

# Analyse Numérique

## Méthodes directes de résolution de systèmes linéaires

Laurence Lamoulié

EISTI

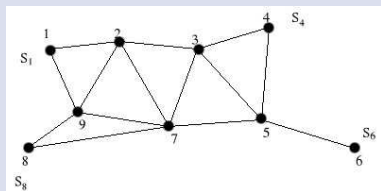
Ecole Internationale des Sciences du Traitement de  
l'Information



## Soyons clairs...

- Plus de résolution à la main car trop d'inconnues, et résultat trop long à obtenir
- Les méthodes connues ne sont plus utilisables
- Résolution d'un système  $\neq$  obtention de la solution exacte en un nombre fini d'étapes.
- Compromis: obtenir une solution approchée en un nombre fini d'étapes, sachant que la solution exacte serait obtenue en un nombre infini d'étapes.  
Ce compromis est le principe même des méthodes itératives.

## Modèle de réseau



Dans chaque arête circule un fluide; à chaque noeud est associé un potentiel. L'intensité (ou le débit) du fluide est proportionnelle à la différence de potentiel entre les deux extrémités de l'arête où il circule; c'est la loi d'Ohm pour les circuits électriques :

$$q_{i,j} = k_{i,j}(u_i - u_j)$$

Une loi physique de conservation (de Kirchoff dans le cas électrique) impose un équilibre : la somme algébrique des intensités en chaque noeud est égale à la valeur de la source (ou du puits) qu'il figure.

Au noeud  $P_i$ , on a dans le cas du circuit électrique :

$$S_i = \sum_j q_{i,j} = \sum_j k_{i,j}(u_i - u_j)$$

Cette somme peut être étendue aux noeuds adjacents de  $P_i$ , les équations d'équilibre s'écrivent :

$$\left\{ \begin{array}{l} S_1 = k_{1,2}(u_1 - u_2) + k_{1,9}(u_1 - u_9) \\ 0 = k_{2,1}(u_2 - u_1) + k_{2,9}(u_2 - u_9) + k_{2,7}(u_2 - u_7) + k_{2,3}(u_2 - u_3) \\ \quad \quad \quad \dots = \dots \\ 0 = k_{9,1}(u_9 - u_1) + k_{9,2}(u_9 - u_2) + k_{9,7}(u_9 - u_7) + k_{9,8}(u_9 - u_8) \end{array} \right.$$

De sorte que l'équilibre du système est connu en résolvant le système linéaire

$$Au = S$$

avec une matrice  $A$  dont les coefficients non nuls sont représentés ci-dessous par une étoile :

$$A = \begin{pmatrix} * & * & & & & & & & * \\ * & * & * & & & & * & & * \\ & & * & * & * & & * & & \\ & & & * & * & * & & & \\ & & & & * & * & * & * & \\ & & & & & * & * & & \\ & & * & * & & * & * & * & \\ & & & & & & * & * & * \\ * & * & & & & & * & * & * \end{pmatrix}$$

Le second membre est défini par  $S^T = (S_1, 0, 0, S_4, 0, S_6, 0, S_8, 0)$ .

# Première idée



## Méthode des déterminants ?

Les formules de Cramer, dites aussi "méthode des déterminants" fournissent une solution exacte du système linéaire  $A\mathbf{x} = b$  dans  $\mathbb{R}^n$  avec  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$  donnée par :

$$x_i = \frac{\det A^{(i)}}{\det A}$$

où  $A^{(i)}$  désigne la matrice obtenue en substituant dans  $A$  la colonne  $i$  par le second membre du système.

# Première idée



## Combien ça coûte ?

Il y a donc  $N + 1$  déterminants à calculer.

Le coût du calcul d'un tel déterminant est :

$$c(n) \sim n \times n! \text{ opérations élémentaires}$$

et le coût global de la méthode est donc

$$C(n) \sim n \times (n - 1)! \text{ opérations}$$

Le coût de la résolution d'un système  $100 \times 100$  peut alors être évalué par la formule de Stirling ( $n! \sim n^{n+1/2} e^{-n} \sqrt{2\pi}$ ).

Cela conduit à l'évaluation :

$$C(n) \sim 9.4 \times 10^{161}$$

# Se ramener à plus simple



## Système diagonal

Si  $A$  est une matrice diagonale, c'est à dire si

$$A = (a_{i,j})_{1 \leq i,j \leq n} \text{ avec } a_{i,j} = 0 \text{ si } i \neq j$$

le système  $Ax = b$  est immédiatement résolu du fait que

$$x_i = \frac{1}{a_{ii}} b_i$$

L'algorithme correspondant est donné par :

Pour  $i$  de 1 à  $n$

$$x_i \leftarrow b_i / a_{ii}$$

FinPour  $i$



# Pas aussi simple



## Système triangulaire

La matrice  $A$  d'un système triangulaire supérieur est telle que

$$A = (a_{i,j})_{1 \leq i,j \leq n} \text{ avec } a_{i,j} = 0 \text{ si } i > j$$

Comme  $A$  est inversible,

$$a_{i,i} \neq 0 \text{ si } 1 \leq i \leq n$$

Le système s'écrit

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$$\vdots$$

$$a_{nn}x_n = b_n$$

# Pas aussi simple



## Algorithme

On obtient alors la solution par un algorithme de *substitution rétrograde*, dit aussi simplement *algorithme de remontée* :

$$x_n \leftarrow b_n / a_{n,n}$$

Pour  $i$  de  $n-1$  à  $1$  par pas de  $-1$

$$x_i \leftarrow \left( b_i - \sum_{j=i+1}^n a_{i,j} x_j \right) / a_{i,i}$$

FinPour  $i$

Le coût du calcul d'un  $x_k$  se décompose en :

- $(n - k)$  additions,
- $(n - k)$  multiplications,
- 1 division

# Pas aussi simple



Le coût total de remontée est donc de

$$\sum_{k=1}^n (n - k) = n^2 - \frac{n(n-1)}{2} \sim \frac{n^2}{2} \text{ additions + multiplications}$$

soit  $n^2$  opérations élémentaires. **Conseil** : Entraînez-vous à retrouver cet algorithme et à calculer sa complexité...

# Se ramener deux systèmes triangulaires



## Principe

Remplacer  $A$  par le produit de deux matrices triangulaires, notées en général  $L$  et  $U$ , respectivement triangulaire inférieure et triangulaire supérieure. En effet on résout alors successivement deux systèmes triangulaires :

$$Ly = b \text{ puis } Ux = y$$

dont la solution  $x$  vérifie évidemment  $Ax = b$ .

En schématisant :

$$\begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} = \begin{pmatrix} * & 0 & 0 \\ * & * & 0 \\ * & * & * \end{pmatrix} \begin{pmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \end{pmatrix}$$

# Etape 1



## Elimination de Gauss sans recherche de pivot

Sous l'hypothèse que  $a_{11} \neq 0$  on peut l'utiliser pour éliminer l'inconnue  $x_1$  des lignes 2 à  $n$ . Le terme  $a_{11}$  est appelé **pivot** et on note pour la suite :

$$\pi_1 = a_{11}$$

La ligne  $i$  devient alors:

$$0x_1 + \left(a_{i2} - \frac{a_{i1}}{\pi_1}a_{12}\right)x_2 + \dots + \left(a_{in} - \frac{a_{i1}}{\pi_1}a_{1n}\right)x_n = b_i - \frac{a_{i1}}{\pi_1}b_1$$

ce que l'on écrit encore

$$a_{i2}^{(2)}x_2 + \dots + a_{in}^{(2)}x_n = b_i^{(2)}, \forall i > 1$$

# Etape 1



en posant

$$a_{i2}^{(2)} = a_{i2} - \frac{a_{i1}}{\pi_1} a_{12}$$

$$\vdots$$

$$a_{in}^{(2)} = a_{in} - \frac{a_{i1}}{\pi_1} a_{1n}$$

$$b_i^{(2)} = b_i - \frac{a_{i1}}{\pi_1} b_1$$

Si on pose aussi, pour  $1 \leq j \leq n$

$$a_{1j}^{(2)} = a_{1j}, \forall 1 \leq j \leq n \text{ et } b_1^{(2)} = b_1$$

on a obtenu le système équivalent :

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

# Étape 1



avec

$$A^{(2)} = \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \cdots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix}$$

# Étape 1



On voit que

$$A^{(2)} = M^{(1)}A \text{ où } M^{(1)} = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & \ddots & & \vdots \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -\frac{a_{n1}}{a_{11}} & 0 & \dots & 0 & 1 \end{pmatrix}$$

Le système s'écrit alors

$$A^{(2)} = M^{(1)}A, \text{ et } b^{(2)} = M^{(1)}b$$

Tout se passe comme si on avait prémultiplié à gauche le système initial par  $M^{(1)}$ .



# Etape k



On suppose que l'on a pu itérer le procédé ci-dessus  $k - 1$  fois, c'est que l'on n'a jamais rencontré de pivot nul :

$$\pi_i = a_{ii}^{(i)} \neq 0 \text{ pour } 1 \leq i \leq k - 1$$

On obtient alors le système équivalent :

$$A^{(k)}x = b^{(k)}$$

## Etape k



avec  $A^{(k)}$  de la forme

$$A^{(k)} = \begin{pmatrix} a_{11}^{(k)} & a_{12}^{(k)} & \dots & \dots & \dots & \dots & \dots & a_{1n}^{(k)} \\ 0 & a_{22}^{(k)} & \ddots & & & & & a_{2n}^{(k)} \\ \vdots & \ddots & a_{33}^{(k)} & \ddots & & & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & & \vdots \\ 0 & \dots & \dots & 0 & a_{k-1,k-1}^{(k)} & \dots & \dots & a_{k-1,n}^{(k)} \\ \vdots & & & \vdots & 0 & & & \\ \vdots & & & \vdots & \vdots & & \widetilde{A^{(k)}} & \\ 0 & \dots & \dots & 0 & 0 & & & \end{pmatrix} \quad (1)$$

# Etape k



où  $\widetilde{A}^{(k)}$  est une matrice carrée d'ordre  $n - k + 1$ :

$$\widetilde{A}^{(k)} = \begin{pmatrix} a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ \vdots & & \vdots \\ a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix} \quad (1)$$

# Etape k



Comme précédemment, on doit effectuer une hypothèse sur la valeur du pivot :

$$\pi_k = a_{kk}^{(k)} \neq 0$$

On peut alors introduire la matrice

$$M^{(k)} = \begin{pmatrix} 1 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 1 & \ddots & & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & & \vdots \\ \vdots & \vdots & 0 & 1 & \ddots & & & \vdots \\ \vdots & & & -\frac{a_{k+1,k}^{(k)}}{\pi_k} & \ddots & \ddots & & \vdots \\ \vdots & & & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & -\frac{a_{n,k}^{(k)}}{\pi_k} & 0 & \dots & 0 & 1 \end{pmatrix} \quad (1)$$

## Etape k



Remarquons que l'inverse de  $M^{(k)}$  est  $L^{(k)}$ :

$$L^{(k)} = \begin{pmatrix} 1 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 1 & \ddots & & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & & \vdots \\ \vdots & \vdots & 0 & 1 & \ddots & & & \vdots \\ \vdots & & & \frac{a_{k+1,k}^{(k)}}{\pi_k} & \ddots & \ddots & & \vdots \\ \vdots & & & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & \frac{a_{n,k}^{(k)}}{\pi_k} & 0 & \dots & 0 & 1 \end{pmatrix} \quad (1)$$

Soit

$$A^{(k+1)} = M^{(k)}A^{(k)} \text{ et } b^{(k+1)} = M^{(k)}b^{(k)}$$

# Etape k



alors

$$A^{(k+1)}x = b^{(k+1)}$$

et

$$A^{(k+1)} = \begin{pmatrix} a_{11}^{(k+1)} & a_{12}^{(k+1)} & \dots & \dots & \dots & \dots & \dots & a_{1n}^{(k+1)} \\ 0 & a_{22}^{(k+1)} & \ddots & & & & & a_{2n}^{(k+1)} \\ \vdots & \ddots & a_{33}^{(k+1)} & \ddots & & & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & & \vdots \\ 0 & \dots & \dots & 0 & a_{k,k}^{(k+1)} & \dots & \dots & a_{k,n}^{(k+1)} \\ \vdots & & & \vdots & 0 & & & \\ \vdots & & & \vdots & \vdots & & \widetilde{A^{(k+1)}} & \\ 0 & \dots & \dots & 0 & 0 & & & \end{pmatrix}$$

# Etape k



où  $\widetilde{A}^{(k+1)}$  est une matrice carrée d'ordre  $n - k$  :

$$\widetilde{A}^{(k+1)} = \begin{pmatrix} a_{k+1,k+1}^{(k+1)} & \cdots & a_{k+1n}^{(k+1)} \\ \vdots & & \vdots \\ a_{nk+1}^{(k+1)} & \cdots & a_{nn}^{(k+1)} \end{pmatrix}$$

# Bilan



## Récapitulons

Si on appelle  $L$  la matrice triangulaire inférieure avec des 1 sur la diagonale, et  $L^{(k)}$  la matrice définie en (??)

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

et  $U$  la matrice triangulaire supérieure

$$U = A^{(n)}$$

on a

$$A = LU$$

On dit qu'on a effectué une **factorisation de Gauss** ou **factorisation LU** de  $A$ .



# Utilisation : Calcul de déterminant



## Calcul de déterminant

Une utilisation de la factorisation LU est le calcul de déterminant de  $A$  : en effet, si  $A$  admet une factorisation LU, on a  $\det A = \det(LU) = \det L \cdot \det U$ . Or  $L$  est triangulaire à diagonale unité donc de déterminant 1, ce qui donne finalement

$$\det A = \det U = \prod_{i=1}^n \pi_i$$

La factorisation LU permet donc de calculer le déterminant de  $A$  avec une complexité de l'ordre de  $\frac{n^3}{3}$  additions et multiplications, au lieu de  $n!$  avec la formule du déterminant !

# Utilisation : Résolution de systèmes linéaires



## Résolution de systèmes linéaires

La factorisation LU permet de résoudre successivement deux systèmes triangulaires :  $Ly = b$  puis  $Ux = y$  dont la solution  $x$  vérifie  $Ax = b$ .

Si on dispose de plusieurs systèmes linéaires de seconds membres différents mais de même matrice  $A$  :

$$Ax = b_i \text{ pour } i = 1, \dots, l$$

il suffit de calculer une seule fois les matrices  $L$  et  $U$  et de les stocker. On peut alors effectuer autant de descentes et de remontées qu'on a de systèmes pour les résoudre tous, sans pour autant refaire la factorisation LU.

Pour  $l$  systèmes à résoudre, la complexité est de l'ordre de  $\frac{n^3}{3} + l.n^2$  additions et multiplications plutôt que  $l.\frac{n^3}{3}$ .

# Stockage



## Principe de l'écrasement

Pour une matrice de taille  $n$ , on remarque que le nombre de termes à connaître pour disposer des matrices  $L$  et  $U$  n'est que de  $n^2$ , puisque la diagonale unité de  $L$  n'est pas à stocker. De ce fait on peut utiliser la place mémoire disponible dans  $A$  pour y enregistrer les termes de  $L$  et  $U$ .

Cette remarque explique que l'algorithme qui suit "écrase" la matrice  $A$  par sa décomposition LU.

# Stockage



$$\begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ * & 1 & 0 \\ * & * & 1 \end{pmatrix} \begin{pmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \end{pmatrix}$$

stocké en

$$\begin{pmatrix} 1 & * & * \\ * & 1 & * \\ * & * & 1 \end{pmatrix}$$