Cvičení 4 - zadání a řešení úloh

Základy numerické matematiky - NMNM201 Verze z 25. října 2023

1 Gram-Schmidtův ortogonalizační proces

1.1 Připomenutí Gram-Schmidtova ortogonalizačního procesu

Gram-Schmidtův ortogonalizační proces je metoda, která z lineárně nezávislé posloupnosti vektorů $a_1, \dots, a_m$ vytvoří posloupnost ortonormálních vektorů $q_1, \dots, q_m$ tak, že platí

$$\mathrm{span}\{a_1, \dots, a_k\} = \mathrm{span}\{q_1, \dots, q_k\}$$

pro všechna $k = 1, \dots, m$.

Nechť již jsme získali $q_1, \dots, q_{k-1}$ pro nějaké k. Popíšeme nyní, jak získáme vektor q_k.

Od vektoru a_k postupně odečítáme jeho projekce na podprostory generované jednotlivými vektory již spočtené ortonormální báze:

$$z = a_k - \sum_{i=1}^{k-1} q_i q_i^* a_k = (I - \sum_{i=1}^{k-1} q_i q_i^*) a_k. \tag{1}$$

Výsledný vektor z pak normalizujeme a získáme tak q_k:

$$q_k = z/\|z\|. \tag{2}$$

Označíme-li $r_{i,k} = q_i^* a_k$ a $r_{k,k} = \|z\|$, dosazením do (1) za z vektor $r_{k,k} q_k$ (viz (2)), dostaneme

$$a_k = \sum_{i=1}^{k-1} r_{i,k}\ q_i + r_{k,k}\ q_k, \quad k = 1, \dots, m,$$

neboli $A = QR$.

1.2 Implementace Gram-Schmidtova procesu

Gram-Schmidtův proces lze přepsat několika matematicky ekvivalentními způsoby.

1) Klasický algoritmus (CGS):

$$z = a_k - \sum_{i=1}^{k-1} q_i q_i^* a_k. \tag{CGS}$$

2) Modifikovaný algoritmus (MGS) odpovídá postupné ortogonalizaci vektoru a_k:

$$z = (I - q_{k-1} q_{k-1}^*) \dots (I - q_2 q_2^*)(I - q_1 q_1^*) a_k \tag{MGS}$$

3) Klasický algoritmus s iteračním zpřesněním (ICGS), kdy stačí jediné opakování ortogonalizace (takže ortogonalizaci provedeme celkem dvakrát).

2 QR rozklad

2.1 QR rozklad a ztráta ortogonality

Definice 1 (QR rozklad)**.** *Nechť $A \in \mathbb{R}^{n \times m}$ je obecná obdélníková matice. Rozklad tvaru*

$$A = QR,$$

kde Q je matice s ortonormálními sloupci a R je horní trojúhelníková, nazýváme QR rozkladem matice A.

Úloha 1. *Sledujte ztrátu ortogonality a přesnost QR rozkladu pro různé implementace rozkladu.*

1. *Úpravou skriptu* `cgs.m` *(klasická implementace GSO) vytvořte skript* `mgs.m` *pro modifikovanou GSO.*

2. *Budete-li mít čas navíc, naimplementujte i* `icgs.m`*, iterovanou klasickou GSO. Volte dvě opakování.*

3. *Na základě přednášky si rozmyslete, jaké lze očekávat normy $\|A - QR\|$ a $\|Q^*Q - I\|$ pro různé implementace QR rozkladu a doplňte je do skriptu* `srovnej_QR.m` *(na řádky 66–71). Hodnoty si poté můžete zkontrolovat v tabulce níže.*

4. *Než skript spustíte, projděte si ho a ujistěte se, že rozumíte jednotlivým příkazům. Pokud ne, zeptejte se cvičícího.*

5. *Skript spusťte a zamyslete se nad výsledky. Zejména odpovězte na otázky:*
   - *Je, dle vašich pozorování, norma rezidua $\|A - QR\|$ vypovídající o „kvalitě" spočteného QR rozkladu?*
   - *Lze pomocí nějaké varianty QR rozkladu získat téměř ortogonální faktor Q i pro matici A s vysokým číslem podmíněnosti?*
   - *Jsou dosažené výsledky v souladu s teoretickými výsledky, viz tabulka?*

Algoritmus	$\|Q^*Q - I\|$	$\|A - QR\|$
CGS	$\kappa^2(A)\varepsilon$	$\varepsilon\|A\|$
MGS	$\kappa(A)\varepsilon$	$\varepsilon\|A\|$
ICGS	ε	$\varepsilon\|A\|$
Householder QR rozklad	ε	$\varepsilon\|A\|$
Givens QR rozklad	ε	$\varepsilon\|A\|$

Řešení.

1. Jak je vidět už z tabulky, norma $\|A - QR\|$ je obvykle malá a nesouvisí s (možnou) ztrátou ortogonality v Q.

2. Ano, při použití ICGS, HH, či Givense lze získat matici Q s ortogonálními sloupci nezávisle na podmíněnosti matice A.

3. Očekávání z tabulky je možná někdy nadsazené, ale závislost na čísle podmíněnosti A je pozorovatelná.

□

2.2 Řešení soustavy lineárních rovnic QR rozkladem

QR rozklad lze použít při řešení soustavy lineárních rovnic s (regulární) maticí A a vektorem pravé strany b:

$$Ax = b \Leftrightarrow QRx = b \Leftrightarrow Rx = Q^*b \tag{3}$$

Úloha 2. *Ukažte, že matice R dědí mnohé vlastnosti původní matice A, například:*

$$\|R\|_2 = \|A\|_2,\ \|R\|_F = \|A\|_F,\ \kappa(R) = \kappa(A)\,.$$

Řešení. První dvě tvrzení jsou triviálním důsledkem invariance norem. Alternativně lze argumentovat i skrze definici maticové normy:

- $\|A\|_2 = \max_{\|x\|=1} \|Ax\|_2$, tedy záleží na velikosti dvojkové vektorové normy, která se přenásobením vektoru unitární maticí nemění.
- Frobeniova maticová norma je odmocninou součtu kvadrátů velikosti dvojkových norem sloupců matice, argumentace je tak stejná, jako v předchozím případě, sloupce výsledné matice R mají stejnou normu jako matice A a tedy Frobeniova norma matice R je stejná jako matice A.

Poslední tvrzení:
$$\kappa(A) = \|A\|\|A^{-1}\| = \|QR\|\|R^{-1}Q^*\| = \|R\|\|R^{-1}\| = \kappa(R).$$

□

Praktická implementace řešení soustavy $Ax = b$ se typicky vyhýbá výpočtu Q a přenásobení pravé strany. Spíše se při řešení soustavy počítá QR rozklad rozšířené matice $[A \,|\, b]$.

$$[A|b] \xrightarrow{\text{QR rozklad}} \text{matice } \bar{Q} \text{ a } [\bar{R}|\bar{b}]; \quad [A|b] = \bar{Q}\,[\bar{R}|\bar{b}]. \tag{4}$$

Úloha 3. *Uvažujte regulární reálnou matici A (pak Q, $\bar{Q} \in \mathbb{R}^{n\times n}$) a rozmyslete si následující otázky*

1. *V přesné aritmetice je řešení soustav $\bar{R}x = \bar{b}$, $Rx = Q^*b$ a $Ax = b$ stejné.*
2. *Pro libovolnou danou implementaci QR rozkladu v přesné aritmetice platí*

$$Q = \bar{Q}, \quad R = \bar{R}, \quad Q^*b = \bar{b}.$$

Řešení.

1. Mezi jednotlivými rovnicemi lze přecházet pomocí unitárních transformací.
2. Stačí si rozmyslet, že všechny varianty QR rozkladu lze interpretovat tak, že postupně transformuji sloupce matice A (od prvního po poslední). Proto přidání posledního sloupce nijak neovlivňuje výpočet těch předcházejících sloupců (a výpočet souvisejících ortogonalizačních koeficientů, respektive výsledek unitárních transformací pomocí matic Householderovy reflexe či Givensových rotací.

□

Úloha 4. *Může se zdát, že MGS je ve všech směrech lepší než CGS, v praxi se však v jistých případech stále používá. Dokážete přijít na jistou implementační výhodu CGS?*

[Hint: Představte si, že ve více lidech počítáte ručně QR rozklad matice o 100 řádcích a chcete to mít co nejrychleji. (Nebo, že počítáte na počítači s větším množstvím procesorů.)]

Řešení. CGS se výrazně lépe paralelizuje: každou ortogonalizaci ve směru q_i lze počítat nezávisle na jiném procesoru. Tuto část výpočtu je tedy možno značně urychlit. U MGS to není možné, protože začít ortogonalizovat oproti dalšímu vektoru je možné až potom, co jsme dokončili ortogonalizaci vůči vektoru předchozímu. □

Úloha 5. *Zkonstruujte matici* `A = gallery('poisson',100);` *a pomocí vestavěné funkce* `qr` *(nebo jakékoli z „vašich“ implementací) spočtěte její QR rozklad v MATLABu. Následně srovnejte zaplnění (tj. počet nenulových prvků) faktorů Q, R a samotné matice A pomocí příkazu* `spy`.

(Navíc) Srovnejte se zaplněním faktorů L a U z LU rozkladu této matice. Pokud byste měli řešit soustavu lineárních algebraických rovnic $Ax = b$, kde matice A je řídká, kterou metodu byste použili a proč?

Řešení. `A = gallery('poisson',100);`
`[Q, R] = qr(A);`
nnz(Q) $= 50\,994\,582$, nnz(R) $= 1\,979\,821$

Faktor Q na rozdíl od R, L a U již nebude řídký. Pro řídké matice je obvykle výhodnější použít LU rozklad kvůli nižším paměťovým nárokům (faktory L a U se tolik nezaplní).

`[L,U] = lu(A);`
nnz(L) $=$ nnz(U) $= 1\,000\,099$ $\square$