

TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Numerische Mechanik

Effiziente Lösungsverfahren für Fluid-Struktur-Interaktions-Probleme

Ulrich Küttler

Vollständiger Abdruck der von der Fakultät für Maschinenwesen
der Technischen Universität München zur Erlangung des akade-
mischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Nikolaus A. Adams

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Wolfgang A. Wall
2. Prof. Matthias Heil, Ph.D.
The University of Manchester / UK

Die Dissertation wurde am 15. Juli 2009 bei der Technischen
Universität München eingereicht und durch die Fakultät für
Maschinenwesen am 13. Oktober 2009 angenommen.

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit dem Lösen von Mehrfeldproblemen am Beispiel der Fluid-Struktur-Interaktion (FSI). Das Augenmerk liegt auf der Interaktion flexibler Strukturen mit inkompressiblen Flüssigkeiten. Dabei werden Strukturmodelle eingesetzt, die sehr große Deformationen realistisch abbilden. Das Fluid wird mit der Navier-Stokes-Gleichung modelliert, wobei die Deformationen der Struktur im Rahmen einer „Arbitrary Lagrangean Eulerian“ (ALE) Beschreibung berücksichtigt werden. Struktur und Fluid sind am Interface zwischen den Feldern gekoppelt. Durch die ALE Beschreibung des Fluidgebiets entsteht ein drittes, unphysikalisches Feld, das eine Volumenkopplung mit dem Fluidfeld aufweist.

In der Arbeit werden effiziente Lösungsverfahren für komplexe dreidimensionale Problemstellungen entwickelt. Darüber hinaus ist es die Absicht der Arbeit, die zur Simulation von Fluid-Struktur-Interaktions-Problemen zur Verfügung stehenden Lösungsverfahren vorzustellen und zu klassifizieren. Die Fluid-Struktur-Interaktion ist dabei das Anwendungsgebiet, das spezialisierte Lösungsverfahren erfordert und somit die hier vorgestellten Bemühungen motiviert.

Das Bedürfnis, Fragestellungen aus der Fluid-Struktur-Interaktion am Rechner zu simulieren, stellt gleichsam den Archimedischen Punkt der vorliegenden Arbeit dar. Gleichzeitig ist es die Absicht der Darstellung, die Konstruktionsprinzipien und Bestandteile der Lösungsverfahren zu beleuchten. Dadurch wird einerseits der Zusammenhang zwischen den Algorithmen deutlich und somit eine bessere Einordnung der Lösungsmethoden ermöglicht. Andererseits stehen so die Bausteine zur Verfügung, um Mehrfeldprobleme jenseits der Fluid-Struktur-Interaktion numerisch behandeln zu können.

Der zentrale Punkt ist deshalb die Abstraktion der Details der mechanischen Problemstellung. Es werden prototypische Lösungsverfahren für alle beteiligten Felder vorgestellt, die, ohne die feldspezifischen Details zu beachten, zu Lösungsverfahren für das gekoppelte Problem zusammengesetzt werden können. Dieser Kopplungsansatz, der die unveränderten Lösungsverfahren der Einzelfelder miteinander verbindet, ist der Startpunkt des betrachteten Verfahrensspektrums. Die weiteren Lösungsverfahren im Spektrum lassen sich ableiten, indem die Position der Kopplung zwischen den Feldern schrittweise tiefer in den Algorithmus gelegt wird. Dabei zeigen die numerischen Beispiele, daß die Position der Kopplung im Lösungsverfahren, und somit der Grad der Spezialisierung des Algorithmus, auf die Effizienz des Verfahrens einen entscheidenden Einfluß nimmt.

Die wesentliche Einsicht, die der vorliegenden Diskussion zu Grunde liegt, ist das Erkennen des FSI Problems als gekoppeltes nichtlineares Problem, das mit den Methoden der numerischen Mathematik gelöst werden kann. Diese einfache Einsicht, die auf der Abstraktion der mechanischen Details fußt, führt zu dem beschriebenen Verfahrensspektrum. Die Abstraktion der mechanischen Details folgt dabei einem Gedankengang, der in der Softwareentwicklung im Zuge der Objekt-Orientierung eine große Verbreitung gefunden hat. Die Implementierung der vorgestellten Verfahren in baci, die die Aussagen dieser Arbeit erst ermöglicht, folgt deshalb der gleichen Aufteilung zwischen allgemeinen Lösungsmethoden und spezifischen Feldproblemen.

Abstract

The present work deals with solution schemes for multi-field problems in general and fluid-structure interaction problems in particular. The emphasis is on the interaction of flexible structures with incompressible flows. The structural models that are used allow for a realistic description of large deformations. The fluid is modeled using the Navier-Stokes equation and uses an "Arbitrary Lagrangean Eulerian" (ALE) formulation in order to take the deformation of the structure into account. Structure and fluid are coupled at the interface. Due to the ALE description of the fluid, a third non-physical field comes into play, which introduces a volume coupling with the fluid field.

In this work efficient solution algorithms for complex three-dimensional problems are developed. Furthermore, the intention of this work is to present and classify the solution schemes available for the simulation of fluid-structure interaction problems. Fluid-structure interaction is an application area that requires specialized solution algorithms and motivates the presented efforts.

The need to simulate fluid-structure interaction problems represents the Archimedean point of this work. At the same time it is the intention of this presentation to describe the design principles and elements of the solution algorithms. This shows the relationship between the algorithms and therefore allows for a better classification of the solution methods. However, these building blocks can be used for multi-field problems beyond fluid-structure interaction.

The central point in this work is the abstraction from the details of the mechanical problem. There are prototypical solution procedures for all fields. These can be combined to obtain algorithms for solving the coupled problem without attention to field-specific details. This coupling of the unmodified solution algorithms of the individual fields is the starting point of the considered spectrum of schemes. The other solution methods in the spectrum can be deduced by placing of the coupling between fields gradually deeper inside the algorithm. The numerical examples show that the position of the coupling in the solution algorithm, and thus the degree of specialization of the algorithm, has a decisive influence on the efficiency of the solution process.

The essential insight that the present discussion is based on is the recognition of the coupled FSI problem as a non-linear problem which can be solved using the methods of numerical mathematics. This simple insight, on which the abstraction of the mechanical details is based, leads to the described spectrum of methods. The abstraction of the mechanical details follows from observations that are common place in object-orientated software development. The implementation of the presented schemes in baci, on which the present work builds, follows the very same division between general solution techniques and specific field problems.

Dank

Die vorliegende Arbeit ist in den Jahren 2003 bis 2009 im Rahmen meiner Tätigkeit als wissenschaftlicher Assistent am Lehrstuhl für Numerische Mechanik entstanden. In dieser Zeit durfte ich am Wachstum des Lehrstuhls teilhaben und die Entwicklung einer ganzen Reihe von Forschungsgebieten miterleben. Rückblickend empfinde ich diese Jahre als ausgesprochen bereichernd, sowohl in fachlicher Hinsicht als auch in Bezug auf meine persönliche Entwicklung.

Zu danken habe ich für diese Jahre an erster Stelle Prof. Dr.-Ing. Wolfgang A. Wall, der mit unermüdlichem Einsatz einen Lehrstuhl geschaffen hat, an dem nicht nur eine sehr große Breite spannender Themen verfolgt wird, sondern darüber hinaus eine ausgesprochen motivierende, optimistische Arbeitsatmosphäre herrscht. Die hier vorliegende Arbeit wäre ohne die weitsichtige Anleitung von Professor Wall nicht entstanden. Dabei ist sein Einfluß nicht auf die fachliche Anleitung meiner Arbeit beschränkt. Professor Wall ist mir als Vorgesetzter stets auch ein Vorbild gewesen. Für all das möchte ich ganz ausdrücklich danken.

Dank schulde ich weiterhin Prof. Ph.D. Matthias Heil. Jede meiner Begegnungen mit Professor Heil war fachlich sehr ertragreich und menschlich ausgesprochen erfreulich. Die zahlreichen Gespräche während des Aufenthalts von Professor Heil am Lehrstuhl für Numerische Mechanik im Sommer 2007 haben ihren unmittelbaren Niederschlag in dieser Arbeit gefunden. Ohne seine Anregungen wäre eine sehr viel schlichtere Arbeit entstanden. Auch als Prüfer mit einer großen Menge sehr ins Detail gehender Anmerkungen hat Professor Heil meinen Erkenntnisprozess erheblich bereichert.

Weiterhin möchte ich meinem Freund und Kollegen Dr.-Ing. Michael Gee sehr herzlich danken. Michael ist als Freund von klarer Aussprache und geordneten Erklärungen nicht müde geworden, mir sein Wissen zu Mehrgitterverfahren, Vorkonditionierern und der Programmierung mit Trilinos weiterzugeben. Seine Entscheidung, das Software-Framework am Lehrstuhl zu modernisieren, hat die Voraussetzung für meine Arbeit an den Lösungsalgorithmen für Fluid-Struktur-Interaktionsprobleme erst geschaffen.

Danken möchte ich auch, wieder einmal, meiner Schwester Dr.-Ing. Christiane Förster, die mir auch während meiner Arbeit am Lehrstuhl zuverlässig vorausgegangen ist. Christiane hat mir nicht nur zahlreiche Details zur Fluidstabilisierung und zur Zeitintegration erläutert, ihr verdanke ich die Möglichkeit am Lehrstuhl für Numerische Mechanik zu arbeiten überhaupt erst. Den Kontakt zu Professor Wall hat Christiane vermittelt, wodurch sie an den letzten sechs Jahren eine erhebliche Mitschuld trägt.

Schließlich habe ich allen meinen Kollegen am Lehrstuhl sehr herzlich zu danken. Es ist ein Privileg, mit so vielen intelligenten und sympathischen Zeitgenossen zusammenarbeiten zu dürfen. Ich habe das Arbeitsumfeld und die Gespräche am Lehrstuhl immer sehr genossen. Es war mir ein Vergnügen und eine Ehre, einer von Euch zu sein.

Im Oktober 2009
Ulrich Küttler

Inhaltsverzeichnis

Zusammenfassung	i
Abstract	iii
Dank	v
Bezeichnungen	ix
Kapitel 1. Einleitung	1
1.1. Die Beziehung zur Wissenschaft	1
1.2. Numerische Experimente	2
1.3. Wirklichkeitsnahes Abbilden der Physik	3
1.4. Ausgangspunkte	4
1.5. Kapitelübersicht	4
Kapitel 2. Parallele Berechnungen	7
Kapitel 3. Lösungsverfahren für nichtlineare Gleichungen	9
3.1. Newton-Verfahren	10
3.2. Lösungsmethoden für lineare Gleichungssysteme	12
3.3. Vorkonditionierung	16
3.4. Newton-Krylov-Verfahren	22
3.5. Fixpunkt-Verfahren	24
3.6. Vektorextrapolation	28
3.7. Abbruchkriterien	30
Kapitel 4. Fluid und Struktur als Einzelfeldprobleme	33
4.1. Kontinuumsmechanik	33
4.2. Das Strukturfeld	34
4.3. Das Fluidfeld	41
4.4. Das ALE-Feld	50
Kapitel 5. Spektrum der FSI Lösungsverfahren	53
Kapitel 6. Partitionierte Lösungsverfahren für FSI Probleme	59
6.1. Partitionierung	60
6.2. Exegese	69
6.3. Lösungsverfahren für das partitionierte FSI Problem	70
6.4. Nichtpassende Netze	72
Kapitel 7. Lösungsmethoden für Dirichlet-Neumann partitionierte FSI Probleme	73
7.1. Fixpunkt-Verfahren	73
7.2. Newton-Verfahren	80
7.3. Vektorextrapolation	84
7.4. Weitere Lösungsverfahren für partitionierte FSI Probleme	86
Kapitel 8. Monolithische Newton-Verfahren zum Lösen des FSI Problems	89

8.1.	Dirichlet-Neumann partitionierte Newton-Verfahren	92
8.2.	Lineare Block-Gauß-Seidel Löser	96
8.3.	Vorkonditionierte Newton-Krylov-Verfahren	100
8.4.	Newton-Krylov-Verfahren mit gekoppeltem Mehrgitter-Vorkonditionierer	103
Kapitel 9.	Bewertung der Verfahren	107
9.1.	Druckwelle in einer flexiblen Röhre	107
9.2.	FSI Simulation eines Bronchialbaums	114
9.3.	FSI Simulation eines abdominalen Aortenaneurysmas	118
Kapitel 10.	Ausblick	123
Anhang A.	Implementierung von FSI Lösungsmethoden	125
A.1.	Literaturhinweise	125
A.2.	Anforderungen an Software zur numerischen Simulation	126
A.3.	Zur Architektur der FSI Algorithmen in baci	127
Literaturverzeichnis		129

Bezeichnungen

Allgemein

$(\cdot)^{n-1}$	Größe zum vorletzten Zeitschritt
$(\cdot)^n$	Größe zum letzten Zeitschritt
$(\cdot)^{n+1}$	Größe zum neuen Zeitschritt
$(\cdot)^S$	Größe des Strukturfeldes
$(\cdot)^F$	Größe des Fluidfeldes
$(\cdot)^G$	Größe des ALE-Feldes
$(\cdot)_\Gamma$	Vektor von Interface-Freiheitsgraden
$(\cdot)_I$	Vektor aller innerer Freiheitsgrade eines Feldes
$(\cdot)_D$	Vektor von Dirichlet-Freiheitsgraden
$(\cdot)_\Omega$	Vektor aller Freiheitsgrade außer der Dirichlet-Freiheitsgrade
$ \cdot $	Unspezifizierte Vektornorm
$ \cdot _2$	L_2 -Norm eines Vektors
$ \cdot _\infty$	L_∞ -Norm eines Vektors
∇	räumlicher Ableitungsoperator
ε_{nl}	Toleranzgrenze beim Testen des nichtlinearen Residuums
ε_{lin}	Toleranzgrenze beim Testen des linearen Residuums
ε^S	Toleranzgrenze beim Testen des nichtlinearen Strukturresiduums
ε^F	Toleranzgrenze beim Testen des nichtlinearen Fluidresiduums
ε^Γ	Toleranzgrenze beim Testen des nichtlinearen Residuums am Interface
\mathcal{O}	nicht berücksichtigte Terme der angegebenen Ordnung und höher

Feldprobleme

$\Omega_x, \Omega_y, \Omega_z$	Referenzgebiet, Raumgebiet, Materialgebiet
Γ_D, Γ_N	Dirichletrand, Neumannrand
\mathbf{X}, \mathbf{x}	Ausgangsposition, Momentanposition
\mathbf{d}, \mathbf{u}	Verschiebung, Geschwindigkeit
\mathbf{c}, \mathbf{u}^G	konvektive Geschwindigkeit, Deformationsgeschwindigkeit
\mathbf{F}	Deformationsgradient
\mathbf{R}, \mathbf{U}	Rotationstensor, rechter Strecktensor
\mathbf{C}	Rechts-Cauchy-Green-Tensor
\mathbf{E}	Lagrangescher Verzerrungstensor
ε	Deformationsgeschwindigkeit
$\delta \mathbf{d}, \delta \mathbf{E}$	virtuelle Verrückung, Variation der Lagrangeschen Verzerrungen
\mathbf{v}, q	Wichtungsfunktionen
\mathbf{S}	zweiter Piola-Kirchhoff Spannungstensor

σ	Cauchy-Spannungstensor
$\mathbf{b}, \mathbf{t}, \mathbf{f}_{\text{res}}$	Volumenlast, Randlasten, Residuum
λ^S, μ^S	Lamé-Konstanten
E, ν^S	Elastizitätsmodul, Querkontraktion
ρ^S, ρ^F	Strukturdichte, Fluidichte
\bar{p}, μ	Druck, dynamischen Viskosität
p, ν	kinematischer Druck, kinematische Viskosität

Diskretisierung

$\mathbf{d}, \dot{\mathbf{d}}, \ddot{\mathbf{d}}$	Verschiebung, Geschwindigkeit, Beschleunigung
$\mathbf{u}, \dot{\mathbf{u}}$	Geschwindigkeit, Beschleunigung
$\dot{\mathbf{u}}$	Zwischengeschwindigkeit
p	kinematischer Druck
\mathbf{f}, \mathbf{r}	Kraftvektor, Residuumsvektor
$\mathbf{M}, \mathbf{D}, \mathbf{K}$	Massenmatrix, Dämpfungsmatrix, Steifigkeitsmatrix
\mathbf{N}, \mathbf{G}	konvektive und viskose innere Kräfte, diskreter Gradientenoperator
\mathbf{L}	diskreter Laplace-Operator
\mathbf{M}_D	diagonalisierte Massenmatrix
$\mathbf{I}, \mathbf{0}$	Einheitsmatrix, Nullmatrix
$\mathbf{N}^d, \mathbf{N}^u, \mathbf{N}^p$	Formfunktionen
$\mathbf{f}_{\text{int}}, \mathbf{f}_{\text{ext}}$	nichtlinearer Vektor der inneren Kräfte, Vektor der äußeren Kräfte
β, γ	Newmark-Parameter
α_f, α_m	generalized- α Parameter
θ	one-step- θ Parameter
Δt	Zeitschritt
τ	Stabilisierungsparameter
$\mathbf{S}, \mathbf{F}, \mathbf{A}$	effektive Systemmatrizen des Struktur-, Fluid- und ALE-Feldes

Interface zwischen Fluid und Struktur

\mathcal{F}_Γ	nichtlinearer Fluidoperator
\mathcal{F}_Γ^{-1}	inverser nichtlinearer Fluidoperator
\mathcal{S}_Γ	nichtlinearer Strukturoperator
\mathcal{S}_Γ^{-1}	inverser nichtlinearer Strukturoperator
$\mathcal{F}_{\Gamma,\text{lin}}$	Linearisierung des Fluidoperators \mathcal{F}_Γ
$\mathcal{F}_{\Gamma,\text{lin}}^{-1}$	Linearisierung des inversen Fluidoperators \mathcal{F}_Γ^{-1}
$\mathcal{S}_{\Gamma,\text{lin}}$	Linearisierung des Strukturoperators \mathcal{S}_Γ
$\mathcal{S}_{\Gamma,\text{lin}}^{-1}$	Linearisierung des inversen Strukturoperators \mathcal{S}_Γ^{-1}
\mathcal{M}_Γ	Vorkonditionierungsoperator
\mathcal{M}_Γ^{-1}	inverser Vorkonditionierungsoperator
$\hat{\mathcal{F}}_\Gamma$	linearer Fluidoperator
$\hat{\mathcal{S}}_\Gamma$	linearer Strukturoperator
\mathbf{J}^{FSI}	Jacobimatrix
\mathbf{M}	Vorkonditionierungsmatrix

Nichtlineare Gleichungssysteme

f	nichtlineare Funktion
x	Unbekanntenvektor
J	Jacobimatrix der nichtlineare Funktion
ω_i	dynamischer Relaxationsparameter

Lineare Gleichungssysteme

n	Anzahl der Unbekannten des Gleichungssystems
A	Koeffizientenmatrix des linearen Systems
b	rechte Seite des linearen Systems
x	Vektor der Unbekannten
L, U	untere und obere Dreiecksmatrix
k	Dimension des Krylov-Raumes
\mathbf{b}_i	Basisvektor des Krylov-Raumes
\mathbf{a}_j	beliebiger Vektor

GMRES

\mathbf{v}_i	orthonormierter Basisvektor des Krylov-Raumes
\mathbf{V}_{k+1}	$n \times (k+1)$ -Matrix \mathbf{V}_{k+1} aus den $k+1$ orthonormierter Basisvektor \mathbf{v}_i
\mathbf{y}_k	Skalierungsfaktoren der k Basisvektoren \mathbf{v}_i
$\bar{\mathbf{H}}_k$	$(k+1) \times k$ -Matrix
$h_{i,j}$	Eintrag der Matrix $\bar{\mathbf{H}}_k$

Vorkonditionierung

M	Approximation der Koeffizientenmatrix A
$\mathbf{M}_L, \mathbf{M}_R$	linksseitiger und rechtsseitiger Vorkonditionierer
D, L, U	Diagonale, untere Dreiecksmatrix und obere Dreiecksmatrix von A
y	beliebiger Vektor zum Anwenden des Vorkonditionierers
z	vorkonditionierter Vektor
$\tilde{\mathbf{x}}$	Unbekanntenvektor bei rechtsseitiger Vorkonditionierung
ω	fester Relaxationsparameter

Mehrgitterverfahren

N_{levels}	Anzahl der Ebenen im V-Zyklus
\mathbf{P}_k	Interpolationsoperator auf Ebene k
\mathbf{R}_k	Vergrößerungsoperator auf Ebene k
$\hat{\mathbf{S}}_k$	Glättungsverfahren auf Ebene k
$\mathbf{P}_k^{\text{FSI}}$	Interpolationsoperator des gekoppelten Mehrgitter-Verfahrens auf Ebene k
$\mathbf{R}_k^{\text{FSI}}$	Vergrößerungsoperator des gekoppelten Mehrgitter-Verfahrens auf Ebene k
$\hat{\mathbf{S}}_k^{\text{FSI}}$	Glättungsverfahren des gekoppelten Mehrgitter-Verfahrens auf Ebene k

Algorithmen

$\max_{\text{n}} \text{niter}$	Maximale Anzahl der Iterationen zum Lösen eines nichtlinearen Problems
$\max_{\text{i}} \text{iiter}$	Maximale Anzahl der Iterationen zum Lösen eines linearen Problems

KAPITEL 1

Einleitung

Diese Arbeit handelt von Lösungsverfahren. Das Problem, das gelöst werden soll, ist die Simulation der Wechselwirkung von Fluid und Struktur mit Hilfe einer auf finiten Elementen basierenden Diskretisierung. Die Bedeutung der Fluid-Struktur-Interaktion (FSI) braucht nicht sonderlich betont zu werden. Jedes Kind, das in Dreckpfützen spielt, ist von der Fluid-Struktur-Interaktion fasziniert. Diese Faszination ist für sich ein ausreichender Grund, die Interaktion zwischen Fluiden und Strukturen wissenschaftlich zu untersuchen. Dabei ist die Interaktion von deformierbaren Festkörpern mit Fluiden—Flüssigkeiten oder Gasen—ein in Natur und Technik häufig anzutreffender Vorgang. Das Studium der Wechselwirkungsvorgänge kann bei ganz unterschiedlichen Fragestellungen zu neuen Einsichten führen.

1.1. Die Beziehung zur Wissenschaft

Wenn an dieser Stelle von Wissenschaft gesprochen wird, ist das Aufstellen und Testen von Theorien über Aufbau und Verhalten der Natur gemeint. Die bloße wissenschaftliche Methode macht noch keine Wissenschaft. Im Besonderen wird eine Arbeit nicht wissenschaftlich durch das Einhalten der an wissenschaftliche Arbeiten gestellten Kriterien; das Anhängen eines Literaturverzeichnisses allein ist nicht ausreichend.

Die abschließende Antwort auf die Frage, was Wissenschaft eigentlich ist, liegt weit außerhalb dieser einleitenden Bemerkungen. Als Richtschnur für die folgenden Überlegungen möge die in Birks [9] wiedergegebene, etwas ruppige Definition von Ernest Rutherford dienen:

All science is either physics or stamp collecting.

Der physikalische Teil der Wissenschaft, und nur der physikalische Teil ist im Kontext dieser Arbeit relevant, basiert auf direkten Beobachtungen der Natur als Grundlage der aufzustellenden Theorien. Das Besinnen auf die direkte Anschauung der Natur und das Aufbereiten von Experimenten für wiederholte Ausführungen gilt gemeinhin als Ausbruch aus der mittelalterlichen Scholastik und Geburtsstunde der modernen Wissenschaft.

Das Aufstellen von Theorien bleibt dennoch eine spekulative Tätigkeit, wie in Popper [122] ausgeführt wird. Die Darstellung in Hamming [70] legt nahe, daß selbst Galileo, der vielfach als Begründer des experimentellen Ansatzes gilt, seine Erkenntnisse durch Überlegung gewonnen hat. Experimente dienen dem Überprüfen von Theorien insofern, als das Nichtzutreffen von Theorien anhand von Experimenten erkannt werden kann. Die Richtigkeit einer Theorie kann durch das Ausführen von Experimenten jedoch nicht bewiesen werden. Im Besonderen ist auch ein Experiment ohne eine begleitende Theorie sehr wenig hilfreich. Die Bedeutung der Erkenntnistheorie für die Wissenschaft wird, neben vielem anderen, in Schilpp [136] besprochen.

Da numerische Verfahren keine Aussage über die Natur beinhalten, kann das Aufstellen von numerischen Verfahren für sich nicht als wissenschaftliche Betätigung gelten, unabhängig vom Nutzen der Verfahren für wissenschaftliche Untersuchungen. In diesem Sinne

ist auch die vorliegende Arbeit nicht wissenschaftlich. Vielmehr werden Werkzeuge für die wissenschaftliche Untersuchung der Fluid-Struktur-Interaktion zusammengestellt. Das Interesse an der Simulation von FSI Problemen ist damit der Archimedische Punkt dieser Arbeit. Ein weiteres Ziel ist es, die Konstruktion dieser Werkzeuge, der Lösungsverfahren, zu veranschaulichen, um die Verfahrensidee auf vergleichbare Fragestellungen übertragen zu können.

Doch die in dieser Arbeit besprochenen Verfahren sind nicht nur im wissenschaftlichen Kontext von Interesse. Auch das Ingenieurwesen lebt von der direkten Anschauung der Natur und dem Anwenden der Naturgesetze. Im Gegensatz zum Wissenschaftler möchte der Ingenieur dabei nicht die Natur beschreiben, sondern Neues erschaffen. Hamming [72] bringt die unterschiedlichen Ansätze von Wissenschaft und Ingenieurwesen prägnant auf den Punkt:

In science if you know what you are doing you should not be doing it.
In engineering if you do not know what you are doing you should not be doing it.

Wobei die Wissenschaft stets einen Anteil Ingenieuraufgaben mit sich bringt. Im Ingenieurwesen dagegen treten immer wieder Probleme auf, die mit den bekannten Methoden allein nicht zu bewältigen sind. Für die in dieser Arbeit besprochenen Lösungsverfahren ist die Unterscheidung zwischen Wissenschaft und Ingenieurwesen deshalb unerheblich.

1.2. Numerische Experimente

Das Verhalten von Fluiden und Strukturen ist der Gegenstand der Mechanik, die Fluid-Struktur-Interaktion ist folglich ein mechanischer Vorgang. Wie in Hamming [70] ausgeführt wird, ist die Mathematik die Sprache der Physik. Die Feldprobleme der Mechanik werden von Differentialgleichungen beschrieben. Die Differentialgleichungen der Mechanik sind für sich wissenschaftliche Theorien, also spekulativ aufgestellte Vermutungen. Der Gültigkeitsbereich der klassischen Mechanik ist dabei sehr gut bekannt, so daß aus mechanischen Modellen mit voller Zuversicht Aussagen abgeleitet werden können. Die historische Entwicklung der Mechanik wird von Szabó [151] sehr ausführlich dokumentiert.

Aufmerksamkeit verlangen die mechanischen Modelle immer dann, wenn Fragestellungen außerhalb des üblichen Einsatzbereichs betrachtet werden sollen. Für die Beschreibung der Fluid-Struktur-Interaktion wird etwa mit der Kontinuumshypothese gearbeitet, also der Aufbau der Natur aus Atomen durch die Annahme einer kontinuierlichen Masse ersetzt. Ohne diese Annahme sind Betrachtungen jenseits der Molekülgröße nicht möglich. Gleichzeitig sind extrem kleine mechanische Systeme denkbar, die mit einer auf der Kontinuumshypothese aufbauenden Theorie nur unzureichend beschrieben werden können. Als Folge der Kontinuumshypothese sind über das Verhalten der verwendeten Materialien Annahmen zu treffen. Dergleichen Annahmen stellen für sich wieder Theorien dar, deren Gültigkeitsbereich beachtet werden muß.

Unter der Annahme, daß zutreffende Modellannahmen gewählt wurden, lassen sich durch das Lösen von Differentialgleichungen die Ergebnisse der entsprechenden Experimente voraussagen. Eine direkte Lösung der Differentialgleichungen ist jedoch im Allgemeinen nicht möglich. Für praktische Abschätzungen wurden deswegen vielfach stark vereinfachte Näherungen verwendet. Eine alternative Herangehensweise, die seit dem Aufkommen der Rechentechnik zur Verfügung steht, ist das Nachrechnen der Differentialgleichungen im Rahmen eines numerischen Experiments. Die in dieser Arbeit vorgestellten Lösungsalgorithmen werden zum Durchführen von numerischen Experimenten an FSI Problemen benötigt.

Numerische Experimente sind wünschenswert, weil sie, verglichen mit tatsächlichen Experimenten, oft mit sehr viel geringeren Mühen, d.h. sehr viel geringeren Kosten, ausgeführt werden können. Die Numerik ist unabhängig von der Größenordnung der Fragestellung, Berechnungen können im Nanobereich genau so ausgeführt werden wie in geologischen Dimensionen. Darüber hinaus ist es bei numerischen Experimenten sehr viel leichter, Beobachtungen anzustellen. Jeder Punkt des berechneten Gebiets ist jederzeit einsehbar. Meßfehler treten nicht auf.

Es ist dennoch zu bedenken, daß numerische Experimente Beobachtungen an mathematischen Modellen ermöglichen, während das tatsächliche Experiment direkt auf die physikalischen Vorgänge sieht. Die Grenzen des numerischen Experiments sind durch die Gültigkeitsgrenzen der mathematischen Modelle gezogen. Wie in Hamming [72] besprochen wird, bricht das numerische Experiment mit dem Imperativ der modernen Wissenschaft, bei der Suche nach Erklärungen auf die Natur selbst zu blicken. Die Wahl des mathematischen Modells bestimmt das Ergebnis der Simulation. Wirkliche Überraschungen können so nicht auftreten. Auf reale Experimente kann deshalb nicht vollständig verzichtet werden.

1.3. Wirklichkeitsnahes Abbilden der Physik

Ein weiterer Aspekt der in dieser Arbeit antizipierten numerischen Experimente ist das wirklichkeitsnahe Abbilden der Physik beim Betrachten komplexer mechanischer Systeme. Während die im Ingenieurwesen traditionell verwendeten Näherungen in aller Regel einzelne Effekte eines komplexen Systems abschätzen und es dem berechnenden Ingenieur überlassen, durch Überlagerung einzelner Effekte eine Näherung an das Verhalten des zusammengesetzten Problems zu finden, zielt die Simulation der vollständigen Differentialgleichung darauf ab, die gesamte Physik darzustellen und das Verhalten des mechanischen Systems vollständig zu beschreiben.

Der Vorteil der vollständigen Simulation ist der stark erweiterte Gültigkeitsbereich der zu Grunde liegenden Theorie. Die Simulation erschließt Fragestellungen einer numerischen Behandlung, die traditionell nur durch Experimente geklärt werden können. Der Preis dafür ist eine oftmals undurchsichtige Berechnung. Bei nichtlinearen Problemstellungen wie der Fluid-Struktur-Interaktion steht das Ergebnis jeder Simulation für sich. Rückschlüsse auf den Einfluß einzelner Modellanteile sind, anders als bei linearen Berechnungen, im Nachhinein kaum möglich. Insofern wird eine Punktlandung angestrebt, bzw. eine Landung nahe genug am Punkt, die es erlaubt, das berechnete Ergebnis als das physikalisch richtige Ergebnis zu interpretieren.

Um Diskretisierungsfehler gering zu halten, werden oftmals sehr große Modelle benötigt. Die damit verbundenen sehr großen Gleichungssysteme stellen auch mit der aktuell verfügbaren Hardware Schranken auf. Aus diesem Grund existieren auch für Simulationen zahlreiche Modellierungsansätze, beispielsweise bei der Behandlung von turbulenten Strömungen, oder Ansätze zur Modellreduktion, etwa durch das Einführen vereinfachter Tragwerke in der Strukturmechanik, die den numerischen Aufwand reduzieren sollen, ohne relevante Fehler in die Simulation hinein zu tragen.

Modellierungen dieser Art werden in der vorliegenden Arbeit nicht betrachtet. Die vorgestellten Verfahren zum Lösen von FSI Problemen sind jedoch unabhängig von der Formulierung der Teilfelder. Die Lösungsverfahren können deshalb losgelöst von eventuellen Modellierungen auf Feldebene verwendet werden.

Die hier diskutierten Lösungsverfahren lösen das diskretisierte FSI Problem im Rahmen von frei wählbaren Toleranzen exakt. Näherungsverfahren, die lediglich approximative Lösungen der diskretisierten Differentialgleichungen finden, werden nur am Rande betrachtet.

1.4. Ausgangspunkte

Diese Arbeit baut auf verschiedenen Arbeiten auf, die am Lehrstuhl für Baustatik und Baudynamik der Universität Stuttgart entstanden sind. Insbesondere sind das die Arbeiten von Wall [162], Mok [117] und Förster [50]. In der sehr umfangreichen Arbeit von Wall [162] werden die Grundlagen der Fluid-Struktur-Interaktion dargelegt, einschließlich einer detaillierten Darstellung der Feldgleichungen. Die vorliegende Arbeit ist als Ergänzung zu Wall [162] zu lesen. In der Arbeit von Mok [117] werden partitionierte Koppelungsverfahren ausführlich diskutiert. Die in Mok [117] besprochene Partitionierung stellt einen alternativen Zugang zu den auch in der vorliegenden Arbeit beschriebenen Partitionierungsansätzen dar. Die Arbeit von Förster [50] beschäftigt sich mit Feinheiten von bewegten Netzen, stabilisierten Fluidelementen und schwach koppelnden Verfahren. Die drei Arbeiten stellen die Grundlage für die vorliegende Diskussion auch an den Stellen dar, an denen sie nicht explizit zitiert werden.

Das Ziel der Arbeit ist die algorithmische Darstellung eines ganzen Spektrums von Verfahren, die zur Lösung allgemeiner FSI Probleme genutzt werden können. Gleichzeitig soll die enge Verwandtschaft zwischen diesen Verfahren aufgezeigt und die Konstruktion der Lösungsverfahren aus einzelnen Bestandteilen beschrieben werden. Auf diese Weise steht eine Beschreibung von Lösungsverfahren für ein spezielles Mehrfeldproblem zur Verfügung, mit der Hoffnung, aus den gleichen Bestandteilen Lösungsverfahren für ähnlich gelagerte Mehrfeldprobleme konstruieren zu können.

1.5. Kapitelübersicht

Die Besprechung der Lösungsverfahren beginnt in Kapitel 2 mit ein paar einleitenden Bemerkungen zu parallelen Berechnungen. Alle hier vorgestellten Lösungsverfahren werden auf Mehrprozessorsystemen ausgeführt. Zu diesem Zweck wird eine Implementierung der linearen Algebra verwendet, die auf eine parallele Berechnung mit verteilten Daten ausgelegt ist. Die Lösungsalgorithmen selbst bleiben damit von Gedanken zur Parallelisierung weitgehend verschont. Die parallele Berechnung wird deshalb nur am Rande erwähnt.

In Kapitel 3 werden Lösungsverfahren für nichtlineare Gleichungen zusammengetragen. Dieses Kapitel stellt eine Übersicht über die in dieser Arbeit verwendeten Verfahren aus der numerischen Mathematik dar. Diese Übersicht soll die relevanten Verfahren und deren Beziehung untereinander aufzeigen. Die mathematischen Details werden mit Bedacht der einschlägigen Literatur überlassen. Die vorgestellten Verfahren sind allgemein, ohne Bezug zur Fluid-Struktur-Interaktion, formuliert.

Fluid und Struktur werden in Kapitel 4 als Einzelfeldprobleme vorgestellt. Die Darstellung baut direkt auf die ausführlichere Diskussion aus Wall [162] auf und spannt den Bogen von der Differentialgleichung der Einzelfelder über die Diskretisierung zum Lösungsalgorithmus. Auf die Algorithmen aus Kapitel 3 wird dabei Bezug genommen. Die vorgestellten Feldlöser sind als exemplarische Lösungsalgorithmen zu verstehen, die im Weiteren zum Lösen des FSI Problems kombiniert werden können.

Kapitel 5 stellt das Spektrum der FSI Lösungsverfahren vor und leitet damit den Kern der Arbeit ein. Im Vorgriff auf die folgenden Kapitel wird der Zusammenhang zwischen unterschiedlichen algorithmischen Ansätzen zum Lösen des FSI Problems aufgezeigt. Im Zentrum der Diskussion steht dabei die Position der Feldkopplung im Lösungsalgorithmus, die als Koordinate des Verfahrensspektrums eingeführt wird. Die folgenden Kapitel füllen den so skizzierten Rahmen mit Details.

Die traditionellen partitionierten Lösungsverfahren für FSI Probleme werden in Kapitel 6 eingeführt. Dazu werden die Möglichkeiten zur Partitionierung des FSI Problems ausführlich besprochen und die jeweils verwendeten Interface-Operatoren definiert. Die

Dirichlet-Neumann-Partitionierung wird als der tragfähige Lösungsansatz herausgestellt. Die Interface-Operatoren basieren auf den in Kapitel 4 präsentierten Lösungsverfahren für die einzelnen Feldgleichungen. Um auf die Lösungsverfahren für das partitionierte FSI Problem fokussieren zu können, wird eine recht abstrakte Darstellung gewählt. Eine alternative und in gewissem Sinne komplementäre Darstellung der Partitionierung wird in Mok [117] gegeben.

Konkrete Lösungsmethoden für Dirichlet-Neumann partitionierte FSI Probleme werden schließlich in Kapitel 7 angegeben. Dazu werden die in Kapitel 3 vorgestellten Lösungsverfahren auf das nichtlineare Interface-Problem der Dirichlet-Neumann-Partitionierung angewendet. Durch die in Kapitel 6 eingeführte abstrakte Formulierung des FSI Problems gestaltet sich das Anwenden der allgemeinen Lösungsmethoden sehr einfach. Um das Verständnis für die Spezifika von partitionierten FSI Problemen zu vertiefen, werden die Lösungsmethoden deshalb auf ein einfaches Modellproblem angewendet.

Kapitel 8 bespricht den weiten Bogen der monolithischen Newton-Verfahren zum Lösen des FSI Problems. Diese Diskussion umfaßt alle Verfahren, die nicht spezifisch partitionierte Verfahren mit unabhängigen Feldlösern sind. Hier wird deshalb das Verfahrensspektrum aus Kapitel 5 erneut aufgegriffen. Die einzelnen Varianten des Newton-Verfahrens bauen nach wie vor auf den Gedanken zur Partitionierung aus Kapitel 6 und den zugehörigen Lösungsmethoden aus Kapitel 7 auf. Die in Kapitel 3 vorbereiteten Lösungsmethoden werden hier in vollem Umfang eingesetzt. Die Newton-Verfahren zum Lösen des FSI Problems bilden den dramatischen Höhepunkt im Spannungsbogen der vorliegenden Arbeit.

Die Bewertung der Verfahren in Kapitel 9 findet schließlich anhand ausgewählter numerischer Beispiele aus der Biomechanik statt. Es zeigt sich, daß die Enden des Verfahrensspektrums die für praktische Berechnungen relevanten Verfahren enthalten, während die Lösungsverfahren in der Spektrennitte lediglich dem Verständnis des Verfahrenszusammenhangs dienen. Desweiteren wird die von der Partitionierung induzierte Schwierigkeit bei ungünstigen Massenverhältnissen deutlich, die bei den in Kapitel 8 vorgestellten vorkonditionierten Newton-Krylov-Verfahren nicht auftritt.

Kapitel 10 beschließt die Arbeit mit einem Ausblick auf weitere mögliche Überlegungen im Zusammenhang mit Lösungsverfahren für FSI Probleme.

Einige Anmerkungen zur Implementierung der FSI Lösungsmethoden, in der Hauptsache kommentierte Literaturhinweise zur Softwaretechnik, sind in Anhang A zusammengetragen.

Parallele Berechnungen

Die in dieser Arbeit vorgestellten Algorithmen sind auf Clustersysteme mit zahlreichen Rechenkernen, verteiltem Speicher und schneller Netzwerkverbindung ausgelegt, werden also parallel ausgeführt. Clustersysteme, bei denen jeder Prozessor eigenen Speicher besitzt und ausschließlich auf den eigenen Speicher zugreifen kann, werden vielfach für wissenschaftliche Berechnungen eingesetzt, da sie aus Standardkomponenten aufgebaut werden können. Die parallele Berechnung auf diesen Systemen basiert auf der Aufteilung der zu bearbeitenden Daten, der Matrizen und Vektoren. Typischerweise werden die Zeilen der Matrizen und die Einträge der Vektoren gleichmäßig auf alle beteiligten Rechenkern verteilt. Jeder an der Berechnung beteiligte Rechenkern arbeitet den gleichen Algorithmus ab. Zwischen den Rechenkernen findet Datenaustausch statt.

Der Sinn der Parallelisierung ist es einerseits, die Arbeit zu verteilen und somit schneller fertig zu werden. Andererseits müssen auch die Matrizen und Vektoren verteilt werden, da bei den hier angestrebten Simulationen große Datenmengen anfallen, die typischerweise die Speicherkapazität eines einzelnen Rechners übersteigen. Dabei ist die Datenmenge oftmals die beschränkende Größe. Berechnungen mit zu wenigen Rechenkernen sind langsam, ohne ausreichenden Speicher dagegen können keine Berechnungen ausgeführt werden.

Das Parallelisieren durch einfaches Aufteilen der Matrizen und Vektoren ist sehr schlicht. Eine große Menge sehr schwer zu beherrschender Kommunikationsprobleme wird vermieden, indem jeder Rechenkern den gleichen Algorithmus abarbeitet. Im Besonderen kann das parallele Programm als ein einheitliches Programm begriffen werden, das verteilte Matrizen und Vektoren verwendet. In der Praxis gelingt es mit diesem einfachen Ansatz parallele Algorithmen zu erstellen, ohne explizit über die Parallelisierung nachdenken zu müssen. Auch in der hier folgenden Diskussion wird die Parallelisierung—von einigen wenigen Punkten abgesehen—keine Rolle spielen.

Um in dieser Art parallele Algorithmen formulieren zu können, wird eine Implementierung der linearen Algebra mit verteilten Matrizen und Vektoren benötigt. Im Zuge dieser Arbeit wird die Epetra Bibliothek aus dem trilinos Projekt verwendet, siehe dazu Heroux u. a. [76] und die Einführung von Sala u. a. [134]. Diese Bibliothek stellt die benötigten Abstraktionen zur Verfügung, so daß die besprochenen Lösungsalgorithmen auch in der Implementierung auf der Basis von Matrizen und Vektoren formuliert werden können.

Fragen der Implementierung werden in dieser Arbeit nur am Rande behandelt. Die Implementierung der Lösungsverfahren wurde für die in Kapitel 9 beschriebenen parallelen Berechnungen verwendet. Weitere Anmerkungen zur Softwareentwicklung für wissenschaftliche Berechnungen sind in Anhang A zusammengetragen.

Lösungsverfahren für nichtlineare Gleichungen

An dieser Stelle lohnt es sich, das Lösen nichtlinearer Gleichungen unabhängig vom physikalischen Hintergrund der Gleichungen zu betrachten. Diese Unabhängigkeit ist eine Illusion, die Physik des Systems und damit die Art der Gleichung bestimmt die anwendbaren Lösungsmethoden. Die Eigenschaften des nichtlinearen Problems werden in diesem Kapitel jedoch nicht diskutiert. Es wird lediglich von der Existenz einer eindeutigen Lösung ausgegangen, was bei vielen physikalischen Problemen selbstverständlich, bei nichtlinearen Gleichungen im Allgemeinen jedoch nicht gegeben ist.

Das Lösen von Systemen nichtlinearer Gleichungen ist ein weites Feld. Eine Einführung in die mathematische Theorie bietet beispielsweise die Monografie von Rheinboldt [128]. Das Newton-Verfahren als das wichtigste Verfahren zur Lösung nichtlinearer Systeme wird in allen Varianten in Deuffhard [34] oder auch Kelley [88, 89] untersucht. Auch die Lehrbücher zur numerischen Mathematik wie Quarteroni u. a. [124], Faires und Burden [43] und auch Strang [147] enthalten Einführungen in dieses weitverzweigte Thema. Hier werden Lösungsmethoden zusammengetragen, die in den folgenden Kapiteln in vielfacher Weise zur Lösung des FSI Problems verwendet werden.

Das hier betrachtete nichtlineare Problem sei als Funktion \mathbf{f} der Unbekannten \mathbf{x} gegeben, deren Nullstelle gesucht ist

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}. \quad (3.0.1)$$

Nun gibt es keine nichtlineare Algebra, es ist deshalb im Allgemeinen nicht möglich, das Gleichungssystem (3.0.1) nach der Unbekannten \mathbf{x} aufzulösen. Die Lösung \mathbf{x} kann nicht explizit berechnet werden. Um das Gleichungssystem (3.0.1) zu lösen werden deshalb iterative Verfahren eingesetzt, die sich der gesuchten Lösung schrittweise nähern. Man spricht bei iterativen Lösungsverfahren für nichtlineare Systeme auch kurz von nichtlinearen Lösern.

Der generelle Aufbau von iterativen Lösungsverfahren für nichtlineare Systeme ist in Algorithmus 1 angegeben. Der Algorithmus besteht aus einer Iteration, in der der aktuelle Lösungsvorschlag \mathbf{x}_i geprüft und, sollte noch keine hinreichende Näherung gefunden sein, durch ein neu berechnetes Inkrement $\Delta\mathbf{x}_i$ verbessert wird. Um Verbesserungen festzustellen, bedarf es natürlich einer Beurteilung der Lösungsvorschläge \mathbf{x}_i . In Algorithmus 1 wird dazu symbolisch eine Norm der nichtlinearen Funktion $\mathbf{f}(\mathbf{x}_i)$ mit einer vorgegebenen Toleranz ε_{nl} verglichen. Möglichkeiten zur Prüfung des Lösungsvorschlags werden in Abschnitt 3.7 besprochen. Die Anzahl der Iterationen in Algorithmus 1 wird auf \max_{nliter} beschränkt, um die Berechnung bei nicht konvergierenden Gleichungen abzubrechen.

Das Berechnen des Lösungsinkrements $\Delta\mathbf{x}_i$ ist in Algorithmus 1 un spezifiziert. Es existieren zahlreiche Möglichkeiten, ein Inkrement, eine Suchrichtung und die zugehörige Schrittweite, zu bestimmen.

Für die Beurteilung von iterativen Lösungsverfahren ist es relevant, für welche Problemstellungen und welche Startwerte \mathbf{x}_0 das Verfahren konvergiert und mit welcher Konvergenzgeschwindigkeit zu rechnen ist. Diese Beurteilung ist nicht Gegenstand der vorliegenden Arbeit. Die mathematischen Hintergründe der Lösungsverfahren für nichtlineare

Algorithmus 1 Allgemeines Lösungsverfahren für nichtlineare Probleme.

wähle \mathbf{x}_0
für alle $i = 0, \dots, \max_{\text{nliter}}$:
 falls $|\mathbf{f}(\mathbf{x}_i)| \leq \varepsilon_{\text{nl}}$:
 fertig
 berechne $\Delta \mathbf{x}_i$
 $\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta \mathbf{x}_i$

Gleichungen können in Rheinboldt [128] bzw. Deuffhard [34] und den dort zitierten Arbeiten gefunden werden.

In den folgenden drei Abschnitten werden die Bestandteile des vorkonditionierten Newton-Krylov-Verfahrens vorgestellt, bevor das Verfahren in Abschnitt 3.4 als Ganzes dargestellt werden kann. Das Newton-Krylov-Verfahren ist das in dieser Arbeit in der Hauptsache eingesetzte Verfahren zum Lösen nichtlinearer Gleichungen. In den Abschnitten 3.5 und 3.6 werden weitere Lösungsverfahren für nichtlineare Gleichungen besprochen, denen eine geringere praktische Bedeutung zukommt. In Abschnitt 3.7 sind schließlich ein paar Überlegungen zu den Abbruchbedingungen von iterativen Verfahren zusammengetragen, die für alle hier betrachteten Lösungsverfahren eine Rolle spielen.

3.1. Newton-Verfahren

Die wichtigste Methode, um ein Lösungsincrement $\Delta \mathbf{x}_i$ zu bestimmen, ist das Newton-Verfahren. Die unzähligen Varianten dieses Verfahrens werden beispielsweise in Kelley [88, 89] oder Deuffhard [34] diskutiert. Hier soll nur die grundsätzliche Idee besprochen werden.

Ausgehend von einer bekannten Näherung \mathbf{x}_i soll eine bessere Näherung \mathbf{x}_{i+1} , im besten Fall die Lösung \mathbf{x} , gefunden werden. Dazu wird das nichtlineare Problem (3.0.1) in eine Taylor-Reihe entwickelt

$$\mathbf{f}(\mathbf{x}_{i+1}) = \mathbf{f}(\mathbf{x}_i) + \left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}_i} \Delta \mathbf{x}_i + \mathcal{O}(\Delta \mathbf{x}_i^2) = \mathbf{0} \quad (3.1.1)$$

mit der Gâteaux-Ableitung

$$\left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}_i} \Delta \mathbf{x}_i = \left. \frac{\partial \mathbf{f}(\mathbf{x}_i + \eta \Delta \mathbf{x}_i)}{\partial \eta} \right|_{\eta=0}. \quad (3.1.2)$$

Unter Vernachlässigung aller Terme mit höherer als linearer Potenz von $\Delta \mathbf{x}_i$ entsteht aus Gleichung (3.1.1) ein lineares Gleichungssystem für das Inkrement

$$\mathbf{J}(\mathbf{x}_i) \Delta \mathbf{x}_i = -\mathbf{f}(\mathbf{x}_i) \quad (3.1.3)$$

mit der Jacobimatrix

$$\mathbf{J}(\mathbf{x}_i) = \left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}_i}. \quad (3.1.4)$$

Die Jacobimatrix ist die Linearisierung des nichtlinearen Gleichungssystems an der aktuellen Position \mathbf{x}_i .

Nach dem Lösen des linearen Systems (3.1.3) ist die verbesserte Näherung mit

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta \mathbf{x}_i \quad (3.1.5)$$

gegeben. Kennzeichnend für das Newton-Verfahren ist das Lösen einer Reihe linearer Probleme (3.1.3) mit der gleichen Anzahl Unbekannter wie das nichtlineare Problem (3.0.1).

Das Newton-Verfahren konvergiert quadratisch, wenn die Anfangsschätzung \mathbf{x}_0 hinreichend nahe an der Lösung \mathbf{x} liegt und die Linearisierung (3.1.4) vollständig berechnet wird,

Algorithmus 2 Newton-Verfahren für nichtlineare Probleme.

wähle \mathbf{x}_0
für alle $i = 0, \dots, \max_{\text{nliter}}$:
 falls $|\mathbf{f}(\mathbf{x}_i)| \leq \varepsilon_{\text{nl}}$:
 fertig
 löse das lineare System $\mathbf{J}(\mathbf{x}_i) \Delta \mathbf{x}_i = -\mathbf{f}(\mathbf{x}_i)$
 bestimme den Faktor ω_i
 $\mathbf{x}_{i+1} = \mathbf{x}_i + \omega_i \Delta \mathbf{x}_i$

womit die Existenz der Linearisierung des nichtlinearen Problems vorausgesetzt wird. Die vielgerühmte quadratische Konvergenz bedeutet, daß der neue Fehler $\mathbf{x}_{i+1} - \mathbf{x}$ proportional zum Quadrat des alten Fehlers $\mathbf{x}_i - \mathbf{x}$ ist. Quarteroni u. a. [124] geben die Beziehung als

$$|\mathbf{x}_{i+1} - \mathbf{x}| \leq C |\mathbf{x}_i - \mathbf{x}|^2 \quad (3.1.6)$$

mit der Konstanten C an. Für praktische Berechnungen bedeutet die Beziehung (3.1.6), daß drei bis vier Iterationen genügen, um mit dem Newton-Verfahren eine sehr gute Näherung der Lösung \mathbf{x} zu berechnen.

Die Voraussetzungen für die quadratische Konvergenz (3.1.6) sind nicht immer erfüllt.

Wird die Linearisierung (3.1.4) im Newton-Verfahren nur approximiert oder das lineare System nur approximativ gelöst, steigt im Vergleich zum vollständig linearisierten Newton-Verfahren die Anzahl der benötigten Iterationen, um die gleiche Genauigkeit $|\mathbf{x}_{i+1} - \mathbf{x}|$ zu erreichen. Es existieren zahlreiche Varianten des Newton-Verfahrens, die als semi-, quasi-, pseudo- oder auch diskrete Newton-Verfahren bezeichnet werden und auf die eine oder andere Art eine Approximation der Linearisierung (3.1.4) einführen. Auf die im Kontext der Finite-Element-Simulationen wichtige Variante der Newton-Krylov-Verfahren wird in Abschnitt 3.4 eingegangen.

Die Approximation der Linearisierung (3.1.4) kann zur Divergenz des Verfahrens führen, so daß überhaupt keine Lösung gefunden wird. Solange das approximiert Newton-Verfahren aber konvergiert, konvergiert es unabhängig von der linken Seite in (3.1.3) stets zur Lösung des nichtlinearen Problems (3.0.1).

Die Forderung nach hinreichender Nähe der Anfangsschätzung \mathbf{x}_0 zur Lösung \mathbf{x} wird mathematisch durch den Konvergenzradius des Verfahrens beschrieben, innerhalb dessen die Anfangsschätzung liegen darf, damit das Newton-Verfahren die gesuchte Lösung findet. Befindet sich die Anfangsschätzung außerhalb des Konvergenzradius, können Line-Search-Verfahren eingesetzt werden, die das berechnete Lösungsinkrement $\Delta \mathbf{x}_i$ skalieren, bis eine Verbesserung des Residuums

$$|\mathbf{f}(\mathbf{x}_i + \omega_i \Delta \mathbf{x}_i)| < |\mathbf{f}(\mathbf{x}_i)| \quad (3.1.7)$$

erzielt wird. An dieser Stelle stellt das Lösungsinkrement $\Delta \mathbf{x}_i$ die Suchrichtung des Verfahrens dar, der Faktor ω_i bestimmt die Schrittweite. Die Addition (3.1.5) ist entsprechend um den Faktor ω_i zu ergänzen.

Da die Line-Search-Techniken auch als unabhängiges Lösungsverfahren für Fixpunkt-Formulierungen der nichtlinearen Probleme verwendet werden können, werden sie getrennt in Abschnitt 3.5 besprochen.

Das Newton-Verfahren zur Lösung des nichtlinearen Problems (3.0.1) ist in Algorithmus 2, einer Spezialisierung des allgemeinen Algorithmus 1, angegeben. In Algorithmus 2 ist un spezifiziert, wie das lineare System (3.1.3) gelöst und auf welche Weise der Line-Search-Faktor ω_i bestimmt wird. Die verschiedenen Möglichkeiten führen zu unterschiedlichen Lösungsalgorithmen.

3.2. Lösungsmethoden für lineare Gleichungssysteme

Ein wesentlicher Bestandteil des Newton-Verfahrens, oftmals der numerisch aufwendigste, ist das Lösen des linearen Gleichungssystems (3.1.3). Hier sollen deshalb Lösungsmethoden für lineare Systeme besprochen werden. Um die Notation an dieser Stelle etwas zu entlasten, soll im Folgenden das allgemeine lineare System

$$\mathbf{Ax} = \mathbf{b} \quad (3.2.1)$$

betrachtet werden.

Das Lösen von linearen Gleichungssystemen ist eine Aufgabe der linearen Algebra und wird daher in den einschlägigen Lehrbüchern wie Trefethen und Bau [154], Quarteroni u. a. [124] und Strang [146] besprochen. Eine detaillierte Darstellung verschiedener Lösertechniken wird beispielsweise in den Büchern von Meister [111] oder Saad [131] geboten. Wirklich umfangreiche Darstellungen lassen sich in den enzyklopädischen Werken von Zurmühl und Falk [169, 170] oder Golub und Van Loan [64] finden.

3.2.1. Direkte Lösungsverfahren. Zum Lösen des linearen Gleichungssystems (3.1.3) können direkte Verfahren verwendet werden, die beispielsweise in Meister [111] oder Quarteroni u. a. [124] beschrieben sind. Mit direkten Verfahren wird die exakte Lösung im Rahmen der Fließkommanumerik mit einer vorab bekannten Anzahl Operationen gefunden.

Die offensichtlichste direkte Lösungsmethode ist das Berechnen der Inversen \mathbf{A}^{-1} der Matrix \mathbf{A} des linearen Gleichungssystems (3.2.1) und das anschließende Multiplizieren des linearen Systems mit der Inversen. Das explizite Berechnen der Inversen ist jedoch eine sehr aufwendige Operation, die in der Numerik in aller Regel nicht ausgeführt werden kann. Es existieren sehr viel effizientere direkte Lösungsverfahren, die auf einer Zerlegung der Matrix \mathbf{A} in eine untere (\mathbf{L}) und eine obere (\mathbf{U}) Dreiecksmatrix mit Hilfe der Gauß-Elimination beruhen

$$\mathbf{A} = \mathbf{LU}. \quad (3.2.2)$$

Nach der LU-Zerlegung besteht das Lösen der Gleichung (3.2.1) aus dem zeilenweisen Einsetzen

$$\mathbf{x} = \mathbf{U}^{-1} \mathbf{L}^{-1} \mathbf{b}. \quad (3.2.3)$$

Die Inversen \mathbf{U}^{-1} und \mathbf{L}^{-1} werden also auch hier nicht explizit berechnet.

Die Nachteile der direkten Lösungsverfahren für lineare Systeme im Kontext von Finite-Element-Simulationen sind der große Speicherbedarf, die damit einhergehenden großen Rechenzeiten und die sehr geringe Skalierbarkeit auf parallelen Computersystemen. Sowohl die Zerlegung (3.2.2) als auch das Einsetzen (3.2.3) sind numerisch aufwendige Operationen.

Der Speicherbedarf der direkten Verfahren ist entscheidend, da die Matrix \mathbf{A} des linearen Gleichungssystems (3.1.3) bei Finite-Element-Simulationen eine sehr dünnbesetzte Matrix mit einer ausgeprägten Bandstruktur ist. Die Gauß-Elimination führt nun zum Auffüllen aller Matrixeinträge innerhalb der Bandstruktur. Bei sehr fein diskretisierten dreidimensionalen Geometrien können die so entstehenden zusätzlichen Matrixeinträge die Anzahl der in der ursprünglichen Matrix vorhandenen Einträge um ein Vielfaches übersteigen. Der Speicherbedarf steigt somit immens. Und mit der steigenden Anzahl Matrixeinträge erhöht sich auch die Anzahl der benötigten Rechenoperationen, um die Matrixeinträge zu bestimmen und weiterzuverwenden.

Sowohl die Zerlegung der Matrix \mathbf{A} als auch das Berechnen der Inversen \mathbf{A}^{-1} erlaubt es, das lineare System (3.1.3) für eine beliebige Anzahl rechter Seiten zu bestimmen. Wenn

sehr viele Lösungen eines linearen Systems gesucht werden, etwa im Zuge einer umfangreichen Eigenwertbestimmung, können direkte Lösungsmethoden trotz des hohen numerischen Aufwands für die Matrixzerlegung vorteilhaft sein.

Mit Hilfe von Gebietszerlegungsmethoden, siehe Smith u. a. [141], können parallele direkte Lösungsverfahren konstruiert werden. Wegen des Auffüllens der Bandmatrix und der damit einhergehenden Abhängigkeit der einzelnen Gleichungen des linearen Systems voneinander, erfordern parallele Versionen der direkten Lösungsverfahren einen sehr umfangreichen Datenaustausch zwischen den beteiligten Prozessoren. Auch die gleichmäßige Lastverteilung stellt eine Herausforderung dar. Bei einer großen Anzahl Prozessoren führt die aufwendige Kommunikation deshalb zu erheblichen Verzögerungen.

Insgesamt sind direkte Lösungsverfahren auf kleine bzw. moderat große Gleichungssysteme beschränkt. Im Kontext von FSI Simulationen, bei denen sehr große lineare Gleichungssysteme auftreten können, werden weniger aufwendige Lösungsverfahren benötigt.

3.2.2. Iterative Lösungsverfahren. Als Alternative zu den direkten Verfahren bieten sich iterative Lösungsverfahren an, die sich der Lösung schrittweise nähern. Der zum Lösen des Systems erforderliche Aufwand wird dabei auf Kosten der Genauigkeit der Lösung reduziert. Als Einstieg in dieses umfassende Thema ist beispielsweise die liebevolle Darstellung des Conjugate Gradient (CG) Verfahrens von Shewchuk [137] geeignet. Ebenfalls sehr hilfreich ist der Originalartikel zum Generalized Minimal Residual (GMRES) Verfahren von Saad und Schultz [132]. Ein Überblick über die Vielzahl der zur Verfügung stehenden Verfahren wird in Saad und van der Vorst [133] und auch in Simoncini und Szyld [140] gegeben. In zahlreichen Lehrbüchern, beispielsweise Meister [111], Quarteroni u. a. [124] oder Saad [131], werden die iterativen Lösungsverfahren im Detail vorgestellt. Implementierungen der wichtigsten Varianten findet man in Barrett u. a. [6].

Iterative Lösungsverfahren beruhen darauf, die Lösung \mathbf{x} der linearen Gleichung (3.2.1) durch schrittweise verbesserte Lösungsvorschläge \mathbf{x}_j anzunähern. Dabei soll der Fehler

$$\mathbf{e}_j = \mathbf{x} - \mathbf{x}_j \quad (3.2.4)$$

minimiert werden. Wenn dieser Fehler (3.2.4) bekannt wäre, wäre das System bereits gelöst. Von den iterativen Verfahren wird deshalb die lineare Abbildung

$$\mathbf{r}_j = \mathbf{A} \mathbf{e}_j = \mathbf{A} \mathbf{x} - \mathbf{A} \mathbf{x}_j = \mathbf{b} - \mathbf{A} \mathbf{x}_j \quad (3.2.5)$$

als zu minimierendes Residuum verwendet. Generell wird ein Startwert \mathbf{x}_0 gewählt und das Anfangsresiduum

$$\mathbf{r}_0 = \Delta \mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0 \quad (3.2.6)$$

bestimmt. Dann werden iterativ Inkremente $\Delta \mathbf{x}_j$ berechnet, mit denen die Lösungsvorschläge \mathbf{x}_j verbessert werden

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \Delta \mathbf{x}_j. \quad (3.2.7)$$

Da ein lineares System (3.2.1) gelöst wird, können die inkrementellen Residuen einzeln berechnet

$$\Delta \mathbf{r}_j = -\mathbf{A} \Delta \mathbf{x}_j \quad (3.2.8)$$

und nachträglich summiert werden

$$\mathbf{r}_j = \mathbf{b} - \mathbf{A} \mathbf{x}_j = \mathbf{r}_0 - \mathbf{A} \sum_{k=1}^j \Delta \mathbf{x}_k = \sum_{k=0}^j \Delta \mathbf{r}_k. \quad (3.2.9)$$

Das lineare System (3.1.3) gilt als hinreichend genau gelöst, wenn die Summe der inkrementellen Residuen hinreichend klein ist

$$|\mathbf{r}_j| = \left| \sum_{k=0}^j \Delta \mathbf{r}_k \right| \leq \varepsilon_{\text{lin}}, \quad (3.2.10)$$

der Lösungsvorschlag \mathbf{x}_j das lineare System (3.2.1) also hinreichend genau erfüllt.

Algorithmus 3 Beispielhaftes iteratives Lösungsverfahren für lineare Gleichungen.

wähle \mathbf{x}_0
 $\mathbf{r}_0 = \Delta \mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0$
für alle $j = 1, \dots, \max_{\text{liniter}}$:
 $\mathbf{a}_j = \mathbf{A} \Delta \mathbf{r}_{j-1}$
 berechne $\Delta \mathbf{x}_j$ und $\Delta \mathbf{r}_j$ aus $\Delta \mathbf{r}_k$ und \mathbf{a}_{k+1} mit $0 \leq k < j$
 $\mathbf{x}_{j+1} = \mathbf{x}_j + \Delta \mathbf{x}_j$
 $\mathbf{r}_{j+1} = \mathbf{r}_j + \Delta \mathbf{r}_j$
 falls $|\mathbf{r}_{j+1}| \leq \epsilon_{\text{lin}}$:
 fertig

In Algorithmus 3 ist ein iteratives Lösungsverfahren beispielhaft dargestellt. Die Ähnlichkeit zum nichtlinearen Lösungsalgorithmus 1 ist auffällig. Im Unterschied zum nichtlinearen Algorithmus 1 werden iterative lineare Löser jedoch meist in Inkrementen formuliert. Das Berechnen des linearen Inkrements $\Delta \mathbf{x}_j$ wird in Algorithmus 3 nicht dargestellt, da diese Berechnung von verschiedenen Verfahren unterschiedlich gehandhabt wird.

Die einzige Stelle, an der in Algorithmus 3 das lineare System (3.2.1) verwendet wird—von der Berechnung des Anfangsresiduums \mathbf{r}_0 abgesehen—ist die Berechnung des Matrix-Vektor-Produkts $\mathbf{A} \Delta \mathbf{r}_{j-1}$. Insbesondere wird die Systemmatrix \mathbf{A} des Systems (3.2.1) nicht verändert, es werden keine neuen Matrixeinträge hinzugefügt. Bei dünn besetzten Matrizen benötigen iterative lineare Löser daher zusätzlich zur Systemmatrix \mathbf{A} nur wenig Speicher.

Das wiederholte Matrix-Vektor-Produkt erzeugt implizit einen Krylov-Raum. Das ist ein linearer Vektorraum, dessen Basisvektoren \mathbf{b}_i jeweils Bilder einer lineare Abbildungen voneinander sind

$$\mathbf{b}_{i+1} = \mathbf{A} \mathbf{b}_i. \quad (3.2.11)$$

Der Startvektor \mathbf{b}_0 kann beliebig gewählt werden. Die Basisvektoren \mathbf{b}_i sind verschieden von der konstanten rechten Seite \mathbf{b} des linearen Systems (3.2.1).

Vektorräume werden in allen Lehrbüchern zur linearen Algebra wie Strang [146] und in großer Breite auch in Saad [131] besprochen. Der Krylov-Raum K_m ist durch den Startvektor \mathbf{b}_0 , die Matrix \mathbf{A} und die Anzahl m der Basen vollständig beschrieben

$$K_m(\mathbf{A}, \mathbf{b}_0) = \text{span}\{\mathbf{b}_0, \mathbf{A} \mathbf{b}_0, \dots, \mathbf{A}^{m-1} \mathbf{b}_0\}. \quad (3.2.12)$$

Die mathematischen Eigenschaften von Krylov-Räumen werden an dieser Stelle nicht betrachtet. Als Krylov-Verfahren werden iterative lineare Lösungsverfahren bezeichnet, die Verbesserungen der Näherungslösung des linearen Systems (3.2.1) innerhalb des Krylov-Raumes $K_m(\mathbf{A}, \mathbf{b}_0)$ suchen

$$\mathbf{x}_{j+1} - \mathbf{x}_0 = \sum_{k=0}^{m-1} \alpha_k \mathbf{A}^k \mathbf{b}_0 \quad (3.2.13)$$

mit jeweils passenden α_k .

Bemerkenswert ist, daß die Basisvektoren \mathbf{b}_i eines Krylov-Raumes im Allgemeinen nicht orthogonal zueinander stehen. Der Krylov-Raum wächst, solange ein neuer Basisvektor nicht als Linearkombination der vorhandenen Basisvektoren dargestellt werden kann

$$\mathbf{b}_m \notin K_m(\mathbf{A}, \mathbf{b}_0). \quad (3.2.14)$$

Offensichtlich kann die Dimension m des Krylov-Raumes die Länge n der Basisvektoren nicht übersteigen. Sobald der neue Basisvektor eine Linearkombination der existierenden Basisvektoren ist, kann auch die exakte Lösung des linearen Systems (3.2.1) als Linearkombination der Basen (3.2.13) angegeben werden. Aus diesem Grund existieren Krylov-Verfahren, die nach n Iterationen die exakte Lösung finden. Der große Vorzug der

Krylov-Verfahren ist es jedoch, oftmals schon nach wenigen Iterationen eine sehr gute Näherungslösungen zu finden.

Im Kontext dieser Arbeit werden ausschließlich Krylov-Verfahren als iterative Lösungsverfahren für lineare Gleichungssysteme eingesetzt.

Für eine ausführliche Darstellung der iterativen Lösungsmethoden für lineare Systeme sei Saad [131] empfohlen.

3.2.3. Das GMRES-Verfahren. Um den Lösungsprozess von linearen Gleichungssystemen zu verstehen, lohnt es sich, ein iteratives Lösungsverfahren im Detail anzusehen. Als Beispiel für ein Lösungsverfahren bietet sich das GMRES-Verfahren von Saad und Schultz [132] an.

Beim GMRES-Verfahren werden die Basisvektoren \mathbf{b}_i des Krylov-Raumes (3.2.12) orthonormiert. Auf diese Weise entstehen die orthonormalen Vektoren \mathbf{v}_i , die den gleichen Raum aufspannen wie die Basen \mathbf{b}_i

$$\text{span}\{\mathbf{b}_0, \mathbf{A}\mathbf{b}_0, \dots, \mathbf{A}^{m-1}\mathbf{b}_0\} = \text{span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\}, \quad (3.2.15)$$

wobei $\mathbf{v}_1 = \mathbf{b}_0 = \mathbf{r}_0 / \|\mathbf{r}_0\|_2$ ist. Jedes neue Residuumsinkrement $\Delta \mathbf{r}_j$ wird in die aus den vorhergehenden Residuumsinkrementen gewonnenen orthonormalen Basen \mathbf{v}_i mit $1 \leq i \leq j$ und einen neuen Basisvektor \mathbf{v}_{j+1} zerlegt. Jedes Residuumsinkrement $\Delta \mathbf{r}_j$ ist damit als Summe von orthonormalen Vektoren mit den Skalierungsfaktoren $h_{i,j}$ darstellbar

$$\Delta \mathbf{r}_j = \sum_{i=1}^{j+1} h_{i,j} \mathbf{v}_i. \quad (3.2.16)$$

Gleichzeitig wird der neue Basisvektor \mathbf{v}_{j+1} verwendet, um das nächste Residuumsinkrement zu bestimmen

$$\Delta \mathbf{r}_{j+1} = \mathbf{A} \mathbf{v}_{j+1}. \quad (3.2.17)$$

Nach k GMRES-Iterationen lassen sich die Faktoren $h_{i,j}$ zur $(k+1) \times k$ -Matrix $\bar{\mathbf{H}}_k$ zusammenfassen. Aus den $k+1$ Vektoren \mathbf{v}_i entsteht die $n \times (k+1)$ -Matrix \mathbf{V}_{k+1} , wobei n die Anzahl der Unbekannten im linearen System ist. Die Lösung des linearen Systems (3.1.3) wird vom GMRES-Löser approximiert, indem zur Startschätzung \mathbf{x}_0 die Linearkombination $\mathbf{V}_k \mathbf{y}_k$ der orthonormalen Vektoren \mathbf{v}_i hinzuaddiert wird

$$\mathbf{x}_k = \mathbf{x}_0 + \mathbf{V}_k \mathbf{y}_k. \quad (3.2.18)$$

Die Linearkombination $\mathbf{V}_k \mathbf{y}_k$ stellt die Summe aller Inkremente $\Delta \mathbf{x}_j$ dar. Die Faktoren der orthonormalen Vektoren \mathbf{v}_i , die als Vektor \mathbf{y}_k mit k Einträgen auftreten, werden im Sinne der Fehlerminimierung (3.2.10) durch die Bedingung

$$\|\mathbf{r}_0 - \mathbf{A} \mathbf{V}_k \mathbf{y}_k\| \rightarrow \min \quad (3.2.19)$$

bestimmt. Die Bedingung (3.2.19) läßt sich auf die Minimierung

$$\|\bar{\mathbf{H}}_k \mathbf{y}_k - \|\mathbf{r}_0\| (1, 0, \dots, 0)^T\| \rightarrow \min \quad (3.2.20)$$

mit dem mit $\|\mathbf{r}_0\|$ skalierten Einheitsvektor mit $k+1$ Einträgen $(1, 0, \dots, 0)^T$ zurückführen, siehe Saad und Schultz [132]. Im GMRES-Verfahren wird die Minimierung (3.2.20), die ein überbestimmtes lineares Gleichungssystem darstellt, mit Hilfe der Methode der kleinsten Quadrate gelöst.

Die orthonormalen Vektoren \mathbf{v}_i bilden demnach einen Raum, in welchem das Lösungsincrement $\mathbf{V}_k \mathbf{y}_k$ gesucht wird. Mit jeder Iteration wird dieser Raum um eine Richtung erweitert. Nach n Iterationen findet das GMRES-Verfahren deshalb, von numerischen Fehlern abgesehen, die exakte Lösung. Für jede Iteration wird ein neuer Vektor \mathbf{v}_i gespeichert und das um eine Zeile und Spalte gewachsene lineare System (3.2.20) gelöst. Somit steigen die Kosten im GMRES mit jeder Iteration.

Für praktische Berechnungen mit großen linearen Systemen können nur sehr viel weniger Iterationen $k \ll n$ ausgeführt werden. Das GMRES-Verfahren ist zum Bestimmen von Näherungslösungen geeignet, da das Residuum (3.2.10) oftmals schon in den ersten Iterationen stark minimiert wird. Sollte nach einer festgelegten Anzahl von k Iterationen die Abbruchbedingung (3.2.10) noch nicht erfüllt sein, wird das Verfahren mit der aktuellen Lösung \mathbf{x}_k als neuer Startschätzung \mathbf{x}_0 erneut angewendet. Dieser Neustart erhöht die Anzahl der insgesamt benötigten Iterationen, begrenzt jedoch den vom GMRES-Verfahren benötigten Speicher- und Zeitaufwand.

Die Nachteile des GMRES-Verfahrens sind der für iterative Verfahren hohe zusätzliche Speicherbedarf für die k Vektoren und der zusätzliche numerische Aufwand für das Lösen des linearen Systems (3.2.20). Es existieren deshalb alternative Lösungsverfahren für allgemeine Gleichungssysteme, wie das BiConjugate Gradient Stabilized (Bi-CGSTAB) Verfahren, die ohne das Speichern von Basisvektoren auskommen, dafür aber in manchen Fällen keine Lösung finden. Für Details hierzu siehe beispielsweise Meister [111], Saad [131].

Da das lineare System (3.2.20) in jedem GMRES-Iterationsschritt gelöst werden muß, ist in jedem Fall ein sehr effizienter Lösungsalgorithmus erforderlich. Das GMRES-Verfahren nutzt dabei den Umstand aus, daß die Matrix $\tilde{\mathbf{H}}_k$ eine Dreiecksmatrix mit einer Nebendiagonale ist, da die Vektoren \mathbf{v}_i orthonormal sind,

$$\mathbf{v}_j^T \mathbf{v}_i = 0 \quad \text{für } i \neq j, \quad (3.2.21)$$

und aus der orthonormalen Zerlegung (3.2.16) des Residuumsinkrements $\Delta \mathbf{r}_j$ die Beziehung

$$\Delta \mathbf{r}_j^T \mathbf{v}_i = (\mathbf{A} \mathbf{v}_j)^T \mathbf{v}_i = 0 \quad \text{für } i > j + 1 \quad (3.2.22)$$

folgt.

Wenn die Matrix \mathbf{A} symmetrisch ist, folgt aus (3.2.22) die Beziehung

$$(\mathbf{A} \mathbf{v}_j)^T \mathbf{v}_i = \mathbf{v}_j^T \mathbf{A}^T \mathbf{v}_i = \mathbf{v}_j^T \mathbf{A} \mathbf{v}_i = (\mathbf{A} \mathbf{v}_i)^T \mathbf{v}_j, \quad (3.2.23)$$

damit gilt (3.2.22) auch für $i < j - 1$. Folglich wird die Matrix $\tilde{\mathbf{H}}_k$ für symmetrische Matrizen \mathbf{A} zur Bandmatrix mit einer Bandbreite von drei. Entsprechend ist es möglich, ein iteratives Lösungsverfahren für symmetrische Gleichungssysteme zu konstruieren, ohne die Basen \mathbf{v}_i explizit speichern zu müssen und ohne das lineare System (3.2.20) zu lösen.

3.3. Vorkonditionierung

Die Konvergenzgeschwindigkeit der iterativen linearen Lösungsverfahren, also die Rate, mit der der Fehler (3.2.4) im Laufe der Iteration abnimmt, hängt sehr stark von den Eigenschaften der Matrix \mathbf{A} ab. Unter anderem spielt die Konditionszahl κ der Matrix \mathbf{A} , das Verhältnis des größten zum kleinsten Eigenwert, eine große Rolle. Generell weisen Systeme mit kleinen Konditionszahlen ein besseres Konvergenzverhalten auf. Zum Konvergenzverhalten von iterativen linearen Lösungsverfahren siehe beispielsweise Quarteroni u. a. [124] oder Saad [131].

Um die Konvergenzgeschwindigkeit von iterativen Lösungsmethoden zu verbessern, kann das lineare Gleichungssystem mit einem beliebigen linearen Operator skaliert werden. Da das Ziel der Skalierung darin besteht, die Konditionszahl der Matrix zu verringern, spricht man von Vorkonditionierung. Eine gute Vorkonditionierung ist in vielen Fällen die Voraussetzung, damit die iterativen Lösungsverfahren überhaupt anwendbar sind.

Die allgemein anwendbaren Techniken zur Vorkonditionierung werden deshalb in den Lehrbüchern zu iterativen Lösungsmethoden wie Meister [111] oder Saad [131] besprochen. Implementierungen der gängigen Methoden finden sich in Barrett u. a. [6]. Einen Überblick über die üblichen Vorkonditionierungsverfahren wird in Benzi [8] gegeben.

Für eine physikalische Motivation der Vorkonditionierung genügt es, ein beliebiges einheitenbehaftetes lineares System zu betrachten. Beispielsweise werden in der linearen Statik die Verschiebungen \mathbf{d} mit Hilfe der Steifigkeitsmatrix \mathbf{K} in Kräfte \mathbf{f} überführt

$$\mathbf{K}\mathbf{d} = \mathbf{f}. \quad (3.3.1)$$

Beim Anwenden des Lösungsalgorithmus 3 auf dieses System entsteht als Residuum die Kraftdifferenz

$$\mathbf{r}_j = \mathbf{f} - \mathbf{K}\mathbf{d}_j. \quad (3.3.2)$$

Der neue Lösungsvorschlag \mathbf{d}_{j+1} , der aus dem Residuum berechnet wird, ist jedoch eine Verschiebung. Der Sinn einer Vorkonditionierung ist es, die Einheiten passend zurückzurechnen.

Ein lineares System (3.2.1) kann von links und von rechts vorkonditioniert werden. Die linksseitige Vorkonditionierung entspricht dem Multiplizieren von links mit der Inversen \mathbf{M}_L^{-1} der Matrix \mathbf{M}_L

$$\mathbf{M}_L^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}_L^{-1}\mathbf{b}. \quad (3.3.3)$$

Die rechtsseitige Vorkonditionierung des linearen Systems (3.2.1) mit der Matrix \mathbf{M}_R führt auf

$$\mathbf{A}\mathbf{M}_R^{-1}\mathbf{M}_R\mathbf{x} = \mathbf{b}. \quad (3.3.4)$$

Beim linksseitigen Vorkonditionieren (3.3.3) wird die rechte Seite der linearen Gleichung skaliert, wodurch für den Konvergenztest nur mehr die vorkonditionierte rechte Seite $\mathbf{M}_L^{-1}\mathbf{b}$ zur Verfügung steht. Die Lösertoleranz muß deshalb relativ zum vorkonditionierten System gewählt werden. Beim rechtsseitigen Vorkonditionieren (3.3.4) wird im iterativen Lösungsverfahren nach der skalierten Unbekannten $\tilde{\mathbf{x}}$

$$\tilde{\mathbf{x}} = \mathbf{M}_R\mathbf{x} \quad (3.3.5)$$

gelöst und im Anschluß an das Lösen des linearen Systems zurückgerechnet

$$\mathbf{x} = \mathbf{M}_R^{-1}\tilde{\mathbf{x}}. \quad (3.3.6)$$

In beiden Fällen wird lediglich die Inverse einer Vorkonditionierungsmatrix \mathbf{M}^{-1} verwendet, nicht die Matrix \mathbf{M} selbst. Für die Vorkonditionierungs-Operation \mathbf{M}^{-1} ist es dabei gleichgültig, ob sie als linksseitiger oder als rechtsseitiger Vorkonditionierer eingesetzt wird. Im Folgenden wird die rechtsseitige Vorkonditionierung bevorzugt. Der rechtsseitig vorkonditionierte iterative Löser ist in Algorithmus 4 dargestellt.

Ein guter Vorkonditionierer \mathbf{M} ist eine möglichst exakte Approximation der Systemmatrix $\mathbf{M} \approx \mathbf{A}$, wobei die Matrix \mathbf{M} sehr viel leichter invertierbar sein sollte, da die Lösung

$$\mathbf{z} = \mathbf{M}^{-1}\mathbf{y}. \quad (3.3.7)$$

in jeder Iteration des Krylov-Lösers zu berechnen ist. Dabei muß weder die Matrix \mathbf{M} noch deren Inverse explizit vorliegen. Die Krylov-Verfahren benötigen lediglich einen linearen Operator, der das Matrix-Vektor-Produkt (3.3.7) darstellt. Da der Vorkonditionierungsoperator linear sein muß, ist er als Matrix \mathbf{M}^{-1} darstellbar, auch wenn es oft sinnvoll ist, die Matrix nicht explizit aufzustellen.

Die geforderte Linearität des Vorkonditionierungs-Operators \mathbf{M}^{-1} stellt eine erhebliche Einschränkung dar. Insbesondere können keine Krylov-Verfahren zum Vorkonditionieren von Krylov-Verfahren verwendet werden. Auch Näherungsverfahren mit variabler Iterationszahl sind als Vorkonditionierer nicht zulässig.

In besonderen Fällen kann es wünschenswert sein, diese Einschränkung zu vermeiden. Es existieren deshalb flexible Krylov-Verfahren wie das FGMRES-Verfahren von Saad [130], die variable Vorkonditionierer zulassen. Ein alternativer Ansatz ist das GMRESR-Verfahren von der Vorst und Vuik [161], de Sturler [150] aber auch van den Eshof u. a. [42], bei dem zwei Krylov-Verfahren ineinander geschachtelt werden. Siehe zu diesem Thema

Algorithmus 4 Rechtsseitig vorkonditionierter linearer Löser.

wähle $\tilde{\mathbf{x}}_0$
 $\mathbf{r}_0 = \Delta \mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{M}^{-1} \tilde{\mathbf{x}}_0$
für alle $j = 0, \dots, \max_{\text{Limiter}}$:
 $\mathbf{a}_j = \mathbf{A} \mathbf{M}^{-1} \Delta \mathbf{r}_j$
 berechne $\Delta \tilde{\mathbf{x}}_j$ und $\Delta \mathbf{r}_{j+1}$ aus $\Delta \mathbf{r}_k$ und \mathbf{a}_k mit $0 \leq k \leq j$
 $\tilde{\mathbf{x}}_{j+1} = \tilde{\mathbf{x}}_j + \Delta \tilde{\mathbf{x}}_j$
 $\mathbf{r}_{j+1} = \mathbf{r}_j + \Delta \mathbf{r}_{j+1}$
 falls $|\mathbf{r}_{j+1}| \leq \epsilon_{\text{lin}}$:
 fertig
 $\mathbf{x}_j = \mathbf{M}^{-1} \tilde{\mathbf{x}}_j$

auch den Abschnitt zu flexiblen Methoden mit variabler Vorkonditionierung in Simoncini und Szyld [139] und die darin enthaltenen Referenzen. Diese Verfahren sind Erweiterungen der üblichen Krylov-Verfahren und ermöglichen das Verwenden beliebiger Näherungsverfahren als Vorkonditionierer, sind aber generell aufwendiger als die gewöhnlichen Krylov-Verfahren. Flexible Krylov-Verfahren werden in dieser Arbeit nicht verwendet.

Für das Lösungsverhalten des Krylov-Verfahrens wäre der perfekte Vorkonditionierer die Systemmatrix selbst $\mathbf{M} = \mathbf{A}$, damit wird die Aufgabe aber lediglich in den Vorkonditionierer verschoben. Je schlechter andererseits die Näherung \mathbf{M} ist, desto weniger bewirkt die Anwendung des Vorkonditionierers (3.3.7) im linearen System (3.3.4). Der einfachste Vorkonditionierer, der nichts nützt, nichts schadet und keine Kosten verursacht, ist kein Vorkonditionierer $\mathbf{M} = \mathbf{I}$. Jeder Vorkonditionierer der eine schlechtere Approximation an die Matrix \mathbf{A} darstellt, ist offensichtlich eine dumme Idee.

Es existiert somit ein ganzes Spektrum an Vorkonditionierern, von perfekt bis wirkungslos. Unter diesen sind immer auch Verfahren, die für sich allein nicht ganz ausreichen, damit das iterative Lösungsverfahren konvergiert. Andere Vorkonditionierer führen zwar zu Konvergenz, aber mit sehr geringer Konvergenzgeschwindigkeit. In diesen Fällen kann es nützlich sein, den Vorkonditionierer im Rahmen einer relaxierten Richardson-Iteration mehrfach anzuwenden. Die Operation (3.3.7) in Gleichung (3.3.4) wird dazu durch die Iteration

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \omega \mathbf{M}^{-1} (\mathbf{y} - \mathbf{A} \mathbf{z}_k) \quad (3.3.8)$$

mit einem festen Parameter ω und ausgehend von $\mathbf{z}_0 = \mathbf{0}$ ersetzt.

Die Richardson-Iteration (3.3.8) stellt die Verbindung zwischen iterativen Lösungsverfahren und Vorkonditionierungsverfahren her und spielt damit eine zentrale Rolle, siehe auch die Diskussion in Quarteroni u. a. [124]. Zunächst ist das Richardson-Verfahren (3.3.8) ein eigenständiges iteratives Lösungsverfahren. Angewendet auf das lineare System (3.2.1) erhält man die Iterationsvorschrift

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \omega_j \mathbf{M}_j^{-1} (\mathbf{b} - \mathbf{A} \mathbf{x}_j), \quad (3.3.9)$$

bei der sowohl der Vorkonditionierer \mathbf{M}_j^{-1} als auch der Relaxationsparameter ω_j in jedem Iterationsschritt neu bestimmt werden können. Als Abbruchkriterium wird eine Residuumsnorm (3.2.10) verwendet.

Das Richardson-Verfahren (3.3.9) ist instationär, da sich die Iterationsvorschrift in jedem Schritt ändert und die Anzahl Iterationsschritte ebenfalls variabel ist. Instationäre Verfahren sind keine linearen Operatoren und damit als Vorkonditionierer nicht geeignet. Demgegenüber ist das Richardson-Verfahren (3.3.8) mit konstanter Vorkonditionierung und konstanter Iterationsanzahl ein stationäres Verfahren, das auch als Vorkonditionierungsverfahren einsetzbar ist.

Als Lösungsverfahren für lineare Gleichungssysteme kommt dem Richardson-Verfahren sehr geringe Bedeutung zu. Es existieren sehr viel effizientere Lösungsverfahren, siehe Abschnitt 3.2.2. Das stationäre Richardson-Verfahren (3.3.8) ist jedoch der Ausgangspunkt zahlreicher Vorkonditionierungsverfahren und ein wichtiger Baustein für die Konstruktion problemspezifischer Vorkonditionierer.

Ein Krylov-Verfahren mit Vorkonditionierer stellt demnach die Kombination zweier Lösungsverfahren dar. Von diesen zwei Verfahren übernimmt der Vorkonditionierer die Hauptaufgabe, die Lösung des linearen Systems könnte auch einfach durch vielfaches Anwenden des Vorkonditionierers (3.3.8) gefunden werden. Das Krylov-Verfahren hat demgegenüber lediglich die Aufgabe, den Lösungsprozeß zu beschleunigen.

Dieser Gedanke ist für die Konstruktion von Lösungsalgorithmen für zusammengesetzte Probleme von großer Bedeutung. Oftmals entstehen beim Modellieren physikalischer Vorgänge sehr spezielle lineare Systeme, die mit den bekannten Krylov-Verfahren gelöst werden können, solange ein geeigneter problemspezifischer Vorkonditionierer zur Verfügung steht. Auf diesem Ansatz basieren die in Kapitel 8.3 vorgestellten Vorkonditionierer für das FSI Problem.

Einige wichtige Vorkonditionierungsverfahren für einfache lineare Systeme sind im Folgenden aufgeführt.

3.3.1. Jacobi- und Gauß-Seidel-Vorkonditionierung. Zu den einfachsten Varianten des Richardson-Verfahrens zählen der Jacobi- und der Gauß-Seidel-Vorkonditionierer. Die Namen Jacobi bzw. Gauß-Seidel beschreiben dabei ein Funktionsprinzip, das auch außerhalb der Vorkonditionierung Verwendung findet. Im Kontext der Gebietszerlegung sind für diese Ansätze die Begriffe additiv und multiplikativ Schwarz gebräuchlich, siehe Smith u. a. [141].

In Quarteroni u. a. [124] und Saad [131] werden Jacobi- und Gauß-Seidel-Verfahren und deren Varianten als Lösungsmethoden vorgestellt.

Die Basis der Verfahren ist die Zerlegung der Matrix \mathbf{A} in die Diagonalmatrix \mathbf{D} sowie die untere Dreiecksmatrix $-\mathbf{L}$ und die obere Dreiecksmatrix $-\mathbf{U}$

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}. \quad (3.3.10)$$

Das Jacobi-Verfahren ist der einfachste Vorkonditionierer. Er folgt aus dem Richardson-Verfahren (3.3.8), indem $\omega = 1$ und $\mathbf{M} = \mathbf{D}$ gewählt wird. Die Berechnungsvorschrift lautet damit

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \mathbf{D}^{-1}(\mathbf{y} - \mathbf{A}\mathbf{z}_k). \quad (3.3.11)$$

Das Invertieren der Diagonalmatrix ist numerisch ohne Aufwand möglich.

Das Gauß-Seidel-Verfahren kann als Erweiterung des Jacobi-Verfahrens (3.3.11) verstanden werden, bei der statt der Diagonalen \mathbf{D} die gesamte untere Dreiecksmatrix $\mathbf{D} - \mathbf{L}$ als Vorkonditionierer verwendet wird

$$\mathbf{z}_{k+1} = \mathbf{z}_k + (\mathbf{D} - \mathbf{L})^{-1}(\mathbf{y} - \mathbf{A}\mathbf{z}_k). \quad (3.3.12)$$

Die Inverse der Dreiecksmatrix $\mathbf{D} - \mathbf{L}$ auf einen Vektor anzuwenden ist ein serieller Prozess, bei dem eine skalare Gleichung nach der anderen abgearbeitet werden muß, da es zu jedem Zeitpunkt genau eine skalare Gleichung mit einer Unbekannten gibt.

Da das Anwenden des Gauß-Seidel-Verfahrens ein serieller Prozess ist, kann das Verfahren nicht direkt als Vorkonditionierer bei einer parallelen Berechnung verwendet werden, wenn das Lösen des linearen Systems auf mehrere Prozessoren verteilt wird. In solchen Fällen, wenn der Vorkonditionierer nicht von sich aus parallel ausführbar ist, wird oftmals ein Block-Jacobi-Verfahren gewählt, wobei jeder Prozessor einen Block zugeordnet bekommt. Damit sind alle Prozessoren unabhängig voneinander und können jeweils für sich

beliebige Vorkonditionierungstechniken einsetzen. Der Nachteil an diesem Ansatz ist, daß das Ergebnis der Vorkonditionierung von der Anzahl der verwendeten Prozessoren abhängig ist.

Natürlich kann in den Iterationsvorschriften (3.3.11) und (3.3.12) analog zum Richardson-Verfahren (3.3.8) ein Relaxationsparameter $\omega \neq 1$ eingesetzt werden. Im Gauß-Seidel-Verfahren kann auch die untere Dreiecksmatrix unabhängig von der Diagonalen gewichtet werden $\mathbf{D} - \omega\mathbf{L}$. Damit erhält man relaxierte Varianten der entsprechenden Verfahren.

Als weitere Variante existiert das symmetrische Gauß-Seidel-Verfahren, bei dem nach einem Gauß-Seidel-Schritt (3.3.12) ein zweiter Richardson-Schritt ausgeführt wird, diesmal mit der oberen Dreiecksmatrix als Vorkonditionierer

$$\mathbf{z}_{k+2} = \mathbf{z}_{k+1} + (\mathbf{D} - \mathbf{U})^{-1} (\mathbf{y} - \mathbf{A}\mathbf{z}_{k+1}). \quad (3.3.13)$$

Natürlich kann auch dieser zweite Schritt relaxiert werden.

Das relaxierte Gauß-Seidel-Verfahren ist auch unter dem Namen Successive Overrelaxation (SOR) bzw. im symmetrischen Fall Symmetric Successive Overrelaxation (SSOR) bekannt.

3.3.2. Unvollständige Zerlegung. Eine Kategorie für sich bilden die Verfahren, die auf einer unvollständigen Zerlegung der Matrix \mathbf{A} beruhen. Die Idee dieser Verfahren ist es, die von direkten Lösungsmethoden bekannte Zerlegung der Matrix zu nutzen, allerdings das Auffüllen der dünn besetzten Matrix zu vermeiden oder wenigstens stark zu reduzieren. Das Berechnen der unvollständigen Zerlegung ist eine verhältnismäßig aufwendige Operation, dafür sind diese Verfahren oft sehr gute Vorkonditionierer.

Verschiedene Varianten dieser Verfahren entscheiden auf unterschiedliche Weise, welche Einträge zur Ausgangsmatrix hinzugefügt werden sollen. Allen diesen Verfahren gemeinsam ist die Beschränkung der unvollständigen Zerlegung auf einen Prozessor. Bei parallelen Berechnungen wird deshalb analog zur Gauß-Seidel-Vorkonditionierung eine Block-Jacobi-Aufteilung entsprechend der beteiligten Prozessoren vorgenommen und die unvollständige Zerlegung auf den jeweiligen Block beschränkt.

Die unvollständige Zerlegung der Matrix \mathbf{A} ist nur im Zuge der Vorkonditionierung relevant. Das Verfahrensprinzip wird kaum anderweitig eingesetzt. Deshalb sind die verwendeten Algorithmen an dieser Stelle von geringem Interesse. In Quarteroni u. a. [124] oder Saad [131] werden diese Verfahren ausführlich beschrieben.

3.3.3. Algebraische Mehrgitterverfahren. Mehrgitterverfahren sind Verfahren, die zum Lösen eines linearen Systems eine Hierarchie von Netzen verwenden. Das originale Netz, die ursprüngliche Diskretisierung der Differentialgleichung, wird dabei als feinstes Gitter verwendet, alle weiteren Netze sind passend gewählte, stufenweise Vergrößerungen. Mehrgitteransätze haben sich als kapriziös und sehr effizient herausgestellt.

Die geometrische Vorstellung einer Netzhierarchie ermöglicht die Diskussion der Mehrgitterverfahren im Kontext der mechanischen Problemstellungen. Für Berechnungen ist es jedoch unvorteilhaft, explizite Hierarchien von Netzen zu erzeugen, zumal die Netze zueinander passen müssen, um wirkungsvoll zu sein. Die Vergrößerung muß in nichtüberlappenden Aggregaten erfolgen, bei denen eine zusammenhängende Menge an Knoten im feinen Netz auf einen einzelnen Knoten im groben Netz abgebildet wird. Solche Netzhierarchien sind in aller Regel nicht verfügbar, deshalb werden algebraische Mehrgitterverfahren als Lösungsverfahren bevorzugt.

Die algebraischen Mehrgitterverfahren übertragen diese Mehrgitteridee auf algebraische Gleichungssysteme und verzichten auf die Konstruktion von geometrischen Netzen. Algebraische Mehrgitterverfahren sind deshalb vergleichsweise abstrakte Verfahren, jedoch sehr viel einfacher als Vorkonditionierungsverfahren einsetzbar.

Es existieren zahllose Varianten von algebraischen Mehrgitterverfahren, die auf unterschiedliche Problembereiche zugeschnitten sind. Eine umfassende Darstellung ist in Lehrbüchern wie Hackbusch [69], Briggs u. a. [14] oder Trottenberg u. a. [155] zu finden. Eine einfache Einführung in die grundlegende Idee liefert beispielsweise Falgout [44]. Eine Einführung am Beispiel der nichtlinearen Strukturmechanik wird in Gee [57] gegeben. Der Einsatz der algebraischen Mehrgitterverfahren beim Lösen von FSI Problemen wird in Gee u. a. [59] diskutiert.

Um das Verfahrensprinzip von Mehrgitterverfahren zu skizzieren, ist die Vorstellung einer tatsächlichen Gitterhierarchie hilfreich. Auf jeder Gitterebene besteht das Residuum des linearen Systems aus kurzwelligen und langwelligen Anteilen, jeweils bezogen auf die Gittergröße. Die kurzwelligen Residuumsanteile können mit vergleichsweise geringem numerischem Aufwand auf dem jeweiligen Gitter geglättet werden. Dazu kommen iterative Verfahren wie das Gauß-Seidel-Verfahren (3.3.12) zum Einsatz. Die langwelligen Anteile werden auf das nächstgrößere Gitter übertragen und sind dort wegen der größeren Abstände einfacher zu handhaben.

Beim algebraischen Mehrgitterverfahren wird für jede Gitterebene k ein linearer Interpolationsoperator \mathbf{P}_k , ein Vergrößerungsoperator \mathbf{R}_k und ein Glättungsverfahren $\hat{\mathbf{S}}_k$ benötigt. Mit Hilfe der Operatoren \mathbf{P}_k und \mathbf{R}_k kann aus der in Gitterebene k definierten Matrix \mathbf{A}_k die nächstgrößere Matrix \mathbf{A}_{k+1} bestimmt werden

$$\mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{A}_k \mathbf{P}_k. \quad (3.3.14)$$

Analog wird auch das Residuum und der Vektor der Unbekannten zwischen den Gittern übertragen. Mit dem Glättungsverfahren $\hat{\mathbf{S}}_k$ wird die Lösung \mathbf{z}_k auf der Gitterebene k verbessert

$$\hat{\mathbf{S}}_k(\mathbf{A}_k, \mathbf{y}_k, \mathbf{z}_k) \rightarrow \mathbf{z}_k. \quad (3.3.15)$$

Aus den drei Komponenten wird ein V-Zyklus zusammengesetzt, indem auf jeder Ebene das Residuum geglättet, das verbleibende Residuum auf der nächstgrößeren Ebene rekursiv bearbeitet, die Lösung aufsummiert und das Residuum erneut geglättet wird. Das Beispiel eines V-Zyklus mit drei Ebenen ist in Abbildung 3.3.1 angegeben.

Die Zahl der Ebenen richtet sich nach der Größe des zu lösenden Systems. Generell sind weitere Vergrößerungen unnötig, sobald das verbleibende lineare Problem effizient mit einem direkten Verfahren gelöst werden kann.

Die rekursive Funktion für einen einfachen V-Zyklus mit N_{levels} Ebenen ist in Algorithmus 5 angegeben. Entscheidend für den Erfolg der Mehrgitterverfahren ist, daß mit $\mathbf{R}_k \mathbf{r}_k$ nur das jeweils verbleibende Residuum auf die nächste Ebene übertragen wird. Entsprechend setzt sich die Lösung \mathbf{z}_k aus der Lösung der aktuellen Ebene und dem Anteil der nächstgrößeren Ebene

$$\mathbf{z}_k = \mathbf{z}_k + \mathbf{P}_k \mathbf{z}_{k+1} \quad (3.3.16)$$

zusammen.

Der hier angegebene V-Zyklus stellt das einfachste mögliche Mehrgitterverfahren dar. Die zahlreichen Varianten und Voraussetzungen der algebraischen Mehrgitterverfahren sollen an dieser Stelle nicht diskutiert werden. Entscheidend für die Effektivität der Mehrgitterverfahren ist die Wahl der Operatoren \mathbf{P}_k und \mathbf{R}_k passend zum Glättungsverfahren $\hat{\mathbf{S}}_k$ der jeweiligen Ebene, siehe beispielsweise Vaněk u. a. [157]. Darüber hinaus existieren zahlreiche Lösungsvorschläge für spezielle Probleme. Für das Lösen der Navier-Stokes-Gleichung werden in Sala und Tuminaro [135] Vorschläge gemacht. In Gee u. a. [58] wird

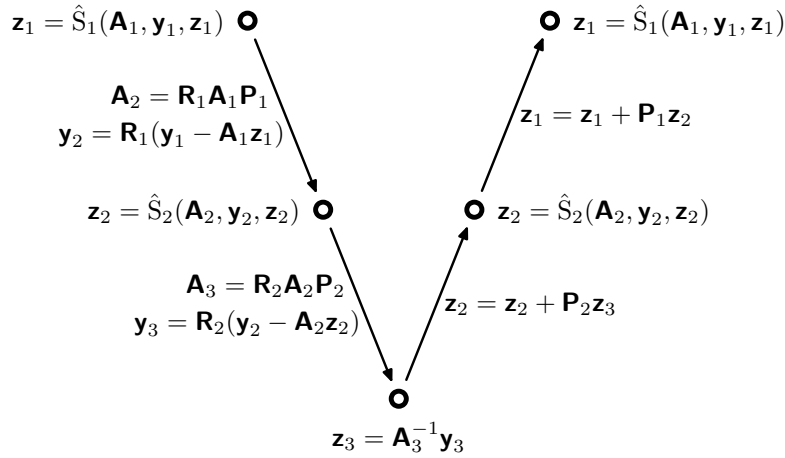


Abbildung 3.3.1: Beispielhafter V-Zyklus mit drei Ebenen.

Algorithmus 5 Rekursiver V-Zyklus mit N_{levels} Ebenen zum Lösen von $\mathbf{A}_1 \mathbf{z}_1 = \mathbf{y}_1$.

V-Zyklus($\mathbf{A}_k, \mathbf{y}_k, \mathbf{z}_k, k$):

falls $k \neq N_{\text{levels}}$

$$\mathbf{z}_k = \hat{\mathbf{S}}_k(\mathbf{A}_k, \mathbf{y}_k, \mathbf{z}_k)$$

$$\mathbf{r}_k = \mathbf{y}_k - \mathbf{A}_k \mathbf{z}_k$$

$$\mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{A}_k \mathbf{P}_k$$

$$\mathbf{z}_{k+1} = \mathbf{0}$$

$$\text{V-Zyklus}(\mathbf{A}_{k+1}, \mathbf{R}_k \mathbf{r}_k, \mathbf{z}_{k+1}, k+1)$$

$$\mathbf{z}_k = \mathbf{z}_k + \mathbf{P}_k \mathbf{z}_{k+1}$$

$$\mathbf{z}_k = \hat{\mathbf{S}}_k(\mathbf{A}_k, \mathbf{y}_k, \mathbf{z}_k)$$

sonst

$$\mathbf{z}_k = \mathbf{A}_k^{-1} \mathbf{y}_k$$

der Einfluß von inhomogenen Netzen untersucht. Eine Zusammenfassung der für die einzelnen Felder des FSI Problems verwendeten Operatoren ist in Gee u. a. [59] gegeben.

Der numerische Aufwand der algebraischen Mehrgitterverfahren ist im Idealfall proportional zur Anzahl der Unbekannten des linearen Systems. Weiterhin können alle algebraischen Operationen sehr gut parallelisiert werden, so daß die Mehrgitterverfahren unabhängig von der Anzahl der an der Berechnung beteiligten Prozessoren sind.

3.4. Newton-Krylov-Verfahren

Mit den drei vorangegangenen Abschnitten ist eine Familie von Verfahren zur Lösung des nichtlinearen Problems (3.0.1) beschrieben. Die Abschnitte bauen dabei aufeinander auf. Die in Abschnitt 3.1 vorgestellten Newton-Verfahren benötigten die Lösungsmethoden für lineare Gleichungssysteme aus Abschnitt 3.2. Iterative Krylov-Verfahren zur Lösung linearer Systeme wiederum benötigen die Vorkonditionierungstechniken aus Abschnitt 3.3. Der Aufbau des gesamten Newton-Krylov-Verfahrens ist in Abbildung 3.4.1 dargestellt.

Bei der Diskussion der Newton-Verfahren in Abschnitt 3.1 wurde allerdings eine exakte Lösung des linearen Gleichungssystems (3.1.3) vorausgesetzt. Wenn für das lineare System iterative Lösungsmethoden zum Einsatz kommen, liegt die lineare Lösung nur approximativ vor. Dies hat wiederum Auswirkungen auf das Newton-Verfahren. Wird zur Lösung

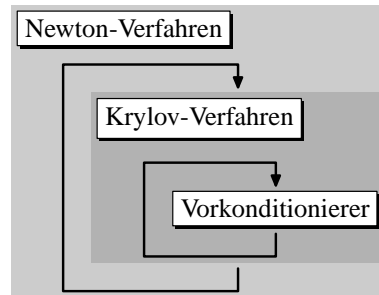


Abbildung 3.4.1: Aufbau des vorkonditionierten Newton-Krylov-Verfahrens.

des linearen Problems (3.1.3) ein Krylov-Verfahren verwendet, spricht man von Newton-Krylov-Verfahren. Die Abbruchkriterien des iterativen linearen Lösungsverfahrens und des Newton-Verfahrens beeinflussen sich gegenseitig und können nicht unabhängig voneinander gewählt werden. In Rheinboldt [128] und Deuffhard [34] werden die gegenseitigen Abhängigkeiten näher untersucht.

Das vorkonditionierte Newton-Krylov-Verfahren ist das in der Hauptsache verwendete Verfahren zur Lösung von nichtlinearen Feldgleichungen in Kapitel 4.

3.4.1. Jacobi-freie Newton-Krylov-Verfahren. Eine Besonderheit stellen Newton-Krylov-Verfahren dar, bei denen die Matrix \mathbf{J} des linearen Gleichungssystems (3.1.3) nicht explizit berechnet wird.

Solange das Produkt $\mathbf{J}\mathbf{M}^{-1}\Delta\mathbf{r}_{j-1}$ in Algorithmus 3 berechnet werden kann, ist die Matrix \mathbf{J} für den Krylov-Löser nicht erforderlich. Sollte die Linearisierung (3.1.4) des nichtlinearen Problems (3.0.1) nicht zur Verfügung stehen, können mit iterativen Lösern deshalb Jacobi-freie Newton-Krylov-Verfahren konstruiert werden, wie sie unter anderem in Knoll und Keyes [92] dargestellt sind. Im einfachsten Fall wird dabei das Matrix-Vektor-Produkt mit der finiten Differenz

$$\mathbf{J}(\mathbf{x}_i)\mathbf{M}^{-1}\Delta\mathbf{r}_{j-1} \approx \frac{\mathbf{f}(\mathbf{x}_i + \delta\mathbf{M}^{-1}\Delta\mathbf{r}_{j-1}) - \mathbf{f}(\mathbf{x}_i)}{\delta} \quad (3.4.1)$$

mit hinreichend kleinem δ approximiert.

Da die Funktion $\mathbf{f}(\mathbf{x})$ nichtlinear ist, ergeben sich beim Verwenden der Differenz (3.4.1) in Krylov-Lösern subtile Schwierigkeiten. Anstelle eines Tangentenproblems wird ein Sekantenproblem gelöst, so daß die vom Krylov-Löser bestimmte Lösung eine Näherung an die tatsächliche Lösung des linearen Problems (3.1.3) darstellt. Mit dem vorgestellten Algorithmus wird das aufsummierte Residuum (3.2.10) zwar wie gewünscht minimiert, aber beim Einsetzen der gefundenen Lösung bleibt dennoch ein Residuum (3.2.5).

Das Krylov-Verfahren setzt eine lineare Operation als Matrix-Vektor-Produkt voraus. Die Nichtlinearität der Funktion $\mathbf{f}(\mathbf{x})$ bleibt in der Approximation des Matrix-Vektor-Produkts durch die Differenz (3.4.1) jedoch erhalten. Die Residuumsnorm (3.2.10) in diesem Fall mit großer Genauigkeit zu erfüllen, ist deshalb unsinnig, das Krylov-Verfahren liefert stets eine Näherungslösung. Für das Newton-Verfahren wirkt diese Näherung als approximierter Linearisierung, entsprechend sind mehr nichtlineare Iterationen im Newton-Verfahren zu erwarten.

Zur Verwirrung der Leser wird bei der Verwendung der Approximation (3.4.1) in Newton-Krylov-Verfahren gelegentlich von nichtlinearen Krylov-Lösern gesprochen, siehe Feng u. a. [47].

Algorithmus 6 Fixpunkt-Verfahren für nichtlineare Probleme.

$\mathbf{x}_0 = \mathbf{0}$
für alle $i = 0, \dots, \max_{\text{nliter}}$:
 falls $|\mathbf{f}(\mathbf{x}_i)| \leq \varepsilon_{\text{nl}}$:
 fertig
 $\Delta \mathbf{x}_i = -\mathbf{f}(\mathbf{x}_i)$
 bestimme den Faktor ω_i
 $\mathbf{x}_{i+1} = \mathbf{x}_i + \omega_i \Delta \mathbf{x}_i$

Wenn die Matrix \mathbf{J} nicht explizit zur Verfügung steht, können die in Abschnitt 3.3 besprochenen Vorkonditionierungsverfahren nicht verwendet werden. Die Vorkonditionierung in Gleichung (3.4.1) ist deshalb auf weitere Informationen angewiesen. Das Fehlen allgemeiner Vorkonditionierungsansätze ist ein Nachteil der Jacobi-freien Newton-Krylov-Verfahren.

3.5. Fixpunkt-Verfahren

Mit Fixpunkt-Verfahren können Fixpunkt-Probleme gelöst werden. Ein Fixpunkt-Problem ist lediglich eine spezielle Formulierung eines allgemeinen Problems, bei dem die gesuchte Unbekannte als Ergebnis eines Funktionsausdrucks auftritt. So kann das nichtlineare Problem (3.0.1) einfach als Fixpunkt-Problem

$$\mathbf{x} = \mathbf{x} - \mathbf{f}(\mathbf{x}) \quad (3.5.1)$$

angeschrieben werden. Fixpunkt-Probleme können linear oder nichtlinear sein. Wird in ein Fixpunkt-Problem die Lösung \mathbf{x} eingesetzt, wird die Lösung reproduziert.

Alle hier besprochenen Lösungsverfahren können als Fixpunkt-Verfahren aufgefaßt werden. Wird die Bezeichnung Fixpunkt-Verfahren für alle Lösungsverfahren verwendet, fehlt die Abgrenzung gegen andere nicht-Fixpunkt-Verfahren und der Begriff wird inhaltsleer.

In dieser Arbeit sollen nur solche Verfahren als Fixpunkt-Verfahren bezeichnet werden, die von der Formulierung (3.5.1) explizit Gebrauch machen. Fixpunkt-Verfahren in diesem Sinne sind Verfahren, die eine gegebene Fixpunkt-Gleichung (3.5.1) mit einer Versuchslösung \mathbf{x}_i auswerten, um einen neuen Lösungsvorschlag zu erhalten. Eine Gemeinsamkeit aller Fixpunkt-Verfahren ist damit die Möglichkeit, die Lösungsschritte zu relaxieren. Der Begriff Fixpunkt-Verfahren wird somit auf iterative Lösungsverfahren beschränkt, die auf der Relaxation eines vergleichsweise einfach bestimmten Lösungsschritts beruhen.

Beispielsweise führt der Newton-Algorithmus 2 mit der radikalen Vereinfachung

$$\mathbf{J}(\mathbf{x}_i) \approx \mathbf{I} \quad (3.5.2)$$

auf das in Algorithmus 6 dargestellte Fixpunkt-Verfahren. Die Möglichkeit und gelegentliche Notwendigkeit eines Relaxationsschrittes im Rahmen des Newton-Verfahrens wurde bereits in Abschnitt 3.1 erwähnt. Im Fall der Approximation (3.5.2) ist jedoch kein lineares Gleichungssystem zu lösen, das Lösungsverfahren ist damit kein Newton-Verfahren mehr.

Im Gegensatz zum Newton-Verfahren wird beim Fixpunkt-Verfahren mit der Approximation (3.5.2) das Verhalten der Funktion $\mathbf{f}(\mathbf{x})$ an der aktuellen Position \mathbf{x}_i nicht berücksichtigt. Dadurch wird das Aufstellen und Lösen des linearen Problems (3.1.3) umgangen, pro Iteration fällt deshalb weniger Aufwand an. Andererseits werden im Allgemeinen jedoch sehr viel mehr Iterationen benötigt, da beim Bestimmen des Lösungssinkrements viel weniger Informationen über das nichtlineare Problem (3.0.1) zur Verfügung stehen.

Das Fixpunkt-Verfahren kann auch auf lineare Probleme angewandt werden. Wenn $\mathbf{f}(\mathbf{x})$ linear ist, also beispielsweise

$$\mathbf{f}(\mathbf{x}) = \mathbf{b} - \mathbf{A}\mathbf{x} \quad (3.5.3)$$

gilt, mit einem beliebigem Vektor \mathbf{b} und der Matrix \mathbf{A} , erhält man die Fixpunkt-Iteration

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \omega_i (\mathbf{b} - \mathbf{A} \mathbf{x}_i). \quad (3.5.4)$$

Mit einem zusätzlichen Vorkonditionierer \mathbf{M}^{-1} , der das Konvergenzverhalten der Iteration verbessert, entspricht das der schon besprochenen Richardson-Iteration (3.3.8).

In beiden hier betrachteten Fällen wird das aktuelle Residuum als Richtung für die Verbesserung verwendet

$$\Delta \mathbf{x}_i = -\mathbf{f}(\mathbf{x}_i). \quad (3.5.5)$$

Die Fixpunkt-Gleichung wird damit zu

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \omega_i \Delta \mathbf{x}_i. \quad (3.5.6)$$

Im Allgemeinen ist nicht sicher, daß das aktuelle Residuum $\mathbf{f}(\mathbf{x}_i)$ einen relevanten Anteil in Richtung der Lösung $\mathbf{x}_i - \mathbf{x}$ besitzt. Das ist der Fall, wenn in der Matrix $\mathbf{J}(\mathbf{x}_i)$ oder \mathbf{A} die Diagonale stark dominiert und somit die Annahme (3.5.2) oder $\mathbf{A} \approx \mathbf{I}$ eine vertretbare Näherung darstellt. Je schlechter die Näherung ist, desto kleiner ist der Faktor ω_i zu wählen, um den Lösungsvorschlag \mathbf{x}_i in Gleichung (3.5.6) zu verbessern. Je stärker die Relaxation ausfällt, desto geringer ist andererseits der Fortschritt im Iterationsschritt. Der Betrag des Parameters ω_i sollte deshalb stets so groß wie möglich und so klein wie nötig sein.

Um den Faktor ω_i zu bestimmen, existieren eine Reihe von Möglichkeiten, auf die im Folgenden kurz eingegangen werden soll.

3.5.1. Vorgegebener Parameter ω . Mit einem festen Parameter läßt sich das Divergieren des Lösungsverfahrens vermeiden, auch wenn der dazu nötige Parameter nicht a priori bekannt ist und gegebenenfalls durch Probieren gefunden werden muß. Gleichzeitig wird ein fester Parameter von dem schlechtesten Schritt bestimmt, der Parameter wird also für fast alle Iterationen zu klein sein. Folglich konvergiert das Fixpunkt-Verfahren mit einem festen Parameter ω sehr langsam und spielt als eigenständiges Lösungsverfahren keine Rolle.

Eine Anwendung für Fixpunkt-Verfahren mit konstantem Parameter ist die Richardson-Iteration (3.3.8) im Vorkonditionierer, bei der die erforderliche Linearität einen konstanten Parameter erzwingt. Außerdem gibt es Lösungsverfahren, die auf einer Serie von Lösungsvorschlägen \mathbf{x}_i beruhen. Um diese Serie zu berechnen, kann es erforderlich sein, das zugrunde liegende Fixpunkt-Problem (3.5.1) mit einem festen Relaxationsparameter auszuwerten. Das trifft im Besonderen auf die in Abschnitt 3.6 besprochene Vektorextrapolation sowie einige Varianten der in Abschnitt 3.5.4 besprochenen Aitken Δ^2 Verfahren zu.

3.5.2. Systematisches Probieren. Ein alternatives Verfahren, das vor allem als Line-Search-Schritt in Newton-Verfahren Verwendung findet, ist das Suchen eines geeigneten Parameters ω_i ausgehend von der Startschätzung $\omega_i = 1$. Solange das aktualisierte Residuum $\mathbf{f}(\mathbf{x}_i + \omega_i \Delta \mathbf{x}_i)$ das derzeitige Residuum $\mathbf{f}(\mathbf{x}_i)$ übersteigt

$$|\mathbf{f}(\mathbf{x}_i + \omega_i \Delta \mathbf{x}_i)| > |\mathbf{f}(\mathbf{x}_i)|, \quad (3.5.7)$$

wird der Wert von ω_i verringert und die Bedingung (3.5.7) erneut getestet. Damit kommt für die meisten Zeitschritte das quadratisch konvergierende Newton-Verfahren ohne Änderung zum Einsatz, während in numerisch schwierigen Situationen das Lösungsverfahren dennoch robust bleibt. Dieses Line-Search-Verfahren ist effizient, wenn das Auswerten des Fehlers $\mathbf{f}(\mathbf{x}_i + \omega_i \Delta \mathbf{x}_i)$ mit deutlich weniger Aufwand verbunden ist als das Lösen des linearen Systems (3.1.3).

Die Suche nach einem geeigneten Parameter ω_i kann durch Probieren in konstanten Schritten oder durch das Halbieren des verbleibenden Intervalls erfolgen. Alternativ kann nach dem Bestimmen von zwei Residuen $\mathbf{f}(\mathbf{x}_i + \omega_i \Delta \mathbf{x}_i)$ auch ein quadratisches Polynom entlang

der Schrittlänge $\Delta \mathbf{x}_i$ gebildet werden, dessen Minimum den Wert des Parameters ω_i bestimmt. Viele weitere Varianten sind möglich. Eine detaillierte Darstellung kann in Kelley [88, 89] gefunden werden.

Das Charakteristische an diesen Verfahren ist die nachträgliche Überprüfung der Bedingung (3.5.7). Dadurch wird ein Parameter ω_i gefunden, der die Residuumsnorm verringert, wenn die Suche nicht erfolglos abgebrochen wird. Im Unterschied dazu wählen alle anderen im Weiteren vorgestellten Verfahren den Parameter ω_i auf der Basis verfügbarer Informationen ohne die Reduktion der Residuumsnorm nachträglich zu überprüfen.

Line-Search-Verfahren werden hier nicht weiter ausgeführt, da sie in dieser Arbeit lediglich als Ergänzung des Newton-Verfahrens verwendet werden. Als eigenständiges Lösungsverfahren für FSI Probleme spielen sie wegen der mehrfachen und im FSI Kontext teuren Auswertung des Residuums $\mathbf{f}(\mathbf{x}_i + \omega_i \Delta \mathbf{x}_i)$ keine Rolle.

3.5.3. Gradientenverfahren. Das Gradientenverfahren, auch als Methode des steilsten Abstiegs bekannt, wird als Lösungsverfahren für lineare Gleichungssysteme von Shevchuk [137] beschrieben. Das Verfahren ist ein Fixpunkt-Verfahren mit einer speziellen Relaxationsvorschrift und kann auch für nichtlineare Probleme verwendet werden. Für FSI Berechnungen ist dieses Verfahren nur sehr eingeschränkt geeignet, da es Symmetrie der Systemmatrix \mathbf{J} voraussetzt. Dennoch wurde es in Le Tallec und Mouro [103] als Relaxationsverfahren vorgeschlagen und in Wall u. a. [163] sowie Mok und Wall [118] eingesetzt.

Beim Gradientenverfahren wird eine Annahme über das Verhalten des Residuums $\mathbf{f}(\mathbf{x}_i + \omega_i \Delta \mathbf{x}_i)$ entlang der Residuumsrichtung (3.5.5) getroffen. Aus dieser Annahme kann die für den aktuellen Schritt optimale Schrittweite gefunden werden. Dazu wird eine skalare Bewertungsfunktion $\phi(\mathbf{x}_i + \omega_i \Delta \mathbf{x}_i)$ angenommen, die für die gesuchte Schrittweite ω_i einen Extremwert besitzt. Mit passender Vorzeichendefinition ist der Extremwert ein Minimum, der Parameter ω_i ist dann die Lösung der Minimierungsaufgabe

$$\omega_i = \arg \min_{\omega_i} \phi(\mathbf{x}_i + \omega_i \Delta \mathbf{x}_i). \quad (3.5.8)$$

Unter der Bedingung, daß $\phi(\mathbf{x})$ hinreichend glatt ist, erhält man die Bedingung, um ω_i zu bestimmen

$$\frac{d\phi}{d\omega_i} = \frac{\partial \phi(\mathbf{x}_i + \omega_i \Delta \mathbf{x}_i)}{\partial \mathbf{x}} \cdot \Delta \mathbf{x}_i = 0. \quad (3.5.9)$$

Die Taylor-Entwicklung von $\phi(\mathbf{x})$ liefert unter Vernachlässigung von Termen höherer Ordnung

$$\phi(\mathbf{x}_i + \omega_i \Delta \mathbf{x}_i) \approx \phi(\mathbf{x}_i) + \omega_i (\phi'(\mathbf{x}_i))^T \Delta \mathbf{x}_i + \frac{\omega_i^2}{2} (\Delta \mathbf{x}_i)^T \phi''(\mathbf{x}_i) \Delta \mathbf{x}_i. \quad (3.5.10)$$

Das Ableiten der Approximation (3.5.10) nach ω_i und Einsetzen in Gleichung (3.5.9) liefert den optimalen Parameter ω_i

$$\omega_i = - \frac{(\phi'(\mathbf{x}_i))^T \Delta \mathbf{x}_i}{(\Delta \mathbf{x}_i)^T \phi''(\mathbf{x}_i) \Delta \mathbf{x}_i}. \quad (3.5.11)$$

Um die Ableitungen in Gleichung (3.5.11) berechnen zu können, muß eine konkrete Bewertungsfunktion angenommen werden. Mit der Annahme

$$\phi'(\mathbf{x}_i) = \Delta \mathbf{x}_i, \quad (3.5.12)$$

die gültig ist, solange das Residuum $\mathbf{f}(\mathbf{x}_i)$ der Gradient einer skalaren Funktion ist, kann der Parameter (3.5.11) berechnet werden. In der Folge erhält man eine symmetrische Jacobimatrix

$$\mathbf{J} = \frac{\partial \mathbf{f}(\mathbf{x}_i)}{\partial \mathbf{x}} = \phi''(\mathbf{x}_i) \quad (3.5.13)$$

und den optimalen Parameter

$$\omega_i = - \frac{(\Delta \mathbf{x}_i)^T \Delta \mathbf{x}_i}{(\Delta \mathbf{x}_i)^T \mathbf{J} \Delta \mathbf{x}_i}. \quad (3.5.14)$$

Die Bedingung (3.5.12) stellt eine starke Einschränkung dar. Ist die Bedingung nicht erfüllt, ist das Gradientenverfahren in aller Regel keine gute Wahl zum Bestimmen eines Relaxationsparameters.

Für die Matrix-Vektor-Multiplikation in Gleichung (3.5.14) kann auch die Approximation (3.4.1) verwendet werden, so daß die Jacobimatrix \mathbf{J} nicht explizit vorliegen muß.

3.5.4. Extrapolation mit Aitkens Δ^2 Verfahren. In der Definition der Schrittweite (3.5.14) wird die Tangente an die Funktion $\mathbf{f}(\mathbf{x}_i)$ verwendet, um das Verhalten der Funktion am Punkt \mathbf{x}_i zu bestimmen. Dazu ist das Matrix-Vektor-Produkt $\mathbf{J} \Delta \mathbf{x}_i$ auszuwerten. Alternativ kann das Verhalten der Funktion mit Hilfe von finiten Differenzen abgeschätzt werden. Aus diesem Gedanken ergeben sich Extrapolationsverfahren, die ohne das Auswerten der Tangente auskommen.

Ausgehend von einer konvergierenden Folge von drei Vektoren $\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}$ können zwei Lösungsvorschläge

$$\hat{\mathbf{x}}_i = \omega_i \mathbf{x}_i + (1 - \omega_i) \mathbf{x}_{i-1} = \mathbf{x}_{i-1} + \omega_i (\mathbf{x}_i - \mathbf{x}_{i-1}) = \mathbf{x}_{i-1} + \omega_i \Delta \mathbf{x}_{i-1} \quad (3.5.15)$$

$$\hat{\mathbf{x}}_{i+1} = \omega_i \mathbf{x}_{i+1} + (1 - \omega_i) \mathbf{x}_i = \mathbf{x}_i + \omega_i (\mathbf{x}_{i+1} - \mathbf{x}_i) = \mathbf{x}_i + \omega_i \Delta \mathbf{x}_i \quad (3.5.16)$$

gewonnen werden. Wenn die Folge konvergiert, finden beide Extrapolationen die gleiche Lösung

$$\lim(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{i+1}) = \mathbf{0}. \quad (3.5.17)$$

Da es sich jedoch um vektorielle Größen handelt, läßt sich die Differenz der Extrapolationen lediglich minimieren

$$|\hat{\mathbf{x}}_{i+1} - \hat{\mathbf{x}}_i| \rightarrow 0. \quad (3.5.18)$$

Die Annahme hinter den Extrapolationen (3.5.15) und (3.5.16) ist ein linearer Zusammenhang zwischen zwei aufeinander folgenden Vektoren der Folge.

Aus der Bedingung (3.5.17) folgt durch Einsetzen der Extrapolationen (3.5.15) und (3.5.16) der Parameter ω_i

$$\Delta \mathbf{x}_{i-1} = \omega_i (\Delta \mathbf{x}_{i-1} - \Delta \mathbf{x}_i). \quad (3.5.19)$$

Mit der Vektorinversen $\mathbf{v}^{-1} = \mathbf{v}/|\mathbf{v}|^2$ erhält man die Berechnungsvorschrift

$$\omega_i = \frac{(\Delta \mathbf{x}_{i-1})^T (\Delta \mathbf{x}_{i-1} - \Delta \mathbf{x}_i)}{|\Delta \mathbf{x}_{i-1} - \Delta \mathbf{x}_i|^2}. \quad (3.5.20)$$

Die Berechnungsvorschrift (3.5.20) kann genauso aus der Minimierung (3.5.18)

$$|\Delta \mathbf{x}_{i-1} - \omega_i (\Delta \mathbf{x}_{i-1} - \Delta \mathbf{x}_i)| \rightarrow 0 \quad (3.5.21)$$

mit Hilfe der Methode der kleinsten Quadrate abgeleitet werden.

Die originale Aitken Δ^2 Verfahren ist eine Methode zur Konvergenzbeschleunigung von Skalarfolgen. Die Definition (3.5.20) eingesetzt in der Relaxationsvorschrift (3.5.6) kann als Aitken Δ^2 Methode interpretiert werden, bei der die Skalarfolge aus der Vektorfolge durch Projektion in Richtung $\Delta \mathbf{x}_{i-1} - \Delta \mathbf{x}_i$ erhalten wurde.

Projektionen in andere Richtungen sind ebenso möglich, jedoch oft weniger natürlich. Die Definition (3.5.20) ist mithin eines der in MacLeod [104] bevorzugten Aitken Δ^2 Verfahren für Vektoren.

Als eine Alternative zur Vektorinversen wurde von Křížek u. a. [101] eine komponentenweise Extrapolation mit einer einfachen Mittelung vorgeschlagen und untersucht.

Der Nachteil von (3.5.20) ist, daß stets zwei Vektoren berechnet werden müssen, bevor einmal extrapoliert werden kann. Irons und Tuck [83] stellen deshalb als Variante von Aitkens Δ^2 Verfahren die Vektorversion der Sekantenmethode vor, die auf zwei unabhängigen Paaren $\mathbf{x}_{i-1}, \tilde{\mathbf{x}}_{i-1}$ und $\mathbf{x}_i, \tilde{\mathbf{x}}_i$ mit

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{f}(\mathbf{x}_i) \quad (3.5.22)$$

beruht. Die Gleichungen (3.5.15) und (3.5.16) ändern sich damit zu

$$\hat{\mathbf{x}}_i = \mathbf{x}_{i-1} + \omega_i(\tilde{\mathbf{x}}_{i-1} - \mathbf{x}_{i-1}) = \mathbf{x}_{i-1} + \omega_i \Delta \tilde{\mathbf{x}}_{i-1} \quad (3.5.23)$$

$$\hat{\mathbf{x}}_{i+1} = \mathbf{x}_i + \omega_i(\tilde{\mathbf{x}}_i - \mathbf{x}_i) = \mathbf{x}_i + \omega_i \Delta \tilde{\mathbf{x}}_i \quad (3.5.24)$$

wobei die Differenzen innerhalb der Paare gebildet werden

$$\Delta \tilde{\mathbf{x}}_i = \tilde{\mathbf{x}}_i - \mathbf{x}_i. \quad (3.5.25)$$

Über die Bedingung (3.5.18) folgt

$$\mathbf{x}_i - \mathbf{x}_{i-1} = \omega_i(\Delta \tilde{\mathbf{x}}_{i-1} - \Delta \tilde{\mathbf{x}}_i) \quad (3.5.26)$$

und durch das Multiplizieren mit der Vektorinversen

$$\omega_i = \frac{(\mathbf{x}_i - \mathbf{x}_{i-1})^T (\Delta \tilde{\mathbf{x}}_{i-1} - \Delta \tilde{\mathbf{x}}_i)}{|\Delta \tilde{\mathbf{x}}_{i-1} - \Delta \tilde{\mathbf{x}}_i|^2}. \quad (3.5.27)$$

Dieser Ausdruck kann unter Verwendung der Beziehung

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \omega_{i-1} \Delta \tilde{\mathbf{x}}_{i-1} \quad (3.5.28)$$

in

$$\omega_i = \omega_{i-1} \frac{(\Delta \tilde{\mathbf{x}}_{i-1})^T (\Delta \tilde{\mathbf{x}}_{i-1} - \Delta \tilde{\mathbf{x}}_i)}{|\Delta \tilde{\mathbf{x}}_{i-1} - \Delta \tilde{\mathbf{x}}_i|^2} \quad (3.5.29)$$

umgeformt werden. Der Parameter aus (3.5.29) ist dem Parameter aus (3.5.20) strukturell sehr ähnlich, kann aber in jedem Iterationsschritt berechnet werden. Entsprechend konvergiert das Fixpunkt-Verfahren mit dem Relaxationsparameter nach (3.5.29) etwa doppelt so schnell wie das Verfahren mit dem Parameter aus (3.5.20).

Die in Irons und Tuck [83] angegebene Formulierung zielt auf einen minimalen Speicher-verbrauch bei der Implementierung ab. Die Darstellung (3.5.29) wurde hier wegen ihrer Schlichtheit bevorzugt.

3.6. Vektorextrapolation

Die Vektorextrapolation kann als Verallgemeinerung des Fixpunkt-Verfahrens mit der Parameterdefinition (3.5.20) verstanden werden. Beim Fixpunkt-Verfahren wird die Schritt-richtung festgelegt und dann aus einer Folge von drei Lösungsvorschlägen die Schrittweite (3.5.20) bestimmt. Die Vektorextrapolation verwendet eine längere Folge und kann damit die Richtung und die Schrittweite gleichzeitig bestimmen.

Ausführlich werden die Verfahren in den Übersichtsartikeln von Smith u. a. [142, 143] und Jbilou und Sadok [86] vorgestellt. Alternative Vektorextrapolationsverfahren, die auf der Familie der ε -Algorithmen basieren, werden in Brezinski und Redivo Zaglia [13], Brezinski [12] vorgestellt, spielen im Kontext dieser Arbeit aber keine Rolle. Die Vektorextrapolationsverfahren zur Lösung von nichtlinearen Gleichungen wurden von Küttler und Wall [99] auf FSI Probleme angewandt, sind aber vornehmlich von akademischem Interesse.

Bei der Vektorextrapolation wird ausgehend von einer konvergierenden Folge von k Lösungsvorschlägen $\mathbf{x}_i, \mathbf{x}_{i+1}, \dots, \mathbf{x}_{i+k}$ die Lösung des nichtlinearen Problems (3.0.1) durch

$$\mathbf{x}_{i+k+1} = \sum_{j=0}^{k-1} \gamma_j \mathbf{x}_{i+j+1} \quad (3.6.1)$$

approximiert. Die Extrapolationsparameter γ_j werden dabei analog zur Bedingungsgleichung für den Fixpunkt-Parameter (3.5.18) durch die Minimierung von

$$\left| \sum_{j=0}^{k-1} \gamma_j (\mathbf{x}_{i+j+1} - \mathbf{x}_{i+j}) \right| = \left| \sum_{j=0}^{k-1} \gamma_j \Delta \mathbf{x}_{i+j} \right| \rightarrow \min \quad (3.6.2)$$

bestimmt. Die Minimierungsbedingung (3.6.2) folgt aus der Differenz zweier konvergierender Folgen. Dabei muß die Minimierung (3.6.2) die Nebenbedingung

$$\sum_{j=0}^{k-1} \gamma_j = 1 \quad (3.6.3)$$

erfüllen, damit die Extrapolation (3.6.1) sinnvoll ist.

Der Extrapolationsschritt (3.6.1) bildet eine Linearkombination der vorhandenen Lösungsvorschläge \mathbf{x}_{i+j} . Bei der Anwendung der Vektorextrapolation zur Lösung nichtlinearer Probleme sind deshalb mehrere Extrapolationsschritte erforderlich, um der Lösung schrittweise näher zu kommen. In Analogie zum Newton-Verfahren entspricht dabei jeder Extrapolationsschritt (3.6.1) dem Berechnen des Inkrements mit Hilfe einer approximierten Linearisierung.

Die Minimierung (3.6.2) mit der Nebenbedingung (3.6.3) führt für $k = 2$ auf die besprochene Aitken Δ^2 Extrapolation mit dem Relaxationsparameter (3.5.20).

Interessanter ist der Fall $k > 2$. Aus der Bedingung (3.6.2) erhält man ein lineares System mit k Unbekannten, mit dem die Faktoren γ_j bestimmt werden. Eine Möglichkeit, dieses lineare System zu erstellen, ist die Methode der kleinsten Quadrate. Allgemein ist jede Projektion in einen linear unabhängigen Unterraum mit k Basisvektoren denkbar. Das lineare System zum Bestimmen der Parameter γ_j besteht allgemein aus den k Gleichungen

$$\sum_{j=0}^{k-1} \gamma_j \alpha_{l,j} = \mathbf{0} \quad (3.6.4)$$

mit $l = 0$ bis $l = k - 1$ und der Nebenbedingung (3.6.3). Je nach Wahl der Koeffizienten $\alpha_{l,j}$ werden die Vektorextrapolationsverfahren unterschieden. In der Literatur werden vor allem drei Verfahren besprochen

- Minimal Polynomial Extrapolation (MPE)

$$\alpha_{l,j} = \Delta \mathbf{x}_{i+l} \cdot \Delta \mathbf{x}_{i+j} \quad (3.6.5)$$

- Reduced Rank Extrapolation (RRE)

$$\alpha_{l,j} = \Delta^2 \mathbf{x}_{i+l} \cdot \Delta \mathbf{x}_{i+j} \quad (3.6.6)$$

- Modified Minimal Polynomial Extrapolation (MMPE)

$$\alpha_{l,j} = \mathbf{y}_{\Gamma,l}^{n+1} \cdot \Delta \mathbf{x}_{i+j} \quad (3.6.7)$$

wobei die Vektoren $\mathbf{y}_{\Gamma,l}^{n+1}$ voneinander linear unabhängig und sonst beliebig sind.

Mit dem linearen Problem (3.6.4) und der Wahl der Faktoren $\alpha_{l,j}$ ist das Vektorextrapolationsverfahren vollständig.

In Algorithmus 7 wird das Verfahren für ein nichtlineares Problem aufgezeigt. Beim Berechnen der Folge von Lösungsvorschlägen wird ein konstanter Relaxationsparameter ω eingesetzt, um Konvergenz sicherzustellen.

Sidi [138] gibt eine Implementierung der MPE- und RRE-Verfahren an, die auf Krylov-Iterationen basiert und bemerkenswerte Analogien zum GMRES-Verfahren für lineare Gleichungssysteme zeigt. Eine Implementierung des MMPE-Verfahrens wurde von Jbilou und Sadok [85] angegeben.

Algorithmus 7 Vektorextrapolation für nichtlineare Probleme.wähle \mathbf{x}_0 **für alle** $i = 0, \dots, \max_{\text{nliter}}$:falls $|\mathbf{f}(\mathbf{x}_i)| \leq \varepsilon_{\text{nl}}$:**fertig****für alle** $j = 0, \dots, k - 1$:

$$\mathbf{x}_{i+j+1} = \mathbf{x}_{i+j} - \omega \mathbf{f}(\mathbf{x}_{i+j})$$

löse $\left| \sum_{j=0}^{k-1} \gamma_j \Delta \mathbf{x}_{i+j} \right| \rightarrow \min$ mit der Bedingung $\sum_{j=0}^{k-1} \gamma_j = 1$

$$\mathbf{x}_{i+k+1} = \sum_{j=0}^{k-1} \gamma_j \mathbf{x}_{i+j+1}$$

$$i = i + k + 1$$

3.7. Abbruchkriterien

Alle iterativen Verfahren benötigen ein Abbruchkriterium. Der Sinn des Abbruchkriteriums ist es, den Lösungsalgorithmus zu beenden, sobald die Lösung mit ausreichender Genauigkeit bestimmt wurde. Dabei hat das Abbruchkriterium keinen Einfluß auf die Konvergenz des Lösungsalgorithmus. Jeglicher Test, der zwischen einer ausreichend genauen Näherung und einer ungenügenden Näherung unterscheiden kann, ist als Abbruchkriterium möglich.

In der Darstellung der Algorithmen 1 und 4 wird beispielsweise die Norm des Residuums $|\mathbf{f}(\mathbf{x}_i)|$ und $|\mathbf{r}_j|$ verwendet. Alternativ könnte auch die Norm der Inkremente $|\Delta \mathbf{x}_i|$ bzw. $|\Delta^2 \tilde{\mathbf{x}}_{i,j}|$ verwendet werden. Sowohl Residuumsnormen als auch Inkrementnormen haben ihre Berechtigung.

Die Residuumsnorm ist in den meisten Fällen die natürliche Wahl, da das Residuum für die richtige Lösung gegen Null geht. In der Numerik kann Null jedoch eine schwer zu fassende Größe sein. Ein Beispiel ist die nichtlineare Strukturmechanik mit großer Struktursteifigkeit. Durch die große Steifigkeit treten bei Strukturverformungen sehr große innere Kräfte auf, so daß das Residuum als Differenz sehr großer Zahlen gebildet wird. Eine minimale Änderung der Strukturverschiebung bewirkt damit eine große Änderung des Residuums. Die Residuumsnorm ist also sehr viel sensitiver als die eigentliche Lösung. Auch wenn die Lösung schon mit großer Genauigkeit bekannt ist, ist die Residuumsnorm in einem solchen Fall noch weit von Null entfernt.

Eine anschauliche Vorstellung für diesen Fall ist die skalare Funktion $y = f(x)$ mit einem sehr steilen Nulldurchgang. Den Nulldurchgang numerisch exakt zu treffen, erfordert offensichtlich eine große Präzision. Im Fall der Strukturmechanik ist dies nur mit hohem numerischen Aufwand zu erreichen.

In einigen sehr unglücklichen Fällen kann auch die begrenzte Genauigkeit der Fließkommazahlen eine Rolle spielen, so daß die Genauigkeitsanforderung an die Residuumsnorm von der Numerik nicht erfüllt werden kann und der Lösungsalgorithmus nicht terminiert. Die Erfahrung zeigt jedoch, daß die aus den Fließkommazahlen resultierende Beschränkung in den wenigsten Fällen am Versagen einer Berechnung Schuld trägt. In aller Regel liegen die Ursachen für ausbleibende Konvergenz in der Abstimmung innerhalb des Lösungsalgorithmus.

Eine andere Komplikation kann analog zu extrem flachen, schleifenden Nulldurchgängen der skalaren Funktion $y = f(x)$ auftreten. Im Beispiel der Strukturmechanik kann die Residuumsnorm für einen weiten Bereich an Verschiebungen sehr klein sein. Beispielsweise sind Starrkörperverschiebungen bei großen Systemen, wenn nur einige wenige Auflagerbedingungen verletzt werden, in der L_2 -Norm schwer zu erkennen. Allgemeiner werden

langwellige Fehler von der L_2 -Norm unterschätzt. In solchen Fällen läuft der Lösungsalgorithmus, der sich nur auf die L_2 -Norm stützt, Gefahr, zu früh abzubrechen und ein unsinniges Ergebnis zur Lösung zu erklären.

Die Norm des Lösungsinkrements dagegen erkennt, ob im Lösungsschritt Fortschritte erzielt werden. Solange sich die Lösung im Lösungsschritt stark ändert, wurde die endgültige Lösung offenbar noch nicht gefunden. Umgekehrt kann es aber auch Plateaus im Lösungsverlauf geben, d.h. Iterationsschritte, in denen keine großen Fortschritte erzielt werden, obwohl die Lösung bei weitem noch nicht gefunden ist. Dergleichen ist vor allem von Krylov-basierten Lösungsmethoden für lineare Systeme bekannt. Die Norm des Lösungsinkrements neigt in diesen Fällen dazu, das Lösungsverfahren zu früh abzubrechen.

Als weitere Varianten können die Normen als absolute oder bezogene Größen verwendet werden. Als absolute Größen übernehmen die Normen, und damit die Toleranzen ε gegen die die Normen getestet werden, die Einheiten der zugrunde liegenden Gleichungen. Die Residuumsnorm beispielsweise hat als absolute Größe oftmals die Einheit einer Kraft, es sei denn, die zu lösende Gleichung wurde skaliert bzw. einheitenlos formuliert. Die Einheit der zugrunde liegenden Gleichung ist damit wichtig für die Wahl des Abbruchkriteriums.

Als relative Größe können die Normen auf das Anfangsresiduum $|\mathbf{f}(\mathbf{x}_i)| / |\mathbf{f}(\mathbf{x}_0)|$ oder das erste Inkrement $|\Delta\mathbf{x}_i| / |\Delta\mathbf{x}_0|$ bezogen werden. Die entsprechenden Toleranzen sind damit ebenfalls einheitenlos und drücken den im Lösungsschritt gewünschten relativen Fortschritt aus.

Weiterhin ist eine Skalierung der Normen mit der Dimension n des Unbekanntenvektors denkbar, um die gewählte Toleranz ε unabhängig von der gewählten Diskretisierung, d.h. unabhängig von der Anzahl Unbekannter, zu halten.

Zur Berechnung der Normen kommt schließlich neben der L_2 -Norm, die regelmäßig verteilte Fehler mißt, auch die auf den größten Fehler abzielende L_∞ -Norm in Frage. Speziell ist die L_∞ -Norm als Test für die langwelligen Fehler geeignet.

Aus diesen Möglichkeiten ist das jeweilige problemspezifische Abbruchkriterium zusammenzustellen. Dabei werden in aller Regel logische Verknüpfungen verschiedener Normen eingesetzt. Weiterhin sollten iterative Lösungsverfahren stets eine maximale zulässige Anzahl Iterationen vorgeben, um endloses Iterieren zu vermeiden.

Zusätzliche Schwierigkeiten entstehen bei zusammengesetzten Verfahren wie der Newton-Krylov-Methode, bei der eine Iteration auf der anderen aufbaut. Die innere Iteration, im Fall der Newton-Krylov-Methode also das Krylov-Verfahren, muß mit ausreichender Genauigkeit gelöst werden, damit das äußere Verfahren konvergieren kann. Für das Newton-Verfahren wurde diese Abhängigkeit beispielsweise in Rheinboldt [128] untersucht.

Fluid und Struktur als Einzelfeldprobleme

Bei der Behandlung von FSI Problemen müssen Fluid und Struktur gekoppelt werden. Sowohl in der Fluid- als auch in der Strukturdomäne sind dabei eigenständige Feldprobleme zu lösen. Das Ziel dieses Kapitels ist es, die mathematische Herleitung der einzelnen Feldprobleme zu umreißen und algorithmisch als Basis für die Lösungsmethoden des gekoppelten FSI Problems aufzubereiten.

4.1. Kontinuumsmechanik

Von alters her ist die Mechanik der Zweig der Physik, der sich mit Bewegungen und Verformungen von Festkörpern oder Fluiden beschäftigt. Zum historischen Abriss siehe beispielsweise Szabó [151]. Die Kontinuumsmechanik ist die mathematische Mechanik, die eine allgemeine Beschreibung der untersuchten Körper lediglich basierend auf der Kontinuumshypothese zum Ziel hat. Insbesondere werden geometrische Vereinfachungen vermieden. Das mechanische Problem wird in ein mathematisches Problem, eine Differentialgleichung, überführt. Das Lösen der Differentialgleichung stellt dann die Schwierigkeit dar. Die numerische Mechanik stellt Näherungsverfahren zum Lösen der kontinuumsmechanischen Differentialgleichungen bereit.

Es existiert eine große Fülle an Literatur zur Kontinuumsmechanik. Eine umfassende und vielzitierte Darstellung ist in Marsden und Hughes [106] zu finden. Als Einführung sei Gurtin [68] empfohlen. Ein Überblick über die Literatur kann an dieser Stelle nicht gegeben werden. Es sollen hier lediglich die Grundgedanken der Kontinuumsmechanik, basierend auf dem Vorlesungsskript Ulbricht [156], kurz angesprochen werden.

Die Differentialgleichung, die das mechanische Problem beschreibt, setzt sich aus vier Anteilen zusammen. Die Grundlage sind die Bilanzgleichungen, die physikalische Gesetzmäßigkeiten darstellen und im betrachteten Gebiet erfüllt sein müssen. Für die hier betrachteten Struktur- und Fluidprobleme ist die Massenbilanz und die Impulsbilanz relevant. Eine zweite Komponente ist die Kinematik, also die rein geometrische Beschreibung der Bewegung des Körpers, die beispielsweise die Verschiebungs-Verzerrungs-Beziehung liefert. Drittens beschreibt die Kinetik den Zusammenhang zwischen Kräften und Bewegungsgrößen. Und schließlich stellen die Materialgleichungen Annahmen über das Verhalten des Materials auf Basis der Kontinuumshypothese dar. Beispielsweise beschreibt in der elastischen Strukturmechanik die Materialgleichung den Zusammenhang zwischen Spannung und Verzerrung.

Die mathematische Beschreibung der physikalischen Zusammenhänge macht von Tensoren Gebrauch. Tensoren sind mathematische Objekte, die spezielle Transformationsregeln erfüllen. Bei weitem zugänglicher, und für eine anschauliche Vorstellung vollkommen ausreichend, ist die Auffassung von Tensoren als physikalische Objekte. Dabei wird hier nicht zwischen Tensoren und Tensorfeldern unterschieden. Alle im Folgenden verwendeten Tensoren beschreiben Felder, in aller Regel physikalische Felder, besitzen also an jedem Punkt im betrachteten Gebiet einen tensoriellen Wert.

Tensoren werden nach Stufen unterschieden, je nachdem, wie viele Raumrichtungen in die Tensorbeschreibung eingehen. So stellen etwa der Verschiebungstensor \mathbf{d} und der Spannungstensor $\boldsymbol{\sigma}$ ein Tensorfeld der ersten bzw. zweiten Stufe dar.

4.1.1. Betrachtungsweise und Konfiguration. Die Bewegung eines Körpers wird ausgehend von einem bekannten Referenzgebiet Ω_x beschrieben. Die Wahl der Ausmessung des Referenzgebiets ist beliebig. Entscheidend ist, daß jedem Punkt im betrachteten Gebiet eindeutige Koordinaten zugeordnet werden.

Beispielsweise kann das Referenzgebiet identisch mit dem Raumgebiet Ω_y sein, dem zeitunabhängigen raumfesten Gebiet mit unveränderlicher Ausmessung, innerhalb dessen die mechanischen Vorgänge betrachtet werden. Alternativ kann das Referenzgebiet identisch zum Materialgebiet Ω_z gewählt werden. Das Materialgebiet ist zeitabhängig und folgt der Bewegung des betrachteten Körpers. Die Ausmessung des Materialgebiets folgt dabei den Materiepunkten. D.h. das Materialgebiet wird von einem körperfesten System beschrieben, in dem jeder Materiepunkt mit unveränderlichen Koordinaten eindeutig bezeichnet wird.

Die Physik ist von der Wahl des Referenzgebiets, also des Koordinatensystems, unabhängig. Für Fluid- und Strukturprobleme sind jedoch unterschiedliche Betrachtungsweisen sinnvoll.

Für reine Fluidprobleme ist es meist zweckmäßig, das Referenzgebiet Ω_x identisch zum Raumgebiet Ω_y zu wählen. Man spricht dann von einer räumlichen, oder Eulerschen, Betrachtungsweise. In der räumlichen Betrachtungsweise werden alle Gleichungen in dem konstanten, raumfesten Koordinatensystem formuliert. Damit ändern die einzelnen Materiepunkte beständig ihre Position im Koordinatensystem, die Materiepunkte werden gleichsam weggespült. Diesem Konvektionseffekt, der eine Folge der Betrachtungsweise ist, muß in den Gleichungen Rechnung getragen werden.

Bei Strukturproblemen ist es dagegen üblich, das Referenzgebiet Ω_x identisch mit dem Materialgebiet Ω_z zu wählen. Damit befinden sich die Materiepunkte an festen Koordinaten und die physikalischen Gesetze können einfacher formuliert werden. Die Deformation des Materialgebiets für jeden Zeitpunkt $t > 0$ ist jedoch unbekannt. Die für einen beliebigen Zeitpunkt $t > 0$ im verformten Gebiet, der Momentankonfiguration, geltenden Gleichungen werden deshalb ins unverformte Gebiet, die Ausgangs- oder Referenzkonfiguration, zurücktransferiert.

Neben diesen beiden Spezialfällen kann das Referenzgebiet auch beliebig zeitlich veränderlich gewählt werden, so daß es weder mit dem Raumgebiet noch mit dem Materialgebiet zusammen fällt. Eine solche „Arbitrary Lagrangean Eulerian“ (ALE) Betrachtungsweise ist eine Möglichkeit, um das zeitveränderliche Fluidgebiet bei FSI Simulationen zu behandeln. Von dieser Möglichkeit wird im Rahmen dieser Arbeit ausgiebig Gebrauch gemacht.

4.2. Das Strukturfeld

Das Strukturfeld beschreibt die Mechanik des Festkörpers. An dieser Stelle wird die Differentialgleichung für einen elastischen Festkörper angegeben und daraus ein beispielhafter finite-Element-basierter Strukturalgorithmus abgeleitet. Weit umfangreichere Darstellungen von Lösungsmethoden für Strukturprobleme können Hughes [79], Crisfield [25, 24] oder auch Wriggers [167] entnommen werden. Die Literatur zu den Lösungsmethoden für Strukturprobleme ist unüberschaubar. Im Rahmen des hier angestrebten knappen Überblicks sollen die zitierten Werke zur Vertiefung genügen.

4.2.1. Kinematik. Die Kinematik hat etwas sehr Erfreuliches an sich, da sie rein geometrisch und deshalb wunderschön exakt ist. Die materielle Betrachtungsweise, auf der die Beschreibung der Strukturprobleme basiert, hat darüber hinaus einen sehr schönen Aufbau, der die dahinterliegende Mechanik spiegelt. Aus diesem Grund soll die Herleitung hier skizziert werden.

Um die Deformation eines Körpers zu beschreiben, wird jeder Materiepunkt einerseits im Ausgangssystem an der Ausgangsposition \mathbf{X} und andererseits in einem Momentansystem an der Momentanposition \mathbf{x} betrachtet. Die Verschiebung des Materiepunktes im Momentansystem ist

$$\mathbf{d} = \mathbf{x} - \mathbf{X}. \quad (4.2.1)$$

Damit geht auch das materielle Linienelement im Ausgangssystem $d\mathbf{X}$ in das Linienelement im Momentansystem $d\mathbf{x}$ über. Die Beziehung zwischen beiden wird als Deformationsgradient \mathbf{F} eingeführt

$$d\mathbf{x} = \mathbf{F} \cdot d\mathbf{X} \quad \text{und} \quad d\mathbf{X} = \mathbf{F}^{-1} \cdot d\mathbf{x}. \quad (4.2.2)$$

Der Deformationsgradient beschreibt die Längenänderung und Drehung von Linienelementen

$$\mathbf{F} = \mathbf{R} \cdot \mathbf{U} \quad (4.2.3)$$

mit dem orthonormalen Rotationstensor \mathbf{R} und dem rechten Strecktensor \mathbf{U} . Die Starrkörperverschiebung wird vom Deformationsgradient nicht abgebildet. Um die Starrkörperrotationen aus dem Deformationsgradienten zu entfernen, kann der symmetrische Rechts-Cauchy-Green-Tensor \mathbf{C}

$$\mathbf{C} = \mathbf{F}^T \cdot \mathbf{F} = \mathbf{U}^T \cdot \mathbf{R}^T \cdot \mathbf{R} \cdot \mathbf{U} = \mathbf{U}^T \cdot \mathbf{U} \quad (4.2.4)$$

eingeführt werden, bei dem der Rotationsanteil gerade verschwindet. Der Rechts-Cauchy-Green-Tensor \mathbf{C} ist ein Deformationstensor, der die Quadrate der Linienlemente beider Konfigurationen miteinander in Beziehung setzt

$$d\mathbf{x} \cdot d\mathbf{x} = d\mathbf{X} \cdot \mathbf{C} \cdot d\mathbf{X}. \quad (4.2.5)$$

Damit werden nicht nur Längenänderungen materieller Linienelemente, sondern auch Winkeländerungen zwischen Linienelementen, also Schubverformungen, abgebildet. Um ein Verzerrungsmaß zu erhalten, das im unverzerrten Zustand gerade $\mathbf{0}$ ist, wird der Lagrange-sche Verzerrungstensor \mathbf{E} eingeführt

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}). \quad (4.2.6)$$

Das ist die gesuchte Verzerrungs-Verschiebungs-Beziehung. Bemerkenswert ist, daß der Lagrange-sche Verzerrungstensor quadratisch von den Linienlementen abhängt. Bei den üblichen Ingenieurverzerrungen werden kleine Verzerrungen vorausgesetzt und die quadratischen Anteile können weggelassen werden.

4.2.2. Impulsbilanz. Die zeitliche Änderung des Impulses eines Körpers entspricht der Summe der auf den Körper wirkenden Kräfte. Diese Aussage gilt zunächst für den tatsächlichen deformierten Körper im zeitveränderlichen Materialgebiet Ω_z

$$\left. \frac{\partial}{\partial t} \right|_z \int_{\Omega_z} \rho^S \mathbf{u} d\Omega_z = \int_{\Gamma_z} \mathbf{t} d\Gamma_z + \int_{\Omega_z} \rho^S \mathbf{b} d\Omega_z. \quad (4.2.7)$$

Hier sind Dichte ρ^S , Geschwindigkeit \mathbf{u} und Volumenlast \mathbf{b} Funktionen der Position und der Zeit. Gleiches gilt für die Randlasten \mathbf{t} . Darüber hinaus ist auch das Integrationsgebiet Ω_z und der belastete Rand Γ_z zeitlich veränderlich, da sich der Körper verformt. Die Impulsbilanz (4.2.7) ist in der Momentankonfiguration angegeben.

Diese Bilanzgleichung kann ins Ausgangssystem Ω_z^0 geholt werden, das dem Referenzsystem Ω_x entspricht, um die Integration über ein bekanntes Gebiet ausführen zu können.

Das Integral über den Rand Γ_z wird mit dem Gaußschen Integralsatz in ein Gebietsintegral überführt. Durch Freischneiden erhält man dann als dynamisches Gleichgewicht am differentiellen Element die starke Form des Anfangs-Randwertproblems

$$\rho^0 \frac{d^2 \mathbf{d}}{dt^2} = \nabla_z \cdot (\mathbf{F} \cdot \mathbf{S}) + \rho^0 \mathbf{b}. \quad (4.2.8)$$

Der Ableitungsoperator ∇_z bezeichnet hier die Ableitungen im Materialgebiet Ω_z . Mit \mathbf{S} werden die zweiten Piola-Kirchhoff Spannungen bezeichnet.

Die Anfangsbedingungen setzen sich aus der Anfangsverschiebung \mathbf{d}^0 und der Anfangsgeschwindigkeit $\dot{\mathbf{d}}^0$

$$\mathbf{d}(0) = \mathbf{d}^0 \quad \text{und} \quad \dot{\mathbf{d}}(0) = \dot{\mathbf{d}}^0 \quad \text{im Gebiet } \Omega_z^0 \quad (4.2.9)$$

zusammen. Die Randbedingungen bestehen aus

$$\mathbf{d} = \bar{\mathbf{d}} \quad \text{auf dem Rand } \Gamma_{S,D}^0 \quad (4.2.10)$$

mit den vorgeschriebenen Verschiebungen $\bar{\mathbf{d}}$, sowie den Randlasten

$$\mathbf{N} \cdot \mathbf{S} = \mathbf{T} \quad \text{auf dem Rand } \Gamma_{S,N}^0. \quad (4.2.11)$$

Der Vektor \mathbf{T} bezeichnet die Randlasten in der Ausgangskonfiguration und \mathbf{N} den Normalenvektor auf dem Neumannrand $\Gamma_{S,N}^0$.

Die Ränder $\Gamma_{S,D}^0$ und $\Gamma_{S,N}^0$ bezeichnen den Dirichlet- bzw. den Neumann-Rand, überlappen sich nicht und decken zusammen den gesamten Rand des Referenzgebiets Ω_z^0 ab.

Die Massenbilanz ist in der materiellen Betrachtungsweise stets erfüllt und wird deshalb nicht weiter betrachtet.

4.2.3. Material. Den zweiten Piola-Kirchhoff Spannungstensor \mathbf{S} , der sich auf die Ausgangskonfiguration bezieht, erhält man aus den physikalischen Cauchy-Spannungen $\boldsymbol{\sigma}$, also den wahren in der Momentankonfiguration wirksamen Spannungen, durch die Rücktransformation

$$\mathbf{S} = \det \mathbf{F} \mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T}. \quad (4.2.12)$$

Es bleibt eine konstitutive Annahme über den Spannungs-Verzerrungs-Zusammenhang zu treffen. Hier soll als einfachstes mögliches Beispiel das lineare Saint-Venant-Kirchhoff-Material angegeben werden, das lediglich für kleine Verzerrungen verwendet werden kann. Die Beziehung zwischen dem zweiten Piola-Kirchhoff Spannungstensor und dem Lagrangeschen Verzerrungstensor lautet dann

$$\mathbf{S} = \lambda^S \text{tr} \mathbf{E} \mathbf{I} + 2\mu^S \mathbf{E}, \quad (4.2.13)$$

wobei die Lamé-Konstanten λ^S und μ^S aus Elastizitätsmodul E und Querkontraktion ν^S bestimmbar sind

$$\lambda^S = \frac{E\nu^S}{(1+\nu^S)(1-2\nu^S)} \quad \text{und} \quad \mu^S = \frac{E}{2(1+\nu^S)}. \quad (4.2.14)$$

Jede Materialgleichung stellt eine Annahme über das Verhalten des Materials dar, die mit Hilfe von experimentell bestimmten Parametern an das beobachtete Materialverhalten angepaßt werden muß. Diese Form der Materialmodellierung ist eine Folge der Kontinuumsannahme.

Mit der starken Form der Impulsbilanz (4.2.8) und der Materialdefinition (4.2.13) ist die Differentialgleichung der nichtlinearen elastischen Strukturmechanik vollständig.

4.2.4. Schwache Form. Wenn das Material Potentialcharakter besitzt, also die innere Arbeit unabhängig vom Weg ist, keine Reibung auftritt usw., kann die schwache Form als Variation des Energiepotentials bestimmt werden. Auf die Variationsprinzipien der Mechanik wird hier nicht weiter eingegangen. Eine sehr schöne Darstellung der Variationsprinzipien ist in Lanczos [102] enthalten.

Im Allgemeinen führt die Methode der gewichteten Residuen, die im Kontext der Strukturmechanik als das Prinzip der virtuellen Verrückungen bekannt ist, und das Umformen der Integrale mit doppelten Ableitungen der unbekanntes Verschiebung \mathbf{d} mit Hilfe des Gaußschen Integralsatzes zu der schwachen Form der Impulsbilanz

$$\int_{\Omega_z^0} \rho^0 \dot{\mathbf{d}} \cdot \delta \mathbf{d} d\Omega_z^0 + \int_{\Omega_z^0} \mathbf{S} : \delta \mathbf{E} d\Omega_z^0 = \int_{\Omega_z^0} \rho^0 \mathbf{b} \cdot \delta \mathbf{d} d\Omega_z^0 + \int_{\Gamma_N^0} \mathbf{T} \cdot \delta \mathbf{d} d\Gamma_N^0. \quad (4.2.15)$$

Durch den Gaußschen Integralsatz entsteht ein Randintegral über die äußeren Lasten. Die Wichtungsfunktion wurde hier bereits als virtuelle Verrückung $\delta \mathbf{d}$ bzw. als Variation der Lagrangeschen Verzerrungen $\delta \mathbf{E}$ erkannt.

Die Anforderungen an die Differenzierbarkeit der unbekanntes Verschiebung \mathbf{d} wurde auf Kosten der Differenzierbarkeit der beliebigen virtuellen Verrückung $\delta \mathbf{d}$ verringert. Das hat eine große Bedeutung für die möglichen Diskretisierungsansätze. Gleichzeitig erlaubt die schwache Form Lösungen, die in einzelnen Teilgebieten mit der Ausdehnung 0 die Differentialgleichung der starken Form nicht erfüllen. Es sind also einzelne Punkte oder Kanten im Gebiet möglich, an denen die starke Form verletzt wird, die jedoch vom Integral der schwachen Form nicht wahrgenommen werden. Damit werden Gebiete mit einspringenden Ecken und dergleichen möglich.

Jede Lösung der starken Form der Differentialgleichung erfüllt auch die schwache Form.

4.2.5. Diskretisierung. Für die Diskretisierung im Raum werden finite Elemente verwendet. Die Literatur zur Methode der finiten Elemente ist ausgesprochen umfangreich. Das Lehrbuch von Hughes [79] sei als eine mögliche Einführung erwähnt.

Das Wesen der Diskretisierung im Raum ist es, das durch die schwache Form definierte Verschiebungsfeld $\mathbf{d}(\mathbf{x}, t)$, das gleichsam unendlich viele Freiheitsgrade besitzt, in einen endlichdimensionalen Verschiebungsvektor $\mathbf{d}(t)$ zu überführen. Im Rahmen der Finite-Elemente-Methode werden im Gebiet Knoten eingeführt und zwischen den Knoten mit Hilfe von Lagrangepolynomen interpoliert. Diese Lagrangepolynome bilden die Matrix der Formfunktionen $\mathbf{N}^d(\mathbf{x})$. Die Diskretisierung im Raum kann als Produkt

$$\mathbf{d}(\mathbf{x}, t) \approx \mathbf{N}^d(\mathbf{x}) \mathbf{d}(t) \quad (4.2.16)$$

angegeben werden. Dabei kommen Polynome mit geringen Ansatzordnungen zum Einsatz, die jeweils nur in einem lokal begrenzten Gebiet verschieden von 0 sind. Die virtuelle Verschiebung $\delta \mathbf{d}$ wird analog zur realen Verschiebung diskretisiert, ebenso wie die Geschwindigkeit $\dot{\mathbf{d}}$, die Beschleunigung $\ddot{\mathbf{d}}$ und die Ortsvektoren \mathbf{x} .

Die Methode der finiten Elemente zeichnet sich dadurch aus, daß die Integrale der schwachen Form (4.2.15) stückweise auf Teilgebieten, den finiten Elementen, ausgeführt werden können. Die Elementintegration wird in aller Regel numerisch durch Gauß-Quadratur ausgeführt. Siehe beispielsweise Hamming [71] oder Press u. a. [123] zu Details der Gauß-Quadratur.

Durch die Assemblierung aller Elementbeiträge entsteht dann eine nichtlineare semidiscrete algebraische Gleichung

$$\mathbf{M} \ddot{\mathbf{d}} + \mathbf{f}_{\text{int}}(\mathbf{d}) = \mathbf{f}_{\text{ext}} \quad (4.2.17)$$

mit der Massenmatrix \mathbf{M} , dem nichtlinearen Vektor der inneren Kräfte $\mathbf{f}_{\text{int}}(\mathbf{d})$ und den äußeren Kräften \mathbf{f}_{ext} . Der Vektor der äußeren Kräfte \mathbf{f}_{ext} enthält die diskretisierten Integrale über Rand- und Volumenlasten, die hier als verschiebungsunabhängig angenommen werden.

Durch die identische Diskretisierung der virtuellen Verschiebung $\delta \mathbf{d}$ und der unbekanntenen Verschiebung \mathbf{d} entspricht die Zahl der Gleichungen gerade der Zahl der Unbekannten. Die Matrix \mathbf{M} ist quadratisch, symmetrisch und dünn besetzt.

Für Strukturprobleme mit elastischem Material stellt die Methode der finiten Elemente im Rahmen der gewählten Diskretisierung die beste mögliche Approximation an die kontinuierliche Lösung dar. Details dazu sind beispielsweise in Hughes [79] angegeben.

Zur Zeitdiskretisierung kommt das von Chung und Hulbert [21] vorgestellte generalized- α Verfahren zum Einsatz. Das generalized- α Verfahren ist eine Erweiterung des Newmark-Verfahrens. Die Interpolation des Newmark-Verfahrens mit den Parametern β und γ

$$\dot{\mathbf{d}}^{S,n+1} = \frac{\gamma}{\beta \Delta t} (\mathbf{d}^{S,n+1} - \mathbf{d}^{S,n}) - \frac{\gamma - \beta}{\beta} \dot{\mathbf{d}}^{S,n} - \frac{\gamma - 2\beta}{2\beta} \Delta t \ddot{\mathbf{d}}^{S,n} \quad (4.2.18)$$

$$\ddot{\mathbf{d}}^{S,n+1} = \frac{\gamma}{\beta \Delta t^2} (\mathbf{d}^{S,n+1} - \mathbf{d}^{S,n}) - \frac{1}{\beta \Delta t} \dot{\mathbf{d}}^{S,n} - \frac{1 - 2\beta}{2\beta} \ddot{\mathbf{d}}^{S,n} \quad (4.2.19)$$

mit den zeitlich diskretisierten Verschiebungen $\mathbf{d}^{S,n+1}$, Geschwindigkeiten $\dot{\mathbf{d}}^{S,n+1}$ und Beschleunigungen $\ddot{\mathbf{d}}^{S,n+1}$ wird dabei um Mittelungen der Variablen

$$\mathbf{d}^{S,n+1-\alpha_f} = (1 - \alpha_f) \mathbf{d}^{S,n+1} + \alpha_f \mathbf{d}^{S,n} \quad (4.2.20)$$

$$\dot{\mathbf{d}}^{S,n+1-\alpha_f} = (1 - \alpha_f) \dot{\mathbf{d}}^{S,n+1} + \alpha_f \dot{\mathbf{d}}^{S,n} \quad (4.2.21)$$

$$\ddot{\mathbf{d}}^{S,n+1-\alpha_m} = (1 - \alpha_m) \ddot{\mathbf{d}}^{S,n+1} + \alpha_m \ddot{\mathbf{d}}^{S,n} \quad (4.2.22)$$

anhand der zusätzlichen Parameter α_f und α_m ergänzt. Die Impulsgleichung wird an dem so gewonnenen generalisierten Mittelpunkt ausgewertet

$$\mathbf{M} \ddot{\mathbf{d}}^{S,n+1-\alpha_m} + \mathbf{f}_{\text{int}}^{S,n+1-\alpha_f} = \mathbf{f}_{\text{ext}}^{S,n+1-\alpha_f}. \quad (4.2.23)$$

Dabei kann nach Kuhl und Crisfield [94] der Vektor der inneren Kräfte entweder gleichfalls gemittelt

$$\mathbf{f}_{\text{int}}^{S,n+1-\alpha_f} = (1 - \alpha_f) \mathbf{f}_{\text{int}}^{S,n+1}(\mathbf{d}^{S,n+1}) + \alpha_f \mathbf{f}_{\text{int}}^{S,n} \quad (4.2.24)$$

oder direkt am Mittelpunkt ausgewertet werden

$$\mathbf{f}_{\text{int}}^{S,n+1-\alpha_f} = \mathbf{f}_{\text{int}}(\mathbf{d}^{S,n+1-\alpha_f}). \quad (4.2.25)$$

Siehe dazu auch Kuhl und Ramm [95].

4.2.6. Linearisierung. Die inneren Kräfte $\mathbf{f}_{\text{int}}^{S,n+1-\alpha_f}$ sind nichtlinear von den unbekanntenen Verschiebungen $\mathbf{d}^{S,n+1}$ abhängig

$$\mathbf{f}_{\text{int}}^{S,n+1-\alpha_f} = \mathbf{f}_{\text{int}}(\mathbf{d}^{S,n+1-\alpha_f}(\mathbf{d}^{S,n+1})) = \mathbf{f}_{\text{int}}^{S,n+1-\alpha_f}(\mathbf{d}^{S,n+1}). \quad (4.2.26)$$

Der Lösungsalgorithmus für das diskrete System (4.2.17) muß diese Nichtlinearität behandeln. In der Strukturdynamik haben sich die in Kapitel 3.1 beschriebenen Newton-Verfahren durchgesetzt. Dazu werden die residuellen Kräfte

$$\mathbf{f}_{\text{res}}^{S,n+1-\alpha_f} = \mathbf{f}_{\text{res}}^{S,n+1-\alpha_f}(\mathbf{d}^{S,n+1}) \quad (4.2.27)$$

eingeführt, die für die gesuchte Lösung gleich $\mathbf{0}$ sind

$$\mathbf{f}_{\text{res}}^{S,n+1-\alpha_f} = \mathbf{M} \ddot{\mathbf{d}}^{S,n+1-\alpha_m} + \mathbf{f}_{\text{int}}^{S,n+1-\alpha_f}(\mathbf{d}^{S,n+1}) - \mathbf{f}_{\text{ext}}^{S,n+1-\alpha_f} = \mathbf{0}. \quad (4.2.28)$$

Eine Taylor-Entwicklung um einen bekannten Lösungsvorschlag $\mathbf{d}_i^{S,n+1}$ liefert dann die Bestimmungsgleichung für die Inkremente $\Delta \mathbf{d}_i^{S,n+1}$

$$\mathbf{f}_{\text{res},i+1}^{S,n+1-\alpha_f} = \mathbf{0} = \mathbf{f}_{\text{res},i}^{S,n+1-\alpha_f} + \frac{\partial \mathbf{f}_{\text{res},i}^{S,n+1-\alpha_f}}{\partial \mathbf{d}^{S,n+1}} \Delta \mathbf{d}_i^{S,n+1} + \mathcal{O}\left(\left(\Delta \mathbf{d}_i^{S,n+1}\right)^2\right). \quad (4.2.29)$$

Werden die Terme höherer Ordnung ignoriert, bekommt man das lineare System

$$\frac{\partial \mathbf{f}_{\text{res},i}^{S,n+1-\alpha_f}}{\partial \mathbf{d}^{S,n+1}} \Delta \mathbf{d}_i^{S,n+1} = -\mathbf{f}_{\text{res},i}^{S,n+1-\alpha_f} \quad (4.2.30)$$

und mit dem Inkrement $\Delta \mathbf{d}_i^{S,n+1}$ die neue Verschiebung

$$\mathbf{d}_{i+1}^{S,n+1} = \mathbf{d}_i^{S,n+1} + \omega_i \Delta \mathbf{d}_i^{S,n+1}. \quad (4.2.31)$$

Das Newton-Verfahren für die Strukturdynamik besteht aus dem Lösen des linearen Gleichungssystems (4.2.30) und dem Aktualisieren der Verschiebung (4.2.31). Der Relaxationsfaktor wird mit $\omega_i = 1$ angenommen. Sollte ein Newton-Schritt zu einer Verschlechterung des Residuums führen (3.5.7), kann wie in Kapitel 3.5.2 beschrieben durch schrittweises Verringern von ω_i eine passende Schrittweise gefunden werden.

Die Ableitung in der Taylor-Entwicklung wird als Richtungsableitung berechnet

$$\frac{\partial \mathbf{f}_{\text{res},i}^{S,n+1-\alpha_f}}{\partial \mathbf{d}^{S,n+1}} \Delta \mathbf{d}_i^{S,n+1} = \left. \frac{\partial \mathbf{f}_{\text{res}}^{S,n+1-\alpha_f}(\mathbf{d}_i^{S,n+1} + \eta \Delta \mathbf{d}_i^{S,n+1})}{\partial \eta} \right|_{\eta=0}. \quad (4.2.32)$$

Das entspricht dem getrennten Ableiten nach jeder Komponente von $\mathbf{d}^{S,n+1}$, wobei wegen $\eta = 0$ alle quadratischen Terme und Terme höherer Ordnung aus der Ableitung entfernt werden. Das Ergebnis ist die Linearisierung von $\mathbf{f}_{\text{res},i}^{S,n+1-\alpha_f}$ an der Stelle $\mathbf{d}_i^{S,n+1}$ bezüglich der Verschiebung. Die Linearisierung der inneren Kräfte ergibt die Steifigkeitsmatrix \mathbf{K}_i

$$\mathbf{K}_i \Delta \mathbf{d}_i^{S,n+1} = \frac{\partial \mathbf{f}_{\text{int},i}^{S,n+1-\alpha_f}}{\partial \mathbf{d}^{S,n+1}} \Delta \mathbf{d}_i^{S,n+1}. \quad (4.2.33)$$

Der Massenterm ist bereits linear und bleibt bestehen.

4.2.7. Lösungsalgorithmus. Das nichtlineare Problem (4.2.28) wurde auf eine Serie von linearen Gleichungen (4.2.30) zurückgeführt. Diese linearen Gleichungen können mit einem der in Kapitel 3.2 besprochenen linearen Gleichungslöser behandelt werden. Bei fein aufgelösten Strukturproblemen mit vielen Unbekannten hat sich zum Lösen des linearen Gleichungssystems (4.2.30) das CG-Verfahren in Kombination mit einem Mehrgitter-Vorkonditionierer als sehr effektiv erwiesen.

Algorithmus 8 stellt einen möglichen Lösungsalgorithmus für die diskretisierten elastodynamischen Gleichungen dar. Der Algorithmus wurde bewußt einfach gehalten, um das Anwenden des Newton-Verfahrens für die Strukturgleichungen zu zeigen. Auf eine geschwindigkeitsproportionale Dämpfung und Line-Search Strategien wurde verzichtet, um die Darstellung einfach zu halten. Die externen Lasten wurden als verformungsunabhängig angenommen.

Für das Lösen von unabhängigen Strukturproblemen ist es sehr oft sinnvoll, einen extrapolierenden Prädiktor einzusetzen. Für die in Kapitel 8 vorgestellten monolithischen FSI Algorithmen ist dagegen ein konstanter Prädiktor erforderlich. In Algorithmus 8 wird deshalb nur der einfache konstante Prädiktor verwendet.

Den Kern des Algorithmus stellt die Newton-Schleife dar, die in jedem Zeitschritt bis zur Konvergenz des nichtlinearen System durchlaufen werden muß. Jeder Durchgang der Newton-Schleife benötigt das Aufstellen und Lösen eines linearen Problems. Zum Aufstellen des linearen Problems sind bei der Methode der finiten Elemente die Elementintegrale

Algorithmus 8 Nichtlinearer Strukturlöser mit generalized- α Zeitdiskretisierung.
Schleife über alle n Zeitschritte:

Konstanter Prädiktor

$$\mathbf{d}_0^{S,n+1} = \mathbf{d}^{S,n}, \quad \dot{\mathbf{d}}_0^{S,n+1} = \dot{\mathbf{d}}^{S,n}, \quad \ddot{\mathbf{d}}_0^{S,n+1} = \ddot{\mathbf{d}}^{S,n} \quad (4.2.34)$$

Bestimmen der Verschiebungen am Dirichletrand

$$\mathbf{d}_{D,0}^{S,n+1} \quad (4.2.35)$$

Bestimmen der externen Lasten (der Neumann-Randbedingungen)

$$\mathbf{f}_{\text{ext}}^{S,n+1} \quad (4.2.36)$$

Interpolation der Verschiebungen

$$\mathbf{d}_0^{S,n+1-\alpha_f} = (1 - \alpha_f)\mathbf{d}_0^{S,n+1} + \alpha_f\mathbf{d}^{S,n}, \quad \dot{\mathbf{d}}_0^{S,n+1-\alpha_f} = \dot{\mathbf{d}}^{S,n}, \quad \ddot{\mathbf{d}}_0^{S,n+1-\alpha_m} = \ddot{\mathbf{d}}^{S,n} \quad (4.2.37)$$

Interpolation der externen Lasten

$$\mathbf{f}_{\text{ext}}^{S,n+1-\alpha_f} = (1 - \alpha_f)\mathbf{f}_{\text{ext}}^{S,n+1} + \alpha_f\mathbf{f}_{\text{ext}}^{S,n} \quad (4.2.38)$$

Newton-Schleife mit $0 \leq i < \max_{\text{nlniter}}$ bzw. bis $|\mathbf{f}_{\text{res},i}^{S,n+1-\alpha_f}| \leq \epsilon^S$:

Aufstellen des effektiven linearen Gleichungssystems aus den inneren Kräften $\mathbf{f}_{\text{int},i}^{S,n+1-\alpha_f}$ und der zugehörigen Systemmatrix \mathbf{K}_i

$$\mathbf{f}_{\text{res},i}^{S,n+1-\alpha_f} = \mathbf{M}\ddot{\mathbf{d}}_i^{S,n+1-\alpha_m} + \mathbf{f}_{\text{int},i}^{S,n+1-\alpha_f} - \mathbf{f}_{\text{ext}}^{S,n+1-\alpha_f} \quad (4.2.39)$$

$$\mathbf{K}_{\text{eff},i} = (1 - \alpha_f)\mathbf{K}_i + \frac{(1 - \alpha_m)\gamma}{\beta\Delta t}\mathbf{M} \quad (4.2.40)$$

Einbringen der Dirichlet-Randbedingungen in das lineare System

$$\Delta\mathbf{d}_{D,i}^{S,n+1} = 0, \quad \mathbf{f}_{D,\text{res},i}^{S,n+1-\alpha_f} = 0, \quad \mathbf{K}_{DD,\text{eff},i} = \mathbf{I}, \quad \mathbf{K}_{D\Omega,\text{eff},i} = 0 \quad (4.2.41)$$

Lösen des linearen Systems

$$\mathbf{K}_{\text{eff},i}\Delta\mathbf{d}_i^{S,n+1} = -\mathbf{f}_{\text{res},i}^{S,n+1-\alpha_f} \quad (4.2.42)$$

Aktualisierung der Verschiebung, Geschwindigkeit und Beschleunigung für den Newton-Schritt

$$\mathbf{d}_{i+1}^{S,n+1-\alpha_f} = \mathbf{d}_i^{S,n+1-\alpha_f} + (1 - \alpha_f)\Delta\mathbf{d}_i^{S,n+1} \quad (4.2.43)$$

$$\dot{\mathbf{d}}_{i+1}^{S,n+1-\alpha_f} = \dot{\mathbf{d}}_i^{S,n+1-\alpha_f} + \frac{(1 - \alpha_f)\gamma}{\beta\Delta t}\Delta\mathbf{d}_i^{S,n+1} \quad (4.2.44)$$

$$\ddot{\mathbf{d}}_{i+1}^{S,n+1-\alpha_m} = \ddot{\mathbf{d}}_i^{S,n+1-\alpha_m} + \frac{1 - \alpha_m}{\beta\Delta t^2}\Delta\mathbf{d}_i^{S,n+1} \quad (4.2.45)$$

Aktualisierung der Verschiebung, Geschwindigkeit und Beschleunigung für den Zeitschritt

$$\mathbf{d}^{S,n+1} = \frac{1}{1 - \alpha_f}\mathbf{d}^{S,n+1-\alpha_f} - \frac{\alpha_f}{1 - \alpha_f}\mathbf{d}^{S,n} \quad (4.2.46)$$

$$\dot{\mathbf{d}}^{S,n+1} = \frac{1}{1 - \alpha_f}\dot{\mathbf{d}}^{S,n+1-\alpha_f} - \frac{\alpha_f}{1 - \alpha_f}\dot{\mathbf{d}}^{S,n} \quad (4.2.47)$$

$$\ddot{\mathbf{d}}^{S,n+1} = \frac{1}{1 - \alpha_m}\ddot{\mathbf{d}}^{S,n+1-\alpha_m} - \frac{\alpha_m}{1 - \alpha_m}\ddot{\mathbf{d}}^{S,n} \quad (4.2.48)$$

Algorithmus 9 Der vereinfachte Strukturalgorithmus.

Schleife über alle n Zeitschritte:

Die vorgeschriebenen Verschiebungen $\mathbf{d}_{D,0}^{S,n+1}$ ermitteln

Die externen Lasten $\mathbf{f}_{\text{ext}}^{S,n+1}$ berechnen

Newton-Schleife mit $0 \leq i < \max_n \text{nliter}$:

$$\begin{aligned} \mathbf{S}_{\Omega\Omega,i} \Delta \mathbf{d}_{\Omega,i}^{S,n+1} &= \mathbf{f}_{\Omega,\text{ext}}^{S,n+1-\alpha_f} - \mathbf{f}_{\Omega}^{S,n+1-\alpha_f}(\mathbf{d}_i^{S,n+1}) \\ \mathbf{d}_{\Omega,i+1}^{S,n+1} &= \mathbf{d}_{\Omega,i}^{S,n+1} + \Delta \mathbf{d}_{\Omega,i}^{S,n+1} \end{aligned}$$

auszuwerten und die dabei bestimmten Elementmatrizen in das globale System zu assemblieren. Die numerische Arbeit wird, in ungleichen Teilen, beim Aufstellen und beim Lösen dieses linearen Problems geleistet, in aller Regel überwiegt der Aufwand des linearen Lö-sers.

Besondere Aufmerksamkeit verdient das Einbringen der Dirichlet-Randbedingungen in das lineare Gleichungssystem. Dieses basiert auf der Zweiteilung des Verschiebungsvek-tors $\mathbf{d}_i^{S,n+1}$ und aller anderen Vektoren in die Freiheitsgrade im Gebiet $\mathbf{d}_{\Omega,i}^{S,n+1}$ und die Freiheitsgrade am Dirichlettrand $\mathbf{d}_{D,i}^{S,n+1}$. In diesem Kontext wird nur zwischen Dirichlet-freiheitsgraden D und Freiheitsgraden im Gebiet Ω unterschieden. Jeder Freiheitsgrad, der kein Dirichletfreiheitsgrad ist, gehört zum Inneren des Gebiets. Die effektive Steifigkeits-matrix $\mathbf{K}_{\text{eff},i}$ ist aus diesem Grunde viergeteilt

$$\mathbf{K}_{\text{eff},i} = \begin{bmatrix} \mathbf{K}_{\Omega\Omega,\text{eff},i} & \mathbf{K}_{\Omega D,\text{eff},i} \\ \mathbf{K}_{D\Omega,\text{eff},i} & \mathbf{K}_{DD,\text{eff},i} \end{bmatrix} \quad (4.2.49)$$

und die untere Zeile wird beim Einbringen der Dirichletbedingungen zu \mathbf{I} gesetzt

$$\mathbf{K}_{D\Omega,\text{eff},i} = \mathbf{0} \quad \text{und} \quad \mathbf{K}_{DD,\text{eff},i} = \mathbf{I}. \quad (4.2.50)$$

Algorithmus 8 stellt einen einfachen Strukturalgorithmus in allen Details dar. Für die Dis-kussion im Kontext der FSI Verfahren ist jedoch eine vereinfachte Sicht auf den Struktur-algorithmus wünschenswert. Diese wird in Algorithmus 9 gegeben. Hier wird die effektive Steifigkeit im Inneren des Gebiets mit $\mathbf{S}_{\Omega\Omega}$ bezeichnet. Die Details zur Zeitintegration oder zu den Dirichlet-Bedingungen entfallen. Übrig bleibt eine sehr einfache Newton-Schleife.

4.3. Das Fluidfeld

Das Fluidfeld beschreibt Flüssigkeiten oder Gase. Das Verhalten von Fluiden wird von der Navier-Stokes-Gleichung modelliert. Im Rahmen dieser Arbeit werden nur inkompressi-ble Strömungen betrachtet. Zur Diskretisierung des Fluidproblems kommen analog zum Strukturproblem finite Elemente zum Einsatz.

Das Modellieren und Lösen von Fluidproblemen mit Hilfe der Methode der finiten Eleme-nte wird sehr ausführlich in Gresho und Sani [65] besprochen. In Elman u. a. [38] werden Lösungs-algorithmen für Fluidprobleme vorgestellt. Eine Einführung in die Fluidstabilisie-rung wird in Donea und Huerta [35] gegeben. In Wall [162] wird die Fluidlösung mit stabi-lisierten finiten Elementen in allen Einzelheiten vorgestellt, Förster [50] enthält zahlreiche Anmerkungen zur Fluidstabilisierung. Über diese Arbeiten hinaus existiert natürlich eine große Fülle weiterer Literatur, die an dieser Stelle jedoch nicht besprochen werden kann.

4.3.1. Kinematik. Das Fluid wird in der räumlichen oder Eulerschen Betrachtungs-weise behandelt, da es die Rotationen im Fluid einerseits unmöglich machen, der Bewe-gung des Fluidkörpers mit einer Parametrisierung zu folgen und andererseits der Fluid-körper als Ganzes in aller Regel nicht von Interesse ist. Die Raumparmetrisierung der

Eulerschen Betrachtungsweise mit Ein- und Ausflußrand entspricht viel mehr dem experimentellen Aufbau.

In der Eulerschen Betrachtungsweise wird die materielle Zeitableitung, die in der Impulsbilanz auftritt, da sich die Impulsbilanz auf ein physikalisches Teilchen bezieht, durch die räumliche Zeitableitung und einen Konvektionsterm ausgedrückt. Die materielle Zeitableitung der Geschwindigkeit \mathbf{u} , also die partielle Zeitableitung im körperfesten Gebiet Ω_z , wird zu

$$\left. \frac{\partial \mathbf{u}}{\partial t} \right|_z = \left. \frac{\partial \mathbf{u}}{\partial t} \right|_y + \mathbf{u} \cdot \nabla_y \mathbf{u}. \quad (4.3.1)$$

Der Ableitungsoperator ∇_y bezieht sich dabei genauso auf das Raumgebiet Ω_y . Nachdem die Fluidgleichung immer im Raumgebiet betrachtet wird, wird das Gebiet im Folgenden nicht explizit angegeben.

Als weitere Größe wird der symmetrische Anteil des Geschwindigkeitsgradienten, die Deformationsgeschwindigkeit,

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \quad (4.3.2)$$

benötigt.

4.3.2. Massenbilanz. In der Eulerschen Betrachtungsweise spielt die Massenbilanz eine Rolle. Da hier inkompressible Strömungen betrachtet werden sollen, also Strömungen mit konstanter Dichte ρ^F , muß die Bilanz der Masse von Zufluß zum Gebiet und Abfluß gerade ausgeglichen sein. Mit der Normalen \mathbf{n} gilt am Gebietsrand für jeden Zeitpunkt

$$\int \mathbf{n} \cdot \mathbf{u} d\Gamma_y = 0. \quad (4.3.3)$$

Diese als Oberflächenintegral ausgedrückte Bedingung entspricht der Forderung, daß im Gebiet weder Quellen noch Senken vorhanden sind. Die Divergenz des Geschwindigkeitsfeldes muß

$$\nabla \cdot \mathbf{u} = 0 \quad (4.3.4)$$

sein im gesamten Gebiet Ω_y .

4.3.3. Impulsbilanz. Aus der allgemeinen Impulsbilanz in Integralform kann man wie bei der Struktur das dynamische Gleichgewicht am differentiellen Element ableiten. Durch den Übergang zur Eulerschen Betrachtungsweise geht der Konvektionsterm der materiellen Zeitableitung in die Gleichung ein. Die Impulsbilanz lautet damit

$$\rho^F \left. \frac{\partial \mathbf{u}}{\partial t} \right|_y + \rho^F \mathbf{u} \cdot \nabla \mathbf{u} = \nabla \cdot \boldsymbol{\sigma} + \rho^F \mathbf{b} \quad (4.3.5)$$

im Raumgebiet Ω_y . Hier ist $\boldsymbol{\sigma}$ die Cauchy-Spannung und \mathbf{b} die Volumenlast.

4.3.4. Material. Ruhende Fluide können keine Scherspannungen aufnehmen, weshalb sich der hydrostatische Spannungszustand $\boldsymbol{\sigma} = -\bar{p}\mathbf{I}$ einstellt, der lediglich von der Schwerkraft bestimmt ist. Bei bewegten Fluiden tritt Reibung auf und der Spannungszustand ist zusätzlich von der Deformationsgeschwindigkeit $\boldsymbol{\varepsilon}$ abhängig. Im einfachsten Fall einer linearen Abhängigkeit zwischen der Spannung und der Deformationsgeschwindigkeit spricht man von Newtonschen Fluiden. Die konstitutive Gleichung lautet dann

$$\boldsymbol{\sigma} = 2\mu\boldsymbol{\varepsilon}(\mathbf{u}) - \bar{p}\mathbf{I} \quad (4.3.6)$$

mit dem Druck \bar{p} und der dynamischen Viskosität μ .

4.3.5. Starke Form. Zusammen mit der dichtebezogenen kinematischen Viskosität

$$\nu = \mu / \rho^F \quad (4.3.7)$$

und dem kinematischen Druck

$$p = \bar{p} / \rho^F, \quad (4.3.8)$$

erhält man die Navier-Stokes-Gleichung für inkompressible Strömungen im Raumgebiet Ω_y , die dem hier betrachteten Fluidlöser zugrunde liegt.

$$\left. \frac{\partial \mathbf{u}}{\partial t} \right|_y + \mathbf{u} \cdot \nabla \mathbf{u} - 2\nu \nabla \cdot \varepsilon(\mathbf{u}) + \nabla p = \mathbf{b} \quad (4.3.9)$$

$$\nabla \cdot \mathbf{u} = \mathbf{0} \quad (4.3.10)$$

Als Anfangsbedingung benötigt die Navier-Stokes-Gleichung ein divergenzfreies Geschwindigkeitsfeld

$$\mathbf{u}(0) = \mathbf{u}^0 \quad \text{im Fluidgebiet } \Omega_y^0.$$

Die Randbedingungen sind Geschwindigkeitsvorgaben am Dirichletrand

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{auf dem Rand } \Gamma_{F,D}$$

und vorgegebene Randlasten auf dem Neumannrand

$$\frac{1}{\rho^F} \mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{t} \quad \text{auf dem Rand } \Gamma_{F,N}.$$

Auch im Fall des Fluidgebiets machen Dirichlet- und Neumannrand zusammen den gesamten Rand des Fluidgebiets aus. Dabei bezeichnet \mathbf{n} die Normale auf dem Rand und \mathbf{t} die Lasten.

Sowohl Anfangs- als auch Randbedingungen müssen die Divergenzfreiheit erfüllen, damit das Fluidproblem keinen Widerspruch enthält. Man spricht in diesem Fall von einem gut gestellten Problem. Das ist insbesondere eine Anforderung an die Dirichlet-Randbedingungen. Zum einen darf es keine Unstetigkeit zwischen Dirichletrand zum Zeitpunkt $t = 0$ und der Anfangsbedingung geben, zum anderen muß bei Fluidproblemen ohne Neumannrand, also Fluidgebiete die ausschließlich mit Dirichleträndern begrenzt sind, die Bilanz der vorgeschriebenen Strömungen über den Ein- und Ausflußrand exakt $\mathbf{0}$ sein.

4.3.6. Zeitlich veränderliche Fluidgebiete. Bei der FSI verschiebt sich das Interface zwischen Fluid und Struktur im Laufe der Berechnung, was für das Fluidgebiet eine Größenänderung oder, im Falle einer eingebetteten Struktur, eine Formänderung bedeutet.

Solange sich die Topologie des Fluidgebiets nicht ändert, besteht die einfachste Möglichkeit, diese Formänderung zu beschreiben, darin, eine deformierbare Ausmessung des Fluidgebiets zu finden. Die Verschiebung des Strukturrandes bewirkt dann eine Deformation der Ausmessung, das Fluidproblem wird in einer „Arbitrary Lagrangean Eulerian“ (ALE) Betrachtungsweise beschrieben. Details zur ALE Formulierung können Donea u. a. [36] entnommen werden. Auf die Behandlung der Netzbewegung als drittes Feld des FSI Problems wird in Abschnitt 4.4 eingegangen. Zur Lösung der Navier-Stokes-Gleichung mit einem ALE Netz siehe Duarte u. a. [37] oder auch die Anmerkungen in Förster [50].

Beim ALE Ansatz wird das Fluidgebiet als elastischer Körper behandelt, der sich vergleichbar zum Strukturfeld verformen kann, wobei dieser elastische Körper lediglich die Gebietsausmessung darstellt und die Navier-Stokes-Gleichung relativ zu dieser Ausmessung angegeben wird. Wegen der beliebig wählbaren Elastizität des Fluidgebiets spricht

man auch von einer ALE Betrachtungsweise, bei der die materielle Zeitableitung der Geschwindigkeit in einem zeitveränderlichen System zu berechnen ist. Die Navier-Stokes-Gleichung wird damit zu

$$\left. \frac{\partial \mathbf{u}}{\partial t} \right|_x + \mathbf{c} \cdot \nabla \mathbf{u} - 2\nu \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{u}) + \nabla p = \mathbf{b} \quad (4.3.11)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (4.3.12)$$

mit der konvektiven Geschwindigkeit $\mathbf{c} = \mathbf{u} - \mathbf{u}^G$ und der Deformationsgeschwindigkeit \mathbf{u}^G . Die Gleichung gilt im tatsächlichen Gebiet Ω_y . Der Ableitungsoperator ∇ bezieht sich ebenfalls auf das Raumgebiet Ω_y .

Daß die ALE Betrachtungsweise ohne Topologieänderungen auskommt, ist gleichermaßen Vor- und Nachteil. Topologieänderungen sind numerisch schwer zu beschreiben, das Lösen der Navier-Stokes-Gleichung auf einem beweglichen Netz stellt demgegenüber keine Schwierigkeit dar. Gleichzeitig ist der ALE Ansatz jedoch auf Fluidgebiete beschränkt, deren zeitliche Änderung als kontinuierliche Deformation dargestellt werden kann. Wenn sich die Gebietsdeformation mit dem ALE Ansatz nicht mehr darstellen lassen, ist eine Neuvernetzung erforderlich.

Alternative Techniken zum ALE Ansatz basieren auf einer festen Gebietsausmessung und variablen Integrationsgrenzen beim Lösen der Navier-Stokes-Gleichung. Die feste Gebietsausmessung entspricht der räumlichen Betrachtungsweise, die auch beim Lösen auf unveränderlichen Fluidgebieten üblich ist. Das Integrieren der Navier-Stokes-Gleichung in einem festen Gebiet stellt somit keine zusätzliche Schwierigkeit dar. Variable Integrationsgrenzen auf einer festen Gebietsausmessung darzustellen, erfordert dafür erheblichen Aufwand, die Gebietsränder müssen über aufwendige adaptive Randvernetzungen dargestellt werden. Dergleichen kann mit Hilfe des XFEM Ansatzes erreicht werden, wie er in Gerstenberger und Wall [63, 62] vorgestellt wird. Der XFEM Ansatz stellt eine Ergänzung der Methode der finiten Elemente dar, mit der Unstetigkeiten innerhalb von Elementen numerisch erfasst werden können.

Das Residuum \mathbf{f}_{res} der Navier-Stokes-Gleichung in ALE-Betrachtungsweise

$$\mathbf{f}_{\text{res}} = \mathbf{0} = \begin{bmatrix} \left. \partial \mathbf{u} / \partial t \right|_x + \mathbf{c} \cdot \nabla \mathbf{u} - 2\nu \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{u}) + \nabla p - \mathbf{b} \\ \nabla \cdot \mathbf{u} \end{bmatrix} \quad (4.3.13)$$

wird im Folgenden zur Stabilisierung der Navier-Stokes-Gleichung verwendet.

Die hier betrachtete Herleitung der Navier-Stokes-Gleichung, und damit die hier verwendete Formulierung des konvektiven Terms, geht vom Impulserhalt für jeden materiellen Punkt aus. Alternativ kann die Navier-Stokes-Gleichung als Bilanz über Kontrollvolumina aufgeschrieben werden. Der konvektive Term tritt dabei in der Divergenz-Formulierung auf. Im Kontinuierlichen sind beide Formulierungen gleichwertig.

Im Zusammenhang mit der ALE-Betrachtungsweise wird die Auswertung der Divergenz-Formulierung von der variablen Divergenz der Gebietsausmessung verkompliziert. Die geometrische Erhaltung tritt als zusätzliche Beschränkung der Netzbewegung auf. Details zur geometrischen Erhaltung und den Formulierungen der Navier-Stokes-Gleichung sind in Förster u. a. [51] sowie Förster [50] angegeben.

4.3.7. Schwache Form. Die Navier-Stokes-Gleichung muß in die schwache Form überführt werden. Dabei ist sowohl die Geschwindigkeit \mathbf{u} als auch der Druck p eine Unbekannte. Entsprechend werden die Wichtungen \mathbf{v} und q eingeführt. Der Spannungsterm wird mit Hilfe des Gaußschen Integralsatzes so umgeformt, daß das dabei entstehende Randintegral ein Integral über physikalische Kräfte ist

$$\mathbf{t} = \frac{1}{\rho^F} \mathbf{n} \cdot \boldsymbol{\sigma}. \quad (4.3.14)$$

Die schwache Form der Navier-Stokes-Gleichung ergibt sich damit zu

$$\begin{aligned} \int \dot{\mathbf{u}} \cdot \mathbf{v} d\Omega + \int (\mathbf{c} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} d\Omega \\ + \int 2\nu \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega - \int p \nabla \cdot \mathbf{v} d\Omega - \int q \nabla \cdot \mathbf{u} d\Omega \\ = \int \mathbf{b} \cdot \mathbf{v} d\Omega + \int \mathbf{t} \cdot \mathbf{v} d\Gamma_N. \end{aligned} \quad (4.3.15)$$

Das Integrationsgebiet Ω ist hierbei das tatsächliche Fluidgebiet Ω_y in seinem jeweiligen verformten Zustand.

4.3.8. Diskretisierung. Für die räumliche Diskretisierung der schwachen Form mit finiten Elementen werden für das Geschwindigkeitsfeld \mathbf{u} und das Druckfeld p im Raum polynomiale Ansätze gewählt

$$\mathbf{u}(\mathbf{x}, t) \approx \mathbf{N}^u(\mathbf{x}) \mathbf{u}(t) \quad \text{und} \quad p(\mathbf{x}, t) \approx \mathbf{N}^p(\mathbf{x}) p(t). \quad (4.3.16)$$

Entsprechend werden auch die Wichtungen \mathbf{v} und q im Raum mit $\mathbf{N}^u(\mathbf{x})$ bzw. $\mathbf{N}^p(\mathbf{x})$ diskretisiert. Das Ergebnis der Diskretisierung ist ein System von nichtlinearen Gleichungen

$$\mathbf{M} \dot{\mathbf{u}} + \mathbf{N}(\mathbf{u}) + \mathbf{G} p = \mathbf{f}_{\text{ext}} \quad (4.3.17)$$

$$\mathbf{G}^T \mathbf{u} = \mathbf{0}. \quad (4.3.18)$$

Hier ist \mathbf{M} die Massenmatrix, $\mathbf{N}(\mathbf{u})$ enthält die linearen inneren Kräfte und den nichtlinearen konvektiven Term, \mathbf{G} stellt den diskreten Gradientenoperator dar und \mathbf{f}_{ext} bezeichnet die äußeren Kräfte.

Für die Zeitdiskretisierung wird das one-step- θ Verfahren verwendet. Die Mittelung für die Fluidbeschleunigung im Zeitschritt lautet damit

$$\frac{\mathbf{u}^{\text{F},n+1} - \mathbf{u}^{\text{F},n}}{\Delta t} = \theta \dot{\mathbf{u}}^{\text{F},n+1} + (1 - \theta) \dot{\mathbf{u}}^{\text{F},n}. \quad (4.3.19)$$

Eingesetzt erhält man das nichtlineare Gleichungssystem

$$\mathbf{M} \mathbf{u}^{\text{F},n+1} + \theta \Delta t (\mathbf{N}(\mathbf{u}^{\text{F},n+1}) + \mathbf{G} p^{\text{F},n+1}) = \mathbf{M} (\mathbf{u}^{\text{F},n} + (1 - \theta) \Delta t \dot{\mathbf{u}}^{\text{F},n}) + \mathbf{f}_{\text{ext}} \quad (4.3.20)$$

$$\mathbf{G}^T \mathbf{u}^{\text{F},n+1} = \mathbf{0} \quad (4.3.21)$$

mit den Unbekannten $\mathbf{u}^{\text{F},n+1}$ und $p^{\text{F},n+1}$. In dem Gleichungssystem spielt die Divergenzfreiheit des Geschwindigkeitsfeldes die Rolle einer Nebenbedingung, so daß auf einem Diagonalblock eine $\mathbf{0}$ steht. Mit einem nichtlinearen Operator \mathbf{N} läßt sich die Gleichung in Matrixnotation angeben

$$\begin{bmatrix} \mathbf{M} + \theta \Delta t \mathbf{N} & \theta \Delta t \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{\text{F},n+1} \\ p^{\text{F},n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{M} (\mathbf{u}^{\text{F},n} + (1 - \theta) \Delta t \dot{\mathbf{u}}^{\text{F},n}) + \mathbf{f}_{\text{ext}} \\ \mathbf{0} \end{bmatrix}. \quad (4.3.22)$$

Dabei sind die Ansatzordnungen der Formfunktionen für das Geschwindigkeits- und das Druckfeld, $\mathbf{N}^u(\mathbf{x})$ und $\mathbf{N}^p(\mathbf{x})$, voneinander abhängig. Das Newtonsche Fluid setzt den Gradienten der Geschwindigkeit mit dem Druck in Beziehung. Und nachdem eine räumliche Ableitung die Ansatzordnung der Formfunktion um eins verringert, muß die Ansatzordnung des Druckansatzes $\mathbf{N}^p(\mathbf{x})$ —ganz grob—eins geringer sein als die Ansatzordnung des Geschwindigkeitsansatzes $\mathbf{N}^u(\mathbf{x})$. Die scharfe Formulierung dieses Zusammenhangs ist als inf-sup-Bedingung bekannt und ein klein wenig anspruchsvoller. Details können beispielsweise in Wall [162], Gresho und Sani [65], Elman u. a. [38], Förster [50] gefunden werden.

4.3.9. Stabilisierung. Es kann sinnvoll sein, Elemente zu verwenden, die die inf-sup-Bedingung nicht erfüllen, insbesondere Elemente mit gleichen Ansatzordnungen für Geschwindigkeit und Druck. Bei solchen Elementen treten künstliche Druckmoden im Lösungsbild auf, da der Druckansatz überzählige Freiheitsgrade einbringt und die Lösung nicht mehr eindeutig ist. Eine eindeutige Lösung kann jedoch durch Einbringen von Stabilisierungstermen in das Gleichungssystem (4.3.22) erreicht werden.

Ein zweites Problem, dem ebenfalls mit einem Stabilisierungsansatz begegnet werden kann, sind Oszillationen im Geschwindigkeitsfeld. Dieses Phänomen entsteht bei einer nicht ausreichenden Netzfeinheit. Der Grund sind die bei viskosen Flüssigkeiten auftretenden Grenzschichten mit hohen Gradienten, also die starken Geschwindigkeitsänderungen über kurze Strecken am Rand des Fluidgebiets senkrecht zur Flußrichtung. Die Lösung der schwachen Form der Navier-Stokes-Gleichung versucht einen Ausgleich zwischen dem Diskretisierungsfehler der Geschwindigkeiten und dem Fehler der Kräfte. Wenn nun das Netz zu grob ist, können künstliche Oszillationen im Geschwindigkeitsfeld entstehen, weil ein glattes Geschwindigkeitsfeld zu absurd falschen Kräften führen würde. Die Lösung für dieses Problem sind feine Netze. Oftmals sind die Anforderungen an die Netzfeinheit aber immens hoch und ein Stabilisierungsansatz kann die Lösungsqualität der Geschwindigkeit auf einem akzeptablen Netz verbessern.

Eine erschöpfende Darstellung der Fluidstabilisierung ist der Inhalt eines eigenen Buches. Eine Einführung ins das Thema kann in Donea und Huerta [35] gefunden werden. In Wall [162] werden die verschiedenen vorgeschlagenen Verfahren übersichtlich zusammengestellt. Die Stabilisierung der Navier-Stokes-Gleichung auf bewegten Netzen wird ausführlich in Förster [50] untersucht.

Zur Illustration der Vorgehensweise sei hier nur das ‘‘Galerkin/Least-Squares’’-Verfahren (GLS) von Hughes u. a. [80] angedeutet, das als Verallgemeinerung vieler einfacher Stabilisierungsverfahren gelten kann. Der Ausgangspunkt ist die Minimierung des Residuums der Navier-Stokes-Gleichung (4.3.13)

$$\int (\mathbf{f}_{\text{res}})^2 d\Omega \rightarrow \min \quad (4.3.23)$$

mit der Methode der kleinsten Quadrate. Daraus folgt eine Wichtung des Residuums mit Termen des Residuums selbst, die im kontinuierlichen Fall, wenn die Lösung gefunden ist, null sind

$$\int \mathbf{f}_{\text{res}} \cdot \left[\begin{array}{c} \partial \mathbf{v} / \partial t|_x + \mathbf{c} \cdot \nabla \mathbf{v} - 2\nu \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{v}) + \nabla q \\ \nabla \cdot \mathbf{v} \end{array} \right] d\Omega = \mathbf{0}. \quad (4.3.24)$$

Die diskrete Version des Integrals (4.3.24) liefert eine Reihe neuer Terme, die der diskreten Navier-Stokes-Gleichung hinzugefügt werden. Die einzelnen Terme werden dabei mit einem Stabilisierungsparameter τ versehen, der die Terme abhängig von der gewählten Diskretisierung und der aktuellen Geschwindigkeit wichtet. Der Stabilisierungsparameter τ regelt damit den Einfluß und die Asymptotik der Stabilisierungsterme, so daß die Lösung geglättet wird, ohne unphysikalisch zu werden.

Bemerkenswert ist, daß mit dem GLS-Verfahren sowohl Konvektions- als auch Druckstabilisierung in die Navier-Stokes-Gleichung eingebracht wird.

Die Druckstabilisierung bringt eine minimale Kompressibilität ins Gleichungssystem ein, wodurch auch die Nulldiagonale verschwindet und das lineare Gleichungssystem einfacher lösbar wird. Insbesondere können iterative lineare Lösungsverfahren mit den in Kapitel 3.3 besprochenen Vorkonditionierern verwendet werden. Die in Elman u. a. [38] diskutierte Block-Vorkonditionierung ist bei stabilisierten Fluidelementen nicht erforderlich.

Die Konvektionsstabilisierung erhöht die Viskosität des Fluids und führt so zum Glätten des Geschwindigkeitsfeldes auf Kosten der Genauigkeit der Fluidkräfte. Im Kontext von

FSI muß man diesen Effekt im Auge behalten, da die Kopplung mit der Struktur am Interface sinnvolle Kräfte voraussetzt. Eine zu starke Stabilisierung kann zu vollkommen unrealistischen Fluidlasten führen. Diese Überlegung führt beispielsweise Gresho und Sani [65] dazu, die Fluidstabilisierung gänzlich abzulehnen.

Zusammenfassend ist die Stabilisierung mit dem Residuum (4.3.13) eine Technik, die die Lösung der Navier-Stokes-Gleichung auf groben Netzen mit gleichen Ansätzen für Geschwindigkeit und Druck möglich macht. Kosten entstehen dabei durch eine ganze Reihe zusätzlicher Integrale auf Elementebene. Der Effekt ist das Einbringen von zusätzlicher Viskosität und leichter Kompressibilität. Durch Überstabilisierung kann man aber auch gänzlich falsche Ergebnisse erzeugen. Es kommt deshalb auf die richtige Wahl des Stabilisierungsparameters τ an. Verschiedene Parameterdefinitionen werden in Wall [162] und in Förster [50] vorgestellt.

4.3.10. Linearisierung. Die diskrete stabilisierte Navier-Stokes-Gleichung (4.3.22) wird linearisiert und mit einem Newton-Algorithmus gelöst. Die Nichtlinearität stammt dabei vom Konvektionsterm. Auch die Stabilisierung (4.3.24) bringt nichtlineare Terme ein. Das Newton-Verfahren wird in Kapitel 3.1 vorgestellt.

Das diskrete Residuum der stabilisierten Navier-Stokes-Gleichung $\mathbf{f}_{\text{res}}^{\text{F},n+1}$ ist eine Funktion der Geschwindigkeit $\mathbf{u}^{\text{F},n+1}$ und des Drucks $p^{\text{F},n+1}$. Aus der Taylor-Entwicklung um einen bekannten Lösungsvorschlag erhält man damit ein iteratives Verfahren basierend auf dem linearen System

$$\frac{\partial \mathbf{f}_{\text{res}}^{\text{F},n+1}}{\partial \mathbf{u}_i^{\text{F},n+1}} \Delta \mathbf{u}_i^{\text{F},n+1} + \frac{\partial \mathbf{f}_{\text{res}}^{\text{F},n+1}}{\partial p_i^{\text{F},n+1}} \Delta p_i^{\text{F},n+1} = -\mathbf{f}_{\text{res},i}^{\text{F},n+1}. \quad (4.3.25)$$

Es ist möglich, die Navier-Stokes-Gleichung in energieerhaltender Form aufzustellen und mit nur einer linearen Lösung im Zeitschritt zu behandeln. Die energieerhaltende Form der Navier-Stokes-Gleichung wird in Förster [50] angegeben und basiert auf einer alternativen Formulierung des Konvektionsterms, der Mittelung von konvektiver und Divergenzformulierung. Da die Divergenzformulierung für zeitlich veränderliche Fluidgebiete jedoch ungeeignet ist, ist auch die energieerhaltende Form der Navier-Stokes-Gleichung im Rahmen der FSI kaum einsetzbar. Hier wird deshalb auf eine Darstellung verzichtet.

4.3.11. Algorithmus. In Algorithmus 10 wird der Aufbau des Fluidlösers für inkompressible Fluide mit monolithischer Geschwindigkeits-Druck-Behandlung skizziert. Die effektive Systemmatrix \mathbf{K}_i und die Kräfte $\mathbf{f}_i^{\text{F},n+1}$ stellen die diskrete linearisierte Navier-Stokes-Gleichung einschließlich aller Stabilisierungsterme dar.

Bemerkenswert ist hier die Abhängigkeit der Navier-Stokes-Gleichung von der Netzverschiebung $\mathbf{d}^{\text{G},n+1}$, die im Vorgriff auf die ALE Betrachtungsweise des Navier-Stokes-Lösers im Rahmen des FSI Algorithmus angegeben wird. Die Netzverschiebung wird als bekannt angenommen. Die Netzverschiebung $\mathbf{d}^{\text{G},n+1} = \mathbf{0}$ entspricht einem unveränderlichen Fluidgebiet. Das Einbringen der Dirichlet-Freiheitsgrade wurde bereits in Abschnitt 4.2.7 diskutiert.

Eine stark verkürzte schematische Darstellung des Fluidlösers ist in Algorithmus 11 angegeben. Hier wird die effektive Systemmatrix für die Freiheitsgrade im Inneren des Gebiets mit $\mathbf{F}_{\Omega\Omega}$ bezeichnet. Zur Vereinfachung der Notation werden die Druckfreiheitsgrade $p^{\text{F},n+1}$ in den Vektor $\mathbf{u}^{\text{F},n+1}$ aufgenommen und nicht mehr explizit erwähnt. Der Algorithmus 11 ist Ausgangspunkt für die Diskussion der FSI Lösungsverfahren.

Algorithmus 10 Impliziter Fluidalgorithmus für inkompressible Fluide.Schleife über alle n Zeitschritte:

Bestimmen der Verschiebungen am Dirichletrand

$$\mathbf{u}_{D,0}^{F,n+1} \quad (4.3.26)$$

Bestimmen der externen Lasten (der Neumann-Randbedingungen)

$$\mathbf{f}_{\text{ext}}^{F,n+1} \quad (4.3.27)$$

Newton-Schleife mit $0 \leq i < \max_{\text{nlniter}}$ bzw. $|\mathbf{f}_i^{F,n+1}| \leq \epsilon^F$:Aufstellen der effektiven Systemmatrix \mathbf{K}_i und der Kräfte $\mathbf{f}_i^{F,n+1}$

$$\mathbf{K}_i = \mathbf{K}(\mathbf{u}_i^{F,n+1}, \mathbf{p}_i^{F,n+1}, \mathbf{d}^{G,n+1}) \quad (4.3.28)$$

$$\mathbf{f}_i^{F,n+1} = \mathbf{f}^F(\mathbf{u}_i^{F,n+1}, \mathbf{p}_i^{F,n+1}, \mathbf{d}^{G,n+1}) \quad (4.3.29)$$

Die Netzverschiebung $\mathbf{d}^{G,n+1}$ wird als vorgegeben angenommen.
Einbringen der Dirichlet-Randbedingungen in das lineare System

$$\Delta \mathbf{u}_{D,i}^{F,n+1} = 0, \quad \mathbf{f}_{D,i}^{F,n+1} = 0, \quad \mathbf{K}_{DD,i} = \mathbf{I}, \quad \mathbf{K}_{D\Omega,i} = 0 \quad (4.3.30)$$

Lösen des linearen Systems

$$\begin{bmatrix} \mathbf{K}_{uu,i} & \mathbf{K}_{up,i} \\ \mathbf{K}_{pu,i} & \mathbf{K}_{pp,i} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_i^{F,n+1} \\ \Delta \mathbf{p}_i^{F,n+1} \end{bmatrix} = \mathbf{f}_{\text{ext}}^{F,n+1} - \mathbf{f}_i^{F,n+1} \quad (4.3.31)$$

Aktualisierung der Geschwindigkeit und des Drucks

$$\mathbf{u}_{i+1}^{F,n+1} = \mathbf{u}_i^{F,n+1} + \Delta \mathbf{u}_i^{F,n+1} \quad (4.3.32)$$

$$\mathbf{p}_{i+1}^{F,n+1} = \mathbf{p}_i^{F,n+1} + \Delta \mathbf{p}_i^{F,n+1} \quad (4.3.33)$$

Algorithmus 11 Der vereinfachte Fluidalgorithmus.Schleife über alle n Zeitschritte:Die vorgeschriebenen Geschwindigkeiten $\mathbf{u}_{D,0}^{F,n+1}$ ermittelnDie externen Lasten $\mathbf{f}_{\text{ext}}^{F,n+1}$ berechnenNewton-Schleife mit $0 \leq i < \max_{\text{nlniter}}$:

$$\mathbf{F}_{\Omega\Omega,i} \Delta \mathbf{u}_{\Omega,i}^{F,n+1} = \mathbf{f}_{\Omega,\text{ext}}^{F,n+1} - \mathbf{f}_{\Omega,i}^{F,n+1}$$

$$\mathbf{u}_{\Omega,i+1}^{F,n+1} = \mathbf{u}_{\Omega,i}^{F,n+1} + \Delta \mathbf{u}_{\Omega,i}^{F,n+1}$$

4.3.12. Druckkorrektur. Eine alternative und in der Fluidodynamik beliebte Form, die Navier-Stokes-Gleichung zu lösen, sind die Druckkorrekturverfahren, die von Chorin [19] im Rahmen einer Diskretisierung mit finiten Differenzen vorgestellt wurden. Es existieren zahlreiche Varianten dieses Verfahrens und eine geradezu unüberschaubare Menge an Literatur, da die Druckkorrektur ein Standardverfahren für Fluidberechnungen basierend auf Diskretisierungen mit finiten Differenzen bzw. finiten Volumen ist. Einen Überblick über die Verfahren zur Lösung von Fluidproblemen kann man beispielsweise aus Rannacher [127], Jameson [84] und Chalot [18] erhalten. Im Rahmen der Finite-Elemente-Methode sind die Vorteile der Druckkorrekturverfahren weniger ausgeprägt, wie im Folgenden diskutiert wird.

Eine Übersicht über die Familie der Druckkorrekturverfahren wird in Quarteroni u. a. [125] gegeben. Die in Christon [20] vorgestellte Implementierung zielt auf maximale Effizienz

für parallele Berechnungen. Ein alternativer Ansatz zur Beschleunigung der Berechnung, der auf dem Vorausberechnen der Elementintegrale beruht, wurde in Soto u. a. [144] sowie Codina und Soto [22] vorgestellt.

Der Gedanke des originalen Druckkorrekturverfahrens ist es, bei der Navier-Stokes-Gleichung (4.3.15) die Impulsgleichung von der Massenbilanz zu trennen und beide Gleichungen unabhängig voneinander zu lösen. Um die Gleichungen voneinander trennen zu können, muß eine Näherung eingeführt werden. Die Lösung der Einzelgleichungen stellt damit eine Näherung an die Lösung der Navier-Stokes-Gleichung (4.3.15) dar. Der Vorteil des getrennten Lösens ist der deutlich reduzierte Berechnungsaufwand, da die Einzelgleichungen kleiner und für iterative lineare Lösungsverfahren leichter zugänglich sind.

Das Herleiten der Druckkorrekturverfahren kann bei der kontinuierlichen Navier-Stokes-Gleichung (4.3.15) ansetzen oder bei der diskretisierten Gleichung (4.3.22). Der zweite Weg ist der einfachere, da man die im kontinuierlichen Fall erforderliche Diskussion um die Randbedingungen vermeidet, indem man erst das fertige algebraische System aufspaltet. Der erste Weg ist für mathematische Untersuchungen oft vorteilhaft.

Analog zu der Darstellung in Codina und Soto [22], Codina [23] wird hier die diskretisierte Navier-Stokes-Gleichung (4.3.22) aufgespalten. Dazu wird zunächst die Impulsbilanz durch das Einführen einer Zwischengeschwindigkeit $\tilde{\mathbf{u}}^{F,n+1}$ und der Approximation $\mathbf{N}(\mathbf{u}^{F,n+1})\mathbf{u}^{F,n+1} \approx \mathbf{N}(\tilde{\mathbf{u}}^{F,n+1})\tilde{\mathbf{u}}^{F,n+1}$ zerlegt.

$$(\mathbf{M} + \theta \Delta t \mathbf{N}(\tilde{\mathbf{u}}^{F,n+1}))\tilde{\mathbf{u}}^{F,n+1} + \theta \Delta t \gamma \mathbf{G} \mathbf{p}^{F,n} = \mathbf{M} (\mathbf{u}^{F,n} + (1 - \theta) \Delta t \dot{\mathbf{u}}^{F,n}) + \mathbf{f}_{\text{ext}} \quad (4.3.34)$$

$$\theta \Delta t \mathbf{G} (\mathbf{p}^{F,n+1} - \gamma \mathbf{p}^{F,n}) = -\mathbf{M} (\mathbf{u}^{F,n+1} - \tilde{\mathbf{u}}^{F,n+1}) \quad (4.3.35)$$

$$\mathbf{G}^T \mathbf{u}^{F,n+1} = \mathbf{0} \quad (4.3.36)$$

Der Faktor γ regelt die Zuordnung des Druckgradienten. Gebräuchliche Werte sind $\gamma = 0$ und $\gamma = 1$, woraus sich Verfahrensvarianten mit unterschiedlichen Eigenschaften ergeben. Oftmals ist es hilfreich, den Druck vom Zeitschrittanfang $\mathbf{p}^{F,n}$ in der Impulsbilanz zu halten, also $\gamma = 1$ zu wählen. Eine detaillierte Diskussion aller denkbaren Varianten kann in Gresho und Sani [67] gefunden werden.

Das Einsetzen der Gleichung (4.3.35) in Gleichung (4.3.36) liefert die Berechnungsvorschrift.

$$(\mathbf{M} + \theta \Delta t \mathbf{N}(\tilde{\mathbf{u}}^{F,n+1}))\tilde{\mathbf{u}}^{F,n+1} + \theta \Delta t \gamma \mathbf{G} \mathbf{p}^{F,n} = \mathbf{M} (\mathbf{u}^{F,n} + (1 - \theta) \Delta t \dot{\mathbf{u}}^{F,n}) + \mathbf{f}_{\text{ext}} \quad (4.3.37)$$

$$\theta \Delta t \mathbf{G}^T \mathbf{M}^{-1} \mathbf{G} (\mathbf{p}^{F,n+1} - \gamma \mathbf{p}^{F,n}) = \mathbf{G}^T \tilde{\mathbf{u}}^{F,n+1} \quad (4.3.38)$$

$$\mathbf{M} (\mathbf{u}^{F,n+1} - \tilde{\mathbf{u}}^{F,n+1}) = -\theta \Delta t \mathbf{G} (\mathbf{p}^{F,n+1} - \gamma \mathbf{p}^{F,n}) \quad (4.3.39)$$

Die Gleichungen (4.3.37)-(4.3.39) müssen gelöst werden. Das Haupthindernis dabei stellt die inverse Massenmatrix \mathbf{M}^{-1} in der Druckkorrektur-Gleichung (4.3.38) dar. Um diese Inverse zu vermeiden, wird in Codina und Soto [22], Codina [23] der diskretisierte Laplace-Operator als Näherung eingeführt

$$\mathbf{G}^T \mathbf{M}^{-1} \mathbf{G} \approx \mathbf{L}. \quad (4.3.40)$$

Um den Sinn dieser Näherung nachzuvollziehen, wäre die Ableitung des Druckkorrekturverfahrens mit kontinuierlichen Operatoren anzusehen.

Einen alternativen Weg, die mit der inversen Massenmatrix verbundenen Schwierigkeiten zu vermeiden, gehen Gresho und Stevens [66], indem sie in den Gleichungen (4.3.37)-(4.3.39) die Massenmatrix diagonalisieren

$$\mathbf{M} \approx \mathbf{M}_D. \quad (4.3.41)$$

Die Matrix \mathbf{M}_D ist die Diagonalmatrix mit der gleichen Zeilensumme wie die Matrix \mathbf{M} . Die Massenmatrix \mathbf{M}_D entspricht damit dem physikalischen Modell mit der gleichen Gesamtmasse wie zuvor, wobei lediglich an den Knoten des Finite-Element-Netzes Massepunkte existieren und der Raum dazwischen masselos ist.

Der große Vorteil der Approximation (4.3.41) im Zusammenhang mit dem Druckkorrekturverfahren ist, daß Gleichung (4.3.39) trivial gelöst werden kann. Andernfalls würde Gleichung (4.3.39) das Lösen eines dritten linearen Systems verlangen und damit den numerischen Aufwand des Verfahrens deutlich erhöhen. Bei der Diskretisierung mit finiten Differenzen oder finiten Volumen stellt die Gleichung (4.3.39) eine Vektoraddition dar.

Die Gleichung (4.3.37) ist nichtlinear, kann aber bei festen Fluidgebieten durch eine lineare Gleichung ersetzt werden. Dazu wird die in Abschnitt 4.3.10 besprochene energierhaltende Fluidformulierung verwendet. Bei bewegten Fluidgebieten steht auch für die Druckkorrektur die energieerhaltende Form der Impulsbilanz praktisch nicht zur Verfügung.

Die Druckgleichung (4.3.38) ist linear. Bei unveränderlichen Fluidgebieten ist die Matrix $\mathbf{G}^T \mathbf{M}^{-1} \mathbf{G}$ zudem konstant.

Druckkorrekturverfahren wurden eingeführt, um sehr fein aufgelöste Fluidgebiete mit sehr vielen Unbekannten berechnen zu können. Ohne das Aufspalten und ohne die Stabilisierung aus Abschnitt 4.3.9 stellt die Navier-Stokes-Gleichung ein Sattelpunkt-Problem dar, das aufwendig zu lösen ist. Hier ist es der Vorkonditionierer des linearen iterativen Lösungsverfahrens, der wegen der Nullen auf der Diagonale der Systemmatrix nicht in gewohnter Weise einsetzbar ist. Die Motivation der Druckkorrektur stammt sehr wesentlich aus den mit dem Lösen des Sattelpunkt-Problems verbundenen Schwierigkeiten.

Durch die in Abschnitt 4.3.9 vorgestellte Stabilisierung wird das Lösen der Navier-Stokes-Gleichung stark vereinfacht. Die Stabilisierung ändert den Charakter der Navier-Stokes-Gleichung und kann hinsichtlich des Lösungsprozesses als Alternative zum Druckkorrekturverfahren aufgefaßt werden. Dabei stellt das Druckkorrekturverfahren einen deutlich größeren Eingriff in die Navier-Stokes-Gleichung dar. Die Fluidstabilisierung hat weiterhin auch über das Lösen des Gleichungssystem hinaus ihre Berechtigung. In der vorliegenden Arbeit wird deshalb der stabilisierten Navier-Stokes-Gleichung der Vorzug gegeben. Der Einsatz von Druckkorrekturverfahren für FSI Probleme wird ausführlicher in Wiesner [166] besprochen.

Eine weitere Möglichkeit, die Lösung der Navier-Stokes-Gleichung ohne Druckkorrektur zu finden, besteht darin, einen speziellen Block-Vorkonditionierer für das Sattelpunkt-Problem (4.3.22) zu konstruieren. Dieser Ansatz wurde in Elman u. a. [41, 40, 39] vorgestellt. Der Vorteil problemspezifischer Vorkonditionierer ist, daß der Lösungsprozess beschleunigt wird, ohne die Lösung zu approximieren.

Eine sehr gute Darstellung der Lösungsmethoden für die Navier-Stokes-Gleichung einschließlich der spezifischen Block-Vorkonditionierer findet sich in Elman u. a. [38].

Die Block-Vorkonditionierung der Navier-Stokes-Gleichung wurde in Heil [73] auf das vollständige FSI Problem ausgedehnt. Diese Lösungsmethode ist der Gegenstand von Kapitel 8.3 und wird dort im Kontext des FSI Problems genauer besprochen.

4.4. Das ALE-Feld

Für die ALE-Betrachtungsweise des Fluidfelds im Kontext der FSI wird die Abbildung des Referenzgebiets Ω_x in das deformierte Gebiet Ω_y benötigt. Die Deformation des Gebiets Ω_y wird von dem Strukturinterface bestimmt, d.h. die Verschiebung entlang sämtlicher Ränder ist vorgegeben, alle Ränder sind Dirichlet-Ränder. Das Deformationsverhalten innerhalb des Gebiets kann beliebig gewählt werden. Um Diskretisierungsfehler beim Lösen der Navier-Stokes-Gleichung gering zu halten, besteht der Wunsch, die Elemente im

Algorithmus 12 Der vereinfachte lineare ALE-Algorithmus.

Schleife über alle n Zeitschritte:

Die vorgeschriebenen Verschiebungen $\mathbf{d}_D^{G,n+1}$ ermitteln

$$\mathbf{A}_{\Omega\Omega}\mathbf{d}_\Omega^{G,n+1} = -\mathbf{A}_{\Omega D}\mathbf{d}_D^{G,n+1}$$

Fluidgebiet möglichst wenig und gleichmäßig zu verzerren. Gleichzeitig sollen große Gebietsdeformationen möglich sein.

Verschiedene Ansätze wurden vorgeschlagen, um diese Anforderungen zu realisieren. Für geometrisch einfache Fluidgebiete wurden vielfach algebraische Ansätze verwendet, die ohne ein zusätzliches lineares Problem auskommen. Siehe beispielsweise Heil [73].

Allgemein, für beliebige Geometrien, ist ein naheliegender Ansatz das Verwenden eines linearen Pseudo-Strukturproblems. Das Fluidgebiet wird damit bezüglich der Gebietsdeformation als Strukturkontinuum behandelt, siehe Wall [162]. Alternative Ansätze, wie der Springs-Algorithmus von Farhat u. a. [46], Degand und Farhat [26], sind gänzlich unphysikalisch, verteilen die Deformation aber unter Umständen besser ins Fluidgebiet. Diese Ansätze führen ein zusätzliches künstliches ALE-Feld ein, das neben Fluid- und Strukturfeld gelöst werden muß.

Im Rahmen dieser Arbeit wird die Netzverschiebung mit einem linearen ALE-Feld realisiert. Die Details des linearen Pseudo-Strukturproblems sind beispielsweise in Wall [162] angegeben. Hier wird auf eine ausführliche Darstellung verzichtet. Algorithmus 12 zeigt lediglich die Prinzipskizze eines reinen ALE-Problems mit der Systemmatrix \mathbf{A} , bei dem in jedem Zeitschritt ein lineares Problem für die Verschiebungen im Gebietsinneren gelöst werden muß. Die Systemmatrix \mathbf{A} wird beispielsweise durch eine FE-Diskretisierung des linearen Pseudo-Strukturproblems gewonnen.

Spektrum der FSI Lösungsverfahren

In Kapitel 3 werden Lösungsmethoden für nichtlineare Probleme vorgestellt. Ganz allgemein bestehen die vorgestellten Verfahren zum Lösen von nichtlinearen Problemen aus einer nichtlinearen Iteration, die wiederum auf dem Lösen eines linearen Problems aufbaut. Für die physikalischen Fluid- und Strukturprobleme, die in Kapitel 4 besprochen werden, hat sich das Newton-Verfahren als Lösungsverfahren durchgesetzt.

Das Newton-Verfahren benötigt die Linearisierung des nichtlinearen Problems, oder eine gute Näherung der Linearisierung, und führt somit auf eine Serie von linearen Problemen mit der gleichen Anzahl an Unbekannten wie das ursprüngliche nichtlineare Problem. Bei den in Kapitel 4 besprochenen Feldproblemen können sehr viele Unbekannte auftreten, deshalb werden im Newton-Verfahren oftmals Krylov-Verfahren als lineare Lösungsmethoden eingesetzt. Krylov-Verfahren sind für sich wiederum iterative Verfahren, die in jeder Iteration die Auswertung eines approximativen Lösungsverfahrens zur Vorkonditionierung verlangen. Das zur Lösung der Feldprobleme verwendete Newton-Krylov-Verfahren besteht damit aus einer nichtlinearen und linearen Iteration, ergänzt um einen Vorkonditionierer, der für sich wiederum ein iteratives Verfahren sein kann. Der Aufbau des Newton-Krylov-Verfahrens wurde in Kapitel 3.4 in Abbildung 3.4.1 schematisch dargestellt.

Das FSI Problem stellt nun die Kopplung zweier physikalischer Felder dar, die sich im Lösungsalgorithmus für das FSI Problem als die Kopplung zweier Feldlöser manifestiert. In den Kapiteln 7 und 8 werden ganz unterschiedliche Lösungsverfahren für das FSI Problem im Detail vorgestellt. Das vorliegende Kapitel bietet im Vorgriff auf die Diskussionen der Kapitel 6 bis 8 ein Ordnungsschema, das einen Bogen über alle Verfahren spannt und die Verwandtschaft der einzelnen Lösungsverfahren darstellt. Die Spanne reicht dabei von den Dirichlet-Neumann partitionierten Fixpunkt-Verfahren aus Kapitel 7.1, die schon von Le Tallec und Mouro [103] und Wall u. a. [163], Mok und Wall [118] eingeführt wurden, bis zu vollständig monolithischen Verfahren, die in Kapitel 8 nur gestreift werden und im Detail beispielsweise von Bazilevs u. a. [7] vorgestellt wurden.

Die Dirichlet-Neumann partitionierten Fixpunkt-Verfahren genießen den großen Vorzug, aus fertigen Lösungsverfahren für die Einzelfelder zusammengesetzt werden zu können. Bei der Suche nach brauchbaren Lösungsverfahren wird der Unabhängigkeit der Feldlöser stets große Bedeutung zugemessen, vor allem in Hinblick auf industrielle Anwendungen, bei denen in aller Regel etablierte Softwaresysteme eingesetzt werden. Eingriffe in die Feldlöser, um die gekoppelte Lösung berechnen zu können, sind dabei oft nicht opportun. Man spricht deshalb bei Lösungsverfahren für FSI Probleme oft von Kopplungsverfahren.

Die monolithischen Verfahren geben die Unabhängigkeit der Feldlöser zugunsten der Lösereffizienz auf. Einige Problemstellungen, wie sie etwa in der Biomechanik auftreten, können nachweislich nur sehr schlecht mit partitionierten Verfahren gelöst werden, wie aus den Untersuchungen von Causin u. a. [17] und Förster u. a. [52] hervorgeht bzw. auch von Heil u. a. [74] und Küttler u. a. [97] demonstriert wurde.

Die Dirichlet-Neumann-Partitionierung auf der einen Seite und der voll monolithische Ansatz auf der anderen Seite markieren zwei Extreme im Spektrum der FSI Lösungsalgorithmen. Die Verfahren zwischen diesen Extremen, die im Einzelnen in den Kapiteln 7 und 8

vorgestellt werden, können als aufeinander aufbauende Entwicklungsschritte vom partitionierten Verfahren hin zum monolithischen Verfahren verstanden werden. Die einzelnen Verfahren können durch algorithmische Umformungen ineinander überführt werden. Das Ordnungskriterium in dem Spektrum der Verfahren ist die Position der Feldkopplung im FSI Lösungsalgorithmus.

Um das Spektrum der FSI Lösungsverfahren zu veranschaulichen, werden im Folgenden die unterschiedlichen Typen der Lösungsalgorithmen schematisch dargestellt. Unter der vereinfachenden Annahme, daß sowohl das Fluidfeld als auch das Strukturfeld für sich mit einem Newton-Krylov-Verfahren gelöst werden, kann der Aufbau der Dirichlet-Neumann partitionierten Lösungsverfahren wie in Abbildung 5.0.1 angegeben skizziert werden. Das nichtlineare Kopplungsverfahren in Abbildung 5.0.1 entspricht einem der Lösungsverfahren aus Kapitel 7, d.h. es handelt sich um ein Lösungsverfahren für das nichtlineare Interfaceproblem. Die Algorithmen für das Fluidproblem und das Strukturproblem in Abbildung 5.0.1 bleiben unangetastet. Die Kopplungsschleife verbindet lediglich die nichtlinearen Feldlöser.

Entscheidend für die Einordnung der Verfahren ist, daß die Kopplungssiteration in Abbildung 5.0.1 ein Lösungsverfahren für ein nichtlineares Problem darstellt, welches für sich auf dem Lösen zweier nichtlinearer Probleme mit Hilfe des Newton-Verfahrens basiert.

Die Vereinfachung, den Fluidalgorithmus als Newton-Krylov-Verfahren aufzufassen und das Lösen des ALE-Feldes nicht zu berücksichtigen, dient der Klarheit der Darstellung. Wie in Kapitel 6.1.3 diskutiert wird, ist das Berücksichtigen der Veränderungen des Fluidgebiets eine Aufgabe des Fluidlösers. Das Einführen eines unphysikalischen ALE-Feldes ist, wie in Kapitel 4.3.6 besprochen, lediglich eine Möglichkeit, die zusammen mit der in Kapitel 6.1.5 diskutierten Dirichlet-Neumann-Partitionierung zu einer Entkopplung der Felder und damit zu einem vereinfachten Lösungsalgorithmus führt. Allgemein kann das Newton-Krylov-Verfahren auf Fluidseite als zusammengesetztes Lösungsverfahren für die Navier-Stokes-Gleichung auf veränderlichem Gebiet verstanden werden.

Eine wichtige Klasse von Lösungsverfahren, die ins Schema der in Abbildung 5.0.1 dargestellten Algorithmen fallen, im Rahmen dieser Arbeit jedoch nicht im Detail besprochen werden, sind die schwach koppelnden Verfahren. Diese Verfahren sind beispielsweise bei Problemstellungen aus der Aeroelastizität, wie sie in Farhat [45] vorgestellt werden, in Gebrauch. Allgemein kommen diese Verfahren bei der Kopplung von kompressiblen Strömungen mit Festkörpern zum Einsatz. Das Wesen der schwachen Kopplung ist dabei, die Kopplung zwischen den Feldern nicht auszuweitern. Der schwach gekoppelte Lösungsalgorithmus benötigt damit nur eine Lösung der nichtlinearen Fluid- und Strukturfelder im Zeitschritt. In aller Regel wird das Fluidfeld an das Strukturfeld gekoppelt, die Rückkopplung vom Fluid auf die Struktur aber in den nächsten Zeitschritt verschoben. Auf diese Verfahren wird in Kapitel 6.3.1 eingegangen.

Aus der in Abbildung 5.0.1 dargestellten Kopplung zweier nichtlinearer Lösungsverfahren erhält man durch Austausch des Kopplungsverfahrens mit dem Newton-Verfahren die in Abbildung 5.0.2 skizzierten partitionierten Newton-Verfahren. Die Auswirkung dieses Austauschs sind weitreichend. Es entsteht ein monolithisches Newton-Verfahren, das auf dem Lösen eines linearen FSI Problems mit den Unbekannten des Fluid- und des Strukturfeldes besteht. Das einfache Vertauschen zweier Iterationen im Lösungsalgorithmus erzeugt somit ein Lösungsverfahren mit monolithischem Charakter aus dem partitionierten Verfahren aus Abbildung 5.0.1.

Die Newton-Schleife zum Lösen des nichtlinearen Problems behandelt das gesamte FSI Problem auf einmal. Doch unterhalb des Newton-Verfahrens bleiben die in Abbildung 5.0.2 skizzierten Verfahren partitioniert. Die linearen Feldprobleme werden nach wie vor unabhängig voneinander gelöst. Die gekoppelte Lösung der linearen Feldprobleme wird mit

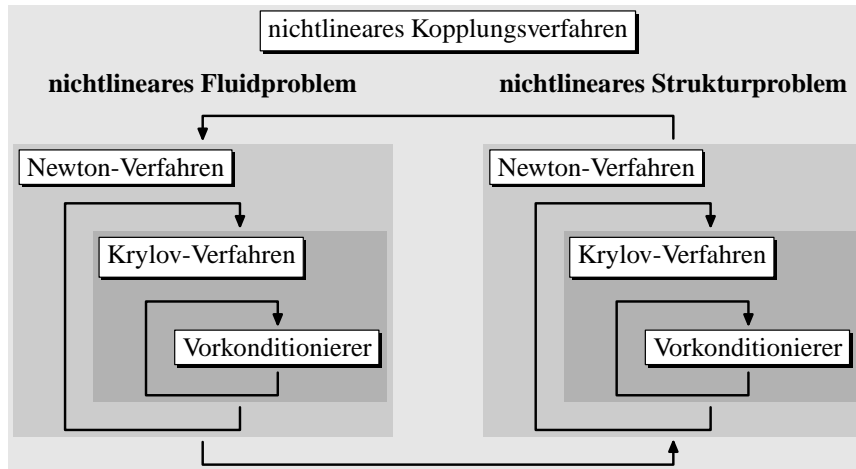


Abbildung 5.0.1: Aufbau der Dirichlet-Neumann partitionierten Lösungsverfahren.

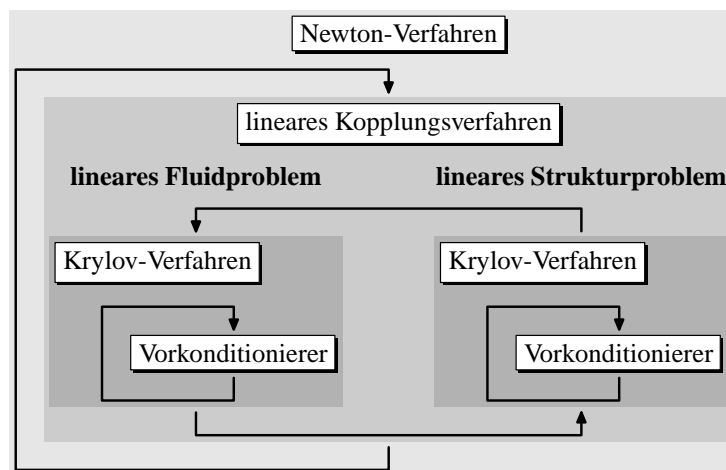


Abbildung 5.0.2: Aufbau der partitionierten Newton-Verfahren.

Hilfe eines weiteren linearen Lösungsverfahrens gefunden. Somit sind am Lösungsalgorithmus 5.0.2 ein nichtlineares und zwei lineare iterative Verfahren beteiligt.

In Kapitel 8.1 wird eine Dirichlet-Neumann-Partitionierung für das lineare FSI Problem eingeführt, ganz analog zu der Dirichlet-Neumann-Partitionierung für das nichtlineare FSI Problem in Kapitel 6. Die in Kapitel 7 für das nichtlineare FSI Problem angewandten Lösungsmethoden können somit auch für das lineare FSI Problem verwendet werden. Alternativ wird in Kapitel 8.2 ein Block-Gauß-Seidel-Verfahren zur Lösung des linearen FSI Problems vorgestellt, das auf die Dirichlet-Neumann-Partitionierung des linearen Problems verzichtet.

In einem weiteren Schritt kann die Iteration zur Kopplung der Felder mit der Iteration des Krylov-Verfahrens vertauscht werden. Die Kopplung der Feldprobleme findet dadurch innerhalb eines gemeinsamen Krylov-Verfahrens, im Vorkonditionierer, statt. Das daraus resultierende Verfahren ist in Abbildung 5.0.3 dargestellt. Als Vorkonditionierer

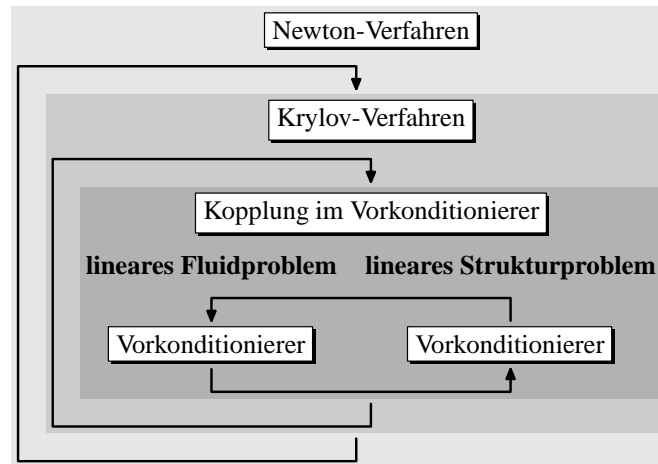


Abbildung 5.0.3: Aufbau der Block-vorkonditionierten Newton-Krylov-Verfahren.

kann im Zuge einer Richardson-Iteration (3.3.8) das Block-Gauß-Seidel-Verfahren verwendet werden, wobei zur Lösung der einzelnen Blöcke die Anwendung des entsprechenden Feld-Vorkonditionierers ausreicht. Das Newton-Krylov-Verfahren für sich ist nicht partitioniert, der Block-Vorkonditionierer ist es. Diese Block-vorkonditionierten Newton-Krylov-Verfahren werden in Kapitel 8.3 besprochen.

Bemerkenswert ist, daß sich durch diesen Austausch die Natur des Lösungsverfahrens grundlegend ändert. Anders als die zuvor besprochenen Verfahren ist in Abbildung 5.0.3 ein gewöhnliches Newton-Krylov-Verfahren dargestellt, das lediglich einen problemspezifischen Vorkonditionierer besitzt. Insbesondere besteht dieses Verfahren nur aus einer nichtlinearen und einer linearen Iteration. Die Kopplungsiteration im Vorkonditionierer ist von den Abbruchbedingungen des Newton-Krylov-Verfahrens entkoppelt. Das Lösungsverfahren ist dadurch erheblich einfacher zu handhaben als die vorher betrachteten Verfahren.

Wenn bei dem Block-vorkonditionierten Newton-Krylov-Verfahren algebraische Mehrgitterverfahren als Block-Vorkonditionierer verwendet werden, kann die Kopplungsiteration auf den einzelnen Gitterebenen ausgeführt werden. Dabei wird gleichsam die Iteration über die Ebenen des Mehrgitterverfahrens mit der Kopplungsiteration zwischen den Feldern vertauscht. Auf diese Weise entsteht ein problemspezifisches Mehrgitterverfahren. Diese Variante des vorkonditionierten Newton-Krylov-Verfahrens wird in Kapitel 8.4 besprochen.

Die natürliche Fortsetzung schließlich stellt das vollständig monolithische Lösungsverfahren dar, das Newton-Krylov-Verfahren mit monolithischem Vorkonditionierer, das die Einzelfelder auf keiner Ebene unterscheidet. Das vollständig monolithische Lösungsverfahren kennt keine Kopplungsschleife mehr.

Die in den Abbildungen 5.0.1, 5.0.2 und 5.0.3 dargestellten Skizzen umreißen somit das Spektrum der in der vorliegenden Arbeit betrachteten Verfahren. Alle Verfahren sind stark koppelnde Verfahren, alle Verfahren konvergieren somit zur gleichen Lösung. Der Unterschied zwischen den Verfahren ist die Position der Feldkopplung, mit der das Fluid- mit dem Strukturfeld verbunden wird.

In den folgenden Kapiteln werden die konkreten Lösungsverfahren im Detail betrachtet.

Nicht berücksichtigt in diesem Spektrum werden Verfahren, die auf approximativen Lösungen der Einzelfelder beruhen. Insbesondere lassen sich aus den in Kapitel 4.3.12 vorgestellten Druckkorrektur-Verfahren nur sehr schlecht monolithische FSI Verfahren konstruieren. Siehe dazu Wiesner [166] und die dort zitierte Literatur. Auch werden in das vorgestellte Spektrum neben den monolithischen Verfahren nur Dirichlet-Neumann partitionierte Verfahren aufgenommen. Alternative Formulierungen am Interface sind im Rahmen des vorliegenden Spektrums zwar denkbar, entsprechende Verfahren sind jedoch durchaus mit Nachteilen verbunden und daher faktisch nicht in Gebrauch. Die Diskussion um die Partitionierungsmöglichkeiten wird in Kapitel 6.1 geführt.

Partitionierte Lösungsverfahren für FSI Probleme

Partitionierte Lösungsverfahren für FSI Probleme setzen die nichtlinearen Feldlöser unabhängig voneinander ein und vermitteln den Austausch von Kopplungsinformationen, um das gekoppelte FSI Problem zu lösen. Der Ausgangspunkt bei der Konstruktion partitionierter Lösungsverfahren ist in aller Regel die Existenz der Feldlöser, die zu einem Lösungsverfahren für das gekoppelte Problem zusammengesetzt werden sollen.

Partitionierte Verfahren für FSI Probleme sind vielfach vorgestellt und diskutiert worden. Frühe Arbeiten auf diesem Gebiet sind Wall u. a. [163], Mok und Wall [118] sowie Le Tallec und Mouro [103]. Alternative Lösungsmethoden wurden beispielsweise in Matties und Steindorf [108, 109], Gerbeau und Vidrascu [60], Gerbeau u. a. [61], Fernández und Moubachir [48] oder Tezduyar u. a. [153], Tezduyar und Sathe [152] vorgestellt. In Küttler und Wall [98] werden partitionierte Fixpunkt-Verfahren verglichen. Küttler und Wall [99] verwendet die Vektorextrapolation als Lösungsverfahren, analog zu Michler u. a. [115, 116]. Der Extrapolationsansatz wird in Vierendeels [159], Vierendeels u. a. [160] und Degroote u. a. [27] zum Erzeugen von Modellproblemen verwendet. In Nobile und Vergara [119] werden alternative Interfacebedingungen und in Deparis u. a. [30] alternative Kopplungsmethoden untersucht. Die Arbeit von Deparis [29] enthält eine gute Einführung in die zur Verfügung stehenden Lösungsmethoden.

Die partitionierten Verfahren sind attraktiv, da sie einen sehr flexiblen Rahmen bereitstellen, der die Kombination nahezu beliebiger Formulierungen für die Einzelfelder erlaubt. Gleichzeitig verringert die Zerlegung des Gesamtproblems in Teile den Speicherbedarf der Lösungsmethoden, möglicherweise jedoch auf Kosten der Rechenzeit. In dieser Arbeit werden die in Kapitel 4 vorgestellten Algorithmen zur Lösung der Feldprobleme eingesetzt. Die partitionierten Lösungsmethoden für FSI Probleme sind jedoch nicht an spezifische Feldlöser gebunden. Diese Flexibilität bei der Auswahl der Lösungsmethoden ist auf mehreren Ebenen von Vorteil.

Zunächst können für das Fluid- und das Strukturfeld sehr unterschiedliche mathematische Modelle und numerische Verfahren eingesetzt werden. Die Wahl des numerischen Verfahrens sollte von der Physik und der Modellierung des jeweiligen Feldes bestimmt sein. Das Kopplungsverfahren sollte die Kombination beliebiger Verfahren erlauben.

Aus Sicht der Softwareentwicklung spielt beim Lösen von FSI Problemen die Modularität der Software eine große Rolle. Wenn Softwarepakete zur Lösung einzelner Felder vorliegen oder von unterschiedlichen Teams entwickelt werden, sind partitionierte Lösungsverfahren vorteilhaft, da sie einfach mit unabhängigen Softwarekomponenten formuliert werden können. Die komplexen feldspezifischen Lösungsalgorithmen können hinter einer Softwareschnittstelle verborgen werden. Auf diese Weise ist auch die Weiterentwicklung, Pflege oder gar der Austausch der Feldlöser möglich.

Auf die Prinzipien der Softwareentwicklung mit unabhängigen Komponenten kann an dieser Stelle nicht eingegangen werden. Ein paar Gedanken und Literaturhinweise zu diesem weiten Themengebiet sind in Anhang A zusammengestellt.

Schließlich erlauben partitionierte Lösungsverfahren, die Diskretisierungen bei der Simulation von FSI Systemen in Raum und Zeit unabhängig voneinander zu wählen. Bei sehr großen komplexen Geometrien, bei denen die Vernetzung ein signifikanter Arbeitsanteil ist, kann es unter Umständen sogar sinnvoll sein, die Diskretisierungen der Felder von Spezialisten für das jeweilige Feld herstellen zu lassen. Die Kopplung nicht passender Diskretisierungen ist mit partitionierten Verfahren sehr leicht möglich. In den folgenden Darstellungen wird jedoch von passenden Diskretisierungen am Interface ausgegangen, um die Darstellung der Lösungsalgorithmen einfach zu halten.

Für die Diskussion der partitionierten Lösungsverfahren ist es hilfreich, die Partitionierung des voll gekoppelten nichtlinearen FSI Problems vom Lösungsalgorithmus für das partitionierte Problem zu unterscheiden, auch wenn beide natürlich unmittelbar miteinander zusammenhängen.

Der Sinn der Partitionierung ist es, die Felder voneinander zu trennen. Dabei entstehen am Interface zwischen den Feldern künstliche Ränder, an denen Randbedingungen gewählt werden müssen. Die Feldlöser können damit unabhängig voneinander verwendet werden. Aus der Partitionierung des diskretisierten nichtlinearen FSI Problems geht ein nichtlineares Problem formuliert in Interfacefreiheitsgraden hervor, das auf den nichtlinearen Feldproblemen beruht.

Um dieses Interfaceproblem zu lösen, werden die nichtlinearen Feldlöser zu einem Lösungsalgorithmus, dem Kopplungsverfahren, zusammengesetzt. Die unabhängigen, entkoppelten Feldlöser bilden die Bausteine für dieses Kopplungsverfahren. Das Kopplungsverfahren selbst ist ein Lösungsverfahren für das nichtlineare Interfaceproblem.

Im Folgenden werden die Partitionierung und der darauf aufgebaute Kopplungsalgorithmus im Detail besprochen.

6.1. Partitionierung

Partitionierte Verfahren sind von den Gebietszerlegungsmethoden bekannt. In der Tat kann die FSI Berechnung mit partitionierten Verfahren als ein Spezialfall der Zerlegung des Gesamtgebiets in zwei nichtlineare Teilgebiete betrachtet werden. Die Nichtlinearität der Teilgebiete, des Fluidfeldes und des Strukturfeldes, bedeutet dabei eine erhebliche Einschränkung für die praktisch anwendbaren Verfahren. Weiterhin kommt, um den Sinn der Partitionierung nicht in Frage zu stellen, lediglich eine nichtüberlappende Zerlegung in ein Fluid- und ein Strukturgebiet in Frage. Die Gebietsgrenze folgt auch zeitlich der Deformation der Struktur und es bleibt dem numerischen Verfahren auf der Fluidseite überlassen, mit der Änderung des Fluidgebiets zurecht zu kommen.

Am Interface Γ , der Trennfläche zwischen Fluid- und Strukturgebiet, müssen für beide Seiten Annahmen über die Randbedingungen getroffen werden. Ganz allgemein kann der künstliche Rand am Interface in jedem Feld für sich entweder als Dirichlet- oder als Neumannrand aufgefaßt werden. Linearkombinationen aus Dirichlet- und Neumannbedingung, also Robin-Ränder, sind im Prinzip auch möglich, werden hier aber nicht näher betrachtet, da Robin-Ränder nur von wenigen Feldlösern unterstützt werden und die auf Robin-Rändern basierenden Kopplungsverfahren bisher von untergeordneter Bedeutung sind.

6.1.1. Kopplungsbedingung. In Förster [50] werden die diskreten kinematischen und kinetischen Kopplungsbedingungen am Interface

$$\mathbf{d}_{\Gamma}^{S,n+1} = \mathbf{d}_{\Gamma}^{S,n} + \frac{\Delta t}{2} \left(\mathbf{u}_{\Gamma}^{F,n+1} + \mathbf{u}_{\Gamma}^{F,n} \right) \quad \text{und} \quad \mathbf{f}_{\Gamma}^{S,n+1} + \mathbf{f}_{\Gamma}^{F,n+1} = \mathbf{0} \quad (6.1.1)$$

Algorithmus 13 Der nichtlineare Interfaceoperator des Strukturfeldes $\mathbf{f}_\Gamma^{n+1} = \mathcal{S}_\Gamma(\mathbf{d}_\Gamma^{n+1})$.

Vorgeschriebene Verschiebungen $\mathbf{d}_{D,0}^{S,n+1}$ ermitteln

Interfaceverschiebung \mathbf{d}_Γ^{n+1} in vorgeschriebene Verschiebungen $\mathbf{d}_{D,0}^{S,n+1}$ einfügen

Externe Lasten $\mathbf{f}_{\text{ext}}^{S,n+1}$ berechnen

Newton-Schleife mit $0 \leq i < \max_{\text{nIter}}$:

$$\mathbf{S}_{\text{II},i} \Delta \mathbf{d}_{\text{I},i}^{S,n+1} = \mathbf{f}_{\text{I,ext}}^{S,n+1-\alpha_f} - \mathbf{f}_\Gamma^{S,n+1-\alpha_f}(\mathbf{d}_i^{S,n+1})$$

$$\mathbf{d}_{\text{I},i+1}^{S,n+1} = \mathbf{d}_{\text{I},i}^{S,n+1} + \Delta \mathbf{d}_{\text{I},i}^{S,n+1}$$

Kraftantwort berechnen $\mathbf{f}_\Gamma^{n+1} = \mathbf{f}_{\Gamma,\text{ext}}^{S,n+1} - \mathbf{f}_\Gamma^{S,n+1}(\mathbf{d}^{S,n+1})$

abgeleitet. Für die kinematische Kopplungsbedingung wird jedoch häufig anstelle der Trapezregel (6.1.1) die einfachere Euler Formel

$$\mathbf{d}_\Gamma^{S,n+1} = \mathbf{d}_\Gamma^{S,n} + \Delta t \mathbf{u}_\Gamma^{F,n+1} \quad (6.1.2)$$

verwendet, um Oszillationen zu vermeiden. Dies ist eine Beziehung zwischen den Interfaceverschiebungen der Struktur und der Interfacegeschwindigkeit des Fluids. Die Druckfreiheitsgrade des Fluids sind an der kinematischen Kopplungsbedingung nicht beteiligt.

6.1.2. Strukturoperator. Im Strukturgebiet ist das Verschiebungsfeld \mathbf{d}^{n+1} gesucht, das das Gleichgewicht der Kräfte \mathbf{f}^{n+1} an jedem Punkt im Gebiet erfüllt. Somit kann dem Strukturlöser im partitionierten Lösungsalgorithmus entweder die Interfaceverschiebung \mathbf{d}_Γ^{n+1} als Dirichlet-Bedingung oder die Interfacekraft \mathbf{f}_Γ^{n+1} als Neumann-Bedingung vorgeschrieben werden. Die Anfangsbedingungen und sämtliche Randbedingungen außerhalb des Interfaces sind von der Partitionierung nicht betroffen.

Das Lösen des Strukturfeldes mit einer vorgeschriebenen Interfaceverschiebung \mathbf{d}_Γ^{n+1} und das anschließende Berechnen der resultierenden Interfacekraft \mathbf{f}_Γ^{n+1} läßt sich als nichtlineare Interfaceoperation auf Strukturseite aufschreiben

$$\mathbf{f}_\Gamma^{n+1} = \mathcal{S}_\Gamma(\mathbf{d}_\Gamma^{n+1}). \quad (6.1.3)$$

Ganz analog wird das Vorgeben einer Interfacekraft \mathbf{f}_Γ^{n+1} und das Berechnen der resultierenden Interfaceverschiebung \mathbf{d}_Γ^{n+1} als inverser nichtlinearer Operator geschrieben

$$\mathbf{d}_\Gamma^{n+1} = \mathcal{S}_\Gamma^{-1}(\mathbf{f}_\Gamma^{n+1}). \quad (6.1.4)$$

Die Notation in den Gleichungen (6.1.3) und (6.1.4) zeigt die Abhängigkeiten von Kraft und Verschiebung am Interface. Alle sonstigen Randbedingungen, Anfangsbedingungen und Zustandsvariablen des Strukturfeldes sind ebenso erforderlich, werden aber, um die Notation prägnant zu halten, nicht dargestellt.

Für ein nichtlineares Strukturfeld können die Interfaceoperatoren \mathcal{S}_Γ und \mathcal{S}_Γ^{-1} nicht explizit angegeben werden. Die Operatoren werden implizit ausgewertet. Die dazu nötigen Algorithmen können aus dem vereinfachten nichtlinearen Strukturalgorithmus, Algorithmus 9, abgeleitet werden.

Die Interfaceoperatoren sind abhängig vom Zeitschritt, ansonsten aber kontextfrei, d.h. innerhalb eines Zeitschritts wird bei mehrfachem Auswerten eines Interfaceoperators \mathcal{S}_Γ oder \mathcal{S}_Γ^{-1} mit dem gleichen Argument stets das gleiche Ergebnis erzielt.

Algorithmus 14 Der Interfaceoperator des Strukturfeldes $\mathbf{d}_\Gamma^{n+1} = \mathcal{S}_\Gamma^{-1}(\mathbf{f}_\Gamma^{n+1})$.

Vorgeschriebene Verschiebungen $\mathbf{d}_{D,0}^{S,n+1}$ ermitteln

Externe Lasten $\mathbf{f}_{\text{ext}}^{S,n+1}$ berechnen

Newton-Schleife mit $0 \leq i < \max_{\text{nliter}}$:

$$\mathbf{S}_{\Omega\Omega,i} \Delta \mathbf{d}_{\Omega,i}^{S,n+1} = \mathbf{f}_{\Omega,\text{ext}}^{S,n+1-\alpha_f} - \mathbf{f}_\Omega^{S,n+1-\alpha_f}(\mathbf{d}_i^{S,n+1}) - (1-\alpha_f)\mathbf{f}_\Gamma^{n+1} - \alpha_f \mathbf{f}_\Gamma^n$$

$$\mathbf{d}_{\Omega,i+1}^{S,n+1} = \mathbf{d}_{\Omega,i}^{S,n+1} + \Delta \mathbf{d}_{\Omega,i}^{S,n+1}$$

Verschiebungswantwort extrahieren $\mathbf{d}_\Gamma^{n+1} = \mathbf{d}_\Gamma^{S,n+1}$

Für die Berechnung der Interfacekraft \mathbf{f}_Γ^{n+1} aus der Interfaceverschiebung \mathbf{d}_Γ^{n+1} nach Gleichung (6.1.3) wird die Menge der vorgeschriebenen Verschiebungen $\mathbf{d}_{D,0}^{S,n+1}$ in Algorithmus 9 um die Interfaceverschiebungen \mathbf{d}_Γ^{n+1} erweitert. Die vorgeschriebene Interfaceverschiebung wird als zusätzliche Dirichlet-Bedingung ins Strukturproblem eingebracht. Somit wird nur mehr nach den Verschiebungen im reduzierten Gebietsinneren $\mathbf{d}_\Gamma^{S,n+1}$, ohne die Interfaceverschiebungen, gelöst. Die Interfacekraft \mathbf{f}_Γ^{n+1} wird durch Einsetzen der berechneten Verschiebungen bestimmt. Der Algorithmus zum impliziten Auswerten von Gleichung (6.1.3) ist in Algorithmus 13 dargestellt.

Die Berechnung der Interfaceverschiebung \mathbf{d}_Γ^{n+1} aus der Interfacekraft \mathbf{f}_Γ^{n+1} entsprechend Gleichung (6.1.4) verlangt dagegen das Erweitern der rechten Seite des linearen Gleichungssystems in Algorithmus 9, wie in Algorithmus 14 angegeben. Die Interfacekraft ist hier eine zusätzliche Neumann-Bedingung für das nichtlineare Strukturproblem.

Die Algorithmen 13 und 14 lösen beide ein nichtlineares Strukturproblem mit Hilfe der Newton-Methode. Der Unterschied besteht in unterschiedlichen Randbedingungen. Das Anwenden des Strukturoperators \mathcal{S}_Γ oder des inversen Strukturoperators \mathcal{S}_Γ^{-1} ist deshalb algorithmisch sehr ähnlich. Insbesondere verlangt der inverse Strukturoperator \mathcal{S}_Γ^{-1} nicht den inversen Lösungsalgorithmus des Strukturoperators \mathcal{S}_Γ .

Die Strukturoperatoren (6.1.3) und (6.1.4) ergänzen einander als Hin- und Rückrechnung. Wegen der Nichtlinearität sind die Strukturoperatoren mit den Definitionen nach den Algorithmen 13 und 14 jedoch auch in ihrer Wirkung nicht exakt invers zueinander

$$\mathbf{d}_\Gamma^{n+1} \neq \mathcal{S}_\Gamma^{-1}(\mathcal{S}_\Gamma(\mathbf{d}_\Gamma^{n+1})). \quad (6.1.5)$$

Beide Operatoren gehen von einer unverformten Struktur aus. Die Interfacekraft \mathbf{f}_Γ^{n+1} , die mit der vorgegebenen Verschiebung \mathbf{d}_Γ^{n+1} aus $\mathcal{S}_\Gamma(\mathbf{d}_\Gamma^{n+1})$ berechnet wird, beschreibt das dynamische Gleichgewicht an der verformten Struktur. Der inverse Operator $\mathcal{S}_\Gamma^{-1}(\mathbf{f}_\Gamma^{n+1})$ wird mit dieser Kraft jedoch wieder im unverformten Zustand belastet. Durch die Nichtlinearität entsteht der Unterschied (6.1.5). Für die Motivation von Lösungsmethoden für partitionierte FSI Probleme kann in (6.1.5) aber Gleichheit angenommen werden.

Eine weitere Einschränkung bei der Anwendung der Strukturoperatoren (6.1.3) und (6.1.4) ergibt sich ebenfalls aus der Nichtlinearität des Strukturproblems. Die Argumente der Operatoren, die Interfaceverschiebung bzw. die Interfacekraft, dürfen nicht zu weit von der tatsächlichen Gleichgewichtslösung des gekoppelten Problems entfernt sein. Was im mathematischen Sinne „zu weit“ ist, ist allgemein dabei schwer zu fassen und wird an dieser Stelle nicht untersucht. Eine Lösung des Strukturproblems kann nur mit physikalisch sinnvollen Randbedingungen gefunden werden. Diese Einschränkung hat Auswirkungen auf die möglichen Lösungsalgorithmen für das gekoppelte Problem.

6.1.3. Fluidoperator. Analog zum Strukturfeld können auch für das Fluidfeld Interfaceoperatoren eingeführt werden. Auf der Fluidseite muß die Navier-Stokes-Gleichung

Algorithmus 15 Der nichtlineare Interfaceoperator des Fluidfeldes $\mathbf{f}_\Gamma^{n+1} = \mathcal{F}_\Gamma(\mathbf{d}_\Gamma^{n+1})$.

ALE Randbedingungen setzen $\mathbf{d}_D^{G,n+1} = \mathbf{0}$ und $\mathbf{d}_\Gamma^{G,n+1} = \mathbf{d}_\Gamma^{n+1}$

$$\mathbf{A}_{\text{II}} \mathbf{d}_I^{G,n+1} = -\mathbf{A}_{\text{II}} \mathbf{d}_\Gamma^{G,n+1}$$

Vorgeschriebene Geschwindigkeiten $\mathbf{u}_{D,0}^{F,n+1}$ ermitteln

$$\text{Geschwindigkeit am Interface einbringen } \mathbf{u}_{\Gamma,0}^{F,n+1} = \frac{2}{\Delta t} (\mathbf{d}_\Gamma^{n+1} - \mathbf{d}_\Gamma^n) - \mathbf{u}_\Gamma^{F,n}$$

Externe Fluidlasten $\mathbf{f}_{\text{ext}}^{F,n+1}$ berechnen

Newton-Schleife mit $0 \leq i < \max_{\text{nIter}}$:

$$\mathbf{F}_{\text{II},i} \Delta \mathbf{u}_{I,i}^{F,n+1} = \mathbf{f}_{I,\text{ext}}^{F,n+1} - \mathbf{f}_{I,i}^{F,n+1}$$

$$\mathbf{u}_{I,i+1}^{F,n+1} = \mathbf{u}_{I,i}^{F,n+1} + \Delta \mathbf{u}_{I,i}^{F,n+1}$$

Kraftantwort berechnen $\mathbf{f}_\Gamma = -\mathbf{f}_{\Gamma,\text{ext}}^{F,n+1} + \mathbf{f}_\Gamma^{F,n+1}(\mathbf{u}^{F,n+1}, \mathbf{d}^{G,n+1})$

gelöst und dabei das sich ändernde Fluidgebiet beachtet werden. Der vereinfachte Fluidalgorithmus zur Lösung der Navier-Stokes-Gleichung wurde in Algorithmus 11 vorgestellt. Die Unbekannten der inkompressiblen Navier-Stokes-Gleichung sind Geschwindigkeit $\mathbf{u}^{F,n+1}$ und Druck $p^{F,n+1}$, wobei der Druck in Algorithmus 11 nicht explizit erwähnt wird, sondern im Geschwindigkeitsvektor $\mathbf{u}^{F,n+1}$ enthalten ist.

Zur Darstellung der partitionierten FSI Verfahren ist es sinnvoll, die Behandlung des zeitlich veränderlichen Fluidgebiets und die Lösung der Navier-Stokes-Gleichung in einen einzigen Operator \mathcal{F}_Γ zu abstrahieren. Ganz allgemein wird deshalb ein nichtlinearer Fluidoperator eingeführt, der aus einer gegebenen Interfaceverschiebung \mathbf{d}_Γ^{n+1} eine Interfacekraft \mathbf{f}_Γ^{n+1} berechnet

$$\mathbf{f}_\Gamma^{n+1} = \mathcal{F}_\Gamma(\mathbf{d}_\Gamma^{n+1}). \quad (6.1.6)$$

Die Umrechnung der Interfaceverschiebung \mathbf{d}_Γ^{n+1} in die Interfacegeschwindigkeit $\mathbf{u}_\Gamma^{F,n+1}$ nach Gleichung (6.1.1) wird ebenfalls im Fluidoperator abstrahiert. Die Vorzeichendefinition der Interfacekraft \mathbf{f}_Γ^{n+1} wird vom Strukturoperator \mathcal{S}_Γ übernommen, um die Kombination der Operatoren bei der Konstruktion von Lösungsalgorithmen für das gekoppelte Problem zu erleichtern.

Zur Behandlung des zeitlich veränderlichen Fluidgebiets stehen, wie in Kapitel 4.3.6 beschrieben, verschiedene Strategien zur Verfügung. In dieser Arbeit wird ein Deformationsansatz für das Fluidgebiet verwendet, bei dem sich das Fluidnetz mit Hilfe des in Kapitel 4.4 vorgestellten künstlichen Feldes verformt und die Navier-Stokes-Gleichung in der „Arbitrary Lagrangean Eulerian“ (ALE) Betrachtungsweise relativ zum deformierten Netz formuliert wird. Die Deformation des Fluidgebiets $\mathbf{d}_\Gamma^{G,n+1}$ hängt dabei an der Interfaceverschiebung \mathbf{d}_Γ^{n+1}

$$\mathbf{d}_\Gamma^{G,n+1} = \mathbf{d}_\Gamma^{n+1}. \quad (6.1.7)$$

Das ALE-Feld bestimmt die Verschiebung der Netzknoten im Gebietsinneren $\mathbf{d}_\Gamma^{G,n+1}$, wie in Algorithmus 12 beispielhaft gezeigt wird.

Die Bemerkungen zum Strukturoperator \mathcal{S}_Γ treffen sinngemäß auch auf den Fluidoperator \mathcal{F}_Γ zu. Der Operator \mathcal{F}_Γ kann nicht explizit angegeben werden. Das Auswerten des Operators \mathcal{F}_Γ verlangt die Kombination der Algorithmen 11 und 12, wie sie in Algorithmus 15 dargestellt ist.

Als Alternative zur Deformation des Fluidgebietes mit dem ALE-Ansatz kann die Verschiebung des Strukturinterfaces und die daraus folgende Änderung des Fluidgebietes auch mit dem von Gerstenberger und Wall [63, 62] vorgestellten XFEM-Ansatz verfolgt werden. Dazu ist lediglich ein Fluidoperator \mathcal{F}_Γ zu verwenden, der das Fluidgebiet entsprechend der vorgegebenen Interfaceposition \mathbf{d}_Γ^{n+1} beschneidet und auf der dabei entstehenden

Schnittfläche die nach Gleichung (6.1.1) bestimmte Interfacegeschwindigkeit $\mathbf{u}_\Gamma^{F,n+1}$ vorschreibt. Da die Schnittfläche bei diesem Ansatz quer durch die Fluidelemente läuft, ist sowohl das Aufbringen der Randbedingungen am Interfaceraud als auch die Integration über das tatsächliche Fluidgebiet mit erheblichen Schwierigkeiten verbunden. Die numerische Behandlung dieser Schwierigkeiten ist der Gegenstand der Arbeiten von Gerstenberger und Wall [63, 62].

Der Wechsel zum XFEM-Ansatz für die Behandlung des Fluidgebietes bedeutet also aus Sicht des FSI Algorithmus lediglich den Austausch des Fluidoperators. In diesem Sinne kann jeder Fluidoperator \mathcal{F}_Γ , der die Interfaceverschiebung \mathbf{d}_Γ^{n+1} auf die Interfacekraft \mathbf{f}_Γ^{n+1} abbildet, zur Konstruktion von partitionierten FSI Algorithmen verwendet werden. Für den Kopplungsalgorithmus spielen die Implementierungsdetails des Fluidoperators keine Rolle. In dieser Arbeit wird der auf dem ALE-Ansatz basierende Fluidoperator aus Algorithmus 15 verwendet.

In Algorithmus 15 werden unabhängig voneinander zwei Felder, das Fluidfeld und das ALE-Feld, gelöst. Der Algorithmus läßt sich damit aus zwei unabhängigen Feldlösern zusammensetzen, vorausgesetzt der Fluidalgorithmus kann die Navier-Stokes-Gleichung auf einem bewegten Fluidnetz lösen.

Der inverse Interface-Fluidoperator läßt sich genauso anschreiben

$$\mathbf{d}_\Gamma^{n+1} = \mathcal{F}_\Gamma^{-1}(\mathbf{f}_\Gamma^{n+1}), \quad (6.1.8)$$

doch die algorithmische Umsetzung des inversen Fluidoperators \mathcal{F}_Γ^{-1} ist deutlich komplexer als die des Fluidoperators \mathcal{F}_Γ . In Gleichung (6.1.8) ist die Interfaceverschiebung \mathbf{d}_Γ^{n+1} unbekannt und damit auch die Deformation des Fluidgebiets $\mathbf{d}^{G,n+1}$. Die Navier-Stokes-Gleichung muß also auf einem unbekanntem Gebiet integriert werden. Die Gebietsdeformation muß gleichzeitig mit der nichtlinearen Navier-Stokes-Gleichung gelöst werden. Als Folge davon wird die Gebietsdeformation $\mathbf{d}^{G,n+1}$ eine zusätzliche Variable der diskreten Navier-Stokes-Gleichung und die Linearisierung (4.3.25) muß um einen Term ergänzt werden

$$\frac{\partial \mathbf{f}_{\text{res}}^{F,n+1}}{\partial \mathbf{u}_i^{F,n+1}} \Delta \mathbf{u}_i^{F,n+1} + \frac{\partial \mathbf{f}_{\text{res}}^{F,n+1}}{\partial \mathbf{p}_i^{F,n+1}} \Delta \mathbf{p}_i^{F,n+1} + \frac{\partial \mathbf{f}_{\text{res}}^{F,n+1}}{\partial \mathbf{d}_i^{G,n+1}} \Delta \mathbf{d}_i^{G,n+1} = -\mathbf{f}_{\text{res},i}^{F,n+1}. \quad (6.1.9)$$

Dementsprechend tritt die Sensitivitätsmatrix

$$\mathbf{F}^G = \frac{\partial \mathbf{f}_{\text{res}}^{F,n+1}}{\partial \mathbf{d}_i^{G,n+1}} \quad (6.1.10)$$

im linearisierten Gleichungssystem in Algorithmus 16, dem Algorithmus des inversen Fluidoperators, auf. Details zur Berechnung der Sensitivitätsmatrix können beispielsweise Braess und Wriggers [11] oder Fernández und Moubachir [48] entnommen werden.

Die enge Verzahnung zwischen dem Navier-Stokes-Löser und dem ALE-Löser, die auch die Sensitivitätsmatrix bedingt, ist mit unabhängigen Feldlösern nicht zu realisieren. Die Frage, wie das zusammengesetzte lineare System in Algorithmus 16 gelöst werden kann, wird im Zusammenhang mit den monolithischen FSI Verfahren in Kapitel 8.3 aufgegriffen. Eine Vereinfachung, um mit unabhängigen Feldlösern auszukommen, wird in Wall u. a. [164] vorgeschlagen, mit der einige Beispiele erfolgreich behandelt werden können. Für eine robuste und effiziente Auswertung des inversen Operators \mathcal{F}_Γ^{-1} ist jedoch die Kombination der Feldlöser wie in Algorithmus 16 erforderlich.

Wird anstelle des ALE-Ansatzes das XFEM-Verfahren verwendet, tritt nicht die Gebietsdeformation $\mathbf{d}^{G,n+1}$, sondern die Schnittposition als zusätzliche Unbekannte auf. Die generelle Schwierigkeit beim Auswerten des inversen Operators \mathcal{F}_Γ^{-1} bleibt somit erhalten.

Algorithmus 16 Der nichtlineare Interfaceoperator des Fluidfeldes $\mathbf{d}_\Gamma^{n+1} = \mathcal{F}_\Gamma^{-1}(\mathbf{f}_\Gamma^{n+1})$.

Vorgeschriebene Geschwindigkeiten $\mathbf{u}_{D,0}^{F,n+1}$ ermitteln

Externe Lasten $\mathbf{f}_{\text{ext}}^{F,n+1}$ berechnen

Newton-Schleife mit $0 \leq i < \max_{\text{nliter}}$:

$$\begin{bmatrix} \mathbf{F}_{\text{II}} & \mathbf{F}_{\text{I}\Gamma} + \frac{\Delta t}{2} \mathbf{F}_{\text{I}\Gamma}^{\text{G}} & \mathbf{F}_{\text{II}}^{\text{G}} \\ \mathbf{F}_{\text{I}\Gamma} & \mathbf{F}_{\text{I}\Gamma} + \frac{\Delta t}{2} \mathbf{F}_{\text{I}\Gamma}^{\text{G}} & \mathbf{F}_{\text{I}\Gamma}^{\text{G}} \\ \mathbf{0} & \frac{\Delta t}{2} \mathbf{A}_{\text{I}\Gamma} & \mathbf{A}_{\text{II}} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_{\text{I},i}^{F,n+1} \\ \Delta \mathbf{u}_{\text{I},i}^{G,n+1} \\ \Delta \mathbf{d}_{\text{I},i}^{G,n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\text{I,ext}}^{F,n+1} - \mathbf{f}_{\text{I},i}^{F,n+1} [-\Delta t \mathbf{F}_{\text{I}\Gamma}^{\text{G}} \mathbf{u}_{\text{I}}^{F,n}]_0 \\ \mathbf{f}_{\text{I,ext}}^{F,n+1} - \mathbf{f}_{\text{I},i}^{F,n+1} [-\Delta t \mathbf{F}_{\text{I}\Gamma}^{\text{G}} \mathbf{u}_{\text{I}}^{F,n}]_0 - \mathbf{f}_\Gamma^{n+1} \\ [-\Delta t \mathbf{A}_{\text{I}\Gamma} \mathbf{u}_\Gamma^{F,n}]_0 \end{bmatrix}$$

$$\begin{aligned} \mathbf{u}_{\text{I},i+1}^{F,n+1} &= \mathbf{u}_{\text{I},i}^{F,n+1} + \Delta \mathbf{u}_{\text{I},i}^{F,n+1} \\ \mathbf{d}_{\text{I},i+1}^{G,n+1} &= \mathbf{d}_{\text{I},i}^{G,n+1} + \Delta \mathbf{d}_{\text{I},i}^{G,n+1} \\ \mathbf{d}_{\text{I},i+1}^{G,n+1} &= \mathbf{d}_{\text{I},i}^{G,n+1} + \frac{\Delta t}{2} \Delta \mathbf{u}_{\text{I},i}^{F,n+1} [+ \Delta t \mathbf{A}_{\text{I}\Gamma} \mathbf{u}_\Gamma^{F,n}]_0 \end{aligned}$$

Die in eckigen Klammern angegebenen Terme in Algorithmus 16 sind lediglich im ersten Durchlauf der Newton-Schleife mit $i = 0$ auszuwerten.

6.1.4. Partitionierte Verfahren. Die Lösung des Gesamtproblems muß sowohl die Einzelfelder für sich als auch die Kopplungsbedingung (6.1.1) erfüllen. Mit Hilfe der Interfaceoperatoren (6.1.3) und (6.1.6) läßt sich das Gleichgewicht am Interface als nichtlineares Problem mit einer unbekanntem Interfaceverschiebung \mathbf{d}_Γ^{n+1} anschreiben

$$\mathcal{S}_\Gamma(\mathbf{d}_\Gamma^{n+1}) = \mathcal{F}_\Gamma(\mathbf{d}_\Gamma^{n+1}). \quad (6.1.11)$$

Da die Interfaceoperatoren die Lösung des jeweiligen Feldes implizieren, ist die Gleichung (6.1.11) eine Bestimmungsgleichung für das nichtlineare FSI Problem. Analog kann auch die kinematische Kopplungsbedingung verwendet werden, um ein nichtlineares Problem für die unbekanntem Interfacekräfte \mathbf{f}_Γ^{n+1} zu bekommen

$$\mathcal{S}_\Gamma^{-1}(\mathbf{f}_\Gamma^{n+1}) = \mathcal{F}_\Gamma^{-1}(\mathbf{f}_\Gamma^{n+1}). \quad (6.1.12)$$

Die Interfaceoperatoren sind nur implizit definiert, die Gleichungen (6.1.11) bzw. (6.1.12) können nicht aufgelöst werden. Lösungsverfahren, die auf der Elimination eines der Felder beruhen, sind Sonderfälle, die hier nicht betrachtet werden. Die partitionierten Lösungsverfahren basieren deshalb auf dem Einsetzen von Versuchslösungen. Beim Einsetzen einer Versuchsverschiebung $\mathbf{d}_{\Gamma,0}^{n+1}$ in Gleichung (6.1.11) erhält man ein Kraftresiduum

$$\Delta \mathbf{f}_{\Gamma,0}^{n+1} = \mathcal{S}_\Gamma(\mathbf{d}_{\Gamma,0}^{n+1}) - \mathcal{F}_\Gamma(\mathbf{d}_{\Gamma,0}^{n+1}), \quad (6.1.13)$$

während beim Einsetzen einer Versuchskraft $\mathbf{f}_{\Gamma,0}^{n+1}$ in Gleichung (6.1.12) ein Verschiebungsresiduum entsteht

$$\Delta \mathbf{d}_{\Gamma,0}^{n+1} = \mathcal{S}_\Gamma^{-1}(\mathbf{f}_{\Gamma,0}^{n+1}) - \mathcal{F}_\Gamma^{-1}(\mathbf{f}_{\Gamma,0}^{n+1}). \quad (6.1.14)$$

Um zu einem Lösungsverfahren zu kommen, muß das Residuum in die Einheit der Unbekannten überführt werden, so daß die Unbekannte mit Hilfe des Differenzvektors verbessert werden kann. Dazu ist ein Interfaceoperator erforderlich, der zwischen den Residuen vermittelt

$$\Delta \mathbf{f}_{\Gamma,0}^{n+1} = \mathcal{M}_\Gamma(\Delta \mathbf{d}_{\Gamma,0}^{n+1}). \quad (6.1.15)$$

Der Operator \mathcal{M}_Γ übernimmt eine Rolle analog zu den Vorkonditionierern in iterativen linearen Gleichungslösern. Zur Vorkonditionierung von linearen Gleichungssystemen siehe Abschnitt 3.3.

Im Prinzip kann der Operator \mathcal{M}_Γ beliebig gewählt werden, doch er sollte einen der Feldoperatoren \mathcal{S}_Γ oder \mathcal{F}_Γ nach Möglichkeit gut approximieren. Darüber hinaus muß der

Operator \mathcal{M}_Γ invertierbar sein. Linearität ist nicht erforderlich, da die Feldoperatoren \mathcal{S}_Γ und \mathcal{F}_Γ ebenfalls nichtlinear sind.

In den folgenden motivierenden Herleitungen werden dennoch alle Feldoperatoren nach Bedarf als lineare Operatoren angenommen. Unter dieser Voraussetzung läßt sich mit der Inversen \mathcal{M}_Γ^{-1} des linearen Operators \mathcal{M}_Γ die Gleichung (6.1.13) umformen

$$\Delta \mathbf{d}_{\Gamma,0}^{n+1} = \mathcal{M}_\Gamma^{-1}(\mathcal{S}_\Gamma(\mathbf{d}_{\Gamma,0}^{n+1}) - \mathcal{F}_\Gamma(\mathbf{d}_{\Gamma,0}^{n+1})) \quad (6.1.16)$$

$$= \mathcal{M}_\Gamma^{-1}(\mathcal{S}_\Gamma(\mathbf{d}_{\Gamma,0}^{n+1})) - \mathcal{M}_\Gamma^{-1}(\mathcal{F}_\Gamma(\mathbf{d}_{\Gamma,0}^{n+1})). \quad (6.1.17)$$

Analog wird die Gleichung (6.1.14) mit \mathcal{M}_Γ umgeformt

$$\Delta \mathbf{f}_{\Gamma,0}^{n+1} = \mathcal{M}_\Gamma(\mathcal{S}_\Gamma^{-1}(\mathbf{f}_{\Gamma,0}^{n+1}) - \mathcal{F}_\Gamma^{-1}(\mathbf{f}_{\Gamma,0}^{n+1})) \quad (6.1.18)$$

$$= \mathcal{M}_\Gamma(\mathcal{S}_\Gamma^{-1}(\mathbf{f}_{\Gamma,0}^{n+1})) - \mathcal{M}_\Gamma(\mathcal{F}_\Gamma^{-1}(\mathbf{f}_{\Gamma,0}^{n+1})). \quad (6.1.19)$$

Allgemein lassen sich diese Gleichungen auch mit nichtlinearen Operatoren \mathcal{M}_Γ anschreiben, die Herleitung aus den Gleichungen (6.1.16) bzw. (6.1.18) dient dabei lediglich der Motivation. Die vorkonditionierten Residuen $\Delta \mathbf{d}_{\Gamma,0}^{n+1}$ bzw. $\Delta \mathbf{f}_{\Gamma,0}^{n+1}$ sind um so besser als Lösungsincrement geeignet, je besser die Operatoren \mathcal{M}_Γ und \mathcal{M}_Γ^{-1} die Feldoperatoren approximieren.

Beide Gleichungen (6.1.17) und (6.1.19) sind mögliche Ausgangspunkte für das Entwickeln von Lösungsverfahren für das partitionierte FSI Problem, die erste verwendet die Verschiebung $\mathbf{d}_{\Gamma,0}^{n+1}$ als Unbekannte, die zweite die Kraft $\mathbf{f}_{\Gamma,0}^{n+1}$. Mit der Wahl der Operatoren \mathcal{M}_Γ bzw. \mathcal{M}_Γ^{-1} erhält man nun ganz unterschiedliche partitionierte Lösungsansätze.

6.1.5. Dirichlet-Neumann-Partitionierung. Eine naheliegende Möglichkeit ist es, für \mathcal{M}_Γ die bekannten Interfaceoperatoren einzusetzen. Mit der Wahl $\mathcal{M}_\Gamma = \mathcal{S}_\Gamma$ erhält man aus Gleichung (6.1.17)

$$\Delta \mathbf{d}_{\Gamma,0}^{n+1} = \mathbf{d}_{\Gamma,0}^{n+1} - \mathcal{S}_\Gamma^{-1}(\mathcal{F}_\Gamma(\mathbf{d}_{\Gamma,0}^{n+1})) \quad (6.1.20)$$

und entsprechend aus Gleichung (6.1.19)

$$\Delta \mathbf{f}_{\Gamma,0}^{n+1} = \mathbf{f}_{\Gamma,0}^{n+1} - \mathcal{S}_\Gamma(\mathcal{F}_\Gamma^{-1}(\mathbf{f}_{\Gamma,0}^{n+1})). \quad (6.1.21)$$

An dieser Stelle wurde die Beziehung $\mathbf{d}_{\Gamma,0}^{n+1} = \mathcal{S}_\Gamma^{-1}(\mathcal{S}_\Gamma(\mathbf{d}_{\Gamma,0}^{n+1}))$ eingesetzt, die exakt auch nur für lineare Operatoren gilt. Mit nichtlinearen Operatoren ist die Herleitung der Gleichungen (6.1.20) und (6.1.21) so nicht zulässig, die Gleichungen selbst sind jedoch sinnvolle Berechnungsvorschriften, um aus einer gegebenen Interfaceverschiebung ein Verschiebungsincrement zu bestimmen bzw. aus einer gegebenen Interfacekraft ein Kraftincrement. Dabei werden beide Operatoren nur einmal ausgewertet, beide nichtlinearen Felder werden nur einmal gelöst, das eine Feld mit Dirichlet-Randbedingungen, das andere mit Neumann-Randbedingungen.

In Gleichung (6.1.20) übernimmt das Fluidfeld die Rolle der Dirichletpartition, das Strukturfeld stellt entsprechend die Neumannpartition dar. In Gleichung (6.1.21) ist die Zuordnung gerade umgekehrt. In beiden Fällen spricht man deshalb von einer Dirichlet-Neumann-Partitionierung des Gesamtproblems.

Der mechanische Inhalt einer Dirichlet-Neumann-Partitionierung ist die Behauptung, das Dirichlet-Feld habe eine starre Wand, die beliebige Kräfte aushält, während das Neumann-Feld am Interface eine Last bekommt und auf diese Last reagiert, ohne eine Abminderung der Last aufgrund der eigenen Nachgiebigkeit in Betracht zu ziehen. Diese harte Annahme bedingt die Neigung von Dirichlet-Neumann-Lösern, über das Ziel hinaus zu schießen.

Mit der alternativen Wahl $\mathcal{M}_\Gamma = \mathcal{F}_\Gamma$ und der Annahme $\mathbf{d}_\Gamma^{n+1} = \mathcal{F}_\Gamma^{-1}(\mathcal{F}_\Gamma(\mathbf{d}_\Gamma^{n+1}))$ erhält man in gleicher Weise aus Gleichung (6.1.17) die Beziehung

$$\Delta \mathbf{d}_{\Gamma,0}^{n+1} = \mathcal{F}_\Gamma^{-1}(\mathcal{S}_\Gamma(\mathbf{d}_{\Gamma,0}^{n+1})) - \mathbf{d}_{\Gamma,0}^{n+1} \quad (6.1.22)$$

und aus Gleichung (6.1.19)

$$\Delta \mathbf{f}_{\Gamma,0}^{n+1} = \mathcal{F}_\Gamma(\mathcal{S}_\Gamma^{-1}(\mathbf{f}_{\Gamma,0}^{n+1})) - \mathbf{f}_{\Gamma,0}^{n+1}. \quad (6.1.23)$$

Auch die Gleichungen (6.1.22) und (6.1.23) stellen Dirichlet-Neumann-Partitionierungen dar.

Es ist jedoch algorithmisch und numerisch sehr aufwendig, den inversen Fluidoperator \mathcal{F}_Γ^{-1} auszuwerten. Die algorithmische Schwierigkeit ist bedingt durch die enge Verzahnung von Fluid- und ALE-Löser, siehe Algorithmus 16. Numerische Schwierigkeiten entstehen durch die Physik der Fluid- und Strukturfelder. Das Fluidfeld in \mathcal{F}_Γ^{-1} reagiert in aller Regel sehr sensibel und mit großen Interfaceverschiebungen auf kleine Schwankungen der Interfacelasten, während auch das Strukturfeld auf vorgeschriebene Verschiebungen mit großen Kräften reagiert. Ein auf (6.1.21) oder (6.1.22) basierender partitionierter Lösungsalgorithmus benötigt deshalb eine ausgesprochen große Dämpfung, so daß erhöhte Anforderungen an die Numerik der beteiligten Feldlöser entstehen.

Das Fluidfeld als Neumannpartition einzusetzen und das Strukturfeld als Dirichletpartition hat sich für physikalische Materialparameter bisher nicht bewährt, siehe beispielsweise Küttler u. a. [96] oder Küttler und Wall [100]. Die Gleichungen (6.1.21) und (6.1.22) werden deshalb kaum für FSI Lösungsverfahren verwendet.

Übrig bleibt die Interfacegleichung (6.1.20) mit unbekanntem Verschiebungen \mathbf{d}_Γ^{n+1} und die Gleichung (6.1.23) mit unbekanntem Kräften \mathbf{f}_Γ^{n+1} . Beide bauen auf den gleichen Interfaceoperatoren auf. Beide können völlig gleichwertig für Lösungsverfahren verwendet werden.

Lösungsverfahren, die auf einer Dirichlet-Neumann-Partitionierung basieren, sind Block-Gauß-Seidel-Verfahren mit nichtlinearen Blöcken. Im besonderen hängen die Interfaceoperatoren voneinander ab und müssen nacheinander ausgewertet werden. Man spricht deshalb auch von einer Multiplikativ-Schwarz-Zerlegung des Gesamtgebiets. Ausgehend von Gleichung (6.1.20) ist die Bedingung für die Lösung \mathbf{d}_Γ^{n+1}

$$\mathbf{d}_\Gamma^{n+1} = \mathcal{S}_\Gamma^{-1}(\mathcal{F}_\Gamma(\mathbf{d}_\Gamma^{n+1})), \quad (6.1.24)$$

analog erhält man aus Gleichung (6.1.23) die Bedingung für \mathbf{f}_Γ^{n+1}

$$\mathbf{f}_\Gamma^{n+1} = \mathcal{F}_\Gamma(\mathcal{S}_\Gamma^{-1}(\mathbf{f}_\Gamma^{n+1})). \quad (6.1.25)$$

Diese Bedingungen sind Fixpunkt-Formulierungen des FSI Problems, die das Verwenden der Lösungsverfahren aus Kapitel 3.5 nahelegen. Diese Lösungsansätze sind der Gegenstand von Kapitel 7.

Die Dirichlet-Neumann-Partitionierung ist der bei partitionierten FSI Lösern dominierende Ansatz. Die vielfach zitierten Arbeiten von Wall u. a. [163], Mok und Wall [118] oder auch Le Tallec und Mouro [103] und die Vielzahl der darauf aufbauenden Arbeiten setzen die Dirichlet-Neumann-Partitionierung zur Trennung der Felder ein. Eine neuere Arbeit, bei der die Dirichlet-Neumann-Partitionierung im FSI Kontext an einem Modellsystem untersucht wird, ist Joosten u. a. [87].

6.1.6. Neumann-Neumann-Partitionierung. Nachdem der Struktur- und der Fluidoperator als Operator \mathcal{M}_Γ geeignet sind, läßt sich auch die Linearkombination beider Operatoren $\mathcal{M}_\Gamma = \alpha_S \mathcal{S}_\Gamma + \alpha_F \mathcal{F}_\Gamma$ mit beliebigen Parametern α_S und α_F verwenden. Werden

beide Parameter zu 1 gesetzt und lineare Operatoren angenommen, erhält man aus Gleichung (6.1.17) den Zusammenhang

$$\Delta \mathbf{d}_{\Gamma,0}^{n+1} = (\mathcal{S}_{\Gamma}^{-1} + \mathcal{F}_{\Gamma}^{-1})(\mathcal{S}_{\Gamma}(\mathbf{d}_{\Gamma,0}^{n+1})) - (\mathcal{S}_{\Gamma}^{-1} + \mathcal{F}_{\Gamma}^{-1})(\mathcal{F}_{\Gamma}(\mathbf{d}_{\Gamma,0}^{n+1})) \quad (6.1.26)$$

$$= \mathbf{d}_{\Gamma,0}^{n+1} + \mathcal{F}_{\Gamma}^{-1}(\mathcal{S}_{\Gamma}(\mathbf{d}_{\Gamma,0}^{n+1})) - \mathcal{S}_{\Gamma}^{-1}(\mathcal{F}_{\Gamma}(\mathbf{d}_{\Gamma,0}^{n+1})) - \mathbf{d}_{\Gamma,0}^{n+1} \quad (6.1.27)$$

$$= \mathcal{F}_{\Gamma}^{-1}(\mathcal{S}_{\Gamma}(\mathbf{d}_{\Gamma,0}^{n+1})) - \mathcal{S}_{\Gamma}^{-1}(\mathcal{F}_{\Gamma}(\mathbf{d}_{\Gamma,0}^{n+1})) \quad (6.1.28)$$

und analog erhält man aus Gleichung (6.1.19) mit der Wahl $\mathcal{M}_{\Gamma}^{-1} = \mathcal{S}_{\Gamma}^{-1} + \mathcal{F}_{\Gamma}^{-1}$

$$\Delta \mathbf{f}_{\Gamma,0}^{n+1} = (\mathcal{S}_{\Gamma} + \mathcal{F}_{\Gamma})(\mathcal{S}_{\Gamma}^{-1}(\mathbf{f}_{\Gamma,0}^{n+1})) - (\mathcal{S}_{\Gamma} + \mathcal{F}_{\Gamma})(\mathcal{F}_{\Gamma}^{-1}(\mathbf{f}_{\Gamma,0}^{n+1})) \quad (6.1.29)$$

$$= \mathbf{f}_{\Gamma,0}^{n+1} + \mathcal{F}_{\Gamma}(\mathcal{S}_{\Gamma}^{-1}(\mathbf{f}_{\Gamma,0}^{n+1})) - \mathcal{S}_{\Gamma}(\mathcal{F}_{\Gamma}^{-1}(\mathbf{f}_{\Gamma,0}^{n+1})) - \mathbf{f}_{\Gamma,0}^{n+1} \quad (6.1.30)$$

$$= \mathcal{F}_{\Gamma}(\mathcal{S}_{\Gamma}^{-1}(\mathbf{f}_{\Gamma,0}^{n+1})) - \mathcal{S}_{\Gamma}(\mathcal{F}_{\Gamma}^{-1}(\mathbf{f}_{\Gamma,0}^{n+1})) \quad (6.1.31)$$

Die Gleichungen (6.1.28) und (6.1.31) stellen Linearkombinationen der Dirichlet-Neumann-Partitionierungen (6.1.21) und (6.1.23) bzw. (6.1.20) und (6.1.22) dar und verlangen, jedes Feld zweimal zu lösen. Diese Ansätze sind als Neumann-Neumann-Partitionierungen bekannt. Als Lösungsverfahren für FSI Probleme sind Neumann-Neumann-Partitionierungen nicht im Einsatz, da bei diesen Verfahren das Fluidfeld auch mit Neumann-Randbedingung am Interface gelöst werden muß. Deparis u. a. [30] enthält eine numerische Untersuchung zu Neumann-Neumann-Verfahren für FSI Probleme.

6.1.7. Allgemeine Jacobi-Verfahren. Alternativ können beliebige Näherungen der Feldlöser für die Operatoren \mathcal{M}_{Γ} bzw. $\mathcal{M}_{\Gamma}^{-1}$ in den Gleichungen (6.1.18) und (6.1.16) verwendet werden. In diesem Fall können der Fluid- und der Strukturoperator gleichzeitig ausgewertet werden, man hat es mit einem nichtlinearen Block-Jacobi-Verfahren bzw. einer Additiv-Schwarz-Zerlegung des Gebiets zu tun.

Im Vergleich zu den Block-Gauß-Seidel-Verfahren (6.1.20) und (6.1.23) entsteht jedoch zusätzlicher numerischer Aufwand für die Auswertungen des Operators \mathcal{M}_{Γ} . Gleichzeitig bleibt die Konvergenz eines Jacobi-Verfahrens hinter der eines Gauß-Seidel-Verfahrens zurück. Block-Jacobi-Verfahren sind zur Lösung partitionierter FSI Probleme deshalb kaum im Einsatz, das gleichzeitige Auswerten der Interfaceoperatoren ist bei praktischen Berechnungen nur selten wünschenswert. Block-Jacobi-Verfahren werden in dieser Arbeit nicht weiter betrachtet.

6.1.8. Robin-Verfahren. Die in Badia u. a. [3] sowie Nobile und Vergara [119] diskutierten Robin-Verfahren, siehe dazu auch Maier [105], basieren auf dem Wunsch, die harte Annahme der Dirichlet-Neumann-Partitionierung zu vermeiden und dennoch partitionierte Lösungsverfahren zu verwenden. Dazu werden an den Kopplungsrändern der Felder Robin-Randbedingungen verwendet. Robin-Randbedingungen sind Linearkombinationen aus Dirichlet- und Neumann-Randbedingungen, bei denen die vorgeschriebene Last $\mathbf{f}_{\Gamma}^{n+1}$ um eine vorgeschriebene Verschiebung $\mathbf{d}_{\Gamma}^{n+1}$ ergänzt wird

$$\mathbf{f}_{\Gamma,R}^{n+1} = \mathbf{f}_{\Gamma}^{n+1} + \alpha_R \mathbf{d}_{\Gamma}^{n+1}. \quad (6.1.32)$$

Das Feldproblem wird mit einer verschiebungsabhängigen Robin-Last $\mathbf{f}_{\Gamma,R}^{n+1}$ im Sinne einer Neumann-Randbedingung belastet, der Parameter α_R übernimmt die Funktion einer Federkonstante.

Dieser Typ Randbedingung kann zur Lösung von partitionierten FSI Problemen eingesetzt werden, indem der Parameter α_R in einem Feld so gewählt wird, daß er das linearisierte Verhalten des jeweils anderen Feldes approximiert. Die harte Dirichlet-Neumann-Partitionierung wird durch diese Information über das andere Feld aufgelöst.

Die für die Dirichlet-Neumann-Partitionierung definierten Interfaceoperatoren können für die Robin-Kopplung nicht verwendet werden, da die kombinierte Robin-Kraft als Randbedingung zwischen den Feldern ausgetauscht werden muß.

Für allgemeine FSI Probleme ist unklar, wie der Parameters α_R zu wählen ist. Numerische Experimente zeigen, daß ein einzelner skalarer Parameter für komplexe Problemstellungen nicht ausreicht. Für sinnvolle Berechnungen wird für jeden Freiheitsgrad am Interface ein eigener Parameter benötigt. Wie diese Parameter aus dem Berechnungsverlauf abgeschätzt werden können, ist gegenwärtig offen. Gleichzeitig verlangt die Robin-Randbedingung beim Lösen des Fluidfeldes die Integration der Navier-Stokes-Gleichung über ein unbekanntes Gebiet, ganz analog zur Neumann-Randbedingung beim inversen Fluidoperator \mathcal{F}_Γ^{-1} . Robin-Verfahren werden deshalb in der vorliegenden Arbeit nicht weiter betrachtet.

6.2. Exegese

Die Dirichlet-Neumann-Partitionierung mit der unbekanntem Interfaceverschiebung \mathbf{d}_Γ^{n+1} führt auf das nichtlineare Interfaceproblem (6.1.24). Dieses Interfaceproblem gilt es zu lösen. Analog zu Gleichung (6.1.24) kann natürlich auch Gleichung (6.1.25) mit der Interfacekraft \mathbf{f}_Γ^{n+1} als Unbekannter verwendet werden. Alternativen zur Dirichlet-Neumann-Partitionierung sind für praktische FSI Lösungsalgorithmen nicht relevant.

In Mok und Wall [118] wurden die Lösungsverfahren für partitionierte FSI Probleme als Kopplungsmethoden vorgestellt. Gekoppelt wurden die unabhängigen Feldlöser. Das Interfaceproblem (6.1.24) wurde dabei nur implizit formuliert. In der Folge ist eine große Zahl kreativer Kopplungsmethoden erdacht worden, mit der Absicht, den Lösungsaufwand für das gekoppelte Problem zu verringern. Da jedoch jeder Autor andere Feldlöser und eine jeweils eigene Notation verwendet, stellt der Vergleich und die Auswahl der Kopplungsmethoden eine Schwierigkeit dar.

Die Diskussion um die Kopplungsmethoden für das partitionierte FSI Problem wird hinfällig, wenn von der expliziten Darstellung des Interfaceproblems (6.1.24) ausgegangen wird. Damit werden die bekannten Lösungsmethoden für nichtlineare Probleme anwendbar. Die Kopplung der Feldlöser und die Lösung des gekoppelten Problems werden zwei verschiedene Aufgaben. Die Kopplung ist die Folge der Partitionierung und wird in Abschnitt 6.1 diskutiert. Allgemeine Lösungsmethoden für nichtlineare Probleme werden in Kapitel 3 vorgestellt, die Anwendung der Lösungsmethoden auf das nichtlineare Interfaceproblem (6.1.24) wird in Kapitel 7 diskutiert. Kopplungsmethoden existieren in dieser Sichtweise nicht.

Mit diesem Perspektivwechsel ist die Frage um die Lösungsverfahren für partitionierte FSI Probleme beantwortet. Die anwendbaren Lösungsverfahren sind bekannt. Es bleiben lediglich einige Besonderheiten beim Anwenden der Verfahren zu besprechen und, als beinahe scholastische Übung, die in der Literatur vorgeschlagenen Kopplungsmethoden als Lösungsmethoden für nichtlineare Probleme zu erkennen und einzuordnen. Diese Gedanken wurden bereits in Küttler und Wall [98, 99] dargestellt.

Dieser Wechsel der Perspektive ist demnach gleichzeitig sehr schlicht und sehr weitreichend. In Mok und Wall [118] und in Küttler und Wall [98] werden die gleichen Lösungsverfahren auf unterschiedliche Weise beschrieben. Das in Mok und Wall [118] vorgestellte Verfahren steht für sich und ist eng an die FSI Problemstellung gebunden. Das Verfahren in Küttler und Wall [98] ist allgemein, es wird lediglich eine Anwendung des allgemeinen Lösungsverfahrens am Beispiel des partitionierten FSI Problems vorgeführt. Der Unterschied zwischen beiden Beschreibungen liegt in der Abstraktion der problemspezifischen Details in den Interfaceoperatoren \mathcal{S}_Γ^{-1} und \mathcal{F}_Γ .

Algorithmus 17 Basisversion des stark koppelnden Lösungsverfahrens.

für alle $i = 0, \dots, \max_{\text{nliter}}$:

$$\tilde{\mathbf{d}}_{\Gamma,i}^{n+1} = \mathcal{S}_{\Gamma}^{-1}(\mathcal{F}_{\Gamma}(\mathbf{d}_{\Gamma,i}^{n+1}))$$

$$\mathbf{r}_{\Gamma,i}^{n+1} = \tilde{\mathbf{d}}_{\Gamma,i}^{n+1} - \mathbf{d}_{\Gamma,i}^{n+1}$$

falls $|\mathbf{r}_{\Gamma,i}^{n+1}| \leq \epsilon_{\text{nl}}$:

fertig

berechne $\Delta \mathbf{d}_{\Gamma,i}^{n+1}$

$$\mathbf{d}_{\Gamma,i+1}^{n+1} = \mathbf{d}_{\Gamma,i}^{n+1} + \Delta \mathbf{d}_{\Gamma,i}^{n+1}$$

In dieser Arbeit wird der allgemeine Ansatz von Küttler und Wall [98] verwendet, bei dem das spezifische Interfaceproblem (6.1.24) mit allgemeinen Lösungsmethoden gelöst wird.

6.3. Lösungsverfahren für das partitionierte FSI Problem

Alle Lösungsmethoden für das nichtlineare Interfaceproblem basieren auf einem iterativen Verfahren, bei dem die Interfaceverschiebung $\mathbf{d}_{\Gamma,i}^{n+1}$ schrittweise verbessert wird. Die Grundversion des Lösungsverfahrens ist in Algorithmus 17 angegeben, einer Adaption des allgemeinen Algorithmus 1 für partitionierte FSI Probleme. Spezifisch für das FSI Problem in Algorithmus 17 ist das Berechnen des Interfaceresiduums $\mathbf{r}_{\Gamma,i}^{n+1}$ aus der Interfaceverschiebung $\mathbf{d}_{\Gamma,i}^{n+1}$

$$\mathbf{r}_{\Gamma,i}^{n+1} = \mathcal{S}_{\Gamma}^{-1}(\mathcal{F}_{\Gamma}(\mathbf{d}_{\Gamma,i}^{n+1})) - \mathbf{d}_{\Gamma,i}^{n+1}, \quad (6.3.1)$$

wobei beide Feldoperatoren ausgewertet werden, also beide nichtlinearen Feldprobleme gelöst werden müssen. Das Berechnen des Interfaceresiduums $\mathbf{r}_{\Gamma,i}^{n+1}$ ist deshalb eine sehr aufwendige Operation.

Das Interfaceresiduum $\mathbf{r}_{\Gamma,i}^{n+1}$ ist eine Verschiebungsdifferenz. Wenn die Verschiebungsdifferenz $\mathbf{0}$ ist, ist die in Gleichung (6.3.1) eingesetzte Interfaceverschiebung $\mathbf{d}_{\Gamma,i}^{n+1}$ die Lösung im Zeitschritt. Eine Stagnation der Verschiebungsdifferenz (6.3.1) kann nicht auftreten. Der Konvergenztest

$$|\mathbf{r}_{\Gamma,i}^{n+1}|_2 \leq \epsilon_{\text{nl}} \quad (6.3.2)$$

mit einer hinreichend klein gewählten Toleranz ϵ und der L_2 -Norm des Interfaceresiduums ist deshalb ausreichend. Es wird an dieser Stelle kein Test auf einzelne Fehlerwerte benötigt, da die nichtlinearen Feldlöser für sich die Fehler glätten.

Die Berechnung des Inkrements der Interfaceverschiebung $\Delta \mathbf{d}_{\Gamma,i}^{n+1}$, die in Algorithmus 17 nicht näher spezifiziert ist, unterscheidet die verschiedenen Lösungsmethoden. Diese Lösungsmethoden werden im Detail in Kapitel 7 besprochen.

Mit Algorithmus 17 kann die Lösung des partitionierten FSI Problems mit beliebiger Genauigkeit bestimmt werden, solange alle Einzelfelder mit hinreichender Genauigkeit gelöst werden. Der zusammengesetzte Lösungsalgorithmus 17 stellt somit erhebliche Anforderungen an die Genauigkeit der Feldlöser. Die Abbruchkriterien der Newton-Verfahren der Einzelfelder müssen hinreichend scharf sein, damit die Verschiebungsdifferenz in Gleichung (6.3.1) mit der geforderten Genauigkeit berechnet werden kann. Andernfalls kann das Kriterium (6.3.2) nicht erfüllt werden und Algorithmus 17 stagniert.

Das Lösen der nichtlinearen Feldgleichungen ist jedoch um so aufwendiger, je exakter die Lösung im Feld bestimmt werden soll. Solange die angenommene Interfaceverschiebung nicht stimmt, muß lediglich die Verschiebungsdifferenz (6.3.1) genau genug bestimmt werden, um das nächste Inkrement der Interfaceverschiebung $\Delta \mathbf{d}_{\Gamma,i}^{n+1}$ bestimmen zu können.

Damit entsteht die Frage, wie die Abbruchkriterien der nichtlinearen Feldlöser auf das Abbruchkriterium der Interface-Iteration (6.3.2) abzustimmen sind.

Das Kriterium (6.3.2) testet Verschiebungsdifferenzen. Um die eingesetzten Toleranzen vergleichen zu können, kann man in den Abbruchkriterien der Newton-Verfahren der Feldlöser auch die Inkremente der Verschiebung bzw. Geschwindigkeit berücksichtigen. Damit wären die zulässigen Fehler auf beiden Ebenen zumindest in der gleichen Größenordnung. In den Feldlösern sind natürlich geringere Toleranzen als auf der Interface-Ebene zu wählen.

Bei praktischen Berechnungen zeigt sich, daß zu geringe Toleranzen in den Feldlösern weniger Aufwand verursachen als zu große. Bei der Berechnung von FSI Problemen mit partitionierten Feldlösern werden deshalb in aller Regel sehr scharfe Abbruchkriterien in den Feldern gewählt.

6.3.1. Schwache und starke Kopplung. In einigen Anwendungsgebieten kann der Lösungsalgorithmus 17 so formuliert werden, daß bereits nach einem Lösungsschritt eine hinreichende Näherung an die Lösung gefunden ist. Das ist insbesondere bei Problemstellungen der Fall, bei denen die Dirichlet-Neumann-Partitionierung nahe an der Physik bleibt. Beispielsweise treffen bei der Aeroelastizität vergleichsweise steife Strukturen auf leichte Fluide. Bei der Partitionierung in ein Fluidgebiet mit Dirichlet-Rand und ein Strukturgebiet mit Neumann-Rand wird in diesem Fall nur ein kleiner Fehler eingebracht. Die Interaktion wird sehr stark von der Struktur dominiert. Lösungsmethoden für die Aufgaben der Aeroelastizität werden beispielsweise in Farhat [45] diskutiert.

Für den Konvergenztest (6.3.2) in Algorithmus 17 müssen jedoch die Einzelfelder erneut gelöst werden. Bei FSI Problemen, die zuverlässig in einem Lösungsschritt konvergieren, erfordert Algorithmus 17 deshalb zwei Auswertungen des Interfaceresiduums. In dem Wissen, daß ohnehin nur ein Lösungsschritt erforderlich ist, kann in diesen Fällen die Iteration und die Prüfung des Residuums in Algorithmus 17 weggelassen werden.

Lösungsverfahren für partitionierte FSI Probleme, die nur einen Lösungsschritt für das nichtlineare Interfaceproblem verwenden und danach, ohne das Residuum zu prüfen, zum nächsten Zeitschritt übergehen, werden als schwach koppelnde Verfahren bezeichnet. Demgegenüber sind Verfahren, die die Lösung nach Algorithmus 17 bis zu einer vorgegebenen Toleranz ausiterieren, stark koppelnd.

Schwach koppelnde Verfahren koppeln das Fluidfeld an das Strukturfeld und versetzen die Rückkopplung vom Fluid zur Struktur um einen Zeitschritt. Diese Art der Kopplung bringt eine explizite Komponente in den Lösungsalgorithmus ein, auch wenn beide Felder für sich voll implizit gelöst werden. Der beim schwachen Koppeln entstehende Fehler kann über einen passend gewählten Prädiktor für die Versuchsverschiebung $\mathbf{d}_{r,0}^{z+1}$ minimiert werden. Mit den schwach koppelnden Verfahren stehen für bestimmte Aufgabengebiete sehr effiziente Lösungsmethoden bereit.

Natürlich ist die Genauigkeit der Lösung bei der schwachen Kopplung begrenzt. Darüber hinaus stellt sich aber auch die Frage nach der Stabilität des Verfahrens über die Zeitschritte. Für inkompressible Fluide sind schwach koppelnde Verfahren stets instabil, siehe dazu Causin u. a. [17] und auch Förster u. a. [52]. Schwach koppelnde Verfahren werden daher in dieser Arbeit nicht weiter betrachtet.

Die stark koppelnden Verfahren nach Algorithmus 17 sind implizite Verfahren, die genauso wie die in Kapitel 8 vorgestellten monolithischen Verfahren die exakte Lösung des FSI Problems im Rahmen der gewählten Toleranz finden.

6.3.2. Semi-implizite Verfahren. Eine zwischen den schwach koppelnden und den stark koppelnden Verfahren angeordnete Variante sind die semi-impliziten Kopplungsverfahren, die beispielsweise in Badia und Codina [4] oder Badia u. a. [5] besprochen werden. Eine umfangreiche Übersicht dieser Verfahren wurde von Wiesner [166] zusammengestellt.

Die semi-impliziten Kopplungsverfahren verwenden einen alternativen Fluidoperator, der die Lösung des Fluidfeldes mit Hilfe eines der Druckkorrektur-Verfahren aus Kapitel 4.3.12 bestimmt. Damit ist das gekoppelte Verfahren für sich ein Näherungsverfahren, so daß eine exakte Kopplungsiteration im Sinne von Algorithmus 17 unnötig ist. Stattdessen wird die Kopplung auf Fluidseite auf die Druckprojektionsgleichung beschränkt, um dem in Causin u. a. [17] und Förster u. a. [52] diskutierten added-mass-Effekt entgegenzuwirken. Dabei werden gerade so viele Kopplungsiterationen ausgeführt, wie nötig sind, um die Stabilität des Algorithmus zu erhalten.

6.4. Nichtpassende Netze

Die bisher betrachteten Übertragungen zwischen Fluid- und Strukturpartition gehen von passenden Netzen am Interface aus. Die Kopplungsverfahren können jedoch sehr leicht um Projektionsoperatoren erweitert werden, um nicht passende Interfacenetze zu ermöglichen. Dazu ist das Übertragen der Interfaceverschiebungen \mathbf{d}_Γ^{n+1} und Interfacekräfte \mathbf{f}_Γ^{n+1} zwischen den Netzen lediglich dem Fluidoperator zuzuschlagen.

Die Interfacegleichung (6.1.24) und die darauf aufbauenden Lösungsverfahren sind von diesem Eingriff nicht betroffen, in derselben Weise wie auch der in Abschnitt 6.1.3 besprochene XFEM-basierte Fluidoperator keine Auswirkung auf die Formulierung der Interfacegleichung (6.1.24) hat.

Eine sehr einfache interpolationsbasierte Übertragung der Netzdaten wurde von Farhat u. a. [46] angegeben. Siehe dazu auch die Implementierung von Henke [75]. In der vorliegenden Arbeit werden nicht passende Netze nicht weiter berücksichtigt.

Lösungsmethoden für Dirichlet-Neumann partitionierte FSI Probleme

Die Dirichlet-Neumann-Partitionierung führt zu einem nichtlinearen Problem mit unbekannten Interfaceverschiebungen (6.1.24), das wiederum auf den in Kapitel 4 vorgestellten nichtlinearen Feldproblemen basiert. Die nichtlinearen Feldprobleme werden von den Interfaceoperatoren \mathcal{S}_Γ^{-1} und \mathcal{F}_Γ abstrahiert. Das Auswerten der Operatoren wurde in Kapitel 6 im Detail besprochen. Das Lösen des Interfaceproblems (6.1.24) ist Gegenstand dieses Kapitels.

Um eine Vorstellung von der Arbeitsweise der Lösungsmethoden für Dirichlet-Neumann partitionierte FSI Probleme zu erhalten, wird ein skalares lineares Modell betrachtet. Abbildung 7.0.1 zeigt das aus zwei linearen Federn bestehende Modell, dessen einziger Freiheitsgrad die horizontale Verschiebung d am Interface ist. Sowohl die Fluidfeder als auch die Strukturfeder reagieren auf Interfaceverschiebungen mit Kräften. Die skalaren Feldoperatoren sind

$$f = F(d) \quad \text{und} \quad f = S(d). \quad (7.0.1)$$

Dieses Modell erzeugt im Verschiebungs-Kraft-Diagramm zwei sich schneidende Geraden. Das Ziel jedes Lösungsverfahrens ist es, den Schnittpunkt dieser Geraden zu finden.

Ein solches Modell bildet natürlich keinerlei Charakteristiken der FSI ab. Das grotesk vereinfachte Modell wurde gewählt, um die Partitionierung und die unterschiedlichen Lösungsansätze veranschaulichen zu können. Natürlich sind die Lösungsalgorithmen erst im Kontext des eigentlichen FSI Problems relevant, die Lösung des linearen Modellproblems kann explizit angegeben werden.

7.1. Fixpunkt-Verfahren

Fixpunkt-Verfahren sind die einfachsten Lösungsverfahren für Dirichlet-Neumann partitionierten FSI Probleme. Ganz allgemein wurden diese Verfahren bereits in Kapitel 3.5 vorgestellt. Wegen ihrer Einfachheit wurden Fixpunkt-Verfahren zum Lösen des Dirichlet-Neumann partitionierten FSI Problems vielfach eingesetzt. Angefangen bei Wall u. a. [163], Le Tallec und Mouro [103], Mok und Wall [118] und der neueren Darstellung in Küttler und Wall [98] bis hin zu Sternel u. a. [145] wurden unzählige Verfahrensvarianten vorgeschlagen.

Wie in Kapitel 3.5 diskutiert, erfordern Fixpunkt-Verfahren das Auswerten einer Fixpunkt-Gleichung (3.5.1). Das Interfaceproblem (6.1.24) der partitionierten FSI liegt bereits als

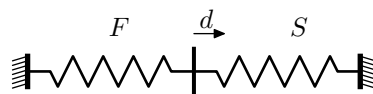


Abbildung 7.0.1: Skalares lineares Modell für ein partitioniertes Fluid-Struktur-Problem.

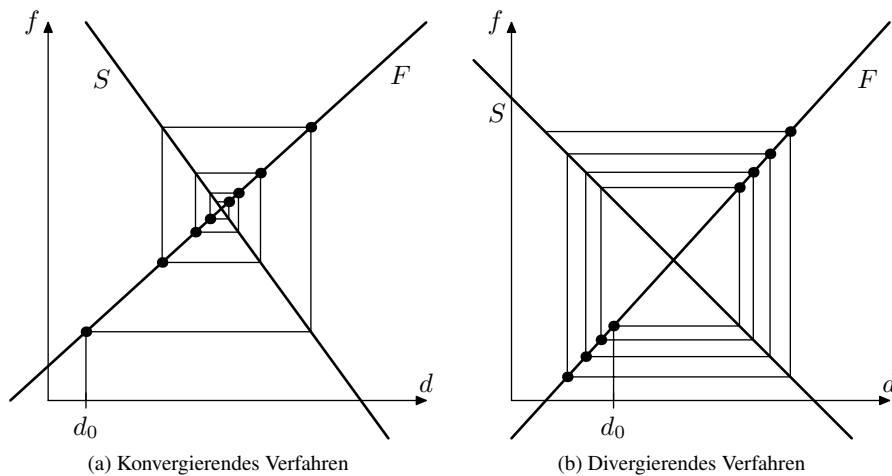


Abbildung 7.1.1: Das Fixpunkt-Verfahren am linearen Modellproblem veranschaulicht

Fixpunkt-Gleichung vor. Im Basisalgorithmus 17 wird deshalb das über Gleichungen (6.3.1) berechnete Verschiebungsresiduum $\mathbf{r}_{\Gamma,i}^{n+1}$ als Verschiebungssinkrement verwendet

$$\Delta \mathbf{d}_{\Gamma,i}^{n+1} = \mathbf{r}_{\Gamma,i}^{n+1}. \quad (7.1.1)$$

Das Verschiebungsresiduum $\mathbf{r}_{\Gamma,i}^{n+1}$ stellt damit sowohl die Richtung als auch die Schrittweite des Fixpunkt-Schritts dar. Das vollständige Fixpunkt-Verfahren entsteht aus dem um das Verschiebungssinkrement (7.1.1) ergänzten Basisalgorithmus 17.

Diese Fixpunkt-Verfahren sind Block-Gauß-Seidel-Verfahren, also Blockvarianten des in Kapitel 3.3.1 vorgestellten Gauß-Seidel-Verfahrens. Beim FSI Problem existieren genau zwei Blöcke, der Strukturoperator und der Fluidoperator. Auch wird das Gauß-Seidel-Verfahren hier auf nichtlineare Operatoren angewendet, das Verfahrensprinzip bleibt jedoch das gleiche wie in Kapitel 3.3.1. Für einen direkten Vergleich der Verfahren siehe die Motivation der Dirichlet-Neumann-Partitionierung in Kapitel 6.1.5.

Für das Modellproblem läßt sich die Fixpunkt-Gleichung analog mit den linearen Operatoren (7.0.1) ausdrücken

$$d_{i+1} = S^{-1}(F(d_i)). \quad (7.1.2)$$

In Abbildung 7.1.1 sind zwei mögliche Abfolgen der Lösung des Modellproblems dargestellt. Ausgehend von einer angenommenen Verschiebung d_0 stellt das Diagramm (a) eine konvergierende Folge dar, das Diagramm (b) eine divergierende. Der Unterschied zwischen beiden Diagrammen liegt in der Steigung der Geraden, also in der Steifigkeit der Fluid- und der Strukturfeder im Modellproblem. Ist bei dem hier verwendeten Ansatz die Strukturfeder steifer als die Fluidfeder, konvergiert die Fixpunkt-Iteration. Ist umgekehrt die Fluidfeder steifer, divergiert das Verfahren. Im Spezialfall von perfekter Symmetrie, wenn beide Modellfedern exakt gleich sind, stagniert das Verfahren. Die Bewegung im Verschiebungs-Kraft-Diagramm erfolgt dann auf einem geschlossenen Pfad und es wird keine Lösung gefunden.

Zwei Dinge sind in Abbildung 7.1.1 bemerkenswert. Erstens läuft das Fixpunkt-Verfahren stets um die gesuchte Lösung herum, die Lösung liegt also in der Mitte zwischen den berechneten Punkten. Zweitens wird der Lösungsfortschritt immer kleiner, je mehr man sich der Lösung nähert. Das Verfahren kommt immer näher an die Lösung heran, doch der Iterationsfortschritt wird immer kleiner und die Lösung wird nie erreicht.

Der große Vorteil der Fixpunkt-Verfahren sind die geringen Anforderungen an die Problemformulierung. Außer dem Auswerten der Interfaceoperatoren in Gleichung (6.1.24) werden keine Informationen über das Problem benötigt. Der Nachteil ist die problemspezifische Konvergenz, wie schon am Modellproblem zu sehen ist. Das einfache Fixpunkt-Verfahren mit dem Verschiebungsinkrement (7.1.1) kann nicht als Lösungsverfahren für partitionierte FSI Probleme verwendet werden, da es für viele Aufgabenstellungen divergiert. Aus diesem Grunde wurde in Kapitel 3.5 eine Relaxation eingeführt. Gleichzeitig kann man mit einer passenden Relaxation die Konvergenzgeschwindigkeit des Verfahrens erheblich beschleunigen.

Die Relaxation bei einem Fixpunkt-Verfahren entspricht dem Bestimmen der Schrittweite im Lösungsalgorithmus für nichtlineare Probleme. Die Schritttrichtung ist durch das Interfaceresiduum $\mathbf{r}_{\Gamma,i}^{n+1}$ vorgegeben. Anders als beim unrelaxierten Fixpunkt-Verfahren (7.1.1) wird jedoch nur ein Anteil des Interfaceresiduums zur Korrektur der Interfaceverschiebungen übernommen

$$\Delta \mathbf{d}_{\Gamma,i}^{n+1} = \omega_i \mathbf{r}_{\Gamma,i}^{n+1} \quad (7.1.3)$$

wobei der Relaxationsparameter ω_i so bestimmt werden soll, daß die aktualisierte Interfaceverschiebung $\mathbf{d}_{\Gamma,i+1}^{n+1}$

$$\mathbf{d}_{\Gamma,i+1}^{n+1} = \mathbf{d}_{\Gamma,i}^{n+1} + \omega_i \Delta \mathbf{d}_{\Gamma,i}^{n+1} \quad (7.1.4)$$

der Lösung im Zeitschritt $\mathbf{d}_{\Gamma}^{n+1}$ möglichst nahe kommt.

Die Notwendigkeit der Relaxation (7.1.3) läßt sich anschaulich aus dem in Abbildung 7.1.1 beobachteten Verhalten motivieren. Die neue Verschiebung d_{i+1} schießt während der Iteration immerzu über den Gleichgewichtspunkt hinaus. Dieses Überschießen ist die Folge der harten Annahme der Dirichlet-Neumann-Partitionierung, bei der die Wirkung des einen Feldes auf das andere immer überschätzt wird.

Die Dirichlet-Partition wird mit einer fest vorgegebenen Verschiebung am Interface gelöst, gegen die beliebige Kräfte wirken können. Im gekoppelten System können sich diese Kräfte nicht aufbauen, da das Interface den Kräften nachgibt. Die Neumann-Partition dagegen berechnet Verschiebungen am Interface, ohne zu wissen, daß im gekoppelten System ein verschiebungsabhängiger Widerstand existiert.

Verschiedene Ansätze zum Bestimmen des Relaxationsparameters ω_i wurden in Kapitel 3.5 besprochen. Diese Ansätze werden im Folgenden im Kontext des partitionierten FSI Problems näher betrachtet.

7.1.1. Relaxation mit festem Parameter. Den Relaxationsparameter festzulegen ist, wie schon in Kapitel 3.5.1 diskutiert, sehr ineffektiv. Für das Lösen von FSI Problemen stellt ein fester Parameter keinen gangbaren Weg dar.

7.1.2. Systematisches Probieren. Suchverfahren mit mehreren Residuumsauswertungen, die in Kapitel 3.5.2 besprochen werden, sind für partitionierte FSI Probleme ebenfalls wenig geeignet, da das Auswerten des Interfaceresiduums (6.3.1) eine sehr aufwendige Operation ist. Das Ziel beim Konstruieren von Fixpunkt-Verfahren für FSI Probleme ist es, die Zahl der nötigen Residuumsauswertungen zu minimieren.

7.1.3. Methode des steilsten Abstiegs. Die Methode des steilsten Abstiegs wird unter dem Namen Gradientenverfahren in Kapitel 3.5.3 im Detail vorgestellt. Das Verfahren wurde in Wall u. a. [163], Mok und Wall [118] als Verfahren des steilsten Abstiegs vorgestellt, siehe dazu auch Mok [117] und die Darstellung in Küttler und Wall [98].

Der Relaxationsparameter ω_i wird, wie in Kapitel 3.5.3 beschrieben, mit Gleichung (3.5.14) bestimmt, wobei das Inkrement $\Delta \mathbf{x}_i$ hier gerade dem Interfaceresiduum $\mathbf{r}_{\Gamma,i}^{n+1}$ entspricht. Mit der Interface-Jacobimatrix \mathbf{J}_Γ

$$\mathbf{J}_\Gamma = \frac{\partial \mathbf{r}_{\Gamma,i}^{n+1}(\mathbf{d}_{\Gamma,i}^{n+1})}{\partial \mathbf{d}_{\Gamma,i}^{n+1}} = \frac{\partial}{\partial \mathbf{d}_{\Gamma,i}^{n+1}} \left(\mathcal{S}_\Gamma^{-1}(\mathcal{F}_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1})) - \mathbf{d}_{\Gamma,i}^{n+1} \right) \quad (7.1.5)$$

lautet Gleichung (3.5.14) demnach

$$\omega_i = - \frac{\left(\mathbf{r}_{\Gamma,i}^{n+1} \right)^T \mathbf{r}_{\Gamma,i}^{n+1}}{\left(\mathbf{r}_{\Gamma,i}^{n+1} \right)^T \mathbf{J}_\Gamma \mathbf{r}_{\Gamma,i}^{n+1}}. \quad (7.1.6)$$

Analog zu Gleichung (3.5.13) wird auch die Interface-Jacobimatrix (7.1.5) als symmetrische zweite Ableitung einer skalaren Bewertungsfunktion aufgefaßt. Für das partitionierte FSI Problem trifft diese Annahme nicht zu, weshalb das Gradientenverfahren zur Berechnung des Relaxationsparameters für praktische Berechnungen nicht eingesetzt wird.

Das Berechnen des Parameters (7.1.6) ist algorithmisch dennoch interessant. Mit einem bekannten Interfaceresiduum $\mathbf{r}_{\Gamma,i}^{n+1}$ erfordert das Auswerten von (7.1.6) das Berechnen des Matrix-Vektor-Produkts $\mathbf{J}_\Gamma \mathbf{r}_{\Gamma,i}^{n+1}$. Nachdem die Interfaceoperatoren \mathcal{S}_Γ^{-1} und \mathcal{F}_Γ aber nicht explizit angegeben werden können, liegt auch die Jacobimatrix \mathbf{J}_Γ nicht explizit vor. Das Matrix-Vektor-Produkt muß deshalb über das Auswerten von Interfaceoperatoren bestimmt werden. Da das Auswerten dieses Matrix-Vektor-Produktes auch für die in Abschnitt 7.2 besprochenen Interface-Newton-Verfahren erforderlich ist, soll es hier im Detail betrachtet werden. Es existieren zwei Möglichkeiten.

7.1.3.1. Approximation von $\mathbf{J}_\Gamma \mathbf{r}_{\Gamma,i}^{n+1}$ mit finiten Differenzen. Eine sehr einfache Möglichkeit, um das Matrix-Vektor-Produkt in Gleichung (7.1.6) zu bestimmen, ist das Verwenden der finiten Differenz

$$\mathbf{J}_\Gamma \mathbf{y} \approx \frac{\mathcal{S}_\Gamma^{-1}(\mathcal{F}_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1} + \delta \mathbf{y})) - \mathbf{d}_{\Gamma,i}^{n+1} - \delta \mathbf{y} - \mathbf{r}_{\Gamma,i}^{n+1}}{\delta} \quad (7.1.7)$$

mit dem Parameter $\delta = \lambda \left(\lambda + \left| \mathbf{d}_{\Gamma,i}^{n+1} \right| / \left| \mathbf{r}_{\Gamma,i}^{n+1} \right| \right)$ und einem hinreichend kleinen λ . An dieser Stelle ist der einfache Ansatz (7.1.7) völlig ausreichend, der das bereits bekannte Residuum $\mathbf{r}_{\Gamma,i}^{n+1}$ verwendet und lediglich eine zusätzliche Auswertung der Interfaceoperatoren, also der nichtlinearen Feldlöser, verlangt. Der numerische Aufwand bleibt dennoch hoch und die Approximation des Matrix-Vektor-Produktes mit Hilfe von finiten Differenzen ist numerisch sensibel.

Der numerische Aufwand der Auswertung von (7.1.7) kann spürbar verringert werden, indem die Genauigkeit beim Lösen der nichtlinearen Felder verringert wird. Oftmals genügt es, bei der Berechnung des Residuums $\mathcal{S}_\Gamma^{-1}(\mathcal{F}_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1} + \delta \mathbf{y})) - \mathbf{d}_{\Gamma,i}^{n+1} - \delta \mathbf{y}$ in (7.1.7) das Newton-Verfahren innerhalb der Interfaceoperatoren nach nur einem linearen Schritt zu beenden.

7.1.3.2. Berechnen von $\mathbf{J}_\Gamma \mathbf{r}_{\Gamma,i}^{n+1}$ über Ableitungsoperatoren. Als Alternative zur finiten Differenz (7.1.7) kann die Ableitung in der Definition der Jacobi-Matrix (3.5.13) implizit berechnet werden

$$\mathbf{J}_\Gamma \mathbf{y} = \left(\mathcal{S}_\Gamma^{-1} \right)' \left(\mathcal{F}_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1}) \right) \mathcal{F}'_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1}) \mathbf{y} - \mathbf{y}. \quad (7.1.8)$$

Hier ist die Ableitung des inversen Strukturoperators $\left(\mathcal{S}_\Gamma^{-1} \right)'$ an der Stelle $\mathcal{F}_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1})$ und die Ableitung des Fluidoperators \mathcal{F}'_Γ an der Stelle $\mathbf{d}_{\Gamma,i}^{n+1}$, jeweils angewendet auf einen gegebenen Vektor, auszuwerten. Nachdem der Strukturoperator \mathcal{S}_Γ^{-1} auf dem Lösen eines

linearen Problems im Rahmen des Newton-Verfahren beruht, ist die Sensitivität des Operator \mathcal{S}_Γ^{-1} durch das Auswerten eben dieses linearen Problems an der Stelle $\mathcal{F}_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1})$ mit dem Neumannrand

$$\mathbf{z} = \mathcal{F}'_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1}) \mathbf{y} \quad (7.1.9)$$

gegeben. Das Auswerten von (7.1.9) stellt dagegen eine Schwierigkeit dar. Die Ableitung des Fluidoperators nach den Interfaceverschiebungen ist wegen der Bedingung (6.1.1) durch

$$\mathcal{F}'_\Gamma = \frac{d\mathcal{F}_\Gamma}{d\mathbf{d}_\Gamma^{n+1}} = \frac{\partial \mathcal{F}_\Gamma}{\partial \mathbf{d}_\Gamma^{n+1}} + \frac{\partial \mathcal{F}_\Gamma}{\partial \mathbf{u}_\Gamma^{n+1}} \frac{\partial \mathbf{u}_\Gamma^{n+1}}{\partial \mathbf{d}_\Gamma^{n+1}} = \frac{\partial \mathcal{F}_\Gamma}{\partial \mathbf{d}_\Gamma^{n+1}} + \frac{\partial \mathcal{F}_\Gamma}{\partial \mathbf{u}_\Gamma^{n+1}} \frac{2}{\Delta t} \quad (7.1.10)$$

gegeben. Die Ableitung nach der Interfacegeschwindigkeit $\partial \mathcal{F}_\Gamma / \partial \mathbf{u}_\Gamma^{n+1}$ ist ganz analog zum inversen Strukturoperator durch die Auswertung des linearen Problems an der Stelle $\mathbf{d}_{\Gamma,i}^{n+1}$ verfügbar. Die Ableitung nach der Interfaceverschiebung $\partial \mathcal{F}_\Gamma / \partial \mathbf{d}_\Gamma^{n+1}$ dagegen führt entsprechend der Linearisierung (6.1.9) auf die Sensitivitätsmatrix \mathbf{F}^G , die mit den Netzverschiebungen $\mathbf{d}^{G,n+1}(\mathbf{y})$ multipliziert als Term auf der rechten Seite des linearen Systems auftritt. In Fernández und Moubachir [48] ist das Aufstellen des vollständigen Problems gezeigt. Oftmals steht die Sensitivitätsmatrix aber nicht zur Verfügung. In diesen Fällen wird der Beitrag der Netzbewegung gern weggelassen und die Ableitung des Fluidoperators vereinfacht sich zu

$$\mathcal{F}'_\Gamma \approx \frac{\partial \mathcal{F}_\Gamma}{\partial \mathbf{u}_\Gamma^{n+1}} \frac{2}{\Delta t}. \quad (7.1.11)$$

Das Auswerten der Beziehung (7.1.9) erfordert damit das Lösen des linearen Systems aus dem Newton-Verfahren des Fluidoperators \mathcal{F}_Γ an der Stelle $\mathbf{d}_{\Gamma,i}^{n+1}$ mit der vorgeschriebenen Interfacegeschwindigkeit $2\mathbf{y}/\Delta t$. Das Auswerten des Matrix-Vektor-Produktes (7.1.8) benötigt damit zwei lineare Lösungen, je eine für das Fluid- und eine für das Strukturfeld.

Die gleiche Erkenntnis läßt sich auch aus dem Limit der finiten Differenz (7.1.7) gewinnen, indem die Feldoperatoren als lokal linear angenommen werden.

$$\begin{aligned} \mathbf{J}_\Gamma \mathbf{y} &= \lim_{\delta \rightarrow 0} \frac{\mathcal{S}_\Gamma^{-1}(\mathcal{F}_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1} + \delta \mathbf{y})) - \mathbf{d}_{\Gamma,i}^{n+1} - \delta \mathbf{y} - \mathbf{r}_{\Gamma,i}^{n+1}}{\delta} \\ &\approx \lim_{\delta \rightarrow 0} \frac{\mathbf{r}_{\Gamma,i}^{n+1} + \mathcal{S}_{\Gamma,\text{lin}}^{-1}(\mathcal{F}_{\Gamma,\text{lin}}(\delta \mathbf{y})) - \delta \mathbf{y} - \mathbf{r}_{\Gamma,i}^{n+1}}{\delta} \\ &\approx \mathcal{S}_{\Gamma,\text{lin}}^{-1}(\mathcal{F}_{\Gamma,\text{lin}}(\mathbf{y})) - \mathbf{y} \end{aligned} \quad (7.1.12)$$

In der Differenz verschwinden alle Anteile außer der Interfaceverschiebung \mathbf{y} . Im Besonderen sind in den linearen Operatoren $\mathcal{S}_{\Gamma,\text{lin}}^{-1}$ und $\mathcal{F}_{\Gamma,\text{lin}}$ weder Anteile aus dem alten Zeitschritt noch von Null verschiedene Randbedingungen enthalten. Die linearen Operatoren in (7.1.12) entsprechen genau den Ableitungen in (7.1.8). Die Annahme eines linearen Fluidoperators $\mathcal{F}_{\Gamma,\text{lin}}(\mathbf{y})$ impliziert das Ignorieren des Beitrags der Netzbewegung (7.1.11).

7.1.4. Aitkens Δ^2 Extrapolation. Aitkens Δ^2 Verfahren wurde in Kapitel 3.5.4 vorgestellt. Die Idee dieses Verfahrens ist es, den Relaxationsparameter aus einer linearen Extrapolation gewinnen. Das originale Aitken Verfahren dient der Konvergenzbeschleunigung von konvergierenden skalaren Folgen.

Das lineare Modellproblem liefert eine Folge von skalaren Verschiebungswerten. Im konvergierenden Fall genügt es dabei, drei aufeinander folgende Verschiebungswerte d_{i-1} , d_i , d_{i+1} zu kennen, um die exakte Lösung angeben zu können. Wegen der Linearität der Operatoren bilden die Verschiebungsdifferenzen

$$\Delta d_i = d_{i+1} - d_i \quad (7.1.13)$$

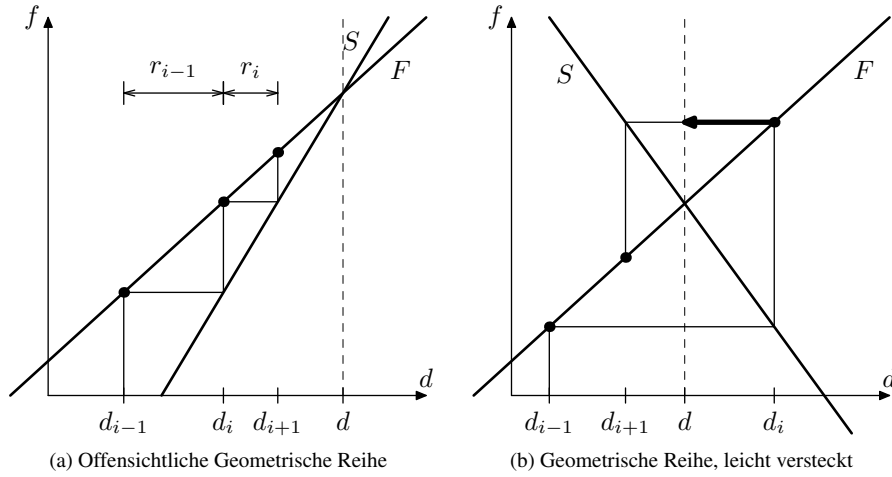


Abbildung 7.1.2: Drei aufeinander folgende Verschiebungswerte genügen für die geometrische Reihe. Die Darstellung (a) gibt die Reihe leicht zu erkennen, das zum Modellproblem passende Diagramm (b) versteckt den Zusammenhang ein wenig.

eine geometrische Reihe wie in Abbildung 7.1.2 dargestellt. Somit ist

$$\frac{\Delta d_i}{\Delta d_{i-1}} = q = \text{const} \quad (7.1.14)$$

und die Verschiebungslösung ist durch die explizite Summe der Reihe gegeben

$$d = d_0 + \frac{\Delta d_0}{1 - q}. \quad (7.1.15)$$

Allgemeiner hat man von einer beliebigen Verschiebung d_i ausgehend den Zusammenhang

$$d = d_i + \frac{\Delta d_i}{1 - q} = d_i + \Delta d_i \frac{\Delta d_{i-1}}{\Delta d_{i-1} - \Delta d_i} = d_i + \omega_i \Delta d_i \quad (7.1.16)$$

mit dem Relaxationsparameter

$$\omega_i = \frac{\Delta d_{i-1}}{\Delta d_{i-1} - \Delta d_i} = \frac{d_i - d_{i-1}}{d_i - d_{i-1} - d_{i+1} + d_i}. \quad (7.1.17)$$

Die Gleichung (7.1.16) ist die skalare Aitken Formel. Dabei geht in (7.1.17) mit Δd_{i-1} auch die Verschiebung d_{i-1} ein, so daß drei Verschiebungen bekannt sein müssen, bevor die Relaxation ausgeführt werden kann. D.h. es kann nur jeden zweiten Schritt relaxiert werden. Beim Modellproblem ist damit natürlich schon die Lösung erreicht.

Beim Lösen von nichtlinearen Problemen möchte man in jedem Lösungsschritt relaxieren. In Kapitel 3.5.4 wird deshalb die Vektorversion von Aitkens Δ^2 Verfahren vorgestellt, die auf Irons und Tuck [83] zurückgeht. Diese Version wurde von Wall u. a. [163], Mok und Wall [118] und auch Mok [117] für FSI Probleme eingesetzt. Eine neuere Darstellung des Verfahrens ist in Küttler und Wall [98] enthalten.

Das Relaxationsverfahren nach Irons und Tuck [83] ist die Erweiterung des Sekantenverfahrens auf Vektorfolgen. Bei diesem Verfahren bilden die Verschiebungsdifferenzen Δd_i des Modellproblems keine geometrische Reihe, siehe Abbildung 7.1.3. Stattdessen geht das Verfahren von Paaren von Verschiebungen aus

$$\tilde{d}_i = S^{-1}(F(d_i)), \quad (7.1.18)$$

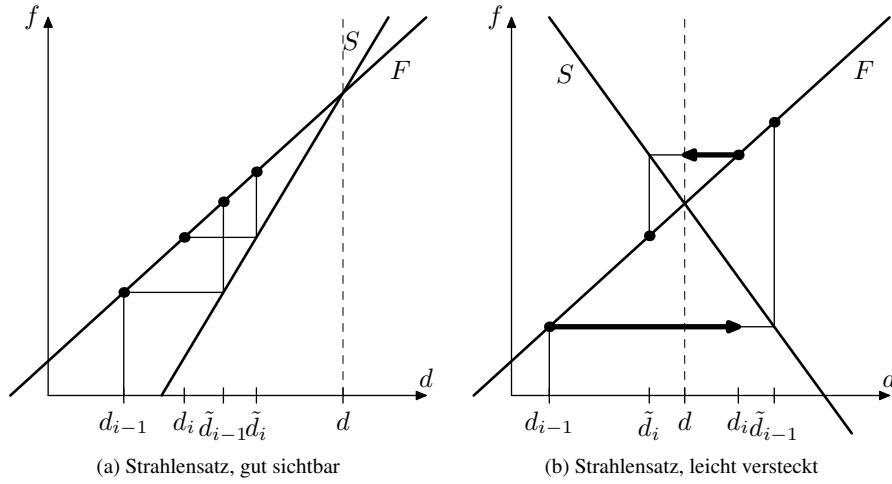


Abbildung 7.1.3: Aitens Δ^2 Relaxation nach Irons und Tuck [83] am Modellproblem illustriert, analog zu Abbildung 7.1.2. Der Strahlensatz ist das ganze Geheimnis.

zwischen denen im linearen Modell der Strahlensatz gilt

$$\frac{d - d_{i-1}}{d - \tilde{d}_{i-1}} = \frac{d - d_i}{d - \tilde{d}_i}, \quad (7.1.19)$$

der jedes Verschiebungspaar d_i, \tilde{d}_i auf die gesuchte Lösung d bezieht. Die Lösung d erhält man mit

$$d = \frac{d_{i-1}\tilde{d}_i - d_i\tilde{d}_{i-1}}{\Delta d_i - \Delta d_{i-1}}. \quad (7.1.20)$$

Dieser Ansatz läßt sich ganz analog auch im nichtlinearen skalaren Fall einsetzen. Die Gleichung (7.1.20) ist dann nicht mehr exakt, sondern liefert eine Verbesserung der Lösung mit Hilfe der Sekantenmethode

$$d_{i+1} = \frac{d_{i-1}\tilde{d}_i - d_i\tilde{d}_{i-1}}{\Delta d_i - \Delta d_{i-1}}. \quad (7.1.21)$$

Aus dem Interpolationsansatz

$$d_{i+1} = d_i + \omega_i(\tilde{d}_i - d_i) = \frac{d_{i-1}\tilde{d}_i - d_i\tilde{d}_{i-1}}{\Delta d_i - \Delta d_{i-1}} \quad (7.1.22)$$

erhält man somit den Relaxationsparameter

$$\omega_i = \frac{d_{i-1} - d_i}{\Delta d_i - \Delta d_{i-1}} = \omega_{i-1} \frac{\Delta d_{i-1}}{\Delta d_{i-1} - \Delta d_i}. \quad (7.1.23)$$

Dieser Parameter hat große Ähnlichkeit zu dem Parameter (7.1.17), ist aber in jedem Schritt anwendbar.

Im Vektorfall erhält man den Relaxationsparameter ω_i aus der Gleichung (3.5.29), die sich für die partitionierten FSI Probleme als

$$\omega_i = -\omega_{i-1} \frac{\left(\mathbf{r}_{\Gamma,i-1}^{n+1}\right)^T \left(\mathbf{r}_{\Gamma,i}^{n+1} - \mathbf{r}_{\Gamma,i-1}^{n+1}\right)}{\left|\mathbf{r}_{\Gamma,i}^{n+1} - \mathbf{r}_{\Gamma,i-1}^{n+1}\right|^2} \quad (7.1.24)$$

darstellen läßt.

Algorithmus 18 Fixpunkt-Verfahren mit Aitkens Δ^2 Relaxation.

für alle Zeitschritte:

$$\mathbf{d}_{\Gamma,0}^{n+1} = \mathbf{d}_{\Gamma}^n$$

für alle $i = 0, \dots, \max_{\text{iter}}$:

$$\tilde{\mathbf{d}}_{\Gamma,i}^{n+1} = \mathcal{S}_{\Gamma}^{-1}(\mathcal{F}_{\Gamma}(\mathbf{d}_{\Gamma,i}^{n+1}))$$

$$\mathbf{r}_{\Gamma,i}^{n+1} = \tilde{\mathbf{d}}_{\Gamma,i}^{n+1} - \mathbf{d}_{\Gamma,i}^{n+1}$$

falls $|\mathbf{r}_{\Gamma,i}^{n+1}| \leq \varepsilon$:

$$\mathbf{d}_{\Gamma}^{n+1} = \tilde{\mathbf{d}}_{\Gamma,i}^{n+1}$$

nächster Zeitschritt

$$\omega_i = -\omega_{i-1} \frac{(\mathbf{r}_{\Gamma,i-1}^{n+1})^T (\mathbf{r}_{\Gamma,i}^{n+1} - \mathbf{r}_{\Gamma,i-1}^{n+1})}{|\mathbf{r}_{\Gamma,i}^{n+1} - \mathbf{r}_{\Gamma,i-1}^{n+1}|^2}$$

$$\mathbf{d}_{\Gamma,i+1}^{n+1} = \mathbf{d}_{\Gamma,i}^{n+1} + \omega_i \mathbf{r}_{\Gamma,i}^{n+1}$$

$$t = t + \Delta t$$

Der besondere Reiz der Relaxation (7.1.24) ist die einfache Implementierung. Die Interface-residuen $\mathbf{r}_{\Gamma,i}^{n+1}$ sind bekannt, darüber hinaus werden lediglich Vektoroperationen benötigt. Der numerische Aufwand bei der Auswertung von Gleichung (7.1.24) ist deshalb sehr gering.

Die von Irons und Tuck [83] vorgeschlagene Formulierung des relaxierten Iterationsverfahrens kommt im Vergleich mit dem unrelaxierten Verfahren mit einem einzigen zusätzlichen Vektor aus. Bei den inzwischen verfügbaren Speicherkapazitäten stellt die Speicheranforderung jedoch nicht mehr das Hauptaugenmerk dar, so daß hier einer schlichteren Darstellung der Vorzug gegeben wurde.

Um den Relaxationsparameter (7.1.24) zu berechnen, sind zwei Verschiebungspaare erforderlich. Nach der ersten Iteration, wenn lediglich der Ausgangszustand und das erste Zwischenergebnis vorliegt, kann deshalb noch nicht relaxiert werden. Der erste Relaxationsparameter in einem Zeitschritt ω_1^{n+1} muß deshalb anders abgeschätzt werden. Irons und Tuck [83] empfehlen, den jeweils letzten Parameter des vorhergehenden Zeitschritts ω^n zu verwenden. Für praktische FSI Berechnungen kann es aber nötig sein, diesen Wert mit einer problemspezifischen Konstanten ω_{\max} zu begrenzen

$$\omega_1^{n+1} = \max(\omega^n, \omega_{\max}), \quad (7.1.25)$$

da ein zu großer Relaxationsparameter zum Divergieren der Rechnung führen kann.

Dem Fixpunkt-Verfahren mit Aitken-Relaxation nach Irons und Tuck [83] kommt für praktische FSI Berechnungen eine große Bedeutung zu, da es sich um einen sehr einfachen und effizienten Algorithmus handelt. Deshalb ist in Algorithmus 18 der vollständige Algorithmus angegeben.

7.2. Newton-Verfahren

Anstatt im Sinne der Fixpunkt-Verfahren die Information über das Verhalten der Interfaceoperatoren für das Berechnen der Schrittweite (7.1.3) zu verwenden, kann aus dem Verhalten der Interfaceoperatoren auch direkt die Schrittrichtung $\Delta \mathbf{d}_{\Gamma,i}^{n+1}$ bestimmt werden. Das ist der Ansatz der Newton-Verfahren. Newton-Verfahren als Lösungsmethoden für allgemeine nichtlineare Probleme wurden in Kapitel 3.1 vorgestellt.

7.2.1. Newton-Verfahren mit dem Jacobi-System. Im Modellproblem entspricht die Jacobi-Matrix gerade der Steigung der Geraden. Wenn man diese Steigungen in die Gleichung einbezieht, lassen sich Gleichgewichtsbeziehungen am Lösungspunkt zum Bestimmen der Lösung verwenden. Abbildung 7.2.1 (a) zeigt diesen Zusammenhang. Ausgehend von einer bekannten Verschiebung d_i bekommt man von jedem Operator eine Kraftantwort

$$f_i^S = S(d_i) \quad \text{und} \quad f_i^F = F(d_i). \quad (7.2.1)$$

Im Schnittpunkt der Geraden sind die Kräfte gerade im Gleichgewicht. Der Schnittpunkt ist um Δd von der Ausgangsposition entfernt

$$f_i^F + J^F \Delta d = f_i^S + J^S \Delta d. \quad (7.2.2)$$

Das zu lösende System zum Berechnen der Verschiebungsdifferenz Δd ist damit

$$(J^F - J^S) \Delta d = f_i^S - f_i^F. \quad (7.2.3)$$

Das reale FSI System lautet für diesen Ansatz

$$\left(\frac{d\mathcal{F}_\Gamma}{d\mathbf{d}_\Gamma^{n+1}} - \frac{d\mathcal{S}_\Gamma}{d\mathbf{d}_\Gamma^{n+1}} \right) \Delta \mathbf{d}_{\Gamma,i}^{n+1} = \mathcal{S}_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1}) - \mathcal{F}_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1}). \quad (7.2.4)$$

Das ist ein lineares Gleichungssystem mit dem unbekanntem Inkrement $\Delta \mathbf{d}_{\Gamma,i}^{n+1}$. Dieses Gleichungssystem ist natürlich mehrfach zu lösen, da die Interfaceoperatoren \mathcal{S}_Γ und \mathcal{F}_Γ nichtlineare Operatoren sind. Entsprechend ist Gleichung (7.2.4) Teil eines iterativen Verfahrens. Die zugehörige Aktualisierung der Interfaceverschiebung lautet

$$\mathbf{d}_{\Gamma,i+1}^{n+1} = \mathbf{d}_{\Gamma,i}^{n+1} + \Delta \mathbf{d}_{\Gamma,i}^{n+1}. \quad (7.2.5)$$

Das hier skizzierte Verfahren ist das Newton-Verfahren angewendet auf die in Kapitel 6.1.7 diskutierte Jacobi-Zerlegung des FSI Problems. Insbesondere baut dieses Verfahren nicht auf die Dirichlet-Neumann-Partitionierung auf. Dieses Verfahren wird nicht für Berechnungen verwendet. Die Diskussion an dieser Stelle soll den Zusammenhang der Newton-Verfahren mit den Fixpunkt-Verfahren zeigen.

Das lineare System (7.2.4) kann wegen der impliziten Definition der Interfaceoperatoren nicht explizit aufgestellt werden. In Analogie zu den in den Abschnitten 7.1.3.1 und 7.1.3.2 diskutierten Matrix-Vektor-Produkten können auch die Produkte

$$\frac{d\mathcal{F}_\Gamma}{d\mathbf{d}_\Gamma^{n+1}} \Delta \mathbf{d}_{\Gamma,i}^{n+1} \quad \text{und} \quad \frac{d\mathcal{S}_\Gamma}{d\mathbf{d}_\Gamma^{n+1}} \Delta \mathbf{d}_{\Gamma,i}^{n+1} \quad (7.2.6)$$

berechnet werden. Zum Lösen des Systems (7.2.4) kommen deshalb Jacobian-freie Krylov-Verfahren in Frage. In Kombination mit dem Newton-Verfahren entsteht so ein Newton-Krylov-Verfahren, wie in Kapitel 3.4 diskutiert. Doch um das System (7.2.4) mit Hilfe eines Krylov-Verfahrens zu lösen, würde ein passender Vorkonditionierer benötigt. Die in Kapitel 3.3 vorgestellten Vorkonditionierungstechniken können hier nicht angewendet werden, da die Matrix des linearen Gleichungssystems (7.2.4) nicht explizit berechnet werden kann. Es steht somit kein geeigneter Vorkonditionierer zur Verfügung. Für die Konstruktion von künstlichen Vorkonditionierern auf der Basis von Modellproblemen gelten sinngemäß die Anmerkungen aus Kapitel 6.1.7. Die auf dem Jacobi-System beruhenden Newton-Verfahren werden nicht für die Berechnungen von FSI Problemen verwendet.

7.2.2. Newton-Verfahren mit dem Gauß-Seidel-System. Abbildung 7.2.1 (b) zeigt eine alternative Herangehensweise am Modellproblem. Hier ist die Frage, wie man beim Übergang von d_i zu d_{i+1} nur den Weg entlang der Fluidgeraden gehen und den Abstieg über die Strukturgerade vermeiden kann. Es gilt also, das Verschiebungsinkrement auf Fluidseite Δd^F zu berechnen, wobei das Fluidinkrement ein Teil des Gesamtinkrements ist

$$r_i = d_{i+1} - d_i = S^{-1}(F(d_i)) - d_i = \Delta d_i = \Delta d^S + \Delta d^F. \quad (7.2.7)$$

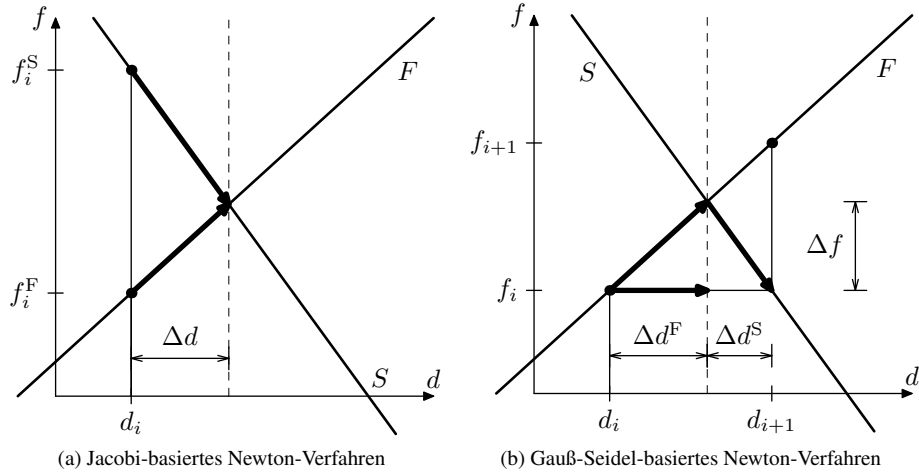


Abbildung 7.2.1: Ein linearer Schritt aus dem Newton-Verfahren.

Da es sich um ein lineares Problem handelt, gilt zwischen den Inkrementen die Beziehung

$$\Delta f^F = J^F \Delta d^F \quad \text{und} \quad \Delta f^S = J^S \Delta d^S. \quad (7.2.8)$$

Aus der Bedingung $\Delta f^F = -\Delta f^S$, der Aufstieg über die Fluidgerade gleich dem Abstieg über die Strukturgerade, erhält man die Gleichung

$$\Delta d^S = -(J^S)^{-1} J^F \Delta d^F. \quad (7.2.9)$$

An dieser Stelle sind beide Inkrement Δd^S und Δd^F unbekannt, doch durch die Addition von Δd^F erhält man die Gleichung

$$\Delta d_i = \left(I - (J^S)^{-1} J^F \right) \Delta d^F \quad (7.2.10)$$

mit einer Unbekannten. Das Verschiebungsincrement Δd^F zu finden, bedeutet also, ein lineares Gleichungssystem mit der Koeffizientenmatrix $I - (J^S)^{-1} J^F$ zu lösen. Diese Matrix ist mit der Ableitung

$$\frac{\partial r_i}{\partial d} = (S^{-1})' (F(d_i)) \cdot F'(d_i) - I = (J^S)^{-1} J^F - I = J^{\text{FSI}} \quad (7.2.11)$$

gegeben. Eingesetzt erhält man das lineare System

$$J^{\text{FSI}} \Delta d^F = -\Delta d^S - \Delta d^F = -r. \quad (7.2.12)$$

Das Ergebnis ist also ein Newton-Verfahren basierend auf der Gauß-Seidel-Formulierung, wie sie aus der Dirichlet-Neumann-Partitionierung entspringt.

Die Lösung Δd^F ist ein beliebiger Vektor und liegt keineswegs nur auf der Geraden zwischen d_i und d_{i+1} . Darin liegt der Vorzug des Newton-Verfahrens. Es werden im Vergleich zum Fixpunkt-Verfahren erheblich weniger Iterationen benötigt, da die einzelnen Iterationsschritte unter Berücksichtigung des Verhaltens der Funktion an der aktuellen Position gewählt werden. Im linearen Modellproblem ist dieser Vorteil nicht zu zeigen, da hier die Lösung direkt angegeben werden kann.

Die Jacobi-Matrix am Interface als Ableitung des Interfaceresiduums (6.3.1) wurde schon in Gleichung (7.1.5) angegeben. Das lineare System im Newton-Verfahren für das Dirichlet-Neumann partitionierte FSI Problem lautet damit

$$\mathbf{J}_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1}) \Delta \mathbf{d}_{\Gamma,i}^{n+1} = -\mathbf{r}_{\Gamma,i}^{n+1}, \quad (7.2.13)$$

komplettiert wird das Verfahren von der Inkrementierung der Interfaceverschiebung (3.1.5).

Die Jacobi-Matrix \mathbf{J}_Γ in (7.2.13) ist dicht besetzt und unsymmetrisch. Allerdings gelingt es mit akzeptablem Aufwand nicht, diese Matrix aufzustellen, da die Interfaceoperatoren nicht explizit angegeben, sondern lediglich ausgewertet werden können. Aus diesem Grund muß auch das lineare System (7.2.13), analog zum System (7.2.4), mit einem matrizenfreien Verfahren gelöst werden, das ausschließlich auf der Matrix-Vektor-Multiplikation beruht. Das Newton-Verfahren zum Lösen des Interfaceproblems der Dirichlet-Neumann-Partitionierung wird so zum Jacobi-freien Newton-Krylov-Verfahren. Diese Verfahren wurden in Kapitel 3.4 eingeführt. Eine umfassende Zusammenstellung dieser Verfahren kann in Knoll und Keyes [92] gefunden werden.

Das Krylov-Verfahren, mit dem Gleichung (7.2.13) gelöst werden soll, wirft die Frage nach dem Vorkonditionierer auf. Wie von Kerkhoven und Saad [90] für elliptische nichtlineare Probleme gezeigt wird, ist für das Lösen eines Interfaceproblems aus einer Dirichlet-Neumann-Partitionierung keine Vorkonditionierung erforderlich. Anschaulich betrachtet wirkt im Jacobi-Operator in Gleichung (7.2.13) der inverse Strukturoperator \mathcal{S}_Γ^{-1} als Vorkonditionierer für den Fluidoperator \mathcal{F}_Γ . Der Jacobi-Operator vermittelt zwischen einem Verschiebungsfehler und einem Verschiebungsinkrement in der gleichen Größenordnung. Damit ist das Krylov-Verfahren in der Lage, eine Lösung zu finden.

Diese Argumentation entspricht der Motivation der Dirichlet-Neumann-Partitionierung in Kapitel 6.1.5, bei der der inverse Strukturoperator \mathcal{S}_Γ^{-1} als Vorkonditionierer in die nichtlineare Gleichgewichtsbedingung am Interface eingebracht wird.

Das Krylov-Verfahren zum Lösen des linearen Systems (7.2.13) basiert auf der Matrix-Vektor-Multiplikation

$$\mathbf{J}_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1})\mathbf{y}. \quad (7.2.14)$$

Zur Auswertung dieses Produkts steht die Approximation mit Hilfe einer finiten Differenz (7.1.7) oder das Auswerten der Ableitungen der Interfaceoperatoren (7.1.8) zur Verfügung.

In Fernández und Moubachir [48] werden die matrizenfreien Newton-Krylov-Verfahren für das Dirichlet-Neumann partitionierte FSI Problem mit exakter Jacobi-Matrix vorgestellt. Die exakte Jacobi-Matrix schließt auch die Sensitivitätsterme (6.1.10) mit ein, die die Abhängigkeit der Navier-Stokes-Gleichung von der Netzbewegung des Fluidnetzes beschreiben. Oftmals stehen diese Terme jedoch nicht zur Verfügung.

Da die Jacobi-Matrix im Rahmen von Quasi-Newton-Verfahren eine Näherung sein kann, können anstelle der exakten Ableitungen in (7.1.8) Näherungen verwendet werden. Als einfache Näherung wird beim Auswerten der Ableitung des Fluidoperators \mathcal{F}_Γ^l in (7.1.8) gern auf die Sensitivitätsterme (6.1.10) verzichtet, wie schon in Gleichung (7.1.11) angedeutet. Die Ableitung des Fluidoperators \mathcal{F}_Γ^l kann jedoch auch grob angenähert werden, beispielsweise durch ein lineares Poisson-Problem, das lediglich das Verhalten des Drucks nachbildet. Diese Quasi-Newton-Verfahren für FSI Probleme werden in Gerbeau und Vidrascu [60] sowie Gerbeau u. a. [61] beschrieben. Eine ausführliche Darstellung der Newton-Verfahren für FSI Probleme findet sich auch in Deparis [29].

Algorithmus 19 zeigt das Newton-Verfahren als Lösungsalgorithmus für das Dirichlet-Neumann partitionierte FSI Problem. Auf das Krylov-Verfahren zur Lösung des linearen Systems (7.2.13) und auf die Auswertung des Matrix-Vektor-Produkts (7.2.14) wird in Algorithmus 19 nicht eingegangen. Da die Jacobi-Matrix in (7.2.14) unsymmetrisch ist, wird in aller Regel das in Kapitel 3.2.3 angedeutete GMRES-Verfahren als lineares Lösungsverfahren eingesetzt.

Das Newton-Krylov-Verfahren ist ein robustes und effizientes Lösungsverfahren für das aus der Dirichlet-Neumann-Partitionierung stammende nichtlineare Interfaceproblem. Das

Algorithmus 19 Jacobi-freies Newton-Krylov-Verfahren zum Lösen des FSI Problems.

für alle Zeitschritte:

$$\mathbf{d}_{\Gamma,0}^{n+1} = \mathbf{d}_{\Gamma}^n$$

für alle $i = 0, \dots, \max_{\text{iter}}$:

$$\tilde{\mathbf{d}}_{\Gamma,i}^{n+1} = \mathcal{S}_{\Gamma}^{-1}(\mathcal{F}_{\Gamma}(\mathbf{d}_{\Gamma,i}^{n+1}))$$

$$\mathbf{r}_{\Gamma,i}^{n+1} = \tilde{\mathbf{d}}_{\Gamma,i}^{n+1} - \mathbf{d}_{\Gamma,i}^{n+1}$$

falls $|\mathbf{r}_{\Gamma,i}^{n+1}| \leq \varepsilon$:

$$\mathbf{d}_{\Gamma}^{n+1} = \tilde{\mathbf{d}}_{\Gamma,i}^{n+1}$$

nächster Zeitschritt

$$\mathbf{J}_{\Gamma}(\mathbf{d}_{\Gamma,i}^{n+1}) \Delta \mathbf{d}_{\Gamma,i}^{n+1} = -\mathbf{r}_{\Gamma,i}^{n+1}$$

$$\mathbf{d}_{\Gamma,i+1}^{n+1} = \mathbf{d}_{\Gamma,i}^{n+1} + \Delta \mathbf{d}_{\Gamma,i}^{n+1}$$

$$t = t + \Delta t$$

Berechnen des Matrix-Vektor-Produktes (7.2.14) im Krylov-Verfahren ist jedoch mit erheblichem numerischen Aufwand verbunden, da für jede Multiplikation ein lineares Fluid- und ein lineares Strukturproblem gelöst werden müssen, so daß Beschleunigungen des Lösungsverfahrens wünschenswert sind.

Dazu kombinieren Deparis u. a. [31] das matrizenfreie Newton-Krylov-Verfahren mit der in Kapitel 3.6 vorgestellten Vektorextrapolation. In jedem Newton-Schritt wird zunächst ein neuer Vorschlag für die Interfaceverschiebung $\mathbf{d}_{\Gamma,i}^{n+1}$ aus den bereits berechneten Verschiebungen extrapoliert und anschließend das lineare System (7.2.13) aufgestellt und gelöst.

Gleichzeitig werden von Deparis u. a. [31] die Krylov-Vektoren des GMRES-Verfahrens aufgehoben und beim Lösen des nächsten linearen Systems (7.2.13) zum Vorkonditionieren eingesetzt. Das Ziel dabei ist es, die beim Lösen gewonnene Information über das System optimal zu nutzen und dadurch die Anzahl der benötigten Iterationen zu minimieren. Der auf diese Art zusammengestellte Algorithmus ist jedoch vergleichsweise komplex und hat deshalb kaum Verbreitung gefunden.

Die Idee beim Lösen mehrerer ähnlicher linearer Systeme, die Krylov-Vektoren wieder zu verwenden, wurde für allgemeine Systeme auch von Parks u. a. [121] verfolgt. Eine solche Wiederverwendung kann mit den in Kapitel 3.3 erwähnten flexiblen Krylov-Verfahren erzielt werden, die variable Vorkonditionierer zulassen. Das Wiederverwenden von Krylov-Vektoren ist sicherlich vielversprechend, wurde im Rahmen dieser Arbeit jedoch nicht untersucht.

7.3. Vektorextrapolation

Die Vektorextrapolation als Verfahren zur Konvergenzbeschleunigung von Vektorfolgen wurde in Kapitel 3.6 vorgestellt. Im Kontext von Dirichlet-Neumann partitionierten FSI Problemen kann die Vektorextrapolation verwendet werden, um das nichtlineare Interfaceproblem (6.1.24) zu lösen. Die Anwendung der Vektorextrapolation als Lösungsverfahren für das Interfaceproblem wurde in Küttler und Wall [99] diskutiert.

Für praktische Berechnungen im FSI Kontext wird die Vektorextrapolation kaum verwendet. Der Grund ist die Komplexität des Algorithmus und die hohe Anzahl an Residuumsauswertungen, die für einen Extrapolationsschritt benötigt werden. Da bei FSI Problemen oftmals einige wenige Extrapolationsschritte ausreichen, ist die Vektorextrapolation oft ein

effizientes Verfahren, jedoch selten effizienter als Fixpunkt-Verfahren mit Aitkens Δ^2 Relaxation (7.1.24).

Als Bindeglied zwischen den in Abschnitt 7.1 diskutierten Fixpunkt-Verfahren und den Newton-Krylov-Verfahren aus Abschnitt 7.2 ist die Vektorextrapolation jedoch von Interesse. Wie bereits in Kapitel 3.6 besprochen, kann die Vektorextrapolation als Verallgemeinerung der Fixpunkt-Verfahren verstanden werden, indem beim Bestimmen des Relaxationsparameters mehr bereits bekannte Vektoren beachtet werden. Das Bestimmen des Relaxationsparameters erfordert damit das Lösen eines linearen Minimierungsproblems.

Die Eigenschaft des Fixpunkt-Verfahrens, lediglich das Interfaceresiduum (6.3.1) mit einer beliebigen Interfaceverschiebung $\mathbf{d}_{\Gamma,i}^{n+1}$ berechnen zu müssen, bleibt auch bei der Vektorextrapolation erhalten. Daneben ist der Lösungsalgorithmus der Vektorextrapolation jedoch um einiges komplexer als das Fixpunkt-Verfahren. Die effiziente Implementierung der Vektorextrapolation von Sidi [138], die auch in dieser Arbeit verwendet wird, basiert auf der Zerlegung der Vektorfolge in Krylov-Vektoren und hat beim Lösen von linearen Problemen große Ähnlichkeiten mit dem GMRES-Verfahren von Saad und Schultz [132].

Der Lösungsalgorithmus von Sidi [138], der mehrere Extrapolationsschritte zum Lösen eines nichtlinearen Problems verwendet, zeigt deshalb auch eine Nähe zum Newton-Krylov-Verfahren aus Kapitel 3.4 mit der Approximation des Matrix-Vektor-Produktes (7.2.14) mittels finiter Differenzen (7.1.7). Beide Verfahren nutzen letztlich Sekanten des nichtlinearen Problems anstelle der Tangenten. Im Unterschied zum Newton-Krylov-Verfahren wird bei der Vektorextrapolation jedoch das lineare System (7.2.13) nicht gelöst. Auch der numerische Parameter λ aus Gleichung (7.1.7) wird nicht benötigt.

Der methodische Unterschied zwischen dem Newton-Krylov-Verfahren mit der Approximation (7.1.7) und der Vektorextrapolation liegt damit in der Wahl der Sekanten. Die Approximation (7.1.7) bemüht sich, das Verhalten der Tangente mit Hilfe möglichst kurzer Sekanten nachzubilden, also möglichst nahe an der optimalen Tangentenrichtung zu bleiben. Bei der Vektorextrapolation werden dagegen Differenzen von Verschiebungsvektoren verwendet, die aus der Anwendung der Fixpunkt-Gleichung (6.1.24) stammen. Während das approximierte Newton-Verfahren also der Linearisierung des nichtlinearen Problems wenigstens näherungsweise folgt, ist die Vektorextrapolation auf die von der Fixpunkt-Gleichung zufällig bestimmten Richtungen beschränkt. Anders gesagt ist das Vektorextrapolationsverfahren darauf angewiesen, daß die Lösungsrichtung mit den zufällig bestimmten Verschiebungsvektoren beschrieben werden kann.

Die Vektorextrapolation liefert damit bessere Ergebnisse, je näher die Jacobi-Matrix einer Diagonalmatrix kommt (3.5.2). Aus diesem Grund ist das Newton-Krylov-Verfahren für FSI Probleme im Allgemeinen effizienter als die Vektorextrapolation.

Als kleine Ergänzung zur Anwendung der Vektorextrapolation auf Dirichlet-Neumann partitionierte FSI Probleme muß die Ausgangsrelaxation besprochen werden. Um die Folge von Interfaceverschiebungen zu erzeugen, auf denen die Extrapolation beruht, muß die Fixpunkt-Gleichung (6.3.1) mehrfach ausgewertet werden. Dadurch soll eine konvergierende Vektorfolge erzeugt werden, deren Konvergenz von der Vektorextrapolation beschleunigt wird. Hinter der Vektorextrapolation liegt demnach ein Fixpunkt-Verfahren. Wie jedoch in Abschnitt 7.1 besprochen, kann das FSI Interfaceproblem nicht mit einem unrelaxierten Fixpunkt-Verfahren gelöst werden. Das Fixpunkt-Verfahren zum Bestimmen der Verschiebungsfolge muß relaxiert werden.

Die dynamische Relaxation mit dem Aitken Δ^2 Verfahren aus Abschnitt 7.1.4 verträgt sich jedoch nicht mit der Vektorextrapolation. Wird das Aitken Δ^2 Verfahren zum Bestimmen

Algorithmus 20 Vektorextrapolation zum Lösen des partitionierten FSI Problems.

für alle Zeitschritte:

$$\mathbf{d}_{\Gamma,0}^{S,n+1} = \mathbf{d}_{\Gamma}^{S,n}$$

für alle $i = 0, \dots, \max_{\text{iter}}$:

von $j = 0$ **solang** $j < k$:

$$\tilde{\mathbf{d}}_{\Gamma,i+j+1}^{S,n+1} = \mathcal{S}_{\Gamma}^{-1}(\mathcal{F}_{\Gamma}(\mathbf{d}_{\Gamma,i+j}^{S,n+1}))$$

$$\mathbf{d}_{\Gamma,i+j+1}^{S,n+1} = \omega_0 \tilde{\mathbf{d}}_{\Gamma,i+j+1}^{S,n+1} + (1 - \omega_0) \mathbf{d}_{\Gamma,i+j}^{S,n+1}$$

$$\mathbf{r}_{\Gamma,i+j+1}^{n+1} = \mathbf{d}_{\Gamma,i+j+1}^{S,n+1} - \mathbf{d}_{\Gamma,i+j}^{S,n+1}$$

$$\mathbf{R} = \left[\mathbf{r}_{\Gamma,i+1}^{n+1}, \dots, \mathbf{r}_{\Gamma,i+k}^{n+1} \right]$$

Löse $\mathbf{R}^T \mathbf{R} \boldsymbol{\gamma} = \mathbf{0}$ nach $\boldsymbol{\gamma}$ unter der Bedingung $\sum_{j=1}^k \gamma_j = 1$

$\omega_j = \gamma_j + \omega_{j+1}$ für alle $1 \leq j \leq k$ wobei $\omega_{k+1} = 0$

$$\tilde{\mathbf{d}}_{\Gamma,i+k+1}^{S,n+1} = \mathbf{d}_{\Gamma,i}^{S,n+1} + \mathbf{R} \boldsymbol{\omega}$$

$$\mathbf{d}_{\Gamma,i+k+1}^{S,n+1} = \mathcal{S}_{\Gamma}^{-1}(\mathcal{F}_{\Gamma}(\tilde{\mathbf{d}}_{\Gamma,i+k+1}^{S,n+1}))$$

$$\mathbf{r}_{\Gamma,i+k+1}^{n+1} = \mathbf{d}_{\Gamma,i+k+1}^{S,n+1} - \tilde{\mathbf{d}}_{\Gamma,i+k+1}^{S,n+1}$$

falls $\left| \mathbf{r}_{\Gamma,i+k+1}^{n+1} \right| \leq \varepsilon^{\text{FSI}}$:

$$\mathbf{d}_{\Gamma}^{S,n+1} = \mathbf{d}_{\Gamma,i+k+1}^{S,n+1}$$

nächster Zeitschritt

$$i = i + k + 1$$

$$t = t + \Delta t$$

des Relaxationsparameters verwendet, kann die hinterher angewendete Vektorextrapolation keine Verbesserung mehr erzielen. Aus diesem Grund wird ein konstanter Relaxationsparameter eingeführt, dessen einzige Aufgabe es ist, das Fixpunkt-Verfahren vom Divergieren abzuhalten. Der konstante Parameter muß problemspezifisch gewählt werden.

Ein weiteres Lösungsverfahren für FSI Probleme, das in Michler u. a. [115] vorgestellt und in Michler u. a. [116] mehr verschleiert als erläutert wurde, ist trotz des von den Autoren gewählten Namens dem Grundsatz nach ein Vektorextrapolationsverfahren. Die Verwirrung geht zurück auf Washio und Oosterlee [165], die ein nichtlineares Mehrgitter-Verfahren als Vorkonditionierer für die Vektorextrapolation verwendet. Damit wird das Aufstellen der Jacobi-Matrix umgangen. Leider wird in Washio und Oosterlee [165] jedoch die Vektorextrapolation als Approximation des Newton-Krylov-Verfahrens vorgestellt.

Das von Michler u. a. [115] vorgeschlagene Verfahren wurde selten verstanden, dafür aber oft zitiert. Das Verfahren hat einen festen Platz in der Literatur als alternatives Lösungsverfahren für FSI Probleme gefunden, auch wenn das Verfahren in der vorgeschlagenen Form zur Lösung von FSI Problemen ungeeignet ist. Die Schwierigkeiten entstehen durch die von Michler u. a. [115] vorgeschlagenen Erweiterungen des zu Grunde liegenden Extrapolationsverfahrens. Detaillierte Anmerkungen zu dem Algorithmus aus Michler u. a. [115] finden sich in Küttler und Wall [99].

7.4. Weitere Lösungsverfahren für partitionierte FSI Probleme

Die bisher vorgestellten Verfahren bestehen im Anwenden einer Lösungsmethode aus der numerischen Mathematik auf das nichtlineare Interfaceproblem (6.1.24). In der Literatur

wurden weitere Verfahren zur Lösung von FSI Problemen vorgestellt, die das FSI Problem modifizieren oder eine ungewöhnliche Lösungsmethode verwenden. Diese Verfahren sollen hier nur erwähnt werden.

7.4.1. Modell reduzierter Ordnung. In Vierendeels [158, 159] und Vierendeels u. a. [160] wird ein Lösungsverfahren vorgestellt, das mit black-box Feldlösern auskommen und effizienter als Aitkens Δ^2 Verfahren sein soll. Die Idee ist es, das Verhalten der Interfaceoperatoren \mathcal{S}_Γ^{-1} und \mathcal{F}_Γ abzuschätzen, indem man die Verschiebungs-Kraft-Paare, die man beim Auswerten der einzelnen Operatoren

$$\mathbf{d}_{\Gamma,i}^{n+1} = \mathcal{S}_\Gamma^{-1}(\mathbf{f}_{\Gamma,i}^{n+1}) \quad \text{und} \quad \mathbf{f}_{\Gamma,i}^{n+1} = \mathcal{F}_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1}) \quad (7.4.1)$$

erhält, für eine lineare Extrapolation verwendet. Auf der Strukturseite kann so aus einer Interfacekraft $\mathbf{f}_{\Gamma,i}^{n+1}$ die Strukturantwort $\mathbf{d}_{\Gamma,i}^{n+1}$ abgeschätzt werden, ohne den Strukturoperator selbst auswerten zu müssen. Dieses Abschätzen geschieht mit der in Abschnitt 7.3 besprochenen Vektorextrapolation und wird als Modell reduzierter Ordnung bezeichnet, da lediglich das kleine lineare Minimierungsproblem der Extrapolation gelöst werden muß. Für den Fluidoperator läßt sich ein analoges reduziertes Modell aufstellen.

Der Charme dieser Idee besteht nun in der Kombination der reduzierten Modelle beider Seiten. Damit kann im allgemeinen Kopplungsalgorithmus 17 in jedem Durchgang ein Inkrement $\Delta \mathbf{d}_{\Gamma,i}^{n+1}$ berechnet werden, mit dem die gekoppelten linearen Modelle erfüllt werden. Und da die Modelle für beide Interfaceoperatoren mit jeder Iteration um ein Verschiebungs-Kraft-Paar ergänzt werden, besteht die Hoffnung, der Lösung des gekoppelten Systems bald nahe zu kommen.

Der Nachteil dieser Verfahren ist die zusätzliche Komplexität, die sich im Gegensatz zu den bisher besprochenen Verfahren etwas künstlich anfühlt. Diese Verfahren haben deshalb bisher nur geringe Verbreitung gefunden. Eine realistische Einschätzung der Leistungsfähigkeit dieser Verfahren ist im Rahmen dieser Arbeit nicht möglich.

7.4.2. Künstliche Kompressibilität. Das Einbringen von künstlicher Kompressibilität in die Navier-Stokes-Gleichung, das beispielsweise in Riemslagh u. a. [129] und Raback u. a. [126] vorgeschlagen wurde, sei hier nur am Rande erwähnt. Dieses Verfahren ist weder zur Beschleunigung der Konvergenz noch zur Behandlung vollständig eingeschlossener Fluidgebiete, die beispielsweise in Küttler u. a. [96] besprochen werden, geeignet.

Monolithische Newton-Verfahren zum Lösen des FSI Problems

Die in Kapitel 7 vorgestellten Lösungsmethoden bauen auf die nichtlinearen Interfaceoperatoren \mathcal{S}_Γ^{-1} und \mathcal{F}_Γ aus Kapitel 6 auf. Die Interfaceoperatoren abstrahieren für sich nichtlineare Lösungsverfahren für die jeweiligen Felder.

Ein alternativer Lösungsansatz für das FSI Problem, der in diesem Kapitel besprochen wird, besteht darin, das nichtlineare gekoppelte FSI Problem mit einem einzigen Newton-Verfahren zu lösen. Dabei werden alle Freiheitsgrade des FSI Problems, also sämtliche Strukturverschiebungen, Fluidgeschwindigkeiten und -drücke sowie die Netzverschiebungen des Fluidnetzes, von dem Newton-Verfahren gleichzeitig bestimmt. Diese Lösungsverfahren werden als monolithische FSI Verfahren bezeichnet, weil das FSI Problem im Gegensatz zu den partitionierten Verfahren als ein monolithisches Problem behandelt wird.

Wie schon für die partitionierten Verfahren in Kapitel 6.1.3 diskutiert, muß jedes Lösungsverfahren für FSI Probleme die zeitliche Änderung des Fluidgebietes berücksichtigen. Im Rahmen dieser Arbeit wird für das monolithische Verfahren ausschließlich der ALE Ansatz verwendet. Alternativen wie das XFEM Verfahren können in vergleichbarer Weise zur Konstruktion eines monolithischen Lösungsverfahrens verwendet werden, diese Möglichkeiten werden hier jedoch nicht verfolgt.

Das nichtlineare monolithische FSI Problem wird durch das FSI Residuum

$$\mathbf{f}^{\text{FSI}} = \mathbf{f}^{\text{FSI}}(\mathbf{d}^{\text{S},n+1}, \mathbf{u}^{\text{F},n+1}, \mathbf{p}^{\text{F},n+1}, \mathbf{d}^{\text{G},n+1}) = \mathbf{0} \quad (8.0.1)$$

bestimmt, das sich aus dem Residuum des Strukturfeldes $\mathbf{f}^{\text{S},n+1}$ aus Gleichung (4.2.28) und dem Residuum des Fluidfeldes $\mathbf{f}^{\text{F},n+1}$ nach Gleichung (4.3.13), ergänzt um die Koppelungsbedingungen (6.1.1), zusammensetzt. Das lineare ALE Feld ist unphysikalisch und geht nicht explizit ins FSI Residuum (8.0.1) ein.

Im Residuum (8.0.1) sind die Unbekannten die Verschiebung des Strukturgebiets $\mathbf{d}^{\text{S},n+1}$, die Geschwindigkeit $\mathbf{u}^{\text{F},n+1}$ und der Druck $\mathbf{p}^{\text{F},n+1}$ des Fluidgebiets und die Netzverschiebung im Fluid $\mathbf{d}^{\text{G},n+1}$. Wie auch bei den partitionierten Lösungsverfahren wird im Folgenden der Fluiddruck $\mathbf{p}^{\text{F},n+1}$ im Geschwindigkeitsvektor $\mathbf{u}^{\text{F},n+1}$ subsummiert, um die Notation einfach zu halten.

Der Nullpunkt des FSI Residuums \mathbf{f}^{FSI} markiert das dynamische Gleichgewicht sowohl im Inneren von Fluid- und Strukturfeld als auch am Interface. Dementsprechend setzt sich die Abbruchbedingung im nichtlinearen Lösungsverfahren aus mindestens drei Tests zusammen

$$\left| \mathbf{f}_I^{\text{S},n+1} \right| \leq \varepsilon^{\text{S}}, \quad \left| \mathbf{f}_I^{\text{F},n+1} \right| \leq \varepsilon^{\text{F}} \quad \text{und} \quad \left| \mathbf{f}_\Gamma^{\text{S},n+1} + \mathbf{f}_\Gamma^{\text{F},n+1} \right| \leq \varepsilon^\Gamma, \quad (8.0.2)$$

mit denen das dynamische Gleichgewicht im Inneren von Fluid- und Strukturfeld und am Interface sichergestellt wird. Welche Normen und Testkombinationen für die Bedingungen (8.0.2) verwendet werden, ist dabei noch offen. Die Möglichkeiten, Abbruchkriterien zu formulieren, werden in Kapitel 3.7 kurz diskutiert.

In aller Regel wird im Fluidgebiet die Impulsgleichung und die Divergenzfreiheit unabhängig voneinander überprüft, wie in Algorithmus 10 dargestellt.

Zusätzlich zu den Bedingungen (8.0.2) muß die kinematische Kopplungsbedingung (6.1.1)

$$\mathbf{d}_\Gamma^{S,n+1} = \mathbf{d}_\Gamma^{S,n} + \frac{\Delta t}{2} \left(\mathbf{u}_\Gamma^{F,n+1} + \mathbf{u}_\Gamma^{F,n} \right) \quad (8.0.3)$$

erfüllt werden.

Die in Kapitel 4 vorgestellten Lösungsalgorithmen der Einzelfelder verwenden zum Lösen des nichtlinearen Feldproblems das Newton-Verfahren. Das Newton-Verfahren ist auch das am besten geeignete Lösungsverfahren für das nichtlineare monolithische FSI Problem (8.0.1). In dieser Arbeit wird ausschließlich das Newton-Verfahren als Lösungsverfahren zur Behandlung des nichtlinearen monolithischen Problems verwendet.

Die monolithischen Verfahren können aus den partitionierten Verfahren gewonnen werden, indem die nichtlineare Iteration in den Feldlösern mit der nichtlinearen Iteration am Interface, die als Kopplungsiteration fungiert, vertauscht wird. Die Voraussetzung dafür ist, daß im partitionierten Verfahren für beide Felder ein Newton-Verfahren zum Einsatz kommt. Die Interfaceoperatoren \mathcal{S}_Γ^{-1} und \mathcal{F}_Γ werden bei diesem Schritt aufgelöst. In Kapitel 5 wird die Verwandtschaft zwischen den Lösungsverfahren ausführlich besprochen.

Um die nichtlineare Gleichung (8.0.1) mit einem Newton-Verfahren zu lösen, wird die Linearisierung des FSI Residuums aus der Taylorentwicklung benötigt. Das lineare System erhält man als

$$\frac{\partial \mathbf{f}^{\text{FSI}}}{\partial \mathbf{d}^{S,n+1}} \Delta \mathbf{d}^{S,n+1} + \frac{\partial \mathbf{f}^{\text{FSI}}}{\partial \mathbf{u}^{F,n+1}} \Delta \mathbf{u}^{F,n+1} + \frac{\partial \mathbf{f}^{\text{FSI}}}{\partial \mathbf{d}^{G,n+1}} \Delta \mathbf{d}^{G,n+1} = -\mathbf{f}^{\text{FSI}}. \quad (8.0.4)$$

Die Linearisierung (8.0.4) soll aus den Linearisierungen der Einzelfelder aufgebaut werden. Die Details der Linearisierung der Felder sind in den Kapiteln 4.2.6 und 4.3.10 bzw. im Kapitel 6.1.3 dargestellt. In Anlehnung an den inversen Strukturoperator \mathcal{S}_Γ^{-1} aus Algorithmus 14 wird das lineare Strukturproblem mit

$$\mathbf{S}_i \Delta \mathbf{d}_i^{S,n+1} = -\mathbf{f}_i^{S,n+1} \quad (8.0.5)$$

abgekürzt. Entsprechend erfordert das lineare Fluidproblem, wie es im inversen Fluidoperator \mathcal{F}_Γ^{-1} nach Algorithmus 16 auftritt, das Lösen der linearen ALE Gleichung

$$\mathbf{A} \Delta \mathbf{d}_i^{G,n+1} = \mathbf{0} \quad (8.0.6)$$

und der linearen Fluidgleichung

$$\mathbf{F}_i \Delta \mathbf{u}_i^{F,n+1} = -\mathbf{f}_i^{F,n+1} - \mathbf{F}_i^G \Delta \mathbf{d}_i^{G,n+1}, \quad (8.0.7)$$

wobei die Interfaceverschiebung des Netzes $\Delta \mathbf{d}_{\Gamma,i}^{G,n+1}$ von der Fluidgeschwindigkeit am Interface $\Delta \mathbf{u}_{\Gamma,i}^{F,n+1}$ abhängt, die Gleichungen (8.0.6) und (8.0.7) also gekoppelt sind und gleichzeitig gelöst werden müssen.

Mit den Gleichungen (8.0.5), (8.0.6) und (8.0.7) sind die effektiven linearen Systeme angegeben, die in jedem Newton-Schritt von den jeweiligen Feldalgorithmen gelöst werden. Zusammen mit den Kopplungsbedingungen (6.1.1) stellen diese Gleichungen auch das im monolithischen Algorithmus in einem Newton-Schritt zu lösende lineare System dar.

Um den Konvergenzradius des Newton-Verfahrens zu vergrößern, kann der hier skizzierte Algorithmus wie in Kapitel 3.1 besprochen um einen Line-Search-Schritt ergänzt werden. Mögliche Line-Search-Verfahren wurden in Kapitel 3.5 vorgestellt. Diese Technik ist nicht spezifisch für das FSI Problem und wird deshalb an dieser Stelle nicht weiter besprochen.

Der große Vorteil der partitionierten Verfahren ist die Unabhängigkeit der Feldlöser und damit die Modularität des Lösungsverfahrens. Auch bei der Konstruktion des monolithischen Newton-Verfahrens sollen soweit wie möglich die existierenden Feldlöser eingesetzt werden, um ein modulares Lösungsverfahren zu erhalten. Naturgemäß findet diese Modularität bei den monolithischen Verfahren nur innerhalb der Newton-Schleife statt. Auf der Ebene des nichtlinearen Problems sind die monolithischen Verfahren nicht modular, die monolithischen Verfahren erfordern deshalb deutlich größere Eingriffe in existierende Feldlöser als partitionierte Verfahren.

Das lineare System (8.0.8) ist nur schwer in der gegebenen Form zu lösen. In aller Regel ist das System (8.0.8) zu groß, um direkte Lösungsverfahren anwenden zu können. Der Einsatz von den in Kapitel 3.2.2 vorgestellten iterativen Lösungsverfahren wird durch die unterschiedlichen Eigenschaften der einzelnen linearen Feldgleichungen erschwert. Diese Schwierigkeit ist vielfach als Einwand gegen monolithische Lösungsverfahren für FSI Probleme vorgebracht worden.

Es existieren verschiedene Alternativen, das lineare System (8.0.8) aufzustellen und zu lösen, die auf jeweils unterschiedliche Lösungsalgorithmen für das FSI Problem führen. Diese Varianten des monolithischen Lösungsalgorithmus werden im Folgenden diskutiert. Einige Verfahrensvarianten, wie die partitionierten Newton-Verfahren und die auf linearen Block-Gauß-Seidel-Lösern beruhenden Newton-Verfahren sind dabei von geringer, die Block-vorkonditionierten Newton-Verfahren dagegen von großer praktischer Bedeutung. Von Interesse ist aber in jedem Fall die Verwandtschaft dieser Verfahrensvarianten mit den partitionierten Verfahren aus Kapitel 7 und die Beziehung der monolithischen Verfahren untereinander.

8.1. Dirichlet-Neumann partitionierte Newton-Verfahren

Wie in Kapitel 6 die nichtlinearen Feldprobleme Dirichlet-Neumann partitioniert wurden, kann auch das lineare System bestehend aus den Gleichungen (8.0.5), (8.0.6) und (8.0.7) partitioniert werden. Die Abhängigkeiten zwischen den Gleichungen sind in beiden Fällen die gleichen. Das Partitionieren entspricht dem Zerlegen des linearen Systems (8.0.8), wobei der Lagrange-Parameter λ eingesetzt und eliminiert wird. Die Lösungsverfahren aus Kapitel 7 sind dann auf das entstehende lineare Interfacesystem anwendbar.

Auf diese Weise entsteht ein Newton-Verfahren für das vollständige FSI Problem aufbauend auf einem Dirichlet-Neumann partitionierten linearen Lösungsverfahren. Das Gesamtverfahren wurde in Fernández u. a. [49] deshalb als partitioniertes Newton-Verfahren bezeichnet. Wie in Kapitel 9 gezeigt wird, sind die partitionierten Newton-Verfahren für tatsächliche Berechnungen ungeeignet. Für die Systematik der Lösungsverfahren spielt die Klasse der partitionierten Newton-Verfahren dennoch eine große Rolle.

Um das lineare System zu partitionieren, werden die linearen Interfaceoperatoren $\hat{\mathcal{S}}_\Gamma$ und $\hat{\mathcal{F}}_\Gamma$ analog zu den nichtlinearen Interfaceoperatoren \mathcal{S}_Γ und \mathcal{F}_Γ eingeführt. Die linearen Interfaceoperatoren beschreiben die Abhängigkeit zwischen den Verschiebungsincrementen am Interface $\Delta \mathbf{d}_{\Gamma,i}^{n+1}$ und den zugehörigen Interfacekräften $\mathbf{f}_{\Gamma,i}^{n+1}$

$$\mathbf{f}_{\Gamma,i}^{n+1} = \hat{\mathcal{F}}_\Gamma(\Delta \mathbf{d}_{\Gamma,i}^{n+1}) \quad \text{und} \quad \mathbf{f}_{\Gamma,i}^{n+1} = \hat{\mathcal{S}}_\Gamma(\Delta \mathbf{d}_{\Gamma,i}^{n+1}) \quad (8.1.1)$$

jeweils für die Fluid- bzw. die Strukturseite. Das lineare Verschiebungsresiduum läßt sich dann als

$$\hat{\mathbf{r}}_\Gamma(\Delta \mathbf{d}_{\Gamma,i}^{n+1}) = \hat{\mathcal{S}}_\Gamma^{-1}(\hat{\mathcal{F}}_\Gamma(\Delta \mathbf{d}_{\Gamma,i}^{n+1})) - \Delta \mathbf{d}_{\Gamma,i}^{n+1} \quad (8.1.2)$$

angeben. Gleichung (8.1.2) beschreibt die Dirichlet-Neumann-Partitionierung des linearen Systems (8.0.8) und ist damit die Grundlage für die in Kapitel 7 diskutierten Lösungsverfahren.

Im Gegensatz zu den nichtlinearen Interfaceoperatoren können die linearen Operatoren aus Gleichung (8.1.1) explizit angegeben werden. Mit der im linearen System (8.0.8) gezeigten Unterteilung in innere und Interface-Freiheitsgrade ist der einfache lineare Strukturoperator $\hat{\mathcal{S}}_\Gamma$ gerade das Schurkomplement der Strukturmatrix

$$\mathbf{f}_{\Gamma,i}^{n+1} = \hat{\mathcal{S}}_\Gamma(\Delta \mathbf{d}_{\Gamma,i}^{n+1}) = \mathbf{S}_{\Gamma,i} \mathbf{S}_{\Pi,i}^{-1} \mathbf{f}_{\Gamma,i}^{S,n+1} + \left(\mathbf{S}_{\Gamma,i} \mathbf{S}_{\Pi,i}^{-1} \mathbf{S}_{\Gamma,i} - \mathbf{S}_{\Gamma,i} \right) \Delta \mathbf{d}_{\Gamma,i}^{n+1} - \mathbf{f}_{\Gamma,i}^{S,n+1}, \quad (8.1.3)$$

der inverse lineare Strukturoperator entspricht dann dem Invertieren des Schurkomplements (8.1.3)

$$\Delta \mathbf{d}_{\Gamma,i}^{n+1} = \hat{\mathcal{S}}_\Gamma^{-1}(\mathbf{f}_{\Gamma,i}^{n+1}) = \left(\mathbf{S}_{\Gamma,i} \mathbf{S}_{\Pi,i}^{-1} \mathbf{S}_{\Gamma,i} - \mathbf{S}_{\Gamma,i} \right)^{-1} \left(\mathbf{f}_{\Gamma,i}^{n+1} + \mathbf{f}_{\Gamma,i}^{S,n+1} - \mathbf{S}_{\Gamma,i} \mathbf{S}_{\Pi,i}^{-1} \mathbf{f}_{\Gamma,i}^{S,n+1} \right). \quad (8.1.4)$$

Wie schon bei den nichtlinearen Strukturoperatoren \mathcal{S}_Γ und \mathcal{S}_Γ^{-1} werden die linearen Operatoren nicht in der kondensierten Form (8.1.3) bzw. (8.1.4) ausgewertet. Das explizite Berechnen der Inversen der Strukturmatrix mit den inneren Freiheitsgraden $\mathbf{S}_{\Pi,i}^{-1}$ liegt im Allgemeinen außerhalb der Möglichkeiten. Alternativ könnte der Ausdruck $\mathbf{S}_{\Pi,i}^{-1} \mathbf{S}_{\Gamma,i}$ auch durch eine Zerlegung der Matrix $\mathbf{S}_{\Pi,i}$ berechnet werden, doch auch dieser Ansatz ist extrem aufwendig. Siehe dazu auch die Anmerkungen zu den direkten Lösungsverfahren für lineare Gleichungssysteme in Kapitel 3.2.1.

Beim Berechnen des inversen linearen Strukturoperators $\hat{\mathcal{S}}_\Gamma^{-1}$, der in Gleichung (8.1.2) eigentlich benötigt wird, kommt als zusätzliche Schwierigkeit das Invertieren des Ausdrucks $\mathbf{S}_{\Gamma,i} \mathbf{S}_{\Pi,i}^{-1} \mathbf{S}_{\Gamma,i} - \mathbf{S}_{\Gamma,i}$ hinzu. Dieses Invertieren kann, solange der Ausdruck selbst berechnet werden kann, als Lösen eines linearen Gleichungssystems mit Hilfe eines der iterativen Lösungsverfahren aus Kapitel 3.2.2 erledigt werden. Dieser Weg wurde für die Nebendiagonalblöcke im Lösungsverfahren von Matthies und Steindorf [108, 109] gewählt, da dort keine andere Darstellung zur Verfügung steht. Wegen des extremen numerischen Aufwands bei der Zerlegung der Matrix $\mathbf{S}_{\Pi,i}$ ist dieser Ansatz wenig erfolgreich; zumal Näherungslösungen an dieser Stelle unzulässig sind, da iterative Krylov-Verfahren lineare Operatoren verlangen.

Anstelle von Gleichung (8.1.4) wird das lineare System

$$\begin{bmatrix} \mathbf{S}_{\Pi,i} & \mathbf{S}_{\Pi,i} \\ \mathbf{S}_{\Gamma,i} & \mathbf{S}_{\Gamma,i} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{d}_{\Gamma,i}^{S,n+1} \\ \Delta \mathbf{d}_{\Gamma,i}^{S,n+1} \end{bmatrix} = - \begin{bmatrix} \mathbf{f}_{\Gamma,i}^{S,n+1} \\ \mathbf{f}_{\Gamma,i}^{S,n+1} + \mathbf{f}_{\Gamma,i}^{n+1} \end{bmatrix} \quad (8.1.5)$$

mit einem iterativen Krylov-Verfahren gelöst, um den inversen linearen Strukturoperator $\hat{\mathcal{S}}_\Gamma^{-1}$ auszuwerten. Gleichung (8.1.5) ist die effektive Strukturgleichung (8.0.5) ergänzt um die Interfacelast $\mathbf{f}_{\Gamma,i}^{n+1}$. Die Antwort des linearen Strukturoperators auf die zusätzliche Interfacelast $\mathbf{f}_{\Gamma,i}^{n+1}$ ist dann das Verschiebungsinkrement am Interface $\Delta \mathbf{d}_{\Gamma,i}^{S,n+1}$.

Ganz analog wird die Auswertung des linearen Fluidoperators $\hat{\mathcal{F}}_\Gamma$ in Gleichung (8.1.2) durch das Lösen des linearen Systems

$$\mathbf{F}_{\Pi,i} \Delta \mathbf{u}_{\Gamma,i}^{F,n+1} = -\mathbf{f}_i^{F,n+1} - \left(\mathbf{F}_{\Pi,i} + \frac{\Delta t}{2} \mathbf{F}_{\Pi,i}^G \right) \Delta \mathbf{u}_{\Gamma,i}^{n+1} - \mathbf{F}_{\Pi,i}^G \Delta \mathbf{d}_{\Gamma,i}^{G,n+1} \quad (8.1.6)$$

mit der Beziehung

$$\Delta \mathbf{u}_{\Gamma,i}^{n+1} = \frac{2}{\Delta t} \Delta \mathbf{d}_{\Gamma,i}^{n+1} \quad (8.1.7)$$

und dem anschließenden Rückrechnen

$$\mathbf{f}_{\Gamma,i}^{n+1} = \mathbf{f}_{\Gamma,i}^{F,n+1} + \mathbf{F}_{\Gamma,i} \Delta \mathbf{u}_{\Gamma,i}^{F,n+1} + \left(\mathbf{F}_{\Gamma,i} + \frac{\Delta t}{2} \mathbf{F}_{\Gamma,i}^G \right) \Delta \mathbf{u}_{\Gamma,i}^{F,n+1} + \mathbf{F}_{\Gamma,i}^G \Delta \mathbf{d}_{\Gamma,i}^{G,n+1} \quad (8.1.8)$$

realisiert. Das Lösen der Gleichung (8.1.6) nach den Geschwindigkeitsinkrementen im Gebietsinneren $\Delta \mathbf{u}_{\Gamma,i}^{F,n+1}$ und das anschließende Bestimmen der Interface>Last $\mathbf{f}_{\Gamma,i}^{n+1}$ in Gleichung (8.1.8) entspricht dem Einbringen der Interfacegeschwindigkeit $\Delta \mathbf{u}_{\Gamma,i}^{n+1}$ als Dirichlet-Bedingung in die effektive lineare Fluidgleichung (8.0.7).

Wie beim nichtlinearen Fluidoperator \mathcal{F}_Γ tritt auch in Gleichung (8.1.6) die Netzverschiebung $\Delta \mathbf{d}_i^{G,n+1}$ auf der rechten Seite auf, nur geschieht das beim nichtlinearen Operator implizit über das Berechnen der inneren Kräfte, während beim linearen Operator die Sensitivitätsmatrix explizit erscheint. Es ist deshalb nötig, im linearen Fluidoperator zuerst die neuen Netzverschiebungen durch das Lösen des ALE Systems

$$\mathbf{A}_{\Pi} \Delta \mathbf{d}_{\Gamma,i}^{G,n+1} = -\frac{\Delta t}{2} \mathbf{A}_{\Pi} \Delta \mathbf{u}_{\Gamma,i}^{n+1} \quad (8.1.9)$$

zu bestimmen, also die ALE Gleichung (8.0.6) mit einem vorgeschriebenen, über Gleichung (8.1.7) bestimmten Verschiebungsinkrement am Interface zu lösen.

Mit den Gleichungen (8.1.5) bis (8.1.9) ist ein partitionierter Lösungsalgorithmus für das lineare System (8.0.8) beschrieben, wobei der Lagrangeparameter λ eliminiert wurde. Die Lösung des linearen Systems (8.0.8) kann damit mit Hilfe von Gleichung (8.1.2) durch ein Iterationsverfahren gefunden werden. Ausgehend von einem Startwert $\Delta \mathbf{d}_{\Gamma,i,0}^{n+1}$ werden die linearen Operatoren ausgewertet

$$\hat{\mathbf{r}}_{\Gamma,j} = \hat{\mathcal{S}}_\Gamma^{-1}(\hat{\mathcal{F}}_\Gamma(\Delta \mathbf{d}_{\Gamma,i,j}^{n+1})) - \Delta \mathbf{d}_{\Gamma,i,j}^{n+1} \quad (8.1.10)$$

und mit Hilfe der in Kapitel 7 diskutierten Lösungsverfahren wird aus dem Verschiebungsresiduum $\hat{\mathbf{r}}_{\Gamma,j}$ ein Verschiebungsinkrement $\Delta^2 \mathbf{d}_{\Gamma,i,j}^{n+1}$ berechnet. Die neue Verschiebung erhält man mit

$$\Delta \mathbf{d}_{\Gamma,i,j+1}^{n+1} = \Delta \mathbf{d}_{\Gamma,i,j}^{n+1} + \Delta^2 \mathbf{d}_{\Gamma,i,j}^{n+1}. \quad (8.1.11)$$

Da Gleichung (8.1.2) ein lineares System ist, sind die relaxierten Fixpunktverfahren aus Kapitel 3.5 und die Vektorextrapolation aus Kapitel 3.6 ohne Modifikation verwendbar. Das Jacobi-freie Newton-Krylov-Verfahren, das in Kapitel 3.4 vorgestellt wird, ist als Verfahren zur Lösung nichtlinearer Systeme nicht anwendbar. Bei der Lösung des linearen Interfaceproblems (8.1.2) entfällt die Newton-Schleife des Newton-Krylov-Verfahrens und es bleibt ein Jacobi-freier Krylov-Löser als Lösungsverfahren.

Die vorgestellte Dirichlet-Neumann-Zerlegung des linearen Systems (8.0.8) entspricht der bevorzugten Dirichlet-Neumann-Zerlegung des nichtlinearen Systems mit dem Fluidfeld als Dirichlet-Partition und dem Strukturfeld als Neumann-Partition. Die umgekehrte Zuordnung mit dem Strukturfeld als Dirichlet-Partition und dem Fluidfeld als Neumann-Partition ist auch an dieser Stelle möglich, ganz analog zur Diskussion in Kapitel 6.1.5. Doch auch im Falle linearer Feldprobleme läßt sich die Fluidgleichung nur in Verbindung mit der ALE Gleichung nach der Interfaceverschiebung lösen. Diese Zuordnung der Partitionen ist deshalb auch hier unattraktiv.

Wird in Gleichung (8.0.8) die Sensitivitätsmatrix \mathbf{F}_i^G weggelassen, wird also der Einfluß der Netzverschiebung $\Delta \mathbf{d}_i^{G,n+1}$ auf die Linearisierung der Navier-Stokes-Gleichung ignoriert, entkoppelt sich die Fluid- von der ALE Gleichung. Die lineare Fluidgleichung (8.1.6) vereinfacht sich zu

$$\mathbf{F}_{\Pi,i} \Delta \mathbf{u}_{\Gamma,i}^{F,n+1} = -\mathbf{f}_i^{F,n+1} - \mathbf{F}_{\Pi,i} \Delta \mathbf{u}_{\Gamma,i}^{n+1} \quad (8.1.12)$$

und die Rückrechnung der Interfacelast (8.1.8) entsprechend zu

$$\mathbf{f}_{\Gamma,i}^{n+1} = \mathbf{f}_{\Gamma,i}^{F,n+1} + \mathbf{F}_{\Pi,i} \Delta \mathbf{u}_{\Gamma,i}^{F,n+1} + \mathbf{F}_{\Pi,i} \Delta \mathbf{u}_{\Gamma,i}^{n+1}. \quad (8.1.13)$$

Diese Vereinfachung ist durchaus gebräuchlich, beispielsweise machen Hübner u. a. [78], Hübner und Dinkler [77] in einem vergleichbaren Kontext davon Gebrauch, da die Sensitivität \mathbf{F}_i^G oftmals nicht zur Verfügung steht und die Entkopplung der Fluid- und ALE Gleichung eine substanzielle Vereinfachung darstellt. Das lineare Residuum (8.1.2) wird damit

unabhängig von der Netzverschiebung. Der lineare Fluidoperator $\hat{\mathcal{F}}_\Gamma$ vereinfacht sich zum Auswerten der Gleichungen (8.1.12) und (8.1.13). Die ALE Gleichung (8.1.9) muß mit dieser Vereinfachung nur einmal gelöst werden, nachdem die Interfacegleichung (8.1.2) erfüllt ist. Die Kopplung zwischen der Navier-Stokes-Gleichung und der Netzverschiebung wird in diesem Fall der umgebenden Newton-Schleife übertragen, die somit nur eine approximierte Linearisierung zur Verfügung hat und dementsprechend mehr Iterationen benötigen wird.

Der hier diskutierte Lösungsalgorithmus für das FSI Problem ist partitioniert, die linearen Feldprobleme werden unabhängig voneinander gelöst. Das ist insofern bedeutend, als das Lösen der linearen Probleme (8.1.5) bis (8.1.9) die numerisch aufwendigen Operationen darstellen. Auf der Ebene der Newton-Schleife sind die nichtlinearen Felder natürlich gekoppelt. Algorithmus 21 zeigt den gesamten Algorithmus mit Newton-Schleife und partitioniertem linearen Löser mit Aitken Relaxation zur Konvergenzbeschleunigung.

Ein weiteres Verfahren, das auch als partitioniertes Newton-Verfahren interpretiert werden kann, wurde in Matthies und Steindorf [108, 109] vorgestellt und in Matthies u. a. [107] weiter diskutiert. Das Ziel dieses Verfahrens ist es, analog zu den Fixpunktverfahren aus Kapitel 7, unabhängige nichtlineare Feldlöser effizient miteinander zu verbinden. Zu diesem Zweck wird ein Block-Newton-Verfahren auf die Dirichlet-Neumann-Partitionierung des FSI Problems angewendet.

Das Verfahren von Matthies und Steindorf [108, 109] beruht im Kern auf dem Auswerten von linearen Interfaceoperatoren $\hat{\mathcal{S}}_\Gamma^{-1}$ und $\hat{\mathcal{F}}_\Gamma$, wobei diese von nichtlinearen Operatoren \mathcal{S}_Γ^{-1} und \mathcal{F}_Γ approximiert werden, indem nur ein Newton-Schritt beim Lösen der nichtlinearen Feldgleichungen ausgeführt wird. Die Darstellung des Lösungsverfahrens in Matthies und Steindorf [108, 109] basiert allerdings auf nicht explizit angegebenen Fixpunkt-Gleichungen für das nichtlineare Fluid- und Strukturfeld, die bereits alle Kopplungsterme enthalten.

$$\mathbf{0} = \mathbf{u}^{\text{F},n+1} - \overline{\mathcal{F}}(\mathbf{u}^{\text{F},n+1}, \mathbf{d}^{\text{S},n+1}) \quad (8.1.14)$$

$$\mathbf{0} = \mathbf{d}^{\text{S},n+1} - \overline{\mathcal{S}}(\mathbf{u}^{\text{F},n+1}, \mathbf{d}^{\text{S},n+1}) \quad (8.1.15)$$

Dieses nichtlineare Gleichungssystem kann formal linearisiert werden, wobei das zentrale 2×2 Blocksystem

$$\begin{pmatrix} \mathbf{I} - D_{\mathbf{u}^{\text{F},n+1}} \overline{\mathcal{F}} & -D_{\mathbf{d}^{\text{S},n+1}} \overline{\mathcal{F}} \\ -D_{\mathbf{u}^{\text{F},n+1}} \overline{\mathcal{S}} & \mathbf{I} - D_{\mathbf{d}^{\text{S},n+1}} \overline{\mathcal{S}} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{u}^{\text{F},n+1} \\ \Delta \mathbf{d}^{\text{S},n+1} \end{pmatrix} = - \begin{pmatrix} \mathbf{u}^{\text{F},n+1} - \overline{\mathcal{F}}(\mathbf{u}^{\text{F},n+1}, \mathbf{d}^{\text{S},n+1}) \\ \mathbf{d}^{\text{S},n+1} - \overline{\mathcal{S}}(\mathbf{u}^{\text{F},n+1}, \mathbf{d}^{\text{S},n+1}) \end{pmatrix} \quad (8.1.16)$$

entsteht. In Gleichung (8.1.16) wurde dabei die Notation aus Matthies und Steindorf [108, 109] übernommen, um die Linearisierung der Fixpunkt-Operatoren $\overline{\mathcal{F}}$ und $\overline{\mathcal{S}}$ nach den Geschwindigkeiten und Verschiebungen anzuzeigen. Das lineare System (8.1.16) rechtfertigt die Einordnung dieses Verfahrens als monolithisches Newton-Verfahren, auch wenn dahinter eine Dirichlet-Neumann-Partitionierung der nichtlinearen Feldlöser zum Einsatz kommt.

Die Lösung des Blocksystems (8.1.16) erfolgt formal über eine LU Zerlegung, siehe Kapitel 3.2.1. Für das Invertieren der Diagonalblöcke werden unabhängig von der Notation in (8.1.14) und (8.1.15) die existierenden Feldlöser verwendet. Die Blöcke der Nebendiagonale sind dagegen nicht verfügbar. Diese wurden durch die Dirichlet-Neumann-Partitionierung künstlich erzeugt und müssen im Krylov-Löser auf der Basis von finiten Differenzen im Sinne von (7.1.7) approximiert werden. Dabei sind Systeme vom Typ des inversen Strukturoperators (8.1.4) zu lösen.

Algorithmus 21 Newton-Verfahren mit partitioniertem linearen Löser.

für alle Zeitschritte:

$$\mathbf{d}_0^{S,n+1} = \mathbf{d}^{S,n}$$

$$\mathbf{u}_0^{F,n+1} = \mathbf{u}^{F,n}$$

$$\mathbf{d}_0^{G,n+1} = \mathbf{d}^{G,n}$$

Newton-Iteration

für alle $i = 0, \dots, \max_{\text{nIter}}$:

$$\text{berechne } \mathbf{f}_i^{\text{FSI}} = \mathbf{f}_i^{\text{FSI}}(\mathbf{d}_i^{S,n+1}, \mathbf{u}_i^{F,n+1}, \mathbf{d}_i^{G,n+1})$$

$$\text{falls } \left| \mathbf{f}_i^{S,n+1} \right| \leq \varepsilon^S \text{ und } \left| \mathbf{f}_i^{F,n+1} \right| \leq \varepsilon^F \text{ und } \left| \mathbf{f}_i^{S,n+1} + \mathbf{f}_i^{F,n+1} \right| \leq \varepsilon^\Gamma:$$

nächster Zeitschritt

$$j = 1$$

$$\Delta \mathbf{d}_{\Gamma,i,1}^{S,n+1} = \mathbf{0}$$

Schleife über die Felder:

$$\Delta \hat{\mathbf{d}}_{\Gamma,i,j}^{S,n+1} = \hat{\mathcal{S}}_\Gamma^{-1}(\hat{\mathcal{F}}_\Gamma(\Delta \mathbf{d}_{\Gamma,i,j}^{S,n+1}))$$

$$\hat{\mathbf{r}}_\Gamma(\Delta \mathbf{d}_{\Gamma,i,j}^{S,n+1}) = \Delta \hat{\mathbf{d}}_{\Gamma,i,j}^{S,n+1} - \Delta \mathbf{d}_{\Gamma,i,j}^{S,n+1}$$

$$\text{falls } \left| \hat{\mathbf{r}}_\Gamma(\Delta \mathbf{d}_{\Gamma,i,j}^{S,n+1}) \right| \leq \varepsilon_{\text{lin}}:$$

$$\Delta \mathbf{d}_i^{S,n+1} = \Delta \hat{\mathbf{d}}_{i,j}^{S,n+1}$$

$$\Delta \mathbf{u}_i^{F,n+1} = \Delta \hat{\mathbf{u}}_{i,j}^{F,n+1}$$

Krylov-Iteration beendet

$$\omega_j = -\omega_{j-1} \frac{(\hat{\mathbf{r}}_{\Gamma,j-1})^T (\hat{\mathbf{r}}_{\Gamma,j} - \hat{\mathbf{r}}_{\Gamma,j-1})}{|\hat{\mathbf{r}}_{\Gamma,j} - \hat{\mathbf{r}}_{\Gamma,j-1}|^2}$$

$$\Delta \mathbf{d}_{\Gamma,i,j+1}^{S,n+1} = \Delta \mathbf{d}_{\Gamma,i,j}^{S,n+1} + \omega_j \hat{\mathbf{r}}_\Gamma(\Delta \mathbf{d}_{\Gamma,i,j}^{S,n+1})$$

$$j = j + 1$$

lineare Netzverformung berechnen

$$\hat{\mathcal{G}}_\Gamma(\Delta \mathbf{d}_{\Gamma,i}^{S,n+1})$$

$$\mathbf{d}_{i+1}^{S,n+1} = \mathbf{d}_i^{S,n+1} + \Delta \mathbf{d}_i^{S,n+1}$$

$$\mathbf{u}_{i+1}^{F,n+1} = \mathbf{u}_i^{F,n+1} + \Delta \mathbf{u}_i^{F,n+1}$$

$$\mathbf{d}_{i+1}^{G,n+1} = \mathbf{d}_i^{G,n+1} + \Delta \mathbf{d}_i^{G,n+1}$$

$$t = t + \Delta t$$

Das Verfahren von Matthies und Steindorf [108, 109] krankt unter anderem an der Lücke zwischen der zur Darstellung gewählte Notation und den tatsächlich ausgeführten Berechnungen. Insgesamt handelt es sich um ein sehr aufwendiges Verfahren, das keine praktische Bedeutung erlangt hat.

8.2. Lineare Block-Gauß-Seidel Löser

Im linearen System (8.0.8) wurde die Kopplungsbedingung (8.0.9) zwischen Fluid und Struktur als Nebenbedingung eingesetzt, wodurch der Lagrange Parameter λ als zusätzliche Unbekannte auftritt. Bei passenden Netzen kann die Nebenbedingung alternativ in

Die Block-Gauß-Seidel-Iteration kann beendet werden, wenn sowohl die Strukturverschiebung $\Delta \mathbf{d}_{I,i}^{S,n+1}$ als auch die Fluidgeschwindigkeit $\Delta \mathbf{u}_i^{F,n+1}$ konvergiert ist

$$\left| \Delta \mathbf{d}_{I,i,j+1}^{S,n+1} - \Delta \mathbf{d}_{I,i,j}^{S,n+1} \right| \leq \epsilon^S, \quad (8.2.7)$$

$$\left| \Delta \mathbf{u}_{i,j+1}^{F,n+1} - \Delta \mathbf{u}_{i,j}^{F,n+1} \right| \leq \epsilon^F. \quad (8.2.8)$$

Die Tests (8.2.7) und (8.2.8) sind hier nur schematisch angegeben. Möglichkeiten, die Konvergenz zu prüfen, werden in Kapitel 3.7 diskutiert. Der Konvergenztest (8.2.8) schließt natürlich auch die Druckinkremente mit ein.

Eine Schwierigkeit mit dem Block-Gauß-Seidel-Verfahren entsteht durch die Summation der Fluidmatrix (8.2.6), die in Gleichung (8.2.5) verwendet wird. Gleichung (8.2.5) wird in aller Regel mit einem iterativen Krylov-Verfahren gelöst und benötigt dementsprechend einen passenden Vorkonditionierer. Für die unmodifizierte Fluidmatrix \mathbf{F}_i sind geeignete Vorkonditionierer vorhanden, die für die modifizierte Fluidmatrix $\mathbf{F}_{\text{ext},i}$ jedoch nicht zwingend geeignet sind. Als Alternative ist es möglich, die Summation der Matrizen in (8.2.6) nicht explizit auszuführen, sondern implizit im Matrix-Vektor-Produkt des Krylov-Lösers zu behandeln. Damit steht die unmodifizierte Fluidmatrix \mathbf{F}_i als Basis für den Vorkonditionierer zur Verfügung. Allerdings stellt dann der Vorkonditionierer damit eine größere Näherung an die Fluidmatrix dar, wodurch das Krylov-Verfahren vor größere Herausforderungen gestellt wird.

Analog zu dem in Abschnitt 8.1 diskutierten partitionierten Newton-Verfahren kann das iterative Block-Gauß-Seidel-Verfahren mit Hilfe der Relaxationsverfahren aus Kapitel 3.5 oder der Vektorextrapolation aus Kapitel 3.6 beschleunigt werden. Im Unterschied zum partitionierten Newton-Verfahren bezieht sich die Extrapolation hier jedoch auf die gesamte Lösungsspalte in (8.2.1), nicht nur auf die Interfacefreiheitsgrade. Das Newton-Krylov-Verfahren ist an dieser Stelle nicht anwendbar. Das Lösen des Systems (8.2.1) mit einem Krylov-Verfahren stellt ein Verfahren für sich dar und wird in Abschnitt 8.3 im Detail besprochen.

Alternativ zur Formulierung (8.2.1) kann das System (8.0.8) so umgeformt werden, daß die Unbekannte am Interface die Strukturverschiebung $\Delta \mathbf{d}_{\Gamma,i}^{S,n+1}$ ist. In diesem Fall wären die Interfacefreiheitsgrade im Lösungsprozess der Strukturgleichung (8.2.3) zuzuordnen und nicht der Fluidgleichung (8.2.5). Der Hauptgrund, die angegebene Zerlegung vorzuziehen, liegt im Vorkonditionieren der Fluidgleichung. Für große Systeme, die auf parallelen Rechnern gelöst werden müssen, kommen als Vorkonditionierer nur die in Kapitel 3.3.3 besprochenen algebraischen Mehrgitterverfahren in Frage. Diese Verfahren sind in aller Regel auf konstante Knotenblockgrößen angewiesen. Es ist deshalb unvorteilhaft, die Teilmatrix mit den Interfacegeschwindigkeiten $\mathbf{F}_{\Gamma,i}$ von der Fluidmatrix \mathbf{F}_i abzuspalten, da in $\mathbf{F}_{\Gamma,i}$ keine Druckfreiheitsgrade enthalten sind und in der Fluidmatrix \mathbf{F}_i eine beispielabhängige Anzahl losgelöster Druckfreiheitsgrade zurückbleiben würde.

Als einfachstes Beispiel ist das Newton-Verfahren mit dem linearen Block-Gauß-Seidel-Löser und Aitkens Δ^2 Relaxation in Algorithmus 22 dargestellt.

Das Newton-Verfahren mit einem linearen Block-Gauß-Seidel-Löser wird als blockiteratives Kopplungsverfahren in Tezduyar und Sathe [152] vorgestellt.

Eine spezielle Variante dieses Verfahrens, die auf der LU Zerlegung der Struktur- und ALE-Gleichungen beruht, wurde von Dettmer und Perić [32, 33] vorgeschlagen. Die Idee dieser Variante ist es, das explizite Aufstellen der Sensitivitätsmatrix zu vermeiden, stattdessen wird das Struktur- und das ALE-System mehrfach mit verschiedenen rechten Seiten gelöst. Die Linearisierung des Newton-Verfahrens wird auf diese Weise vollständig bestimmt. Allerdings sind Dettmer und Perić [32, 33] wegen des vielfachen Lösens der linearen Systeme

Algorithmus 22 Newton-Verfahren mit linearem Block-Gauß-Seidel-Löser.

für alle Zeitschritte:

$$\mathbf{d}_0^{S,n+1} = \mathbf{d}^{S,n}$$

$$\mathbf{u}_0^{F,n+1} = \mathbf{u}^{F,n}$$

$$\mathbf{d}_0^{G,n+1} = \mathbf{d}^{G,n}$$

Newton-Iteration

für alle $i = 0, \dots, \max_{\text{nIter}}$:

$$\text{berechne } \mathbf{f}_i^{\text{FSI}} = \mathbf{f}_i^{\text{FSI}}(\mathbf{d}_i^{S,n+1}, \mathbf{u}_i^{F,n+1}, \mathbf{d}_i^{G,n+1})$$

$$\text{falls } \left| \mathbf{f}_i^{S,n+1} \right| \leq \varepsilon^S \text{ und } \left| \mathbf{f}_i^{F,n+1} \right| \leq \varepsilon^F \text{ und } \left| \mathbf{f}_i^{S,n+1} + \mathbf{f}_i^{F,n+1} \right| \leq \varepsilon^\Gamma:$$

nächster Zeitschritt

$$j = 1$$

Schleife über die Felder:

$$\mathbf{S}_{\text{II},i} \Delta \hat{\mathbf{d}}_{i,j+1}^S = -\mathbf{f}_{i,i}^{S,n+1} - \frac{\Delta t}{2} \mathbf{S}_{\text{IF},i} \Delta \mathbf{u}_{i,j}^{F,n+1}$$

$$\mathbf{A}_{\text{II}} \Delta \hat{\mathbf{d}}_{i,j+1}^G = -\frac{\Delta t}{2} \mathbf{A}_{\text{IF}} \Delta \mathbf{u}_{i,j}^{F,n+1}$$

$$\mathbf{F}_{\text{ext},i} \Delta \hat{\mathbf{u}}_{i,j+1}^F = -\mathbf{f}_i^{F,n+1} - (\mathbf{F}_{\text{II},i}^G + \mathbf{F}_{\text{IF},i}^G) \Delta \hat{\mathbf{d}}_{i,j+1}^G - \mathbf{S}_{\text{IF},i} \Delta \hat{\mathbf{d}}_{i,j+1}^S$$

$$\text{falls } \left| \Delta \hat{\mathbf{d}}_{i,j+1}^{S,n+1} - \Delta \mathbf{d}_{i,j}^{S,n+1} \right| \leq \varepsilon_{\text{lin}}^S \text{ and } \left| \Delta \hat{\mathbf{u}}_{i,j+1}^{F,n+1} - \Delta \mathbf{u}_{i,j}^{F,n+1} \right| \leq \varepsilon_{\text{lin}}^F:$$

$$\Delta \mathbf{d}_i^{S,n+1} = \Delta \hat{\mathbf{d}}_{i,j+1}^{S,n+1}$$

$$\Delta \mathbf{u}_i^{F,n+1} = \Delta \hat{\mathbf{u}}_{i,j+1}^{F,n+1}$$

$$\Delta \mathbf{d}_i^{G,n+1} = \Delta \hat{\mathbf{d}}_{i,j+1}^{G,n+1}$$

Krylov-Iteration beendet

$$\Delta \mathbf{d}_{i,j+1}^{S,n+1} = \omega_{j+1} \Delta \hat{\mathbf{d}}_{i,j+1}^{S,n+1} + (1 - \omega_{j+1}) \Delta \mathbf{d}_{i,j}^{S,n+1}$$

$$\Delta \mathbf{u}_{i,j+1}^{F,n+1} = \omega_{j+1} \Delta \hat{\mathbf{u}}_{i,j+1}^{F,n+1} + (1 - \omega_{j+1}) \Delta \mathbf{u}_{i,j}^{F,n+1}$$

$$\Delta \mathbf{d}_{i,j+1}^{G,n+1} = \omega_{j+1} \Delta \hat{\mathbf{d}}_{i,j+1}^{G,n+1} + (1 - \omega_{j+1}) \Delta \mathbf{d}_{i,j}^{G,n+1}$$

$$j = j + 1$$

$$\mathbf{d}_{i+1}^{S,n+1} = \mathbf{d}_i^{S,n+1} + \Delta \mathbf{d}_i^{S,n+1}$$

$$\mathbf{u}_{i+1}^{F,n+1} = \mathbf{u}_i^{F,n+1} + \Delta \mathbf{u}_i^{F,n+1}$$

$$\mathbf{d}_{i+1}^{G,n+1} = \mathbf{d}_i^{G,n+1} + \Delta \mathbf{d}_i^{G,n+1}$$

$$t = t + \Delta t$$

auf direkte Lösungsverfahren angewiesen, siehe Kapitel 3.2.1. Gleichzeitig entsteht ein erheblicher dicht besetzter Block in der Systemmatrix. Das Verfahren ist deshalb aufwendig und auf Problemstellungen mit moderater Größe beschränkt.

Im Gegensatz zu dem hier vorgestellten Block-Gauß-Seidel Verfahren enthält das Verfahren von Dettmer und Perić [32] keine Relaxation zur Konvergenzbeschleunigung. Außerdem werden die Felder mit Hilfe von Lagrange-Nebenbedingungen gekoppelt, um auch nicht passende Netze verwenden zu können, siehe dazu auch Park u. a. [120].

8.3. Vorkonditionierte Newton-Krylov-Verfahren

Die linearen Systeme (8.0.8) und (8.2.1) sind zwei Darstellungen des gleichen Problems. Für das lineare System (8.0.8) wurde in Abschnitt 8.1 eine Partitionierung angegeben, auf der aufbauend die Lösungsverfahren verwendet werden können, die in Kapitel 7 auch für das partitionierte nichtlineare Interfaceproblem (6.1.24) Verwendung finden. Das lineare System (8.2.1) kann mit dem in Abschnitt 8.2 vorgestellten Gauß-Seidel-Verfahren gelöst werden. Vom Newton-Krylov-Verfahren abgesehen können die Lösungsverfahren aus Kapitel 7 beim Lösen des Systems (8.2.1) zur Konvergenzbeschleunigung eingesetzt werden.

Das Anwenden des Newton-Krylov-Verfahrens auf das lineare System (8.2.1) führt zu einer neuen Lösungsmethode. Anders als bei der Partitionierung des Systems (8.0.8) erfolgt in Gleichung (8.2.1) keine Reduktion auf das Interface, der lineare Krylov-Löser wird deshalb auf das gesamte System angewendet. Das lineare System (8.2.1) wird somit nicht mehr mit dem Block-Gauß-Seidel-Verfahren, sondern mit dem Krylov-Verfahren gelöst. Die Matrix in Gleichung (8.2.1) wird also direkt für das Matrix-Vektor-Produkt verwendet.

Das Newton-Krylov-Verfahren als Lösungsansatz ist somit die logische Fortführung der Lösungsverfahren aus Abschnitt 8.2, hat aber im Vergleich zu diesen eine neue Qualität, da die Zerlegung der Matrix in Blöcke zunächst vermieden wird. Ganz analog könnte das Newton-Krylov-Verfahren auch für das gekoppelte System (8.0.8) verwendet werden. In diesem Fall würde die Partitionierung, die auf das lineare Interfaceproblem (8.1.2) führt, vermieden werden.

Das Newton-Krylov-Verfahren zu verwenden, um das vom FSI Residuum (8.0.1) definierte nichtlineare Problem zu lösen, ist der direkteste und wegen seiner Einfachheit schönste Lösungsansatz. Dabei spielt die Formulierung des Residuums (8.0.1) eine untergeordnete Rolle. Mit diesem Ansatz entfällt eine Iterationsebene, da der Krylov-Löser gleichzeitig das Lösen der linearen Feldprobleme und die Kopplung der linearen Feldprobleme übernimmt. Das Ergebnis ist das vollständig monolithische FSI Verfahren, wie es beispielsweise in Bazilevs u. a. [7] oder Badia u. a. [5] vorgestellt wurde.

Ein Hauptargument, das gegen die vollständig monolithischen FSI Verfahren vorgetragen wurde, ist das Fehlen eines geeigneten Vorkonditionierers für die zusammengesetzten Systeme (8.0.8) bzw. (8.2.1). Die einzelnen Blöcke der zusammengesetzten Systeme können sehr unterschiedliche Eigenschaften besitzen, so daß die Systeme insgesamt für Krylov-Verfahren denkbar schlecht geeignet sind. Wie in Kapitel 3.2.2 besprochen, werden für Krylov-Verfahren effektive Vorkonditionierer benötigt. Keine der Techniken aus Kapitel 3.3 ist jedoch für die Vorkonditionierung von Blocksystemen wie (8.0.8) oder (8.2.1) geeignet.

Um einen geeigneten Vorkonditionierer zu erhalten, können die in Kapitel 3.3 besprochenen Techniken zu einem Block-Vorkonditionierer zusammengesetzt werden. Block-Vorkonditionierer werden beispielsweise in Knoll u. a. [91] besprochen und in Heil [73] auf das FSI Problem angewendet.

Der Block-Vorkonditionierer für die Systeme (8.0.8) und (8.2.1) folgt den in den Abschnitten 8.1 bzw. 8.2 vorgestellten Zerlegungen. Das lineare System (8.2.1) mit den überlappenden Feldgleichungen ist dabei für das Krylov-Verfahren besser zu handhaben, da die Nebenbedingung im System (8.0.8) ein Sattelpunktproblem erzeugt. Im Folgenden wird deshalb zunächst der Vorkonditionierer für das überlappende System (8.2.1) besprochen.

Aus dem gegebenen Blocksystem (8.2.1) erhält man einen sehr geeigneten Vorkonditionierer, indem man aus der Systemmatrix zwei Kopplungsblöcke herausstreicht, so daß eine Dreiecksmatrix entsteht. In Analogie zu dem Block-Gauß-Seidel-Verfahren aus Abschnitt 8.2 wird die Rückkopplung vom Fluidblock zum Strukturblock und die Kopplung

zum Einsatz, die jeweils für sich einen passenden Vorkonditionierer mitbringen, während die Gleichungen (8.3.7)-(8.3.9) lediglich näherungsweise mit den feldspezifischen Vorkonditionierer gelöst werden.

Diese Näherung trägt erheblich zur Beschleunigung des Newton-Krylov-Verfahrens bei. Das effiziente Lösen der Gleichungen (8.3.7)-(8.3.9) ist wesentlich für das vorkonditionierte Newton-Krylov-Verfahren, da der Vorkonditionierer in jeder Krylov-Iteration angewendet wird. Gleichzeitig besteht kein Bedarf an einer exakten Lösung der Einzelfelder, da das umgebende Krylov-Verfahren gleichzeitig die linearen Blockgleichungen löst und die Kopplung zwischen den Blöcken behandelt.

Die zusätzliche Anforderung, die den Gauß-Seidel-Vorkonditionierer (8.3.7)-(8.3.9) vom Gauß-Seidel-Verfahren aus Abschnitt 8.2 unterscheidet, ist die in Kapitel 3.3 besprochene Linearität des Vorkonditionierers. In Abschnitt 8.2 wird eine dynamische Relaxation mit Aitkens Δ^2 Verfahren zur Konvergenzbeschleunigung vorgeschlagen. Genauso werden mit (8.2.7) und (8.2.8) Konvergenzkriterien für das Beenden der Gauß-Seidel-Iteration angegeben. Beides ist im Vorkonditionierer von Krylov-Lösern nicht möglich. Sowohl der Relaxationsparameter als auch die Anzahl der Iterationen müssen konstant vorgegeben werden, solange nicht die in Kapitel 3.3 erwähnten aufwendigeren flexiblen Krylov-Verfahren eingesetzt werden.

Die Konvergenz des Krylov-Verfahrens beim Lösen des linearen Systems (8.2.1) läßt sich erheblich verbessern, indem die Struktur- und ALE-Anteile der Systemmatrix mit ihren Zeilen- und Spaltensummen skaliert werden. Diese Skalierung entspricht dem Einführen spezieller Diagonalmatrizen als linksseitiger bzw. rechtsseitiger Vorkonditionierer. Am Beispiel des linearen Systems (3.2.1) wären die Vorkonditionierer nach (3.3.3) und (3.3.4) über die Gleichungen

$$\mathbf{M}_{L,ii} = \sum_{j=0}^n |\mathbf{A}_{ij}| \quad (8.3.10)$$

und

$$\mathbf{M}_{R,ii} = \sum_{j=0}^n |\mathbf{A}_{ji}| \quad (8.3.11)$$

gegeben, wobei n die Anzahl der Zeilen und Spalten der Matrix \mathbf{A} ist. Mit \mathbf{A}_{ij} wird der Eintrag der Matrix \mathbf{A} auf der Zeile i und Spalte j bezeichnet.

Um die Vorkonditionierer (8.3.10) und (8.3.11) auf die Struktur- und ALE-Anteile des Blocksystems (8.2.1) anwenden zu können, sind diese Summationen jeweils mit dem Strukturblock $\mathbf{S}_{IL,i}$ und dem ALE-Block \mathbf{A}_{II} auszuführen. Für den Fluidblock ist eine derartige Skalierung nicht sinnvoll. Die zusammengesetzten Vorkonditionierer für das Blocksystem (8.2.1) sind damit

$$\mathbf{M}_L = \begin{bmatrix} \mathbf{M}_L^S & & \\ & \mathbf{I} & \\ & & \mathbf{M}_L^A \end{bmatrix} \quad \text{und} \quad \mathbf{M}_R = \begin{bmatrix} \mathbf{M}_R^S & & \\ & \mathbf{I} & \\ & & \mathbf{M}_R^A \end{bmatrix}. \quad (8.3.12)$$

Diese Vorkonditionierer sind einfach und wirkungsvoll. Allerdings stellen die Vorkonditionierer (8.3.12) keine Alternative zum Block-Vorkonditionierer (8.3.2) dar. Die Vorkonditionierer (8.3.12) werden vielmehr direkt auf das lineare System (8.2.1) angewendet, der Block-Vorkonditionierer (8.3.2) verwendet dann das bereits skalierte System.

Der Algorithmus 23 zeigt beispielhaft das rechtsseitig vorkonditionierte Newton-Krylov-Verfahren zum Lösen des nichtlinearen FSI Problems (8.0.1). Das FSI Problem wird in der in Abschnitt 8.2 diskutierten überlappenden Formulierung verwendet. Die Darstellung eines beliebigen Krylov-Verfahrens folgt dem Schema aus Algorithmus 4. Als Vorkonditionierer im Krylov-Löser kommt lediglich ein Block-Gauß-Seidel-Schritt (8.3.3)-(8.3.5) zum Einsatz.

In Algorithmus 23 wird der Block-Vorkonditionierer (8.3.2) analog zu Algorithmus 4 an drei Stellen verwendet. Um den Berechnungsablauf zu verdeutlichen, zeigt Algorithmus 23 jeweils die angepaßten Blockgleichungen (8.3.3)-(8.3.5). An allen Stellen wäre auch der Einsatz einer Richardson-Iteration (8.3.6) mit den Lösungsschritten (8.3.7)-(8.3.9) möglich. Genauso kann auch das Anwenden der einzelnen Block-Vorkonditionierer, die in Algorithmus 23 als Inversen angedeutet sind, durch eine Richardson-Iteration (8.3.6) verstärkt werden, wie in Kapitel 3.3 diskutiert wird.

Von den Block-Vorkonditionierern wird lediglich eine hinreichend exakte Approximation der einzelnen Block-Gleichungen gefordert. Weitere Annahmen werden an dieser Stelle über die Block-Vorkonditionierer nicht getroffen.

Das Anwenden der zusätzlichen Skalierung des linearen Systems mit (8.3.12) ist in Algorithmus 23 nicht dargestellt.

Das Newton-Krylov-Verfahren mit Block-Vorkonditionierung für FSI Probleme wurde von Heil [73] vorgestellt. Die Motivation für die in diesem Abschnitt vorgestellte Implementierung des Newton-Krylov-Verfahrens geht auf den sehr fruchtbaren persönlichen Austausch mit Matthias Heil im Sommer 2007 zurück, siehe auch die Diskussionen in Heil u. a. [74] und Küttler u. a. [97]. In Heil [73] wurde eine algebraische Behandlung der Netzverschiebung gewählt, so daß ein lineares 2×2 System bestehend aus Fluid- und Strukturgleichung gelöst werden muß.

Ein verwandtes, auf Gleichung (8.0.8) aufbauendes Verfahren mit einem explizitem ALE-Feld wurde in Hübner u. a. [78], Hübner und Dinkler [77] vorgestellt. In diesem Verfahren wurde auf das Berechnen der Sensitivität \mathbf{F}^G verzichtet, so daß der ALE-Block in der Gleichung (8.0.8) vollständig entkoppelt ist und wie in Abschnitt 8.1 besprochen im Nachgang gelöst werden kann. Weiterhin werden in Hübner u. a. [78] direkte Löser als Vorkonditionierer für die Einzelfelder vorgeschlagen, da direkte Löser exakte Lösungsverfahren sind und somit zulässige lineare Vorkonditionierer darstellen. Der Einsatz von direkten Lösern als Vorkonditionierer stellt jedoch eine starke Beschränkung der Parallelität des Lösungsverfahrens dar, siehe Kapitel 3.2.1, und bringt gleichzeitig einen erheblichen numerischen Aufwand mit sich.

In Zhang und Hisada [168] wurde das vorkonditionierte Newton-Krylov-Verfahren mit entkoppeltem ALE-Löser als stark koppelndes Verfahren vorgestellt. In Tezduyar und Sathe [152] wird diese Variante als quasi-direkte Methode bezeichnet. Die direkte Methode in Tezduyar und Sathe [152] ist das vorkonditionierte Newton-Krylov-Verfahren mit voll gekoppelter ALE-Gleichung (8.2.1). Als Vorkonditionierer kommen in Tezduyar und Sathe [152] die Jacobi- und Gauß-Seidel-Verfahren aus Kapitel 3.3.1 zum Einsatz.

8.4. Newton-Krylov-Verfahren mit gekoppeltem Mehrgitter-Vorkonditionierer

Algebraische Mehrgitterverfahren werden in Kapitel 3.3.3 vorgestellt. Im Block-Gauß-Seidel-Prozess können für jeden Block unabhängig algebraische Mehrgitterverfahren als Vorkonditionierer verwendet werden. Abbildung 8.4.1 zeigt schematisch den Block-Gauß-Seidel-Vorkonditionierer aus Abschnitt 8.3 mit algebraischen Mehrgitterverfahren als Vorkonditionierer für die einzelnen Blöcke. Eine Erweiterung der in Abschnitt 8.3 vorgestellten Block-Gauß-Seidel-Vorkonditionierer stellen die gekoppelten Mehrgitter-Vorkonditionierer dar.

Die gekoppelten Mehrgitter-Vorkonditionierer wurden zuerst in Gee u. a. [59] vorgestellt folgen aus dem in Kapitel 5 diskutierten Gedanken, aus der Höhe der Kopplungsschleife im gesamten FSI Lösungsalgorithmus das Spektrum der Lösungsverfahren abzuleiten. Wenn jeder Block des Block-Gauß-Seidel-Prozess (8.3.7)-(8.3.9) mit einem Mehrgitterverfahren gelöst wird, können die einzelnen Ebenen der Mehrgitterverfahren je für sich

Algorithmus 23 Rechtsseitig vorkonditioniertes Newton-Krylov-Verfahren.**für alle Zeitschritte:**

$$\mathbf{d}_0^{S,n+1} = \mathbf{d}^{S,n}$$

$$\mathbf{u}_0^{F,n+1} = \mathbf{u}^{F,n}$$

$$\mathbf{d}_0^{G,n+1} = \mathbf{d}^{G,n}$$

Newton-Iteration

für alle $i = 0, \dots, \max_{\text{nIter}}$:

$$\text{berechne } \mathbf{f}_i^{\text{FSI}} = \mathbf{f}^{\text{FSI}}(\mathbf{d}_i^{S,n+1}, \mathbf{u}_i^{F,n+1}, \mathbf{d}_i^{G,n+1})$$

$$\text{falls } \left| \mathbf{f}_i^{S,n+1} \right| \leq \varepsilon^S \text{ und } \left| \mathbf{f}_i^{F,n+1} \right| \leq \varepsilon^F \text{ und } \left| \mathbf{f}_i^{S,n+1} + \mathbf{f}_i^{F,n+1} \right| \leq \varepsilon^\Gamma:$$

nächster Zeitschritt

$$\tilde{\mathbf{x}}_0 = \left(\mathbf{d}_{I,i}^{S,n+1}, \mathbf{u}_i^{F,n+1}, \mathbf{d}_{I,i}^{G,n+1} \right)^T$$

Anwenden des Block-Vorkonditionierers $\mathbf{p}_0 = \mathbf{M}^{-1} \tilde{\mathbf{x}}_0$

$$\mathbf{p}_0^S = \mathbf{S}_{II,i}^{-1} \mathbf{d}_{I,i}^{S,n+1}$$

$$\mathbf{p}_0^G = \mathbf{A}_{II}^{-1} \mathbf{d}_{I,i}^{G,n+1}$$

$$\mathbf{p}_0^F = \mathbf{F}_{\text{ext},i}^{-1} \left(\mathbf{u}_i^{F,n+1} - \mathbf{S}_{\Gamma I,i} \mathbf{p}_0^S - (\mathbf{F}_{II,i}^G + \mathbf{F}_{\Gamma I,i}^G) \mathbf{p}_0^G \right)$$

$$\mathbf{p}_0 = (\mathbf{p}_0^S, \mathbf{p}_0^F, \mathbf{p}_0^G)^T$$

$$\mathbf{r}_0 = \Delta \mathbf{r}_0 = \mathbf{f}_i^{\text{FSI}} - \mathbf{J}_i^{\text{FSI}} \mathbf{p}_0$$

Krylov-Iteration

für alle $j = 0, \dots, \max_{\text{lIter}}$:# Anwenden des Block-Vorkonditionierers $\mathbf{p}_j = \mathbf{M}^{-1} \Delta \mathbf{r}_{j-1}$

$$\mathbf{p}_j^S = \mathbf{S}_{II,i}^{-1} \Delta \mathbf{r}_{j-1}^S$$

$$\mathbf{p}_j^G = \mathbf{A}_{II}^{-1} \Delta \mathbf{r}_{j-1}^G$$

$$\mathbf{p}_j^F = \mathbf{F}_{\text{ext},i}^{-1} \left(\Delta \mathbf{r}_{j-1}^F - \mathbf{S}_{\Gamma I,i} \mathbf{p}_j^S - (\mathbf{F}_{II,i}^G + \mathbf{F}_{\Gamma I,i}^G) \mathbf{p}_j^G \right)$$

$$\mathbf{v}_j = \mathbf{J}_i^{\text{FSI}} \mathbf{p}_j$$

berechne $\Delta \tilde{\mathbf{x}}_j$ und $\Delta \mathbf{r}_{j+1}$ aus $\Delta \mathbf{r}_k$ und \mathbf{v}_k mit $0 \leq k \leq j$

$$\tilde{\mathbf{x}}_{j+1} = \tilde{\mathbf{x}}_j + \Delta \tilde{\mathbf{x}}_j$$

$$\mathbf{r}_{j+1} = \mathbf{r}_j + \Delta \mathbf{r}_{j+1}$$

$$\text{falls } |\mathbf{r}_{j+1}| / |\mathbf{r}_0| \leq \varepsilon_{\text{lin}}:$$

Krylov-Iteration beendet

Anwenden des Block-Vorkonditionierers $\mathbf{x}_j = \mathbf{M}^{-1} \tilde{\mathbf{x}}_j$

$$\Delta \mathbf{d}_{I,i}^{S,n+1} = \mathbf{S}_{II,i}^{-1} \tilde{\mathbf{x}}_j^S$$

$$\Delta \mathbf{d}_{I,i}^{G,n+1} = \mathbf{A}_{II}^{-1} \tilde{\mathbf{x}}_j^G$$

$$\Delta \mathbf{u}_i^{F,n+1} = \mathbf{F}_{\text{ext},i}^{-1} \left(\tilde{\mathbf{x}}_j^F - \mathbf{S}_{\Gamma I,i} \tilde{\mathbf{x}}_j^S - (\mathbf{F}_{II,i}^G + \mathbf{F}_{\Gamma I,i}^G) \tilde{\mathbf{x}}_j^G \right)$$

$$\Delta \mathbf{d}_{\Gamma,i}^{S,n+1} = \Delta \mathbf{d}_{\Gamma,i}^{G,n+1} = \Delta t \left(1/2 \Delta \mathbf{u}_{\Gamma,i}^{F,n+1} + \mathbf{u}_{\Gamma}^{F,n} \right)$$

$$\mathbf{d}_{i+1}^{S,n+1} = \mathbf{d}_i^{S,n+1} + \Delta \mathbf{d}_i^{S,n+1}$$

$$\mathbf{u}_{i+1}^{F,n+1} = \mathbf{u}_i^{F,n+1} + \Delta \mathbf{u}_i^{F,n+1}$$

$$\mathbf{d}_{i+1}^{G,n+1} = \mathbf{d}_i^{G,n+1} + \Delta \mathbf{d}_i^{G,n+1}$$

$$t = t + \Delta t$$

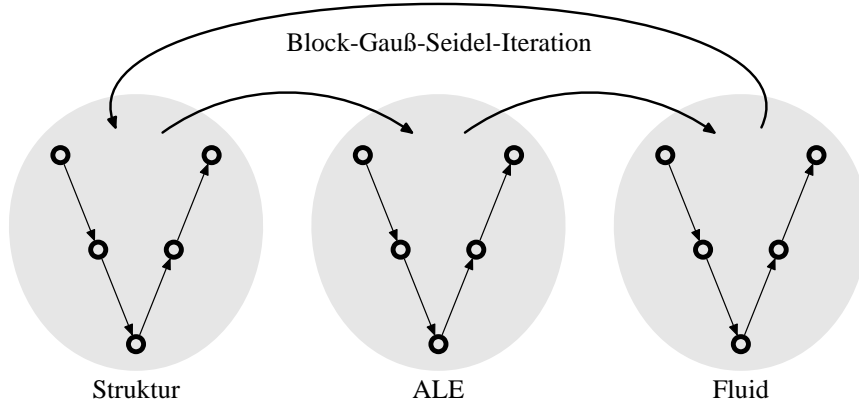


Abbildung 8.4.1: Schematischer Block-Gauß-Seidel-Vorkonditionierer mit Mehrgitterverfahren als Vorkonditionierer für die Einzelblöcke.

mit einem Block-Gauß-Seidel-Verfahren gekoppelt werden. Die Gauß-Seidel-Iteration im Vorkonditionierer, die für die Kopplung der drei Felder zuständig ist, wird dadurch nicht mehr zwischen Block-Vorkonditionierern ausgeführt, sondern auf jeder Ebene im Mehrgitterverfahren. Die Gauß-Seidel-Iteration wird somit mit dem Lauf durch den V-Zyklus der Blöcke vertauscht. Auf diese Weise entsteht ein gekoppelter problemspezifischer Vorkonditionierer, der sich aus den Komponenten der Mehrgitterverfahren zusammensetzt.

In Abbildung 8.4.2 ist der gekoppelte Mehrgitter-Vorkonditionierer schematisch dargestellt. Wie skizziert, besteht das gekoppelte Mehrgitter-Verfahren aus einem V-Zyklus, der wie in Algorithmus 5 angegeben durchlaufen wird. Die Operatoren $\mathbf{P}_k^{\text{FSI}}$ und $\mathbf{R}_k^{\text{FSI}}$ setzen sich aus den entsprechenden Operatoren der Einzelfelder zusammen

$$\mathbf{P}_k^{\text{FSI}} = \begin{bmatrix} \mathbf{P}_k^{\text{S}} & & \\ & \mathbf{P}_k^{\text{F}} & \\ & & \mathbf{P}_k^{\text{G}} \end{bmatrix} \quad \text{und} \quad \mathbf{R}_k^{\text{FSI}} = \begin{bmatrix} \mathbf{R}_k^{\text{S}} & & \\ & \mathbf{R}_k^{\text{F}} & \\ & & \mathbf{R}_k^{\text{G}} \end{bmatrix}. \quad (8.4.1)$$

Die Operatoren $\mathbf{P}_k^{\text{FSI}}$ und $\mathbf{R}_k^{\text{FSI}}$ in Gleichung (8.4.1) sind somit durch die unabhängigen Mehrgitter-Verfahren gegeben. Dementsprechend erhält man nach Gleichung (3.3.14) die Systemmatrix $\mathbf{J}_{k+1}^{\text{FSI}}$ der Ebene $k+1$ aus $\mathbf{J}_k^{\text{FSI}}$ mit

$$\mathbf{J}_{k+1}^{\text{FSI}} = \mathbf{R}_k^{\text{FSI}} \begin{bmatrix} \mathbf{S}_{\text{II}} & \frac{\Delta t}{2} \mathbf{S}_{\text{II}}^{\text{T}} & & \\ \mathbf{S}_{\text{I}\Gamma} & \mathbf{F}_{\text{ext}} & \mathbf{F}_{\text{I}\Gamma}^{\text{G}} + \mathbf{F}_{\text{II}}^{\text{G}} & \\ & \frac{\Delta t}{2} \mathbf{A}_{\text{I}\Gamma}^{\text{T}} & & \mathbf{A}_{\text{II}} \end{bmatrix} \mathbf{P}_k^{\text{FSI}} \quad (8.4.2)$$

$$= \begin{bmatrix} \mathbf{R}_k^{\text{S}} \mathbf{S}_{\text{II}} \mathbf{P}_k^{\text{S}} & \frac{\Delta t}{2} \mathbf{R}_k^{\text{S}} \mathbf{S}_{\text{II}} \mathbf{P}_k^{\text{F}} & & \\ \mathbf{R}_k^{\text{F}} \mathbf{S}_{\text{I}\Gamma} \mathbf{P}_k^{\text{S}} & \mathbf{R}_k^{\text{F}} \mathbf{F}_{\text{ext}} \mathbf{P}_k^{\text{F}} & \mathbf{R}_k^{\text{F}} (\mathbf{F}_{\text{I}\Gamma}^{\text{G}} + \mathbf{F}_{\text{II}}^{\text{G}}) \mathbf{P}_k^{\text{G}} & \\ & \frac{\Delta t}{2} \mathbf{R}_k^{\text{G}} \mathbf{A}_{\text{I}\Gamma} \mathbf{P}_k^{\text{F}} & & \mathbf{R}_k^{\text{G}} \mathbf{A}_{\text{II}} \mathbf{P}_k^{\text{G}} \end{bmatrix}. \quad (8.4.3)$$

Die rechte Seite des linearen Systems und der Vektor der Unbekannten werden ganz analog zwischen den Ebenen übertragen.

In Gleichung (8.4.2) wird die Matrixnotation freizügig verwendet, um inhaltsarme formale Definitionen zu vermeiden. Die Diagonalblöcke der Matrix $\mathbf{J}_{k+1}^{\text{FSI}}$ sind aus den unabhängigen Mehrgitter-Verfahren bekannt. Für den Fluidblock wurde zur Entlastung der Notation die Definition (8.2.6) verwendet. Die Nebenblöcke in Gleichung (8.4.2) kommen beim gekoppelten Mehrgitter-Verfahren neu hinzu. Die Addition $\mathbf{F}_{\text{I}\Gamma}^{\text{G}} + \mathbf{F}_{\text{II}}^{\text{G}}$ stellt dabei das Zusammensetzen der beiden nicht überlappenden Matrizen $\mathbf{F}_{\text{I}\Gamma}^{\text{G}}$ und $\mathbf{F}_{\text{II}}^{\text{G}}$ zu einer rechteckigen Matrix dar. Bei der Multiplikation $\mathbf{R}_k^{\text{F}} \mathbf{S}_{\text{I}\Gamma} \mathbf{P}_k^{\text{S}}$ beschreibt die Matrix $\mathbf{S}_{\text{I}\Gamma}$ die Kopplung der

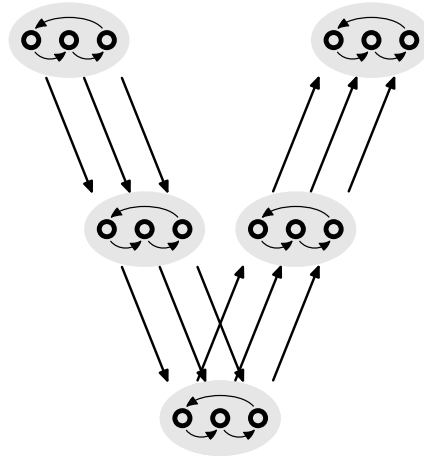


Abbildung 8.4.2: Schematischer gekoppelter Mehrgitter-Vorkonditionierer mit Gauß-Seidel-Iteration auf jeder Gitterebene.

inneren Strukturfreiheitsgrade mit allen Fluidfreiheitsgraden, die Multiplikation mit \mathbf{R}_k^F und \mathbf{P}_k^S ist damit wohldefiniert. Naturgemäß ist die so interpretierte Matrix \mathbf{S}_{Γ} extrem dünn besetzt. Das Produkt $\mathbf{R}_k^F \mathbf{S}_{\Gamma} \mathbf{P}_k^S$ kann deshalb mit sehr geringem Aufwand berechnet werden. Analoge Überlegungen gelten für die Matrizen \mathbf{S}_{Π} und \mathbf{A}_{Π} .

Als dritte Komponente für das gekoppelte Mehrgitter-Verfahren ist das Glättungsverfahren $\hat{\mathbf{S}}_k^{\text{FSI}}$ für jede Ebene zu definieren. Als Glättungsverfahren wird auf jeder Ebene k der Block-Gauß-Seidel-Prozess (8.3.7)-(8.3.9) verwendet, wobei zur Approximation der linearen Systeme (8.3.7)-(8.3.9) lediglich die entsprechenden Glättungsverfahren der unabhängigen Mehrgitter-Verfahren zum Einsatz kommen. Somit kann das Glättungsverfahren für das gekoppelte Problem vollständig aus bereits vorhandenen Komponenten zusammengesetzt werden.

Dieses gekoppelte Glättungsverfahren läßt sich naturgegeben nur auf den Ebenen einsetzen, die in den V-Zyklen aller drei Felder vorkommen. Wenn sich die Größen der linearen Feldprobleme sehr stark unterscheiden und die unabhängigen Mehrgitterverfahren deshalb eine unterschiedliche Anzahl Ebenen besitzen, können nur die gemeinsamen Ebenen über den Block-Gauß-Seidel-Prozess (8.3.7)-(8.3.9) gekoppelt werden. Unterhalb dieser gemeinsamen Ebenen werden dann wie gehabt getrennte Block-Vorkonditionierer eingesetzt.

Eine weitergehende Untersuchung des gekoppelten Mehrgitter-Verfahrens für FSI Probleme liegt außerhalb der vorliegenden Arbeit. Eine detaillierte Darstellung des Verfahrens einschließlich einer numerischen Evaluation ist in Gee u. a. [59] zu finden.

Bewertung der Verfahren

In diesem Kapitel werden die in Küttler u. a. [97] dargestellten numerischen Beispiele wieder aufgegriffen und im Kontext der in dieser Arbeit vorgestellten Verfahren diskutiert. Das Ziel ist es die Lösungsverfahren anhand einiger anspruchsvoller Beispiele miteinander zu vergleichen.

9.1. Druckwelle in einer flexiblen Röhre

Als erstes Beispiel wird die Druckwelle in einer flexiblen Röhre betrachtet, ein Beispiel, das von Gerbeau und Vidrascu [60] als 3D Benchmark für FSI Simulationen in der Hemodynamik vorgeschlagen wurde. Dieses Beispiel ist für Testrechnungen sehr gut geeignet, da kaum Konvektion auftritt und somit ein vergleichsweise grobes Netz verwendet werden kann. Dadurch sind die Testrechnungen vergleichsweise wenig aufwendig und können leicht mehrfach ausgeführt werden. Andererseits ist der Druck bei diesem Beispiel sehr sensibel von der Interfaceverschiebung abhängig. Das nichtlineare FSI Problem muß mit großer Genauigkeit gelöst werden, andernfalls entsteht keine glatte Drucklösung im Zeitverlauf. Das Beispiel ist deshalb zum Vergleichen von partitionierten Lösungsverfahren sehr geeignet.

Gerbeau und Vidrascu [60] verwenden ein elastisches Rohr mit einer Länge von $l = 5$ cm, einem inneren Radius von $r_i = 0.5$ cm und einem äußeren Radius von $r_o = 0.6$ cm. Das Rohr wird an beiden Stirnflächen gehalten. Die Dichte der Struktur ist $\rho^S = 1.2$ g/cm³. Es wird das Saint Venant-Kirchhof Material mit einem E-Modul von $E = 3 \cdot 10^6$ dynes/cm² und der Querdehnzahl $\nu^S = 0.3$ verwendet. Das Fluid hat eine Viskosität von $\mu = 0.03$ Poise und eine Dichte von $\rho^F = 1.0$ g/cm³. Für die Simulation wird eine Zeitschrittweite von $\Delta t = 0.0001$ s gewählt.

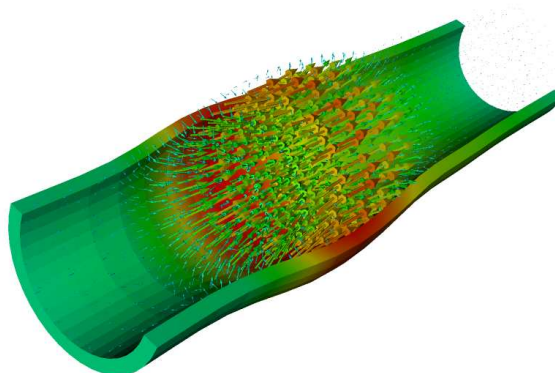


Abbildung 9.1.1: Das elastische Rohr mit einer Strukturichte von $\rho^S = 1.2$ g/cm³ dargestellt mit 10-fach überhöhten Verformungen und Fluidgeschwindigkeiten zum Zeitpunkt $t = 0.0055$ s.

Das Fluid ist zunächst in Ruhe und wird am Einflußrand für $3 \cdot 10^{-3}$ s mit einem Impuls von $1.3332 \cdot 10^4$ dynes/cm² belastet. Als Folge davon läuft eine Druckwelle durch das Rohr, wie in Abbildung 9.1.1 für den Zeitpunkt $t = 0.0055$ s dargestellt ist. Bemerkenswert an diesem Beispiel ist, daß der Einflußrand ein Neumannrand ist. Neumannränder sind als Einflußränder eigentlich nicht zulässig, da die einfließende Strömung von der Randlast allein nicht exakt bestimmt ist. Im vorliegenden Beispiel kommt es wegen des kurzen Zeitintervalls der Last jedoch nur zu einem minimalen Massestrom. Die Viskosität des Fluids genügt in diesem Fall, ein gut gestelltes Problem zu erhalten.

Da beim Berechnen der Druckwelle eine hohe Genauigkeit erforderlich ist, muß für die partitionierten Verfahren eine niedrige Toleranz (6.3.2) am FSI Interface gewählt werden. Für die Berechnungen wurde die mit der Anzahl der Freiheitsgrade am Interface n_Γ gewichtete Toleranz

$$\frac{\left| \mathbf{d}_{\Gamma,i+1}^{n+1} - \mathbf{d}_{\Gamma,i}^{n+1} \right|_2}{\sqrt{n_\Gamma}} = \frac{\left| \mathbf{r}_{\Gamma,i+1}^{n+1} \right|_2}{\sqrt{n_\Gamma}} \leq \varepsilon = 1.0 \cdot 10^{-7} \text{ cm} \quad (9.1.1)$$

gewählt. Die damit vorgeschriebene Genauigkeit muß von den nichtlinearen Feldlösern natürlich erreicht werden, damit das nichtlineare Interfaceproblem gelöst werden kann.

Im Gegensatz dazu wird bei den monolithischen Newton-Verfahren aus Kapitel 8 das gesamte FSI Residuum in der Bedingung (8.0.2) getestet. Dieser Test prüft die Lösung im Inneren beider Gebiete und die Kopplung am Rand. Entsprechend kann dieser Test auch weniger streng ausfallen als die Abbruchbedingungen in den partitionierten Feldlösern. Für das hier betrachtete Beispiel wurden die Bedingungen

$$\left| \mathbf{f}_I^{S,n+1} \right|_2 / \sqrt{n_I^S} \leq \varepsilon^S = 1.0 \cdot 10^{-6} \text{ N} \quad (9.1.2)$$

$$\left| \mathbf{f}_I^{F,n+1} \right|_2 / \sqrt{n_I^F} \leq \varepsilon^F = 1.0 \cdot 10^{-6} \text{ N} \quad (9.1.3)$$

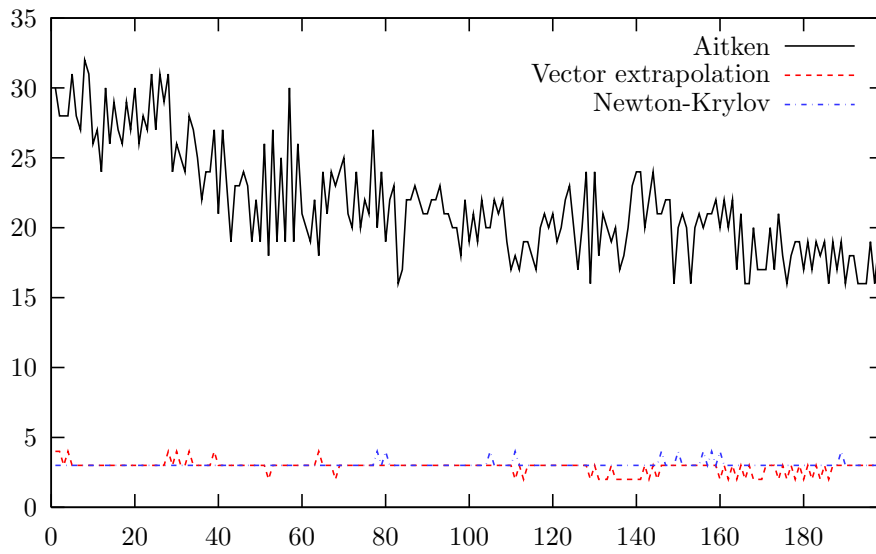
$$\left| \mathbf{f}_\Gamma^{S,n+1} + \mathbf{f}_\Gamma^{F,n+1} \right|_2 / \sqrt{n_\Gamma} \leq \varepsilon^\Gamma = 1.0 \cdot 10^{-6} \text{ N} \quad (9.1.4)$$

gewählt, wobei mit n_I^S , n_I^F und n_Γ jeweils die Anzahl der inneren Freiheitsgrade im Strukturfeld, im Fluidfeld und die Anzahl der Freiheitsgrade am Interface bezeichnet werden.

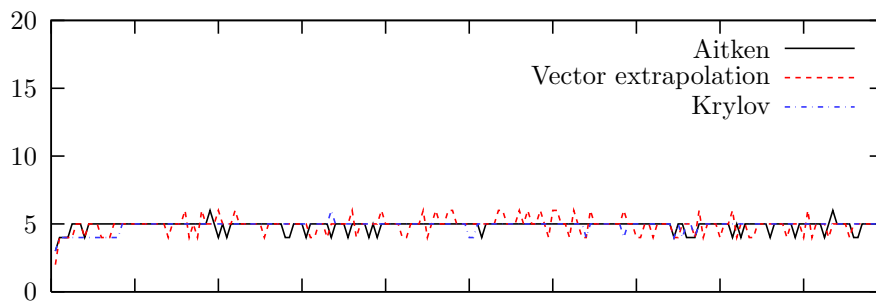
Abbildung 9.1.2 zeigt die Anzahl nichtlinearer Iterationen pro Zeitschritt, die zum Lösen des Beispiels mit unterschiedlichen Verfahren erforderlich sind. Im Falle der partitionierten Lösungsverfahren in Abbildung 9.1.2 (a), bei denen zwei nichtlineare Lösungsverfahren aufeinander aufbauen, wird die Anzahl Iterationen zur Lösung der nichtlinearen Interfacegleichungen dargestellt. Abbildung 9.1.3 zeigt die zur Berechnung benötigte Zeit. Alle Berechnungen wurden parallel mit zwei Prozessen auf einem Knoten mit zwei opteron Prozessoren ausgeführt.

Insgesamt werden die Ergebnisse dreier Lösungsverfahren aufgezeigt: (a) Partitionierte Verfahren, (b) partitionierte Newton-Verfahren und (c) Newton-Verfahren mit linearen Block-Gauß-Seidel-Lösern. Von allen drei Lösungsverfahren werden drei Varianten gezeigt, indem jeweils Aitkens Δ^2 Relaxation, die Vektorextrapolation und das Newton-Krylov-Verfahren auf der jeweiligen Ebene angewendet wird.

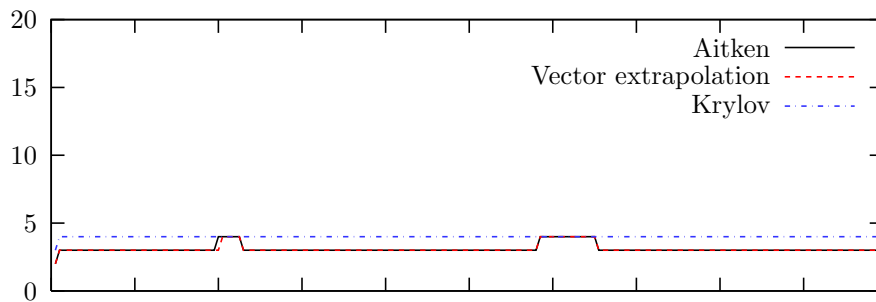
Bei den partitionierten Verfahren werden die drei Lösungsmethoden für das nichtlineare Interfaceproblem verwendet. Die partitionierten Newton-Verfahren verwenden die drei Lösungsverfahren zur Lösung des linearen partitionierten Interfaceproblems. Das Interface-Newton-Krylov-Verfahren wird dabei zu einem Interface-Krylov-Verfahren verkürzt. Im letzten Diagramm werden schließlich zwei Newton-Verfahren mit linearen Block-Gauß-Seidel-Lösern und ein Block-Gauß-Seidel-vorkonditioniertes Newton-Krylov-Verfahren



(a) Partitionierte Verfahren mit nichtlinearem Löser am Interface

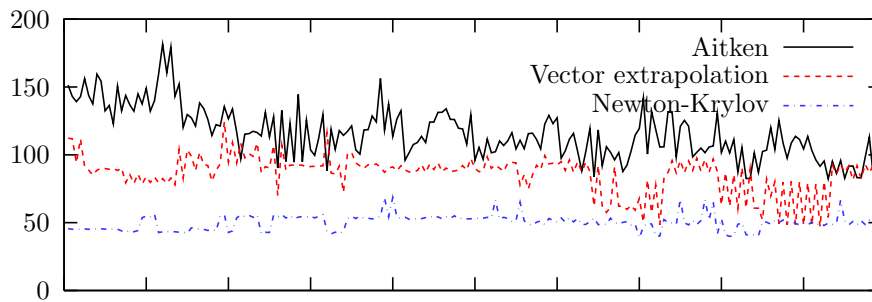


(b) Partitionierte Newton-Verfahren mit linearem Löser am Interface

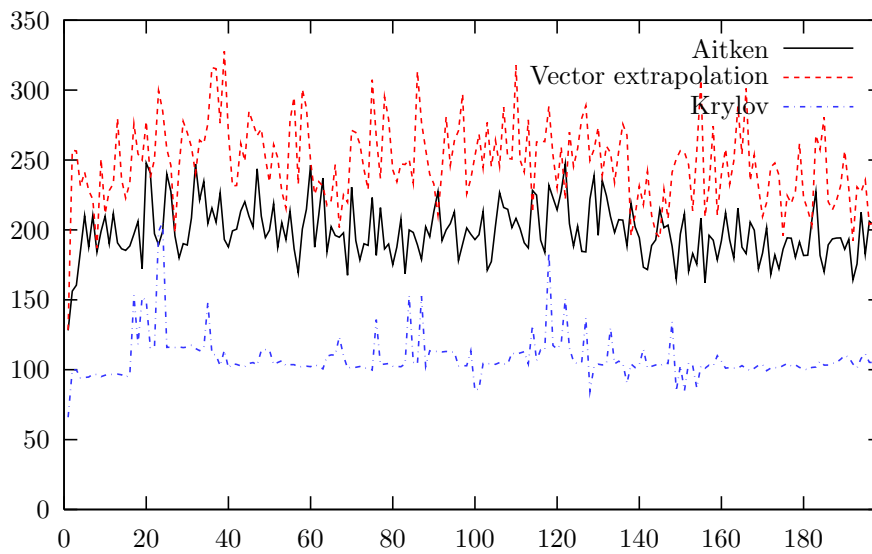


(c) Newton-Verfahren mit linearem Block-Gauß-Seidel Löser

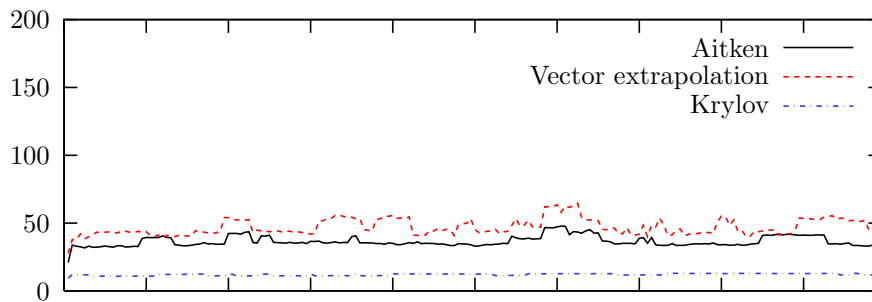
Abbildung 9.1.2: Anzahl nichtlinearer Iterationen pro Zeitschritt für die Druckwelle im elastischen Rohr mit der Strukturichte $\rho^S = 1.2 \text{ g/cm}^3$. Als Lösungsmethoden werden (a) partitionierte Verfahren mit nichtlinearem Löser am Interface, (b) partitionierte Newton-Verfahren mit linearem Löser am Interface und (c) Newton-Verfahren mit linearem Block-Gauß-Seidel-Löser eingesetzt.



(a) Partitionierte Verfahren mit nichtlinearem Löser am Interface



(b) Partitionierte Newton-Verfahren mit linearem Löser am Interface



(c) Newton-Verfahren mit linearem Block-Gauß-Seidel Löser

Abbildung 9.1.3: Rechenzeit pro Zeitschritt für die Druckwelle im elastischen Rohr mit der Strukturichte $\rho^S = 1.2 \text{ g/cm}^3$. Als Lösungsmethoden werden (a) partitionierte Verfahren mit nichtlinearem Löser am Interface, (b) partitionierte Newton-Verfahren mit linearem Löser am Interface und (c) Newton-Verfahren mit linearem Block-Gauß-Seidel-Löser eingesetzt.

gezeigt. In Abbildung 9.1.2 (c) werden also unterschiedliche Lösungsverfahren nebeneinander gestellt, die sich jedoch als direkte Anwendung der drei Lösungsmethoden auf den linearen Block-Gauß-Seidel-Löser begreifen lassen.

Das partitionierte Fixpunkt-Verfahren mit Aitkens Δ^2 Relaxation benötigt eine Menge nichtlinearer Iterationen, wie in Abbildung 9.1.2 (a) gezeigt wird. Die Vektorextrapolation und das Jacobi-freie Newton-Krylov-Verfahren benötigen dagegen etwa drei Iterationen. Für die Newton-Verfahren ist die Zahl der nichtlinearen Iterationen noch weniger aufregend. Im Durchschnitt werden für alle drei Varianten des partitionierten Newton-Verfahrens fünf nichtlineare Iterationen benötigt. Die durchschnittliche Anzahl an Iterationen für die Gauß-Seidel basierten Newton-Verfahren liegt zwischen drei und vier. Solange das lineare Problem innerhalb des Newton-Verfahrens mit hinreichender Genauigkeit gelöst wird, sollte die Anzahl an Iterationen stets klein sein, andernfalls wurde der Zeitschritt zu groß gewählt oder aus einem anderen Grund der Konvergenzradius des Newton-Verfahrens verlassen. Die Effizienz der Newton-Verfahren hängt damit entscheidend von der Effizienz der verwendeten linearen Lösungsverfahren ab.

Die in Abbildung 9.1.2 (b) gezeigte höhere Anzahl an Iterationen für die partitionierten Newton-Verfahren im Vergleich zu den Gauß-Seidel basierten Newton-Verfahren in Abbildung 9.1.2 (c) tritt auf, weil bei den partitionierten Newton-Verfahren die Sensitivitätsmatrix \mathbf{F}^G nicht berücksichtigt wurde, um das ALE Feld außerhalb des Lösungsverfahrens für das lineare Interfaceproblem berechnen zu können. Darüber hinaus zeigen sich die partitionierten Newton-Verfahren als numerisch anspruchsvoll. Insbesondere wird vom linearen Strukturlöser eine große Robustheit gefordert, da die nicht auskonvergierten Fluidlasten im Lösungsverfahren eine beachtliche Herausforderung für den Strukturlöser darstellen. Bei der Berechnung des Beispiels wurde für die partitionierten Newton-Verfahren deshalb ein direkter Löser im Strukturfeld eingesetzt. Gleichzeitig wurden bei allen drei Verfahrensvarianten sehr viele Iterationen für das Lösen des linearen Interfaceproblems benötigt. Aus diesem Grund erfordern die partitionierten Newton-Verfahren mehr Rechenzeit als die anderen Lösungsverfahren, wie in Abbildung 9.1.3 gezeigt wird.

An dem gezeigten Beispiel ist zu sehen, daß die in Kapitel 8.1 vorgestellten partitionierten Newton-Verfahren für FSI Berechnungen nicht geeignet sind. Diese Erkenntnis wurde auch von Fernández u. a. [49] gewonnen. Partitionierte Newton-Verfahren werden in dieser Arbeit deshalb nicht weiter verwendet.

Das partitionierte Fixpunkt-Verfahren mit Aitken Relaxation, das in Abbildung 9.1.2 (a) mit „Aitken“ bezeichnet wird, benötigt vergleichsweise viele nichtlineare Iterationen. Da in jeder Iteration aber lediglich beide nichtlineare Felder einmal gelöst werden müssen, eine Iteration also lediglich aus dem Berechnen des nichtlinearen Interface-Residuums besteht, ist die in Abbildung 9.1.3 (a) gezeigte Lösungszeit aller partitionierter Verfahren noch immer vergleichbar, auch wenn das Jacobi-freie Newton-Krylov-Verfahren einen deutlichen Vorsprung aufweist.

Beim Jacobi-freien Newton-Krylov-Verfahren wurde an dieser Stelle die Darstellung des Matrix-Vektor-Produkts im Krylov-Verfahren über die Ableitungen der nichtlinearen Interfaceoperatoren (7.1.8), unter Einbeziehung der Sensitivitätsanteile in der Fluidableitung (7.1.10), gewählt. Das Newton-Krylov-Verfahren zeigt damit das beste mögliche Verhalten. Im Besonderen wird kein Differenzenparameter benötigt. Gleichzeitig ist dieses Lösungsverfahren nicht ohne Eingriffe aus vorhandenen Einzelalgorithmen zusammensetzbar.

Das Interfaceproblem (6.1.24), das mit einem Jacobi-freien Newton-Krylov-Verfahren gelöst werden soll, ist nichtlinear. Das Krylov-Verfahren für sich geht jedoch von einem linearen Verhalten des Matrix-Vektor-Produkts aus. Auf die damit verbundenen Schwierigkeiten wurde in Kapitel 3.4.1 bereits hingewiesen. Vor allem kann das Abbruchkriterium

des Krylov-Verfahrens nur sehr grob sein. Im vorliegenden Beispiel wurde eine relative Toleranzgrenze von

$$\frac{|\mathbf{r}_{\Gamma,i}|_2}{|\mathbf{r}_{\Gamma,0}|_2} \leq 0.1 \quad (9.1.5)$$

im iterativen linearen Löser gewählt. Dieser Wert hat sich für das Lösen von partitionierten FSI Problemen mit Jacobi-freien Newton-Krylov-Verfahren bewährt. Da der eigentliche Lösungsaufwand bei den partitionierten Verfahren von den Feldalgorithmen übernommen wird, die Interfacegleichung wie in Kapitel 7 besprochen die Tendenz hat, um die gesuchte Lösung herum zu laufen, genügt es für das Lösungsverfahren am Interface, eine vergleichsweise grobe Verbesserung der Interfaceverschiebung vorzuschlagen. Hohe Genauigkeit macht sich an dieser Stelle nicht bezahlt.

Die gleiche Überlegung trifft auch auf die Vektorextrapolation als Lösungsverfahren für partitionierte FSI Probleme zu. Auch für die Vektorextrapolation wurde die relative Toleranz (9.1.5) verwendet. Darüber hinaus müssen für die Vektorextrapolation noch zwei Parameter gewählt werden. Zum einen benötigt jeder Extrapolationsschritt k vorliegende Interfaceverschiebungen. Im vorliegenden Beispiel wurde $k = 10$ gewählt. Außerdem müssen die k Interfaceverschiebungen für sich eine konvergierende Folge bilden. Um das zu erreichen, ist eine feste Basis-Relaxation zu wählen. Als Folge dieser Basis-Relaxation ist die Vektorextrapolation ein ausgesprochen robustes Lösungsverfahren, auch bei Problemstellungen, die mit partitionierten Verfahren nicht mehr effizient gelöst werden können. In diesem Beispiel wurde die feste Relaxation $\omega = 0.01$ gesetzt.

Die auf Block-Gauß-Seidel-Lösern basierenden Newton-Verfahren sind die effizientesten Verfahren im Vergleich, wie in Abbildung 9.1.3 (c) zu sehen ist. Hier existiert jedoch ein großer Unterschied zwischen den auf linearen Block-Gauß-Seidel-Lösern basierenden Verfahren einerseits und dem Krylov-Verfahren mit Block-Gauß-Seidel-Vorkonditionierer andererseits.

Vereinfacht gesagt bewirkt das Krylov-Verfahren eine Beschleunigung des Lösungsverhaltens des Block-Gauß-Seidel-Vorkonditionierers. Damit ist das Krylov-Verfahren ein Beschleunigungsverfahren, ganz analog zur Aitken Relaxation oder der Vektorextrapolation. Der große Geschwindigkeitsvorteil entsteht jedoch, weil das Krylov-Verfahren geringe Anforderungen an die Genauigkeit des Vorkonditionierers stellt. Auch im hier betrachteten Beispiel werden im Block-Gauß-Seidel-Vorkonditionierer für das Lösen der Blockgleichungen lediglich Vorkonditionierungsverfahren eingesetzt.

Um weitere Einblicke zu erhalten, wurde das gleiche Beispiel mit einer Strukturdichte von $\rho^S = 12 \text{ g/cm}^3$ erneut berechnet. In diesem Fall läuft die Druckwelle langsamer durch das Rohr. Die benötigte Rechenzeit für die einzelnen Verfahren—wegen Untauglichkeit wurden die partitionierten Newton-Verfahren bereits ausgeschlossen—wurde in Abbildung 9.1.4 dargestellt. Für das partitionierte Verfahren mit Aitken Relaxation werden hier substantiell weniger nichtlineare Iterationen benötigt. Davon abgesehen ändert sich die Anzahl der nichtlinearen Iterationen nicht, auf eine Darstellung wird deshalb verzichtet.

Wie in Förster u. a. [52] diskutiert wird, erleichtert das Erhöhen der Strukturdichte das Lösen des partitionierten FSI Problems. Entsprechend sinkt die pro Zeitschritt benötigte Rechenzeit in Abbildung 9.1.4 im Vergleich zu Abbildung 9.1.3. Die einzige Ausnahme ist das vorkonditionierte Newton-Krylov-Verfahren, dessen Rechenzeit konstant bei etwa 10s bleibt. Es handelt sich hier um ein monolithisches Verfahren mit einer technischen Partitionierung im Vorkonditionierer.

In einem weiteren Schritt wurde die Strukturdichte auf $\rho^S = 120 \text{ g/cm}^3$ erhöht. Das Beispiel hat damit nichts mehr mit dem ursprünglichen biomechanischen Hintergrund zu tun und dient lediglich dem Verständnis der Berechnungsverfahren. Die Druckwelle läuft hier

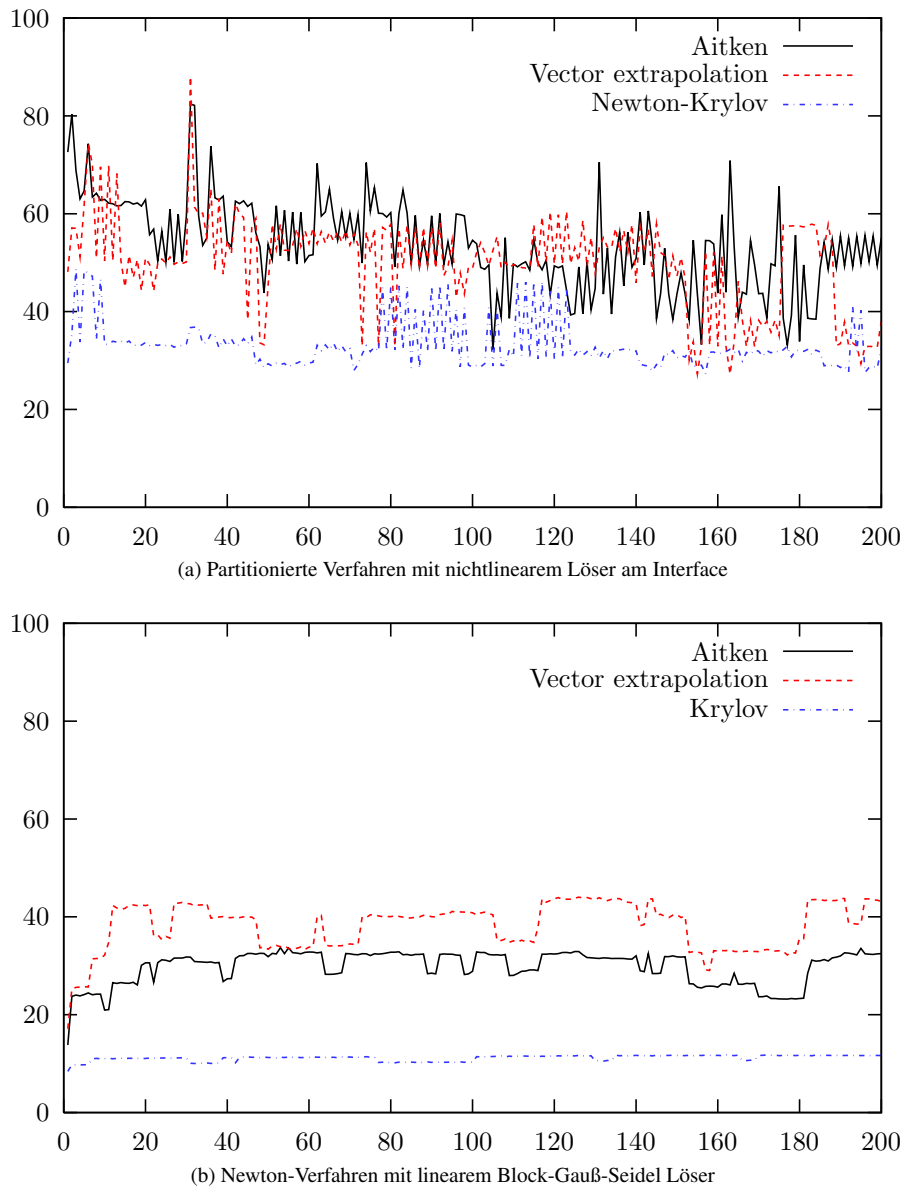


Abbildung 9.1.4: Rechenzeit pro Zeitschritt für die Druckwelle im elastischen Rohr mit der Strukturweite $\rho^S = 12 \text{ g/cm}^3$. Als Lösungsmethoden werden (a) partitionierte Verfahren mit nichtlinearem Löser am Interface und (b) Newton-Verfahren mit linearem Block-Gauß-Seidel-Löser eingesetzt.

sehr langsam durch das Rohr. Die benötigten Rechenzeiten sind in Abbildung 9.1.5 dargestellt.

Der Berechnungsaufwand aller Rechenverfahren mit Ausnahme des vorkonditionierten Newton-Krylov-Verfahrens ist erneut gesunken. Das vorkonditionierte Newton-Krylov-Verfahren ist unabhängig von der Strukturweite, bleibt jedoch das mit Abstand schnellste der betrachteten Verfahren.

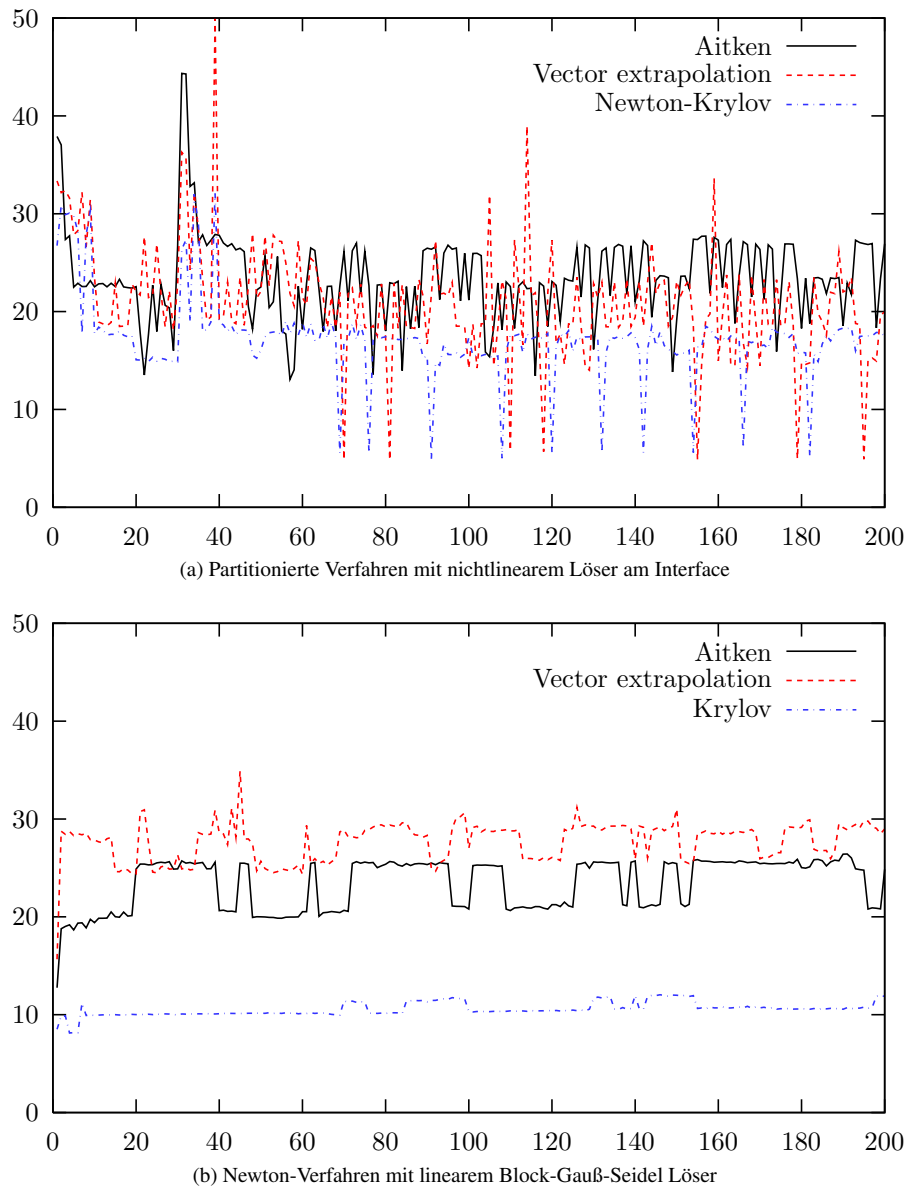


Abbildung 9.1.5: Rechenzeit pro Zeitschritt für die Druckwelle im elastischen Rohr mit der Strukturichte $\rho^S = 120 \text{ g/cm}^3$. Als Lösungsmethoden werden (a) partitionierte Verfahren mit nichtlinearem Löser am Interface und (b) Newton-Verfahren mit linearem Block-Gauß-Seidel-Löser eingesetzt.

9.2. FSI Simulation eines Bronchialbaums

Beim zweiten Beispiel handelt es sich um die Simulation der Luftströmung im oberen Teil der unteren Atemwege in einer patientenspezifischen Geometrie. Die Geometrie wurde aus medizinischen Bilddaten gewonnen. Um den Berechnungsaufwand im Rahmen zu halten—es geht an dieser Stelle um eine Demonstration der Algorithmen, nicht um eine

realistische Simulation der Atmung—werden nur die ersten sechs Generationen betrachtet. Die kleinskaligen Anteile unterhalb der sechsten Generation wurden weggeschnitten. Damit hat der hier betrachtete Teil der unteren Atemwege eine Höhe von 239 mm, eine Breite von 169 mm und eine Tiefe von 78 mm. Es gibt 67 Ausflußränder, die hier als spannungsfreie Ränder modelliert werden. In der Simulation wird die Einatmung mit einer vorgeschriebenen Einflußgeschwindigkeit von bis zu 1.5 mm/ms betrachtet.

Für die Luft wird ein Newtonsches Fluid mit der Dichte $\rho^F = 1 \cdot 10^{-6} \text{ g/mm}^3$ und der kinematischen Viskosität $\nu^F = 0.015 \text{ mm}^2/\text{ms}$ verwendet. Um das Fluidgebiet herum wird das Strukturgebiet—d.h. die Atemwegswand—mit einer Dicke von 0.1 mm angenommen. Für die Struktur wird das Saint Venant-Kirchhof Material, mit dem E-Modul $E = 6.75 \cdot 10^{-3} \text{ N/mm}^2$, der Querdehnzahl $\nu^S = 0.42$ und der Dichte $\rho^S = 1 \cdot 10^{-3} \text{ g/mm}^3$, verwendet. Die patientenspezifische Geometrie ist in Abbildung 9.2.1 dargestellt.

Das Fluidnetz ist hexaeder-dominant mit insgesamt 1113350 Elementen und 884931 Knoten. Das ALE-Gebiet entspricht dem Fluidgebiet, somit besitzt das Problem 3539724 Freiheitsgrade im Fluid und 2654793 Freiheitsgrade im ALE. Das Strukturnetz besteht aus 171988 Elementen und 248594 Knoten. Entsprechend existieren 745782 Freiheitsgrade in der Struktur.

Die Berechnungen wurden auf 24 Prozessoren eines opteron-Clusters mit Infiniband Netzwerk ausgeführt. In Abbildung 9.2.2 wird die Anzahl der nichtlinearen Iterationen und die Berechnungszeit pro Zeitschritt für die ersten 100 Schritte dargestellt. Die drei Lösungsverfahren, die für dieses Beispiel zum Einsatz kommen, sind das Fixpunkt-Verfahren mit Aitken Relaxation, das Jacobi-freie Newton-Krylov-Verfahren und das Block-vorkonditionierte Newton-Krylov-Verfahren. Die anderen Verfahren, die in Abschnitt 9.1 verwendet werden, sind für realistische FSI Probleme weniger geeignet und kommen hier deshalb nicht zum Einsatz.

Das Fixpunkt-Verfahren mit Aitken Relaxation erfordert auch in diesem Beispiel sehr viel mehr nichtlineare Iterationen als beide Newton-Verfahren. Die Newton-Verfahren benötigen etwa vier Iterationen, was für ein Newton-Verfahren mit passender Linearisierung erwartet werden kann, wenn der Ausgangspunkt der Iteration innerhalb des Konvergenzradius des Verfahrens liegt. In Bezug auf die Berechnungszeiten zeigen sich die drei Verfahren deutlich differenzierter. Die beiden partitionierten Verfahren benötigen deutlich mehr Rechenzeit als das Block-vorkonditionierte Newton-Krylov-Verfahren. Diese Beobachtung wiederholt sich. Das Block-vorkonditionierte Newton-Krylov-Verfahren ist sowohl einfacher zu handhaben, da nur zwei Abbruchkriterien aufeinander abgestimmt werden müssen, als auch deutlich effizienter als die partitionierten Verfahren. Die Einstellungen für die linearen Löser und die Vorkonditionierer wurden in allen drei Verfahren vergleichbar gewählt. Insbesondere wurden algebraische Mehrgitterverfahren als Vorkonditionierer für alle Felder eingesetzt.

Dieses Beispiel dient vornehmlich dazu zu zeigen, daß die an einem kleinen Testbeispiel demonstrierten Lösungsverfahren auch für große parallele Simulationen einsetzbar sind. Besondere Aufmerksamkeit benötigen bei großen Simulationen die Abbruchkriterien (9.1.1) und (9.1.2). Die Wichtung mit der Anzahl an Unbekannten, die eingeführt wird, um dem Aufsummieren vieler kleiner Fehlerquadrate in der L_2 -Norm entgegenzuwirken, versteckt einzelne größere Fehler, wenn ausschließlich die L_2 -Norm geprüft wird. Als Ausweg sollte deshalb zusätzlich die L_∞ -Norm geprüft werden. In dem hier vorgestellten Beispiel wurde anstatt dessen eine sehr geringe Toleranz von $1 \cdot 10^{-10}$ in (9.1.1) vorgegeben. Analog wurde die gewichtete Toleranz in (9.1.2) zu $1 \cdot 10^{-11}$ gesetzt, da das gesamte Problem deutlich mehr Freiheitsgrade enthält als das Interface allein.

Das Lösen sehr großer FSI Probleme mit dem block-vorkonditionierten Newton-Krylov-Verfahren verdient eine gesonderte Betrachtung. In diesem Verfahren werden effiziente

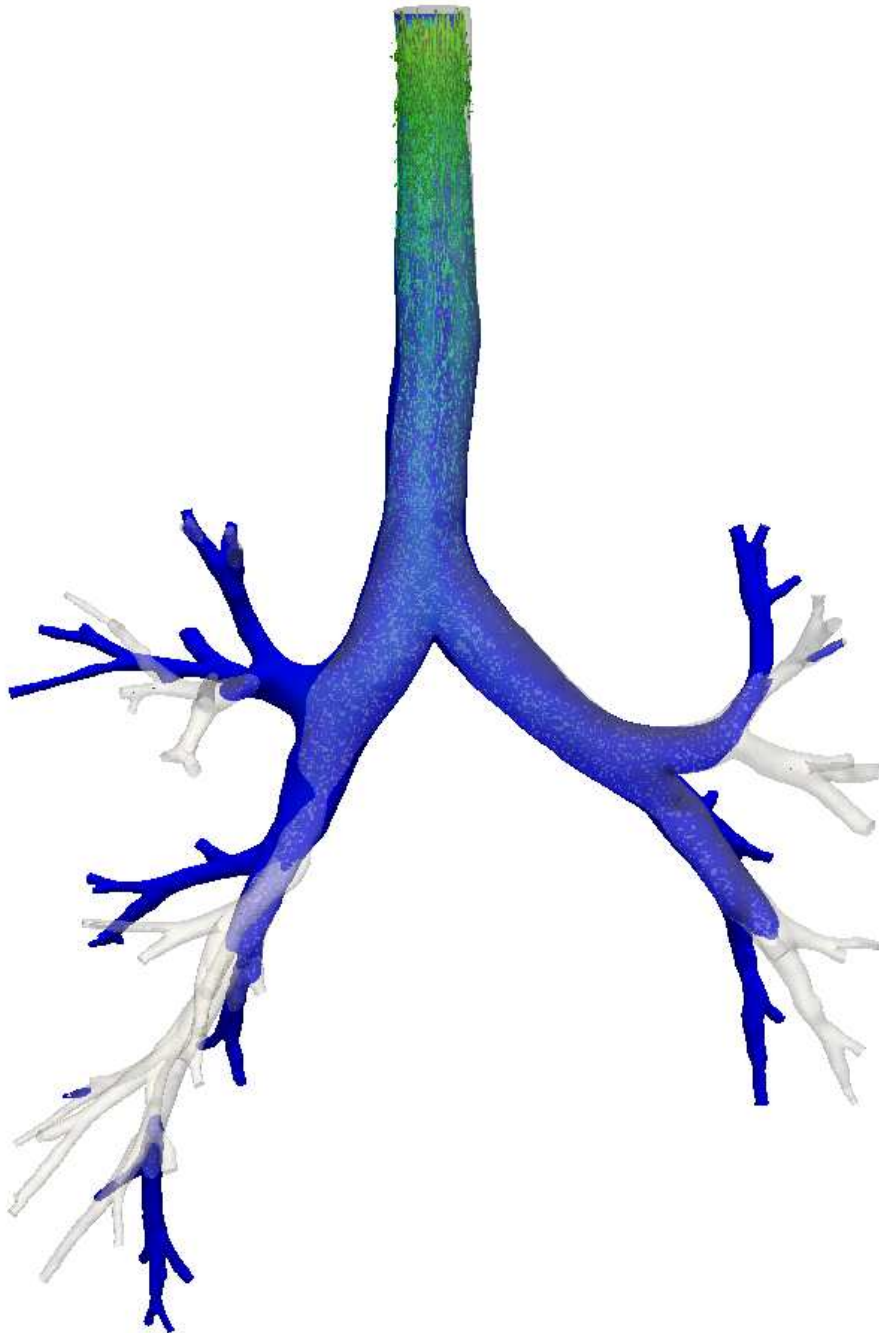


Abbildung 9.2.1: Patientenspezifisches Modell der unteren Atemwege mit sechs Generationen.

Mehrgitter-Verfahren als Block-Vorkonditionierer genutzt, um die Lösung im Inneren der einzelnen Felder abzuschätzen, während das Erfüllen der Kopplungsbedingung die Aufgabe der Gauß-Seidel-Iteration ist. Dabei kann ein Ungleichgewicht in der Vorkonditionierung entstehen, bei dem die Vorkonditionierung im Inneren der Felder sehr gut, am Interface dafür aber recht schlecht ist. In einem solchen Fall entstehen auffällig große Residuen

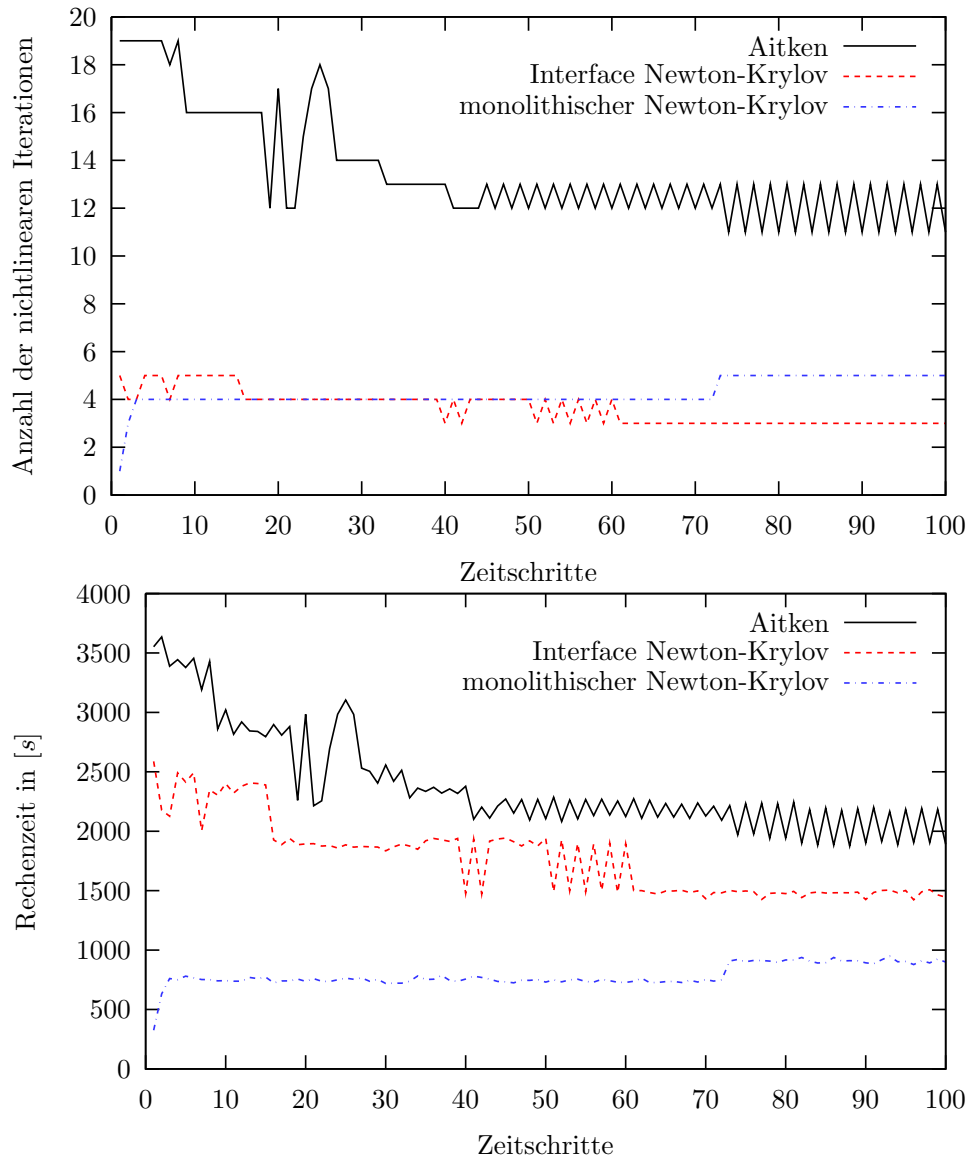


Abbildung 9.2.2: Anzahl der nichtlinearen Iterationen und Rechenzeit pro Zeitschritt für die Atemwegssimulation auf 24 Prozessoren.

am Interface, die wie oben diskutiert mit einem auf der L_∞ -Norm basierenden Test geprüft werden sollten. Dieser L_∞ -Norm-Test betrifft sowohl das nichtlineare Residuum im Newton-Verfahren als auch das lineare Residuum im Krylov-Verfahren. Insbesondere im Krylov-Verfahren sind entsprechende Tests notwendig, da der lineare Löser andernfalls dazu tendiert, die Lösung nach nur wenigen Iterationen als konvergiert zu betrachten. Darunter leidet die Kopplung zwischen den Feldern und im Ergebnis werden mehr (kostspielige) Newton-Iterationen benötigt als eigentlich erforderlich.

Dieses beobachtete Ungleichgewicht im Vorkonditionierer ist die Motivation für die in Kapitel 8.4 vorgestellten gekoppelten Mehrgitter-Vorkonditionierer. Detaillierte Vergleiche der unterschiedlichen Block-Vorkonditionierer für das Newton-Krylov-Verfahren beim Lösen von FSI Problemen werden in Gee u. a. [59] vorgestellt.

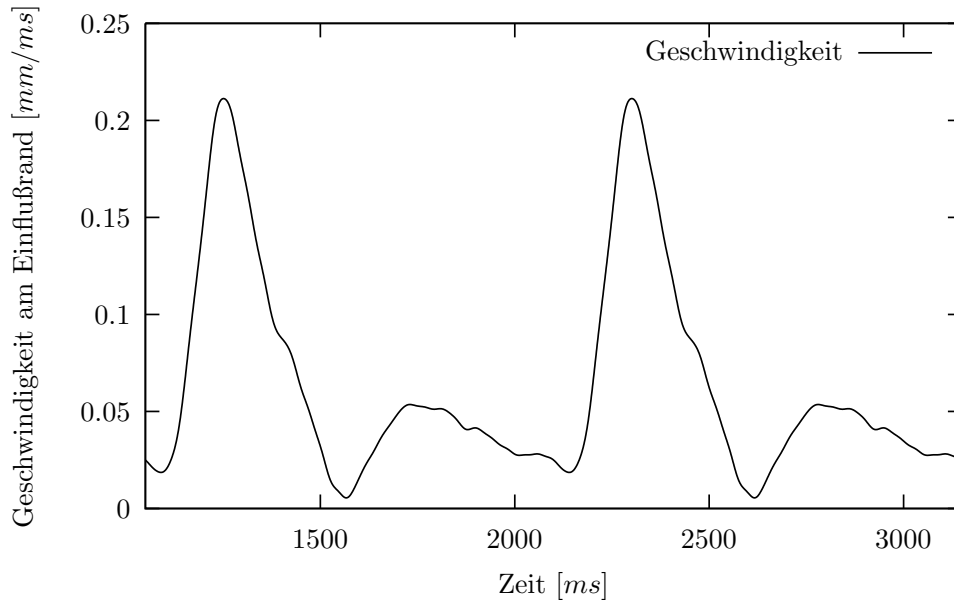


Abbildung 9.3.1: Geschwindigkeit am Einflußrand für zwei Herzzyklen mit einer Dauer von je 1050ms.

9.3. FSI Simulation eines abdominellen Aortenaneurysmas

Das dritte Beispiel ist die Simulation der Blutströmung in einem abdominellen Aortenaneurysma. Bei abdominellen Aortenaneurysmen handelt es sich um bleibende krankhafte Erweiterungen der abdominellen Aorta, die mit einem erheblichen Rupturrisiko verbunden sind.

Ein Reißen des Aneurysmas tritt auf, wenn im Zuge der Lastwechsel während des Herzzyklus die vom Gewebe aufnehmbaren Spannungen überschritten werden. Aus diesem Grund muß zur Beurteilung des Rupturrisikos sowohl die Belastung des Gewebes als auch dessen Widerstandsfähigkeit abgeschätzt werden. Das realistische Schätzen von patientenspezifischen Materialparametern bildet einen Forschungsgegenstand für sich. Die Spannungen in der Aterienwand können dagegen mit den zur Verfügung stehenden Berechnungsmethoden bestimmt werden. Der Ausgangspunkt sind auch hier die aus medizinischen Bilddaten gewonnenen patientenspezifischen Geometrien.

Im Rahmen dieser Arbeit bildet die Anwendung von FSI Lösungsverfahren auf CT-basierten Geometrien den Gegenstand des Interesses. Die medizinische Diskussion um die tatsächliche Vorhersage des Rupturrisikos kann an dieser Stelle nicht geführt werden. Diese Diskussion wird beispielsweise von Humphrey und Taylor [81] aufgenommen.

An dieser Stelle soll als Beispiel ein patientenspezifisches abdominelles Aortenaneurysma simuliert werden. Dazu wird am proximalen Rand die physiologische periodische Einflußgeschwindigkeit vorgeschrieben. In Abbildung 9.3.1 sind zwei Perioden der Einflußgeschwindigkeit dargestellt. Am distalen Ende wird der Zusammenhang zwischen Geschwindigkeit und Druck mit einem Windkesselmodell mit drei Parametern beschrieben.

Das umschließende Gewebe wird an beiden Enden künstlich gehalten. Dieses Festhalten entspricht zwar nicht den physiologischen Bedingungen, stellt aber andererseits keine Einschränkung für die FSI Simulation dar. Die Dicke der Aortenwand wird mit 1 mm angenommen. Die Wand wird mit einem Neo-Hooke Material modelliert. Der E-Modul wird

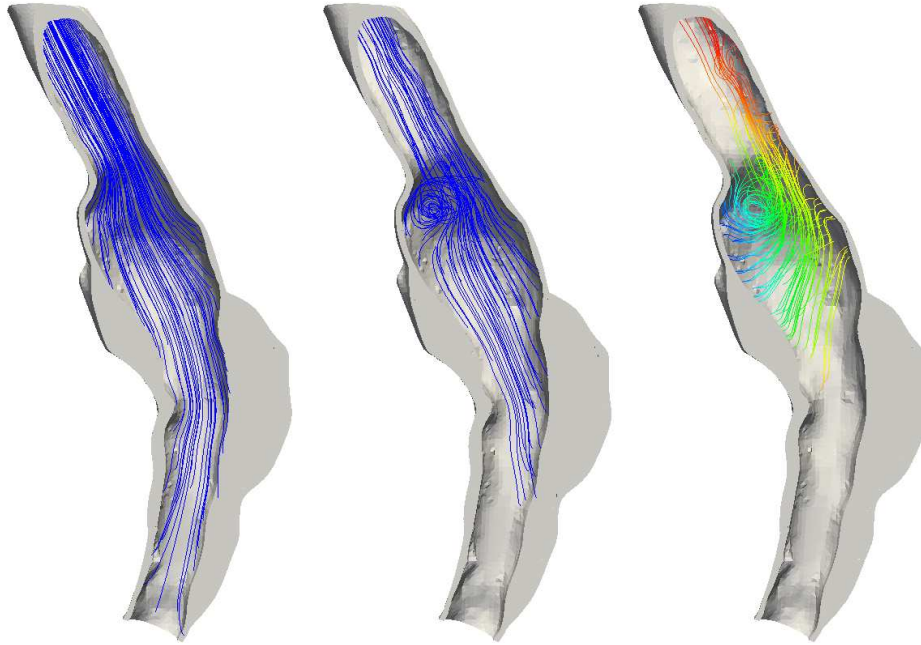


Abbildung 9.3.2: Schnitt durch das Aneurysma mit Stromlinien zu den Zeiten $t = 152$ ms, $t = 788$ ms und $t = 961$ ms.

zu $E = 1.044 \text{ N/mm}^2$ gewählt, die Querdehnzahl zu $\nu^S = 0.45$ und die Dichte zu $\rho^S = 1 \cdot 10^{-3} \text{ g/mm}^3$. Für die Ablagerungen im Aneurysma wird ebenfalls ein Neo-Hooke Material mit dem E-Modul $E = 1.044 \cdot 10^{-1} \text{ N/mm}^2$, der Querdehnzahl $\nu^S = 0.45$ und der Dichte $\rho^S = 1 \cdot 10^{-3} \text{ g/mm}^3$ verwendet. Das gesamte Strukturgebiet wird mit 159603 Elementen vernetzt, wodurch 271830 Freiheitsgrade entstehen.

Zur Modellierung des Blutes wird das Newtonsche Fluid mit einer kinematischen Viskosität von $\nu^F = 0.004 \text{ mm}^2/\text{ms}$ und einer Dichte von $\rho^F = 1 \cdot 10^{-3} \text{ g/mm}^3$ verwendet. Das Dichteverhältnis zwischen Fluid und Struktur ist damit sehr kritisch für die FSI Lösungsalgorithmen. Am Ausflußrand wird das Fluidgebiet verlängert, um einen störungsfreien Ausfluß zu erhalten. Das Fluidnetz besteht insgesamt aus 71974 Elementen und enthält 226852 Freiheitsgrade. Das ALE-Netz, das das gleiche Gebiet beschreibt, trägt weitere 170139 Freiheitsgrade bei. Die Abbildung 9.3.2 zeigt drei Schnitte durch das Aneurysma in der jeweiligen deformierten Lage einschließlich der Stromlinien des Fluids.

Das Beispiel wurde mit dem Block-vorkonditionierten Newton-Krylov-Verfahren aus Kapitel 8.3 auf 8 opteron Prozessoren berechnet. In Abbildung 9.3.3 wird die Anzahl der nichtlinearen Iterationen dargestellt, genauso wie die zur Berechnung benötigte Zeit und die Summe der Anzahl an Krylov-Iterationen im Zeitschritt. Obwohl die Anzahl nichtlinearer Iterationen praktisch konstant bleibt, gibt es einen erheblichen Anstieg an Krylov-Iterationen mit zunehmender Verformung des Aneurysmas. Dieser Anstieg ist in der Berechnungszeit nur gedämpft wiederzufinden, da eine Krylov-Iteration lediglich aus einer Matrix-Vektor-Multiplikation und dem Auswerten des Block-Vorkonditionierers besteht.

Die hier gezeigte Aneurysmen-Simulation konnte in einem voll entwickelten Flußzustand nur mit dem Block-vorkonditionierten Newton-Krylov-Verfahren ausgeführt werden. Alle partitionierten Verfahren sind aufgrund der für die Partitionierung sehr ungünstigen Materialparameter beim Auskonvergieren des ersten Zeitschritts gescheitert. Es kann zwar nicht

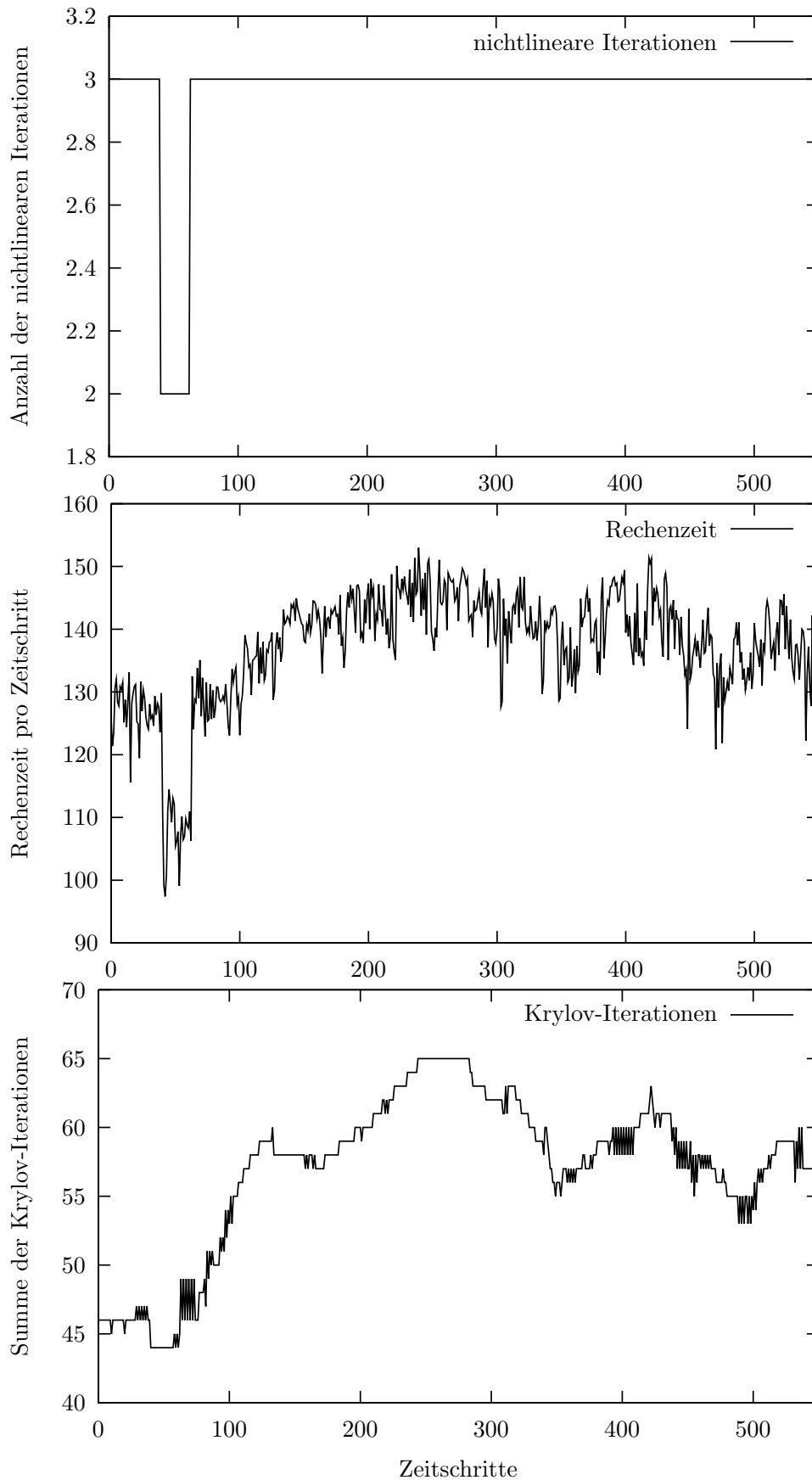


Abbildung 9.3.3: Aneurysma-Simulation mit dem Newton-Krylov-Verfahren: Anzahl der nichtlinearen Iterationen, Rechenzeit und Summe der Krylov-Iterationen.

ausgeschlossen werden, daß einzelne partitionierte Verfahren mit sehr sorgfältig gewählten Parametern in der Lage wären eine Lösung zu finden, von effizientem Lösungsverhalten kann in einem solchen Fall aber keine Rede mehr sein. Partitionierte Verfahren werden aus diesem Grund nicht für die Aneurysmen-Simulation herangezogen.

Ausblick

Mit dem besprochenen Spektrum an Lösungsverfahren für das diskretisierte FSI Problem steht eine breite Auswahl an Lösungsalgorithmen zur Verfügung. Als tragfähig haben sich dabei die traditionellen partitionierten Verfahren und die vorkonditionierten Newton-Krylov-Verfahren erwiesen. Die dazwischen liegenden Verfahren vervollständigen das Methodenspektrum, besitzen jedoch keine praktische Relevanz.

Das Newton-Krylov-Verfahren mit einem problemspezifischen Vorkonditionierer hat sich im direkten Vergleich als das überlegene Verfahren zum Lösen aufwendiger FSI Probleme gezeigt. Bei der Auswahl eines Lösungsverfahrens für ein konkretes FSI Problem sind jedoch auch Faktoren zu berücksichtigen, die außerhalb der hier diskutierten algorithmischen Aspekte liegen. So bleibt die einfachere Konstruktion von partitionierten Lösungsverfahren aus vorhandenen Feldlösern ein schwerwichtiges Argument zugunsten des partitionierten Ansatzes.

Für die Weiterentwicklung der Lösungsverfahren für FSI Probleme stellt das vorkonditionierte Newton-Krylov-Verfahren die zu empfehlende Basis dar. Die Beschleunigung der vorhandenen Methoden bleibt die Herausforderung, um reale Probleme berechnen zu können. Das Newton-Krylov-Verfahren läßt sich durch das Einsetzen von sehr effektiven, problemangepaßten Vorkonditionierern wesentlich beschleunigen. Die in dieser Arbeit vorgetragene Erfolge basieren auf algebraischen Mehrgitterverfahren. Die Suche nach den am besten geeigneten Mehrgitterverfahren für das FSI Problem ist dabei jedoch noch nicht beendet.

Auch kann die für das Verfahrensspektrum genutzte Sichtweise auf die Lösungsalgorithmen verwendet werden, um weitere, möglicherweise effektive Lösungsalgorithmen zu konstruieren. Speziell kann das Newton-Krylov-Verfahren mit gekoppeltem Mehrgitter-Vorkonditionierer, bei dem der V-Zyklus des Mehrgitterverfahrens als Bestandteil des Lösungsverfahrens in Betracht gezogen wird, als Ausgangspunkt verwendet werden, um das Mehrgitterverfahren weiter oben im Gesamtalgorithmus anzuordnen. Entsprechende Verfahren sind softwaretechnisch sehr anspruchsvoll. Über die zu erwartende Leistungsfähigkeit können an dieser Stelle jedoch nur Mutmaßungen angestellt werden.

Eine weitere Möglichkeit die Effizienz der Newton-Krylov-Verfahren zu erhöhen, wäre die zeitliche und räumliche Adaptivität des Lösungsverfahrens. Dazu sind adaptive Feldlöser miteinander zu verbinden, wobei die Steuerung der Adaptivität beide Felder gleichzeitig beachten müßte.

Weiterhin sind die vorgestellten Lösungsmethoden unabhängig vom eingesetzten Diskretisierungsverfahren. Bei Bedarf können die Algorithmen leicht auf NURBS-basierte Diskretisierungen oder dergleichen erweitert werden.

Schließlich ist es auch stets möglich, die Formulierungen der Einzelfelder zu überdenken. Wenn es gelingt, den numerischen Aufwand der einzelnen Felder für sich zu reduzieren, beispielsweise indem lediglich Näherungslösungen gesucht werden, kann das gesamte Lösungsverfahren erheblich beschleunigt werden. An dieser Stelle haben die partitionierten Verfahren den größeren Vorteil, da sie keine Annahme über den algorithmischen Aufbau

der Feldlöser machen und demzufolge jegliches Näherungsverfahren einsetzen können. Dem steht natürlich der zum Teil erhebliche Effizienzvorteil der Newton-Krylov-Verfahren ohne jede Näherung gegenüber.

Die Newton-Krylov-Verfahren mit spezifischem Vorkonditionierer versprechen aber auch für weitere ähnlich gelagerte Problemstellungen effizient zu sein. Beispielsweise sind die vorgestellten Algorithmen leicht um einen Teilchen- oder Temperaturtransport zu ergänzen. Auch die Behandlung von Fluiden mit freien Oberflächen verlangt nach einem Newton-Krylov-Verfahren. Eine Erweiterung auf die Interaktion von Struktur mit kompressiblen Flüssigkeiten oder Strukturkontakt ist denkbar. Dabei werden die numerischen Eigenschaften der zusammengesetzten Algorithmen sicherlich von dem in dieser Arbeit vorgestellten Verhalten abweichen. Der algorithmische Aufbau wird jedoch der gleiche bleiben.

Für die Newton-Krylov-Verfahren wurde in dieser Arbeit eine ALE Formulierung verwendet, um die Deformation des Fluidgebiets zu berücksichtigen. Das zugehörige ALE Feld und die damit verbundene Beschränkung auf eine Netztopologie ist ein Schwachpunkt der vorgestellten Verfahren. Um dem abzuweichen, können auch in den Newton-Krylov-Verfahren alternative Methoden zur Beschreibung des Fluidinterfaces verwendet werden. Eine Formulierung analog zu den partitionierten Verfahren mit einem XFEM Ansatz im Fluid ist naheliegend.

Eine weitere interessante Fragestellung ist der Einsatz von Nebenbedingungen im Vorkonditionierer des Newton-Krylov-Verfahrens. Die vorgestellten Block-Vorkonditionierer sind blind gegenüber blockübergreifenden Bedingungen wie der Inkompressibilität eines vollständig eingeschlossenen Fluidgebiets. Es kann deshalb erforderlich sein, bestimmte Bedingungen im Vorkonditionierer explizit zu beachten, auch wenn die vollständige Matrix diese Bedingungen bereits enthält.

Implementierung von FSI Lösungsmethoden

Die in den Kapiteln 7 und 8 vorgestellten Lösungsverfahren wurden im Rahmen dieser Arbeit in das Finite-Elemente-Programm baci eingebaut. Baci wurde am Lehrstuhl für Numerische Mechanik neu implementiert, baut jedoch sehr stark auf den von Wall [162], Mok [117] und Förster [50] verwendeten finite Elemente Systemen auf. Das Pflegen eines einheitlichen Softwarepakets am Lehrstuhl, in das alle Lösungsmethoden integriert werden, verlangt eine erhebliche Anstrengung von allen Beteiligten. Ein guter Anteil der Arbeitszeit wird für Softwareentwicklung benötigt. Gleichzeitig sind gekoppelte Lösungsverfahren nur dann in endlicher Zeit zu erstellen, wenn auf vorhandene Implementierungen für Teilprobleme zurückgegriffen werden kann. In diesem Kapitel werden deshalb ein paar Überlegungen und Lehrbuchreferenzen zur Softwareentwicklung zusammengetragen.

A.1. Literaturhinweise

Softwareentwicklung ist Handwerk. Das Werkzeug des Programmierers ist die Abstraktion. Wichtiger noch als das handwerkliche Schreiben der Software ist die Planung und, eng damit verbunden, die Kommunikation in der Gruppe. Aus diesem Grund beschäftigen sich beispielsweise DeMarco und Lister [28] oder auch Brooks [15] mit dem Management von Programmierprojekten, wobei nicht das Programm, sondern die Programmierer im Zentrum des Interesses stehen. Eng damit verbunden sind die von Hunt und Thomas [82] diskutierten Stilfragen im Umfeld der Programmierung. Das Buch liest sich sehr angenehm und sei jedem Softwareautor ans Herz gelegt.

Die Abstraktion im Kontext einer objektorientierten Denkweise wird in aller Breite von Booch [10] besprochen. Gleichzeitig bietet Booch [10] eine der vielen Methoden an, die helfen soll, ein Entwicklungsprojekt erfolgreich anzugehen und abzuschließen. Das Buch ist sehr umfangreich und bemüht sich um eine vollständige Darstellung aller Konzepte. Zur Illustration der methodischen Darstellung werden verschiedene Software-Anwendungen diskutiert. Insgesamt enthält die Darstellung sicherlich mehr, als für gelegentliches Programmieren erforderlich ist. Die Diskussion der Softwareentwicklung als Ingenieurkunst liefert jedoch auch für andere Ingenieurgebiete bemerkenswerte Gedanken.

Einige interessante Anmerkungen zum Entwurf und Design von Softwaresystemen finden sich auch im großen C++ Lehrbuch von Stroustrup [149], welches vor allem als gründliche, wenn auch anspruchsvolle Vorstellung der Programmiersprache C++ bemerkenswert ist. Als ernsthafter Programmierer sollte man dieses Buch gelesen haben. Zum schnellen Einstieg in C++ eignen sich andere Darstellungen jedoch eher. Ein sehr empfehlenswertes Lehrbuch für die C++ Programmiersprache ist beispielsweise das Buch von Koenig und Moo [93]. Tieferes Verständnis für die Sprache liefert die Darstellung der Entwicklung in Stroustrup [148]. Sehr zu empfehlen sind weiterhin die in Meyers [112, 113] und auch Meyers [114] zusammengetragenen Einzelpunkte, die gut ausgewählte Aspekte der Sprache beleuchten. Bei spezifischen Fragen zur Programmierung mit C++ lohnt es sich oft, in Meyers [112, 113] nach einer Erklärung zu suchen. Sehr fortgeschrittene Techniken, die auf dem massiven Einsatz von Templates basieren, werden von Alexandrescu [2]

und Abrahams und Gurtovoy [1] beschrieben. Diese Techniken gehen deutlich über das normale Maß an Programmierkunst hinaus.

Sehr zu empfehlen sind darüber hinaus die Bücher von Gamma u. a. [56], Fowler [53] und Buschmann u. a. [16], die Lösungsansätze für wiederkehrende Probleme zusammentragen. Das wichtigste dieser drei ist das bahnbrechende Buch von Gamma u. a. [56]. Dieses Buch ist Pflichtlektüre für alle ambitionierten Programmierer. Auf wunderbar einfache Weise zeigen Gamma u. a. [56], wie sich mit den Ausdrucksmitteln von C++ komplexe Funktionalitäten darstellen lassen. Um einen etwas schiefen Vergleich zu bemühen: dieses Buch zeigt wie man mit der Sprache C++ Sätze formulieren und Gedanken ausdrücken kann, und vermittelt dabei eine Vorstellung, welche Gedanken überhaupt möglich sind. Das Buch von Buschmann u. a. [16] verfolgt den gleichen Ansatz, ist jedoch breiter angelegt und daher etwas schwerer zugänglich. Fowler [53] benennt ebenfalls einzelne wiederkehrende Fragestellungen und diskutiert Lösungsmöglichkeiten. Er bespricht jedoch nicht Fragen der Programmierung, sondern Fragen des Programmentwurfs.

In Fowler und Scott [55] wird eine graphische Notation vorgestellt, die für das Skizzieren von Softwareentwürfen üblich ist. Fragen des Programmierstils werden sehr gründlich von McConnell [110] betrachtet und auch in dem schon erwähnten Buch von Hunt und Thomas [82] diskutiert. In Fowler [54] werden Techniken vorgestellt, mit denen sich existierender Programmcode verbessern läßt.

A.2. Anforderungen an Software zur numerischen Simulation

Softwaresysteme wie das am Lehrstuhl für Numerische Mechanik verwendete finite Elemente System *baci* stellen ganz eigene Anforderungen. Nachdem die Simulationen nie groß genug sein können, der Bedarf an Rechenleistung und Speicherkapazität also beliebig anwachsen kann, spielt die Effizienz der Software eine große Rolle. In einigen kritischen Programmteilen, die im Zuge einer Simulation viele Male durchlaufen werden, ist die Ausführungsgeschwindigkeit entscheidend. Bei den finite Elemente Systemen sind das die Elementroutinen und Teile der linearen Gleichungslöser. An diesen Stellen können sich Millisekunden zu Stunden summieren. Der Speicherbedarf ist dagegen eine Beschränkung für die einsetzbaren Algorithmen. Hier kommt es auf die Skalierung des Speicherbedarfs mit der Problemgröße an. Im Allgemeinen sind Algorithmen, deren Speicherbedarf deutlich stärker als linear mit der Problemgröße anwächst, für große Simulationen ungeeignet.

Eine zweite Anforderung, die die Simulationssoftware mit anderen Softwaresystemen teilt, ist die gewünschte Einfachheit. Das Programm muß sich in endlicher Zeit schreiben und—ganz wichtig—lesen lassen. Um das zu erreichen, soll die Implementierung der Lösungsverfahren die aus der algorithmischen Darstellung bekannten Begriffen verwenden. Das ist der Sinn der Abstraktion, wie sie von Booch [10] beschrieben wird. Unglücklicherweise steht die Abstraktion der Effizienz im Wege, da jede neue Abstraktionsebene zusätzlichen Aufwand bedeutet.

Als eine Lösung für dieses Problem werden von der Sprache C++ *template*-Mechanismen angeboten. Damit steht eine Möglichkeit zur Verfügung, die Konzepte des Problemereichs, also Matrizen, Vektoren usw., als Bausteine für die Software zu verwenden. Die Effizienz wird dabei sichergestellt, indem die so eingeführten Abstraktionen beim Übersetzen des Programms aufgelöst werden. Somit entsteht erheblicher Aufwand beim Übersetzen und entsprechend erhebliche Übersetzungszeiten. Darüber hinaus sind diese Techniken durchaus komplex, erfordern also einigen Lernaufwand, um wirklich beherrscht zu werden. Auch die Fehlersuche im Programm wird durch den Einsatz von *templates* deutlich erschwert.

Glücklicherweise sind viele Programmteile im Hinblick auf die Rechengeschwindigkeit jedoch vergleichsweise unkritisch, so daß die Forderung nach Einfachheit dominiert und die

üblichen Abstraktionstechniken, siehe Gamma u. a. [56], bedenkenlos eingesetzt werden können.

A.3. Zur Architektur der FSI Algorithmen in baci

An dieser Stelle soll die Architektur von baci nicht im Detail vorgestellt werden, was ein müßiges und schnell veraltetes Unterfangen wäre. Um baci zu verstehen, kommt man nicht umhin, den Quellcode zu lesen. Es sollen jedoch einige wenige Verweise auf die wesentlichen Klassen der FSI Algorithmen gegeben werden, um den Einstieg in den Code zu erleichtern.

Eine wesentliche Rolle in baci spielen die parallelen Objekte und Lösungsverfahren aus den Bibliotheken des trilinos Projekts. Die gesamte Parallelisierung wird mit diesen Objekten organisiert, so daß die parallele Bearbeitung in die Algorithmen integriert ist. Ein Überblick über das trilinos Projekt wird von Heroux u. a. [76] gegeben. Siehe Sala u. a. [134] für eine Einführung in trilinos. Die Parallelisierung wurde bereits in Kapitel 2 angesprochen.

Die zentrale Klasse in baci, die von allen Algorithmen verwendet wird, ist die Klasse `DRT::Discretization`. Diese Klasse stellt ein über mehrere Prozessoren verteiltes Finite-Elemente-Netz dar. Die Knoten, Elemente und Randbedingungen werden hier verwaltet. Auch das Management der Freiheitsgrade läuft über die Klasse `DRT::Discretization`.

Die Algorithmen, die auf der Klasse `DRT::Discretization` aufbauen und im Kern die Routinen zur Abarbeitung der Algorithmen 8 oder 10 bereitstellen, werden ebenfalls in jeweils einer Klasse organisiert. Beispiele für Algorithmenklassen sind die Klassen `StruGenAlpha` und `FLD::FluidImplicitTimeInt`, es existieren jedoch mehrere durchaus verschiedene Implementierungen für Struktur-, Fluid- und ALE-Algorithmen in baci. Der Grund dafür sind einerseits die unterschiedlichen Anforderungen bei verschiedenen Anwendungsgebieten, denen mit unterschiedlichen Lösungsstrategien begegnet wird, andererseits aber auch die unterschiedlichen Herangehensweisen der einzelnen Autoren von baci. Die Kommunikationswege am Lehrstuhl hinterlassen deutliche Spuren im Quelltext. Eine abschließende Liste aller Algorithmenklassen kann bei einem lebendigen Softwaresystem wie baci nicht angegeben werden.

Die Aufgabe der einfachen Algorithmenklassen ist es, die Implementierung für einen Lösungsalgorithmus bereit zu stellen. Die Lösungsalgorithmen in Klassen zu verpacken hat dabei den Vorteil, die Algorithmen später einfacher miteinander kombinieren zu können. Für sich speichert die Algorithmenklasse alle Vektoren und Bandmatrizen, die innerhalb der Zeitschleife vom Newton-Verfahren und den linearen Lösungsverfahren benötigt werden.

Neben den einfachen Algorithmenklassen existieren Klassen für die zusammengesetzten FSI Algorithmen. Die in Kapitel 7 besprochenen Lösungsverfahren für das Dirichlet-Neumann partitionierte FSI Problem werden von der Klasse `FSI::DirichletNeumann` zur Verfügung gestellt. Die Klasse `FSI::MonolithicStructureSplit` implementiert das lineare System (8.2.1), auf dem sowohl das Block-Gauß-Seidel-Verfahren aus Kapitel 8.2 als auch das vorkonditionierte Krylov-Verfahren aus den Kapiteln 8.3 und 8.4 basiert.

Die Klassen für die zusammengesetzten FSI Algorithmen implementieren jeweils ihre eigene Zeitschleife und übernehmen somit die Steuerung des jeweiligen FSI Lösungsalgorithmus. Für das Behandeln der einzelnen Felder wird jedoch auf die Klassen für die Einzelalgorithmen zurückgegriffen. Um verschiedene Implementierungen der Einzelfelder in den FSI Algorithmen einsetzen zu können, werden spezielle Adapter-Klassen zwischen die FSI Algorithmen und die einzelnen Feldalgorithmen geschaltet. Diese Adapter stellen den

FSI Algorithmen eine einheitliche Schnittstelle zur Verfügung, hinter der die Unterschiede der verschiedenen Feldalgorithmen verborgen werden.

Alle FSI Algorithmen verwenden die NOX Bibliothek aus trilinos zum Lösen der nichtlinearen Gleichungssysteme.

Neben den großen Algorithmen-Klassen spielen zwei Hilfsklassen eine entscheidende Rolle. Die erste ist die Klasse `LINALG::MultiMapExtractor` und, davon abgeleitet, die Klasse `LINALG::MapExtractor`. Mit dieser Klasse wird das Zerlegen eines großen `Epetra_Map`-Objekts in mehrere nichtüberlappende `Epetra_Map`-Objekte beschrieben. Diese Zerlegung wird zum Beispiel verwendet, um den Vektor der Verschiebungen $\mathbf{d}^{S,n+1}$ in die Verschiebungen im Inneren des Gebiets $\mathbf{d}_I^{S,n+1}$ und die Verschiebungen am Rand $\mathbf{d}_F^{S,n+1}$ zu zerlegen. Auch für das Formulieren des Block-Systems (8.2.1) wird diese Klasse herangezogen.

Die zweite Hilfsklasse, der eine große Bedeutung zukommt, ist `ADAPTER::Coupling`. Diese Klasse koppelt die Freiheitsgrade von ausgewählten Knoten aus zwei Netzen aneinander. Damit wird die Kopplung der Netze am FSI Interface erzeugt, d.h. Objekte der Klasse `ADAPTER::Coupling` sind dafür zuständig, Interfacevektoren zwischen den Struktur-, Fluid- und ALE-Diskretisierungen auszutauschen.

Literaturverzeichnis

- [1] ABRAHAMSON, D. ; GURTOVOY, A.: *C++ Template Metaprogramming*. Addison-Wesley, 2005
- [2] ALEXANDRESCU, A.: *Modern C++ Design*. Addison-Wesley, 2001
- [3] BADIA, A. ; NOBILE, F. ; VERGARA, C.: Fluid-Structure Partitioned Procedures based on Robin Transmission Conditions. In: *J. Comput. Physics* 227 (2008), Nr. 14, S. 7027–7051
- [4] BADIA, S. ; CODINA, R.: On Some Fluid-Structure Iterative Algorithms using Pressure Segregation Methods. Application to Aeroelasticity. In: *Int. J. Numer. Meth. Engng.* 72 (2007), Nr. 1, S. 46–71
- [5] BADIA, S. ; QUAINI, A. ; QUARTERONI, A.: Modular vs. Non-Modular Preconditioners for Fluid-Structure Systems with Large Added-Mass Effect. In: *Comp. Meth. in Appl. Mech. and Engng.* 197 (2008), Nr. 49–50, S. 4216–4232
- [6] BARRETT, R. ; BERRY, M. ; CHAN, T. F. ; DEMMEL, J. ; DONATO, J. ; DONGARRA, J. ; EIJKHOUT, V. ; POZO, R. ; ROMINE, C. ; VORST, H. van der: *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Philadelphia, PA : SIAM, 1994
- [7] BAZILEVS, Y. ; CALO, V.M. ; ZHANG, Y. ; HUGHES, T.J.R.: Isogeometric Fluid-Structure Interaction Analysis with Applications to Arterial Blood Flow. In: *Comput. Mech.* 38 (2006), Nr. 4–5, S. 310–322
- [8] BENZI, M.: Preconditioning Techniques for Large Linear Systems: A Survey. In: *J. Comput. Physics* 182 (2002), S. 418–477
- [9] BIRKS, J. B.: *Rutherford at Manchester*. W. A. Benjamin, 1963
- [10] BOOCH, G.: *Objektorientierte Analyse und Design*. Addison-Wesley, 1994
- [11] BRAESS, H. ; WRIGGERS, P.: Arbitrary Lagrangian Eulerian Finite Element Analysis of Free Surface Flow. In: *Comp. Meth. in Appl. Mech. and Engng.* 190 (2000), S. 95–109
- [12] BREZINSKI, C.: Convergence Acceleration during the 20th Century. In: *J. Comp. and Appl. Math.* 122 (2000), S. 1–21
- [13] BREZINSKI, C. ; REDIVO ZAGLIA, M.: *Extrapolation Methods: Theory and Practice*. Amsterdam : North-Holland, 1991
- [14] BRIGGS, William L. ; HENSON, Van E. ; MCCORMICK, Steve: *A Multigrid Tutorial*. Philadelphia : SIAM, 2000
- [15] BROOKS, F. P.: *The Mythical Man Month*. Addison-Wesley, 1995
- [16] BUSCHMANN, F. ; R., Meunier ; ROHNERT, H. ; SOMMERLAD, P. ; STAL, M.: *Pattern-orientierte Software-Architektur*. Addison-Wesley, 1998
- [17] CAUSIN, P. ; GERBEAU, J.-F. ; NOBILE, F.: Added-Mass Effect in the Design of Partitioned Algorithms for Fluid-Structure Problems. In: *Comp. Meth. in Appl. Mech. and Engng.* 194 (2005), S. 4506–4527
- [18] CHALOT, F. L.: Industrial Aerodynamics. In: STEIN, E. (Hrsg.) ; BORST, R. D. (Hrsg.) ; HUGHES, T.J.R. (Hrsg.): *Encyclopedia of Computational Mechanics* Bd. 3. John Wiley & Sons, 2004, Kap. 12, S. 407–458
- [19] CHORIN, A.J.: Numerical Method for Solving Incompressible Viscous Flow Problems. In: *J. Comput. Physics* 2 (1967), Nr. 1, S. 12–26

- [20] CHRISTON, M.A.: A Domain-Decomposition Message-Passing Approach to Transient Viscous Incompressible Flow Using Explicit Time Integration. In: *Comp. Meth. in Appl. Mech. and Engng.* 148 (1997), S. 329–352
- [21] CHUNG, J. ; HULBERT, G.M.: A Time Integration Algorithm for Structural Dynamics with Improved Numerical Dissipation: The Generalized- α Method. In: *J. Appl. Math.* 60 (1993), S. 371–375
- [22] CODINA, R. ; SOTO, O.: Approximation of the Incompressible Navier-Stokes Equations Using Orthogonal Subscale Stabilization and Pressure Segregation on Anisotropic Finite Element Meshes. In: *Comp. Meth. in Appl. Mech. and Engng.* 193 (2004), S. 1403–1419
- [23] CODINA, S.: On Some Pressure Segregation Methods of Fractional-Step Type for the Finite Element Approximation of Incompressible Flow Problems. In: *Comp. Meth. in Appl. Mech. and Engng.* 195 (2006), S. 2900–2918
- [24] CRISFIELD, M.A.: *Non-linear Finite Element Analysis of Solids and Structures.* Bd. 2. Wiley, 1997
- [25] CRISFIELD, M.A.: *Non-linear Finite Element Analysis of Solids and Structures.* Bd. 1. Wiley, 1998
- [26] DEGAND, C. ; FARHAT, C.: A Three-Dimensional Torsional Spring Analogy Method for Unstructured Dynamic Meshes. In: *Comput. Struct.* 80 (2002), S. 305–316
- [27] DEGROOTE, J. ; BRUGGEMAN, P. ; HAELTERMAN, R. ; VIERENDEELS, J.: Stability of a Coupling Technique for Partitioned Solvers in FSI Applications. In: *Comput. Struct.* (2008)
- [28] DEMARCO, T. ; LISTER, T.: *Peopleware: Productive Projects and Teams.* Dorset House, 1999
- [29] DEPARIS, S.: *Numerical Analysis of Axisymmetric Flows and Methods for Fluid–Structure Interaction Arising in Blood Flow Simulation*, EPFL, Dissertation, 2004
- [30] DEPARIS, S. ; DISCACCIATI, M. ; FOURESTEY, G. ; QUARTERONI, A.: Fluid–Structure Algorithms based on Steklov-Poincaré Operators. In: *Comp. Meth. in Appl. Mech. and Engng.* 195 (2006), S. 5797–5812
- [31] DEPARIS, Simone ; GERBEAU, Jean-Frédéric ; VASSEUR, Xavier: A Dynamic Preconditioner for Newton-Krylov Algorithms. Application to Fluid–Structure Interaction. 2004. – Forschungsbericht. Inria Report RR-5352 <http://www.inria.fr/rrrt/rr-5352.html>
- [32] DETTMER, W.G. ; PERIĆ, D.: A Computational Framework for Fluid–Structure Interaction: Finite Element Formulation and Applications. In: *Comp. Meth. in Appl. Mech. and Engng.* 195 (2006), S. 5754–5779
- [33] DETTMER, W.G. ; PERIĆ, D.: On the Coupling Between Fluid Flow and Mesh Motion in the Modelling of Fluid–Structure Interaction. In: *Comput. Mech.* 43 (2008), S. 81–90
- [34] DEUFLHARD, P.: *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms.* Springer, Berlin, 2004
- [35] DONEA, J. ; HUERTA, A. ; WILEY (Hrsg.): *Finite Element Methods for Flow Problems.* John Wiley & Sons, Inc., 2003
- [36] DONEA, J. ; HUERTA, A. ; PONTHOT, J.-Ph. ; RODRÍGUEZ-FERRAN, A.: Arbitrary Lagrangian–Eulerian Methods. In: STEIN, E. (Hrsg.) ; BORST, R. D. (Hrsg.) ; HUGHES, T.J.R. (Hrsg.): *Encyclopedia of Computational Mechanics* Bd. 1. John Wiley & Sons, 2004, Kap. 14, S. 414–437
- [37] DUARTE, F. ; GORMAZ, R. ; NATESAN, S.: Arbitrary Lagrangian–Eulerian Method for Navier–Stokes Equations with Moving Boundaries. In: *Comp. Meth. in Appl. Mech. and Engng.* 193 (2004), S. 4819–4836
- [38] ELMAN, H. ; SILVESTER, D. ; WATHEN, A.: *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics.* Oxford University

- Press, 2005
- [39] ELMAN, H. C. ; HOWLE, V. E. ; SHADID, J. N. ; SHUTTLEWORTH, R. ; TUMINARO, R. S.: A Taxonomy and Comparison of Parallel Block Multi-Level Preconditioners for the Incompressible Navier-Stokes Equations. In: *J. Comput. Physics* 227 (2008), S. 1790–1808
- [40] ELMAN, H. C. ; HOWLE, V. E. ; SHADID, J. N. ; SILVESTER, D. ; TUMINARO, R. S.: Least Squares Preconditioners for Stabilized Discretizations of the Navier-Stokes Equations. In: *SIAM. J. Sci. Comput.* 30 (2007), S. 290–311
- [41] ELMAN, H. C. ; HOWLE, V. E. ; SHADID, J. N. ; TUMINARO, R. S.: A Parallel Block Multi-level Preconditioner for the 3D Incompressible Navier-Stokes Equations. In: *J. Comput. Physics* 187 (2003), S. 504–523
- [42] ESHOF, Jasper van den ; SLEIJPEN, Gerard L. ; GIJZEN, Martin B. van: Relaxation Strategies for Nested Krylov Methods. In: *J. Comp. and Appl. Math.* 177 (2005), S. 347–365
- [43] FAIRES, J.D. ; BURDEN, R.L.: *Numerische Methoden - Näherungsverfahren und ihre praktische Anwendung*. Spektrum Akademischer Verlag, 2000
- [44] FALGOUT, R.D.: An Introduction to Algebraic Multigrid. In: *Computing in Science and Engng.* 8 (2006), Nr. 6, S. 24–33. – ISSN 1521-9615
- [45] FARHAT, C.: CFD-based Nonlinear Computational Aeroelasticity. In: STEIN, E. (Hrsg.) ; BORST, R. D. (Hrsg.) ; HUGHES, T.J.R. (Hrsg.): *Encyclopedia of Computational Mechanics* Bd. 3. John Wiley & Sons, 2004, Kap. 13, S. 459–480
- [46] FARHAT, C. ; DEGAND, C. ; KOOBUS, B. ; LESOINNE, M.: Torsional Springs for Two-Dimensional Dynamic Unstructured Fluid Meshes. In: *Comp. Meth. in Appl. Mech. and Engng.* 163 (1998), S. 231–245
- [47] FENG, Z. ; SOULAÏMANI, A. ; SAAD, Y.: Nonlinear Krylov Acceleration for CFD-based Aeroelasticity. In: *J. Fluids and Struct.* (in press)
- [48] FERNÁNDEZ, M.Á. ; MOUBACHIR, M.: A Newton Method Using Exact Jacobians for Solving Fluid–Structure Coupling. In: *Comput. Struct.* 83 (2005), Nr. 2–3, S. 127–142
- [49] FERNÁNDEZ, M.A. ; GERBEAU, J.-F. ; GLORIA, A. ; VIDRASCU, M.: A Partitioned Newton Method for the Interaction of a Fluid and a 3D Shell Structure. In: *INRIA* (2008)
- [50] FÖRSTER, Ch.: *Robust Methods for Fluid–Structure Interaction with Stabilised Finite Elements*, Institut für Baustatik und Baudynamik Universität Stuttgart, Dissertation, 2007
- [51] FÖRSTER, Ch. ; WALL, W. A. ; RAMM, E.: On the Geometric Conservation Law in Transient Flow Calculations on Deforming Domains. In: *Int. J. Numer. Meth. Fluids* 50 (2005), S. 1369–1379
- [52] FÖRSTER, Ch. ; WALL, W. A. ; RAMM, E.: Artificial Added Mass Instabilities in Sequential Staggered Coupling of Nonlinear Structures and Incompressible Viscous Flows. In: *Comp. Meth. in Appl. Mech. and Engng.* 196 (2007), S. 1278–1293
- [53] FOWLER, M.: *Analysis Patterns: Reusable Object Models*. Addison-Wesley, 1996
- [54] FOWLER, M.: *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, 1999
- [55] FOWLER, M. ; SCOTT, K.: *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, 1999
- [56] GAMMA, E. ; HELM, R. ; JOHNSON, R. ; VLISSIDES, J.: *Entwurfsmuster*. Addison-Wesley, 1996
- [57] GEE, M.: *Effiziente Lösungsstrategien in der nichtlinearen Schalenmechanik*, Institut für Baustatik, Universität Stuttgart, Dissertation, 2004
- [58] GEE, M.W. ; HU, J.J. ; TUMINARO, R.S.: A New Smoothed Aggregation Multigrid Method for Anisotropic Problems. In: *Numer. Linear Algebra Appl.* 16 (2009),

- S. 19–37
- [59] GEE, M.W. ; KÜTTLER, U. ; WALL, W.A.: Truly Monolithic Algebraic Multigrid for Fluid–Structure Interaction. (in preparation)
- [60] GERBEAU, J.-F. ; VIDRASCU, M.: A Quasi-Newton Algorithm based on a Reduced Model for Fluid–Structure Interaction Problems in Blood Flows. In: *Math. Model. and Numer. Anal.* 37 (2003), Nr. 4, S. 631–647
- [61] GERBEAU, J.-F. ; VIDRASCU, M. ; FREY, P.: Fluid–Structure Interaction in Blood Flows on Geometries coming from Medical Imaging. In: *Comput. Struct.* 83 (2005), S. 155–165
- [62] GERSTENBERGER, A. ; WALL, W.A.: Enhancement of Fixed-Grid Methods towards Complex Fluid–Structure Interaction Applications. In: *Int. J. Numer. Meth. Fluids* 57 (2008), S. 1227–1248
- [63] GERSTENBERGER, A. ; WALL, W.A.: An eXtended Finite Element Method / Lagrange Multiplier based Approach for Fluid–Structure Interaction. In: *Comp. Meth. in Appl. Mech. and Engng.* 197 (2008), S. 1699–1714
- [64] GOLUB, G. H. ; VAN LOAN, C. F.: *Matrix Computations*. The Johns Hopkins University Press, 1996
- [65] GRESHO, P. M. ; SANI, R. L. ; WILEY (Hrsg.): *Incompressible Flow and the Finite Element Method, Isothermal Laminar Flow*. Bd. 2. John Wiley & Sons, Inc., 2000
- [66] GRESHO, P. M. ; STEVENS, T. C.: On the Theory of Semi-Implicit Projection Methods for Viscous Incompressible Flow and Its Implementation via a Finite Element Method that also Introduces a Nearly Consistent Mass Matrix, Part 2: Implementation. In: *Int. J. Numer. Meth. Fluids* 11 (1990), S. 621–659
- [67] GRESHO, P.M. ; SANI, R.L.: *Incompressible Flow and the Finite Element Method Advection-Diffusion*. Bd. 1. Wiley, 2000
- [68] GURTIN, M.E.: *An Introduction to Continuum Mechanics*. Academic Press, 1981
- [69] HACKBUSCH, W.: *Computational Mathematics*. Bd. 4: *Multigrid Methods and Applications*. Berlin : Springer-Verlag, 1985
- [70] HAMMING, R.W.: The Unreasonable Effectiveness of Mathematics. In: *The American Mathematical Monthly* 87 (1980), Nr. 2, S. 81–90
- [71] HAMMING, R.W.: *Numerical Methods for Scientists and Engineers*. Dover Publication, 1987
- [72] HAMMING, R.W.: *The Art of Doing Science and Engineering*. Taylor & Francis Ltd, 1997
- [73] HEIL, M.: An Efficient Solver for the Fully Coupled Solution of Large-Displacement Fluid–Structure Interaction Problems. In: *Comp. Meth. in Appl. Mech. and Engng.* 193 (2004), S. 1–23
- [74] HEIL, M. ; HAZEL, A.L. ; BOYLE, J.: Solvers for Large-Displacement Fluid–Structure Interaction Problems: Segregated Versus Monolithic Approaches. In: *Comput. Mech.* 43 (2008), Nr. 1, S. 91–101
- [75] HENKE, F.: *Untersuchung von Projektionsalgorithmen für FSI-Probleme mit nicht-passenden Netzen*, Lehrstuhl für Numerische Mechanik, TU München, Semesterarbeit, 2006
- [76] HEROUX, M.A. ; BARTLETT, R.A. ; HOWLE, V.E. ; HOEKSTRA, R.J. ; HU, J.J. ; KOLDA, T.G. ; LEHOUCQ, R.B. ; LONG, K.R. ; PAWLOWSKI, R.P. ; PHIPPS, E.T. ; SALINGER, A.G. ; THORNQUIST, H.K. ; TUMINARO, R.S. ; WILLENBRING, J.M. ; WILLIAMS, A. ; STANLEY, K.S.: An Overview of the Trilinos Project. In: *ACM Trans. Math. Softw.* 31 (2005), Nr. 3, S. 397–423. – ISSN 0098-3500
- [77] HÜBNER, B. ; DINKLER, D.: A Simultaneous Solution Procedure for Strong Interactions of Generalized Newtonian Fluids and Viscoelastic Solids at Large Strains. In: *Int. J. Numer. Meth. Engng.* 64 (2005), Nr. 7, S. 920–939

- [78] HÜBNER, B. ; WALHORN, E. ; DINKLER, D.: A Monolithic Approach to Fluid–Structure Interaction using Space-Time Finite Elements. In: *Comp. Meth. in Appl. Mech. and Engng.* 193 (2004), S. 2087–2104
- [79] HUGHES, T.J.R.: *The Finite Element Method*. Dover publications, 2000
- [80] HUGHES, T.J.R. ; FRANCA, L.P. ; HULBERT, G.M.: A new Finite Element Formulation for Computational Fluid Dynamics: VIII. The Galerkin/Least-Squares Method for Advective-Diffusive Equations. In: *Comp. Meth. in Appl. Mech. and Engng.* 73 (1989), Nr. 2, S. 173–189
- [81] HUMPHREY, J.D. ; TAYLOR, C.A.: Intracranial and Abdominal Aortic Aneurysms: Similarities, Differences, and Need for a New Class of Computational Models. In: *Annu. Rev. Biomed. Eng.* 10 (2008), S. 221–246
- [82] HUNT, A. ; THOMAS, D.: *The Pragmatic Programmer*. Addison-Wesley, 1999
- [83] IRONS, B. ; TUCK, R.C.: A Version of the Aitken Accelerator for Computer Implementation. In: *Int. J. Numer. Meth. Engng.* 1 (1969), S. 275–277
- [84] JAMESON, A.: Aerodynamics. In: STEIN, E. (Hrsg.) ; BORST, R. D. (Hrsg.) ; HUGHES, T.J.R. (Hrsg.): *Encyclopedia of Computational Mechanics* Bd. 3. John Wiley & Sons, 2004, Kap. 11, S. 325–406
- [85] JBILOU, K. ; SADOK, H.: LU Implementation of the Modified Minimal Polynomial Extrapolation Method for Solving Linear and Nonlinear Systems. In: *IMA J. Numer. Anal.* 19 (1999), Nr. 4, S. 549–561
- [86] JBILOU, K. ; SADOK, H.: Vector Extrapolation Methods. Applications and Numerical Comparison. In: *J. Comp. and Appl. Math.* 122 (2000), S. 149–165
- [87] JOOSTEN, M.M. ; DETTMER, W.G. ; PERIC, D.: Analysis of the Block Gauss–Seidel Solution Procedure for a Strongly Coupled Model Problem with Reference to Fluid–Structure Interaction. In: *Int. J. Numer. Meth. Engng.* 78 (2009), S. 757–778
- [88] KELLEY, C.T.: *Iterative Methods for Linear and Nonlinear Equations*. SIAM, 1995
- [89] KELLEY, C.T.: *Solving Nonlinear Equations with Newton’s Method*. SIAM, 2003
- [90] KERKHOVEN, T. ; SAAD, Y.: On Acceleration Methods for Coupled Nonlinear Elliptic Systems. In: *Numer Math* 60 (1992), S. 525–548
- [91] KNOLL, D. A. ; MOUSSEAU, V. A. ; CHACÓN, L. ; REISNER, J.: Jacobian-Free Newton-Krylov Methods for the Accurate Time Integration of Stiff Wave Systems. In: *J. Sci. Comput.* 25 (2005), S. 213–230
- [92] KNOLL, D.A. ; KEYES, D.E.: Jacobian-free Newton-Krylov Methods: A Survey of Approaches and Applications. In: *J. Comput. Physics* 193 (2004), S. 357–397
- [93] KOENIG, A. ; MOO, B. E.: *Accelerated C++*. Addison-Wesley, 2000
- [94] KUHL, D. ; CRISFIELD, M.A.: Energy-Conserving and Decaying Algorithms in Non-linear Structural Dynamics. In: *Int. J. Numer. Meth. Engng.* 45 (1999), S. 569–599
- [95] KUHL, D. ; RAMM, E.: Time Integration in the Context of Energy Control and Locking Free Finite Elements. In: *Arch. Comput. Meth. Engng.* 7 (2000), Nr. 3, S. 299–332
- [96] KÜTTLER, U. ; FÖRSTER, Ch. ; WALL, W. A.: A Solution for the Incompressibility Dilemma in Partitioned Fluid–Structure Interaction with Pure Dirichlet Fluid Domains. In: *Comput. Mech.* 38 (2006), S. 417–429
- [97] KÜTTLER, U. ; GEE, M. ; FÖRSTER, Ch. ; COMERFORD, A. ; WALL, W.A.: Coupling Strategies for Biomedical Fluid–Structure Interaction Problems. In: *Commun. Numer. Meth. Engng* (accepted, 2009)
- [98] KÜTTLER, U. ; WALL, W. A.: Fixed-Point Fluid–Structure Interaction Solvers with Dynamic Relaxation. In: *Comput. Mech.* 43 (2008), Nr. 1, S. 61–72
- [99] KÜTTLER, U. ; WALL, W. A.: Vector Extrapolation for Strong Coupling Fluid–Structure Interaction Solvers. In: *J. Appl. Mech.* 76 (2009)

- [100] KÜTTLER, U. ; WALL, W.A.: The Dilemma of Domain Decomposition Approaches in Fluid-Structure Interactions with Fully Enclosed Incompressible Fluids. In: LANGER, U. (Hrsg.) ; DISCACCIATI, M. (Hrsg.) ; KEYES, D.E. (Hrsg.) ; WIDLUND, O.B. (Hrsg.) ; ZULEHNER, W. (Hrsg.): *Domain Decomposition Methods in Science and Engineering XVII* Bd. 60. Heidelberg : Springer, 2008, S. 155–164. – Proceedings of the 17th International Conference on Domain Decomposition Methods held at St. Wolfgang / Strobl, Austria, July 3–7, 2006
- [101] KRÍŽEK, M. ; LIU, L. ; NEITTAANMÄKI, P.: Post-Processing of Gauss–Seidel Iterations. In: *Numer. Linear Algebra Appl.* 6 (1999), S. 147–156
- [102] LANCZOS, C.: *The Variational Principles of Mechanics*. Dover publications, 1986
- [103] LE TALLEC, P. ; MOURO, J.: Fluid Structure Interaction with Large Structural Displacements. In: *Comput. Methods Appl. Mech. Engrg.* 190 (2001), Nr. 24–25, S. 3039–3067
- [104] MACLEOD, A.J.: Acceleration of Vector Sequences by Multi-Dimensional Δ^2 Methods. In: *Comm. in Appl. Num. Meth.* 2 (1986), S. 385–392
- [105] MAIER, A.: *Partitionierte Lösungsverfahren für Fluid-Struktur-Interaktion mit Robin-Randbedingungen*, Lehrstuhl für Numerische Mechanik, TU München, Semesterarbeit, 2008
- [106] MARSDEN, J.E. ; HUGHES, T.J.R.: *Mathematical Foundations of Elasticity*. Dover publications, 1994
- [107] MATTHIES, H.G. ; NIEKAMP, R. ; STEINDORF, J.: Algorithms for Strong Coupling Procedures. In: *Comp. Meth. in Appl. Mech. and Engng.* 195 (2008), S. 2028–2049
- [108] MATTHIES, H.G. ; STEINDORF, J.: Partitioned but Strongly Coupled Iteration Schemes for Nonlinear Fluid–Structure Interaction. In: *Comput. Struct.* 80 (2002), S. 1991–1999
- [109] MATTHIES, H.G. ; STEINDORF, J.: Partitioned Strong Coupling Algorithms for Fluid–Structure Interaction. In: *Comput. Struct.* 81 (2003), S. 805–812
- [110] MCCONNELL, S.: *Code Complete*. Microsoft Press, 1993
- [111] MEISTER, A.: *Numerik linearer Gleichungssysteme*. Vieweg, 1999
- [112] MEYERS, S.: *More Effective C++*. Addison-Wesley, 1995
- [113] MEYERS, S.: *Effective C++*. Addison-Wesley, 1998
- [114] MEYERS, S.: *Effective STL*. Addison-Wesley, 2001
- [115] MICHLER, C. ; BRUMMELEN, E.H. van ; BORST, R. de: An Interface Newton-Krylov Solver for Fluid–Structure Interaction. In: *Int. J. Numer. Meth. Fluids* 47 (2005), S. 1189–1195
- [116] MICHLER, C. ; BRUMMELEN, E.H. van ; BORST, R. de: Error-Amplification Analysis of Subiteration-Preconditioned GMRES for Fluid–Structure Interaction. In: *Comp. Meth. in Appl. Mech. and Engng.* 195 (2006), S. 2124–2148
- [117] MOK, D.P.: *Partitionierte Lösungsansätze in der Strukturdynamik und der Fluid-Struktur-Interaktion*, Institut für Baustatik Universität Stuttgart, Dissertation, 2001
- [118] MOK, D.P. ; WALL, W. A.: Partitioned Analysis Schemes for the Transient Interaction of Incompressible Flows and Nonlinear Flexible Structures. In: *Trends in Computational Structural Mechanics*, W.A. Wall, K.-U. Bletzinger and K. Schweitzerhof (Eds.), 2001
- [119] NOBILE, F. ; VERGARA, C.: An Effective Fluid–Structure Interaction Formulation for Vascular Dynamics by Generalized Robin Conditions. In: *SIAM. J. Sci. Comput.* 30 (2008), Nr. 2, S. 731–763
- [120] PARK, K.C. ; FELIPPA, C.A. ; OHAYON, R.: Partitioned Formulation of Internal Fluid–Structure Interaction Problems by Localized Lagrange Multipliers. In: *Comp. Meth. in Appl. Mech. and Engng.* 190 (2001), Nr. 24–25, S. 2989–3007
- [121] PARKS, M.L. ; STURLER, E. de ; MACKAY, G. ; JOHNSON, D.D. ; MAITI, S.: Recycling Krylov Subspaces for Sequences of Linear Systems. In: *SIAM. J. Sci.*

- Comput.* 28 (2006), Nr. 5, S. 1651–1674
- [122] POPPER, K.R.: *Logik der Forschung*. 10. verb. u. verm. A. Mohr Siebeck, 2005
- [123] PRESS, W.H. ; TEUKOLSKY, S.A. ; VETTERLING, W.T. ; FLANNERY, B.P.: *Numerical Recipes in C, the Art of Scientific Computing*. Cambridge University Press, 1993
- [124] QUARTERONI, A. ; SACCO, R. ; SALERI, F.: *Numerical Mathematics*. Springer, 2000
- [125] QUARTERONI, A. ; SALERI, F. ; VENEZIANI, A.: Factorization Methods for the Numerical Approximation of Navier-Stokes Equations. In: *Comp. Meth. in Appl. Mech. and Engng.* 188 (2000), S. 505–526
- [126] RABACK, P. ; RUOKOLAINEN, J. ; LYLÄ, M. ; JÄRVINEN, E.: Fluid–Structure Interaction Boundary Conditions by Artificial Compressibility. In: *Computational Fluid Dynamics Conference ECCOMAS (Veranst.)*, 2001
- [127] RANNACHER, R.: Incompressible Viscous Flows. In: STEIN, E. (Hrsg.) ; BORST, R. D. (Hrsg.) ; HUGHES, T.J.R. (Hrsg.): *Encyclopedia of Computational Mechanics* Bd. 3. John Wiley & Sons, 2004, Kap. 6, S. 155–181
- [128] RHEINBOLDT, W.C.: *Methods for Solving Systems of Nonlinear Equations*. Siam, 1998
- [129] RIEMSLAGH, K. ; VIERENDEELS, J. ; DICK, E.: An Efficient Coupling Procedure for Flexible Wall Fluid–Structure Interaction. In: *European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS (Veranst.)*, 2000
- [130] SAAD, Y.: A Flexible Inner-Outer Preconditioned GMRES Algorithm. In: *SIAM J. Sci. Comput.* 14 (1993), Nr. 2, S. 461–469
- [131] SAAD, Y.: *Iterative Methods for Sparse Linear Systems*. SIAM, 2003
- [132] SAAD, Y. ; SCHULTZ, M.H.: GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. In: *SIAM J. Sci. Stat. Comput.* 7 (1986), Nr. 3, S. 856–869
- [133] SAAD, Y. ; VORST, H.A. van der: Iterative Solution of Linear Systems in the 20th Century. In: *J. Comp. and Appl. Math.* 123 (2000), S. 1–33
- [134] SALA, M. ; HEROUX, M. A. ; (EDITORS), D. M. D.: Trilinos Tutorial / Sandia National Laboratories. 2007 (SAND-2189). – Forschungsbericht
- [135] SALA, M. ; TUMINARO, R.S.: A New Petrov-Galerkin Smoothed Aggregation Preconditioner for Nonsymmetric Linear Systems. In: *Siam J. Scientific Comp.* 31 (2008), S. 143–166
- [136] SCHILPP, P.A. (Hrsg.): *Albert Einstein als Philosoph und Naturforscher*. Kohlhammer, 1949
- [137] SHEWCHUK, J.R.: *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. 1994. – <http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf>
- [138] SIDI, A.: Efficient Implementation of Minimal Polynomial and Reduced Rank Extrapolation Methods. In: *J. Comp. and Appl. Math.* 36 (1991), Nr. 3, S. 305–337
- [139] SIMONCINI, V. ; SZYLD, D.B.: Flexible Inner-Outer Krylov Subspace Methods. In: *SIAM J. Numer. Anal.* 40 (2002), Nr. 6, S. 2219–2239
- [140] SIMONCINI, V. ; SZYLD, D.B.: Recent Computational Developments in Krylov Subspace Methods for Linear Systems. In: *Numer. Linear Algebra Appl.* 14 (2007), S. 1–59
- [141] SMITH, B. ; GROPP, W. ; BJORSTAD, P.: *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996
- [142] SMITH, D.A. ; FORD, W.F. ; SIDI, A.: Extrapolation Methods for Vector Sequences. In: *SIAM Review* 29 (1987), Nr. 2, S. 199–233

- [143] SMITH, D.A. ; FORD, W.F. ; SIDI, A.: Erratum: Correction to “Extrapolation Methods for Vector Sequences”. In: *SIAM Review* 30 (1988), Nr. 4, S. 623–624
- [144] SOTO, O. ; LÖHNER, R. ; CEBRAL, J. ; CAMELLI, F.: A Stabilized Edge-based Implicit Incompressible Flow Formulation. In: *Comp. Meth. in Appl. Mech. and Engng.* 193 (2004), S. 2139–2154
- [145] STERNEL, D. C. ; SCHÄFER, M. ; HECK, M. ; YIGIT, S.: Efficiency and Accuracy of Fluid–Structure Interaction Simulations using an Implicit Partitioned Approach. In: *Comput Mech* 43 (2008), S. 103–113
- [146] STRANG, G.: *Lineare Algebra*. Springer, 2003
- [147] STRANG, G.: *Computational Science and Engineering*. Wellesley-Cambridge Press, 2007
- [148] STROUSTRUP, B.: *The Design and Evolution of C++*. Addison-Wesley, 1994
- [149] STROUSTRUP, B.: *Die C++ Programmiersprache*. Addison-Wesley, 1998
- [150] STURLER, E. de: Nested Krylov Methods based on GCR. In: *J. Comp. and Appl. Math.* 67 (1996), Nr. 1, S. 15–41
- [151] SZABÓ, I.: *Geschichte der mechanischen Prinzipien*. Birkhäuser Verlag, 1996
- [152] TEZDUYAR, T.E. ; SATHE, S.: Modelling of Fluid–Structure Interactions with the Space-Time Finite Elements: Solution Techniques. In: *Int. J. Numer. Meth. Fluids* 54 (2007), Nr. 6–8, S. 855–900
- [153] TEZDUYAR, T.E. ; SATHE, S. ; CRAGIN, T. ; NANNA, B. ; CONKLIN, B.S. ; PAUSEWANG, J. ; SCHWAAB, M.: Modelling of Fluid–Structure Interactions with the Space-Time Finite Elements: Arterial Fluid Mechanics. In: *Int. J. Numer. Meth. Fluids* 54 (2007), Nr. 6–8, S. 901–922
- [154] TREFETHEN, L.N. ; BAU, D.: *Numerical linear algebra*. SIAM, 1997
- [155] TROTTEBERG, U. ; OOSTERLEE, C. ; SCHÜLLER, A.: *Multigrid*. London : Academic Press, 2001
- [156] ULBRICHT, V.: Studienbrief Kontinuumsmechanik / TU Dresden. 1997. – Forschungsbericht
- [157] VANĚK, P. ; BREZINA, M. ; MANDEL, J.: Convergence of Algebraic Multigrid based on Smoothed Aggregation. In: *Numer. Math.* 88 (2001), Nr. 3, S. 559–579
- [158] VIERENDEELS, J.: Implicit Coupling of Partitioned Fluid–Structure Interaction Solvers using a Reduced Order Model. In: *AIAA Fluid Dynamics Conference and Exhibit* 35 (2005), S. 1–12
- [159] VIERENDEELS, J.: Implicit Coupling of Partitioned Fluid–Structure Interaction Solvers using Reduced-Order Models. In: *Fluid–Structure Interaction: Modelling, Simulation, Optimisation* Bd. 53. Springer, 2006, S. 1–18
- [160] VIERENDEELS, J. ; LANOYE, L. ; DEGROOTE, J. ; VERDONCK, P.: Implicit Coupling of Partitioned Fluid–Structure Interaction Problems with Reduced Order Models. In: *Comput. Struct.* 85 (2007), S. 970–976
- [161] VORST, H.A. V. der ; VUIK, C.: GMRESR: A Family of Nested GMRES Methods. In: *Num. Lin. Alg. with Appl* 1 (1994), Nr. 4, S. 369–386
- [162] WALL, W. A.: *Fluid-Struktur-Interaktion mit stabilisierten Finiten Elementen*, Institut für Baustatik, Universität Stuttgart, Dissertation, 1999
- [163] WALL, W. A. ; MOK, D.P. ; RAMM, E.: Partitioned Analysis Approach of the Transient Coupled Response of Viscous Fluids and Flexible Structures. In: *W. Wunderlich (Ed.), Solids, Structures and Coupled Problems in Engineering, Proc. ECCM '99, Munich*, 1999
- [164] WALL, W.A. ; GENKINGER, S. ; RAMM, E.: A Strong Coupling Partitioned Approach for Fluid–Structure Interaction with Free Surfaces. In: *Comput. Fluids* 36 (2007), Nr. 1, S. 169–183

- [165] WASHIO, T. ; OOSTERLEE, C. W.: Krylov Subspace Acceleration Method for Non-linear Multigrid Schemes. In: *ETNA, Electron. Trans. Numer. Anal* 6 (1997), S. 271–290
- [166] WIESNER, T.: *Semi-implizite Kopplungsverfahren für Fluid-Struktur-Interaktionsprobleme*, Lehrstuhl für Numerische Mechanik, TU München, Diplomarbeit, 2009
- [167] WRIGGERS, P.: *Nichtlineare Finite-Element-Methoden*. Springer, 2004 (XI)
- [168] ZHANG, Q. ; HISADA, T.: Studies of the Strong Coupling and Weak Coupling Methods in FSI Analysis. In: *Int. J. Numer. Meth. Engng.* 60 (2004), S. 2013–2029
- [169] ZURMÜHL, R. ; FALK, S.: *Matrizen und ihre Anwendungen Teil 2: Numerische Methoden*. Springer, 1986
- [170] ZURMÜHL, R. ; FALK, S.: *Matrizen und ihre Anwendungen Teil 1: Grundlagen*. Springer, 1997