Řešení nelineárních rovnic

1 Úvod

Numerické řešení nelineárních (NL) rovnic – vždy **iterační**
řešení odhadneme a pak ho postupně zpřesňujeme

Typy úloh - (dle počtu rovnic – proměnných)

- 1. NL rovnice

$$f(x) = 0 \tag{1}$$

  Řešení často nazýváme **kořen**. Kořen nemusí $\exists$, může být 1 nebo jich může být více.
  Jedná se o relativně snadnou úlohu, vždy lze kořen odhadnout (ohraničit) a následně najít.

- Řešení systému rovnic n rovnic o n neznámých. Pomocí n proměnných bude možno splnit najednou n rovnic.

$$\vec{f}(\vec{x}) = \vec{0} \tag{2}$$

  kde $\vec{f}$ je n-dimenzionální vektorová funkce, jejímiž složkami jsou jednotlivé rovnice, které mají být simultánně splněny.
  Řešení nemusí $\exists$, může být 1 nebo více bodových řešení. V degenerovaném případě může ale existovat i spojitá množina řešení.
  Ve více dimenzích není k dispozici žádná obecná metoda řešení, pokud není k dispozici dobrý odhad řešení.

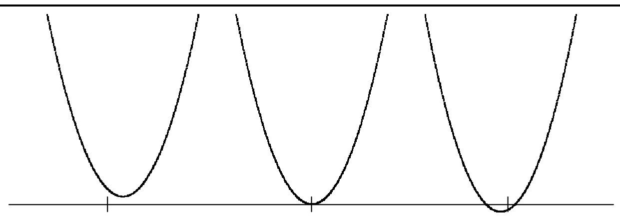
2 Řešení jedné nelineární rovnice

2 kroky řešení

1. Ohraničení kořenů (separace, bracketing) – určení intervalů, které obsahují jeden kořen.
2. Zpřesňování hodnoty kořene na požadovanou přesnost

Polynomy – $\exists$ speciální metody

Dvojnásobné kořeny – hledání řešení v reálném oboru v okolí dvojnásobného kořenu = **nekorektní úloha**
libovolně malá změna koeficientů může $\Rightarrow \neg\exists$.



Vliv nepatrné změny zadání u dvojnásobných kořenů

Pozn. V komplexním oboru je úloha vždy korektní.

Pozn. Hledání dvojnásobného kořene se provádí pomocí hledání extrému.

Ohraničení kořene – Pokud pro $x_1 < x_2$ platí, že $f(x_1)f(x_2) < 0$ je v intervalu (x_1, x_2) alespoň jeden kořen.

Algoritmus ohraničení spočívá v rozšiřování, příp. zkracování původně navrženého intervalu.

Hledání ohraničeného kořene – Obvyklé jsou metody, které nepoužívají derivace. Užití derivace $\Leftrightarrow$ pro derivaci je analytický vzorec $\wedge$ rychlý numerický výpočet.

2.1 Metoda půlení intervalů

Nechť je kořen ohraničen $\langle a_0, b_0 \rangle$, tak že $f(a_0)f(b_0) < 0$. Označme $x_1 = (a_0 + b_0)/2$. Jeden krajní bod ponecháme a druhý posuneme do x_1 tak, aby opět platilo $f(a_1)f(b_1) < 0$.

Po n-tém kroku kořen omezený body a_n a b_n a nepřesnost určení kořene je $\epsilon_n = |b_n - a_n|$.

Platí

$$|\epsilon_{n+1}| = \frac{|\epsilon_n|}{2}$$

Pozn. Obecně lze zapsat $|\epsilon_{n+1}| = C|\epsilon_n|^m$, kde $m \geq 1$.

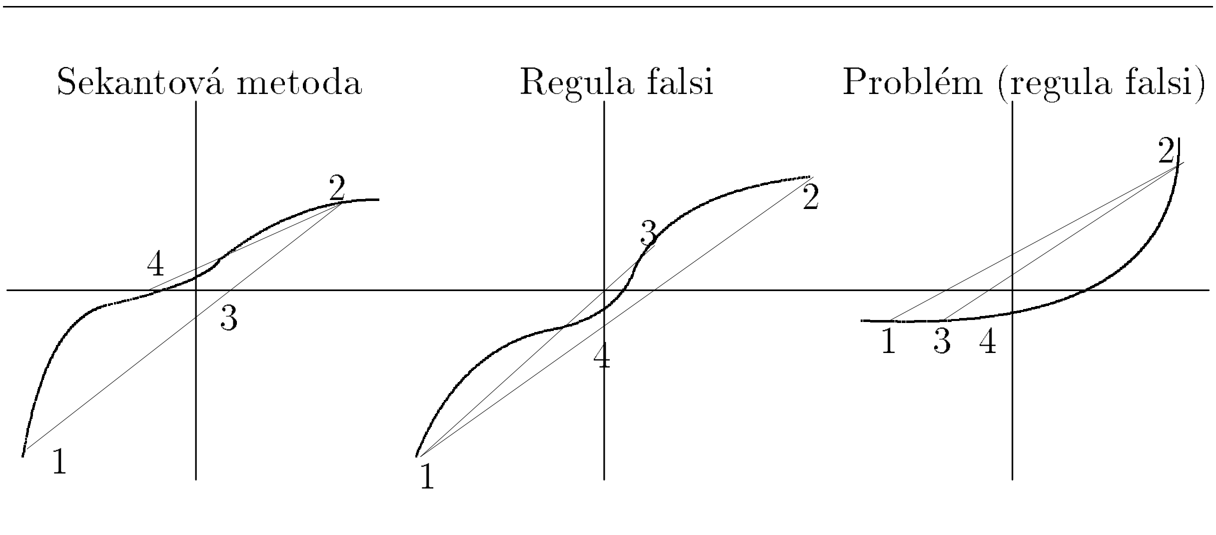
Půlení intervalů je lineární metoda $m = 1$ a $C = \frac{1}{2}$.

Počet kroků pro výpočet kořene s přesností ϵ je při počáteční chybě ϵ_0 roven

$$n = \log_2 \frac{\epsilon_0}{\epsilon} \quad .$$

Metoda půlení intervalů je spolehlivá (vždy konverguje), ale v blízkosti kořene pomalá.

2.2 Metody, užívající sečnu



Sekantová metoda, metoda regula falsi a problém pomalé konvergence

Jsou-li body a_{n-1} a a_n, pak bod a_{n+1} zvolíme v průsečíku spojnice bodů $(a_{n-1}, y(a_{n-1}))$ a $(a_n, y(a_n))$ s osou x.

Sekantová metoda – body a_n, $a_{n+1} \rightarrow$ bod a_{n+2}.
Ohraničení kořene nemusí být zachováno $\Rightarrow$ konvergence není zaručena!
V blízkosti kořene rychlejší než regula falsi. Pro rychlost konvergence platí $\lim\limits_{k\to\infty} |\epsilon_{k+1}| = C|\epsilon_k|^{1.618}$.
Metoda regula falsi - Po určení a_{n+1} si k němu vyberu z z a_{n-1} a a_n bod $\tilde{a_n}$ tak, aby kořen zůstal ohraničen $f(a_{n+1})f(a_n) < 1$. Konvergence je tudíž zaručena.
Metoda pomalejší než sekantová, ale je superlineární ($m > 1$).

Problém superlineárních metod – možnost velmi pomalé konvergence (malých kroků) daleko od kořene (viz obr).

2.3 Brentova metoda

Metody je založena na přepínání mezi lineární metodou (metodou půlení intervalů) a superlineární metodou (inverzní kvadratická interpolace). Pokud je superlineární metoda pomalá (daleko od kořene), využívá se půlení intervalů.

Inverzní kvadratická interpolace využívá funkci $x = g(y)$, hledáme $x = g(y = 0)$. Při iteraci z 3 známých bodů $a,\ b,\ c$, je funkce y interpolována podle **Lagrangeova vzorce**

$$\begin{aligned} x &= \frac{[y - f(a)][y - f(b)]c}{[f(c) - f(a)][f(c) - f(b)]} + \\ &+ \frac{[y - f(b)][y - f(c)]a}{[f(a) - f(b)][f(a) - f(c)]} + \frac{[y - f(c)][y - f(a)]b}{[f(b) - f(c)][f(b) - f(a)]} \quad . \end{aligned}$$

Pro $y \equiv 0$ lze Lagrangeův vzorec napsat ve tvaru

$$x = b + \frac{P}{Q}\,, \quad \text{kde} \quad \begin{aligned} P &= S[T(R-T)(c-b) - (1-R)(b-a)] \\ \text{a} \quad Q &= (T-1)(R-1)(S-1)\ . \end{aligned}$$

$$\text{a kde} \qquad R \equiv \frac{f(b)}{f(c)}\,, \qquad S \equiv \frac{f(b)}{f(a)}\,, \qquad T \equiv \frac{f(a)}{f(c)}\ .$$

2.4 Newton–Raphsonova (tečnová) metoda

Využívá první derivaci zadané funkce, proto je vhodná zejména pokud lze hodnoty derivací rychle počítat. Zadanou funkci $f(x)$ rozvineme do Taylorova rozvoje v okolí bodu x_i. Je-li $x = x_i + \delta$, pak platí

$$f(x) = f(x_i + \delta) = f(x_i) + \delta f'(x_i) + \frac{\delta^2}{2} f''(x_i) + \dots$$

Řešíme $f(x) = 0$, nahradíme Taylorovu řadu tečnou přímkou, δ určíme z podmínky

$$\delta = -\frac{f(x_i)}{f'(x_i)} \;\Rightarrow\; x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Pro nepřesnost $\epsilon_{i+1} = x - x_{i+1}$ $(i+1)$-ní aproximace kořene platí

$$\epsilon_{i+1} = \epsilon_i - \delta = \epsilon_i + \frac{f(x_i)}{f'(x_i)} \simeq -\epsilon_i^2 \, \frac{f''(x_i)}{2\, f'(x_i)}$$

Newton–Raphsonova metoda je tedy kvadratická metoda $\rightarrow$ rychlá blízko u kořene. Konvergence není zaručená, nutná kontrola ohraničení kořene a kombinace s metodou půlení intervalů.

3 Kořeny polynomů

3.1 Ohraničení maximální a minimální velikosti kořene

Nechť $f(x)$ je polynom ve tvaru $f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = 0$, kde $a_n \neq 0$.

Ohraničení kořenů polynomů:

1. Všechny kořeny jsou v mezikruží $\frac{|a_0|}{B+|a_0|} \leq |x| \leq 1 + \frac{A}{|a_n|}$, kde pro A a B platí $A = \max\{|a_{n-1}|, |a_{n-2}|, \ldots, |a_0|\}$ a $B = \max\{|a_n||a_{n-1}|, \ldots, |a_1|\}$.

2. Dále nechť $a_n > 0$ a a_{n-k} je první záporný koeficient, platí pro všechna $x_i > 0$, že $x_i < R = 1 + \sqrt[k]{\frac{A}{a_n}}$, kde $A = \max\limits_{j, a_j < 0} |a_j|$.

Druhé ohraničení lze po substitucích využít i k dalším odhadům:

- Substitucí $y = \frac{1}{x}$ odhadneme minimální kladný kořen.

- Pomocí substituce $y = -x$ omezíme v absolutní hodnotě největší záporný kořen.
- Substituce $y = -\frac{1}{x}$ omezí v absolutní hodnotě nejmenší záporný kořen.

3.2 Sturmova věta

Nejprve definujeme **Sturmovu posloupnost**.To je posloupnost polynomů, kde první dva členy jsou polynom $f_0(x) = f(x)$ a jeho derivace $f_1(x) = f'(x)$. Další členy f_{i+1} získáme jako mínus zbytek po dělení f_{i-1}/f_i. Posloupnost končí členem $f_k = \mathrm{const.}$, kde $k \leq n$, kde n je řád polynomu $f(x)$.

Sturmova věta – Nechť algebraická rovnice má pouze jednoduché kořeny, potom počet reálných kořenů na intervalu $\langle \alpha, \beta \rangle$ je roven rozdílu počtu znaménkových změn ve Sturmově posloupnosti $f_0, f_1, \ldots, f_k$ v bodech α a β.

Pokud má algebraická rovnice násobné kořeny, tedy $f_k = 0$, dělíme ji polynomem f_{k-1} a použijeme Sturmovu větu. Odtud potom dostaneme počet kořenů (bez násobnosti) na daném intervalu.

Příklad na Sturmovu větu Máme polynom $f(x) = 4x^3 - 2x^2 - 4x - 3 = 0$, potom členy Sturmovy posloupnosti jsou

$$\begin{aligned}
f_0 &= 4x^3 - 2x^2 - 4x - 3, \\
f_1 &= 3x^2 - x - 1, \qquad \text{(bylo vyděleno 4)} \\
f_2 &= 26x + 29, \\
f_3 &= -1.
\end{aligned}$$

Protože $f_3 = -1$, neexistují žádné násobné reálné kořeny. Znaménka členů Sturmovy posloupnosti zaneseme do tabulky.

x	$-\infty$	0	2	$+\infty$
$\mathrm{sgn} f_0(x)$	-	-	$+$	$+$
$\mathrm{sgn} f_1(x)$	$+$	-	$+$	$+$
$\mathrm{sgn} f_2(x)$	-	$+$	$+$	$+$
$\mathrm{sgn} f_3(x)$	-	-	-	-
$n_{\text{změn}}$	2	2	1	1

Rozdíl počtu znaménkových změn v bodech $-\infty$ a $+\infty$ je jedna, zadaný polynom má tedy v tomto intervalu právě jeden kořen. Tento kořen leží mezi body 0 a 2, protože rozdíl počtu znaménkových změn v těchto bodech je opět jedna.

3.3 Müllerova metoda hledání kořene

V bodech x_i, x_{i-1} a x_{i-2} budeme interpolovat zadaný polynom polynomem kvadratickým. Definujeme

$$t = \frac{x - x_i}{x_i - x_{i-1}} \quad .$$

Funkci $f(x)$ interpoluje $\tilde{f} = A\,t^2 + B\,t + C$. Při hledání kořene této kvadratické rovnice substituujeme $u = 1/t$ a dostaneme $A + B\,u + C\,u^2 = 0$ s kořeny

$$u_{1,2} = \frac{-B \pm \sqrt{B^2 - 4\,A\,C}}{2C}$$

Odtud iterační vztah pro hledání kořene polynomu ve tvaru

$$x_{i+1} = x_i - (x_i - x_{i-1}) \left[\frac{2C}{B \pm \sqrt{B^2 - 4AC}} \right],$$

kde $\pm$ nahradíme znaménkem $+$ nebo $-$ tak, aby byla maximální absolutní hodnota jmenovatele.

Pokud definujeme $q \equiv \frac{x_i - x_{i-1}}{x_{i-1} - x_{i-2}}$, můžeme koeficienty A, B a C zapsat ve tvaru

$$\begin{aligned}
A &\equiv q\,P(x_i) - q\,(1+q)\,P(x_{i-1}) + q^2\,P(x_{i-2}), \\
B &\equiv (2q+1)\,P(x_i) - (1+q)^2\,P(x_{i-1}) + q^2\,P(x_{i-2}), \\
C &\equiv (1+q)\,P(x_i).
\end{aligned}$$

Pozn. I při hledání reálných kořenů můžeme při využití této metody dostat komplexní mezivýsledky.

Pozn. Používá se i pro komplexní kořeny analytických funkcí.

3.4 Laguerrova metoda hledání kořene

Polynom n-tého stupně $P_n(x) = (x - x_1)(x - x_2)\ldots(x - x_n)$ má logaritmus $\ln|P_n(x)| = \ln|x - x_1| + \ln|x - x_2| + \ldots + \ln|x - x_n|$. Definujeme

$$\begin{aligned}
G &\equiv \frac{P'_n}{P_n} = \frac{d\ln|P_n(x)|}{dx} = \frac{1}{x - x_1} + \frac{1}{x - x_2} + \ldots + \frac{1}{x - x_n} \\
H &\equiv \left[\frac{P'_n}{P_n}\right]^2 - \frac{P''_n}{P_n} = -\frac{d^2\ln|P_n(x)|}{dx^2} = \frac{1}{(x - x_1)^2} + \ldots + \frac{1}{(x - x_n)^2}
\end{aligned}$$

Nyní G a H vyjádříme tak, že pro kořen x_1 nejbližší k x, položíme $a \equiv x - x_1$ a pro ostatní kořeny x_i předpokládáme, že $b \sim x - x_i$. Potom

$$G = \frac{1}{a} + \frac{n-1}{b} \qquad \text{a} \qquad H = \frac{1}{a^2} + \frac{n-1}{b^2} \quad .$$

Odtud pro a máme vztah

$$a = \frac{n}{G \pm \sqrt{(n-1)\,(n\,H - G^2)}},$$

kde za $\pm$ bereme takové znaménko, aby byl jmenovatel v absolutní hodnotě maximální.

Pozn. I při hledání reálných kořenů můžeme při využití této metody dostat komplexní mezivýsledky.

Pozn. Existuje ryze reálná metoda na výpočet reálných kořenů pomocí rozkladu na kvadratické polynomy (**Bairstowova metoda**).

3.5 Hledání dalších kořenů polynomu

Najdeme-li kořen x_i nahradíme původní polynom $P(x)$ polynomem $\tilde{P}(x) = P(x)/(x - x_i)$.
Výhody a nevýhody:

- Vyhneme opětovné konvergenci ke kořeni x_i (pokud k němu metoda opět konverguje, je to vícenásobný kořen)
- U polynomů nižšího stupně snáze hledají kořeny
- Ztrácíme přesnost $\rightarrow$ kořen zpřesnit pomocí původního polynomu $P(x)$

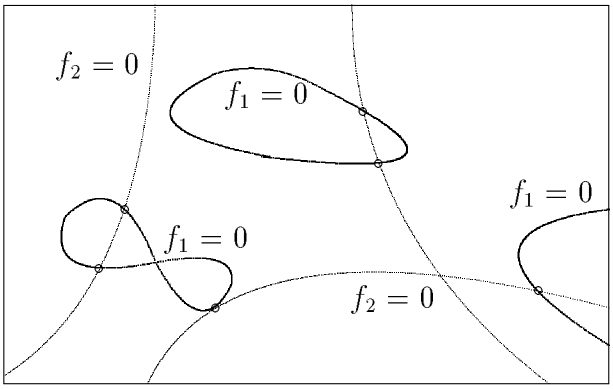
Syntetické dělení polynomů – způsob výpočtu koeficientů podílu polynomů
Koeficienty podílu a zbytku po dělení dvou polynomů dostaneme pomocí procedury `POLDIV` z knihovny Numerical Recipies. Výpočet probíhá následovně

$$\frac{a_n x^n + a_{n-1} x^{n-1} + \ldots + a_0}{b_m x^m + b_{m-1} x^{m-1} + \ldots + b_0} = c_{n-m} x^{n-m} + \ldots + c_0 + \frac{d_{m-1} x^{m-1} + \ldots + d_0}{b_m x^m + \ldots + b_0}.$$

Koeficienty podílu počítáme podle těchto vztahů

$$c_{n-m} = \frac{a_n}{b_m}\,, \qquad c_{n-m-1} = \frac{a_{n-1} - c_{n-m} b_{m-1}}{b_m}\,, \qquad \ldots$$

4 Soustavy nelineárních rovnic



Soustava dvou nelineárních rovnic pro dvě neznámé

V obecném případě je řešení soustavy nelineárních rovnic velmi obtížné, neexistuje žádná dobrá obecně použitelná metoda. Již ve dvou dimenzích nepoznáme, zda jsme blízko u řešení $\rightarrow$ obr.

Je zadána soustava $\vec{f}(\vec{x}) = \vec{0}$, kterou můžeme rozepsat ve tvaru

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0, \\ f_2(x_1, x_2, \ldots, x_n) &= 0, \\ &\vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0. \end{aligned} \tag{3}$$

<u>Pozn.</u> Pokud je to možné, nahrazujeme hledání řešení systému rovnic hledáním **<u>extrému</u>** !!
Pokud $f_i = -\partial V(\vec{x})/\partial x_i$ pro $\forall i = 1, \ldots, n$, hledáme extrém potenciálu V.

4.1 Prostá iterace

Soustavu (3) lze přepsat do tvaru $\vec{x} = \vec{\varphi}(\vec{x})$, tedy

$$\begin{aligned} x_1 &= \varphi_1(\vec{x}), \\ x_2 &= \varphi_2(\vec{x}), \\ &\vdots \\ x_n &= \varphi_n(\vec{x}). \end{aligned}$$

Tato soustava má stejná řešení jako soustava (3). Iterační vzorec má tvar

$$\vec{x}^{(k+1)} = \vec{\varphi}(\vec{x}^{(k)}) \quad .$$

Postačující podmínka konvergence

Nechť v jistém okolí G řešení ξ platí, že pro $\forall \vec{x} \in G$ je $\vec{\varphi}(\vec{x}) \in G$, a $\exists q \in (0,1)$ takové, že pro $\forall \vec{x}, \vec{y} \in G$ je $||\vec{\varphi}(\vec{x}) - \vec{\varphi}(\vec{y})|| \leq q||\vec{x} - \vec{y}||$ (*kontrahující zobrazení*). Pak iterační posloupnost konverguje k řešení ξ a platí

$$||\vec{x}^{(k)} - \vec{\xi}|| \leq \frac{q}{1-q}||\vec{x}^{(k)} - \vec{x}^{(k-1)}|| \quad .$$

Pozn. Má-li $\vec{\varphi}$ všechny první parciální derivace v okolí řešení, pak lze postačující podmínku konvergence zapsat $|J\varphi(x_1, x_2, \ldots, x_n)| \leq q < 1$, kde J označuje Jacobián zobrazení $\vec{\varphi}(x_1, x_2, \ldots, x_n)$.

Pozn. Neexistuje žádný univerzální návod jak vhodné zobrazení $\vec{\varphi}$ sestrojit.

4.2 Newton–Raphsonova metoda pro systémy nelineárních rovnic

Přesné řešení $\vec{\xi}$ vyjádříme ve tvaru $\vec{\xi} = \vec{x} + \delta\vec{x}$. Hodnotu funkce v bodě ξ vyjádříme pomocí Taylorovy věty

$$f_i(\vec{x} + \delta\vec{x}) = \underbrace{f_i(\vec{x}) + \sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j}\,\delta x_j}_{=0} + O(\delta\vec{x}^2).$$

Systém rovnic linearizujeme v bodě $\vec{x}^{(k)}$ (k-tý odhad řešení). Máme soustavu n lineárních rovnic o n neznámých

$$\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & & \frac{\partial f_2}{\partial x_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \cdot \begin{pmatrix} \delta\vec{x}_1^{(k)} \\ \delta\vec{x}_2^{(k)} \\ \vdots \\ \delta\vec{x}_n^{(k)} \end{pmatrix} = - \begin{pmatrix} f_1(\vec{x}^{(k)}) \\ f_2(\vec{x}^{(k)}) \\ \vdots \\ f_n(\vec{x}^{(k)}) \end{pmatrix}.$$

Iterační vztah je tedy $x_i^{(k+1)} = x_i^{(k)} + \delta x_i^{(k)}$, kde $i = 1, \ldots, n$.
Vektorově $\vec{x}^{(k+1)} = \vec{x}^{(k)} - \left[Jf\left(\vec{x}^{(k)}\right)\right]^{-1} \vec{f}\left(\vec{x}^{(k)}\right)$.
Při dostatečně dobrém odhadu tato metoda vždy konverguje. V případě nutnosti počítáme derivace numericky.

Pozn. Metodu lze modifikovat tak, aby se omezilo nebezpečí příliš dlouhých kroků daleko od řešení. Ve $\forall$ iteračních krocích chceme pokles $\sum_{i=1}^{n} f_i^2$. Pokud k němu v některém kroku nedojde, místo kroku $\delta x_i^{(k)}$ Newtonovy metody užijeme vztah

$$x_i^{(k+1)} = x_i^{(k)} + \lambda\,\delta x_i^{(k)} \,, \qquad \text{kde} \quad \lambda \in (0, 1\rangle \;.$$