergence je pak prvního řádu.

- V metodě prosté iterace záleží pouze na použitém iteračním vzorci, nikoli na násobnosti kořene původní rovnice.

4.10 Podobné úlohy

4.10.1 Hledání násobných kořenů

• V okolí kořene *sudé násobnosti* funkce nemění znaménko, takže nelze použít metody bisekce a regula falsi.

• Metoda sečen a Newtonova metoda jsou sice použitelné pro hledání násobných kořenů, ale jejich konvergence je pak prvního řádu.

• V metodě prosté iterace záleží pouze na použitém iteračním vzorci, nikoli na násobnosti kořene původní rovnice.

1. metoda: Najdeme (všechny) kořeny funkce f' a vyzkoušíme, zda některý z nich je kořenem funkce f. Tam, kde má f kořen sudé násobnosti, má f' kořen liché násobnosti a mění znaménko.

Hledání násobných kořenů

2. metoda: Uvažujme funkci $h(x) = \frac{f(x)}{f'(x)}$ (kde „odstraníme odstranitelné nespojitosti“).

Tvrzení 4.2 *Nechť $\overline{x}$ je k-násobný kořen funkce f, v jehož okolí má f spojitou derivaci řádu k. Pak $\overline{x}$ je jednoduchým kořenem funkce $h = f/f'$.*

Hledání násobných kořenů

2. metoda: Uvažujme funkci $h(x) = \frac{f(x)}{f'(x)}$ (kde „odstraníme odstranitelné nespojitosti").

$$\square \cdot \frac{(x-\bar{x})^p}{(x-\bar{x})^{p-1}} = \square \cdot (x-\bar{x})$$

Tvrzení 4.2 *Nechť $\overline{x}$ je k-násobný kořen funkce f, v jehož okolí má f spojitou derivaci řádu k. Pak $\overline{x}$ je jednoduchým kořenem funkce $h = f/f'$.*

4.10 Podobné úlohy

4.10.1 Hledání násobných kořenů

- V okolí kořene *sudé násobnosti* funkce nemění znaménko, takže nelze použít metody bisekce a regula falsi.

- Metoda sečen a Newtonova metoda jsou sice použitelné pro hledání násobných kořenů, ale jejich konvergence je pak prvního řádu.

- V metodě prosté iterace záleží pouze na použitém iteračním vzorci, nikoli na násobnosti kořene původní rovnice.

1. metoda: Najdeme (všechny) kořeny funkce f' a vyzkoušíme, zda některý z nich je kořenem funkce f. Tam, kde má f kořen sudé násobnosti, má f' kořen liché násobnosti a mění znaménko.

4.10 Podobné úlohy

4.10.1 Hledání násobných kořenů

- V okolí kořene *sudé násobnosti* funkce nemění znaménko, takže nelze použít metody bisekce a regula falsi.

- Metoda sečen a Newtonova metoda jsou sice použitelné pro hledání násobných kořenů, ale jejich konvergence je pak prvního řádu.

- V metodě prosté iterace záleží pouze na použitém iteračním vzorci, nikoli na násobnosti kořene původní rovnice.

4.10 Podobné úlohy

4.10.1 Hledání násobných kořenů

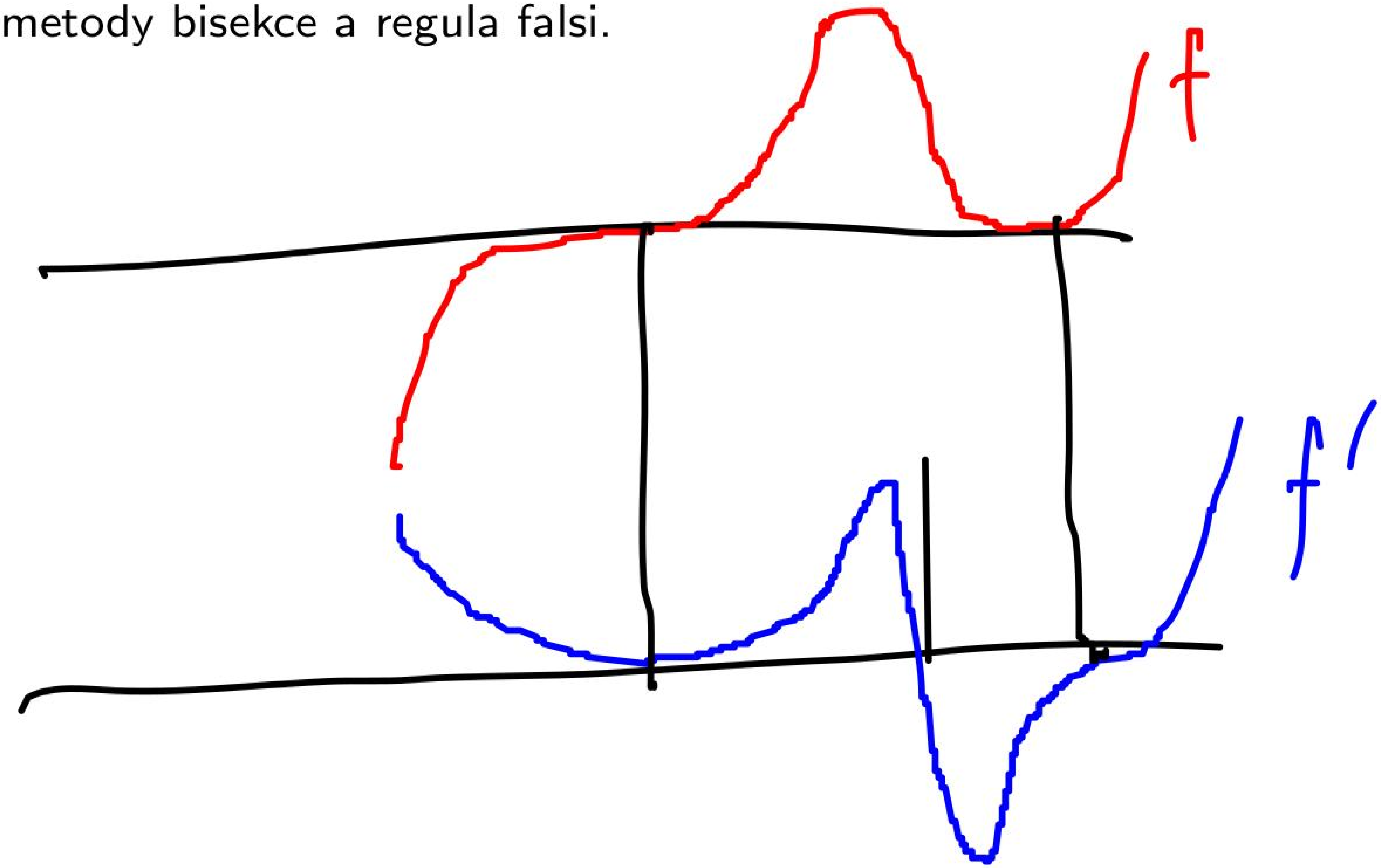
- V okolí kořene *sudé násobnosti* funkce nemění znaménko, takže nelze použít metody bisekce a regula falsi.

- Metoda sečen a Newtonova metoda jsou sice použitelné pro hledání násobných kořenů, ale jejich konvergence je pak prvního řádu.

4.10 Podobné úlohy

4.10.1 Hledání násobných kořenů

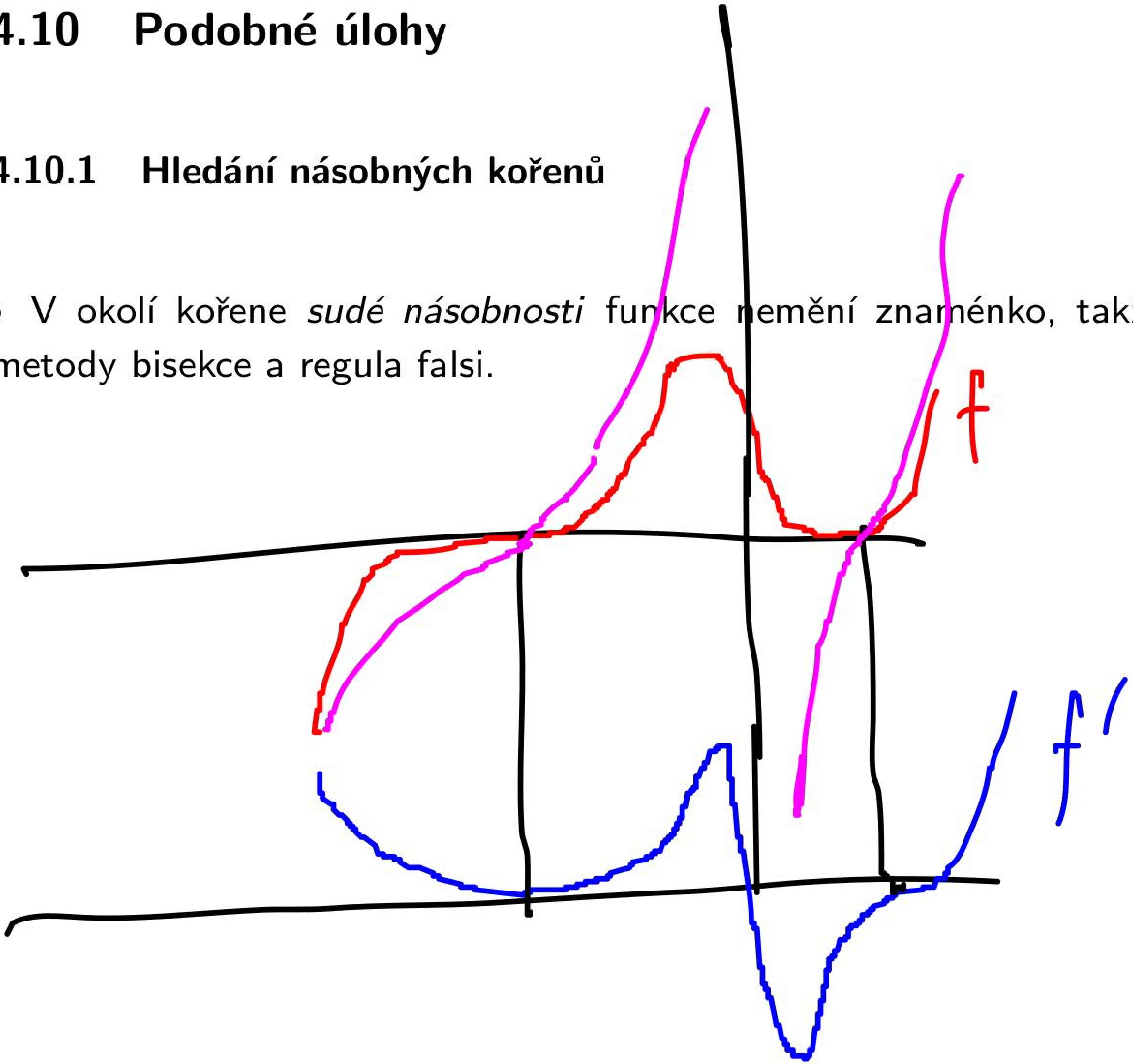
- V okolí kořene *sudé násobnosti* funkce nemění znaménko, takže nelze použít metody bisekce a regula falsi.



4.10 Podobné úlohy

4.10.1 Hledání násobných kořenů

- V okolí kořene *sudé násobnosti* funkce nemění znaménko, takže nelze použít metody bisekce a regula falsi.



4.10 Podobné úlohy

4.10.1 Hledání násobných kořenů

- V okolí kořene *sudé násobnosti* funkce nemění znaménko, takže nelze použít metody bisekce a regula falsi.

- Metoda sečen a Newtonova metoda jsou sice použitelné pro hledání násobných kořenů, ale jejich konvergence je pak prvního řádu.

4.10 Podobné úlohy

4.10.1 Hledání násobných kořenů

- V okolí kořene *sudé násobnosti* funkce nemění znaménko, takže nelze použít metody bisekce a regula falsi.

- Metoda sečen a Newtonova metoda jsou sice použitelné pro hledání násobných kořenů, ale jejich konvergence je pak prvního řádu.

- V metodě prosté iterace záleží pouze na použitém iteračním vzorci, nikoli na násobnosti kořene původní rovnice.

4.10 Podobné úlohy

4.10.1 Hledání násobných kořenů

- V okolí kořene *sudé násobnosti* funkce nemění znaménko, takže nelze použít metody bisekce a regula falsi.

- Metoda sečen a Newtonova metoda jsou sice použitelné pro hledání násobných kořenů, ale jejich konvergence je pak prvního řádu.

- V metodě prosté iterace záleží pouze na použitém iteračním vzorci, nikoli na násobnosti kořene původní rovnice.

1. metoda: Najdeme (všechny) kořeny funkce f' a vyzkoušíme, zda některý z nich je kořenem funkce f. Tam, kde má f kořen sudé násobnosti, má f' kořen liché násobnosti a mění znaménko.

Hledání násobných kořenů

2. metoda: Uvažujme funkci $h(x) = \frac{f(x)}{f'(x)}$ (kde „odstraníme odstranitelné nespojitosti“).

$$\Box \cdot \frac{(x-\bar{x})^p}{(x-\bar{x})^{p-1}} = \Box \cdot (x-\bar{x})$$

Tvrzení 4.2 *Nechť $\overline{x}$ je k-násobný kořen funkce f, v jehož okolí má f spojitou derivaci řádu k. Pak $\overline{x}$ je jednoduchým kořenem funkce $h = f/f'$.*

Hledání násobných kořenů

2. metoda: Uvažujme funkci $h(x) = \frac{f(x)}{f'(x)}$ (kde „odstraníme odstranitelné nespojitosti“).

$$\square \cdot \frac{(x-\bar{x})^p}{(x-\bar{x})^{p-1}} = \square \cdot (x-\bar{x})$$

Tvrzení 4.2 *Nechť $\overline{x}$ je k-násobný kořen funkce f, v jehož okolí má f spojitou derivaci řádu k. Pak $\overline{x}$ je jednoduchým kořenem funkce $h = f/f'$.*

Hledání násobných kořenů

2. metoda: Uvažujme funkci $h(x) = \frac{f(x)}{f'(x)}$ (kde „odstraníme odstranitelné nespojitosti“).

Tvrzení 4.2 *Nechť $\overline{x}$ je k-násobný kořen funkce f, v jehož okolí má f spojitou derivaci řádu k. Pak $\overline{x}$ je jednoduchým kořenem funkce $h = f/f'$.*

Důkaz. Definice k-násobného kořene říká, že $f^{(j)}(\overline{x}) = 0$ pro $j < k$ a $f^{(k)}(\overline{x}) \neq 0$. Opakovaným užitím l'Hospitalova pravidla odvodíme nenulovou limitu

$$\lim_{x\to\overline{x}} \frac{f(x)}{(x-\overline{x})^k} = \lim_{x\to\overline{x}} \frac{f'(x)}{k\,(x-\overline{x})^{k-1}} = \ldots = \lim_{x\to\overline{x}} \frac{f^{(k)}(x)}{k!} \neq 0\,.$$

Tedy podíl prvních dvou výrazů je definován a je jednotkový,

$$\lim_{x\to\overline{x}} \frac{f(x)}{(x-\overline{x})^k} \cdot \frac{k\,(x-\overline{x})^{k-1}}{f'(x)} = 1\,,$$

tím dostáváme pro funkci h limitu

$$\lim_{x\to\overline{x}} \frac{h(x)}{x-\overline{x}} = \frac{1}{k} \neq 0\,.$$

V poslední limitě konverguje jmenovatel k nule, musí k ní tedy konvergovat i čitatel, takže $\lim_{x\to\overline{x}} h(x) = 0$ a $\overline{x}$ je kořenem funkce h. Nenulová je podle l'Hospitalova pravidla též $\lim_{x\to\overline{x}} h'(x)$, takže $\overline{x}$ je jednoduchý kořen funkce h. $\square$

tím dostáváme pro funkci h limitu

$$\lim_{x \to \overline{x}} \frac{h(x)}{x - \overline{x}} = \frac{1}{k} \neq 0\,.$$

V poslední limitě konverguje jmenovatel k nule, musí k ní tedy konvergovat i čitatel, takže $\lim_{x\to\overline{x}} h(x) = 0$ a $\overline{x}$ je kořenem funkce h. Nenulová je podle l'Hospitalova pravidla též $\lim_{x\to\overline{x}} h'(x)$, takže $\overline{x}$ je jednoduchý kořen funkce h. $\square$

Pokud funkce f má pouze kořeny konečné násobnosti, pak funkce h má tytéž kořeny, ale jednoduché (není však spojitá).

4.10.2 Řešení algebraických rovnic neboli hledání kořenů polynomů

4.10.2 Řešení algebraických rovnic neboli hledání kořenů polynomů

Speciální případ rovnice $f(x) = 0$, kde f je polynom.

4.10.2 Řešení algebraických rovnic neboli hledání kořenů polynomů

Speciální případ rovnice $f(x) = 0$, kde f je polynom.

Věta 4.11 (Odhad polohy kořenů polynomu) *Všechny (komplexní) kořeny rovnice*

$$\sum_{i=0}^{n} a_i\, x^i = 0$$

mají absolutní hodnotu nejvýše

$$1 + \frac{\max\Big(|a_0|, \dots, |a_{n-1}|\Big)}{|a_n|}\,.$$

4.10.2 Řešení algebraických rovnic neboli hledání kořenů polynomů

Speciální případ rovnice $f(x) = 0$, kde f je polynom.

Věta 4.11 (Odhad polohy kořenů polynomu) *Všechny (komplexní) kořeny rovnice*

$$\sum_{i=0}^{n} a_i\, x^i = 0 \iff \sum_{i=0}^{n} \frac{a_i}{a_n} x^i = 0$$

mají absolutní hodnotu nejvýše

$$1 + \frac{\max\Big(|a_0|, \ldots, |a_{n-1}|\Big)}{|a_n|}.$$

4.10.3 Řešení rovnic v komplexním oboru

Metoda bisekce a metoda regula falsi jsou závislé na úplném uspořádání reálných čísel $\Longrightarrow$ ve větší dimenzi nepoužitelné.

4.10.3 Řešení rovnic v komplexním oboru

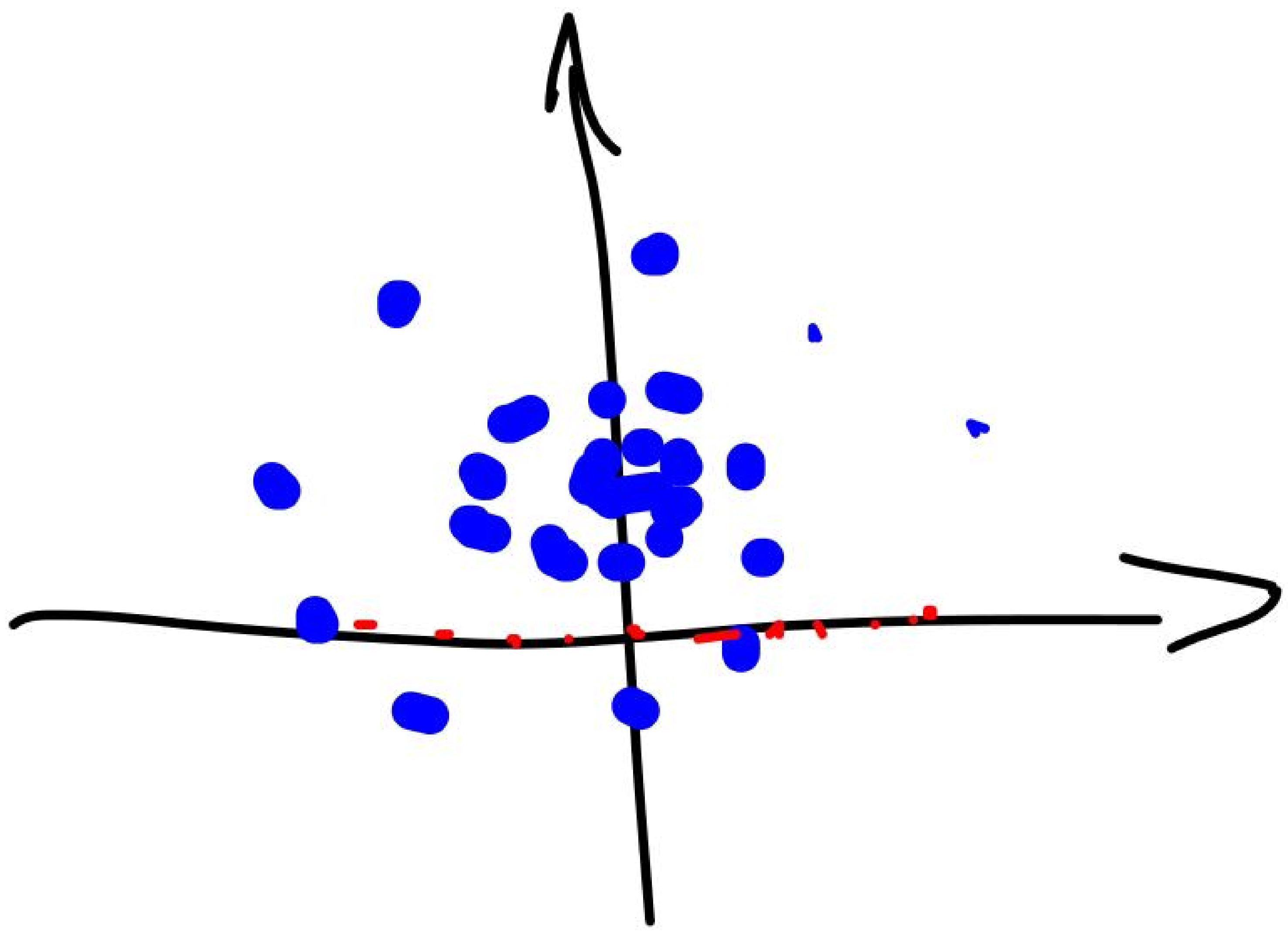
Metoda bisekce a metoda regula falsi jsou závislé na úplném uspořádání reálných čísel $\Longrightarrow$ ve větší dimenzi nepoužitelné.

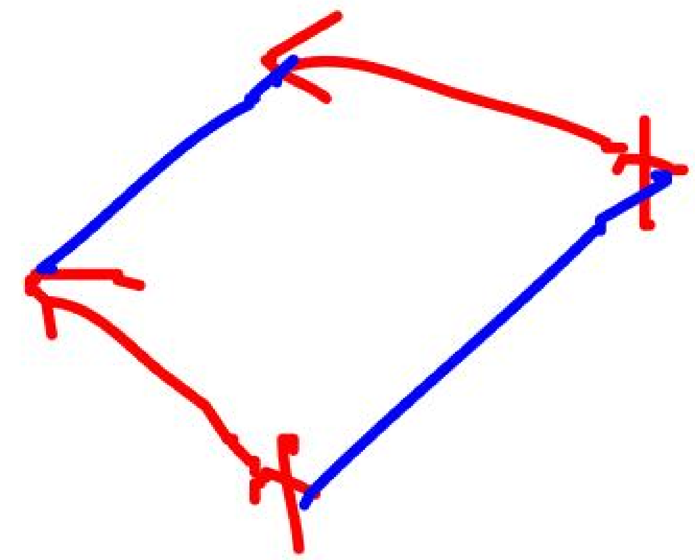
Metoda sečen a Newtonova metoda jsou použitelné pro komplexní kořeny.

4.10.3 Řešení rovnic v komplexním oboru

Metoda bisekce a metoda regula falsi jsou závislé na úplném uspořádání reálných čísel $\Longrightarrow$ ve větší dimenzi nepoužitelné.

Metoda sečen a Newtonova metoda jsou použitelné pro komplexní kořeny.

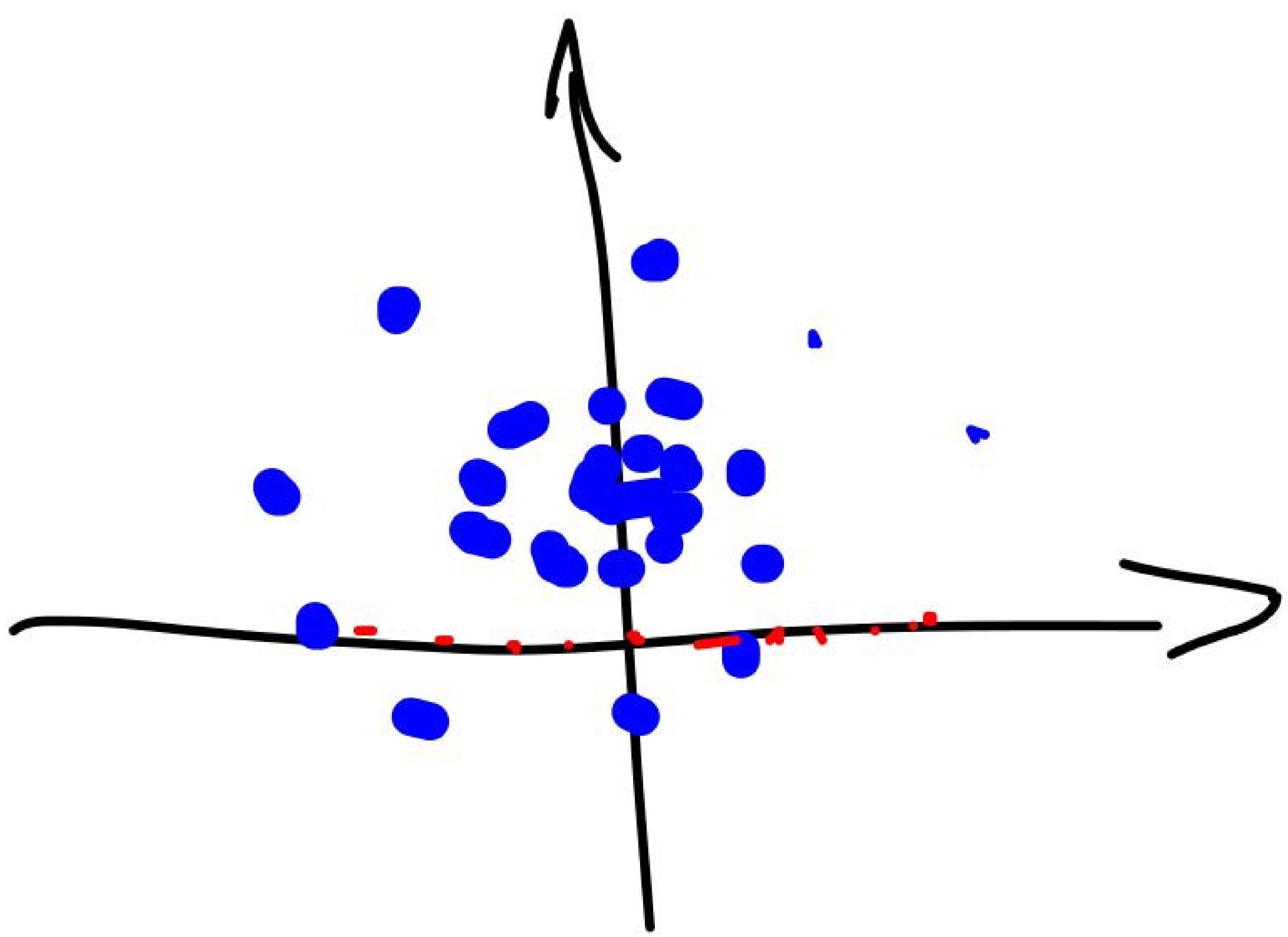


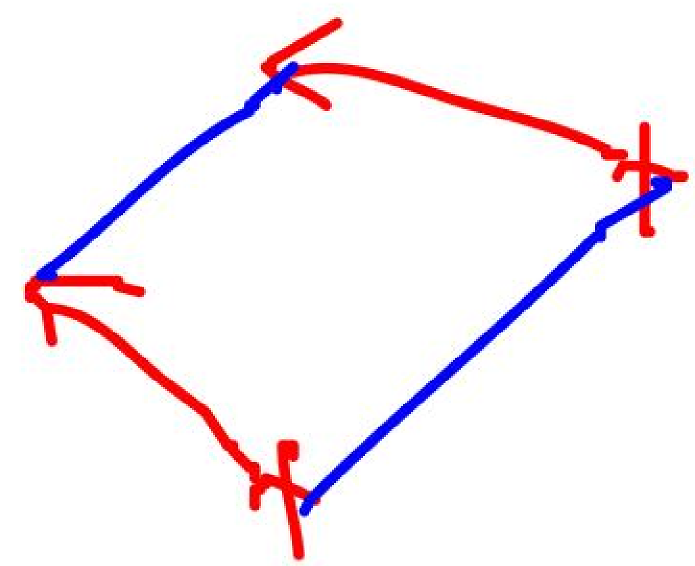


4.10.3 Řešení rovnic v komplexním oboru

Metoda bisekce a metoda regula falsi jsou závislé na úplném uspořádání reálných čísel $\Longrightarrow$ ve větší dimenzi nepoužitelné.

Metoda sečen a Newtonova metoda jsou použitelné pro komplexní kořeny.

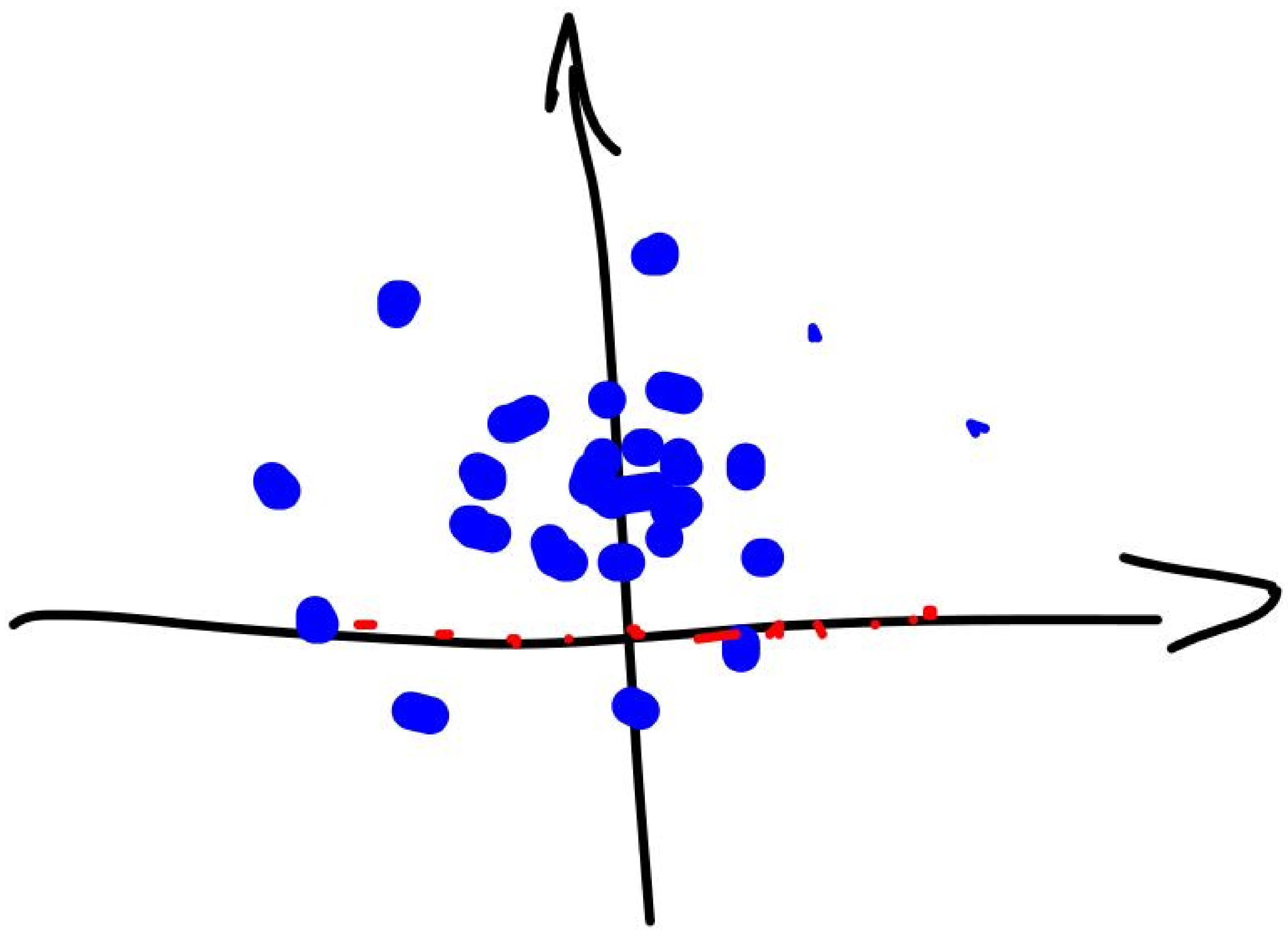


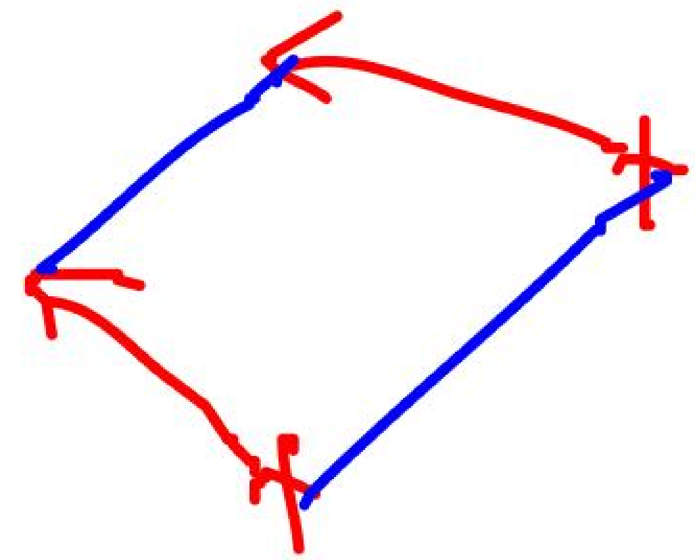


4.10.3 Řešení rovnic v komplexním oboru

Metoda bisekce a metoda regula falsi jsou závislé na úplném uspořádání reálných čísel $\Longrightarrow$ ve větší dimenzi nepoužitelné.

Metoda sečen a Newtonova metoda jsou použitelné pro komplexní kořeny.





4.10.3 Řešení rovnic v komplexním oboru

Metoda bisekce a metoda regula falsi jsou závislé na úplném uspořádání reálných čísel $\Longrightarrow$ ve větší dimenzi nepoužitelné.

Metoda sečen a Newtonova metoda jsou použitelné pro komplexní kořeny.

Pro nalezení komplexních kořenů může být nutný počáteční odhad s nenulovou imaginární částí.

4.10.4 Řešení soustav rovnic

Newtonova metoda má i zobecnění pro soustavy nelineárních rovnic; pak místo derivace pracujeme s jacobiánem a místo dělení jej potřebujeme invertovat, čímž se jednak zvyšuje složitost výpočtu, jednak vznikají problémy s body, v nichž je jacobián singulární. Podmínky konvergence jsou opět složitější než v reálném případě.

4.10.4 Řešení soustav rovnic

Newtonova metoda má i zobecnění pro soustavy nelineárních rovnic; pak místo derivace pracujeme s jacobiánem a místo dělení jej potřebujeme invertovat, čímž se jednak zvyšuje složitost výpočtu, jednak vznikají problémy s body, v nichž je jacobián singulární. Podmínky konvergence jsou opět složitější než v reálném případě.

4.10.4 Řešení soustav rovnic

Newtonova metoda má i zobecnění pro soustavy nelineárních rovnic; pak místo derivace pracujeme s jacobiánem a místo dělení jej potřebujeme invertovat, čímž se jednak zvyšuje složitost výpočtu, jednak vznikají problémy s body, v nichž je jacobián singulární. Podmínky konvergence jsou opět složitější než v reálném případě.

Metoda prosté iterace je použitelná i v prostorech větší (konečné) dimenze. Zajištění kontraktivity použitého zobrazení může být problém.

4.10.4 Řešení soustav rovnic

Newtonova metoda má i zobecnění pro soustavy nelineárních rovnic; pak místo derivace pracujeme s jacobiánem a místo dělení jej potřebujeme invertovat, čímž se jednak zvyšuje složitost výpočtu, jednak vznikají problémy s body, v nichž je jacobián singulární. Podmínky konvergence jsou opět složitější než v reálném případě.

Metoda prosté iterace je použitelná i v prostorech větší (konečné) dimenze. Zajištění kontraktivity použitého zobrazení může být problém.

Kvůli obtížím se zajištěním konvergence se pro řešení soustav rovnic často používají metody založené na jiných principech než v jednodimenzionálním případě.

Literatura

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

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

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

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

4.10.4 Řešení soustav rovnic

Newtonova metoda má i zobecnění pro soustavy nelineárních rovnic; pak místo derivace pracujeme s jacobiánem a místo dělení jej potřebujeme invertovat, čímž se jednak zvyšuje složitost výpočtu, jednak vznikají problémy s body, v nichž je jacobián singulární. Podmínky konvergence jsou opět složitější než v reálném případě.

Metoda prosté iterace je použitelná i v prostorech větší (konečné) dimenze. Zajištění kontraktivity použitého zobrazení může být problém.

Kvůli obtížím se zajištěním konvergence se pro řešení soustav rovnic často používají metody založené na jiných principech než v jednodimenzionálním případě.

Numerické metody — numerické řešení soustav lineárních rovnic

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

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

23. listopadu 2022

4 NUMERICKÉ ŘEŠENÍ SOUSTAV LINEÁRNÍCH ROVNIC

4.1 Formulace úlohy a její obtíže

Úloha: Hledáme řešení soustavy n lineárních rovnic o n neznámých $x_1, x_2, \ldots, x_n$

$$\begin{aligned} a_{1,1}\,x_1 + a_{1,2}\,x_2 + \cdots + a_{1,n}\,x_n &= b_1 \\ a_{2,1}\,x_1 + a_{2,2}\,x_2 + \cdots + a_{2,n}\,x_n &= b_2 \\ &\vdots \\ a_{n,1}\,x_1 + a_{n,2}\,x_2 + \cdots + a_{n,n}\,x_n &= b_n \end{aligned}$$

Maticový tvar:

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

kde $\mathbf{A} = (a_{i,j})_{i,j=1,\ldots,n}$ je (regulární) matice soustavy,
$\vec{b} = (b_1, b_2, \ldots, b_n)^\top$ vektor pravých stran,
$\vec{x} = (x_1, x_2, \ldots, x_n)^\top$ vektor neznámých.

4 NUMERICKÉ ŘEŠENÍ SOUSTAV LINEÁRNÍCH ROVNIC

4.1 Formulace úlohy a její obtíže

Úloha: Hledáme řešení soustavy n lineárních rovnic o n neznámých $x_1, x_2, \ldots, x_n$

$$
\begin{array}{rcl}
a_{1,1}\,x_1 + a_{1,2}\,x_2 + \cdots + a_{1,n}\,x_n & = & b_1 \\
a_{2,1}\,x_1 + a_{2,2}\,x_2 + \cdots + a_{2,n}\,x_n & = & b_2 \\
& \vdots & \\
a_{n,1}\,x_1 + a_{n,2}\,x_2 + \cdots + a_{n,n}\,x_n & = & b_n
\end{array}
$$

Maticový tvar:

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

kde $\mathbf{A} = (a_{i,j})_{i,j=1,\ldots,n}$ je $\boxed{\text{(regulární)}}$ matice soustavy,
$\vec{b} = (b_1, b_2, \ldots, b_n)^\top$ vektor pravých stran,
$\vec{x} = (x_1, x_2, \ldots, x_n)^\top$ vektor neznámých.

Cramerovo pravidlo má velkou výpočetní složitost a numerické chyby.

4.1.1 Druhy problémů

Matice soustavy:

- plné, ne příliš velké,
- řídké, často velmi velké (mj. u kubického splinu).

4.1.2 Špatná podmíněnost

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

Malá změna koeficientů soustavy nebo pravé strany může způsobit velkou změnu řešení.

Zpětné dosazení (nepřesného) řešení $\vec{x}_c$ dá **reziduum řešení**:

$$\vec{r} = \vec{b} - \mathbf{A}\,\vec{x}_c\,,$$

Pokud matice $\mathbf{A}^{-1}$ má velké prvky, může být reziduum $\vec{r}$ malé, i když se vektor $\vec{x}_c$ podstatně liší od přesného řešení $\vec{\vec{x}}$.

$$\begin{aligned} \vec{r} &= \mathbf{A}\,\vec{\vec{x}} - \mathbf{A}\,\vec{x}_c = \mathbf{A}\,(\vec{\vec{x}} - \vec{x}_c)\,, \\ \vec{\vec{x}} - \vec{x}_c &= \mathbf{A}^{-1}\,\vec{r}\,. \end{aligned}$$

Jsou-li prvky matice $\mathbf{A}^{-1}$ velké, může i malá složka vektoru $\vec{r}$ způsobit velký rozdíl $\vec{\vec{x}} - \vec{x}_c$.

Malé reziduum nezaručuje malou chybu řešení!
Takové soustavy nazýváme **špatně podmíněné**.

Příklad 4.1 *Soustava*

$$\begin{aligned} 2\,x + \qquad 6\,y &= 8 \\ 2\,x + 6.00001\,y &= 8.00001 \end{aligned}$$

má řešení $x = 1$, $y = 1$;

Příklad 4.1 *Soustava*

$$\begin{aligned} 2\,x + 6\,y &= 8 \\ 2\,x + 6.00001\,y &= 8.00001 \end{aligned}$$

má řešení $x = 1$, $y = 1$*;*

minimální změna koeficientů na soustavu

$$\begin{aligned} 2\,x + 6\,y &= 8 \\ 2\,x + 5.99999\,y &= 8.00002 \end{aligned}$$

změní řešení na $x = 10$, $y = -2$*.*

Inverzní matice k oběma soustavám mají prvky řádově 10^5*, což ukazuje na jejich špatnou podmíněnost.*
Rovnice v soustavách jsou „skoro lineárně závislé“.

4.1.3 Zdroje chyb

- nepřesnost koeficientů soustavy a pravé strany,
- zaokrouhlovací chyby při výpočtu,
- chyby metody – nekonečný proces je nahrazen konečným počtem kroků (u iteračních metod).

4.2 Přímé metody

Po konečném počtu kroků vedou (teoreticky) k přesnému řešení.

4.2.1 Gaussova eliminace (GEM)

Postupné úpravy matice soustavy pomocí ekvivalentních úprav (nemění řešení soustavy) na horní trojúhelníkovou matici, ze které lze zpětným dosazením snadno získat řešení.

Rozšířená matice soustavy má prvky

$$\begin{aligned} a_{i,j}^{(0)} &= a_{i,j}\,, \qquad \text{pro } i = 1,2,\ldots,n,\; j = 1,2,\ldots,n; \\ a_{i,n+1}^{(0)} &= b_i\,, \qquad \text{pro } i = 1,2,\ldots,n\,. \end{aligned}$$

Soustavu

$$\begin{aligned} a_{1,1}^{(0)} x_1 + a_{1,2}^{(0)} x_2 + \cdots + a_{1,n}^{(0)} x_n &= a_{1,n+1}^{(0)} \\ a_{2,1}^{(0)} x_1 + a_{2,2}^{(0)} x_2 + \cdots + a_{2,n}^{(0)} x_n &= a_{2,n+1}^{(0)} \\ &\vdots \\ a_{n,1}^{(0)} x_1 + a_{n,2}^{(0)} x_2 + \cdots + a_{n,n}^{(0)} x_n &= a_{n,n+1}^{(0)} \end{aligned}$$

převedeme povolenými úpravami na tvar

$$\begin{array}{rcl}
a_{1,1}^{(0)} x_1 + a_{1,2}^{(0)} x_2 + \cdots + a_{1,n}^{(0)} x_n & = & a_{1,n+1}^{(0)} \\
a_{2,2}^{(1)} x_2 + \cdots + a_{2,n}^{(1)} x_n & = & a_{2,n+1}^{(1)} \\
& \vdots & \\
a_{n,n}^{(n-1)} x_n & = & a_{n,n+1}^{(n-1)} ,
\end{array}$$

ze kterého zpětnou substitucí vypočítáme vektor řešení.

převedeme povolenými úpravami na tvar

$$\begin{aligned}
a_{1,1}^{(0)} x_1 + a_{1,2}^{(0)} x_2 + \cdots + a_{1,n}^{(0)} x_n &= a_{1,n+1}^{(0)} \\
a_{2,2}^{(1)} x_2 + \cdots + a_{2,n}^{(1)} x_n &= a_{2,n+1}^{(1)} \\
&\vdots \\
a_{n,n}^{(n-1)} x_n &= a_{n,n+1}^{(n-1)},
\end{aligned}$$

ze kterého zpětnou substitucí vypočítáme vektor řešení.

Pokud vyjde na diagonále nulový prvek, stačí provést záměnu řádků (resp. sloupců – v tom případě musíme **zaměnit i odpovídající složky vektoru řešení!**).
To lze, pokud je matice soustavy regulární.

Algoritmus 4.1 *Pro* $k = 1, 2, \ldots, n-1$, *pro* $i = k+1, k+2, \ldots, n$, $j = k+1, k+2, \ldots, n+1$

$$a_{i,j}^{(k)} = a_{i,j}^{(k-1)} - \frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}} a_{k,j}^{(k-1)}.$$

Pokud po provedení **přímého chodu** je nějaký diagonální prvek $a_{i,i}^{(i-1)} = 0$ (resp. $|a_{i,i}^{(i-1)}| < \varepsilon$), matice soustavy je (resp. může být) singulární.

V opačném případě použijeme **zpětnou substituci**

$$x_i = \frac{1}{a_{i,i}^{(i-1)}} \Big(a_{i,n+1}^{(i-1)} - \sum_{j=i+1}^{n} a_{i,j}^{(i-1)} \, x_j \Big), \quad \text{pro } i = n, n-1, \ldots, 1 \, .$$

4.2.2 Výběr hlavního prvku

Pokud číslo na diagonále je v absolutní hodnotě malé, jeho malá změna vyvolá velkou změnu výsledku při dělení a rostou zaokrouhlovací chyby.

Pokud po provedení **přímého chodu** je nějaký diagonální prvek $a_{i,i}^{(i-1)} = 0$ (resp. $|a_{i,i}^{(i-1)}| < \varepsilon$), matice soustavy je (resp. může být) singulární.

V opačném případě použijeme **zpětnou substituci**

$$x_i = \frac{1}{\boxed{a_{i,i}^{(i-1)}}} \left(a_{i,n+1}^{(i-1)} - \sum_{j=i+1}^{n} a_{i,j}^{(i-1)} x_j \right), \quad \text{pro } i = n, n-1, \ldots, 1\,.$$

4.2.2 Výběr hlavního prvku

Pokud číslo na diagonále je v absolutní hodnotě malé, jeho malá změna vyvolá velkou změnu výsledku při dělení a rostou zaokrouhlovací chyby.

Pokud po provedení **přímého chodu** je nějaký diagonální prvek $a_{i,i}^{(i-1)} = 0$ (resp. $|a_{i,i}^{(i-1)}| < \varepsilon$), matice soustavy je (resp. může být) singulární.

V opačném případě použijeme **zpětnou substituci**

$$x_i = \frac{1}{a_{i,i}^{(i-1)}} \left(a_{i,n+1}^{(i-1)} - \sum_{j=i+1}^{n} a_{i,j}^{(i-1)} x_j \right), \quad \text{pro } i = n, n-1, \ldots, 1\,.$$

4.2.2 Výběr hlavního prvku

Pokud číslo na diagonále je v absolutní hodnotě malé, jeho malá změna vyvolá velkou změnu výsledku při dělení a rostou zaokrouhlovací chyby.

Proto v každém kroku eliminace vybereme na diagonálu koeficient s co největší absolutní hodnotu = **hlavní prvek** (**pivot**).

GEM s výběrem hlavního prvku

- **úplným** – vybíráme z $(n-k)^2$ prvků zbylé čtvercové podmatice (výpočetně složité),

- **sloupcovým** – vybíráme v rámci sloupce a pouze vyměníme řádky,

- **řádkový** – vybíráme v rámci řádku a vyměníme sloupce (**i pořadí neznámých!**).

GEM s výběrem hlavního prvku

- **úplným** – vybíráme z $(n-k)^2$ prvků zbylé čtvercové podmatice (výpočetně složité),
- **sloupcovým** – vybíráme v rámci sloupce a pouze vyměníme řádky,
- **řádkový** – vybíráme v rámci řádku a vyměníme sloupce (**i pořadí neznámých!**).