

Kopplungswerkzeuge für aeroelastische Simulationen

Von der Fakultät für
Mathematik, Informatik und Naturwissenschaften
der Rheinisch-Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades eines Doktors der
Naturwissenschaften genehmigte Dissertation
vorgelegt von

Diplom-Mathematiker
Alexander Boucke
aus Aachen

Berichter:
Universitätsprofessor Dr.-Ing. J. Ballmann
Universitätsprofessor Dr. W. Dahmen
Tag der mündlichen Prüfung: 31. Juli 2003

Diese Dissertation ist auf den Internetseiten
der Hochschulbibliothek online verfügbar.

Für Karin

Vorwort

Während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehr- und Forschungsgebiet für Mechanik an der RWTH Aachen ist die vorliegende Arbeit entstanden. Herrn Professor Dr.-Ing. Josef Ballmann gilt mein besonderer Dank für die Anregung und durchgehende Unterstützung zu dieser Arbeit.

Für seine bereitwillige Übernahme des Zweitreferates der Arbeit möchte ich Herrn Professor Dr. Wolfgang Dahmen danken.

Meine Kollegen vom Institut möchte ich an dieser Stelle nicht vergessen zu erwähnen, mit deren Unterstützung durch fachliche Hilfe, Bereithalten der Computer — oder einfach nur gute Laune — vieles einfacher wurde.

Besonders hervorheben möchte ich an dieser Stelle Dipl.-Phys. Michael Hanke, ohne dessen Mitarbeit auf der aerodynamischen Seite der Simulation diese Arbeit gar nicht zu Stande gekommen wäre, und Dipl.-Ing. Carsten Braun, der mir freundlicherweise einige seiner Ergebnisse zur Verfügung gestellt hat und mir als „Beta“-Tester für das Gitterdeformationsprogramm große Teile der Fehlersuche abgenommen hat.

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung | 1 |
| 1.1 | Stand der Technik | 2 |
| 1.2 | Zielsetzung | 3 |
| 2 | Gitterdeformationsmodell | 5 |
| 2.1 | Problemstellung | 5 |
| 2.2 | Fiktive Rahmentragwerke | 7 |
| 2.3 | Implementierung für Mehrblocknetze vom FLOWer-Typ | 10 |
| 2.3.1 | Aufbau des fiktiven Rahmentragwerkes | 10 |
| 2.3.2 | Interpolation im Blockinneren | 13 |
| 2.3.3 | Problemfälle | 15 |
| 2.4 | Beispiele | 16 |
| 2.4.1 | Ebene Platte mit Gelenk (2D) | 16 |
| 2.4.2 | Rakete (3D) | 16 |
| 3 | Modellierung des elastischen Festkörpers | 21 |
| 3.1 | FE und Timoshenko-Balken | 21 |
| 3.1.1 | Friedman-Kosmatka-Element | 22 |
| 3.1.2 | Validierung mittels Beispielproblem | 22 |
| 3.2 | Erweitertes Balkenelement für die Aeroelastik | 30 |
| 3.2.1 | Bewegungsgleichungen | 31 |
| 3.2.2 | Formfunktionen | 34 |
| 3.2.3 | Elementmatrizen nach dem Hamilton-Prinzip | 38 |
| 4 | Datentransfer | 41 |
| 4.1 | Zuordnung FE-Modell und Oberfläche | 42 |
| 4.2 | Lastübertragung auf das erweiterte Balkenelement | 45 |
| 4.3 | Verschiebung der Oberfläche über einem Balkenelement | 49 |
| 4.4 | Interpolation bei Zuordnung zweier Balkenelementgruppen zu einer Oberfläche | 50 |
| 5 | Freiflugmodellierung | 53 |
| 5.1 | „Mean Axes“ | 54 |
| 5.1.1 | Massenträgheits-gemittelte Bewegung | 54 |
| 5.1.2 | Separation der elastischen Deformation | 56 |
| 5.1.3 | Freiflug des starren Flugkörpers | 58 |
| 5.2 | Funktionsbeispiele | 58 |

| | | |
|----------|---|-----------|
| 5.3 | Modell für Steuerflächen | 62 |
| 6 | Kopplung an FLOWer | 63 |
| 6.1 | Kopplungsschema | 63 |
| 6.2 | Stationäre Windkanalströmung und Startwerteberechnung . . . | 64 |
| 6.3 | Schnittstelle der Aeroelastik-Programmbibliothek für FLOWer . | 66 |
| 7 | Simulationsergebnisse | 69 |
| 7.1 | Tragflügel quasistatisch | 69 |
| 7.2 | Lenkflugkörper elastisch mit Lenkbewegung im Freiflug | 73 |
| 8 | Zusammenfassung und Ausblick | 81 |
| A | Implementierungsdetails | 83 |
| A.1 | Die Programmiersprache Ada95 | 83 |
| A.2 | Finite Elemente | 86 |
| A.3 | Hilfsbibliotheken | 87 |
| A.3.1 | Rechnen mit Vektoren und Matrizen | 87 |
| A.3.2 | Kontrolle des Rundungsfehlers | 90 |
| B | FE-Matrizen für erweitertes Balkenelement | 93 |
| B.1 | Elementsteifigkeitsmatrix | 94 |
| B.2 | Elementmassenmatrix | 98 |

Kapitel 1

Einleitung

Der numerischen Simulation der Interaktion von Strömungen und elastischen Strukturen unter Berücksichtigung aller Nichtlinearitäten des Problems kommt in der Aeroelastik immer größere Bedeutung zu. Treibende Kraft hierfür ist u.a. die zunehmende Größe projektierter Transportflugzeuge für den schallnahen Reiseflug und die steigende Verwendung von Leichtbautechniken. Dies führt im Flug zu so großen elastischen Auslenkungen der Tragflächen, daß dies bereits in frühen Stadien der Projektierung berücksichtigt werden muß, da die Einflüsse der elastischen Verformungen des Flugzeugs für die aerodynamischen Eigenschaften sehr bedeutend werden können. Dies gilt nicht nur für instationäre Vorgänge mit einem wechselseitigen periodischen Energieaustausch der Struktur- und Strömungskräfte, der zu Instabilitäten, z.B. Flattern, führen kann, sondern auch für den stationären oder quasistationären Flug. Eine widerstandsminimale Konfiguration im Reiseflug oder eine Steig- bzw. Sinkflug-optimierte Hochauftriebskonfiguration ist für große Transportflugzeuge ohne Berücksichtigung der Deformation der Flugzeugoberfläche im Flug nicht möglich. Sogar Modelluntersuchungen in transsonischen Windkanälen, bei denen gleichzeitig Machzahl und Reynoldszahl des Reisefluges eingehalten werden, sind ohne Erfassung der Modellverformungen nicht richtig zu interpretieren.

Bisher wurde der Einfluß der Elastizität im wesentlichen durch iteratives (abwechselndes) Betrachten der (nichtlinearen) aerodynamischen Eigenschaften und der Struktur unter Last berücksichtigt und im Anschluß daran, in der Schlußphase der Entwicklung, die aerodynamische Stabilität des Gesamtsystems z.B. hinsichtlich der Flattergrenze mit linearen Methoden untersucht.

Durch die Nichtlinearität der gasdynamischen Gleichungen, die zunehmende Größe der Flugzeuge und den erweiterten Einsatz von Leichtbaustrukturen entsteht durch die komplexe Wechselwirkung zwischen Strömung und Struktur ein i. allg. hochgradig nichtlineares System, bei dem die Prognose des Verhaltens in der Praxis mit den herkömmlichen Methoden kaum noch möglich ist. Insbesondere kommen neben der Untersuchung der Reiseflugkonfiguration oder der Flattergrenze auch zunehmend Einflüsse von Steuerbewegungen in Betracht, da die Eigenfrequenzen der Tragflächen niedriger werden und die Eigenschwingungen durch Steuerbewegungen oder Windböen

somit leichter angeregt werden können.

All dies verlangt nach dem Einsatz der direkten Simulation des gekoppelten aeroelastischen Gesamtsystems bereits im Vorfeld der Entwicklung, um evtl. auftretende Einbußen der aerodynamischen Leistung oder Steuerbarkeit eines Flugzeugs bereits frühzeitig erkennen und vermeiden zu können.

In dieser Arbeit werden Bauteile zu einem Programm vorgestellt, mit deren Hilfe die Durchführung solcher direkten aeroelastischen Simulationen (auch Computational Aeroelastic Simulation, CAS) unter Verwendung vorhandener Verfahren für die numerische Strömungssimulation (Computational Fluid Dynamics, CFD) möglich wird.

1.1 Stand der Technik

Generell ist zwischen Frequenzbereichsverfahren und Zeitbereichsverfahren zu unterscheiden. Erstere sind gut geeignet für vollständig linearisierte Probleme um schlanke Flugkörper [WS98] und Tragflügel ohne Reibung. Bei der Auslegung heutiger Fluggeräte werden zumeist auch noch Frequenzbereichsverfahren verwendet. Praktisch werden in der Regel mehrere entkoppelte Probleme betrachtet: So wird die Auslegung eines Tragflügels im Windkanalversuch oder auch per Simulationsberechnung zwar unter Berücksichtigung der Verdrehung der einzelnen Querschnitte (sog. Twist) durchgeführt, die zu erwartende Verbiegung des Tragflügels im Flug jedoch nachträglich aus den so ermittelten aerodynamischen Belastungen ermittelt. Durch den hohen Grad der Nichtlinearität des gekoppelten aeroelastischen Problems, z.B. im transsonischen Flug eines sehr elastischen Flugzeuges, wird diese Vorgehensweise den Anforderungen nicht mehr gerecht und die Anwendung von Zeitbereichsverfahren notwendig, bei denen mit numerischer Zeitintegration in beiden Gebieten operiert wird. Derartige multidisziplinäre Verfahren mit genaueren mathematischen Modellen und numerischen Berechnungsverfahren für die vollständig gekoppelte Aerostrukturodynamik sind daher weltweit Gegenstand der aktuellen Forschung [Hön01].

Unter den im Zeitbereich operierenden Verfahren können im wesentlichen zwei Grundverfahren zur numerischen Simulation von Strömungs-Struktur-Interaktion unterschieden werden. Zumeist wird eine sogenannte „lose Kopplung“ von zwei separaten Verfahren für Struktur und Strömung vorgenommen, die mathematisch als eine Gebietszerlegung betrachtet werden kann [Sch97, Bri03, Bal03, HB01, KHS, FPD01, MNF01]. Seltener, vor allem aufgrund der hohen Komplexität, die eine wirtschaftliche Anwendung erschwert, ist eine „monolithische“ Mehrfeldformulierung durch ein System von gekoppelten Differentialgleichungen [BH97, Mas02].

Das erste Vorgehen bietet den Vorteil, vorhandene Verfahren durch eine geeignete Schnittstelle algorithmisch miteinander verbinden zu können. Es wird in der vorliegenden Arbeit zum Einsatz gebracht. Bei diesem Verfahren sind zwei grundsätzliche Probleme zu beachten: eine einfache, alternierende Kopplung führt in der Regel nicht zu einem zeitgenauen System. In [Hur02, HB01, Mas02, FPD01] wurden Ansätze vorgestellt, diesen sogenann-

ten Kopplungsphasenfehler zu vermeiden. Der zweite, für die Qualität des Gesamtverfahrens entscheidende Punkt ist die Übergabe der aerodynamischen Lasten bzw. Oberflächenbewegungen zwischen Strömungs- und Struktursimulation. Dabei ist die Energiebilanz des Gesamtsystems ebenso zu beachten [Mas02, CL97, FLL98] wie eine konsistente Beschreibung der Oberflächendeformation. Da beide Punkte stark von den verwendeten Verfahren, insbesondere für die Simulation der Strukturmechanik, abhängen, bilden diese Punkte einen Schwerpunkt dieser Arbeit.

Für die Strömungssimulation sind im wesentlichen Finite-Elemente oder Finite-Volumen-Verfahren im Einsatz, wobei die letztgenannten bei weitem überwiegen. Je nach Einsatzzweck werden hier die Euler- oder die Navier-Stokes-Gleichungen gelöst. Die vorliegende Arbeit ist nicht auf ein spezielles Modell der bekannten Lösungsverfahren fokussiert, jedoch wird der Strömungslöser `FLOWer`¹ [FLO00, KRBT00], ein dreidimensional rechnendes Finite-Volumen-Verfahren für strukturierte Hexaedernetze, als typisches Programm verwendet.

Auf der Strukturseite kommen heutzutage fast ausschließlich Finite-Elemente-Verfahren zum Einsatz, mit deren Hilfe sich auch komplizierte Tragstrukturen gut modellieren lassen. So ist `NASTRAN` das bei der Flugzeugentwicklung zumeist eingesetzte Programm (z.B. auch in [KHS]). Für viele Untersuchungen ist jedoch eine reduzierte Modellierung der Tragstrukturen durch ein Balkenmodell [BBH⁺01, Bri03, Nel96] möglich. In dieser Arbeit wird ein solches reduziertes Modell verwendet, bei dem die tragende Struktur unter Verwendung verallgemeinerter Timoshenko-Balkenelemente modelliert wird.

1.2 Zielsetzung

Aufbauend auf vorhergehende Arbeiten im Rahmen des SFB 401² [Bal03, HB01, Bri03, Mas02] werden in dieser Arbeit fehlende Komponenten für die Kopplung der verwendeten Strömungslöser, vornehmlich `FLOWer`, und eines Strukturmodells implementiert, wie auch das Strukturmodell selbst.

Einen wesentlichen Bestandteil stellt ein neues Verfahren für die Deformation der Rechengitter des Strömungslösers dar, das mit Hinblick auf instationäre Berechnungen mit längerer Laufzeit effizient arbeiten soll. Es wurde ein einfacher Ansatz gewählt, der die Vorteile verschiedener Verfahren verknüpft. Praktisch wird dabei ein grobes Gitter als elastisches System betrachtet, das die Winkeltreue der Gitterlinien in ihren Kreuzungspunkten beachtet. Das benötigte feine Gitter wird darin per Interpolation bestimmt.

Insgesamt wird eine Bibliothek von Programmteilen zur Verfügung gestellt, die mit wenigen Änderungen im Strömungslöser die Berechnung von Strömungs-Struktur-Interaktionen im stationären oder dynamischen Fall er-

¹Vom DLR in Zusammenarbeit mit verschiedenen Partnern aus Industrie und Forschung entwickelt. Das Lehr- und Forschungsgebiet für Mechanik ist ebenfalls daran beteiligt.

²Sonderforschungsbereich 401 der Deutschen Forschungsgemeinschaft, Bonn. Thema: Strömungsbeeinflussung und Strömungs-Struktur-Wechselwirkung an Tragflügeln. Seit 1997 an der RWTH Aachen gefördert.

laubt. Dabei soll das betrachtete Fluggerät über beliebige Freiheitsgrade der starren Bewegung verfügen, also frei fliegend oder als Modell im Windkanal irgendwie gelagert sein können. Die einzelnen Bestandteile werden zunächst bzgl. ihrer Funktionsweise und Implementierung beschrieben, abschließend werden verschiedene Simulationsergebnisse aus der Anwendung des Gesamtprogrammes dargestellt.

Um die Stabilität und Steuerbarkeit, etwa die Wirkung von Lenkmanövern, besser bestimmen zu können oder entsprechende aerodynamische oder aeroelastische Beiwerte effektiv berechnen zu können, ist ein Ziel der Verfahren, die Berechnung der Flugbahn für einen elastischen oder starren Flugkörper mit Hilfe direkter numerischer Simulation durchzuführen.

Kapitel 2

Gitterdeformationsmodell

2.1 Problemstellung

Für die Erfüllung der Strömungsrandbedingungen während einer direkten Simulation von aeroelastischen Vorgängen ist es notwendig, die Geometrie des Rechennetzes für die Strömung den Deformationen der aerodynamischen Oberfläche des elastischen Körpers anzupassen, in der englischsprachigen Literatur wurde dafür der Begriff „dynamic mesh“ geprägt. Bei den hier verwendeten Programmen für die Strömungssimulation kommen Rechennetze zum Einsatz, die aus mehreren Blöcken strukturierter Hexaeder- oder Vierecksnetze aufgebaut sind. Dabei können Situationen auftreten, in denen nicht nur die Netze deformiert werden müssen, die direkt an den elastischen Körper angrenzen, sondern in denen auch entfernt liegende bzw., je nach Anwendungsfall, alle Bereiche des Rechengitters betroffen sind.

Bei strukturierten oder unstrukturierten Hexaedernetzen für Differenzenverfahren ist es für eine gute Simulation der Strömung wesentlich, daß die einzelnen Zellen eine möglichst rechteckige Form haben. Insbesondere im Grenzschichtbereich an festen Wänden hängt die Qualität der Lösung von der Richtung der von der Oberfläche abgehenden Gitterlinien ab. An der Grenze zweier Rechenblöcke ist es für eine schnelle Konvergenz der Lösung wichtig, daß die Gitterlinien beim Übergang von einem Block in den anderen möglichst wenig abknicken. Diese Punkte sind bereits bei der Erstellung solcher Rechennetze zu beachten, können aber bei der Deformation des Gitters leicht verletzt werden. Für Tetraedernetze, z.B. von FE-Verfahren, gelten ähnliche Bedingungen, teilweise in abgeschwächter Form. Aufgrund der besseren Qualität der Simulationsergebnisse für Hexaedernetze im körpernahen Bereich, werden auch oft hybride Netze verwendet, die beide Typen miteinander vereinen, indem im Grenzschichtbereich quasi-strukturierte Hexaeder- oder Prismennetze verwendet werden und außerhalb davon unstrukturierte Tetraedernetze.

Das Deformationsprogramm muß also in der Lage sein, das Strömungsgitter bei vorgegebener Deformation bestimmter Oberflächen so zu deformieren, daß die Schnittwinkel der Gitterlinien, und damit die Form der einzelnen Zellen, weitgehend erhalten bleiben. Damit mögliche Fehlerquellen bei der Anwendung des Programms minimiert werden, soll zudem die Zahl der

Eingriffsmöglichkeiten durch den Anwender auf ein Minimum reduziert werden. Dies bedingt einen weitgehend automatisierten Ablauf.

Für die Nachführung von Gittern werden verschiedene Techniken verwendet. Neben der Interpolation der Verschiebungen [Bri03, HBB00, Wer97], ist hier vor allem die Modellierung der Gitter als fiktives Fachwerk aus gelenkig verbundenen Federn zu nennen [Bat89, Wic01]. Beide Vorgehensweisen haben für Hexaedernetze, auf denen der Schwerpunkt dieser Arbeit liegt, erhebliche Nachteile. Bei der reinen Interpolation ist es schwierig, Mehrblocknetze zu deformieren, wenn mehr als ein Block deformiert werden muß. Ein wesentlicher Nachteil bei Verwendung des Federmodelles ist die fehlende Winkeltreue sowohl an den Oberflächen, als auch an den inneren Knoten des Gitters. Dies kann in der Regel durch eine nachgeschaltete elliptische Glättung des Gitters teilweise wieder kompensiert werden. Jedoch ist diese in der Anwendung nicht nur recht teuer, sondern an konkaven Konturknicken auch gitterverschlechternd, indem die Verteilung der Gitterlinien in der Grenzschicht lokal vergrößert wird. Dadurch geht der Vorteil der Einfachheit der genannten Verfahren mindestens teilweise wieder verloren. Ein anderer wichtiger Nachteil des Federmodelles ist, daß hängende Knoten nicht möglich sind. Dieser Mangel kann auch nicht durch einen Alternativvorschlag von [DF02] beseitigt werden, indem in den Knoten des Netzes eine zusätzliche Torsionssteifigkeit hinzugefügt wird, die die Winkeländerungen bei der Deformation des Netzes reduzieren.

Ein anderer möglicher Ansatz ist die Betrachtung des Gitters als elastischer Körper. Hier machen [JT94, HCAK02] einen Vorschlag, das komplette Gitter als homogenen, linearelastischen, massiven Festkörper aufzufassen. Für die Deformation großer Gitter ist dafür allerdings auch ein hoher Rechenaufwand notwendig. In [Wic01] wird, um entsprechend große starre Lageveränderung des gesamten Gitters auffangen zu können, die Festkörperanalogie auf einen hyperelastischen Ansatz erweitert. Allerdings wird auch festgestellt, daß bei moderaten Drehwinkeln (im Beispiel 30°) das linearelastische Modell eine gute Gitterqualität bei wesentlich geringerer Rechenzeit liefert.

In dieser Arbeit wird ein Mittelweg zwischen dem linearelastischen massiven Festkörper zur Darstellung des Netzes und der Federanalogie beschritten. Ähnlich wie bei der Federanalogie werden die Kanten des Netzes durch physikalische Strukturen nachgebildet, jedoch werden anstelle von Federn linearelastische Balken verwendet. In Abb. 2.1 wurde ein Gitter über einer gelenkig gelagerte Platte einer starken Deformation unterzogen. In der linken Bildhälfte ist das deformierte Gitter nach linearer Interpolation dargestellt, in der rechten Hälfte wurde für die Berechnung der Deformation ein elastisches Modell herangezogen, bei dem wie oben angesprochen alle Gitterkanten durch Balken dargestellt werden. Der zweite Ansatz hat klar sichtbare Vorteile, da keine Knicke in den Gitterlinien auftreten und trotzdem die Verteilung der Linien am Rand relativ gut gewährleistet bleibt, wie in Abb. 2.2 zu sehen ist. Allerdings entsteht, ähnlich wie bei der Modellierung als elastischer Festkörper, ein relativ großer Rechenaufwand.

Der hier vorgestellte Ansatz versucht die Vorteile beider Wege zu verbinden, indem für weite Teile des Gitters ein schneller und einfacher Interpola-

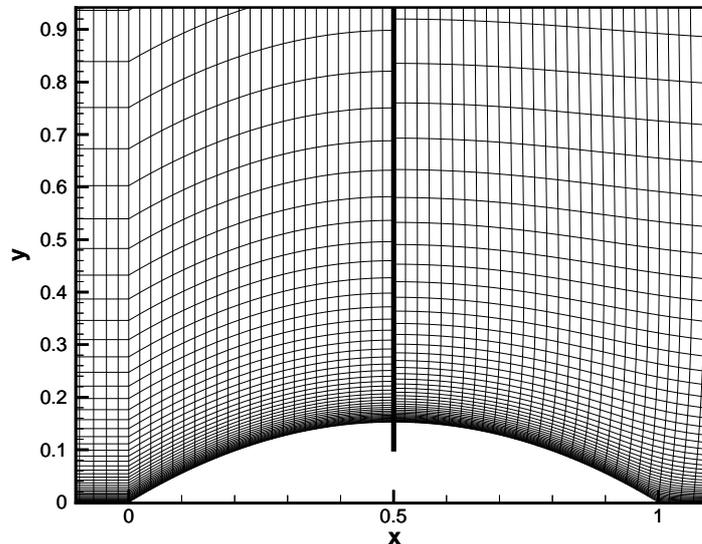


Abbildung 2.1: Vergleich der Gitterdeformation via Interpolation (linke Bildhälfte) und einer Deformation unter Verwendung von elastischen Balken für alle Gitterkanten (rechte Bildhälfte).

tionsansatz verwendet wird und nur ausgewählte Linien als Festkörper deformiert werden. Entscheidend für die Qualität derart erzeugter Netze ist offensichtlich die Auswahl der elastisch zu modellierenden Linien, wie weiter unten ausführlich erläutert wird.

Auch wenn die Deformation strukturierter Mehrblocknetze in der aktuellen Aufgabe im Vordergrund steht, so bietet diese Vorgehensweise die Möglichkeit, beliebige Gitterstrukturen zu deformieren, indem ein geeignetes, sehr grobes Netz der elastischen Deformation unterworfen wird.

2.2 Fiktive Rahmentragwerke

Soll eine Verbindungslinie zwischen zwei Punkten an eine Änderung der Lage einzelner Punkte auf dieser Linie und in deren Umgebung angepaßt werden, so bietet sich die Modellierung dieser Linie bei nicht zu großen Dehnungen als linearelastischer Balken an. Aufgrund der physikalischen Eigenschaften eines elastischen Balkens können sowohl Verschiebungen als auch Winkeländerungen übertragen werden. Hinzu kommt, daß unnötige Krümmungen der deformierten Linie vermieden werden, entsprechend dem Prinzip der Minimierung der Formänderungsenergie, die weit überwiegend aus Verkrümmungsenergie besteht.

Verwendet man nun für eine geeignete Auswahl an Gitterlinien fiktive, durchlaufende Balken, die an den Kreuzungspunkten fest miteinander

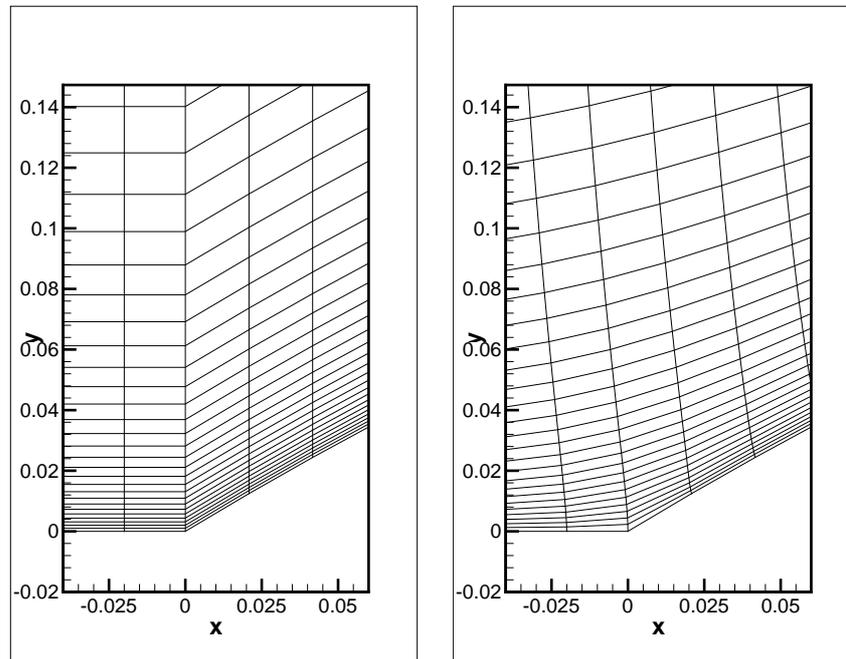


Abbildung 2.2: Vergleich der Gitterdeformation via Interpolation (linke Bildhälfte) und einer Deformation unter Verwendung von elastischen Balken für alle Gitterkanten (rechte Bildhälfte). Jeweils Ausschnitte der Gitter aus Abbildung 2.1

verschweißt sind, so erhält man ein fiktives Rahmentragwerk. Für ebene Strömungsprobleme ist es ein ebenes Tragwerk, wie es in Abb. 2.1 in der rechten Bildhälfte dargestellt ist. Für dreidimensionale Strömungen ergibt sich entsprechend ein fiktives, räumliches Rahmentragwerk. Die fiktiven Balken werden dort, wo sie auf die umströmte Oberfläche des elastischen Festkörpers auftreffen, fest eingespannt. Dies ist für die Erhaltung der Winkel des Rechengitters für die Strömung an den Oberflächen des Festkörpers wesentlich, da diese durch die Einspannung gewährleistet wird. Durch an den Einspannungen im Festkörper vorgegebene Verschiebungen und Winkeländerungen wird das gesamte fiktive Rahmentragwerk deformiert. Aus den Deformationen der einzelnen Balken kann nun die Anpassung des gesamten Rechnetzes konstruiert werden.

Das fiktive Rahmentragwerk ist masselos, so daß es in jeder Lage, auch bei beliebigen zeitabhängigen Verformungen, eine elastische Gleichgewichtskonfiguration darstellt. Deshalb unterliegt die aktuelle Verformung bei Annahme linearelastischen Materialverhaltens unter den aufgeprägten Rand- und Übergangsbedingungen dem Minimumprinzip des elastischen Potentials.

Für die Modellierung des Rahmentragwerkes wird ein Finite-Elemente-Modell verwendet. Da bereits für die Modellierung des elastischen Flugkörpers ein entsprechendes Programm im Rahmen dieser Arbeit erstellt wurde (s. Kapitel 3), bot sich diese Vorgehensweise als vorteilhaft an.

Das Rahmentragwerk wird hier in Form von Listen aus Knoten und Linien abgebildet. Die Knoten sind weiterhin in verschiedene Klassen, entsprechend den dort herrschenden Randbedingungen, unterteilt:

1. Rumpfknoten: Knoten, die auf der Oberfläche des elastischen Festkörpers liegen. Zu jedem dieser Knoten muß eine Deformation, bestehend aus Verschiebung und Verdrehung, vorgegeben werden.
2. Wandknoten: Diese Knoten sind am äußeren Rand des Rechengebietes unbeweglich fest eingespannt und können z.B. für Fernfeldränder verwendet werden, die nicht verschoben werden dürfen. Wand- und Rumpfknoten schließen sich gegenseitig aus.
3. Symmetrieknoten: Jedem Knoten können eine oder mehrere Symmetriebedingungen zugeordnet werden. Eine x -Symmetrie bedeutet, daß der Knoten in Richtung der x -Achse nicht verschoben werden kann. Wandknoten erfüllen naturgemäß alle Symmetriebedingungen. Bei Rumpfknoten muß bei der Vorgabe der Deformation bereits auf eventuelle Symmetrien geachtet werden.
4. Freie Knoten: Alle übrigen Knoten sind frei beweglich.
5. Hängende Knoten sind zugelassen. Es sind freie oder Symmetrieknoten, in denen ein fiktiver Balken auf einem anderen durchlaufenden fiktiven Balken endet.

Mit dieser Diskretisierung ergibt sich für das Prinzip der Minimierung der Formänderungsenergie mit der potentiellen Energie U des Rahmens (für einen einzelnen Balken in Gl. (3.20) auf Seite 38 dargestellt)

$$U = \frac{1}{2} \Delta^T \mathbf{K} \Delta$$

das zu lösende Minimalproblem

$$U - W = \min.$$

Darin repräsentiert das Symbol Δ die generalisierten Verschiebungen in den Knoten des fiktiven Rahmens und \mathbf{K} die Steifigkeitsmatrix, die symmetrisch ist (vgl. Kapitel 3). Ferner ist $W = \Delta^T \mathbf{F}$ die Arbeit der aus den Randbedingungen abgeleiteten generalisierten Kräfte \mathbf{F} .

Zur Bestimmung der Deformation des Rahmens zu einer gegebenen Deformation an den Rumpfknoten wird das oben bereits genannte Minimumprinzip für die elastische Energie des fiktiven Rahmentragwerkes gelöst, das die stationäre Lösung liefert. Zur Lösung des daraus resultierenden, großen Gleichungssystems $\mathbf{K} \Delta = \mathbf{F}$ wird ein Verfahren der konjugierten Gradienten mit Vorkonditionierung durch ein symmetrisches Einzelschrittverfahren verwendet [Gre97, BBC⁺94].

2.3 Implementierung für Mehrblocknetze vom FLOWer-Typ

Alle oben gemachten Ausführungen über das Netzdeformationsverfahren an sich sind für beliebige Gitterstrukturen anwendbar. Das Verfahren kann für zwei- oder dreidimensionale Gitter strukturierter wie unstrukturierter Art implementiert werden. Nachfolgend wird die Implementierung für die Verwendung mit dem Strömungslöser FLOWer beschrieben. FLOWer rechnet auf Netzen, die aus mehreren, sich nicht überschneidenden Blöcken strukturierter Hexaedernetze bestehen.

Die Implementierung für strukturierte Mehrblocknetze sorgt für den automatischen Aufbau eines passenden fiktiven Rahmenmodells zum gegebenen Rechennetz. Nach der Bestimmung von dessen Deformation wird diese hier in das Rechennetz übertragen und das neue, deformierte Rechennetz erstellt.

2.3.1 Aufbau des fiktiven Rahmentragwerkes

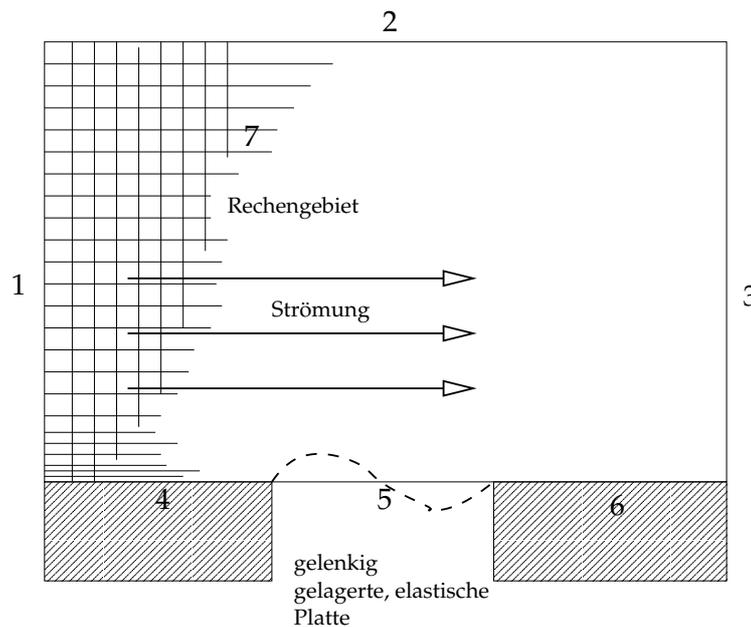


Abbildung 2.3: Aeroelastische Problemstellung nach [HB01, BD95, Mas02]. Die Randbereiche 1,2 und 3 (durchströmbare Gebietsränder) des Gitters werden wahlweise als freie Knoten oder aber als Wandknoten in das fiktive Rahmentragwerk übernommen, die nicht durchströmbaren Segmente 4,5 und 6 als Rumpfknoten. Evtl. im Inneren des Gebietes liegende Blockgrenzen (7) werden als Linien mit freien Knoten übernommen. Wird das Problem eben betrachtet (z.B. zweidimensional), so müssen alle genannten freien Knoten natürlich eine entsprechende Symmetriebedingung für die Berechnungsebene erfüllen.

Damit das Programm zur Gitternachführung einfach handhabbar und auch in vollständig gekoppelten Simulationsrechnungen einsetzbar ist, muß das fiktive Rahmentragwerk aus einem gegebenen FLOWer-Gitter vollautomatisch extrahiert werden. Dieser Vorgang kann durch verschiedene Parameter beeinflußt werden, von denen zwei für den Anwender wählbar sind, um eventuell nichtidealen Gegebenheiten Rechnung zu tragen.

Es werden generell drei Typen von Gitterlinien unterschieden: Kantenlinien von Gitterblöcken, Grenzen zwischen verschiedenen Randbedingungen auf den Blockoberflächen, den sogenannten Segmenten, und sonstige Linien. Alle Linien der ersten beiden Kategorien sind für die Konstruktion des Gitters, und damit auch seine Anpassung, wesentlich und werden automatisch in das fiktive Rahmentragwerk aufgenommen.

Treten Blöcke auf, die geometrisch große Streckungsverhältnisse aufweisen, so kann es notwendig sein, zusätzliche Linien zur guten Nachführung des Gitters heranzuziehen. Wenn beispielsweise ein flaches C-Netz einen gesamten Tragflügel umgibt, so könnte eine Durchdringung der Flügeloberfläche mit der Außenfläche des C-Netzes auftreten, falls nicht entsprechende Balken als „Stützen“ in den Rahmen eingefügt werden.¹ Um diesem Problem zu begegnen, wird ein vom Anwender wählbarer Prozentsatz aller übrigen Gitterlinien mit in das Rahmentragwerk aufgenommen.

In den Abbildungen 2.4 und 2.5 ist solch ein Fall dargestellt. Im Gitter wurde entlang einer horizontal nahe der überströmten Fläche verlaufenden Linie eine Blockgrenze eingeführt. Bei der Deformation besteht jedoch keine Verbindung zwischen der deformierten Oberfläche und der Blockgrenze, so daß diese nicht deformiert wird. Wählt man dagegen zusätzlich zu den Blockgrenzen und ähnlichen Linien 6% aller Gitterlinien für das fiktive Rahmentragwerk aus, so wird die Blockgrenze gegenüber der sich deformierenden Oberfläche abgestützt und entsprechend mitbewegt. In den Abbildungen wird die Position der Knoten in den grau hinterlegten Gebieten durch Interpolation (s.u.) festgelegt, wohingegen die von einem weißen Band eingefassten Gitterlinien Bestandteile des fiktiven Rahmentragwerkes sind.

Anhand der vorgegebenen Randbedingungen für die Strömungssimulation werden verschiedene Randbedingungen für das Rahmentragwerk festgelegt. Alle äußeren oder inneren Flächen vom Typ „feste Wand“ werden als Rumpfknoten modelliert. Aus- und Einstromflächen und übrige Fernfeldränder werden auf Wunsch unbeweglich (als Wandknoten) oder beweglich gestaltet. In den Abb. 2.6 und 2.7 wird an einem starr gedrehten Tragflügelprofil dargestellt, wie sich die Gitternachführung mit festgehaltenem bzw. beweglichem Außenrand verhält. Die Verwendung eines beweglichen Außenrandes ist außer bei großen Winkeländerungen, wie im vorangehenden Beispiel, gerade bei Gittern für Über- oder Hyperschallsimulationen, bei denen das Gitter an der Einströmseite oft sehr dicht an den Flugkörper heranreicht, wichtig, da das Gitter unter Umständen zu klein ist, um die Bewegungen des Flugkörpers aufzufangen. Im übrigen werden Symmetriebedingungen

¹Ein typischer Anwendungsfall für diese Vorgehensweise wäre auch die Deformation eines oberflächennahen Netzbereiches, der mit Hilfe der Offset-Technik [Bra92] erstellt wurde.

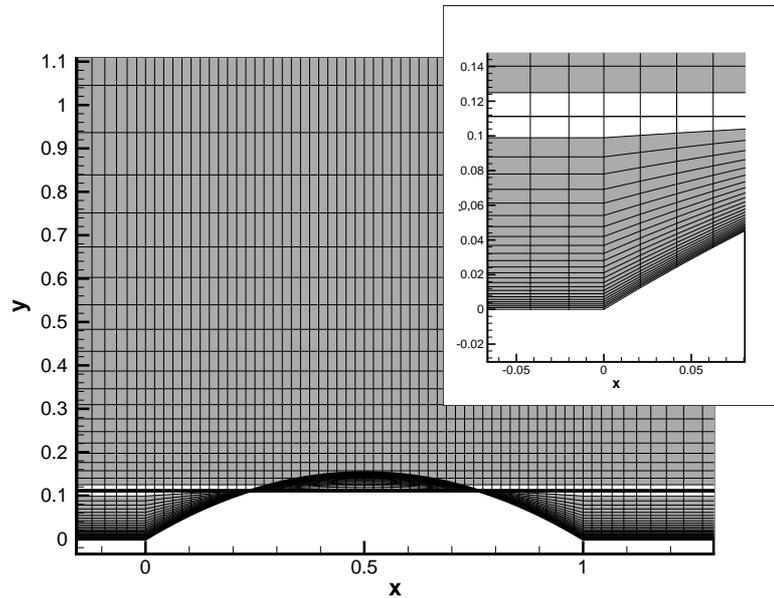


Abbildung 2.4: Darstellung der Deformation einer gelenkig gelagerten Platte. Das Gitter wurde deformiert, indem nur Blockgrenzen (dicke Linien) als Teile des fiktiven Rahmentragwerkes herangezogen wurden. Eine sinnvolle Deformation des fiktiven Rahmens findet nicht statt, da zu wenig aussteifende Balken verwendet werden.

gen vollständig beachtet.

Auf diese Weise entsteht ein Rahmen, der ebenfalls als grobes Netz aufgefaßt werden kann, das eine Einteilung des gesamten Gitters in strukturierte Blöcke vornimmt.

Die bei der Bildung des Rahmens verwendeten Balkenelemente haben alle die gleichen Querschnitts- und Materialdaten. Da keine physikalischen Eigenschaften nachgebildet werden müssen, ist die Wahl der Daten weitgehend frei möglich. Durch die Verhältnisse der Biege-, Scher-, Längs- und Torsionssteifigkeiten untereinander kann ein gewisser Einfluß auf das Deformationsverhalten des Gitters ausgeübt werden. Für die dargestellten Beispiele wurde verwendet: $E = G = 10^{10} \text{ N/m}^2$, Scherkoeffizient $\kappa = 0.833$, Querschnitt $A = 0.1 \text{ cm}^2$, Biegeflächenträgheitsmomente $I_{B22} = I_{B33} = 0.25\pi \text{ cm}^4$ und das Torsionsträgheitsmoment $I_T = 0.5\pi \text{ cm}^4$. Es ist auch denkbar, die Steifigkeit der Balkenelemente umgekehrt proportional zu ihrer Länge zu gestalten, sofern wie hier bei der Modellierung der FLOWer-Gitter für jede Kante im Netz ein einzelnes Balkenelement verwendet wird. Dies kann dazu beitragen, die Deformation in Bereichen feiner Diskretisierung besser zu erhalten, so wie es bei der Deformation mittels des Federmodelles [Bat89, DF02] notwendig ist. Die erreichte Qualität der erzeugten Gitter mit konstanten Daten ist jedoch ausreichend gut.

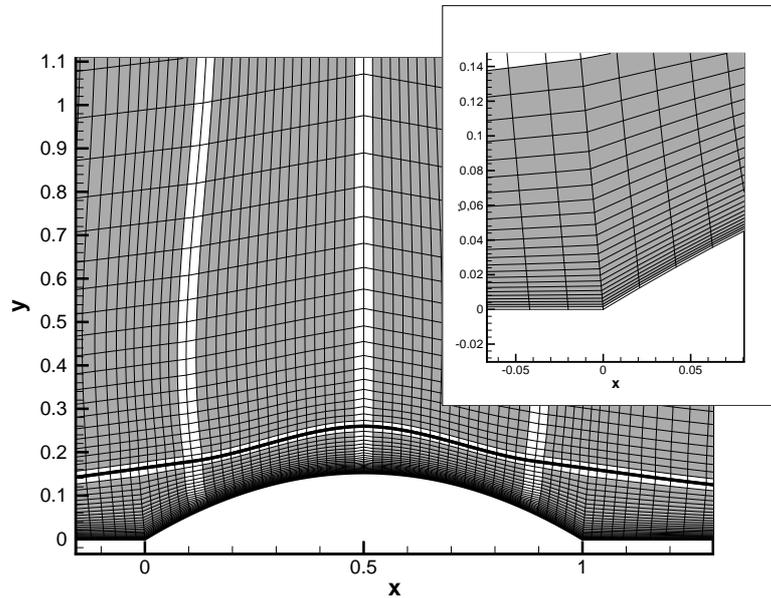


Abbildung 2.5: Analog wie in Abb. 2.4, jedoch wurden zusätzlich 6% der Gitterlinien als aussteifende Elemente des Rahmens verwendet.

Um die Nachführung des Gitters an der Oberfläche des elastischen Körpers möglichst fehlerfrei zu halten, werden zusätzlich alle Punkte auf der Oberfläche des Körpers als Rumpfknoten definiert. Dies erhöht nicht den Rechenaufwand bei der Deformation des zugehörigen Rahmens, sondern dient alleine der genaueren Rekonstruktion der Gitterdeformation in der Nähe des elastischen Festkörpers bei der Ausgabe. Die Entscheidung über die Genauigkeit der Approximation der Deformation der Festkörperoberfläche ist somit aus dem Gitterdeformationsprogramm herausgenommen worden und muß an anderer Stelle entschieden werden.

2.3.2 Interpolation im Blockinneren

Die gewählte Struktur zum Aufbau des Rahmentragwerkes liefert ein analog zur Blockstruktur aufgebautes, grobes Netz, an dem die Deformation bestimmt wurde. Um die Anpassung der übrigen Gitterpunkte zu bestimmen, bietet sich eine Interpolation an. Je nach Lage des Punktes wird eine zwei- oder dreidimensionale Interpolation vorgenommen, die die geometrische Lage des Punktes innerhalb der umgebenden Rahmenbalken gut berücksichtigt.

Liegt der zu bestimmende Knoten in einer Gitterfläche, die von Rahmenbalken eingeschlossen ist, so ist nur eine zweidimensionale Interpolation notwendig. Es wird eine bilineare Interpolation angewendet, bei der eine Wichtung zur Berücksichtigung der geometrischen Lage des zu interpolierenden Punktes innerhalb des Rahmens enthalten ist, die im Folgenden erklärt wird:

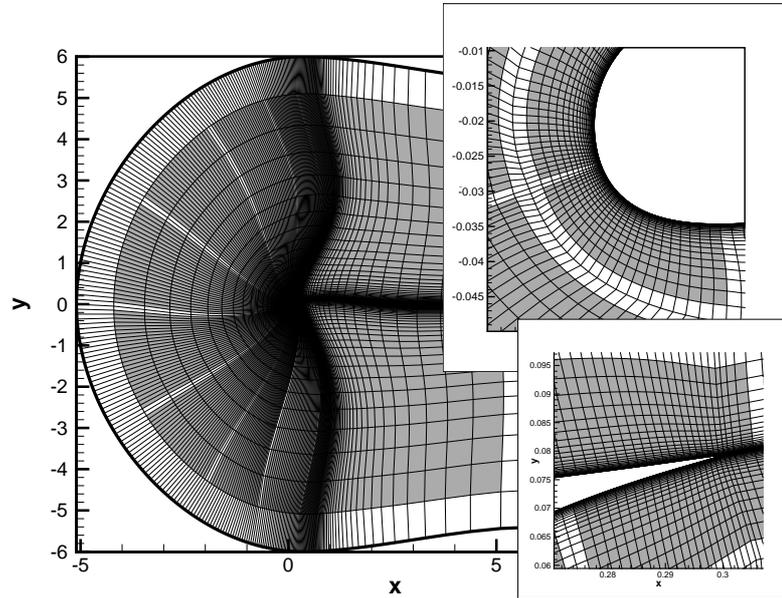


Abbildung 2.6: Flügelpprofil, starr um 20° gedreht, Außenrand des Gitters festgehalten, ein C-Block mit 4% Stützlinien.

Auf den vier begrenzenden Balken sind die gesuchten Werte f_1, f_2, f_3 und f_4 bekannt. Der Wert f an der Kreuzung eines bestimmten Paares von i - und j -Linien ergibt sich aus

$$f = \frac{f_1(1 - I_1)I_4 + f_2I_1I_4}{I_3 + I_4} + \frac{f_3(1 - I_2)I_3 + f_4I_2I_3}{I_3 + I_4} \quad (2.1)$$

mit

$$I_1 = \frac{d_1}{d_2}, \quad I_2 = \frac{d_3}{d_4} \quad \text{und} \quad I_3 = \frac{d_1^n(d_2 - d_1)^n}{(d_2/2)^m}, \quad I_4 = \frac{d_3^n(d_4 - d_3)^n}{(d_4/2)^m}.$$

Die Abstände d_1, \dots, d_4 werden entlang den Gitterlinien bestimmt, indem die Kanten zwischen zwei Knoten als gerade angesehen werden (vgl. Abb. 2.8). Die Faktoren I_3 und I_4 sorgen dafür, daß der geometrische Abstand eines Punktes vom nächsten Rahmenbalken bei der Interpolation berücksichtigt wird. Für den Exponenten n wird in der Regel der Wert 2 oder 3 verwendet, für den Exponenten m Werte zwischen 0 und $2n$. $n = 2$ und $m = 2(n - 1)$ haben sich als günstig erwiesen. Diese Wichtung stellt sicher, daß die erste(n) Schicht(en) neben einem Balken nahezu parallel zu diesem verschoben wird/werden.

Im allgemeinen, dreidimensionalen Fall wird eine analoge trilineare Interpolation in allen drei Raumrichtungen durchgeführt.

In [Wer97, HBB00] werden für die Konstruktion von FLOWER-Netzen bzw. deren Deformation Interpolationsverfahren vorgestellt, bei denen die Rolle

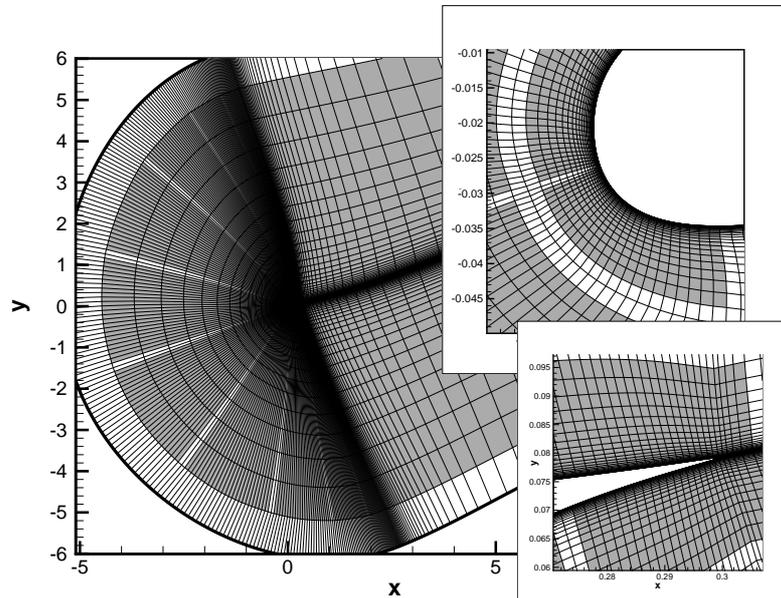


Abbildung 2.7: Flügelprofil, starr um 20° gedreht, Außenrand des Gitters mitbewegt, ein C-Block mit 4% Stützlinien.

der Faktoren I_3 und I_4 von sogenannten „Blending“-Funktionen in der Gestalt entsprechend ausgewählter Polynome höherer Ordnung übernommen wird.

2.3.3 Problemfälle

Neben den bereits erwähnten geometrisch langgestreckten Gitterblöcken, können auch entstehende Knicke bei größeren Winkeln Probleme bei der Gitterdeformation verursachen.

Im ersteren Fall kann Abhilfe geschaffen werden, indem die Einteilung des Gitters in Blöcke ggf. verändert wird. Ideal sind Blöcke mit geometrisch etwa gleichen Kantenlängen. Theoretisch denkbar sind für die Zukunft auch adaptive Methoden bei der Auswahl entsprechender Netzlinien als Bestandteile des elastischen Rahmens.

Bei Knicken, wie sie z.B. am Ende einer gelenkig gelagerten Platte auftreten können, kann es Probleme mit der Überschneidung von Gitterlinien oder auch dem Auseinanderziehen der Linien nahe der Oberfläche geben, was bei der Lösung der Navier-Stokes-Gleichungen unerwünscht ist. Diese Schwierigkeit tritt auch bei der Konstruktion von Offsetflächen auf [Bra92]. Sie kann wie dort durch eine geeignete Wahl von Blockgrenzen bzw. der Dichte virtueller Balken innerhalb der Blöcke i.allg. jedoch umgangen werden.

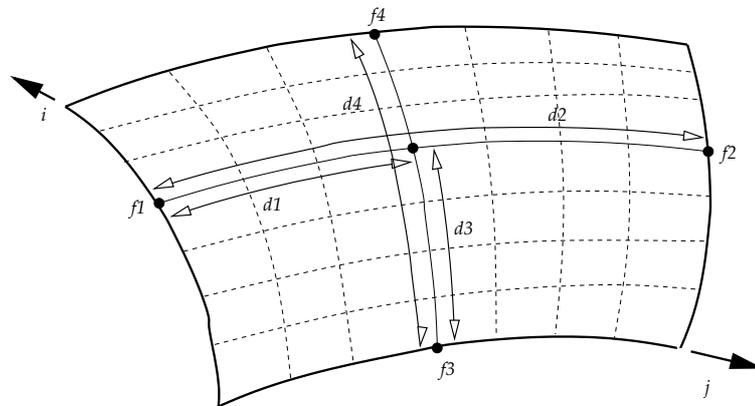


Abbildung 2.8: Skizze zur zweidimensionalen Interpolation.

2.4 Beispiele

2.4.1 Ebene Platte mit Gelenk (2D)

An diesem einfachen Beispiel ist gut zu erkennen, daß die Gitterlinien auch nach der Deformation noch weitgehend orthogonal zueinander verlaufen. Insbesondere in der Nähe der Platte bleibt die Orthogonalität sehr gut gewährleistet. Jedoch kann man auch das bereits angesprochene Problem mit dem Auseinanderziehen der Grenzschicht in der Nähe des Gelenks beobachten. Da es sich um ein Gitter mit einem einzigen Block handelt, wurde versucht, diesen Effekt mit der Wahl verschiedener Balkendichten innerhalb des Blockes abzuschwächen (s. Abb. 2.9).

2.4.2 Rakete (3D)

Abb. 2.10 zeigt ein Beispielgitter einer einfachen zylindrischen Rakete. Dieses Gitter besteht aus 10 Blöcken. Das fiktive Rahmentragwerk ist an den dicken Linien zu erkennen. In diesem Fall wurden nur die Blockgrenzen in das Rahmenmodell aufgenommen (und alle Punkte auf der Oberfläche des Flugkörpers als Rumpfknoten, s.o.).

Der an der Spitze in einem Kugelgelenk festgehaltene Flugkörper wurde um 10 Grad verdreht. Da das Gitter in diesem Fall sehr dicht an den Flugkörper heranreicht, mußte der Außenrand des Gitters ebenfalls beweglich sein. Es ist gut zu sehen, wie sich das Gitter mit dem Flugkörper zusammen bewegt. Entsprechend sieht man in Abb. 2.11, wie das Gitter dem um einen Durchmesser nach oben gebogenen Bug des Flugkörpers folgt. Abb. 2.12 zeigt einen Detailausschnitt des Gitters am Bug, in dem man gut erkennen kann, wie wenig das Gitter in Körperrnähe deformiert wird.

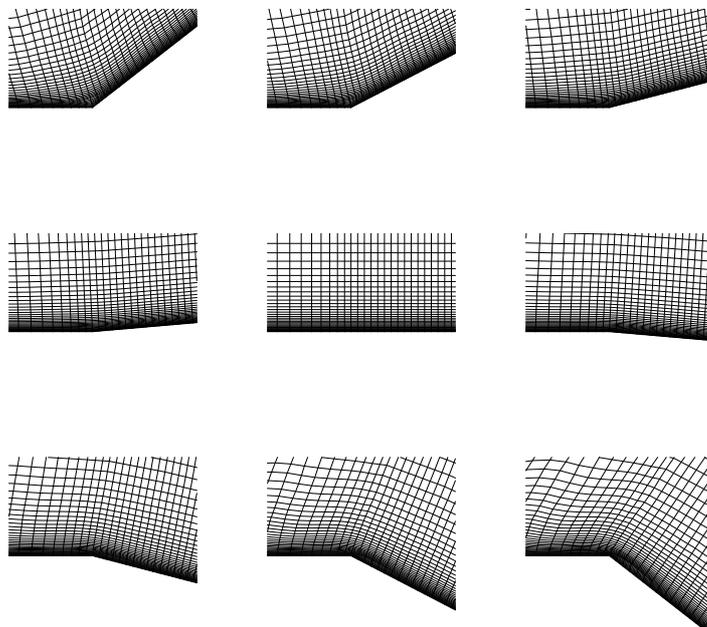


Abbildung 2.9: Deformation eines Gitters über einer Platte mit Gelenk bei verschiedenen Winkeln zwischen $+45^\circ$ und -45° .

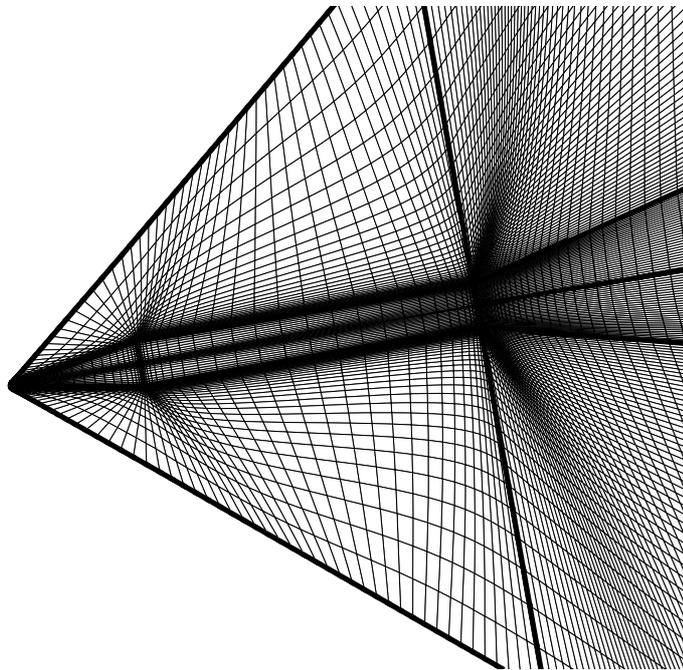


Abbildung 2.10: Ansicht der mittleren Ebene des Gitters. Der Flugkörper wurde am Bug in einem Kugelgelenk gelagert und ist in dieser Ansicht um 10° starr darum gedreht. Dick eingetragene Linien stellen FE-Balken des elastischen Balkenrahmens außerhalb der Körperoberflächen dar.

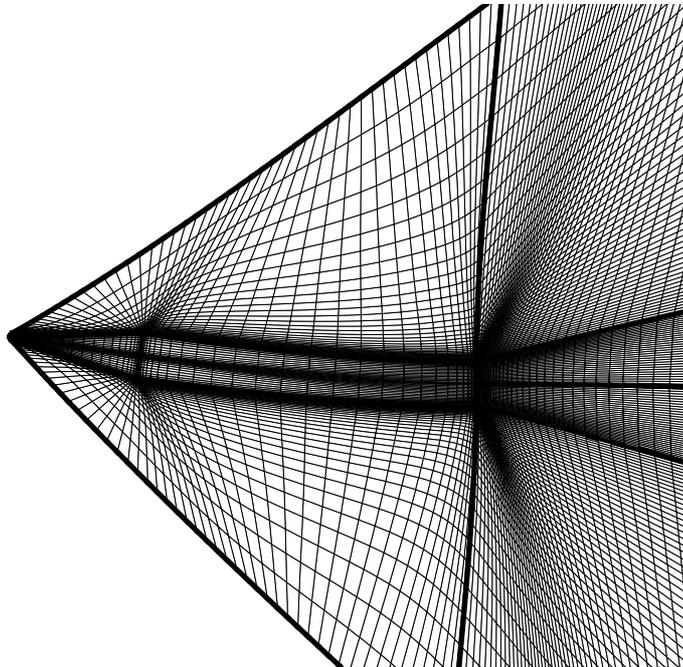


Abbildung 2.11: Ansicht analog Abb. 2.10, jedoch wurde der Flugkörper hier am Heck eingespannt und am Bug um einen Durchmesser in Querrichtung nach oben ausgelenkt.

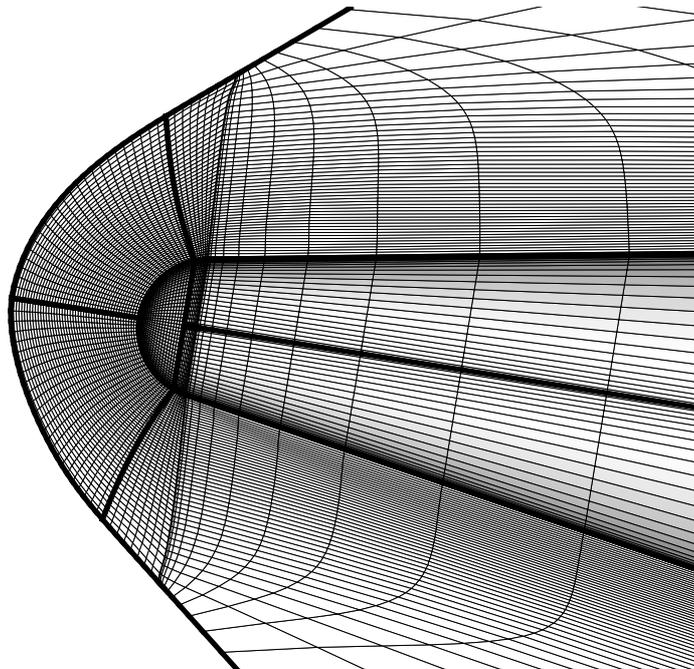


Abbildung 2.12: Ausschnitt des deformierten Gitters aus Abb. 2.11 in Bugnähe.

Kapitel 3

Modellierung des elastischen Festkörpers

Für die Modellierung des elastischen Flugkörpers wurde eine Finite-Elemente-Formulierung gewählt, da diese besonders flexibel an unterschiedliche Anwendungen anzupassen ist. Für die während der Erstellung dieser Arbeit betrachteten Beispiele wurde eine reduzierte Modellierung der elastischen Struktur gewählt, bei der ausschließlich Balkenelemente zum Einsatz kommen. Das implementierte Modell ist jedoch sehr einfach um andere Elementtypen, z.B. Schalenelemente, zu erweitern, um so komplexere Strukturen darstellen zu können.

3.1 FE und Timoshenko-Balken

Für die Darstellung der Balkenelemente wurde eine Formulierung nach dem Modell von Timoshenko verwendet. Der Timoshenko-Balken berücksichtigt im Gegensatz zur weit verbreiteten Bernoulli-Hypothese, die Scherstarrheit voraussetzt, so daß Balkenquerschnitte ständig senkrecht zur Achse bleiben, auch die Scherung des Balkens als von der Biegung unabhängigen Freiheitsgrad. Zusätzlich wird die Drehträgeit der Balkenquerschnitte modelliert. Die Verwölbung der Querschnitte wird also nicht berücksichtigt, indem die Scherung ebenso über den ganzen Querschnitt gleichverteilt angenommen wird. Damit wird am Querschnittsrand die Schubspannungsfreiheit scheinbar verletzt. Dieses wird durch einen Faktor bewerkstelligt, der die „richtige“ Schubspannungsverteilung in eine äquivalente konstante Verteilung umwandelt, d.h. ebene Querschnitte bleiben unter Deformation eben, stehen jedoch gegebenenfalls nicht mehr senkrecht auf der deformierten Balkenachse (vgl. Abb. 3.1). Damit wird vor allem eine korrekte Wellenausbreitungsgeschwindigkeit im Balken erreicht, die bei der Simulation dynamischer Vorgänge von Bedeutung sein kann [Lau90].

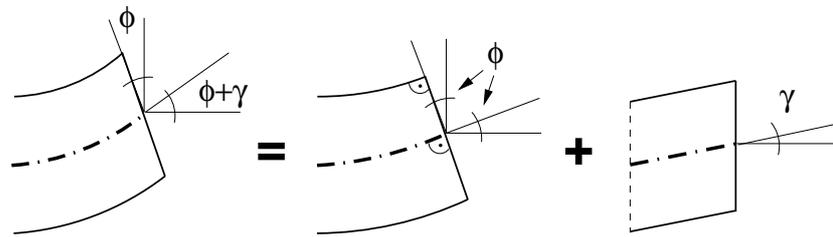


Abbildung 3.1: Überlagerung von Biegung ϕ und Scherung γ beim Timoshenko-Balken

3.1.1 Friedman-Kosmatka-Element

In [FK93] wird ein Finites Element für den Timoshenko Balken vorgestellt, das unter Verwendung von Hermite-Polynomen 2. und 3. Grades für jeweils die Rotations- und Translationsfreiheitsgrade besonders gute Approximationseigenschaften aufweist. Die Autoren stellen unter Verwendung des Hamilton-Prinzips und unter Beachtung der homogenen Differentialgleichungen der Statik des Timoshenko-Balkens ein Zwei-Knoten-Element auf, das frei von „shear-locking“¹ ist. Das Element weist außerdem einen ausgesprochen geringen Diskretisierungsfehler auf, indem die Deformation eines kurzen Balkens bei der Verwendung von nur einem Element bereits exakt wiedergegeben werden und der Fehler bei der Berechnung von Einspannreaktionen und Eigenschwingungen geringer ist als bei vielen anderen Elementen für den Timoshenko Balken. Zudem ist dieses Element letztlich, ähnlich wie das von [Mar84] eingeführte Element, als Erweiterung des üblichen Balkenelements für die Bernoulli-Formulierung [Sch80] zu betrachten, aber in der Wahl der Hermite-Polynome für die Ansatzfunktionen einen Grad niedriger.

3.1.2 Validierung mittels Beispielproblem

Anhand eines einfachen wellenmechanischen Beispielproblems, für das eine gute Referenzlösung existiert, wird hier die gute Qualität des Friedman-Kosmatka-Elements gezeigt. Als Beispiel wird ein zweifach gelagerter, elastischer Stahlbalken betrachtet. Für den Balken gelten folgende Daten:

$$\begin{aligned}
 l &= 1\text{m} & b &= 0,05\text{m} \\
 h &= 0,1\text{m} & G &= 8 \cdot 10^{10} \frac{\text{N}}{\text{m}^2} \\
 E &= 2,1 \cdot 10^{11} \frac{\text{N}}{\text{m}^2} & \kappa &= 0,833 \\
 \rho &= 7850 \frac{\text{kg}}{\text{m}^3}
 \end{aligned}$$

¹Mit „locking“ wird ein Phänomen bezeichnet, bei dem durch die Wahl des Diskretisierungsansatzes eine unnatürliche Steifigkeit oder Bindung eingeführt wird. Wenn es auftritt, ist die Scherung regelmäßig davon betroffen.

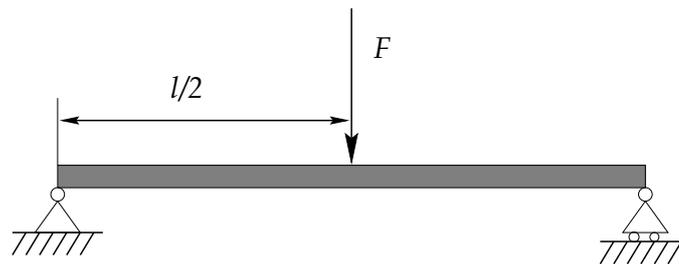


Abbildung 3.2: Darstellung des Beispielproblems

Auf den Balken wird zum Zeitpunkt $t = 0$ nach Art einer Heaviside-Funktion plötzlich eine Kraft $F = 1000\text{N}$ aufgebracht, die bis zum Zeitpunkt $t_0 = 19,8\mu\text{s}$ andauernd wirkt und dann ebenso sprunghaft wieder entfernt wird. Der Belastungsvorgang ist mit einem stark idealisierten Hammerschlag zu vergleichen.

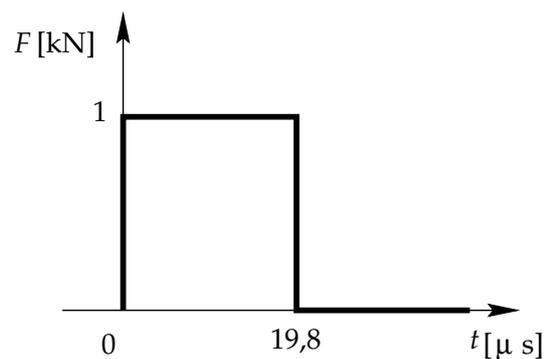


Abbildung 3.3: Kraftverlauf des idealisierten Hammerschlages

Mithilfe des Finite-Elemente-Paketes FE1t [GA], in dem das Friedman-Kosmatka-Element für ebene Deformationen implementiert ist, wurden zwei Modelle erstellt. Der Balken wurde jeweils in 100 gleichlange Elemente unterteilt. Beim ersten Modell wurden Bernoulli-Balkenelemente verwendet, beim zweiten Timoshenko-Elemente. Die Berechnung wurde jeweils mit einer Zeitschrittweite von $\Delta t = 1,98\mu\text{s}$ für 10000 Schritte durchgeführt.

Zum Vergleich wurde eine Referenzlösung für dasselbe Problem mittels eines Charakteristikenverfahrens (Programm rotor) [Lau90] durchgeführt. Durch eine Besonderheit in der Formulierung — für jede Wellenart wird mit CFL=1 gerechnet — ist dieses Charakteristikenverfahren im Gegensatz zu dem von [Pav79] verwendeten Charakteristikenverfahren, das von [Mar84] zum Vergleich benutzt wurde, nahezu vollständig frei von numerischen Dissipations- und Dispersionseffekten. Hier mußte aus programmtechnischen Gründen die Belastung als Streckenlast formuliert werden. Die Kraft wurde dazu auf einer Breite von 2% der Balkenlänge verteilt aufgebracht. Es wurden 100000 Zeitschritte mit $\Delta t = 0,198\mu\text{s}$ berechnet. Die Einteilung des Balkens in 100 Elemente ist hierbei nur für die Visualisierung von Bedeutung, da die Elementeinteilungen für die Berechnung abhängig von der Zeitschrittweite be-

stimmt werden. Hier wurden für die Biegewelle 1732 und für die Scherwelle 975 Zellen benötigt.

Ergänzend wurde für die beiden FE-Modelle die statische Biegelinie unter der gleichen Belastung von $F = 1000\text{N}$ ermittelt.

Zeitintegration der finiten Elemente

Die Zeitintegration wird im Finite-Elemente-Paket `FELT` mittels einer Newmark-Integration durchgeführt. Die Massen wurden hier als verteilt angenommen, alternativ kann auch ein „lumped mass“-Modell verwendet werden. Mit der Steifigkeitsmatrix \mathbf{K} , der Massenmatrix \mathbf{M} und dem Kraftvektor \mathbf{F} lautet die zeitlich diskrete Bewegungsgleichung wie folgt:

$$\mathbf{M} \mathbf{a}_{i+1} + \mathbf{K} \mathbf{u}_{i+1} = \mathbf{F}(t_{i+1})$$

Die Knotenverschiebungen \mathbf{u}_{i+1} und -Geschwindigkeiten \mathbf{v}_{i+1} zum Zeitpunkt t_{i+1} werden mit Newmark-Differenzen approximiert.

$$\begin{aligned} \mathbf{u}_{i+1} &= \tilde{\mathbf{u}}_{i+1} + \beta \Delta t^2 \mathbf{a}_{i+1}, \\ \mathbf{v}_{i+1} &= \tilde{\mathbf{v}}_{i+1} + \gamma \Delta t \mathbf{a}_{i+1}, \end{aligned}$$

worin die Prädiktor-Variablen $\tilde{\mathbf{u}}_{i+1}$ und $\tilde{\mathbf{v}}_{i+1}$ wie folgt definiert sind:

$$\begin{aligned} \tilde{\mathbf{u}}_{i+1} &= \mathbf{u}_i + \Delta t \mathbf{v}_i + \Delta t^2 \left(\frac{1}{2} - \beta \right) \mathbf{a}_i, \\ \tilde{\mathbf{v}}_{i+1} &= \mathbf{v}_i + \Delta t (1 - \gamma) \mathbf{a}_i. \end{aligned}$$

Über die Parameter β und γ kann das Verfahren von vollständig explizit (beide = 0) bis vollständig implizit ($\beta = 1/2$ und $\gamma = 1$) eingestellt werden.

Durch Umformen obiger Gleichungen erhält man eine Gleichung, in der nur noch die Verschiebungen \mathbf{u}_{i+1} als unabhängige Variable enthalten sind. Zu lösen ist

$$\mathbf{K}' \mathbf{u}_{i+1} = \mathbf{F}'_{i+1} + \mathbf{F}(t_{i+1})$$

mit

$$\mathbf{K}' = \mathbf{M} + \beta \Delta t^2 \mathbf{K} \quad \text{und} \quad \mathbf{F}' = \mathbf{M} \tilde{\mathbf{u}}_{i+1}.$$

Die hier vorgestellten Berechnungen wurden mit $\beta = 1/4$ und $\gamma = 1/2$ durchgeführt, was auf eine Trapezregel bei der Zeitintegration führt.

Eine genauere Beschreibung des Newmark-Verfahrens in Zusammenhang mit FEM findet sich in [Hug87].

Statische Biegelinie

Die statische Durchbiegung des Timoshenko-Balkens fällt ca. 10% stärker aus als beim Bernoulli-Balken (Abb. 3.4). Die bei der Bernoulli-Hypothese vorausgesetzte Scherstarrheit bewirkt insgesamt eine Steifigkeitserhöhung. Die Biegewinkel ϕ , d.h. die Verdrehung der Querschnitte gegenüber der Ausgangslage, stimmen bei beiden Modellen genau überein (Abb. 3.5).

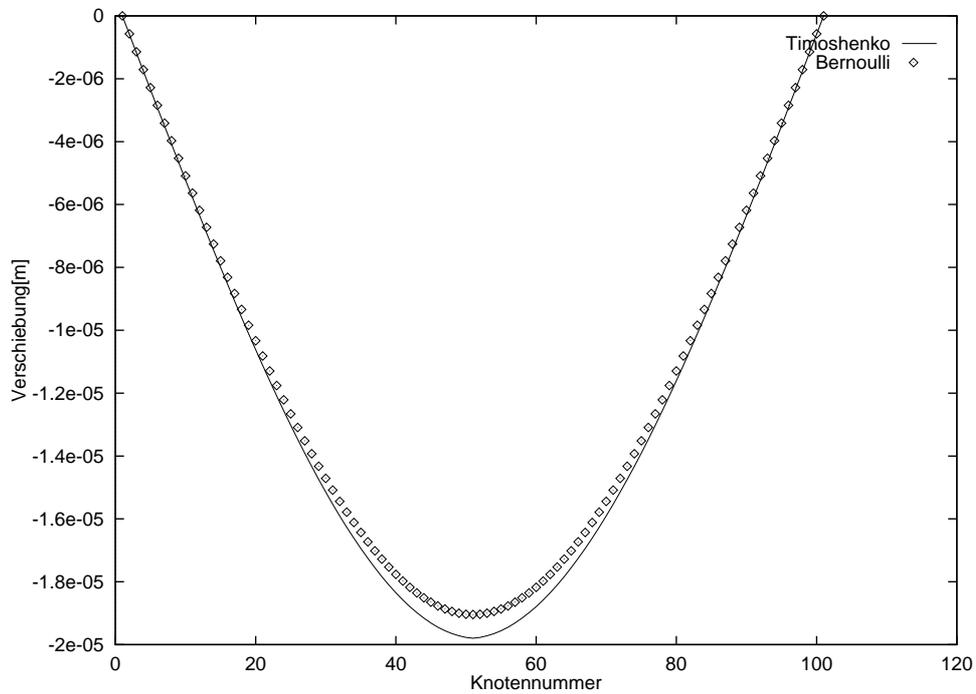


Abbildung 3.4: Statische Biegelinien der beiden FEM-Modelle im Vergleich

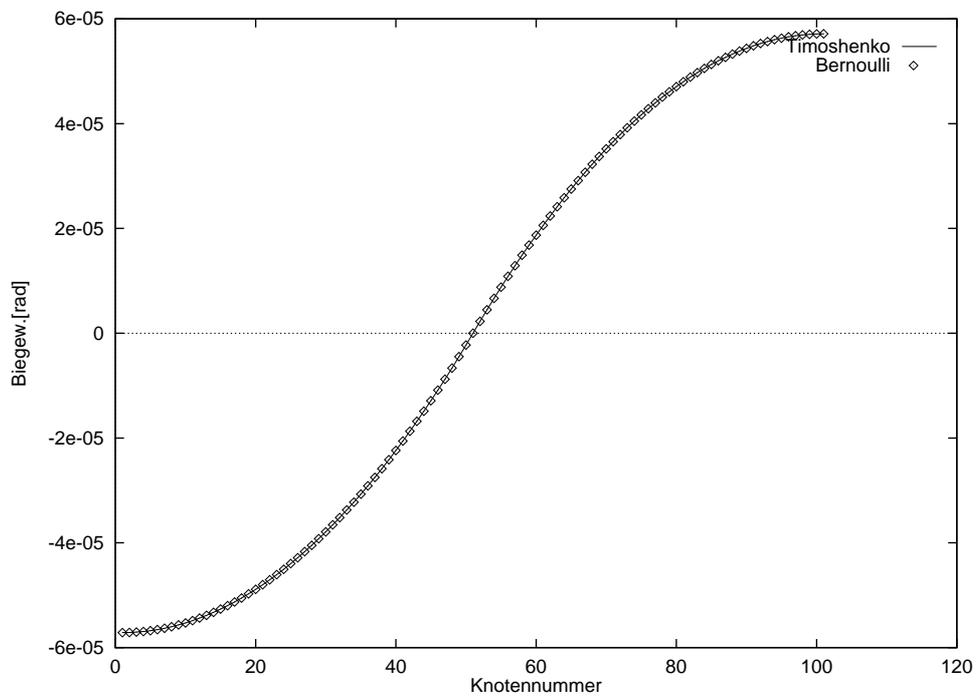


Abbildung 3.5: Statische Biegewinkel der beiden FEM-Modelle im Vergleich

Dynamische Ergebnisse

Betrachtet man nur die Verschiebungen in der Balkenmitte von Bernoulli- und Timoshenko-Balken, so stellt man fest, daß die Grundschiwingung sehr ähnlich ist. Aber auch wenn die Abweichungen in Amplitude und Frequenz nur gering sind, so zeigt sich die höhere Frequenz des Bernoulli-Modells, die einer kleinen Erhöhung der Steifigkeit entspricht, deutlich (Abb. 3.6).

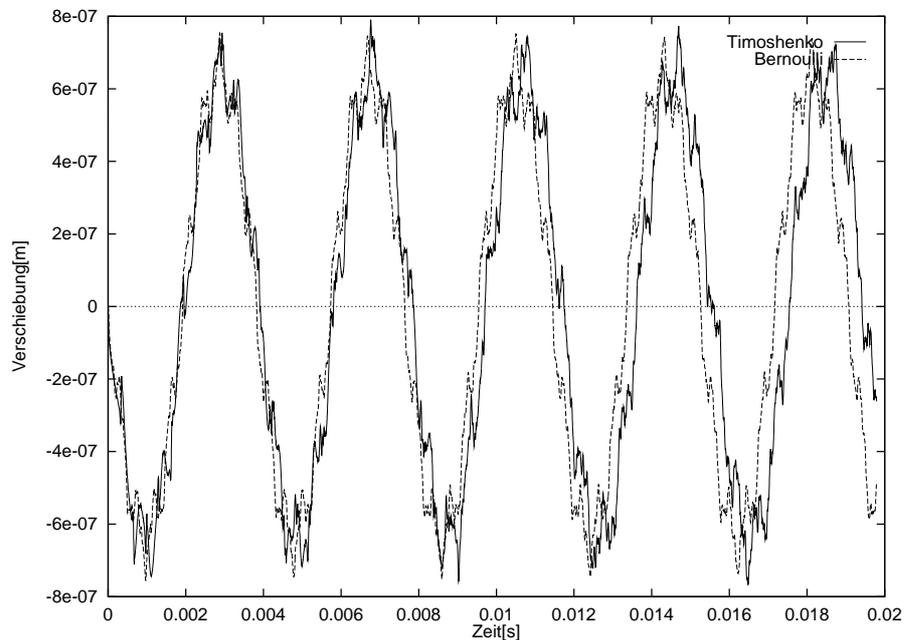


Abbildung 3.6: Verschiebung der Balkenmitte nach den beiden FEM-Modellen im Vergleich. Belastungsende 0,000.019.8 s.

Betrachtet man jedoch die Entwicklung der Biegelinie über die Zeit, so werden die Unterschiede beider Modelle deutlich. Beim Bernoulli-Balken erfolgt die Ausbreitung der Störung mit unendlicher Geschwindigkeit: Direkt nach Beginn der Berechnung sind alle Stützstellen in Bewegung. Der Abhängigkeitsbereich für einen Punkt des Balkens auf einer neuen Zeitebene ist der gesamte Balken der vorherigen Zeitebene. Physikalisch ist dies jedoch nicht richtig, da sich Querwellen in einem Balken mit endlicher Geschwindigkeit ausbreiten. Beim Timoshenko-Balken wird diese Eigenschaft der Ausbreitung von Wellen richtig wiedergegeben. Anfänglich gibt es links und rechts des Hammerschlages noch „Ruhegebiete“, die noch nicht von der Störung betroffen sind (s. Abb. 3.7–3.9). Die Darstellung der Schnittgrößen Biegemoment und Querkraft wären ein schärferer Vergleich [Mar84, Pav79], jedoch war deren Ausgabe im verwendeten FE-Programm nicht vorgesehen.

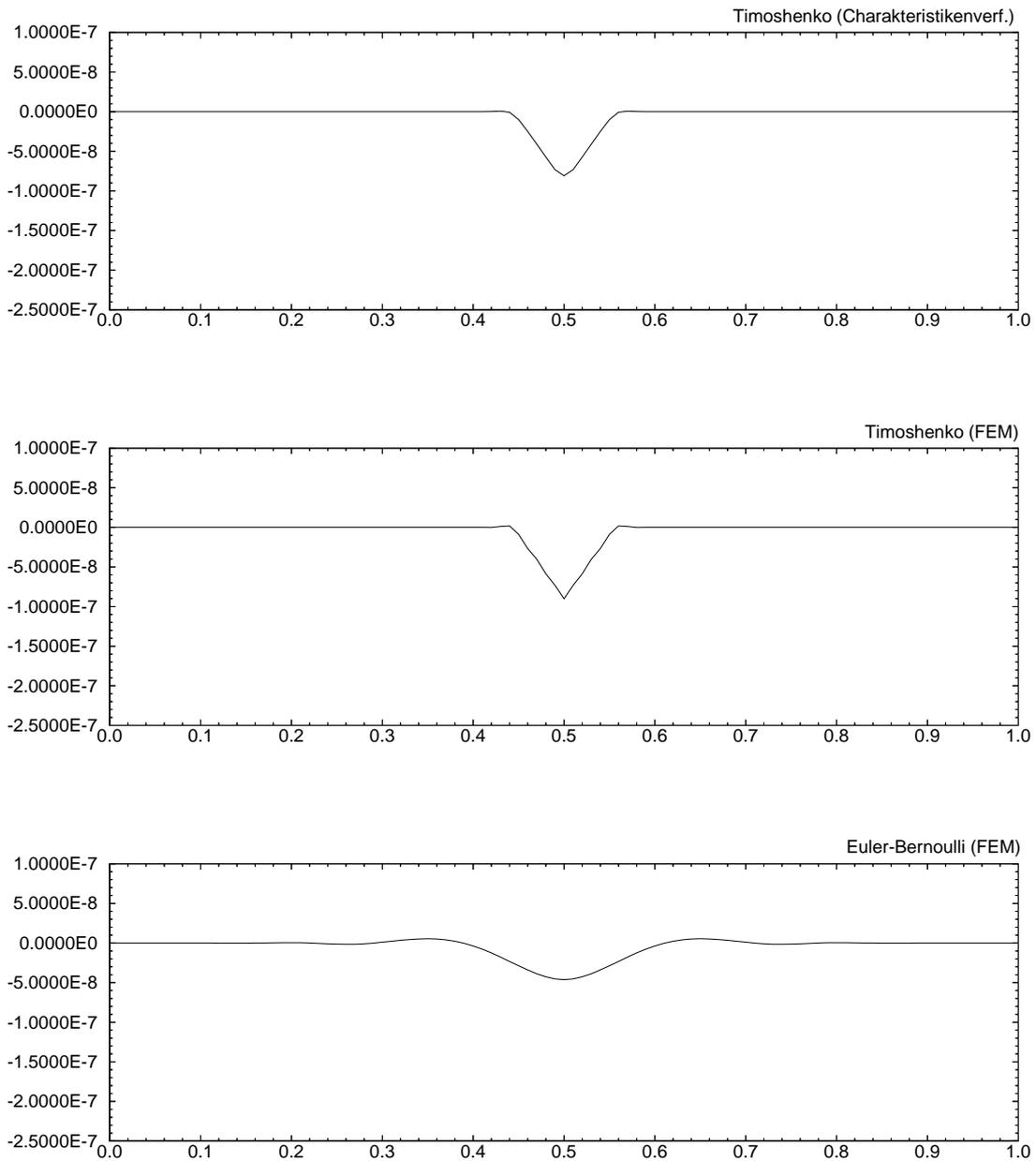


Abbildung 3.7: Momentane Biegelinie des Balkens gemäß drei Modellen/Verfahren nach $19,8 \mu\text{s}$ (Ende der Belastung durch den Hammerschlag).

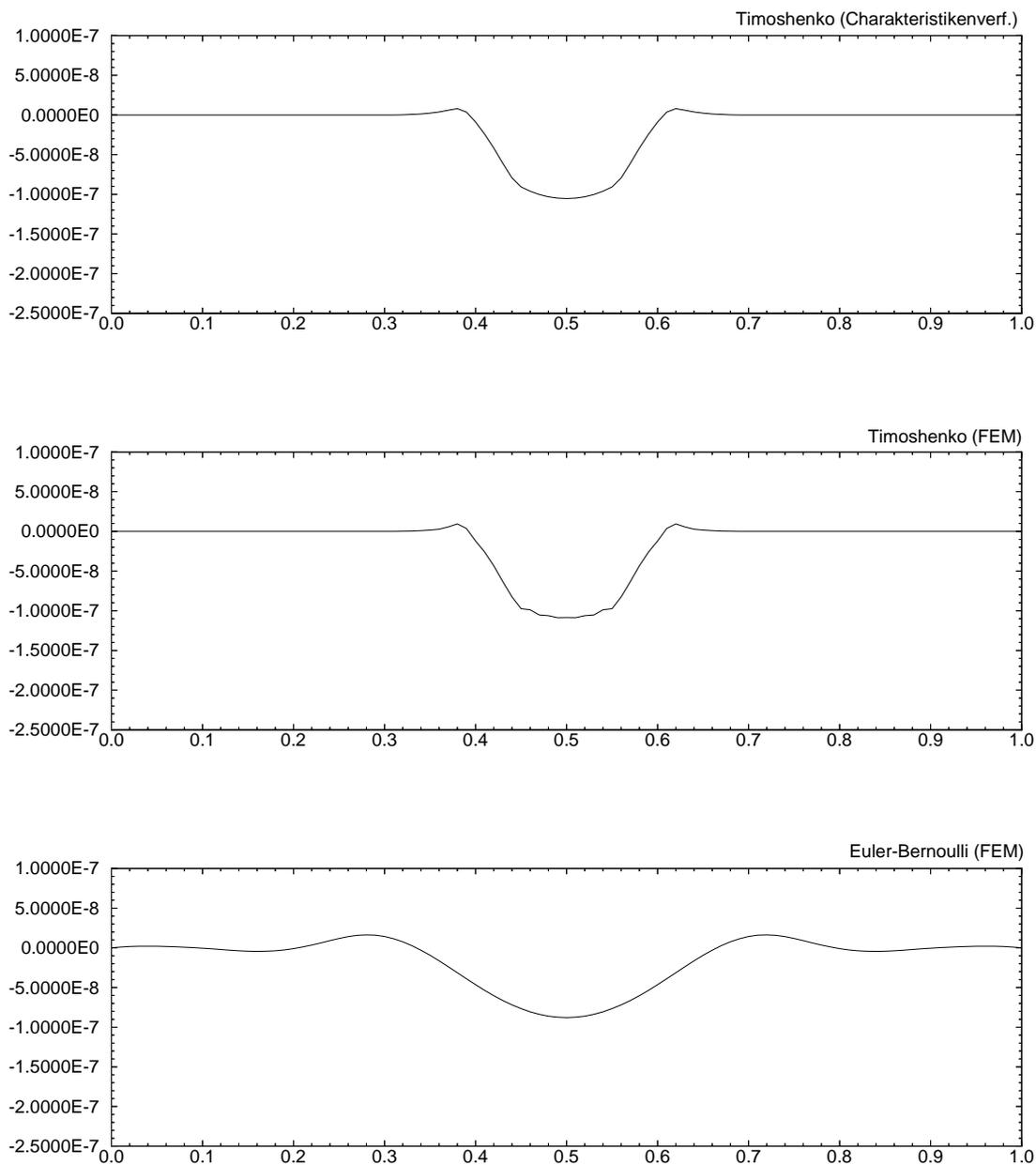


Abbildung 3.8: Momentane Biegelinie des Balkens gemäß drei Modellen/Verfahren nach $39,6\mu\text{s}$.

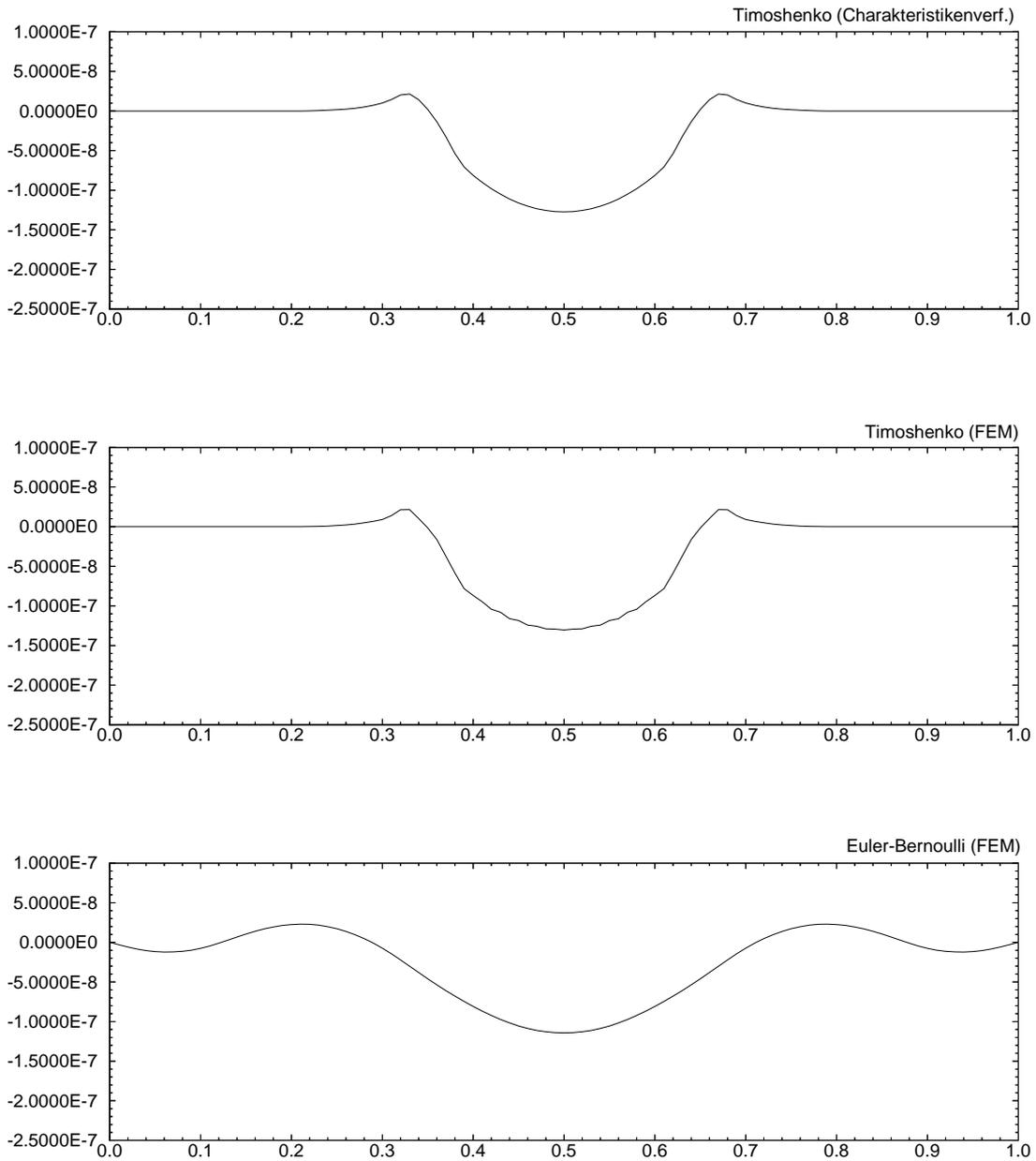


Abbildung 3.9: Momentane Biegelinie des Balkens gemäß drei Modellen/Verfahren nach $59,4 \mu\text{s}$.

Fazit

Das Beispielproblem zeigt, daß alle Modelle bzw. Verfahren geeignet sind, um (quasi-)statische oder stationäre Probleme wie die Berechnung der statischen Biegelinie oder der Frequenz und Form der Grundschiwingung durchzuführen. Sollen jedoch die Auswirkungen einer eingeleiteten Störung in Raum und Zeit genau untersucht werden, so ist das Bernoulli-Modell ungeeignet. Die Ausbreitung der Störung wird dagegen vom Timoshenko-Modell physikalisch richtig beschrieben. Insbesondere können bei der zeitlichen Dauer der durchgeführten Berechnungen die Charakteristiken- und die Finite-Elemente-Methode mit dem Friedman-Kosmatka-Element und stabiler Zeintintegration als gleichwertig angesehen werden.

3.2 Erweitertes Balkenelement für die Aeroelastik

Für die Modellierung von Flügelprofilen, durch deren komplexen inneren Aufbau die Zentren und Hauptachsen für Masse, Biegung und Torsion nicht mehr zusammenfallen, ist eine Erweiterung des Friedman-Elementes im Sinne von [Lau90] notwendig. In [Lau90] wird eine Charakteristiken-Formulierung für einen Balken vorgestellt, bei dem die Torsions- oder Massenlinie gegebenenfalls nicht mit der Biegelinie zusammenfallen, die im Folgenden in das FE-Modell mit dem Ansatz von [FK93] übertragen wird.

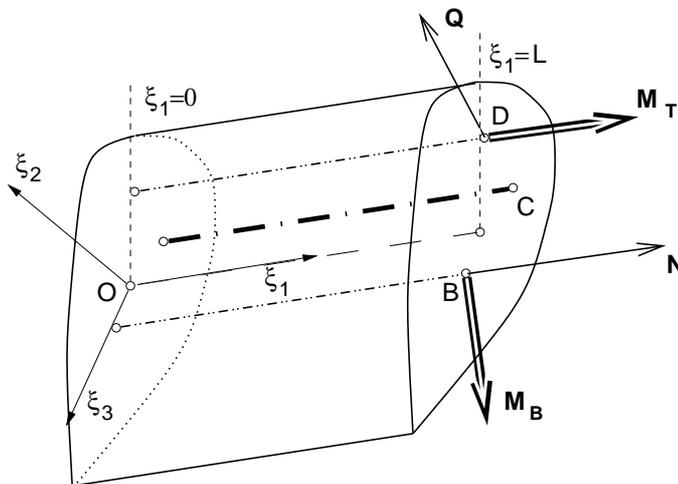


Abbildung 3.10: Balkenelement als Tragflügelsektion: Bezugsachse ξ_1 , Massenlinie C, Drillzentrum D und Biegezentrum B.

In Abb. 3.10 ist ein solches Balkenelement mit der Länge L beispielhaft dargestellt. Das Koordinatensystem $O(\xi) = O(\xi_1, \xi_2, \xi_3)$ beschreibt die materiellen Koordinaten in den jeweiligen Balkenquerschnitten in der aktuellen, d.h. deformierten Konfiguration (a.K.). Als Bezugsachse für den Balken wird die ξ_1 -Achse verwendet ($\xi_2 = \xi_3 = 0$). Die ξ_i sind materielle Koordinaten und deshalb bei der Bewegung konstant. Das Koordinatensystem $O(\xi)$ für den

nicht deformierten Balken (Bezugsconfiguration, BK.) wird mit $O(\mathbf{x})$ bezeichnet, die zugehörigen Basisvektoren sind $\mathbf{e}_1, \mathbf{e}_2$ und \mathbf{e}_3 .

Für den jeweiligen Balkenquerschnitt in der a.K. wird ein lokales Koordinatensystem eingeführt, indem ein Einheitsvektor \mathbf{e}_{S1} senkrecht auf dem materiellen Querschnitt definiert wird und \mathbf{e}_{S2} entlang ξ_2 und \mathbf{e}_{S3} entlang ξ_3 diesen zu einer Orthonormalbasis $\mathbf{e}_{Si} = \mathbf{e}_{Si}(\xi, t)$ ($i = 1, 2, 3$) ergänzen. Wegen der zugelassenen Scherverformung ist \mathbf{e}_{S1} i. allg. nicht parallel zum Tangentenvektor der verformten Balkenachse an der entsprechenden Stelle.

Die Wirkungslinie der Normalkraft $\mathbf{N} = N\mathbf{e}_{S1}$ verläuft, wenn keine Längskraftbiegung auftritt, durch das Biegezentrum B, die der Querkraft $\mathbf{Q} = Q_2\mathbf{e}_{S2} + Q_3\mathbf{e}_{S3}$ durch das Drillzentrum D, wenn keine Verdrillung auftritt. Dementsprechend wird das Biegemoment $\mathbf{M}_B = M_{B2}\mathbf{e}_{S2} + M_{B3}\mathbf{e}_{S3}$ auf das Biegezentrum B bezogen und das Torsionsmoment $\mathbf{M}_T = M_T\mathbf{e}_{S1}$ auf das Drillzentrum D. Der Massenschwerpunkt des Querschnittes liegt im Punkt C.

Nachfolgend wird angenommen, daß alle Materialeigenschaften und Querschnittsgrößen des Balkens innerhalb eines Elementes entlang seiner Länge als konstant vorauszusetzen sind. D.h., daß auch die Koordinaten ξ_{C2} und ξ_{C3} des Massenzentrums und entsprechend die der Punkte D (ξ_{Di}) und B (ξ_{Bi}) bzgl. ξ_1 konstant sind. In der Praxis auftretende, nicht konstante Daten können bei der Diskretisierung in der Regel durch eine Abfolge von Elementen mit jeweils konstanten Daten, d.h. mit Sprüngen an den Übergängen der Elemente, gut dargestellt werden.

3.2.1 Bewegungsgleichungen

Die Bewegungsgleichungen werden mit Hilfe der Bilanzen für Impuls und Drall eines materiellen Balkenquerschnitts und der Schnittkraft-Verformungsbeziehungen gewonnen.

Bilanzgleichungen

Die Impulsbilanz lautet

$$\rho A^{(\rho)} \mathbf{a}_C = \frac{\partial}{\partial \xi_1} (\mathbf{N} + \mathbf{Q}) \quad (3.1)$$

mit der Beschleunigung des Massenzentrums C

$$\mathbf{a}_C = \mathbf{a} + \dot{\boldsymbol{\omega}} \wedge \sum_{\alpha=2,3} \xi_{C\alpha} \mathbf{e}_\alpha.$$

Dabei bezeichnet $\mathbf{a} = \ddot{\mathbf{u}}$ den Beschleunigungsvektor auf der Bezugsachse ξ_1 und $\boldsymbol{\omega} = \dot{\boldsymbol{\varphi}}$ mit $\boldsymbol{\varphi} = \varphi_T \mathbf{e}_1 + \sum_{\alpha=2,3} \varphi_{B\alpha} \mathbf{e}_\alpha$ (Torsionswinkel φ_T , Biegewinkel φ_{Bi} , $i = 2, 3$) die Winkelgeschwindigkeit.

Die Drallbilanz bezogen auf das Massenzentrum C lautet

$$\begin{aligned} \rho \mathbf{I}_C^{(\rho)} \dot{\boldsymbol{\omega}} &= \frac{\partial}{\partial \xi_1} (\mathbf{M}_T + \mathbf{M}_B) + \\ &\frac{\partial}{\partial \xi_1} (\mathbf{r}_{CD} \wedge \mathbf{Q} + \mathbf{r}_{CB} \wedge \mathbf{N}) + \\ &\left(\mathbf{e}_1 + \frac{\partial}{\partial \xi_1} (\mathbf{r}_{CB} + \mathbf{u}) \right) \wedge (\mathbf{N} + \mathbf{Q}) \end{aligned} \quad (3.2)$$

mit $\mathbf{r}_* = \sum_{\alpha=2,3} \xi_{*\alpha} \mathbf{e}_{S\alpha}$, $\mathbf{r}_{CD} = \mathbf{r}_D - \mathbf{r}_C$ und $\mathbf{r}_{CB} = \mathbf{r}_B - \mathbf{r}_C$.

Nach der Wahl der Vergleichsdichte ρ kann mit der Wichtungsfunktion

$$n^{(\rho)}(\xi_2, \xi_3) = \frac{\rho'(\xi_2, \xi_3)}{\rho}$$

die Fläche $A^{(\rho)}$ und der Flächenträgheitstensor $\mathbf{I}_C^{(\rho)}$ bzgl. C des trägen Querschnittes durch Integration über den gesamten Querschnitt A bestimmt werden.

$$A^{(\rho)} = \int_A n^{(\rho)} d\xi_2 d\xi_3$$

Die Lage des Massenschwerpunktes C ist bestimmt durch

$$\mathbf{r}_C = \frac{1}{A^{(\rho)}} \int_A \mathbf{r} n^{(\rho)} d\xi_2 d\xi_3.$$

Der Tensor der Flächenträgheitsmomente des trägen Querschnittes $A^{(\rho)}$ auf die Balkenachse ξ_1 bezogen ist

$$\mathbf{I}^{(\rho)} = \int_A ((\mathbf{r}\mathbf{r})\mathbf{1} - \mathbf{r} \circ \mathbf{r}) n^{(\rho)} d\xi_2 d\xi_3$$

mit dem Einheitstensor $\mathbf{1} = \sum_{\alpha=1,2,3} \mathbf{e}_{S\alpha} \circ \mathbf{e}_{S\alpha}$. Nach dem Satz von Steiner erhält man

$$\mathbf{I}_C^{(\rho)} = \mathbf{I}^{(\rho)} - A^{(\rho)} ((\mathbf{r}_C \mathbf{r}_C)\mathbf{1} - \mathbf{r}_C \circ \mathbf{r}_C).$$

Schnittkraft–Deformations–Beziehungen

Analog zum trägen Querschnitt ergeben sich durch die Wahl eines Vergleichselastizitätsmoduls E die Ersatzquerschnittsfläche $A^{(E)}$, die Lage des Biegezenters B und der Flächenträgheitstensor des tragenden elastischen Querschnittes $\mathbf{I}_B^{(E)}$ bzgl. B mit der Wichtungsfunktion

$$n^{(E)}(\xi_2, \xi_3) = \frac{E'(\xi_2, \xi_3)}{E}.$$

Für den linearelastischen Balken gelten folgende linearisierte Näherungen für die Zusammenhänge zwischen Schnittgrößen und Verformungen:

$$\mathbf{N} = EA^{(E)} u_{B1, \xi_1} \mathbf{e}_{S1}, \quad \mathbf{Q} = GA^{(E)} \mathbf{k}\boldsymbol{\gamma} \quad (3.3)$$

Dabei ist $\boldsymbol{\gamma} = \sum_{\alpha=2,3} \gamma_{\alpha} \mathbf{e}_{\alpha}$ die Scherung und

$$\mathbf{u}_B = \mathbf{u} + \boldsymbol{\varphi} \wedge \sum_{\alpha=2,3} \xi_{B\alpha} \mathbf{e}_{\alpha}$$

die Verschiebung der materiellen Punkte auf der Biegezentrum–Achse B. Der Tensor der Schubbeiwerte \mathbf{k} ist symmetrisch und wird gebildet, indem die Schubbeiwerte für die Hauptachsen des Flächenträgheitstensors $\mathbf{I}_B^{(E)}$ bestimmt werden und diese dann auf die aktuellen Bezugsrichtungen transformiert werden, d.h. \mathbf{k} und $\mathbf{I}_B^{(E)}$ besitzen die gleichen Hauptachsen. Für die spätere Anwendung ist es zweckmäßig, die Scherung durch die Verschiebung der materiellen Punkte auf der elastischen Linie (Drillzentrum D)

$$\mathbf{u}_D = \mathbf{u} + \boldsymbol{\varphi} \wedge \sum_{\alpha=2,3} \xi_{D\alpha} \mathbf{e}_{\alpha} \quad (3.4)$$

und den Biegewinkel φ_{Bi} ($i = 2, 3$) auszudrücken. Damit gilt:

$$\begin{aligned} \gamma_2 &= \mathbf{u}_{2,\xi_1} - (\varphi_T \xi_{D3})_{,\xi_1} - \varphi_{B3} \\ \gamma_3 &= \mathbf{u}_{3,\xi_1} + (\varphi_T \xi_{D2})_{,\xi_1} + \varphi_{B2}. \end{aligned} \quad (3.5)$$

Für die Torsions- und Biegemomente gilt:

$$\mathbf{M}_T = G I_T \varphi_{T,\xi_1} \mathbf{e}_{S1}, \quad \mathbf{M}_B = E \mathbf{I}_B^{(E)} \sum_{\alpha=2,3} \varphi_{B\alpha,\xi_1} \mathbf{e}_{S\alpha}. \quad (3.6)$$

Linearisierte Bewegungsgleichungen

Setzt man die Schnittkraft–Verformungsbeziehungen (3.3) und (3.6) in die Bilanzgleichungen (3.1) und (3.2) ein und ersetzt darin die Scherung gemäß (3.5), so erhält man für die Impulsbilanz

$$\begin{aligned} \rho A^{(\rho)} \mathbf{a}_C &= E A^{(E)} \frac{\partial}{\partial \xi_1} ((\mathbf{u}_{1,\xi_1} + \varphi_{B2,\xi_1} \xi_{B3} - \varphi_{B3,\xi_1} \xi_{B2}) \mathbf{e}_{S1}) + \\ &G A^{(E)} k_{22} \frac{\partial}{\partial \xi_1} ((\mathbf{u}_{2,\xi_1} - \varphi_{T,\xi_1} \xi_{D3} - \varphi_{B3}) \mathbf{e}_{S2}) + \\ &G A^{(E)} k_{33} \frac{\partial}{\partial \xi_1} ((\mathbf{u}_{3,\xi_1} + \varphi_{T,\xi_1} \xi_{D2} + \varphi_{B2}) \mathbf{e}_{S3}) \end{aligned} \quad (3.7)$$

und für die Drallbilanz

$$\begin{aligned} \rho \mathbf{I}_C^{(\rho)} \dot{\boldsymbol{\omega}} &= G \frac{\partial}{\partial \xi_1} \left((I_T \varphi_{T,\xi_1} + A^{(E)} (\xi_{CD2} k_{33} \gamma_3 - \xi_{CD3} k_{22} \gamma_2)) \mathbf{e}_{S1} \right) + \\ &E \frac{\partial}{\partial \xi_1} \left((I_{B22}^{(E)} \varphi_{B2,\xi_1} + A^{(E)} \xi_{CB3} \mathbf{u}_{1,\xi_1}) \mathbf{e}_{S2} \right) + \\ &E \frac{\partial}{\partial \xi_1} \left((I_{B33}^{(E)} \varphi_{B3,\xi_1} - A^{(E)} \xi_{CB2} \mathbf{u}_{1,\xi_1}) \mathbf{e}_{S3} \right) + \\ &A^{(E)} \left(\mathbf{e}_1 + \sum_{\alpha=2,3} \xi_{CB\alpha} \mathbf{e}_{S\alpha,\xi_1} + \mathbf{u}_{,\xi_1} \right) \wedge \\ &\left(E \mathbf{u}_{1,\xi_1} \mathbf{e}_{S1} + G \sum_{\alpha=2,3} k_{\alpha\alpha} \gamma_{\alpha} \mathbf{e}_{S\alpha} \right). \end{aligned} \quad (3.8)$$

Zur Vereinfachung wird hier angenommen, daß die Hauptachsen von $\mathbf{I}_B^{(E)}$ und \mathbf{k} in ξ_2 - und ξ_3 -Richtung liegen, was keine Einschränkung der Allgemeinheit darstellt.

Für kleine Verformungen und Lageänderungen des Balkens können die Gleichungen (3.7) und (3.8) vollständig nach den Verschiebungen \mathbf{u} und Verdrehungen φ sowie deren Ableitungen linearisiert werden. Für die auftretenden partiellen Ableitungen der lokalen Einheitsvektoren \mathbf{e}_{s_i, ξ_1} ($i = 1, 2, 3$) gilt dabei mit $\mathbf{e}_{s_i} = \sum_{\lambda=1,2,3} \alpha_{i\lambda} \mathbf{e}_\lambda$

$$\frac{\partial \mathbf{e}_{s_i}}{\partial \xi_1} = \frac{\partial}{\partial \xi_1} \sum_{\lambda=1,2,3} \alpha_{i\lambda} \mathbf{e}_\lambda = \sum_{\lambda=1,2,3} \alpha_{i\lambda, \xi_1} \mathbf{e}_\lambda$$

mit der Transformationsmatrix

$$\boldsymbol{\alpha} = \begin{bmatrix} 1 & \varphi_{B3} & -\varphi_{B2} \\ -\varphi_{B3} & 1 & \varphi_T \\ \varphi_{B2} & -\varphi_T & 1 \end{bmatrix}$$

für kleine Winkel in linearisierter Näherung.

Somit lauten die vollständig linearisierten Bewegungsgleichungen für den Timoshenko-Balken:

$$\begin{aligned} \rho A^{(\rho)} \mathbf{a}_C &= EA^{(E)} ((u_{1, \xi_1 \xi_1} + \varphi_{B2, \xi_1 \xi_1} \xi_{B3} - \varphi_{B3, \xi_1 \xi_1} \xi_{B2}) \mathbf{e}_1) + \quad (3.9) \\ &GA^{(E)} k_{22} ((u_{2, \xi_1 \xi_1} - \varphi_{T, \xi_1 \xi_1} \xi_{D3} - \varphi_{B3, \xi_1}) \mathbf{e}_2) + \\ &GA^{(E)} k_{33} ((u_{3, \xi_1 \xi_1} + \varphi_{T, \xi_1 \xi_1} \xi_{D2} + \varphi_{B2, \xi_1}) \mathbf{e}_3) \end{aligned}$$

$$\begin{aligned} \rho \mathbf{I}_C^{(\rho)} \dot{\boldsymbol{\omega}} &= G \left(I_T \varphi_{T, \xi_1 \xi_1} + A^{(E)} (\xi_{CD2} k_{33} \gamma_{3, \xi_1} - \xi_{CD3} k_{22} \gamma_{2, \xi_1}) \right) \mathbf{e}_1 \quad (3.10) \\ &+ \left(EI_{B22}^{(E)} \varphi_{B2, \xi_1 \xi_1} + EA^{(E)} \xi_{CB3} u_{1, \xi_1 \xi_1} - GA^{(E)} k_{33} \gamma_3 \right) \mathbf{e}_2 \\ &+ \left(EI_{B33}^{(E)} \varphi_{B3, \xi_1 \xi_1} - EA^{(E)} \xi_{CB2} u_{1, \xi_1 \xi_1} + GA^{(E)} k_{22} \gamma_2 \right) \mathbf{e}_3 \end{aligned}$$

Hier sind die γ_i ($i = 2, 3$) noch gemäß (3.5) zu ersetzen.

3.2.2 Formfunktionen

In [FK93] wird für die ebene Deformation des Timoshenko-Balkens ohne Berücksichtigung der Längsdehnung ein Satz von Formfunktionen vorgestellt, die so gewählt sind, daß der entstehende Funktionenraum stets die stationären, homogenen Gleichgewichtsbedingungen für den Timoshenko-Balken erfüllt. Es entsteht ein Satz exakt integrierbarer Polynome 3. Grades für die Verschiebung bzw. 2. Grades für die Verdrehung, die zu einem Zwei-Knoten-Balkenelement führen, das gegen Scher-Locking unempfindlich ist.

Dieser Ansatz wird hier weiter ausgebaut, indem er auf eine räumliche Deformation inklusive der Längsdehnung und Torsion des Balkens gemäß den weiter oben entwickelten Bewegungsgleichungen erweitert wird. Dazu werden die Formfunktionen des Friedman-Kosmatka-Elements für die Balkenbiegung verwendet, sowie lineare Ansatzfunktionen für Längsdehnung und

Torsion [Sch80]. Für die sechs Formfunktionen des Elementes mit insgesamt 12 Freiheitsgraden werden 10 verschiedene Polynome N_1 bis N_{10} benötigt, die in Tabelle 3.1 wiedergegeben sind.

$$\begin{aligned}
 N_1 &= 1 - \frac{\xi_1}{L} \\
 N_2 &= \frac{\xi_1}{L} \\
 N_3(\phi) &= \frac{1}{(1+\phi)} \left(2 \left(\frac{\xi_1}{L} \right)^3 - 3 \left(\frac{\xi_1}{L} \right)^2 - \phi \left(\frac{\xi_1}{L} \right) + (1+\phi) \right) \\
 N_4(\phi) &= \frac{L}{(1+\phi)} \left(\left(\frac{\xi_1}{L} \right)^3 - \left(2 + \frac{\phi}{2} \right) \left(\frac{\xi_1}{L} \right)^2 + \left(1 + \frac{\phi}{2} \right) \left(\frac{\xi_1}{L} \right) \right) \\
 N_5(\phi) &= -\frac{1}{(1+\phi)} \left(2 \left(\frac{\xi_1}{L} \right)^3 - 3 \left(\frac{\xi_1}{L} \right)^2 - \phi \left(\frac{\xi_1}{L} \right) \right) \\
 N_6(\phi) &= \frac{L}{(1+\phi)} \left(\left(\frac{\xi_1}{L} \right)^3 - \left(1 - \frac{\phi}{2} \right) \left(\frac{\xi_1}{L} \right)^2 - \frac{\phi}{2} \left(\frac{\xi_1}{L} \right) \right) \\
 N_7(\phi) &= \frac{6}{(1+\phi)L} \left(\left(\frac{\xi_1}{L} \right)^2 - \left(\frac{\xi_1}{L} \right) \right) \\
 N_8(\phi) &= \frac{1}{(1+\phi)} \left(3 \left(\frac{\xi_1}{L} \right)^2 - (4+\phi) \left(\frac{\xi_1}{L} \right) + (1+\phi) \right) \\
 N_9(\phi) &= -\frac{6}{(1+\phi)L} \left(\left(\frac{\xi_1}{L} \right)^2 - \left(\frac{\xi_1}{L} \right) \right) \\
 N_{10}(\phi) &= \frac{1}{(1+\phi)} \left(3 \left(\frac{\xi_1}{L} \right)^2 - (2-\phi) \left(\frac{\xi_1}{L} \right) \right)
 \end{aligned}$$

Tabelle 3.1: Liste der Polynome, aus denen die Formfunktionen für das verallgemeinerte Balkenelement gebildet werden. Die Polynome N_3 bis N_{10} stammen aus [FK93].

Die Vektoren der Verschiebung $\mathbf{u}(\xi_1) = [u_1(\xi_1), u_2(\xi_1), u_3(\xi_1)]^T$ und Verdrehung $\boldsymbol{\varphi}(\xi_1) = [\varphi_T(\xi_1), \varphi_{B2}(\xi_1), \varphi_{B3}(\xi_1)]^T$ des Balkenquerschnittes an der Stelle ξ_1 werden durch die folgende Finite-Elemente-Darstellung approximiert:

$$\begin{aligned}
 \tilde{\mathbf{u}}(\xi_1) &= \mathbf{N}_u(\xi_1)\Delta \\
 \tilde{\boldsymbol{\varphi}}(\xi_1) &= \mathbf{N}_\varphi(\xi_1)\Delta
 \end{aligned} \tag{3.11}$$

Hier ist

$$\Delta = [u_1^{(1)}, u_2^{(1)}, u_3^{(1)}, \varphi_T^{(1)}, \varphi_{B2}^{(1)}, \varphi_{B3}^{(1)}, u_1^{(2)}, u_2^{(2)}, u_3^{(2)}, \varphi_T^{(2)}, \varphi_{B2}^{(2)}, \varphi_{B3}^{(2)}]^T$$

der Vektor der Verschiebungen und Verdrehungen in den beiden Endknoten

des Balkens (bei $\xi_1 = 0$ mit dem Index ⁽¹⁾ und bei $\xi_1 = L$ mit ⁽²⁾). Die Formfunktionen werden durch die Matrizen

$$\mathbf{N}_u(\xi_1) = \begin{bmatrix} N_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_3(\phi_2) & 0 & 0 & 0 & N_4(\phi_2) \dots \\ 0 & 0 & N_3(\phi_3) & 0 & -N_4(\phi_3) & 0 \\ \dots & N_2 & 0 & 0 & 0 & 0 \\ 0 & N_5(\phi_2) & 0 & 0 & 0 & N_6(\phi_2) \\ 0 & 0 & N_5(\phi_3) & 0 & -N_6(\phi_3) & 0 \end{bmatrix} \quad (3.12)$$

$$\mathbf{N}_\varphi(\xi_1) = \begin{bmatrix} 0 & 0 & 0 & N_1 & 0 & 0 \\ 0 & 0 & -N_7(\phi_3) & 0 & N_8(\phi_3) & 0 \dots \\ 0 & N_7(\phi_2) & 0 & 0 & 0 & N_8(\phi_2) \\ \dots & 0 & 0 & N_2 & 0 & 0 \\ 0 & 0 & -N_9(\phi_3) & 0 & N_{10}(\phi_3) & 0 \\ 0 & N_9(\phi_2) & 0 & 0 & 0 & N_{10}(\phi_2) \end{bmatrix} \quad (3.13)$$

gegeben. Darin sind ϕ_2 und ϕ_3 die Verhältnisse der Biegesteifigkeit zur Schersteifigkeit, die durch

$$\phi_2 = \frac{12}{L^2} \left(\frac{EI_{B33}^{(E)}}{GA^{(E)}k_{22}} \right) \quad \text{bzw.} \quad \phi_3 = \frac{12}{L^2} \left(\frac{EI_{B22}^{(E)}}{GA^{(E)}k_{33}} \right) \quad (3.14)$$

definiert sind.

Die Polynome N_3 bis N_{10} und die entsprechenden Konstanten ϕ_2 und ϕ_3 wurden durch [FK93] so gewählt, daß die entstehenden Formfunktionen die homogenen stationären Gleichgewichtsbedingungen für die ebene Biegung eines Timoshenko-Balkens erfüllen, wenn die Bezugsachse mit dem Drill- und dem Biegezentrum zusammenfällt. Da dies ein wesentlicher Grund für die gute Qualität des entstehenden Elementes ist, wird nachfolgend untersucht, ob die in Gleichung (3.11) definierten Formfunktionen ebenfalls die Gleichgewichtsbedingungen erfüllen.

Die homogenen stationären Gleichgewichtsbedingungen für den linearisierten Timoshenko-Balken folgen aus den Gleichungen (3.9) und (3.10), indem die dort auftretenden Beschleunigungen zu Null gesetzt werden. Werden \mathbf{u} und φ nun durch $\tilde{\mathbf{u}}$ respektive $\tilde{\varphi}$ substituiert, so erhält man:

$$\begin{aligned} 0 &= \underbrace{E\tilde{u}_{1,\xi_1\xi_1}}_1 \mathbf{e}_1 + E \left(\underbrace{\tilde{\varphi}_{B2,\xi_1\xi_1} \xi_{B3} - \tilde{\varphi}_{B3,\xi_1\xi_1} \xi_{B2}}_2 \right) \mathbf{e}_1 + \quad (3.15) \\ &Gk_{22} \left(\underbrace{\tilde{u}_{2,\xi_1\xi_1}}_{3a} - \underbrace{\tilde{\varphi}_{T,\xi_1\xi_1} \xi_{D3}}_1 - \underbrace{\tilde{\varphi}_{B3,\xi_1}}_{3a} \right) \mathbf{e}_2 + \\ &Gk_{33} \left(\underbrace{\tilde{u}_{3,\xi_1\xi_1}}_{3b} + \underbrace{\tilde{\varphi}_{T,\xi_1\xi_1} \xi_{D2}}_1 + \underbrace{\tilde{\varphi}_{B2,\xi_1}}_{3b} \right) \mathbf{e}_3 \end{aligned}$$

$$\begin{aligned}
 0 = & G \left(\underbrace{I_T \tilde{\varphi}_{T, \xi_1 \xi_1}}_1 + \underbrace{A^{(E)} (\xi_{CD2} k_{33} \tilde{\gamma}_{3, \xi_1} - \xi_{CD3} k_{22} \tilde{\gamma}_{2, \xi_1})}_4 \right) \mathbf{e}_1 \quad (3.16) \\
 & + \left(\underbrace{EI_{B22}^{(E)} \tilde{\varphi}_{B2, \xi_1 \xi_1}}_{5a} + \underbrace{EA^{(E)} \xi_{CB3} \tilde{u}_{1, \xi_1 \xi_1}}_1 - \underbrace{GA^{(E)} k_{33}}_{5a, 6a} \left(\underbrace{\tilde{u}_{3, \xi_1}}_{5a} + \underbrace{\tilde{\varphi}_{T, \xi_1} \xi_{D2}}_{6a} + \underbrace{\tilde{\varphi}_{B2}}_{5a} \right) \right) \mathbf{e}_2 \\
 & + \left(\underbrace{EI_{B33}^{(E)} \tilde{\varphi}_{B3, \xi_1 \xi_1}}_{5b} - \underbrace{EA^{(E)} \xi_{CB2} \tilde{u}_{1, \xi_1 \xi_1}}_1 + \underbrace{GA^{(E)} k_{22}}_{5b, 6b} \left(\underbrace{\tilde{u}_{2, \xi_1}}_{5b} - \underbrace{\tilde{\varphi}_{T, \xi_1} \xi_{D3}}_{6b} - \underbrace{\tilde{\varphi}_{B3}}_{5b} \right) \right) \mathbf{e}_3
 \end{aligned}$$

Die mit 1 unterklammerten Terme sind Null, da \tilde{u}_1 und $\tilde{\varphi}_T$ lineare Funktionen von ξ_1 sind. Gemäß der Definition der Formfunktionen durch Friedman sind 3a bzw. 3b insgesamt jeweils auch Null, was bedeutet, daß die Scherung $\tilde{\gamma}_2$ bzw. $\tilde{\gamma}_3$ innerhalb des Elementes konstant ist. Demnach ist der Term 4 ebenfalls Null. Nach [FK93] entfallen auch die Terme 5a bzw. 5b, wenn sie zusammengefaßt werden. Entfernt man alle diese Terme, so verbleiben von Gleichung (3.15)

$$0 = \underbrace{\tilde{\varphi}_{B2, \xi_1 \xi_1} \xi_{B3} - \tilde{\varphi}_{B3, \xi_1 \xi_1} \xi_{B2}}_2, \quad (3.17)$$

und Gleichung (3.16) wird zu

$$0 = \underbrace{k_{33} \tilde{\varphi}_{T, \xi_1} \xi_{D2}}_{6a} \mathbf{e}_2 + \underbrace{k_{22} \tilde{\varphi}_{T, \xi_1} \xi_{D3}}_{6b} \mathbf{e}_3. \quad (3.18)$$

Dies bedeutet, daß es aufgrund von (3.18) zu einem „Locking“ für den Torsionswinkel φ_T kommt, wenn die Bezugsachse ξ_1 nicht mit dem Drillzentrum D zusammenfällt und damit $\mathbf{r}_D \neq 0$ ist. Fällt die Biegelinie B nicht mit der Bezugsachse ξ_1 zusammen, so ergibt sich aufgrund von (3.17) Scherlocking, da in dem gegebenen Finite-Elemente-Raum gilt $\tilde{\varphi}_{B2, \xi_1 \xi_1} \sim \tilde{\gamma}_3$ und $\tilde{\varphi}_{B3, \xi_1 \xi_1} \sim \tilde{\gamma}_2$.

Fallen die Hauptachsen von $\mathbf{I}_B^{(E)}$ nicht mit den Richtungen ξ_2 und ξ_3 zusammen, so muß Gleichung (3.16) um folgende weitere Terme erweitert werden:

$$\begin{aligned}
 0 = & \dots + \quad (3.19) \\
 & \left(EI_{B23}^{(E)} \tilde{\varphi}_{B3, \xi_1 \xi_1} - GA^{(E)} k_{23} \tilde{\gamma}_3 \right) \mathbf{e}_2 + \\
 & \left(EI_{B23}^{(E)} \tilde{\varphi}_{B2, \xi_1 \xi_1} + GA^{(E)} k_{23} \tilde{\gamma}_2 \right) \mathbf{e}_3
 \end{aligned}$$

Wenn also die Hauptachsen des elastischen Balkenquerschnittes nicht mit den Ebenen der Formfunktionen zusammenfallen, so muß mit Scherlocking gerechnet werden. Für die Praxis bedeutet dies, daß die Elemente immer in entsprechender Lage zu den Hauptachsen korrespondierend diskretisiert werden müssen. Die so erzeugten Elementmatrizen können dann nach Bedarf durch Rotation für die gewünschte Lage dargestellt werden.

3.2.3 Elementmatrizen nach dem Hamilton-Prinzip

Für linearelastische Strukturen existiert ein elastisches Potential U . Die Elementmatrizen des dynamischen Problems können deshalb aus dem Hamiltonprinzip

$$\delta \Pi = \delta \int_{t_1}^{t_2} (T - U) dt = 0$$

für den Timoshenko-Balken gewonnen werden. $\delta \Pi$ stellt eine Variation des „Wirkungsintegrals“ der verformten elastischen Struktur dar, bei der die Zeit nicht variiert wird. Die Arbeit externer Lasten wurde hier fortgelassen. Mit den weiter oben dargestellten Linearisierungen und Schnittkraft-Verformungsbeziehungen ist die potentielle Energie eines Balkenelements mit

$$U = \frac{1}{2} \int_0^L \left(EA^{(E)} (u_{B1, \xi_1})^2 + GA^{(E)} [\gamma_2 \gamma_3] \mathbf{k} \begin{bmatrix} \gamma_2 \\ \gamma_3 \end{bmatrix} + GI_T (\varphi_{T, \xi_1})^2 + E [\varphi_{B2, \xi_1} \ \varphi_{B3, \xi_1}] \mathbf{I}_B^{(E)} \begin{bmatrix} \varphi_{B2, \xi_1} \\ \varphi_{B3, \xi_1} \end{bmatrix} \right) d\xi_1 \quad (3.20)$$

gegeben. Entsprechend ist die kinetische Energie

$$T = \frac{1}{2} \int_0^L \left(\rho \omega \mathbf{I}_C^{(\rho)} \omega + \rho A^{(\rho)} \mathbf{v}_c \mathbf{v}_c \right) d\xi_1 \quad (3.21)$$

mit $\mathbf{v}_c = \mathbf{v} + \omega \wedge \sum_{\alpha=2,3} \xi_{C\alpha} \mathbf{e}_\alpha$ ($\mathbf{v} = \dot{\mathbf{u}}$).

Die Massenmatrix \mathbf{M} und Steifigkeitsmatrix \mathbf{K} der diskretisierten Bewegungsgleichungen

$$\mathbf{M} \ddot{\Delta} + \mathbf{K} \Delta = \mathbf{F} \quad (3.22)$$

erhält man, indem die approximierten Verschiebungen und Verdrehungen aus (3.11) in die Energiegleichungen (3.20) und (3.21) eingesetzt werden.

Die Massenmatrix lautet

$$\mathbf{M} = \int_0^L \mathbf{N}_M^T \begin{bmatrix} \rho A^{(\rho)} & 0 & 0 & 0 & 0 & 0 \\ 0 & \rho A^{(\rho)} & 0 & 0 & 0 & 0 \\ 0 & 0 & \rho A^{(\rho)} & 0 & 0 & 0 \\ 0 & 0 & 0 & \rho I_{C11}^{(\rho)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \rho I_{C22}^{(\rho)} & \rho I_{C23}^{(\rho)} \\ 0 & 0 & 0 & 0 & \rho I_{C23}^{(\rho)} & \rho I_{C33}^{(\rho)} \end{bmatrix} \mathbf{N}_M d\xi_1 \quad (3.23)$$

mit

$$\mathbf{N}_M = \begin{bmatrix} \bar{\mathbf{N}}_u \\ \mathbf{N}_\varphi \end{bmatrix} \quad \text{und} \quad \bar{\mathbf{N}}_u = \begin{bmatrix} \mathbf{N}_u(1, \star) - \xi_{C2} \mathbf{N}_\varphi(3, \star) + \xi_{C3} \mathbf{N}_\varphi(2, \star) \\ \mathbf{N}_u(2, \star) - \xi_{C3} \mathbf{N}_\varphi(1, \star) \\ \mathbf{N}_u(3, \star) + \xi_{C2} \mathbf{N}_\varphi(1, \star) \end{bmatrix}.$$

Entsprechend wird die Steifigkeitsmatrix gebildet. Die nachfolgend angegebene Matrix ist für eine allgemeine Lage der Bezugsachse ξ_1 und der Richtungen ξ_2 und ξ_3 gültig, jedoch sollte beachtet werden, daß zur Vermeidung

von Locking die Bezugsachsen mit den Hauptachsen von $\mathbf{I}_B^{(E)}$ zusammenfallen sollten (s. vorangehenden Abschnitt).

$$\mathbf{K} = \int_0^L \mathbf{N}_K^T \begin{bmatrix} GI_T & 0 & 0 & 0 & 0 & 0 \\ 0 & EI_{B22}^{(E)} & EI_{B23}^{(E)} & 0 & 0 & 0 \\ 0 & EI_{B23}^{(E)} & EI_{B33}^{(E)} & 0 & 0 & 0 \\ 0 & 0 & 0 & EA^{(E)} & 0 & 0 \\ 0 & 0 & 0 & 0 & GA^{(E)}k_{22} & GA^{(E)}k_{23} \\ 0 & 0 & 0 & 0 & GA^{(E)}k_{23} & GA^{(E)}k_{33} \end{bmatrix} \mathbf{N}_K d\xi_1 \quad (3.24)$$

mit

$$\mathbf{N}_K = \begin{bmatrix} \mathbf{N}_{\varphi, \xi_1} \\ \hat{\mathbf{N}}_u \end{bmatrix} \quad \text{und} \quad \hat{\mathbf{N}}_u = \begin{bmatrix} \mathbf{N}_u(1, \star)_{, \xi_1} + \xi_{B3} \mathbf{N}_{\varphi}(2, \star)_{, \xi_1} - \xi_{B2} \mathbf{N}_{\varphi}(3, \star)_{, \xi_1} \\ \mathbf{N}_u(2, \star)_{, \xi_1} - \xi_{D3} \mathbf{N}_{\varphi}(1, \star)_{, \xi_1} - \mathbf{N}_{\varphi}(3, \star) \\ \mathbf{N}_u(3, \star)_{, \xi_1} + \xi_{D2} \mathbf{N}_{\varphi}(1, \star)_{, \xi_1} + \mathbf{N}_{\varphi}(2, \star) \end{bmatrix}.$$

Die Elementmatrizen für dieses erweiterte Balkenelement werden in Anhang B vollständig dargestellt.

Der Lastvektor \mathbf{F} auf der rechten Seite der diskretisierten Gleichung (3.22) enthält die entsprechenden Knotenlasten

$$\mathbf{F} = \left[N^{(1)}, Q_2^{(1)}, Q_3^{(1)}, M_T^{(1)}, M_{B2}^{(1)}, M_{B3}^{(1)}, N^{(2)}, Q_2^{(2)}, Q_3^{(2)}, M_T^{(2)}, M_{B2}^{(2)}, M_{B3}^{(2)} \right].$$

Kapitel 4

Datentransfer

Das Zusammenspiel von Aerodynamik und Strukturodynamik wird anhand von Abb. 4.1 an einem Tragflügel verdeutlicht. Dieser befindet sich zum Zeitpunkt t_n in der mittig groß dargestellten Lage. Der Ortsvektor des Profilschwerpunktes C an der Tragflächenspitze zur Zeit t_n ist

$$\mathbf{r}_S = \mathbf{r}_{O'C}(0) + \mathbf{u}_C,$$

wobei $\mathbf{r}_{O'C}(0)$ die Anfangslage des Querschnittsschwerpunktes im nicht deformierten Flügel und \mathbf{u}_C seine Verschiebung zur Zeit t_n ist. Der Referenzpunkt O' wird problemabhängig geeignet gewählt, beim hier dargestellten Flügel an der Einspannung der Flügelachse als körperfester Punkt, oder für einen frei fliegenden, elastischen Flugkörper im momentanen Schwerpunkt des Flugkörpers (vgl. Kapitel 5 ab Seite 53). Mit der Lage $\mathbf{r}_P = \mathbf{r}_S(t) + \mathbf{r}_{SP}$ der Oberflächenpunkte P des Flügels in der aktuellen Konfiguration und deren Geschwindigkeit \mathbf{v}_S^n als Randbedingungen für die Strömung werden mit dem Strömungslöser die Kräfte $\mathbf{p}\Delta A$ für jedes Oberflächenelement ΔA berechnet. Diese werden dem FEM-Programm für die Struktur übergeben, welches die Form und Geschwindigkeit des Flügels für den neuen Zeitschritt $n + 1$ berechnet.

Die natürliche Schnittstelle zwischen der Berechnung des elastischen Verhaltens und der Strömungssimulation stellt bei diesem Vorgang die aerodynamische Oberfläche, d.h. die vom Strömungsmedium benetzte Oberfläche, des Flugkörpers dar. Da diese auch gleichzeitig mit den Randbedingungen für den Strömungslöser belegt ist, ist bei der Übertragung der elastischen Deformation auf die Oberfläche besondere Sorgfalt notwendig. Umgekehrt muß bei der Weitergabe der aerodynamischen Lasten an den elastischen Flugkörper darauf geachtet werden, daß die Gesamtenergie erhalten bleibt.

Dafür müssen an mehreren Stellen Daten auf der Oberfläche des Flugkörpers in Daten für das verwendete Strukturmodell umgewandelt werden und umgekehrt. Dazu wurde eine Programmbibliothek erstellt, die als Schnittstelle zwischen dem Strömungslöser, dem Strukturlöser und dem Gitterdeformationsprogramm fungiert und die Umwandlung und den Transport der Daten zwischen den einzelnen Programmen vornimmt. Die wichtigsten Bestandteile in diesem Zusammenhang sind die zur Übertragung der Ober-

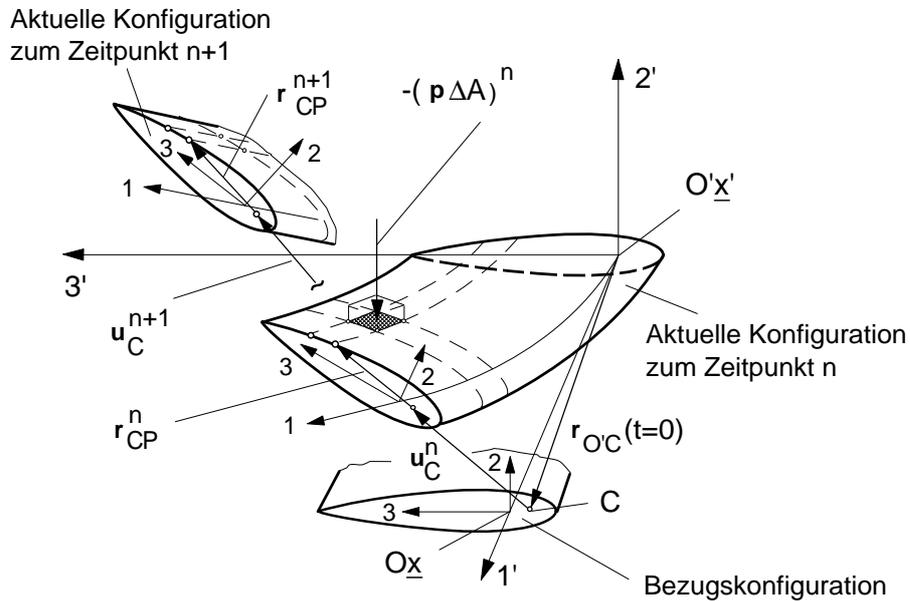


Abbildung 4.1: Schema zur Berechnung der Strukturbewegung

flächenkräfte auf dem Flugkörper auf die elastische Tragstruktur und die Rückübertragung der elastischen Deformation auf die Oberfläche.

4.1 Zuordnung FE-Modell und Oberfläche

Aufgrund der reduzierten Modellierung durch Balkenmodelle in dieser Arbeit, fällt die aerodynamische Oberfläche des Flugkörpers nicht mit der des FE-Modells zusammen. Auch bei der Verwendung eines allgemeineren Finite-Elemente-Modells mit verschiedenen Elementtypen, wie z.B. Schalenelementen, ist es für die Modellierung des elastischen Festkörpers hilfreich, wenn dessen Oberfläche nicht mit der aerodynamischen Oberfläche identisch sein muß. Daher ist es vor der Übertragung der Lasten bzw. Deformationen notwendig, eine Zuordnung zwischen aerodynamischer Oberfläche und Finiten Elementen zu definieren. Hier wurde ein ebenso einfacher, wie pragmatischer Ansatz gewählt: Einem Punkt auf der Oberfläche wird das geometrisch nächstliegende Element zugeordnet. Wie dann mittels dieser Zuordnung der weitere Datentransfer gestaltet wird, muß vom jeweiligen Typ des Finiten Elements abhängig sein, um ein mit der Diskretisierung des Flugkörpers konsistentes Verhalten zu erreichen. Wie weiter unten erläutert, stellt diese geometrische Zuordnung für Balkenelemente eine Zuordnung zu einem materiellen Querschnitt dar.

Werden keine adaptiven Gitter oder Remeshing-Techniken angewandt, so können alle Gitterpunkte auf der Oberfläche als materielle Punkte aufgefaßt werden, so daß die Zuordnung zu einem Element nur einmal im nicht deformierten Zustand erfolgen muß.

Bei komplexen Geometrien oder Mehrkörperproblemen (z.B. der Simulation eines elastischen Flügels mit ausgefahrener, elastischer Klappe) stößt dieser einfache Ansatz an seine Grenzen, da die Zuordnung aufgrund des Abstandes hier zumeist falsche Ergebnisse erzeugt. Als Abhilfe ist es notwendig, die Oberfläche in Bereiche einzuteilen und jeder Teilfläche eine Gruppe Finiter Elemente zuzuordnen, so daß dort die über Abstände definierte Suche wieder korrekt funktioniert (vgl. Abb. 4.2).

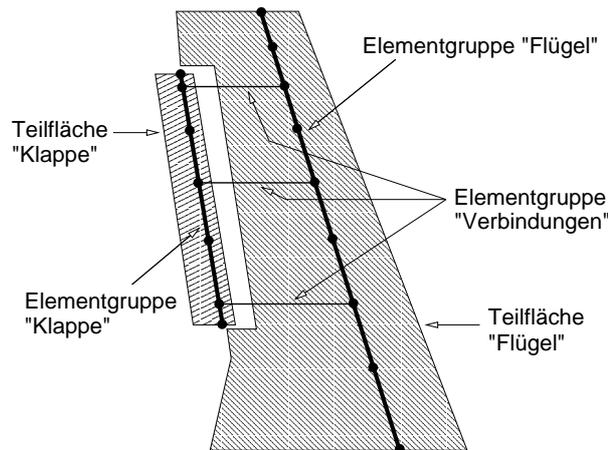


Abbildung 4.2: Beispiel für Konfiguration mit definierter Zuordnung von Teiloberflächen und Elementgruppen: Modellierung eines Tragflügels mit ausgefahrener Klappe, Flügel und Klappe sollen elastisch betrachtet werden. Elementgruppen und Teilflächen mit jeweils gleichen Namen („Klappe“ oder „Flügel“) werden einander zugeordnet. Die Elementgruppe „Verbindungen“ übernimmt somit keine direkten aerodynamischen Lasten, sondern koppelt die beiden anderen Elementgruppen untereinander.

In bestimmten Situationen, z.B. bei der Modellierung eines Tragflügels über den Rahmen, d.h. ohne Modellierung der Bespannung, kann einem Punkt auf der Oberfläche in der Regel kein einzelnes Element eindeutig zugeordnet werden. Für diesen Fall kann eine der Modellierung angepaßte Datenübertragung erreicht werden, indem einer Teiloberfläche mehrere Elementgruppen zugeordnet werden, aus denen jeweils ein Element eindeutig ausgewählt werden kann.

Dabei gibt es folgende Möglichkeiten der Zuordnung zwischen einer aerodynamischen Oberfläche und Elementgruppen:

1. Keine Zuordnung: Die aerodynamischen Lasten der betreffenden Fläche werden nicht berücksichtigt. Dies kann z.B. für die Oberfläche eines Flugzeugrumpfes gelten, wenn nur die Tragflächen elastisch modelliert werden.
2. Eine Elementgruppe zugeordnet: Für den entsprechenden Oberflächenpunkt wird das nächstliegende Element der zugeordneten Elementgruppe

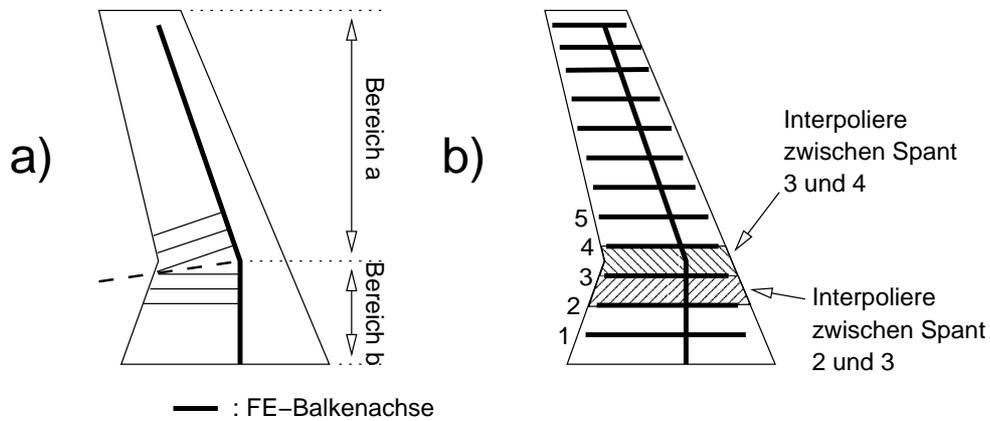


Abbildung 4.3: Beispiele für Konfigurationen, in denen Interpolation der Daten zwischen Elementgruppen notwendig wird. Im Fall a) ergibt sich entlang der eingezeichneten Winkelhalbierenden im Knick der Balkenachse eine Unstetigkeit. Typischerweise ist diese unkritisch, da die Deformationen einer Tragfläche in diesem Bereich noch so klein sind, daß dadurch keine Beeinträchtigung erfolgt. Diese Unstetigkeit kann jedoch durch eine Interpolation der Daten von den Bereichen a und b in der Nähe des Knicks vermieden werden. Bei b) wurde die Tragfläche so modelliert, daß die Querschnitte selbst auch elastisch sind. Dazu werden in einzelnen Querschnitten „Spannten“ aus Balken angebracht, von denen jeweils zwei der dazwischen liegenden Teilfläche zugeordnet sind. Der Flügelachse wird keine Oberfläche zugeordnet. Damit werden alle aerodynamischen Lasten auf die Spanten interpoliert und entsprechend die Deformation der Oberfläche durch Interpolation der Daten der beiden nächstliegenden Spanten berechnet.

pe gesucht.

3. Mehr als eine Elementgruppe zugeordnet: In jeder zugeordneten Elementgruppe wird das dem Oberflächenpunkt nächstgelegene Element gesucht. Für die Übergabe der Last bzw. Deformation wird eine Gewichtung anhand des geometrischen Abstandes zwischen Oberflächenpunkt und Elementgruppe vorgenommen.

Für die in Abb. 4.3 gezeigten Konfigurationen, in denen eine Interpolation von Daten notwendig ist, wurde eine einfache Strategie entwickelt, die für die in dieser Arbeit verwendeten Finite-Elemente-Strukturen aus Balkenelementen mit einer Zuordnung von jeweils maximal zwei Elementgruppen zu einer Oberfläche geeignet ist.

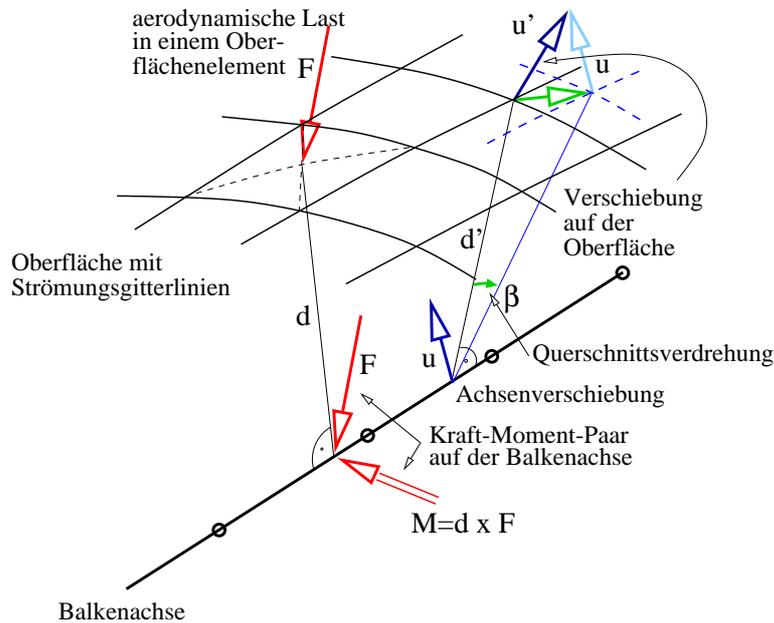


Abbildung 4.4: Schema zur Übertragung der aerodynamischen Lasten auf Balkenelemente bzw. der Deformation des Balkens auf die Oberfläche.

4.2 Lastübertragung auf das erweiterte Balkenelement

Ist eine Gruppe von Balkenelementen einer Oberfläche zugeordnet, so werden alle an dieser Oberfläche auftretenden aerodynamischen Lasten, ggf. gewichtet, in Knotenlasten für die beteiligten Knoten umgerechnet. Dazu werden zunächst die aerodynamischen Lasten und ihre jeweiligen Angriffspunkte in der nicht deformierten Konfiguration bestimmt. Bei dem Strömungslöser FLOWer sind dies die Mittelpunkte der Zellen des Strömungsnetzes auf der Flugkörperoberfläche, in denen aus dem herrschenden Druck und dem Flächeninhalt ein resultierender Lastvektor bestimmt wird. Im Falle von Berechnungen mit Verwendung der Navier–Stokes–Gleichungen wird die Reibung ebenfalls in diesem Lastvektor berücksichtigt, der deshalb i.allg. nicht mehr orthogonal auf dem betrachteten Element der aerodynamischen Oberfläche steht.

Für jeden dieser Lastangriffspunkte wird das nächstgelegene Balkenelement aus der Elementgruppe gesucht. Innerhalb dieses Elementes wird dem Oberflächenpunkt ein Punkt auf der Achse des Elementes zugeordnet, entweder als Fußpunkt des Lotes von der Oberfläche auf die Achse oder, wenn das Lot die Achse außerhalb des Elementes trifft, der entsprechende Endpunkt der diskreten Elementachse. Über die so bestimmten Vektoren ist jeder Lastangriffspunkt auf der Oberfläche fest mit einem Punkt auf der Achse eines Balkenelementes verbunden (vgl. Abb. 4.4). Da diese Zuordnung in der B.K. zu erfolgen hat, kann der gefundene Lotvektor mit einem Balkenquerschnitt gleichgesetzt werden. Weil im verwendeten Balkenmodell die Querschnitte bei der

Deformation des Balkens nicht ihre Form sondern nur ihre Lage ändern, kann der Querschnitt in der weiteren Betrachtung als starre Scheibe angesehen werden.

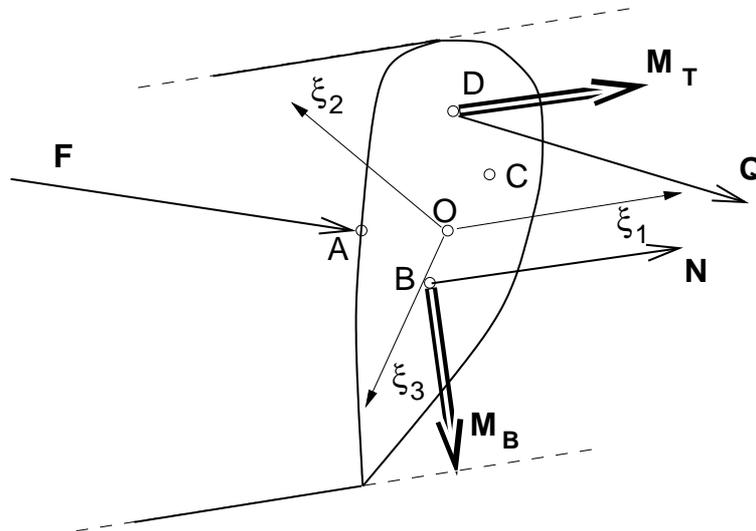


Abbildung 4.5: Materieller Querschnitt eines Balkenelementes mit angreifender Last F im Punkt A .

Eine an einer starren Scheibe in einem Randpunkt angreifende Kraft kann in einen hinsichtlich der virtuellen Arbeit und der tatsächlich übertragenen Energie äquivalenten Satz von Kräften und Momenten für einen anderen Angriffspunkt umgerechnet werden. Im Falle des erweiterten Balkenelementes ist dabei zusätzlich die Lage der einzelnen Wirkungslinien für Quer- und Längskräfte bzw. -Momente zu beachten (vgl. Abb. 4.5). Über die in A angreifende äußere Kraft F seien die übrigen Kraft- und Momentenvektoren wie folgt definiert (vgl. auch Abb. 3.10 und die Definitionen in Abschnitt 3.2):

$$N = F e_{s1} \tag{4.1}$$

$$= F_1$$

$$Q_2 = F e_{s2} \tag{4.2}$$

$$= F_2$$

$$Q_3 = F e_{s3}$$

$$= F_3$$

$$M_T = (r_{DA} \wedge F) e_{s1} \tag{4.2}$$

$$= (\xi_{A2} - \xi_{D2}) F_3 - (\xi_{A3} - \xi_{D3}) F_2$$

$$M_{B2} = (r_{BA} \wedge F) e_{s2} \tag{4.2}$$

$$= (\xi_{A3} - \xi_{B3}) F_1$$

$$M_{B3} = (r_{BA} \wedge F) e_{s3}$$

$$= -(\xi_{A2} - \xi_{B2}) F_1$$

Mit Hilfe des Prinzips der virtuellen Verrückungen für die starre Balkenquerschnitts-Scheibe kann gezeigt werden, daß die oben beschriebene Art der Übertragung der außen angreifenden Last auf das Drill- und Biegezentrum des Balkens energieerhaltend ist. Sei δr_A eine virtuelle Verrückung des Lastangriffspunktes A , so kann diese durch eine virtuelle Verrückung des

Ursprungs O $\delta \mathbf{r}_O$ und eine virtuelle Drehung $\delta \varphi$ dargestellt werden:

$$\delta \mathbf{r}_A = \delta \mathbf{r}_O + \delta \varphi \wedge \mathbf{r}_A.$$

Als virtuelle Verrückung ist jede kinematisch mögliche, kleine Verschiebung des Punktes A zugelassen. Dazu gehört selbstverständlich auch die in einem Zeitintervall Dt tatsächlich stattfindende Verschiebung des Punktes A , die sich mit $D\mathbf{r}_O = \mathbf{v}_O Dt$ und $D\varphi = \boldsymbol{\omega} \wedge \mathbf{r}_A Dt$ gemäß $D\mathbf{r}_A = (\mathbf{v}_O + \boldsymbol{\omega} \wedge \mathbf{r}_A)Dt = \mathbf{v}_A Dt$ darstellt. Entsprechendes gilt an den Punkten D und B . Damit gilt dann für die von einer im Punkt A angreifenden Kraft \mathbf{F} tatsächlich verrichtete Arbeit $DW_F = \mathbf{F}D\mathbf{r}_A$ und für die bei einer virtuellen Verrückung $\delta \mathbf{r}_A$ ausgeführte virtuelle Arbeit

$$\delta W_F = \mathbf{F}\delta \mathbf{r}_A.$$

Bildet man die virtuelle Arbeit zu den in den Gleichungen (4.1) und (4.2) definierten Kräften und Momenten, so erhält man:

$$\begin{aligned} \delta W &= \mathbf{N}\delta \mathbf{r}_B + \mathbf{Q}\delta \mathbf{r}_D + \delta \varphi \mathbf{M}_T + \delta \varphi \mathbf{M}_B & (4.3) \\ &= \mathbf{N}\delta \mathbf{r}_O + \delta \varphi (\mathbf{r}_B \wedge \mathbf{N}) + \mathbf{Q}\delta \mathbf{r}_O + \delta \varphi (\mathbf{r}_D \wedge \mathbf{Q}) + \\ &\quad \delta \varphi (\mathbf{r}_{DA} \wedge \mathbf{Q}) + \delta \varphi (\mathbf{r}_{BA} \wedge \mathbf{N}) \\ &= (\mathbf{N} + \mathbf{Q})\delta \mathbf{r}_O + \delta \varphi (\mathbf{r}_A \wedge (\mathbf{N} + \mathbf{Q})) \\ &= \mathbf{F}\delta \mathbf{r}_O + \mathbf{F}(\delta \varphi \wedge \mathbf{r}_A) \\ &= \mathbf{F}\delta \mathbf{r}_A \end{aligned}$$

Die wirklich geleistete Arbeit DW in einem Zeitintervall Dt ist das Produkt der übertragenen Leistung mit Dt , also $DW = \mathbf{F}\mathbf{v}_A Dt$. Die in [Mas02] geforderte richtige Energieübertragung, die für die korrekte Kopplung der beiden Verfahren wichtig ist, ist für die Übertragung von der Oberfläche auf die entsprechenden Balkenachsen somit gegeben.

Die innerhalb eines Elementes aufgebrachtten Kräfte und Momente werden auf geeignete Weise in einen Satz von Knotenkräften und -Momenten umgerechnet und an den beteiligten Knoten hinzugefügt. Dazu sind zwei Varianten untersucht worden:

1. Mit Berücksichtigung der Deformation eines Balkens bei mittig angreifenden Lasten unter Beachtung der Erhaltung der dadurch erzeugten virtuellen Arbeit. Die virtuellen Verrückungen in einem Punkt ξ_1 auf der Balkenachse können mittels der Ansatzfunktionen aus Gleichung (3.11) wie folgt dargestellt werden:

$$\begin{aligned} \delta \hat{\mathbf{u}}(\xi_1) &= \mathbf{N}_u(\xi_1)\delta \Delta \\ \delta \hat{\varphi}(\xi_1) &= \mathbf{N}_\varphi(\xi_1)\delta \Delta \end{aligned}$$

Sei ξ_F die ξ_1 -Koordinate des Lastangriffspunktes auf der Balkenachse. Die Indizes ⁽¹⁾ und ⁽²⁾ stehen für den Knoten 1 ($\xi_1 = 0$) bzw. Knoten 2

$(\xi_1 = L)$. Über die virtuelle Arbeit

$$\begin{aligned} \delta W &= \delta \tilde{\mathbf{u}}(\xi_F) \mathbf{F} + \delta \tilde{\boldsymbol{\varphi}}(\xi_F) \mathbf{M} \\ &= (\delta u_1^{(1)} N_1(\xi_F) + \delta u_1^{(2)} N_2(\xi_F)) F_1 + \\ &\quad (\delta u_2^{(1)} N_3(\xi_F, \phi_2) + \delta \varphi_{B3}^{(1)} N_4(\xi_F, \phi_2) + \\ &\quad \delta u_2^{(2)} N_5(\xi_F, \phi_2) + \delta \varphi_{B3}^{(2)} N_6(\xi_F, \phi_2)) F_2 + \dots \end{aligned}$$

erhält man durch Einsetzen der Formfunktionen und Sortieren nach den virtuellen Verschiebungen und Verdrehungen der Endknoten die äquivalenten Knotenlasten wie folgt:

$$\begin{aligned} N^{(1)} &= N_1(\xi_F) F_1 \\ N^{(2)} &= N_2(\xi_F) F_1 \\ M_T^{(1)} &= N_1(\xi_F) M_T \\ M_T^{(2)} &= N_2(\xi_F) M_T \\ Q_2^{(1)} &= N_3(\xi_F, \phi_2) F_2 + N_7(\xi_F, \phi_2) M_{B3} \\ Q_2^{(2)} &= N_5(\xi_F, \phi_2) F_2 + N_9(\xi_F, \phi_2) M_{B3} \\ Q_3^{(1)} &= N_3(\xi_F, \phi_3) F_3 - N_7(\xi_F, \phi_3) M_{B2} \\ Q_3^{(2)} &= N_5(\xi_F, \phi_3) F_3 - N_9(\xi_F, \phi_3) M_{B2} \\ M_{B2}^{(1)} &= -N_4(\xi_F, \phi_3) F_3 + N_8(\xi_F, \phi_3) M_{B2} \\ M_{B2}^{(2)} &= -N_6(\xi_F, \phi_3) F_3 + N_{10}(\xi_F, \phi_3) M_{B2} \\ M_{B3}^{(1)} &= N_4(\xi_F, \phi_2) F_2 + N_8(\xi_F, \phi_2) M_{B3} \\ M_{B3}^{(2)} &= N_6(\xi_F, \phi_2) F_2 + N_{10}(\xi_F, \phi_2) M_{B3} \end{aligned}$$

Die so bestimmten Knotenlasten entsprechen, wenn die Scherung vernachlässigt wird ($\phi_2 = \phi_3 = 0$), denjenigen, die nach [SGH95, Czi89] für die klassische Balkenbiegung analytisch ermittelbar sind.

2. Bei der zweiten Variante darf sich die Balkenachse des freigeschnittenen Elementes unter der Belastung nicht verbiegen. Daraus folgt, daß an den Knoten keine Momente aufgebracht werden dürfen. Alle angreifenden Kräfte und Momente werden hier wie folgt ausschließlich in Knoten-

kräfte umgerechnet.

$$\begin{aligned}
 N^{(1)} &= N\left(1 - \frac{\xi_F}{L}\right) & N^{(2)} &= N\frac{\xi_F}{L} \\
 M_T^{(1)} &= M_T\left(1 - \frac{\xi_F}{L}\right) & M_T^{(2)} &= M_T\frac{\xi_F}{L} \\
 Q_2^{(1)} &= Q_2\left(1 - \frac{\xi_F}{L}\right) - M_{B3}/L & Q_2^{(2)} &= Q_2 - Q_2^{(1)} \\
 Q_3^{(1)} &= Q_3\left(1 - \frac{\xi_F}{L}\right) + M_{B2}/L & Q_3^{(2)} &= Q_3 - Q_3^{(1)} \\
 M_{B2}^{(1)} &= 0 & M_{B2}^{(2)} &= 0 \\
 M_{B3}^{(1)} &= 0 & M_{B3}^{(2)} &= 0
 \end{aligned}$$

Dies entspricht einem Balkenmodell, bei dem für die Verschiebung ein linearer Ansatz und die Verdrehung konstante Werte verwendet werden.

Der erste Fall hat naheliegende Vorteile, wenn es um die Berechnung einer Biegelinie eines Balkens mit wenigen, diskret angreifenden Lasten unter Verwendung weniger Elemente geht. Außerdem wird dort die Erhaltung der Energie bei der Umrechnung in Knotenlasten für den diskretisierten Flugkörper beachtet, die laut [Mas02] für die lose Kopplung wünschenswert ist. Der wesentliche Nachteil ist, daß eine mittels Einzellasten diskretisierte konstante Streckenlast auf einem endlich langen Balken stets auch zu einer Deformation an den Enden des Balkens führt, die physikalisch nicht richtig ist.

Im zweiten Fall kann eine konstante Streckenlast, die nur zu einer Verschiebung oder Rotation des gesamten Balkens führt, auch mittels diskreter Einzellasten gut wiedergegeben werden. Allerdings wird eine feinere Diskretisierung eines Balkens mit örtlich unterschiedlicher Belastung benötigt, um eine gleich gute Approximation der Deformation zu erhalten wie im ersten Fall. Aufgrund der genannten Eigenschaften wird, vor allem für Simulationen des frei fliegenden, elastischen Flugkörpers, die 2. Variante verwendet. Nachteile, die durch die Verletzung der Energieerhaltung entstehen könnten, sind in der praktischen Anwendung nicht aufgefallen.

4.3 Verschiebung der Oberfläche über einem Balkenelement

Zu jeder Ecke eines Flächenelementes des Strömungsnetzes wird auf der Oberfläche des Flugkörpers das in der nicht deformierten Konfiguration nächstliegende und damit zugehörige Balkenelement bestimmt. Innerhalb dieses Elementes wird dem Oberflächenpunkt ein Punkt auf der Achse des Elements zugeordnet, entweder der Fußpunkt des Lotes von der Oberfläche auf die Achse oder der entsprechende Endpunkt der diskreten Elementachse. Über die so bestimmten Vektoren ist jeder Punkt auf der Oberfläche fest mit einem Punkt auf der Achse eines Balken-Elementes verbunden. Diese Vektoren werden nun,

der Balkenhypothese „ebene Querschnitte bleiben eben“ folgend, bei der Deformation der Achse starr mitgeführt (vgl. Abb. 4.4). Ein ähnlicher Zugang wurde in [Sch97] verwendet.

Die in den Knoten des diskreten Modells vorliegende Deformation wird durch Auswertung der Formfunktionen für das jeweilige Element im Lotfußpunkt berechnet. Aus der so bestimmten Verschiebung und Verdrehung des entsprechenden Balkenquerschnittes wird die Verschiebung des Gitterknotens berechnet.

4.4 Interpolation bei Zuordnung zweier Balkenelementgruppen zu einer Oberfläche

Wie bereits in Abb. 4.3 illustriert können bereits bei relativ einfachen Geometrien Situationen auftreten, in denen Deformationen aus dem Biegezustand mindestens zweier Elemente interpoliert werden müssen. Entsprechend sollten an derselben Stelle angreifende äußere Kräfte auf die Elemente verteilt werden. Nachfolgend wird eine Interpolationsstrategie für aus Balkenelementen bestehende, lineare Strukturen vorgestellt, bei denen zwischen zwei Elementgruppen interpoliert werden kann.

Einem Punkt A auf der Oberfläche (Kraftangriffspunkt, Deformationsabfragepunkt) werden auf den zwei der entsprechenden Oberfläche zugeordneten Balken-Elementgruppen die nächstgelegenen Punkte P_1 und P_2 (jeweils in Elementgruppe 1 und 2) zugewiesen. In diesen Punkten werden die Kraftanteile in die Balken eingeleitet, bzw. die Deformationen abgefragt. Um festzustellen, mit welchen Gewichten die Interpolation zu erfolgen hat, oder ob überhaupt eine Interpolation notwendig ist, wird das durch die Punkte A , P_1 , P_2 aufgespannte Dreieck betrachtet, das im Extremfall auch zu einer Linie oder einem Punkt entarten kann. Gilt in diesem Dreieck

$$r_{AP_1}^2 + r_{P_1P_2}^2 > r_{AP_2}^2 \quad \text{und} \quad r_{AP_2}^2 + r_{P_1P_2}^2 > r_{AP_1}^2 \quad (4.4)$$

dann muß interpoliert werden, da der Punkt im Bereich „zwischen“ den beiden Elementgruppen liegt. Die Ungleichungen (4.4) besagen, daß in den Punkten P_1 und P_2 spitze Winkel vorliegen müssen.

Dies soll anhand der zwei Beispielsituationen aus Abb. 4.3 geometrisch erläutert werden. Die Situation a) ist in Abb. 4.6 mit drei möglichen Situationen nochmals aufgetragen. Werden Daten für den Oberflächenpunkt $A(1)$ übertragen, so fällt der auf Elementgruppe 2 nächstgelegene Punkt mit dem Endpunkt der Gruppe zusammen. Aufgrund der Linearität der Achsen der Elementgruppen ist das Dreieck $A(1)P_1(1)P_2(1)$ im Punkt $P_1(1)$ rechtwinklig, und der Datenaustausch zwischen dem Punkt $A(1)$ und den Finiten Elementen soll nur mit dem Punkt $P_1(1)$ geschehen. Im Fall des Oberflächenpunktes $A(2)$ hat das entstehende Dreieck in den Punkten $P_1(2)$ und $P_2(2)$ jeweils spitze Winkel, d.h. Ungleichungen (4.4) sind erfüllt. Hier wird der Datenaustausch mittels einer Interpolation bzw. mittels Wichtungsfaktoren zwischen dem Punkt $A(2)$ und beiden Elementgruppen stattfinden. Für den Punkt $A(3)$

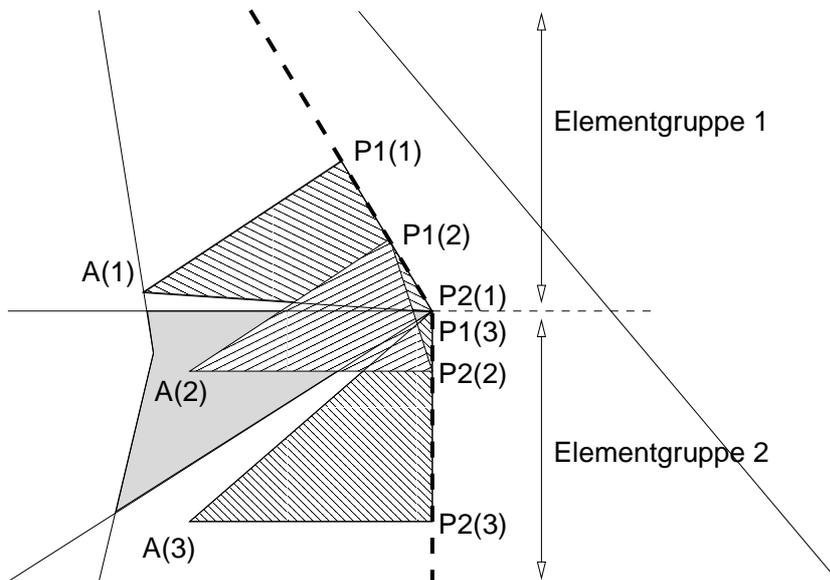


Abbildung 4.6: Interpolation in der „Knicksituation“: Innerhalb der hellgrau dargestellten Fläche wird interpoliert, z.B. die Daten an Punkt A(2), außerhalb nicht, z.B. an den Punkten A(1) und A(3).

gilt eine ähnliche Situation wie bei A(1), lediglich ist nun der rechte Winkel in $P_2(3)$ und demnach die Zuordnung zwischen A(3) und der Elementgruppe 2 ausschließlich hergestellt. Die in der Abbildung grau skizzierte Fläche umfaßt den Bereich, in dem die Interpolationsbedingung gültig ist.

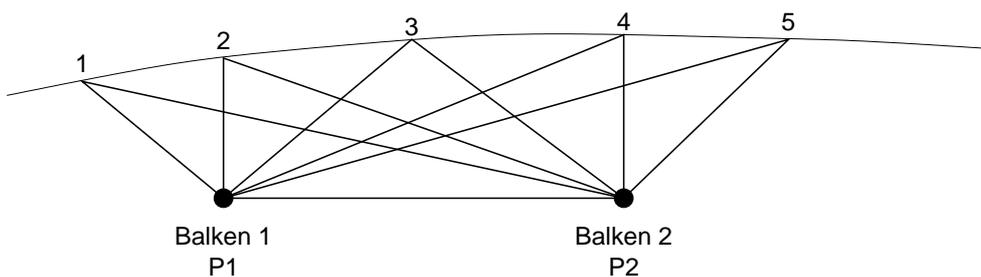


Abbildung 4.7: Interpolation bei nebeneinander verlaufenden Balkengruppen: Fällt der Oberflächenpunkt A in den Bereich zwischen den Punkten 2 und 4, so wird interpoliert, ansonsten nicht.

Ähnliches ergibt sich in der Situation b) aus Abb. 4.3: hier sind die Bedingungen (4.4) für alle Oberflächenpunkte A zwischen den Punkten 2 und 4 in der Skizze 4.7 erfüllt. Für den Punkt 1 ist die Bedingung im Punkt P_1 verletzt, da dort ein stumpfer Winkel ist, für den Punkt 5 entsprechend in Punkt P_2 .

Insgesamt ergibt sich folgender Algorithmus: Bestimme

$$\begin{aligned}d_1 &:= r_{AP_1}^2 + r_{P_1P_2}^2 - r_{AP_2}^2 \\d_2 &:= r_{AP_2}^2 + r_{P_1P_2}^2 - r_{AP_1}^2.\end{aligned}\tag{4.5}$$

Falls

$$\begin{aligned}d_1 < 0 & & d_1 &:= 0 \\d_2 < 0 & \text{ setze } & d_2 &:= 0 \\d_1 = d_2 = 0 & & d_1 &:= d_2 := 1.\end{aligned}\tag{4.6}$$

Damit ergeben sich als Wichtungs- bzw. Interpolationsfaktoren

$$I_1 := \frac{d_2}{d_1 + d_2} \quad \text{und} \quad I_2 := \frac{d_1}{d_1 + d_2}.\tag{4.7}$$

Bei der Übertragung von Daten für Kräfte von der Oberfläche zu den Balkenelementen bedeutet dies, daß die Kräfte mit dem Wert I_1 gewichtet auf den Punkt P_1 in Elementgruppe 1 und mit dem Faktor I_2 gewichtet auf P_2 in Elementgruppe 2 übertragen werden.

Umgekehrt werden Daten von den Balkenelementen, hier die Verschiebungen, Verdrehwinkel und deren Zeitableitungen entsprechend mit den Faktoren I_1 und I_2 interpoliert in den Oberflächenpunkt A übertragen.

Die Fallunterscheidung (4.6) sorgt dafür, daß a) bei Verletzung einer der Bedingungen (4.4) der passende Wichtungsfaktor zu Null gesetzt wird und damit die Verteilung bzw. Interpolation effektiv ausgeschaltet wird, und daß b) bei Zusammenfallen des Dreiecks zu einem einzigen Punkt beide Gewichte gleich groß ausfallen.

Kapitel 5

Freiflugmodellierung

Bevor die Simulation von Lenkbewegungen mit dem gekoppelten Verfahren möglich ist, müssen die evtl. auftretenden freien Translations- und Rotationsbewegungen des gesamten Körpers im Verfahren berücksichtigt werden. Weil der Körper diese Freiheitsgrade auch bei Starrheit wahrnehmen könnte, nennen wir sie auch Starrkörperfreiheitsgrade. Zu den Starrkörperfreiheitsgraden gehören in der Regel sehr große Lageänderungen. Diese großen Lageänderungen müssen entweder in der Formulierung der Finite-Elemente-Beschreibung für den elastischen Festkörper berücksichtigt werden, oder die Gesamtbewegung wird näherungsweise als Summe aus einer Bewegung in Starrkörperkoordinaten und kleinen Deformationsbewegungen angesetzt.

Im Rahmen dieser Arbeit wurde ein entsprechender Separationsansatz zur getrennten Berechnung der Bewegung des Flugkörpers in den Starrkörperfreiheitsgraden und elastischen Deformationen verfolgt, da dieser auch die Möglichkeit bietet, einen starren Körper auf einfache Weise mit der Strömung zu koppeln, indem die Berechnung der elastischen Deformation unterdrückt wird. Hat der Flugkörper mindestens einen starren Freiheitsgrad, d.h. ein Eigenwert ist unter Vakuumbedingungen Null, so wird eine Aufteilung der Bewegung in einen quasi starren Anteil und eine Deformation des Flugkörpers vorgenommen. Dazu wird ein neues, am deformierten Körper schwerpunktfestes, kartesisches Koordinatensystem eingeführt, dessen räumliche Orientierung über den auf den momentanen Schwerpunkt bezogenen Drallsatz für das Gesamtsystem, mit dem momentanen Trägheitstensor des deformierten Körpers, bestimmt wird. Mit Schwerpunkt ist dabei das momentane Massenzentrum gemeint, das wegen der zeitlich veränderlichen Körperform kein materieller Punkt des Körpers ist, also in jedem Zeitschritt neu bestimmt werden muß. In diesem neuen Koordinatensystem wird die Deformationsänderung des elastischen Körpers und auch dessen Umströmung berechnet. In der englischsprachigen Literatur werden diese Koordinaten oftmals mit „Mean Axes“ bezeichnet [Mil68, WS98, MM01]. Wenn der Körper einen Fixpunkt hat, z.B. ein Kugelgelenk, tritt dieser an die Stelle des Schwerpunktes.

5.1 „Mean Axes“

Abhängig von der Konfiguration des Modells kann der Ursprung des neuen Koordinatensystems zwei Positionen annehmen:

1. der Körper hat je drei starre Translations- und Rotationsfreiheitsgrade: Der Koordinatenursprung liegt immer im momentanen Schwerpunkt des Flugkörpers.
2. der Körper hat nur starre Rotationsfreiheitsgrade: In dem Fall sei der Körper z.B. in einem Gelenk gelagert. Der Koordinatenursprung des bewegten Koordinatensystems liegt dann in diesem Gelenk. Der Punkt ist damit körperfest.

Der Ablauf des Verfahrens innerhalb eines Zeitschrittes ist im folgenden grob skizziert (vgl. auch Abb. 6.2). Dabei wird die Bestimmung der momentanen Lage der „Mean Axes“ und der davon entkoppelten Deformation für den frei fliegenden Flugkörper beschrieben, d.h. für den ersten Fall, in dem der Ursprung im momentanen Schwerpunkt liegt. Für den anderen Fall erfolgt die Herleitung analog, jedoch müssen nur die Rotationsanteile betrachtet werden.

Die dynamischen Bilanzgleichungen für Impuls und Drall, die zunächst in einem Inertialsystem formuliert seien, können formal auf ein beliebig bewegtes sog. Laborsystem umgerechnet werden. Dieses kann wie beschrieben so spezialisiert werden, daß es das System der „Mean Axes“ repräsentiert. Nachfolgend wird das neu eingeführte Koordinatensystem der „mean axes“ deshalb auch mit Laborsystem bezeichnet.

5.1.1 Massenträgheits-gemittelte Bewegung

Zunächst wird im Inertialsystem $O'(x', y', z')$, auf das die Bewegung des Flugkörpers bezogen wird, die Lage des Schwerpunktes in der nicht deformierten Konfiguration (Bezugskonfiguration, B.K.) in der Ausgangsposition bestimmt. Im weiteren werden alle Größen, die bzgl. des Inertialsystems zu bilden sind, mit einem ' gekennzeichnet. Das Laborsystem wird mit $O(x, y, z)$ bezeichnet. Alle Größen mit einem Index o beziehen sich auf den momentanen Schwerpunkt bzw. den Ursprung O des Laborsystems.

Die aerodynamischen Kräfte werden über die Schnittstelle von Strömungslöser und Festkörpermodell auf die elastische Struktur übertragen und im FE-Modell als Knotenkräfte F_j^a und Knotenmomente M_j^a aufgebracht¹. Aus diesen Knotenlasten des FE-Modells werden die resultierende Kraft und das resultierende Moment, bezogen auf den Ursprung des Laborsystems, bestimmt.

$$F_R = \sum_{\text{Knoten } j} F_j^a \quad (5.1)$$

$$M_{OR} = \sum_{\text{Knoten } j} M_j^a + r_j \wedge F_j^a$$

¹Die Momente sind auf den jeweiligen Knoten des FE-Modells bezogen.

Der Ortsvektor $\mathbf{r}_j = \bar{\mathbf{r}}_j + \mathbf{u}_j$ gibt die momentane Lage des Knotens j in $O(x, y, z)$ an, $\bar{\mathbf{r}}_j$ dessen Lage im undeformierten Zustand, \mathbf{u}_j die Relativverschiebung des Knotens im Laborsystem. Entsprechend ist $\mathbf{r}_{S,j} = \bar{\mathbf{r}}_{S,j} + \mathbf{u}_{S,j}$ der Ortsvektor, $\mathbf{u}_{S,j}$ die Verschiebung und $\mathbf{v}_{S,j}$ die Verschiebungsgeschwindigkeit des momentanen Schwerpunktes von Element j . $\theta_{S,j}$ ist der momentane Trägheitstensor des Elementes j bezogen auf dessen Schwerpunkt.

Nach der Bestimmung der Gesamtmasse

$$m_{\text{ges}} = \sum_{\text{Elemente } j} m_j$$

und des momentanen Trägheitstensors des diskretisierten Flugkörpers

$$\begin{aligned} \theta_O &= \sum_{\text{Elemente } j} \theta_{S,j} + m_j [(\mathbf{r}_{S,j} \mathbf{r}_{S,j}) \mathbf{E} - (\mathbf{r}_{S,j} \circ \mathbf{r}_{S,j})] \\ &= \sum_{\text{Elemente } j} \theta_{S,j} + m_j [(\bar{\mathbf{r}}_{S,j} \bar{\mathbf{r}}_{S,j}) \mathbf{E} - (\bar{\mathbf{r}}_{S,j} \circ \bar{\mathbf{r}}_{S,j})] + \\ &\quad m_j [(2\bar{\mathbf{r}}_{S,j} \mathbf{u}_{S,j} + \mathbf{u}_{S,j} \mathbf{u}_{S,j}) \mathbf{E} - (\bar{\mathbf{r}}_{S,j} \circ \mathbf{u}_{S,j} + \mathbf{u}_{S,j} \circ \bar{\mathbf{r}}_{S,j} + \mathbf{u}_{S,j} \circ \mathbf{u}_{S,j})], \end{aligned} \quad (5.2)$$

sowie dessen zeitlicher Änderung im Laborsystem

$$\begin{aligned} \frac{D}{Dt}(\theta_O) &= \sum_{\text{Elemente } j} \frac{D}{Dt}(\theta_{S,j}) + \\ &\quad m_j [(2\bar{\mathbf{r}}_{S,j} \mathbf{v}_{S,j} + 2\mathbf{u}_{S,j} \mathbf{v}_{S,j}) \mathbf{E} - \\ &\quad (\bar{\mathbf{r}}_{S,j} \circ \mathbf{v}_{S,j} + \mathbf{v}_{S,j} \circ \bar{\mathbf{r}}_{S,j} + \mathbf{v}_{S,j} \circ \mathbf{u}_{S,j} + \mathbf{u}_{S,j} \circ \mathbf{v}_{S,j})], \end{aligned} \quad (5.3)$$

können die Bewegungsgleichungen für die massengemittelte Bewegung im Inertialsystem gelöst werden. Dazu wurden die Impulsbilanz im Inertialsystem

$$\mathbf{a}'_O m_{\text{ges}} = \mathbf{F}_R \quad (5.4)$$

und die Drallbilanz im Inertialsystem, bezogen auf den momentanen Schwerpunkt,

$$\frac{D'}{Dt}(\theta_O \boldsymbol{\omega}_O) = \frac{D}{Dt}(\theta_O) \boldsymbol{\omega}_O + \theta_O \dot{\boldsymbol{\omega}}_O + \boldsymbol{\omega}_O \wedge (\theta_O \boldsymbol{\omega}_O) = \mathbf{M}_{OR} \quad (5.5)$$

herangezogen. Über die so bestimmten Größen Beschleunigung \mathbf{a}'_O und Winkelgeschwindigkeit $\boldsymbol{\omega}_O$ wird die Lageänderung des Laborsystems $O(x, y, z)$ gegenüber dem Inertialsystem $O'(x', y', z')$ definiert.

Die so bestimmten aktuellen Translations- und Rotationsgeschwindigkeiten sowie die neue Lage des bewegten Koordinatensystems werden an den Strömungslöser übergeben, damit sie dort in der Strömungssimulation entsprechend berücksichtigt werden können.

5.1.2 Separation der elastischen Deformation

Zur Berechnung des Deformationsanteils der Bewegung mit Hilfe des FE-Modelles müssen die aus den aerodynamischen Kräften gebildeten Knotenlasten so verändert werden, daß die gesamte Impuls- bzw. Drehimpulsänderung des FE-Modells zu Null wird, d.h. die zur Lösung der massengemittelten Bewegungsgleichungen (5.4) und (5.5) „verbrauchten“ Kraftanteile müssen abgezogen werden. Aus der mittleren Translations- und Rotationsbeschleunigung des Körpers resultieren Trägheitskräfte bzw. -momente, die jedem einzelnen Element des FE-Modells zugeordnet werden können. Diese Kräfte und Momente müssen, nach Umwandlung in äquivalente Knotenlasten, von den gegebenen aerodynamischen Knotenlasten abgezogen werden. Die verbleibenden Kräfte und Momente tragen ausschließlich zur Deformationsbewegung des Modells bei. Die so in dem Laborsystem bestimmte Deformationsbewegung kann nun genutzt werden, um die entsprechende Nachführung des Rechnernetzes des Strömungslösers zu bestimmen.

Betrachtet man das einzelne Element j als Starrkörper, so erhält es durch die relative Bewegung der beiden Koordinatensysteme und die Deformation die Schwerpunktsbeschleunigung

$$\mathbf{a}'_{S,j} = \mathbf{a}'_O + \dot{\boldsymbol{\omega}}_O \wedge \mathbf{r}_{S,j} + \boldsymbol{\omega}_O \wedge (\boldsymbol{\omega}_O \wedge \mathbf{r}_{S,j}) + 2\boldsymbol{\omega}_O \wedge \mathbf{v}_{S,j} + \mathbf{a}_{S,j} \quad (5.6)$$

im Inertialsystem, die sich anteilig aus der Laborbeschleunigung, der Coriolis-Beschleunigung $2\boldsymbol{\omega}_O \wedge \mathbf{v}_{S,j}$ und der Relativbeschleunigung im Laborsystem zusammensetzt. Für die nachfolgende Bestimmung der „verbrauchten“ Lastanteile betrachten wir

$$\bar{\mathbf{a}}'_{S,j} = \mathbf{a}'_{S,j} - \mathbf{a}_{S,j} \quad (5.7)$$

Daraus ergibt sich für die Trägheitskraft für das Element j

$$\mathbf{F}_j^{\text{korr}} = \bar{\mathbf{a}}'_{S,j} m_j. \quad (5.8)$$

Aus der Änderung des Eigendralls der einzelnen Elemente durch die Winkelgeschwindigkeit und -beschleunigung des Laborsystems resultiert ein Anteil des „verbrauchten“ Momentes

$$\mathbf{M}_j^{\text{korr}} = \frac{D'}{Dt}(\boldsymbol{\theta}_{S,j} \boldsymbol{\omega}_O) = \frac{D}{Dt}(\boldsymbol{\theta}_{S,j}) \boldsymbol{\omega}_O + \boldsymbol{\theta}_{S,j} \dot{\boldsymbol{\omega}}_O + \boldsymbol{\omega}_O \wedge (\boldsymbol{\theta}_{S,j} \boldsymbol{\omega}_O). \quad (5.9)$$

Werden $\mathbf{F}_j^{\text{korr}}$ und $\mathbf{M}_j^{\text{korr}}$ an allen Elementen abgezogen, so müssen der Gesamtimpuls und -drall des diskreten Flugkörpers im Laborsystem zu Null werden, d.h. es muß gelten:

$$\sum_{\text{Elemente } j} \mathbf{F}_j^{\text{korr}} = \mathbf{F}_R \quad \text{und} \quad \sum_{\text{Elemente } j} \mathbf{M}_j^{\text{korr}} + \mathbf{r}_{S,j} \wedge \mathbf{F}_j^{\text{korr}} = \mathbf{M}_{OR}. \quad (5.10)$$

Die Gültigkeit der Gleichungen (5.10) soll nachfolgend kurz gezeigt werden:

$$\begin{aligned}
 \sum_{\text{Elemente } j} \mathbf{F}_j^{\text{kor}} &= \sum_{\text{Elemente } j} (\mathbf{a}'_O + \dot{\boldsymbol{\omega}}_O \wedge \mathbf{r}_{S,j} + \boldsymbol{\omega}_O \wedge (\boldsymbol{\omega}_O \wedge \mathbf{r}_{S,j}) + 2\boldsymbol{\omega}_O \wedge \mathbf{v}_{S,j}) \mathbf{m}_j \\
 &= \mathbf{a}'_O \mathbf{m}_{\text{ges}} + \boldsymbol{\omega}_O \wedge \sum_{\text{Elemente } j} (\mathbf{r}_{S,j} \mathbf{m}_j) + \\
 &\quad \boldsymbol{\omega}_O \wedge (\boldsymbol{\omega}_O \wedge \sum_{\text{Elemente } j} (\mathbf{r}_{S,j} \mathbf{m}_j)) + 2\boldsymbol{\omega}_O \wedge \sum_{\text{Elemente } j} (\mathbf{v}_{S,j} \mathbf{m}_j) \\
 &= \mathbf{F}_R,
 \end{aligned}$$

da die drei Summen aufgrund der Übereinstimmung von Ursprung O und momentanem Schwerpunkt entfallen.

$$\begin{aligned}
 \sum_{\text{Elemente } j} \mathbf{M}_j^{\text{kor}} + \mathbf{r}_{S,j} \wedge \mathbf{F}_j^{\text{kor}} &= \sum_{\text{Elemente } j} \underbrace{\left(\left(\frac{D}{Dt} \boldsymbol{\theta}_{S,j} \right) \boldsymbol{\omega}_O + \mathbf{r}_{S,j} \wedge (2\boldsymbol{\omega}_O \wedge \mathbf{v}_{S,j} \mathbf{m}_j) \right)}_{= \left(\frac{D}{Dt} \boldsymbol{\theta}_O \right) \boldsymbol{\omega}_O} \\
 &\quad - \mathbf{a}'_O \wedge \underbrace{\sum_{\text{Elemente } j} (\mathbf{r}_{S,j} \mathbf{m}_j)}_{=0} \\
 &\quad + \underbrace{\sum_{\text{Elemente } j} (\boldsymbol{\theta}_{S,j} \dot{\boldsymbol{\omega}}_O + \mathbf{r}_{S,j} \wedge (\dot{\boldsymbol{\omega}}_O \wedge \mathbf{r}_{S,j} \mathbf{m}_j))}_{= \boldsymbol{\theta}_O \dot{\boldsymbol{\omega}}_O} \\
 &\quad + \underbrace{\sum_{\text{Elemente } j} (\boldsymbol{\omega}_O \wedge (\boldsymbol{\theta}_{S,j} \boldsymbol{\omega}_O) + \mathbf{r}_{S,j} \wedge (\boldsymbol{\omega}_O \wedge (\boldsymbol{\omega}_O \wedge \mathbf{r}_{S,j} \mathbf{m}_j)))}_{= \boldsymbol{\omega}_O \wedge (\boldsymbol{\theta}_O \boldsymbol{\omega}_O)} \mathbf{m}_j \\
 &= \mathbf{M}_{OR}.
 \end{aligned}$$

Ist $\boldsymbol{\omega}_O$ klein und fällt mit einer der Hauptachsen des gesamten Trägheitstensors $\boldsymbol{\theta}_O$ zu allen Zeiten zusammen, so vereinfachen sich die Trägheitskräfte und -momente wie folgt:

$$\mathbf{F}_j^{\text{kor}} = (\mathbf{a}'_O + \dot{\boldsymbol{\omega}}_O \wedge \mathbf{r}_{S,j} + 2\boldsymbol{\omega}_O \wedge \mathbf{v}_{S,j}) \mathbf{m}_j \quad (5.11)$$

und

$$\mathbf{M}_j^{\text{kor}} = \frac{D}{Dt} \mathbf{L}_{S,j} = \frac{D}{Dt} (\boldsymbol{\theta}_{S,j}) \boldsymbol{\omega}_O + \boldsymbol{\theta}_{S,j} \dot{\boldsymbol{\omega}}_O. \quad (5.12)$$

Das durch die Definition der „Mean Axes“ geforderte Verschwinden der Änderung des Gesamtimpulses und -dralls im Laborsystem ist immer noch erfüllt, da die vernachlässigten Zentrifugalkräfte hierzu keinen Beitrag liefern.

Bleibt die Deformation des Körpers in Bezug auf die Rotation $\boldsymbol{\omega}_O$ klein, z.B. wenn der Drehvektor orthogonal zur hauptsächlichen Deformationsebene ist, so kann auch $\frac{D}{Dt} (\boldsymbol{\theta}_{S,j}) \boldsymbol{\omega}_O$ vernachlässigt werden, ebenso wie in diesem Fall näherungsweise $\mathbf{r}_{S,j} = \bar{\mathbf{r}}_{S,j}$ gesetzt werden kann.

5.1.3 Freiflug des starren Flugkörpers

Die Simulation des freifliegenden, starren Flugkörpers ist so ebenfalls möglich. Werden im oben beschriebenen Verfahren alle elastischen Deformationen vernachlässigt, so entsprechen die massenträgheitsgemittelten Gleichungen, die dort gelöst werden, genau den Bewegungsgleichungen des starren Flugkörpers unter dem Einfluß der aerodynamischen Kräfte. Die „Mean Axes“ bilden für diesen Fall ein echt körperfestes Koordinatensystem. Die Korrektur der Knotenlasten des FE-Modells braucht in diesem Fall natürlich nicht durchgeführt zu werden.

5.2 Funktionsbeispiele

In den Abbildungen 5.1–5.3 werden drei Beispielergebnisse wiedergeben, in denen die im Laborsystem berechneten Deformationsbewegungen der gesamten Bewegung im Inertialsystem gegenübergestellt werden. In den Abbildungen wird im oberen Bild jeweils die Gesamtbewegung des elastischen Balkens im Inertialsystem dargestellt, die entsprechend aus Translation und Rotation der Laborkoordinaten und der elastischen Deformation zusammengesetzt ist. In den unteren Bildern wird mit der Verschiebung u_z im Laborsystem die reine Deformation dargestellt.

Der im Vakuum frei schwebende Balken wird im 1. Beispiel (Abb. 5.1) plötzlich mittig belastet, so daß eine symmetrische Deformation und eine reine Translation des Laborsystems zustande kommt. Dagegen erfolgt die Belastung im 2. Beispiel außermittig, bei ansonsten gleichen Bedingungen. Wie in Abb. 5.2 zu sehen ist, erfährt der Balken neben den (nicht mehr symmetrischen) Deformationen und der Translation des Laborsystems auch eine starre Rotation um seinen momentanen Schwerpunkt.

Im 3. Beispiel (Abb. 5.3) ist der elastische Körper nicht frei, sondern rechts in einem Gelenk drehbar gelagert. Dies führt dazu, daß der Ursprung des Laborsystems in dem Gelenk liegt. Da das Laborsystem keine Translationsbewegung mehr ausführen kann, wird zur Bestimmung von dessen Lage nur noch die Änderung des Drehimpulses Gl. (5.5) herangezogen. Auch hier wird eine Drehbewegung mit nichtsymmetrischen Verformungsbewegungen ähnlich Bsp.2 eingeleitet, jedoch bleibt das rechte Ende im Inertialsystem und im Laborsystem fest. Der Unterschied in der Darstellung der beiden Bilder besteht nur noch aus der Drehung des Laborsystems gegenüber dem Inertialsystem im Gelenk.

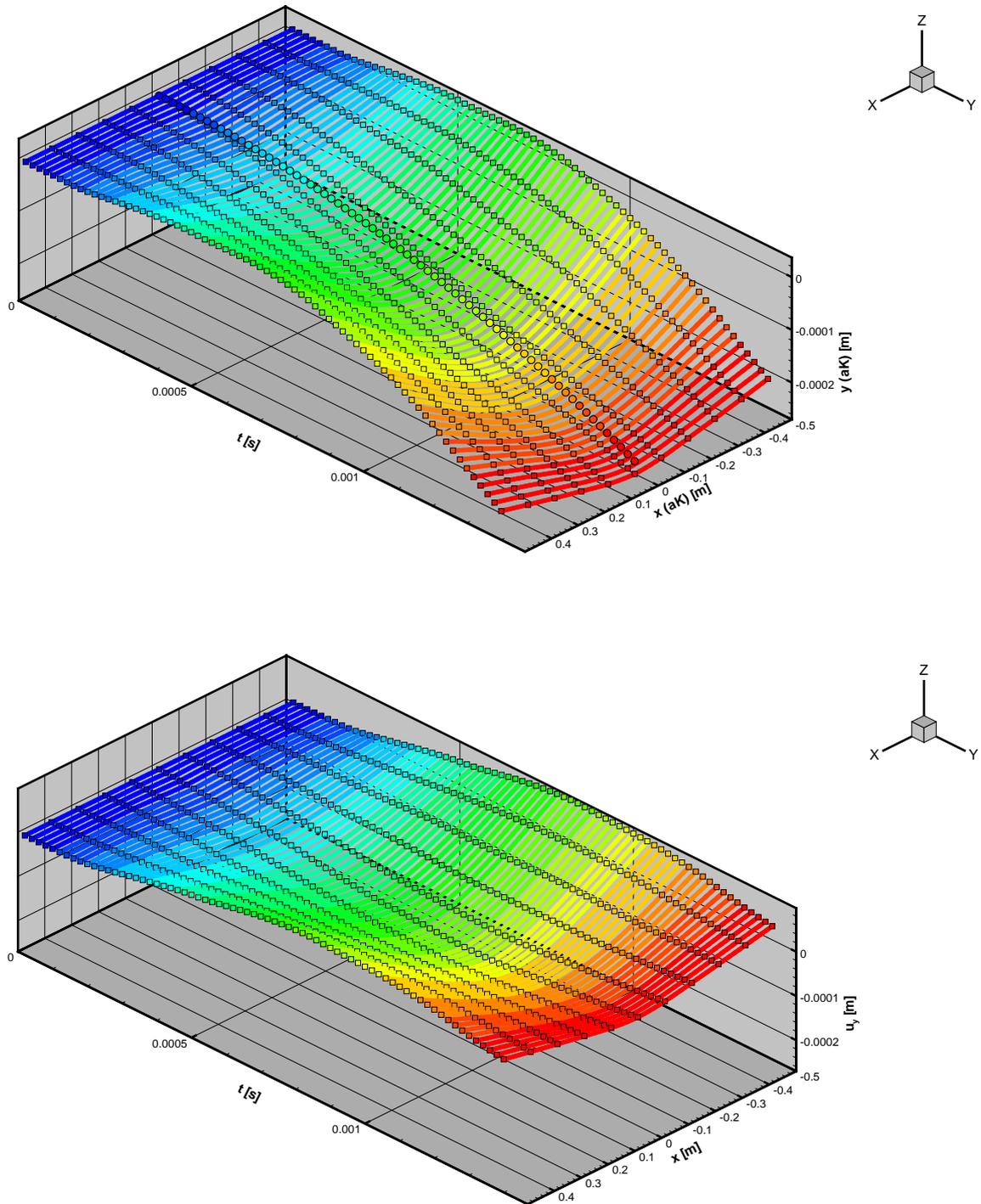


Abbildung 5.1: Beispiel 1: Plötzliche Belastung in negativer z -Richtung auf die Mitte eines freien Stahlbalkens, dessen Achse anfangs mit der x -Achse zusammenfällt. Die kleinen Kreise entlang der Balkenmitte im oberen Bild beschreiben die momentane Lage des Schwerpunktes.

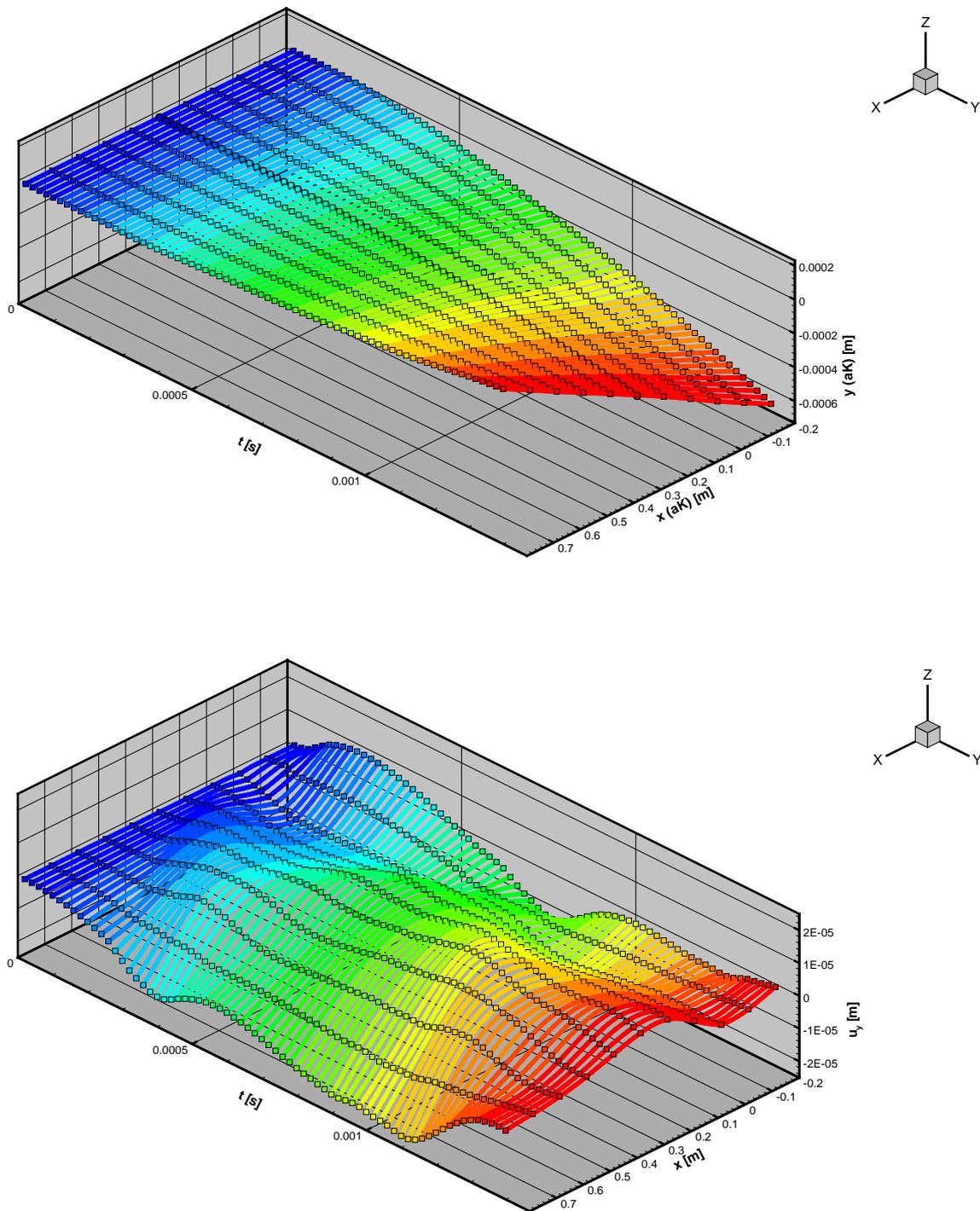


Abbildung 5.2: Beispiel 2: Plötzliche Belastung in negativer z-Richtung auf einen freien Stahlbalken, dessen Achse anfangs mit der x-Achse zusammenfällt, bei 20% der Balkenlänge vom rechten Ende aus, ansonsten wie Bsp.1 (Abb. 5.1). Die Skalen der beiden Abbildungen in z-Richtung sind hier unterschiedlich gewählt worden, da andernfalls die Deformation fast nicht erkennbar wäre.

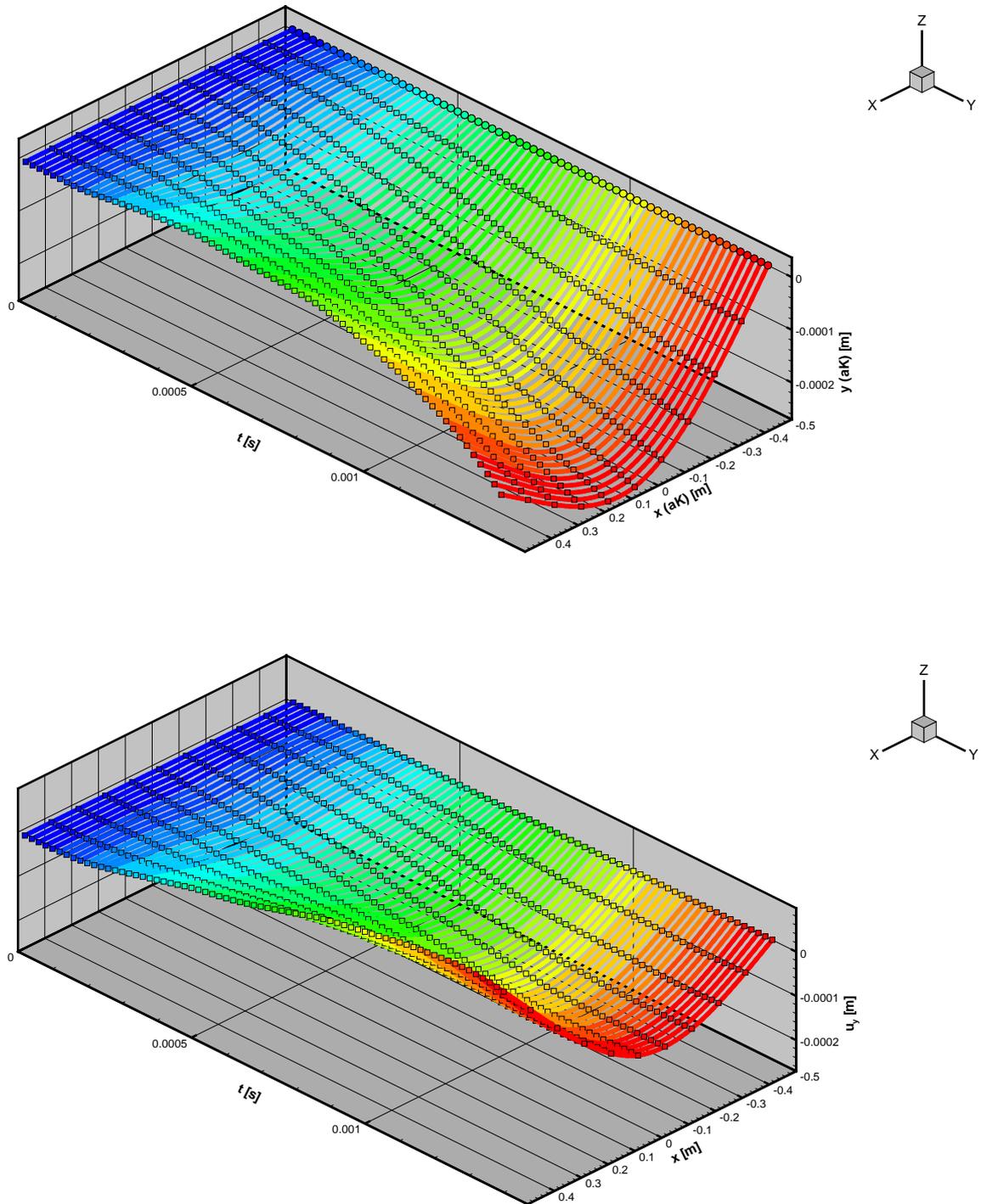


Abbildung 5.3: Beispiel 3: Plötzliche Belastung in negativer z -Richtung auf die Mitte eines Stahlbalkens, der am rechten Ende drehbar gelagert ist und dessen Achse anfangs mit der x -Achse zusammenfällt, ansonsten wie Bsp.1 (Abb. 5.1).

5.3 Modell für Steuerflächen

Mit der oben beschriebenen Entkopplung der trägheitsgemittelten globalen Bewegung des gesamten Körpers von der Deformation, kann nun die Flugbahn eines frei fliegenden, gelenkten Flugkörpers direkt simuliert werden. Aufgrund der einfachen Geometrie im Vergleich zu einem Flugzeug wurde eine über ein Heckleitwerk steuerbare Rakete als Funktionsmuster implementiert.

Zur Steuerung der frei fliegenden Rakete werden vereinfacht modellierte Leitwerke verwendet. Bei diesen wird eine Anstellung gegenüber der Strömung durch eine Verwindung der Leitwerksfläche erreicht, wobei deren Erzeugende in einer Ebene, die die Hauptströmungsrichtung enthält, als drehbar, aber starr angenommen werden. In der Torsionsachse des Leitwerkes wird dazu ein Balkenelement als Träger dieser Erzeugenden angebracht. Zur Kontrolle der Steuerung wird an der Außenkante des Leitwerkes ein Soll-Winkel vorgeschrieben, dessen Vergleich mit dem Ist-Wert eine Veränderung des Momentes ergibt, durch das die Leitfläche tordiert wird. Die zugehörigen Soll-Werte werden als Funktion der Zeit vorgeschrieben. Denkbar und mit sehr geringem Aufwand umsetzbar ist auch eine Steuerung durch bestimmte Parameter des aktuellen Flugzustandes, wie z.B. den Steigwinkel.

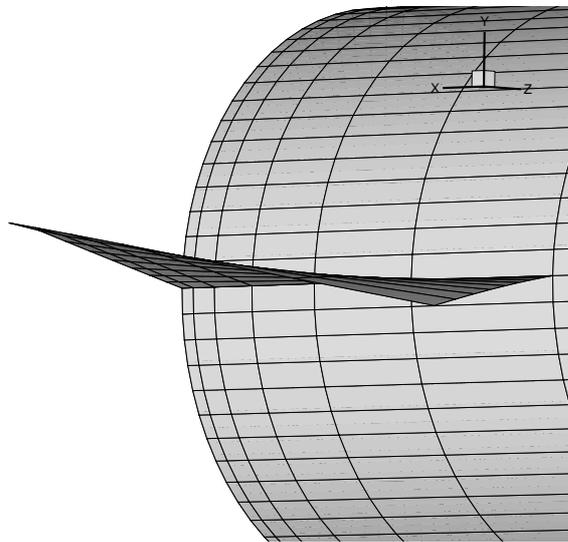


Abbildung 5.4: Darstellung eines Leitwerkes, dessen Tragstruktur (radial verlaufender Balken) zur Steuerung des Flugkörpers einer Torsion unterworfen wird.

Kapitel 6

Kopplung an FLOWer

Beispielhaft wurden die entwickelten Komponenten an den Strömungslöser FLOWer angekoppelt, um so ein vollständiges Verfahren zur aeroelastischen Direktsimulation zu erhalten. Der FLOWer-Code ist ein Finite-Volumen-Verfahren zur Lösung der Euler- oder Reynolds-gemittelten Navier-Stokes-Gleichungen im Bereich von Unter- bis Hyperschall. Zu diesem Zweck sind für den reibungs- und wärmeleitungsfreien Teil der Gleichungen verschiedene Diskretisierungen implementiert: ein zentrales Schema und mehrere Upwind-Schemata. Die Reibung und Wärmeleitung enthaltenden Teile der Navier-Stokes-Gleichungen sind zentral diskretisiert. Durch die Eigenschaften der dargestellten Teilverfahren und die verwendete lose Kopplung war es möglich, eine Programmbibliothek zu erstellen, die über nur wenige Schnittstellenfunktionen verfügt. Die Änderungen an FLOWer sollten möglichst gering und lokal ausfallen, damit die Erweiterung schnell verfügbar gemacht werden konnte. Eine vollständige Beschreibung der FLOWer-seitigen Änderungen ist in [Han03] enthalten.

6.1 Kopplungsschema

Die vorangehend beschriebenen Verfahren zur Berechnung der instationären Strukturmechanik und der Strömungslöser FLOWer müssen so miteinander gekoppelt werden, daß eine zeitgenaue Lösung des aeroelastischen Gesamtsystems errechnet werden kann. Da das Aufstellen und Lösen der vollständigen Differentialgleichungen für dieses System aufgrund der enormen Komplexität wenig Aussicht auf Erfolg bietet, werden im Rahmen dieser Arbeit die oben beschriebenen, separat formulierten Verfahren so miteinander gekoppelt, daß das gleiche Ziel erreicht wird.

Da in der Wirklichkeit die beiden Systeme sich zu jedem Zeitpunkt gegenseitig beeinflussen, liegt es auf der Hand, daß mit einem einfachen Alternieren der beiden Verfahren keine zeitgenaue Lösung erreicht werden kann. In [Hur02] findet sich eine Untersuchung derartiger Kopplungsvorgänge, insbesondere mit Hinblick auf eine einfache Realisierung bei gleichzeitig zeitgenauer Bestimmung der Lösung. Dabei wurde festgestellt, daß bereits eine kleine Verbesserung des abwechselnden Rechnens den gewünschten Effekt erzielt.

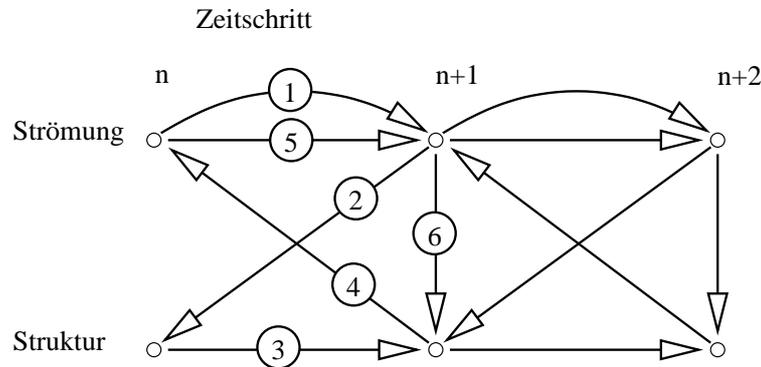


Abbildung 6.1: Kopplungsschema nach Hurka²: 1) Extrapolation der Strömungsdaten, 2) Übertragung der extrapolierten Daten an die Struktur, 3) Lösungsschritt Struktur, 4) Übertragung der Strukturdaten an die Strömung, 5) Lösungsschritt Strömung, 6) Übertragung der berechneten Strömungsdaten an die Struktur.

Dazu wird in jedem Zeitschritt zu Beginn eine Extrapolation der aerodynamischen Belastungen des Flugkörpers für das Ende des geplanten Zeitschrittes vorgenommen. Mit diesen Daten wird die strukturdynamische Lösung bestimmt. Die so bestimmte Deformation dient als Grundlage für das Fortschreiten der aerodynamischen Lösung. Die aus dieser Lösung berechneten Lasten werden dann an den Strukturlöser übergeben (vgl. Abb. 6.1). Prinzipiell kann dieses Vorgehen nun noch mehrfach unter Auslassung des Extrapolationsschrittes iteriert werden, in [Hur02] wurde jedoch festgestellt, daß das Ergebnis dadurch in der Regel nicht mehr nachhaltig verbessert wird. Diese Feststellung wurde im Rahmen einer „monolithischen“ Formulierung in [Mas02] weitgehend bestätigt.

6.2 Stationäre Windkanalströmung und Startwertebe- rechnung

Mit der oben erläuterten Kopplung können auch schnell stationäre Probleme für elastische Flugkörper gelöst werden. So stellt sich z.B. bei einem im Windkanal eingespannten Flugkörper im Fall, daß eine stabile Gleichgewichtslage und -konfiguration existiert, ein stationärer Zustand für die Strömung und die Deformation des Flugkörpers ein. Ähnliches gilt für die Berechnung des Anfangszustandes für eine dynamische Simulation, die mit einer ausgelenkten Lage begonnen werden soll.

In diesen Fällen wird das gekoppelte Programm mit leichten Veränderungen verwendet. Sowohl der Strömungs- als auch der Strukturlöser benutzen dann anstelle der dynamischen Lösungsroutinen solche zur Bestimmung der statischen Lösung. Da eine zeitgenaue Lösung in diesem Fall ebenso nicht er-

²Schema B in [Hur02]

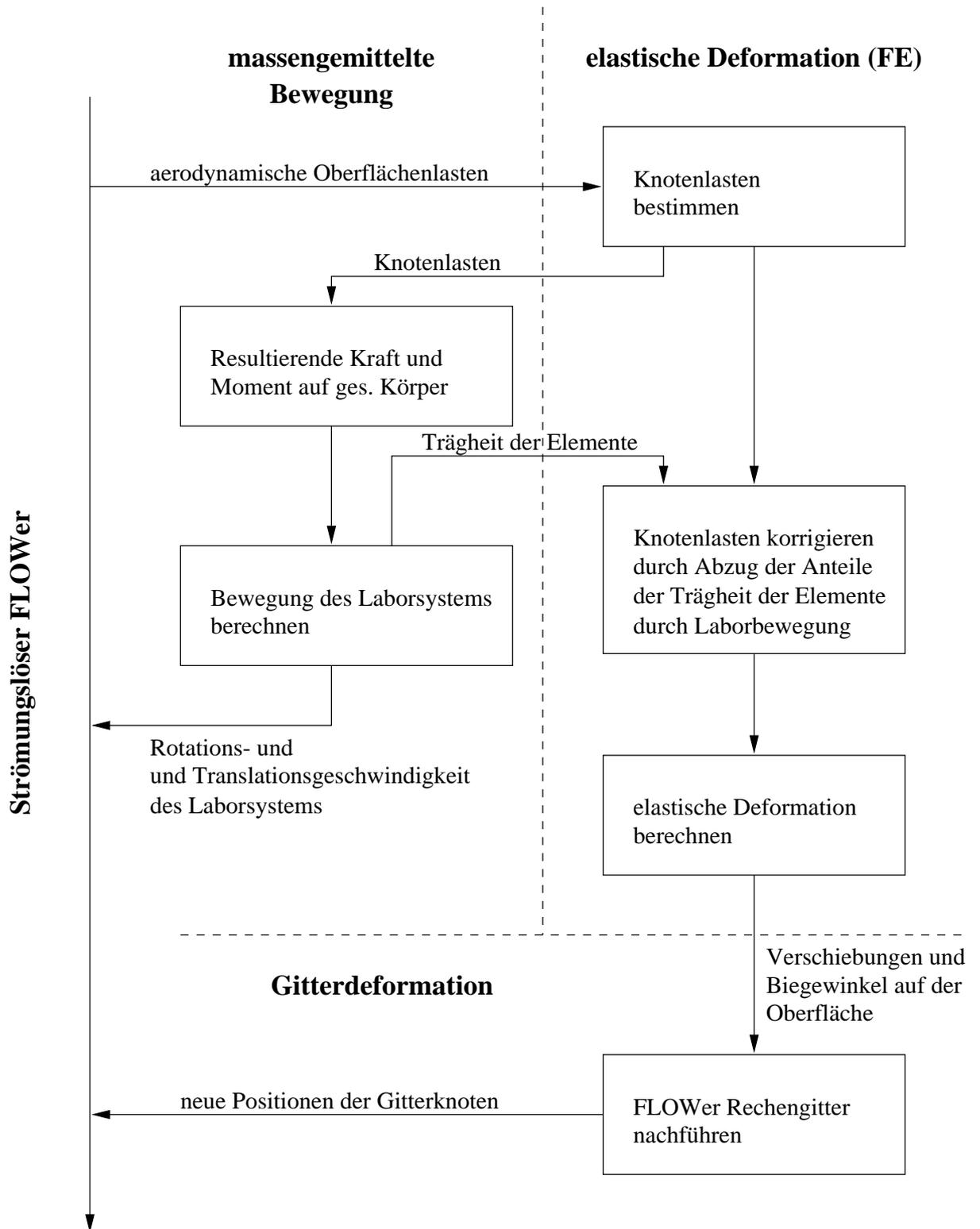


Abbildung 6.2: Datenfluß in vereinfachter Darstellung für die Kopplung zwischen der Strömung und einem elastischen Festkörper mit starrem Freiheitsgrad.

forderlich ist, vereinfacht sich das Kopplungsschema (vgl. Abb. 6.1) auf ein schlichtes Abwechseln der beiden Verfahren, d.h. der für die instationäre Berechnung vorgenommene Schritt der Extrapolation entfällt.

Da in der Regel 3-6 Iterationen ausreichend sind, um die stationäre Lösung in guter Näherung zu erreichen, stellt diese Modifikation eine enorme Beschleunigung gegenüber dem alternativen Weg dar, in dem eine dynamische Lösung mit einem stark gedämpften System bis zum Erreichen der Ruhelage berechnet wird. Allerdings erfordert unsere Vorgehensweise die Voraussetzung, daß es eine stabile Gleichgewichtskonfiguration gibt, die sich nur durch kleine Verschiebungen von der Anfangslage unterscheidet.

6.3 Schnittstelle der Aeroelastik-Programmbibliothek für FLOWer

Da die im FLOWer-Code vorzunehmenden Änderungen möglichst gering ausfallen sollen, wurden alle Funktionen für die Berechnung der Strukturdeformation, der Gitterdeformation, der Flugbahnbestimmung, sowie der Datenübertragung in einer externen Programmbibliothek zusammengefaßt, die neben der Einbindung in FLOWer auch die Erstellung separater Programme, z.B. zur Gitterdeformation für einen gegebenen Last- oder Deformationszustand erlaubt. Die Schnittstelle der Programmbibliothek enthält nur wenige Funktions- bzw. Prozeduraufrufe, die eine hohe Abstraktionsebene aufweisen. Auf diese Art wird eine weitgehende Entkopplung von FLOWer und den neuen Programmteilen hinsichtlich der Implementierung erreicht. Diese starke Entkopplung wurde an einigen Stellen mit dem Verlust an Laufzeiteffizienz und mit der Speicherung redundanter Daten erkauft, der Gewinn an Sicherheit und Flexibilität bei der Implementierung wiegt dies jedoch wieder auf.

Im folgenden werden die einzelnen Funktionen der Schnittstelle beschrieben:

Init Initialisierung der benötigten Daten z.B. durch Einlesen der Steuerdateien. Zentrale Steuerdaten, wie Einlesen von Restart-Daten und deren Ausgabe, sowie die Fallunterscheidung, ob mit bzw. ohne Elastizität des Körpers gerechnet werden soll, werden hier ausgetauscht.

Solve Bestimmung der Lösung für den gegebenen Zeitpunkt t . Da im FLOWer Zeitschritte unterschiedlicher Länge verwendet werden können, wurde für die Lösungsroutinen für die Strukturdynamik ein Runge-Kutta-Verfahren mit Fehlerkontrolle und automatischer Schrittweitensteuerung [PFTV86] implementiert. So braucht auf die mögliche Schrittweite für den Strukturlöser keine Rücksicht genommen zu werden. Durch das Runge-Kutta-Verfahren werden ggf. zu große Zeitschritte des FLOWer-Codes durch mehrere ausreichend kleine Schritte überbrückt.

Set.Forces Addiert einen Kraftvektor an den gegebenen Koordinaten. Die Lasten werden in einem Lastvektor akkumuliert, der für das Ende des nächsten Zeitschrittes verwendet wird.

Move_Loads_to_Current_Time Um bei der Iteration mit dem gegebenen Kopplungsschema neue Lastdaten für den aktuellen Zeitpunkt setzen zu können (z.B. die berechneten Luftlasten, Schritt 6) wird der Lastvektor, in dem die Lasten via Set_Forces akkumuliert wurden, an den aktuellen Zeitpunkt verschoben und für das Ende des folgenden Zeitschrittes ein neuer Lastvektor mit Null initialisiert.

Resolve Wiederholung des letzten Zeitschrittes, z.B. bei der Reiteration des Kopplungsschemas mit anderen Lasten oder bei der Korrektur der Zeitschrittweite nach unten.

Output Ausgabe des aktuellen Zustandes des elastischen Modells oder von Restart-Daten für den Festkörper.

Has_Rigid_DOF Abfrage, ob der Flugkörper Starrkörperfreiheitsgrade besitzt. Dies wird durch das Einlesen der Beschreibung des Modells in der Eingabedatei automatisch festgestellt. Die Unteroutine soll FLOWer die entsprechende Information zukommen lassen, damit entsprechende Zweige bei der Strömungsberechnung für starre Rotationen und Translationen durchlaufen werden können.

Rigid_Movement_Centre Ursprung der „Mean Axes“ als Koordinaten der BK.

Rigid_Movement Verschiebung, Verdrehung und die zugehörigen Geschwindigkeiten des Laborsystems gegenüber dem Inertialsystem (=Rigid_Movement_Centre).

Init_Moving_Grid Initialisierung des Gitterdeformationsprogramms.

Calculate_New_Grid Bestimmung der Gitterdeformation zum aktuellen Zeitpunkt und Erstellen des neuen Gitters.

Kapitel 7

Simulationsergebnisse

7.1 Tragflügel quasistatisch

Im Rahmen des von der Europäischen Union geförderten Forschungsvorhaben „High Reynolds-Number Tools & Techniques Project“ wurden Windkanalversuche mit einer elastischen Flügel-Rumpf-Kombination im Europäischen Transsonischen Windkanal (ETW) durchgeführt. Bei der Vornahme begleitender Simulationsrechnungen zu den Messungen wurden von C. Braun die in dieser Arbeit vorgestellten Techniken benutzt [BBB⁺03]. Es zeigte sich, daß für die Bewertung der Versuche bzw. die Validierung numerischer Simulationsergebnisse die korrekte Wiedergabe des im Windkanal vorliegenden stationären Verformungszustandes des Windkanalmodelles wesentlich ist. In den Abb. 7.3 und 7.4 sind die Druckbeiwerte für verschiedene Anstellwinkel in sieben spannweiten Profilschnitten des Tragflügels wiedergegeben¹. Deutlich erkennbar ist, daß das Ergebnis der auskonvergierten stationären aeroelastischen Lösung (durchgezogene Linien) sehr gut mit den Windkanalmessungen (Kästchen) korreliert, während die Simulation ohne Beachtung der Deformation des Flügels teilweise erhebliche Abweichungen der Druckverteilung liefert, die durch die dimensionslose Größe c_p („Druckbeiwert“) dargestellt wird.

Zur besseren Anschaulichkeit wird in Abb. 7.1 eine Teilansicht der Flügel-Rumpf-Konfiguration gezeigt, in der der deformierte Flügel zu verschiedene Punkten der quasi-stationären Iteration eingetragen ist. Weiterhin wird in Abb. 7.2 das deformierte Gitter um den Flügel herum gezeigt.

¹Anstellwinkel und die Absolutwerte der Druckbeiwerte können aus Geheimhaltungsgründen nur normiert wiedergegeben werden.

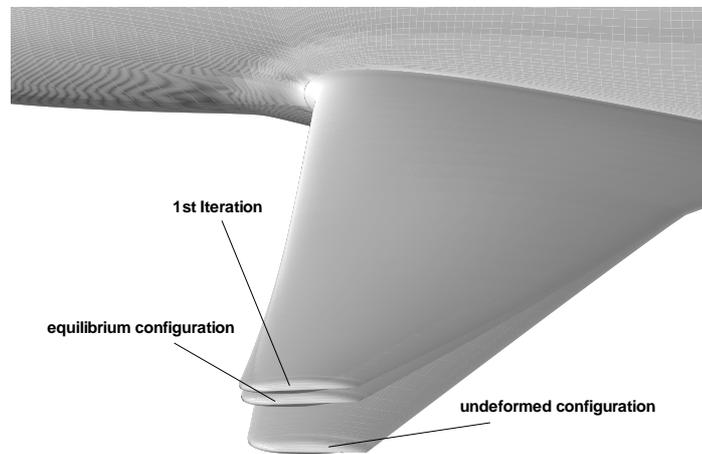


Abbildung 7.1: Form des Flügels zu Beginn der Iteration, nach dem ersten Schritt und im Endzustand (Gleichgewichtszustand).

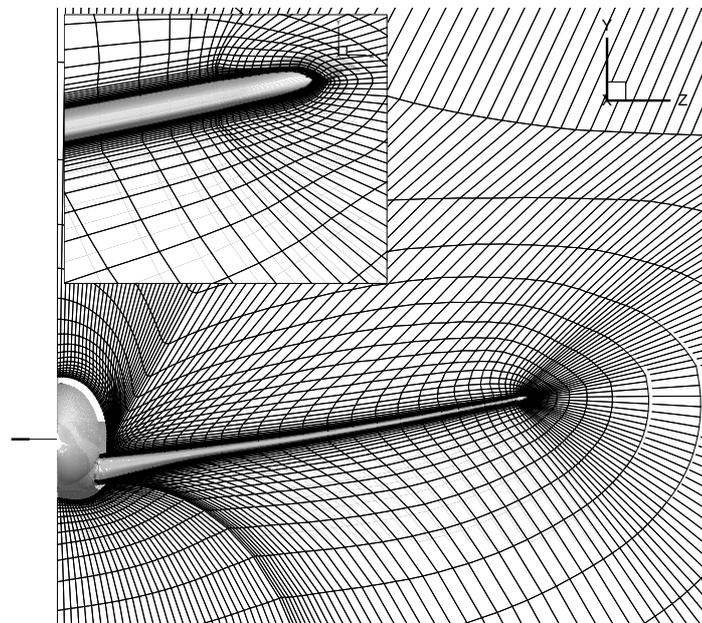


Abbildung 7.2: Eine Gitterebene parallel zur Flügelachse im deformierten Zustand.

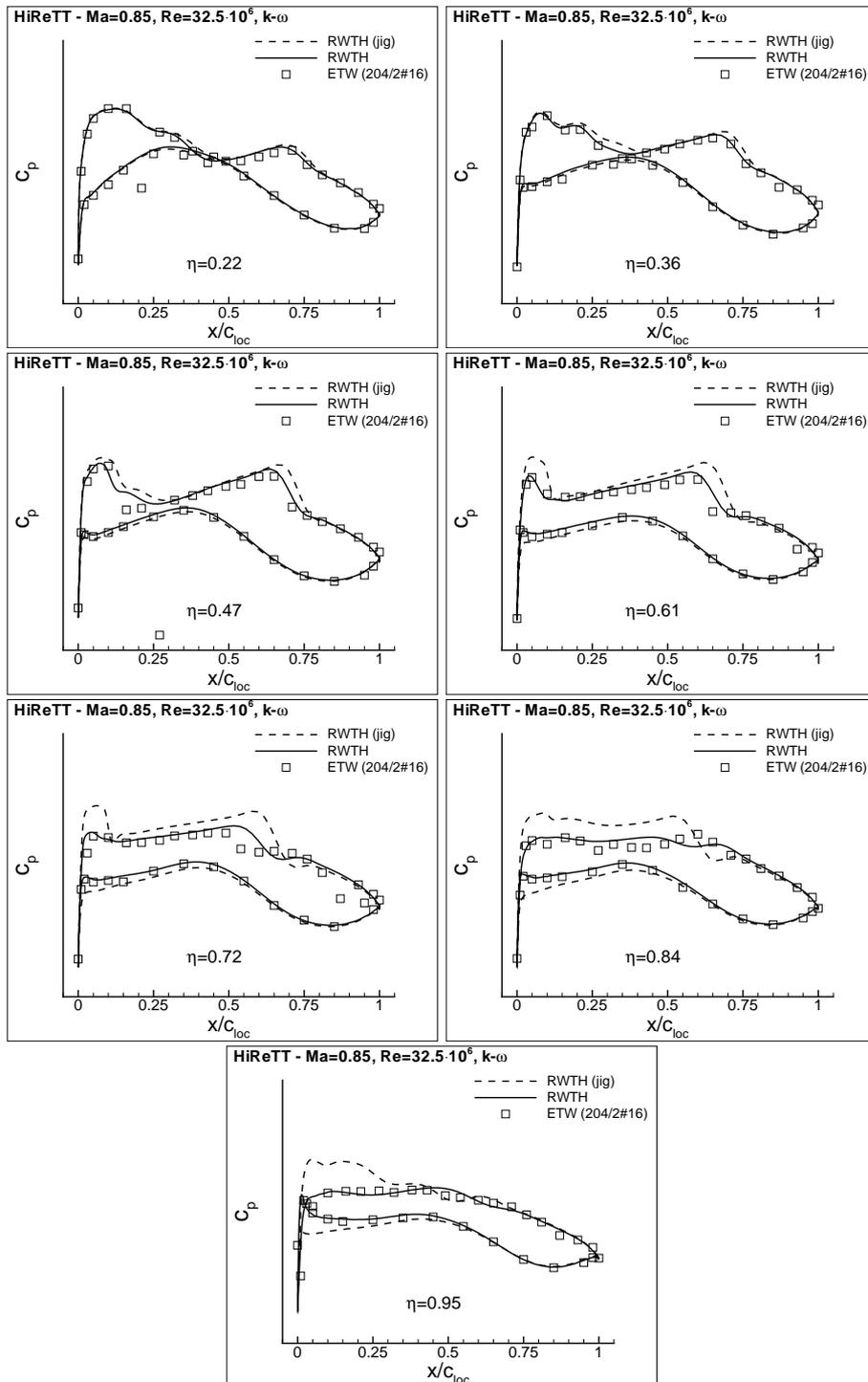


Abbildung 7.3: Vergleich der Druckbeiwerte zwischen Messung (Kästchen) und Simulation ohne Berücksichtigung (gestrichelt) bzw. mit Berücksichtigung der Modellelastizität (durchgezogen) in verschiedenen Schnitten des Tragflügels, Reiseflugbedingungen.

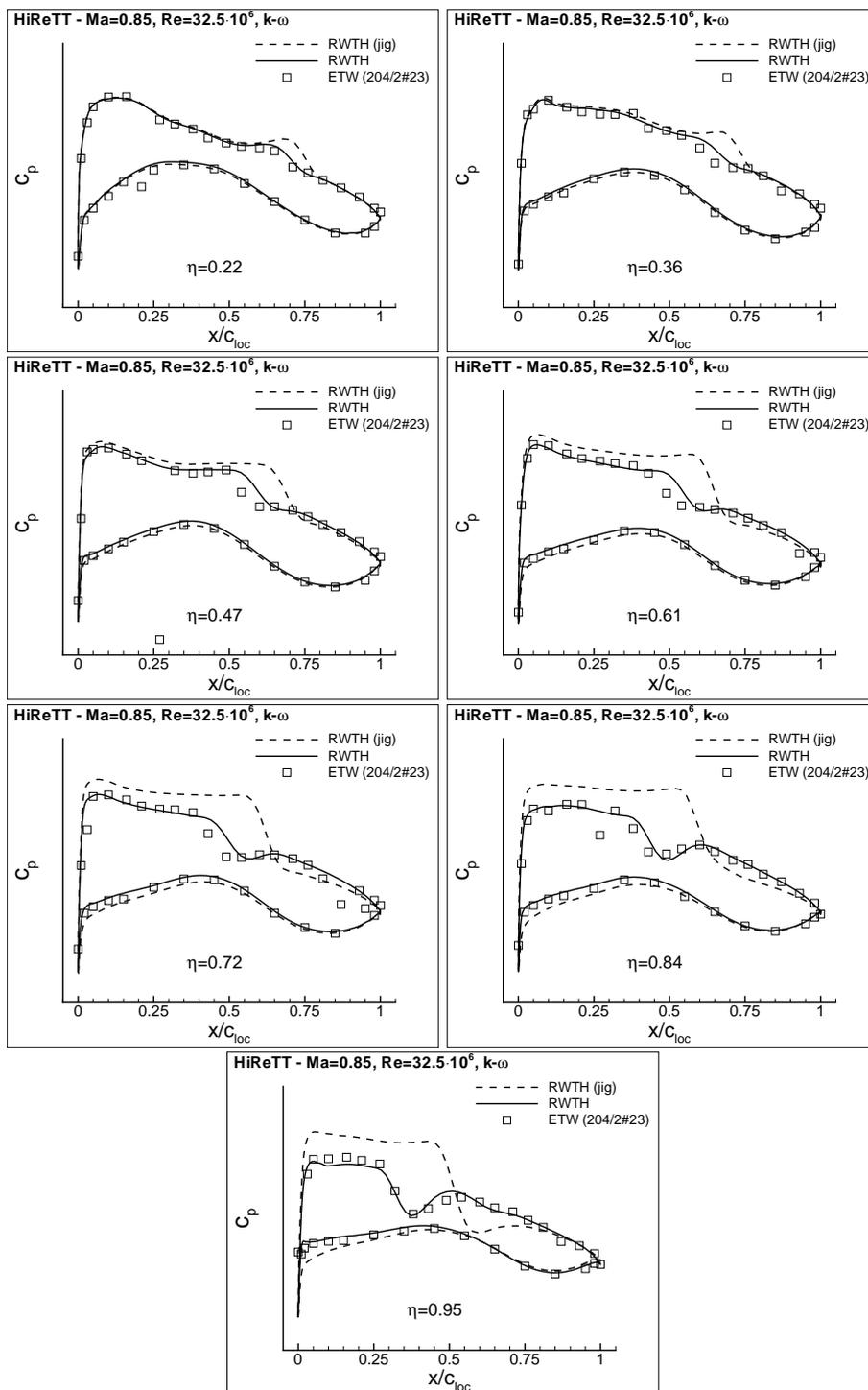


Abbildung 7.4: Vergleich der Druckbeiwerte zwischen Messung (Kästchen) und Simulation ohne Berücksichtigung (gestrichelt) bzw. mit Berücksichtigung der Modellelastizität (durchgezogen) in verschiedenen Schnitten des Tragflügels, Daten wie in Abb. 7.3, jedoch größerer Anstellwinkel.

7.2 Lenkflugkörper elastisch mit Lenkbewegung im Freiflug

Zusammen mit M. Hanke wurde ein einfacher Lenkflugkörper, basierend auf den Daten aus [TÖG00] untersucht. Aus dieser Reihe wird nachfolgend eine Simulation, in der alle implementierten Bestandteile der Kopplungsbibliothek verwendet wurden, vorgestellt, für weitere Ergebnisse siehe [Han03].

Bei dem Flugkörper handelt sich um einen einfachen Drehkörper mit konischem, an der Spitze abgerundeten Bug. Am Heck sind vier ebene, dünne Leitwerke angebracht. Im aerodynamischen Modell wurden die Leitwerke nur als Teil einer nicht durchströmten Gitterebene modelliert, d.h. sie haben kein Volumen. Für die hier gezeigte Simulation wurde das Modell zu einem Lenkflugkörper erweitert, indem die Leitwerke wie in Abschnitt 5.3 beschrieben tordiert werden. Die Simulation wurde dreidimensional an einem vollständigen Modell durchgeführt.

Um den Einfluß von Lenkbewegungen während des Fluges auf die Flugbahn und die Erregung starrer und elastischer Freiheitsgrade zu untersuchen, wurde ein Ausschlag der seitlichen Leitwerksflügel als Höhenruder simuliert. Als Vorgabe wurde verlangt, die in der xz -Ebene liegenden Flügel binnen 0.01s von 0° auf 10° zu verstellen und diese Anstellung für die Dauer von 0.1s beibehalten. Danach sollen die Leitwerksflügel wieder innerhalb von 0.01s in die Ausgangsstellung gebracht werden. Die dieser Vorgabe folgende Klappenbewegung ist in Abb. 7.5 zu sehen.

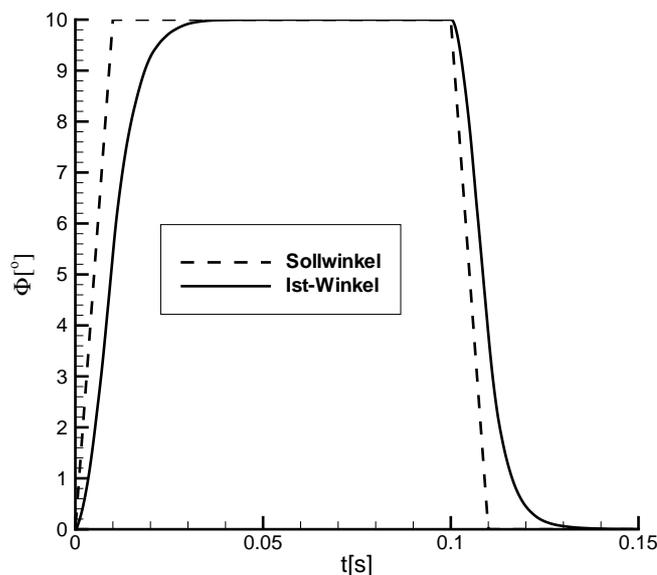


Abbildung 7.5: Freiflug mit Lenkmanöver, zeitlicher Ablauf der Leitwerksbewegung.

Die Simulation startet mit einer stationären Strömung für eine Mach-

zahl 1.5 und ohne Anstellung. Die Erdschwere wird hier vernachlässigt. Der Flugkörper befindet sich im stationären Reiseflug in seiner Achsenrichtung, so daß er keine Verformungen besitzt und auch die Verformungsgeschwindigkeiten Null sind, d.h. es finden keine elastischen Schwingungen statt. Durch die schnelle Lenkbewegung wird nun eine Veränderung der aerodynamischen Lasten herbeigeführt und damit einhergehend auch eine Anregung elastischer Schwingungen des Flugkörpers.

Betrachtet man den Zeitverlauf des Auftriebsbeiwertes in Abb. 7.10 auf Seite 79 oben, dann erkennt man, wie die Höhenruder zunächst leichten Abtrieb bewirken. Durch die einsetzende Drehung des Flugkörpers um die Querachse in z -Richtung und den dadurch erzeugten effektiven Anstellwinkel entstehen auf der Unterseite des Flugkörpers, vor allem am Bug, höhere Luftkräfte und damit Auftrieb. Dieser überwiegt ab $t = 0.07\text{s}$ den am Leitwerk erzeugten Abtrieb (vgl. Abb. 7.6). Auf diese Weise erhält die Flugbahn des Massenschwerpunktes eine Komponente in y -Richtung, siehe Abb. 7.11 oben links (u_y), die zunächst negativ wird und dann stetig ansteigt. Nach der Simulationszeit $t = 0.1\text{s}$ werden die Flügel wieder in die Ausgangslage gebracht. Dadurch erhöht sich die Luftangriffsfläche und der Auftrieb steigt deutlich an: in Abb. 7.11 oben rechts hat die Auftriebskraft F_y an dieser Stelle einen Knick. Zeitgleich bewirkt der Auftrieb am Leitwerk ein positives Nickmoment M_z (Abb. 7.11 unten links), das als Rückstellmoment wirkt. Dadurch wird die Pendelbewegung abgebremst und es kommt nicht zu einem Überschlag. Der Pendelwinkel ϕ_z fällt ab einer Zeit von $t = 0.25\text{s}$ wieder, so daß nach einigen weiteren Pendelschwingungen eine Flugrichtungsänderung vollzogen sein wird.

Die elastischen Schwingungen beeinflussen diesen Vorgang auch nicht negativ. Die Verformungen sind in Abb. 7.9 oben dargestellt. Es zeigt sich, daß Schwingungen angeregt wurden, die etwa zur Zeit des größten effektiven Anstellwinkels die höchsten Amplituden erreichen. Danach klingen sie monoton wieder ab.

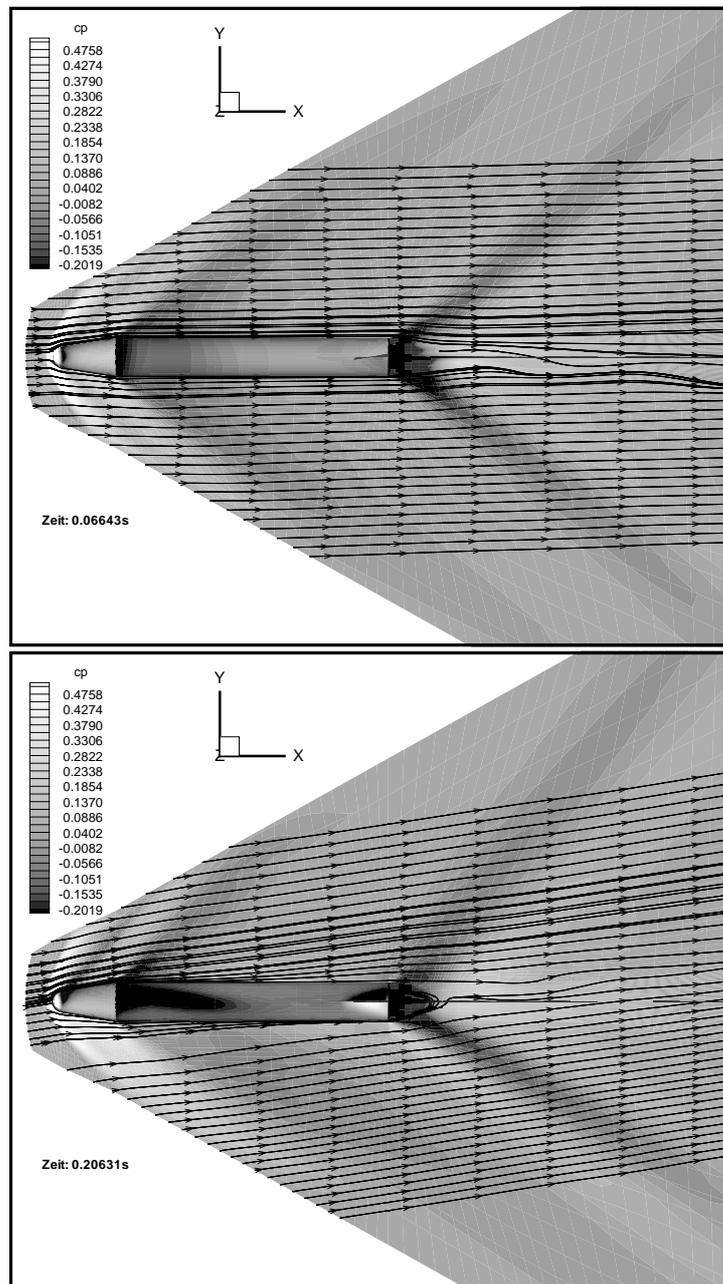


Abbildung 7.6: Freiflug eines elastischen Drehkörpers mit elastischen Flügeln, Anstellung $\alpha_0 = 0^\circ$, Leitwerklenkbewegung 10° , Dauer 0.1s, $M_\infty = 1.5$, Konturdarstellung der c_p -Verteilung in der Anstellebene. Stromliniendarstellung der x - und y -Komponenten der Geschwindigkeit in der Symmetrieebene. Das Koordinatensystem xyz bezeichnet die Lage der Laborkoordinaten zur Zeit 0.0664387s (oberes Bild) und zur Zeit 0.20631s (unteres Bild).

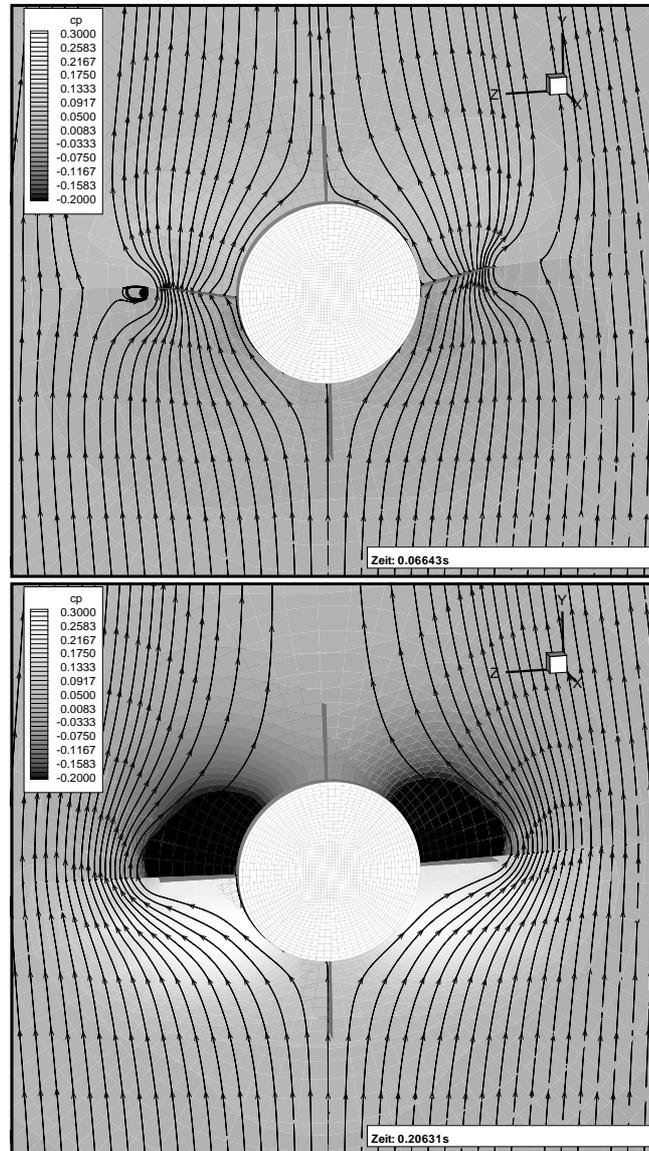


Abbildung 7.7: Freiflug eines elastischen Drehkörpers mit elastischen Flügeln, Anstellung $\alpha_0 = 0^\circ$, Leitwerklenkbewegung 10° , Dauer 0.1s, $M_\infty = 1.5$, Blick auf den Heckbereich von hinten auf eine Koordinatenfläche senkrecht zur Körpermittellachse bei $x = 0.01m$. Stromliniendarstellung der y - und z -Komponenten der Geschwindigkeit. Konturdarstellung der c_p -Verteilung zur Zeit 0.0664387s (oberes Bild) und zur Zeit 0.20631s (unteres Bild). Das Koordinatensystem xyz stellt die Lage der Laborkoordinaten zur entsprechenden Zeit dar.

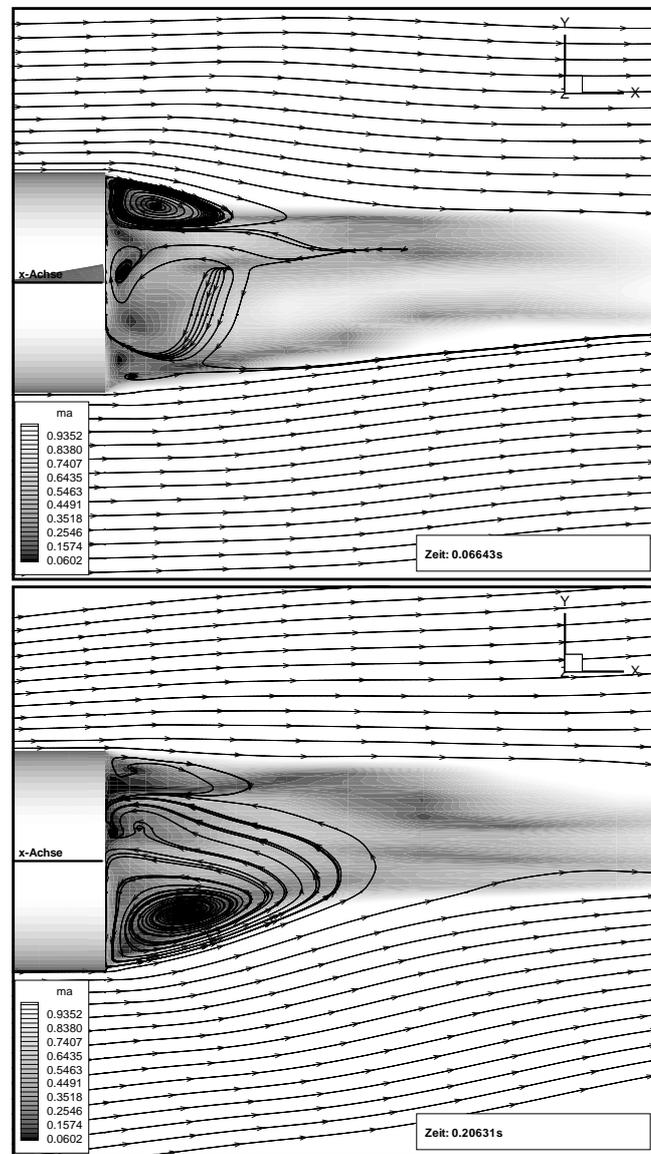


Abbildung 7.8: Freiflug eines elastischen Drehkörpers mit elastischen Flügeln, Anstellung $\alpha_0 = 0^\circ$, Leitwerklenkbewegung 10° , Dauer 0.1s, $M_\infty = 1.5$, Konturdarstellung der Machzahl-Verteilung in der xy -Ebene. Stromliniendarstellung der x - und y -Komponenten der Geschwindigkeit. Ausschnitt des Heck- und Nachlaufbereiches zur Zeit 0.0664387s (oberes Bild) und zur Zeit 0.20631s (unteres Bild). Weiße Flächen entsprechen Überschallgebieten. Das Koordinatensystem xyz stellt die Lage der Laborkoordinaten zur entsprechenden Zeit dar.

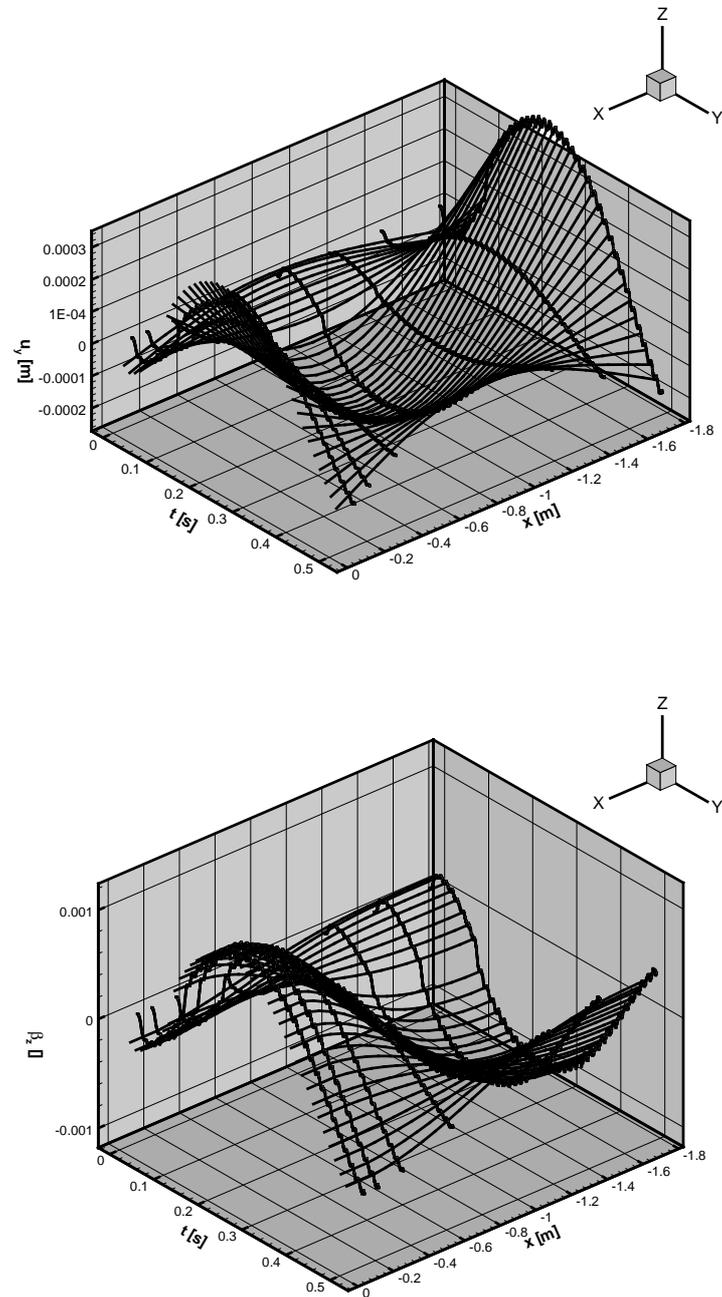


Abbildung 7.9: Freiflug eines elastischen Drehkörpers mit elastischen Flügeln, Anstellung $\alpha_0 = 0^\circ$, Leitwerklenkbewegung 10° , Dauer 0.1s, $M_\infty = 1.5$, Zeitliche Entwicklung der elastischen Verschiebung u_y und des Biegewinkels β_z der Deformation innerhalb der Laborkoordinaten.

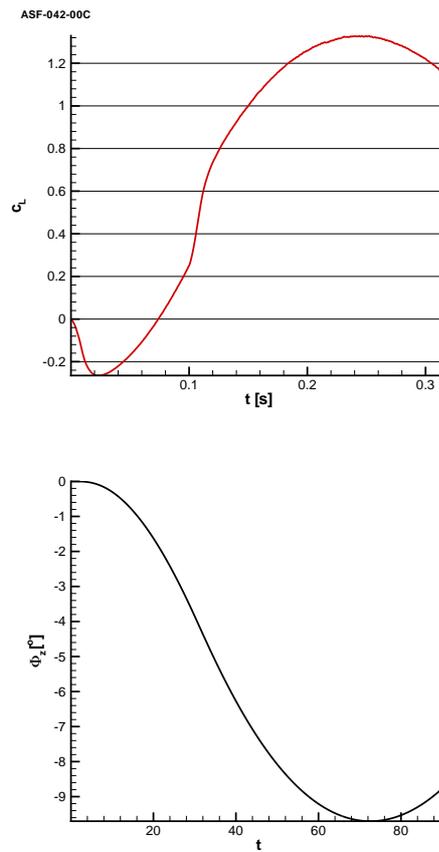


Abbildung 7.10: Freiflug eines elastischen Drehkörpers mit elastischen Flügeln, Anstellung $\alpha_0 = 0^\circ$, Leitwerklenkbewegung 10° , Dauer 0.1s, $M_\infty = 1.5$, Zeitliche Entwicklung des Auftriebsbeiwertes c_L und der Rotationswinkels ϕ_z der Laborkoordinaten gegenüber dem Inertialsystem.

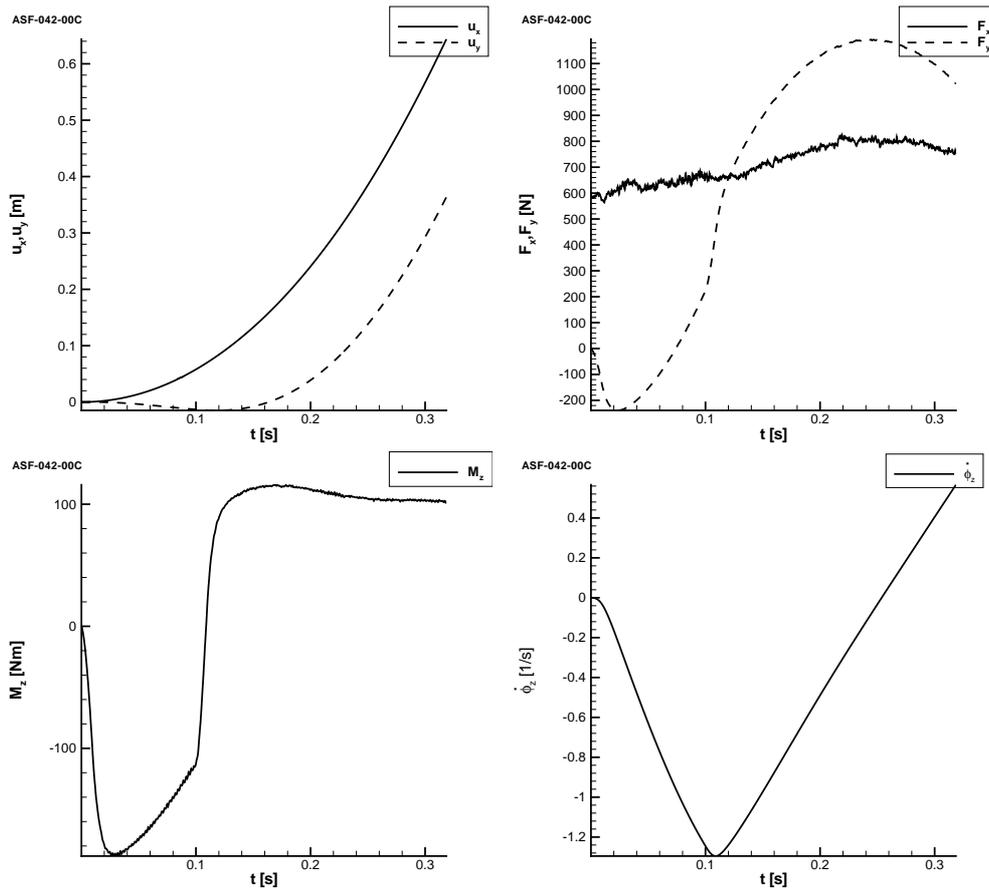


Abbildung 7.11: Freiflug eines elastischen Drehkörpers mit elastischen Flügeln, Anstellung $\alpha_0 = 0^\circ$, Leitwerklenkbewegung 10° , Dauer 0.1s, $M_\infty = 1.5$, Zeitliche Entwicklung der Verschiebungen u_x, u_y ($u_z = 0$) der Laborkoordinaten relativ zum mit der Anfangsfluggeschwindigkeit gleichförmig mitbewegten Inertialsystem (oben links), der resultierenden Luftkräfte F_x, F_y (oben rechts), des resultierenden Luftmoments M_z (unten links) und der Winkelgeschwindigkeit $\dot{\phi}_z$ (unten rechts)

Kapitel 8

Zusammenfassung und Ausblick

Mit den in dieser Arbeit formulierten und implementierten Werkzeugen für aeroelastische Simulationen ist es gelungen, in Zusammenarbeit mit M. Hanke den Strömungslöser `FLOWer` zu einem für vielfältige Anwendungen der aeroelastischen Direktsimulation geeigneten Programm zu erweitern (siehe auch [Han03]). Auch wenn die Erfahrungen im Umgang mit dem erweiterten Programmpaket noch nicht sehr groß sind, da derartige Simulationsberechnungen immer sehr viel Zeit in Anspruch nehmen, so hat es sich bisher als ausgesprochen robust erwiesen und wurde sehr erfolgreich eingesetzt, ohne daß sich bisher ausgeprägten Schwächen gezeigt hätten.

Alle in dieser Arbeit beschriebenen Verfahren sind in einer Programmibibliothek implementiert worden, die mit dem `FLOWer`-Code gekoppelt worden ist, um die Simulation von dreidimensionalen Strömungen um Fluggeräte mit flexibler Geometrie zu ermöglichen. Einen zentralen Punkt nimmt dabei das neuartige Gitterdeformationsverfahren mittels fiktiver, räumlicher Rahmen-tragwerke ein, da für die strukturierten Mehrblocknetze von `FLOWer` bisher kein Programm zur Gitternachführung existierte, das allgemein anwendbar war und deformierte Gitter von ausreichend hoher Qualität lieferte. Von dem Gitterdeformationsverfahren wurde auch ein einzeln lauffähiges Programm erstellt, mit dem ein Rechengitter gemäß einer vorgegebenen Deformation oder Belastung des elastischen Flugkörpers oder Deformationen von Komponenten eines Fluggerätes aus dem nicht deformierten Gitter einer Bezugskonfiguration (jig shape) gewonnen werden kann.

Unter Einsatz der in der Arbeit vorgestellten Simulationswerkzeuge konnte bereits eine Vielzahl von aeroelastischen Simulationen erfolgreich durchgeführt werden. Untersucht wurde sowohl das stationäre, als auch das instationäre Verhalten verschiedener elastischer Tragflügel. Diese Simulationen dienten zum Teil der Validierung des Gesamtverfahrens anhand von Messungen für den stationären aeroelastischen Zustand einer Rumpf-Flügel-Konfiguration in transsonischer Strömung (s. Abschnitt 7.1, [BBB⁺03]). Die von M. Hanke in den `FLOWer`-Code eingefügten Erweiterungen für die instationäre aeroelastische Simulation unter Verwendung der beschriebenen Programmibi-

bliothek wurden von ihm selbst für eine instationäre Schwingung eines elastischen Tragflügels validiert [Han03]. Anhand eines einfachen Lenkflugkörpers wurden mit dem Verfahren weitergehende aeroelastische Untersuchungen durchgeführt, wie z.B. auch in Abschnitt 7.2 dargestellt. Dazu gehört ferner eine Methode zur Bestimmung aeroelastischer Derivativa mittels aeroelastischer Direktsimulation [Han03].

Das vorgestellte Verfahren zur Gitterdeformation ist, u.a. weil es unstrukturierte Gitter und hängende Knoten zulässt, nicht nur auf die Anwendung mit dem auf strukturierten Gittern operierenden Strömungslöser `FLOWer` beschränkt. So ist geplant, es auch in Verbindung mit dem im SFB 401 in Entwicklung befindlichen, adaptiven Löser `Quadflow` einzusetzen. Gleiches gilt für den erstellten Strukturlöser und die Routinen zur Übertragung von Lasten und Deformationen zwischen den Programmen. Erste Schritte zur Erstellung einer an die Erfordernisse von `Quadflow` angepassten Schnittstelle zu der Programmbibliothek sind bereits unternommen worden. Da das Gitterdeformationsverfahren sich in der Praxis gut bewährt hat, ist es wünschenswert, den Vorgang weiter zu automatisieren und zu optimieren. Vorstellbar ist hier eine automatische, evtl. adaptive Auswahl von Gitterlinien, die ein für die Deformation optimales fiktives Rahmentragwerk liefert.

Anhang A

Implementierungsdetails

A.1 Die Programmiersprache Ada95

Im wissenschaftlichen Rechnen ist die Programmiersprache Ada95 sehr wenig verbreitet [Sti98]. Obwohl die Sprache gegenüber weit verbreiteten Sprachen wie C++ oder FORTRAN^{9x} zahlreiche Vorteile aufweist, ist sie außerhalb ihrer traditionellen Nische kaum vertreten. Verglichen mit FORTRAN ist Ada eine junge Programmiersprache, die ihren Ursprung in einem Wettbewerb des Verteidigungsministeriums der USA Anfang der 70er Jahre hat. Dort suchte man nach einer neuen Programmiersprache, die allgemein anwendbar sein sollte, damit das vorhandene Durcheinander von Programmiersprachen nach und nach abgelöst werden könne. Kernanforderungen waren eine Modularisierung, so daß wiederverwendbare Komponenten einfach erstellt werden können, und eine weitreichende Unterstützung des Programmierers für die Entwicklung korrekter Programme [DoD78]. Dies führte dazu, daß die 1983 erstmals standardisierte Programmiersprache Ada überwiegend in militärischen Bereichen zum Einsatz kam. 1995 folgte die erste Überarbeitung des Standards, die vor allem die Möglichkeiten für objektorientierte Programmier-techniken wesentlich erweitert hat — die nun Ada95 genannte Sprache war die erste offiziell standardisierte objektorientierte Programmiersprache [TD97]. In der Folge hat sich die Verwendung der Sprache weiter verbreitet, auch wenn immer noch sicherheitsorientierte Anwendungen im Vordergrund stehen, wie z.B. Flugzeugsysteme, Eisenbahnsignalanlagen oder Banksysteme. Nachfolgend werden einige aus Sicht des Autors wesentliche Merkmale der Sprache kurz angerissen, für eine ausführliche Einführung und Übersicht seien z.B. [Bar98, Coh96] empfohlen.

Safety First!

Bei der Entwicklung von Ada stand die sichere Entwicklung und vor allem Weiterentwicklung bzw. Wartung von sicheren Programmen ganz weit oben im Lastenheft. Da für die Weiterentwicklung und Wartung von größeren Programmen die meiste Zeit benötigt wird, war eine hervorragende Lesbarkeit der Sprache ein wichtiges Kriterium. Ein weiteres wesentliches Merkmal ist

die sehr strenge Typisierung der Sprache. Für jeden Einsatzzweck können angemessene Datentypen erzeugt werden, deren korrekte Verwendung dann vom Compiler entsprechend überwacht wird. Im Vergleich zu FORTRAN fällt hierbei auf, daß ein globaler Datentyp bzw. eine globale Variable nur genau einmal definiert wird und nicht die Definition in jedem Programmteil wiederholt werden muß. All dies hilft bereits deutlich in der Entwicklung von Programmen, da viele typische Fehler der Programmierung in FORTRAN oder C automatisch vermieden werden.

Wesentliche Eigenschaften

Modularisierung

Ähnlich wie in N. Wirths Modula Programmiersprachen, die genauso wie Ada auf Pascal basieren, können zusammengehörige Funktionen und Datensätze in Modulen — in Ada *packages* genannt — zusammengefaßt werden. Neben der besseren Übersicht über die Struktur eines Programmes, bietet das Konzept den großen Vorteil, daß Namen von Unterprogrammen und Variablen nie global wie in FORTRAN oder C sind, sondern stets nur in dem jeweiligen Package lokal definiert werden. Dies vereinfacht die nachträgliche Erweiterung von Software oder auch das Erstellen wiederverwendbarer Module sehr, da im Gegensatz zu FORTRAN keine Rücksicht auf die bereits vorhandene Namensgebung genommen werden muß, was gerade bei großen Programmsystemen sehr mühsam ist und einer sinnvollen Benennung der Programmteile oft im Wege steht. In FORTRAN90 wurde dieses Konzept teilweise übernommen, da es sich in anderen Sprachen bereits bewährt hat.

Weiterhin ist es in Ada vorgeschrieben, eine von der Implementierung separate Spezifikationsdatei für Module zu erstellen, die alle nach außen hin sichtbaren Bestandteile des Moduls enthält. Die Implementierung, der sogenannte *body*, kann so jederzeit geändert werden. Solange die Spezifikation nicht berührt wird, hat dies keine Auswirkungen auf die Funktion des Programmes. Bei FORTRAN oder C Programmen kann man dagegen nicht vor Nebenwirkungen geschützt sein, da die Sprache es nicht verbietet, z.B. eine Variable aus dem Inneren einer Implementierung an einer beliebigen anderen Stelle zu verwenden.

Generisches Programmieren

Eine für die Erstellung von wissenschaftlicher Software (und die Erstellung von Software im allgemeinen) wesentliche Technik ist das generische Programmieren. Es kommt regelmäßig vor, daß ein bestimmter Algorithmus, z.B. ein Sortierverfahren, für verschiedene Datentypen genutzt werden soll. Die generische Programmierung ermöglicht es, diesen Algorithmus nur einmal, praktisch als Funktionsmuster zu erstellen und für jeden Datentyp, der zur Anwendung kommen soll, eine Kopie entsprechend automatisch zu erstellen. Prinzipiell ist diese Technik mit Hilfe eines Präprozessors auch bei Sprachen wie FORTRAN oder C möglich, jedoch nur mit Schwierigkeiten. Ada bietet hier

seit dem ersten Standard 1983 eine umfassende und mit dem streng typisierten Sprachmodell konsistente Lösung. Insgesamt ermöglicht die generische Programmierung eine günstige Art, vielseitig einsetzbare Funktions-Bibliotheken zu erstellen.

Als Beispiel sei hier eine kleine Routine genannt, die zwei Objekte eines beliebigen Datentyps vertauscht.

```
generic
  type My_Object is private; -- alle möglichen Datentypen
procedure Swap (Obj_1, Obj_2 : in out My_Object) is
  Tmp : My_Object;
begin
  Tmp := Obj_1;
  Obj_1 := Obj_2;
  Obj_2 := Tmp;
end Swap;
```

Um diese Routine nun nutzen zu können, z.B. um Real-Zahlen zu tauschen, muß eine entsprechende Instanz gebildet werden:

```
type Real is digits 12; -- mind. 12-stellige Real-Zahlen
procedure R_Swap is new Swap(Real);
```

Mit dem Aufruf von `R_Swap` können nun zwei Real-Zahlen vertauscht werden. Entsprechend kann das Unterprogramm auch für Integer-Zahlen oder Finite-Elemente verwendet werden, solange ein entsprechender Datentyp zum Ersatz von `My_Object` dafür definiert ist.

Neben einzelnen Unterprogrammen, wie hier dargestellt, können auch ganze Bibliotheken (in Form von Packages) von generischen Parametern abhängig gemacht werden.

Mixed Language Programming

Es war ein wesentliches Kriterium für die Auswahl von Ada95 für die Implementierung der Software dieser Arbeit, daß die Definitionen zum Import und Export von Daten und Prozeduren aus C und FORTRAN Programmen bereits im Standard verankert sind. Sogar die spezielle Speicherung von mehrdimensionalen Feldern in FORTRAN (spaltenweise) wird bei einer entsprechenden Deklaration automatisch berücksichtigt. Für C++ gilt dies nur eingeschränkt, da für C++ selbst noch keine verbindliche Regelung für die Speicherung von Datenstrukturen oder für die Schnittstelle zu Unterprogrammen definiert ist. Jedoch können C++ und Ada95 Programme immer über die gemeinsam bekannten Datentypen von C kommunizieren.

Late Binding (OOP), Parallelität, Portabilität

Auch wenn es für die Erstellung von numerischer Software eine weniger wichtige Eigenschaft ist, so erlaubt Ada95 eine objekt-orientierte Programmierung (OOP) im üblichen Sinne, d.h. entsprechend deklarierte Datentypen können

nachträglich erweitert und mit neuen Eigenschaften versehen werden. Außerdem können für die so entstandene Klasse von Datentypen Unterprogramme definiert werden, bei denen erst zur Laufzeit des Programmes, abhängig von der jeweiligen Unterart des Datensatzes, die Ausführung des zu dem Datensatz gehörigen Unterprogrammes angestoßen wird.

Ein weiterer wichtiger Bestandteil der Sprache Ada95 ist die Möglichkeit zur Erstellung parallel laufender Programme, ohne eine zusätzliche Bibliothek wie MPI oder fremde Sprachkonstrukte wie OpenMP zu verwenden. Ähnlich wie in High-Performance-FORTRAN allerdings ist diese Technik für die Ausführung auf Shared-Memory Computern geeigneter als das klassische Message-Passing typischer numerischer Software. Allerdings können mit dieser Technik leicht Client-Server-Anwendungen wie z.B. ein Banksystem mit externen Kundenterminals realisiert werden.

Aufgrund der bereits beim Entwurf der Sprache gewünschten Allgemeinheit ist die Sprache sehr portabel. Für die Anwendung im wissenschaftlichen Rechnen ist die bereits gegebene Existenz von Compilern für alle üblichen Workstation- und PC-Architekturen nützlich.

A.2 Finite Elemente

Bei der Implementierung des Finite Elemente Modells wurden die meisten zuvor genannten Konstrukte von Ada95 verwendet. Es wurde ein Modul `Generic_FE_System` erstellt, das die grundlegenden Datentypen und Eigenschaften des Finite-Elemente Modells beschreibt. Das Paket ist generisch von der Wahl eines numerischen Zahlentyps, eines Datentyps für 3D-Vektoren mit zugehörigen Matrizen (z.B. für Koordinaten) und eines Datentyps für algebraische Vektoren und Matrizen (z.B. für die Steifigkeitsmatrix) abhängig.

Das gesamte Finite-Elemente Modell ist mit den Techniken der OOP erstellt und besteht im wesentlichen aus 6 Datentypen (Objekttypen) und den zugeordneten Eigenschaften. Die Datentypen sind:

- Freiheitsgrade,
- Knoten,
- Elemente,
- Randbedingungen,
- Lasten,
- Materialien.

Dabei sind Elemente und Materialien sogenannte abstrakte Typen, von denen keine direkten Objekte angelegt werden können. Um z.B. Elemente anlegen zu können, muß zunächst ein spezieller Elementtyp, etwa ein Element für einen elastischen Balken, zusammen mit den zugehörigen Funktionen z.B. zur Definition der Elementmatrizen, definiert werden. Dies geschieht durch Definition eines entsprechend von dem abstrakten Grundtyp für Elemente

abgeleiteten Datentyps für das konkrete Element in einem *Child Package* zu `Generic_FE_System`.

Auf diese Weise ist in den Grundzügen ein Finite-Elemente-Programm entstanden, das sehr einfach um neue Elementtypen oder Materialien erweiterbar ist. Prinzipiell ist es auch denkbar, darin Elemente zur Simulation von Fluiden zu definieren.

Durch die Verwendung generischer Parameter für die zu verwendenden Vektoren kann auch hier das selbst geschriebene Vektor-Package durch andere ersetzt werden, um z.B. einen Vektor-Datentyp zu verwenden, der paralleles Rechnen unterstützt.

A.3 Hilfsbibliotheken

Für die Implementierung des Gesamtprogrammes für Gitterdeformation und Finite-Elemente wurde auf eine Reihe von Programmbibliotheken zurückgegriffen, um die Programmierung einfacher zu gestalten. Ein Teil dieser Routinen wurde selbst erstellt, ein anderer im Rahmen freier Software im Internet gefunden. Unter letzteren seien insbesondere die sog. Booch-Komponenten [Boo86, Wri02] zu nennen, die Listen und ähnliche Datenstrukturen implementieren. Desweiteren hat [Kaz00] mit seinen Routinen zum Editieren von Strings und dem Rechnen mit dimensionsbehafteten Zahlen zwei Bibliotheken erstellt, die besonders bei der Implementierung der Ein- und Ausgabe von großer Hilfe waren. So ist z.B. die Eingabedatei für die Diskretisierung des Flugkörpers durchgängig dimensioniert, um Eingabefehler durch Mißverständnisse bei der Dimension (mm vs. m) zu vermeiden.

A.3.1 Rechnen mit Vektoren und Matrizen

Bei der Erstellung numerischer Software gibt es viele Stellen, an denen lineare Algebra eine wichtige Rolle spielt. Daher gibt es für FORTRAN auch übliche Implementierungen wie BLAS und LAPACK, die viele Funktionen der Linearen Algebra unter Berücksichtigung der Eigenarten der Numerik und einer optimalen Rechengeschwindigkeit zur Verfügung stellen. Die Eigenschaften der Sprache Ada95 gestatten jedoch eine andere Form der Implementierung anstelle der üblichen Unterprogramme mit mehreren Eingabeparametern und einem Rückgabewert (z.B. `matmul` für Matrix-Vektor-Multiplikation). Ideal wäre eine Implementierung, die die Notation der Mathematik (+, -, *, /) übernimmt und die gleiche Geschwindigkeit wie die klassischen Unterprogramme bietet.

Die Verwendung von Funktionen anstelle von Prozeduren bedeutet, daß der zurückzugebende Wert als Datensatz erst in der Funktion angelegt wird und dann an eine übergeordnete Variable übergeben wird. Dies hat in der Regel deutlich mehr Speicheroperationen zur Folge als wenn ein bereits angelegter Ausgabedatensatz in einer Prozedur lediglich neu beschrieben wird. Benutzt man nun eine einfache Form der Implementierung für vektorielle Operationen in Funktionsform, so wird ein Programm in der Ausführung deutlich

langsamer werden, da vor der Ausführung jeder Operation ein Vektor für das Ergebnis im Speicher angelegt werden muß. Diese Einbuße an Geschwindigkeit wird üblicherweise mit unter dem Begriff „abstraction penalty“ gefaßt.

Verwendung einer Prozedur mit Rückgabe über einen der Parameter (in FORTRAN die notwendige Methode, da Funktionen keinen Vektor zurückliefern können.):

```
procedure Add_Vec(x : out Vector; a, b : in Vector) is
begin
  for i in a'Range loop
    x(i) := a(i) + b(i);
  end loop;
end Add_Vec;
```

Bei Ausführung der Zeile `Add_Vec(x, a, b)` geschieht folgendes:

1. Vor der Ausführung sind alle drei Vektoren `a`, `b` und `x` im Speicher angelegt.
2. Ausführung: Die Einträge von `a` und `b` werden addiert und in `x` geschrieben.

Bei einer Funktion wird es komplizierter:

```
function "+"(a, b : in Vector) return Vector is
  Result: Vector;
begin
  for i in a'Range loop
    Result(i) := a(i) + b(i);
  end loop;
  return Result;
end "+";
```

Bei Ausführung der Zeile `x := a + b` geschieht folgendes:

1. Vor der Ausführung sind die drei Vektoren `a`, `b` und `x` im Speicher angelegt.
2. Der Vektor `Result` wird im Speicher angelegt.
3. Ausführung: Die Einträge von `a` und `b` werden addiert und in `Result` geschrieben.
4. Die Werte aus `Result` werden in `x` kopiert.
5. der Vektor `Result` wird aus dem Speicher entfernt.

Gegenüber der oben dargestellten Form des Programms als Prozedur fällt vor allem die Erstellung, Kopie und abschließend Löschung des temporären Vektors `Result` auf. Diese Schritte wirken sich spürbar negativ auf die Rechenzeit aus und sollen daher vermieden werden.

In der vorliegenden Arbeit wird angestrebt, eine Bibliothek zu erstellen, die durch die Verwendung entsprechender Datenstrukturen und anderer Techniken diese Nachteile vermeidet, aber die Notation in Funktionsform beibehält.

Kernstück dafür ist die Verwaltung der Vektoren über Zeiger auf deren Speicherstelle, so daß die Kopie eines „Vektors“ zunächst nur die Kopie eines Zeigers beinhaltet. Natürlich kann so nicht vermieden werden, daß regelmäßig neue Vektoren erzeugt werden müssen, dafür wird ein weiteres Konstrukt eingeführt: Ein temporärer Vektor, in dem Zwischenergebnisse gespeichert werden und der bei der Rückgabe einfach gegen den Ergebnisvektor getauscht wird. So wird die Anlage neuer Vektoren in der Praxis auf ein minimal notwendiges Maß reduziert, lediglich der Speicherplatz für einen zusätzlichen Vektor muß in Kauf genommen werden.

Aufbau einer Funktion unter Verwendung von Zeigern auf die Vektoren und einem globalen, temporären Vektor:

```
Temp_Vec : Vector_Access
function "+"(a, b : in Vector_Access) return Vector_Access is
begin
  if Temp_Vec = Null then -- noch nicht angelegt?
    Temp_Vec := new Vector;
  end if;
  for i in a'Range loop
    Temp_Vec(i) := a(i) + b(i);
  end loop;
  return Temp_Vec;
end "+";
```

Bei Ausführung der Zeile $x := a + b$ geschieht folgendes:

1. Vor der Ausführung sind für die drei Zeiger a , b und x im Speicher Vektoren angelegt.
2. Wenn der Zeiger $Temp_Vec$ noch keinem Vector im Speicher zugewiesen ist, so wird einer neuer Vektor im Speicher angelegt und $Temp_Vec$ zugewiesen.
3. Ausführung: Die Einträge von a und b werden addiert und in $Temp_Vec$ geschrieben.
4. Die Zeiger x und $Temp_Vec$ werden einander zugewiesen, so daß x nun auch auf den Speicher von $Temp_Vec$ weist. Da der Speicherplatz, auf den x vorher gewiesen hat, nun unbenutzt ist, kann er für $Temp_Vec$ benutzt werden.

Man kann sehen, daß wenn $Temp_Vec$ schon ein Vektor zugewiesen ist, nur noch die Addition und das Vertauschen zweier Zeiger auf Speicherinhalte ausgeführt werden müssen. Das teure Kopieren eines Vektorinhaltes in einen anderen entfällt.

Um die Verkettung mehrerer Operationen weiter zu optimieren, wird erlaubt, daß ein Datenfeld eines Vektors von mehreren Vektor-Zeigern gleichzeitig benutzt wird. So kann z.B. für mehrere aufeinander folgende Additionen der gleiche Vektor für die Zwischenergebnisse genutzt werden, ohne für jedes Zwischenergebnis einen anderen anlegen zu müssen. Damit bei Multiplikationen keine Zwischenergebnisse zerstört werden, wird die Zahl der Zeiger, die auf einen Datensatz weisen, gezählt (sog. „Reference-Counting“), so daß der Temporärvektor hier nur dann für die Ablage eines Zwischenergebnisses benutzt wird, wenn der Zähler auf 1 steht und somit keine weitere Operation auf diesen Datensatz Bezug nimmt. Sollte der Zähler größer als 1 sein, so wird ein neues temporäres Objekt für das entsprechende Zwischenergebnis angelegt.

Soll $x := a + b + c$ gerechnet werden, so wird der temporäre Vektor `Temp_Vec` mehrfach verwendet, was eine weitere Optimierung darstellt:

1. Lege ggf. einen neuen `Temp_Vec` an.
2. Berechne `Temp_Vec = b + c`. `Temp_Vec` ist dann Eingabeparameter für die zweite Addition.
3. Berechne `Temp_Vec = a + Temp_Vec`.
4. Vertausche die Zeiger zu `x` und `Temp_Vec`.

Bei gemischter Multiplikation und Addition, z.B. $x := r*a + s*b$ mit Skalaren r und s , darf der Temporärvektor natürlich nicht mehrfach verwendet werden, da sonst Zwischenergebnisse zerstört werden. Bei jeder Multiplikation wird also ein neuer Temporärvektor als Speicher für ein Zwischenergebnis angelegt.

A.3.2 Kontrolle des Rundungsfehlers

Für die Kontrolle des Rundungsfehlers gibt es verschiedene Ansätze. Neben der klassischen Vorwärts- und Rückwärts-Analyse der Algorithmen gibt es auch numerische Methoden, die eine gute Abschätzung des Rundungsfehlers während der Berechnung erlauben. Zum einen bietet sich das Rechnen mit Intervallen an, zum anderen eine statistische Methode zur Erfassung der Präzision [CCF96, CGV96], die als „Cestac“-Methode bekannt ist. Da die Analyse eines gesamten Programmes sehr umfangreich und aufwendig ist, liefert der Einsatz einer numerischen Methode hier eine erhebliche Erleichterung, zumal evtl. kritische Stellen, wie z.B. eine instabile Abfrage oder Division, hier leicht festgestellt werden können. Zudem können bei derartigen Methoden auch Datenfehler, z.B. die typische Genauigkeit von Meßdaten, berücksichtigt werden.

Für die Anwendung in der gekoppelten aeroelastischen Simulation wurde die Cestac Methode implementiert, da diese gegenüber der Intervallrechnung bei der Implementierung Vorteile hat, wenn man über die Grundrechenarten hinaus geht. Die Wirkungsweise wird nachfolgend an einem kleinen Beispiel verdeutlicht.

Zu berechnen ist eine Rekursionsfolge [Mul87]

$$u(n+1) = 111 - 1130/u(n) + 3000/(u(n) * u(n-1))$$

mit $u(1) = 5.5$ und $u(2) = 61/11$. Der exakte Grenzwert ist $\lim_{n \rightarrow \infty} u(n) = 6$. Der zugehörige Ada95-Programmausschnitt ist wie folgt:

```

declare
  A : Float := 5.5;
  B : Float := 61.0/11.0;
  C : Float;
begin
  for I in 3..25 loop
    C := B;
    B := 111.0 - 1130.0/B + 3000.0/(A*B);
    A := C;
    Put_Line ("U(" & I'Img & ") = " & B'Img);
  end loop;
end;
```

Wird für Float die maximale Rechengenauigkeit einer 32-bit Workstation (HP 9000) verwendet, so lautet das Ergebnis:

```

U( 3) = 5.5901639344262E+00
U( 4) = 5.6334310850440E+00
U( 5) = 5.6746486205100E+00
U( 6) = 5.7133290523783E+00
U( 7) = 5.7491209196646E+00
U( 8) = 5.7818109198243E+00
U( 9) = 5.8113142268599E+00
U(10) = 5.8376563522579E+00
U(11) = 5.8609481538324E+00
U(12) = 5.8813197515411E+00
U(13) = 5.8981770256150E+00
U(14) = 5.8979652475565E+00
U(15) = 5.6470110840386E+00
U(16) = 9.6833994452975E-01
U(17) = -5.0732160516247E+02
U(18) = 1.0712063523281E+02
U(19) = 1.0039594218944E+02
U(20) = 1.0002351860606E+02
U(21) = 1.0000140357456E+02
U(22) = 1.0000008385280E+02
U(23) = 1.0000000501314E+02
U(24) = 1.0000000029989E+02
U(25) = 1.0000000001795E+02
```

Wie man leicht sehen kann, wendet sich die Folge nach der 15. Iteration von dem richtigen Grenzwert ab, um dann später scheinbar gegen einen Wert bei 100 zu konvergieren. Wendet man die implementierte Cestac-Methode an, so erhält man:

```
U( 3) = 5.5901639344262E+00
U( 4) = 5.63343108504E+00
U( 5) = 5.6746486205E+00
U( 6) = 5.71332905E+00
U( 7) = 5.749120E+00
U( 8) = 5.78181E+00
U( 9) = 5.811E+00
U( 10) = 5.83E+00
U( 11) = 5.E+00
U( 12) = @.0
U( 13) = @.0
Fehler: INSTABILE DIVISION
```

Eine instabile Division tritt dann auf, wenn der Nenner keine signifikante Zahl mehr ist. Dann kann die errechnete Folge die „Richtung“ ändern, wie oben durch die Konvergenz zu einem anderen Grenzwert hin erkennbar ist. In der Darstellung des Programmablaufes mit der Cestac-Methode oben ist erkennbar, wie die Zahl der verlässlich bestimmten Ziffern des Ergebnisses abnimmt. Mit @.0 wird eine Zahl ohne signifikanten Wert bezeichnet, die mathematisch in der Wirkung einer Null ähnlich ist [CGV96].

Prinzipiell kann natürlich die gesamte aeroelastische Simulation mit Einsatz dieser Methode durchgeführt werden. Jedoch ist dies aufgrund der sehr viel längeren Rechenzeit und des höheren Speicherbedarfes nicht praktikabel, wenn nicht nach einer speziellen Fehlerursache gesucht werden soll. Die Anwendung an speziellen Stellen, die besonders kritisch sind, kann jedoch sinnvoll sein.

Die Bibliothek wird bei der Datenübertragung zwischen der Bewegung des Laborsystems für den gesamten Flugkörper und der elastischen Bewegung eingesetzt, um eventuelle Instabilitäten aufgrund der dort eingesetzten Kreuzprodukte feststellen zu können. Der Einsatz der entsprechenden Bibliothek innerhalb des Gesamtprogramms ist noch nicht ausgereizt. Denkbar ist z.B. der Einsatz innerhalb der iterativen Löser zur Bestimmung eines guten Abbruchkriteriums oder auch bei der Übergabe von Belastungswerten vom Strömungslöser an den Festkörperanteil des Programms mit einer entsprechend angegebenen Genauigkeit.

Anhang B

FE-Matrizen für erweitertes Balkenelement

Für das in Kapitel 3.2 erstellte Finite Balkenelement werden im Folgenden die vollständigen Elementsteifigkeits- und Elementmassenmatrix vorgestellt. Der besseren Übersicht halber werden beide Matrizen in den Anteil der reinen Balkenbiegung (entsprechend dem Rayleigh-Balken, die eine Drehträgheit der Balkenquerschnitte bereits berücksichtigt) und den zusätzlichen Anteil der im Timoshenko-Modell berücksichtigten Scherung aufgeteilt.

Der zugrunde liegende Ansatz von Friedman und Kosmatka [FK93] bietet diese Möglichkeit, indem das Verhältnis der Biege- zur Schersteifigkeit (ϕ_2 und ϕ_3) zu Null gesetzt wird, einer unendlich hohen Schersteifigkeit entsprechend. Zusätzlich werden für den Rayleigh-Anteil alle Terme mit dem Scherkoeffizienten k vernachlässigt.

In der Steifigkeitsmatrix gibt es zwischen dem Bernoulli-Balken (ohne Berücksichtigung der Drehträgheit der Balkenquerschnitte) und dem Rayleigh-Modell keinen Unterschied. Die Massenmatrix des Bernoulli-Balkens kann jedoch aus der des Rayleigh-Balkens abgeleitet werden, indem alle darin auftretenden Flächenträgheitsmomente zu Null gesetzt werden.

B.1 Elementsteifigkeitsmatrix

Der Rayleigh-Anteil der Elementsteifigkeitsmatrix ist

$$\mathbf{K}_R = \frac{EA^{(E)}}{L} \begin{bmatrix} \mathbf{K}_{R1} & \mathbf{K}_{R12} \\ \mathbf{K}_{R12}^T & \mathbf{K}_{R2} \end{bmatrix} \quad (\text{B.1})$$

$$\mathbf{K}_{R1} = \begin{bmatrix} 1 & 0 & 0 & 0 & \xi_{B3} & -\xi_{B2} \\ 0 & 12 \frac{I'_{B33}}{L^2} & -12 \frac{I'_{B23}}{L^2} & 0 & 6 \frac{I'_{B23}}{L} & 6 \frac{I'_{B33}}{L} \\ 0 & -12 \frac{I'_{B23}}{L^2} & 12 \frac{I'_{B22}}{L^2} & 0 & -6 \frac{I'_{B22}}{L} & -6 \frac{I'_{B23}}{L} \\ 0 & 0 & 0 & G'I'_T & 0 & 0 \\ \xi_{B3} & 6 \frac{I'_{B23}}{L} & -6 \frac{I'_{B22}}{L} & 0 & 4I'_{B22} & 4I'_{B23} \\ \xi_{B2} & 6 \frac{I'_{B33}}{L} & -6 \frac{I'_{B23}}{L} & 0 & 4I'_{B23} & 4I'_{B33} \end{bmatrix} \quad (\text{B.2})$$

$$\mathbf{K}_{R12} = \begin{bmatrix} -1 & 0 & 0 & 0 & -\xi_{B3} & \xi_{B2} \\ 0 & -12 \frac{I'_{B33}}{L^2} & 12 \frac{I'_{B23}}{L^2} & 0 & 6 \frac{I'_{B23}}{L} & 6 \frac{I'_{B33}}{L} \\ 0 & 12 \frac{I'_{B23}}{L^2} & -12 \frac{I'_{B22}}{L^2} & 0 & -6 \frac{I'_{B22}}{L} & -6 \frac{I'_{B23}}{L} \\ 0 & 0 & 0 & -G'I'_T & 0 & 0 \\ -\xi_{B3} & -6 \frac{I'_{B23}}{L} & 6 \frac{I'_{B22}}{L} & 0 & 2I'_{B22} & 2I'_{B23} \\ \xi_{B2} & -6 \frac{I'_{B33}}{L} & 6 \frac{I'_{B23}}{L} & 0 & 2I'_{B23} & 2I'_{B33} \end{bmatrix} \quad (\text{B.3})$$

$$\mathbf{K}_{R2} = \begin{bmatrix} 1 & 0 & 0 & 0 & \xi_{B3} & -\xi_{B2} \\ 0 & 12 \frac{I'_{B33}}{L^2} & -12 \frac{I'_{B23}}{L^2} & 0 & -6 \frac{I'_{B23}}{L} & -6 \frac{I'_{B33}}{L} \\ 0 & -12 \frac{I'_{B23}}{L^2} & 12 \frac{I'_{B22}}{L^2} & 0 & 6 \frac{I'_{B22}}{L} & 6 \frac{I'_{B23}}{L} \\ 0 & 0 & 0 & G'I'_T & 0 & 0 \\ \xi_{B3} & -6 \frac{I'_{B23}}{L} & 6 \frac{I'_{B22}}{L} & 0 & 4I'_{B22} & 4I'_{B23} \\ -\xi_{B2} & -6 \frac{I'_{B33}}{L} & 6 \frac{I'_{B23}}{L} & 0 & 4I'_{B23} & 4I'_{B33} \end{bmatrix} \quad (\text{B.4})$$

dabei gilt

$$\mathbf{I}'_B = \begin{bmatrix} I_{B22}^{(E)}/A^{(E)} + \xi_{B3}^2 & I_{B23}^{(E)}/A^{(E)} - \xi_{B2}\xi_{B3} \\ \text{symm.} & I_{B33}^{(E)}/A^{(E)} + \xi_{B2}^2 \end{bmatrix}$$

und $I'_T = I_T/A^{(E)}$ sowie $G' = G/E$. Die Matrix \mathbf{K}_R und die Teilmatrizen \mathbf{K}_{R1} , \mathbf{K}_{R2} sind symmetrisch. Die Matrix \mathbf{K}_R wird zur Elementsteifigkeitsmatrix des vollständigen Timoshenko-Balkens erweitert, indem

$$\mathbf{K}_T = \mathbf{K}_R + \frac{EA^{(E)}}{L} \begin{bmatrix} \mathbf{K}_{T1} & \mathbf{K}_{T12} \\ \mathbf{K}_{T12}^T & \mathbf{K}_{T2} \end{bmatrix} \quad (\text{B.5})$$

gebildet wird. Die resultierende Matrix \mathbf{K}_T ist ebenfalls symmetrisch.

$\mathbf{K}_{T1} =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\phi_2^2 G' k_{22}}{(1+\phi_2)^2} & \frac{\phi_2 G' k_{23} \phi_3}{(1+\phi_2)(1+\phi_3)} & -\frac{G' \phi_2 (k_{22} \xi_{D3} - k_{23} \xi_{D2})}{1+\phi_2} & -\frac{1}{2} \frac{\phi_2 G' k_{23} \phi_3 L}{(1+\phi_2)(1+\phi_3)} & \frac{1}{2} \frac{\phi_2^2 G' k_{22} L}{(1+\phi_2)^2} \\ 0 & \frac{\phi_2 G' k_{23} \phi_3}{(1+\phi_2)(1+\phi_3)} & \frac{\phi_3^2 G' k_{33}}{(1+\phi_3)^2} & -\frac{G' \phi_3 (k_{23} \xi_{D3} - k_{33} \xi_{D2})}{1+\phi_3} & -\frac{1}{2} \frac{\phi_3^2 G' k_{33} L}{(1+\phi_3)^2} & \frac{1}{2} \frac{\phi_2 G' k_{23} \phi_3 L}{(1+\phi_2)(1+\phi_3)} \\ 0 & -\frac{G' \phi_2 (k_{22} \xi_{D3} - k_{23} \xi_{D2})}{1+\phi_2} & -\frac{G' \phi_3 (k_{23} \xi_{D3} - k_{33} \xi_{D2})}{1+\phi_3} & G' (\xi_{D3}^2 k_{22} - 2 \xi_{D3} k_{23} \xi_{D2} + \xi_{D2}^2 k_{33}) & \frac{1}{2} \frac{G' \phi_3 (k_{23} \xi_{D3} - k_{33} \xi_{D2}) L}{1+\phi_3} & -\frac{1}{2} \frac{G' \phi_2 (k_{22} \xi_{D3} - k_{23} \xi_{D2}) L}{1+\phi_2} \\ 0 & -\frac{1}{2} \frac{\phi_2 G' k_{23} \phi_3 L}{(1+\phi_2)(1+\phi_3)} & -\frac{1}{2} \frac{\phi_3^2 G' k_{33} L}{(1+\phi_3)^2} & \frac{1}{2} \frac{G' \phi_3 (k_{23} \xi_{D3} - k_{33} \xi_{D2}) L}{1+\phi_3} & \frac{1}{4} \frac{\phi_3^2 G' k_{33} L^2}{(1+\phi_3)^2} & -\frac{1}{4} \frac{\phi_2 G' k_{23} \phi_3 L^2}{(1+\phi_2)(1+\phi_3)} \\ 0 & \frac{1}{2} \frac{\phi_2^2 G' k_{22} L}{(1+\phi_2)^2} & \frac{1}{2} \frac{\phi_2 G' k_{23} \phi_3 L}{(1+\phi_2)(1+\phi_3)} & -\frac{1}{2} \frac{G' \phi_2 (k_{22} \xi_{D3} - k_{23} \xi_{D2}) L}{1+\phi_2} & -\frac{1}{4} \frac{\phi_2 G' k_{23} \phi_3 L^2}{(1+\phi_2)(1+\phi_3)} & \frac{1}{4} \frac{\phi_2^2 G' k_{22} L^2}{(1+\phi_2)^2} \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -12 \frac{\phi_2 (2+\phi_2) I'_{B33}}{(1+\phi_2)^2 L^2} & 12 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L^2 (1+\phi_3)} & 0 & -6 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L (1+\phi_3)} & -6 \frac{\phi_2 (2+\phi_2) I'_{B33}}{(1+\phi_2)^2 L} \\ 0 & 12 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L^2 (1+\phi_3)} & -12 \frac{\phi_3 (2+\phi_3) I'_{B22}}{(1+\phi_3)^2 L^2} & 0 & 6 \frac{\phi_3 (2+\phi_3) I'_{B22}}{(1+\phi_3)^2 L} & 6 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L (1+\phi_3)} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -6 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L (1+\phi_3)} & 6 \frac{\phi_3 (2+\phi_3) I'_{B22}}{(1+\phi_3)^2 L} & 0 & -3 \frac{\phi_3 (2+\phi_3) I'_{B22}}{(1+\phi_3)^2} & -3 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) (1+\phi_3)} \\ 0 & -6 \frac{\phi_2 (2+\phi_2) I'_{B33}}{(1+\phi_2)^2 L} & 6 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L (1+\phi_3)} & 0 & -3 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) (1+\phi_3)} & -3 \frac{\phi_2 (2+\phi_2) I'_{B33}}{(1+\phi_2)^2} \end{bmatrix} \quad (\text{B.6})$$

$$\mathbf{K}_{T12} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\phi_2^2 G' k_{22}}{(1+\phi_2)^2} & -\frac{\phi_2 G' k_{23} \phi_3}{(1+\phi_2)(1+\phi_3)} & \frac{G' \phi_2 (k_{22} \xi_{D3} - k_{23} \xi_{D2})}{1+\phi_2} & -\frac{1}{2} \frac{\phi_2 G' k_{23} \phi_3 L}{(1+\phi_2)(1+\phi_3)} & \frac{1}{2} \frac{\phi_2^2 G' k_{22} L}{(1+\phi_2)^2} \\ 0 & -\frac{\phi_2 G' k_{23} \phi_3}{(1+\phi_2)(1+\phi_3)} & -\frac{\phi_3^2 G' k_{33}}{(1+\phi_3)^2} & \frac{G' \phi_3 (k_{23} \xi_{D3} - k_{33} \xi_{D2})}{1+\phi_3} & -\frac{1}{2} \frac{\phi_3^2 G' k_{33} L}{(1+\phi_3)^2} & \frac{1}{2} \frac{\phi_2 G' k_{23} \phi_3 L}{(1+\phi_2)(1+\phi_3)} \\ 0 & \frac{G' \phi_2 (k_{22} \xi_{D3} - k_{23} \xi_{D2})}{1+\phi_2} & \frac{G' \phi_3 (k_{23} \xi_{D3} - k_{33} \xi_{D2})}{1+\phi_3} & -G' (\xi_{D3}^2 k_{22} - 2 \xi_{D3} k_{23} \xi_{D2} + \xi_{D2}^2 k_{33}) & \frac{1}{2} \frac{G' \phi_3 (k_{23} \xi_{D3} - k_{33} \xi_{D2}) L}{1+\phi_3} & -\frac{1}{2} \frac{G' \phi_2 (k_{22} \xi_{D3} - k_{23} \xi_{D2}) L}{1+\phi_2} \\ 0 & \frac{1}{2} \frac{\phi_2 G' k_{23} \phi_3 L}{(1+\phi_2)(1+\phi_3)} & \frac{1}{2} \frac{\phi_3^2 G' k_{33} L}{(1+\phi_3)^2} & -\frac{1}{2} \frac{G' \phi_3 (k_{23} \xi_{D3} - k_{33} \xi_{D2}) L}{1+\phi_3} & \frac{1}{4} \frac{\phi_3^2 G' k_{33} L^2}{(1+\phi_3)^2} & -\frac{1}{4} \frac{\phi_2 G' k_{23} \phi_3 L^2}{(1+\phi_2)(1+\phi_3)} \\ 0 & -\frac{1}{2} \frac{\phi_2^2 G' k_{22} L}{(1+\phi_2)^2} & -\frac{1}{2} \frac{\phi_2 G' k_{23} \phi_3 L}{(1+\phi_2)(1+\phi_3)} & \frac{1}{2} \frac{G' \phi_2 (k_{22} \xi_{D3} - k_{23} \xi_{D2}) L}{1+\phi_2} & -\frac{1}{4} \frac{\phi_2 G' k_{23} \phi_3 L^2}{(1+\phi_2)(1+\phi_3)} & \frac{1}{4} \frac{\phi_2^2 G' k_{22} L^2}{(1+\phi_2)^2} \end{bmatrix} \\
 + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 12 \frac{\phi_2 (2+\phi_2) I'_{B33}}{(1+\phi_2)^2 L^2} & -12 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L^2 (1+\phi_3)} & 0 & -6 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L (1+\phi_3)} & -6 \frac{\phi_2 (2+\phi_2) I'_{B33}}{(1+\phi_2)^2 L} \\ 0 & -12 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L^2 (1+\phi_3)} & 12 \frac{\phi_3 (2+\phi_3) I'_{B22}}{(1+\phi_3)^2 L^2} & 0 & 6 \frac{\phi_3 (2+\phi_3) I'_{B22}}{(1+\phi_3)^2 L} & 6 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L (1+\phi_3)} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L (1+\phi_3)} & -6 \frac{\phi_3 (2+\phi_3) I'_{B22}}{(1+\phi_3)^2 L} & 0 & -3 \frac{\phi_3 (2+\phi_3) I'_{B22}}{(1+\phi_3)^2} & -3 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) (1+\phi_3)} \\ 0 & 6 \frac{\phi_2 (2+\phi_2) I'_{B33}}{(1+\phi_2)^2 L} & -6 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L (1+\phi_3)} & 0 & -3 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) (1+\phi_3)} & -3 \frac{\phi_2 (2+\phi_2) I'_{B33}}{(1+\phi_2)^2} \end{bmatrix} \quad (B.7)$$

$\mathbf{K}_{T2} =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\phi_2^2 G' k_{22}}{(1+\phi_2)^2} & \frac{\phi_2 G' k_{23} \phi_3}{(1+\phi_2)(1+\phi_3)} & -\frac{G' \phi_2 (k_{22} \xi_{D3} - k_{23} \xi_{D2})}{1+\phi_2} & \frac{1}{2} \frac{\phi_2 G' k_{23} \phi_3 L}{(1+\phi_2)(1+\phi_3)} & -\frac{1}{2} \frac{\phi_2^2 G' k_{22} L}{(1+\phi_2)^2} \\ 0 & \frac{\phi_2 G' k_{23} \phi_3}{(1+\phi_2)(1+\phi_3)} & \frac{\phi_3^2 G' k_{33}}{(1+\phi_3)^2} & -\frac{G' \phi_3 (k_{23} \xi_{D3} - k_{33} \xi_{D2})}{1+\phi_3} & \frac{1}{2} \frac{\phi_3^2 G' k_{33} L}{(1+\phi_3)^2} & -\frac{1}{2} \frac{\phi_2 G' k_{23} \phi_3 L}{(1+\phi_2)(1+\phi_3)} \\ 0 & -\frac{G' \phi_2 (k_{22} \xi_{D3} - k_{23} \xi_{D2})}{1+\phi_2} & -\frac{G' \phi_3 (k_{23} \xi_{D3} - k_{33} \xi_{D2})}{1+\phi_3} & G' (\xi_{D3}^2 k_{22} - 2 \xi_{D3} k_{23} \xi_{D2} + \xi_{D2}^2 k_{33}) & -\frac{1}{2} \frac{G' \phi_3 (k_{23} \xi_{D3} - k_{33} \xi_{D2}) L}{1+\phi_3} & \frac{1}{2} \frac{G' \phi_2 (k_{22} \xi_{D3} - k_{23} \xi_{D2}) L}{1+\phi_2} \\ 0 & \frac{1}{2} \frac{\phi_2 G' k_{23} \phi_3 L}{(1+\phi_2)(1+\phi_3)} & \frac{1}{2} \frac{\phi_3^2 G' k_{33} L}{(1+\phi_3)^2} & -\frac{1}{2} \frac{G' \phi_3 (k_{23} \xi_{D3} - k_{33} \xi_{D2}) L}{1+\phi_3} & \frac{1}{4} \frac{\phi_3^2 G' k_{33} L^2}{(1+\phi_3)^2} & -\frac{1}{4} \frac{\phi_2 G' k_{23} \phi_3 L^2}{(1+\phi_2)(1+\phi_3)} \\ 0 & -\frac{1}{2} \frac{\phi_2^2 G' k_{22} L}{(1+\phi_2)^2} & -\frac{1}{2} \frac{\phi_2 G' k_{23} \phi_3 L}{(1+\phi_2)(1+\phi_3)} & \frac{1}{2} \frac{G' \phi_2 (k_{22} \xi_{D3} - k_{23} \xi_{D2}) L}{1+\phi_2} & -\frac{1}{4} \frac{\phi_2 G' k_{23} \phi_3 L^2}{(1+\phi_2)(1+\phi_3)} & \frac{1}{4} \frac{\phi_2^2 G' k_{22} L^2}{(1+\phi_2)^2} \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -12 \frac{\phi_2 (2+\phi_2) I'_{B33}}{(1+\phi_2)^2 L^2} & 12 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L^2 (1+\phi_3)} & 0 & 6 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L (1+\phi_3)} & 6 \frac{\phi_2 (2+\phi_2) I'_{B33}}{(1+\phi_2)^2 L} \\ 0 & 12 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L^2 (1+\phi_3)} & -12 \frac{\phi_3 (2+\phi_3) I'_{B22}}{(1+\phi_3)^2 L^2} & 0 & -6 \frac{\phi_3 (2+\phi_3) I'_{B22}}{(1+\phi_3)^2 L} & -6 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L (1+\phi_3)} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L (1+\phi_3)} & -6 \frac{\phi_3 (2+\phi_3) I'_{B22}}{(1+\phi_3)^2 L} & 0 & -3 \frac{\phi_3 (2+\phi_3) I'_{B22}}{(1+\phi_3)^2} & -3 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) (1+\phi_3)} \\ 0 & 6 \frac{\phi_2 (2+\phi_2) I'_{B33}}{(1+\phi_2)^2 L} & -6 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) L (1+\phi_3)} & 0 & -3 \frac{(\phi_3 + \phi_2 + \phi_3 \phi_2) I'_{B23}}{(1+\phi_2) (1+\phi_3)} & -3 \frac{\phi_2 (2+\phi_2) I'_{B33}}{(1+\phi_2)^2} \end{bmatrix} \quad (\text{B.8})$$

B.2 Elementmassenmatrix

Der Rayleigh-Anteil der Elementmassenmatrix ist

$$\mathbf{M}_R = \rho A^{(\rho)} \begin{bmatrix} \mathbf{M}_{R1} & \mathbf{M}_{R12} \\ \mathbf{M}_{R12}^T & \mathbf{M}_{R2} \end{bmatrix} \quad (\text{B.9})$$

$$\mathbf{M}_{R1} = \begin{bmatrix} \frac{1}{3}L & \frac{1}{2}\xi_{C2} & \frac{1}{2}\xi_{C3} & 0 & \frac{1}{12}\xi_{C3}L & -\frac{1}{12}\xi_{C2}L \\ \frac{1}{2}\xi_{C2} & \frac{1}{35}\frac{13L^2+42I'_{C33}}{L} & -\frac{6}{5}\frac{I'_{C23}}{L} & -\frac{7}{20}\xi_{C3}L & \frac{1}{10}I'_{C23} & \frac{11}{210}L^2 + \frac{1}{10}I'_{C33} \\ \frac{1}{2}\xi_{C3} & -\frac{6}{5}\frac{I'_{C23}}{L} & \frac{1}{35}\frac{13L^2+42I'_{C22}}{L} & \frac{7}{20}\xi_{C2}L & -\frac{11}{210}L^2 - \frac{1}{10}I'_{C22} & -\frac{1}{10}I'_{C23} \\ 0 & -\frac{7}{20}\xi_{C3}L & \frac{7}{20}\xi_{C2}L & \frac{1}{3}I'_{C11}L & -\frac{1}{20}L^2\xi_{C2} & -\frac{1}{20}L^2\xi_{C3} \\ \frac{1}{12}\xi_{C3}L & \frac{1}{10}I'_{C23} & -\frac{11}{210}L^2 - \frac{1}{10}I'_{C22} & -\frac{1}{20}L^2\xi_{C2} & \frac{1}{105}L(14I'_{C22} + L^2) & \frac{2}{15}LI'_{C23} \\ -\frac{1}{12}\xi_{C2}L & \frac{11}{210}L^2 + \frac{1}{10}I'_{C33} & -\frac{1}{10}I'_{C23} & -\frac{1}{20}L^2\xi_{C3} & \frac{2}{15}LI'_{C23} & \frac{1}{105}L(14I'_{C33} + L^2) \end{bmatrix} \quad (\text{B.10})$$

$$\mathbf{M}_{R12} = \begin{bmatrix} \frac{1}{6}L & -\frac{1}{2}\xi_{C2} & -\frac{1}{2}\xi_{C3} & 0 & -\frac{1}{12}\xi_{C3}L & \frac{1}{12}\xi_{C2}L \\ \frac{1}{2}\xi_{C2} & \frac{3}{70}\frac{3L^2-28I'_{C33}}{L} & \frac{6}{5}\frac{I'_{C23}}{L} & -\frac{3}{20}\xi_{C3}L & \frac{1}{10}I'_{C23} & -\frac{13}{420}L^2 + \frac{1}{10}I'_{C33} \\ \frac{1}{2}\xi_{C3} & \frac{6}{5}\frac{I'_{C23}}{L} & \frac{3}{70}\frac{3L^2-28I'_{C22}}{L} & \frac{3}{20}\xi_{C2}L & \frac{13}{420}L^2 - \frac{1}{10}I'_{C22} & -\frac{1}{10}I'_{C23} \\ 0 & -\frac{3}{20}\xi_{C3}L & \frac{3}{20}\xi_{C2}L & \frac{1}{6}I'_{C11}L & \frac{1}{30}L^2\xi_{C2} & \frac{1}{30}L^2\xi_{C3} \\ -\frac{1}{12}\xi_{C3}L & -\frac{1}{10}I'_{C23} & -\frac{13}{420}L^2 + \frac{1}{10}I'_{C22} & -\frac{1}{30}L^2\xi_{C2} & -\frac{1}{420}L(14I'_{C22} + 3L^2) & -\frac{1}{30}LI'_{C23} \\ \frac{1}{12}\xi_{C2}L & \frac{13}{420}L^2 - \frac{1}{10}I'_{C33} & \frac{1}{10}I'_{C23} & -\frac{1}{30}L^2\xi_{C3} & -\frac{1}{30}LI'_{C23} & -\frac{1}{420}L(14I'_{C33} + 3L^2) \end{bmatrix} \quad (\text{B.11})$$

$$\mathbf{M}_{R2} = \begin{bmatrix} \frac{1}{3}L & -\frac{1}{2}\xi_{C2} & -\frac{1}{2}\xi_{C3} & 0 & \frac{1}{12}\xi_{C3}L & -\frac{1}{12}\xi_{C2}L \\ -\frac{1}{2}\xi_{C2} & \frac{1}{35}\frac{13L^2+42I'_{C33}}{L} & -\frac{6}{5}\frac{I'_{C23}}{L} & -\frac{7}{20}\xi_{C3}L & -\frac{1}{10}I'_{C23} & -\frac{11}{210}L^2 - \frac{1}{10}I'_{C33} \\ -\frac{1}{2}\xi_{C3} & -\frac{6}{5}\frac{I'_{C23}}{L} & \frac{1}{35}\frac{13L^2+42I'_{C22}}{L} & \frac{7}{20}\xi_{C2}L & \frac{11}{210}L^2 + \frac{1}{10}I'_{C22} & \frac{1}{10}I'_{C23} \\ 0 & -\frac{7}{20}\xi_{C3}L & \frac{7}{20}\xi_{C2}L & \frac{1}{3}I'_{C11}L & \frac{1}{20}L^2\xi_{C2} & \frac{1}{20}L^2\xi_{C3} \\ \frac{1}{12}\xi_{C3}L & -\frac{1}{10}I'_{C23} & \frac{11}{210}L^2 + \frac{1}{10}I'_{C22} & \frac{1}{20}L^2\xi_{C2} & \frac{1}{105}L(14I'_{C22} + L^2) & \frac{2}{15}LI'_{C23} \\ -\frac{1}{12}\xi_{C2}L & -\frac{11}{210}L^2 - \frac{1}{10}I'_{C33} & \frac{1}{10}I'_{C23} & \frac{1}{20}L^2\xi_{C3} & \frac{2}{15}LI'_{C23} & \frac{1}{105}L(14I'_{C33} + L^2) \end{bmatrix} \quad (\text{B.12})$$

mit

$$\mathbf{I}'_C = \begin{bmatrix} I'_{C11}/A^{(\rho)} + (\xi_{C2}^2 + \xi_{C3}^2) & 0 & 0 \\ & I'_{C22}/A^{(\rho)} + \xi_{C3}^2 & I'_{C23}/A^{(\rho)} - \xi_{C2}\xi_{C3} \\ & \text{symm.} & I'_{C33}/A^{(\rho)} + \xi_{C2}^2 \end{bmatrix}.$$

Die Matrix \mathbf{M}_R wird zur Elementmassenmatrix des vollständigen Timoshenko-Balkens erweitert, indem

$$\mathbf{M}_T = \rho A^{(\rho)} \begin{bmatrix} \mathbf{M}_{T1} & \mathbf{M}_{T12} \\ \mathbf{M}_{T12}^T & \mathbf{M}_{T2} \end{bmatrix} \quad (\text{B.13})$$

gebildet wird. Auch hier sind \mathbf{M}_R und \mathbf{M}_T symmetrisch.

$$\mathbf{M}_{T1} = \begin{bmatrix} 0 & -\frac{1}{2} \frac{\xi_{C2} \phi_2}{1+\phi_2} & -\frac{1}{2} \frac{\xi_{C3} \phi_3}{1+\phi_3} & 0 & \frac{1}{4} \frac{\xi_{C3} L \phi_3}{1+\phi_3} & -\frac{1}{4} \frac{\xi_{C2} L \phi_2}{1+\phi_2} \\ -\frac{1}{2} \frac{\xi_{C2} \phi_2}{1+\phi_2} & -\frac{\phi_2}{210} \frac{L^2(9+8\phi_2)+(504+252\phi_2)I'_{C33}}{L(1+\phi_2)^2} & \frac{6}{5} \frac{I'_{C23}(\phi_3+\phi_2+\phi_2\phi_3)}{(1+\phi_2)L(1+\phi_3)} & \frac{1}{60} \frac{\xi_{C3} L \phi_2}{1+\phi_2} & -\frac{1}{10} \frac{I'_{C23}(6\phi_3+\phi_2+\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} & -\frac{\phi_2}{840} \frac{L^2(11+9\phi_2)+(588+84\phi_2)I'_{C33}}{(1+\phi_2)^2} \\ -\frac{1}{2} \frac{\xi_{C3} \phi_3}{1+\phi_3} & \frac{6}{5} \frac{I'_{C23}(\phi_3+\phi_2+\phi_2\phi_3)}{(1+\phi_2)L(1+\phi_3)} & -\frac{\phi_3}{210} \frac{L^2(9+8\phi_3)+(504+252\phi_3)I'_{C22}}{L(1+\phi_3)^2} & -\frac{1}{60} \frac{\xi_{C2} L \phi_3}{1+\phi_3} & \frac{\phi_3}{840} \frac{L^2(11+9\phi_3)+(588+84\phi_3)I'_{C22}}{(1+\phi_3)^2} & \frac{1}{10} \frac{I'_{C23}(\phi_3+6\phi_2+\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} \\ 0 & \frac{1}{60} \frac{\xi_{C3} L \phi_2}{1+\phi_2} & -\frac{1}{60} \frac{\xi_{C2} L \phi_3}{1+\phi_3} & 0 & \frac{1}{120} \frac{\xi_{C2} L^2 \phi_3}{1+\phi_3} & \frac{1}{120} \frac{\xi_{C3} L^2 \phi_2}{1+\phi_2} \\ \frac{1}{4} \frac{\xi_{C3} L \phi_3}{1+\phi_3} & -\frac{1}{10} \frac{I'_{C23}(6\phi_3+\phi_2+\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} & \frac{\phi_3}{840} \frac{L^2(11+9\phi_3)+(588+84\phi_3)I'_{C22}}{(1+\phi_3)^2} & \frac{1}{120} \frac{\xi_{C2} L^2 \phi_3}{1+\phi_3} & -\frac{L\phi_3}{840} \frac{L^2(2+\phi_3)+(\mathbf{8}-168\phi_3)I'_{C22}}{(1+\phi_3)^2} & \frac{1}{20} \frac{LI'_{C23}(-\phi_3-\phi_2+4\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} \\ -\frac{1}{4} \frac{\xi_{C2} L \phi_2}{1+\phi_2} & -\frac{\phi_2}{840} \frac{L^2(11+9\phi_2)+(588+84\phi_2)I'_{C33}}{(1+\phi_2)^2} & \frac{1}{10} \frac{I'_{C23}(\phi_3+6\phi_2+\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} & \frac{1}{120} \frac{\xi_{C3} L^2 \phi_2}{1+\phi_2} & \frac{1}{20} \frac{LI'_{C23}(-\phi_3-\phi_2+4\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} & -\frac{L\phi_2}{840} \frac{L^2(2+\phi_2)+(\mathbf{8}-168\phi_2)I'_{C33}}{(1+\phi_2)^2} \end{bmatrix} \quad (B.14)$$

$$\mathbf{M}_{T12} = \begin{bmatrix} 0 & \frac{1}{2} \frac{\xi_{C2} \phi_2}{1+\phi_2} & \frac{1}{2} \frac{\xi_{C3} \phi_3}{1+\phi_3} & 0 & \frac{1}{4} \frac{\xi_{C3} L \phi_3}{1+\phi_3} & -\frac{1}{4} \frac{\xi_{C2} L \phi_2}{1+\phi_2} \\ -\frac{1}{2} \frac{\xi_{C2} \phi_2}{1+\phi_2} & \frac{\phi_2}{210} \frac{L^2(9+8\phi_2)+(504+252\phi_2)I'_{C33}}{L(1+\phi_2)^2} & -\frac{6}{5} \frac{I'_{C23}(\phi_3+\phi_2+\phi_2\phi_3)}{(1+\phi_2)L(1+\phi_3)} & -\frac{1}{60} \frac{\xi_{C3} L \phi_2}{1+\phi_2} & -\frac{1}{10} \frac{I'_{C23}(6\phi_3+\phi_2+\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} & -\frac{\phi_2}{840} \frac{L^2(11+9\phi_2)+(588+84\phi_2)I'_{C33}}{(1+\phi_2)^2} \\ -\frac{1}{2} \frac{\xi_{C3} \phi_3}{1+\phi_3} & -\frac{6}{5} \frac{I'_{C23}(\phi_3+\phi_2+\phi_2\phi_3)}{(1+\phi_2)L(1+\phi_3)} & \frac{\phi_3}{210} \frac{L^2(9+8\phi_3)+(504+252\phi_3)I'_{C22}}{L(1+\phi_3)^2} & \frac{1}{60} \frac{\xi_{C2} L \phi_3}{1+\phi_3} & \frac{\phi_3}{840} \frac{L^2(11+9\phi_3)+(588+84\phi_3)I'_{C22}}{(1+\phi_3)^2} & \frac{1}{10} \frac{I'_{C23}(\phi_3+6\phi_2+\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} \\ 0 & \frac{1}{60} \frac{\xi_{C3} L \phi_2}{1+\phi_2} & -\frac{1}{60} \frac{\xi_{C2} L \phi_3}{1+\phi_3} & 0 & \frac{1}{120} \frac{\xi_{C2} L^2 \phi_3}{1+\phi_3} & \frac{1}{120} \frac{\xi_{C3} L^2 \phi_2}{1+\phi_2} \\ \frac{1}{4} \frac{\xi_{C3} L \phi_3}{1+\phi_3} & \frac{1}{10} \frac{I'_{C23}(6\phi_3+\phi_2+\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} & -\frac{\phi_3}{840} \frac{L^2(11+9\phi_3)+(588+84\phi_3)I'_{C22}}{(1+\phi_3)^2} & -\frac{1}{120} \frac{\xi_{C2} L^2 \phi_3}{1+\phi_3} & -\frac{L\phi_3}{840} \frac{L^2(2+\phi_3)+(\mathbf{8}-168\phi_3)I'_{C22}}{(1+\phi_3)^2} & \frac{1}{20} \frac{LI'_{C23}(-\phi_3-\phi_2+4\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} \\ -\frac{1}{4} \frac{\xi_{C2} L \phi_2}{1+\phi_2} & \frac{\phi_2}{840} \frac{L^2(11+9\phi_2)+(588+84\phi_2)I'_{C33}}{(1+\phi_2)^2} & -\frac{1}{10} \frac{I'_{C23}(\phi_3+6\phi_2+\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} & -\frac{1}{120} \frac{\xi_{C3} L^2 \phi_2}{1+\phi_2} & \frac{1}{20} \frac{LI'_{C23}(-\phi_3-\phi_2+4\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} & -\frac{L\phi_2}{840} \frac{L^2(2+\phi_2)+(\mathbf{8}-168\phi_2)I'_{C33}}{(1+\phi_2)^2} \end{bmatrix} \quad (B.15)$$

$$\mathbf{M}_{T2} = \begin{bmatrix}
 0 & \frac{1}{2} \frac{\xi_{C2} \phi_2}{1+\phi_2} & \frac{1}{2} \frac{\xi_{C3} \phi_3}{1+\phi_3} & 0 & \frac{1}{4} \frac{\xi_{C3} L \phi_3}{1+\phi_3} & -\frac{1}{4} \frac{\xi_{C2} L \phi_2}{1+\phi_2} \\
 \frac{1}{2} \frac{\xi_{C2} \phi_2}{1+\phi_2} & -\frac{\phi_2}{210} \frac{L^2(9+8\phi_2)+(504+252\phi_2)I'_{C33}}{L(1+\phi_2)^2} & \frac{6}{5} \frac{I'_{C23}(\phi_3+\phi_2+\phi_2\phi_3)}{(1+\phi_2)L(1+\phi_3)} & \frac{1}{60} \frac{\xi_{C3} L \phi_2}{1+\phi_2} & \frac{1}{10} \frac{I'_{C23}(6\phi_3+\phi_2+\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} & \frac{\phi_2}{840} \frac{L^2(11+9\phi_2)+(588+84\phi_2)I'_{C33}}{(1+\phi_2)^2} \\
 \frac{1}{2} \frac{\xi_{C3} \phi_3}{1+\phi_3} & \frac{6}{5} \frac{I'_{C23}(\phi_3+\phi_2+\phi_2\phi_3)}{(1+\phi_2)L(1+\phi_3)} & -\frac{\phi_3}{210} \frac{L^2(9+8\phi_3)+(504+252\phi_3)I'_{C22}}{L(1+\phi_3)^2} & -\frac{1}{60} \frac{\xi_{C2} L \phi_3}{1+\phi_3} & -\frac{\phi_3}{840} \frac{L^2(11+9\phi_3)+(588+84\phi_3)I'_{C22}}{(1+\phi_3)^2} & -\frac{1}{10} \frac{I'_{C23}(\phi_3+6\phi_2+\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} \\
 0 & \frac{1}{60} \frac{\xi_{C3} L \phi_2}{1+\phi_2} & -\frac{1}{60} \frac{\xi_{C2} L \phi_3}{1+\phi_3} & 0 & -\frac{1}{120} \frac{\xi_{C2} L^2 \phi_3}{1+\phi_3} & -\frac{1}{120} \frac{\xi_{C3} L^2 \phi_2}{1+\phi_2} \\
 \frac{1}{4} \frac{\xi_{C3} L \phi_3}{1+\phi_3} & \frac{1}{10} \frac{I'_{C23}(6\phi_3+\phi_2+\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} & -\frac{\phi_3}{840} \frac{L^2(11+9\phi_3)+(588+84\phi_3)I'_{C22}}{(1+\phi_3)^2} & -\frac{1}{120} \frac{\xi_{C2} L^2 \phi_3}{1+\phi_3} & -\frac{L\phi_3}{840} \frac{L^2(2+\phi_3)+(84-168\phi_3)I'_{C22}}{(1+\phi_3)^2} & \frac{1}{20} \frac{L I'_{C23}(-\phi_3-\phi_2+4\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} \\
 -\frac{1}{4} \frac{\xi_{C2} L \phi_2}{1+\phi_2} & \frac{\phi_2}{840} \frac{L^2(11+9\phi_2)+(588+84\phi_2)I'_{C33}}{(1+\phi_2)^2} & -\frac{1}{10} \frac{I'_{C23}(\phi_3+6\phi_2+\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} & -\frac{1}{120} \frac{\xi_{C3} L^2 \phi_2}{1+\phi_2} & \frac{1}{20} \frac{L I'_{C23}(-\phi_3-\phi_2+4\phi_2\phi_3)}{(1+\phi_2)(1+\phi_3)} & -\frac{L\phi_2}{840} \frac{L^2(2+\phi_2)+(8-168\phi_2)I'_{C33}}{(1+\phi_2)^2}
 \end{bmatrix} \quad (B.16)$$

Literaturverzeichnis

- [Bal98] J. Ballmann. *Grundlagen der Dynamik*. Wissenschaftsverlag Mainz, Aachen, 5. Auflage, 1998. Vorlesungsskript.
- [Bal03] J. Ballmann, Herausgeber. *Flow Modulation and Fluid–Structure Interaction at Airplane Wings, Research Results of the Collaborative Research Center SFB 401 at RWTH Aachen*, Band 84 d. Reihe *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*. Springer, Berlin, 2003.
- [Bar98] J. Barnes. *Programming in Ada95*. Addison Wesley Longman LTD, Harlow, Essex, 2. Auflage, 1998.
- [Bat89] J. T. Batina. Unsteady Euler algorithm with unstructured dynamic mesh for complex aircraft aeroelastic analysis. *AIAA-89-1189*, 1989.
- [BBB⁺03] C. Braun, A. Boucke, J. Ballmann, M. Hanke, und A. Karavas. Numerical prediction of the model deformation of a high speed transport aircraft type wing by direct aeroelastic simulation. Technischer bericht, RWTH Aachen, Lehr- und Forschungsgebiet für Mechanik, 2003. HiReTT, European High Reynolds-Number Tools & Techniques Project, HIRETT/TR/RWTH/CB/18032003/1.
- [BBC⁺94] R. Barret, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, und H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Software, Environments, Tools. SIAM, Philadelphia, PA, 1994.
- [BBH⁺01] J. Ballmann, G. Britten, J. Hurka, M. Hesse, und C. Braun. Computational aeroelasticity in aeroelastic wing design with reduced structural models. In H. Hönlinger, Herausgeber, *CEAS Conference on Multidisciplinary Aircraft Design and Optimization*, DGLR-Bericht 2001-05, Seiten 167–176, Bonn, 2001.
- [BD95] O. O. Bendiksen und G. A. Davis. Nonlinear travelling wave flutter of panels in transonic flow. *AIAA 95-1496*, 1995.
- [BH97] O. O. Bendiksen und G.-Y. Hwang. Nonlinear flutter calculations for transonic wings. In *CEAS International Conference on Aeroelasticity and Structural Dynamics*, Band II, Seiten 105–114, Rome, 1997.

- [Boo86] G. Booch. *Software Components with Ada: Structures, Tools and Subsystems*. The Benjamin/Cummings Publishing Company, Inc., 2. Auflage, 1986.
- [Bra92] K.-H. Brackhage. Numerical treatment of surface-surface-intersection and contouring. In T. Lyche und L. L. Schumaker, Herausgeber, *Mathematical Methods in Computer Aided Geometric Design II*. Academic Press, 1992.
- [Bri03] G. Britten. *Numerische Aerostrukturodynamik von Tragflügeln großer Spannweite*. Dissertation, RWTH Aachen, 2003. Shaker Verlag, Aachen.
- [CCF96] F. Chaitin-Chatelin und V. Frayssé. *Lectures on Finite Precision Computations*. Software, Environments, Tools. SIAM, Philadelphia, 1996.
- [CGV96] J.-M. Chesneaux, S. Guilain, und J. Vignes. *La Bibliothèque CADNA, Présentation et utilisation*. Université Pierre et Marie Curie, Paris VI, Nov. 1996.
- [CL97] J. R. Cebral und R. Löhner. Conservative load projection and tracking for fluid-structure problems. *AIAA Journal*, 35(4):687–692, 1997.
- [Coh96] N. Cohen. *Ada as a Second Language*. McGraw-Hill, New York, 2. Auflage, 1996.
- [Czi89] H. Czichos, Herausgeber. *Hütte: die Grundlagen der Ingenieurwissenschaften*. Springer, Berlin, 29. Auflage, 1989.
- [DF02] C. Degand und C. Farhat. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computers and Structures*, 80(3-4):305–316, 2002.
- [DoD78] Department of defense requirements for high order computer programming languages: “steelman”. Technischer bericht, U.S. Department of Defence, 1978. <http://www.adahome.com/History/Steelman/intro.htm>.
- [FK93] Z. Friedman und J. B. Kosmatka. An improved two-node Timoshenko beam finite element. *Computers & Structures*, 47(3):473–481, 1993.
- [FLL98] C. Farhat, M. Lesoinne, und P. LeTallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157:95–114, 1998.

- [FLO00] Deutsches Zentrum für Luft- und Raumfahrt, Institut für Entwurfsaerodynamik, Braunschweig. *FLOWer: Installation and User Handbook, Release 116*, Mai 2000.
- [FPD01] C. Farhat, K. Pierson, und C. Degand. Multidisziplinäre Simulation of the maneuvering of an aircraft. *Engineering with Computers*, 17:16–27, 2001.
- [GA] J. Gobat und D. Atkinson. FElT, Software für Finite-Elemente-Berechnungen. <http://felt.sourceforge.net>.
- [Gre97] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. Nr. 17 in *Frontiers in Applied Mathematics*. SIAM, Philadelphia, PA, 1997.
- [Han03] M. Hanke. *Bestimmung erweiterter dynamischer Derivativa allgemeiner Flugkörper durch numerische Simulation der aeroelastischen Vorgänge*. Dissertation, RWTH Aachen, 2003.
- [HB01] J. Hurka und J. Ballmann. Elastic panels in transonic flow. In *31st AIAA Fluid Dynamics Conference*, Band AIAA paper 2001-2722, Reston, VA 20191-4344, June 2001. AIAA.
- [HBB00] M. Hesse, G. Britten, und J. Ballmann. A multi-block grid deformation algorithm for aeroelastic analysis. In B. K. Soni, J. Häuser, J. F. Thomson, und P. Eisenau, Herausgeber, *7th Int. Conference on Numerical Grid Generation in Computational Field Simulations*, Seiten 161–170, Whistler, Canada, 2000.
- [HCAK02] A. Hosangadi, P. A. Cavallo, S. Arunajatesan, und K. Kennedy. Parallel structured/unstructured simulation of missile dynamic flowfields. DoD HPCMP User's Group Conference 2002, University of Texas, Austin, June 2002. Vortragsfolien.
- [Hön01] H. Hönlinger, Herausgeber. *CEAS Conference on Multidisciplinary Aircraft Design and Optimization*, DGLR-Bericht 2001-05, Bonn, 2001.
- [Hug87] T. J. R. Hughes. *The Finite Element Method*. Prentice Hall, Englewood Cliffs, 1987.
- [Hur02] J. Hurka. *Numerische Untersuchung zur Aeroelastik dünner Platten*. Dissertation, RWTH Aachen, 2002.
- [JT94] A. A. Johnson und T. E. Tezduyar. Mesh updating strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 119:73–94, 1994.
- [Kaz00] D. A. Kazakov. Ada95 Bibliotheken strings_edit und units. http://www.dmitry-kazakov.de/reusable_code.htm, 2000.

- [KHS] W.-R. Krüger, R. Heinrich, und M. Spiek. Fluid-structure coupling using CFD and multibody simulation methods. In *ICAS 2002 Congress*. DLR.
- [KRBT00] N. Kroll, C.-C. Rossow, K. Becker, und F. Thiele. The MEGAFLOW Project. *Aerospace Science and Technology*, 4(10):223–237, 2000.
- [Lau90] H. D. Lauermann. *Ein Charakteristiken-Algorithmus zur Berechnung der instationären Transversalschwingungen von Hubschrauberrotorblättern oder anderer quasi-eindimensionaler elastischer Strukturen*. Dissertation, RWTH Aachen, 1990.
- [Mar84] A. Martin. *Ein Finite-Elemente-Verfahren für den Timoshenko-Balken unter bewegter Punktmasse mit Berücksichtigung lokaler Unstetigkeiten*. Dissertation, RWTH Aachen, 1984.
- [Mas02] Ralf Massjung. *Numerical Schemes and Well-Posedness in Nonlinear Aeroelasticity*. Dissertation, RWTH Aachen, 2002.
- [Mil68] R. D. Milne. Some remarks on the dynamics of deformable bodies. *AIAA Journal*, 6(3):556–558, March 1968.
- [MM01] C. H. Murphy und W. H. Mermagen Sr. Flight mechanics of an elastic symmetric missile. *Journal of Guidance, Control, and Dynamics*, 24(6):1125–1132, Nov/Dec 2001.
- [MNF01] K. Maute, M. Nikbay, und C. Farhat. Coupled analytical sensitivity analysis and optimization of three-dimensional nonlinear aeroelastic systems. *AIAA Journal*, 39(11):2051–2061, Nov. 2001.
- [Mul87] J.-M. Muller. *Arithmetique des Ordinateurs*. Ed. Masson, 1987.
- [Nel96] D. Nellessen. *Schallnahe Strömungen um elastische Tragflügel*. Dissertation, RWTH Aachen, 1996. Fortschritt-Berichte VDI Reihe 7, Strömungstechnik, Nr 302.
- [Pav79] W. Pavel. *Berechnung von Scher- und BiegeWellen im Timoshenko-Balken unter bewegten Punktlasten mit Hilfe der Charakteristikentheorie*. Diplomarbeit, RWTH Aachen, Lehr- und Forschungsgebiet für Mechanik, 1979.
- [PFTV86] W. H. Press, B. P. Flannery, S. A. Teukolsky, und W. T. Vetterling. *Numerical Recipes*. Cambridge University Press, 1986.
- [Sch80] H. R. Schwarz. *Methode der Finiten Elemente*. Studienbücher Mathematik. Teubner, Stuttgart, 1980.
- [Sch97] S. Schulze. Transonic aeroelastic simulation of a flexible wing section. In *AGARD SMP Meeting on "Numerical Unsteady Aerodynamic and Aeroelastic Simulation"*, veröffentlicht in R-822, Aalborg, Denmark, Oktober 1997.

- [SGH95] W. Schnell, D. Gross, und W. Hauger. *Technische Mechanik*, Band 2 Elastostatik d. Reihe *Springer Lehrbuch*. Springer, Heidelberg, 5. Auflage, 1995.
- [Sti98] M. J. Stift. Scientific programming with Ada95: Object-oriented, parallel, and safe. *Computers in Physics*, 12(2):150–158, Mar/Apr 1998.
- [TD97] S. Tucker Taft und R. A. Duff, Herausgeber. *Ada 95 Reference Manual: Language and Standard Libraries, International Standard ISO/IEC 8652:1995(E)*. Springer, Berlin, 1997.
- [TÖG00] Ö. Tanrıku, C. Önen, und T. Gökhan. Nonlinear yaw-pitch-roll coupling of unguided missiles with wraparound fins. *J. Spacecraft Rockets*, 37(6):731–739, 2000.
- [Wer97] M. Werle. Gittergenerierung für die Strömungssimulation von elastischen Auftriebskörpern. Diplomarbeit, RWTH Aachen, Lehr- und Forschungsgebiet für Mechanik, 1997.
- [Wic] A. Wick. Grid deformation techniques for two-dimensional hybrid grids. Hermann-Föttinger Institute for Fluid Dynamics, TU Berlin.
- [Wic01] A. Wick. Generation of dynamic grids using structural analogy. In *Proceedings of the 17th GAMM-Seminar*, Seiten 147–157, Leipzig, 2001. Max-Planck-Institute for Mathematics in the Sciences.
- [Wri02] S. Wright. The Ada95 Booch Components. <http://www.adapower.net/booch/>, <http://www.rivatech.com/booch>, 2002.
- [WS98] M. R. Waszak und D. K. Schmidt. Flight dynamics of aeroelastic vehicles. *J. Aircraft*, 25(6):563–571, June 1998.

Lebenslauf

Alexander Boucke

| | | |
|-----------------------|--|------------------------|
| Geburtsdatum: | 8. Januar 1969 | |
| Geburtsort: | Aachen | |
| Eltern: | Dipl.–Architekt Michael Boucke Hedda Boucke, geb. Fladda | |
| Familienstand: | ledig | |
| Schulbildung: | Ev. Grundschule Annastraße, Aachen | 1975–1979 |
| | Kaiser–Karls–Gymnasium, Aachen | 1979–1988 |
| Wehrdienst: | Heeres–Musik–Korps Lüneburg | 1988–1989 |
| Hochschulausbildung: | Studium der Mathematik an der RWTH Aachen | 1989–1995 |
| Studienabschlüsse: | Vordiplom | 20.9.1991 |
| | Diplom–Mathematiker | 13.1.1995 |
| Berufliche Tätigkeit: | Studentische Hilfskraft am Lehr- und Forschungs- gebiet für Mechanik, RWTH Aachen | 1990-1994 |
| | Wissenschaftlicher Angestellter am Lehr- und For- schungsgebiet für Mechanik, RWTH Aachen | 1.4.1995- 31.5.2000 |
| | Ingenieurbüro ITAM GmbH, Herzogenrath– Kohlscheid | 1.6.2000- 31.7.2002 |
| | Wissenschaftlicher Angestellter am Lehr- und For- schungsgebiet für Mechanik, RWTH Aachen | 1.8.2002- 31.5.2003 |