

## MATLAB - TP n°4

### Exercice 1

Programmer les fonctions suivantes sous Matlab et indiquer ce qu'elles réalisent :

<pre>function r = ps1(U,V) r = U * V' ;</pre>	<pre>function r = ps2(U,V) n = size(U,2); r = 0. ; for i = 1:n,     r = r + U(i) * V(i); end</pre>	<pre>function Y=insere(X,a) [m,n] = size(X); Y=a*ones(1,2*n); Y(2 : 1+m , n+1 : end) = X;</pre>
-----------------------------------------------	----------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------

### Exercice 2

Ecrire une fonction Matlab permettant de résoudre une équation du second degré :

$$ax^2 + bx + c = 0.$$

### Exercice 3

Écrire une fonction Matlab qui reçoit en entrée les coordonnées cartésiennes  $(x,y)$  d'un point, et qui obtient en sortie ses coordonnées polaires : le rayon  $r = \sqrt{x^2 + y^2}$  et l'argument  $a = \arctg\left(\frac{x}{y}\right)$ .

### Exercice 4

Ecrire une fonction Matlab permettant de calculer le développement limité de  $\sin(x)$  avec une précision  $\epsilon$  ( $\epsilon$  sera donné en paramètre), en utilisant la boucle *while* :

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} + \dots + \epsilon(x)$$

Indication : Dans le cas de  $\sin(x)$ , quelque soit  $\epsilon$ , il existe un  $n$  tel que :  $\frac{x^{n+1}}{n+1!} < \epsilon(x) < \frac{x^n}{n!}$

### Exercice 5

Ecrire une fonction Matlab,  $Z = \text{put0}(X,Y)$ , qui permet d'ajouter des zéros à la matrice  $X$  aux endroits spécifiés par la matrice à 2 lignes  $Y=[y_{1j}]$  où  $(y_{1j}, y_{2j})$  indiquent respectivement le numéro de ligne et de colonne de la matrice  $X$  à mettre à zéro .

### Exercice 6

Les commandes Matlab *tic* et *toc* permettent de calculer le temps d'exécution des instructions qu'elles englobent. Saisir le script suivant pour s'en apercevoir :

```
tic % démarrage du chronomètre
a = (1:5)';
D = [];
for i = 1:5
    D = [D, i*a];
end
D
plot(D, '-o')
toc % arrêt du chronomètre
```

## Exercice 7

Ecrire un script Matlab où l'on définit le vecteur  $N = 1 : 2 : 12$ , puis on calcule le factoriel (le factoriel de l'entier  $k$  est  $1*2*...*k$ ) de chacun des éléments de  $N$  à l'aide de trois méthodes :

1. en utilisant la fonction built-in *factorial* de Matlab,
2. en utilisant la boucle *for*,
3. en utilisant la boucle *while*.

A l'aide des fonctions *tic* et *toc*, calculer les temps d'exécution de chacune de ces méthodes.

Comparez dans un même graphe ces temps. (voir le corrigé sur la dernière page)

## Exercice 8

Saisir le script Matlab suivant puis expliquer son utilité :

```
s = 'hello'
name = upper(s) % convertir en majuscule
fun = strrep('hahaha','a','i') % remplacer la lettre 'a' par 'i'
greet = 'Welcome';
where = 'to Joan''s';
party = 'birthday party';
final = str2mat(greet, where, party) % construire une matrice de chaines de caractères
text = 'Monday,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday';
[day,rest] = strtok(text,',') % extraire le mot qui se trouve avant ','
[day2,rest] = strtok(rest,',')
```

## Exercice 9

Dans un premier script, *data1.m*, saisir la matrice A et le vecteur x suivants :

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \text{ et } x = [-5 \quad -10 \quad -15]$$

Dans un autre script, *trait1.m*, appeler le script *data1.m* pour charger les matrices A et X, puis saisir les instructions suivantes :

```
D1 = diag(A)
D2 = diag(diag(A))
DX = diag(x)
D3 = diag(A,2)
D4 = diag(diag(A,2))
D5 = diag(diag(A,2),2)
```

## Exercice 10

Rédiger la fonction suivante dans l'éditeur de Matlab. Visualiser le help des nouvelles commandes : *nargin*, *fopen*, ..., puis expliquer l'utilité de cette fonction :

```
function afic(nomfic, nblignes)
if nargin == 1
    nblignes = 20;
end
fic_id = fopen(nomfic,'r');
compt = 0;
while ~feof(fic_id)
    ligne = fgetl(fic_id);
    compt = compt + 1;
    if rem(compt, nblignes) == 0
        pause
    else
        disp(ligne)
    end
end
fclose(fic_id)
```

Pour voir comment elle fonctionne, tester cette fonction dans la fenêtre Matlab avec comme argument un fichier texte, par exemple : **>> afic('afic.m', 3)**

## Corrigé de l'Exercice 7

Voici un script *fact.m* possible pour répondre à cet exercice :

```
% Calcul du factoriel par 3 méthodes
N = 0 : 2 :12;
n = size(N,2);
RES = zeros(3,n); % matrice des résultats
TE = zeros(3,n); % matrice des temps d'exécution

% 1) A l'aide de la fonction built-in "factorial" de Matlab
for k = 1: n
    tic
    RES(1,k) = factorial(N(k));
    TE(1,k) = toc;
end

% 2) A l'aide de la boucle for
for k = 1:n
    tic
    RES(2,k) = 1;
    for i = 1 : N(k),
        RES(2,k) = RES(2,k) * i;
    end
    TE(2,k) = toc;
end

% 3) A l'aide de la boucle while
for k=1:n
    tic
    RES(3,k) = 1;
    i = 2;
    while i <= N(k)
        RES(3,k) = RES(3,k) * i;
        i = i + 1;
    end
    TE(3,k) = toc;
end

format long
disp 'Resultats obtenus par les 3 méthodes : '
RES
disp ' Comparison des temps d''execution : '
TE

plot(N, TE')
title('Comparison des temps d''execution')
xlabel('valeur de N')
ylabel('Temps d''execution en seconde')
legend('factorial', 'for', 'while')
grid
```