TP 2 Calcul matriciel

Le but de cette séance de TP est de vous permettre d´approfondir les notions fondamentales de calcul matriciel vues en cours de mathématiques et ceci à l'aide du logiciel **Scilab**. En effet, tout est matrice en Scilab comme vous avez déjà pu le remarquer en faisant par l´exemple la représentation temporelle ou bien fréquentielle d´un signal. En Scilab, tout calcul, programmation, calcul ou tracé graphique se fait à partir des matrices rectangulaires : ainsi les scalaires sont des matrices *1x1*, les vecteurs des abscisses temporelles sont des vecteurs lignes (*1xn*), etc...

1.1 – Déclarer une matrice sous Scilab: premiers pas

Commençons par apprendre ou revoir les notions élémentaires utiles pour créer des matrices :

- ➔ Créer un vecteur ligne avec des nombres compris entre 1.2 et 4.8 par incrément de 1 :
  ```
  -->v1=1.2:4.8
  ```
- ➔ Créer un vecteur ligne avec des nombres compris entre 1.2 et 4.8 par incrément de 0.5 :
  ```
  -->v2=1.2:0.5:4.8
  ```
- ➔ Créer avec la commande ***linspace(a,b,n)*** un vecteur ligne constitué de ***n*** nombres régulièrement espacés entre ***a*** et ***b*** :
  ```
  -->v3=linspace(1.2,4.8,8)
  ```
  NB: Les vecteurs v2 et v3 constitués tous les deux de 8 nombres compris entre 1.2 et 4.8 ne sont pas cependant identiques.
- ➔ Créer un vecteur ligne par affectation:
  Par exemple, le vecteur (-1, 1.4, 2.3, π) est obtenu avec la commande :
  ```
  -->v4=[-1,1.4,2.3,%pi]
  ```
- ➔ Créer une matrice avec ***n*** lignes et ***m*** colonnes :
  Par exemple, la matrice A avec 2 lignes et 3 colonnes : $A=\begin{pmatrix}1 & 3 & 5\\ 2 & 4 & 6\end{pmatrix}$
  est obtenue avec la commande :
  ```
  -->A=[1 ,3 ,5 ; 2, 4, 6]
  ```
  ou bien
  ```
  -->A=[1 3 5 ; 2 4 6]
  ```
  N.B : Comme vous pouvez le noter, le changement de colonne est marqué par un espace ou une virgule, tandis que ; indique un changement de ligne.
  La commande ***size*** permet de récupérer la taille d´une matrice. Ainsi
  ```
  -->size(A)
   ans= 2.  3.
  ```
  Quelle est la réponse affichée si vous tapez ?
  -->size(A,1)
  Ou bien ?
  ->size(A,2)

Tapez le programme suivant et executez-le:

```
clear;
n=4;
m=5;
function [a]=sign_matrix(n, m)
   for i=1:n
      for j=1:m
            a(i,j)=(-1)^(i+j)
      end
end
endfunction
A=sign_matrix(n,m)
disp(A)
```

Quelle est la particularité de la matrice ***A*** ? Notez les lignes de script qui permettent de déclarer la fonction sign_matrix. Après l´execution de ce script, vous pourrez continuer à l´appeler dans la console ou dans un autre script tant que vous n´utilisez pas la commande clear.

Ecrivez maintenant une fonction qui permet de construire la matrice ***B*** de taille ***(n x m)*** telle que : $(b_{ij})=1$ si $i \leq j$ sinon $(b_{ij})=0$. Affichez la matrice ***B*** pour ***n=3*** et ***m=4***. Ou encore pour ***n=m=4***. Quelle est la particularité de la dernière matrice ?

1.2 – Matrices particulières : matrice zéro, identité, matrice diagonale, matrice remplie que de 1, matrice vide, etc...

En Scilab, il existe des matrices particulières. On a ainsi des matrices constantes telles que ones, zeros, eye. La matrice ones est une matrice n'ayant que des 1 pour coefficient, et idem pour zeros mais avec des 0. La matrice créée par l´instruction eye(n,m) est une matrice ***n x m*** avec des 1 sur la diagonale et des 0 ailleurs, et qui devient la matrice identité d´ordre ***n*** si ***n=m***.

On peut construire d´autres matrices particulières :
Tapez la commande suivante :

```
-->x=[1 2 3 -1], D=diag(x)
```

Quelle est la particularité de la matrice ***D*** ?

Tapez ensuite :

```
-->D1=diag(x,2), D2=diag(x,-1)
```

Quelles sont les tailles de ces deux matrices et quelle est leur particularité ?

Enfin, on peut citer deux matrices particulières : La matrice vide est créée en tapant simplement []. La instruction rand(n,m) crée une matrice aléatoire ***n x m*** avec comme coefficients des nombres aléatoires entre 0 et 1.

Tapez les commandes suivantes:

```
-->R=rand(2,3)
-->a1=rand();a2=rand(1-3*%i)
-->a3=rand()
```

Que pouvez vous dire sur les variables a1, a2 et a3 ?

1.3 – Autres méthodes pour créer de matrices :

➔ Création de matrices par concaténation :
De grandes matrices peuvent être définies à partir de plus petites.
Tapez :
```
-->M1  =[1 2 ; 4 5; 7 8],  M2=eye(3,4), M3=ones(1,6)
-->M4  =[M1, M2],  M5=[M1,M2;M3], M6=[M4;M3]
```
Quelles sont les tailles de nouvelles matrices ***M4***, ***M5*** et ***M6*** ? Quelles sont les précautions à prendre quand on « colle » de petites matrices ? Quelle est la différence entre les matrices ***M5*** et ***M6*** ? Quelle est le rôle des séparateurs **,** et ***;*** ci-dessus ?

➔ Extraction de matrices, suppression de vecteurs lignes ou de vecteurs colonnes :
En Scilab, il existe de nombreuses méthodes ou commandes pour extraire des sous-blocs d´une matrice.
Tapez :
```
-->M5(2,4), M5(2,:),M5(:,4)
```
A quoi correspondent les valeurs/vecteurs affichées ?
Tapez ensuite:
```
-->n1=M5(2:4,2:4), n2=M5([2,4],[2,4])
```
ainsi que
```
-->n3=M5([2,3,4],2:4)
```
Les matrices ***n1, n2*** et ***n3*** sont sous-blocs de la matrice ***M5*** obtenues en utilisant des instructions différentes. Sont-ces matrices identiques ? Justifiez.

➔ Il existe également des fonctions d'extraction incorporées, telles que `diag, tril, triu.` Qu´est-ce qui est affiché si on applique ces 3 fonctions à la matrice ***M5*** ?

1.4 – Opérations et fonctions sur les matrices

On peut effectuer des opérations sur les matrices :

➔ Addition : ***A+B***, à condition que les 2 matrices soient de mêmes tailles
➔ Le produit : ***A*B***, à condition que les 2 matrices aient des tailles compatibles (voir le cours de mathématiques).
➔ Produit par un scalaire : ***3*A***
➔ Puissance ***n***-ième d´une matrice carrée : ***A^n***
➔ La transposée d´une matrice est créée par : ***A'***
➔ Le produit coefficient par coefficient de deux matrices de mêmes tailles : ***A.*B***
➔ La matrice constituée des puissances ***n***-ièmes des coefficients de ***A*** : ***A.^n***
➔ La matrice issue de la division coefficient par coefficient de deux matrices de mêmes tailles : ***A.\B*** et ***A.\B***.
Tapez :
```
-->A=ones(2,3) , B=2*A
-->AB1=A./B
-->AB2=A.\B
```
Les matrices ***AB1*** et ***AB2*** ne sont pas égales. Justifiez.

On peut également utiliser des matrices comme argument pour les fonctions usuelles numériques telles que : `abs, sin, cos, tan, exp, log, log10`, etc..., à condition bien sûr que leurs coefficients appartiennent aux domaines de définition de ces fonctions.

Il existe également des fonctions incorporées qui calculent la somme et le produit de tous les coefficients d´une matrice : `sum, prod.`

1.6 – Matrices carrées : déterminant, inverse

Pour les matrices carrées, l´application des fonctions `det` et `inv` permettent de calculer leur déterminant et leur inverse, s´il existe bien sûr.
Reprenons les questions b) et c) de l´exercice 4 du TD2 de MA3.
Déclarer la matrice M et le vecteur b

```
-->M=[2 1 -1; 2 -1 2; 3 0 1], b=[ 2 ;1; 3]
```

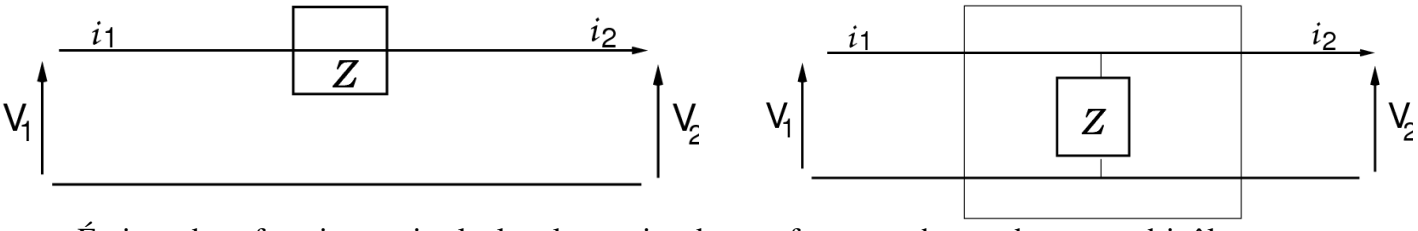
Tapez ensuite :

```
- ->x1=inv(M)*b, x2=M\b
```

Que remarquez vous ?

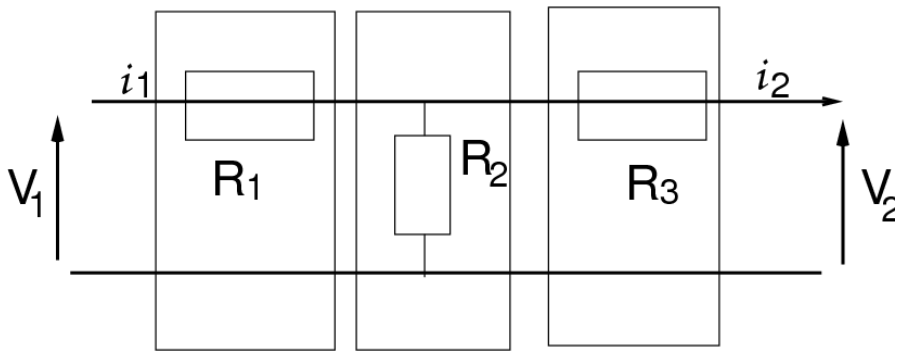
2 – Applications : quadripôles

2.1 - Soient les quadripôles élémentaires **série et parallèle** représentés ci-dessous :



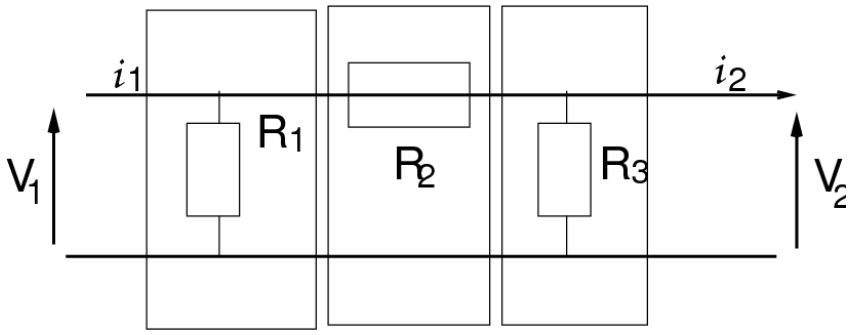
Écrivez deux fonctions qui calculent la matrice de transfert pour chacun de ces quadripôles.

2.2 - On considère le quadripôle **en T** issue de l'association de trois quadripôles en cascade comme représenté ci-dessous.



Écrivez un programme calculant la matrice de transfert équivalente en utilisant les fonctions définies pour les quadripôles élémentaires.

2.3 - On considère le quadripôle **en Π** issue de l'association de trois quadripôles en cascade comme représenté ci-dessous.



Écrivez pour ce montage aussi un programme calculant sa matrice de transfert en utilisant les fonctions définies pour les quadripôles élémentaires.