Résolution d'un système d'équations linéaires

M. ROUAN-SERIK

Institut de maintenance et de sécurité industrielle IMSI
Université d'Oran 2 Mohamed Ben Ahmed.
mehdi.rouan@gmail.com

Plan

$Ax = b$

- Résoudre un système d'équation à n inconnues.
- Les méthodes traditionnelles (*Cramer* par eg.) ne sont plus pratique au delà de certaines valeurs de n.
- Méthodes directes plus efficaces ou itératives plus rapide
- Plusieurs problèmes en sont liés : Modélisation d'un système de refroidissement, etc. . .

Méthode de *Cramer*

Cette méthode consiste à résoudre un système en calculant plusieurs déterminants.

$$\begin{cases} x + y = 5 \\ x - y = 1 \end{cases}$$

On aura comme solution :

$$x = \frac{\begin{vmatrix} 5 & 1 \\ 1 & -1 \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}} = 3, \qquad y = \frac{\begin{vmatrix} 1 & 5 \\ 1 & 1 \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}} = 2$$

Cette méthode nécessite $(n+1)(n!-1)$ additions, et $(n+1)(n-1)n!$ multiplications et n divisions.

Plan

Méthode de la descente I

On veut résoudre le système $Ax = b$, avec A une matrice triangulaire inférieur i-e : $\forall i, j = 1, 2, \ldots, n,\ i < j \Leftrightarrow a_{ij} = 0$.

$$\begin{array}{lllll} a_{11}x_1 & & & = & b_1 \\ a_{21}x_1 & +a_{22}x_2 & & = & b_2 \\ \vdots & \vdots & \ddots & & \vdots \\ a_{n1}x_1 & +a_{n2}x_2+ \cdots + & a_{nn}x_n & = & b_n \end{array}$$

La solution sera :

$$x_1 = \frac{b_1}{a_{11}}$$

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

Méthode de la descente II

Algorithme 1 : La descente ;

Entrées : La matrice A et le second membre b

Sorties : x solution du système

$x_1 \leftarrow b_1/a_{11}$;

pour i ***de*** *2* ***à*** n **faire**

  $x_i \leftarrow b_i$;

  pour j ***de*** *1* ***à*** $i-1$ **faire**

    $x_i \leftarrow x_i - a_{ij} * x_j$;

  finpour;

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

finpour;

retourner x

La remontée

On l'utilise que si la matrice A est triangulaire supérieure et donc des systèmes :

$$\begin{array}{llll} a_{11}x_1 & +a_{12}x_2+ \quad \cdots + & a_{1n}x_n & = b_1 \\ & \ddots & \vdots & \vdots \\ & a_{n-1n-1}x_{n-1} & +a_{n-1n}x_n & = b_{n-1} \\ & & a_{nn}x_n & = b_n \end{array}$$

La solution sera :

$$x_n = \frac{b_n}{a_{nn}}$$

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

L'algorithme de cette méthode se déduit facilement à partir de la précédente.

Méthode de *Gauss*

$$A = \begin{pmatrix} 2 & 8 & 4 \\ 2 & 10 & 6 \\ 1 & 8 & 2 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Méthode de *Gauss*

$$A = \begin{pmatrix} 2 & 8 & 4 \\ 2 & 10 & 6 \\ 1 & 8 & 2 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$A^{(1)} = \begin{pmatrix} 2 & 8 & 4 \\ 0 & 2 & 2 \\ 0 & 4 & 0 \end{pmatrix}, b^{(1)} = \begin{pmatrix} 1 \\ 0 \\ 1/2 \end{pmatrix}$$

Méthode de *Gauss*

$$A = \begin{pmatrix} 2 & 8 & 4 \\ 2 & 10 & 6 \\ 1 & 8 & 2 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$A^{(1)} = \begin{pmatrix} 2 & 8 & 4 \\ 0 & 2 & 2 \\ 0 & 4 & 0 \end{pmatrix}, b^{(1)} = \begin{pmatrix} 1 \\ 0 \\ 1/2 \end{pmatrix}$$

$$A^{(2)} = \begin{pmatrix} 2 & 8 & 4 \\ 0 & 2 & 2 \\ 0 & 0 & -4 \end{pmatrix}, b^{(2)} = \begin{pmatrix} 1 \\ 0 \\ 1/2 \end{pmatrix}$$

Méthode de *Gauss*

$$A = \begin{pmatrix} 2 & 8 & 4 \\ 2 & 10 & 6 \\ 1 & 8 & 2 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$A^{(1)} = \begin{pmatrix} 2 & 8 & 4 \\ 0 & 2 & 2 \\ 0 & 4 & 0 \end{pmatrix}, b^{(1)} = \begin{pmatrix} 1 \\ 0 \\ 1/2 \end{pmatrix}$$

$$A^{(3)} = \begin{pmatrix} 2 & 8 & 4 \\ 0 & 2 & 2 \\ 0 & 0 & -4 \end{pmatrix}, b^{(3)} = \begin{pmatrix} 1 \\ 0 \\ 1/2 \end{pmatrix}$$

Ce système pourra ensuite être résolu à l'aide de la méthode de la remontée.

Algorithme de la méthode

Algorithme 2 : La méthode de Gauss ;

Entrées : La matrice $[A \mid b]$ augmentée

Sorties : La matrice échelonnée A' augmentée

$A' \leftarrow [A \mid b]$;

pour k ***de 1 à*** $n-1$ **faire**

  pour i ***de*** $k+1$ ***à*** n **faire**

    pour j ***de*** $k+1$ ***à*** $n+1$ **faire**

      $a'_{ij} \leftarrow a'_{ij} - a'_{ik} * a'_{kj}/a'_{kk}$;

    finpour;

  finpour;

  pour j ***de*** $k+1$ ***à*** n **faire**

    $a'_{jk} \leftarrow 0$;

  finpour;

finpour;

retourner A'

Méthode de *Gauss-Jordan*

```
function [Ap X]=Gauss_Jordan(A,b)
% Retourne la solution X du systeme d'equation Ax=b
 Ap=[A b];
 [n m]=size(Ap);
 for k=1:n
  Ap(k,:)=Ap(k,:)/Ap(k,k);
  for i=[1:k-1;k+1:n]
   for j=k+1:n+1
    Ap(i,j)=Ap(i,j)-Ap(i,k)*Ap(k,j);
   end
  end
  for i=1:n
   if(i ~= k)
    Ap(i,k)=0;
   end
  end
 X=Ap(:,[n+1:m]);
 end
end
```

Plan

Méthode générale

On veut résoudre le système $Ax = b$ avec A à DSD :

$$\forall i = 1, 2, \ldots, n \quad |a_{ii}| > \sum_{j \neq i} |a_{ij}|$$

On cherche à décomposer A en $A = M - N$ avec M inversible. On aura ensuite cette formule de récurrence :

$$\begin{cases} x^0 \text{ arbitraire,} \\ x^{k+1} = M^{-1}Nx^k + M^{-1}b, \\ ||x^{i+1} - x^i|| < \varepsilon, \ \forall i \end{cases}$$

Méthode de *Jacobi*

Par cette méthode A se décompose comme suit :

$$\begin{cases} A = M - N, & M \text{ inversible} \\ M = D, & D : (a_{ii}) \\ N = E + F, & E = (-a_{ij}, i > j)\,, F = (-a_{ij}, i < j) \end{cases}$$

Et la formule de récurrence suivante :

$$\begin{cases} x^0 \text{ arbitraire}, \\ x^{k+1} = D^{-1}(E+F)x^k + D^{-1}b, \\ ||x^{i+1} - x^i|| < \varepsilon, \ \forall i \end{cases}$$

Méthode de *Jacobi*

Algorithme 3 : La méthode de Jacobi ;

Entrées : $A, b, \varepsilon, n, x^0$

Sorties : x^{new}

pour i ***de*** *1* ***à*** n **faire**

  $x_i^{new} \leftarrow x_i^0$;

finpour;

tant que $\| Ax^{new} - b \| > \varepsilon$ **faire**

  pour i ***de*** *1* ***à*** n **faire**

    $x_i^{old} \leftarrow x_i^{new}$;

  finpour;

  pour i ***de*** *1* ***à*** n **faire**

$$x_i^{new} \leftarrow \frac{b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j^{old}}{a_{ii}};$$

  finpour;

fin tq

retourner x^{new}

Méthode de *Gauss-Seidel*

Par cette méthode A se décompose comme suit :

$$\begin{cases} A = M - N \\ M = D - E \\ N = F \end{cases}$$

Et la formule de récurrence suivante :

$$\begin{cases} x^0 \text{ arbitraire,} \\ x^{k+1} = (D-E)^{-1}Fx^k + (D-E)^{-1}b, \\ ||x^{i+1} - x^i|| < \varepsilon, \ \forall i \end{cases}$$

Méthode de *Gauss-Seidel*

Algorithme 4 : La méthode de Jacobi ;

Entrées : $A, b, \varepsilon, n, x^0$

Sorties : x^{new}

pour i ***de 1 à*** n **faire**

  $x_i^{new} \leftarrow x_i^0$;

finpour;

tant que $\| Ax^{new} - b \| > \varepsilon$ **faire**

  pour i ***de 1 à*** n **faire**

    $x_i^{old} \leftarrow x_i^{new}$;

  finpour;

  pour i ***de 1 à*** n **faire**

$$x_i^{new} \leftarrow \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{new} - \sum_{j=i+1}^{n} a_{ij} x_j^{old}}{a_{ii}};$$

  finpour;

fin tq

retourner x^{new}

Plan

Conclusion

- Méthode directe de résolution sont les plus efficaces en terme de la qualité de solution.
- Ces méthodes ont une limite quant aux nombres d'équations (quand n devient grand).
- Méthode itérative donnent une bonne solution approchées dans un nombre d'itérations raisonnable.
- Elles sont limitées en utilisation de part la restriction sur des matrices particulière (critères de convergence).
- Se sont des méthodes stables numériquement et peuvent être utilisées pour des matrices de grande taille.
- Il existe d'autre méthodes de résolution itératives telles que : L'algorithme du gradient conjugué et la méthode de *Krylov*.