

Introduction à la méthode des éléments finis

Andreas Stahel, traduction par Pierre–André Chevalier

Juin 2004

Résumé

Dans une première section, les principes de la méthode des éléments finis sont introduits à partir d’une structure principale réalisée avec des barres. Les idées de base sont décrites dans [Zien84, p. 26]. Les notions matricielle de rigidité de l’élément, matricielle de rigidité de la structure et contraintes sont expliquées par l’exemple. Dans la deuxième section, on construit la matricielle de rigidité d’une poutre fléchie. Dans la troisième section, on discute plus en détails la structure spéciale des systèmes d’équations linéaires. Les équations formulées sous forme matricielle peuvent être résolues par différentes méthodes. Des comparaisons de quelques-unes de ces méthodes sont données.

Table des matières

1	Structure constituée de barres sous tension et compression	3
1.1	Description des données	3
1.2	Matricielle de rigidité élémentaire	3
1.3	Dérivation de la matricielle de rigidité élémentaire	4
1.4	Calcul de toutes les matricielles de rigidité des éléments	6
1.4.1	élément a du point 1 au point 3	6
1.4.2	élément b du point 1 au point 2	7
1.4.3	élément c de point 2 au point 3	7
1.4.4	élément d du point 3 au point 4	7
1.4.5	élément e du point 2 au point 4	8
1.5	La matricielle de rigidité de la structure	8
1.6	Placement des contraintes et solution des équations	9
1.7	Interprétation des résultats	11
1.8	Octave–Code	11
1.8.1	formulation des matricielles de rigidité d’un élément	11
1.8.2	insérer une matricielle de rigidité dans la matricielle de rigidité global	12
1.8.3	formulation et solution du système d’équations	12
1.8.4	interprétation des résultats	13
2	Matricielle de rigidité d’un élément — matricielle pour un barreau plié	14
2.1	Construction de la matricielle de rigidité élémentaire en situation horizontale	14
2.2	Charges avec tension et compression	16
2.3	Matricielle de rigidité complète d’un barreau horizontal	16
2.4	Rotation d’un barreau en position horizontale	16
3	Matricielles bandes et complexité des calculs, méthodes itératives	19
3.1	Structure de bande et numérotation des noeuds	19
3.1.1	premier essai	19
3.1.2	deuxième essai	20

3.2	Complexité des calculs, algorithme de Cholesky	21
3.2.1	algorithme de Gauss	21
3.2.2	algorithme de Cholesky	23
3.2.3	algorithme de Cholesky pour des matrices bande	24
3.3	Une structure un peu plus réaliste	25
3.4	Méthode frontale de résolution	26
3.5	Comparaison des différentes méthodes	27
3.6	Méthodes itératives	28
3.6.1	L'idée de base	28
3.6.2	La méthode du gradient	28
3.6.3	Méthode du gradient conjugué	30
3.6.4	Un problème modèle	31

Références

- [GoerRoosTobi93] H. Goering, H.-G. Roos, and L. Tobiska. *Finite-Elemente-Methode*. Akademie Verlag, Berlin, dritte edition, 1993.
- [GoluVanLoan89] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, second edition, 1989.
- [LascTheo86] P. Lascaux and R. Théodor. *Analyse numérique matricielle appliquée a l'art de l'ingénieur, Tome 1*. Masson, Paris, 1986.
- [Loga92] D. L. Logan. *A First Course in the Finite Element Method*. PWS-Kent, second edition, 1992.
- [Schw88] H. R. Schwarz. *Finite Element Method*. Academic Press, 1988.
- [VarFEM] A. Stahel. Calculus of Variations and Finite Elements. Lecture Notes used at HTA Biel, 2000.
- [Zien84] O. Zienkiewicz. *Methode der Finiten Elemente*. Carl Hanser Verlag, München, Wien, 1984.

1 Structure constituée de barres sous tension et compression

1.1 Description des données

Dans la figure 1, une structure simple formée de cinq barres (a, b, c, d, e) est représentée. Les barres sont connectées en quatre points de jonction. Toutes les barres sont de même longueur L , de section A et sont constituées d'un matériau de module d'élasticité E . Le point de connection 2 est fixé et le point 1 est mobile seulement dans la direction x . Une force extérieure de 100 N agit au point 4 dans la direction verticale.

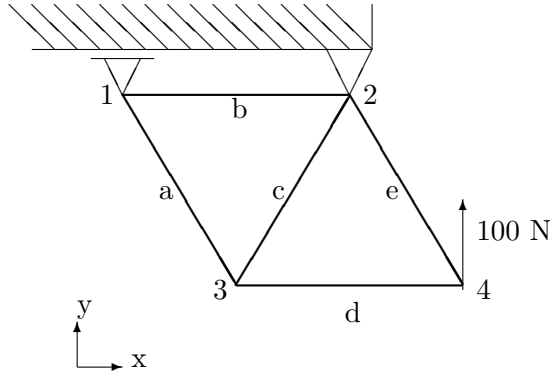


Figure 1: structure des barres sous tension et compression

1.2 Matrice de rigidité élémentaire

Considérons l'élément e, qui va du point i au point n et forme un angle α avec l'axe x . La figure 2 montre la situation généralisée.

Les déplacements des points i et n par rapport à la position originale sont décrits par les vecteurs

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} u_n \\ v_n \end{pmatrix}$$

Il faut maintenant formuler les forces $\vec{F}_{e,i}$ et $\vec{F}_{e,n}$, qui sont les réactions dans les bouts de la barre engendrées par ces vecteurs de déplacement. Dans la structure complète, ces forces sont introduites par les autres barres ou les supports. Les déductions dans la section 1.3 montrent que ces forces sont calculables au moyen de la multiplication des déformations avec **la matrice de rigidité de l'élément K_e** .

$$\begin{aligned} \vec{F}_e &= \begin{pmatrix} \vec{F}_{e,i} \\ \vec{F}_{e,n} \end{pmatrix} = \begin{pmatrix} U_i \\ V_i \\ U_n \\ V_n \end{pmatrix} = & (1) \\ &= \frac{EA}{L} \begin{bmatrix} \cos^2 \alpha & \sin \alpha \cos \alpha & -\cos^2 \alpha & -\sin \alpha \cos \alpha \\ \sin \alpha \cos \alpha & \sin^2 \alpha & -\sin \alpha \cos \alpha & -\sin^2 \alpha \\ -\cos^2 \alpha & -\sin \alpha \cos \alpha & \cos^2 \alpha & \sin \alpha \cos \alpha \\ -\sin \alpha \cos \alpha & -\sin^2 \alpha & \sin \alpha \cos \alpha & \sin^2 \alpha \end{bmatrix} \begin{pmatrix} u_i \\ v_i \\ u_n \\ v_n \end{pmatrix} \end{aligned}$$

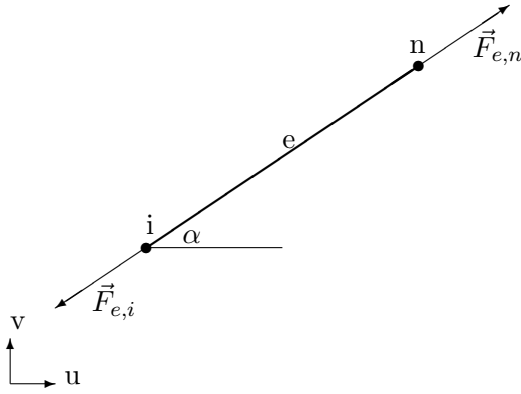


Figure 2: un élément de la structure

La même équation écrite sous une forme courte devient

$$\vec{F}_e = \mathbf{K}_e \vec{a}_e$$

Si les déplacements en fonction des forces sont „petits“, l'angle de l'élément ne changera pratiquement pas. Dans les problèmes avec de grandes déformations, les changements des angles sont à prendre en considération. Dans ce cas la matrice de rigidité de l'élément est à reformuler. Nous ne discuterons pas ces problèmes et nous travaillerons avec la supposition que

Les vecteurs de déformations sont petits comparativement aux dimensions de l'élément. On peut alors admettre que les angles des éléments restent constants.

1.3 Dérivation de la matrice de rigidité élémentaire

Voici une dérivation possible de cette matrice. Elle est basée sur des opérations matricielles (rotation dans le plan) et la règle de Hooke. Cette section n'est pas à lire, si on utilise l'équation (1) sans vérification.

Prenons une barre de longueur L , de surface de section A et constituée d'une matière de module d'élasticité E . Si on allonge cette barre d'une longueur ΔL , la force de réaction est donné par

$$F = \frac{E A}{L} \Delta L$$

Pour une barre horizontale on a $|\Delta L| = |u_i - u_n|$, et alors

$$\begin{aligned} U_i &= \frac{E A}{L} (u_i - u_n) \\ V_i &= 0 \\ U_n &= \frac{E A}{L} (-u_i + u_n) \\ V_n &= 0 \end{aligned}$$

Les mêmes relations peuvent être écrites sous forme matricielle

$$\begin{pmatrix} U_i \\ V_i \\ U_n \\ V_n \end{pmatrix} = \frac{E A}{L} \begin{pmatrix} u_i - u_n \\ 0 \\ -u_i + u_n \\ 0 \end{pmatrix} = \frac{E A}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} u_i \\ v_i \\ u_n \\ v_n \end{pmatrix}$$

C'est la matrice de rigidité de l'élément de équation (1) pour le cas spécial $\alpha = 0$. De ce cas, nous dérivons la règle générale au moyen de rotations.

Si on tourne un vecteur $\begin{pmatrix} u \\ v \end{pmatrix}$ d'un angle α en sens inverse des aiguilles d'une montre, on obtient le nouveau vecteur

$$\begin{pmatrix} \cos \alpha u - \sin \alpha v \\ \sin \alpha u + \cos \alpha v \end{pmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

Une rotation d'angle $-\alpha$ donne une matrice inverse

$$\begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}^{-1} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}$$

Si nous tournons les vecteurs de déformation $\begin{pmatrix} u_i \\ v_i \end{pmatrix}$ et $\begin{pmatrix} u_n \\ v_n \end{pmatrix}$ d'une barre ayant un angle α d'un l'angle $-\alpha$, les vecteurs de forces d'une barre horizontale sont donnés par la rotation des vecteurs $\vec{F}_{e,i}$ et $\vec{F}_{e,n}$, avec le même angle $-\alpha$. Cela nous donne la condition

$$\begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & \cos \alpha & \sin \alpha \\ 0 & 0 & -\sin \alpha & \cos \alpha \end{bmatrix} \begin{pmatrix} U_i \\ V_i \\ U_n \\ V_n \end{pmatrix} = \frac{E A}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & \cos \alpha & \sin \alpha \\ 0 & 0 & -\sin \alpha & \cos \alpha \end{bmatrix} \begin{pmatrix} u_i \\ v_i \\ u_n \\ v_n \end{pmatrix}$$

Et ainsi, la matrice de rigidité est donnée par

$$\mathbf{K}_e = \frac{E A}{L} \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & \cos \alpha & \sin \alpha \\ 0 & 0 & -\sin \alpha & \cos \alpha \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & \cos \alpha & \sin \alpha \\ 0 & 0 & -\sin \alpha & \cos \alpha \end{bmatrix}$$

Par multiplication des matrices, on trouve

$$\mathbf{K}_e = \frac{E A}{L} \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & \cos \alpha & -\sin \alpha \\ 0 & 0 & \sin \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} \cos \alpha & \sin \alpha & -\cos \alpha & -\sin \alpha \\ 0 & 0 & 0 & 0 \\ -\cos \alpha & -\sin \alpha & \cos \alpha & \sin \alpha \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Le résultat est alors

$$\mathbf{K}_e = \frac{E A}{L} \begin{bmatrix} \cos^2 \alpha & \sin \alpha \cos \alpha & -\cos^2 \alpha & -\sin \alpha \cos \alpha \\ \sin \alpha \cos \alpha & \sin^2 \alpha & -\sin \alpha \cos \alpha & -\sin^2 \alpha \\ -\cos^2 \alpha & -\sin \alpha \cos \alpha & \cos^2 \alpha & \sin \alpha \cos \alpha \\ -\sin \alpha \cos \alpha & -\sin^2 \alpha & \sin \alpha \cos \alpha & \sin^2 \alpha \end{bmatrix}$$

La matrice

$$\mathbf{T}(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} \cos \alpha & \sin \alpha \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \cos^2 \alpha & \sin \alpha \cos \alpha \\ \sin \alpha \cos \alpha & \sin^2 \alpha \end{bmatrix}$$

est appelée **matrice de transformation**, et nous pouvons utiliser la notation

$$\mathbf{K}_e = \frac{E A}{L} \begin{bmatrix} \mathbf{T}(\alpha) & -\mathbf{T}(\alpha) \\ -\mathbf{T}(\alpha) & \mathbf{T}(\alpha) \end{bmatrix}$$

pour cette matrice.

1.4 Calcul de toutes les matrices de rigidité des éléments

On applique maintenant les résultats des chapitres ci-dessus sur les 5 éléments de la structure montrée dans la figure 1.

1.4.1 élément a du point 1 au point 3

Cet élément a l'angle $\alpha = -60^\circ = -\pi/3$ alors

$$\cos \alpha = \frac{1}{2} = 0.5 \quad \text{et} \quad \sin \alpha = -\frac{\sqrt{3}}{2} \approx -0.866$$

La matrice de transformation devient

$$\mathbf{T}\left(-\frac{\pi}{3}\right) = \begin{bmatrix} 0.25 & -0.433 \\ -0.433 & 0.75 \end{bmatrix}$$

et ainsi

$$\mathbf{K}_a = \frac{E A}{L} \begin{bmatrix} 0.25 & -0.433 & -0.25 & 0.433 \\ -0.433 & 0.75 & 0.433 & -0.75 \\ -0.25 & 0.433 & 0.25 & -0.433 \\ 0.433 & -0.75 & -0.433 & 0.75 \end{bmatrix}$$

1.4.2 élément b du point 1 au point 2

Cet élément a l'angle $\alpha = 0$ alors

$$\cos \alpha = 1 \quad \text{et} \quad \sin \alpha = 0$$

On obtient alors les matrices

$$\mathbf{T}(0) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$
$$\mathbf{K}_b = \frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

1.4.3 élément c de point 2 au point 3

Cet élément a l'angle $\alpha = -120^\circ = -2\pi/3$ alors

$$\cos \alpha = -\frac{1}{2} = -0.5 \quad \text{et} \quad \sin \alpha = -\frac{\sqrt{3}}{2} \approx -0.866$$

On obtient les matrices

$$\mathbf{T}\left(-\frac{2\pi}{3}\right) = \begin{bmatrix} 0.25 & 0.433 \\ 0.433 & 0.75 \end{bmatrix}$$
$$\mathbf{K}_c = \frac{EA}{L} \begin{bmatrix} 0.25 & 0.433 & -0.25 & -0.433 \\ 0.433 & 0.75 & -0.433 & -0.75 \\ -0.25 & -0.433 & 0.25 & 0.433 \\ -0.433 & -0.75 & 0.433 & 0.75 \end{bmatrix}$$

1.4.4 élément d du point 3 au point 4

Cet élément a l'angle $\alpha = 0$ alors

$$\cos \alpha = 1 \quad \text{et} \quad \sin \alpha = 0$$

On obtient les matrices

$$\mathbf{T}(0) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$
$$\mathbf{K}_d = \frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

1.4.5 élément e du point 2 au point 4

Cet élément a l'angle $\alpha = -60^\circ = -\pi/3$ alors

$$\cos \alpha = \frac{1}{2} = 0.5 \quad \text{et} \quad \sin \alpha = -\frac{\sqrt{3}}{2} \approx -0.866$$

On obtient les matrices

$$\mathbf{T}\left(-\frac{\pi}{3}\right) = \begin{bmatrix} 0.25 & -0.433 \\ -0.433 & 0.75 \end{bmatrix}$$
$$\mathbf{K}_e = \frac{E A}{L} \begin{bmatrix} 0.25 & -0.433 & -0.25 & 0.433 \\ -0.433 & 0.75 & 0.433 & -0.75 \\ -0.25 & 0.433 & 0.25 & -0.433 \\ 0.433 & -0.75 & -0.433 & 0.75 \end{bmatrix}$$

1.5 La matrice de rigidité de la structure

Pour simplifier les calculs, nous travaillerons dans les deux prochaines sections avec

$$\frac{E A}{L} = 1$$

Il faut maintenant composer les cinq matrices de rigidité des éléments en une matrice de rigidité pour toute la structure. Puisque les quatre points du système possèdent chacun deux degrés de liberté, et puisque l'on n'a pas encore introduit les contraintes, on a principalement 8 inconnues.

Commençons avec une matrice remplie de 0

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{pmatrix} = \begin{pmatrix} 0 \\ f_{1,y} \\ f_{2,x} \\ f_{2,y} \\ 0 \\ 0 \\ 0 \\ 100 \end{pmatrix}$$

La première ligne de ce système d'équations linéaires est donnée par l'équilibre des forces des composantes x au point 1. La deuxième ligne de ce système d'équations linéaires est donnée par l'équilibre des forces des composantes y au point 1. La troisième et la quatrième ligne sont données par l'équilibre des forces au point 2, etc... Le vecteur des forces à droite est donné par les forces extérieures. Les forces de support aux points 1 et 2 sont encore inconnues.

Dans une première étape, nous insérons les termes donnés par les éléments a et d . Cela nous

donne le résultat intermédiaire ci-dessous, où on a remplacé les entrées 0 par un point.

$$\begin{bmatrix} 0.2500 & -0.4330 & \cdot & \cdot & -0.2500 & 0.4330 & \cdot & \cdot \\ -0.4330 & 0.7500 & \cdot & \cdot & 0.4330 & -0.7500 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ -0.2500 & 0.4330 & \cdot & \cdot & 1.2500 & -0.4330 & -1.0000 & \cdot \\ 0.4330 & -0.7500 & \cdot & \cdot & -0.4330 & 0.7500 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & -1.0000 & \cdot & 1.0000 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{pmatrix} = \begin{pmatrix} 0 \\ f_{1,y} \\ f_{2,x} \\ f_{2,y} \\ 0 \\ 0 \\ 0 \\ 100 \end{pmatrix}$$

En insérant les équations données par les éléments b , c et e , on trouve finalement

$$\begin{bmatrix} 1.250 & -0.433 & -1.000 & 0.000 & -0.250 & 0.433 & \cdot & \cdot \\ -0.433 & 0.750 & 0.000 & 0.000 & 0.433 & -0.750 & \cdot & \cdot \\ -1.000 & 0.000 & 1.500 & -0.000 & -0.250 & -0.433 & -0.250 & 0.433 \\ 0.000 & 0.000 & -0.000 & 1.500 & -0.433 & -0.750 & 0.433 & -0.750 \\ -0.250 & 0.433 & -0.250 & -0.433 & 1.500 & -0.000 & -1.000 & 0.000 \\ 0.433 & -0.750 & -0.433 & -0.750 & -0.000 & 1.500 & 0.000 & 0.000 \\ \cdot & \cdot & -0.250 & 0.433 & -1.000 & 0.000 & 1.250 & -0.433 \\ \cdot & \cdot & 0.433 & -0.750 & 0.000 & 0.000 & -0.433 & 0.750 \end{bmatrix} \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{pmatrix} = \begin{pmatrix} 0 \\ f_{1,y} \\ f_{2,x} \\ f_{2,y} \\ 0 \\ 0 \\ 0 \\ 100 \end{pmatrix}$$

Sous forme courte, il est possible d'écrire

$$\mathbf{K} \cdot \vec{a} = \vec{F}_{ext}$$

Nous avons maintenant 8 équations linéaires. Mais, au premier coup d'oeil, nous avons 11 inconnues, dont les 8 déplacements et les 3 forces dans les supports. Plus tard, nous utiliserons le fait que $v_1 = u_2 = v_2 = 0$. La **matrice de rigidité de la structure** \mathbf{K} possède des propriétés spéciales:

- elle est symétrique.
- quelques entrées sont 0. Puisque le noeud 1 ne possède pas de connection directe avec le noeud 4, les 2×2 blocs dans le coin en haute à droite et dans le coin en bas à gauche de cette matrice sont 0. Dans des structures plus grandes (plus de barres), cet effet devient plus prononcé, et on a typiquement des matrices avec beaucoup de 0, les termes non nuls étant localisés sur et autour de la diagonale principale.
- la matrice est **définie positive**, ce qui signifie que toutes ses valeurs propres sont plus grandes ou égales à 0.

1.6 Placement des contraintes et solution des équations

Dans la figure 1, il est visible que $v_1 = u_2 = v_2 = 0$. Par conséquent, les entrées de la 2ème, 3ème et 4ème colonne de la matrice ci-dessus sont toujours 0. Nous éliminons ces colonnes. Pour ne pas être obligé de prendre en considération les forces dans les supports, nous pouvons éliminer

aussi les équations sur les lignes 2, 3 et 4. On a donc maintenant 5 équations avec 5 inconnues.

$$\begin{bmatrix} 1.250 & -0.250 & 0.433 & \cdot & \cdot \\ -0.250 & 1.500 & -0.000 & -1.000 & 0.000 \\ 0.433 & -0.000 & 1.500 & 0.000 & 0.000 \\ \cdot & -1.000 & 0.000 & 1.250 & -0.433 \\ \cdot & 0.000 & 0.000 & -0.433 & 0.750 \end{bmatrix} \begin{pmatrix} u_1 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 100 \end{pmatrix}$$

Cette nouvelle matrice hérite de \mathbf{K} les propriétés:

- elle est symétrique.
- quelques entrées sont 0.
- la matrice est **strictement définie positive**, ce qui signifie que toutes ses valeurs propres sont plus grandes que 0.

Ce système peut être résolu numériquement, et le résultat est

$$\begin{pmatrix} u_1 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{pmatrix} = \begin{pmatrix} 28.868 \\ 129.904 \\ -8.3333 \\ 187.639 \\ 241.667 \end{pmatrix}$$

Par conséquent, le vecteur de déformation \vec{a} est

$$\vec{a} = \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{pmatrix} = \begin{pmatrix} 28.868 \\ 0 \\ 0 \\ 0 \\ 129.904 \\ -8.3333 \\ 187.639 \\ 241.667 \end{pmatrix}$$

Si l'on prend en considération que $\frac{EA}{L} \neq 1$, il faut multiplier le vecteur de déformation \vec{a} avec $\frac{L}{EA}$ pour trouver les déformations correctes. Puisque l'on a „supprimé“ dans nos matrices le facteur $\frac{EA}{L}$, on trouve les résultats corrects (c'est-à-dire les forces) si on multiplie des matrices de rigidité avec des vecteurs de déformation.

1.7 Interprétation des résultats

Les résultats ci-dessus \vec{a} pour les déformations sont utilisables dans $\mathbf{K}\vec{a} = \vec{F}_{ext}$, et on trouve

$$\vec{F}_{ext} = \begin{pmatrix} -3.323e-15 \\ 5.000e+01 \\ -1.310e-14 \\ -1.500e+02 \\ 2.842e-14 \\ -2.665e-15 \\ -1.532e-14 \\ 1.000e+02 \end{pmatrix} = \begin{pmatrix} 0 \\ 50 \\ 0 \\ -150 \\ 0 \\ 0 \\ 0 \\ 100 \end{pmatrix}$$

Parallèlement, on peut tester les résultats numériques du système d'équations, et déterminer les forces sur les supports. La réaction (la force) dans le support 1 est une force de 50 N en direction verticale vers le haut.

A l'aide des matrices de rigidité des éléments, on calcule les charges sur les différentes barres. Prenons comme exemple l'élément c . On a l'équation

$$\mathbf{K}_c \begin{pmatrix} u_2 \\ v_2 \\ u_3 \\ v_3 \end{pmatrix} = \vec{F}_c$$

et les calculs donnent

$$\begin{bmatrix} 0.25 & 0.433 & -0.25 & -0.433 \\ 0.433 & 0.75 & -0.433 & -0.75 \\ -0.25 & -0.433 & 0.25 & 0.433 \\ -0.433 & -0.75 & 0.433 & 0.75 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 129.904 \\ -8.3333 \end{pmatrix} = \begin{pmatrix} -28.87 \\ -50.00 \\ 28.87 \\ 50.00 \end{pmatrix} = \begin{pmatrix} U_2 \\ V_2 \\ U_3 \\ V_3 \end{pmatrix}$$

Comme charge sur la barre c , on a une compression de valeur $\sqrt{28.87^2 + 50^2} N = 57.74 N$.

1.8 Octave-Code

1.8.1 formulation des matrices de rigidité d'un élément

Le fichier `BarElementMatrix.m` donne une fonction pour le calcul d'une matrice de rigidité d'une barre.

```

----- BarElementMatrix.m -----
function elmat = BarElementMatrix(Coeff, Angle)
% BarElementMatrix(Coeff, Angle)
% generates the 4x4 Element stiffness matrix, where
% Coeff= E A /L
% Angle = Angle of the bar with respect to the horizontal

c = cos(Angle); s=sin(Angle);
T = Coeff* [ c^2  s*c
            s*c  s^2] ;
elmat = zeros(4);

```

```

elmat(1:2 , 1:2 ) = T ;
elmat(3:4 , 1:2 ) = -T ;
elmat(1:2 , 3:4 ) = -T ;
elmat(3:4 , 3:4 ) = T ;
endfunction

```

1.8.2 insérer une matrice de rigidité dans la matrice de rigidité global

Le fichier `assemble.m` contient une fonction pour insérer une matrice de rigidité d'un élément dans la matrice de rigidité de la structure.

```

----- assemble.m -----
function res = assemble(k,n,Mat,StMat)
% The element stiffnes matrix belonging to the connection from
% point k to point n is included in the global stiffness matrix
k2 = 2*k ; n2 = 2*n;

StMat((k2-1):k2,(k2-1):k2) = StMat((k2-1):k2,(k2-1):k2) + Mat(1:2,1:2);
StMat((k2-1):k2,(n2-1):n2) = StMat((k2-1):k2,(n2-1):n2) + Mat(1:2,3:4);
StMat((n2-1):n2,(k2-1):k2) = StMat((n2-1):n2,(k2-1):k2) + Mat(3:4,1:2);
StMat((n2-1):n2,(n2-1):n2) = StMat((n2-1):n2,(n2-1):n2) + Mat(3:4,3:4);
res = StMat;
endfunction

```

1.8.3 formulation et solution du système d'équations

A l'aide des deux fonctions ci-dessus, nous pouvons formuler les équations pour notre problème et trouver les résultats. Le code est sauvegardé dans le fichier `BarSystem.m`.

```

----- Octave -----
% BarSystem, documentation in the handouts
output_precision = 4 ;

% find the element stiffness matrices
Ka = BarElementMatrix(1,-pi/3);
Kb = BarElementMatrix(1,0);
Kc = BarElementMatrix(1,-2*pi/3);
Kd = BarElementMatrix(1,0);
Ke = BarElementMatrix(1,-pi/3);

% Global stiffness matrix of the correct size
global StMat = zeros(8);

% taking care of all the elements
StMat = assemble(1,3,Ka,StMat)
StMat = assemble(1,2,Kb,StMat)
StMat = assemble(2,3,Kc,StMat)
StMat = assemble(3,4,Kd,StMat)
StMat = assemble(2,4,Ke,StMat)

% Now StMat contains the global stiffness matrix
% remove 2nd, 3rd and 4th rows and columns
mat = StMat([1 5 6 7 8],[1 5 6 7 8]);

% the vector sol contains the unknown displacements of the points
% answer is produced by solving the correct system of equations

```

```

sol = mat\[0 0 0 0 100]';
% Now compute the actual vector with all displacements
a = [sol(1) 0 0 0 sol(2:5)]';
% and find all the forces by a simple matrix multiplication
Fext = StMat * a

```

Ce programme produit les résultats

Octave

```

a =
  28.8675
   0.0000
   0.0000
   0.0000
 129.9038
  -8.3333
 187.6388
 241.6667

Fext =
 -3.323e-15
  5.000e+01
 -1.310e-14
 -1.500e+02
  2.842e-14
 -2.665e-15
 -1.532e-14
  1.000e+02

```

1.8.4 interprétation des résultats

Avec les deux lignes

Octave

```
Kc*[ 0 0 a(5:6)]'
```

on calcule les forces internes de l'élément c et on trouve les résultats

Octave

```

ans =
 -28.87
 -50.00
  28.87
  50.00

```

2 Matrice de rigidité d'un élément — matrice pour un barreau plié

Dans la section précédente, seules les tensions et les compressions pures du barreau étaient permises. Nous voulons maintenant aussi examiner les flexions de barreaux. La construction de la matrice de rigidité d'un élément a été partiellement extraite du livre [Loga92, p. 156].

2.1 Construction de la matrice de rigidité élémentaire en situation horizontale

Considérons un barreau de longueur L placé en situation horizontale. La déviation verticale de ce barreau peut être exprimée par une fonction $y(x)$, où $0 \leq x \leq L$.

En premier lieu, **aucune force horizontale ni déplacement** ne seront pris en compte.

Ecrivons les forces verticales f_y et les moments m en fonction du déplacement y et de l'angle de déplacement, en s'aidant d'une multiplication matricielle

$$\begin{pmatrix} f_y(0) \\ m(0) \\ f_y(L) \\ m(L) \end{pmatrix} = \mathbf{M} \cdot \begin{pmatrix} y(0) \\ y'(0) \\ y(L) \\ y'(L) \end{pmatrix}$$

Pour des petits déplacements et des forces faibles, la relation entre forces/moments et les déplacements est certainement linéaire, ce qui autorise le procédé ci-dessus.

La matrice \mathbf{M} sera appelée par la suite **Matrice de rigidité élémentaire**.

Nous avons maintenant à calculer les 16 coefficients de la 4×4 -matrice \mathbf{M} . Pour cela, nous examinerons 4 situations spéciales, ce qui permettra d'obtenir suffisamment d'équations pour déterminer ces coefficients.

La marche à suivre nécessite l'utilisation de la formule physique fondamentale

$$EI y''(x) = m(x)$$

1. D'abord, on déplace le barreau entier d'une unité vers le haut. Pour la fonction de déplacement simple $y(x) = 1$ les forces et les moments résultants sont tous nuls, et ainsi

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \mathbf{M} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

2. Pour la fonction de déplacement $y(x) = x$, les moments et les forces résultants sont aussi tous nuls, et ainsi

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \mathbf{M} \cdot \begin{pmatrix} 0 \\ 1 \\ L \\ 1 \end{pmatrix}$$

3. Considérons maintenant un barreau encastré à gauche, sur lequel une force 1 est appliquée vers le haut à la position $x = L$:

$$y(0) = y'(0) = 0 \quad \text{und} \quad f_y(L) = 1$$

De cette façon, le moment est donné par

$$m(x) = (L - x)$$

et nous avons les équations suivantes à résoudre

$$\begin{aligned} EI y''(x) &= L - x \\ EI y'(x) &= Lx - \frac{x^2}{2} + 0 \\ EI y(x) &= L\frac{x^2}{2} - \frac{x^3}{6} + 0 \end{aligned}$$

La solution s'obtient par intégration élémentaire et $y(0) = y'(0) = 0$, et elle conduit aux résultats

$$EI y(L) = \frac{L^3}{3}, \quad EI y'(L) = \frac{L^2}{2}, \quad m(0) = +L \quad \text{et} \quad m(L) = 0$$

Puisque la somme de toutes les forces en présence doit s'annuler, alors $f_y(0) = -1$.

$$\begin{pmatrix} -1 \\ +L \\ 1 \\ 0 \end{pmatrix} = \frac{1}{EI} \mathbf{M} \cdot \begin{pmatrix} 0 \\ 0 \\ \frac{L^3}{3} \\ \frac{L^2}{2} \end{pmatrix}$$

4. Considérons maintenant un barreau encastré à droite, sur lequel une force 1 est appliquée en $x = 0$ vers le haut:

$$y(L) = y'(L) = 0 \quad \text{et} \quad f_y(0) = 1$$

Des raisonnements sur la symétrie conduisent à $f_y(L) = -1$ et

$$EI y(0) = \frac{L^3}{3}, \quad EI y'(0) = \frac{-L^2}{2}, \quad m(0) = 0 \quad \text{et} \quad m(L) = -L$$

$$\begin{pmatrix} 1 \\ 0 \\ -1 \\ -L \end{pmatrix} = \frac{1}{EI} \mathbf{M} \cdot \begin{pmatrix} \frac{L^3}{3} \\ \frac{-L^2}{2} \\ 0 \\ 0 \end{pmatrix}$$

En combinant les résultats des 4 cas examinés ci-dessus, on obtient l'équation matricielle

$$EI \begin{bmatrix} 0 & 0 & -1 & 1 \\ 0 & 0 & +L & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -L \end{bmatrix} = \mathbf{M} \cdot \begin{bmatrix} 1 & 0 & 0 & \frac{L^3}{3} \\ 0 & 1 & 0 & \frac{-L^2}{2} \\ 1 & L & \frac{L^3}{3} & 0 \\ 0 & 1 & \frac{L^2}{2} & 0 \end{bmatrix}$$

ou aussi

$$\mathbf{M} = EI \begin{bmatrix} 0 & 0 & -1 & 1 \\ 0 & 0 & +L & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -L \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & \frac{L^3}{3} \\ 0 & 1 & 0 & \frac{-L^2}{2} \\ 1 & L & \frac{L^3}{3} & 0 \\ 0 & 1 & \frac{L^2}{2} & 0 \end{bmatrix}^{-1}$$

$$\begin{aligned}
&= \frac{EI}{L^3} \begin{bmatrix} 0 & 0 & -1 & 1 \\ 0 & 0 & +L & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -L \end{bmatrix} \cdot \begin{bmatrix} -L^3 & \frac{-2L^4}{3} & 2L^3 & \frac{-4L^4}{3} \\ 3L^2 & 2L^3 & -3L^2 & 2L^3 \\ -6 & -4L & 6 & -2L \\ 6 & 2L & -6 & 4L \end{bmatrix} \\
&= \frac{EI}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ -6L & -4L^2 & +6L & -2L^2 \\ -12 & -6L & 12 & -6L \\ -6L & -2L^2 & +6L & -4L^2 \end{bmatrix}
\end{aligned}$$

De cette façon, nous avons trouvé la **matrice de rigidité élémentaires** d'un barreau plié en position horizontale. Néanmoins, nous n'avons pas encore pris en compte des forces et des déplacements horizontaux.

$$\begin{pmatrix} f_y(0) \\ m(0) \\ f_y(L) \\ m(L) \end{pmatrix} = \frac{EI}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ -6L & -4L^2 & +6L & -2L^2 \\ -12 & -6L & 12 & -6L \\ -6L & -2L^2 & +6L & -4L^2 \end{bmatrix} \cdot \begin{pmatrix} y(0) \\ y'(0) \\ y(L) \\ y'(L) \end{pmatrix}$$

2.2 Charges avec tension et compression

Si applique au même barreau des forces f_x en direction horizontale, alors les extrémités $x(0)$ et $x(L)$ du barreau seront déplacés horizontalement. En vertu de la loi de Hook, on a

$$\begin{pmatrix} f_x(0) \\ f_x(L) \end{pmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{pmatrix} x(0) \\ x(L) \end{pmatrix}$$

2.3 Matrice de rigidité complète d'un barreau horizontal

Comme degrés de liberté d'un élément, nous prendrons les déplacements x_1 , x_2 , y_1 et y_2 aux deux extrémités et les déviations de l'angle θ_1 , θ_2 . Pour des petites déviations, $\theta_1 \approx y'(0)$ et $\theta_2 \approx y'(L)$. Nous avons alors

$$\begin{pmatrix} f_x(0) \\ f_y(0) \\ m(0) \\ f_x(L) \\ f_y(L) \\ m(L) \end{pmatrix} = \frac{E}{L^3} \begin{bmatrix} AL^2 & 0 & 0 & -AL^2 & 0 & 0 \\ 0 & 12I & 6LI & 0 & -12I & 6LI \\ 0 & -6LI & -4L^2I & 0 & +6LI & -2L^2I \\ -AL^2 & 0 & 0 & AL^2 & 0 & 0 \\ 0 & -12I & -6LI & 0 & 12I & -6LI \\ 0 & -6LI & -2L^2I & 0 & +6LI & -4L^2I \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \\ \theta_1 \\ x_2 \\ y_2 \\ \theta_2 \end{pmatrix}$$

2.4 Rotation d'un barreau en position horizontale

Si l'on tourne un vecteur $(u, v)^T$ d'un angle α dans le sens inverse aux aiguilles de la montre, on obtient un nouveau vecteur

$$\begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \cdot \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u \cos \alpha - v \sin \alpha \\ u \sin \alpha + v \cos \alpha \end{pmatrix}$$

Puisque la **dévi**ation de l'angle ne change pas, nous pouvons représenter la rotation du barreau par une multiplication matricielle

$$\mathbf{D}(\alpha) \cdot \begin{pmatrix} x_1 \\ y_1 \\ \theta_1 \\ x_2 \\ y_2 \\ \theta_2 \end{pmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & 0 & 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \\ \theta_1 \\ x_2 \\ y_2 \\ \theta_2 \end{pmatrix}$$

De même, le vecteur des forces et des moments peut aussi être tourné par multiplication:

$$\mathbf{D}(\alpha) \cdot \begin{pmatrix} f_x(0) \\ f_y(0) \\ m(0) \\ f_x(L) \\ f_y(L) \\ m(L) \end{pmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & 0 & 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} f_x(0) \\ f_y(0) \\ m(0) \\ f_x(L) \\ f_y(L) \\ m(L) \end{pmatrix}$$

Considérons maintenant un barreau oblique qui forme avec l'axe x un angle α . Si l'on tourne les déplacements et les forces de l'angle, on obtient alors un barreau horizontal et on peut utiliser la matrice de rigidité de la section précédente. On a:

$$\mathbf{D}(-\alpha) \begin{pmatrix} f_x(0) \\ f_y(0) \\ m(0) \\ f_x(L) \\ f_y(L) \\ m(L) \end{pmatrix} = \frac{E}{L^3} \begin{bmatrix} AL^2 & 0 & 0 & -AL^2 & 0 & 0 \\ 0 & 12I & 6LI & 0 & -12I & 6LI \\ 0 & -6LI & -4L^2I & 0 & +6LI & -2L^2I \\ -AL^2 & 0 & 0 & AL^2 & 0 & 0 \\ 0 & -12I & -6LI & 0 & 12I & -6LI \\ 0 & -6LI & -2L^2I & 0 & +6LI & -4L^2I \end{bmatrix} \mathbf{D}(-\alpha) \cdot \begin{pmatrix} x_1 \\ y_1 \\ \theta_1 \\ x_2 \\ y_2 \\ \theta_2 \end{pmatrix}$$

ou aussi

$$\begin{pmatrix} f_x(0) \\ f_y(0) \\ m(0) \\ f_x(L) \\ f_y(L) \\ m(L) \end{pmatrix} = \frac{E}{L^3} \mathbf{D}(\alpha) \begin{bmatrix} AL^2 & 0 & 0 & -AL^2 & 0 & 0 \\ 0 & 12I & 6LI & 0 & -12I & 6LI \\ 0 & -6LI & -4L^2I & 0 & +6LI & -2L^2I \\ -AL^2 & 0 & 0 & AL^2 & 0 & 0 \\ 0 & -12I & -6LI & 0 & 12I & -6LI \\ 0 & -6LI & -2L^2I & 0 & +6LI & -4L^2I \end{bmatrix} \mathbf{D}(-\alpha) \begin{pmatrix} x_1 \\ y_1 \\ \theta_1 \\ x_2 \\ y_2 \\ \theta_2 \end{pmatrix}$$

Nous avons utilisé ici le fait que

$$\mathbf{D}(\alpha)^{-1} = \mathbf{D}(-\alpha) = \mathbf{D}(\alpha)^T = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & 0 & 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

De cette façon, la matrice de rigidité élémentaire M d'un barreau plié d'un angle α peut être calculée:

$$\mathbf{M} = \frac{E}{L^3} \mathbf{D}(\alpha) \cdot \begin{bmatrix} AL^2 & 0 & 0 & -AL^2 & 0 & 0 \\ 0 & 12I & 6LI & 0 & -12I & 6LI \\ 0 & -6LI & -4L^2I & 0 & +6LI & -2L^2I \\ -AL^2 & 0 & 0 & AL^2 & 0 & 0 \\ 0 & -12I & -6LI & 0 & 12I & -6LI \\ 0 & -6LI & -2L^2I & 0 & +6LI & -4L^2I \end{bmatrix} \cdot \mathbf{D}(-\alpha)$$

On économisera ici cette multiplication matricielle, pour ne pas donner une expression encore plus compliquée. *Mathematica* ou *Octave* peut calculer très facilement cette expression. Pour obtenir \mathbf{M} , nous devons seulement connaître les données E , I , A , L et α .¹ La matrice de rigidité d'un barreau plié vous trouver dans le livre [Loga92, p. 197]. On a trouver la matrice de rigidité d'un barreau plié sous tension et compression. Cette matrice élémentaire est symétrique et strictement définie positive. Avec ce résultat considerer des structures avec der barreau pliés.

¹Si l'on pose $\cos \alpha = \text{Co}$ et $\sin \alpha = \text{Si}$, alors la matrice est donnée

$$\frac{E}{L^3} \begin{bmatrix} A \text{Co}^2 L^2 + 12I \text{Si}^2 & -\text{CoSi}(12I - AL^2) & -6IL \text{Si} & -A \text{Co}^2 L^2 - 12I \text{Si}^2 & -\text{CoSi}(-12I + AL^2) & -6IL \text{Si} \\ -\text{CoSi}(12I - AL^2) & 12\text{Co}^2 I + AL^2 \text{Si}^2 & 6\text{Co} IL & -\text{CoSi}(-12I + AL^2) & -12\text{Co}^2 I - AL^2 \text{Si}^2 & 6\text{Co} IL \\ +6IL \text{Si} & -6\text{Co} IL & -4IL^2 & -6IL \text{Si} & +6\text{Co} IL & -2IL^2 \\ -A \text{Co}^2 L^2 - 12I \text{Si}^2 & -\text{CoSi}(-12I + AL^2) & 6IL \text{Si} & A \text{Co}^2 L^2 + 12I \text{Si}^2 & -\text{CoSi}(12I - AL^2) & 6IL \text{Si} \\ -\text{CoSi}(-12I + AL^2) & -12\text{Co}^2 I - AL^2 \text{Si}^2 & -6\text{Co} IL & -\text{CoSi}(12I - AL^2) & 12\text{Co}^2 I + AL^2 \text{Si}^2 & -6\text{Co} IL \\ +6IL \text{Si} & -6\text{Co} IL & -2IL^2 & -6IL \text{Si} & +6\text{Co} IL & -4IL^2 \end{bmatrix}$$

3 Matrices bandes et complexité des calculs, méthodes itératives

Dans ce chapitre, vous apprendrez quelques problèmes typiques que l'on trouve en résolvant des systèmes d'équations linéaires. Une présentation plus détaillée n'est pas possible dans un temps si court. Le but est de vous montrer que les notions **matrice bande**, **numérotation des noeuds** et **méthode frontale** ont une influence décisive sur votre travail avec les éléments finis: un choix „mauvais“ des numérotations des noeuds peut faire la différence entre un problème vite calculé et un problème apparemment insoluble. Quelques aspects techniques sont formulés de façon volontairement très vague. Les effets importants sont bien visibles avec ces arguments formels. Pour plus des détails, voyez la littérature (exemple [GoerRoosTobi93], [GoluVanLoan89], [Schw88], [LascTheo86]).

Heureusement, beaucoup de systèmes d'équations qu'on trouve en FEM ont les mêmes propriétés spéciales. Pour la matrice \mathbf{A} dans le système

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

on a:

1. elle est symétrique, c'est-à-dire $a_{i,j} = a_{j,i}$.
2. elle a une structure de bande, c'est-à-dire les termes non nuls de la matrice se trouvent dans une bande de largeur $2b$ (largeur de bande) concentrée sur la diagonale principale de la matrice
3. elle est strictement définie positive, et donc

$$\vec{x}^T \cdot \mathbf{A} \vec{x} > 0 \quad \text{pour tous les vecteurs } \vec{x} \neq \vec{0}$$

Cette propriété se base dans la plupart des cas sur la règle que l'énergie potentielle est à minimiser. Cela nous donne la relation entre les éléments finis et le **calcul des variations**.

Il faut maintenant tirer profit de cette propriété. Puisque la matrice \mathbf{A} est définie positive, on a moins de problèmes numériques dans la solution des systèmes d'équations:

- dans l'algorithme de Gauss, il ne faut pas échanger de lignes (pas de nécessité de choisir des pivots). Dans ce cas on peut utiliser l'algorithme de Cholesky, car il n'y a jamais à calculer la racine d'un nombre négatif.
- les algorithmes de Gauss et Cholesky sont **stables**, cela signifie qu'une petite faute d'arrondi ne donne pas une grande faute dans les solutions.

La deuxième affirmation est très utile, parce qu'on peut exclure des grandes fautes dans les calculs des équations linéaires en utilisant ces algorithmes.

3.1 Structure de bande et numérotation des noeuds

3.1.1 premier essai

Regardons une structure plus complexe constituée de 15 points de connection (noeuds) et 27 barres (éléments). Puisque chaque noeud possède deux degrés de liberté, la matrice de rigidité de la structure \mathbf{K} devient une matrice 30×30 , qui est symétrique. Essayons maintenant de trouver, parmi les 900 termes de cette matrice, ceux qui ne sont pas nuls. Si nous partageons la matrice en blocs de taille 2×2 nous avons une structure de taille 15×15 , chaque terme consistant en une matrice 2×2 . Un de ces termes est différent de zéro si un élément (barre) fait une connection entre les noeuds de numéros correspondants. Pour qu'on puisse le déterminer, il faut numérotter les noeuds de la structure. Une variante possible est donnée dans la figure 3.

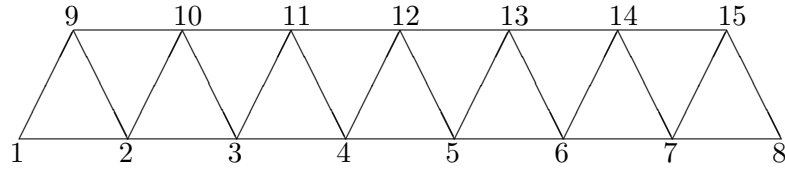


Figure 3: grande structure constituée de barres sous tension et pression, première numérotation

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
1	x	x	x
2	x	x	x	x	x
3	.	x	x	x	x	x
4	.	.	x	x	x	x	x	.	.	.
5	.	.	.	x	x	x	x	x	.	.
6	x	x	x	x	x	.
7	x	x	x	x	x
8	x	x	x
9	x	x	x	x
10	.	x	x	x	x	x
11	.	.	x	x	x	x	x	.	.	.
12	.	.	.	x	x	x	x	x	.	.
13	x	x	x	x	x	.
14	x	x	x	x	x
15	x	x	x	x

La matrice de rigidité d'après la numérotation de la figure 3 est une matrice symétrique, de largeur de bande $b = 18$. Si on veut résoudre le système d'équations grâce à l'algorithme de Cholesky, il faut remplir les bandes „vides“ entre les termes non nuls avec des 0, parce que pendant le calcul ces termes sont remplacés par des composantes non nulles. Les bandes extérieures ne sont pas utilisées pendant le calcul de la solution, et il n'est pas nécessaire de les sauver en mémoire. On peut estimer qu'on a besoin approximativement de $2 \cdot 18 \cdot 30 = 1080$ places en mémoire. Puisque la matrice est symétrique, et que cette symétrie est conservée durant toute l'exécution de l'algorithme de Cholesky, il suffit de réserver à peu près 540 places en mémoire.

3.1.2 deuxième essai

Une autre possibilité de numérotation de la même structure est montrée dans la figure 4 .

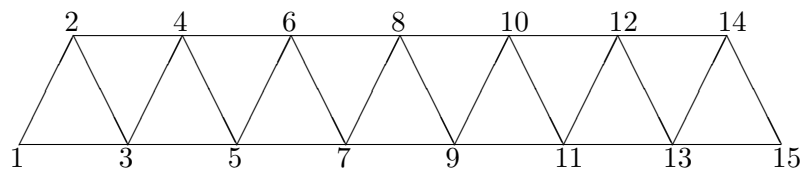


Figure 4: grande structure constituée de barres sous tension et pression, deuxième numérotation

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
1	x	x	x
2	x	x	x	x
3	x	x	x	x	x
4	.	x	x	x	x	x
5	.	.	x	x	x	x	x
6	.	.	.	x	x	x	x	x
7	x	x	x	x	x
8	x	x	x	x	x
9	x	x	x	x	x
10	x	x	x	x	x	.	.	.
11	x	x	x	x	x	.	.
12	x	x	x	x	x	.
13	x	x	x	x	x
14	x	x	x	x
15	x	x	x

La matrice de rigidité de la structure avec la numérotation donnée dans la figure 4 est une matrice symétrique, de largeur de bande 10 (demilargeur $b = 6$). Pour la solution de ce système d'équations au moyen de l'algorithme de Cholesky, on a besoin d'à peu près $10 \cdot 30/2 = 150$ places en mémoire.

3.2 Complexité des calculs, algorithme de Cholesky

Nous présentons ici un petit résumé des points principaux. Une présentation plus détaillée est donnée dans le chapitre 2 du livre [GoerRoosTobi93].

Lorsqu'on travaille avec les éléments finis, on obtient très souvent des systèmes linéaires de la forme

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

ou \mathbf{A} est une **matrice symétrique, strictement définie positive** de taille $n \times n$, de **largeur de demie-bande** b . Il faut maintenant profiter au maximum des propriétés spéciales de ce type de matrice, afin d'améliorer la précision et économiser le temps de calcul. Pour cela il faut chercher et utiliser un algorithme spécialisé.

3.2.1 algorithme de Gauss

Représentons le système d'équations

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & \dots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & \dots & a_{3,n} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & a_{n,4} & \dots & a_{n,n} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix}$$

sous forme d'une matrice augmentée

$$\left[\begin{array}{cccccc|c} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & \dots & a_{1,n} & b_1 \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & \dots & a_{2,n} & b_2 \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & \dots & a_{3,n} & b_3 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & a_{n,4} & \dots & a_{n,n} & b_n \end{array} \right]$$

Dans l'algorithme classique de Gauss, on détermine, grâce à la première équation, x_1 en fonction de x_2, \dots, x_n et b_1 , et on place le résultat dans toutes les autres équations pour éliminer x_1 . Cela donne un **nouveau**, système d'équations, qui est équivalent et de la forme

$$\left[\begin{array}{cccccc|c} r_{1,1} & r_{1,2} & r_{1,3} & r_{1,4} & \dots & r_{1,n} & y_1 \\ 0 & a_{2,2} & a_{2,3} & a_{2,4} & \dots & a_{2,n} & b_2 \\ 0 & a_{3,2} & a_{3,3} & a_{3,4} & \dots & a_{3,n} & b_3 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n,2} & a_{n,3} & a_{n,4} & \dots & a_{n,n} & b_n \end{array} \right]$$

Attention au fait que **les valeurs** changent, c'est-à-dire que la „nouvelle“ valeur de $a_{2,2}$ n'est pas égale à „l'ancienne“ valeur de $a_{2,2}$. Au prochain pas, on élimine à l'aide de la nouvelle deuxième équation l'inconnue x_2 de toutes les autres équations. Après ce pas, on obtient la structure

$$\left[\begin{array}{cccccc|c} r_{1,1} & r_{1,2} & r_{1,3} & r_{1,4} & \dots & r_{1,n} & y_1 \\ 0 & r_{2,2} & r_{2,3} & r_{2,4} & \dots & r_{2,n} & y_2 \\ 0 & 0 & a_{3,3} & a_{3,4} & \dots & a_{3,n} & b_3 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & a_{n,3} & a_{n,4} & \dots & a_{n,n} & b_n \end{array} \right]$$

Après n pas d'élimination, on a le système d'équations sous la forme

$$\left[\begin{array}{cccccc|c} r_{1,1} & r_{1,2} & r_{1,3} & r_{1,4} & \dots & r_{1,n-1} & r_{1,n} & y_1 \\ 0 & r_{2,2} & r_{2,3} & r_{2,4} & \dots & r_{2,n-1} & r_{2,n} & y_2 \\ 0 & 0 & r_{3,3} & r_{3,4} & \dots & r_{3,n-1} & r_{3,n} & y_3 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & r_{n-1,n-1} & r_{n,n} & y_{n-1} \\ 0 & 0 & 0 & 0 & \dots & 0 & r_{n,n} & y_n \end{array} \right]$$

Si on regarde ces calculs soigneusement, on voit que la matrice \mathbf{A} est réécrite comme un produit de deux matrices

$$\mathbf{A} = \mathbf{L} \cdot \mathbf{R}$$

où

$$\mathbf{R} = \left[\begin{array}{cccccc|c} r_{1,1} & r_{1,2} & r_{1,3} & r_{1,4} & \dots & r_{1,n-1} & r_{1,n} \\ 0 & r_{2,2} & r_{2,3} & r_{2,4} & \dots & r_{2,n-1} & r_{2,n} \\ 0 & 0 & r_{3,3} & r_{3,4} & \dots & r_{3,n-1} & r_{3,n} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & r_{n-1,n-1} & r_{n,n} \\ 0 & 0 & 0 & 0 & \dots & 0 & r_{n,n} \end{array} \right]$$

et

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ l_{2,1} & 1 & 0 & 0 & \dots & 0 & 0 \\ l_{3,1} & l_{3,2} & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ l_{n-1,1} & l_{n-1,2} & l_{n-1,3} & l_{n-1,4} & \dots & 1 & 0 \\ l_{n,1} & l_{n,2} & l_{n,3} & l_{n,4} & \dots & l_{n,n-1} & 1 \end{bmatrix}$$

Au lieu du système

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

on résout

$$\mathbf{L} \vec{y} = \vec{b} \quad \text{puis} \quad \mathbf{R} \vec{x} = \vec{y}$$

On a supposé au départ les conditions que les équations sont solubles sans ambiguïté, et que pendant les calculs il n'arrive jamais de division par 0 (mot-clé: stratégie des pivots). Si \mathbf{A} est une matrice symétrique et strictement définie positive, ces conditions sont satisfaites.

Pour l'estimation de la complexité des calculs, on considère qu'une addition et une multiplication sont des opérations typiques. Ainsi, on estime pour l'algorithme classique de Gauss une complexité de calculs de

$$G(n) \approx \frac{1}{3} n^3$$

opérations typiques pour la solution d'un système de n équations, pour lequel il faut mettre en mémoire n^2 nombres.

3.2.2 algorithme de Cholesky

Cet algorithme est comparable avec l'algorithme de Gauss, mais il tire profit de la symétrie de la matrice. L'algorithme calcule, à partir de la matrice \mathbf{A} qui est symétrique et définie positive, une matrice triangulaire supérieure

$$\mathbf{R} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & r_{1,4} & \dots & r_{1,n-1} & r_{1,n} \\ 0 & r_{2,2} & r_{2,3} & r_{2,4} & \dots & r_{2,n-1} & r_{2,n} \\ 0 & 0 & r_{3,3} & r_{3,4} & \dots & r_{3,n-1} & r_{3,n} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & r_{n-1,n-1} & r_{n,n} \\ 0 & 0 & 0 & 0 & \dots & 0 & r_{n,n} \end{bmatrix}$$

avec

$$\mathbf{A} = \mathbf{R}^T \cdot \mathbf{R}$$

Le système d'équations peut être résolu par

$$\mathbf{R}^T \vec{y} = \vec{b} \quad \text{puis} \quad \mathbf{R} \vec{x} = \vec{y}$$

Grâce à la symétrie, la quantité de calculs se divise en deux, avec

$$C(n) = \frac{1}{6} n^3,$$

et on a besoin de $\frac{1}{2} n^2$ places en mémoire.

Voici une réalisation possible de l'algorithme de Cholesky (en *Octave*).

```

function RES = cholesky(A)
% RES = cholesky(A) if A is a symmetric positive definite matrix
%           returns a lowertriangular matrix RES such that
%           A = RES * RES'
[n,m] = size(A);
if (n!=m) error ("cholesky: matrix has to be square ") endif
for k = 1:n-1
  if ( A(k,k) <= 0) error ("cholesky: matrix not positive definite") endif
  A(k,k) = sqrt(A(k,k));
  A(k+1:n,k) = A(k+1:n,k)/A(k,k);
  for j = k+1:n
    A(j:n,j) = A(j:n,j) - A(j:n,k)*A(j,k);
  endfor
endfor
if ( A(n,n) <= 0) error ("cholesky: matrix not positive definite") endif
A(n,n) = sqrt(A(n,n));
RES = tril(A);
endfunction

```

3.2.3 algorithme de Cholesky pour des matrices bande

Nous présupposons que la matrice \mathbf{A} ($n \times n$) est symétrique, strictement définie positive, de demi largeur de bande b . Nous présupposons aussi que

$$1 \ll b \ll n$$

ce qui signifie que la largeur de bande est beaucoup plus petit que la largeur totale de la matrice. La matrice \mathbf{R} de la décomposition de Cholesky

$$\mathbf{A} = \mathbf{R}^T \cdot \mathbf{R}$$

est produite par des calculs similaires à ceux de l'algorithme de Gauss. La matrice \mathbf{R} hérite la structure de bande de \mathbf{A} et a la demie largeur de bande b .

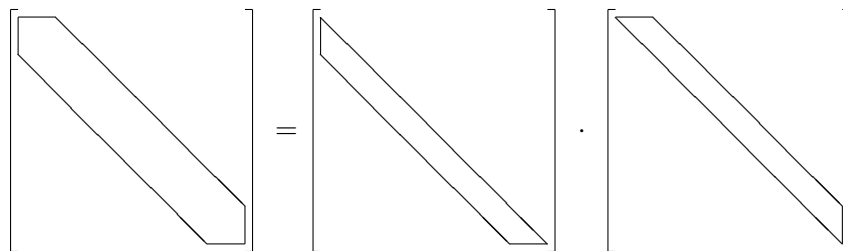


Figure 5: Algorithme de Cholesky pour des matrices bande

Pour le calcul de la solution de l'équation

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

il faut effectuer les pas suivants:

1. calculer la décomposition

$$\mathbf{A} = \mathbf{R}^T \cdot \mathbf{R}$$

2. résoudre le système d'équations

$$\mathbf{R}^T \vec{y} = \vec{b}$$

pour déterminer \vec{y} . Ce système d'équations se laisse résoudre dans le style élémentaire „de haut en bas“.

3. résoudre le système d'équations

$$\mathbf{R} \vec{x} = \vec{y}$$

pour déterminer \vec{x} . Ce système d'équations se laisse résoudre dans le style élémentaire „de bas en haut“.

A cause de

$$\mathbf{A} \vec{x} = \mathbf{R}^T \cdot (\mathbf{R} \vec{x}) = \mathbf{R}^T \vec{y} = \vec{b}$$

on a trouvé la solution du problème initial.

Puisque que \mathbf{A} est symétrique, il faut mémoriser approximativement $n \cdot b$ nombres, et \mathbf{R} occupe la même quantité de mémoire. Une analyse plus approfondie nous montre que pendant l'exécution des calculs les entrées de \mathbf{A} peuvent être remplacées par les entrées de \mathbf{R} . Il n'est donc pas nécessaire de mémoriser les deux matrices en même temps. On peut alors réduire les **places dont on a besoin en mémoire** à environ $n \cdot b$ nombres. Estimons encore la quantité d'opérations typiques dont on a besoin pour les trois étapes du calcul.

1. Pour l'élimination de x_1 , à l'aide de la première équation, de la deuxième jusqu'à la b -ième équation, on a besoin d'environnement b^2 d'opérations. Avec l'aide de la nouvelle deuxième équation on élimine x_2 de la troisième jusqu'à la $(b + 1)$ -ième équation et on a besoin environ b^2 opérations. Nous sommes obligés de faire à peu près n pas d'élimination, chacun avec b^2 opérations. Au total, on a à faire $n \cdot b^2$ opérations élémentaires. Une analyse plus approfondie nous montre qu'à cause de la symétrie on a besoin de seulement $n \cdot b^2/2$ opérations.
2. Chacun des n pas pour trouver le \vec{y} (en utilisant le \vec{b}) nécessite b opérations. Cela nous coûte un effort total de $n \cdot b$ opérations. Puisque $1 \ll b \ll n$, cet effort est négligeable en comparaison des $n \cdot b^2/2$ opérations.
3. La quantité d'opérations dont on a besoin pour le dernier pas est négligeable.

La **complexité totale des calculs** est donc de $n \cdot b^2/2$ opérations.

3.3 Une structure un peu plus réaliste

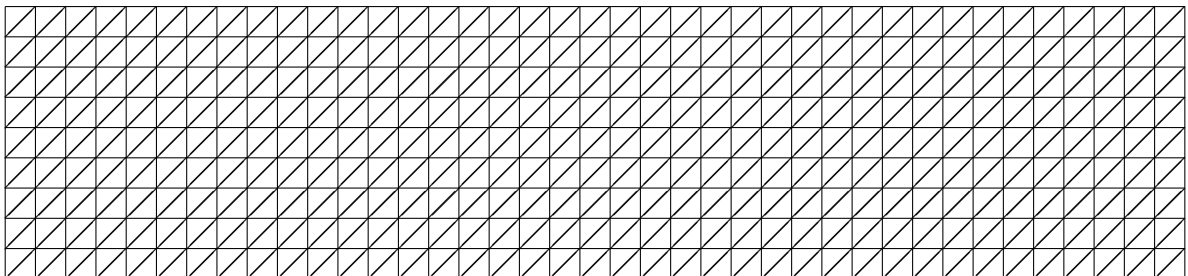


Figure 6: Une „petite“ structure de barres de tension et compression

Pour que vous ne commenciez pas à compter, je vous donne comme information qu'on trouve sur chaque ligne de la structure 40 noeuds et au total 10 lignes de noeuds. Le système complet a

donc 800 degrés de liberté, et on a besoin de stocker une matrice de taille 800×800 en mémoire. Ces 640'000 nombres en mémoire sont déjà un problème pour certains systèmes d'exploitation. On est obligé de profiter mieux de la structure de bande et de la symétrie de la matrice. Le but, dans la suite est de trouver une numérotation „optimale“. En principe il existe deux stratégies de numérotation:

1. **par ligne**: on commence à gauche en bas avec la première ligne, puis deuxième ligne, . . . , toujours de gauche à droite.
2. **par colonne**: on commence à gauche en bas avec la première colonne, puis deuxième colonne, . . . , toujours de bas en haut

La structure de la matrice de rigidité deviendra différente. Comparez dans le tableau suivant

numérotation	par ligne	par colonne
largeur	800×800	800×800
largeur de demie bande	$2 \cdot 41 + 2 = 84$	$2 \cdot 11 + 2 = 24$
places en mémoire (quantité)	74'400	18'400
nombres d'instructions avec Gauss ($n^3/3$)	$170 \cdot 10^6$	$170 \cdot 10^6$
nombres d'instructions avec Cholesky ($n^3/6$)	$85 \cdot 10^6$	$85 \cdot 10^6$
nombres d'instructions avec Cholesky et structure de bande ($n \cdot b^2/2$)	$2.8 \cdot 10^6$	$0.21 \cdot 10^6$

La numérotation de la structure par colonnes a besoin d'un temps de calcul qui est à peu près 13 fois plus petit qu'avec la numérotation par lignes. Dans des structures plus grandes, ce facteur devient considérablement plus grand.

3.4 Méthode frontale de résolution

Une simple structure en trois dimensions faite de barres avec $20 \times 20 \times 10$ points de connection à 4'000 noeuds, cela donne $n = 12'000$ degrés de liberté. La matrice de rigidité correspondante a une largeur de 12000×12000 et une demie largeur de bande de $b = 600$. Si nous calculons en comptant 10 bytes pour chaque nombre, cette matrice nécessite une place en mémoire de $10 \times 12000 \times 12000$ Byte, ou 1.44 GB. Si on tire avantage de la structure de bande et de la symétrie, la quantité de mémoire nécessaire diminue à $10 \times 600 \times 12000$ Byte, ou 72 MB. Aujourd'hui, ce n'est plus un problème d'avoir cette place sur un disque dur, mais chaque ordinateur ne possède pas 72 MB de RAM. Il est donc avantageux d'utiliser un algorithme qui n'a pas besoin d'avoir toujours toute la matrice en mémoire. L'algorithme de Cholesky permet de travailler avec seulement b^2 nombres en mémoire principale. À l'aide de symétrie on a besoin de seulement $b^2/2$ nombres. Dans ce cas, nous avons besoin pour notre exemple d'une place en mémoire de seulement 1.8 MB. Ceci est aujourd'hui (1997) possible sur un PC normal.

Dans la méthode frontale pour résoudre la matrice de rigidité de la structure, on calcule seulement jusqu'à ce qu'il soit possible d'éliminer la première ligne dans la région active. Puis cette région active est déplacée, et les éléments qui manquent sont calculés. On recommence alors à éliminer la ligne la plus haute.

1. Dans une première étape tous les éléments de la matrice dans la **région active** en haut à gauche sont calculés. Puis on exécute les pas décrits dans l'algorithme de Cholesky pour la première ligne.
2. On peut maintenant écrire la première ligne de cette matrice réduite (partiellement) sur le disque dur. Puis on exécute les pas correspondants de la réduction pour la deuxième ligne.

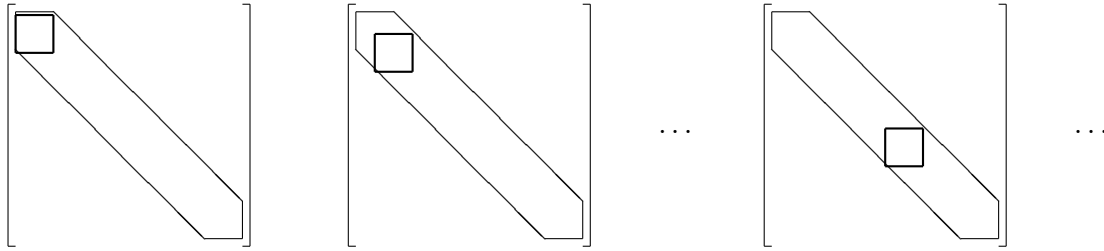


Figure 7: Méthode frontale

3. Après chaque pas, on déplace la région active une ligne en bas à droite.
4. Après n pas dans ce style, la matrice \mathbf{R} de la réduction de $\mathbf{A} = \mathbf{R}^T \cdot \mathbf{R}$ se trouve sur la mémoire extérieure.
5. La place en mémoire dont on a besoin pour la solution des équations

$$\mathbf{R}^T \vec{y} = \vec{b} \quad \text{et} \quad \mathbf{R} \vec{x} = \vec{y}$$

est minimale.

Pendant la résolution d'un problème FEM, des changements entre production de la matrice de coefficients et l'élimination des inconnues sont toujours nécessaires, et cela donne une structure de programme plus complexe. Ce désavantage est moins grave que l'avantage d'avoir besoin de beaucoup moins de mémoire du type RAM.

Computer	FLOP par seconde
NeXT (68040/25MHz)	1.0 M
HP 710/100	11.5 M
SUN Sparc ULTRA 10 (440MHz)	50.0 M
Pentium III 800 (hors cache)	50.0 M
Pentium III 800 (en cache)	185 M
Pentium 4 2.6 GHz (hors cache)	370 M
Pentium 4 2.6 GHz (en cache)	450 M
Intel I7-920, 2.6 GHz	700 M
Intel Haswell 5930 3.5 GHz	2'000 M

Tableau 1: Puissance de quelques processeurs

3.5 Comparaison des différentes méthodes

Illustrons les réflexions du dernier chapitre par des exemples. Nous estimons pour ces exemples la place en mémoire dont on a besoin et le temps de calcul. Il y a ci-dessous un tableau 1 comparant les performances des différents ordinateurs de notre école. Dans un petit programme de test, on produit une matrice qui est symétrique et définie positive. Puis on mesure le temps que l'algorithme de Cholesky nécessite pour le calcul de la solution.

La structure tri-dimensionnelle dans le chapitre 3.4 donne une matrice de taille 12000×12000 , avec une largeur de demi-bande de 600. Pour chaque nombre on a besoin de 10 bytes de place en

mémoire. Trouver les résultats en tableau 2. Les estimations pour les temps sont faits seulement pour la décomposition de Cholesky. Mais les mêmes effets qualitatifs sont visibles si on considère tous les aspects dans une solution FEM. Le lecteur peut tirer lui-même les conclusions correctes.

méthode	Gauss toute la matrice	Cholesky avec structure de bande	méthode frontale
taille de la matrice	12000 × 12000	12000 × 600	12000 × 600
mémoire RAM	1440 MB	72 MB	1.8 MB
place sur le disque	1440 MB	72 MB	72 MB
nombre d'opérations	$n^3/3$ $576 \cdot 10^9$	$nb^2/2$ $2.2 \cdot 10^9$	$nb^2/2$ $2.2 \cdot 10^9$
temps de calcul	$5.76 \cdot 10^5$ sec	$2.2 \cdot 10^3$ sec	$2.2 \cdot 10^3$ sec
NeXT	7 jours	36 min	36 min
HP710	14 h	3.2 min	3.2 min
Pentium III 800	3.2 h	45 sec	45 sec
Pentium 4 2.6 GH	0.5 h	6 sec	6 sec

Tableau 2: effort de calcul pour Gauss, Cholesky et pour la méthode frontale

3.6 Méthodes itératives

La matrice de rigidité de la structure donnée dans le chapitre 3.4 a seulement 30 nombres différents de zéro par ligne. La structure de bande a besoin au total de 600 entrées. Pour une optimisation plus approfondie, on peut utiliser pour la résolution des système d'équations de ce type des **méthodes itératives**. L'étude de ces algorithmes n'entre pas le thème de ce cours, mais voici seulement quelques mots-clé: algorithme de Jacobi, Gauss–Seidel, méthode des gradients conjugués préconditionnés. Vous pourrez trouver des détails dans [VarFEM], ou dans beaucoup de livres sur l'analyse numérique.

3.6.1 L'idée de base

Pour des problèmes FEM il faut résoudre des systèmes $\mathbf{A}\vec{x} + \vec{b} = \vec{0}$ avec des matrices \mathbf{A} symétriques et définies positives. Il s'agit des matrices **creuses**, ce qui veut dire que seulement très peu des nombres sont différents de zéro. L'effort de calcul pour multiplier cette matrice avec un vecteur est donc très petit, par rapport à l'effort pour résoudre le système. Quelques méthodes itératives cherchent des solutions approximatives à l'aide de nombreuses multiplications.

3.6.2 La méthode du gradient

Pour une matrice \mathbf{A} définie positive et symétrique la solution de $\mathbf{A}\vec{x} + \vec{b} = \vec{0}$ correspond au minimum de la fonction

$$f(\vec{x}) = \frac{1}{2} \langle \vec{x}, \mathbf{A}\vec{x} \rangle + \langle \vec{x}, \vec{b} \rangle$$

avec le gradient

$$\nabla f(\vec{x}) = \mathbf{A}\vec{x} + \vec{b}$$

Dans la figure 8, on voit le graphe et quelques courbes de niveau d'une telle fonction. Pour chaque point \vec{x} la fonction $f(\vec{x})$ diminue le plus rapidement possible dans la direction du gradient

négative. Nous cherchons donc le minimum suivant la droite passant par le point \vec{x} et dans la direction du gradient. Après quelques calculs, on trouve l'algorithme dans le tableau 3.

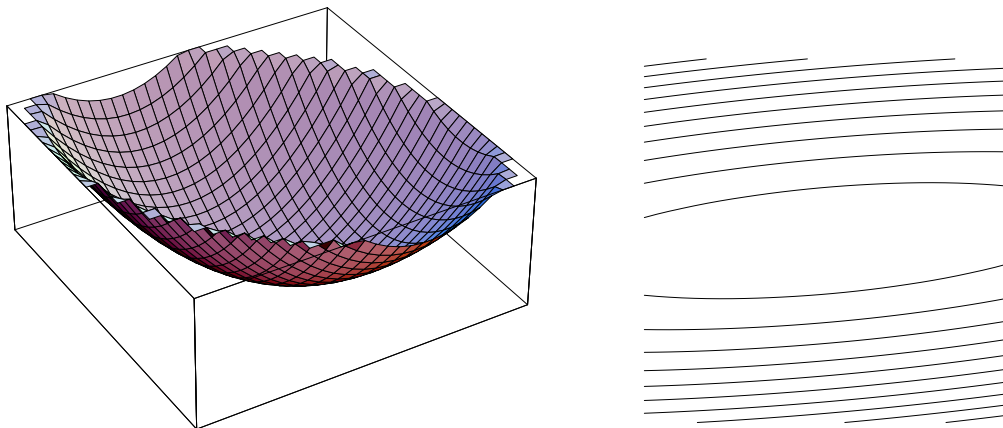


Figure 8: Graphe et courbe de niveau

Après une analyse de l'algorithme, on trouve les opérations arithmétiques suivantes pour chaque itération:

- une multiplication matrice*vecteur et deux produits scalaires;
- deux additions vecteur+nombre*vecteur;
- place en mémoire pour la matrice \mathbf{A} et 3 vecteurs.

Si la matrice \mathbf{A} (dimension $N \times N$) contient en moyenne nz nombres différents de zéro par ligne, il faut environ $(4 + nz) N$ opérations arithmétiques (addition/multiplication) pour effectuer un pas complet de l'itération.

```

choose initial point  $\vec{x}_0$ 
 $k = 0$ 
while  $\|\vec{r}_k\| = \|\mathbf{A}\vec{x}_k + \vec{b}\|$  too large
     $\vec{d}_k = -\vec{r}_k$ 
     $\alpha = -\frac{\langle \vec{r}_k, \vec{d}_k \rangle}{\langle \mathbf{A}\vec{d}_k, \vec{d}_k \rangle}$ 
     $\vec{x}_{k+1} = \vec{x}_k + \alpha \vec{d}_k$ 
     $k = k + 1$ 
endwhile

```

Tableau 3: Méthode du gradient pour résoudre $\mathbf{A}\vec{x} + \vec{b} = \vec{0}$

Au cours d'un pas de l'itération l'erreur est (typiquement) multipliée par le facteur $\left(1 - \frac{2}{\kappa}\right)$. On peut montrer (voir [VarFEM]) que pour augmenter la précision de D chiffres il faut k itérations, avec

$$k \geq \frac{D \ln 10}{2} \kappa \quad .$$

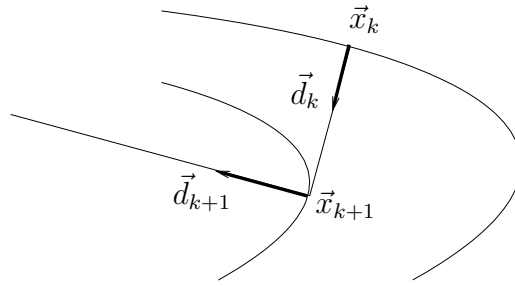


Figure 9: Un pas de la méthode de gradient

La matrice \mathbf{A} possède presque toujours un nombre de conditionnement κ très grand, et donc des courbes de niveau comme dans la figure 10. Puis la méthode va osciller dans le vallée au lieu de trouver le point minimal le plus rapidement possible. On a donc besoin d'un nombre énorme d'itérations pour trouver une bonne solution.

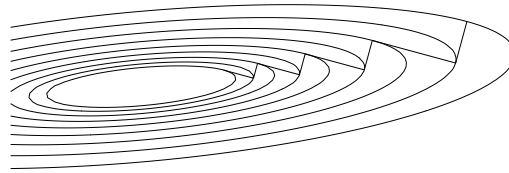


Figure 10: Comportement de la méthode du gradient dans une vallée étroite

3.6.3 Méthode du gradient conjugué

A l'aide d'une analyse exacte (voir [VarFEM]) on peut améliorer la méthode ci-dessus, de sorte que on arrive dans un seul pas au point minimal. La situation est esquissée dans la figure 11. L'algorithme est donnée en tableau 4.

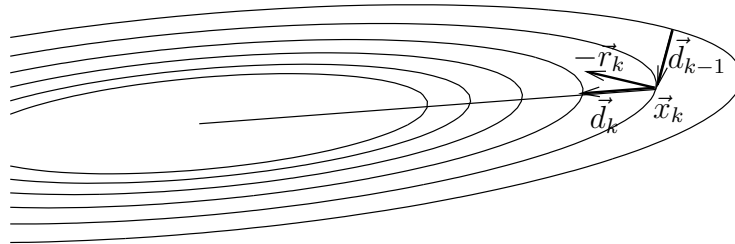


Figure 11: Un pas de la méthode du gradient conjugué

Après une analyse de l'algorithme, on trouve les opérations arithmétiques suivantes pour chaque itération:

- une multiplication matrice*vecteur et deux produits scalaires;
- trois additions vecteur+nombre*vecteur;
- place en mémoire pour la matrice \mathbf{A} et 4 vecteurs.

Si la matrice \mathbf{A} (dimension $N \times N$) contient en moyenne nz nombres différents de zéro par ligne, il faut environ $(5 + nz)N$ opérations arithmétiques (addition/multiplication) pour un pas

```

choose initial point  $\vec{x}_0$ 
 $\vec{r}_0 = \mathbf{A} \vec{x}_0 + \vec{b}$ 
 $\vec{d}_0 = -\vec{r}_0$ 
 $\alpha_0 = -\frac{\langle \vec{r}_0, \vec{d}_0 \rangle}{\langle \mathbf{A} \vec{d}_0, \vec{d}_0 \rangle}$ 
 $\vec{x}_1 = \vec{x}_0 + \alpha_0 \vec{d}_0$ 
 $k = 1$ 
while  $\|\vec{r}_k\| = \|\mathbf{A} \vec{x}_k + \vec{b}\|$  too large
     $\beta_k = \frac{\langle \vec{r}_k, \mathbf{A} \vec{d}_{k-1} \rangle}{\langle \vec{d}_{k-1}, \mathbf{A} \vec{d}_{k-1} \rangle}$ 
     $\vec{d}_k = -\vec{r}_k + \beta_k \vec{d}_{k-1}$ 
     $\alpha_k = -\frac{\langle \vec{r}_k, \vec{d}_k \rangle}{\langle \mathbf{A} \vec{d}_k, \vec{d}_k \rangle}$ 
     $\vec{x}_{k+1} = \vec{x}_k + \alpha_k \vec{d}_k$ 
     $k = k + 1$ 
endwhile

```

Tableau 4: La méthode du gradient conjugué pour résoudre $\mathbf{A} \vec{x} + \vec{b} = \vec{0}$

complet de l'itération. L'effort est donc comparable à la méthode du gradient. Au cours d'un pas de l'itération, l'erreur est (typiquement) multipliée par le facteur $(1 - \frac{2}{\kappa})$. A cause de $\sqrt{\kappa} \ll \kappa$ ce facteur diffère nettement plus de 1 que le facteur de la méthode du gradient. On peut montrer (voir [VarFEM]) que pour augmenter la précision de D chiffres il faut k itérations, avec

$$k \geq \frac{D \ln 10}{2} \sqrt{\kappa}$$

3.6.4 Un problème modèle

Comme problème simple examinons la distribution de la température dans une carrée avec un maillage uniforme avec $n \times n$ nœuds intérieurs.

$$\begin{aligned} u_{xx} + u_{yy} &= f(x, y) && \text{dans le domaine} \\ u(x, y) &= 0 && \text{au bord} \end{aligned}$$

La matrice \mathbf{A} de ce problème consiste en n^2 lignes et de n^2 colonnes avec une largeur de demie bande approximative $n + 1 \approx n$. Dans chaque ligne seulement 5 nombres sont différents de zéro. Le nombre de conditionnement est donné par $\kappa \approx \frac{4}{\pi^2} n^2$. Dans le tableau 5 on peut voir la quantité des places en mémoire est le nombre d'opérations arithmétiques nécessaires pour résoudre le système $\mathbf{A} \vec{x} = \vec{b}$ à l'aide de trois algorithmes différents. Dans le tableau 5 et les figures 12 on peut lire que:

- la méthode du gradient (steepest descent) est comparable à la méthode de Cholesky.
- pour un petit problème 2D Cholesky est plus efficace que la méthode du gradient conjugué.
- pour un grand problème 2D Cholesky est moins efficace que la méthode du gradient conjugué.

- pour tout problème 3D la méthode du gradient conjugué est plus efficace que Cholesky.

Les calculations sont faites avec $D = 6$, ce qui veut dire qu'on accepte un résultat 6 chiffres près. Pour estimer le temps de calcul dans la figure 12 il faut tenir compte des approximations $1h = 3600 \text{ sec}$, $1 \text{ jour} \approx 10^4 \text{ sec}$, $1 \text{ mois} \approx 2.5 \cdot 10^5 \text{ sec}$ et $1 \text{ année} \approx 3 \cdot 10^6 \text{ sec}$.

	2-D		3-D	
	storage	flops	storage	flops
Cholesky, banded	$\frac{1}{2} n^3$	$\frac{1}{2} n^4$	$\frac{1}{2} n^5$	$\frac{1}{2} n^7$
Gradient method	$8 n^2$	$9 \frac{2D \ln 10}{\pi^2} n^4$	$10 n^3$	$11 \frac{2D \ln 10}{\pi^2} n^5$
Conjugate Gradient	$9 n^2$	$10 \frac{D \ln 10}{\pi} n^3$	$11 n^3$	$12 \frac{D \ln 10}{\pi} n^4$

Tableau 5: Comparaison des méthodes pour le problème modèle

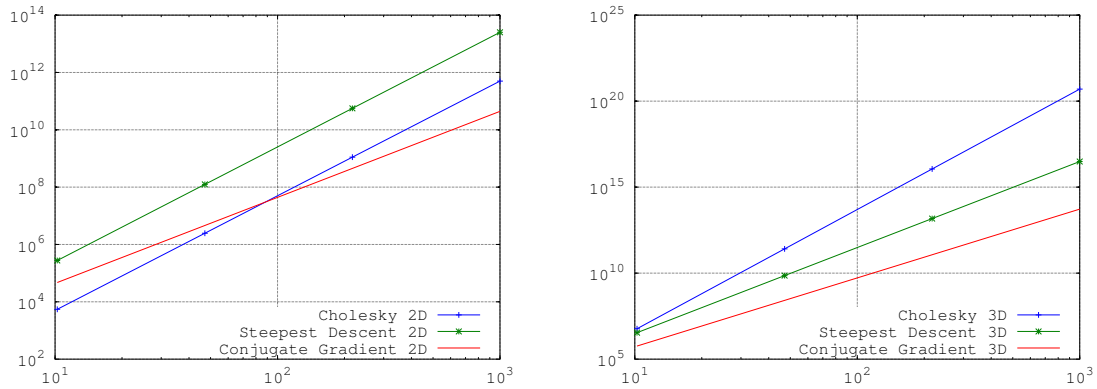


Figure 12: Effort de calcul pour bande Cholesky, gradient et gradient conjugué