Cvičení 4 - zadání pro studenty

Základy numerické matematiky - NMNM201 Verze z 21. října 2024

1 Gramův–Schmidtův ortogonalizační proces

1.1 Připomenutí Gramova–Schmidtova ortogonalizačního procesu

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

$$\operatorname{span}\{a_1, \ldots, a_k\} = \operatorname{span}\{q_1, \ldots, q_k\}$$

pro všechna $k = 1, \ldots, m$.

Nechť již jsme získali $q_1, \ldots, q_{k-1}$ 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

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

neboli $A = QR$.

1.2 Implementace Gramova–Schmidtova procesu

Gramův–Schmidtův proces lze přepsat dvěma 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:

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

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. *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.*

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

4. *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	$\kappa(A)\varepsilon$	$\varepsilon\|A\|$
Householder QR rozklad	ε	$\varepsilon\|A\|$
Givens QR rozklad	ε	$\varepsilon\|A\|$

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:

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

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

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

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*

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

Poznámka. *Může se zdát, že MGS je ve všech směrech lepší než CGS, v praxi se však CGS stále používá. 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 4. *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?*