APONTAMENTOS MATLAB PARA O TESTE

vetor linha com os números naturais menores ou iguais a 10 : n = 1:10
se quiser com um espaçamento 0,5 : m = 1:0.5:10

Matriz identidade: eye(...) (matriz quadrada) ou eye(...,...)
Matriz números aleatórios (entre 0 e 1): rand(...) ou rand(...,...)
Matriz zeros: zeros(...,...)
Matriz de uns: ones(...) ou ones(...)
Se queremos a diagonal de uma matriz num vetor coluna: diag(matriz)

A*B - Faz a multiplicação entre 2 matrizes (pela álgebra linear)
A.*B - Faz a multiplicação das matrizes elemento a elemento

---

m_file: ficheiros do MATLAB - podemos criar funções

Numa função MATLAB:

function [variáveis a calcular] = nome_da_função(variáveis dadas)

Exemplo função que calcula a soma e o produto dos elementos de um vetor:

```
function [soma, produto] = somaprod(x)
soma = sum(x);
produto = prod(x);
end
```

Na command window:

```
[s,p] = somaprod[1 5 7 9 -1]
```

Resultado:
s = 24
p = -945

Funções que dão o valor máximo e mínimo de um vetor/matriz:

max(x) - valor máximo
min(x) - valor mínimo

Como fazer a EGPP (Eliminação de Gauss com Pivotagem Parcial) no MATLAB:

Seja A = [1 2 3 ; 4 5 6; 7 8 9]
Seja b = [10 11 12]

A\b'- Resolve o sistema de equações lineares
det(A) - Calcula o determinante da matriz A

format long - aumenta a visualização dos números para 15 casas decimais
format short - diminui a visualização dos números para 4 casas decimais

inv(A) - calcula a matriz inversa de A

Capítulo 2 - Método de Gauss-Seidel

Objetivo: Resolver sistemas de equações lineares

_ _ _ _ _ _ _ _ _ _

Exemplo Exercício:

x1 + 0.5 x2 + 0.5 x3 = 2
0.5 x1 + x2 + 0.5 x3 = 2
0.5 x1 + 0.5 x2 + x3 = 2

Daqui resulta:

A = [1 0.5 0.5; 0.5 1 0.5; 0.5 0.5 1]
b = [2 2 2]'

Estude a convergência através das condições suficientes.

Condições suficientes de convergência no método de Gauss-Seidel:

1º condição: Matriz A é estrita e diagonavelmente dominante - Apenas pode ser verificado analiticamente

2º condição: A é simétrica e definida positiva

Para verificar se é simétrica, verificar se A = inv(A);

Para verificar se é definida positiva, verificar as seguintes condições:

det(A(1,1))
det(A(1:2,1:2))
det(A)

Se todos estes valores derem positivos, é definida positiva

3º condição: Norma 1 e Norma Infinita da matriz iteração do Método de Gauss Seidel tem que ser inferior a 1.

U (Upper)= matriz diagonal inferior de A (sem diagonal)
L (Lower) = matriz diagonal superior de A (sem diagonal)
D (Diagonal) = matriz diagonal de A
C_GS - Matriz iteração do Método de Gauss Seidel

Etapas:

trim(A); - matriz diagonal superior (c/ diagonal)
trim(A,1); - matriz diagonal superior (s/ diagonal)
U = - trim(A,1);
D = diag(diag(A));
tril(A,-1) - matriz diagonal inferior (s/ diagonal)
L = - tril(A,-1);
C_GS = inv(D-L)*U

Tendo a matriz iteração, verificar o valor das normas

norm(C_GS,1) - norma 1 (a maior soma dos módulos dos elementos de uma coluna da matriz)
norm(C_GS,inf) - norma infinita (a maior soma dos módulos dos elementos de uma linha da matriz)

Se nenhuma das 3 condições não for verificada, nada se pode concluir.

Capítulo 3 - Equações não lineares

Objetivo: Calcular soluções de equações não lineraes

Funções MATLAB utilizadas:

op = optimset('tolx', E1, 'tolfun', E2, 'display', 'iter'); - Esta função permite alterar o erro que pretendemos (para E1 e E2). O comando 'display', 'iter' permite mostrar-nos as iterações feitas.

[x,f,exitflag,output] = fsolve('nome da função','valor inicial onde se inicia a iteração', op); - (calcula zeros pelo método de Newton)

'valor inicial onde se inicia a iteração' = número (1 equação) ou vetor linha (várias equações).

exitflag: diz se o processo decorreu de forma correta: 1 (correto) ou 0 (incorreto)

_ _ _ _ _ _ _ _ _ _ _ _ _

Exemplo: Exercício 3.4) (ver atrás, já resolvido numericamente)

1º - criar uma função:

```
function[f] = distancia(x)
f = 2552-30*x^2 + x^3
end
```

Na command window

```
op = optimset('tolx', 0.001, 'tolfun', 0.001, 'display', 'iter');
[x,f,exitflag,output] = fsolve('distancia', 10, op);
```

Capítulo 4 - Polinómio interpolador de Newton

Objetivo: Encontrar uma aproximação, neste caso concreto através de um polinómio p(x) de uma função f(x) com o menor erro possível

Funções MATLAB utilizadas:

polyfit(vetor x, vetor f, grau do polinómio) - ajuste polinomial aos pontos, polinómio dado na forma canónica, (grau3, grau2, grau1, grau0)

polyval(polinómio obtido, x para o qual queremos descobrir f(x))

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Suponha-se que nos dão os seguintes valores

```
x = [0 1 2.5 3 7 8 12]
f = [-5 4 3 1 4 6 -12]
```

Suponhamos que queremos descobrir o valor de f(4) através de um polinómio de grau 3 (necessários 4 pontos, os mais próximos de 4)

Neste caso os valores mais próximos serão [1 2.5 3 7] - ou x(2:5)

Assim

```
p3 = polyfit(x(2:5),f(2:5),3)
f4 = polyval(p3,4)
```

Capítulo 5 - Splines

Considerando o exercício M5.1)

x = [5.0 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8 5.9 6.0]
f= [0.0639 0.0800 0.0988 0.1203 0.1442 0.1714 0.2010 0.2331 0.2673 0.3036 0.3414]

1) Qual o valor aproximado da função no ponto x = 5.45, bem como o segmento da spline correspondente desse ponto? - SPLINE NATURAL

s3 = spline(x,f) - cria a spline
s3.coefs - dá nos os coeficientes dos vários segmentos numa matriz. Cada linha corresponde ao respetivo segmento. O ponto x=5.45 está compreendido entre x=5.4 e x=5.5 - corresponde ao 5º segmento

Logo s3.coefs(5,:) = [-0.3521   0.2004   0.2555   0.1442] - isso significa que s3.5 = -0.3521(x-5.4)^3 + 0.2004(x-5.4)^2 + 0.2555(x-5.4) + 0.1442
(x(5) = 5.4, devido a ser o 5 segmento)

Valor aproximado em x=5.45 - spline(x,f,5.45) - neste caso será igual a 0.1574

2) Qual o valor aproximado da função no ponto x = 5.45, bem como o segmento da spline correspondente desse ponto? - SPLINE COMPLETA

s3 = spline(x,[dd0 f ddn]) - dd0 e ddn são as derivadas no ponto inicial e no ponto final, respetivamente

---

Capítulo 6 - Integração Numérica

x = [0.0 0.5 1.0 1.5 2.0 2.5 3.0 4.0 5.0]
f = [-4271 -2522 -499 1795 4358 7187 10279 13633 17247]

Se quisermos calcular o integral da função f entre 0 e 5 pela fórmula do trapézio: UTILIZAR APENAS QUANDO TEMOS UM CONJUNTO DE PONTOS DA FUNÇÃO

trapz(x,f) - neste caso = 34058

Se quisermos calcular o integral da função f dada por uma expressão:

Exemplo:

fun = @(x) 4./(1+x.^2)

integral(fun,0,1) - 0 e 1 são os limites de integração - (se quisermos até infinito - inf).

Capítulo 7 - Equações Diferenciais

ode45 - Função do MATLAB que resolve equações diferenciais. Mas só resolve com condições iniciais (não resolve com condições de fronteira)

```
[x,y] = ode45('nome da equação diferencial', valores em estudo, condição inicial y(0))
```

Nota: se colocar 2 valores em "valores em estudo", vai aparecer todos os valores compreendidos entre esses 2 que o programa faz
Se colocar 3 valores, já só aparecem os 3 que pedi

Exemplo da resolução de um exercício:

```
function[dy] = eqdif(x,~)
dy = 7000*(20-x)/(100-2.5*x);
end
```

```
[x,y] = ode45('eqdif', [0 0.5], 0)
```

Caso queiramos apenas os valores pretendidos - adicionamos mais um

```
[x,y] = ode45('eqdif', [0 0.5], 0)
```

Capítulo 8 - Mínimos Quadrados

```
[p,r] = polyfit(x,y,n)
```

em que p é a matriz que dá os coeficientes do polinómio (em potências descendentes).
r dá uma estrutura que contém no seu último termo a norma do resíduo.
x e y são os valores dados através de uma tabela.
n é o grau do polinómio.

para calcular o valor da função num determinado ponto utiliza-se a função polyval.