Přednáška 17 – Lineární algebraické systémy II

Opáčko Lingebra 1:

Vektorové normy

- $\|\vec{x}\| \equiv \|\vec{x}\|_2 := \sqrt{x_1^2 + \dots + x_n^2}$ ... obecněji $\|\vec{x}\|_p := \sqrt[p]{|x_1|^p + \dots + |x_n|^p} \quad p \in \mathbb{N}$, $\|\vec{x}\|_\infty := \max\{|x_1|, \dots, |x_n|\}$

Maticové normy

- $\|A\| \equiv \|A\|_2 := \max\limits_{\vec{v} \in \mathbb{R}^n \setminus \{0\}} \dfrac{\|A\vec{v}\|_2}{\|\vec{v}\|_2} \equiv \max\limits_{\substack{\|\vec{v}\|_2 = 1 \\ \vec{v} \in \mathbb{R}^n \setminus \{0\}}} \|A\vec{v}\|_2$

- $\|A\|_p := \max\limits_{\vec{v} \in \mathbb{R}^n \setminus \{0\}} \dfrac{\|A\vec{v}\|_p}{\|\vec{v}\|_p} \equiv \max\limits_{\substack{\|\vec{v}\|_p = 1 \\ \vec{v} \in \mathbb{R}^n \setminus \{0\}}} \|A\vec{v}\|_p$

Vektorová norma nám měří velikost vektoru ⟿ ⟿ maticová norma nám měří „velikost matice" ve smyslu „matice ≡ zobrazení" → „velikost zobrazení ~ jak nejvíce může změnit velikost inputu (relativně)"

- Lze snadno vidět, že:
  - $\|A\vec{v}\|_p \leq \|A\|_p \cdot \|\vec{v}\|_p$
  - $\|AB\|_p \leq \|A\|_p \cdot \|B\|_p$

  $\forall p \in \{1, 2, \dots, \infty\}$

- „velikost matice" lze měřit i „vektorově" → → tzv. Frobeniova norma $\|A\|_F := \sqrt{a_{11}^2 + a_{12}^2 + \dots + a_{1n}^2 + \dots + a_{n1}^2 + \dots + a_{nn}^2}$

- matice, které zachovávají normu se nazývají unitární

  Q je unitární $\iff \forall \vec{v}: \|Q\vec{v}\|_2 = \|\vec{v}\|_2 \iff$ sloupce Q tvoří ortonormální posloupnost

<u>Opáčko</u>:
- Gaussova eliminace odpovídá okamžité aplikaci LU faktorizace
- pro obecnou regulární matici A můžeme potřebovat pivotaci
- i s pivotací může být G.E./LU <u>nestabilní</u>, tj. zaokrouhlovací chyby můžou úplně zkreslit a zkazit vypočtené faktory $\hat{L}, \hat{U}$.
- Pro SPD matice (např. kovarianční matice) lze ukázat, že nepotřebujeme pivotaci a zároveň, že G.E./LU je stabilní

Z toho přirozeně plynou 2 otázky:

1. <u>Existují stabilní (stabilnější) algoritmy pro řešení $A\vec{x}=\vec{b}$?</u>
2. <u>Jak poznáme/vypočteme podmíněnost konkrétního systému $A\vec{x}=\vec{b}$?</u>

→ <u>opáčko:</u> stabilita ≡ jak konkrétní algoritmus ne/akumuluje zaokrouhlovací chyby

podmíněnost ≡ jak je konkrétní matematický problém citlivý na „malé" perturbace (změny v datech)

→ stabilní algoritmy jsou nakonec k ničemu pro špatně podmíněné problémy ⇝ <u>nestačí se zajímat pouze o stabilitu</u>.

<u>Podmíněnost problému $A\vec{x}=\vec{b}$</u>

Mějme tedy malou perturbaci $A\vec{x}=\vec{b}$:

relativní velikosti perturbací << 1

$$\tilde{A}\tilde{x} = \tilde{b} \quad \text{kde} \quad \frac{\|\tilde{A}-A\|}{\|A\|} \leq \varepsilon_A \;\&\; \frac{\|\tilde{b}-b\|}{\|b\|} \leq \varepsilon_b$$

Pak máme:

$$\tilde{x} - x = A^{-1}(A\tilde{x} - Ax) = A^{-1}(A\tilde{x} - \tilde{A}\tilde{x} + \tilde{A}\tilde{x} - Ax) = A^{-1}(-(A-\tilde{A})\tilde{x} + \tilde{b} - b)$$

Tudíž:

$$\|\tilde{x}-x\| \leq \|A^{-1}\| \cdot \left(\varepsilon_A \cdot \|A\| \cdot \|\tilde{x}\| + \varepsilon_b \cdot \|b\| \right) \leq$$

$$\leq \|A^{-1}\| \cdot \left(\varepsilon_A \|A\| \cdot \|\tilde{x}\| + \varepsilon_b \cdot \|A\| \cdot \|x\| \right)$$

$$\leq \|A^{-1}\| \cdot \|A\| \cdot \left(\varepsilon_A \cdot \|\tilde{x}-x+x\| + \varepsilon_b \|x\| \right)$$

$$\leq \|A^{-1}\| \cdot \|A\| \cdot \left(\varepsilon_A \|\tilde{x}-x\| + (\varepsilon_A+\varepsilon_b) \|x\| \right)$$

$$\|\tilde{x}-x\| \cdot \left(1 - \|A^{-1}\| \cdot \|A\| \cdot \varepsilon_A\right) \leq \|A^{-1}\| \cdot \|A\| \cdot (\varepsilon_A+\varepsilon_b) \|x\|$$

$$\frac{\|\tilde{x}-x\|}{\|x\|} \leq \frac{\|A^{-1}\| \cdot \|A\|}{1 - \|A^{-1}\| \cdot \|A\| \cdot \varepsilon_A} \cdot (\varepsilon_A+\varepsilon_b)$$

velikost změny řešení na základě perturbace

velikost perturbace dat

podmíněnost problému $A\vec{x} = \vec{b}$

Číslo $\|A^{-1}\| \cdot \|A\|$ je tzv. číslo podmíněnosti matice A

⇝ klasické značení je $\kappa(A) := \|A\| \cdot \|A^{-1}\|$

Vidíme, že pro obecnou perturbaci dat velikosti ε lze matematicky očekávat přesnost řešení přinejlepším řádu $\mathcal{O}(\kappa(A) \cdot \varepsilon)$ ⇝

⇝ tj. i pro zcela stabilní alg. máme chyby $\mathcal{O}(\kappa(A) \cdot \varepsilon_{mach})$

Dovětek k efektivitě LU / G.E.

Viděli jsme, že pro některé aplikace máme A tzv. řídkou (≡ sparse)
tj. většina prvků A je rovna 0 a ty nenulové prvky jsou „řídké", např. aproximace druhé derivace vedla na matici $A = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & -2 \end{bmatrix}$ ⇝

⇝ z n^2 prvků jich je pouze $3n-2$ nenulových.

Podobně tomu je pro většinu problémů, které v nějakém smyslu aproximují nebo pracují s derivacemi.

$\Longrightarrow$ abychom měli efektivní výpočty, stačí nám pracovat pouze s těmi nenulovými prvky ⇝

⇝ tzv. „sparse-matrix algorithms".

Jejich odvozování & implementace je technická a pokročilá, ale jejich používání snadné:

Python demo: sparse vs. dense

storage & time

Stabilnější algoritmy

→ jak nám vznikala nestabilita? Dělili jsme disproporčně velkým nebo malým číslem.

V řeči matic: matice $M^{(k)}$ velmi výrazně měnili normu sloupců & řádků matice $A^{(k-1)}$.

⇝ chtěli bychom navrhnout algoritmy / metody, které převedou A na horní-△, ale za použití unitárních maticových transformací, tj. tak, že jejich aplikací neměníme normu (nebo ne příliš).