Departamento de Matemática – Instituto Superior Técnico

Projeto de Álgebra Linear Numérica

LMAC - 4° Período, 2021/22

Justifique e comente todos os resultados.

I [6 valores]

1. Implemente em Matlab o **método de Gram-Schmidt modificado**. Para tal, deve construir uma função que recebe uma matriz $A \in \mathbb{R}^{m\times n}$ e devolve as matrizes Q e R da decomposição QR da matriz A. [2.5 valores]

2. **Aplicação: Fatorização QR de secções da matriz de Hilbert.**

   Com o objetivo de testar a estabilidade do algoritmo de Gram-Schmidt modificado, aplique-o aos seguintes casos:
   $$A_n \in \mathbb{R}^{100\times n}, \qquad n = 10, ..., 100,$$
   onde cada entrada da matriz A_n é dada por
   $$(A_n)_{ij} = \frac{1}{i+j-1}.$$

   (a) Calcule as normas de Frobenius
   $$\|Q_nR_n - A_n\|_F \text{ e } \|Q_n^\top Q_n - I_n\|_F,$$
   onde $A_n = Q_nR_n$. Comente os resultados. [2 valores]

   (b) Seja $b \in \mathbb{R}^{100}$ definido por
   $$b_i = \sum_{j=1}^{100} \frac{1}{i+j-1}.$$
   A solução exata do sistema $A_{100}x = b$ é o vector $x = [1\,1...1]^\top$. Obtenha a solução do sistema $A_{100}x = b$, usando a decomposição QR. Como explica os resultados obtidos, face à solução que se esperava? [1.5 valores]

   Nota: Pode usar a função `hilb` do Matlab que permite obter imediatamente a matriz de Hilbert.

II [8 valores]

Compressão de Imagens usando o Matlab

Uma imagem pode ser tratada como uma matriz onde cada entrada é um pixel. No caso de imagens em tons de cinzento, tipicamente, cada pixel pode assumir valores entre 0 e 255. Um pixel de valor 0 corresponde ao preto e 255 ao branco. Em imagens a cores, cada pixel possui 3 valores, um para cada quantidade de R (vermelho), G (verde) e B (azul), que variam numa escala de 0 a 255. Isto resulta num total de $256^3 = 16777216$ cores disponíveis.

Para se trabalhar com imagens no Matlab, estas devem estar armazenadas em ficheiros, em formato .jpg, por exemplo, e a pasta que contenha estes ficheiros deve ser referenciada através do Current Folder. A função `imread` do Matlab permite carregar uma imagem. Por exemplo,

```
A = imread('imagem.jpg');
```

As matrizes criadas desta forma são do tipo `uint8`, ou seja, números inteiros de oito bits. Se estivermos a trabalhar com uma imagem a cores, podemos decompô-la em

```
A1=A(:,:,1);
A2=A(:,:,2);
A3=A(:,:,3);
```

de forma obter as camadas de vermelho (A1), verde (A2) e azul (A2).

Em Matlab estão disponíveis as funções `svd` e `svds` que permitem obter os valores singulares de uma matriz. No entanto, para se poder aplicar estas funções à matriz que resulta de uma imagem, deve-se converter a matriz numa outra cujas entradas estejam em precisão dupla, usando o comando `double`. Por exemplo, considerando apenas a matriz A2,

```
A2=double(A2);
```

e depois faz-se

```
[U2,S2,V2] = svds(A2,50);
```

supondo que se pretende os primeiros 50 valores singulares da matriz A2. Para visualizar a imagem comprimida, faz-se

```
X2=U2*S2*V2';
Anew2 = uint8(X2);
imshow(Anew2);
```

Para recuperar a imagem original após a compressão, faz-se

```
Aend(:,:,1)=Anew1;
Aend(:,:,2)=Anew2;
Aend(:,:,3)=Anew3;
imshow(Aend)
```

A qualidade da imagem pode ser definida pela proporção

$$\sigma = \frac{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_p^2}{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_p^2 + \cdots + \sigma_r^2}$$

entre a soma dos quadrados dos valores singulares retidos e a soma dos quadrados de todos os valores singulares da matriz que representa a imagem. A percentagem de qualidade é $100\% \times \sigma$.

1. Considere uma imagem a preto e branco à sua escolha.

   (a) Escreva um script MATLAB que, tendo a imagem, a converta numa matriz A de tipo `double` e encontre a SVD para A usando o comando `svd`.

   Quantos valores singulares obteve?

   (b) Simplifique a imagem preservando apenas 25%, 50% e 75% dos valores singulares. [2 valores]

   Qual é a percentagem de qualidade de dados preservada? Imprima os resultados. [2 valores]

2. Considere agora uma imagem a cores à sua escolha. Simplifique a imagem preservando apenas 25%, 50% 75% dos valores singulares em cada camada RGB. Apresente os resultados. [4 valores]

III [6 valores]

Sendo $A \in \mathbb{R}^{n\times n}$ uma matriz não singular e simétrica, o seu número de condição, na norma Euclidiana, é

$$cond_2(A) = \|A\|_2 \|A^{-1}\|_2 = \varrho(A)\varrho(A^{-1}). \tag{1}$$

O objetivo deste problema é obter uma aproximação de $cond_2(A)$ com base na fórmula (1).

Para aproximar o maior valor próprio de A (em módulo), aplica-se o **método das potências**, cujo algoritmo se descreve a seguir:

(i) Escolher um vetor $x^{(0)} \in \mathbb{R}^n$, tal que $\|x^{(0)}\|_\infty = 1$.

(ii) Para $k = 1, 2, 3...$, calcular:

$$z^{(k)} = Ax^{(k-1)};\ \lambda^{(k)} = \frac{z_i^{(k)}}{x_i^{(k-1)}}\ (i \text{ qualquer índice tal que } x_i^{(k-1)} \neq 0);\ x^{(k)} = \frac{z^{(k)}}{\lambda^{(k)}}.$$

(iii) Continuar o processo até que $|\lambda^{(k)} - \lambda^{(k-1)}| < \varepsilon$ onde ε é uma tolerância dada. Em alternativa, pode-se usar o critério de paragem: $|\lambda^{(k)} - \lambda^{(k-1)}|/|\lambda^{(k)}| < \varepsilon$.

No caso de matrizes simétricas, sabe-se que o método das potências gera uma sucessão de números $\lambda^{(k)}$, a qual converge para λ tal que $|\lambda| = \varrho(A)$, isto é, λ é o valor próprio dominante (de maior módulo) da matriz dada. Além disso, a sucessão de vetores $x^{(k)}$ converge para um vetor próprio de A associado a λ.

1. Escreva um programa em MATLAB para calcular $cond_2(A)$ para uma matriz A simétrica, usando o **método das potências**. Os dados de entrada do programa devem ser a matriz A, o vetor $x^{(0)}$ e a tolerância ε. [3 valores]

2. **Aplicação: Condicionamento das matrizes de Hilbert.**

   Teste o programa da alínea anterior aplicando-o às matrizes de Hilbert de dimensão n, H_n, com $n = 5, 6, 7, ..., 20, ...$. Calcule ainda $\log(cond(H_n))$ e verifique que o crescimento do número de condição de H_n, quando n aumenta, é exponencial. Apresente os resultados num gráfico e comente os resultados das questões anteriores. [3 valores]

   Nota: Pode usar a função `hilb` do MATLAB que permite obter imediatamente a matriz de Hilbert.