



Technisch-Naturwissenschaftliche
Fakultät

Ein Gebietszerlegungsverfahren für geometrisch nichtlineare Elastizität

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplomingenieur

im Masterstudium

Industriemathematik

Eingereicht von:
Klemens Reindl

Angefertigt am:
Institut für Numerische Mathematik

Beurteilung:
Dr. Clemens Pechstein

Linz, März 2013

Inhaltsverzeichnis

1	Einführung	7
2	Herleitung der Differentialgleichung	9
2.1	Axiom des statischen Gleichgewichts	10
2.2	Das Hook'sche Gesetz für den ebenen Verzerrungszustand	11
3	Herkömmliche Lösungsmethoden	13
3.1	Nichtlineares vs.linearisiertes Modell	13
3.1.1	Herleitung der Variationsformulierung für den linearen Fall	14
3.1.2	Herleitung der Variationsformulierung für den nichtlinearen Fall	15
3.2	Existenz und Eindeutigkeit für den linearen Fall	16
3.3	Lösungstheorie für das nichtlineare Problem	18
3.4	Ein zugehöriges Energieminimierungsproblem	18
3.5	Finite Elemente (linearer Fall)	20
3.6	Lösungsmethoden für das lineare Problem	23
3.7	Lösungsmethoden für das nichtlineare Modell	23
3.7.1	Fixpunktverfahren mit Liniensuche	24
3.7.2	Newtonverfahren	25
4	Floating Frame of Reference Formulation	28
4.1	Gebietszerlegung	28
4.2	Parametrisierung der Verschiebung mittels „Frame“	28
4.2.1	Ansatz für die totale Verschiebung	28
4.2.2	Fixierung der lokalen Verschiebung am Rahmen	30
4.2.3	Betrachten des Verzerrungstensors	32
4.3	Diskretisierung	32
4.4	Linearisierung „um den Frame“	33
4.5	Variablentransformation	33

4.5.1	Notation	33
4.5.2	Restriktionen und Randbedingungen	34
4.6	Das zugehörige Lagrangefunktional	36
4.6.1	Gradient des Lagrangefunktionals	38
4.7	Lösen des Minimierungsproblems via Lagrange Funktional	40
4.7.1	Newton Verfahren	40
4.7.2	Jacobimatrix des Langrangefunktionals	41
4.7.3	Lösung der Newton-Udate Gleichung	44
4.7.4	Struktur der modifizierten Newton-Update Gleichung	45
4.7.5	FETI System	46
4.8	FFRF-FETI-Algorithmus	53
4.9	Vergleich Newton vs. FFRF-FETI	53
4.10	Implementierung	54
5	Numerische Ergebnisse	59
5.1	Untersuchen des Modellfehlers	59
5.2	Konvergenzvergleich	66
5.2.1	Anzahl der Newton-Iterationsschritte	66
5.2.2	Konvergenz des CG-Verfahrens	71
5.2.3	Konvergenzverhalten der Newtonverfahren	72
6	Ausblick	75

Zusammenfassung

Das Ziel dieser Diplomarbeit ist, die geometrisch nichtlineare Elastizitätsgleichung numerisch mit der Finite Elemente Methode zu lösen. Die bekannte Vereinfachung der linearisierten Elastizität ist nur für kleine Verformungen (kleine Kräfte) zulässig. Für große Deformationen ist der quadratische Term im Verzerrungstensor zu groß um vernachlässigt zu werden. Somit muss das Problem nichtlinear gerechnet werden. Das diskretisierte Problem kann mit dem Newton Verfahren gelöst werden. Allerdings muss in jedem Schritt die Tangentensteifigkeitsmatrix neu assembliert werden.

Der Fokus dieser Diplomarbeit liegt darauf, Situationen auszunutzen, in denen die Verschiebung lokal betrachtet (im Vergleich zur unmittelbaren Umgebung) klein ist. Wir verwenden das Konzept der "Floating Frame of Reference Formulation" (FFRF). Dabei wird das Gebiet in Teilgebiete (Frames) zerlegt. Dadurch können wir die gesamte Verschiebung in eine Translation und Rotation des Rahmens und in eine kleine lokale Verschiebung relativ zum Frame zerlegen. Für die lokale Verschiebung kann der Verzerrungstensor linearisiert werden. Das gesamte System ist nach wie vor nichtlinear, aber die Nichtlinearität beschränkt sich auf die Translationen und Rotationen der Rahmen. Um dieses System effizient zu lösen, verwenden wir eine Kombination aus FFRF und "Finite Element Tearing and Interconnecting" (FETI).

Letztere Methode ist ein iterativer Löser von großen Problemen, die auf der Lösung von (kleineren) Problemen auf Teilgebieten basiert. Dies wird auf das FFRF System angewendet. Der Vorteil dieses Zugangs ist der, dass die Steifigkeitsmatrix in jedem Frame konstant bleibt und nur einmal assembliert und faktorisiert werden muss. In jedem Newtonschritt muss nur das nichtlineare System für die Translationen und Rotationen der Frames assembliert und gelöst werden.

Abschließend werden numerische Resultate für ein 2D Problem gezeigt.

Abstract

The goal of this diploma thesis is the numerical solution of the geometrically nonlinear elasticity equation by finite elements. The well-known simplification of linearized elasticity is only valid for small deformations (or small forces). For large deformations, the displacement is large, the quadratic term in the strain tensor cannot be anymore neglected, and so the problem becomes nonlinear. Usually, one solves the corresponding discrete system with the Newton method, where in each step, a new stiffness matrix has to be assembled.

In the scope of the diploma thesis, we want to exploit situations where the displacement is small under a local point of view (i.e. relative to its neighbourhood) and reduce the degree of nonlinearity. We use the established Floating Frame of Reference Formulation (FFRF): by partitioning the computational domain into subdomains (frames), we can divide the total displacement into a rigid displacement of the corresponding frame and a (small) local displacement relative to the frame. For the latter, we are then allowed to linearized the strain tensor. The overall system is still nonlinear, but the nonlinearity is restricted to the (few) rigid displacements of the frames. To solve the system efficiently, we use a combination of FFRF and the finite element tearing and interconnecting (FETI) method. The latter method is an iterative solver for large-scale problems based on the solution of (smaller) problems on subdomains, and it can be adapted to the FFRF system. The advantage of this approach is, that the stiffness matrix of each frame stays constant in the whole scheme, and thus needs to be assembled and factorised only once. In each Newton step, only a small system corresponding to the rigid frame displacements has to be assembled and solved. Finally, we show numerical results for a 2D model problem.

Danksagung

Ein ganz besonderer Dank gilt meinem Betreuer Dr. Clemens Pechstein, der mir die Möglichkeit gegeben hat meine Masterarbeit am Institut für Numerische Mathematik zu schreiben. Darüber hinaus wurde ich bei der Einarbeitung des Materials und der Implementierung immer sehr gut betreut und meine zahlreichen Fragen wurden immer sehr detailliert diskutiert.

Ein Dankeschön gilt auch meinen Eltern und meinen Geschwistern, die mich immer unterstützt haben. Ohne die Hilfe meiner Familie wäre das Studium in dieser Form sicher nicht möglich gewesen.

Zum Schluss möchte ich mich bei meinen Studienkollegen für die tolle gemeinsame Zeit bedanken. Ohne das gemeinsame Bewältigen der Vorlesungen und Übungen wäre es mir nicht möglich gewesen, so viele Facetten des Studiums kennenzulernen. Es werden mir sehr viele Gespräche im und abseits des Studiums in Erinnerung bleiben.

Klemens Reindl
März 2013

1 Einführung

Ziel dieser Masterarbeit ist es, eine numerische Lösung der geometrisch nichtlinearen Elastizitätsgleichung zu berechnen. Die bekannte linearisierte Elastizitätsgleichung liefert nur für kleine Deformationen (kleine Kräfte) eine gute Näherung. Sobald die Deformation groß wird, kann der quadratische Term des Green St. Venant Verzerrungstensors nicht mehr vernachlässigt werden, und es muss zwischen der Langrangeschen und der Eulerschen Betrachtungsweise unterschieden werden. Dadurch ergibt sich ein System, in dem der Gradient der Verschiebung nichtlinear (kubisch) vorkommt.

In dieser Masterarbeit wählen wir zweidimensionale Probleme im ebenen Verzerrungszustand. Weiters betrachten wir als Materialgesetz das (lineare) Hooksche Gesetz.

Es wird ein Überblick über die Herleitung der Differentialgleichungen für den nichtlinearen und den linearisierten Fall gegeben.

Der Fokus dieser Diplomarbeit liegt auf der numerischen Lösung dieser Gleichungen via Finite Elemente. Nach dem Diskretisieren ergibt sich ein nichtlineares Gleichungssystem, welches mit dem gedämpften Newtonverfahren gelöst werden kann. Es stellt sich dabei heraus, dass die Jacobimatrix in jedem Newtonschritt neu assembliert und faktorisiert werden muss. Dies kann vom Rechenaufwand her sehr teuer werden, wenn es sich um große Systeme handelt.

In dieser Masterarbeit wird ein alternatives Näherungsverfahren diskutiert, welches mit FFRF-FETI bezeichnet und im Folgenden näher dargelegt wird.

In vielen Anwendungen ist die Verschiebung nur aus globaler Sicht groß. Lokal (im Vergleich zur unmittelbaren Umgebung) ist die Verschiebung eher klein. Ein solcher Zugang ist in der Literatur als Floating Frame of Reference Formulation (FFRF) bekannt und wird in dieser Masterarbeit verwendet. Dabei wird das Rechengebiet in disjunkte Teilgebiete zerlegt. Diese Teilgebiete werden als Frames bezeichnet. Die gesamte Verschiebung wird nun in eine Rotation und Translation des Frames und in eine Verschiebung relativ zum Frame zerlegt. Ziel ist es, dass die relative Verschiebung klein ist. In vielen Anwendungsbeispielen ist diese Bedingung erfüllbar. Es zeigt sich, dass der Verzerrungstensor lokal nur von der relativen Verschiebung abhängt. Da diese relative Verschiebung klein ist, kann der Verzerrungstensor durch den linearisierten Verzerrungstensor ersetzt werden. Damit reduziert sich der Grad der Nichtlinearität des Problems.

Teile des nichtlinearen Operators sind konstante, lokale Steifigkeitsmatrizen. Dieser Umstand soll im Weiteren ausgenutzt werden. Damit die oben genannte relative Verschiebung eindeutig bestimmt ist bzw. die Zerlegung der gesamten Verschiebung in Rotation, Translation des Frames und die relative Verschiebung bijektiv ist, müssen drei Einträge in der relativen Verschiebung auf jedem Frame geeignet fixiert werden.

Um die Stetigkeit der somit parametrisierten Gesamtverschiebung zwischen den Frames zu garantieren, benötigen wir Zusatzbedingungen. Die Stetigkeit zwischen den Rahmen bzw die Einbindung der Dirichletrandbedingungen und der Sperrung der drei Einträge der relativen Verschiebung erfolgt über Lagrangeparameter. Das durch die Parametrisierung

entstehende System im gedämpften Newtonverfahren ist größer als das ursprüngliche voll-nichtlineare System, welches ebenfalls mit dem gedämpften Newtonverfahren gelöst wird, aber es ergeben sich Vorteile nach einer „kleinen“ Modifikation der Jacobimatrix im parametrisierten System. Das zu lösende System kann nämlich mit bekannten FETI (Finite Element Tearing and Interconnecting) Techniken gelöst werden. Bei der FETI Technik wird das System iterativ gelöst. Allerdings ist hier ein großer Teil (die Einträge der lokalen Steifigkeitsmatrizen) der Matrix, die in jedem Schritt assembliert werden müsste, konstant, und muss nur einmalig assembliert und faktorisiert werden. Lediglich ein kleinerer Teil muss in jedem Schritt neu assembliert und mittels CG-Verfahren gelöst werden. Diese Kombination von FFRF und FETI ist nach unserem Wissen neuartig.

Es wurden die Algorithmen für das linearisierte Problem, für das vollnichtlineare Problem und für die FFRF-FETI Methode in C++ programmiert. Das vollnichtlineare Verfahren wurde implementiert um einerseits eine Referenzlösung zu erhalten, aber auch um die Effizienz des FFRF-FETI Verfahrens anhand von Beispielen zu testen. Hierbei wurde auf vorhandenen Code des Betreuers im Zusammenhang mit FETI aufgebaut, die Implementierung der eigentlichen Algorithmen, sowie die Assemblierung der Matrizen wurde jedoch im Rahmen dieser Masterarbeit vorgenommen. Bei der Form der Gebiete bzw. Teilgebiete beschränken wir uns auf Rechtecke. Weiters wurden die Ergebnisse des vollnichtlinearen Newton Verfahren für das ursprüngliche nichtlineare System denen des modifizierten FETI-FFRF Newton Verfahrens für das FFRF System gegenübergestellt.

2 Herleitung der Differentialgleichung

Im Folgenden sei $\Omega \subset \mathbb{R}^2$ ein Gebiet (d.h. eine offene, zusammenhängende und beschränkte Menge) mit lipschitzstetigem Rand $\Gamma = \partial\Omega$.

Das Gebiet Ω wird als Referenzgebiet bezeichnet. Es modelliert den ursprünglichen Zustand eines Körpers vor der Verformung. Eine Deformation ist eine hinreichend glatte und injektive Abbildung

$$\phi : \bar{\Omega} \rightarrow \mathbb{R}^2$$

mit der Eigenschaft $\det(\nabla\phi(X)) > 0 \forall X \in \bar{\Omega}$. Sie beschreibt den Zustand eines Punktes $X \in \bar{\Omega}$ nach dem Einwirken von entsprechenden Kräften. Die Menge $\phi(\bar{\Omega})$ besteht aus allen Punkten

$$x = \phi(X),$$

wobei $X \in \bar{\Omega}$.

Die mit dem Großbuchstaben X benannten Variablen werden als Lagrange Koordinaten bezeichnet. Die mit Kleinbuchstaben benannten Variablen x werden als Euler'sche Koordinaten bezeichnet und beschreiben den Ort des Punktes X nach der Deformation.

Die Matrix

$$F(X) = \nabla\phi(X) = \begin{pmatrix} \frac{\partial\phi_1}{\partial X_1}(X) & \frac{\partial\phi_1}{\partial X_2}(X) \\ \frac{\partial\phi_2}{\partial X_1}(X) & \frac{\partial\phi_2}{\partial X_2}(X) \end{pmatrix}$$

wird Deformationsgradient genannt.

Die Verschiebung $u(X)$ ergibt sich wie folgt:

$$u(X) = x - X = \phi(X) - X$$

Also:

$$x = X + u(X) = \phi(X)$$

Die Herleitung des Green St. Venant Verzerrungstensors ist im Buch [1] im Kapitel 6 angeführt. Der Term $\nabla\phi(X)^T \nabla\phi(X)$ wird als rechter Cauchy-Green Verzerrungstensor und mit dem Buchstaben $C(X)$ bezeichnet.

$$C(X) = \nabla\phi(X)^T \nabla\phi(X) = F(X)^T F(X)$$

Der symmetrische Tensor $C(x)$ beschreibt die lokale Änderung der Abstände in Abhängigkeit der Verschiebung $u(X)$. Die Abweichung von $C(X)$ vom unbelasteten (kräftefreien) Fall I wird durch den Tensor $E(X)$ gemessen.

$$E(X) = \frac{1}{2} (C(X) - I)$$

Es lässt sich leicht zeigen, dass der Tensor $E(X)$ symmetrisch ist. Der Tensor $E(X)$ wird als Green St. Venant Verzerrungstensor bezeichnet. Der Tensor $E(X)$ kann auch in Abhängigkeit der Verschiebung $u(X)$ angegeben werden:

$$\begin{aligned} E[u](X) &= \frac{1}{2} (C(X) - I) \\ &= \frac{1}{2} (F(X)^T F(X) - I) = \frac{1}{2} ((\nabla u(X) + I)^T (\nabla u(X) + I)) \\ &= \frac{1}{2} (\nabla u(X)^T + \nabla u(X) + \nabla u(X)^T \nabla u(X)) \end{aligned}$$

Der Tensor $E(X)$ ist also nichtlinear in $u(X)$.

2.1 Axiom des statischen Gleichgewichts

Die genaue Herleitung ist wieder in [1] zu finden. Mit dem Kräfte- und Momentengleichgewicht ergibt sich folgender Zusammenhang für die Gleichgewichtsbedingung:

$$-div(FS) = f \quad \text{in } \Omega$$

S ist dabei der Spannungstensor.

Definition 2.1. Die Divergenz einer Matrix $M \in \mathbb{R}^{2 \times 2}$ ist wie folgt definiert:

$$div(M) = \begin{pmatrix} \frac{\partial M_{11}}{\partial x} + \frac{\partial M_{12}}{\partial y} \\ \frac{\partial M_{21}}{\partial x} + \frac{\partial M_{22}}{\partial y} \end{pmatrix}$$

Der Spannungstensor beschreibt die Kraft pro Fläche.

$$S = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}$$

Die Einträge S_{11} und S_{22} beschreiben die Spannung, die in Richtung der jeweiligen Achse auf die dazugehörige Fläche wirken. S_{12} bzw. S_{21} beschreiben, die Spannungen die tangential zur entsprechenden Fläche wirken. f sind die Volumskräfte. Letztere wirken im Inneren des Gebietes Ω (z.B. Schwerkraft). Der Zusammenhang zwischen Spannungs- und Verzerrungstensor wird über das Hook'sche Gesetz beschrieben.

$$S = S(E(u)) \quad \text{in } \Omega$$

Darauf wird im folgenden Kapitel eingegangen.

Natürlich wirken auf das Gebiet Ω nicht nur die Volumskräfte f . Es wirken auch Kräfte auf die Oberfläche bzw. wird die Verschiebung auf einem Teil des Randes fixiert. Es gibt Kräfte, die normal auf die Oberfläche (Rand) Γ_N wirken und eine zusätzliche Verschiebung des Gebietes verursachen. Diese Art der Kräfte führt dann auf die sogenannten Neumannrandbedingungen und werden mit g_N bezeichnet.

Es gilt:

$$(FS)n = g_N \quad \text{auf } \Gamma_N,$$

n bezeichnet dabei den Außennormalvektor auf dem Rand des Gebietes.

Weiters kann die Verschiebung auf einem Teil des Randes des Gebietes Γ_D fixiert werden, sodass keine Verschiebungen auf diesem Teil des Randes möglich sind. Das führt dann auf die sogenannten Dirichletrandbedingungen:

$$u = g_D \quad \text{auf } \Gamma_D$$

Letztendlich erhält man nun folgendes Randwertproblem:

$$\begin{aligned}
 -\operatorname{div}(FS) &= f && \text{in } \Omega \\
 S &= S(E) && \text{in } \Omega \\
 E &= \frac{1}{2}(F^T F - I) && \text{in } \Omega \\
 F &= I + \nabla u && \text{in } \Omega \\
 u &= g_D && \text{auf } \Gamma_D \\
 (FS)n &= g_N && \text{auf } \Gamma_N
 \end{aligned}$$

2.2 Das Hook'sche Gesetz für den ebenen Verzerrungszustand

Der ebene Verzerrungszustand liegt dann vor, wenn die z Komponente (die Komponente in Richtung der 3ten Achse) der Deformation verschwindet. Das Hook'sche Gesetz stellt eine Verbindung zwischen dem Spannungstensor $S(u)$ und dem Verzerrungstensor $E(u)$ her. In unserem Fall betrachten wir ein sogenanntes St. Venant Kirchhoff Material. Diese Eigenschaft des Materials führt dann zu folgendem Gesetz, dem Hook'schen Gesetz, das den Zusammenhang zwischen Spannungs- und Verzerrungstensor beschreibt:

$$S = \overline{\overline{D}}E = \lambda \operatorname{tr}(E)I + 2\mu E \quad (2.1)$$

$\overline{\overline{D}}$ ist hierbei ein Tensor 4ter Ordnung und durch die Materialparameter λ und μ , die sogenannten Lamé-Konstanten, definiert. $\operatorname{tr}(M) = M_{11} + M_{22}$ bezeichnet die Spur der Matrix $M \in \mathbb{R}^{2 \times 2}$.

Oft werden auch physikalisch aussagekräftigere Parameter benutzt. Nämlich das Elastizitätsmodul \tilde{E} (das Elastizitätsmodul wird in der Literatur auch mit dem Buchstaben E bezeichnet, hier wird schon der Verzerrungstensor mit E bezeichnet), und der Querkontraktionszahl oder Poisson Zahl ν . Die beiden Parameterpaare lassen sich durch folgende Formeln ineinander überführen:

$$\begin{aligned}
 \nu &= \frac{\lambda}{2(\lambda + \mu)} \\
 \tilde{E} &= \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu} \\
 \lambda &= \frac{\tilde{E}\nu}{(1 + \nu)(1 - 2\nu)} \\
 \mu &= \frac{\tilde{E}}{2(1 + \nu)}
 \end{aligned}$$

Wegen physikalischer Voraussetzungen ergibt sich, dass $\tilde{E} > 0$ und $0 < \nu < \frac{1}{2}$. Die Bedingungen für \tilde{E} und ν ergeben die Einschränkungen $\lambda > 0$ und $\mu > 0$.

Damit bekommt man:

$$\begin{aligned}
 -\operatorname{div}(FS) &= f && \text{in } \Omega \\
 S &= \overline{DE} && \text{in } \Omega \\
 E &= \frac{1}{2}(F^T F - I) && \text{in } \Omega \\
 F &= I + \nabla u && \text{in } \Omega \\
 u &= g_D && \text{auf } \Gamma_D \\
 (FS)n &= g_N && \text{auf } \Gamma_N
 \end{aligned}$$

Für die Umsetzung im Programm selbst bietet sich eine alternative Schreibweise an. Dies ist die Darstellung des Hook'schen Gesetzes in Vektor und Matrixschreibweise. Die Tensoren S und E sind beide symmetrisch. Das heißt, in jedem Tensor gibt es nur 3 unterschiedliche Einträge. Die 3 unterschiedlichen Einträge der Tensoren S und E werden in die Vektoren \vec{E} und \vec{S} geschrieben. Im weiteren Verlauf werden Vektoren mit \rightarrow gekennzeichnet, um den Unterschied zur Tensorschreibweise darzustellen.

$$\begin{aligned}
 \vec{E} &= \begin{pmatrix} E_{11} \\ E_{22} \\ E_{12} \end{pmatrix} \\
 \vec{S} &= \begin{pmatrix} S_{11} \\ S_{22} \\ S_{12} \end{pmatrix}
 \end{aligned}$$

Wegen der Symmetrie der Tensoren gilt $E_{12} = E_{21}$ und $S_{12} = S_{21}$.

Aus dem Hook'schen Gesetz folgt dann:

$$\begin{aligned}
 S &= \lambda \operatorname{tr}(E) I + 2\mu E \\
 \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} &= \lambda \begin{pmatrix} E_{11} + E_{22} & 0 \\ 0 & E_{11} + E_{22} \end{pmatrix} + 2\mu \begin{pmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{pmatrix}
 \end{aligned}$$

Daraus ergibt sich folgender Zusammenhang in Vektorschreibweise:

$$\begin{pmatrix} S_{11} \\ S_{22} \\ S_{12} \end{pmatrix} = \lambda \begin{pmatrix} E_{11} + E_{22} \\ E_{11} + E_{22} \\ 0 \end{pmatrix} + 2\mu \begin{pmatrix} E_{11} \\ E_{22} \\ E_{12} \end{pmatrix} \quad (2.2)$$

$$\begin{pmatrix} S_{11} \\ S_{22} \\ S_{12} \end{pmatrix} = \begin{pmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & 2\mu \end{pmatrix} \begin{pmatrix} E_{11} \\ E_{22} \\ E_{12} \end{pmatrix} \quad (2.3)$$

$$\vec{S} = D \vec{E} \quad (2.4)$$

wobei

$$\begin{aligned}
 D &= \begin{pmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & 2\mu \end{pmatrix} \\
 &= \begin{pmatrix} \frac{E\nu}{(1+\nu)(1-2\nu)} + 2\frac{E}{2(1+\nu)} & \frac{E\nu}{(1+\nu)(1-2\nu)} & 0 \\ \frac{E\nu}{(1+\nu)(1-2\nu)} & \frac{E\nu}{(1+\nu)(1-2\nu)} + 2\frac{E}{2(1+\nu)} & 0 \\ 0 & 0 & 2\frac{E}{2(1+2\nu)} \end{pmatrix} \\
 &= \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & 1-2\nu \end{pmatrix}
 \end{aligned}$$

3 Herkömmliche Lösungsmethoden

3.1 Nichtlineares vs. linearisiertes Modell

Aus dem vorherigen Kapitel bekommen wir die nichtlineare Elastizitätsgleichung:

$$\begin{aligned}
 -\operatorname{div}(FS) &= f && \text{in } \Omega \\
 S &= \overline{\overline{D}}E && \text{in } \Omega \\
 E &= \frac{1}{2}(\nabla u + \nabla u^T + \nabla u^T \nabla u) && \text{in } \Omega \\
 F &= \nabla u + I \\
 u &= g_D && \text{auf } \Gamma_D \\
 (FS)n &= g_N && \text{auf } \Gamma_N
 \end{aligned}$$

Im linearisierten Fall vernachlässigen wir den nichtlinearen Teil im Spannungstensor E . Dies ist eine gute Näherung, wenn die Verschiebung u nicht sehr groß ist, da dann der quadratische Teil vernachlässigbar klein ist. Den linearisierten Spannungstensor bezeichnen wir mit ϵ .

$$\epsilon = \frac{1}{2}(\nabla u + \nabla u^T)$$

Weiters bezeichnen wir im linearen Fall den Spannungstensor S mit σ . Darüberhinaus wird im linearisierten Modell nicht zwischen Eulerscher und Langrangscher Darstellung unterschieden. Deswegen kann FS durch σ ersetzt werden. Dabei ist der Fehler klein, solange die Verschiebung auch klein ist.

Das Hooksche Gesetz im linearisierten Modell im ebenen Verzerrungszustand:

$$\sigma = \overline{\overline{D}}\epsilon$$

Somit ergibt sich folgende Differentialgleichung im linearisierten Fall:

$$\begin{aligned} -\operatorname{div}(\sigma) &= f && \text{in } \Omega \\ \sigma &= \overline{D}\epsilon && \text{in } \Omega \\ \epsilon &= \frac{1}{2}(\nabla u + \nabla u^T) && \text{in } \Omega \\ u &= g_D && \text{auf } \Gamma_D \\ \sigma n &= g_N && \text{auf } \Gamma_N \end{aligned}$$

Somit hängt σ linear von der Unbekannten u ab.

Wir leiten zuerst die Variationsformulierung für den linearen Fall her. Die Variationsformulierung für den nichtlinearen Fall kann auf ähnliche Weise hergeleitet werden.

3.1.1 Herleitung der Variationsformulierung für den linearen Fall

Die Gleichung

$$-\operatorname{div}\sigma = f$$

soll nun als Variationsproblem angeschrieben werden.

Sei $v = (v_1, v_2) \in V$ eine Testfunktion aus dem zulässigen Raum V , wobei für die Testfunktion folgende Bedingung gelten muss: $v = 0$ auf dem Dirichletrand Γ_D . Die obige Gleichung wird nun beidseitig mit der Testfunktion multipliziert. Danach wird über das gesamte Gebiet Ω integriert. Damit bekommt man folgende Gleichung:

$$-\int_{\Omega} \operatorname{div}\sigma \cdot v dx = \int_{\Omega} f \cdot v dx$$

Der Operator \cdot bezeichnet dabei das herkömmliche euklidische Skalarprodukt. Durch partielle Integration der linken Seite der Gleichung erhält man:

$$\int_{\Omega} \operatorname{div}\sigma \cdot v dx = \int_{\Gamma} \sigma n \cdot v ds - \int_{\Omega} \sigma : \nabla v dx$$

Der Operator $:$ ist wie folgt definiert. Für $d \times d$ Matrizen A, B gilt $A : B = \sum_{i,j=1}^d a_{ij} b_{ij}$. Somit ergibt sich für die eigentliche Gleichung

$$\int_{\Omega} \sigma : \nabla v dx = \int_{\Omega} f \cdot v dx + \int_{\Gamma} \sigma n \cdot v ds$$

Da σ symmetrisch ist, kann man die folgende Vereinfachung durchführen:

$$\sigma : \nabla v = \frac{1}{2}\sigma : \nabla v + \frac{1}{2}\sigma^T : \nabla v = \frac{1}{2}\sigma : \nabla v + \frac{1}{2}\sigma : \nabla v^T$$

Da der Operator $:$ linear ist, bekommt man

$$\sigma : \nabla v = \sigma : \left(\frac{1}{2}(\nabla v + \nabla v^T) \right) = \sigma : \epsilon(v).$$

Da v die homogenen Dirichletbedingungen erfüllt ($v = 0$ auf Γ_D) und $\sigma n = g_N$ auf Γ_N ergibt sich

$$\int_{\Gamma} \sigma n \cdot v ds = \int_{\Gamma_N} \sigma n \cdot v ds = \int_{\Gamma_N} g_N \cdot v ds$$

Somit erhalten wir folgende Gleichung:

$$\int_{\Omega} \sigma : \epsilon(v) dx = \int_{\Omega} f \cdot v dx + \int_{\Gamma_N} g_N \cdot v ds$$

Setzt man die Definition des Spannungstensors σ in Abhängigkeit von u ein, erhält man das folgende Variationsproblem:

Suche ein $u \in V_g = \{v \in V : v = g_D \text{ auf } \Gamma_D\}$, sodass

$$a(u, v) = \langle f, v \rangle$$

für alle $v \in V_0 = \{v \in V : v = 0 \text{ auf } \Gamma_D\}$ mit der Bilinearform

$$a(u, v) = \int_{\Omega} \overline{\overline{D}} \epsilon(u) : \epsilon(v) dx$$

und dem linearen Funktional

$$\langle f, v \rangle = \int_{\Omega} f \cdot v dx + \int_{\Gamma_N} g_N \cdot v ds$$

Die Wahl der Räume V , V_0 und V_g ergibt sich aus den Erfordernissen an u und v . In unserem Fall, mit homogenen Dirichlettrandbedingungen ist $V = V_0 = V_g$ der Sobolevraum $H_{0,D}^1(\Omega, \mathbb{R}^2)$ mit

$$H_{0,D}^1(\Omega, \mathbb{R}^2) = \{v \in H^1(\Omega, \mathbb{R}^2) : v = 0 \text{ auf } \Gamma_D\}$$

3.1.2 Herleitung der Variationsformulierung für den nichtlinearen Fall

Die Herleitung der Variationsformulierung erfolgt im wesentlichen wie im linearen Fall. Allerdings ergeben sich andere Anforderungen an die Räume, da u nicht mehr linear vorkommt.

Es ergibt sich folgendende Variationsformulierung

Suche ein $u \in V_g = \{v \in V : v = g_D \text{ auf } \Gamma_D\}$, sodass

$$a(u, v) = \langle f, v \rangle \tag{3.1}$$

für alle $v \in V_0 = \{v \in V : v = 0 \text{ auf } \Gamma_D\}$

mit der Biform

$$a(u, v) = \int_{\Omega} (\nabla u + I) \overline{\overline{D}} E(u) : \nabla v dx$$

und

$$\langle f, v \rangle = \int_{\Omega} f \cdot v dx + \int_{\Gamma_N} g_N \cdot v ds$$

Die Wahl der Räume V , V_0 und V_g ist kompliziert, da man mindestens kubische Integrierbarkeit von ∇u voraussetzen muss. Eine entsprechende Darstellung findet sich in [8]. Da der Fokus dieser Arbeit auf dem diskretisierten System liegt, wird nicht genauer darauf eingegangen.

3.2 Existenz und Eindeutigkeit für den linearen Fall

In diesem und in den folgenden Abschnitten konzentrieren wir uns der Einfachheit halber auf den Fall homogener Dirichletrandbedingungen $g_D = 0$. In diesem Kapitel folgen wir zu einem großen Teil dem Vorlesungsskript [2].

Satz 3.1 (Satz von Lax Milgram). [2] Sei V ein reeller Hilbertraum und

1. $f \in V^*$

2. $a : V \times V \rightarrow \mathbb{R}$ ist eine Bilinearform, die

(a) auf V beschränkt ist, d.h. es gibt eine Konstante $\mu_2 > 0$, sodass

$$|a(u, v)| \leq \mu_2 \|u\|_V \|v\|_V \quad \forall u, v \in V,$$

(b) auf V elliptisch ist, d.h. es gibt eine Konstante $\mu_1 > 0$, sodass

$$|a(v, v)| \geq \mu_1 \|v\|_V^2 \quad \forall v \in V.$$

Dann hat das Variationsproblem:

Finde ein $u \in V$, sodass

$$a(u, v) = \langle f, v \rangle \quad \forall v \in V$$

eine eindeutige Lösung und es gilt die folgende Ungleichung:

$$\frac{1}{\mu_2} \|f\|_{V^*} \leq \|u\|_V \leq \frac{1}{\mu_1} \|f\|_{V^*}$$

Beweis. siehe [3]. □

Im Folgenden ist:

$$\|v\|_{1,\Omega} := \|v\|_{H^1(\Omega)} = \left(\|v\|_{L^2(\Omega)}^2 + \|\nabla v\|_{L^2(\Omega)}^2 \right)^{\frac{1}{2}}$$

und

$$\|v\|_{0,D} := \|v\|_{L^2(D)} \quad \text{für } D = \Omega \text{ oder } \Gamma_N$$

Nun prüfen wir die Voraussetzungen für unser Variationsproblem.

Voraussetzung 1.:

Die Linearität von f folgt sofort aus der Definition von f . Mit der Cauchy Ungleichung folgt:

$$|\langle f, v \rangle| \leq \|f\|_{0,\Omega} \|v\|_{0,\Omega} + \|g_N\|_{0,\Gamma_N} \|v\|_{0,\Gamma_N}$$

Aus $\|v\|_{0,\Omega} \leq \|v\|_{1,\Omega}$ und $\|v\|_{0,\Gamma_N} \leq c(\Gamma_N) \|v\|_{1,\Omega}$ folgt (aufgrund des Spursatzes) somit die Beschränktheit von f .

2. Man sieht, dass a eine Bilinearform ist, da sowohl u und v nur linear vorkommen.

2 (a) a ist beschränkt: Mit der Formel (2.2) folgt:

$$|\overline{D}\epsilon(u) : \epsilon(v)| = |D\vec{\epsilon}(u) \cdot \vec{\epsilon}(v)|$$

und somit

$$|D\vec{\epsilon}(u) \cdot \vec{\epsilon}(v)| \leq \lambda_{max}(D) \|\vec{\epsilon}(u)\|_{l^2} \|\vec{\epsilon}(v)\|_{l^2} \leq \lambda_{max}(D) \|\epsilon(u)\|_F \|\epsilon(v)\|_F$$

Dabei ist $\lambda_{max}(D)$ der maximale Eigenwert der Matrix D und $\|A\|_F = (A : A)^{\frac{1}{2}}$ die Frobeniusnorm. Es gilt also:

$$\begin{aligned} |a(u, v)| &\leq \lambda_{max}(D) \int_{\Omega} \|\epsilon(u)\|_F \|\epsilon(v)\|_F dx \\ &\leq \lambda_{max}(D) \left(\int_{\Omega} \|\epsilon(u)\|_F^2 dx \right)^{\frac{1}{2}} \left(\int_{\Omega} \|\epsilon(v)\|_F^2 dx \right)^{\frac{1}{2}} \end{aligned}$$

Mit der Ungleichung

$$\left[\frac{1}{2}(a + b) \right]^2 \leq \frac{1}{2}(a^2 + b^2)$$

folgt

$$\begin{aligned} \int_{\Omega} \|\epsilon(v)\|_F^2 dx &= \sum_{i,j=1}^2 \int_{\Omega} \epsilon_{ij}(v)^2 dx = \sum_{i,j=1}^2 \int_{\Omega} \left[\frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right)^2 \right] dx \\ &\leq \sum_{i,j=1}^2 \int_{\Omega} \left[\frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} \right)^2 + \left(\frac{\partial v_j}{\partial x_i} \right)^2 \right] dx = \frac{1}{2} |v|_{1,\Omega}^2 \leq \|v\|_{1,\Omega}^2 \end{aligned}$$

Damit folgt nun die Ungleichungskette

$$|a(u, v)| \leq \lambda_{max}(D) |u|_{1,\Omega} |v|_{1,\Omega} \leq \lambda_{max}(D) \|u\|_{1,\Omega} \|v\|_{1,\Omega}$$

und somit die Voraussetzung.

2 (b) a ist elliptisch :

Wir wissen

$$D\vec{\epsilon}(v) : \vec{\epsilon}(v) \geq \lambda_{min}(D) \vec{\epsilon}(v) : \vec{\epsilon}(v)$$

und damit folgt

$$a(v, v) = \int_{\Omega} D\vec{\epsilon}(v) : \vec{\epsilon}(v) dx \geq \lambda_{min}(D) \int_{\Omega} \vec{\epsilon}(v) : \vec{\epsilon}(v) dx$$

Um den Term $\int_{\Omega} \epsilon(v) : \epsilon(v) dx$ abschätzen zu können, werden die beiden Korn Ungleichungen benötigt.

Lemma 3.1 (2. Korn Ungleichung). *Sei $\Omega \subset \mathbb{R}^2$ offen und beschränkt durch einen lip-schitzstetigen Rand. Dann gibt es eine Konstante $c_K = c_K(\Omega) > 0$, sodass*

$$\int_{\Omega} \epsilon(v) : \epsilon(v) dx + \|v\|_{0,\Omega}^2 \geq c_K^2 \|v\|_{1,\Omega}^2 \quad \forall v \in H^1(\Omega, \mathbb{R}^2)$$

Beweis. siehe [2] □

Lemma 3.2. *Sei $\Omega \subset \mathbb{R}^2$ offen, zusammenhängend und beschränkt durch einen lipschitzstetigen Rand. Dann gilt*

1. *Wenn $\Gamma_D \subset \Gamma$ und $\Gamma_D \neq \emptyset$, dann gibt es eine Konstante $c_K = c_K(\Omega) > 0$, sodass*

$$\int_{\Omega} \epsilon(v) : \epsilon(v) dx \geq c_K^2 |v|_{1,\Omega}^2 \quad \forall v \in V = H_{0,D}^1(\Omega, \mathbb{R}^2).$$

2. *Wenn $\Gamma_D = \emptyset$, dann gibt es eine Konstante $c_K = c_K(\Omega) > 0$, sodass*

$$\int_{\Omega} \epsilon(v) : \epsilon(v) dx \geq c_K^2 |v|_{1,\Omega}^2 \quad \forall v \in \widehat{H} = \{v \in H^1(\Omega, \mathbb{R}^2) : \int_{\Omega} v dx = 0, \int_{\Omega} \text{curl}(v) dx = 0\}.$$

Beweis. siehe [2] □

Mit diesem Lemma ergibt sich für $\mu_1 = \lambda_{\min}(D)c_K^2$ im Fall von $\Gamma_N \neq \emptyset$.

Somit sind die Voraussetzungen für den Satz von Lax Milgram geprüft und das Variationsproblem hat eine eindeutige Lösung.

3.3 Lösungstheorie für das nichtlineare Problem

Unter bestimmten Glattheitsvoraussetzungen an das Gebiet und für hinreichend kleine Kräfte kann die Existenz einer Lösung des nichtlinearen Problems nachgewiesen werden. Allerdings gilt nicht immer Eindeutigkeit, siehe [8].

Im Folgenden wird die Existenz einer eindeutigen Lösung vorausgesetzt.

3.4 Ein zugehöriges Energieminimierungsproblem

Satz 3.2. *Sei $V_0 = V_g \subset H^1(\Omega, \mathbb{R}^2)$ ein Banachraum, sodass der untenstehende Ausdruck wohldefiniert ist. Dann ist jede Lösung des Minimierungsproblems*

$$\min_{u \in V_g} \frac{1}{2} \int_{\Omega} \overline{\overline{DE}}(u) : E(u) dx - \langle f, u \rangle$$

auch Lösung des Variationsproblems (3.1), d.h.

Finde ein $u \in V_g$:

$$\int_{\Omega} (\nabla u + I) \overline{\overline{DE}}(u) : \nabla v dx = \langle f, v \rangle$$

$\forall v \in V_0$

Das Funktional in Satz 3.2 ist physikalisch gesehen die innere Energie. Wir setzen im Folgenden die Existenz einer eindeutigen Lösung des Minimierungsproblems voraus.

Zum Beweis benötigen wir das folgende Lemma:

Lemma 3.3. Die Richtungsableitung von $E(u)$ in Richtung v ist wie folgt:

$$\nabla_v (E(u)) = \frac{1}{2} F(u) \nabla^T v + \frac{1}{2} F(u)^T \nabla v$$

Beweis.

$$\begin{aligned} \nabla_v (E(u)) &= \lim_{t \rightarrow 0} \frac{E(u + tv) - E(u)}{t} \\ &= \frac{1}{2} \lim_{t \rightarrow 0} \left(\frac{\nabla u + t \nabla v + \nabla^T u + t \nabla^T v + (\nabla^T u + t \nabla^T v) (\nabla u + t \nabla v)}{t} \right. \\ &\quad \left. - \frac{(\nabla u + \nabla^T u + \nabla^T u \nabla u)}{t} \right) \\ &= \frac{1}{2} \lim_{t \rightarrow 0} \left(\frac{\nabla u + \nabla^T u + \nabla^T u \nabla u + t (\nabla v + \nabla^T v + \nabla^T u \nabla v \nabla v \nabla^T u)}{t} \right. \\ &\quad \left. + \frac{t^2 \nabla^T v \nabla v - (\nabla u + \nabla^T u + \nabla^T u \nabla u)}{t} \right) \\ &= \frac{1}{2} (\nabla v + \nabla^T v + \nabla^T u \nabla v + \nabla v \nabla^T u) \\ &= \frac{1}{2} (I + \nabla u) \nabla^T v + \frac{1}{2} (I + \nabla^T u) \nabla v \\ &= \frac{1}{2} F(u) \nabla^T v + \frac{1}{2} (F(u))^T \nabla v \end{aligned}$$

□

Beweis von Satz 3.2. Das Ziel ist es zu zeigen, dass das Minimierungsproblem

$$\min_{u \in V_0} \frac{1}{2} \int_{\Omega} \overline{\overline{D}} E(u) : E(u) dx - \langle f, u \rangle$$

auf unser Variationsproblem

$$\int_{\Omega} P(u) : \nabla v dx = \langle f, v \rangle \quad \forall v \in V_0$$

führt. Dazu bilden wir die Richtungsableitung des Minimierungsfunktionals bei u in Richtung v und setzen diese gleich Null.

$$\nabla_v \frac{1}{2} \int_{\Omega} \overline{\overline{D}} E(u) : E(u) dx - \langle f, u \rangle = 0$$

Aufgrund der Produktregel gilt:

$$\begin{aligned} & \nabla_v \frac{1}{2} \int_{\Omega} \overline{\overline{D}}E(u) : E(u) dx - \langle f, u \rangle \\ &= \frac{1}{2} \int_{\Omega} 2\overline{\overline{D}}E(u) : \nabla_v E(u) - \langle f, v \rangle \end{aligned}$$

Mit Lemma 3.3 ergibt sich die Bedingung

$$\int_{\Omega} \overline{\overline{D}}E(u) : \left(\frac{1}{2} F(u) \nabla^T v + \frac{1}{2} F(u)^T \nabla v \right) dx - \langle f, v \rangle = 0$$

welche zum Minimierungsproblem äquivalent ist. (da das Minimierungsfunktional konvex ist)

Da der $:$ Operator linear ist, bekommt man

$$\int_{\Omega} \overline{\overline{D}}E(u) : \frac{1}{2} F(u) \nabla^T v + \overline{\overline{D}}E(u) : \frac{1}{2} F(u)^T \nabla v dx - \langle f, v \rangle = 0$$

Aus $A : F^T B = F A : B$ folgt

$$\int_{\Omega} \overline{\overline{D}}E(u) : \frac{1}{2} F(u) \nabla^T v + \frac{1}{2} F(u) \overline{\overline{D}}E(u) : \nabla v dx - \langle f, v \rangle = 0$$

Da $\overline{\overline{D}}E(u)$ symmetrisch ist ergibt sich

$$\begin{aligned} & \int_{\Omega} \frac{1}{2} F(u) \overline{\overline{D}}E(u) : \nabla v + \frac{1}{2} F(u) \overline{\overline{D}}E(u) : \nabla v dx - \langle f, v \rangle \\ &= \int_{\Omega} F(u) \overline{\overline{D}}E(u) : \nabla v dx - \langle f, v \rangle \\ &= \int_{\Omega} F(u) S(u) : \nabla v dx - \langle f, v \rangle \\ &= \int_{\Omega} P(u) : \nabla v dx - \langle f, v \rangle = 0 \end{aligned}$$

□

3.5 Finite Elemente (linearer Fall)

Wir verwenden zum numerischen Lösen des Variationsproblems die Methode der Finiten Elemente (siehe z.B. citebraess). In unserem Fall wird das Gebiet in Dreiecke zerlegt (Triangulierung), auf denen die Basisfunktionen definiert sind. Die Dreiecke der Zerlegung \mathcal{T}_h werden mit T bezeichnet.

$$\overline{\Omega} = \bigcup_{T \in \mathcal{T}_h} T$$

Um das Problem zu diskretisieren, verwenden wir das Galerkin Prinzip. Unser Ausgangsproblem ist das lineare Problem:

Finde ein $u \in V_0$, sodass:

$$a(u, v) = \langle f, v \rangle \quad \forall v \in V$$

Sei $V_h \subset V$ ein endlich dimensionaler Unterraum und $V_{0h} := V_h \cap V_0$. Dann ist das zugehörige diskrete Problem:

Finde ein $u_h \in V_{0h}$, sodass:

$$a(u_h, v_h) = \langle f, v_h \rangle \quad \forall v_h \in V_{0h} \quad (3.2)$$

Der Raum V_h wird nun auf Basis einer Triangulierung \mathcal{T}_h festgelegt.

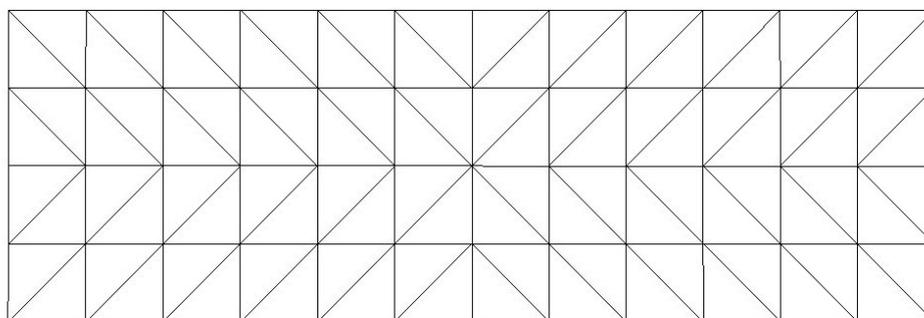
$$V_h = \{v \in C(\bar{\Omega}, \mathbb{R}^2) : \forall T \in \mathcal{T}_h : v \in (P_1)^2\} \quad (3.3)$$

Wir nehmen an, dass Γ_D von der Triangulierung aufgelöst wird, dann ist V_{0h} der Raum jener Funktionen aus V_h die auf allen Knoten $x_j \in \bar{\Gamma}_D$ verschwinden.

Als Raum der Formfunktionen wählen wir auf jedem Element T den Raum P_1 , der Polynome vom Grad ≤ 1 , also:

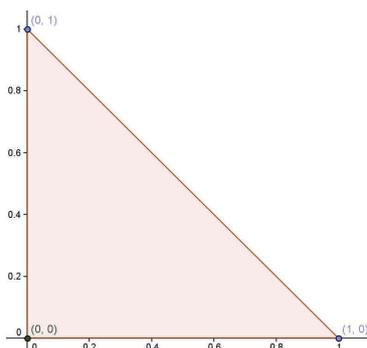
$$P_1 = \text{span}\{1, \xi_1, \xi_2\}$$

Die Funktionen in V_h sind vektorwertige, stückweise lineare, stetige Funktionen, die durch die Basisfunktionen festgelegt werden sollen.



Die obige Grafik zeigt eine Beispiel-Triangulierung des Gebietes $\Omega = (0, 3) \times (0, 1)$.

Wir definieren nun die Formfunktionen auf dem Referenzdreieck mit den Eckpunkten $(0, 0)$, $(1, 0)$, $(0, 1)$.



ξ_1 und ξ_2 entsprechen dabei den x bzw y Koordinaten im Referenzgebiet.

Definition 3.1 (skalare Formfunktionen).

$$\begin{aligned} p^{(0)} &= 1 - \xi_1 - \xi_2 \\ p^{(1)} &= \xi_1 \\ p^{(2)} &= \xi_2 \end{aligned}$$

Man sieht, dass die Formfunktionen nur auf einem Eckpunkt den Wert 1 haben und auf den anderen beiden Eckpunkten 0 sind.

In der Ebene benötigen wir die Formfunktionen für 2 Richtungen (x , y Richtung):

Definition 3.2 (Vektorwertige Formfunktionen).

$$\left(\begin{pmatrix} p^{(0)} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ p^{(0)} \end{pmatrix}, \begin{pmatrix} p^{(1)} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ p^{(1)} \end{pmatrix}, \begin{pmatrix} p^{(2)} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ p^{(2)} \end{pmatrix} \right)$$

Die Betrachtungsweise via Formfunktionen ist wichtig für die FE-Implementierung (über die Transformation aufs Referenzelement). Man kann den Raum V_h aber auch direkt über eine Knotenbasis $\{\varphi_i\}_{i=1}^{2n_h}$ darstellen. Der Index i durchläuft dabei jeden Knoten der Triangulierung. Im Knoten x_j gilt für $i = 1 \dots n_h$:

$$\begin{aligned} \varphi_{2i-1}(x_j) &= \begin{pmatrix} \delta_{ij} \\ 0 \end{pmatrix} \\ \varphi_{2i}(x_j) &= \begin{pmatrix} 0 \\ \delta_{ij} \end{pmatrix} \end{aligned}$$

Die Lösung u_h kann bezüglich der Basis entwickelt werden:

$$u_h(x) = \sum_{i=1}^{2n_h} u_i \varphi_i(x)$$

Damit wird nach der Berücksichtigung der Dirichletrandbedingungen aus dem diskreten Problem (3.2) das lineare Gleichungssystem:

$$K_h \underline{u}_h = \underline{f}_h \tag{3.4}$$

Der Vektor \underline{u}_h gibt die Werte der Funktion u_h in den Knoten der Triangulierung an (ausgenommen jene die auf $\overline{\Gamma_D}$ liegen).

K_h ist die sogenannte Steifigkeitsmatrix und wie folgt definiert:

$$[K_h]_{ij} = a(\varphi_j, \varphi_i)$$

für alle Indices $i, j \in \{1, \dots, n_h\}$, die zu keinen Knoten mit Dirichletrandbedingungen gehören.

Da die Basisfunktionen nur lokal (auf den Dreiecken mit dem zur Basisfunktion gehörenden Knoten) ungleich null sind, sind nur wenige Einträge der Steifigkeitsmatrix ungleich null. Die Matrix ist also nur dünn besetzt.

\underline{f}_h ist der sogenannte Lastvektor:

$$[\underline{f}_h]_i = \langle f, \varphi_i \rangle$$

Mit den Konstanten μ_1 und μ_2 aus dem Satz von Lax Migram lässt sich die Konditionszahl der Matrix für den Fall einer quasi-uniformen Triangulierung wie folgt abschätzen.

$$\kappa(K_h) = \frac{\mu_2}{\mu_1} \mathcal{O}(h^{-2})$$

h ist dabei der maximale Durchmesser der Dreiecke der Triangulierung. Das heißt je feiner die Triangulierung, umso schlechter ist die Konditionszahl der Matrix K_h .

3.6 Lösungsmethoden für das lineare Problem

Um das lineare Gleichungssystem (3.4) zu lösen kann man direkte oder iterative Löser verwenden. Die Matrix K_h ist im Allgemeinen nur dünn besetzt. Beim Faktorisieren ergeben sich dann aber Matrizen, die im Allgemeinen voll besetzt sind, und damit ergeben sich je nach Größe der Matrizen und des Speichers Probleme beim Abspeichern der Matrizen. Iterative Verfahren benötigen hingegen nur die Anwendung der Matrix. Es kann z.B. das CG Verfahren für SPD Matrizen (in diesem Fall erfüllt) eingesetzt werden. Genauere Information zum Lösen von Gleichungssystemen findet man im Buch [1, Kapitel 4].

3.7 Lösungsmethoden für das nichtlineare Modell

Im Unterschied zum linearisierten Fall ist hier der Zusammenhang zwischen S und u nicht mehr linear. Durch das Diskretisieren erhält man dann auch kein lineares Gleichungssystem mehr. Also benötigen wir alternative Überlegungen. Im ersten Teil werden das Fixpunktverfahren und das Newtonverfahren vorgestellt.

Die FEM Diskretisierung des Randwertproblems

$$\begin{aligned} -div(FS) &= f && \text{in } \Omega \\ S &= \overline{DE} && \text{in } \Omega \\ E &= \frac{1}{2}(F^T F - I) && \text{in } \Omega \\ F &= \nabla u + I \\ u &= 0 && \text{auf } \Gamma_D \\ (FS)n &= g_N && \text{auf } \Gamma_N \end{aligned}$$

bzw. des Variationsproblems (3.1)

$$a(u, v) = \langle f, v \rangle \tag{3.5}$$

für alle $v \in V_0 = \{v \in V : v = 0 \text{ auf } \Gamma_D\}$
führen auf die Gleichung:

$$\underline{k}_h(\underline{u}_h) = \underline{f}_h \tag{3.6}$$

Hierbei ist die nichtlineare Abbildung

$$\underline{k}_h : \mathbb{R}^{2n_h} \rightarrow \mathbb{R}^{2n_h}$$

wie folgt definiert:

$$[\underline{k}_h(\underline{u}_h)]_i = a(u_h, \varphi_i) \quad \forall i \in \{1, \dots, 2n_h\}$$

Durch den Ritz Isomorphismus [3, Seite 21] besteht der Zusammenhang:

$$\begin{aligned} \underline{u}_h &\leftrightarrow u_h, \quad u_h(x) = \sum_{i=1}^{2n_h} [\underline{u}_h]_i \varphi_i \\ \underline{\omega}_h &\leftrightarrow \omega_h, \quad \omega_h(x) = \sum_{i=1}^{2n_h} [\underline{\omega}_h]_i \varphi_i \end{aligned}$$

\underline{f}_h ist definiert als:

$$[\underline{f}_h]_i = \langle f, \varphi_i \rangle$$

3.7.1 Fixpunktverfahren mit Liniensuche

Beim Fixpunktverfahren wird die Gleichung

$$\underline{k}_h(\underline{u}_h) = \underline{f}_h$$

auf Nullpunktform und anschließend auf Fixpunktform gebracht.

$$\underline{f}_h - \underline{k}_h(\underline{u}_h) = 0$$

Damit ergibt sich folgende Iterationsvorschrift

$$\underline{u}_h^{(k+1)} = \underline{u}_h^{(k)} + \left(\underline{f}_h - \underline{k}_h(\underline{u}_h^{(k)}) \right)$$

Um Konvergenz zu erhalten, wird das Residuum mit dem Dämpfungsparameter τ_k multipliziert. Um eine höhere Konvergenzgeschwindigkeit zu erreichen, wird das Residuum noch mit einem Vorkonditionierer multipliziert. Als Vorkonditionierer wird die Steifigkeitsmatrix K_h des linearen Verfahrens verwendet.

$$\underline{u}_h^{(k+1)} = \underline{u}_h^{(k)} + \tau_k K_h^{-1} \left(\underline{f}_h - \underline{k}_h(\underline{u}_h^{(k)}) \right)$$

Der Parameter τ_k wird, von 1 ausgehend, so lange halbiert, bis die Norm des Residums im $(k+1)$ ten Schritt kleiner als im k ten Schritt ist. Der Vektor $\underline{k}_h(\underline{u}_h^{(k)})$ muss natürlich in jedem Schritt berechnet werden, was analog zum linearen Fall durch Assemblierung geschieht.

Es ergibt sich folgender Algorithmus:

```

 $u_{alt} = u_{Start}$ 
 $r_{alt} = f - \underline{k}_h(u_k)$ 
while  $\|r_{alt}\| > eps$  do

```

```

Löse  $K_h \omega_h = r_{alt}$ 
 $\tau = 1$ 
 $r_{neu} = K_h^{-1} \left( \underline{f}_h - \underline{k}_h(u_{neu}) \right)$ 
while  $\|r_{neu}\| > \|r_{alt}\|$  do
   $u_{neu} = u_{alt} + \tau \omega_h$ 
   $r_{neu} = \underline{f}_h - \underline{k}_h(u_{neu})$ 
   $\tau = \frac{\tau}{2}$ 
end while
 $r_{alt} = r_{neu}$ 
 $u_{alt} = u_{neu}$ 
end while

```

Das Fixpunktverfahren ist eine Methode, die in diesem Fall sehr langsam konvergiert und daher keine wirkliche Option ist, um ein derartiges Problem effizient zu lösen.

3.7.2 Newtonverfahren

Eine weitere Methode, ein nichtlineares System zu lösen, ist das Newton Verfahren. Die Konvergenz ist oft wesentlich schneller als beim Fixpunktverfahren mit Liniensuche. Es hat aber auch den Nachteil, dass es nur lokal konvergiert.

Der Ausgangspunkt für das Newtonverfahren ist wie oben die Gleichung

$$\underline{k}_h(\underline{u}_h) = \underline{f}_h$$

bzw. ihre Nullpunktform

$$\underline{f}_h - \underline{k}_h(\underline{u}_h) = 0.$$

Mit dem Newtonverfahren erhält man daraus folgende Iterationsvorschrift:

$$\underline{u}_h^{(k+1)} = \underline{u}_h^{(k)} + K'_h \left(\underline{u}_h^{(k)} \right)^{-1} \left(\underline{f}_h - \underline{k}_h \left(\underline{u}_h^{(k)} \right) \right)$$

wobei $K'_h \left(\underline{u}_h^{(k)} \right)$ die Fréchet-Ableitung von \underline{k}_h an der Stelle $\underline{u}_h^{(k)}$ bezeichnet. Mit dem Ansatz

$$\underline{u}_h^{(k+1)} = \underline{u}_h^{(k)} + \underline{\omega}_h^{(k)}$$

ergibt sich dann das lineare Gleichungssystem

$$K'_h \left(\underline{u}_h^{(k)} \right) \underline{\omega}_h^{(k)} = \left(\underline{f}_h - \underline{k}_h \left(\underline{u}_h^{(k)} \right) \right)$$

für die Korrektur $\underline{\omega}_h^{(k)}$. Um die Tangentensteifigkeitsmatrix $K'_h \left(\underline{u}_h^{(k)} \right)$ zu berechnen, ist es am besten, gleich die Richtungsableitung von \underline{k}_h in Richtung $\underline{\omega}_h^{(k)}$ an der Stelle $\underline{u}_h^{(k)}$ zu berechnen.

$$\left[K'_h \left(\underline{u}_h^{(k)} \right) \underline{\omega}_h^{(k)} \right]_i = \lim_{t \rightarrow 0} \frac{1}{t} \left(a \left(\underline{u}_h^{(k)} + t \underline{\omega}_h^{(k)}, \varphi_i \right) - a \left(\underline{u}_h^{(k)}, \varphi_i \right) \right)$$

$a(u, v)$ ist wie in der Formel (3.1) definiert:

$$a(u, v) = \int_{\Omega} P(u) : \nabla v dx$$

wobei

$$P(u) = (\nabla u + I) \overline{\overline{D}} \left[\frac{1}{2} (\nabla u + \nabla u^T + \nabla u^T \nabla u) \right].$$

Um diesen Grenzwert zu berechnen, benötigen wir folgendes Hilfsresultat:

$$\begin{aligned} & \lim_{t \rightarrow 0} \frac{1}{t} (P(u + t\omega) - P(u)) \\ &= \lim_{t \rightarrow 0} \frac{1}{t} \left[(\nabla(u + t\omega) + I) \overline{\overline{D}} \frac{1}{2} (\nabla(u + t\omega) + \nabla^T(u + t\omega) + \nabla^T(u + t\omega) \nabla(u + t\omega)) \right. \\ & \quad \left. - (\nabla u + I) \overline{\overline{D}} \frac{1}{2} (\nabla u + \nabla^T u + \nabla^T u \nabla u) \right] \\ &= \frac{1}{2} \lim_{t \rightarrow 0} \frac{1}{t} \left[(\nabla u + t \nabla \omega + I) \overline{\overline{D}} (\nabla u + t \nabla \omega + \nabla^T u + t \nabla^T \omega + \nabla^T u \nabla u \right. \\ & \quad \left. + t \nabla^T \omega \nabla u + t \nabla^T u \nabla \omega + t^2 \nabla^T \omega \nabla \omega) - (\nabla u + I) \overline{\overline{D}} (\nabla u + \nabla^T u + \nabla^T u \nabla u) \right] \\ &= \frac{1}{2} \lim_{t \rightarrow 0} \frac{1}{t} \left[(\nabla u + t \nabla \omega + I) \overline{\overline{D}} (\nabla u + \nabla^T u + \nabla^T u \nabla u + t (\nabla \omega + \nabla^T \omega + \right. \\ & \quad \left. \nabla^T \omega \nabla u + \nabla^T u \nabla \omega) + t^2 \nabla^T \omega \nabla \omega) - (\nabla u + I) \overline{\overline{D}} (\nabla u + \nabla^T u + \nabla^T u \nabla u) \right] \\ &= \frac{1}{2} \lim_{t \rightarrow 0} \frac{1}{t} \left[t \left(\nabla \omega \overline{\overline{D}} (\nabla u + \nabla^T u + \nabla^T u \nabla u) \right) + t (\nabla u + I) \overline{\overline{D}} (\nabla \omega + \nabla^T \omega \right. \\ & \quad \left. + \nabla^T \omega \nabla u + \nabla^T u \nabla \omega) + t^2 \nabla \omega \overline{\overline{D}} (\nabla \omega + \nabla^T \omega + \nabla^T \omega \nabla u \right. \\ & \quad \left. + \nabla^T u \nabla \omega) + t^2 (\nabla u + I) \overline{\overline{D}} (\nabla^T \omega \nabla \omega) + t^3 \nabla \omega \overline{\overline{D}} \nabla^T \omega \nabla \omega \right] \\ &= \frac{1}{2} \lim_{t \rightarrow 0} \frac{1}{t} \left[t \left(\nabla \omega \overline{\overline{D}} (\nabla u + \nabla^T u + \nabla^T u \nabla u) + (\nabla u + I) \overline{\overline{D}} (\nabla \omega + \nabla^T \omega \right. \right. \\ & \quad \left. \left. + \nabla^T \omega \nabla u + \nabla^T u \nabla \omega) \right) + t^2 (\nabla \omega \overline{\overline{D}} (\nabla \omega + \nabla^T \omega + \nabla^T \omega \nabla u + \nabla^T u \nabla \omega) \right. \\ & \quad \left. + (\nabla u + I) \overline{\overline{D}} (\nabla^T \omega \nabla \omega)) + t^3 \nabla \omega \overline{\overline{D}} \nabla^T \omega \nabla \omega \right] \\ &= \frac{1}{2} \left(\nabla \omega \overline{\overline{D}} (\nabla u + \nabla^T u + \nabla^T u \nabla u) \right. \\ & \quad \left. + (\nabla u + I) \overline{\overline{D}} (\nabla \omega + \nabla^T \omega + \nabla^T \omega \nabla u + \nabla^T u \nabla \omega) \right) \end{aligned}$$

Dadurch ergibt sich:

$$\begin{aligned} \left[K'_h \left(\underline{u}_h^{(k)} \right) \underline{\omega}_h^{(k)} \right]_i &= \int_{\Omega} \frac{1}{2} \left[\nabla \omega_h^{(k)} \overline{\overline{D}} \left(\nabla u_h^{(k)} + \nabla^T u_h^{(k)} + \nabla^T u_h^{(k)} \nabla u_h^{(k)} \right) \right. \\ & \quad \left. + (\nabla u_h^{(k)} + I) \overline{\overline{D}} (\nabla \omega_h^{(k)} + \nabla^T \omega_h^{(k)} + \nabla^T \omega_h^{(k)} \nabla u_h^{(k)} \right. \\ & \quad \left. + \nabla^T u_h^{(k)} \nabla \omega_h^{(k)}) \right] : \nabla \varphi_i dx \end{aligned}$$

Wir fassen nun das obige Newtonverfahren zusammen:

```
 $u_{alt} = u_{Start}$   
 $r_{alt} = \underline{f}_h - \underline{k}_h(u_{start})$   
while  $\|r_{alt}\| > eps$  do  
   $\tau = 1$   
  while  $\|r_{neu}\| > \|r_{alt}\|$  do  
    Berechne  $K'_h(u_{alt})$   
    Löse  $K'_h(u_{alt})\omega_h = r_{alt}$  mittels direktem oder iterativen Löser  
     $u_{neu} = u_{alt} + \tau\omega_h$   
     $\tau = \frac{\tau}{2}$   
     $r_{neu} = \underline{f}_h - \underline{k}_h(u_{neu})$   
  end while  
   $r_{alt} = r_{neu}$   
   $u_{alt} = u_{neu}$   
end while
```

4 Floating Frame of Reference Formulation

4.1 Gebietszerlegung

Wir treffen für dieses Kapitel folgende Annahmen an das Gebiet:

$$\Omega \subset \mathbb{R}^2$$

An den Rand des Gebietes stellen wir wie in der gesamten Arbeit folgende Bedingungen:

$$\partial\Omega = \Gamma_D \cup \Gamma_N, \Gamma_D \cap \Gamma_N = \emptyset, \Gamma_D \neq \emptyset$$

Wir zerlegen das Gebiet Ω in disjunkte Teilgebiete $\Omega_i, i = 1, \dots, N$ mit

$$\bar{\Omega} = \bigcup_{i=1}^N \bar{\Omega}_i, \Omega_i \cap \Omega_j = \emptyset \quad \forall i \neq j.$$

Speziell für diese Arbeit (insbesondere für die Implementierung) nehmen wir an, dass das Gebiet Ω und die Teilgebiete Ω_i rechteckig sind.

Der folgende Zugang kann aber relativ einfach auf polygonale Gebiete und Teilgebiete erweitert werden.

Ein Beispiel einer solchen Gebietszerlegung:

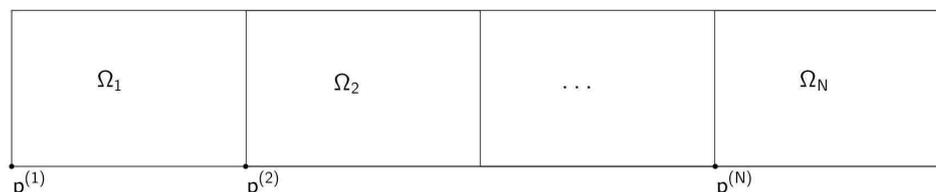


Abbildung 1: Ein Beispiel einer Gebietszerlegung

Die Punkte $p^{(i)}$ bezeichnen jeweils den linken unteren Punkt des Gebietes Ω_i , dies ist wichtig für den Abschnitt 4.2.

Weiters nehmen wir der Einfachheit halber an, dass $\Gamma_D \cap \partial\Omega_i$ und $\Gamma_N \cap \partial\Omega_i$ ein Vereinigung von Kanten des Rechtecks Ω_i ist. Auch diese Annahme stellt in der Herleitung der FFRF Formulierung keine wirkliche Einschränkung dar.

4.2 Parametrisierung der Verschiebung mittels „Frame“

4.2.1 Ansatz für die totale Verschiebung

Es ergibt sich auf jedem der Teilgebiete Ω_i eine Aufspaltung der Verschiebung $u(X)$, im Folgenden bezeichnet mit $u_{total}(X)$, in eine flexible Verschiebung $u_f^{(i)}(X)$, die relativ zum

Rahmen wirkt, und eine Verschiebung des Rahmens $u_r^{(i)}(X)$. Die Verschiebung des Rahmens besteht aus einer Drehung um einen Winkel $\varphi^{(i)}$ und einer Translation um den Vektor $u_t^{(i)}$.

Dazu führen wir folgende Notation ein:

X ist ein Punkt im nicht deformierten Gebiet im absoluten Koordinatensystem.

$\bar{x}^{(i)}$ ist der Punkt X relativ zu $p^{(i)}$, d.h.:

$$X = p^{(i)} + \bar{x}^{(i)}$$

Sei $r = \phi(X)$, d.h. der Punkt X nach der Deformation.

Zuerst wird der relative Punkt $\bar{x}^{(i)}$ um die flexible Verschiebung $u_f^{(i)}$ verschoben. Dieser Punkt $\bar{x}^{(i)} + u_f^{(i)}$ wird dann um den Winkel $\varphi^{(i)}$ gedreht. Somit erhalten wir $A^{(i)}(\bar{x}^{(i)} + u_f^{(i)})$, wobei $A^{(i)}$ die Rotationsmatrix um den Winkel φ_i bezeichnet. $A^{(i)}$ rotiert einen Vektor gegen den Uhrzeigersinn, also:

$$A^{(i)} = \begin{pmatrix} \cos(\varphi^{(i)}) & -\sin(\varphi^{(i)}) \\ \sin(\varphi^{(i)}) & \cos(\varphi^{(i)}) \end{pmatrix}$$

Es gilt:

$$A^{(i)(-1)} = A^{(i)T}$$

$A^{(i)}$ ist also eine orthogonale Matrix.

Dazu kommt noch der Anteil der Translation $u_t^{(i)}$ und der Referenzpunkt des Frames $p^{(i)}$. Somit ergibt sich für r :

$$r = p^{(i)} + u_t^{(i)} + A^{(i)}(\bar{x}^{(i)} + u_f^{(i)})$$

bzw. lässt sich r auch wie folgt darstellen:

$$r = X + u_{total}(X)$$

Somit ergibt sich folgende Gleichung:

$$\begin{aligned} X + u_{total} &= p^{(i)} + u_t^{(i)} + A^{(i)}(\bar{x}^{(i)} + u_f^{(i)}) \\ p^{(i)} + \bar{x}^{(i)} + u_{total} &= p^{(i)} + u_t^{(i)} + A^{(i)}(\bar{x}^{(i)} + u_f^{(i)}) \end{aligned}$$

Also ergibt sich für u_{total} :

$$u_{total} = u_t^{(i)} + (A^{(i)} - I)\bar{x}^{(i)} + A^{(i)}u_f^{(i)}$$

Man sieht, dass die Verschiebung auf dem Gebiet Ω_i von der Translation $u_t^{(i)}$ der Rotationsmatrix $A^{(i)}$ und somit dem Rotationswinkel $\varphi^{(i)}$ und der flexiblen Verschiebung $u_f^{(i)}$ abhängt.

Es ergibt sich folgende Funktion:

$$u_{total}(X) = u_t^{(i)} + (A^{(i)} - I)\bar{x}^{(i)}(X) + A^{(i)}u_f^{(i)}(X) \quad \forall X \in \Omega^{(i)} \quad (4.1)$$

Diese Funktion kann auch parametrisiert dargestellt werden.

$$u_{total}^{(i)}(u_t, \varphi, u_f) = u_t + (A - I) \bar{x}^{(i)} + Au_f \quad (4.2)$$

wobei

$$A = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix}$$

mit $u_t \in \mathbb{R}^2$, $\varphi \in [0, 2\pi]$, $u_f : \Omega^{(i)} \rightarrow \mathbb{R}^2$.

Die Parametrisierung in der obigen Form ist nicht eindeutig bestimmt. Durch eine freie Wahl von u_t und φ ist u_f bestimmt. Das Ziel ist es aber, dass die lokale Verschiebung u_f geeignet klein ist.

Ziel des folgenden Abschnitts ist es drei Bedingungen an $u_f^{(i)}$ zu finden, so dass $u_t^{(i)}$ und $\varphi^{(i)}$ eindeutig bestimmt sind.

4.2.2 Fixierung der lokalen Verschiebung am Rahmen

Grundlage für die oben angesprochene Fixierung ist das folgende Lemma.

Lemma 4.1. *Sei $\Omega_i \subset \Omega$ und sei $X_1^{(i)} \in \overline{\Omega}_i$, $X_2^{(i)} \in \overline{\Omega}_i$ und $X_1^{(i)} \neq X_2^{(i)}$. Sei $u_{total} \in C(\overline{\Omega}_i)$. Dann führen die Bedingungen*

$$u_f \left(X_1^{(i)} \right) = 0 \quad (2 \text{ Bedingungen}) \quad (4.3)$$

$$d^\perp \cdot u_f \left(X_2^{(i)} \right) = 0 \quad (1 \text{ Bedingung}) \quad (4.4)$$

mit $d = X_2^{(i)} - X_1^{(i)}$, d^\perp ist ein Normalvektor auf d , immer auf eindeutig bestimmte Größen $u_t^{(i)}$, $\varphi^{(i)}$ und $u_f^{(i)}$, sodass (4.1) gilt.

Beweis. Zur Wiederholung Formel (4.1):

$$u_{total}(X) = u_t^{(i)} + (A^{(i)} - I) \bar{x}^{(i)}(X) + A^{(i)} u_f^{(i)}(X) \quad (4.5)$$

Für $X = X_1^{(i)}$ ergibt sich daraus aufgrund von (4.3):

$$\begin{aligned} u_{total} \left(X_1^{(i)} \right) &= u_t^{(i)} + (A^{(i)} - I) \bar{x}^{(i)} \left(X_1^{(i)} \right) + A^{(i)} u_f^{(i)} \left(X_1^{(i)} \right) \\ &= u_t^{(i)} + (A^{(i)} - I) \bar{x}^{(i)} \left(X_1^{(i)} \right) + 0 \end{aligned}$$

und somit für $u_t^{(i)}$:

$$u_t^{(i)} = u_{total} \left(X_1^{(i)} \right) - (A^{(i)} - I) \bar{x}^{(i)} \left(X_1^{(i)} \right)$$

Einsetzen von $X = X_2^{(i)}$ und von $u_t^{(i)}$ in (4.5):

$$\begin{aligned}
 e := u_{total}^{(i)}(X_2^{(i)}) - u_{total}^{(i)}(X_1^{(i)}) &= (A^{(i)} - I) \left(\underbrace{\bar{x}^{(i)}(X_2^{(i)}) - \bar{x}^{(i)}(X_1^{(i)})}_{=d} \right) \\
 &\quad + A^{(i)} u_f^{(i)}(X_2^{(i)}) \\
 A^{(i)T} e &= (I - A^{(i)T}) d + u_f^{(i)}(X_2^{(i)}) \\
 d^\perp \cdot A^{(i)T} e &= d^\perp \cdot d - d^\perp \cdot A^{(i)T} d + d^\perp \cdot u_f^{(i)}(X_2^{(i)})
 \end{aligned}$$

Mit der dritten Bedingung (4.4) folgt:

$$\begin{aligned}
 d^\perp \cdot A^{(i)T} e &= 0 - d^\perp \cdot A^{(i)T} d + 0 \\
 d^\perp \cdot A^{(i)T} (e + d) &= 0
 \end{aligned}$$

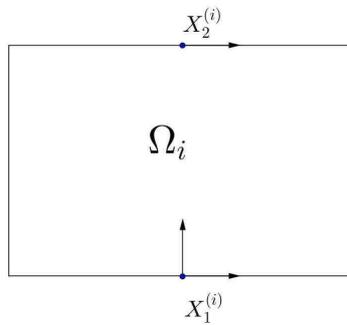
Somit gilt für den Winkel:

$$\varphi^{(i)} = \arg(e + d) - \arg(d^\perp)$$

$u_f^{(i)}$ ist schließlich aus (4.5) ableitbar und erfüllt (4.1) per Konstruktion. □

Somit ist die Parametrisierung in der Formel (4.1) mit den definierten Sperrungen (4.3) und (4.4) bijektiv. Wichtig dabei ist, dass die dritte Bedingung im zweiten Punkt im Winkel von 90 Grad zur Verbindungsgeraden der beiden Punkte $X_1^{(i)}$, $X_2^{(i)}$ wirkt. Diverse ähnliche Varianten findet man in [5]

Aufgrund unserer Annahme, dass Ω_i rechteckig ist, können wir z.B. die Verschiebung in beide Richtungen im unteren mittleren Punkt sperren (d.h. der Wert der lokalen Verschiebung muss 0 sein). Weiters wird die lokale Verschiebung in x -Richtung im oberen mittleren Punkt ebenfalls gesperrt (siehe nachfolgende Skizze). Diese Sperrung wird in jedem der Teilgebiet Ω_i gefordert.



Auf die genaue Einbindung dieser Sperrungen wir später eingegangen.

4.2.3 Betrachten des Verzerrungstensors

Nun betrachten wir den Verzerrungstensor angewandt auf die totale Verschiebung $u_{total}^{(i)}$. Dabei ergibt sich folgende Vereinfachung:

$$\begin{aligned}
E(u_{total}^{(i)}) &= \frac{1}{2} \left(\nabla^T u_{total}^{(i)} + \nabla u_{total}^{(i)} + \nabla^T u_{total}^{(i)} \nabla u_{total}^{(i)} \right) \\
&= \frac{1}{2} \left(\nabla^T \bar{x}^{(i)} (A^{(i)} - I)^T + \nabla^T u_f^{(i)} A^{(i)T} + (A^{(i)} - I) \nabla \bar{x}^{(i)} + A^{(i)} \nabla u_f^{(i)} \right. \\
&\quad + \nabla^T \bar{x}^{(i)} (A^{(i)} - I)^T (A^{(i)} - I) \nabla \bar{x}^{(i)} + \nabla^T u_f^{(i)} A^{(i)T} (A^{(i)} - I) \nabla \bar{x}^{(i)} \\
&\quad \left. + \nabla^T \bar{x}^{(i)} (A^{(i)} - I)^T A^{(i)} \nabla u_f^{(i)} + \nabla^T u_f^{(i)} A^{(i)T} A^{(i)} \nabla u_f^{(i)} \right)
\end{aligned}$$

Der Gradient von $\bar{x}^{(i)}$ ist die 2×2 Einheitsmatrix und somit ergibt sich:

$$\begin{aligned}
E(u_{total}^{(i)}) &= \frac{1}{2} \left((A^{(i)} - I)^T + \nabla^T u_f^{(i)} A^{(i)T} + (A^{(i)} - I) + A^{(i)} \nabla u_f^{(i)} \right. \\
&\quad + (A^{(i)} - I)^T (A^{(i)} - I) + \nabla^T u_f^{(i)} A^{(i)T} (A^{(i)} - I) \\
&\quad \left. + (A^{(i)} - I)^T A^{(i)} \nabla u_f^{(i)} + \nabla^T u_f^{(i)} A^{(i)T} A^{(i)} \nabla u_f^{(i)} \right)
\end{aligned}$$

Unter der Ausnutzung, dass $A^{(i)}$ orthogonal ist, d.h. insbesondere:

$$I = A^{(i)T} A^{(i)},$$

kann der obige Ausdruck für $E(u_{total}^{(i)})$ wie folgt vereinfacht werden:

$$E(u_{total}^{(i)}) = \frac{1}{2} \left(\nabla^T u_f^{(i)} + \nabla u_f^{(i)} + \nabla^T u_f^{(i)} \nabla u_f^{(i)} \right) \quad (4.6)$$

$$= E(u_f^{(i)}) \quad (4.7)$$

4.3 Diskretisierung

Seien V_h und $V_{0,h} = V_h \cap H_{0,D}^1(\Omega)$ die Räume aus Abschnitt 3.5 basierend auf der Triangulierung $\mathcal{T}_h(\Omega)$.

Ziel ist es die Lösung des Minimierungsproblems

$$\min_{u_{total} \in V_{0h}} \frac{1}{2} \int_{\Omega} \overline{\overline{D}} E(u_{total}) : E(u_{total}) - \langle f, u_{total} \rangle \quad (4.8)$$

zu finden, von dem wir eindeutige Lösbarkeit voraussetzen.

Aufgrund von Satz 3.2 ist die Lösung von (4.8) auch Lösung von Problem (3.6), d.h. $\underline{k}_h(\underline{u}_h) = \underline{f}_h$. Annahme: Jedes Teilgebiet Ω_i ist die Vereinigung von Elementen von $\mathcal{T}_h(\Omega)$.

Somit besitzt Ω_i die lokale Triangulierung $\mathcal{T}_h(\Omega_i)$.
 Restriktion von V_{0h} auf Ω_i :

$$V_h^{(i)} = \left\{ v \in C\left(\overline{\Omega}^{(i)}, \mathbb{R}^2\right) : \forall T \in \mathcal{T}_h : v \in (P_1)^2 \right\}$$

4.4 Linearisierung „um den Frame“

Im Folgenden treffen wir eine Modellannahme, welche den Grad der Nichtlinearität vereinfachen wird. Wir nehmen an, dass die reaktive Verschiebung $u_f^{(i)}$ abhängig von der Lösung u_{total} , gegeben durch Lemma 4.1, im Gebiet Ω_i klein ist. Aus Kapitel 2 wissen wir, dass bei kleinen Verschiebungen das Approximieren des nichtlinearen Verzerrungstensors $E(u_f^{(i)})$ durch den linearen Verzerrungstensor $\epsilon(u_f^{(i)})$ zulässig ist. Wegen Formel (4.6) gilt $E(u_{total}|_{\Omega_i}) = E(u_f^{(i)})$. Also insgesamt:

$$E(u_{total}|_{\Omega_i}) = E(u_f^{(i)}) \approx \epsilon(u_f^{(i)}) \tag{4.9}$$

Das Integral in (4.8) über Ω lässt sich problemlos als Summe über Teilintegrale schreiben. Wir erhalten also nach Benutzung der Approximation (4.9) folgendes Minimierungsproblem:

$$\min_{u_{total} \in V_{0h}} \frac{1}{2} \sum_{i=1}^N \left[\int_{\Omega_i} \overline{D}\epsilon(u_f^{(i)}) : \epsilon(u_f^{(i)}) \right] - \langle f, u_{total} \rangle \tag{4.10}$$

mit $u_t^{(i)}, \varphi^{(i)}, u_f^{(i)}$ aus dem Lemma 4.1, wobei

$$\langle f^{(i)}, u \rangle = \int_{\Omega^{(i)}} f \cdot u \, dx + \int_{\partial\Omega^{(i)} \cap \Gamma_N} g_N \cdot u \, ds$$

Es ist nun das Ziel, eine (bijektive) Variablentransformation durchzuführen, sodass nicht mehr u_{total} die gesuchte Funktion ist, sondern $(u_t^{(i)}, \varphi^{(i)}, u_f^{(i)})_{i=1}^N$. Außerdem soll auf eine Vektorschreibweise übergegangen werden.

4.5 Variablentransformation

Annahme: Ω ist ein Rechteck und wird wie in Beispiel (4.1) so zerlegt, dass es maximal einen linken und rechten Nachbar hat. Dies ist keine wesentliche Einschränkung, aber es müssen die Kopplungsbedingungen entsprechend angepasst werden.

4.5.1 Notation

Um den Überblick über die verschiedenen Indices nicht zu verlieren, wird hier die Notation festgelegt.

Das Gebiet Ω wird in $N \in \mathbb{N}$ Teilgebiete Ω_i zerlegt.

Die Zerlegung des Teilgebietes Ω_i besteht aus $N^{(i)}$ Knoten.

Das Teilgebiet Ω_i hat $N_D^{(i)}$ Knoten mit Dirichletrandbedingungen.

Die Anzahl der Knoten des Gebietes Ω_i , die an eine benachbartes Gebiet Ω_{i-1} oder Ω_{i+1} angrenzen, wird mit $N_C^{(i)}$ bezeichnet.

Die Variablen, die mit u bezeichnet werden, sind Funktionen.

$u_f^{(i)} \in V_h^{(i)}$ bezeichnet den flexiblen Teil der Verschiebung im Gebiet i

$u_t^{(i)}$ bezeichnet die Translation des Rahmens i

$\varphi^{(i)}$ bezeichnet den Drehwinkel des i ten Rahmens

Die Variablen, die mit q bezeichnet werden, sind die Vektoren, die zu den jeweiligen Funktionen gehören.

Die gesamte Verschiebung hängt eben von diesen drei Komponenten ab:

$q_t^{(i)} \in \mathbb{R}^2$ ist der Anteil der Translation des Rahmens.

$q_\varphi^{(i)} = \varphi_i \in \mathbb{R}$ ist der Drehwinkel des Rahmens.

$q_f^{(i)} \in \mathbb{R}^{2N^{(i)}}$ ist der Anteil der flexiblen Verschiebung relativ zum Rahmen Ω_i .

$x^{(i)} \in \mathbb{R}^{2N^{(i)}}$ sind die Koordinaten der Knoten relativ zum jeweiligen Frame Ω_i .

Diese Komponenten fassen wir in dem Vektor $q^{(i)} \in \mathbb{R}^{2N^{(i)}+3}$ in folgender Form zusammen:

$$q^{(i)} = \left[q_t^{(i)}, q_\varphi^{(i)}, q_f^{(i)} \right]$$

4.5.2 Restriktionen und Randbedingungen

Wir betrachten die Randbedingungen und Restriktionen auf einem der Teilgebiete Ω_i , da dies wesentlich übersichtlicher ist, als das gesamte Gebiet auf einmal zu betrachten.

Es gibt zwei Arten von Restriktionen. Die erste Art hat nur Einfluss auf die relative Verschiebung, also auf den lokalen Teil der Verschiebung. Die zweite Art fixiert die gesamte(globale) Verschiebung.

Bei der ersten Art der Restriktionen geht es darum, die Bedingungen (4.3), (4.4) aus Lemma 4.1 zu fordern, damit die flexible Verschiebung eindeutig bestimmt ist. Wir wählen für $X_1^{(i)}, X_2^{(i)}$ die mittleren Punkte am unteren und oberen Rand des Gebietes Ω_i . Außerdem nehmen wir an, dass diese der k -te und l -te Knoten sind. So ist nun im Knoten k die x - und y - Komponente von u_f gesperrt. Im Knoten l führt die Bedingung (4.4) darauf, dass nur der Wert der x - Komponente auf Null gesetzt ist.

In Vektorschreibweise:

$$q_{f,k,1} = 0$$

$$q_{f,k,2} = 0$$

$$q_{f,l,1} = 0$$

Somit gibt es eine Matrix $B_D^{(i)} \in \mathbb{R}^{3 \times 2N^{(i)}}$, sodass die Bedingung (4.3), (4.4) äquivalent ist zu:

$$B_D^{(i)} q_f^{(i)} = 0 \tag{4.11}$$

Die Einträge der Matrix sind fast überall null. Nur die folgenden Einträge sind ungleich null:

$$\begin{aligned} B_D^{(i)}[2k, 2k] &= 1 \\ B_D^{(i)}[2k + 1, 2k + 1] &= 1 \\ B_D^{(i)}[2l, 2l] &= 1 \end{aligned}$$

Mit Bedingung (4.11) ist die flexible Verschiebung $q_f^{(i)}$ eindeutig bestimmt.

Die zweite Art von Restriktionen wirkt auf die gesamte Verschiebung. Dies betrifft einerseits die Dirichletbedingungen, andererseits eine Stetigkeitsbedingung.

Für die Dirichletbedingungen auf $\partial\Omega_i \cap \Gamma_D$ führen wir eine Matrix $B_{Dir}^{(i)}$ ein, sodass $u_{total}^{(i)} = 0$ auf $\partial\Omega_i \cap \Gamma_D$ äquivalent ist zu

$$B_{Dir}^{(i)} q_{total}^{(i)} = 0, \tag{4.12}$$

wobei $q_{total}^{(i)}$ den $u_{total}^{(i)}$ entsprechenden Vektor bezeichnet.

Damit die Lösung auch auf $\overline{\Omega_i} \cap \overline{\Omega_{i+1}}$ stetig ist, muss explizit die Gleichheit der Werte an den Knoten am rechten Rand von Ω_i und am linken Rand von Ω_{i+1} gefordert werden. Natürlich ist auch wie bei den Dirichletbedingungen die gesamte Verschiebung maßgeblich. Dafür ergibt sich die Matrix $B_C^{(i)} \in \mathbb{R}^{N_C^{(i)} \times N^{(i)}}$.

Die Matrix hat als Eintrag eine Eins, falls der zugehörige Knoten am linken Rand des Rahmens liegt oder minus Eins, falls der Knoten am rechten Rand des Rahmens liegt. Falls der Rand nicht mit einem anderen Rand gekoppelt ist (linke und rechter Rand des Gebietes), ist der Eintrag Null.

Die Stetigkeit einer stückweisen Verschiebung $u_{total}^{(i)}$ ist dann äquivalent zu

$$\sum_{i=1}^N B_C^{(i)} q_{total}^{(i)} = 0 \tag{4.13}$$

Wie bereits oben angeführt, haben wir uns auf den Fall beschränkt, dass die Schnittmenge zwischen zwei Teilen $\overline{\Omega_i}$ und $\overline{\Omega_j}$ nur eine Kante besitzt oder leer ist. Diese Bedingung kann natürlich weggelassen werden, wenn man die Kopplungsbedingungen entsprechend anpasst. Siehe z.B. [6] oder [7].

Für die eingangs erwähnte Variablentransformation müssen wir nun $q_{total}^{(i)}$ in (4.12) und (4.13) durch $[q_t^{(i)}, q_\varphi^{(i)}, q_f^{(i)}]$ darstellen.

Aus $u_{total}^{(i)} = u_{total}^{(i)}(u_t^{(i)}, u_\varphi^{(i)}, u_f^{(i)})$ ergibt sich für $q_{total}^{(i)}$

$$q_{total}^{(i)} = B_t^{(i)} q_t^{(i)} + A^{(i)}(x^{(i)} + q_f^{(i)}) - x^{(i)}$$

wobei

$$A^{(i)} \in \mathbb{R}^{2N^{(i)} \times 2N^{(i)}}$$

$$A^{(i)} = \begin{pmatrix} \tilde{A}^{(i)} & 0 & 0 \dots & 0 & 0 \\ 0 & \tilde{A}^{(i)} & 0 \dots & 0 & 0 \\ \vdots & 0 & \ddots & 0 & \vdots \\ 0 & \dots & 0 & \tilde{A}^{(i)} & 0 \\ 0 & \dots & 0 & 0 & \tilde{A}^{(i)} \end{pmatrix}$$

mit

$$\tilde{A}^{(i)} = \begin{pmatrix} \cos(\varphi_i) & -\sin(\varphi_i) \\ \sin(\varphi_i) & \cos(\varphi_i) \end{pmatrix}$$

und

$$B_t^{(i)} \in \mathbb{R}^{2N^{(i)} \times 2}$$

$$B_t^{(i)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

4.6 Das zugehörige Lagrangefunktional

Zur Optimierung unter Nebenbedingungen verwendet man das Lagrangefunktional. Wir betrachten hier zuerst nicht die Finite Elemente Formulierung, sondern arbeiten mit der Funktion u und setzen dann die Finite Elemente Formulierung ein.

Wir suchen eine Funktion $u_{total}(u_t, \varphi, u_f)$, sodass das

$$\frac{1}{2} \sum_{i=1}^N \int_{\Omega_i} \overline{\overline{D}} \epsilon(u_f^{(i)}) : \epsilon(u_f^{(i)}) - \langle f, u_{total}(u_t, \varphi, u_f) \rangle$$

möglichst klein wird. Die Funktion u muss sowohl die Dirichletbedingungen, die Koppelungsbedingungen und auch die Restriktionen auf dem Gebiet Ω_i erfüllen.

Also:

$$\min \begin{matrix} u_t^{(i)} \in \mathbb{R}^2 \\ \varphi^{(i)} \in \mathbb{R} \\ u_f^{(i)} \in V_h^{(i)} \\ B_D^{(i)} u_f^{(i)} = 0 \\ B_{Dir}^{(i)} u_{total}^{(i)}(u_t^{(i)}, \varphi^{(i)}, u_f^{(i)}) = 0 \\ \sum_{i=1}^N B_C^{(i)} u_{total}^{(i)}(u_t^{(i)}, \varphi^{(i)}, u_f^{(i)}) = 0 \end{matrix} \quad \sum_{i=1}^N \left[\frac{1}{2} (\overline{\overline{D}} \epsilon(u_f^{(i)}), \epsilon(u_f^{(i)}))_{L^2(\Omega_i)} - \langle f, u_{total}^{(i)}(u_t^{(i)}, \varphi^{(i)}, u_f^{(i)}) \rangle_{\Omega_i} \right]$$

Hier haben wir der Einfachheit halber für die Operatoren $B_D^{(i)}$, $B_{Dir}^{(i)}$, $B_C^{(i)}$ dieselbe Bezeichnung wie für ihre zugehörigen Matrizen verwendet.

Somit bekommt man für das Lagrangefunktional folgenden Ausdruck:

$$\begin{aligned} \mathcal{L}((u_t, \varphi, u_f), \lambda) &= \sum_{i=1}^N \left[\frac{1}{2} \left(\overline{D} \epsilon(u_f^{(i)}), \epsilon(u_f^{(i)}) \right)_{L^2(\Omega_i)} \right. \\ &\quad - \langle f^{(i)}, u_{total}^{(i)}(u_t^{(i)}, \varphi^{(i)}, u_f^{(i)}) \rangle_{\Omega_i} + \langle \lambda_D^{(i)}, B_D^{(i)} u_f^{(i)} \rangle_{\Omega_i} \\ &\quad \left. + \langle \lambda_{Dir}^{(i)}, B_{Dir}^{(i)} u_{total}^{(i)}(u_t^{(i)}, \varphi^{(i)}, u_f^{(i)}) \rangle_{\Omega_i} \right] \\ &\quad + \langle \lambda_C^{(i)}, \sum_{i=1}^N B_C^{(i)} u_{total}^{(i)}(u_t^{(i)}, \varphi^{(i)}, u_f^{(i)}) \rangle_{\Omega} \end{aligned}$$

Das Langrangefunktional hängt also von den Funktionen u_t , φ und u_f und den Langrangeparametern λ_D , λ_C und λ_{Dir} ab.

Der Langrangeparameter λ_D gehört zu den lokalen Restriktionen (Sperrung der Komponenten der flexiblen Verschiebung u_f), λ_C ist der zu den Kopplungen gehörige Parameter, λ_{Dir} berücksichtigt die Dirichletrandbedingungen.

Die Zusammenhänge werden dann im nachfolgenden Teil ausgeführt. Für die bessere Lesbarkeit gehen wir von der Funktionenschreibweise auf eine Vektorschreibweise über, die durch den Ritz-Isomorphismus gegeben ist (vgl Abschnitt 4.5.1):

$$\begin{aligned} u_t^{(i)} &= q_t^{(i)} \\ \varphi^{(i)} &= q_\varphi^{(i)} \\ u_f^{(i)} &\rightsquigarrow q_f^{(i)} \\ f^{(i)} &\rightsquigarrow f_h^{(i)} \end{aligned}$$

In der Vektorschreibweise ergeben sich folgende Zusammenhänge:

$$\begin{aligned} \frac{1}{2} \left(D\varepsilon(u_f^{(i)}), \varepsilon(u_f^{(i)}) \right)_{L^2(\Omega_i)} &= \left(\frac{1}{2} K_{ff} q_f^{(i)}, q_f^{(i)} \right)_{l_2} \\ \langle f^{(i)}, u_{total}(u_t, \varphi, u_f) \rangle_{\Omega} &= \left(f_h^{(i)}, B_t^{(i)} q_t^{(i)} + A^{(i)} (x^{(i)} + q_f^{(i)}) - x^{(i)} \right)_{l_2} \\ & \quad B_t^{(i)} \in \mathbb{R}^{N^{(i)} \times 2} \\ \langle \lambda_D^{(i)}, B_D^{(i)} u_f^{(i)} \rangle_{\Omega^{(i)}} &= \lambda_D^{(i)T} B_D^{(i)} q_f^{(i)} \\ & \quad B_D^{(i)} \in \mathbb{R}^{3 \times N^{(i)}} \\ \langle \lambda_{Dir}^{(i)}, B_{Dir}^{(i)} u_{total}(u_t, \varphi, u_f) \rangle_{\Omega^{(i)}} &= \lambda_{Dir}^{(i)T} B_{Dir}^{(i)} \left(B_t q_t^{(i)} + A^{(i)} (x^{(i)} + q_f^{(i)}) - x^{(i)} \right) \\ & \quad B_{Dir}^{(i)} \in \mathbb{R}^{N_{Dir}^{(i)} \times N^{(i)}} \\ \langle \lambda_C, \sum_{i=1}^N B_C^{(i)} u_{total}(u_t, \varphi, u_f) \rangle_{\Omega} &= \lambda_C^T \sum_{i=1}^N B_C^{(i)} \left(B_t q_t^{(i)} + A^{(i)} (x^{(i)} + q_f^{(i)}) - x^{(i)} \right) \\ & \quad B_C^{(i)} \in \mathbb{R}^{N_C^{(i)} \times N^{(i)}} \end{aligned}$$

Somit erhält man dann folgenden Ausdruck für das Lagrangefunktional:

$$\begin{aligned} \mathcal{L}(q, \lambda) &= \sum_{i=1}^N \left[\left(\frac{1}{2} K_{ff}^{(i)} q_f^{(i)}, q_f^{(i)} \right)_{l_2} - \left(f_h^{(i)}, B_t^{(i)} q_t^{(i)} + A^{(i)} (x^{(i)} + q_f^{(i)}) - x^{(i)} \right)_{l_2} \right. \\ & \quad \left. + \lambda_D^{(i)T} B_D^{(i)} q_f^{(i)} + \lambda_{Dir}^{(i)T} B_{Dir}^{(i)} \left(B_t^{(i)} q_t^{(i)} + A^{(i)} (x^{(i)} + q_f^{(i)}) - x^{(i)} \right) \right] \\ & \quad + \lambda_C^T \sum_{i=1}^N B_C^{(i)} \left(B_t^{(i)} q_t^{(i)} + A^{(i)} (x^{(i)} + q_f^{(i)}) - x^{(i)} \right) \end{aligned}$$

4.6.1 Gradient des Lagrangefunktionals

Für die Lösung des nichtlinearen Problems benötigt man die Ableitung des Lagrangefunktionals. Um diese zu berechnen wird nach den einzelnen Komponenten $(q_t, q_\varphi, q_f, \lambda_D, \lambda_C, \lambda_{Dir})$ abgeleitet. Um die Übersicht zu verbessern wird der Index (i) bei den Vektoren sowie die Summe $1, \dots, N$ weggelassen.

Ableitung nach q_t :

$$\begin{aligned} \frac{\partial \mathcal{L}(q, \lambda)}{\partial q_t} &= f_h^T B_t + B_t^T \lambda_{Dir}^T B_{Dir} + B_t^T \lambda_C^T B_C \\ &= B_t^T (f_h + \lambda_{Dir}^T B_{Dir} + \lambda_C^T B_C) \end{aligned}$$

Ableitung nach q_φ :

$$\begin{aligned} \frac{\partial \mathcal{L}(q, \lambda)}{\partial q_\varphi} &= (-A'(x + q_f))^T f_h + \lambda_{Dir}^T B_{Dir} A'(x + q_f) + \lambda_C^T B_C A'(x + q_f) \\ &= \left(-f_h^T + \lambda_{Dir}^T B_{Dir} + \lambda_C^T B_C \right) A'(x + q_f) \end{aligned}$$

wobei

$$A^{(i)'} \in \mathbb{R}^{2N^{(i)} \times 2N^{(i)}}$$

$$A^{(i)'} = \begin{pmatrix} \tilde{A}^{(i)'} & 0 & 0 \dots & 0 & 0 \\ 0 & \tilde{A}^{(i)'} & 0 \dots & 0 & 0 \\ \vdots & 0 & \ddots & 0 & \vdots \\ 0 & \dots & 0 & \tilde{A}^{(i)'} & 0 \\ 0 & \dots & 0 & 0 & \tilde{A}^{(i)'} \end{pmatrix}$$

mit

$$\tilde{A}^{(i)'} = \begin{pmatrix} -\sin(\varphi_i) & -\cos(\varphi_i) \\ \cos(\varphi_i) & -\sin(\varphi_i) \end{pmatrix}$$

Ableitung nach q_f :

$$\begin{aligned} \frac{\partial \mathcal{L}(q, \lambda)}{\partial q_f} &= K_{ff} q_f - f_h^T A + \lambda_D^T B_D + A^T B_{Dir}^T \lambda_{Dir} + A^T B_C^T \lambda_C \\ &= K_{ff} q_f + \lambda_D^T B_D + A^T (-f_h + B_{Dir}^T \lambda_{Dir} + B_C^T \lambda_C) \end{aligned}$$

Ableitung nach λ_D :

$$\frac{\partial \mathcal{L}(q, \lambda)}{\partial \lambda_D} = B_D q_f$$

Ableitung nach λ_{Dir} :

$$\frac{\partial \mathcal{L}(q, \lambda)}{\partial \lambda_{Dir}} = B_{Dir} (B_t q_t + A(x + q_f) - x)$$

Ableitung nach λ_C :

$$\frac{\partial \mathcal{L}(q, \lambda)}{\partial \lambda_C} = B_C (B_t q_t + A(x + q_f) - x)$$

Somit bekommt man für $\nabla_{q, \lambda} \mathcal{L}(q, \lambda)$ folgenden Vektor:

$$\nabla_{q, \lambda} \mathcal{L}(q, \lambda) = \begin{pmatrix} B_t^T (f_h + \lambda_{Dir}^T B_{Dir} + \lambda_C^T B_C) \\ \left(-f_h^T + \lambda_{Dir}^T B_{Dir} + \lambda_C^T B_C \right) A'(x + q_f) \\ K_{ff} q_f + \lambda_D^T B_D + A^T (-f_h + B_{Dir}^T \lambda_{Dir} + B_C^T \lambda_C) \\ B_D q_f \\ B_{Dir} (B_t q_t + A(x + q_f) - x) \\ B_C (B_t q_t + A(x + q_f) - x) \end{pmatrix}$$

Auch hier wurde wegen der besseren Übersicht auf die Indices (i) und Summen verzichtet.

4.7 Lösen des Minimierungsproblems via Lagrange Funktional

Das Ziel ist es, das Lagrange Funktional zu minimieren. Um den kritischen Punkt zu berechnen, setzen wir den Gradient auf Null.

$$\nabla_{q,\lambda}\mathcal{L}(q, \lambda) = 0 \quad (4.14)$$

Da diese Gleichung nach wie vor nichtlinear ist, wird zum Lösen das Newton Verfahren verwendet.

4.7.1 Newton Verfahren

Analog zu Kapitel 3.7.2 auf Seite 25 verwenden wir für die Gleichung (4.14) wieder das gedämpfte Newton Verfahren.

Mit folgender Notation für die Jacobimatrix des Lagrangefunktionals,

$$J(q, \lambda) := \nabla\nabla\mathcal{L}(q, \lambda),$$

ergibt sich folgender Iterationsalgorithmus:

Sei q^0, λ^0 gegeben, $k = 0, 1, \dots$

Löse

$$J\left(\begin{pmatrix} q^{(k)} \\ \lambda^{(k)} \end{pmatrix}\right) w^{(k)} = d^{(k)} \quad (4.15)$$

Mit

$$w^{(k)} = \begin{pmatrix} w_q^{(k)} \\ w_\lambda^{(k)} \end{pmatrix}$$

und

$$d^{(k)} = \nabla\mathcal{L}\left(\begin{pmatrix} q^{(k)} \\ \lambda^{(k)} \end{pmatrix}\right)$$

und subtrahiere dann das gedämpfte Update $\begin{pmatrix} w_q^{(k)} \\ w_\lambda^{(k)} \end{pmatrix}$.

$$\begin{pmatrix} q^{(k+1)} \\ \lambda^{(k+1)} \end{pmatrix} = \begin{pmatrix} q^{(k)} \\ \lambda^{(k)} \end{pmatrix} - \tau^{(k)} \begin{pmatrix} w_q^{(k)} \\ w_\lambda^{(k)} \end{pmatrix}$$

Wobei wiederum $\tau^{(k)}$ so klein gewählt wird, dass $\|\mathcal{L}\left(\begin{pmatrix} q^{(k)} \\ \lambda^{(k)} \end{pmatrix}\right)\| > \|\mathcal{L}\left(\begin{pmatrix} q^{(k+1)} \\ \lambda^{(k+1)} \end{pmatrix}\right)\|$

Die Updates bei den Iterationsschritten werden mit $w^{(k)}$ bezeichnet. Der Index bezieht sich auf Zugehörigkeit des Eintrags. Der Vektor $w^{(k)}$ besteht aus:

$$w_q^{(k)} = \begin{pmatrix} w_t^{(k)} \\ w_\varphi^{(k)} \\ w_f^{(k)} \end{pmatrix}, \quad w_\lambda^{(k)} = \begin{pmatrix} w_D^{(k)} \\ w_{Dir}^{(k)} \\ w_C^{(k)} \end{pmatrix}$$

4.7.2 Jacobimatrix des Langrangefunktional

Für das Newtonverfahren wird die Jacobimatrix benötigt. Um diese zu berechnen benötigt man folgende Ableitungen:

$$J \left(\begin{pmatrix} q_k \\ \lambda_k \end{pmatrix} \right) = \begin{pmatrix} \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_t \partial q_t} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_\varphi \partial q_t} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_f \partial q_t} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_D \partial q_t} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_{Dir} \partial q_t} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_C \partial q_t} \\ \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_t \partial q_\varphi} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_\varphi \partial q_\varphi} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_f \partial q_\varphi} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_D \partial q_\varphi} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_{Dir} \partial q_\varphi} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_C \partial q_\varphi} \\ \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_t \partial q_f} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_\varphi \partial q_f} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_f \partial q_f} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_D \partial q_f} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_{Dir} \partial q_f} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_C \partial q_f} \\ \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_t \partial \lambda_D} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_\varphi \partial \lambda_D} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_f \partial \lambda_D} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_D \partial \lambda_D} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_{Dir} \partial \lambda_D} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_C \partial \lambda_D} \\ \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_t \partial \lambda_{Dir}} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_\varphi \partial \lambda_{Dir}} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_f \partial \lambda_{Dir}} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_D \partial \lambda_{Dir}} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_{Dir} \partial \lambda_{Dir}} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_C \partial \lambda_{Dir}} \\ \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_t \partial \lambda_C} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_\varphi \partial \lambda_C} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_f \partial \lambda_C} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_D \partial \lambda_C} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_{Dir} \partial \lambda_C} & \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_C \partial \lambda_C} \end{pmatrix}$$

Es ergeben sich dann folgende Ableitungen:

1. Zeile:

$$\begin{aligned} \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_t \partial q_t} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_\varphi \partial q_t} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_f \partial q_t} &= \begin{pmatrix} 0 & \dots & 0 \\ 0 & \dots & 0 \end{pmatrix} \\ \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_D \partial q_t} &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_{Dir} \partial q_t} &= B_t^T B_{Dir}^T \\ \frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_C \partial q_t} &= B_t^T B_C^T \end{aligned}$$

2. Zeile:

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_t \partial q_\varphi} = \begin{pmatrix} 0 & 0 \end{pmatrix}$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_\lambda \partial q_\varphi} = (A(x + q_f))^T (-f^T + B^T \lambda)$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_f \partial q_\varphi} = (\lambda^T B - f^T) A'$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_D \partial q_\varphi} = (0 \ 0 \ 0)$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_{Dir} \partial q_\varphi} = (A'(x + q_f))^T B_{Dir}^T$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_C \partial q_\varphi} = (A'(x + q_f))^T B_C^T$$

3. Zeile:

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_t \partial q_f} = \begin{pmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix}$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_\varphi \partial q_f} = A'^T (B^T \lambda - f)$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_f \partial q_f} = K_{ff}$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_D \partial q_f} = B_D^T$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_{Dir} \partial q_f} = A'^T B_{Dir}^T$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_C \partial q_f} = A'^T B_C^T$$

4. Zeile:

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_t \partial \lambda_D} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_\varphi \partial \lambda_D} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_f \partial \lambda_D} = B_D$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_D \partial \lambda_D} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_{Dir} \partial \lambda_D} = \begin{pmatrix} 0 & \dots & 0 \\ 0 & \dots & 0 \\ 0 & \dots & 0 \end{pmatrix}$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_C \partial \lambda_D} = \begin{pmatrix} 0 & \dots & 0 \\ 0 & \dots & 0 \\ 0 & \dots & 0 \end{pmatrix}$$

5. Zeile:

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_t \partial \lambda_{Dir}} = B_{Dir} B_t$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_\varphi \partial \lambda_{Dir}} = B_{Dir} A' (x + q_f)$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_f \partial \lambda_{Dir}} = B_{Dir} A$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_D \partial \lambda_{Dir}} = \begin{pmatrix} 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{pmatrix}$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_{Dir} \partial \lambda_{Dir}} = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{pmatrix}$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_C \partial \lambda_{Dir}} = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{pmatrix}$$

6. Zeile:

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_t \partial \lambda_C} = B_C B_t$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_\varphi \partial \lambda_C} = B_C A' (x + q_f)$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_f \partial \lambda_C} = B_C A$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_D \partial \lambda_C} = \begin{pmatrix} 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{pmatrix}$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_{Dir} \partial \lambda_C} = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{pmatrix}$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial \lambda_C \partial \lambda_C} = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{pmatrix}$$

Somit ergibt sich für die Jacobimatrix:

$$J(q, \lambda) =$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & B_t^T B_{Dir}^T & B_t^T B_C^T \\ 0 & (A(x + q_f))^T (-f^T + B^T \lambda) & (\lambda^T B - f^T) A' & 0 & (A'(x + q_f))^T B_{Dir}^T & (A'(x + q_f))^T B_C^T \\ 0 & A'^T (B^T \lambda - f) & K_{ff} & B_D^T & A^T B_{Dir}^T & A^T B_C^T \\ 0 & 0 & B_D & 0 & 0 & 0 \\ B_{Dir} B_t & B_{Dir} A'(x + q_f) & B_{Dir} A & 0 & 0 & 0 \\ B_C B_t & B_C A'(x + q_f) & B_C A & 0 & 0 & 0 \end{pmatrix} \quad (4.16)$$

4.7.3 Lösung der Newton-Udate Gleichung

Das Lösen von (4.15) mittels direktem Löser wäre teurer als das Lösen der nichtlinearen Gleichung in Kapitel 3.7.2 mittels direktem Löser.

Das Ziel ist es, einen iterativen Löser für die Gleichung (4.15) zu finden. Bei der Betrachtung der Jacobimatrix (4.16) stellt man eine Ähnlichkeit zu Systemen, die bei FETI Methoden [6] vorkommen, fest. Um die bekannten FETI-Techniken voll auszunutzen, wird (4.15) durch folgendes System ersetzt:

$$\tilde{J} \begin{pmatrix} q^{(k)} \\ \lambda^{(k)} \end{pmatrix} w^{(k)} = d^{(k)} \quad (4.17)$$

Die modifizierte Jacobimatrix \tilde{J} ergibt sich aus der Vernachlässigung folgender Terme:

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_\varphi \partial q_\varphi} \approx 0$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_\varphi \partial q_f} \approx (0 \quad \dots \quad 0)$$

$$\frac{\partial^2 \mathcal{L}(q, \lambda)}{\partial q_f \partial q_\varphi} \approx \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

Mit dieser Änderung ergibt sich für die modifizierte Jacobimatrix $\tilde{J}(q, \lambda)$:

$$\tilde{J}(q, \lambda) = \begin{pmatrix} 0 & 0 & 0 & 0 & B_t^T B_{Dir}^T & B_t^T B_C^T \\ 0 & 0 & 0 & 0 & (A'(x+q_f))^T B_{Dir}^T & (A'(x+q_f))^T B_C^T \\ 0 & 0 & K_{ff} & B_D^T & A^T B_{Dir}^T & A^T B_C^T \\ 0 & 0 & B_D & 0 & 0 & 0 \\ B_{Dir} B_t & B_{Dir} A'(x+q_f) & B_{Dir} A & 0 & 0 & 0 \\ B_C B_t & B_C A'(x+q_f) & B_C A & 0 & 0 & 0 \end{pmatrix}$$

Dies führt auf ein modifiziertes Newtonverfahren.

Wir hoffen, dass diese Änderung die Konvergenz nicht allzusehr beeinträchtigt. Dazu siehe Kapitel 5.

4.7.4 Struktur der modifizierten Newton-Update Gleichung

Um die Struktur der Matrix \tilde{J} besser hervorzuheben und somit in FETI Notation zu bringen, werden einzelne Bereiche der Matrix zusammengefasst.

Auf den Iterationsindex k wird ab jetzt verzichtet.

Wir definieren dazu einige neue Matrizen.

$$\begin{aligned} B_r &= \begin{pmatrix} B_{Dir} B_t & B_{Dir} A'(x+q_f) \\ B_C B_t & B_C A'(x+q_f) \end{pmatrix} \\ B_f &= \begin{pmatrix} B_{Dir} A & 0 \\ B_C A & 0 \end{pmatrix} \\ K_{ff}^* &= \begin{pmatrix} K_{ff} & B_D^T \\ B_D & 0 \end{pmatrix} \end{aligned}$$

Dafür ergibt sich für den Vektor w folgende Gruppierung

$$w_r = \begin{pmatrix} w_t \\ w_\varphi \end{pmatrix}$$

$$w_f = \begin{pmatrix} w_f \\ w_D \end{pmatrix}$$

$$w_\lambda = \begin{pmatrix} w_{Dir} \\ w_C \end{pmatrix}$$

Mit dieser Setzung ist das System (4.17), also

$$\tilde{J}(q, \lambda) w = d$$

von der Form

$$\begin{pmatrix} 0 & 0 & B_r^T \\ 0 & K_{ff}^* & B_f^T \\ B_r & B_f & 0 \end{pmatrix} \begin{pmatrix} w_r \\ w_f \\ w_\lambda \end{pmatrix} = \begin{pmatrix} d_r \\ d_f \\ d_\lambda \end{pmatrix}. \quad (4.18)$$

Der Vektor $(d_r \ d_f \ d_\lambda)^T$ ist hierbei wie folgt definiert:

$$\begin{pmatrix} d_r \\ d_f \\ d_\lambda \end{pmatrix} = \nabla \mathcal{L}(q, \lambda),$$

also

$$d_r = \begin{pmatrix} \frac{\partial \mathcal{L}(q, \lambda)}{\partial q_t} \\ \frac{\partial \mathcal{L}(q, \lambda)}{\partial q_\varphi} \end{pmatrix} = \begin{pmatrix} B_t^T (f_h + \lambda_{Dir}^T B_{Dir} + \lambda_C^T B_C) \\ (-f_h^T + \lambda_{Dir}^T B_{Dir} + \lambda_C^T B_C) A' (x + q_f) \end{pmatrix}$$

$$d_f = \begin{pmatrix} \frac{\partial \mathcal{L}(q, \lambda)}{\partial q_f} \\ \frac{\partial \mathcal{L}(q, \lambda)}{\partial \lambda_D} \end{pmatrix} = \begin{pmatrix} K_{ff} q_f + \lambda_D^T B_D + A^T (-f_h + B_{Dir}^T \lambda_{Dir} + B_C^T \lambda_C) \\ B_D q_f \end{pmatrix}$$

$$d_\lambda = \begin{pmatrix} \frac{\partial \mathcal{L}(q, \lambda)}{\partial \lambda_{Dir}} \\ \frac{\partial \mathcal{L}(q, \lambda)}{\partial \lambda_C} \end{pmatrix} = \begin{pmatrix} B_{Dir} (B_t q_t + A(x + q_f) - x) \\ B_C (B_t q_t + A(x + q_f) - x) \end{pmatrix}$$

4.7.5 Lösung der modifizierten Newton-Update Gleichung

In diesem Teil beschäftigen wir uns mit der Lösung der modifizierten Newton-Update Gleichung (4.18) mittels FETI-Technik.

Aus der zweiten Zeile der Formel (4.18) wird der Vektor w_f ausgedrückt:

$$K_{ff}^* w_f + B_f^T w_\lambda = d_f \quad (4.19)$$

$$w_f = K_{ff}^{*-1} (d_f - B_f^T w_\lambda) \quad (4.20)$$

Lemma 4.2. K_{ff}^* ist invertierbar.

Beweis.

$$K_{ff}^* = \begin{pmatrix} K_{ff} & B_D^T \\ B_D & 0 \end{pmatrix}$$

Nach Umordnung der Einträge in K_{ff}^* erhalten wir eine Blockdiagonalmatrix bestehend aus den Matrizen der Teilgebiete.

$$K_{ff}^* = \text{diag} \left(\begin{pmatrix} K_{ff}^{(i)} & B_D^{(i)T} \\ B_D^{(i)} & 0 \end{pmatrix} \right)_{i=1}^N$$

Jeder Block dieser Matrix ist invertierbar, falls ihr Kern nur aus dem Nullvektor besteht.

Der Kern gehört zu folgendem Variationsproblem:

$$\begin{aligned} \langle K_{ff}^{(i)} x, v \rangle + \langle B_D^{(i)T} \lambda, v \rangle &= 0 \quad \forall v \in V_h \\ \langle B_D^{(i)} x, \mu \rangle &= 0 \quad \forall \mu \in \Lambda_h \end{aligned}$$

Aus der zweiten Gleichung $\langle x, B_D^{(i)T} \mu \rangle = 0$ sehen wir, dass x nur Null ist, wenn $\ker(B_D^{(i)T}) = \{0\}$. Aus der ersten Gleichung bekommen wir $\langle K_{ff}^{(i)} x, v \rangle + \langle \lambda, B_D^{(i)} v \rangle = 0$. Damit muss $\ker(K_{ff}^{(i)}) \cap \ker(B_D^{(i)}) = \{0\}$ gelten.

Diese Bedingungen müssen nun gezeigt werden:

$$\begin{aligned} \ker(K_{ff}^{(i)}) \cap \ker(B_D^{(i)}) &= \{0\} \\ \ker(B_D^{(i)T}) &= \{0\} \end{aligned}$$

Der Kern der Matrix $K_{ff}^{(i)}$ sind Funktionen bestehend aus linearisierter Rotation und Translation. Die Einträge der Matrix $B_D^{(i)}$ wurden so gewählt (siehe (4.3)), dass auch diese Funktionen ausgeschlossen sind. Deswegen gilt :

$$\ker(K_{ff}^{(i)}) \cap \ker(B_D^{(i)}) = \{0\}$$

Da die Matrix $B_D^{(i)T}$ nur 3 Zeilen hat und nur 3 Einträge in unterschiedlichen Spalten ungleich Null sind, gilt auch

$$\ker(B_D^{(i)T}) = \{0\}.$$

□

Einsetzen der Formel (4.19) für w_f in die dritte Gleichung von (4.18) führt auf

$$\begin{aligned} B_r w_r + B_f K_{ff}^{*-1} (d_f - B_f^T w_\lambda) &= d_\lambda \\ B_r w_r + B_f K_{ff}^{*-1} d_f - B_f K_{ff}^{*-1} B_f^T w_\lambda &= d_\lambda \end{aligned}$$

Die Matrix F wird wie folgt definiert:

$$F := B_f K_{ff}^{*-1} B_f^T$$

Somit erhalten wir folgendes System in Matrix Form:

$$\begin{pmatrix} F & -B_r \\ B_r^T & 0 \end{pmatrix} \begin{pmatrix} w_\lambda \\ w_r \end{pmatrix} = \begin{pmatrix} B_f K_{ff}^{*-1} d_f - d_\lambda \\ d_r \end{pmatrix}$$

Um dieses System zu lösen muss w_λ beide Gleichungen erfüllen. Man suche zuerst eine Lösung, die die zweite Gleichung erfüllt:

$$B_r^T w_\lambda = d_r$$

Eine Lösung dieser Gleichung werden wir mit w_{λ_0} bezeichnen. Es lässt sich leicht feststellen, dass

$$w_{\lambda_0} = B_r (B_r^T B_r)^{-1} d_r$$

die Bedingung erfüllt.

Um die erste Gleichung zu lösen, suchen wir ein \widetilde{w}_λ , sodass

$$B_r^T \widetilde{w}_\lambda = 0$$

und

$$w_\lambda = w_{\lambda_0} + \widetilde{w}_\lambda$$

Durch das Einsetzen von w_λ in die erste Gleichung bekommt man:

$$F w_\lambda - B_r w_r = B_f K_{ff}^{*-1} d_f - d_\lambda \quad (4.21)$$

$$F (w_{\lambda_0} + \widetilde{w}_\lambda) - B_r w_r = B_f K_{ff}^{*-1} d_f - d_\lambda \quad (4.22)$$

$$F \widetilde{w}_\lambda - B_r w_r = B_f K_{ff}^{*-1} d_f - d_\lambda - F w_{\lambda_0} \quad (4.23)$$

Per Konstruktion soll \widetilde{w}_λ im Kern von B_r^T liegen. Dafür definiert man die Projektion P .

Lemma 4.3. *Sei P wie folgt definiert:*

$$P := I - B_r (B_r^T B_r)^{-1} B_r^T$$

und es gilt:

$$\forall \mu \in \text{range}(P) : B_r^T \mu = 0$$

Dann ist P eine Projektion und symmetrisch. (Die Invertierbarkeit von $B_r^T B_r$ wird in Lemma 4.5 diskutiert.)

Beweis. Man sieht sofort aus der Struktur, dass P symmetrisch ist.

Es bleibt zu zeigen, dass P ein Projektion ist: $P = P^2$

$$\begin{aligned} P^2 &= \left(I - B_r (B_r^T B_r)^{-1} B_r^T \right)^2 \\ &= I^2 - 2B_r (B_r^T B_r)^{-1} B_r^T + B_r (B_r^T B_r)^{-1} B_r^T B_r (B_r^T B_r)^{-1} B_r^T \\ &= I - 2B_r (B_r^T B_r)^{-1} B_r^T + B_r (B_r^T B_r)^{-1} B_r^T \\ &= I - B_r (B_r^T B_r)^{-1} B_r^T \\ &= P \end{aligned}$$

Es lässt sich leicht nachrechnen, dass $B_r^T P = 0$ gilt. □

Die Projektion angewendet auf die Gleichung (4.23) ergibt:

$$P F \widetilde{w}_\lambda - P B_r w_r = P \left(B_f K_{ff}^{*-1} d_f - d_\lambda - F w_{\lambda_0} \right)$$

Wir betrachten nun $P B_r w_r$

$$\begin{aligned} P B_r w_r &= \left(I - B_r (B_r^T B_r)^{-1} B_r^T \right) B_r w_r \\ &= B_r w_r - B_r w_r = 0 \end{aligned}$$

Damit ergibt sich

$$PF\widetilde{w}_\lambda = P \left(B_f K_{ff}^{*-1} d_f - d_\lambda - F w_{\lambda_0} \right)$$

Da $P = P^2$ und $\widetilde{w}_\lambda = P w_\lambda = P^2 w_\lambda = P \widetilde{w}_\lambda$ und $P = P^T$ folgt:

$$PF P^T \widetilde{w}_\lambda = P \left(B_f K_{ff}^{*-1} d_f - d_\lambda - F w_{\lambda_0} \right) \quad (4.24)$$

Lemma 4.4. *F ist symmetrisch positiv definit.*

Beweis. Da K_{ff}^* symmetrisch ist, folgt, dass auch K_{ff}^{*-1} symmetrisch ist. Da von links mit B_f und von rechts mit der transponierten B_f^T multipliziert wird, bleibt die Symmetrie für $F = B_f K_{ff}^{*-1} B_f^T$ erhalten.

Wir zeigen zunächst, dass

$$\langle F\lambda, \lambda \rangle \geq 0 \quad \forall \lambda$$

Dies folgt aus der Definition der lokalen Steifigkeitsmatrix K_{ff} : Die Einträge B_D bzw B_D^T , welche in K_{ff}^* vorkommen, werden mit den 0-Einträgen von B_f bzw- B_f^T multipliziert und haben daher keinen Einfluss. Da K_{ff} positiv semidefinit ist, folgt $\langle F\lambda, \lambda \rangle \geq 0 \forall \lambda$. Es bleibt die Definitheit zu zeigen:

$$\langle F\lambda, \lambda \rangle = 0 \Rightarrow \lambda = 0$$

Aus

$$\begin{aligned} \langle F\lambda, \lambda \rangle &= \langle B_f K_{ff}^{*-1} B_f^T \lambda, \lambda \rangle \\ &= \langle K_{ff}^{*-1} B_f^T \lambda, B_f^T \lambda \rangle = 0 \end{aligned}$$

folgt, dass $B_f^T \lambda = 0$ gelten muss, da $K_{ff}^{*(-1)}$ definit ist.

Die Matrix

$$B_f^T = \begin{pmatrix} B_{Dir} A & 0 \\ B_C A & 0 \end{pmatrix}^T$$

besteht (abgesehen von den 0-Einträgen) aus 2×2 Blöcken

$$\begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix}^T,$$

wobei pro zwei zusammengehörigen Zeilen nur ein Block auftritt. Da jeder Block regulär ist, folgt $\ker(B_f^T) = \{0\}$. Somit besitzt $B_f^T \lambda = 0$ auch nur die Lösung $\lambda = 0$.

Also

$$F = B_f K_{ff}^{*-1} B_f^T$$

ist SPD. □

Die Matrix F ist also symmetrisch positiv definit. Somit ist auch $PFPT^T$ auf dem Unterraum $\tilde{\Lambda} := \{w_\lambda | B_r^T w_\lambda = 0\} = \ker(B_r^T)$ SPD. Um die Gleichung (4.24) zu lösen, kann man wie in [7] das bekannte CG-Verfahren mit einem Startwert aus $\tilde{\Lambda}$, z.B. Null, verwenden. Somit erhalten wir:

$$w_\lambda = w_{\lambda_0} + \tilde{w}_\lambda$$

Mit w_λ lässt sich nun aus dem System

$$\begin{pmatrix} F & -B_r \\ B_r^T & 0 \end{pmatrix} \begin{pmatrix} w_\lambda \\ w_r \end{pmatrix} = \begin{pmatrix} B_f K_{ff}^{*-1} d_f - d_\lambda \\ d_r \end{pmatrix}$$

w_r berechnen:

Aus der ersten Zeile erhält man:

$$\begin{aligned} Fw_\lambda - B_r w_r &= B_f K_{ff}^{*-1} d_f - d_\lambda \\ B_r w_r &= Fw_\lambda - B_f K_{ff}^{*-1} d_f + d_\lambda \\ B_r^T B_r w_r &= B_r^T (Fw_\lambda - B_f K_{ff}^{*-1} d_f + d_\lambda) \\ w_r &= (B_r^T B_r)^{-1} (Fw_\lambda - B_f K_{ff}^{*-1} d_f + d_\lambda) \end{aligned}$$

Wir weisen nun nach, dass $B_r^T B_r$ invertierbar ist.

Lemma 4.5. *Die Matrix $B_r^T B_r$ ist SPD.*

Beweis. Dass $B_r^T B_r$ symmetrisch ist, sieht man. $B_r^T B_r$ ist auch positiv semidefinit:

$$\langle B_r^T B_r v, v \rangle = \langle B_r v, B_r v \rangle \geq 0 \quad \forall v \in \mathbb{R}^{2(N_{Dir} + N_C)}$$

Es bleibt noch zu zeigen, dass $B_r^T B_r$ definit ist, also

$$B_r v = 0 \Rightarrow v = 0$$

Dazu betrachten wir die Struktur von B_r :

$$B_r = \begin{pmatrix} B_{Dir} B_t & B_{Dir} A'(x + q_f) \\ B_C B_t & B_C A'(x + q_f) \end{pmatrix}$$

$$v = \begin{pmatrix} w_x \\ w_y \\ w_\varphi \end{pmatrix}$$

Der folgende Beweis ist induktiv aufgebaut und hängt auch von der der Art der Kopplungen ab. Wir betrachten zuerst einen Rahmen auf dem Dirchletbedingungen vorgegeben sind. Für die weiteren Rahmen kann man dann analog vorgehen, da der Kopplungsrand zu einem Dirichletrand wird, da (aufgrund der Induktionsannahme) mit der Nullfunktion

gekoppelt wird.

Wir betrachten nun den ersten Rahmen mit Dirichletrand. Da die Matrix A' blockdiagonal ist betrachten wir die Werte zweier beliebiger verschiedener Punkte mit Dirichlet Randbedingungen (werden in N_{Dir} gezählt) oder Punkte, die zwischen zwei Rahmen liegen (werden in N_C gezählt). Da die Punkte im gleichen Rahmen liegen, ist der Translationsanteil und der Rotationsanteil der Verschiebung in beiden Punkten gleich und wird mit (w_x, w_y) für die Translation und w_φ für die Rotation bezeichnet.

Bei der Matrix B_r wird der flexible (lokale) Anteil für den ersten Punkt mit $(q_{f_{1x}}, q_{f_{1y}})$ und für den zweiten Punkte mit $(q_{f_{2x}}, q_{f_{2y}})$ bezeichnet.

φ ist in der Drehmatrix A' berücksichtigt und (x_{1x}, x_{1y}) und (x_{2x}, x_{2y}) bezeichnen die lokale Position der zwei gewählten Punkte im Rahmen

Also ergeben sich folgende Gleichungen aus den entsprechenden Zeilen von $B_r v = 0$:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} w_x \\ w_y \end{pmatrix} + w_\varphi \begin{pmatrix} A^{*'} & 0 \\ 0 & A^{*'} \end{pmatrix} \left(\begin{pmatrix} x_{1x} \\ x_{1y} \\ x_{2x} \\ x_{2y} \end{pmatrix} + \begin{pmatrix} q_{f_{1x}} \\ q_{f_{1y}} \\ q_{f_{2x}} \\ q_{f_{2y}} \end{pmatrix} \right) = 0$$

wobei

$$A^{*'} = \begin{pmatrix} -\sin(\varphi) & -\cos(\varphi) \\ \cos(\varphi) & -\sin(\varphi) \end{pmatrix}$$

die Ableitung der lokalen Rotationsmatrix ist.

Wir suchen nun $B_r v = 0$. Aus der ersten und zweiten bzw dritten und vierten Gleichung ergibt sich für (w_{1x}, w_{1y}) :

$$\begin{pmatrix} w_{1x} \\ w_{1y} \end{pmatrix} = -w_\varphi A^{*'} \begin{pmatrix} x_{1x} + q_{f_{1x}} \\ x_{1y} + q_{f_{1y}} \end{pmatrix} \quad (4.25)$$

$$\text{bzw.} \quad (4.26)$$

$$\begin{pmatrix} w_x \\ w_y \end{pmatrix} = -w_\varphi A^{*'} \begin{pmatrix} x_{2x} + q_{f_{2x}} \\ x_{2y} + q_{f_{2y}} \end{pmatrix} \quad (4.27)$$

Durch das Gleichsetzen der beiden Gleichungen ergibt sich:

$$\begin{aligned} w_\varphi A^{*'} \begin{pmatrix} x_{1x} + q_{f_{1x}} \\ x_{1y} + q_{f_{1y}} \end{pmatrix} &= w_\varphi A^{*'} \begin{pmatrix} x_x + q_{f_{2x}} \\ x_y + q_{f_{2y}} \end{pmatrix} \\ w_\varphi A^{*'} \begin{pmatrix} x_{1x} + q_{f_{1x}} - x_{2x} - q_{f_{2x}} \\ x_{1y} + q_{f_{1y}} - x_{2y} - q_{f_{2y}} \end{pmatrix} &= 0 \\ w_\varphi \begin{pmatrix} x_{1x} + q_{f_{1x}} - (x_{2x} + q_{f_{2x}}) \\ x_{1y} + q_{f_{1y}} - (x_{2y} + q_{f_{2y}}) \end{pmatrix} &= 0 \end{aligned}$$

Das heißt, falls $w_\varphi \neq 0$, gilt:

$$\begin{pmatrix} x_{1x} + q_{f_{1x}} - (x_{2x} + q_{f_{2x}}) \\ x_{1y} + q_{f_{1y}} - (x_{2y} + q_{f_{2y}}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Das bedeutet, dass die Punkte nach der Verformung die selben Koordinaten haben. Dies widerspricht aber der Voraussetzung, dass die Verschiebung q_f im modifizierten Newtonschritt zulässig ($\det \nabla \phi \geq 0$ ist). Somit muss $w_\varphi = 0$ gelten. Aus der Gleichung (4.25) folgt

dann auch, dass $w_x = 0$ und $w_y = 0$ gelten muss, und somit $v = 0$ im ersten Rahmen. Die obige Argumentation kann nun wegen der Kopplungsbedingungen für die weiteren Teilgebiete (induktiv) genauso abgehandelt werden. \square

Abschließend wird w_f berechnet:

$$w_f = K_{ff}^{*-1} (d_f - B_r^T w_\lambda)$$

4.8 FFRF-FETI-Algorithmus

Im Folgenden ist der in Abschnitt 4.7 hergeleitete Algorithmus zur numerischen Lösung der Gleichung (4.14) zusammengefasst:

```

 $w = 0, q_{alt} = 0, q_{neu} = 0, d = 0$ 
 $\lambda_{alt} = 0, \lambda_{neu} = 0$ 
Generiere das Netz
Berechne  $K_{ff}^{(i)}$  und faktorisiere  $K_{ff}^{*(i)}$  für jedes  $i = 1, \dots, N$ 
Berechne  $r_{alt} = d = \nabla_{q,\lambda} \mathcal{L}(q_{alt}, \lambda_{alt})$ 
while  $\|r_{alt}\| > eps$  do
   $\tau = 1$ 
  Berechne  $w_{\lambda_0} = B_r (B_r^T B_r)^{-1} d_r$ 
  Löse  $PF\widetilde{w}_\lambda = P (B_f K_{ff}^{*-1} d_f - d_\lambda - Fw_{\lambda_0})$  mit CG
  Berechne  $w_\lambda = \widetilde{w}_\lambda + w_{\lambda_0}$ 
  Löse  $(B_r^T B_r) w_r = Fw_\lambda - B_f K_{ff}^{*-1} d_f + d_\lambda$ 
  Berechne  $w_f = K_{ff}^{*-1} (d_f - B_r^T w_\lambda)$ 
  Berechne

```

$$\begin{pmatrix} q_{neu} \\ \lambda_{neu} \end{pmatrix} = \begin{pmatrix} q_{alt} \\ \lambda_{alt} \end{pmatrix} + w$$

```

Berechne  $r_{neu} = \nabla_{q,\lambda} \mathcal{L}(q_{neu}, \lambda_{neu})$ 
while  $\|r_{neu}\| > \|r_{alt}\|$  do
   $\tau = \frac{\tau}{2}$ 
  Berechne

```

$$\begin{pmatrix} q_{neu} \\ \lambda_{neu} \end{pmatrix} = \begin{pmatrix} q_{alt} \\ \lambda_{alt} \end{pmatrix} + \tau w$$

```

  Berechne  $r_{neu} = \nabla_{q,\lambda} \mathcal{L}(q_{neu}, \lambda_{neu})$ 
end while
 $r_{alt} = r_{neu}$ 
 $q_{alt} = q_{neu}$ 
end while

```

4.9 Vergleich Newton vs. FFRF-FETI

Wir diskutieren den Aufwand für das voll nichtlineare Problem, welches mit dem Newtonverfahren gelöst wird, und dem Verfahren mit FFRF unter Einbeziehung von FETI.

	voll nichtlineares Newtonverfahren	Newtonverfahren für FFRF unter Einbeziehung von FETI
Aufwand im Bezug auf die Assemblierung	Hier muss die Matrix K'_h in jedem Schritt des Newtonverfahrens neu assembliert werden (siehe Abschnitt 3.7.2). Die Größe hängt von der Anzahl der Knoten im Gebiet Ω ab. Ebenso muss die rechte Seite des Systems assembliert werden	Die blockdiagonale Matrix K_{ff}^* muss zu Beginn einmal assembliert werden. Dieses System ist größer, da die Knoten an den Kanten zweimal vorkommen. Dafür ändert sich diese nicht mehr. Die Matrix B_r muss in jedem Newtonschritt neu erstellt werden. Wie beim Newtonverfahren muss die rechte Seite des Systems erstellt werden
Aufwand in Bezug auf die Faktorisierung	Zum Lösen der Gleichung für das Newtonupdate muss die Matrix K'_h in jedem Newtonschritt neu faktorisiert werden.	Die Faktorisierung der Matrix K_{ff}^* kann in jedem der Newtonschritt verwendet werden. Es muss allerdings die dünnbesetzte Matrix $B_r^T B_r$ in jedem Newtonschritt wieder neu faktorisiert werden. Die Größe von $B_r^T B_r$ ist aber nur proportional zur Anzahl der Unterteilungen des Gebiets Ω .
Gesamtaufwand	Der Gesamtaufwand hängt im wesentlichen von der Anzahl der benötigten Newtonschritte ab und vom Aufwand pro Newtonschritt	Der Aufwand hängt natürlich auch von den Newtonschritten ab. Zusätzlich muss in jedem Newtonschritt auch noch die Gleichung für $\tilde{\omega}_\lambda$ mittels CG Verfahren gelöst werden.

4.10 Implementierung

Im Rahmen der Masterarbeit wurden die Lösungsalgorithmen in C++ implementiert. Bei der Implementierung wurde auf bereits bestehende Strukturen des Instituts Numerische Mathematik für die Behandlung von FETI Methoden zurückgegriffen. Dabei wurde im speziellen die Generierung des Mesh verwendet bzw. auf das Implementieren der Kopplungen zurückgegriffen. Weiters wurden bestehende Algorithmen zum Implementieren bzw. Faktorisieren von dünnbesetzten Matrizen verwendet. Auch für das CG-Verfahren wurde auf vorgefertigte Routinen zurückgegriffen.

Es wurde beim Arbeiten auf diverse Löser und auf Routinen zur Implementierung von Vektoren im FFRF Bereich zurückgegriffen. Zuerst wurde als Einführung ein Lösungsverfahren für das linearisierte Problem implementiert. Weiters wurde der Code für das Newtonverfahren für das nichtlineare Problem ausprogrammiert. Dies dient auch als Referenz für das Ergebnis des Gebietszerlegungsverfahrens.

Des weiteren ist der Code für das Lösen des nichtlinearen Problems mit dem FFRF-FETI Ansatz implementiert worden. Es wurde versucht, dass beim Programmieren im wesent-

lichen den Schritten in der Masterarbeit gefolgt wurde.

Um eine kurze Übersicht über die Implementierung des Algorithmuses zu geben, wird das Headerfile der Implementierung des FFRF FETI Algorithmuses angefügt. Es wurde versucht, die Schritte aus der Diplomarbeit möglichst ähnlich im Programm auszuführen.

```

1 #ifndef __NewtonFFRF_HH
2 #define __NewtonFFRF_HH
3 /*!
4  \file      NewtonFFRF.hh
5  \brief     Calculate a Newton step for a domain
6  \brief     decomposition method for nonlinear elasticity
7  \date      March 2012
8  \author    Klemens Reindl & Clemens Pechstein
9 */
10
11 #include "utilities.hh"
12 #include "basiclinalg.hh"
13 #include "mesh.hh"
14 #include "subdomains.hh"
15 #include "linalg.hh"
16 #include "fem.hh"
17 #include "composegmv.hh"
18 #include "fetilagrange.hh"
19
20 namespace feti
21 {
22     using namespace std;
23     using namespace blalg;
24     using namespace msh;
25     using namespace parallel;
26     using namespace linalg;
27     using namespace fem;
28
29     class NewtonFFRF
30     {
31     public:
32         /// Default constructor
33         NewtonFFRF();
34
35         /**
36          * constructor
37          * @param q-phi vector of the angels of the rotating frames
38          * @param q-t vector of the translation of the frames
39          * @param q-f vector the local displacement in the frame
40          * @param f the load vector
41          * @param x the local position of the the knots in the frame
42          * @param lambda the lagrange multipliers
43          * @param Kff the matrix Kff
44          * @param Kffinv the inverse of the matrix Kff
45          */
46         NewtonFFRF (const Vector<>& q-phi, const Vector<Vec<2>> & q-t,
47                   const FETISplitField<>& q-f, const Vector<Vec<3>> & d-D,
48                   const FETISplitField<>& x, const FETISplitField<>& f,
49                   const FETILagrange& lagrange, const PVector<>& lambda,
50                   const Array< DirectInverse<>> & KffInv,
51                   const Array< SparseMatrix<>> & Kff);

```

```

52
53 // destructor
54 ~NewtonFFRF ();
55
56 // calculates the newton updates with the righten side d_phi,
57 // d_t, d_f, d_lambda and stores the result in w_phi, w_t, w_f, mu
58 void calcNewtonUpdate(Vector<& w_phi, Vector<Vec<2>> & w_t,
59   FETISplitField<& w_f, Vector<Vec<3>> & w_D, PVector<& mu,
60   const Vector<& d_phi, const Vector<Vec<2>> & d_t,
61   const FETISplitField<& d_f, const Vector<Vec<3>> & d_D,
62   const PVector<& d_lambda ) const;
63
64 // calculates the gradient of the Lagrange functional
65 void calcLagrangeGradient (const Vector<Vec<2>> & q_t,
66   const Vector<& q_phi, const FETISplitField<& q_f,
67   const Vector<Vec<3>> & q_D, const PVector<& q_lambda,
68   Vector <Vec<2>> & d_t, Vector <& d_phi, FETISplitField <& d_f,
69   Vector <Vec<3>> & d_D, PVector <& d_lambda) const;
70
71 // calculates the global displacment u from
72 // the local displacment q_f,
73 // the rotation q_phi and the translation q_t
74 void calcglobalu (const FETISplitField<& q_f,
75   const Vector<Vec<2>> & q_t, const Vector<& q_phi,
76   FETISplitField<& u ) const;
77
78 // calculates the norm of the vector d_t, d_phi, d_f, d_D, d_lambda
79 double calcNorm (Vector <Vec<2>> & d_t, Vector <& d_phi,
80   FETISplitField<& d_f, Vector <Vec<3>> & d_D,
81   PVector<& d_lambda) const;
82
83 //private:
84 // applies the matrix B_f to an vector v of length
85 // of the total number of nodes and stores mu
86 void applyB_f(const FETISplitField<& v, PVector<& mu) const;
87
88 // applies the matrix B_f transpose to a vector of length of
89 // the Langrange multipliers and stores the result in v
90 void applyB_fTrans(const PVector<& mu, FETISplitField<& v) const;
91
92 // applies the matrix B_r, v is of length 3 N
93 void applyB_r(const Vector<& v, PVector<& mu) const;
94
95 // applies the matrix B_r transpose, v is of length 3 N
96 void applyB_rTrans(const PVector<& mu, Vector<& v) const;
97
98 // applies the matrix F=B_f KKin v B_f^T to an vector x and stores
99 // the result in y
100 void applyF(const PVector<& x, PVector<& y) const;
101
102 // applies the the projection P to the vector x
103 // and stores the result in y
104 void applyP(const PVector<& x, PVector<& y) const;
105
106 inline
107 void applyPF(const PVector<& x, PVector<& z) const

```

```

108 | {
109 |   PVector<> y;
110 |   lagrange_.initLagrangeVector (y, NON_PARALLEL);
111 |
112 |   applyF (x, y);
113 |   /*
114 |   PVector<> xx;
115 |   lagrange_.initLagrangeVector (xx, NON_PARALLEL);
116 |   applyP (x, xx);
117 |   applyF (xx, y);
118 |   */
119 |   applyP (y, z);
120 | }
121 |
122 | /// solves the System PF(P) mu = y
123 | void solvePFP(const PVector<& y, PVector <& mu) const;
124 |
125 | /// calculates the matrix R
126 | void calcR ();
127 |
128 | /// applies the matrix R[i] to x and stores the result in y[i]
129 | void applyRi (const int i, const Vector<& x,
130 |   FETISplitField<& y) const;
131 |
132 | /// applies the matrix R transpose to y
133 | /// and stores the result in x
134 | void applyRTrans (const FETISplitField<& y,
135 |   Vector<& x ) const;
136 |
137 | /// calculates the matrixproduct of B_r B_r^T
138 | /// B_r=[B_t|A*'(x+q_f)] the result is stored at
139 | /// the matrix BrTBr_
140 | void calcBrTBr ();
141 |
142 | /// calculates the inverse of the matrixproduct auf B_r B_r^T
143 | /// the result is stored at the matrix BrTBrInv_
144 | void calcBrTBrInv ();
145 |
146 | /// get number of Lagrange multipliers
147 | const int numLagrange() const
148 |   { return lagrange_.numLagrange(); }
149 |
150 | /// calculates the vector mu0
151 | void calcMu0 (const Vector<& d_r, PVector <& mu0) const;
152 |
153 | /// calculates the vector muTilde by solving the system
154 | /// PF muTilde=P(B_f K-f^-1 d_f-d_mu-Fmu0)
155 | void calcMuTilde (const FETISplitField<& d_f,
156 |   const Vector<Vec<3>> & d_D, const PVector<& d_lambda,
157 |   const PVector <& mu0, PVector <& muTilde) const;
158 |
159 | /// applies the matrix Kff[i] to the vektor d_f[i]
160 | /// and d_D[i] for all i
161 | /// and stores the result in w_f[i] and w_D[i]
162 | void applyKff (const FETISplitField<& d_f,
163 |   const Vector<Vec<3>> & d_D, FETISplitField<& w_f,

```

```

164     Vector<Vec<3> >& w_D) const;
165
166     /// applies the matrix KffInv[i] to the vektor d_f[i] and d_D[i]
167     /// for all i and stores the result in w_f[i] and w_D[i]
168     void applyKffInv (const FETISplitField<>& d_f,
169         const Vector<Vec<3> >& d_D, FETISplitField<>& w_f,
170         Vector<Vec<3> >& w_D) const;
171
172     /// calculates the update for
173     /// w_r = (Br^T B_r)^-1 B_r^T (F mu - B_f K_ff^-1 d_f - d_mu)
174     void calcw_r (const FETISplitField<>& d_f,
175         const Vector<Vec<3> >& d_D, const PVector <>& mu,
176         const PVector <>& d_mu, Vector<> & w_r) const;
177
178     /// calculates the update for w_f = Kff^-1 (d_f - B^T mu)
179     void calcw_f (const FETISplitField<>& d_f,
180         const Vector<Vec<3> >& d_D, const PVector <>& mu,
181         FETISplitField <> & w_f, Vector<Vec<3> >& w_D) const;
182
183     /// applies the matrix A to the Vector Vecin
184     /// and stores the result in Vecout
185     void applyA (const FETISplitField<> & Vecin, const Vector<>& q_phi,
186         FETISplitField<> & Vecout) const;
187
188     /// applies the matrix A transpose to the Vector Vecin
189     /// and stores the result in Vecout
190     void applyATrans (const FETISplitField<> & Vecin,
191         const Vector<>& q_phi, FETISplitField<> & Vecout) const;
192
193 private:
194     int N_; ///< number of frames
195     Vector<> q_phi_; ///< Vector of the rotating angles
196     Vector<Vec<2> > q_t_; ///< Vector of the translation of the frames
197     FETISplitField<> q_f_; ///< local displacement in the frames
198     Vector<Vec<3> > q_D_; ///< Vector of the local fixed nodes
199     FETISplitField<> x_; ///< relative position of the knot in the frame
200     FETISplitField<> f_; ///< Load vector
201     const FETILagrange& lagrange_;
202         ///< Lagrange handler
203     PVector<> lambda_; ///< the lagrange multipliers
204     const Array<SparseMatrix<> >& Kff_;
205         ///< The local stiffness matrices of the frames
206     const Array<DirectInverse<> >& KffInv_;
207         ///< The local stiffness matrices of the frames
208     Array<Matrix<> > R_; ///< the matrices R_i (B_r = B R)
209     Matrix<> Ri; ///< Test
210     Matrix<> BrTBr_; ///< matrix Br^T Br
211     Matrix<> BrTBrInv_; ///< inverse of Br^T Br
212
213 }; /// class NewtonFFRF
214 /// namespace feti
215
216 #endif /// _NewtonFFRF_HH

```

5 Numerische Ergebnisse

In diesem Kapitel untersuchen und vergleichen wir die Ergebnisse der unterschiedlichen Methoden. Natürlich ergibt sich durch die getroffenen Vereinfachungen des lokalen Verzerrungstensors ein Modellfehler. Wir betrachten den im FFRF-FETI Newtonverfahren entstandenen Fehler im Vergleich zum vollnichtlinearen Newtonverfahren. Um zu zeigen, dass das lineare Modell bei zu großer Kraft bzw zu großer Verformung kein gutes Modell ist, wird auch das Ergebnis des linearen Modells gezeigt.

Weiters betrachten wir in diesem Kapitel die Iterationszahlen des Newtonverfahrens bzw. die Iterationsschritte für das FFRF-FETI Newtonverfahren für ein konkretes Beispiel. Wir betrachten auch die Konvergenz des CG- Verfahrens innerhalb des FFRF-FETI Verfahrens.

5.1 Untersuchen des Modellfehlers

Für die Simulation wurden für die Werkstoffparameter die Werte von Stahl verwendet. Es wird der Modellfehler zwischen dem vollnichtlinearen Problem, dem FFRF-FETI Problem und dem linearisierten Problem anhand von zwei Beispielen mit den Dimensionen von einerseits $[0, 7] \times [0, 1]$ und andererseits $[0, 25] \times [0, 1]$ verglichen.

Beispiel 5.1. *E-modul $\tilde{E} = 2,1 \cdot 10^{11} \frac{N}{m^2}$*

Poisson ratio $\nu = 0,3$

Kraft auf Neumann Rand $g_N = -10^9 N$

Volumskräfte $f = 0$

$\Omega = [0, 7] \times [0, 1]$

Mesh: 141 Knoten in x Richtung, 21 Knoten in y Richtung

Es werden einige Bilder gezeigt, die nach der Anwendung der Deformation durch die numerische Berechnung entstanden sind. Die Abbildung 2 zeigt den nach dem vollnichtlinearen Modell deformierten Balken. In Abbildung 3 wird ein deformierter Balken gezeigt, welcher mit dem linearisierten Modell berechnet wurde. Die Abbildung 4 zeigt die mit dem FFRF-FETI Modell berechnete Deformation.

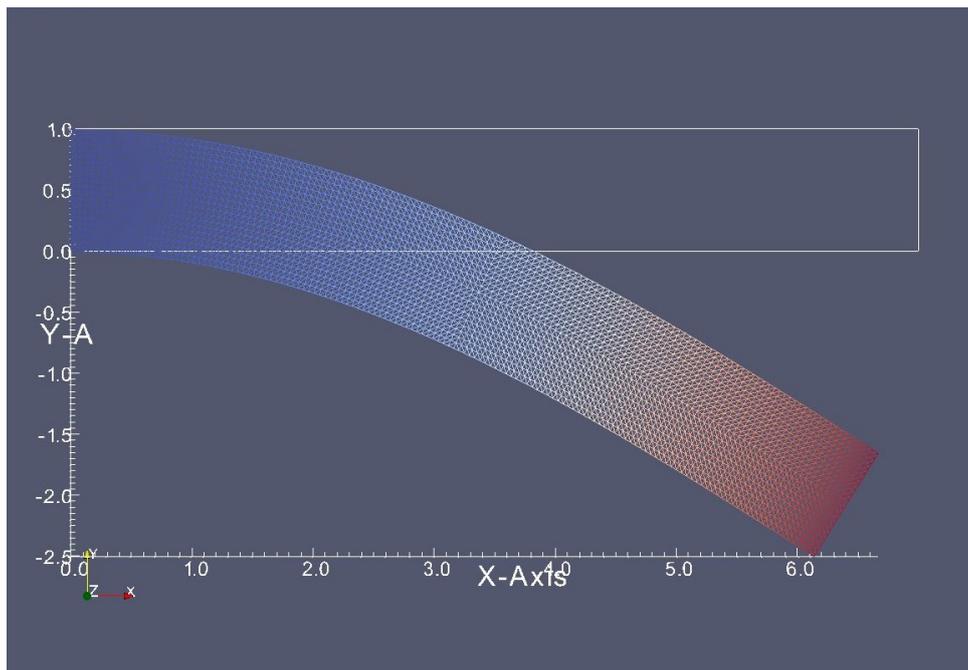


Abbildung 2: Ergebnis der Berechnung für das Beispiel 5.1 mit dem vollnichtlinearen Newtonverfahren

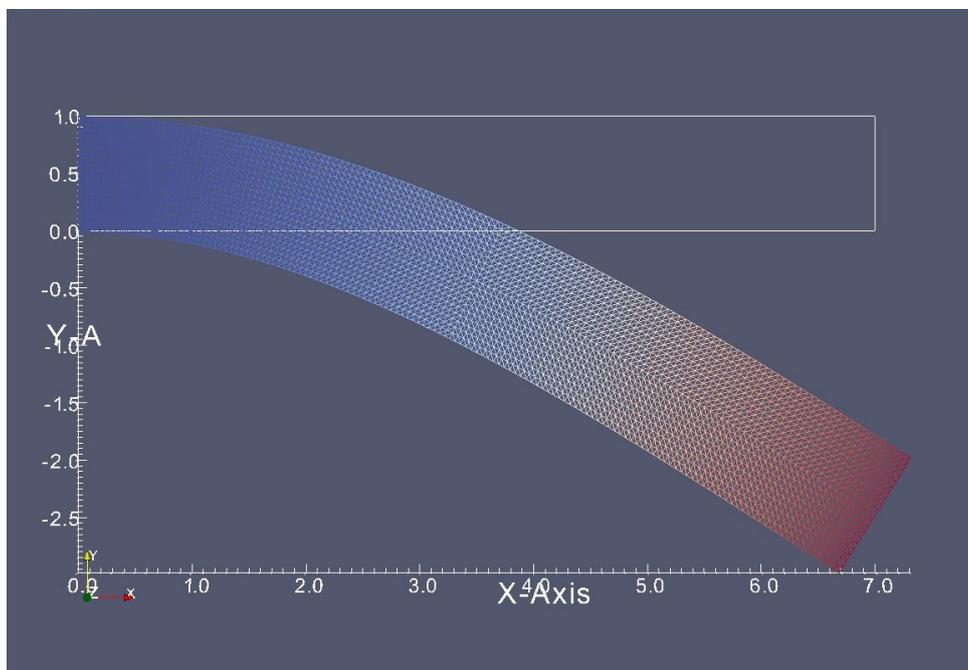


Abbildung 3: Lösung des linearisierten Problems für das Beispiel 5.1

Beim Vergleich zwischen Abbildung 2 und Abbildung 3 ist der Fehler offensichtlich groß. Man sieht augenscheinlich an der x -Skala (rechte untere Ecke auf ca. 6,5 statt auf

ca. 6,1 wie im vollnichtlinearen Modell) bzw. an der Breite des Balkens, dass die Lösung des linearisierten Systems Abbildung 3 nicht mit der Lösung des nichtlinearen Problems übereinstimmt.

In Abbildung 4 wird das entsprechende Ergebnis mit dem FFRF-FETI Verfahren dargestellt. Lediglich das Mesh ändert sich entsprechend, da der Balken in 7 Frames unterteilt wird und die einzelnen Frames in je 21 Knoten in x Richtung bzw. in y Richtung unterteilt werden.

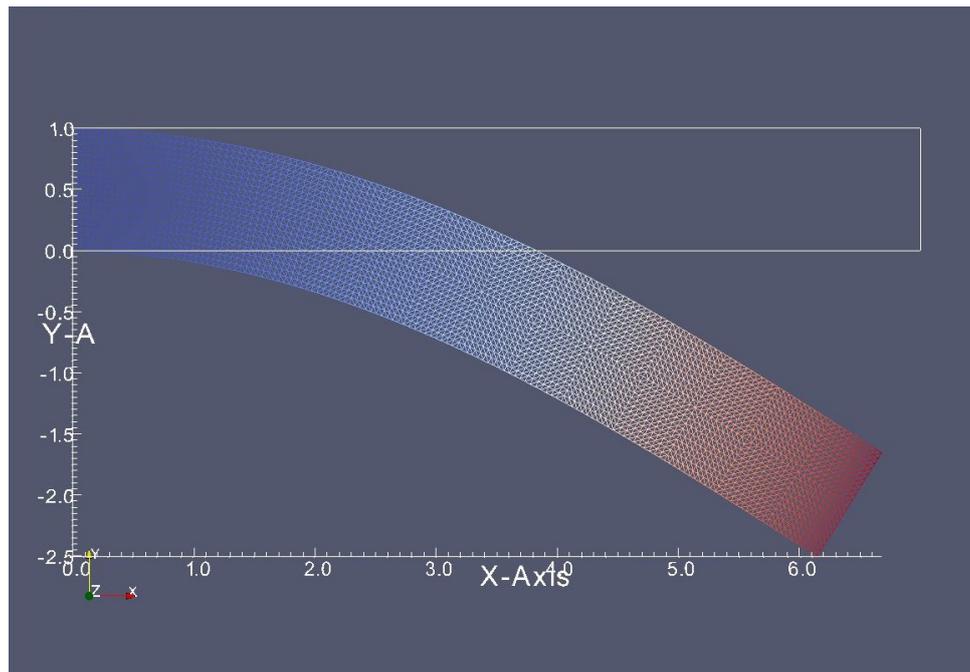


Abbildung 4: Ergebnis der Berechnung für das Beispiel 5.1 mit dem FFRF-FETI Newtonverfahren

Augenscheinlich lässt sich kein Fehler zwischen der Abbildung 2 und der Abbildung 4 feststellen, allerdings lässt sich der Fehler rechnerisch erfassen. Der relative Fehler in der l_2 Norm zwischen der Lösung mit dem Newton Verfahren bzw. mit dem FFRF FETI Verfahren liegt in diesem Beispiel bei $6,4 \cdot 10^{-3}$, die genaue Fehlerverteilung ist in Abbildung 5 dargestellt.

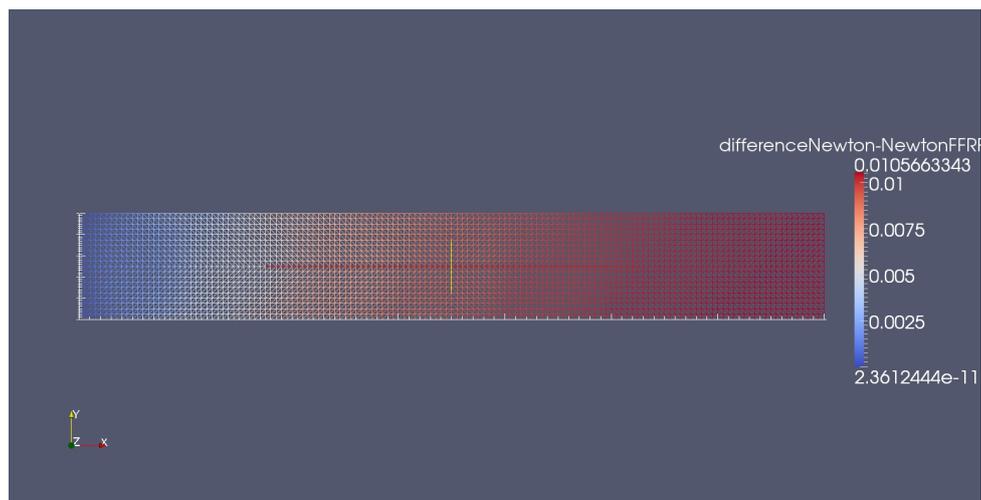


Abbildung 5: Fehlerverteilung zwischen dem vollnichtlinearen und dem FFRF-FETI Newton Verfahren

Der Ansatz die Verschiebung in eine Rotation und Translation bzw. in eine lokale Verschiebung zu unterteilen, kann hier sehr gut veranschaulicht werden. Diese Lösung mit FFRF-FETI Verfahren Abbildung 6 setzt sich aus der flexiblen Verschiebung Abbildung 7 und der Verschiebung durch die Translation bzw Drehung der Rahmen Abbildung 8 zusammen. Die verschiedenen Farben zeigen die einzelnen Frames.

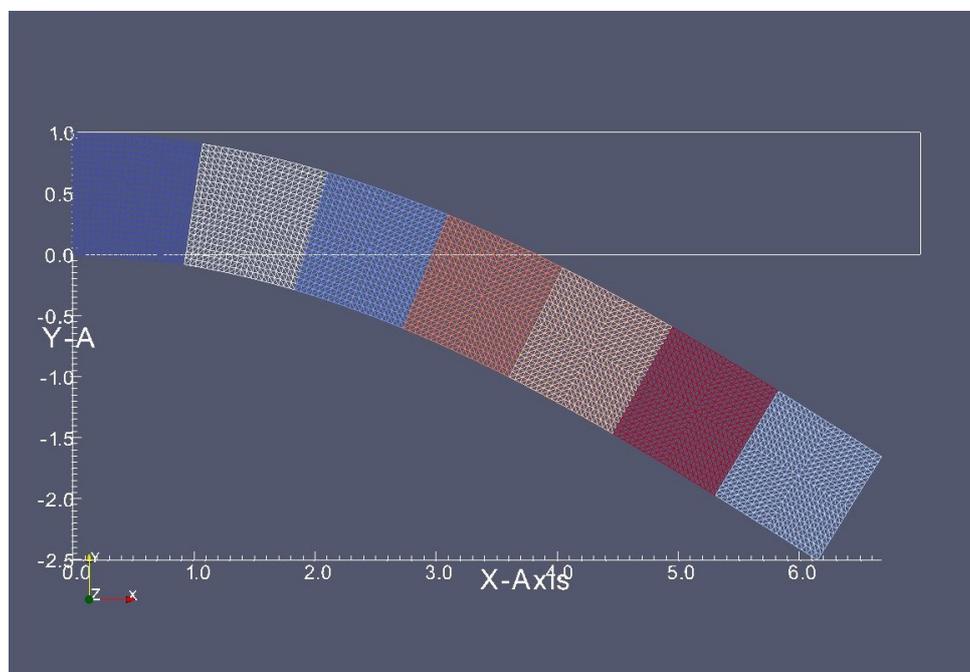


Abbildung 6: Deformation mit FFRF-FETI Verfahren für das Beispiel 5.1 mit den unterschiedlich eingefärbten Frames

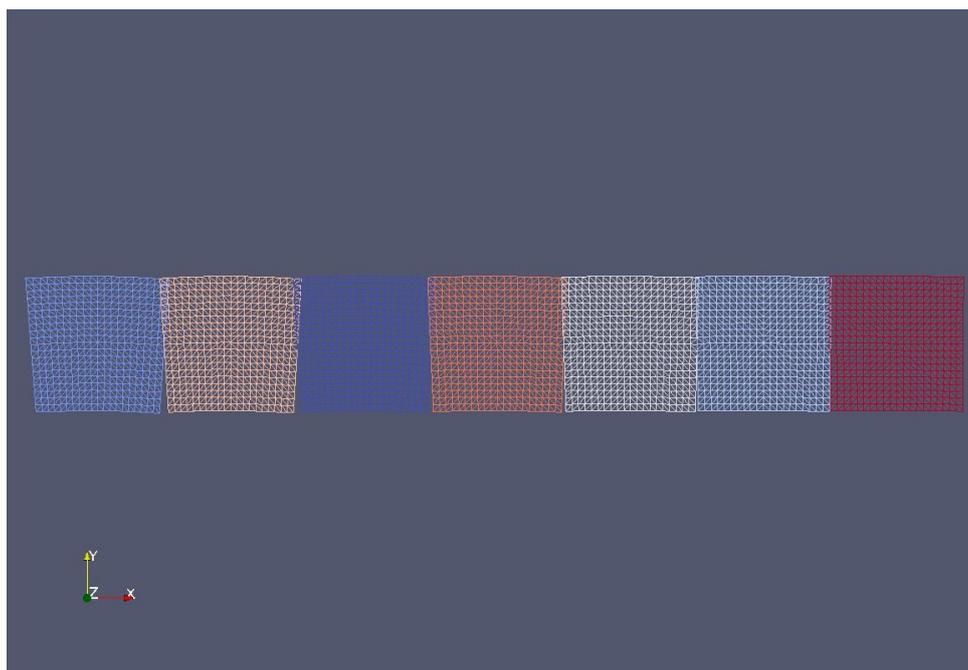


Abbildung 7: lokaler Anteil der gesamten Verschiebung für das Beispiel 5.1

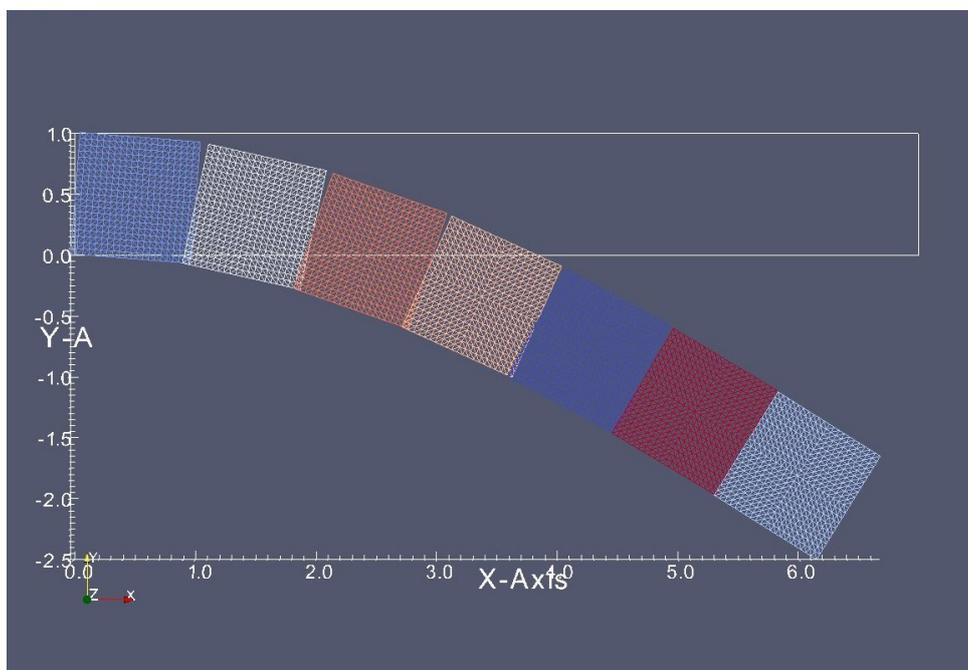


Abbildung 8: Translation und Drehung der Rahmen für das Beispiel 5.1

Es wurde weiters ein Balken mit den Maßen 25×1 gerechnet (Beispiel 5.2). Dazu musste die Kraft entsprechend verändert werden.

Auch hier werden wieder die Ergebnisse des FFRF-FETI Verfahrens, dem vollnichtlinearen Newtonverfahren und dem linearisierten Verfahren verglichen:

Beispiel 5.2. E -modul $\tilde{E} = 2,1 \cdot 10^{11} \frac{N}{m^2}$

Poisson ratio $\nu = 0,3$

Kraft auf Neumann Rand $g_N = -10^8 N$

Volumskräfte $f = 0$

$\Omega = [0, 25] \times [0, 1]$

Mesh: 201 Knoten in x Richtung, 9 Knoten in y Richtung

bzw.

Mesh beim FFRF-FETI: 25 Frames in x Richtung, je Frame 9 Knoten in x Richtung und 9 Knoten in y Richtung

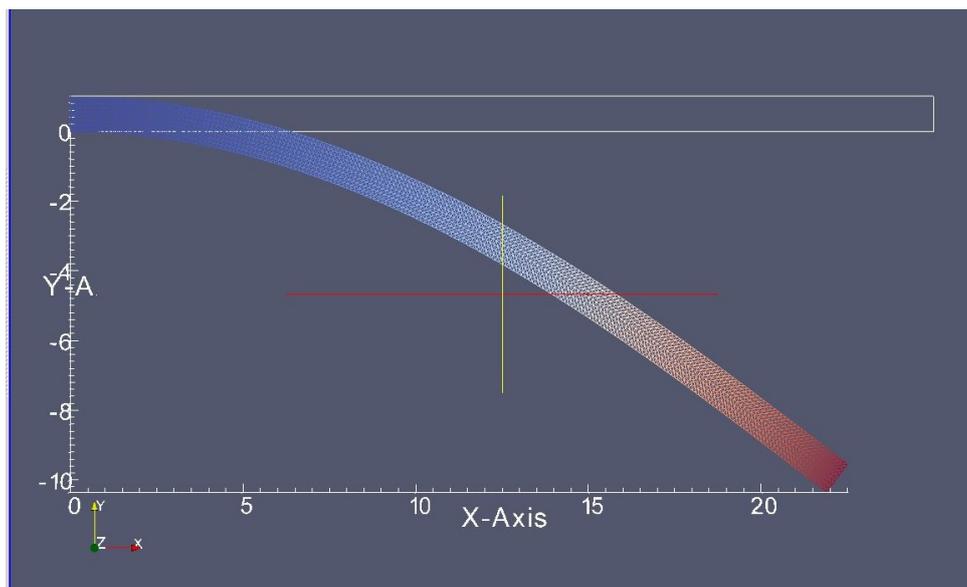


Abbildung 9: Ergebnis der Berechnung für das Beispiel 5.2 mit dem vollnichtlinearen Newtonverfahren

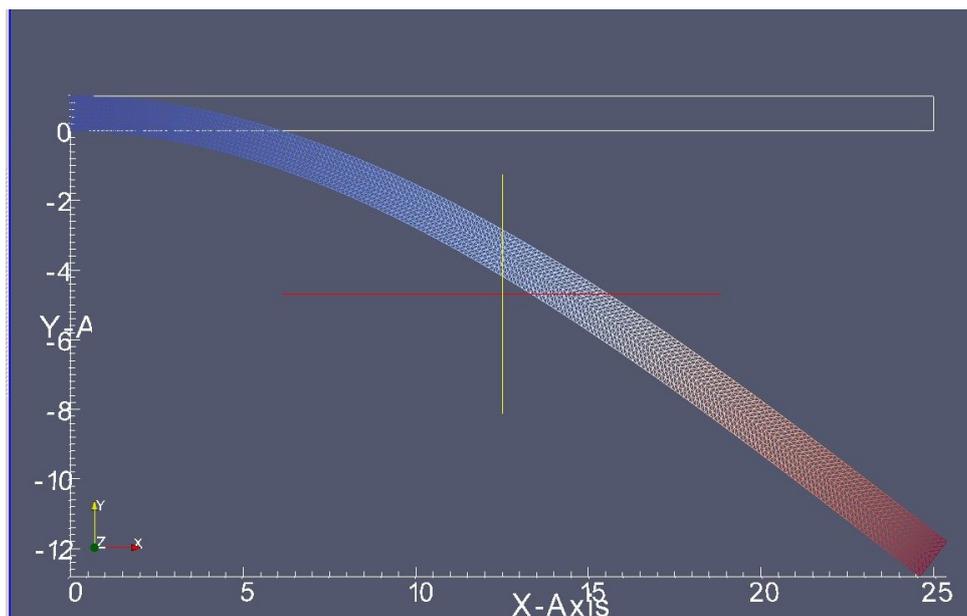


Abbildung 10: Lösung des linearisierten Problems für das Beispiel 5.2

Man sieht wie in Beispiel 5.1 augenscheinlich an der x-Skala (rechte untere Ecke auf ca. 24 statt auf ca. 23 wie im vollnichtlinearen Modell) und auch auf der y-Skala (rechte untere Ecke auf ca. -12 statt auf ca. -10 wie im vollnichtlinearen Modell) bzw. an der Breite des Balkens, dass die Lösung des linearisierten Systems Abbildung 10 nicht mit der Lösung des nichtlinearen Problems übereinstimmt.

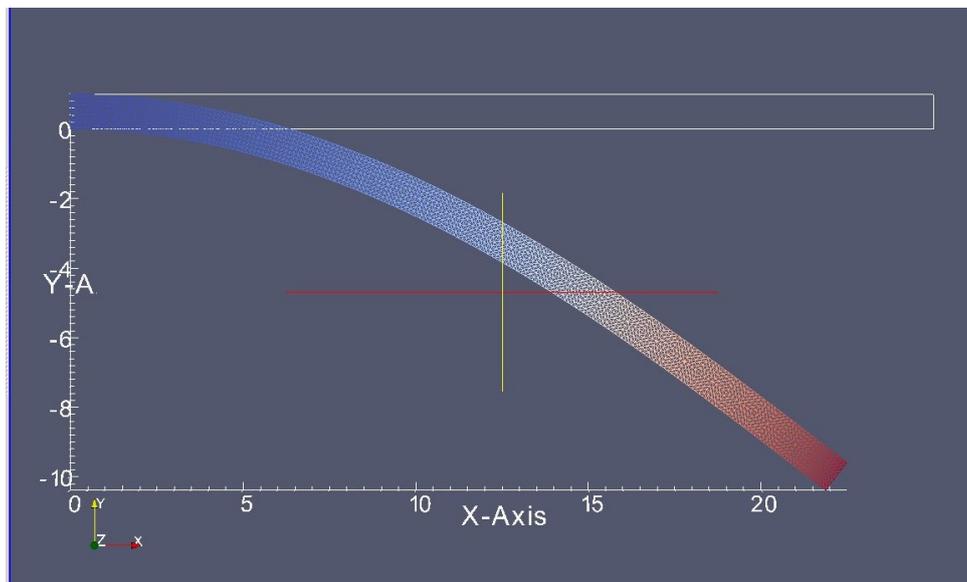


Abbildung 11: Ergebnis der Berechnung für das Beispiel 5.2 mit dem FFRF-FETI Newtonverfahren

Augenscheinlich lässt sich kein Fehler zwischen der Abbildung 9 und der Abbildung 11

feststellen. Der relative Fehler in der l_2 Norm zwischen der Lösung mit dem Newton Verfahren bzw. mit dem FFRF FETI Verfahren liegt in diesem Beispiel bei $6,9 \cdot 10^{-3}$.

5.2 Konvergenzvergleich

5.2.1 Anzahl der Newton-Iterationsschritte

Es zeigt sich in der praktischen Anwendung, dass das Newton-Verfahren bei diesen Beispielen 5.1 und 5.2 aus Abschnitt 5.1 mit dem FFRF-FETI Ansatz wesentlich schneller konvergiert. Die Anzahl der Schritte mit dem voll nichtlinearen Newton Verfahren steigt mit Feinheit der Unterteilung. Die Anzahl der Iterationsschritte mit dem FFRF-FETI Ansatz bleibt annähernd gleich. Dieses Verhalten wird nun im Folgenden detailliert dargestellt.

Beispiel 5.3. Die Problem-Parameter sind wie in Beispiel 5.1 gewählt

$$\Omega = [0, 7] \times [0, 1]$$

Das Mesh wird von 2 Unterteilungen pro Einheit auf 20 Unterteilungen pro Einheit verfeinert.

Mesh: von 15 bis 141 Knoten in x Richtung, von 3 bis 21 Knoten in y Richtung bzw.

Bei FFRF-FETI gilt für das Mesh: von 3 bis 21 Knoten in x Richtung, von 3 bis 21 Knoten in y Richtung pro Einheit je Frame

Startvektor für das jeweilige Newtonverfahren ist der Nullvektor.

Auf der x -Achse der folgenden Grafiken wird immer die Feinheit der Unterteilung pro Einheit aufgetragen. D.h. für jede Einheit erhalten wir bei einer äquidistanten Unterteilung der x Richtung in 20 gleich große Abstände und in y Richtung in 20 äquidistante Abstände pro Einheit 21×21 Knoten. Insgesamt erhalten wir damit dann $441 \times 7 = 3087$ Knoten in dem Gebiet Ω . Also ergeben sich 6174 Unbekannte bei der feinsten Unterteilung.

In Abbildung 12 wird der Verlauf der Anzahl der Newtonschritte für das Newtonverfahren ohne FFRF-FETI Ansatz und in Abbildung 13 wird die Anzahl der Schritte für das Newtonverfahren mit FFRF-FETI Ansatz gezeigt.

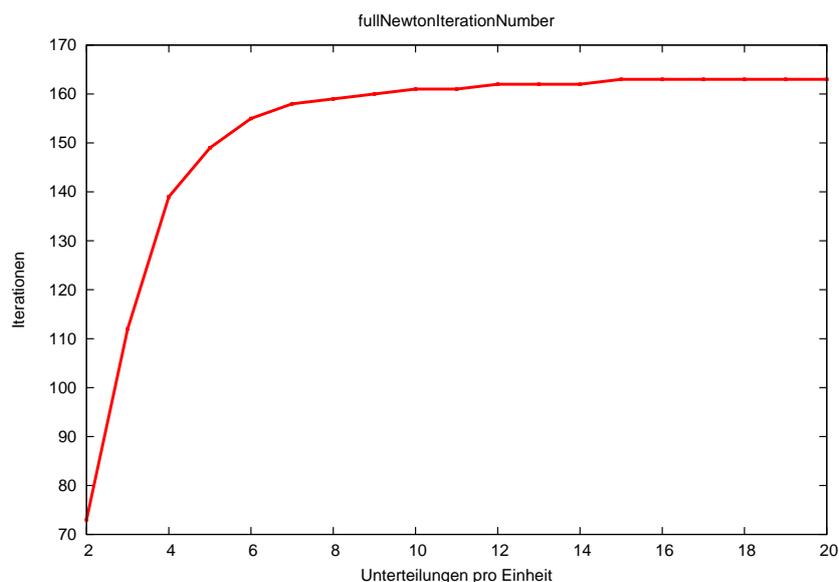


Abbildung 12: Anzahl der Iterationen mit dem vollnichtlinearen Newtonverfahren

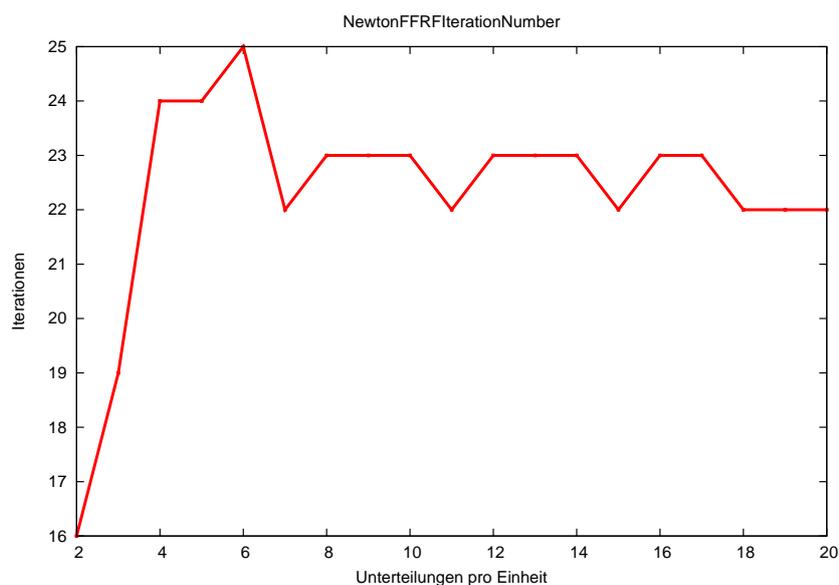


Abbildung 13: Anzahl der Iterationen mit dem Newtonverfahren mit FFRF-FETI Ansatz

Hier sieht man, dass in diesem Beispiel die Anzahl der Newtonschritte im vollnichtlinearen Newtonverfahren mit der Anzahl der Unterteilungen stark steigt, wobei aber beim FFRF-FETI Verfahren die Anzahl der Iterationsschritte annähernd gleich bleibt.

Die obige Konvergenzuntersuchung wird nun für das Beispiel 5.2 aus Abschnitt 5.1 durchgeführt.

Beispiel 5.4. Die Parameter sind wie in Beispiel 5.2

$$\Omega = [0, 25] \times [0, 1]$$

Das Mesh wird von 2 Unterteilungen pro Einheit auf 8 Unterteilungen pro Einheit verfeinert.

Mesh: von 15 bis 201 Knoten in x Richtung, von 3 bis 9 Knoten in y Richtung bzw.

Mesh: von 3 bis 9 Knoten in x Richtung, von 3 bis 9 Knoten in y Richtung je Frame
Startvektor für das jeweilige Newtonverfahren ist der Nullvektor.

In Abbildung 14 sieht man den Verlauf der Anzahl der Newtonschritte für das vollnichtlineare Newtonverfahren, und in Abbildung 15 wird die Anzahl der Schritte für das Newtonverfahren mit FFRF-FETI Ansatz gezeigt.

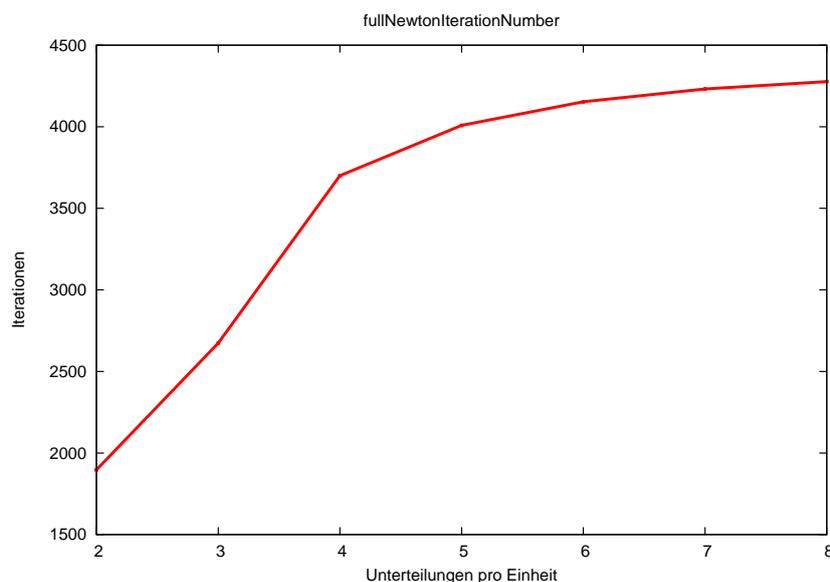


Abbildung 14: Anzahl der Iterationen mit dem vollnichtlinearen Newtonverfahren

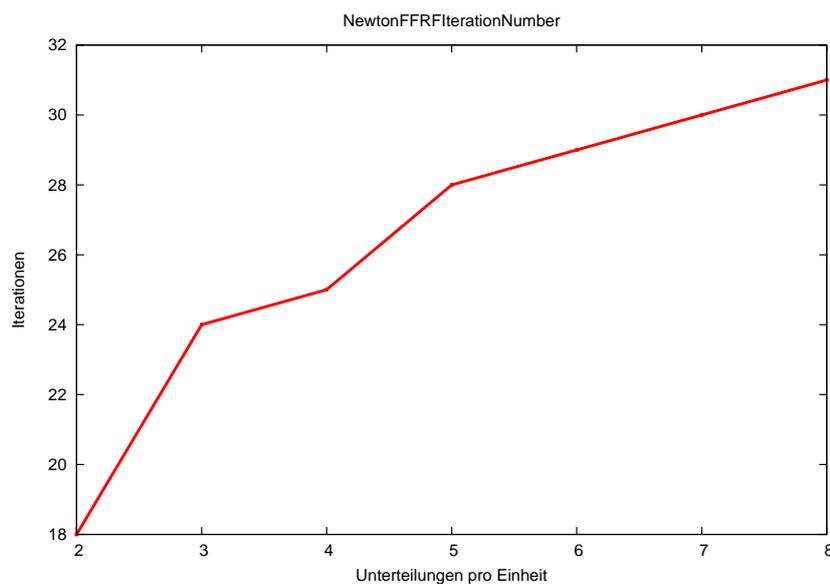


Abbildung 15: Anzahl der Iterationen mit dem Newtonverfahren mit FFRF-FETI Ansatz

Hier sieht man, dass die Anzahl der Newtonschritte im vollnichtlinearen Newtonverfahren mit der Anzahl der Unterteilungen sehr stark steigt, wobei aber beim FFRF-FETI Verfahren die Anzahl der Iterationsschritte nur sehr wenig steigt. Man beachte auch den großen Unterschied in der Anzahl der Iterationen (bis 32 Iterationen beim FFRF-FETI Verfahren im Vergleich zum vollnichtlinearen Verfahren mit bis zu 4200 Iterationen). Es wurde nach 8 Unterteilungen abgebrochen, da beim vollnichtlinearen Problem mit der vorliegenden Implementierung auf einem Laptop mit 2000 MHz und 1 GB Arbeitsspeicher bis zur Konvergenz fast 2 Stunden benötigt wurden.

Dazu zeigen wir in [Abbildung 16](#) und in [Abbildung 17](#) auch die benötigte Rechenzeit für das [Beispiel 5.4](#) in Abhängigkeit der Feinheit der Unterteilung:

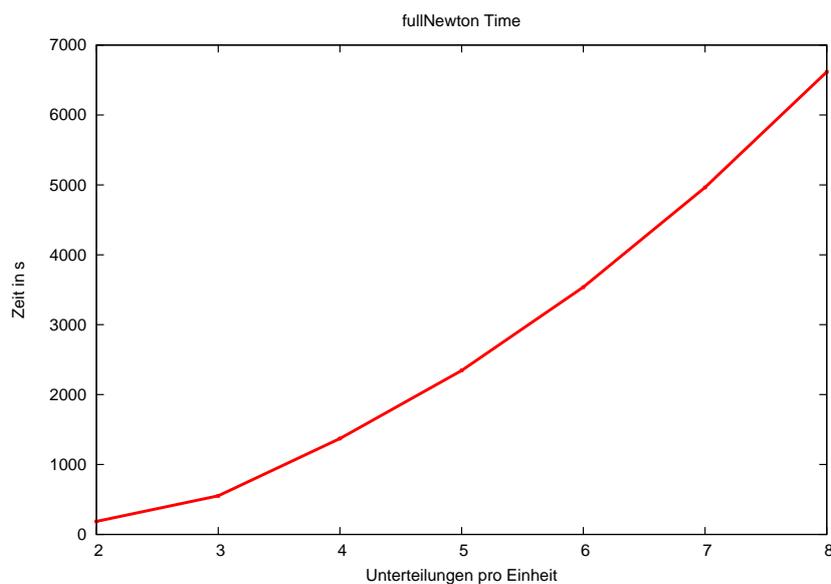


Abbildung 16: Die Zeit bis zur Konvergenz in Abhängigkeit der Feinheit der Unterteilung für das vollnichtlineare Problem für das Beispiel 5.4

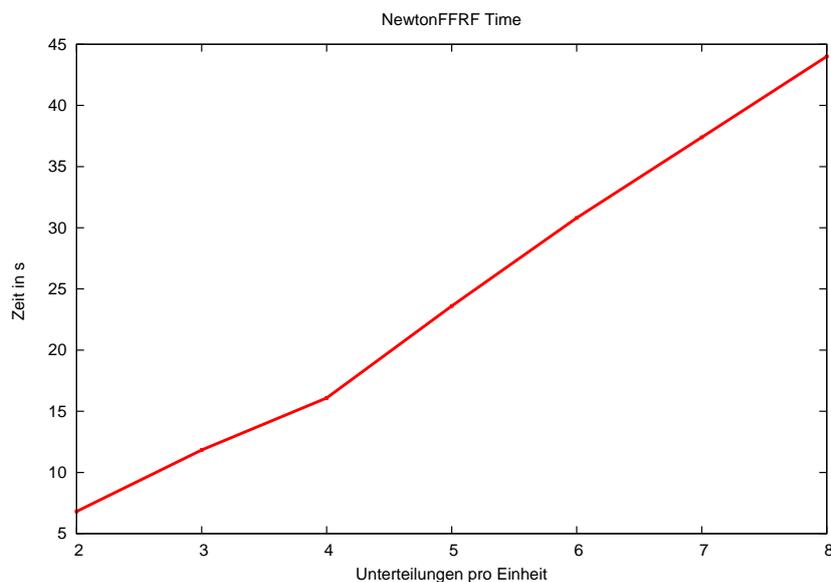


Abbildung 17: Zeitdauer bis zur Konvergenz in Abhängigkeit der Feinheit der Unterteilung für das FFRF-FETI Verfahren für das Beispiel 5.4

Man sieht, dass bei dem Beispiel 5.4 das FFRF-FETI Verfahren sehr viel schneller durchgeführt wurde als bei dem entsprechenden vollnichtlinearen Problem.

Aus diesen Ergebnissen lässt sich natürlich nicht automatisch auf das Verhalten der Iterationsschritte in anderen Fällen schließen. In den Beispielen 5.3 und Beispiel 5.4 ist

der Vorteil der FFRF-FETI Methode im Vergleich zum vollnichtlinearen Problem jedoch deutlich zu sehen.

5.2.2 Konvergenz des CG-Verfahrens

In diesem Abschnitt soll das Verhalten der Konditionszahl der Matrix PF in Abhängigkeit der Feinheit des Mesh pro Frame untersucht werden. Bei klassischen FETI Verfahren (siehe etwa [6]), verhält sich die Konditionszahl wie $\mathcal{O}\left(\frac{H}{h}\right)$, wobei H den Teilgebietsdurchmesser und h die Feinheit des Mesh bezeichnet. ($\frac{H}{h}$ ist proportional zu Anzahl der Unterteilungen pro Frame in jede Richtung).

In den Abbildungen 18 und 19 wird das Verhalten der Konditionszahl von PF gegenüber der Anzahl der Unterteilungen pro Einheit $\frac{H}{h}$ für das Beispiel 5.3 und 5.4 dargestellt. In beiden Graphen sieht man klar ein lineares Verhalten, so wie beim klassischen FETI Algorithmus.

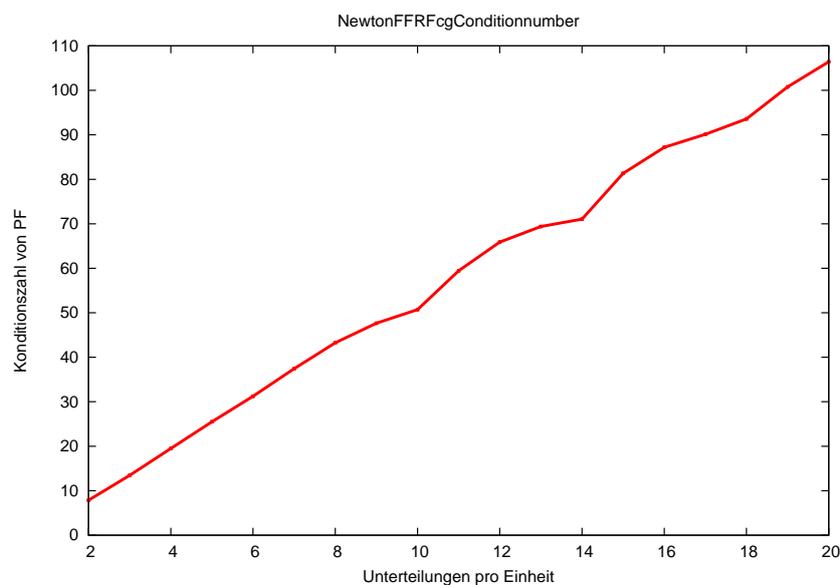


Abbildung 18: Konditionszahl im Beispiel 5.3 der Matrix PF mit steigender Feinheit

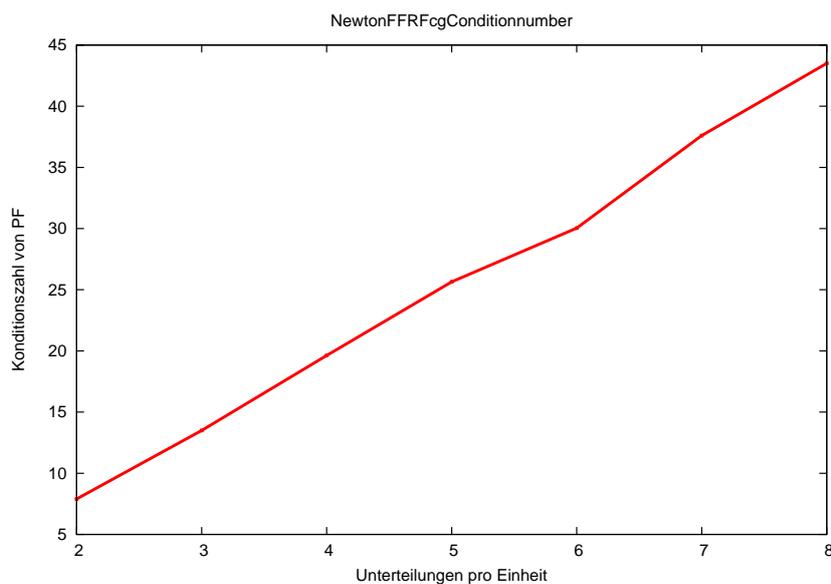


Abbildung 19: Konditionszahl im Beispiel 5.4 der Matrix PF mit steigender Feinheit

5.2.3 Konvergenzverhalten der Newtonverfahren

Weiters wurde der Verlauf der relativen Norm des Residuums der Newtonverfahren für das Beispiel 5.1 (fixes Mesh) beobachtet.

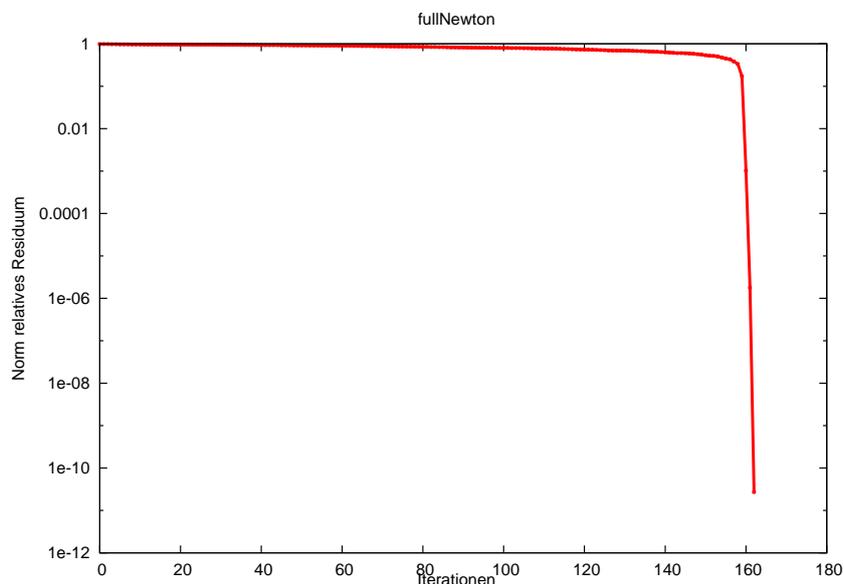


Abbildung 20: Norm des relativen Residuums für das Beispiel 5.1 im Verlauf der Newtoniteration für das vollnichtlineare Newtonverfahren

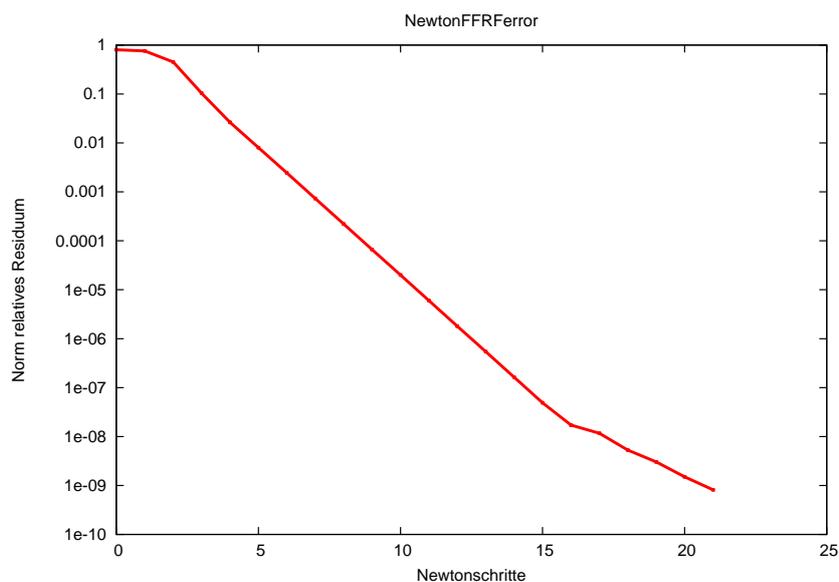


Abbildung 21: Norm des relativen Residuums für das Beispiel 5.1 im Verlauf der Newtoniteration für das FFRF-FETI Newtonverfahren

In Abbildung 20 sieht man, dass beim vollnichtlinearen Newtonverfahren die relative Norm erst sehr stark sinkt, wenn man in der Nähe der Lösung ist. In Abbildung 21 beim FFRF-FETI Newtonverfahren sinkt die relative Norm des Residuums gleichmäßiger.

Auch für das Beispiel 5.2 ist der Verlauf des relativen Residuums ähnlich (Abbildung 22 und Abbildung 23):

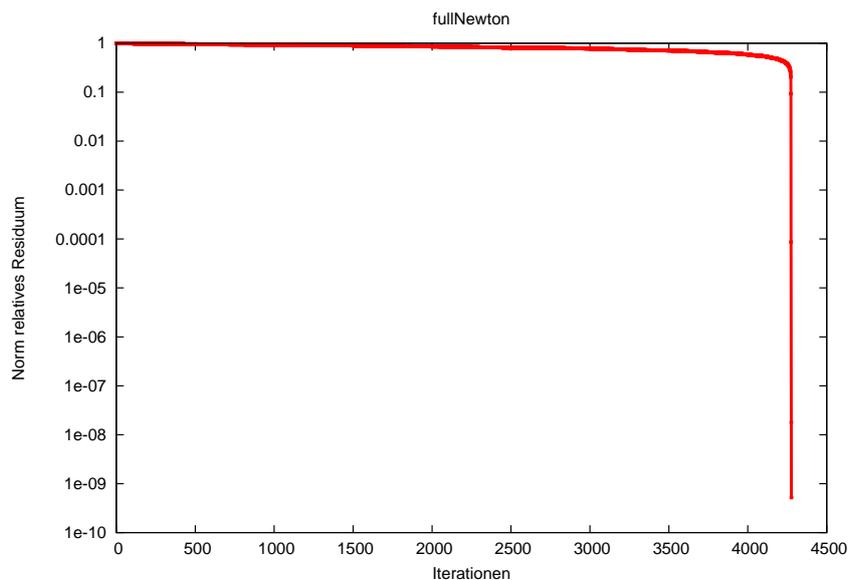


Abbildung 22: Norm des relativen Residuums für das Beispiel 5.2 im Verlauf der Newtoniteration für das vollnichtlineare Newtonverfahren

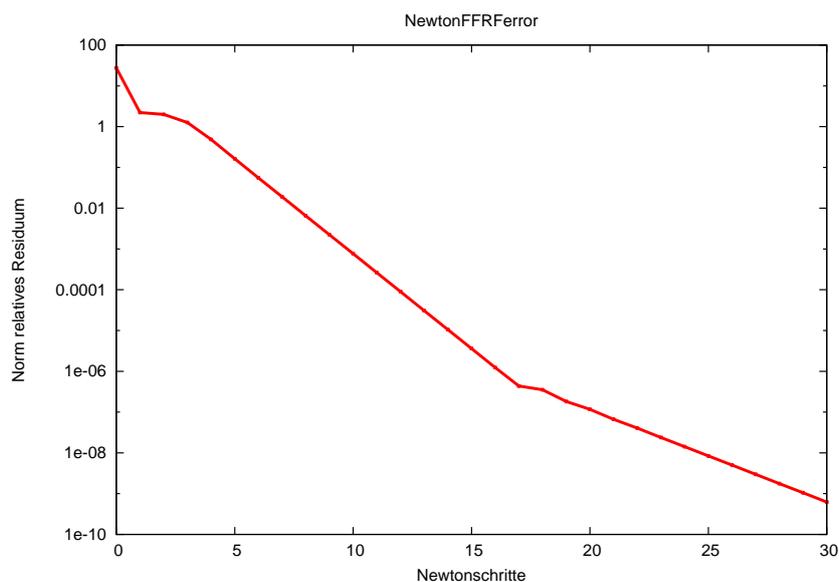


Abbildung 23: Norm des relativen Residuums für das Beispiel 5.2 im Verlauf der Newtoniteration für das FFRF-FETI Newtonverfahren

In Abbildung 23 sieht man, dass das erste relative Residuum nicht bei 1 liegt. Beim Ausführen mit Algorithmus 4.8 mit dem Null-Vektor als Startvektor konvergiert das Newton FFRF-FETI Verfahren nicht. Das Problem dabei liegt im ersten Newtonschritt. Also wurde die Dämpfung nur für den ersten Newtonschritt ausgeschaltet. Dies kann so betrachtet werden, dass die Lösung im ersten ungedämpften Newtonschritt als Startvektor gewählt wird.

6 Ausblick

Die hier vorliegende Masterarbeit kann in mehrere Richtungen erweitert werden:

- Das rechteckige Gebiet könnte gleichzeitig in x -Richtung und in y -Richtung zerlegt werden. Weiters könnte die Methode auf komplexere Gebiete erweitert werden. Dies führt dann jedoch auf eine komplexere Art der Zerlegung in Frames und müsste im Programm entsprechend implementiert werden. Die Wahl der Sperrungen bei komplexeren Frames funktioniert nicht mehr so einfach wie in Abschnitt 4.2.2, kann jedoch durch simple Heuristiken und durch Lemma 4.1 angepasst werden.
- In dieser Arbeit wird für den FFRF Ansatz ein modifiziertes Newton Verfahren verwendet (die Jacobimatrix (4.16) wurde modifiziert). Um diese Modifikation zu vermeiden, müsste unsere FETI Methode zur Lösung der Gleichung (4.14) entsprechend angepasst werden. Es ist nicht a priori klar wie dabei vorgegangen werden müsste.
- Das CG-Verfahren zur Lösung der Gleichung (4.24) könnte vorkonditioniert werden. Es gibt dazu in der Literatur [7, Kapitel 6] einen bekannten Ansatz, der jedoch nicht direkt (eins zu eins) auf unseren Fall übertragbar ist. Dabei könnte die Veränderung des Konvergenzverhaltens im Vergleich zum nicht vorkonditionierten Problems beobachtet werden. Im Erfolgsfall würde das zu wesentlich kürzeren Laufzeiten bei großdimensionalen Problemen führen.
- Eine weitere Möglichkeit wäre die Erweiterung auf den dreidimensionalen Fall. Dabei ergeben sich neben dem Mesh auch Schwierigkeiten bei der Parametrisierung der totalen Verschiebung. Es ergeben sich z.B. 2 Drehwinkel für den Frame (siehe etwa [5]). Folglich ergibt sich eine andere Struktur der Jacobi-Matrix als in Abschnitt 4.7.2, und es ist a priori nicht klar, ob etwa ein entsprechend modifiziertes Newton-FETI-Verfahren möglich ist, bzw. so schnell konvergent wie im zweidimensionalen Fall.

Literatur

- [1] Dietrich Breass, Finite Elemente. Theorie, schnelle Löser und Anwendung in der Elastizitätstheorie., Berlin, Springer, 2007
- [2] Walter Zulehner, Lecture Notes for the Course Numerical Methods for Continuum Mechanics 1, Institute for Computational Mathematics Johannes Kepler University Linz , Summer Semester 2011
- [3] Walter Zulehner, Numerische Mathematik. Eine Einführung anhand von Differentialgleichungsproblemen Band 1: Stationäre Probleme, Birkhäuser, Berlin, 2008
- [4] Walter Zulehner überarbeitet von Ewald Lindner, Skriptum zur Vorlesung Numerische Analysis, Institut für Numerische Mathematik Johannes Kepler Universität Linz, Wintersemester 2055/06, 4. Auflage
- [5] Ahmed A. Shabana, Dynamics of multibody systems, Second Edition, Cambridge university press, Cambridge, 1998
- [6] A. Toselli and O. B. Widlund, Domain Decomposition Methods – Algorithms and Theory, Springer Series in Computational Mathematics, Springer, Berlin, 2005
- [7] Clemens Pechstein, Skriptum zur Vorlesung Fast Solvers Schwarz Methods, Domain Decomposition Methods, and FETI , Institute for Computational Mathematics Johannes Kepler University Linz, Summer Semester 2011
- [8] P.G. Ciarlet, Mathematical Elasticity, Vol. I : Three-Dimensional Elasticity, Series Studies in Mathematics and its Applications, North-Holland, Amsterdam, 1988.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Masterarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe. Die vorliegende Masterarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Linz, März 2013

Klemens Reindl