

Numerické metody

KI/NME

doc. RNDr. Jiří Felcman, CSc.

RNDr. Petr Kubera, Ph.D.

RNDr. Jiří Škvor, Ph.D.



Ústí nad Labem 2016

Kurz:	Numerické metody
Obor:	Informační systémy, Informatika (dvouoborové studium).
Klíčová slova:	numerické metody, numerická matematika, přibližné řešení
Anotace:	Základní kurz numerické matematiky pro informatiky. Záběr tohoto kurzu odpovídá rozsahem základním kurzům matematiky, tj. pokrývá úvod do matematické analýzy a lineární algebry. Konkrétně se jedná o tematiku interpolace, aproximace, integrace a derivace funkce jedné proměnné, řešení lineárních, nelineárních a diferenciálních rovnic a jejich soustav.

Autoři by zde chtěli poděkovat doc. RNDr. Jiřímu Felcmanovi, CSc. za poskytnutí zdrojů jeho přednášek v \LaTeX .

Jazyková korektura nebyla provedena, za jazykovou stránku odpovídá autor.

© Katedra informatiky, PřF, UJEP v Ústí nad Labem, 2016

Autor: doc. RNDr. Jiří Felcman, CSc., RNDr. Petr Kubera, Ph.D.,
RNDr. Jiří Škvor, Ph.D.

Obsah

1	Aproximace a interpolace v \mathbb{R}	6
1.1	Lagrangeův interpolační polynom	7
2	Kubický spline	10
2.1	Konstrukce přirozeného kubického spline	11
3	Numerická integrace funkcí	15
3.1	Newtonovy-Cotesovy vzorce	15
4	Rombergova a Gaussova kvadratura	19
4.1	Rombergova kvadratura	19
4.2	Gaussova kvadratura	20
5	Metody řešení nelineárních rovnic I	25
5.1	Řešení jedné nelineární rovnice	25
6	Metody řešení nelineárních rovnic II	29
6.1	Metoda postupných aproximací	29
6.2	Hledání kořenů polynomu	30
7	Řešení soustav lineárních rovnic I	33
7.1	Podmíněnost matic	34
7.2	Gaußova eliminace	35
8	Řešení soustav lineárních rovnic II	37
8.1	Gaußova eliminace jako faktorizační metoda	37
8.2	LU rozklad v obecném případě	38
8.3	Choleského rozklad	40
9	Řešení soustav lineárních rovnic III	41
9.1	Odvození klasických iteračních metod	42
9.2	Problematika konvergence iteračních metod	45
10	Řešení soustav lineárních rovnic IV	48
11	Výpočet vlastních čísel matice	50
11.1	Mocninná metoda	50

12 Numerická řešení obyčejných diferenciálních rovnic	53
12.1 Jednokrokové metody	55

Úvodní slovo

Organizace výuky

Tento předmět je ukončen zápočtem a zkouškou.

- Zápočet je udělen za **vypracování seminární práce**, která bude zaměřena na praktickou aplikaci probraných úloh. Téma práce bude upřesněno při výuce.
- Zkouška je ústní a je kladen důraz na **porozumnění probírané látky**, součástí zkoušky je i rozbor vypracované seminární práce.

Numerická matematika je matematická disciplína, která má široké aplikace od předpovědi počasí, informatiky, ekonomie, přes zpracování obrazu až po třeba biologii. Bohužel (pro nás) spektrum jejich nástrojů je velmi široké a navíc celá řada metod vychází z matematického aparátu, který přesahuje Vaše znalosti matematiky. Zde jsou vybrány jen některé metody, které Vám doplňují a rozšiřují Vaše znalosti matematiky a navíc nacházejí i uplatnění v informatice (např. spline křivky).

Za klíčové při studiu této disciplíny považujeme pochopení podstaty problému a z něj pramenící schopnost aplikace probraných metod. Za nejméně vhodný způsob považujeme pouhé formální se naučení vět a definic a mechanický způsob použití probraných algoritmů. U studentů informatiky považujeme za důležité i schopnost dané metody implementovat (naprogramovat) se zřetelem, jak k funkčnosti, tak i efektivitě.

Text je doplněn otázkami k zamyšlení a úkoly. Otázky k zamyšlení jsou koncipovány tak, aby Vám ukázali daný problém z jiného pohledu a též rozšířili Váš přehled o dané problematice, jsou spíše teoretičtějšího rázu. Úkoly jsou praktické, jsou zaměřeny na aplikaci daných poznatků a implementaci metod. Volbu programovacího jazyka necháváme na Vás. Zde jen poznamenejme, že probírané algoritmy a metody jsou v naprosté většině již implementovány (většinou efektivně) v různých numeckých knihovnách daných jazyků, či nástrojích pro matematické modelování MATLAB, Octave, Scilab,... A v praxi se spíše jedná o užití těchto knihoven a nástrojů, než o vlastní implementaci daných algoritmů.

1 Aproximace a interpolace v \mathbb{R}



CÍLE KAPITOLY

Cílem této kapitoly je seznámit čtenáře s obecným pohledem na aproximaci funkce v \mathbb{R} . Dále se konkrétně věnujeme jedné z aproximačních metod, konkrétně Lagrangeově interpolaci.



KLÍČOVÁ SLOVA

Lagrangeova interpolace, chyba interpolace

Jedna ze základních úloh numerické matematiky je aproximace dané funkce f jinou funkcí φ
Zadání aproximované funkce - analyticky, nebo je k dispozici

- tabulka hodnot $(x_i, f_i), x_i, f_i \in \mathbb{R}, i = 0, \dots, n, n \in \mathbb{N}, f_i = f(x_i)$
- tabulka hodnot funkce a hodnot derivací funkce do určitého řádu v uzlech x_i

Pro funkci f definovanou na uzavřeném intervalu $[a, b]$ uvažujeme *dělení intervalu* $[a, b]$:
 $a = x_0 < x_1 < \dots < x_n = b, n \in \mathbb{Z}^+ = \{0, 1, \dots\}$ a nazýváme ho *sítí*. $x_i, i = 0, \dots, n$ nazýváme *uzly* (ekvidistantní, je-li $x_i = a + ih$, kde $h \in \mathbb{R}$ je krok sítě.)

Požadavky na aproximující funkci φ :

(A) jednoduchý tvar, snadno vyčíslitelná

- polynom $\{1, x, x^2, x^3, \dots\}$
- trigonometrický polynom $\{1, \sin x, \cos x, \sin 2x, \cos 2x, \dots\}$
- racionální funkce $\frac{P_m(x)}{Q_n(x)}$, kde $P_m(x)$, resp. $Q_n(x)$ jsou polynomy m -tého, resp. n -tého stupně
- exponenciální funkce ae^{bx}

(B) rovnost hodnot, event. derivací až do řádu m v uzlech:

$$\varphi^{(j)}(x_i) = f^{(j)}(x_i), \quad \forall i = 0, \dots, n, j = 0, \dots, m \quad (1.1)$$

(C) $\|\varphi - f\|$ ‘malá’, kde $\|\cdot\|$ značí normu

Poznámka: 1.1. Od požadavku (B, 1.1) někdy upouštíme (např. proložení přímkou třemi body)

Nejčastější způsoby aproximace:

1. Interpolace - k funkci f sestrojíme funkci φ z jisté třídy \mathcal{M} splňující (B)

2. Aproximace metodou nejmenších čtverců - k funkci f sestrojíme funkci φ z jisté třídy \mathcal{M} splňující (B) ve smyslu nejmenších čtverců

- diskrétní případ

$$\sum_{i=0}^n w_i (f(x_i) - \varphi(x_i))^2 = \min_{\psi \in \mathcal{M}} \sum_{i=0}^n w_i (f(x_i) - \psi(x_i))^2$$

kde $w_i > 0, i = 0, \dots, n$ jsou zadaná čísla, zvaná váhy.

Název ‘nejmenší čtverce’ je patrný z následující ukázky:

Pro dané dělení intervalu $[a, b]$ a dané kladné váhy w_i uvažujme normu funkce f danou vztahem

$$\|f\| := \sqrt{\sum_{i=0}^n w_i (f(x_i))^2}$$

$\varphi \in \mathcal{M}$ se hledá tak, že

$$\|f - \varphi\|^2 = \min_{\psi \in \mathcal{M}} \|f - \psi\|^2$$

- spojitý případ

$$\int_a^b w(x) (f(x) - \varphi(x))^2 dx = \min_{\psi \in \mathcal{M}} \int_a^b w(x) (f(x) - \psi(x))^2 dx$$

w je kladná váhová funkce¹.

3. Čebyševova (stejněměrná) aproximace - k funkci f sestrojíme funkci φ z jisté třídy \mathcal{M} splňující

$$\max_{[a,b]} |\varphi(x) - f(x)| \leq \max_{[a,b]} |\psi(x) - f(x)|$$

pro všechny funkce $\psi \in \mathcal{M}$, kde \mathcal{M} je zvolená množina funkcí.

1.1 Lagrangeův interpolační polynom

Hledáme polynom L_n stupně nejvýše n , píšeme $L_n \in \Pi_n$, kde Π_n značí prostor polynomů stupně nejvýše n . takový že

$$L_n(x_i) = f(x_i) \quad i = 0, \dots, n, \quad (1.2)$$

x_i - navzájem různé uzly, obecně neekvidistantní. Takový polynom nazveme *Lagrangeovým interpolačním polynomem*.

Věta 1.1. *Nechť x_0, \dots, x_n jsou navzájem různé uzly. Pak existuje právě jeden interpolační polynom $L_n \in \Pi_n$:*

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

Důkaz. 1. Existence

Uvažujme polynomy

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

(tzv. *Lagrangeovy polynomy*).

¹Obecně se stačí, aby existoval její integrál v druhé mocnině na intervalu $[a, b]$, tedy $w \in L^2(a, b)$ a kladnost funkce byla porušena v konečném počtu bodů.

Platí

$$\begin{aligned} \alpha) \quad & l_i(x) \in \Pi_n, \\ \beta) \quad & l_i(x_j) = \delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases} \quad (\text{Kroneckerovo delta}). \end{aligned}$$

Položme

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

2. Jednoznačnost

Nechť $L_n^1, L_n^2 \in \Pi_n$ splňují (viz (1.2))

$$L_n^1(x_i) = L_n^2(x_i) = f(x_i) \quad \forall i = 0, \dots, n.$$

Potom $L_n^1 - L_n^2 \in \Pi_n$ je polynom, který má $(n+1)$ různých kořenů. Podle základní věty algebry je $L_n^1 - L_n^2$ nulový polynom.

□

Poznámka: 1.2. Položme

$$\omega_{n+1}(x) = (x - x_0) \dots (x - x_n).$$

Potom platí

$$\ell_i(x) = \frac{\omega_{n+1}(x)}{(x - x_i) \cdot \omega'_{n+1}(x_i)},$$

kde čárka v $\omega'_{n+1}(x_i)$ označuje derivaci.

V případě, že potřebujeme vyčíslit Lagrangeův interpolační polynom v jednom bodě, je možné použít tzv. Aitken–Nevillovo schéma.

Věta 1.2. *Bud' $L_{i0} = f(x_i)$ hodnoty interpolované funkce v uzlových bodech x_i , $i = 0, \dots, n$. Označíme-li pomocí L_{ik} Lagrangeův interpolační polynom stupně nejvýše k , který interpoluje funkci f v uzlových bodech x_{i-k}, \dots, x_i , $1 \leq k \leq i$, $i = 1, \dots, n$, pak platí*

$$L_{ik}(x) = \frac{(x - x_{i-k}) L_{i,k-1}(x) - (x - x_i) L_{i-1,k-1}(x)}{x_i - x_{i-k}}, \quad i = 1, \dots, n, \quad 1 \leq k \leq i. \quad (1.3)$$

Při použití Aitkenova–Nevillova schématu pomocí rekurentního předpisu (1.3) spočteme hodnotu $L_{nn}(x)$, pro kterou platí $L_n(x) = L_{nn}(x)$.

1.1.1 Chyba Lagrangeovy interpolace

Věta 1.3. *Nechť $f \in C^{n+1}(I)$, kde I je nejmenší interval obsahující x_0, \dots, x_n, x^* a x_0, \dots, x_n jsou navzájem různé uzly. Nechť $L_n \in \Pi_n$ je Lagrangeův interpolační polynom pro funkci f . Pak $\exists \xi \in I$*

$$f(x^*) - L_n(x^*) = \frac{f^{(n+1)}(\xi) \cdot \omega_{n+1}(x^*)}{(n+1)!}$$

(chyba Lagrangeovy interpolace v bodě x^*).

Důkaz. Pro $x^* = x_i$ je důkaz zřejmý. Pro $x^* \neq x_i$ uvažujeme funkci :

$$F(x) = f(x) - L_n(x) - t \cdot \omega_{n+1}(x)$$

kde $t \in \mathbb{R}$. Platí:

$$F(x_i) = 0 \quad \forall i = 0, \dots, n.$$

Pro vhodnou volbu

$$t := \frac{f(x^*) - L_n(x^*)}{\omega_{n+1}(x^*)} \quad (1.4)$$

platí, že $F(x^*) = 0$. F má tedy $n + 2$ nulových bodů (uzly x_i a bod x^*). Podle Rolleovy věty:

$$\begin{aligned} F' & \text{ má aspoň } n + 1 \text{ nulových bodů,} \\ & \vdots \\ F^{(n+1)} & \text{ má aspoň 1 nulový bod, označme ho } \xi. \end{aligned}$$

$$F^{(n+1)}(\xi) = 0 = f^{(n+1)}(\xi) - 0 - t \cdot (n+1)! \quad \bigg/ \frac{\omega_{n+1}(x^*)}{(n+1)!}$$

kde jsme využili toho, že $(n+1)$ -ní derivace L_n je nulová a $(n+1)$ -ní derivace ω_{n+1} je $(n+1)!$. Dosadíme-li za t ze vztahu (1.4), dostaneme

$$f(x^*) - L_n(x^*) = \frac{f^{(n+1)}(\xi) \cdot \omega_{n+1}(x^*)}{(n+1)!}.$$

□



OTÁZKY K ZAMYŠLENÍ

1. Rozmyslete si případná zjednodušení tvaru Lagrangeových polynomů v případě ekvidistantních uzlů.



ÚKOLY

1. Naprogramujte Lagrangeovu interpolační metodu. Vstupem jsou uzlové body a hodnoty funkce v nich, výstupem je graf funkce.
2. Implementujte Aitkenovo–Nevillovo schéma.
3. Pomocí Lagrangeovy interpolace interpolujte funkci $f(x) = \frac{1}{1+25x^2}$ na intervalu $[-1, 1]$ v ekvidistantních uzlových bodech. Vyzkoušejte vliv počtu uzlových bodů na kvalitu interpolace.
4. Pomocí Lagrangeovy interpolace se pokuste určit derivaci interpolované funkce $f(x)$, resp. porovnejte derivaci vstupní funkce $f'(x)$ s $L'_n(x)$. Budete tedy muset naprogramovat derivaci L. polynomu, pro její výpočet použijte její aproximaci pomocí diferencí: $L'_n(x) = \frac{L_n(x+h) - L_n(x-h)}{2h}$, kde $h > 0$ je malé.

2 Kubický spline



CÍLE KAPITOLY

V této kapitole se budeme věnovat konstrukci přirozeného kubického spline. Cílem je opět navrhnout postup prokládání zadaných bodů, který budeme moci naprogramovat.



KLÍČOVÁ SLOVA

spline křivky, kubický spline

V praktických úlohách jsou poměrně časté situace, kdy máme proložit (interpolovat) velké množství bodů. Použití Lagrangeovy interpolace povede k polynomům vysokých stupňů. Dalším důvodem je to, že i v případě poměrně jednoduchých funkcí, je hodnota interpolované funkce a Lagrangeova interpolačního polynomu značně rozdílná (mimo uzlové body), viz úkol číslo 3 v předchozí kapitole. Nabízí se tedy možnost použití tzv. *spline křivek*.

Definice 2.1. *Nechť je dáno dělení intervalu $[a, b]$, $a = x_0 < x_1 < \dots < x_n = b$ (x_i navzájem různé). Řekneme, že funkce $\varphi : [a, b] \rightarrow \mathbb{R}$ je kubický spline, jestliže*

1. φ'' je spojitá ($\varphi \in C^2[a, b]$),
2. $\varphi|_{[x_i, x_{i+1}]}$ je kubický polynom, pro $i = 0, 1, \dots, n-1$.

Poznámka: 2.1. Spline - elastické pravítko používané při stavbě lodí

Poznámka: 2.2. Kubický spline je speciálním případem *spline k -tého řádu* pro $k = 3$. Důvodem častého použití kubického spline je fakt, že lidské oko je schopné rozlišit ještě změny 2. derivace.

Věta 2.2. *Nechť $f \in C^2[a, b]$. Pak pro každý kubický spline φ splňující*

$$\varphi(x_i) = f(x_i), \quad i = 0, \dots, n,$$

platí

$$\|\varphi\| \leq \|f\|, \quad \text{kde } \|u\|^2 := \int_a^b |u''(x)|^2 dx,$$

jestliže je splněna některá z následujících třech podmínek:

$$\begin{aligned} \text{(a)} \quad & \varphi''(a) = 0 = \varphi''(b) \\ \text{(b)} \quad & \varphi'(a) = f'(a) \quad a \quad \varphi'(b) = f'(b) \\ \text{(c)} \quad & \varphi'(a) = \varphi'(b) \quad a \quad \varphi''(a) = \varphi''(b) \end{aligned} \tag{2.1}$$

Poznámka: 2.3. Ve všech třech případech (a), (b), (c) je kubický spline určen jednoznačně.

2.1 Konstrukce přirozeného kubického spline

V následujícím uvažujeme značení:

$$\begin{aligned} f_i &:= f(x_i) \quad \forall i = 0, \dots, n, \\ \varphi_i &:= \varphi|_{[x_i, x_{i+1}]} \quad \forall i = 0, \dots, n-1, \\ h_i &:= x_{i+1} - x_i \quad \forall i = 0, \dots, n-1. \end{aligned}$$

Kubický polynom φ_i je na intervalu $[x_i, x_{i+1}]$ určen čtyřmi koeficienty (a, b, c, d při kubickém polynomu $\varphi(x) = ax^3 + bx^2 + cx + d$). Počet intervalů je n , celkem máme tedy pro určení φ počet *stupňů volnosti* $4n$. Pro tyto stupně volnosti sestavíme příslušné rovnice.

Počet neznámých	$4 \times$ počet intervalů	$4n$	
Počet rovnic	$\varphi(x_i) = f(x_i), i = 0, \dots, n$	$n + 1$	
	spojitost φ v $x_i, i = 1, \dots, n-1$	$n-1$	
	spojitost φ' v $x_i, i = 1, \dots, n-1$	$n-1$	Počet rovnic je o dvě menší
	spojitost φ'' v $x_i, i = 1, \dots, n-1$	$n-1$	
		$4n-2$	

než počet neznámých. Doplníme je proto některou z podmínek (2.1), (a)–(b). Uvažujme např. podmínku (2.1), (a), tj. podmínku nulových druhých derivací v krajních bodech. Takový spline nazýváme *přirozeným kubickým splinem*. Pro určení přirozeného kubického splinu hledáme φ_i ve vhodném tvaru. Ukazuje se, že efektivní metoda není založena na vyjádření

$$\varphi_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$$

ani na vyjádření

$$\varphi_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i,$$

ale na vyjádření pomocí tzv. *momentů*, což jsou hodnoty druhé derivace φ v uzlech. Označme je M_i :

$$M_i := \varphi''(x_i), \quad i = 0, \dots, n$$

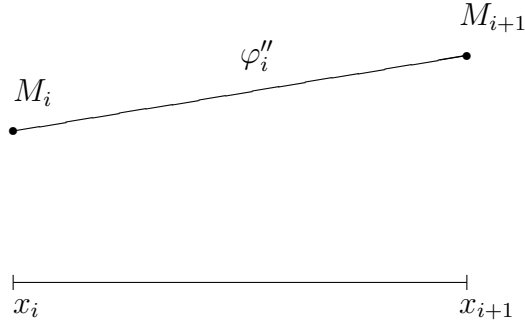
a předpokládejme, že tyto momenty známe. Později ukážeme, jak je určit. Platí

$$\begin{aligned} \varphi_i &- \text{ kubický polynom} \\ \varphi'_i &- \text{ parabola} \\ \varphi''_i &- \text{ přímka} \end{aligned}$$

Z předpokladu spojitosti druhé derivace φ v uzlech dostáváme

$$\begin{aligned} M_i &= \varphi''_i(x_i), \\ M_{i+1} &= \varphi''_i(x_{i+1}). \end{aligned}$$

Je tedy φ''_i přímka, procházející body (x_i, M_i) a (x_{i+1}, M_{i+1}) (viz obr. 2.1).



Obrázek 2.1: Přímka φ_i''

$$\begin{aligned}\varphi_i''(x) &= \frac{(x - x_i) \cdot M_{i+1}}{x_{i+1} - x_i} + \frac{M_i \cdot (x - x_{i+1})}{x_i - x_{i+1}}, \\ \varphi_i''(x) &= -\frac{M_i}{h_i} \cdot (x - x_{i+1}) + \frac{M_{i+1}}{h_i} \cdot (x - x_i).\end{aligned}$$

Integrací odvodíme

$$\begin{aligned}\varphi_i'(x) &= -\frac{M_i}{2h_i} \cdot (x - x_{i+1})^2 + \frac{M_{i+1}}{2h_i} \cdot (x - x_i)^2 + A_i, \\ \varphi_i(x) &= -\frac{M_i}{6h_i} \cdot (x - x_{i+1})^3 + \frac{M_{i+1}}{6h_i} \cdot (x - x_i)^3 + A_i(x - x_i) + B_i.\end{aligned}\quad (2.2)$$

vhodný rozpis integrační konstanty \uparrow

Ve vyjádření φ_i ve tvaru (2.2) nejprve určíme koeficienty A_i, B_i , $i = 0, \dots, n-1$ pomocí momentů a potom sestavíme rovnice pro momenty. Využijeme k tomu podmínky interpolace

$$\begin{aligned}\varphi_i(x_i) &= f_i, \\ \varphi_i(x_{i+1}) &= f_{i+1}, \quad i = 0, \dots, n-1.\end{aligned}$$

Což vede na dvě rovnice pro dvě neznámé A_i, B_i , $i = 0, \dots, n-1$. Dostaneme

$$\begin{aligned}\varphi_i(x_i) &= \frac{M_i}{6} \cdot h_i^2 + B_i = f_i, \\ \rightarrow \quad B_i &= f_i - \frac{M_i}{6} \cdot h_i^2, \\ \varphi_i(x_{i+1}) &= \frac{M_{i+1}}{6} \cdot h_i^2 + A_i h_i + f_i - \frac{M_i}{6} \cdot h_i^2 = f_{i+1}, \\ \rightarrow \quad A_i &= \frac{f_{i+1} - f_i}{h_i} + \frac{M_i - M_{i+1}}{6} \cdot h_i.\end{aligned}$$

Rovnice pro momenty sestavíme ekvivalentním vyjádřením podmínky spojitosti derivace kubického spline v uzlech:

$$\varphi_{i-1}'(x_i) = \varphi_i'(x_i), \quad i = 1, \dots, n.$$

Připomeňme si tvar φ_i'

$$\varphi_i'(x) = -\frac{M_i}{2h_i} \cdot (x - x_{i+1})^2 + \frac{M_{i+1}}{2h_i} \cdot (x - x_i)^2 + A_i$$

resp. φ'_{i-1}

$$\varphi'_{i-1}(x) = -\frac{M_{i-1}}{2h_{i-1}} \cdot (x - x_i)^2 + \frac{M_i}{2h_{i-1}} \cdot (x - x_{i-1})^2 + A_{i-1}.$$

S využitím vyjádření pro A_i , resp. A_{i-1} pomocí momentů dostaneme

$$\begin{aligned}\varphi'_{i-1}(x_i) &= 0 + \frac{M_i}{2h_{i-1}} \cdot h_{i-1}^2 + \frac{f_i - f_{i-1}}{h_{i-1}} + \frac{M_{i-1} - M_i}{6} \cdot h_{i-1} \\ &= -\frac{M_i}{2h_i} \cdot h_i^2 + 0 + \frac{f_{i+1} - f_i}{h_i} + \frac{M_i - M_{i+1}}{6} \cdot h_i = \varphi'_i(x_i).\end{aligned}$$

Protože konstruuje přirozený kubický spline, je $M_0 = \varphi''(x_0) = 0 = \varphi''(x_n) = M_n$ a dostáváme tak $n-1$ rovnic ($i = 1, \dots, n-1$) pro neznámé momenty M_1, M_2, \dots, M_{n-1} . Tyto rovnice lze přepsat ve tvaru

$$\frac{h_{i-1}}{6} M_{i-1} + \underbrace{\left(\frac{h_{i-1}}{2} - \frac{h_{i-1}}{6} + \frac{h_i}{2} - \frac{h_i}{6} \right)}_{\frac{h_{i-1}+h_i}{3}} \cdot M_i + \frac{h_i}{6} M_{i+1} = \underbrace{-\frac{f_i - f_{i-1}}{h_{i-1}} + \frac{f_{i+1} - f_i}{h_i}}_{g_i}.$$

Maticový zápis vede na soustavu s třídiagonální maticí.

$$\begin{pmatrix} \frac{h_0+h_1}{3} & \frac{h_1}{6} & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & \frac{h_{i-1}}{6} & \frac{h_{i-1}+h_i}{3} & \frac{h_i}{6} \\ & & & & \ddots & \ddots & \ddots \\ & & & & & \frac{h_{n-2}}{6} & \frac{h_{n-2}+h_{n-1}}{3} \end{pmatrix} \begin{pmatrix} M_1 \\ \vdots \\ M_{i-1} \\ M_i \\ M_{i+1} \\ \vdots \\ M_{n-1} \end{pmatrix} = \begin{pmatrix} g_1 \\ \vdots \\ g_{i-1} \\ g_i \\ g_{i+1} \\ \vdots \\ g_{n-1} \end{pmatrix}$$

Pro ekvidistanční dělení s krokem h má matice soustavy tvar

$$\frac{h}{6} \begin{pmatrix} 4 & 1 & & & \\ \ddots & \ddots & \ddots & & \\ & 1 & 4 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & 4 \end{pmatrix}$$

Nyní zbývá jen soustavu vyřešit a získat jednotlivé momenty M_i , které můžeme použít pro konstrukci spline křivky ve tvaru 2.2.

Při vyšetřování řešitelnosti této soustavy lze využít následující definici a větu z algebry:

Definice 2.3. Řekneme, že matice \mathbb{A} typu $n \times n$, $n \geq 2$ je ostře diagonálně dominantní (ODD), jestliže

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

Lemma 2.1. Všechna vlastní čísla matice \mathbb{A} typu $n \times n$, n leží v komplexní rovině ve sjednocení kruhů (Geršgorinových)

$$|z - a_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad i = 1, \dots, n. \quad (2.3)$$

Věta 2.4. Nechť $\mathbb{A} \in \mathbb{R}^{n \times n}$ je ODD. Pak \mathbb{A} je nesingulární.

Důkaz. pomocí Geršgorinových kruhů

\mathbb{A} je nesingulární $\Leftrightarrow \det \mathbb{A} \neq 0 \Leftrightarrow$ rovnice $\det(\mathbb{A} - \lambda \mathbb{I}) = 0$ nemá kořen $\lambda = 0 \Leftrightarrow$ nula není vlastním číslem matice \mathbb{A} . Nechť λ je vlastní číslo matice \mathbb{A}

$$\begin{aligned} \mathbb{A}x &= \lambda x & y &:= \frac{x}{\|x\|}, \quad \|x\| := \max_i |x_i| \\ \mathbb{A}y &= \lambda y & |y_i| &\leq 1, \quad \exists i_0 :: |y_{i_0}| = 1 \\ & \sum_{j \neq i_0} a_{i_0 j} y_j + a_{i_0 i_0} y_{i_0} &= \lambda y_{i_0} \\ & \left| \sum_{j \neq i_0} a_{i_0 j} y_j \right| &= |\lambda - a_{i_0 i_0}| |y_{i_0}| \\ & |\lambda - a_{i_0 i_0}| &\leq \sum_{j \neq i_0} |a_{i_0 j}| \\ & \text{(Geršgorinův kruh o středu } a_{i_0 i_0} \text{ a poloměru } \sum_{j \neq i_0} |a_{i_0 j}|) \end{aligned}$$

Kdyby $\lambda = 0$ bylo vlastním číslem

$$|a_{i_0 i_0}| \leq \sum_{j \neq i_0} |a_{i_0 j}| \quad \text{Spor s ODD}$$

$\lambda = 0$ tedy není vlastní číslo a matice \mathbb{A} je nesingulární. □

Matice soustavy rovnic pro momenty je ODD, soustava je tedy podle výše uvedené věty jednoznačně řešitelná a protože matice soustavy je třídiagonální, lze pro řešení použít např. Gaussovu eliminaci.



OTÁZKY K ZAMYŠLENÍ

1. Rozmyslete si jak implementovat uložení třídiagonální matice soustavy rovnic pro výpočet momentů M_i .



ÚKOLY

1. Naprogramujte interpolaci pomocí přirozeného kubického spline. Vstup uzlové body a hodnoty funkce v nich, výstupem je graf funkce.
2. Pomocí kubického spline interpolujte funkci $f(x) = \frac{1}{1+25x^2}$ na intervalu $\langle -1, 1 \rangle$ v ekvidistantních uzlových bodech. Porovnejte s Lagrangeovou interpolací.

3 Numerická integrace funkcí



CÍLE KAPITOLY

V této kapitole se budem zabývat numerickým výpočtem určitého integrálu (numerickou kvadraturou) funkcí jedné proměnné a ukážeme si základní kvadraturní formule.



KLÍČOVÁ SLOVA

numerická kvadratura, Newtonovy–Cotesovy vzorce, složené Newtonovy–Cotesovy vzorce

Výpočet určitého integrálu je poměrně obtížná úloha matematické analýzy. Obvykle při ní postupujeme tak, že určíme nejprve primitivní funkci (neurčitý integrál) a poté pomocí Newtonovy–Leibnitzovy formule spočítáme určitý integrál. Bohužel i v případě poměrně jednoduchých funkcí (např. e^{-x^2}) je nalezení primitivní funkce komplikované. A i v případě, že umíme nalézt primitivní funkci, může být velmi komplikovaná a výpočetně náročná, nebo může být integrovaná funkce zadána jen v jistých bodech. Nabízí se tedy provádět výpočet určitého integrálu přibližně, tj. numericky.

Nechť je dáno dělení intervalu $[a, b]$, $a \leq x_0 < x_1 < \dots < x_n \leq b$ (x_i navzájem různé). Označme $h = \max_{i \in \{0, \dots, n-1\}} |x_{i+1} - x_i|$.

$$\text{Cíl: } I(f) = \int_a^b f(x) dx \approx I_h(f) = \sum_{i=0}^n \alpha_i f(x_i). \quad (3.1)$$

Vzorec

$$I_h(f) = \sum_{i=0}^n \alpha_i f(x_i)$$

se nazývá *kvadraturní formule*, α_i jsou *koefficienty* kvadraturní formule a x_i jsou *uzly* kvadraturní formule. Motivace hledání aproximace určitého integrálu ve tvaru lineární kombinace hodnot funkce f v uzlech x_i je zřejmá z následujícího odstavce.

3.1 Newtonovy–Cotesovy vzorce

Pro dané $n \in \mathbb{N}$ uvažujme ekvidistantní dělení intervalu $[a, b]$ s krokem $h = \frac{b-a}{n}$, $x_i = a + ih$, $i = 0, \dots, n$. Aproximujeme-li funkci f Lagrangeovým interpolačním polynomem L_n pro uzly x_0, \dots, x_n , lze určitý integrál z funkce f aproximovat následujícím způsobem:

$$\int_a^b f(x) dx \approx \int_a^b L_n(x) dx = \int_a^b \sum_{i=0}^n f(x_i) \overbrace{\frac{(x-x_0) \dots (x-x_{i-1})(x-x_{i+1}) \dots (x-x_n)}{(x_i-x_0) \dots (x_i-x_{i-1})(x_i-x_{i+1}) \dots (x_i-x_n)}}^{\ell_i(x)} dx =$$

$$\sum_{i=0}^n \underbrace{\int_a^b \ell_i(x) dx}_{\alpha_i} f(x_i) \quad (3.1)$$

Tento vzorec nazýváme pro ekvidistantní uzly Newtonův-Cotesův. Pro výpočet koeficientů α_i použijeme následující substituci

$$\begin{aligned} \text{subst. } x &= a + th \\ x_i &= a + ih, \quad h = \frac{b-a}{n} \\ \alpha_i &:= \int_a^b \prod_{j=0, j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)} dx = \frac{b-a}{n} \int_0^n \prod_{j=0, j \neq i}^n \frac{(t - j)}{(i - j)} dt \end{aligned} \quad (3.2)$$

Uveďme si příklady některých N-C vzorců. V případě aproximace integrované funkce pomocí lineárního polynomu, mluvíme o tzv. *lichoběžníkovém pravidle* ($n = 1$):

$$T_h(f) = (b-a) \frac{f(a) + f(b)}{2}. \quad (3.3)$$

V případě aproximace integrované funkce pomocí paraboly ($n = 2$), dostáváme tzv. *Simpsonovo pravidlo*

$$S_h(f) = \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right). \quad (3.4)$$

Z konstrukce Lagrangeovy interpolace L_n funkce $f \in \Pi_n$ plyne, že $L_n(x) = f(x)$, a tedy N-C vzorec je přesný pro polynomy stupně nejvýše n . To nás vede k následující definici.

Definice 3.1. Řekneme, že kvadrurní formule $\sum_{i=0}^n \alpha_i f(x_i)$ má řád přesnosti m , jestliže $m \in \mathbb{N} \cup \{0\}$ je maximální číslo takové, že

$$\int_a^b p(x) dx = \sum_{i=0}^n \alpha_i p(x_i) \quad \forall p \in \Pi_m. \quad (3.5)$$

Lemma 3.1. Je-li kvadrurní formule $\sum_{i=0}^n \alpha_i f(x_i)$ symetrická, t.j. pro $i = 0, \dots, n$ platí

$$\begin{aligned} b - x_{n-i} &= x_i - a, \\ \alpha_i &= \alpha_{n-i}, \end{aligned}$$

a je-li její řád $\geq n$, n sudé, pak je její řád $\geq n+1$.

Lemma 3.2. Newtonův-Cotesův vzorec je symetrická kvadrurní formule.

Důsledek 3.1. Pro n sudé je řád N-C vzorce $\geq n+1$.

Nyní se budeme zabývat odhadem chyby, které se dopustíme při náhradě přesné hodnoty určitého integrálu $I(f)$ funkce $f(x)$ pomocí N-C vzorců.

Definice 3.2. (zbytek kvadrurního vzorce) Rozdíl

$$E_h(f) = I(f) - I_h(f),$$

kde

$$I(f) = \int_a^b f(x) dx, \quad I_h(f) = \sum_{i=0}^n \alpha_i f(x_i),$$

nazýváme zbytek kvadrurního vzorce.

Pro odhad chyby používáme následující lemata:

Lemma 3.3. (Odhad chyby lichoběžníkového pravidla) Necht $f \in C^2[a, b]$. Označme $T_h(f)$ N-C vzorec pro $n = 1$ (lichoběžníkové pravidlo). Pak $\exists \xi \in [a, b]$:

$$I(f) - T_h(f) = -\frac{f''(\xi)}{2} \cdot \frac{h^3}{6}, \quad h = (b - a). \quad (3.6)$$

Lemma 3.4. (Odhad chyby Simpsonova pravidla)

Necht $f \in C^3[a, b]$. Označme $S_h(f)$ N-C vzorec pro $n = 2$ (Simpsonovo pravidlo). Pak $\exists \xi \in [a, b]$:

$$I(f) - S_h(f) = -\frac{h^5}{90} \cdot f'''(\xi), \quad h = \frac{(b - a)}{2}. \quad (3.7)$$

Z výše uvedených lemat vyplývá že chyba, které se dopustíme závisí na jednak na hodnotě derivace integrované funkce a dále též na velikosti kroku h . Při zmenšování kroku h dochází ke zmenšení chyby. Tento fakt je základní myšlenkou pro složené kvadraturní vzorce.

3.1.1 Složené Newtonovy–Cotesovy vzorce

Newtonovy–Cotesovy vzorce lze také aplikovat tak, že interval $[a, b]$ rozdělíme na n ekvidistantních subintervalů $[x_i, x_{i+1}]$ velikosti h a na každém z těchto subintervalů použijeme Newtonův–Cotesův vzorec pro $m + 1$ ekvidistantních uzlů $x_i = x_{i_0} < \dots < x_{i_m} = x_{i+1}$ s krokem H

$$x_i = a + ih, \quad h = \frac{b - a}{n}, \quad i = 0, \dots, n, \quad x_{i_j} = x_i + jH, \quad H = \frac{h}{m} \quad n, m \in \mathbb{N}$$

$$I(f) := \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x) dx \approx \sum_{i=0}^{n-1} I_H^i(f) = \sum_{i=0}^{n-1} \sum_{j=0}^m \alpha_{i_j} f(x_{i_j}) =: I_H(f)$$

Věta 3.3. (složené N-C vzorce) Necht $f \in C^{m+1}[a, b]$. Pak pro složené N-C vzorce platí

$$|I(f) - I_H(f)| \leq cH^{m+1}, \quad (3.8)$$

kde $c > 0$ je konstanta nezávislá na H .

Důkaz. plyne z odhadu chyby Lagrangeova interpolačního polynomu □



OTÁZKY K ZAMYŠLENÍ

1. Rozmyslete si jak postupovat v případě určitých integrálů, kde jedna z mezí je nevlastní.
2. Uvažujte složené lichoběžníkové pravidlo s postupným zjemňováním intervalu $[a, b]$ tak, že v každém kroku dochází k **rozpuštění** dílčích subintervalů:
 $[a, b]$ na počátku
 $[a, (a + b)/2]$ a $[(a + b)/2, b]$ v druhém kroku
 $[a, (a + b)/4]$, $[(a + b)/4, (a + b)/2]$, $[(a + b)/2, 3(a + b)/4]$ a $[3(a + b)/4, b]$ ve třetím kroku atd.
 Navrhněte takový postup, kdy se použije pouze hodnota odhadu $I_h(f)$ z posledního kroku a hodnoty funkce f v nově přidaných bodech.

3. Rozmyslete si, jak určit hodnotu m ze vzorce 3.8, viz druhý z praktických úkolů.

ÚKOLY

1. Implementujte složené lichoběžníkové a Simpsonovo složené pravidlo. Vstupem je interval, integrovaná funkce a velikost kroku H .
2. Experimentálně určete pro obě složená pravidla závislost chyby které se dopustíme při numerické kvadratuře na velikosti kroku H . Použijte program z předchozího úkolu a funkci pro kterou umíte snadno přesně spočítat její kvadraturu. Zjištěnou závislost zakreslete do grafu, kde jsou obě osy logaritmické.

4 Rombergova a Gaussova kvadratura



CÍLE KAPITOLY

V předchozí kapitole jsme se zabývali základními metodami numerické integrace. Postup je sice přímočarý, ale pro dosažení určité přesnosti je potřeba relativně vysokého počtu kroků. Nyní se zaměříme na metody, které tento nedostatek odstraňují.



KLÍČOVÁ SLOVA

numerická kvadratura, Rombergova kvadratura, Richardsonova extrapolace, Gaussova kvadratura

4.1 Rombergova kvadratura

Uvažujme výpočet $\int_a^b f(x) dx$ pomocí složeného lichoběžníkového pravidla pro $n + 1$ uzlů.

$$\begin{aligned}h &= \frac{b-a}{n}, \\m &= 1, \\H &= h.\end{aligned}$$

Věta 4.1. (Eulerova-MacLaurinova) Nechť $f \in C^{2N+2}[a, b]$, $h = \frac{b-a}{n}$, $n \in \mathbb{N}$. Potom pro složené lichoběžníkové pravidlo (označme ho $CT_h(f)$) platí:

$$CT_h(f) = p(h^2) + O(h^{2N+2}) \quad (4.1)$$

$$= I(f) + a_1 h^2 + a_2 h^4 + \dots + a_N h^{2N} + O(h^{2N+2}), \quad (4.2)$$

$$\text{kde } p \in \Pi_N, \quad p = p(t) = a_0 + a_1 t + \dots + a_N t^N,$$

$$a_0 = p(0) = \int_a^b f(x) dx = I(f).$$

Rombergova kvadratura je založena na tom, že konstruujeme lineární kombinaci vzorců $CT_h(f)$ pro vhodné h tak, abychom získali vzorec, který je přesnější. Například uvažujme dvě aproximace $\int_a^b f(x) dx$ získané pomocí lichoběžníkového pravidla s krokem h ($CT_h(f)$) a s krokem $h/2$ ($CT_{h/2}(f)$):

$$CT_h(f) = I(f) + a_1 h^2 + O(h^4) \quad / - 1$$

$$CT_{h/2}(f) = I(f) + a_1 \frac{h^2}{4} + O(h^4) \quad / 4$$

$$\frac{4CT_{h/2}(f) - CT_h(f)}{3} = I(f) + O(h^4)$$

$$\text{lin. k.} = I(f) + \text{chyba} \quad (N = 1)$$

$\frac{h^2}{16}$	$\frac{h^2}{4}$	h^2
$CT_{\frac{h}{4}}(f)$	$CT_{\frac{h}{2}}(f)$	$CT_h(f)$

Tabulka 4.1: Tabulka hodnot pro Lagrangeovu interpolaci

Vhodnou lineární kombinací vzorců, z nichž každý aproximuje integrál $I(f)$ s chybou $O(h^2)$, jsme tak odvodili vzorec, který aproximuje integrál $I(f)$ s chybou $O(h^4)$. Za předpokladu dostatečné hladkosti funkce f (viz Eulerova–MacLaurinova věta) můžeme tímto způsobem odvodit vzorec, který aproximuje integrál $I(f)$ s chybou $O(h^{2N+2})$.

Předpokládejme, že pro tabulku (4.1) máme sestrojen Lagrangeův interpolační polynom v následujícím tvaru $L_2(t) = \underbrace{b_0}_{L_2(0)} + b_1 t + b_2 t^2$. Všimněme si, jakou roli hraje v tomto postupu

vyčíslení Lagrangeova interpolačního polynomu L_2 . Protože se jedná o interpolaci, tak na základě Eulerovy–MacLaurinovy věty platí následující rovnosti

$$\begin{array}{ccc} \text{Euler–MacLaurin} \downarrow & \text{Lagrange} \downarrow & \\ CT_h(f) = & I(f) + a_1 h^2 + a_2 h^4 + O(h^6) & = L_2(0) + b_1 h^2 + b_2 h^4 \end{array} \quad (4.3)$$

$$CT_{\frac{h}{2}}(f) = I(f) + a_1 \frac{h^2}{4} + a_2 \frac{h^4}{16} + O(h^6) = L_2(0) + b_1 \frac{h^2}{4} + b_2 \frac{h^4}{16} \quad (4.4)$$

$$CT_{\frac{h}{4}}(f) = I(f) + a_1 \frac{h^2}{16} + a_2 \frac{h^4}{256} + O(h^6) = L_2(0) + b_1 \frac{h^2}{16} + b_2 \frac{h^4}{256} \quad (4.5)$$

$$\text{lin. k. } \uparrow = \quad I(f) + 0 + 0 + O(h^6) \quad = L_2(0) + 0 + 0 \quad (N = 2)$$

kde rovnost v posledním řádku je získána pomocí lineární kombinace hodnot tak, aby došlo k vzájemné eliminaci členů obsahujících a_1, a_2 . Na základě podobnosti polynomů v obou sloupcích, je jasné, že ta samá lineární kombinace povede k eliminaci členů obsahujících b_1 a b_2 .

Z výše uvedeného je vidět, že $L_2(0)$ aproximuje $\int_a^b f(x) dx$ s chybou $O(h^6)$. Při konstrukci $L_2(0)$ se jedná o tzv. Richardsonovu extrapolaci. Uvedený postup lze provést až do řádu $2N+2$ pro uzly $(\frac{h}{2^i})^2$ a hodnoty $CT_{\frac{h}{2^i}}$, $i = 0, \dots, N$, pomocí nichž konstruujeme L_N .

Pro vyčíslení Lagrangeova interpolačního polynomu v jediném bodě (zde konkrétně v 0) použijeme Aitkenovo–Nevilleovo schéma, viz věta 1.2, kde dosadíme za $x = 0$ a budeme uvažovat postupné zjemňování kroku $h^2, (h/2)^2, \dots, (h/2^N)^2$ jako ve výše uvedených příkladech. Dostáváme tak následující vzorec

$$T_{ik} = T_{i,k-1} + \frac{T_{i,k-1} - T_{i-1,k-1}}{4^k - 1}, \quad i = 0, 1, \dots, N, \quad k = 1, 2, \dots, i, \quad (4.6)$$

kde hodnoty T_{i0} jsou dány $T_{i0} = CT_{\frac{h}{2^i}}(f)$. V praxi se používá tak, že inkrementujeme i dokud rozdíl dvou dosud nejpřesnějších hodnot $|T_{ik} - T_{i-1,k}|$ není menší než předepsaná hodnota.

4.2 Gaussova kvadratura

Víme, že N-C vzorce (3.1) mají řád aspoň n (pro n sudé dokonce aspoň $n+1$). Jakého řádu může být formule typu $\sum_{i=0}^n \alpha_i f(x_i)$?

Uvažujme pro dané dělení intervalu $[a, b]$, $a \leq x_0 < \dots < x_n \leq b$ kvadrturní formuli

$$I_h(f) = \sum_{i=0}^n \alpha_i f(x_i). \quad (4.7)$$

Lemma 4.1. (*Řád kvadrurní formule*) Řád kvadrurní formule (4.7) je nejvýše $2n + 1$.

Důkaz. Uvažujme polynom $\tilde{p}(x) = \prod_{i=0}^n (x - x_i)^2 \in \Pi_{2n+2}$. Tento polynom je nezáporná funkce na intervalu $[a, b]$ a platí pro něj

$$\int_a^b \tilde{p}(x) dx > 0.$$

Kvadrurní formule typu (4.7) dává pro tento polynom

$$\sum_{i=0}^n \alpha_i \tilde{p}(x_i) = 0.$$

Pro polynom \tilde{p} není tedy kvadrurní formule (4.7) přesná a její řád je tedy nejvýše $2n + 1$. \square

Gaußova kvadratura je způsob konstrukce vzorce $\sum_{i=0}^n \alpha_i f(x_i)$, který je přesný pro všechny polynomy stupně nejvýše $2n + 1$.

Definice 4.2. *Skalární součin v $C[a, b]$ je definován*

$$(u, v) = \int_a^b u(x)v(x) dx. \quad (4.8)$$

Poznámka: 4.1. Lze snadno ověřit, že takto definovaný součin má obvyklé vlastnosti skalárního součinu, viz otázky a úkoly k této kapitole.

Poznámka: 4.2. Podobně jako v definici 4.2 lze definovat skalární součin v $C[a, b]$

$$(u, v) = \int_a^b u(x)v(x)w(x) dx, \quad (4.9)$$

kde w je nezáporná a integrovatelná funkce na intervalu $[a, b]$. Zde vidět, že výše uvedená definice 4.2 odpovídá volbě váhové funkce $w(x) = 1$.

Definice 4.3. *Množina normovaných polynomů*

$$\tilde{\Pi}_n = \{p \in \Pi_n; \quad p(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_0\}. \quad (4.10)$$

Věta 4.4. (*Ortogonalní polynomy*) Existují jednoznačně určené polynomy p_i , pro které platí

$$1. \quad p_i \in \tilde{\Pi}_i, \quad i \in \mathbb{N} \cup \{0\},$$

$$(p_i, p_j) = 0, \quad i \neq j,$$

$$2. \quad \text{Kořeny } x_0, \dots, x_n \text{ polynomu } p_{n+1}, n \in \mathbb{N} \cup \{0\}, \text{ jsou reálné, jednoduché a leží v } (a, b)$$

3.

$$A = \begin{pmatrix} p_0(x_0) & p_0(x_1) & \cdots & p_0(x_n) \\ p_1(x_0) & p_1(x_1) & \cdots & p_1(x_n) \\ & & \ddots & \\ p_n(x_0) & p_n(x_1) & \cdots & p_n(x_n) \end{pmatrix} \quad \text{je nesesingulární.}$$

Poznámka: 4.3. Posloupnost ortogonálních polynomů lze sestavit pomocí Gramova-Schmidtova ortogonalizačního procesu. V dalším budeme uvažovat skalární součin z definice 4.2 a interval $[a, b] = [-1, 1]$. Polynomy, které takto získáme se nazývají *Legendrovy polynomy*. Platí pro ně následující rekurentní předpis¹

$$p_{n+1}(x) = \frac{2n+1}{n+1}xp_n(x) - \frac{n}{n+1}p_{n-1}(x), \quad p_0(x) = 1, \quad p_1(x) = x. \quad (4.11)$$

¹Nemusíme je tedy nutně konstruovat pomocí Gramova-Schmidtova ortogonalizačního procesu.

Protože v dalším budeme používat Legendrovy polynomy, nazývají se odvozené kvadraturní formule *Gaußovy-Legendrovy*². Myšlenka konstrukce Gaußovy kvadratury: Uzly x_i volíme jako kořeny polynomu p_{n+1} z množiny ortogonálních polynomů $\{p_0, p_1, \dots, p_{n+1}\}$. S využitím ortogonálních polynomů p_0, \dots, p_{n+1} a kořenů x_i polynomu p_{n+1} určíme koeficienty α_i Gaußovy kvadraturní formule tak, aby platilo:

$$\int_a^b q(x) dx = \sum_{i=0}^n \alpha_i q(x_i), \quad \forall q \in \Pi_{2n+1}. \quad (4.12)$$

K tomu vyjádříme polynom q ve tvaru

$$q(x) = r(x)p_{n+1}(x) + s(x), \quad r, s \in \Pi_n,$$

(dělení polynomu q polynomem p_{n+1}) a polynomy $r(x), s(x) \in \Pi_n$ vyjádříme jako lineární kombinaci ortogonálních polynomů, speciálně nechť

$$s(x) = \sum_{j=0}^n \gamma_j p_j(x).$$

Na základě tohoto vyjádření má výraz na levé straně v (4.12) tvar

$$\begin{aligned} \int_a^b q(x) dx &= \underbrace{\int_a^b r(x)p_{n+1}(x) dx}_{=0} + \int_a^b s(x) dx \\ &= \int_a^b \sum_{j=0}^n \gamma_j p_j(x) dx = \int_a^b \sum_{j=0}^n \gamma_j \overbrace{p_0(x)}^{=1} p_j(x) dx \\ &= \sum_{j=0}^n \gamma_j \int_a^b p_0(x)p_j(x) dx = \gamma_0 \int_a^b p_0(x)p_0(x) dx = \gamma_0 \int_a^b dx. \end{aligned}$$

Levá strana v (4.12) je tedy rovna

$$\gamma_0(b-a).$$

Pravá strana v (4.12) má na základě výše uvedených vyjádření tvar

$$\sum_{i=0}^n \alpha_i \underbrace{[r(x_i)p_{n+1}(x_i) + s(x_i)]}_{=0} = \sum_{i=0}^n \alpha_i \sum_{j=0}^n \gamma_j p_j(x_i).$$

Vidíme, že levou a pravou stranu v (4.12) lze tedy vyjádřit jako lineární kombinací jistých výrazů s koeficienty γ_j

$$\begin{aligned} &\gamma_0(b-a) + \gamma_1 \cdot 0 + \dots + \gamma_n \cdot 0 \\ &= \gamma_0 \sum_{i=0}^n p_0(x_i)\alpha_i + \gamma_1 \sum_{i=0}^n p_1(x_i)\alpha_i + \dots + \gamma_n \sum_{i=0}^n p_n(x_i)\alpha_i \end{aligned}$$

Porovnáním výrazů u koeficientů γ_j na levé a pravé straně dostaneme rovnice pro určení hledaných koeficientů α_i :

$$\left. \begin{aligned} \sum_{i=0}^n p_0(x_i)\alpha_i &= (b-a) \\ \sum_{i=0}^n p_1(x_i)\alpha_i &= 0 \\ \vdots &\vdots \\ \sum_{i=0}^n p_n(x_i)\alpha_i &= 0 \end{aligned} \right\} \Leftrightarrow \begin{pmatrix} p_0(x_0) & \cdots & p_0(x_n) \\ p_1(x_0) & \cdots & p_1(x_n) \\ & \ddots & \\ p_n(x_0) & \cdots & p_n(x_n) \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} b-a \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (4.13)$$

²Při použití skalárního součinu (4.9), resp. jiných polynomů, dostaneme jinou třídu Gaußových vzorců.

$x_0 = -\frac{1}{\sqrt{3}}$	$x_1 = \frac{1}{\sqrt{3}}$
$\alpha_0 = 1$	$\alpha_1 = 1$

Tabulka 4.2: Koefficienty a uzly pro $n = 1$

$x_0 = -\sqrt{\frac{3}{5}}$	$x_1 = 0$	$x_2 = \sqrt{\frac{3}{5}}$
$\alpha_0 = \frac{5}{9}$	$\alpha_1 = \frac{8}{9}$	$\alpha_2 = \frac{5}{9}$

Tabulka 4.3: Koefficienty a uzly pro $n = 2$

Poslední soustava rovnic v (4.13) pro koeficienty α_i je jednoznačně řešitelná, viz věta 4.4 a její poslední tvrzení.

Nyní si pro ilustraci odvodíme Gaußův kvadraturní vzorec řádu 3 ($n = 1$) na intervalu $[-1, 1]$. Použijeme ortogonální polynomy $\{1, x, x^2 - \frac{1}{3}\}$. Poslední z polynomů $p_2(x) = x^2 - \frac{1}{3}$ má dva reálné kořeny $x_{0,1} = \pm \frac{1}{\sqrt{3}}$. Dosazením do (4.13) dostáváme následující soustavu rovnic:

$$\begin{pmatrix} 1 & 1 \\ -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

Jejím řešením nalezneme hledané koeficienty $\alpha_0 = \alpha_1 = 1$. Obdobně bychom mohli pokračovat s využitím polynomů vyšších stupňů. Váhy a uzly některých vzorců jsou uvedeny v tabulkách 4.2 a 4.3.

Z hlediska stability je výhodné, že koeficienty α_i Gaußova kvadraturního vzorce $\sum_{i=0}^n \alpha_i f(x_i)$ jsou kladné.

Věta 4.5. (pozitivita α_i) *Koeficienty α_i Gaußova kvadraturního vzorce jsou kladné.*

Důkaz. Položme:

$$\begin{aligned} \tilde{p}_k(x) &= \prod_{i=0, i \neq k}^n (x - x_i)^2 \in \Pi_{2n} \\ 0 &< \int_a^b \tilde{p}_k(x) dx = \sum_{i=0}^n \alpha_i \tilde{p}_k(x_i) = \alpha_k \underbrace{\tilde{p}_k(x_k)}_{>0} \\ &\Rightarrow \alpha_k \text{ kladné } \forall k = 0, 1, \dots, n. \end{aligned}$$

□

Je nutné umět integrovat i přes jiný interval než $[-1, 1]$ pro který jsme je odvodili. Nabízí se použít větu o substituci a za použití substituce $x = \frac{1}{2}((b-a)t + a + b)$ převést obecný interval $[a, b]$ na interval $[-1, 1]$

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{1}{2}((b-a)t + a + b)\right) dt, \quad (4.14)$$

což aplikací Gaußova kvadraturního vzorce na pravou stranu předešlé rovnice vede na finální kvadraturní vzorec

$$\int_a^b f(x) dx = \frac{b-a}{2} \sum_{i=0}^n \alpha_i f\left(\frac{1}{2}((b-a)t_i + a + b)\right). \quad (4.15)$$

Poznámka: 4.4. Poznamenejme, že je samozřejmě možné používat složené Gaußovy kvadraturní vzorce.



OTÁZKY K ZAMYŠLENÍ

1. Vyjděte ze vztahu pro Aitkenovo–Nevillovo schéma (1.3) a odvoďte vzorec (4.6) pro Rombergovu kvadraturu.
2. Buď pomocí vztahu (4.11), nebo pomocí Gramova-Schmidtova ortogonalizačního procesu odvoďte Legendrovy polynomy pro $n = 2, 3$.
3. S využitím Legendrových polynomů z předchozího úkolu odvoďte Gaußovy kvadraturní vzorce pro $n = 2$, viz tabulka 4.2.



ÚKOLY

1. Implementujte Rombergovu kvadraturu. Vstupem je interval, integrovaná funkce a velikost kroku H .
2. Implementujte složené Gaußovy kvadraturní vzorce. Vstupem je interval, integrovaná funkce a velikost kroku H .
3. Experimentálně určete pro obě metody závislost chyby které se dopustíme při numerické kvadratuře na velikosti kroku H . Použijte programy z předchozích úkolů a funkci pro kterou umíte snadno přesně spočítat její kvadraturu. Zjištěnou závislost zakreslete do grafu, kde jsou obě osy logaritmické.

5 Metody řešení nelineárních rovnic I



CÍLE KAPITOLY

V této kapitole si uvedeme základní metody pro řešení jak jedné nelineární rovnice, tak i jejich soustav.



KLÍČOVÁ SLOVA

metoda půlení intervalu, metoda Regula falsi, Newtonova metoda

5.1 Řešení jedné nelineární rovnice

Věnujme se nejprve řešení úlohy $f(x) = 0$ na intervalu $[a, b]$. Takové x které tuto rovnici splňuje nazveme kořenem a budeme jej značit α . V dalším předpokládejme, že $f \in C[a, b]$ a $f(a)f(b) < 0$ a tedy kořen dané rovnice se opravdu nachází v intervalu $[a, b]$. Nejprve si uvedeme dvě nejjednodušší vždy konvergentní metody, které se používají jako startovací metody pro metody komplikovanější. Jedná se *metodu bisekce* a *metodu regula falsi*.

5.1.1 Metoda bisekce

Metoda bisekce (půlení intervalu) patří mezi nejjednodušší metody pro řešení rovnice $f(x) = 0$. Její princip spočívá v tom, že postupně půlením počátečního intervalu $[a, b]$ vytváříme posloupnost dílčích subintervalů $[a_k, b_k]$ takových, že vždy obsahují kořen α .

Položme $[a_0, b_0] = [a, b]$, střed intervalu $[a_k, b_k]$ je dán

$$c = \frac{a + b}{2}. \quad (5.1)$$

Nový interval $[a_{k+1}, b_{k+1}]$ zkonstruujeme tak, aby platilo $f(a_{k+1})f(b_{k+1}) < 0$. Což vede na následující podmínku

$$[a_{k+1}, b_{k+1}] = \begin{cases} [c, b_k], & \text{jestliže } f(c)f(b_k) < 0 \\ [a_k, c], & \text{jinak.} \end{cases} \quad (5.2)$$

Pokud budeme považovat za odhad kořene střed daného intervalu, tak pro odhad chyby platí

$$|\alpha - c| \leq \frac{b - a}{2^k}, \quad (5.3)$$

kde k je počet iterací. Na základě tohoto vzorce lze odhadnout počet nutných iterací pro dosažení dané přesnosti. Opakované půlení intervalu provádíme tak dlouho, dokud není dosaženo předepsaného počtu iterací, nebo dokud $|f(c)|$ není dostatečně malá. Poznamenejme, že pokud nastane situace, že $f(c) = 0$, tak je nalezen kořen a iterační cyklus metody by měl být ukončen.

5.1.2 Metoda Regula Falsi

Nedostatkem metody půlení intervalu bylo to, že při určování bodu c , viz vzorec (5.1), nebylo přihlíženo k průběhu funkce $f(x)$.

V metodě regula falsi postupujeme při odhadu kořene následujícím způsobem. Uvažujme dva body $(a_k, f(a_k))$ a $(b_k, f(b_k))$ a zkonstruujeme sečnu procházející těmito body. Průsečík tečny s osou x prohlásíme za odhad kořene

$$c = \frac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)}. \quad (5.4)$$

Ostatní části výpočetního postupu jsou analogické jako v případě metody půlení intervalu.

5.1.3 Newtonova metoda

Zaměřme se nyní na metodu, která potřebuje silnější podmínky konvergence než předešlé dvě metody, avšak rychlost její konvergence je mnohem vyšší. Metodu budeme rovnou formulovat pro soustavu nelineárních rovnic s tím, že jedna nelineární rovnice je pouze speciálním případem. Nechť je dáno nelineární zobrazení

$$F : \mathbb{R}^N \rightarrow \mathbb{R}^N.$$

$$\text{Hledáme } \alpha :: F(\alpha) = 0.$$

Problém $F(x) = 0$ nahradíme posloupností lineárních problémů $L_k(x) = 0$, $L_k : \mathbb{R}^N \rightarrow \mathbb{R}^N$, $k = 0, 1, \dots$, takových, že jejich řešení tvoří posloupnost konvergující k řešení problému $F(x) = 0$.

$$\alpha \approx x^{(k+1)}, \text{ kde } L_k(x^{(k+1)}) = 0.$$

Nechť $x^{(0)}$ je dáno. Pro danou aproximaci $x^{(k)}$ uvažujeme $L_k(x)$ jako lineární část Taylorova rozvoje zobrazení F v bodě $x^{(k)} \in \mathbb{R}^N$ ($J(x)$ značí Jakobiho matici zobrazení F v bodě x):

$$F(x) = \underbrace{F(x^{(k)}) + J(x^{(k)})(x - x^{(k)})}_{L_k(x)} + O(|x - x^{(k)}|^2).$$

(za předpokladu dostatečné hladkosti zobrazení F). Nelineární problém nahradíme problémem lineárním

$$\{F(x) = 0\} \approx \underbrace{\{F(x^{(k)}) + J(x^{(k)})(x - x^{(k)}) = 0\}}_{L_k(x)} \quad (5.5)$$

$$\text{řešení } \alpha \text{ nelin. pb.} \quad \text{aproximujeme} \quad \text{řešením } x^{(k+1)} \text{ lin. pb.} \quad (5.6)$$

$$\alpha \approx x^{(k+1)} := x^{(k)} - J^{-1}(x^{(k)})F(x^{(k)}) \quad (5.7)$$

Vzorec v (5.7), kterým je definována $(k+1)$ -ní aproximace $x^{(k+1)}$ řešení nelineárního problému je formální, ve skutečnosti se inverzní matice nepočítá a algoritmus má následující dva kroky:

Algoritmus:

1. $J(x^{(k)}) \underbrace{(x - x^k)}_{\delta x^{(k)}} = -F(x^{(k)})$ - řešíme lineární úlohu pro $\delta x^{(k)}$
2. $x^{(k+1)} := x^{(k)} + \delta x^{(k)}$ - provedeme update předchozí aproximace.

Newtonova metoda pro $f(x) = 0$

Pro $N = 1$ (nelineární skalární rovnice pro jednu neznámou) má Newtonova metoda názorný geometrický význam. Nelineární funkci $f(x)$ nahradíme lineární funkcí (přímku), která je tečnou ke grafu funkce f v bodě $(x^{(k)}, f(x^{(k)}))$ (má tedy směrnici $f'(x^{(k)})$ a prochází bodem $(x^{(k)}, f(x^{(k)}))$). V tomto případě se Newtonova metoda nazývá metodou tečen. V případě jedné dimenze se vzorec (5.7) výrazně zjednoduší (náhrada $J^{-1}(x^{(k)})$ za $f'(x^{(k)})^{-1}$)

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}, \quad x^{(0)} \text{ dáno, } f'(x^{(k)}) \neq 0. \quad (5.8)$$

Výpočetní schéma se používá tak dlouho, dokud $|x^{(k)} - x^{(k-1)}|$ nedosáhne předepsané hodnoty, nebo dokud není proveden předepsaný počet iterací, případně dokud není $|f(x^{(k)})| < \epsilon$.

O konvergenci a jejích podmínkách Newtonovy metody pro $f(x) = 0$ hovoří následující věta.

Věta 5.1. *Nechť $f \in C^2[a, b]$, $f(a)f(b) < 0$, $f'(x) \neq 0$ pro všechna $x \in [a, b]$ a nechť $f''(x) \leq 0$, nebo $f''(x) \geq 0$ (tj. druhá derivace nemění znaménko). Zvolme $x_0 \in [a, b]$ tak, aby platilo $f(x_0)f''(x_0) > 0$, pak posloupnost x_k generovaná předpisem (5.8) je konvergentní.*

Poznámka: 5.1. Newtonova metoda je speciálním případem náhrady funkce f lineární funkcí

$$l_k(x) := f(x^{(k)}) + (x - x^{(k)})q_k,$$

kde směrnice q_k se volí

$$q_k := f'(x^{(k)}).$$

Věta 5.2. *(Konvergence Newtonovy metody pro soustavy) Nechť $F \in C(D)$, $D \subset \mathbb{R}^N$ konvexní, otevřená množina, která obsahuje $\alpha :: F(\alpha) = 0$. Nechť $\exists J^{-1}(\alpha)$, nechť $\exists R > 0, c > 0, L > 0$ takové že:*

$$\begin{aligned} \|J^{-1}(\alpha)\| &\leq c, \\ \underbrace{\|J(x) - J(y)\|}_{\text{maticová norma}} &\leq L \underbrace{\|x - y\|}_{\text{vekt. norma}} \quad \forall x, y \in B(\alpha, R), \end{aligned}$$

kde $B(\alpha, R)$ je koule o středu α a poloměru R . Potom $\exists r, \forall x^{(0)} \in B(\alpha, r)$, posloupnost 5.7 je jednoznačně definována a konverguje k α a platí

$$\|\alpha - x^{(k+1)}\| \leq cL \|\alpha - x^{(k)}\|^2. \quad (5.9)$$

Výše uvedená věta tedy říká, že pokud jsou v kořeni $F(\alpha) = 0$ splněny výše uvedené podmínky, tak za předpokladu, že zvolíme x_0 dostatečně blízko kořene α , tak Newtonova metoda konverguje. Tato věta samozřejmě platí i pro jednu nelineární rovnici. Je zde vidět, že Newtonova metoda je citlivá na volbu počáteční iterace, srovnejte s větou 5.1. Dále nás bude zajímat rychlost konvergenčního procesu.

Řád konvergence

Definice 5.3. (Řád konvergence iterační metody pro řešení $F(x) = 0$) Řekneme, že posloupnost $\{x^{(k)}\}$ generovaná numerickou metodou konverguje k α s řádem $p \geq 1$, pokud $\exists c > 0$

$$\frac{\|\alpha - x^{(k+1)}\|}{\|\alpha - x^{(k)}\|^p} \leq c \quad \forall k \geq k_0.$$

V takovém případě se numerická metoda nazývá řádu p .

Věta 5.2 říká, že Newtonova metoda je kvadraticky konvergentní,

$$\|\alpha - x^{(k+1)}\| \leq cL \|\alpha - x^{(k)}\|^2,$$

pokud je $x^{(0)}$ dostatečně blízko α a pokud je $J(\alpha)$ nesingulární.

Poznámka: 5.2. Newtonovu metodu je možné modifikovat a tím urychlit, či zjednodušit výpočetní proces, možné modifikace Newtonovy metody:

- Jacobiho matice se nemění pro $p \geq 2$ kroků
- Nepřesné řešení soustavy lin. rovnic
- Vyčíslení Jacobiho matice pomocí diferencí $f'(x) \approx \frac{f(x+h)-f(x)}{h}$. V případě jedné dimenze náhrada derivace $f'(x^{(k)}) = \frac{f(x^{(k)})-f(x^{(k-1)})}{x^{(k)}-x^{(k-1)}}$ vede na známou metodu sečen.



OTÁZKY K ZAMYŠLENÍ

1. Odvoďte vzorec pro odhad chyby pomocí metody bisekce, viz vzorec (5.3).
2. Odvoďte vzorec 5.4.
3. Odvoďte dvoubodovou metodu sečen, viz poslední uvedená modifikace Newtonovy metody.
4. Pomocí Newtonovy metody odvoďte vzorec pro výpočet odmocniny z čísla c , tj. řešte rovnici $x^2 - c = 0$.



ÚKOLY

1. Implementujte metodu půlení intervalu a metodu regula falsi. Vstupem je interval, funkce $f(x)$, maximální počet iterací a velikost chyby.
2. Implementujte Newtonovu metodu pro $f(x) = 0$. Vstupem je počáteční odhad $x^{(0)}$, funkce $f(x)$ a její derivace, maximální počet iterací a velikost chyby.
3. Pro úlohu se známým řešením porovnejte rychlost konvergence všech tří metod, tj. sledujte chybu již se dopustíte v závislosti na iteraci.
4. Implementujte Newtonovu metodu pro soustavu dvou nelineárních rovnic o dvou neznámých. Vstupem je počáteční odhad $x^{(0)}$, funkce $F(x)$ a její Jacobiho matice $J(x)$ (tj. čtyři další funkce), maximální počet iterací a velikost chyby.

6 Metody řešení nelineárních rovnic II



CÍLE KAPITOLY

V této kapitole si uvedeme pokročilejší metody řešení nelineárních rovnic a zaměříme se též na speciální případ, a to na řešení algebraických rovnic, tj. hledání kořenů polynomu.



KLÍČOVÁ SLOVA

metoda postupných aproximací pro nelineární rovnice, kořeny polynomu, Hornerovo schéma

6.1 Metoda postupných aproximací

Metoda postupných aproximací je založena na faktu, že pro dané zobrazení $F : M \subset \mathbb{R}^N \rightarrow \mathbb{R}^N$ je vždy možné transformovat problém $F(x) = 0$ na ekvivalentní problém $x - \phi(x) = 0$, kde pomocná funkce ϕ je volena tak, aby $\phi(\alpha) = \alpha$ právě když $F(\alpha) = 0$. Bod α , pro který platí $\phi(\alpha) = \alpha$, se nazývá *pevným bodem* zobrazení ϕ . Nalezení nulových bodů zobrazení F se tak převede na nalezení pevného bodu zobrazení ϕ , které se realizuje pomocí následujícího algoritmu:

Budiž dáno $x^{(0)}$

$$x^{(k+1)} := \phi(x^{(k)}), \quad k \geq 0. \quad (6.1)$$

Definice 6.1. (kontrahující zobrazení) Řekneme, že zobrazení $G : D \subset \mathbb{R}^N \rightarrow \mathbb{R}^N$ je kontrahující na $D_0 \subset D$, jestliže $\exists L < 1$:

$$\|G(x) - G(y)\| \leq L \|x - y\| \quad \forall x, y \in D_0.$$

Věta 6.2. (Banachova věta o pevném bodě) Nechť $G : D \subset \mathbb{R}^N \rightarrow \mathbb{R}^N$ kontrahující na uzavřené množině $D_0 \subset D$, $G(x) \in D_0 \quad \forall x \in D_0$. Pak G má právě jeden pevný bod. Tento bod je limitou posloupnosti $x^{(k+1)} = \phi(x^{(k)})$, $x^{(0)} \in D_0$ libovolné.

Nyní si ukážeme aplikaci výše uvedeného na řešení rovnice $f(x) = 0$. Toto rovnici můžeme transformovat do obecného tvaru

$$x = x + \psi(x)f(x) = \phi(x), \quad (6.2)$$

kde $\psi(x)$ je vhodně zvolená funkce. Lze vidět, že pokud $f(\alpha) = 0 \Rightarrow \alpha = \phi(\alpha)$. Příkladem budiž Newtonova metoda, kde funkce ψ je zvolena $\psi(x) = -1/f'(x)$ a pak užitím algoritmu (6.1) dostáváme schéma Newtonovy metody (5.8).

V dalším zvolme $\psi(x) = -1$, tj. $x = x - f(x) = \phi(x)$. Pokud zaručíme, že zobrazení $\phi(x)$ bude kontrahující, tak díky předchozí větě o pevném bodě 6.2, získáme užitím schématu (6.1) metodu

(metoda prosté iterace) pro řešení rovnice $f(x) = 0$. Kontrakce zobrazení je zde klíčová, pokud zobrazení $\phi(x)$ nebude kontrahující, pak iterační schéma nebude fungovat. Jednou z možností jak zaručit konvergenci je splnit podmínky následující věty.

Věta 6.3. *Nechť α je pevný bod zobrazení $\phi \in C^1[a, b]$ a $|\phi(x)| < 1$, $x \in [a, b]$, pak iterační metoda daná schématem (6.1) je konvergentní pro každé $x^{(0)} \in [a, b]$.*

Použití metody je takové, že problém $f(x) = 0$ převedeme na problém hledání pevného bodu zobrazení $x = x - f(x) = \phi(x)$ pomocí schématu (6.1), kde $\phi(x)$ bude splňovat podmínky věty 6.3. Iterace pak provádíme obdobně jako v případě Newtonovy metody (například tak dlouho, dokud $|x^{(k)} - x^{(k-1)}|$ není menší než předepsaná hodnota).

6.2 Hledání kořenů polynomu

Pro nalezení kořenů polynomu $p_n(x) = a_0 + a_1x + \dots + a_nx^n$ potřebujeme nejprve lokalizovat jeho kořeny a poté je aproximovat pomocí nějaké vhodné numerické metody z předchozích kapitol. Pro jejich lokalizaci můžeme použít následující věty.

Věta 6.4 (Cauchy). *Kořeny polynomu leží v kruhu*

$$\Gamma = \left\{ z \in \mathbb{C}; |z| \leq 1 + \eta, \eta = \max_{0 \leq k \leq n-1} \left| \frac{a_k}{a_n} \right| \right\}$$

Věta 6.5 (Budanova-Fourierova). *Nechť koeficient a_n polynomu $p_n(x)$ je kladný, krajní body intervalu $[a, b]$ nechť nejsou kořeny $p_n(x)$. Označme $ZZ(x)$ počet znaménkových změn v posloupnosti $p_n(x), p'_n(x), \dots, p_n^{(n)}(x)$ v bodě x . Počet reálných kořenů (včetně násobnosti) polynomu $p_n(x)$ na intervalu $[a, b]$ je dán $ZZ(a) - ZZ(b)$, nebo je o sudé číslo menší.*

Věta 6.6 (Descartes). *Počet kladných kořenů (včetně násobnosti) polynomu $p_n(x) = a_0 + a_1x + \dots + a_nx^n$ je roven počtu znaménkových změn v posloupnosti a_0, a_1, \dots, a_n , nebo je o sudé číslo menší.*

6.2.1 Hornerovo schéma

V dalším budeme potřebovat vyčíslení hodnoty polynomu

$$p_n(x) = a_0 + a_1x + \dots + a_nx^n$$

v daném bodě x . Pro vyčíslení polynomu použijeme Hornerovo schéma, které je efektivnější než triviální metoda vyhodnocení polynomu. Porovnejme si nyní dva způsoby vyčíslení polynomu:

1. neefektivní

```

r = 1; s = a0;
for i = 1 to n do
    r = r · x;
    s = s + ai · r;
end for

```

$p_n(x) = s$, počet násobení $2n$.

2. Hornerovo schéma

```

s = an;
for i = n - 1 downto 0 do
    s = s · x + ai;
end for

```

$p_n(x) = s$, počet násobení n .

Poznámka: 6.1. Zapišme Hornerovo schéma pro vyčíslení $p_n(z)$ takto:

```

bn = an;
for i = n - 1 downto 0 do
    bi = bi+1 · z + ai;
end for

```

$p_n(z) = b_0$.

Ukážeme, že tento zápis je vhodný pro vyčíslení derivace p'_n (a následně použijeme Newtonovu metodu pro určení kořene $p_n(x)$). Pro dělení polynomu polynomem platí

$$(a_n x^n + a_{n-1} x^{n-1} + \dots + a_0) : (x - z) = \underbrace{a_n}_{b_n} x^{n-1} + \underbrace{(a_{n-1} + a_n z)}_{b_{n-1}} x^{n-2} + \dots + b_1 + \text{zbytek}$$

$$p_n(x) = q_{n-1}(x; z)(x - z) + b_0$$

$$\text{kde } q_{n-1}(x; z) = b_n x^{n-1} + b_{n-1} x^{n-2} + \dots + b_1$$

Je-li z kořen, pak $b_0 = 0$.

Nyní aplikujeme Newtonovu metodu pro nalezení kořene polynomu p_n .

$$\text{Newtonova metoda: } x^{(k+1)} = x^{(k)} - \frac{\overbrace{p_n(x^{(k)})}^{\text{Hornerovo sch.}}}{\underbrace{p'_n(x^{(k)})}_{\text{Hornerovo sch.}}}, \quad x^{(0)} \text{ dáno}$$

Vzorec, který dostaneme s využitím Hornerova schématu, se nazývá **Newtonova-Hornerova metoda**:

$$x^{(k+1)} = x^{(k)} - \frac{p_n(x^{(k)})}{q_{n-1}(x^{(k)}; x^{(k)})}$$

Výraz ve jmenovateli dostaneme z následujících vztahů

$$\begin{aligned} p'_n(x) &= q'_{n-1}(x; z)(x - z) + q_{n-1}(x; z), \\ p'_n(z) &= q_{n-1}(z; z), \\ z &:= x^{(k)}. \end{aligned}$$

Algoritmus pro nalezení kořenů polynomu p_n :

```

for m = n downto 1 do
    Najdi kořen  $r$  polynomu  $p_m$  (Newtonova metoda)
    Vyčisli koeficienty  $q_{m-1}(x; r)$  (pomocí Hornerova schématu)
     $p_{m-1} := q_{m-1}$ 
end for

```

Poznámka: 6.2. Kvůli zaokrouhlovacím chybám je výhodné začít od kořene nejmenšího v absolutní hodnotě.

ÚKOLY

1. Implementujte metodu prosté iterace. Vstupem je počáteční odhad $x^{(0)}$, funkce $f(x)$, maximální počet iterací a velikost chyby.
2. Implementujte metodu, která na základě vět 6.4–6.6 odhadne polohy kořenů. Vstupem jsou koeficienty a_0, \dots, a_n .
3. Implementujte Newtonovu metodu pro řešení polynomiální rovnice. Vstupem jsou koeficienty a_0, \dots, a_n , maximální počet iterací a velikost chyby.

7 Řešení soustav lineárních rovnic I



CÍLE KAPITOLY

V této kapitole se budeme zabývat matematickým pozadím problematiky řešení soustav lineárních rovnic. Dále se zde budeme věnovat některým přímým metodám pro řešení soustav lineárních rovnic.



KLÍČOVÁ SLOVA

metoda přímé a iterační, vliv zaokrouhlovacích chyb, podmíněnost soustav, Gaußova eliminace

Formulace úlohy je následující, hledáme $x \in \mathbb{R}^N$ takové, že

$$\mathbb{A}x = b, \quad \mathbb{A} \in \mathbb{R}^{N \times N}, \mathbb{A}\text{-nesingulární.}$$

Protože je daná matice \mathbb{A} nesingulární, má úloha právě jedno řešení.

Pro řešení výše uvedeného problému rozlišujeme metody:

- **přímé** - konečný, předem známý počet kroků pro nalezení řešení
- **iterační** - konstruujeme (nekonečnou) posloupnost vektorů konvergujících k řešení

Definice 7.1. *Nechť X je vektorový prostor. Normou $\|\cdot\|$ rozumíme zobrazení $X \rightarrow \mathbb{R}$, které pro všechna $x, y \in X$ splňuje:*

1. $\|x\| \geq 0 \wedge \|x\| = 0 \rightarrow x = 0$, kde 0 rozumíme nulový prvek X ,
2. $\|\alpha x\| = |\alpha| \|x\|$, pro všechna $\alpha \in \mathbb{R}$,
3. $\|x + y\| \leq \|x\| + \|y\|$.

Nejčastěji používané vektorové normy v \mathbb{C}^N , $x \in \mathbb{C}^N$ jsou následující

$$\begin{aligned}\|x\|_1 &= \sum_i |x_i|, \\ \|x\|_2 &= \sqrt{\left(\sum_i |x_i|^2\right)} \quad \text{Euklidova,} \\ \|x\|_p &= \left(\sum_i |x_i|^p\right)^{\frac{1}{p}} \quad 1 \leq p < \infty, \\ \|x\|_\infty &= \max_i |x_i|.\end{aligned}$$

Maticové normy $A \in \mathbb{C}^{N \times N}$ lze definovat jedním z následujících způsobů

$$\begin{aligned}\|A\| &= \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}, \\ \|A\|_1 &= \max_j \sum_i \underbrace{|a_{ij}|}_{\text{sloupcový součet}}, \\ \|A\|_2 &= \sqrt{\rho(A^H A)} = \sqrt{\rho(AA^H)}, \\ A^H &- \text{transponovaná a kompl. združená (hermitovská),} \\ \rho(B) &- \text{největší vlastní číslo B v abs. hodnotě (spektrální poloměr),} \\ \|A\|_F &= \sqrt{\sum_{i,j} |a_{ij}|^2} \quad \text{Frobeniova,} \\ \|A\|_\infty &= \max_i \sum_j |a_{ij}| \quad \text{řádkový součet.}\end{aligned}$$

Pro vektorové a maticové normy platí

1. $\|I\|_F = \sqrt{N}$
2. $\|I\| = 1, \|Ax\| \leq \|A\| \cdot \|x\|$
3. $\|AB\| \leq \|A\| \|B\|$ sub-multiplikativita

7.1 Podmíněnost matic

Protože na počítači máme vždy pouze aproximaci vstupních hodnot (např. číslo π je reprezentováno konečným počtem desetinných míst), budeme se též věnovat vlivu zaokrouhlovacích chyb na výpočetní proces.

Malice se nazývá **dobře podmíněná**, jestliže relativně malé změny v koeficientech způsobí relativně malé změny v řešení. Matice se nazývá **špatně podmíněná**, jestliže relativně malé změny v koeficientech způsobí relativně velké změny v řešení.

Analýza zaokrouhlovacích chyb - chyby ve výpočtu se obvykle reprezentují chybami ve vstupních datech. Vzhledem k zaokrouhlovacím chybám poskytuje numerická metoda přibližné řešení, které splňuje perturbovaný systém. Numerická metoda poskytuje (přesné) řešení $x + \delta x$ perturbovaného systému

$$(A + \delta A)(x + \delta x) = b + \delta b.$$

Z výše uvedené rovnice lze chybu δx lze ("zhruba") odhadnout následujícím způsobem¹

$$\begin{aligned}\delta x &\doteq A^{-1}\delta b - A^{-1}\delta Ax, \\ \|\delta x\| &\leq \|A^{-1}\| \|\delta b\| + \|A^{-1}\| \|\delta A\| \|x\|, \\ \frac{\|\delta x\|}{\|x\|} &\leq \frac{\|A^{-1}\| \|\delta b\|}{\|x\|} + \frac{\|A^{-1}\| \|\delta A\| \|A\|}{\|A\|} \\ &\leq \frac{\|A^{-1}\| \|A\| \|x\| \|\delta b\|}{\|x\| \|b\|} + \frac{\|A^{-1}\| \|\delta A\| \|A\|}{\|A\|}\end{aligned}$$

Dostáváme pak následující nerovnost

$$\frac{\|\delta x\|}{\|x\|} \leq \underbrace{\|A\| \|A^{-1}\|}_{\text{číslo podmíněnosti } K(A)} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right)$$

¹Záměrně vynecháváme kroky mezivýpočtu, který je poměrně komplikovaný a pro nás v podstatě nedůležitý.

Závěr:

Velikost relativní chyby $\frac{\|\delta x\|}{\|x\|}$ je závislá na čísle podmíněnosti $K(A)$. Čím je větší číslo podmíněnosti $K(A)$, tím je úloha hůře podmíněná.

7.2 Gaußova eliminace

Gaußova eliminace je přímá metoda, kde je naším cílem převést soustavu $Ax = b$ na tvar $Ux = \hat{b}$, kde U je horní trojúhelníková matice. K tomu budeme nad řádky matice provádět *ekvivalentní úpravy*, tj. úpravy které nemění hodnotu matice. Jedná se o násobení řádků rozšířené matice soustavy $A|b$ nenulovým číslem, nebo jejich vzájemné sčítání. Celý postup lze popsat pomocí následujícího algoritmu

Převod na horní trojúhelníkovou matici

```

for sloupec  $j = 1$  to  $n - 1$  do
    najdi  $a_{pj} \neq 0$ ,  $p \in \{j, \dots, n\}$ 
    if  $a_{pj} = 0 \ \forall p$  then
        STOP (singularita)
    else
        záměna  $p$  a  $j$ -tého řádku
    end if
    for řádek  $i = j + 1$  to  $n$  do
         $l_{ij} = \frac{a_{ij}}{a_{jj}}$ ;
        for  $k = j + 1$  to  $n$  do
             $a_{ik} = a_{ik} - l_{ij}a_{jk}$ ;
        end for
         $b_i = b_i - l_{ij}b_j$ ;
    end for
end for

```

$u_{ij}, i \leq j$ jsou pak poslední hodnoty a_{ij}
 \hat{b}_i jsou pak poslední hodnoty b_i

Počet operací	v j -tém kroku	celkem
Hledání $a_{pj} \neq 0$	$n - j + 1$	$\sum_{j=2}^n j = \frac{(2+n)(n-1)}{2}$
Výpočet l_{ij}	$n - j$	$\sum_{j=1}^{n-1} j = \frac{n(n-1)}{2}$
Výpočet a_{ik}	$2(n - j)^2$	$2 \sum_{j=1}^{n-1} j^2 = 2 \frac{2n^3 - 3n^2 + n}{6}$
Výpočet b_i	$2(n - j)$	$2 \sum_{j=1}^{n-1} j = 2 \frac{n(n-1)}{2}$

Celkový počet operací: $\frac{2}{3}n^3 + O(n^2)$

Poté, co jsme soustavu převedli do tvaru $Ux = \hat{b}$, ji lze snadno řešit pomocí *zpětného chodu*, tj. nejprve spočteme proměnnou x_n z poslední rovnice, poté proměnnou x_{n-1} z předposlední, atd.

Počet operací pro řešení $Ux = \hat{b}$:

násobení	sčítání
$\frac{(n+1)n}{2}$	$\frac{n(n-1)}{2}$

7.2.1 Pivotace

Výpočet $l_{ij} = \frac{a_{ij}}{a_{jj}}$ v Algoritmu 7.2, $a_{jj} \neq 0$.

$$\text{Částečná pivotace} \quad |a_{pj}| = \max_{l=j, \dots, n} |a_{lj}| \quad (7.1)$$

$$\text{Úplná pivotace} \quad |a_{pj}| = \max_{l,m=j, \dots, n} |a_{lm}| \quad (7.2)$$

Důvod: I když Gaußova eliminace je proveditelná bez záměny řádků a sloupců, mohou malé hodnoty a_{jj} způsobit velké chyby v řešení.

Příkladem budiž následující úloha.

$$\begin{pmatrix} 6 & 1 & & & \\ 8 & 6 & 1 & & \\ & 8 & 6 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 8 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{50} \end{pmatrix} = \begin{pmatrix} 7 \\ 15 \\ 15 \\ \vdots \\ 14 \end{pmatrix}, \quad (7.3)$$

kde pomocí Gaußovy eliminace dostáváme

$$x_{GE} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ -3 \times 10^7 \end{pmatrix}.$$

Gaußova eliminace je **numericky nestabilní**. Pivotace je podstatná pro stabilitu elim. procesu. Ani velké hodnoty pivotů však nejsou zárukou dostatečně přesného řešení.

Důvod: velké změny v koeficientech

Náprava: škálování, dělení i -tého řádku $d_i = \sum_{j=1}^n |a_{ij}|$, ale toto dělení opět vnáší zaokrouhlovací chyby.



OTÁZKY K ZAMYŠLENÍ

1. Odhadněte počet operací pro tzv. Gaußovu–Jordanovu eliminační metodu. Tato metoda je podobná Gaußově eliminační metodě, liší se od ní v tom, že místo převodu soustavy na tvar s horní trojúhelníkovou maticí U vzniká diagonální matice. Tj. dochází k úpravám i nad diagonálou matice. Porovnejte celkový počet operací nutných pro řešení soustavy rovnic s Gaußovou eliminační metodou.
2. Rozmyslete si úpravu Gaußovy eliminační metody pro trídigonální systém, tj. pro případ kdy matice A obsahuje ke každému diagonálnímu prvku a_{ii} i prvky $a_{i-1,i}$ a $a_{i,i+1}$ (vyjma $i = 1, n$). Rozmyslete si i uložení této matice v paměti tak, aby zabírala co nejméně místa.

8 Řešení soustav lineárních rovnic II



CÍLE KAPITOLY

V této kapitole se budeme věnovat faktorizačním metodám pro řešení soustav lineárních rovnic.



KLÍČOVÁ SLOVA

Gaussova eliminace, LU rozklad, Choleského rozklad

8.1 Gaußova eliminace jako faktorizační metoda

$$Ax = b \Leftrightarrow LUx = b \quad \begin{cases} Ux = \hat{b}, \\ L\hat{b} = b. \end{cases} \quad (8.1)$$

Nechť P_j je matice, která v j -tém kroku Gaußovy eliminace realizuje záměnu p -tého a j -tého řádku matice A v Algoritmu 7.2

$$\begin{matrix} & j & & p \\ j & \begin{pmatrix} 1 & \cdot & & \cdot \\ \cdot & 0 & \cdot & \cdot & 1 & \cdot \\ & \cdot & 1 & & \cdot \\ & \cdot & & 1 & \cdot \\ p & \cdot & 1 & \cdot & \cdot & 0 & \cdot \\ & \cdot & & & \cdot & 1 \end{pmatrix} & \begin{pmatrix} \times & \times & \times & \times & \times & \times \\ \circ & \circ & \circ & \circ & \circ & \circ \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \diamond & \diamond & \diamond & \diamond & \diamond & \diamond \\ \times & \times & \times & \times & \times & \times \end{pmatrix} & = & \begin{pmatrix} \times & \times & \times & \times & \times & \times \\ \diamond & \diamond & \diamond & \diamond & \diamond & \diamond \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \circ & \circ & \circ & \circ & \circ & \circ \\ \times & \times & \times & \times & \times & \times \end{pmatrix} \end{matrix}$$

a nechť L_j je matice, pomocí níž se provádí nulování prvků j -tého sloupce pod diagonálou.

$$\begin{pmatrix} 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & -\ell_{43} & 1 & \cdot & \cdot \\ \cdot & \cdot & -\ell_{53} & \cdot & 1 & \cdot \\ \cdot & \cdot & -\ell_{63} & \cdot & \cdot & 1 \end{pmatrix} \begin{pmatrix} \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \end{pmatrix} = \begin{pmatrix} \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \end{pmatrix}$$

Algoritmus Gaußovy eliminace lze maticově zapsat (GE s částečnou pivotací):

$$\underbrace{L_{n-1}P_{n-1} \cdots L_1P_1}_M A = U.$$

Označme

$$\begin{aligned} P &= P_{n-1} \cdots P_1, \\ M &= L_{n-1} P_{n-1} \cdots L_1 P_1. \end{aligned}$$

Potom

$$\begin{aligned} MA &= U, \\ MP^{-1}PA &= U, \\ PA &= \underbrace{PM^{-1}}_L U, \\ PA &= LU. \end{aligned}$$

Lze-li provést Gaußovou eliminaci bez záměny řádků a sloupců, dostáváme

$$A = LU. \quad (8.2)$$

Věta 8.1. *Nechť $A \in \mathbb{R}^{n \times n}$, A regulární. Pak existuje permutační matice $P \in \mathbb{R}^{n \times n}$, nesingulární U a L s jedničkami na diagonále ::*

$$PA = LU \quad (8.3)$$

Algoritmus Matici L výše uvedenou dostaneme pomocí Algoritmu 7.2 tak, že l_{ij} uložíme do a_{ij} , jejichž hodnoty nejsou v Gaußově eliminaci potřeba a při pivotaci je zaměníme.

Řešení úlohy $Ax = b$ ve třech krocích

1. $PA = LU$
2. $PAx = L \underbrace{Ux}_{\hat{b}} = Pb$
 $L\hat{b} = Pb$
3. $Ux = \hat{b}$

8.2 LU rozklad v obecném případě

$$\begin{aligned} A &= \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix} && \exists! LU \text{ rozklad} \\ A &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} && \text{neexistuje } LU \text{ rozklad} \\ A &= \begin{pmatrix} 0 & 1 \\ 0 & 2 \end{pmatrix} && LU \text{ není jednoznačný} \end{aligned}$$

Při konstrukci LU rozkladu matice $A \in \mathbb{R}^{n \times n}$ postupujeme tak, že postupně počítáme m -tý řádek matice U a m -tý sloupec matice L , $m = 1, \dots, n$. Příslušné vzorce odvodíme pomocí vzorce pro násobení matic.

$$\begin{aligned} A &= LU, \\ a_{ij} &= \sum_{k=1}^n l_{ik} u_{kj}. \end{aligned}$$

Máme n^2 rovnic pro určení neznámých l_{ij} , $i \leq j$ a u_{ij} , $i \geq j$ (prvků dolní trojúhelníkové matice L a horní trojúhelníkové matice U). Počet neznámých je $2(1+n)n/2 = n^2 + n$. Předepíšeme tedy hodnoty některých prvků, například položíme diagonální prvky matice L rovny jedné. Dostáváme následující vzorce pro $m = 1, \dots, n$:

m -tý řádek matice U , u_{mj} , $j \geq m$ (křížky označují již spočtené hodnoty, počítáme prvek \diamond):

$$\underbrace{m \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \boxed{\times} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}}_A = \underbrace{m \begin{pmatrix} 1 & & & & & \\ \times & 1 & & & & \\ \times & \times & 1 & & & \\ \times & \times & \cdot & 1 & & \\ \times & \times & \cdot & \cdot & 1 & \\ \times & \times & \cdot & \cdot & \cdot & 1 \end{pmatrix}}_L \underbrace{\begin{pmatrix} \times & \times & \times & \times & \times & \times \\ & \times & \times & \times & \times & \times \\ & & \diamond & \rightarrow & \cdot & \cdot \\ & & & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot \\ & & & & & \cdot \end{pmatrix}}_U$$

$$a_{mj} = \sum_{k=1}^n l_{mk} u_{kj} = \sum_{k=1}^{m-1} l_{mk} u_{kj} + 1 \cdot u_{mj}, \quad u_{mj} = \diamond,$$

m -tý sloupec matice L , l_{im} , $i > m$:

$$\underbrace{i \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \boxed{\times} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}}_A = \underbrace{i \begin{pmatrix} 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & 1 & \cdot & \cdot & \cdot & \cdot \\ \times & \times & 1 & \cdot & \cdot & \cdot \\ \times & \times & \circ & 1 & \cdot & \cdot \\ \times & \times & \downarrow & \cdot & 1 & \cdot \\ \times & \times & \cdot & \cdot & \cdot & 1 \end{pmatrix}}_L \underbrace{\begin{pmatrix} \times & \times & \times & \times & \times & \times \\ \cdot & \times & \times & \times & \times & \times \\ \cdot & \cdot & \diamond & \diamond & \diamond & \diamond \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}}_U$$

$$a_{im} = \sum_{k=1}^n l_{ik} u_{km} = \sum_{k=1}^{m-1} l_{ik} u_{km} + l_{im} \cdot u_{mm}, \quad l_{im} = \circ.$$

Věta 8.2. *Nechť $A \in \mathbb{R}^{n \times n}$ je obecná matice. Faktorizace $A = LU$ existuje a je jednoznačná právě když všechny hlavní minory A , t.j. $\det \begin{pmatrix} a_{11} & \cdots & a_{1k} \\ & \ddots & \\ a_{k1} & \cdots & a_{kk} \end{pmatrix}$, $k = 1, \dots, n-1$ jsou nenulové.*

Věta 8.3. *Je-li matice řádkově nebo sloupcově diagonálně dominantní, t.j.*

$$|a_{ii}| \geq \sum_{j=1, j \neq i}^n |a_{ij}|, \quad (\text{řádkově}) \quad (8.1)$$

nebo

$$|a_{jj}| \geq \sum_{i=1, i \neq j}^n |a_{ij}|, \quad (\text{sloupcově}) \quad (8.2)$$

pak LU rozklad existuje. Speciálně, je-li matice sloupcově diagonálně dominantní, je $|l_{ij}| \leq 1 \forall i, j = 1, \dots, n$.

8.3 Choleského rozklad

Věta 8.4. Pro každou symetrickou, pozitivně definitní ($x^T A x > 0$, $\forall x \neq 0$, $x \in \mathbb{R}^n$) matici $A \in \mathbb{R}^{n \times n}$ existuje právě jedna dolní trojúhelníková matice L s kladnými prvky na diagonále tak, že platí

$$A = L \cdot L^T \quad (8.3)$$

Důkaz. indukcí □

Věta 8.5. Nechť $A \in \mathbb{R}^{n \times n}$ je symetrická, ostře diag. dominantní ($|a_{ii}| > \sum_{j \neq i} |a_{ij}|$), $a_{ii} > 0$, pak A je pozitivně definitní.

Poznámka: 8.1. QR rozklad

Existují i jiné typy faktorizace matic. Např. ke každé nesingulární matici $A \in \mathbb{R}^{n \times n}$ existuje ortogonální matice $Q \in \mathbb{R}^{n \times n}$ ($QQ^T = Q^T Q = I$) a nesingulární horní trojúhelníková R taková, že

$$A = Q \cdot R. \quad (8.4)$$

Opět je vidět, že tuto faktorizaci lze použít pro řešení soustavy lineárních rovnic. Konstrukce QR rozkladu však přesahuje rámec těchto skript. Poznamenejme jen, že běžně užívaný software pro numerické výpočty jej implementuje.

ÚKOLY

1. Implementujte Gaußovu eliminační metodu a LU rozklad pro řešení soustavy lineárních rovnic. Vstupem je matice A a vektor pravých stran b .
2. Implementujte program pro výpočet determinantu matice, použijte buď LU rozklad, nebo Gaußovu eliminační metodu. Vstupem je matice A .
3. Pomocí implementovaných programů řešte úlohu (7.3) a porovnejte s přesným řešením. Prozkoumejte i vliv pivotace.
4. Experimentálně ověřte časovou náročnost GEM a LU rozkladu pro různé velikosti vstupních úloh $n = 10, \dots, 10000$.

9 Řešení soustav lineárních rovnic III



CÍLE KAPITOLY

V této kapitole se budeme věnovat iteračním metodám pro řešení soustav lineárních rovnic.



KLÍČOVÁ SLOVA

iterační metody, Jacobiho iterační metoda, Gaußova–Seidelova iterační metoda, SOR metoda

Formulace úlohy je stejná jako v předchozí kapitole, tedy hledáme $x \in \mathbb{R}^N$ takové, že

$$\mathbb{A}x = b, \quad \mathbb{A} \in \mathbb{R}^{N \times N}, \mathbb{A}\text{-nesingulární.}$$

Protože je daná matice \mathbb{A} nesingulární, má úloha právě jedno řešení.

Idea iteračních metod: konstrukce $\{x^{(k)}\}$

$$x = \lim_{k \rightarrow \infty} x^{(k)}, \quad \text{kde } x \text{ je řešení } Ax=b. \quad (9.1)$$

Poznámka: 9.1. Posloupnost $x^{(0)}, x^{(1)}, \dots, x^{(k)}, \dots$, je nekonečná, cílem je nalezení řešení x^* s předepsanou přesností, t.j. $\|x^* - x^{(k)}\| \leq \varepsilon$. Otázkou je určení vhodného stopping kritéria (např. omezenost rezidua $\|b - Ax^{(k)}\| \leq \varepsilon$).

Princip iteračních metod je na základě předchozích aproximací konstrukce nové aproximace

$$x^{(k+1)} = \varphi(x^{(k)}), \quad \text{resp. } x^{(k+1)} = \varphi(x^{(k)}, x^{(k-1)}, \dots, x^{(0)})$$

takové, že

$$x^{(k+1)} \rightarrow x \text{ pro } k \rightarrow \infty,$$

kde x je hledané řešení. Požadavky:

- rychlá konvergence
- snadné vyčíslení φ (méně operací než matice \times vektor, řádově $O(n)$)
- řešení s předepsanou přesností

Základní myšlenkou klasických metod je konstrukce zobrazení G tak, abychom mohli aplikovat větu o pevném bodě 6.2

$$Ax = b \Leftrightarrow x = G(x). \quad (9.2)$$

Pro dané $x^{(0)}$ se řešení hledá jako limita posloupnosti $x^{(k+1)} = G(x^{(k)})$. V dalším textu si nejprve ukážeme konstrukci daného zobrazení G pro různé metody a poté si ukážeme, kdy je vůbec G kontrahující a tím daný proces konvergentní.

9.1 Odvození klasických iteračních metod

9.1.1 Richardsonova metoda

Nejprve si zkonstruujeme zobrazení G

$$\begin{aligned}x &= x + b - Ax, \\x &= \underbrace{(I - A)}_{B_R} x + b,\end{aligned}$$

toto nás vede k volbě iteračního schématu ve tvaru

$$x^{(k+1)} = B_R x^{(k)} + b. \quad (9.3)$$

9.1.2 Jacobiho metoda

Vyjádřeme A ve tvaru

$$A = E + D + F, \quad \text{kde}$$

E je ostře dolní trojúhelníková,

D je diagonální,

F je ostře horní trojúhelníková.

$$\begin{aligned}Ax = b &\iff (E + D + F)x = b, \\Dx &= -(E + F)x + b, \\x &= \underbrace{-D^{-1}(E + F)x}_{B_J} + \underbrace{D^{-1}b}_{f_J},\end{aligned}$$

toto nás vede k volbě iteračního schématu ve tvaru

$$x^{(k+1)} = B_J x^{(k)} + f_J. \quad (9.4)$$

9.1.3 Gaußova–Seidelova metoda

Opět si rozložme matici A

$$\begin{aligned}Ax = b &\iff (D + E)x + Fx = b, \\(D + E)x &= -Fx + b, \\x &= \underbrace{-(D + E)^{-1}F}_{B_{GS}} x + \underbrace{(D + E)^{-1}b}_{f_{GS}},\end{aligned}$$

což vede k iteračnímu schématu

$$x^{(k+1)} = B_{GS} x^{(k)} + f_{GS}. \quad (9.5)$$

Poznámka: 9.2. Porovnejme způsob algoritmizace Jacobiho a Gaußovy-Seidelovy metody. K tomu je třeba si nejprve uvědomit, že

$$D^{-1} = \begin{pmatrix} \frac{1}{a_{11}} & & & & \\ & \frac{1}{a_{22}} & & & \\ & & \frac{1}{a_{33}} & & \\ & & & \ddots & \\ & & & & \ddots & \\ & & & & & \frac{1}{a_{nn}} \end{pmatrix},$$

$$D^{-1}X = \begin{pmatrix} \frac{1}{a_{11}} \times & \frac{1}{a_{11}} \times & \frac{1}{a_{11}} \times & \frac{1}{a_{11}} \times & \frac{1}{a_{11}} \times & \frac{1}{a_{11}} \times \\ \frac{1}{a_{22}} \times & \frac{1}{a_{22}} \times & \frac{1}{a_{22}} \times & \frac{1}{a_{22}} \times & \frac{1}{a_{22}} \times & \frac{1}{a_{22}} \times \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{a_{nn}} \times & \frac{1}{a_{nn}} \times & \frac{1}{a_{nn}} \times & \frac{1}{a_{nn}} \times & \frac{1}{a_{nn}} \times & \frac{1}{a_{nn}} \times \end{pmatrix}.$$

Vyčíslení inverzní matice v Gaußově-Seidelově metodě se vyhneme následujícím způsobem. Na základě vyjádření

$$x^{(k+1)} = \underbrace{-(D+E)^{-1}F}_{B_{GS}} x^{(k)} + \underbrace{(D+E)^{-1}b}_{f_{GS}}$$

přepíšeme Gaußovu-Seidelovu metodu ve tvaru

$$\begin{aligned} (D+E)x^{(k+1)} &= -Fx^{(k)} + b, \\ Dx^{(k+1)} &= -Ex^{(k+1)} - Fx^{(k)} + b, \\ x^{(k+1)} &= -D^{-1}Ex^{(k+1)} - D^{-1}Fx^{(k)} + D^{-1}b, \end{aligned}$$

$$x^{(k+1)} = -D^{-1}E \quad x^{(k+1)} - D^{-1}F \quad x^{(k)} + \underbrace{D^{-1}b}_{f_J}, \quad (9.6)$$

t.j.

$$\begin{pmatrix} x^{(k+1)} \\ \times \\ \times \\ \times \\ \times \end{pmatrix} = - \begin{pmatrix} \cdot & & & \\ \times & \cdot & & \\ \times & \times & \cdot & \\ \times & \times & \times & \cdot \end{pmatrix} \begin{pmatrix} x^{(k+1)} \\ \times \\ \times \\ \times \\ \times \end{pmatrix} - \begin{pmatrix} \cdot & \times & \times & \times \\ & \cdot & \times & \times \\ & & \cdot & \times \\ & & & \cdot \end{pmatrix} \begin{pmatrix} x^{(k)} \\ \times \\ \times \\ \times \\ \times \end{pmatrix} + \begin{pmatrix} \times \\ \times \\ \times \\ \times \\ \times \end{pmatrix}.$$

Při použití Jacobiho metody

$$x^{(k+1)} = B_J x^{(k)} + f_J,$$

t.j.

$$\begin{matrix} x^{k+1} \\ \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} \end{matrix} = - \begin{matrix} \\ \begin{pmatrix} \cdot & \times & \times & \times \\ \times & \cdot & \times & \times \\ \times & \times & \cdot & \times \\ \times & \times & \times & \cdot \end{pmatrix} \end{matrix} \begin{matrix} x^k \\ \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} \end{matrix} + \begin{matrix} f_J \\ \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} \end{matrix},$$

je třeba si pamatovat celý vektor $x^{(k)}$ pro výpočet nové iterace $x^{(k+1)}$. U metody Gaußovy-Seidelovy se v paměti počítače rezervuje místo pro jediný vektor $x^{(k)}$, na jehož místo se postupně ukládají složky vektoru $x^{(k+1)}$ jak vyplývá z rozepsání po složkách vztahu (9.6):

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

Dostáváme tak algoritmus **Gaußovy-Seidelovy metody**, který lze vyjádřit následujícím způsobem. Vydeme z Jacobiho metody a spočtenou složku $x_i^{(k+1)}$ uložíme do $x_i^{(k)}$ a následně počítáme $x_{i+1}^{(k+1)}$, $i = 1, \dots, n$.

$$i+1 \begin{matrix} x^{k+1} \\ \begin{pmatrix} \boxed{\times} \\ \boxed{\times} \\ \times \\ \times \end{pmatrix} \end{matrix} = - \begin{matrix} \\ \begin{pmatrix} \cdot & \times & \times & \times \\ \times & \cdot & \times & \times \\ \times & \times & \cdot & \times \\ \times & \times & \times & \cdot \end{pmatrix} \end{matrix} \begin{matrix} x^k \\ \begin{pmatrix} \boxed{\times} \\ \times \\ \times \\ \times \end{pmatrix} \end{matrix} + \begin{matrix} f_J \\ \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} \end{matrix}.$$

9.1.4 SOR (superrelaxační)

Základní myšlenkou superrelaxační metody (successive over-relaxation) je ovlivnit možnost a rychlost konvergence metody pomocí jistého parametru ω . Metoda se skládá ze dvou základních kroků. Nejprve se udělá krok pomocí Gaußovy-Seidelovy metody

$$\tilde{x}^{(k+1)} = -D^{-1}Ex^{(k+1)} - D^{-1}Fx^{(k)} + D^{-1}b, \quad (9.7)$$

poté se udělá následující lineární kombinace řešení $\tilde{x}^{(k+1)}$ a $x^{(k)}$

$$x^{(k+1)} = x^{(k)} + \omega(\tilde{x}^{(k+1)} - x^{(k)}). \quad (9.8)$$

Což za použití rozkladu $Ax = (E + D + F)x = b$ a dosazením (9.7) do (9.8) a po úpravě

$$\begin{aligned} x^{(k+1)} &= (1 - \omega)x^{(k)} + \omega\tilde{x}^{(k+1)}, \\ x^{(k+1)} &= -\omega D^{-1}Ex^{(k+1)} + \left[(1 - \omega)I - \omega D^{-1}F\right]x^{(k)} + \omega D^{-1}b, \\ x^{(k+1)} &= \left(D^{-1}ID + \omega D^{-1}E\right)^{-1} \left[(1 - \omega)D^{-1}ID - \omega D^{-1}F\right]x^{(k)} \\ &\quad + (D^{-1}ID + \omega D^{-1}E)^{-1}\omega D^{-1}b, \\ x^{(k+1)} &= \underbrace{(D + \omega E)^{-1}[(1 - \omega)D - \omega F]}_{B_{SOR}} x^{(k)} \\ &\quad + \underbrace{(D + \omega E)^{-1}\omega b}_{f_{SOR}}. \end{aligned}$$

vede na iterační předpis

$$x^{(k+1)} = B_{SOR}x^{(k)} + f_{SOR}.$$

Pro výpočty pomocí výše uvedených metod se používá jejich zápis do složek:

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

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

$$\tilde{x}_i^{(k+1)} = \frac{1}{a_{ii}} \left(-\sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right) + \frac{b_i}{a_{ii}}, \quad (\text{SOR})$$

$$x_i^{(k+1)} = x_i^{(k)} + \omega(\tilde{x}_i^{(k+1)} - x_i^{(k)}).$$

9.2 Problematika konvergence iteračních metod

Uvažujme iterační metodu

$$x^{(k+1)} = Bx^{(k)} + f. \quad (9.9)$$

Definice 9.1. Řekneme, že iterační metoda $x^{(k+1)} = Bx^{(k)} + f$ je konzistentní s $Ax = b$, jestliže

$$x = Bx + f, \text{ kde } x \text{ je řešení úlohy } Ax = b.$$

Ekvivalentně

$$f = (I - B)x = (I - B)A^{-1}b.$$

Věta 9.2. Nechť $x^{(k+1)} = Bx^{(k)} + f$ je konzistentní metoda. Pak posloupnost $\{x^{(k)}\}$ konverguje k x^* , kde x^* splňuje $Ax^* = b$, **pro libovolné** $x^{(0)}$, **právě když** $\rho(B)$ (spektrální poloměr matice B , $\rho(B) = \max_{\lambda \text{ vl. č. } B} |\lambda|$) je menší než 1.

Důkaz.

$$\begin{aligned} x^* &= Bx^* + f \quad (\text{podmínka konzistence}), \\ x^{(k+1)} &= Bx^{(k)} + f \quad (\text{iterační metoda}), \\ e^{(k+1)} &:= x^* - x^{(k+1)} \quad (\text{chyba v } (k+1)\text{-ní iteraci}). \end{aligned}$$

Chybu v k -té iteraci lze vyjádřit jako součin k -té **mocniny** matice B a chyby počáteční aproximace

$$e^{(k)} = Be^{(k-1)} = B^2e^{(k-2)} = \dots = B^ke^{(0)}.$$

Podle definice limity

$$x^{(k)} \rightarrow x^* \Leftrightarrow \|e^{(k)}\| \rightarrow 0.$$

Platí

$$\|e^{(k)}\| \rightarrow 0 \Leftrightarrow \|B^ke^{(0)}\| \rightarrow 0 \Leftrightarrow \rho(B) < 1.$$

Poslední ekvivalenci dokážeme na základě následující věty z algebry. Vyhnete se tak klasickému důkazu pomocí převedení matice B na **Jordanův** kanonický tvar.

Lemma 9.1. *Nechť $A \in \mathbb{C}^{n \times n}$, $\varepsilon > 0$. Pak existuje konzistentní ($\|Ax\| \leq \|A\| \|x\|$) maticová norma $\|\cdot\|_{A,\varepsilon}$ taková, že*

$$\|A\|_{A,\varepsilon} \leq \rho(A) + \varepsilon.$$

Pokračování v důkazu předchozí věty

\Leftarrow Nechť $\rho(B) < 1$. Potom $\exists \varepsilon :: \rho(B) < 1 - \varepsilon$ a dále existuje $\|\cdot\|_{B,\varepsilon} ::$

$$\|B\|_{B,\varepsilon} \leq \rho(B) + \varepsilon < 1$$

a tedy

$$\|B^k\|_{B,\varepsilon} \leq \|B\|_{B,\varepsilon}^k \rightarrow 0.$$

Platí tedy

$$\|e^{(k)}\| = \|B^k e^{(0)}\| \leq \|B^k\| \|e^{(0)}\| \leq \|B\|^k \|e^{(0)}\| \rightarrow 0.$$

\Rightarrow Předpokládejme (sporem), že $\rho(B) > 1$. Existuje tedy vlastní číslo λ matice B (viz definice spektrálního poloměru výše) takové, že $|\lambda| > 1$. Zvolme počáteční aproximaci $x^{(0)}$ tak, že $e^{(0)}$ je vlastní vektor odpovídající vlastnímu číslu λ . Potom

$$e^{(k)} = B^k e^{(0)} = \lambda^k e^{(0)}.$$

V důsledku tohoto vztahu $e^{(k)}$ **nekonverguje k nule** (pro danou volbu $x^{(0)}$), protože $|\lambda| > 1$.

□

Poznámka: 9.3. Lemma 9.1 jsme využili pro důkaz vztahu $\rho(B) < 1 \Rightarrow B^k \rightarrow 0$. Obrácená implikace se dokáže snadno.

V praxi je možné použít k ověření konvergence spíše věty hovořící o matici soustavy A , nikoliv iterační matici B . Uvedme si některé z nich.

Věta 9.3. *Jacobiova a Gaußova–Seidelova iterační metoda v případě ostře diagonálně dominantní matice A konvergují pro libovolné $x^{(0)}$.*

Věta 9.4. *Gaußova–Seidelova iterační metoda v případě pozitivně definitní matice A konverguje pro libovolné $x^{(0)}$.*

Věta 9.5. *Nechť A je symetrická matice s kladnými prvky na diagonále. Pak spektrální poloměr $\rho(B_{SOR}) < 1$, právě když A je pozitivně definitní a $\omega \in (0, 2)$. V tomto případě metoda SOR konverguje.*

Poznámka: 9.4. Při srovnání rychlosti konvergence Jacobiovy a Gaußovy–Seidelovy metody platí, že většinou druhá zmiňovaná konverguje rychleji.

Na závěr si porovnejme přímé metody z předchozí kapitoly s metodami iteračními. **Přímé metody** (např. Gaußova eliminace) :

- pro libovolné plné matice
- počet operací $O(\frac{2}{3}n^3)$

Nevýhoda:

- a) nevyužívají informaci o struktuře matice (řídkost, blokově diagonální)

- b) nákladné, je-li n velké
- c) pro řídké matice mohou být nevhodné (zaplnění)

Iterační metody

- formálně poskytují řešení po nekonečném počtu kroků
- v každém kroku požadují výpočet rezidua $r = b - Ax$, výpočetní náročnost $O(n^2)$
- mohou soupeřit s přímými metodami, je-li počet iterací k získání řešení s danou tolerancí nezávislý na n nebo menší než n
- používají se, stačí-li získat řešení pouze s určitou přesností (Fyzika \rightarrow model \rightarrow matematický model)



OTÁZKY K ZAMYŠLENÍ

1. Dokažte větu 9.3 pro Jacobiho metodu. Vyjděte z konstrukce iterační matice B_J Jacobiho metody a použijte větu 9.2 a lemma 2.1.
2. Pomocí věty 9.5 dokažte větu 9.4. Uvědomte si vztah mezi Gaußovou–Seidelovou metodou a SOR.



ÚKOLY

1. Implementujte výše uvedené iterační metody pro řešení soustavy lineárních rovnic. Vstupem je matice A a vektor pravých stran b , počáteční iterace $x^{(0)}$, maximální počet iterací, případně chyba metody (v případě SOR i ω).
2. Experimentálně ověřte časovou náročnost implementovaných metod pro různé velikosti vstupních úloh $n = 10, \dots, 10000$ a porovnejte s metodami z předchozí kapitoly. Musíte ovšem generovat vstupní matice tak, aby byly zaručeny podmínky konvergence.

10 Řešení soustav lineárních rovnic IV



CÍLE KAPITOLY

V této kapitole se budeme zabývat principy moderních metod pro řešení soustav lineárních rovnic. Konkrétně se zaměříme na gradientní metody. Tuto kapitolu bereme jako rozšiřující.



KLÍČOVÁ SLOVA

gradientní metody, metoda největšího spádu

Gradientní metody jsou metody používané v optimalizaci pro hledání extrémů funkcí více proměnných. Zde je použijeme pro řešení soustav lineárních rovnic. Hledejme extrém následující funkce

$$\Phi(x) = \frac{1}{2}x^T Ax - x^T b, \quad (10.1)$$

kde $A = A^T \in R^{n \times n}$ je symetrická pozitivně definitní matice (SPD), $x, b \in R^n$. Nutná podmínka pro extrém této funkce je, aby

$$\nabla \Phi(x) = 0 \quad (10.2)$$

- Lze snadno dokázat, že tento extrém je lokálním minimem funkce Φ (ověřte).
- Ověřte, že podmínka $\nabla \Phi(x) = 0 \Rightarrow Ax - b = 0 \Rightarrow Ax = b$.

Z předchozího vyplývá, že místo hledání řešení rovnice $Ax = b$, stačí nalézt minimum funkce Φ . Ovažujme reziduum $r = b - Ax$, je vidět, že reziduum odpovídá záporně vzatému gradientu. Použijme následující obecné iterační schéma pro *spádové metody*

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}, \quad (10.3)$$

kde $d^{(k)} \in R^n$ představuje směr posunu a $\alpha^{(k)} \in R$ je velikost kroku.

V dalším zvolme

$$d^{(k)} = r^{(k)} = b - Ax^{(k)}, \quad (10.4)$$

protože reziduum r odpovídá gradientu, tak pokud dojde k situaci $|x^{(k+1)} - x^{(k)}| \rightarrow 0$ pak i $\nabla \Phi(x^{(k)}) \rightarrow 0$, čili bylo nalezeno minimum Φ , tedy řešení $Ax = b$.

Volbu optimální délky kroku získáme řešením rovnice

$$\frac{d\Phi(x^{(k+1)})}{d\alpha} = 0, \quad (10.5)$$

čili minimalizujeme funkci Φ ve směru $d^{(k)}$. Řešením je následující vztah (ověřte)

$$\alpha^{(k)} = \frac{r^{(k)T} r^{(k)}}{r^{(k)T} A r^{(k)}}. \quad (10.6)$$

Výsledný algoritmus *gradientní metody* spočívá v iterování pomocí schématu (10.3) za použití (10.4) a (10.6).

Konvergence tohoto procesu není moc rychlá a závisí na poměru dvou největších (v absolutní hodnotě) vlastních čísel matice A

Konvergenci lze podstatně urychlit volbou jiných směrů než je gradient. Příkladem je *metoda sdružených gradientů*, jejíž popis je možné nalézt v doporučené literatuře uvedené v závěru textu.



OTÁZKY K ZAMYŠLENÍ

1. Odvoďte (dokažte) v textu naznačená tvrzení.
2. Prostudujte si základní principy metody sdružených gradientů



ÚKOLY

1. Implementujte gradientní metodu a metodu sdružených gradientů. Vstupem je matice A a vektor pravých stran b , počáteční iterace $x^{(0)}$, maximální počet iterací, případně chyba metody.

11 Výpočet vlastních čísel matice



CÍLE KAPITOLY

V této kapitole se budeme věnovat metodám pro výpočet vlastních čísel matic. Konkrétně se zaměříme na problém dominantního vlastního čísla matice.



KLÍČOVÁ SLOVA

vlastní čísla, mocninná metoda

Formulace problému je následující. Buď $A \in \mathbb{C}^{n \times n}$, hledáme $\lambda \in \mathbb{C} :: \exists x \in \mathbb{C}^n, x \neq 0$,

$$Ax = \lambda x. \quad (11.1)$$

Zaměříme se na nalezení dominantního vlastního čísla, tj. vlastního čísla největšího v absolutní hodnotě. Poznamenejme, že znalost vlastních čísel matic je důležitá např. pro vyšetřování konvergence iteračních metod pro lineární soustavy.

Použití klasického způsobu, tj. řešení problému

$$\det(A - I\lambda) = 0, \quad (11.2)$$

je v případě větších matic ($n > 3$) komplikované a výpočetně náročné.

11.1 Mocninná metoda

V dalším předpokládejme, že $A \in \mathbb{C}^{n \times n}$ je diagonalizovatelná, tedy platí

$$A = X\Lambda X^{-1}, \quad X = \begin{pmatrix} \vdots & & \vdots \\ x_1 & \cdots & x_n \\ \vdots & & \vdots \end{pmatrix} \in \mathbb{C}^{n \times n},$$

kde x_i jsou vlastní vektory ($Ax_i = \lambda_i x_i$). Obecně je určen pouze směr vlastních vektorů, nikoliv jejich velikost, můžeme je tedy znormovat tak, aby platilo $\|x_i\| = 1$.

Nechť $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots |\lambda_n|$, λ_1 má násobnost 1. Pak λ_1 nazveme dominantním vlastním číslem.

Nechť je dáno $q^{(0)} \in \mathbb{C}^n$, $\|q^{(0)}\| = 1$ ($\|\cdot\| := \|\cdot\|_2$ - Euklidovská). Konstruuje posloupnost vektorů

$$q^{(k)} = \frac{Aq^{(k-1)}}{\|Aq^{(k-1)}\|} = \cdots = \frac{A^k q^{(0)}}{\|A^k q^{(0)}\|} \quad (\text{odtud název mocninná metoda}). \quad (11.3)$$

Je-li A diagonalizovatelná, má matice X za sloupce vlastní vektory matice A . Tyto vlastní vektory jsou lineárně nezávislé a tvoří bázi \mathbb{C}^n . Lze tedy psát:

$$q^{(0)} = \sum_{i=1}^n \alpha_i x_i, \quad \alpha_i \in \mathbb{C}, \quad i = 1, \dots, n. \quad (11.4)$$

Budeme uvažovat takové $q^{(0)}$, pro které $\alpha_1 \neq 0$. Jinak bychom v dalším postupu narazili na problém dělení nulou. Do rovnice 11.3 dosadíme vyjádření z rovnice 11.4 a využijeme vztahu $Ax_i = \lambda_i x_i$ pro $i = 1, \dots, n$ (resp. vztahu $A^k x_i = \lambda_i^k x_i$, který z něj přímo vyplývá)

$$q^{(k)} = \frac{A^k q^{(0)}}{\|A^k q^{(0)}\|} = \frac{A^k \sum_{i=1}^n \alpha_i x_i}{\|A^k \sum_{i=1}^n \alpha_i x_i\|} = \frac{\sum_{i=1}^n \alpha_i A^k x_i}{\|\sum_{i=1}^n \alpha_i A^k x_i\|} = \frac{\sum_{i=1}^n \alpha_i \lambda_i^k x_i}{\|\sum_{i=1}^n \alpha_i \lambda_i^k x_i\|}.$$

Dále vytkneme $\alpha_1 \lambda_1^k$ a dostaneme

$$q^{(k)} = \frac{\alpha_1 \lambda_1^k (x_1 + y^{(k)})}{|\alpha_1 \lambda_1^k| \|x_1 + y^{(k)}\|}$$

kde $(\sum_{i=1}^n \alpha_i \lambda_i^k x_i = \alpha_1 \lambda_1^k (x_1 + \frac{\alpha_2 \lambda_2^k}{\alpha_1 \lambda_1^k} x_2 + \dots + \frac{\alpha_n \lambda_n^k}{\alpha_1 \lambda_1^k} x_n))$

$$y^{(k)} = \sum_{i=2}^n \frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k x_i$$

a v důsledku předpokladu, že λ_1 je dominantní vlastní číslo matice A a $\|x_i\| = 1$,

$$\|y^{(k)}\| = \left\| \sum_{i=2}^n \frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k x_i \right\| \leq \underbrace{\sum_{i=2}^n \left| \frac{\alpha_i}{\alpha_1} \right| \left| \frac{\lambda_i}{\lambda_1} \right|^k}_C \leq C \left| \frac{\lambda_2}{\lambda_1} \right|^k.$$

Odtud dostáváme

$$y^{(k)} \rightarrow 0 \quad \text{pro } k \rightarrow \infty.$$

Pro $k \rightarrow \infty$ se tedy směr $q^{(k)}$ bude blížit směru x_1 . O rychlosti konvergence rozhoduje podíl $|\lambda_2/\lambda_1|$.

Uvažujme $Aq^{(k)}$ a $q^{(k)H} Aq^{(k)}$ ($x^H = \bar{x}^T$). Ukážeme, že

$$q^{(k)H} Aq^{(k)} \rightarrow \lambda_1 \quad \text{pro } k \rightarrow \infty. \quad (11.5)$$

$$\begin{aligned} q^{(k)H} &= \frac{\alpha_1 \lambda_1^k (x_1 + y^{(k)})^H}{|\alpha_1 \lambda_1^k| \|x_1 + y^{(k)}\|}, \quad Aq^{(k)} = \frac{\alpha_1 \lambda_1^k (\lambda_1 x_1 + Ay^{(k)})}{|\alpha_1 \lambda_1^k| \|x_1 + y^{(k)}\|}, \\ q^{(k)H} Aq^{(k)} &= \frac{(\alpha_1 \lambda_1^k)^2 (\lambda_1 + x_1^H Ay^{(k)} + \lambda_1 y^{(k)H} x_1 + y^{(k)H} Ay^{(k)})}{|\alpha_1 \lambda_1^k|^2 \|x_1 + y^{(k)}\|^2} \rightarrow \lambda_1, \end{aligned}$$

kde jsme využili toho, že $x_1^H x_1 = 1$.

Dále ukážeme, že

$$q^{(k)} \rightarrow x_1. \quad (11.6)$$

K tomu uvažujme

$$\begin{aligned} \mathbf{A}q^{(k)} - \lambda_1 q^{(k)} &= \frac{\alpha_1 \lambda_1^k (\lambda_1 x_1 + \mathbf{A}y^{(k)})}{|\alpha_1 \lambda_1^k| \|x_1 + y^{(k)}\|} - \frac{\alpha_1 \lambda_1^k (\lambda_1 x_1 + \lambda_1 y^{(k)})}{|\alpha_1 \lambda_1^k| \|x_1 + y^{(k)}\|} \\ &= \frac{\alpha_1 \lambda_1^k (\mathbf{A}y^{(k)} - \lambda_1 y^{(k)})}{|\alpha_1 \lambda_1^k| \|x_1 + y^{(k)}\|} \rightarrow 0 \end{aligned}$$

v důsledku toho, že $y^{(k)} \rightarrow 0$ pro $k \rightarrow \infty$.

Poznámka: 11.1. Výpočet tedy probíhá tak, že zvolíme počáteční $q^{(0)}$ a pomocí schématu (11.3) generujeme posloupnost $q^{(k)}$. Na základě vztahu (11.5) vidíme, že $q^{(k)H} \mathbf{A}q^{(k)}$ konverguje k vlastnímu číslu λ_1 a navíc (na základě (11.6)) $q^{(k)}$ konverguje k vlastnímu vektoru x_1 . Výpočet je zastaven, pokud $\|q^{(k)} - q^{(k-1)}\|$ je menší než předepsaná odchylka. Pokud nebylo $q^{(0)}$ zvoleno tak, aby platilo $\alpha_1 \neq 0$ může dojít k dělení nulou. V tom případě zvolíme jiné počáteční $q^{(0)}$ a výpočet opakujeme.

Pokud potřebujeme další vlastní čísla $(\lambda_2, \dots, \lambda_n)$ symetrické matice $A = A^T \in R^{n \times n}$, můžeme ze znalosti vlastního čísla λ_1 a vektoru x_1 snadno sestavit jinou symetrickou matici W , která má vlastní čísla $0, \lambda_2, \dots, \lambda_n$ a vlastní vektory x_1, \dots, x_n . Tuto konstrukci nazýváme *Hotellingova redukce* matice

$$W = A - \lambda_1 x_1 x_1^T. \quad (11.7)$$

Na matici W můžeme opět aplikovat mocninou metodu a získat další vlastní čísla a vektory.

! ÚKOLY

1. Implementujte algoritmus mocninné metody spolu s Hotellingovou redukcí. Vstupem je matice A a počáteční $q^{(0)}$, chyba a maximální počet iterací.

12 Numerická řešení obyčejných diferenciálních rovnic



CÍLE KAPITOLY

V této kapitole se budeme zabývat řešením obyčejných diferenciálních rovnic. Zaměříme se jen na základní metody numerického řešení.



KLÍČOVÁ SLOVA

obyčejné diferenciální rovnice, jednokrokové metody, Eulerova metoda, metody Rungeho–Kutty

Formulace problému je následující. Budiž dáno $f : [a, b] \times \mathbb{R} \rightarrow \mathbb{R}$, $f = f(x, y)$, $x \in [a, b]$, $y \in \mathbb{R}$. Dále je dána tzv. počáteční podmínka $\eta \in \mathbb{R}$. Hledáme zobrazení $y : [a, b] \rightarrow \mathbb{R}$ splňující

$$\begin{aligned}y'(x) &= f(x, y(x)), & x \in [a, b], \\y(a) &= \eta.\end{aligned}\tag{12.1}$$

O funkci f předpokládáme, že je spojitá a dále předpokládáme, že f je (lokálně) lipschitzovská v druhé proměnné, což nám (na základě vět matematické analýzy) zaručí (lokální) existenci a jednoznačnost řešení.

Za splnění výše uvedených předpokladů, lze problém (12.1) převést na ekvivalentní formulaci řešení následující integrální rovnice

$$y(x) = y(a) + \int_a^x f(t, y(t)) dt\tag{12.2}$$

V případě numerických metod je řešení diskrétní, tj. pro každé x_i získáme hodnotu $y(x_i)$, resp. její aproximaci, čímž se lišíme od řešení získaného klasickým řešením (viz matematická analýza), kde řešením je funkce $y(x)$.

Ukažme si některé principy odvození numerických metod pro ODR. Uvažujme dělení intervalu $[a, b]$ s uzly $x_i = a + ih$, $i = 0, \dots, n$ (s konstantním krokem, obecně lze uvažovat nekonstantní). Hodnotu řešení $y(x_i)$ aproximujeme pomocí hodnoty y_i :

$$y(x_i) \approx y_i, \quad i = 0, \dots, n.$$

V uzlu x_i platí

$$y'(x_i) = f(x_i, y(x_i)).$$

Předpokládejme, že funkce y je dostatečně hladká. Z Taylorova rozvoje dostaneme

$$y'(x_i) = \frac{y(x_{i+1}) - y(x_i)}{h} + O(h).$$

Dosadíme-li tento vztah do diferenciální rovnice (12.1), dostaneme

$$\frac{y(x_{i+1}) - y(x_i)}{h} + O(h) = f(x_i, y(x_i)).$$

To nás vede k myšlence, zanedbat chybu řádu $O(h)$ a počítat přibližné hodnoty $y_i, i = 1, \dots, n$ ze vztahu

$$\frac{y_{i+1} - y_i}{h} = f(x_i, y_i)$$

tak, že přibližnou hodnotu funkce y v uzlu x_{i+1} vyjádříme pomocí přibližné hodnoty v uzlu x_i :

$$y_{i+1} = y_i + h f(x_i, y_i), \quad y_0 = \eta \text{ (dáno)}. \quad (12.3)$$

Tato metoda se nazývá *dopředná Eulerova metoda* pro řešení úlohy $y' = f(x, y)$ s danou počáteční podmínkou.

Při odvozování Eulerovy metody mohla být derivace aproximována též jako $y'(x_i) \approx \frac{y(x_i) - y(x_{i-1}))}{h}$ a analogickým dosazením bychom získali

$$y_i = y_{i-1} + h f(x_i, y_i), \quad y_0 = \eta \text{ (dáno)}. \quad (12.4)$$

Tuto metodu nazýváme *zpětná Eulerova metoda*.

Povšimněme si rozdíl ve schématech (12.3) a (12.4). V prvním uvedeném schématu lze snadno vypočítat hodnotu y_{i+1} , neboť tato hodnota se vyskytuje pouze na levé straně výrazu (12.3), na pravé jsou již známé hodnoty (tj. hodnoty s indexem o jedničku nižší). Schéma tohoto typu se nazývá *explicitní schéma*. V druhém schématu (12.4) je vidět, že neznámá hodnota y_i je na obou stranách rovnice. V tomto případě metodu nazýváme *implicitní*. Pro její použití převedeme rovnici (12.4) do tvaru

$$y_i - y_{i-1} - h f(x_i, y_i) = 0, \quad (12.5)$$

a tuto rovnici pro každé x_i řešíme pomocí vhodné numerické metody pro řešení nelineárních rovnic.

Lze též vyjít ze vztahu (12.2) a integrál na pravé straně nahradit pomocí numerické kvadratury. Toto obecně vede na vícezkrokové metody (Adamsova typu), kterými se zde nebudeme detailně zabývat (y_{i+1} závisí na více hodnotách y_i, y_{i-1}, \dots , viz otázky k zamyšlení). Pro ilustraci uvažujme aplikaci lichoběžníkového pravidla a zvolme integrační body x_{i+1} a x_i . Úpravou dostáváme následující schéma (*Crankovo-Nicolsonovo*)

$$y_{i+1} = y_i + \frac{1}{2}h (f(x_i, y_i) + f(x_{i+1}, y_{i+1})), \quad (12.6)$$

které je implicitní.

Nabízí se i kombinace obou přístupů. Uvažujme metodu (12.6), kde odhad y_i na pravé straně rovnice spočteme pomocí explicitní metody (12.3), dostáváme tak schéma (*Heunovo*)

$$y_{i+1} = y_i + \frac{1}{2}h (f(x_i, y_i) + f(x_{i+1}, y_i + h f(x_i, y_i))). \quad (12.7)$$

Tento postup kdy použijeme explicitní schémata pro vstup do implicitních metod se nazývá metody typu *prediktor–korektor*.

12.1 Jednokrokové metody

Obecně uvažujme metody typu:

$$y_{i+1} = y_i + h \underbrace{\phi(x_i, y_i, h)}_{\text{přírůstkové zobrazení}}. \quad (12.8)$$

Tato metoda se nazývá jednokroková, protože hodnotu aproximace y_{i+1} počítáme pomocí hodnoty y_i . Pro Eulerovu metodu máme

$$\phi(x_i, y_i, h) := f(x_i, y_i).$$

Při použití Eulerovy metody dále platí pro hodnoty přesného řešení

$$\underbrace{\frac{y(x+h) - y(x)}{h}}_{\text{přesný relativní přírůstek}} = y'(x) + O(h) = f(x, y(x)) + O(h) = \phi(x, y(x), h) + O(h).$$

V Eulerově metodě se tedy liší přesný relativní přírůstek a přírůstkové zobrazení o veličinu řádu $O(h)$:

$$\frac{y(x+h) - y(x)}{h} = \phi(x, y(x), h) + O(h).$$

To nás vede k definici řádu metody:

Definice 12.1. Řekneme, že metoda (12.8) je řádu p , jestliže

$$\frac{y(x+h) - y(x)}{h} = \phi(x, y(x), h) + O(h^p). \quad (12.9)$$

Jinými slovy definice říká, že obecná jednokroková metoda (12.8) je řádu p , jestliže přesné řešení splňuje vztah (12.8) s chybou $hO(h^p)$.

Definice 12.2. Řekneme, že obecná jednokroková metoda je konvergentní, jestliže

$$\forall i = 0, \dots, n, \quad |y(x_i) - y_i| \leq \varphi(h),$$

kde $\varphi(h)$ je infinitesimální vzhledem k h . V takovém případě řekneme, že metoda je konvergentní s řádem p , jestliže $\varphi(h) = O(h^p)$.

Věta 12.3. Metoda (12.8) je konvergentní, právě když $f(x, y) = \phi(x, y, 0)$, za předpokladu spojitosti f, ϕ a lipschitzovskosti f a ϕ v druhé proměnné.

Věta 12.4. (Odhad chyby) Je-li metoda řádu p , potom \exists konstanta $C \geq 0$ taková, že

$$|y(x_i) - y_i| \leq C \cdot h^p \cdot \frac{e^{L(x_i - x_0)} - 1}{L},$$

za předpokladu spojitosti f, ϕ a lipschitzovskosti f a ϕ v druhé proměnné. Zde L je konstanta lipschitzovskosti přírůstkového zobrazení ϕ .

Poznámka: 12.1. Obdobně se odvodí jednokrokové metody pro řešení soustav obyčejných diferenciálních rovnic, kde $f : [a, b] \times \mathbb{R}^m \rightarrow \mathbb{R}^m, \eta \in \mathbb{R}^m$

$$\begin{aligned} y'(x) &= f(x, y(x)), & x \in [a, b], \\ y(a) &= \eta, \end{aligned}$$

12.1.1 Metody typu Runge–Kutta

Podobně jako při konstrukci Eulerovy metody, která je metodou prvního řádu, můžeme postupovat při odvození metody vyššího řádu. Z Taylorova rozvoje funkce y dostaneme

$$\begin{aligned} y(x_{i+1}) &= y(x_i) + h y'(x_i) + \frac{1}{2} h^2 y''(x_i) + O(h^3), \\ y(x_{i+1}) &= y(x_i) + h f(x_i, y(x_i)) + \frac{1}{2} h^2 \frac{df}{dx}(x_i, y(x_i)) + O(h^3). \end{aligned}$$

Podle věty o derivaci složené funkce máme

$$\frac{df}{dx} = f_x + f_y f.$$

Můžeme tak zkonstruovat metodu druhého řádu s přírůstkovým zobrazením

$$\phi(x, y, h) = f(x, y) + \frac{1}{2} h (f_x(x, y) + f_y(x, y) f(x, y)),$$

které ale závisí na derivacích zobrazení f . Proto se používají metody typu **Runge–Kutta**: konstruuje se ϕ , splňující (12.9), bez použití derivací f . Základní myšlenka spočívá v tom, že přírůstkové zobrazení se hledá ve speciálním tvaru tak, aby se lišilo od přesného relativního přírůstku o veličinu $O(h^p)$. Tvar, ve kterém se hledá přírůstkové zobrazení, je následující:

$$\phi(x, y, h) = \sum_{i=1}^s \omega_i k_i = \omega_1 k_1 + \omega_2 k_2 + \cdots + \omega_s k_s,$$

kde ω_i jsou konstanty. Veličiny k_i jsou vyjádřeny pomocí hodnot zobrazení f bez použití jeho derivací.

$$\begin{aligned} k_1 &= f(x, y), \\ k_2 &= f(x + \alpha_2 h, y + \beta_{21} h k_1), \\ &\vdots \\ k_i &= f(x + \alpha_i h, y + h \sum_{j=1}^{i-1} \beta_{ij} k_j), \\ &\vdots \\ k_s &= f(x + \alpha_s h, y + h \sum_{j=1}^{s-1} \beta_{sj} k_j), \end{aligned}$$

kde α_i, β_{ij} jsou konstanty. Ve výše uvedených vzorcích je obecně

$$s \neq p.$$

Pro požadovaný řád metody $p \leq 4$ lze volit $s := p$. Pro $p > 4$ musí být $s > p$.

Rungeova–Kuttova metoda 2. řádu

Ukážeme, jak se určí konstanty $\omega_i, \alpha_i, \beta_{ij}$ na příkladu odvození Rungeovy–Kuttovy metody 2. řádu, tj. pro $p = 2$. Přírůstkové zobrazení hledáme ve tvaru

$$\phi(x, y, h) = \omega_1 f(x, y) + \omega_2 f(x + \alpha h, y + \beta h f(x, y)).$$

Cílem je určit konstanty $\omega_1, \omega_2, \alpha, \beta$ tak, aby metoda byla 2. řádu, tj. aby

$$\frac{y(x+h) - y(x)}{h} = \phi(x, y, h) + O(h^2).$$

Myšlenka je založena na vyjádření přesného relativního přírůstku pomocí Taylorova rozvoje ve tvaru

$$\frac{y(x+h) + y(x)}{h} = \text{výraz 1} + O(h^2)$$

a vyjádření přírůstkového zobrazení ϕ ve tvaru

$$\phi(x, y, h) = \text{výraz 2} + O(h^2).$$

Konstanty $\omega_1, \omega_2, \alpha, \beta$ ve ‘výraz 2’ nastavíme tak, aby ‘výraz 1’ = ‘výraz 2’. Z Taylorova rozvoje funkce y dostaneme

$$y(x+h) = y(x) + hf + \frac{1}{2}h^2 \left(\underbrace{f_x + f_y f}_{y''(x) = \frac{d}{dx}f(x, y(x)) = f_x + f_y f} \right) + O(h^3),$$

odkud

$$\frac{y(x+h) - y(x)}{h} = \underbrace{f + \frac{1}{2}hf_x + \frac{1}{2}hf_y f}_{\text{výraz 1}} + O(h^2).$$

Na základě Taylorova rozvoje funkce dvou proměnných f

$$f(x+h_1, y+h_2) = f(x, y) + h_1 \frac{\partial f}{\partial x}(x, y) + h_2 \frac{\partial f}{\partial y}(x, y) + O(|h|^2)$$

pro $h_1 = \alpha h, h_2 = h\beta f$ a definice přírůstkového zobrazení v metodě typu Runge–Kutta

$$\phi(x, y, h) = \omega_1 f(x, y) + \omega_2 f(x + \alpha h, y + \beta h f(x, y))$$

vyjádříme přírůstkové zobrazení ve tvaru

$$\phi(x, y, h) = \underbrace{\omega_1 f + \omega_2 [f + \alpha h f_x + \beta h f f_y]}_{\text{výraz 2}} + O(h^2).$$

‘výraz 2’ ještě upravíme, abychom ho mohli porovnat s ‘výrazem 1’:

$$\phi(x, y, h) = \underbrace{(\omega_1 + \omega_2)f + \omega_2 \alpha h f_x + \omega_2 \beta h f f_y}_{\text{výraz 2}} + O(h^2).$$

Porovnáním koeficientů u f, hf_x a $hf_y f$ ve ‘výraz 1’ a ‘výraz 2’ získáme rovnice pro $\omega_1, \omega_2, \alpha$ a β

$$1 = \omega_1 + \omega_2, \quad \frac{1}{2} = \alpha \omega_2, \quad \frac{1}{2} = \beta \omega_2.$$

Odvodili jsme tak 3 rovnice pro 4 neznámé. Zvolíme např. $\omega_1 = 0$ a určíme zbývající konstanty:

$$\omega_2 = 1, \\ \alpha = \frac{1}{2}, \quad \beta = \frac{1}{2}.$$

Runge–Kuttovu metodu 2. řádu lze tedy zapsat ve tvaru

$$y_{i+1} = y_i + h\phi(x_i, y_i, h),$$
$$\phi(x_i, y_i, h) = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hf(x_i, y_i)\right).$$



OTÁZKY K ZAMYŠLENÍ

1. Odvoďte pomocí integrace pravé strany (12.2) za využití Simpsonova pravidla víceukrokový vzorec (*Simpsonovo pravidlo*). Použijte integraci přes uzlové body x_{i-1} až x_{i+1} .



ÚKOLY

1. Implementujte algoritmy všech zde uvedených metod pro řešení ODR. Vstupem je funkce f , počáteční podmínka, velikost kroku h a interval, na kterém nás zajímá řešení. Výstupem je numerické řešení v daných uzlových bodech.
2. Pro úlohu se známým řešením porovnejte výstupy explicitních a implicitních metod s přesným řešením. Vykoušejte i pro různé velikosti kroku h . A zároveň v případě explicitních jednokrokových metod ověřte vliv řádu metody na přesnost.



ODKAZY NA LITERATURU

Základní:

FELCMAN, Jiří. *Numerické metody [online]*. Ústí nad Labem: UCP SVP UJEP, 2015 [cit. 2016-04-06].

QUARTERONI, Alfio, Riccardo SACCO a Fausto SALERI. *Numerical mathematics*. New York: Springer, c2000. ISBN 0387989595.

SEGETHOVÁ, Jitka. *Základy numerické matematiky*. 1. vyd. Praha: Karolinum, 1998. ISBN 80-7184-596-5.

VITÁSEK, Emil. *Numerické metody*. 1. vyd. Praha: SNTL, 1987.

Rozšiřující:

PRESS, William H. *Numerical recipes: the art of scientific computing*. 3rd ed. New York: Cambridge University Press, 2007. ISBN 0521706858.

QUARTERONI, Alfio a Fausto SALERI. *Scientific computing with MATLAB*. Berlin: Springer, 2003. ISBN 3540443630.