RÉSOLUTION DE SYSTÈMES LINÉAIRES PAR DÉCOMPOSITION DE MATRICE

David Ryckelynck

Centre des Matériaux, Mines ParisTech

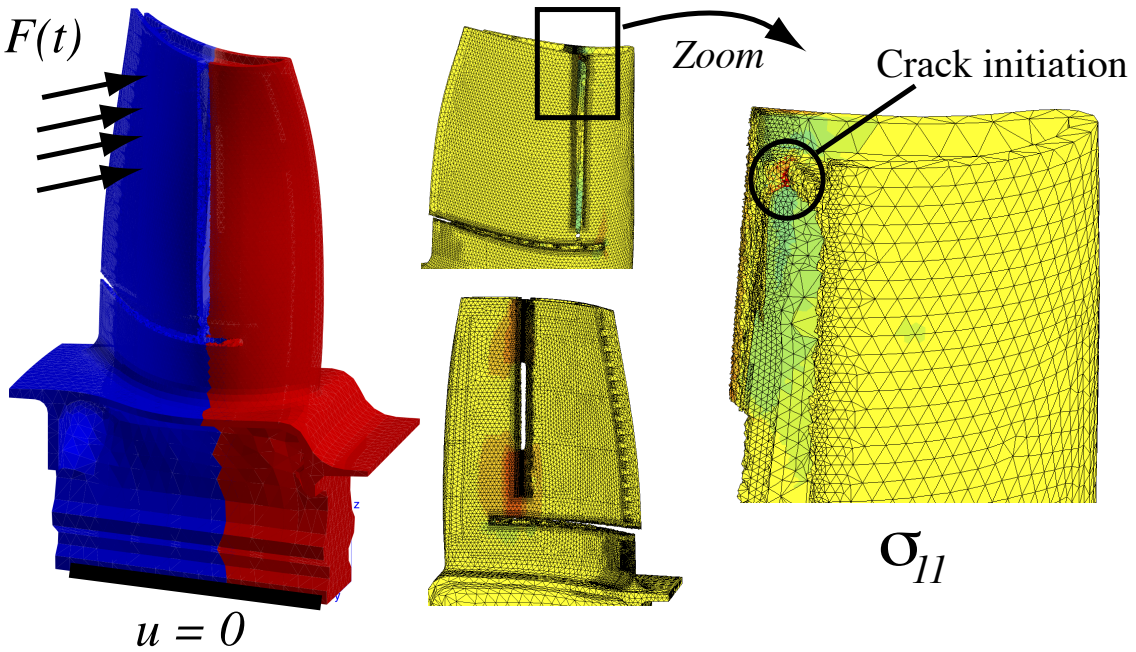
24 septembre 2015

Plan du cours

Motivations

- La résolution d'équations aux dérivées partielles, linéaire ou non linéaire, conduit à la résolution de systèmes linéaires de grande taille.
- La matrice d'un système linéaire incorpore tous les couplages issus du modèle physique. C'est le cas par exemple pour les couplages induits par des conditions d'équilibre.
- Pour les problèmes à variables dépendantes du temps et de l'espace, la résolution de systèmes linéaires intervient dans l'intégration implicite des équations d'évolution.
- Nous abordons dans ce cours les **méthodes directes**, en opposition aux méthodes itératives qui construisent une suite convergente d'approximations.
- Les méthodes directes sont adaptées à de grandes classes de problèmes linéaires.

Illustration



Calcul d'une aube entaillée soumise à un chargement cyclique [thèse Mélanie Leroy 2013].

Il s'agit de la simulation d'un essai de flexion pour modéliser la durée de vie d'un joint de grain [thèse Mélanie Leroy 2013]. Le but de simulation est d'estimer l'état de contrainte dans le voisinage du joint de grain.

La simulation non linéaire de 5 cycles de chargement dure 67 heures, pour 131 pas de temps et 1,7 millions variables de déplacements à déterminer à chaque pas de temps.

La résolution de systèmes linéaires représente 75% de la durée totale de la simulation. Réduire de 20% ce temps de calcul permet de gagner environ 13h !

Remarque sur les systèmes non linéaires

A l'aide de l'algorithme de Newton Raphson, la recherche d'une solution approchée d'un problème non linéaire s'obtient par une suite de corrections linéaires.

Soit $\mathbf{R}(\mathbf{q}) = 0$ le système de N équations non linéaires dont la solution est $\mathbf{q} \in \mathbb{R}^N$. On suppose qu'il existe une matrice jacobienne $\mathbf{J}$:

$$\mathbf{R}(\mathbf{q} + d\mathbf{q}) = \mathbf{R}(\mathbf{q}) + \mathbf{J}(\mathbf{q})\, d\mathbf{q}, \text{ au premier ordre quand } d\mathbf{q} \to 0$$

Connaissant une estimation de $\mathbf{q}$ notée $\mathbf{q}_n$ on cherche une correction $\delta\mathbf{q}$ tel que :

$$0 = \mathbf{R}(\mathbf{q}_n) + \mathbf{J}(\mathbf{q}_n)\, \delta\mathbf{q}, \quad \mathbf{q}_{n+1} = \mathbf{q}_n + \delta\mathbf{q}$$

Ce qui nécessite de résoudre le système linéaire suivant :

$$\mathbf{J}(\mathbf{q}_n)\, \delta\mathbf{q} = -\mathbf{R}(\mathbf{q}_n)$$

Notations

Dans la suite du cours, on s'intéressera au système linéaire suivant :

$$\mathbf{A}.\mathbf{x} = \mathbf{b}, \quad \mathbf{A} \in \mathbb{R}^{N \times N}, \quad \mathbf{b} \in \mathbb{R}^{N}, \quad \mathbf{x} \in \mathbb{R}^{N}$$

Dans le cas d'une résolution multiple, pour différent second membres, on adopte la notation suivante :

$$\mathbf{A}.\mathbf{X} = \mathbf{B}, \quad \mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_m], \ \mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_m]$$

On aura une attention particulière au cas des matrices creuses ou bandes. Ce type de matrice est fréquent quand on résout des EDP en thermomécanique des structures.

Notation par bloc :

$$[\overline{\mathbf{B}}, \widetilde{\mathbf{B}}] = [\overline{\mathbf{b}}_1, \ldots, \overline{\mathbf{b}}_{\overline{m}}, \widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_{\widetilde{m}}], \quad \overline{\mathbf{B}} \in \mathbb{R}^{N \times \overline{m}}, \quad \widetilde{\mathbf{B}} \in \mathbb{R}^{N \times \widetilde{m}}$$

$$\overline{\mathbf{b}}_1 = \begin{bmatrix} \vdots \\ \overline{B}_{i1} \\ \vdots \end{bmatrix}$$

Remarques préliminaires

Etude d'un cas de découplage d'équations.

Cas d'un bloc d'équations découplé de la première variable x_1 :

$$\mathbf{A} = \begin{bmatrix} 1 & \mathbf{W} \\ 0 & \widetilde{\mathbf{A}} \end{bmatrix}, \quad \mathbf{x}^T = [x_1, \widetilde{\mathbf{x}}^T], \quad \mathbf{b}^T = [b_1, \widetilde{\mathbf{b}}^T]$$

On peut alors déterminer $\widetilde{\mathbf{x}}$ puis déterminer x_1 de façon explicite :

$$\widetilde{\mathbf{A}}.\widetilde{\mathbf{x}} = \widetilde{\mathbf{b}}$$

$$x_1 = b_1 - \mathbf{W}.\widetilde{\mathbf{x}}$$

La **complexité numérique** de l'équation donnant x_1 est inférieure à N produits et N additions (moins de $2N$ Flops, opérations à virgule flottante).

Cas des matrices triangulaires supérieures

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ 0 & A_{22} & A_{23} & A_{24} \\ 0 & 0 & A_{33} & A_{34} \\ 0 & 0 & 0 & A_{44} \end{bmatrix}$$

On calcul $x_4 = \frac{b_4}{A_{44}}$ (x_N dans le cas général), puis x_3 ... jusqu'à x_1.

Il y a une division, puis une multiplication, une addition et une division, ...

Il faut N divisions, et $(N-1)^2 - \frac{N-1}{2}$ multiplications et additions. Soit une complexité numérique proportionnelle à N^2 quand N est grand. Pour une matrice creuse ayant ω termes non nuls en moyenne par ligne, on a une complexité proportionnelle à $N\,\omega$. En général ω est fonction de N.

Méthode d'élimination de Gauss-Jordan

Cette méthode permet de comprendre la problématique de la résolution numérique de systèmes linéaires. Ce n'est pas la plus performante.

Considérons le cas suivant :

$$\mathbf{A}.\mathbf{X} = \mathbf{B}$$

On peut permuter deux lignes de **A** sans changer la solution **x**, en effectuant la même permutation de ligne sur **B**.
On peut modifier une ligne de **A** avec une combinaison linéaire d'autres lignes de **A** sans changer la solution **x**, en effectuant la même combinaison de lignes sur **B**.

On peut permuter deux colonnes de **A** si l'on permute les lignes correspondantes de **X**. Il faudra dans ce cas reconstruire les solutions avec la numérotation d'origine.

On cherche à transformer **A** en **I** par des permutations et des combinaisons linéaires.

Gauss (1777 - 1855)

Elimination de Gauss-Jordan sans pivot de Gauss

On ne réalise que des combinaisons linéaires de lignes de **A**, de **B** et **I**.

$$\mathbf{A}^{(1)} = \mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}$$

On suppose $A_{11} \neq 0$. On divise la première ligne par A_{11} et on soustrait cette ligne à chaque ligne i divisée par A_{i1} si $A_{i1} \neq 0$.

$$\widehat{\mathbf{A}}^{(1)} = \begin{bmatrix} 1 & \frac{A_{12}}{A_{11}} & \frac{A_{13}}{A_{11}} & \frac{A_{14}}{A_{11}} \\ A_{21} - c_2^{(1)} A_{11} = 0 & A_{22} - c_2^{(1)} A_{12} & A_{23} - c_2^{(1)} A_{13} & A_{24} - c_2^{(1)} A_{14} \\ A_{31} - c_3^{(1)} A_{11} = 0 & A_{32} - c_3^{(1)} A_{12} & A_{33} - c_3^{(1)} A_{13} & A_{34} - c_3^{(1)} A_{14} \\ A_{41} - c_4^{(1)} A_{11} = 0 & A_{42} - c_4^{(1)} A_{12} & A_{43} - c_4^{(1)} A_{13} & A_{44} - c_4^{(1)} A_{14} \end{bmatrix}$$

avec $c_2^{(1)} = \frac{A_{21}}{A_{11}}$, ..., $c_j^{(1)} = \frac{A_{j1}}{A_{11}}$

Elimination de Gauss-Jordan sans pivot de Gauss

On obtient alors un certain découplage :

$$\widehat{\mathbf{A}}^{(1)} = \begin{bmatrix} 1 & \mathbf{W} \\ 0 & \mathbf{A}^{(2)} \end{bmatrix}, \quad \mathbf{b}^T = [\frac{b_1}{A_{11}}, (b_j - c_j^{(1)}\, b_1)_{j=2}^N]$$

$$\mathbf{W} = [\frac{A_{12}}{A_{11}}, \frac{A_{13}}{A_{11}}, \frac{A_{14}}{A_{11}}]$$

$$\mathbf{A}^{(2)} = \begin{bmatrix} A_{22} - c_2^{(1)} A_{12} & A_{23} - c_2^{(1)} A_{13} & A_{24} - c_2^{(1)} A_{14} \\ A_{32} - c_3^{(1)} A_{12} & A_{33} - c_3^{(1)} A_{13} & A_{34} - c_3^{(1)} A_{14} \\ A_{42} - c_4^{(1)} A_{12} & A_{43} - c_4^{(1)} A_{13} & A_{44} - c_4^{(1)} A_{14} \end{bmatrix}$$

Il reste à traiter $\mathbf{A}^{(2)}$ de la même façon, afin d'obtenir $\widehat{\mathbf{A}}^{(2)}$:

$$\widehat{\mathbf{A}}^{(2)} = \begin{bmatrix} 1 & \mathbf{L} \\ 0 & \mathbf{A}^{(3)} \end{bmatrix}$$

puis on continue jusqu'à obtenir une matrice triangulaire.

Dans la méthode de Gauss-Jordan, on élimine également sur la colonne traitée, les termes des lignes précédentes. On construit ainsi une matrice identité.

Elimination de Gauss-Jordan sans pivot de Gauss

Les opérations précédentes sont linéaires. Il existe une matrice $\mathbf{P}^{(1)}$ telle que :

$$\widehat{\mathbf{A}}^{(1)} = \mathbf{P}^{(1)}.\mathbf{A}^{(1)}$$

avec un produit à gauche pour modifier des lignes, avec un traitement identique de toutes les colonnes.

Par identification, on en déduit que :

$$\mathbf{P}^{(1)} = \begin{bmatrix} \frac{1}{A_{11}} & 0 & 0 & 0 \\ -c_2^{(1)} & 1 & 0 & 0 \\ -c_3^{(1)} & 0 & 1 & 0 \\ -c_4^{(1)} & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{P}^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{A_{11}^{(2)}} & 0 & 0 \\ 0 & -c_2^{(2)} & 1 & 0 \\ 0 & -c_3^{(2)} & 0 & 1 \end{bmatrix}$$

Par récurrence on a :

$$\widehat{\mathbf{A}}^{(2)} = \mathbf{P}^{(2)}.\mathbf{P}^{(1)}.\mathbf{A}$$

A la fin, on obtient la matrice triangulaire supérieur **U** (Upper) :

$$\mathbf{U} = \mathbf{P}.\mathbf{A}, \quad \mathbf{P} = \mathbf{P}^{(4)}.\mathbf{P}^{(3)}.\mathbf{P}^{(2)}.\mathbf{P}^{(1)}$$

où **P** est la matrice de permutation, avec **U** et **P** inversibles.

Décomposition LU, cas $\mathbf{A}.\mathbf{X} = \mathbf{B}$

On note **L** l'inverse de **P**. On obtient alors la décomposition LU de **A** :

$$\mathbf{L}.\mathbf{U} = \mathbf{A}$$

Propriété : **P** étant triangulaire inférieure, **L** est une matrice triangulaire inférieure (Lower).

Ici l'analyse est purement formelle. En pratique on conserve l'algorithme d'élimination. On a :

$$\mathbf{U} = \mathbf{P}\,\mathbf{A} \quad \Rightarrow \quad \mathbf{U}.\mathbf{X} = \mathbf{P}.\mathbf{B}$$

Or $\widehat{\mathbf{X}} = \mathbf{P}.\mathbf{B}$ est la solution de $\mathbf{L}.\widehat{\mathbf{X}} = \mathbf{B}$. Après permutation par la matrice **P**, il reste à résoudre un système triangulaire supérieur pour trouver **X** :

$$\mathbf{U}.\mathbf{X} = \widehat{\mathbf{X}}$$

Décomposition LU, cas supplémentaire $\widetilde{\mathbf{b}}$

Dans ce cas, on commence par rechercher $\widehat{\mathbf{x}}$, puis on en déduit la solution recherchée $\widetilde{\mathbf{x}}$ tel que :

$$\mathbf{L}.\widehat{\mathbf{x}} = \widetilde{\mathbf{b}}$$

$$\mathbf{U}.\widetilde{\mathbf{x}} = \widehat{\mathbf{x}}$$

$$\Rightarrow \quad \widetilde{\mathbf{b}} - \mathbf{A}.\widetilde{\mathbf{x}} = 0$$

Il faut donc stocker en mémoire **L** et **U**.

Le pivot de Gauss

Si A_{11} est nul, il faut permuter la ligne 1 avec une autre. Dans le cas du **pilotage partiel**, on ne fait que des permutations de lignes. On cherche la ligne qui maximise $|A_{i1}|$ pour la permuter avec la première ligne.

La matrice de permutation de la ligne 3 avec la ligne 1 s'écrit :

$$\mathbf{P}^{(1-3)} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Dans le cas du **pivotage total**, on réalise aussi une recherche sur les colonnes, pour trouver un terme A_{ij} maximum (ou significatif), qui remplace A_{11}.

Prorpiété de la décomposition LU

Définition : Une méthode numérique de résolution de système linéaire est dite mathématiquement stable lorsque ”quelque soit la matrice **A** régulière, l'algorithme réussit”.

Théorème : La méthode de GAUSS avec une stratégie de pivotage est mathématiquement stable pour toute matrice régulière.

Corollaire : Si au cours d'une factorisation de GAUSS avec pivotage, un pivot nul est détecté alors la matrice est singulière et ce système n'a pas de solution unique.

Théorème : La méthode de GAUSS (sans pivotage) est stable pour des matrices réelles définies positives.

Proposition : La décomposition LU n'est pas unique. Pour la rendre unique, il faut spécifier la diagonale de **L** ou de **U**.

Décomposition de Crout

Pour les matrices **A** ne nécessitant pas de pivotage, on peut utiliser la décomposition de Crout.

On procède par identification du produit **L**.**U**, avec $L_{ii} = 1$:

$$A_{ij} = \sum_{k=1}^{N} L_{ik}\ U_{kj}, \quad \text{avec } L_{ik} = 0 \text{ pour } k > i,\ U_{kj} = 0 \text{ pour } k > j$$

Donc

$$\text{pour } i \leq j \quad A_{ij} = U_{ij} + \sum_{k=1}^{i-1} L_{ik}\ U_{kj}$$

$$\text{pour } i > j \quad A_{ij} = \sum_{k=1}^{j} L_{ik}\ U_{kj}$$

On commence par la première ligne de **A** pour en déduire U_{1j}, de façon progressive de $j = 1$ à $j = N$. Puis on traite la première colonne de la deuxième ligne :

$$L_{21} = \frac{A_{i1}}{U_{11}}$$

Puis on traite le reste de la deuxième ligne...

Décomposition de matrices symétriques

La décomposition de Crout appliquée aux matrice symétrique régulière (inversible) conduit à la proposition suivante : Il existe **L** triangulaire inférieure à diagonale unitaire et **D** matrice diagonale régulière tel que :

$$\mathbf{A} = \mathbf{L}.\mathbf{D}.\mathbf{L}^T$$

Décomposition de Cholesky des matrices symétriques définies positives

Si la matrice $\mathbf{A}$ est symétrique positive, $\mathbf{x}^T.\mathbf{A}.\mathbf{x} > 0 \quad \forall \mathbf{x} \neq 0$ alors les termes de la diagonale de $\mathbf{D}$ sont strictement positifs. On a alors : $\mathbf{D} = \mathbf{D}^{1/2}.(\mathbf{D}^{1/2})^T$.

On obtient donc la décomposition de Cholesky :

$$\mathbf{A} = \widetilde{\mathbf{L}}.\widetilde{\mathbf{L}}^T$$

avec $\widetilde{\mathbf{L}}$ matrice triangulaire inférieure et $\widetilde{\mathbf{L}} = \mathbf{L}.\mathbf{D}^{1/2}$.

Pour construire la décomposition de Cholesky on procède également par identification du produit :

$$\text{pour } i \leq j \quad A_{ij} = \sum_{k=1}^{i} \widetilde{L}_{ik}\, \widetilde{L}_{jk}$$

On parcours la matrice comme pour Crout.

Complexité des méthodes du type Gauss

Pour les matrices de largeur de bande ω la complexité numérique des méthodes de type Gauss est proportionnelle à $N\,\omega^2$. Soit pour les matrices pleines N^3...

Mais, lorsque l'on change uniquement de second membre la complexité numérique est proportionnelle à N^2 pour le traitement des systèmes triangulaires.

Une renumérotation des inconnues permettant de réduire la largeur de bande permettra une décomposition plus rapide.

Erreurs d'arrondi et conditionnement

Par définition :

$$\text{Cond}(\mathbf{A}) = \|\mathbf{A}\|\,\|\mathbf{A}^{-1}\|, \quad \|\mathbf{A}\| = \sup_{\mathbf{v}} \frac{\|\mathbf{A}.\mathbf{v}\|}{\|\mathbf{v}\|}$$

Considérons l'erreur $\delta\mathbf{A}$ sur la matrice :

$$(\mathbf{A} + \delta\mathbf{A}).(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b}, \quad \mathbf{A}.\mathbf{x} = \mathbf{b}$$

avec $\delta\mathbf{A}.\delta\mathbf{x}$ négligeable. Ainsi :

$$\delta\mathbf{x} = -\mathbf{A}^{-1}.\delta\mathbf{A}.\mathbf{x}$$

Donc :

$$\|\delta\mathbf{x}\| \leq \|\mathbf{A}^{-1}\|\,\|\mathbf{A}\|\,\frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|}\,\|\mathbf{x}\|$$

En conclusion :

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \text{Cond}(\mathbf{A})\,\frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|}$$

Un mauvais conditionnement amplifie les erreurs d'arrondi générées lors de la décomposition de $\mathbf{A}$.

Rappel sur les normes matricielles : $\|\mathbf{A}.\mathbf{B}\| \leq \|\mathbf{A}\|\,\|\mathbf{B}\|$

Réduire le conditionnement

Pour réduire le conditionnement il est possible de réaliser une mise à l'échelle avec deux matrices diagonales $\mathbf{W}^1$, $\mathbf{W}^2$ tel que :

$$\text{Cond}(\mathbf{W}^1.\mathbf{A}.\mathbf{W}^2) < \text{Cond}(\mathbf{A})$$

On peut utiliser par exemple :

$$\mathbf{W}^1 = \mathbf{W}^2$$

$$(\mathbf{W}^1)_{ii} = \frac{1}{A_{ii}} \text{ si } A_{ii} \neq 0, \text{ sinon } (\mathbf{W}^1)_{ii} = 1$$

Exercice

On considère une matrice de rigidité associée à un système de ressorts. L'énergie potentielle $E(\mathbf{x})$ est la somme des énergies de chaque ressort reliant x_i à x_j, moins le travail des efforts extérieurs (ici x_1 F) :

$$E(\mathbf{x}) = \sum_{k=1}^{m} \frac{1}{2}\lambda \left(x_{L(k,1)} - x_{L(k,2)}\right)^2 + \frac{1}{2}\lambda\, x_1^2 - x_1\, F$$

$L(k,:)$ est une table de connectivité où une ligne indique les variables reliées par un ressort.
Par définition la matrice de rigidité $\mathbf{A}$ est symétrique positive et elle vérifie la propriété suivante :

$$E(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T.\mathbf{A}.\mathbf{x} - \mathbf{x}^T.\mathbf{b}$$

avec $\mathbf{b}^T = F\,[1,\ 0, \ldots, 0]$. La minimisation de l'énergie potentielle conduit à l'équation d'équilibre suivante :

$$\mathbf{A}.\mathbf{x} = \mathbf{b}$$

- Construire $\mathbf{A}$ pour 10 ressorts reliant 11 points tel que $L(k,1) = k$, $L(k,2) = k+1$.
- Quelle est la largeur de bande de la matrice $\mathbf{A}$?
- Etudier les temps de calcul de la décomposition pour N=10, 100, 1000, 10000.
- Traiter un cas ayant une matrice mal conditionnée.
- Augmenter la largeur de bande de la matrice en rajoutant des ressorts entre les inconnues k et $k+2$, dans le cas $N = 1000$.
- Quel est la durée de la décomposition de cette matrice.
- comment tenir compte du fait que la matrice est creuse ?

Références bibliographiques

[Press94] Numerical Recipies in C, The Art of Scientific Computing, second edition, William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, Cambridge University Press, 1994.

Fascicule r6.02 : Méthodes directes, D. SELIGMANN, J. PELLET, Code Aster, EDF R & D, 2009, `http://www.code-aster.org/V2/doc/v11/fr/man_r/r6/r6.02.01.pdf`