

Einführung in die finite Elemente Methode

Andreas Stahel

September 2019

Zusammenfassung

In einem ersten Abschnitt werden die Grundideen der Methode der Finiten Elemente an einem einfachen System von Stäben illustriert. Die Präsentation ist teilweise [Zien84, p. 26] entnommen. Die Begriffe Elementsteifigkeitsmatrix, Gesamtsteifigkeitsmatrix und Zwangsbedingungen werden am konkreten Beispiel untersucht. Im zweiten Abschnitt wird die Steifigkeitsmatrix eines Biegestabes hergeleitet. In einem dritten Abschnitt gehen wir etwas genauer auf die spezielle Struktur der linearen Gleichungssysteme ein. Die entsprechenden Matrixgleichungen können mit verschiedenen Verfahren gelöst werden und wir stellen einige Vergleiche an.

Inhaltsverzeichnis

1	Struktur aus Zug- und Druckstäben	3
1.1	Beschreibung der Situation	3
1.2	Elementsteifigkeitsmatrix	3
1.3	Herleitung der Elementsteifigkeitsmatrix	4
1.4	Berechnen der einzelnen Elementsteifigkeitsmatrizen	6
1.4.1	Element a von Punkt 1 zu Punkt 3	6
1.4.2	Element b von Punkt 1 zu Punkt 2	7
1.4.3	Element c von Punkt 2 zu Punkt 3	7
1.4.4	Element d von Punkt 3 zu Punkt 4	7
1.4.5	Element e von Punkt 2 zu Punkt 4	8
1.5	Gesamtsteifigkeitsmatrix	8
1.6	Einsetzen der Zwangsbedingungen und Lösen der Gleichungen	9
1.7	Interpretation der Resultate	10
1.8	Octave-Code	11
1.8.1	Aufstellen einer Elementsteifigkeitsmatrix	11
1.8.2	Einfügen einer Elementsteifigkeitsmatrix in die Gesamtsteifigkeitsmatrix	11
1.8.3	Aufstellen und Lösen des Systems	12
1.8.4	Interpretation der Resultate	13
2	Elementsteifigkeitsmatrix-Matrix für einen Biegestab	14
2.1	Erzeugen der Elementsteifigkeitsmatrix in horizontaler Lage	14
2.2	Mit Zug-Druck-Belastung	16
2.3	Vollständige Steifigkeits-Matrix eines horizontalen Stabes	16
2.4	Drehen eines Stabes in horizontale Lage	16

3	Bandstruktur der Matrizen und Rechenaufwand, iterative Verfahren	19
3.1	Bandstruktur und numerieren der Knoten	19
3.1.1	Erster Versuch	19
3.1.2	Zweiter Versuch	20
3.2	Rechenaufwand, Verfahren von Cholesky	21
3.2.1	Verfahren von Gauss	21
3.2.2	Verfahren von Cholesky	23
3.2.3	Verfahren von Cholesky für Bandmatrizen	24
3.3	Eine leicht realistischere Struktur	25
3.4	Frontlösmethode	26
3.5	Vergleich von verschiedenen Methoden	27
3.6	Iterative Verfahren	28
3.6.1	Die Grundidee	28
3.6.2	Das Gradientenverfahren	28
3.6.3	Die Methode der konjugierten Gradienten	30
3.6.4	Ein Modelproblem	31

Literatur

- [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 Struktur aus Zug- und Druckstäben

1.1 Beschreibung der Situation

In Abbildung 1 ist eine einfache Struktur mit 5 Stäben (a, b, c, d, e) gegeben. Die Stäbe sind in den vier Verbindungspunkten gelenkig miteinander verbunden. Alle Stäbe haben dieselbe Länge L , Querschnittfläche A und sind aus einem Material mit Elastizitätsmodul E . Der Verbindungspunkt 2 ist fixiert und Punkt 1 kann nur in x -Richtung bewegt werden. Eine externe Last von 100 N wirkt in vertikaler Richtung auf Punkt 4.

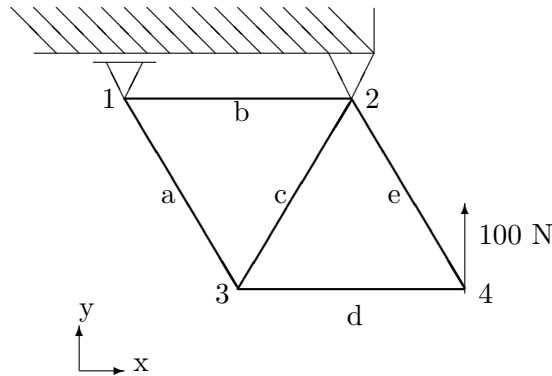


Abbildung 1: Struktur aus Zug- und Druckstäben

1.2 Elementsteifigkeitsmatrix

Wir betrachten das Element e, welches vom Punkt i zum Punkt n geht und einen Winkel α mit der x -Achse bildet. Abbildung 2 zeigt die typische Situation.

Die Verschiebungen der Punkte i und n aus der ursprünglichen Lage sind gegeben durch die Vektoren

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

Nun müssen die an den Stabenden angreifenden Kräfte $\vec{F}_{e,i}$ und $\vec{F}_{e,n}$ durch diese Verschiebungsvektoren ausgedrückt werden. **Wir betrachten hier die auf den Stab wirkenden Kräfte** und nicht die vom Stab ausgeübten Kräfte (Vorzeichen). In der Gesamtstruktur werden diese Kräfte durch die anderen Stäbe oder Lager erzeugt. Die Überlegungen in Abschnitt 1.3 zeigen, dass sich diese Kräfte aus den Verschiebungen durch Multiplikation mit der **Elementsteifigkeitsmatrix** \mathbf{K}_e berechnen lassen.

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

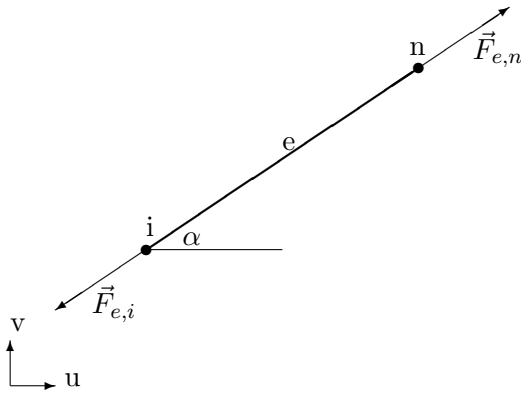


Abbildung 2: Ein einzelnes Element der Struktur

Dasselbe kann kurz auch als

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

geschrieben werden. Sind die Verschiebungen \vec{a}_e durch die auftretenden Kräfte „klein“, so wird sich der Winkel α des Elementes nicht wesentlich ändern. Bei grossen Verschiebungen sind aber die Änderungen der Winkel unbedingt zu berücksichtigen, indem die Elementsteifigkeitsmatrizen neu bestimmt werden. Wir wollen hier nicht auf diese Probleme eingehen und arbeiten deshalb unter der Voraussetzung

Die Verschiebungsvektoren sind klein im Vergleich zu den Abmessungen der Elemente. Somit können die Neigungswinkel der Elemente als konstant angesehen werden.

1.3 Herleitung der Elementsteifigkeitsmatrix

Hier ist eine mögliche Herleitung dieser Matrix. Sie beruht auf Matrizenoperationen (Drehungen in der Ebene) und dem Hookeschen Gesetz. Dieser Abschnitt kann übersprungen werden, falls man die Gleichung (1) ohne Verifikation weiterverwendet.

Wir betrachten einen Stab der Länge L mit Querschnittsfläche A aus einem Material mit Elastizitätsmodul E . Dehnt man diesen Stab um die Länge ΔL , so ist die Reaktionskraft gegeben durch

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

Für einen horizontalen Stab gilt $|\Delta L| = |u_i - u_n|$ und somit

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

Dieselben Beziehungen können auch mittels einer Matrix dargestellt werden durch

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

Dies entspricht der Elementsteifigkeitsmatrix in Gleichung (1) für den Spezialfall $\alpha = 0$. Hieraus leiten wir nun durch Drehungen die allgemeine Formel ab.

Drehen wir einen Vektor $\begin{pmatrix} u \\ v \end{pmatrix}$ um den Winkel α im Gegenuhrzeigersinn, so wird daraus der neue Vektor

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

Eine Drehung um den Winkel $-\alpha$ führt zur inversen Matrix

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

Wenn wir die Verschiebungsvektoren $\begin{pmatrix} u_i \\ v_i \end{pmatrix}$ und $\begin{pmatrix} u_n \\ v_n \end{pmatrix}$ eines Stabes mit Winkel α um $-\alpha$ drehen und auf einen horizontalen Stab anwenden, so müssen die Kraftvektoren des horizontalen Stabes entstehen durch Drehung der Vektoren $\vec{F}_{e,i}$ und $\vec{F}_{e,n}$ um denselben Winkel $-\alpha$. Das führt auf die Bedingung

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

Somit ist die Steifigkeitsmatrix gegeben durch

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

Durch Ausmultiplizieren der Matrizen erhält man

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

und somit

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

Die Matrix

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

heißt auch **Transformationsmatrix** und wir können auch die Notation

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

verwenden.

1.4 Berechnen der einzelnen Elementsteifigkeitsmatrizen

Nun sind die Resultate der beiden vorangehenden Abschnitte auf die 5 Elemente der Struktur in Abbildung 1 anzuwenden.

1.4.1 Element a von Punkt 1 zu Punkt 3

Bei diesem Element ist der Winkel $\alpha = -60^\circ = -\pi/3$ und somit

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

Die Transformationsmatrix ist

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

und somit

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

1.4.2 Element b von Punkt 1 zu Punkt 2

Bei diesem Element ist der Winkel $\alpha = 0$ und somit

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

Die Transformationsmatrix ist

$$\mathbf{T}(0) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

und somit

$$\mathbf{K}_b = \frac{E A}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

1.4.3 Element c von Punkt 2 zu Punkt 3

Bei diesem Element ist der Winkel $\alpha = -120^\circ = -2\pi/3$ und somit

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

Die Transformationsmatrix ist

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

und somit

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

1.4.4 Element d von Punkt 3 zu Punkt 4

Bei diesem Element ist der Winkel $\alpha = 0$ und somit

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

Die Transformationsmatrix ist

$$\mathbf{T}(0) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

und somit

$$\mathbf{K}_d = \frac{E A}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

1.4.5 Element e von Punkt 2 zu Punkt 4

Bei diesem Element ist der Winkel $\alpha = -60^\circ = -\pi/3$ und somit

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

Die Transformationsmatrix ist

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

und somit

$$\mathbf{K}_e = \frac{EA}{L} \begin{bmatrix} 0.25 & -0.433 & -0.25 & 0.433 \\ -0.433 & 0.75 & 0.433 & -0.75 \\ -0.25 & 0.433 & 0.25 & -0.433 \\ 0.433 & -0.75 & -0.433 & 0.75 \end{bmatrix}$$

1.5 Gesamtsteifigkeitsmatrix

Um etwas einfachere Zahlen zu erhalten rechnen wir in den nächsten beiden Abschnitten mit

$$\frac{EA}{L} = 1$$

Nun müssen wir diese fünf Elementsteifigkeitsmatrizen zur einer Steifigkeitsmatrix für das Gesamtsystem zusammensetzen. Da jeder der 4 Punkte zwei Freiheitsgrade hat, ergeben sich vorerst 8 Unbekannte, da die Zwangsbedingungen noch nicht berücksichtigt werden.

Hierzu beginnen wir mit einer mit 0 gefüllten Matrix

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

Die erste Zeile dieses Systems von linearen Gleichungen soll der x -Komponente der Kräftegleichgewichtsbedingung im ersten Punkt entsprechen, die zweite Zeile der y -Komponente. Die dritte und vierte Zeile beinhalten die Gleichgewichtsbedingung im zweiten Punkt, usw. Der Kraftvektor auf der rechten Seite enthält die äusseren Kräfte. Die Lagerkräfte in den Punkten 1 und 2 sind noch unbekannt.

Als erstes fügen wir die Terme ein, die durch die Elemente a und d erzeugt werden. Das führt

auf das folgende Zwischenresultat, hierbei wurden Einträge 0 ersetzt durch einen Punkt.

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

Durch einsetzen der Elemente b , c und e erhält man schliesslich

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

Mit den passenden Abkürzungen kann das auch geschrieben werden als

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

Somit haben wir 8 lineare Gleichungen, aber auf den ersten Blick 11 Unbekannte: die 8 Verschiebungen und die 3 Lagerkräfte. Wir werden erst später verwenden, dass $v_1 = u_2 = v_2 = 0$. Die **Gesamtsteifigkeitsmatrix** \mathbf{K} hat einige spezielle Eigenschaften:

- Sie ist symmetrisch.
- Einige der Einträge sind 0. Da der Knoten 1 keine direkte Verbindung zum Knoten 4 hat müssen die 2×2 Blöcke in der oberen rechten und unteren linken Ecke der Matrix 0 sein. Bei grösseren Strukturen (mehr Stäbe) wird dies noch viel ausgeprägter der Fall sein. Dort erhalten wir typischerweise **schwach besetzte Matrizen**, d.h. fast alle Einträge sind 0.
- Die Matrix ist **positiv definit**, d.h. all Ihre Eigenwerte sind grösser oder gleich 0.

1.6 Einsetzen der Zwangsbedingungen und Lösen der Gleichungen

Aus Abbildung 1 ist ersichtlich, dass $v_1 = u_2 = v_2 = 0$. Somit sind die Beiträge aus der 2., 3. und 4. Spalte der obigen Matrix immer 0. Wir entfernen diese Spalten. Um die Lagerkräfte (noch) nicht berücksichtigen zu müssen, können wir im obigen System von Gleichungen die 2., 3. und 4. Zeile streichen. Das führt auf 5 Gleichungen für 5 Unbekannte.

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

Diese neue Matrix erbt von \mathbf{K} die Eigenschaften:

- Sie ist symmetrisch.
- Einige der Einträge sind 0.
- Die Matrix ist **strikt positiv definit**, d.h. all Ihre Eigenwerte sind grösser als 0.

Diese System kann nun numerisch gelöst werden mit dem Resultat

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

und somit haben wir den Verschiebungsvektor \vec{a} des Systems bestimmt als

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

Berücksichtigt man $\frac{EA}{L} \neq 1$, so ist für den „richtigen“ Verschiebungsvektor \vec{a} noch mit $\frac{L}{EA}$ zu multiplizieren. Da bei den hier verwendeten Steifigkeitsmatrizen ein Faktor $\frac{EA}{L}$ „fehlt“ erhalten wir aber für Multiplikationen von Steifigkeitsmatrizen mit Verschiebungsvektoren (d.h. für Kräfte) trotzdem die richtigen Zahlen.

1.7 Interpretation der Resultate

Das obige Resultat \vec{a} für die Verschiebungen kann nun in $\mathbf{K}\vec{a} = \vec{F}_{ext}$ eingesetzt werden und wir erhalten

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

Hiermit können wir gleichzeitig die numerische Lösung des Gleichungssystems überprüfen und die Lagerkräfte bestimmen. So übt das Lager 1 auf die Struktur eine Kraft von 50 N nach oben aus.

Mit Hilfe der Elementsteifigkeitsmatrizen kann man nun auch die Belastung der einzelnen Stäbe bestimmen. Wir betrachten als Beispiel das Element c . Es gilt

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

und wir erhalten

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

Somit wirkt auf den Stab c eine Druckbelastung der Stärke $\sqrt{28.87^2 + 50^2} \text{ N} = 57.74 \text{ N}$.

1.8 Octave-Code

1.8.1 Aufstellen einer Elementsteifigkeitsmatrix

Das File `BarElementMatrix.m` enthält eine Funktion, um die Elementsteifigkeitsmatrix eines Stabes zu bestimmen.

```

----- BarElementMatrix.m -----
function elmat = BarElementMatrix(Coeff, Angle)
% BarElementMatrix(Coeff, Angle)
% generates the 4x4 Element stiffnes 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 Einfügen einer Elementsteifigkeitsmatrix in die Gesamtsteifigkeitsmatrix

Das File `assemble.m` enthält eine Funktion, um die eine Elementsteifigkeitsmatrix in die Gesamtsteifigkeitsmatrix einzufügen.

```

----- 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 Aufstellen und Lösen des Systems

Mit den beiden obigen Funktionen können wir das System nun leicht aufstellen und lösen. Der Code ist im File `BarSystem.m` gespeichert.

```

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

```

Dieses Programm erzeugt die Resultate

```

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 Interpretation der Resultate

Mit den beiden Zeilen

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

kann man die internen Kräfte auf Element c bestimmen mit dem Resultat

```
Octave  
ans =  
-28.87  
-50.00  
28.87  
50.00
```

2 Elementsteifigkeitsmatrix–Matrix für einen Biegestab

Im vorangehenden Abschnitt waren nur reine Zug- und Druck–Belastungen des Stabes zugelassen. Nun wollen wir auch Biegung von Stäben untersuchen. Die Herleitung der Elementsteifigkeits–Matrix wurde teilweise dem Buch [Loga92, p. 156] entnommen.

2.1 Erzeugen der Elementsteifigkeitsmatrix in horizontaler Lage

Wir untersuchen einen Stab der Länge L in horizontaler Lage. Damit kann die vertikale Auslenkung beschrieben werden durch eine Funktion $y(x)$ wobei $0 \leq x \leq L$. Vorerst werden **keine horizontalen Kräfte und Verschiebungen** berücksichtigt.

Wir schreiben die vertikalen Kräfte f_y und Momente m als Funktion der Auslenkungen y und der Auslenkungswinkel y' mit Hilfe einer Matrizenmultiplikation

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

Für kleine Auslenkungen und Kräfte ist die Beziehung zwischen Kraft/Moment und Auslenkungen sicher linear, deshalb ist der obige Ansatz gerechtfertigt. Die Matrix \mathbf{M} wird zur **Elementsteifigkeitsmatrix**.

Nun sind die 16 Koeffizienten in der 4×4 –Matrix \mathbf{M} zu bestimmen. Dazu untersuchen wir 4 spezielle Situationen. Daraus erhalten wir genügend Gleichungen um die Koeffizienten zu bestimmen. Die Überlegungen verwenden die physikalische Grundformel

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

1. Zuerst verschieben wir den ganzen Stab um eine Einheit nach oben. Für die einfache Auslenkungsfunktion $y(x) = 1$ sind alle resultierenden Momente und Kräfte 0, also gilt

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

2. Auch für die Auslenkungsfunktion $y(x) = x$ sind alle resultierenden Momente und Kräfte 0, also gilt

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

3. Nun betrachten wir einen links eingespannten Stab, auf den bei $x = L$ eine Kraft 1 nach oben wirkt, d.h.

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

Somit ist das Moment gegeben durch

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

und die folgende Gleichung zu lösen.

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

Die Lösung beruht auf elementaren Integrationen und der Einspannbedingung bei $x = 0$. Das führt auf die Resultate

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

Da die Summe aller wirkenden Kräfte Null sein muss ist $f_y(0) = -1$.

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

4. Nun betrachten wir einen rechts eingespannten Stab, auf den bei $x = 0$ eine Kraft 1 nach oben wirkt, d.h.

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

Symmetrieüberlegungen führen auf $f_y(L) = -1$ und

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

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

Kombinieren der obigen vier Resultate ergibt die Matrixgleichung

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

oder auch

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

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

Somit haben wir die **Elementsteifigkeitsmatrix** eines Biegestabes in horizontaler Lage gefunden. Allerdings wurden horizontale Kräfte und Verschiebungen noch nicht berücksichtigt.

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

2.2 Mit Zug–Druck–Belastung

Belastet man den selben Stab mit Kräften f_x in horizontaler Richtung, so ergeben sich horizontale Verschiebungen $x(0)$ und $x(L)$ der Endpunkte. Gemäss dem Gesetz von Hook gilt

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

2.3 Vollständige Steifigkeits–Matrix eines horizontalen Stabes

Als Freiheitsgrade des Elementes verwenden wir nun die Auslenkungen x_1, x_2, y_1 und y_2 an den beiden Endpunkten und die Winkelabweichungen θ_1, θ_2 . Für kleine Winkelabweichungen gilt $\theta_1 \approx y'(0)$ und $\theta_2 \approx y'(L)$. Somit haben wir

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

2.4 Drehen eines Stabes in horizontale Lage

Dreht man einen Vektor $(u, v)^T$ um einen Winkel α im Gegenuhrzeigersinn, so entsteht der neue Vektor

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

Da sich die Winkelabweichung durch eine Drehung nicht ändert können wir das Drehen des Stabes durch eine Matrizenmultiplikation darstellen.

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

Ebenso kann durch

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

der Vektor der Kräfte und Momente gedreht werden.

Nun untersucht man einen Stab, der mit der x -Achse einen Winkel α einschliesst. Dreht man Verschiebungen und Kräfte um den Winkel $-\alpha$ so entsteht ein horizontaler Stab und man kann die Steifigkeitsmatrix des vorangehenden Abschnitts verwenden. Es gilt

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

oder auch

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

Hierbei haben wir verwendet, dass

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

Somit kann die Elementsteifigkeitsmatrix M eines Biegestabes mit Winkel α bestimmt werden

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

Das Ausmultiplizieren ersparen wir uns hier, um den Ausdruck nicht noch komplizierter erscheinen zu lassen. *Mathematica* oder *Octave* kann diese Rechnung für Sie ausführen. Um \mathbf{M} zu berechnen müssen nur die Daten E, I, A, L und α gegeben sein.¹ Die Elementsteifigkeits-Matrix des reinen Biegestabes finden Sie in [Loga92, p. 197]. Wir haben die Steifigkeitsmatrix eines gemischeten Zug/Druck und Biege-Stabes hergeleitet. Auch diese Elementmatrix ist symmetrisch und positiv definit. Nun können auch Stabstrukturen mit Biegestäben untersucht werden.

¹Setzt man $\cos \alpha = \text{Co}$ und $\sin \alpha = \text{Si}$ so ist die Matrix gegeben durch

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

3 Bandstruktur der Matrizen und Rechenaufwand, iterative Verfahren

In diesem Abschnitt sollen Sie mit einigen typischen Problemen des Lösens der linearen Gleichungssysteme bekannt gemacht werden. Ein sorgfältiges Studium dieser Probleme ist leider im gegebenen Zeitrahmen nicht möglich. Es kann nur darum gehen Ihnen aufzuzeigen, weshalb die Begriffe **Bandstruktur**, **Knotennumerierung** und **Frontallösungsmethode** einen entscheidenden Einfluss haben auf Ihre Arbeit mit Finiten Elementen: eine „falsche“ Wahl der Knotennumerierung kann der Unterschied sein zwischen einer Aufgabe die schnell gelöst werden kann und einem scheinbar unlösbaren Problem. Einige der technischen Aspekte in diesem Abschnitt sind bewusst vage formuliert. Die wesentlichen Effekte lassen sich durch diese formalen Argumente gut aufzeigen. Für sorgfältigere Untersuchungen sei auf weitergehende Literatur verwiesen (z. B. [GoerRoosTobi93], [GoluVanLoan89], [Schw88], [LascTheo86]).

Glücklicherweise haben viel in FEM auftretenden linearen Gleichungssysteme die selben speziellen Eigenschaften. Für die Matrix \mathbf{A} im System

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

gilt

1. sie ist symmetrisch, d.h. $a_{i,j} = a_{j,i}$.
2. sie hat eine Bandstruktur, d.h. die von 0 verschiedenen Zahlen sind in einem Band der Breite $2b$ (Bandbreite) um die Diagonale konzentriert.
3. sie ist strikt positiv definit, d.h.

$$\vec{x}^T \cdot \mathbf{A} \vec{x} > 0 \quad \text{für alle Vektoren } \vec{x} \neq \vec{0}$$

Diese Eigenschaft beruht bei den meisten Anwendungen, auf der Tatsache, dass man versucht die potentielle Energie des Systems zu minimieren. Darauf beruht auch der sehr enge Zusammenhang der Methode der finiten Elemente und der **Variationsrechnung**.

Nun geht es darum diese Eigenschaften geschickt auszunutzen. Da die Matrix \mathbf{A} positiv definit ist, ergeben sich beim Lösen des Gleichungssystems viel weniger numerische Probleme:

- beim Gauss-Algorithmus müssen keine Zeilenvertauschungen vorgenommen werden (keine spezielle Pivotwahl ist nötig). Das Verfahren von Cholesky kann in dieser Situation angewandt werden, es muss nie die Wurzel einer negativen Zahl berechnet werden.
- die Verfahren von Gauss und Cholesky sind **stabil**, d.h. ein kleiner Rundungsfehler in der Rechnung wird nicht zu einem grossen Fehler in der Lösung führen.

Vor allem die zweite Aussage ist für Anwendungen sehr nützlich, da wir grosse Fehler aufgrund des Lösungsverfahrens für die linearen Gleichungen ausschliessen können.

3.1 Bandstruktur und numerieren der Knoten

3.1.1 Erster Versuch

Wir betrachten eine etwas kompliziertere Struktur mit 15 Verbindungspunkten (Knoten) und 27 Stäben (Elementen). Da jedem Knoten zwei Freiheitsgrade entsprechen wird die Gesamtsteifigkeitsmatrix \mathbf{K} hier eine symmetrische 30×30 Matrix sein. Nun versuchen wir herauszufinden, welche der 900 möglichen Einträge nicht Null sind. Dazu teilen wir die Matrix in 2×2 Blöcke auf und erhalten eine 15×15 Struktur von Einträgen (je eine 2×2 Matrix). Ein Eintrag in dieser Struktur ist von Null verschieden, falls ein Element (Stab) die Knoten mit den entsprechenden Nummern verbindet. Hierzu müssen wir natürlich zuerst die Knoten der Struktur numerieren und eine mögliche Lösung finden Sie in Abbildung 3.

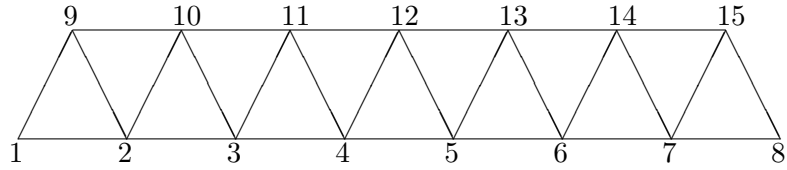


Abbildung 3: Grosse Struktur aus Zug- und Druckstäben, erste Numerierung

	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

Die Gesamtsteifigkeitsmatrix der Numerierung in Abbildung 3 ist also eine symmetrische Matrix mit halber Bandbreite $b = 18$. Will man das entsprechende System von Gleichungen mit der Methode von Cholesky lösen, so müssen die „leeren“ Bänder zwischen den Zahlen durch 0 aufgefüllt werden, da sie im Verlaufe der Rechnung durch von 0 verschiedene Zahlen ersetzt werden. Die beiden Randbänder werden durch der Lösungsvorgang nicht touchiert und brauchen deshalb auch nicht gespeichert zu werden. Man kann leicht abschätzen, dass ca. $2 \cdot 18 \cdot 30 = 1080$ Speicherplätze benötigt werden. Da die Matrix symmetrisch ist und diese Symmetrie durch das Verfahren von Cholesky erhalten bleibt kommt man mit ca. 540 Speicherplätzen aus.

3.1.2 Zweiter Versuch

Eine andere mögliche Numerierung derselben Struktur ist in Abbildung 4 gezeigt.

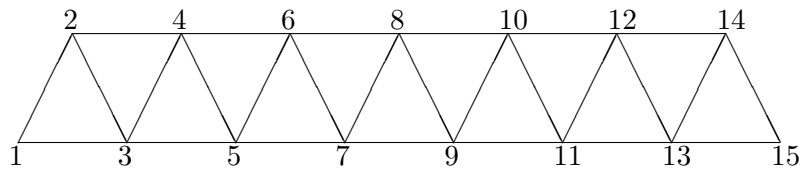


Abbildung 4: Grosse Struktur aus Zug- und Druckstäben, zweite Numerierung

	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

Die Gesamtsteifigkeitsmatrix der Numerierung in Abbildung 4 ist also eine symmetrische Matrix mit halber Bandbreite 6 oder voller Bandbreite 10. Um dieses Gleichungssystem mit dem Verfahren von Cholesky lösen zu können benötigt man ca. $10 \cdot 30/2 = 150$ Speicherplätze.

3.2 Rechenaufwand, Verfahren von Cholesky

Wir können hier nur eine kurze Zusammenfassung der wesentlichen Aspekte geben. Eine etwas ausführlichere Präsentation finden Sie in Kapitel 2 des Buches [GoerRoosTobi93].

Das Verfahren der Finiten Elemente führt sehr oft auf lineare Systeme von Gleichungen der Form

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

wobei \mathbf{A} eine **symmetrische, strikt positiv definite Matrix** der Grösse $n \times n$ mit **halber Bandbreite** b ist. Nun geht es darum diese speziellen Eigenschaften geschickt auszunutzen, um die Genauigkeit zu verbessern und Rechenzeit zu sparen. Hierzu muss ein geeigneter Algorithmus gefunden und eingesetzt werden.

3.2.1 Verfahren von Gauss

Wir können das System von Gleichungen

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

auch darstellen durch die erweiterte Matrix

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

Beim klassischen Verfahren von Gauss bestimmt man mittels der ersten Gleichung x_1 als Funktion von x_2, \dots, x_n und b_1 und setzt das Resultat in alle anderen Gleichungen ein, um x_1 zu eliminieren. Dies führt auf ein **neues**, äquivalentes System von Gleichungen der Form

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

Hierbei ist unbedingt zu beachten, dass **die Zahlen** ändern, das heisst der „neue“ Wert von $a_{2,2}$ ist nicht gleich dem „alten“ Wert von $a_{2,2}$. Beim nächsten Schritt eliminiert man mit Hilfe der neuen zweiten Gleichung die Unbekannte x_2 aus allen weiteren Gleichungen um zur folgenden Struktur zu gelangen.

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

Nach n Eliminationsschritten erhält man ein Gleichungssystem der Form

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

Sieht man diese Rechnungen etwas sorgfältiger an, so erkennt man, dass die Matrix **A** umgeschrieben wird als Produkt von zwei Matrizen

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

wobei

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

und

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

Statt des Systems

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

löst man

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

Hierbei sind wir davon ausgegangen, dass die Gleichungen eindeutig lösbar sind und im Verlaufe der Rechnungen nie durch 0 dividiert werden muss (Stichwort Pivotstrategie). Ist \mathbf{A} eine strikt positiv definite, symmetrische Matrix, so sind diese Bedingungen erfüllt.

Um den Rechenaufwand abzuschätzen fasst man eine Addition und Multiplikation als typische Operation zusammen und erhält für das klassische Verfahren von Gauss einen Rechenaufwand von ca.

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

wesentlichen Operationen um ein System von n Gleichungen zu lösen und man muss n^2 Zahlen speichern.

3.2.2 Verfahren von Cholesky

Diese Methode ist vergleichbar zum Verfahren von Gauss, verwendet aber die Symmetrie der Matrix. Der Algorithmus bestimmt aus der positiv definiten, symmetrischen Matrix \mathbf{A} eine obere (rechte) Dreiecksmatrix

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

mit

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

Das System von Gleichung kann nun gelöst werden durch

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

Aufgrund der Symmetrie halbiert sich der wesentliche Rechenaufwand auf

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

und man benötigt $\frac{1}{2} n^2$ Speicherplätze.

Hier ist ein mögliche Realisierung der Verfahrens von Cholesky (in *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 Verfahren von Cholesky für Bandmatrizen

Wir gehen davon aus, dass \mathbf{A} eine symmetrische, strikt positiv definite $n \times n$ Matrix mit halber Bandbreite b ist. Wir gehen davon aus, dass

$$1 \ll b \ll n$$

d.h. die Bandbreite ist viel kleiner als die Gesamtgröße der Matrix. Die Matrix \mathbf{R} der Cholesky-Zerlegung

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

wird erzeugt durch ein Rechenschema das vergleichbar ist zum Gauss-Algorithmus. Die Matrix \mathbf{R} erbt die Bandstruktur von \mathbf{A} und hat somit halbe Bandbreite b .

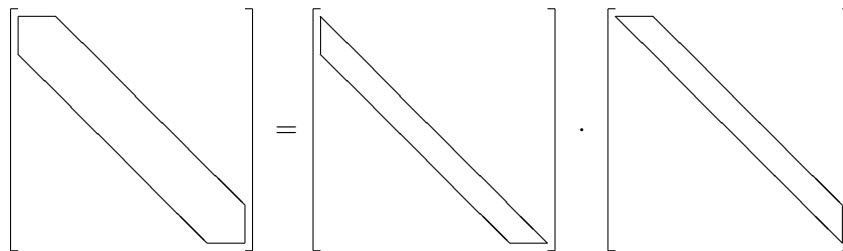


Abbildung 5: Verfahren von Cholesky für Bandmatrizen

Um eine Gleichung

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

zu lösen kann man die folgenden Schritte ausführen:

1. Bestimme die Zerlegung

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

2. Löse das Gleichungssystem

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

nach \vec{y} auf. Dieses System von linearen Gleichungen lässt sich elementar von „oben nach unten“ auflösen.

3. Löse das Gleichungssystem

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

nach \vec{y} auf. Dieses System von linearen Gleichungen lässt sich elementar von „unten nach oben“ auflösen.

Wegen

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

hat man die Lösung des ursprünglichen Problems gefunden.

Da \mathbf{A} symmetrisch ist müssen ca. $n \cdot b$ Zahlen abgespeichert werden und \mathbf{R} benötigt genau gleich viel Platz. Eine genau Analyse der Rechnungen zeigt, dass man sogar im Verlaufe der Rechnungen die Einträge für \mathbf{A} durch die Einträge für \mathbf{R} ersetzen kann. Man muss nicht beide Matrizen gleichzeitig zur Verfügung haben und es ergibt sich ein totaler **Speicherbedarf** von ca. $n \cdot b$ Zahlen. Nun schätzen wir noch die Anzahl der notwendigen Operationen für die drei Rechenschritte ab.

1. Um mit Hilfe der ersten Gleichung x_1 aus der zweiten bis b -ten Gleichung zu eliminieren brauchen wir ca. b^2 Operationen. Um mit Hilfe der neuen zweiten Gleichung x_2 aus der dritten bis $(b + 1)$ -ten Gleichung zu eliminieren brauchen wir auch ca. b^2 Operationen. Insgesamt müssen wir ca. n solche Eliminationsprozesse mit je b^2 Operationen ausführen. Das ergibt ein Total von $n \cdot b^2$ wesentlichen Operationen. Eine genauere Analyse zeigt, dass man aufgrund der Symmetrie nur $n \cdot b^2/2$ Operationen benötigt.
2. Bei jedem der n Schritte der Vorwärtseinsetzen sind weniger als b Operationen auszuführen, was einen Totalaufwand von $n \cdot b$ ergibt. Wegen $1 \ll b \ll n$ ist dieser Aufwand vernachlässigbar im Vergleich zu $n \cdot b^2/2$.
3. Auch der Rechenaufwand für das Rückwärtseinsetzen ist vernachlässigbar.

Somit erhalten wir einen **totalen Rechenaufwand** von $n \cdot b^2/2$ Operationen.

3.3 Eine leicht realistischere Struktur

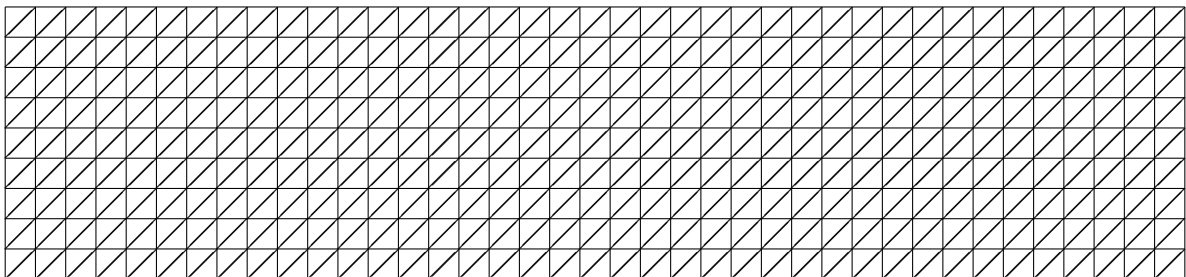


Abbildung 6: Eine „kleine“ Struktur aus Zug- und Druckstäben

Damit Sie nicht zählen müssen sei verraten, dass in jeder Zeile der Struktur 40 Knoten sind und insgesamt 10 Zeilen von Knoten. Somit hat das System 800 Freiheitsgrade und man müsste eine 800×800 Matrix speichern. Diese 640'000 Zahlen im Speicher eines Rechner zu behalten kann bereits Probleme verursachen. Man muss unbedingt die Bandstruktur und Symmetrie der Matrix ausnutzen. Nun geht es darum diese 400 Knoten „richtig“ zu numerieren. Dazu gibt es zwei einfache Varianten:

1. **Zeilenweise:** Unten links beginnend, erste Zeile, dann zweite Zeile, \dots , immer von links nach rechts.

2. **Spaltenweise:** Unten links beginnend, erste Spalte, dann zweite Spalte, \dots , immer von unten nach oben.

Die Struktur der entsprechenden Gesamtsteifigkeitsmatrix wird verschieden sein. Verifizieren Sie nun die Einträge in der folgenden Tabelle

Numerierung	Zeilenweise	Spaltenweise
Grösse	800×800	800×800
halbe Bandbreite	$2 \cdot 41 + 2 = 84$	$2 \cdot 11 + 2 = 24$
Speicherbedarf ca. (in Zahlen)	74'400	18'400
Rechenaufwand mit Gauss ($n^3/3$)	$170 \cdot 10^6$	$170 \cdot 10^6$
Rechenaufwand mit Cholesky ($n^3/6$)	$85 \cdot 10^6$	$85 \cdot 10^6$
Rechenaufwand mit Cholesky und Bandstruktur ($n \cdot b^2/2$)	$2.8 \cdot 10^6$	$0.21 \cdot 10^6$

Numeriert man diese Struktur spaltenweise, so wird die Rechenzeit also ca. 13 mal kleiner als bei zeilenweiser Numerierung. Bei grösseren Strukturen kann der Unterschied noch erheblich grösser sein.

3.4 Frontlösungsmethode

Eine einfache, dreidimensionale Struktur aus Stäben mit $20 \times 20 \times 10$ Verbindungspunkten hat 4'000 Knoten und somit $n = 12'000$ Freiheitsgrade. Die entsprechende Gesamtsteifigkeitsmatrix hat eine Grösse von 12000×12000 und eine halbe Bandbreite von $b = 600$. Rechnen wir für eine Zahl einen Speicherbedarf von 10 Byte, so würde die gesamte Matrix $10 \times 12000 \times 12000$ Byte oder 1.44 GB Speicherplatz belegen. Verwendet man Bandstruktur und Symmetrie, so reduziert sich der Speicherbedarf auf $10 \times 600 \times 12000$ Byte oder 72 MB. Es ist heute kein grösseres Problem diesen Platz auf einer Harddisk zu finden, aber nicht jeder Computer ist mit mehr als 72 MB RAM ausgerüstet. Ein Verfahren, das die Matrix nie vollständig im Speicher halten muss, ist deshalb sehr nützlich. Das Verfahren von Cholesky erlaubt es uns mit nur b^2 Zahlen im Speicher auszukommen. Mit Hilfe der Symmetrie kann das sogar auf $b^2/2$ Zahlen reduziert werden. Für unser Beispiel ergibt dies einen Speicherplatzbedarf von 1.8 MB. Selbst auf PC's sind heutzutage (1997) etliche MB RAM-Speicher die Regel.

Bei der Frontlösungsmethode wird die Steifigkeitsmatrix immer nur so weit aufgebaut, wie dies für die Elimination der obersten Zeile des aktiven Bereichs nötig ist. Dann wird die aktive Zone verschoben, die fehlenden Elemente der Steifigkeitsmatrix berechnet und die neue oberste Zeile bearbeitet.

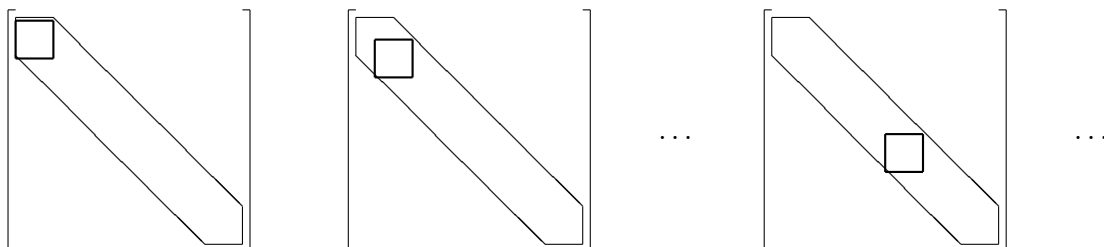


Abbildung 7: Frontlösungsmethode für Bandmatrizen

1. In einem ersten Schritt werden alle Matrixelemente des **aktiven Bereichs** oben links aufgebaut und anschliessend die zur ersten Zeile gehörenden Rechenschritte des Choleskyverfahrens ausgeführt.
2. Die erste Zeile der (teilweise) reduzierten Matrix kann nun auf die Harddisk geschrieben werden und die nächste Zeile der Matrix kann aufgebaut werden. Dann werden die der zweiten Zeile entsprechenden Schritte der Reduktion durchgeführt.
3. Bei jedem weiteren Schritt wird der aktive Bereich um eine Zeile nach unten rechts geschoben.
4. Nach n Schritten dieser Art befindet sich die Matrix \mathbf{R} der Zerlegung $\mathbf{A} = \mathbf{R}^T \cdot \mathbf{R}$ auf dem externen Speichermedium.
5. Der Speicherbedarf beim Lösen der Gleichungen

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

ist gering.

Beim Lösen der FEM-Probleme wechseln sich Aufbauschnitte der Koeffizientenmatrix und Eliminationsschritte ab, was zu einer etwas komplexeren Programmstruktur führt. Dieser Nachteil wird aber bei weitem aufgehoben durch den wesentlich kleineren Bedarf an RAM.

Computer	Anzahl FLOP pro Sekunde
NeXT (68040/25MHz)	1.0 M
HP 710/100	11.5 M
SUN Sparc ULTRA 10 (440MHz)	50.0 M
Pentium III 800 (zu wenig Cache)	50.0 M
Pentium III 800 (in Cache)	185 M
Pentium 4 2.6 GHz (zu wenig Cache)	370 M
Pentium 4 2.6 GHz (in Cache)	450 M
Intel I7-920, 2.6 GHz	700 M
Intel Haswell 5930 3.5 GHz	2'000 M

Tabelle 1: Rechenleistung einiger Computer

3.5 Vergleich von verschiedenen Methoden

Die Überlegungen des vorangehenden Abschnitt illustrieren wir mit einem konkreten Beispiel, für das wir Speicherbedarf und Rechenzeit abschätzen. Dazu setzen wir die Tabelle 1 von Rechenleistungen von an unserer Schule eingesetzten Computern ein. In einem kleinen Testprogramm wird eine symmetrische, positiv definite Matrix \mathbf{A} aufgebaut. Anschliessend wird die Zeit für das Berechnen der Cholesky-Zerlegung gemessen. Die Resultate sind in Tabelle 1 aufgeführt.

Die dreidimensionale Struktur in Abschnitt 3.4 ergibt eine Matrix der Grösse 12000×12000 mit halber Bandbreite 600. Pro Zahl wird ein Speicherbedarf von 10 Byte verwendet und die Schätzung der Rechenzeit bezieht sich nur auf die Cholesky-Zerlegung. Die Resultate sind in Tabelle 2 aufgeführt. Die selben qualitativen Effekte treten bei Berücksichtigung aller Aspekte einer FEM Lösung ebenfalls auf. Es ist dem Leser überlassen die richtigen Schlussfolgerungen zu ziehen.

Verfahren	Gauss volle Matrix	Cholesky mit Bandstruktur	Frontallösungsmethode
Grösse der Matrix	12000 × 12000	12000 × 600	12000 × 600
Speicherbedarf Cache/RAM	1440 MB	72 MB	1.8 MB
Speicherbedarf RAMD/extern	1440 MB	72 MB	72 MB
Anzahl Operationen	$n^3/3$ $576 \cdot 10^9$	$n b^2/2$ $2.2 \cdot 10^9$	$n b^2/2$ $2.2 \cdot 10^9$
Rechenzeit	$5.76 \cdot 10^5$ sec	$2.2 \cdot 10^3$ sec	$2.2 \cdot 10^3$ sec
NeXT	7 Tage	36 min	36 min
HP710	14 Std	3.2 min	3.2 min
Pentium III 800	3.2 Std	45 sec	45 sec
Pentium 4 2.6 GH	0.5 Std	6 sec	6 sec

Tabelle 2: Rechenaufwand für Gauss, Cholesky und Frontallösung

3.6 Iterative Verfahren

Die Steifigkeitsmatrix der Struktur in Abschnitt 3.4 hat tatsächlich weniger als 30 von Null verschiedene Einträge pro Zeile. Die Bandstruktur „verwendet“ 600 Einträge. Um diese Sparpotential auszunutzen kann man **iterative Methoden** zum Lösen der Gleichungssysteme einsetzen. Deren Studium überschreitet den hier gegebenen Rahmen und es seien nur einige Stichworte angegeben: Verfahren von Jacobi, Gauss–Seidel, Verfahren der konjugierten Gradienten, Vorkonditionierung. Detailliertere Beschreibungen finden Sie in [VarFEM] oder vielen guten Büchern über Numerik.

3.6.1 Die Grundidee

Für FEM Probleme sind Systeme linearer Gleichungen $\mathbf{A}\vec{x} + \vec{b} = \vec{0}$ zu lösen, wobei die Matrix \mathbf{A} oft symmetrisch und positiv definit ist. Die Matrix \mathbf{A} ist immer **dünn besetzt**, d.h. nur sehr wenige Einträge sind verschieden von Null. Somit ist der Rechenaufwand für die Multiplikation einer Matrix mit einem Vektor sehr klein, im Vergleich zum Lösen des Gleichungssystems. Einige iterative Verfahren lösen das System von linearen Gleichungen mit Hilfe von vielen, schnellen Multiplikationen.

3.6.2 Das Gradientenverfahren

Für positiv definite, symmetrische Matrizen ist die Lösung von $\mathbf{A}\vec{x} + \vec{b} = \vec{0}$ auch ein Minimum der Funktion

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

mit dem Gradienten.

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

In Abbildung 8 finden Sie den Graphen einer solchen Funktion und die entsprechenden Niveaueurven. An jeder Stelle nimmt die Funktion $f(\vec{x})$ am schnellsten ab in der Richtung des negativen Gradienten $-\nabla f(\vec{x})$. Somit suchen wir das Minimum der Funktion entlang der Geraden durch den Punkt \vec{x} mit der durch den Gradienten gegebenen Richtung. Das führt, nach einigen Rechnungen, auf den Algorithmus in Tabelle 3.

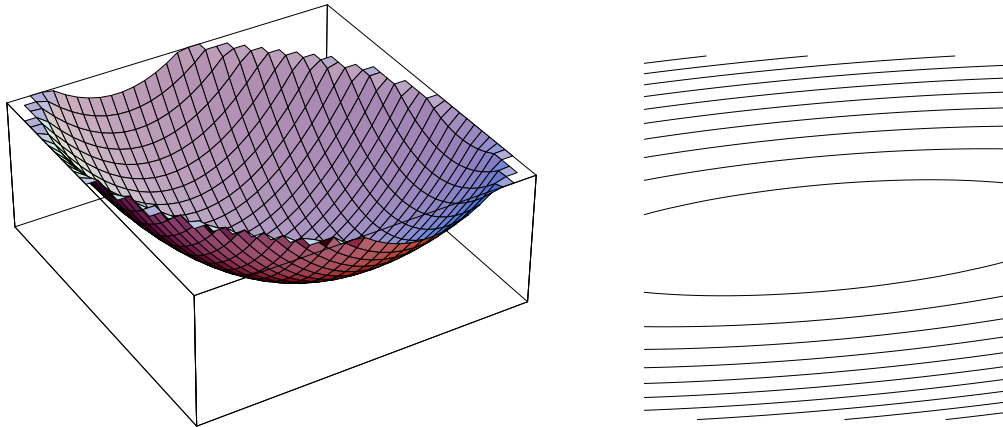


Abbildung 8: Graph der Funktion und die Niveaukurven

Für jeden Iterationsschritt sind (nach sorgfältiger Analyse) die folgenden Rechenoperationen auszuführen

- Eine Multiplikation Matrix*Vektor und zwei Skalarprodukte.
- Zwei Additionen Vektor+Zahl*Vektor
- Speicherbedarf: die Matrix \mathbf{A} und 3 Vektoren.

Hat die Matrix $N \times N$ - \mathbf{A} durchschnittlich in jeder Zeile nz von Null verschiedene Einträge so sind etwa $(4 + nz) N$ Operationen (Multiplikation/Addition) auszuführen für einen Iterationsschritt.

```

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

```

Tabelle 3: Das Gradientenverfahren um $\mathbf{A} \vec{x} + \vec{b} = \vec{0}$ zu lösen

Bei jedem Iterationsschritt wird der Fehler typischerweise mit dem Faktor $\left(1 - \frac{2}{\kappa}\right)$ multipliziert. Man kann zeigen (z.B. [VarFEM]) dass, um D Stellen Genauigkeit zu gewinnen, sind k Iterationsschritte notwendig, wobei

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

Die Matrix \mathbf{A} hat fast immer eine grosse Konditionierungszahl κ und somit sehen die Niveaukurven aus wie in Abbildung 10 und das Verfahren springt im engen Tal hin und her, statt sich schnell dem Minimum anzunähern. Dies führt dazu, dass sehr viele Iterationen notwendig sind um eine genaue Lösung zu erhalten.

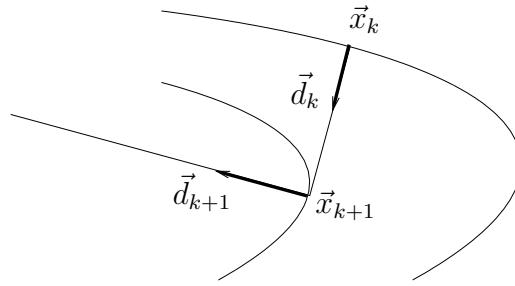


Abbildung 9: Ein Schritt des Gradientenverfahrens

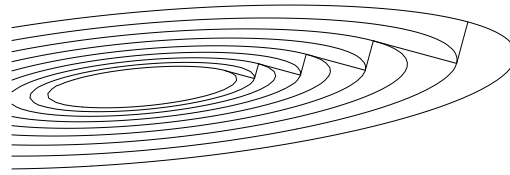


Abbildung 10: Verhalten des Gradientenverfahrens in einem schmalen Tal

3.6.3 Die Methode der konjugierten Gradienten

Durch eine sorgfältige Analyse (siehe z.B. [VarFEM]) kann die Methode so verbessert werden, dass in einem Schritt das Minimum des Tals (2 unabhängige Variablen) gefunden wird. Die Situation ist in Abbildung 11 skizziert. Der resultierende Algorithmus ist in Tabelle 4 aufgeführt.

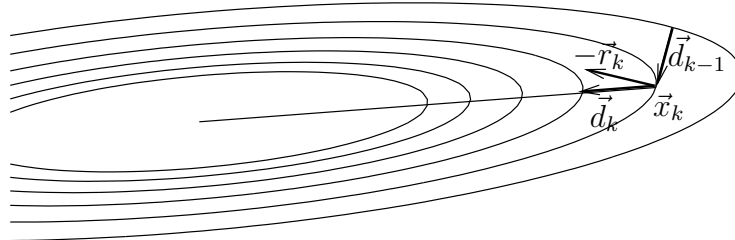


Abbildung 11: Ein Schritt der Methode der konjugierten Gradienten

Für jeden Iterationsschritt sind (nach sorgfältiger Analyse) die folgenden Rechenoperationen auszuführen

- Eine Multiplikation Matrix*Vektor und zwei Skalarprodukte.
- Drei Additionen Vektor+Zahl*Vektor
- Speicherbedarf: die Matrix \mathbf{A} und 4 Vektoren.

Hat die Matrix $N \times N$ - \mathbf{A} durchschnittlich in jeder Zeile nz von Null verschiedene Einträge so sind etwa $(5 + nz)N$ Operationen (Multiplikation/Addition) auszuführen für einen Iterationsschritt. Der Rechenaufwand ist somit nur minimal grösser als für das Gradientenverfahren. Bei jedem Iterationsschritt wird der Fehler typischerweise mit dem Faktor $(1 - \frac{2}{\sqrt{\kappa}})$ multipliziert. Wegen $\sqrt{\kappa} \ll \kappa$ ist dieser Faktor erheblich weiter von 1 entfernt als der Faktor für das Gradientenverfahren und somit konvergiert das Verfahren der konjugierten Gradienten schneller. Man kann zeigen (z.B. [VarFEM]) dass, um D Stellen Genauigkeit zu gewinnen, sind k Iterationsschritte

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

Tabelle 4: Die Methode der konjugierten Gradienten um $\mathbf{A} \vec{x} + \vec{b} = \vec{0}$ zu lösen

notwendig, wobei

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

3.6.4 Ein Modelproblem

Als typisches Modelproblem untersuchen wir eine statische Wärmeverteilung auf einem Quadrat, vernetzt mit $n \times n$ inneren Gitterpunkten.

$$\begin{aligned}
 u_{xx} + u_{yy} &= f(x, y) && \text{im Gebiet} \\
 u(x, y) &= 0 && \text{auf dem Rand}
 \end{aligned}$$

Die resultierende Matrix \mathbf{A} hat je n^2 Zeilen und Spalten und eine halbe Bandbreite von $n+1 \approx n$. In jeder Zeile sind 5 Zahlen ungleich Null. Die Konditionierungszahl ist gegeben durch $\kappa \approx \frac{4}{\pi^2} n^2$. In Tabelle 5 finden Sie den Speicher- und Rechenaufwand für drei gute Algorithmen um das Gleichungssystem $\mathbf{A} \vec{x} = \vec{b}$ zu lösen. Hierbei soll der relative Fehler der Lösung 10^{-D} sein, d.h. D Stellen sind richtig. In der Tabelle 5 und der resultierenden Abbildung 12 (mit $D = 6$) liest man ab, dass:

- das Gradientenverfahren (steepest descent) mit dem Choleskyverfahren vergleichbar ist.
- für kleine $2D$ -Probleme Cholesky effizienter ist als das Verfahren der konjugierten Gradienten.
- für grosse $2D$ -Probleme Cholesky weniger effizient ist als das Verfahren der konjugierten Gradienten.
- für alle $3D$ -Probleme das Verfahren der konjugierten Gradienten effizienter ist als Cholesky.

Um mit Hilfe von Abbildung 12 die Rechenzeiten abschätzen zu können sind die folgenden Angaben nützlich: $1h = 3600 \text{ sec}$, $1 \text{ Tag} \approx 10^4 \text{ sec}$, $1 \text{ Monat} \approx 2.5 \cdot 10^5 \text{ sec}$ und $1 \text{ Jahr} \approx 3 \cdot 10^6 \text{ sec}$.

	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	$9 n^2$	$10 \frac{D \ln 10}{\pi} n^3$	$11 n^3$	$12 \frac{D \ln 10}{\pi} n^4$

Tabelle 5: Vergleich der Algorithmen für das Modelproblem

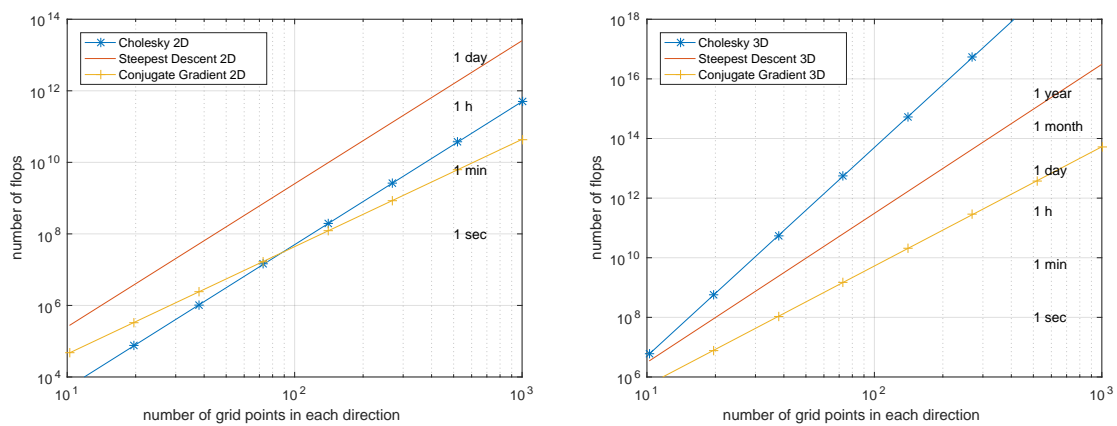


Abbildung 12: Rechenaufwand für Band-Cholesky, Gradientenverfahren und konjugierte Gradienten