I– Résolution d'un système linéaire

1. Rappel de la résolution matricielle

feuilles de TD Scilab online

Propriété

Un système linéaire $\begin{cases} ax+by+cz=k_1 \\ dx+ey+fz=k_2 \\ gx+hy+iz=k_3 \end{cases}$ peut s'écrire sous la forme matricielle $A \times X = Y$,

avec $A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$, $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ et $Y = \begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix}$

Si la matrice A est inversible, alors les solutions de ce système s'obtiennent par $X = A^{-1} \times Y$

2. Résolution grâce à Scilab

`syntaxe Scilab`

- Si `A` est une matrice carrée inversible, les instructions `inv(A)` ou `A^(-1)` renvoient la matrice inverse de A.
- Si `AX = Y` est la représentation matricielle d'un système linéaire, et `A` est une matrice carrée inversible, alors l'instruction `X = inv(A)*Y` renvoie le vecteur `X` solution du système.
  L'instruction `X = A\Y` est équivalente à `X = inv(A)*Y`, mais moins parlante mathématiquement.

Exercice 1
Résoudre (avec Scilab...) le système linéaire

$$\begin{cases} 2x-y-z=0 \\ x+4y+z=12 \\ -3x+y+2z=2 \end{cases}$$

Exercice 2
La société Ultra-eau (U) règne sur le marché des fontaines d'eau à bonbonnes dans les entreprises d'une grande ville. Elle fournit ses clients en recharges pour les fontaines et dispose des résultats antérieurs suivants :

Nombre de recharges en milliers	1	3	5
Coût total annuel de production en centaines d'euros	11	27,4	83

Le coût total de production est modélisé par une fonction C définie pour tout nombre réel x de l'intervalle $[0\,;\,10]$ par :

$$C(x) = ax^3 + bx^2 + cx + 10 \qquad a, b \text{ et } c \text{ sont des nombres réels.}$$

Lorsque le nombre x désigne le nombre de milliers de recharges produites, $C(x)$ est le coût total de production en centaines d'euros.

1. Écrire le système d'équations vérifié par les nombres a, b et c.
2. À l'aide de Scilab, résoudre matriciellement ce système.
3. Donner le coût total annuel de production pour 8000 recharges d'eau produites.

II– Simulation d'expériences aléatoires

`rand()` et `grand()`

- *(rappel)* La commande `rand()` renvoie un nombre pseudo-aléatoire entre 0 et 1.
- La commande `grand(m,n,'uin',a,b)` renvoie une matrice de taille $m \times n$ dont les éléments sont des **entiers** suivant une loi uniforme sur $\{a, a+1, \dots, b\}$.
- La commande `grand(m,n,'unf',a,b)` renvoie une matrice de taille $m \times n$ dont les éléments sont des réels suivant une loi uniforme sur $[a; b[$.
- La commande `grand(m,n,'bin',N,p)` renvoie une matrice de taille $m \times n$ dont les éléments suivent une loi binômiale de paramètres (N, p).

Exercice 3

1. Construire une matrice 3×3 contenant des nombres réels pseudo-aléatoires entre 0 et 1.
2. Construire un vecteur contenant le résultat de 10 lancers successifs d'un dé à 6 faces.
3. Jeu : on lance 3 fois une pièce de monnaie et on compte le nombre de Pile obtenus. Construire un vecteur contenant le résultat de 10 parties de ce jeu.

III– Extraits de sujets de concours

Exercice 4 *extrait BCE ESCP 2018*
On suppose que l'on dispose d'un stock illimité de boules rouges et de boules blanches indiscernables au toucher. Une urne contient initialement une boule rouge et une boule blanche indiscernables au toucher.
On effectue dans cette urne une succession d'expériences aléatoires selon le protocole suivant : on extrait une boule de l'urne et après chaque tirage, la boule tirée est remise dans l'urne et on ajoute dans l'urne une boule de même couleur que la boule tirée.
Pour tout entier $n \geqslant 1$ on note X_n (respectivement Y_n) la variable aléatoire égale au nombre de boules rouges (respectivement blanches) contenues dans l'urne à l'issue de la n-ième expérience, c'est-à-dire après le tirage d'une boule et la remise d'une boule supplémentaire.
Compléter le script `Scilab` suivant pour qu'il simule une réalisation de la variable X_n où l'entier n est entré au clavier :

```
n = input('entrer la valeur de n : ')
r = 1 ; b = 1
for k = 1:n
    if rand()<r/(r+b) then .........
                      else .........
    end
end
x = ...
disp(x)
```

rajout : étudier graphiquement l'évolution de la proportion des boules rouges.

Exercice 5 *extrait BCE BSB 2018*
Une urne contient une boule rouge et deux boules blanches. Un joueur effectue trois tirages successifs d'une boule dans cette urne. Il remet la boule obtenue dans l'urne après chaque tirage.
À partir du deuxième tirage, le joueur reçoit un point à chaque fois que la couleur obtenue à un tirage n'est pas celle qui a été obtenue au tirage précédent. Dans le cas contraire il ne reçoit aucun point.
Ainsi, si les trois tirages successifs amènent : blanc, rouge, rouge le joueur marque un point au deuxième tirage et aucun au troisième tirage.
Soit G la variable aléatoire égale au nombre total de points marqués lors des trois tirages. [...]
On suppose maintenant que le joueur effectue n tirages successifs, n étant un entier naturel supérieur ou égal à 2. Il remet toujours la boule obtenue dans l'urne après chaque tirage. On rappelle qu'on note G le nombre total de points marqués lors de ces n tirages.

1. Expliquer avec précision pourquoi l'instruction `X=grand(1,n,'bin',1,1/3)` permet de simuler n tirages dans l'urne.
2. Quelle instruction faut-il ajouter à la ligne 6 pour que le programme ci-dessous simule n tirages successifs dans l'urne (l'entier $n \geqslant 2$) étant donné par l'utilisateur) et qu'il affiche le nombre de points marqués par le joueur ? On justifiera la réponse.

```
1  n = input('n ?')
2  X = grand(1,n,'bin',1,1/3)
3  G = 0
4  for i = 2:n
5      if X(i)<>X(i-1) then
6          .....
7      end
8  end
9  disp(G)
```

Exercice 6 *extrait BCE BSB 2019*
Une cible circulaire de rayon 1 mètre est posée au sol. Deux joueurs 1 et 2 lancent chacun un palet en direction de la cible. On note U_1 et U_2 les variables aléatoires égales aux distances respectives en mètres entre le centre de la cible et les points d'impact des palets des joueurs 1et 2. On suppose que U_1 et U_2 suivent des lois uniformes sur $[0;1]$ et qu'elles sont indépendantes.
On note Z la variable aléatoire égale à la plus grande valeur entre U_1 et U_2.
Compléter le programme suivant afin qu'il simule l'expérience ci-dessus et qu'il affiche la valeur de Z.

```
U1=grand(1,1,'unf',0,1)
U2=grand(1,1,'unf',0,1)
if U1 >= U2 then
    ......
          else
    ......
end
disp(Z)
```