Département d'informatique
$2^{ème}$ année licence informatique
2020/2021
Dr. Chemseddine Chohra

Examen : Méthodes Numériques (Durée : 2h00)

Questions : (2 pts)

Compléter le tableau suivant :

Format	Bits de signe	Bits d'exposant	Bits de mantisse	Biais d'exposant
binary16	1	5	10	15
binary32	**1**	**8**	**23**	**127**
binary64	**1**	**11**	**52**	**1023**

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).

- $A = (01101110)_{binary8}$.
  s = 0.
  $E = (1101)_2 = 13$.
  e = E – 7 = 6.
  f = 110.
  $A = (-1)^s * 1.f * 2^e = (-1)^0 * 1.110 * 2^6 = (1110000)_2 = 112$.
- $B = (10111101)_{binary8}$.
  s = 1.
  $E = (0111)_2 = 7$.
  e = E – 7 = 0.
  f = 101.
  $B = (-1)^1 * 1.101 * 2^0 = -(1.101)_2 = -1.625$.
- $C = (11001111)_{binary8}$.
  $C = (-1)^1 * 1.111 * 2^{9-7} = -(111.1)_2 = -7.5$.

Coder les nombres décimaux suivants au format ***binary8***.

- D = -0.47.
  0.47 * 2 = ***0***.94.
  0.94 * 2 = ***1***.88.
  0.88 * 2 = ***1***.76.
  0.76 * 2 = ***1***.52.
  0.52 * 2 = ***1***.04.
  0.04 * 2 = ***0***.08.
  $-0.47 = -(0.011110\ldots)_2 = -1.1110\ldots * 2^{-2}$.
  Arrondi vers : $-1.111 * 2^{-2}$ car il y a un 0 après le troisième chiffre.
  - Signe (-) => 1.
  - Mantisse => 111.
  - Exposant => -2 + 7 = 5 = 0101.

  $D = (10101111)_{binary8}$.
- E = -4.5.
  $-(4.5)_{10} = (100.1)_2 = 1.001 * 2^2$.
  - Signe (-) => 1.
  - Mantisse => 001.
  - Exposant => 2 + 7 = 9 = 1001.

E = $(11001001)_{binary8}$.

- F = 48.

$(48)_{10} = (110000)_2 = 1.100 * 2^5$.

- Signe (+) => 0.
- Mantisse => 100.
- Exposant => 5 + 7 = 12 = 1100.

F = $(01100100)_{binary8}$.

Exercice 2 : (8 pts)

Soit le système linéaire suivant :

$$\begin{cases} 4x_1 - 4x_2 + 2x_3 &= 2 \\ -8x_1 + 12x_2 - x_3 &= -3 \\ 12x_1 - 4x_2 + 15x_3 &= 5 \end{cases}$$

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

$$A = \begin{pmatrix} 4 & -4 & 2 \\ -8 & 12 & -1 \\ 12 & -4 & 15 \end{pmatrix}, b = \begin{pmatrix} 2 \\ -3 \\ 5 \end{pmatrix}.$$

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

Initialisation :

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, U = \begin{pmatrix} 4 & -4 & 2 \\ -8 & 12 & -1 \\ 12 & -4 & 15 \end{pmatrix}.$$

Elimination de $U_{2,1}$:

- $U_{2,1} / U_{1,1} = -8 / 4 = -2$.
- $L_{2,1} = -2$.
- Multiplier la première ligne de U par 2 et la soustraire à la deuxième ligne.
- Résultat :

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, U = \begin{pmatrix} 4 & -4 & 2 \\ 0 & 4 & 3 \\ 12 & -4 & 15 \end{pmatrix}.$$

On fait la même chose pour $U_{3,1}$:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix}, U = \begin{pmatrix} 4 & -4 & 2 \\ 0 & 4 & 3 \\ 0 & 8 & 9 \end{pmatrix}.$$

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

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

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

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

- $y_1 = b_1 / L_{1,1} = 2 / 1 = 2$.
- $y_2 = (b_2 – b_1 * L_{2,1}) / L_{2,2} = (-3 – 2 * (-2)) / 1 = 1$.
- $y_3 = (5 – (2 * 3 + 1 * 2)) / 1 = -3$.

$$y = \begin{pmatrix} 2 \\ 1 \\ -3 \end{pmatrix}.$$

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

- $x_3 = y_3 / U_{3,3} = -3 / 3 = -1$.
- $x_2 = (y_2 – x_3 * U_{2,3}) / U_{2,2} = (1 – (-1) * 3) / 4 = 1$.
- $x_1 = (2 – ((-1) * 2 + 1 * (-4))) / 4 = 8 / 4 = 2$.

$$x = \begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix}.$$

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

Det(A) = det(L) * det(U) = det(U) = 4 * 4 * 3 = 48.

Exercice 3 : (4 pts)

La fonction Matlab suivante doit calculer la solution d'un système triangulaire inférieur :

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.

Les instructions fausse avec les commentaires sont notés en rouge.
Les corrections sont notées en vert.

```
function x = resoudre_triangulaire_inferieure(A, b)
    n = size(A, 1);
    x = zeros(n, 1);
    x(1) = b(1) / A(1, 1);
    for i = 2:n
        % s = b(i); // Il faut initialiser S par 0 et non pas par b(i).
        s = 0; % Nouvelle instruction juste.
        % for j = 1:i // Il faut s'arrêter à i - 1.
        for j = 1:i-1 % Nouvelle instruction juste.
            % s = s + A(i, j) * x(i); // Il faut multiplier par x(j)
            s = s + A(i, j) * x(j); % Nouvelle instruction juste.
        end
        x(i) = (b(i) - s) / A(i, i);
    end
end
```

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