Numerická Analýza – Semestrální Práce

1. **Ortogonální podobnostní transformace.**

   Nechť A je čtvercová matice, $A \in \mathbb{R}^{n\times n}$. Napište program v Matlabu, který převede matici A pomocí ortogonálních podobnostních transformací na matici v horním Hessenbergově tvaru, tj. bude platit

   $$QAQ^T = H,$$

   kde Q je ortogonální matice a H je matice ve tvaru

   $$H = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,n-1} & h_{n,n} \\ h_{2,1} & h_{2,2} & h_{2,3} & \dots & h_{2,n} \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & h_{n-1,n-2} & h_{n-1,n-1} & h_{n-1,n} \\ 0 & \dots & 0 & h_{n,n-1} & h_{n,n} \end{bmatrix}.$$

   Ke konstrukci matice Q použijte Householderovy reflexe. Vstupem do funkce bude matice A, výstupem budou matice Q a H. Otestujte funkčnost programu na náhodných maticích. [4 body]

2. **Ortogonální podobnostní transformace symetrické matice.**

   Co se stane, když použijeme program z 1. úlohy na matici A, která je symetrická? Jakou strukturu bude mít matice H? Zdůvodněte teoreticky. [2 body]

3. **Ortogonální transformace.**

   Nechť A je čvercová matice řádu n, $A \in \mathbb{R}^{n\times n}$. Napište program v Matlabu, který převede matici A pomocí ortogonálních transformací aplikovaných zprava a zleva, na matici v horním bidiagonálním tvaru, tj. bude platit

   $$U^T AV = B,$$

   kde U a V jsou ortogonální matice a matice B je ve tvaru

   $$B = \begin{bmatrix} b_{1,1} & b_{1,2} & 0 & \dots & 0 \\ 0 & b_{2,2} & b_{2,3} & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & b_{n-1,n-1} & b_{n-1,n} \\ 0 & \dots & 0 & 0 & b_{n,n} \end{bmatrix}.$$

   Ke konstrukci matic U a V použijte Householderovy reflexe. Vstupem do funkce bude matice A, výstupem budou matice U, V a B. Otestujte funkčnost na náhodných maticích. [4 body]

4. **Příprava kódů na výpočet QR rozkladu.**

   Nechť je dána obdélníková matice $A \in \mathbb{R}^{n\times m}$. Napište funkci v Matlabu, která spočte QR rozklad matice A pomocí Householderových reflexí, tj. bude platit

   $$A = QR,$$

   kde $Q \in \mathbb{R}^{n\times n}$ je ortogonální matice a matice $R \in \mathbb{R}^{n\times m}$ je horní trojúhelníková matice. Vstupem do matlabovské funkce bude matice A, výstupem budou matice Q a R. Dále si prostudujte matlabovské funkce, které spočtou QR rozklad matice A pomocí (iterované) verze klasického Gram-Shmidtova procesu (CGS) a (iterované) verze modifikovaného Gram-Shmidtova procesu (MGS). [4 body]

5. **Test numerické stability výpočtu QR rozkladu.**

Uvažujte Lauchliho matici

$$A = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ \delta & 0 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \delta & 0 & 0 \\ \vdots & \ddots & \ddots & \delta & 0 \\ 0 & \dots & 0 & 0 & \delta \end{bmatrix}$$

velikosti 21×20 s parametrem $\delta = 10^{-7}$. Tuto matici získáte matlabovským příkazem

```
A=gallery('lauchli',20,1e-7);
```

Použijte předchozí algoritmy (QR rozklad spočtený pomocí Householderových reflexí, QR rozklad pomocí 4 variant Gram-Schmidtova procesu - CGS, MGS, iterovaný CGS, iterovaný MGS) ke spočtení QR rozkladu Lauchliho matice. Pro každý algoritmus spočtěte ztrátu ortogonality $\|I - Q^T Q\|_F$ a normu rezidua $\|A - QR\|$. Výsledky komentujte. [4 body]

6. **Test rychlosti a přesnosti algoritmů.**

Na dostatečně velkých náhodných maticích porovnejte výsledky spočtené Vašimi programy s výsledky spočtenými matlabovským příkazem

```
[Q,R]=qr(A)
```

z následujících hledisek

(a) čas výpočtu

(b) přesnost výpočtu měřená ztrátou ortogonality $\|I - Q^T Q\|_F$ a normou rezidua $\|A - QR\|$.

Výsledky komentujte. K měření výpočetního času lze v Matlabu použít buď funkce `tic` a `toc`, tj.

```
tic  výpočet  toc
```

nebo funkci `cputime`, např.

```
t=cputime;  surf(peaks(40));  e=cputime-t
```

[3 body]

7. **Řešení soustav lineárních rovnic.**

Uvažujte následující postup. Pro danou regulární matici $A \in \mathbb{R}^{n \times n}$ zvolte vektor x (přesné řešení) jako náhodný vektor, `x = randn(n,1)` a dopočtěte příslušný vektor pravé strany $b = Ax$. Spočtěte číslo podmíněnosti dané matice (příkaz `cond(A)`) a řešte systém lineárních rovnic pomocí:

(a) LU rozkladu s částečnou pivotací (Návod: použijte příkaz `[L,U,P] = lu(A)` , kterým získáte LU rozklad matice PA, kde P je vhodná permutační matice. matice. Z předchozího známe rozklad matice $PA = LU$.

(b) LU rozkladu s částečnou pivotací a iteračního zpřesnění, použijte LU rozklad spočtený v bodě (a) a proveďte dvě iterace iteračního zpřesnění,

(c) QR rozkladu matice A (ke spočtení QR rozkladu použijte příkaz `[Q,R]=qr(A)`),

(d) singulárního rozkladu matice A (ke spočtení singulárního rozkladu použijte příkaz `[W,S,V]=svd(A)`).

Při řešení soustav nezapomeňte využít toho, že inverze ortogonální matice Q je matice Q^T.

Označme x přesné řešení uvažované soustavy (to jste zvolili na počátku) a $\hat{x}$ řešení spočtené uvažovaným algoritmem. Pro jednotlivé způsoby řešení soustavy rovnic spočtěte následující charakteristiky, pomocí nichž lze hodnotit kvalitu spočteného řešení:

- relativní normu reziduua $\dfrac{\|b - A\hat{x}\|}{\|b\|}$,
- relativní normu chyby $\dfrac{\|x - \hat{x}\|}{\|x\|}$,
- velikost normované relativní zpětné chyby $\dfrac{\|b - A\hat{x}\|}{\|A\|\|x\| + \|b\|}$,
- v případě použití LU rozkladu spočtěte velikost růstového faktoru, tj. číslo $\dfrac{\|U\|_\infty}{\|A\|_\infty}$.

Předchozí postup proveďte pro dvě konkrétní testovací matice,

- matici

$$A = \begin{bmatrix} 1 & 0 & \dots & 0 & 1 \\ -1 & 1 & \ddots & \vdots & 1 \\ \vdots & \ddots & \ddots & 0 & 1 \\ -1 & \ddots & -1 & 1 & 1 \\ -1 & \ddots & -1 & -1 & 1 \end{bmatrix}$$

  velikosti 80×80,

- Hilbertovu matici, $A = [a_{i,j}]$, velikosti 10,

$$a_{i,j} = \frac{1}{i + j - 1},$$

  kterou získáte příkazem `A = hilb(10);`

Výsledky získané z numerického experimentu komentujte. [7 bodů]

Poznámka.

Charakteristika

$$\beta(\hat{x}) \equiv \frac{\|b - A\hat{x}\|}{\|A\|\|x\| + \|b\|}$$

se nazývá *normovaná relativní zpětná chyba* a může být interpretována následujícím způsobem. Spočtenou aproximaci řešení $\hat{x}$ si lze představit jako přesné řešení perturbované soustavy

$$(A + \Delta A)\hat{x} = b + \Delta b.$$

Ptáme se, jak moc musíme změnit původní soustavu, aby bylo spočtené řešení přesným řešením modifikované soustavy. Přesněji, ptáme se na velikosti nejmenších možných změn vstupních dat A a b takových, že $\hat{x}$ je přesným řešením perturbované soustavy, přičemž velikosti změn měříme v relativním smyslu pomocí podílů

$$\frac{\|\Delta A\|}{\|A\|} \text{ a } \frac{\|\Delta b\|}{\|b\|}.$$

Lze ukázat, že tyto velikosti nejmenších možných změn vstupních dat lze spočíst právě pomocí čísla $\beta(\hat{x})$. Jinak řečeno, $\beta(\hat{x})$ představuje nejmenší číslo β, pro nějž existují změny ΔA a Δb splňující $\|\Delta A\| \le \beta\|A\|$, $\|\Delta b\| \le \beta\|b\|$ takové, že $\hat{x}$ je přesným řešením perturbované soustavy.

8. **Numerická hodnost matice.**

- Pomocí singulárního rozkladu zjistěte numerickou hodnost r matice ZENIOS. Zjištěnou numerickou hodnost porovnejte s výsledkem příkazu `rank(A);` *Návod:* Numerická hodnost matice je počet singulárních čísel výrazně větších než strojová přesnost. V tomto příkladu považujte za numerickou hodnost počet singulárních čísel větších než tolerance

$$n\mathbf{u}\|A\|,$$

kde $\mathbf{u} \approx 2.2 \times 10^{-16}$, n je velikost matice. Matici načtěte do prostředí Matlabu příkazy

```
load zenios; A = full(spconvert(zenios));
```

Ke spočtení singulárních čísel použijte příkaz `svd`.

- Pouze ze znalosti spočtených singulárních čísel určete velikost $\|A - A_r\|$, kde A je původní matice ZENIOS a A_r je její nejlepší aproximace hodnosti r, kde r je vámi určená numerická hodnost. Poté ověřte numericky, tj. spočtěte $\|A - A_r\|$ pomocí příkazu `norm`.

[4 body]

9. **Aproximace matice pomocí singulárního rozkladu.**

- Pomocí příkazu `[A,map]=imread('obrazek.bmp'); A=double(A);` načtěte obrázek do matice A. Potom příkazem `svd` spočtěte její singulární rozklad. Nechť A_k je nejlepší aproximace matice A hodnosti k,

$$A_k = \sum_{j=1}^{k} \sigma_j u_j v_j^*,$$

kde u_j a v_j jsou levé a pravé singulární vektory matice A, σ_j jsou příslušná singulární čísla. Určete k tak, že se obrázek uložený v matici A_k příliš neliší od obrázku uloženého v matici A (podle Vašeho subjektivního dojmu). K vykreslení obrázku uloženého v matici X lze použít příkaz `imagesc(X), axis image, colormap(map)`.

- Jako testovací obrázek použijte buď černobílý obrázek `strom.bmp` nebo jiné vámi zvolené obrázky. Pokud je vámi zvolený obrázek barevný, je nutné s ním zacházet trochu složitěji. Výše zmíněným příkazem použitým na barevnou fotografii získáte matici A, která má tři rozměry. V matici `X=A(:,:,1)` je uložen červený kanál obrázku, v `Y=A(:,:,2)` je uložen zelený kanál obrázku a v `Z=A(:,:,3)` je uložen modrý kanál obrázku. Místo jedné matice tak musíte pracovat se třemi maticemi, spočíst singulární rozklady matic X, Y, Z, jejich nejlepší aproximace X_k, Y_k a Z_k. Před zobrazením je nutné tyto tři matice poskládat opět do jedné trojrozměrné matice např. B, a zobrazit ji např. příkazem `image(uint8(B))`.

[8 bodů]