TP 5 (programmation octave) - corrigé

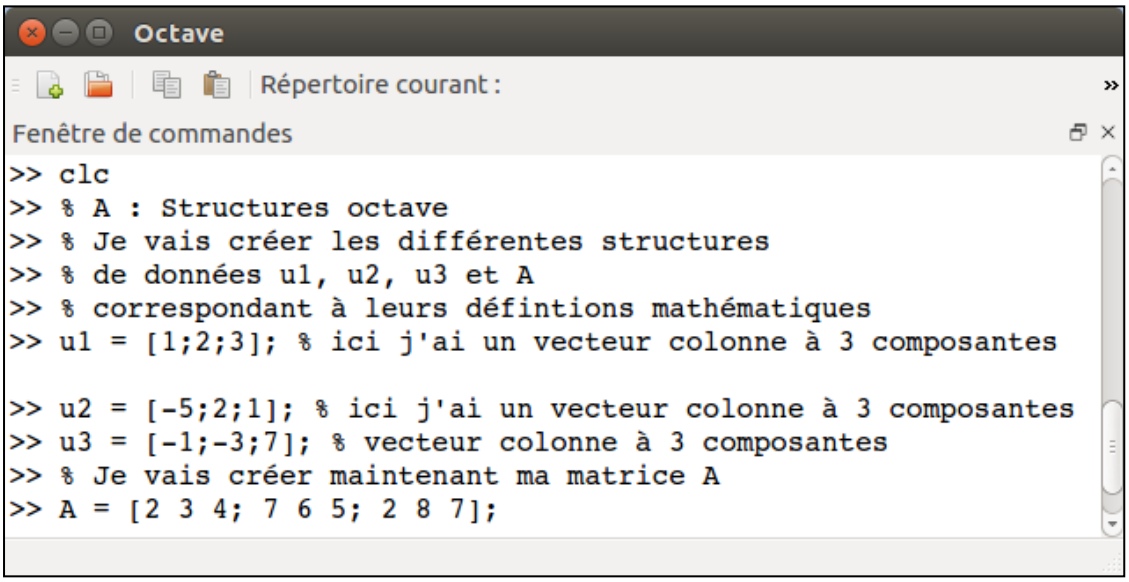
Exercice 1 : Soient les vecteurs colonnes et la matrice suivants

$$\vec{u}_1 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \ \vec{u}_2 = \begin{pmatrix} -5 \\ 2 \\ 1 \end{pmatrix}, \ \vec{u}_3 = \begin{pmatrix} -1 \\ -3 \\ 7 \end{pmatrix}, \ A = \begin{pmatrix} 2 & 3 & 4 \\ 7 & 6 & 5 \\ 2 & 8 & 7 \end{pmatrix}.$$

A - Structures Octave

1. Entrer ces données sous Octave.

   Réponse :



2. Calculer $\vec{u}_1 + 3\vec{u}_2 - \vec{u}_3/5$

   Réponse :

```
Fenêtre de commandes
>> % Calculer u1+3u2-u3/5
>> u1+3*u2-u3/5
ans =

  -13.8000
    8.6000
    4.6000
```

3. Calculer le produit scalaire entre les vecteurs $\vec{u}_1$et $\vec{u}_2$

   Réponse :

   $(\vec{u}_1 * \vec{u}_2)$ *Produit matriciel. Le nombre de colonnes de l'argument de gauche doit être égal au nombre de lignes de l'argument de droite, à moins que l'un des deux arguments ne soit un scalaire auquel cas le produit applique le scalaire sur tous les éléments du vecteur ou de la matrice.*

```
Fenêtre de commandes
>> % Calcul du produit scalaire
>> u1*u2
error: operator *: nonconformant arguments (op1 is 3x1, op2 is 3x1)
```

Vous voyez ici que le produit scalaire entre u1 et u2 ne fonctionne pas car le nombre de lignes du premier vecteur est différent du nombre de colonne du second vecteur.
Essayons maintenant avec la fonction prédéfinie « dot » :

```
Fenêtre de commandes
>> % Calcul du produit scalaire avec la fonction dot:
>> dot(u1, u2)
ans =  2
```

*Hum ! Paradoxalement, ma fonction dot me contredis, car avant ça n'a pas marché avec l'opérateur « * ». En fait, dot (u1, u2) calcule la produit scalaire des 2 vecteurs u1 et u2 (ligne ou colonne). Equivalent à **u1 * u2'** s'il s'agit de vecteurs-ligne, ou à **u1' * u2** s'il s'agit de vecteurs-colonne*

4. Calculer le produit $A\,\vec{u}_1$

   Réponse :

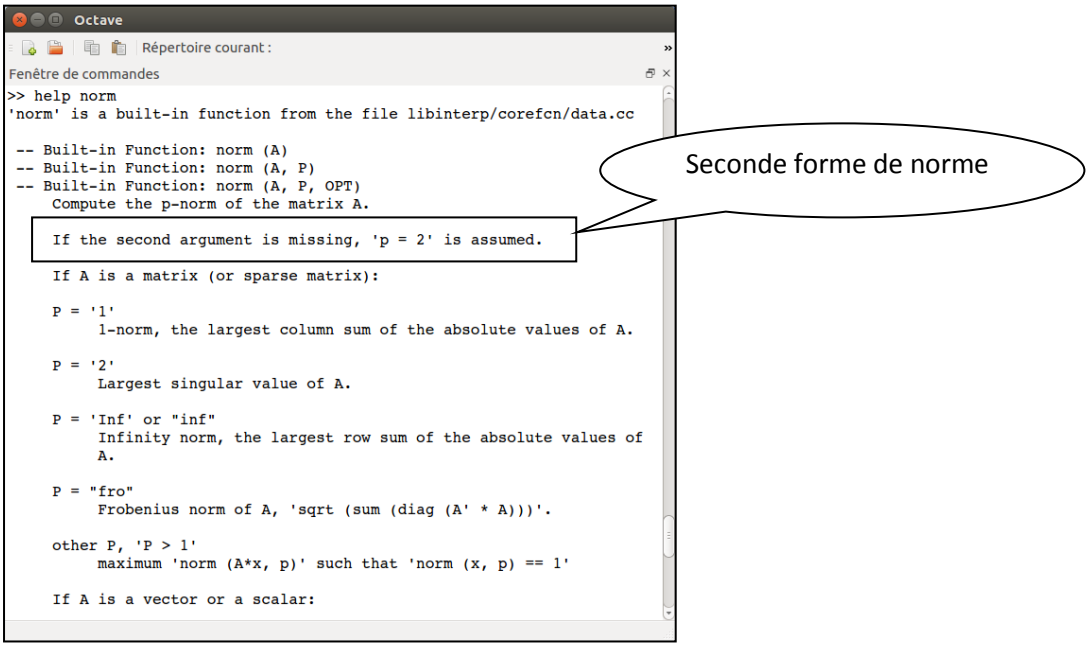
```
Fenêtre de commandes
>> A*u1
ans =

   20
   34
   39
```

B – Commandes Octave : Trouvez les commandes octave permettant de :

1. Calculer les normes (mot clé **norm**) des vecteurs $\vec{u}_1$, $\vec{u}_2$ et $\vec{u}_3$

   Réponse :

   D'abord, je fais une recherche dans l'aide d'octave en tapant « help norm)



*Oulla ! Je me rends compte qu'il y a plusieurs manières de calculer la norme d'un vecteur ou d'une matrice. C'est la fonction « **norm** » qui fait cela. Mais je constate que je peux lui fournir de 1 à 3 paramètres. Le premier indique la matrice ou le vecteur pour lequel je calcul la norme, le second paramètre indique le type de norme que je veux calculer, le troisième*

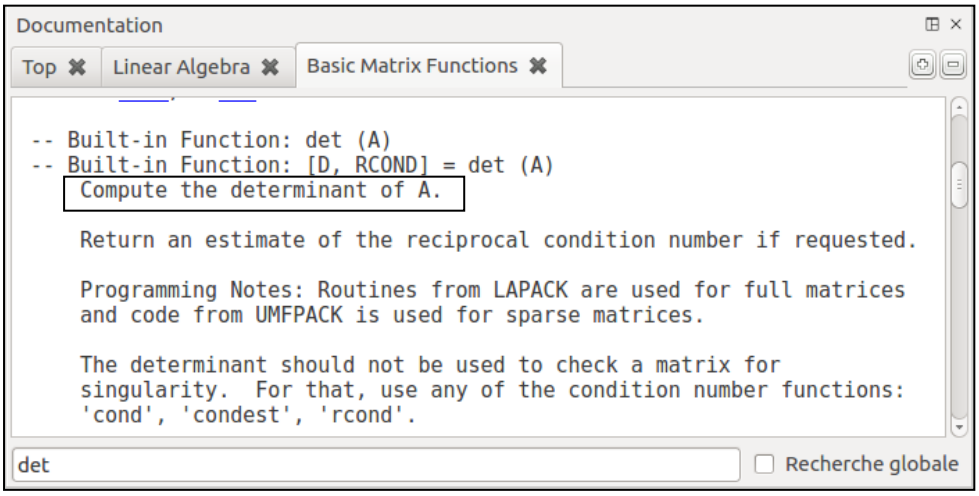
indique d'autre options. Si j'indique uniquement le premier paramètre, alors on calcul le second type de norme (2-norm). C'est ce qu'on va faire ici :

```
Fenêtre de commandes
>> disp(norm(u1))
 3.7417
>> disp(norm(u2))
 5.4772
>> disp(norm(u3))
 7.6811
```

2. déterminer les dimensions de la matrice A et d'en extraire le nombre de colonnes
   Réponse : *Pour déterminer les dimensions d'une matrice, on utiliser la fonction builtins « size »). Cette dernière me renvoi un vecteur-ligne à deux composantes : le premier est le nombre de lignes, la seconde est le nombre de colonnes. Je peux, donc extraire directement, à partir de cette fonction, le nombre de colonne comme suit :*

```
Fenêtre de commandes
>> size(A)(2)
ans =  3
>>
>> % équivalent à:
>> columns(A)
ans =  3
```

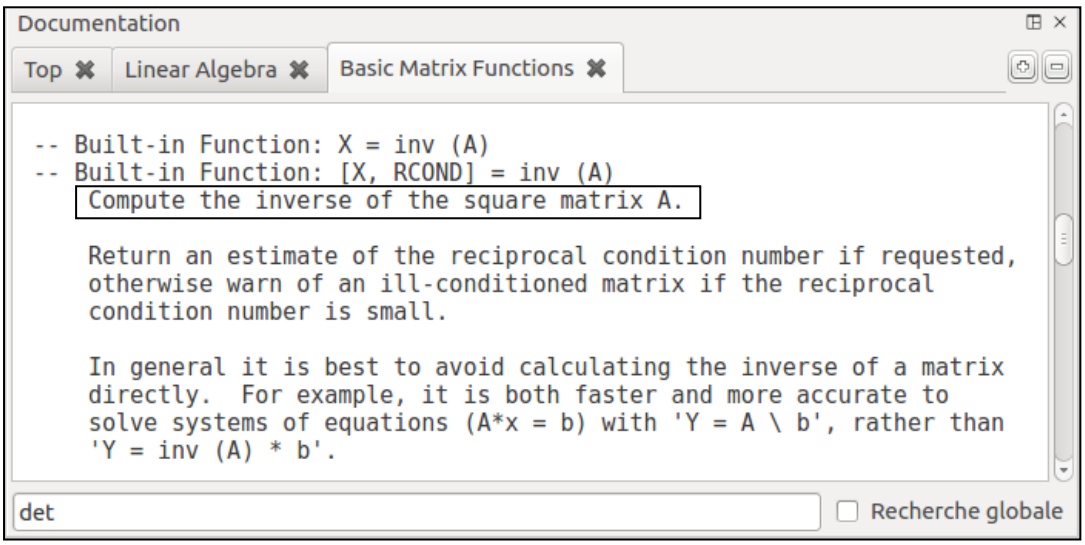
3. calculer le déterminant et l'inverse de *A*
   Réponse : *Nous n'avions pas vu, dans le cours la fonction permettant de calculer le déterminant d'un vecteur ou d'une matrice. Je vais, donc chercher dans la documentation d'octave en allant à la fenêtre de documentation. Afin d'tre efficace, je prends en compte que je m'interesse à la fonction permettant de calculer un déterminant qui relève de l'algèbre linéaire et il s'agit d'une fonction de base sur les matrices. C'est ce qui m'oriente vers la rubrique : « Basic Maxtrix Funcitons ». En cherchant dans cette page, je peux trouver la définition de cette fonction comme illustré ci-dessous :*



J'applique donc cette fonction :

```
Fenêtre de commandes
>> det(A)
ans =  63.000
```

Je passe maintenant à la fonction qui calcul l'inverse d'une matrice. Je reste donc dans ma page de documentation « Basic Maxtrix Funcitons » et je recherche une fonction qui répond à mon besoin (inversion de matrice) et je tombre sur ceci :



Je vois, ici que cette fonction n'est possible que sur des matrices carrées, ce qui est le cas pour ma matrice A. Je vais, donc appliquer cette fonction sur A comme suit :

```
Fenêtre de commandes
>> inv(A)
ans =

   0.031746   0.174603  -0.142857
  -0.619048   0.095238   0.285714
   0.698413  -0.158730  -0.142857
```

Youpi : j'ai obtenu une matrice carré. En principe, si je multiple (produit scalaire) A fois son inverse, je devrais trouver la matrice unité. Essayons cela :

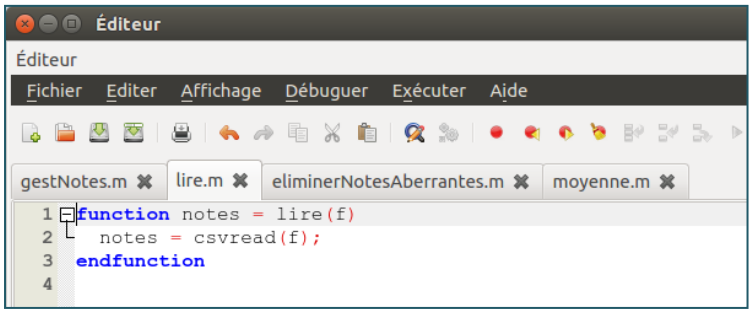
```
>> A * inv(A)
ans =

   1.00000   0.00000   0.00000
   0.00000   1.00000   0.00000
   0.00000  -0.00000   1.00000
```

C - Résolution de systèmes d'équations linéaires.

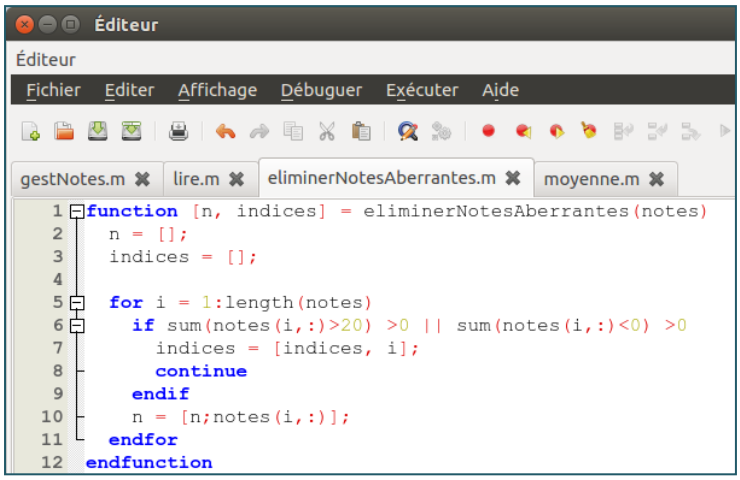
Je vous informe que vous pouvez à l'aide de la fonction « ***inv***() » calculer l'inverse d'une matrice. Vous pouvez aussi faire la division d'une matrice par un vecteur avec l'opérateur « **** ». A l'aide de cette fonction et de cet opérateur, je vous demande de résoudre le système d'équation suivant : $A\vec{x} = \vec{u}_1$.

Exercice 2 : Je mets à votre disposition un fichier texte nommé « **notes.data** ». Ce fichier contient 20 lignes et 4 colonnes. Chaque ligne correspond à un étudiant. Chaque colonne correspond à un module (algèbre, analyse, algorithmique et programmation). Je vous demande :

1. lire depuis le fichier « notes.data » les notes dans une variable nommée « notes ». Vous utiliserez la fonction « csvread » (reportez-vous à l'aide d'octave pour avoir des explications de son usage).

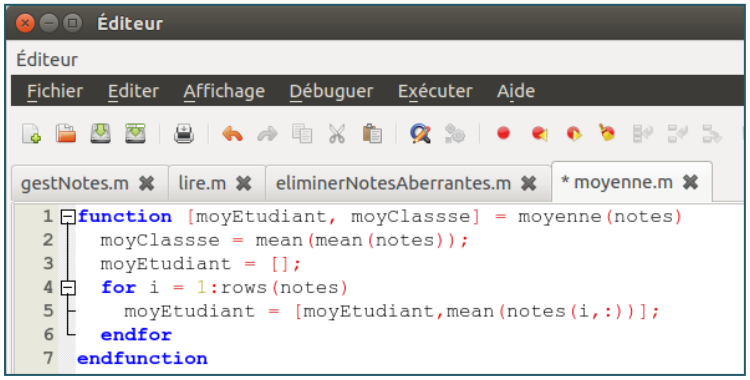


2. Ecrire une fonction « eliminerNotesAberrantes.m » qui recherche dans la matrice des notes (les lignes représentent les étudiants et les colonnes représentent les modules) les lignes comportant des notes aberrantes (non comprises entre 0 et 20). Cette fonction doit rendre une matrice débarrassée de ces lignes

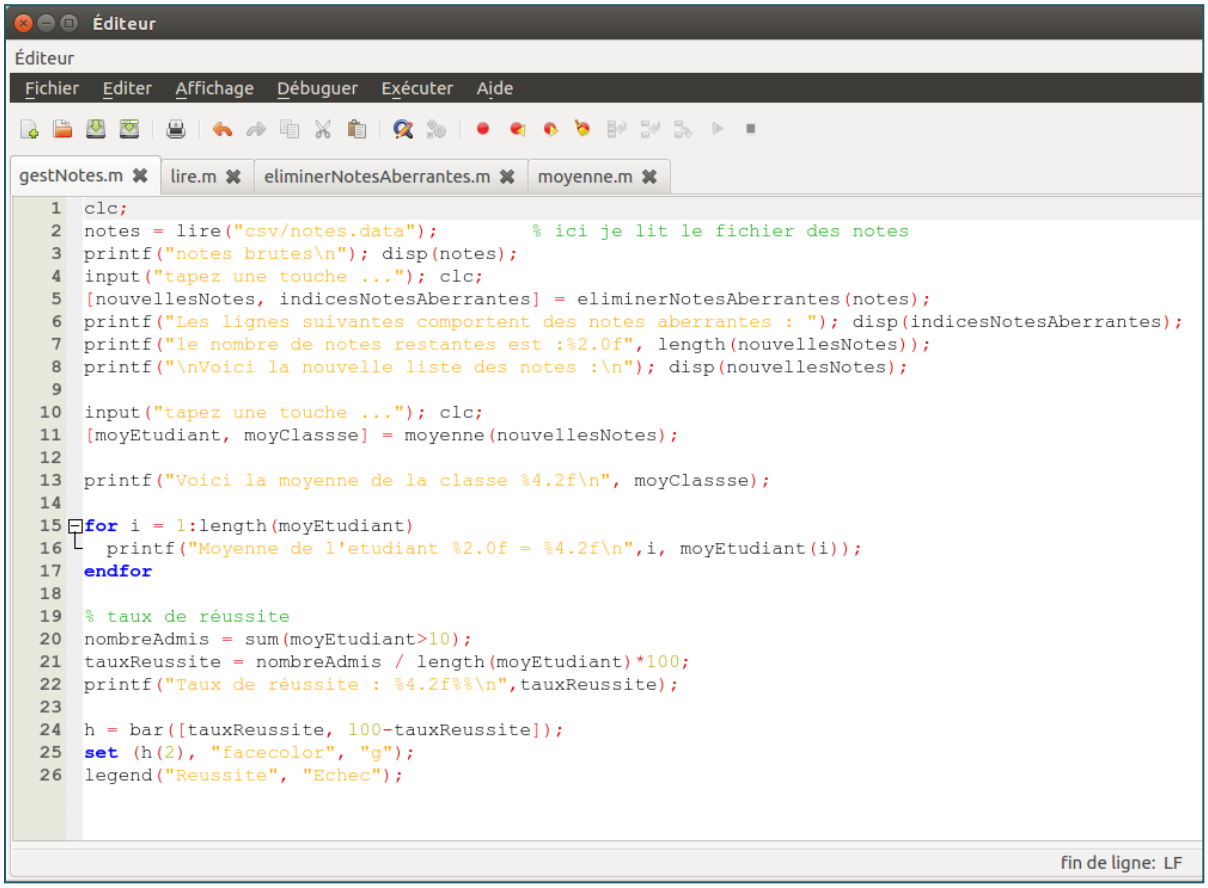


3. Ecrire une fonction « eliminerNotesManquantes.m » qui recherche dans la matrice des notes les lignes comportant des notes manquantes (une note manquante correspondant à la valeur NA pour dire note avalable). Cette fonction doit rendre une matrice débarrassée de ces lignes…
   Non fait !

4. Une fois les notes débarrassées des lignes comportant des notes aberrantes ou manquantes, vous devez écrire une fonction qui renvoi la moyenne pour chaque étudiant et la moyenne de la classe. Il faut aussi qu'elle affiche un graphique du taux de réussite (pourcentage des étudiants ayant eu plus de 10/20).



5. Ecrire un script « gestNotes.m » qui automatise les 4 opérations précédentes



Exercice 3 : Boucles et fonctions

Question 1 : Créez un tableur ***tab*** contenant des entiers multiples de 3 compris entre 3 et 39 puis écrivez un script qui parcourt ce tableau et remplace chacune des valeurs par son carré.

Question 2 : Ecrivez une fonction « **cos2** » permettant de calculer le carré du cosinus de x

$$\cos 2 : \left\{ \begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R} \\ x & \mapsto & cos^2(x) \end{array} \right.$$

Question 3 : Ecrivez un script qui affiche un menu composé de 3 options :

1 – affichez la courbe 2D d'une fonction

2 – afficher la courbe 3D d'une fonction

3 – Quitter

Lorsque l'utilisateur choisi l'option 1 : votre script doit faire appel à la fonction « courbe2D.m »

Lorsque l'utilisateur choisi l'option 2 : votre script doit faire appel à la fonction « courbe3D.m »

Lorsque l'utilisateur choisi l'option 3 : votre script doit s'arrêter sinon, il doit réafficher le menu.

La fonction « courbe2D.m » doit afficher le graphique 2D de la fonction $f(x)=\sin^2(x)$ sur l'intervalle $[-2\pi, +2\pi]$.

Vous devez afficher le titre du graphique qui est « cours 2D » le label de l'axe x est « temps », le label de l'axe des y est « tension », la couleur de la courbe doit être rouge, l'épaisseur du trait doit être 2.

La fonction « courbe3D.m » doit afficher le graphique 3D de la courbe définie par les vecteurs x, y et z.

- le vecteur x est composé de 100 valeurs équidistantes comprises entre $[-4\pi, +4\pi]$
- les éléments de y sont les sinus des éléments de x
- les éléments de y sont les cosinus des éléments de x