

# Trabalho em MatLab 03

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 (5 valores)

---

Dado o vector  $\mathbf{v} = [0 \ 2 \ 2 \ 1]^T$  e a matriz

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 3 \\ 1 & 4 & 2 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

1. Decomponha  $\mathbf{A}$  em valores singulares.
2. Determine a solução de mínimo erro quadrático do sistema  $\mathbf{Ax} = \mathbf{v}$ .
3. Determine a matriz de projecção ortogonal no espaço coluna de  $\mathbf{A}$ .
4. Determine a matriz pseudo-inversa da matriz  $\mathbf{A}$ .
5. Dados os pontos de  $\mathbb{R}^2$ ,  $(-2, -2)$ ,  $(-1, 0)$ ,  $(0, 1)$ ,  $(1, 2)$ ,  $(2, 3)$ , e  $(3, 3)$ , determine os parâmetros  $a_0$ ,  $a_1$ ,  $a_2$ , e  $a_3$ , do polinómio  $y = a_0 + a_1x + a_2x^2 + a_3x^3$  que melhor se adapta ao conjunto dos pontos, segundo um critério de minimização do erro quadrático cometido. Represente graficamente o conjunto de pontos e o polinómio encontrado.

**Nota:** Pode calcular um integral em MatLab recorrendo à função `int()` da biblioteca simbólica.

Por exemplo, para calcular o integral definido

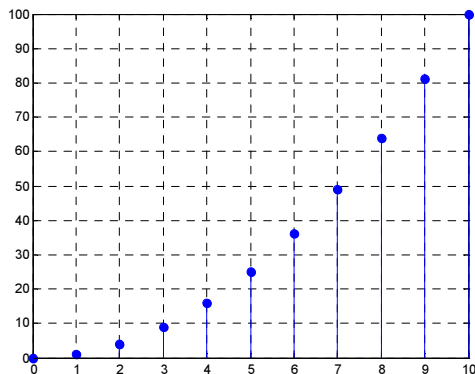
$$\int_{-\pi}^{\pi} x^2 \cos(x) dx$$

, pode fazer

```
>> syms x
>> int(x^2*cos(x), x, -pi, pi)
ans =
-4*pi
```

Recorra ao help do MatLab para mais esclarecimentos.

6. Considerando o espaço vectorial  $\mathcal{C}[-\pi, \pi]$  com produto interno  $f \cdot g = \int_{-\pi}^{\pi} f(t)g^*(t)dt$ , verifique que  $x_1(t) = \cos(t)$ ,  $x_2(t) = \cos(3t)$  e  $x_3(t) = \sin(2t)$  são ortogonais.



**Nota:** Para representar um sinal discreto em MatLab pode recorrer à função `stem()`.

Por exemplo, para representar o sinal

$$x_1[n] = n^2 \sum_{i=0}^{10} \delta[n-i] \text{ pode fazer}$$

```
n=0:10;
x=n.^2;
stem(n,x,'filled')
grid on
```

7. Represente graficamente, para  $n \in [0, 19]$ , os sinais discretos  $x_1[n] = \sin\left[n \frac{\pi}{10}\right] \sum_{i=0}^{19} \delta[n-i]$  e

$x_2[n] = \cos\left[n \frac{\pi}{10}\right] \sum_{i=0}^{19} \delta[n-i]$ . Verifique se os sinais são ortogonais em  $l^2$  com produto interno

$$x_1 \cdot x_2 = \sum_{n=-\infty}^{\infty} x_1[n]x_2^*[n].$$

8. Calcule a projecção ortogonal de  $f(x) = \sin(\pi x)$  no subespaço de  $\mathcal{C}[0, 1]$  gerado por  $g(x) = x - x^2$ , com produto interno  $f \cdot g = \int_0^1 f(x)g^*(x)dx$ .

**Nota:** Em MatLab, pode determinar o conjugado de um vector com a função `conj()`.

9. Recorrendo à ortogonalização de Gram-Schmidt determine uma base ortogonal,  $\mathcal{B} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ ,

para o subespaço  $\mathbb{W}$  de  $\mathcal{C}[0, \pi]$  com produto interno  $f \cdot g = \int_0^{\pi} f(x)g^*(x)dx$  gerado pelos vectores

$$\mathbf{u}_1 = 1, \mathbf{u}_2 = 1 - e^{-j2x}, \text{ e } \mathbf{u}_3 = 1 + e^{j2x}.$$

10. Considerando os vectores  $\mathbf{v}_1$ ,  $\mathbf{v}_2$  e  $\mathbf{v}_3$  calculados na alínea anterior, calcule os parâmetros da função  $\hat{f}(x) = a\mathbf{v}_1 + b\mathbf{v}_2 + c\mathbf{v}_3$ , que, no intervalo  $[0, \pi]$ , melhor se adapta à função  $f(x) = \cos(2x)$  no sentido da minimização do erro quadrático, ou seja, calcule a projecção ortogonal de  $f(x)$  no subespaço vectorial  $\mathbb{W}$  com produto interno  $f \cdot g = \int_0^{\pi} f(x)g^*(x)dx$ . Caso não tenha resolvido a alínea anterior considere  $\mathbf{v}_1 = 1$ ,  $\mathbf{v}_2 = e^{-j2x}$ , e  $\mathbf{v}_3 = e^{j2x}$ .

## Parte C (15 valores)

### “BoundingBox”

(Aplicações da decomposição em valores singulares)

**Objectivo:** Escrever um *script* MatLab que determina a *boundingbox* de um conjunto de pontos em  $\mathbb{R}^2$ .

Leia com atenção as páginas 2 a 6 do módulo 8.

O que deve saber:

Como fazer a decomposição de uma matriz em valores singulares,  $U\Sigma V^T$ .

Interpretar geometricamente as transformações lineares associadas às matrizes  $U$ ,  $\Sigma$ , e  $V$ .

Em MatLab, pode determinar o valor máximo e mínimo de um vector com as funções `max()` e `min()`.

Em MatLab, pode determinar o valor médio de um conjunto de valores de um vector com a função `mean()`.

1. Leia o ficheiro `pontos.mat`

» `load pontos`

Ficou no seu ambiente de trabalho com a variável `A`, correspondente a uma matriz  $2 \times 500$  com informação das coordenadas de 500 pontos em  $\mathbb{R}^2$

$$A = \begin{bmatrix} x_1 & x_2 & \cdots & x_{500} \\ y_1 & y_2 & \cdots & y_{500} \end{bmatrix}$$

2. Represente graficamente os pontos, de modo semelhante ao que se mostra na figura 1.

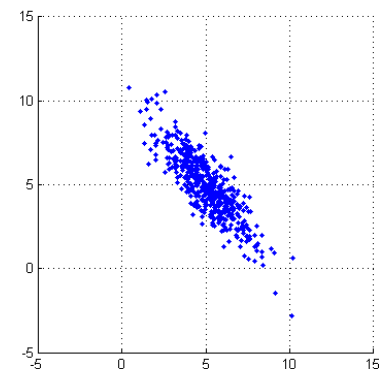


Figura 1

3. Recorrendo às funções `max()` e `min()` determine os valores máximo e mínimos das coordenadas dos pontos segundo  $x$  e  $y$ , e, com base nessa informação, represente graficamente o rectângulo alinhado com os eixos coordenados que melhor se adapta ao conjunto de pontos, de modo semelhante ao que se mostra na figura 2.

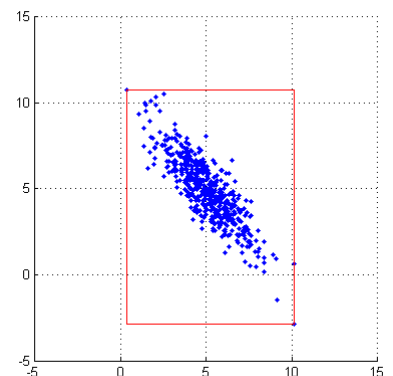


Figura 2

Suponha que existiam no plano  $\mathbb{R}^2$  diversas “manchas” de pontos idênticas às que representou, e que estas se movimentavam livremente sobre o plano, sem se deformarem. Admita ainda que se pretendia fazer a representação do movimento das manchas sobre o plano, e que se pretendia que estas não se inter-cruzassem, devendo modificar a sua

trajectória sempre que se aproximassem umas das outras. Tal poderia ser feito de modo eficaz recorrendo à informação dada pelos rectângulos que conteriam cada uma das manchas. No entanto, o rectângulo alinhado com os eixos coordenados ajusta-se mal à mancha, de onde resultaria que as manchas se repeliriam quando estivessem a distância bastante significativas umas das outras.

O objectivo deste trabalho é, com base em informação retirada da decomposição em valores singulares de uma matriz, encontrar o rectângulo que melhor se adapta a um conjunto de pontos, no sentido em que contém todos os pontos e “minimiza” o espaço livre entre os pontos e os lados do rectângulo.

4. Recorrendo à função `mean()` determine o valor médio das coordenadas segundo  $x$  e segundo  $y$  (a partir da informação contida na matriz **A**, e crie uma nova matriz, **B**, com o conjunto de pontos correspondentes à matriz original, mas centrados na origem dos eixos coordenados, ou seja, proceda a uma transformação afim correspondente a uma **translação** dos pontos originais. Represente o novo conjunto de pontos, de modo semelhante ao que se mostra na figura 3.

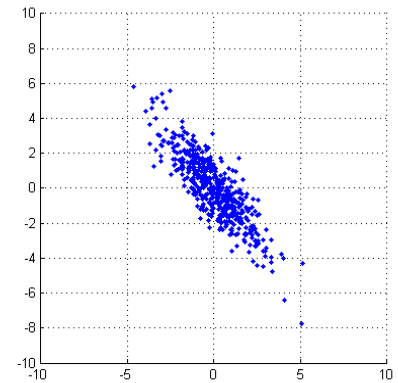


Figura 3

5. Calcule os valores singulares da matriz **B**. Com base na informação da(s) matriz(es) resultante(s) da decomposição, proceda a uma **rotação** dos pontos da matriz **B**, de modo a que os vectores próprios associados aos valores singulares fiquem alinhados com os eixos coordenados. Represente o novo conjunto de pontos, de modo semelhante ao que se mostra na figura 4.

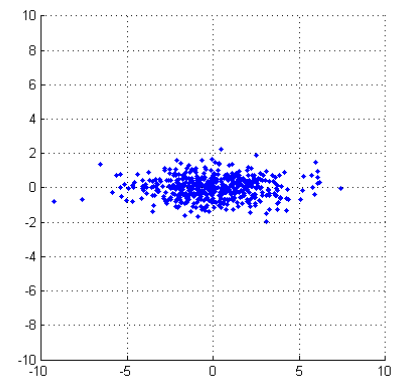


Figura 4

6. De modo idêntico ao ponto 3, determine os valores máximos e mínimos das coordenadas dos pontos segundo  $x$  e  $y$ , e, com base nessa informação, represente graficamente o rectângulo alinhado com os eixos coordenados que melhor se adapta ao conjunto de pontos, de modo semelhante ao que se mostra na figura 5.

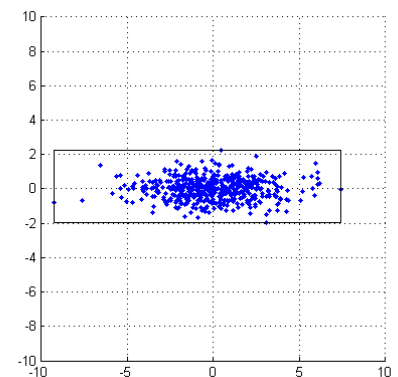


Figura 5

7. Recorrendo de novo à informação sobre a decomposição em valores singulares da matriz **B**, proceda à rotação do rectângulo obtido em 6, agora em sentido inverso (note que toda a matriz de rotação é ortogonal), e à sua translação, e represente-o graficamente, juntamente com os pontos originais (matriz **A**), de modo semelhante ao que se mostra na figura 6.

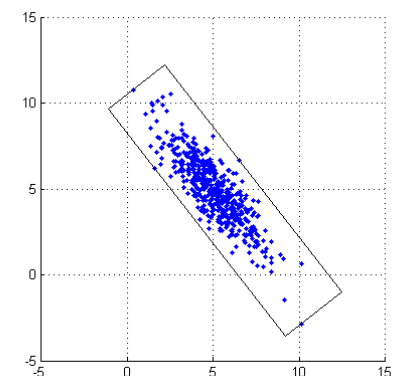


Figura 6