Vybrané kapitoly z numerickej algebry 2020/21

- Kurz je pokračovaním kurzov 1-MAT-270 MATICOVÝ POČET
  1-MAT-530 NUMERICKÉ METÓDY LIN. ALGEBRY

Má nahradiť pôvodné kurzy Vybrané kapitoly z num. algebry, METÓDY riešenia sústav s riedkymi maticami a špeciálne matice a algoritmy.

→ nerealistické aby sa pokrylo všetko, trochu iný obsah.

Plán:

Prvá časť kurzu (cca. 6 prednášok) - lekcie 24-40 z knihy Numerical Linear Algebra, Trefethen-Bau

Druhá časť kurzu (cca. 4. prednášky) - riedke matice (podľa kurzu J. Saada - U of Minnesota)

Tretia časť kurzu (cca. 3 prednášky) - moderné metódy (randomizovaných) založené na pravdepodobnostných výpočtoch

---

knižka od Trefethena a Baua je zaujímavá aj v prvej časti (kapitoly 1-23), môže byť zaujímavé sa k nej vrátiť a prelistovať si. Niektoré odvolávky budú explicitné v našej prednáške.

- web: thales.dca.fmph.uniba.sk/niepel/MC/matice3.html

- Google Classroom

(Trefethen-Bau, lekcia 24)

Problémy vlastných hodnôt

- zväčša len opakovanie, dôkazy povačšinou boli na predošlých kurzoch

Vlastné hodnoty & vlastné vektory

$A \in \mathbb{C}_{m\times m}$ - štvorcová $m\times m$ matica, $x \in \mathbb{C}^m$, $\lambda \in \mathbb{C}$

$$Ax = \lambda x$$

x - vl. vektor
λ - vl. hodnota

→ akcia A na nejakom podpriestore vyzerá ako násobenie skalárom - vlastný podpriestor.

Množina vl. hodnôt - spektrum.

- Kde sa vlastné hodnoty a vl. vektory vyskytujú?
- Obdĺžnikové $m\times n$ matice zodpovedajú zobrazeniam $\mathbb{R}^n \to \mathbb{R}^m$, t.j. priestory sú v zásade rôzne, tu o vl. hodnotách nemá zmysel hovoriť
- Pre vl. hodnoty potrebujeme aby štartovací a cieľový priestor boli rovnaké → to zodpovedá problémom, kde sa matica A používa iterovane - t.j. mocnina A^k, alebo ako funkcia e^{tA} (~~e.g.~~ napr. pomocou Taylorovho radu, kde sčítavame mocniny A)

- prečo sú užitočné?

Decoupling - zníženie dimenzionality problému
- rozloženie viacrozmerného problému na viacero skalárnych (1-dim) problémov

Vhľad do kvalitatívneho správania systémov
- stabilita riešení (rast/útlm)
- frekvenčná analýza (rezonancie, kmity)

Spektrálny rozklad matice

$$A = X \Lambda X^{-1}$$

(nemusí vždy existovať, súvis s diagonalizovateľnosťou)

Ekvivalentne
$$AX = X\Lambda$$

po stĺpcoch

$$\begin{bmatrix} A \end{bmatrix} \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix},$$ čo

dáva
$$A x_j = \lambda_j x_j.$$

Tento rozklad udáva zmenu súradníc vzhľadom na bázu z vlastných vektorov.

Riešením $Ax = b$ pre $A = X\Lambda X^{-1}$ máme

$$(X^{-1}b) = \Lambda(X^{-1}x)$$

– v súradniciach vzhľadom na bázu tvorenú vl. vektormi sa posiela
$$X^{-1}x \longmapsto \Lambda(X^{-1}x) = X^{-1}b$$

(t.j. násobenie diagonálnou maticou).

Geometrická násobnosť

Vl. vektory pre vl. hodnotu λ spolu s $\vec{0}$ tvoria vlastný podpriestor E_λ. To je príklad <u>invariantného podpriestoru</u>

$$AE_\lambda \subseteq E_\lambda.$$

– jeho dimenzia – geometrická násobnosť vl. hodn. λ.

– $\dim N(A - \lambda I)$

– max. počet lin. nez. vl. vektorov pre λ.

Charakteristický polynóm

môžeme zvoliť rôzne konvencie

$$\chi_A(\lambda) = \det(A - \lambda I)$$

$$p_A(z) = \det(zI - A)$$

→ výhodou druhej je vždy +1 pri vedúcom člene - monický polynóm

Tvrdenie: λ je vlastná hodnota $\Leftrightarrow$ $p_A(\lambda) = 0$.

→ Dôsledok Reálne matice mávajú komplexné vl. hodnoty.

-> fyzikálne: reálne dynamické systémy môžu mať oscilatórne správanie (okrem rastu/útlmu).

-> algoritmicky: vstup výpočtu je reálny, ale výstup môže byť komplexný -> algoritmus to musí zohľadniť na niektorom mieste.

Algebraická násobnosť

Základná veta algebry:

$$p_A(z) = (z-\lambda_1)(z-\lambda_2)\ldots(z-\lambda_n) \quad \text{pre } \lambda_j \in \mathbb{C}$$

-> algebraická násobnosť vl. hodnoty je jej násobnosť ako koreňa char. polynómu.

-> $n \times n$ matica má n vl. hodnôt (počítaných s násobnosťami), špeciálne, má aspoň jednu.

Alg. násobnosť $\geq$ geom. násobnosť, ale to dokážeme pomocou podobnostnej transformácie.

Podobnosť matíc

Pre regulárnu $X \in M_{n,n}(\mathbb{C})$ sa zobrazenie

$$A \longmapsto X^{-1}AX$$

nazýva podobnostná transformácia matice A.

A, B sú podobné, ak medzi nimi existuje podobnostná transformácia. → ? idea t.j. $B = X^{-1}AX$

– ide o zmenu bázy.

Matice X a $X^{-1}AX$ majú spoločné vlastnosti.

<u>Tvrdenie</u> Pre regulárnu X majú A a $X^{-1}AX$ rovnaké char. polynómy, vl. hodnoty s rovnakými alg. a geom. násobnosťami.

<u>Tvrdenie</u> Alg. násobnosť vl. hodnoty λ je aspoň taká veľká ako jej geom. násobnosť.

<u>Dôkaz:</u> Nech n je geom. násobnosť vl. hodnoty λ matice A. Majme $m \times n$ maticu $\hat{V}$ – jej stĺpce tvoria ortonormálnu bázu $E_\lambda = \{x \mid Ax = \lambda x\}$. Rozšírme $\hat{V}$ na štvorcovú unitárnu maticu V (t.j. pridajme $m-n$ ortonormálnych vektorov z $\hat{V}^\perp$). Potom $V^{-1} = V^*$ a máme $B = V^*AV$.

Potom

$$B = V^* A V = \begin{bmatrix} \lambda I & C \\ 0 & D \end{bmatrix}$$

$A\hat{V}$

Pre char. polynóm matice B máme: $\det(zI - B) = \det(zI - \lambda I)\det(zI - D) = (z-\lambda)^n \det(zI - D)$,

teda alg. násobnosť λ pre maticu B (a aj A) je aspoň n. ☐

- Nediagonalizovateľné matice
- diagonalizovateľnosť

Vzťah medzi determinantom, stopou a vl. hodnotami

Tvrdenie $\det(A) = \prod_{j=1}^{m} \lambda_j \qquad tr(A) = \sum_{j=1}^{m} \lambda_j$

Unitárna diagonalizovateľnosť

Niekedy $m \times m$ matica má nielen m LN vl. vektorov, ale dajú sa vybrať tak, že sú navzájom kolmé.

V tom prípade je A unitárne diagonalizovateľná:

$$A = Q \Lambda Q^*$$

(teda modulo znamienka komplexné ta máme aj SVD rozklad matice A).

Tvrdenie Hermitovské matice sú unitárne diagonalizovateľné, vl. hodnoty sú reálne.

Tvrdenie Matica je unitárne diagonalizovateľná $\Longleftrightarrow$ je normálna $(AA^* = A^*A)$.

Schurova rozklad

Tento rozklad bude z numerického hľadiska najužitočnejší, lebo postihuje všetky matice – aj nediagonalizovateľné.

Schurov rozklad:

$$A = QTQ^*,$$

kde Q je unitárna a T horná trojuholníková.

A a T sú podobné, teda vl. hodnoty A budú na diagonále T.

<u>Veta</u> každá štvorcová matica má Schurov rozklad.

<u>Dôkaz</u> Indukciou vzhľadom na m.

Prípad $m=1$ triviálne platí, nech $m \geq 2$.

Ak je x vl. vektor pre vl. hodnotu λ, normalizujme ho a doplňme ortonormálnou bázou $x^\perp$ na unitárnu U

potom $U^*AU = \begin{bmatrix} \lambda & B \\ 0 & C \end{bmatrix}$ $\qquad U = [\, x \; \underbrace{\cdots}_{\text{báza } x^\perp} \,]$

Podľa IP existuje Schurov rozklad matice $C = VTV^*$

Potom pre $Q = U \begin{bmatrix} 1 & 0 \\ 0 & V \end{bmatrix}$ máme

$$Q^*AQ = \begin{bmatrix} \lambda & BV \\ 0 & T \end{bmatrix}.$$

Videli sme rozklady matice, ktoré "odhaľujú" ich vl. hodnoty

- diagonalizácia $A = X\Delta X^{-1}$ - pre diagonaliz.
- unitárna diagonaliz. $A = Q\Delta Q^*$ - pre normálne
- unitárna triangularizácia (Schurov rozklad) $A = QTQ^*$ $\rightarrow$ vždy

- Na nájdenie vl. hodnôt budeme potrebovať zostrojiť niektorý z týchto rozkladov. - Spravidla Schurov rozklad, lebo je všeobecný
- Unitarita $\rightarrow$ stabilita výpočtu

(Trefethen - Bau, Lekcia 25)

Prehľad algoritmov pre vlastné hodnoty

Toto bude prehľadová lekcia, ktorá uvedie základný pohľad na klasické "priame" algoritmy výpočtu vl. hodnôt a vl. vektorov, a tiež ich moderné(jšie) varianty.

Väčšina algoritmov postupuje v dvoch krokoch

- predbežná redukcia z všeobecného na štruktúrovaný tvar matice
- iteratívny proces ~~pre~~ konvergujúci k riešeniu

Táto časť prednášky vysvetlí prečo je takýto postup vhodný a tiež sa dotkne niektorých "filozofických" otázok numerickej analýzy.

Zlyhania prvoplánových algoritmov

Ukazuje sa, že napriek ľahkej definícii vl. hodnôt a vl. vektorov nie sú efektívne metódy ich výpočtu zďaleka zjavné.

- nájsť char. polynóm → korene

(odkaz ~~prednáška~~ 15) úloha na DÚ)

Tento postup však nie je vhodný, lebo hľadanie koreňov polynómov je zle podmienený problém, aj keď samotný problém hľadania vl. hodnôt ~~[illegible]~~ je dobre podmienený.

(V skutočnosti, hľadanie koreňov polynómov je síce klasický "numerický problém" a podľa kurzov by sa mohlo zdať, že aj v súčasnosti významný, prax sa mu však snaží vyhýbať, práve kvôli zlej podmienenosti — uvažujte korene polynómov stupňa ≥ 2 (?))

- Iná myšlienka: mocninová ~~metóda~~ iterácia
  postupne hľadajme limitu postupnosti
  $$\frac{x}{\|x\|},\ \frac{Ax}{\|Ax\|},\ \frac{A^2x}{\|A^2x\|},\ \frac{A^3x}{\|A^3x\|},$$
  tá konverguje, za mierných predpokladov, k vl. vektoru ~~k~~ najväčšej vl. hodnote (abs. hodnote).
  - opäť - táto metóda je známa, ale, žiaľ, vo všeobecnosti príliš pomalá

Tak, ako sa to naznačilo v predošlej ~~pre~~ lekcii, najlepšie všeobecné algoritmy sú založené na nájdení rozkladov "odhaľujúcich" vl. hodnoty — vl. hodnoty sú zložky niektorej matice v rozklade.

Ako taký algoritmus funguje?
Treba nájsť postupnosť transformácií matice A, ktoré vyrobia nuly na potrebných miestach.
Teda ide o podobný princíp, ako pri GEM, QR, MNŠ (metóda najmen. štvorcov)!
Algoritmy numerickej lineárnej algebry sú postavené na jednej technike, ktorá sa používa znova a znova:
"vynulujme zložky matice!!!"

Podstata zložitosti

Na rozdiel od spomenutých algoritmov, pri hľadaní vlastných hodnôt narazíme na fundamentálny algebraický problém, ktorému musíme venovať pozornosť.

Príklad Matica

$$\begin{bmatrix} -z & & & & -a_0 \\ 1 & -z & & & -a_1 \\ & 1 & -z & & -a_2 \\ & & & & \vdots \\ & & & -z & -a_{m-2} \\ & & & 1 & -z-a_{m-1} \end{bmatrix}$$

(sprievodná matica)

má determinant $p(z) = (-1)^m \left(z^m + a_{m-1}z^{m-1} + \dots + a_1 z + a_0\right)$

Teda matica

$$\begin{bmatrix} 0 & & & & -a_0 \\ 1 & 0 & & & -a_1 \\ & 1 & 0 & & \vdots \\ & & 1 & & -a_{m-2} \\ & & & 1 & -a_{m-1} \end{bmatrix}$$

má vlastné hodnoty práve korene polynómu $p(z)$.

(Dá sa overiť aj priamo, ľavý vl. vektor by bol $(1, z, z^2, \dots, z^{m-1})^T$)

Problém vl. hodnôt teda ~~musí byť rovnako zložitý~~ (z algebraického hľadiska) ~~ako~~ v sebe obnáša hľadanie koreňov ľubovoľného polynómu.

Veta (Abel, 1824) pre $m \geq 5$ existuje polynóm s racionálnymi koeficientami, ktorého korene sa nedajú vyjadriť ako funkcie ~~zo~~ koeficientov, sčítania, odčítania, násobenia, delenia, brania n-tých odmocnín.

(lebo A_m sú jednoduché)

Preto, ak by sme pracovali v presnej aritmetike, alebo v symbolickom výpočte, tak neexistuje žiaden algoritmus, ktorý by vedel vyprodukovať presné ~~riešenie~~ korene v konečnom počte krokov

(Na rozdiel od QR, MNŠ, GEM, ktoré sú konečné a presné.)

~~Keď~~ Následne, niečo podobné platí aj pre problém výpočtu vlastných hodnôt matíc (s rac. alebo celočíselnými zložkami)
Dobré algoritmy na hľadanie vl. hodnôt však existujú
- máme v MATLABE "eig" a funguje...
Takže tu si treba uvedomiť základný rozdiel oproti predošlým algoritmom - tie boli konečné a presné.
teda v presnej aritmetike sme sa dopracovali po konečnom počte krokov k presnému riešeniu.

na druhej strane,
ľubovoľný hľadač vl. hodnôt musí byť iteratívny (t.j. v zásade nekonečný)

Cieľom teda je nájsť postupnosť čísel, ktorá konverguje rýchlo k vl. hodnotám. Keď klesneme pod strojovú presnosť, výpočet môžeme useknúť. A ako uvidíme, toto sa deje relatívne rýchlo.

Na tomto mieste je možno vhodné odkázať na esej v Appendixe : Aká je definícia numerickej analýzy?
odpoveď by mala zahŕňať základnú "misiu" numeriky:
"vypočítať veličiny, ktoré sú typicky nevypočítateľné, z algebraického/analytického pohľadu, a urobiť to rýchlosťou blesku!"

Potreba použitia iteračných metód sa môže zdať odrádzajúca, veď ~~sme~~ sme naučení "dosadzovať do vzorca", ale algoritmy naozaj konvergujú rýchlo. Predstava - počet presných číslic sa zdvoj- až strojnásobuje v každom kroku výpočtu, teda v 16-cifernej aritmetike nám stačí 5-6 iterácií.

Schurov rozklad a diagonalizácia

Väčšina bežných algoritmov pre výpočet vl. hodnôt používa Schurov rozklad.

Rozklad
$$A = Q T Q^*$$
nájdeme postupnosťou unitárnych podobnostných transformácií
$$X \longmapsto Q_j^* X Q_j,$$
tak, aby súčin
$$\underbrace{Q_j^* Q_{j-1}^* \cdots Q_2^* Q_1^*}_{Q^*} A \underbrace{Q_1 Q_2 \cdots Q_j}_{Q}$$
konvergoval k hornej trojuholníkovej T pre $j \to \infty$.

Pre reálnu A (ale nie symetrickú) môžeme mať komplexné vl. hodnoty výpočet sa však dá robiť reálne (s ortogonálnymi Q_j), potom T bude mať na diagonále 2×2 bloky zodpovedajúce páru vl. hodnôt λ a $\bar{\lambda}$.

Na druhej strane, Ak je A hermitovská, potom $Q_j^* \cdots Q_2^* Q_1^* A Q_1 Q_2 \cdots Q_j$ je tiež hermitovská, takže limitná matica bude horná troj & hermitovská ⇒ reálna diagonálna.

Takže všeobecný algoritmus rátajúci Schurov rozklad dá aj (ale) spektrálny rozklad Hermitovskej matice.

- existujú však urýchlenia, ktoré využívajú symetriu a spektrálny rozklad Hermitovskej matice sa dá nájsť o čosi rýchlejšie.

Na úvod tejto lekcie sme sľubovali

Dve fázy výpočtu vl. hodnôt

V prvej pomocou priamej metódy z matice A vyrobíme maticu H v hornom Hessenbergovom tvare, t.j. matica s nulami pod prvou vedľajšou dolnou diagonálou

$$h_{ij} = 0 \quad \text{ak} \quad i > j+1$$

V druhej fáze budeme iterovať a vytvoríme postupnosť (v princípe nekonečnú) horných Hessenbergových matíc, ktorých zložky pod diagonálou ($h_{i,i-1}$) konvergujú k nule, teda $H \longrightarrow T$ ide k hornej trojuholníkovej.

Schematicky:

$$\underset{A}{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}} \xrightarrow{\text{Fáza 1}} \underset{H}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}} \xrightarrow{\text{Fáza 2}} \underset{T}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}}$$

(tu $A \neq A^*$, t.j. nemáme Hermitovskú maticu)

Počas prvej fázy sa vykoná $O(m^3)$ operácií (flop) (floating point operation).

Druhá fáza je potenciálne nekonečná, teda počet operácií by ~~vyžadoval~~ bol tiež nekonečný. V praxi je však počet iterácií do dosiahnutia strojovej presnosti $O(m)$, každá z iterácií (vďaka vhodnému tvaru) vyžaduje $O(m^2)$ flopov, teda celkovo máme $O(m^3)$ flopov aj v druhej fáze.

Pozn. Ak by sme nespravili prípravnú fázu č. 1, každá z iterácií vo fáze 2 by vyžadovala $O(m^3)$ flopov a celkovo by

Sme potrebovali $O(m^4)$ operácií (alebo viac, lebo by konvergencia by sa mohla "pokaziť" a trvať dlhšie ako $O(m)$.) [9.7

[Toto nevyzerá ako veľký problém – $O(m^3)$ vs. $O(m^4)$ – to sú len polynomiálne veľkosti, ale pri naozaj veľkých dátach to znamená rádové rozdiely v tom, čo sa dá spočítať

výpočtová sila vzrástla medzi 1950 – 2000 násobkom 10^9 od 2020 a [illegible]
Maticové problémy 10^3, reprezentujúce práve "$O(m^3)$" – ~~[illegible]~~ (pozri aj v Lekcii 32)

V Hermitovskom prípade je dvoj-fázový prístup ešte rýchlejší. Medzistupňová matica je hermitovská horná Hessenbergova → je tridiagonálna. Výsledná matica je, ako sme už videli, hermitovská a horná trojuholníková → diagonálna.

$$\underset{A=A^*}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\text{Fáza 1}} \underset{T}{\begin{bmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}} \xrightarrow{\text{Fáza 2}} \underset{D}{\begin{bmatrix} \times & & & & \\ & \times & & & \\ & & \times & & \\ & & & \times & \\ & & & & \times \end{bmatrix}}$$

Ak sa počítajú len vl. hodnoty (a nie vektory) potom každá z iterácií vo fáze 2 vyžaduje iba $O(m)$ flopov, teda celkovo ~~vo~~ fáze 2 budeme potrebovať len $O(m^2)$ flopov.

Teda v tomto prípade sa ukazuje paradox, že priama (konečná) fáza výpočtu sa vyžaduje $O(m^3)$ flopov ale "nekonečná" iba $O(m^2)$, – teda je rádovo rýchlejšia ako konečná

(Trefethen-Bau, lekcia 26)

Redukcia na Hessenbergov alebo tri-diagonálny tvar.

V tejto časti prednášky si ukážeme, čo treba robiť v prvej, prípravnej fáze, ~~*~~ ako previesť maticu z všeobecného tvaru na Hessenbergov. Ako sme už videli, unitárne transformácie dajú v prípade hermitovskej matice tridiagonálny tvar.

Nesprávna myšlienka

Pri hľadaní Schurovho rozkladu $A = QTQ^*$ by sme chceli pomocou unitárnych podobnostných transformácií v matici A vynulovať zložky pod diagonálou.
Prvotný nápad by mohol byť podobný ako pri Householderovej triangularizácii (pozri Lekcia 10, str. 69; resp. MNLA).
Tam sme pomocou Householderových reflektorov postupne nulovali pozície pod diagonálou:

Q_1^* vynuluje zložky v prvom stĺpci $\quad x \mapsto \|x\| e_1$

ale zároveň zmení aj zvyšné stĺpce:

$$\underset{A}{\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix}} \xrightarrow{Q_1^* \cdot} \underset{Q_1^* A}{\begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix}}$$

(zmenené pozície sú značené farebne)

Na podobnostnú transformáciu musíme však násobiť aj sprava Q_1

$$\underset{Q_1^* A}{\begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix}} \xrightarrow{\cdot Q_1} \underset{Q_1^* A Q_1}{\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix}}$$

- čo zodpovedá stĺpcovým operáciám, teda nuly v prvom stĺpci sa stratia

Samozrejme, my už vieme, že takto to fungovať nemôže

- z dôkazu Schurovej lemmy vieme, že prvý stĺpec Q by mal byť vlastný vektor (ktorý nepoznáme), pričom prvý stĺpec Q_1 je len normovaný prvý stĺpec matice A
- tiež z predchádzajúcej lekcie vieme, že pri počítaní vl. hodnôt narážame na fundamentálny algebraický problém – žiadna metóda nemôže dať vl. hodnoty presne v konečnom čase.

Ako však uvidíme neskôr, takýto postup sa ukáže užitočný, lebo zložky pod diagonálou v matici $Q_1^* A Q_1$ budú menšie (v abs. hodnote) ako zložky A.

Správna myšlienka

Pre fázu 1 sa ukazuje ako správny menej ambiciózny cieľ – treba zaútočiť iba na menšie územie, ktoré zvládneme ubrániť.

V prvom kroku sa zvolí Householderov reflektor, ktorý nemení prvý riadok (t.j. Q_1 nemení prvý stĺpec). Chceme, aby Q_1^* kombinovala iba riadky $2, 3, \dots, m$ tak, aby v prvom stĺpci vznikli nuly na pozíciách $3, \dots, m$. Potom máme:

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{Q_1^* \cdot} \underset{Q_1^* A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\cdot Q_1} \underset{Q_1^* A Q_1}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}}$$

Túto myšlienku môžeme zopakovať na dosiahnutie núl v nasledujúcich stĺpcoch. Householderov projektor Q_2^* bude fixovať prvé dva riadky a vyrobí nuly ~~od tre~~ štvrtého nižšie

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}}_{Q_1^* A Q_1} \xrightarrow{Q_2^* \cdot} \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{bmatrix}}_{Q_2^* Q_1^* A Q_1} \xrightarrow{\cdot Q_1} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{bmatrix}$$

Po m-2 opakovaniach dostaneme maticu v Hessenbergovom tvare:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}$$

$$\underbrace{Q_{m-2}^* \cdots Q_2^* Q_1^*}_{Q^*} A \underbrace{Q_1 Q_2 \cdots Q_{m-2}}_{Q} = H$$

Algoritmus:

for $k = 1$ to $m-2$

$\quad x = A_{k+1:m,\,k}$

$\quad v_k = \mathrm{sign}(x_1)\, \|x\|_2 e_1 + x$ (toto je vektor udávajúci reflexiu cez $v_k^\perp$)

$\quad v_k = v_k / \|v_k\|_2$

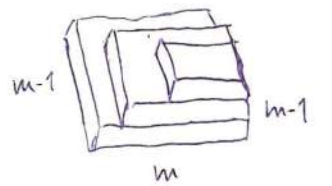
$\quad A_{k+1:m,\,k:m} = A_{k+1:m,\,k:m} - 2 v_k (v_k^* A_{k+1:m,\,k:m})$ ← násobenie Q_k^* zľava

$\quad A_{1:m,\,k+1:m} = A_{1:m,\,k+1:m} - 2 (A_{1:m,\,k+1:m} v_k) v_k^*$ ← násobenie Q_k sprava

Podobne ako v Lekcii 10, maticu Q nemusíme počítať explicitne, stačí si pamätať vektory v_k, pomocou ktorých sa Q dá zistiť dodatočne. (pozri Householderovu triangularizáciu v Lekcii 10)

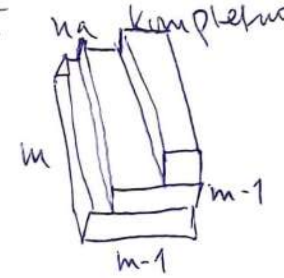
Počet operácií

V knižke sa zvykne počet operácií počítať geometricky. Na jedno políčko treba zhruba 4 flopy, takže sa pozrieme koľko políčok sa zmení v priebehu dvoch zmien (updatov) matice. k-ty reflektor Q_k^* zľava mení m-k posledných riadkov, tie však majú prvých k-1 pozícií nulových, takže nám zostáva m-k+1 zložiek v každom z riadkov:



Pre $m \to \infty$ objem bude $\frac{1}{3} m^3$, čo dáva $\sim \frac{4}{3} m^3$ flopov.

V druhej zmene, pri násobení Q_k ~~sprava~~ sprava kombinuje m-k posledných stĺpcov, ale teraz už nemáme nuly naľavo, tak operácie treba vykonať na kompletných m zložkových stĺpcoch



Pre $m \to \infty$ to je $\frac{1}{2} m^3$, teda to dá $\sim 2m^3$ flopov.

Celkovo na unitárnu redukciu $m \times m$ matice na Hessenbergov tvar treba $\sim \frac{10}{3} m^3$ flopov.

Hermitovský prípad

V hermitovskom prípade predošlý algoritmus dá vo výsledku tridiagonálnu maticu. Teda nuly vznikajú v riadkoch ale aj stĺpcoch. – niektoré zložky netreba počítať. Teda reflektor zľava aj sprava budú rovnako drahé, čím dosiahneme pre pravý update úsporu z $2m^3$ na $\frac{4}{3} m^3$. To dá spolu $\frac{8}{3} m^3$. Táto úspora však vyplynula len z riedkosti matice (nuly nad diagonálou), dodatočné šetrenie dostaneme využitím symetrie - to dá faktor 2, čím sa v hermitovskom prípade dostaneme k odhadu počtu operácií

$$\sim \frac{4}{3} m^3 \text{ flopov} \quad \left(\begin{array}{c}\text{tridiagonálna}\\ \text{redukcia}\end{array}\right)$$

(bez detailov implementácie)

Stabilita pojmy ~~[illegible]~~ stability a (~~forward~~ stability)

~~[illegible]~~ spätnej stability (backward stability)

sú rozpracované v ~~[illegible]~~ lekciách 14, a 15. Potom v lekcii 16 sa analyzuje (spätná) stabilita Householderovej redukcie. Je pravdepodobné, že k pojmu stability sa ešte budeme musieť vrátiť, ale teraz zakončíme prednášku len tvrdením o stabilite predstaveného algoritmu:

Nech $\tilde{H}$ je vypočítaná matica v Hessenbergovom tvare (používajúc floating point aritmetiku) a $\tilde{Q}$ je unitárna matica $\tilde{Q} = \tilde{Q}_1 \tilde{Q}_2 \cdots \tilde{Q}_{m-2}$ zodpovedajúca vektorom odrazu/reflexie $\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_{m-2}$ (podľa algoritmu)

Potom platí:

Tvrdenie Presná Hessenbergova redukcia $A = QHQ^*$ a vypočítané faktory $\tilde{Q}, \tilde{H}$ spĺňajú:

$$\tilde{Q}\tilde{H}\tilde{Q}^* = A + \delta A, \quad \text{pre} \quad \frac{\|\delta A\|}{\|A\|} = O(\varepsilon_{machine})$$

pre nejaké $\delta A \in M_{m,m}(\mathbb{C})$.