

Lehrstuhl für Bauinformatik  
Fakultät für Bauingenieur- und Vermessungswesen  
Technische Universität München

**Tragwerksanalyse am volumenorientierten Gesamtmodell -  
Ein Ansatz zur Verbesserung der computergestützten  
Zusammenarbeit im konstruktiven Ingenieurbau**

Andreas Konrad Niggel

Vollständiger Abdruck der von der Fakultät für Bauingenieur- und Vermessungswesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing., Dr.-Ing. habil. G. H. Müller

Prüfer der Dissertation:

1. Univ.-Prof. Dr.rer.nat. E. Rank
2. Univ.-Prof. Dr.-Ing. M. Schnellenbach-Held,  
Universität Duisburg Essen

Die Dissertation wurde am 28.12.2006 bei der Technischen Universität München eingereicht und durch die Fakultät für Bauingenieur- und Vermessungswesen am 20.06.2007 angenommen.



*Für Carolin.*



## Zusammenfassung

In einem Planungsprozess des konstruktiven Ingenieurbaus arbeiten die Beteiligten, die aus ganz unterschiedlichen Fachdisziplinen stammen, in der Regel raum-zeitlich verteilt. Sie bearbeiten und analysieren ihr Modell oft gleichzeitig in verschiedenen Stufen der Detaillierung. Viele Computeranwendungen und die darin verwendeten Datenmodelle sind dagegen nur für einen speziellen Aufgabenbereich konzipiert. Es existiert eine Vielzahl von Teilmodellen mit ganz unterschiedlicher fachspezifischer Semantik, was die Konsistenthaltung und den Austausch digitaler Daten erschwert. Aus dieser Motivation heraus schlägt die vorliegende Arbeit vor, Arbeitsprozesse der konstruktiven Tragwerksplanung auf strikt volumenorientierte Modelle zu gründen. Um eine statische Analyse durchzuführen, wird ein Tragwerk üblicherweise je nach Tragverhalten in dimensionsreduzierte Teilstrukturen, wie Balken, Platten oder Stabwerke eingeteilt. Im Gegensatz dazu soll nun für alle Aufgaben ein einheitlicher, volumenorientierter Ansatz basierend auf finiten Elementen hoher Ordnung zum Einsatz kommen. Dadurch lässt sich insgesamt die Komplexität aus der Verwendung unterschiedlicher Teilmodelle verringern und die Integration der Tragwerkssimulation in den Gesamtplanungsprozess verbessern. Um zudem eine Bearbeitung mehrerer Planer an einem gemeinsamen Modell zu ermöglichen, sollen alle Berechnungsaufgaben verschiedener Detaillierungsstufen, von der Analyse des globalen Verhaltens bis hin zu lokalen Detailuntersuchungen, immer am gemeinsamen Modell des Gesamtsystems erfolgen. Damit kann sowohl geometrische als auch mechanische Konsistenz leicht gewährleistet werden.

Das Ziel dieser Arbeit liegt darin, ein solches Modell zu beschreiben, einige sich daraus ergebenden Fragen zu diskutieren und Lösungsmöglichkeiten aufzuzeigen. Im Rahmen eines interdisziplinären Forschungsprojekts wurde hierzu ein Softwarekonzept zur verteilten Struktursimulation entwickelt und prototypisch umgesetzt. Darin wird ein Berechnungsmodell zentral vorgehalten, welches von verschiedenen Planern gleichzeitig bearbeitet werden kann. Wesentlich an dem Konzept ist eine räumlich-hierarchische Organisation des zentralen Modells. Damit lässt sich zum einen das Berechnungsmodell um raumbezogene Informationen erweitern, zum anderen erlaubt die Hierarchie die Anwendung von Lösungsverfahren, die nach dem Prinzip der rekursiven Substrukturierung arbeiten. Diese bieten den Vorteil, dass bei lokalen Modifikationen am Modell der Aufwand für eine Neuberechnung weitgehend auf den modifizierten Bereich beschränkt werden kann, wodurch sich gerade bei wiederholten Änderungen am System die Rechenzeiten deutlich reduzieren lassen.

Ein besonderes Augenmerk liegt zudem auf der Frage, inwieweit sich die volumenorientierten Berechnungsergebnisse für den konstruktiven Bemessungsentwurf im Stahlbetonbau nutzen lassen und welche Vorteile sich daraus ergeben. Die Idee der hier entwickelten Methode liegt darin, die Dimensionsreduktion nicht mehr vor, sondern nach einer Berechnung durchzuführen. Dadurch eröffnen sich Möglichkeiten zu einer flexiblen und interaktiven Arbeitsweise. Hierzu werden in der vorliegenden Arbeit Hilfsmittel vorgestellt, die eine schnelle Integration von Spannungsgrößen und zudem eine Erkennung von Bauteilformen erlauben. Darüber hinaus lassen sich die volumenorientierten Ergebnisgrößen auch nutzen, um die Gültigkeit von Modellannahmen der normativen Nachweise zu überprüfen. Bereiche, in denen allgemeine Spannungszustände herrschen, werden automatisch erkannt. Das gibt dem Ingenieur ein zusätzliches Entscheidungshilfsmittel an die Hand, die passenden Bemessungsverfahren auszuwählen.



# Abstract

People involved in building design processes usually work on one common project from scattered locations and on an unsynchronised time schedule. It is also a known fact that, especially in building industry, the participants originate from many different fields of expertise. As a result, there are a variety of computer applications with domain specific data representations which are generally not designed to facilitate the exchange of data or support their integration into an overall computer-based planning process. Motivated by these circumstances, this thesis sets out to present a new approach, in which planning processes in civil engineering are based on entirely three-dimensional, volume-oriented models. The focus is on applications within the scope of structural analysis and design. In order to perform a computation, a building structure is no longer divided into a set of dimensionally reduced components, such as plates, slabs or beams, depending on their respective mode of load transmission. Instead, all structures are strictly modeled according to volume using high-order, hexahedral finite elements. The consistent geometry between the architectural and structural domain makes it easier to integrate different computational tasks on the basis of this common data representation. It is possible to perform all the analytical tasks – from local detail studies to investigations of the global behaviour – on a single model, which implicitly ensures both geometrical and mechanical consistency even when multiple planners need to work simultaneously.

The aim of this thesis is to describe this approach in detail, to discuss questions that might arise and to present possible solution concepts. Parts of this essay originate from an interdisciplinary research project in which a software framework was developed for cooperative work in simulation processes. The framework provides a centrally stored model, which can be accessed concurrently by different planners as well as providing mechanisms for detecting geometric conflicts. An important aspect of this concept is the hierarchical organisation of the central simulation model. By using octrees, this hierarchy can not only be used to extend the internal data to include spatial information but also to organize the computation of the finite element problem. In the latter case, this is done by applying recursive substructuring methods. An important aspect of this kind of equation solver is that, in case of local modifications, the effort of re-computing the whole structure can be mainly restricted to the modified parts. Examples will show that, in such cases, it is possible to reduce the computational cost by an order of magnitude.

Important questions arise when the three-dimensional result values of the volume-oriented computation are required for designing building structures with reinforced concrete. The design codes are usually based on results computed on dimensionally reduced systems. The main idea of the approach described in this thesis is to continue using integrated stress resultants for design while applying the dimensional reduction after the computation stage, rather than beforehand. In this case, instead of being part of the modeling process, the dimensional reduction is only a matter of post-processing. This approach allows for a much more flexible, interactive work in designing reinforced concrete structures. This thesis accordingly presents an algorithm for the fast integration of cross-sectional values. In addition, the volume-oriented approach also provides the opportunity to verify the basic model assumptions of the design codes. Areas with general stress conditions, where dimensionally reduced models cease to be valid, can be automatically detected and visualized, which might be an additional source of information for the engineer.



## Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Bauinformatik der Technischen Universität München. Sie wurde im Rahmen des Projekts „Volumenorientierte Modellierung als Grundlage einer vernetzt kooperativen Planung im konstruktiven Ingenieurbau“ von der Deutschen Forschungsgemeinschaft gefördert. An dieser Stelle möchte ich mich bei all den Menschen bedanken, die zum Gelingen dieser Arbeit beigetragen haben.

Ganz besonders möchte ich mich bei meinem Doktorvater, Herrn Prof. Dr.rer.nat. Ernst Rank bedanken. Seine Begeisterungsfähigkeit für die unterschiedlichsten Themen machten es leicht, eigene Ideen zu entwickeln und diese mit dem notwendigen Freiraum zu verfolgen.

Frau Prof. Dr.-Ing. Martina Schnellenbach-Held danke ich für ihr Interesse an meiner Arbeit und die Übernahme des Zweitgutachtens.

Ein weiterer Dank geht an PD Dr.-Ing. Alexander Düster. Schon während meiner Studienzeit weckte er in mir das Interesse an der Bauinformatik und stand mir danach während der ganzen Jahre am Lehrstuhl jederzeit mit Rat und Tat zur Seite.

Bedanken möchte ich mich auch bei meinen Projekt-Kollegen Dr.-Ing. Richard Romberg und Dr.rer.nat. Ralf Mundani für die gute Zusammenarbeit, die wichtigen Diskussionen sowie bei Ralf Mundani für die interessanten Einblicke in die Arbeits- und Denkweise der klassischen Informatik. Ein Dank geht auch an Jérôme Frisch, der mir mit seinem unermüdlichen Einsatz viel Programmierarbeit abnahm.

Bei allen Kolleginnen und Kollegen des Lehrstuhls für Bauinformatik bedanke ich mich für das sehr gute Arbeitsklima und das freundschaftliche Miteinander. An dieser Stelle möchte ich besonders meine beiden langjährigen Zimmernachbarn, Frau Dr.-Ing. Vera Nübel und Herrn Prof. Jamshid Parvizian hervorheben und ihnen für die angenehme Arbeitsatmosphäre und die schöne gemeinsame Zeit im Büro danken.

Vor allem danke ich aber meiner Frau Carolin. Ihre Zuneigung und Liebe gibt mir immer den notwendigen Rückhalt.

Andreas Niggel,  
November 2007



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Motivation</b>	<b>1</b>
1.1	Planung im Bauwesen	2
1.2	Computergestützte Zusammenarbeit	4
1.2.1	Kommunikation, Koordination, Kollaboration	4
1.2.2	Standardisierte Bauwerksmodelle	5
1.3	Tragwerksplanung im konstruktiven Ingenieurbau	6
1.4	Volumenmodellierung als Basis der Integration	9
1.5	Zielsetzung und Gliederung der Arbeit	12
<b>2</b>	<b>Tragwerkssimulation am Volumenmodell</b>	<b>15</b>
2.1	Vom Gebäudemodell zum volumenorientierten Finite-Elemente-Netz	15
2.1.1	Ein erweitertes Geometriemodell als Grundlage	16
2.1.2	Zerlegung in ein Verbindungsmodell	17
2.1.3	Vernetzung der Struktur	20
2.2	Die $p$ -Version der Finite-Elemente-Methode	22
2.2.1	Hierarchische Ansatzfunktionen	22
2.2.2	Hexaederelemente zur Berechnung dünnwandiger Strukturen	25
2.2.3	Abbildung gekrümmter Elementoberflächen	27
2.3	Beispiel: UNIQA-Tower	30
<b>3</b>	<b>Hierarchie als Grundlage der Modell- und Berechnungsorganisation</b>	<b>35</b>
3.1	Geometrische Modellierung	36
3.2	Räumlich hierarchische Geometriemodelle	37
3.2.1	Baumdatenstrukturen	37
3.2.2	Quadtrees, Octrees	38
3.2.3	Darstellung von Bäumen	42
3.3	Das Prinzip der rekursiven Substrukturierung	44
3.3.1	Statische Kondensation und Lösung der FE-Gleichungen	45
3.3.2	Diskussion	48
3.4	Substrukturierung in der $p$ -Version der FEM	52
3.4.1	Rekursive Zerlegung des Modells	53
3.4.2	Hierarchische Sortierung aller Freiheitsgrade	56
3.4.3	Assemblierung und Lösung des Gleichungssystems	58
3.4.4	Berechnungsbeispiele	59
3.5	Vorteile der Hierarchie	74
3.5.1	Partielle Neuberechnung	74

---

3.5.2	Verteilung von Daten und Berechnung . . . . .	78
<b>4</b>	<b>Kooperatives Arbeiten am dreidimensionalen Modell</b>	<b>81</b>
4.1	Ein Framework für verteilt kooperatives Arbeiten . . . . .	82
4.1.1	Kontrollbaum zur räumlichen Organisation . . . . .	83
4.1.2	Regulierung von Zugriffen . . . . .	84
4.1.3	Einbettung von Simulationsaufgaben . . . . .	88
4.2	Simulation von Bauphasen . . . . .	90
4.2.1	Softwarerahmen . . . . .	90
4.2.2	Beispiel: Brücke im Freivorbau . . . . .	94
<b>5</b>	<b>Volumenmodelle in der Bemessung von Stahlbetonbauteilen</b>	<b>97</b>
5.1	Bemessung im konstruktiven Betonbau . . . . .	98
5.1.1	Aufgaben der Bemessung . . . . .	98
5.1.2	Betrachtungsebene: Querschnitt . . . . .	99
5.1.3	Betrachtungsebene: System . . . . .	101
5.2	Volumenmodellierung als Grundlage . . . . .	104
5.2.1	Dimensionsreduktion der Spannungen . . . . .	104
5.2.2	Aspekte der Tragwerksidealisierung . . . . .	114
5.2.3	Erkennung von B- und D-Zonen . . . . .	117
5.3	Hilfsmittel zur grafisch interaktiven Bemessung . . . . .	120
5.3.1	Erkennung von Bauteilform und Bauteilachse bzw. -fläche . . . . .	120
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>125</b>
<b>A</b>	<b>Bestimmung der Kovarianzen einer konvexen Hülle</b>	<b>129</b>
A.1	Kovarianzen einer mehrdimensionalen Zufallsgröße . . . . .	129
A.2	Berechnung der Kovarianzen einer triangulierten Oberfläche . . . . .	130
	<b>Literaturverzeichnis</b>	<b>133</b>

# Kapitel 1

## Einleitung und Motivation

In der Planung von Ingenieurbauwerken ist der Einsatz von Computertechnik heutzutage nicht mehr wegzudenken. Dennoch ist zu erkennen, dass besonders im Bauingenieurwesen die Durchdringung mit Informationstechnik noch nicht den Umfang erreicht hat, wie in anderen Bereichen der Industrie oder der Dienstleistung. Dies zeigt ein Beispiel aus dem Flugzeugbau. Im Jahre 2005 wurde beim französischen Luftfahrtkonzern DASSAULT in Zusammenarbeit mit allen weltweit verstreuten Zulieferern erstmalig die Entwicklung eines Passagierflugzeuges vollständig auf Basis eines gemeinsamen digitalen Computermodells durchgeführt [45]. Nachdem das Flugzeug detailgetreu als Modell am Computer erstellt wurde, wurden alle Schritte des Entwurfs, der Produktion, des Betriebs bis hin zu möglichen Wartungsarbeiten ausschließlich am Rechner simuliert. Der Konzern ließ am Ende das Flugzeug bauen (und natürlich auch fliegen), ohne auch nur einen Prototyp vorher kostenintensiv entwickelt und getestet zu haben. Die wirtschaftliche Bedeutung, die einer derartigen Planung am digitalen Modell zukommt, zeigt ein weiteres Beispiel aus der Flugzeugindustrie. Die Verzögerungen und die damit verbundenen Kosten in Milliardenhöhe, die der europäische Flugzeughersteller AIRBUS bei der Auslieferung des neuen A380 zu erleiden hat, liegen teilweise an inkompatiblen Softwaresystemen, die in den europaweit verteilten Produktionsstätten verwendet werden [1]. Fehlende Einbauuntersuchungen am dreidimensionalen Computermodell (*3D Digital Mockup*) führten dazu, dass sich beim Bau die Verkabelung nicht wie gewünscht im Flugzeuginnenraum verlegen ließ [17].

Die Beispiele zeigen, dass aufgrund einer steigenden Komplexität der Produkte und kürzeren Entwicklungszeiten einer Planung am digitalen Modell und hier insbesondere einer Planung am *dreidimensionalen Modell* eine immer größere Bedeutung zukommt. Während in anderen Industriezweigen durchaus ähnliche Entwicklungen zu beobachten sind, ist in der Bauindustrie eine Integration aller Planungsprozesse auf diesem Niveau noch nicht gegeben. Viele Planungsaktivitäten basieren auf fachspezifischen Teilmodellen (2D-Zeichnungen, dimensionsreduzierte Strukturen), was den Austausch digitaler Daten und damit eine effektive Zusammenarbeit zwischen den Planungsbeteiligten erschwert. Die Gründe hierzu liegen sicherlich zu einem großen Teil an den besonderen Eigenschaften von Bauwerken und den daraus resultierenden Randbedingungen im Planungs- und Bauprozess, welche sich von denen anderer Industriebereiche unterscheiden. Diese Gegebenheiten bilden somit den Rahmen, in dem Informationstechnik im Ingenieurbau stattfindet. Sie werden im ersten Abschnitt angesprochen, bevor dann Aspekte der rechnergestützten Zusammenarbeit diskutiert und schließlich als zentrales Thema dieser Arbeit ein Ansatz vorgestellt werden soll, der eine Planung von Tragwerksstrukturen auf strikt

dreidimensionale Volumenmodelle gründet.

## 1.1 Planung im Bauwesen

Der Lebenszyklus eines Bauwerks gliedert sich in unterschiedliche Phasen. Nach EASTMAN [46] lassen sich sechs Abschnitte unterscheiden: *Machbarkeitsstudie*, *Planung*, *Ausführungsplanung*, *Bauausführung*, *Betreiben*, *Rückbau*. Jede dieser Phasen besteht aus einer Reihe von Arbeitsschritten, die teils sequentiell, teils zyklisch ablaufen und in die verschiedene Personen beziehungsweise Arbeitsgruppen involviert sind, die jeweils ganz unterschiedliche Sichten auf das gemeinsame Bauobjekt besitzen. Geschichtlich gewachsen – aber vor allem aufgrund der besonderen Randbedingungen – haben sich im Hinblick auf den Einsatz von Computertechnik Strukturen gebildet, die äußerst heterogen sind mit einer Vielfalt unterschiedlicher Spezialanwendungen und darin verwendeter Datenmodelle. Folgende Liste soll einige dieser Randbedingungen deutlich machen. Die Argumentation folgt dabei im Wesentlichen der von EASTMAN [46], wobei hier der Schwerpunkt auf dem Planungsprozess selbst liegt.

- **Iterative Arbeitsweise:** Planungsprozesse finden im Wechsel von Entwurf, Untersuchung der Auswirkungen und Modifikation des Entwurfs statt. Insbesondere wenn Pläne detailliert ausgearbeitet werden, kann oft die Notwendigkeit entstehen, dass Aspekte früher getroffener Entscheidungen nochmals überarbeitet werden müssen. Meist finden derartige iterative Zyklen nicht nur innerhalb eines Arbeitsbereiches statt, sondern erstrecken sich über Fachbereichsgrenzen hinweg, wenn etwa zwischen Architekt und Tragwerksplaner statisch notwendige Modifikationen am Bauwerksentwurf vorgenommen werden müssen oder Zwänge in der Bauausführung eine Änderung notwendig machen.
- **Top-down Vorgehen:** Planungstätigkeiten bestehen zu einem großen Teil aus einer sukzessiven Detaillierung einmal getroffener Konzepte. Am Anfang steht in der Regel ein Entwurf eines Architekten, welcher im Verlauf der Planung immer weiter ausgearbeitet wird, bis beispielsweise detaillierte Konstruktions- und Ausführungspläne erstellt sind. Dabei werden auch immer wieder Iterationen zwischen unterschiedlichen Niveaus der Detaillierung stattfinden.
- **Vielzahl an Fachdisziplinen:** Bei der Planung von Bauwerken sind immer unterschiedliche Fachdisziplinen beteiligt. Im Vergleich mit anderen Industriezweigen wird dies besonders deutlich, wenn man allein die Vielzahl an Materialien betrachtet, die in Bauwerken eingesetzt werden (z.B. Stahlbeton, Holz, Stahl), für welche jeweils ganz unterschiedliche Techniken erforderlich sind. Darüber hinaus müssen in Baumaßnahmen auch immer wieder weitere Spezialdisziplinen eingebunden werden, wenn etwa Fragen der Geologie, der Städteplanung oder des Umweltschutzes berücksichtigt werden müssen. Neben den rein technischen Gesichtspunkten sind natürlich auch immer Aspekte des Managements, der Kalkulation oder der Bauausführung zu berücksichtigen.
- **Wechselnde Zusammensetzung der Arbeitsgruppen:** Im Laufe einer Planung müssen stetig neue Personengruppen in den Entwicklungsprozess eingebunden werden. Steht am Anfang vielleicht nur ein Architekt und ein Bauherr, so kommen im Weiteren Fachleute der Tragwerksplanung, technischen Gebäudeausstattung, Kalkulation oder Bau-

ausführung hinzu. Die Anzahl der Beteiligten ändert sich: Während zu Beginn nur wenige Personen am Entwurf beteiligt sind, erweitert sich der Personenkreis, wenn detaillierte Ausführungspläne und Konstruktionszeichnungen zu erstellen sind.

- Geringe Strukturierung: Prozesse der Bauplanung sind oft nur schwach strukturiert und weisen eine hohe Dynamik auf. Aufgrund ihres stark iterativen Charakters ist meist kaum eine zeitliche oder inhaltsorientierte Strukturierung der einzelnen Aktivitäten zu erkennen. Zudem können während einer Planung immer wieder Ausnahmesituationen (z.B. Grundbau) auftreten, die schnelle und kreative Lösungskonzepte erfordern. Anders als in der Entwicklung standardisierter Industrieprodukte erweist sich daher bei der Errichtung von Bauwerken eine formale Steuerung der Aktivitäten – etwa mittels Prozessmodellen – als schwieriger.
- Unikatcharakter von Bauwerken: Dieser Umstand wirkt sich direkt auf den Anteil an finanziellen Mitteln aus, die für die Planung eines einzelnen Bauwerks herangezogen werden können. Vergleicht man in dem Zusammenhang das Verhältnis zwischen Entwicklungskosten und dem Wert eines konkreten Endprodukts in unterschiedlichen Industriezweigen, wird schnell deutlich, welchen Randbedingungen die Bauplanung unterworfen ist. EASTMAN [46] verdeutlicht dies am Beispiel der Halbleiterentwicklung. Der finanzielle Aufwand, der zur Entwicklung eines neuen Prozessors eingesetzt wird, liegt um Größenordnungen über dem Preis, mit dem ein einzelner Prozessor an den Kunden verkauft wird. Nur über hohe Stückzahlen lässt sich ein Gewinn erzielen. Bei Bauwerken verhält es sich umgekehrt. Bauwerke werden nur selten aus Serienbauteilen gefertigt, die Planungskosten betragen oft nur ein Zehntel der Bausumme. EASTMAN schreibt dazu: *„From this perspective, it is surprising that buildings can be designed that are interesting and sometimes innovative. When compared to other design fields, it is truly impressive that an architect can undertake the design of a building for a tenth or less of its construction cost.“*
- Heterogene Struktur der Bauindustrie: Im Vergleich zu anderen Sparten der produzierenden Industrie weist das Bauhauptgewerbe eine stark heterogene Struktur und eine weitgehend niedrige Kapitalisierungsquote der beteiligten Unternehmen auf. Von den insgesamt 86,6 Milliarden Euro Umsatz die in Deutschland im Jahre 2003 im Bauhauptgewerbe getätigt wurden, fielen nur 17,6 Prozent innerhalb von Unternehmen mit mehr als 500 Beschäftigten an. Der größere Teil von 31,5 Prozent entfiel auf Kleinstunternehmen mit maximal 19 Beschäftigten. Die durchschnittliche Zahl der Beschäftigten je Betrieb betrug dabei nur etwa zehn Personen [58]. Diese strukturellen Gegebenheiten wirken sich direkt auf den jeweiligen Anteil aus, den Unternehmen in neue Technologien investieren können. Standards lassen sich von einem einzelnen Unternehmen wesentlich schwieriger durchsetzen als in anderen Industriezweigen.

Vor dem Hintergrund dieser Gegebenheiten speziell im Bauwesen diskutiert der folgende Abschnitt Aspekte einer rechnergestützten Zusammenarbeit. Dabei kommt gerade einer Kooperation über Fachbereichsgrenzen hinweg eine besondere Bedeutung zu, wenn insgesamt eine höhere Effizienz und Qualität in der Planung von Bauwerken erreicht werden soll.

## 1.2 Computergestützte Zusammenarbeit

Allgemein wird das interdisziplinäre Forschungsfeld der rechnergestützten Gruppenarbeit, mit dem englischen Begriff „*computer supported cooperative work*“, kurz *CSCW* bezeichnet. Die reale Umsetzung der theoretischen Konzepte des *CSCW* in Hard- und Software bezeichnet man wiederum als *Groupware*. ELLIS et al. [47] definieren *Groupware* als: „*computer-based systems, that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment.*“ Die Autoren heben hervor, dass *Groupware*-Systeme im Wesentlichen drei Gruppenprozesse unterstützen müssen: *Kommunikation*, *Koordination* und *Kollaboration*. Diese Klassifizierung nach dem so genannten *3K-Modell* findet sich auch in anderen Arbeiten [14], wobei *Kollaboration* oft dem Begriff *Kooperation* gleichgesetzt wird. Die Intensität der Zusammenarbeit nimmt dabei von *Kommunikation* bis *Kollaboration* hin zu. Während bei Ersterem nur ein reiner Informationsaustausch im Vordergrund steht, setzt eine *Kooperation* bzw. eine *Kollaboration* ein gemeinsames Datenmaterial sowie ein gemeinsames Ziel aller Beteiligten voraus [19].

### 1.2.1 Kommunikation, Koordination, Kollaboration

Kommunikation in Form eines schnellen und hinreichend verlässlichen Austauschs an Informationen ist sicherlich die Grundvoraussetzung für eine kooperative Arbeit. Kommunikation kann dabei synchron, beispielsweise am Telefon oder asynchron wie etwa beim Versenden einer E-Mail stattfinden. Für ELLIS et al. [47] stellt jedoch die Tatsache, dass üblicherweise je nach Art der Kommunikation unterschiedliche, voneinander unabhängige Systeme verwendet werden, ein Hindernis für eine barrierefreie und effektive Zusammenarbeit dar. Ziel sollte es vielmehr sein, ein einziges Werkzeug zu haben, mit dem sich verschiedene Kommunikationswege gleichzeitig bedienen lassen.

Um die konkurrierenden Aktivitäten von Gruppenmitgliedern in der Zusammenarbeit aufeinander abzustimmen, ist eine Koordination der Arbeitsprozesse sinnvoll. Zur Beschreibung von Arbeitsabläufen werden hierzu spezielle Prozessmodelle angewandt. Dabei sind nach RÜPPEL und KLAUER [107] formale Methoden mit einer eindeutigen Syntax und eindeutigen Semantik, wie etwa Petri-Netze oder Workflow-Graphen, semi-formalen (z.B. Netzplan, UML) oder nicht-formalen Methoden (menschliche Sprache) überlegen. Die Verwendung formaler Methoden zur Modellierung von Arbeitsabläufen erlaubt eine detaillierte Analyse sowie eine Simulation des Prozessablaufs, um etwa Konflikte zu erkennen oder die Einteilung von Ressourcen zu organisieren. Basierend auf einem einmal definierten Prozessmodell lassen sich zudem während der Ausführung die jeweiligen Aktivitäten rechnergestützt steuern und kontrollieren. Softwaresysteme, mit denen sich auf diese Weise Arbeitsprozesse definieren, erzeugen und deren Ausführung steuern lassen, werden als *Workflow-Management-Systeme* bezeichnet. Derartige Systeme werden bevorzugt für Aufgaben mit strukturierten Abläufen eingesetzt, in denen gleichartige Vorgänge häufiger auftreten und die Arbeitsprozesse meist vorhersagbar sind. Im Bauwesen ist daher eine Modellierung von Prozessen mit Schwierigkeiten verbunden. Die Arbeitsprozesse sind in der Regel stark dynamisch und schwach strukturiert. Sie hängen zudem extrem von äußeren Einflüssen ab. Eine explizite Modellierung und Steuerung von Prozessabläufen in der Planung von Bauwerken ist daher nach WILLENBACHER nicht sinnvoll [130]. Um diesem Problem begegnen zu können, um auch dynamische Bauabläufe – zumindest teilweise – steuern zu können, schlagen RÜPPEL und KLAUER vor, das Prozessmodell so

zu gestalten, dass auch noch während einer Ausführung so genannte *Ad-hoc*-Modifikationen vorgenommen werden können.

Die Begriffe Kollaboration bzw. Kooperation bilden die dritte Dimension des 3K-Modells zur Klassifizierung von Gruppenprozessen. Kollaboration setzt dabei voraus, dass ein *gemeinsames Arbeitsmaterial* vorhanden sein muss [19]. ELLIS et al. [47] schreiben hierzu: „*Effective collaboration demands, that people share information.*“ Sie fügen hinzu, dass der Zugriff auf die gemeinsamen Informationen immer unter Berücksichtigung aller Teilnehmer erfolgen muss. Die Benutzer müssen stets über die Aktivitäten der anderen Teilnehmer im Bilde sein. Dieser Sachverhalt wird gemeinhin auch als *Gruppenbewusstsein* oder *Group-Awareness* bezeichnet. Kollaboration geht damit über einen reinen Datenaustausch hinaus. Mit der Benutzung eines gemeinsamen Datenbestandes tritt weiterhin die Frage der Konsistenthaltung in den Vordergrund. Solange ein Benutzer nur seinen eigenen Datensatz zu verwalten hat, sind inkonsistente Zustände in der Regel offensichtlich und können leicht lokal behoben werden. In dem Moment, in dem für einen einzelnen Bearbeiter ein vollständiger Überblick über den gesamten Datenbestand aber nicht mehr gegeben ist, müssen technische Hilfsmittel bereitgestellt werden, die eine Sicherung der Konsistenz unterstützen. In heterogenen Arbeitsumgebungen ist eine vollständige Konsistenthaltung allerdings nur schwer zu realisieren. Konflikte oder Inkonsistenzen, die vor allem verschiedene Fachbereiche betreffen, lassen sich oft nur schwer erkennen. Es sind daher immer Werkzeuge erforderlich, die neben einer Erkennung auch eine manuelle Auflösung von Konflikten unterstützen.

### 1.2.2 Standardisierte Bauwerksmodelle

Um im Sinne einer Kollaboration Daten gemeinsam nutzbar zu machen, müssen sie von allen Beteiligten auch (digital) interpretiert werden können. Neben der rein technischen Realisierung eines verteilten Datenzugriffs ist es also notwendig, zunächst eine gemeinsame „Sprache“ zu finden. Die beteiligten Computeranwendungen müssen in der Lage sein, zumindest die für sie relevanten Informationen zu verstehen und gegebenenfalls weiterverarbeiten zu können. Mit der Definition einer gemeinsamen Datenbeschreibung lassen sich dann Redundanzen vermeiden und die Anzahl an notwendigen Schnittstellen verringern. Die prinzipiell  $m \cdot n$  Schnittstellen zwischen individuellen Softwaresystemen lassen sich durch Nutzung einer gemeinsamen Datengrundlage auf  $m + n$  Schnittstellen reduzieren [100]. Die Kopplung von Anwendungen über ein gemeinsames Datenmodell wird gemeinhin als *Datenintegration* bezeichnet.

Zur Datenintegration im Bauwesen werden digitale Bauwerksmodelle entwickelt. Das Ziel von Bauwerksmodellen – oder im Allgemeinen *Produktmodellen* – liegt in der einheitlichen Beschreibung für alle Daten, die während der Lebensphase eines Produkts von der Entstehung bis hin zur Entsorgung anfallen können. Ein Produktmodell bildet damit die Grundlage für einen durchgängigen Informationsfluss und ermöglicht so eine fachübergreifende Zusammenarbeit im Planungsprozess. Zur Reduktion der aus der Vielzahl an Teilmodellen gegebenen Komplexität wird in Produktmodellen meist ein hybrider Modellansatz verfolgt. Bis zu einer bestimmten Ebene wird für alle Fachrichtungen ein gemeinsames Modell definiert. Darüber hinaus wird dieses fachspezifisch um so genannte *Partialmodelle* ergänzt. Die zentrale Modellkomponente soll dabei prozessübergreifend Daten bereitstellen, während fachgebundene Informationen in den Partialmodellen vorgehalten werden. Dieser Ansatz findet sich beispielsweise in den *Industry Foundation Classes* (IFC), die im Jahre 1995 von einer in den USA eigens hierfür ge-

gründeten Vereinigung, der *International Alliance of Interoperability*<sup>1</sup> (IAI) entworfen wurde, um einen herstellerneutralen Standard für den Austausch von Bauwerksinformationen festzulegen [70]. Das objektorientierte Modell der IFC enthält dabei Klassen zur Beschreibung von Gebäudestruktur, Räumen, Bauelementen sowie deren gegenseitigen Abhängigkeiten und deren Attribute. Ziel der IFC ist es, ein Bauwerk nur bis zu einem bestimmten Detaillierungsgrad zu beschreiben. Zur weiteren Detaillierung der Objekte stehen dann innerhalb der IFC so genannte Eigenschaftendatensätze zur Verfügung, die in beliebiger Weise verknüpft und den Bauteilen zugeordnet werden können [71]. Die IFC sind im Rahmen der ISO<sup>2</sup>-Norm 16739 als Industriestandard festgelegt. Neben dem objektorientierten Bauwerksmodell sind darin auch Definitionen für ein Datenbank- und Dateiaustauschformat spezifiziert.

Die Bestimmung eines gemeinsamen Produktmodells erweist sich jedoch gerade im Bauentwicklungsprozess als außerordentlich schwierig. Aufgrund der Vielzahl an Fachdisziplinen existieren eine Menge spezialisierter Teilmodelle mit zum Teil rein fachspezifischer Semantik. Für einen Architekten stellt ein stabförmiges Objekt wie eine Stütze etwas ganz anderes dar, als etwa für einen Statiker oder einen Kostenplaner. Mit jeder Erweiterung des gemeinsamen Bauwerksmodells um fachspezifische Teilaspekte nimmt die Komplexität zu, was sich dann besonders im Aufwand zur Sicherstellung der Konsistenz als auch zur Wartung beteiligter Fachapplikationen bemerkbar macht. Der oben genannte Partialmodellansatz birgt auch nach wie vor die Problematik, dass Daten redundant beschrieben werden und ineinander überführt werden müssen.

### 1.3 Tragwerksplanung im konstruktiven Ingenieurbau

Die Aufgabe in der Tragwerksplanung besteht darin, ein Tragwerk so zu konzipieren, dass dieses äußeren Einwirkungen während seiner gesamten Nutzungsdauer standhält sowie den wirtschaftlichen und technischen Anforderungen gerecht wird. Die Tragwerksplanung als Teil des generellen Entwurfsprozesses eines Bauwerkes gliedert sich in einzelne Arbeitsschritte - Vorentwurf/Vorbemessung, strukturanalytische Berechnung, Nachweis/Bemessung sowie konstruktive Durchbildung [57]. Als Resultat der Tragwerksplanung werden Positions-, Schal- und Bewehrungspläne erzeugt, die weitgehend in Papierform an die Bauausführung weitergegeben werden. Es ist dabei durchaus üblich, dass einzelne Arbeitsschritte von unterschiedlichen Planungsgruppen an verschiedenen Orten durchgeführt werden. Gerade die Erstellung der konstruktiven Ausführungspläne, die einen Großteil der Aktivitäten ausmachen, werden in der Praxis oft unabhängig von Berechnung und Bemessung an externe Subunternehmer ausgelagert. Die Arbeitsschritte in der Tragwerksplanung sind jedoch in hohem Maße voneinander abhängig und werden in der Regel mehrmals durchlaufen. Eine Trennung von Statik und Konstruktion in dieser Form kann daher einen effektiven Einsatz von computergestützten Planungsmethoden behindern und sollte nach BEUKE „für einen reibungslosen Integrationsablauf überdacht werden“ [12].

Eine weitere Differenzierung der einzelnen Arbeitsprozesse in der Tragwerksplanung nehmen MEISSNER und RÜPPEL et al. [108, 75] in ihren Arbeiten vor (vgl. dazu auch [132]). Abbildung 1.1 zeigt hierzu die von den Autoren identifizierten Teilprozesse zusammen mit

<sup>1</sup>IAI – Industrieallianz für Interoperabilität e.V., <http://www.buildingsmart.de>

<sup>2</sup>ISO - International Organization for Standardization: <http://www.iso.org/>

den eingesetzten Softwaresystemen sowie den verwendeten Datenmodellen und gegenseitigen Abhängigkeiten.

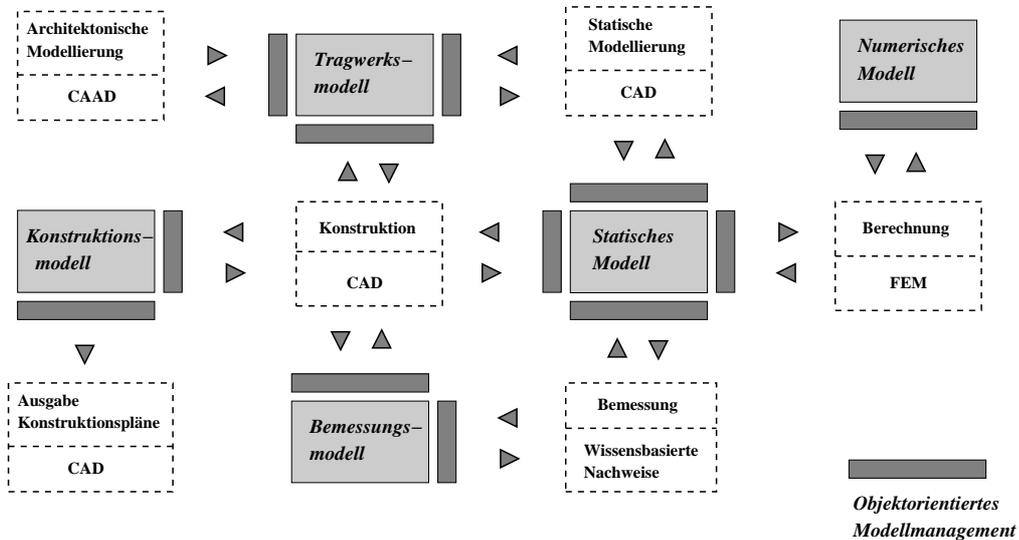


Abbildung 1.1: Teilprozesse und Teilproduktmodelle in der Tragwerksplanung nach RÜPPEL und MEISSNER [108].

Im Detail lassen sich folgende Modelle in der Tragwerksplanung unterscheiden:

- **Architekturmodell:** Das Architekturmodell ist das Ergebnis der ersten großen Planungsphase und bildet gewissermaßen den Ausgangspunkt für die weiteren Schritte der Tragwerksplanung. Der Architekt beschreibt das Bauwerk im Allgemeinen als dreidimensionales Modell, in welchem die Bauteile als „Körper“, d.h. als Volumenstrukturen geometrisch dargestellt werden. Neben den geometrischen Daten beinhaltet das Architekturmodell weitere semantische Bauwerksinformationen. Der Architekt erstellt idealerweise sein Modell in einem 3D-CAD System und wird es unter Umständen in Form eines standardisierten Produktdatenmodells (IFC) an den Tragwerksplaner übergeben.
- **Tragwerksmodell:** Die tragende Struktur eines Bauwerks ist prinzipiell eine Untermenge des Architekturmodells und wird durch das Tragwerksmodell repräsentiert. Der Ingenieur identifiziert tragende Elemente und übernimmt diese idealerweise aus dem Architekturmodell. Es ist in der Praxis aber durchaus üblich, dass der Tragwerksplaner sein Modell mit eigenen CAD-Werkzeugen neu entwickelt.
- **Statisches Modell:** Das statische Modell stellt eine Abstraktion des Tragwerksmodells dar. Je nach Tragverhalten werden die Objekte des statischen Modells durch mechanische Systeme unterschiedlicher Dimension beschrieben. Typische Systeme im Hochbau sind Platten, Scheiben, Rahmen, Stützen, Balken, Federelemente sowie massive Strukturen. Neben der Systemgeometrie werden an dieser Stelle Querschnittswerte, Lagerungsbedingungen, angreifende Lasten sowie Assoziationen zu anderen Bauteilen modelliert.
- **Numerisches Modell:** Zur statischen Analyse des Tragsystems mittels mathematisch-numerischer Verfahren wird das statische Modell in ein numerisches Modell überführt.

Zur Berechnung wird in der Regel die Methode der finiten Elemente verwendet. Dabei können je nach statischem System unterschiedliche Finite-Elemente-Programme zum Einsatz kommen. Nach der Analyse sind die auf die Systemachsen bzw. Systemflächen bezogenen Verschiebungs- und Schnittgrößen bekannt.

- **Bemessungsmodell:** Zur Durchführung der normativen Nachweise werden Bemessungsmodelle erzeugt. In diese fließen vorab angenommene Querschnittsabmessungen, Materialparameter sowie die in der Analyse ermittelten Schnittgrößen ein. Je nach statischem System werden unterschiedliche Nachweisverfahren angewandt. Dabei sind zum einen die Annahmen zu berücksichtigen, die den Bemessungsverfahren zugrunde liegen (z.B. Ebenbleiben der Querschnitte), zum anderen die Modellannahmen der statischen Berechnung. Bei der Beurteilung der Ergebnisse sind immer auch die tatsächlichen Verhältnisse am dreidimensionalen Bauwerksmodell mit einzubeziehen.
- **Modell der Gebrauchstauglichkeit:** In [75, 132] wird zusätzlich ein Modell der Gebrauchstauglichkeit definiert. Im Stahlbetonbau sind neben der Sicherstellung der Tragfähigkeit Nachweise erforderlich, die eine Gebrauchstauglichkeit beziehungsweise Dauerhaftigkeit der Bauteile gewährleisten sollen. Die Verfahren unterscheiden sich weitgehend von den Methoden des Bemessungsmodells und sind auch je nach Bauteiltyp verschieden.
- **Konstruktionsmodell:** Auf Grundlage der in der Bemessung und den Nachweisen zur Gebrauchstauglichkeit ermittelten Daten können schließlich Vorgaben für die konstruktive Durchbildung der Bauteile getroffen werden. Hierfür werden Konstruktionsmodelle aufgebaut, in denen Informationen zur Schalung und Bewehrung (Bewehrungspläne, Bügelabstände, Betondeckung etc.) verwaltet werden. Idealerweise wird hierzu ein Volumenmodell verwendet.
- **Modell der Computergrafik:** In Erweiterung zu den Klassifizierungen in [75, 108, 132] sollen an dieser Stelle auch Modelle zur Visualisierung von Geometrie und Daten berücksichtigt werden. In allen Phasen der Tragwerksplanung sind Berechnungsergebnisse und Konstruktionsangaben grafisch darzustellen. Gerade um Verständnis für das Tragverhalten zu entwickeln und zur Prüfung der Ergebnisse auf Plausibilität, ist eine Visualisierung, möglichst am Gesamtsystem, wichtig für den Ingenieur. Während eine Darstellung von Schnittgrößen in der Regel am dimensionsreduzierten Modell einzelner Bauteile erfolgt, werden Verschiebungen auch am Volumenmodell des Gesamtsystems abgebildet.

Zur Integration all dieser Teilmodelle in den Prozess der Tragwerksplanung sind Konzepte zu entwickeln, die einerseits eine flexible Einbindung (bestehender) Softwaresysteme erlauben, andererseits die Komplexität nicht zu stark ansteigen lassen. Hierzu schlagen oben genannte Autoren in [75, 132] vor, durchgängig Konzepte der objektorientierten Modellierung zu verwenden. Durch die in objektorientierten Programmen gegebene Modularität, bessere Lesbarkeit und Verständlichkeit ist es leichter, die in der Tragwerksplanung eingesetzten Softwaresysteme erweitern, warten und wiederverwenden zu können. Um eine gleichzeitige Bearbeitung eines Projekts durch mehrere Fachplaner zu erlauben und dabei die Konsistenz gemeinsamer Daten sichern zu können, favorisieren die Autoren zudem eine Integration über ein zentrales Datenbankmanagementsystem gegenüber einem reinen dateibasierten Austausch.

Eine ähnliche Systemarchitektur mit einer zentralen, netzwerkfähigen Projektdatenbank, die eine gleichzeitige Bearbeitung von Fachplanern erlaubt, wird auch von RANK et al. [98]

vorgestellt. Wichtig ist dabei laut RANK et al. eine strikte Trennung von Strukturmodell und numerischem Modell. Indem alle berechnungsrelevanten Daten ausschließlich netzunabhängig auf dem Strukturmodell formuliert werden, gehen diese bei einer Neuvernetzung des Berechnungsmodells nicht verloren, so dass Modifikationen der Tragstruktur mit einem Minimum an zusätzlichem Bearbeitungsaufwand untersucht werden können.

Es bleibt jedoch festzuhalten, dass allein in der Tragwerksplanung eine Modellvielfalt existiert, welche die Komplexität und den Aufwand zur Integration einzelner Teilprozesse erheblich erhöht. Betrachtet man die Abhängigkeiten in Abbildung 1.1, so wird deutlich, dass zwischen den einzelnen Arbeitsprozessen insgesamt zwölf Schnittstellen zu berücksichtigen sind. Hinzu kommen unterschiedliche Varianten der Modellbeschreibung, die aus der Dimensionsreduzierung resultieren. Der Schritt der Dimensionsreduzierung beim Übergang vom Architekturmodell zum statischen Modell lässt sich dabei nur schwer automatisieren. Bei der Modellierung muss für jedes Bauteil entschieden werden, welches Tragverhalten vorliegt und die entsprechende (dimensionsreduzierte) Struktur ausgewählt werden. Hierzu können für geometrisch einfache Körper durchaus automatische Verfahren gefunden werden. Bei komplizierten Strukturen mit wechselnden Querschnitten oder mit Öffnungen gestaltet sich dies jedoch schwieriger. Ob beispielsweise ein Plattenbalken als Faltwerk modelliert werden muss oder allein durch seine Systemlinie beschrieben werden kann, muss in der Regel vom Ingenieur beurteilt werden. Weiterhin sind Kopplungen zwischen den Tragstrukturen zu modellieren. Welche Art von Kopplung dabei vorliegt, hängt von den Auflagerbedingungen, den geometrischen Gegebenheiten am dreidimensionalen System sowie von eventuell vorhandenen Auflasten ab. Weil zudem die Systemlinien der Elemente nicht immer zusammenfallen, sind die daraus entstehenden Exzentrizitäten bei einer Modellierung in geeigneter Weise zu berücksichtigen.

Die Dimensionsreduzierung stellt damit ein wesentliches Hindernis in der Kooperation zwischen Architekt und Tragwerksplaner dar. Während sich die weiteren Arbeitsschritte durchaus automatisieren lassen ist für den Transfer vom Architekturmodell zum statischen Modell ein durchgängiger Datenaustausch derzeit noch nicht möglich. WÖRNER et al. [132] schreiben hierzu: *„Eine computergestützte automatische Generierung ingenieurpraktischer Tragwerksmodelle und statischer Modelle ist mit dem vorhandenen Wissen nicht möglich. Es bedarf zukünftiger Forschung, um ein Verfahren zur Bildung von Tragsystemen zu erarbeiten. Dieses Verfahren schließt die noch vorhandene Lücke zwischen Architekturmodell und Tragwerksmodell für eine durchgängige computergestützte Planung.“*

## 1.4 Volumenmodellierung als Basis der Integration

Inwieweit lässt sich die Zusammenarbeit im Bauplanungsprozess – und hier insbesondere in der Tragwerksplanung – verbessern, wenn eine Modellierung und Berechnung auf einheitlich dreidimensionale Volumenstrukturen gegründet wird? Gerade der Übergang vom Architekturmodell zu den dimensionsreduzierten Tragwerksstrukturen stellt einen deutlichen Schnitt im Modelltransfer dar. Die unterschiedliche Semantik der Modelle bedingt, dass eine eindeutige Transformation von einer Darstellung in die andere in der Regel nicht gegeben ist. Ein konkretes (Volumen-)Bauteil im Architekturmodell kann durch unterschiedliche statische Systeme repräsentiert werden. Last- und Verschiebungsrandbedingungen sind je nach Art des Systems eigens zu bestimmen und können damit nicht einheitlich am Ausgangsmodell definiert werden. Es sind vor allem auch dann Schwierigkeiten zu erwarten, wenn Daten (z.B. Berechnungser-

gebnisse) zurück übertragen werden sollen, um sie etwa anderen Planungsprozessen wieder zur Verfügung zu stellen. Nicht zuletzt bildet ein Volumenmodell die tatsächliche Geometrie eines Bauteils realistischer ab. Geometrische Effekte, wie Einspannbedingungen oder kombinierte Tragwirkungen lassen sich im Dreidimensionalen realistischer erfassen.

Das Vorgehen, alle Tragstrukturen einheitlich mittels Volumenelementen zu berechnen, steht jedoch in erheblichem Gegensatz zur derzeit üblichen Praxis. Ein wesentlicher Grund für die Verwendung dimensionsreduzierter Modelle liegt sicherlich an den exorbitant hohen Rechenzeiten, die mit den heutzutage üblicherweise verwendeten numerischen Methoden für die Analyse eines komplexen Volumenmodells notwendig wären. Es treten auch Fragen der geometrischen Modellierung wesentlich stärker in den Vordergrund, wenn alle Bauelemente strikt mittels Volumenstrukturen beschrieben werden sollen. Diese numerisch und geometrisch höheren Anforderungen zusammen mit den in der technischen Mechanik zum Teil geschichtlich gewachsenen Gepflogenheiten haben dazu beigetragen, dass Bauwerke bevorzugt mittels dimensionsreduzierter Strukturen berechnet werden. Entwicklungen im Bereich des CAD (Computer Aided Design) im Bauwesen wie in anderen Industriebereichen zeigen jedoch, dass computergestützte Planungstätigkeiten zunehmend an dreidimensionalen Volumenmodellen im Sinne eines „*Solid Model*“ durchgeführt werden.

Die im Folgenden betrachtete strikt volumenorientierte Vorgehensweise war der Schwerpunkt eines von der Deutschen Forschungsgemeinschaft geförderten Projekts [99], in dessen Rahmen auch die vorliegende Arbeit entstanden ist. Das Projekt ging dabei von folgender Hypothese aus: Durch Steigerung der Rechen- und Speicherleistung von Arbeitsplatzrechnern sowie neue Konzepte der geometrischen Modellierung und der numerischen Berechnungsverfahren kann in wenigen Jahren ein erheblicher Teil der computergestützten Planung an streng volumenorientierten Modellen durchgeführt werden. Es wird angenommen, dass die oben genannten, dominierenden Hindernisse in naher Zukunft wegfallen werden und damit eine durchgängige Tragwerksanalyse am Volumenmodell möglich sein wird. Ein derartiges Vorgehen stellt jedoch eine Veränderung dar, die sich nicht allein auf die Art der computergestützten Zusammenarbeit, sondern mehr oder minder auf alle Bereiche der konstruktiven Tragwerksplanung auswirkt. Es treten Fragestellungen in den Vordergrund, die etwa eine realistische Modellierung von Anschlüssen und Auflagern, von Stahl im Verbund oder die Problematik der Bemessung von Bauteilen betreffen. Ziel des oben genannten Forschungsprojekts war es, einige dieser Fragen genauer zu betrachten und Konzepte aufzuzeigen, die eine strikt volumenorientierte Herangehensweise ermöglichen. Folgende Aufzählung stellt einige davon vor, die auch innerhalb des im Jahre 1999 eingereichten Forschungsantrages diskutiert wurden:

- Welche Methoden sind geeignet, um aus einem räumlichen Architekturmodell weitgehend automatisch ein dreidimensionales numerisches Modell auf der Grundlage eines Finite-Elemente-Netzes zu generieren?
- Welche Simulationsmethoden sind geeignet, die typischerweise dünnen Strukturen im Bauingenieurwesen als Volumenmodell, jedoch mit vertretbarem Rechenaufwand und hoher Genauigkeit zu analysieren?
- Welche Visualisierungsmethoden lassen sich anwenden, um bauingenieurspezifische Fragestellungen am Volumenmodell darzustellen?
- Welche Eigenschaften muss ein für den Transfer vom CAD-Modell des Architekten zum räumlichen Strukturmodell des Tragwerksplaners geeignetes Objektmodell aufweisen?

- Auch beim Übergang vom räumlichen Architekturmodell zum volumenorientierten Tragwerks- und Berechnungsmodell sind Abstraktionsschritte notwendig, weil es nicht sinnvoll ist, jedes architektonische Detail im Tragwerksmodell abzubilden. Wie kann dieser Abstraktionsprozess unterstützt werden?
- Welches Softwaremodell ist geeignet, eine volumenorientierte komplexe Simulation aus überschaubaren Softwarebausteinen zusammenzusetzen?
- Wie kann eine verteilte Software- und Datenstruktur realisiert werden, die eine flexible Kooperation zwischen Architekt und Tragwerksplaner auf der Basis des räumlichen Modells erlaubt?
- Welche Methoden und Datenmodelle sind zu verwenden, um die im Planungsprozess ständig auftretenden Modifikationen transparent und effizient verwalten zu können?
- Wie kann der Baufortschritt effizient modelliert werden?
- Wie ist es möglich, die Ergebnisse einer räumlichen Berechnung den im Bauingenieurwesen üblichen Nachweisverfahren zugänglich zu machen, die meist auf dimensionsreduzierten Modellen beruhen? Können dazu die Berechnungsergebnisse *a posteriori* auf reduzierte Modelle übertragen werden?
- Wie kann eine konsistente Reduktion der dreidimensionalen Ergebnisdaten durchgeführt werden, um dem Bauingenieur eine Auswertung und Kontrolle der Berechnungsergebnisse zu ermöglichen?
- Können Nachweisverfahren entwickelt werden, die direkt auf dreidimensionalen Analyse-Ergebnissen aufbauen und für den alltäglichen Gebrauch geeignet sind?
- Wie kann das räumliche Modell im weiteren Planungs- und Konstruktionsprozess (von der Bewehrungsführung bis zur Bauüberwachung) genutzt werden?
- Wie können komplexe Struktureigenschaften, wie z.B. die Bewehrungsführung im Beton, berücksichtigt und unter Umständen in nichtlineare Berechnungsmethoden einbezogen werden?

Obige Liste erhebt natürlich keinen Anspruch auf Vollständigkeit. Es ist auch völlig klar, dass die Vielzahl an Problemstellungen im Rahmen eines Forschungsprojekts, oder gar innerhalb einer einzigen Doktorarbeit nicht umfassend beantwortet werden können. So soll auch nicht der Eindruck entstehen, dass mithilfe einer strikt volumenorientierten Vorgehensweise alle Probleme leichter zu lösen sind. Einige der oben genannten Punkte sind weiter zu diskutieren und erfordern sicherlich zukünftige Forschungsaktivitäten. Es wird aber erwartet, dass in der Summe die Vorteile einer strikt volumenorientierten Herangehensweise in der Planung von Tragwerken die Nachteile überwiegen werden.

## 1.5 Zielsetzung und Gliederung der Arbeit

Die vorliegende Arbeit greift die Hypothese der strikten Volumenorientierung auf und konzentriert sich auf einige der im vorherigen Abschnitt aufgeworfenen Fragestellungen. Im Rahmen des zuvor zitierten Forschungsprojekts entstanden weitere Arbeiten im Umfeld zu dieser Thematik, in welchen wichtige Grundlagen für die vorliegende Dissertation erarbeitet wurden. In seiner Arbeit „Gebäudemodell-basierte Strukturanalyse im Bauwesen“ stellt ROMBERG [103] einen Weg vor, wie aus einem standardisierten Bauwerksmodell ein strikt volumenorientiertes Finite-Elemente-Netz erzeugt werden kann. Das von ROMBERG vorgestellte Verfahren, welches in Kapitel 2 erläutert wird, macht damit die hier betrachtete Vorgehensweise überhaupt erst möglich. Weiterhin wurden von MUNDANI „Hierarchische Geometriemodelle zur Einbettung verteilter Simulationsaufgaben“ [79] erarbeitet. Seine Ausführungen stellen dann vor allem im Rahmen der Kapitel 3 und 4 eine wichtige Grundlage für die dort betrachteten Untersuchungen dar.

Ausgehend von der Motivation in der Einleitung soll in Kapitel 2 ein Weg aufgezeigt werden, wie aus einem Gebäudemodell eines Architekten ein volumenorientiertes Finite-Elemente-Netz erzeugt und berechnet werden kann. Hierzu werden neben den Arbeiten von ROMBERG zur Generierung von Hexaedernetzen vor allem die Untersuchungen zur Berechnung dünnwandiger Strukturen mittels Volumenelementen der  $p$ -Version der Finite-Elemente-Methode betrachtet, welche unter anderem auch am Lehrstuhl für Bauinformatik der Technischen Universität München entstanden sind. Am Beispiel eines mehrstöckigen Bürogebäudes sollen am Ende dieses Kapitels alle Schritte vom Gebäudemodell zur volumenorientierten Strukturanalyse nochmalig konkret dargestellt werden.

Auch mit modernen Rechnersystemen kann der Aufwand zur Berechnung volumenorientierter Strukturen durchaus erheblich sein, besonders wenn eine Analyse des Modells immer am Gesamtsystem durchgeführt werden soll. Es sind daher effiziente Konzepte zur Organisation der Modelle und der Berechnungsverfahren notwendig. In Kapitel 3 sollen hierzu hierarchische Datenmodelle besprochen und ein Weg vorgestellt werden, wie sich diese sowohl zur Organisation geometrischer Objekte im Raum als auch zur Organisation der strukturmechanischen Berechnung heranziehen lassen. Unter Anwendung der rekursiven Substrukturierung liegt dabei das wesentliche Ziel darin, die Hierarchie so zu nutzen, dass nach lokalen Modifikationen am Modell eine Neuberechnung wesentlich schneller durchgeführt werden kann.

Kapitel 4 beschäftigt sich mit Fragen der Kooperation und skizziert dazu ein Software-Rahmenwerk, welches ein hierarchisch organisiertes Volumenmodell des Gesamtsystems ins Zentrum stellt. Verschiedene Fachplaner können konkurrierend auf das zentrale Modell zugreifen und dieses bearbeiten. Weiterhin wird auf die Vorteile einer Einbettung gegenüber einer Integration von Softwareanwendungen hingewiesen. Es wird dabei beispielhaft gezeigt, wie durch Einbettung einer Finite-Elemente-Simulation unter Ausnutzung der zuvor besprochenen Konzepte der Hierarchie in interaktiver und effizienter Weise eine Berechnung von Bauphasen organisiert werden kann.

In Kapitel 5 wird die Frage diskutiert, in welcher Form sich die Ergebnisse einer volumenorientierten Strukturanalyse zur Bemessung von Stahlbetonbauteilen heranziehen lassen. Die in

dieser Arbeit verfolgte Grundidee besteht dabei darin, eine Dimensionsreduktion nicht mehr *vor*, sondern *nach* der Berechnung vorzunehmen. Hierzu wird ein Algorithmus vorgestellt, mit dem die dreidimensionalen Spannungen innerhalb eines Volumenbauteils schnell auf integrale Schnittgrößen reduziert werden können. Es soll dabei gezeigt werden, welche Vorteile sich bieten, wenn die Bestimmung der Schnittgrößen auf Basis der dreidimensionalen Ergebnisse erfolgt und eine Möglichkeit besprochen werden, wie auf Grundlage der Spannungsgrößen am Volumenbauteil die Grundannahme der Biegebemessung, das Ebenbleiben der Querschnitte überprüft werden kann. Zuletzt wird ein Verfahren vorgestellt, mit welchem sich für einen gegebenen Baukörper die Bauteilform und die für die Reduktion der Spannungsgrößen notwendige Systemachse automatisch bestimmen lassen.

Kapitel 6 fasst schließlich die Ergebnisse zusammen und diskutiert abschließend Vor- und Nachteile sowie noch offene Probleme, die sich ergeben, wenn eine Berechnung von Tragwerksstrukturen auf strikt volumenorientierte Modelle gestützt wird.



# Kapitel 2

## Tragwerkssimulation am Volumenmodell

Inhalt dieses Kapitels sind Methoden, mit denen eine strikt volumenorientierte Vorgehensweise in der Tragwerksplanung realisiert werden kann. Es soll dabei der gesamte Weg vom Architekturmodell bis hin zur Tragwerkssimulation betrachtet werden. In einem ersten Abschnitt wird hierzu ein Verfahren vorgestellt, welches erlaubt, typische Gebäudestrukturen einheitlich mithilfe von Volumenelementen vernetzen zu können. Die hierbei durchgeführte Zerlegung des geometrischen Modells stellt dabei den Ausgangspunkt zu weiteren Simulationsverfahren dar, was in dem Zusammenhang kurz angesprochen werden soll. In einem zweiten Unterpunkt stehen dann finite Elemente hoher Ordnung im Zentrum der Betrachtung. Die so genannte  $p$ -Version der Finite-Elemente-Methode bietet eine sehr effiziente Möglichkeit, die in Bauwerken vorkommenden massiven, dünnwandigen und stabförmigen Strukturen einheitlich mittels Volumenelementen zu diskretisieren. Unter Anwendung anisotroper Ansätze ist es möglich, auch sehr schlanke Strukturen strikt mit Volumenelementen zu modellieren und dennoch eine Effizienz zu erreichen, die mit reinen Schalenformulierungen vergleichbar ist. Das Kapitel schließt mit einem Beispiel, an welchem der Weg vom Architekturmodell zur volumenbasierten Analyse nochmals nachvollzogen werden soll.

### 2.1 Vom Gebäudemodell zum volumenorientierten Finite-Elemente-Netz

Im Folgenden werden Arbeiten betrachtet, die in erster Linie von ROMBERG [104, 103] und VAN TREECK [128, 125, 124, 126] am Lehrstuhl für Bauinformatik der Technischen Universität München durchgeführt wurden. Die Autoren beschäftigten sich unter anderem mit der Frage, wie sich auf Basis eines gemeinsamen Geometriemodells verschiedene Simulationsverfahren in einen computergestützten Bauplanungsprozess einbinden lassen. Grundlage der Arbeiten ist ein geometrisches Volumenmodell, von welchem in mehreren Zerlegungsschritten sowohl ein Finite-Elemente-Netz für die Tragwerksanalyse als auch ein Raummodell für eine Innenraumströmungs- und Behaglichkeitsanalyse abgeleitet wird. Das Geometriemodell kann dabei einerseits in einem CAD-System erzeugt, andererseits von einem standardisierten Bauwerksmodell der Industry Foundation Classes abgeleitet werden [104, 127]. Ausgehend von diesem einheitlichen Modell können so unterschiedliche „Sichten“ der Planungsarbeit berücksichtigt

und die Kooperation zwischen den Fachdisziplinen unterstützt werden. Abbildung 2.1 stellt hierzu schematisch den Weg vom Bauwerksmodell zu den Simulationsmodellen mit den dazwischenliegenden Zerlegungsschritten dar. Die Abbildung beschreibt zudem auch die wesentlichen Methoden, die während der Überführung eines Modells in ein anderes zu tätigen waren.

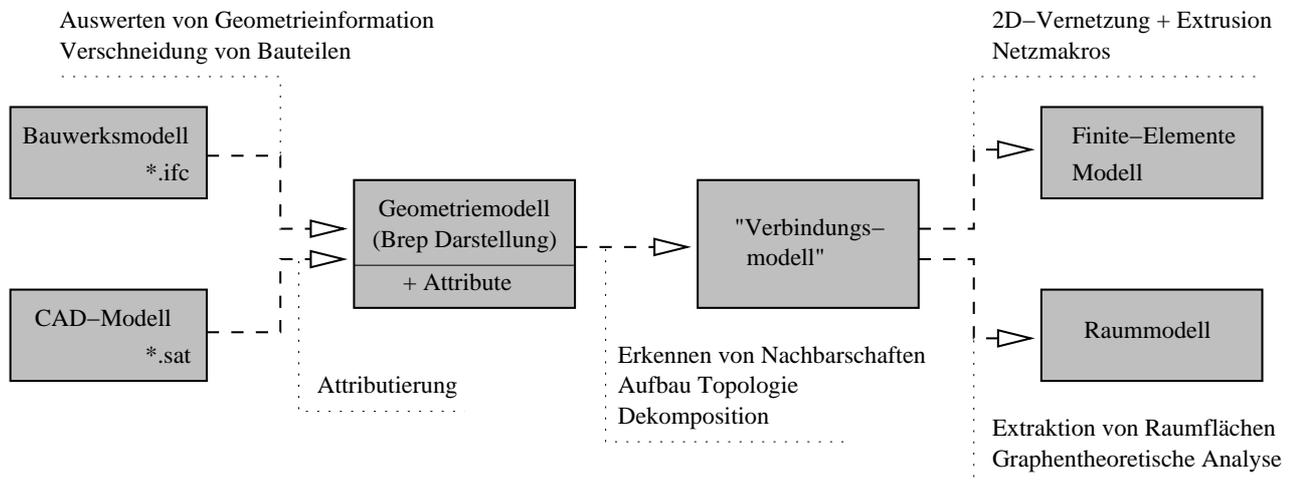


Abbildung 2.1: Weg der Modellableitung vom Bauwerksmodell zu volumenorientierten Simulationsmodellen nach [103, 124].

### 2.1.1 Ein erweitertes Geometriemodell als Grundlage

Voraussetzung für eine computergestützte Kooperation (oder: Kollaboration) ist ein gemeinsamer Datenbestand, der von allen Beteiligten genutzt werden kann. Zu diesem Zweck wurden von der International Alliance of Interoperability die Industry Foundation Classes, IFC [71] entworfen, um einen einheitlichen Standard für die digitale Beschreibung von Informationen zu schaffen, die während der gesamten Lebensdauer eines Bauwerks – von den ersten Planungsaufgaben bis zum Abbruch – anfallen können. Das objektorientierte Modell der IFC enthält unter anderem Klassen zur Beschreibung von Gebäudestruktur, Räumen, Bauelementen sowie deren gegenseitigen Abhängigkeiten und Attribute. Für die geometrische Beschreibung der Objekte innerhalb der IFC sind verschiedene Darstellungsarten möglich. Meist basieren diese auf *indirekten* Darstellungsschemata, etwa in Form von Produktionsmodellen (*Sweeping-Modell*), in denen eine Wand durch die Parameter Grundlinie, Dicke und Stockwerkshöhe beschrieben wird. Innerhalb der Klassenhierarchie der IFC wird dabei die Geometriebeschreibung nur als Attribut betrachtet, welches einem Bauwerksobjekt zugeordnet ist. Eine geometrische Darstellung ist damit nicht immer zwingend notwendig. In verschiedenen Planungsaufgaben spielt die exakte geometrische Form der Bauteile keine wesentliche Rolle, beziehungsweise sie ist zu dem Zeitpunkt der Planung noch gar nicht detailliert bekannt. Um die Bandbreite aller Planungsaufgaben unterstützen zu können ist es daher das Ziel der IFC, die Beschreibung von Objekten auch ohne detaillierte Geometrieinformation zu ermöglichen. Die Betrachtung der Geometrie als Attribut erlaubt zudem einen einfachen Wechsel der Darstellungsart, wenn etwa während des Planungsfortschritts die Form eines Bauteils durch eine genauere Darstellung ersetzt werden soll. Die IFC stellen auf diese Weise eine gute Grundlage zur Beschreibung

von Daten im Bauwesen dar, insbesondere was deren Möglichkeiten betrifft, eine Vielzahl von Fachdisziplinen zu vereinigen und die sukzessive Erweiterung des Modells während des Planungsfortschritts zu unterstützen.

Eine direkte Verwendung des Objektmodells im Rahmen von Simulationsaufgaben ist jedoch mit Schwierigkeiten verbunden. Für numerische Simulationen sind in der Regel andere Darstellungen von Geometrie Voraussetzung, etwa in Form von Oberflächenmodellen oder zellbasierter Aufzählungsschemata. Um Algorithmen effizient implementieren zu können, sind Darstellungsarten wichtig, die einen schnellen und direkten Zugriff auf den gesamten vom Modell eingenommenen Raum ermöglichen [118]. Auch für das Setzen von Randbedingungen ist es unter Umständen erforderlich, dass Kanten und Flächen unmittelbar zugänglich sind. Weiterhin werden für Simulationsaufgaben andere Anforderungen an die Genauigkeit gestellt. Aufgrund von Rundungsfehlern, aber auch, um eine robuste Durchführung von Operationen zu ermöglichen, werden in CAD-Systemen mehr oder minder große geometrische Toleranzen zugelassen [11]. Zur Anwendung in Simulationsverfahren sind derartige Toleranzen, die zu Überschneidungen oder Spalten zwischen den Körpern führen können, in der Regel aber nicht zulässig.

Aus diesem Grunde wird in den oben zitierten Arbeiten zur Integration verschiedener Simulationsaufgaben als zentrales Modell ein volumenorientiertes Geometriemodell in *Boundary-Representation (B-Rep)*-Darstellung verwendet. Mithilfe des geometrischen Modellierkernes *ACIS*<sup>1</sup>[24] wird dieses aus den Geometriedaten eines IFC-Modells abgeleitet. Dabei wird für jedes Bauteil ein eigenständiger geometrischer Körper erzeugt, wobei die Objekte in geeigneter Weise miteinander verschnitten werden. Nichtgeometrische Attribute aus dem Gebäudemodell werden parallel in einer relationalen Datenbank abgelegt und über eindeutige Objektschlüssel referenziert. Detaillierte Informationen zu dieser Transformation des Geometriemodells aus einem standardisierten Bauwerksmodell sind der Arbeit von ROMBERG [103] zu entnehmen. Ein von MUNDANI [79] vorgestelltes Verfahren – welches in dieser Arbeit im Rahmen hierarchischer Datenstrukturen unter Abschnitt 3.2.2.2 näher diskutiert werden soll – erlaubt zudem, die aus den Toleranzen entstehenden Spalten oder Überschneidungen schnell zu erkennen.

### 2.1.2 Zerlegung in ein Verbindungsmodell

In einem weiteren Schritt wird das geometrische Volumenmodell in ein so genanntes *Verbindungsmodell* überführt. Zur Erzeugung des Verbindungsmodells werden unter Anwendung boolescher Operationen und Ausnutzung der topologischen Abhängigkeiten die geometrischen Objekte in geeigneter Weise verschnitten und eine Zerlegung in so genannte *Kopplungs-* und *Differenzobjekte* vorgenommen. Abbildung 2.2 illustriert diesen Vorgang am Beispiel einer Raumecke. Jedes Kopplungs- und Differenzobjekt ist dabei ein abgeschlossener Körper in  $\mathbb{R}^3$ . Am Beispiel des Modells in Abbildung 2.2 gilt nach einer Zerlegung für die Objektmengen:

---

<sup>1</sup>ACIS geometric modeling kernel: <http://www.spatial.com>

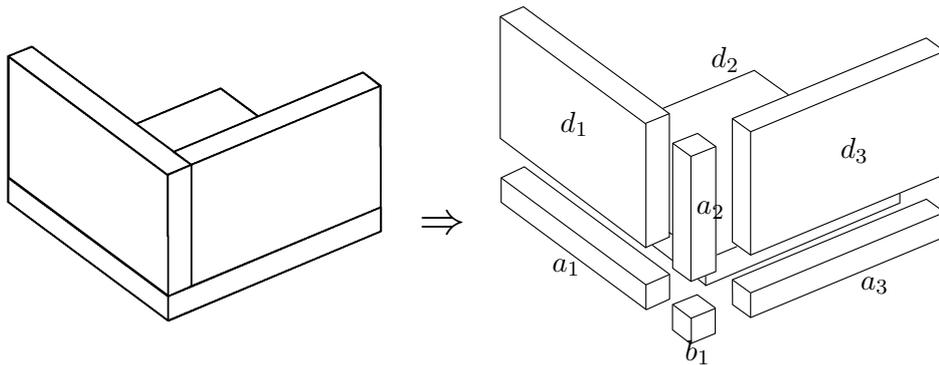


Abbildung 2.2: Zerlegung einer Raumecke in Kopplungs- und Differenzobjekte nach [103].

$$\begin{aligned}
 \text{Menge aller Differenzobjekte: } M_D &= \{d_1, d_2, d_3\} \\
 \text{Kopplungsobjekte Typ 1: } M_{K1} &= \{a_1, a_2, a_3\} \\
 \text{Kopplungsobjekte Typ 2: } M_{K2} &= \{b_1\} \\
 \text{Menge aller Kopplungsobjekte: } M_K &= M_{K1} \cup M_{K2} \\
 \text{Menge aller Verbindungsobjekte: } M_V &= M_K \cup M_D
 \end{aligned}$$

Die Menge der Kopplungsobjekte  $M_K$  kann weiter unterteilt werden in die Menge der *Kopplungsobjekte des Ausgangsmodells* (Typ 1)  $M_{K1}$  und die Menge der *Kopplungsobjekte von Kopplungsobjekten* (Typ 2)  $M_{K2}$ .

Dem Algorithmus der Zerlegung geht zunächst eine Analyse des geometrischen Bauwerkmodells voraus. Hierzu werden in einem ersten Schritt alle vollflächigen Nachbarschaften zwischen Bauteilen identifiziert. Mithilfe des Geometrikerns *ACIS* [24] werden an den Verbindungen *eindeutige* Kontaktflächen erzeugt, an denen dann mittels *Sweeping* oder *Lofing* die Kopplungsobjekte abgespalten werden. Der Zerlegungsalgorithmus wurde ursprünglich von ROMBERG im Rahmen des DFG-Projekts [99] entwickelt und ist unter anderem in [103, 124] ausführlich dargestellt. Es sei an dieser Stelle darauf hingewiesen, dass der Algorithmus gemeinsam mit dem im nächsten Abschnitt erläuterten Verfahren zur Netzerzeugung hauptsächlich für Strukturen konzipiert ist, die in einem baupraktischen Zusammenhang relevant sind, also für Modelle, die in erster Linie aus Scheiben, Platten, Stützen etc. bestehen.

### 2.1.2.1 Grundlage: topologische Beziehungen

Für das hier betrachtete Vorgehen stellt die topologische Struktur des Gebäudemodells eine zentrale Grundlage dar. Erst mit dem Wissen aller Nachbarschaftsbeziehungen lässt sich das geometrische Modell in das Verbindungsmodell überführen und weiterhin die automatische Erzeugung eines volumenorientierten Finite-Elemente-Netzes als auch eine Erkennung von Räumen realisieren. In den Arbeiten von ROMBERG und VAN TREECK werden hierzu verschiedene Typen von Relationen definiert, welche in Abbildung 2.3 dargestellt sind:

- Bauteilgraph  $G_B$ : Dieser bildet sich aus der Menge der *Verbindungsobjekte*  $M_V$  und Relationen  $R_{PC}$ , die durch eindeutige flächige Verbindungen zwischen den Objekten in  $M_V$  definiert sind.

- Raumflächengraph: Aus der Menge der Verbindungsobjekte  $M_V$  extrahiert VAN TREECK die Menge aller Bauteilflächen  $M_F$  und deren Nachbarschaftsbeziehungen  $R_{NF}$ , um daraus den Raumflächengraphen  $G_F$  abzuleiten. Durch Analyse von  $G_F$  lassen sich dann unter Anwendung graphentheoretischer Algorithmen die Menge der *kleinsten geschlossenen Volumenkörper*  $M_{AV}$  (d.h. sämtliche Innenraumvolumina) im Modell bestimmen [128, 124].
- Raumgraph: Ist die Menge der Innenraumvolumina bekannt, lassen sich diese zusammen mit den daran angrenzenden Bauteilen aus  $M_V$  innerhalb des Raumgraphen  $G_I$  organisieren, welcher Nachbarschaftsbeziehungen zwischen Bauteilen und Räumen beschreibt und beispielsweise als Grundlage der Zonierung für eine thermische Gebäudesimulation dienen kann [124].

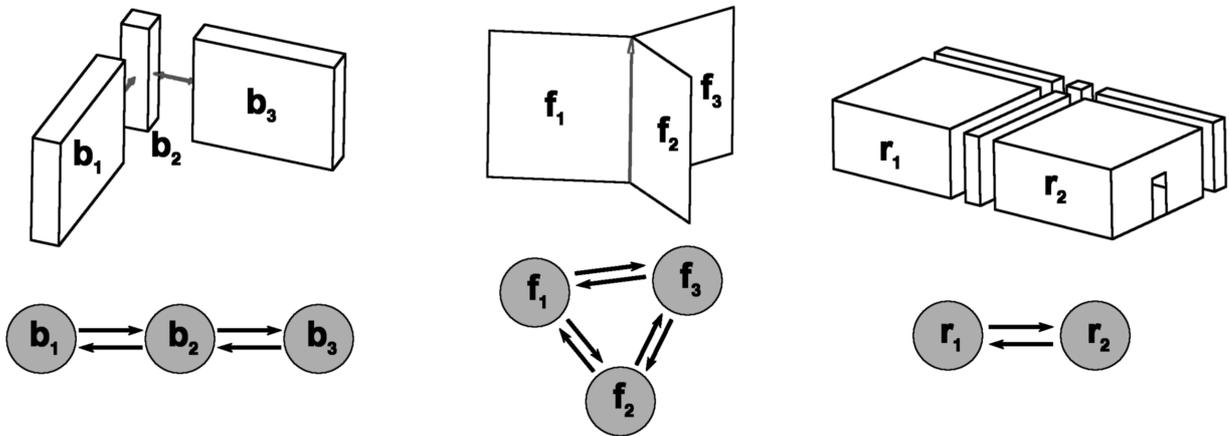


Abbildung 2.3: Bauteilgraph, Raumflächengraph, Raumgraph nach [124].

Darüber hinaus bietet sich der Bauteilgraph  $G_B$  auch als Ausgangspunkt für eine hierarchische Zerlegung in Substrukturen an. Eine Vorgehensweise wird hierzu in Kapitel 3 dieser Arbeit aufgezeigt. Dort wird die Nachbarschaftsinformation in einem Finite-Elemente-Netz genutzt, um mithilfe von Verfahren der Graphenpartitionierung das Berechnungsmodell so zu organisieren, dass unter Anwendung der rekursiven Substrukturierung insbesondere bei wiederholten Änderungen am Modell das entstehende Finite-Elemente Gleichungssystem sehr effizient gelöst werden kann. Die dort gezeigten Prinzipien der Zerlegung lassen sich aber in gleicher Weise auch direkt auf den Bauteilgraphen aus Abbildung 2.3 anwenden. Dies würde die zusätzliche Möglichkeit bieten, eine Partitionierung des Gebäudemodells schon *vor* einer Vernetzung der Struktur vorzunehmen, wodurch sich die aus der Hierarchie ergebenden Vorteile hinsichtlich einer Verteilung sowie der Organisation bei Modelländerungen auch auf diese Aufgaben anwenden ließen.

### 2.1.2.2 Vorteile

Nach erfolgter Zerlegung weist das Verbindungsmodell einige wichtige Eigenschaften auf, die besonders im Hinblick auf die folgende Vernetzung der Struktur von Bedeutung sind:

- Die geometrische Schnittmenge aller Differenzobjekte in  $M_D$  liefert nur gemeinsame Knoten und Kanten, aber keine gemeinsamen Flächen. Das bedeutet, Differenzobjekte sind immer nur zu Kopplungsobjekten vollflächig benachbart. Diese Eigenschaft wird während der Vernetzung ausgenutzt, indem eine konsistente Einteilung benachbarter Knoten nur auf den Kopplungsobjekten definiert werden muss und somit die Differenzkörper voneinander unabhängig vernetzt werden können.
- Für die hier betrachteten Bauwerksmodelle wird davon ausgegangen, dass Kopplungsobjekte eine „prismen“-artige Struktur (Hexaeder, Pentaeder) aufweisen. Durch Zerlegung lassen sich aus diesen finite Elemente erzeugen. Die nach der Abspaltung übrigen Differenzobjekte haben in der Regel eine „plattenartige“ Form, womit für diese mittels einer 2D-Vernetzung und Extrusion in die dritte Richtung ein Finite-Elemente-Netz erzeugt werden kann.
- Im Sinne der Strukturmechanik bildet das Verbindungsmodell in natürlicher Weise die Wechselwirkungen zwischen den Bauteilen ab. Es werden somit die wesentlichen Verbindungen zwischen den Tragwerkelementen sichtbar gemacht. Wie oben angesprochen kann zudem der Bauteilgraph des Verbindungsmodells als Ausgangspunkt für eine Strukturierung des Modells herangezogen werden.

### 2.1.3 Vernetzung der Struktur

Ausgangspunkt zur Erzeugung eines Finite-Elemente-Netzes ist das oben beschriebene Verbindungsmodell mit seinen Kopplungs- und Differenzobjekten. Die spezifische Struktur des Modells ist dabei von besonderer Bedeutung für das Vorgehen während der Vernetzung. Das Verfahren bedient sich einer Reihe von Schritten, in denen abhängig von der Bauteilform unterschiedliche Vorgehensweisen zur Anwendung kommen. Detailliertere Erläuterungen sind in [104, 103] zu finden. Das wesentliche Ziel des Algorithmus ist, ein kompatibles Netz für die *gesamte* Bauwerksstruktur zu erzeugen. Die Abbildungen 2.4 und 2.5 zeigen dazu die wesentlichen Schritte des Verfahrens.

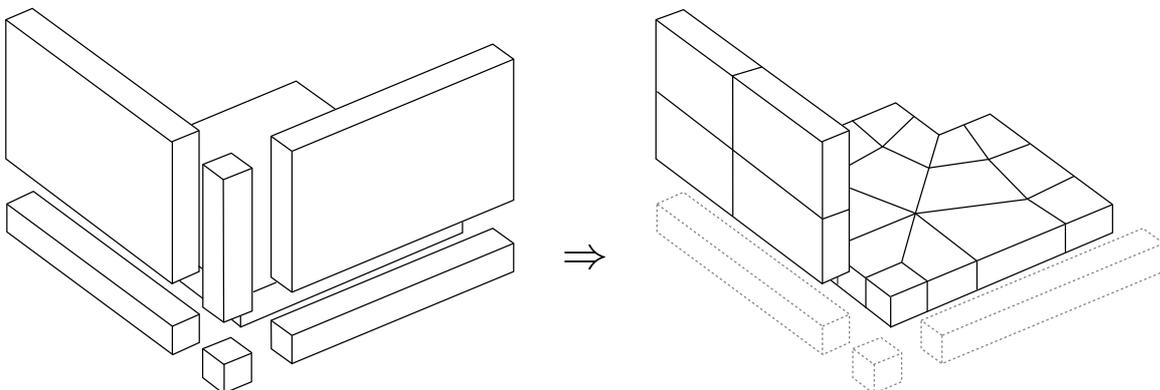


Abbildung 2.4: Links: Verbindungsmodell mit Kopplungs- und Differenzobjekten. Rechts: Vernetzung der Differenzobjekte unabhängig voneinander.

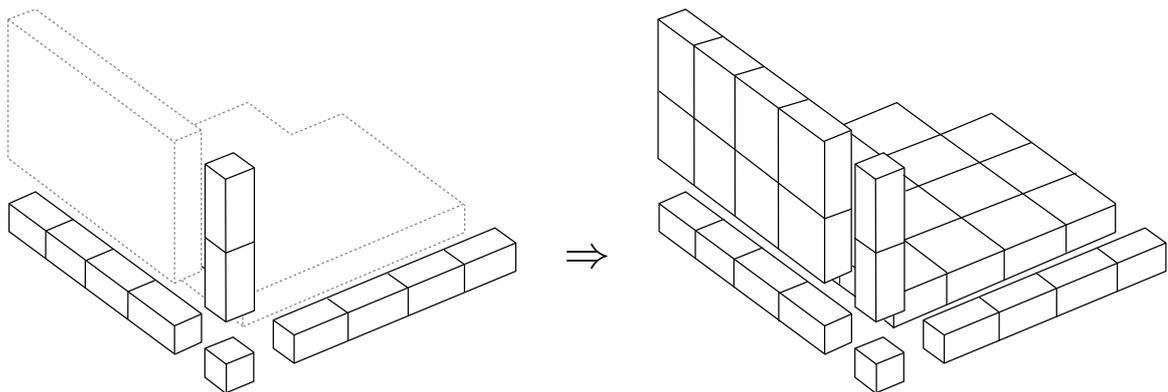


Abbildung 2.5: Links: Finden einer kompatiblen Einteilung auf den Kopplungsobjekten. Rechts: Zweiter Vernetzungsdurchlauf für die inneren Differenzobjekte unter Berücksichtigung der vorgegebenen Randdiskretisierung.

Ausgehend von einem Verbindungsmodell werden in einem ersten Schritt die Differenzobjekte zunächst unabhängig voneinander vernetzt. Dies geschieht für die in der Regel plattenartigen Strukturen mittels Freivernetzung der Mittelebene und Extrusion in die Dickenrichtung. Die nach der Vernetzung gegebene Verteilung von Knoten am Rande der Mittelfläche wird dann auf die benachbarten Kopplungsobjekte übertragen. Im Allgemeinen ist an einem stabförmigen Kopplungsobjekt die Verteilung der Knoten aller benachbarter Objekte jedoch nicht einheitlich. Für unterschiedliche Knotenverteilungen wird nun eine einheitliche Diskretisierung bestimmt und die Kopplungsobjekte selbst auf Basis dieser Diskretisierung vernetzt. Die so ermittelte Verteilung der Knoten wird dann auf alle angrenzenden Differenzobjekte wieder übertragen und diese schließlich unter Berücksichtigung der neuen Randdiskretisierung nochmalig neu vernetzt. Als Resultat ergibt sich ein kompatibles Finite-Elemente-Netz für die gesamte Bauwerksstruktur.

## 2.2 Die $p$ -Version der Finite-Elemente-Methode

Zur Berechnung von Bauwerksstrukturen auf Basis einer strikten Volumendarstellung kommt in dieser Arbeit die Finite-Elemente-Methode hoher Ordnung, die  $p$ -Version der FEM, zur Anwendung. Die Methode der finiten Elemente stellt zumindest in den Ingenieurwissenschaften sicherlich das am meisten verwendete Verfahren zur numerischen Lösung partieller Differentialgleichungen dar. Es ist dabei in der Literatur zu diesem Thema eine nicht zu überschauende Vielfalt zu finden. Es soll daher an dieser Stelle zunächst nur auf die Arbeit von SZABÓ und BABUŠKA [120] verwiesen werden, die als erste im Rahmen finiter Elemente hoher Ordnung die im Folgenden betrachteten hierarchischen Ansatzfunktionen vorgeschlagen haben.

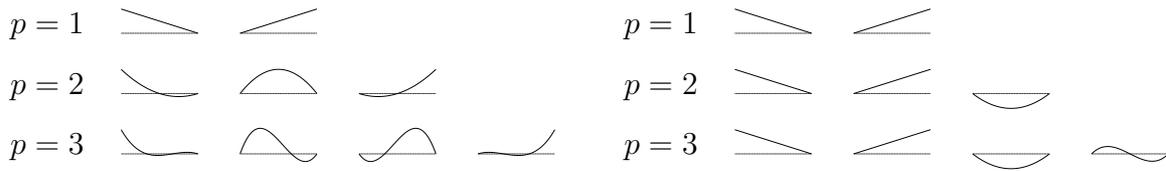
Im Unterschied zum klassischen Vorgehen mit linearen oder quadratischen Ansätzen der  $h$ -Version der FEM erfolgt bei Verwendung der  $p$ -Version eine Verbesserung der Genauigkeit nicht durch Verfeinerung des Netzes, sondern durch Erhöhung des Polynomgrades der elementweisen Ansatzfunktionen. Das Finite-Elemente-Netz ändert sich dabei nicht. Eine glatte Lösung vorausgesetzt, bieten Elemente der  $p$ -Version exponentielle Konvergenzraten des Fehlers in der Energienorm. Dadurch lässt sich bei gleicher Anzahl an Freiheitsgraden im Vergleich zur  $h$ -Version der FEM eine wesentlich höhere Genauigkeit erzielen. Bei geeigneter Wahl des Netzes lässt sich dies auch im Falle von Singularitäten (z.B. einspringende Ecke) erreichen (vgl. dazu [120, 121, 35]). Neben der hohen Genauigkeit ist zudem die Robustheit als Vorteil von Elementen hoher Ordnung zu nennen. Elemente der  $p$ -Version der FEM verhalten sich weitgehend robust gegenüber *Locking*, wenn ein gewählter niedriger Elementansatz bestimmte Nebenbedingungen eines Modells nicht ausreichend abbilden kann. Bei Verwendung höherer Polynomgrade können derartige Effekte wie Querschub-, Membranlocking oder Locking infolge inkompressiblen Materialverhaltens weitgehend vernachlässigt werden.

Von besonderem Interesse im Rahmen dieser Arbeit ist die Möglichkeit, gemischt massive und dünnwandige Strukturen einheitlich mittels dreidimensionaler Volumenelemente hoher Ordnung berechnen zu können. Aufgrund ihres robusten Verhaltens erlauben Elemente der  $p$ -Version sehr große Seitenverhältnisse mit bis zu einem Faktor von 1:1000. Weiterhin lässt sich innerhalb eines Elements der Polynomgrad in verschiedene Richtungen beliebig variieren und so an die Verhältnisse in Platten- oder Schalenstrukturen anpassen. Diese Möglichkeit der *anisotropen* Ansatzräume wurde erstmalig von SZABÓ und SAHRMANN [122] vorgeschlagen und von RANK et al. sowie DÜSTER et al. erweitert [35, 37, 43, 95, 94, 97] und dort sowohl für thermoelastische [41] als auch für Probleme der Elastoplastizität [40, 42] und Hyperelastizität [38, 39] untersucht.

### 2.2.1 Hierarchische Ansatzfunktionen

#### 2.2.1.1 Eindimensionale Ansatzfunktionen

Wie zu Beginn erwähnt, erzielt die  $p$ -Version der FEM eine Verbesserung der Genauigkeit durch Erhöhung des Polynomgrades der Ansatzfunktionen. Die von SZABÓ und BABUŠKA [120] hierzu eingeführten hierarchischen Ansatzfunktionen weisen im Vergleich zu höherpolynomialen Ansätzen der Standardbasis einen fundamentalen Unterschied auf. Dieser sei am Beispiel des eindimensionalen Falles in Abbildung 2.6 illustriert. Die Abbildung zeigt auf der linken Seite die klassischerweise verwendeten Knotenansatzfunktionen auf einem eindimensionalen Standardelement  $\Omega_{st} = [-1, 1]$ , welche durch LAGRANGE-Polynome definiert sind. Es

Abbildung 2.6: Eindimensionale Standard- und hierarchische Ansatzfunktionen für  $p = 1, 2, 3$ .

wird deutlich, dass für jede Ordnung ein eigener Satz von Polynomen verwendet werden muss.

Die Grundidee der hierarchischen Basis nach SZABÓ und BABUŠKA, die auf der rechten Seite der Abbildung 2.6 zu sehen sind, besteht darin, die Familie der Ansatzfunktionen so aufzubauen, dass bei Erhöhung des Polynomgrades alle Ansätze niedriger Ordnung in der Menge der Ansatzfunktionen einer höheren Ordnung enthalten sind. Die Funktionen erster Ordnung entsprechen denen der LAGRANGE-Standardbasis. Für einen Polynomgrad  $p > 1$  wird der bestehende Ansatzraum der linearen Funktionen  $N_1(\xi)$  und  $N_2(\xi)$  um weitere Ansatzfunktionen  $N_i(\xi), i = 3, 4, \dots$  erweitert:

$$N_1(\xi) = \frac{1}{2}(1 - \xi) \quad (2.1)$$

$$N_2(\xi) = \frac{1}{2}(1 + \xi) \quad (2.2)$$

$$N_i(\xi) = \phi_{i-1}(\xi), \quad i = 3, 4, \dots, p + 1. \quad (2.3)$$

Die Polynome  $\phi_{i-1}(\xi)$  ergeben sich durch Integration von LEGENDRE-Polynomen  $L_j(\xi)$ :

$$\phi_j(\xi) = \sqrt{\frac{2j-1}{2}} \int_{-1}^{\xi} L_{j-1}(x) dx = \frac{1}{\sqrt{4j-2}}(L_j(\xi) - L_{j-2}(\xi)), \quad j = 2, 3, \dots \quad (2.4)$$

Diese sind als Lösung der LEGENDRE-Differentialgleichung definiert und lassen sich auf unterschiedliche Arten bestimmen. Siehe hierzu beispielsweise in [21].

Mit den Gleichungen (2.1) bis (2.3) sind damit alle Ansatzfunktionen einer eindimensionalen, hierarchischen Basis nach SZABÓ und BABUŠKA definiert. Die beiden ersten Basisfunktionen  $N_1(\xi)$  und  $N_2(\xi)$  verlaufen dabei linear über das eindimensionale Element und weisen an jeweils einem Endknoten den Wert 1, an dem anderen den Wert 0 auf. Diese werden als Knotenmoden (engl. *nodal modes*) bezeichnet. Alle höherpolynomialen Ansätze  $N_i(\xi)$  besitzen dagegen nur im Inneren des Elementes Werte verschieden von Null

$$N_i(-1) = N_i(1) = 0, \quad i = 3, 4, \dots, p + 1 \quad (2.5)$$

und werden als interne Moden (engl. *internal modes*) oder *bubble modes* bezeichnet.

LEGENDRE-Polynome sind orthogonal. Aus dieser Eigenschaft folgt für die paarweisen Ableitungen der (integrierten) Ansatzfunktionen in (2.3):

$$\int_{-1}^1 \frac{dN_i}{d\xi} \frac{dN_j}{d\xi} d\xi = \delta_{ij}, \quad i, j \geq 3. \quad (2.6)$$

Diese Orthogonalitätseigenschaft hat unmittelbare Auswirkungen auf die Kondition der resultierenden Steifigkeitsmatrix, da sich die Anteile stärker auf der Hauptdiagonalen konzentrieren.

Im Vergleich zur Standardbasis unter Verwendung von LAGRANGE-Polynomen kann sich dies mitunter positiv auf die Lösung des Gesamtgleichungssystems auswirken.

Die Hierarchie ihrer Ansatzfunktionen hat zudem auch einen unmittelbaren Einfluss auf die Struktur der resultierenden Steifigkeitsmatrizen. Werden die Gleichungen so geordnet, dass alle linearen Ansätze von 1 bis  $n_1$ , alle quadratischen von  $n_1 + 1$  bis  $n_2$  usw. nummeriert sind, sind die Steifigkeitsmatrizen einer niedrigeren Ordnung  $p_1 < p_2$  immer als Untermatrix in einer zu  $p_2$  korrespondierenden Steifigkeit enthalten. Dieser Umstand lässt sich beispielsweise nutzen, wenn auf Elementebene innere Freiheitsgrade eliminiert werden sollen, um eine Reduzierung der Gesamtfreiheitsgrade zu erreichen. In diesem Falle liegen die entsprechenden Gleichungen innerhalb eines Blockes vor, so dass keine Zeilvertauschungen mehr vorzunehmen sind (vgl. [105]).

### 2.2.1.2 Ansatzfunktionen für Hexaederelemente

Eine Erweiterung auf zwei und drei Raumdimensionen ergibt sich prinzipiell aus den tensoriellen Produkten der in Gleichung (2.1) bis (2.3) definierten Ansatzfunktionen. Die Berechnung der in dieser Arbeit betrachteten Volumenstrukturen basiert dabei auf einer Diskretisierung mittels Hexaederelementen. Das entsprechende Standardelement ist in Abbildung 2.7 dargestellt. Auf diesem werden nun Elementansätze in Abhängigkeit der drei lokalen Richtungen

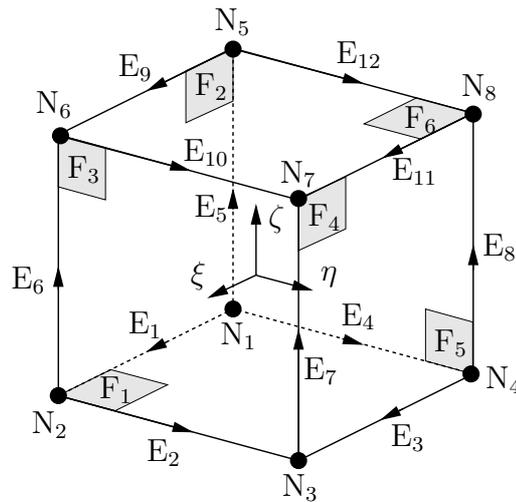


Abbildung 2.7: Definition von Knoten und Kanten und Flächen im Hexaederelement auf dem Standardgebiet  $\Omega_{st}^h = [(-1, 1) \times (-1, 1) \times (-1, 1)]$

$\xi, \eta$  und  $\zeta$  definiert. Es werden dabei vier Typen von Ansatzfunktionen unterschieden:

- Knotenmoden (*nodal modes*): Die acht Knotenmoden ergeben sich in gleicher Weise wie bei klassischen trilinearen Elementansatzfunktionen erster Ordnung.  $(\xi_i, \eta_i, \zeta_i)$  entsprechen den lokalen Koordinaten des Knotens  $i$ :

$$N_{1,1,1}^{N_i}(\xi, \eta, \zeta) = \frac{1}{8} (1 + \xi_i \xi)(1 + \eta_i \eta)(1 + \zeta_i \zeta), \quad i = 1, \dots, 8 \quad (2.7)$$

- **Kantenmoden (*edge modes*):** Kantenmoden sind immer jeweils einer der zwölf Elementkanten zugeordnet. Sie werden zu den gegenüberliegenden Kanten auf den benachbarten Flächen hin linear ausgeblendet und verschwinden auch auf allen Knoten. Durch geeignete Wahl der Vorzeichen innerhalb der Blendingterme  $(1 \pm \eta)$  und  $(1 \pm \zeta)$  wird die entsprechende Kante „aktiviert“. Beispielsweise bestimmen sich die Ansatzfunktionen der Kante  $E_3$  des Standardelements aus Abbildung 2.7 zu:

$$N_{i,1,1}^{E_3}(\xi, \eta, \zeta) = \frac{1}{4}(1 + \eta)(1 - \zeta)\phi_i(\xi) \quad (2.8)$$

- **Flächenmoden (*face modes*):** In analoger Weise bestimmen sich die Flächenmoden aus dem Tensorprodukt zweier eindimensionaler Ansatzfunktionen  $\phi_i(\xi)$ , wobei diese zur gegenüberliegenden Fläche hin ausgeblendet werden. Für die Fläche  $F_1$  in Abbildung 2.7 gilt:

$$N_{i,j,1}^{F_1}(\xi, \eta, \zeta) = \frac{1}{2}(1 - \zeta)\phi_i(\xi)\phi_j(\eta) \quad (2.9)$$

- **Innere Moden (*internal modes*):** Innere Moden sind rein lokal und verschwinden an allen Knoten, Kanten und Flächen des Elements:

$$N_{i,j,k}^{\text{int}}(\xi, \eta, \zeta) = \phi_i(\xi)\phi_j(\eta)\phi_k(\zeta) \quad (2.10)$$

### 2.2.1.3 Ansatzräume der Diskretisierung

Aus der Menge der in (2.7) bis (2.10) definierten Ansatzfunktionen werden Finite-Elemente-Ansatzräume  $\mathcal{S}$  aufgespannt, im Rahmen derer eine Lösung des betrachteten Problems approximiert werden kann. Im dreidimensionalen Fall unterscheidet man gemäß SZABÓ und BABUŠKA [120] drei Ansatzräume: *trunk space*,  $\mathcal{S}_{\text{ts}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^h)$ , *tensor product space*  $\mathcal{S}_{\text{ps}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^h)$  und einen *anisotropic tensor product space*  $\mathcal{S}^{p, p, q}(\Omega_{\text{st}}^h)$ . Der *volle* Ansatzraum  $\mathcal{S}_{\text{ps}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^h)$  ergibt sich aus dem Tensorprodukt aller bis zu den gegebenen Polynomgraden  $p_\xi, p_\eta, p_\zeta$  definierten Ansätze hinsichtlich der drei lokalen Richtungen. Zur Definition der beiden anderen Räume wird der volle Ansatzraum um bestimmte Funktionen reduziert, um ein effizienteres Approximationsverhalten zu erzielen. Die Ansatzräume  $\mathcal{S}_{\text{ts}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^h)$  und  $\mathcal{S}_{\text{ps}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^h)$  sind hinsichtlich den Polynomgraden  $p_\xi, p_\eta$  und  $p_\zeta$  der drei lokalen Richtungen beliebig frei wählbar. Man spricht in diesem Falle von einem *anisotropen* Ansatz, im Gegensatz zu einem *isotropen* Ansatzraum, wenn alle lokalen Polynomgrade gleich gewählt werden. Für detaillierte Informationen zum Aufbau hierarchischer Ansatzräume wird auf die Literatur verwiesen [120, 121, 35].

## 2.2.2 Hexaederelemente zur Berechnung dünnwandiger Strukturen

Im Folgenden soll eine Methodik zur Berechnung platten- oder schalenartiger Strukturen auf Basis strikt dreidimensionaler Elemente hoher Ordnung betrachtet werden. Die hier gezeigte Formulierung mittels anisotroper Ansätze lässt sich dabei sehr flexibel an die Gegebenheiten in dünnwandigen (wie auch stabartigen) Strukturen anpassen. Abbildung 2.8 illustriert hierzu

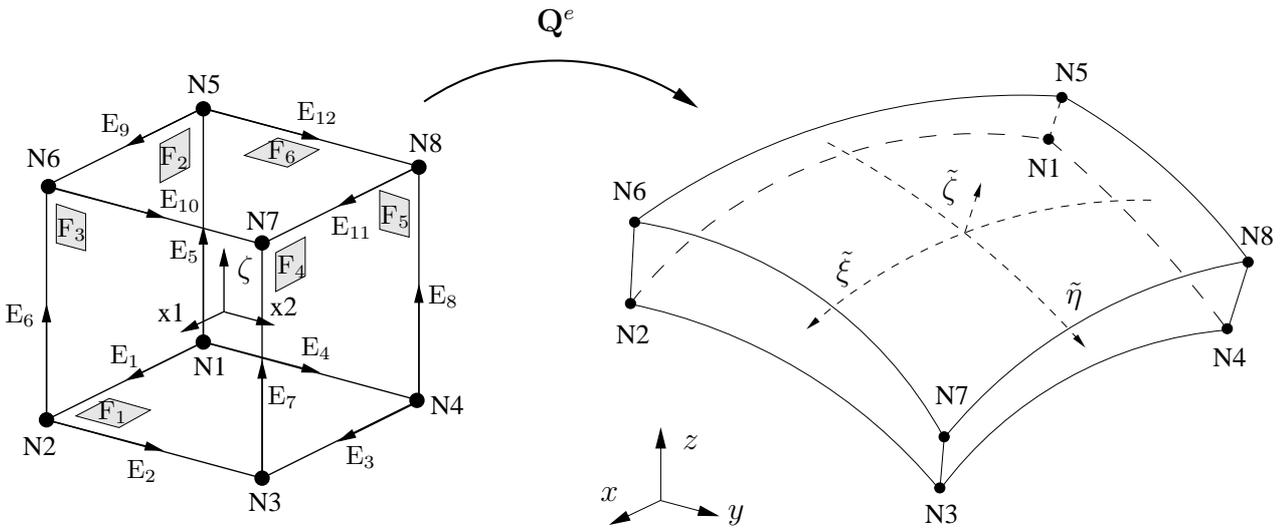


Abbildung 2.8: Abbildung eines Hexaederelements auf schalenartige Struktur [35].

das Prinzip. Eine Schalenstruktur wird mithilfe von Hexaederelementen diskretisiert. Der Polynomgrad der drei lokalen Richtungen  $\xi$ ,  $\eta$  und  $\zeta$  kann beliebig entsprechend den Verhältnissen innerhalb der Schale bestimmt werden.

Die Idee zur Berechnung dünnwandiger Strukturen mittels anisotroper Volumenelemente der  $p$ -Version geht zurück auf SZABÓ und SAHRMANN [122]. Das Konzept wurde später von DÜSTER et al. [35, 37, 41] auf alle in Abschnitt 2.2.1.3 angesprochenen Ansatzräume erweitert und zusätzlich die Möglichkeit geschaffen, neben den Polynomgraden der lokalen Richtungen auch die der Komponenten des Verschiebungsansatzes  $\mathbf{u} = (u_x, u_y, u_z)^T$  beliebig anpassen zu können. Es ergeben sich damit insgesamt  $3 \times 3$  Polynomgrade, mit denen das Approximationsverhalten eines Ansatzes innerhalb eines Elements definiert werden kann. Diese lassen sich in einer Matrix zusammenfassen:

$$\mathbf{p} = \begin{matrix} & u_x & u_y & u_z \\ \begin{matrix} \xi \\ \eta \\ \zeta \end{matrix} & \begin{pmatrix} p_{\xi x} & p_{\xi y} & p_{\xi z} \\ p_{\eta x} & p_{\eta y} & p_{\eta z} \\ p_{\zeta x} & p_{\zeta y} & p_{\zeta z} \end{pmatrix} & = & \begin{pmatrix} p_{\xi} \\ p_{\eta} \\ p_{\zeta} \end{pmatrix} \end{matrix} \quad (2.11)$$

Gewöhnlich werden zur Approximation von plattenartigen Strukturen die Polynomgrade in Plattenrichtung  $p_{\xi}$  und  $p_{\eta}$  höher gewählt, als derjenige in Dickenrichtung  $p_{\zeta}$ . Bei reinen Biegeverhältnissen muss in Dickenrichtung der Platte oft nur ein linearer Spannungsverlauf darstellbar sein, während in Plattenrichtung natürlich beliebige Verschiebungsformen möglich sein können. Grundsätzlich wird für platten- oder schalenartige Strukturen folgendes Schema vorgeschlagen:

$$\mathbf{p} = \begin{matrix} & u_x & u_y & u_z \\ \begin{matrix} \xi \\ \eta \\ \zeta \end{matrix} & \begin{pmatrix} p & p & p \\ p & p & p \\ q & q & q \end{pmatrix} & = & \begin{pmatrix} p \\ p \\ q \end{pmatrix}, \end{matrix} \quad (2.12)$$

in dem der Polynomgrad  $p$  in Plattenrichtung  $\xi, \eta$  höher gewählt wird, als derjenige in Dickenrichtung  $q$ .

Untersuchungen von RANK et al. [97] sowie DÜSTER [35] zeigen, dass der hier betrachtete Ansatz dimensionsreduzierten Schalenmodellen hinsichtlich der Genauigkeit im globalen Sinne ebenbürtig ist. Vergleicht man Berechnungen mit gleicher Anzahl an Freiheitsgraden, liefern beide Verfahren eine Approximation mit ähnlichen Werten des globalen Fehlers in der Energienorm. Lokal sind die Unterschiede jedoch größer. In Bereichen mit echt dreidimensionalen Spannungszuständen, wie etwa bei Lasteinleitungen oder im Übergang zu massiven Strukturen, können dimensionsreduzierte Modelle die wahren Verhältnisse nicht mehr ausreichend wiedergeben. Hier ist ein volumenorientierter Ansatz klar überlegen.

Es ist in dem Zusammenhang zu bemerken, dass die Entscheidung, in welcher Form die Polynomgrade für ein Element zu verteilen sind, in erster Linie *a priori*, also vor einer Berechnung getroffen werden muss. Dies ist allerdings bei Verwendung dimensionsreduzierter Modelle genauso. Es muss auch dort vor einer Berechnung entschieden werden, ob ein gegebenes Tragwerksmodell die realen Verhältnisse ausreichend genau beschreiben kann. Es sind in beiden Fällen also die Modellannahmen zu überprüfen und der entsprechende Polynomgrad beziehungsweise das entsprechende (dimensionsreduzierte) Berechnungsmodell zu wählen. Mit der Verwendung von Volumenelementen bietet sich nun aber ein wesentlicher Unterschied: Während im Falle dimensionsreduzierter Strukturen eine unzureichende Beschreibung der realen Verhältnisse einen *Modellfehler* darstellt, ist dies im Falle zu geringer Polynomgrade nur ein *Diskretisierungsfehler* der Finite-Elemente-Approximation. Letzterer lässt sich numerisch schätzen und adaptiv steuern. Ein entsprechendes Konzept wurde hierzu von SCHOLZ et al. [116] vorgestellt (vgl. auch [96, 43]). Der Autor macht sich dabei die Hierarchie der Ansatzfunktionen zu Nutze. Im Vergleich der hierarchischen Ansatzräume schätzt er den Fehler jeweils getrennt für die lokalen Richtungen und bestimmt daraus *a posteriori* eine verbesserte Einteilung der Polynomgrade innerhalb der Matrix in (2.11), die für eine gegebene Anzahl an Freiheitsgraden den Fehler aus der Finite-Elemente-Diskretisierung minimiert. Für das Beispiel am Ende dieses Kapitels (Abschnitt 2.3) soll dieses adaptive Verfahren zur Anwendung kommen und dabei mit einer isotropen Polynomgradverfeinerung verglichen werden.

### 2.2.3 Abbildung gekrümmter Elementoberflächen

Wichtig für die Modellierung von Bauwerksstrukturen mittels Elementen der  $p$ -Version sind Konzepte zur Beschreibung gekrümmter Elementoberflächen. Die  $h$ -Version der Methode der finiten Elemente senkt den Diskretisierungsfehler durch globale oder lokale Elementverfeinerung. Infolge der damit einhergehenden Verringerung des Elementdurchmessers nähert sich der Rand des Netzes automatisch einer exakt vorgegebenen Geometrie an. Bei Elementen hoher Ordnung bleibt die Elementgröße dagegen konstant. Um bei einer schrittweisen Erhöhung des Polynomgrades neben dem Diskretisierungsfehler auch den Fehler aus der geometrischen Approximation reduzieren zu können, müssen Verfahren angewendet werden, die auch hier eine Annäherung an die tatsächliche Geometrie ermöglichen. Ein entsprechendes Werkzeug dazu ist die *Blending-Funktionen-Methode*, die erstmals von GORDON und HALL [53] vorgestellt wurde und welche eine Beschreibung von finiten Elementen mit beliebig berandeten Oberflächen erlaubt. Bild 2.9 motiviert das entsprechende Vorgehen am Beispiel eines Hexaederelements, welches auf ein geometrisches Objekt mit einer gekrümmten Seitenfläche abgebildet wird.

Im Folgenden sollen nur die wesentlichen Kernpunkte skizziert werden. Für detaillierte Informationen sei wiederum auf die Literatur verwiesen [53, 120, 20]. Wie im vorherigen Abschnitt angesprochen, werden in der Finite-Elemente-Technik alle Ansatzfunktionen auf einem

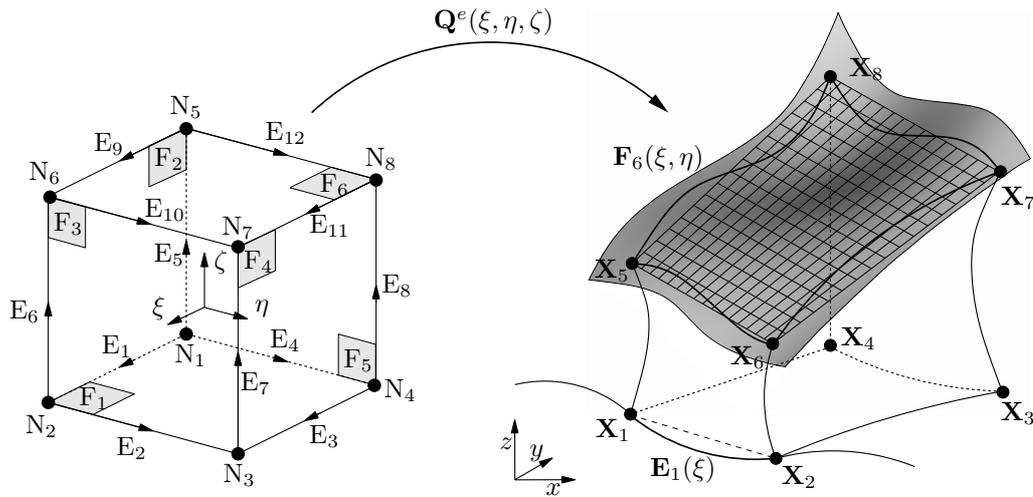


Abbildung 2.9: Abbildung eines Hexaederelementes auf Elementgeometrie mit gekrümmten Seitenflächen.

Standardelement definiert. Zur Berücksichtigung der tatsächlichen Elementgeometrie wird eine Transformation  $\mathbf{Q}^e$  eingeführt, die die lokalen Koordinaten  $(\xi, \eta, \zeta)^T$  auf den globalen Raum  $(x, y, z)^T$  abbildet. Im Sinne des so genannten isoparametrischen Abbildungskonzepts wird bei Elementen niedriger Ordnung die Transformation  $\mathbf{Q}^e$  aus den gleichen Ansatzfunktionen gebildet, wie der Ansatz zur Approximation des Verschiebungsfeldes:

$$\mathbf{x} = \mathbf{Q}^e(\xi, \eta, \zeta) = \sum_{i=1}^8 N_{1,1,1}^{N_i}(\xi, \eta, \zeta) \mathbf{X}_i. \quad (2.13)$$

Die Funktionen  $N_{1,1,1}^{N_i}(\xi, \eta, \zeta)$  entsprechen dabei den acht Knotenansatzfunktionen aus Gleichung (2.7). Die Blending-Funktionen-Methode erweitert nun Gleichung (2.13) um so genannte Flächen- und Kantenblendingterme:

$$\mathbf{x} = \mathbf{Q}^e(\xi, \eta, \zeta) = \sum_{i=1}^8 N_{1,1,1}^{N_i}(\xi, \eta, \zeta) \mathbf{X}_i + \sum_{i=1}^6 \mathbf{f}_i(\xi, \eta, \zeta) - \sum_{i=1}^{12} \mathbf{e}_i(\xi, \eta, \zeta). \quad (2.14)$$

In die sechs Flächen bzw. zwölf Kanten terme fließen nun die geometrischen Beschreibungen der jeweiligen Flächen oder Kanten ein. Diese sind in der Regel in Form einer parametrischen Beschreibung einer Kurve oder einer Fläche des dreidimensionalen Raumes  $\mathbb{R}^3$  gegeben. Dabei müssen im Allgemeinen die lokalen Koordinaten der Flächen- oder Kantenbeschreibungen in geeigneter Weise auf die Koordinaten  $\xi, \eta, \zeta$  des Standardelements abgebildet werden, wobei in der Regel eine Reparametrisierung notwendig ist. Zur Motivation soll an dieser Stelle beispielhaft der Kantenblendingterm  $\mathbf{f}_6(\xi, \eta, \zeta)$  in Gleichung (2.14) näher betrachtet werden: Gegeben sei hierzu die parametrische Darstellung der in Abbildung 2.9 farblich hervorgehobenen Oberfläche  $\mathbf{F}_6(r, s) = (F_x(r, s), F_y(r, s), F_z(r, s))^T$  in lokalen Flächenkoordinaten  $r, s$ . Nach geeigneter Transformation der allgemeinen Flächenbeschreibung auf die lokalen Koordinaten des finiten Elements gilt:  $\mathbf{F}_6 = \mathbf{F}_6(\xi, \eta)$ . Für diese Fläche berechnet sich der Blendingterm aus

Gleichung (2.14) dann folgendermaßen:

$$\mathbf{f}_6(\xi, \eta, \zeta) = \left[ \mathbf{F}_6(\xi, \eta) - \frac{1}{4}(1-\xi)(1-\eta)\mathbf{X}_5 + \frac{1}{4}(1+\xi)(1-\eta)\mathbf{X}_6 + \frac{1}{4}(1+\xi)(1+\eta)\mathbf{X}_7 + \frac{1}{4}(1-\xi)(1+\eta)\mathbf{X}_8 \right] \left( \frac{1+\zeta}{2} \right). \quad (2.15)$$

Der in eckigen Klammern gegebene Ausdruck beschreibt dabei die Differenz zwischen der Flächendarstellung und einer bilinearen Oberfläche, die von den Knoten  $\mathbf{X}_5$  bis  $\mathbf{X}_8$  aufgespannt wird. Dieser Differenzausdruck wird dann mittels des Terms  $[(1+\zeta)/2]$  zur gegenüberliegenden Fläche  $\mathbf{F}_1$  hin „ausgeblendet“. Die Blendingterme aller anderen Flächen und Kanten werden in analoger Weise bestimmt. Es ist noch zu bemerken, dass zur Berechnung der Abbildungsfunktion in (2.14) die Terme des Kantenblendings  $\mathbf{e}_i(\xi, \eta, \zeta)$  abgezogen werden. Dies liegt daran, dass die Kanten schon zweimal in den angrenzenden Flächen berücksichtigt wurden und somit nun einmal abgezogen werden müssen.

Mithilfe der Blending-Funktionen-Methode können so beliebige Geometriedarstellungen auf das Finite-Elemente Netz abgebildet werden. Dabei kann die Verwaltung von Geometrie und Berechnung in einer verteilten Softwareumgebung weitgehend entkoppelt werden. In der Arbeit von BRÖKER [20] wird hierzu ein Konzept vorgestellt, in dem das geometrische Modell in einem CAD-System verwaltet wird und nur Informationen der Abbildungsfunktion zur Laufzeit per Netzwerkkommunikation an ein Finite-Elemente-Programm übermittelt werden. Im CAD-System können so leicht Modifikationen am geometrischen Modell vorgenommen werden, welche dann bei einer Neuberechnung innerhalb der Struktursimulation direkt berücksichtigt werden.

## 2.3 Beispiel: UNIQA-Tower

Im Folgenden soll nun an einem konkreten Beispiel der gesamte Weg vom Architekturmodell zur einer volumenorientierten Tragwerkssimulation nochmalig vorgestellt werden. Das hier betrachtete Modell ist dabei einem realen Bauwerk, der Konzernzentrale der UNIQA-Versicherung, dem so genannten UNIQA-Tower<sup>2</sup> in Wien, nachempfunden. Die Zentrale der

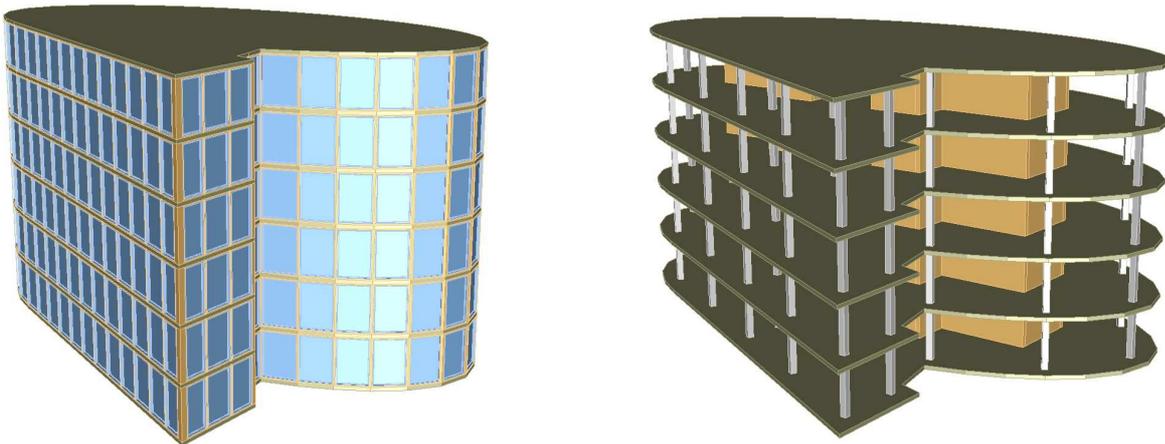


Abbildung 2.10: Bauwerksmodell (links), Tragwerksmodell (rechts) von fünf Stockwerken des UNIQA-Towers.

Versicherung besteht aus einem Hochhaus mit insgesamt 21 Stockwerken, mehreren Kellergeschossen sowie einem Nebengebäude. Zur Vereinfachung wurden für das folgende Beispiel fünf Stockwerke des Hochhauses betrachtet. Abbildung 2.10 zeigt auf der linken Seite das entsprechende Gebäudemodell, wie es beispielsweise vom Architekten erzeugt wird. Das Modell ist 15m hoch, 25m lang und 18m breit. Es wurde in diesem Falle mithilfe des CAD-Systems *Architectural Desktop*<sup>3</sup> von *Autodesk* modelliert und als Datei im Format der Industry Foundation Classes (IFC) exportiert. Nach dem in Abschnitt 2.1.1 beschriebenen Vorgehen wurde dabei in einem ersten Schritt das Bauwerksmodell in ein explizites Geometriemodell in B-Rep-Darstellung umgewandelt und die zur Tragwerksberechnung wichtigen Material- und Bauteilinformationen aus dem IFC-Modell übernommen.

Abbildung 2.10 zeigt auf der rechten Seite das Tragwerksmodell des Gebäudes. Dazu wurden aus dem Modell des Architekten alle Bauteile vom Typ Wand, Stütze und Decke extrahiert. Dann wurden Lasten gesetzt, Verschiebungsrandbedingungen festgelegt sowie weitere numerische Berechnungsparameter (z.B. Polynomgrad) den Bauteilen zugewiesen. Die Ableitung des Geometriemodells aus den Gebäudemolldaten sowie die Attributierung erfolgte dabei in einer am Lehrstuhl für Bauinformatik hierfür entwickelten grafisch interaktiven Anwendung. Eine Attributierung kann dabei sowohl direkt am Tragwerksmodell als auch noch später am Finite-Elemente-Netz erfolgen. Nähere Informationen zu den verwendeten Softwarewerkzeugen sowie dem Verfahren der Ableitung des Geometriemodells finden sich in der Arbeit von

<sup>2</sup>Der UNIQA Tower: <http://tower.uniqa.at>

<sup>3</sup>Autodesk Architectural Desktop: <http://www.autodesk.de/architecturaldesktop>

ROMBERG [103]. Auf Basis der im Gebäudemodell gegebenen Materialinformationen wurden zusätzlich aus einer Datenbank Materialkennwerte wie Elastizitätsmodul und Querdehnzahl übernommen und den Elementen des Tragwerksmodells zugeordnet. Für das hier betrachtete Beispiel wird als Verschiebungsrandbedingung die Bodenplatte im Bereich der Stützen und Wände als eingespannt festgelegt und das Gebäude mit seinem Eigengewicht belastet. Tabelle 2.1 listet die verwendeten Materialparameter auf.

Elastizitätsmodul	$E$	$=$	32000	[MN/m <sup>2</sup> ]
Querdehnzahl	$\nu$	$=$	0,2	
Eigengewicht (Beton)	$v_z$	$=$	0,025	[MN/m <sup>3</sup> ]

Tabelle 2.1: Materialkennwerte und Lastannahmen für die Berechnung.

Vereinfachend wird bei der Modellierung des Tragsystems angenommen, dass alle Bauteile monolithisch miteinander verbunden sind. Dies kann natürlich die realen Verhältnisse in Bauwerken nicht immer ausreichend beschreiben. Insbesondere im Zusammenhang mit der hier betrachteten volumenorientierten Tragwerksanalyse besteht im Hinblick auf diesen Aspekt sicherlich noch die Notwendigkeit, weitere Konzepte zur Modellierung von Bauteilverbindungen zu entwickeln. Dies soll jedoch nicht Thema der vorliegenden Arbeit sein. Abbildung 2.11 zeigt

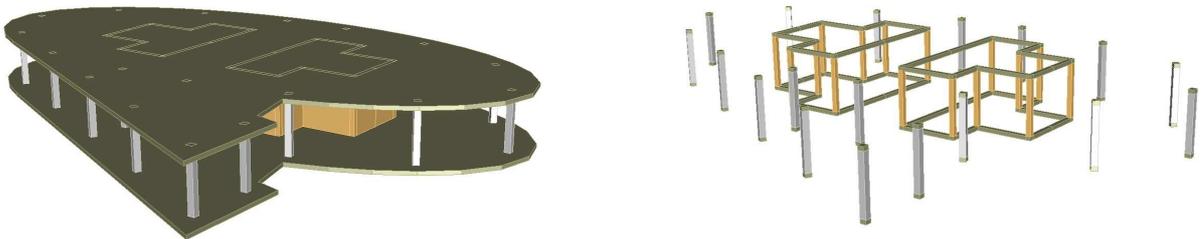


Abbildung 2.11: Verbindungsmodell nach [103]. Links: Gesamtmenge  $M_V$  aller Verbindungsobjekte des Modells. Rechts: Teilmenge  $M_K$  aller Kopplungsobjekte.

am Beispiel eines einzelnen Stockwerks den nächsten Schritt, die Generierung des so genannten Verbindungsmodells entsprechend dem von ROMBERG [103] entwickelten Vorgehen. Auf der linken Seite sind alle Objekte der Menge  $M_V$  des Verbindungsmodells zu sehen. Man kann dabei die Zerlegung der Deckenplatte an der Verbindung zum Gebäudekern erkennen. Auf der rechten Seite von Abbildung 2.11 sind die stabförmigen Kopplungsobjekte<sup>4</sup> der Menge  $M_K$  dargestellt. Die Funktion der Kopplungsobjekte wurde in Abschnitt 2.1.2 diskutiert. Sie stellen innerhalb des Verbindungsmodells eindeutige Nachbarschaftsrelationen über gemeinsame Flächen sicher und dienen zudem, im weiteren Verlauf der Vernetzung eine konsistente Einteilung der Elementknoten an benachbarten Bauteilrändern sicherzustellen.

Abbildung 2.12 zeigt das resultierende Finite-Elemente-Netz zusammen mit einem Detailausschnitt der Struktur. Das Modell besteht dabei aus 8313 Hexaederelementen. Abbildung

<sup>4</sup>Zum besseren Verständnis werden in dem Bild auch alle Stützen dargestellt, die eigentlich nicht den Kopplungsobjekten in  $M_K$ , sondern den Differenzobjekten in  $M_D$  zugerechnet werden.

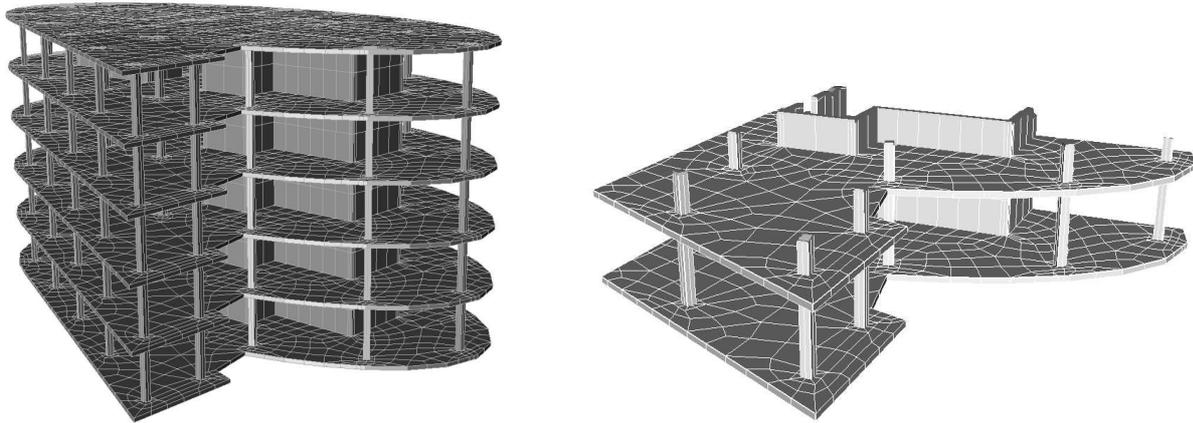


Abbildung 2.12: Hexaedernetz des Gebäudes bestehend aus 8313 Elementen mit einem Detailausschnitt der Struktur.

2.13 zeigt schließlich die Verschiebungen und Spannungen innerhalb des Gebäudes unter einer Belastung mit Eigengewicht. Für die Berechnungen wurde jedoch eine schrittweise Steigerung

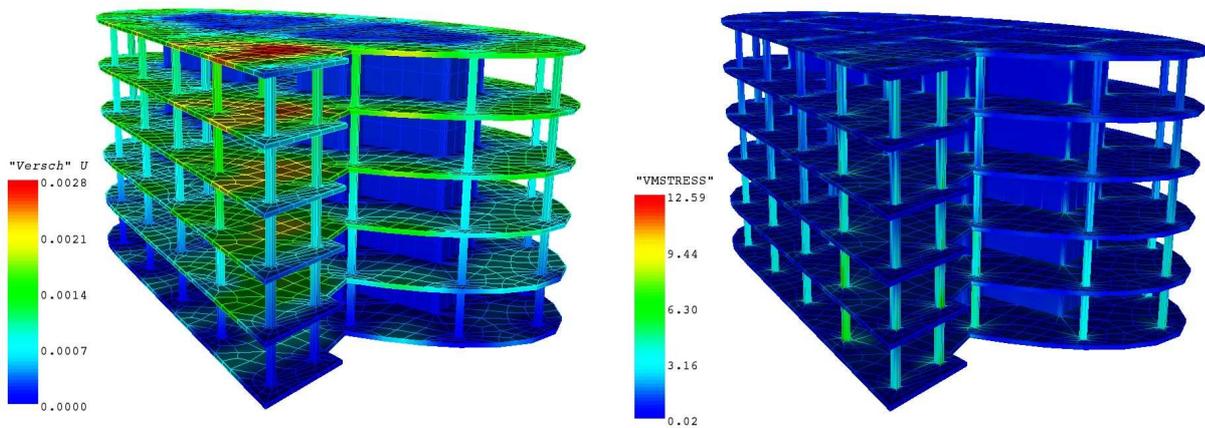


Abbildung 2.13: Links: Verschiebungen  $|\mathbf{u}|$  des Gesamtsystems. Rechts: Darstellung der VON MISES-Vergleichsspannungen unter Belastung aus Eigengewicht.

der Belastung während einzelner Bauphasen nicht berücksichtigt. Es ist zu erkennen, dass die vom Gebäudekern entfernten Stützen wesentlich stärker belastet sind, als kernnahe Stützen. Der Gebäudekern selbst weist nahezu keine Verformungen auf – was zu erwarten ist, da dieser ohnehin für ein wesentlich höheres Gebäude konzipiert ist.

Die folgende Untersuchung soll eine Abschätzung liefern, mit welchem Aufwand sich ein derartiges Volumenmodell mithilfe der  $p$ -Version der Finite-Elemente-Methode berechnen lässt. Hierzu wurde in mehreren Berechnungsdurchläufen der Polynomgrad der Elemente von  $p = 1$

bis  $p = 6$  gleichmäßig (isotrop) erhöht und mit der dafür notwendigen Anzahl an Freiheitsgraden verglichen. Tabelle 2.2 stellt die Ergebnisse dar. Weitere Untersuchungen am gleichen

$p$	Freiheitsgrade	$\mathcal{U}(\mathbf{u}_{FE})$
1	46626	$3,062072780 \cdot 10^{-3}$
2	166437	$4,424545563 \cdot 10^{-3}$
3	286248	$4,713956380 \cdot 10^{-3}$
4	503871	$4,794534846 \cdot 10^{-3}$
5	819306	$4,838013542 \cdot 10^{-3}$
6	1257492	$4,858802547 \cdot 10^{-3}$

Tabelle 2.2: Anzahl an Freiheitsgraden und Verzerrungsenergie  $\mathcal{U}(\mathbf{u}_{FE})$  [MNm] für steigende Polynomgrade.

Modell zum numerischen Aufwand in Form von Rechenzeiten finden sich zudem in Abschnitt 3.4.4 dieser Arbeit.

Um die Güte der hier gewählten Finite-Elemente-Diskretisierung abschätzen zu können soll als Maß der Fehler in der Energienorm  $\|\mathbf{e}\|_{E(\Omega)}$  betrachtet werden. Dieser ergibt sich aus dem Vergleich der auf dem Gesamtgebiet  $\Omega$  berechneten elastischen Verzerrungsenergie einer FE-Lösung  $\mathbf{u}_{FE}$  mit der exakten Lösung  $\mathbf{u}$ :

$$\|\mathbf{e}\|_{E(\Omega)} = \sqrt{\mathcal{U}(\mathbf{u}) - \mathcal{U}(\mathbf{u}_{FE})}. \quad (2.16)$$

Die elastische innere Verzerrungsenergie entspricht dabei:

$$\mathcal{U}(\mathbf{u}) = \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\sigma}(\mathbf{u}) \, dV. \quad (2.17)$$

Um den Fehler in Gleichung (2.16) bestimmen zu können, lassen sich Extrapolationsverfahren anwenden, die auf Basis mehrerer Berechnungsdurchläufe eine Abschätzung für die exakte Verzerrungsenergie  $\mathcal{U}(\mathbf{u})$  liefern. Das entsprechende Vorgehen findet sich in SZABÓ und BABUŠKA [120]. Für das oben betrachtete Beispiel ergibt sich durch Vergleich der Berechnungen mit Polynomgraden  $p = \{4, 5, 6\}$  eine Abschätzung der Verzerrungsenergie zu  $\mathcal{U}(\mathbf{u}) = 4.89 \cdot 10^{-3}$  MNm.

Es soll an dieser Stelle nun zusätzlich eine anisotrope und adaptive Polynomgradanpassung betrachtet werden (vgl. hierzu Abschnitt 2.2.2). Das in [96, 43, 116] vorgeschlagene Verfahren schätzt dabei im Vergleich der hierarchischen Ansatzräume den Fehler hinsichtlich der lokalen Richtungen sowie hinsichtlich der räumlichen Verschiebungskomponenten ab und passt damit die Polynomgrade innerhalb der in (2.11) dargestellten Matrix für jedes Element an. Abbildung 2.14 vergleicht hierzu die isotrope mit der anisotrop adaptiven Polynomgradverfeinerung für das obige Berechnungsbeispiel. Die Kurven stellen den relativen Fehler in der Energienorm aufgetragen über die Anzahl der Freiheitsgrade dar. Es ist zu erkennen, dass mit einer optimalen Anpassung des Polynomgrades bei gleicher Genauigkeit eine Reduzierung der notwendigen Freiheitsgrade auf etwa die Hälfte erreicht werden kann. Die beiden Kurven zeigen dabei weitgehend eine algebraische Konvergenz (vgl. [120, 35, 116]). Da in der  $p$ -Version für glatte Probleme bei Steigerung des Polynomgrades eine exponentielle Konvergenz erwartet wird, ist zu vermuten, dass in diesem Beispiel der Fehler weitgehend von

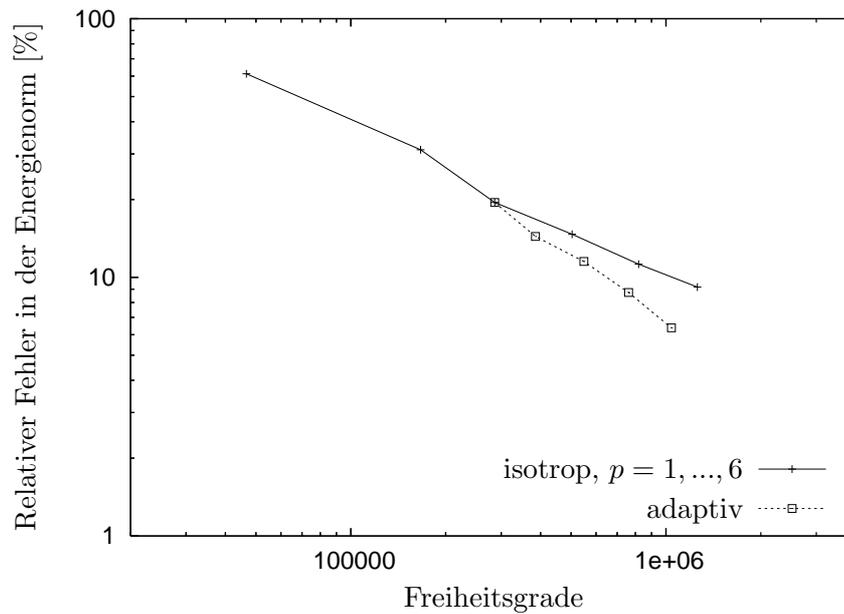


Abbildung 2.14: Prozentualer relativer Fehler in der Energienorm  $\eta = \sqrt{\frac{\mathcal{U} - \mathcal{U}_{FE}}{\mathcal{U}}}$  gegenüber der Anzahl an Freiheitsgraden für steigende Polynomgrade. Vergleich von isotroper Polynomgradverfeinerung mit einer anisotrop adaptiven Verfeinerung. Doppelt logarithmische Darstellung.

auftretenden Singularitäten dominiert wird. Es ist daher anzunehmen, dass die Konstruktion des Finite-Elemente-Netzes besonders an den Verbindungsstellen benachbarter Bauteile noch stärker beachtet werden muss und dort geeignete Verfeinerungen vorzunehmen sind, wenn eine weitere effektive Reduzierung des Fehlers erreicht werden soll.

# Kapitel 3

## Hierarchie als Grundlage der Modell- und Berechnungsorganisation

Ziel dieser Arbeit ist es, Konzepte aufzuzeigen, die eine strikt volumenorientierte Herangehensweise in der Tragwerksplanung ermöglichen. Um zudem während einer Bearbeitung der Modelle die mechanische Konsistenz gewährleisten zu können, sollen die Berechnungen immer am Gesamtmodell erfolgen. Aufgrund der damit einhergehenden höheren Komplexität ergeben sich allerdings erweiterte Anforderungen an die zugrunde liegenden Datenstrukturen, Berechnungsverfahren sowie Methoden zur Organisation und Interaktion. Konzepte der Hierarchie sollen in diesem Zusammenhang eine grundlegende Rolle einnehmen. Als Raumdatenstrukturen dienen diese zur Indizierung von Objekten und stellen die Basis für eine effiziente Ausführung geometrischer Algorithmen, wie beispielsweise in der Kollisionserkennung dar. Neben den rein geometrischen Aufgabenstellungen lassen sich hierarchische Datenstrukturen aber auch in anderen Bereichen sehr gewinnbringend einsetzen. Mit dem Prinzip der rekursiven Substrukturierung bieten diese im Rahmen der Strukturmechanik ein effizientes und zudem sehr flexibles Lösungsverfahren. Das grundlegende Prinzip ist dabei in allen Fällen gleich. Es basiert auf einer rekursiven Dekomposition eines Modells beziehungsweise des Raumes. Dabei lassen sich Algorithmen anwenden, die in der Informatik als *Teile und Herrsche*-Strategien bekannt sind und welche oft eine sehr effiziente Lösung von Problemen erlauben. Neben den rein algorithmischen Vorteilen bietet eine hierarchische Zerlegung zudem ideale Voraussetzungen für eine Verteilung von Aufgaben und Daten in einem Rechnernetz, was besonders im Hinblick auf die kooperative Zusammenarbeit durchaus von Bedeutung ist.

Fragen der Geometrie spielen in den in dieser Arbeit betrachteten Problemstellungen eine wichtige Rolle. Deutlich wird dies beispielsweise in den in Abschnitt 2.1 diskutierten Verfahren zur Dekomposition und Vernetzung eines Bauwerksmodells von ROMBERG sowie der automatischen Erkennung von Räumen von VAN TREECK. Grundlage beider Verfahren sind korrekt aufgebaute topologische Beziehungen zwischen den Bauteilen. Die mit CAD-Systemen erzeugten Modelle weisen allerdings oft größere Toleranzen und Ungenauigkeiten auf, was die Erkennung von Nachbarschaften deutlich erschwert. Hierarchische Datenstrukturen können hier helfen, um einerseits für die Nachbarschaftssuche den Bereich potentieller Kandidaten einzugrenzen, sowie andererseits Toleranzen im Modell zu erkennen und gegebenenfalls zu beseitigen. In der computergestützten Planung kommt somit einer Behandlung geometrisch-räumlicher Fragestellungen eine wesentliche Rolle zu, insbesondere, wenn die zugrunde liegenden Modelle immer größer und komplexer werden.

In diesem Zusammenhang stellt die Arbeit von MUNDANI [79], die – wie die vorliegende Arbeit – im Rahmen des DFG-Forschungsprojekts „Volumenorientierte Modellierung als Grundlage der kooperativen Arbeit im konstruktiven Ingenieurbau“ [99] entstanden ist, eine wichtige Grundlage dar. MUNDANI schlägt darin ein „Hierarchisches Geometriemodell zur Einbettung verteilter Simulationsmodelle“ vor. Ausgangspunkt einer Zusammenarbeit verschiedener Fachbereiche ist darin ein geometrisches „Metamodell“, von welchem bei Bedarf mithilfe echtzeitfähiger Algorithmen sowohl volumenorientierte als auch oberflächenorientierte Darstellungsarten für unterschiedliche Simulationsaufgaben abgeleitet werden. Neben der Generierung von Simulationsmodellen sollen hierarchische Datenstrukturen darüber hinaus zur Organisation und Steuerung von Berechnungsverfahren herangezogen werden. Dieses Konzept der hierarchischen Strukturierung von Berechnungsmodellen soll in der vorliegenden Arbeit aufgegriffen und im Rahmen einer strukturmechanischen Tragwerksanalyse näher untersucht werden.

Nach einer Einführung in räumlich hierarchische Geometriemodelle, wobei hier in erster Linie Oktalbäume betrachtet werden, soll in den weiteren Abschnitten dieses Kapitels das Prinzip der rekursiven Substrukturierung diskutiert und eine Anwendung für Probleme der  $p$ -Version der Finite-Elemente-Methode beschrieben werden. Zur Einordnung des Lösungsverfahrens soll neben einer Strukturierung mithilfe von Oktalbäumen auch eine rekursive Bisektion des Nachbarschaftsgraphen diskutiert und untersucht werden. Dabei wird gezeigt, dass für Probleme mit Elementen hoher Ordnung die Methode der rekursiven Substrukturierung als Lösungsverfahren für schwach besetzte Gleichungssysteme anderen Verfahren durchaus ebenbürtig ist. Besondere Vorteile ergeben sich darüber hinaus aber, wenn lokale Modifikationen am Modell vorgenommen werden, da aufgrund der hierarchischen Struktur der Aufwand zur Neuberechnung weitgehend auf den modifizierten Bereich beschränkt werden kann. Hierbei lassen sich zum Teil erhebliche Geschwindigkeitsvorteile erzielen.

### 3.1 Geometrische Modellierung

In den Computerwissenschaften spielen Aufgaben der *geometrischen Modellierung* eine wichtige Rolle. Wesentliche Schwerpunkte liegen dabei im Bereich des Computer-Aided Design (CAD) oder der Computergrafik. Zur Darstellung geometrischer Modelle wird in der Regel zwischen direkten und indirekten Schemata unterschieden. Indirekte Darstellungen beschreiben über die Oberfläche eines Körpers implizit sein Volumen. Die Oberfläche ist dabei meist aus mehreren Flächenstücken zusammengesetzt, die planar sind (z.B. Dreiecke, Vierecke oder ebene Polygone) oder eine gekrümmte Form haben können. Im Gegensatz dazu beschreiben direkte Schemata unmittelbar das Volumen eines Körpers. Mögliche Darstellungen bedienen sich einer Assemblierung eines komplexen Körpers durch Zellen (Normzellen-Aufzählungsschema) oder einfachen Grundprimitiven. Weitere Möglichkeiten bestehen in der Anwendung Boolescher Operationen (Vereinigung, Durchschnitt und Differenz) auf einen Satz von einfachen Grundobjekten (Kugel, Würfel, Zylinder, ...), um daraus komplexe Körper aufzubauen.

Eine Diskussion verschiedener geometrischer Darstellungsmöglichkeiten soll nicht Gegenstand dieser Arbeit sein. Eine Einführung in die Thematik findet man beispielsweise von BUNGARTZ et al. in [22] sowie eine Klassifizierung der Darstellungsschemata von Volumenmodellen von MÄNTYLÄ in [73]. Einen guten Überblick zu diesem Thema bietet auch SHAPIRO in [118].

## 3.2 Räumlich hierarchische Geometriemodelle

### 3.2.1 Baumdatenstrukturen

Bäume gehören sicherlich zu eine der wichtigsten in der Informatik verwendeten Datenstrukturen. Nach OTTMANN und WIDMAYER [90] lassen sich Bäume als Verallgemeinerung linearer Listen auffassen, bei denen ein Element nicht nur einen, sondern mehrere Nachfolger haben kann. Ausgehend von einem Startknoten, der *Wurzel* des Baumes, ergibt sich eine hierarchische Anordnung der Elemente. Abbildung 3.1 zeigt beispielsweise einen Baum mit 7 Knoten.

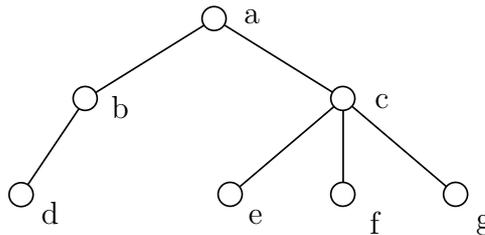


Abbildung 3.1: Baum der Höhe 3 mit Wurzel a und Blättern d,e,f,g.

Mit Ausnahme der Wurzel hat jeder Knoten genau einen Vorgänger und eine endliche begrenzte Anzahl von Nachfolgern, die auch als *Söhne* bezeichnet werden. Knoten, die am unteren Ende des Baumes stehen und keinen Nachfolger haben, bezeichnet man als *terminierende Knoten* oder *Blätter*. Die maximale Anzahl von Söhnen eines Knotens bezeichnet man als *Ordnung* des Baumes. Als weitere strukturelle Eigenschaften lassen sich die Begriffe *Höhe* und *Tiefe* angeben. Während die Höhe  $h$  für einen gesamten Baum definiert ist und den maximalen Abstand eines Blattes zur Wurzel darstellt, bezeichnet die Tiefe (oder Level) den Abstand eines bestimmten Knotens zur Wurzel. Knoten mit gleicher Tiefe fasst man zu *Niveaus* zusammen.

Die Wahl geeigneter Datenstrukturen für ein bestimmtes Problem hängt immer von der Art der Daten und den zu erwartenden Operationen ab. Häufige Operationen sind das *Suchen* nach oder der Zugriff auf einen bestimmten Wert, das *Einfügen*, das *Löschen* von Daten oder das *Durchlaufen* aller Elemente in einer bestimmten Reihenfolge. Bäume können diese Operationen sehr effizient umsetzen. Sowohl die Suche als auch das Einfügen und Löschen weisen meist eine logarithmische Komplexität auf, welche besonders bei großen Datensätzen sehr günstig ist. Bäume bieten sich daher als Datenstruktur an, die einerseits ähnlich wie verkettete Listen dynamische Operationen unterstützen und andererseits zusätzlich einen schnellen (Index-) Zugriff auf bestimmte Elemente bieten können. Bäume werden daher beispielsweise für die Implementierung assoziativer Container zur Speicherung von Schlüssel-Werte-Paaren oder zur Verwaltung von Mengen verwendet, bei denen die Eindeutigkeit der Elemente gewährleistet sein muss [119].

Sehr wichtig für Baumdatenstrukturen ist das Prinzip der Rekursion. Jeder Knoten im Baum lässt sich strukturell immer als Wurzel eines Unterbaumes all seiner nachfolgenden Objekte identifizieren. Die Rekursion erlaubt hier eine sehr elegante Formulierung von Algorithmen. Eine Funktion wird nur für einen Knoten - einer Wurzel - formuliert und ruft sich dann für alle Nachfolger, die ja wiederum Wurzelknoten von Subbäumen sind, einfach selbst auf.

Viele Algorithmen, die auf Baumstrukturen arbeiten, basieren auf einem systematischen Durchlauf aller Elemente, wobei oft gewährleistet sein muss, dass jedes Element genau einmal besucht wird. Die verschiedenen Verfahren dieser *Traversierung* von Bäumen unterscheiden sich durch den Zeitpunkt, wann ein Knoten besucht wird, bevor der Durchlauf mit all seinen Unterbäumen fortgesetzt wird. In *Hauptreihenfolge* oder *Preorder* wird der Knoten gleich zu Beginn bearbeitet, bei einem Durchlauf in *Nebenreihenfolge* oder *Postorder* erfolgt dies erst, nachdem all seine Nachfolger besucht worden sind. Ein Durchlauf in Postorder wird auch als Tiefentraversierung bezeichnet, da diese schnell in die Tiefe läuft. Die Methoden lassen sich sehr schön rekursiv definieren. Der Algorithmus 3.1 zeigt das Gerüst einer Preorder-Traversierung<sup>1</sup>:

---

**Algorithmus 3.1** Traversierung eines Baumes in Preorder- bzw. Hauptreihenfolge

---

```
function: preorder_walk(root)
    visit root
    for all children i
        preorder_walk( child [i] )
    end for
end function
```

---

Wird beispielsweise der Baum aus Abbildung 3.1 in Preorder durchlaufen, ergibt sich die Reihenfolge der Knoten zu

a b d c e f g

und in Postorder zu

d b e f g c a.

### 3.2.2 Quadrees, Octrees

Einer Definition von SAMET [110] folgend, stellen *Quadrees* eine Klasse von Datenstrukturen dar, die auf dem Prinzip der rekursiven Dekomposition des Raumes beruhen. Dabei unterscheidet der Autor verschiedene Typen, die sich in der Art der dargestellten Daten, in den Prinzipien der Zerlegung sowie der Existenz einer bestimmten Auflösung unterscheiden. Quadrees lassen sich für Punktdaten, Rechtecke, Kurven, Regionen oder Oberflächen verwenden. Die Zerlegung erfolgt in der Regel *raumbezogen* nach einem fest vorgegebenen Schema, es gibt aber auch Varianten, die von der Dateneingabe abhängig gesteuert werden. Die Entsprechung für den dreidimensionalen Raum nennt man *Octrees* oder *Oktalbäume*. Zum besseren Verständnis werden die folgenden Beispiele immer für den zweidimensionalen Fall erläutert. Für Oktalbäume verhält es sich aber meist analog. Auf existierende Unterschiede soll an entsprechender Stelle hingewiesen werden.

---

<sup>1</sup>KNUTH [67] liefert hierzu eine schönes Beispiel. Er vergleicht eine Traversierung in Preorder mit der Nachfolgeregelung eines Königshauses: Stirbt der König, geht sein Titel zunächst auf seinen ersten Sohn über und von diesem auf seinen Nachfolger. Falls die Nachfolgerlinie des ersten Sohnes irgendwann ausstirbt, wird der Titel dann an den zweiten Sohn des Königs und dessen Nachkommen weitergegeben.

### 3.2.2.1 Darstellung von Rasterdaten

Eine klassische Verwendung von Quadrees besteht in der Darstellung von Objekten, die in einer Raster- oder Pixeldarstellung<sup>2</sup>, auch Normzellen-Aufzählungsschema genannt, gegeben sind. Das Konstruktionsschema ist dabei wie folgt definiert: Ausgehend von einem Quadrat, das die gesamte Geometrie umschreibt, wird dieses in vier gleiche kongruente Teile unterteilt. Dieser Prozess wird rekursiv weiterverfolgt, bis eine Zelle entweder komplett innerhalb oder außerhalb eines Objekts liegt. SAMET [110] spricht dann von *schwarzen* und *weißen* Zellen. Es ergibt sich eine Baumstruktur, die an den Blättern entweder weiße oder schwarze Zellen besitzt. Die inneren Knoten werden als *grau* bezeichnet. Abbildung 3.2 zeigt den Quadtree einer Region in Rasterdarstellung.

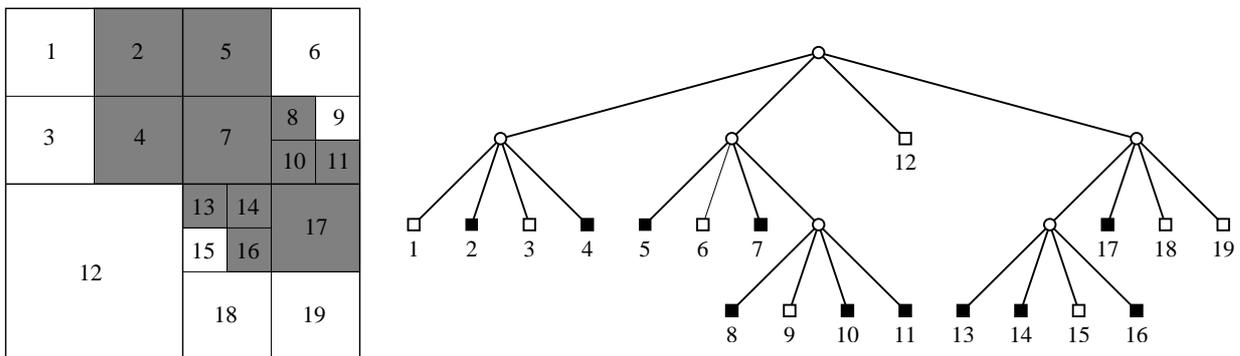


Abbildung 3.2: Darstellung eines zweidimensionalen Körpers mithilfe eines Quadtrees. Links: Einteilung des Körpers (dunkel dargestellt) nach Zerlegung des Raumes in Quadranten; Rechts: Organisation der Elemente im Baum

Komplexitätsuntersuchungen (*Quadtree Complexity Theorem*, [61]) zeigen, dass für die Darstellung von Objekten in einem Raster mit  $2^q \times 2^q$  Zellen ein Baum notwendig ist, dessen Knotenanzahl von der Ordnung  $O(p + q)$  ist. Die Größe  $p$  bezeichnet dabei den Umfang der darzustellenden Objekte in Pixelweiten. Da der Exponent  $q$  vernachlässigt werden kann, ergibt sich, dass die Knotenanzahl weitgehend von der Berandung der darzustellenden Objekte abhängt. Diese Abschätzung gilt auch für Oktalbäume, wobei der Umfang durch die Oberfläche der Objekte ersetzt wird. Allgemein lässt sich hiermit die Aussage treffen, dass sich mithilfe von Quadrees - und ihrer höherdimensionalen Verwandten - der Aufwand eines  $d$ -dimensionalen Problems auf den in der niedrigeren Dimension ( $d - 1$ ) reduzieren lässt. In anderen Worten: Quadrees und Octrees sind Werkzeuge zur Dimensionsreduzierung von Problemen [110].

### 3.2.2.2 Erzeugung von Oktalbüumen aus Modellen in Oberflächendarstellung

Sind die geometrischen Objekte in einer indirekten Darstellung gegeben, zum Beispiel als Körper berandet durch eine polygonale Oberfläche, können Oktalbäume verwendet werden, um das Modell in eine Rasterdarstellung überzuführen. Ein Oktant wird dabei immer dann unterteilt, wenn eine gemeinsame Schnittmenge zwischen der Oberfläche des Körpers und des

<sup>2</sup>Im Dreidimensionalen spricht man von *Voxeldarstellung*.

Oktanten vorliegt. Dieser Prozess wird so lange fortgeführt, bis eine bestimmte Rekursionstiefe erreicht ist. Als zeitkritisch erweist sich bei diesem Verfahren die Schnittberechnung zwischen Oktalbaumzelle und Oberfläche. Solange die Körper von ebenen Flächen berandet sind, kann diese Berechnung relativ schnell ausgeführt werden, schwieriger sind jedoch Objekte mit Freiformoberflächen. Diese werden vor einer Oktalbaumzerlegung meist trianguliert.

Um diesen Aufwand weiter zu reduzieren, geht ein von MUNDANI [79] vorgeschlagenes Verfahren nicht mehr unmittelbar von einer Oberflächendarstellung eines Körpers aus, sondern beschreibt diesen als Serie von Halbräumen, die mittels Boolescher Operationen verknüpft sind. Die Verschneidung zwischen einer den Halbraum begrenzenden Ebene und einer Oktalbaumzelle kann dabei äußerst effizient implementiert werden. Gesteuert von der Sequenz der Booleschen Operationen werden die Oktalbäume aller Halbräume schließlich zu einem gemeinsamen Baum verschnitten, der das entsprechende Objekt repräsentiert. Durch eine Linearisierung der Bäume als Bitstrom lassen sich diese dabei nicht nur sehr platzsparend speichern, sondern können auch bereits während der Generierung in anderen (streamingfähigen<sup>3</sup>) Applikationen weiterverarbeitet werden. Neben der Generierung eines Oktalbaums für ein gesamtes Modell kann dieses Verfahren aber auch dazu genutzt werden, die geometrische Lage zweier Objekte hinsichtlich Überschneidung, Kontakt oder Spalt zu überprüfen. Hierzu werden die Zellen des Oktalbaumes hinsichtlich **Innen** (Inneres eines Körpers), **Rand** (Oberfläche eines Körpers) oder **Aussen** klassifiziert. Eine Anwendung des Booleschen Operators **Durchschnitt** auf zwei als Binärstrom vorliegende Bauteile liefert dann das Ergebnis einer Überprüfung.

Eine Transformation geometrischer Objekte von einem Oberflächenmodell in die direkte Form der Rasterdarstellung wirft jedoch immer die Problematik auf, dass eine Rücktransformation in der Regel nicht mehr eindeutig durchführbar ist. Selbst eine Approximation der ursprünglichen Oberflächen ist meist nur sehr aufwändig zu erreichen. Aus diesem Grunde ist das Vorgehen von MUNDANI, Objekte primär in Oberflächendarstellung zu speichern und nur bei Bedarf eine Oktalbaumdarstellung abzuleiten, sicherlich vorzuziehen, wenn unterschiedliche Darstellungsvarianten von einem Modell ausgehend bedient werden sollen.

### 3.2.2.3 Oktalbäume als hierarchisches Organisationsschema

Neben der Möglichkeit die Geometrie direkt mittels einer Baumstruktur darzustellen, lässt sich ein Baum auch als abstraktes Organisationsschema zur Verwaltung von Objekten im Raum verwenden. Hierzu kann ein geometrischer Körper auf einen oder mehrere eindeutige Eigenschaften reduziert werden, welche als mehrdimensionale Schlüssel innerhalb einer Baumstruktur abgelegt werden. Beispielsweise lässt sich eine Linie durch ihre Endpunkte  $((x_1, y_1), (x_2, y_2))$  darstellen und innerhalb eines Quadtrees speichern (*Edge Quadtree*, vgl. [110]).

Eine andere Möglichkeit besteht darin, das Objekt durch seine minimalen und maximalen Koordinaten zu beschreiben und so in einen Quadtree bzw. Octree einzubetten, dass dieses gerade noch von einer Zelle komplett umschlossen wird (*MX-CIF Quadtree*, vgl. [110], siehe Abbildung 3.3). Objekte werden dabei neben den Blättern auch an inneren Knoten des Baumes abgelegt. Dabei können Knoten auch mehrere Objekte enthalten. Interessant ist diese Art der Organisation insbesondere in Kombination mit dem im vorigen Abschnitt diskutier-

<sup>3</sup>Als Streaming bezeichnet man in der Informatik die kontinuierliche Übertragung von Daten, beispielsweise in Audio- und Videostreamen. Die Daten werden innerhalb einer Applikation gleichzeitig empfangen und verarbeitet, wobei ein Ende des Datenstroms im Voraus nicht absehbar ist.

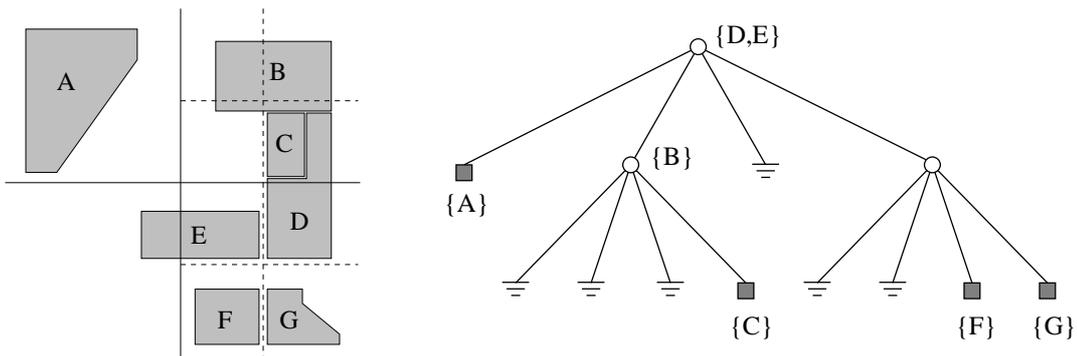


Abbildung 3.3: MX-CIF-Quadtree: Nach einer Quadtree-Zerlegung des Raumes wird ein geometrisches Objekt dem Quadranten zugeordnet, der diese gerade noch komplett umschließt.

ten Verfahren zur Generierung von Oktalbäumen. Bis zu einer gewissen Tiefe des Baumes werden alle Objekte des Modells in Oberflächendarstellung vorgehalten und innerhalb des Baumes hierarchisch verwaltet. Für Aufgaben, die eine zellbasierte Darstellung benötigen, wie beispielsweise eine Kollisionserkennung, lässt sich diese dann sehr schnell ausgehend von dem entsprechenden Oktanten im Baum erzeugen.

Als weitere Möglichkeit zur Speicherung von Objekten bietet es sich an, für jeden Körper einen eindeutigen (geometrischen) Repräsentanten zu bestimmen und diesen einer Zelle des Oktalbaums zuzuordnen. In der Literatur spricht man in diesem Zusammenhang von einem *PR-Quadtree* [110] (P für Punkt, R für Region). Meist wird gefordert, dass nur ein Punkt innerhalb einer Zelle liegen darf. Dies bedeutet, dass bei der Erzeugung des Baumes ein Oktant immer dann unterteilt wird, solange dieser mehr als einen Koordinatenpunkt beinhaltet. Objekte werden dabei ausschließlich an den Blättern des Baumes positioniert, welche auch leer sein können. Abbildung 3.4 zeigt einen PR-Quadtree für eine gegebene Menge von Punkten.

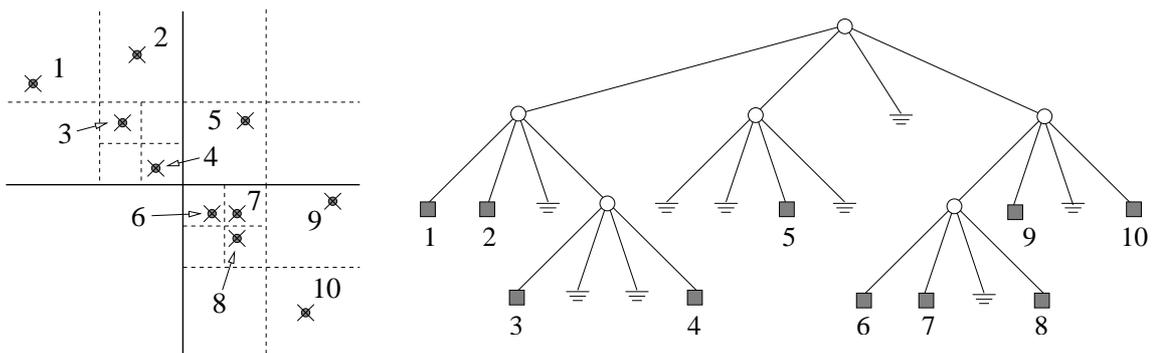


Abbildung 3.4: PR-Quadtree: Punktkoordinaten werden so in einen Quadtree einsortiert, dass maximal ein Punkt innerhalb einer Zelle liegt.

Im allgemeinen Fall ist bei diesem Vorgehen zu beachten, dass die geometrischen Koordinaten eindeutig einem Objekt zugeordnet werden können. Bei der Verwendung des Schwerpunkts beispielsweise ist diese Eindeutigkeit nur bei konvexen Körpern, die sich nicht überschneiden, sichergestellt. Ein Nachteil des PR-Quadtrees (bzw. Octrees) ist zudem, dass die maximale Rekursionstiefe vom minimalen Abstand zweier Punkte abhängen kann. Das heißt, wenn zwei Punkte sehr nahe zusammenliegen, kann es zu einer sehr starken Verfeinerung an der Stelle

kommen. Dieses Problem kann umgangen werden, wenn mehrere Punkte pro Oktant zugelassen werden und die Zerlegung bei einer vordefinierten Rekursionstiefe abgebrochen wird. Ein weiterer Nachteil liegt darin, dass bei dieser Art der Darstellung die räumliche Ausdehnung der Körper nicht vom Oktalbaum erfasst wird. Obwohl zwei Referenzpunkte innerhalb eines gemeinsamen Oktanten liegen, können sich dennoch die zugehörigen Körper über weitere Bereiche des Raumes erstrecken. In geometrischen Operationen kann dies Schwierigkeiten aufwerfen, wenn beispielsweise zu einem gegebenen Objekt alle Körper innerhalb eines bestimmten Abstands gefunden werden sollen. Zwar nimmt die Wahrscheinlichkeit ab, in „entfernten“ Zellen noch potentielle Objekte zu finden, ganz ausgeschlossen werden kann dies jedoch nicht, was bedeutet, dass die Suche immer den ganzen Baum durchlaufen muss. Eine ähnliche Aufgabe in diesem Zusammenhang ist die so genannte *Punktlokation*. Für eine Koordinate im Raum soll festgestellt werden, ob mit einer Menge von Objekten ein Schnittpunkt existiert und gegebenenfalls das entsprechende Objekt zurückgeliefert werden. Im schlechtesten Falle ist hier wiederum der ganze Baum zu durchlaufen. Da hier die Suche beim Finden des Elements aber abgebrochen werden kann, ist im Mittel weniger Aufwand notwendig. Ein von KRAUSE und RANK [69] vorgeschlagenes Verfahren bezieht zusätzlich die Nachbarschaftsinformation in einem Finite-Elemente-Netz mit ein, um eine Punktlokation festzustellen. Ein Quadtree dient dort vor allem, um einen guten Startpunkt für eine lokale Suche zu bestimmen. Während der Aufbau des Baumes eine Komplexität von  $O(n \times \log(n))$  hat, lässt sich eine Suche dann sehr schnell mit  $O(\log(n))$  durchführen.

### 3.2.3 Darstellung von Bäumen

Klassischerweise erfolgt eine rechnergerechte Darstellung von Bäumen unter Verwendung von Zeigern. Dies ist in vielen Fällen die erste Wahl, gerade wenn die Strukturen dynamisch verändert werden müssen oder Algorithmen effizient implementiert werden sollen. Andererseits kann jedoch der zusätzliche Aufwand, die inneren Knoten zu speichern, schnell zu einem erheblichen Speicherverbrauch führen. Aus diesem Grund werden gemeinhin auch weitere Darstellungsarten verwendet, die auf eine Codierung mit Zeigern verzichten.

Erste Möglichkeit stellt einen Baum *sequenziell* in Form einer Traversierung (z.B. Preorder) über alle Knoten dar. Gerade bei Bäumen, die nur binäre Größen an den Blättern speichern, ist diese Darstellung sehr kompakt, da für einen Knoten maximal zwei Bit notwendig sind. Dazu sind drei Symbole „0“, „1“, und „(“ zu reservieren, die weiße, schwarze bzw. graue Knoten repräsentieren. Der Baum aus Abbildung 3.2 lässt sich somit als Zeichenkette in folgender Form darstellen:

((0101(101(10110((1101100

Aus dieser Darstellung kann der ursprüngliche Baum jederzeit wieder rekonstruiert werden, wenn man berücksichtigt, dass bei einem Quadtree die Ordnung jedes inneren Knoten immer 4 ist. Die starke Komprimierung während der Codierung als Binärstrom bietet vor allem dann Vorteile, wenn Bäume auf Hintergrundspeichern abgelegt oder per Netzwerk versendet werden. Auch können verschiedene Operationen direkt auf dem Binärstrom formuliert werden, so zum Beispiel Mengenoperationen wie Verschneidung oder Vereinigung. Dynamische Operationen, wie der Zugriff oder das Löschen von Teilen des Baumes erweisen sich jedoch als schwierig.

Eine zweite Möglichkeit, Baumstrukturen abzubilden, betrachtet diesen als lineare Liste von Knoten, bei der jeder Knoten mittels eines Schlüssels indiziert werden kann. Dieses Vorge-

hen wurde erstmals von MORTON [77] zur Verwaltung von Indizes in einer geografischen Datenbank vorgeschlagen und wird geläufig auch als *Positionscodierung* bezeichnet. Der Schlüssel eines Blattes ergibt sich dabei während eines Baumabstiegs von der Wurzel zu dem jeweiligen Knoten, aus einer Konkatenation von Ziffern aus dem Zeichenvorrat  $\{0,1,2,3\}$ , die an jeder Ebene des Baumes entsprechend der besuchten Richtung  $\{NW,NO,SW,SO\}$  vergeben werden. Abbildung 3.5 zeigt einen Baum mit entsprechenden Positionscodierungen.

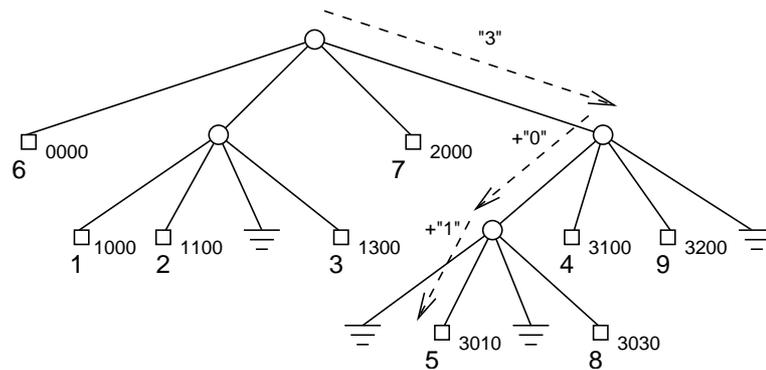


Abbildung 3.5: Positionscodierung der Blattknoten eines Quadtrees. Beispielsweise ergibt sich der Index des Knotens mit der Nummer 5 aus der Konkatenation der Ziffern „3“, „0“ und „1“. Die Indizes sind als Zahl zur Basis 4 dargestellt. Es ist zu beachten, dass diese während des Abstiegs von links aufgefüllt werden.

Diesem Verfahren, einen Baum in einer linearen Weise zu speichern, kommt in dieser Arbeit eine besondere Bedeutung zu. Vorteile bieten sich, wenn bestehende Programme um hierarchische Strukturen erweitert werden sollen, ohne dass dafür eine explizite Zeigerdarstellung implementiert werden soll. Im Falle der vorliegenden Arbeit wird eine Positionscodierung verwendet, um Elemente innerhalb eines Finite-Elemente-Programms hierarchisch zu organisieren. Im Unterschied zur Codierung gemäß Abbildung 3.5 müssen in dieser Arbeit auch innere Knoten eindeutig indiziert werden können. Aus diesem Grund wird eine Ergänzung vorgeschlagen, bei der der Index zusätzlich um die Tiefe des Knotens erweitert wird. Prinzipiell ist der Index dann als Nummernpaar zu verwalten. Um jedoch die Indizierung im Programm zu vereinfachen, kann der Positionscodierung sowie die Tiefe innerhalb einer einzigen Zahl dargestellt werden. Zur Speicherung der Tiefe werden dazu am „rechten“ Ende der Zahl mehrere Bits reserviert. In den hier vorgestellten Beispielen sind 5 Bits für darstellbare Tiefen bis 32 völlig ausreichend. Unter Verwendung einer 64 Bit Integer-Variablen für den gesamten Index können damit insgesamt  $(64 - 5)/3 \geq 19$  Level dargestellt werden. Abbildung 3.6 zeigt am Beispiel des Baumes in Bild 3.5 diese Art der Darstellung.

Eine derartige Codierung mithilfe von Positionscodes hat nun die interessante Eigenschaft, dass eine Sortierung aller Knoten nach aufsteigenden Indizes exakt einer Preorder-Traversierung des Baumes entspricht. Weiterhin ist nach einer Sortierung jeder Subbaum in einem zusammenhängenden Intervall innerhalb der Liste abgelegt. Beispielsweise befinden sich in Abbildung 3.6 alle Elemente des Subbaumes mit der Wurzel 1000|1 innerhalb des Intervalls  $[1000|1; 2000|0[$ . Damit können auch bei einer linearen Anordnung des Baumes – also etwa in einem Finite-Elemente-Programm – Teilbäume schnell identifiziert und bearbeitet werden.

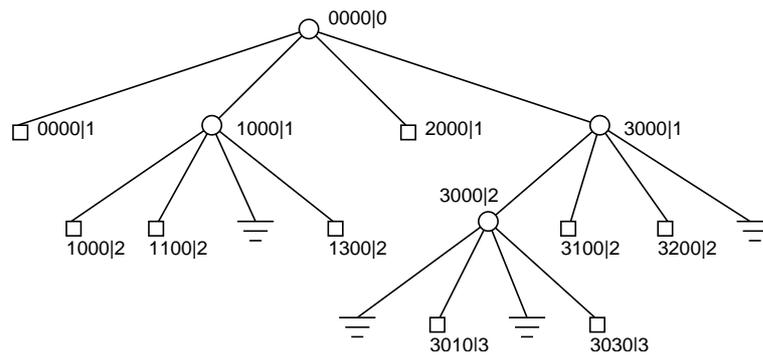


Abbildung 3.6: Indizierung zusätzlich mit inneren Knoten unter Verwendung eines kombinierten Nummernpaars aus Positionscode und Tiefe.

### 3.3 Das Prinzip der rekursiven Substrukturierung

Das Prinzip, ein gegebenes Problem dadurch zu lösen, indem es in einfachere Teilprobleme unterteilt wird, spielt in vielen Aufgaben der Informatik sowie den Ingenieurwissenschaften eine wichtige Rolle. In der Informatik spricht man in diesem Zusammenhang von Verfahren, die nach dem Prinzip *Teile und Herrsche* arbeiten. Im Bereich der Strukturmechanik wird eine ähnliche Vorgehensweise unter dem Begriff der *Substrukturierung* seit den 60er Jahren des vergangenen Jahrhunderts angewandt. Aufgrund der Tatsache, dass in der Strukturmechanik die Modelle oft aus einer Vielzahl unterschiedlicher mechanischer Systeme zusammengesetzt werden, lag es nahe, eine Unterteilung hinsichtlich dieser Systeme vorzunehmen [92, 93]. Ein weiterer Grund für die damals sehr beliebte Anwendung des Verfahrens lag in den begrenzten Kapazitäten der Rechnersysteme. Eine Unterteilung der Strukturen war oft die einzige Möglichkeit, größere Modelle rechnen zu können.

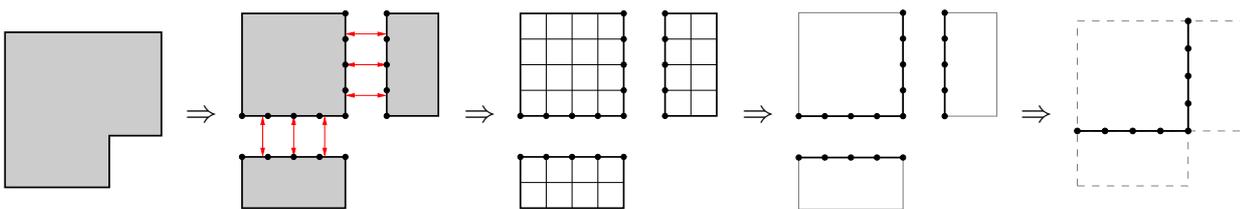


Abbildung 3.7: Vorgehen in der Substrukturierung: Zerlegung eines Gebiets; Vernetzung der Teilstrukturen, Elimination der inneren Freiheitsgrade, Assemblierung der Separatorgrößen.

Abbildung 3.7 zeigt das prinzipielle Vorgehen bei der Substrukturierung. Zunächst wird das Berechnungsgebiet  $\Omega$  unter Einführung von *Separatoren* in eine Anzahl nichtüberlappender Teilgebiete  $\Omega_i$  unterteilt. Gemäß der Finite-Elemente-Methode wird jedes Teilgebiet diskretisiert und ein lokales Gleichungssystem aufgestellt. Mittels statischer Kondensation werden die inneren Freiheitsgrade jeder Teilstruktur eliminiert und die strukturellen Eigenschaften auf die Separatorvariablen übertragen. Das globale Gleichungssystem bestimmt sich nun nur noch aus den Steifigkeiten an den Separatoren. Nach Lösung dieses globalen Problems und Bestimmung

der Separatorwerte können in einem letzten Schritt schließlich die Unbekannten im Inneren der Teilgebiete berechnet werden.

Führt man eine Zerlegung des Systems nicht nur einmal durch, sondern unterteilt man die Substrukturen wiederum in weitere kleinere Einheiten, so spricht man von einer Mehrebenen-Substrukturtechnik [87, 18] oder, vor allem wenn die Prinzipien der Zerlegung immer nach dem gleichen Schema ablaufen, auch von einer *rekursiven Substrukturierung* [63, 62, 114]. Es ergibt sich eine baumartige Anordnung der Teilgebiete, bei dem jeder Knoten das Gebiet all seiner nachfolgenden Substrukturen repräsentiert (Abbildung 3.8). Das gesamte Gebiet stellt sich nur noch in der Wurzel des Baumes dar.

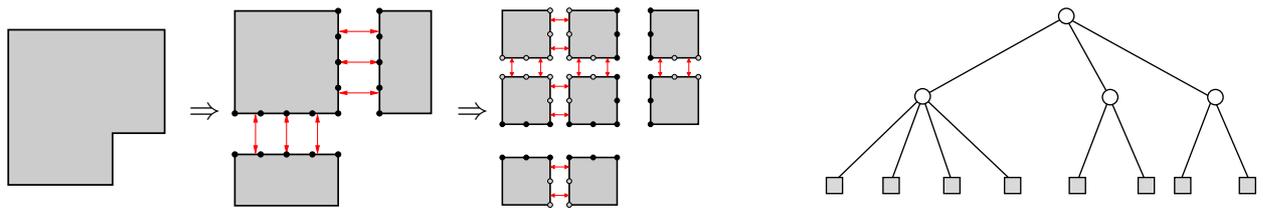


Abbildung 3.8: Rekursive Substrukturierung eines Gebiets. Rechts: Substrukturhierarchie.

Ein wesentlicher Unterschied zur üblichen Vorgehensweise in der Finite-Elemente-Methode besteht bei der rekursiven Substrukturierung darin, dass an keiner Stelle das Problem mehr als Ganzes betrachtet werden muss. Das globale System in der Wurzel des Baumes ist nur noch über die Größen an den Separatoren der ersten Zerteilungsebene gegeben. Alle Aufgaben in den Teilgebieten laufen entkoppelt voneinander auf den verschiedenen Ebenen des Baumes ab. Diese wesentlich lokalere Sichtweise der rekursiven Substrukturierung erweist sich gerade für Aufgaben der Verteilung, sei es hinsichtlich einer physikalischen Verteilung in einem Rechnernetz aber auch, um eine organisatorische Verteilung in einer kooperativen Arbeitsumgebung zu organisieren, als sehr hilfreich.

### 3.3.1 Statische Kondensation und Lösung der FE-Gleichungen

Für die Zerlegung der Struktur lassen sich unterschiedliche Prinzipien anwenden. Die Struktur kann nach Bauteilen oder hinsichtlich verschiedener mechanischer Modelle unterteilt werden. Es können aber auch streng hierarchische Verfahren, wie zum Beispiel Oktalbäume zur Anwendung kommen. Am Beispiel der Oktalbäume und der rekursiven Bisektion sollen in dieser Arbeit zwei Prinzipien der Zerlegung genauer untersucht werden. Unabhängig davon läuft die Lösung des Gesamtgleichungssystems aber immer nach dem gleichen Schema ab. Ausgangspunkt ist die statische Kondensation von inneren Freiheitsgraden jeder Substruktur. Für jede dieser Teilgebiete kann ein lokales Gleichungssystem der Form

$$Ku = f \quad (3.1)$$

aufgestellt werden. Die Spalten und Zeilen des Systems lassen sich hinsichtlich der Freiheitsgrade im Inneren  $\Omega_I$  und denjenigen auf dem Separatorrand  $\Gamma_A$  sortieren. Man erhält ein Gleichungssystem der Form

$$\begin{pmatrix} K_{ii} & K_{ia} \\ K_{ai} & K_{aa} \end{pmatrix} \begin{pmatrix} u_i \\ u_a \end{pmatrix} = \begin{pmatrix} f_i \\ f_a \end{pmatrix}, \quad (3.2)$$

welches in zwei gekoppelte Teilgleichungssysteme zerlegt werden kann:

$$K_{ii} u_i + K_{ia} u_a = f_i \quad (3.3)$$

$$K_{ai} u_i + K_{aa} u_a = f_a. \quad (3.4)$$

Durch Umformung von (3.3)

$$u_i = K_{ii}^{-1} f_i - K_{ii}^{-1} K_{ia} u_a, \quad (3.5)$$

und Einsetzen in (3.4) können die inneren Unbekannten eliminiert werden und man erhält ein Gleichungssystem

$$(K_{aa} - K_{ai} K_{ii}^{-1} K_{ia}) u_a = f_a - K_{ai} K_{ii}^{-1} f_i, \quad (3.6)$$

welches die Substruktur nur noch in Abhängigkeit der äußeren Unbekannten  $u_a$  beschreibt. Gleichung (3.6) lässt sich in kurzer Form darstellen:

$$\tilde{K}_a u_a = \tilde{f}_a. \quad (3.7)$$

Der Ausdruck  $\tilde{K}_a | \tilde{f}_a$  wird als SCHUR-Komplement des Gleichungssystems in (3.2) bezeichnet. Es beschreibt das elastische Verhalten einer Struktur nur noch durch seine äußeren Freiheitsgrade.

In der rekursiven Substrukturierung wird zur Lösung des Gesamtsystems das SCHUR-Komplement in (3.7) zur weiteren Verarbeitung an den nächsthöheren Level in der Hierarchie gereicht. Dieses wird mit allen Ergebnissen aus den anderen Sohngebieten zu einem neuen Gleichungssystem assembliert, welches nun wiederum als lokales Problem dieser Hierarchieebene aufgefasst und ausgehend von Gleichung (3.1) mit einer erneuten statischen Kondensierung fortgefahren wird. Als äußere Freiheitsgrade gelten nun diejenigen auf den Schnitten zur nächst höheren und aller weiteren Hierarchieebenen. Eine sukzessive Wiederholung dieser Schritte - statische Kondensation und Assemblierung der Ergebnisse aus den Sohngebieten - bis hin zum obersten Level liefert ein Gleichungssystem an der Wurzel des Baumes, welches nun die gesamte Struktur repräsentiert. Nach Lösung des globalen Problems sind die Verschiebungsgrößen  $u_a$  an den Separatoren der obersten Ebene bekannt und es können mithilfe von Gleichung (3.5) in einem *top-down* Prozess alle lokalen Unbekannten rekursiv für alle Teilgebiete bestimmt werden.

### 3.3.1.1 Berechnung des SCHUR-Komplements

Der weitaus größte Aufwand während der Assemblierungsphase resultiert aus der Berechnung des SCHUR-Komplements  $\tilde{K}_a$ . Um die hierfür notwendige explizite Invertierung von  $K_{ii}^{-1}$  in Gleichung (3.6) zu vermeiden, kann das SCHUR-Komplement beispielsweise mithilfe einer partiellen GAUSS-Zerlegung bestimmt werden. Dazu wird nach Sortierung der inneren und äußeren Freiheitsgrade auf dem Gleichungssystem in (3.2) eine Zerlegung nach GAUSS durchgeführt, diese aber nach Elimination aller inneren Freiheitsgrade abgebrochen. Als Resultat ergibt sich ein Gleichungssystem

$$\begin{pmatrix} R_{ii} & \tilde{K}_{ia} \\ 0 & \tilde{K}_a \end{pmatrix} \begin{pmatrix} u_i \\ u_a \end{pmatrix} = \begin{pmatrix} \tilde{f}_i \\ \tilde{f}_a \end{pmatrix}, \quad (3.8)$$

mit einer rechten, oberen Dreiecksmatrix  $R_{ii}$  und in welchem der untere Teil von den inneren Unbekannten  $u_i$  entkoppelt ist. Dies setzt voraus, dass  $K_{ii}$  aus Gleichung (3.2) positiv definit ist und keine Zeilenvertauschungen während der Zerlegung vorzunehmen sind. Da der untere Teil nur noch von den äußeren Unbekannten abhängig ist, entspricht dieser dem SCHUR-Komplement aus (3.7). Nach Berechnung der Separatorgrößen  $u_a$  lassen sich die inneren Unbekannten  $u_i$  durch Lösung (Rückwärtssubstitution) von

$$R_{ii}u_i = \tilde{f}_i - \tilde{K}_{ia}u_a \quad (3.9)$$

bestimmen. Die Operationen der GAUSS-Elimination laufen in der Regel zeilen- bzw. spaltenweise über die Steifigkeitsmatrix. Hardwareeffiziente Bibliotheken wie LAPACK<sup>4</sup> [5] und BLAS<sup>5</sup> [30] können jedoch ihre maximale Leistungsfähigkeit erst dann vollständig entwickeln, wenn die zugrunde liegenden Algorithmen als Matrix-Matrix Operationen (Level 3 BLAS) ausgeführt werden. Um die Berechnung des SCHUR-Komplements für die Anwendung von LAPACK und BLAS zu optimieren, soll einem Vorgehen nach SCHWARZ und KÖCKLER [117] folgend, dazu die partielle GAUSS-Elimination der inneren Freiheitsgrade in Form von Blockmatrix Operationen formuliert werden. Abbildung 3.9 symbolisiert dazu die betrachteten Blöcke innerhalb der Steifigkeitsmatrix sowie des Lastvektors vor und nach der partiellen Zerlegung:



Abbildung 3.9: Block-Elimination der inneren Freiheitsgrade.

Zur Bestimmung der notwendigen Block-Matrix Operationen wird das lokale Gleichungssystem in (3.2) dazu zunächst formal in ein Produkt einer linken-unteren und einer rechten-oberen Dreiecksmatrix zerlegt:

$$\begin{aligned} \begin{pmatrix} K_{ii} & K_{ia} \\ K_{ai} & K_{aa} \end{pmatrix} &= \begin{pmatrix} L_{ii} & 0 \\ \tilde{K}_{ai} & L_{aa} \end{pmatrix} \begin{pmatrix} R_{ii} & \tilde{K}_{ia} \\ 0 & R_{aa} \end{pmatrix} \\ &= \begin{pmatrix} L_{ii}R_{ii} & L_{ii}\tilde{K}_{ia} \\ \tilde{K}_{ai}R_{ii} & \tilde{K}_{ai}\tilde{K}_{ia} + L_{aa}R_{aa} \end{pmatrix}. \end{aligned} \quad (3.10)$$

Wird diese  $LR$ -Zerlegung nun partiell nur für die inneren Freiheitsgrade durchgeführt, bleibt in der unteren, rechten Submatrix in (3.10) das Produkt  $L_{aa}R_{aa}$  bestehen, welches dem SCHUR-Komplement  $\tilde{K}_a$  entspricht. Unter Verwendung der Beziehungen in (3.10) lässt sich nun eine partielle GAUSS-Elimination mithilfe von vier Blockmatrix-Operationen durchführen:

1. Zerlege:  $K_{ii} = L_{ii}R_{ii} \rightarrow$  speichere auf  $K_{ii}$
2. Löse:  $R_{ii}^T \tilde{K}_{ai}^T = K_{ai}^T \rightarrow$  speichere  $\tilde{K}_{ai}$  auf  $K_{ai}$

<sup>4</sup>LAPACK – Linear Algebra PACKage: <http://www.netlib.org/lapack/>

<sup>5</sup>BLAS – Basic Linear Algebra Subprograms: <http://www.netlib.org/blas/>

3. Löse:  $L_{ii}\tilde{K}_{ia} = K_{ia} \rightarrow$  speichere  $\tilde{K}_{ia}$  auf  $K_{ia}$
4. Berechne:  $\tilde{K}_a = K_{aa} - \tilde{K}_{ai}\tilde{K}_{ia} \rightarrow$  speichere  $\tilde{K}_a$  auf  $K_{aa}$

Geht man nun von einer symmetrisch, positiv definiten Steifigkeitsmatrix aus, lässt sich unter Punkt 1 eine CHOLESKY-Zerlegung  $K_{ii} = L_{ii}L_{ii}^T$  durchführen (vgl. [117]). Weiterhin entspricht bei einer symmetrischen Matrix  $K_{ai}^T = K_{ia}$ . Damit sind die beiden Lösungsschritte unter Punkt 2 und 3 identisch und es gilt  $\tilde{K}_{ai}^T = \tilde{K}_{ia}$ . Die Berechnung des SCHUR-Komplements sowie der zugehörigen Lastvektoren für symmetrisch positiv definite Matrizen kann somit auf drei Schritte reduziert werden:

1. Zerlege:  $K_{ii} = L_{ii}L_{ii}^T$
2. Löse:  $L_{ii}\tilde{K}_{ia} = K_{ia}$  (bzw.  $L_{ii}\tilde{f}_i = f_i$ )
3. Berechne:  $\tilde{K}_a = K_{aa} - \tilde{K}_{ia}^T\tilde{K}_{ia}$  (bzw.  $\tilde{f}_a = f_a - \tilde{K}_{ia}^T\tilde{f}_i$ )

Aufgrund der Symmetrie können diese drei Schritte ausschließlich auf einer Dreiecksmatrix ausgeführt werden und die Anzahl an Operationen verringert sich etwa um die Hälfte im Vergleich zur  $LR$ -Zerlegung. In dieser Arbeit wurden diese Schritte unter Verwendung von zwei LAPACK- und einer BLAS-Routine durchgeführt. Wie später in den Beispielen gezeigt wird, sind damit für die gesamte Assemblierungsphase Leistungen möglich, die annähernd 50 Prozent der theoretisch möglichen Leistung des verwendeten Rechners betragen. Die Berechnung der inneren Unbekannten während der top-down Lösungsphase erfolgt analog zu Gleichung (3.9) durch Lösung von:

$$L_{ii}^T u_i = \tilde{f}_i - \tilde{K}_{ia} u_a. \quad (3.11)$$

### 3.3.2 Diskussion

#### 3.3.2.1 Iterative Lösungsverfahren

Das hier vorgestellte Verfahren der Substrukturierung mittels statischer Kondensation auf lokaler Ebene und Assemblierung der SCHUR-Komplemente aller Teilgebiete stellt den Standardansatz der nichtüberlappenden Methoden der Gebietszerlegung (Domain Decomposition) dar [68]. Dies sind Verfahren zur Lösung von Gleichungssystemen, die durch Zerlegung des Berechnungsgebiets eine Anwendung für massiv parallele Rechensysteme ermöglichen sollen. Bei modernen Methoden der Gebietszerlegung werden die SCHUR-Komplemente allerdings nicht mehr auf globaler Ebene assembliert und das entstehende Gleichungssystem direkt gelöst. Dies würde insbesondere bei großen Problemen einen zu hohen Aufwand bedeuten. An dieser Stelle kommen meist iterative Verfahren zur Anwendung, deren Operationen zudem weitgehend parallel auf den Teilgebieten ausgeführt werden. Diese werden auch als iterative Substrukturmethoden bezeichnet. Im Gegensatz zu Verfahren der Domain-Decomposition findet in der rekursiven Substrukturierung allerdings eine Zerlegung des Gebiets in mehreren Ebenen statt. Durch diese mehrstufige Zerteilung ist die Problematik zur Lösung des globalen Gleichungssystems entschärft.

Ein iteratives Lösungsverfahren für die rekursive Substrukturierung wurde von HÜTTL [62] untersucht. Im Gegensatz zur direkten Lösung der Gleichungen bleiben bei einer iterativen Vorgehensweise die Matrizen an den oberen Knoten dünnbesiedelt, was sich günstig auf den

Speicher und Rechenzeitbedarf des Löser auswirkt. HÜTTL wies für das iterative Verfahren einen Speicherbedarf von  $O(n)$  und einen Aufwand an arithmetischen Operationen von  $O(n \cdot \log(n)^2)$  nach. Der Nachweis fand jedoch an einem ebenen, quadratischen Modellproblem mit regelmässigem Netz statt, was die Übertragbarkeit auf reale Probleme einschränkt.

Eine grundsätzliche Problematik iterativer Lösungsverfahren ist deren Abhängigkeit von der Kondition des Problems. Die Kondition ist ein Maß in der numerischen Mathematik, welches die Abhängigkeit der Lösung eines Problems von einer Störung seiner Eingangsdaten beschreibt. Die Kondition eines Problems ist dabei unabhängig von der Art des angewendeten Lösungsverfahrens. Bei iterativen Lösungsverfahren können schlecht konditionierte Systeme eine schwache Konvergenz zur Folge haben, so dass die Rechenzeit erheblich ansteigen kann. Durch Wahl geeigneter Vorkonditionierungsmethoden kann die Konvergenz meist erheblich verbessert werden, jedoch ist der Erfolg einer Methode stark von der Art des betrachteten Problems abhängig.

Besonders bei den in dieser Arbeit betrachteten Strukturen mit gemischt dick- und dünnwandigen Elementen sind die entstehenden Gleichungssysteme meist sehr schlecht konditioniert. Dies liegt zu einem Teil an den höchst unterschiedlichen Steifigkeitsverhältnissen innerhalb der Strukturen. Infolge der damit schlechten Kondition des Systems, sind iterative Löser nur mit sehr aufwändigen Vorkonditionierungsmaßnahmen zu verwenden. WALL et al. [129] schlagen hierzu vor, bei plattenartigen Strukturen eine Skalierung des Direktors in Dickenrichtung vorzunehmen. Für die in dieser Arbeit betrachteten Probleme mit dünnwandigen Volumenelementen der  $p$ -Version stellt MANDEL in [72] einen speziellen Vorkonditionierer vor. Er macht sich dabei die Hierarchie der Ansatzfunktionen zu Nutze und spaltet den Finite-Elemente-Ansatzraum in Unterräume auf, die dann adaptiv unter Schätzung der jeweiligen Kondition angepasst werden. Der Löser zeigt gerade bei großen realitätsnahen Problemen sehr gute Ergebnisse. Die Schwierigkeiten, die aus dünnwandigen Strukturen resultierenden Gleichungssysteme mittels iterativer Verfahren zu lösen, sind auch innerhalb unserer Arbeitsgruppe beobachtet worden [35, 83]. Allgemein ist festzustellen, dass für derartige Problemstellungen bevorzugt direkte Löser eingesetzt werden. Dies gilt in ähnlicher Weise auch für Probleme der  $hp$ -Version der FEM. Dort kommen vor allem zur Parallelisierung direkte Lösungsverfahren zur Anwendung [91].

### 3.3.2.2 Direkte Lösungsverfahren für schwach besetzte Gleichungssysteme

Diskretisierungen mit der Finite-Elemente-Methode resultieren meistens in Gleichungssystemen mit *schwach besetzten* Matrizen, in denen die weitaus größte Anzahl an Einträgen Null ist. Lösungsverfahren nutzen in der Regel diese Eigenschaft aus, um sowohl den Berechnungsaufwand als auch den benötigten Speicherplatz zu reduzieren. Das Problem bei schwach besetzten Gleichungssystemen besteht darin, dass während der Faktorisierung des Systems - meist basieren diese auf einer  $LU$ -Zerlegung [117] - viele dieser Nulleinträge mit Werten aufgefüllt werden. Bei der Addition eines Vielfachen einer Zeile zu einer anderen können bestehende Nulleinträge mit Werten überschrieben werden, die für den weiteren Verlauf berücksichtigt werden müssen und damit einen Mehraufwand an numerischen Operationen sowie einen höheren Speicherverbrauch verursachen.

Ziel direkter Lösungsverfahren ist es, dieses Auffüllen von Nulleinträgen, den so genannten *Fill-In*, so gut wie möglich zu reduzieren. Dabei existieren verschiedene Strategien. Ein in der Finite-Elemente-Methode klassisches Vorgehen besteht darin, die Nummerierung der Knoten

im Netz so zu wählen, dass die Einträge in der Steifigkeitsmatrix eine Bandstruktur annehmen [25]. Bei der Zerlegung werden dann Nulleinträge nur innerhalb des Bandes überschrieben. Zudem kann eine derartige Matrix auch sehr platzsparend im Arbeitsspeicher des Rechners angelegt werden.

Das grundlegende Prinzip zur Lösung schwach besetzter Gleichungssystemen besteht darin, die Reihenfolge der Elimination so zu ändern, dass sich ein möglichst geringer Fill-In ergibt. Moderne Verfahren sind dabei in der Regel nicht an das Vorhandensein eines konkreten Berechnungsgebiets gebunden, sondern können auf allgemeine Gleichungssysteme angewendet werden. Nach DUFF et al. [33] erfolgt die Lösung in drei Schritten: *Analyse* des Gleichungssystems, *Faktorisierung* der Matrix (z.B. *LU*-Zerlegung) und *Lösung*. Dem ersten Schritt kommt eine wesentliche Bedeutung zu, da hier die Eliminationsreihenfolge festgelegt wird. Die zugrunde liegenden Algorithmen basieren meist auf graphentheoretischen Ansätzen, bei denen die Belegungsstruktur der Lösungsmatrix als Graph interpretiert wird, dessen Anzahl an Knoten der Dimension der Matrix entspricht. Zwischen zwei Knoten  $i, j$  des Graphen wird eine Kante  $e(i, j)$  eingetragen, wenn an der entsprechenden Stelle der Matrix ein Eintrag ungleich Null existiert, wenn also gilt:  $a_{ij} \neq 0$ . Im Falle symmetrischer Matrizen ( $a_{ij} = a_{ji}$ ) erhält man einen ungerichteten Graphen.

Zur Lösung symmetrisch, positiv definiten Gleichungssysteme lassen sich zwei wichtige Strategien zur Bestimmung der Eliminationsreihenfolge unterscheiden: *Minimum-Degree-Ordering* und *Nested-Dissection*. Grundlage beider Verfahren ist der einer Matrix assoziierte Graph  $G$ . Bei ersterem Verfahren, der Minimum-Degree-Ordering, bestimmt sich die Reihenfolge der Elimination aus dem *Grad* der Knoten in  $G$ . Der Grad  $deg(v)$  (auch: Valenz) eines Knotens beschreibt die Anzahl an Kanten, die mit diesem Knoten verbunden sind [90]. Bei gerichteten Graphen unterscheidet man zudem zwischen einem Ausgangs- und einem Eingangsgrad. Beim Verfahren der Minimum-Degree wird nun schrittweise immer der Knoten mit minimalem Grad in  $G$  gesucht. Dieser wird als Nächstes in der Eliminationsreihenfolge festgelegt und aus  $G$  entfernt. Auf dem immer weiter reduzierten Graphen wird nach diesem Schema so lange fortgefahren, bis für alle Knoten die Reihenfolge der Elimination feststeht. Mit dem Knotengrad nutzt das Verfahren dabei eine *lokale* Eigenschaft eines Graphen.

Im Gegensatz dazu wählt das Verfahren der Nested-Dissection eine *globale* Herangehensweise, um den Fill-In während der Zerlegung zu reduzieren. Dieses basiert auf einer rekursiven Unterteilung eines Berechnungsgebiets und wurde erstmalig im Jahre 1973 von GEORGE [51] vorgeschlagen. Der Autor betrachtet dazu ein Finite-Elemente-Problem auf einem Einheitsquadrat  $(0, 1) \times (0, 1)$ , welches strukturiert in  $n \times n$  Elemente unterteilt ist. Diesem Gebiet wird ein symmetrisch positiv definites Gleichungssystem mit  $(n + 1)^2$  Unbekannten entsprechend den Knoten im Netz assoziiert. Zur Bestimmung einer Eliminationsreihenfolge wird auf dem Berechnungsgitter ein so genannter *Separator*, eine Menge von Knoten, so bestimmt, dass nach Wegnahme dieser Knoten das Gebiet in vier gleiche, unabhängige Teilgebiete zerfällt. Dieser Prozess wird auf jedes der Teilgebiete rekursiv weiter angewandt, so lange bis keine weitere sinnvolle Zerteilung mehr möglich ist. Die Nummerierung der Knoten und damit die Reihenfolge der Elimination bestimmt sich nun in umgekehrter Reihenfolge zur Zerlegung. Beginnend an den Blättern der Hierarchie werden zunächst Knoten im Inneren der Teilgebiete immer zuerst vor denjenigen auf dem Separator (und allen weiteren Separatoren innerhalb der Hierarchie) nummeriert. Knoten auf dem ersten 'Schnitt' bekommen die höchsten Nummern und werden demnach als Letztes eliminiert.

Am Beispiel dieses ebenen quadratischen Problems mit  $(n + 1) \times (n + 1)$  Knoten wurde

von GEORGE für den Aufwand an Gleitpunktoperationen eine Ordnung von  $O(n^3)$  ermittelt. Damit hat der Autor erstmals nachgewiesen, dass sich ein Gleichungssystem mit  $N = (n + 1)^2$  Unbekannten mit nur  $O(N^{3/2})$  Gleitpunktoperationen direkt lösen lässt. Für das betrachtete Modellproblem wurde zudem gezeigt, dass die damit erzielte Lösungskomplexität in Bezug auf die Anzahl an Gleitkommaoperationen sowie Fill-In das bestmögliche Ergebnis für ein direktes Verfahren darstellt [32]. In einem weiteren Beitrag von GEORGE und LIU [52] wurde das Verfahren der Nested-Dissection auf Probleme mit unstrukturierten Netzen erweitert. Allerdings gestaltet sich die Lösung bei unstrukturierten Netzen wesentlich schwieriger. Erst in neuerer Zeit setzen sich Verfahren der Nested-Dissection gegenüber beispielsweise einer Minimum-Degree Sortierung zur direkten Lösung schwach besetzter Gleichungssysteme durch [10, 56]. Die Separatoren werden dabei nicht mehr direkt auf dem Berechnungsnetz festgelegt, sondern auf dem Graphen der Matrix. Die wesentlichen Algorithmen, die hierbei zur Anwendung kommen, orientieren sich an Verfahren der Graphenpartitionierung [66]. Üblicherweise wird hierzu eine rekursive Bisektion, d.h. eine Zerteilung des Graphen in zwei möglichst gleich große Partitionen auf jedem Level, durchgeführt. Die Eliminationsreihenfolge ergibt sich wie vorher, beginnend von den Blättern der Hierarchie werden immer zuerst innere Knoten vor den äußeren eliminiert. Im Falle allgemeiner Graphen, die nicht an ein spezielles Berechnungsgebiet gekoppelt sind, spricht man in dem Zusammenhang auch von *Generalized Nested-Dissection*. Allerdings ist die klassische Vorgehensweise in der Nested-Dissection besonders für komplexe Beispiele nicht immer optimal [10, 56]. Aus diesem Grunde wird in der vorliegenden Arbeit im Rahmen der rekursiven Substrukturierung eine modifizierte Variante angewendet. Eine Diskussion hierzu findet sich in Abschnitt 3.4.1.2.

### 3.3.2.3 Substrukturtechniken

Das Prinzip der rekursiven Substrukturierung entspricht in vielen Aspekten dem Vorgehen in der Nested-Dissection, wobei im Gegensatz zur Substrukturierung bei Letzterem in der Regel keine explizite Zerlegung des Berechnungsgebiets erfolgt. Es lassen sich aber in beiden Verfahren die gleichen Prinzipien der Unterteilung anwenden.

Auf die Zusammenhänge zwischen Substrukturierung und Methoden zur Lösung schwach besetzter Gleichungssysteme wurde schon früh in der Literatur hingewiesen [131, 87]. In [87] wird beispielsweise gezeigt, dass in einer Mehrebenen-Substrukturtechnik und der Nested-Dissection der gleiche Aufwand an numerischen Operationen benötigt wird, wenn in der Substrukturierung die Elimination nach dem gleichen Schema erfolgt, also beginnend an den Blättern immer zuerst interne Freiheitsgrade vor denjenigen auf dem Separator eliminiert werden. In der Arbeit von SCHNEIDER [114] wurde der numerische Aufwand in der rekursiven Substrukturierung detailliert untersucht und unter anderem auch mit dem Verfahren der Nested-Dissection verglichen. Dabei ergibt sich - am Beispiel eines ebenen strukturierten Netzes mit einer binären Zerteilungsstrategie - für den Aufwand an Gleitkommaoperationen in der Substrukturierung mit  $O \leq 12.28 \cdot n^{1.5}$  die gleiche Ordnung wie in der Nested-Dissection nach [51] ( $O < 10 \cdot n^{1.5}$ ).

Darüber hinaus bietet aber die Substrukturtechnik durch die explizite Zerlegung weitere Vorteile, die sich in allgemeinen Lösungsverfahren für schwach besetzte Gleichungssysteme nicht ohne weiteres ausnützen lassen. Beispielsweise lässt sich der Berechnungsaufwand reduzieren, wenn Teile innerhalb von Strukturen identisch sind. Es muss die Steifigkeit nur einmal berechnet und die inneren Freiheitsgrade kondensiert werden. Diese einmal berechneten Sepa-

ratorsteifigkeiten lassen sich dann - auch unter Zuhilfenahme von Transformationsmatrizen - zu komplexeren Strukturen zusammensetzen [63]. Eine weitere Verringerung des numerischen Aufwands lässt sich erreichen, wenn auf den globalen Separatoren Freiheitsgrade durch kinematische Kopplungen eliminiert werden [87]. Dadurch verkleinert sich die Größe des globalen Gleichungssystems und es ist an dieser Stelle weniger Berechnungsaufwand notwendig. Die richtige Anbringung der kinematischen Abhängigkeiten erfordert jedoch ein hohes Maß an Erfahrung.

Weitere Vorteile in der Substrukturtechnik ergeben sich hinsichtlich einer Verteilung von Daten und Prozessen innerhalb eines Netzwerks, aber auch durch die Möglichkeit einer schnelleren Neuberechnung nach lokalen Modifikationen der Struktur, da in diesen Fällen nur Teilgebiete des Substrukturbaumes in den Berechnungsprozess mit einbezogen werden müssen. Besonders dieser Punkt wird Gegenstand genauerer Untersuchungen in Abschnitt 3.5 dieser Arbeit sein.

### 3.4 Substrukturierung in der $p$ -Version der FEM

Inhalt dieses Abschnitts ist die Anwendung der rekursiven Substrukturierung im Rahmen der Finite-Elemente-Methode hoher Ordnung. Ausgangspunkt ist das  $p$ -Version Finite-Elemente-Programm AdhoC<sup>4</sup> [36], welches am Lehrstuhl für Bauinformatik entwickelt wurde und in verschiedenen Forschungsarbeiten sehr erfolgreich eingesetzt wird. Auf einen wichtigen Unterschied im Vergleich zu klassischen Finite-Elemente-Ansätzen der  $h$ -Version soll schon hier hingewiesen werden. In der  $p$ -Version der FEM werden Freiheitsgrade nicht nur den Knoten zugeordnet, sondern können auch auf Kanten, Flächen und im Inneren eines Elementes existieren, was eine zusätzliche Verwaltung von Freiheitsgraden innerhalb der Hierarchie notwendig macht.

Eine wichtige Prämisse bei der Erweiterung hinsichtlich des hierarchischen Lösungsverfahrens bestand darin, vorhandene Programmstrukturen weitgehend beizubehalten und die Modifikationen möglichst einfach zu gestalten. Um dies zu erreichen, wurde eine strikte Trennung hinsichtlich der Schritte Zerlegung der Struktur, Finite-Elemente-Berechnung und Lösung des Gleichungssystems vorgenommen, die in jeweils separaten Modulen implementiert und ausgeführt werden. Die Verwaltung von Daten basiert dabei im Wesentlichen auf die in Abschnitt 3.2.3 beschriebene Indizierung von Bäumen mithilfe von Positionscodes. Innerhalb des Finite-Elemente-Programms können so bestehende Listen- und Datenstrukturen weitgehend beibehalten werden, alle hierarchischen Informationen der Elemente oder Freiheitsgrade werden mittels Positionscodes ausgedrückt. Zwischen den einzelnen Lösungsschritten ergibt sich dadurch eine klare und einfache Schnittstelle. Ein Wechsel eines Programmbausteins, um zum Beispiel ein anderes Prinzip der Zerlegung anzuwenden, ist damit leicht möglich. Insgesamt gliedert sich das Verfahren der rekursiven Substrukturierung in fünf Schritte:

1. Einlesen und Setup der Finite-Elemente-Geometrie (z.B. Herstellung der Konnektivität zwischen den Elementen).
2. Rekursive Dekomposition des FE-Modells. Die Zerlegung kann dabei schon - durch manuelle Einteilung der Bauwerksstruktur oder aufgrund einer anderweitig hierarchischen Sortierung (z.B. Oktalbaum) - gegeben sein, sie kann aber auch erst jetzt - z.B. durch rekursive Bisektion des Nachbarschaftsgraphen - festgelegt werden. Da dieser Schritt in

einem getrennten Programmmodul stattfindet, können hier unterschiedliche Verfahren leicht angewendet werden. Dem Finite-Elemente-Programm wird am Ende eine Liste mit (Elementnummer, Positionscode)-Paaren übergeben.

3. Setup der Finite-Elemente-Datenstruktur (Ermittlung einer konsistenten Polynomgradverteilung an den Elementgrenzen, Festlegung der Orientierungen und Vergabe der Freiheitsgrade), Einbringen von Randbedingungen und Berechnung aller Elementsteifigkeiten und Lastvektoren. Innerhalb des Finite-Elemente-Programms findet hier die Einsortierung aller Freiheitsgrade hinsichtlich ihrer Position in der Hierarchie statt.
4. Assemblierung und Lösung des Gesamtgleichungssystems. Dieser Schritt beinhaltet den Kern des Verfahrens der rekursiven Substrukturierung. Nach dem in Abschnitt 3.3.1 beschriebenen Vorgehen, gliedert sich dieses in einen bottom-up und einem top-down Durchlauf innerhalb der Hierarchie. Rekursiv werden alle Elementsteifigkeiten assembliert, das resultierende globale SCHUR-Komplement Problem an der Wurzel gelöst und schließlich die lokalen Unbekannten berechnet.
5. Postprocessing. Berechnung von Verzerrungen, Spannungen auf Elementebene und Ausgabe der Ergebnisse.

Schritt 2, 3 und 4 werden in den folgenden Abschnitten genauer erläutert.

### 3.4.1 Rekursive Zerlegung des Modells

Bevor das Verfahren der rekursiven Substrukturierung zur Lösung des linearen Gleichungssystems angewendet werden kann, sind die Elemente bzw. die Freiheitsgrade der Finite-Elemente-Diskretisierung hierarchisch zu organisieren. Grundsätzlich existieren unterschiedliche Prinzipien, nach denen diese Zerlegung erfolgen kann. Die Wahl des jeweiligen Prinzips wird dabei von den zu erwartenden Aufgaben und Anforderungen abhängen. Besonders bei der in dieser Arbeit betrachteten Aufgabenstellung, bei der Organisation und Berechnung großer Strukturmodelle, sind neben der reinen Gleichungslösung weitere Aspekte zu berücksichtigen, in denen hierarchische Datenstrukturen ein wichtiges Hilfsmittel sein können. Ein Ziel soll es daher sein, verschiedene hierarchische Organisationsprinzipien gerade im Hinblick auf diese Vielzahl von Aufgabengebieten zu untersuchen und zu vergleichen. Es soll also nicht allein die Anwendbarkeit im Rahmen der Gleichungslösung - und hier vor allem die schnelle Neuberechnung nach lokalen Modifikationen - im Mittelpunkt stehen. Aus diesem Grunde werden zunächst Oktalbäume betrachtet, die als raumpartitionierende Datenstrukturen besonders für geometrische Fragestellungen sehr wichtig sind. Um aber dennoch die volle Leistungsfähigkeit der hier untersuchten rekursiven Substrukturierung zur Lösung dünn besetzter Gleichungssysteme im Rahmen der  $p$ -Version der FEM zu demonstrieren, soll neben den geometrischen Organisationsprinzipien mit einer rekursiven Bisektion auch ein Verfahren der Graphenpartitionierung näher betrachtet werden.

#### 3.4.1.1 Oktalbäume

Die Generierung eines Oktalbaums zur hierarchischen Organisation der Elemente folgt dem in Abschnitt 3.2.2.3 beschriebenen Vorgehen: Jedes finite Element wird durch einen eindeutigen

Punkt im dreidimensionalen Raum repräsentiert. Dies kann der Elementmittelpunkt oder der Schwerpunkt sein. Es kann im Prinzip jeder beliebige Repräsentant gewählt werden, so lange eine Eindeutigkeit gewährleistet ist. Es ergibt sich eine Punktmenge  $P \subset \mathbb{R}^3$ , für die sich ein Oktalbaum generieren lässt. Dazu wird zunächst das Gebiet, welches die Punktmenge umschließt, mithilfe der Abbildung  $f : \mathbb{R}^3 \rightarrow [-1, 1]^3$  normiert. Dieser Einheitswürfel wird rekursiv immer weiter unterteilt, solange sich mehr als ein Punkt innerhalb eines Teilgebiets befindet. Damit erhält man einen Baum, in dem finite Elemente nur in den Blättern gespeichert sind, jedes Blatt repräsentiert maximal ein Element, Blätter können auch leer sein. Abbildung 3.10 zeigt die hierarchische Zerlegung eines ebenen Netzes mittels Oktalbaum.

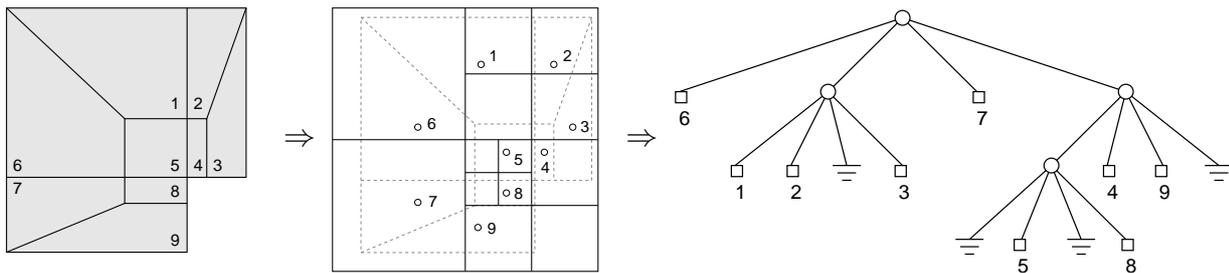


Abbildung 3.10: Hierarchische Organisation eines ebenen Finite-Elemente-Netzes. Die Elemente (links) werden an Hand ihres Schwerpunktes (mitte) in einen Quadtree (rechts) einsortiert.

Nachdem für jedes Element dessen Position innerhalb der Hierarchie feststeht, lassen sich für alle Positionscodes bestimmen (siehe Abschnitt 3.2.3) und als Liste dem Finite-Elemente-Programm übergeben.

### 3.4.1.2 Rekursive Bisektion

Das zweite betrachtete Organisationsschema basiert auf einer rekursiven Bisektion eines Graphen  $G$ , der in folgender Weise von dem Finite-Elemente-Netz abgeleitet ist: Jedes Element entspricht einem Knoten  $v$  in  $G$ . Eine Kante  $e(v_i, v_j)$  ergibt sich, wenn Element  $i$  und Element  $j$  eine gemeinsame Fläche haben.

Die rekursive Bisektion basiert auf dem klassischen Problem der  $k$ -Wege Partitionierung eines Graphen. Diese gehorcht folgendem Prinzip: Gegeben sei ein Graph  $G = (V, E)$  mit  $|V| = n$ . Die Knotenmenge  $V$  wird so in  $k$  Untermengen  $V_1, V_2, \dots, V_k$  zerlegt, dass  $V_i \cap V_j = \emptyset$  für  $i \neq j$ ,  $\bigcup_i V_i = V$ ,  $|V_i| \approx n/k$  und die Anzahl an Kanten in  $E$ , deren benachbarte Knoten in verschiedenen Untermengen liegen, minimiert wird. Die Anzahl der Kanten, die während der Partitionierung so quasi „durchschnitten“ werden, wird auch als *Edge-Cut* bezeichnet. Bei einer rekursiven Bisektion erfolgt diese Zerlegung in zwei Teile. Jede dieser Partitionen wird dann sukzessive weiter unterteilt, so lange bis keine Unterteilung mehr möglich ist. Es ergibt sich ein Binärbaum, in dem jedes Blatt genau ein finites Element repräsentiert. Abbildung 3.11 zeigt die prinzipielle Vorgehensweise für ein ebenes Netz (Eine Nachbarschaftsbeziehung ergibt sich im Falle eines ebenen Netzes bei einer gemeinsamen Kante zwischen den Elementen).

Die Anwendung der rekursiven Bisektion ist die klassische Vorgehensweise bei der Nested-Dissection, um den Fill-In während der Zerlegung einer Matrix und damit den Aufwand an Rechenoperationen zu reduzieren. Dies trifft natürlich in gleicher Weise auf die rekursive Substrukturierung zu. Eine Minimierung des Edge-Cuts auf jedem Level der Hierarchie hat einen

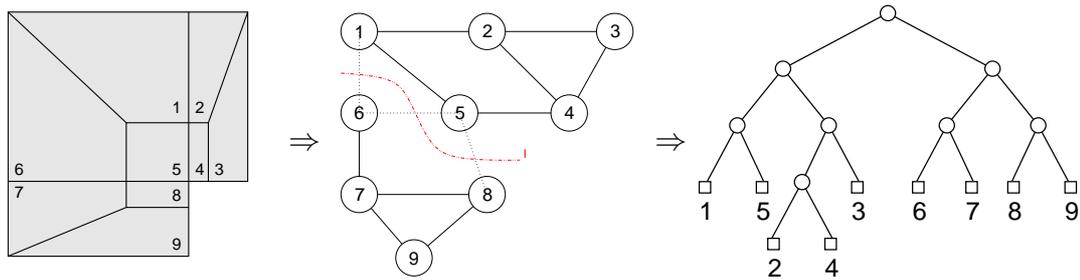


Abbildung 3.11: Zerlegung eines ebenen Finite-Elemente-Netzes durch rekursive Bisektion des Nachbarschaftsgraphen. Rechts: Binärbaum der Zerlegung

direkten Einfluss auf die Anzahl der externen Freiheitsgrade, die später auf dem entsprechenden Separator zu liegen kommen. Je weniger äußere Freiheitsgrade auf den Separatoren liegen, desto mehr innere Freiheitsgrade können schon in einer frühen Lösungsphase eliminiert werden, was sich schließlich positiv auf die Anzahl an insgesamt notwendigen Lösungsschritten auswirkt.

Die Bedingung einer ausgewogenen Verteilung der Knoten in den Teilgebieten ( $|V_i| \approx n/k$ ), kann jedoch für das Problem der Gleichungslösung Ergebnisse liefern, die nicht immer optimal sind. In Abbildung 3.11 ist die Zerlegung unter Berücksichtigung möglichst gleich großer Teilgebiete erfolgt. Dabei wird deutlich, dass bei anderen Schnittführungen unter Umständen bessere Unterteilungen möglich wären. Eine Zerlegung in die Partitionen  $V_1 = \{1, 5, 6, 7, 8, 9\}$  und  $V_2 = \{2, 3, 4\}$  hätte nur zwei Kanten durchschnitten, die Partitionen wären dann jedoch unterschiedlich groß. ASHCRAFT und LIU [10] weisen auf diese Problematik hin. Besonders bei Problemstellungen, bei denen das zugrunde liegende Berechnungsgebiet ein großes Seitenverhältnis aufweist, liefert eine klassische Nested-Dissection-Sortierung (Zerlegung mittels Bisektion) relativ schlechte Ergebnisse.

Um bessere Sortierungen zu finden, verwenden moderne Verfahren daher erweiterte Kriterien zur Optimierung der Schnittführung, die eine stärkere Minimierung des Edge-Cuts gegenüber der Ausgewogenheit der Partitionen fordern [9]. Aktuelle Löser wie SPOOLES<sup>6</sup> [7] oder WSMP<sup>7</sup> [55] verwenden meist eine Kombination unterschiedlicher Verfahren. Statt einer Zweiteilung wird der Graph mittels eines so genannten *Multisectors* in mehrere Partitionen zerlegt. In einer zweistufigen Strategie werden dann unabhängig voneinander jeweils im Inneren der Teilgebiete sowie für die Separatorvariablen auf dem Multisector unterschiedliche Sortierkriterien angewandt [10]. Empirische Untersuchungen in [10, 56] zeigen dabei, dass derartige Sortierkriterien anderen Verfahren wie beispielsweise Varianten der Minimum-Degree, insbesondere bei sehr unregelmäßigen Strukturen, überlegen sind.

Bei dem in dieser Arbeit betrachteten Problem der rekursiven Substrukturierung kommt einer Minimierung des Edge-Cuts besonders auf den obersten Ebenen der Hierarchie eine besondere Bedeutung zu. Von verschiedenen Autoren [62, 114] sowie in dieser Arbeit wurde beobachtet, dass die meisten Operationen (durchaus  $> 50\%$ ) in den obersten Ebenen der Hierarchie ausgeführt werden. Kann durch eine geeignete Schnittführung die Größe der Separatoren hier verringert werden, hat dies die größten Auswirkungen auf die Geschwindigkeit

<sup>6</sup>SPOOLES – SParse Object Oriented Linear Equations Solver: <http://www.netlib.org/linalg/spooles/>

<sup>7</sup>WSMP – Watson Sparse Matrix Package: <http://www-users.cs.umn.edu/~agupta/wsmp.html>

des Lösungsverfahrens. Diesem Umstand kommt eine noch größere Bedeutung zu, wenn man den Aufwand für eine partielle Neuberechnung nach einer lokalen Modifikation der Struktur betrachtet. Da sich die Lösung in diesem Falle nur auf einzelne Äste der Hierarchie auswirkt, findet nahezu der ganze Aufwand in den oberen Ebenen statt.

Die hierarchische Zerlegung des Graphen in den Beispielen dieser Arbeit erfolgt unter Anwendung des Softwarepakets CLUTO<sup>8</sup> ('CLUstering Toolkit') [64]. CLUTO ist eine Sammlung von Programmen zur Clusteranalyse von Datensätzen. Die Sammlung bietet unterschiedliche Algorithmen, die auf partitionierenden, agglomerativen oder graphentheoretischen Prinzipien beruhen. Einige Funktionen in dem Paket können auf gewichtete Graphen angewendet werden, in der Regel werden aber Vektoren im  $n$ -dimensionalen Raum als Eingangsdaten verwendet, die hinsichtlich verschiedener Kriterien klassifiziert oder gruppiert werden. Ein wichtiges Merkmal von CLUTO ist, dass in den meisten Algorithmen die Gruppierung der Daten als Optimierungsprozess betrachtet wird, in dem das Ergebnis hinsichtlich einer bestimmten Zielfunktion maximiert oder minimiert werden kann. Die Methoden zur Graphenpartitionierung in CLUTO basieren auf Multilevel Techniken und sind weitgehend von denen des Softwarepakets METIS [65] abgeleitet.

In den Beispielen dieser Arbeit erfolgt die Zerlegung des Nachbarschaftsgraphen unter Verwendung einer speziellen Optimierungsfunktion [29]. Die Funktion wird innerhalb von CLUTO angeboten. Sie zerlegt einen gewichteten Graph  $G = (V, E)$  so in  $k$  Partitionen, dass gilt:

$$\sum_{i=1}^k \frac{W(V_i, V - V_i)}{W(V_i, V_i)} \Rightarrow MIN. \quad (3.12)$$

Die Größe  $W(V_1, V_2) = \sum w_e(v_i, v_j), v_i \in V_1, v_j \in V_2$  bezeichnet die Summe der Gewichte<sup>9</sup> aller Kanten zwischen zwei Knotenmengen  $V_1$  und  $V_2$ , was dem Edge-Cut zwischen den Partitionen entspricht. Da in Gleichung (3.12) die alleinige Minimierung der externen Kantengewichte  $W(V_i, V - V_i)$  auf triviale Lösungen führen würde, wird diese Größe zusätzlich mit den Gewichten aller inneren Kanten jeder Partition skaliert. Die Hinzunahme dieser Optimierung während der Zerlegung des Graphen lieferte besonders bei den komplexen und großen Beispielen in dieser Arbeit die besten Ergebnisse.

### 3.4.2 Hierarchische Sortierung aller Freiheitsgrade

In der Finite-Elemente-Methode wird das Berechnungsgebiet in Elemente eingeteilt und auf den Elementen lokale Formfunktionen formuliert. Das globale Verschiebungsfeld ergibt sich aus einer Linearkombination dieser abschnittsweise definierten Funktionen. Da in der Regel ein stetiger Übergang zwischen den Elementen gefordert wird, teilen sich Formfunktionen benachbarter Elemente gemeinsame Freiheitsgrade. In der  $p$ -Version der FEM werden die lokalen Ansatzfunktionen hinsichtlich Knoten-, Kanten-, Flächen- und innere Moden unterschieden. Gemeinsam verwendete Freiheitsgrade können daher neben den Knoten auch auf Kanten und Flächen existieren.

<sup>8</sup>CLUTO – Family of Data Clustering Software Tools: <http://glaros.dtc.umn.edu/gkhome/views/cluto>

<sup>9</sup>In den hier betrachteten Beispielen wurden die Gewichte aller Kanten einheitlich auf 1 gesetzt. Es ist aber auch möglich, abhängig vom Polynomgrad eine Gewichtung der Nachbarschaftsbeziehungen vorzunehmen. Dies wurde vom Autor dieser Arbeit auch untersucht. Die Anpassung der Gewichte hatte jedoch zu keinen wesentlichen Unterschieden im Ergebnis geführt.

Beim Verfahren der rekursiven Substrukturierung werden zur Lösung des Gesamtgleichungssystems Elemente zu Teilgebieten und diese zu immer größeren Substrukturen zusammengefasst. Die dazugehörigen Steifigkeitsmatrizen werden assembliert und danach die jeweils inneren Freiheitsgrade eliminiert. Ein Freiheitsgrad kann immer dann eliminiert werden, wenn dieser im Inneren eines Teilgebiets liegt, wenn also alle finiten Elemente, die einen Einfluss darauf besitzen, komplett dieser Substruktur angehören. Dieses Kriterium kann auch geometrisch gedeutet werden, da jeder Freiheitsgrad einem Knoten, Kante, Fläche oder dem Inneren eines Elementes zugeordnet ist. Eine Elimination kann stattfinden, wenn das Objekt (Knoten, Kante, Fläche) auf dem ein Freiheitsgrad formuliert ist, zu keinem anderen (finiten) Element ausserhalb einer Substruktur mehr benachbart ist. Für das hier betrachtete Lösungsverfahren bedeutet dies, dass die Eliminationsreihenfolge schon innerhalb des Finite-Elemente-Netzes vor dem Setup aller Ansatzfunktionen festgelegt werden kann. Sie ist damit nur von den topologischen Eigenschaften eines Netzes abhängig, bei einer Änderung des Polynomgrades muss keine neue Hierarchisierung mehr vorgenommen werden.

Die Einordnung aller Knoten, Kanten und Flächen innerhalb der Hierarchie zur Festlegung der Eliminationsreihenfolge folgt nun dem Prinzip des *kleinsten gemeinsamen Vaterknotens*. Ein Objekt wird genau demjenigen Baumknoten zugeordnet, von dem bei einem Abstieg aus alle benachbarten finiten Elemente gerade noch erreicht werden können. Je tiefer das Objekt innerhalb des Baumes plaziert werden kann, desto früher können die darauf liegenden Freiheitsgrade eliminiert werden, was sich letztendlich auf die Anzahl an Operationen im Lösungsverfahren auswirkt. Abbildung 3.12 zeigt die hierarchische Sortierung von Knoten und Kanten nach dem Prinzip des kleinsten gemeinsamen Vaterknotens am Beispiel eines ebenen Netzes mit vier Elementen.

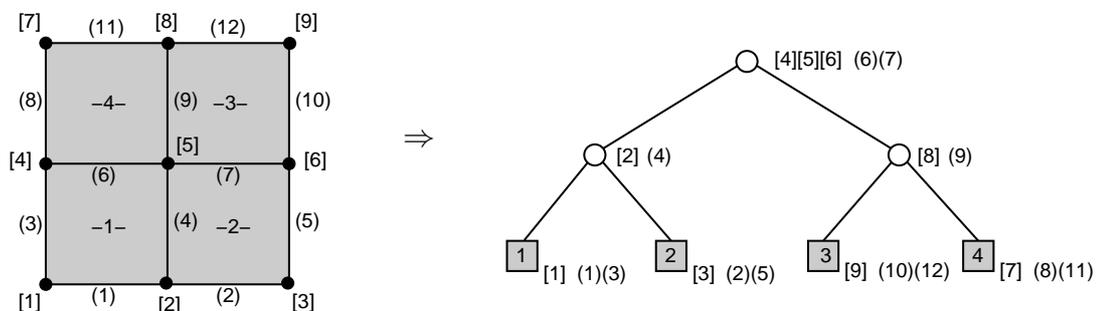


Abbildung 3.12: Hierarchische Sortierung von Knoten [\*] und Kanten (\*) und damit deren Freiheitsgrade in einem ebenen Finite-Elemente-Netz.

Zur Bestimmung des gemeinsamen Vaterknotens von  $n$  Elementen lässt sich nun die in Abschnitt 3.2.3 vorgestellte Möglichkeit zur Indizierung von Baumdatenstrukturen mithilfe von Positionscodes nutzen. Der Positionscod  $p$  wird als Zeichenkette  $a_1, \dots, a_H$  mit Zeichen aus der Menge  $a_i \in \{0, 1, \dots, k\}$  dargestellt, wobei die Anzahl  $k$  an Ziffern der maximalen Ordnung des Baumes entspricht. Für einen bestimmten Knoten im Baum ergibt sich der jeweilige Positionscod durch Konkatenation dieser Ziffern, die bei einem Baumabstieg von der Wurzel bis zu dem indizierten Knoten entsprechend den besuchten Ästen vergeben werden. Zur Bestimmung eines gemeinsamen Vaterknotens von  $n$  Elementen wird eine neue Zeichenkette  $r$  angelegt und dann die Positionscodes  $p_i$  aller  $n$  beteiligten Elemente zeichenweise von links

beginnend verglichen. So lange ein Zeichen an der Position  $m$  bei allen  $p_i$  übereinstimmt, wird dieses dem Ausdruck  $r$  hinzugefügt. Ist mindestens ein  $p_i$  unterschiedlich, ist der gemeinsame Wurzelknoten bestimmt. Die Länge der Zeichenkette  $r$  entspricht dann der Tiefe des gemeinsamen Vaterknotens. Sie kann auch leer sein. Dann handelt es sich um die Wurzel des Baumes. Abbildung 3.13 veranschaulicht die Positionierung der Kante (9) aus Bild 3.12. Diese ist zu den Elementen 3 und 4 mit Positionscodes 010 bzw. 011 benachbart und es ergibt sich ein Positionscod von 01 für den gemeinsamen Vaterknoten. Für Bäume niedriger Ordnung lässt

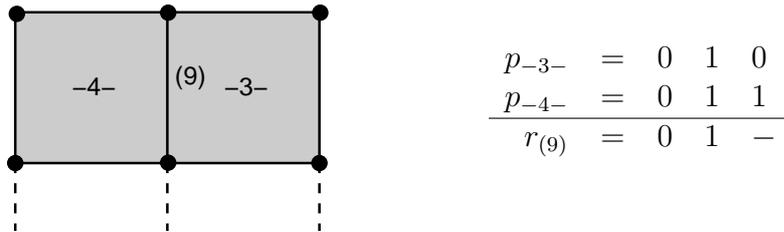


Abbildung 3.13: Bestimmung der Position von Kante (9) durch Bestimmung der kleinsten gemeinsamen Wurzel von Element 3 und Element 4.

sich der Positionscod auch innerhalb einer Zahl zur Basis  $k$  (z.B. Oktalbäume zur Basis 8) kodieren. Die entsprechenden Vergleiche lassen sich dann sehr effizient mittels Bitoperationen durchführen.

Nach diesem Prozeß sind neben den Elementen auch alle weiteren geometrischen Objekte des Finite-Elemente-Netzes wie Knoten, Kanten und Flächen innerhalb der Hierarchie positioniert. Für die diesen Objekten zugeordneten Freiheitsgrade ist so die Position innerhalb der Substrukturhierarchie und damit der Zeitpunkt der Elimination festgelegt.

### 3.4.3 Assemblierung und Lösung des Gleichungssystems

Die Berechnung des Finite-Elemente-Gleichungssystems erfolgt in der rekursiven Substrukturierung nun gleichzeitig während der Assemblierung der lokalen Steifigkeitsmatrizen. In einer Postorder-Traversierung (Bottom-Up) werden rekursiv an jedem Knoten des Baumes Steifigkeiten der Sohnknoten zusammengefasst und die jeweils inneren Freiheitsgrade eliminiert.

Um  $C^0$ -Stetigkeit an den Elementrändern zu gewährleisten, muss die Assemblierung lokaler Steifigkeitsmatrizen nach einem bestimmten Schema erfolgen. Analog der Zuordnung der lokalen Formfunktionen  $N_i^e$  zu den globalen Freiheitsgraden  $u_j$  werden die Einträge einer lokalen Steifigkeit additiv der globalen Gesamtsteifigkeitsmatrix an der entsprechenden Stelle hinzugefügt. Dieser Prozess der Assemblierung wird in der Regel über ein Indexfeld organisiert

$$i \rightarrow j = c^e(i), \quad (3.13)$$

welches beim Setup der Elementansatzfunktionen innerhalb des Finite-Elemente-Programms für jedes finite Element  $e$  bestimmt wird. Genauere Informationen über den Aufbau der Ansatzfunktionen sowie die Assemblierung der Steifigkeiten im Rahmen der  $p$ -Version der FEM finden sich in [44]. Das Indexfeld in (3.13) wird auch als Konnektivität der lokalen Ansatzfunktionen bezeichnet [26].

Ausgangspunkt der hierarchischen Assemblierung im Rahmen der rekursiven Substrukturierung sind die lokalen Elementsteifigkeiten  $K^e$ , die zusammen mit dem Lastvektor und dem Indexfeld  $c^e$  an den Blättern des Baumes gespeichert sind. Neben den Krastrandbedingungen im Lastvektor müssen auch die Verschiebungsrandbedingungen schon auf lokaler Ebene der Elemente eingebracht sein. Dies kann durch Entfernen der entsprechenden Zeilen und Spalten aus dem lokalen Gleichungssystem geschehen, genauso aber auch mithilfe des Penalty-Verfahrens [133]. Die Assemblierung der Steifigkeiten erfolgt nun bottom-up beispielsweise durch eine Postorder-Traversierung über den Baum. Für jeden Knoten werden die folgenden Schritte durchgeführt (an Blattknoten nur Punkt 2 und 4):

1. Bestimme ein neues Konnektivitätsfeld aus der Vereinigung aller äußeren Freiheitsgrade der Söhne  $s$

$$c = \bigcup_{s=1}^{n_s} c_a^s. \quad (3.14)$$

2. Sortiere die Freiheitsgradindices in  $c$  nach Innen und Außen

$$c = [c_i | c_a], \quad (3.15)$$

wobei die zu eliminierenden, inneren Freiheitsgrade  $c_i$  nach dem Verfahren des kleinsten gemeinsamen Vaterknotens in Abschnitt 3.4.2 bestimmt worden sind.

3. Assembliere die SCHUR-Komplemente aller Söhne in einer neuen Steifigkeitsmatrix  $K = \mathbb{R}^{n \times n}$  mit  $n = \dim(c)$ :

$$K = \mathbf{A} \begin{matrix} n_s \\ s=1 \end{matrix} \tilde{K}_a^s. \quad (3.16)$$

Dabei müssen die Einträge aus den Matrizen der Söhne unter Berücksichtigung beider Konnektivitätsfelder  $c$  und  $c^s$  an den entsprechenden Stellen in  $K$  eingetragen werden.

4. Eliminiere die inneren Freiheitsgrade in  $K$  entsprechend Abschnitt 3.3.1.1.

Nach Traversierung des gesamten Baumes und Zerlegung des resultierenden Gleichungssystems im Wurzelknoten können schließlich top-down, etwa in Form einer Preorder-Traversierung, alle lokalen Unbekannten nach Gleichung (3.9) bestimmt werden.

### 3.4.4 Berechnungsbeispiele

Um die Tragfähigkeit des hier vorgestellten Ansatzes zu demonstrieren, sollen im Folgenden Beispiele unterschiedlicher Komplexität untersucht werden. Die Beispiele reichen von einfachen geometrischen Körpern bis hin zu realistischen Tragwerksstrukturen von Bauwerken. Dazu wurden jeweils die Laufzeiten sowie die Anzahl an notwendigen Gleitkommaoperationen zur Assemblierung und Lösung der Elementgleichungssysteme gemessen. Besonders die Messung der Gleitkommaoperationen bietet die Möglichkeit, ein Verfahren unabhängig vom verwendeten Rechnersystem und der spezifischen Implementierung vergleichen zu können.

Es soll in den Beispielen gezeigt werden, dass der hier gewählte Ansatz ein sehr effizientes Verfahren zur Lösung dünn besetzter Gleichungssysteme, die aus einer Finite-Elemente-Diskretisierung resultieren, darstellt. Neben Zeitmessungen und Untersuchungen von Leistungswerten sollen die Ergebnisse zudem auch mit anderen direkten und iterativen Lösungsverfahren verglichen werden. Bei der Interpretation der Ergebnisse soll aber immer berücksichtigt werden, dass in dieser Arbeit nicht primär ein schnelles Lösungsverfahren entwickelt werden soll, sondern hierarchische Datenstrukturen zur effektiven Organisation im Hinblick auf die Berechnung von Gesamtmodellen im konstruktiven Ingenieurbau verwendet werden sollen.

Sämtliche Berechnungszeiten wurden auf einem AMD Opteron 250 mit 2,4 GHz Taktfrequenz und 1024 kByte Level-2-Cache erhoben. Als Betriebssystem kam SuSE Linux mit einem gcc-Compiler Version 3.3.3 und den Optionen `-O3 -mcpu=k8` zum Einsatz. Die wesentlichen Berechnungsschritte bei der Kondensation der inneren Freiheitsgrade wurden gemäß Abschnitt 3.3.1.1 auf LAPACK- und BLAS-Routinen ausgelagert. Hierzu kam die selbstoptimierende ATLAS-Bibliothek<sup>10</sup> zum Einsatz, welche auf dem Rechner die besten Ergebnisse liefert.

#### 3.4.4.1 Testbeispiele unterschiedlicher Struktur

Die geometrische und topologische Struktur<sup>11</sup> eines Finite-Elemente-Problems hat einen wesentlichen Einfluss auf die Struktur des resultierenden Gleichungssystems, welches sich dann wiederum auf das Lösungsverhalten eines bestimmten Verfahrens auswirkt. Um diesen Einfluss zu berücksichtigen sollen im Folgenden vier Beispiele untersucht werden, die sich hinsichtlich ihrer Seitenverhältnisse und Struktur unterscheiden. Dabei handelt es sich um einen würfelförmigen Block mit gleicher Kantenlänge, einer Platte, einem Balken sowie einer fachwerkartigen Struktur. Alle Beispiele wurden strukturiert vernetzt, an zwei Seiten eingespannt und unter Berücksichtigung einer Volumenlast berechnet. Abbildung 3.14 zeigt die vier Beispiele. Für die Untersuchungen wurden die Finite-Elemente-Netze nach zwei verschiedenen

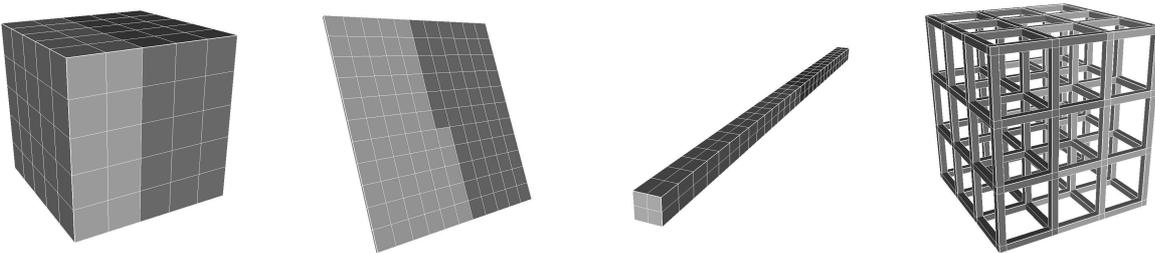


Abbildung 3.14: Berechnungsgebiete mit unterschiedlichen Seitenverhältnissen. Von links nach rechts: Block (125 Elemente), Platte (121), Balken (120) und Fachwerk (208). Es sind jeweils die ersten beiden Partitionen farblich unterschieden.

Prinzipien zerlegt: Zunächst mittels rekursiver Bisektion, wobei für die oberen drei Ebenen der Hierarchie die Zerlegung unter Berücksichtigung der Optimierungsfunktion in (3.12) erfolgte,

<sup>10</sup>ATLAS - Automatically Tuned Linear Algebra Software: <http://www.netlib.org/atlas/>

<sup>11</sup>Je nach Art der Zerlegung: Ein Oktalbaum zerlegt ein Modell nur nach geometrischen Kriterien, während eine rekursive Bisektion rein von den topologischen Nachbarschaftsbeziehungen ausgeht.

sowie mittels einer Oktaalbaumzerlegung. Ergebnisse sind in den Tabellen 3.1 und 3.2 dargestellt. Diese listet für alle Beispiele die Gesamtlaufzeiten (Assemblierung + Lösung) sowie die Anzahl an Gleitkommaoperationen für zunehmende Polynomgrade auf. Die Abbildungen 3.15 und 3.16 zeigen jeweils für beide Zerlegungsarten zudem die ermittelten Laufzeiten aufgetragen über die Anzahl an Freiheitsgraden des Gesamtgleichungssystems in einer grafischen Darstellung.

Man erkennt, dass sich die Kompaktheit der Struktur auf den erforderlichen Lösungsaufwand auswirkt. Je kompakter die Struktur, desto größer werden die Separatoren und damit die lokalen Matrizen in den oberen Bereichen des Baumes. Bei einer Zerlegung mittels rekursiver Bisektion lassen sich besonders stabförmige Strukturen, die sich schon in den ersten Hierarchiestufen mit kleinen Separatoren zerlegen lassen, mit einem sehr geringen Aufwand lösen. Weiterhin erkennt man, dass die Ordnung des Verfahren bei allen betrachteten Beispielen annähernd konstant ist. Dies liegt daran, dass der Polynomgrad immer hinsichtlich aller drei Raumrichtungen erhöht wurde und damit in allen Fällen prinzipiell ein Volumenproblem vorliegt.

$p$	block		platte		balken		fachwerk	
	$t$ [s]	$n_{op}$ [ $10^9$ ]	$t$ [s]	$n_{op}$ [ $10^9$ ]	$t$ [s]	$n_{op}$ [ $10^9$ ]	$t$ [s]	$n_{op}$ [ $10^9$ ]
1	0,041	0,006	0,040	0,005	0,008	0,003	0,041	0,033
2	0,141	0,129	0,145	0,079	0,061	0,045	0,249	0,289
3	0,440	0,555	0,320	0,317	0,172	0,178	0,677	1,005
4	1,255	2,139	0,729	1,085	0,477	0,617	1,686	2,998
5	3,320	6,875	1,784	3,136	1,189	1,836	3,867	7,791
6	7,997	18,912	3,922	7,942	2,676	4,793	8,176	18,156
7	17,623	45,809	7,985	18,105	5,517	11,260	15,846	38,768
8	35,844	100,246	15,559	37,952	10,842	24,298	30,195	77,160
9	70,987	202,133	28,211	74,348	20,152	48,934	53,549	145,018
10	129,380	381,254	50,453	137,798	35,914	93,112	94,792	259,961

Tabelle 3.1: Gesamtlaufzeiten  $t$  [s] und Anzahl an Gleitkommaoperationen  $n_{op}$  [ $10^9$ ] für steigende Polynomgrade. Zerlegung mittels rekursiver Bisektion.

$p$	block		platte		balken		fachwerk	
	$t$ [s]	$n_{op}$ [ $10^9$ ]	$t$ [s]	$n_{op}$ [ $10^9$ ]	$t$ [s]	$n_{op}$ [ $10^9$ ]	$t$ [s]	$n_{op}$ [ $10^9$ ]
1	0,089	0,011	0,009	0,005	0,027	0,038	0.104	0.067
2	0,141	0,255	0,075	0,091	0,332	0,809	0.305	0.626
3	0,614	1,150	0,241	0,385	1,229	3,547	0.890	2.217
4	1,699	4,589	0,699	1,339	4,226	13,470	2.441	6.661
5	5,012	15,183	1,759	3,903	12,428	42,684	5.853	17.350
6	13,637	42,606	4,067	9,917	32,311	115,639	12.990	40.305
7	30,704	104,609	8,570	22,615	78,204	276,059	26.130	85.420
8	66,011	230,922	17,008	47,315	160,761	595,574	50.324	168.094
9	132,892	467,991	31,074	92,358	324,109	-	91.820	311.326
10	255,771	884,693	55,056	170,346	602,160	-	-	-

Tabelle 3.2: Gesamtlaufzeiten  $t$  [s] und Anzahl an Gleitkommaoperationen  $n_{op}$  [ $10^9$ ] für steigende Polynomgrade. Zerlegung mittels Oktalbaum.

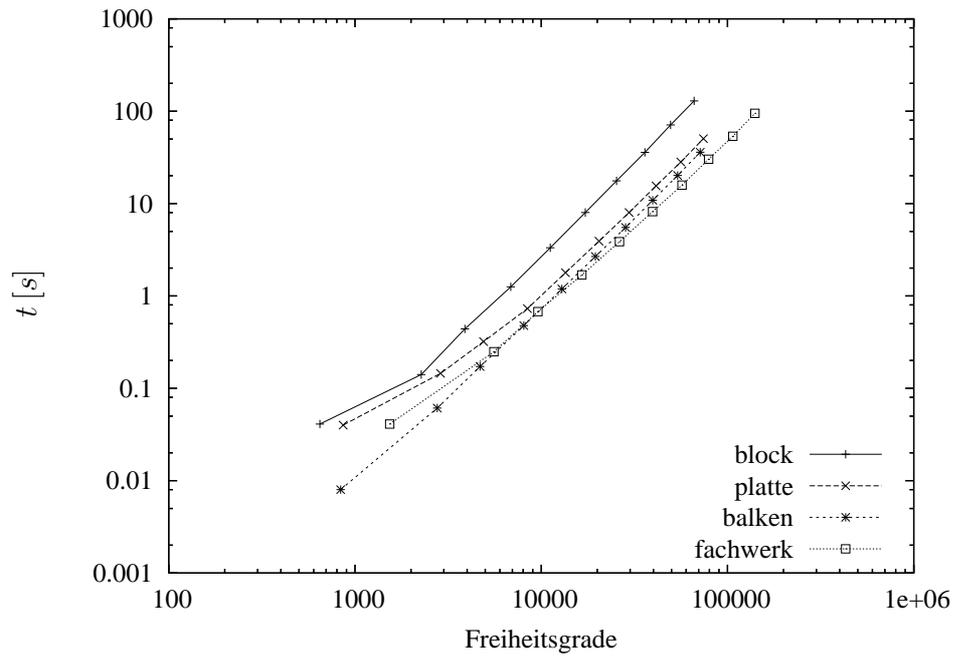


Abbildung 3.15: Berechnungszeiten  $t$  gegenüber Anzahl an Freiheitsgraden bei Zerlegung mittels rekursiver Bisektion für die Beispiele aus Abbildung 3.14.

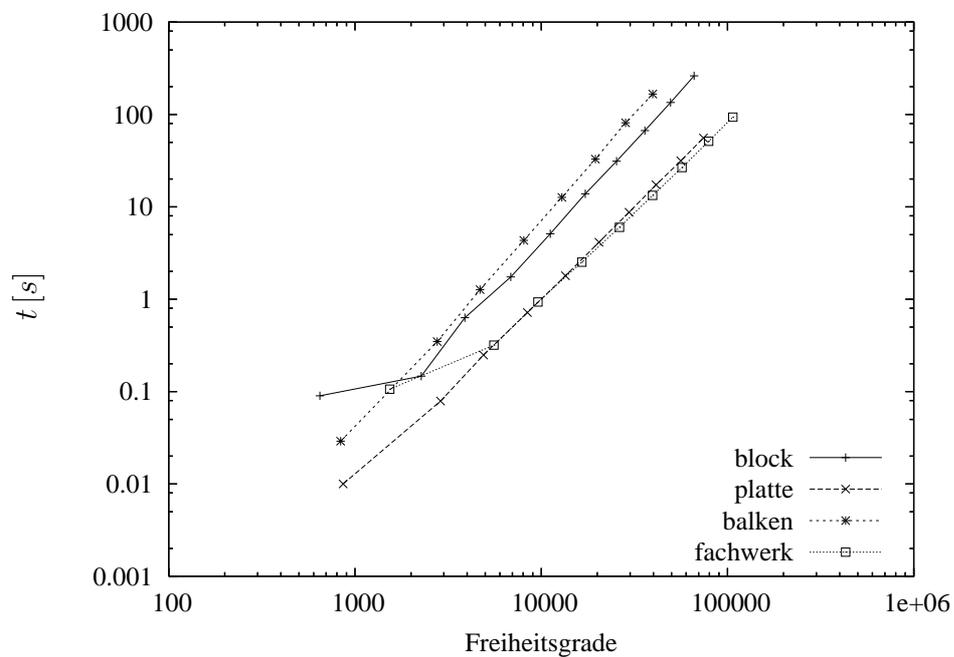


Abbildung 3.16: Berechnungszeiten  $t$  gegenüber Anzahl an Freiheitsgraden bei Zerlegung mittels Oktalbaum für die Beispiele aus Abbildung 3.14.

Teilt man die Anzahl an notwendigen Gleitkommaoperationen durch die Zeitdauer erhält man die Leistung in GFlop/s die für die jeweilige Implementierung auf dem Rechnersystem erreicht wurden. Mit Anwendung der rekursiven Substrukturierung in der  $p$ -Version der FEM lassen sich dabei sehr hohe Leistungswerte erzielen. Aufgrund der Tatsache, dass während der Block-Gauss Zerlegung gemäß Abschnitt 3.3.1.1 alle Operationen als Matrix-Matrix Operationen ausgeführt werden können, lassen sich hier die entsprechenden Level 3 Routinen der BLAS und LAPACK Bibliotheken in optimaler Weise ausnutzen. Wie in den Abbildungen 3.17 und 3.18 zu sehen ist, konvergieren in den obigen Beispielen die Leistungswerte für die Assemblierungsphase, welche die Gesamtlösung bei weitem dominiert, gegen einen Grenzwert von über 3,5 GFlop/s. Für den hier verwendeten Prozessor entspricht dies nahezu 50 Prozent der theoretisch möglichen Leistung<sup>12</sup>.

Die Berücksichtigung einer zugrunde liegenden Rechnerarchitektur spielt bei der Implementierung von Lösungsverfahren eine wichtige Rolle [32]. Neben einer Minimierung der notwendigen Gleitkommaoperationen ist immer auch zu beachten, dass sich auf modernen Rechnersystemen mit einer hierarchischen Speicherstruktur Matrix-Matrix-Operationen wesentlich effizienter ausführen lassen, als etwa Matrix-Vektor oder Vektor-Vektor-Operationen [5]. In [32] wird darauf hingewiesen, dass klassische Nested-Dissection-Verfahren gerade in diesem Zusammenhang nicht immer optimal sind. Besser eignen sich hier Frontal- bzw. Multifrontallöser [34, 33], die weitgehend auf dichtbesetzten Teilmatrizen operieren können.

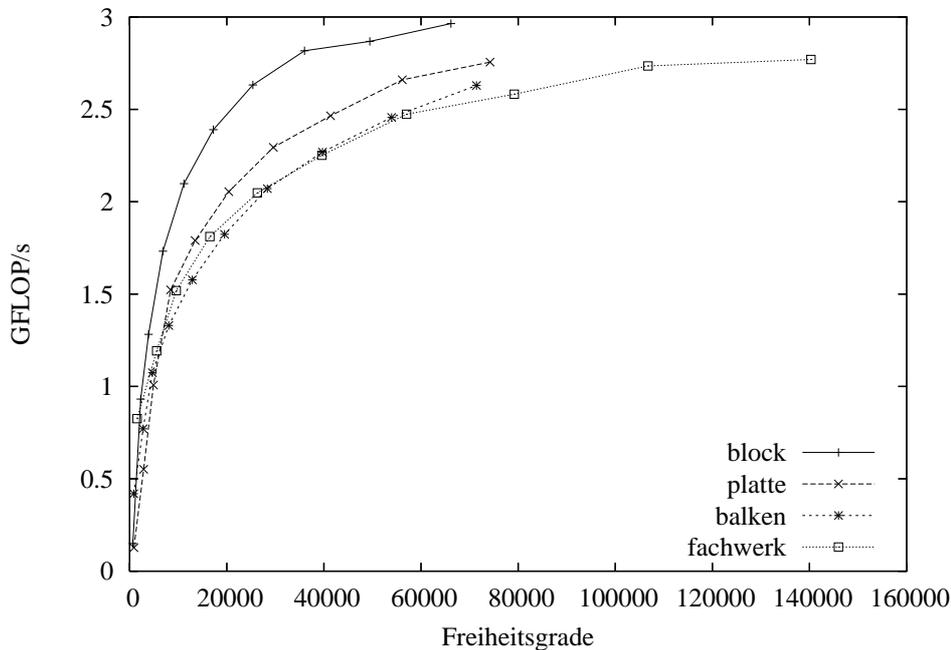


Abbildung 3.17: Leistungswerte in GFLOP/s für die Assemblierungsphase bei rekursiver Bisektion.

<sup>12</sup>Laut AMD wird für ein Einprozessorsystem eines Opteron mit 2.4 GHz ein Wert von 7400 Millionen theoretischen Operationen pro Sekunden (MTOPS) angegeben (Composite Theoretical Performance). Es ist noch zu bemerken, dass hier sowie in den Beispielen immer 64 bit Gleitkommaoperationen betrachtet werden.

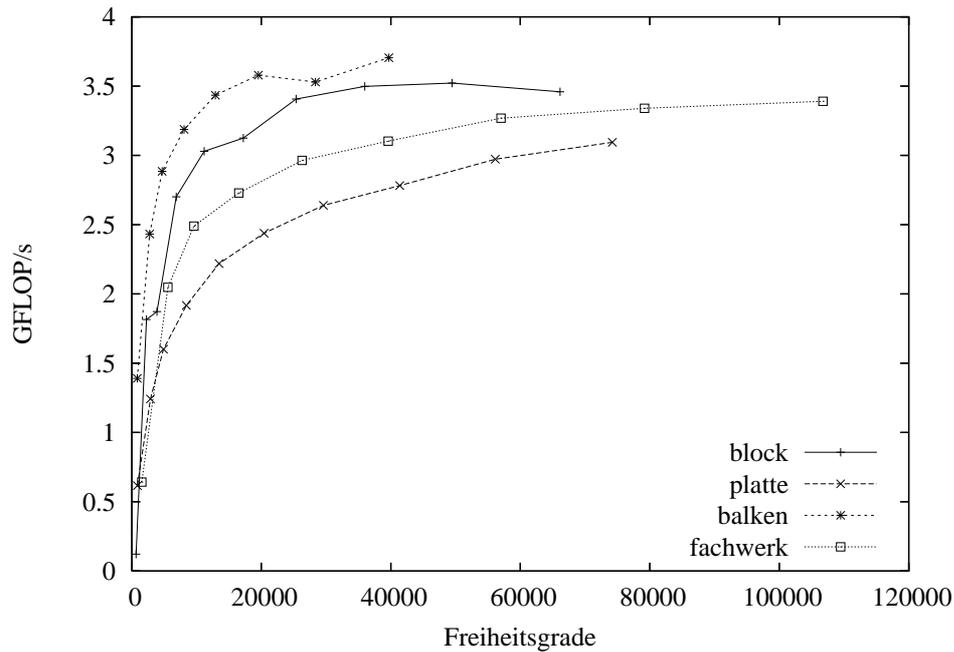


Abbildung 3.18: Leistungswerte in GFLOP/s für die Assemblierungsphase bei Zerlegung mittels Oktalbaum.

#### 3.4.4.2 Vergleich mit anderen Lösungsverfahren

Um das hier vorgestellte Konzept besser einordnen zu können, soll dieses mit anderen Methoden zur Lösung dünn besetzter Gleichungssysteme verglichen werden. Als direktes Verfahren kommt hierzu die Software-Bibliothek SPOOLES [8] zur Anwendung. Diese bietet Lösungsverfahren für schwach besetzte Gleichungssysteme, in denen mit Nested-Dissection, Minimum-Degree und Multisection auch unterschiedliche Methoden zur Bestimmung der Eliminationsreihenfolge gewählt werden können. In den hier betrachteten Beispielen wurde hierzu eine Multisection-basierte Sortierung gewählt [10]. Daneben soll mit einem CG-Verfahren mit SSOR-Vorkonditionierung auch ein iterativer Löser zum Vergleich herangezogen werden. Der betrachtete CG-Löser stellt bislang die Standardimplementierung im FE-Code AdhoC [36] am Lehrstuhl für Bauinformatik der TU München dar. Der Löser nutzt dabei auch Techniken zur Multiplikation dünn besetzter Matrizen aus. Weitere Details findet der Leser in [105].

Abbildungen 3.19 bis 3.22 zeigen einen Vergleich der Gleitkommaoperationen für die betrachteten Testbeispiele aus Abbildung 3.14. Zur Bestimmung des Aufwands wurden in SPOOLES nur Gleitkommaoperationen während der numerischen Lösung gezählt, die Anzahl an Tauschoperationen während der Sortierung der Einträge ist in der Regel geringer und kann für die hier betrachteten Beispiele vernachlässigt werden. Die Abbildungen zeigen, dass die Anzahl an Gleitkommaoperationen zur Lösung eines bestimmten Problems bei der rekursiven Substrukturierung im Vergleich zu SPOOLES ähnlich ist. Weiterhin fällt auf, dass sich das iterative gegenüber den beiden direkten Verfahren hinsichtlich des Typs der Struktur genau entgegengesetzt verhält. Während die direkten Lösungsverfahren bei weniger kompakten Strukturen bessere Ergebnisse liefern, benötigt das CG-Verfahren für das kompakte würfelförmige Beispiel weniger Operationen bei gleicher Anzahl an Freiheitsgraden. Dies liegt sicherlich an der

besseren Kondition des Problems. Zudem ist in den Abbildungen die niedrigere Ordnung des iterativen Verfahrens zu erkennen, welches bei zunehmender Problemgröße günstigere Lösungszeiten erlaubt.

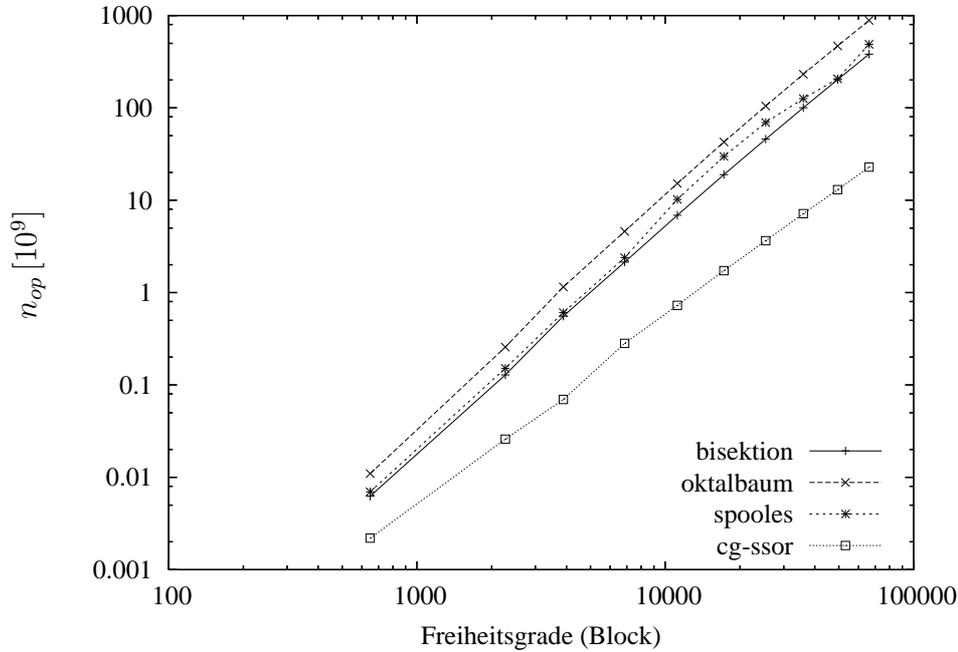


Abbildung 3.19: Vergleich unterschiedlicher Lösungsverfahren nach Anzahl an Gleitkommaoperationen für das Beispiel des Blockes aus Abbildung 3.14.

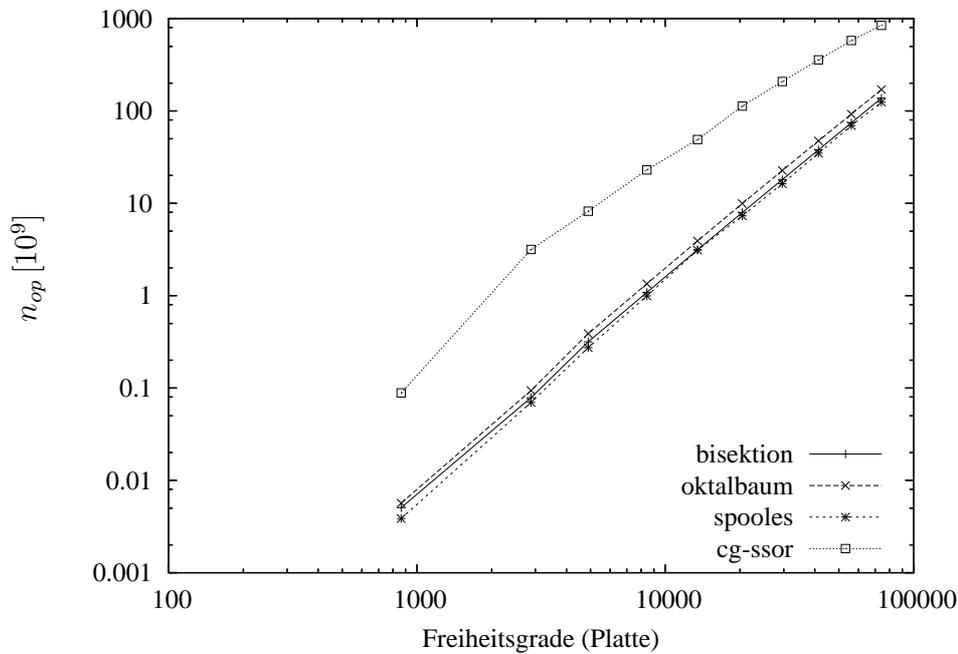


Abbildung 3.20: Vergleich unterschiedlicher Lösungsverfahren nach Anzahl an Gleitkommaoperationen für das Beispiel der Platte aus Abbildung 3.14.

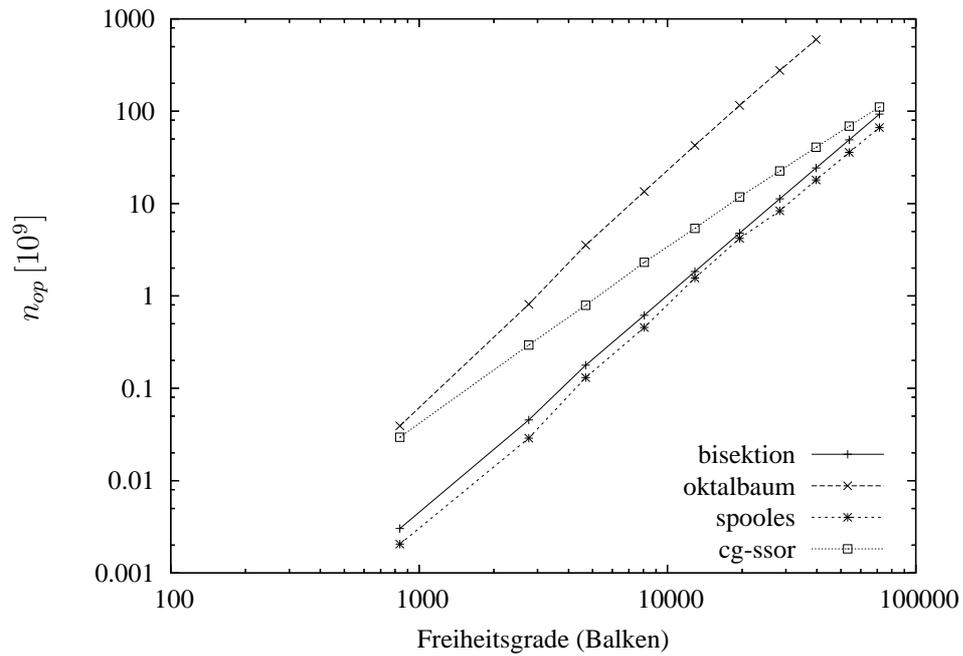


Abbildung 3.21: Vergleich unterschiedlicher Lösungsverfahren nach Anzahl an Gleitkommaoperationen für das Beispiel des Balkens aus Abbildung 3.14.

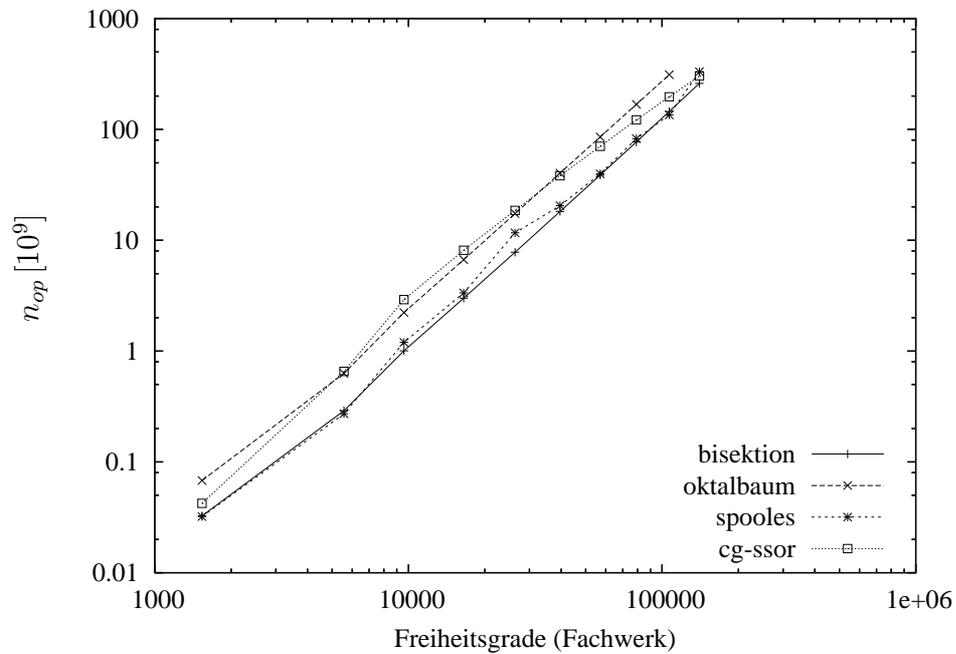


Abbildung 3.22: Vergleich unterschiedlicher Lösungsverfahren nach Anzahl an Gleitkommaoperationen für das Beispiel des Raumfachwerks aus Abbildung 3.14.

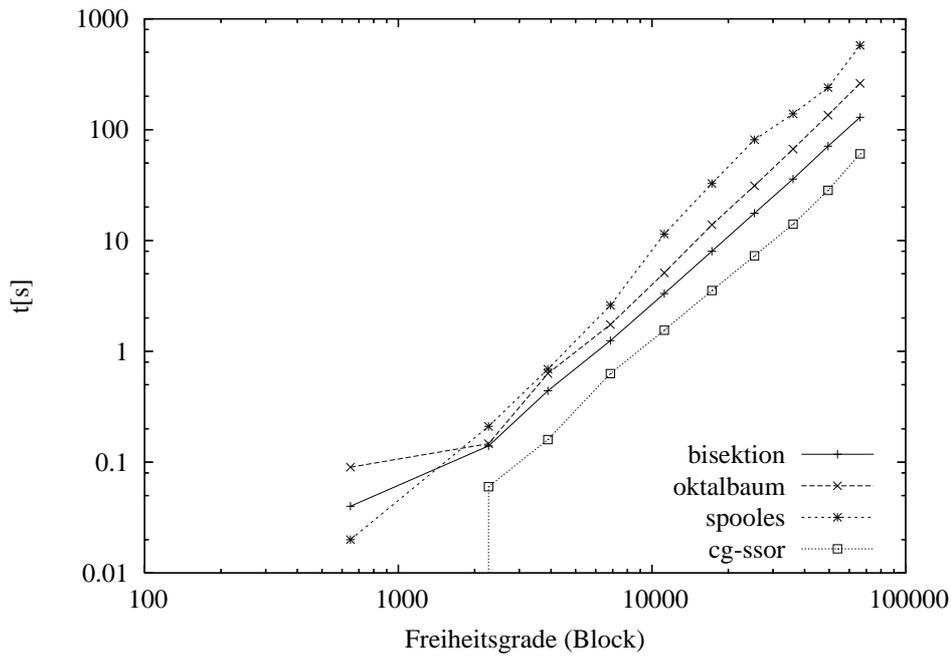


Abbildung 3.23: Vergleich der Laufzeiten unterschiedlicher Lösungsverfahren für das Beispiel des Blockes aus Abbildung 3.14.

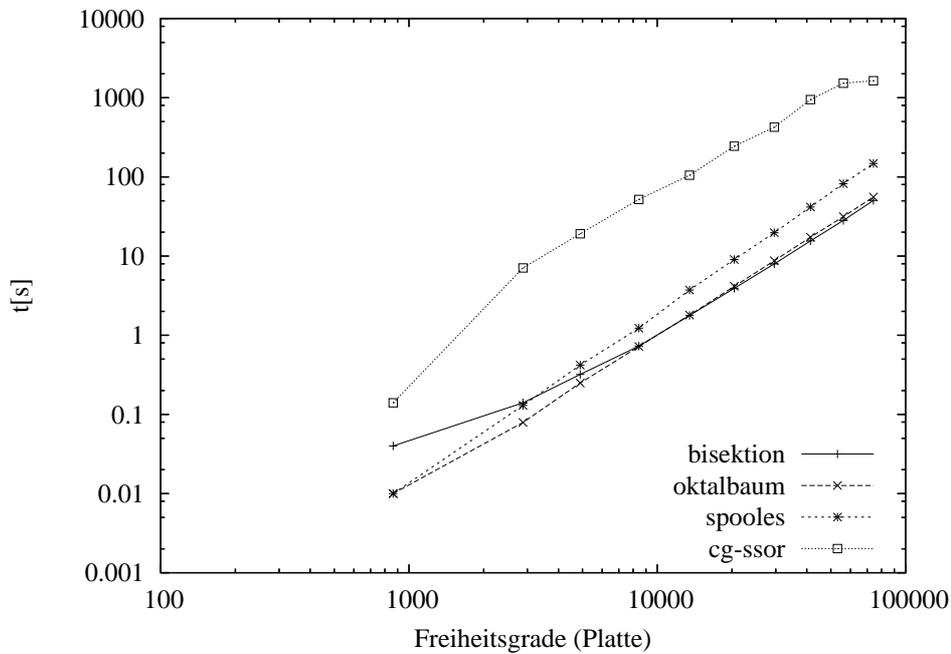


Abbildung 3.24: Vergleich der Laufzeiten unterschiedlicher Lösungsverfahren für das Beispiel der Platte aus Abbildung 3.14.

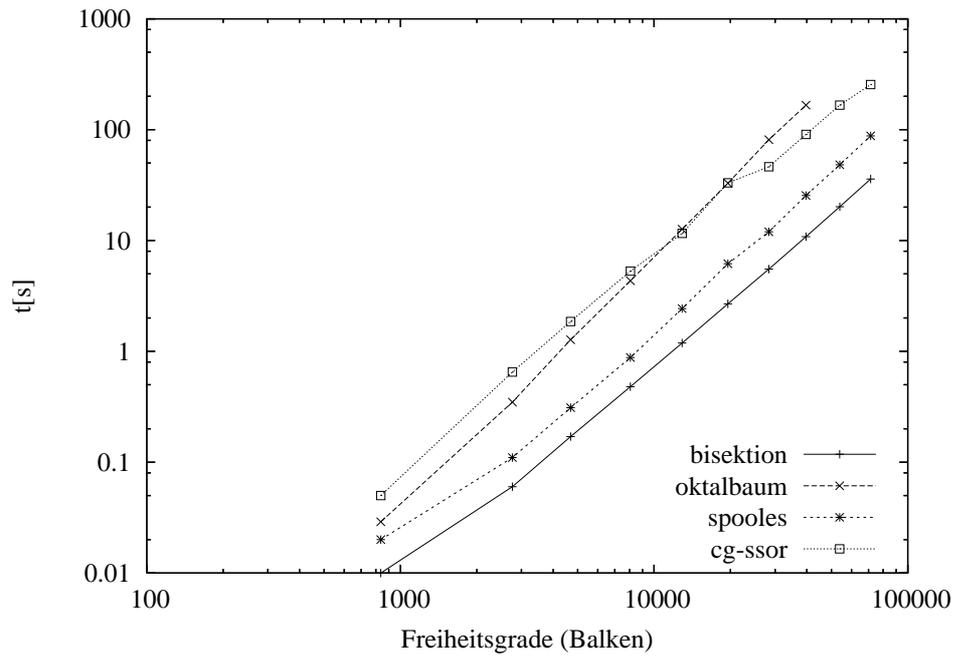


Abbildung 3.25: Vergleich der Laufzeiten unterschiedlicher Lösungsverfahren für das Beispiel des Balkens aus Abbildung 3.14.

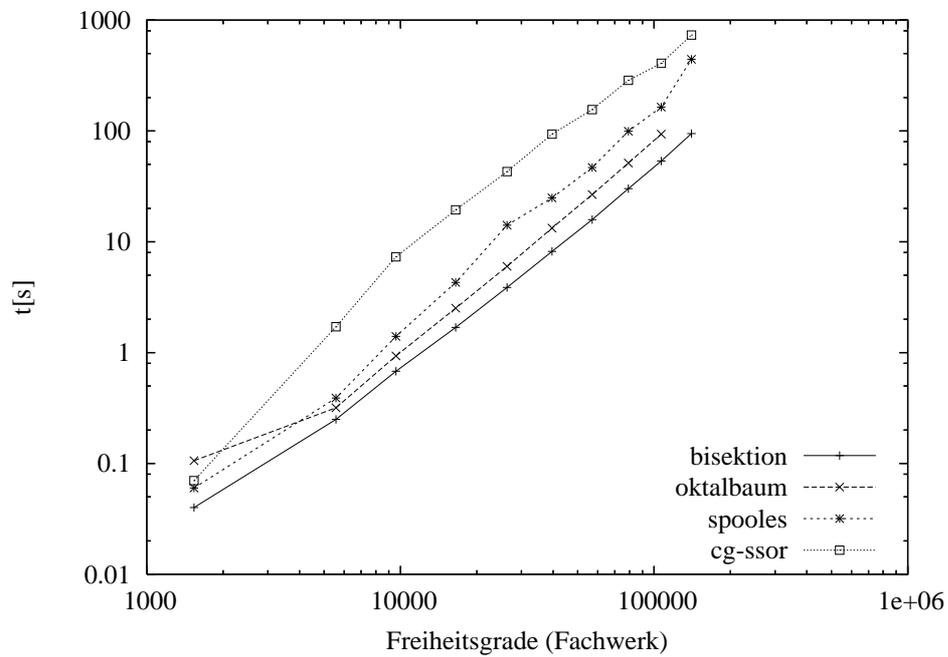


Abbildung 3.26: Vergleich der Laufzeiten unterschiedlicher Lösungsverfahren für das Beispiel des Raumfachwerks aus Abbildung 3.14.

Zusammenfassend ist zu erkennen, dass das hier vorgestellte Konzept der Substrukturierung in der  $p$ -Version der FEM als Löser für dünn besetzte Gleichungssysteme sicherlich konkurrenzfähig zu anderen direkten und - wenn man insbesondere die schlechte Kondition platten- und stabförmiger Strukturen betrachtet - auch iterativen Verfahren ist. Die Effizienz des hier untersuchten Ansatzes wird besonders dann deutlich, wenn man die Rechenzeiten miteinander vergleicht (vgl. Abbildungen 3.23 bis 3.26). Die hohen Leistungswerte bei der rekursiven Substrukturierung führen gerade hier zu besonders niedrigen Laufzeiten.

### 3.4.4.3 Beispiel: UNIQA-Tower

Im Folgenden soll das Lösungsverfahren an einem realistischen Beispiel untersucht werden. Hierzu soll wiederum das schon in Abschnitt 2.3 betrachtete Modell des „UNIQA-Towers“ verwendet werden. Abbildung 3.27 zeigt das Finite-Elemente-Netz von 7 Stockwerken des Turmes zusammen mit dem mittels rekursiver Bisektion in den ersten beiden Hierarchiestufen zerlegte Modell. Es ist dabei deutlich die stark asymmetrische Zerlegung zu erkennen, die gemäß Ab-

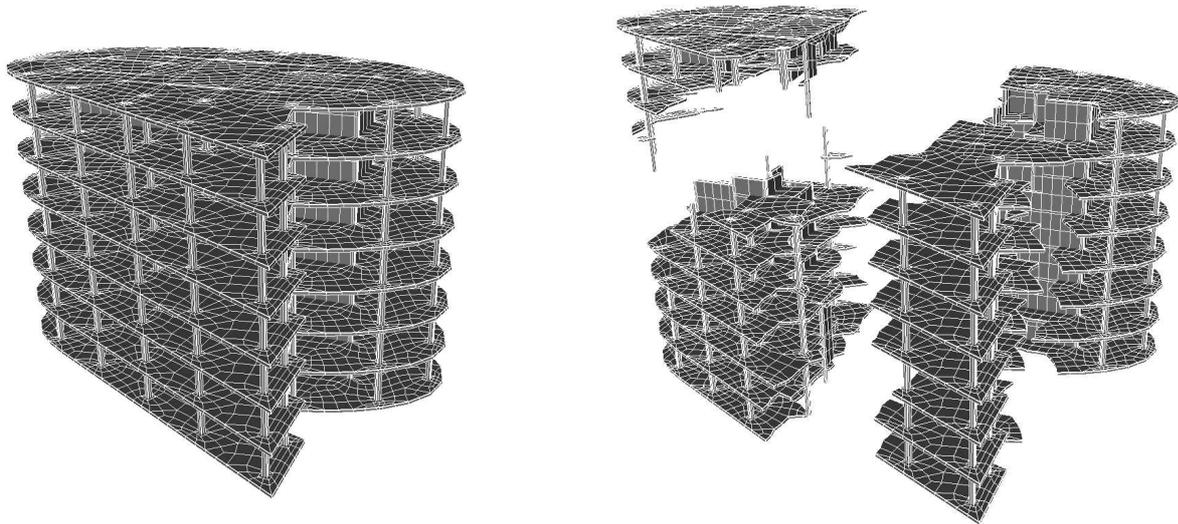


Abbildung 3.27: Finite-Elemente-Modell des UNIQA-Towers bestehend aus 11762 Hexaederelementen. Rechts: Zerlegung in 2 Hierarchiestufen mittels Bisektion.

schnitt 3.4.1.2 aufgrund der zusätzlichen Minimierung des Edge-Cuts nach Gleichung (3.12) resultiert. Tabelle 3.3 vergleicht die Berechnungszeiten sowie Anzahl an Gleitkommaoperationen

$p$	dof	Bisektion		Oktalbaum		Spooles		cg-ssor	
		$t$ [s]	$n_{op}$ [ $10^9$ ]	$t$ [s]	$n_{op}$ [ $10^9$ ]	$t$ [s]	$n_{op}$ [ $10^9$ ]	$t$ [s]	$n_{op}$ [ $10^9$ ]
1	65532	20,1	50,75	33,4	105,23	26,5	24,15	122,0	35,75
2	234396	267,0	807,11	540,2	1746,44	534,4	424,27	1360,3	477,27
3	403260	1048,1	3320,40	2684,3	-	2100,6	1729,72	3751,1	1449,76

Tabelle 3.3: Berechnungszeiten und Anzahl an Gleitkommaoperationen unterschiedlicher Verfahren für 7 Stockwerke des UNIQA-Towers aus Abbildung 3.27.

nen jeweils für die rekursive Bisektion sowie der Oktaalbaumzerlegung. Zusätzlich sind auch die Werte einer Berechnung mit SPOLES und dem iterativen CG-Löser aufgelistet. Aufgrund der Größe des Modells konnten auf dem Rechnersystem mit 8 GB Arbeitsspeicher nur noch Polynomgrade von 1 bis 3 betrachtet werden. Die Abbildungen 3.28 und 3.29 zeigen weiterhin die prozentuale Verteilung aller Freiheitsgrade auf den unterschiedlichen Tiefen des Baumes für Polynomgrade 1 bis 3. Für die rekursive Substrukturierung werden auf den obersten Levels des Baumes nur noch etwa zwei Prozent aller Freiheitsgrade positioniert, die weitgehend größte Anzahl befindet sich etwa ab einer Tiefe von acht.

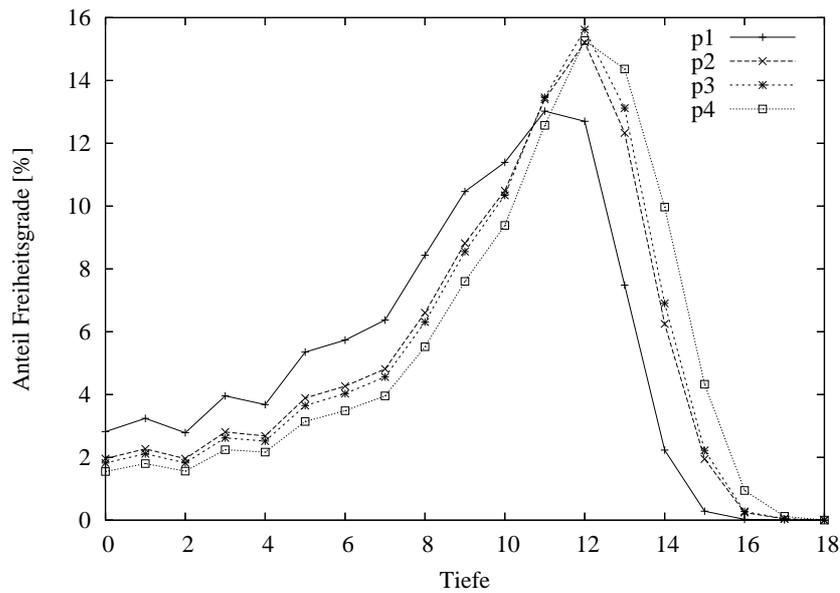


Abbildung 3.28: UNIQA-Tower: Verteilung der Freiheitsgrade auf allen Tiefen des Baumes bei rekursiver Bisektion.

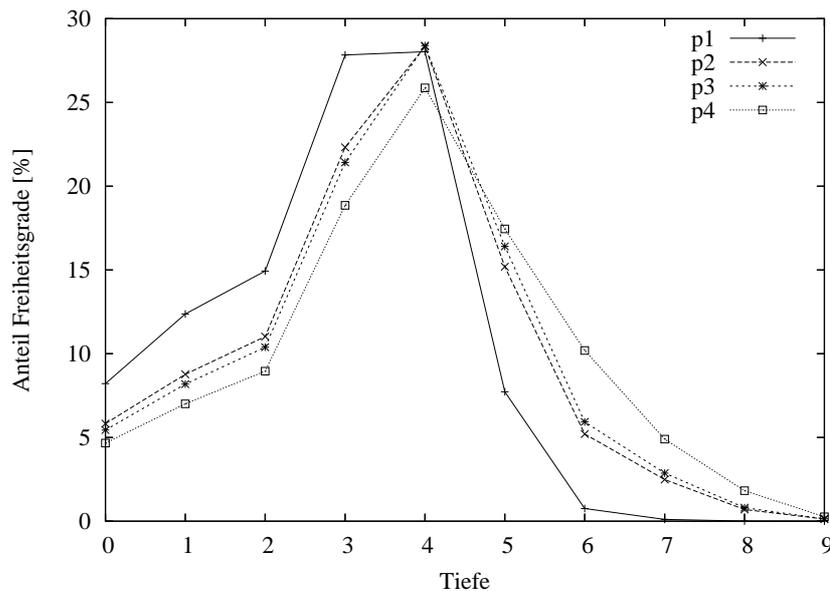


Abbildung 3.29: UNIQA-Tower: Verteilung der Freiheitsgrade auf allen Tiefen des Baumes bei Zerlegung mittels Oktaalbaum.

### 3.4.4.4 Beispiel: Brücke

Als weiteres Beispiel soll das Modell einer Brücke betrachtet werden. Dieses unterscheidet sich vom vorhergehenden durch seine stark anisotropen Strukturen. Abbildung 3.30 zeigt das Finite-Elemente-Netz der Brücke zusammen wiederum mit dem zerlegten Modell.

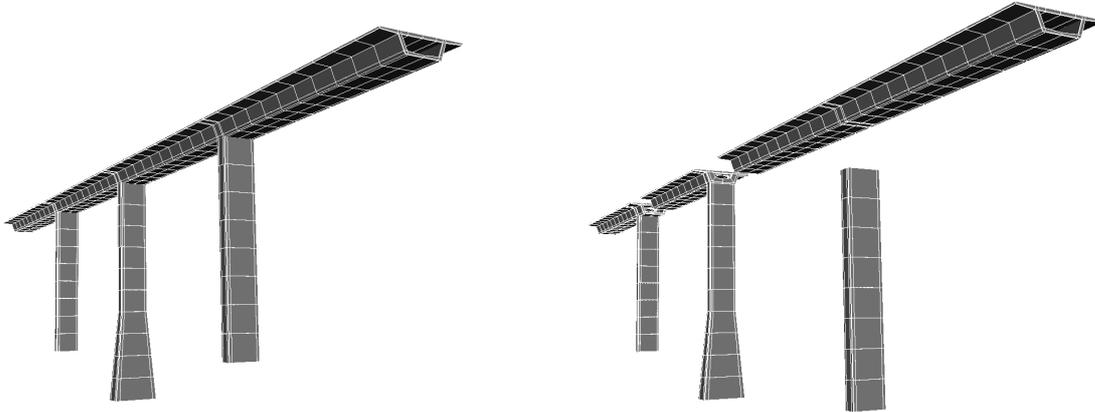


Abbildung 3.30: Finite-Elemente-Modell der Brücke bestehend aus 918 Hexaederelementen. Rechts: Zerlegung in 2 Hierarchiestufen mittels Bisektion.

Tabelle 3.4 listet hierzu die Berechnungszeiten sowie die Anzahl an Gleitkommaoperationen für eine Bisektions- bzw. Oktalbaumzerlegung auf und vergleicht die Ergebnisse mit den anderen Lösungsverfahren. Die Abbildungen 3.31 und 3.32 stellen analog zu vorher die prozentuale Verteilung der Freiheitsgrade auf allen Ebenen des Baumes dar. Im Vergleich zur rekursiven Bisektion werden bei einer Oktalbaumzerlegung wesentlich mehr Freiheitsgrade in den oberen Niveaus des Baumes positioniert. Ist das Modell parallel zu den Koordinatenachsen ausgerichtet, läuft der oberste Separator des Oktalbaums quer durch die gesamte Struktur. Dies erklärt auch die relativ hohen Berechnungszeiten in diesem Falle.

$p$	dof	Bisektion		Oktalbaum		Spooles		cg-ssor	
		$t$ [s]	$n_{op}$ [ $10^9$ ]	$t$ [s]	$n_{op}$ [ $10^9$ ]	$t$ [s]	$n_{op}$ [ $10^9$ ]	$t$ [s]	$n_{op}$ [ $10^9$ ]
1	5556	0,15	0,099	0.56	1,480	0,17	0,075	6,60	2,584
2	19434	1,28	1,662	4.90	16,001	1,65	1,235	83,24	25,293
3	33312	4,12	6,894	17.70	56,172	6,05	5,029	232,71	68,638
4	58248	12,21	23,870	56.40	148,213	20,42	16,779	423,53	179,928
5	94242	30,54	69,453	157.85	378,856	65,48	53,810	1108,47	412,132
6	144048	69,80	175,775	388.91	-	163,77	133,783	2013,35	844,015
7	210420	145,63	398,165	868.98	-	462,29	338,317	3703,32	1564,688
8	296112	288,31	825,445	-	-	960,57	642,786	6396,63	2688,973

Tabelle 3.4: Berechnungszeiten und Anzahl an Gleitkommaoperationen unterschiedlicher Verfahren für das Beispiel der Brücke aus Abbildung 3.30.

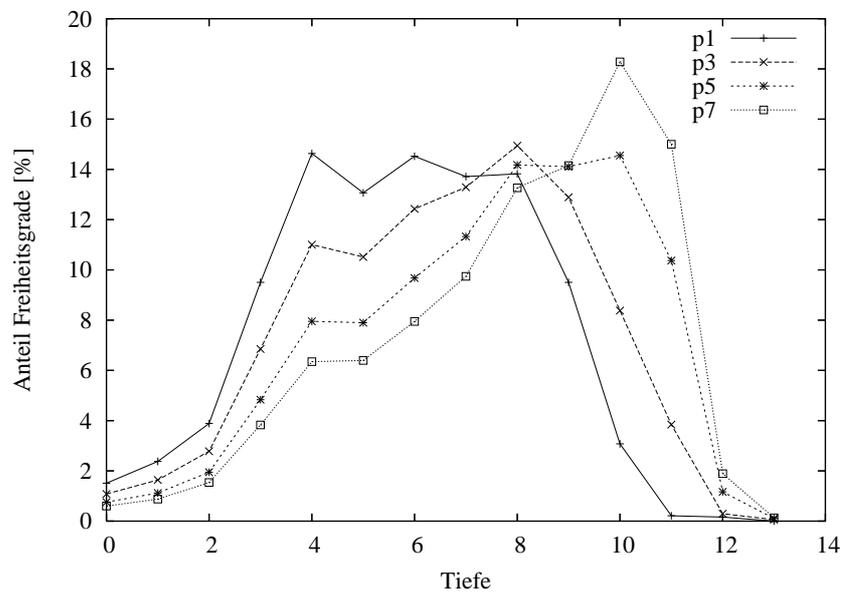


Abbildung 3.31: Beispiel Brücke: Verteilung der Freiheitsgrade auf allen Tiefen des Baumes bei rekursiver Bisektion.

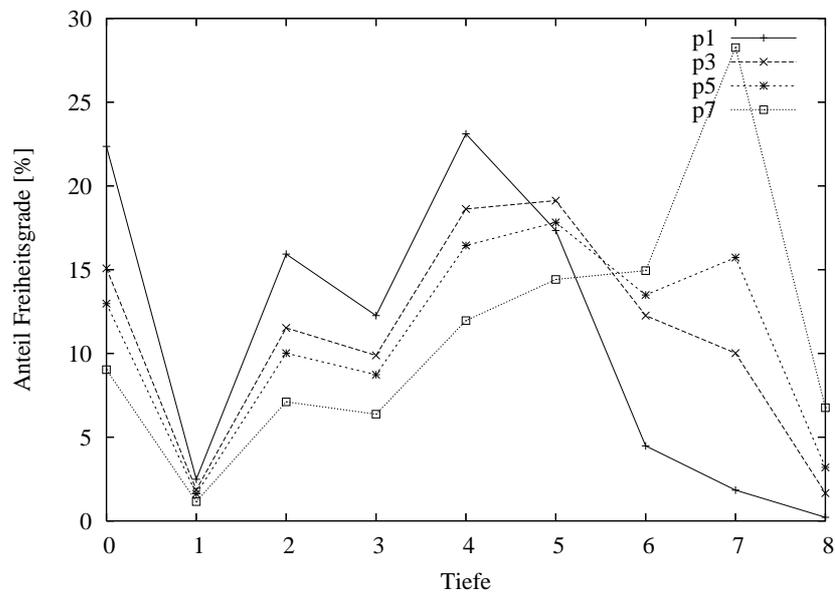


Abbildung 3.32: Beispiel Brücke: Verteilung der Freiheitsgrade auf allen Tiefen des Baumes bei Zerlegung mittels Oktaalbaum.

## 3.5 Vorteile der Hierarchie

Hierarchische Datenstrukturen bieten die Möglichkeit, den Fokus der Computerressourcen schnell auf den zu interessierenden Teil eines Problems zu richten [109]. Diese Eigenschaft lässt sich nun in besonderer Weise bei dem hier betrachteten Verfahren ausnützen, um den Aufwand nach lokalen Modifikationen des Modells zu reduzieren, da bei einer Neuberechnung nur noch Teile des Baumes neu assembliert werden müssen.

Die Zerlegung bietet zudem die Möglichkeit, eine Verteilung von Daten und Aufgaben zu organisieren. Dies lässt sich zum Beispiel bei der Parallelisierung in einem Rechnernetzwerk aber auch zur Speicherung und Verwaltung von Daten auf persistenten Hintergrundspeichern ausnützen. Mithilfe der Hierarchie können die betroffenen Teile schnell lokalisiert und in den Hauptspeicher geladen werden, um zum Beispiel eine Modifikation oder eine Neuberechnung vorzunehmen. Die Konsistenz des gesamten Modells bleibt aber dennoch immer gewährleistet.

Weiterhin spiegeln hierarchische Strukturen auch das Organisationsschema von Arbeitsgruppen wider, in denen mehrere Bearbeiter parallel an einem Modell beschäftigt sind und eine Person die Hauptverantwortung für die gesamte Struktur hat. Teilt man hier das Modell entsprechend den Arbeitsbereichen ein, stellen die Elemente an den Separatoren den Übergangsbereich zu anderen Bearbeiterdomänen dar. Innerhalb jeder Substruktur ist aber eine unabhängige Bearbeitung möglich.

### 3.5.1 Partielle Neuberechnung

Bauteiländerungen beziehen sich oft nur auf begrenzte Bereiche innerhalb eines Modells, etwa die Herausnahme oder die Änderung des Typs einer Stütze oder einer Wand. Obwohl derartige Modifikationen nur einen lokalen Einflussbereich haben, können die Auswirkungen jedoch das *gesamte* statische System betreffen. Um mechanische Inkonsistenzen ausschließen zu können, muss prinzipiell auch nach nur lokalen Änderungen das gesamte Modell neu berechnet werden. Für komplexe Bauwerke kann dieser Aufwand jedoch durchaus erheblich sein. Zusätzlich ist bei einer verteilten Datenhaltung oder bei einer persistenten Speicherung dabei immer das gesamte Modell neu zu laden. Die Hierarchie in der rekursiven Substrukturierung kann nun jedoch gezielt zur Steuerung der Neuberechnung ausgenutzt werden, indem nur betroffene Teilbereiche des Baumes neu assembliert werden. Nachdem in einem initialen Berechnungsdurchlauf alle lokalen Steifigkeiten und Lastvektoren an jedem Knoten des Baumes bekannt sind, muss bei einer Neuberechnung die Assemblierung nur noch an den modifizierten Ästen des Baumes durchgeführt werden. Die restlichen Teile der Struktur werden über ihren Separator repräsentiert und können wiederverwendet werden. Die Lösung der lokalen Unbekannten wird dann wiederum für den gesamten Baum durchgeführt, wobei dieser Aufwand gegenüber der Assemblierung weitgehend zu vernachlässigen ist. Insgesamt kann damit besonders bei mehrmalig wiederholten Änderungen am Modell der Gesamtaufwand während einer FE-Simulation erheblich reduziert werden. Die Berechnungen selbst finden aber zu jedem Zeitpunkt im Kontext des Gesamtsystems statt.

Neben lokalen Modifikationen am Modell lassen sich auch weitere dynamische Änderungen sehr einfach mithilfe der Hierarchie organisieren. Die Unterteilung in Teilstrukturen kann zum Beispiel ausgenutzt werden, um verschiedene Varianten „durchspielen“ zu können, etwa um Konstruktionsvarianten von Brückenquerschnitten zu untersuchen, oder um ein Bauwerk während verschiedener Bauphasen zu simulieren. Die Vorgehensweise ist jedesmal ähnlich.

Bevor die Assemblierung gestartet wird, werden zunächst innerhalb des Baumes betroffene Teilbereiche identifiziert. Falls sich innerhalb eines Unterbaumes nichts ändert, wird dieser bei der Neuberechnung nicht mehr traversiert und das entsprechende Teilgebiet nur über die Separatorgrößen berücksichtigt.

Zusammenfassend lassen sich verschiedene Szenarien identifizieren, die alle dynamische Änderungen des Modells erfordern und für die sich eine Organisation der Berechnung über den Substrukturbaum anbietet:

- Lokale Modifikationen am Modell: Dies können geometrische Änderungen sein, aber auch Änderungen in den Berechnungsparametern wie Materialien, Lasten, Festhaltungen.
- Lokale Detailstudien: Während zunächst das Gesamtmodell relativ grob diskretisiert und berechnet wird, werden kritische Bereiche nochmalig genauer aufgelöst und berechnet. Eine Verbesserung der Genauigkeit des Ansatzes kann durch Erhöhung des Polynomgrades oder durch lokale Verfeinerung des Netzes erfolgen.
- „Aus-“ oder „Einblenden“ von Teilstrukturen: Unterschiedliche Konstruktionsvarianten werden vorbereitet und wie in einer Bibliothek vorgehalten. Die Simulation verschiedener Kombinationen lässt sich dann einfach durch Ein- und Aushängen von Teilbäumen durchführen.
- Berücksichtigung linear-elastischer Bereiche bei nichtlinearen Berechnungen: Bei nichtlinearen Problemstellungen beschränkt sich die Nichtlinearität oft auf einen kleinen Teilbereich innerhalb der Struktur. Die restlichen Bereiche verhalten sich linear und lassen sich über die Separatorgrößen beschreiben. Wenn man bedenkt, dass bei nichtlinearen Berechnungen die Last meist in mehreren Lastschritten aufgebracht wird und in jedem Schritt während einer Iteration das Gleichungssystem mehrmals gelöst werden muss, kann durch eine Verringerung des Aufwands für einen Lösungsdurchgang die Gesamtdauer sicherlich erheblich reduziert werden.
- Darstellung von Bauphasen: Zunächst wird das Bauwerk einmal als Ganzes berechnet und alle Separatorsteifigkeiten der Substrukturen bestimmt. Je nach Bauzeitpunkt werden dann einzelne Teilbereiche ausgeblendet und das Problem neu durchgerechnet. Neben der Reduktion des Berechnungsaufwandes kommt hier auch der Steuerung der Simulation von Seiten des Benutzers eine besondere Bedeutung zu. Da Bauzeiten meist mithilfe von Balkenplänen dargestellt werden, ist eine Steuerung über ein Projektplanungswerkzeug wünschenswert. Ein Prototyp einer solch integrierten Simulationsumgebung wird in dieser Arbeit in Abschnitt 4.2 vorgestellt.

Diese verschiedenen Aufgabenstellungen können im Rahmen einer hierarchischen Sortierung alle in ähnlicher Art und Weise behandelt werden. Die Knoten im Baum werden dazu hinsichtlich drei verschiedener Attribute klassifiziert: *sichtbar*, *unsichtbar*, *modifiziert*. Nachdem für alle finiten Elemente an den Blattknoten im Baum eine entsprechende Einteilung getroffen wurde, lässt sich die Assemblierung der Steifigkeiten nun mithilfe dieser drei Attribute steuern. Das Vorgehen geschieht in gleicher Weise wie in Abschnitt 3.4.3 beschrieben, der Prozess wird aber an jedem inneren Baumknoten zusätzlich hinsichtlich der Attributierung gesteuert. Algorithmus 3.2 erklärt das prinzipielle Vorgehen. Dieses ist zudem in Abbildung 3.33 illustriert.

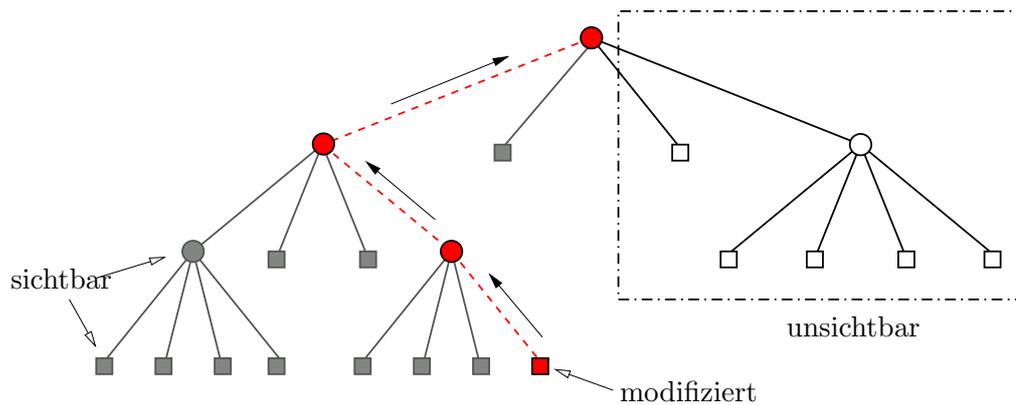


Abbildung 3.33: Partielle Assemblierung nach lokaler Modifikation. Teile des Baumes bleiben „ausgeblendet“.

---

**Algorithmus 3.2** Steuerung der Lösung in unterschiedlichen Szenarien.

---

1. Alle Blätter werden hinsichtlich {sichtbar, unsichtbar, modifiziert} klassifiziert.
  2. Postorder-Traversierung über alle inneren Knoten im Baum.  
Für jeden Knoten:
    - (a) Sind *alle* Söhne sichtbar?  
→ markiere Knoten als sichtbar.
    - (b) Sind *alle* Söhne unsichtbar?  
→ markiere Knoten als unsichtbar.
    - (c) Sonst:  
markiere Knoten als modifiziert und assembliere die Steifigkeiten aller sichtbaren und modifizierten Knoten.
- 

### 3.5.1.1 Beispiel: Veränderung eines Brückenpfeilers

An einem Beispiel sollen nun die Vorteile des hierarchischen Lösungsverfahrens nach einer lokalen Änderung eines Modells erläutert werden. Abbildung 3.34 zeigt hierzu folgendes Szenario: Ein Pfeiler der schon in Abschnitt 3.4.4.4 betrachteten Brücke wird im unteren Bereich etwas verbreitert, um beispielsweise eine bessere Standfestigkeit gegenüber horizontalen Lasten zu gewährleisten. Dazu wurde das Finite-Elemente-Netz geometrisch modifiziert. Tabelle 3.5 vergleicht nun die Zeiten für eine initiale Berechnung des gesamten Systems mit denen einer Neuberechnung nach der Modifikation für steigende Polynomgrade  $p$ . Die Zeiten sind dabei zusätzlich hinsichtlich der bottom-up Assemblierungs- und der top-down Lösungsphase unterschieden. Als hierarchisches Organisationsschema wurde eine rekursive Bisektion gewählt.

Es wird deutlich, dass für derartige anisotrope Strukturen mit relativ stark entkoppelten Gleichungssystemen nach einer lokalen Modifikation erhebliche Zeiteinsparungen möglich sind. Auch für größere Probleme können hier Antwortzeiten von wenigen Sekunden erwartet werden. Dies erlaubt durchaus eine interaktive Arbeitsweise.

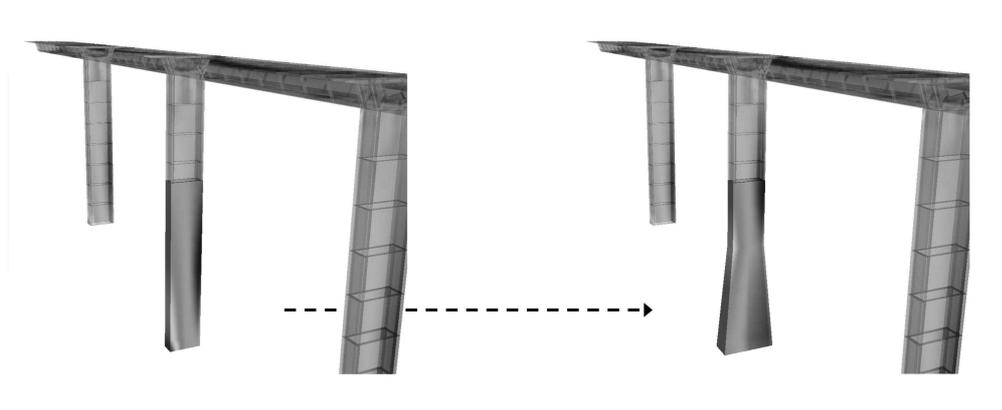


Abbildung 3.34: Veränderung eines Brückenpfeilers. 32 Elemente werden geändert.

$p$	dof	Initiale Berechnung		Neuberechnung	
		Assmbl	Lösung	Assmbl	Lösung
1	5556	0,171	0,0070	0,0040	0,0070
2	19434	1,310	0,0370	0,0410	0,0370
3	33312	4,002	0,0840	0,1350	0,0830
4	58248	11,810	0,1890	0,3889	0,1880
5	94242	29,773	0,3769	0,9919	0,3749
6	144048	69,874	0,8479	2,3716	0,8459

Tabelle 3.5: Berechnungszeiten  $t[s]$  (Assemblierung bzw. Lösung) für die Gesamtstruktur sowie nach Modifikation des Pfeilers.

### 3.5.1.2 Beispiel: UNIQA-Tower

Als zweites Beispiel soll eine partielle Neuberechnung an einem Modell mit 5 Stockwerken des schon in Abschnitt 3.4.4.3 betrachteten UNIQA-Towers betrachtet werden. Ein Nachteil des hier untersuchten Vorgehens ist der relativ hohe Speicherverbrauch. Werden zur Neuberechnung die lokalen Steifigkeitsmatrizen an allen Knoten der Hierarchie gespeichert, können auf dem Rechnersystem mit 8 GByte Arbeitsspeicher nur Modelle mit maximal etwa 200000 Freiheitsgrade berechnet werden. Es müssen aber grundsätzlich nicht immer alle Steifigkeiten ständig im Arbeitsspeicher vorgehalten werden. Diese können auf externe Speichermedien ausgelagert werden, insbesondere dann, wenn verschiedene Bearbeiter in zeitlichen Abständen auf das Modell zugreifen. Zur Organisation der (Neu-)Berechnung müssen dann allerdings intelligente Konzepte eines Speichermanagements angewendet werden. Dieses wurde in der vorliegenden Arbeit aber nicht untersucht.

Tabelle 3.6 soll nun prinzipiell die Zeitersparnis in einem Szenario aufzeigen in welchem ein Bearbeiter ein lokales Teilgebiet bearbeitet, modifiziert und wiederkehrend Neuberechnungen anstößt. Hierzu wurde an dem Modell des Towers bestehend aus 8313 Hexaederelementen zunächst eine Berechnung der gesamten Struktur durchgeführt und dann in einem zweiten Schritt die Steifigkeiten von vier Elementen geändert und eine Neuberechnung gestartet. Die Substrukturhierarchie wurde in diesem Beispiel mithilfe einer Oktaalbaumzerlegung aufgebaut.

Im Gegensatz zum vorherigen Beispiel in Abschnitt 3.5.1.1 ist in diesem Beispiel die Re-

$p$	dof	Initiale Berechnung		Neuberechnung	
		Assmbl	Lösung	Assmbl	Lösung
1	46626	9,93	0,126	2,55	0,130
2	166437	152,94	1,245	48,48	1,193

Tabelle 3.6: Berechnungszeiten  $t[s]$  (Assemblierung bzw. Lösung) für die Gesamtstruktur sowie nach Modifikation mehrerer Elemente.

duzierung des Rechenaufwandes bei einer lokalen Modifikation nicht mehr so deutlich. Dies liegt an der weitaus kompakteren Struktur des Modells. Die Matrizen in den oberen Bereichen der Hierarchie sind in diesem Beispiel wesentlich größer, was sich dann vor allem bei einer Neuberechnung bemerkbar macht.

### 3.5.2 Verteilung von Daten und Berechnung

Im Gegensatz zu der klassischen Vorgehensweise in der Finite-Elemente-Methode erlaubt das Verfahren der Substrukturierung eine vermehrte Nebenläufigkeit von Prozessen. Nach der Zerlegung eines Modells kann die Berechnung der Elementsteifigkeiten, weite Teile des Lösungsverfahrens und das Postprocessing auf einer lokalen Ebene durchgeführt werden. Das Gesamtsystem muss dabei an keiner Stelle mehr im Ganzen betrachtet werden. Innerhalb des Lösungsverfahrens wird dieses nur über die Separatorsteifigkeiten auf einer globalen Ebene erfasst. Dadurch eröffnen sich Möglichkeiten zu weitergehenden Konzepten der Verteilung. Dies kann einerseits eine Verteilung von Berechnungsaufgaben im Rahmen einer Parallelisierung sein, andererseits lassen sich aber auch Daten weitgehend unabhängig voneinander auf verteilten Rechnern speichern und bearbeiten.

Insbesondere bei der  $p$ -Version der FEM bietet sich eine Parallelisierung des Verfahrens an. Für höhere Polynomgrade nimmt der Anteil des Aufwandes zur Berechnung der Elementsteifigkeiten zu. Da die Integration der Elementsteifigkeiten unabhängig voneinander auf lokaler Ebene abläuft, lässt sich diese optimal auf mehrere Rechner verteilen. In einer Arbeit von RÜCKER [105] wird dies ausgenutzt und ein Konzept zur Parallelisierung der  $p$ -Version der FEM vorgestellt. Dabei werden in ähnlicher Weise wie in dieser Arbeit auf Elementebene die inneren Freiheitsgrade (Bubble-Modes) eliminiert und nur die äußeren zum Gesamtgleichungssystem zusammengesetzt. Dieses Vorgehen entspricht prinzipiell einer einstufigen Substrukturierung.

Durch die Hierarchie ergeben sich bei der rekursiven Substrukturierung weitergehende Möglichkeiten der Parallelisierung des Lösungsverfahrens. Der Assemblierungs- sowie der Lösungsprozess von zwei disjunkten Unterbäumen lässt sich prinzipiell unabhängig voneinander durchführen. Von AMESTOY et al. [2, 3] wird ein derartiges Vorgehen im Rahmen von Multifrontallösungsverfahren untersucht, um die Lösung dünn besetzter Gleichungssysteme auf massiv parallelen Systemen zu ermöglichen<sup>13</sup>. Ähnlich wie bei der rekursiven Substrukturierung wird in Frontalverfahren das Gleichungssystem nicht sofort als Ganzes assembliert, sondern die Assemblierung findet gleichzeitig mit der Elimination unabhängiger Freiheitsgra-

<sup>13</sup>MUMPS: a MULTifrontal Massively Parallel sparse direct Solver:  
<http://www.enseiht.fr/irit/apo/MUMPS/>

de innerhalb einer oder mehrerer dichtbesetzter, so genannter Frontmatrizen statt [34, 33]. Da der Berechnungsaufwand in einzelnen Teilgebieten ganz unterschiedlich sein kann, müssen für die Parallelisierung aber zunächst die Teilbäume der Assemblierung in geeigneter Weise auf die Rechner im Netzwerk verteilt werden. Hierzu schlagen die Autoren ein Verfahren zur Lastverteilung vor, mit dem sich eine weitgehend ausgewogene Verteilung sowohl des Berechnungs- als auch des Speicheraufwandes ergibt. Es wird dabei zunächst für jeden Knoten im Baum a priori der Berechnungsaufwand abgeschätzt. Basierend auf einem von GEIST und NG [50] vorgestellten Algorithmus werden dann in einer Suchwelle von der Wurzel ausgehend Unterbäume unter Berücksichtigung ihres Berechnungsaufwandes geeignet zusammengefasst und auf eine gegebene Anzahl von Prozessen verteilt. In den Lastverteilungsalgorithmus können dabei zusätzlich noch weitere Randbedingungen mit einfließen wie beispielsweise die ganz unterschiedlichen Kommunikationskosten in Netzwerken mit verteilten Mehrprozessorsystemen [4].

Auf ein Problem weisen AMESTOY et al. jedoch hin: Besonders in den oberen Niveaus des Baumes ist oft der größte Aufwand zur Berechnung notwendig. Nach oben hin nimmt jedoch der Grad der Verteilung ab. Als Lösung schlagen sie vor, neben einer Verteilung des Baumes zusätzlich eine Parallelisierung der Matrix-Operationen innerhalb der Knoten vorzunehmen. Mit dieser Kombination von Verfahren lässt sich damit auf Systemen mit 32 Prozessoren ein Speed-Up von etwa 10 [3] beziehungsweise auf 128 Prozessoren von etwa 35 erzielen [4].

Neben einer reinen Parallelisierung bietet es sich mit dem hier gezeigten Ansatz an, Konzepte einer verteilten Datenhaltung und Berechnung in Betracht zu ziehen. Dies ist insofern sinnvoll, da Kommunikation im Wesentlichen nur zweimal während des gesamten Lösungsverfahrens notwendig ist. Zum einen während der Assemblierung, wenn die Schur-Komplemente von einem Rechner an den anderen weitergereicht werden, sowie während der Lösung, wenn die Ergebnisse zurückgegeben werden. Abbildung 3.35 zeigt hierzu schematisch die Verteilung des Baumes in einer Client-Server Umgebung. Die Berechnungen aller Unterbäume findet jeweils auf den Clients im Netzwerk statt. Auf dem Master-Knoten selbst wird nur noch das globale Separatorproblem verwaltet und gelöst.

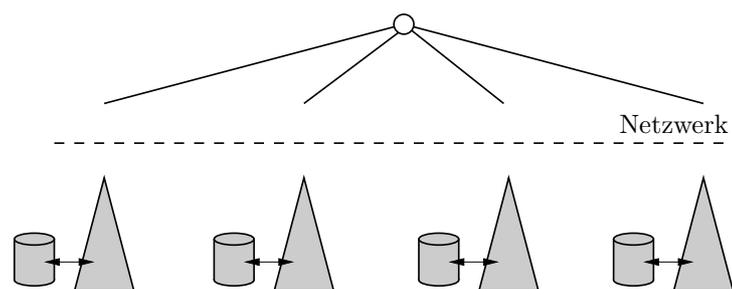


Abbildung 3.35: Verteilung von Berechnung und Datenhaltung [82]. Teilgebiete werden auf Rechner im Netz ausgelagert und dort vorgehalten. Bei Änderungen werden nur Separatorgrößen nach oben gegeben und neue Lösungsvektoren empfangen.

Lokale Daten eines Teilgebiets, wie Netz, Steifigkeiten oder Berechnungsergebnisse können während der gesamten Bearbeitungsdauer unabhängig voneinander auf Rechnern im Netz vorgehalten werden. Durch die gezielte Steuerung der Berechnung ist es dabei nicht notwendig, alle Ressourcen ständig gebunden zu halten. Da nach Modifikation des Modells meist nur ein Teil der Baumstruktur betroffen ist, genügt es, nur diesen auf dem entsprechenden Rechner

in den Arbeitsspeicher zu laden, während die übrigen Teile auf einer Datenbank ausgelagert bleiben. Während der gesamten Laufzeit muss somit nur die Baumstruktur selbst in einem zentralen Rechner ständig vorgehalten werden. Bei Bedarf werden von diesem gesteuert die benötigten Teile auf den entfernten Systemen nachgeladen, verarbeitet und anschließend wieder zurückgespeichert. Derartige Prozesse lassen sich zudem weitgehend transparent für den Benutzer umsetzen. Ein Vorschlag für ein Konzept basierend auf Grid-Diensten findet sich hierzu in [82].

# Kapitel 4

## Kooperatives Arbeiten am dreidimensionalen Modell

Ein Ziel in der Planung von Bauwerken ist die Realisierung einer durchgehend virtuellen Produktentwicklung, in einer Form wie sie beispielsweise im Flugzeugbau durchaus schon zu finden ist [45]. Im Zentrum steht dabei ein dreidimensionales Computermodell, welches um weitere Informationen wie Kosten oder Zeit angereichert ist und damit die Möglichkeit bietet, besonders die Zusammenarbeit verschiedener Fachdisziplinen unterstützen zu können. Eine derartige Integration verschiedener Entwurfsperspektiven auf Basis eines gemeinsamen Bauwerksmodells wird in der Forschung auch mit dem Begriff *n*-dimensionale Modellierung (*nD-modelling*) [6] bezeichnet. Ein Planungsmodell soll gleichzeitig hinsichtlich *n* verschiedener Dimensionen wahrgenommen werden können. Der Nutzen dieser integrierten Herangehensweise wird dabei zunehmend auch in der Praxis erkannt [59]. So lassen sich auf Grundlage des virtuellen Computermodells Visualisierungen für Marketing und Vertrieb erstellen oder im Vorfeld Fragen der Nutzung klären, wie etwa mit einer Simulation von Besucherströmen oder der Ermittlung des Energieverbrauchs. Auch lassen sich geometrische Daten auswerten, um Mengen zu ermitteln oder Kosten abschätzen zu können. Insbesondere durch Hinzunahme von zeitlichen Informationen zum dreidimensionalen Bauwerksmodell, was gemeinhin als *4D-Modellierung* oder *4D-CAD* [74] bezeichnet wird, lassen sich Planungsumgebungen schaffen, in denen komplexe Sachverhalte vermittelt und die interdisziplinäre Zusammenarbeit zwischen Architekten, Ingenieuren, Investoren und Nutzern eines Bauwerks unterstützt werden kann. Als Folge dieser engen Zusammenarbeit treten nun aber verstärkt Fragen der Konsistenthaltung sowie der Organisation konkurrierender Zugriffe auf einen gemeinsamen Datenbestand in den Vordergrund.

Im Zusammenhang derartiger Fragestellungen wurde im Rahmen des in der Einleitung erwähnten Forschungsprojekts [99] ein entsprechendes Software-Konzept zur Unterstützung einer kooperativen Arbeit am Volumenmodell vorgestellt [78, 79, 80, 86, 85]. Neben einem verteilten Zugriff auf eine gemeinsame Datenbasis und der damit verbundenen Konsistenzsicherung standen dabei vor allem Fragen der Integration von Simulationsaufgaben im Mittelpunkt. Es wurde untersucht, wie sich Oktalbäume nutzen lassen, um damit einerseits ein Modell geometrisch hierarchisch zu organisieren und gleichzeitig numerische Simulationverfahren effektiv unterstützen zu können. Dieses Software-Framework zur verteilt kooperativen Arbeit soll an dieser Stelle genauer erläutert werden. Das Konzept wird in einem folgenden Abschnitt dann aufgegriffen und zur Simulation von Bauphasen genutzt. Am Beispiel eines Szenarios

aus dem Brückenbau soll deutlich werden, wie sich durch Einbindung einer Prozessplanungsanwendung und unter Ausnutzung der hierarchischen Struktur des Finite-Elemente-Modells ein interaktives Werkzeug zur schnellen Analyse von Bauzuständen realisieren lässt.

## 4.1 Ein Framework für verteilt kooperatives Arbeiten

Personen im Bauplanungsprozess arbeiten in der Regel räumlich und zeitlich verteilt an einem gemeinsamen Projekt. In [78, 79] wird zur Unterstützung dieser kooperativen Arbeit ein Software-Rahmenwerk vorgestellt, welches die Integration von Simulationsaufgaben in einer verteilten Umgebung erlauben soll. Die Struktur des Systems entspricht dabei einer klassischen Client-Server-Architektur, mit zentraler Projektdatenbasis und verteilt arbeitenden Clients. Im Gegensatz zu einer reinen Datenbankanwendung, die eine Einbindung von Applikationen auf Basis eines Datenaustausches erlaubt, sollen hier jedoch Simulationsanwendungen direkt innerhalb der zentralen Komponente platziert werden. Diese so genannte Funktionsintegration oder *Einbettung* bietet den Vorteil, dass zentrale Ressourcen gemeinsam genutzt werden können und der Informationsfluss und Datenaustausch zwischen den intern befindlichen Prozessen wesentlich verbessert werden kann [79]. Abbildung 4.1 zeigt das entsprechende Schema. Die zentrale Komponente soll im Weiteren auch als Modellserver bezeichnet werden.

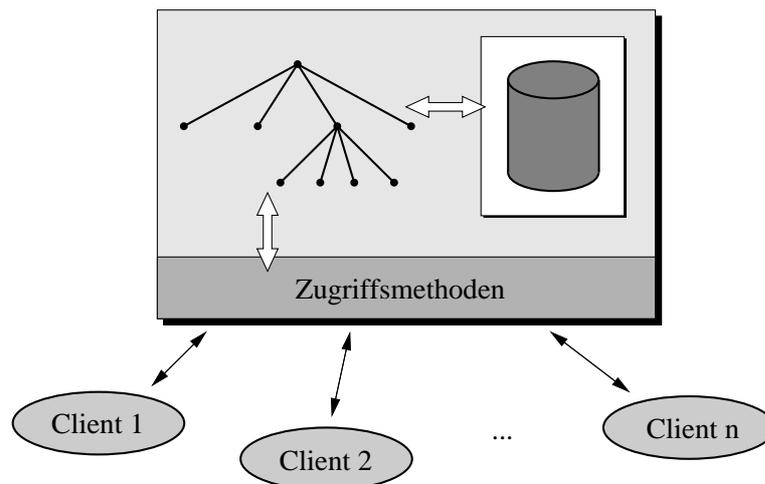


Abbildung 4.1: Schematische Darstellung des Modellserver nach BUNGARTZ und MUNDANI [78, 79]. Ein relationales Datenbankmanagementsystem wird durch einen Kontrollbaum erweitert, welcher geometrische Objekte räumlich-hierarchisch organisiert.

Im Zentrum des Modellserver steht ein relationales Datenbankmanagementsystem zur persistenten Speicherung des Gesamtmodells als attributierter *vef*-Graph<sup>1</sup>. Die Primärschlüssel der Datenbankeinträge sind dabei zusätzlich innerhalb eines Kontrollbaumes organisiert, der so einen Raumbezug für die in der Datenbank gespeicherten Objekte herstellen kann. Der direkte Zugriff auf die zentrale Datenbasis seitens der Clients ist nicht erlaubt. Dieser kann nur über vom Modellserver bereitgestellten Methoden erfolgen. Die Zugriffsmethoden sind dabei

<sup>1</sup>*vertex, edge, face*-Graph zur Beschreibung der topologischen Abhängigkeiten zwischen Knoten, Kanten und Flächen eines Körpers in Oberflächendarstellung; siehe etwa in [22].

fest mit dem internen Kontrollbaum gekoppelt, der so eine zusätzliche Schicht um die zentrale Datenbasis bildet. Durch diese Schicht wird die Integrität der Daten geschützt, indem ein vorgelagerter geometrischer Konsistenztest (vgl. [79] bzw. Abschnitt 3.2.2.2) geänderte beziehungsweise neu hinzugekommene Objekte mithilfe eines Oktalbaum-basierten Tests auf Überschneidungen und geringfügige Spalten überprüft und gegebenenfalls zurückweist.

#### 4.1.1 Kontrollbaum zur räumlichen Organisation

Bauwerksmodelle beinhalten in der Regel keinerlei oder nur geringe Information über räumliche Zusammenhänge und Abhängigkeiten ihrer Geometrie. Durch die hierarchische Organisation der einzelnen Bauteile mittels Oktalbaum lässt sich ein derartiger Zusammenhang herstellen und für Fragestellungen unter Einbeziehung eines räumlichen Kontextes nutzen. Dazu wird nach [78, 79] für alle Bauteile ein Oktalbaum aufgespannt, in dem nach dem in Abschnitt 3.2.2.3 beschriebenen Schema des MX-CIF Quadrees [110], ein geometrisches Objekt genau der Oktalbaumzelle zugeordnet wird, welche dieses gerade noch komplett umschließt. Innerhalb des Baumes werden dabei die Primärschlüssel der relationalen Datenbankeinträge gespeichert, um einen schnellen Zugriff auf alle weiteren Bauwerksinformationen zu erhalten. Auf Basis dieser räumlich hierarchischen Strukturierung lassen sich nun Fragestellungen beantworten, die über die rein datenbanktypischen, relationalen Anfragen hinausgehen, wie etwa die Suche nach benachbarten Objekten oder die Lage zweier Objekte zueinander. Neben solcherart *horizontalen* Anfragetypen, die primär auf einer gleichen, horizontalen Ebene des Baumes beantwortet werden können, lassen sich nach MUNDANI [79] mithilfe von Oktalbäumen aber auch so genannte *vertikale* Relationen beantworten, die Aspekte von Teilmengeneigenschaften beinhalten („Welche Objekte befinden sich innerhalb eines bestimmten Raumes“) und bei dem der Baum in erster Linie vertikal durchsucht wird.

Neben den rein technischen Gesichtspunkten sind bei einer Erweiterung eines Modells um räumliche Zusammenhänge aber auch die Schnittstellen zwischen Datenbasis und Benutzer zu erweitern, um eine einfachere Navigation und Informationsgewinnung in einem komplexen Bauwerksmodell nach räumlichen Kriterien zu ermöglichen. BORRMANN et al. [16] stellen hierzu auf Basis der in Datenbanken üblicherweise verwendeten *Structured Query Language, SQL* ein Konzept einer räumlichen Anfragesprache (*Spatial Query Language*) für Bauwerksinformationsmodelle vor. Zur Definition einer eindeutigen Semantik der Anfragesprache werden hierzu die vorkommenden Datentypen und Operationen zunächst formal beschrieben und klassifiziert. Es werden vier Typen unterschieden:

- **Direktionale Operationen:** Diese bestimmen relative Positionen zweier Objekte zueinander, wie z.B. *oben, unten, nördlich, südlich*.
- **Topologische Operationen:** Topologische Beziehungen können als „Nachbarschaftsbeziehungen“ bezeichnet werden und ergeben sich prinzipiell aus dem Ergebnis einer Schnittoperation von zwei Körpern. Da ein Objekt grundsätzlich den Raum in ein Inneres, einen Rand und ein Äusseres teilt, können als Ergebnis dieser Operation neun verschiedene Fälle auftreten. Diese lassen sich anschaulich in einer  $3 \times 3$  Matrix anordnen.
- **Metrische Operationen:** Metrische Operationen liefern den Abstand zweier Körper. In der Regel wird dabei der minimale Abstand der Punktmenge zweier Objekte bestimmt.

Es lassen sich aber auch andere „Typen“ von Abstand klassifizieren. Das Ergebnis der Operation ist ein numerischer Wert.

- Boolesche Operationen: Die Booleschen Operationen *union*, *intersection* und *difference* entsprechen den klassischen Verfahren in volumenorientierten CAD-Systemen. Das Ergebnis einer booleschen Operation ist wieder ein geometrisches Objekt.

In einer weiteren Arbeit von BORRMANN [15] werden diese räumlichen Anfrageoperationen präzisiert und zudem ein Konzept zu ihrer Umsetzung auf Basis objektrelationaler Datenbanken vorgestellt.

Über die rein geometrischen Fragestellungen hinaus bietet das in Abbildung 4.1 dargestellte Framework gerade durch die Verbindung des Oktalbaums mit einer Datenbank besondere Vorteile. Es lassen sich Aufgaben lösen, in denen semantische Informationen innerhalb der Datenbank mit einem geometrischen Kontext verbunden werden. Fragen die hierbei von Interesse sein können, lauten beispielsweise „Gib mir alle Objekte mit dem Attribut 'tragend', die zu einer bestimmten Wand benachbart sind.“ oder „Stelle ein bestimmtes Stockwerk von Höhe  $h=10\text{m}$  bis  $h=15\text{m}$  dar, blende aber alle Fensterelemente aus.“. Ähnliche Verknüpfungen von geometrischen mit semantischen Daten stellen beispielsweise in Geoinformationssystemen eine wichtige Grundlage der Arbeit dar („Welche Städte mit mehr als 100.000 Einwohner befinden sich im Umkreis von 100km?“), werden aber sicherlich auch im Bereich der Bauwerks- und Gebäudemodellierung immer wichtiger.

### 4.1.2 Regulierung von Zugriffen

Zur Regulierung der Zugriffe auf Seiten des in Abbildung 4.1 dargestellten Modellservers kommt ein Schema zum Einsatz, welches sich prinzipiell an das in der Software-Entwicklung verwendete *Concurrent Versions Systems*, CVS orientiert. CVS wird dort üblicherweise zur Versionsverwaltung von Quellcode eingesetzt. Die gesamten Daten eines Projekts sind dabei an einer zentralen Stelle, dem so genannten *Repository*, gespeichert, von dem Benutzer sich Teile lokal kopieren und nach Modifikation wieder dem zentralen Datenbestand hinzufügen können [23]. Im Gegensatz zu CVS geschieht bei dem hier betrachteten Framework die Konsistenthaltung des zentralen Datenbestandes allerdings nicht auf Basis einzelner Dateien, sondern auf Grundlage geometrischer Objekte. Es wird ein Körper nur dann dem zentralen Datenbestand hinzugefügt, wenn dieser sich geometrisch konfliktfrei (d.h. ohne Überlappungen oder geringe Spalten) in das Gesamtmodell einfügen lässt.

Im Gegensatz zu einer Versionsverwaltung wie in der Softwareentwicklung wird in dem hier vorgestellten System keine Versionierung von Daten angeboten. Es existiert immer nur ein aktueller Stand innerhalb der Datenbank, frühere Versionen des Modells können nicht wiederhergestellt werden. Dies stellt sicherlich eine Einschränkung dar. Für vergleichende Analysen oder Modellstudien müssen in der Regel Varianten erzeugt werden, die man unter Umständen auch wieder verwerfen können muss. Auch um Planungsfehler rückgängig zu machen, muss das System in der Lage sein, eine alte Version wiederherstellen zu können. Derartige Problemstellungen wurden im Rahmen des hier betrachteten Frameworks ausgeklammert. Es soll an dieser Stelle auf die Arbeiten der Gruppe um FIRMENICH [49] verwiesen werden, in denen auf Basis formaler Methoden der Mengenalgebra Konzepte zur Versionierung von CAD-Modellen erarbeitet werden.

Für den Zugriff auf Objekte innerhalb der zentralen Datenbasis seitens der Clients werden hierzu dem Benutzer ähnlich wie in CVS drei Möglichkeiten angeboten:

**read:** Der Benutzer wird für ein bestimmtes Objekt als nur lesend vermerkt.

**read-write:** lesender und schreibender Zugriff; d.h. eine Kopie kann vom Benutzer verändert und gegebenenfalls auf den Modellserver zurückgeschrieben werden.

**exclusive-write:** Das Objekt wird auf dem Sever „gesperrt“ (exklusives Locking). Alle anderen Benutzer haben nur noch lesenden Zugriff auf das Objekt.

Mit der Möglichkeit, einen „read-write“ Zugriff zuzulassen, wird die Philosophie vertreten, dass Objekte nicht jedesmal zu sperren sind, wenn diese bearbeitet werden sollen. In Planungsaufgaben kann die Bearbeitung von Teilmengen des Modells lokal bei einem Benutzer durchaus sehr lange dauern (sog. „*Long Transactions*“). Es ist daher zu erwarten, dass eine strikte Sperrung von Objekten die Arbeit anderer Teilnehmer zu stark einschränken würde. Aus diesem Grunde wird gewöhnlicherweise ein lesender und schreibender Zugriff erfolgen, der natürlich das Problem beinhaltet, dass zwei Benutzer ein Objekt unabhängig voneinander bearbeiten und dem zentralen Datenbestand wieder hinzufügen möchten. In diesem Falle muss der entstehende Konflikt von den betreffenden Personen manuell gelöst werden und ein konsistenter Zustand gemeinsam wiederhergestellt werden (sog. „*Merge*“ [23]). Zur Unterstützung werden hierzu Benachrichtigungsdienste bereitgestellt, die dem Benutzer erlauben, stets über die Aktivitäten der anderen informiert zu bleiben, um gegebenenfalls darauf reagieren zu können (siehe hierzu unten). In diesem Zusammenhang ist zu bemerken, dass das hier beschriebene System nur Hilfsmittel für die kooperative Arbeit bereitstellen soll, Fehler können damit nicht grundsätzlich ausgeschlossen werden. Es ist daher immer anzuraten, schon im Vorfeld einer Maßnahme alle Aktivitäten und Zuständigkeiten der Projektbeteiligten entsprechend zu koordinieren, um einen reibungslosen Arbeitsablauf gewährleisten zu können. Die Kollisionsüberprüfung des Modellserver während des Hochladens verhindert aber jederzeit geometrisch inkonsistente Zustände innerhalb der gemeinsamen Datenbank.

Wie in Abbildung 4.1 angedeutet, ist eine Kommunikation mit dem Modellserver nur über spezielle Zugriffsmethoden möglich, mittels derer die einzelnen Clients Daten und Informationen vom Server holen beziehungsweise wieder zurückübertragen können. Hierzu wird folgender Umfang an Methoden zur Verfügung gestellt:

**update():** Ein „Update“ liefert für alle noch nicht im lokalen Arbeitsbereich befindlichen Objekte eine Kopie des zentralen Modellbestandes mit **read**-Zugriffsrechten.

**check-out():** Vor einer Bearbeitung lokaler Objekte müssen diese „ausgecheckt“ werden. Dabei werden für die entsprechenden Objekte die gewünschten Zugriffsrechte im Modellserver vermerkt. Im Falle, dass diese zentral gesperrt sind (**exclusive-read**) beziehungsweise der Benutzer eine Sperrung wünscht, obwohl andere Benutzer einen schreibenden Zugriff (**read-write**) haben, wird die Anfrage zurückgewiesen.

**check-in():** Nach einer Bearbeitung können lokale Objekte mittels eines „Check-In“ wieder dem zentralen Modellbestand hinzugefügt bzw. dort ersetzt werden. Dies erfolgt aber erst nach erfolgreicher geometrischer Konsistenzprüfung. Es ist dabei durchaus möglich, dass bei einem „Check-In“ eine andere Version eines weiteren Benutzers wieder überschrieben

wird. Um einen reibungslosen Arbeitsablauf zu gewährleisten, wird dieses Verhalten zugelassen, muss aber durch geeignete Benachrichtigungsdienste unterstützt werden (vgl. hierzu unten).

`delete()`: Mittels eines „Delete“ können vom Benutzer selektierte Bauteile aus der zentralen Datenbasis des Modellservers gelöscht werden.

`query()`: Diese Methode erlaubt dem Benutzer für eine Auswahl von Objekten weiterführende Statusinformationen anzeigen zu lassen, wie bereits bestehende Zugriffe, Zeitstempel über die letzte Modifikation oder Informationen über die aktuell an einem Objekt arbeitenden Personen.

Im Rahmen des eingangs erwähnten Forschungsprojekts wurde vom Autor der vorliegenden Arbeit Client-seitig eine grafische Anwendung an das zentrale Framework angebunden und die obengenannten Funktionen zur Kommunikation mit dem Modellservers implementiert [86]. Im Folgenden soll hierzu exemplarisch ein Arbeitsvorgang aus der Sicht eines Clients beschrieben werden. Ein Mitarbeiter eines Projekts, der in der Datenbasis als Benutzer registriert ist, möchte Änderungen am gemeinsamen Modell vornehmen. Dazu wird er sich mit dem Server verbinden und zunächst mittels der bereitgestellten Funktion `update()` eine Kopie des aktuellen Standes in den lokalen Arbeitsbereich seiner Anwendung holen. Es werden dabei nur die Objekte kopiert, die noch nicht lokal vorhanden sind. Will der Benutzer Objekte ersetzen, muss er diese in seinem Arbeitsbereich zunächst löschen. In dem Zusammenhang bietet es sich an, die Informationen des Oktalbaums innerhalb der zentralen Datenbank zu nutzen, um beispielsweise bei Aufruf der „Update“-Funktion eine Auswahl von Objekten anhand raumbezogener („alle Objekte von  $z = 0.0\text{m}$  bis  $z = 5\text{m}$ “) oder raumbezogener und semantischer Kriterien („alle Objekte mit dem Attribut 'tragend' innerhalb eines vorgegebenen Bereichs“) vornehmen zu können. Eine derartige Funktionalität wurde in der derzeitigen prototypischen Implementierung allerdings noch nicht umgesetzt.

Will der Benutzer nun Teile seines Modells bearbeiten, wird dieser die entsprechenden Objekte selektieren und mittels eines „Check-Out“ eine Schreibberechtigung anfordern. Der Benutzer kann dabei entscheiden, ob er die Auswahl auf dem Server sperren (`exclusive-write`) oder aber weitere Schreibzugriffe anderer Benutzer (`read-write`) erlauben möchte. Abbildung 4.2 zeigt als Beispiel das Arbeitsfenster der Client-seitigen Anwendung in welcher ein Benutzer einen „Check-Out“ von Bauteilen vornimmt. Solange keine anderen Zugriffsrechte mit seiner Anfrage verletzt werden, wird der „Check-Out“ erlaubt, und für jedes korrespondierende Objekt auf dem zentralen Repository der Benutzername zusammen mit der Art der Berechtigung eingetragen. Wird die Anfrage zurückgewiesen, werden die entsprechenden Objekte in der Anwendung farblich hervorgehoben und der Benutzer kann dann mittels der Funktion `query()` die genauen Ursachen erfragen.

Nachdem das Modell in seinem lokalen Arbeitsbereich bearbeitet wurde, kann der Benutzer nun mittels eines „Check-In“ die Objekte dem zentralen Repository hinzufügen, wo diese entweder eine alte Version überschreiben oder, falls das Objekt neu erzeugt wurde, eine neue Instanz zentral angelegt wird. Zunächst wird auf dem Modellservers in jedem Fall aber eine geometrische Konsistenzüberprüfung durchgeführt. Ist diese konfliktfrei, wird der „Check-In“ zugelassen. Abbildung 4.3 zeigt hierzu den „Check-In“ einer Stütze im lokalen Arbeitsbereich eines Clients. Die Stütze wurde zuvor in ihrer Position verschoben.

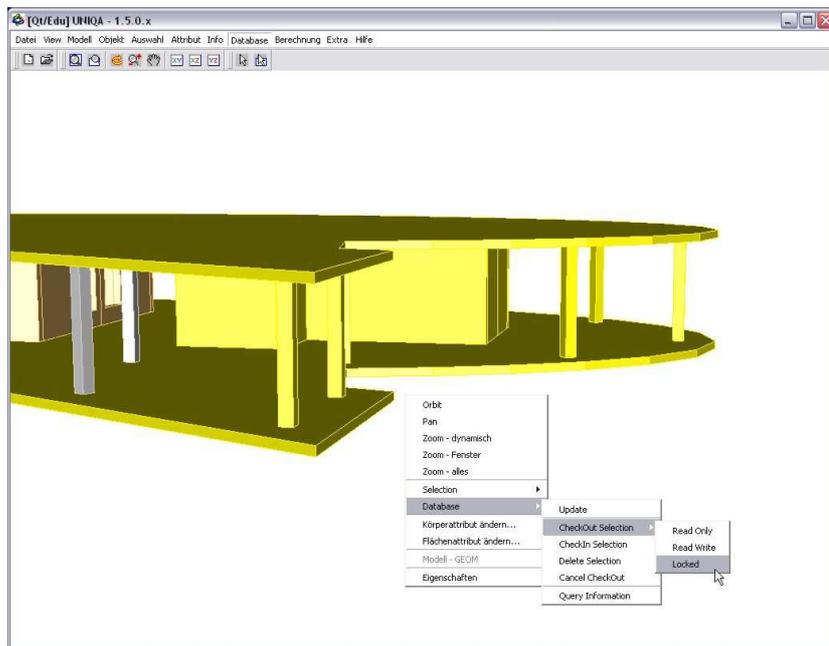


Abbildung 4.2: „Check-Out“ einer Auswahl von Bauteilen mit der Vorgabe, diese auf dem Modellservers exklusiv zu sperren.

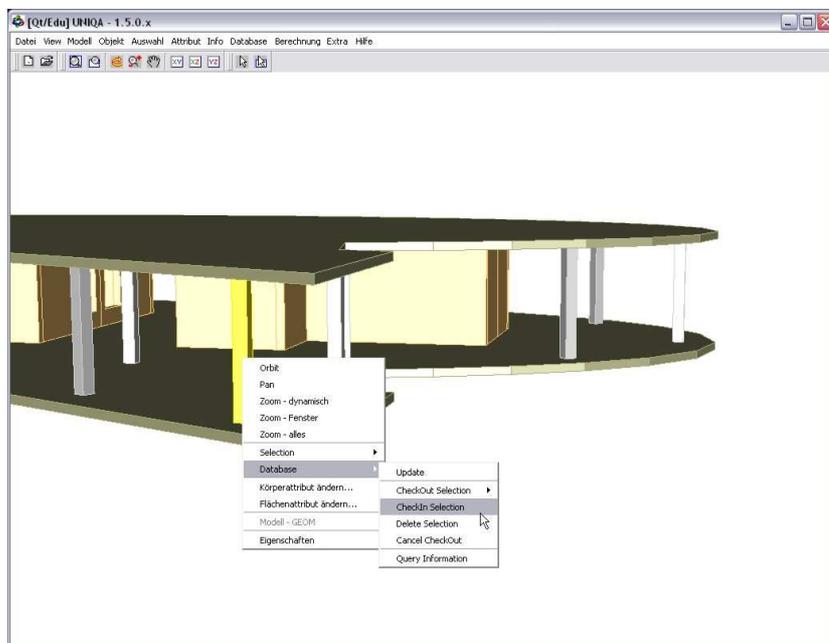


Abbildung 4.3: „Check-In“ einer Stütze (gelb markiert), nachdem diese etwas nach links verschoben wurde.

Beim „Check-In“ eines Objekts kann es nun allerdings passieren, dass eine geänderte Version eines anderen Benutzers dadurch wieder überschrieben wird. Dieses Verhalten ist aber durchaus so beabsichtigt. Alternativ müsste jeder „Check-In“ nach der ersten Modifikation eines anderen Benutzers zurückgewiesen werden. Dies würde aber bedeuten, dass nach jeder Änderung einer zentralen Instanz alle lokal existierenden Kopien als ungültig erklärt werden müssen, so dass die jeweiligen Benutzer immer gezwungen sind, sich stets die aktuellen Versionen vom Modellserver herunterzuladen und die bereits vorgenommenen eigenen Änderungen wieder daran anzupassen. Das Verhalten würde damit genau dem einer exklusiven Sperrung entsprechen. Ein derartiges Vorgehen würde zudem erfordern, dass alle Planungsbeteiligten permanent mit dem Modellserver verbunden bleiben müssen, um stets den eigenen Arbeitsbereich mit dem zentralen Datensatz konsistent halten zu können. Eine Arbeit im *offline*-Modus wäre damit nicht möglich. Andererseits können mit einem solchen Verhalten Konflikte nicht ausgeschlossen werden. Es sind daher zusätzliche Hilfsmittel bereitzustellen, mit denen Probleme frühzeitig erkannt und beseitigt werden können.

Um in dem Zusammenhang über die Aktivitäten anderer Teilnehmer informiert bleiben zu können werden im Rahmen des Frameworks Benachrichtigungsdienste angeboten, die clientseitig vom Benutzer gestartet werden, um diesen in bestimmten Zeitabständen über Bauteiländerungen am zentralen Modell zu informieren. Die Zeitabstände sowie die Art der Nachrichten können dabei vom Benutzer vorgegeben werden. Es kann ein Nachrichtenfilter erstellt werden, der auf eine Auswahl von Objekten begrenzt werden kann, die der Benutzer innerhalb seines Arbeitsbereiches selektiert. Darin lassen sich verschiedene Nachrichtenklassen definieren, mit denen Meldungen über Modifikationen an der geometrischen Form oder Änderungen an Attributen und Berechnungsparametern bestimmter Simulationsverfahren herausgefiltert werden können. Abbildung 4.4 zeigt auf der linken Seite das entsprechende Fenster mit allen Einstellungsmöglichkeiten. Der Benutzer startet also lokal den Dienst und nimmt eine Einstellung des Filters vor. Danach setzt sich der Nachrichtendienst in den Hintergrund, fragt in gewissen Zeitabständen einen Nachrichtenpuffer am Server ab und erwacht, wenn entsprechende Meldungen für den Benutzer vorliegen. In diesem Falle wird ein Fenster geöffnet und eine Liste der betroffenen Objekte angezeigt, wobei sich diese auch innerhalb des Modells farblich hervorheben lassen (siehe Abbildung 4.4 rechts). Um weiterhin detaillierte Informationen über die modifizierten Objekte zu erhalten, kann der Benutzer mittels der Methode `query()` die aktuellen Zugriffe, der Zeitpunkt der letzten Modifikation oder die Kontaktdaten des beteiligten Mitarbeiters abfragen.

### 4.1.3 Einbettung von Simulationsaufgaben

In der Praxis läuft kein Planungsprozess strikt linear ab. Jede Planung ist dadurch gekennzeichnet, dass Veränderungen zu berücksichtigen sind, dass bereits gefundene Lösungen verworfen und neue eingebunden werden müssen. Maßgebend für eine effiziente Planungsarbeit sind dabei Softwaresysteme, die eine schnelle Reaktion auf Änderungen erlauben sowie eine Integration verschiedener Fachanwendungen ermöglichen, um besonders die interdisziplinären Auswirkungen von Planungsentscheidungen früh genug abschätzen zu können. Neben der Bereitstellung adäquater Modelldarstellungen sind daher auch Softwarekonzepte notwendig, die eine optimale Nutzung der zugrunde liegenden Modelle ermöglichen. Zur Unterstützung verteilter Simulationsaufgaben auf Basis eines gemeinsamen digitalen Modells schlagen MUNDANI und BUNGARTZ et al. [78, 79, 80] hierfür Konzepte der *Einbettung* vor. Grundsätzlich kann

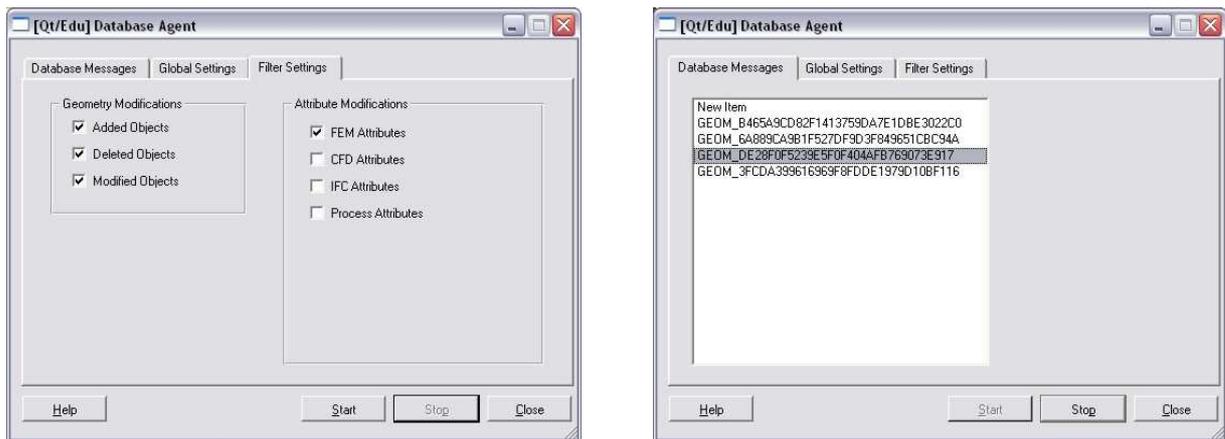


Abbildung 4.4: Dialogfenster des Benachrichtigungsdienstes. Links: Ansicht zur Konfiguration des Nachrichtenfilters. Rechts: Mitteilung über Änderungen von Objekten von Seiten anderer Benutzer.

eine Kopplung von Arbeitsprozessen entweder als Integration oder durch Einbettung erfolgen. Integration bezeichnet dabei die Kopplung von Arbeitsprozessen auf Basis eines gemeinsamen Datenmodells. Es lassen sich Redundanzen vermeiden, die Anzahl an Schnittstellen verringern und es ist durch die zentrale Datenhaltung eine Sicherung der Konsistenz möglich. Einbettung geht hier allerdings noch einen Schritt weiter. Durch Einbettung wird eine Applikation selbst zum Teil einer zentralen Komponente, welche damit neben Daten auch Funktionalität nach außen hin bereitstellen kann. Funktionalität, die von anderen Prozessen über vorbezeichnete (Funktions-)Schnittstellen wiederum genutzt werden kann. Eine Applikation nimmt damit nicht nur Dienste einer zentralen Datenbasis in Anspruch, sondern wird selbst zu einem Teil der Dienstsicht. Durch Einbettung von Prozessen kann ein wesentlich besserer Daten- und Informationsfluss zwischen Anwendung und zentraler Datenhaltung realisiert werden. Als Nachteil wird im Allgemeinen angesehen, dass auf diese Weise schwer wartbarer Code entstehen kann und die Austauschbarkeit der eingebetteten Applikationen nicht mehr ohne weiteres gegeben ist.

An Hand der Struktursimulation zeigen MUNDANI et al. in ihren Arbeiten [79, 81, 84] nun exemplarisch Vorteile, die sich aus einer derartigen Einbettung einer Simulationsaufgabe ergeben können. Durch Anwendung der rekursiven Substrukturierung lässt sich die hierarchische Organisation des zentralen Modells innerhalb des in Abbildung 4.1 dargestellten Frameworks sowohl zur Herstellung von räumlichen Informationen als auch zur Lösung des zugrunde liegenden Finite-Elemente-Gleichungssystems nutzen. Insbesondere bei lokalen Modifikationen am Modell bietet nun die Hierarchie der rekursiven Substrukturierung Vorteile. Durch Speicherung früherer Berechnungsergebnisse an den Knoten des Baumes lässt sich bei Neuberechnungen der Aufwand weitgehend auf den modifizierten Bereich reduzieren. Wie unter anderem die Untersuchungen in dieser Arbeit zeigen (vgl. Abschnitt 3.4.4), sind damit Antwortzeiten möglich, die auch für realistische Modelle durchaus eine interaktive Arbeitsweise erlauben.

## 4.2 Simulation von Bauphasen

In Kapitel 3 dieser Arbeit wurden Konzepte zur hierarchischen Organisation von Berechnungsmodellen vorgestellt. Es wurde gezeigt, wie auf Basis der rekursiven Substrukturierung ein sehr schnelles Lösungsverfahren implementiert werden kann, welches gerade für die in dieser Arbeit betrachteten Probleme der  $p$ -Version der Finite-Elemente-Methode auch bei komplexen Modellen anderen iterativen und direkten Gleichungslösern ebenbürtig ist. Der hierarchische Löser bietet aber darüber hinaus den Vorteil, während einer Neuberechnung nach Modifikationen am Modell den Aufwand weitgehend auf die modifizierten Bereiche beschränken zu können, wodurch teilweise erhebliche Zeiteinsparungen möglich sind. Dieser Vorteil lässt sich nun in idealer Weise zur Simulation von Bauphasen nutzen. Gemäß dem in Abschnitt 3.5.1 beschriebenen Vorgehen wird das Modell zunächst als Ganzes berechnet und die Separatorsteifigkeiten auf den Ebenen der Hierarchie zwischengespeichert. Wenn nun eine Berechnung für einen speziellen Bauzustand erfolgen soll, können Teile des Systems „ausgeblendet“ und der Lösungsvorgang nur noch auf die sichtbaren Bereiche angewendet werden.

Um jedoch das volle Potential des Verfahrens ausnützen zu können, sind neben den rein numerischen Gesichtspunkten zusätzlich Hilfsmittel bereitzustellen, die eine einfache Interaktion und Steuerung des Modells während der Simulation des Baufortschritts erlauben. Zur zeitlichen Organisation einer Baumaßnahme werden in der Praxis hierzu Softwaresysteme des Prozessmanagements eingesetzt. Neben der Modellierung und Simulation von zeitlichen Zusammenhängen kann mit derartigen Systemen zusätzlich der Einsatz von Produktionsmitteln wie Material, Geräte oder Personen und die damit zusammenhängenden Kosten geplant werden. Es fehlt jedoch der gerade bei Bauwerken wichtige räumliche Kontext. Es fehlt der Zusammenhang zwischen zeitlichen Informationen und räumlichem Modell und – was im Blickpunkt dieser Arbeit liegt – die Verbindung zum Tragwerksmodell und der Finite-Elemente-Simulation. Änderungen am Prozessplan können nur indirekt auf manuellem Wege auf das Tragwerksmodell abgebildet werden. Deshalb soll an dieser Stelle ein Vorschlag gemacht werden, in dem auf Grundlage des im vorherigen Abschnitt beschriebenen Frameworks durch Anbindung einer Projektplanungssoftware die direkte Verbindung von Bauablauf zum Tragwerksmodell realisiert werden kann. Durch Hinzunahme von zeitlichen Informationen zu den räumlichen des Tragsystems können so – im Sinne einer 4D-Modellierung – unterschiedliche Sichten der Bauplanung in einem Modell bedient und damit die direkte Zusammenarbeit von Fachdisziplinen unterstützt werden.

### 4.2.1 Softwarerahmen

Abbildung 4.2.1 zeigt in einer schematischen Darstellung die zur Simulation von Bauphasen notwendigen Komponenten. Innerhalb eines Projektmanagement-Systems werden zunächst die zeitlichen Abläufe mit allen Zusammenhängen modelliert und die Projektdaten in einer zentralen Datenbank abgelegt. Mithilfe einer grafischen Anwendung werden Vorgänge des Projektplans den Bauteilen beziehungsweise den entsprechenden finiten Elementen zugeordnet. Die entsprechende Relation „Vorgang – Element“ wird wiederum innerhalb der Datenbank gespeichert. Mithilfe dieser Verbindung kann aus den zeitlichen Daten des Prozessplans nun der hierarchische Löser gemäß dem Vorgehen in Abschnitt 3.5.1 entsprechend gesteuert und eine Berechnung zu beliebigen Bauzeitpunkten durchgeführt werden. Wichtig ist dabei eine enge Kopplung der hierarchischen Substrukturinformationen und der zeitlichen Daten inner-

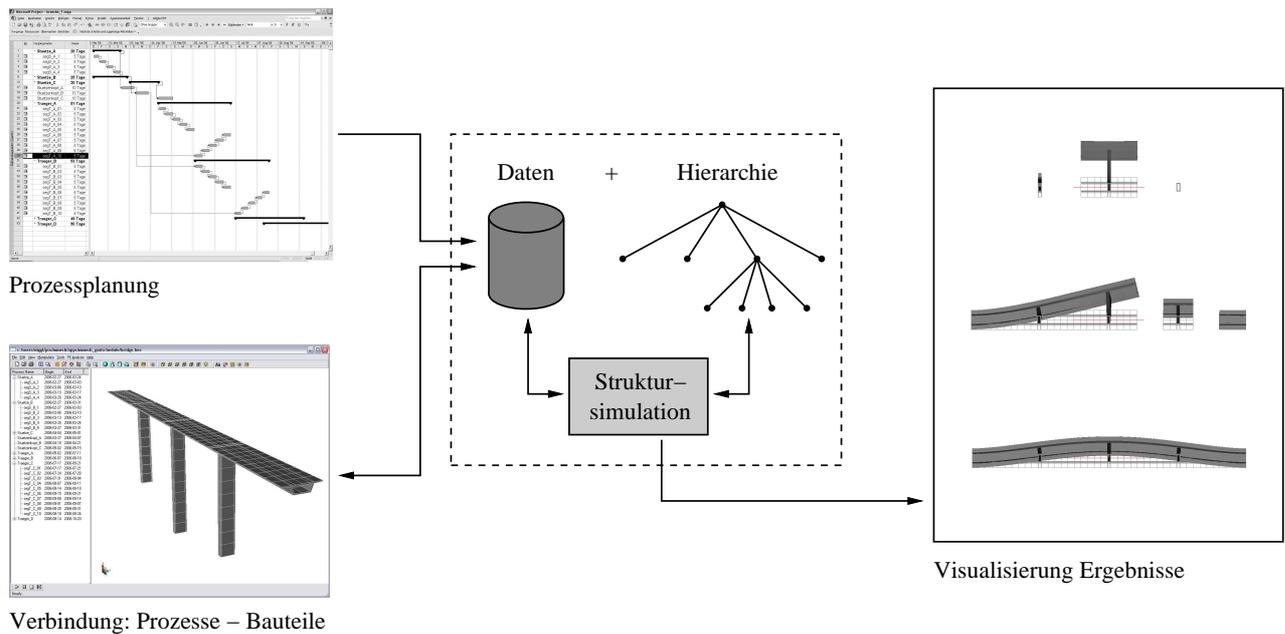


Abbildung 4.5: Darstellung aller Komponenten zur Bauphasensimulation: Definition eines Prozessplanes (MS-Project), Verbindung von Prozessen mit finiten Elementen in einer grafischen Anwendung, Berechnung und Visualisierung der Ergebnisse.

halb der Finite-Elemente-Simulation, welche in dieser Form nur möglich ist, wenn eine enge Verbindung zur Datenbasis, also eine Einbettung der Simulationsanwendung gegeben ist.

#### 4.2.1.1 Modellierung von Bauabläufen

Zur Modellierung der Bauabläufe kommt das Projektmanagement-System Microsoft Project<sup>2</sup> 2003 zum Einsatz. Microsoft Project ist eine Anwendung zum Planen, Steuern und Überwachen von Projekten. Es ist Bestandteil der Microsoft Office-Familie und lässt sich sowohl server- als auch clientseitig gut mit Software von Drittanbietern integrieren. Als zentrales Gestaltungselement zur Modellierung von Projekten dienen in Microsoft Project so genannte *Vorgänge*. Ein Vorgang kann dabei eine einzelne Aufgabe aus einem Arbeitspaket, eine Zusammenfassung von Aufgaben oder gar ein ganzes Teilprojekt abbilden. Vorgänge lassen sich hierarchisch gliedern und haben gewöhnlich eine zeitliche Ausdehnung. Sie können auch ohne Zeitdauer angelegt werden und definieren dann so genannte *Meilensteine*, festgelegte Termine, zu denen definierte, wichtige Teilziele erreicht werden müssen. Durch Definition von Anordnungsbeziehungen zwischen den Vorgängen lässt sich ein Netzplan definieren, welcher zur Berechnung von Projektdauer und Zeitreserven oder zur Analyse unerwarteter Zeitverzögerungen herangezogen werden kann. Daneben können den Vorgängen auch Ressourcen zugeordnet werden, wodurch sich die Möglichkeit bietet, die Verteilung von Mitarbeitern, Material und Kosten im Verlauf eines Projekts optimal verteilen und organisieren zu können. Schließlich können auch laufende Projekte überwacht werden, indem innerhalb der Vorgänge Fortschrittsgrade und verbrauchte Zeiten festgelegt werden. Dies erlaubt einem Projektmanager auch, laufende Aktivitäten anschaulich darstellen und präsentieren zu können.

<sup>2</sup>Microsoft Office Project: <http://office.microsoft.com/project/>

Zum Abgleich der Bauablaufinformationen mit der zentralen Datenbank (vgl. Abbildung 4.2.1) wurde mittels einer VBA<sup>3</sup>-Anwendung eine Anbindung an eine relationale Datenbank implementiert. Dabei werden von jedem Vorgang die Nummer, der Vorgangsname, das Start-Datum, das Ende-Datum, die Dauer sowie, falls vorhanden, die Nummer eines übergeordneten Sammelvorganges in der zentralen Datenbank innerhalb einer Tabelle abgelegt.

#### 4.2.1.2 Verbindung: Vorgänge – Elemente

Um die Verbindungen zwischen Vorgängen und Elementen des Berechnungsmodells herzustellen, kommt eine grafisch interaktive Anwendung zum Einsatz, die in der Arbeit von ROMBERG [103] entworfen und dort als Preprocessor für numerische Simulationen verwendet wurde. Diese Applikation wurde im Rahmen der vorliegenden Arbeit zur Visualisierung und Berechnung von Bauphasen am Finite-Elemente-Modell weiterentwickelt. Zur Herstellung der Verbindungen kann innerhalb der Anwendung auf eine Zweifenster-Ansicht gewechselt werden, in der eine Vorgangsliste mit Anfangs- und Enddatum sowie eine grafische Darstellung des Modells angezeigt werden können. Abbildung 4.6 zeigt die Applikation mit der Zweifensteransicht. Die Vorgangsliste im linken Fenster entspricht dabei der Datenbanktabelle, welche im vorherigen Schritt mithilfe von Microsoft Project erzeugt wurde.

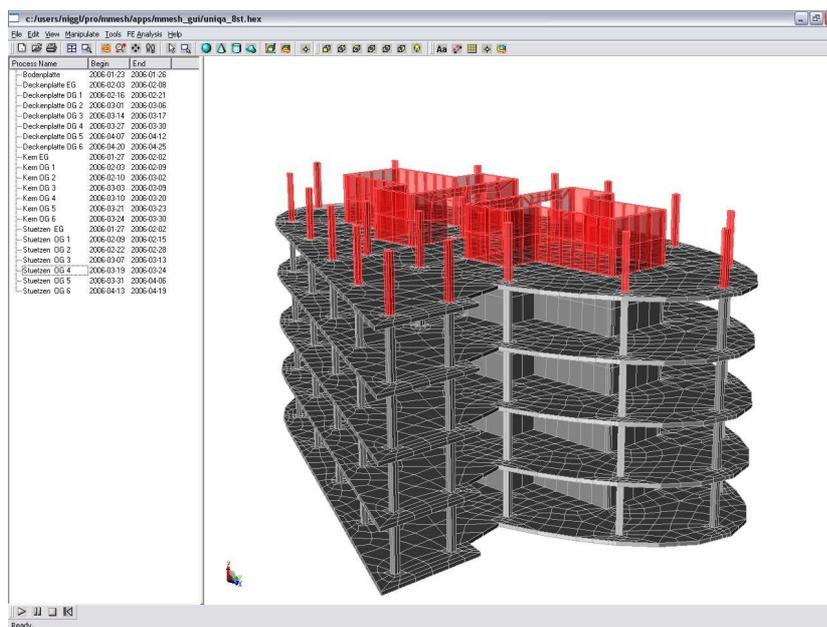


Abbildung 4.6: Grafische Anwendung zur Herstellung der Verbindung zwischen Bauvorgängen und finiten Elementen.

Um eine Verbindung herzustellen, können nun Elemente im dreidimensionalen Modell ausgewählt und einem Vorgang zugewiesen werden. Nach Aufbau aller Beziehungen lässt sich dann in beliebiger Weise das Modell zu bestimmten Bauzeitpunkten darstellen oder gesamte Bauablauf in einer Sequenz visualisieren.

<sup>3</sup>Visual Basic for Applications = Skriptsprache, welche unter anderem in den Microsoft Office Produkten zur Steuerung von Programmabläufen verwendet wird: <http://msdn.microsoft.com/vba/>

Abbildung 4.7 zeigt einen Ausschnitt des innerhalb der Datenbank befindlichen Objektmodells mit den entsprechenden Assoziationen. Für die prototypische Implementierung wurde dabei im Rahmen dieser Arbeit eine Verbindung zwischen Vorgängen und den finiten Elementen modelliert. Prinzipiell ist jedoch eine Verbindung zwischen Vorgängen und Bauteilen sinnvoll, welche in der Abbildung grau dargestellt ist, um auf einer allgemeineren Ebene weitere Bereiche der Planung berücksichtigen zu können. Ist eine Verbindung zwischen Bauteilen und finiten Elementen vorhanden – was während der Zerlegung eines Bauwerksmodells nach ROMBERG [103] (vgl. auch Abschnitt 2.1) so geschieht – lässt sich die entsprechende Verbindung Vorgang - Element leicht ableiten.

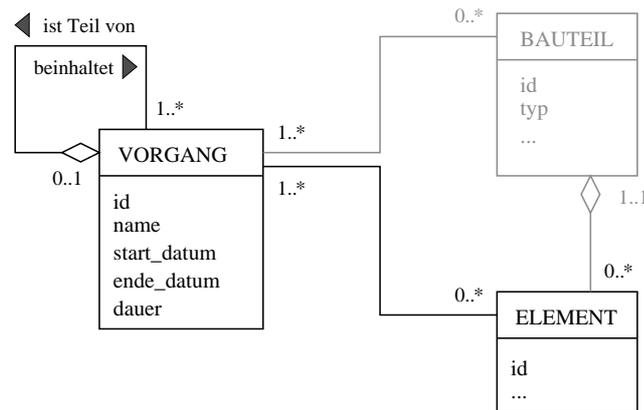


Abbildung 4.7: Beziehungen zwischen Vorgängen, Bauteilen und Elementen. Aktuell werden nur Verbindungen zwischen Vorgängen und Elementen modelliert.

Die Verbindung zwischen den Vorgängen des Prozessplanes und den finiten Elementen beziehungsweise den Bauteilen des geometrischen Modells weist im allgemeinen Falle eine Kardinalität vom Typ  $n:m$  auf. Das heisst, einem Vorgang können einerseits mehrere Elemente zugewiesen werden, es kann andererseits aber ein einzelnes Bauteil oder Element mit verschiedenen Bauablaufvorgängen verbunden sein (z.B. mit den Vorgängen *schalen*, *bewehren*, *betonieren*). Um Assoziationen vom Typ  $n:m$  auf eine relationale Datenbank abbilden zu können, muss diese als eigenständiges Objekt modelliert und in einer separaten Tabelle aufgelöst werden, von der  $1:n$  bzw.  $1:m$  Relationen ausgehen [48]. Innerhalb der in Abbildung 4.2.1 dargestellten Datenbank sowie im nachfolgenden Beispiel wird diese Relationstabelle mit dem Namen `ProcessesElements` bezeichnet.

## 4.2.2 Beispiel: Brücke im Freivorbau

Gerade bei Brückenbauwerken sind in der Tragwerksanalyse eine Vielzahl unterschiedlicher Bauzustände und Lastfälle zu berücksichtigen. Üblicherweise werden hierzu die einzelnen Bauzustände im Vorhinein definiert und mehr oder minder sequentiell durchgerechnet. Bei Änderungen am Modell oder am Zeitplan müssen unter Umständen neue Berechnungsmodelle erstellt und für diese wiederum Analysen durchgeführt werden. Eine enge Kopplung mit den zeitlichen Modellen der Projektplanung ist dabei in der Regel nicht gegeben. Zeitliche Änderungen am Projektlauf können nicht unmittelbar hinsichtlich ihren Auswirkungen auf das Tragwerksverhalten untersucht werden.

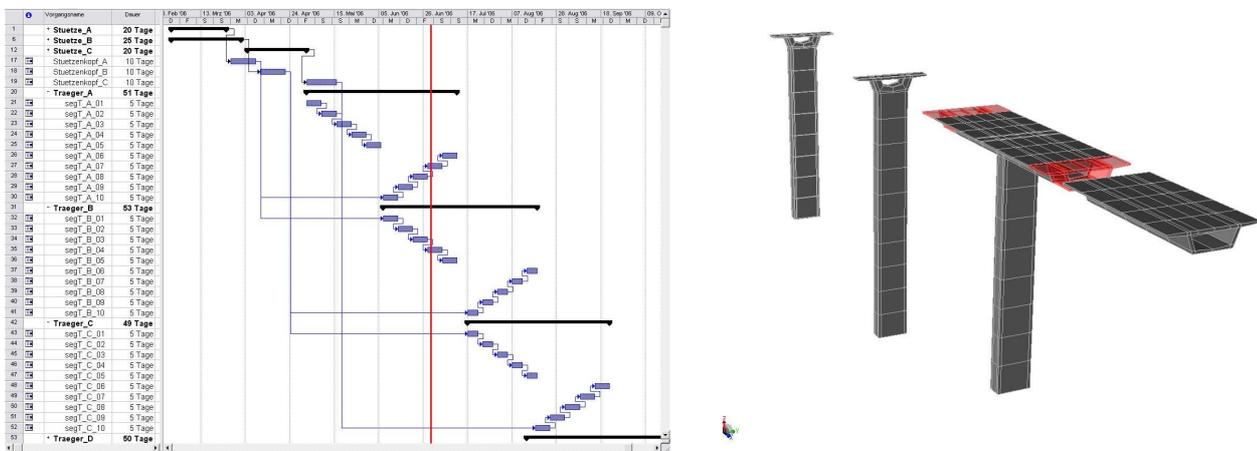


Abbildung 4.8: Darstellung des Bauablaufes eines Brückenbauwerkes im Balkendiagramm (links) zusammen mit einer Visualisierung des Berechnungsmodells zu einem bestimmten Bauzeitpunkt (rechts). Der Bauzeitpunkt ist im Balkendiagramm als senkrechte, rote Linie gekennzeichnet.

Das folgende Beispiel soll nun die Arbeit im Zusammenspiel zwischen Projekt- und Tragwerksplanung verdeutlichen. Es soll gezeigt werden, wie auf Grundlage der in Kapitel 3 vorgestellten hierarchischen Konzepte und der Einbindung eines Projektmanagement-Systems Änderungen am Projektplan sofort und direkt hinsichtlich ihrer Auswirkungen auf das Tragverhalten einer Struktur untersucht werden können.

Beispielhaft soll hierzu die Errichtung einer Brücke im Freivorbauverfahren betrachtet werden. Das exemplarisch modellierte Brückenbauwerk besteht aus einem durchlaufenden Brückenträger, der monolithisch verbunden auf drei Pfeilern lagert. Es wurde hierzu ein Finite-Elemente-Modell erzeugt, welches aus 918 Hexaederelementen besteht. Parallel dazu wurde mithilfe von Microsoft Project 2003 ein Projektplan aufgestellt, der in vereinfachter Weise den Bauablauf der Brücke modelliert. Dabei wird angenommen, dass nach Bau der drei Pfeiler die Fahrbahnträger von den Pfeilerköpfen ausgehend im Freivorbau zu beiden Seiten errichtet werden sollen. Die finiten Elemente aller Pfeiler und des Trägers wurden hierzu in Segmente gruppiert, wobei zum Bau jedes Segments fünf Tage veranschlagt wurden. Bei Modellierung des Netzplanes wurde als weitere Randbedingung angenommen, dass maximal zwei Arbeitskolonnen auf der Baustelle beschäftigt sein sollen. Weitere Gewerke im Bauablauf, wie etwa Gründungsmaßnahmen oder die Errichtung technischer Hilfsmittel wurden nicht berücksich-

sichtigt. Abbildung 4.8 zeigt auf der linken Seite einen Ausschnitt des in Microsoft Project modellierten Bauablaufes in einer Darstellung als Balkenplan. Auf der rechten Seite der Abbildung ist das Finite-Elemente-Netz zu einem bestimmten Bauzeitpunkt dargestellt. Dieser ist innerhalb des Projektplans als senkrechte, rote Linie gekennzeichnet.

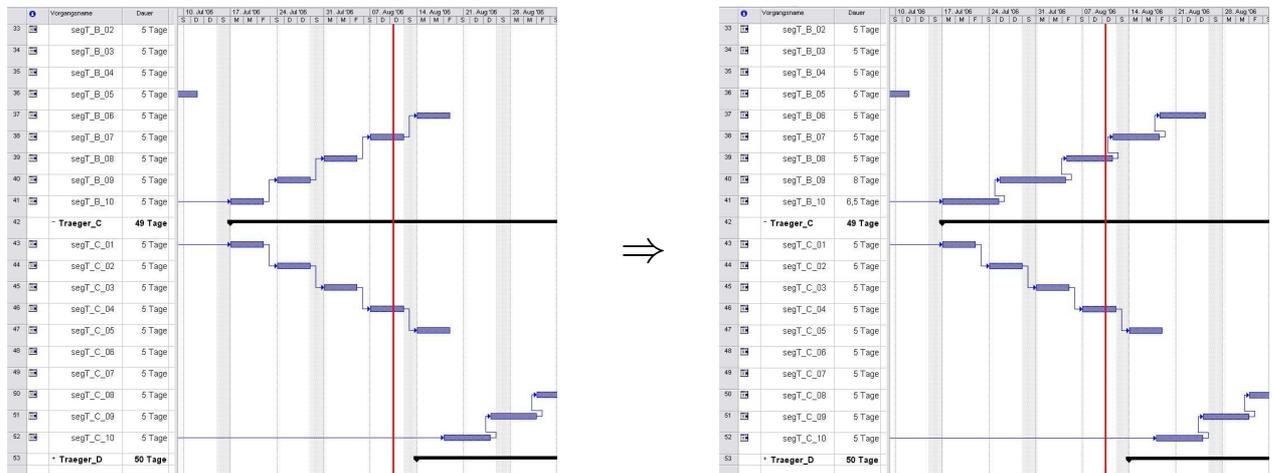


Abbildung 4.9: Beispielsszenario: Zeitliche Verzögerung beim Bau zweier Vorbausegmente. Kann die daraus entstehende Asymmetrie von der Struktur getragen werden?

Es wird nun folgendes Szenario betrachtet. Beim beidseitigen Vorbau des Brückenträgers vom mittleren Pfeiler ausgehend ergeben sich auf einer Vorbauseite aufgrund technischer Gründe außerplanmäßige Verzögerungen. Prinzipiell muss in diesem Falle nun die parallel arbeitende Vorbaukolonne auch verzögert werden, um eine stark asymmetrische Belastung des Pfeilers ausschließen zu können. Kritisch ist hierbei der Zeitpunkt der größten Auskrugung zu beiden Seiten des Pfeilers. Es wird nun angenommen, dass in einem ersten Schritt das Bauwerk auf Basis des planmässigen Projektablaufes zum Zeitpunkt der größten Auskrugung (fiktives Datum im Projektplan: 10.08.06) berechnet wurde. Tritt nun die Vorbauverzögerung ein, wird vom verantwortlichen Planer innerhalb des Projektplans die zeitliche Verzögerung in den entsprechenden Vorgängen berücksichtigt (vgl. Abbildung 4.9) und in der zentralen Datenbank abgespeichert. Auf Grundlage dieser Modifikation kann nun innerhalb der Finite-Elemente-Simulation mithilfe einer entsprechenden SQL-Abfrage an die Datenbank,

```
SELECT ProcessesElements.element FROM Processes, ProcessesElements
WHERE Processes.end_date <= "2006-08-10"
AND Processes.id = ProcessesElements.process
```

durch Verknüpfung der beiden Tabellen `ProcessesElements` und `Processes` eine neue Liste aller „sichtbaren“ Elemente zum Zeitpunkt „10.08.06“ bestimmt und eine Berechnung gestartet werden. Dabei muss gemäß dem in Abschnitt 3.5.1 beschriebenen Vorgehen nur die „Sichtbarkeit“ der Blattknoten im Substrukturbaum neu gekennzeichnet und eine Assemblierung sowie Lösung des Gesamtgleichungssystems durchgeführt werden. Aufgrund der Tatsache, dass sich im Vergleich zu vorher nur wenige Elemente (von „sichtbar“ auf „unsichtbar“) geändert haben, lassen sich die neuen Berechnungsergebnisse wesentlich schneller bestimmen, als bei einer Berechnung des gesamten Bauwerks. Siehe hierzu die Untersuchungen in Abschnitt 3.5.1.1. Als Resultat ergibt sich eine deutliche Biegebelastung innerhalb des Pfeilers mit entsprechenden

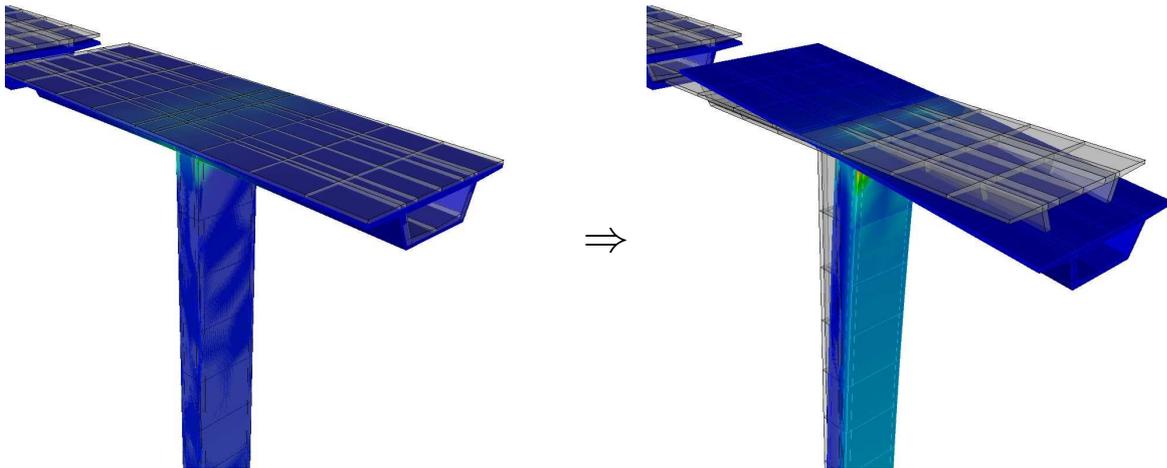


Abbildung 4.10: Auswirkungen der Vorbauverzögerung. Aufgrund der asymmetrischen Belastung tritt nun ein Biegemoment innerhalb des Pfeilers auf.

Verschiebungen des Pfeilerkopfes und der Trägersegmente wie in Abbildung 4.10 zu erkennen ist.

# Kapitel 5

## Volumenmodelle in der Bemessung von Stahlbetonbauteilen

Bemessung im Stahlbetonbau hat primär zur Aufgabe, die erforderlichen Querschnitte und Abmessungen von Beton und Stahl innerhalb eines Tragwerks zu bestimmen, um die Standsicherheit während der gesamten Nutzungsdauer eines Bauwerks sicherzustellen. Neben der reinen Bestimmung der Querschnittsgrößen ist darüber hinaus die *konstruktive Durchbildung* aller Bauteile ein weiterer wichtiger Bestandteil einer Tragwerksplanung. Hier werden die Ergebnisse der Bemessung konkret in Schal- und Bewehrungspläne umgesetzt. Dabei müssen besonders die Annahmen, die der statischen Berechnung und der Bemessung zugrunde liegen, mit einbezogen werden, um das tatsächliche Verhalten eines Tragwerks zu erfassen. Während der konstruktiven Durchbildung ist zudem immer das ganze Tragwerk zu betrachten und nicht nur die in der Bemessung erfassten Bereiche. Neben der Sicherstellung der Standsicherheit bei extremen Einwirkungen soll eine Bemessung und Konstruktion zudem die Gebrauchstauglichkeit sowie die Dauerhaftigkeit des Bauwerks gewährleisten. Weiterhin sind bei der konstruktiven Planung Fragen der Wirtschaftlichkeit sowie Aspekte der Fertigung zu beachten.

Der Baustoff Stahlbeton oder besser Konstruktionsbeton - dieser Begriff umfasst sowohl Stahlbeton als auch Spannbeton - ist ein klassischer Verbundbaustoff. Bei bewehrten Bauteilen werden Druckkräfte weitgehend über den Beton und Zugkräfte bei auftretender Rissbildung über den Stahl abgetragen. Aufgrund der hohen Festigkeit des Betons auf Druck sowie des Stahls auf Zug lassen sich damit sehr wirtschaftliche Tragstrukturen entwerfen. Weitere Vorteile des Baustoffes Stahlbeton liegen in der nahezu uneingeschränkten Formbarkeit sowie der Möglichkeit, Bauteile monolithisch miteinander verbinden zu können, wodurch sich mithilfe der damit vielfach statischen Unbestimmtheit hohe Tragreserven erzielen lassen [135].

Ziel dieser Arbeit ist, den Einsatz einer durchgängigen Modellierung und Berechnung mit Volumenstrukturen zu diskutieren. Ein derartiges Vorgehen steht im Gegensatz zur üblichen Praxis des Bauwesens. Dies liegt sicherlich zu einem guten Teil daran, dass die grundlegenden Annahmen der Nachweisverfahren auf Ergebnisse dimensionsreduzierter Berechnungen basieren. Im Folgenden sollen daher zunächst am Beispiel der Bemessung für Moment und Normalkraft sowie der Bemessung mittels Stabwerkmodellen zwei Nachweisverfahren und deren grundlegenden Annahmen aufgezeigt werden. Ein wichtiger Baustein des in dieser Arbeit vorgestellten Konzepts ist schließlich ein Verfahren zur Integration von Spannungsgrößen am Volumenmodell, welches im weitesten Sinne eine „Dimensionsreduktion“ der Berechnungsergebnisse erlaubt, um damit die notwendigen Eingangsgrößen für die Standardbemessungsver-

fahren zu erhalten. Hierzu soll im zweiten Teil dieses Kapitels ein entsprechendes Verfahren vorgestellt werden. Es soll dann im Weiteren an Beispielen aufgezeigt werden, wo sich bei einer dimensionsreduzierten Modellierung Schwierigkeiten ergeben und in welchen Fällen sich durch die Verwendung von Volumenstrukturen Vorteile erzielen lassen. In einem weiteren Abschnitt soll dann untersucht werden, in wie weit sich auf Basis der dreidimensionalen Ergebnisdaten die grundlegenden Annahmen der Bemessungsverfahren überprüfen lassen. Am Ende dieses Kapitels werden schließlich Hilfsmittel vorgestellt, die im Rahmen einer interaktiven Bemessung von Bauteilen eingesetzt werden können.

## 5.1 Bemessung im konstruktiven Betonbau

### 5.1.1 Aufgaben der Bemessung

Bemessung hat primär zum Ziel, ein Versagen des Tragwerks oder eines Bauteils durch planmäßige aber in begrenztem Umfang auch unplanmäßige Einwirkungen während der gesamten Lebensdauer eines Bauwerks sicher zu verhindern. Gleichzeitig sollen darüber hinaus Anforderungen, die während der Nutzung an das Bauwerk gestellt werden - etwa eine Begrenzung auftretender Verformungen - mit einer ausreichenden Zuverlässigkeit sichergestellt werden. Entsprechend diesen Forderungen werden gemäß DIN 1045-1, „Tragwerke aus Beton, Stahlbeton und Spannbeton, Teil 1: Bemessung und Konstruktion“ [27] Nachweise nach unterschiedlichen Grenzzuständen geführt: dem Grenzzustand der Tragfähigkeit (GZT), Grenzzustand der Gebrauchstauglichkeit (GZG) und der Dauerhaftigkeit. Die Nachweise werden dabei rechnerisch als auch durch Einhaltung konstruktiver Regeln erbracht. Gerade die Berücksichtigung konstruktiver Vorgaben spielt eine wichtige Rolle im Bemessungskonzept der DIN 1045-1. Sie sind wesentlich zur Sicherstellung der Dauerhaftigkeit sowie der Gebrauchstauglichkeit verantwortlich. Darüber hinaus stellen konstruktive Regeln sicher, dass die in den Rechenmodellen getroffenen Annahmen, wie etwa eine hinreichend plastische Verformbarkeit oder ein linearer Verlauf der Dehnungen, sichergestellt werden [135].

Die rechnerischen Nachweise im Grenzzustand der Tragfähigkeit basieren im Allgemeinen auf einer Gegenüberstellung von Bemessungswerten der Einwirkungen  $E_d$  (engl. **E**ffects of **a**ctions) - auch Beanspruchungen - mit einem Bemessungswert des Bauteilwiderstandes  $R_d$  (engl. **R**esistance):

$$E_d \leq R_d. \quad (5.1)$$

Für die Anwendung gilt hierbei das in DIN 1055-100 [28] festgelegte Sicherheitskonzept. Die Nachweisführung erfolgt in allen Grenzzuständen mit *Bemessungswerten* (Subscript „d“, engl. **d**esign), die die stochastischen Unsicherheiten von Einwirkungen, Baustoffeigenschaften oder Abmessungen berücksichtigen und unter Hinzunahme von Teilsicherheitsbeiwerten ein ausreichendes Sicherheitsniveau gewährleisten.

$E_d$  und  $R_d$  stellen dabei zunächst allgemeine Vergleichsgrößen dar. Wegen des nichtlinearen Verhaltens des Baustoffes Stahlbeton und den Schwierigkeiten einer direkten Modellierung der Baustoffe im Verbund können die Nachweise nicht - wie beispielsweise im Holzbau oder im Stahlbau - auf der Ebene der Spannungen an maximal belasteten Querschnittspunkten geführt werden. Die Sicherheitsnachweise werden vielmehr auf dem Niveau der Schnittgrößen am Querschnitt (z.B.  $M_{Ed} \leq M_{Rd}$ ) oder global auf Lastniveau ( $q_{Ed} \leq q_{Rd}$ ) erbracht [135].

### 5.1.2 Betrachtungsebene: Querschnitt

In der Bemessung von Stahlbetonbauteilen lassen sich zwei Herangehensweisen unterscheiden. ZILCH und ZEHETMAIER [135] sprechen von verschiedenen Betrachtungsebenen: Betrachtungen am Querschnitt sowie am System. Erstere bestimmt das Tragverhalten eines Bauteils anhand der Verhältnisse von Dehnungen und Spannungen an ausgezeichneten Querschnitten bzw. den daran wirkenden Kraftverhältnissen. Auf der anderen Seite lassen sich aber auch Betrachtungen am System bzw. an Teilsystemen einer Struktur anstellen, in denen das Tragverhalten auf Grundlage einer vereinfachten Vorstellung des Kraftflusses untersucht wird. Beide Verfahren besitzen hinsichtlich der geforderten Modellannahmen unterschiedliche Voraussetzungen und kommen für verschiedene Nachweiskonzepte zum Einsatz.

Im Folgenden soll als Beispiel für Betrachtungen am Querschnitt das Verfahren der Bemessung für Beanspruchungen aus Biegemomenten und Längskraft diskutiert werden. Die Bemessung erfolgt hier durch den Nachweis, dass der Querschnitt an einer kritischen Stelle im Tragwerk in der Lage ist, die einwirkenden Schnittgrößen  $M_{Ed}$  und  $N_{Ed}$  durch innere Kraftgrößen aufnehmen zu können:

$$\begin{pmatrix} M_{Ed} \\ N_{Ed} \end{pmatrix} \leq \begin{pmatrix} M_{Rd} \\ N_{Rd} \end{pmatrix}. \quad (5.2)$$

Die zu erwartenden Beanspruchungen für einen Nachweis im Grenzzustand der Gebrauchstauglichkeit (GZG) sind im Allgemeinen wesentlich geringer als im Zustand der Grenztragfähigkeit. Es kann daher näherungsweise ein linear-elastisches Verhalten der Baustoffe angenommen werden, wodurch sich die erforderlichen Querschnittswerte aus dem inneren Spannungs- und Dehnungszustand nach den Regeln der Technischen Biegelehre [115] unter Verwendung ideeller Querschnittsgrößen ermitteln lassen.

Im Grenzzustand der Tragfähigkeit wird dagegen angenommen, dass der Beton innerhalb der Zugzone vollständig gerissen ist. Damit können vom Beton keine Zugkräfte mehr aufgenommen werden. Diese konzentrieren sich innerhalb der Stahleinlage. Druckkräfte werden dagegen in erster Linie über den Beton abgetragen. Weiterhin gelten bei höheren Belastungen auch keine linearen Materialbeziehungen mehr. Der druckbeanspruchte Beton verhält sich materiell nichtlinear. Aufgrund dieser Gegebenheiten kann aus den äußeren Kraftgrößen nicht mehr unmittelbar auf die Spannungsverhältnisse im Inneren eines Querschnittes geschlossen und damit die Grenze der Tragfähigkeit bestimmt werden. Abbildung 5.1 zeigt hierzu das so idealisierte Verhalten eines Rechteckquerschnitts mit Stahleinlage im Grenzzustand der Tragfähigkeit. Darin herrscht am Querschnitt ein innerer Kräftezustand  $M_{Rd}$  und  $N_{Rd}$ , der sich aus dem Kräftepaar der Spannungsergebenden in Stahl und Beton ergibt und welcher mit den außen einwirkenden Schnittkräften im Gleichgewicht steht. Die Spannungsverteilungen selbst ergeben sich unter Berücksichtigung idealisierter, nichtlinearer Materialgesetze aus der vorgegebenen Dehnungsverteilung.

Die Aufgabe zur Bemessung eines Querschnittes lautet nun: Bei gegebenen Querschnittsabmessungen und Lage der Bewehrung muss ein Dehnungszustand so bestimmt werden, dass die inneren Kraftgrößen den äußeren Schnittgrößen  $M_{Ed}$  und  $N_{Ed}$  entsprechen und die Menge der Bewehrung minimal wird. Hierbei wird angenommen, dass die Dehnungsverteilung linear über den Querschnitt läuft, so dass aus den Dehnungen am Rande auf den Verlauf im Inneren geschlossen werden kann und dass weiterhin zwischen Spannungen und Dehnungen - vereinfachte - nichtlineare Materialgesetze gelten. Aufgrund der Nichtlinearität ist eine geschlossene

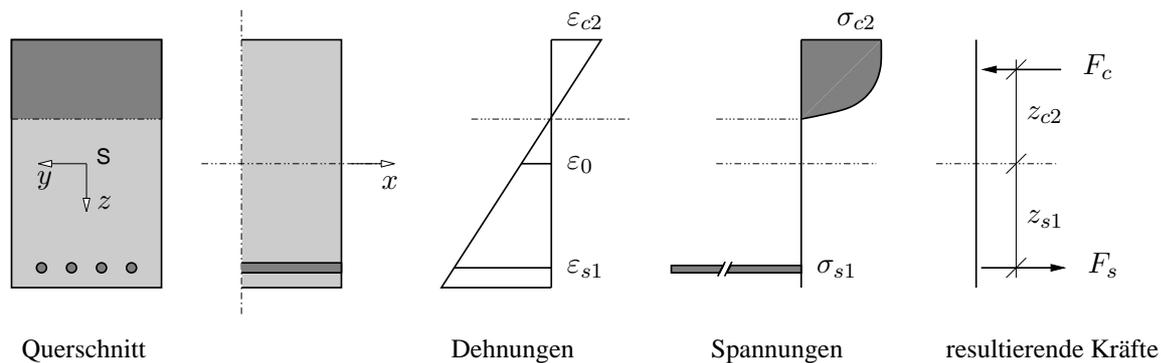


Abbildung 5.1: Dehnungen, Spannungen und innere Kräfte eines durch Biegung und Längskraft beanspruchten Balkens im Grenzzustand der Tragfähigkeit.

Lösung zur Bestimmung der erforderlichen Stahlquerschnitte nicht mehr möglich. Die erforderliche Bewehrungsmenge muss iterativ bestimmt werden, beispielsweise durch Anwendung des NEWTON-Verfahrens [134]. Zur Vereinfachung werden in den Normen hierzu Diagramme und Tabellen angeboten, in denen die erforderlichen Stahlquerschnitte direkt abgelesen werden können.

Die grundlegende Voraussetzung für das Standardverfahren der Bemessung von Balken und Platten unter Einwirkung von Normalkraft und Moment ist die Annahme einer linearen Verteilung der Dehnung über den Querschnitt. Zusammen mit der Vernachlässigung von Schubverformungen entspricht dies der Hypothese von BERNOULLI vom *Ebenbleiben der Querschnitte* in der Balkentheorie [115]. Nach den gängigen Regeln trifft diese für Balken ab einem Verhältnis von Stützweite zur Bauteilhöhe von  $l_{eff}/h \geq 2$  bzw. bei Kragträgern von  $l_{eff}/h \geq 1$  zu. Neben der BERNOULLI-Hypothese werden bei einer Bemessung biege- und normalkraftbeanspruchter Querschnitte im Grenzzustand der Tragfähigkeit noch weitere Annahmen getroffen. Zusammengefasst lauten diese [134]:

- Es gilt die BERNOULLI-Hypothese vom Ebenbleiben der Querschnitte. Es kann damit eine lineare Verteilung der Dehnungen über den Querschnitt angenommen werden.
- Die Bemessung erfolgt unter Annahme eines vollständig gerissenen Betons in der Zugzone. Alle Zugkräfte müssen durch die Bewehrung aufgenommen werden.
- Für das Zusammenwirken zwischen Bewehrung und Beton wird starrer Verbund vorausgesetzt.
- Für den Beton und Stahl gelten vereinfachte, idealisierte Spannungs-Dehnungsbeziehungen gemäß DIN 1045-1, Abs. 9.1.6 und 9.3.3 [27].

Die Einhaltung obiger Modellannahmen wird zum Teil durch konstruktive Regeln in den Normen sichergestellt. Beispielsweise kann die Hypothese vom Ebenbleiben der Querschnitte nur bei geringen Rissabständen im Beton gewährleistet werden. Zur Begrenzung der Rissbreiten ist innerhalb der DIN 1045-1 hierzu eine Mindestbewehrung vorgesehen.

Grundsätzlich unterliegt die Überprüfung dieser Annahmen und damit die Anwendbarkeit eines konkreten Bemessungsverfahrens der Entscheidung des Ingenieurs. Einfache Regeln wie

beispielsweise das Verhältnis von Länge zu Höhe ( $l_{eff}/h \geq 2$ ) sind allerdings nicht immer zweckmäßig. SCHLAICH und SCHÄFER [112] weisen auf diesen Umstand hin. In Bereichen mit konzentrierten Lasten oder bei geometrischen Diskontinuitäten gilt auch in schmalen Balken die BERNOULLI-Hypothese vom Ebenbleiben der Querschnitte nicht. Die Standardbemessungsverfahren sind in diesen Bereichen daher nicht mehr anwendbar. SCHLAICH und SCHÄFER schlagen vor, ein Tragwerk vor einer Bemessung konsequent in so genannte *B-Zonen* (B entsprechend Bernoulli, Balkentheorie) und *D-Zonen* (D entsprechend Diskontinuität, Detail) einzuteilen. Für die Bereiche mit nicht-ebenen Dehnungsverläufen der D-Zonen müssen dann Verfahren zur Anwendung kommen, die eine Betrachtung am gesamten System vorsehen und welche im nächsten Abschnitt kurz erläutert werden.

### 5.1.3 Betrachtungsebene: System

Eine grundsätzlich andere Möglichkeit, das Tragverhalten eines Bauteils zu beurteilen, liegt in der Betrachtung des gesamten Systems oder von Teilsystemen auf Grundlage einer vereinfachten Vorstellung des inneren Kraftflusses. Ausgehend von den angenommenen Kraftverläufen wird hierzu im Inneren eines Bauteils ein *Stabwerkmodell* bestehend aus Druck- und Zugstäben gebildet, welches mit den äußeren Kräften im Gleichgewicht steht. Abbildung 5.2 zeigt hierzu einen Balken mit den Elementen des Stabwerkmodells. Die Kräfte in den Druckstreben repräsentieren darin die Verhältnisse innerhalb der Betondruckzone, während die Lasten der Zugstreben vom Betonstahl übernommen werden. Voraussetzung ist bei letzterem natürlich, dass die Bewehrung nach Lage und Richtung mit den Zugstreben im Modell übereinstimmt.

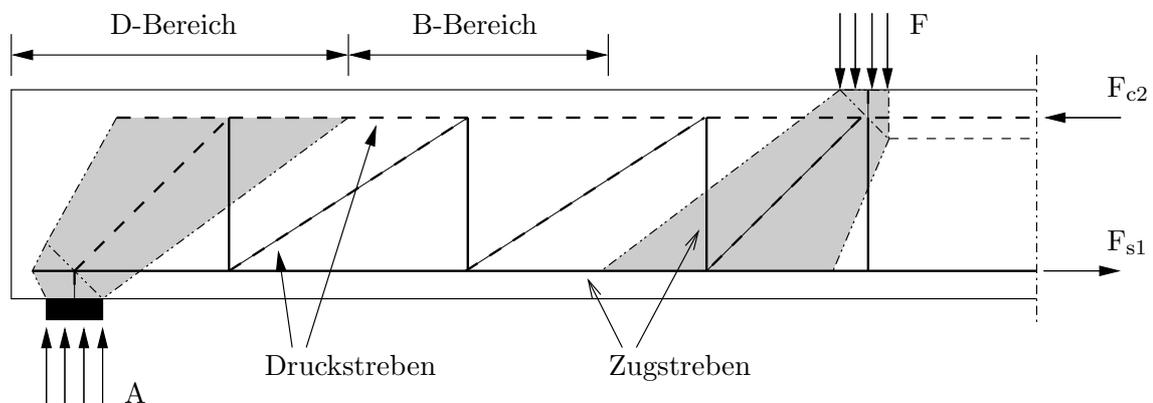


Abbildung 5.2: Endauflagerbereich eines Balkens mit B- und D-Zonen nach REINECK [102]. Die Abbildung zeigt die Elemente eines Stabwerkmodells sowie die trapezförmigen Druckfelder.

Um die Tragfähigkeit einer Struktur zu gewährleisten, muss nun der Nachweis erbracht werden, dass in allen Elementen des Stabwerkmodells die zulässigen Festigkeitswerte der Baustoffe eingehalten werden, wobei hier in erster Linie die Knotenbereiche zu überprüfen sind. Die Gültigkeit des Bemessungskonzepts wird durch den *1. Grenzwertsatz der Plastizitätstheorie* sichergestellt (vgl. [112]). Dieser besagt, dass ein Tragwerk aus plastisch verformbaren Werkstoffen nicht versagt, wenn zu einer gegebenen Belastung irgendeine Spannungsverteilung gefunden werden kann, die die Gleichgewichtsbedingungen erfüllt und nirgends die Fließgrenze des Werkstoffes überschreitet. Das Tragwerk passt sich durch plastische Umlagerungen

dem inneren statischen System an, welches durch das Stabwerkmodell vorgegeben ist. Voraussetzung ist jedoch, dass sich die Kraftverhältnisse einstellen können, bevor das plastische Verformungsvermögen („Rotationsfähigkeit“) der Strukturen lokal bereits überschritten ist. SCHLAICH und SCHÄFER [112] geben hierzu die Empfehlung, die Modelle an den Spannungsverteilungen der linearen Elastizitätstheorie zu orientieren, d.h. die Neigungen der Streben an den Verlauf der Hauptspannungstrajektorien anzupassen. Mit der Orientierung an der Spannungsverteilung der linearen Elastizitätstheorie wird jedoch auf die Möglichkeit wesentlicher Umlagerungen von inneren Kräften und damit auf die Ausnutzung zusätzlicher Tragreserven verzichtet. Dafür sind andererseits auch keine besonderen Nachweise zur Sicherstellung der Verformungsfähigkeit zu erbringen [102, 27]. Die Orientierung an der Elastizitätstheorie stellt zudem die kinematische Verträglichkeit des gewählten Modells sicher. Weiterhin kann das gewählte Stabwerkmodell auch für Nachweise der Gebrauchstauglichkeit herangezogen werden, da in diesen Fällen das Tragverhalten weitestgehend den Verhältnissen in der linearen Elastizitätstheorie entspricht.

In [111, 112] schlagen SCHLAICH und SCHÄFER Konzepte zur Anwendung von Stabwerkmodellen für allgemeine Bauteile vor. Sie empfehlen, diese zur Bemessung für die Bereiche eines Tragwerks heranzuziehen, in denen die Standardbemessungsverfahren nicht mehr angewendet werden können. Fachwerkmodelle wurden allerdings schon seit Beginn des „Eisenbetonbaus“ zur Beschreibung des Tragverhaltens von Stahlbetonbauteilen angewendet [76]. Sie stellen auch die Grundlage der normativen Bemessungsverfahren für Querkraft und Torsion dar. Durch die konsequente Einführung von Stabwerkmodellen für allgemeine Bauteile nach [111, 112] ist nun aber ein durchgängiges Bemessungskonzept gegeben, welches sich für *alle* Bereiche eines Tragwerks anwenden lässt. Nach dem bisher üblichen Vorgehen in der Praxis werden in der Regel nur die kritischen Querschnitte mithilfe der in Abschnitt 5.1.2 vorgestellten Standardverfahren nachgewiesen. Deren Anwendung ist jedoch nur in Bereichen mit ebenem Dehnungszustand gültig, in denen die Voraussetzungen der technischen Biegetheorie gelten - alle restlichen Bereiche eines Tragwerks werden bislang nur bei der konstruktiven Durchbildung berücksichtigt. Sie werden damit nicht in die eigentliche Bemessung mit einbezogen. Für die Sicherheit eines Bauwerkes sind aber gerade die Bereiche mit nichtlinearen Dehnungsverläufen maßgeblich verantwortlich [102]. Das Versagen eines Bauteils tritt meist in Zonen mit großen Lasteinleitungen oder geometrischen Änderungen auf.

Eine durchgängige Berücksichtigung aller Bereiche eines Tragwerks unter Verwendung von Stabwerkmodellen bietet also den Vorteil, dass die Abhängigkeit der Ergebnisse von den subjektiven Fähigkeiten und Erfahrungen des Ingenieurs verringert wird, wodurch die „Qualität der Tragwerke insgesamt gleichmäßiger und besser wird“ [112]. Die Methode der Stabwerkmodelle wurde daher inzwischen in verschiedene Normen Europas sowie Nordamerikas aufgenommen (z.B. DIN 1045-1 [27], Abs. 10.6). Es ist jedoch zu betonen, dass im Falle linearer Dehnungsverteilungen nach wie vor die Standardbemessungsverfahren für Normalkraft und Moment am Querschnitt zur Anwendung kommen sollten, da diese durch zahlreiche Versuche getestet und optimiert sind und daher meist wirtschaftlichere Lösungen liefern.

Zur Bemessung von Bauteilen geben SCHLAICH und SCHÄFER nun die Empfehlung, schon während der Modellierung ein Tragwerk konsequent in so genannte B- und D-Zonen einzuteilen. In den B-Bereichen werden die notwendigen Stahlmengen mithilfe der in Abschnitt 5.1.2 angesprochenen Standardbemessungsverfahren ermittelt. Für Bereiche mit nicht-ebenen Dehnungsverteilungen, in denen die Standardverfahren nicht mehr anwendbar sind, erfolgt die Bemessung dann mittels Stabwerkmodellen. Zur Bestimmung und Abgrenzung der D-Bereiche

ist in [112] eine entsprechende Vorgehensweise aufgeführt. Hierzu werden innerhalb eines Tragwerks zunächst Stellen mit geometrischen und statischen Diskontinuitäten identifiziert. Diese findet man in Bereichen mit sich plötzlich ändernder Tragwerksgeometrie (geometrische Diskontinuitäten) oder an Orten mit konzentrierten oder sich verändernden Lasteinleitungen (statische Diskontinuitäten). Beispiele hierfür sind Rahmenecken, Endauflager, Öffnungen, Konsolen oder Einzellasten. Zur Bestimmung der Ausdehnung der Diskontinuitätsbereiche ziehen die Autoren schließlich das Prinzip von DE SAINT-VENANT von der „elastischen Gleichwertigkeit statisch äquivalenter Lastgruppen“ (vgl. [123]) heran. Dieses Prinzip wird auf Bereiche mit geometrischen und statischen Diskontinuitäten am Balken angewandt und die Aussage abgeleitet, dass die Ausdehnung nach beiden Seiten näherungsweise durch die Höhe des Balkens begrenzt ist. Eine genaue quantitative Aussage über die Längenausdehnung kann jedoch nicht getroffen werden. Diese wird zudem vom Verlauf der Rissausbreitung im Beton sowie der angenommenen Neigung der Zug- und Druckstreben beeinflusst.

Sind die einzelnen Bereiche erfaßt, können innerhalb der D-Zonen schließlich Stabwerke zur Beschreibung des Lastabtrages modelliert werden. Hierzu können unterschiedliche Verfahren zur Anwendung kommen:

- Automatische Generierung von Stabwerken aus den Hauptspannungstrajektorien: Für ebene Spannungszustände wird beispielsweise von HÖRMANN [60] ein Konzept zur automatischen Generierung von Stabwerken vorgestellt.
- Freie Modellierung der Stäbe: Hierzu können beispielsweise in einem CAD-System die einzelnen Stäbe interaktiv innerhalb der Bauteile modelliert werden [106]. Die Berechnung der Stabkräfte sowie die Nachweise der Tragfähigkeiten laufen dann weitgehend automatisch ab.
- Verwendung standardisierter Modelle: Für Standardfälle (Auflager, Konsole, Rahmenecke) lassen sich standardisierte Stabwerkmodelle anfertigen, die dann - beispielsweise parametrisiert vorgehalten in einer Bibliothek [106] - für konkrete Bemessungsbeispiele angepasst werden.
- Anwendung der „Lastpfadmethode“: Hierzu werden an ausgewählten Schnitten Spannungsblöcke zu Resultierenden zusammengefasst, die dann die Größe und Richtung der Stabkräfte vorgeben [112].

Bei räumlichen Spannungszuständen werden die Tragwerke üblicherweise in mehreren zueinander senkrechten Ebenen getrennt untersucht. Die Zusammenhänge zwischen den Ebenen sind dabei aber immer zu berücksichtigen.

Durch Anwendung einer strikt volumenorientierten Modellierung in der Tragwerksberechnung bietet sich nun allerdings die Möglichkeit, Bereiche mit nichtlinearen Dehnungsverteilungen automatisch erkennen zu lassen, um so die Gültigkeit der Voraussetzungen der Bemessungsverfahren zu überprüfen. In Bereichen mit ebenen Dehnungsverteilungen können durch Integration der dreidimensionalen Spannungen über den Querschnitt nach wie vor die Schnittgrößen entlang einer Balkenachse bestimmt werden, worauf die Standardbemessungsverfahren für Moment und Normalkraft zur Anwendung kommen können. Für die übrigen Bereiche mit diskontinuierlichen Spannungsverläufen liegt dann aber jederzeit der dreidimensionale Spannungszustand vor und es lassen sich beispielsweise die Hauptspannungsrichtungen darstellen,

die als Grundlage zur Modellfindung eines Stabwerkes nach einem der oben genannten Verfahren dienen können.

## 5.2 Volumenmodellierung als Grundlage

Grundsätzlich ist festzustellen, dass bei einer strikt volumenorientierten Modellierung aller Bauteile die dimensionsreduzierten Größen jederzeit in dem höherwertigen Modell enthalten sind. Es ist kein Problem, beispielsweise die für die Standardbemessungsverfahren notwendigen Querschnittswerte durch Integration aus den dreidimensionalen Spannungen zu bestimmen. Allerdings mit einem Unterschied: Die Reduktion findet unter Berücksichtigung der tatsächlichen Abmessungen der Bauteile statt, die normalerweise nur implizit in Betracht gezogen werden und innerhalb der Normen in der Regel nur durch Konstruktionsvorgaben berücksichtigt werden.

Die Reduktion der dreidimensionalen Spannungsgrößen auf Querschnittswerte ist hierbei prinzipiell ein Postprocessing-Schritt, der nach einer Finite-Elemente-Berechnung mit entsprechenden Algorithmen in Echtzeit und interaktiv stattfinden kann. Aufgrund der Tatsache, dass die Dimensionsreduktion dabei nicht mehr *vor*, sondern *nach* einer Berechnung stattfindet, ergibt sich ein entscheidender Vorteil: Der Ingenieur ist nicht mehr auf ein Modell festgelegt. Er kann die Spannungen auf beliebige integrale Größen reduzieren. Dies ist insbesondere dann von Bedeutung, wenn sich durch Modifikationen an der Struktur die Voraussetzungen für ein bestimmtes Modell ändern. Beispielsweise kann es passieren, dass bei Änderung der Höhe eines Unterzuges dieser nicht mehr als Balken, sondern als Scheibe bemessen werden muss. Sind die Größen primär immer im 3D-Spannungszustand gegeben, ist hierzu nur noch eine andere Art der Ergebnisauswertung vorzunehmen.

Aus einer strikt volumenorientierten Herangehensweise ergeben sich neben einer Reduktion auf integrale Größen nun aber weitergehende Möglichkeiten. So lässt sich im Vorfeld schon die grundlegende Annahme der Standardbemessungsverfahren vom Ebenbleiben der Querschnitte überprüfen. Es kann automatisch eine Unterscheidung hinsichtlich B- und D-Zonen getroffen werden, worauf dann die jeweiligen Bemessungsverfahren zur Anwendung kommen können. Diesem Umstand kommt eine besondere Bedeutung zu, wenn man den in der Praxis erkennbaren Trend zu Berechnungen am Gesamtsystem berücksichtigt. Aufgrund der höheren Datenmengen und Komplexität der Modelle besteht wesentlich leichter die Gefahr, dass kritische Bereiche übersehen werden können. Dabei sind vor allem die Bereiche mit diskontinuierlichen Spannungsverläufen der D-Zonen besonders zu berücksichtigen, von deren sorgfältiger Bemessung und konstruktiver Durchbildung die Standsicherheit von Tragwerken maßgeblich abhängt (vgl. hierzu [101]).

Die hier aufgeführten Aspekte sollen nun Gegenstand der Diskussion in den nächsten Unterabschnitten sein. Eine Grundlage stellt dabei ein Verfahren zur Reduktion der dreidimensionalen Spannungsgrößen auf Schnittkräfte dar, welches im Folgenden erläutert wird.

### 5.2.1 Dimensionsreduktion der Spannungen

Ein wichtiges Hilfsmittel für das hier betrachtete Konzept zur Bemessung volumenorientierter Modelle sind Verfahren zur Reduktion der dreidimensionalen Spannungen auf Schnittgrößen bezüglich einer Systemfläche oder Systemachse. Wie vorher angesprochen, liegt die grund-

gende Idee darin, eine Dimensionsreduktion nicht mehr vor, sondern nach einer Berechnung durchzuführen. Damit hat der Planer die Möglichkeit, die Ergebnisse des Volumenmodells hinsichtlich beliebiger integraler Größen auswerten zu lassen. Er ist damit in der Lage, schnell auf Änderungen am Modell zu reagieren und verschiedene Bemessungsverfahren zu bedienen.

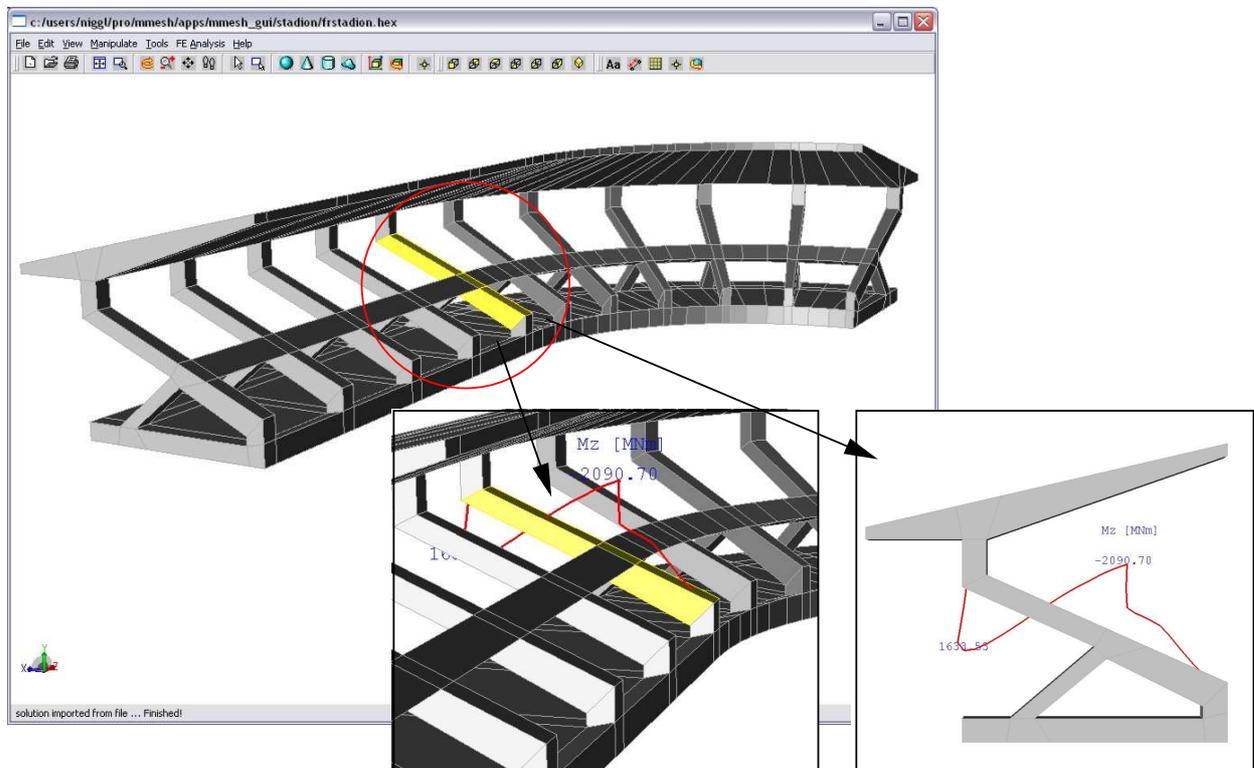


Abbildung 5.3: Vorgehen bei der Bemessung am Volumenmodell: Auswahl der Elemente im Gesamtsystem, Integration und Darstellung des Schnittgrößenverlaufes sowohl am dreidimensionalen Objekt als auch in einer Schnittansicht.

Abbildung 5.3 gibt einen Einblick in die Arbeitsweise: Nachdem unter Anwendung der in Kapitel 3 diskutierten Konzepte der Hierarchie am Gesamtmodell eine Finite-Elemente-Berechnung durchgeführt wurde, kann der Planer in der 3D-Ansicht nun die Elemente selektieren, die er im Weiteren näher untersuchen möchte. Dies werden in der Regel Elemente eines bestimmten Bauteils sein, er kann aber durchaus auch nur ein Teil davon auswählen, wenn er beispielsweise den Flansch eines Plattenbalkens genauer betrachten möchte. Bevor eine Integration von Spannungen durchgeführt wird, wird zunächst für die Elementauswahl eine Stabachse oder Systemfläche festgelegt. Dies kann manuell geschehen, es kann aber auch mithilfe des in Abschnitt 5.3 vorgestellten Verfahrens eine Systemachse automatisch bestimmt werden. Nach Integration der Spannungen lassen sich schließlich die Schnittkraftverläufe direkt in der 3D-Ansicht oder als Diagramm in einem eigenen Fenster darstellen. Da in den meisten Fällen die vorgegebene Systemachse nicht mit der Schwerpunktsachse zusammenfällt, kann vorher noch eine Transformation der Größen auf die Schwerpunktsachse vorgenommen werden.

In dieser Arbeit wurde das Verfahren zur Dimensionsreduktion der volumenbasierten Span-

nungsgrößen exemplarisch für den Fall stabförmiger Bauteile umgesetzt. Alle weiteren Ausführungen orientieren sich an dieser Problemklasse. Prinzipiell lässt sich das Verfahren aber ohne Weiteres auch auf flächenartige Strukturen erweitern.

### 5.2.1.1 Beziehungen am Balken unter schiefer Biegung und Normalkraft

Bevor das Verfahren zur Integration von dreidimensionalen Spannungsgrößen genauer betrachtet wird, sollen zunächst die wichtigsten Beziehungen am Balken aufgezeigt werden. Die folgenden Ausführungen orientieren sich dabei an denen des Lehrbuchs von SCHNELL, GROSS und HAUGER [115]. Abbildung 5.4 zeigt dazu das positive Schnittufer eines Balkens mit allen Schnittgrößen. Die Vorzeichen der Schnittgrößen folgen dabei der Konvention, nach der positive Momente am positiven Schnittufer im Sinne einer Rechtsschraube drehen. Zur Be-

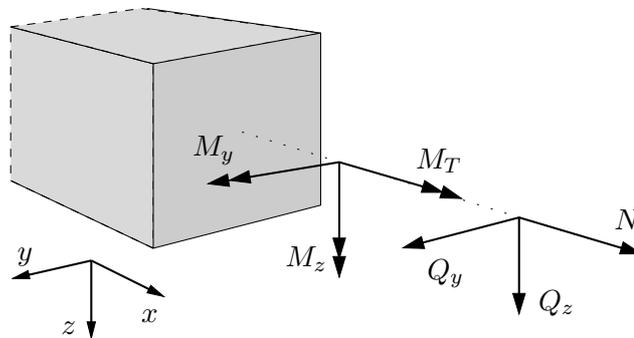


Abbildung 5.4: Momente, Normal- und Querkräfte am Balken.

stimmung der Schnittgrößen werden die über den Querschnitt verteilten Normalspannungen  $\sigma_x$  und Schubspannungen  $\tau_y$  und  $\tau_z$  integriert. Im Einzelnen ergeben sich diese zu:

$$M_y = \int z \sigma dA \quad (5.3) \quad N = \int \sigma dA \quad (5.6)$$

$$M_z = - \int y \sigma dA \quad (5.4) \quad Q_y = \int \tau_y dA \quad (5.7)$$

$$M_T = \int (\tau_{zy} - \tau_{yz}) dA \quad (5.5) \quad Q_z = \int \tau_z dA. \quad (5.8)$$

Neben den Schnittgrößen wird später auch die Fläche und die Flächenmomente erster und zweiter Ordnung benötigt:

$$A = \int dA \quad (5.9) \quad I_y = \int z^2 dA \quad (5.12)$$

$$S_y = \int z dA \quad (5.10) \quad I_z = \int y^2 dA \quad (5.13)$$

$$S_z = \int y dA \quad (5.11) \quad I_{yz} = I_{zy} = - \int yz dA. \quad (5.14)$$

Die vom Benutzer vorgegebene oder automatisch bestimmte Systemachse ist in der Regel nicht mit der Schwerpunktsachse des Systems identisch. Es ist auch weiterhin zu beachten, dass die Achsen  $y$  und  $z$  im Allgemeinen nicht den Hauptachsen des Querschnitts entsprechen. Damit ist das Deviationsmoment  $I_{yz}$  nicht Null und die Biegeanteile  $M_y$  und  $M_z$  dürfen im Weiteren nicht entkoppelt voneinander betrachtet werden.

Nach Integration aller obigen Größen hinsichtlich einer gegebenen Achse lassen sich nun mithilfe der statischen Momente  $S_y$  und  $S_z$  für jeden Querschnitt die Koordinaten des Schwerpunkts bestimmen:

$$y_s = \frac{S_z}{A}, \quad z_s = \frac{S_y}{A}. \quad (5.15)$$

Damit lassen sich obige Größen auf den Schwerpunkt transformieren. Abbildung 5.5 zeigt hierzu nochmalig einen Balkenquerschnitt in der Ansicht mit den beiden Bezugssystemen.

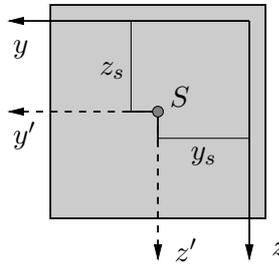


Abbildung 5.5: Transformation der Schnittgrößen auf den Schwerpunkt.

Die Momente und Trägheitsmomente am Schwerpunkt ergeben sich schließlich zu:

$$M_y^S = M_y - N z_s \quad (5.16) \quad I_y^S = I_y - z_s^2 A \quad (5.19)$$

$$M_z^S = M_z + N y_s \quad (5.17) \quad I_z^S = I_z - y_s^2 A \quad (5.20)$$

$$M_T^S = M_T - Q_z y_s + Q_y z_s \quad (5.18) \quad I_{yz}^S = I_{yz} + y_s z_s A. \quad (5.21)$$

Die Normalkraft  $N$  sowie die beiden Querkräfte  $Q_y$ ,  $Q_z$  ändern sich dagegen während einer Transformation in den Schwerpunkt nicht. Die beiden statischen Momente  $S_y$  und  $S_z$  verschwinden.

### 5.2.1.2 Integration der Spannungen auf Schnittgrößen

Im Weiteren soll nun besprochen werden, wie ein dreidimensionaler Spannungszustand innerhalb eines Volumenbauteils auf integrale Schnittgrößen entlang einer vorgegeben Systemachse reduziert werden kann. Ziel ist es vor allem, das Verfahren so zu gestalten, dass eine schnelle und damit weitgehend interaktive Auswertung der Größen möglich ist. Abbildung 5.6 zeigt hierzu einen ebenen Schnitt durch ein Volumenbauteil mit einer vorgegebenen Stabachse  $x$ , hinsichtlich derer Schnittgrößen bestimmt werden sollen.

Gegeben sei dort eine Feldgröße  $s(x, y, z)$  welche beispielsweise eine Spannungsgröße, zur Bestimmung der Querkräfte  $(\sigma_{xz}, \sigma_{xy})$ , Normalkraft  $(\sigma_{xx})$  oder Momente  $(z \sigma_{xx}, y \sigma_{xx})$  beziehungsweise eine Größe zur Berechnung eines Flächenmoments  $n$ -ter Ordnung (z.B.  $z^2$  für  $I_y$ )

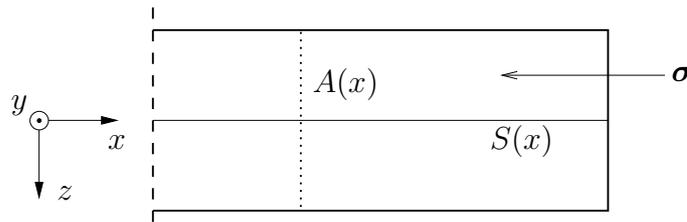


Abbildung 5.6: Integration von Spannungen auf Schnittgrößen.

symbolisieren soll. Zur Bestimmung der auf die Stabachse bezogenen Größe  $S(x)$  wird eine Integration über den Balkenquerschnitt  $A(x)$  durchgeführt:

$$S(x) = \int s(x, y, z) dA = \iint s(x, y, z) dy dz \quad (5.22)$$

Zur numerischen Auswertung des Integrals wird die Stabachse in  $n$  Abschnitte eingeteilt und eine Integration über die jeweils  $n + 1$  Querschnittsflächen vorgenommen. Der entsprechende Funktionsverlauf  $S(x)$  ergibt sich dann näherungsweise aus einer Linearinterpolation von Funktionswerten  $S(x_i)$  an den  $n + 1$  Stützstellen. Für eine Stützstelle  $x = x_i$  gilt:

$$S(x_i) = \int_{A_k} s(x_i, y, z) dA \quad (5.23)$$

Zur Berechnung der Größen  $S(x_i)$  sind Integrale über die Querschnitte an den Stützpunkten  $x_i$  auszuwerten. Hierzu müssen die Flächen zunächst geometrisch beschrieben und in geeigneter Weise (z.B. mittels Triangulation) diskretisiert werden. Die numerische Integration erfolgt dann mithilfe von Quadraturformeln beispielsweise in Form einer GAUSS-Quadratur [21]

$$S(x_i) \approx \sum_{\nu=0}^m s_{\nu} w_{\nu} \quad \text{mit} \quad s_{\nu} = s(x_i, y_{\nu}, z_{\nu}), \quad (5.24)$$

die das Integral durch eine Summe diskreter Funktionswerte  $s_{\nu}$  multipliziert mit Wichtungsfaktoren  $w_{\nu}$  approximiert. Zur Bestimmung der Werte  $s_{\nu}$  ist das Feld  $s(x, y, z)$  an den Stützstellen der Integration auszuwerten. Hierzu müssen für jeden Integrationspunkt (zumindest für Momente und Querkräfte) die Spannungen innerhalb der finiten Elemente des ursprünglichen Volumenbauteils bestimmt werden. Dazu ist eine Transformation der Koordinaten des Integrationspunktes in das lokale Koordinatensystem jedes finiten Volumenelementes vorzunehmen.

Grundsätzlich sind in diesem Zusammenhang drei Koordinatensysteme zu berücksichtigen: lokale Koordinaten  $\mathbf{r} = (r, s, t)^T$  innerhalb der finiten Elemente des Volumenmodells, globale Koordinaten  $\mathbf{x} = (x, y, z)^T$  sowie das (lokale) Koordinatensystem  $(\boldsymbol{\xi} = (\xi, \eta, \zeta)^T)$  entlang der zuvor definierten Stabachse. Abbildung 5.7 stellt die Zusammenhänge dar. Innerhalb der finiten Volumenelemente werden auf einem Standardelement Formfunktionen in den lokalen Koordinaten  $\mathbf{r} = (r, s, t)^T \in [-1; 1]^3$  formuliert. Dieses wird unter Verwendung der Transformation  $\mathbf{x} = \mathbf{Q}^e(\mathbf{r}) = (x, y, z)^T$  auf die tatsächliche Elementgeometrie im globalen Koordinatensystem abgebildet. Die Abbildungsfunktion  $\mathbf{Q}^e$  wird dabei für jedes Element definiert

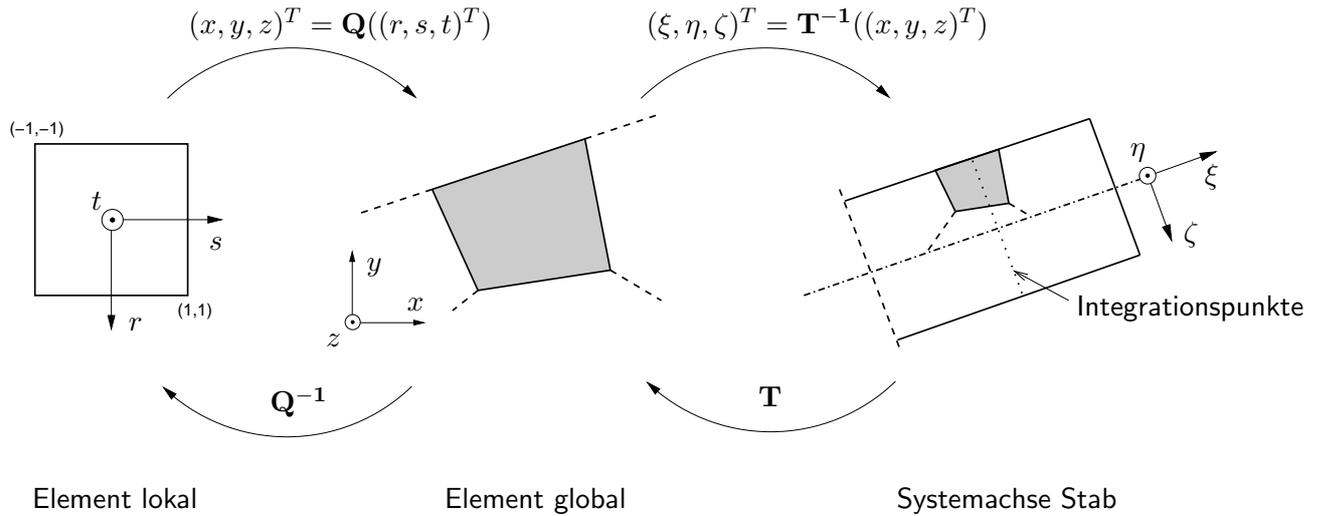


Abbildung 5.7: Koordinatensysteme und Abbildungsfunktionen.

und bestimmt sich aus dessen Geometrieinformation (i.A. Knotenkoordinaten, im Rahmen der  $p$ -Version der FEM auch aus den Kanten und Flächenbeschreibungen unter Verwendung der Blending-Funktionen-Methode [53, 20], siehe hierzu auch Abschnitt 2.2.3 dieser Arbeit). Für die Umkehrabbildung  $\mathbf{r} = \mathbf{Q}^{-1}(\mathbf{x})$  kann nur in Sonderfällen eine geschlossene Lösung gefunden werden. Die Lösung muss in der Regel iterativ bestimmt werden.

Das dritte Koordinatensystem entlang der Stabachse lässt sich im Falle eines geraden Balkens durch orthogonale Basisvektoren  $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$  definieren, welche vom Benutzer vorgegeben oder gemäß Abschnitt 5.3.1 automatisch bestimmt werden können. Weiterhin wird auf der Balkenachse ein Koordinatenursprung  $\mathbf{p}_0$  festgelegt. Die drei Basisvektoren definieren eine Rotationsmatrix  $\mathbf{R} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$  mit deren Hilfe die lokalen Koordinaten  $\boldsymbol{\xi} = (\xi, \eta, \zeta)^T$  des Balkens in globale Koordinaten transformiert werden können:

$$\mathbf{x} = \mathbf{T}(\boldsymbol{\xi}) = \mathbf{R}\boldsymbol{\xi} + \mathbf{p}_0. \quad (5.25)$$

Aufgrund der Orthogonalität von  $\mathbf{R}$  gilt:  $\mathbf{R}^{-1} = \mathbf{R}^T$ . Damit ergibt sich die Umkehrabbildung der Transformation (5.25) zu:

$$\boldsymbol{\xi} = \mathbf{T}^{-1}(\mathbf{x}) = \mathbf{R}^T(\mathbf{x} - \mathbf{p}_0). \quad (5.26)$$

Zur Berechnung der Flächenintegrale in (5.23) bzw. (5.24) sind nun für jede Integrationsstützstelle auf den Querschnittsflächen  $A_i$  die Spannungen zu bestimmen. Hierzu ist der Integrationspunkt  $\boldsymbol{\xi}_\nu = (\xi_\nu, \eta_\nu, \zeta_\nu)$ , gegeben im Koordinatensystem der Balkensystemachse, auf die lokalen Koordinaten des Standardelements zu transformieren:

$$\mathbf{r}_\nu = \mathbf{Q}^{-1}(\mathbf{T}(\boldsymbol{\xi}_\nu)) = \mathbf{Q}^{-1}(\mathbf{T}(\xi_\nu, \eta_\nu, \zeta_\nu)). \quad (5.27)$$

Es ist zu erwarten, dass mit der hier geschilderten Vorgehensweise der Aufwand zur Bestimmung der Schnittgrößen durchaus erheblich ist. Für jeden Integrationspunkt auf allen  $n$  Querschnittsflächen ist eine Transformation nach Gleichung (5.27) durchzuführen. Dabei ist vor allem die Berechnung der Umkehrabbildung  $\mathbf{Q}^{-1}$  als zeitkritisch anzusehen. Außerdem muss

für jeden Auswertepunkt das entsprechende finite Element zunächst lokalisiert werden. Vor einer Integration müssen zuvor auch alle Querschnittsflächen bestimmt und geeignet diskretisiert werden, wofür im Allgemeinen komplexe geometrische Verschnideoperationen notwendig sind.

### 5.2.1.3 Algorithmus zur Dimensionsreduktion

Um den im vorherigen Unterabschnitt erläuterten Aufwand zur Integration von Schnittgrößen zu reduzieren, wird im Folgenden ein Algorithmus vorgeschlagen, der auf eine explizite Bestimmung der Querschnittsflächen verzichtet, eine Berechnung der Umkehrabbildung  $\mathbf{Q}^{-1}$  vermeidet und stattdessen die Integration der Spannungsgrößen auf den Elementen des Volumenbauteils stattfinden lässt. Folgender Zusammenhang soll das entsprechende Vorgehen motivieren. Hierzu wird das Flächenintegral aus Gleichung (5.22) formal als Volumenintegral dargestellt

$$S(\xi) = \int s(\xi, \eta, \zeta) dA \quad \rightarrow \quad S(\xi) = \frac{d}{d\xi} \int_0^\xi s(\xi, \eta, \zeta) dV = \frac{d}{d\xi} W(\xi). \quad (5.28)$$

Zur numerischen Bestimmung der Ableitung in (5.28) lässt sich diese mittels Differenzenquotient darstellen

$$S(\xi_i) \approx \frac{W(\xi_{i+1}) - W(\xi_{i-1})}{\xi_{i+1} - \xi_{i-1}} = \frac{\Delta W}{h}, \quad (5.29)$$

wobei  $h$  der Schrittweite auf der mit  $n + 1$  Stützstellen diskretisierten Systemachse entspricht. Wie zuvor angesprochen, werden die Beiträge  $\Delta W$  nun aber nicht durch Integration über die Querschnittsflächen ermittelt, sondern während einer Volumenintegration innerhalb der finiten Elemente des ursprünglichen Bauteils. Dabei werden die Integrationsbeiträge nicht innerhalb der Elemente aufsummiert, sondern entsprechend ihrer Position zur Systemachse  $\xi$  des Balkens den dort befindlichen Stützstellen hinzu addiert. Abbildung 5.8 stellt das Prinzip an einem ebenen Schnitt durch ein finites Element dar. In einer Schleife über alle Integrationsstützstellen

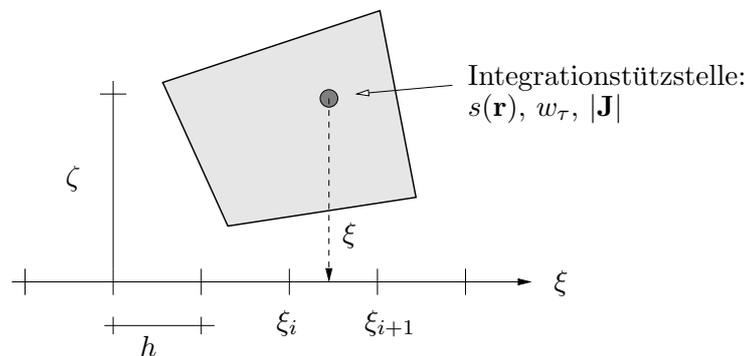


Abbildung 5.8: Prinzip des Algorithmus zur Reduktion. Schnitt durch ein finites Volumenelement.

$\mathbf{r}_\tau$  der Volumenelemente – hierzu lassen sich etwa die Stützstellen und Wichtungsfaktoren einer

GAUSS-Quadratur verwenden – werden die Spannungen ausgewertet und entsprechend ihrer Position  $\xi$  auf die benachbarten Stützstellen der Systemachse verteilt:

$$\Delta S(\xi_i) = \frac{(\xi - \xi_i)}{h} s(\mathbf{r}_\tau) w_\tau |\mathbf{J}| \quad (5.30)$$

$$\Delta S(\xi_{i+1}) = \frac{(\xi_{i+1} - \xi)}{h} s(\mathbf{r}_\tau) w_\tau |\mathbf{J}|. \quad (5.31)$$

Zur Bestimmung der Position  $\boldsymbol{\xi}_\tau = (\xi, \eta, \zeta)_\tau^T$  eines Integrationspunktes hinsichtlich der Stabachse müssen diese vom Standardelement in das Koordinatensystem der Balkenachse transformiert werden:

$$\boldsymbol{\xi}_\tau = \mathbf{T}^{-1}(\mathbf{Q}(\mathbf{r}_\tau)). \quad (5.32)$$

Im Gegensatz zur Transformation (5.27) ist hier keine Umkehrabbildung von  $\mathbf{Q}$  mehr zu bestimmen. Außerdem müssen keine Querschnitte explizit festgelegt und mit dem Bauteil verschnitten und diskretisiert werden.

Da die Integration über das Volumenelement auf dem Standardgebiet formuliert wird, muss in (5.30) und (5.31) die entsprechende Volumenänderung aus der Abbildung vom elementlokalen  $(r, s, t)$  ins globale Koordinatensystem  $(x, y, z)$  mithilfe der Determinanten der JACOBI-Matrix  $|\mathbf{J}|$  berücksichtigt werden. Diese ergibt sich durch Ableitung der Abbildungsfunktion  $\mathbf{Q}$  an den Integrationsstützstellen  $\mathbf{r}_\tau = (r, s, t)_\tau^T$ . Falls die Basisvektoren zur Beschreibung des lokalen Koordinatensystems des Balkens  $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$  normiert wurden, muss hinsichtlich der Transformation  $\mathbf{T}^{-1}$  keine Volumenänderung mehr berücksichtigt werden.

Bei der Auswertung der Spannungen ist zu beachten, dass diese auf das globale Koordinatensystem bezogen sind. Die Spannungsgrößen aus der Finite-Elemente-Berechnung sind dabei als symmetrischer Tensor 2. Stufe  $\boldsymbol{\sigma} = \sigma_{ij}$  definiert. Zur Integration der Schnittkräfte nach den Gleichungen (5.3) bis (5.8) sind die Spannungen dagegen in einem Schnitt senkrecht zur Stabachse  $\boldsymbol{\sigma}^b = (\sigma_\xi, \tau_\eta, \tau_\zeta)^T$  zu bestimmen. Hierzu muss der globale Spannungstensor noch mit dem (normierten) Basisvektor der Systemachse  $\xi$  skalar multipliziert werden:

$$\boldsymbol{\sigma}^b = \boldsymbol{\sigma} \mathbf{e}_1 \Leftrightarrow \sigma_i^b = \sum_{j=1}^3 \sigma_{ij} e_j \quad i = 1, 2, 3. \quad (5.33)$$

Die Güte des Ergebnisses ist bei diesem Vorgehen abhängig von der Einteilung der Stabachse  $h$  und der Integrationsordnung auf dem finiten Element. Der Fehler lässt sich aber gut kontrollieren, wenn zusätzlich die Fläche der Querschnitte in gleicher Weise integriert und dieser Wert mit der tatsächlichen Fläche in Beziehung gesetzt wird. Der Quotient aus beiden Flächenmaßen ergibt einen Faktor, mit dem alle weiteren Größen skaliert werden können. Für gleichmäßige Querschnittsverläufe sind die tatsächlichen Querschnittsflächen leicht zu bestimmen. Bei komplexeren Bauteilen können hierzu aber die Querschnittsebenen mit der Oberfläche der finiten Elemente verschnitten werden und aus dem Randintegral der damit bestimmten Polygonzüge die Referenzflächen auch relativ leicht berechnet werden.

#### 5.2.1.4 Beispiel: auskragender Plattenbalken

Zur Illustration des oben beschriebenen Verfahrens, soll im Folgenden das Beispiel eines auskragenden Plattenbalkens betrachtet werden. Das Modell, dessen Abmessungen und Lasten in

Abbildung 5.9 gegeben sind, könnte im Allgemeinen ein Ausschnitt eines Gesamtmodells sein, welches der Ingenieur für eine detaillierte Bemessung genauer betrachten möchte. Das Modell zeigt eine Stütze mit einem daran befindlichen Kragarm. Der Steg des Plattenbalkens weist zusätzlich einen linearen Verlauf der Höhe auf.

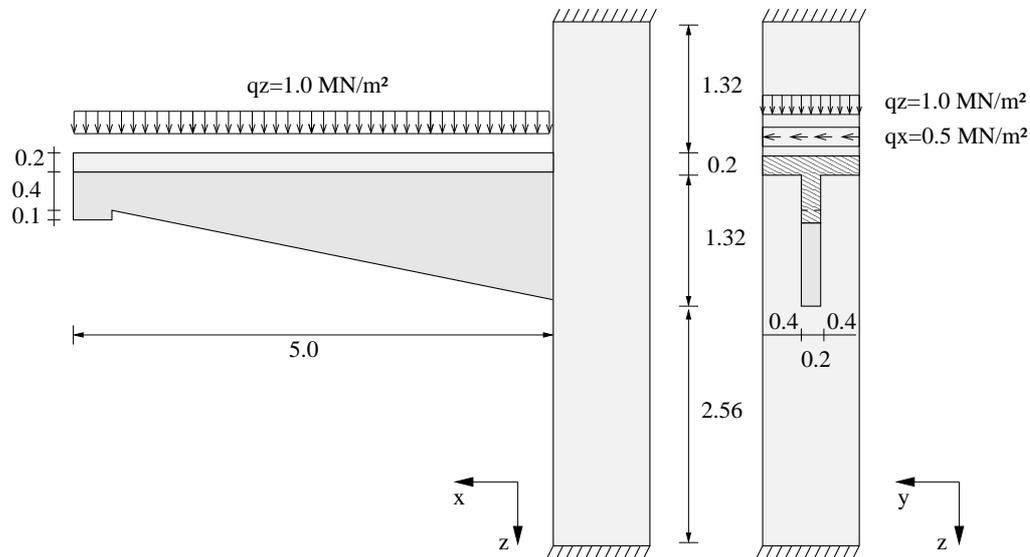


Abbildung 5.9: Abmessungen und Lasten.

Nach einer linear-elastischen Finite-Elemente-Berechnung am Volumenmodell liegen alle Verschiebungen und Spannungen als dreidimensionale, globale Größen vor. In Abbildung 5.10 links sind die Verschiebungen dargestellt, die aus den vertikalen und horizontalen Flächenlasten resultieren.

Basierend auf den Ergebnisdaten im Volumenmodell kann nun die Integration vorgenommen werden. Hierzu wird, wie in Abbildung 5.10 zu sehen ist, eine Systemachse festgelegt, die in einem ersten Schritt als Integrationsachse dient.

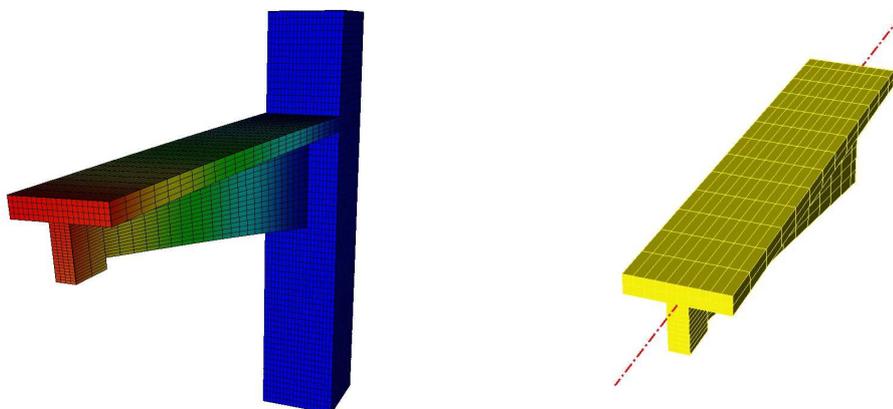


Abbildung 5.10: Verschiebungen am Volumenmodell (rechts) sowie Darstellung der Systemachse, auf die im ersten Schritt alle Schnittgrößen und Flächenmomente bezogen werden (links).

Die tatsächlichen Schnittgrößen werden schließlich nach Bestimmung der Schwerachse nach den Gleichungen (5.16 – 5.21) auf die Schwerpunktsachse transformiert. Die Verläufe der Momente und Querkräfte lassen sich dann direkt am dreidimensionalen Modell oder in eigenen Diagrammen anzeigen. Abbildungen 5.11 und 5.12 zeigen den Verlauf von Schnittmomenten und Querkräften. Die Schnittkräfte sind dabei auf die Schwerachse bezogen, die in den Abbildungen am Verlauf der Nulllinien zu erkennen ist.

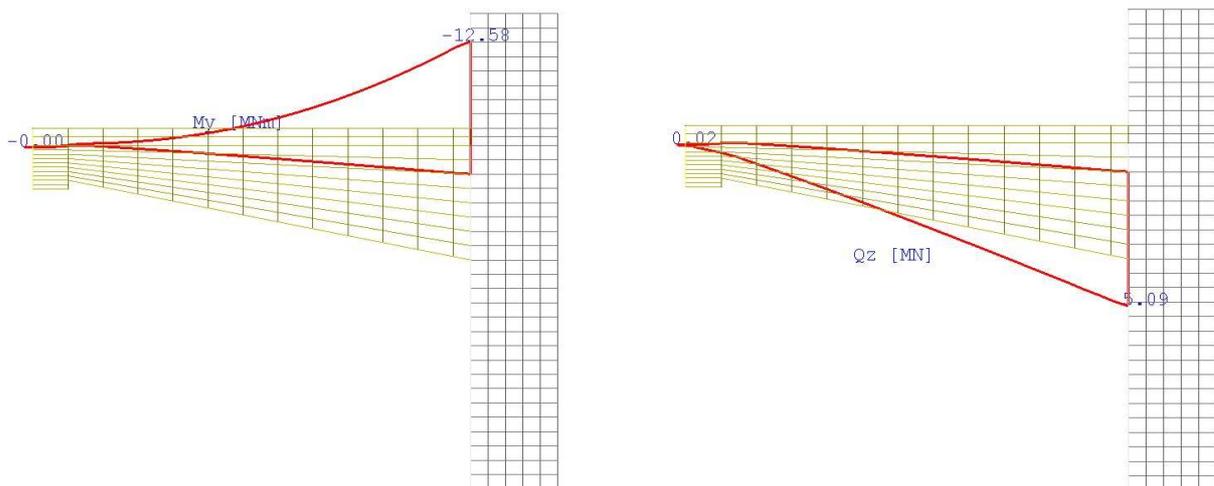


Abbildung 5.11: Verlauf des Schnittmoments  $M_y$  und Querkraft  $Q_z$ .

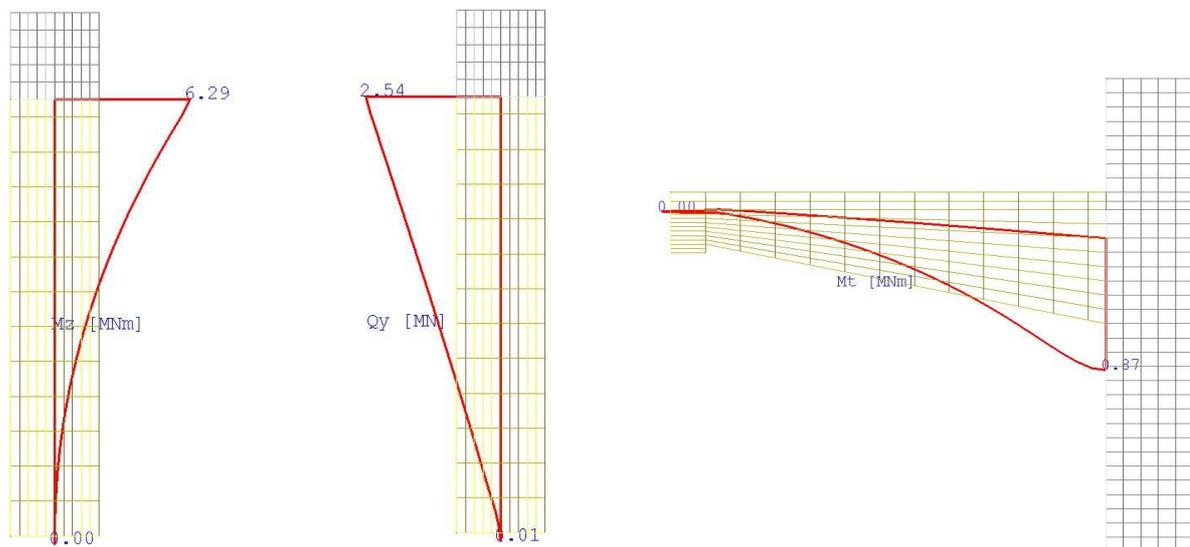


Abbildung 5.12: Verlauf des Schnittmoments  $M_z$ , der Querkraft  $Q_y$  sowie des Torsionsmoments  $M_T$ .

Eine Überprüfung der Ergebnisse am eindimensionalen Modell des Kragarms ergibt für die

Momente

$$M_{y,\max} = -\frac{1}{2} q_z l^2 = -\frac{1}{2} \cdot 1.0 \cdot 5.0^2 = -12.5 \text{ MNm}, \quad (5.34)$$

$$M_{z,\max} = \frac{1}{2} q_y l^2 = \frac{1}{2} \cdot 0.5 \cdot 5.0^2 = 6.25 \text{ MNm}, \quad (5.35)$$

sowie für die Querkräfte

$$Q_{z,\max} = q_z l = 1.0 \cdot 5.0 = 5.0 \text{ MN}, \quad (5.36)$$

$$Q_{y,\max} = q_y l = 0.5 \cdot 5.0 = 2.5 \text{ MN}. \quad (5.37)$$

Was jedoch bei einer Berechnung am eindimensionalen Balkenmodell leicht übersehen werden kann ist die Tatsache, dass die Resultierende der Last aus  $q_y$  und  $q_z$  nicht durch die Schwerachse des Balkens – im allgemeinen Fall die Schubmittelpunktsachse – geht und somit zusätzlich ein Streckentorsionsmoment  $m_T$  eingetragen wird. Bei einer Integration am Volumenmodell wird dies natürlich automatisch berücksichtigt, wie der Verlauf des Torsionsmoments in Abbildung 5.12 ganz rechts zeigt.

Eine Überschlagsrechnung unter Annahme eines linearen Verlaufs des Streckentorsionsmomentes  $m_T$  mit den Hebelarmen  $a_1 = 0.186 \text{ m}$  bzw.  $a_2 = 0.532 \text{ m}$  (= Abstand der Schwerachse von der Oberfläche der Platte) an beiden Enden des Kragarmes liefert für das maximale Torsionsmoment an der Einspannung:

$$M_{T,\max} = \frac{1}{2}(a_1 + a_2) q_x l = \frac{1}{2} \cdot (0.186 + 0.532) \cdot 0.5 \cdot 5.0 = 0.897 \text{ MNm}, \quad (5.38)$$

welches in etwa mit dem in Abbildung 5.12 dargestellten Maximalwert übereinstimmt.

### 5.2.2 Aspekte der Tragwerksidealisierung

Im Rahmen der Modellbildung wird ein reales Tragwerk auf ein oder mehrere statische Systeme abgebildet. Dabei werden Annahmen getroffen und Idealisierungen vorgenommen, die bei der Bemessung von Bauteilen berücksichtigt werden müssen, um etwaige negative Auswirkungen auf das Tragverhalten ausschließen zu können. Die Annahmen der dimensionsreduzierten Modelle mit stab- und flächenhaften Strukturen sowie punktförmigen Last- und Lagerungsbedingungen vernachlässigen in der Regel die realen Bauteilabmessungen. Aufgrund der infinitesimal kleinen Lagerbreiten kann es zu Extremwerten im Momentenverlauf kommen, welche bei endlichen Abmessungen nicht auftreten. Zur Berücksichtigung dieser Aspekte sind innerhalb der Normen Anwendungsregeln formuliert, die etwa bei der Bestimmung des maßgebenden Schnittmoments oder der effektiven Stützweite herangezogen werden. Gerade die Wahl der richtigen Lagerungsbedingungen wirkt sich entscheidend auf die Größe und die Verteilung der Schnittgrößen aus. Eindeutige Lagerbedingungen sind dabei üblicherweise selten und können meist nur dann angenommen werden, wenn entsprechende technische Vorrichtungen, wie etwa bei Brückenlagern, installiert werden. Die realen Bedingungen bewegen sich prinzipiell zwischen den Extremen „frei drehbar“ und „vollständige Einspannung“. Der reale Einspannungsgrad hängt dabei von verschiedenen Faktoren ab. Neben der konkreten Art der Verbindung (monolithisch oder frei aufgelegt) wird dieser auch durch vorhandene Auflasten sowie den Steifigkeitsverhältnissen der benachbarten Bauteile beeinflusst. Derartige Fragen, die im Zusammenhang mit der Idealisierung der Tragsysteme auftreten, sind innerhalb der DIN 1045-1 [27] ab Abschnitt 7.3 geregelt.

Die folgenden zwei Beispiele sollen zeigen, inwieweit sich durch eine strikt volumenorientierte Vorgehensweise Verbesserungen im Hinblick auf die Berücksichtigung der Modellannahmen im Bemessungsprozess ergeben können. Hierzu soll ein Einfeldbalken mit einer konzentrierten Last in der Mitte betrachtet werden. Abbildung 5.13 zeigt als erstes Beispiel einen Einfeldbalken, der an beiden Enden frei drehbar gelagert ist. Mithilfe der  $p$ -Version der FEM wurden hierfür am Volumenmodell die dreidimensionalen Spannungen und Verschiebungen berechnet. Die „ideale“ freie Drehbarkeit an den Auflagern wird durch entsprechende Elemente mit geringerem Elastizitätsmodul gewährleistet. Zusätzlich wird die Last mit einer endlichen Ausdehnung modelliert. Die Diagramme in Abbildung 5.13 zeigen einen Vergleich zwischen einem Momentenverlauf, der durch Integration der Spannungen am Volumenmodell bestimmt wurde und einem, der sich aus der Balkentheorie am idealisierten eindimensionalen Tragsystem ergibt.

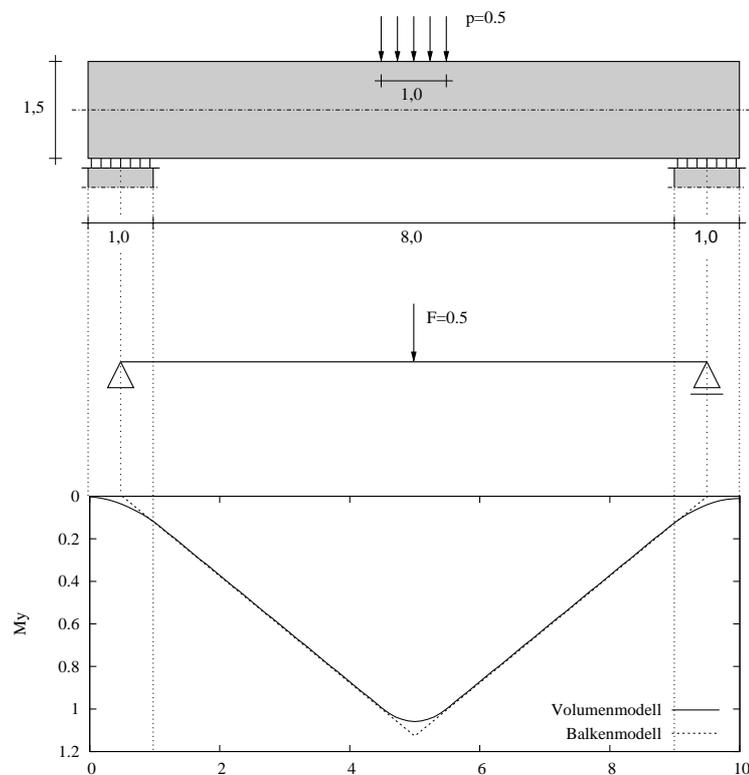


Abbildung 5.13: Vergleich der Schnittmomente  $M_y$  ermittelt aus Integration über die Spannungen des Volumenmodells (oben) sowie berechnet am eindimensionalen Tragsystem (mitte). Annahme einer frei drehbaren Lagerung an den Auflagern.

Wird eine Berechnung am eindimensionalen Tragwerkmodell durchgeführt, ist bei der Modellierung immer eine *effektive Stützweite* des Systems zu berücksichtigen. Diese ist im Allgemeinen durch den Abstand der Resultierenden der Auflagerpressungen vorgegeben [135]. Zur Bestimmung der effektiven Stützweite sind innerhalb der DIN 1045-1 Regeln zu finden. Dort ergibt sich die effektive Stützweite aus einer lichten Stützweite  $l_n$  und zwei Abstandsgrößen ( $a_1, a_2$ ) an den Auflagern:

$$l_{eff} = l_n + a_1 + a_2. \quad (5.39)$$

Die beiden Maße  $a_1$  und  $a_2$  sind dabei von den Auflager- und Einspannbedingungen abhängig und „in geeigneter Weise festzulegen“ (DIN 1045-1 [27], Abs. 7.3.1, (6)). Für das in Abbildung 5.13 betrachtete Beispiel kann die effektive Stützweite zu  $l_{eff} = 9.0$  m angenommen werden, da sich aufgrund der vollständig freien Drehbarkeit die Resultierenden der Auflagerpressung genau in der Mitte der Auflager befinden.

Zur Bestimmung des *maßgebenden Moments* sind innerhalb der DIN 1045-1 weitere Anwendungsregeln definiert. So darf bei einer frei drehbaren Lagerung das negative Stützmoment um einen Betrag  $\Delta M_{Ed}$  reduziert („ausgerundet“) werden (DIN 1045-1 [27], Abs. 7.3.2, (1)). Die Verringerung resultiert aus dem gegenwirkenden Moment der Auflagerpressung, welches von der anderen Seite auf die Stabachse wirkt [135]. Das gleiche gilt in analoger Weise an Bereichen mit lokalen Lastenleitungen.

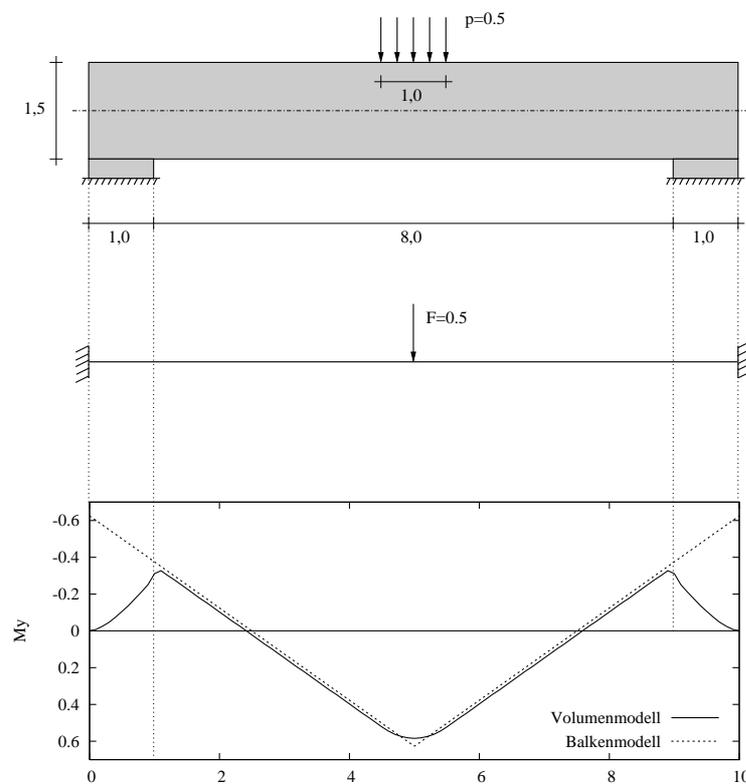


Abbildung 5.14: Vergleich der Schnittmomente  $M_y$  ermittelt aus Integration über die Spannungen des Volumenmodells (oben) sowie berechnet am eindimensionalen Balkensystem (mitte). Annahme einer vollständigen Einspannung an den Auflagern.

Abbildung 5.14 zeigt als zweites Beispiel einen Einfeldbalken mit vollständiger Einspannung an den Auflagern. Wiederum wird deutlich, dass die tatsächlichen Schnittkraftverläufe von der Balkentheorie nicht ausreichend berücksichtigt werden. Zwar stimmt das Biegemoment aus der Balkentheorie in der Mitte des Balkens ( $M_{max} = 1/8 Fl = 0.625$ ) annähernd mit den Werten aus der Dimensionsreduktion überein, an den Auflagern weicht dieses jedoch erheblich vom tatsächlichen Verhalten ab. Innerhalb der Norm wird dieser Umstand bei monolithischer Lagerung an Innenauflagern dadurch berücksichtigt, dass als maßgebende Schnittkraft das Moment an der Auflagerkante verwendet werden darf (DIN 1045-1 [27], Abs. 7.3.2, (3)).

Am Beispiel dieser beiden einfachen Standardfälle wird deutlich, dass mit Verwendung eines Volumenmodells und Berücksichtigung der realen Bauteilabmessungen die integralen Schnittkräfte das tatsächliche Verhalten wesentlich besser abbilden als bei einer Berechnung am dimensionsreduzierten Modell. Dies gilt umso mehr bei komplizierten Strukturen, die nicht mehr einem speziellen Fall eindeutig zugeordnet werden können. Hier können die tatsächlichen Einspannverhältnisse - die beispielsweise aus Auflasten oder unterschiedlichen Steifigkeitsverhältnissen resultieren - durch das Volumenmodell besser abgebildet werden und sind damit nicht mehr ausschließlich von der Beurteilung des Ingenieurs abhängig. Natürlich müssen auch am Volumenmodell Annahmen getroffen werden, insbesondere was die Modellierung der Lagerungen und der Bauteilverbindungen betrifft. Durch die Verwendung des höheren Modells kann aber die Abhängigkeit von den subjektiven Erfahrungen des Planers verringert werden. Gerade im Hinblick auf die Planung am dreidimensionalen Gesamtsystem, in denen das tatsächliche Verhalten der Bauteile noch schwieriger zu erkennen ist, kann hier der Ingenieur durch Verwendung eines Volumenmodells besser unterstützt werden. Modellierfehler lassen sich einfacher kontrollieren und der Aufwand, der für jedes einzelne Bauteil notwendig ist, kann verringert werden, was letztendlich einer Automatisierung des gesamten Berechnungs- und Bemessungsverfahrens entgegenkommt.

### 5.2.3 Erkennung von B- und D-Zonen

Voraussetzung für das Standardbemessungsverfahren von biege- und normalkraftbelasteten Bauteilen ist die Annahme einer linearen Dehnungsverteilung über den Querschnitt. Dies gilt sowohl für Balken als auch für Platten. Gemäß den Erläuterungen in Abschnitt 5.1.2 kann damit aus einer Annahme der Dehnungen im Stahl sowie am Rande der Druckzone im Beton eine Spannungsverteilung ermittelt werden, die mit den einwirkenden Schnittkräften im Gleichgewicht steht. Die *Bernoulli*-Hypothese vom Ebenbleiben der Querschnitte gilt für schlanke Balken ( $l/h \geq 2$ ) mit konstantem oder leicht veränderlichem Querschnittsverlauf hinreichend genau. Ob diese Annahme erfüllt ist, muss allerdings schon während der Modellbildung vom Ingenieur überprüft werden, der dann je nach Bauteil das entsprechende Tragsystem verwendet. Neben der Schlankheit müssen für die Bemessung aber zusätzlich auch Bereiche mit nichtlinearen Dehnungsverteilungen (D-Bereiche) berücksichtigt werden, um überhaupt die entsprechenden Bemessungsverfahren anwenden zu können. Es müssen im Vorfeld der Bemessung Zonen mit konzentrierten Lasteinleitungen oder geometrischen Diskontinuitäten identifiziert und eine Unterscheidung hinsichtlich den B- und D-Zonen gemäß Abschnitt 5.1.3 getroffen werden. In Bereichen mit nicht ebenen Dehnungsverteilungen müssen dann statt den Standardbemessungsverfahren andere Methoden wie etwa Stabwerkmodelle angewendet werden.

Geht man von einer strikt volumenorientierten Modellierung aus, besteht die Möglichkeit, die Unterscheidung von B- und D-Zonen innerhalb eines Bauteils automatisch vornehmen zu lassen. Hierzu bietet es sich an, die ursprünglichen Spannungen  $\sigma$  aus der FE-Berechnung am Volumenmodell mit einem linearen Spannungsverlauf  $\bar{\sigma}$  aus der Balkentheorie, der mit den vorherigen Spannungen im Gleichgewicht steht, zu vergleichen. Da zwischen Spannungen und Dehnungen ein linearer Zusammenhang besteht, kann diese Differenz zwischen den Spannungsverläufen herangezogen werden, um einen Hinweis auf den Verlauf der Dehnungen zu liefern. Gemäß dem in Abschnitt 5.2.1 beschriebenen Verfahren werden hierzu zunächst durch Integration über alle Querschnittsflächen nach den Gleichungen (5.3) bis (5.8) die dreidimen-

sionalen Spannungen auf Schnittgrößen am Querschnitt reduziert. Nach der Balkentheorie kann nun aus diesen Schnittgrößen wiederum eine dreidimensionale Verteilung der Spannungen im Bauteil bestimmt werden. Hierbei muss berücksichtigt werden, dass die  $y$  und  $z$ -Achsen des Koordinatensystems im Allgemeinen nicht mit den Hauptachsen identisch sein müssen. Die Verteilung der Normalspannung ergibt sich daher zu (siehe bspw. in [115]):

$$\bar{\sigma}_{xx}(y, z) = \frac{N}{A} + \frac{1}{\Delta} [(M_y I_z - M_z I_{yz}) z - (M_z I_y - M_y I_{yz}) y] \quad (5.40)$$

mit

$$\Delta = I_y I_z - I_{yz}^2. \quad (5.41)$$

Gleichung (5.40) liefert einen senkrecht zur Stabachse linearen Spannungszustand, der mit den ursprünglichen Größen bezogen auf den Querschnitt im Gleichgewicht steht. Aus beiden Spannungsverläufen lässt sich eine Differenz bestimmen

$$\Delta\sigma_{xx} = |\sigma_{xx} - \bar{\sigma}_{xx}|, \quad (5.42)$$

die schließlich einen Hinweis für die Abweichung von der Balkentheorie liefern kann. Abbildung 5.15 zeigt hierzu die Verteilung von  $\Delta\sigma_{xx}$  für die beiden Beispiele des frei drehbar bzw. eingespannt gelagerten Einfeldbalkens aus dem vorherigen Abschnitt 5.2.2. Die dargestellten Isoflächen zeigen den Verlauf des Fehlers in (5.42) ab einer Abweichung von 5 Prozent, bezogen auf den Maximalwert der aus den Schnittgrößen berechneten Spannungen  $\max(\bar{\sigma}_{xx})$ .

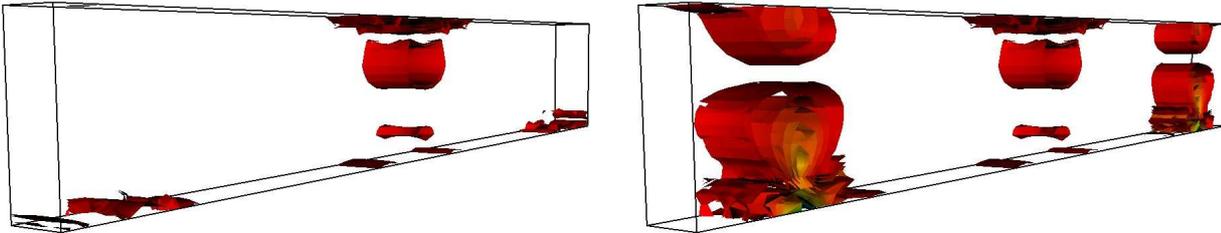


Abbildung 5.15: Darstellung von  $(|\sigma_{xx} - \bar{\sigma}_{xx}| / \max(\bar{\sigma}_{xx})) > 0.05$  für den gelenkig gelagerten (links) sowie beidseitig eingespannten Balken (rechts) aus Abschnitt 5.2.2.

Die beiden Bilder zeigen hier deutlich den stärkeren Einfluss im Einspannbereich des monolithisch gelagerten Balkens. Eine punktweise Verteilung des Fehlers liefert allerdings kein eindeutiges Maß für die Abweichung vom linearen Dehnungszustand. Eine „Ebenheit“ des Dehnungsverlaufes kann sinnvoll nur hinsichtlich einer Stabachse festgestellt werden. Aus diesem Grund betrachtet man den Fehler besser als integrales Maß bezogen auf die Stabachse.

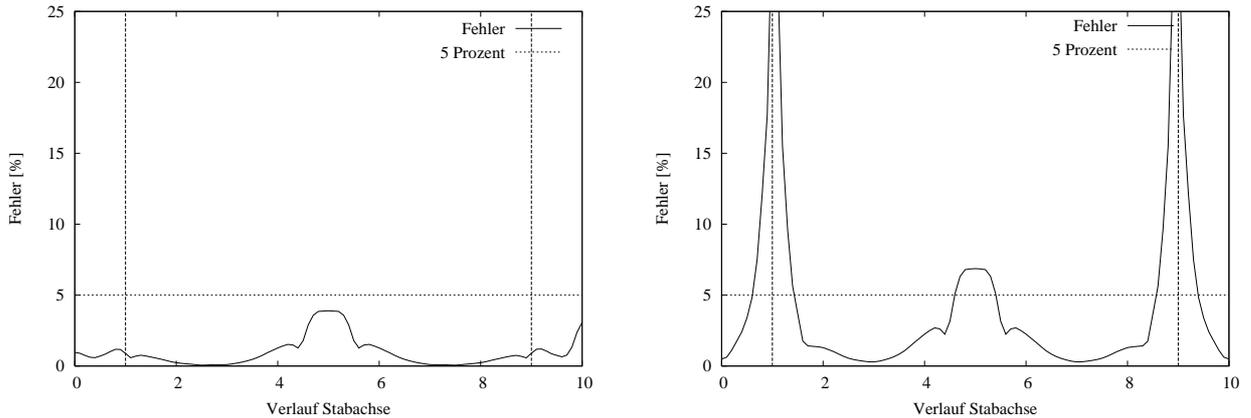


Abbildung 5.16: Darstellung von  $e_x$  für den gelenkig gelagerten (links) sowie beidseitig eingespannten Balken (rechts) aus Abschnitt 5.2.2.

In den von SCHLAICH und SCHÄFER [112] gezeigten Beispielen wird die Ausdehnung der Diskontinuitätsbereiche auch nur senkrecht zur Balkenachse angegeben.

Abbildung 5.16 zeigt hierzu für die Beispiele des gelenkig sowie des monolithisch gelagerten Einfeldbalkens die Abweichung von einer linearen Spannungsverteilung bezogen auf die Stabachse. Nach Berechnung von  $\bar{\sigma}$  laut Gleichung (5.40), wurde das integrale Fehlermaß folgendermaßen bestimmt:

$$e_x = \frac{\int_A (\sigma_{xx} - \bar{\sigma}_{xx})^2}{\max\left(\int_A (\sigma_{xx})^2\right)}. \quad (5.43)$$

In den Diagrammen ist eine Fehlermarke von fünf Prozent gekennzeichnet, die beispielsweise als Unterscheidungskriterium zwischen B- und D-Zonen herangezogen werden kann. Die Ausdehnung der Diskontinuitätsbereiche erstreckt sich dann auf etwa einen Meter längs der Stabachse. Dies deckt sich in etwa mit den Beispielen in [112], in denen eine Ausdehnung der D-Bereiche entsprechend der Höhe des Balkens angenommen wird, welche in den hier gezeigten Beispielen 1,5 Meter beträgt.

## 5.3 Hilfsmittel zur grafisch interaktiven Bemessung

### 5.3.1 Erkennung von Bauteilform und Bauteilachse bzw. -fläche

Ausgangspunkt für die Bemessung nach Standardverfahren sind integrale Spannungsgrößen reduziert auf Momente, Normal- und Querkräfte auf der Systemachse eines Balkens bzw. Systemebene eines flächenartigen Bauteils. In einer interaktiven Arbeitsweise soll der Ingenieur am Gesamtmodell einzelne Bereiche auswählen und die Spannungsgrößen auf die Systemachse reduzieren lassen. Als Unterstützung sind hierbei zusätzliche Hilfsmittel wünschenswert, die es erlauben, automatisch den Bauteiltyp zu erkennen und eine Systemachse bzw. -ebene festzulegen. Prinzipiell könnten derartige Informationen schon im Gebäudemodell vorhanden sein, welche hier nun herangezogen werden können. In der Praxis sind jedoch nicht immer alle Informationen vollständig und fehlerfrei. Weiterhin kann es bei dem hier betrachteten Szenario durchaus auch vorkommen, dass die Auswertung nur auf Teilbereiche einer Struktur beschränkt werden soll.

Im Folgenden soll hierzu ein Hilfsmittel zur Erkennung von Bauteilform und Bestimmung einer Systemachse vorgestellt werden. Die Aufgabe, die sich dabei stellt, ist prinzipiell eine Aufgabe der *Mustererkennung* (engl. *pattern recognition*). In der Mustererkennung werden Verfahren zur maschinellen Analyse und Klassifizierung digitaler Daten, etwa zur Erkennung von Sprache oder Schrift, zur Auswertung von Bilddaten in der Robotik und der Biometrie oder für Aufgaben der Datenanalyse (*Data Mining*) angewandt. Nach DUDA et al. [31] werden in der Mustererkennung drei Teilschritte klar voneinander getrennt: Zunächst müssen die Daten in geeigneter Weise erfasst und aufbereitet werden (*Preprocessing*). In der so genannten *Merkmalsgewinnung* werden dann die maßgebenden Eigenschaften der zuvor ermittelten Daten herausgearbeitet. Hierzu werden oft Methoden der Statistik angewandt, etwa in Form von Erwartungswerten oder Varianzen einer Verteilung. Basierend auf einen Satz aussagekräftiger Merkmale können schließlich die Daten *klassifiziert* und eine Aussage über deren Inhalt getroffen werden.

Die Grundidee des hier gezeigten Vorgehens zur Erkennung der Bauteilform folgt prinzipiell diesem Schema, wenn auch auf einem einfacheren Niveau. Nachdem innerhalb des Finite-Elemente-Modells vom Planer eine Auswahl von Elementen getroffen wurde, werden diese zunächst geeignet aufbereitet, bevor dann ein *minimal umhüllender Quader* bestimmt wird, der die ausgewählten Objekte mit einem möglichst geringem Volumen umschließt. An Hand dieses Quaders (auch: *oriented bounding box (OBB)* [54]) kann dann die Bauteilform bestimmt, sowie eine Stabachse bzw. eine Systemfläche festgestellt werden. In dem Zusammenhang ist anzumerken, dass Aufgaben der Erkennung und Klassifizierung von Bauteilen auch in ganz anderen Bereichen der Bauplanung von Interesse sind. Von SCHLEINKOFER et al. [113] wird hierzu beispielsweise ein Verfahren vorgestellt, mit welchem aus den räumlichen Daten einer optischen Laseraufnahme eines Gebäudes Bauteile erkannt, klassifiziert und daraus ein Produktdatenmodell abgeleitet werden kann. Aufgaben wie diese sind etwa in der Altbausanierung von Interesse, wenn keine digitalen Pläne vorliegen. Andere Möglichkeiten ergeben sich in der Bauausführung, wenn etwa Baufortschritte mithilfe digitaler Systeme kontrolliert werden müssen.

Zur Bestimmung minimal umhüllender Quader existieren unterschiedliche Verfahren (z.B. [88]). Minimal umhüllende Quader finden in vielen Bereichen der geometrischen Modellierung wie beispielsweise der Kollisionserkennung, der Bildbearbeitung oder eben in der Formerken-

nung Anwendung. Das Vorgehen in dieser Arbeit orientiert sich an den Arbeiten von GOTT-SCHALK et al. [54], in denen *Oriented Bounding Boxes* zur Interferenzerkennung polygonaler Objekte verwendet werden.

Vor einer detaillierten Erläuterung des Verfahrens sollen zunächst die einzelnen Arbeitsschritte kurz im Überblick aufgelistet werden. Ausgangspunkt ist eine Auswahl finiter Elemente  $E_s$ , für die eine Reduktion der Spannungsgrößen durchgeführt werden soll. Folgende Schritte sind notwendig, die auch in Abbildung 5.17 bildlich dargestellt sind:

1. Bestimme zunächst für die Menge aller Knoten  $V$  in  $E_s$  die konvexe Hülle  $conv(V)$ .
2. Bestimme für die Oberfläche der konvexen Hülle die Kovarianzmatrix  $\mathbf{C} \in \mathbb{R}^{3 \times 3}$ .
3. Die Eigenvektoren der Kovarianzmatrix bestimmen die drei Richtungen der Quaderkanten. Nach Normierung bilden diese eine orthogonale Basis.
4. Die Punkte der konvexen Hülle mit extremalen Koordinaten bezüglich der neuen Basis bestimmen die konkreten Abmessungen des Quaders, aus denen schließlich der Typ des Bauteils bestimmt werden kann.

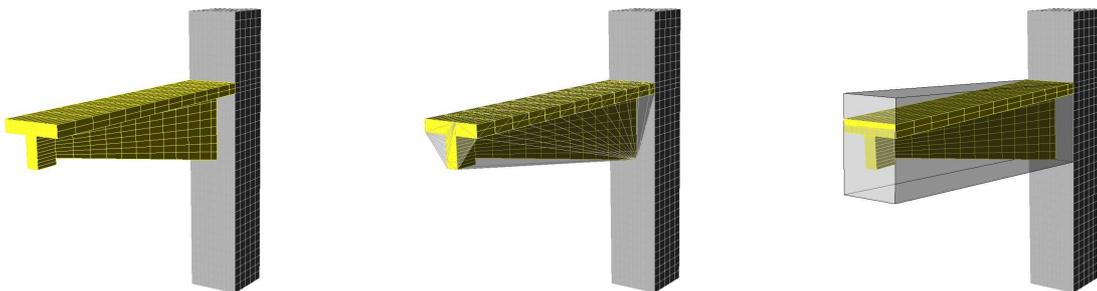


Abbildung 5.17: Auswahl der Elemente, Ermittlung der konvexen Hülle und Bestimmung des minimal umhüllenden Quaders.

### 5.3.1.1 Konvexe Hülle

Die Form und Richtung der minimal umschließenden Box soll unabhängig sein von der Verteilung der Punkte im „Inneren“ der selektierten Elemente. Um den Einfluss des Inneren der Struktur auszuschließen, wird zunächst die *konvexe Hülle* gebildet [88]. Die konvexe Hülle  $conv(S)$  einer Menge von Punkten  $S$  im Raum ist durch das kleinste konvexe Polygon gegeben, welches alle Punkte in  $S$  enthält. Zur Bestimmung der konvexen Hülle existieren verschiedene Verfahren. Der hier verwendete so genannte *inkrementelle Algorithmus* zur Bestimmung der konvexen Hülle von Punktmengen im dreidimensionalen Raum hat eine Komplexität von  $O(n^2)$  und wird eingehend von O'ROURKE [89] erläutert. Der Autor bietet zudem eine Implementierung des Verfahrens als Quellcode im Internet zum Download<sup>1</sup> bereit. In den hier

<sup>1</sup><http://maven.smith.edu/~orourke/books/ftp.html>

betrachteten Beispielen wurden bilineare Elementflächen angenommen und die Konvexe Hülle für die Knoten des Finite-Elemente-Netzes gebildet. Bei gekrümmten Elementoberflächen muss zunächst noch eine Triangulierung oder eine Auswertung der Elementgeometrie auf einem feinen Hintergrundnetz vorgenommen werden.

### 5.3.1.2 Kovarianzmatrix

Zur Bestimmung minimal umhüllender Quader wenden GOTTSCHALK et al. [54] Verfahren der Statistik mehrdimensionaler Zufallsgrößen an. Als Erweiterung zum eindimensionalen Fall lassen sich für eine  $n$ -dimensionale Zufallsgröße  $\mathbf{X} = (X_1, X_2, \dots, X_n)^T$  der Vektor der Erwartungswerte sowie die  $n^2$  paarweisen Kovarianzen  $cov(X_j, X_k)$  angeben. In [21] werden diese auch als die ersten und zweiten zentralen Momente der Zufallsgröße bezeichnet. Nähere Informationen zum Thema Kovarianzen einer Zufallsgröße findet der Leser im Anhang A.1 sowie in der Literatur [13, 21].

Analog dem Vorgehen in der Statistik lassen sich diese ersten und zweiten Momente auch für Punktmengen des dreidimensionalen Raumes bestimmen. Im Falle der Oberfläche  $H$  der konvexen Hülle handelt es sich dabei um stetige Größen, so dass die beiden Momente durch Integration gebildet werden. Die ersten Momente  $\alpha_j^H$  bestimmen sich aus:

$$\alpha_j^H = \frac{1}{A^H} \int_H x_j dA := s_j, \quad j = 1, 2, 3. \quad (5.44)$$

Diese beschreiben grundsätzlich die „Erwartungswerte“ der Punkte in  $H$  bezüglich jeder Dimension. Sie entsprechen geometrisch den Koordinaten des Schwerpunkts der konvexen Hülle. Die  $3 \times 3$  zentralen Momente zweiter Ordnung

$$\mu_{jk}^H = \frac{1}{A^H} \int_H (x_j - \alpha_j^H)(x_k - \alpha_k^H) dA := C_{jk}, \quad j, k = 1, 2, 3 \quad (5.45)$$

stellen in der Statistik die Kovarianzen bezogen auf den Erwartungswert dar und können in diesem Falle auch als eine Art von „Trägheitsmomenten“ der Oberfläche gedeutet werden. Die neun Einträge lassen sich in der so genannten Kovarianzmatrix  $\mathbf{C}$  speichern. Weitere Informationen hierzu sind wiederum im Anhang A.2 zu finden.

Die Anwendung des inkrementellen Algorithmus in [89] liefert die konvexe Hülle in Form einer triangulierten Oberfläche. Für ein derartiges Netz aus Dreiecken lassen sich für obige Flächenintegrale (5.44) und (5.45) Lösungen in geschlossener Form herleiten, die allein aus den Koordinaten der Dreiecksknoten bestimmt werden können. Im Folgenden werden hierzu die notwendigen Schritte zur Berechnung aufgeführt. Eine Herleitung dieser Ausdrücke findet sich wiederum in Anhang A.2.

Gegeben ist die Oberfläche einer konvexen Hülle bestehend aus  $n$  Dreiecken. Jedes Dreieck  $i$  ist durch seine drei Eckpunkte  $\{\mathbf{p}^i, \mathbf{q}^i, \mathbf{r}^i\}$  beschrieben. Die Fläche jedes Dreiecks ergibt sich zu

$$A^i = \frac{1}{2} \|(\mathbf{q}^i - \mathbf{p}^i) \times (\mathbf{r}^i - \mathbf{p}^i)\| \quad (5.46)$$

und eine Summation über alle Dreiecke ergibt die Oberfläche der gesamten Hülle:

$$A^H = \sum_{i=1}^n A^i. \quad (5.47)$$

Mit der Berechnung der Schwerpunkte eines jeden Dreiecks

$$\mathbf{s}^i = \frac{1}{3}(\mathbf{p}^i + \mathbf{q}^i + \mathbf{r}^i) \quad (5.48)$$

lässt sich der zentrale Schwerpunkt der konvexen Hülle

$$\mathbf{s}^H = \frac{\sum_i \mathbf{s}^i A^i}{\sum_i A^i} = \frac{\sum_i \mathbf{s}^i A^i}{A^H} \quad (5.49)$$

als gewichtete Summe aller Dreiecksmittelpunkte bestimmen. Mit diesen Größen lassen sich schließlich die Elemente der Kovarianzmatrix berechnen:

$$C_{jk} = \frac{1}{12 A^H} \sum_{i=1}^n A^i (9s_j^i s_k^i + p_j^i p_k^i + q_j^i q_k^i + r_j^i r_k^i) - s_j^H s_k^H, \quad j, k = 1, 2, 3. \quad (5.50)$$

### 5.3.1.3 Bestimmung von Größe und Richtung des Quaders

Die Eigenvektoren der Kovarianzmatrix bestimmen die drei Richtungen der jeweils paarweise parallelen Quaderkanten. Die Eigenvektoren einer symmetrischen Matrix wie der Kovarianzmatrix  $\mathbf{C}$  sind dabei zueinander orthogonal. Die Bestimmung von Eigenwerten und Eigenvektoren kann für eine Matrix des  $\mathbb{R}^{3 \times 3}$  noch durch Lösung des charakteristischen Polynoms dritter Ordnung aus  $\det(\mathbf{C} - \lambda \mathbf{I}) = 0$  bestimmt werden. Eine andere Möglichkeit zur approximativen Bestimmung der Eigenvektoren einer reellen symmetrischen Matrix ist die Anwendung einer *zyklischen Jacobi Iteration* (z.B. [117]), die den Vorteil bietet, dass nach der Lösung neben allen Eigenwerten auch gleich die normierten Eigenvektoren bekannt sind und nicht mehr nachträglich bestimmt werden müssen.

Die normierten Eigenvektoren bilden nun eine orthogonale Basis innerhalb des minimal umhüllenden Quaders. Aus diesen Vektoren kann die in Gleichung (5.25) verwendete Rotationsmatrix  $\mathbf{R} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$  bestimmt werden, mit der sich die Transformation der lokalen Koordinaten am Balkenquerschnitt auf globalen Koordinaten herstellen lässt:

$$\mathbf{x} = \mathbf{T}(\boldsymbol{\xi}) = \mathbf{R} \boldsymbol{\xi} + \mathbf{p}_0. \quad (5.51)$$

Die Ausdehnung des minimal umhüllenden Quaders lässt sich nach Transformation aus den extremalen Koordinaten der Hülle  $H$  bestimmen. Sind die Abmessungen bekannt, kann durch Vergleich der Achsen der Typ des Bauteils sowie die maßgebende Stabachse bzw. -fläche ermittelt werden. Auf dieser wird dann schließlich ein Ursprung  $\mathbf{p}_0$  für das stablokale Koordinatensystem festgelegt.



# Kapitel 6

## Zusammenfassung und Ausblick

Im Mittelpunkt der vorliegenden Arbeit stand eine neue Herangehensweise zur Berechnung von Tragwerken des konstruktiven Ingenieurbaus. Ziel war es, Tragstrukturen nicht mehr je nach Tragverhalten mittels dimensionsreduzierter Elemente zu modellieren und berechnen, sondern alle Analysen einheitlich am Volumenmodell durchzuführen. Die Arbeit entstand im Rahmen eines Forschungsvorhabens, in welchem auf Grundlage strikt volumenorientierter Modelle die computergestützte Zusammenarbeit zwischen den Planungsbeteiligten verbessert werden soll. Einige der dort diskutierten Fragestellungen wurden in der vorliegenden Arbeit aufgegriffen, im Hinblick auf Arbeitsprozesse der Tragwerksplanung genauer untersucht und Lösungsmöglichkeiten aufgezeigt.

Eine Modellierung und Berechnung von Tragwerken einheitlich auf Basis volumenorientierter Strukturen stellt im Vergleich zur gängigen Praxis eine komplett neue Herangehensweise dar, die wichtige Vorteile mit sich bringt. Diese lassen sich grob nach drei Stichpunkten untergliedern:

### **Kooperative Zusammenarbeit:**

- Wird einheitlich ein Volumenmodell verwendet, verringert sich die Anzahl unterschiedlicher Teilmodelle. Damit sind weniger Schnittstellen zu berücksichtigen, was die Wartung angeschlossener Softwaresysteme vereinfacht. Zudem lassen sich Redundanzen zwischen den Modellen leichter vermeiden.
- Die geringere Anzahl an Modellvarianten sowie eine eindeutige geometrische Darstellung erleichtern die Konsistenzsicherung am gemeinsamen Modell.
- Zudem ist aufgrund einer einheitlichen Semantik kein Bruch zwischen Architektur- und Tragwerksmodell mehr gegeben. Lasten beispielsweise können direkt am Volumenmodell definiert werden und gehen bei einer Modifikation am Ausgangsmodell nicht mehr verloren. Auch ist eine Rücktransformation von Ergebnisdaten einfacher.

### **Numerische Berechnung:**

- Mit Verwendung der  $p$ -Version der Finite-Elemente-Methode lassen sich gemischt massive und dünnwandige Strukturen einheitlich mittels Volumenelementen berechnen. Elemente der  $p$ -Version verhalten sich im Allgemeinen sehr robust gegenüber Locking und verzerrter Elementgeometrie. Dieser Umstand erleichtert in Kombination mit der einheitlichen Elementformulierung die Automatisierung von Berechnungsaufgaben.

- Die Verwendung anisotroper Polynomgrade erlaubt die beliebige Anpassung des Finite-Elemente-Ansatzes an die jeweiligen Verhältnisse innerhalb eines Bauteils. Im Gegensatz zu dimensionsreduzierten Elementen tritt dabei eine ungenaue Anpassung an das reale Bauteilverhalten nicht als Modellfehler, sondern nur als Diskretisierungsfehler auf. Der Fehler aus der Diskretisierung lässt sich aber numerisch kontrollieren und mittels adaptiver Verfahren automatisch reduzieren.

### Modellierung:

- Bereiche mit dreidimensionalen Spannungsverhältnissen sind oft ausschlaggebend für das Tragverhalten einer Struktur. Mit einer einheitlichen Verwendung von Volumenelementen werden derartige Zonen sicher erfasst. Dieser Punkt ist besonders wichtig, wenn Tragstrukturen am Gesamtsystem berechnet werden, in denen der Überblick schwer zu behalten ist.
- Bei der Ausbildung von Knotenpunkten werden Einflüsse aus der geometrischen Form automatisch berücksichtigt. Werden stattdessen dimensionsreduzierte Elemente verwendet, müssen Exzentrizitäten und unterschiedliche Bauteilabmessungen bei der Beurteilung der Einspannverhältnisse vom Planer explizit modelliert werden.

Die vorliegende Arbeit hatte zum Ziel, Teilaspekte im Rahmen einer strikt volumenorientierten Vorgehensweise in der Tragwerksplanung zu diskutieren und Lösungsmöglichkeiten aufzuzeigen. Ein Schwerpunkt beschäftigte sich mit Konzepten der hierarchischen Organisation eines Berechnungsmodells. Ausgangspunkt waren die Arbeiten von MUNDANI et al. [79], in denen Oktalbäume sowohl der Herstellung von Rauminformationen innerhalb eines volumenorientierten Gesamtmodells als auch der Organisation einer Finite-Elemente-Berechnung dienen. Dieses Vorgehen wurde aufgegriffen und neben Oktalbüumen auch ein Graphen-basierter Ansatz zur hierarchischen Strukturierung untersucht. Dabei stellte sich heraus, dass das hierarchische Lösungsverfahren der rekursiven Substrukturierung in Verbindung mit den relativ dicht besetzten Matrizen der  $p$ -Version optimale Voraussetzungen zur Anwendung von Hardwareeffizienten Bibliotheken bietet. Damit lassen sich während der Lösung des Finite-Elemente-Gleichungssystems Leistungen erzielen, die mit anderen modernen direkten Lösungsverfahren für schwach besetzte Matrizen vergleichbar sind. Der besondere Vorteil des hier gewählten Ansatzes liegt aber darin, dass nach lokalen Modifikationen der Struktur der Aufwand zur Lösung des Gleichungssystems weitgehend auf den veränderten Bereich beschränkt werden kann, wodurch sich zusätzliche Geschwindigkeitsvorteile erzielen lassen. Gerade bei „stabartigen“ Strukturen, wie etwa bei Brückensystemen lässt sich dies optimal ausnutzen, da die entsprechenden Element-Gleichungssysteme relativ stark entkoppelt sind. Darauf aufbauend wurde in Kapitel 4 ein Ansatz vorgestellt, in dem zur Simulation von Bauphasen ein Finite-Elemente-Modell mit einer Projektplanungs-Software gekoppelt wird. Welche Auswirkungen Änderungen am Projektplan auf das Tragwerksmodell ausüben, kann so schnell und interaktiv ermittelt werden.

In einem zweiten Schwerpunkt wurde untersucht, wie sich die volumenorientierten Ergebnisdaten zur Bemessung von Stahlbetonbauteilen heranziehen lassen. Es wird dabei vorgeschlagen, die Dimensionsreduktion nicht mehr vor, sondern nach einer Berechnung durchzuführen und auf Basis dieser Daten die Standard-Bemessungsverfahren anzuwenden. Hierzu wurden in der Arbeit Hilfsmittel zur Dimensionsreduktion und zur Erkennung von Bauteilformen

---

vorgestellt. Im Vergleich zur linear-elastischen Berechnung von Tragwerken mittels dimensionsreduzierter Strukturen bietet dieses Vorgehen folgende Vorteile:

- Die Dimensionsreduktion findet nur noch innerhalb des Postprocessing statt und lässt sich damit schnell und interaktiv durchführen. Dies eröffnet eine wesentlich flexiblere Arbeitsweise: Bauteile lassen sich in beliebiger Art auswerten. Besonders bei Änderungen an der Struktur (z.B. ein Balken wird zur Scheibe) sind keine neuen Modelle mehr zu erstellen, sondern nur noch eine andere Art der Auswertung vorzunehmen.
- Bei der Reduktion der dreidimensionalen Spannungen auf Schnittgrößen werden Effekte aus der tatsächlichen geometrischen Form besser erfasst, wie etwa bei der Bestimmung des maßgebenden Schnittmoments oder bei der Ermittlung der effektiven Stützweite.
- Auf Basis der dreidimensionalen Spannungen lässt sich eine der Grundannahmen der Standard-Bemessungsverfahren, das „Ebenbleiben der Querschnitte“, überprüfen. Damit lassen sich Hilfsmittel entwickeln, die automatisch eine Unterscheidung nach B- und D-Zonen treffen können und so eine Entscheidungsgrundlage zur Wahl entsprechender Bemessungsverfahren bieten.

Das hier beschriebene Vorgehen wirft neben den oben genannten Vorteilen völlig neue Fragen auf. Sie können Ausgangspunkt für weitergehende Forschungsaktivitäten sein:

- Eine durchgehend volumenorientierte Herangehensweise stellt höhere Anforderungen an die geometrische Modellierung. Um insbesondere komplexe Strukturen sicher und robust bearbeiten zu können, ist an dieser Stelle vor allem das in Kapitel 2 betrachtete Verfahren zur Dekomposition und Vernetzung eines Volumenmodells weiterzuentwickeln. Wie das Beispiel in Abschnitt 2.3 gezeigt hat, ist das Netz an entsprechenden Stellen geeignet zu verfeinern, um eine deutlichere Reduzierung des Diskretisierungsfehlers zu erreichen. Zudem reicht in vielen Fällen die alleinige Verwendung von Hexaederelementen nicht aus. Die Elementfamilie der  $p$ -Version müsste hier topologisch um Tetraeder und Pentader erweitert werden. Es ist aber zu bedenken, dass diese Elemente anisotrope Ansätze verbieten.
- Eine der Voraussetzungen der betrachteten Vorgehensweise besteht darin, dass das Architekturmodell schon als Volumenmodell vorliegen muss. Dies widerspricht jedoch der top-down Charakteristik des Bauplanungsprozesses. Gewöhnlicherweise wird die exakte geometrische Form eines Bauteils erst im Laufe einer Planung entwickelt. Moderne CAD-Systeme erlauben allerdings schon in frühen Planungsphasen eine Modellierung am strikt volumenorientierten Objektmodell. Während der Modellierung werden dort zunächst Standardbauteile eingefügt, die aber schon eine geometrische Darstellung beinhalten und welche in weiteren Planungsschritten durch Änderung des Objekttyps detaillierter spezifiziert werden können.
- In den Beispielen dieser Arbeit wurden ausschließlich monolithische Verbindungen zwischen den Volumenbauteilen betrachtet. Neben einer monolithischen sind in der Praxis allerdings auch andere Lagerungsbedingungen anzutreffen. Es sind an dieser Stelle Verfahren zu entwickeln, die auch für Volumenmodelle eine realistische Modellierung der

Bauteilverbindungen erlauben. Dies könnte unter Umständen in Form geeigneter Übergangselemente geschehen. Es ist in dem Zusammenhang zu berücksichtigen, dass dimensionsreduzierte Strukturen oft leichter die Möglichkeit bieten, ein bestimmtes Tragverhalten zu simulieren. Wenn etwa ein Balken nicht auf Torsion bemessen werden soll, kann diese Tragwirkung leicht während der Berechnung ausgeschlossen werden. Eine derartige „Vorwegnahme“ der Tragwirkung ist am Volumenmodell nur schwer zu modellieren.

- Eine rein linear-elastische Betrachtung ohne Berücksichtigung von Beton und Stahl im Verbund ist im Stahlbetonbau oft nicht ausreichend. Durch Einführung ideeller Querschnitte oder nichtlinearer Momenten-Krümmungs-Beziehungen lässt sich bei dimensionsreduzierten Elementen das nichtlineare Bauteilverhalten relativ leicht erfassen. Eine Darstellung der Baustoffe im Dreidimensionalen unter Berücksichtigung des nichtlinearen Materialverhaltens gestaltet sich dagegen schwieriger. Auf der anderen Seite eröffnen sich dadurch aber auch Möglichkeiten zu einer noch realistischeren Modellierung.

Neben Fragestellungen des konstruktiven Hochbaus sind für weitergehende Aktivitäten auch andere Bereiche des Bauingenieurwesens zu betrachten. Im Grundbau spielt eine dreidimensionale Modellierung der Strukturen eine ganz wesentliche Rolle. Das Verhalten der Materialien hängt stark von der Baugeschichte ab, wobei die einzelnen Bauphasen detailliert zu planen sind. In dem Zusammenhang können die in der vorliegenden Arbeit betrachteten Konzepte der hierarchischen Strukturierung des Berechnungsmodells zur Organisation einer Neuberechnung sowie der Integration einer Prozessplanung zur Bauphasensimulation einen wichtigen Beitrag liefern.

# Anhang A

## Bestimmung der Kovarianzen einer konvexen Hülle

### A.1 Kovarianzen einer mehrdimensionalen Zufallsgröße

In der Wahrscheinlichkeitsrechnung mehrdimensionaler Zufallsgrößen lassen sich ähnlich wie bei eindimensionalen Zufallsgrößen Erwartungswerte und so genannte Kovarianzen angeben. Gegeben sei hierzu eine  $n$ -dimensionale Zufallsgröße  $\mathbf{X} = (X_1, X_2, \dots, X_n)^T$ . An die Stelle eines einzelnen Erwartungswerts tritt nun ein Vektor von  $n$  Erwartungswerten:

$$E(\mathbf{X}) = (E(X_1), E(X_2), \dots, E(X_n))^T, \quad (\text{A.1})$$

Als natürliche Erweiterung der Varianz des eindimensionalen Falles lassen sich nun jeweils  $n^2$  paarweise Kovarianzen  $cov(X_j, X_k)$

$$cov(X_j, X_k) = E[(X_j - E(X_j))(X_k - E(X_k))] \quad (\text{A.2})$$

$$= E(X_j X_k) - E(X_j)E(X_k) \quad j, k = 1, 2, \dots, n \quad (\text{A.3})$$

bestimmen. Nach [21] werden diese Größen auch als die ersten bzw. zweiten zentralen Momente der Zufallsgröße bezeichnet. Für den stetigen Fall einer mehrdimensionalen Zufallsverteilung  $f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$  ergeben sich die ersten Momente  $\alpha_j$  zu

$$\alpha_j = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} x_j f(x_1, \dots, x_n) dx_1 \cdots dx_n. \quad (\text{A.4})$$

Diese entsprechen den Erwartungswerten der Verteilung bezüglich der jeweiligen Richtungen. Die zweiten Momente  $\alpha_{jk}$  und zweiten zentralen Momente  $\mu_{jk}$  sind folgendermaßen definiert:

$$\alpha_{jk} = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} x_j x_k f(x_1, \dots, x_n) dx_1 \cdots dx_n. \quad (\text{A.5})$$

$$\mu_{jk} = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} (x_j - \alpha_j)(x_k - \alpha_k) f(x_1, \dots, x_n) dx_1 \cdots dx_n \quad (\text{A.6})$$

$$= \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} x_j x_k f(x_1, \dots, x_n) dx_1 \cdots dx_n - \alpha_j \alpha_k. \quad (\text{A.7})$$

## A.2 Berechnung der Kovarianzen einer triangulierten Oberfläche

In analoger Weise lassen sich Erwartungswerte und Kovarianzen auch für Punktmengen des dreidimensionalen Raumes angeben. Werden diese für die Oberfläche der konvexen Hülle  $H$  ermittelt, tritt an Stelle des unbegrenzten Integrals ein endliches Integral über die Hüllenfläche. Das erste Moment ergibt sich zu

$$\alpha_j^H = \frac{1}{A^H} \int_H x_j dA \quad j = 1, 2, 3. \quad (\text{A.8})$$

Da in diesem Falle nicht wie bei einer Zufallsverteilung das Integral über die Funktion  $\int f = 1$  ergibt, muss der Ausdruck noch mit der Fläche der Hülle  $A^H$  normiert werden. Das Moment erster Ordnung  $\alpha$ , welches den Vektor der „Erwartungswerte“ aller Punkte auf der Oberfläche widerspiegelt, entspricht dem Schwerpunkt der Hülle. Die zweiten zentralen Momente ergeben sich wiederum wie folgt

$$\mu_{jk}^H = \frac{1}{A^H} \int_H (x_j - \alpha_j^H)(x_k - \alpha_k^H) dA \quad (\text{A.9})$$

$$= \frac{1}{A^H} \int_H x_j x_k dA - \alpha_j^H \alpha_k^H \quad (\text{A.10})$$

$$= \alpha_{jk}^H - \alpha_j^H \alpha_k^H \quad j, k = 1, 2, 3. \quad (\text{A.11})$$

Nach einer Triangulierung ist die Oberfläche der konvexen Hülle als Dreiecksnetz gegeben. Obige Integrale ergeben sich nun durch Summation über alle Dreiecksflächen:

$$\alpha_j^H = \frac{1}{A^H} \sum_{i=1}^n \int_D x_j dA \quad j = 1, 2, 3, \quad (\text{A.12})$$

beziehungsweise

$$\mu_{jk}^H = \frac{1}{A^H} \sum_{i=1}^n \int_D x_j x_k dA - \alpha_j^H \alpha_k^H \quad j, k = 1, 2, 3. \quad (\text{A.13})$$

Die Integrale über die Dreiecksfläche lassen sich nun in Abhängigkeit der Eckknoten bestimmen. Hierzu führt man, wie in Abbildung A.1 dargestellt auf dem Dreieck ein neues Koordinatensystem ein.

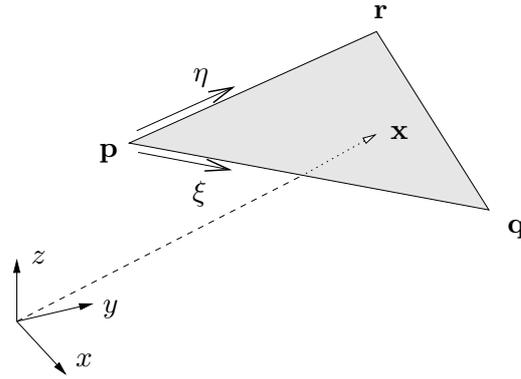


Abbildung A.1: Dreieck im Raum.

Damit lässt sich eine Transformation der lokalen Koordinaten  $\xi, \eta$  ins globale Koordinatensystem folgendermaßen angeben:

$$\mathbf{x} = \mathbf{p} + (\mathbf{q} - \mathbf{p})\xi + (\mathbf{r} - \mathbf{p})\eta \quad (\text{A.14})$$

Wird eine Integration über die Dreieckskoordinaten durchgeführt, muss zudem das Flächenelement  $dA$  berücksichtigt werden, welches sich wie folgt bestimmt [133]:

$$dA = \left| \frac{d\mathbf{x}}{d\xi} \times \frac{d\mathbf{x}}{d\eta} \right| = |(\mathbf{q} - \mathbf{p}) \times (\mathbf{r} - \mathbf{p})| = 2A_D \quad (\text{A.15})$$

Damit können nun die Dreiecksintegrale in (A.12) und (A.13) in Abhängigkeit der Eckpunkte ausgewertet werden. Für die drei Komponenten  $j = \{1, 2, 3\}$  der ersten Momente gilt:

$$\begin{aligned} \int_D x_j dA &= \int_{\eta}^1 \int_{\xi}^{1-\eta} x_j dA = 2A_D \int_{\eta}^1 \int_{\xi}^{1-\eta} x_j d\xi d\eta \\ &= 2A_D \int_{\eta}^1 \int_{\xi}^{1-\eta} [p_j + (q_j - p_j)\xi + (r_j - p_j)\eta] d\xi d\eta \\ &= \frac{1}{3}A_D(p_j + q_j + r_j), \quad j = 1, 2, 3, \end{aligned} \quad (\text{A.16})$$

und für die Kovarianzen:

$$\begin{aligned} \int_D x_j x_k dA &= \frac{1}{12}A_D ((p_j + q_j + r_j)(p_j + q_k + r_k) + p_j p_k + q_j q_k + r_j r_k) \\ &= \frac{1}{12}A_D (9s_j s_k + p_j p_k + q_j q_k + r_j r_k), \quad j, k = 1, 2, 3. \end{aligned} \quad (\text{A.17})$$

$s_j, s_k$  bezeichnen die Koordinaten des Schwerpunkts eines Dreiecks:

$$s_j = \frac{1}{3}(p_j + q_j + r_j), \quad j = 1, 2, 3. \quad (\text{A.18})$$

Basierend darauf lassen sich nun der Gesamtschwerpunkt aus Gleichung (A.12) der konvexen Hülle

$$\alpha_j^H = \frac{1}{A^H} \sum_{i=1}^n s_j^i A_D^i, \quad (\text{A.19})$$

sowie deren Kovarianzen (A.13) in einer geschlossenen Form nur in Abhängigkeit der Eckknoten aller  $n$  Dreiecke angeben:

$$\mu_{jk}^H = \frac{1}{12 A^H} \sum_{i=1}^n A_D^i (9 s_j^i s_k^i + p_j^i p_k^i + q_j^i q_k^i + r_j^i r_k^i) - \alpha_j^H \alpha_k^H. \quad (\text{A.20})$$

# Literaturverzeichnis

- [1] Airbus: *Airbus confirms further A380 delay and launches company restructuring plan*. press release, 10/3/2006, <http://www.airbus.com>, 2006.
- [2] P. R. Amestoy, I. S. Duff, and J.-Y. L'Excellent: *Multifrontal parallel distributed symmetric and unsymmetric solvers*. *Computer Methods in Applied Mechanics and Engineering*, 184:501–520, 2000.
- [3] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster: *A fully asynchronous multifrontal solver using distributed dynamic scheduling*. *SIAM Journal on Matrix Analysis & Applications*, 23(1):15–41, 2002.
- [4] P. R. Amestoy, I. S. Duff, S. Pralet, and C. Vömel: *Adapting a parallel sparse direct solver to architectures with clusters of SMPs*. *Parallel Computing*, 29:1645–1668, 2003.
- [5] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammarling, J. Demmel, C. Bischof, and D. Sorensen: *Lapack: A portable linear algebra library for high-performance computers*. In *Proceedings of Supercomputing '90*, 1990.
- [6] G. Aouad, A. Lee, and W. S.: *From 3D to nD modelling*. *ITcon*, 10:15–16, 2005.
- [7] C. Ashcraft and R. Grimes: *SPOOLES: an object oriented sparse matrix library*. In *Proceedings of the 1999 SIAM Conference on Parallel Processing for Scientific Computing*, 1999.
- [8] C. Ashcraft, D. Pierce, D. K. Wah, and J. Wu: *The Reference Manual for SPOOLES, Release 2.2: An Object Oriented Software Library for Solving Sparse Linear Systems of Equations*. Boeing Shared Services Group, 1999.
- [9] C. Ashcraft and J. W. H. Liu: *Using domain decomposition to find graph bisectors*. Technical Report 94-020, Boeing Computer Services, Seattle, USA, 1995.
- [10] C. Ashcraft and J. W. H. Liu: *Robust ordering of sparse matrices using multisection*. *SIAM Journal on Matrix Analysis & Applications*, 19(3):816–833, 1998.
- [11] M. Beall, J. Walsh, and M. Shephard: *Accessing CAD geometry for mesh generation*. In *12th International Meshing Roundtable*, Sandia National Laboratories, 2003.
- [12] K. Beuke: *Stand der Integration von Statik und Bemessung im Entwurfs- und Konstruktionsprozeß*. *Bauingenieur*, 71:67–72, 1996.

- [13] O. Beyer, H. Hackel, V. Pieper und J. Tiedge: *Wahrscheinlichkeitsrechnung und mathematische Statistik*. B.G. Teubner, Leipzig, 7. Auflage, 1995.
- [14] U. M. Borghoff and J. H. Schlichter: *Computer-Supported Cooperative Work - Introduction to Distributed Applications*. Springer, 2000.
- [15] A. Borrmann: *Computerunterstützung verteilt-kooperativer Bauplanung durch Integration interaktiver Simulationen und räumlicher Datenbanken*. Doktorarbeit, in Bearbeitung, Lehrstuhl für Bauinformatik, Technische Universität München, 2006.
- [16] A. Borrmann, C. van Treeck, and E. Rank: *Towards a 3D spatial query language for building information models*. In *Proceedings of the 11th International Conference on Computing in Civil and Building Engineering*, Montreal, Canada, 2006.
- [17] G. Braunberger: *Kabelsalat im A380: Die Flugzeugbauer von Airbus haben sich total verheddert*. Frankfurter Allgemeine Sonntagszeitung vom 08. Oktober 2006, 2006.
- [18] G. Breitzke, K.-R. Leimbach und W. Wunderlich: *SUBSTAR: ein Finite-Element-Programm mit Mehrebenen-Substrukturtechnik zur linearen statischen Berechnung von Tragwerken*. Technisch-wissenschaftliche Mitteilungen, Institut für Konstruktiven Ingenieurbau, Ruhr-Universität Bochum, 1987.
- [19] D. Bretschneider: *Modellierung rechnergestützter, kooperativer Arbeit in der Tragwerksplanung*. Doktorarbeit, Lehrstuhl für Ingenieurinformatik, Ruhr-Universität Bochum, Düsseldorf, 1998.
- [20] H. Bröker: *Integration von geometrischer Modellierung und Berechnung nach der p-Version der FEM*. Doktorarbeit, Lehrstuhl für Bauinformatik, Technische Universität München, 2001.
- [21] I. Bronstein, K. Semendjajew, G. Musiol und H. Mühlig: *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 6. Auflage, 2005.
- [22] H.-J. Bungartz, M. Griebel und C. Zenger: *Einführung in die Computergraphik*. Vieweg Verlag, 2. Auflage, 2002.
- [23] P. Cederqvist et al.: *Version Management with CVS (version 1.11.3)*. Free Software Foundation, <http://ximbiot.com/cvs/manual/>, 1993.
- [24] J. Corney and T. Lim: *3D Modeling with ACIS*. Saxe-Coburg Publications, 2001.
- [25] E. Cuthill and J. McKee: *Reducing the bandwidth of sparse symmetric matrices*. In *Proceedings of the 1969 24th ACM national conference*, pages 157–172, New York, USA, 1969. ACM Press.
- [26] L. Demkowicz: *Computing with hp-adaptive Finite Elements*. Chapman & Hall/CRC, 2006.
- [27] DIN 1045-1: *Tragwerke aus Beton, Stahlbeton und Spannbeton, Teil 1: Bemessung und Konstruktion*. DIN Deutsches Institut für Normung e.V., Berlin, Juli 2001.

- [28] DIN 1055-100: *Einwirkungen auf Tragwerke, Teil 100: Grundlagen der Tragwerksplanung, Sicherheitskonzept und Bemessungsregeln*. DIN Deutsches Institut für Normung e.V., Berlin, März 2001.
- [29] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon: *A min-max cut algorithm for graph partitioning and data clustering*. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, San José, USA, 2001.
- [30] J. Dongarra, J. Du Croz, S. Hammarling, and I. S. Duff: *A set of level 3 basic linear algebra subprograms*. *ACM Transactions on Mathematical Software*, 16(1):1–17, 1990.
- [31] R. O. Duda, P. E. Hart, and D. G. Stork: *Pattern classification*. John Wiley & Sons, 2001.
- [32] I. S. Duff: *Sparse numerical linear algebra: Direct methods and preconditioning*. Technical Report 19980137367, NASA, 1996.
- [33] I. S. Duff, A. M. Erisman, and J. K. Reid: *Direct Methods for Sparse Matrices*. Clarendon Press, Oxford, 1986.
- [34] I. S. Duff and J. K. Reid: *The multifrontal solution of indefinite sparse symmetric linear equations*. *ACM Transactions on Mathematical Software*, 9(3):302–325, 1983.
- [35] A. Düster: *High order finite elements for three-dimensional, thin-walled nonlinear continua*. Doktorarbeit, Lehrstuhl für Bauinformatik, Technische Universität München, 2001.
- [36] A. Düster, H. Bröker, H. Heidkamp, U. Heißenberger, S. Kollmannsberger, R. Krause, A. Muthler, A. Niggel, V. Nübel, M. Rücker, and D. Scholz: *AdhoC<sup>4</sup> – User’s Guide*. Lehrstuhl für Bauinformatik, Technische Universität München, 2004.
- [37] A. Düster, H. Bröker, and E. Rank: *The p-version of the finite element method for three-dimensional curved thin walled structures*. *International Journal for Numerical Methods in Engineering*, 52:673–703, 2001.
- [38] A. Düster and S. Hartmann: *High order anisotropic finite elements for three-dimensional isotropic hyperelastic continua*. In *Proceedings of the Second M.I.T Conference on Computational Fluid and Solid Mechanics*, Cambridge, USA, 2003.
- [39] A. Düster, S. Hartmann, and E. Rank: *p-fem applied to finite isotropic hyperelastic bodies*. *Computer Methods in Applied Mechanics and Engineering*, 192:5147–5166, 2003.
- [40] A. Düster, A. Niggel, V. Nübel, and E. Rank: *A numerical investigation of high-order finite elements for problems of elasto-plasticity*. *Journal of Scientific Computing*, 17:429–437, 2002.
- [41] A. Düster, A. Niggel, and E. Rank: *Thermo-elastic computations of geometrically nonlinear three-dimensional thin-walled continua based on high order finite elements*. In H. Mang, F. Rammerstorfer, and J. Eberhardsteiner (editors): *Proceedings of the Fifth World Congress on Computational Mechanics*, Vienna, Austria, 2002.

- [42] A. Düster and E. Rank: *A  $p$ -version finite element approach for two- and three-dimensional problems of the  $J_2$  flow theory with non-linear isotropic hardening*. International Journal for Numerical Methods in Engineering, 53:49–63, 2002.
- [43] A. Düster, D. Scholz, and E. Rank:  *$pq$ -Adaptive solid finite elements for three-dimensional plates and shells*. Submitted to Computer Methods in Applied Mechanics and Engineering, 2005.
- [44] A. Düster: *High order FEM*. Lecture Notes, Summer Term 2006, Lehrstuhl für Bauinformatik, Technische Universität München, 2006.
- [45] M. Dworschak: *Flieger aus dem Netz*. Der Spiegel, 33:156–158, 2005.
- [46] C. M. Eastman: *Building Product Models: Computer Environments Supporting Design and Construction*. CRC Press, 1999.
- [47] C. Ellis, S. Gibbs, and G. Rein: *Groupware: Some issues and experiences*. Communications of the ACM, 34(1):39–58, 1991.
- [48] R. Elmasri and S. B. Navathe: *Fundamentals of Database Systems*. Addison Wesley, 4. edition, 2004.
- [49] B. Firmenich: *Consistency of a shared versioned model for distributed cooperation*. Computer-Aided Civil and Infrastructure Engineering, 20(6):424–430, 2005.
- [50] G. Geist and E. Ng: *Task scheduling for parallel sparse cholesky factorization*. Int. Journal of Parallel Programming, 18(4):291–314, 1989.
- [51] A. George: *Nested dissection of a regular finite element mesh*. SIAM Journal on Numerical Analysis, 10(2):345–363, 1973.
- [52] A. George and J. W. Liu: *An automatic nested dissection algorithm for irregular finite element problems*. SIAM Journal on Numerical Analysis, 15(5):1053–1069, 1978.
- [53] W. Gordon and C. Hall: *Transfinite element methods: Blending function interpolation over arbitrary curved element domains*. Numerische Mathematik, 21:109–129, 1973.
- [54] S. Gottschalk, M. C. Lin, and D. Manocha: *OBBTree: a hierarchical structure for rapid interference detection*. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 171–180, New York, USA, 1996.
- [55] A. Gupta: *WSMP: Watson sparse matrix package, part I – direct solution of symmetric sparse systems*. Technical Report RC 21886, IBM Research Division, T.J. Watson Research Center, 2000.
- [56] A. Gupta: *Recent advances in direct methods for solving unsymmetric sparse systems of linear equations*. ACM Transactions on Mathematical Software, 28(3):301–324, 2002.

- [57] D. Hartmann, D. Bretschneider und U. Kolender: *Objektorientierte Strukturanalyse, Bemessung und konstruktive Durchbildung von Industriebauten unter besonderer Berücksichtigung parallel ablaufender Prozesse*. In: D. Hartmann (Herausgeber): *Objektorientierte Modellierung in Planung und Konstruktion, Forschungsbericht*, Seiten 219–246. Wiley-VCH, 2000.
- [58] Hauptverband der deutschen Bauindustrie e.V. (Herausgeber): *Zahlen und Fakten: Bau-datenkarte 05*. <http://www.bauindustrie.de>, 2005.
- [59] Hochtief AG: *ViCon – build digitally first: virtuelle Produktentwicklung für die Welt von morgen*. Produktinformation, <http://www.hochtief-vicon.de>, 2006.
- [60] A. Hörmann: *Ermittlung optimierter Stabwerkmodelle auf Basis des Kraftflusses als Ankopplung plattformunabhängiger Prozesskopplung*. Doktorarbeit, Lehrstuhl für Statik, Technische Universität München, 2006.
- [61] G. Hunter: *Efficient computation and data structures for graphics*. PhD thesis, Department of Electrical Engineering and Computer Science, Princeton University, 1978.
- [62] R. Hüttl: *Ein iteratives Lösungsverfahren bei der Finite-Elemente-Methode unter Verwendung von rekursiver Substrukturierung und hierarchischen Basen*. Doktorarbeit, Institut für Informatik, Technische Universität München, 1996.
- [63] J. Jönsson, S. Krenk, and L. Damkilde: *Recursive substructuring of finite elements*. *Computers & Structures*, 54(3):395–404, 1995.
- [64] G. Karypis: *CLUTO, a clustering toolkit, release 2.1.1*. Technical Report 02-017, University of Minnesota, Department of Computer Science, 2003.
- [65] G. Karypis and V. Kumar: *A fast and high quality multilevel scheme for partitioning irregular graphs*. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.
- [66] B. Kernighan and S. Lin: *An efficient heuristic procedure for partitioning graphs*. *Bell System Technical Journal*, 49(2):291–307, 1970.
- [67] D. E. Knuth: *The Art of Computer Programming*, volume 1: Fundamental Algorithms. Addison Wesley, 2. edition, 1973.
- [68] V. Korneev and U. Langer: *Domain decomposition methods and preconditioning*. In E. Stein, R. de Borst, and T. Hughes (editors): *Encyclopedia of Computational Mechanics*, volume 1, pages 59–137. John Wiley & Sons, 2004.
- [69] R. Krause and E. Rank: *A fast algorithm for point-location in a finite element mesh*. *Computing*, 57(1):49–63, 1996.
- [70] T. Liebich und T. Hoffeller (Herausgeber): *Anwenderhandbuch Datenaustausch BIM/IFC*. IAI - Industrieallianz für Interoperabilität, <http://www.buildingsmart.de>, 2006.
- [71] T. Liebich and J. Wix: *IFC Technical Guide - Release 2x*. IAI - International Alliance of Interoperability, 2000.

- [72] J. Mandel: *Iterative methods for p-version finite elements: preconditioning thin solids*. Computer Methods in Applied Mechanics and Engineering, 133:247–257, 1996.
- [73] M. Mäntylä: *An Introduction to Solid modeling*. Computer Science Press, 1988.
- [74] K. McKinney and M. Fischer: *Generating, evaluating and visualizing construction schedules with CAD tools*. Automation in Construction, 7:443–447, 1998.
- [75] U. Meißner, B. Möller, J. Diaz, U. Rüppel, F. Peters, V. Kirchner und M. Petersen: *Objektintegrierte Tragwerksmodelle für die Systemintegration von Planungs- und Konstruktionsvorgängen im Bauwesen*. In: D. Hartmann (Herausgeber): *Objektorientierte Modellierung in Planung und Konstruktion, Forschungsbericht*, Seiten 266–288. Wiley-VCH, 2000.
- [76] E. Mörsch: *Der Eisenbetonbau, seine Theorie und Anwendung*. Verlag Konrad Witwer, Stuttgart, 1912.
- [77] G. Morton: *A computer oriented geodetic data base and a new technique in file sequencing*. Technical report, IBM, Ltd., Ottawa, Kanada, 1966.
- [78] R.-P. Mundani and H.-J. Bungartz: *An octree-based framework for process integration in structural engineering*. In *Proceedings of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics*, volume II, pages 197–202. International Institute of Informatics and Systemics, 2004.
- [79] R.-P. Mundani: *Hierarchische Geometriemodelle zur Einbettung verteilter Simulationsaufgaben*. Doktorarbeit, Fakultät für Informatik, Elektrotechnik und Informationstechnik, Universität Stuttgart, 2005.
- [80] R.-P. Mundani, H.-J. Bungartz, A. Niggel, and E. Rank: *Embedding, organisation, and control of simulation processes in an octree-based CSCW framework*. In *Proceedings of the 11th International Conference on Computing in Civil and Building Engineering*, pages 3208–3215, Montreal, Canada, 2006.
- [81] R.-P. Mundani, H.-J. Bungartz, E. Rank, A. Niggel, and R. Romberg: *Extending the p-version of finite elements by an octree-based hierarchy*. In *Proceedings of the 16th Int. Conf. on Domain Decomposition Methods*, New York, USA, 2005.
- [82] R.-P. Mundani, I. Muntean, H.-J. Bungartz, A. Niggel, and E. Rank: *Applying grid techniques to an octree-based CSCW framework*. In *Proceedings of the 12th Europ. Parallel Virtual Machine and Message Passing Interface Conference*, pages 504–511, Sorrento, Italy, 2005.
- [83] A. Muthler: *Berechnung der elastischen Rückfederung von Tiefziehbauteilen mit der p-Version der Finite-Elemente-Methode*. Doktorarbeit, Lehrstuhl für Bauinformatik, Technische Universität München, 2001.
- [84] A. Niggel, E. Rank, R.-P. Mundani, and H.-J. Bungartz: *Organizing a p-version finite element computation by an octree-based hierarchy*. In P. Díez and N.-E. Wiberg (editors): *Proceedings of the Second International Conference on Adaptive Modeling and Simulation*, pages 26–35, Barcelona, Spain, 2005.

- [85] A. Niggel, E. Rank, R.-P. Mundani, and H.-J. Bungartz: *A framework for embedded structural simulation: Benefits in building design*. In *Proceedings of the 11th International Conference on Computing in Civil and Building Engineering*, pages 1768–1777, Montreal, Canada, 2006.
- [86] A. Niggel, R. Romberg, E. Rank, R.-P. Mundani, and H.-J. Bungartz: *A framework for concurrent structure analysis in building industry*. In *Proceedings of the 5th European Conference on Product and Process Modelling in the Building and Construction Industry*. A.A. Balkema Publishers, 2004.
- [87] A. Noor, A. Kamel, and R. Fulton: *Substructuring techniques - status and projections*. *Computers & Structures*, 2:621–632, 1978.
- [88] J. O'Rourke: *Finding minimal enclosing boxes*. *International Journal of Computer and Information Sciences*, 14(3):183–199, 1985.
- [89] J. O'Rourke: *Computational geometry in C*. Cambridge University Press, 2. edition, 1998.
- [90] T. Ottmann und P. Widmayer: *Algorithmen und Datenstrukturen*. Spektrum, Akademischer Verlag, 4. Auflage, 2002.
- [91] M. Paszyński, J. Kurtz und L. Demkowicz: *Parallel, fully automatic hp-adaptive 2d finite element package*. *Computer Methods in Applied Mechanics and Engineering*, 195:711–741, 2006.
- [92] J. Przemieniecki: *Matrix structural analysis of substructures*. Am. Instit. Aeronaut. Astronaut., 1(1):138–147, 1963.
- [93] J. Przemieniecki: *Theory of Matrix Structural Analysis*. McGraw-Hill, 1968.
- [94] E. Rank, H. Bröker, A. Düster, and V. Nübel: *High order solid elements for thin-walled structures: No tricks? - no crimes!* In W. Wall, K.-U. Bletzinger, and K. Schweizerhof (editors): *Trends in computational structural mechanics*, CIMNE, Barcelona, Spain, 2001.
- [95] E. Rank, H. Bröker, A. Düster und M. Rucker: *Integrierte Modellierungs- und Berechnungssoftware für den konstruktiven Ingenieurbau: Die p-Version und geometrische Elemente*. *Bauingenieur*, 76:53–61, 2001.
- [96] E. Rank, A. Düster, and Z. Alam: *Adaptive computation of thin-walled structures based on high order solid finite elements*. In *Proceedings of WCCM VI in conjunction with APCOM'04*, Beijing, China, 2004.
- [97] E. Rank, A. Düster, V. Nübel, K. Preusch, and O. Bruhns: *High order finite elements for shells*. *Computer Methods in Applied Mechanics and Engineering*, 194:2494–2512, 2005.
- [98] E. Rank, A. Halfmann, M. Rucker, C. Katz und S. Gebhard: *Integrierte Modellierungs- und Berechnungssoftware für den konstruktiven Ingenieurbau: Systemarchitektur und Netzgenerierung*. *Bauingenieur*, 75:60–66, 2000.

- [99] E. Rank, R. Romberg, A. Niggel, H.-J. Bungartz und R.-P. Mundani: *Volumenorientierte Modellierung als Grundlage einer vernetzt kooperativen Planung im konstruktiven Ingenieurbau*. In: U. Rüppel (Herausgeber): *Abschlussbericht zum DFG-Schwerpunktprogramm „Vernetzt-kooperative Planungsprozesse im Konstruktiven Ingenieurbau“*. Springer, erscheint 2007.
- [100] E. Rank: *Skriptum Bauinformatik I (CAD)*. Lehrstuhl für Bauinformatik, Technische Universität München, 2001.
- [101] K.-H. Reineck: *Der Schadensfall Slepner und die Folgerungen für den computergestützten Entwurf von Tragwerken aus Konstruktionsbeton*. In: E. Ramm, E. Stein und W. Wunderlich (Herausgeber): *Finite Elemente in der Baupraxis: Modellierung, Berechnung und Konstruktion*, Seiten 63–72. Ernst & Sohn, 1995.
- [102] K.-H. Reineck: *Modellierung der D-Bereiche von Fertigteilen*. In: *Beton-Kalender 2005*, Band 2, Seiten 243–296. Ernst & Sohn, 2001.
- [103] R. Romberg: *Gebäudemodell-basierte Strukturanalyse im Bauwesen*. Doktorarbeit, Lehrstuhl für Bauinformatik, Technische Universität München, 2005.
- [104] R. Romberg, A. Niggel, C. van Treeck, and E. Rank: *Structural analysis based on the product model standard IFC*. In *Proceedings of the 10th International Conference on Computing in Civil and Building Engineering*, Weimar, Germany, 2004.
- [105] M. Rücker: *A parallel p-version finite element approach for civil engineering and structural mechanics problems*. Doktorarbeit, Lehrstuhl für Bauinformatik, Bauingenieur- und Vermessungswesen, Technische Universität München, 2000.
- [106] K. Rückert: *Entwicklung eines CAD-Programmsystems zur Bemessung von Stahlbetontragwerken mit Stabwerkmodellen*. Doktorarbeit, Institut für Tragwerksentwurf und -konstruktion, Universität Stuttgart, 1992.
- [107] U. Rüppel und T. Klauer: *Eine prozessorientierte Kooperationsplattform für Bauprojekte*. *Bauingenieur*, 80:71–78, 2005.
- [108] U. Rüppel und U. Meißner: *Integrierte Planung, Fertigung und Nutzung von Bauwerken auf der Basis von Produktmodellen*. *Bauingenieur*, 71:47–55, 1996.
- [109] H. Samet: *The quadtree and related hierarchical data structures*. *ACM Computing Surveys*, 16(2):187–260, 1984.
- [110] H. Samet: *Spatial data structures*. In W. Kim (editor): *Modern Database Systems: The Object Model, Interoperability, and Beyond*, pages 361–385. ACM Press, Addison Wesley, 1995.
- [111] J. Schlaich und K. Schäfer: *Konstruieren im Stahlbetonbau*. In: *Beton-Kalender 1984*, Band 2, Seiten 787–1005. Ernst & Sohn, 1984.
- [112] J. Schlaich und K. Schäfer: *Konstruieren im Stahlbetonbau*. In: *Beton-Kalender 2001*, Band 2, Seiten 311–492. Ernst & Sohn, 2001.

- [113] M. Schleinkofer, T. Schäfer, C. van Treeck und E. Rank: *Vom Laserscan zum planungstauglichen Produktmodell*. In: K. von Gürebeck und C. Könke (Herausgeber): *Internationales Kolloquium ber Anwendungen der Informatik und Mathematik in Architektur und Bauwesen (IKM 2006)*, Weimar, 2006.
- [114] M. Schneider: *Verteilte adaptive numerische Simulation auf der Basis der Finite-Elemente-Methode*. Doktorarbeit, Institut für Informatik, Technische Universität München, 1995.
- [115] W. Schnell, D. Gross und W. Hauger: *Technische Mechanik, Band 2: Elastostatik*. Springer-Verlag, 5. Auflage, 1995.
- [116] D. Scholz: *An anisotropic  $p$ -adaptive method for linear elastostatic and elastodynamic analysis of thin-walled and massive structures*. Doktorarbeit, Lehrstuhl für Bauinformatik, Technische Universität München, 2006.
- [117] H.-R. Schwarz und N. Köckler: *Numerische Mathematik*. B.G. Teubner, 5 Auflage, 2004.
- [118] V. Shapiro: *Solid modeling*. In G. Farin, J. Hoschek, and M.-S. Kim (editors): *Handbook of Computer Aided Geometric Design*, pages 473–518. Elsevier Science, 2002.
- [119] B. Stroustrup: *The C++ Programming Language, Special Edition*. Addison Wesley, 2000.
- [120] B. Szabó and I. Babuška: *Finite element analysis*. John Wiley & Sons, 1991.
- [121] B. Szabó, A. Düster, and E. Rank: *The  $p$ -version of the Finite Element Method*. In E. Stein, R. de Borst, and T. Hughes (editors): *Encyclopedia of Computational Mechanics*, volume 1, chapter 5, pages 119–139. John Wiley & Sons, 2004.
- [122] B. Szabó and G. Sahrman: *Hierarchic plate and shell models based on  $p$ -extension*. International Journal for Numerical Methods in Engineering, 26:1855–1881, 1988.
- [123] R. A. Toupin: *Saint-Venant's Principle*. Archive for Rational Mechanics and Analysis, 18(2):83–96, 1965.
- [124] C. van Treeck: *Gebäudemodell-basierte Simulation von Raumluftströmungen*. Doktorarbeit, Lehrstuhl für Bauinformatik, Technische Universität München, 2004.
- [125] C. van Treeck and E. Rank: *Analysis of building structure and topology based on graph theory*. In *Proceedings of the 10th International Conference on Computing in Civil and Building Engineering*, Weimar, Germany, 2004.
- [126] C. van Treeck and E. Rank: *Dimensional reduction of 3D building models using graph theory and its application in building energy simulation*. Engineering with Computers, to appear, 2006.
- [127] C. van Treeck, R. Romberg, and E. Rank: *Simulation based on the product model standard IFC*. In *Proc. 8th Int. IBPSA Conf. Building Simulation, Eindhoven, Netherlands*, 2003.

- [128] C. van Treeck, R. Romberg und E. Rank: *Graphentheoretische Analyse von Produktmodellaten hinsichtlich Raumtopologie und Gebäudestruktur*. In: *Festschrift 60. Geb. Prof. Hartmann, VDI-Tagungsband*, 2004.
- [129] W. Wall, M. Gee, and E. Ramm: *The challenge of a three-dimensional shell formulation, the conditioning problem*. In *Proceedings of ECCM '99, European Conference on Computational Mechanics*, München, Germany, 1999.
- [130] H. Willenbacher: *Interaktive verknüpfungsbasierte Bauwerksmodellierung als Integrationsplattform für den Bauwerkslebenszyklus*. Doktorarbeit, Fakultät Bauingenieurwesen, Bauhaus-Universität Weimar, 2002.
- [131] F. W. Williams: *Comparison between sparse stiffness matrix and sub-structure methods*. *International Journal for Numerical Methods in Engineering*, 5:383–394, 1973.
- [132] J.-D. Wörner, J. Reymendt, S. Gunnarson und R. Stürmer: *Objektorientierte Integration von Teilprozessen im Bauwesen mit Hilfe einer objektorientierten Datenbank für den Bereich der Bemessung und Konstruktion von Hochbauteilen im Massivbau*. In: D. Hartmann (Herausgeber): *Objektorientierte Modellierung in Planung und Konstruktion, Forschungsbericht*, Seiten 321–345. Wiley-VCH, 2000.
- [133] O. Zienkiewicz and R. Taylor: *The Finite Element Method – The Basis*, volume 1. Butterworth-Heinemann, 5th edition, 2000.
- [134] K. Zilch und A. Rogge: *Grundlagen der Bemessung von Beton-, Stahlbeton- und Spannbetonbauteilen nach DIN 1045-1*. In: *Beton-Kalender 2002*, Band 2, Seiten 217–359. Ernst & Sohn, 2002.
- [135] K. Zilch und G. Zehetmaier: *Bemessung im konstruktiven Betonbau*. Springer-Verlag, 2006.