

Lehrstuhl für Bauinformatik
Fakultät Bauingenieur- und Vermessungswesen
Technische Universität München

Gebäudemodell-basierte Strukturanalyse im Bauwesen

Richard Romberg

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. Matthäus Schilcher
Prüfer der Dissertation: 1. Univ.-Prof. Dr. rer.nat. Ernst Rank
2. Univ.-Prof. Dr.-Ing. Karl Beucke
Bauhaus-Universität Weimar

Die Dissertation wurde am 11.01.2005 bei der Technischen Universität München eingereicht und durch die Fakultät für Bauingenieur- und Vermessungswesen am 23.05.2005 angenommen.

für
Alexandra, Lena und Elisa

für meine Mutter Inge Romberg
† 16. Mai 2004

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Bauinformatik der Technischen Universität München.

Viele Personen haben zum Gelingen dieser Arbeit beigetragen. An dieser Stelle möchte ich ihnen meinen Dank aussprechen. Mein besonderer Dank gilt Herrn Prof. Ernst Rank, der diese Arbeit betreut hat. Er hat mit wertvollen Tipps und fruchtbaren Diskussionen erheblich zum Gelingen der Arbeit beigetragen. Herrn Prof. Karl Beucke danke ich für das Interesse an dieser Arbeit und für die Übernahmen des Korreferats.

Anerkennung und Dank spreche ich hiermit allen meinen Kollegen des Instituts für Bauinformatik aus, die mich bei meiner Arbeit durch Diskussion und Ermutigung in fachlicher und menschlicher Hinsicht unterstützt haben. Besonderer Dank gilt meinen beiden Zimmernachbarn Dr. Jonas Tölke und Alexander Muthler für die interessanten Gespräche und die gute Zeit im gemeinsamen Büro.

Schließlich danke ich allen, die indirekt zum Gelingen dieser Arbeit beigetragen haben. Allen voran meinen Eltern Inge und Peter und meiner Frau Alexandra, für ihre stetige Unterstützung und den großen Rückhalt, den sie mir zu jeder Zeit entgegengebracht haben.

Kurzfassung

In der vorliegenden Arbeit wird ein neu entwickeltes Verfahren zur Unterstützung der Tragwerksplanung vorgestellt. Auf der Grundlage eines Bauwerksmodells wird untersucht, wie die Bearbeitungsschritte zur Erstellung des numerischen Modells bei der Strukturanalyse ohne Bruch der digitalen Modellkette automatisch durchgeführt werden können und die Integration von numerischer Berechnung in Bauplanungsabläufen dadurch effektiver gestaltet wird. Vorteile sind insbesondere deshalb zu erwarten, weil zur Analyse aller Strukturen strikt finite Volumenelemente eingesetzt werden und dadurch die problematische Verwendung von dimensionsreduzierten Teilmodellen entfällt. In Verbindung mit hierarchischen Elementansätzen höherer Polynomordnung erlaubt der damit vollzogene Übergang zur ganzheitlichen Berechnung von Gebäudestrukturen (vollständige 3D-Berechnung) eine genauere Darstellung des Tragverhaltens mit qualitativ verbesserten Analyseergebnissen.

Abstract

This thesis presents a new procedure that has been devised to support structural engineering. Using a building model as a basis, we investigate how the processing steps required to create the numerical model for structural analysis can be accomplished automatically without interrupting the digital model chain while providing new impulses for the integration of numerical computation in the building design workflow. Our confidence in achieving a beneficial advancement is particularly high since strictly three-dimensional finite elements are used throughout, which accordingly dispense with the problematic use of dimensionally reduced partial models. In conjunction with polynomial, higher-order hierarchical element approaches, this completes the transition to the holistic computation of building structures (pure 3D calculation) and allows a more accurate representation of the static behaviour of the structures with qualitatively upgraded analytical results.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Einführung und Motivation	1
1.2	Softwaretechnischer Rahmen	4
1.3	Gliederung der Arbeit	4
2	Aspekte der numerischen Simulation.....	6
2.1	Allgemeines.....	6
2.2	Grundlagen der geometrischen Modellierung	9
2.2.1	Einführung.....	9
2.2.2	Methoden der Volumenmodellierung	9
2.2.3	Normzellen-Aufzählungsschema	11
2.2.4	Oktalbaum-Schema	12
2.2.5	Constructive Solid Geometry (CSG)	13
2.2.6	Boundary Representation (BRep)	14
2.2.7	Produktionsmodelle (Sweeping-Modelle)	16
2.3	Diskretisierung der Geometrie	18
2.3.1	Allgemeines.....	18
2.3.2	Methode der Gebietsteilung mit uniformer Elementkantenlänge.....	19
2.3.3	Konvertierung des Dreiecksnetzes	22
2.3.4	Adaptive Netzerzeugung	24
2.4	Strukturanalyse mit der p-Version der FEM.....	25
2.4.1	Einführung.....	25
2.4.2	Finite Elemente mit hoher Ordnung.....	26
3	Tragwerksplanung am dreidimensionalen Modell.....	29
3.1	Produktmodellierung im Bauwesen	29
3.1.1	Einführung.....	29
3.1.2	Der Produktmodell-Standard STEP – ISO 10303	30
3.1.3	Der Produktmodell-Standard IFC	31
3.1.4	IFC Erweiterung für die Tragwerksplanung (ST-4).....	34

3.2	Überführung von Produktmodell­daten in ein erweitertes Volumenmodell.....	35
3.2.1	Einführung.....	35
3.2.2	Der geometrische Modellierkern ACIS.....	38
3.2.3	Einlesen der IFC-Produktmodell­daten	42
3.2.4	Auswerten geometrischer/semantischer Modell­daten.....	44
3.3	Analyse und Aufbereitung der Gebäudestruktur	58
3.3.1	Einführung.....	58
3.3.2	Dekomposition des geometrischen Modells.....	61
3.3.3	Details zur Implementierung.....	66
3.4	Automatische makrobasierte Hexaeder Netzgenerierung.....	73
3.4.1	Einführung.....	73
3.4.2	Teilschritte der Hexaeder-Vernetzung	75
3.4.3	Differenzobjekte in Dickenrichtung vernetzen	85
4	Modellbeispiel.....	87
5	Zusammenfassung und Ausblick.....	95
5.1	Zusammenfassung.....	95
5.2	Ausblick	96

1 Einleitung

1.1 Einführung und Motivation

Als Ergebnis einer rapiden Fortentwicklung der Rechnertechnik und der allgemeinen Verbreitung leistungsfähiger Workstations und Personalcomputer sind heute Simulationsverfahren auch für komplexe technische Problemstellungen möglich und erleben im industriellen Alltag eine stetig wachsende Akzeptanz und Anwendung. Bei der Simulation ist die Vorgehensweise folgende: Ein realer Sachverhalt (z.B. Tragverhalten eines statischen Systems) wird mit einem mathematischen Modell beschrieben, wobei die Zusammenhänge zwischen Ursache (Einflussgrößen, z.B. Lasten) und Wirkung (Reaktionsgrößen, z.B. Verschiebungen, Spannungen) über Modellgleichungen beschrieben werden. Die am abstrakten Modell gefundene Lösung wird anschließend interpretiert und auf den realen Sachverhalt zurück übertragen.

Die numerische Simulation tritt nicht als losgelöste Disziplin auf, sondern ist Teil der übergreifenden Strategie, den Rechner als Hilfsmittel zum ‚schnellen‘ Erkenntnisgewinn einzusetzen. Betrachtet man aktuelle Entwicklungen, so geht der Trend dahin, die gesamte Produktentwicklung weitestgehend zu digitalisieren und den Lebenszyklus eines Produktes von der Planung bis zur Entsorgung ohne Bruch der digitalen Prozesskette abzubilden [Abramovici 02]. Besonders weit fortgeschritten ist diese Entwicklung des *Product Data Modeling (PDM)* im Maschinenbau und dort besonders im Automobil- und Flugzeugbau, wo 3D-Modelle Stand der Technik sind. Weiterhin werden Digital-Mockup-Systeme als Ersatz für physikalische Prototypen für Designstudien und Montageuntersuchungen herangezogen.

Im Bauwesen ist die Anwendung von PDM-Systemen aufgrund der nachfolgend genannten Gründe weniger weit fortgeschritten:

- Bauwerke sind i.d.R. Unikate. Planung und Baudurchführung werden oftmals nur für ein einziges Gebäude vorgenommen.
- Projekte unterliegen einer wachsenden Komplexität und einem zunehmenden Detaillierungsgrad mit dem Fortschreiten der Planung bzw. Realisierung.
- Unabhängige Fachplaner bearbeiten Projekte räumlich und zeitlich getrennt voneinander (oftmals keine übergeordnete Projektsteuerung).
- Verschiedene Fachplaner sehen das Bauwerksmodell aus ihrer fachspezifischen Perspektive (Views) mit unterschiedlichem semantischen Gehalt und entsprechend unterschiedlicher Abstraktion.

- Im Bauwesen existiert durch Spezialisierung der Fachapplikationen ein stark heterogenes Softwareumfeld.
- Bauwerke haben i.d.R. im Vergleich mit Softwareprodukten einen sehr langen Lebenszyklus. Daraus ergibt sich für die Zukunft potentiell eine Zugriffproblematik auf alte Datenbestände.
- Bauwerke sind oftmals stark wechselnden äußeren Randbedingungen unterworfen (Umplanung bei unerwarteten Änderungen der Baugrundeigenschaften, Änderung der Belastung infolge geänderter Nutzung).

Die genannten Punkte machen es schwierig, für ein Bauwerk zum Planungszeitpunkt wichtige Zustände oder Ereignisse über den gesamten Lebenszyklus vorhersagen zu können. Auch existieren in der Praxis Simulationsmodelle meist nur innerhalb begrenzter Anwendungsbereiche für ganz bestimmte Aufgaben. Dabei geschieht der Austausch von Daten zum Planungsgegenstand oft unter Verlust an Informationen oder wird sogar zeitaufwendig per Hand durchgeführt. Eine durchgängige Unterstützung der Planung durch rechnerbasierte Planungswerkzeuge ist somit nur ansatzweise erkennbar. Integrierte Anwendungen zur fachübergreifenden Unterstützung der Planungsbeteiligten basieren vielmehr auf ausgewählten proprietären Systemen bei Bauvorhaben mit zentralen Planungsstrukturen (Generalunternehmen). Darüber hinaus verlangen gestiegene Anforderungen (z.B. Kostenstruktur, Termintreue, Einsatz neuer Technologien, etc.) von den Beteiligten bei der Bauwerksplanung und Durchführung zunehmend eine Arbeitsteilung durch vernetzte Gruppen. Dies führt zu veränderten Arbeitsweisen, die durch ein gestiegenes Kommunikationsaufkommen zwischen den Arbeitsgruppen gekennzeichnet sind und die zur Folge haben, dass die Digitalisierung der Planung von Bauwerken durch entsprechende Unterstützung durch Computersysteme zur parallelen Bearbeitung (concurrent engineering) immer wichtiger wird. Der Trend führt dazu, dass auch im Bauwesen Datenmodelle zunehmend zentrale Bedeutung im Entwicklungszyklus bekommen. Information muss schnell und möglichst überall verfügbar sein [Hauschild 02].

Die Produktdatentechnologie adressiert diese Problematik, indem Methoden und Techniken für einen effizienten Daten- und Informationsaustausch, zur Sicherstellung der Kommunikation und Kooperation unter den Beteiligten und zur Verbesserung der Integration von Prozessen zu Prozessketten entwickelt werden. Ein Beispiel dafür stellen die Bemühungen zur direkten Übernahme von Informationen aus der Konstruktion in die Produktion bei Bewehrungsverfahren im Massivbau dar [Bamtec 04]. Die Integration über einheitliche Datenmodelle ermöglicht die Automatisierung von Routinearbeitsschritten wie z.B. der redundanten Eingabe von Planungsdaten und resultiert in der Steigerung der Produktivität der gesamten Prozesskette.

Im Kontext dieser Prozesskette zeichnen sich die einzelnen Software-Systeme dadurch aus, dass sie auf gemeinsame genutzte Daten zugreifen, so dass diese Daten nicht mehr einem System zuzuordnen sind, sondern in ihrer Gesamtheit die relevanten Merkmale eines Bauwerks repräsentieren. Daraus folgt, dass die Integration der Software-Systeme die Integration der benötigten Daten voraussetzt. In diesem Zusammenhang besteht eine wesentliche Aufgabe bei der Entwicklung von Produktdatenmodellen in der Analyse von Prozessketten mit dem Ziel, relevante, d.h. gemeinsam nutzbare Daten und Informationen über Produkte und Prozesse zu identifizieren, in einer geeigneten Form zu formalisieren (im informationstechnischen Sinne) und schließlich zu einem konsistenten Modell zu integrieren. Richtungsweisend ist hierbei die Norm ISO 10303.

Sie ist die normative Basis zweier wichtiger Produktmodell-Spezifikationen mit industriellen Bedeutung: STEP¹ und IFC². STEP wird hauptsächlich im Automobil- und Flugzeugbau eingesetzt, während im Bauwesen die IFC eine große Verbreitung erlangt hat.

In einem gemeinsamen Schwerpunktprogramm³ diverser deutscher Forschungsinstitute aus dem Bereich Bauingenieurwesen wird seit dem Jahr 2000 untersucht, welche Methoden und Kooperationsmodellen mit disziplinübergreifender Semantik für die rechnergestützte Bauplanung und Baudurchführung geeignet sind, um eine Steigerung der Effizienz der arbeitsteiligen Projektbearbeitung bei gleichzeitigen Qualitätsverbesserungen zu erreichen. Insbesondere werden dabei folgende Ziele verfolgt:

- Erarbeitung methodischer Grundlagen zur Unterstützung kooperativ ausgerichteter Arbeits- und Organisationsstrukturen.
- Untersuchung der Bearbeitungsprozesse im Informations- und Kommunikationsverbund zur Neugestaltung teamorientierter Planungs-Szenarien.
- Erhöhung der Transparenz und der Entscheidungssicherheit im Bauplanungs- und Herstellungsprozess.

Ziel dieser Arbeit ist es nun, ein Konzept zur Neugestaltung der rechnergestützten Kooperation zwischen Entwurfsverfasser und Tragwerksplaner vorzulegen. Wesentliche Neuerung wird dabei die direkte Anbindung der Strukturanalyse an das Architekturmodell sein, durch die die bisherige Problematik der unterschiedlichen Gebäudesemantiken in beiden Disziplinen aufgelöst wird, indem mittels einer durchgängig dreidimensionalen Beschreibung und Bearbeitung aller Objekte die schwierige und fehlerträchtige Reduzierung der Dimension bei der Tragwerksplanung hinfällig wird. Dies soll in einer Art und Weise verwirklicht werden, die den Benutzer bei der Erstellung des numerischen Modells unterstützt, indem weite Teile des Transferprozesses automatisch ablaufen. Mit dieser Unterstützung soll es dem Tragwerksplaner möglich werden, sein Hauptaugenmerk auf den Entwurf des Tragwerks zu legen. Die integrierte Verwaltung der Tragwerksdaten soll den Planer zudem in die Lage versetzen, eine Vielzahl an Berechnung mit unterschiedlichen Parametern durchzuführen und damit Optimierungsprozesse, Parameterstudien oder in frühen Phasen der Planung schnell Vorbemessungen durchführen zu können, die sonst aufgrund des zeitlichen Aufwandes nicht möglich wären.

¹ **ST**andard for the **EX**change of **P**roduct **M**odel **D**ata

² **I**ndustrie **F**oundation **C**lasses.

³ Das Schwerpunktprogramm ‚*Vernetzt-kooperative Planungsprozesse im Konstruktiven Ingenieurbau*‘ wird von der Deutschen Forschungsgemeinschaft (DFG) finanziell unterstützt. Nähere Informationen sind [DFG 04] zu entnehmen.

1.2 Softwaretechnischer Rahmen

Zum Nachweis der Funktionsfähigkeit wurde eine prototypische Implementierung vorgenommen, in deren Rahmen alle wesentlichen Bearbeitungsschritte durchgeführt werden. Durch Voruntersuchungen wurde herausgefunden, dass ein abgeschlossenes CAD-System ein zu starrer Rahmen für die Implementierung ist. Es wurde deshalb eine Software-Umgebung aus leistungsfähigen Einzelkomponenten zusammengestellt, die aus der professionellen Anwendung kommen und jedes für sich herausragende, weil spezialisierte Techniken mitbringt.

- Zur Bearbeitung von geometrischen Modellen wurde ACIS von Spatial Inc. ausgewählt. Das ist eine in der Sprache C++ geschriebene Bibliothek, die die Beschreibung geometrischer Objekte über eine BRep-Struktur vornimmt [ACIS 02].
- Zur Visualisierung wurde die Bibliothek HOOPS ausgewählt. Wesentlich für die Performance bei der Darstellung ist hierbei die Verwendung eines Szenegraphen. Die Bibliothek nutzt intern zur visuellen Darstellung die OpenGL Bibliothek [HOOPS 02].
- Den Applikations-Rahmen bildet die Bibliothek Qt von Trolltech. Qt ist eine plattformunabhängiges C++ GUI⁴-Framework. Es bietet dem Entwickler die gesamte Funktionalität zur Generierung von ‚*state-of-the-art*‘ Applikationen mit graphischem Benutzerinterface. Qt ist vollständig objektorientiert, einfach zu erweitern und unterstützt die komponentenbasierte Softwareentwicklung. Neben den Klassen zur GUI-Entwicklung beinhaltet die Bibliothek zahlreiche weitere nützliche Klassen (Netzwerkkommunikation, Datenbank, XML, etc.) [QT 04].

Die genannten Bibliotheken sind in C++ entwickelt und plattformübergreifend verfügbar und wurden in dieser Arbeit mittels einer IDE⁵, auf der Plattform WINDOWS in einer Applikation integriert. Dieser Rahmen ist nun geeignet, die Manipulationen am geometrischen Modell bzw. Teilmodellen effizient durchzuführen und zu visualisieren.

1.3 Gliederung der Arbeit

In Kapitel 2 werden Grundlagen zu Methoden und Technologien im Rahmen der hier eingesetzten numerischen Simulation umrissen. Diese lassen sich in drei wesentliche Teile gliedern: Geometrie, Diskretisierung und numerisches Rechenverfahren. Zuerst werden geometrische Modelle und deren spezifische Eigenschaften vorgestellt, die beim Zugriff auf Bauwerksmodellldaten eine Rolle spielen. Es folgt die Beschreibung der Methode zur Diskretisierung der Geometrie, die innerhalb des Näherungsverfahrens zur Lösung von partiellen Differenzialgleichungen genutzt wird. Die Art der Diskretisierung ist eng verbunden mit der mathematischen Modellbildung, da die Genauigkeit der

⁴ Graphical-User-Interface

⁵ Integrated Development Environment. Hierfür wurde das Visual Studio von Microsoft eingesetzt.

Lösung einerseits mit einer feineren Auflösung der Geometrie (h -Methode) oder durch eine Erhöhung des polynominalen Verschiebungsansatzes (p -Methode) erreicht wird. Im letzten Teil von Kapitel 2 werden die Grundzüge der Finite-Element-Methode (FEM) mit den wesentlichen mathematischen Grundlagen vorgestellt.

Kapitel 3 bildet den zentralen Teil dieser Arbeit. Dabei geht es zunächst um Techniken zur Produktmodellierung im Bauwesen. Nach der Beschreibung der Problematik der Überführung von Bauwerksmodellldaten in eine für die numerische Simulation geeignete Form folgt die Analyse der Gebäudestruktur. Die Kenntnis über die Bauteilstruktur ist Voraussetzung für eine Zerlegung des geometrischen Modells in sog. Separator-Objekte an Kopplungsbereichen von Bauteilen. Dieser Zerlegungsprozess wird detailliert beschrieben. Die Zerlegung des geometrischen Modells bildet die Grundlage für die im nachfolgenden Unterkapitel beschriebene automatische Hexaeder-Netzgenerierung. Darin wird ein Verfahren vorgestellt, durch eine Kombination von 3D-Vernetzungsmakros und 2D-Freivernetzung eine geeignete Struktur automatisch in Hexaeder zu diskretisieren.

Kapitel 4 beschreibt anhand eines komplexen Beispiels die Leistungsfähigkeit des Verfahrens. Dabei wird ein numerisches Modell mit entsprechenden Randbedingungen automatisch aus einem Bauwerksmodell abgeleitet.

Im Kapitel 5 werden die Ergebnisse dieser Arbeit zusammengefasst und Vorschläge für weitere Forschungsarbeiten gemacht.

2 Aspekte der numerischen Simulation

2.1 Allgemeines

Bei der numerischen Berechnung von strukturmechanischen Systemen unterscheidet man meist zwischen verschiedenen funktionalen Teilprozessen:

- **Preprocessing:** Ausgangspunkt für die Simulation ist die geometrische Beschreibung des zu untersuchenden Gebiets. Das geometrische Modell wird mit Randbedingungen angereichert (Problemdefinition) und in eine diskrete Form überführt. Die Diskretisierung ist dabei abhängig von dem zu simulierenden physikalischen Phänomen und dem verwendeten mathematischen Modell. Mit Zunahme der Leistungsfähigkeit der Rechner werden ebenfalls die Simulationsmodelle größer und damit wird es für den Planer zunehmend schwieriger, das Modell vollständig zu überblicken. Deshalb erlangen Techniken, die die Richtigkeit bzw. Konsistenz der Modelle prüfen, eine immer größere Bedeutung. Beispielsweise ist es für ein Bauwerksmodell nicht immer ohne weiteres möglich festzustellen, ob zwischen Bauteilen infolge fehlerhafter Modellierung Klaffungen⁶ auftreten.
- **Berechnung:** Dabei werden zunächst bestimmte Vorarbeiten durchgeführt (z.B. Berechnung der Elementsteifigkeiten) um schließlich die Lösung der Modellgleichungen zu ermitteln. Die Ergebnisse der Berechnung spiegeln die simulierte Realität niemals exakt wieder. Vielmehr werden Modellfehler (Realität \Leftrightarrow mathematisches Modell) und Diskretisierungsfehler in Kauf genommen, die es im Rahmen der Simulation zu minimieren gilt.

⁶ Eine Lösung für diese Problematik wird von MUNDANI mittels einem raumpartitionierendem Verfahren vorgestellt [Mundani 03].

- **Postprocessing:** Hierbei wird das Ergebnis der Berechnung analysiert. Die Analyse wird entweder über eine Ergebnis-Visualisierung vorgenommen oder über eine anwendungsinterne Weiterverarbeitung. Bei sehr großen Modellen spielen Techniken zur Reduktion und Aufbereitung von Ergebnisdaten eine wichtige Rolle. Dabei wird versucht, das Datenaufkommen mit möglichst geringer Einschränkung der Aussagekraft derart zu reduzieren, dass die Datenmenge für eine Weiterverarbeitung geeignet ist (hinsichtlich Netzwerkbelastung, Visualisierung, etc.). Von KÜHNER werden in [Kühner 03] Reduktionstechniken vorgestellt.

Grundlegendes Ziel bei der Einbettung der numerischen Simulation in die Prozesskette der Bauplanung ist es, Transformationen über Modellgrenzen hinweg mit einem Maximum an Flexibilität möglichst automatisch durchzuführen. Infolge der unterschiedlichen Anforderungen der oben genannten funktionalen Teilprozesse kommt es aber oftmals dazu, dass unterschiedliche Datenstrukturen verwendet werden und dass unterschiedlich spezialisierte Hardware in verteilten Rechnerumgebungen zum Einsatz kommt. Die Integration der verschiedenen Modelle in einen Gesamtprozess erfordert deshalb oftmals Kopplungen durch geeignete Schnittstellen, die zudem gewährleisten, dass in den Teilbereichen auf möglichst wenig Funktionalität und Leistung verzichtet werden muss (siehe dazu [Frank 00] und weiterführende Quellen).

Modelldesign

Ein möglicher Lösungsansatz liegt darin, dem Bauwerksmodell mit allen geometrischen, topologischen sowie semantischen Daten auch im numerischen Simulationsverfahren eine zentrale Bedeutung zukommen zu lassen. Damit ist eine neutrale Beschreibung des Gebäudes oder des Bauteils verfügbar, von dem sich entsprechend der jeweiligen Aufgabe nachfolgende mathematische Modelle ableiten lassen. Das in Abbildung 1 gezeigte Schaubild verdeutlicht zwei Modellketten im Rahmen einer bauwerksmodellbasierten numerischen Simulation. Neben der in dieser Arbeit vorgestellten Modellkette ‚Strukturanalyse‘, wird die Modellkette ‚thermisches Mehrzonenmodell‘ in [Treeck 04b] detailliert beschrieben. Ein wesentlicher Teil der softwaretechnischen Umsetzung konnte mit diesem Ansatz für beide Modellketten über eine gemeinsame Codebasis realisiert werden.

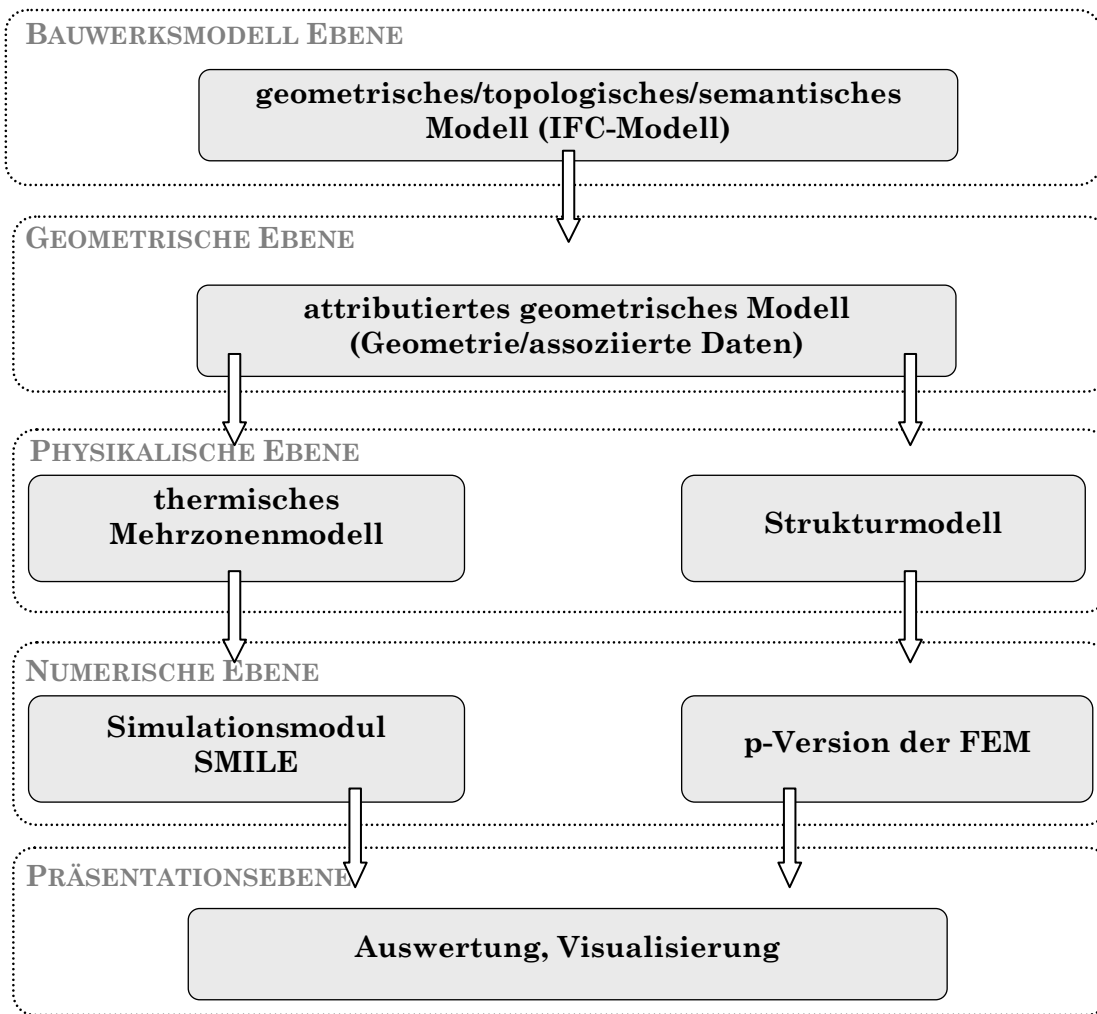


Abbildung 1: Konzept der Modellkette zur numerischen Simulation basierend auf einem Bauwerksmodell.

2.2 Grundlagen der geometrischen Modellierung

2.2.1 Einführung

Geometrische Modellierung ist ein wichtiger Aspekt des Planungs- und Produktionsprozesses von Produkten (z.B. Bauwerken). Die dazu in diesem Abschnitt erläuterten Grundlagen beleuchten überblicksartig verschiedene Modelle, die im Bauwesen eine Rolle spielen. Geometrische Modellierung wird dabei als *Sammelbegriff* für die unterschiedlichsten Anwendungsgebiete verwendet. Das sind im 2D-Bereich hauptsächlich vektorbasierte Grafikanwendungen (zeichnungsorientierter Ansatz im Bauwesen). Im 3D-Bereich kommt dagegen eine größere Anzahl verschiedener Modellierungsarten unterschiedlicher mathematischer Struktur und Informationsgehalte zur Anwendung. Dabei ist es je nach Intention erforderlich zu entscheiden, was ein geometrisches Objekt ausmacht und welche Datenstruktur für die Darstellung der Eigenschaften am geeignetsten ist. Allerdings gibt es hierbei einige Schwierigkeiten, denn nicht alles was darstellbar ist, existiert in der Realität. Zum Beispiel hat jeder Körper ein Volumen, Objekte wie etwa eine (unendliche dünne) Linie, einen Punkt oder eine Fläche gibt es nicht als eigenständige Körper. Entscheidend ist weiterhin die Eindeutigkeit der Abbildung, damit ein Körper wiederholt aus derselben Datenstruktur erstellt werden kann und umgekehrt. Dies führt zur Definition unterschiedlicher Modelle, bei denen die Integrität, die Genauigkeit der Modellierung und die Verarbeitbarkeit durch Algorithmen gewährleistet sind.

Neben der Geometrie (Lage von Punkten im Raum, Verlauf von Kurven, Form von Flächen) und der Topologie⁷ (Nachbarschaftsrelationen der geometrischen Primitive Punkte, Kante und Flächen zueinander) spielen zusätzliche Informationen für den effektiven Einsatz dieser Modelle in der Praxis eine Rolle. Das sind zum einen aus dem geometrischen Modell abgeleitete Informationen (Flächeninhalt, Schwerpunkt, etc.) und zum anderen *semantische* Informationen (anwendungsbezogene Attribute im Kontext eines höheren Ordnungsschemas, z.B. Produktmodell).

2.2.2 Methoden der Volumenmodellierung

Ziel der Volumenmodellierung ist es, die komplette geometrische Information von realen physikalischen Objekten im dreidimensionalen Raum zur Verfügung zu stellen. Definitionsgemäß ist ein *Solid* ein dreidimensionaler Körper, dessen Oberfläche so beschaffen ist, dass in jedem Punkt der Oberfläche deren unmittelbare Umgebung als zweidimensional betrachtet werden kann. Man spricht auch von einer zweidimensionalen geschlossenen und orientierbaren Mannigfaltigkeit [Bungartz 96]. Es gibt verschiedene Ansätze, um dreidimensionale Modelle von Körpern rechnergerecht zu beschreiben.

⁷ Die Topologie eines Körpers kann aus der geometrischen Struktur eines Körpers extrahiert werden. Sie ist invariant gegenüber der Translation und Rotation des Körpers im Raum.

Nach der Art der internen Darstellung werden Volumenmodelle folgendermaßen unterschieden:

- Generative Modelle (*nicht ausgewertete Modelle*). Hier wird die Erzeugungsvorschrift, die das geometrische Objekt definiert, in Form eines Algorithmus gespeichert.
- Akkumulative Modelle (*ausgewertete Modelle*). Hier werden die Datenstrukturen, die das Objekt beschreiben, gespeichert.

Im nachfolgenden Bild wird ein Überblick über wichtige Modellierungsformen für Volumenmodelle angegeben.

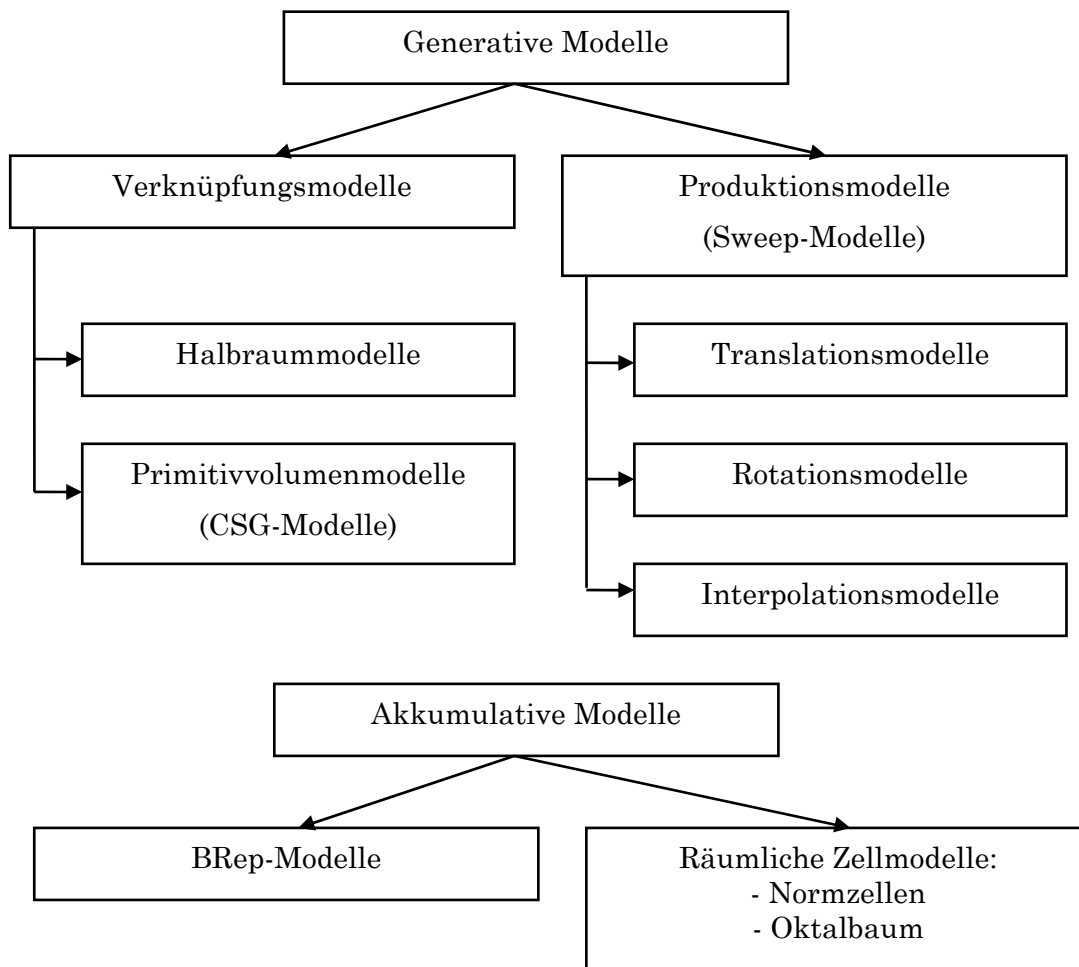


Abbildung 2: Modellierungsformen für Volumenmodelle.

2.2.3 Normzellen-Aufzählungsschema

Beim Normzellen-Aufzählungsschema wird der zu beschreibende Raum in ein Gitter gleich großer Zellen eingeteilt (siehe Abbildung 3). Als Datenmodell reicht eine Bit-Matrix, in der ein gesetztes Bit den Zustand ‚innerhalb‘ kennzeichnet. Ein nicht gesetztes Bit kennzeichnet entsprechend ‚außerhalb‘. Das Schema ist vorteilhaft in der Anwendung, da die Position einer Zelle in der Bit-Matrix einfach aus der Position im Raum berechnet werden kann. Jedem Gittermittelpunkt (Voxel) wird dabei eine Stelle in der Bit-Matrix zugeordnet. In Analogie zur dreidimensionalen Beschreibung findet das Verfahren für zwei Raumdimensionen Anwendung. Die Zellen (Pixel) werden dann in 2D-Matrizen gehalten. Das Verfahren ist gekennzeichnet durch die ausgeprägte Approximation der realen Geometrie (Treppenbildung im Modell). Dennoch gibt es Anwendungen, in denen sich dieses Speicherschema vorteilhaft anwenden lässt [Treeck 01].

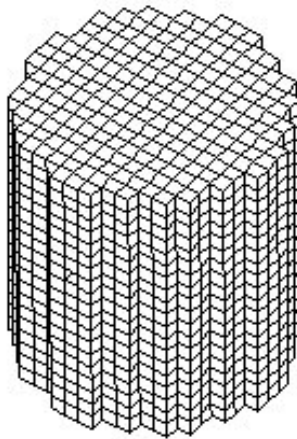


Abbildung 3: Voxelmodell eines Zylinders.

2.2.4 Oktalbaum-Schema

Ein Oktalbaum-Schema ist eine raumpartitionierende Struktur, ähnlich wie das Normzellenschema. Im Gegensatz zum Normzellenschema haben die Zellen des Oktalbaumes keine einheitliche Größe. Die Zellengröße richtet sich nach Approximationskriterien, um lokal eine ausreichende Auflösung zu erhalten. Informationen über die Anordnung der Zellen werden in einer Baumstruktur abgelegt (siehe Abbildung 4).

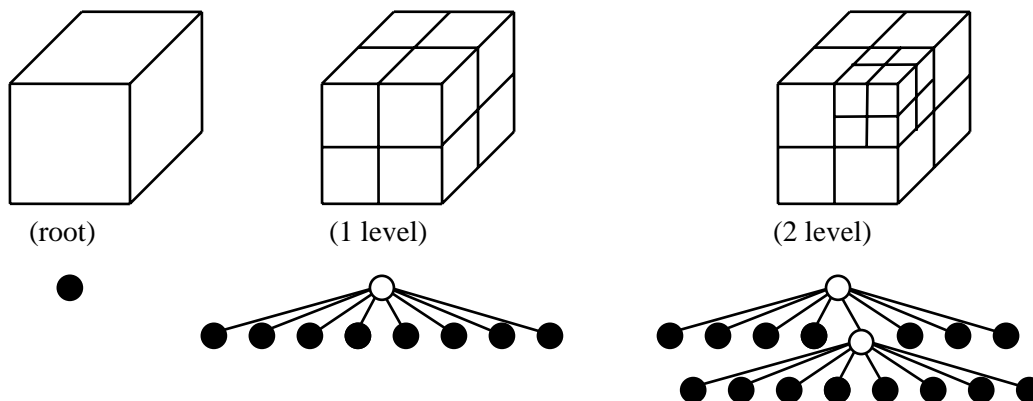


Abbildung 4: Hierarchische Struktur eines Oktalbaumes.

Zellen werden dabei in 8 Subzellen mit der halben Kantenlänge der Elternzelle verfeinert. Elemente ohne Nachfolger werden als Blätter bezeichnet und beschreiben den diskretisierten Raum. Dabei entsprechen die Ebenen (*Levels*) dem Verfeinerungsgrad der Elemente. Die Oktalbaumerzeugung erfolgt durch *rekursive* Algorithmen. Dabei wird ein Oktant nur dann verfeinert, wenn der Oktant eine nicht leere Schnittmenge mit der Oberfläche des zu beschreibenden Objekts aufweist. Die Verfeinerung wird durch eine Genauigkeitsschranke oder durch einen maximalen Auflösungs-Level gesteuert. Vorteil des Oktalbaumes hinsichtlich Speicherkomplexität ist die Lokalität der Verfeinerung hin zur Objektberandung. Damit verbleiben Zellen, die weiter von der Objektberandung entfernt liegen, in einem niedrigeren Baum-Level (größere Zellen). Für bestimmte numerische Verfahren (z.B. Strömungsberechnung) wird der Baum nach der Erzeugung geglättet. Dabei werden Oktanten nachträglich verfeinert, bis sich benachbarte Zellen nicht mehr als um einen Level unterscheiden. Mit einem Baum der Tiefe t lässt sich durch die rekursive Halbierung der Oktanten ein Genauigkeit $h = O(2^{-t})$ erreichen. Für einen Würfel mit dem Raumvolumen von 1 m^3 und einer geforderten Auflösung von 1 mm wird damit eine Baumtiefe von $t = 10$ benötigt.

Die Einsatzgebiete von Oktalbäumen sind vielfältig (z.B. numerische Simulation, geometrische Modellierung oder Visualisierung [Crouse 02, Frank 00]). Für die vorliegende Arbeit wird das Oktalbaum-Schema als übergeordnetes hierarchisches Organisationsschema zur Verwaltung von Bauwerkselementen verwendet (siehe Abbildung 5).

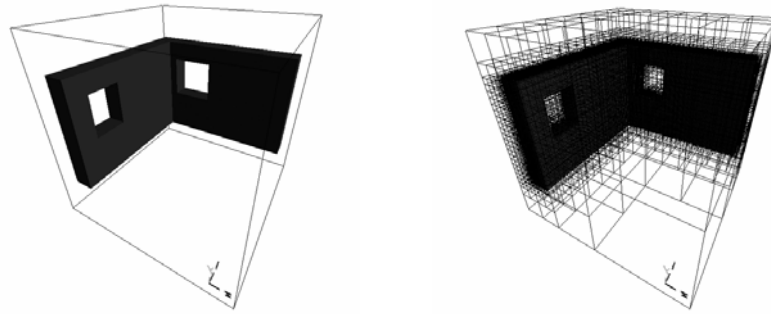


Abbildung 5: Oktalbaum zweier Körper (Wanddecke) zur Erkennung von Modellierungsfehlern (Klaffung) [Mundani 03].

2.2.5 Constructive Solid Geometry (CSG)

Die CSG baut komplexe Körper aus Grundkörpern (Primitiven) zusammen. Mittels Boolescher Operationen lässt sich eine Vielfalt von Körpern erzeugen. Drei Operatoren stehen zur Verfügung: Vereinigung (\cup), Durchschnitt (\cap) und Differenz ($/$). Primitive sind normalerweise Körper mit verhältnismäßig einfachen Oberflächen wie Quader, Kugel, Zylinder, Torus und Kegel. Wie in der schematischen Darstellung (Abbildung 6) zu sehen, ist ein *Solid* im CSG-Modell ähnlich einem binären Baum aufgebaut; die Knoten repräsentieren die Operationen und die Blätter die Primitive. Der Vorteil der CSG-Darstellung ist, dass jeder CSG-Baum einen eindeutig realisierbaren physikalischen Körper darstellt und dass der CSG-Baum oftmals den ‚Herstellungsprozess‘ des Körpers widerspiegelt. Subtrahiert man zum Beispiel einen Zylinder, so entspricht das dem Bohren eines Lochs. Der Nachteil der CSG-Repräsentierung ist, dass die Oberflächen des Körpers nicht explizit verfügbar sind. Die Kenntnis der Geometrie und Topologie der Oberfläche ist aber erforderlich, um weitere Eigenschaften eines *Solids* bestimmen zu können oder um Attribute an Oberflächen anheften zu können.

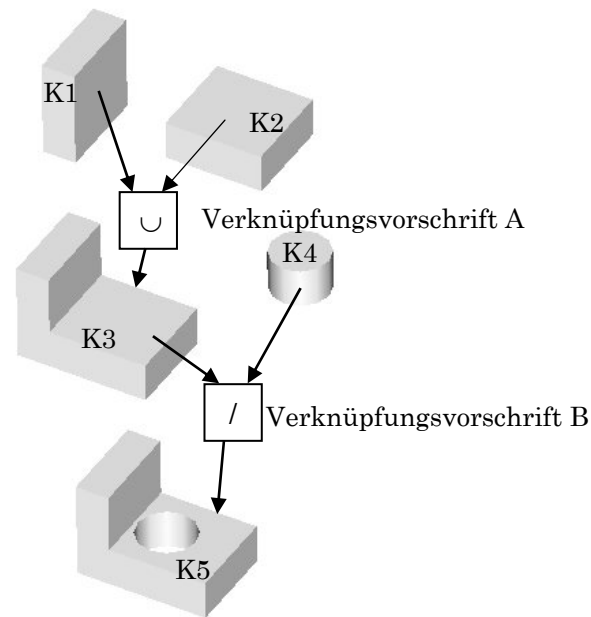


Abbildung 6: CSG-Baum mit Primitiven (K1,K2,K4), den Verknüpfungsvorschriften (A,B) und dem ausgewerteten Modell (K5).

2.2.6 Boundary Representation (BRep)

Die Boundary Representation eines *Solids* ähnelt einem Flächenmodell. Zusätzlich zur kompletten geometrischen Information wird aber noch die topologische Struktur der Oberfläche gespeichert. Körper mit gleicher Topologie können dabei unterschiedliche geometrische Informationen tragen (topologische Invarianz) [Bungartz 96]. Anschaulich betrachtet kann man sich einen deformierten Gummikörper vorstellen. Wie bei der CSG-Methode geht man bei der Modellierung von Primitiven aus. Die Vorteile des BRep sind offensichtlich: Die explizite geometrische und topologische Beschreibung des Solids ermöglicht es, alle Maßeigenschaften zu bestimmen (Volumen, Schwerpunkt, etc.) und den geometrischen wie topologischen Elementen zusätzliche Attribute anzuhängen (z.B. für das semantische Modell).

Geometrie und Topologie sind bei einer BRep weitgehend unabhängig voneinander. Ein Modellierungssystem, das auf komplexen Freiformflächen basiert, kann zur Speicherung der topologischen Information dieselbe Datenstruktur benutzen wie ein System, das ausschließlich mit ebenen Flächen arbeitet.

Die Leistungsfähigkeit eines *Solid-Modeling* Systems hängt stark davon ab, welche Klassen von Oberflächen exakt modelliert werden können. Der einfachste Ansatz besteht darin, nur mit ebenen Flächen zu arbeiten. Man spricht dann von polyedrischen *Solid-Modeling* Systemen. Da beliebig gekrümmte Oberflächen durch genügend kleine ebene Flächenstücke approximiert werden können, sind einem polyedrischen *Solid-Modeler* prinzipiell kaum Grenzen gesetzt. Bei hoher Auflösung der Geometrie haben die Modellierer jedoch auf Grund der Vielzahl der notwendigen *Patches* eine ungünstige Speicher- und Zeitkomplexität. Ebene Flächen haben den Vorteil, dass sich die Schnittkurven zweier beliebiger Flächenstücke einfach berechnen lassen: Es resultieren immer Strecken. Werden weitere Klassen von Oberflächen in ein *Solid-Modeling-System* eingeführt, so wird der Aufwand zur Manipulation von Körpern hauptsächlich durch die Berechnung der Schnittkurven von Flächenstücken bestimmt. In speziellen Anwendungsfällen lässt sich die Berandung nur noch mit Freiformflächen beschreiben

(B-Splines, NURBS). Ein Beispiel zu B-Splines ist in Abbildung 7 angegeben. Die Berechnung der Schnittkurven von Freiformflächen kann oft nur approximativ erfolgen. Der in einem nachfolgenden Kapitel vorgestellte Geometriemodellierer ACIS beinhaltet Klassen für Freiformflächen zur Beschreibung von Oberflächen.



Abbildung 7: Modellierung eines umgeformten Bauteils aus dem Automobilbau mit B-Spline Oberfläche [Muthler 04].

Topologische Information

Die Halbkanten-Datenstruktur (Halfedge-Datenstruktur) zur Speicherung topologischer Informationen ist Bestandteil der ursprünglich von Baumgart 1974 eingeführten *Winged-Edge*-Datenstruktur [Borkner 96]. *Face* bezeichnet ein einzelnes, berandetes Flächenstück. Für jeden Punkt eines *Face* ist eine Normale definiert, die ins Innere des *Solid* zeigt. Die *Faces* sind entlang ihrer Kanten (*Edges*) zusammengefügt. Jede *Edge* wird von zwei Punkten (*Vertices*) berandet. Die Objekte *Face*, *Edge* und *Vertex* bilden die Hauptentitäten der Halbkanten-Datenstruktur.

Eine *Face* kann beliebig viele Ringe, sogenannte *Loops* referenzieren. Auf den *Loops* wird eine Orientierung definiert, die man sich mittels der Rechtehandregel verdeutlichen kann, wobei der Daumen dabei in Richtung der Flächennormale zeigt und die Finger die Orientierung des *Loop* ergeben. Da die Oberfläche eines *Solids* an jeder Stelle zweidimensional ist (zwei-mannigfach), können sich entlang jeder *Edge* höchstens zwei *Faces* treffen. Definitionsgemäß ist die Oberfläche eines *Solids* immer geschlossen, also treffen sich an jeder *Edge* auch mindestens zwei *Faces* (*Radial-Edges* können mehr als zwei anliegende *Faces* verwalten⁸). Eine Datenstruktur zur Speicherung der topologischen Information zusammengesetzter Oberflächen macht von dieser Eigenschaft Gebrauch. Dazu dürfen in der Struktur nur Objekte repräsentiert sein, die die Konsistenzbedingung erfüllen. In der Datenstruktur wird das erreicht, in dem jeder *Edge* zwei Halbkanten (*halfedges*) zugeordnet werden. Ein solches Auftrennen von *Edges* in *Halfedges* ermöglicht es, jeden *Loop* als zirkuläre Liste von *Halfedges* aufzufassen. Die Orientierung des *Loops* spiegelt sich in der Abfolge der zirkulären Liste wider. Das Datenschema führt dazu, dass immer zwei benachbarte *Halfedges* entgegengesetzt orientiert sind (siehe Abbildung 8).

⁸ *Solids* mit Kanten, die mehr als zwei angrenzenden *Faces* verwalten, gehören zur Klassen der physikalisch nicht realisierbaren (*non-manifold*) Gebilde.

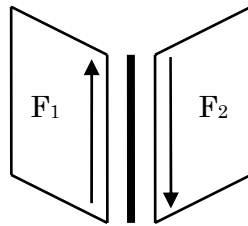


Abbildung 8: Halbkanten-Modell.

Für die topologischen Elemente *Vertex*, *Edge*, *Face*, *Loop* und *Halfedge* ergeben sich folgende Zusammenhänge:

- **Face:** speichert die Menge der *Loops* (Berandung und Löcher).
- **Loop:** speichert *Face* und beliebig viele *Halfedges*.
- **Edge:** speichert die zugehörigen *Halfedges*.
- **Halfedge:** speichert nächste/vorangegangene *Halfedge* im *Loop*, zugehörige *Edge*, zugehörigen *Loop*.
- **Vertex:** speichert *Halfedges*, die den *Vertex* als Start- oder Endvertex verwenden.

Wichtiger Aspekt der *Halfedge*-Datenstruktur ist, dass die topologische Information jeder Oberfläche damit abgebildet werden kann, unabhängig von der Art der Flächenmodellierung. Die *Boundary Representation* (BRep) ist dann vollständig, wenn für die Elemente *Face*, *Edge* und *Vertex* der *Halfedge*-Datenstruktur zusätzlich noch geometrische Informationen gespeichert werden.

2.2.7 Produktionsmodelle (Sweeping-Modelle)

Translationsmodell

Hierbei wird eine zweidimensionale Fläche F längs einer Verschiebekurve v um den Betrag dv bewegt und der überstrichene Raum als Volumen betrachtet.

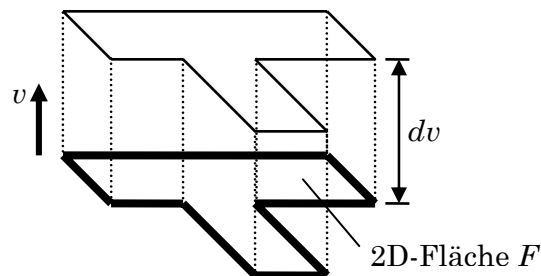


Abbildung 9: Generierung eines Volumenkörpers durch translatorisches Sweeping einer ebenen Fläche.

Rotationsmodell

Anders als beim *translatorischen Sweeping* wird hier eine Kreislinie k für die rotationssymmetrische Bewegung der Grundfläche F verwendet.

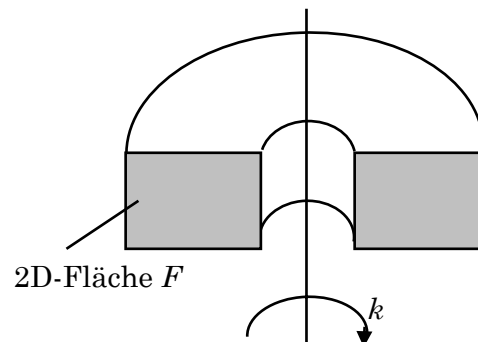


Abbildung 10: Generierung eines Volumenkörpers mittels Rotation längs einer Kreislinie.

Interpolationsschema (Lofting)

Dabei wird der resultierende Körper durch Kurveninterpolation zwischen zwei Punkten $P1$ auf der Fläche $F1$ und $P2$ auf der Fläche $F2$ erzeugt. Im einfachsten Fall sind die Kurven Geraden.

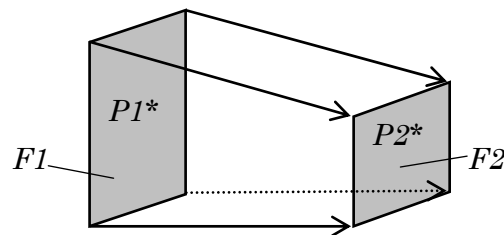


Abbildung 11: Generierung von Volumenkörper mittels Kurveninterpolation.

Die drei genannten Produktionsmodelle werden auch als $2^{1/2}$ D-Modelle bezeichnet, da hierbei Volumenkörper aus zweidimensionalen Strukturen gewonnen werden. Die Produktionsmodelle spielen in der vorliegenden Arbeit eine wichtige Rolle bei der geometrischen Beschreibung von Bauelementen im Rahmen von Bauwerksmodellen.

2.3 Diskretisierung der Geometrie

2.3.1 Allgemeines

Voraussetzung für FE-basierte Berechnungsverfahren ist die Diskretisierung kontinuierlicher Körper in Form von Gittern oder Netzen, d.h. eine überschneidungsfreie Zerlegungen von Kontinua in einfache geometrische Elemente. Im zweidimensionalen Fall sind diese Elemente häufig Dreiecke oder Vierecke, im dreidimensionalen Fall können es verschiedenste Polyeder, beispielsweise Tetraeder oder Hexaeder sein. Das in dieser Arbeit angewandte numerische Verfahren zur Berechnung räumlicher Spannungszustände, die p -Version der FEM, verwendet als Diskretisierungselemente Hexaeder. Diese werden mittels Kombination zweier unterschiedlicher Generierungstechniken erzeugt. Zum einen sind das Makro-Netzgeneratoren (siehe dazu Kapitel 3.4) und zum anderen eine Technik, die Hexaederelemente durch Extrusion von 2D-Elementen generiert. Makronetzgeneratoren können dort sinnvoll eingesetzt werden, wo die Ausgangsgeometrie leicht in Hexaederelemente unterteilt werden kann, z.B. durch Teilung 'stabförmiger' Strukturen. Allgemeine geometrische Bereiche lassen sich dagegen i.d.R. nur über eine Freivernetzung diskretisieren. Eine direkte Vernetzung allgemeiner räumlicher Strukturen in Hexaeder scheitert oft an der Komplexität der Aufgabe. Dagegen ist es relativ einfach, platten-/scheibenartige Strukturen über eine 2D-Freivernetzung der Oberflächen zu vernetzen und die 2D-Elemente durch Extrusion in Hexaederelemente zu überführen (siehe dazu [Rücker 01]). Im Bereich des Massivbaus werden Bauwerke meist monolithisch in Ortbetonbauweise hergestellt und als Kombination aus Stützen, Scheiben und Platten zusammengesetzt. Für derartige Strukturen hat sich das im folgenden beschriebene kombinierte Makro-/Freivernetzung-Verfahren als geeignet erwiesen.

Der hier eingesetzte 2D-Netzgenerator basiert auf der Idee der rekursiven Gebietsteilungstechnik, die in [Bank 90] erstmals vorgestellt wurde. Dabei werden zuerst die Ränder gemäß einer global vorgegebenen Elementkantenlänge unterteilt. Die Elemente im Inneren einer Fläche werden in der Art generiert, dass die Fläche so lange rekursiv in Teilflächen unterteilt wird, bis sich die letzte Teilfläche nur noch aus drei Knoten zusammensetzt. Diese Teilfläche wird als Dreieckselement abgespeichert und der Vorgang so lange wiederholt, bis sich die gesamte Fläche nur noch aus Dreiecken zusammensetzt. Über eine Konvertierung mittels geeigneter Strategien lässt sich aus dem Dreiecksnetz ein Vierecksnetz erzeugen. Damit benachbarte Elementgrößen nicht zu stark variieren und im Sinne einer FE-Berechnung schlechte Elemente generiert werden, steuert ein geometrischer Progressionsfaktor den Übergang von groben zu feinen Elementen. Eine Angabe weiterer Parametereinstellungen entfällt, da der gesamte Ablauf optimiert ist und vollautomatisch abläuft.

2.3.2 Methode der Gebietsteilung mit uniformer Elementkantenlänge

Übergeordnete Struktur bei der Vernetzung zweidimensionaler, ebener Bereiche ist die Region. Diese gliedert das zu vernetzende Gebiet in mehrere mathematisch einfach zusammenhängende Teilgebiete, d.h. die Kantenfolgen der Regionen müssen geschlossen sein und es dürfen keine Löcher in der Beschreibung vorkommen. Der Umlaufsinn der Kantenfolge einer Region ist bedeutsam. Der Umlauf der Berandung verläuft im mathematisch positiven Sinne, d.h. das zu vernetzende Gebiet liegt auf der linken Seite der Kantenfolge. Das im Rahmen der vorliegenden Arbeit entwickelte geometrische Zerlegungsverfahren entkoppelt in einem ersten Schritt alle Regionen, die im Vernetzungsdurchlauf mittels Freivernetzung diskretisiert werden. Die Regionen werden dabei sequentiell nacheinander vernetzt. Einzelheiten zur geometrischen Zerlegung und zur Sicherstellung der Kompatibilität der getrennt erstellten Diskretisierungen werden in Kapitel 3 beschrieben.

Eine Beeinflussung der Netzgenerierung ist durch zwei Parameter möglich, die max. Elementkantenlänge (auch Knotenabstandswert genannt) h und den Progressionsfaktor p . Der h -Wert wird vor der Vernetzung den Strukturkanten zugewiesen und beeinflusst durch eine Übertragung auf erzeugte Teilungskanten ebenfalls die Diskretisierung im Inneren der Region. Bei unterschiedlichen h -Werten an den Anfangs- und Endknoten einer Berandungskante wird der h -Wert über die Länge der Berandungskante mittels eines Wertes p' abgestuft, wobei p' maximal den Wert p erreicht. Ziel des p -Wertes ist es, einen kontrolliert abgestuften Verlauf des h -Wertes über die Berandungskanten zu gewährleisten. Nachfolgende Abbildung zeigt die BRep-Beschreibung einer Region sowie die diskretisierten Strukturkanten.

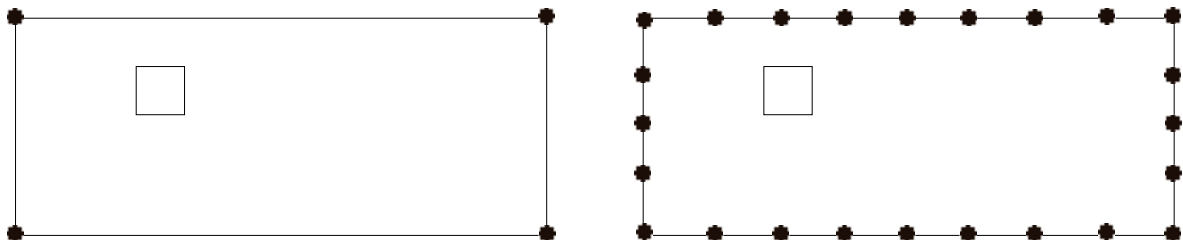


Abbildung 12: Unterteilung der Regionsberandung.

Strukturelemente im Inneren der Region (z.B. Löcher) werden über neue Kanten (die Verbindungskanten) an den Regionsrand angebunden. Von allen Kanten, die ein Strukturelement im Inneren der Region an den Rand verschneidungsfrei anbinden könnten, wird diejenige ausgewählt, deren Länge minimal ist.

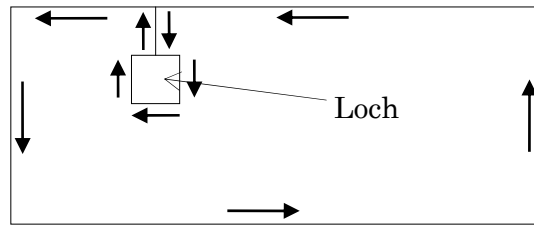


Abbildung 13: Anbindung innerer Strukturen an den Regionsrand.

Durch den im Vergleich zum Strukturrand entgegen gesetzten Umlaufsinn (Uhrzeigersinn) der Löcher und den eingefügten Verbindungskanten bleibt der Polygonzug der Regionsberandung geschlossen und die umschriebene Fläche einfach zusammenhängend. Der Algorithmus zur Vernetzung kann somit unverändert angewandt werden.

Bei der Generierung von Dreiecken im Inneren der Region wird zunächst überprüft, ob sich ein Dreieck durch Generierung einer Kante zwischen zwei Knoten der Berandung abspalten lässt. Die Prüfung erfolgt unter Anwendung eines Gütekriteriums, das in [Bank 90, Rank 94] näher beschrieben ist. Wird ein geeignetes Dreieck gefunden, erfolgt die Abspeicherung des Dreiecks und die Aktualisierung der Regionsbeschreibung. Dieser Vorgang wird rekursiv durchgeführt, bis kein geeignetes Dreieck mehr abgespalten werden kann. Andernfalls wird die Region mittels einer geraden Kante geteilt, indem beginnend mit einem willkürlich ausgewählten Knoten eine Teilungskante zwischen zwei Knoten der Berandung gesucht wird, die folgende Kriterien erfüllen:

$$\begin{aligned}
 \alpha_1 &> \frac{\pi}{4} \\
 \alpha_2 &> \frac{\pi}{4} \\
 \alpha_3 &< \frac{\pi}{4} \\
 \alpha_4 &< \frac{\pi}{4}
 \end{aligned} \tag{1}$$

Die Winkel $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ haben die nachfolgend dargestellte geometrische Bedeutung:

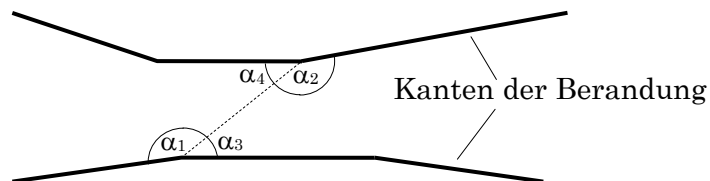


Abbildung 14: Winkel bei der Generierung von Teilungskanten.

Nach der Teilung der Region ergibt sich beispielhaft folgende geometrische Situation:

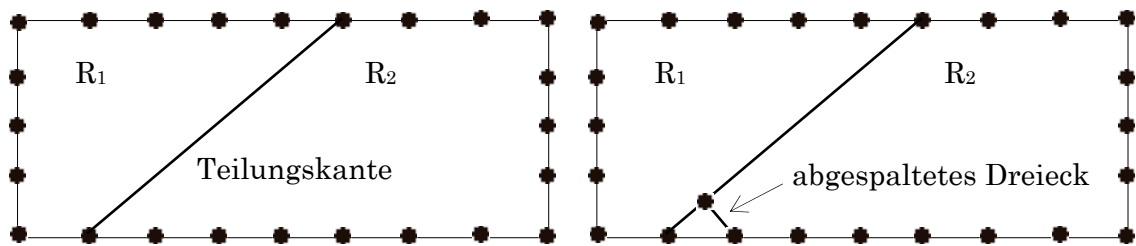


Abbildung 15: Teilung der Region mittels gerader Kante (links), Abspaltung eines Dreiecks aus Region 2 (rechts).

Die rekursive Teilung von Regionen und Abspaltungen von Dreiecken wird solange fortgeführt, bis die gesamte gegebene Struktur vollständig mit Dreiecken vernetzt ist. Dabei werden auf jeder Teilungskante gemäß den Knotenabstandswerten der Anfangs- und Endknoten weitere Knoten erzeugt (siehe Abbildung 15 rechts).

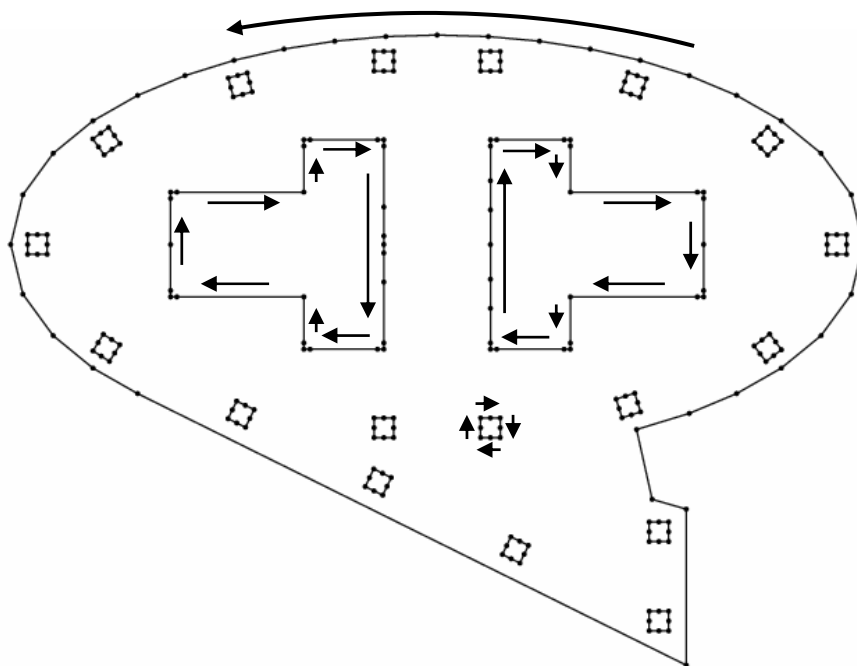


Abbildung 16: Geometrische Beschreibung eines komplexen Beispiels.

In Abbildung 16 ist die Regionsberandung mit zugeordnetem Umlaufsinn (mathematisch positiv) dargestellt. Kanten von Löchern werden mit entgegengesetztem Umlaufsinn angeordnet.

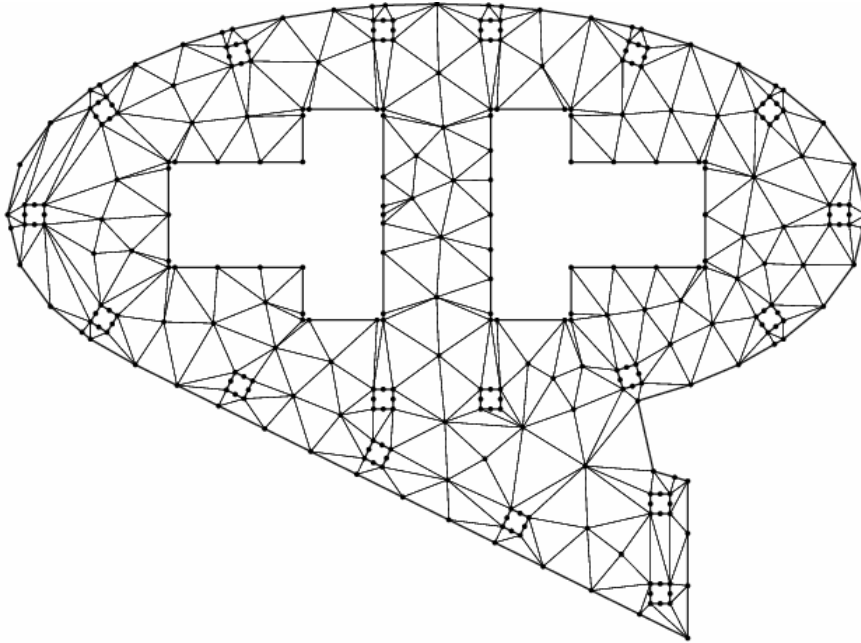


Abbildung 17: Mittels Dreiecken vernetzte Struktur.

2.3.3 Konvertierung des Dreiecksnetzes

Das generierte Dreiecksnetz wird nun im nächsten Schritt in ein reines Vierecksnetz überführt. Grund dafür sind höhere Genauigkeiten der Ergebnisse in der FE-Berechnung bei Verwendung von Viereckselementen (siehe dazu [Szabó 91]). Die Netzkonvertierung wird mit drei nacheinander ausgeführten Verfahren durchgeführt:

- Überführung von vier Dreiecken mit einem gemeinsamen Knoten in vier Vierecke gemäß Abbildung 18 links.
- Vereinigung zweier benachbarter Dreiecke zu einem Viereck. Danach Zerlegung des Vierecks in vier Vierecke gemäß Abbildung 18 Mitte.
- Auflösung isolierter Dreiecke gemäß Abbildung 18 rechts.

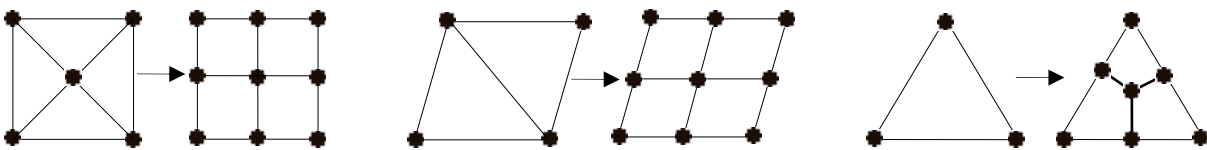


Abbildung 18: Drei Verfahren der Netzkonvertierung in ein Vierecksnetz.

Alle drei Verfahren zur Erzeugung von Viereckselementen generieren Knoten auf den Kantenmittelpunkten der ursprünglichen Dreieckselemente. Dieses Vorgehen sichert die Kompatibilität zwischen den Elementen und vermeidet hängende Knoten. Die Form der Elemente ist abhängig von der Reihenfolge der Verknüpfung zweier Dreiecke in Schritt zwei der Konvertierung. Deshalb werden vor der Konvertierung alle Kombinationen benachbarten Dreiecke untersucht und wie in [Rank 94] beschrieben bewertet. Bei der Konvertierung von Dreiecken in Vierecke werden Kombinationen mit der besten Bewertung bevorzugt.

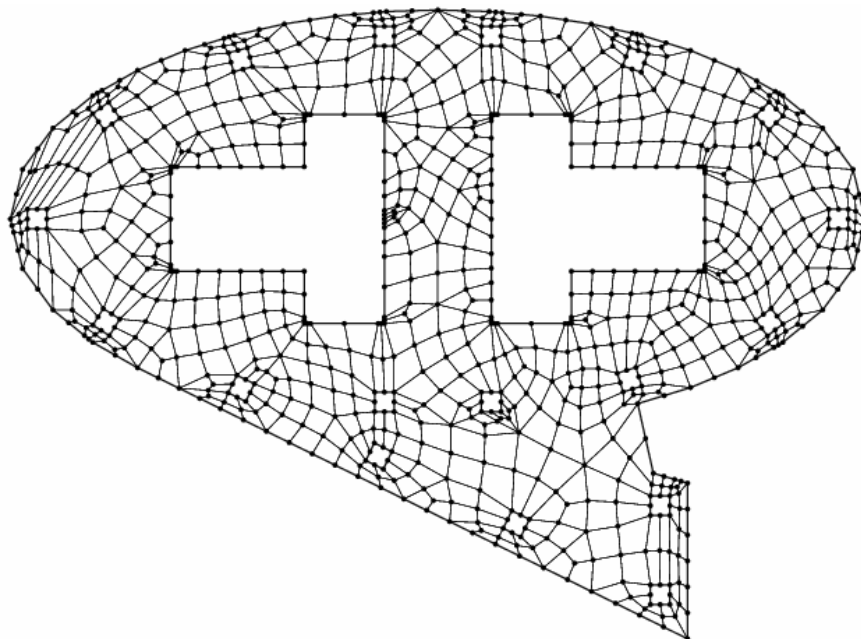


Abbildung 19: Konvertiertes Netz mit Viereckselementen.

Das in Abbildung 19 gezeigte Vierecksnetz⁹ wäre für eine Berechnung mit der h -Methode der FEM nur schlecht geeignet, da die Form vieler Elemente stark von der für die h -Version optimalen Quadratgestalt abweicht. Die in dieser Arbeit verwendete p -Version der FEM setzt jedoch andere Kriterien für die Netzelemente. Die Elemente können im Vergleich zu Elementen der h -Version groß und lang gestreckt sein (Siehe dazu Kapitel 2.4). Für die Vernetzung der Platte aus Abbildung 19 wurden der Eingangsparameter h (max. Elementkantenlänge) so gewählt, dass ein Kompromiss zwischen Verzerrung und Größe der Elemente erzielt wurde.

⁹ Das eingesetzte Programm DOMESH zur Freivernetzung wurde ursprünglich für die Generierung von h -Netzen entwickelt, hier aber durch geeignete Parameterwahl auch für die p -Version genutzt.

2.3.4 Adaptive Netzerzeugung

Bei der adaptiven Netzgenerierung werden die Knotenabstandswerte im Inneren einer Region nicht über Abstandswerte von Knoten der Regionsberandung gesteuert, sondern über vorgegebene h -Werte in der Region. Dabei werden die h -Werte über eine auf einem Hintergrundnetz definierte Dichtefunktion bestimmt. Mit dieser Methode lassen sich neben Verfeinerungen ebenso Vergrößerungen des Netzes erreichen. Diese Vorgehensweise zur adaptiven Netzgenerierung erfordert jedoch zwei Vernetzungsläufe, einen ersten Lauf zur Erzeugung des Hintergrundnetzes und einen zweiten Vernetzungslauf, bei dem die Dichtefunktion ausgewertet wird.

Knotenabstandswerte auf dem Hintergrundnetz können entweder *a priori* an Orten mit zu erwarteten Lösungssingularitäten oder automatisch aus Ergebnissen einer Fehlerschätzung (*a posteriori*) vorgegeben werden. Es können dabei Übergangsbereiche entstehen, die durch eine sprunghafte Änderung der h -Werte gekennzeichnet sind. Die Korrektur zur Anpassung der h -Werte erfolgt mittels eines Glättungsalgorithmus. Dabei werden alle Knoten anhand ihres Knotenabstandswertes in eine sortierte Liste eingefügt. Ausgehend vom Knoten mit dem kleinsten Abstandwert wird über den Progressionsfaktor der maximal zulässige Abstandwert des Nachbarknotens ermittelt und gegebenenfalls dem Nachbarknoten zugewiesen. Nach einer Korrektur wird der Knoten mit dem geänderten Abstandwert erneut in die Liste einsortiert. Die Korrektur wird für alle Knoten der sortierten Liste vorgenommen. Kleine h -Werte bei Knoten in der Region beeinflussen damit die h -Werte bei Knoten auf der Regionsberandung. Wird beispielsweise eine Platte auf Stützen gelagert, bei der die Stützen nahe am Plattenrand angeordnet sind, wird über den Glättungsalgorithmus der Regionsrand in diesen Bereichen beim adaptiven Vernetzungslauf feiner vernetzt. Eine detaillierte Beschreibung des Vorgehens zur adaptiven Netzgenerierung ist in [Schweingruber 99] beschrieben.

Die Glättung der h -Werte und damit der Transfer von Informationen zur Netzdichte von Bereichen in der Region auf den Regionsrand ist für das gesamte in dieser Arbeit vorgestellte Verfahren von besonderer Bedeutung, da Informationen zur Diskretisierung in einem ersten Schritt in sog. Verbindungsobjekten abgelegt werden, die mittels einer Zerlegung des Eingangsmodells generiert werden (siehe nachfolgende Kapitel). Die sog. Verbindungselemente erfüllen damit bei der Vernetzung in 3D – neben weiteren hier noch nicht genannten Aufgaben – annähernd die Funktion der Koppelkanten benachbarter Regionen in 2D.

2.4 Strukturanalyse mit der p -Version der FEM

2.4.1 Einführung

Die Finite-Element-Methode ist eines der wichtigsten Werkzeuge für die Berechnung von Näherungslösungen für partielle Differentialgleichungen. Der Näherungsfehler hat zwei Ursachen: Zum einen wird er durch die Modellannahme eingeführt, die darauf abzielt, das zugrunde liegende physikalische Problem zu vereinfachen. Dagegen wird der Diskretisierungsfehler durch die Finite-Element-Methode selbst eingeführt. Beide Einflüsse müssen kontrolliert werden und sollten sich nur im vordefinierten Umfang auf die Lösung auswirken, um so die Zuverlässigkeit der Berechnung garantieren zu können.

Die Kontrolle des Diskretisierungsfehlers erfolgt entweder durch Anwendung der h -Version, bei der das Netz lokal oder global verfeinert wird, oder durch die p -Version. Bei der p -Version bleibt das finite Element-Netz unverändert, es wird jedoch der polynomiale Ansatz der Formfunktionen so lange erhöht, bis hinreichende Konvergenz der Näherung erreicht ist. Es wurde bereits in den 80er Jahren gezeigt, dass die p -Version für lineare elliptische Probleme mit glatten Lösungen exponentiell in der Energienorm konvergiert. Singularitäten in der Lösung können oft durch Verfeinerungen *a priori* berücksichtigt werden, so dass eine prä-asymptotische exponentielle Konvergenzrate für die reine p -Version eine Genauigkeit erbringt, die durch die klassische h -Version kaum erreicht werden kann.

Der Modellfehler wird durch die Einführung verschiedener physikalisch motivierter Vereinfachungen, wie z.B. der Vernachlässigung zeitabhängiger Effekte oder der Annahme vereinfachter Materialgesetze verursacht. Eine weitere Fehlerquelle ist die Dimensionsreduzierung bei der Anwendung von Platten- und Scheibenmodellen zur Berechnung dünnwandiger Strukturen. Wesentlich dabei ist die Annahme einer bestimmten polynominalen Verschiebung in einer bestimmten lokalen Richtung (z.B. die Dickenrichtung einer Platte oder einer Schale). Bei der Reissner-Mindlin Theorie für Platten oder der Naghdi-Theorie für Schalen wird z.B. das Ebenbleiben der Querschnitte angenommen. Ein bedeutender Fehler kann deshalb in jenen Bereichen auftreten, in denen die getroffenen Annahmen der Dimensionsreduzierung verletzt werden. Zur Überwindung derartiger Probleme wird in der vorliegenden Arbeit eine Hexaeder-Elementformulierung hoher Ordnung verwendet. Es können damit unterschiedliche Polynomgrade für die drei lokalen Richtungen sowie für den Ansatz der drei Komponenten des Verschiebungsfeldes gewählt werden. Die Verwendung eines anisotropen Ansatzes führt besonders für dünnwandige Strukturen zu einer effizienten Approximation und ermöglicht darüber hinaus gleichzeitig die Steuerung des Modellfehlers. Weitere Einzelheiten sind in [Düster 01, Düster 02, Düster 03, Rank 04b] nachzulesen.

2.4.2 Finite Elemente mit hoher Ordnung

2.4.2.1 Hierarchische Formfunktionen für eindimensionale Problemstellungen

Die Standardansatzfunktion (*Lagrange*-Polynome) im Standardintervall $\Omega_{st}=[-1,1]$ eines Elements, können ebenso durch eine Kombination hierarchischer Basisfunktionen (*Legendre*-Polynome) repräsentiert werden (siehe Tabelle 1). Ein grundlegender Unterschied zwischen beiden Ansätzen liegt darin, dass bei einer hierarchischen Basis alle Ansatz-Funktionen niederer Ordnung auch in der Menge der Ansatzfunktionen höherer Ordnung enthalten sind.

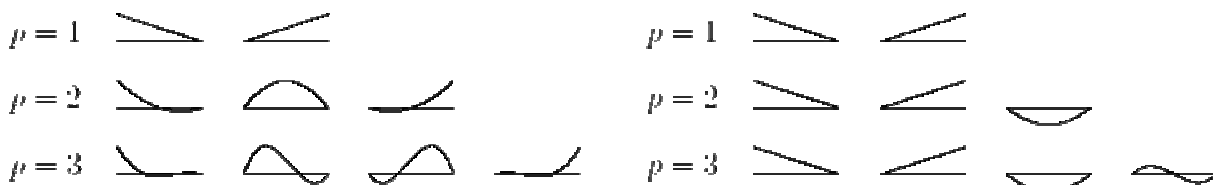


Tabelle 1: Standardfunktion (links) und hierarchische Formfunktionen (rechts) für $p = 1, 2, 3$.

Die hier verwendeten eindimensionalen hierarchischen Formfunktionen wurden von Szabó und Babuška vorgestellt [Szabó 91] und sind gegeben durch

$$\begin{aligned} N_1(\xi) &= 1/2(1 - \xi) \\ N_2(\xi) &= 1/2(1 + \xi) \\ N_i(\xi) &= \phi_{i-1}(\xi), i = 3, 4, \dots, p + 1 \end{aligned} \quad (2)$$

mit

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

wobei $L_j(\xi)$ die bekannten *Legendre*-Polynome darstellen. Die Funktionen $N_1(\xi), N_2(\xi)$ werden als Knoten-Formfunktionen oder Knoten-Moden bezeichnet.

Die Funktionen $N_i(\xi), i=3, 4, \dots$ werden innere Formfunktionen, innere Moden (engl. *bubble modes*) genannt.

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

2.4.2.2 Hierarchische Formfunktionen für Hexaeder

Die Implementierung dreidimensionaler finiter Elemente hoher Ordnung basiert auf einer Hexaeder Element-Formulierung unter Verwendung der eindimensionalen hierarchischen Ansatzfunktionen (Tabelle 1). Die Elemente sind für ‚dicke‘ sowie für dünnwandige Strukturen geeignet. Bei scheibenartigen Strukturen kann ein anisotroper Ansatz zur Anwendung kommen, bei dem in Dickenrichtung des Elements ein anderer Ansatz als in den beiden anderen Richtungen gewählt wird (siehe [Düster 01, Düster 02]).

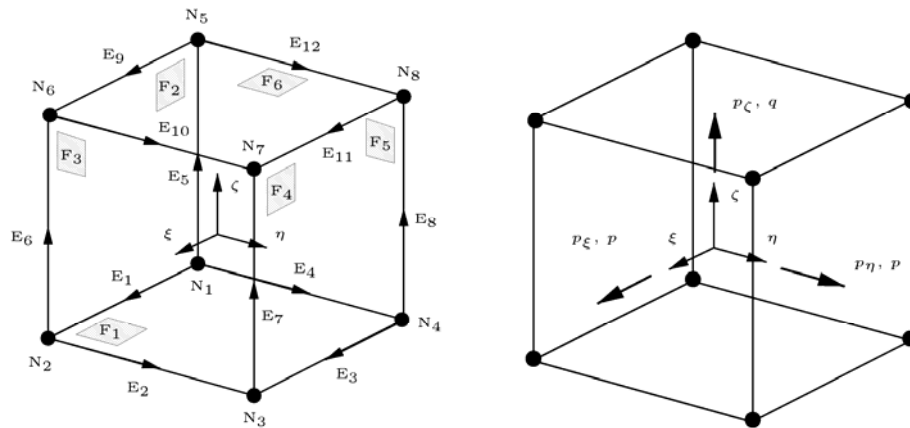


Abbildung 20: Standard Hexaederelement Ω_{st}^h : Definition von Knoten, Kanten, Flächen und Polynomgraden.

Dreidimensionale Formfunktionen können in vier Gruppen eingeteilt werden:

1. Knotenmoden (nodal nodes):

Knotenmoden bilden die tri-linearen Formfunktionen. Diese sind bereits durch das isoparametrische, achtknotige Hexaederelement allgemein bekannt. (ξ_i, η_i, ζ_i) sind die lokalen Koordinaten des i -ten Knoten.

$$N_{1,1,1}^{N_i}(\xi, \eta, \zeta) = \frac{1}{8}(1 + \xi_i \xi)(1 + \eta_i \eta)(1 + \zeta_i \zeta) \quad i = 1, \dots, 8 \quad (5)$$

2. Kantenmoden (edge modes):

Diese Moden werden für jede Kante getrennt definiert. Zur Kante E_1 in Abbildung 20 lauten beispielsweise die korrespondierenden Kantenmoden:

$$N_{i,1,1}^{E_1}(\xi, \eta, \zeta) = \frac{1}{4}(1 - \eta)(1 - \zeta)\phi_i(\xi) \quad (6)$$

3. Flächenmoden (face modes):

Diese Moden werden für jede Fläche definiert. Zur Fläche F_1 korrespondieren folgende Flächenmoden:

$$N_{i,j,1}^{F_1}(\xi, \eta, \zeta) = \frac{1}{2}(1 - \zeta)\phi_i(\xi)\phi_j(\eta) \quad (7)$$

4. Interne Moden (bubble modes):

Interne Moden sind nur im Element definiert und verschwinden an den Elementrändern:

$$N_{i,j,k}^{\text{int}}(\xi, \eta, \zeta) = \phi_i(\xi)\phi_j(\eta)\phi_k(\zeta) \quad (8)$$

Die Indizes i, j, k der Formfunktionen bezeichnen den Polynomgrade in den lokalen Richtungen ξ, η, ζ (siehe Abbildung 21).

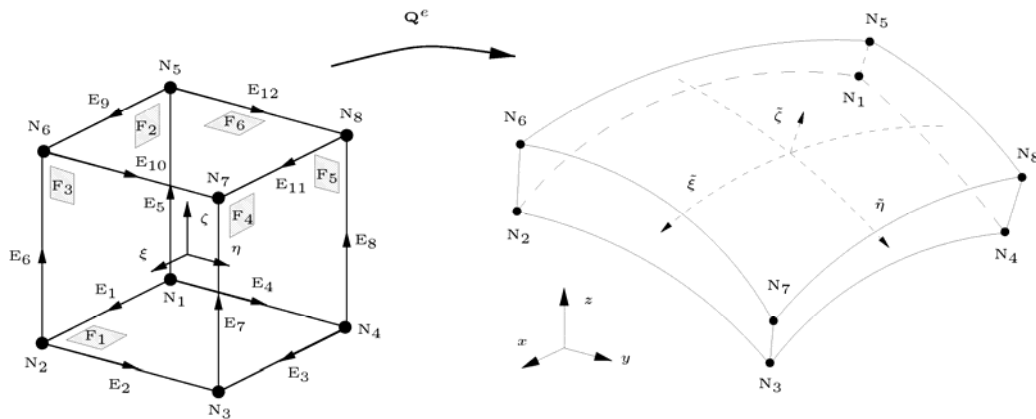


Abbildung 21: Zweifach gekrümmtes plattenartiges Element hoher Ordnung.

Es ist zweckmäßig, für die Knotenpunktkoordinaten neben einem elementbezogenen lokalen ein globales Koordinatensystem zu verwenden. Beide werden durch Transformationsfunktionen z.B. über die Methode der Blending-Funktionen miteinander verknüpft [Bröker 01, Abbildung 21]. Die Verschiebungen werden nun aus der Bedingung ermittelt, dass im gesuchten Gleichgewichtszustand die potentielle Energie ein Minimum hat. Die potentielle Energie einer Konstruktion ist die Summe aus der inneren Verzerrungsenergie (der elastischen Formänderungsenergie) und dem Potential der aufgebrachtten Lasten (der von äußeren Kräften geleisteten Arbeit). Dabei wird letztlich ein Gleichungssystem aufgestellt, das aus einer Gesamtsteifigkeitsmatrix, einem Verschiebungsvektor und einem Kraftvektor besteht. Wird es gelöst, erhält man die gesuchten Verformungen.

3 Tragwerksplanung am dreidimensionalen Modell

3.1 Produktmodellierung im Bauwesen

3.1.1 Einführung

Der Begriff *Produkt* ist in der ISO 9002 als ein ‚*Ergebnis von Tätigkeiten und Prozessen*‘ definiert. *Produkte* können Dienstleistungen, Hardware, verfahrenstechnische Produkte, Software sowie Kombinationen daraus sein. Ein Produkt kann generell materiell (Gebäude) oder immateriell (Wissen, Entwürfe) oder eine Kombination aus beidem sein [Rüppel 96]. Im Sinne dieser Begriffsdefinition ist ein Bauwerk als Produkt zu bezeichnen.

Im Bauwesen ergeben sich nun spezifische Anforderungen an ein Produktmodell. Bei einem Bauwerk handelt es sich um ein Produkt, welches in der Regel durch seine Einzigartigkeit, Vielgestaltigkeit und lange Lebensdauer besondere Schwierigkeiten bei der Beschreibung hervorruft. Der Unikatcharakter begründet sich in den zu differenzierenden Randbedingungen, welche unter anderem durch wechselnde Baugrundeigenschaften, Einwirkungsverhältnisse oder Nutzungsanforderungen hervorgerufen werden. Allgemeine Klassifizierungskriterien und Ordnungsprinzipien (Objekt-Taxonomie) lassen sich nur in wenigen Fällen finden. Dies macht eine formale Beschreibung schwierig. Ein Bauwerk lässt sich zwar als Ansammlung von Bauteilen wie Wänden, Trägern oder Decken verstehen. Die Vielgestaltigkeit von Bauwerken und die große Zahl von Bauteilvariationen (z.B. Wandverschneidungen) erschweren die Klassifizierung der Bauelemente nach besonderen Merkmalen und Eigenschaften und damit die Definition eines im Rechner abbildbaren Modells.

Ein Produktmodell (für das Bauwesen wird nach [Rüppel 96] auch die Bezeichnung *Bauwerksmodell* verwendet) definiert für alle Phasen der Planung, Erstellung, Nutzung, Instandhaltung, Umnutzung bis hin zur Entsorgung eine formale Beschreibung der Objekte (geometrische Beschreibung, Planungsdaten, numerische Ergebnisse uvm.) und deren Zusammenhänge und Abhängigkeiten [IAI 04]. Das Datenschema eines solchen Gesamtinformationsbestandes wird als *Produktdatenmodell* bezeichnet; eine konkrete Ausprägung heißt *spezifisches Produktmodell*. Die digitale Abbildung eines Bauwerks erfolgt dann durch abstrahierte Sichten der Fachplaner (Domänen). Diese als Domänen bezeichneten Teilmodelle repräsentieren einen bestimmten Aufgabenbereich eines Fachplaners. Jede Domäne definiert Begriffe und Bezeichnungen, die fachspezifische Semantik, deren formale Beschreibung das Domänenmodell bildet. Die Normierung des

Datenschemas sichert den fehlerfreien Datenaustausch zwischen den Domänenmodellen durch Verhinderung von Inkonsistenzen im Modell und Vermeidung von Redundanz durch wiederholte domänenspezifische Dateneingabe. Trotz aller Fortschritte in der Modellintegration bleibt es auf Grund der Komplexität der Modelle nach wie vor schwierig, einen ‚durchgängigen und verlustfreien Datenfluss‘ vom Architekten bis zur Fertigung zu erreichen.

Ein Vertreter von Produktdatenmodellen, der hauptsächlich in der Automobilindustrie eingesetzt wird, ist der Standard STEP¹⁰. Für das Bauwesen gibt es zwei Aspekte, die eine direkte Übernahme nicht ratsam erscheinen lassen:

- Die Vielzahl der domänenspezifischen Betrachtungsaspekte, die sich z.T. erheblich von der Sichtweise in der Automobilindustrie unterscheiden.
- Die mit fortschreitendem Detaillierungsgrad wachsende Komplexität der Beschreibung der Domänenmodelle.

Deshalb wurde mit den *Industry Foundation Classes (IFC)* für das Bauwesen eine branchenspezifische Lösung entwickelt. Bei der Entwicklung der IFC wurde auf die Erfahrungen zu STEP zurückgegriffen; damit ergeben sich gewissen Ähnlichkeiten zwischen beiden Spezifikationen. Deshalb werden zunächst die Grundzüge von STEP besprochen.

3.1.2 Der Produktmodell-Standard STEP – ISO 10303

STEP ist ein internationaler Standard (ISO 10303) für ein Produktdatenmodell, der die rechnerinterpretierbare Repräsentation und den Austausch von Produktmodelldaten über den gesamten Produktlebenszyklus hinweg ermöglichen soll [ISO 04].

Zur konsistenten und semantisch eindeutigen Beschreibung des Produktdatenmodells von STEP wurde die formale Beschreibungssprache EXPRESS¹¹ definiert. EXPRESS ist eine Spezifikationssprache, die objektorientierte und prozedurale Konzepte aus dem Bereich der Datenbanken vereinigt. EXPRESS spezifiziert eine Informationsdomäne in Form von Entities, die gemeinsame Eigenschaften besitzen. Diese Eigenschaften werden durch assoziierte Attribute und Randbedingungen (Constraints) repräsentiert. Die Entities sind jedoch reine Datenbeschreibungen, d.h. sie besitzen keine Methoden für den Zugriff auf die Daten.

STEP ist in eine Reihe von Teilspezifikationen gegliedert, die sog. Anwenderprotokolle (AP). Die in den Anwenderprotokollen definierten Informationsmodelle spezifizieren branchenspezifische Besonderheiten im Produktlebenszyklus. Für die Baubranche ist hier das AP 225 (Building Elements Using Explicit Shape Representation) zu nennen. Es erfasst sämtliche Informationen zur Darstellung komplexer räumlicher Gebäudemodelle vom Rohbau bis zum schlüsselfertigen Gebäude. Einen umfassenden Einblick dazu liefert [Anderl 00].

¹⁰ Standard for the Exchange of Product Model Data.

¹¹ Spezifikation zu EXPRESS siehe ISO 10303 Teil 21. Die graphische Variante ist in EXPRESS-G spezifiziert.

Das einzige STEP-Format, das in der Baupraxis Bedeutung erlangt hat, ist STEP-CDS, das bei der Fabrikplanung in der Automobilindustrie Anwendung findet. Hierfür hat die Firma WeltWeitBau [WWB 04] als Partner der CAD-Hersteller (Bentley, AutoDesk, Nemetschek und Graphisoft) die jeweiligen Konverter zu den proprietären Formaten entwickelt. Das Format STEP-CDS kann neben 2D-Geometrie zusätzlich die Zeichnungsstruktur (Modell, Layer, Gruppen, etc.) und Produktdaten (Kodierung, Version, Beschreibung, etc.), Bemaßung (explizit und assoziativ), Layoutangaben (Farbe, Fonts, Schraffur) und Referenzen zu externen Dokumenten beschreiben. Durch die Einbettung des Formats in die Familie der STEP-Standards ist die Beschreibung von Produktdaten mit STEP-CDS zudem für die langfristige Archivierung geeignet.

3.1.3 Der Produktmodell-Standard IFC

Die International Alliance for Interoperability (IAI) ist ein gemeinnütziger Zusammenschluss bedeutender Firmen und Organisationen aus dem weiteren Umfeld des internationalen Bauwesens. Das Anliegen der 1995 in den USA gegründeten Allianz war es in erster Linie, die unterschiedlichen Teilaspekte der AEC¹²/FM¹³-Industrie zu integrieren und in einem einheitlichen Datenmodell zusammenzuführen. Das gemeinsame Datenmodell IFC entspricht mit seinen Basisdaten weitgehend dem STEP-Standard und profitiert dadurch von den Erfahrungen der Automobilindustrie im Umgang mit Produktdatenmodellen. Nationale Gruppen (sog. Chapters) arbeiten an Weiterentwicklungen des Datenschemas, wie z.B. in den Projekten ST-4 (Structure Analysis Domain), ST-5 (Timber Constuction), BS-7 (Building Performance Monitoring), BS-8 (HVAC Extension). Von STEP wurde darüber hinaus die Beschreibungssprache EXPRESS und dessen grafische Variante zur Darstellung von Objektbeziehungen EXPRESS-G übernommen.

Die IFC spezifizieren ein semantisches Objektmodell, das neben der hierarchischen Gliederung des Bauwerks (Projekt, Gelände, Gebäude, Raum, Element) weitere zugeordnete Informationen vorhält:

- Die logische Struktur des Gebäudes.
- Zuordnung von Räumen zur Gebäudestruktur.
- Öffnungen in Bauelementen (Fenster-, Türöffnungen, Durchbrüche, etc.).
- Details zu Bauelementen (Türart, Öffnungsrichtung, etc.).
- Adjazente Verknüpfungen (von Bauelementen, etc.).
- Inzidente Verknüpfungen (von Bauelementen mit Strukturelementen).
- Attribute (eindeutige Objekt-Kennung, etc.).

¹² Architecture Engineering Construction.

¹³ Facilitiy Management.

Die IFC organisiert alle Elemente (auch Klassen genannt) in einem hierarchischen Schichtenmodell¹⁴ (siehe Abbildung 22), bei dem Elemente aus höheren Schichten (z.B. Architecture Domain) auf Elementen aus unteren Schichten aufbauen (z.B. Geometry Resource) (siehe dazu Abbildung 22). Die in der Tragwerksplanung üblichen ‚Entities‘, wie z.B. Wände, Stützen, Platten werden durch Objekte der allgemeineren Klasse¹⁵ *IfcProduct* zusammengefasst. Allgemeine Eigenschaften wie Standort (*IfcLocalPlacement*) oder die Darstellung (*IfcProductRepresentation*) werden durch spezialisierte Klassen verwaltet. Die Gliederung der Objektdaten zur geometrischen Beschreibung wird dabei bis zum kartesischen Punkt (*IfcCartesianPoint*) aus dem *Geometry Resource Layer* fortgesetzt.

Die Strukturierung der Elemente hilft der Weiterentwicklung, Dokumentation und Normierung der Spezifikation. Eine Untermenge von zentralen Elementen aus der IFC (hauptsächlich aus *Resource Layer*, *Kernel Layer* und *Core Extension*) wurden als Teil der so genannten *IFC 2x Plattform* besonders hervorgehoben und im Jahre 2002 als ISO/PAS 16739 registriert. Ziel war die Schaffung eines über einen längeren Zeitraum unveränderlichen Kern als Teil der IFC, der der Software-Industrie ausreichend zeitlichen Spielraum zur Implementierung geben sollte [IAI 04]. Praxistests zum Datenaustausch reeller Bauwerke zwischen CAD-Systemen führender Softwarehäuser (Nemetschek, AutoDesk, Graphisoft) bescheinigen die Tauglichkeit der verfügbaren IFC-Prozessoren [Geiger 01]. Manche Hersteller bieten lediglich Schnittstellen zu Teilmodellen der *IFC 2x Plattform* an. Diese bauen auf sog. *Views* auf, die einen Teil des Modells ausblenden und damit die Komplexität der zu implementierenden Spezifikation reduzieren. Im Vordergrund stehen dabei meist geometriebeschreibende Elemente [ISG Steinmann].

¹⁴ Elemente des Schichtenmodell der IFC sind in die Layer: Resource Layer, Core Layer, Core Interoperability Layer und Domain Layer gegliedert.

¹⁵ Basisklasse im Sinne des objektorientierten Designs.

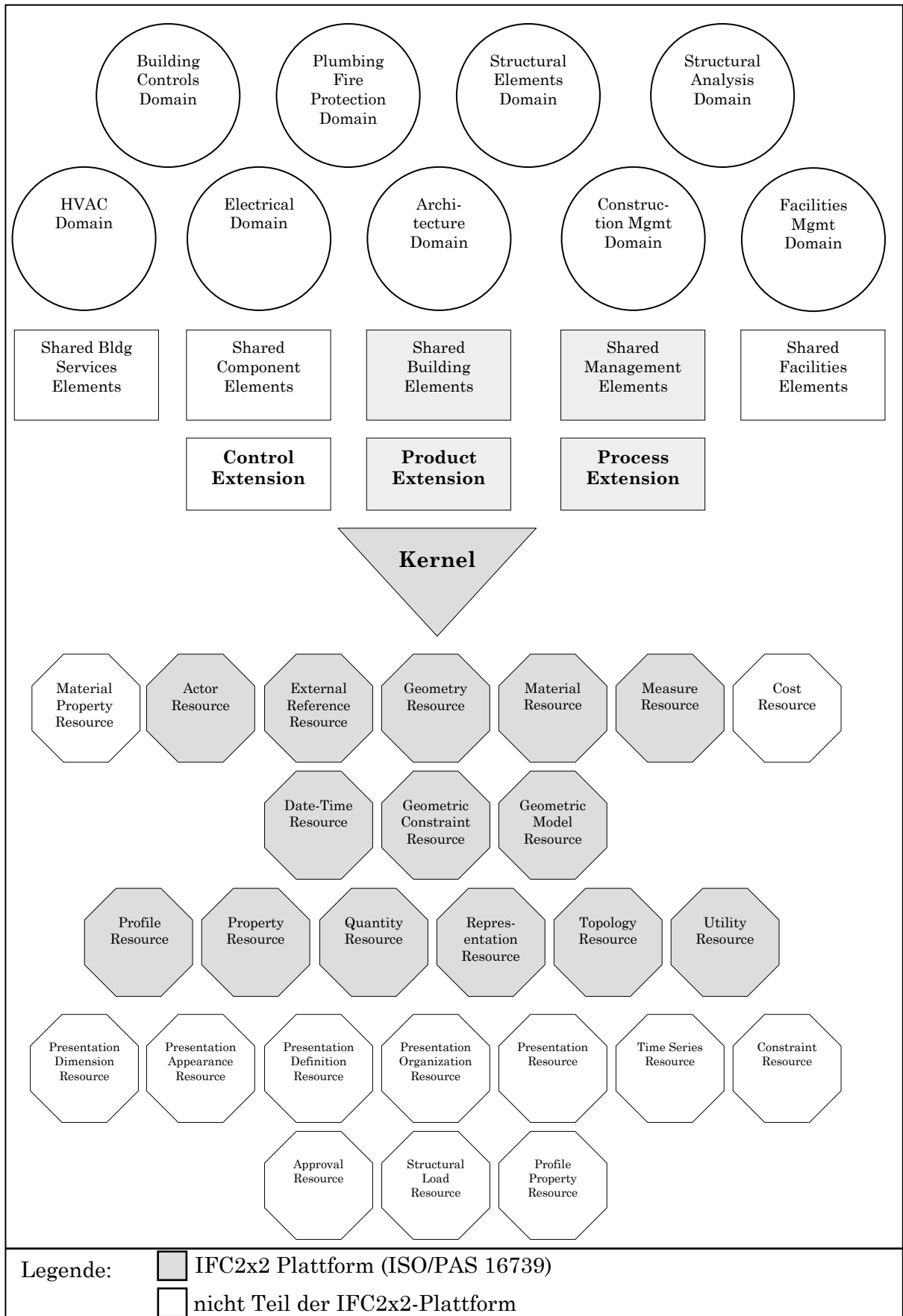


Abbildung 22: Strukturierung der IFC-Module nach der Spezifikation 2x2 [IAI 04].

3.1.4 IFC Erweiterung für die Tragwerksplanung (ST-4)

Für die vorliegende Arbeit hat die bereits oben genannte IFC-Erweiterung ST-4 besondere Bedeutung. Ziel des vom deutschen ‚Chapter‘ geleiteten Arbeitskreises *Tragwerksplanung* ist es, die IFC-Spezifikation um die Fähigkeit zu erweitern, Modelldaten für die Tragwerksplanung rechnergestützt und domänenübergreifend zwischen Architektur und Tragwerksplanung oder innerhalb der Domänen Tragwerksplanung zwischen Planer und Prüfer auszutauschen [IAI ST-4]. In Abgrenzung zur vorliegenden Arbeit, die auf durchgängig dreidimensionalen Strukturen basiert, wird im Rahmen von ST-4 mit dimensionsreduzierten mechanischen Modellen gearbeitet (Mittellebene bei Scheiben, etc.). Zusatzinformationen wie Diskretisierung der Geometrie für eine FE-Berechnung sind nicht Bestandteil des Erweiterungsvorschlages. Hinsichtlich der geometrischen Beschreibung der Bauelemente gibt es die Beschränkung auf ‚einfache‘ Systeme (z.B. keine Schalen, keine mit NURBs berandeten Bauelemente). Die über die mechanische Beschreibung hinausgehenden Informationen müssen damit weiter in den entsprechenden Anwendungen integriert werden. Die Problematik der Dimensionsreduzierung von 3D-Elementen zu 2D-, oder 1D-Systemen bleibt dabei bestehen (siehe dazu auch [Rüppel 96]). Der Nutzen des Erweiterungsvorschlages für die vorliegende Arbeit liegt in der Normierung der mit dem mechanischen Modell verknüpften Daten wie:

- Lasten sowie Lastfälle
- mechanische Bauteilkopplungen
- Lagerungsbedingungen.

Die Spezifikation ST-4 kann aktuell noch nicht eingesetzt werden, da die in der finalen Fassung vorliegende Erweiterung noch durch die IAI in den IFC-Standard aufgenommen werden muss. Darüber hinaus fehlt noch die Erweiterung des IFC-Processors, der in dieser Arbeit eingesetzt wird [IFC Toolbox]. Nach der Integration der Erweiterung ST-4 in Norm und Implementierung kann das proprietäre Datenschema der vorliegenden Arbeit ersetzt werden.

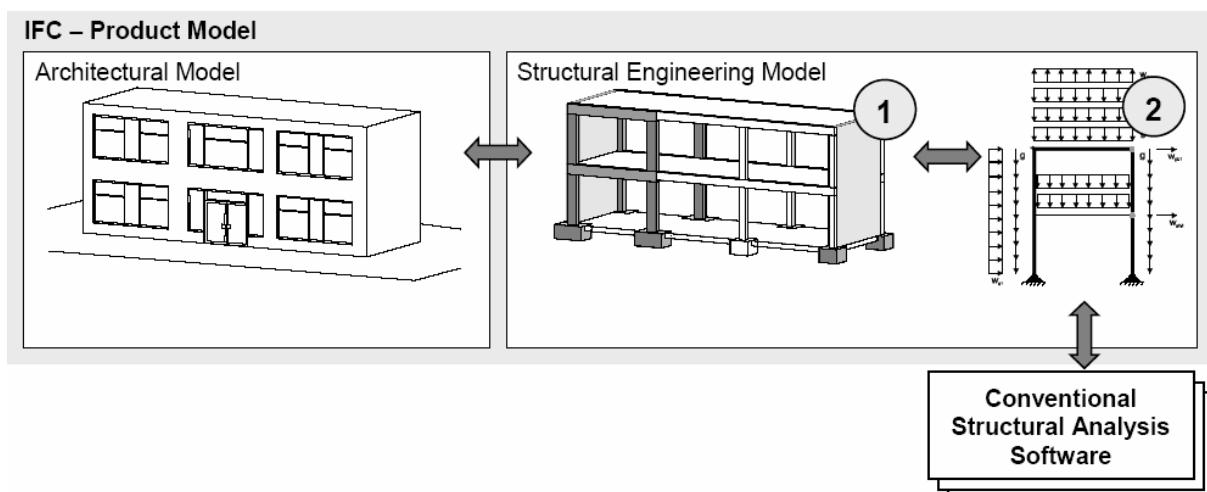


Abbildung 23: Erweiterung des Architekturmodells um Angaben zum mechanischen Modell [Weise 02].

3.2 Überführung von Produktmodellldaten in ein erweitertes Volumenmodell

Konventionen: Zur Abgrenzung der Mengen-Operationen Schnittmenge, Vereinigung und Differenz zu topologischen Unterelementen werden erstere mit einem ‚b‘ indiziert als $\cap_b, \cup_b, \setminus_b$ und letztere mit einem ‚t‘ indiziert als \in_t geschrieben.

3.2.1 Einführung

Die Erzeugung geometrischer/semantischer Beschreibungen für Bauwerke kann mit diversen Tools durchgeführt werden. Diese lassen sich in zwei Gruppen einteilen: Das sind zum einen Tools, die reine Geometrie-Modelle mittels Formaten wie SAT, IGES, etc. exportieren (z.B. AutoCAD von AutoDesk, CATIA von Dessault Systems) und zum anderen Tools, die den Datenaustausch über die IFC-Schnittstelle zur Verfügung stellen (z.B. ADT von AutoDesk, ArchiCAD von Graphisoft, Allplan FT von Nemetschek).

Der Vorteil der ersten Gruppe liegt im meist mächtigeren geometrischen Modell der Austausch-Formate (z.B. Unterstützung von Freiformflächen). Der Nachteil dabei ist, dass nur eingeschränkt semantische Informationen ausgetauscht werden können (z.B. keine Nachbarschaftsrelationen, keine Typisierung der Elemente). Die IFC-Schnittstelle schafft zwar den Datenaustausch geometrischer wie semantischer Informationen, definiert aber ein eingeschränktes geometrisches Modell (i.d.R. Produktionsmodelle). Der Import bei der im Rahmen dieser Arbeit entwickelten Software erfolgt deshalb, wo notwendig, über entsprechende Konverter (siehe Abbildung 24). Der IGES/SAT-Konverter wird hier nicht explizit beschrieben, da er Teil der ACIS-Distribution ist und transparent über die Datei-Lade-Operation eingesetzt werden kann. Der IFC/SAT-Konverter ist ein wichtiges Werkzeug für diese Arbeit und wird im nachfolgenden beschrieben.

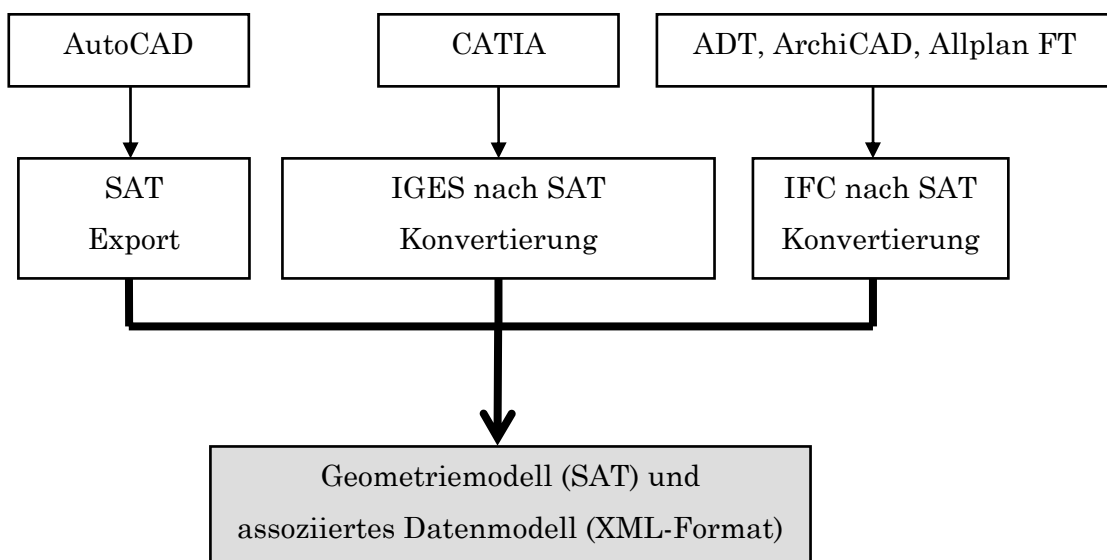


Abbildung 24: Import der geometrischen/semantischen Gebäudedaten.

Die in Abbildung 24 aufgezeigten Import-Pfade ermöglichen die Nutzung vieler CAD-Systeme mit ihren spezifischen Besonderheiten hinsichtlich Benutzerinterface (Einarbeitungsaufwand, Bedienbarkeit) und der Mächtigkeit des verwendeten geometrischen Modells.

Beim Import-Pfad von Bauwerksmodellen über die IFC-Schnittstelle steht neben den semantischen Daten die geometrische Beschreibung besonders im Vordergrund. Die verschiedenen geometrischen Modelle, die sich in der IFC Anwendung finden (Produktionsmodelle, Halbraummodelle, CSG, ...), werden bei der Konvertierung in ein ausgewertetes Modell (siehe Kapitel 2.2 'Grundlagen der geometrischen Modellierung') übertragen. Der Grund ist, dass nur im ausgewerteten Modell Knoten, Kanten, Flächen und Volumen explizit modelliert sind und für eine dauerhafte (persistente) Attributierung zur Verfügung stehen. Die Möglichkeit zur Attributierung von Elementen (Kanten, Flächen, Körper) des geometrischen Modells ist ein wichtiger Aspekt im vorgestellten Verfahren. Nähere Erläuterungen zur Attributierung des geometrischen Modells liefert Kapitel 3.2.4.5 'Attributierung des geometrischen Modells'.

Die Datenhaltung des hier beschriebenen Modells basiert auf zwei zentralen Datenstrukturen. Zum einen ist das die Datenstruktur des ACIS-Geometriekerns für die Speicherung und Manipulation geometrischer Modelle [ACIS 02] und zum anderen die hierarchisch organisierte Datenstruktur von XML¹⁶ zur Speicherung zusätzlicher Daten (Simulationsparameter, Objekteigenschaften, Modellstrukturierung mittels Layering, etc.). Die Verwendung eindeutiger Objekt-IDs ermöglicht die eindeutige Zuordnung (1:1 Beziehung) von Körpern (BODIES) im geometrischen Modell zu Strukturknoten in der XML-Struktur (im weiteren Verlauf der Arbeit wird die XML-Struktur als Datenbank bezeichnet). Die Objektstruktur der Datenbank ist in Abbildung 25 gezeigt.

Die Besonderheiten des ACIS-Geometriekerns sind im nachfolgenden Kapitel beschrieben. Die Datenstruktur zum XML-Format wird mit den Klassen der QT-Bibliothek [Herold 01] verwaltet. Beide Strukturen zusammen ergeben einen leistungsfähigen Rahmen, der in dieser Arbeit genutzt wird.

¹⁶ EXtended Modelling Language. Siehe dazu [Goldfarb 00]. Zur Abgrenzung von HTML, das wie XML auf SGML basiert, sein darauf hingewiesen, das HTML hauptsächlich das Aussehen der Daten beschreibt, während XML mitteilt, was die Daten bedeuten.

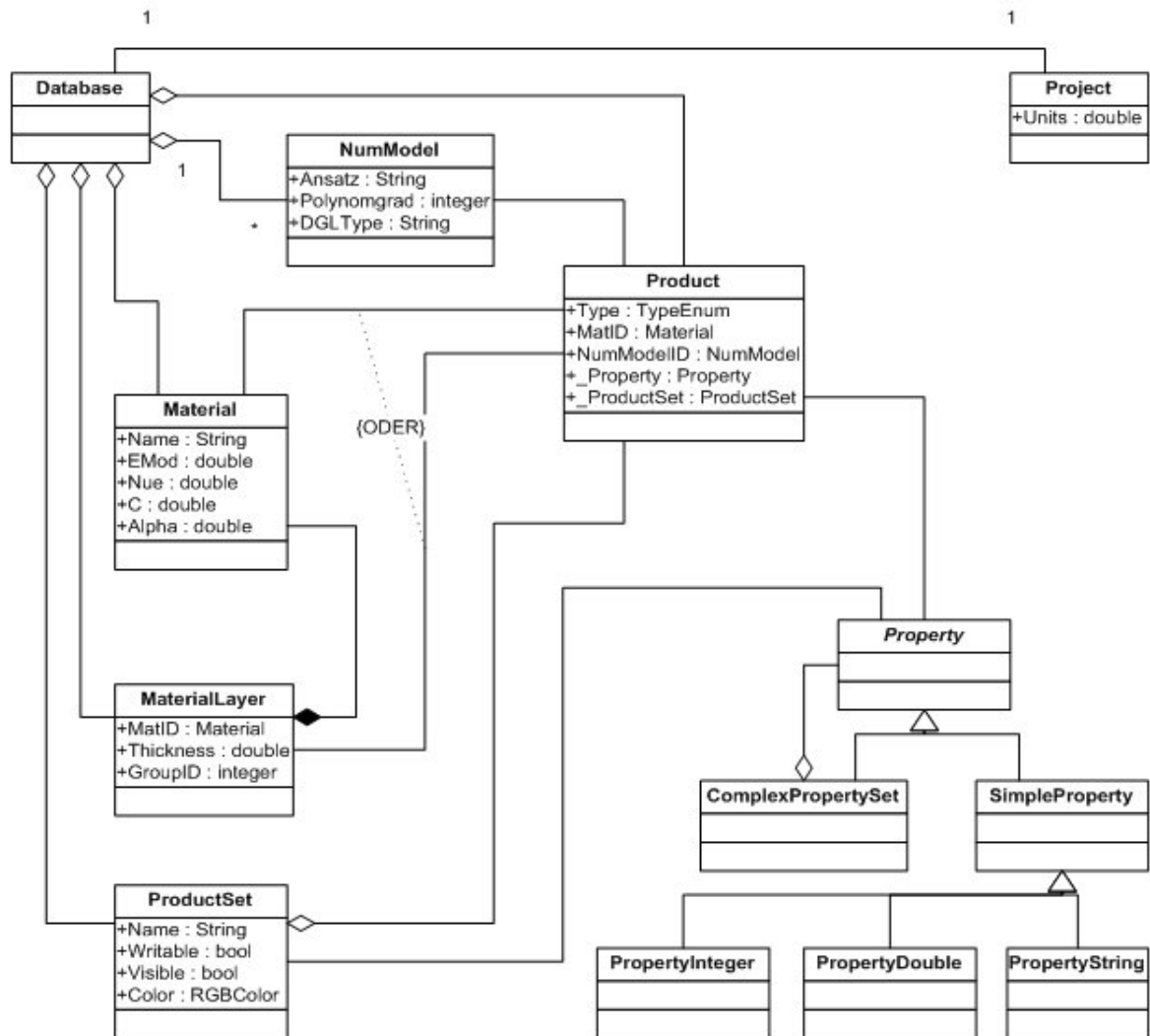


Abbildung 25: Das statische Klassenmodell der assoziierten Datenbank.

Nachfolgende Erläuterung behandelt in knapper Form zwei Besonderheiten im statische Klassenmodell aus Abbildung 25. Ein *Produkt* (z.B. Wand) kann direkt einen *Material*-Eintrag referenzieren (nur eine Materialschicht über die Wanddicke) oder indirekt über ein Objekt der Klasse *MaterialLayer* falls dem *Produkt* eine Wandschichtung zugeordnet wird (z.B. Innenputz, Ziegelwand, Außenputz). Die Orientierung der Schichtung hinsichtlich des zugeordneten geometrischen Objekts (Wand) wird über den Schichtungsvektor gesteuert. Dieser wird aus der IFC-Beschreibung übernommen (siehe Kapitel 3.2.3). Für das oben genannte Beispiel mit Innenputz, Ziegelwand und Außenputz zeigt der Schichtungsvektor von der Wandinnenseite zur Wandaußenseite. Erwähnenswert ist weiterhin die Gruppe der *Property*-Klassen. Die Struktur entspricht dem bekannten ‚Composite‘ Pattern [Gamma 98]. *Properties* können neben Objekten der Klasse *Product* auch Kollektionen von *Product* (z.B. alle Wände im Modell) zugeordnet werden. Damit lassen sich Attributierung auf Mengen von Objekten anwenden (z.B. alle Stützen bekommen einen Materialeintrag mit Betongüte C35).

3.2.2 Der geometrische Modellierkern ACIS

Zur Erzeugung, Verwaltung und Manipulation geometrischer Daten wird sinnvollerweise eine Software-Bibliothek zur Volumenmodellierung eingesetzt. Für die vorliegende Arbeit standen zwei Modellierer zur Auswahl:

- ACIS von Spatial Corporation [ACIS 02].
- OpenCascade von EADS Matra Datavision [OpenCascade 04].

Die Wahl fiel auf ACIS. Der Grund waren Geschwindigkeitsvorteile von ACIS bei Booleschen Mengen-Operationen, die bei diesem Projekt häufig eingesetzt werden. ACIS ist ein BRep Modellierer, der die Verwaltung von Draht-, Flächen- und Volumenmodellen in ein und derselben Programmumgebung erlaubt. Die verschiedenen Modelle sind in der Datenstruktur koexistent. Darüber hinaus können neben mannigfacher Geometrie (manifold) auch nicht-mannigfache Geometrie (non manifold) verwaltet und bearbeitet werden. ACIS ist in der Programmiersprache C++ [Stroustrup 00] entwickelt und wird in Form dynamischer Bibliotheken (Dlls) ausgeliefert. Damit ist es möglich, eigene leistungsfähige Anwendungen zu entwickeln. Alle Angaben in diesem Kapitel basieren auf [Corney 97].

Zur Anwendung von ACIS stehen zwei verschiedene Zugriffsmöglichkeiten zur Verfügung. Dies ist zum einen das *Application Procedural Interface* (API) und zum anderen das *Direct Object Interface* (DI). Das *DI* bietet den Zugriff auf alle internen Klassen des Geometriemodellierers (z.B. BODY, LUMP, SHELL, FACE, COEDGE, EDGE, ...) und erlaubt es darüber hinaus bei Bedarf, eigene spezialisiertere Klassen abzuleiten. Das *API* definiert höherwertige Operationen, die sich unkompliziert und ohne genauere Kenntnis der internen Datenstrukturen anwenden lassen. Ein Beispiel ist die Funktion *api_copy_body(BODY* b1, const BODY* b2)*. Diese Funktion dupliziert einen Body *b2 auf einen Body *b1. Dabei werden intern alle topologischen und geometrischen Objekte bearbeitet, die in der Beschreibung des Body *b2 verfügbar sind. Mit dem API-Interface werden viele regelmäßig wiederkehrende Aufgaben wie Erzeugung, Manipulation, Verknüpfung und Evaluierung geometrischer Objekte behandelt. Einen Überblick über die topologischen und geometrischen Klassen in ACIS gibt Abbildung 26.

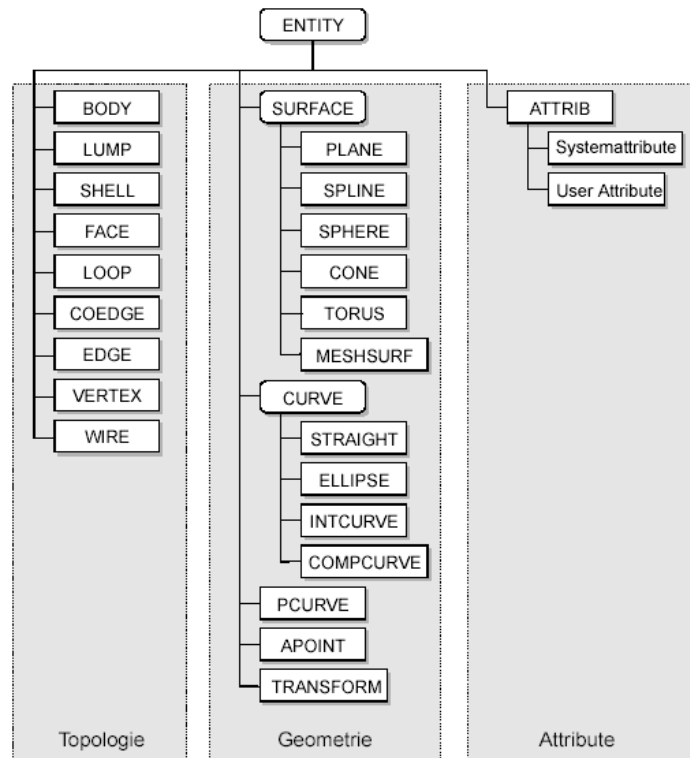


Abbildung 26: Klassendiagramm von ACIS [Corney 97].

Wie aus der obigen Abbildung ersichtlich ist, sind die Klassen zur Verwaltung von Geometrie, Topologie und Attribut von der abstrakten Klasse ENTITY abgeleitet. Die Verwaltung von Objekten wird über die Kollektions-Klasse ENTITY_LIST vorgenommen. Das hat den Vorteil, dass damit alle von ENTITY abgeleiteten Klassen verwaltet werden können. Weiterhin bietet die Klasse ENTITY den Zugang zur Speicherverwaltung sowie zur Undo/Redo bzw. Rollback-/Rollforward-Funktionalität.

Geometrische Klassen

Die Klassen APOINT, CURVE, SURFACE und PCURVE bilden die Gruppe der sog. *abstrakten Geometrie*. Die Klassen enthalten keine mathematische Beschreibung der Geometrie, sondern werden zur Verwaltung der Objekte der *konstruktiven Geometrie* verwendet. Die Klassen der Gruppe *konstruktive Geometrie* definieren dagegen mathematische Beschreibungen für allgemeine (unbegrenzte) Kurven und Flächen. Die Klasse *spline* fasst die Modellierung von Freiformflächen zusammen. Zur Unterscheidung der Klassen der beiden Gruppen haben die konstruktiven Klassen einen kleingeschriebenen Namen und die Klassen der *abstrakten Geometrie* einen Namen in Großbuchstaben (siehe Abbildung 27). Die endliche Ausdehnung erhalten die geometrischen Beschreibungen durch die Berandung mit Objekten niedrigerer Dimension ($d-1$). Die SURFACE wird durch CURVES, die CURVE durch APOINTS begrenzt.

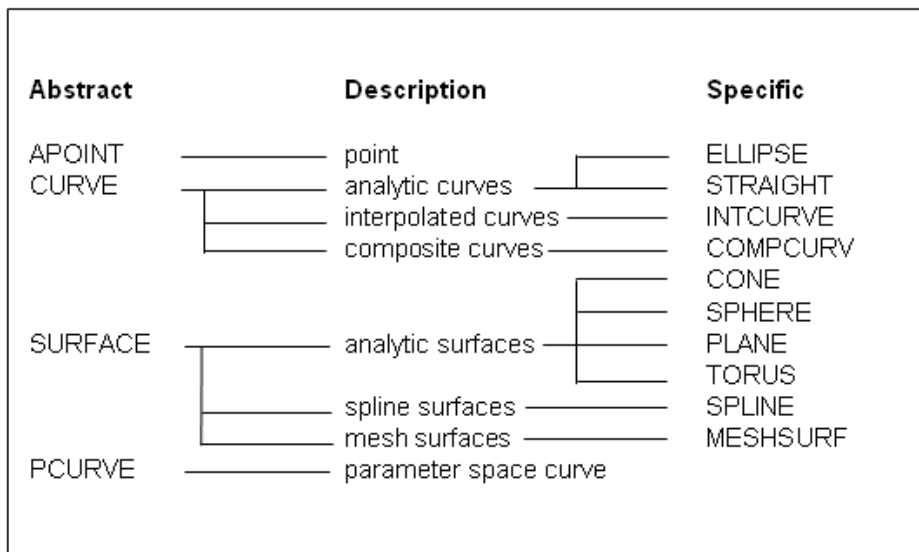


Abbildung 27: Zusammenhang zwischen den Elementen der konstruktiven und abstrakten Geometrie in ACIS [Corney 97].

Topologische-Klassen

Die Topologie bildet das ‚Rückgrad‘ der BRep Datenstruktur (siehe Kapitel 2.2.6). Das grundlegende Objekt (BODY) kann eine Beschreibung eines Draht-, Flächen-, Volumenmodelles oder auch Kombinationen davon enthalten. Dabei kann die *Boundary Representation* vollständig (damit ein Körper) oder unvollständig (kein Körper) sein.

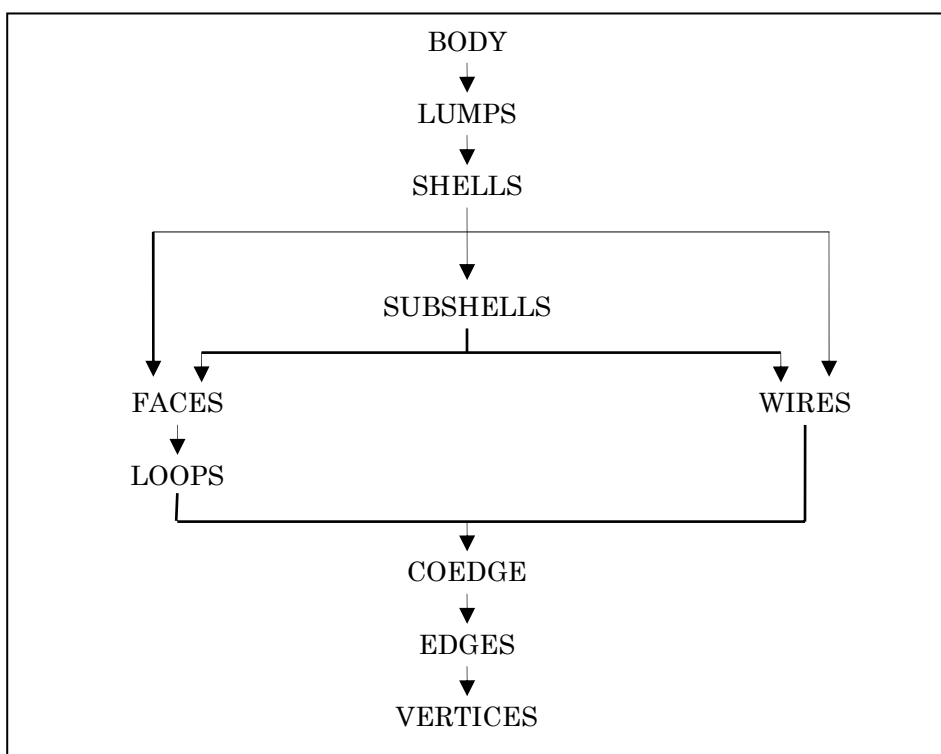


Abbildung 28: Schematischer Aufbau topologischer Elemente in ACIS [Corney 97].

Das geometrische Objekt wird i.d.R. durch ein Objekt der Klasse BODY repräsentiert. Damit startet auch der Zugriff auf weitere untergeordnete Datenstrukturen. Ein BODY setzt sich aus (1..*) LUMPS zusammen, die sog. logisch zusammenhängenden Teilkörper. Die Oberfläche eines LUMPS wird über FACES beschrieben, die entweder direkt über SHELLS verwaltet werden oder indirekt über das Gruppierungsobjekt SUBSHELLS. Alternativ beschreibt eine SHELL bzw. SUBSHELL ein Drahtmodell über ein WIRE. FACES werden durch LOOPS von COEDGES beschrieben, die sich bei zwei-mannigfachen Körpern ein EDGE teilen. Die VERTICES bilden die letzte Gruppe der Hierarchie und beranden ein EDGE (siehe dazu Abbildung 28).

Zur leichteren Orientierung wird im nachfolgenden Bild die topologische Struktur anhand eines Würfels verdeutlicht.

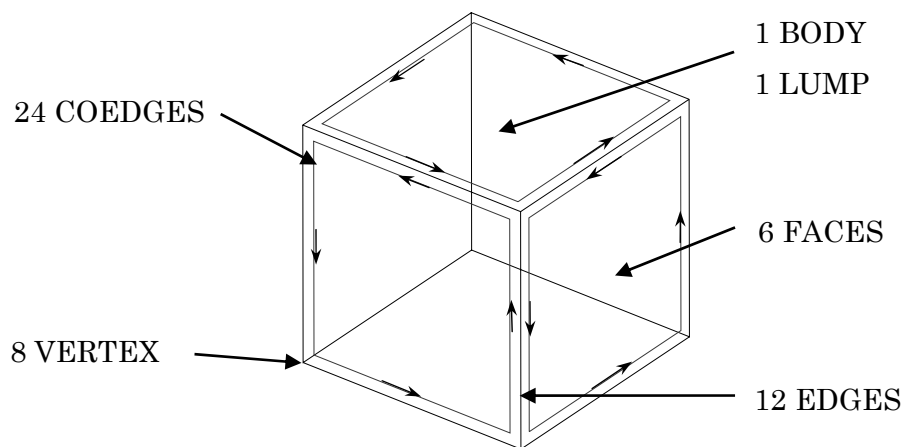


Abbildung 29: Topologische Struktur eines Würfels in ACIS.

Attributierung geometrischer Objekte

Alle von der Klasse ENTITY abgeleiteten Klassen stellen Funktionalitäten zur Verwaltung externer Daten¹⁷ zur Verfügung. Die Verwaltung von Attributen umfasst auch die flexible Festlegung zur Behandlung von assoziierten Daten bei Manipulationen am geometrischen Modell, z.B. Teilung eines Volumenkörpers. Dabei lässt sich für alle wesentlichen Operationen die Vererbungsstrategie feingranular festlegen. Für die vorliegende Arbeit ist diese Fähigkeit des ACIS-Kerns von großer Bedeutung. Nähere Informationen zur Attributierung sind im Kapitel 3.2.4.5 nachzulesen.

¹⁷ Dies können Werte vom Typ INTEGER (32-Bit Ganzzahlwert), REAL (64 Bit Gleitkommawert) und STRING (Zeichenarray der Sprache C) oder ACIS-eigene Typen wie POSITION, VECTOR, POINTER sein.

3.2.3 Einlesen der IFC-Produktmodellldaten

Das Einlesen von IFC-Produktmodellldaten wird in der vorliegenden Arbeit mittels der in C++ programmierten *IFC Toolbox* der Firma *Eurostep Group* vorgenommen [IFC Toolbox]. Mit dem Einsatz der Toolbox wird der hohe Aufwand reduziert, der sonst für das Auslesen relevanter Informationen aus der IFC-Datei notwendig wäre. Die Toolbox bildet sämtliche EXPRESS Datentypen auf entsprechende C++-Klassen mit verschiedensten Zugriffsfunktionen für Lese- und Schreiboperationen ab. Zur Verwaltung von Objekt-Mengen stehen verschiedene Kollektionsklassen zur Verfügung.

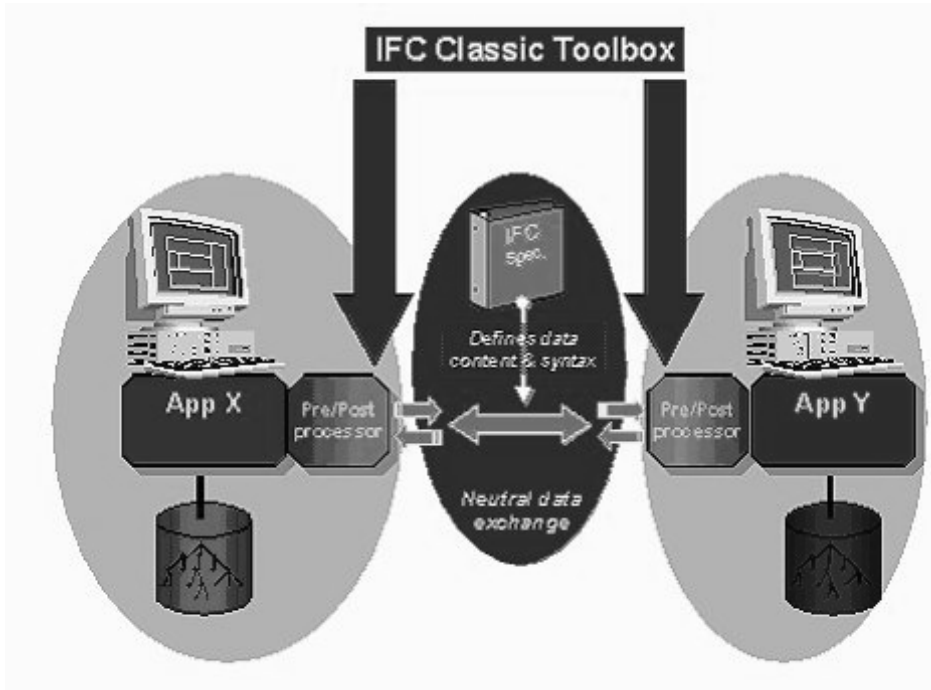


Abbildung 30: Schematisch dargestellter Einsatz der Toolbox beim applikationsübergreifenden Datenaustausch.

Die zwei nachfolgenden Absätze ‚Bezeichnungen in der IFC‘ und ‚Mapping von EXPRESS-Datentypen in C++-Klassen‘ wurden für ein besseres Verständnis des Pseudocodes auf Seite 44 eingefügt (in gekürztem Umfang).

Bezeichnungen in der IFC-Toolbox

Die IFC definiert verschiedene Zusätze zu Bezeichnungen, um Benennungen eindeutig zu halten und um die semantischen Einordnung der bezeichneten Elemente auszudrücken. In der IFC-Toolbox werden diese Bezeichnungen übernommen und weitere hinzugefügt (z.B. `_Hnd`). Die folgende Auflistung zeigt die wichtigsten Präfix und Suffix Definitionen:

- **Ifc** (Präfix): Diese Vorsilbe erhalten alle Klassen, Enumerationen, Select-Types und Funktionen. Diese Konvention vermeidet Namenskonflikte beim Hinzulinken der Toolbox zu anderen Softwaremodulen.

- **_Hnd** (Suffix): Der Zugriff auf ein *Entity* erfolgt nicht direkt, sondern über eine assoziierte Handle-Klasse (Stellvertreter), die Lese- bzw. Schreibfunktionen implementiert. Die Handle-Klassen erhalten die Nachsilbe *_Hnd*. Zum *Entity IfcBuildingStory* korrespondiert demnach die Handle-Klasse *IfcBuildingStory_Hnd*.

Mapping von EXPRESS-Datentypen in C++-Klassen

- **Types** sind Basisdatentypen wie *STRING*, *INT*, *REAL*. Diese werden auf C++-Klassen abgebildet. Ein *INT* wird dabei zu der Klasse *IfcInt*. Die C++-Klassen sind allesamt Wrapperklassen¹⁸ mit internen Variablen und eine Reihe von Funktionen für Zugriff-, Konvertierung- und Vergleichsoperationen.
- **Aggregationen** liefern Objekt-Sammlungen und stehen in vier Ausprägungen zu Verfügung: *Bag*, *Set*, *List* und *Array*. Während in *Bags* Objekte mehrfach vorkommen dürfen, sind Objekte in *Sets* eindeutig. Der Zugriff auf Listen erfolgt sequenziell, der auf Arrays wahlfrei. Die Toolbox stellt diese Objekt-Sammlungen über Template-Klassen *IfcBag<T>*, *IfcSet<T>*, *IfcList<T>* und *IfcArray<T>* zur Verfügung. Weiterführende Informationen zu Speicherstrukturen sind in [Sedgewick 02] nachzulesen.

Um den Einsatz der IFC-Toolbox zu verdeutlichen, wird nachfolgend ein Pseudocode-Fragment dargestellt. Darin sind am Funktionsanfang verschiedene *Handles* bzw. ein Kollektionsobjekt definiert, z.B. das *Handle hPresentation* vom Typ *IfcProductRepresentation_Hnd*. In der Zeile vor der Schleife wird die Kollektion *bagGeom* mit allen *GeometricRepresentationItems* der Wand *w* gefüllt. Grund für die Verwendung einer Liste ist der Umstand, dass ein Produkt (z.B. Wand) über verschiedene geometrische Modelle beschrieben werden kann, z.B. über ein Extrusionsmodell (*SweptAreaSolid*) oder über den kleinsten umhüllenden, achsenparallelen Quader (*BoundingBox*). Die verschiedenen geometrischen Repräsentationen werden innerhalb der Schleife abgeprüft und entsprechend geändert. Aufgabe in dem dargestellten Beispiel ist es, die Höhe der Wand (*IfcWall*) um 0.5 Einheiten zu erhöhen (Die Maßeinheit wird hier nicht berücksichtigt). Das Beispiel ist [Kühner 01] entnommen. Eine detaillierte Erläuterung zur geometrischen Beschreibung von Objekten in der IFC wird in Kapitel 3.2.4 vorgenommen.

¹⁸ Eine Wrapper-Klasse definiert eine Hülle um ein bestehendes Datum oder um eine bestehende Implementierung, um objektorientierte Methoden auf nicht objektorientierte Daten oder Implementierungen anzuwenden (z.B. 'Information Hiding') oder um höhere Abstraktionsstufen in der Implementierung einzuführen.

Pseudocode 1: Änderung der Höhe einer IFC-Wand (IfcWall)

Beginn Funktion changeWall(Referenz IfcWall w)

<u>Variable:</u>	<u>Type:</u>
hPresentation	IfcProductRepresentation_Hnd
bagGeom	IfcBag<IfcGeometricRepresentationItem_Hnd>
hItem	IfcGeometricRepresentationItem_Hnd
hSolid	IfcExtrudedAreaSolid_Hnd
hBox	IfcBoundingBox_Hnd

hPresentation = IfcProductRepresentation_Hnd aus w
 bagGeom = alle IfcGeometricRepresentationItem_Hnd aus Element hPresentation

Schleife über alle Elemente e1 in bagGeom

 hItem = nächstes Elemente aus e1

Wenn hItem == BoundingBox

 hBox = hItem

 hBox.ZDim = zBox.ZDim + 0.5

sonst wenn hItem == SweptAreaSolid

 hSolid = hItem

 hSolid.ExtrudedDirection = hSolid.ExtrudedDirection + 0.5

Ende Wenn

Ende Schleife

Ende Funktion

Im nachfolgenden Kapitel wird beschrieben, wie die Konvertierung der IFC-Produktmodellldaten mit Hilfe der IFC-Toolbox in ein ausgewertetes Volumenmodell in ACIS durchgeführt wird.

3.2.4 Auswerten geometrischer/semantischer Modelldaten

3.2.4.1 Die grundlegenden Klassen der IFC

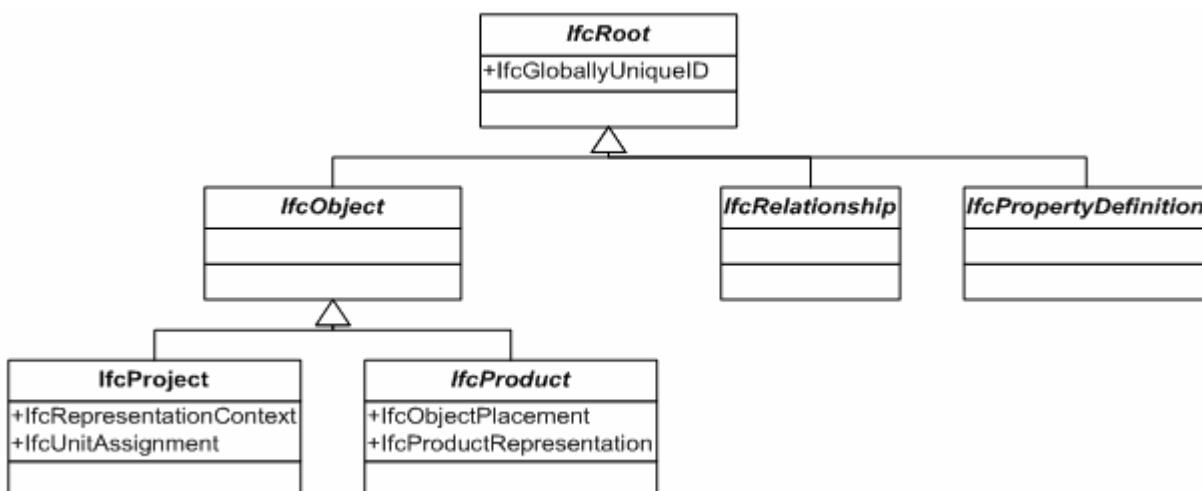


Abbildung 31: Grundlegende Klassen der IFC.

Die Klasse IFC-Root

Die abstrakte Klasse ist Basisklasse aller Entities. Sie definiert über das Attribut *IfcGloballyUniqueID* eine weltweit eindeutige ID (GUID). Im Original ist die GUID eine 128-bit Folge (32 Zeichen). Aus Gründen der Platzersparnis definiert die IFC eine komprimierte Variante mit 20 Zeichen und mit folgender Basis:

0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ!#\$%&^|*+,-./:;<=>?~`@_.

Die Basis wurde mit der IFC2x nochmals verändert und auf 64 Zeichen begrenzt. Grund waren Kompatibilitätsforderungen zum ISO 10303-21 Standard und zur Element-Definition der W3C hinsichtlich des XML-Formats. Damit gilt für die *IfcGloballyUniqueID* folgende Basis:

0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ!#\$%&^|*+,-./:;<=>?~`@_.\$

Die Beschreibung *IFCWALLSTANDARDCASE('1fbk5Ikor6CQSXNW6_VCC6',#16,"","",#1568,#1577,\$)* aus einer IFC-Datei enthält die *IfcGloballyUniqueID* beispielsweise im ersten Parameter.

Die Klasse IfcRelationship

IfcRelationship ist die abstrakte Basisklasse aller Objekt-Relationen. Es gibt zwei Arten von Relationen: Die 1-zu-1 Relationen und die 1-zu-viele Relationen. Im Falle der 2-stelligen Relationen werden die Namenskonventionen

- Relating+<Objektname>.
- Related+<Objektname>.

verwendet. Im Falle der 1-zu-viele Relation wird die Aggregationsklasse *IfcSet* verwendet.

Die Klasse IfcPropertyDefinition

Diese Klasse definiert Werte, die von mehreren Objekt-Instanzen referenziert werden (z.B. Material). Es gibt einfache Wert-Definitionen (simple Property) oder Type-Definitionen, die für sich wieder aus untergeordneten Wert-/Type Definitionen bestehen können (komplex Property). Mit dieser Struktur lassen sich nahezu alle denkbaren Informationen zu Instanzen von *IfcObject* assoziieren. Diese Werte werden in der vorliegenden Arbeit gelesen und in die zum ACIS-Modell assoziierte Datenbank beschrieben. In beiden Modellen (IFC und XML) wird dazu eine hierarchische Baumstruktur verwendet, die auf der Seite der Datenbank so organisiert ist, dass die Konvertierung der Elemente über eine 1-zu-1 Abbildung durchgeführt werden kann.

Die Klasse IfcProject

Die Klasse *IfcProject* definiert den Kontext für die Repräsentation geometrischer Objekte (*IfcRepresentationContext* und *IfcUnitAssignment*). Darin finden sich folgende Informationen:

- Maßeinheiten.
- Das Welt-Koordinaten-System.
- Die Dimension der Koordinaten im Raum (2D oder 3D).

- Die Genauigkeit der geometrischen Repräsentation.
- Die Nordrichtung.

Die Klasse IfcProduct

Die Klasse steht für alle Objekte, die Lage im Raum (*IfcObjectPlacement*) und Gestalt (*IfcProductRepresentation*) haben. Beide Elemente haben bezüglich ihrer Aggregatklasse die Kardinalität (0..1).

3.2.4.2 Die Lage geometrischer Objekte im Raum

Zur korrekten Lagebestimmung geometrischer Objekte im Projektkoordinatensystem (ACIS-Modell) werden bis zu vier Transformationen (Rotation, Translation) erforderlich. Jedes geometrische Objekt definiert ein lokales Koordinatensystem (z.B. Wand), dieses kann relativ zu einem anderen, übergeordneten Objekt positioniert sein (z.B. Fenster zu Wand). Die Lage der Bauteilkoordinatensysteme wird durch das Koordinatensystem des Gebäudes beschrieben. Zuletzt definiert das Projekt ein weiteres globales Koordinatensystem, das Projektkoordinatensystem.

Elementar für die Lage und die Orientierung von Objekten im Raum sind dabei die Klassen *IfcCartesianPoint* und *IfcDirection*. Beide Klassen werden je nach Anzahl der verwendeten Raumdimensionen von *IfcAxis2Placement3D* und *IfcAxis2Placement2D* aggregiert.

Die Bedeutung der Klassenattribute *Axis* und *RefDirection* sind in Abbildung 32 dargestellt. Der Ursprung des lokalen Koordinatensystems wird durch das von *IfcPlacement* geerbte Attribut *Location* (siehe Abbildung 38) definiert. Im Falle der Klasse *IfcAxis2Placement3D* erfolgt die Orientierung des Koordinatensystems durch (idealerweise) zwei zueinander senkrecht stehende Vektoren. Wird der x-Vektor durch den Vektor *RefDirection* lediglich approximiert (*RefDirection* und P[1] sind nicht parallel), erfolgt die Definition von P[1] durch eine Projektion von *RefDirection* auf eine Ebene. Diese Ebene geht durch den Koordinatenursprung und liegt normal zu *Axis*. Der Vektor P[2] wird dann aus den Vektoren P[3] (*Axis*) und P[1] generiert, so dass die drei Vektoren P[1], P[2], P[3] ein Rechtssystem bilden. Die Konstruktion des lokalen Koordinatensystems im zweidimensionalen Raum erfolgt entsprechend.

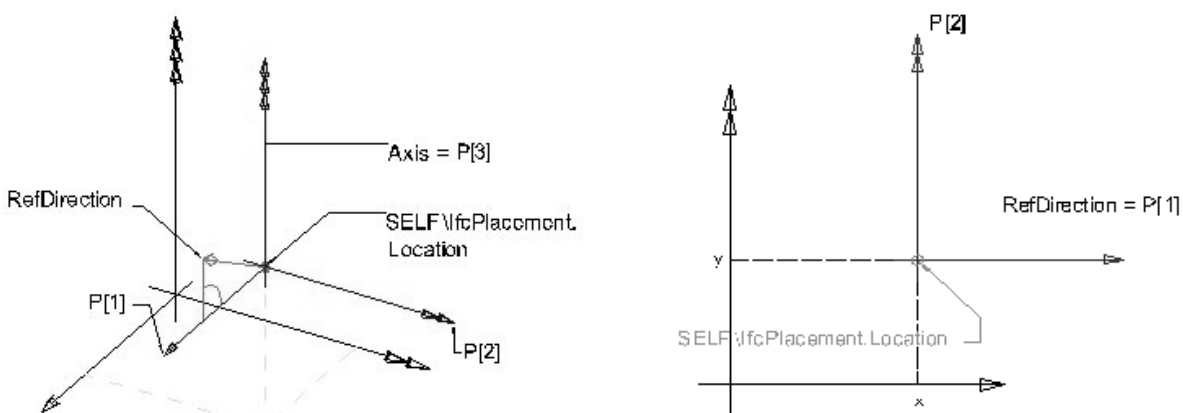


Abbildung 32: Lage und Orientierung in 3 Raumdimensionen (links) und in 2 Raumdimensionen (rechts).

Ist die Lage eines Objektes relativ zu einem übergeordneten Objekt definiert (z.B. Fenster zu Wand), steht das Bezugskordinatensystem im Attribut *PlacementRelTo* der Klasse *IfcLocalPlacement*. In den nachfolgenden Abbildungen ist beispielhaft die Transformation eines geometrischen Objekts in die Position im Projektkoordinatensystem gezeigt. Das Beispielobjekt ist eine Fensteröffnung (*IfcRelVoidsElement*), die mit Grundfläche und Höhe beschrieben ist (linkes Bild im Koordinatensystem x' , y' , z'). Das Objekt wird durch Auswertung der Attribute aus *IfcAxis2Placement3D* in die lokale Lage rotiert (Transformation 1 bezüglich der Koordinaten X_1 , Y_1 , Z_1). Über die Klasse *IfcLocalPlacement* wird der geometrische Bezug zum *IfcObjectPlacement*-Objekt der übergeordneten Wand hergestellt (Transformation 2 bezüglich der Koordinaten X_2 , Y_2 , Z_2).

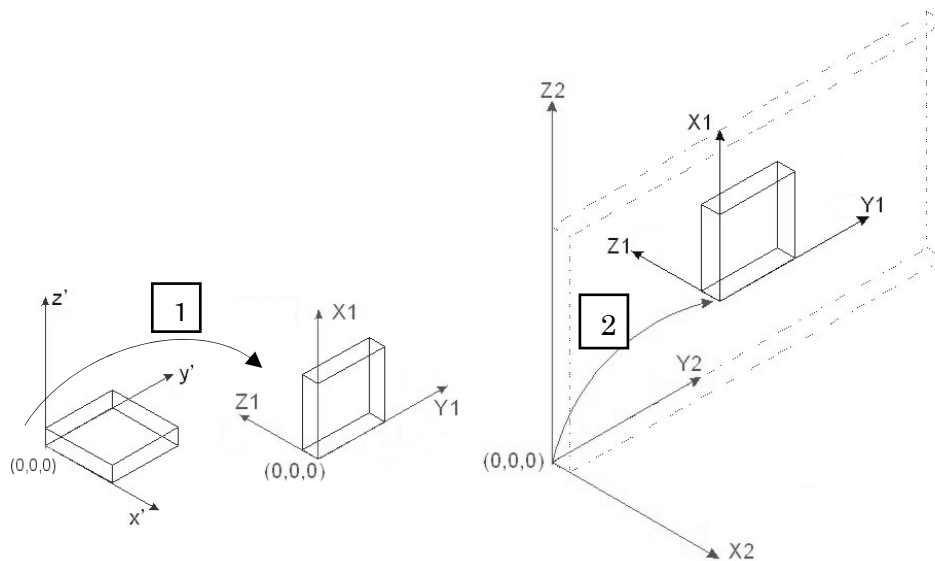


Abbildung 33: Transformation einer Fensteröffnung relativ zur Wand.

Wand und Fensteröffnung liegen nun im lokalen Koordinatensystem der Wand vor. Beide Objekte werden nun in die korrekte Lage innerhalb des Gebäudekoordinatensystems (X_3 , Y_3 , Z_3) verschoben und rotiert. Zuletzt werden alle Objekte eines Gebäudes in die richtige Lage innerhalb des Projektkoordinatensystems überführt (X_4 , Y_4 , Z_4). Die Fensteröffnung ist nun nach vier Transformationen in der endgültigen Position. Transformationen in homogenen Koordinaten¹⁹ werden von ACIS unterstützt, d.h. Translation und Rotation werden für jede der vier Transformationen zu jeweils einer Transformationsmatrix zusammengefasst. Die richtige Lage im Projektkoordinatensystem erhalten die Objekte dann durch Anwendung der entsprechenden (drei oder vier) Transformationsmatrizen auf jedes einzelne Objekt. Für den Fall, dass das IFC-Modell nur ein Bauwerk enthält, bleiben die Transformationsmatrizen für die 3. und 4. Transformation konstant und können zusammengefasst werden.

¹⁹ Siehe dazu [Bungartz 96]

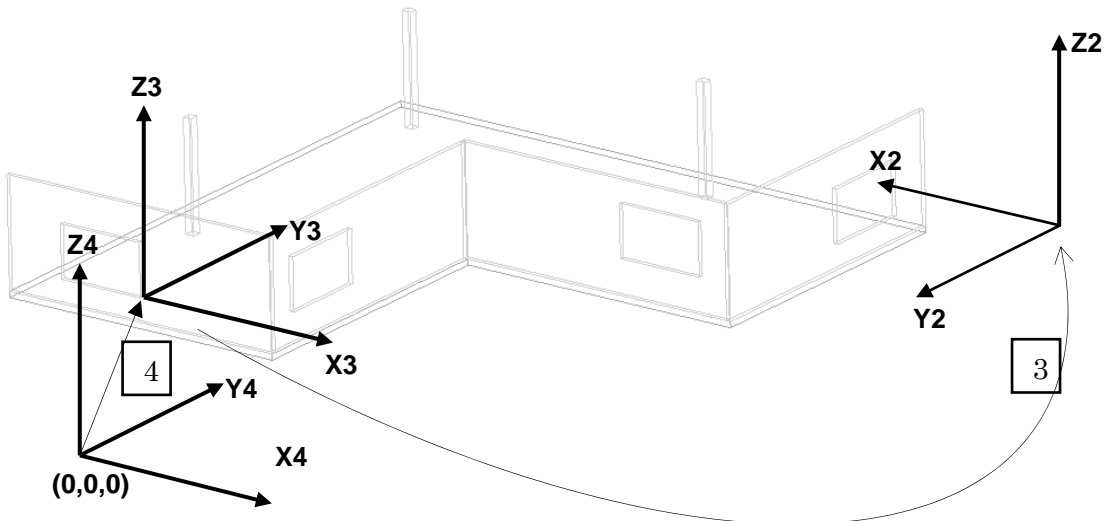


Abbildung 34: Transformation der Wand mit Fenster in das Bauwerkskoordinatensystem (X_3, Y_3, Z_3) und in das Projektkoordinatensystem (X_4, Y_4, Z_4).

3.2.4.3 Die geometrische Beschreibung von Objekten

Objekte, die eine geometrische/topologische Beschreibung beinhalten (z.B. *IfcWall*) definieren diese durch das IFC-Entity *IfcProductRepresentation*. Die geometrische Produktbeschreibung kann aus einer oder mehreren Varianten (*IfcShapeRepresentation*) bestehen. Eine Wand kann beispielsweise durch eine Bounding-Box approximiert werden und daneben über ein Produktionsmodell (*Sweeping-Modell*) exakt beschrieben sein. Die Bounding-Box ist bei der Bestimmung von geometrischen Nachbarschaften hilfreich, da dabei in einem vorab durchgeführten Ausschlussverfahren (Schnittmenge zweier Bounding-Boxen) eine Reduktion der Menge der zu überprüfenden Objekte effizient vorgenommen werden kann.

Die Klasse *IfcProductRepresentation* spezifiziert den Darstellungs-Kontext über das Aufzählungs-Attribut *RepresentationType*. Dieser kann folgende Werte annehmen:

- **SurfaceModel:** Oberflächenmodell.
- **SweptSolid:** Translationsmodell oder Rotationsmodell (Produktionsmodell).
- **BRep:** Boundary Representation
- **CSG** Constructive Solid Geometry
- **BoundingBox:** umschließender Quader
- **MappedPresentation:** Referenz zu global definierten Präsentation (zwischen diversen Objekten geteilte geometrische Beschreibung im Sinne der Wiederverwendung).

Die Klasse *IfcShapeRepresentation* fasst das Konzept einer konkreten geometrischen Darstellung eines Produktes in einem speziellen Darstellungskontext zusammen. Alle Beschreibungs-Elemente, aus denen die geometrische Beschreibung des Produkts zusammengesetzt wird, werden über die Klasse *IfcGeometricRepresentationItem*

aggregiert (siehe dazu Abbildung 38). Eine Wand kann beispielsweise über ein BRep-Modell beschrieben werden, eine in der Wand liegende Fensteraussparung könnte über ein anderes geometrisches Modell beschrieben werden (z.B. Translationsmodell).

Entsprechend des geometrischen Beschreibungs-Kontextes definiert das IFC-Modell Ableitungen der Klasse *IfcGeometricRepresentationItem* (siehe Abbildung 38). Darüber hinaus gibt es Hilfsklassen wie *IfcPlacement* (ein Punkt im kartesischen Koordinatensystem), *IfcDirection* (Definition einer Richtung über einen Ortsvektor), *IfcHalfSpaceSolid* (Halbraummodell zur Beschneidung von Objekten, z.B. abgeschrägte Wand) und *IfcSurface* (Oberflächenbeschreibung beim BRep-Modell). Für die vorliegende Arbeit von besonderer Bedeutung ist die Klasse *IfcSolidModel* mit der abgeleiteten Klasse *IfcExtrudedAreaSolid*. Diese bildet die abstrakte Oberklasse für alle verwendeten Produktionsmodelle (Translationsmodell, Rotationsmodell). Ebenfalls von *IfcSolidModel* abgeleitet sind die Klasse für die BRep-Darstellung (*IfcManifoldSolidBrep*) und die CSG-Darstellung (*IfcCSG*) dreidimensionaler Körper. Das BRep-Modell wird in dieser Arbeit lediglich für die Beschreibung von Fenstern oder Türen eingesetzt, spielt damit eine untergeordnete Rolle und wird nicht weiter beschrieben. Das CSG-Modell wird nicht eingesetzt. Weiterführende Informationen zu CSG- und BRep-Darstellung in den IFCs können [IFC2x2] entnommen werden.

Die Klasse *IfcProfileDef* ist die Basisklasse für alle Definitionen von 2D-Profilen, die über Attribute beschrieben werden. Die Beschreibung der Profile über Attribute wird in der IFC auch *Attribute Driven Profile Definition* genannt. Die gebräuchlichsten sind:

- *IfcRectangleProfileDef*: Beschreibung über Länge (*XDim*) und Breite (*YDim*). Siehe dazu Abbildung 35.
- *IfcArbitraryClosedProfileDef*: Beliebige, über *Curves* berandete Fläche. Siehe dazu Abbildung 37.

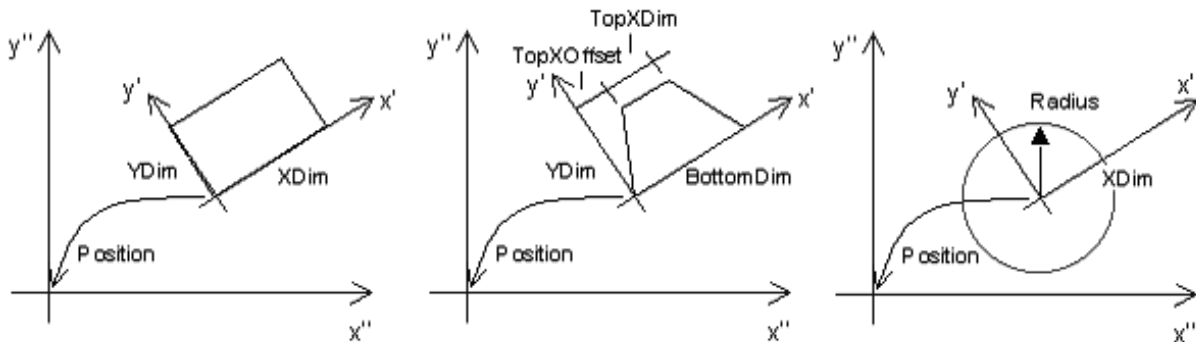


Abbildung 35: *IfcRectangleProfileDef* (linkes Bild), *IfcTrapeziumProfileDef* (mittleres Bild) und *IfcEllipseProfileDef* (rechtes Bild) mit zugehörigen Attributen.

Im nachfolgenden Bild wird die Anwendung der Attribute der involvierten Klassen gezeigt. Die überwiegende Zahl der Objekte in der Tragwerksplanung werden mit diesem *representation context* beschrieben (z.B. Stützen, Wände, Unterzüge, rechteckige Platten,...). Die beiden Attribute *IfcRectangleProfileDef.XDim* und *IfcRectangleProfileDef.YDim* sind im nachfolgenden Bild nicht dargestellt.

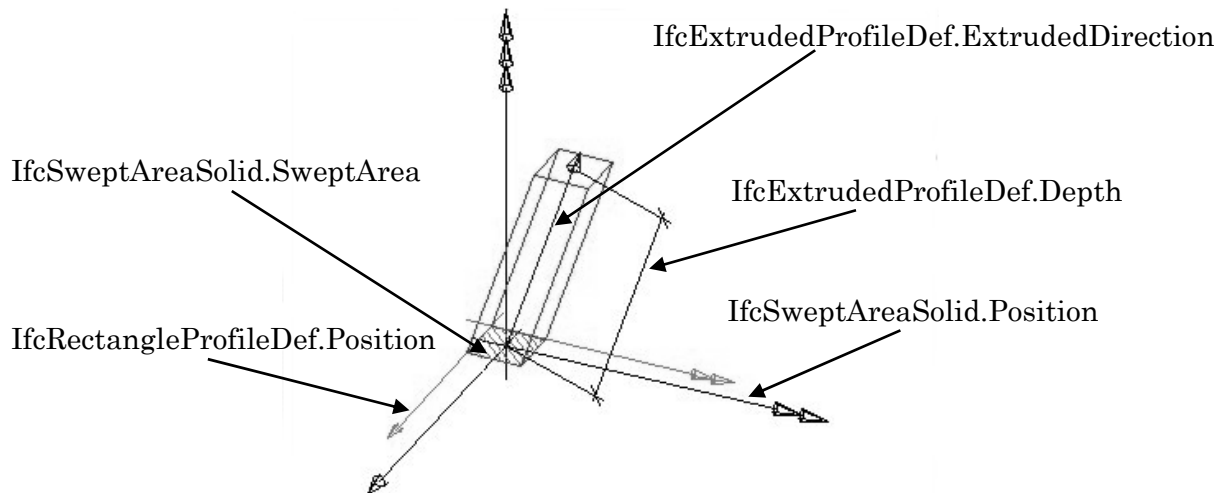


Abbildung 36: Bedeutung der Attribute der Klasse *IfcExtrudedAreaSolid*.

Die Erzeugung des in Abbildung 36 dargestellten Quaders geschieht mit folgender Vorgehensweise: Mit den Werten aus *IfcRectangleProfile.XDim* und *IfcRectangleProfile.YDim* wird im Koordinatensystem aus *IfcRectangleProfileDef.Position* eine Fläche erzeugt (Grundfläche der Extrusion). Der Zugriff auf die genannten Werte erfolgt über das Attribut *IfcSweptAreaSolid.SweptArea*. Die erzeugte Fläche wird in Richtung von *IfcExtrudedProfileDef.ExtrudedDirection* um den Betrag *IfcExtrudedProfileDef.Depth* extrudiert. Der von der Grundfläche überstrichene Bereich des dreidimensionalen Raumes ergibt den zu erzeugenden Körper.

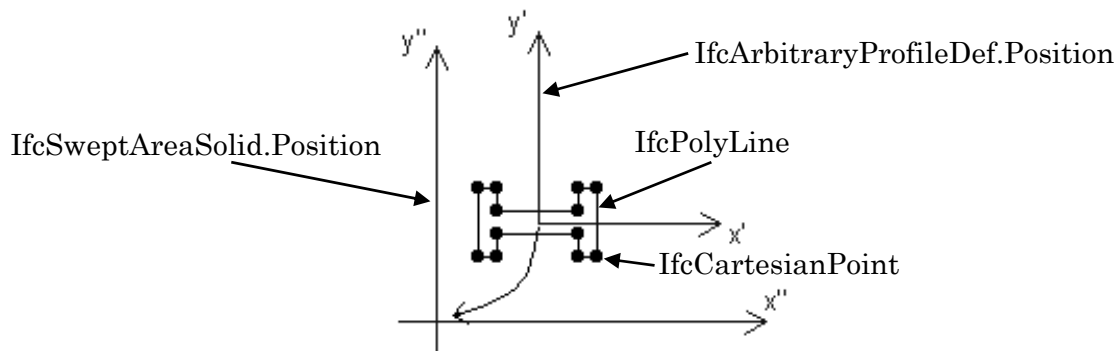


Abbildung 37: Die Attribute der Klasse *IfcArbitraryProfileDef*.

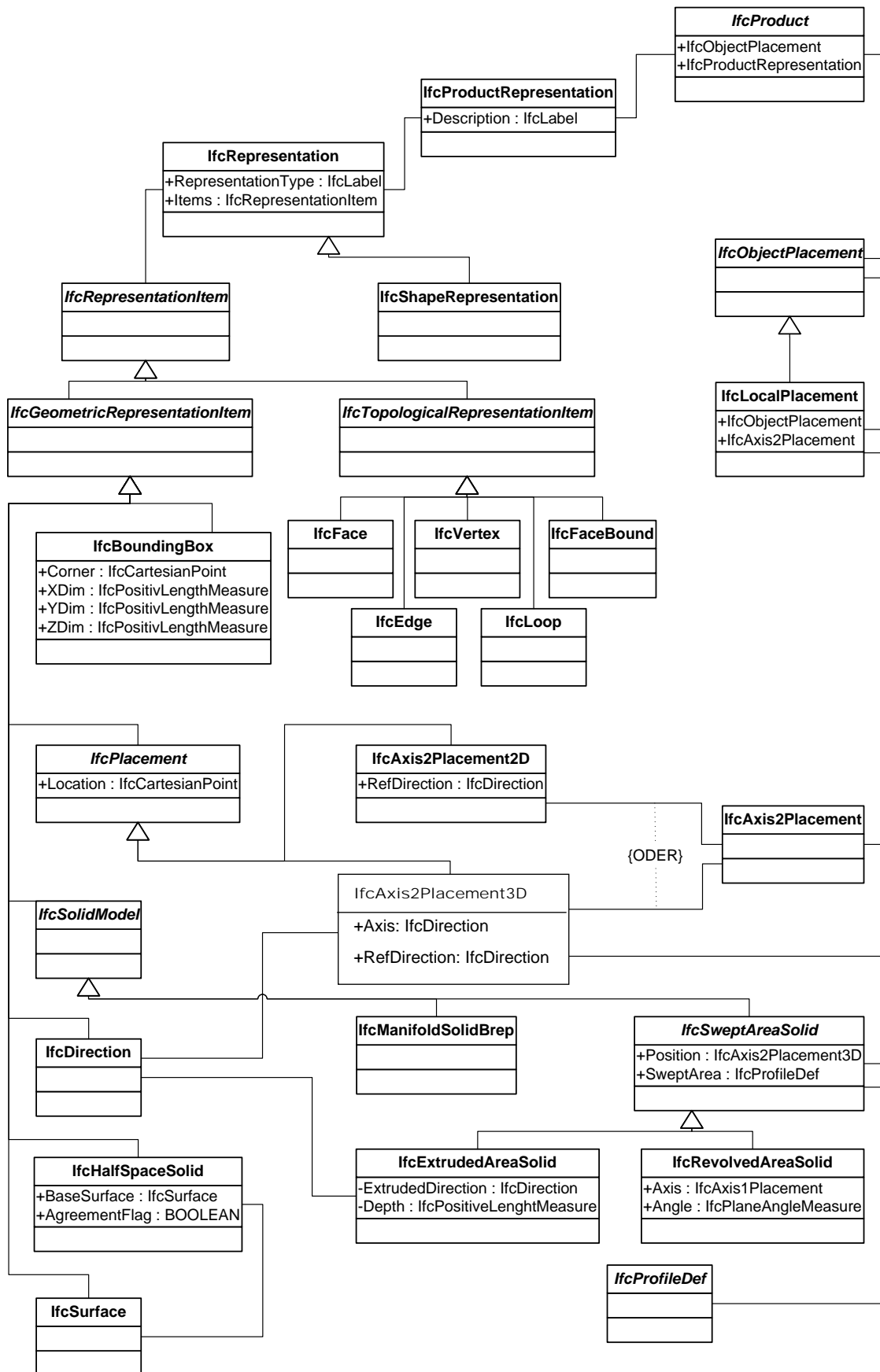


Abbildung 38: Auszug aus dem statischen Klassenmodell der IFC.

3.2.4.4 Behandlung von Wandanschlüssen

Die Auswertung der geometrischen Beschreibung, wie sie in den vorangegangenen Kapiteln dargestellt wurde, ergibt noch kein gültiges geometrisches Modell des darzustellenden Bauwerks. Der Grund liegt in der Modellierung von Wandverschneidungen, die einer Auswertung der geometrischen Beschreibung folgen muss, um Lücken oder Überschneidungen zwischen Wänden zu korrigieren. Es werden folgende Verschneidungsarten angewandt:

- Beschreibung durch ein *IfcConnectionCurveGeometry*-Objekt. Dabei wird ein Verschneidungspfad als Folge von *Curves* (i.d.R. Linien) beschrieben, die die Trennlinie zwischen Wänden definiert.
- Beschreibung durch ein *IfcArbitraryProfileDef*-Objekt. Dabei wird die Grundfläche einer Wand durch ein Polygon beschrieben. Ein Verschneidungspfad ist damit überflüssig.
- Stumpfer Wandanschluss. Dieser Methode ist zwar in der IFC nicht vorgesehen, wird jedoch hier alternativ eingesetzt, da das Verfahren zur Dekomposition der Bauwerksgeometrie diese Art der Wandverschneidung voraussetzt. Siehe dazu Kapitel 3.3.2.

Wandverschneidung mittels *IfcConnectionCurveGeometry*

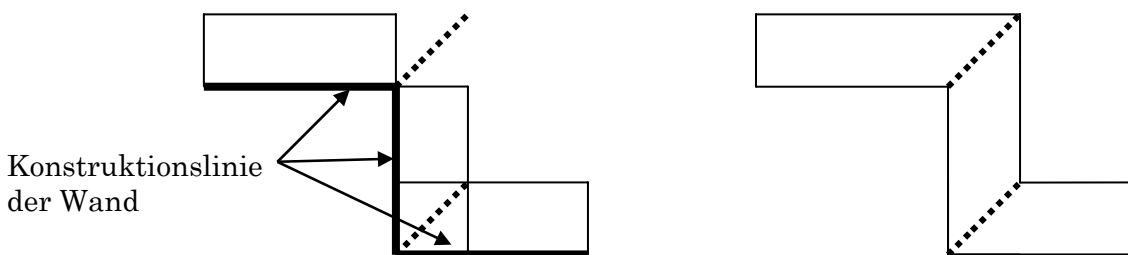


Abbildung 39: Verschneidung der Wandenden einer Wandfolge (links als *IfcRectangleProfileDef*) mittels *IfcConnectionCurveGeometry*-Objekt (gestrichelte Linie).

Im linken Bild der Abbildung 39 ist eine Wandfolge in der geometrischen Beschreibung über die Klasse *IfcRectangleProfileDef* abgebildet. Die dicke Linie markiert die Ausrichtung der Wände, die bestrichelte Linie wird über die Klasse *IfcConnectionCurveGeometry* beschrieben und definiert die Trennlinie zwischen den Wänden. Im rechten Bild derselben Abbildung ist der Wandzug nach der geometrischen Korrektur der Wand-Geometrie dargestellt. Anhand des nachfolgenden Beispiels werden die einzelnen Schritte der Korrektur gezeigt.

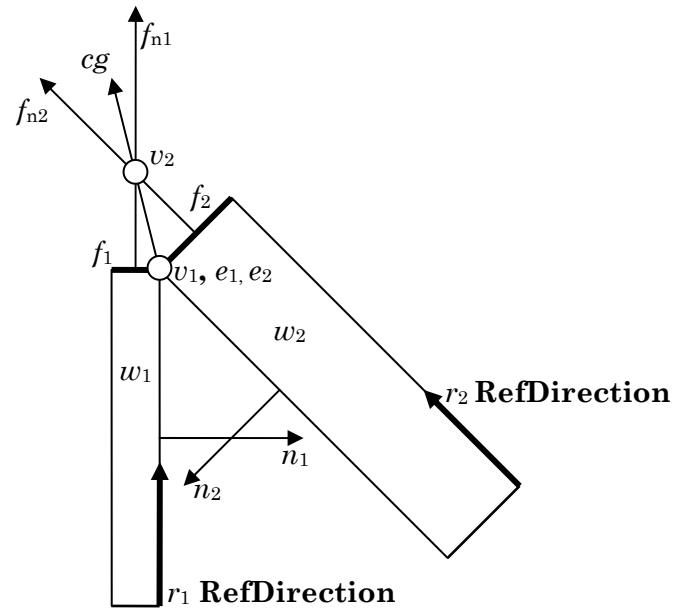


Abbildung 40: Benennung der Elemente zur Korrektur der Wand-Objekte

Nach der Erzeugung der Wand-Grundkörper sind folgende Objekte gegeben:

- Die Wandobjekte nach der Übernahme der geometrischen Beschreibung aus dem IFC-Modell (w_1, w_2).
- Die horizontale Richtung (Grundlinie) zu jeder Wand (RefDirection). Dieser Vektor wird aus dem ursprünglichen lokalen Koordinatensystem *IfcAxis2Placement2D* des Attributes *IfcProfileDef* (siehe Abbildung 32) extrahiert und im Wandobjekt als Ortsvektor vorgehalten (r_1, r_2).
- Ein Vektor je Wand, der normal zur Seitenfläche der Wand angeordnet ist und die Richtung der Wandschichten vorgibt. Dieser wird bei der Übernahme der geometrischen Beschreibung erzeugt und im Wandobjekt als Ortsvektor vorgehalten (n_1, n_2).

In einem ersten Schritt werden die Kanten e_1, e_2 der Körper w_1 bzw. w_2 bestimmt, die beide vom Knoten v_1 berandet werden und senkrecht zur Papierebene liegen. Die Bestimmung muss für IFC-Modelle über geometrische Informationen vorgenommen werden, da topologische Elemente (*Edges*) für jeden Körper einmalig sind und nicht zwischen den Körpern w_1 und w_2 geteilt werden. Die Vektoren der Kante e_1 und cg (Vektor, der durch *IfcConnectionCurveGeometry* beschrieben wird) spannen nun eine Ebene (*Surface*) s_1 auf.

Die nachfolgenden Erläuterungen beziehen sich nun auf die Wand w_1 , für die Wand w_2 gilt entsprechendes. Im nächsten Schritt wird die Flächen f_1 bestimmt. Das ist diejenige Fläche, deren Normalenvektor zum Vektor r_1 parallel und zum Vektor n_1 rechtwinklig liegt. Diese Fläche wird nun in Richtung des Vektors f_{n1eg} extrudiert. Der Vektor f_{n1eg} ist die senkrechte Projektion von cg auf f_{n1} . Der erzeugte Körper wird nun mit der bestehenden Wand vereinigt (siehe Abbildung 41). Im letzten Schritt wird das Volumen der erweiterten Wand mittels der zuvor erzeugten Surface s_1 gestutzt.

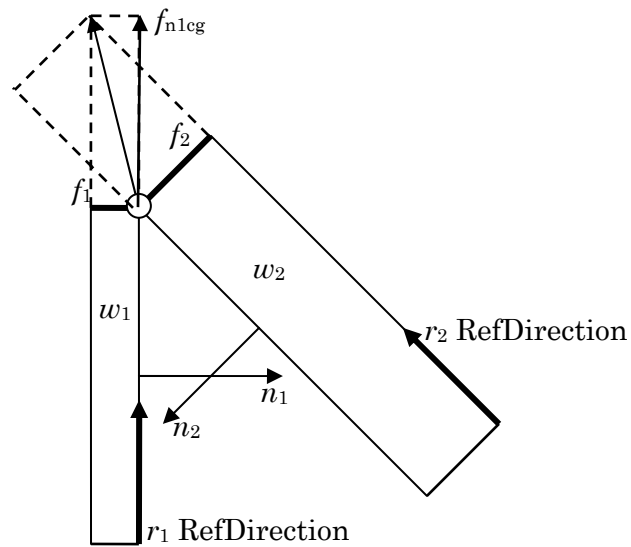


Abbildung 41: Erweiterung der Wand-Objekte bis zur vollständigen Überlappung.

Beispiel 4.1: Einfache Wandfolge

Das Korrekturverfahren wird nun auf alle Wandverbindungen angewandt. Das nachfolgende Beispiel zeigt die Korrektur an einem konkreten Beispiel.

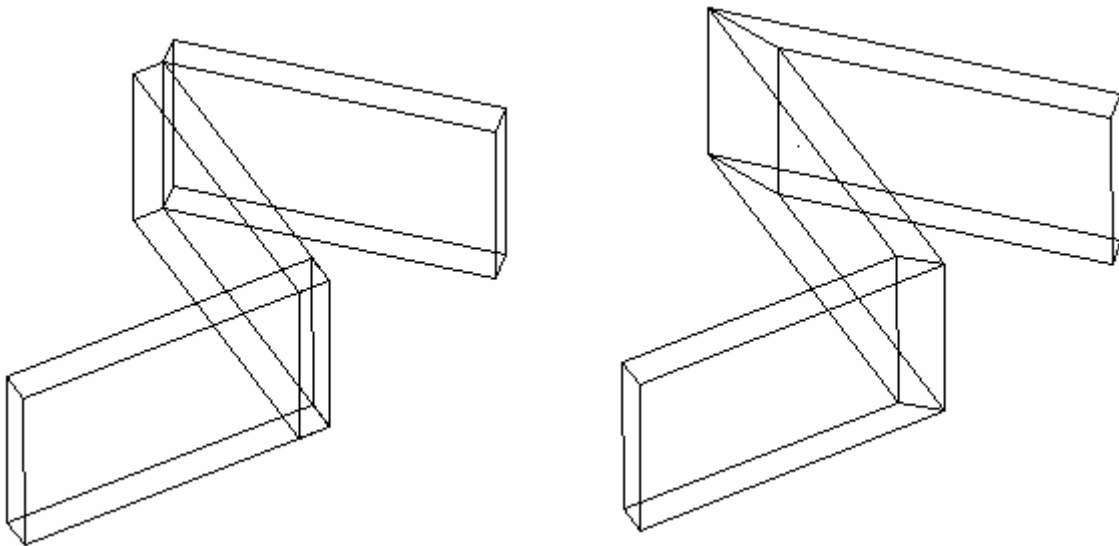


Abbildung 42: Wandzug mit drei Wandobjekten. Vor der Korrektur (links) und nach der Korrektur (rechts).

3.2.4.5 Attributierung des geometrischen Modells

Der Vererbung von Attributen bei Manipulationen am geometrischen Modell kommt eine wichtige Rolle bei der Bauwerksmodell-basierten Tragwerksplanung zu. Die Überführung vom ursprünglichen Bauwerksmodell hin zum numerischen Modell ist nur dann vollständig programmgesteuert möglich, wenn Manipulationen am geometrischen Modell nicht zu einem Informationsverlust am Attribut-System führen. Zum Beispiel muss die Belegung einer Fläche im Bauwerksmodell mit einem bestimmten Lastattribut zu der gleichen Belegung der korrespondierenden Fläche im finiten Element (=Teilfläche der ursprünglich attributierten Fläche) des numerischen Modells führen.

Im Abbildung 43 ist stellvertretend für die Zuordnung von Attributen zu Elementen des geometrischen Modells eine Schraffur angetragen. Nach der Abtrennung eines Teilvolumens werden die Attribute (hier die Schraffur) – je nach Funktionalität der Attributverwaltung - der korrespondierenden Fläche des neu erzeugten Körper zugewiesen oder nicht. Im Falle der Vererbung der Attribute wird die Schraffur der ‚oberen‘ Flächen auf die ‚obere‘ Fläche des neuen Körpers übertragen (rechtes Bild in Abbildung 43). Es ist nun leicht nachvollziehbar, dass mit numerischen und/oder alphanumerischen Attributen anstelle der Schraffur nahezu beliebige Verknüpfungen zu Kanten, Flächen und Volumen in die assoziierte Datenbank zur Attributierung des geometrischen Modells möglich werden.

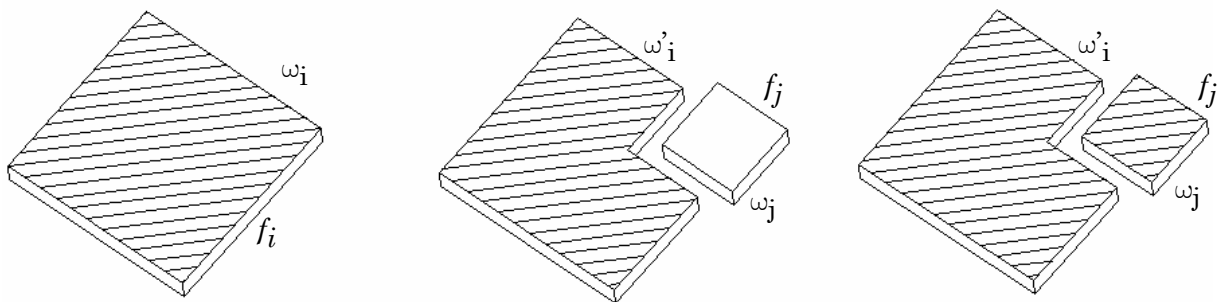


Abbildung 43: Zu einer Fläche f_i des Körpers ω_i zugeordnetes Attribut, dargestellt anhand einer Schraffur (links). Die Abtrennung eines Teilvolumens ω_j ohne Attributvererbung (Mitte) und mit Attributvererbung (rechts).

Im Falle, dass der Modellierkern die Vererbung von Attributen nicht unterstützt, muss nach der Abtrennung eines Teilvolumens die Vererbung nachträglich vorgenommen werden. Bei den mit Körpern assoziierten Attributen ist eine Übertragung von ω_i nach ω_j (siehe dazu Abbildung 43) unmittelbar möglich. Hinsichtlich der Übertragung von Flächenattributen ist die Identifikation korrespondierender Flächen f_i, f_j der Körper ω_i und ω_j notwendig. Es gilt

$$(f_j \in_t \omega_j \mid f_i \in_t \omega_i \vee f_j \subset_b f_i) \quad (9)$$

Der Aufwand liegt dabei in der Bestimmung von $f_j \subset_b f_i$, da nur in wenigen Fällen der Vergleich über Normalenvektoren der Flächen ($n_j \parallel n_i$) als Ersatzkriterium zu Hilfe genommen werden kann.

Bei der Auswahl des Modellierkerns wurde deshalb darauf geachtet, dass die Vererbung von Attributen unterstützt wird. ACIS [Corney 97] bietet diese Funktionalität bei geometrischen Mengenoperationen für alle von ENTITY²⁰ abgeleitete Objekten. Sie wird in der vorliegenden Arbeit für folgende Funktionalität genutzt:

- Verknüpfung von Objekte des geometrischen Modells (Bodies) mit Strukturknoten in der assoziierten Datenbank.
- Protokollierung der Vererbungshistorie für neu generierte Objekte im Verlauf der Modellzerlegung/Modellmodifizierung. Damit hat z.B. ein finites Element im numerischen Modell die Information, aus welchem IFC-Entity es erzeugt wurde.
- Speicherung von Informationen zu geometrischen Nachbarschaften im Modell.
- Attributierung der geometrischen Objekte für die numerische Simulation. Dies gilt für konstante Werte hinsichtlich der Brandung, andernfalls müssen andere Verfahren angewandt werden (z.B. Referenz auf die ursprüngliche Definition bei einer Trapezlast).

Die Verwaltung von Attributen²¹ über geometrische Objekte im Modell eröffnet nun bei der vorgestellten Methode die Möglichkeit, das originale BRep-Modell (=Gebäudemodell) mit allen Werten, Randbedingungen, etc. zu versehen, die für das eigentliche Ziel (die numerische Simulation) benötigt werden. Hinsichtlich der unterschiedlichen Semantik der Objekt-Typen für das Simulationsverfahren werden folgende Attribute verwendet:

Körperattribute

- Objekte ID (entspricht der ‚Globally Unique ID‘ der IFC; auch GUID).
- Objekt-Typ (aus den IFCs übernommen), z.B. Wall, Column, etc.
- Vererbungsinformationen (Objekt ID des Objektes, von dem das betrachtete Objekt abgeleitet wurde).
- Schichtungsrichtung als Ortsvektor mit den Komponenten x,y und z. (Vektor, der bei geschichteten Bauteilen, z.B. Wand die Richtung der Schichtung angibt).
- Material ID (bei geschichteten Bauteilen wird der Schichtaufbau referenziert).
- Eigenschaften (hierarchisches Schema mit einfachen und komplexen Strukturknoten für die Speicherung beliebiger Daten ohne standardisiertes Format). Siehe dazu auch [IFC2x2] unter dem Stichwort ‚PropertySet‘.

²⁰ In der vorliegenden Arbeit werden zur Attributierung des geometrischen Modells lediglich die von ENTITY abgeleiteten Klassen EDGE, FACE und BODY verwendet.

²¹ Die Art der Attributierung ist grundsätzlich geeignet, alle möglichen Arten von Eingangswerten, Randbedingungen, etc. für Simulationen an das geometrische Modell zu ‚heften‘ und damit das Gesamtmodell über Manipulationen hinweg konsistent zu halten. Siehe dazu auch [Treeck 04b].

- Informationen zur numerischen Simulation (Ansatzgrad und physikalisches Modell).
- Informationen zur Diskretisierung (Wert zur maximal zulässigen Elementkantenlänge).
- Typ des Objektes hinsichtlich der Zerlegung des Modells. Aufgrund der unterschiedlichen Bedeutung der Objekte bei der Zerlegung werden die so genannten Differenzobjekte mit M^1 , die Verbindungsobjekte M^2 oder M^3 und die Stützenobjekte mit M^4 markiert (siehe Kapitel 3.3.2 ‚Dekomposition des geometrischen Modells‘).
- Fehlerwert für die visuelle Information des Benutzers nach Fehlern.

Flächenattribute

- Objekt ID des über die Fläche benachbarten Objektes.
- Randbedingungen für die Strukturanalyse (Last ID, Lagerungs ID).

Kantenattribut

- Information zur Diskretisierung. (Der Wert zur maximal zulässigen Elementkantenlänge wird bei der Freivernetzung von scheibenförmigen Objekten auf dessen Kanten übertragen)

Darüber hinaus werden bei der Ausführung von Modell-Operationen, z.B. der Zerlegung des geometrischen Modells, diverse temporäre Attribute eingeführt. Die Möglichkeit der Attributierung des geometrischen Modells in Zusammenhang mit der Speicherung von Daten in der assoziierten Datenbank erlaubt nun die vollständige Attributierung des geometrischen Modells hinsichtlich Lasten, Lagerungen, Diskretisierungsparameter, numerischer Parameter. Die Ableitung der nachfolgenden Modelle (Verbindungsmodell, numerisches Modell) kann damit automatisch ablaufen.

In der nachfolgenden Abbildung wird illustriert, wie (ausgewählte) Attribute über Modelloperationen hinweg übertragen werden. Das Modell mit der diskretisierten Geometrie (rechts) wurde aus dem Verbindungsmodell abgeleitet, dieses wiederum aus dem Gebäudemodell (links). Einerseits lässt sich damit die Entwicklungsgeschichte des Modells aus der Attributierung ablesen, andererseits werden zugeordnete Daten bei der Manipulation des Modells dupliziert und können zu jeder Zeit programmtechnisch ausgewertet werden.

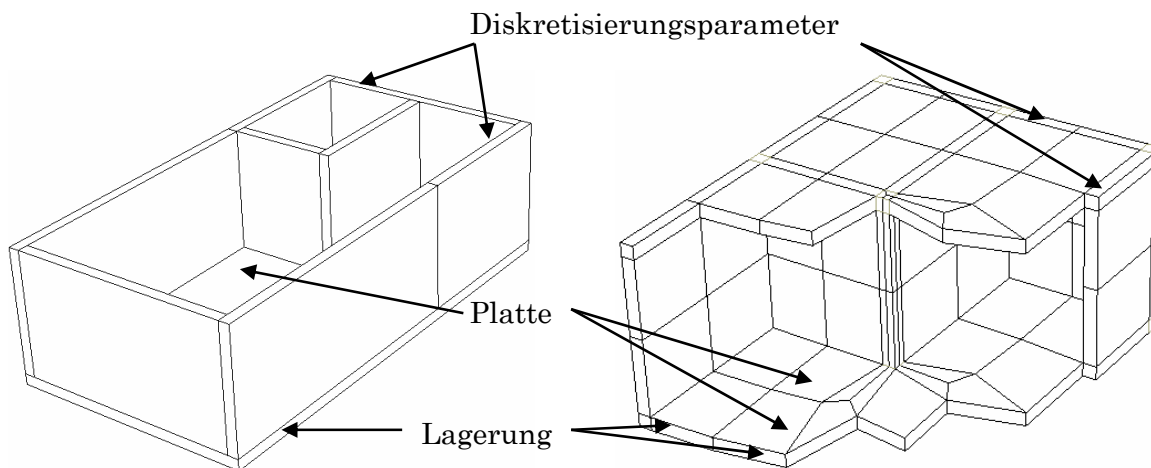


Abbildung 44: Bauwerksmodell mit Attributen (links) und abgeleitetes Modell mit vererbten Attributen (rechts).

3.3 Analyse und Aufbereitung der Gebäudestruktur²²

3.3.1 Einführung

Die Vorteile, die ein Produktmodell-basierter Ansatz für numerische Simulationen bietet, werden wesentlich erweitert, wenn neben der geometrischen und semantischen Beschreibung des Modells auch topologische Informationen (Gebäudestruktur) herangezogen werden. Es lassen sich zwei ‚duale‘ topologische Strukturen nennen. Die Gebäudestruktur wird zum einen durch die Menge der Bauteile definiert, die für das Tragwerk, für die Raunteilung, für die Fassade usw. vorhanden sind. Zum anderen definiert sie sich durch ‚Räume‘, die den Aspekt der Nutzung (Wegebeziehungen, thermische Zonen, etc.) widerspiegeln. Beide Strukturen beeinflussen sich wechselseitig und werden zur Konsistenzsicherung gemeinsam definiert und verwaltet. Während in herkömmlichen Produktmodellen (z.B. im IFC-Modell) die bauteilorientierte Sicht im Zentrum steht, führt die Notwendigkeit der Konsistenzsicherung dazu, dass Räume mit ihrem Luftvolumen zwar als eigenständige geometrische Objekte angesehen werden, die jedoch aus der bauteilorientierten Sichtweise abzuleiten sind. Die Beziehungen der Objekte – innerhalb der jeweiligen Sichtweise sowie übergreifend – definieren die gesamte topologische Struktur des Gebäudes. Zur formalen Beschreibung können dafür verschiedene Graphen identifiziert werden:

- Der Bauteilgraph (Relationen zwischen Bauteilen).
- Der Verbindungsgraph (Nachbarschaften im Verbindungsmodell, siehe dazu nachfolgenden Abschnitt).

²² Teile des Kapitels 3.3 wurden bereits im Forschungsbericht zum DFG Schwerpunktprogramm 1103 vorveröffentlicht.

- Der Raumflächengraph (Flächen, die einen Raum begrenzen).
- Der Raumgraph (Relationen zwischen Räumen).

Ausgehend vom geometrischen Basismodell können mit Hilfe dieser Graphen alle wesentlichen topologischen Beziehungen des Bauwerks erkannt und dargestellt werden. Der Bauteilgraph ergibt sich aus den Relationen zwischen den Bauteilen des originalen Produktmodells. Ungeachtet der Bauteilrelationen im Bauteilgraph sind die einzelnen Bauteile aus dem Konstruktionsverfahren heraus i.d.R. unterschiedlich modelliert (durchgehende Wand oder gestoßener Wandzug). Die Grundidee zur Generierung des Verbindungsgraphen ist nun, das gesamte geometrische Modell zunächst in ‚Separatoren‘ (Verbindungsobjekte) und Teilmodelle (Differenzobjekte) zu zerlegen. Damit lässt sich zu jedem Element im Verbindungsgraph ein geometrisches Objekt im Modell identifizieren (siehe dazu Kapitel 3.3.2 ‚Dekomposition des geometrischen Modells‘). Der Raumflächengraph nutzt als Eingangsdaten das Verbindungsmodell in Zusammenhang mit dem Verbindungsgraphen und identifiziert alle Flächen, die einen Raum begrenzen. Die Flächen werden im nächsten Schritt ‚invertiert‘ und wieder zu neuen Körpern zusammengesetzt. Diese generierten Körper definieren nun die Räume und ergeben topologisch betrachtet den Raumgraphen. Das zum Verbindungsgraph zugeordnete Verbindungsmodell wurde im Rahmen dieser Arbeit entwickelt und wird im Kapitel 3.3.2 ‚Dekomposition des geometrischen Modells‘ näher beschrieben. Weiterführende Informationen zum Raumflächen- und Raummodell sind in [Treeck 04a, Treeck 04b] zu finden. Der Nutzen des Verbindungsmodells erstreckt sich neben der oben genannten Verwendung hinsichtlich der Gebäudetopologie auch auf die numerische Diskretisierung. Die Aufteilung des geometrischen Modells in Verbindungskörper und Differenzkörper ist Voraussetzung für eine automatische Vernetzung in Hexaeder. Siehe dazu nachfolgendes Kapitel 3.4.

Im folgenden Beispiel eines IFC-Modells ist eine Wand/Decken-Verbindung sowie der zugehörige Bauteilgraph, der durch flächenhafte Verbindungen zwischen den Bauteilen definiert ist, dargestellt. Das Beispiel zeigt, dass Nachbarschaften im Graphen zwar beschrieben sind, der geometrische Bezug aber verloren geht. Für die in Kapitel 3.4 ‚Automatische makrobasierte Hexaeder Netzgenerierung‘ vorgestellte Methode, sind folgende Anfragen an das Modell von großer Bedeutung:

- Welches Objekt ω_i liegt dem Objekt ω_j bezüglich eines dritten Objektes ω_k unmittelbar gegenüber? In Abbildung 45 liegen ω_3 und ω_4 bezüglich ω_1 gegenüber.
- Lagebestimmung der flächigen Bauteilverbindung: An welchen Orten berühren sich Bauteile? Betrachtet man in Abbildung 45 die Objekte ω_2 und ω_1 , so lässt sich die Lage der Bauteilverbindung für das Objekt ω_1 leicht beantworten, da Kanten und eine Fläche zur Verfügung stehen, die für die Bauteilverbindung (eine Kante im Graph) stellvertretend herangezogen werden können. Für Objekt ω_2 existieren diese Elemente nicht.
- Mit welcher Fläche treten Nachbar-Objekte in Kontakt? Diese Frage ist hinsichtlich des Objekts ω_2 gegenüber den anderen Objekten nicht eindeutig zu beantworten (Die Oberseite von ω_2 wird in Abbildung 45 mittels *einer* Fläche beschrieben).

Zusammenfassend kann festgehalten werden, dass die Daten, die im Graphen vorgehalten werden i.d.R. nur zu einem Teil zur Beantwortung von Anfragen beitragen können. Der geometrische Bezug ist in vielen Fällen wichtiger Bestandteil der Lösung.

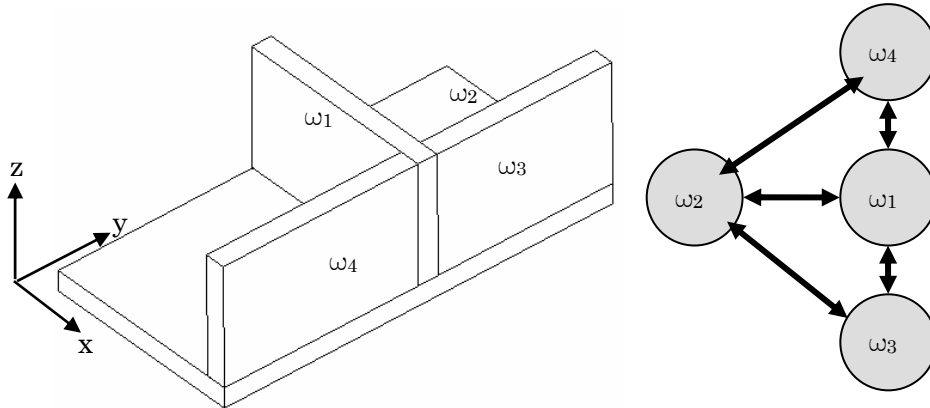


Abbildung 45: Beispiel eines Modells (links) mit zugeordnetem Bauteilgraph (rechts).

Zusammenführung von Objektgraph und geometrischem Modell

Um die Vorteile beider Strukturen – derjenigen des geometrischen Modells sowie der des Graphen – zu verbinden, wird der Graph über Elementen des geometrischen Modells gespeichert. Dafür muss garantiert werden, dass es für jede Verbindung zwischen zwei Bauteilen eine eindeutige verbindende Fläche im Modell gibt. Im Laufe der Zerlegung des geometrischen Modells (siehe nachfolgendes Kapitel) werden im 1. Schritt die erforderlichen Flächen erzeugt. Im Zuge der Generierung neuer Flächen im Modell werden Daten zu den Nachbarschaftsbeziehungen (RPC) gespeichert. Damit wird die topologische Information zur Struktur auf geometrische Objekte (Faces) abgebildet.

Gegeben sei ein geometrisches Gebäudemodell $\Omega \subset \mathbb{R}^3$ bestehend aus n BRep-Objekten

ω_k :

$$\omega_k \subseteq \Omega, \quad k = 1, \dots, n \text{ und} \quad (10)$$

$$\bigcup_{k=1, \dots, n} \omega_k = \Omega = M_{\omega_k}. \quad (11)$$

Damit lässt sich folgender Graph²³ definieren:

$$G_V = (M_{\omega_k}, RPC), \quad RPC \subseteq M_{\omega_k} \times M_{\omega_k} \quad (12)$$

²³ V = Verbindung, B = Body (Volumenkörper), PC = Plane connection (flächige Bauteilverbindung)

Operatoren, die zur Auswertung des Modells entworfen werden, können damit auf geometrische sowie auf topologische Informationen des Modells zugreifen. Darüber hinaus liefert das semantische Modell weitere Informationen zu einzelnen Flächen und Körpern (Objekttyp, Schichtungsrichtung, etc.) im Modell. Damit ist z.B. folgende Abfrage ohne rechenaufwendige Bearbeitung des Modells möglich: ‚Gibt es zwei Wände ω_i , ω_j , die zu einer dritten Wand ω_k benachbart sind und deren Verbindungsflächen zu ω_k gegenüber²⁴ liegen‘. Die Antwort ist die Schnittmenge dreier Teilanfragen:

- Alle Nachbarn (topologische Information).
- Alle Wände (semantische Information).
- Verbindungsflächen, die über den Schichtungsvektors in Beziehung zueinander stehen (geometrische Information).

Die Anwendung von Modelloperatoren, die die Vorteile der kombinierten Datenhaltung nutzen, sind im Kapitel 3.4 ‚Automatische makrobasierte Hexaeder-Netzgenerierung‘ beschrieben.

3.3.2 Dekomposition des geometrischen Modells

Die automatische Generierung von Hexaedern stellt besondere Anforderungen an die geometrischen ‚Separatoren‘ des Verbindungsmodells. Da bestimmte Separatoren in Abhängigkeit vom Knotengrad im Verbindungsgraphen nicht weiter bearbeitet werden, müssen sie bereits bei der Zerlegung des geometrischen Modells den Anforderungen des numerischen Verfahrens an die geometrischen Elemente genügen, d.h. bei Verwendung des in Kapitel 2.4 skizzierten FEM-Verfahrens müssen es acht-knotige räumliche Körper sein. Während dieser Ansatz prinzipiell auf allgemeine räumliche Strukturen angewendet werden kann, sobald die (nicht-triviale) Identifikation von geeigneten Separatoren erfolgt ist, wurde er in dieser Arbeit für typische geometrische Strukturen von Bauwerken ausgearbeitet. Als geeignete Separatoren stellen sich ‚Raumkanten‘ und ‚Raumecken‘ zwischen dünnwandigen, also scheiben- und plattenartigen Strukturen heraus. Ausgangspunkt für die im nächsten Abschnitt beschriebene Vernetzung ist also ein Modell (Verbindungsmodell), das nach genau festgelegten Bearbeitungsschritten aus dem geometrischen Modell generiert wurde.

²⁴ Bezogen auf das Objekt ω_1 in Abbildung 45 werden die Kontaktflächen der benachbarten Objekte ω_3 und ω_4 zur Erzeugung des Verbindungskörpers durch Extrusion gesucht.

Definition des Verbindungsmodells

Gegeben sind die Objekte M_C des betrachteten geometrischen Gebäudemodells²⁵ $\Omega \subset \mathbb{R}^3$ als Menge eines oder mehrerer BRep-Körper. Es lassen sich folgende Schnittmengentypen klassifizieren (siehe Abbildung 46):

- Typ N: Objekte haben aus Knoten bestehende Schnittmenge.
- Typ NE: Objekte sind über Kanten verbunden. Als Schnittmenge ergeben sich Kanten und Knoten.
- Typ NEF: Die Schnittmenge flächig verbundener Bauteile besteht aus Flächen, Kanten und Knoten.

Die Objekte des Modells werden nun abhängig vom Schnittmengentyp zu anderen Objekten klassifiziert und auf Grund der unterschiedlichen Bedeutung für das weitere Verfahren zu *Verbindungsobjekten* M_K und *Differenzobjekten* M_D gruppiert (siehe Abbildung 47). Der Schnittmengentyp zwischen Differenzobjekten ist N oder NE , zwischen Verbindungsobjekten vom Type N , NE oder NEF und zwischen Differenzobjekten und Verbindungsobjekten vom Type N , NE oder NEF .

Die Menge der Verbindungsobjekte M_K wird weiter unterteilt in die *Verbindungsobjekte des Ausgangsmodells* M_{K1} und die *Verbindungsobjekte des Verbindungsmodells* M_{K2} . Dabei gelten folgende Mengenbeziehungen:

$$M_{K1}, M_{K2} \subseteq M_K, \quad M_{K1} \cap M_{K2} = \emptyset \quad \text{and} \quad M_{K1} \cup M_{K2} = M_K \subseteq M_C, \quad (13)$$

$$M_K \cap M_D = \emptyset, \quad M_K \cup M_D = M_C. \quad (14)$$

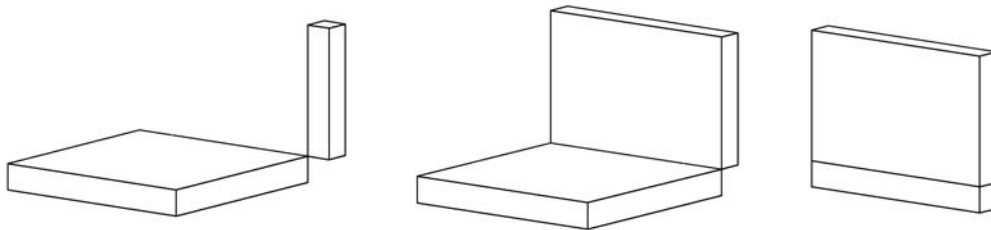


Abbildung 46: Schnittmengen-Typen N, NE and NEF

²⁵ Als Voraussetzung an die Konsistenz des geometrischen Modells gilt: Keine Spalten, Klaffungen und Überlappungen zwischen zwei beliebigen Körpern des Modells. Gewollte Abstände zwischen Objekten des Modells (Fugen) werden explizit modelliert.

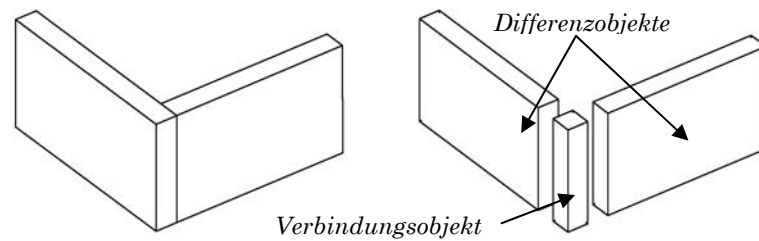


Abbildung 47: Objekte der Menge M_C (links), zerlegtes Modell mit Verbindungsobjekt und Differenzobjekten (rechts)

Algorithmus zur Dekomposition des Modells

Für jedes Objekt-Tupel ω_i, ω_j , für das eine Schnittfläche $f_{i,j}$ existiert (siehe Abbildung 48), wird folgender Algorithmus unter Verwendung des Booleschen Operators \setminus_b aufgeführt:

1. Erzeuge Schnittfläche $f_{i,j}$
2. Erzeuge Normalenvektor $n_{i,j}$
3. Extrudiere $f_{i,j}$ entlang $-n_{i,j}$ um Δd und erzeuge dabei $a_1 \in M_{K1}$
4. $d_1 := \omega_i \setminus_b a_1 \in M_D$
5. $d_2 := \omega_j \setminus_b a_1 \in M_D$

Nach der Zerlegung des Modells Ω besitzen alle Verbindungsobjekte M_K eine topologische Hexaeder-Struktur. Differenzobjekte M_D hingegen sind polygonal berandet.

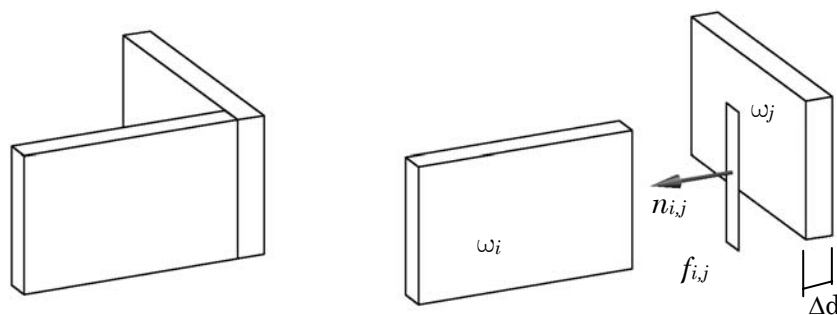


Abbildung 48: Verbindungsfläche $f_{i,j}$ mit Normalenvektor $n_{i,j}$ der Objekte ω_i und ω_j .

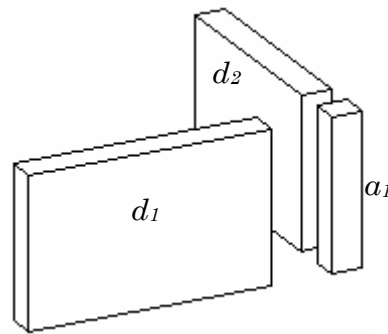


Abbildung 49: Generierung des Verbindungsobjektes a_1 und des Differenzobjektes d_2 aus ω_j .

Beispiel 1: Wandecke

Zur Verdeutlichung der grundlegenden Idee wird nachfolgend in den Abbildung 50 bis Abbildung 52 eine Ausgangskonfiguration Ω und die zugehörige Zerlegung dargestellt. Nach der Ermittlung der Objekt-Beziehungen zwischen je zwei Objekten, werden Imprint-Flächen $f_{i,j}$ erzeugt und Verbindungsobjekte generiert. Die Dekomposition erfolgt rekursiv in zwei Schritten. Im ersten Durchlauf werden die *Verbindungsobjekte des Ausgangsmodells* M_{K1} erzeugt und im zweiten Durchlauf werden die *Verbindungsobjekte des Verbindungsmodells* M_{K2} generiert.

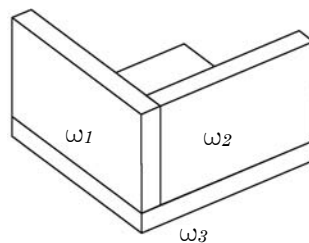


Abbildung 50: Konfiguration Ω mit den Objekten ω_1 , ω_2 and ω_3 .

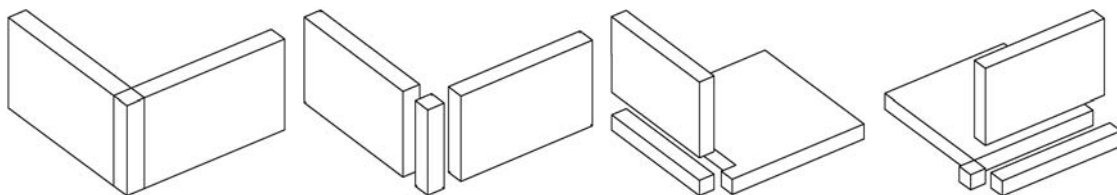


Abbildung 51: Zweifache rekursive Anwendung des Algorithmus.

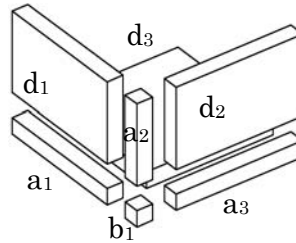


Abbildung 52: Gebiet Ω mit den Objekten ω_1 , ω_2 and ω_3 (links), Modell mit Verbindungsobjekten a_1 , a_2 , a_3 , b_1 und Differenzobjekten d_1 , d_2 , d_3 .

Beispiel 2: Modell mit zwei Räumen

Als zweites Beispiel wird ein einfaches geometrisches Modell mit zwei Räumen verwendet. Eine Verbindungstür zwischen den Räumen ist aus Gründen der Übersichtlichkeit bewusst nicht modelliert worden. Das Beispiel fungiert für die Zerlegung wie für die nachfolgend beschriebene Vernetzung als Standardmodell.

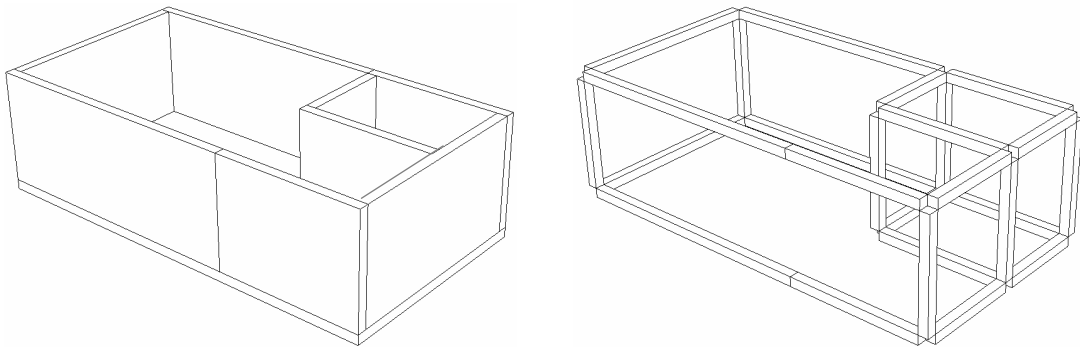


Abbildung 53: Das Ausgangsmodell²⁶ M_C (links), die Verbindungsobjekte des Ausgangsmodells M_{K1} (rechts).

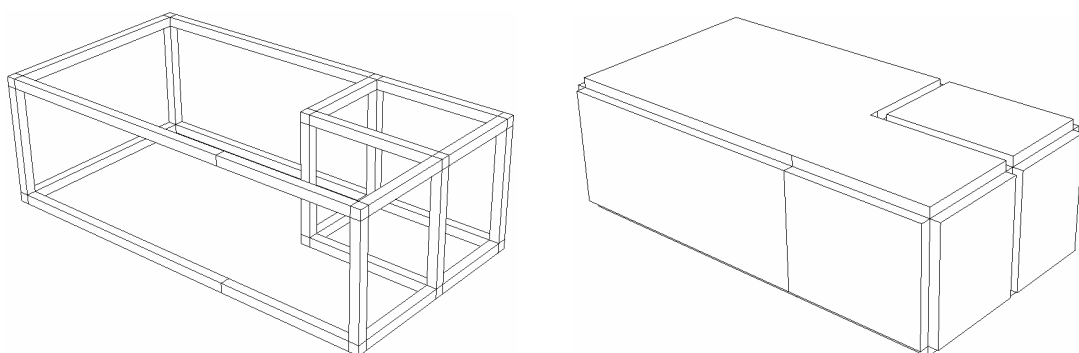


Abbildung 54: Die Verbindungsobjekte des Verbindungsmodells M_{K2} (links), die Differenzobjekte M_D (rechts).

²⁶ Objekte ‚Decke‘ wurden aus Darstellungsgründen ausgeblendet.

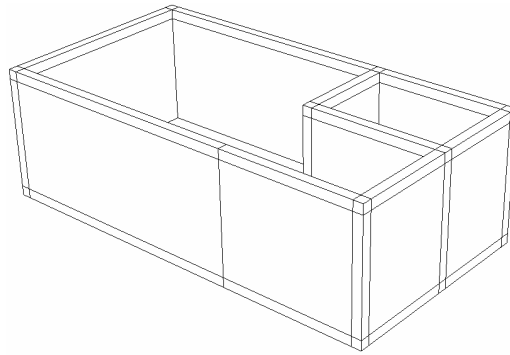


Abbildung 55: Die assemblierten Objektmengen M_D , M_{K1} , M_{K2} werden zum Verbindungsmodell zusammengefügt.

3.3.3 Details zur Implementierung

Wie bereits im Abschnitt 3.3.2 angedeutet, ist das Verfahren zur Dekomposition geometrischer Modelle geeignet, die typischerweise in der Baupraxis relevant sind, wie z.B. aus Scheiben, Platten, Stützen und Balken zusammengesetzte Bauwerke. Aufgrund der Schwierigkeit der Bestimmung von geeigneten ‚Separator‘-Objekten ist das Verfahren allerdings nicht allgemeingültig hinsichtlich beliebiger Bauwerksgeometrie. Werden mit dem hier beschriebenen Verfahren Bauwerke mit monolithischer Bauweise behandelt (z.B. Bürogebäude mit Tragstrukturen aus Beton), sind die Einschränkungen jedoch von untergeordneter Bedeutung.

Ausgeschlossen werden:

- Objekte, die sich flächig berühren aber zueinander versetzt sind. Dies würde zu einer Schichtung in Plattendicke führen (Abbildung 56).
- Objekte, deren Seitenflächen aufeinander liegen (Abbildung 57).
- Objekte, die sich gegenseitig durchdringen (Abbildung 58).
- Stabförmige Objekte (Stützen) mit anderen als rechteckigen Querschnitten.
- Objekte die keine ebenen Oberflächen besitzen²⁷.
- Platten- bzw. scheibenartige Objekte mit variabler Dicke (veränderlicher Wandquerschnitt)²⁷.

²⁷ Grundsätzlich lassen sich diese Beschränkungen teilweise oder vollständig aufheben. Dies wäre aber mit einem erheblichen Modellierungs- und Implementierungsaufwand verbunden.

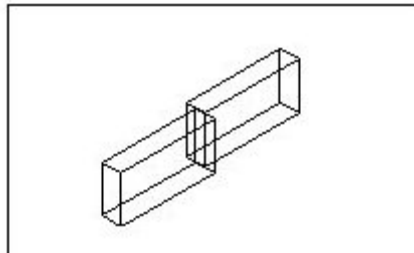
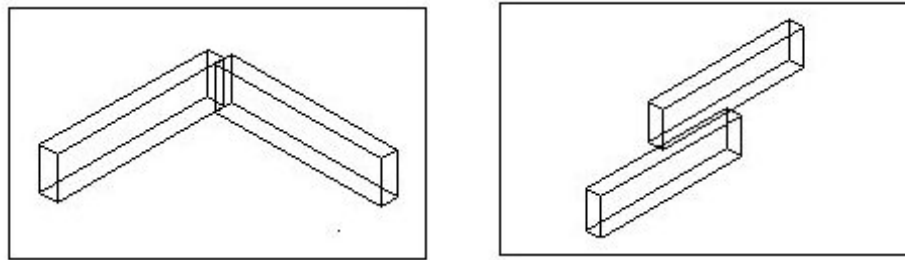


Abbildung 56: Körper mit flächigem Kontakt, jedoch zueinander versetzt.

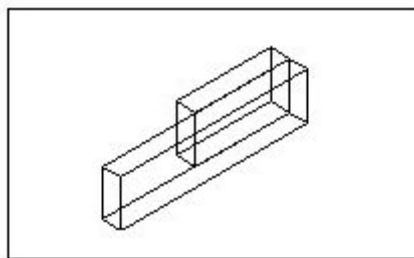


Abbildung 57: Körper, deren Seitenflächen sich berühren.

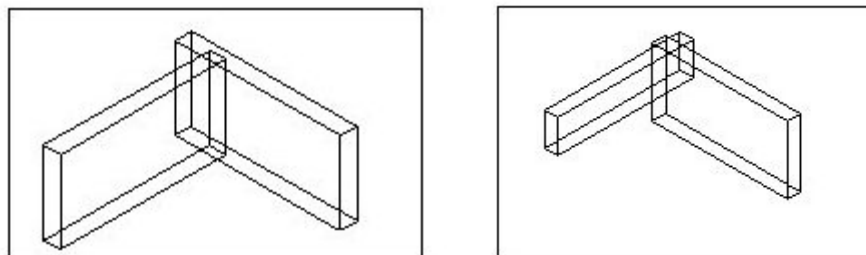


Abbildung 58: Körper, die sich gegenseitig durchdringen.

Bestimmung von Relationen zwischen Körpern (Schnittflächen)

Eine große Anzahl von Bauteilen in üblichen Gebäudemodellen haben keine Relationen hinsichtlich einer geometrischen Nachbarschaft. Zur Minimierung des Rechenaufwandes bei der Bestimmung von Schnittmengen zwischen jeweils zwei betrachteten Körpern hat es sich als vorteilhaft herausgestellt, die Menge der detailliert zu untersuchenden Körper mit einem vor geschalteten Box-Test zu reduzieren. Dabei wird für jedes zu untersuchende Objekt der kleinst-mögliche achsenparallele Quader gesucht, der das Objekt vollständig beinhaltet („bounding box“). Durch einfachen Koordinatenvergleich lässt sich damit sehr effizient feststellen, ob die zwei umschreibenden Quader der betrachteten Körper eine Schnittmenge haben. Im Fall der leeren Menge wird das Objekt-Paar verworfen. Haben die zwei Quader eine nichtleere Schnittmenge, wird der nachfolgende Intersektions-Graph zu exakten Analyse der Nachbarschaft generiert.

Der Intersektions-Graph

Die Relation zweier Objekte zueinander wird mit einem sog. Intersektions-Graph ermittelt. Eingangsdaten sind die BRep-Beschreibungen zweier Körper. Dieser Graph²⁸ ist ein ungerichteter Kantengraph und beinhaltet die Menge der Kanten M_E , die die Schnittfläche der zu untersuchenden Körper beranden.

$$G_I = (M_C, R_E)$$

Ist der Graph G_I leer, d.h. die Körper berühren sich nicht, wird das Objekt-Paar verworfen. Andernfalls werden folgende Fälle unterschieden:

- Der Graph ist azyklisch:
 - Der Graph ist auf einen Knoten reduziert, d.h. die Objekte berühren sich lediglich in einem Knoten. Dieser Fall braucht nicht weiter betrachtet zu werden (siehe Abbildung 59).
 - Der Graph enthält eine Kante, d.h. die Körper haben lediglich eine gemeinsame Kante. Dieser Fall braucht nicht weiter betrachtet zu werden (siehe Abbildung 60).
- Der Graph enthält einen geschlossenen Kantenzug (Zyklus). Damit existiert eine Schnittfläche (siehe Abbildung 61).
- Der Graph enthält mehrere Zusammenhangskomponenten, d.h. die Objekte berühren sich damit an verschiedenen Stellen. Die Schnittmengen können gemeinsame Knoten, Kanten und Flächen sein. Dieser Fall kann grundsätzlich durch ‚Teilen‘ eines Körpers in einen vorangegangenen Fall überführt werden (siehe Abbildung 62).

²⁸ E = Edge

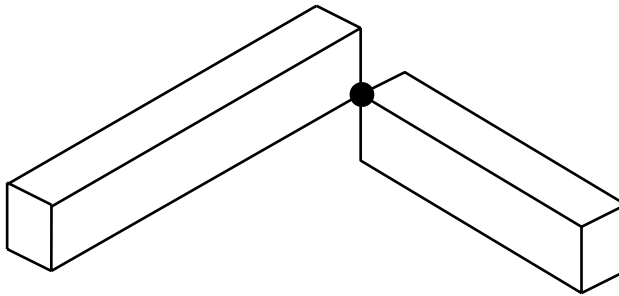


Abbildung 59: Bei dieser Objekt-Konfiguration reduziert sich der Intersektion-Graph zu einem Knoten.

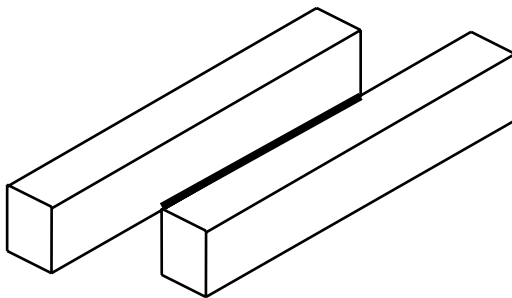


Abbildung 60: Der Intersektion-Graph für diese Konfiguration enthält lediglich eine (nicht kraftschlüssige) Kante.

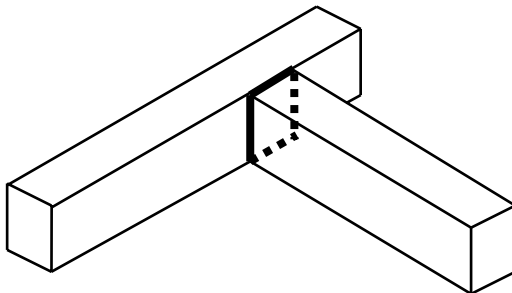


Abbildung 61: Der Intersektion-Graph für diese Konfiguration enthält einen geschlossenen Kantenzug.

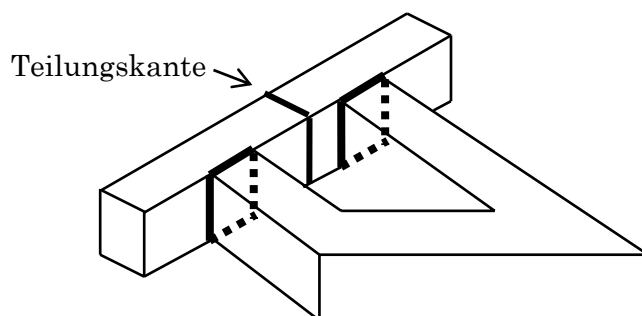


Abbildung 62: Beispiel eines Modells, dessen Intersektion-Graph zwei Zusammenhangskomponenten besitzt.

Erzeugen der Verbindungs-Objekte

Zur Erzeugung von Verbindungsobjekten werden die im Abschnitt 2.2 erläuterten Modellierungsmodelle eingesetzt. Die Körper des ursprünglichen Gebäudemodells können grundsätzlich in beliebigen Formen zusammengesetzt sein. Bei der Untersuchung von Modellen typischer Bürogebäude haben sich jedoch drei wesentliche Problemklassen herausgebildet:

1. Die Bauteile stoßen stumpf aneinander (siehe Abbildung 63).
2. Gegenüberliegende Schnittflächen zu benachbarten Bauteilen liegen versetzt zueinander oder die Schnittflächen haben unterschiedliche Größe, z.B. bei gegenüberliegenden Wänden unterschiedlicher Dicke oder bei Stützen oberhalb und unterhalb von Geschoßdecken mit unterschiedlichen Abmessungen (siehe Abbildung 64).
3. Die Bauteile liegen derart zueinander, dass bei einer einfachen (rechtwinkligen) Extrusion der Schnittfläche Restkörper entstehen würden, z.B. sechsknotige Körper (siehe Abbildung 65).

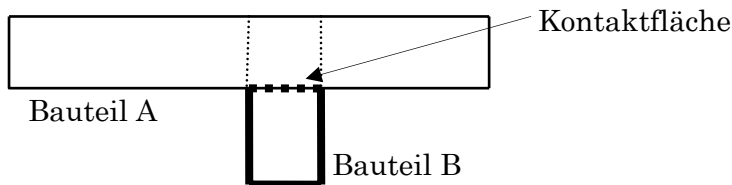


Abbildung 63: Stumpfer Bauteilanschluss.

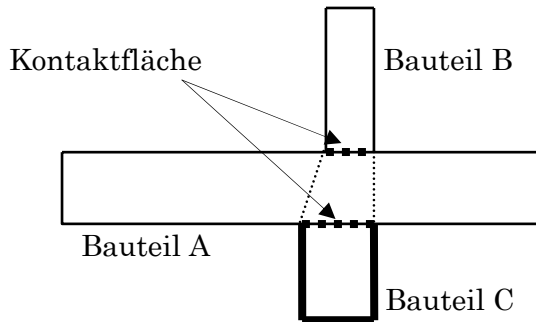


Abbildung 64: Bauteilanschluss mit gegenüberliegenden Schnittflächen.

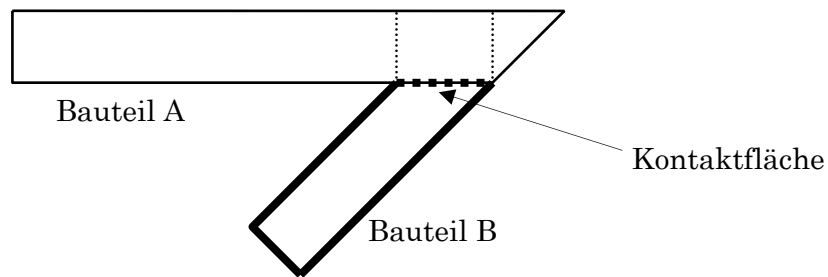
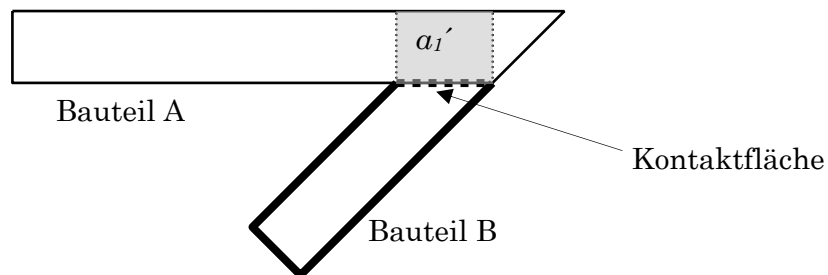


Abbildung 65: Schiefwinkliger Anschluss an ein Wandende.

Verbindungskörper der Problemklasse 1 werden mittels Extrusion der Schnittfläche über die Wanddicke erzeugt. Bei Problemklasse 2 wird durch geradlinige Kurveninterpolation (siehe 2.2.7 ‚Produktionsmodelle‘) ein Körper zwischen beiden Schnittflächen generiert. Verbindungskörper bei Problemklasse 3 werden in zwei Schritten erzeugt. Zuerst wird das in der Problemklasse 1 verwendete Verfahren durchgeführt. Danach wird der temporäre Verbindungskörper a_1' (siehe dazu Punkt 6 des Zerlegungsalgorithmus im Abschnitt 3.3.2) derart in Wandlängsachse verlängert, dass der entstandenen Restkörper vollständig umfasst wird. Die Schnittmenge aus dem erzeugten Körper a_1'' und dem Bauteil A ergibt den Verbindungskörper a_1 .

Abbildung 66: Temporärer Verbindungskörper a_1' .

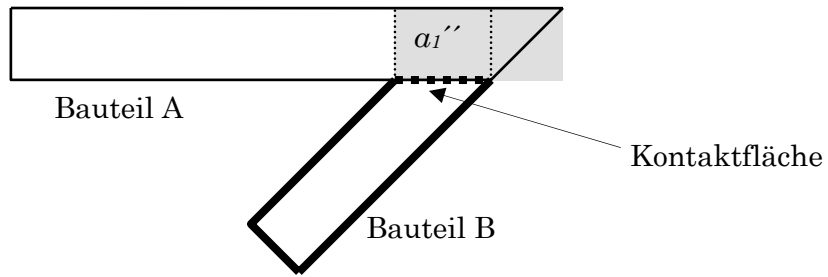


Abbildung 67: Extrusion des Verbindungskörpers a_1'' in Längsrichtung von Bauteil A.

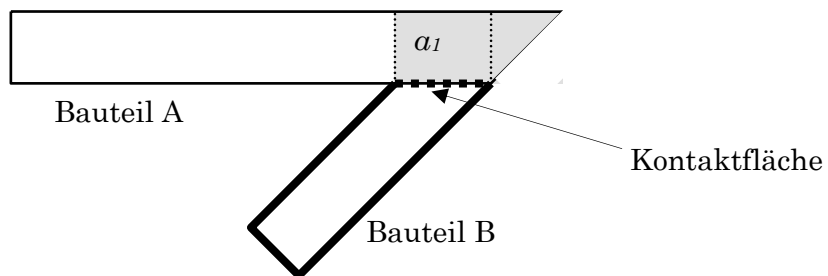


Abbildung 68: Schnittmengenbildung von Bauteil A und Körper a_1'' .

3.4 Automatische makrobasierte Hexaeder Netzgenerierung

3.4.1 Einführung

Die automatische Vernetzung setzt als Eingangsmodell ein Verbindungsmodell voraus (siehe Kapitel 3.3). Zur Vernetzung der Objekte der Mengen M_D , M_{K1} , M_{K2} , M_S ²⁹ werden dabei typabhängig aufeinander abgestimmte Vernetzungsmakros (z.B. Teilung der Objekte in ‚Stablängsachse‘ bei Elementen der Objektmenge M_{K1}) oder Extrusion von 2D-Vierecksnetzen (bei platten-/scheibenartigen Objekten) angewandt. Dabei ist zu berücksichtigen, dass die Vernetzung in Hexaeder besondere Anforderungen an die Objekte der Menge M_{K1} und M_{K2} stellt. Beide Mengen dürfen lediglich Hexaeder-Elemente aufweisen. Die Anwendung der vorgestellten Methode auf Modelle mit den in Kapitel 3.3.2 beschriebenen Einschränkungen garantiert aber die Einhaltung dieser Forderung.

In der nachfolgenden Abbildung ist ein Verbindungsmodell mit den Objektmengen M_D (Objekte d_1, d_2, d_3, d_4), M_{K1} (Objekte a_1, a_2, a_3, a_4, a_5), M_{K2} (Objekt b_1, b_2, b_3) und M_S (Objekt s_1) dargestellt. Dieses (IFC)-Modell dient für das nachfolgende Kapitel 3.4.2 als Referenzmodell zur Erläuterung des Vernetzungsalgorithmus.

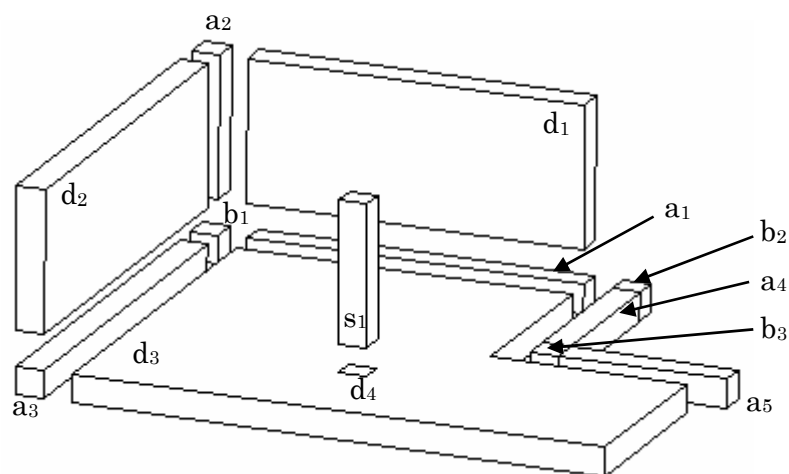


Abbildung 69: Verbindungsmodell nach der Dekomposition mit den Objektmengen M_D , M_{K1} , M_{K2} und M_S .

²⁹ M_S bezeichnet die Menge der Stützenobjekte. Die Typinformationen zu Bauteilen werden dem IFC-Bauwerksmodell entnommen.

Das Vorgehen bei dem hier vorgestellten Vernetzungsverfahren ist nun, die Objekte der Menge M_D (scheibenförmige Bauteile) in zwei Vernetzungsdurchläufen mittels extrudierter 2D-Netze räumlich zu diskretisieren. Im ersten Schritt werden alle Objekte ohne Berücksichtigung angrenzender Objekte ‚freivernetzt‘. Die generierte Diskretisierung der Berandungen benachbarter Objekte sind daher nicht oder nur zufällig kompatibel. In den nachfolgenden Schritten werden die Diskretisierungen zu den jeweils angrenzenden Verbindungsobjekten a_i der Menge M_{K1} übertragen und dort vereinheitlicht (die feinste Auflösung der Koppelkanten ist maßgebend; andere Verfahren wie z.B. eine Dichtefunktion zur Definition der Längen von Elementen an den Koppelkanten wären ebenfalls möglich). Nach der Rückübertragung der ‚vereinheitlichten‘ Diskretisierungen der Objekte a_i auf benachbarte Objekte d_k erfolgt der zweite Vernetzungslauf der Objekte d_k , nun aber mit vorgegebener Diskretisierung³⁰ der Objektberandungen. Die grundlegende Idee bei diesem Verfahren ist folgende: Beim ersten 3D-Vernetzungslauf wird über den als Teilschritt durchgeführten adaptiven 2D-Vernetzungslauf eine Bündelung der Informationen zur Diskretisierung der gesamten Region am Regionsrand (in 2D) und damit in den Verbindungselementen a_i (in 3D) vorgenommen. Durch die Speicherung der Informationen zur Diskretisierung aller angrenzenden Differenzobjekte d_k ist nun die Möglichkeit gegeben, allein durch die richtige Diskretisierung (Teilung) der Verbindungselemente a_i die Netzgenerierung für die gesamte Struktur zu steuern und dabei vernetzungsrelevante geometrische Besonderheiten (z.B. Stützenanschluss) in den Differenzobjekten d_k zu berücksichtigen. Die Vernetzung einer 3D-Struktur wird damit in eine 1D-Vernetzungproblematik überführt. Der überwiegende Teil der nachfolgend dargestellten Schritte behandelt die ‚Vereinheitlichung‘ der Diskretisierungen. Die anderen Schritte (z.B. Schritt 5) sind notwendig, um die Kompatibilität der Teilnetze zu gewährleisten.

Die Vernetzung des Verbindungsmodells in Hexaeder läuft in folgenden Schritten ab:

1. Vernetzung des Koppelbereichs von Stützen und Platten mittels (Stützen)-Makro zur feineren numerischen Auflösung. Im Modell von Abbildung 69 für Objekt d_3 und d_4 .
2. Vernetzung der Objekte d_k der Menge M_D mit einem 2D-Vierecksvernetzer und anschließender Extrusion der Viereckselemente zu Hexaederobjekten.
3. Übertragung der generierten Diskretisierung der Objekte d_k zu den benachbarten Verbindungsobjekten a_i sowie Löschung der generierten Hexaederelemente aus Schritt 2.
4. Teilung der Verbindungsobjekte a_i der Menge M_{K1} .
5. Verfeinerung der Verbindungsobjekte b_j der Menge M_{K2} bei einspringenden Ecken.
6. Übertragung der Diskretisierungen der Verbindungsobjekte a_i zu angrenzenden Differenzobjekten d_k der Menge M_D .

³⁰ Vorgegebene Diskretisierungen werden nur für ‚geteilte‘ Kanten verwaltet (‚geteilt‘ im Sinne von ‚shared‘).

7. Verfeinerung der Verbindungsobjekte a_i der Menge M_{K1} .
8. Zweiter Lauf zur Vernetzung der Objekte d_k der Menge M_D mit einem 2D-Vierecksvernetzer unter Berücksichtigung der vorgegebenen Diskretisierungen der Objektberandungen.
9. Assemblierung der Teilmodelle.

3.4.2 Teilschritte der Hexaeder-Vernetzung

Das Stützenmakro (Schritt 1)

Das Makro wird auf alle Objekte der Menge M_S angewandt (Im Referenzmodell enthält die Menge M_S nur das Objekt s_1). Für jedes Objekt s_n der Menge M_S wird der Nachbar d_b' und dessen Nachbar d_b'' , also der Körper, der d_b' an vier Seiten umschließt, aus der Objektmenge M_D identifiziert.

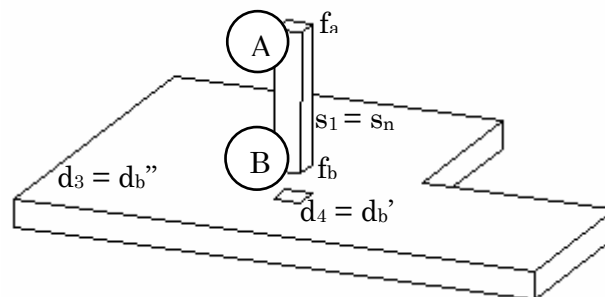


Abbildung 70: Teilmodell des Referenzmodells mit den Objekten s_1 , d_b' und d_b'' und der Verbindungsfläche f_a (a = Stützenkopf) und f_b (b = Stützenfuß) von s_1 und d_b' bzw. s_1 und d_b'' .

Die Objekte d_a' und d_a'' der Menge M_D im Bereich A der Stütze sind aus Gründen der Übersichtlichkeit im Referenzmodell aus Abbildung 69 nicht dargestellt. Die nachfolgend beschriebene Methodik zur Generierung von Hexaedern zur feineren Auflösung des Platten/Stützen-Kopplungsbereichs gilt für den Bereich A entsprechend.

Das Stützenmakro beinhaltet die drei Teilschritte:

- Verfeinerung der Stütze s_n .
- Verfeinerung des Verbindungsobjekts d_b'
- Generierung neuer Hexaederobjekte als Teilmodelle des Differenzobjekts d_b'' .

Zur Verfeinerung der Stütze wird zunächst die Kontaktfläche f_b mittels Halbierung³¹ der Berandungskanten in vier Teile f_{b1}' , f_{b2}' , f_{b3}' , f_{b4}' geteilt (siehe dazu Abbildung 71 links

³¹ Aufgrund der Vernetzungsmethode wird jede Modellkante in eine gerade Anzahl von Kanten jedoch mindestens einmal geteilt. Siehe dazu Kapitel 2.3.

oben) und danach jeweils um den Wert d stückweise in Richtung der Verbindungsfläche f_a extrudiert. Der Wert d ist dabei:

$$d = (l_s / n) \quad (15)$$

mit

$$n = \text{int}(l_s / l_e) + 1 \quad (16)$$

l_s : Stützenlänge,

l_e : maximale zulässige Elementkantenlänge,

n : Teilungszahl; wird zur nächst größeren Ganzzahl aufgerundet.

Die Diskretisierung des Objekts d_b' erfolgt nun ebenfalls auf Basis der in die vier Teile f_{b1}' , f_{b2}' , f_{b3}' , f_{b4}' geteilten Fläche f_b . Die Flächen werden mit dem Wert der Ausdehnung von d_b' in Richtung der Stützenlängsachse (die Plattendicke) extrudiert (siehe dazu auch Kapitel 3.3.3).

Für die Generierung der Hexaeder zur Verfeinerung der Platten werden zunächst die Knoten p_2 , p_3 und p_4 generiert (siehe Abbildung 71). Dabei ist:

$$p_2 = p_1 + \overrightarrow{p_1 p_2'} * z_p \quad (17)$$

$$p_3 = p_1 + \overrightarrow{p_1 p_3'} * z_p \quad (18)$$

$$p_4 = p_2 + \overrightarrow{p_2 p_3'} / 2 \quad (19)$$

wobei z_p standardmäßig so gewählt wird, dass der Betrag von $\overrightarrow{p_4' p_4}$ der Dicke der Scheibe entspricht. Siehe dazu [Rücker 01].

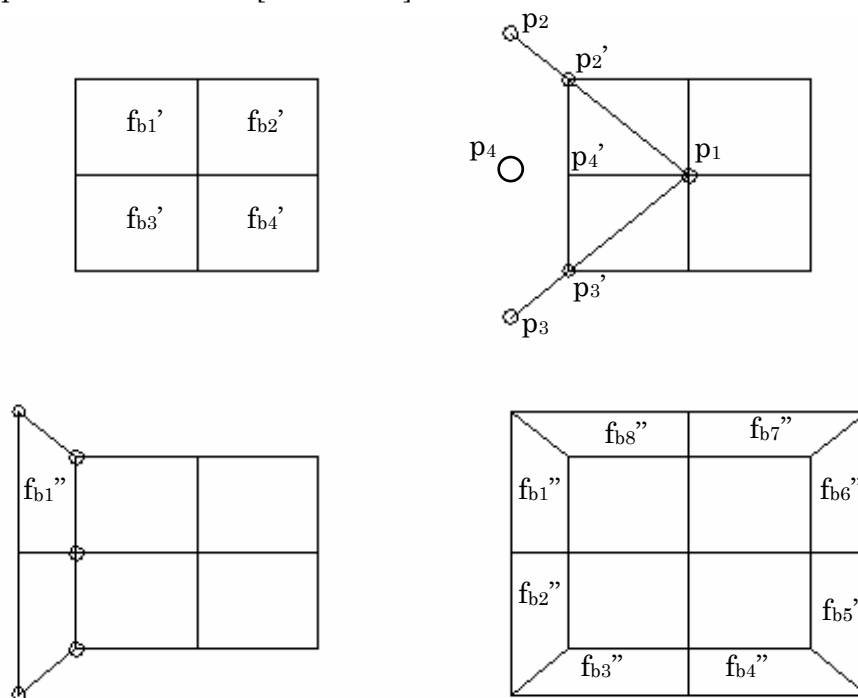


Abbildung 71: Schrittweise Generierung der Vierecksflächen zur Stützenverfeinerung.

Die Fläche f_{b1} ergibt sich dann aus den Knoten p_4' , p_2' , p_2 und p_4 . Entsprechendes gilt für die weiteren 7 Flächen (siehe Abbildung 71 rechts unten). Die neu generierten Flächen werden nun wiederum durch Extrusion über die Objektdicke d_b zu Hexaedern erweitert. Abschließend werden die Volumina der acht neu generierten Hexaeder mittels des Mengenoperators \setminus_b von d_b abgetrennt.

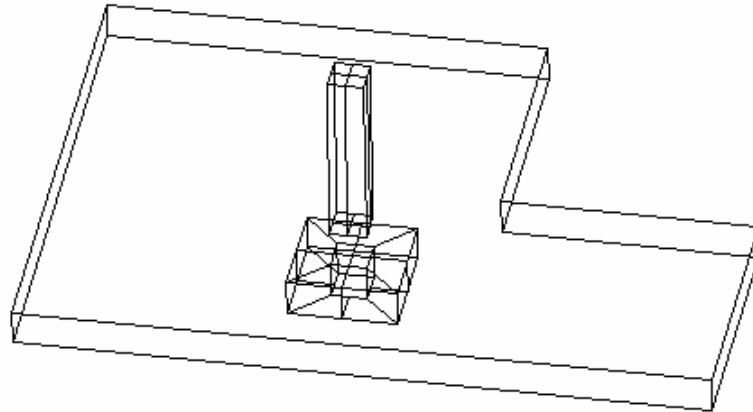


Abbildung 72: Diskretisierter Kopplungsbereich von Platte und Stütze des Referenzmodells. (In der Abbildung sind nicht alle Kanten dargestellt).

Hexaeder-Vernetzung der Objekte d_k der Menge M_D (Schritt 2)

Die Freivernetzung umfasst alle Objekte der Menge M_D . Der Algorithmus zur Vernetzung wird anhand des Objekts d_u stellvertretend für alle weiteren Objekte der Menge M_D erläutert (siehe Abbildung 73 oben links). Zuerst wird das Objekt analysiert und die beiden größten Flächen f_{m1} und f_{m2} identifiziert³². Der Abstand der Flächen f_{m1} und f_{m2} ergibt die Objektdicke d_{fm} . Für das zu vernetzende Objekt wird eine Parameterebene mit dem lokalen Koordinatensystem $\overline{u, v}$ generiert. Nun wird jeder Knoten der Flächen-Berandung im Koordinatensystem der Parameterebene abgegriffen, danach folgen die Objektberandungen im Inneren des Objekts (Löcher). Entsprechend der Festlegung für die Freivernetzung (siehe Kapitel 2.3) wird die äußere Objektberandung im mathematisch positiven Sinne durchlaufen, die Objektberandungen im Inneren des Objekts im mathematisch negativen Sinne. Mit den abgegriffenen 2D-Knotenkoordinaten kann der Vierecks-Freivernetzer aufgerufen werden. Die Knoten der generierten Vierecke werden in das Objektkoordinatensystem transformiert und mit dem Wert der Objekt-Dicke d_{mf} extrudiert. Da der Verarbeitungsschritt für alle Differenzobjekte unabhängig durchgeführt wird, sind die Berandungen der Objekte der Menge M_D nun an den Koppelkanten auf Grund der Freivernetzung nicht oder nur zufällig kompatibel (siehe Abbildung 73).

³² Bei Platten und Scheiben sind zwei Flächen groß im Vergleich zu allen anderen Flächen. Weiterhin sind die beiden größten Flächen zueinander parallel angeordnet und der Schichtungsvektor liegt jeweils parallel zum Normalenvektor der Flächen.

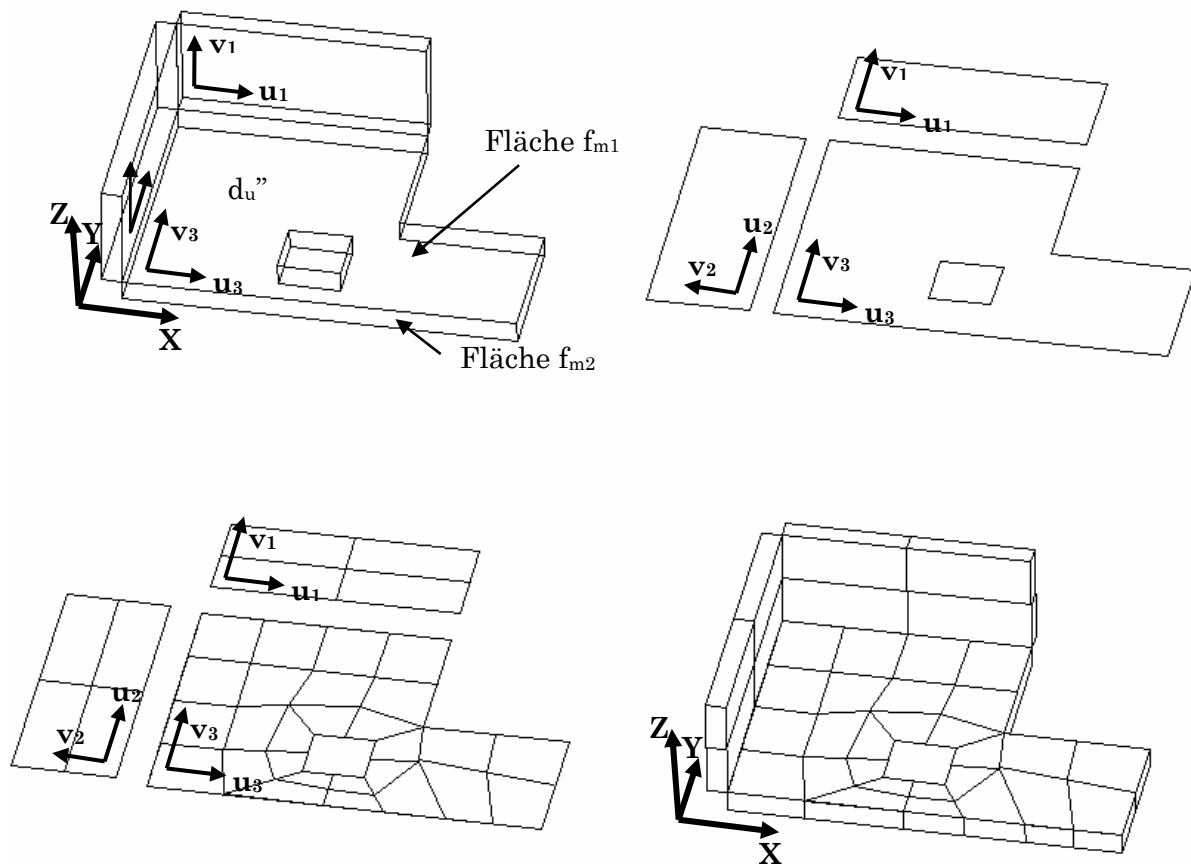


Abbildung 73: Vernetzung der Differenzobjekte der Menge M_D mittels eines 2D-Freivernetzer.

Übertragung der Diskretisierung auf Verbindungsobjekte a_i (Schritt 3)

Die Diskretisierung der Objektberandungen wird jetzt auf die benachbarten Verbindungselemente a_i der Menge M_{K1} übertragen (siehe Abbildung 74 links). Da Verbindungselemente per Definition Hexaeder sind, können maximal vier Differenzobjekte d_k zu jedem Verbindungsobjekt a_i benachbart sein. Bei Objekt a_2 ist die Diskretisierung bereits kompatibel, eine Vereinheitlichung ist hier nicht notwendig. Die Untergliederung der Objekte a_3 und a_1 wird jeweils auf die feinere Diskretisierung abgeändert, die von Objekt d_3 vorgegeben ist. Nach der Übertragung der Diskretisierung werden die Hexaederelemente aus der Freivernetzung nicht mehr benötigt und werden deshalb gelöscht.

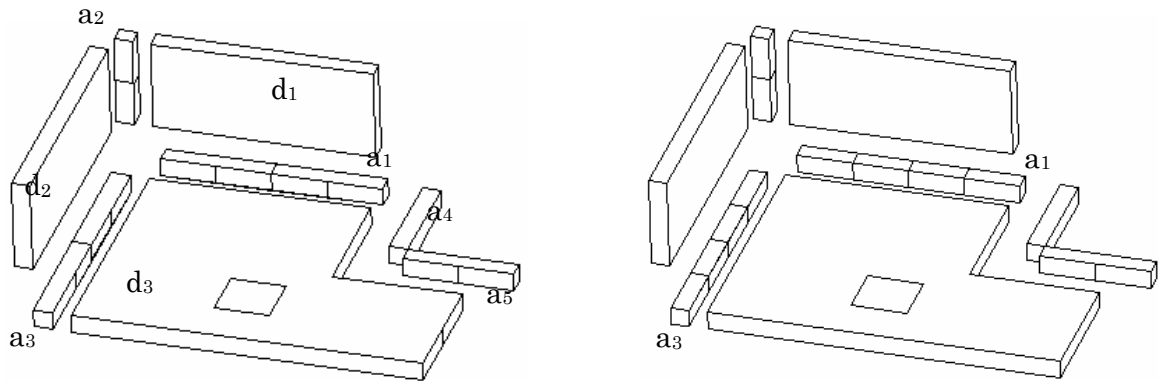


Abbildung 74: Inkompatible Diskretisierung der Objekte a_1 und a_3 (links), kompatible Diskretisierung (rechts).

Verfeinerung der Verbindungsobjekte b_j (Schritt 4)

Die Verfeinerung von Verbindungsobjekten b_j der Menge M_{K2} wird notwendig, da für eine Freivernetzung mit der in Kapitel 2.3 beschriebenen Methode jede Kante mindestens einmal unterteilt wird (zusätzliche Knoten, die bei der Konvertierung von zwei Dreiecken zu vier Vierecken eingefügt werden). Die Verfeinerung wird durchgeführt, wenn Objekte der Menge M_{K2} und Objekte der Menge M_D zwei gemeinsame Verbindungsflächen haben³³. Diese Bedingung kann über den Objektgraph getestet werden. Hintergrund für diese Vorgehensweise ist wiederum die Forderung aus dem Verfahren zur Freivernetzung, Regionen mit einer geraden Kantenzahl zu erstellen. Durch Teilung der Kanten k_d' und k_d'' ist die Forderung für das Verbindungsobjekt b_3 (Abbildung 69) erfüllt.

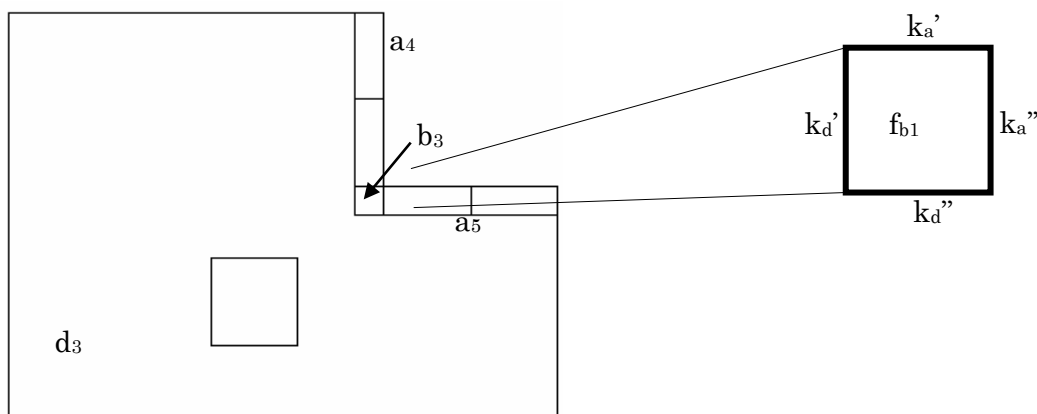


Abbildung 75: Teilmodell des Referenzmodells mit dem Objekt b_3 sowie dessen Nachbarobjekte a_4 , a_5 und d_3 .

Zuerst werden diejenigen Flächen f_{b1} und f_{b2} des Hexaeders b_3 identifiziert, bei denen zwei Kanten in Nachbarschaft zu Objekt d_3 und zwei Kanten in Nachbarschaft zu zwei Objekten der Menge M_{K1} liegen. Diese Bedingung trifft bei Objekt b_3 auf zwei Flächen

³³ Anschaulich betrachtet haben die Objekte d_3 und b_3 zwei gemeinsame Flächen (siehe Abbildung 75).

zu. Der Abstand $d_{f_{b1}, f_{b2}}$ der Flächen f_{b1} und f_{b2} ist der Wert, der später zur Extrusion herangezogen wird. Eine der Flächen f_{b1} , f_{b2} wird für die weitere Bearbeitung ausgewählt.

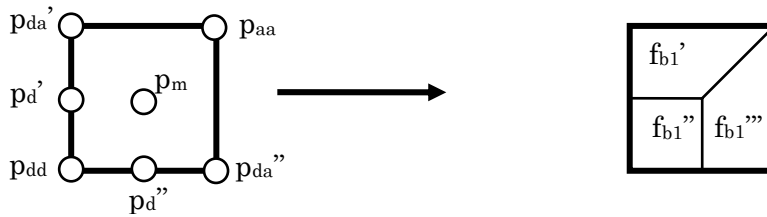


Abbildung 76: Verfeinerung von Objekten der Menge M_{K2} bei einspringenden Ecken.

Auf Basis der gegebenen Knoten $p_{da'}$, p_{dd} und $p_{da''}$ ergeben sich die Knoten p_d' , p_m und p_d'' wie folgt:

$$p_d' = p_{da'} + \overline{p_{da'} p_{dd}} / 2 \quad (20)$$

$$p_d'' = p_{da''} + \overline{p_{da''} p_{dd}} / 2 \quad (21)$$

$$p_m = p_{aa} + \overline{p_{aa} p_{da'}} / 2 + \overline{p_{aa} p_{da''}} / 2 \quad (22)$$

Durch Verknüpfung von jeweils vier Knoten ergeben sich dann die Flächen f_{b1}' , f_{b1}'' und f_{b1}''' . Diese werden durch Extrusion mit dem Vektor $v_{f_{b1}, f_{b2}}$ zu Hexaedern erweitert. Die Normalenvektoren kennzeichnen die Verbindungsfläche der Objekte b_3' , b_3'' und b_3''' zu Objekt d_k (hier Objekt d_3). Die Vektoren werden auf die über die Flächen f_{b1}' , f_{b1}'' und f_{b1}''' bzw. f_{b2}' , f_{b2}'' und f_{b2}''' benachbarten Objekten der Menge M_{K1} übertragen (siehe Schritt 7 weiter unten). Sie dienen dort als Hilfsmittel zur Verfeinerung der Diskretisierung.

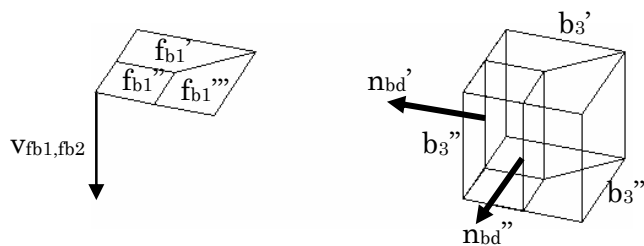


Abbildung 77: Extrusion der Flächen f_{b1}' , f_{b1}'' und f_{b1}''' um den Wert $d_{f_{b1}, f_{b2}}$.

Teilung der Verbindungsobjekte a_i (Schritt 5)

Die Vereinheitlichung der Diskretisierung bei Objekten a_i der Menge M_{K1} hinsichtlich der zwei, drei oder vier angrenzenden Differenzobjekte d_k wurde in Schritt 3 bereits vorgenommen. Daraus resultiert im einfachsten Fall eine äquidistante Teilung der Objekte a_i oder bei Anwendung anderer Algorithmen zur Vereinheitlichung der Diskretisierung (z.B. Verteilungsfunktion) eine Teilung in unterschiedlich lange Teilobjekte³⁴.

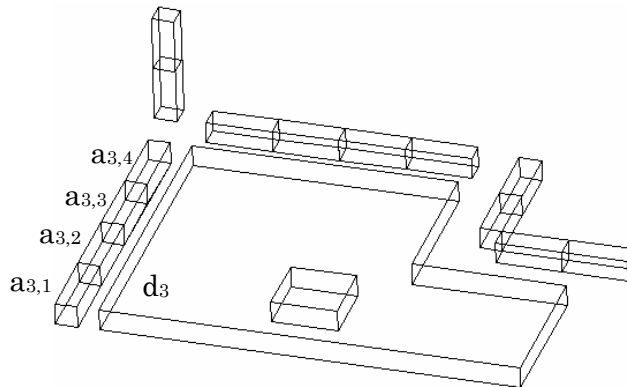


Abbildung 78: Teilung der Verbindungsobjekte a_i nach vorgegebener Diskretisierung aus Schritt 3.

Übertragung der Diskretisierung der Verbindungsobjekte a_i zu Differenzobjekten d_k (Schritt 6)

Die Übertragung der Diskretisierung geschieht beispielsweise durch Projizierung der Fläche f_{ad} des Verbindungsobjekts $a_{3,1}$ auf die Berandung von d_3 (siehe Abbildung 79). Dabei werden beim Objekt d_3 zwei bzw. drei neue Kanten sowie eine neue Fläche f_{ad}' generiert. Das topologische Element Loop l_a der Fläche f_{d3} (siehe Kapitel 3.2.2) des Objekts d_3 erweitert sich um eine Kante. Der vor der Projizierung durchgeführte Test zu Bestimmung von Lagebeziehungen zwischen den beiden Objekten d_3 und $a_{3,1}$ entfällt hier, da aus dem Datenmodell (Objektgraph für die Nachbarschaften und ACIS-Body für die Flächen) unmittelbar die zu verknüpfenden Objekte und deren Verbindungsflächen f_{ad} bzw. f_{ad}' ermittelt werden. Die Projektion der Flächen der Verbindungsobjekte a_i auf die Objekte d_k wird nun über alle Verbindungsobjekte a_i durchgeführt (siehe Abbildung 80). Der Vorteil dieser Vorgehensweise ist, dass für die Erstellung der 2D-Regionen zur Vierecksvernetzung dasselbe Verfahren wie zur Freivernetzung angewendet werden kann, d.h. das Durchlaufen der Loop l_a der Fläche f_{d3} und das Abgreifen der Koordinaten über eine Parameterebene sind identisch.

³⁴ Die Teilung erfolgt immer mit einer geraden Anzahl jedoch in mindestens zwei Teilobjekte. Diese Randbedingung hat ihre Ursache in der Freivernetzung der Differenzobjekte d_k (siehe dazu Kapitel 2.3). Die Übertragung der Diskretisierung auf die Objekte a_i führt damit ebenfalls zu mindestens einer Teilung der Objekte a_i .

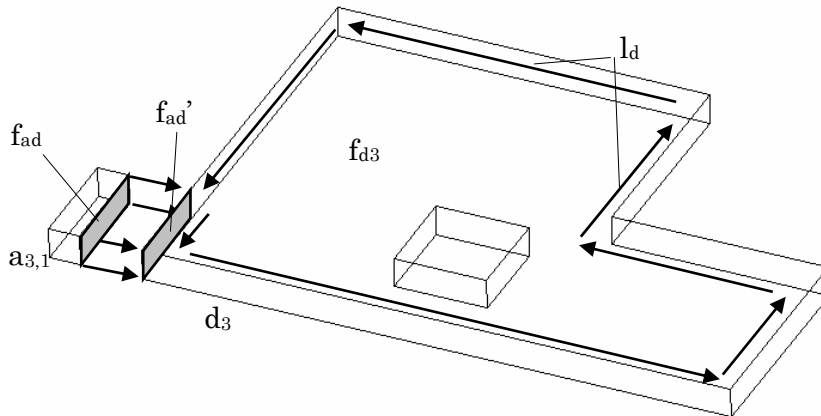


Abbildung 79: Projektion der Flächen des Verbindungsobjekts a_i auf die Nachbarobjekte d_k . f_{d3} bezeichnet dabei die Oberseite der Platte.

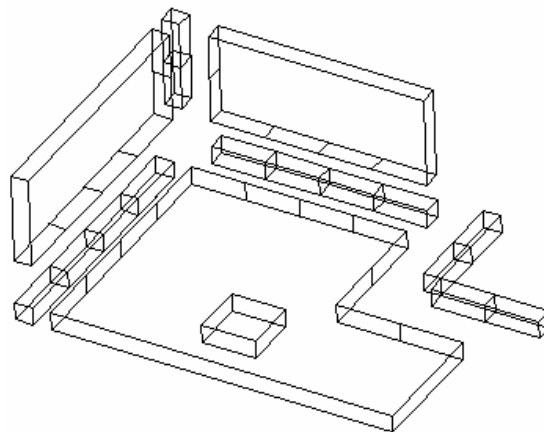


Abbildung 80: Übertragung der Flächen von Verbindungsobjekten a_i auf Differenzobjekte d_k .

Verfeinerung der Verbindungsobjekte a_i (Schritt 7)

Ergibt sich aus der Topologie des Bauwerksmodells die Situation, dass Verbindungsobjekte b_j im Schritt 3 feiner diskretisiert werden müssen (einspringende Ecken), entstehen temporär inkompatible Diskretisierungen zwischen Objekten a_i und b_j . Im Referenzmodell von Abbildung 69 ist diese Situation nicht dargestellt, sondern wird anhand des Modells der nachfolgenden Abbildung erläutert. Im linken Modell in dieser Abbildung ist die Inkompatibilität zwischen der Verbindungsflächen $f_{a_1 b_1}$ von Objekt a_1 und den benachbarten Flächen $f_{b_1 a_1}'$, $f_{b_1 a_1}''$ und $f_{b_1 a_1}'''$ der Objekte b_1' , b_1'' und b_1''' sichtbar. In Schritt 3 wurden die Objekte der Menge M_{K1} (hier Objekt a_1 und a_2), die benachbart sind zu feiner aufgelösten Objektberandungen der Objekte b_j (hier $f_{b_1 a_1}'$, $f_{b_1 a_1}''$ und $f_{b_1 a_1}'''$), bereits mit den Normalenvektoren n_{bd}' bzw. n_{bd}'' markiert. Diese legen die Orientierung der zu verfeinernden Objekte a_i fest. Damit lässt sich zur Verfeinerung der Objekte a_i dasselbe Makro wie in Schritt 3 für die Verfeinerung der Objekte b_j (siehe rechtes Modell in der nachfolgenden Abbildung) einsetzen.

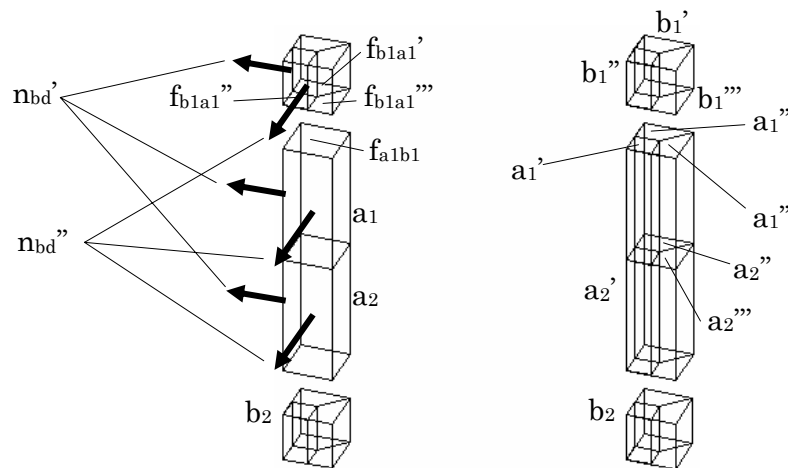


Abbildung 81: Verfeinerung der Diskretisierung der Verbindungsobjekte a_i .

Hexaeder-Vernetzung der Objekte d_k der Menge M_D (Schritt 8)

Die Vernetzung der Objekte der Menge M_D läuft analog zu Schritt 2 ab (siehe oben), d.h. die Koordinaten der Berandung werden über lokale 2D-Parameterebenen abgegriffen. Der einzige Unterschied liegt in der Behandlung von Kanten, die durch eine Übertragung der Diskretisierung von Objekte a_i generiert wurden (hervorgehobene Kanten in Abbildung 82 oben rechts) und bereits als Netzkanten feststehen. Aufgrund der Vernetzungsmethode bei der Erzeugung von Viereckselementen, bei der jede Regionskante in eine gerade Anzahl jedoch mindestens einmal unterteilt wird, ergibt ein Kantentupel³⁵ des Objekts d_k eine Regionskanten für die Freivernetzung.

³⁵ Die Teilung der Objektkanten in eine gerade Anzahl von Teilkanten ermöglicht die Bildung von Kantentupeln.

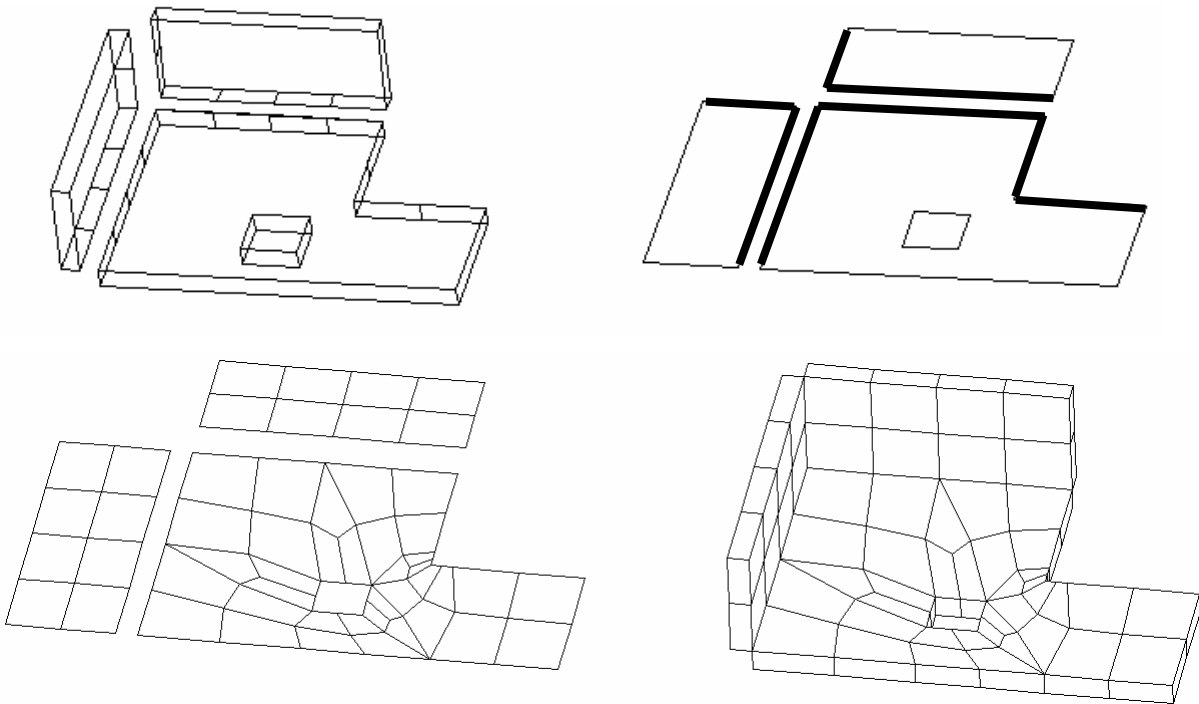


Abbildung 82: Vernetzung der Differenzobjekte d_k der Menge M_D mit teilweise vorgegebener (hervorgehobene Kanten) und teilweise freier Regionsberandung.

Assemblierung der Teilmodelle (Schritt 9)

Zuletzt werden die generierten Objekte der Teilmodelle aus den Schritten 1 bis 8 zusammengefasst und ergeben zusammen das Netzmodell.

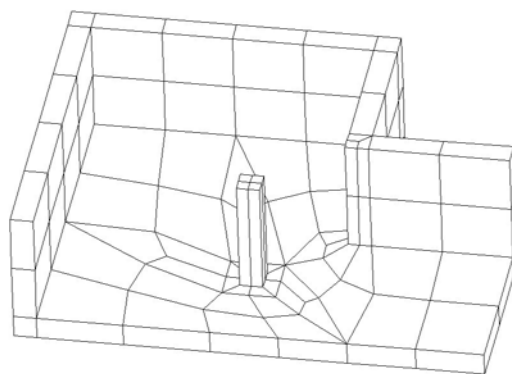


Abbildung 83: Assemblierung der Teilmodelle zum Gesamtmodell.

3.4.3 Differenzobjekte in Dickenrichtung vernetzen

Bisher wurden Modelle gezeigt, bei denen die platten- und scheibenartigen Strukturen (Differenzobjekte) mit einem Element in Dickenrichtung diskretisiert wurden. Es gibt aber Gebäudestrukturen, bei denen besondere Anordnungen von Differenzobjekten im Modell es erforderlich machen, diese in Dickenrichtung zu unterteilen. Der Grund dafür liegt in der Vernetzungstechnik und dort speziell in der Konvertierung von Dreiecksnetzen in Vierecksnetzen (siehe dazu auch Kapitel 2.3).



Abbildung 84: Raum mit einspringender Wand.

Bei der Vernetzung wird auf einer Regionskante, im Beispiel aus Abbildung 84, 85 an der Stirnfläche der Innenwand, ein Zwischenknoten generiert. Das macht eine Unterteilung des gesamten Differenzobjekts in Dickenrichtung erforderlich (siehe nachfolgende Abbildung).

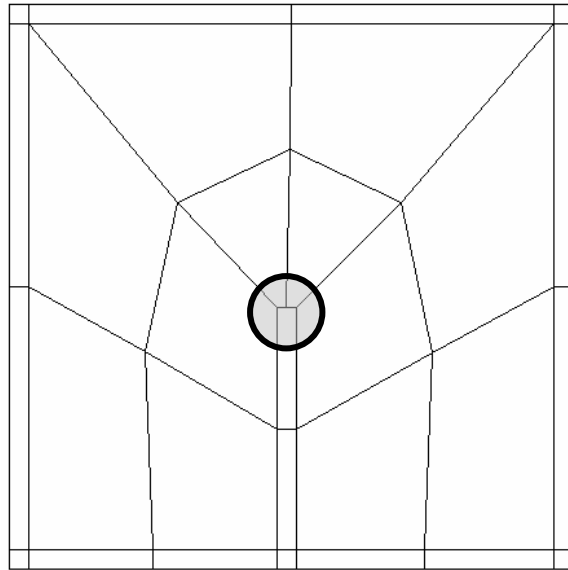


Abbildung 85: Draufsicht auf die vernetzte Struktur des Modells aus Abbildung 84.

Die in Dickenrichtung eingefügten Zwischenknoten führen letztendlich dazu, dass ohne besondere Maßnahmen alle platten- und scheibenartigen Strukturen des Modells in Dickenrichtung unterteilt werden müssten. Dies führt jedoch zu erheblichen Nachteilen beim Rechenaufwand (nahezu doppelte Elementanzahl). Eine mögliche Lösung liegt darin, angrenzende Verbindungselemente zu untersuchen, ob sich dort die Teilung über die Plattendicke durch Zwischenelemente aufheben lässt (siehe Abbildung 86). Dieser Test muss rekursiv über alle Platten und Verbindungselemente durchgeführt werden, bis entweder Zwischenelemente platziert werden können oder aber alle Differenzobjekte in Dickenrichtung unterteilt sind.

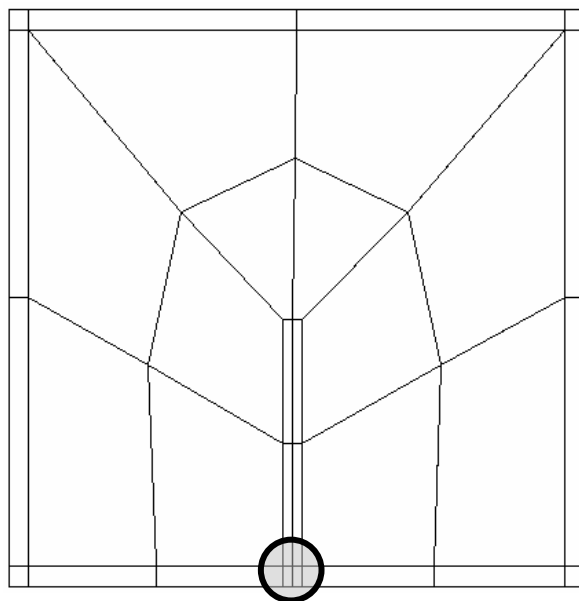


Abbildung 86: Zwischenelement zur Sicherung der Kompatibilität der Teilnetze.

4 Modellbeispiel

Abbildung 91 zeigt einen Teil eines Bürogebäude mit den Abmessungen in der Grundfläche von 30×40 Meter. Das Beispiel ist einem Bauzustand des 21 Stockwerke hohen Uniqa-Tower in Wien nachempfunden. Die Tragstruktur besteht aus einer Stützen/Platten/Scheiben-Beton-Konstruktion mit zwei innen liegenden Gebäudekernen. Der hier nicht dargestellte untere Bereich des Gebäudes ist durch ein integriertes Atrium komplizierter als die darüber liegenden gleichartigen Geschosse. Für die Anwendung des hier vorgestellten Simulationsverfahrens wurden aufgrund der Komplexität des Modells, der Objektanzahl und der daraus resultierenden Größe des zu lösenden Gleichungssystems fünf Stockwerke aus dem oberen Bereich extrahiert.

Technische Angaben: Belastung (Eigengewicht) mit $25\text{E-}06 \text{ kN/m}^3$, E-Modul: 30500 MN/m^2 , Querdehnzahl $\mu = 0,2$. Dabei wird ein homogenes linear-elastisches Materialverhalten angenommen. Der Polynomgrad über die Elementdicken ist für alle Berechnungen konstant auf $q = 2$ festgelegt.

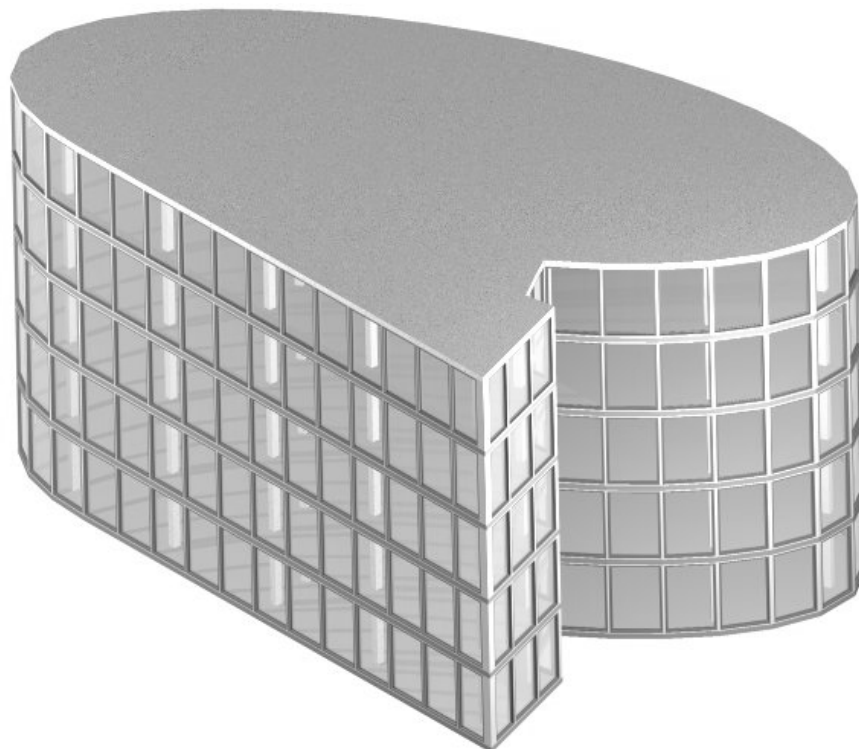


Abbildung 87: IFC-Modell eines Teils des Uniqa-Towers mit fünf Stockwerken.

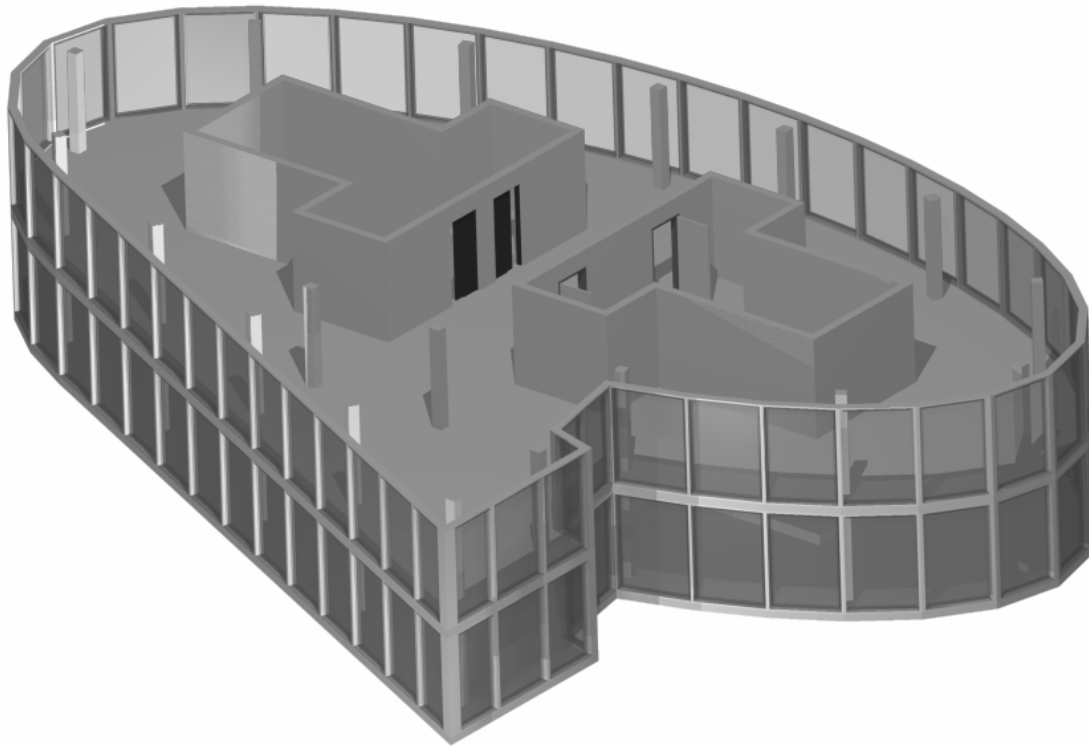


Abbildung 88: Teilmodell mit der Darstellung der inneren Struktur.

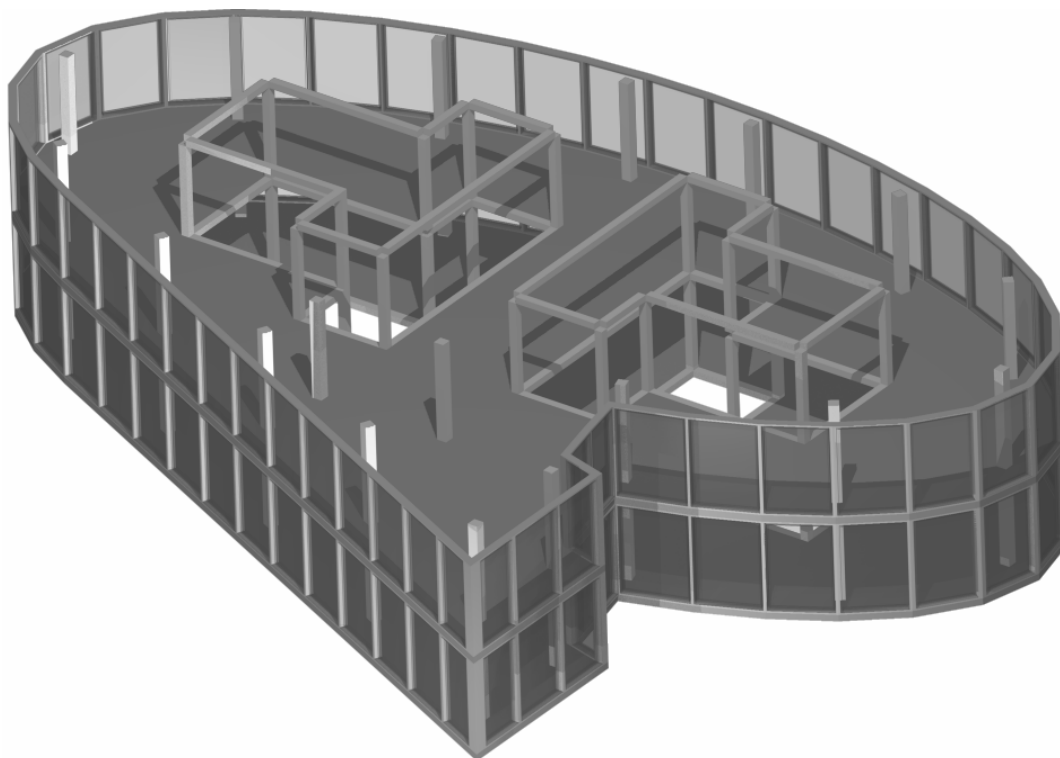


Abbildung 89: Modell mit Verbindungsobjekten des Gebäudekerns sowie Stützen.

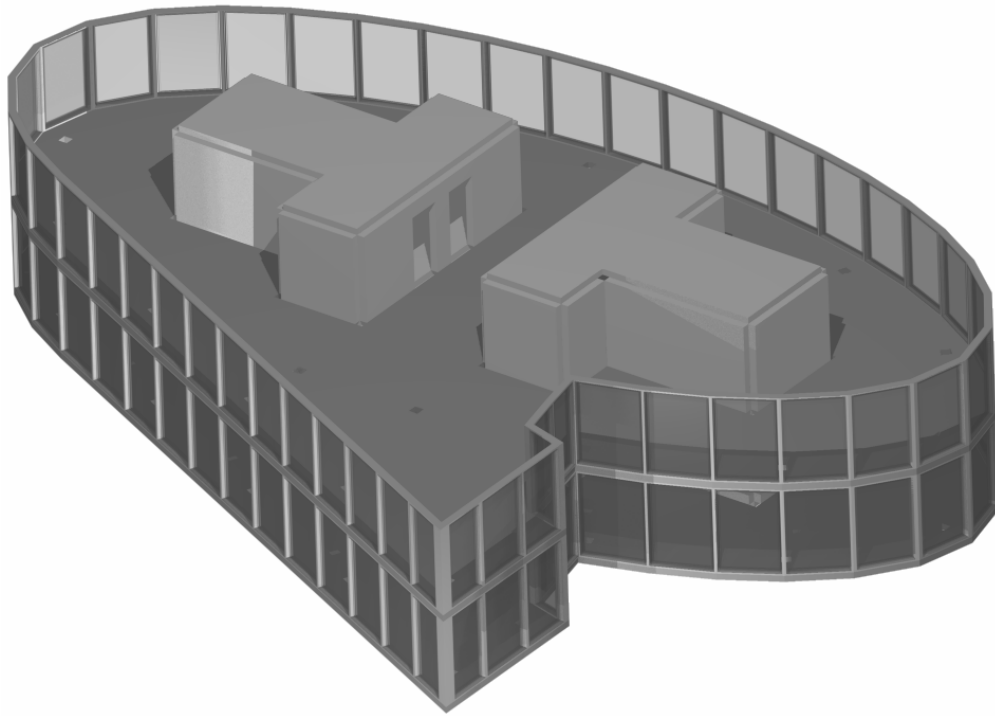


Abbildung 90: Modell mit Differenzobjekten.

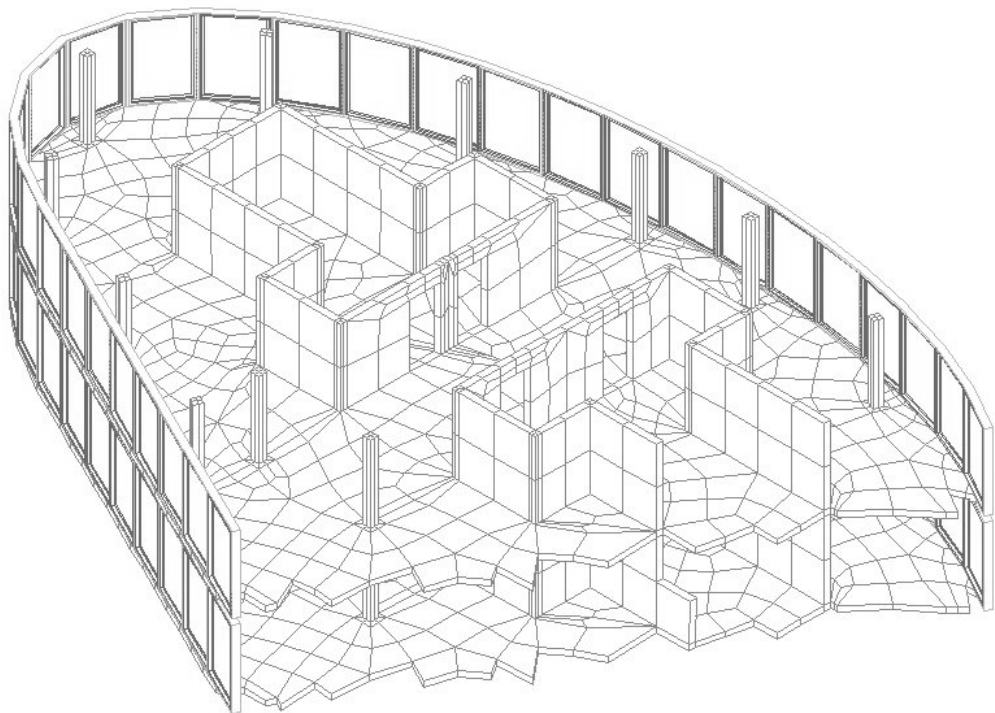


Abbildung 91: Mit Hexaedern vernetzte Teilstruktur³⁶.

³⁶ Fensterelemente sind nur aus Darstellungsgründen berücksichtigt und im statischen Modell nicht abgebildet.

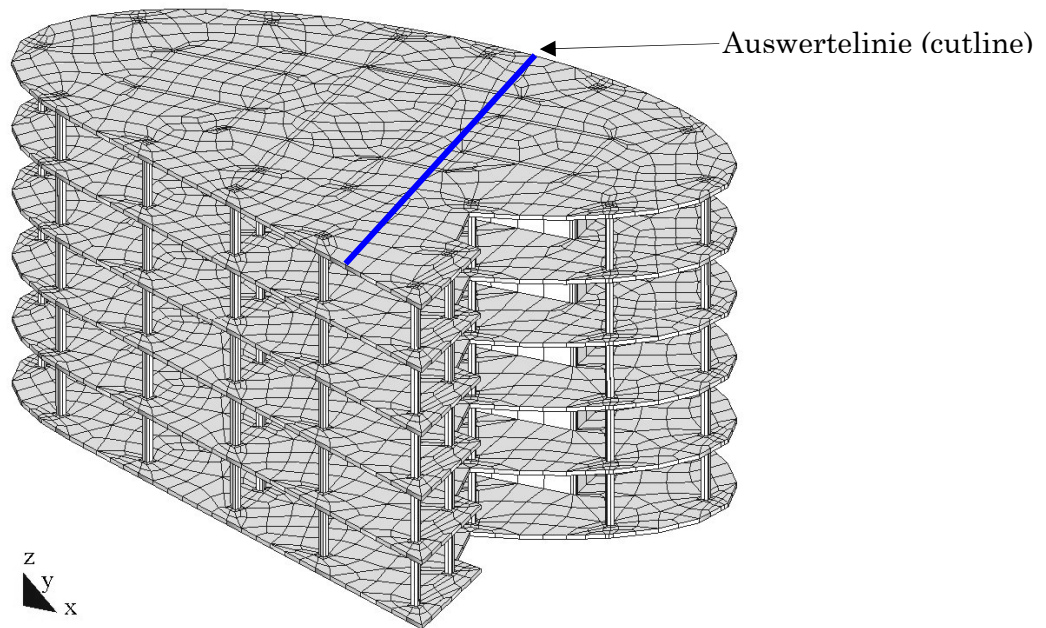


Abbildung 92: Mit 8313 Hexaedern vollständig vernetztes Gesamtmodell.

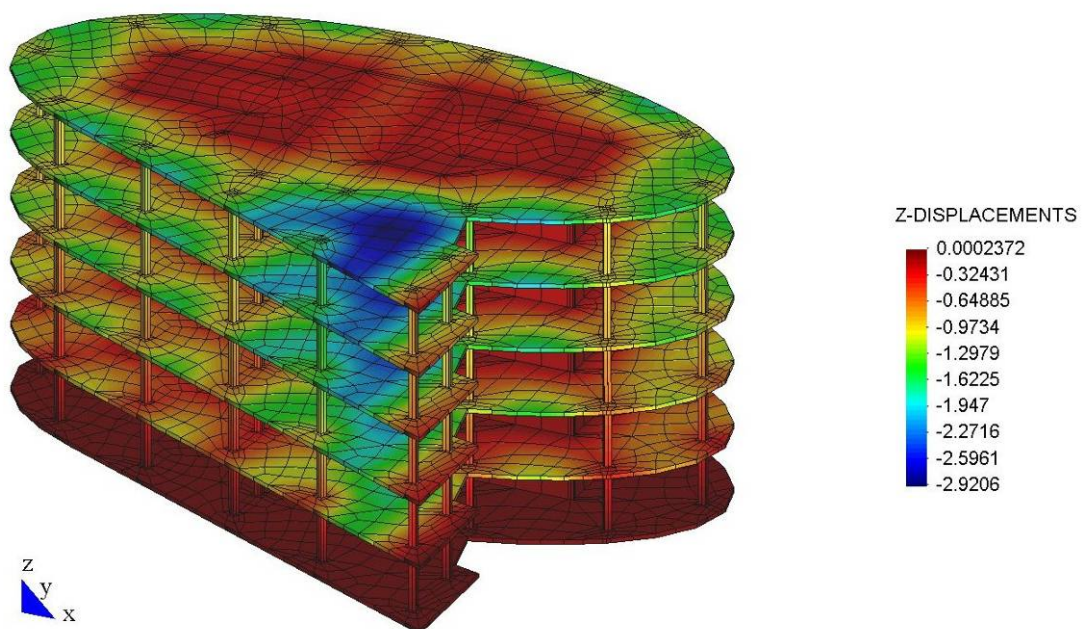


Abbildung 93: Vertikale Verschiebungen unter Eigengewicht.

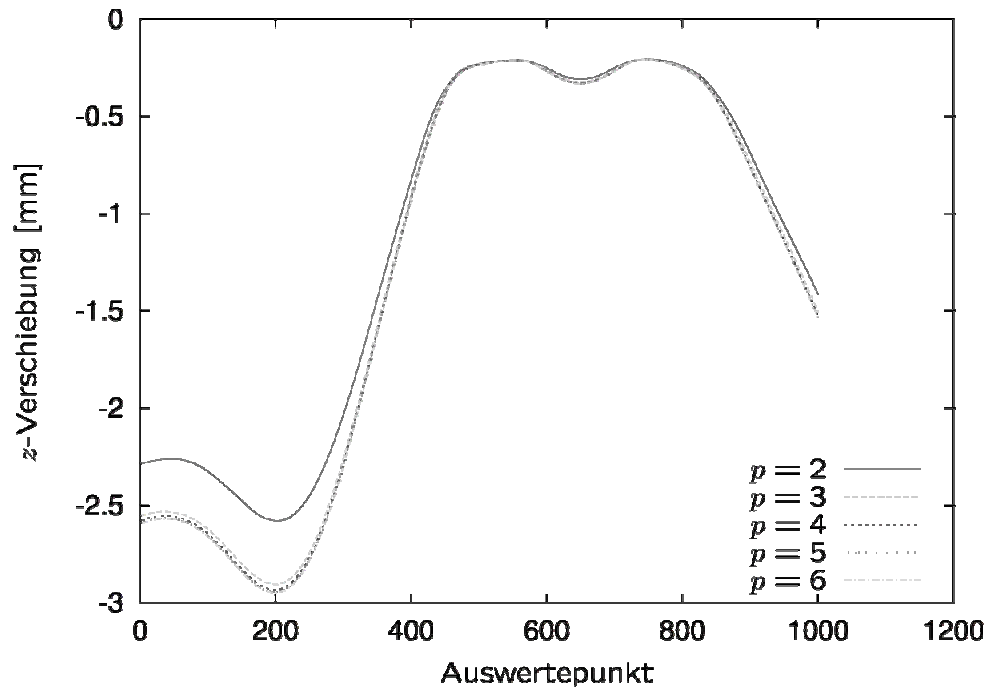


Abbildung 94: Darstellung der absoluten Verschiebungen an der Auswertelinie berechnet mit den Polynomgraden zwei bis sechs.

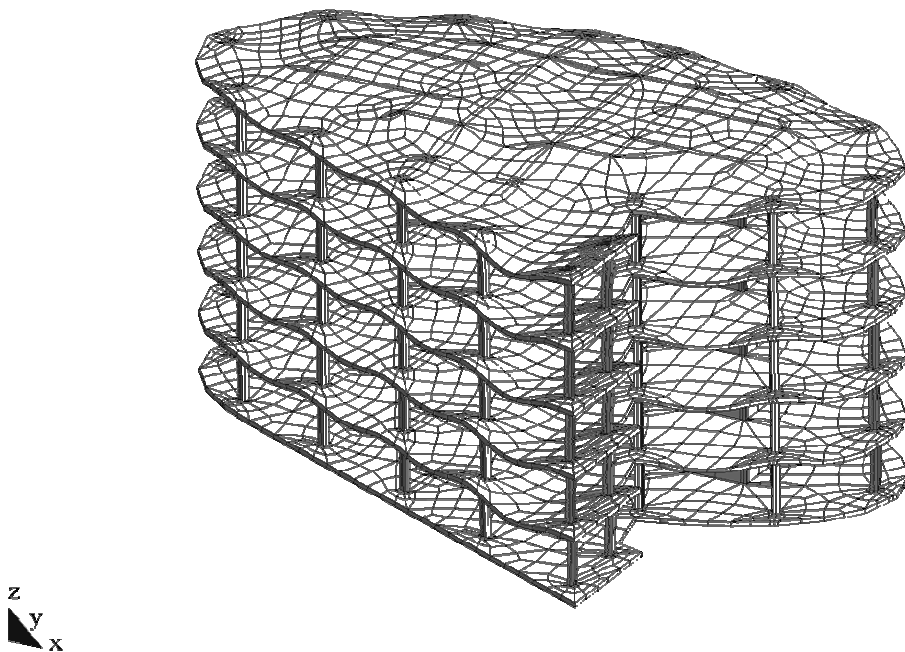


Abbildung 95: Plot der vertikalen Verschiebungen.

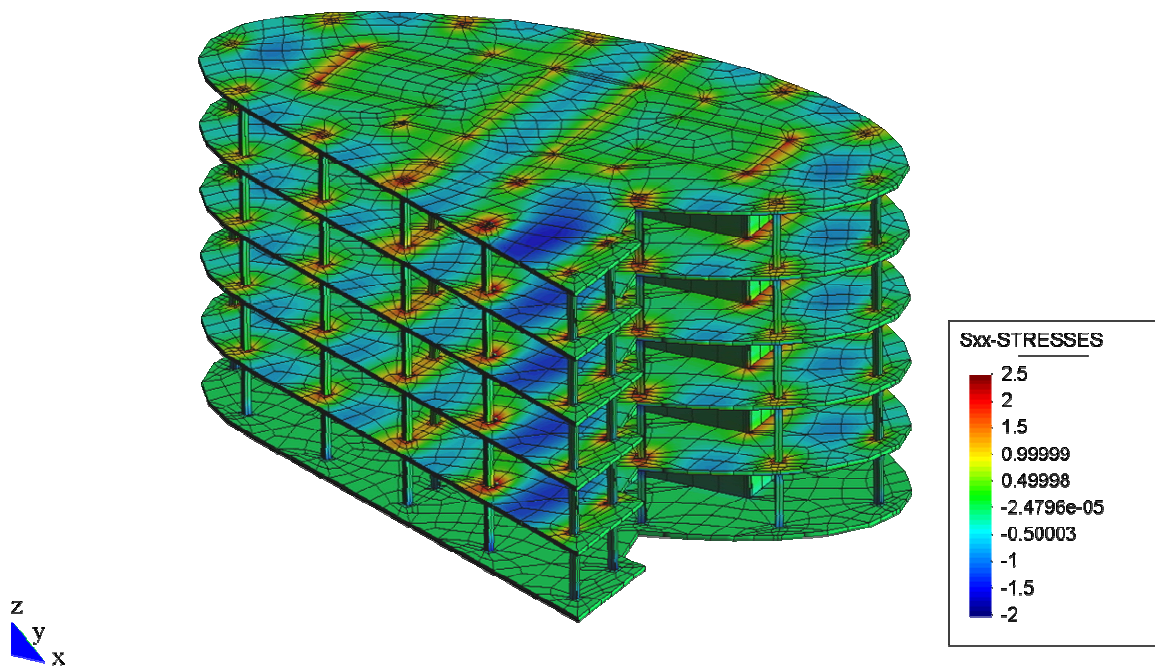


Abbildung 96: Darstellung der σ_{xx} - Spannungen.

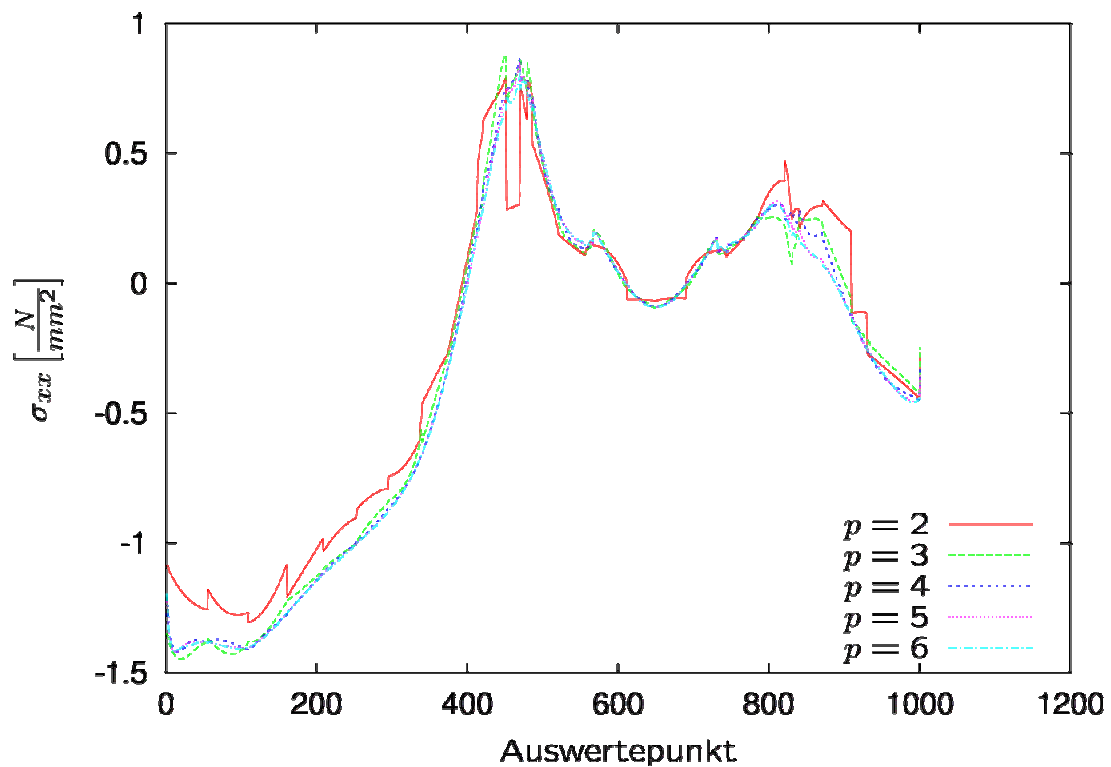


Abbildung 97: Darstellung der σ_{xx} - Spannungen an der Auswertelinie bei verschiedenen Polynomgraden.

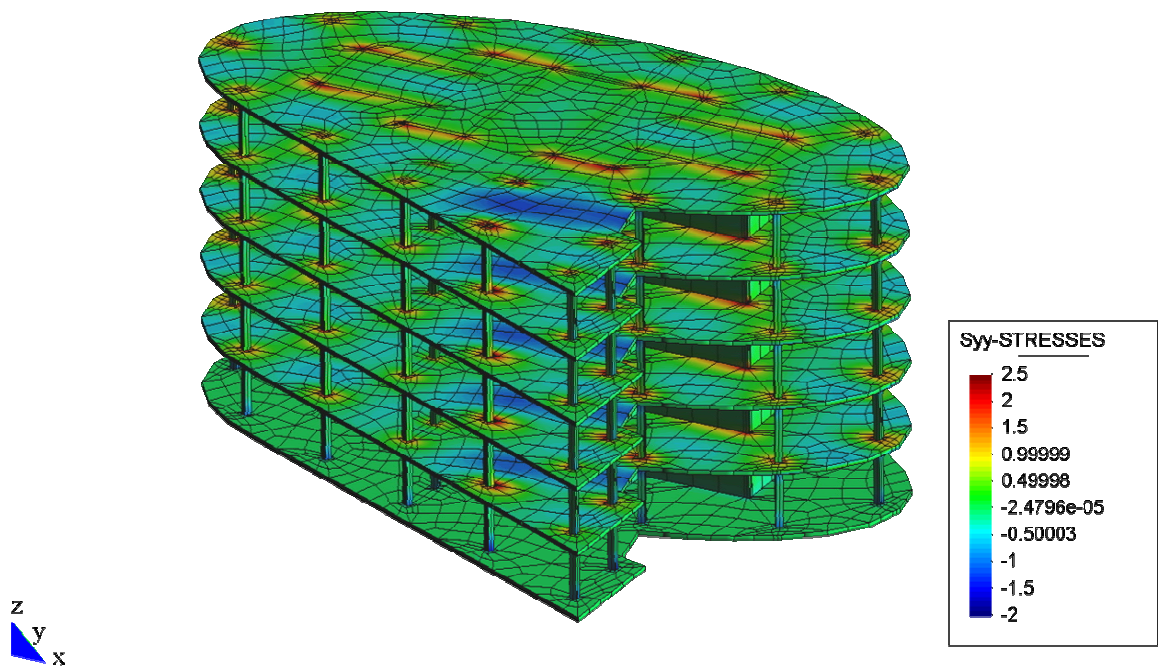


Abbildung 98: Darstellung der σ_{yy} - Spannungen

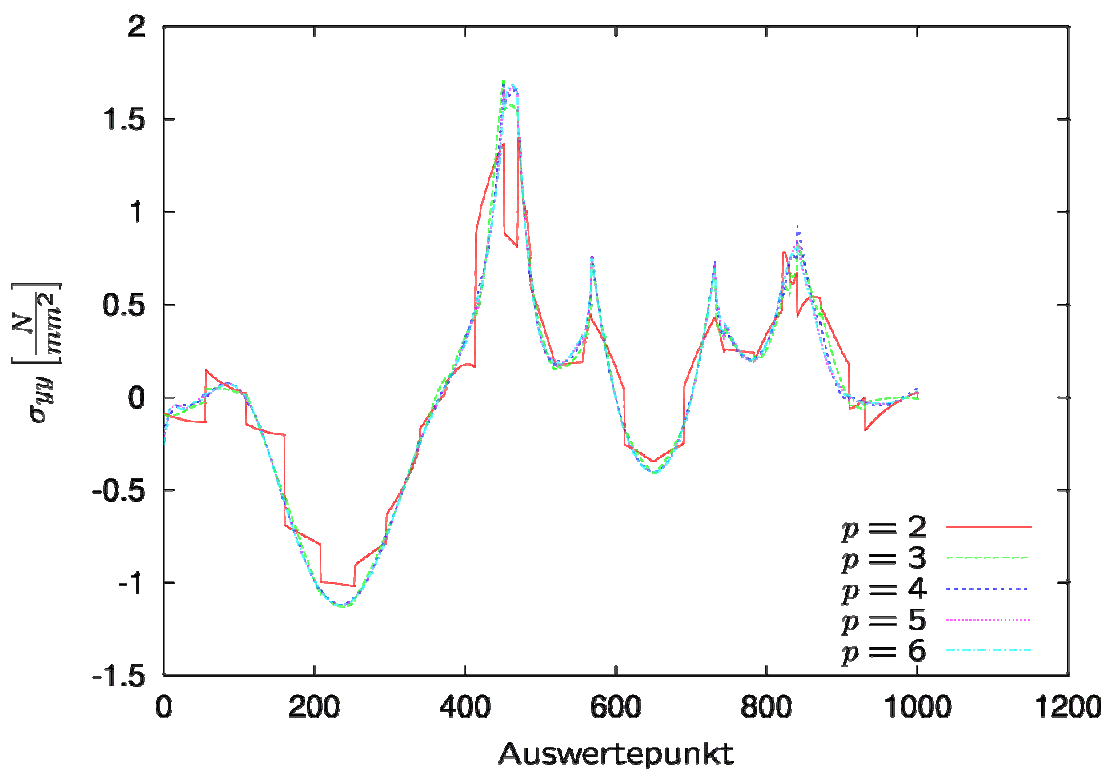


Abbildung 99: Darstellung der σ_{yy} - Spannungen an der Auswertelinie bei verschiedenen Polynomgraden.

Die oben dargestellten Modelle zeigen beginnend mit dem originalen IFC-Modell wichtige Stationen bei der Ableitung des numerischen Modells. In Abbildung 92 wird die diskretisierte Struktur mit einer Auswertelinie dargestellt. Die in den Abbildung 94, Abbildung 97 und Abbildung 99 dargestellten Auswertung über 1000 Auswertepunkte beziehen sich auf diese Auswertelinie. Bei der Darstellung der Analyseergebnisse wurde aufgrund des großen Umfangs eine Auswahl getroffen. Es sind nun lediglich die Abbildungen der absoluten, vertikalen Verschiebungen und die Membranspannungen gezeigt. Es wird jedoch ersichtlich, dass für dieses Modell die Lösung mit einem Ansatz von $p = 5$ bereits konvergiert ist.

Die nachfolgende Tabelle zeigt die Anzahl der Freiheitsgrade des Systems (bei $q = 2$), die im Rahmen der Konvergenzstudie zustande kamen.

Polynomgrad p	Freiheitsgrade
2	156078
3	253158
4	409536
5	625212
6	905970

Tabelle 2: Anzahl der Freiheitsgrade für das oben gezeigte Beispiel für unterschiedliche Polynomgrade.

5 Zusammenfassung und Ausblick

5.1 Zusammenfassung

In der vorliegenden Arbeit wird ein Verfahren vorgestellt, welches es erlaubt, die Strukturanalyse auf einem Bauwerksmodell aufzusetzen und damit Vereinfachungen bei der Durchführung der Tragwerksplanung zu erreichen. Die in einem Bauwerksmodell enthaltenen topologischen und semantischen Informationen machen Methoden möglich, für die die Kenntnis der Gebäudestruktur wesentliche Voraussetzung ist. Nachbarschaftsinformationen von Bauteilen spielen eine zentrale Rolle bei der Zerlegung des geometrischen Modells des Bauwerks, das zum Ziel hat, ein Verbindungsmodell zu identifizieren, bei dem spezielle Verbindungsobjekte an Orten von Bauteilverbindungen generiert werden. Das Verbindungsmodell trägt wesentlich dazu bei, dass Teile der vorgestellten Modellkette universell einsetzbar sind. Dies zeigen Arbeiten bei einer am Lehrstuhl für Bauinformatik durchgeführten thermischen Gebäudesimulation, bei der im Vorfeld ‚Räume‘ zu identifizieren sind. Das Verfahren zur Raumerkennung nutzt das Verbindungsmodell als Basismodell. Details dazu werden in [Treeck 03] beschrieben. Hinsichtlich der Strukturanalyse ist das Verbindungsmodell wesentliche Voraussetzung für die Diskretisierung des geometrischen Modells.

Der Zerlegung des Modells schließt sich die automatische Vernetzung mit Hexaederelementen an. Dabei wurde gezeigt, dass die Vernetzung einer 3D-Struktur grundsätzlich in eine 2D bzw. 1D-Vernetzungsproblematik überführt werden kann. Die automatische Vernetzung wird dabei mittels einer Kombination aus 2D-Freivernetzer und 3D-Vernetzungsmakros durchgeführt. Einschränkungen gibt es bezüglich geometrischer Formen, die mit der vorgestellten Methode vernetzt werden können. Bisher wurde die Vorgehensweise erfolgreich auf Bauwerke angewandt, die im wesentlichen aus ebenen Standardstrukturen wie Platten-, Scheiben- und Stützen-Objekten aufgebaut sind.

Hinsichtlich der Berechnung der Tragstruktur wurde eine Variante der FEM (p-Version) eingesetzt, die ein einheitliches 3D-Hexaederelement verwendet und auch für dünnwandige Strukturen geeignet ist. Die dabei verwendeten hohen Ansätze erlauben lang gestreckte Elemente. Dabei kann ein anisotroper Ansatz weitere Vorteile dieser Methode durch eine örtlich angepasste Wahl des polynomialen Ansatzes und letztlich durch eine Reduktion der Anzahl der Freiheitsgrade ausnützen. Die in der Tragwerksplanung übliche Reduktion der Dimension bei der Bildung physikalischer Modelle (Identifizierung von Teilstrukturen wie Platte, Scheibe, Stütze, Rahmen, etc.) ist beim Einsatz reiner 3D-Elemente nicht mehr nötig.

Die eingesetzten und im Rahmen dieser Arbeit entwickelten Methoden in den Bereichen Basismodellerzeugung, Zerlegung, Diskretisierung und Berechnung wurden zu einer aufeinander abgestimmten Prozesskette zusammengefügt, die den Tragwerksplaner wesentlich dabei unterstützt, auch Berechnungen an komplexen Strukturen mit vertretbarem Aufwand durchzuführen. Die Integration verschiedener Fachplaner (hier Architekt und Tragwerksplaner) wird im zyklisch ablaufenden Planungsprozess durch den Bauwerksmodell-basierten Ansatz gefördert. Am Beispiel des Uniqua-Towers wird die Leistungsfähigkeit und der Nutzen dieses Ansatzes deutlich: Bereits heute können komplexe Modelle in der Tragwerksplanung direkt vom Bauwerksmodell abgeleitet und relativ einfach mit ausreichender Genauigkeit vollständig dreidimensional analysiert werden.

Abschließend soll aber noch betont werden, dass der vorgestellte Ansatz zwar ein gewisser Schritt in Richtung ‚*Statik auf Knopfdruck*‘ ist, die Anforderungen an den Tragwerksplaner dennoch nicht geringer werden sondern weiterhin die genaue Kenntnis der inneren Zusammenhänge der Tragstruktur, sowie die Fähigkeit zur überschlägigen Kontrolle der Ergebnisse erforderlich sind.

5.2 Ausblick

Betrachtet man die baupraktische Nutzbarkeit, so lässt sich feststellen, dass die Verfahren zur Zerlegung und Diskretisierung noch weiter verbessert werden müssen, um ein größeres, realitätsnaheres Spektrum an geometrischen Modellen bearbeiten zu können. Weiterhin wäre es sinnvoll, die momentan bestehende Beschränkung auf eben berandete Bauteile gleich bleibender Dicke aufzuheben. Die numerische Realisierung dazu wurde bereits in Arbeiten von BRÖKER [Bröker 01] vorgestellt.

Auf numerischer Seite gibt es weiteres Verbesserungspotential hinsichtlich der Anwendung eines anisotropen Verschiebungsansatzes. Details sind in [Düster 01, Düster 02, Muthler 04, Rank 04a] nachzulesen. Zentrales Thema darin sind Berechnungen von platten- und schalenartige Bauteilen, wobei die Polynomgrade des Verschiebungsansatzes in Dickenrichtung der Platte bzw. Schale gegenüber denjenigen in tangentialer Richtung verringert werden. Beim Beispiel aus Kapitel 4 wurde für die Berechnung der Struktur aus Gründen der Berechenbarkeit (bei 8 Gigabyte Arbeitsspeicher) ein erster Schritt in diese Richtung unternommen. Die Anzahl der Freiheitsgrad wurde dadurch reduziert, dass bei platten- und scheibenartigen Strukturen konstant mit einem Polynomgrad $q = 2$ in Dickenrichtung gearbeitet wurde. Besonders interessant zur Reduktion des rechnerischen Aufwandes ist jedoch ein adaptiver Ansatz. Dabei wird, basierend auf einem Fehlerindikator, eine automatische Ermittlung unterschiedlicher Polynomgrade in den verschiedenen Richtungen vorgenommen. Arbeiten dazu sind in [Rank 04b] beschrieben.

Im Hinblick auf die in der Einleitung genannte Einbettung dieser Arbeit in das Schwerpunktprogramm zur Verbesserung der vernetzt-kooperativen Planungsprozesse sind über den bisher erreichten Stand der entwickelten Werkzeuge Erweiterungen sinnvoll, die die nachfolgend genannten Punkte behandeln:

- Zuführung der Ergebnisse aus der volumenorientierten Tragwerksplanung einer normativen Bemessung.
- Hinzunahme des zeitlichen Faktors zur volumenorientierten Tragwerksplanung zur Berücksichtigung von Bauphasen im Herstellungsprozess.

- Automatische selektive Neuberechnung von Teilmodellen bei geändertem Entwurfsmodell.

Die hier vorgestellte Gebäudemodell-basierte Strukturanalyse endet mit der Berechnung der dreidimensionalen Spannungen am Tragwerk. Es sind deshalb Überlegungen anzustellen, inwieweit diese Ergebnisse direkt oder über integrale Größen für eine Bemessung geeignet sind. Dies könnte einerseits dazu führen, dass eine neue Bemessungsmethodik notwendig wird (evtl. ebenfalls dreidimensional) oder dass die dreidimensionalen Spannungen für existierende Bemessungsverfahren aufbereitet werden müssen.

Im Bereich der Produktmodellierung ist die Integration verschiedener Teilprozesse und Teilmodelle nach wie vor Gegenstand aktueller Forschungsarbeiten. Aus der Sicht der vorliegenden Arbeit wäre in diesem Zusammenhang die Erweiterung der volumenorientierten Tragwerksplanung um prozessorientierte Aspekte sinnvoll, d.h. die Verknüpfung von zeitabhängigen Planungsmethoden mit der räumlichen Sichtweise eines geometrischen Modells. Das würde die Simulation und Visualisierung des Bauablaufes ermöglichen, um beispielsweise frühzeitig eventuelle Konflikte erkennen und auflösen zu können.

Letztendlich ist die lokale Neuvernetzung von Teilstrukturen im Hinblick auf kleine Änderungen am zugrunde liegenden Bauwerksmodell interessant (z.B. Verschieben einer Stütze). Dazu ist es notwendig, die Methoden zur Zerlegung und Vernetzung dahingehend zu überarbeiten, dass Teilstrukturen nach bisherigem Muster verarbeitet werden können, aber gleichzeitig unverändert gebliebene Nachbarstrukturen mit berücksichtigt werden. Bei der Vernetzung wäre darüber hinaus noch zu prüfen, ob die geänderte Diskretisierung der neu vernetzten Struktur Auswirkungen auf benachbarte Strukturen hat und damit eine Glättung der Elementkantenlängen über einen größeren Bereich der Gebäudestruktur notwendig wird.

Literaturverzeichnis

- [Abramovici 02] M. Abramovici, O. Sieg: Status and Development Trends of Product Lifecycle Management Systems. In: Proceedings of IPPD '02, Wroclaw, Polen, 2002.
- [Abramowski 91] S. Abramowski, H. Müller: Geometrisches Modellieren. BI-Wissenschaftsverlag, Mannheim, 1991.
- [ACIS 02] Spatial Corporation: ACIS – 3D Geometrie-Modellierer. Westminster, Colorado, USA, Internet: www.spatial.com.
- [Anderl 00] R. Anderl, D. Trippner: STEP – Standard for the Exchange of Product Model Data. B. G. Teubner Verlag, Stuttgart, 2000.
- [Balzert 99] H. Balzert: Lehrbuch der Objektmodellierung – Analyse und Entwurf. Spektrum Akademischer Verlag, Heidelberg, 1999.
- [Balzert 01] H. Balzert: Lehrbuch der Software-Technik. Spektrum Akademischer Verlag, Heidelberg, 2001.
- [Bank 90] R. E. Bank: PLTMG – A software package for solving elliptic partial differential equations. Society for Industrial and Applied Mathematics (SIAM), 1990.
- [Bamtec 04] Internet: www.bamtec.de.
- [Baumgart 72] B. G. Baumgart: Winged-Edge Polyhedron Representation. Technical Report STAN-CS-72-320, Computer Science Department, Stanford University, Palo Alto, USA, 1972.
- [Baumgarten 90] B. Baumgarten: Petri-Netze – Grundlagen und Anwendungen. BI-Wissenschaftsverlag, Mannheim, 1990.
- [Booch 99] G. Booch, J. Rumbaugh, I. Jacobson: Das UML-Benutzerhandbuch. Addison-Wesley, München, 1999.
- [Borkner 96] O. Borkner-Delcarlo: Eine Halbfacetten-Datenstruktur für die Modellierung von ‚non-manifold‘-Körper. Dissertation, Fachgebiet Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 1996.
- [Bretschneider 98] D. Bretschneider: Modellierung rechnerunterstützter, kooperativer Arbeit in der Tragwerksplanung. Dissertation, Verein Deutscher Ingenieure, Fortschrittsberichte VDI, Düsseldorf, 1998.
- [Bröker 01] H. Bröker: Integration von geometrischer Modellierung und Berechnung nach der p-Version der FEM. Dissertation, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2001.
- [Bungartz 96] H. J. Bungartz, M. Griebel, C. Zenger: Einführung in die Computergraphik. Vieweg & Sohn Verlag, Braunschweig, 1996.

- [Buschmann 98] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal: Pattern-orientierte Software-Architektur. Addison-Wesley, München, 1998.
- [Cockburn 02] A. Cockburn: Agile Software Development. Addison-Wesley (Pearson Education), Indianapolis, USA, 2002.
- [Corney 97] J. Corney: 3D-Modeling with the ACIS Kernel and Toolkit. John Wiley & Sons, 1997.
- [Crouse 02] B. Crouse: Lattice-Boltzmann Strömungssimulationen auf Baumdatenstrukturen. Dissertation, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2003.
- [DFG] DFG-Schwerpunktprogramm 1103: Vernetzt-kooperative Planungsprozesse im konstruktiven Ingenieurbau. Internet: <http://www.dfg-spp1103.de/>.
- [Düster 01] A. Düster, H. Bröker, 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.
- [Düster 02] A. Düster: High order finite elements for three-dimensional, thin-walled nonlinear continue. PhD thesis, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2002.
- [Düster 03] A. Düster, S. Hartmann, E. Rank: p-FEM applied to finite isotropic hyper elastic bodies. Computer Methods in Applied Mechanics and Engineering, 192, 5147-5166, 2003.
- [Frank 00] A. Frank: Organisationsprinzipien zur Integration von geometrischer Modellierung, numerischer Simulation und Visualisierung. Dissertation, Technische Universität München, 2000.
- [Gamma 98] E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns – Elements of Reusable Object-Oriented Software. Addison-Wesley, USA, 2000.
- [Geiger 01] A. Geiger: Produktmodelle im Bauwesen – IFC im Praxistest. Semesterarbeit, FH Karlsruhe, Fachbereich Architektur und Bauingenieurwesen, Karlsruhe, 2001.
- [Goldfarb 00] C. F. Goldfarb, P. Prescod: Das XML-Handbuch - Anwendungen, Produkte, Technologien. Addison-Wesley, München, 2000.
- [Haas 93] W. Haas: CAD-Datenaustausch-Knigge für Architekten und Bauingenieure. Springer Verlag, 1993.
- [Hauschild 02] T. Hauschild: Computer Supported Cooperative Work – Applikationen in der Bauwerksplanung auf Basis einer integrierten Bauwerksmodellverwaltung. Dissertation, Fakultät Bauingenieurwesen, Bauhaus-Universität Weimar, 2002.
- [Herold 01] H. Herold: Das QT-Buch – Portable GUI-Programmierung unter Linux/UNIX/Windows. SuSE Press, Erlangen, 2001.

- [Hofstadter 91] D. R. Hofstadter: Gödel, Escher, Bach – ein Endloses Geflochtenes Band. dtv/Klett-Cotta, München, 1991.
- [HOOPS 02] HOOPS 3D Product Suite: Reference Manual. Tech Soft America, Oakland, USA, 2002, Internet: www.hoops3d.com.
- [IAI 04] Industrieallianz für Interoperabilität: Ein neuer Standard aus der CAD-Welt. Internet: www.iai-ev.de bzw. <http://www.iai-international.org>.
- [IAI ST-4] Industrieallianz für Interoperabilität: IFC Erweiterung ST-4 (Tragwerksplanung), Internet: www.iai-ev.de/projekte/documents/pdf/IFC_ST4.pdf.
- [IFC2x2] International Alliance for Interoperability (IAI): Online-Dokumentation der Industry Foundation Classes (IFC) – Release 2x Edition 2, 2003, Internet: <http://www.iai-ev.de/spezifikation/Ifc2x2/index.htm>.
- [IFC Toolbox] Eurostep Group: The IFC STEP Toolbox – Version 2.x, Eurostep, Stockholm, Internet: www.eurostep.com.
- [ISG Steinmann] R. Steinmann: Implementation Support Group (ISG). Internet: http://www.bauwesen.fh-muenchen.de/iai/iai_isg/.
- [ISO 04] International Organisation for Standardization (ISO), Internet: www.iso.org.
- [Kühner 01] S. Kühner, R. Romberg: Manipulation von IFC Produktmodell-daten mit der 'Eurostep IFC Toolbox'. In: R. Romberg und M. Schulz (Hrsg.), Forum Bauinformatik, Fortschritts-Berichte VDI, Reihe 4: Bauingenieurwesen, VDI-Verlag, 2001.
- [Kühner 03] S. Kühner: Virtual Reality basierte Analyse und interaktive Steuerung von Strömungssimulationen im Bauingenieurwesen. Dissertation, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2003.
- [Lämmle 03] A. Lämmle: Konvertieren eines IFC-Produktmodells in ein ACIS Geometriemodell. Diplomarbeit, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2003.
- [Mäntylä 88] M. Mäntylä: An Introduction to Solid Modelling. Computer Science Press, 1988.
- [Mundani 03] R. P. Mundani, H. J. Bungartz, E. Rank, R. Romberg, A. Niggel: Efficient Algorithms for Octree-Based Geometric Modeling. In: Proceedings of the International Conference on Civil and Structural Engineering Computing, B.H.V. Topping (editor), Civil-Comp Press, 2003.
- [Muthler 04] A. Muthler, D. Düster, W. Volk, E. Rank: High order finite elements applied to the computation of elastic spring back in sheet metal forming. In: Proceedings of NUMIFORM '04, International Conference on Numerical Methods in Industrial Forming Process, Columbus, Ohio, USA, 2004.

- [Oestereich 01] B. Oestereich: Objektorientierte Softwareentwicklung: Analyse und Design mit der UML. Oldenbourg, München, 2001.
- [OpenCascade 04] OpenCascade - 3D Geometrie-Modellierer, Internet: <http://www.opencascade.org>.
- [QT 04] QT Online-Referenz, Internet: www.trolltech.com.
- [Pahl 00] J. Pahl, R. Damrath: Mathematische Grundlagen der Ingenieurinformatik. Springer Verlag, 2000.
- [PRO 04] PROSTEP AG. Darmstadt, Internet: <http://www.prostep.de>.
- [Rank 94] E. Rank, M. Rücker, M. Schweingruber: Automatische Generierung von Finite-Element-Netzen. Bauingenieur, 69:373-379, 1994.
- [Rank 04a] E. Rank, A. Düster, A. Muthler, R. Romberg: High order solid elements for thin-walled structures with applications to linear and non-linear structural analysis. In: Proceedings of ECCOMAS '04, P. Neittaanmäki (editor), European Congress on Computational Methods in Applied Sciences and Engineering, Jyväskylä, Finland, 2004.
- [Rank 04b] E. Rank, A. Düster, Z. Alam: Adaptive computation of thin-walled structures based on high order solid finite elements. In: Proceeding of WCCM VI in conjunction with APCOM '04, Beijing, China, 2004.
- [Rücker 01] M. Rücker: A parallel p-version finite element approach for civil engineering and structural mechanics problems. PhD thesis, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2001.
- [Rüppel 96] U. Rüppel, U. Meißner: Integrierte Planung, Fertigung und Nutzung von Bauwerken auf der Basis von Produktmodellen. Bauingenieur, 71:47-55, Springer-Verlag, 1996.
- [Schneider 01] U. Schneider, D. Werner: Taschenbuch der Informatik. Fachbuchverlag Leipzig im Carl Hanser Verlag, Wien, 2001.
- [Schwarz 91] H. R. Schwarz: Methode der finiten Elemente. Teubner Studienbücher, Stuttgart, 1991.
- [Schweingruber 99] M. Schweingruber-Straten: Generieren von Oberflächennetzen nach der Gebietsteilungstechnik. Dissertation, Fakultät für Bauingenieurwesen, Technische Universität Dortmund, 1999.
- [Sedgewick 02] R. Sedgewick: Algorithmen in C++. Addison-Wesley, München, 2002.
- [Seitz 02] A. G. Seitz: Entwurf und Implementierung einer geometrischen Partitionierung von Gebäudemodellen unter Verwendung des Geometriemodellierkerns ACIS. Diplomarbeit, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2002.
- [Sperber 03] K. H. Sperber: ArchiCAD – Einführung, Workshop, Referenz. Addison-Wesley, München, 2003.

- [Stroustrup 00] B. Stroustrup: Die C++ Programmiersprache. Addison-Wesley, München, 2000.
- [Szabó 91] B. A. Szabó, I. Babuška: Finite element analysis. John Wiley & Sons, USA, 1991.
- [Szabó 04] B. A. Szabó, A. Düster, E. Rank: The p-version of the finite element method. In: E. Stein, R. de Borst and T.J.R. Hughes (editors), Encyclopedia of Computational Mechanics, John Wiley & Sons, USA, 2004.
- [Treeck 01] C. van Treeck, M. Krafczyk, M. Schulz: Lattice-Boltzmann Verfahren im Bauwesen – Tutorial eines grafisch-interaktiven Strömungssimulators. Forum Bauinformatik '01, Fortschritts-Berichte VDI-Verlag, Düsseldorf, 2001.
- [Treeck 04a] C. van Treeck, R. Romberg, E. Rank: Graphentheoretische Analyse von Produktmodelldaten hinsichtlich Raumtopologie und Gebäudestruktur. Festschrift 60. Geb. Prof. Hartmann, VDI-Tagungsband, 2004.
- [Treeck 04b] C. van Treeck: Gebäudemodell-basierte Simulation von Raumluftrömungen. Dissertation, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2004.
- [Turau 96] V. Turau: Algorithmische Graphentheorie. Addison-Wesley, Bonn, 1996.
- [Weiler 88] K. Weiler: The radial edge structure – A topological representation for non-manifold geometric boundary modeling. In: M. J. Wozhny, H. W. McLaughlin und J. L. Encarnacao (editor), Geometric Modeling for CAD Applications, Amsterdam, Holland, 1988.
- [Weise 02] M. Weise, P. Katranuschkov, T. Liebich, R. J. Scherer: Structural analysis extension of the IFC modelling framework. Väino Tarandi (editor), In: Electronic Journal of Information Technology (ITcon) Vol. 8, Special Issue IFC - Product models for the AEC area, page 181-200, 2003, Internet: <http://www.itcon.org/2003/14>.
- [WWB 04] WeltWeitBau, Ingenieurgesellschaft für angewandte Bauinformatik, Berlin, Internet: www.wwbau.de.

Vorveröffentlichte Teilergebnisse

Die folgende Auflistung enthält die eigenen Publikationen, in denen bereits Teile dieser Arbeit vorveröffentlicht wurden. Die Erlaubnis zur Vorveröffentlichung wurde von der Fakultät für Bauingenieur- und Vermessungswesen der Technischen Universität München erteilt.

- [1] E. Rank, A. Düster, A. Muthler, R. Romberg: High order solid elements for thin-walled structures with applications to linear and non-linear structural analysis. In: Proceedings of ECCOMAS '04, Computational Modeling, Numerical Methods and Simulation in Engineering and Applied Sciences, Jyväskylä, Finland, 2004.
- [2] R. Romberg, A. Niggel, C. van Treeck, E. Rank: Structural Analysis based on the Product Model Standard IFC. In: Proceedings of ICCCB'E '04, Xth International Conference on Computing in Civil and Building Engineering, Weimar, Germany, 2004.
- [3] C. van Treeck, R. Romberg, E. Rank: Graphentheoretische Analyse von Produktmodelldaten hinsichtlich Raumtopologie und Gebäudestruktur. In: Festschrift 60. Geburtstag Prof. Hartmann, VDI-Tagungsband, 2004.
- [4] A. Niggel, R. Romberg, E. Rank, R. Mundani, H. Bungartz: A framework for concurrent structure analysis in building industry. In: Proceedings of ECPPM '04, European Conference on Product and Process Modelling in the AEC Industry, Istanbul, Turkey, 2004.
- [5] R. Mundani, H. J. Bungartz, E. Rank, R. Romberg, A. Niggel: Efficient Algorithms for Octree-Based Geometric Modelling. In: Proceedings of the International Conference on Civil and Structural Engineering Computing, B.H.B. Topping (editor), Civil-Comp Press, 2003.
- [6] C. van Treeck, R. Romberg, E. Rank: Simulation based on the product model standard IFC. In: Proceedings of 8th International IBPSA Conference Building Simulation, Eindhoven, Netherlands, 2003.
- [7] C. van Treeck, R. Romberg, E. Rank, T. Schrag: A simulation framework for the building energy simulation based on the IFC data exchange standard. In: Proceedings of ICCE '03, International Congress on Civil Engineering, Isfahan, Iran, 2003.
- [10] R. Romberg, C. van Treeck, A. Niggel, E. Rank: Numerische Simulation auf Basis des IFC-Bauwerksmodells. In: Proceedings of IKM '03, Internationales Kolloquium über Anwendung der Informatik und der Mathematik in Architektur und Bauwesen, Weimar, Germany, 2003.
- [11] R. Romberg, A. Düster, E. Rank: Solid modeling as a basis of co-operative planning in structural engineering. In: Proceedings of ECPPM '02, European Conference of Product and Process Modeling in the Building and Construction Industry, Portoroz, Slowenien, 2002.