

Projet tuteuré Reconnaissance faciale

Travail 1 : Prise en main de Matlab/Octave

Enseignant: Pierre Dreyfuss

email: dreyfuss@unice.fr

CONSIGNES: Suivre l'ordre des questions et y répondre sur une feuille blanche. Ne pas hésiter à utiliser l'aide de Matlab/Octave ou à solliciter l'enseignant !

EXERCICE N. 1

Utilisation des matrices et vecteurs - fichiers .m

- 1) Pour définir des vecteurs lignes,
comme $\mathbf{u} = (1 \ 2 \ 3)$ on peut procéder comme suit:

- i) Méthode directe.

Taper successivement les commandes suivantes:

$$u = [1 \ 2 \ 3]$$

$$u = [1, 2, 3]$$

$$u = [1 \ 2 \ 3];$$

Noter les éventuelles différences.

- ii) Fonctions plus avancées.

Taper successivement les commandes suivantes:

$$u = \text{linspace}(1, 3, 3)$$

$$u = 1 : 1 : 3$$

Comprendre la syntaxe de la dernière écriture et créer un vecteur ligne

$$u1 = (0 \ 0.01 \ 0.02 \ .. \ 0.99 \ 1).$$

- 2) Pour définir des vecteurs colonnes,
on procède de manière similaire. Taper la commande $v = [-5; 2; 1]$. En déduire comment définir les vecteurs

$$\mathbf{w} = \begin{pmatrix} -1 \\ -3 \\ 7 \end{pmatrix} \quad \text{et } \mathbf{w}_1 \text{ comme } u1 \text{ mais en colonne cette fois-ci.}$$

- 3) L'espace et l'environnement de travail.

Taper la commande *whos* et commenter.

Taper *clear u1; whos* et commenter.

Taper *clear all; whos* et commenter.

Pour agir sur votre répertoire de travail vous pouvez utiliser les commandes suivantes: *pwd, ls, cd*.

Que font ces commandes ?

- 4) Les fichiers .m

- i) créer un fichier *ex01.m* contenant uniquement la définition de u comme en 1.i).

Taper ensuite *ex01*.

- ii) créer la même chose mais par l'intermédiaire d'une fonction:

$$\begin{aligned} \text{function } [u] &= \text{ex1} \\ u &= 1 : 1 : 3; \end{aligned}$$

sauver le fichier sous le nom *ex1.m*. Taper *ex1*.

Les procédés 4.i (script) et 4.ii (fonction) sont-ils équivalents ?

- 5) Les matrices

Soit la matrice
$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \quad (1)$$

- i) méthode directe.

Taper la commande: $A = [2 \ -1 \ 0; -1 \ 2 \ -1; 0 \ -1 \ 2]$.

Extraire ensuite le deuxième coefficient de la troisième colonne.

- ii) utilisation de boucles.

Créer un script *exo5.m* dans lequel vous construirez A en complétant (et en corrigeant si besoin) l'esquisse ci-dessous:

```
A = zeros(3,3);
i = 1;
while (...)
A(i,i) = 2;
....
end
for i = 1 : 2
A(i,i+1) = -1;
A(i,i+1) = -1;
end
disp(A) % (ou bien simplement A)
```

iii) extraction-concaténation.

Matlab/Octave dispose d'outils simples et très puissants pour construire ou modifier des matrices. Taper:

```
A = 2 * eye(3,3)
B = -1 * eye(2,2)
A(1:2,2:3)
```

En déduire une manière de construire la matrice (1).

6) Opérations matricielles élémentaires

i) S'ils ne sont plus dans l'espace de travail, reconstruire les vecteurs u, v, w comme précisé aux points 1) et 2). Taper ensuite:

```
u'
u' + 3 * v - w/5
```

Que fait l'opération 'prime' sur un vecteur ?

ii) En utilisant l'aide Matlab:

- * calculer le produit scalaire $\mathbf{v} \cdot \mathbf{w}$ et la norme euclidienne $\|\mathbf{u}\|^2$
- * déterminer les dimensions de A et en extraire le nombre de colonnes
- * calculer le déterminant et l'inverse de A
- * résoudre le problème $A\mathbf{x} = \mathbf{v}$ par deux méthodes différentes
- * calculer le 'produit terme à terme' z de v par w , i.e. $z(k) = v(k)w(k)$ pour tout k

EXERCICE N. 2

Entrées/sorties: clavier et fichiers

1) Entrées/sorties avec le clavier

- i) Compléter le script ci-dessous (puis sauver le par ex. comme *exo2 - 1.m*), afin qu'il permette de résoudre $A\mathbf{x} = \mathbf{v}$ par l'une des 2 méthodes vues dans l'exercice 1 point 6.iii):

```
disp('1→ méthode1 pour résoudre Ax=v ');
disp('2→ méthode 2 pour résoudre Ax=v');
i=input('Votre choix :');
if (i==1)
.....
```

- ii) Créer une fonction (prototype: *function[x] = f6(A,v)*; fichier *f6.m*) qui fait la même chose en renvoyant x si on lui passe la matrice A et le vecteur v .

2) Entrées/sorties de fichiers (image numérique)

Le logiciel Matlab/Octave permet de lire les principaux formats d'image numérique, de convertir l'image en une matrice et de faire un grand nombre d'opérations numériques. Par ailleurs, puisque l'image est au final représentée par une matrice, vous pourrez créer vos propres opérations numériques (en combinant des algorithmes que vous développerez et des fonctions Matlab préexistantes). Pour le moment nous nous limitons aux opérations de base.

- i) lecture de fichiers.

Sur le serveur U:, dans le répertoire P_ Dreyfuss, vous trouverez un répertoire images qui contient des images numériques sous différents formats. Récupérer ces images, et charger les en mémoire.

Taper par exemple :

```
I1=imread('mandril.tif');
I2=imread('lena.tif');
I3=imread('o-png24.png');
I4=imread('mandril_color.tif');
```

Remarque: si les fichiers en question ne sont pas dans votre répertoire courant il faut soit indiquer le chemin d'accès correct pour les fichiers, soit changer de répertoire de travail.

Quels sont les formats de ces fichiers image ?

Quels sont les tailles des matrices $I1, I2, ..$ correspondantes ?

Que représentent les valeurs quelles contiennent ?

Qu'y-a-t-il comme différence entre une image en niveaux de gris et une image couleur ?

- ii) Affichage les images correspondantes.

Utiliser la commande *imshow*

- iii) Modification des matrices - sauvegarde des fichiers

Modifier l'image représentée par *I1* en insérant une ligne blanche horizontale au milieu. Stocker le résultat dans la variable (matrice) *J1* puis sauvegarder l'image correspondante (voir commande *imwrite*) dans un fichier que vous nommerez *toto*.

EXERCICE N. 3

Tracés de courbes et de surfaces

- 1) Tracé de courbes.

Construire un vecteur ligne x contenant 50 points uniformément répartis sur $[0, 2\pi]$. Construire un vecteur ligne y de la même taille que x et tel que $y(k) = \sin(x(k))$. Tracer le graphe de $\sin(x)$ sur $[0, 2\pi]$ à l'aide de la fonction *plot*.

- 2) Tracé de surfaces.

Garder x comme au point 1) et reconstruire y pour qu'il contienne 100 points uniformément répartis sur $[0, 4\pi]$.

Utiliser *meshgrid* pour réaliser une grille à 50×100 points du rectangle $[0, 2\pi]_x \times [0, 4\pi]_y$.

Utiliser *mesh* pour tracer la surface d'équation $z = \sin(x) + \sin(y)$ sur la grille précédente.