Département d'informatique
$2^{ème}$ année licence informatique
2021/2022
Dr. Chemseddine Chohra

Examen : Méthodes Numériques (Durée : 2h00)

Questions : (2 pts)

1. Quels sont les critères qui assurent la convergence de la méthode de Jacobi.
   - ☐ **Une matrice de coefficients (A) à diagonale strictement dominante.**
   - ☐ Une matrice d'itération (B) à diagonale strictement dominante.
   - ☐ Une matrice de coefficients (A) ayant une norme < 1.
   - ☐ **Une matrice d'itération (B) ayant une norme < 1.**
2. Parmi les méthodes suivantes, quelles sont les méthodes itératives de résolution de systèmes linéaires
   - ☐ L'élimination de Gauss.
   - ☐ **Méthode de Jacobi.**
   - ☐ **Méthode de Gauss-Seidel.**
   - ☐ Factorisation LU.

Exercice 1 : (6 pts)

Pour coder un nombre flottant, le format ***binary8*** utilise : 1 bit de **signe**, 4 bits pour l'**exposant** et 3 bits de **mantisse** (+1 bit de normalisation) avec un biais d'exposant **X = 7**.
Décoder les nombres suivants codés en format ***binary8*** (Donner le résultat en décimal).

- $A = (01101101)_{binary8}$.

  $s = 0$.
  $E = (1101)_2 = 13$.
  $e = E - 7 = 6$.
  $f = 101$.
  $A = (-1)^s * 1.f * 2^e = (-1)^0 * 1.101 * 2^6 = (1101000)_2 = 104$.

- $B = (01000001)_{binary8}$.

  $s = 0$.
  $E = (1000)_2 = 8$.
  $e = E - 7 = 1$.
  $f = 001$.
  $B = (-1)^s * 1.f * 2^e = (-1)^0 * 1.001 * 2^1 = (10.01)_2 = 2.25$.

- C = $(11001011)_{binary8}$.

  s = 1.
  E = $(1001)_2$ = 9.
  e = E – 7 = 2.
  f = 011.
  C = $(-1)^s * 1.f * 2^e = (-1)^1 * 1.011 * 2^2 = -(101.1)_2 = -5.5$.

Effectuer les opérations suivantes au format ***binary8*** (en arrondi au plus proche).

- A ⊕ C.

  $A \oplus C = round(A + C) = round(1.101 * 2^6 + (-1.011) * 2^2)$
  $= round(1.101 * 2^6 - 0.0001011 * 2^6) = round(1.1000101 * 2^6)$
  $= 1.100 * 2^6$

- B ⊖ C.

  $B \ominus C = round(B - C) = round(1.001 * 2^1 - (-1.011) * 2^2)$
  $= round(0.1001 * 2^2 + 1.011 * 2^2) = round(1.1111 * 2^2)$
  $= 10.000 * 2^2 = 1.000 * 2^3$.

  (Je considère « $1.111 * 2^2$ » aussi comme une bonne réponse)

- B ⊗ C.

  $B \otimes C = round(B * C) = round(1.001 * 2^1 * (-1.011) * 2^2)$
  $= -round(1.001 * 1.011 * 2^3) = -round(1.100011 * 2^3)$
  $= -1.100 * 2^3$.

Exercice 2 : (8 pts)

Soit le système linéaire suivant (tournez la page) :

$$\begin{cases} 2x_1 - x_2 & = 5 \\ -4x_1 + x_2 - 3x_3 & = -10 \\ 6x_1 + 9x_2 - 9x_3 & = -6 \end{cases}$$

1. Donner la matrice de coefficients ***A*** et le vecteur du second membre ***b*** correspondant au système précédent.

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -4 & 1 & -3 \\ 6 & 9 & -9 \end{pmatrix}, b = \begin{pmatrix} 5 \\ -10 \\ -6 \end{pmatrix}.$$

2. Donner la factorisation ***LU*** de la matrice ***A***.

Initialisation :

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, U = \begin{pmatrix} 2 & -1 & 0 \\ -4 & 1 & -3 \\ 6 & 9 & -9 \end{pmatrix}.$$

Elimination de $U_{2,1}$:

- $U_{2,1} / U_{1,1} = -4 / 2 = -2$.

- $L_{2,1}$ = -2.
- Multiplier la première ligne de U par 2 et la soustraire à la deuxième ligne.
- Résultat :

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, U = \begin{pmatrix} 2 & -1 & 0 \\ 0 & -1 & -3 \\ 6 & 9 & -9 \end{pmatrix}.$$

On fait la même chose pour $U_{3,1}$:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix}, U = \begin{pmatrix} 2 & -1 & 0 \\ 0 & -1 & -3 \\ 0 & 12 & -9 \end{pmatrix}.$$

Et puis pour $U_{3,2}$:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & -12 & 1 \end{pmatrix}, U = \begin{pmatrix} 2 & -1 & 0 \\ 0 & -1 & -3 \\ 0 & 0 & -45 \end{pmatrix}.$$

3. Utiliser la factorisation ***LU*** obtenue pour calculer la solution du système ***Ax* = *b***.

   Nous allons d'abord résoudre le système linéaire : ***Ly* = *b***

   - $y_1 = b_1 / L_{1,1} = 3 / 1 = 5$.
   - $y_2 = (b_2 - b_1 * L_{2,1}) / L_{2,2} = (-10 - (-2) * 5) / 1 = 0$.
   - $y_3 = (-6 - (3 * 5 + (-12) * 0)) / 1 = -21$.

$$y = \begin{pmatrix} 5 \\ 0 \\ -21 \end{pmatrix}.$$

   Puis, nous allons résoudre le système : **Ux = y**.

   - $x_3 = y_3 / U_{3,3} = -21 / (-45) = 7/15$.
   - $x_2 = (y_2 - x_3 * U_{2,3}) / U_{2,2} = (0 - (-3) * 7/15) / (-1) = -7/5$.
   - $x_1 = (5 - ((-1) * (-7/5) + (15/7) * 0)) / 2 = 9/5$.

$$x = \begin{pmatrix} 9/5 \\ -7/5 \\ 7/15 \end{pmatrix}.$$

4. Utiliser la factorisation ***LU*** obtenue pour calculer le déterminant de ***A***.

   Det(A) = det(L) * det(U) = det(U) = 2 * (-1) * (-45) = 90.

Exercice 3 : (4 pts)

La fonction Matlab suivante doit renvoyer **1** si la matrice (supposée carrée) passée comme paramètre est à diagonale strictement dominante et **0** sinon :

```
function dsd = diagonale_strictement_dominante(A)
    n = size(A);
    for i = 1:n
        dsd = 1;
```

```
        diag = A(i, i);
        for j = 1:i-1
            s = s + abs(A(i, j));
        end
        for j = i+1:n
            s = s + abs(A(i, j));
        end
        if diag < s
            dsd = 0;
        end
    end
end
```

- La fonction comporte quelques erreurs, trouver et corriger les erreurs pour que la fonction marche correctement.

Remarque : Il y a plusieurs façons de corriger ce code, tout changement juste du code avec les bons commentaires explicatifs est considéré comme une réponse juste.

```
function dsd = diagonale_strictement_dominante(A)
    n = size(A, 1);
    dsd = 1;
    for i = 1:n
        diag = abs(A(i, i));
        s = 0;
        for j = 1:i-1
            s = s + abs(A(i, j));
        end
        for j = i+1:n
            s = s + abs(A(i, j));
        end
        if diag < s
            dsd = 0;
        end
    end
end
```