Algorithmique Numérique

saida.bouakaz@univ-lyon1.fr

1

Plan du Cours

- **Rappel sur les matrice**
  - ➢Définitions
  - ➢Opérations sur les matrices
  - ➢Déterminant & méthode de cramer
- **Résolution de système linéaire**
  - ➢Méthodes directes
    - Triangulation de Gauss
    - Décomposition LU
  - ➢Méthodes itératives
    - Méthode de Jacobi
    - Méthode de Seidel
- **Racines de fonctions F(x)=0**
  - ➢Introduction
  - ➢Méthode de Newton
  - ➢Méthode de la sécante
  - ➢Méthode de dichotomie

2

- **Interpolation**
  - ➢Interpolation linéaire et quadratique
  - ➢Formule de Lagrange, polynôme de Newton,
  - ➢Différences finis
  - ➢Splines
- **Approximation polynomiale**
  - ➢Méthode des moindres carrés, moindres carrés pondérées
  - ➢Polynômes de Chebychev
- **Intégration numérique**
  - ➢Introduction
  - ➢Méthode des trapèzes
  - ➢Méthode de Simpson
  - ➢Méthodes améliorées

3

Chapitre 2
Résolution de systèmes linéaires

- Méthode de Gauss: basée sur la triangulation
- Méthode de factorisation : LU
- Méthodes itératives

Méthode de Gauss

- Idée : méthode basée sur la triangulation
- Utilise une suite de combinaison linaires entre les différentes lignes, travaille sur la matrice élargie.
- AX=B $\longrightarrow$ $A^{(k)}$ $X=B^{(k)}$ avec $A^{(k)}$ triangulaire.
- Complexité
  - Complexité de la résolution du système triangulaire en $O(n^2)$:
  - Complexité de la triangulation en $O(n^3)$:

Méthode de Gauss

- **Procédé du pivot avec normalisation de la diagonale**

Le principe consiste à transformer le système *A X = B* en un système triangulaire équivalent

$$T \times X = C \equiv \begin{cases} x_1 + t_{1,2}x_2 + \cdots + t_{1,n}x_n & = & c_1 \\ x_2 + \cdots + t_{1,n}x_n & = & c_2 \\ & \vdots & \\ x_n & = & c_n \end{cases}$$

La solution se calcule par remontée.

- La transformation de *A* en *T* se compose de deux étapes itérées *n* fois.

A l'étape *i* :

- normalisation : on divise la ligne *i* par $a_{i;i}$ (le *pivot*)
  si $a_{i;i} \neq 0$ pour obtenir $a_{i;i} = 1$,
- annulation sous la diagonale : pour $i + 1 = k \rightarrow n$,
  on soustrait la ligne du pivot multipliée par $a_{k;i}$ à la ligne *k* pour obtenir $a_{k;i} = 0$

Méthode de Gauss

Procédé du pivot sans normalisation de la diagonale

On garde le principe de transformer le système A X = B en un système équivalent.
On travaille tjrs avec la matrice élargie.

On note par $m_{i1} = \frac{a_{i1}}{a_{11}}$ pour $1 \le i \le n$ d'où

$$
\begin{array}{ccccccc}
a_{11}x_1 & + & a_{12}x_2 & + \cdots + & a_{1n}x_n & = & b_1 \\
 & & (a_{22} - m_{21}a_{12})x_2 & + \cdots + & (a_{2n} - m_{21}a_{1n})x_n & = & b_2 - m_{21}b_1 \\
 & & \vdots & & \vdots & & \\
 & & (a_{i2} - m_{i1}a_{12})x_2 & + \cdots + & (a_{in} - m_{i1}a_{1n})x_n & = & b_i - m_{i1}b_1 \\
 & & \vdots & & \vdots & & \\
 & & (a_{n2} - m_{n1}a_{12})x_2 & + \cdots + & (a_{nn} - m_{n1}a_{1n})x_n & = & b_n - m_{n1}b_1
\end{array}
$$

7

A l'issue de la première transformation, la matrice du nouveau système est

$$
A^{(2)} = \begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
0 & a_{22} - m_{21}a_{12} & \cdots & a_{2n} - m_{21}a_{1n} \\
\vdots & \vdots & \vdots & \vdots \\
0 & a_{i2} - m_{i1}a_{12} & \cdots & a_{in} - m_{i1}a_{1n} \\
\vdots & \vdots & \vdots & \vdots \\
0 & a_{n2} - m_{n1}a_{12} & \cdots & a_{nn} - m_{n1}a_{1n}
\end{pmatrix}
$$

le second membre est

$$
b^{(2)} = \begin{pmatrix}
b_1 \\
b_2 - m_{21}b_1 \\
\vdots \\
b_i - m_{i1}b_1 \\
\vdots \\
b_n - m_{n1}b_1
\end{pmatrix}
$$

8

Le nouveau système s'écrit :

$$A^{(2)}X = b^{(2)}$$

À l'étape k, on a

$$
A^{(k)} = \begin{pmatrix}
a_{11}^{(k)} & a_{12}^{(k)} & \cdots & \cdots & & a_{1n}^{(k)} \\
0 & a_{22}^{(k)} & & & & a_{2n}^{(k)} \\
\vdots & \vdots & & & & \\
0 & 0 & a_{k-1k-1}^{(k)} & a_{k-1k}^{(k)} & \cdots & a_{kn}^{(k-1)} \\
\vdots & \vdots & 0 & \vdots & & \\
\vdots & \vdots & \vdots & \vdots & & \vdots \\
0 & 0 & 0 & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)}
\end{pmatrix}
$$

$$
b^{(2)} = \begin{pmatrix}
b_1 \\
b_2 - m_{21}b_1 \\
\vdots \\
b_i - m_{i1}b_1 \\
\vdots \\
b_n - m_{n1}b_1
\end{pmatrix}
$$

9

Cas générique : à l'étape k

Étape $(K\text{-}1)$: $A^{(K-1)} \; X = B^{(K-1)}$

$$A^{(k-1)} = \begin{bmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \cdots & a_{1k-1}^{(0)} & a_{1k}^{(0)} & \cdots & a_{1n}^{(0)} \\ 0 & a_{22}^{(1)} & \cdots & a_{2k-1}^{(1)} & a_{2k}^{(1)} & \cdots & a_{2n}^{(1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & a_{k-1}^{(k-2)} & a_{k-1\,k}^{(k-2)} & \cdots & a_{k-1\,n}^{(k-2)} \\ 0 & 0 & \cdots & 0 & a_{kk}^{(k-1)} & \cdots & a_{kn}^{(k-1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & 0 & a_{nk}^{(k-1)} & \cdots & a_{nn}^{(k-1)} \end{bmatrix} \qquad B^{(k-1)} = \begin{bmatrix} b_1^{(0)} \\ b_2^{(1)} \\ \vdots \\ b_{k-1}^{(k-2)} \\ b_k^{(k-1)} \\ \vdots \\ b_n^{(k-1)} \end{bmatrix}$$

Expression générale

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}$$

$$b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)}$$

avec $\begin{cases} i = k+1, \dots, n \\ j = k+1, \dots, n \end{cases}$

où : $m_{ik} = \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \qquad i = k+1, \dots, n$

Méthode de Gauss

Algorithme de triangulation sans normalisation de la diagonale

pour $k = 1$ à n

  si $a_{kk} = 0$ alors

    s'il existe $i > k$ tel que $a_{ik} \neq 0$

    alors échanger les lignes i et k

    sinon | la matrice est singulière

       stop

  {le pivot $a_{kk} \neq 0$ }

  pour $i = k+1$ à n, retrancher à la ligne i

    la nouvelle ligne k multipliée par a_{ik}/a_{kk}

     (colonnes de k à n

      ou même, seulement $k+1$ à n)

$x_n = a_{n,n+1}/a_{nn}$

pour $i = n-1$ à 1

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

ou : en rangeant les valeurs des solutions dans la $(n+1)^{ème}$ colonne de A :

$a_{n,n+1} = a_{n,n+1}/a_{nn}$

pour $i = n-1$ à 1

  $a_{i,n+1} = (a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij} * a_{j,n+1})/a_{ii}$

Méthode de Gauss

Algorithme de triangulation : Recherche du pivot maximal

On a intérêt à avoir des pivots les plus grands possibles, sinon ils peuvent devenir trop petits et pris égaux à 0 en machine. On choisira donc comme pivot les plus grand des a_{kl} en valeur absolue, et on échangera les lignes k et l

Complexité

Nombre d'opérations effectuées

nouvelle ligne k (ligne du pivot)

$k = 1 : n$ $\quad i = k : n+1$	1 division soit au total

$\sum_{k=1}^{n}(n-k+2) = \frac{n(n+3)}{2} \simeq \frac{n^2}{2}$ divisions

nouvelles lignes i

$k = 1 : n$ $\quad i = k+1 : n$ $\quad\quad j = n+1 : -1 : k$	1 multiplication et 1 addition soit au total

$\sum_{k=1}^{n}\sum_{i=k+1}^{n}(n-k+2) = \frac{n(n+1)(2n-1)}{6} \simeq \frac{n^3}{3}$

remontée

$i = n-1 : -1 : 1$ $\quad j = i+1 : n$	1 multiplication et 1 additions soit au total

$\sum_{i=1}^{n-1}(n-i) = \frac{n(n-1)}{2} \simeq \frac{n^2}{2}$

Au total, de l'ordre de $\frac{n^2}{2}$ divisions, $\frac{n^3}{3}$ multiplication additions, soit $\boxed{2\frac{n^3}{3}}$ opérations

Pivot de gauss : technique pratique pour inverser une matrice

Technique : elle s'appuie sur : A . A^{-1} = I

- la matrice A et la matrice identité I sont juxtaposée (on parle de matrice augmentée [A|I]
- On applique une série de transformation aux ligne de façon à obtenir une matrice identité à la place de A, la matrice situé à droite sera la matrice inverse → [A. A^{-1} | A^{-1} .I]
- La méthode du pivot de gauss permet d'obtenir cette matrice

Méthode de factorisation *LU* (ou LR)

- Méthode : basée sur une factorisation A
- Le principe de cette méthode de recherche de solution consiste à décomposer

la matrice A sous forme d'un produit A = L . U

A=L .U → (L . U) X= B

A=L .U → L .(U X)= B si on pose UX=Y

$$AX = B \Rightarrow \begin{cases} LY = B \\ UX = Y \end{cases}$$

16

Méthode de factorisation *LU* (ou LR)

Si on peut décomposer la matrice A en le produit de 2 matrices A=L .U (ou A= L.R)

- L : Triangulaire inférieure (L pour Lower triangular matrix)
- U : Triangulaire supérieure (U pour Upper triangular matrix)
- AX = B ⇔ (L.U)X = B ⇔ L. (UX) = B
- On pose UX = Y d'où LY = B

3 étapes :

1. Trouver les matrices L et U
2. Résolution du système LY = B (L triangulaire inférieure)
3. Résolution du système UX = Y (U triangulaire supérieure)

Remarque

LR : L pour Left triangular matrix et R pour Right triangular matrix

17

- L est une matrice triangulaire inférieure avec diagonale unité
- U est une matrice triangulaire supérieure.
- On utilisera la méthode LU lorsque l'on veut résoudre une famille de systèmes de la forme

  $A . X = B_i$

- où seul le vecteur B_i (les données) varie, le modèle (matrice A) reste la même. le calcul de L et R est totalement indépendant de B

18

Comment déterminer L et U et quelle est la complexité de la décomposition (en ?? opérations).

- Deux méthodes :
  - décomposition de Gauss
  - Algorithme de Crout (identification)

Représentation matricielle de l'élimination de Gauss

$$\text{AX= B} \Rightarrow \begin{cases} LY = B \\ UX = Y \end{cases}$$

Rappelle : à chaque étape de l'algorithme de gauss, A se transforme...

$$\text{pour} \quad i = k+1, \dots, n$$
$$\begin{cases} a_{ij}^{(k+1)} \leftarrow a_{ij}^{(k)} - \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}} a_{kj}^{(k)} \quad \text{pour } j = k+1, \dots, n \end{cases}$$

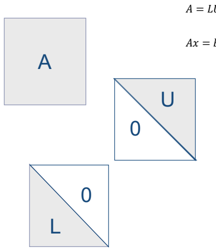
$$\text{notation matricielle } : A^{(k+1)} = M^{(k)} A^{(k)} ;$$

LU : principe

Il est si facile le résoudre un système « triangulaire » !

$A = LU$

$Ax = b \Leftrightarrow \begin{cases} (1) & Ly = b \\ (2) & Ux = y \end{cases}$



Comment construire Let U ?

idée :

reprendre l'étape de triangularisation de la méthode de Gauss

De Gauss à LU (ou LR)

Représentons une étape de la triangularisation par la multiplication de A par une matrice $M^{(k)}$

$$A^{(k+1)} = M^{(k)}A^{(k)} \qquad A^{(1)} = A \quad \text{et} \quad A^{(n)} = U$$

$$\boxed{\ell_{i,k} = \frac{a_{ik}}{a_{kk}} = -m_{i,k}}$$

$$\begin{cases} a_{ij} \leftarrow a_{ij} - \dfrac{a_{ik}}{a_{kk}} a_{kj} \\ b_i \leftarrow b_i - \dfrac{a_{ik}}{a_{kk}} b_k \end{cases}$$

$$M^{(k)} = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 1 & \ddots & 0 \\ \vdots & \ddots & -\ell_{k+1,k} & \ddots & \vdots \\ 0 & \cdots & -\ell_{n,k} & 0 & 1 \end{pmatrix}$$

$$U = M^{(n-1)} \ldots M^{(k)} \ldots M^{(1)} A = MA$$

$$A = M^{-1}U = LU$$

donc $L = M^{-1}$

22

LU : récapitulatif

Les matrices élémentaires $M^{(k)}$ sont **in**versibles et leurs inverses sont les matrices $L^{(k)}$ triangulaires inférieures telles que :

$$L^{(k)} = \begin{cases} l_{ii} = 1 & i = 1, n \\ l_{ik} = \ell_{ik} & i = k+1, n \\ l_{ij} = 0 & \text{sinon} \end{cases} \qquad L^{(k)} = I - \left(M^{(k)} - I\right)$$

$$M^{(k)} = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 1 & \ddots & 0 \\ \vdots & \ddots & -\ell_{k+1,k} & \ddots & \vdots \\ 0 & \cdots & -\ell_{n,k} & 0 & 1 \end{pmatrix} \qquad L^{(k)} = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 1 & \ddots & 0 \\ \vdots & \ddots & \ell_{k+1,k} & \ddots & \vdots \\ 0 & \cdots & \ell_{n,k} & 0 & 1 \end{pmatrix}$$

$$L = L^{(n-1)} \ldots L^{(k)} \ldots L^{(1)}$$

C'est la matrice l_{ik}

23

Exemple

$$A = \begin{pmatrix} 1 & 1 & -1 & 2 \\ -1 & 2 & 1 & 1 \\ 1 & 0 & 1 & -1 \\ 1 & -1 & 0 & 2 \end{pmatrix}$$

La décomposition de A=LU donne :

$$\begin{pmatrix} 1 & 1 & -1 & 2 \\ -1 & 2 & 1 & 1 \\ 1 & 0 & 1 & -1 \\ 1 & -1 & 0 & 2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & \frac{-1}{3} & 1 & 0 \\ 1 & \frac{-2}{3} & \frac{1}{2} & 1 \end{pmatrix}}_{L} \times \underbrace{\begin{pmatrix} 1 & 1 & -1 & 2 \\ 0 & 3 & 0 & 3 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & 3 \end{pmatrix}}_{U}$$

Détail de la décomposition

$$\begin{pmatrix} 1 & 1 & -1 & 2 \\ -1 & 2 & 1 & 1 \\ 1 & 0 & 1 & -1 \\ 1 & -1 & 0 & 2 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 1 & -1 & 2 \\ 0 & 3 & 0 & 3 \\ 0 & -1 & 2 & -3 \\ 0 & -2 & 1 & 0 \end{pmatrix}$$

Pivot =1
lig2 ←lig2 -(-1)lig1
lig3 ←lig3 -(1)lig1
lig4 ←lig4 -(1)lig1

Pivot =3
lig3 ←lig3 -(-1/3)lig2
lig4 ←lig4 -(-2/3)lig2

Etape 2

$$\begin{pmatrix} 1 & 1 & -1 & 2 \\ 0 & 3 & 0 & 3 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & 1 & 2 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 1 & -1 & 2 \\ 0 & 3 & 0 & 3 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

Pivot =2
lig4 ←lig4 -(1/2)lig3

24

24

L'algorithme de décomposition

Fonction L, U = décompose(A)

```
pour k = 1 jusqu'à n − 1
    pivot ← a_kk        (* stratégie de pivot *)
    si pivot ≠ 0 alors
        ℓ_kk ← 1
        pour i = k + 1 jusqu'à n
            ℓ_ik ← a_ik / pivot
            pour j = k + 1 jusqu'à n
                a_ij ← a_ij − ℓ_ik a_kj
            finpour
        finpour
    sinon "problème"
finpour
```


Calcul des matrice L et U (ou L et R) par identification : Algorithme de Crout

Pour calculer L et U, il suffit de remarquer que

$$\begin{pmatrix} 1 & 0 & 0 \\ l_{2,1} & 1 & 0 \\ l_{3,1} & l_{3,2} & 1 \end{pmatrix} \times \begin{pmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ 0 & u_{2,2} & u_{2,3} \\ 0 & 0 & u_{3,3} \end{pmatrix}$$

$$= \begin{pmatrix} \boxed{u_{1,1}} & \boxed{u_{1,2}} & \boxed{u_{1,3}} \\ \boxed{l_{2,1}} u_{1,1} & l_{2,1}u_{1,2} + \boxed{u_{2,2}} & l_{2,1}u_{1,3} + \boxed{u_{2,3}} \\ \boxed{l_{3,1}} u_{1,1} & l_{3,1}u_{1,2} + \boxed{l_{3,2}} u_{2,2} & l_{3,1}u_{1,3} + l_{3,2}u_{2,3} + \boxed{u_{3,3}} \end{pmatrix}$$

En prenant les équations obtenues dans le bons ordres (les colonnes de gauche à droite et les lignes de haut en bas) on remarque que l'on obtient un système à résoudre où à chaque étape, il n'y a qu'une seule inconnue.

$U_{11}=a_{11}$; $u_{12}=a_{12}$; $u_{13}=a_{13}$; $l_{21}=a_{21}/u_{11}$; $l_{31}= a_{31}/u_{11}$

$u_{22}=a_{22} - l_{21} u_{12}$; $l_{32}= (a_{31}-l_{31} u_{12}) /u_{22}$; $u_{33} = a_{33}-l_{31}u_{13} -l_{32}u_{23}$

Algorithme de Crout

```
pour j de 1 à n faire
   pour i de 1 à j faire          // Calcul des u_{i,j}
     u_{i,j} ← a_{i,j}
      pour k de 1 à i − 1 faire
        u_{i,j} ← u_{i,j} − l_{i,k} r_{k,j}
      fin pour
   fin pour
  pour i de j + 1 à n faire       // Calcul des l_{i,j}
     l_{i,j} ← a_{i,j}
      pour k de 1 à j − 1 faire
        l_{i,j} ← l_{i,j} − l_{i,k} k,j
      fin pour
         l_{i,j} ← l_{i,j}/u_{j,j}
    fin pour
fin pour
```

Méthodes itératives

- L'idée construire une suite de vecteurs qui converge vers le vecteur ($X^{(k)}$), solution du système ***A . X = B***
- Principe du calcul d'un point fixe : limite de la suite construite.
- Procédé → transformer ***A . X = B*** ⟷ en une égalité

$$A\,.\,X = B \quad \Leftrightarrow \quad X = \varphi(X)=MX+N$$

- On est alors ramené à un problème de recherche de point fixe :

$$X^* = \varphi(X^*)$$

On définie la suite récurrente par :

- $X^{(0)}$(vecteur initial fixé).
- la règle de récurrence pour $(X^{(k+1)})_{k \in N}$:

$$X^{(k+1)} = \varphi(X^{(k)}) = MX^{(k)} + N$$: un système linéaire

- Si la suite converge (k vers + ∞), alors sa limite est solution du système

Si on écrit A sous la forme A= - E + D - F (*une somme de matrices*)

$$AX = B \Rightarrow (-E+D-F)X = B$$
$$DX = B + EX + FX$$
$$X = D^{-1}(B + EX + FX)$$

On choisit D pour qu'elle soit facilement inversible

$$A = \underbrace{\begin{pmatrix} 0 & \cdots & 0 \\ a_{21} & \ddots & \vdots \\ \vdots & & \\ a_{n1} & \cdots\ a_{n(n-1)} & 0 \end{pmatrix}}_{(-E)} + \underbrace{\begin{pmatrix} a_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_{nn} \end{pmatrix}}_{(D)} + \underbrace{\begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & & & \vdots \\ \vdots & & \ddots & a_{n-1n} \\ 0 & & \cdots & 0 \end{pmatrix}}_{(-F)}$$

On constate que la matrice D^{-1} est facile à calculer

$$D^{-1} = \left(\frac{1}{a_{ii}}\right)_{i=1\cdots n} \quad \text{où } a_{ii} \neq 0$$

Sous cette forme $AX = (-E + \boldsymbol{D} - F)X$

Les méthodes Jacobi, Gauss-Seidel se distinguent dans la façon de répartir : $\boldsymbol{D}, -\boldsymbol{E}$ et $-\boldsymbol{F}$

Méthode de Jacobi

On pose : $M = D^{-1}(+E+F)$ et $N = D^{-1}B$

$$AX = b \Rightarrow X = D^{-1}(B + EX + FX)$$

$$\begin{cases} X^{(0)} : \text{(vecteur initial fixé)} \\ X^{(k+1)} = D^{-1}(B + EX^{(k)} + FX^{(k)}) \end{cases}$$

En écrivant le système sous forme d'équations on a :

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

A l'étape 1 on a :

$$x_1^1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2^0 - \cdots - a_{1n}x_n^0)$$

$$x_2^1 = \frac{1}{a_{22}}(b_2 - a_{21}x_1^0 - a_{23}x_3^0 - \cdots - a_{2n}x_n^0)$$

$$\vdots$$

$$x_n^1 = \frac{1}{a_{nn}}(b_n - a_{n1}x_1^0 - a_{n2}x_2^0 - \cdots - a_{nn-1}x_{n-1}^0)$$

Méthode de Gauss-Seidel

A partir de $A = D - E - F$ on répartit $D; E; F$

$$AX = B \Rightarrow X = (D-E)^{-1}FX + (D-E)^{-1}B$$

Le calcul effectif peut se faire par un calcul matriciel
En calculant : $(D-E)^{-1}$:
$M = (D-E)^{-1}F$ et $N = (D-E)^{-1}B$

$$\begin{cases} X^{(0)} : \text{(vecteur initial fixé)} \\ X^{(k+1)} = MX^{(k)} + N \end{cases}$$

Ce calcul suppose le calcul de $(D-E)^{-1}$

Le calcul effectif se fait de la façon suivante

$$\begin{cases} X^{(0)} : \text{(vecteur initial fixé)} \\ X^{(k+1)} = (D-E)^{-1}\left(B + FX^{(k)}\right) = MX^{(k)} + N \\ Avec : \ M = (D-E)^{-1}F \quad ; \quad N = (D-E)^{-1}B \end{cases}$$

En général on passe par la formulation sous forme d'équations (plus simple à calculer) C'est cette méthode qu'on adoptera ici
Soit :

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

En écrivant le système sous forme d'équations on a :

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

A l'étape 1 on a :

$$x_1^{\boxed{1}} = \frac{1}{a_{11}}(b_1 - a_{12}x_2^{\boxed{0}} - \cdots - a_{1n}x_n^{\boxed{0}})$$

$$x_2^{\boxed{1}} = \frac{1}{a_{22}}(b_2 - a_{21}x_1^{\boxed{1}} - a_{23}x_3^{\boxed{0}} - \cdots - a_{2n}x_n^{\boxed{0}})$$

$$x_n^1 = \frac{1}{a_{nn}}(b_n - a_{n1}x_1^{\boxed{1}} - a_{n2}x_2^{\boxed{1}} - \cdots - a_{nn-1}x_{n-1}^{\boxed{1}})$$

Condition de convergence

- Une matrice A est dite à diagonale dominante si

$$\forall i, |a_{i,i}| > \sum_{j \neq i} |a_{i,j}|.$$

- **Théorème (CS)** : Les méthodes de Jacobi et
- Gauss-Seidel s'appliquent sur (A.X=B) et convergent si A est à diagonale dominante.
- Soit $\rho(M) = \sup\{ |\lambda i| \}$ où les λi sont les valeurs propres de la matrice
- $\rho(M)$ est appelé rayon spectral de M
- **Théorème (CNS)** : si $P = M^{-1} \times N$ est diagonalisable, alors pour tout $X^{(0)}$, la suite $(X^{(k)})$ converge ssi $\rho(M) < 1$.

Conditions d'arrêt

- Condition d'arrêt
  en général :
  $\frac{\|AX^{(k)} - B\|}{\|B\|} < \varepsilon$

  ou bien :
  $\left\|X^{(k+1)} - X^{(k)}\right\| < \varepsilon$

Complexité

- Chaque itération nécessite n(2n - 1) opérations, et plus précisément :
  - n divisions
  - n(n - 1) soustractions
  - n(n - 1) multiplications
- Remarque1: plus on fait d'itérations, plus le résultat est précis.
- Remarque 2 : Ces méthodes sont particulièrement intéressantes lorsqu'il s'agit de très grandes matrices (n > 100) et on se contente dans ce cas d'une dizaine d'itérations.

Exemple : méthode Gauss-Seidel : passage par inversion de $(D-E)^{-1}$

$$A=\begin{pmatrix}2 & 1 & 1\\ -1 & 1 & 2\\ 1 & 0 & 2\end{pmatrix};\ B=\begin{pmatrix}6\\3\\2\end{pmatrix}\ \text{ on suppose } X^{(0)}=\begin{pmatrix}0\\0\\0\end{pmatrix}$$

1ère méthode de résolution : calcul de $(D-E)^{-1}$

On part de : $(-E\ +\ D-F)X=B \Rightarrow (D-E)X=B+FX$

$X=(D-E)^{-1}(B+FX)$

On construit la suite récurrente $X^{(k)}$ comme suit

$$\begin{cases}X^{(0)}\ Valeur\ initial\\ X^{(k+1)}=(D-E)^{-1}(B+FX^{(k)}\\ \ Condition\ d'arrêt\end{cases}$$

méthode Gauss-Seidel : passage par inversion de $(D-E)^{-1}$

$$(\mathrm{D}-\mathrm{E})^{-1}=\begin{pmatrix}1/2 & 0 & 0\\ 1/2 & 1 & 0\\ -1/4 & 0 & 1/2\end{pmatrix};$$

$$(\mathrm{D}-\mathrm{E})^{-1}\mathrm{F}=\begin{pmatrix}1/2 & 0 & 0\\ 1/2 & 1 & 0\\ -1/4 & 0 & 1/2\end{pmatrix}\begin{pmatrix}0 & -1 & -1\\ 0 & 0 & -2\\ 0 & 0 & 0\end{pmatrix}$$

$$(\mathrm{D}-\mathrm{E})^{-1}\mathrm{F}=\begin{pmatrix}0 & -1/2 & -1/2\\ 0 & -1/2 & -5/2\\ 0 & 1/4 & 1/4\end{pmatrix};\ (\mathrm{D}-\mathrm{E})^{-1}\mathrm{B}=\begin{pmatrix}3\\6\\-1/2\end{pmatrix};$$

$$X^0=\begin{pmatrix}0\\0\\0\end{pmatrix}$$

méthode Gauss-Seidel : passage par inversion de $(D-E)^{-1}$ – suite

$$X^{(1)} = \begin{pmatrix} 0 & -0.5 & -0.5 \\ 0 & -0.5 & -2.5 \\ 0 & 0.25 & 0.25 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 3 \\ 6 \\ -0.5 \end{pmatrix} = \begin{pmatrix} 3 \\ 6 \\ -0.5 \end{pmatrix}$$

$$X^{(2)} = \begin{pmatrix} 0 & -0.5 & -0.5 \\ 0 & -0.5 & -2.5 \\ 0 & 0.25 & 0.25 \end{pmatrix} \times \begin{pmatrix} 3 \\ 6 \\ -0.5 \end{pmatrix} + \begin{pmatrix} 3 \\ 6 \\ -0.5 \end{pmatrix} = \begin{pmatrix} 0.25 \\ 4.25 \\ 0.875 \end{pmatrix}$$

$$X^{(3)} = \begin{pmatrix} 0 & -0.5 & -0.5 \\ 0 & -0.5 & -2.5 \\ 0 & 0.25 & 0.25 \end{pmatrix} \times \begin{pmatrix} 0.25 \\ 4.25 \\ 0.875 \end{pmatrix} + \begin{pmatrix} 3 \\ 6 \\ -0.5 \end{pmatrix} = \begin{pmatrix} 0.4375 \\ 1.6875 \\ 0.7813 \end{pmatrix}$$

$$X^{(4)} = \begin{pmatrix} 0 & -0.5 & -0.5 \\ 0 & -0.5 & -2.5 \\ 0 & 0.25 & 0.25 \end{pmatrix} \times \begin{pmatrix} 0.4375 \\ 1.6875 \\ 0.7813 \end{pmatrix} + \begin{pmatrix} 3 \\ 6 \\ -0.5 \end{pmatrix} = \begin{pmatrix} 1.7656 \\ 3.203 \\ 0.1172 \end{pmatrix}$$

Résolution par expressions équationnelles

Le calcul effectif se fait de la façon suivante

$$\begin{cases} X^{(0)} : \text{(vecteur initial fixé)} \\ (D-E)X^{(k+1)} = \left(B + FX^{(k)}\right) \Rightarrow DX^{(k+1)} = B + EX^{(k+1)} + FX^{(k)} \end{cases}$$

Soit :

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

$A = \begin{pmatrix} 2 & 1 & 1 \\ -1 & 1 & 2 \\ 1 & 0 & 2 \end{pmatrix}$; B $= \begin{pmatrix} 6 \\ 3 \\ 2 \end{pmatrix}$ on a $X^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

Résolution par expressions explicite (analytique)

$$x_1^{(k+1)} = \frac{1}{2}\left(6 - x_2^{(k)} - 0x_3^{(k)}\right)$$
$$x_2^{(k+1)} = \frac{1}{2}\left(3 + x_1^{(k+1)} - 2x_3^{(k)}\right)$$
$$x_3^{(k+1)} = \frac{1}{2}\left(2 - x_1^{(k+1)} - 0x_2^{(k+1)}\right)$$

1ère itération

$$x_1^{(1)} = \frac{1}{2}\left(6 - x_2^{(0)} - 0x_3^{(0)}\right) = \frac{1}{2}(6 - 0 - 0) = 3$$
$$x_2^{(1)} = \frac{1}{1}\left(3 + x_1^{(1)} - 2x_3^{(0)}\right) = (3 + 3 - 2 \times 0) = 6$$
$$x_3^{(1)} = \frac{1}{2}\left(2 - x_1^{(1)} - 0x_2^{(1)}\right) = \frac{1}{2}(2 - 3 - 0 \times 0) = -\frac{1}{2}$$

2ème itération

$$x_1^{(2)} = \frac{1}{2}\left(6 - x_2^{(1)} - 0x_3^{(1)}\right) = \frac{1}{2}\left(6 - 6 - (-\frac{1}{2})\right) = \frac{1}{4} = 0.25$$
$$x_2^{(2)} = \frac{1}{1}\left(3 + x_1^{(2)} - 2x_3^{(1)}\right) = \left(3 + \frac{1}{4} - 2 \times (-\frac{1}{2})\right) = \frac{17}{4} = 4.25$$
$$x_3^{(3)} = \frac{1}{2}\left(2 - x_1^{(2)} - 0x_2^{(2)}\right) = \frac{1}{2}\left(2 - \frac{1}{4} - 0 \times \frac{17}{4}\right) = \frac{7}{8} = 0.875$$

Résolution par expressions équationnelles - suite

3ère itération

$$x_1^{(3)} = \frac{1}{2}\left(6 - x_2^{(2)} - 0x_3^{(2)}\right) = \frac{1}{2}\left(6 - \frac{17}{4} - \frac{7}{8}\right) = \frac{7}{16} = 0.4375$$
$$x_2^{(3)} = \frac{1}{1}\left(3 + x_1^{(3)} - 2x_3^{(2)}\right) = \left(3 + \frac{7}{16} - 2 \times \frac{7}{8}\right) = \frac{27}{16} = 1.6875$$
$$x_3^{(3)} = \frac{1}{2}\left(2 - x_1^{(3)} - 0x_2^{(3)}\right) = \frac{1}{2}\left(2 - \frac{7}{16}\right) = -\frac{25}{32} = 0.7813$$

4ème itération

$$x_1^{(4)} = \frac{1}{2}\left(6 - x_2^{(3)} - 0x_3^{(3)}\right) = \frac{1}{2}\left(6 - \frac{27}{16} - \frac{25}{32}\right) = \frac{113}{64} = 1.7656$$
$$x_2^{(4)} = \frac{1}{1}\left(3 + x_1^{(4)} - 2x_3^{(3)}\right) = \left(3 + \frac{113}{64} - 2 \times (-\frac{25}{32})\right) = \frac{205}{64} = 3.2031$$
$$x_3^{(4)} = \frac{1}{2}\left(2 - x_1^{(4)} - 0x_2^{(4)}\right) = \frac{1}{2}\left(2 - \frac{113}{64} - 0 \times \frac{205}{64}\right) = \frac{15}{128} = 0.1172$$

Retour aux méthodes : méthode de Jordan

- Méthode : basée sur une diagonalisation
- Utilise une suite de combinaison linaires entre les différentes lignes, travaille sur la matrice élargie (voir méthode de gauss)
- Utilisation particulière de Gauss
- AX=B ⟶ L X=$B^{(k)}$ avec L matrice diagonale.
- Complexité (globalement la même que Gauss)
- Complexité de la résolution du système triangulaire en ?:
- Complexité de la triangulation en ?