TPC#1
Data de entrega: 9 de Março de 2023
Autores: António Santos (100%); Diogo Reis (100%); Pedro Coelho (100%)

1. (a)

Ora, pretendemos provar que o método de Newton converge linearmente para x_*, ou seja, que a sucessão x_k converge linearmente para x_* se existir r $\in$ (0,1) tal que:

$$\frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} \leq r, \ \forall k > k_0$$

.

Sabemos que o método de Newton nos é dado por: $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$.

Utilizando a expansão do Método de Taylor para um ponto na vizinhaça de x_* vem:

$$f(x_n) = f(x_*) + f'(x_*)(x - x_*) + \frac{1}{2}f''(\eta)(x - x_*)^2$$

, tal que $\eta = x_* + \theta x_n, \theta \in (0, 1)$.

Utilizando o facto de que $f(x_*) = f'(x_*) = 0$ vem que:

$$f(x_n) = \frac{1}{2}f''(\eta)(x - x_*)^2$$

, tal que $\eta = x_* + \theta x_n, \theta \in (0, 1)$.

Aplicando então o método de Newton no ponto x_*:

$$x_{n+1} - x_* = x_n - x_* - \frac{f''(\eta)(x_n - x_*)^2}{2f'(x_n)} \Leftrightarrow$$

$$\Leftrightarrow \ x_{n+1} - x_* = (x_n - x_*)(1 - \frac{f''(\eta)(x_n - x_*)}{2f'(x_n)})$$

Como $f''(x_*) \neq 0$, existe r tal que $|1 - \frac{f''(\eta)(x_n - x_*)}{2f'(x_n)}| \leq r$.

Assim,

$$|x_{n+1} - x_*| = |x_n - x_*||1 - \frac{f''(\eta)(x_n - x_*)}{2f'(x_n)}|$$

$$\Leftrightarrow |x_{n+1} - x_*| \leq |x_n - x_*|r$$

Como tal, prova-se que o método de Newton converge linearmente para x_*.

(b)

O método do ponto fixo é dado por: $x_{n+1} = g(x_n)$

Podemos implementar o método do ponto fixo a partir do método de Newton da seguinte forma: Seja $g(x_n) = x_n - \frac{f(x_n)}{f'(x_n)}$ e $g(x_*) = x_*$.

Desenvolvendo a série de Taylor de $g(x_n)$ em torno de x_* vem:

$$g(x_k) = g(x_*) + g'(x_*)(x_n - x_*) + \frac{1}{2}g''(x_*)(x_n - x_*) + \frac{1}{6}g'''(\eta)(x_n - x_*)^3$$

, tal que $\eta = x_* + \theta x_n, \theta \in (0, 1)$

Ora,

$$g'(x_*) = 1 - \frac{(f'(x_*))^2 - (f''(x_*))f(x_*)}{(f'(x_*)^2)}$$

$$g''(x_*) = \frac{-2f(x_*)(f''(x_*))^2 - f(x_*)f'(x_*)f'''(x_n*) - (f'(x_*))^2 f''(x_*)}{(f'(x_*))^3}$$

$$g'''(x_*) = \frac{-(f'(x_*))^3 f'''(x_*) - (f'(x_*))^3(f'''(x_*)) + 3(f'(x_*))^2(f''(x_*))^2}{(f'(x_n))^4}$$
$$\frac{-f(x_*)(f'(x_*))^2 f'''(x_*) + 6f(x_*)f'(x_*)f''(x_*)f'''(x_*) - 6f(x_*)(f''(x_*))^3}{(f'(x_n))^4}$$

Utilizando o facto de que $f(x_*) = f''(x_*) = 0$ podemos simplificar as derivadas $g(x_*)$.

Vem então que:

$g'(x_*) = g''(x_*) = 0$ e $g'''(x_*) \neq 0$

Como tal, e, aplicando as propriedades das normas, vem que:

$$x_{n+1} = g(x_*) + \frac{1}{6}g'''(\eta)(x_n - x_*)^3 \Leftrightarrow$$

, tal que $\eta = x_* + \theta x_n, \theta \in (0, 1)$

$$\Leftrightarrow x_{n+1} - x_* = \frac{1}{6}g'''(\eta)(x_k - x_*)^3 \Leftrightarrow$$

, tal que $\eta = x_* + \theta x_n, \theta \in (0, 1)$

$$\|x_{k+1} - x_*\| \leq \|x_k - x_*\|^3 \|\frac{1}{6}g'''(\eta)\|$$

Portanto, atendendo ao facto de que $\|\frac{1}{6}g'''(\eta)\| > 0$ mostramos que quando $f(x_*) = 0$ $f''(x_*) = 0$ e $f'(x_*) \neq 0$ e $f'''(x_*) \neq 0$ o método de Newton converge cubicamente para x_*.

2. Vamos, inicialmente, obter uma relação entre os erros absolutos entre $\|x_1 - x_*\|$ e $\|x_0 - x_*\|$ com $F(x_*) = 0$, assumindo $J(x_0)$ invertível (o que provamos de seguida) e $\|e_0\| \leq c\|F(x_0)\|^2$. .

   Pela fórmula recursiva do método de Newton vem:

$$x_1 - x_* = x_0 - x_* - J(x_0)^{-1}(F(x_0) + e_0) \Leftrightarrow$$
$$\Leftrightarrow x_1 - x_* = J(x_0)^{-1}J(x_0)(x_0 - x_*) - J(x_0)^{-1}(F(x_0) + e_0 - F(x_*)) \Leftrightarrow$$
$$\Leftrightarrow x_1 - x_* = J(x_0)^{-1}[F(x_*) - F(x_0) + e_0 + J(x_0)(x_0 - x_*)] \Leftrightarrow$$

, $F(x_*) = 0$

$$\Leftrightarrow \|x_1 - x_*\| \leq \|J(x_0)^{-1}\|[\|e_0\| + \| - F(x_0) + J(x_0)(x_0 - x_*)\|] \Leftrightarrow$$

, $\|e_0\| \leq c\|F(x_0)\|^2$

$$\Leftrightarrow \|x_1 - x_*\| \leq \|J(x_0)^{-1}\|[c\|F(x_0)\|^2 + \| - F(x_0) + J(x_0)(x_0 - x_*)\|] \Leftrightarrow$$

,

$$\Leftrightarrow \|x_1 - x_*\| \leq \|J(x_0)^{-1}\|[c\|F(x_0) - F(x_*)\|^2 + \frac{\gamma}{2}\|(x_0 - x_*)\|^2] \Leftrightarrow$$

, $c\|F(x_0) - F(x_*)\|^2 \leq cl^2\|x_0 - x_*\|^2$ pois é continua à Lipschitz.

$$\Leftrightarrow \|x_1 - x_*\| \leq \|J(x_0)^{-1}\|(cl^2 + \frac{\gamma}{2})\|(x_0 - x_*)\|^2$$

Vamos agora analisar o caso geral:

Comecemos por colocar uma bola centrada em x_* dentro de D, o que é possível pois D é aberto. Seja $r > 0$ o raio da bola. Vamos considerar $\epsilon \leq r$ tal que $\|x_0 - x_*\| \leq \epsilon$, ou seja: x_0 está na bola, e, consequentemente em D.

Provámos inicialmente que se $J(x_0)$ for invertível:

$$\|x_1 - x_*\| \leq \|J(x_0)^{-1}\|(cl^2 + \frac{\gamma}{2})\|(x_0 - x_*)\|^2$$

.

Temos agora de garantir que $J(x_0)$ é invertível, uma vez que pelas condições do teorema 1 $J(x_*)$ é não singular e $\|(J(x_*))^{-1}\| \leq \beta$.

Seja $J(x_0)$ tal que $\|[J(x_*)]^{-1}(J(x_*) - J(x_0))\| \leq 1$

Então $J(x_0)$ invertível e:

$$\|[J(x_0)]^{-1}\| \leq \frac{\|[J(x_*)]^{-1}\|}{1 - \|[J(x_*)]^{-1}(J(x_*) - J(x_0))\|}$$

Ora,

$$\|[J(x_*)]^{-1}(J(x_*) - J(x_0))\| \leq \|J(x_*)^{-1}\|\|(J(x_*) - J(x_0))\| \leq \beta\gamma\|x_* - x_0\| \leq \frac{1}{2} < 1$$

,

$\|x_* - x_0\| = \epsilon$ e $\epsilon \leq \frac{1}{2\beta\gamma}$

Assim,

$$\frac{\|[J(x_*)]^{-1}\|}{1 - \|[J(x_*)]^{-1}(J(x_*) - J(x_0))\|} \leq \frac{\beta}{1 - \frac{1}{2}} = 2\beta$$

Raciocionando indutivamente estabelece-se que para todo o k tal que $J(x_k)$ é não singular que

$$\|x_{k+1} - x_*\| \leq 2\beta(cl^2 + \frac{\gamma}{2})\|x_k - x_*\|^2$$

.

o que, tomando $M = 2\beta(cl^2 + \frac{\gamma}{2})$ equivale a:

$$\|x_{k+1} - x_*\|M \leq \|x_k - x_*\|^2$$

.

e portanto, pelas condições do Teorema 1, mostra-se que a sucessão x_k converge quadráticamente para x_*.

3. O método de Newton é uma das principais formas para calcular valores e vectores próprios de uma matriz através de métodos iterativos. Para tal, é necessária uma aproximação inicial do vector próprio correspondente ao maior valor próprio da matriz.

   Para esta aproximação deve ser escolhido um vector unitário, pois assim garantimos que em cada iteração o vector resultante também será unitário, além disso, o vector deve ser arbitrário normalizado fazendo com que o método comece a convergir a partir de qualquer ponto no espaço vectorial em que a matriz actua.

   Quando se aplica o método de Newton a uma matriz A com um vector próprio unitário, o vector que obtemos é uma aproximação do maior vector próprio da mesma.

   Posto isto, devemos ter em atenção a escolha do vector inicial a fim de obtermos uma convergência rápida e estável.

   Para a o programa pretendido começámos por definir uma matriz com quadrada com 4 elementos inteiros a variar entre 0 e 10, uma matriz identidade de igual ordem e um vector com o mesmo número de elementos. Normalizámos o vector anterior e calculámos uma aproximação para o valor próprio, tal como sugerido no enunciado.

```
clc; close all; clear;

n       = 4;                 % Ordem da matriz
A       = randi([0 10],n);   % Matriz arbitrária de ordem n com elemntos a variar entre 0 e 10
I       = eye(n);            % Matriz identidade de ordem n
x       = rand(n, 1);        % Vector arbitrário
x       = x / norm(x);       % Normalizaçao do vector arbitrário anterior, sendo norm(x) a norma do vector
y       = x'*A*x;            % Valor próprio de A
tol     = 1e-10;             % Tolerância
kmax    = 100;               % #máximo de iteracões
k       = 0;                 % Iteração inicial
erro    = tol + 1;           % Iniciar o ciclo
aux     = 1;                 % Variável auxiliar para o cálculo do vetor próprio
```

   Seguidamente implementámos um ciclo *while* com o método de Newton. O ciclo começa por resolver o sistema linear que nos é dado no enunciado $J_F(x_k, \lambda_k)p_k = -F(x_k, \lambda_k)$, a variável b representa a matriz $\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix}$ apresentada no enunciado.

```
fprintf(' \n  Matriz A: \n');
    disp(A);

disp('---------------------------------');
disp(' Iteração        Valor prório   ');
disp('---------------------------------');
fprintf('    %2.0f   %20.16f \n',k,y);

%Ciclo de Newton
while (erro > tol) && (k < kmax)
    F       = Funcao(A,x,y);
    J       = Jacobiana(A,x,y,I);
    pk      = - J \ F;
    v       = [x ; y];             % Vector normalizado anteriormente com y na ultima linha
    b       = v + pk;              % Soma do vector v com o vector pk, ficando b com n+1 elems
    x       = b(1:n);              % x toma os valores do n primeiro elementos de b
    y       = b(n+1);              % y toma o último valor de b
    erro    = norm(F);
    k       = k + 1;
    fprintf('    %2.0f   %20.16f \n',k,y);  % print do valor próprio na iteração k

end
```

   Fizemos um novo ciclo para a impressão do vector próprio. No final estão definidas as funções dadas no enunciado $F(x, \lambda)$ e $J_F(x, \lambda)$.

   Seguidamente ficam dois testes feitos com matrizes escolhidas aleatoriamente.

```matlab
    fprintf(' \n  Vector próprio: \n');

        % Impressão do vector próprio
while (aux<n+1)
    fprintf('    %20.16f \n',x(aux));
    aux=aux+1;
end

    fprintf(' \n  Matriz A: \n');
    disp(A);


function F = Funcao(A, x, y) % Funcao F conforme o enunciado
F = [A*x - y*x; x'*x - 1];
end

function J = Jacobiana(A, x, y, I) % Jacobiana conforme o enunciado
J = [A - y*I, -x; 2*x', 0];
end
```

```
Command Window

  Matriz A:
     3    10     4     5
     9     3     6     8
     6     8     0    10
     6     8     0     1

 ---------------------------------
 Iteração          Valor prório
 ---------------------------------
     0     17.7071146259727463
     1     22.6467477879226919
     2     21.8854648613331122
     3     21.7535203626229396
     4     21.7498164484588763
     5     21.7498148824928670
     6     21.7498148824927249

  Vector próprio:
     0.5145704880856856
     0.5714950333917288
     0.5218750227636221
     0.3691301941257513
fx >>
```

```
Command Window

  Matriz A:
     1     1     7     2
     8     6     8    10
     3     2     4     1
     5     7     0     9

 ---------------------------------
 Iteração          Valor prório
 ---------------------------------
     0     15.6698891524314536
     1     19.1983368701525379
     2     18.6652069951966340
     3     18.6085600953214261
     4     18.6080608670862553
     5     18.6080608465419814
     6     18.6080608465419814

  Vector próprio:
     0.1852362314025996
     0.7321835896286510
     0.1814004008076112
     0.6298322191195872
fx >>
```