

NI-MPI přednáška 8

Mocninná metoda a QR algoritmus

Štěpán Starosta

11. 11. 2024

FIT ČVUT

14. Přímé a iterační metody obecně

15. Vlastní čísla a vektory: připomenutí

16. Norma – připomenutí

17. Mocninná metoda

18. QR algoritmus

14. Přímé a iterační metody obecně

15. Vlastní čísla a vektory: připomenutí

16. Norma – připomenutí

17. Mocninná metoda

18. QR algoritmus

Přímá metoda počítá řešení nějakého problému v konečném počtu kroků tak, že v teoretické absolutní přesnosti dává (přesné) řešení.

Příklady z lineární algebry:

- Gaussova eliminační metoda (GEM),
- metoda hledání inverze matice pomocí GEM,
- ...pomocí rozkladů matic (Choleského, LU, QR, ...),
- ...

Obecná myšlenka iteračních metod

Iterační metody hledají přibližná řešení matematických problémů tak, že konstruují posloupnost přibližných „řešení“:

$$x_0, x_1, x_2, \dots$$

¹v nějaké normě

Obecná myšlenka iteračních metod

Iterační metody hledají přibližná řešení matematických problémů tak, že konstruují posloupnost přibližných „řešení“:

$$x_0, x_1, x_2, \dots$$

Každé další přibližné „řešení“ je odvozeno z předchozího:

$$x_k = T(x_{k-1}),$$

pro $k > 0$ a zatím blíže neurčené zobrazení T .

¹v nějaké normě

Obecná myšlenka iteračních metod

Iterační metody hledají přibližná řešení matematických problémů tak, že konstruují posloupnost přibližných „řešení“:

$$x_0, x_1, x_2, \dots$$

Každé další přibližné „řešení“ je odvozeno z předchozího:

$$x_k = T(x_{k-1}),$$

pro $k > 0$ a zatím blíže neurčené zobrazení T .

Zobrazení T je voleno tak, aby posloupnost $(x_k)_{k=0}^{\infty}$ **měla limitu¹**, která je **skutečným řešením** dané úlohy.

¹v nějaké normě

Obecná myšlenka iteračních metod

Iterační metody hledají přibližná řešení matematických problémů tak, že konstruují posloupnost přibližných „řešení“:

$$x_0, x_1, x_2, \dots$$

Každé další přibližné „řešení“ je odvozeno z předchozího:

$$x_k = T(x_{k-1}),$$

pro $k > 0$ a zatím blíže neurčené zobrazení T .

Zobrazení T je voleno tak, aby posloupnost $(x_k)_{k=0}^{\infty}$ **měla limitu¹**, která je **skutečným řešením** dané úlohy.

Poznámka: Pokud je T neměnné pro všechny iterace k , metoda se nazývá **stacionární**.

¹v nějaké normě

14. Přímé a iterační metody obecně

15. Vlastní čísla a vektory: připomenutí

16. Norma – připomenutí

17. Mocninná metoda

18. QR algoritmus

- Komplexní číslo λ nazýváme **vlastním číslem matice** $M \in \mathbb{C}^{n,n}$, právě když existuje nenulový vektor $\mathbf{u} \in \mathbb{C}^n$ splňující rovnici

$$M\mathbf{u} = \lambda\mathbf{u}.$$

Takovýto vektor \mathbf{u} pak nazýváme **vlastním vektorem matice** M příslušejícím k vlastnímu číslu λ .

- Komplexní číslo λ nazýváme **vlastním číslem matice** $M \in \mathbb{C}^{n,n}$, právě když existuje nenulový vektor $\mathbf{u} \in \mathbb{C}^n$ splňující rovnici

$$M\mathbf{u} = \lambda\mathbf{u}.$$

Takovýto vektor \mathbf{u} pak nazýváme **vlastním vektorem matice** M příslušejícím k vlastnímu číslu λ .

- Všechny vlastní vektory matice M příslušející vlastnímu číslu λ spolu s nulovým vektorem tvoří podprostor $\ker(M - \lambda E)$.

- Komplexní číslo λ nazýváme **vlastním číslem matice** $M \in \mathbb{C}^{n,n}$, právě když existuje nenulový vektor $\mathbf{u} \in \mathbb{C}^n$ splňující rovnici

$$M\mathbf{u} = \lambda\mathbf{u}.$$

Takovýto vektor \mathbf{u} pak nazýváme **vlastním vektorem matice** M příslušejícím k vlastnímu číslu λ .

- Všechny vlastní vektory matice M příslušející vlastnímu číslu λ spolu s nulovým vektorem tvoří podprostor $\ker(M - \lambda E)$.
- Vlastní čísla matice M jsou právě kořeny **charakteristického polynomu matice** M , tj. polynomu

$$p_M(\lambda) := \det(M - \lambda E).$$

Každá takováto matice M má proto nejvýše n různých komplexních vlastních čísel.

- Hledat kořeny charakteristického polynomu „velkého“ stupně není snadné. Postup pro hledání vlastních čísel známý z lineární algebry není použitelný.

Vlastní čísla a vektory

- Hledat kořeny charakteristického polynomu „velkého“ stupně není snadné. Postup pro hledání vlastních čísel známý z lineární algebry není použitelný.
- Dokonce platí následující pozorování: umíme-li hledat vlastní čísla matice, pak umíme hledat kořeny polynomů. Skutečně, pro monický polynom

$$q(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_1x + c_0$$

jsou vlastní čísla matice (tzv. **matice společnice** (*companion matrix*) monického polynomu q)

$$C(q) := \begin{pmatrix} 0 & 0 & \cdots & 0 & -c_0 \\ 1 & 0 & \cdots & 0 & -c_1 \\ 0 & 1 & \cdots & 0 & -c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -c_{n-1} \end{pmatrix}$$

přesně kořeny polynomu q . Polynom q je totiž (případně až na multiplikační znaménko) charakteristickým polynomem matice $C(q)$.

Diagonalizovatelnost matice

- Matice $M \in \mathbb{C}^{n,n}$ je **diagonalizovatelná**, právě když existují diagonální matice $D \in \mathbb{C}^{n,n}$ a regulární matice $P \in \mathbb{C}^{n,n}$ splňující

$$M = PDP^{-1}.$$

Diagonalizovatelnost matice

- Matice $M \in \mathbb{C}^{n,n}$ je **diagonalizovatelná**, právě když existují diagonální matice $D \in \mathbb{C}^{n,n}$ a regulární matice $P \in \mathbb{C}^{n,n}$ splňující

$$M = PDP^{-1}.$$

- **Připomenutí:** V předchozí přednášce jsme zkoumali mocniny matice M , platí $M^k = PD^kP^{-1}$. Vlastní čísla kontrolují asymptotické chování těchto mocnin pro $k \rightarrow +\infty$.

Diagonalizovatelnost matice

- Matice $M \in \mathbb{C}^{n,n}$ je **diagonalizovatelná**, právě když existují diagonální matice $D \in \mathbb{C}^{n,n}$ a regulární matice $P \in \mathbb{C}^{n,n}$ splňující

$$M = PDP^{-1}.$$

- **Připomenutí:** V předchozí přednášce jsme zkoumali mocniny matice M , platí $M^k = PD^kP^{-1}$. Vlastní čísla kontrolují asymptotické chování těchto mocnin pro $k \rightarrow +\infty$.
- **Poznámka:** Matice P obsahuje ve sloupcích vlastní vektory matice M (Lze ověřit. Tyto vlastní vektory tvoří bázi \mathbb{C}^n . Prvky na diagonále matice D jsou právě vlastní čísla matice M (včetně násobností).

Využití vlastních čísel

Vlastní čísla hrají důležitou roli v řadě aplikací, například

- Klasifikace kuželoseček a kvadrik (geometrie).
- Kvantové počítání, kvantová mechanika, asymptotické chování různých systémů (fyzika).
- Analýza hlavních komponent: PCA neboli *principal component analysis* (big data).
- Rozpoznávání 2D i 3D objektů pomocí spektrálních metod (AI).

Využití vlastních čísel

Vlastní čísla hrají důležitou roli v řadě aplikací, například

- Klasifikace kuželoseček a kvadrik (geometrie).
- Kvantové počítání, kvantová mechanika, asymptotické chování různých systémů (fyzika).
- Analýza hlavních komponent: PCA neboli *principal component analysis* (big data).
- Rozpoznávání 2D i 3D objektů pomocí spektrálních metod (AI).
- Konkrétnější příklad: **PageRank** měří relativní důležitost WWW stránek zkoumáním odkazů mezi nimi.
 - Jeho výpočet spočívá ve výpočtu vlastního vektoru k dominantnímu číslu modifikované matice sousednosti grafu těchto odkazů.
 - K výpočtu **PageRanku** se používá se tzv. **mocinná metoda** (*power method*), kterou si probereme dále.

14. Přímé a iterační metody obecně

15. Vlastní čísla a vektory: připomenutí

16. Norma – připomenutí

17. Mocninná metoda

18. QR algoritmus

Definice 17.1 (Norma (*norm*))

Norma na vektorovém prostoru V (nad \mathbb{R} nebo \mathbb{C}) je zobrazení $\| \cdot \| : V \rightarrow \mathbb{R}_0^+$ splňující:

- 1 $\|x\| = 0 \Rightarrow x = 0$,
- 2 $\|\alpha x\| = |\alpha| \cdot \|x\|$,
- 3 $\|x + y\| \leq \|x\| + \|y\|$ (trojúhelníková nerovnost),

pro všechna $x, y \in V$ a všechny skaláry α .

Definice 17.1 (Norma (*norm*))

Norma na vektorovém prostoru V (nad \mathbb{R} nebo \mathbb{C}) je zobrazení $\| \cdot \| : V \rightarrow \mathbb{R}_0^+$ splňující:

- 1 $\|x\| = 0 \Rightarrow x = 0$,
- 2 $\|\alpha x\| = |\alpha| \cdot \|x\|$,
- 3 $\|x + y\| \leq \|x\| + \|y\|$ (trojúhelníková nerovnost),

pro všechna $x, y \in V$ a všechny skaláry α .

Na \mathbb{R}^n (\mathbb{C}^n) je nejznámější pravděpodobně **eukleidovská** norma:

$$\|x\|_2 := \left(\sum_{i=1}^n |x_i|^2 \right)^{\frac{1}{2}},$$

kde $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ (\mathbb{C}^n). (Ověření trojúhelníkové nerovnosti není triviální!)

Norma – další příklady znovu

Dalšími často užívanými normami jsou:

$$\|x\|_{\infty} := \max \{ |x_i| \mid i \in \{1, \dots, n\} \} \quad (\text{tzv. maximová norma})$$

$$\|x\|_1 := \sum_{i=1}^n |x_i| \quad (\text{tzv. součtová norma})$$

Norma – další příklady znovu

Dalšími často užívanými normami jsou:

$$\|x\|_{\infty} := \max \{|x_i| \mid i \in \{1, \dots, n\}\} \quad (\text{tzv. maximová norma})$$

$$\|x\|_1 := \sum_{i=1}^n |x_i| \quad (\text{tzv. součtová norma})$$

Obecně, pro libovolné $p \geq 1$ je

$$\|x\|_p := \sqrt[p]{\sum_{i=1}^n |x_i|^p}$$

normou na \mathbb{R}^n (resp. \mathbb{C}^n). (Ověření trojúhelníkové nerovnosti není triviální!)

Řekneme, že dvě normy $\|\cdot\|_a$ a $\|\cdot\|_b$ na prostoru V jsou *ekvivalentní*, pokud existují konstanty C_1 a C_2 takové, že

$$\forall x \in V, \quad C_1\|x\|_b \leq \|x\|_a \leq C_2\|x\|_b.$$

Věta 17.2

Nechť V je vektorový prostor konečné dimenze nad \mathbb{R} nebo \mathbb{C} a necht' $\|\cdot\|_a$ a $\|\cdot\|_b$ jsou dvě normy na V . Platí, že $\|\cdot\|_a$ a $\|\cdot\|_b$ jsou ekvivalentní.

14. Přímé a iterační metody obecně

15. Vlastní čísla a vektory: připomenutí

16. Norma – připomenutí

17. Mocninná metoda

18. QR algoritmus

Mocninná metoda: úvod a předpoklady (1/3)

- Ve své základní variantě slouží mocninná metoda k nalezení v absolutní hodnotě největšího vlastního čísla (takovému se říká **dominantní** vlastní číslo) a příslušného vlastního vektoru.

Mocninná metoda: úvod a předpoklady (1/3)

- Ve své základní variantě slouží mocninná metoda k nalezení v absolutní hodnotě největšího vlastního čísla (takovému se říká **dominantní** vlastní číslo) a příslušného vlastního vektoru.
- Mějme matici $M \in \mathbb{C}^{n,n}$. **Předpokládejme**, že je diagonalizovatelná, tj. existuje regulární matice $P \in \mathbb{C}^{n,n}$ splňující

$$M = PDP^{-1},$$

kde $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ a dále navíc **předpokládejme**, že můžeme její vlastní čísla označit takto:

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|.$$

Mocninná metoda: úvod a předpoklady (1/3)

- Ve své základní variantě slouží mocninná metoda k nalezení v absolutní hodnotě největšího vlastního čísla (takovému se říká **dominantní** vlastní číslo) a příslušného vlastního vektoru.
- Mějme matici $M \in \mathbb{C}^{n,n}$. **Předpokládejme**, že je diagonalizovatelná, tj. existuje regulární matice $P \in \mathbb{C}^{n,n}$ splňující

$$M = PDP^{-1},$$

kde $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ a dále navíc **předpokládejme**, že můžeme její vlastní čísla označit takto:

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|.$$

Poznámky:

- Z výše uvedeného plyne, že dominantní vlastní číslo λ_1 není degenerované (příslušný vlastní podprostor matice M má dimenzi 1).
- Předpoklad diagonalizovatelnosti lze odstranit, v zájmu jednoduchosti výkladu ho ale přijmeme.

Mocninná metoda: úvod a předpoklady (2/3)

Jak moc je předpoklad o vlastních číslech restriktivní?

Mocninná metoda: úvod a předpoklady (2/3)

Jak moc je předpoklad o vlastních číslech restriktivní?

Splňují jej velké třídy matic: kladné matice a obecněji primitivní² matice (tzv. Perronova-Frobeniova věta).

²definice v handoutu

Mocninná metoda: úvod a předpoklady (2/3)

Jak moc je předpoklad o vlastních číslech restriktivní?

Splňují jej velké třídy matic: kladné matice a obecněji primitivní² matice (tzv. Perronova-Frobeniova věta).

Jak moc je předpoklad o diagonalizovatelnosti restriktivní?

²definice v handoutu

Mocinná metoda: úvod a předpoklady (2/3)

Jak moc je předpoklad o vlastních číslech restriktivní?

Splňují jej velké třídy matic: kladné matice a obecněji primitivní² matice (tzv. Perronova-Frobeniova věta).

Jak moc je předpoklad o diagonalizovatelnosti restriktivní?

Není, ten máme pouze pro zjednodušení. V obecnější variantě lze opět použít Jordanův normální tvar (bez změny principu).

²definice v handoutu

- Hledáme vlastní vektor přidružený k vlastnímu číslu λ_1 , tedy nenulový vektor \mathbf{u}_1 splňující

$$M\mathbf{u}_1 = \lambda_1\mathbf{u}_1.$$

Mocninná metoda: úvod a předpoklady (3/3)

- Hledáme vlastní vektor přidružený k vlastnímu číslu λ_1 , tedy nenulový vektor \mathbf{u}_1 splňující

$$M\mathbf{u}_1 = \lambda_1\mathbf{u}_1.$$

- Mocninná metoda je **iterativní metoda**. Zvolme $\mathbf{x}_0 \in \mathbb{C}^n$ a sestrojme posloupnost $(\mathbf{x}_k)_{k=0}^{\infty}$ zadanou rekurentně vztahem

$$\mathbf{x}_{k+1} = M\mathbf{x}_k, \quad k \in \mathbb{N}.$$

Ekvivalentně máme explicitní vyjádření

$$\mathbf{x}_k = M^k\mathbf{x}_0, \quad k \in \mathbb{N}.$$

Mocninná metoda: úvod a předpoklady (3/3)

- Hledáme vlastní vektor přidružený k vlastnímu číslu λ_1 , tedy nenulový vektor \mathbf{u}_1 splňující

$$M\mathbf{u}_1 = \lambda_1\mathbf{u}_1.$$

- Mocninná metoda je **iterativní metoda**. Zvolme $\mathbf{x}_0 \in \mathbb{C}^n$ a sestrojme posloupnost $(\mathbf{x}_k)_{k=0}^{\infty}$ zadanou rekurentně vztahem

$$\mathbf{x}_{k+1} = M\mathbf{x}_k, \quad k \in \mathbb{N}.$$

Ekvivalentně máme explicitní vyjádření

$$\mathbf{x}_k = M^k\mathbf{x}_0, \quad k \in \mathbb{N}.$$

- Tento vzorec je původem vžitého názvu *mocninná metoda* (též *power iteration*).

Mocninná metoda: vlastní vektor (1/3)

- Vektor \mathbf{x}_0 lze napsat jako lineární kombinaci vlastních vektorů matice M , tj.

$$\mathbf{x}_0 = P\boldsymbol{\alpha},$$

kde $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^T \in \mathbb{C}^n$ je vektor obsahující koeficienty příslušné lineární kombinace.

Předpokládejme, že $\alpha_1 \neq 0$. (Při náhodné volbě vektoru \mathbf{x}_0 je rovnost $\alpha_1 = 0$ nepravděpodobná.)

Mocninná metoda: vlastní vektor (1/3)

- Vektor \mathbf{x}_0 lze napsat jako lineární kombinaci vlastních vektorů matice M , tj.

$$\mathbf{x}_0 = P\boldsymbol{\alpha},$$

kde $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^T \in \mathbb{C}^n$ je vektor obsahující koeficienty příslušné lineární kombinace.

Předpokládejme, že $\alpha_1 \neq 0$. (Při náhodné volbě vektoru \mathbf{x}_0 je rovnost $\alpha_1 = 0$ nepravděpodobná.)

- Pro naši posloupnost $(\mathbf{x}_k)_{k=0}^{\infty}$ potom platí

$$\begin{aligned}\mathbf{x}_k &= M^k \mathbf{x}_0 = (PDP^{-1})^k \mathbf{x}_0 = PD^k \boldsymbol{\alpha} = P \operatorname{diag}(\lambda_1^k, \dots, \lambda_n^k) \boldsymbol{\alpha} = \\ &= \lambda_1^k P \operatorname{diag}\left(1, \left(\frac{\lambda_2}{\lambda_1}\right)^k, \dots, \left(\frac{\lambda_n}{\lambda_1}\right)^k\right) \boldsymbol{\alpha},\end{aligned}$$

kde $D = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$.

Mocninná metoda: vlastní vektor (2/3)

- Nyní vektor \mathbf{x}_k normalizujeme tak, aby jeho první složka byla rovna 1 (vizte poznámku dále), tedy

$$\begin{aligned} \mathbf{y}_k &:= \frac{\mathbf{x}_k}{(\mathbf{x}_k)_1} = \frac{P \operatorname{diag}\left(1, \left(\frac{\lambda_2}{\lambda_1}\right)^k, \dots, \left(\frac{\lambda_n}{\lambda_1}\right)^k\right) \boldsymbol{\alpha}}{\left(P \operatorname{diag}\left(1, \left(\frac{\lambda_2}{\lambda_1}\right)^k, \dots, \left(\frac{\lambda_n}{\lambda_1}\right)^k\right) \boldsymbol{\alpha}\right)_1} \longrightarrow \\ &\longrightarrow \frac{P \operatorname{diag}(1, 0, \dots, 0) \boldsymbol{\alpha}}{(P \operatorname{diag}(1, 0, \dots, 0) \boldsymbol{\alpha})_1} =: \mathbf{y}_\infty \in \ker(M - \lambda_1 E), \end{aligned}$$

když $k \rightarrow +\infty$. Skutečně, dle našeho předpokladu totiž

$$\lim_{k \rightarrow \infty} \left(\frac{\lambda_i}{\lambda_1}\right)^k = 0$$

pro každé $i = 2, 3, \dots, n$ a

$$P \operatorname{diag}(1, 0, \dots, 0) \boldsymbol{\alpha} = \alpha_1 \mathbf{u}_1 \in \ker(M - \lambda_1 E).$$

Mocninná metoda: vlastní vektor (3/3)

Hned na tomto místě je vhodné učinit několik poznámek:

Mocninná metoda: vlastní vektor (3/3)

Hned na tomto místě je vhodné učinit několik poznámek:

- Při normalizaci můžeme/musíme zvolit i **jinou složku než první**, která je nenulová. Vlastní vektor je nenulový a jedna z jeho složek proto nutně musí být nenulová.

Mocnná metoda: vlastní vektor (3/3)

Hned na tomto místě je vhodné učinit několik poznámek:

- Při normalizaci můžeme/musíme zvolit i **jinou složku než první**, která je nenulová. Vlastní vektor je nenulový a jedna z jeho složek proto nutně musí být nenulová.
- Kdybychom věděli, že $\lambda_1 > 0$, pak lze vektory \mathbf{x}_k skutečně normalizovat a tím se také zbavit mocnin λ_1^k . Ve výpočtu výše bychom kladli (stručný zápis)

$$\mathbf{y}_k := \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|} = \frac{\lambda_1^k(P \dots)}{\|\lambda_1^k(P \dots)\|} = \frac{\lambda_1^k(P \dots)}{|\lambda_1^k| \|(P \dots)\|} = \frac{\lambda_1^k(P \dots)}{\lambda_1^k \|(P \dots)\|} = \frac{(P \dots)}{\|(P \dots)\|}$$

a posloupnost $(\mathbf{y}_k)_{k=1}^{+\infty}$ by opět byla konvergentní.

Mocninná metoda: vlastní vektor (3/3)

Hned na tomto místě je vhodné učinit několik poznámek:

- Při normalizaci můžeme/musíme zvolit i **jinou složku než první**, která je nenulová. Vlastní vektor je nenulový a jedna z jeho složek proto nutně musí být nenulová.
- Kdybychom věděli, že $\lambda_1 > 0$, pak lze vektory \mathbf{x}_k skutečně normalizovat a tím se také zbavit mocnin λ_1^k . Ve výpočtu výše bychom kladli (stručný zápis)

$$\mathbf{y}_k := \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|} = \frac{\lambda_1^k(P \dots)}{\|\lambda_1^k(P \dots)\|} = \frac{\lambda_1^k(P \dots)}{|\lambda_1^k| \|(P \dots)\|} = \frac{\lambda_1^k(P \dots)}{\lambda_1^k \|(P \dots)\|} = \frac{(P \dots)}{\|(P \dots)\|}$$

a posloupnost $(\mathbf{y}_k)_{k=1}^{+\infty}$ by opět byla konvergentní.

- Pro kladné a nezáporné ireducibilní matice platí $\lambda_1 > 0$ a složky \mathbf{u}_1 jsou také kladné (opět Perronova-Frobeniova věta).

- Vlastní číslo, resp. jeho aproximaci, nyní snadno zjistíme následovně

$$\frac{\langle \mathbf{y}_k, M\mathbf{y}_k \rangle}{\|\mathbf{y}_k\|^2} \longrightarrow \frac{\langle \mathbf{y}_\infty, M\mathbf{y}_\infty \rangle}{\|\mathbf{y}_\infty\|^2} = \frac{\langle \mathbf{y}_\infty, \lambda_1 \mathbf{y}_\infty \rangle}{\|\mathbf{y}_\infty\|^2} = \lambda_1,$$

zde $\langle x, y \rangle$ značí standardní skalární součin³ na \mathbb{C}^n a $\|x\|$ je euklidovská norma.

³Tedy $\langle x, y \rangle = \sum \overline{x_i} y_i$.

- Vlastní číslo, resp. jeho aproximaci, nyní snadno zjistíme následovně

$$\frac{\langle \mathbf{y}_k, M\mathbf{y}_k \rangle}{\|\mathbf{y}_k\|^2} \longrightarrow \frac{\langle \mathbf{y}_\infty, M\mathbf{y}_\infty \rangle}{\|\mathbf{y}_\infty\|^2} = \frac{\langle \mathbf{y}_\infty, \lambda_1 \mathbf{y}_\infty \rangle}{\|\mathbf{y}_\infty\|^2} = \lambda_1,$$

zde $\langle x, y \rangle$ značí standardní skalární součin³ na \mathbb{C}^n a $\|x\|$ je euklidovská norma.

- Alternativně bychom mohli vzít libovolné (nenulové na $\ker(M - \lambda_1 E)$) lineární zobrazení $\varphi : \mathbb{C}^n \rightarrow \mathbb{C}$ a počítat podíly

$$\frac{\varphi(M\mathbf{y}_k)}{\varphi(\mathbf{y}_k)} \longrightarrow \frac{\varphi(M\mathbf{y}_\infty)}{\varphi(\mathbf{y}_\infty)} = \frac{\varphi(\lambda_1 \mathbf{y}_\infty)}{\varphi(\mathbf{y}_\infty)} = \lambda_1.$$

Například $\varphi(\mathbf{v}) = (\mathbf{v})_i$ pro zvolené $i \in \{1, 2, \dots, n\}$.

³Tedy $\langle x, y \rangle = \sum \overline{x_i} y_i$.

- **Kritérium zastavení** ($\|\cdot\|$ je jistá norma na \mathbb{C}^n): máme-li poslední iteraci aproximace vlastního vektoru \mathbf{y}_k a vlastního čísla $\lambda_1^{(k)}$, pak otestujeme reziduum

$$\|M\mathbf{y}_k - \lambda_1^{(k)}\mathbf{y}_k\| < \varepsilon.$$

Mocnná metoda: konvergence

- **Kritérium zastavení** ($\|\cdot\|$ je jistá norma na \mathbb{C}^n): máme-li poslední iteraci aproximace vlastního vektoru \mathbf{y}_k a vlastního čísla $\lambda_1^{(k)}$, pak otestujeme reziduum

$$\|M\mathbf{y}_k - \lambda_1^{(k)}\mathbf{y}_k\| < \varepsilon.$$

- Pro rychlost konvergence posloupnosti platí (odhad explicitně neprovádíme)

$$\|\mathbf{y}_k - \mathbf{y}_\infty\| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$$

a vlastních čísel platí

$$\|\lambda_1^{(k)} - \lambda_1\| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right).$$

Mocnná metoda: konvergence

- **Kritérium zastavení** ($\|\cdot\|$ je jistá norma na \mathbb{C}^n): máme-li poslední iteraci aproximace vlastního vektoru \mathbf{y}_k a vlastního čísla $\lambda_1^{(k)}$, pak otestujeme reziduum

$$\|M\mathbf{y}_k - \lambda_1^{(k)}\mathbf{y}_k\| < \varepsilon.$$

- Pro rychlost konvergence posloupnosti platí (odhad explicitně neprovádíme)

$$\|\mathbf{y}_k - \mathbf{y}_\infty\| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$$

a vlastních čísel platí

$$\|\lambda_1^{(k)} - \lambda_1\| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right).$$

- **Obecné pozorování:** Jsou-li vlastní čísla „blízko u sebe“, pak může být výpočet velmi pomalý.

- Dejme tomu, že jsme mocninnou metodou našli dominantní vlastní číslo λ_1 a jemu příslušející normalizovaný (Euklidovou normou) vlastní vektor \mathbf{u}_1 . Jak hledat **další** vlastní čísla?

⁴a vektor je stále sloupec

Mocninná metoda: poznámky

- Dejme tomu, že jsme mocninnou metodou našli dominantní vlastní číslo λ_1 a jemu příslušející normalizovaný (Euklidovou normou) vlastní vektor \mathbf{u}_1 . Jak hledat **další** vlastní čísla?
- Předpokládejme, že matice M je normální ($MM^\dagger = M^\dagger M$, dýka označuje transpozici a komplexní sdružení), pak má ortogonální vlastní vektory.

⁴a vektor je stále sloupec

Mocninná metoda: poznámky

- Dejme tomu, že jsme mocninnou metodou našli dominantní vlastní číslo λ_1 a jemu příslušející normalizovaný (Euklidovou normou) vlastní vektor \mathbf{u}_1 . Jak hledat **další** vlastní čísla?
- Předpokládejme, že matice M je normální ($MM^\dagger = M^\dagger M$, dýka označuje transpozici a komplexní sdružení), pak má ortogonální vlastní vektory.
- Můžeme matici upravit následovně (tečka označuje maticové násobení⁴; pruh komplexní sdružení složek)

$$M' := M - \lambda_1 \mathbf{u}_1 \cdot \mathbf{u}_1^\dagger.$$

⁴a vektor je stále sloupec

Mocninná metoda: poznámky

- Dejme tomu, že jsme mocninnou metodou našli dominantní vlastní číslo λ_1 a jemu příslušející normalizovaný (Euklidovou normou) vlastní vektor \mathbf{u}_1 . Jak hledat **další** vlastní čísla?
- Předpokládejme, že matice M je normální ($MM^\dagger = M^\dagger M$, dýka označuje transpozici a komplexní sdružení), pak má ortogonální vlastní vektory.
- Můžeme matici upravit následovně (tečka označuje maticové násobení⁴; pruh komplexní sdružení složek)

$$M' := M - \lambda_1 \mathbf{u}_1 \cdot \mathbf{u}_1^\dagger.$$

- Matice M' má nyní vektor \mathbf{u}_1 také jako vlastní vektor, ale s vlastním číslem 0! Skutečně,

$$M' \mathbf{u}_1 = M \mathbf{u}_1 - \lambda_1 \mathbf{u}_1 \cdot \|\mathbf{u}_1\|_2^2 = \lambda_1 \mathbf{u}_1 - \lambda_1 \mathbf{u}_1 = 0.$$

⁴a vektor je stále sloupec

Mocninná metoda: poznámky

- Dejme tomu, že jsme mocninnou metodou našli dominantní vlastní číslo λ_1 a jemu příslušející normalizovaný (Euklidovou normou) vlastní vektor \mathbf{u}_1 . Jak hledat **další** vlastní čísla?
- Předpokládejme, že matice M je normální ($MM^\dagger = M^\dagger M$, dýka označuje transpozici a komplexní sdružení), pak má ortogonální vlastní vektory.
- Můžeme matici upravit následovně (tečka označuje maticové násobení⁴; pruh komplexní sdružení složek)

$$M' := M - \lambda_1 \mathbf{u}_1 \cdot \mathbf{u}_1^\dagger.$$

- Matice M' má nyní vektor \mathbf{u}_1 také jako vlastní vektor, ale s vlastním číslem 0! Skutečně,

$$M' \mathbf{u}_1 = M \mathbf{u}_1 - \lambda_1 \mathbf{u}_1 \cdot \|\mathbf{u}_1\|_2^2 = \lambda_1 \mathbf{u}_1 - \lambda_1 \mathbf{u}_1 = 0.$$

- Nyní aplikujeme mocninnou metodu (jsou-li splněny její předpoklady) na M' a získáme **druhé** v absolutní hodnotě největší vlastní číslo matice M .

⁴a vektor je stále sloupec

Mocninná metoda: ukázka (1/2)

Uvažme matici

$$M = \begin{pmatrix} 36408 + 16769i & -5412 - 2481i & 107256 + 49397i & -492 - 214i \\ -10656 - 5164i & 1584 + 762i & -31392 - 15210i & 144 + 66i \\ -12876 - 5954i & 1914 + 881i & -37932 - 17539i & 174 + 76i \\ 4329 - 262i & -643 + 39i & 12753 - 771i & -58 + 6i \end{pmatrix}$$

Je zkonstruována tak, že její vlastní čísla jsou $-2i$, $-i$, $3i/2$, $3/2$.

Mocninná metoda: ukázka (1/2)

Uvažme matici

$$M = \begin{pmatrix} 36408 + 16769i & -5412 - 2481i & 107256 + 49397i & -492 - 214i \\ -10656 - 5164i & 1584 + 762i & -31392 - 15210i & 144 + 66i \\ -12876 - 5954i & 1914 + 881i & -37932 - 17539i & 174 + 76i \\ 4329 - 262i & -643 + 39i & 12753 - 771i & -58 + 6i \end{pmatrix}$$

Je zkonstruována tak, že její vlastní čísla jsou $-2i$, $-i$, $3i/2$, $3/2$.

Požadujeme přesnost (mez pro reziduum) $\varepsilon = 10^{-6}$. Aproximace λ_1 z posledních 7 iterací:

$$\begin{aligned} & 0.0000477588150960872 - 1.99991424541241i \\ & -0.0000479821875446196 - 1.99998019901599i \\ & -0.0000272650944159076 - 2.00002375338328i \\ & 0.0000271520045767515 - 2.00002973125038i \\ & 0.0000154506695115737 - 1.99997272532314i \\ & 0.0000152424622102764 - 1.99999240227182i \end{aligned}$$

14. Přímé a iterační metody obecně

15. Vlastní čísla a vektory: připomenutí

16. Norma – připomenutí

17. Mocninná metoda

18. QR algoritmus

- Mocninná metoda se nehodí na hledání všech vlastních čísel matice M .
- Navíc maticové násobení je obecně náročně a vyplatí se až pro velké řídké matice.

QR faktorizace a QR algoritmus (1/2)

- Mocninná metoda se nehodí na hledání všech vlastních čísel matice M .
- Navíc maticové násobení je obecně náročně a vyplatí se až pro velké řídké matice.
- Existují další algoritmy založené na sérii podobnostních transformací: cílem je sestrojít posloupnost matic $(M_k)_{k=0}^{\infty}$, $M_0 = M$ tak, že

$$M_k = P_k M_{k-1} P_k^{-1}, \quad k \in \mathbb{N},$$

kde každá P_k je regulární, $M_k \rightarrow M_{\infty}$ a pro limitní matici umíme snadno spočítat vlastní čísla (například je horní trojúhelníková).

QR faktorizace a QR algoritmus (2/2)

- **QR faktorizace** matice spočívá ve vyjádření reálné (resp. komplexní) matice M ve tvaru součinu

$$M = Q \cdot R,$$

kde Q je ortogonální (resp. unitární) a R je horní trojúhelníková.

QR faktorizace a QR algoritmus (2/2)

- **QR faktorizace** matice spočívá ve vyjádření reálné (resp. komplexní) matice M ve tvaru součinu

$$M = Q \cdot R,$$

kde Q je ortogonální (resp. unitární) a R je horní trojúhelníková.

- Existuje několik algoritmů počítajících tuto faktorizaci (Gram-Schmidt, Householderova transformace, Givensova rotace).

QR faktorizace a QR algoritmus (2/2)

- **QR faktorizace** matice spočívá ve vyjádření reálné (resp. komplexní) matice M ve tvaru součinu

$$M = Q \cdot R,$$

kde Q je ortogonální (resp. unitární) a R je horní trojúhelníková.

- Existuje několik algoritmů počítajících tuto faktorizaci (Gram-Schmidt, Householderova transformace, Givensova rotace).
- **QR algoritmus** tuto faktorizaci provádí v každém kroku. Zhruba řečeno, pro M_k spočteme její QR faktorizaci

$$M_k = Q_k R_k$$

a položíme

$$M_{k+1} := R_k Q_k = Q_k^{-1} Q_k R_k Q_k = Q_k^{-1} M_k Q_k$$

Iterace začíná s $M_0 = M$. Všechny matice M_k jsou podobné naší M a mají tedy stejná vlastní čísla. Za jistých předpokladů M_k konverguje k trojúhelníkové matici.