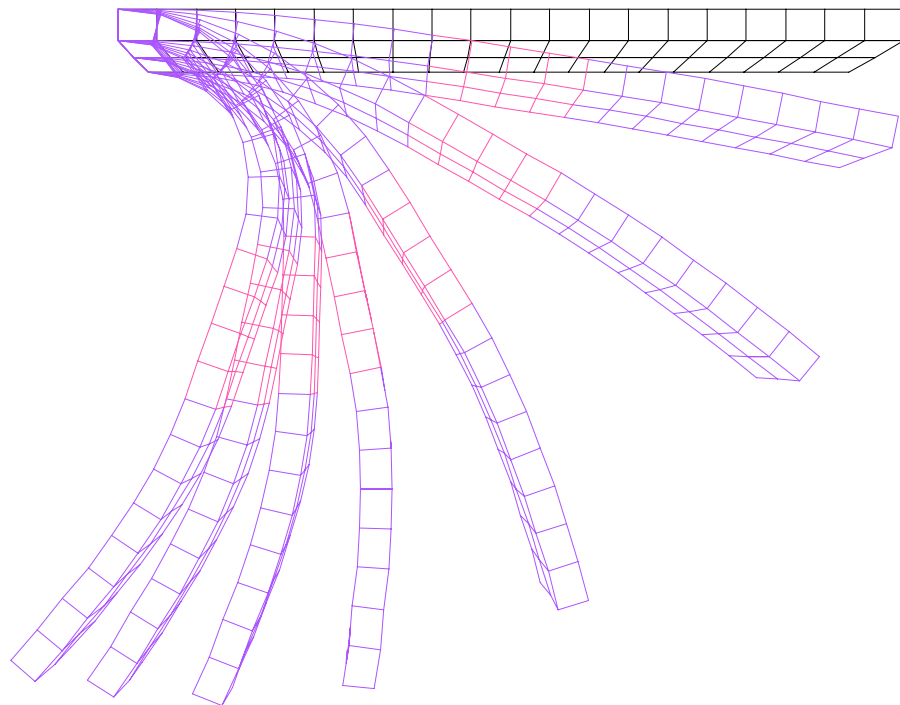


Kopplung starrer und flexibler Strukturbereiche in der Methode der Finiten Elemente

Vertieferarbeit von Axel Volkwein

Juli 1999



Institut für Mechanik

Fakultät für Bauingenieurwesen

Universität Karlsruhe (TH)

Hiermit erkläre ich, daß ich vorliegende Arbeit zum Thema

Kopplung starrer und flexibler Strukturbereiche in der Methode der
Finiten Elemente

selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe. Alle Stellen, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, sind durch Angabe der Quellen als Entlehnungen kenntlich gemacht.

Karlsruhe, den 16. Januar 2006

Für die Betreuung möchte ich mich bei Herrn Prof. Dr. Schweizerhof und bei Herrn Dipl.-Ing. Burkhard Göttlicher herzlich bedanken. Ohne die stete und hilfreiche Unterstützung wäre diese Arbeit so nicht machbar gewesen.

Inhaltsverzeichnis

Einleitung	5
1 Kinematik von Körpern im dreidimensionalen Raum	6
1.1 Geometrische Beschreibungen im Raum	7
1.2 Verbindung von lokalem und globalem Koordinatensystem	8
1.3 Rotationen	9
1.3.1 Drehungen im ebenen System	9
1.3.2 Drehungen im Raum - abhängig von der Drehreihenfolge	10
1.3.3 Drehungen im Raum für kleine Winkel unabhängig von der Drehreihenfolge	11
1.3.4 Große räumliche Rotationen	14
1.3.5 Einführung von Quaternionen	15
2 Starrkörper in der Methode der Finiten Elemente	17
2.1 Behandlung von Verdrehungen im Finite-Element-Modell	18
2.2 Möglichkeiten zur Einbindung von Starrkörpern	20
3 Implementierung von Starrkörpern in das FE-Programm FEAP	23
3.1 Getroffene Annahmen und Einschränkungen	24

3.2	Konzept der Implementierung	24
3.2.1	Aufbau des Eingabedatensatzes	26
3.2.2	Varianten zum Einbau der Kopplung	27
3.3	Schilderung der Änderungen in FEAP	30
3.3.1	Allgemeine Starrkörperorganisation	30
3.3.2	Struktur der Koppelimplementierung	30
3.3.3	Aufbau der Transformationsmatrix	31
3.3.4	Einschränkung wegen Inkompatibilität zu anderen Programm- komponenten	32
4	Berechnungen mit Starrkörpern	33
4.1	Konvergenzverhalten	34
4.2	Belastungsvarianten an einem Kragarm	37
4.2.1	Reine Normalkraft	37
4.2.2	Einachsige Biegung	38
4.2.3	Schiefe Biegung	40
4.2.4	Lasten direkt auf Starrkörper	41
	Schlußbemerkungen	44
	Literaturverzeichnis	46
	A Programmcode	48
	B Eingabedatensatz	53

Einleitung

Mit Hilfe der Methode der Finite Elemente kann man verschiedenste Modelle zur Berechnung von Problemen nahezu jeder Art erstellen. Dabei handelt es sich meist nicht um homogene Systeme. Die Modelle müssen deshalb für eine möglichst realitätsnahe Abbildung in verschiedene Bereiche eingeteilt werden. Dies erfordern meist die unterschiedlichen Materialeigenschaften, wie zum Beispiel der Wärmeleitungskoeffizient oder die Dichte eines Körpers. In strukturmechanischen Berechnungen spielt die unterschiedliche Festigkeit verschiedener Materialien eine große Rolle. Ebenso ist das elastische Verhalten von entscheidender Bedeutung. Es beeinflusst die Größe der Dehnungen und damit die Verschiebungen im Modell.

Unterscheidet sich das elastische Verhalten zweier in ein und demselben Modell verwendeter Materialien um mehrere Größenordnungen, können bei Anwendung des Verschiebungsgrößenverfahrens Numerikprobleme beim Aufstellen der Gesamtsteifigkeitsmatrix auftreten. Man möchte deshalb die nahezu starren Strukturelemente so in ein Finite-Element-Programm implementieren, daß obige numerische Schwierigkeiten nicht mehr vorkommen.

Dazu ist es notwendig, die Knoten der steifen Struktur mit denen der weicheren zu koppeln. Wenn man zudem alle Knoten, die Teil eines Starrkörperelementes sind, miteinander verknüpft, ist es möglich, im modellierten Problem die Anzahl der vorhandenen Freiheitsgrade einzuschränken. Mit der Erstellung einer herkömmlichen Elementsteifigkeitsmatrix ist dies jedoch nicht zu erreichen.

In dieser Arbeit soll nun aufgezeigt werden, wie auf Systemebene Starrkörper in ein Finite-Element-Programm eingebunden werden können. Verwendet wird hierfür das Programm FEAP und es werden statische Systeme im dreidimensionalen Raum behandelt. Dazu wird eine bestehende 2D-Implementierung sinngemäß auf 3D erweitert, es werden jedoch keine kinematischen Ketten berücksichtigt.

Kapitel 1

Kinematik von Körpern im dreidimensionalen Raum

1.1 Geometrische Beschreibungen im Raum

Ein beliebiger Körper läßt sich im Raum auf verschiedene Arten eindeutig beschreiben. Dazu ist es notwendig, die räumliche Position von mindestens einem Punkt des Körpers und die Orientierung seines lokalen Koordinatensystems zum globalen Koordinatensystem zu kennen. Je nachdem, wieviele Punkte bzw. Rotationen zur Lagebeschreibung herangezogen werden, ergeben sich unterschiedliche Möglichkeiten, die Körperposition im Raum eindeutig zu beschreiben:

- Drei – paarweise nicht identische – Punkte
- Zwei verschiedene Punkte und eine Rotation
- Ein Punkt und 3 Rotationen

Ein Punkt läßt sich im Raum auf unterschiedliche Weisen eindeutig im Raum beschreiben. Dies hängt weitgehend vom verwendeten globalen Koordinatensystem ab. Benutzt man Kugelkoordinaten, dann wird der Punkt durch den Skalar r und die Winkelorientierung φ , ψ und χ dargestellt. Im (kartesischen) Koordinatensystem bestimmt sich der Punkt eindeutig durch die Angabe der drei Koordinaten x , y und z .

Während man für die erste Variante vier Parameter zu Beschreibung benötigt, kommt man in der zweiten mit nur drei Parametern aus. Letztere ist deshalb auch für fast alle Anwendungen maßgebend geworden.

Eine Rotation um eine der drei Koordinatenachsen beschreibt man durch die skalare Angabe des Drehwinkels um diese Achse. Der zugehörige Vektor zeigt in Richtung der entsprechenden Koordinatenachse. Die Drehrichtung bestimmt sich aus der 'rechten-Hand-Regel' (Der Daumen zeigt in Richtung des Vektors, die restlichen Finger geben die Drehrichtung an).

Somit benötigt man für die eindeutige Lagebestimmung eines Körpers im Raum maximal 9 und minimal 6 skalare Angaben. Am sinnvollsten, weil am 'sparsamsten', ist demnach die Verwendung von einem Bezugspunkt und dessen Winkelorientierung bezüglich der drei Koordinatenachsen. Als Wahl des Bezugspunktes bietet sich hier der Körperschwerpunkt an. In ihm befinden sich die Massen im Gleichgewicht, was besonders für

dynamische Berechnungen wichtig ist.

Im folgenden wird für allgemeine Körper als Bezugspunkt immer der Schwerpunkt angenommen.

Mit den drei Koordinaten $\begin{bmatrix} x_P^g \\ y_P^g \\ z_P^g \end{bmatrix}$ kann man nun jeden Punkt P bzgl. des globalen Koordinatensystems eindeutig bestimmen. Dazu gehören auch einzelne Punkte des betrachteten Körpers, welche man nun ebenfalls bezüglich dessen Schwerpunktes S bestimmen möchte. Legt man in diesen ein lokales Koordinatensystem parallel zum globalen Koordinatensystem, so läßt sich jeder Körperpunkt durch den lokalen Vektor

$$\mathbf{t}_1^{\mathbf{P}} = \begin{bmatrix} x_g^P - x_g^S \\ y_g^P - y_g^S \\ z_g^P - z_g^S \end{bmatrix} = \begin{bmatrix} x_l^P \\ y_l^P \\ z_l^P \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (1.1)$$

eindeutig beschreiben und die globale Position des Körperschwerpunktes ist

$$\mathbf{r}_g^{\mathbf{S}} = \begin{bmatrix} x_g^S \\ y_g^S \\ z_g^S \end{bmatrix} \quad (1.2)$$

Außerdem ist die globale Darstellung eines beliebigen Körperpunktes darstellbar durch

$$\mathbf{r}_g^{\mathbf{P}} = \mathbf{r}_g^{\mathbf{S}} + \Delta\mathbf{r}_g^{\mathbf{P}} \quad (1.3)$$

Die Bestimmung von $\Delta\mathbf{r}_g^{\mathbf{P}}$ soll im folgenden behandelt werden.

1.2 Verbindung von lokalem und globalem Koordinatensystem

Betrachtet man den Körper in Grundstellung und nimmt an, daß sein lokales Koordinatensystem parallel zum globalen ausgerichtet ist, so gilt

$$\mathbf{r}_g^{\mathbf{P}} = \mathbf{r}_g^{\mathbf{S}} + \mathbf{t}_1^{\mathbf{P}} \quad (1.4)$$

$$\text{In dieser Grundstellung sei nun } \mathbf{t}^0 := \mathbf{t}_1^{\mathbf{P}} \quad (1.5)$$

Wird nun der Körper in eine andere Lage versetzt, muß $\mathbf{r}_g^{\mathbf{P}}$ aktualisiert werden. Der translatorische Anteil $\Delta \mathbf{r}_g^{\mathbf{S}}$ der Lageänderung geht direkt in $\mathbf{r}_g^{\mathbf{S}}$ und damit in $\mathbf{r}_g^{\mathbf{P}}$ ein.

Als nächstes muß $\Delta \mathbf{r}_g^{\mathbf{P}}$ angepaßt werden. Betrachtet man relativ steife Körper, so kann man davon ausgehen, daß innerhalb des Körpers keine Verzerrungen auftreten. Damit bleibt der Abstand $|\mathbf{t}_1^{\mathbf{P}}|$ eines Körperpunkts zu dessen Schwerpunkt immer konstant. Entscheidend für $\Delta \mathbf{r}_g^{\mathbf{P}}$ ist deshalb die erfolgte Drehung des Körpers von \mathbf{t}^0 nach \mathbf{t}^1 .

1.3 Rotationen

1.3.1 Drehungen im ebenen System

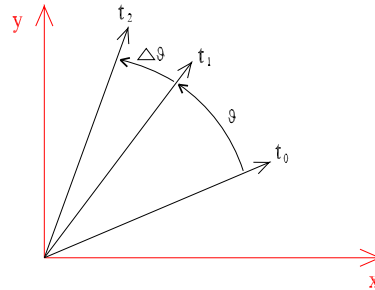


Abbildung 1.1: Drehung in der Ebene

Im zweidimensionalen Fall ist die Drehung eines Vektors \mathbf{t}^0 nach \mathbf{t}^1 um den Winkel ϑ durch die Transformation

$$\mathbf{t}^1 = \begin{bmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{bmatrix} \mathbf{t}^0 \quad (1.6)$$

ausreichend und eindeutig beschrieben. Jede weitere Drehung um $\Delta \vartheta$ ist dann analog

$$\mathbf{t}^2 = \begin{bmatrix} \cos \Delta \vartheta & -\sin \Delta \vartheta \\ \sin \Delta \vartheta & \cos \Delta \vartheta \end{bmatrix} \mathbf{t}^1 = \begin{bmatrix} \cos \vartheta + \Delta \vartheta & -\sin \vartheta + \Delta \vartheta \\ \sin \vartheta + \Delta \vartheta & \cos \vartheta + \Delta \vartheta \end{bmatrix} \mathbf{t}^0 \quad (1.7)$$

Den oben beschriebenen zugehörigen Drehvektor θ kann man sich senkrecht zur zweidimensionalen Zeichenebene vorstellen.

1.3.2 Drehungen im Raum - abhängig von der Drehreihenfolge

Betrachtet man im dreidimensionalen Raum eine Drehung um eine der drei Koordinatenachsen, so hat man zum Beispiel

$$\theta_{\mathbf{x}} = \begin{bmatrix} \vartheta_x \\ 0 \\ 0 \end{bmatrix} \quad (1.8)$$

Die zugehörige Transformationsmatrix für \mathbf{t} lautet

$$\mathfrak{R}_{\mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \vartheta & -\sin \vartheta \\ 0 & \sin \vartheta & \cos \vartheta \end{bmatrix} \quad (1.9)$$

Analoge Transformationen $\mathfrak{R}_{\mathbf{y}}$ und $\mathfrak{R}_{\mathbf{z}}$ erhält man für Rotationen um die beiden anderen Achsen y und z .

Kombiniert man die drei Rotationen, erhält man durch die Gesamttransformation die Verschiebung \mathbf{u}^P des Punktes P in Abhängigkeit von der Schwerpunktsverschiebung \mathbf{u}^S und der Körperrotation θ

$$\begin{aligned} u_x^P &= u_x^S - t_y \sin \vartheta_z + t_z \sin \vartheta_y - t_x(1 - \cos \vartheta_z) - t_x(1 - \cos \vartheta_y) \\ u_y^P &= u_y^S + t_x \sin \vartheta_z - t_z \sin \vartheta_x - t_y(1 - \cos \vartheta_x) - t_y(1 - \cos \vartheta_z) \\ u_z^P &= u_z^S - t_x \sin \vartheta_y + t_y \sin \vartheta_x - t_z(1 - \cos \vartheta_x) - t_z(1 - \cos \vartheta_y) \end{aligned} \quad (1.10)$$

Alternativ läßt sich die Drehung durch eine Gesamtrrotationsmatrix ausdrücken.

$$\mathfrak{R} = \mathfrak{R}_{\mathbf{x}} \mathfrak{R}_{\mathbf{y}} \mathfrak{R}_{\mathbf{z}} \quad (1.11)$$

Bei obigem Vorgehen ist jedoch zu beachten, daß es beim Zusammenbau der Rotationsmatrix auf die Reihenfolge der drei Einzeldrehungen ankommt. Crisfield zeigt dies anschaulich in [5]. Desweiteren ist es erwünscht, eine Matrixbeziehung herzustellen, die nicht mehr von den Winkelfunktionen abhängig ist. Es sollen nur die direkten Winkel auftreten. Hierfür wird eine Linearisierung der Transformation notwendig sein, die weiter unten behandelt wird.

1.3.3 Drehungen im Raum für kleine Winkel unabhängig von der Drehreihenfolge

Mit einer Rotationsmatrix ist es mathematisch möglich, einen Vektor so zu drehen, daß er in jede gewünschte Richtung zeigt. Die Drehung erfolgt um den Fußpunkt des Vektors und seine Länge bleibt unverändert.

Gesucht ist nun eine Möglichkeit, jede beliebige Drehung eindeutig durch eine geeignete Transformationsmatrix zu beschreiben. Dazu betrachtet man nicht mehr einzelne Drehungen um die drei Koordinatenachsen, sondern man bildet aus den drei Drehkomponenten ϑ_x , ϑ_y und ϑ_z einen neuen Drehvektor (siehe auch Abbildung 1.2).

$$\theta = \begin{bmatrix} \vartheta_x \\ \vartheta_y \\ \vartheta_z \end{bmatrix} \quad (1.12)$$

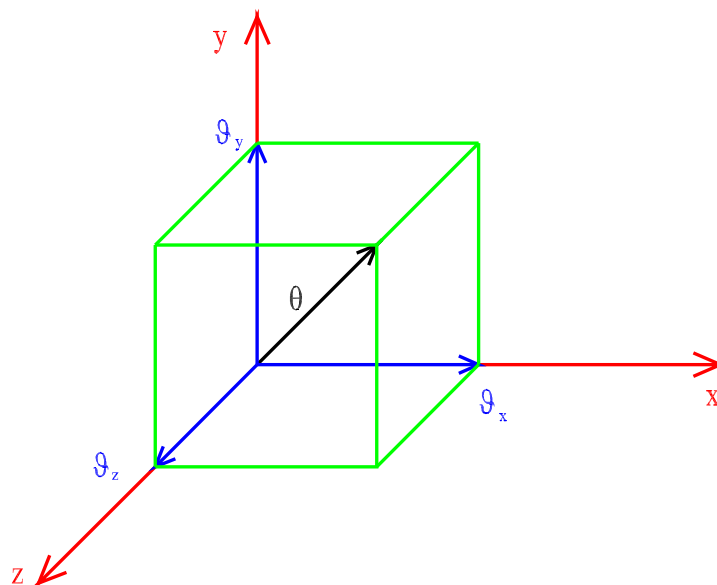


Abbildung 1.2: Zusammensetzung eines räumlichen Drehvektors

Dieser Vektor begründet sich auf einen Satz von Euler [3], der besagt, daß es bei jeder im Raum durchgeführten Drehung eine Gerade gibt, die durch die Drehung auf sich

selbst abgebildet wird. Die Orientierung dieser Gerade entspricht θ . Der entsprechende Drehwinkel θ entspricht der Länge des Vektors $\theta = |\theta|$.

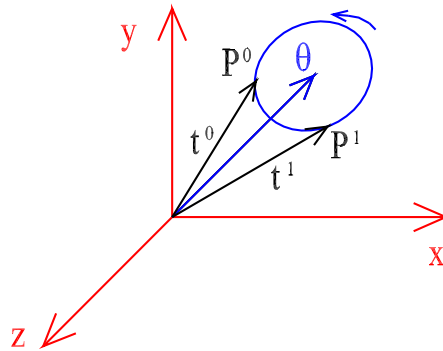


Abbildung 1.3: Drehung eines Vektors im Raum um den Fußpunkt

Die Translation eines Schwerpunktes wird im folgenden vorerst nicht betrachtet. Sie kann später einfach superponiert werden. Hat man deshalb nur eine einfache Drehung im Raum, so verläuft die Wirkungslinie des Drehvektors durch den Schwerpunkt des Körpers. Man kann nun den Vektor \mathbf{t}^0 in eine zu θ senkrechte Komponente \mathbf{t}_\perp und in eine parallele Komponente \mathbf{t}_\parallel zerlegen:

$$\mathbf{t}^0 = \mathbf{t}_\parallel + \mathbf{t}_\perp \quad (1.13)$$

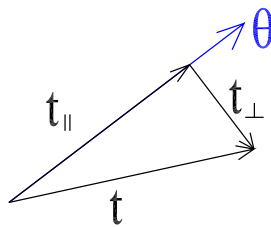


Abbildung 1.4: Zerlegung des gedrehten Vektors bzgl. des Drehvektors

Blickt man nun auf die zu θ senkrechte Drehebene, kann man einen durch die Drehung verschobenen Punkt wie folgt darstellen:

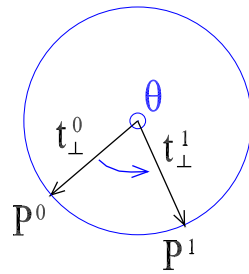


Abbildung 1.5: Ebene Darstellung der räumlichen Dehnung

Der Punkt P hat vor ($= P^0$) und nach ($= P^1$) der Drehung den gleichen Abstand $|\mathbf{t}_\perp|$ zum Drehvektor. Der Verschiebungsvektor $P^0\vec{P}^1$ ist eine Sekante des Kreises um den Drehvektor mit dem Radius $|\mathbf{t}_\perp|$. Bildet man das Kreuzprodukt aus Drehvektor θ und $P\vec{P}'$, so liegt der Ergebnisvektor parallel zur Tangente an den Drehkreis durch den Punkt P . Hier setzt nun die oben erwähnte Linearisierung an: Für kleine Rotationen nimmt man an, daß der sich aus dem Kreuzprodukt ergebende Vektor und $P^0\vec{P}^1$ parallel liegen. Der dabei gemachte Fehler liegt in der Größenordnung von $f = (\theta \times \mathbf{t}_\perp) \sin \vartheta$

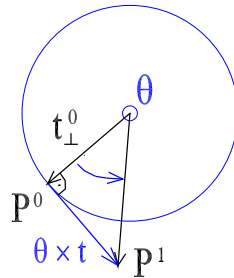


Abbildung 1.6: Linearisierung der Drehung

Da \mathbf{t}_\parallel kollinear zu θ ist, gilt für das Vektorprodukt [3]

$$\theta \times \mathbf{t}_\parallel = \vec{0} \tag{1.14}$$

und damit

$$\theta \times \mathbf{t} = \theta \times (\mathbf{t}_\perp + \mathbf{t}_\parallel) = \theta \times \mathbf{t}_\perp + \theta \times \mathbf{t}_\parallel = \theta \times \mathbf{t}_\perp \tag{1.15}$$

Somit gilt als Näherung für die erfolgte Drehung:

$$\mathbf{t}^1 = \theta \times \mathbf{t}^0 \tag{1.16}$$

Mathematisch läßt sich das Kreuzprodukt auch durch eine Matrizenmultiplikation ersetzen. Hierfür bildet man zu θ eine schiefsymmetrische Matrix $\hat{\theta}$:

$$\hat{\theta} = \begin{bmatrix} 0 & -\vartheta_z & \vartheta_y \\ \vartheta_z & 0 & -\vartheta_x \\ -\vartheta_y & \vartheta_x & 0 \end{bmatrix} \tag{1.17}$$

womit Gl.(1.16) auch wie folgt ausdrückbar ist:

$$\mathbf{t}^1 = \hat{\theta} \mathbf{t}^0 \tag{1.18}$$

1.3.4 Große räumliche Rotationen

Obiges Schema funktioniert nun für beliebige Verdrehungen, jedoch nur für jeweils kleine Winkeländerungen, da sonst der Näherungsfehler zu groß wird.

Behandelt man die Transformation ohne Näherung, gibt es folgende Vorgehensweise (z.B. nach [11]):

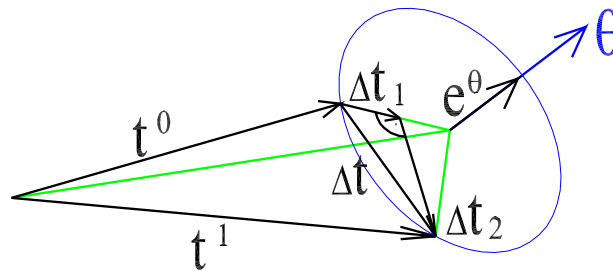


Abbildung 1.7: Räumliche Drehung eines Vektors um seinen Fußpunkt

Man bildet einen Einheitsvektor \mathbf{e}^θ parallel zu θ . Die Verschiebung $\Delta \mathbf{t} = \mathbf{P}^0 \vec{\mathbf{P}}_1$ wird ausgedrückt durch $\Delta \mathbf{t}_1 \parallel \mathbf{t}_\perp$ und $\Delta \mathbf{t}_2 \perp \mathbf{t}_\perp$ mit

$$\Delta \mathbf{t}_2 = (\mathbf{e}^\theta \times \mathbf{t}^0) (1 - \cos \vartheta) \tag{1.19}$$

$$\text{und } \Delta \mathbf{t}_1 = (\mathbf{e}^\theta \times \Delta \mathbf{t}_2) \sin \vartheta \tag{1.20}$$

Das bedeutet für die Transformation:

$$\begin{aligned} \mathbf{t}^1 &= \mathbf{t}^0 + \Delta \mathbf{t}_1 + \Delta \mathbf{t}_2 \\ &= \mathbf{t}^0 + (1 - \cos \vartheta) \mathbf{e}^\vartheta \times (\mathbf{e}^\vartheta \times \mathbf{t}^0) + \sin \vartheta \mathbf{e}^\vartheta \times \mathbf{t}^0 \quad (1.21) \\ \text{mit } \mathbf{e}^\vartheta &= \frac{\boldsymbol{\theta}}{|\boldsymbol{\theta}|} \end{aligned}$$

Mit dieser Formulierung kann man nun auch große Rotationen beschreiben. Allerdings ist diese Variante für Verdrehungen, die größer als $\theta = 180^\circ$ sind, nicht mehr eindeutig.

1.3.5 Einführung von Quaternionen

Dieses Problem kann gelöst werden, wenn man nur noch mit dem halben Winkel $\theta/2$ arbeitet. Dafür werden an Stelle von θ und dessen Winkelfunktionen sogenannte Quaternionen eingeführt:

$$q_0 = \cos \frac{\theta}{2}, \quad q_1 = \frac{\theta_x}{|\boldsymbol{\theta}|} \sin \frac{\theta}{2}, \quad q_2 = \frac{\theta_y}{|\boldsymbol{\theta}|} \sin \frac{\theta}{2}, \quad q_3 = \frac{\theta_z}{|\boldsymbol{\theta}|} \sin \frac{\theta}{2} \quad (1.22)$$

Zerlegt man den Vektor $\mathbf{q}_{\text{ges}} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$ in einen räumlichen Vektor $\mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \end{bmatrix}$ und den

Skalar q_0 , kann man Gl.(1.21) wie folgt umschreiben:

$$\mathbf{t}^1 = \mathbf{t}^0 + 2\mathbf{q} \times (\mathbf{q} \times \mathbf{t}^0) + 2q_0 \mathbf{q} \times \mathbf{t}^0 \quad (1.23)$$

Benutzt man jetzt für \mathbf{q} die Matrixformulierung des Kreuzproduktes (Gl.(1.17)) und die dreidimensionale Einheitsmatrix \mathbf{E} , erhält man folgende Beziehung

$$\mathbf{t}^1 = (\mathbf{E} + 2\mathbf{q}\mathbf{q} + 2q_0\mathbf{q}) \mathbf{t}^0 \quad (1.24)$$

Dies führt letztendlich zu der Rotationsmatrix

$$\mathfrak{R} = \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1q_2 - q_0q_3) & 2(q_1q_2 + q_0q_3) \\ 2(q_1q_2 + q_0q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_2 - q_0q_3) & 2(q_2q_3 + q_0q_1) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix} \quad (1.25)$$

Mit ihr kann man jeden beliebigen Vektor um seinen Fußpunkt in jede gewünschte Position drehen.

Desweiteren gilt die Beziehung:

$$\mathfrak{R}^{-1} = \mathfrak{R}^T \quad \text{und damit} \quad \mathfrak{R}^T \mathfrak{R} = \mathbf{E} \quad (\mathbf{E} = \text{Einheitsmatrix}) \quad (1.26)$$

Das bedeutet, daß die Transponierte \mathfrak{R}^T der Rotationsmatrix den Vektor \mathbf{r}^0 genau entgegengesetzt zu \mathfrak{R} verdreht.

Kapitel 2

Starrkörper in der Methode der Finiten Elemente

2.1 Behandlung von Verdrehungen im Finite-Element-Modell

Ein Starrkörper ist ein Teil eines Finite-Element-Modells, dessen einzelne Bestandteile sich nicht relativ zueinander bewegen können. Man kann sich nun darauf festlegen, daß ein Starrkörper aus mehreren Einzelementen besteht, die, sofern sie sich einander im Modell berühren, wie ein Starrkörper angesehen werden.

Für diesen Starrkörper muß nun ein Bezugspunkt bestimmt werden. Damit kann man sämtliche Innen- und Randpunkte des Starrkörpers in Abhängigkeit der Bezugspunktverschiebungen und der Körperrotationen um diesen Punkt beschreiben.

Wichtig hierbei ist, daß man für die Berechnung nicht nur eine einmalige Verdrehung berücksichtigt, sondern die Möglichkeit hat, für verschiedene Lastschritte und Rechenverfahren die Verdrehung schrittweise immer weiter zu führen. Das Verfahren hierfür sei im folgenden kurz beschrieben: Ein Starrkörper mit der Ruhelage '0' befindet sich in der Position '1' und soll nun in die Position '2' versetzt werden. Es gelten folgende Bezeichnungen:

- \mathbf{u}_p beschreibt die globale Position eines Starrkörperpunktes P
- \mathbf{u}_s ist die globale Position des Starrkörperbezugspunktes S
- \mathbf{t} ist der Vektor von S nach P in Globalkoordinaten
- \mathfrak{R} ist eine Rotationsmatrix (zum Beispiel aus Quaternionen gebildet)

Im Zustand '1' befindet sich P an der Stelle

$$\mathbf{u}_p = \mathbf{u}_s^1 + \mathfrak{R}^1 \mathbf{t}^0 = \mathbf{u}_s^1 + \mathbf{t}^1 \quad (2.1)$$

Durch den Übergang von Zustand '1' nach '2' verschiebt sich der Starrkörper um $\Delta \mathbf{u}$ und verdreht sich um $\Delta \theta$. Mit der Linearisierung aus Gl.(1.16) erhält P den Ortsvektor

$$\mathbf{u}_p^2 = \mathbf{u}_s^1 + \Delta \mathbf{u}_s + \mathbf{t}^1 + \Delta \theta \times \mathbf{t}^1 \quad (2.2)$$

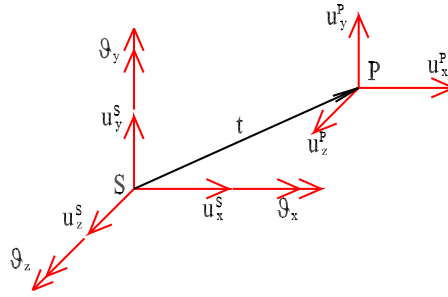


Abbildung 2.1: Freiheitsgrade von Randknoten und Starrkörperschwerpunkt

Unter Zuhilfenahme von Gl.(1.18) ergibt sich eine Änderung von \mathbf{u}_p um

$$\Delta \mathbf{u}_p = \Delta \mathbf{u}_s + \Delta \hat{\boldsymbol{\theta}} \mathbf{t}^1 \quad (2.3)$$

Ziel ist es nun, die Verschiebungen $\Delta \mathbf{u}_p$ in Abhängigkeit von $\begin{bmatrix} \Delta \mathbf{u}_s \\ \Delta \boldsymbol{\theta} \end{bmatrix}$ auszudrücken:

$$\begin{aligned} \begin{bmatrix} \Delta u_p^x \\ \Delta u_p^y \\ \Delta u_p^z \end{bmatrix} &= \Delta \mathbf{u}_s + \begin{bmatrix} 0 & -\Delta \vartheta_x & \Delta \vartheta_y \\ \Delta \vartheta_z & 0 & -\Delta \vartheta_x \\ -\Delta \vartheta_y & \Delta \vartheta_x & 0 \end{bmatrix} \begin{bmatrix} t_x^1 \\ t_y^1 \\ t_z^1 \end{bmatrix} \\ &= \Delta \mathbf{u}_s + \begin{bmatrix} 0 & -\Delta \vartheta_x t_x^1 & \Delta \vartheta_y t_x^1 \\ \Delta \vartheta_z t_y^1 & 0 & -\Delta \vartheta_x t_y^1 \\ -\Delta \vartheta_y t_z^1 & \Delta \vartheta_x t_z^1 & 0 \end{bmatrix} \quad (2.4) \\ &= \Delta \mathbf{u}_s + \begin{bmatrix} 0 & t_z^1 & -t_y^1 \\ -t_z^1 & 0 & t_x^1 \\ t_y^1 & -t_x^1 & 0 \end{bmatrix} \begin{bmatrix} \Delta \vartheta_x \\ \Delta \vartheta_y \\ \Delta \vartheta_z \end{bmatrix} \end{aligned}$$

Somit gilt für $\Delta \mathbf{u}_p$:

$$\begin{bmatrix} \Delta u_p^x \\ \Delta u_p^y \\ \Delta u_p^z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & t_z^1 & -t_y^1 \\ 0 & 1 & 0 & -t_z^1 & 0 & t_x^1 \\ 0 & 0 & 1 & t_y^1 & -t_x^1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u_s^x \\ \Delta u_s^y \\ \Delta u_s^z \\ \Delta \vartheta_x \\ \Delta \vartheta_y \\ \Delta \vartheta_z \end{bmatrix} \quad (2.5)$$

Ist \mathbf{E} eine 3×3 -Einheitsmatrix und $-\hat{\mathbf{t}}^1$ die schiefsymmetrische Matrix zu $\begin{bmatrix} -t_x^1 \\ -t_y^1 \\ -t_z^1 \end{bmatrix}$, erhält man die gesuchte Transformationsmatrix \mathbf{T} zu:

$$\mathbf{T} = \begin{bmatrix} \mathbf{E} & -\hat{\mathbf{t}}^1 \end{bmatrix} \text{ mit } \Delta \mathbf{u}_p = \mathbf{T} \begin{bmatrix} \Delta \mathbf{u}_s \\ \Delta \theta \end{bmatrix} \quad (2.6)$$

Für die vollständige Aktualisierung wird nun $\mathbf{u}_p^2 = \mathbf{u}_p^1 + \Delta \mathbf{u}_p$ gesetzt. Desweiteren wird die Verdrehung θ , die in der Rotationsmatrix \mathfrak{R}^1 gespeichert ist, erweitert um $\Delta \theta$. Dazu bildet man eine Rotationsmatrix zu $\mathfrak{R}^{\Delta \theta}$ und führt ein sogenanntes Quaternionenupdate nach

$$\mathfrak{R}^2 = \mathfrak{R}^{\Delta \theta} \cdot \mathfrak{R}^1 \quad (2.7)$$

durch und erhält die Gesamtverdrehung \mathfrak{R}^2 . Die neue Position von P ist (siehe auch Abb.??)

$$\mathbf{u}_p^2 = \mathbf{u}_s^2 + \mathfrak{R}^2 \mathbf{t}_0 \quad (2.8)$$

2.2 Möglichkeiten zur Einbindung von Starrkörpern

Dieser Abschnitt beschäftigt sich mit den verschiedenen Möglichkeiten, Starrkörper in Finite-Element-Modelle zu implementieren. Die in dieser Arbeit nicht behandelten Varianten werden nicht detailliert, sondern nur anschaulich beschrieben. Sie wer-

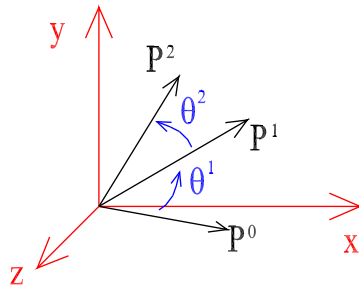


Abbildung 2.2: Mehrfachdrehung eines Vektors im Raum

den größtenteils schon in kommerziellen Programmen verwendet. In erster Linie beschränken sich diese Anwendungen jedoch nur auf Kontaktprobleme.

Einbindung auf Elementebene

Es ist möglich, einen starren Körper durch Elemente mit sehr steifem Materialverhalten zu simulieren. Der Vorteil hierbei ist ein Eingriff in das Programm nur auf Elementebene. Liegt allerdings das Materialverhalten des Starrkörpers etliche Größenordnungen über dem der übrigen Elemente, führt dies beim Zusammenbau der Gesamtsteifigkeitsmatrix zu numerischen Schwierigkeiten.

Selbst wenn man die Numerikprobleme in den Griff bekommen sollte, muß man immer noch abschätzen, auf welche Werte man die einzelnen Komponenten der Elastizitätsmatrix setzt. Diese Abschätzung bzw. Anpassung müßte für jede Berechnung neu durchgeführt werden. Liegen sie zu niedrig, entspricht das Element keinem Starrkörper, liegen sie zu hoch, bekommt man evtl. wieder Schwierigkeiten mit der schlecht konditionierten Gesamtsteifigkeitsmatrix.

Lagrange-Multiplikatoren

Es wird hierbei angenommen, daß alle Knoten, die am Rand eines Starrkörpers liegen, sich während der Berechnung vom Starrkörper losgelöst bewegen. Anschließend werden sie dann durch eine Kraft in entgegengesetzter Richtung zurückgedrückt, bis die Lücke

zum Starrkörper wieder geschlossen ist.

Dieses Verfahren wird zum Beispiel im kommerziellen Finite-Element-Programm ABAQUS angewendet. Benutzt man dort Kontaktbeschreibungen zwischen nachgiebigen und starren Elementen, wird ein Eindringen der Elemente ineinander durch geeignete Lagrange-Multiplikatoren korrigiert.([1])

Penalty-Verfahren

Hier kann man sich vorstellen, daß an Stelle des Starrkörpers eine sehr steife Feder wirkt, die die Verzerrungen innerhalb des Starrkörpers auf ein Minimum reduziert.

Augmented Lagrange

Man kann sich bei dieser Methode eine Kombination aus Penalty-Verfahren und Lagrange Multiplikatoren vorstellen. Die 'Lagrange-Kraft', die die 'Penalty-Feder' wieder zurückdrückt, wirkt so stark, daß die Feder wieder ihre ursprüngliche Dehnung erhält.

Transformierung auf Minimalkoordinaten

Nach Aufstellung der Elementsteifigkeitsmatrix wird diese dahingehend modifiziert, daß die Freiheitsgrade aller Knoten, die einen Starrkörper berühren, in Abhängigkeit der Freiheitsgrade des Starrkörpers gebracht werden. Um dies zu bewirken, müssen die entsprechenden Zeilen und Spalten der Elementsteifigkeitsmatrix angepaßt werden. Damit werden die Verschiebungen aller Starrkörperrandknoten direkt auf dessen Schwerpunkt transformiert und sind fest mit diesem gekoppelt.

Während die ersten vier Verfahren schon als Teil von FEAP verwendbar sind, soll auf das Koppelverfahren in dieser Arbeit eingegangen werden. Die vorhandenen Varianten werden später für Vergleichsrechnungen herangezogen.

Kapitel 3

Implementierung von Starrkörpern in das FE-Programm FEAP

3.1 Getroffene Annahmen und Einschränkungen

Im Rahmen dieser Arbeit wurden folgende Annahmen getroffen:

- Berühren sich zwei Starrkörperelemente, so werden sie wie ein Starrkörper behandelt. Das gilt auch, wenn die Berührstelle nur aus einem einzelnen Knoten besteht. D.h. Starrkörperknoten werden biegesteif miteinander verbunden. Kinetische Ketten, wie sie in [6] behandelt werden, können nicht auftreten.
- Die Berechnung in FEAP kann nur durchgeführt werden, wenn maximal sechs Knoten eines nachgiebigen Elementes an Starrkörper grenzen (Erklärung siehe in Abschnitt 3.3.4)
- Die Berechnung berücksichtigt nur statische Berechnungen. Die Kinetik wird nicht behandelt.
- An Starrkörper angehängte nachgiebige Elemente mit Verzerrungsverhalten erhalten an der Kontaktstelle unter Umständen eine ungewollte Zwangsbedingung. Durch den Starrkörper sind die Knoten der nachgiebigen Elemente in Quer- und Dickenrichtung festgehalten.

3.2 Konzept der Implementierung

Starre Elemente werden als normale 8-Knoten Elemente in einem Eingabedatensatz für ein dreidimensionales Modell eingegeben. Die Freiheitsgrade der Knoten entsprechen den Verschiebungen in den drei Koordinatenrichtungen.

Im Programm geschieht bei der Aufbereitung des Modells folgendes:

- Alle zusammenhängenden Starrelemente werden jeweils einem Starrkörper zugewiesen.
- Es wird bestimmt, ob die einzelnen Elementknoten innerhalb oder am Rand des Starrkörpers liegen.

- Die Masse und die Lage der Schwerpunkte der Starrkörper wird ermittelt.
- Für jeden Starrkörper wird ein neuer Bezugsknoten definiert. Er erhält die Koordinaten des Körperschwerpunktes und speichert dessen drei Verschiebungsfreiheitsgrade. Die Verdrehung eines jeden Starrkörpers wird in Quaternionenform in einem Extrafeld gespeichert. Einen weiteren Knoten ohne explizite Koordinaten kann man zur Speicherung der inkrementellen Rotationen verwenden.
- Kräfte und Verschiebungen, die als Last auf Starrkörperknoten wirken, werden so übersetzt, daß sie als Kräfte, Verschiebungen und Momente auf den jeweiligen Schwerpunkt wirken.
- Beginnt das Programm damit, die globale Steifigkeitsmatrix zusammenzubauen, muß die Elementsteifigkeitsmatrix eines nachgiebigen Elementes nach ihrer Erstellung in der Elementroutine modifiziert werden. Dafür werden alle Zeilen und Spalten der Freiheitsgrade der Starrkörperberührknoten auf die Freiheitsgrade des jeweiligen Schwerpunktes transformiert:

$$\mathbf{u}_1 = \mathbf{T} \mathbf{u}_g \quad (3.1)$$

Die Transformationsmatrix muß also so beschaffen sein, daß man \mathbf{u}_1 auf \mathbf{u}_g abbilden kann und umgekehrt. Mit der entsprechenden Umrechnung für den Lastvektor

$$\mathbf{f}_g = \mathbf{T}^T \mathbf{f}_1 \quad (3.2)$$

möchte man nun auf Elementbasis aus der gegebenen Beziehung

$$\mathbf{K}_1 \mathbf{u}_1 = \mathbf{f}_1 \quad (3.3)$$

die gewünschte transformierte Form gewinnen:

$$\mathbf{K}_g \mathbf{u}_g = \mathbf{f}_g \quad (3.4)$$

Demzufolge muß die Elementsteifigkeitsmatrix nach der Vorschrift

$$\mathbf{K}_g = \mathbf{T}^T \mathbf{K}_1 \mathbf{T} \quad (3.5)$$

modifiziert werden.

- Alle Knoten, die zu einem Starrkörper gehören, werden als aufgelagert markiert. (Modifizierung des id-Feldes)
- Da alle Starrkörperknoten als aufgelagert markiert werden, müssen bei der Aufstellung der globalen Steifigkeitsmatrix die Gleichungszuordnungen (ld-Feld) aktualisiert werden. Die zu den geometrischen Auflagerungen zusätzlich aufgelagerten Freiheitsgrade gehen ansonsten nicht in das Gesamtgleichungssystem ein.
- Wird während der Berechnung in der Elementroutine der Elementlastvektor bestimmt, muß dieser nach Gl.(3.2) angepaßt werden.

3.2.1 Aufbau des Eingabedatensatzes

Bezüglich der allgemeinen Organisation von Starrkörpern in FEAP konnte ich auf schon vorhandene Programmelemente zurückgreifen. Die notwendigen Eingriffe in das Programm FEAP sollten derart gestaltet werden, daß der Benutzer Starrkörper wie jedes Element verwenden kann. Für die Organisation des Speicherbedarfs muß zusätzlich angegeben werden, wieviele Starrkörper maximal im Modell verwendet werden. Für jeden einzelnen werden zwei neue Knoten im System gebildet (siehe oben).

Da es die Möglichkeit gibt, zwischen verschiedenen Implementierungen zu wählen, muß auch das gewünschte Verfahren angegeben werden.

Die Starrelemente werden wie normale 8-Knotenelemente eingegeben. Ihr Materialdatensatz wird an den Elementtyp 71 übergeben. Da im Programm die Anzahl der Knoten pro Element erhöht wurde (siehe 3.2.2), ist es erforderlich, nach der Definition eines jeden Elementes eine Leerzeile einzufügen. FEAP versucht dann nicht mehr, die noch unbesetzten Knotenpositionen eines Elementes mit den Angaben des nächsten Elementes zu belegen.

Die Berechnung erfolgt im Rahmen dieser Arbeit iterativ, da geometrisch nichtlineare Ergebnisse produziert werden sollen und da wegen der angewendeten Linearisierung eine iterative Lösung der FE-Probleme erforderlich ist. Nach dem Lösen des gesamten Gleichungssystems prüft der Löseralgorithmus, ob die gefundene Lösung dem gewünschten Gleichgewicht entspricht. Dazu wird die gesamte rechte Seite der

ursprünglichen Gleichung jeweils mit den Reaktionsvektoren verglichen, die elementintern aus der Zuordnung $\mathbf{f} = \mathbf{K} \mathbf{d}$ ermittelt werden.

Mit der Differenz aus diesen Werten wird eine Norm gebildet. Liegt sie unterhalb einer vom Benutzer gewählten Grenze, gilt die Gleichung als konvergiert. Die maximale Iterationsanzahl wird ebenfalls vom Benutzer vorgegeben.

Ein Beispiel für einen lauffähigen Datensatz (Kragarm mit schiefer Kraft am freien Ende) ist in Anhang B zu sehen.

3.2.2 Varianten zum Einbau der Kopplung

Variante 1

Die Anzahl der Knoten pro Element (nen) wird um zwei erhöht. Im ix-Feld (Angabe der globalen Knotennummern eines jeden Elements) erweitert man die bestehenden Einträge dementsprechend um die Knotennummern, die für jeden Starrkörper anfangs neu kreiert wurden. Dorthin speichert man die Verschiebungen des zum Berührknoten zugehörigen Starrkörperschwerpunktes und die Starrkörperverdrehungen.

Nach den oben beschriebenen Annahmen gilt, daß keine zwei verschiedenen Starrkörper an einem Knoten angreifen können. Das ergibt maximal so viele Starrkörper an einem Element, wie jenes Knoten hat. Auf der sicheren Seite liegend verdoppelt man deshalb das ix-Feld ($nen = 2nen$).

Allerdings wird durch diese Manipulation das ix-Feld der Berührelemente so verfälscht, daß diese Elemente vom Programm/User nicht mehr so angesprochen werden können, wie dies ohne Starrkörper der Fall wäre. Aus diesem Grund wird die beschriebene Variante verworfen.

Variante 2

In der nächsten Variante wird die Anzahl der Knoten pro Element ebenfalls verdoppelt, wodurch sich auch die Größe der Elementsteifigkeitsmatrix verdoppelt. Das ix-Feld wird jedoch nicht verändert. Durch die Transformationsmatrix werden die Freiheitsgrade des zugehörigen Starrkörpers eines Berührknotens in die Elementsteifigkeitsmatrix

eingebaut. Die Starrkörperverschiebungsfreiheitsgrade werden an Stelle der Knotenverschiebungen eingesetzt und die Verdrehung findet Platz im neu geschaffenen Bereich der Elementsteifigkeitsmatrix.

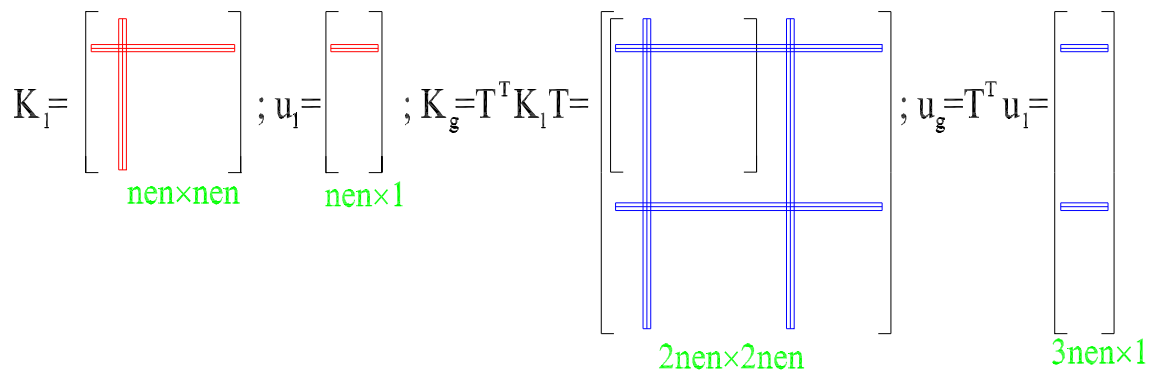


Abbildung 3.1: Modifikation der K- und u-Matrix für Implementierungsvariante 2

Eine entsprechende Transformation muß, wie schon im vorigen Abschnitt beschrieben, auch für die Berechnung des Lastvektors durchgeführt werden. Desweiteren ist eine Anpassung von id- und ld-Feld erforderlich, da alle Starrkörperknoten als aufgelagert betrachtet werden und sich die Gleichungszuordnung im globalen Gleichungssystem geändert hat.

Das ld-Feld existiert jedoch schon vor der Erstellung der Elementsteifigkeitsmatrix, d.h. das Profil der Gesamtsteifigkeitsmatrix existiert schon und soll nachträglich nicht mehr geändert werden.

Man kann dem abhelfen, indem man die entsprechenden Felder schon zu Beginn des Berechnungsvorganges entsprechend abändert. Allerdings müssen vor jedem Aufruf der Elementroutine die veränderten Einträge wieder so aufbereitet werden, daß innerhalb des Elementes so gearbeitet werden kann, als ob die Starrkörper nicht existieren würden. Dies ist relativ umständlich und nicht sehr elegant.

Variante 3

Man erweitert nen von vornherein auf $nen = 3nen$. Die Elementsteifigkeitsmatrix wird nun relativ zu der ursprünglichen sehr groß, jedoch kann man so sicherstellen, daß die Informationen der Knoten des Elements nicht verändert werden. Die sechs Freiheitsgrade des Starrkörpers werden nicht mehr zur einen Hälfte im Bereich der ursprünglichen Elementsteifigkeitsmatrix und zur anderen Hälfte in ihrem neu kreierten Teil gespeichert. Statt dessen werden alle im neu gebildeten Bereich abgelegt.

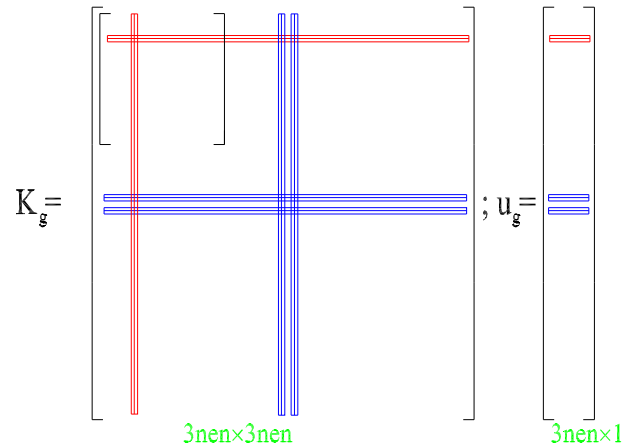


Abbildung 3.2: Modifikation der K- und u-Matrix für Implementierungsvariante 3

Innerhalb eines Elements werden nur die Informationen für die tatsächliche Anzahl von Knoten für dieses Element verarbeitet. Dies gilt für die Freiheitsgrade, die Steifigkeitsmatrix und den Lastvektor.

Diese Variante erfordert durch die großen Matrizen zwar einen erheblich höheren Rechenaufwand, der aber dadurch kompensiert wird, daß für die übrigen Elemente innerhalb des Programms keine Veränderungen auftreten.

3.3 Schilderung der Änderungen in FEAP

In Abschnitt 3.3.4 ist eine Einschränkung des Verfahrens beschrieben. Vorerst soll das Verfahren aber beschrieben werden, als ob diese Einschränkung keine Gültigkeit hätte.

3.3.1 Allgemeine Starrkörperorganisation

Zu Beginn der Berechnung wird abgeprüft, ob im Problem Starrkörper vorhanden sind. Ist dies der Fall, wird die Anzahl von Knoten pro Element (nen) verdreifacht. Für die zusätzlich gebildeten Starrkörperschwerpunktknoten wird die Anzahl der Knoten im Problem um die doppelte maximal vorkommende Anzahl von Starrkörpern erhöht ($numnp = numnp + 2 * mnrb$)

Im Rahmen der Aufbereitung des Datensatzes für die Berechnung startet die allgemeine Starrkörperverwaltung. Die Programmelemente hierfür waren schon vorhanden:

- Ermittlung von Masse und Schwerpunkt der Starrkörper
- Bestimmung und Zuweisung der Bezugsknoten für jeden Starrkörper
- Übertrag von Kräften auf Starrkörperknoten auf den entsprechenden Schwerpunkt
- Zuordnung Starrelemente \rightarrow Starrkörper
- Zuordnung Knoten \rightarrow Starrkörper
- Anlegen der Quaternionenorganisation
- u.a.

3.3.2 Struktur der Koppelimplementierung

In FEAP gibt es für statische Berechnungen drei Aufrufe der Elementroutine, die eine anschließende Transformation benötigen. Einmal ist es die Erstellung der Elementsteifigkeitsmatrix für den Zusammenbau der globalen Steifigkeitsmatrix und zweimal muß der im Element ermittelte Lastvektor angepaßt werden. Das erfordert jeweils einmal einen Aufruf des im Anhang A beigefügten Unterprogramms.

In diesem wird nach einer Prüfung, ob eine Transformierung überhaupt nötig ist, zuerst eine Transformationmatrix \mathbf{T} erstellt (siehe Abschnitt 3.3.3). Je nach Zeitpunkt des Aufrufs wird nun entweder die Elementsteifigkeitsmatrix nach Gl.(3.5) oder der nach Gl.(3.2) modifizierte Elementlastvektor berechnet.

3.3.3 Aufbau der Transformationsmatrix

Für diesen Abschnitt beschreibe \mathbf{E} eine $[3 \times 3]$ -Einheitsmatrix und $\mathbf{0}$ eine $[3 \times 3]$ -Nullmatrix. Mit *skew* ist nach Abschnitt 1.3.3 die schiefsymmetrische Matrix $-\hat{t}$ zum Vektor $-\mathbf{t} = -\mathfrak{R} \mathbf{t}_0$ gemeint.

Für Elemente, die keine Starrkörper berühren, besteht die $[24 \times 72]$ -Transformationsmatrix aus einer $[24 \times 24]$ -Einheitsmatrix; die noch unbesetzten Stellen einer jeden Zeile werden mit Nullen aufgefüllt.

$$\mathbf{T}_{\text{neutral}} = [\mathbf{E}_{24 \times 24} \quad \mathbf{0}_{24 \times 48}] \quad (3.6)$$

Berührt nun ein Knoten einen Starrkörper, so müssen seine drei Freiheitsgrade auf diejenigen des Körpers transformiert werden. Die entsprechenden 'Einsen' der in $\mathbf{T}_{\text{neutral}}$ enthaltenen Einheitsmatrix werden zu Null gesetzt. Im bisherigen 'Nullbereich' von $\mathbf{T}_{\text{neutral}}$ wird nun die Transformationsmatrix nach Gl.(2.6) in die gleichen Zeilen eingebaut. Die Spaltenposition richtet sich hierbei danach, um den wievielten Starrkörperberührknoten eines Elements es sich handelt.

Man hätte die Spaltenzuordnung auch so wählen können, daß sie sich nur an der Knotenfolge innerhalb des Elements orientiert. Hierbei wäre es egal, ob auch andere Knoten des Elements Starrkörperberührknoten sind. Davon wurde zum einen wegen weiter unten beschriebener Einschränkung und zum anderen zur Vermeidung von evtl. vorkommenden Nullen in der Gesamtsteifigkeitsmatrix Abstand genommen.

Für ein 8-Knoten-Element, dessen 2. und 6. Knoten einen Starrkörper berühren, ergibt sich zum Beispiel folgende Transformationsmatrix:

$$\mathbf{T} = \left[\begin{array}{cccc|cccc} \mathbf{E} & & & & 0 & 0 & 0 & 0 \\ 0 & & & & \mathbf{E} \text{ skew} & 0 & 0 & \\ & \mathbf{E} & & \mathbf{0} & 0 & 0 & 0 & 0 \\ & & \mathbf{E} & & 0 & 0 & 0 & 0 \\ & & & \mathbf{E} & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & \mathbf{E} \text{ skew} & \\ \mathbf{0} & & & & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 \end{array} \right] \mathbf{0}$$

Abbildung 3.3: Aufbau einer Transformationsmatrix

3.3.4 Einschränkung wegen Inkompatibilität zu anderen Programmkomponenten

Da FEAP in der mir zur Verfügung stehenden Version mit einer Anzahl von Knoten pro Element von $nen > 20$ Probleme hat, wurde $nen = 20$ festgelegt. Die Größe der entsprechenden Steifigkeitsmatrix ist somit $[nst(= nen \cdot ndf = 20 \cdot 3) \times nen] = [60 \times 60]$ und die der Transformationsmatrix $[20 \times 60]$. Diese Reduzierung hat zur Folge, daß nur maximal sechs Knoten eines Elements irgendeinen Starrkörper berühren dürfen. Die Einschränkung sollte generell nicht nötig sein, da FEAP normalerweise mit jedem beliebigen Wert für nen arbeiten können sollte.

Kapitel 4

Berechnungen mit Starrkörpern

Die Berechnungen zur Verifizierung der Starrkörperkopplung werden in diesem Abschnitt mit einem dreidimensionalen Kragarm durchgeführt. Es geht unter anderem auch darum, die FEAP-Ergebnisse mit der analytischen Lösung zu vergleichen.

Der $l_x = 500\text{cm}$ lange Kragarm hat eine Höhe von $l_z = 20\text{cm}$ und eine Breite von $l_y = 200\text{cm}$. Die einseitige Einspannung wird durch Festhalten aller Knoten eines Kragarmendes bzgl. aller ihrer Freiheitsgrade erreicht. Für die Modellierung werden 8-Knoten-Elemente mit drei Verschiebungsfreiheitsgraden pro Knoten verwendet. Das angenommene Material hat einen E-Modul von $E = 21000\text{kN/cm}^2$ und zeigt kein Querkontraktionsverhalten.

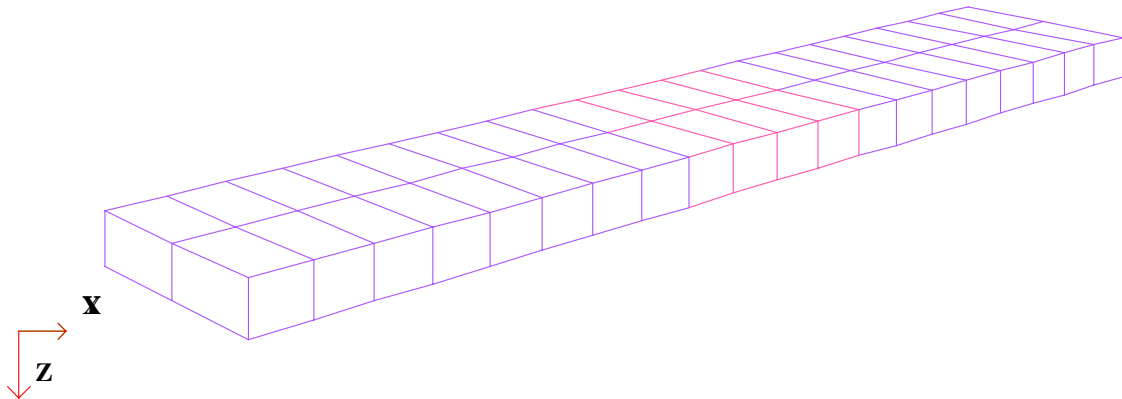


Abbildung 4.1: Kragarmmodell

4.1 Konvergenzverhalten

Das Gleichgewicht der geometrisch nichtlinearen Rechnung wird iterativ gefunden. Ein Kriterium für die Effizienz eines Verfahrens ist die Anzahl der Iterationen und der Weg, den der Löser zur Gleichgewichtsbildung einschlägt. Bringt man in einem Schritt sehr große Lasten auf, geht die Anzahl der benötigten Iterationen unweigerlich in die Höhe. Andererseits zeigt sich hier auch am deutlichsten das Konvergenzverhalten.

Im folgenden Diagramm (Abb.4.2) wurde obiger Kragarm mit einer Normalkraft und auf schiefe Biegung belastet (siehe auch Abschnitt 4.2). Gezeigt werden die Verschiebungskomponenten eines Knotens am freien Kragarmende.

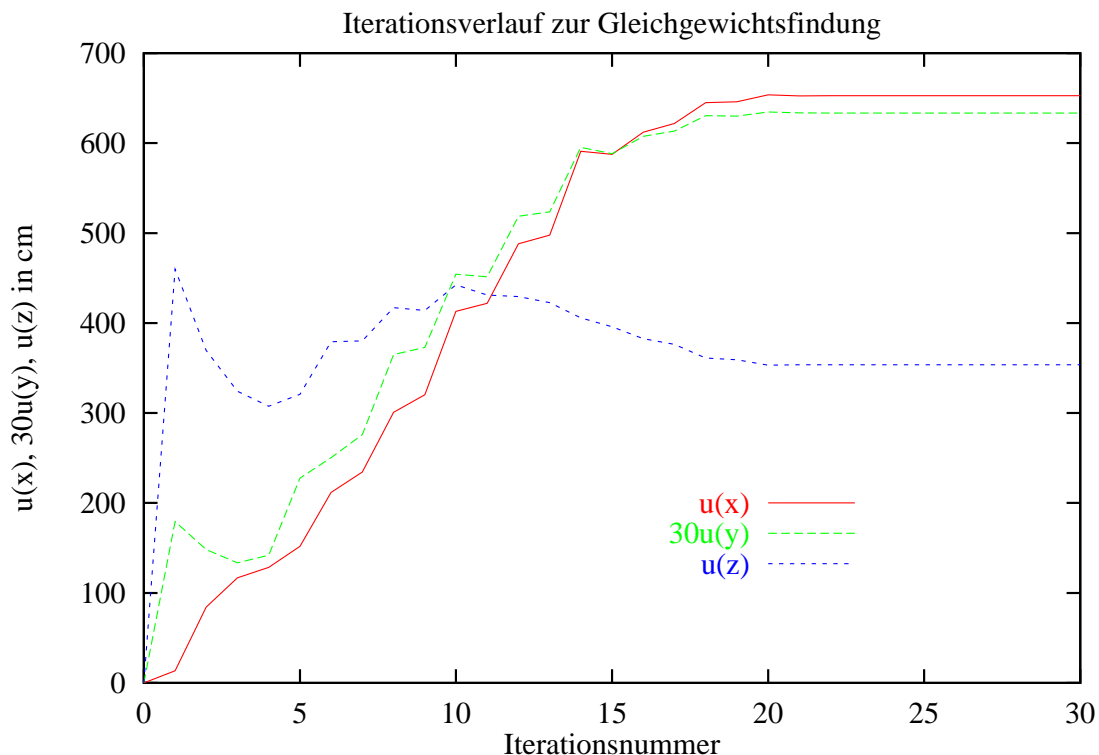


Abbildung 4.2: Iterationsverlauf

Bemerkenswert sind die Verschiebungsspitzen der 1. Iteration in die beiden Querrichtungen, während sich die Längenänderung kaum verändert. Erklären läßt sich dies folgendermaßen: Die aufgebrachten Kräfte bewirken im ersten Schritt eine sehr hohe Verschiebung in y - und z -Richtung, während die Längenänderung nur sehr klein ist (Abb.4.3).

Anschaulich kann man sich dies so vorstellen, daß das Ende nur seitlich verschoben wird, aber seine x -Position kaum verändert. Aufgrund der geometrischen Nichtlinearität würde dies eine zu große Verlängerung des Kragarmes ergeben ($l^1 > l^0$). Um diese in den Griff zu bekommen, wird die Querverschiebung im 2. Iterationsschritt wieder zurückgesetzt.

Interessant ist auch zu sehen, daß die Verschiebung infolge der Biegung um die schwache Achse erst ein höheres Niveau hat und sich dann wieder verringert. Mit zunehmender Krümmung nach unten durch die Vertikalkraft vergrößert sich das Moment um das Auf-

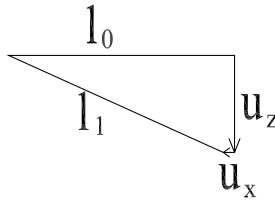


Abbildung 4.3: Geometrische Nichtlinearität im Iterationsablauf

lager infolge der ursprünglichen Normalkraft. Aufgrund der relativ konstanten Länge des Modells bewegt sich das Ende wieder nach oben.

Die Größenordnung der Verschiebung u_y durch die Biegung um die starke Achse liegt etwa um den Faktor 30 unter u_x und u_z . Der Einfluß der geometrischen Nichtlinearität wirkt sich in diese Richtung deshalb nicht so stark aus. Auch das durch die hinzukommende Normalkraftkomponente vergrößerte Moment kann keine großen weiteren Verschiebungen bewirken. Deshalb verläuft u_y -abgesehen von den ersten Iterationen- annähernd parallel zu u_x .

Die Graphen verlaufen im hinteren Abschnitt horizontal. Das bedeutet, daß sich die Ergebnisse der einzelnen Iterationen kaum noch verändern. Die Änderungen in diesem Bereich sind um mehrere Größenordnungen kleiner als die anfänglichen und beeinflussen nur die Exaktheit der Berechnung. Sie hängen nur davon ab, auf welchen Wert die Toleranz für die Berechnung im Eingabedatensatz eingestellt wurde.

Interessant ist auch ein Vergleich des Iterationsverhaltens der einzelnen Starrkörperimplementierungsverfahren. Gezeigt wird deshalb in Abbildung 4.4 jeweils die Verschiebung des Kragarmendes in x-Richtung.

Man kann sagen, daß alle Implementierungsverfahren annähernd gleich schnell konvergieren. Lediglich die Transformierung auf Minimalkoordinaten benötigt am Anfang etwas mehr Schritte, welche dann durch eine etwas schnellere Konvergenz gegen Ende ausgeglichen werden. Nach 30 Iterationen konvergieren alle Verfahren nach der gewählten Toleranz bis auf das Augmented-Lagrange-Verfahren. Bei diesem Verfahren konvergiert zuerst die Lösung ähnlich dem Penaltyverfahren und anschließend erfolgt eine Korrektur durch Lagrange-Multiplikatoren. Die dadurch erzielte Verbesserung des Er-

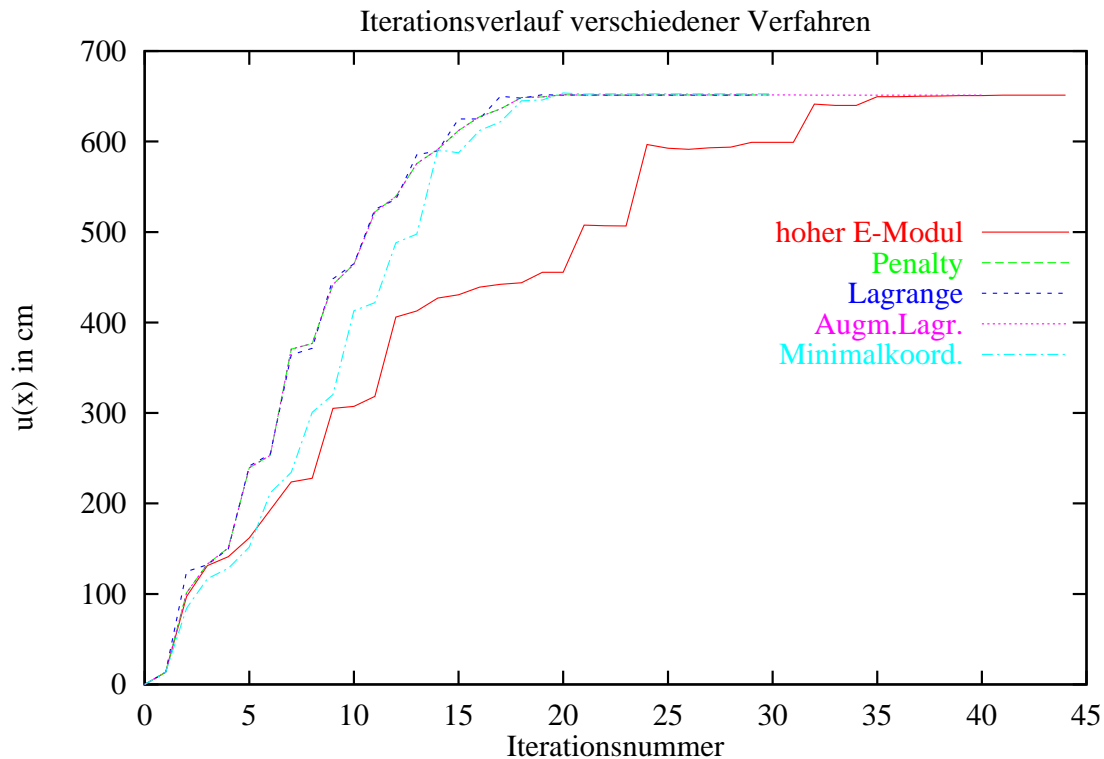


Abbildung 4.4: Iterationsverhalten verschiedener Starrkörperimplementierungen

gebnisses liegt jedoch nicht mehr im sichtbaren Bereich.

Ersetzt man im Modell die Starrkörper durch sehr steife nachgiebige Elemente, benötigt FEAP deutlich mehr Iterationen für das gewünschte Ergebnis.

4.2 Belastungsvarianten an einem Kragarm

4.2.1 Reine Normalkraft

Setzt man an die Knoten des nichteingespannten Kragarmendes eine Gesamtlast von $F = -6 \cdot 10^6 \text{ kN}$, so müßte sich nach der Elastizitätstheorie folgende Endverschiebung in Längsrichtung ergeben:

$$\Delta l = \frac{Fl}{Ea} = \frac{-6 \cdot 10^6 (500 - 100)}{21000 \cdot 20 \cdot 100} = -5.7142 \quad (4.1)$$

Für den Vergleich mit dem von FEAP errechneten Ergebnis wird die Verschiebung der Knoten auf der Querschnittsfläche zwei Elemente weiter innen verwendet, da hier Inhomogenitäten in der Lasteinleitung und damit im Verzerrungsbild ausgeglichener sind. Die analytische Verschiebung für diese Stelle ist $\Delta l = 450/500 \cdot -5.7142 = 5.1428$ und stimmt mit der von FEAP gefundenen und über die Knoten dieses Querschnitts gemittelten Lösung $\Delta l = -5,1114$ recht gut überein. Der Fehler liegt dabei unter einem Prozent.

4.2.2 Einachsige Biegung

Der Kragarm mit einem Starrkörper in der Mitte wird nun mit einer am freien Ende angesetzten Querkraft ($F_z = 18 \cdot 10^3 kN$) um seine schwache Achse gebogen. Die analytische Lösung ergibt hierfür eine Enddurchbiegung von $\Delta z = 45.42857$. Das mit obigem Modell ermittelte Ergebnis lautet $\Delta z = 45.063$ und hat damit mit 0.8% wie bei der reinen Normalkraftbelastung eine Abweichung von weniger als einem Prozent.

Um größere Rotationen zu erhalten, wird nun die Querkraft erhöht, und gleichzeitig wirkt auch eine horizontale Druckkraft auf das Balkenmodell. Die Verformung des Modells zu verschiedenen Zeitpunkten zeigt Abb. 4.5

Zum Vergleich wird das gleiche Modell auch nach anderen in Abschnitt 3.2.2 beschriebenen Starrkörperimplementierungsverfahren berechnet. Nachfolgende Tabelle zeigt die horizontale und vertikale Endverschiebung der Querschnittsmitte des freien Kragarmendes.

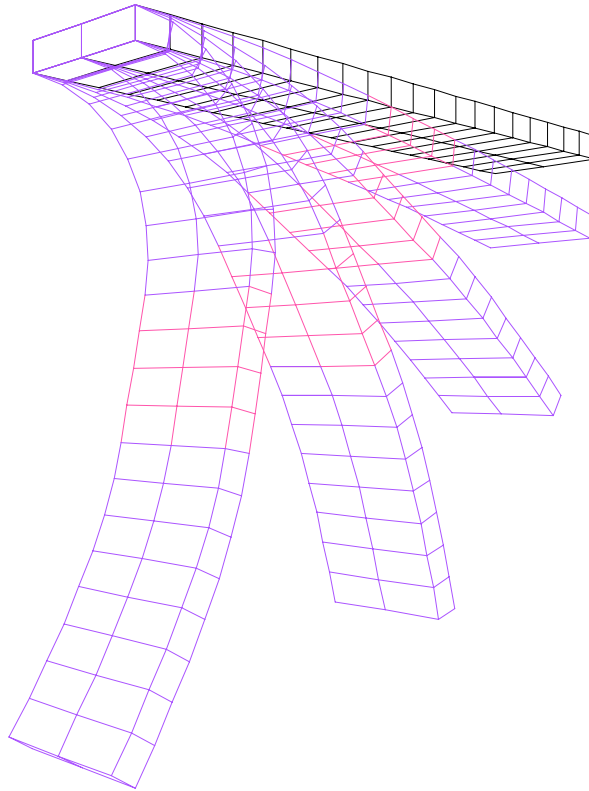


Abbildung 4.5: Durch einachsige Biegung verformter Kragarm

Berechnungsverfahren	Δx [cm]	Δz [cm]
a Gleicher Elastizitätsmodul \forall Elemente	531.577	393.132
b 10-facher Elastizitätsmodul	511.102	418.329
c 100-facher Elastizitätsmodul	508.077	420.874
d 10000-facher Elastizitätsmodul	507.837	421.112
e Penalty-Verfahren	508.079	420.943
f Langrangemultiplikatoren	507.875	421.094
g Augmented Lagrange	507.880	421.094
h Transformierung auf Minimalkoordinaten	507.878	421.094

Man sieht nun schön, daß alle Implementierungsverfahren ((e) bis (h)) nahezu dieselben Ergebnisse liefern. Obwohl die Abweichungen voneinander mit maximal 0.04% verschwindend gering sind, ist zu sehen, daß die mit dem Penalty-Verfahren durch-

geführte Berechnung, welche intern mit großen Steifigkeiten arbeitet, näher an der Berechnung mit mäßig starker Erhöhung des Elastizitätsmoduls (c) liegt. Die restlichen Verfahren stimmen so gut wie überein und zeigen plausible Ergebnisse für den eingebundenen Starrkörper. Die Erhöhung des Elastizitätsmoduls um den Faktor 10000 ist hier numerisch noch machbar und zeigt praktisch die gleichen Ergebnisse wie die Starrkörpermodelle. Deshalb wird dieses Modell im folgenden zum Vergleich bzw. zur Verifizierung der Starrkörperrechnung herangezogen.

4.2.3 Schiefe Biegung

Genauso ist es nun möglich, auf den Kragarm zusätzlich noch eine Kraft aufzubringen, die diesen um die starke Achse krümmt.

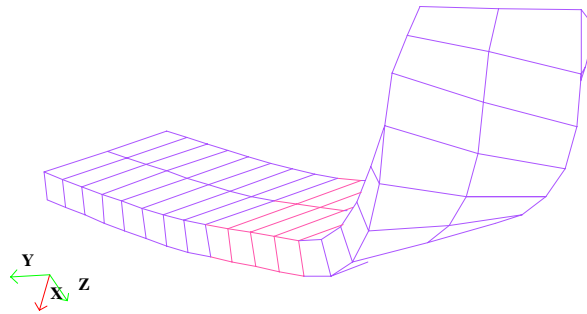


Abbildung 4.6: Kragarm unter schiefer Biegung

Das freie Ende des Modells mit Transformation der Starrkörperknoten auf Minimalkoordinaten hat an einem Knoten den Verschiebungsvektor $\Delta u_{105} = \begin{bmatrix} -464.034 \\ 387.382 \\ 139.960 \end{bmatrix}$.

Die Vergleichsrechnung mit steifen Ersatz-Elastik-Elementen liefert die gleichen Verschiebungen zu $\Delta u_{105} = \begin{bmatrix} -464.035 \\ 387.381 \\ 139.957 \end{bmatrix}$

4.2.4 Lasten direkt auf Starrkörper

Es ist auch möglich, Lasten direkt auf Starrkörperknoten aufzubringen. Diese werden dann so umgerechnet, daß sie als Kräfte und Momente auf den Starrkörperbezugspunkt wirken. Die Kräfte werden genau wie bei anderen Elementen auf die Knoten aufgebracht. Auf diesen Fall soll hier nicht genauer eingegangen werden.

Eine Besonderheit bei den Starrkörpern sind jedoch die Bezugsknoten. In der Starrelementdefinition im Eingabedatensatz ist es bereits möglich, über die Materialeigenschaften zu bestimmen, wo der Bezugspunkt eines Starrkörpers liegen soll. Dieser Bezugspunkt ist FEAP-intern ein ganz normaler Knoten. In ihm sind die Kräfte gespeichert, die auf den Starrkörper wirken. Man kann nun auf zwei zum Schwerpunkt symmetrische Knoten ein Kräftepaar aufbringen, so daß auf den Starrkörper ein reines Moment ohne Querkrafteinfluß wirkt. Dieses Kräftepaar folgt aber globalen Koordinaten, so daß das Moment im verformten Zustand nicht mehr dasselbe ist wie das ursprüngliche.

Alternativ kann man sich deshalb auf den zweiten Knoten konzentrieren, der für jeden Starrkörper neu gebildet wird. Er hat als Knotenlasten die Momente gespeichert, die auf den entsprechenden Starrkörper wirken. Deshalb müßte es theoretisch möglich sein, eine Last auf diesen zweiten Knoten aufzubringen, die dann im Modell als Moment interpretiert wird.

Problem hierbei ist, daß der Benutzer bei Erstellung des Eingabedatensatzes noch nicht weiß, welcher Knoten für die Speicherung der Momente benutzt wird. In der normalen Lasteingabe kann diese also nicht erfolgen. Eine Möglichkeit wäre es, die aufzubringenden Momente im Materialdatensatz für einen Starrkörper unterzubringen. Von dort können sie dann im Rahmen der Modellaufbereitung als Lasten auf den Knoten aufgebracht werden.

Im nachfolgenden Beispiel wurden die Momente auf den Starrkörper am Kragarmende direkt in den Quellcode von FEAP eingearbeitet. Bringt man dort ein Endmoment von $M = 9 \cdot 10^5 \text{ kNcm}$ auf, so bewirkt dies analytisch eine Durchbiegung am Kragarmende von $w = 80.36 \text{ cm}$. Mit FEAP gelangt man zu $w = 79.479 \text{ cm}$. Wenn man bedenkt, daß der Momenteneintrag nicht genau am Ende des Kragarms, sondern auf Höhe des Starrkörperschwerpunktes liegt, ist das Ergebnis sehr genau; die Abweichung

beträgt nur 1%. Für genauere Ergebnisse könnte man die Größe der Starrelemente weiter verringern.

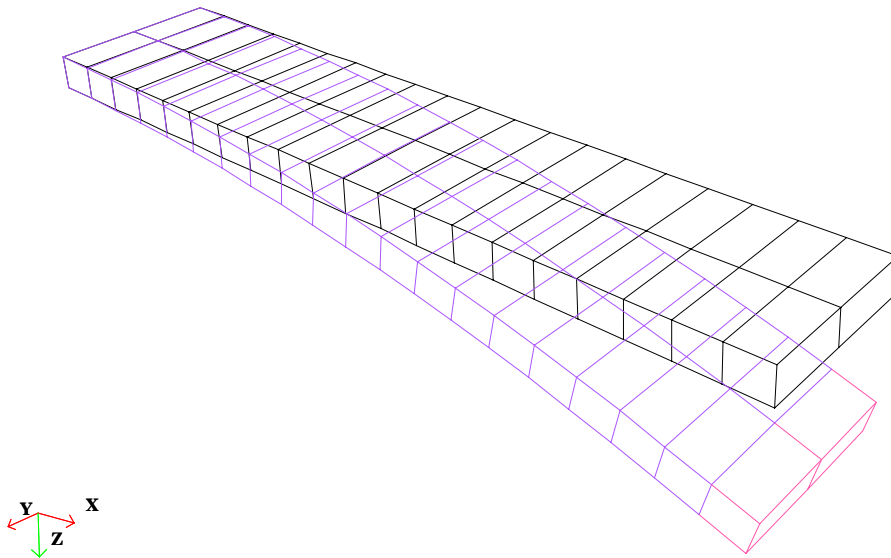


Abbildung 4.7: Kragarm mit Moment am freien Ende

Damit lassen sich nun ohne Probleme Momente in FE-Modelle nur mit Verschiebungsfreiheitsgraden einbringen. Große Verformungen sind dabei auch ohne weiteres machbar.

Vergrößert man das Moment noch weiter, gibt es Schwierigkeiten mit der Konvergenz des Problems. Grund hierfür sind nicht die Starrelemente, sondern die verwendeten nachgiebigen Elemente. Die Verzerrungen innerhalb der Schalenelemente werden zu groß (hourglassing). Dies sieht man teilweise auch schon in Abbildung 4.8. Abhilfe schaffen würde zum Beispiel ein schlankeres Kragarmmodell mit mehr Elementen. Damit stellt die Erzeugung eines Vollkreises kein Problem mehr dar.

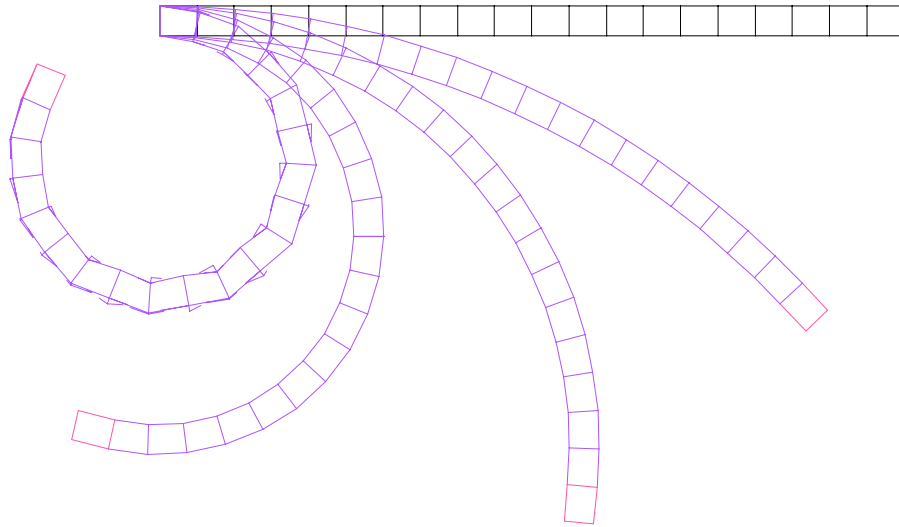


Abbildung 4.8: Durch Endmoment stark verformter Kragarm

Schlußbemerkungen

Zusammenfassung

Starrkörper werden verwendet, um zum Beispiel in strukturmechanischen Berechnungen Bereiche mit sehr steifem Materialverhalten zu ersetzen, da so eine schlecht konditionierte Steifigkeitsmatrix vermieden werden kann. In dieser Arbeit wurden Starrkörper in ein Finite-Element-Modell durch die Transformation auf Minimalkoordinaten implementiert. Dabei wurden die Koordinaten und Verschiebungen der Knoten des Starrkörpers auf diejenigen seines Schwerpunktes transformiert.

Normalerweise werden die Elementeigenschaften in der Materialdefinition eines Eingabedatensatzes für ein Finite-Element-Programm festgelegt. Nun geschieht die Einbindung der Starrelemente nicht mehr auf Elementebene. Die Elementsteifigkeitsmatrix aller an ein Starrelement grenzenden Elemente wird nach ihrer Erstellung so angepaßt, daß die Freiheitsgrade der Berührknoten auf diejenigen des angrenzenden Starrkörpers umgerechnet werden. Das starre Materialverhalten wird somit in geometrische Beziehungen umgewandelt.

Treten im Laufe der Berechnungen große Verformungen auf, kommt man in den Bereich der geometrischen Nichtlinearität. Die verwendete Linearisierung der geometrischen Beziehung liefert auch für große Verschiebungen richtige Ergebnisse.

Die Lageänderung eines Starrkörpers im Raum wird durch die Verschiebungen des Schwerpunktes bzgl. der drei Koordinatenrichtungen und durch die Verdrehung um die Koordinatenachsen beschrieben. Durch die Umrechnung von Kräften auf Starrkörperknoten in Kräfte und Momente auf den Starrkörperschwerpunkt, kann man auch in einem System nur mit Verschiebungsfreiheitsgraden direkt Momente als Belastung aufbringen.

Ausblick

Auf den Schwerpunkt aufgebrachte Momente wurden in dieser Arbeit direkt in den Code von Feap geschrieben. Die Implementierung könnte so erweitert werden, daß die Schwerpunktlasten direkt im Eingabedatensatz z.B. über die Materialeigenschaften eingegeben werden können.

Diese Abhandlung hat nur statische Probleme behandelt. Für eine Vervollständigung der Implementierung fehlt noch die Erweiterung auf kinetische Prozesse.

Die Berechnung der Positionen der Starrkörperknoten basiert auf globalen Koordinaten. Genauso ist es möglich, die geometrische Abhängigkeit in Bezug auf lokale Starrkörperkoordinaten einzubinden.

In den durchgeführten Berechnungen wurde angenommen, daß die verwendeten nachgiebigen Elemente kein Querkontraktionsverhalten aufzeigen. Koppelt man ein Starr-element mit einem nachgiebigen Element mit Querkontraktionseigenschaften, so entstehen ungewollte Zwangsbedingungen. Man könnte die Starrelemente demnach so erweitern, daß sie sich auf Wunsch in Bezug auf das Querkontraktionsverhalten an die benachbarten Elemente anpassen lassen.

Die Einbindung der Starrkörper erfordert zusätzliche Rechenoperationen im Finite-Element-Programm. Gerade durch die hier angewandte Vergrößerung der Steifigkeitsmatrizen bekommt das Gesamtgleichungssystem eine ganz andere Größe. Bei großen Problemen kann sich dies enorm auf die Rechenzeit auswirken. Dazu wäre es interessant zu wissen, ab welchem Verhältnis von nachgiebigen zu starren Elementen sich Starrkörper in einer Berechnung überhaupt rentieren. Ersetzt man die Starrelemente durch nachgiebige Elemente mit erhöhter Steifigkeit erhält man zwar eine schlecht konditionierte Steifigkeitsmatrix, aber die Rechner sind heutzutage so leistungsfähig, daß solche numerischen Schwierigkeiten im normalen Anwendungsfall nicht auftreten.

Literaturverzeichnis

- [1] ABAQUS Theory Manual Version 5.8, Hibbitt, Karlsson & Sorensen, Inc., Pawtucket (1998)
- [2] ABAQUS/Standard User's Manual I-III Version 5.8, Hibbitt, Karlsson & Sorensen, Pawtucket (1998)
- [3] Bronstein, I.N., Semendjajew, K.A., Taschenbuch der Mathematik, Verlag Harri Deutsch, Thun 1976
- [4] Crisfield, M.A., Non-linear finite element analysis of solids and structures, Vol.1: Essentials, John Wiley & Sons (1991)
- [5] Crisfield, M.A., Non-linear finite element analysis of solids and structures, Vol.2: Advanced Topics, John Wiley & Sons (1991)
- [6] Göttlicher, B., Berechnung von Tragwerksmodellen mit starren Bereichen, Diplomarbeit am Institut für Baustatik, Karlsruhe (1989)
- [7] Kugler, J., Finite-Element-Modellierung von Starrkörper- und Übergangsbedingungen in der Statik, Institut für Baustatik, Karlsruhe (1999)
- [8] Schweizerhof, K., Finite Elemente I, Institut für Mechanik, Karlsruhe (1996)
- [9] Wagner, W., Skriptum zu Vorlesung Baustatik III, Institut für Baustatik, Karlsruhe (1996)
- [10] Ward, J.P., Quaternions and Cayley Numbers, Mathematics and Its Applications, Kluwer Academic Publishers, Dordrecht (1997)

- [11] Wittenburg, J., Dynamics of Systems of Rigid Bodies, B.G. Teubner, Stuttgart (1977)

Anhang A

Programmcode

c

```

=====
c   Unterprogramme fuer Vertieferarbeit von Axel Volkwein
=====

c Variablen: Aktueller Knoten      actnod
c             Aktuelles Element    actelem=iel???
c             Aktueller Starrkoerper actrb
c             ix-Feld(nen=max nodes per element,nel)
c             Anzahl Knoten pro Element = nen
c             Knoten eines Elements in ix( (i=1..nen), actelem)
c
c Organisation: Deklaration saemtlicher Variablen in starrvo.f
c               Von dort aus Uebertragung an weitere Routinen
=====
      subroutine starrvo(s,p,u,isw)

c.... declaration of variables

      implicit none

      integer numnp,numel,nummat,nen,neq,neq,nie,ipr,iow,ior
      integer na,nal,nau,nc,nl,nm,nv,nw
      integer nx,nxl,nxu,nul
      integer ndf,ndm,nen1,nst,ndd
      integer nir,nil,nkn,nkf,nco,nma,ntr,nqu,iaug,nrb,nim,nima
      integer nel,nvr,nfr,nsm,iprexp(10),iprk(10),nprpld
      integer platzhalter,i,j,k,isw,impl,mnrb,lkno
      integer n,ma,mct,iel
      integer l,lv,lve,lvs,msk
      integer nrk,nrc,nrm,nrt,nop
      integer nn,n0,n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13
      logical start

      real*8  ttim,dt,c1,c2,c3,c4,c5,pe,pfak,panf,ptol,prop,pra(6,10)
      real*8  wel,theta,dm
      real*8  s(nen*ndf,*),p(nen*ndf),u(ndf,*),T(nen/3*ndf,nen*ndf)

      common /cdata/  numnp,numel,nummat,nen,neq,ipr
      common /prlod/  prop,pra,iprexp,iprk,nprpld
      common /iofile/ ior,iow
      common /sdata/  ndf,ndm,nen1,nst
      common /cdat1/  ndd,nie
      common /tdata/  ttim,dt,c1,c2,c3,c4,c5
      common /mdata/  nn,n0,n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13
      common /ndata/  na,nal,nau,nc,nl,nm,nv,nw
      common /ndatx/  nx,nxl,nxu,nul
      common /ldata/  l,lv,lvs(9),lve(9)
      common /ddata/  theta(3),nrk,nrc,nrm,nrt,nop
      common /eldata/ dm,n,ma,mct,iel,platzhalter

```

```

        common /rbdata/ impl,mnrb,pe,pfak,panf,ptol,lkno
        common /stdata/ nir,nil,nkn,nkf,nco,nma,ntr,nqu,iaug,nrb,nim,
>         nima,nel,nvr,nfr,start,nsm,wel
c       common /psize/  nint

        include 'feap_com.h'

c.... Werte in m-Feld
c       -----
c       id(ndf,*)    -> m(n7)      ix(nen1,*)    -> m(n9)
c       ie(nie,*)    -> m(n5)      urate(nneq,*) -> m(nv)
c       x(ndm,*)     -> m(n8)      f (ndf,*)     -> m(n10)
c       jp           -> m(n12)     f0(ndf,*)     -> m(n13)
c       d(nummat,*) -> m(n6)      d(ndd,*)     -> m(n6)
c       -----
c       irigid(*)   -> m(nir)      ilagr(*)      -> m(nil)
c       kni(*)      -> m(nkn)      comass(3,*)  -> m(nco)
c       knf(3,mks)  -> m(nkf)      ielem(numel) -> m(nel)
c       mass(nrb)   -> m(nma)      traeg(3,3,nrb) -> m(ntr)
c       quat(4,9,nrb)-> m(nqu)     imate(5,mnrb) -> m(nim)
c       vrb(3,2,nrb) -> m(nvr)     frb(3,2,nrb)  -> m(nfr)
c       smate(5,mnrb)-> m(nsm)
c       -----

c.... Erstellung der Transformationsmatrix
        call stcreateT(T,nel,nen,ndm,ndf,numnp,n,m(n9),mnrb
>         ,m(nim),m(n8),u,m(nqu))

        if (isw.eq.3) then
c....   Modifizierung der Elementsteifigkeitsmatrix
            call stmodK(s,T,nen,ndf)

        else if (isw.eq.6) then
c....   Rueckrechnung des Lastvektors fuer Residuuum (fgl=T^T*fel)
            call streF(p,T,nen,ndf)

        endif

        return

c=====
        subroutine stcreateT(T,nel,nen,ndm,ndf,numnp,actelem,ix,mnrb
>         ,imate,x,u,quat)

c.... declaration of variables

        implicit none

        integer nen,ndf,actelem,actnode,ix(nen,*),mnrb,imate(5,*)
        integer numnp,nel,ndm,actelem,actrb
        integer z,i,j,k

```

```

real*8  T(nen/3*ndf, nen*ndf), rbtrans(3), rbrot(3), u(ndf,*)
real*8  v(3), quat(4,9,*), at(3,3), x(ndm,*)

c.... Erstellen einer "neutralen" Transformationsmatrix
do j = 1, nen/3*ndf
  do k = 1, nen*ndf
    t(j,k) = 0.0d0
    if (j.eq.k) then
      T(j,k) = 1.0d0
    endif
  enddo
enddo

do i = 1, nel
c.... Schleife ueber Elementknoten
  actnode = ix (i, actelem)
  actrb = irigid(actnode)
  if (actrb.lt.0) then
c.... Nur Aktion fuer Knoten=Starrkoerper-&Randknoten ( irigid<0 )
  do j = 1,3
cccc  rbtrans(j) = u( j, (2*numnp+imate(3,actrb)) )
cccc  rbrot(j) = u( j, (2*numnp+imate(4,actrb)) )
c.... Vektor von Starrkoerperschwerpunkt zu Randknoten v(3)
  v(j) = x(j,actnode) - x(j,imate(3,actrb))
  at(j) = 0.0d0
  enddo

c.... Bestimmung der schiefsymmetrischen lokalen Matrix at zu v = "R*v"
  call strotmaquat(r,actrb,quat,1) !TRANSPONIERT???
  call stabv(v,r,v,3)
  call stskew(at,v) ! v*(-1) ???

c.... Anpassung der Transformationsmatrix
  do j = 1,ndf
    T( INT((i-1)*ndf+j) , INT((i-1)*ndf+j) ) = 0.0d0
    T( INT((i-1)*ndf+j) , INT(nen/3+(i-1)*ndf+j) ) = 1.0d0
    do k = 1,ndf
      T( INT((i-1)*6+j) , INT(nen/3+(i-1)*ndf+k) ) = at(j,k)
    enddo
  enddo

  endif ! SK-Randknoten
enddo ! Schleife ueber Elementknoten
return

c=====
subroutine stmodK(s,T, nen,ndf)
c.... Modifizierung der Elementsteifigkeitsmatrix S= T^T*Kalt*T

implicit none
integer nen,ndf

```

```
real*8  s(nen*ndf,*),T(nen/3*ndf,*)  
call stabtcb(s,T,s,nen*ndf)  
return
```

```
c=====
```

```
subroutine streF(p,T,ndf,nen)  
c.... Rueckrechnung des Lastvektors fuer Residuum  
  
implicit none  
integer ndf,nen  
real*8  p(nen*ndf,1),T(nen/3*ndf,*)  
  
call stabv(p,T,p,nen*ndf/3,nen)  
  
return
```

```
c=====
```

```
c
```

Anhang B

Eingabedatensatz

feap

126,40,3,3,3,8,,,,,3,4

coor

1,	0,	0.00000,	-50.00000,	-10.00000
2,	0,	25.00000,	-50.00000,	-10.00000
3,	0,	50.00000,	-50.00000,	-10.00000
4,	0,	75.00000,	-50.00000,	-10.00000
5,	0,	100.00000,	-50.00000,	-10.00000
6,	0,	125.00000,	-50.00000,	-10.00000
7,	0,	150.00000,	-50.00000,	-10.00000
8,	0,	175.00000,	-50.00000,	-10.00000
9,	0,	200.00000,	-50.00000,	-10.00000
10,	0,	225.00000,	-50.00000,	-10.00000
11,	0,	250.00000,	-50.00000,	-10.00000
12,	0,	275.00000,	-50.00000,	-10.00000
13,	0,	300.00000,	-50.00000,	-10.00000
14,	0,	325.00000,	-50.00000,	-10.00000
15,	0,	350.00000,	-50.00000,	-10.00000
16,	0,	375.00000,	-50.00000,	-10.00000
17,	0,	400.00000,	-50.00000,	-10.00000
18,	0,	425.00000,	-50.00000,	-10.00000
19,	0,	450.00000,	-50.00000,	-10.00000
20,	0,	475.00000,	-50.00000,	-10.00000
21,	0,	500.00000,	-50.00000,	-10.00000
22,	0,	0.00000,	0.00000,	-10.00000
23,	0,	25.00000,	0.00000,	-10.00000
24,	0,	50.00000,	0.00000,	-10.00000
25,	0,	75.00000,	0.00000,	-10.00000
26,	0,	100.00000,	0.00000,	-10.00000
27,	0,	125.00000,	0.00000,	-10.00000
28,	0,	150.00000,	0.00000,	-10.00000
29,	0,	175.00000,	0.00000,	-10.00000
30,	0,	200.00000,	0.00000,	-10.00000
31,	0,	225.00000,	0.00000,	-10.00000
32,	0,	250.00000,	0.00000,	-10.00000
33,	0,	275.00000,	0.00000,	-10.00000
34,	0,	300.00000,	0.00000,	-10.00000
35,	0,	325.00000,	0.00000,	-10.00000
36,	0,	350.00000,	0.00000,	-10.00000
37,	0,	375.00000,	0.00000,	-10.00000
38,	0,	400.00000,	0.00000,	-10.00000
39,	0,	425.00000,	0.00000,	-10.00000
40,	0,	450.00000,	0.00000,	-10.00000
41,	0,	475.00000,	0.00000,	-10.00000
42,	0,	500.00000,	0.00000,	-10.00000
43,	0,	0.00000,	50.00000,	-10.00000
44,	0,	25.00000,	50.00000,	-10.00000
45,	0,	50.00000,	50.00000,	-10.00000
46,	0,	75.00000,	50.00000,	-10.00000

47,	0,	100.00000,	50.00000,	-10.00000
48,	0,	125.00000,	50.00000,	-10.00000
49,	0,	150.00000,	50.00000,	-10.00000
50,	0,	175.00000,	50.00000,	-10.00000
51,	0,	200.00000,	50.00000,	-10.00000
52,	0,	225.00000,	50.00000,	-10.00000
53,	0,	250.00000,	50.00000,	-10.00000
54,	0,	275.00000,	50.00000,	-10.00000
55,	0,	300.00000,	50.00000,	-10.00000
56,	0,	325.00000,	50.00000,	-10.00000
57,	0,	350.00000,	50.00000,	-10.00000
58,	0,	375.00000,	50.00000,	-10.00000
59,	0,	400.00000,	50.00000,	-10.00000
60,	0,	425.00000,	50.00000,	-10.00000
61,	0,	450.00000,	50.00000,	-10.00000
62,	0,	475.00000,	50.00000,	-10.00000
63,	0,	500.00000,	50.00000,	-10.00000
64,	0,	0.00000,	-50.00000,	10.00000
65,	0,	25.00000,	-50.00000,	10.00000
66,	0,	50.00000,	-50.00000,	10.00000
67,	0,	75.00000,	-50.00000,	10.00000
68,	0,	100.00000,	-50.00000,	10.00000
69,	0,	125.00000,	-50.00000,	10.00000
70,	0,	150.00000,	-50.00000,	10.00000
71,	0,	175.00000,	-50.00000,	10.00000
72,	0,	200.00000,	-50.00000,	10.00000
73,	0,	225.00000,	-50.00000,	10.00000
74,	0,	250.00000,	-50.00000,	10.00000
75,	0,	275.00000,	-50.00000,	10.00000
76,	0,	300.00000,	-50.00000,	10.00000
77,	0,	325.00000,	-50.00000,	10.00000
78,	0,	350.00000,	-50.00000,	10.00000
79,	0,	375.00000,	-50.00000,	10.00000
80,	0,	400.00000,	-50.00000,	10.00000
81,	0,	425.00000,	-50.00000,	10.00000
82,	0,	450.00000,	-50.00000,	10.00000
83,	0,	475.00000,	-50.00000,	10.00000
84,	0,	500.00000,	-50.00000,	10.00000
85,	0,	0.00000,	0.00000,	10.00000
86,	0,	25.00000,	0.00000,	10.00000
87,	0,	50.00000,	0.00000,	10.00000
88,	0,	75.00000,	0.00000,	10.00000
89,	0,	100.00000,	0.00000,	10.00000
90,	0,	125.00000,	0.00000,	10.00000
91,	0,	150.00000,	0.00000,	10.00000
92,	0,	175.00000,	0.00000,	10.00000
93,	0,	200.00000,	0.00000,	10.00000
94,	0,	225.00000,	0.00000,	10.00000
95,	0,	250.00000,	0.00000,	10.00000
96,	0,	275.00000,	0.00000,	10.00000
97,	0,	300.00000,	0.00000,	10.00000

98,	0,	325.00000,	0.00000,	10.00000
99,	0,	350.00000,	0.00000,	10.00000
100,	0,	375.00000,	0.00000,	10.00000
101,	0,	400.00000,	0.00000,	10.00000
102,	0,	425.00000,	0.00000,	10.00000
103,	0,	450.00000,	0.00000,	10.00000
104,	0,	475.00000,	0.00000,	10.00000
105,	0,	500.00000,	0.00000,	10.00000
106,	0,	0.00000,	50.00000,	10.00000
107,	0,	25.00000,	50.00000,	10.00000
108,	0,	50.00000,	50.00000,	10.00000
109,	0,	75.00000,	50.00000,	10.00000
110,	0,	100.00000,	50.00000,	10.00000
111,	0,	125.00000,	50.00000,	10.00000
112,	0,	150.00000,	50.00000,	10.00000
113,	0,	175.00000,	50.00000,	10.00000
114,	0,	200.00000,	50.00000,	10.00000
115,	0,	225.00000,	50.00000,	10.00000
116,	0,	250.00000,	50.00000,	10.00000
117,	0,	275.00000,	50.00000,	10.00000
118,	0,	300.00000,	50.00000,	10.00000
119,	0,	325.00000,	50.00000,	10.00000
120,	0,	350.00000,	50.00000,	10.00000
121,	0,	375.00000,	50.00000,	10.00000
122,	0,	400.00000,	50.00000,	10.00000
123,	0,	425.00000,	50.00000,	10.00000
124,	0,	450.00000,	50.00000,	10.00000
125,	0,	475.00000,	50.00000,	10.00000
126,	0,	500.00000,	50.00000,	10.00000

elem

1,	1,	1,	2,	23,	22,	64,	65,	86,	85
2,	1,	2,	3,	24,	23,	65,	66,	87,	86
3,	1,	3,	4,	25,	24,	66,	67,	88,	87
4,	1,	4,	5,	26,	25,	67,	68,	89,	88
5,	1,	5,	6,	27,	26,	68,	69,	90,	89
6,	1,	6,	7,	28,	27,	69,	70,	91,	90
7,	1,	7,	8,	29,	28,	70,	71,	92,	91
8,	1,	8,	9,	30,	29,	71,	72,	93,	92
9,	2,	9,	10,	31,	30,	72,	73,	94,	93
10,	2,	10,	11,	32,	31,	73,	74,	95,	94

11,	2,	11,	12,	33,	32,	74,	75,	96,	95
12,	2,	12,	13,	34,	33,	75,	76,	97,	96
13,	1,	13,	14,	35,	34,	76,	77,	98,	97
14,	1,	14,	15,	36,	35,	77,	78,	99,	98
15,	1,	15,	16,	37,	36,	78,	79,	100,	99
16,	1,	16,	17,	38,	37,	79,	80,	101,	100
17,	1,	17,	18,	39,	38,	80,	81,	102,	101
18,	1,	18,	19,	40,	39,	81,	82,	103,	102
19,	1,	19,	20,	41,	40,	82,	83,	104,	103
20,	1,	20,	21,	42,	41,	83,	84,	105,	104
21,	1,	22,	23,	44,	43,	85,	86,	107,	106
22,	1,	23,	24,	45,	44,	86,	87,	108,	107
23,	1,	24,	25,	46,	45,	87,	88,	109,	108
24,	1,	25,	26,	47,	46,	88,	89,	110,	109
25,	1,	26,	27,	48,	47,	89,	90,	111,	110
26,	1,	27,	28,	49,	48,	90,	91,	112,	111
27,	1,	28,	29,	50,	49,	91,	92,	113,	112
28,	1,	29,	30,	51,	50,	92,	93,	114,	113
29,	2,	30,	31,	52,	51,	93,	94,	115,	114
30,	2,	31,	32,	53,	52,	94,	95,	116,	115
31,	2,	32,	33,	54,	53,	95,	96,	117,	116
32,	2,	33,	34,	55,	54,	96,	97,	118,	117
33,	1,	34,	35,	56,	55,	97,	98,	119,	118
34,	1,	35,	36,	57,	56,	98,	99,	120,	119
35,	1,	36,	37,	58,	57,	99,	100,	121,	120
36,	1,	37,	38,	59,	58,	100,	101,	122,	121

```
37, 1, 38, 39, 60, 59, 101, 102, 123, 122
38, 1, 39, 40, 61, 60, 102, 103, 124, 123
39, 1, 40, 41, 62, 61, 103, 104, 125, 124
40, 1, 41, 42, 63, 62, 104, 105, 126, 125
```

```
boun
```

```
1,0,1,1,1
22,0,1,1,1
43,0,1,1,1
64,0,1,1,1
85,0,1,1,1
106,0,1,1,1
```

```
forc
```

```
21,0,-1000.0,0.0,300.0
42,0,-1000.0,0.0,300.0
63,0,-1000.0,0.0,300.0
84,0,-1000.0,0.0,300.0
105,0,-1000.0,0.0,300.0
126,0,-1000.0,0.0,300.0
```

```
mate
```

```
1,20
21000.0,0.0,0.0,1,1,3,2.0e-5
0.0
2,71
2.0e-5
3,71
2.0e-5
```

```
end
```

```
sols
```

```
0,0
```

```
macr
```

```
taur,open
dt,,10.0
prop,,1
tol,,1.0e-14
taur,time
loop,time1,4
time
loop,iter,250
tang
form
solve
```

```
next,iter
taur,time
disp,,105
next,time1
time
taur,time
taur,end
end
1,1,0.0,10000.0,0.0,0.10,0.0,0.0
stop
```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Bemerkungen:

 Laststeuerungsfunktion: $\text{prop}(t)=a_1+a_2*t+a_3*[\sin(a_4*t+t_{\min})]^1$

 Eingabe: 1,1,tmin,tmax,a1,a2,a3,a4

ribo,,1.0e7,2.0,1.0e-5

