ECE2

TP n°1 : Matrices et vecteurs

Exercice 1. Tests sur la création de vecteurs et de matrices

On suppose que n un entier supérieur ou égal à 2 a été créé numériquement (prenez par exemple $n = 5$).

1. Écrivez une commande permettant de créer une liste de -3 à n avec un pas de 2.
2. Écrivez une commande permettant de créer une liste de pas constant de -3 à n et comportant 6 points.
3. Quelle est la différence entre les commandes suivantes ?

   `u=n:3:1` et `linspace(n,1,3)`
4. Quel est le vecteur renvoyé par la commande suivante ?

   `u=n:-n`
5. Quelle est la matrice renvoyée par la suite d'instructions qui suit ?

   `u=1:3; v=ones(1,3); w=-1:2:4; x=[u;v;w]`

Exercice 2. Création efficace de vecteurs à l'aide d'opérations arithmétiques pointées

1. Écrire une seule commande permettant de créer le vecteur $x = \left(5, \frac{5}{2}, \frac{5}{3}, \frac{5}{4}, \dots, \frac{5}{10}\right)$ sans saisir un à un les éléments.
2. Même question pour le vecteur $y = \left(1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \dots, \frac{1}{100}\right)$.
3. Même question pour le vecteur $z = \left(1, 2, 4, \dots, 2^{10}\right)$.

Exercice 3. Création de matrices

Écrire une ligne de commandes permettant de créer la matrice de $\mathcal{M}_n(\mathbb{R})$ dont les éléments diagonaux sont égaux à a et les autres éléments égaux à b, pour a, b et n entrés par l'utilisateur.

Indication : On rappelle que `eye(n,n)` *permet de construire la matrice carrée identité de taille n.*

Exercice 4. Min et Max

`Scilab` sait trouver le plus grand élément (c'est la fonction `max`) et le plus petit élément (c'est la fonction `min`) d'un vecteur ou d'une matrice.

1. Écrire une commande permettant de déterminer le plus grand multiple de 7 inférieur ou égal à 1000.
2. Écrire une commande permettant de déterminer le plus petit multiple de 7 supérieur ou égal à 1000.
3. Écrire une commande permettant d'entrer trois entiers naturels n, p et q puis de déterminer le plus petit entre le plus grand multiple de p inférieur ou égal à n et le plus grand multiple de q inférieur ou égal à n.

Exercice 5. Autour de `sum` et `cumsum`

On suppose que n a été créé numériquement.

1. Écrire une ligne de commandes qui renvoie la somme des n premiers entiers naturels non nuls.
2. Écrire une ligne de commandes renvoyant la somme $\sum_{k=1}^{n} \frac{1}{k}$.
3. Que calculent les commandes suivantes ?
   (a) `x=ones(1,n); y=cumsum(x)`
   (b) `x=ones(1,n); y=sum(cumsum(x))`
   (c) `x=ones(1,n); y=sum(cumsum(cumsum(x)))`

Exercice 6. La boucle `for`

On vient de voir que l'on pouvait écrire $\sum_{k=1}^{n} \frac{1}{k}$ avec la commande `sum`. Sans utiliser la commande `sum`, proposer une suite de commandes utilisant une boucle `for` et permettant de calculer, pour un entier naturel n donné la valeur de

$$u_n = \left(\sum_{k=1}^{n} \frac{1}{k}\right) - \ln(n).$$

Tester pour $n = 100$, $n = 1000$, $n = 10000$.

Exercice 7. La boucle `while`

Écrire une ligne de commandes permettant de déterminer le plus petit entier naturel n pour lequel

$$\sum_{k=1}^{n} \frac{1}{k} > 10$$

Exercice 8. Suite récurrente

On définit la suite (u_n) par $u_0 \in \mathbb{R}$ et pour $n \in \mathbb{N}$

$$u_{n+1} = \frac{1}{2}\sqrt{3 + u_n^2}.$$

1. Écrire une ligne de commandes demandant une valeur de n et u_0 et permettant de calculer le terme u_n.
2. Ajouter une commande permettant de calculer la somme $\displaystyle\sum_{k=0}^{n} u_k$.
3. On admet que la suite (u_n) converge et que sa limite vaut 1. Écrire une suite de commandes qui permette de déterminer le plus petit entier naturel n pour lequel $|u_n - 1| \leq 10^{-3}$, u_0 étant entré par l'utilisateur.