



Trabalho em MatLab 02

Nota: Este trabalho pode contribuir (dependendo de discussão final) até um máximo de 1,33 valores para a sua nota final.

Resolva cada uma das partes do trabalho num ficheiro diferente, de nome `parte#.m`.

Coloque todos os ficheiros numa pasta com o nome do seu grupo de trabalho, `grupo#`.

Comprima (zip) a pasta e envie-a por email para o endereço `jda@isel.pt`

Parte A (4 valores)

Dada a matriz

$$\mathbf{A} = \begin{bmatrix} 5 & 0 & 2 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 3 & 5 & 1 \\ 2 & 1 & 2 & 1 \end{bmatrix}$$

1. Determine os valores próprios e os vectores próprios da matriz.
2. Verifique que a matriz é diagonalizável ($\mathbf{A} = \mathbf{VDV}^{-1}$).
3. Escreva o vector $\mathbf{u} = [1 \ 0 \ 2 \ 3 \ 5]^T$ como uma combinação linear dos vectores $\mathbf{u}_1 = [0 \ 1 \ 2 \ 1 \ 0]^T$, $\mathbf{u}_2 = [1 \ 0 \ 1 \ 0 \ 2]^T$, $\mathbf{u}_3 = [2 \ 1 \ 0 \ 2 \ 0]^T$, $\mathbf{u}_4 = [1 \ 0 \ 2 \ 0 \ 1]^T$, e $\mathbf{u}_5 = [0 \ 2 \ 1 \ 1 \ 0]^T$.

Dada a matriz

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 0 & 3 & 2 \\ 2 & 1 & 0 & 3 & 1 \\ 0 & 1 & 2 & 1 & 3 \\ 3 & 0 & 1 & 3 & 1 \\ 1 & 2 & 3 & 3 & 5 \end{bmatrix}$$

4. Determine uma base para o espaço linha da matriz.
5. Determine uma base para o espaço coluna da matriz.
6. Determine uma base para o núcleo da matriz.
7. Determine uma base para o espaço nulo à esquerda da matriz.
8. Sendo \mathbf{W} o subespaço de \mathbb{R}^5 gerado pelos vectores $\mathbf{u}_1 = [0 \ 1 \ 2 \ 1 \ 0]^T$ e $\mathbf{u}_2 = [1 \ 0 \ 1 \ 0 \ 2]^T$, determine uma base para o complemento ortogonal de \mathbf{W} .

9. Dados o conjunto de vectores linearmente independentes $\mathcal{U} = \{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$, com, $\mathbf{u}_1 = [0 \ 1 \ 2 \ 1 \ 0]^T$, $\mathbf{u}_2 = [1 \ 0 \ 1 \ 0 \ 2]^T$, e $\mathbf{u}_3 = [2 \ 1 \ 0 \ 2 \ 0]^T$ que constituem uma base de um subespaço \mathbb{W} de \mathbb{R}^5 , obtenha uma base ortonormada para esse subespaço.

10. Determine a matriz da transformação linear $T: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ que consiste numa rotação de $\theta = \pi/3$ sobre o versor $\mathbf{u} = [1/\sqrt{3} \ 1/\sqrt{3} \ 1/\sqrt{3}]^T$ seguida de uma reflexão sobre o plano \mathbf{xy} , e verifique se é uma isometria.

Dada matriz

$$\mathbf{A} = \begin{bmatrix} 0 & 2 & 0 & 3 & 2 \\ 2 & 1 & 0 & 3 & 1 \\ 2 & 1 & 2 & 1 & 3 \\ 3 & 0 & 1 & 1 & 1 \\ 5 & 2 & 3 & 3 & 1 \end{bmatrix}$$

11. Proceda à factorização $\mathbf{A} = \mathbf{QR}$.

12. Recorrendo à factorização QR, calcule iterativamente os valores próprios da matriz.

Dada a matriz

$$\mathbf{A} = \begin{bmatrix} 0 & 2 & 2 & 3 & 5 \\ 2 & 1 & 1 & 3 & 1 \\ 2 & 1 & 2 & 1 & 3 \\ 3 & 3 & 1 & 1 & 3 \\ 5 & 1 & 3 & 3 & 1 \end{bmatrix}$$

13. Determine a matriz ortogonal que diagonaliza \mathbf{A} .

14. Determine a decomposição espectral da matriz.

15. Sendo $\mathbf{u} = [1 \ 2 \ 3 \ 4 \ 5]^T$ determine a sua projecção sobre o subespaço próprio do maior valor próprio da matriz.

16. Determinar a matriz simétrica \mathbf{A} com valores próprios $\lambda_1 = 20$, $\lambda_2 = 2$, $\lambda_3 = 1$, e $\lambda_4 = 0$, e vectores próprios $\mathbf{v}_1 = [-2 \ 2 \ 0 \ 0]^T$, $\mathbf{v}_2 = [0 \ 0 \ 2 \ -2]^T$, $\mathbf{v}_3 = [2 \ 2 \ 2 \ 2]^T$, e $\mathbf{v}_4 = [2 \ 2 \ -2 \ -2]^T$.

Parte B (8 valores)

Reconhecimento de Caracteres

(O produto interno como medida de semelhança)

Objectivo: Escrever um *script* MatLab de que resulte a classificação do caracter contido no vector `digitogrupos` como um dos 26 caracteres do alfabeto contidos na matriz `alfabeto`.

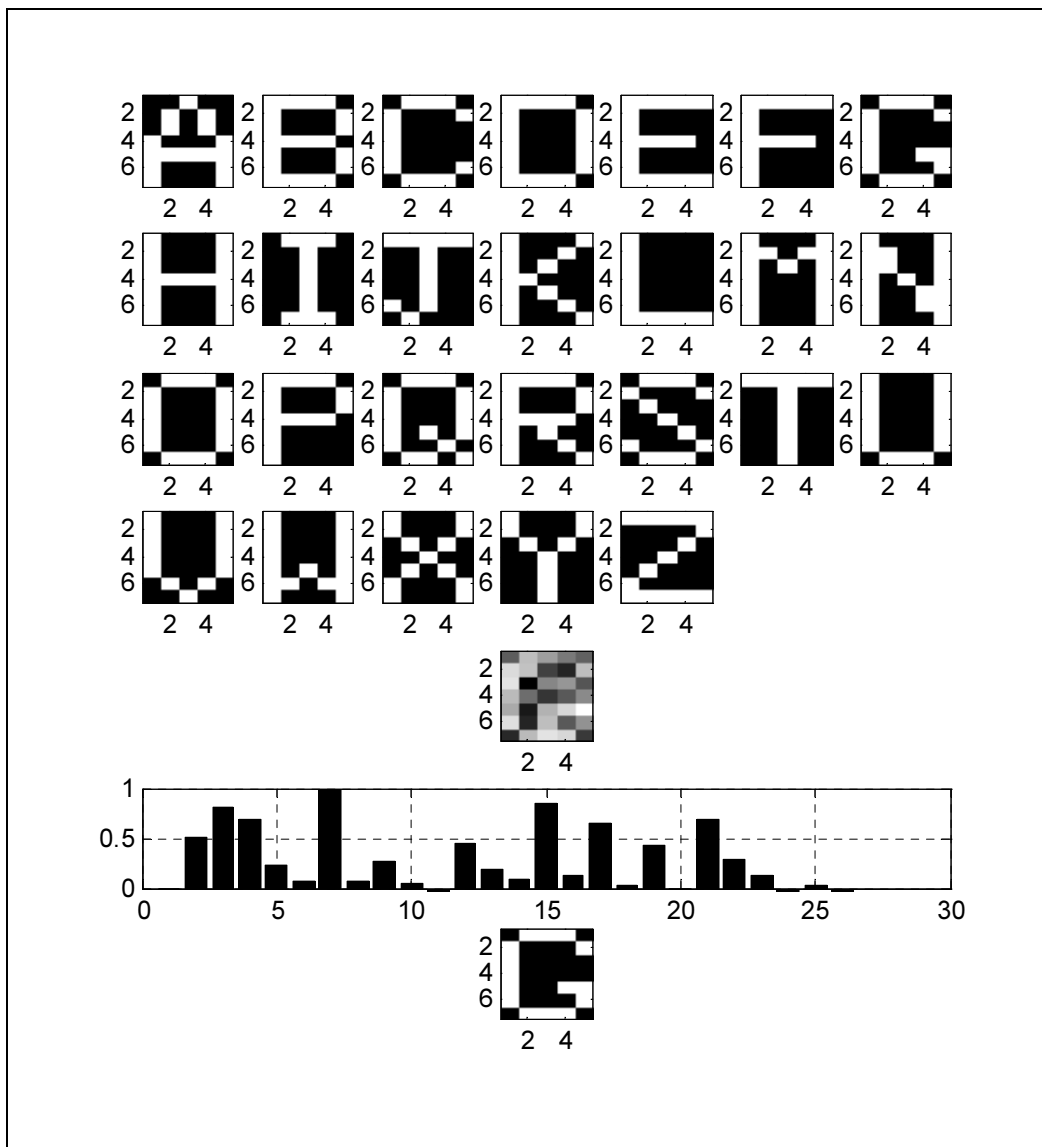


figura 1

1. Leia o ficheiro `alfabeto.mat`

```
» load alfabeto
```

Ficou no seu ambiente de trabalho com a variável `alfabeto`, que contém uma matriz 35×26 .

2. Veja a dimensão da matriz

```
» size(alfabeto)
```

Cada coluna da matriz corresponde, ordenadamente, a uma das 26 letras do alfabeto.

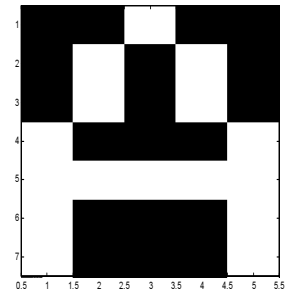
3. Selecciona a 1ª coluna da matriz e veja a sua dimensão

```
» a = alfabeto(:,1)
» size(a)
```

Antes de ser posta na forma de um vector coluna, cada uma das letras correspondia a uma matriz 7×5 em que cada um dos coeficientes da matriz correspondia a um 0 ou um 1, de acordo com a forma da letra.

4. Veja a matriz original e a correspondente imagem em 256 níveis de cinzento (0, preto; 1, branco).

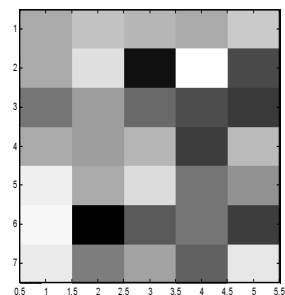
```
» A = (reshape(a,5,7))'
» figure(1);
» colormap('gray');
» imagesc(A);
```



5. No seu ambiente de trabalho tem uma variável de nome `digitogrupo##`, em que `##` corresponde ao seu número de grupo. Esta variável, uma matriz coluna 35×1 , correspondente a uma letra do alfabeto que foi corrompida com ruído.

O objectivo do trabalho é descobrir a que letra correspondia originalmente esta matriz, ou seja classificar esta letra como uma das 26 letras do alfabeto.

6. Vai para isso comparar o vector corrompido com os vectores originais, utilizando o produto interno como medida de semelhança.

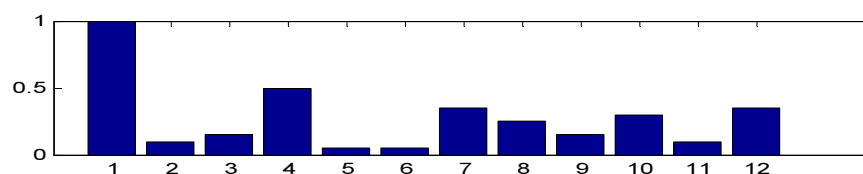


Calculando a projecção ortogonal do vector corrompido sobre cada um dos vectores originais obtemos o vector dos coeficientes de representação do dígito corrompido no espaço, \mathbb{R}^{26} , gerado pelos dígitos originais.

Para ter uma maior sensibilidade à relação de semelhança entre os dígitos, podemos fazer um gráfico de barras do valor absoluto do vector das projecções ortogonais normalizado pelo seu valor máximo.

Num exemplo em \mathbb{R}^{12} teríamos

```
» po=[ 20 2 -3 10 1 1 -7 5 3 6 2 7];
» po=abs(po/max(po));
» figure(1);bar(po)
```



Vamos classificar o dígito corrompido como aquele a que corresponde a maior (em módulo) projecção ortogonal.

7. Resumindo, o seu *script* deve seguir os seguintes passos

1. Ler o ficheiro `alfabeto.mat`
2. Representar cada um dos dígitos originais num espaço gráfico 7×7 .
3. Representar o dígito corrompido (`digitogrupo##`) num espaço gráfico 7×7 .
4. Calcular a projecção ortogonal do dígito corrompido sobre cada um dos dígitos originais.

5. Representar o vector normalizado das projecções ortogonais num espaço gráfico 7×1 .
6. Representar o dígito que esteve na origem do dígito corrompido num espaço gráfico 7×7 .

A figura 1 exemplifica um resultado típico, e o *output* desejado da execução do código do seu script.

A figura 2 mostra as matrizes correspondentes aos [digitogrup01](#) a [digitogrup014](#).

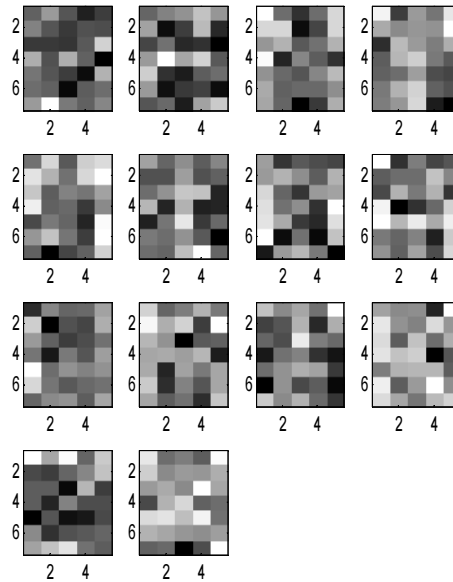


figura 2

Parte C (8 valores)

Animação 3D

(Transformações lineares em R3)

Objectivo: Escrever um *script* MatLab de que resulte uma animação idêntica à de *rotcube.avi*, que consiste numa sequência de rotações de um cubo sobre um eixo cuja direcção muda aleatoriamente.

O que deve saber:

1. Como representar um cubo em R3.
2. Como gerar um versor com direcção aleatória.
3. Como gerar, em R3, uma matriz de rotação sobre um eixo arbitrário.
4. Como simular movimento em resultado da aplicação de uma sequência de transformações lineares.

Leia com atenção as páginas 27 a 35 do módulo 6. Tenha em atenção o ficheiro `randomvector.m` em anexo.