

Examen Final de MATLAB (LCS)

Février 2011

Exercice 01: (06 points)

- Qu'obtient-on lorsqu'on exécute les instructions suivantes :

```
>> A = diag(11:-3:1)
>> B = [diag(A);4]
>> C = -2*(ones(3)-2*eye(3))*[5: -2: 0]
>> D = B(3:5) .* C
>> E = A.^2
>> F = B-C
```

- Traduire ces expressions dans le langage Matlab :  $V_{\max} \cos(\omega t + \theta)$  ;  $a e^{(yt+z)}$  ;  $\frac{\sqrt{|b^2 - 4ac|}}{\ln(d + \frac{h}{2})}$

Exercice 02: (04 points)

➤ Ecrire un fichier script Matlab pour l'algorithme suivant :

- Lire les variables :  $a$ ,  $b$  et  $\varepsilon$
- Répéter l'exécution de :

```
➤  $a \leftarrow a/2$ 
➤  $b \leftarrow b/2$ 
```

- Jusqu'à ce que  $|b - a| < \varepsilon$

➤ Dérouler le programme pour  $a=7$  ;  $b=5$  et  $\varepsilon = 10^{-1}$  en donnant, pour chaque itération, les valeurs de  $a$ ,  $b$  et  $|b - a|$ .

Exercice 03: (05 points)

Soit le fichier fonction suivant. Corriger les erreurs de syntaxe que contient le programme et déduire ce qu'il fait.

```
functione (M)=Fait_Quoi[M]
[n,m]=length(M);
for i= 1 to n
    maxL=M(i,1); C=1;
    for j=2 to n
        if maxL < M(i,j) then
            maxL=M(i,j);
            C=j;
        end
    x=M[i,i];
    M(i,i)=maxL;
    M(i,C)=x;
end
```

Exercice 04: (05 pts)

Ecrire une fonction Matlab qui donne le nombre de lignes de la matrice  $A$  contenant au moins un zéro.

Exemple : la matrice  $A = \begin{pmatrix} 1 & 6 & 3 \\ 2 & 0 & -4 \\ 0 & 7 & 0 \end{pmatrix}$  contient deux lignes contenant chacune au moins un zéro.

## CORRIGE DE L'EXAMEN FINAL DE LCS 2010

### Exercice 01: (06 points)

<b>A =</b>				
11	0	0	0	
0	8	0	0	
0	0	5	0	
0	0	0	2	

(0,75)

<b>B =</b>	
11	
8	
5	
2	
4	

(0,75)

<b>C =</b>	
2	
-6	
-14	

(0,75)

<b>D =</b>	
10	
-12	
-56	

(0,75)

<b>E =</b>				
121	0	0	0	
0	64	0	0	
0	0	25	0	
0	0	0	4	

(0,75)

**F = B-C**

??? Error using ==> minus

Matrix dimensions must agree.

(0,75)

$V_{max} * \cos(w*t + teta)$

(0,5)

$a * \exp(y*t + z)$

(0,5)

$\sqrt{\text{abs}(b^2 - 4*a*c)} / \log(d+h/2)$

(0,5)

### Exercice 02: (04 points)

a=7; b=5; epsi=0.1;

(0,75)

k=0;

while abs(b-a) >= epsi

(1)

k=k+1

b=b/2

(1)

a=a/2

err=abs(b-a)

end

Le déroulement du programme :

(0,25) (0,25) (0,25) (0,25) (0,25)

Iterations	0	1	2	3	4	5
a	7	2.5	1.25	0.625	0.3125	0.1563
b	5	3.5	1.75	0.875	0.4375	0.2188
$ b-a $	2	1	0.5	0.25	0.125	0.0625

### Exercice 03: (05 points)

```

functione (M)=Fait_Quoi [M]
[n,m]=length(M);
for i= 1 to n
    maxL=M(i,1);
    C=1;
    for j=2 to n
        if maxL < M(i,j) then
            maxL=M(i,j);
            C=j;
        ..
    end
    x=M[i,i];
    M(i,i)=maxL;
    M(i,C)=x;
end
    
```

#### Programme Corrigé

```

function M=Fait_Quoi (M)
n=length(M);
for i= 1: n
    maxL=M(i,1);
    C=1;
    for j=2:n
        if maxL < M(i,j)
            maxL=M(i,j);
            C=j;
        end
    end
    x=M(i,i);
    M(i,i)=maxL;
    M(i,C)=x;
end
    
```

(0,75)  
 (0,5)  
 (0,25)  
 (0,25)  
 (0,5)  
 (0,25)  
 (0,25)  
 (0,25)

Le programme permet de placer le maximum de chaque ligne d'une matrice carrée A ( $n \times n$ ) sur la diagonale principale.

2

**Exercice4** (05 pts)

```
function [NLZ] = emd_Lcs(A) 1
[n,m]=size(A); 0,5
NLZ=0; 0,5
for i=1:n 0,5
    nz=0;
    for j=1:m 0,5
        if A(i,j)==0 0,5
            nz=1; 0,5
        end
    end
    NLZ=NLZ+nz; 1
end
```