**POLYTECH LYON** — **Méthodes Numériques de Base**
**INFO 3A** — Année 2017-2018

## Ch 4 - Résolution de systèmes d'équations algébriques. Synthèse des éléments du Cours.

N. Debit

- Systèmes d'équations **linéaires**

$$\begin{cases} 9x_1 - 2x_2 + x_3 &= 13 \\ -x_1 + 5x_2 - x_3 &= 9 \\ x_1 - 2x_2 + 9x_3 &= -11 \end{cases}$$

- Systèmes d'équations **non-linéaires**

$$\begin{cases} x_1^2 - 10x_1 + x_2^2 &= -8 \\ x_1 x_2^2 + x_1 - 10x_2 &= -8 \end{cases}$$

Nous considérons d'abord les

## I)- Systèmes d'équations linéaires.

### 1 Introduction et formalisme

On cherche $\vec{x} = [x_1, x_2, x_3, \ldots, x_n]$ solution de

$$\begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \ldots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

i.e

$$A\vec{x} = \vec{b}$$

Dans la suite on notera $x$ pour $\vec{x}$ et $b$ pour $\vec{b}$.

**Remarque:** *Certaines difficultés à prendre en compte:*



- *En pratique $n$ est très grand (1000 à 300000);*
- *Le coût de calcul croît rapidement avec $n$;*
- *La place mémoire machine est un élément à prendre en compte.*

**Dans la présentation des algorithmes, on supposera que les matrices sont inversibles (non-singulières).**

**N.B.:** *L'algèbre linéaire élémentaire est un prérequis pour ce chapitre. Elle est supposée connue !*

$$A\,x = b \iff x = A^{-1}\,b.$$

**Le calcul de $A^{-1}$ est plus difficile et plus coûteux que la résolution du système linéaire de départ (voir comparaison de coût plus loin). En pratique, on n'inverse jamais $A$ !**

**Remarque:** *La méthode de substitution successive est*

- *théoriquement possible,*
- *( mais ) difficile à transcrire sous forme d'algorithme performant.*

I

**Certains systèmes sont faciles à résoudre:**

- **systèmes diagonaux (rares en pratique),**

- **systèmes triangulaires (inférieurs ou supérieurs) i.e systèmes à matrice triangulaire.**

**Définition:** *Une matrice triangulaire inférieure est de la forme*

$$\begin{bmatrix} a_{11} & & \\ \vdots & \ddots & \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$$

*Pour une matrice triangulaire supérieure, il suffit de transposer.*

Les systèmes triangulaires sont bien faciles à résoudre: Commencer par l'équation à la pointe du triangle et résoudre une à une les équations. On parle de **descente triangulaire** ou de **remontée triangulaire** selon que la matrice soit triangulaire inférieure ou triangulaire supérieure.

**Exemple** *1: Le système*

$$\begin{bmatrix} 3 & 0 & 0 \\ 1 & 2 & 0 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 7 \\ 14 \end{bmatrix}$$

*a pour solution* $x = [3, 2, 1]^T$.

II

La généralisation est immédiate,



**Théorème:**

- *Si la matrice $A$ est triangulaire inférieure, la solution du système $Ax = b$ est donnée par*

$$x_1 = \frac{b_1}{a_{11}}$$
$$x_i = \frac{(b_i - \sum_{k=1}^{i-1} a_{ik}x_k)}{a_{ii}} \qquad pour \quad i = 2, 3, \ldots, n.$$

- *Si la matrice $A$ est triangulaire supérieure, la solution du système $Ax = b$ est donnée par*

$$x_n = \frac{b_n}{a_{nn}}$$
$$x_i = \frac{(b_i - \sum_{k=i+1}^{n} a_{ik}x_k)}{a_{ii}} \qquad pour \quad i = n-1, n-2, \ldots, 1.$$

**Remarque:** *Les équations sont valides si les $a_{ii}$ sont tous non nuls ! Sinon la matrice $A$ n'est pas inversible et le système n'a pas de solution unique.*

**En effet, si la matrice $A$ est triangulaire son déterminant est égal au produit des termes diagonaux.**

$$det(A) = \prod_{i=1}^{n} a_{ii}.$$

**Idée: Ramener un système linéaire quelconque à 1 ou plusieurs systèmes triangulaires.**

**Définition:** *Une méthode de résolution d'un système linéaire est dite* **directe** *si la solution du système peut être obtenue par cette méthode en* **un nombre fini et prédéterminé d'opérations.** *Par opposition, une méthode* **itérative** *peut* **converger** *en peu ou beaucoup d'itérations,* **ou diverger** *selon le système.*

## 2 Méthodes directes.

Les 2 principales sont **l'élimination de Gauss** et **la décomposition LU**.

**Remarque:** *On peut voir la méthode de Gauss comme un cas particulier de décomposition LU.*

### 2.1 Méthode d'élimination de Gauss

**Idée:** Au moyen d'opérations élémentaires, éliminer tous les termes sous la diagonale de la matrice.

On introduit la matrice augmentée de dimension $(n+1)$

$$\left(\begin{array}{c|c} \mathrm{A} & \mathrm{b} \end{array}\right)$$

**les opérations sont effectuées sur la matrice augmentée $\Longrightarrow$ un système équivalent.**

**Exemple** *2:*

$$\left(\begin{array}{ccc|c} \mathbf{2} & 1 & 2 & 10 \\ 6 & 4 & 0 & 26 \\ 8 & 5 & 1 & 35 \end{array}\right) \quad \begin{array}{l} (l_2 \leftarrow l_2 - (6/\mathbf{2})\, l_1) \\ (l_3 \leftarrow l_3 - (8/\mathbf{2})\, l_1) \end{array}$$

**2** *est* **le pivot.**

$$\left(\begin{array}{ccc|c} 2 & 1 & 2 & 10 \\ 0 & \mathbf{1} & -6 & -4 \\ 0 & 1 & -7 & -5 \end{array}\right) \quad (l_3 \leftarrow l_3 - (1/\mathbf{1})\, l_2)$$

**1** *est* **le pivot.**

$$\left(\begin{array}{ccc|c} 2 & 1 & 2 & 10 \\ 0 & 1 & -6 & -4 \\ 0 & 0 & -1 & -1 \end{array}\right)$$



*Une remontée donne le résultat $x_3 = 1; x_2 = 2; x_1 = 3$.*

**Exemple** *3: Cas de pivot nul !*

$$\left(\begin{array}{cccc|c} \mathbf{1} & -1 & 2 & -1 & -8 \\ 2 & -2 & 3 & -3 & -20 \\ 1 & 1 & 1 & 0 & -2 \\ 1 & -1 & 4 & 3 & 4 \end{array}\right) \quad \begin{array}{l} (l_2 \leftarrow l_2 - (2/\mathbf{1})\, l_1) \\ (l_3 \leftarrow l_3 - (1/\mathbf{1})\, l_1) \\ (l_4 \leftarrow l_4 - (1/\mathbf{1})\, l_1) \end{array}$$

$a_{11} = \mathbf{1}$ *est* **pivot.**

$$\left(\begin{array}{cccc|c} 1 & -1 & 2 & -1 & -8 \\ 0 & \mathbf{0} & -1 & -1 & -4 \\ 0 & 2 & -1 & 1 & 6 \\ 0 & 0 & 2 & 4 & 12 \end{array}\right) \quad \begin{array}{l} \mathbf{a_{22} = 0} \text{ , } \textbf{Pivot nul !!} \\ (l_2 \leftrightarrow l_3) \Longleftrightarrow \mathbf{2}\ \textit{Pivot} \end{array}$$

$$\left(\begin{array}{cccc|c} 1 & -1 & 2 & -1 & -8 \\ 0 & 2 & -1 & 1 & 6 \\ 0 & 0 & \mathbf{-1} & -1 & -4 \\ 0 & 0 & 2 & 4 & 12 \end{array}\right) \quad (l_4 \leftarrow l_4 - (2/(\mathbf{-1}))\, l_3)$$

$$\left(\begin{array}{cccc|c} 1 & -1 & 2 & -1 & -8 \\ 0 & 2 & -1 & 1 & 6 \\ 0 & 0 & -1 & -1 & -4 \\ 0 & 0 & 0 & 2 & 4 \end{array}\right)$$

*Par une remontée triangulaire, on a $x = [-7, 3, 2, 2]^T$.*

**Remarque: Dans le cas de plusieurs seconds membres, $A$ étant fixée, la matrice augmentée s'écrit** *(On y reviendra plus tard).*

$$\left(\begin{array}{c|c} A & b_1 \ldots b_k \end{array}\right)$$

**Algorithme** .

*(***initialisation***)*:

- $A^{(1)} = A$, $b^{(1)} = b$

*(***Triangularisation***)*

- *Pour* $k = 1, \ldots, n-1$,

$$\begin{aligned}
a_{ij}^{(k+1)} &= a_{ij}^{(k)} && i = 1, \ldots, k;\ j = 1, \ldots, n \\
a_{ij}^{(k+1)} &= 0 && i = k+1, \ldots, n;\ j = 1, \ldots, k \\
a_{ij}^{(k+1)} &= a_{ij}^{(k)} - \frac{a_{ik}^{(k)} a_{kj}^{(k)}}{a_{kk}^{(k)}} && i = k+1, \ldots, n;\ j = k+1, \ldots, n \\
b_i^{(k+1)} &= b_i^{(k)} && i = 1, \ldots, k \\
b_i^{(k+1)} &= b_i^{(k)} - \frac{a_{ik}^{(k)} b_k^{(k)}}{a_{kk}^{(k)}} && i = k+1, \ldots, n
\end{aligned}$$

*(***Résolution du système triangulaire** $A^{(n)}x = b^{(n)}$*)*

- $x_n = \dfrac{b_n^{(n)}}{a_{(nn)}^{(n)}}$

- *Pour* $i = n-1, \ldots, 1$

$$x_i = \frac{b_i^{(n)} - \sum_{j=i+1}^{n} a_{ij}^{(n)} x_j}{a_{ii}^{(n)}}$$

**Proposition:** *L'élimination de Gauss revient à une factorisation $L \times U$ de la matrice seulement en l'absence de permutations de lignes !*



**Sur l'importance du pivotage - partiel ? global ?**

**Pivotage**

Technique de **pivotage partiel** :

Permute 2 lignes pour avoir le pivot maximum en valeur absolue.

Technique de **pivotage complet** :

Permute 2 lignes de la matrice augmentée, puis interchange 2 inconnues du système pour avoir le pivot maximum en valeur absolue.

**Raisons du pivotage**

Division par un très petit pivot

$$\frac{valeur + erreur}{Pivot} = \frac{valeur + erreur}{10^{-15}} = valeur 10^{15} + erreur 10^{15}$$

Exemple :

$$\left[\begin{array}{ccc|c} 0 & 4 & 3 & 1 \\ 4 & 1 & -32 & 3 \\ 6 & 3 & 7 & 2 \end{array}\right]$$

Pivotage partiel ( $l_1 \leftrightarrow l_3$)

$$\left[\begin{array}{ccc|c} 6 & 3 & 7 & 2 \\ 4 & 1 & -32 & 3 \\ 0 & 4 & 3 & 1 \end{array}\right]$$

Suffisant pour éviter les divisions par 0.
Peut aussi améliorer la précision des calculs.

## Pivotage Complet

Étape 1 : ( $l_1 \leftrightarrow l_2$)

$$\left[\begin{array}{ccc|c} 4 & 1 & -32 & 3 \\ 0 & 4 & 3 & 1 \\ 6 & 3 & 7 & 2 \end{array}\right]$$

Étape 2 : ( $c_1 \leftrightarrow c_3$)

$$\left[\begin{array}{ccc|c} -32 & 1 & 4 & 3 \\ 3 & 4 & 0 & 1 \\ 7 & 3 & 6 & 2 \end{array}\right]$$

**<u>Attention</u> :**

**Garder l'ordre des inconnues.**

O = ( 3, 2, 1)

$$\begin{bmatrix} 4 & 1 & -32 \\ 0 & 4 & 3 \\ 6 & 3 & 7 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 6 \end{bmatrix} x_1 + \begin{bmatrix} 1 \\ 4 \\ 3 \end{bmatrix} x_2 + \begin{bmatrix} -32 \\ 3 \\ 7 \end{bmatrix} x_3 = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} -32 \\ 3 \\ 7 \end{bmatrix} x_3 + \begin{bmatrix} 1 \\ 4 \\ 3 \end{bmatrix} x_2 + \begin{bmatrix} 4 \\ 0 \\ 6 \end{bmatrix} x_1 = \begin{bmatrix} -32 & 1 & 4 \\ 3 & 4 & 0 \\ 7 & 3 & 6 \end{bmatrix}\begin{bmatrix} x_3 \\ x_2 \\ x_1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$



## Autre exemple sur l'intérêt du Pivotage

Cet exemple montre que la précision peut être améliorée par le pivotage. On effectue un Arrondi à 4 chiffres à chaque opération.

Soit le système A x = b tel que :

$$A = \begin{bmatrix} -0.002 & 4.000 & 4.000 \\ -2.000 & 2.906 & -5.387 \\ 3.000 & -4.031 & -3.112 \end{bmatrix} \quad b = \begin{bmatrix} 7.998 \\ -4.481 \\ -4.143 \end{bmatrix}$$

La matrice triangulaire supérieure sans pivotage est : (augmentée)

$$\left[\begin{array}{ccc|c} -0.002 & 4.000 & 4.000 & 7.998 \\ 0 & -3.997 & -4.005 & -8.002 \\ 0 & 0 & -10.00 & 0.000 \end{array}\right]$$

ce qui donne la solution : $x^* = (-1496, 2.000, 0.000)$

Si on utilise le pivot partiel :

$$A = \begin{bmatrix} 3.000 & -4.031 & -3.112 \\ -0.002 & 4.000 & 4.000 \\ -2.000 & 2.906 & -5.387 \end{bmatrix} \quad b = \begin{bmatrix} -4.143 \\ 7.998 \\ -4.481 \end{bmatrix}$$

La matrice triangulaire supérieure avec pivotage est :

$$\left[\begin{array}{ccc|c} 3.000 & -4.031 & -3.112 & -4.143 \\ 0 & 3.997 & 3.998 & 7.995 \\ 0 & 0 & -7.681 & -7.681 \end{array}\right] \quad \text{ce qui donne} \quad x^* = \begin{bmatrix} 1.000 \\ 1.000 \\ 1.000 \end{bmatrix}$$

## En cas de plusieurs membres de droite

$$\begin{bmatrix} 3 & -1 & 2 \\ 1 & 2 & 3 \\ 2 & -2 & -1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 11 \\ 2 \end{bmatrix} \quad et \quad \begin{bmatrix} 3 & -1 & 2 \\ 1 & 2 & 3 \\ 2 & -2 & -1 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 14 \\ 14 \\ 2 \end{bmatrix}$$

On a en fait deux systèmes avec la même matrice.

On travaille alors avec la matrice augmentée de plusieurs membres de droite.

$$\left[\begin{array}{ccc|cc} 3 & -1 & 2 & 12 & 14 \\ 1 & 2 & 3 & 11 & 14 \\ 2 & -2 & -1 & 2 & 2 \end{array}\right]$$

Réduction de cette matrice

$$\left[\begin{array}{ccc|cc} 3 & -1 & 2 & 12 & 14 \\ 0 & 7/3 & 7/3 & 7 & 28/3 \\ 0 & 0 & -1 & -2 & -2 \end{array}\right]$$

On résout le système pour tous les membres de droite à la fois.

**Problème**

Si on veut résoudre le système pour un nouveau membre de droite plus tard.

On trouve :

$x_1 = 3, x_2 = 1, x_3 = 2$

et

$y_1 = 4, y_2 = 2, y_3 = 2$

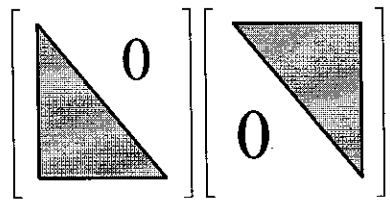


## 2.2 Méthode de factorisation triangulaire (décomposition LU)

Décomposer la matrice A en un produit de deux matrices triangulaire.

A x = b

A =




Étape 1. Décomposition

A . x = L . U . x = b

où L et U sont simples

Étape 2. Résolution

Poser y = U . x

1. Résoudre L . y = b par descente.

2. Résoudre U . x = y par remontée.

## Factorisation LU

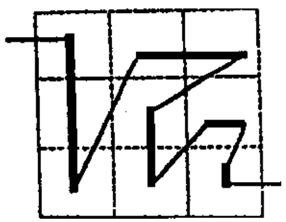
On désire factoriser une matrice 3 x 3
en posant que

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 & U_{12} & U_{13} \\ 0 & 1 & U_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Ce système a 9 inconnues et 9 équations.

Il faut calculer les coefficients $L_{ij}$ et $U_{ij}$ dans l'ordre suivant :

Ordre des calculs



d'abord une colonne puis une ligne

**NB :** $U_{ii} = 1$ pour unicité



1. Calcul de $L_{i1}$ et $U_{1j}$ :

$L_{11} = a_{11}$

$L_{21} = a_{21}$

$L_{31} = a_{31}$

$$U_{12} = \frac{a_{12}}{L_{11}}$$

$$U_{13} = \frac{a_{13}}{L_{11}}$$

2. Calcul de $L_{i2}$ et $U_{2j}$ :

$L_{22} = a_{22} - L_{21}\,U_{12}$
$L_{32} = a_{32} - L_{31}\,U_{12}$

$$U_{23} = \frac{a_{23} - L_{21}U_{13}}{L_{22}}$$

3. Calcul de $L_{33}$ :

$L_{33} = a_{33} - L_{31}\,U_{13} - L_{32}\,U_{23}$

**Formules générales de factorisation d'une matrice d'ordre N sous la forme L.U**

Pour les éléments de L :

$$L_{ij} = a_{ij} - \sum_{k=1}^{j-1} L_{ik} U_{kj} \qquad j \le i \qquad i = 1, 2, 3, \ldots, N$$

Pour les éléments de U :

$$U_{ij} = \frac{1}{L_{ii}} \left[ a_{ij} - \sum_{k=1}^{i-1} L_{ik} U_{kj} \right] \qquad i \le j \qquad j = 1, 2, 3, \ldots, N$$



**Stockage compact**

III

$$A = \begin{bmatrix} 2 & -1 & -1 \\ 0 & -4 & 2 \\ 6 & -3 & 0 \end{bmatrix}$$

$$A = L \cdot U = \begin{bmatrix} 2 & 0 & 0 \\ 0 & -4 & 0 \\ 6 & 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1 \end{bmatrix}$$

Décomposition stockée sous la forme :

$$A = L.U = \begin{bmatrix} 2 & -1/2 & -1/2 \\ 0 & -4 & -1/2 \\ 6 & 0 & 3 \end{bmatrix}$$

On met en mémoire **L+U-I (format compact)** et on peut détruire A

## Avantages

1. Une factorisation même s'il y a plusieurs seconds membres.

Factorisation = $O(n^3)$ flops. (> Gauss)
Résolution = $O(n^2)$ flops. (économie)

2. Peu de mémoire.

Stockage compact de LU dans A.

3. Préserve le profil de la matrice originale :
Si A matrice bande (creuse) , L et U le sont aussi

## Inconvénients

1. Pour une matrice symétrique, l'information de la symétrie n'est pas prise en compte.

2. Le pivotage se complique par rapport à la méthode de Gauss.



## Algorithme de la méthode de décomposition LU

### Décomposition de Crout
(sans permutation de lignes).

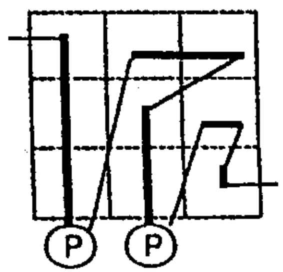
```
Faire pour i = 1 jusqu'à n , pas = 1 :
        L[i,1] = A[i,1] ;U[1,i] = A[1,i]/L[1,1];
fin faire pour.

Faire pour j = 2 jusqu'à n , pas = 1 :

   Faire pour i = j jusqu'à n , pas = 1 :
      Faire pour k = 1 jusqu'à j-1 , pas = 1 :
        Accumuler Somme de L[i,k] * U[k,j]
              en double précision;
      fin faire pour.
      L[i,j] = A[i,j] - Somme ;
   fin faire pour.

   U[j,j] = 1 ;

   Faire pour i = j + 1 jusqu'à n , pas = 1 :
      Faire pour k = 1 jusqu'à j-1 , pas = 1 :
          Accumuler Somme de L[j,k] * U[k,i]
                   en double précision;
      fin faire pour.
      U[j,i] = ( A[j,i] - Somme ) / L[j,j] ;
   fin faire pour.
fin faire pour.
```

**Factorisation triangulaire avec pivotage**

1. Utilise les opérations élémentaires de ligne sur la partie non augmentée de la matrice. Garder un vecteur des permutations.

2. Pivotage
Immédiatement
après calcul d'une colonne de L.

Ordre des calculs :




D'abord colonne, ensuite pivot puis ligne.

Remarque :

Le pivotage détruit la structure de la matrice originale.



**Calcul du déterminant après factorisation triangulaire avec pivotage:**

1. Multiplier les pivots

2. Faire un changement de signe à chaque fois qu'on permute deux lignes

3. Ne pas employer l'opération 1

de multiplication par une constante

**Ne pas oublier le dernier coefficient!!**

Exemple :

Calcul du déterminant de la matrice factorisée ci dessous.

**(-1)(3)(-1)(2)(-1/3)**
**= - 2**

Matrice originale

$$\begin{bmatrix} 0 & 2 & 1 \\ 1 & 0 & 0 \\ 3 & 0 & 1 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix}$$

Calcul de $L_{i1}$

$$\begin{bmatrix} 0 & 2 & 1 \\ 1 & 0 & 0 \\ 3 & 0 & 1 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix}$$

Premier pivot

$$\begin{bmatrix} 3 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 2 & 1 \end{bmatrix} \begin{matrix} 3 \\ 2 \\ 1 \end{matrix}$$

Calcul de $U_{1j}$

$$\begin{bmatrix} 3 & 0 & 1/3 \\ 1 & 0 & 0 \\ 0 & 2 & 1 \end{bmatrix} \begin{matrix} 3 \\ 2 \\ 1 \end{matrix}$$

Calcul de $L_{i2}$

$$\begin{bmatrix} 3 & 0 & 1/3 \\ 1 & 0 & 0 \\ 0 & 2 & 1 \end{bmatrix} \begin{matrix} 3 \\ 2 \\ 1 \end{matrix}$$

Second pivot

$$\begin{bmatrix} 3 & 0 & \frac{1}{3} \\ 0 & 2 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{matrix} 3 \\ 1 \\ 2 \end{matrix}$$

Calcul de $U_{2j}$

$$\begin{bmatrix} 3 & 0 & 1/3 \\ 0 & 2 & 1/2 \\ 1 & 0 & 0 \end{bmatrix} \begin{matrix} 3 \\ 1 \\ 2 \end{matrix}$$

Calcul de $L_{33}$

$$\begin{bmatrix} 3 & 0 & 1/3 \\ 0 & 2 & 1/2 \\ 1 & 0 & -1/3 \end{bmatrix} \begin{matrix} 3 \\ 1 \\ 2 \end{matrix}$$

**Inversion de matrices carrées**

L'inverse de A est notée par $A^{-1}$:

$$A^{-1} A = A A^{-1} = I$$

Calculer $A^{-1}$ => résoudre un système avec N membres de droites :

$$\left[\begin{array}{ccc|ccc} a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 1 \end{array}\right]$$

Exemple :

Calculer l'inverse de

$$\begin{bmatrix} 0 & 2 & 1 \\ 1 & 0 & 0 \\ 3 & 0 & 1 \end{bmatrix}$$

1. Par élimination de gauss

2. Par factorisation triangulaire.

Remarque :
Pour la factorisation, utiliser la propriété suivante :

$$(A.B)^{-1} = B^{-1} . A^{-1}$$

appliqué à la décomposition
$A = L . U$

on en déduit
$A^{-1} = U^{-1} . L^{-1}$

$U^{-1}$ et $L^{-1}$ se trouve rapidement par descente et remontée.



**Résolution de systèmes linéaires par l'inverse**

Si on veut résoudre
$A x = b$
on multiplie de chaque côté par l'inverse $A^{-1}$

$$A^{-1} A x = A^{-1} b$$

$$\Rightarrow x = A^{-1} b$$

Exemple : Résoudre par l'inverse le système

$$\left[\begin{array}{ccc|c} 0 & 2 & 1 & 5 \\ 1 & 0 & 0 & 1 \\ 3 & 0 & 1 & 4 \end{array}\right]$$

L'inverse de la matrice est

$$\begin{bmatrix} 0 & 1 & 0 \\ 1/2 & 3/2 & -1/2 \\ 0 & -3 & 1 \end{bmatrix}$$

La solution est donc :

$$\begin{bmatrix} 0 & 1 & 0 \\ 1/2 & 3/2 & -1/2 \\ 0 & -3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

Méthode avantageuse seulement si:

1. Il y a beaucoup de systèmes à résoudre avec la même matrice
$A x_i = b_i \quad i = 1,2,3...$

2. La matrice est pleine (elle ne comporte que peu de 0), car l'inversion détruit le profil.

**NB : Seules les matrices non singulières sont inversibles.**

## 3. Méthodes itératives

### 1. Jacobi :

Soit le système

$$\begin{cases} 8x_1 + x_2 - x_3 = 8 & (1) \\ 2x_1 + x_2 + 9x_3 = 12 & (2) \\ x_1 - 7x_2 + 2x_3 = -4 & (3) \end{cases}$$

Nous pouvons réarranger le système selon (1)-(3)-(2) :

$$\begin{cases} x_1 = 1.000 - 0.125x_2 + 0.125x_3 \\ x_2 = 0.571 + 0.143x_1 + 0.286x_3 \\ x_3 = 1.333 - 0.222x_1 - 0.111x_2 \end{cases}$$

A compléter avec les indices d'itération !

Estimations successives de la solution $x^{(n)} = (x_1^{(n)}, x_2^{(n)}, x_3^{(n)})$ :

| n | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $x_1$ | 0 | 1.000 | 1.095 | 0.995 | 0.993 | 1.002 | 1.001 | 1.000 |
| $x_2$ | 0 | 0.571 | 1.095 | 1.026 | 0.990 | 0.998 | 1.001 | 1.000 |
| $x_3$ | 0 | 1.333 | 1.048 | 0.969 | 1.000 | 1.004 | 1.001 | 1.000 |

C'est la même méthode que la méthode appliquée à une simple équation mais appliquée à un ensemble d'équations.
Nous pouvons réécrire le système sous la forme :

$$x^{(n+1)} = G\, x^{(n)} = b' - Tj\ x^{(n)}$$

**soit une méthode de Point fixe !**

où $x^{(n+1)}$ et $x^{(n)}$ sont les $n^{ième}$ et $(n+1)^{ième}$ itérés du vecteur x.
G est une transformation linéaire plutôt qu'une fonction non linéaire



Ecriture matricielle du système

$$A \cdot x = b \Leftrightarrow \begin{bmatrix} 8 & 1 & -1 \\ 1 & -7 & 2 \\ 2 & 1 & 9 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ -4 \\ 12 \end{bmatrix}$$

Posons A = L + D + U

$$L = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 2 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 8 & 0 & 0 \\ 0 & -7 & 0 \\ 0 & 0 & 9 \end{bmatrix}, U = \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A\,x = (L + D + U)\,x = b$$

$$\Leftrightarrow$$

$$D\,x = -(L + U)\,x + b$$

$$x = -D^{-1}(L + U)\,x + D^{-1}b$$

$$x^{n+1} = -D^{-1}(L + U)\,x^n + D^{-1}b$$

$$x^{n+1} = G(x^n)$$

Retour à l'Exemple :

$$b' = D^{-1}b = \begin{bmatrix} 1.000 \\ 0.571 \\ 1.333 \end{bmatrix}, \quad Tj = D^{-1}(L+U) = \begin{bmatrix} 0 & .124 & -.125 \\ -.143 & 0 & -.286 \\ .222 & .111 & 0 \end{bmatrix}$$

**Algorithme de la méthode de Jacobi**

Pour résoudre un système linéaire de N équations, réarranger les lignes pour que chaque élément de la diagonale soit le plus grand des éléments de sa colonne.
Soit A x = b ce réarrangement.

En partant d'une solution initiale $x^{(0)}$, chaque élément du vecteur $x^{(n+1)}$ est donné par :

$$x_i^{(n+1)} = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j\neq i}}^{N} \frac{a_{ij}}{a_{ii}} x_j^{(n)} \qquad n = 0,1,2,\ldots$$

Une condition suffisante de convergence est

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{N} |a_{ij}| \qquad i = 1,2,\ldots,N$$

Si cette condition est vérifiée, alors $x^{(n)}$ va converger vers la solution quelque soit le vecteur initial $x^{(0)}$.



**2.Gauss-Seidel :**

$$A\,x = (L + D + U)\,x = b$$

$$(L + D)\,x = -U\,x + b$$

Nous avons le processus itératif :

$$x^{(n+1)} = -D^{-1} L\, x^{(n+1)} - D^{-1} U\, x^{(n)} + D^{-1} b$$

Estimations successives de la solution :

| n | | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| $x_1$ | 0 | 1.000 | 1.041 | 0.997 | 1.001 | 1.000 |
| $x_2$ | 0 | 0.714 | 1.014 | 0.996 | 1.000 | 1.000 |
| $x_3$ | 0 | 1.032 | 0.990 | 1.002 | 1.000 | 1.000 |

Exemple :

$$\begin{cases} x_1^{(n+1)} = 1.000 - 0.125 x_2^{(n)} + 0.125 x_3^{(n)} \\ x_2^{(n+1)} = 0.571 + 0.143 x_1^{(n+1)} + 0.286 x_3^{(n)} \\ x_3^{(n+1)} = 1.333 - 0.222 x_1^{(n+1)} - 0.111 x_2^{(n+1)} \end{cases}$$

en commençant avec $x^{(0)} = (0,0,0)^T$.

## Algorithme de la méthode de Gauss-Seidel

Pour résoudre un système linéaire de N équations, réarranger les lignes pour que chaque élément de la diagonale soit le plus grand des éléments de sa colonne.
Soit A x = b ce réarrangement.

En partant d'une solution initiale $x^{(0)}$, chaque élément du vecteur $x^{(n+1)}$ est donné par :

$$x_i^{(n+1)} = \frac{b_i}{a_{ii}} - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(n+1)} - \sum_{j=i+1}^{N} \frac{a_{ij}}{a_{ii}} x_j^{(n)} \qquad n = 0,1,2,\ldots$$

Une condition suffisante de convergence est

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{N} |a_{ij}| \qquad i = 1,2,\ldots,N$$

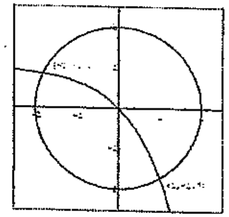
Si cette condition est vérifiée, alors $x^{(n)}$ va converger vers la solution quelque soit le vecteur initial $x^{(0)}$.



**II)- Systèmes d'équations NON linéaires.**

On ceherche à résoudre le système d'équations,

$$\begin{cases} x^2 + y^2 &= 4 \\ e^x + y &= 1 \end{cases}$$



# 1 Cas d'une équation - Recherche de racines ou zéro d'une équation algébrique non linéaire.

**Exemple du Calcul de Taux de rente**
On veut calculer le taux de rente moyen $i$ d'un fonds de placement sur plusieurs années. On a investi dans le fonds $v = 500$ € chaque année et on se retrouve après 5 ans avec un montant de $p = 3000$ €.
On sait que la relation qui lie $p, v, i$ et le nombre d'années $n$ est

$$p = v\sum_{k=1}^{n}(1+i)^k = v\frac{1+i}{i}[(1+i)^n - 1]$$

Ce problème est donc ramené à trouver $i$ tel que :

$$\Psi(i) = p - v\frac{1+i}{i}[(1+i)^n - 1] = 0.$$

équation non linéaire, dont on n'est pas capable de trouver une solution exacte.

## 1.1 Formalisme du problème :

Etant donnée une application quelconque $f : \mathbb{R} \longrightarrow \mathbb{R}$, déterminer une racine ou zéro $r$ de l'équation $f(x) = 0$, i.e par définition un certain $r \in \mathbb{R}$ tel que $f(r) = 0$.

**Nous considérerons au tableau la méthode intuitive de la bissection ou dichotomie et la méthode de Newton (-Raphson)qui est généralisée pour le cas de systèmes à la section suivante. Toutefois, les notes sur la méthode de point fixe et le thm de convergence sont conservées.**

2 Méthode de Newton pour un système d'équations non linéaires illustrée sur l'exemple de départ:

## Méthode de Newton à plusieurs variables

1. Ecrire sous la forme F(x) = 0

Prendre un réarrangement du système

$$\begin{cases} x^2 + y^2 = 4 \\ e^x + y = 1 \end{cases} \Leftrightarrow \begin{cases} x^2 + y^2 - 4 = 0 \\ e^x + y - 1 = 0 \end{cases}$$

2. Calcul de la matrice jacobienne de F notée ici F'

$$F' = \begin{bmatrix} \dfrac{\partial F_1}{\partial x} & \dfrac{\partial F_1}{\partial y} \\ \dfrac{\partial F_2}{\partial x} & \dfrac{\partial F_2}{\partial y} \end{bmatrix} = \begin{bmatrix} -2x & -2y \\ -e^x & -1 \end{bmatrix}$$

3. Algorithme : $x^{(0)}$ donné et itérations

$$x^{(n+1)} = x^{(n)} - \left[F'(x^{(n)})\right]^{-1} F(x^{(n)})$$

L'opération $\left[F'(x^{(n)})\right]^{-1} F(x^{(n)})$ consiste à résoudre un système linéaire 2x2 à chaque itération.

Avec $x^{(0)} = (-2, 1)^T$, nous avons :

$$F(x^{(0)}) = \begin{bmatrix} -1 \\ -0.1353 \end{bmatrix} \quad et \quad F'(x^{(0)}) = \begin{bmatrix} 4 & -2 \\ -0.1353 & -1 \end{bmatrix}$$

$$\left[F'(x^{(0)})\right]^{-1} F(x^{(0)}) = \begin{bmatrix} -0.1708 \\ 0.1584 \end{bmatrix} \quad et \quad x^{(1)} = \begin{bmatrix} -1.8292 \\ 0.8416 \end{bmatrix}$$