



TECHNISCHE UNIVERSITÄT MÜNCHEN

LEHRSTUHL FÜR MASSIVBAU

INSTITUT FÜR BAUSTOFFE UND KONSTRUKTION

**Entwicklung eines Add Ins basierend auf Siemens
NX zum Datenaustausch in die CADINP Sprache
von SOFiSTiK**

MASTERS'S THESIS IM STUDIENGANG BAUINGENIEURWESEN

ELENA SEESER, B.SC.


MÜNCHEN 2015

MASTER`S THESIS IM FACH MASSIVBAU
EINGEREICHT AN DER TECHNISCHEN UNIVERSITÄT MÜNCHEN
AM LEHRSTUHL FÜR MASSIVBAU
INSTITUT FÜR BAUSTOFFE UND KONSTRUKTION

IN KOOPERATION MIT DER PLANUNGSGESELLSCHAFT
OBERMEYER PLANEN + BERATEN GMBH

THEMA:

**Entwicklung eines Add Ins basierend auf Siemens
NX zum Datenaustausch in die CADINP Sprache
von SOFiSTiK**

Verfasser:	Elena Seeser, B.Sc.
Matrikelnummer:	
Referent:	Univ.-Prof. Dr.-Ing. Dipl.-Wirt.-Ing. Oliver Fischer M.Sc. Josef Landler M.Sc. Dominic Singer
Betreuer Obermeyer:	Dipl.-Ing. Markus Hochmuth Dipl.-Ing. Stephan Nemeth
Begonnen am:	27.04.2015
Eingereicht am:	30.09.2015
Abschließend beurteilt am:	Note:

Erklärung:

Hiermit erkläre ich, dass die vorliegende Master`s Thesis von mir selbst angefertigt wurde, und nur die aufgeführten Quellen und Hilfsmittel Verwendung fanden. Ich erkläre mich damit einverstanden, dass meine Master`s Thesis auf unbefristete Zeit zu Hochschulzwecken aufbewahrt werden darf.

München, den 30.09.2015

Unterschrift

Kurzfassung

Die zentrale Datenhaltung ist ein Hauptmerkmal des Building Information Modeling. Die Frage des Datenaustausches ist jedoch noch nicht ausreichend geklärt. Die vorliegende Arbeit befasst sich mit dem Thema Datenaustausch zwischen Siemens NX und der CADINP Sprache von SOFiSTiK. Anhand eines Anwenderbeispiels werden Ansätze aufgezeigt, wie aus dem digitalen Gebäudemodell in NX strukturelevante Daten über ein neutrales Dateiformat in SOFiSTiK übergeben werden können. Es soll dabei aufgezeigt werden, ob eine exakte Übergabe der Geometrie zielführend für die statische Berechnung ist. Im Zuge dieser Arbeit sollen dazu drei unterschiedliche Methoden verglichen werden. Die ersten beiden Methoden beinhalten die Übergabe der geometrischen Daten aus NX als Volumen- und Flächenmodell. Dazu wurden zwei Direktschnittstellen entwickelt. Als dritte Methode soll das Bauwerk konventionell in SOFiSTiK eingegeben werden. Abschließend werden die drei Methoden bewertet und ein „Best Practise“ Ansatz definiert. Zum einen wird dazu die Geometrie der drei Modelle verglichen. Zum anderen werden exemplarische Lastfälle angesetzt und die resultierenden Schnittgrößen gegenübergestellt.

Abstract

The central data storage is the main characteristic of Building Information Modeling. Regarding the exchange of information however, there are still unresolved problems. This Master thesis deals with the data exchange between Siemens NX and the CADINP language of SOFiSTiK. Demonstrated by a real-world example, solution approaches are pointed out that demonstrate how structural data can be transferred from a digital 3D model in NX to SOFiSTiK by a neutral data format. It should show, if transferring the precise geometry is expedient for the structural analysis. Concerning this, three different methods will be compared. The first two methods involve the transfer of geometrical data from NX using a volume and an area model. Two Java-based interfaces have been developed for this purpose. The third method is the conventional, manual input of the model in SOFiSTiK. Finally the three methods are rated. With this in mind the geometry of the three models was compared. Furthermore typical load cases were set and the resulting internal forces were analyzed. By evaluating the results a “best practice” approach was defined.

Vorwort

Die vorliegende Arbeit ist bei der Firma OBERMEYER PLANEN UND BERATEN GmbH in Kooperation mit dem Lehrstuhl für Massivbau der Technischen Universität entstanden. Viele Personen haben zur Realisierung dieser Arbeit beigetragen, daher möchte ich mich an dieser Stelle bei ihnen bedanken.

Mein erster Dank geht an den Lehrstuhlinhaber, Herrn Prof. Dr.-Ing. Oliver Fischer, sowie an den geschäftsführenden Gesellschafter Herrn Dipl.-Ing. Maximilian Grauvogl und den Leiter der Abteilung Ingenieurbau Herrn Dipl.-Ing. Erhard Lederhofer der Firma OBERMEYER.

Besonders danke ich meinen Betreuern Markus Hochmuth und Stephan Nemeth für die Themenstellung und ihre fachliche Betreuung. Ihre wertvollen Ratschläge und Unterstützung haben maßgebend zum Gelingen dieser Arbeit beigetragen.

Weiterhin danke ich meinen Betreuern seitens der Universität. Ich danke Dominic Singer, vom Lehrstuhl für Computergestützte Modellierung und Simulation für seine fachliche Unterstützung im Bereich Informatik. Ebenfalls danke ich Josef Landler, vom Lehrstuhl für Massivbau. Sein Einsatz über den gesamten Zeitraum, sowie die Koordination aller Beteiligten haben diese Arbeit ermöglicht.

Abschließend bedanke ich mich bei Andrea Schumacher für das Korrekturlesen, sowie bei meiner Familie und meinem Freund für ihren großen Rückhalt und ihre Unterstützung.

Inhaltsverzeichnis

Erklärung:	
Kurzfassung.....	I
Abstract	II
Vorwort.....	III
Inhaltsverzeichnis	IV
1. Einleitung	1
1.1 Einleitung und Problemstellung.....	1
1.2 Ziel der Arbeit	2
1.3 Aufbau der Arbeit.....	3
2. Projekt EÜ Ventschow	5
2.1 Projektvorstellung.....	5
2.2 Ausführung der EÜ Ventschow.....	5
3. Building Information Modeling	6
3.1 BIM Definition.....	6
3.2 Einsatz von BIM	7
3.3 Dateiformate und Standards	8
3.3.1 Proprietäre Dateiformate	9
3.3.2 Offene Dateiformate.....	10
3.3.3 IFC als BIM-Standard	10
3.4 Klassifizierung der BIM-Lösung.....	11
4. Geometrische Modellierung von Volumenmodellen.....	14
4.1 Direkte Darstellungsmethoden	14
4.1.1 Normzellenschema	14

4.1.2	Oktalbäume (Octrees)	15
4.1.3	Constructive Solid Geometry, CSG	16
4.2	Indirekte Darstellungsmethoden	16
4.2.1	Drahtmodell.....	16
4.2.2	Boundary Representation (B-rep).....	17
4.2.3	Facettenmodell	18
4.2.4	Extrusionsmodelle.....	19
4.2.5	Hybridmodelle.....	20
5.	Das digitale Gebäudemodell als Basis der Tragwerksplanung	21
5.1	Überführung der Gebäudemolldaten in die Tragwerksplanung	22
5.2	Implementierungsmöglichkeiten unter Siemens NX	23
6.	Modellierung der EÜ Ventschow in Siemens NX.....	26
6.1	Allgemeine Vorgehensweise bei Brückenbauwerken	26
6.1.1	Modellierstruktur und Baugruppen-Prinzip.....	26
6.1.2	Vorgehensweise bei parametrischer Brückenmodellierung	29
6.2	Programmübergreifende Dateiformate in Siemens NX.....	31
6.3	Maßnahmen für den Modell-Export in FEM-Programme	32
6.3.1	Export als Volumenmodell.....	34
6.3.2	Export als Flächenmodell	35
7.	Konventionelle Modelleingabe in SOFiSTiK.....	38
7.1	Softwareübersicht.....	38
7.2	Modelleingabe im TEDDY-Editor	39
7.2.1	AQUA Materialdefinition	40
7.2.2	SOFIMSHC Strukturelemente	40
7.2.3	SOFILOAD Lastfalldefinition.....	43
7.2.4	ASE Berechnung des Systems.....	46
8.	Schnittstellenprogrammierung	47
8.1	Einlesen der JT-Datei	47
8.2	Schreiben der DAT-Datei in der CADINP Sprache.....	50
8.2.1	Volumenmodell	52

8.2.2	Flächenmodell.....	52
9.	Auswertung der Schnittstellen anhand des Anwenderbeispiels.....	56
9.1	Geometrische Abmessungen.....	57
9.1.1	Fahrbahnplatte.....	57
9.1.2	Flügelwände.....	58
9.1.3	Kammerwände.....	58
9.1.4	Mittelpfeiler.....	59
9.1.5	Fundamente.....	59
9.2	Lasten und Schnittgrößen.....	59
9.3	Bewertung der in NX generierten Modelle.....	61
9.3.1	Volumenmodell.....	61
9.3.2	Flächenmodell.....	61
10.	Zusammenfassung und Ausblick.....	64
10.1	Zusammenfassung.....	64
10.2	Fazit.....	65
10.3	Ausblick.....	66
Anhang A.....		i
Anhang B.....		iv
Anhang C.....		xvii
Abbildungsverzeichnis.....		xxxii
Tabellenverzeichnis.....		xxxiv
Abkürzungsverzeichnis.....		xxxv
Literaturverzeichnis.....		xxxvii

1. Einleitung

1.1 Einleitung und Problemstellung

Building Information Modeling (BIM), als 3D-gestützte Planungsmethode, soll im Bauwesen zu mehr Transparenz, besserer Planung und zu Termin- und Kostensicherheit führen. Was in anderen Branchen schon Stand der Technik ist, etabliert sich nun auch in der Baubranche. Die Projektbeteiligten arbeiten zusammen und generieren dabei ein digitales Gebäudemodell, das sämtliche Planungsinformationen enthält. Im Hochbau findet BIM immer mehr Anwendung, während sich der Ingenieurbau immer noch diesem Trend entzieht. Ingenieurbauwerke werden in der Regel weiterhin konventionell anhand 2D-Planungsmethoden ausgeführt.

Als Mitbegründer der buildingSMART e.V. und des Leonhard Obermeyer Centers der TU München nimmt die Planungsgesellschaft OBERMEYER PLANEN + BERATEN GmbH eine Vorreiterrolle im Bereich moderner IT-Lösungen im Bauwesen ein. Ziel dieser Organisationen ist es, Projektabwicklungen durch Hilfe intelligenter Softwarelösungen effizienter zu gestalten. Als einer der größten unabhängigen Planungsgesellschaften im Bauwesen in Deutschland weist Obermeyer einen großen Erfahrungsschatz in der interdisziplinären Integration von Prozessen auf und nutzt diesen, um die BIM-Methode zu fördern. Ihr Ziel ist es, die BIM-Integration nicht nur im Bereich Hochbau voranzutreiben, sondern auch Lösungen für den konstruktiven Ingenieurbau zu erarbeiten und auszubauen.

Brückenbauwerke, als Teilbereich des Ingenieurbaus, können als linienorientierte Bauwerke betrachtet werden. Ihr Verlauf hängt in erster Linie stark von der vorhandenen Trassierung und dem Kreuzungswinkel der Trassenkurven ab. Ausrichtung und Lage der Bauteile sind weitreichend davon abhängig. Die Einführung einer parametrischen 3D-Modellierung ist demnach sinnvoll. Obermeyer hat hierzu ein Modellierungskonzept entwickelt Brücken anhand einer konzipierten Hierarchiestruktur parametrisch zu erstellen. Am Ende steht ein Prototyp, der variabel über Parameter an Änderungen angepasst werden kann.

BIM-fähige Softwaresysteme sind vor allem für den Bereich des Hochbaus ausgerichtet. Parametrische Modellierungswerkzeuge für Ingenieurbauwerke sind im Vergleich dazu noch

nicht so gut aufgestellt. Um die Parametrik dennoch optimal anwenden zu können greift Obermeyer daher auf Siemens NX, eine CAD-Software aus der Maschinenbaubranche, zurück.

Eine zentrale Herausforderung der 3D-Planung ist die Problematik des Datenaustauschs. Zur Realisierung eines Bauvorhabens haben sich diverse Softwaresysteme entwickelt, die auf die verschiedenen Aufgabenbereiche der Projektbeteiligten zugeschnitten sind. Da es nicht möglich bzw. sinnvoll ist alle Aufgaben und Funktionen der Gewerke in eine einzige BIM-fähige Software zu integrieren, ist ein konsistenter Datenaustausch zwischen diesen Systemen daher maßgebend. BIM als Basis des Informationsflusses führt zu der Fragestellung wie die Daten exportiert -und die daraus gewonnenen Erkenntnisse wiederum importiert werden können. Der von der buildingSMART entwickelte IFC Datei-Standard ist derzeit nur auf den Hochbau ausgerichtet. Die IFC bridge-Erweiterung für Brückenbauwerke ist noch nicht standardisiert.

In vorangegangenen Arbeiten wurde NX bereits auf die Anwendbarkeit im Bauwesen untersucht. Den Ergebnissen zufolge ist NX für die parametrische 3D-Modellierung von Brücken sehr gut geeignet. Digitale Geländemodelle und Trassierungen können gut aus anderen Programmen in das System importiert werden. 2D-Schalpläne lassen sich einfach ableiten. Des Weiteren können Flächen und Volumen in NX abgefragt werden und daraus die Mengen einzelner Positionen ermittelt werden.

Der Austausch der Daten zwischen NX und der Berechnungssoftware ist als weitere sinnvolle Lösung zu untersuchen. Bauwerke müssen stets auf ihre Standsicherheit und Gebrauchstauglichkeit überprüft werden. Die geometrischen Daten werden meist manuell in die Berechnungssoftware übertragen. Aus der Bemessung des Bauwerks folgen dann ggfs. Anpassungen der Bauteilgeometrie, die wiederum ins 3D-Modell übertragen werden müssen. Besonders in der Entwurfsphase kommt es dabei des Öfteren zu Änderungen des Bauwerks. Ein konsistenter Datenaustausch würde zum einen den großen Aufwand der manuellen Eingabe begrenzen und zum anderen würden mögliche Fehler bei der statischen Bemessung reduziert werden.

1.2 Ziel der Arbeit

Ziel dieser Arbeit ist es, mit Hilfe von IT-Lösungen die Vorteile und Möglichkeiten der BIM-Integration zu erweitern. Es wird dabei untersucht, mit welchen Möglichkeiten das digitale Gebäudemodell als Basis der Tragwerksplanung dienen kann. Da es bis heute kein einheitliches Datenmodell im Ingenieurbau gibt, welches einen konsistenten Datenaustausch aller

Gewerke ermöglicht, werden Direktschnittstellen entwickelt, die den Austausch geometrischer Daten zwischen zwei Softwaresystemen schafft. Konkret sollen die geometrischen Daten aus Siemens NX über ein neutrales Datenformat in das Berechnungsprogramm SOFiSTiK übergeben werden.

Es gibt mehrere Ansätze wie aus 3D-Modellen Tragwerksmodelle generiert werden können. Die buildingSMART Organisation sieht für die statische Bemessung eine Dimensionsreduzierung des 3D-Modells vor, da Bauwerke konventionell als Stab-, Flächen- oder Scheibentragwerke bemessen werden. Andere Forschungsgruppen verfolgen den Ansatz Bemessungen von Bauwerken künftig am dreidimensionalen Gesamtsystem vorzunehmen. Beide Ansätze sollen im Rahmen dieser Arbeit untersucht werden, inwiefern sie in NX integrierbar sind und wie anwendbar die abgeleiteten Modelle sind. Dazu werden zwei Schnittstellen entwickelt, *NX2Sofistik_Vol* und *NX2Sofistik_Area*. Beide Schnittstellen sollen die geometrischen Daten über ein neutrales Datenformat in SOFiSTiK importieren. Über *NX2Sofistik_Vol* wird ein Volumenmodell als Tragwerk in SOFiSTiK erstellt. Mit *NX2Sofistik_Area* wird ein Flächentragwerk generiert.

Die Qualität der beiden Modelle soll anhand eines Anwenderbeispiels überprüft werden. Aus einem schon bestehenden 3D-Modell einer Rahmenbrücke werden über die Schnittstellen jeweils ein Volumen- und ein Flächenmodell als Tragwerk in SOFiSTiK generiert. Zusätzlich wird die Rahmenbrücke konventionell als Flächentragwerk in SOFiSTiK eingegeben. Anschließend werden die geometrischen Abmessungen der drei Modelle verglichen und exemplarische Lastfälle angesetzt. Anhand der resultierenden Schnittgrößen sollen die generierten Modelle bewertet werden und daraus ein „Best Practise“ Prozess bestimmt werden.

1.3 Aufbau der Arbeit

Am Anfang dieser Arbeit wird das Projekt „Eisenbahnüberführung Ventschow“ (kurz EÜ Ventschow) vorgestellt, welches im Rahmen dieser Arbeit als Anwenderbeispiel dient. Im darauffolgenden Kapitel wird das Prinzip von Building Information Modeling erörtert. Weiterhin wird auf die IFC als Standard eingegangen und eine prinzipielle Unterscheidung von Dateiformaten vorgenommen. Anschließend wird die Einteilung der BIM-Methode in die Klassifizierung von F. Jernigan erläutert.

In Kapitel 4 werden die verschiedenen Methoden der Volumenmodellierung vorgestellt. Die heutigen 3D-CAD Systeme bedienen sich unterschiedlicher Darstellungsmethoden, was beim Datenaustausch zu Konvertierungs-Problemen führen kann.

Anschließend werden die möglichen Ansätze diskutiert, die eine Integration der Tragwerksplanung in den BIM-Prozess ermöglichen. Es werden dabei die zwei Ansätze der buildingSMART Organisation und des Schwerpunktprogramms „Vernetzt-kooperative Planungsprozesse im Konstruktiven Ingenieurbau“ aufgegriffen, das digitale Gebäudemodell als dimensionsreduziertes Modell oder als dreidimensionales Gesamtsystem für die Tragwerksplanung wiederzugeben.

Aufbauend auf der theoretischen Grundlage werden in Kapitel 6 und 7 die genannten Tragwerksmodelle generiert. Dazu wird anfangs die parametrische Brückenmodellierung in NX und der strukturelle Aufbau der Komponenten erklärt. Anschließend werden die Maßnahmen erläutert, die für einen erfolgreichen Export der generierten Tragwerksmodelle notwendig sind. In Kapitel 7 wird die konventionelle Systemeingabe der Brücke in SOFiSTiK mit Hilfe der CADINP Sprache aufgezeigt. Zusätzlich werden exemplarisch typische Lastfälle einer Eisenbahnbrücke festgelegt und in SOFiSTiK implementiert.

In Kapitel 8 werden schließlich die Implementierungen der Schnittstellen und deren Klassenstruktur erläutert.

Danach werden die Schnittgrößen und Abmessungen der drei Modelle verglichen und bewertet. Als Resultat wird anhand der Ergebnisse der Modelle ein „Best Practise“ Prozess bestimmt.

Zuletzt schließt die Arbeit mit einer Zusammenfassung aller Erkenntnisse und einem Ausblick über mögliche künftige Forschungsarbeiten ab.

2. Projekt EÜ Ventschow

2.1 Projektvorstellung

Die Anwendbarkeit der im Rahmen dieser Arbeit erstellten Schnittstellen zur automatisierten Generierung von Tragwerksmodellen soll anhand eines Anwenderbeispiels überprüft werden. Ein reales Brückenbauwerk wird als Referenzbauwerk verwendet. Dabei handelt es sich um die Eisenbahnüberführung bei Ventschow, im Folgenden als EÜ Ventschow bezeichnet, die über die Autobahn A14 spannt. Das geplante Bauwerk soll die alte Eisenbahnüberführung ersetzen. Sie wurde im März 2009 im Auftrag des Straßenbauamts Schwerin fertiggestellt.

2.2 Ausführung der EÜ Ventschow

Bei der neuerstellten Eisenbahnüberführung handelt es sich um eine zweiteilige Rahmenbrücke, die in zwei Bauabschnitten in Stahlbetonbauweise hergestellt wurde. In den Bauabschnitten wurden zwei Rahmenbauwerke nördlich und südlich der Endlage hergestellt (vgl. 0). Anschließend wurde der Bahnbetrieb außer Betrieb genommen und die Bestandsbrücke abgebrochen. Daraufhin wurden die neuerstellten Brückenabschnitte parallel über eine Verschubbahn in Endlage verschoben und das Gleisbett wiederhergestellt. Beide Abschnitte sind über eine Längsfuge mit innenliegendem Fugenband über die gesamte Längsachse voneinander getrennt. Da beide Gleisabschnitte geometrisch vergleichbar ausgeführt wurden, kann die Bemessung stellvertretend nur an einem Bauabschnitt erfolgen. Aus diesem Grund wird in dieser Arbeit nur der südliche Gleisabschnitt (Bauabschnitt 1) betrachtet.

3. Building Information Modeling

3.1 BIM Definition

Unter Building Information Modeling (kurz BIM) versteht man eine neuartige Methode der Organisation und Dokumentation von Bauprojekten. Die Idee ist für jedes Bauvorhaben ein einheitliches digitales Gebäudemodell zu erstellen, welches alle Informationen erfasst, die das Bauwerk definieren. Die Projektbeteiligten können die für ihr Gewerk relevanten Daten entnehmen, bearbeiten und die daraus gewonnenen Resultate wiederum ins Modell übertragen. Dies hat den Zweck den Informationsaustausch zu erhöhen und Unstimmigkeiten zu verhindern, da alle auf den gleichen Grundlagen planen (Günthner, et al., 2011).

Zu unterscheiden sind die Begriffe Building Information Modeling und Building Information Model: Building Information Modeling ist die Organisationsmethode zur optimierten Planung und Ausführung von Bauvorhaben und sollte nicht mit Building Information Model, dem digitalen Gebäudemodell verwechselt werden. Das Gebäudemodell entspricht dem CAD-Produkt, das aus dem BIM-Verfahren herausgeht. Viele CAD-Softwarehersteller vermarkten ihre Produkte als „BIM-fähig“. Gemeint ist, dass mit Hilfe der Software das Gebäudemodell erstellt und über die verschiedenen Planungsphasen fortgeschrieben werden kann. Der Detaillierungsgrad des Gebäudemodells steigt somit mit den Leistungsphasen. Zum einen wird die Geometrie im Gebäudemodell visualisiert. Zum anderen werden physikalische Eigenschaften (z.B. Material) und funktionale Beziehungen zwischen den Bauteilen gespeichert. Wird beispielsweise eine Wand verschoben, so verschiebt sich auch das Fenster in der Wand. Da im Idealfall die Projektbeteiligten mit einem einheitlichen Gebäudemodell arbeiten, sind alle Informationen stets konsistent und aktuell.

Beim BIM-Verfahren werden Planung, Konstruktion, Betrieb und Abbruch eines Gebäudes analysiert, was somit dem kompletten Lebenszyklus eines Bauwerkes entspricht. Nach Fertigstellung des Gebäudes wird das digitale Gebäudemodell an das Facility Management übergeben. Es dient dann zur Verwaltung und Instandhaltung des Gebäudes. Sämtliche Ände-

rungen und Sanierungsmaßnahmen werden wiederum im Modell als Information abgespeichert. Am Ende der Nutzungsphase kann mithilfe des Modells ein optimierter Abbruch oder Rückbau, auch im Hinblick auf Recycling, geplant werden (Eastmann, et al., 2011).

3.2 Einsatz von BIM

Die Anforderungen an die Baubranche werden stets größer: Bauwerke werden immer filigraner und ausgefallener, die Kosten sollen dennoch möglichst gering bleiben, während Nachhaltigkeit und Energieeffizienz eine große Rolle spielen. Dennoch kommt es immer wieder zu allbekannten Problemen wie Nichteinhaltung des Fertigstellungstermins, Qualitätsmängel oder Kostenmehrung. Herkömmliche Verfahren der Planung und Organisation scheinen daher, trotz Optimierungen, den heutigen Ansprüchen nicht mehr gerecht werden zu können (Günthner, et al., 2011).

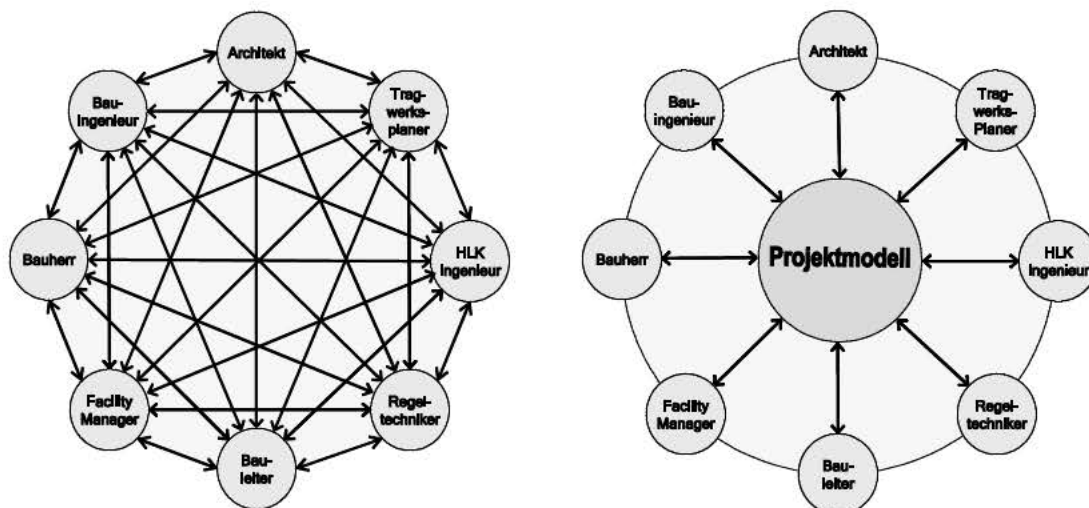


Abbildung 1: BIM-Vergleich mit konventionellen Planungsverfahren nach (Tonke, 1999)

Bei der konventionellen Planung und Ausführung von Projekten gibt es eine Vielzahl von Projektbeteiligten die alle miteinander in Beziehung stehen. Durch die vielen Kommunikations-Schnittstellen ist der Informationsaustausch unübersichtlich und zum Teil lückenhaft. Oftmals arbeiten die Gewerke ungewollt mit überholten oder widersprüchlichen Unterlagen.

Des Weiteren wird das heutige Potenzial der CAD-Systeme bei weitem nicht ausgeschöpft. Sie werden oftmals nur als „digitales Zeichenbrett“ eingesetzt. Bauwerke werden mittels einfacher Linien und Kurven in Grundriss und Schnitten dargestellt. Änderungen bei der Planung müssen dann manuell in alle Pläne übertragen werden. Kommt es dabei zu Unstimmigkeiten zwischen Grundriss und Schnitten sind diese nur schwer zu erkennen. Da viele Programme nicht kompatibel sind, können Pläne und Informationen meist nur in Papierform oder als PDF an andere Beteiligte übergeben werden, sodass es zur wiederholten Eingabe der

Angaben kommt. Inkonsistenz und widersprüchliche Ergebnisse sind oft die Folgen der manuellen Dateneingabe. Die Planung auf reiner CAD-Basis birgt somit die Gefahr zeitintensiv oder fehleranfällig zu werden (Günthner, et al., 2011).

Building Information Modeling als neuer Ansatz weicht sehr von den herkömmlichen Planungsmethoden ab und ist damit eine neue Herausforderung an das Baumanagement. Dennoch bringt die BIM-Methode viele Vorteile mit sich und ist daher sinnvoll. Durch den Ansatz der verstärkten Zusammenarbeit der Projektbeteiligten werden viele Fehlerquellen ausgeschlossen und die Qualität der Bauplanung und Ausführung wird erhöht. Die Informationen werden parametrisch in einer integrierten Datenbank verwaltet. Werden Änderungen vorgenommen, so wirkt sich das auf alle abhängigen Daten aus. Dateninkonsistenz und Informationsverluste werden somit stark reduziert. Alle Daten und Erkenntnisse werden kontinuierlich erweitert und gespeichert und müssen somit nicht mehr innerhalb der Gewerke und Leistungsphasen wiederholt erarbeitet und eingegeben werden. Wird nur auf ein einheitliches Gebäudemodell zugegriffen, wird eine Vielzahl der Schnittstellen zwischen den Gewerken eliminiert. Jedes Gewerk kann die für sich wichtigen Informationen aus dem Gebäudemodell extrahieren. Da das BIM-Modell alle aktuellen Informationen und Änderungen besitzt, arbeiten alle Gewerke stets mit dem neuesten Planungsstand. Die verschiedenen Bauteile werden nicht mehr als reine Linien und Kurven repräsentiert. Jedes Bauteil beinhaltet seine spezifischen Eigenschaften, wie Material, Steifigkeit, Gewicht. Diese Informationen sind maschinenlesbar, wohingegen konventionelle 2D-Pläne nur durch den Mensch interpretiert werden können (Zilch, et al., 2013).

Dank der 3D-Planung können automatisiert konsistente Schnitte und Grundrisse erstellt werden, was eine schnellere und fehlerfreie Planerstellung ermöglicht. Da den Bauteilen bauspezifische Eigenschaften zugewiesen werden, kann die Ressourcenmenge automatisch ermittelt werden und somit auch eine realistische Kostenschätzung erfolgen. Im Allgemeinen werden durch BIM eine Verbesserung der Qualität, Effektivität und eine Risikominimierung erwartet, so wie es sich in anderen Branchen wie der Fahrzeugindustrie schon eingestellt hat (Eastmann, et al., 2011).

3.3 Dateiformate und Standards

Im Bereich Building Information Modeling wurden verschiedenste Dateiformate zur Speicherung der Projekte entwickelt. Neben der Geometrie müssen u.a. Materialien, visuelle Eigenschaften und prozessbegleitende Informationen übergeben werden. Die Schwierigkeit ergibt sich dabei aus der großen Anzahl der beteiligten Gewerke, da jedes seine eigene Software zur Projektbearbeitung nutzt. Der mangelhafte Datenaustausch zwischen den Programmen und dem digitalen Gebäudemodell führt oft zu manueller und wiederholter Eingabe der Daten,

was sehr zeitintensiv ist. Zusätzlich kann es durch willkürliche Annahmen bei der Eingabe zu fehlerhaften Ergebnissen und widersprüchlichen Aussagen kommen (Liebich, et al., 2011).

Viele Software-Unternehmen bieten daher allumfassende BIM-Lösungen an. Mit ihren Software-Produkten kann ein großer Teil der Gebäudeplanung, Analyse und Simulation bearbeitet werden und sie sind so aufeinander abgestimmt, dass ein reibungsloser Ablauf innerhalb der Programme funktioniert. Ihr Ziel ist es die Kunden an ihre Software zu binden, daher erfolgt der Datenaustausch zwischen den Programmen über proprietäre Dateiformate.

Besonders für öffentliche Auftraggeber ist es jedoch wichtig Leistungen neutral auszuschreiben, sodass Firmen an keine spezielle Softwarelösung gebunden sind. Das Augenmerk des BIM-Ansatzes ist daher die Einführung eines unternehmensübergreifenden, wiederverwendbaren Dateiformates.

Es findet daher grundsätzlich eine Unterscheidung zwischen proprietären und offenen Dateiformaten statt. Im Folgenden werden diese daher kurz vorgestellt.

3.3.1 Proprietäre Dateiformate

Proprietäre Dateiformate sind meist vom Softwarehersteller entwickelt und an die jeweilige Software gebunden. Aus wirtschaftlichen Interessen sind sie urheberrechtlich geschützt und für Drittanbieter nicht öffentlich, sodass firmenfremde Programme diese Dateien nicht lesen können (Born, 1996). Unternehmen können selber wählen für welche Software sie sich entscheiden. Da proprietäre Dateien aber nicht von anderen Programmen gelesen werden können, sind sie für das BIM-Verfahren zum Datenaustausch nicht geeignet (Eastmann, et al., 2011). Dateiformate wie DXF und DWG der Firma Autodesk sind weit verbreitet und werden oft als „quasi“ Standard gesehen. Dennoch handelt es sich auch bei ihnen um proprietäre Dateiformate.



Abbildung 2: Anzahl benötigter Direktschnittstellen (Friedewald, et al., 2011)

Um den Datenaustausch mit geschlossenen Formaten zu ermöglichen werden oft sog. Direktschnittstellen entwickelt, die jedoch meist mit Lizenzgebühren an den Softwarehersteller verbunden sind. Man spricht von einer Direktschnittstelle, da der Datenaustausch nur direkt von einer Software zu einer anderen erfolgen kann. Des Weiteren gehen durch die Konvertierung oftmals Informationen verloren (Willibald, et al., 2011).

3.3.2 Offene Dateiformate

Dateiformate, die uneingeschränkt bearbeitet und editiert werden können, werden als offene Formate bezeichnet. Ihr klarer Vorteil ist, dass diese Dateien von jedem Benutzer ungehindert geöffnet und bearbeitet werden können, unabhängig von der konkreten Ausrüstung des Computers. Rechtliche oder technische Restriktionen wie bei proprietären Formaten gibt es nicht. Somit haben Softwareentwickler die Möglichkeit Softwarelösungen auszuarbeiten oder weiterzuentwickeln. Offene Dateiformate beruhen auf offenen Standards wie der ISO (Born, 1996).



Abbildung 3: Anzahl benötigter Neutralschnittstellen (Friedewald, et al., 2011)

Bei offenen Dateiformaten spricht man von Neutralschnittstellen. Der Arbeitsaufwand zur Wartung der Schnittstellen ist deutlich kleiner als bei Direktschnittstellen. Neutralformate sind oftmals leichtgewichtig, sodass sie weniger Speicherplatz benötigen und schnelle Übermittlungszeiten ermöglichen, was für eine Handhabung der großen Informationsmenge eines digitalen Gebäudemodells entscheidend ist (Friedewald, et al., 2011).

3.3.3 IFC als BIM-Standard

Für die optimale Umsetzung der BIM-Methode ist es wichtig, dass der Datenaustausch zukünftig über einen einheitlichen offenen Standard erfolgt. Zu diesem Zweck wurde die buildingSMART Organisation, früher IAI (International Alliance für Interoperability) gegründet. Sie besteht aus einem Zusammenschluss von Firmen, Hochschulen und Privatpersonen

aus dem Bausektor. Ihr Ziel ist es, das digitale Gebäudemodell zur gemeinsamen Datennutzung verfügbar zu machen und es dabei nicht an eine bestimmte Software zu binden. Dazu wurden die Industry Foundation Classes (IFC) definiert, einem offenen Standard, welcher den Datenaustausch zwischen zahlreichen Softwareprogrammen ermöglicht (Willibald, et al., 2011).

Die IFC entstammt zu großen Teilen dem STEP-Standard (Standard for the exchange of product model data), welches weitreichend in der Automobil-, Luftfahrt-, und Schiffsbauindustrie etabliert ist. Mit der aktuellen Version IFC4 entspricht die IFC nun dem offiziellen ISO-Standard ISO 16739. Es sei aber erwähnt, dass sie momentan nur die Bereiche des Hochbaus erfasst. Die IFC bridge für Brückenbauwerke beispielsweise kommt derzeit noch keinem Standard gleich (Günthner, et al., 2011).

Man unterscheidet zwischen dem IFC-Datenschema und der IFC-Datei. Die IFC Datei (.ifc) dient als Container zur Übergabe und Speicherung der Daten. Das IFC Datenschema ist in der EXPRESS Sprache geschrieben, welche ebenfalls beim STEP-Standard Anwendung findet. Ziel ist es, das Bauwerksmodell neutral in einem immer gleichen Schema zu beschreiben. Nur wenn alle Informationen eines Bauwerks in diesem Schema vorliegen, ist die Kompatibilität des Datenflusses möglich (Kuloyants, 2014).

```
ENTITY IfcDoor
  SUBTYPE OF (IfcBuildingElement);
  OverallHeight : OPTIONAL IfcPositiveLengthMeasure;
  OverallWidth : OPTIONAL IfcPositiveLengthMeasure;
END_ENTITY;
```

Abbildung 4: IfcDoor (Tür) beschrieben mit dem EXPRESS-Schema (buildingSMART, 2014)

3.4 Klassifizierung der BIM-Lösung

Je nach Durchgängigkeit der Wertschöpfungsketten kann man die BIM-Methode in verschiedene „Stufen“ einteilen. Nach F. Jernigan (Jernigan, 2008) lässt sich die Skalierbarkeit in zwei Kategorien einteilen:

- Durchgängigkeit der BIM-Lösung
- Offenheit des BIM-Standards

Es ist also entscheidend, ob BIM von allen Beteiligten über alle Leistungsphasen verwendet wird und in wie fern der Datenaustausch zwischen verschiedenen Softwarelösungen funktioniert. Abbildung 5 zeigt die Einteilung der BIM Anwendung in vier Stufen:

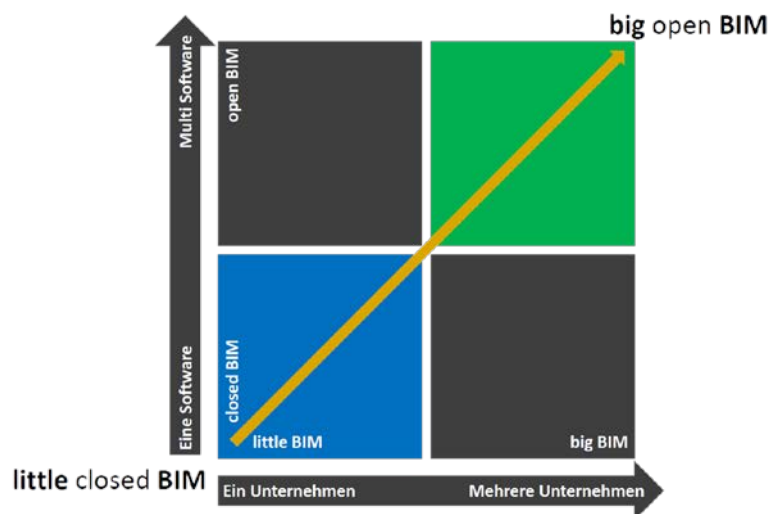


Abbildung 5: Klassifizierung der BIM Anwendung (buildingSMART, 2014)

„little closed BIM“

Die BIM-Methode wird lediglich von einem Projektbeteiligten angewandt. Er nutzt das virtuelle Gebäudemodell nur für sich innerhalb seines Aufgabenfeldes, ohne es zum Datenaustausch mit anderen Mitwirkenden bereitzustellen. Meist arbeitet der Anwender nur mit der Software eines Herstellers. Das Gebäudemodell liegt nur unter dem proprietären Dateiformat vor. Diese Stufe kann daher als geschlossene Insellösung betrachtet werden.

„little open BIM“

Die offene Insellösung charakterisiert sich dadurch, dass BIM innerhalb eines Unternehmens angewandt wird. Typischerweise werden nur die Softwareprodukte eines Herstellers genutzt, deren Schnittstellen zwar über proprietäre Dateiformate laufen, für den Informationsaustausch werden die Daten aber den anderen Projektbeteiligten mittels offenem Standard übergeben.

„big closed BIM“

Bei der geschlossenen BIM-Integration wird vertraglich vereinbart, dass die Projektbeteiligten in Kooperation die BIM-Methode anwenden. Die eingesetzte Software für das virtuelle Gebäudemodell ist festgelegt und stammt von einem einzigen Softwareanbieter. Der Datenaustausch zwischen den Mitwirkenden erfolgt innerhalb der proprietären Softwareumgebung.

„big open BIM“

Die Projektbeteiligten verschiedener Disziplinen arbeiten in Kooperation gemeinsam an einem virtuellen Gebäudemodell. Die eingesetzte Software ist nicht vorbestimmt, der Datenaustausch ist vertraglich mittels offenem Datenschema festgelegt. Alle Informationen werden in einem einheitlichen virtuellen Gebäudemodell abgespeichert.

Die offene BIM-Integration stellt die höchste Stufe der BIM Einbindung dar, da nur hier alle Vorteile der BIM-Methode ausgeschöpft werden können (Liebich, et al., 2011).

4. Geometrische Modellierung von Volumenmodellen

Die BIM-Integration ist von der Qualität der Schnittstellen zwischen der BIM-Software und anderer Software abhängig. Neben der Frage des Datentransfers gibt es noch die Problematik der Anpassung der Daten an die jeweilige Aufgabenstellung. Selbst wenn ein gemeinsames Datenformat verwendet wird, ist nicht sichergestellt, dass die Daten von jeder Software gleich interpretiert werden. Besonders schwierig ist die Darstellung geometrischer Verhältnisse.

Unter Geometrischer Modellierung versteht man die computergestützte Beschreibung geometrischer Objekte. Die heutigen 3D CAD-Systeme arbeiten mit sog. Körpermodellen (engl.: Solid Model). Dabei gibt es unterschiedliche Methoden Kurven, Flächen oder Volumenmodelle abzubilden. Je nach Anwendungsgebiet gibt es bevorzugte Repräsentationsformen, die für die jeweilige Aufgabe am besten geeignet sind. Es gibt zwar Möglichkeiten die Darstellungsarten in andere zu konvertieren, dies geschieht jedoch selten ohne Informationsverlust.

Hier soll nun auf die Beschreibung von Volumenmodellen eingegangen werden. Es gibt direkte und indirekte Darstellungsmethoden von Volumenkörpern, sowie Hybridschemata, die beide Methoden kombinieren.

4.1 Direkte Darstellungsmethoden

Bei den direkten Darstellungsmethoden wird das Volumen selbst mittels einfacher Grundkörper modelliert. Im Folgenden wird auf die wichtigsten Methoden kurz eingegangen.

4.1.1 Normzellenschema

Der Definitionsraum des Volumenkörpers wird in gleich große Zellen (Normzellen) unterteilt. Meist handelt es sich bei den Normzellen um Würfel (Voxel), theoretisch sind aber auch andere Grundkörper wie Quader oder Prismen möglich. Mit dem Normzellenschema lassen

sich einfache Bit-Matrizen erstellen (Zelle enthalten, Zelle nicht enthalten). Die Darstellungsgenauigkeit des Volumenkörpers ist von der Größe der Zellen abhängig. Je kleiner die Zellen sind, desto besser wird der Körper approximiert. Jedoch steigt auch der Speicheraufwand mit steigender Genauigkeitsanforderung stark an. Ein weiterer Nachteil ist, dass gekrümmte Objekte nie exakt dargestellt werden können und auch sämtliche Informationen der Oberflächenkrümmung verloren gehen (Encarnaçã, et al., 1990).

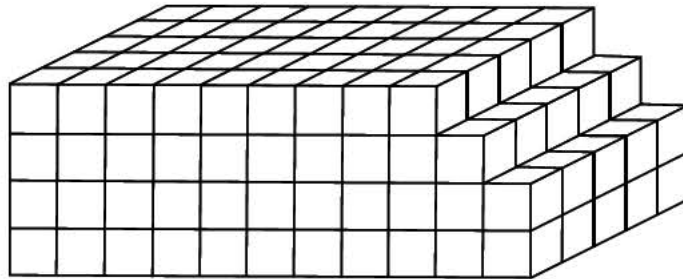


Abbildung 6: Normzellenschema (Bungartz, et al., 1996)

4.1.2 Oktalbäume (Octrees)

Ähnlich wie beim Normzellenschema wird bei dieser Methode der Volumenkörper durch ein Gitter aus Zellen (Würfel) aufgeteilt. Allerdings ist die Größe der Zellen hier nicht konstant. Geometrisch kompliziertere Stellen werden durch kleinere Zellen dargestellt, während einfachere Stellen aus größeren Zellen gebildet werden. Die Herangehensweise ist folgendermaßen: Ein Würfel, der den gesamten Volumenkörper umschließt, wird so lange in Würfel halber Kantenlänge geteilt bis die Teilwürfel entweder gänzlich innerhalb (in) oder außerhalb (off) des Volumenkörpers liegen oder dessen Rand schneiden (on). Abbildung 7 zeigt ein Quadtree, das zweidimensionale Abbild eines Oktalbaums. Im Gegensatz zu Normalzellen sind Oktalbäume deutlich speichereffizienter. Für Simulationszwecke sind Oktalbäume jedoch ungeeignet, da die Oktalbaum-Struktur keine Objektorientierung beinhaltet. Dennoch werden sie als Hilfsmittel in vielen geometrischen Algorithmen integriert (Encarnaçã, et al., 1990).

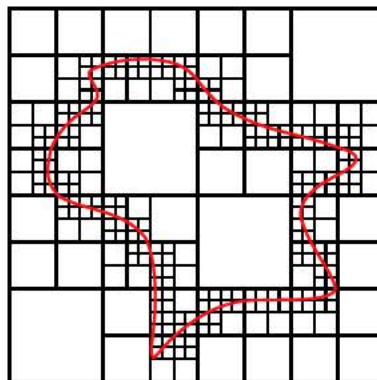


Abbildung 7: zweidimensionaler Quadtree (Encarnaçã, et al., 1990)

4.1.3 Constructive Solid Geometry, CSG

Bei der Constructive Solid Geometry (deutsch: konstruktive Festkörpergeometrie) werden Objekte mittels primitiven Grundkörpern, wie Zylinder, Quader, Kugel und booleschen Operatoren (Vereinigung, Schnitt, Differenz) zusammengesetzt. CSG lässt sich mit Hilfe eines Binärbaumes beschreiben. Die Blätter stellen die Grundprimitive dar, die Knoten entsprechen den Operatoren. Das CSG-Schema kann erweitert werden, indem Halbräume und Transformationen wie Rotation, Translation und Skalierung der Menge hinzugefügt werden. Es lassen sich damit intuitiv komplexe Geometrien modellieren, weshalb CSG weit verbreitet ist (Frank, 2000).

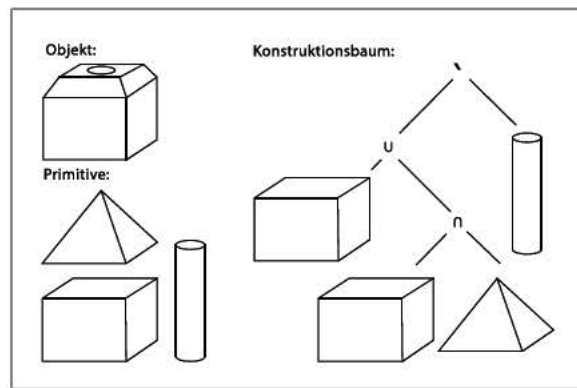


Abbildung 8: CSG als Binärbaum nach (Frank, 2000)

4.2 Indirekte Darstellungsmethoden

Man spricht von indirekten Darstellungsmethoden, wenn Volumenkörper über deren Oberfläche, Kanten und Knoten beschrieben werden. Im Gegensatz zu direkten Darstellungsmethoden muss die Ausrichtung des Volumens definiert werden um den Volumenkörper eindeutig zu beschreiben.

4.2.1 Drahtmodell

Beim Drahtmodell (engl.: wire frame model) wird das Objekt nur über dessen Kanten definiert. Der Vorteil dieser einfachen linienbasierten Methode ist, dass sie besonders schnell und effizient ist. Da es allerdings keinerlei Information über den Oberflächenverlauf gibt, kann es leicht zu Mehrdeutigkeit kommen. Verdeckte Linien, die z.B. auf der Rückseite des Körpers liegen, können vom Computer nicht erkannt oder ausgeblendet werden. Ein Drahtmodell kann unter Umständen mehrere Körper darstellen oder sogar keinen realen Körper entsprechen. Viele 3D CAD-Systemen bieten die Möglichkeit Objekte als Drahtmodell darzustellen.

Eine reine Anwendung dieser Methode ist, auf Grund der Mehrdeutigkeit, nicht sinnvoll (Bungartz, et al., 1996).

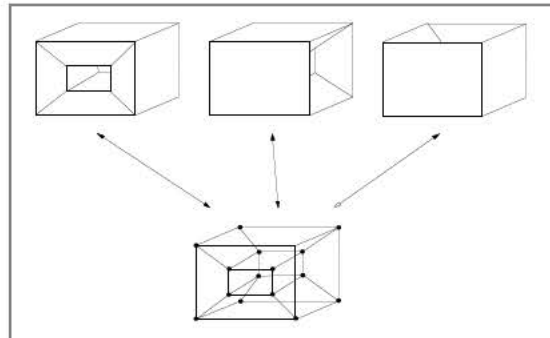


Abbildung 9: Mehrdeutigkeit beim Drahtmodell (Bungartz, et al., 1996)

4.2.2 Boundary Representation (B-rep)

B-reps sind das wohl am häufigsten eingesetzte Darstellungsschema. Bei diesem Modell wird das Volumen eines Objekts über seine Körperoberfläche beschrieben. Zur eindeutigen Definition muss sichergestellt sein, dass es sich um eine geschlossene Hülle handelt. Außerdem muss eine Orientierbarkeit gegeben sein und eine Selbstdurchdringung ausgeschlossen werden. Die Berandung eines B-rep Modells setzt sich in der Regel aus Knoten (vertex), Kanten (edge) und Flächen (face) zusammen. Die Topologie (Nachbarschaftsbeziehungen zwischen Punkten, Kanten und Flächen) eines Körpers wird separat von seiner Geometrie betrachtet.

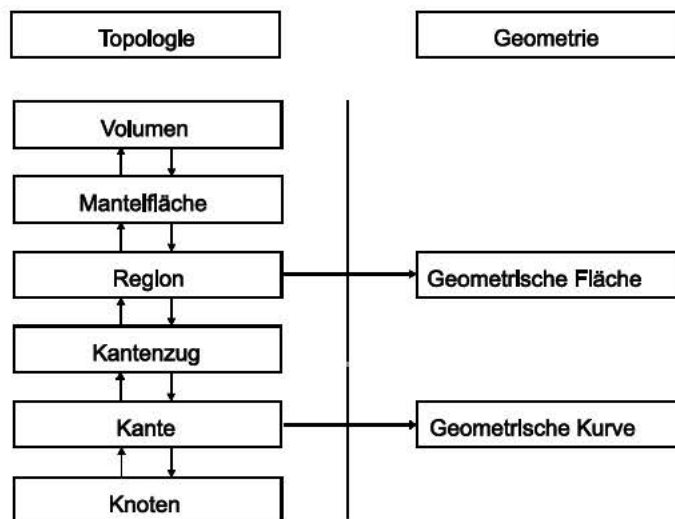


Abbildung 10: Topologische und geometrische Hierarchie nach (Sorgner, 2012)

Die Topologie wird anhand des sogenannten *vef*-Graphen (vertex, edge, face) beschrieben. Jeder Körper wird durch eine geschlossene Außenhülle, bestehend aus mehreren Flächen, begrenzt. Jede Fläche ist wiederum durch einen geschlossenen Kantenzug umgeben und jede

Kante wird durch einen Anfangs- und Endpunkt definiert. Besitzt der Körper zusätzlich noch Hohlräume, so werden diese ebenfalls durch eine innere Randbeschreibung definiert.

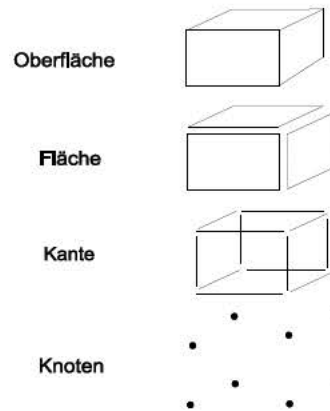


Abbildung 11: Oberflächendarstellung nach (Encarnaçao, et al., 1990)

Die Topologie bezieht sich nur auf die Nachbarschaftsverhältnisse. Die Geometrie gibt die Lage, die Gestalt und den Verlauf eines Objekts wieder. Um einen Körper exakt zu beschreiben müssen den topologischen Elementen geometrische Informationen zugefügt werden. Zur Realisierung von gekrümmten Kurven und Flächen werden sogenannte Freiformkurven und Freiformflächen mittels Splines integriert (Sorger, 2012).

4.2.3 Facettenmodell

Eine spezielle Form des B-rep Modells ist das Facettenmodell, bei der die Oberfläche des Körpers aus einem Netz aus Drei- oder Vierecken zusammengesetzt wird. Viele Algorithmen der Computergrafik arbeiten nach diesem Prinzip, da plane Flächen einfacher zu handhaben sind als komplexe Freiformen. Es sei jedoch erwähnt, dass gekrümmte Flächen nur approximiert werden und nie exakt dargestellt werden können. Die Auflösung der Netzdichte richtet sich nach der Komplexität der Oberflächenstruktur. Durch die große Anzahl an Elementen ist der Speicherbedarf oft umfangreich. Operationen wie Schattierung der Oberflächen, Bestimmung von innen/außen sind einfacher, als bei anderen Darstellungsschemata. Auch viele Finite-Elemente-Methoden greifen auf diese Darstellungsmethode zurück. Zur Volumenmodellierung werden dann die Dreiecksnetze zu Tetraeder-Elementen erweitert.

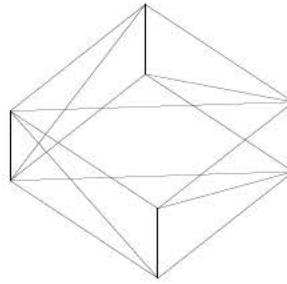


Abbildung 12: Facettenmodell eines Quaders

Es kann eine vereinfachte Version des *vef*-Graphen verwendet werden, indem immer drei Knoten eine Dreiecksfläche definieren. Zuerst wird ein Anfangsdreieck aus drei Knoten definiert. Aus jedem weiteren Knoten kann dann ein weiteres Dreieck gebildet werden (Frank, 2000).

4.2.4 Extrusionsmodelle

Der Volumenkörper wird durch Verschiebung der Mittelfläche entlang einer Leitkurve erzeugt. Die meisten Bauwerke lassen sich durch eine Fläche im Raum und Bauteildicke definieren. Bei einfachen Extrusionsmodellen ist die Dicke des Körpers konstant und die Ober- und Unterfläche topologisch identisch. Auch wenn so nur sehr einfache Volumenkörper erzeugt werden können, wird diese Methode oft für Berechnungsverfahren oder im CAD angewendet.

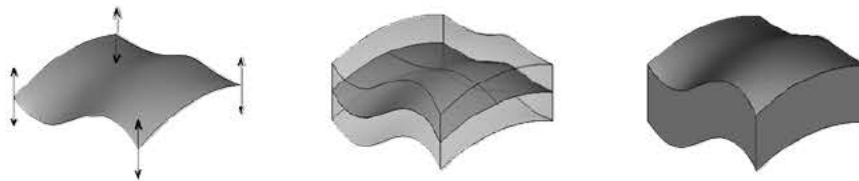


Abbildung 13: Extrusion der Mittelfläche (Sorgner, 2012)

Die extrudierten Körper können grundsätzlich in CSG-Modellen eingefügt werden. Zur Weiterverwendung wird jedoch geraten das Modell in ein B-rep-Modell umzuwandeln (Bungartz, et al., 1996).

4.2.5 Hybridmodelle

Bei Hybridmodellen ergänzen sich mehrere unterschiedliche Darstellungsmethoden und deren mathematische Beschreibungen zur Volumenmodellierung. Da für jedes Anwendungsgebiet andere Darstellungsmethoden geeignet sind, können so alle Vorteile genutzt werden. Durch Konvertierungsalgorithmen können die Daten von einem Darstellungsschema in ein anderes konvertiert werden. Bei der Konvertierung kann es jedoch zu Konsistenz- wie auch Gültigkeitsproblemen kommen. Werden die Daten in ein weniger mächtiges Schema konvertiert kommt es zu Informationsverlusten. Die Umstellung von einer Darstellungsart in die andere ist somit nicht für jedes Schema gut geeignet. Viele CAD-Softwarehersteller halten daher mehrere Schemata, wie z.B. das CSG- und B-rep Schema, vor. Während der Benutzer mit dem CSG-Modell leichter modellieren kann wird für die Visualisierung das B-rep Schema verwendet (Encarnaç o, et al., 1990).

5. Das digitale Gebäudemodell als Basis der Tragwerksplanung

Aufgabe der Tragwerksplanung ist es, bei komplexen Bauvorhaben ein statisches Tragwerk zu konzipieren und zu bemessen, so dass während der gesamten Nutzungsdauer Standsicherheit und Gebrauchstauglichkeit gewährleistet sind. Die statischen Berechnungen werden anhand von Tragwerksmodellen durchgeführt, die das Tragverhalten des realen Bauwerks vereinfacht beschreiben. Als Ergebnis entstehen am Ende Schal- und Bewehrungspläne, die der Bauausführung in Papierform übergeben werden. Zur Berechnung des Tragwerks werden üblicherweise dimensionsreduzierte Systeme, wie Balken, Platten oder Scheiben, herangezogen. Das Tragwerksmodell stellt somit immer eine Abstraktion des eigentlichen Bauwerks dar. Durch die Dimensionsreduzierung können mehrere statische Systeme interpretiert werden. Der Tragwerksplaner entscheidet für jedes Bauteil individuell, welches Tragverhalten er ansetzt (Hartmann, 2000).

BIM sieht eine Integration aller Gewerke in der computergestützten Planung vor. Während der Planung ist eine Vielzahl an Fachplanern eingebunden, mit unterschiedlichen Anforderungen an die fachspezifische Software. Eine durchgängige computergestützte Planung muss auf einer zentralen Datenverwaltung basieren. Die Einbeziehung aller Informationen in einem einzigen Datenmodell ist jedoch nicht immer sinnvoll. Es können auch mehrere Partialmodelle abgeleitet werden, die die verschiedenen Fachbereiche abbilden. Partialmodelle enthalten nur die gefilterten Daten, die für den fachspezifischen Aufgabenbereich notwendig sind. Alle Partialmodelle müssen die gleiche Grundstruktur aufweisen, um die Daten ineinander überführen zu können. Wartung und Konsistenzprüfung werden jedoch durch fachspezifische Erweiterungen des Gebäudemodells deutlich aufwendiger.

5.1 Überführung der Gebäudemolldaten in die Tragwerksplanung

Die Eingliederung der Tragwerksplanung in die BIM-Koordination ist ein Einschnitt in den Datentransfer. Konventionell werden Tragwerke als Stab- oder Flächenmodelle berechnet. Der Übergang vom dreidimensionalen Gebäudemolld zum dimensionsreduzierten Tragwerksmodell als vollautomatische Generierung gestaltet sich jedoch schwierig. Die Semantik der Modelle bedingt, dass sich Bauteile des Gebäudemollds durch mehrere statische Systeme darstellen lassen. Durch Algorithmen können inzwischen einige Hersteller innerhalb ihrer Softwareprodukte geometrisch einfache Bauwerke als dimensionsreduzierte Modelle darstellen. Komplexe Tragstrukturen müssen weiterhin vom Ingenieur beurteilt und statische Systeme abgeleitet werden. Auch die Rückübertragung der Ergebnisse der Bemessung in das Gebäudemolld ist noch nicht ausreichend geklärt (Niggli, 2008).

Es gibt mehrere Ansätze zur Lösung dieser Problematik. Die IAI Arbeitsgruppe Tragwerksplanung arbeitet an einer Erweiterung der IFC Spezifikation. Das Projekt ST-4 (Structure Analysis Domain) hat das Ziel Molldaten für die Tragwerksplanung computergestützt auszutauschen. Zusatzinformationen zur Dimensionsreduzierung der Modelle sind Bestandteil der ST-4. Allerdings beschränkt sie sich bisher ebenfalls nur auf einfache geometrische Modelle. Die Problematik der Dimensionsreduzierung ist somit auch hier nicht ausreichend geklärt. Vorteile der ST-4-Erweiterung sind die Normierung der Lasten und Lastfälle, die mechanische Kopplung der Bauteile, sowie die Lagerbedingungen (Romberg, 2007). Die meisten CAD-Programme benutzen physikalische Modelle für die Geometriebeschreibung. Statik-Programme basieren jedoch auf analytischen Modellen, die Informationen über Knoten und Wirkungslinien, Auflagerbedingungen, Querschnitte und Steifigkeiten übergeben. Für einen erfolgreichen Datenaustausch über die IFC Statik-Schnittstelle ist eine Implementierung des analytischen Modells in CAD-Applikationen notwendig, so dass diese neben den physikalischen Modellen vorgehalten werden. Derzeit bieten noch wenige CAD-Softwarehersteller diese Implementierung an (buildingSMART, 2014).

Ein anderer Ansatz wurde von der Deutschen Forschungsgemeinschaft (DFG) im Schwerpunktprogramm (SPP) „Vernetzt-kooperative Planungsprozesse im Konstruktiven Ingenieurbau“ verfolgt. Sie sehen für den Einsatz einer durchgehend kooperativen Arbeitsumgebung nur die Möglichkeit einer strikten Verwendung von dreidimensionalen, volumenorientierten Modellen und das Rechnen am Gesamtsystem. Dies steht im starken Kontrast zu herkömmlichen Methoden, wo Bauwerke in Tragstrukturen unterteilt werden und diese einzeln bemessen werden. Ziel ist es die Tragwerksplanung besser zu integrieren und die Kom-

plexität der zentralen Datenhaltung zu verringern. In Branchen, wie der Luft- und Raumfahrt oder dem Maschinenbau, wurde dieser Vorteil bereits erkannt und Berechnungen an Volumenmodellen haben sich durchgesetzt. Dass die Baubranche in diesem Trend hinterherhinkt liegt an den besonderen Eigenschaften von Bauwerken und den Randbedingungen des Planungs- und Bauprozesses. Der Ansatz des SPP, alle Berechnungen an einem Gesamtsystem durchzuführen, setzt große Veränderungen in der Tragwerksplanung voraus. Neben einer allgemeinen Umstrukturierung der Herangehensweise müssen höhere Anforderungen an die Softwaresysteme gestellt werden (Rüppel, 2007).

5.2 Implementierungsmöglichkeiten unter Siemens NX

Die Firma Obermeyer arbeitet im Bereich Ingenieurbauwerke mit der Software NX zur Erstellung von digitalen Gebäudemodellen. NX ist eine CAD/CAM/CAE-Software der Siemens Industry Solution and Services GmbH und wurde für Konstruktions- und Fertigungsprozesse entwickelt. Grundsätzlich wurde NX für die Maschinenbau-Branchen konzipiert. In diesen Branchen wurden die Vorteile der parametrischen Modellierung bereits früh entdeckt und eingebunden. Im Gegensatz dazu, bieten die klassischen bauspezifischen CAD Programme noch keine hinreichenden Möglichkeiten der parametrischen Modellierung im Brückenbau. Brückenbauwerke sind linienorientierte Bauwerke, die stark von der umgebenden Trassierung und der Straßenachse abhängig sind. Durch parametrische Abhängigkeiten kann das Bauwerk schnell bei Planungsänderungen angepasst werden (Ji, et al., 2010). Das parametrische Modellieren eines Brückenbauwerks in NX wird in Kapitel 6 weiter erläutert.

Anforderungen aus dem Bauingenieurwesen werden in NX nicht berücksichtigt, was sich als Einschränkung erweist. NX bietet keine Schnittstelle zu bauspezifischer Software an. Im Rahmen dieser Arbeit werden daher Lösungen vorgestellt, wie die vorhandenen Daten des in NX modellierten Gebäudemodells zur Tragwerksanalyse exportiert werden können. Zu diesem Zweck werden nur neutrale Dateiformate für den Datenaustausch betrachtet. Es werden dabei die in Kapitel 5.1 genannten Varianten untersucht, das Modell als Volumen- und als dimensionsreduziertes Flächenmodell wiederzugeben.

Für den Export als Flächenmodell stellt sich die Problematik der Dimensionsreduzierung des Gebäudemodells. NX bietet keine Möglichkeit ein vollautomatisches Flächenmodell im Sinne der Tragwerksplanung anhand des Gebäudemodells zu erstellen. Die IFC-Spezifikation ist in NX nicht enthalten und die vorhandenen neutralen Dateiformate sind nicht zur Dimensionsreduzierung ausgelegt. Daher sind folgende Punkte zu klären, um eine Dimensionsreduzierung dennoch zu ermöglichen:

- Erzeugen der Mittelflächen
- Lage der Mittelfläche
- Übergabe der Bauteilstärke
- Ausbildung der Mittelfläche bei Überbauten
- Kopplungsbedingungen der Bauteile
- Gründung

Das Gebäudemodell stellt das geplante Bauwerk möglichst realitätsnah dar. Bauteilverrundungen, Fasen oder gevoutete Bauteile werden somit ebenfalls abgebildet. Die Ermittlung eines idealisierten Tragwerks wird dadurch jedoch erschwert. In der Tragwerksplanung wird die Mittelfläche im vereinfachten Bauteil angesetzt. Fasen o.ä. werden oft dabei vernachlässigt. Die Lage der Mittelfläche und die Wahl der Bauteilstärke werden für geometrisch vereinfachte Systeme angenommen. Eine vollautomatische Generierung eines statischen Tragwerks in NX gestaltet sich somit als schwierig.

Besonders beim Überbau werden Bauteile auf Grund der Lasteinleitung oftmals besonders approximiert. So wird z.B. bei Brückenplatten für die Lasteinleitung über die Kragarme das Flächentragwerk konventionell als gekoppelte Flächen modelliert. Wie in Abbildung 14 dargestellt, kann der Überbau somit nicht als automatisch generierte Mittelfläche dargestellt werden.

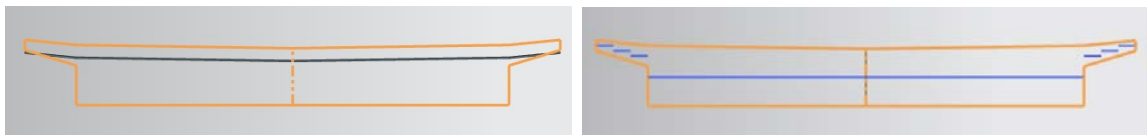


Abbildung 14: von NX generierte Mittelfläche (links) und gekoppelte Mittelfläche (rechts)

Als weitere Schwierigkeit ergibt sich, wie die Mittelflächen angrenzender Bauteile miteinander gekoppelt werden können. Um ein statisches Tragwerk zu erzeugen, müssen die Flächen verbunden werden. Dies kann entweder über Verlängerungen der Flächen bis hin zum anschließenden Bauteil geschehen oder die Bauteile werden mittels Federelemente als Kopplung verbunden.

Eine vollautomatische Generierung eines Tragwerksmodells aus NX ist auf Grund der aufgeführten Punkte daher äußerst komplex. Es ist jedoch möglich durch Verwendung weniger Funktionen Flächenmodelle in NX anhand des bestehenden Gebäudemodells zu erstellen.

Ziel dieser Arbeit ist es, zu überprüfen, ob eine Erstellung des Tragwerkmodells auf Basis des Gebäudemodells sinnvoll erscheint. Es werden daher zunächst vereinfachte Annahmen für die Modelle getroffen:

- Konstante Bettung der Fundamente
- Konstante Bauteildicken innerhalb eines Bauteils
- Bestimmung der Bauteildicke in der FEM-Software

Die gewählten Methoden zur Erstellung der Tragwerksmodelle werden in Kapitel 6.3 erläutert.

6. Modellierung der EÜ Ventschow in Siemens NX

Im Rahmen dieser Arbeit werden Ansätze untersucht, wie aus dem digitalen Gebäudemodell in NX strukturelevante Daten in SOFiSTiK übergeben werden können. In diesem Kapitel wird daher zuerst das allgemeine Vorgehen der parametrischen Brückenmodellierung erläutert und wichtige Befehle in NX vorgestellt. Da der Datenaustausch zwischen NX und SOFiSTiK über ein neutrales Datenformat erfolgen soll, ist zu klären welches Format für diesen Zweck gut geeignet ist und welche Anwendergrenzen sich daraus ergeben. Anhand dieser Randbedingungen sind Maßnahmen zu entwickeln, wie eine erfolgreiche Datenübertragung möglich ist. Diese sind für die Methoden „Volumenmodell“ und „Mittelflächenmodell“ separat zu betrachten.

6.1 Allgemeine Vorgehensweise bei Brückenbauwerken

Brückenbauwerke sind linienorientierte Bauwerke, die stark von der umgebenden Trassierung abhängig sind. Ziel ist es, bei Änderungen der Trassierung oder bei Bauteilanpassungen, das Gesamtbauwerk daran konsistent und automatisiert anzugleichen. Dies wird durch Einhaltung einer Hierarchiestruktur und durch parametrische Eingabe garantiert. Die Firma Obermeyer hat einen solchen Strukturaufbau für sich in Siemens NX entworfen, um Anpassungsfähigkeit, Wiederverwendbarkeit und Übersichtlichkeit zu optimieren. Nachfolgend wird dieser genauer erläutert (Hochmuth, 2012).

6.1.1 Modellierstruktur und Baugruppen-Prinzip

Die Baugruppe ist das übergeordnete Element und repräsentiert das geplante Bauwerk. Hier werden alle enthaltenen Komponenten zusammengeführt. In der Baugruppe sind nur noch Lagebeziehungen enthalten, jedoch keine Geometrieinformationen. Werden Änderungen an den Komponenten vorgenommen, so wirken sich diese auf die Baugruppe aus.

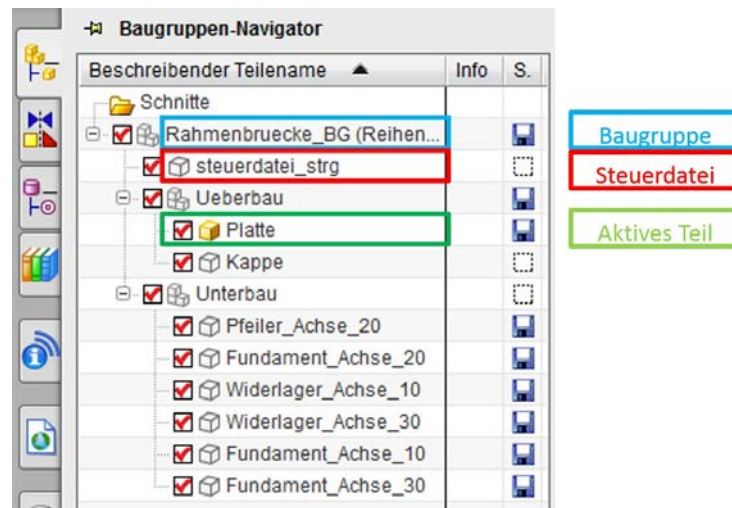


Abbildung 15: Baugruppenstruktur in Siemens NX nach (Weinholzer, 2014)

Die Steuerdatei beinhaltet einzig die Trassierung und den Anschluss an das digitale Geländemodell (DGM) bzw. die Böschung der unterführten Straße. Durch die Eingliederung des Bauwerks in den Geländeausschnitt können realitätsnahe Visualisierungen erstellt werden und geben einen Gesamtüberblick über das Bauvorhaben. Die Steuerdatei entspricht der höchsten Ebene der Modellier-Struktur, d.h. Änderungen der Trassierung müssen sich automatisch auf alle unterordneten Elemente auswirken. Die Trassen und das DGM werden über eine Schnittstelle mit ProVI in NX eingelesen (Hochmuth, 2012). ProVI ist ein von Obermeyer entwickeltes Softwaresystem, basierend auf AutoCAD, das für Infrastrukturplanungen im Bereich Straße, Schiene und Kanal eingesetzt werden kann.

Zu den untergeordneten Elementen zählen Bauabschnitte und Bauteile. Untergeordnete Elemente können selbst ebenfalls Unterordnungen haben. Diese sind nicht notwendig, dienen aber der besseren Übersicht. Die Komponenten einer Brücke sind meist in „Überbau“, „Unterbau“ und ggfs. „optionale Bauteile“ eingeteilt. Eine Aufteilung der Komponenten in die verschiedenen Bauabschnitte ist ebenfalls denkbar. Nachfolgende Grafik zeigt alle möglichen Bauteile einer Brücke (Wang, 2012).

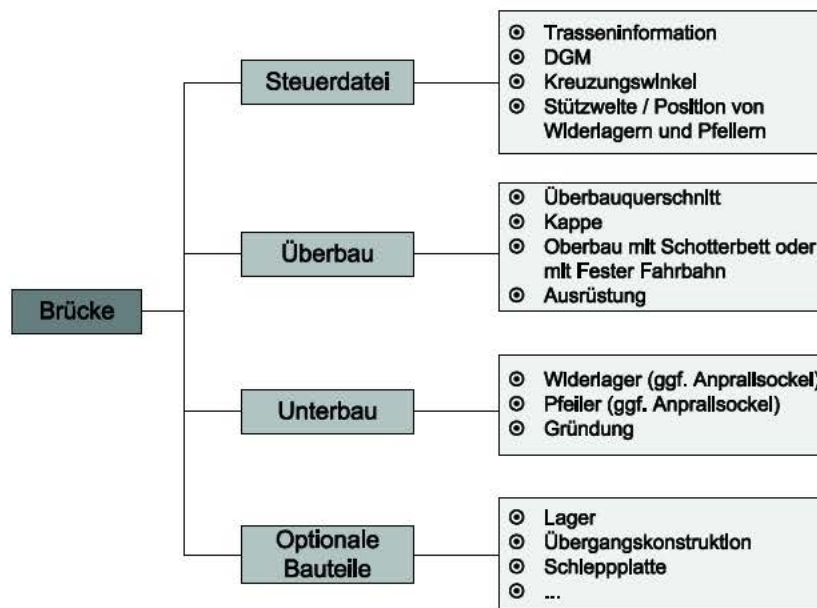


Abbildung 16: Bauwerkstruktur einer Brücke nach (Wang, 2012)

Bauteile werden in NX in den entsprechenden Komponenten konstruiert. Sie sind dreidimensional und beinhalten alle Geometrieinformationen. Für jedes Bauteil wird eine eigene Komponente angelegt. Alle Bauteile sind direkt von Teilen der Steuerdatei abhängig, sodass sich Änderungen und Anpassungen immer auf das Gesamtbauwerk auswirken und sie auf Konsistenz geprüft werden (Hochmuth, 2012).

Im Teile-Navigator sind die Modellieroperationen der Komponenten chronologisch aufgelistet, wodurch eine Modellhistorie angelegt wird. Zur besseren Übersicht können den Elementen und Funktionen Namen zugewiesen werden. Über den Teile-Navigator können Elemente ausgewählt, bearbeitet, verschoben, ausgeblendet oder deaktiviert werden. Durch die chronologische Auflistung bleibt der Modellierungsvorgang stets nachvollziehbar (Hochmuth, 2012).

- Materialbibliothek/Systemmaterialien
- Materialien im Teil



Abbildung 17: Ressourcenliste mit geöffnetem Teile-Navigator, in Anlehnung an (Hochmuth, 2012)

6.1.2 Vorgehensweise bei parametrischer Brückenmodellierung

Als erstes werden die Baugruppendatei, sowie die Steuerdatei angelegt. Mittels der ProVI-Schnittstelle wird die Trassierung in die Steuerdatei importiert. Des Weiteren enthält die Steuerdatei die Bezugsebenen der Komponenten, die später für die Erstellung von assoziativen Bauteilskizzen notwendig sind. Mögliche Bezugsebenen bei Brücken sind z.B. die Achsen der Widerlager und Pfeiler. Es gibt mehrere Möglichkeiten eine Bezugsebene zu positionieren, wie z.B. auf einer Ebene des Koordinatensystems, an einem Pfad ausgerichtet oder auf einem Objekt. Bezugsebenen können fest oder assoziativ sein. Damit die Komponenten später auf ihre Bezugsebenen zugreifen können und sich Änderungen assoziativ auf das gesamte 3D-Modell auswirken gibt es den WAVE-Geometrie-Linker Befehl. Die ausgewählten Elemente können dann zur Modellierung der Bauteile genutzt werden (Weinholzer, 2014).

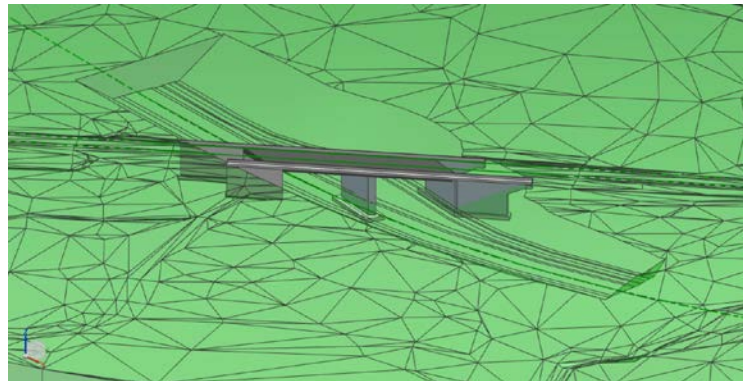


Abbildung 18: Rahmenbrücke in Siemens NX

Für die Modellierung eines Bauteils wird die Komponente als „aktives Teil“ ausgewählt. Zunächst wird für den Querschnitt des Bauteils eine Skizze erstellt, welche sich auf eine Bezugsebene bezieht. Die Bezugsebene kann entweder als übergeordnete Ebene über den WAVE-Geometrie-Linker in die Komponente eingefügt werden oder innerhalb der Komponente erstellt werden. Es ist sinnvoll für jede Skizze eine eigene Bezugsebene zu definieren. Skizzen können manuell erstellt, als DWG-Datei importiert oder über die Wiederverwendungsbibliothek aufgerufen werden. Die Skizzen werden automatisch vermaßt. Es ist oftmals sinnvoll eine Leitskizze zu definieren, deren Parameter eine bestimmte Länge zugeordnet wird. Mit dieser Leitskizze, z.B. die Fahrbahnbreite, können andere Längen oder weitere Skizzen damit parametrisch verknüpft werden. Änderungen der Leitskizze wirken sich dann automatisch auf alle gekoppelten Werte aus. Viele Querschnitte wiederholen sich im Ingenieurbau in abgewandelter Form. Werden bereits erstellte Querschnitte parametrisiert in der Wiederverwendungsbibliothek abgelegt, können sie durch Aufrufen und schnelles Anpassen der Parameter in neuen Projekten verwendet werden. Als weitere Funktion können bestimmte Freiheitsgrade blockiert werden (Seiten sind parallel, lotrecht, etc.) um bestimmte geometrische Eigenschaften zwingend beizubehalten und sie somit nicht versehentlich geändert werden.

Nach der Erstellung der Skizze kann diese anschließend zu einem Volumenkörper extrudiert werden. Dabei wird die Skizze entlang einer Leitlinie geführt. Sind auf der Leitlinie mehrere Skizzen platziert, um z.B. eine Querschnittsänderung über die Länge darzustellen, erstellt NX einen sich über die Länge ändernden Körper. Die Volumenkörper können über die booleschen Operationen „Vereinigen“, „Subtrahieren“, und „Schneiden“ oder weiteren Befehlen wie „Körper trimmen“ bearbeitet werden (Hochmuth, 2012).

6.2 Programmübergreifende Dateiformate in Siemens NX

Neben dem proprietären Standardspeicherformat Unigraphics Part File (.prt) bietet Siemens NX eine breite Auswahlmöglichkeit Projekte zu speichern und zu exportieren. Für viele Aufgabenbereiche außerhalb der Modellkonstruktion sind gar nicht alle Informationen der Originaldaten erforderlich. Durch den Export in andere Dateiformate können so gezielt nur die Daten gespeichert werden, die für den jeweiligen Aufgabenbereich notwendig sind. Im Rahmen dieser Arbeit geht es primär um den Datenaustausch geometrischer und materieller Informationen. Für diesen Zweck eignen sich vor allem Visualisierungsformate. Bei Visualisierungsformaten handelt es sich idealerweise um neutrale Formate, die geometrische und ggfs. geometriebezogene Modelldaten enthalten. Die wichtigsten Visualisierungsformate, die Siemens NX anbietet, sind IGES, STEP, VRML und JT. Diese werden im Folgenden kurz vorgestellt.

IGES (Initial Graphics Exchange Specification): IGES gilt als das älteste Visualisierungsformat und wurde ursprünglich nur für 2D-Zeicheninformationen konzipiert. Die aktuelle Version 5.3 wurde 1996 Standard der ANSI und kann inzwischen auch 3D Geometrien übertragen. Da das Format als überholt gilt, wurde die Weiterentwicklung eingestellt.

STEP (Standard for the Exchange of Product Model Data): verschiedene Applikationsprotokolle (AP) des STEP Standards sollen die verschiedenen Anwendungsbereiche abdecken. Siemens NX bietet STEP203 und STEP214 an. STEP203 wird für den allgemeinen Maschinenbau verwendet, STEP214 ist für die Automobilindustrie zugeschnitten. Es können Produktinformationen über den gesamten Lebenszyklus abgebildet werden (Friedewald, et al., 2011). STEP ist, wie das IFC Format, in der EXPRESS Sprache geschrieben.

VRML (Virtual Reality Modeling Language): VRML wurde ursprünglich für 3D Inhalte im Internet entwickelt. Es ist jedoch nicht speziell für den Ingenieurbereich ausgerichtet. Viele 3D Programme ermöglichen den Im- und Export von VRML Dateien, wodurch es sich auch zu einem Austauschformat entwickelt hat (Friedewald, et al., 2011).

JT (Jupiter Tessellation): JT gibt es seit Ende der 90er Jahre. Es wurde als Visualisierungsformat für den Ingenieurbereich entwickelt. Das Format dient zur Beschreibung von 3D-Daten, wobei auch Objekt und Metadaten, wie Farbe oder Textur, unterstützt werden. Die Volumenkörper können, auf Grund der „Container-Struktur“ des Formats, als tesselierte Dreieckselemente oder als exakte B-rep Geometrie dargestellt werden. Das JT Format hat sich auf Grund seiner Leichtigkeit schnell durchgesetzt. Zusätzlich kann über das Level of detail (LOD) die Geometrie-Präzision bestimmt werden und somit auch die Datenmenge. JT ist seit 2012 ISO Standard.

Da die Weiterentwicklung des IGES Formats eingestellt wurde ist eine Implementierung der Schnittstelle zwischen Siemens NX und SOFiSTiK darüber nicht sinnvoll. VRML eignet sich nur bedingt für den Ingenieurbereich, da weder Baugruppenstruktur noch Metadaten oder Textur abgespeichert werden. STEP ist sehr umfangreich und hält damit viele Informationen vor. Dadurch ist der Aufbau jedoch auch sehr komplex. NX bietet beim STEP Format nur die Maschinenbau relevanten APs 203 und 214 an. Des Weiteren ist es nachteilig, dass die Spezifikation zwar offen aber kostenpflichtig ist.

Im Rahmen dieser Arbeit wird die Schnittstelle zwischen Siemens NX und SOFiSTiK über das JT Format implementiert. Dank des öffentlichen und kostenfreien Zugangs des Formats kann die Schnittstelle auch in Zukunft angepasst und gewartet werden. Trotz der Kompaktheit ist JT ein besonders inhaltreiches Format und bietet die Option Texturen und Farbinformationen abzuspeichern. Durch die Möglichkeit des Facettierens der Volumenkörper in primitive Dreiecksflächen ist es äußerst handhabbar. Über das level of detail kann man die Feinheit des Modells an seine Anforderungen anpassen. Weitere Repräsentationsformen sind JT B-rep und XT B-rep. Beim JT B-rep werden die Daten durch Algorithmen komprimiert und als B-rep Modell gespeichert. Das XT B-rep ist die Vorgängerversion, die auf dem früheren Boundaray Representation basiert. Als weitere Option kann bestimmt werden, ob die Produktknoten eines Bauteils in einer individuellen JT-Datei gespeichert werden (Assembly-JT) oder ob die gesamte Produktstruktur in einer einzigen JT-Datei abgelegt wird. JT hat sich in den letzten Jahren stark verbreitet, sodass neben NX noch weitere Softwarehersteller, wie Dassault Systemes CATIA oder Autodesk Inventor, dieses Format unterstützen (Sendler, 2009).

6.3 Maßnahmen für den Modell-Export in FEM-Programme

Ziel des digitalen Gebäudemodells ist es, das geplante Bauwerk realitätsnah abzubilden, während für die Bemessung des Bauwerks normalerweise vereinfachte statische Systeme herangezogen werden. Dieser Widerspruch führt dazu, dass der Ingenieur erst festlegen muss, wie das digitale Gebäudemodell in Statik-Programme exportiert werden soll. Wird der Import in ein FEM-Programm als Black-Box-Methode, ohne vorherige Anpassungen, vorgenommen, kann es zu fatalen Fehlern bei der Bemessung kommen. Dies ist besonders bei komplexeren Bauvorhaben kritisch, da hier der Anwender leichter dazu tendiert die Ergebnisse der Bemessung zu glauben, da er sie nicht mehr ohne weiteres überprüfen kann. Es kann daher notwendig sein, Anpassungen des Modells vorzunehmen.

Viele Anpassungen des digitalen Gebäudemodells können ohne größeren Aufwand ausgeführt werden. Weder Steuerdatei, noch Oberbau, Kappen, Ausrüstung oder sonstige optionale

Bauteile sind für die Bemessung eines Bauwerks relevant. Diese Komponenten können einfach durch Entfernen des Häkchens ausgeschaltet werden. Auch Anpassungen innerhalb einer Komponente können zum Teil durch Entfernen des Häkchens im Teile-Navigator angepasst werden. Das Entfernen der Häkchen innerhalb Komponenten hat Auswirkungen auf die Bauteil-Geometrie. Dies sollte also nur angewendet werden, wenn die Geometrie bewusst anpasst werden soll, wie z.B. durch das Ausschalten des Befehls „Kantenverrundung“.

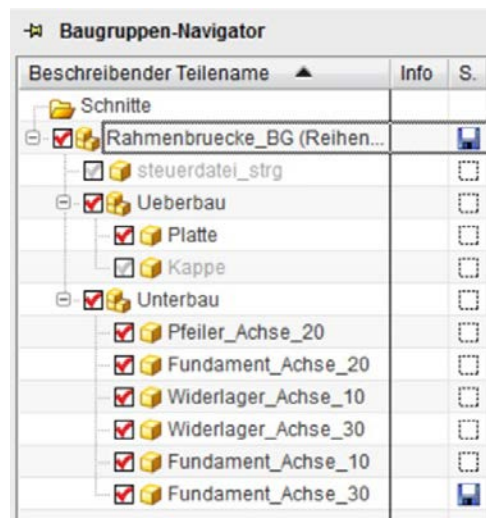


Abbildung 19: Auswahl der relevanten Komponenten für FEM-Export

Objekte und Modellieroperationen, wie Bezugsebenen, Skizzen oder verbundene Flächen, sind für die Modellierung der Komponenten zwingend notwendig. Ihre Häkchen müssen gesetzt bleiben. Sie können aber über den Befehl „ausblenden“ unsichtbar gemacht werden. Deaktivierte und ausgeblendete Elemente sind in der Ressourcenliste hellgrau aufgeführt.

Um den Komponenten die jeweiligen Betongüten zuzuordnen und diese automatisiert in SOFiSTiK zu übergeben wurde einmalig die Komponente „Beton“ angelegt. In dieser wurden die verschiedenen herkömmlichen Betongüten definiert. In „Materialien im Teil“ in der Ressourcenliste kann die entsprechende Betongüte ausgewählt und kopiert werden. Anschließend wird in die passende Komponente gewechselt und um wiederum in „Materialien im Teil“ das Material einzufügen. Anschließend kann das Material einfach durch die Drag & Drop Funktion auf das Bauteil abgelegt werden. Die entsprechenden Betongüten der Bauteile können dem 0 entnommen werden.

Skizzen und Bauteile werden in NX anhand lokaler Koordinatensysteme ausgerichtet. Zusätzlich gibt es für jedes Element oder Komponente ein Arbeitskoordinatensystem WCS (Work Coordinate System), welches beliebig verschoben und gedreht werden kann. Darüber hinaus verfügt NX über ein übergeordnetes, absolutes Koordinatensystem, das im Raum

fixiert ist (Klette, et al., 2011). Anders als im Bauwesen ist die Z-Achse in der Maschinenbaubranche dabei nach oben definiert. Für den Datenexport in bauspezifische Software ist daher darauf zu achten, dass das Bauwerk am absoluten Bauwerk ausgerichtet wird und die Z-Achse nach unten zeigt.

Wie bereits in Kapitel 2 erwähnt, besteht das Rahmenbauwerk aus zwei Bauabschnitten, die in Längsachse über ein innenliegendes Fugenband getrennt sind. Da beide Bauabschnitte etwa die gleichen Abmessungen aufweisen, kann die Bemessung stellvertretend nur an einem Bauabschnitt erfolgen. Es wird daher nur der südliche Bauabschnitt (BA1) betrachtet.

Es ist die Aufgabe des Ingenieurs zu entscheiden, welche Anpassungen notwendig sind. Es ist jedoch nicht Sinn des BIM-Gedankens, das komplette digitale Gebäudemodell an die jeweiligen Anwendungen anzugleichen. Es ist daher je nach Anwendungsfall zu entscheiden, ob durch den Mehraufwand ein Export des erstellten Modells noch zielführend ist. In den nachfolgenden Unterkapiteln werden die vorgenommenen Anpassungen erläutert.

6.3.1 Export als Volumenmodell

Modelle bestehend aus Volumenelementen sind am vielseitigsten einsetzbar. Jedes beliebige System lässt sich damit modellieren. Annahmen wie die Bernoulli-Hypothese können nicht angewendet werden. Volumenmodelle bestehen aus Elementen wie Hexaedern oder Tetraedern. Die Wahl der Volumenelemente hat Auswirkungen auf die Steifigkeit und beeinflusst somit die Ergebnisse.

Jeder Knotenpunkt besitzt drei Verschiebungsfreiheitsgrade u , v , w in x -, y -, und z - Richtung. Dies führt zu einer großen Anzahl von Freiheitsgraden und damit verbunden zu einem hohen Rechenaufwand. Auch die Interpretation der Ergebnisse ist aufwändiger und bedarf einer guten Kenntnis über das Verhalten und Beschränkungen von Volumenelementen. Volumenmodelle werden daher im konstruktiven Ingenieurbau bisher nur in Sonderfällen angewendet (Werkle, 2008).

Das exportierte Volumenmodell der EÜ Ventschow sollte eine vereinfachte Geometrie aufweisen, um den Rechenaufwand in Grenzen zu halten und um die Ergebnisse noch nachvollziehbar überprüfen zu können. Nachfolgend werden die Anpassungen des digitalen Gebäudemodells aufgezählt.

Wie in 6.3 beschrieben werden die Steuerdatei, sowie die Komponente „Kappen“ durch Entfernen des Häkchens ausgeschaltet. Des Weiteren können in den Komponenten die Operatoren „Kantenverrundung“ in den Widerlagern, wie auch „Fase“ im Mittelpfeilerfundament ausgeschaltet werden. Wie in Abbildung 20 dargestellt, schließt der obere Teilkörper der

Fundamente an die Widerlager und den Pfeiler an. Da der obere Teilkörper für die Bemessung nicht relevant ist wurde er ebenfalls entfernt. Weitere Anpassungen wurden nicht vorgenommen. Abschließend wurden den Komponenten noch, wie in 6.3 beschrieben, die jeweiligen Betongüten zugewiesen.

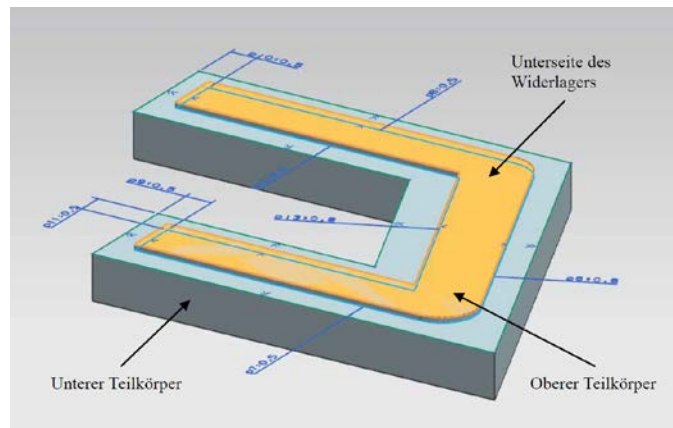


Abbildung 20: oberer und unterer Teilkörper des Fundaments (Wang, 2012)

6.3.2 Export als Flächenmodell

Das Flächentragwerk ist das wohl am häufigsten angewendete FE-Modell in der Baupraxis. Es kommt überall dort zum Einsatz, wo Länge und Breite dominant im Vergleich zur Dicke sind. Flächenelemente besitzen eine hohe Genauigkeit bei vergleichsweise geringer Rechenleistung. Dennoch sei erwähnt, dass es sich um Näherungslösungen handelt, die stets überprüft werden müssen, um fatale Fehler auszuschließen. Das Tragwerk wird in der Regel in Vierecks- oder Dreieckselemente unterteilt. Da Dreieckselemente steifer reagieren als Viereckselemente, werden meist Viereckselemente bevorzugt eingesetzt. Jedes Element hat zwei Verschiebungsfreiheitsgrade u und v in x - und y - Richtung (Werkle, 2008).

Im digitalen Gebäudemodell ist das Flächentragwerk noch nicht hinterlegt und muss manuell hinzu konstruiert werden. Dafür können die Funktionen „Mittelfläche durch Flächenpaare“ und „Anwenderdefinierte Mittelfläche“ genutzt werden. Das Prinzip ist bei allen Komponenten gleich: Mittelfläche konstruieren, Volumenkörper ausblenden, Mittelflächen verbinden und Material zuordnen.



Abbildung 21: erzeugtes Flächenmodell in Siemens NX

Die Mittelfläche der Fundamente ergibt sich aus der Ober- und Unterseite des Bauteils. Dazu wird der Befehl „Mittelfläche durch Flächenpaare“ aufgerufen. Als erstes wird der entsprechende Volumenkörper ausgewählt. Als gewählte Strategie empfiehlt sich die „manuelle“ Strategie, da so die Flächenpaare selbst bestimmt werden können, anhand denen die Mittelfläche erzeugt werden soll. Anschließend werden Ober- und Unterseite als Seitenflächen ausgewählt. Daraufhin wird die Mittelfläche erzeugt und der Volumenkörper kann ausgeblendet werden.

Im Ingenieurbau wird die Brückenplatte als gekoppelte Flächen modelliert, um die Lasteinleitung statisch korrekt darzustellen. Die gekoppelten Flächen können nicht über Flächenpaare erzeugt werden und müssen daher selber erstellt werden. Im ersten Schritt wird die Skizze des Plattenquerschnittes kopiert. Die Kopie liegt direkt auf der originalen Skizze. Anschließend können die Linien der Mittelfläche in die kopierte Skizze eingezeichnet werden und die übrigen Querschnittslinien können entfernt werden. Als nächstes wird die kopierte Skizze entlang des Pfades „Gleis“ extrudiert. Der extrudierte Körper wird mit dem Befehl „Körper trimmen“ an den Bezugsebenen „Achse 10“ und „Achse 30“ getrimmt und über den Befehl „Anwenderdefinierte Mittelfläche“ als Mittelfläche festgelegt.

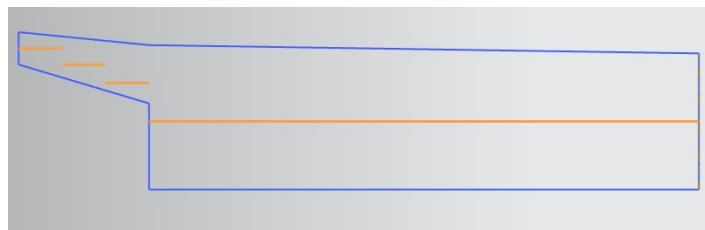


Abbildung 22: halber Brückenquerschnitt (blau) mit Mittelfläche (gelb)

Als nächstes wird vom Mittelpfeiler die Mittelfläche konstruiert. Analog zu den Fundamenten wird mit dem Befehl „Mittelfläche durch Flächenpaare“ und der manuellen Auswahl der beiden Seitenflächen die Mittelfläche erzeugt und der Volumenkörper ausgeblendet. Wie in

Abbildung 23 dargestellt berühren sich die Mittelflächen der Bauteile noch nicht. Um die Flächen zu verbinden wird mit dem WAVE Geometrie-Linker-Befehl die Mittelfläche des Fundaments und der Platte als verbundene Fläche importiert. Anschließend kann mit dem Befehl „Trimmen/Verlängern“ die obere und untere Kante des Mittelpfeilers bis an die angrenzenden Flächen verlängert werden.

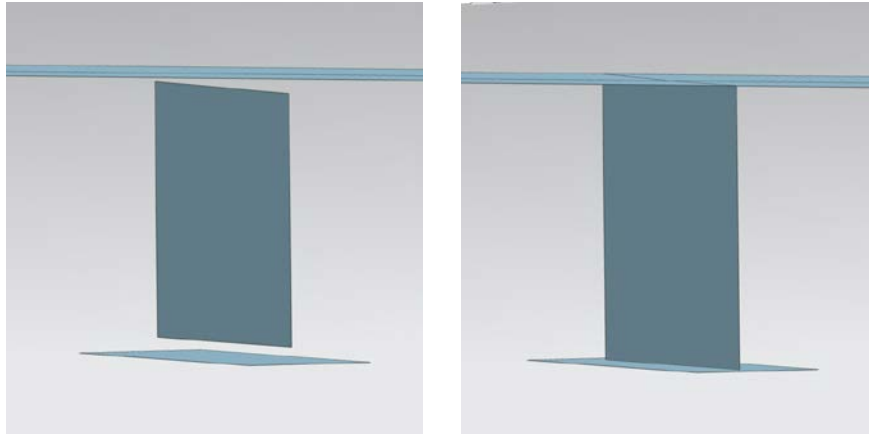


Abbildung 23: nicht verbundene Mittelflächen (links) und verbundene Mittelflächen (rechts)

Analog zum Mittelpfeiler können nun die Widerlager bearbeitet werden. Die Mittelflächen werden separat für Kammerwand und Flügelwände konstruiert. Dies hat den Vorteil, dass sie später auch separat betrachtet werden können und somit auch verschiedene Bauteildicken und Betongüten erhalten können. Es werden mit dem Befehl „Mittelfläche aus Flächenpaaren“ nur die Seitenflächen der Kammerwand oder der Flügel ausgewählt und die Mittelfläche erzeugt. Bei den Flügelwänden werden nur die Seitenwände mit der konstanten Bauteildicke ausgewählt, da NX die Mittelfläche im Bereich von Vouten mittelt und somit ein falscher geometrischer Körper entstehen würde. Anschließend wird das Ganze für die anderen Seitenwände wiederholt. Die Mittelflächen werden wieder zu den angrenzenden Flächen der Fundamente und Platte verlängert. Als letztes können analog zu 5.3.1 den Mittelflächen Betongüten übergeben werden.

7. Konventionelle Modelleingabe in SOFiSTiK

Die SOFiSTiK AG ist ein Softwarehersteller für die Berechnung, Bemessung und Konstruktion von Bauprojekten im konstruktiven Ingenieurbau. Die Rahmenbrücke des BA 1 der EÜ Ventschow wurde als Flächentragwerk mit Hilfe des Texteditors TEDDY eingegeben und beispielhaft belastet. Die Ergebnisse der Schnittgrößen des konventionellen Modells werden in Kapitel 9.2 mit denen der Schnittstellenmodelle verglichen und bewertet. Als Norm wurde für die Eisenbahnüberführung der aktuelle Eurocode 2 angegeben. Die gewählte Belastung wurde aus der Genehmigungsstatik übernommen und entspricht noch dem DIN-Fachbericht 101.

7.1 Softwareübersicht

Das besondere Merkmal an SOFiSTiK ist der modulare Aufbau und die offene Schnittstelle. Der Rechenkern besteht aus mehreren Batch-Programmen. Im Mittelpunkt steht eine sehr effiziente Datenbasis über die der Datenaustausch der vielen Programme abläuft. Die Programme können über die TEDDY Eingabedatei oder einem grafischen Interface angesprochen werden.

Der SOFiSTiK Structural Desktop (SSD) bietet eine einheitliche Benutzeroberfläche für die Programme von SOFiSTiK. Das statische System kann mit Hilfe von SOFIPLUS(-X) grafisch oder mit dem TEDDY Editor parametrisch eingegeben werden. SOFIPLUS ist ein CAD Programm zur grafischen Systemeingabe und kann als Aufsatz für AutoCAD oder auch als stand alone Produkt (SOFIPLUS(-X)) genutzt werden. In dem Programm werden Strukturen und Belastungen definiert. Die Daten werden in der Datenbasis gespeichert und an die SOFiSTiK Berechnungsprogramme übergeben.

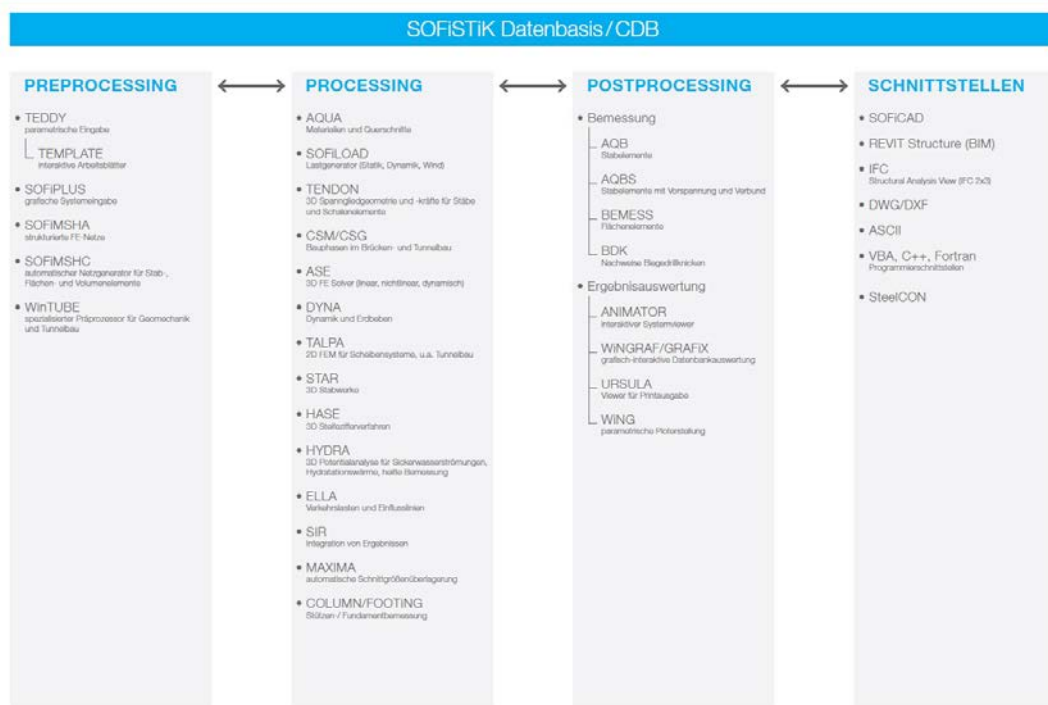


Abbildung 24: SOFiSTiK Structural Desktop (SOFiSTiK, 2014)

Die TEDDY Eingabedatei wird in der programmeigenen CADINP Sprache verfasst und hat den Vorteil der flexiblen Gestaltung des Inputs. Der Nutzer hat so die Möglichkeit FE-Systeme parametrisch zu generieren, sowie Berechnung und Ausgabe gezielt zu steuern. Mit Hilfe von Kommentaren bleibt die Datei stets übersichtlich und nachvollziehbar. Die Eingabe wird in Zeilen formuliert. Dabei werden gleichartige oder zusammenhängende Daten zu Sätzen zusammengefasst. Die einzelnen Werte eines Satzes können Zahlen oder Literale sein und werden durch Trennzeichen getrennt (SOFiSTiK, 2014). In 7.2 wird der Aufbau einer TEDDY Eingabe anhand des Anwenderbeispiels „EÜ Ventschow“ dargestellt.

7.2 Modelleingabe im TEDDY-Editor

Die TEDDY Eingabedatei ist modular aufgebaut, d.h. es werden hintereinander Batch-Programme (Stapelprogramme) geschrieben. In den verschiedenen Modulen werden Materialeigenschaften, geometrische Anordnung, Lastfälle, etc. definiert. Die Programme können einzeln aufgerufen werden, was den Vorteil schafft, dass bei Änderungen nur das entsprechende Modul neu berechnet werden muss und nicht mehr das gesamte System. Die Ergebnisse werden in der CDBASE Datenbank abgespeichert, wo sie jederzeit von anderen Modulen aufgegriffen werden können.

Jedes Modul beginnt mit dem Wort *prog* gefolgt vom Programmnamen und schließt mit *ende*. Innerhalb des Moduls erfolgt die Eingabe dann in Zeilen, die mit bestimmten Schlüsselwörtern beginnen. Im Handbuch (F1-Taste) sind alle Schlüsselwörter und Eingaben dokumentiert.

7.2.1 AQUA – Materialdefinition

Als erstes werden im Programm AQUA die verwendeten Materialien definiert. Die Eingabezeile *Kopf* dient zur besseren Übersicht in der Ausgabedatei. Mit dem Befehl *Echo* wird der Ausgabeumfang der Ausgabedatei festgelegt. Unter *Norm* wird die Berechnungsgrundlage bestimmt. Jedes Material bekommt eine eigene Materialnummer zugewiesen. Die Überbauplatte hat eine Betonfestigkeitsklasse von C35/45, Widerlager und Pfeiler von C30/37 und die Fundamente von C25/30. Mit Hilfe von Kommentaren bleibt die Eingabedatei stets verständlich und übersichtlich. Kommentare beginnen mit dem \$ Zeichen und sind im TEDDY Editor grün hinterlegt.

```
+prog aqua urs:1

KOPF EÜ Ventschow
ECHO MAT,QUER VOLL
NORM DIN EN1992-2004 cat D

$ -----
$ - MATERIALIEN - - - - -

BETO 1 C 35          $ BETON Überbau
BETO 2 C 30          $ BETON aufgehende Unterbauten
BETO 3 C 25          $ BETON Fundamente

Stah 4 B 500B

Ende
```

Abbildung 25: AQUA-Modul im TEDDY Editor

7.2.2 SOFIMSHC – Strukturelemente

Das Modul SOFIMSHC dient zur Generierung und Verarbeitung geometrischer Modelle. Das FE-Netz wird automatisch erzeugt, sodass nur noch die Randbedingungen als Strukturelemente eingegeben werden. Das gesamte Brückenbauwerk wird durch Flächenelemente modelliert. Hierzu wird die Geometrie der Mittelflächen definiert, welcher anschließend eine Bauteildicke und das entsprechende Material übergeben wird.

```

+prog sofimshc urs:2 $ Definition Strukturelemente

#include Variablen

Kopf System
Syst 3D gdiv 10000
STEU mesh 2 +64
STEU HMIN 0.5

$ -----

$Knoten Fundament Achse 10

SPT NR 100 X 0.000 Y 3.280 Z 7.200
      101 0.000 6.280 7.200
      102 5.490 6.280 7.200
      103 7.320 0.000 7.200
      104 3.480 0.000 7.200
      105 2.520 3.280 7.200

$Strukturlinien Fundament Achse 10

SLN NR 100 NPA 100 NPE 101 SDIV #d1
      101 101 102 SDIV #d1
      102 102 103 SDIV #d1
      103 103 104 SDIV #d1
      104 104 105 SDIV #d1
      105 105 100 SDIV #d1

$Strukturfläche Fundament Achse 10

SAR NR 1 GRP 10 T 950 MNR 3 MBW 4 CT 10000 CB 8000 MCTL +1
SARB out 100,101,102,103,104,105

```

Abbildung 26: Systemgenerierung unter SOFIMSHC

Um das Flächentragwerk zu generieren müssen zuerst die Knoten (Strukturpunkte *SPT*) festgelegt werden, die später durch Strukturlinien (*SLN*) verbunden werden. Jeder Strukturpunkt bekommt eine Punktnummer zugewiesen. Die Strukturlinien beziehen sich jeweils auf einen Anfangs- und Endknoten (*NPA* und *NPE*) und bekommen ebenfalls eine Liniennummer zugewiesen. Zur besseren Übersicht hat jede Strukturlinie i.d.R. die gleiche Nummer wie der Anfangsknoten. Anschließend wird aus den Strukturlinien eine Strukturfläche (*SAR*) gebildet. Der Strukturfläche werden zusätzlich eine Bauteildicke *T*, sowie das zugehörige Material zugeordnet. Die Wahl des Betons bzw. des Betonstahls wird über *MNR* bzw. *MBW* festgelegt, wobei die in AQUA festgelegte Materialnummer angegeben wird. Bei den Fundamenten werden zusätzlich noch die Module für Senk- bzw. Querbettung der Fläche angegeben (*CT* bzw. *CB*). Der Satz *SARB out* definiert welche Strukturlinien die äußere Randkante der Strukturfläche bilden.

Unter SOFIMSHC wird das FE-Netz automatisch erzeugt. Dennoch kann und sollte man Einfluss auf die gewählten Finite-Elemente nehmen. Unter *STEU* kann mit dem Literal *+64* festgelegt werden, dass nur Viereckselemente erzeugt werden. *HMIN* steuert die globale Netzdichte des Systems. In der Eingabezeile *SAR* kann durch *MCTL +1* bestimmt werden, dass,

wenn möglich, ein regelmäßiges Netz erzeugt werden soll. Als weiteren Operator kann man durch *SDIV* die Unterteilungen der Strukturlinien festlegen. Wird hier kein Wert explizit festgelegt, so wird die globale Netzdichte *HMIN* verwendet. Es macht jedoch Sinn das FE-Netz mit *SDIV* zu verfeinern. Da der Wert iterativ optimiert werden kann, wurde er als Variable deklariert. So muss nicht jede Eingabezeile erneut angepasst werden, sondern es kann einfach der Wert der Variablen erneut bestimmt werden.

Variablen werden meist außerhalb der Module in Unterprogrammen definiert. Unterprogramme für Variablen beginnen mit *#define* gefolgt von dem Namen des Unterprogramms (z.B. *Variablen*) und enden mit *#enddef*. Nach Angabe von *let#* folgt direkt der Name der Variablen. Die Variable kann dann über ihren Namen mit vorangestellten #-Zeichen jederzeit aufgerufen werden. Durch den Befehl *#include* mit dem Namen des Unterprogramms können die Variablen im jeweiligen Modul aufgerufen werden.

```
#define Variablen  
  
let#d1 0.3 $ SDIV Fundamente  
let#d2 0.4 $ SDIV Kammerwand  
let#d3 0.45 $ SDIV Flügel  
let#d4 0.5 $ SDIV Flügel Voute  
let#d5 0.5 $ SDIV Fahrbahn  
let#d6 0.3 $ SDIV Kragarm  
let#d7 0.55 $ SDIV Pfeiler  
  
#enddef
```

Abbildung 27: Variablenzuweisung

Die Schnittgrößen der Biege- und Schubschnitte sind für die Bemessung des Tragwerks maßgeblich. Für einen klaren Verlauf der Schnittgrößen ist es wichtig, dass sich die FE-Elemente geordnet entlang dieser Schnittlinien erstrecken. Da sich die automatische Vernetzung in SOFIMSHC immer entlang der Strukturelemente orientiert, werden weitere Strukturlinien entlang der Biege- und Schubschnitte definiert, sodass sich das FE-Netz daran ausrichtet.

Um die Lasteinleitung der Kragarme in die Fahrbahnplatte statisch korrekt abzubilden, wird die Mittelfläche der Kragarme in drei gekoppelte Flächen aufgeteilt, die wiederum mit der Fahrbahnplatte gekoppelt sind. Die Flächen der Kragarme sind horizontal und stehen damit

senkrecht zur Lastrichtung. Es werden jeweils die übereinander liegenden Kanten der Teilflächen gekoppelt.

```

$Strukturlinie Kragarm

$Teilfläche 1
SLN  NR 420   NPA 140   NPE 141   SDIV #d6
      421   NPA 141   NPE 341   SDIV #d6
      422   NPA 341   NPE 340   SDIV #d6
      423   NPA 340   NPE 240   SDIV #d6 ; SLNS FIX F REFT >SLN REF 411
      424   NPA 240   NPE 140   SDIV #d6 ; SLNS FIX F REFT >SLN REF 401

$Teilfläche 2
SLN  NR 430   NPA 142   NPE 143   SDIV #d6
      431   NPA 143   NPE 343   SDIV #d6
      432   NPA 343   NPE 342   SDIV #d6
      433   NPA 342   NPE 142   SDIV #d6 ; SLNS FIX F REFT >SLN REF 421

$Teilfläche 3
SLN  NR 440   NPA 144   NPE 145   SDIV #d6
      441   NPA 145   NPE 345   SDIV #d6
      442   NPA 345   NPE 344   SDIV #d6
      443   NPA 344   NPE 144   SDIV #d6 ; SLNS FIX F REFT >SLN REF 431

```

Abbildung 28: Kopplung der Teilflächen

Mit der Eingabe *SLNS* können Strukturlinien kinematisch gekoppelt werden. Dazu wird mit *FIX F* die Koppelbedingung Festhaltung gegen Verschiebung definiert und mit *REFT >* die dazugehörige Referenzlinie angegeben.

7.2.3 SOFILOAD – Lastfalldefinition

Im Rahmen dieser Arbeit wurden verschiedene Lastfälle auf das Bauwerk angesetzt. Es handelt sich dabei nur um exemplarische Lastfälle, um die Schnittgrößen der verschiedenen Modelle in Kapitel 9.2 vergleichen zu können. Eine komplette statische Bemessung ist daher nicht notwendig. Es wurde der Fall betrachtet, dass sich das System in Endlage befindet und horizontal gebettet auf Höhe der Fundamente Achse 10 bis 30 gehalten wird. Als exemplarische Lastfälle wurde das Eigengewicht als ständige Last, das Lastmodell 71 als veränderliche Last und Temperatur als Zwang angesetzt. Die Werte stammen aus der Genehmigungsstatik, welche nach dem DIN-Fachbericht 101 bemessen wurde. Sie entsprechen in etwa den Werten des Eurocodes 2, welcher im Weiteren für das Anwenderbeispiel als Bemessungsstandard angesetzt wird.

Im Lastfall 1 wurde die Last aus dem Eigengewicht angesetzt. Diese ergibt sich aus der Betonwichte $\gamma_c = 25 \text{ kN/m}^3$ und wird vom Bemessungsprogramm automatisch generiert.

Das Lastmodell 71, dargestellt in Abbildung 29, stellt den statischen Anteil der Einwirkungen aus dem Regelverkehr dar und wirkt als Vertikallast auf das Gleis.

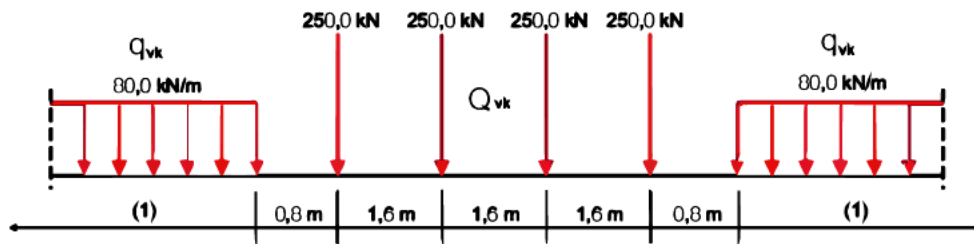


Abbildung 29: Lastmodell 71 für Eisenbahnbrücken (DIN e.V, 2013)

Die Einzel- bzw. Streckenlasten wurden zu konstanten Flächenlasten umgerechnet. Sie ergeben sich aus:

$$b = 3,0 \text{ m} + 2 * \frac{1}{2} * 1,05 \text{ m} \approx 4,0 \text{ m}$$

Achslast: $Q'_{v,k} = 1,0 * 1,20 * \frac{4 * 250 \text{ kN}}{6,4 \text{ m} * 4,0 \text{ m}} = 46,9 \text{ kN/m}^2$

Streckenlast: $q'_{v,k} = 1,0 * 1,2 * \frac{80 \text{ kN}}{4,0 \text{ m}} = 24,0 \text{ kN/m}^2$

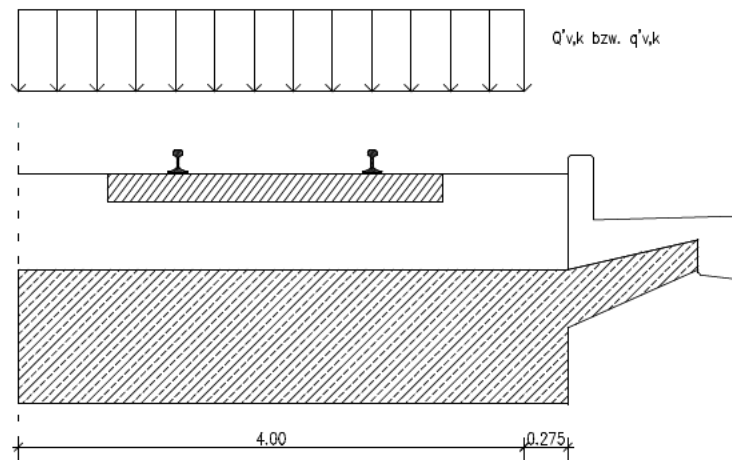


Abbildung 30: Flächenlast $Q'_{v,k}$ bzw. $q'_{v,k}$

Zur Ermittlung der ungünstigsten Stellung werden die Achs- bzw. Streckenlasten in unterschiedlichen Lastfällen generiert. Die Streckenlast $q'_{v,k}$ wird über den Bereich der Achslasten $Q'_{v,k}$ angesetzt. Die eigentliche Achslast ergibt sich als Differenz aus Achs- und Streckenlast.

$$q'_{v,k} = 24,0 \text{ kN/m}^2$$

$$\Delta Q'_{v,k} = 46,9 \text{ kN/m}^2 - 24,0 \text{ kN/m}^2 = 22,9 \text{ kN/m}^2$$

Die Streckenlast $q'_{v,k}$ wird feldweise angesetzt (LF 11, LF 12), die Achslasten $Q'_{v,k}$ wird als wandernde Flächenlast im Intervall von $x = 3,5 \text{ m}$ verschoben (LF 21- LF 29).

Die Lastfälle aus Temperatureinwirkung ergeben sich aus den konstanten und veränderlichen Temperaturanteilen nach DIN Fachbericht 101. Der konstante Temperaturanteil wirkt auf die Fahrbahnplatte, sowie auf die aufgehenden Bauteile in Achse 10 und 30. Im Rahmen dieser Arbeit wird er, wie der veränderliche Anteil, nur auf die Fahrbahnplatte angesetzt.

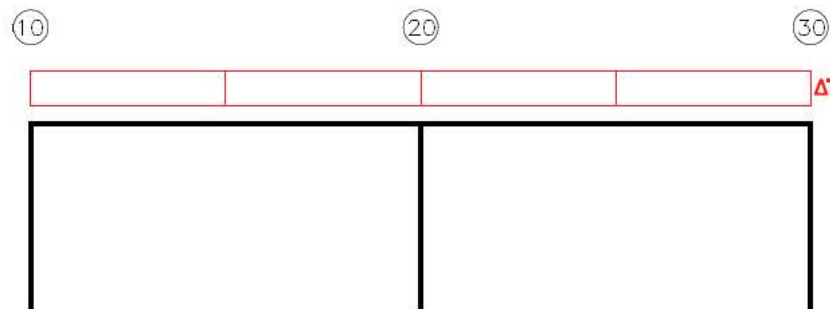


Abbildung 31: vereinfachter Ansatz des Temperaturlastfalls

Konstanter Temperaturanteil

Der konstante Temperaturanteil setzt sich aus dem minimalen und maximalen Wert der Außentemperatur, sowie der Aufstelltemperatur zusammen. Es gilt für Betonbrücken (Gruppe 3):

$$T_{e,min} = -17 \text{ K}$$

$$T_{e,max} = +37 \text{ K}$$

Aufstelltemperatur: +10 °C

Der charakteristische Wert der minimalen positiven Änderung (Ausdehnung) $\Delta T_{N,pos}$ und der maximalen negativen Änderung (Verkürzung) $\Delta T_{N,neg}$ ergeben somit:

$$\Delta T_{N,pos} = +37 \text{ K} - 10 \text{ K} = +27 \text{ K} \quad (\text{LF 31})$$

$$\Delta T_{N,neg} = -17 \text{ K} - 10 \text{ K} = -27 \text{ K} \quad (\text{LF 32})$$

Veränderlicher Temperaturanteil

Der veränderliche Anteil ergibt sich aus einer vorübergehenden Erwärmung bzw. Abkühlung des Brückenüberbaus. Diese Beanspruchung wird mit $T_{M,pos}$ (Oberseite wärmer als Unterseite) und $T_{M,neg}$ (Unterseite wärmer als Oberseite) berücksichtigt. Für Brücken mit Betonplatte als Überbau gilt:

$$\Delta T_{M,pos} = +15 \text{ K} \quad (\text{LF 33})$$

$$\Delta T_{M,neg} = -8 K \quad (\text{LF } 34)$$

Der Oberbelag der Fahrbahnplatte wird mit dem Faktor k_{sur} berücksichtigt. Für Schotterbelag $h \geq 60$ cm gilt:

$$\Delta T'_{M,pos} = k_{sur} * \Delta T_{M,pos} = 0,60 * 15 K = 9,00 K$$

$$\Delta T'_{M,neg} = k_{sur} * \Delta T_{M,neg} = 1,00 * (-8) K = -8,00 K$$

Tabelle 1 zeigt die möglichen Überlagerungskombinationen des konstanten und linearen Temperaturanteils. Die Kombinationen ergeben sich aus:

$$\Delta T'_M + \omega_N * \Delta T'_N \quad \text{bzw.} \quad \Delta T'_N + \omega_M * \Delta T'_M$$

Kombinationen		$\Delta T'_{N,pos}$	$\Delta T'_{N,neg}$	$\Delta T'_{M,pos}$	$\Delta T'_{M,neg}$	Lastfallnummer
		ω_N	ω_N	ω_M	ω_M	
Oberseite wärmer als Unterseite	Fall 1	0,35	-	1,0	-	35
		-	0,35	1,0	-	36
	Fall 2	1,0	-	0,75	-	37
		-	1,0	0,75	-	38
Oberseite kälter als Unterseite	Fall 1	0,35	-	-	1,0	39
		-	0,35	-	1,0	40
	Fall 2	1,0	-	-	0,75	41
		-	1,0	-	0,75	42

Tabelle 1: Lastkombinationen und Lastfälle

7.2.4 ASE – Berechnung des Systems

Mit dem Modul ASE können statische und dynamische Berechnungen beliebig geformter Tragwerke durchgeführt werden. Es wird vorwiegend bei komplexen 3D-Strukturen für lineare, nichtlineare und dynamische Berechnungen eingesetzt. Es werden hier die unter SOFILOAD definierten Lastfälle berechnet.

8. Schnittstellenprogrammierung

Im Rahmen dieser Arbeit wurde der Datenaustausch zwischen Siemens NX als BIM-Software und der Statik Software SOFiSTiK untersucht. Dazu wurden zwei Schnittstellen entwickelt, die geometrische Daten von NX an SOFiSTiK übergeben. Die Schnittstelle *NX2Sofistik_Vol* übergibt die CAD-Daten als Volumenmodell, während *NX2Sofistik_Area* ein Mittelflächenmodell erzeugt und an SOFiSTiK übergibt.

Die Schnittstelle wurde in der Entwicklungsumgebung NetBeans IDE mit Hilfe der objektorientierten Programmiersprache Java ausgearbeitet. Das NX Modell wird zunächst als JT-Datei mithilfe von Siemens NX exportiert und über das Parser Programm RaidaJTLoader eingelesen. RaidaJTLoader ist ein von Johannes Raida zur Verfügung gestelltes Java-Modul zum Importieren von JT Dateien für die weitere Verarbeitung (Raida, 2015). Anschließend wird ein DAT-File erzeugt, welches die geometrischen Daten und Materialien enthält. Das DAT File kann mit einem beliebigen Texteditor, so wie auch dem Teddy Editor von SOFiSTiK geöffnet und gelesen werden. Nachfolgend wird die Implementierung der Schnittstellen erläutert. Im Anhang A befinden sich die entsprechenden UML-Diagramme beider Schnittstellen. Ihre Strukturen werden im Folgenden weiter erläutert.

8.1 Einlesen der JT-Datei

Mit Hilfe der Raida-Bibliotheken werden die JT Dateien eingelesen. Das NX Modell wird bei Raida als reines Facettenmodell importiert. Dazu wird über das Objekt *CADImportInterface* und die Methode *importFile* die ausgewählte Datei geladen. Die geometrischen Daten sind beim JT Format in einem sogenannten Entity Container abgelegt, eine logische Gruppierung von Entitäts- und Assoziationssätzen. Die in der Array Liste *<EntityContainer>* hinterlegten Datensätze können über die Getter-Methode aufgerufen werden. Die Dreiecksflächen werden über einen sog. Indexed Face Set erzeugt, bei dem jedes Dreieck über drei Indexe definiert wird. Jeder Index zeigt auf einen Knoten, dem wiederum seine x-, y-, und z-Koordinaten und seine Flächen-Normale zugeordnet wird. Jedes Bauteil wird aus mehreren Dreiecksflächen zusammengesetzt. Die Anzahl der Dreiecksflächen ist von der Komplexität und

Größe des Bauteils abhängig. Die Gesamtmenge der Flächen ergibt sich über die Anzahl der Indexe, da jede Dreiecksfläche aus drei Indexen besteht.

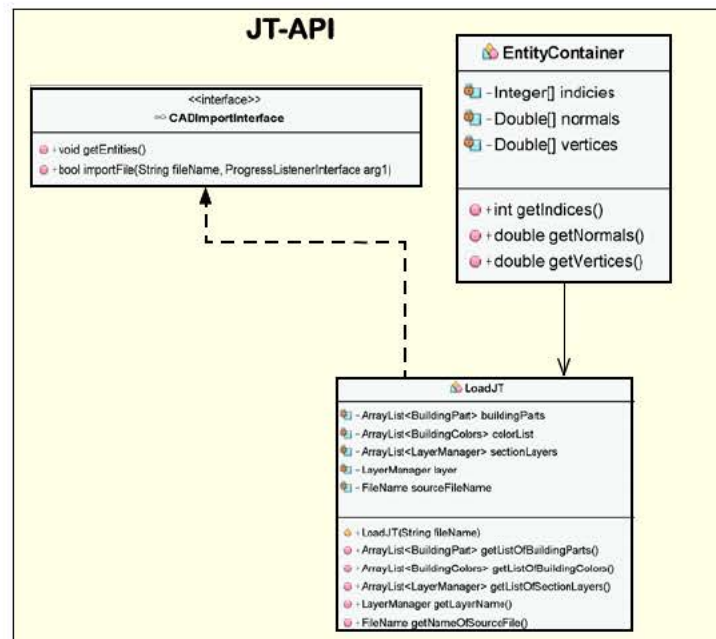


Abbildung 32: Ausschnitt des UML-Diagramms – JT-API nach (Weinholzer, 2014)

Weinholzer (Weinholzer, 2014) hat sich im Zuge seiner Arbeit mit der geordneten Speicherung dieser Daten beschäftigt. Die erstellte Klassenstruktur wurde übernommen und geringfügig angepasst. Die Klasse *Vertex* speichert die Koordinaten und Normalen jedes Knotens. Die Klasse *Face* erbt alle Informationen der *Vertex* Klasse als Arrayliste. Auf diese Weise kann eine Instanz *x* der Klasse *Face* die Informationen über drei Knoten und Normalen enthalten. Die Superklasse *BuildingPart* erbt wiederum alle Informationen der Klasse *Face* in einer Arrayliste (vgl. Abbildung 33). Zusätzlich existieren Methoden um den Bauteilnamen sowie die Bauteilfarbe zu speichern. Da über das JT-Format keine Materialinformationen übergeben werden können, wird ein Umweg über die Farbe gegangen. Mithilfe einer im Quellcode hinterlegten Farb-Materialmatrix können auf diese Weise die Materialinformationen wieder hergestellt werden.

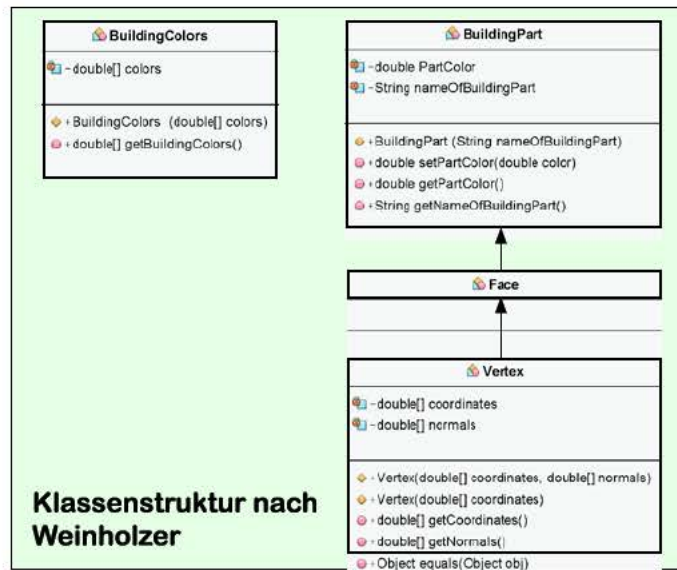


Abbildung 33: Ausschnitt des UML-Diagramms – die Klassen *Vertex*, *Face* und *BuildingPart* nach (Weinholzer, 2014)

Raida stellt eine *LoadJT*-Klasse zur Verfügung, die das Einlesen und Verarbeiten von *JT*-Dateien ermöglicht. Diese Klasse wurde von Weinholzer um die oben beschriebenen Klassen erweitert und im Zuge dieser Arbeit weiter angepasst. Abbildung 34 zeigt den Ausschnitt der *LoadJT*-Klasse die dem Objekt *buildingPart* die jeweiligen *Faces* hinzufügt. Hierfür wird über eine *for*-Schleife über die gesamte Anzahl an Flächen iteriert. Bei jedem Durchlauf werden die Informationen über die *vertices* (Knoten) und *normals* (Normalen) mit den Indices der jeweiligen Dreiecksflächen in Arrays abgespeichert. Anschließend wird ein leeres Objekt *face* der Klasse *Face* angelegt und die Knoten und Normalen werden gespeichert. Im letzten Schritt wird das *face* dem Objekt *buildingPart* hinzugefügt.

```

for (int j = 0; j < numberOfFaces; j++) {

    int index1 = indices[(j * 3) + 0];
    int index2 = indices[(j * 3) + 1];
    int index3 = indices[(j * 3) + 2];

    double[] vertex1 = new double[]{
        vertices[(index1 * 3) + 0], vertices[(index1 * 3) + 1], vertices[(index1 * 3) + 2]};
    double[] vertex2 = new double[]{
        vertices[(index2 * 3) + 0], vertices[(index2 * 3) + 1], vertices[(index2 * 3) + 2]};
    double[] vertex3 = new double[]{
        vertices[(index3 * 3) + 0], vertices[(index3 * 3) + 1], vertices[(index3 * 3) + 2]};

    double[] normal1 = new double[]{
        normals[(index1 * 3) + 0], normals[(index1 * 3) + 1], normals[(index1 * 3) + 2]};
    double[] normal2 = new double[]{
        normals[(index2 * 3) + 0], normals[(index2 * 3) + 1], normals[(index2 * 3) + 2]};
    double[] normal3 = new double[]{
        normals[(index3 * 3) + 0], normals[(index3 * 3) + 1], normals[(index3 * 3) + 2]};

    Face face = new Face();
    face.add(new Vertex(vertex1, normal1));
    face.add(new Vertex(vertex2, normal2));
    face.add(new Vertex(vertex3, normal3));
    buildingPart.add(face);
}
  
```

Abbildung 34: Ausschnitt der Klasse *LoadJT*: Generieren der Dreiecksflächen

Außerhalb der *for*-Schleife werden weitere Informationen wie die Bauteilfarbe sowie der Bauteilname des Bauteils ausgelesen und dem *buildingPart* zugeordnet. Im letzten Schritt wird das Objekt der Arrayliste *buildingParts* hinzugefügt.

8.2 Schreiben der DAT-Datei in der CADINP Sprache

In beiden Schnittstellen existiert die Hauptklasse *WriteDatFile*, die für das Einlesen und die Ausgabe der Informationen zuständig ist. Nachdem mithilfe der *LoadJT*-Klasse die Daten zwischengespeichert wurden folgt der Aufruf der statischen Methode *generateDatFile*, die für die Ausgabe der Daten zuständig ist. Der strukturelle Aufbau der Ausgabe richtet sich nach der CADINP Sprache von SOFiSTiK. Im ersten Schritt wird der Java-Klasse *FileWriter* eine leere Dat-Datei angelegt. *FileWriter* ist ein Bestandteil des *Java.io* Packet weshalb dieses der *WriteDatFile*-Klasse über einen *import* hinzugefügt ist. Über die Methode *writer.write()* kann in die erstellte Datei geschrieben werden. Der strukturelle Aufbau der Ausgabe ist in beiden Schnittstellen gleich, inhaltlich unterscheiden sie sich jedoch zum Teil.

In beiden Dateien wird zuerst der Pfad geschrieben, wo das zugehörige JT File zu finden ist, aus dem das Modell generiert wird. Anschließend werden Variablen erzeugt, deren Werte der Benutzer manuell in TEDDY festlegen muss. Im weiteren Verlauf werden die Programme AQUA und SOFIMSHC erzeugt. In AQUA werden die verwendeten Materialien und die Norm ausgegeben. Die Norm ist statisch im Quellcode hinterlegt und kann bei Bedarf in den Schnittstellen oder in der erzeugten Datei angepasst werden.

Die Materialienliste wird anhand der eingelesenen Farben dynamisch erzeugt. Wie in Kapitel 6.3 beschrieben, wurden in NX den Bauteilen die Materialien als Attribut zugewiesen. Jeder Betonfestigkeitsklasse wurde ein anderer Farbton zugeteilt. Die verwendeten Materialien werden in der Klasse *LoadJT* im Array *colors* abgespeichert. Die Arrayfelder entsprechen dabei den Rot-, Grün-, Blau-Anteilen (RGB-Werte) der Farbe. In der Klasse *WriteDatFile* wurde das Array *concrete* angelegt. Darin befinden sich die Betonfestigkeitsklassen (Feld 0), die dazugehörigen RGB-Werte (Feld 1) sowie die Materialnummer (Feld 2).

```

String[][] concrete = new String[8][3];

concrete[0][0] = "C 16";
concrete[0][1] = "42.82353222370148";
concrete[0][2] = "";

concrete[1][0] = "C 20";
concrete[1][1] = "40.00000357627869";
concrete[1][2] = "";

...
...
...

for (BuildingColors colors : buildingColors) {
    double color = colors.getBuildingColors()[0];
    for (String[] concrete1 : concrete) {
        if (concrete1[1].equals(String.valueOf(color))) {
            if (concrete1[2].equals("")) {
                [2] = String.valueOf(MatNr);
                MatNr++;
            }
        }
    }
}
}

```

Abbildung 35: Ausschnitt der Klasse WriteDatFile: Zuordnung der Materialien

Um die Liste zu erzeugen werden die Arrays *colors* und *concrete* verglichen. Zur Vereinfachung wird nur der Rotanteil betrachtet. Wird ein Farbwert erkannt, erhält das Feld 2 des *concrete* Arrays eine fortlaufende Materialnummer, beginnend bei 2. Die Zählung beginnt nicht bei 1, da Betonstahl immer vorkommt und im Quellcode fest die Materialnummer 1 besitzt. Anschließend wird über *concrete* iteriert und die Materialien im Programm AQUA ausgegeben, die über eine Materialnummer verfügen.

In SOFiSTiK werden Bauteile in dem Programm SOFIMSHC erzeugt. Zur besseren Übersicht wird jedes Bauteil aus der *buildingParts* Liste in einem eigenen SOFIMSHC erzeugt. Dazu werden im ersten SOFIMSHC Modul unter *Syst* alle Angaben zur Systemdefinition getroffen und in allen weiteren Modulen die Angabe *Syst Rest* festgelegt. Mit *Syst Rest* wird angegeben, das System mit vorherigen zu vernetzen.

```

writer.write("+prog sofimshc\r\n\r\n");
writer.write("#include Bauteilstärken\r\n");
writer.write("#include Bettung\r\n\r\n");
writer.write("Kopf System Bauteil ' ' + bp_Buffer.getPartName() + "'\r\n");
if (isFirst) {
    writer.write("Syst 3D gdiv 100000 GDIR POSZ $ ggfs. gdiv anpassen\r\n");
    isFirst = false;
} else {
    writer.write("Syst Rest\r\n");
}
writer.write("Steu mesh 2 +64\r\n");
writer.write("Steu Hmin 0.5\r\n\r\n");

```

Abbildung 36: Ausschnitt der Klasse WriteDatFile: Jedes Bauteil wird in einem eigenem SOFIMSHC Modul generiert

Wie in Abbildung 36 zu erkennen, wird mit der Zeile "+prog sofimshc" das Programm definiert. Die Verzweigung mit der Variable *isFirst* ist notwendig, um für das erste SOFIMSHC-Modul festzulegen. Jedes Bauteil wird zur besseren Übersicht in einem eigenen

SOFIMSHC-Modul erzeugt. In nachfolgenden Kapiteln wird auf die Unterschiede der Programmiercodes von NX2Sofistik_Vol und NX2Sofistik_Area eingegangen.

8.2.1 Volumenmodell

Volumenelemente werden in SOFiSTiK unter *Steu mesh 3* eingestellt. Damit wird die Vernetzung von Stäben, Flächen, und Volumenkörpern aktiviert. Zur Generierung der Volumenkörper werden über eine *for*-Schleife die drei Knoten der Dreiecksfläche *face* abgefragt. Jede Strukturlinie *SLN* hat einen Anfangs- und Endpunkt, die über *SLNB* definiert werden. Der Aufbau ist dann wie folgt:

Strukturlinie 1: Vertex 1 Vertex 2

Strukturlinie 2: Vertex 2 Vertex 3

Strukturlinie 3: Vertex 3 Vertex 1

Der Strukturfläche *SAR* werden dann die drei Strukturlinien zugewiesen. Aus den Strukturflächen eines *buildingPart* lässt sich dann der Volumenkörper mit *SVO* generieren. Jeder Volumenkörper bekommt eine eigene Gruppennummer, sowie die dazugehörige Materialnummer zugewiesen. Über *SVOS* werden die referenzierenden Strukturflächen angegeben.

Den Fundamenten werden Bettungen zugeordnet, welche die Nachgiebigkeit des Bodens realitätsnah abbilden. Die Bettungsmodule werden als Variablen im Satz *SAR* übergeben. Der Benutzer wird am Anfang der Textdatei aufgefordert die Werte der Variablen zu bestimmen. Den Flächen der Unterseite der Fundamente wird das Bettungsmodul *CB* für Senkbettung normal zur Fläche zugeordnet. Die Seitenflächen der Fundamente bekommen das Bettungsmodul *CT* für Querbettung in der Fläche zugewiesen. Die Zuordnung der Bettungen zu den jeweiligen Dreiecksflächen erfolgt über die Normalen *normals* der Fundamente, da diese senkrecht auf den Dreiecksflächen stehen und somit die Orientierung angeben.

8.2.2 Flächenmodell

Das Flächenmodell wird wie das Volumenmodell über das JT Format importiert, was bedeutet, dass die Mittelflächen in kleinere Dreiecksflächen unterteilt sind. Bei ebenen Flächen besteht jedoch die Möglichkeit, das System so zu approximieren, dass die innenliegenden Strukturlinien entfernt werden und das System nur noch über die Randelemente erzeugt wird (vgl. Abbildung 37). Der Vorteil einer großen Fläche ist, dass der Netzgenerator von SOFiSTiK ein homogeneres Netz erzeugen kann, als bei Dreiecksflächen, die besonders in den Ecken ein sehr heterogenes FE-Netz aufweisen.

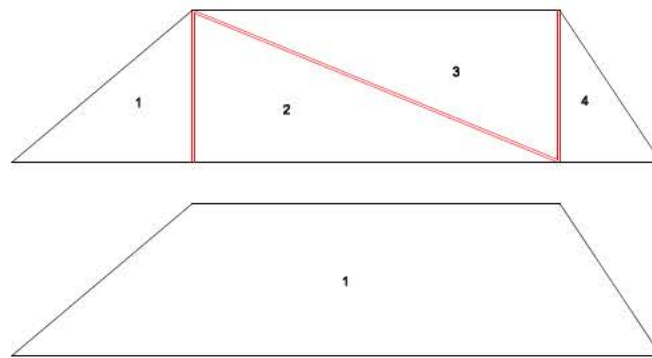


Abbildung 37: Entfernen doppelter Strukturlinien (rot)

Alle innenliegenden Strukturlinien liegen doppelt vor, da sie immer zu zwei Dreiecksflächen gehören. Diese Linien müssen entfernt werden, wobei die Fläche über die äußeren Linien geschlossen wird. Um dies durchzuführen, wurde ein Ausgabe-Puffer konzipiert. Die Klassen *OutputBuffer* sowie *BuildingPartBuffer* stellen die Struktur der zu erzeugenden Dat Datei für SOFiSTiK dar. Die Klasse *OutputBuffer* enthält dazu eine Arrayliste des Typs *BuildingPartBuffer*, welcher für die Zwischenspeicherung der einzelnen Bauteile verantwortlich ist. In der *generateDatFile* Methode werden nun nicht direkt die Daten auf der *LoadJT*-Klasse ausgegeben, sondern verarbeitet und in dem Objekt *output_buffer* der Klasse *OutputBuffer* zwischengespeichert (vgl. Abbildung 38).

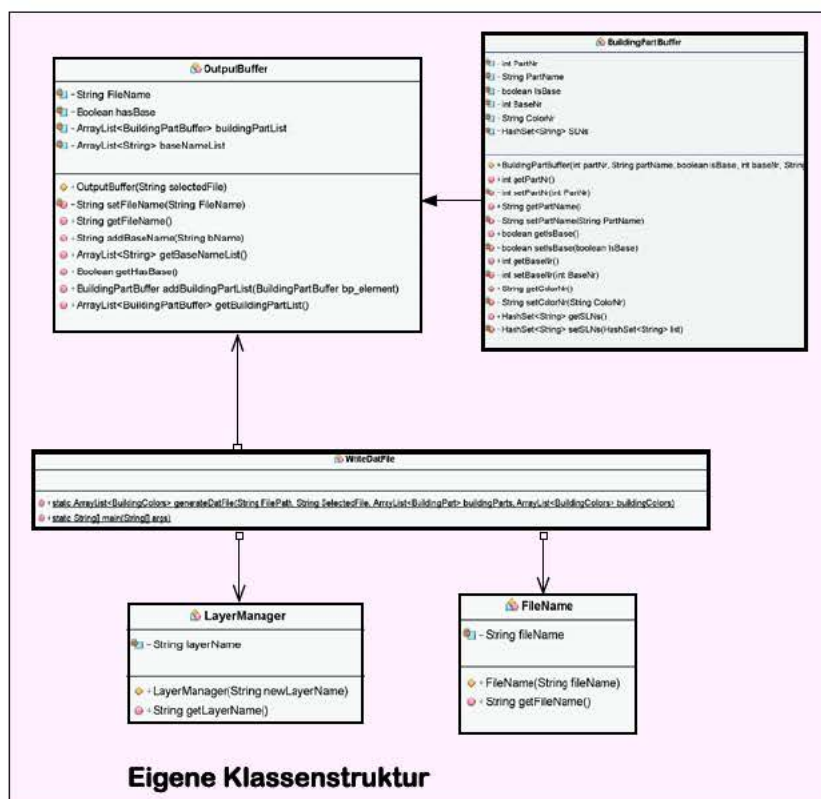


Abbildung 38: Ausschnitt des UML-Diagramms – die Klassen *BuildingPartBuffer*, *OutputBuffer* und *WriteDatFile*

Ähnlich 8.2.1 werden zunächst alle Bauteile durchlaufen, jedoch werden die Informationen formatiert und zwischengespeichert. Im ersten Schritt werden das Material sowie der Bauteilname gespeichert. Anschließend folgt die Überprüfung auf ein Fundament. Im nächsten Schritt wird über die Dreiecksflächen des jeweiligen Bauteils iteriert. Abbildung 39 zeigt die zugehörige *for*-Schleife, in welcher die eindeutigen Strukturlinien zwischengespeichert werden.

```

for (Face face : bp) {
    for (int i = 0; i < (face.size()); i++) {
        int j = (i + 1) % 3;
        double x1 = face.get(i).getCoordinates()[0];
        double y1 = face.get(i).getCoordinates()[1];
        double z1 = face.get(i).getCoordinates()[2];
        double x2 = face.get(j).getCoordinates()[0];
        double y2 = face.get(j).getCoordinates()[1];
        double z2 = face.get(j).getCoordinates()[2];

        String K1K2 = x1 + ";" + y1 + ";" + z1 + ";" + x2 + ";" + y2 + ";" + z2;
        String K2K1 = x2 + ";" + y2 + ";" + z2 + ";" + x1 + ";" + y1 + ";" + z1;

        if (hsUnique.contains(K1K2) || hsUnique.contains(K2K1)) {
            hsUnique.remove(K1K2);
            hsUnique.remove(K2K1);
            slnNr--;
        } else {
            hsUnique.add(K1K2);
            slnNr++;
        }
    }
}

```

Abbildung 39: Verarbeitung der Dreiecksflächen

Die äußere Schleife durchläuft alle in dem Objekt *bp* des Typs *BuildingPart* gespeicherten Flächen. Für jede Fläche werden in der inneren Schleife sowohl die Anfangs- als auch Endpunkte der einzelnen Strukturlinien ermittelt (*x1*, *y1*, ...). Da die Linien nicht notwendigerweise in die gleiche Richtung laufen, werden zwei Strings *K1K2* und *K2K1* erzeugt um richtungsunabhängig vergleichen zu können. Besitzt das temporär angelegte *HashSet* *hsUnique* bereits einen der Strings, so muss die Strukturlinie vollständig entfernt werden, da es sich um eine doppelte Linie handelt. Andernfalls wird die soeben erzeugte Linie dem *HashSet* hinzugefügt. Nach Abschluss der äußeren Schleife sind in dem Objekt *hsUnique* nur die eindeutigen Strukturlinien erfasst.

Die Klasse *BuildingPartBuffer* besitzt ebenfalls ein *HashSet* *SLNs*, welches abschließend die Informationen des *hsUnique* bekommt. Nachdem über alle Bauteile iteriert wurde, wird eine *Dat* Datei erzeugt und mit den gespeicherten Informationen des *output_buffer* beschrieben. Der Umweg über weitere Klassen für die Datenspeicherung in Form eines Ausgabe-Puffers wurde gewählt, da auf diese Weise beide Schnittstellen über denselben Kern verfügen können.

Gibt es Updates von der Raida Bibliothek bzw. Änderungen an der LoadJT-Klasse, ist es möglich mit geringem Aufwand beide Schnittstellen zu aktualisieren.

9. Auswertung der Schnittstellen anhand des Anwenderbeispiels

In diesem Kapitel werden die erzeugten Modelle und deren Ergebnisse gegenübergestellt und ausgewertet. Betrachtet werden das aus NX erzeugte Flächenmodell, sowie das manuell erstellte Flächenmodell. Zunächst werden die geometrischen Abmessungen der Bauteile verglichen, da geometrische Abweichungen den Verlauf und die Größe der Schnittgrößen beeinflussen und somit auch für die Bemessung maßgebend sind. Anschließend werden die in 7.2.3 definierten Lastfälle angesetzt und die daraus resultierenden Schnittgrößen an den maßgebenden Stellen betrachtet. Dazu wird Ausmaß und Ursache der Differenzen der Schnittgrößen bewertet.

Eine Auswertung des Volumenmodells ist nicht möglich, da es bei der Systemgenerierung in SOFiSTiK zum Abbruch kommt. Grund dafür ist, dass die Erzeugung des Volumenmodells anhand von Facetten sehr komplex ist. Die Bauteile werden über facettierte Dreiecksflächen erzeugt. Jedes neu erzeugte Bauteil wird mit den bereits generierten Bauteilen zu einem Gesamtsystem zusammengefügt. Die detaillierte Struktur der Bauteile führt zu einer hohen Anzahl an Facetten. Abbildung 40 zeigt das erzeugte Modell bestehend aus dem Widerlager Achse 10 und dem dazugehörigen Fundament. Auf Grund geometrischer Toleranzen bricht der Algorithmus des Netzgenerators beim Vernetzen ab, womit ein weiteres Anfügen anderer Bauteile nicht mehr möglich ist.

Im Folgenden wird daher nur auf die Flächenmodelle eingegangen. Das manuell erstellte Flächenmodell wird im Folgenden als konventionelles Modell bezeichnet, das aus NX generierte Modell als NX Modell.

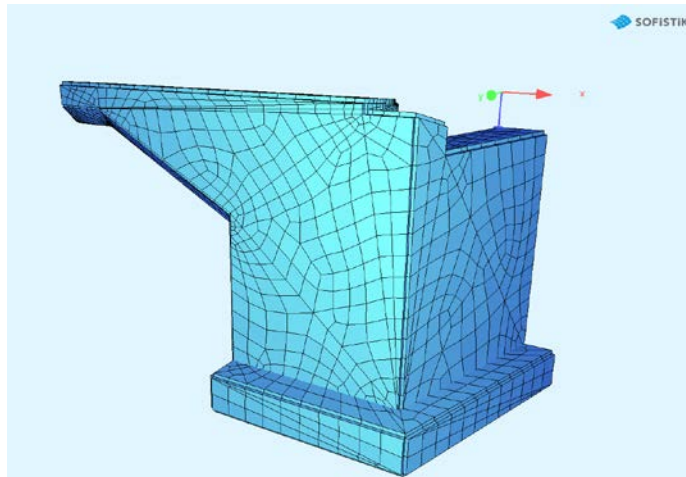


Abbildung 40: Widerlager Achse 10 mit Fundament als Volumenmodell in SOFiSTiK

9.1 Geometrische Abmessungen

Die geometrische Dimensionierung eines Modells ist maßgebend für die resultierenden Schnittgrößen. Es gibt nur in wenigen Bauteilen leichte Abweichungen der Geometrie zwischen dem konventionellen und dem NX Modell. Insgesamt stimmt die Geometrie der beiden Modelle im Großen überein. Im Anhang A sind die dazugehörigen Abbildungen zusammengefasst. Ursache der geometrischen Abweichungen kann sein:

- Abweichende Modellierung der Bauteile in NX
- Statische Vereinfachungen wurden nicht getroffen
- Abweichungen durch Dimensionsreduzierung

Des Weiteren wurden in der TEDDY Datei des NX Modells noch manuell Strukturlinien eingefügt, die den Biege- und Schubschnitten der jeweiligen Bauteile entsprechen. In 0 ist die geometrische Lage der Biege- und Schubschnitte der Bauteile dargestellt.

Nachfolgend werden die einzelnen Bauteile betrachtet und die Geometrie verglichen sowie die Ursache geometrischer Abweichungen untersucht.

9.1.1 Fahrbahnplatte

Der Rahmenriegel besteht bei beiden Flächenmodellen aus einer zweifeldrigen Fahrbahnplatte mit gekoppelten Kragarmen. Die Schnittgrößen hängen somit vor allem von der Stützweite der beiden Felder ab. Die entsprechenden Abbildungen sind dem Anhang zu entnehmen (vgl. Anhang B).

Bei beiden Modellen beträgt die Stützweite der Felder etwa 16,80 m und entspricht somit der realen Stützweite des ausgeführten Bauwerks. Die Gesamtlänge der Fahrbahn beträgt jeweils 33,60 m. Die Fahrbahnbreite (ohne Kragarme) beläuft sich beim konventionellen Modell auf 4,27 m, beim NX Modell auf 4,26 m. Die Abweichung ist somit vernachlässigbar klein.

Für die im NX Modell generierten Flächen des Kragarms müssen nachträglich noch die entsprechenden Kopplungen der benachbarten Flächen manuell in die TEDDY Datei eingefügt werden. Es werden jeweils die in Z-Richtung übereinander liegenden Strukturlinien der Kragarm-Flächen gekoppelt.

9.1.2 Flügelwände

Die Widerlager bestehen aus einer Kammerwand und einer Flügelwand. Beim konventionellen Modell wurde bei der Flügelwand der gevoutete Teil berücksichtigt und eine veränderliche Bauteildicke in diesem Bereich angesetzt.

In NX wird die Mittelfläche aus den Seitenflächen der Flügelwand generiert. Der gevoutete Bereich wird dabei nicht beachtet, so dass die erzeugte Mittelfläche nur der Fläche des konstanten Bereichs entspricht. Der Steifigkeitsanteil des gevouteten Bereichs entfällt, so dass die Flügelwand somit auf der sicheren Seite bemessen wird. Alternativ kann der Bereich der Voute auch noch nachträglich in SOFiSTiK hinzugefügt werden.

Des Weiteren ergeben sich Unterschiede in der Höhe der Bauteile. Im konventionellen Modell wurde die Längsneigung der Eisenbahnüberführung als statische Vereinfachung vernachlässigt, sodass die Bauteile in Achse 10 die gleiche Höhe aufweisen wie die Bauteile in Achse 30. Im NX Modell ist die Längsneigung enthalten. Dadurch haben die Bauteile in Achse 10 eine kleinere Höhe als in Achse 30. Insgesamt sind beide Flügelwände des NX Modells etwas höher als die Flügelwände des konventionellen Modells. Dies lässt sich auf leichte Abweichungen bei der Modellierung der Widerlager in NX zurückführen.

9.1.3 Kammerwände

Im konventionellen Modell wurde der auskragende Versprung der Kammerwände beachtet. Dieser ergibt sich aus dem Höhenunterschied zur Widerlagerwand. Durch die automatisierte Mittelflächengenerierung in NX ist der auskragende Teil im NX Modell nicht enthalten.

Wie auch bei den Flügelwänden weisen die Kammerwände, auf Grund der Längsneigung, eine unterschiedliche Höhe auf. Die Kammerwand hat in Achse 10 eine Höhe von 7.06 m, in

Achse 30 eine Höhe von 7,42 m. Ihre durchschnittliche Höhe von 7,24 m entspricht jedoch etwa der Höhe der Kammerwände des konventionellen Modells (= 7,20 m).

9.1.4 Mittelpfeiler

Der Mittelpfeiler hat in beiden Modellen etwa die gleichen geometrischen Abmessungen. Der Mittelpfeiler ist im NX Modell lediglich 3,0 cm höher und 6,0 cm schmaler als im konventionellen Modell.

9.1.5 Fundamente

Die Abmessungen der Fundamente stimmen ebenfalls in etwa überein. Das Mittelpfeilerfundament ist im NX Modell 20 cm kürzer als im konventionellen Modell, was sich auf eine abweichende Modellierung in NX zurückführen lässt. Das Fundament in Achse 30 ist im Bereich der Flügelwand etwas länger als im konventionellen Modell, dies hängt auf Grund parametrischer Abhängigkeiten mit der geometrischen Modellierung der Flügelwand zusammen.

9.2 Lasten und Schnittgrößen

Es wurden typische Lastfälle einer Eisenbahnbrücke auf beide Modelle angesetzt. Da jeweils die gleichen Lasten angesetzt wurden, sind Abweichungen der Schnittgrößen auf geometrische Unterschiede zurückzuführen. Anhand der Ergebnisse können damit die Auswirkungen der Abweichungen beurteilt werden und Rückschlüsse daraus gezogen werden.

In Kapitel 7.2.3 wurden bereits die gewählten Lastfälle definiert. Als typische Lastfälle werden das Eigengewicht als ständige Last, das Lastmodell LM 71 als veränderliche Last und Temperatur als Zwang angesetzt. Für die Auswertung wurde repräsentativ für das Lastmodell LM 71 eine Belastung angesetzt, die die Streckenlast $q'_{v,k}$ und die Achslast $Q'_{v,k}$ in der Feldmitte kombiniert (LF 30). Als Temperaturlastfall wurde der veränderliche Temperaturanteil $\Delta T_{M, \text{pos}} = +15 \text{ K}$ (LF 33) für die Auswertung gewählt.

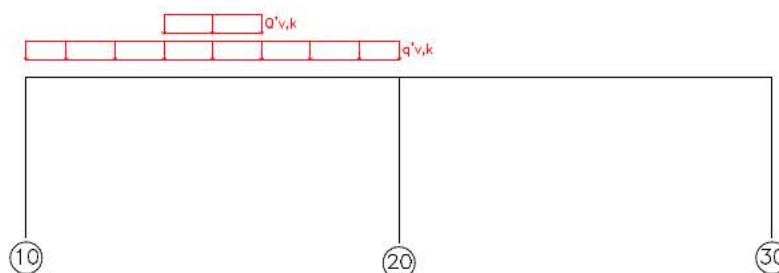


Abbildung 41: LM 71 Laststellung in Feldmitte (LF30)

Da keine weitere Bemessung der EÜ Ventschow Arbeit stattfindet, ist eine Überlagerung der Schnittgrößen nicht notwendig. In Anhang C sind die Ergebnisse der beiden Flächenmodelle dargestellt. Die Schnittkraftverläufe wurden entlang des Längsschnitts A-A durch das Rahmenbauwerk wiedergegeben. Stellvertretend wurde das Biegemoment m_{-xx} für die oben genannten Lastfälle verwendet.

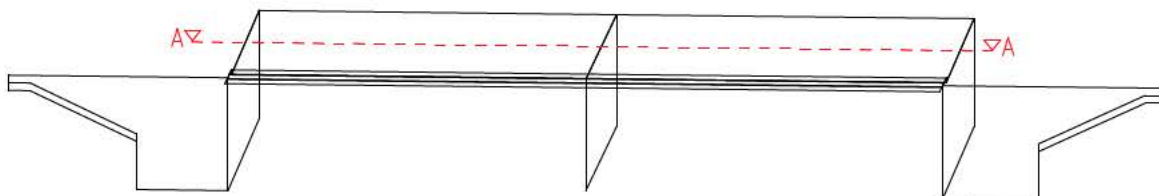


Abbildung 42: Längsschnitt A-A durch Flächentragwerk

Ein Vergleich der Schnittgrößenverläufe zeigt, dass beide Modelle ähnliche Ergebnisse aufweisen. Die nachfolgenden Tabellen zeigen die Ergebnisse des Biegemoments m_{-xx} der Fahrbahnplatte. Das Stützmoment in Achse 20 ist für den Lastfall 30 im NX Modell etwas erhöht. Ebenso weichen die Stützmomente des NX Modells für den Lastfall 33 geringfügig vom konventionellen Modell ab. Dies kann auf die vorhandene Längsneigung der Fahrbahnplatte des NX Modells zurückgeführt werden.

	Konventionelles Modell	NX Modell	Delta
Achse 10-20	330,7	349,1	18,4
Achse 20	-880,6	-911,6	31,0
Achse 20-30	379,5	338,5	41,0

Tabelle 2: Biegemoment m_{-xx} [kNm], Eigengewicht (LF1)

	Konventionelles Modell	NX Modell	Delta
Achse 10	-409.0	-423.1	14,1
Achse 10-20	588.3	590.0	1,7
Achse 20	-527.1	-584.4	57,3

Tabelle 3: Biegemoment m_{-xx} [kNm], LM 71 (LF30)

	Konventionelles Modell	NX Modell	Delta
Achse 10	423.7	478.7	55,0
Achse 20	645.1	698.8	53,7
Achse 30	392.0	458.2	66,2

Tabelle 4: Biegemoment m_{-xx} [kNm], Temperatur (LF33)

9.3 Bewertung der in NX generierten Modelle

9.3.1 Volumenmodell

Die Auswertung des Volumenmodells konnte im Rahmen dieser Arbeit nicht erfolgen. Die Volumenkörper der Bauteile werden über tesselierte Dreiecksflächen erzeugt. Die hohe Anzahl der Facetten führte auf Grund geometrischer Toleranzen zum Abbruch bei der Netzgenerierung in SOFiSTiK. Bei einer zu langen Generierung entstehen Rundungsfehler, die zum Abbruch führen können, wenn der Endwert mit keiner ausreichenden Genauigkeit erreicht wird oder eine bestimmte Anzahl an Iterationen überschritten wird (SOFiSTiK, 2014). Viele Facetten, die in einem Punkt zusammenfallen, oder lange, spitzwinklige Dreiecke sind mögliche Gründe dieser Problematik bei der Vernetzung.

Des Weiteren ist die Generierung eines Volumenmodells über Facetten allgemein kritisch zu betrachten. In SOFiMSHC wird das FE-Netz innerhalb der vorgegebenen Strukturlinien gebildet. Auf Grund der Facetten entsteht ein sehr inhomogenes Netz, dessen qualitative Auswertung der Ergebnisse zu prüfen wäre.

Demzufolge kann keine Empfehlung für den Export eines digitalen Gebäudemodells über das neutrale Dateiformat JT und für den Einsatz der Schnittstelle *NX2Softistik_Vol* ausgesprochen werden. Das Volumentragwerk mit facettierter Oberflächenstruktur führt aus oben genannten Gründen zu keinen verwertbaren Ergebnissen.

9.3.2 Flächenmodell

Anhand des Anwenderbeispiels konnte die Anwendbarkeit der Schnittstelle *NX2Softistik_Area* überprüft werden. Es wurde in der Software NX anhand des digitalen Gebäudemodells das entsprechende Flächentragwerk konstruiert. Über die Schnittstelle konnten die geometrischen Informationen sowie Materialeigenschaften erfolgreich in die CADINP Sprache von SOFiSTiK importiert werden. Vergleicht man die geometrischen Abmessungen des NX Modells mit denen des konventionellen Modells, so kann eine hinreichende Übereinstimmung festgestellt werden. Allerdings ist eine manuelle Nachbearbeitung in SOFiSTiK weiterhin nötig. Durch den Export gehen die Informationen über die Bauteildicke der Bauteile verloren, so dass diese im Nachgang in SOFiSTiK eingegeben werden müssen. Um diese Eingabe zu erleichtern, können die Bauteildicken am Anfang der generierten TEDDY Datei über Variablen festgelegt werden. Anpassungen der Bauteildicke können dadurch bequem übernommen werden. Da in NX keine Informationen über die Baugrundsituation vorliegen, müssen diese ebenfalls nachträglich in SOFiSTiK ergänzt werden. Das Bettungsmodul der Fundamente kann ebenfalls über Variablen bestimmt werden. Des Weiteren müssen Bauteil-

Kopplungen, wie sie etwa bei dem Kragarm der Fahrbahnplatte vorkommen, vom Anwender festgelegt werden.

Ein Vergleich beider Systeme unter Last zeigt, dass die resultierenden Schnittgrößen vergleichbare Ergebnisse liefern. Die Längsneigung des Fahrbahnüberbaus hat keinen nennenswerten Einfluss auf die Ergebnisse der Schnittgrößenverläufe. Dies lässt sich auf die geringe Neigung zurückführen. Es sei jedoch erwähnt, dass größere Längs- wie auch Querneigungen durchaus Auswirkungen auf die Schnittgrößenverläufe haben können. In diesem Fall wäre eine weitere Vereinfachung des Modells in NX notwendig.

Ein großer Vorteil der automatisierten Tragwerksgenerierung ist, dass bei Änderungen des Bauwerks das Flächenmodell in NX automatisch angepasst wird. Die Mittelflächen der Bauteile, aus denen das Flächentragwerk zusammengesetzt wird, werden aus den Seitenflächen der Bauteile generiert. Ändert sich die Geometrie eines Bauteils, so passt sich die Mittelfläche automatisch an die neuen Seitenverhältnisse an. Da es sich jedoch um eine Direktschnittstelle handelt, ist ein erneutes Exportieren der aktualisierten geometrischen Daten notwendig.

Als nachteilig erweist sich die Generierung von Mittelflächen gevouteter Bauteile. Je nach geometrischer Ausbildung der Voute wird sie bei der Generierung der Mittelfläche gar nicht oder nur gemittelt berücksichtigt. Die Mittelfläche wird anhand der Seitenflächen des Bauteils ermittelt. Ist eine Seitenfläche größer als die andere, so wird das Mittel beider Flächen errechnet und als Mittelfläche angesetzt. In beiden Fällen ist somit für eine korrekte Darstellung der Voute eine manuelle Nachbearbeitung in SOFiSTiK nötig.

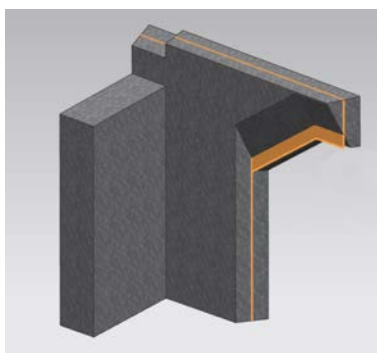


Abbildung 43: gemittelte Fläche bei gevouteten Bauteilen

Zusammenfassend kann festgestellt werden, dass eine Systemgenerierung eines Flächentragwerks anhand eines digitalen Gebäudemodells in NX möglich ist, aber derzeit eine Anwendung der Schnittstelle nicht empfohlen wird. Vor dem Export müssen erste Maßnahmen zur Generierung des Flächenmodells in NX getroffen werden und es sind weiterhin manuelle Anpassungen in SOFiSTiK notwendig. Es kann daher nur von einer Verlagerung des Arbeits-

und Zeitaufwandes von der Berechnungssoftware in Richtung BIM-Software gesprochen werden. Eine deutliche Zeitersparnis kann somit noch nicht festgestellt werden.

10. Zusammenfassung und Ausblick

10.1 Zusammenfassung

Im Zentrum dieser Arbeit stand die Entwicklung einer Schnittstelle für den Datenaustausch zwischen Siemens NX und SOFiSTiK zur Erweiterung der BIM-Integration. Ziel war es, geometrische Daten aus dem digitalen Gebäudemodell in NX über ein neutrales Datenformat zu exportieren und in das FE-Programm SOFiSTiK zu übergeben. Es wurde dabei untersucht, wie das digitale Gebäudemodell als Grundlage für die Tragwerksplanung herangezogen werden kann. Dazu wurden zwei Methoden entwickelt, wie aus einem bestehenden 3D-Modell ein Volumen- bzw. Flächentragwerk generiert werden kann.

Der erste Teil der Arbeit befasste sich mit der parametrischen Modellierung eines Brückenbauwerkes mit Hilfe der Software Siemens NX, welches gleichzeitig als Anwenderbeispiel diente. Für den Datenaustausch wurde das in NX vorhandene Grafikformat JT gewählt. Das leichtgewichtige Format eignet sich besonders gut für den Austausch geometrischer Informationen zwischen unterschiedlicher Software. Anschließend wurden getroffene Maßnahmen erläutert, die für eine erfolgreiche Tragwerksgenerierung mit dem Grafikformat notwendig sind.

Der zweite Teil befasste sich mit der konventionellen Modelleingabe in SOFiSTiK. Das Brückenbauwerk wurde dabei als Flächentragwerk modelliert und exemplarisch mit typischen Lastfällen belastet. Die Eingabe erfolgte im TEDDY-Editor in der CADINP Sprache von SOFiSTiK.

Im nächsten Schritt wurden zwei Direktschnittstellen in der Programmiersprache Java entwickelt, die die geometrischen Daten aus den erstellten JT-Dateien in die CADINP-Sprache von SOFiSTiK konvertieren. Die erste Schnittstelle wurde für den Datenaustausch von Volumenmodellen entwickelt, die zweite für Flächenmodelle.

Abschließend wurden die Schnittstellen anhand des Anwenderbeispiels überprüft. Dazu wurde das aus NX erzeugte Flächen- und Volumenmodell dem konventionellen Modell ge-

genübertestellt, welches manuell in SOFiSTiK eingegeben wurde. Es wurden die geometrischen Abmessungen der Modelle verglichen und die resultierenden Schnittgrößen aus der aufgetragten Belastung beurteilt.

10.2 Fazit

Building Information Modeling soll Planungs- und Ausführungsprozesse im Bauwesen verbessern und dabei Kosten- und Zeitaufwand reduzieren. Die Eingliederung der Tragwerksplanung in die BIM-Koordination gilt als Einschnitt in den Datenaustausch. Im Rahmen dieser Arbeit wurden daher zwei Direktschnittstellen entwickelt, die den Datentransfer von NX als BIM-Software zu der FE-Software SOFiSTiK über das Dateiformat JT ermöglichen soll.

Die Generierung eines Volumenmodells über das JT-Format erwies sich als ungeeignet. Durch das Facettieren der Oberfläche der Bauteile entsteht eine hohe Anzahl an Dreieckselementen. Diese können zu Problemen bei der Netzgenerierung in SOFiSTiK führen, da Rundungsfehler zum Abbruch des Algorithmus führen können. Des Weiteren werden die FE-Elemente innerhalb der Dreiecksflächen generiert, wodurch ein inhomogenes FE-Netz entsteht. Die Untersuchungen zeigen daher, dass der Modellexport von NX über das Grafikformat JT für die Erstellung von Volumentragwerken nicht geeignet ist.

Eine vollautomatische Generierung von dimensionsreduzierten Tragwerken aus einem bestehenden 3D-Modell ist mittels neutraler Grafikformate nicht ohne weiteres möglich, da diese nicht für diesen Zweck ausgerichtet sind. Mit Hilfe weniger Befehle können jedoch reduzierte Flächenelemente in NX erzeugt werden und diese als Flächentragwerk angepasst werden. Über die Schnittstelle *NX2Sofistik_Area* kann die daraus erzeugte JT-Datei in SOFiSTiK eingelesen werden. Die Schnittstelle ermöglicht es ebenfalls, dass nur die äußere Berandung der Flächenelemente übergeben wird und die Bauteile somit nicht mehr in tessellierter Form vorliegen. Es sind weiterhin manuelle Anpassungen in der erzeugten SOFiSTiK-Datei notwendig. Bauteildicken, Fundamenten-Bettungen und Bauteilkopplungen müssen nachträglich festgelegt werden. Eine Gegenüberstellung des generierten Flächenmodells mit dem Flächenmodell einer konventionellen Eingabe zeigt, dass die Systeme in Geometrie und Schnittgrößenverlauf in summa übereinstimmen. Lediglich bei gevouteten Bauteilen kann es zu Abweichungen kommen.

Die Systembildung in SOFiSTiK über die Schnittstelle *NX2Sofistik_Area* ist demnach möglich. Ein Vorteil dieser automatisierten Tragwerksgenerierung ist, dass sich bei Änderungen des Bauwerks in NX das enthaltene Flächenmodell automatisch anpasst. Dennoch wird der Export über die Schnittstelle derzeit nicht empfohlen. Durch die notwendige Anpassung des

Modells in NX und die manuelle Nachbearbeitung in SOFiSTiK ist noch keine echte Optimierung des Arbeits- und Zeitaufwandes zu verzeichnen. Der Arbeitsaufwand verlagert sich daher nur mehr von der Berechnungssoftware zu NX. Die konventionelle Eingabe von Tragwerksystemen wird daher weiterhin als „Best Practise“ Prozess empfohlen.

10.3 Ausblick

Der Trend zur BIM-basierten Projektabwicklung nimmt in Deutschland und auf internationaler Ebene immer mehr zu. Auch im Ingenieurbau werden verstärkt Lösungen untersucht, wie die BIM-Methode erfolgreich integriert werden kann.

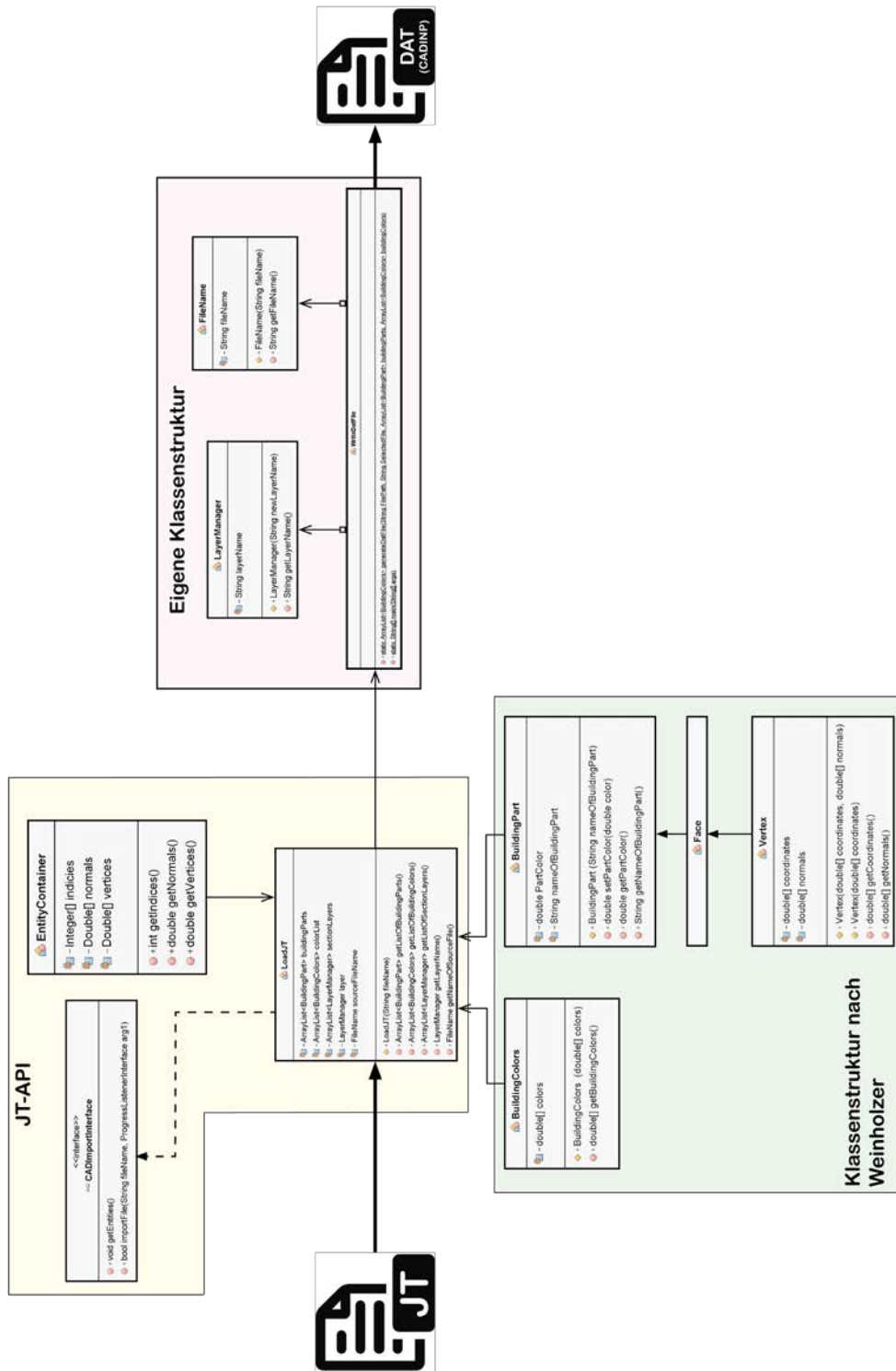
Die vorliegende Arbeit hat diesbezüglich Lösungsansätze im Bereich Optimierung von Brückenbauprozessen untersucht. Problematisch an den entwickelten Schnittstellen ist jedoch ein fehlender Rückkanal zur BIM-Software. Wird das Modell in Sofistik angepasst, werden die Änderungen nicht automatisch in das Ausgangsmodell von NX übernommen. Um Inkonsistenzen zu vermeiden, muss der Benutzer folglich das NX-Modell manuell anpassen. Es können daher nicht alle Vorteile des Building Information Modeling ausgenutzt werden. Die vorgestellte Arbeit kann daher nach Jernigan nur als „kleine“ BIM-Lösung klassifiziert werden (Jernigan, 2008). Des Weiteren ist eine Weiterentwicklung der vorgestellten Schnittstelle *NX2Sofistik_Area* denkbar. Weitere Untersuchungen zur Optimierung der automatisierten Dimensionsreduzierung sind nötig. Die vorgestellte Lösung ist derzeit nur für Flächentragwerke anwendbar. Eine Erweiterung der Schnittstelle für Stabsysteme wäre sinnvoll. Zusätzlich ist die allgemeine Anwendbarkeit der Schnittstellen anhand weiterer Bauwerksmodelle zu überprüfen. Darüber hinaus ist eine Einbindung bauspezifischer Aspekte in NX für eine durchgängige Wertschöpfungskette notwendig.

Die Einführung einer „großen“ BIM-Lösung wird im Bereich Ingenieurbauwerke noch mehrere Jahre dauern. Eine verstärkte Weiterentwicklung eines einheitlichen Dateiformats für Ingenieurbauwerke ist unerlässlich. Die Einführung eines IFC-Bridge Standards für Brückenbauwerke ist eine wichtige Voraussetzung für eine erfolgreiche BIM-Integration im Ingenieurbau. Dies kann nur durch weitere Forschungsarbeiten und eine enge Zusammenarbeit von Firmen und Organisationen wie buildingSMART erreicht werden.

Anhang A

UML-Klassendiagramme

A 1: UML-Klassendiagramm NX2Sofistik_Vol



Anhang B

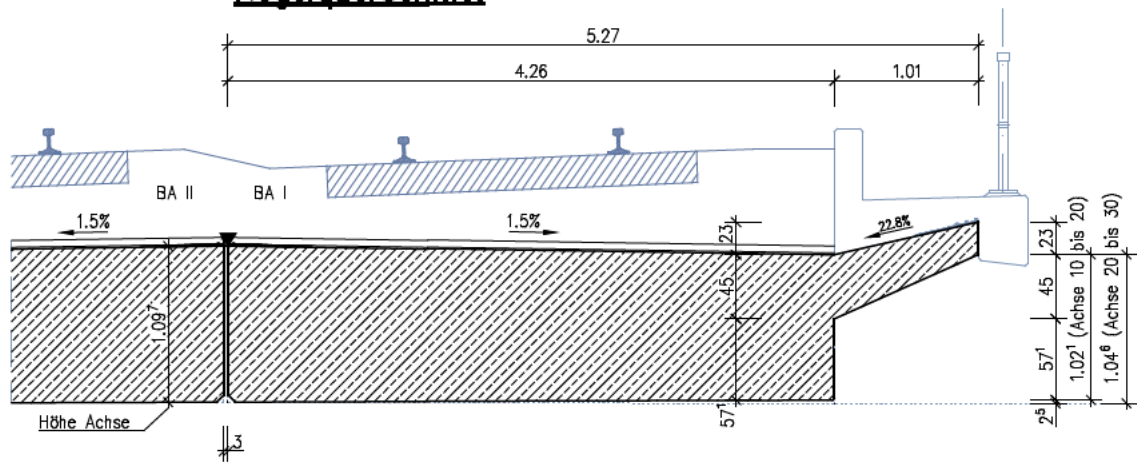
Bauteilabmessungen und Eigenschaften

B 1: Bauteileigenschaften

Bauteil	Elementdicke [m]	Betonfestigkeit
Fahrbahnplatte	1,05	C35/45
Kragarm	0,42	C35/45
Kragarm	0,35	C35/45
Kragarm	0,28	C35/45
Kammerwand Achse 10	1,20	C30/37
Flügel Achse 10	1,00	C30/37
Flügel Voute Achse 10	0,50	C30/37
Fundament Achse 10	0,95	C25/30
Mittelpfeiler Achse 20	1,00	C30/37
Mittelpfeilerfundament Achse 20	0,95	C30/37
Kammerwand Achse 30	1,20	C30/37
Flügel Achse 30	1,00	C30/37
Flügel Voute Achse 30	0,50	C30/37
Fundament Achse 30	0,95	C25/30

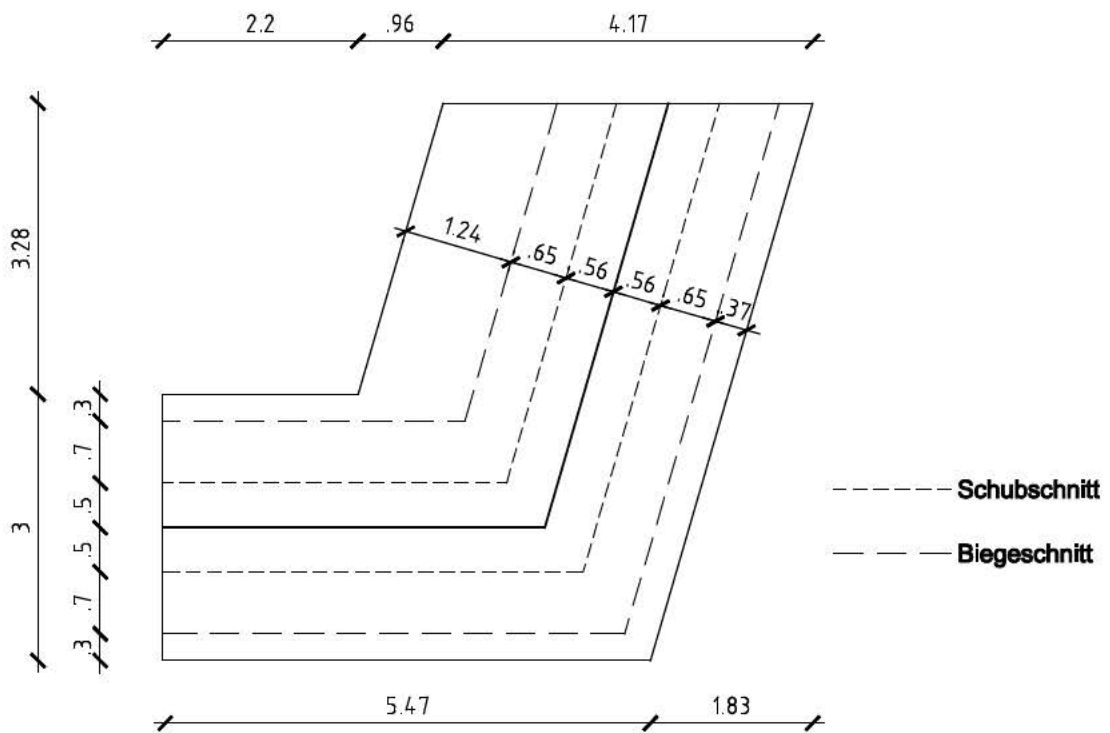
B 4: Regelquerschnitt aus Schalplan

Regelquerschnitt

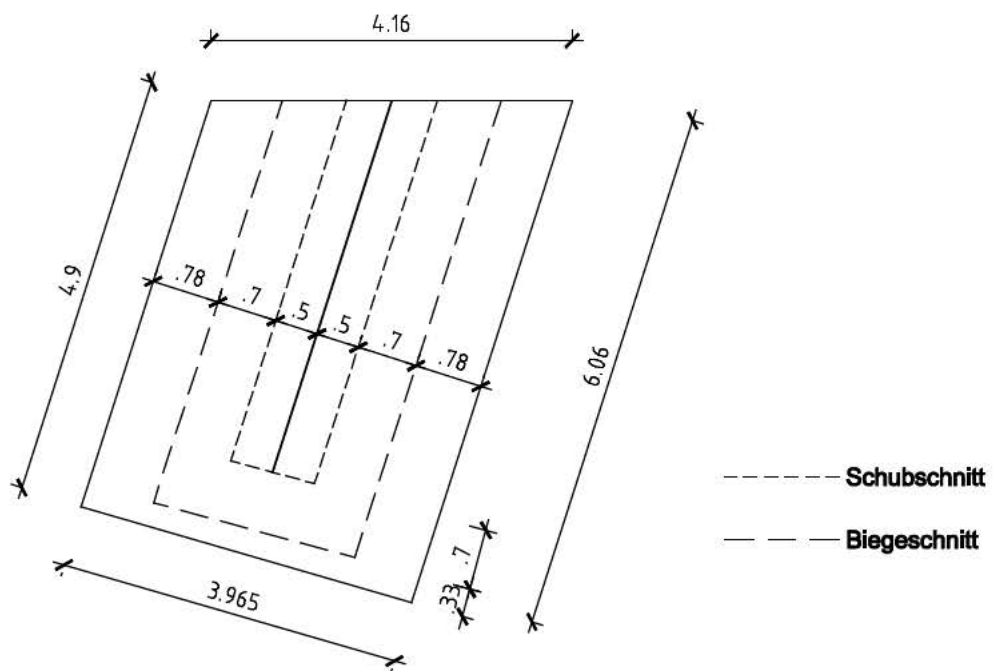


Lage Biegeschnitt und Schubschnitt

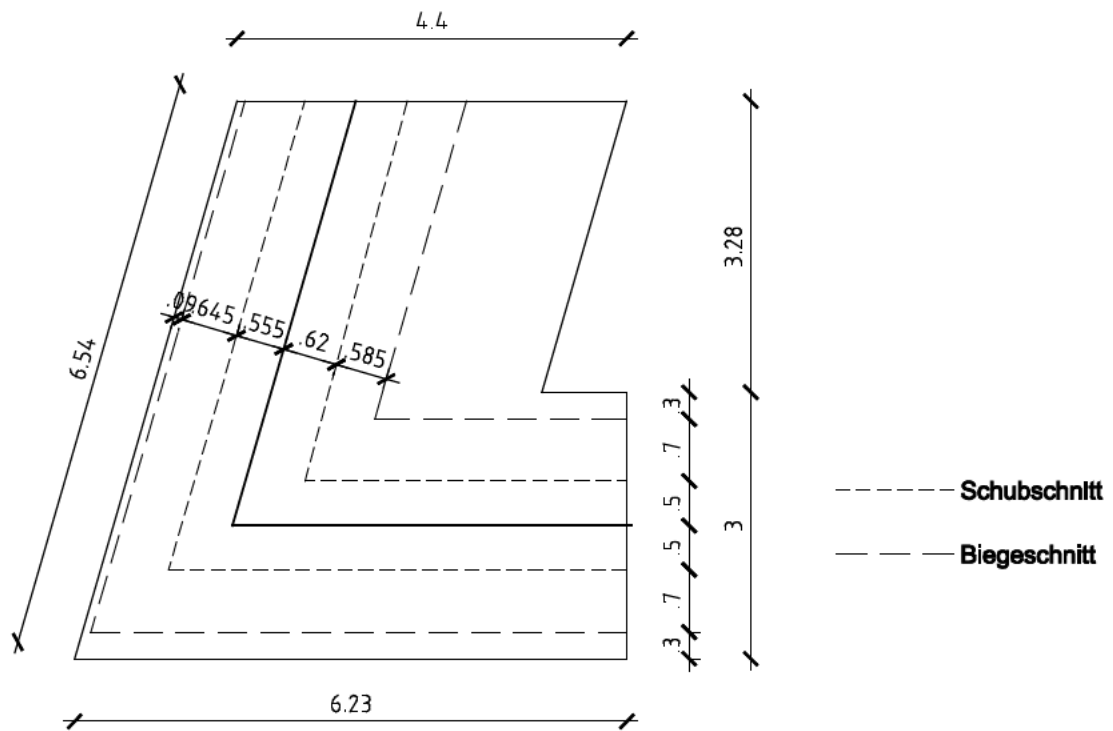
B 5: Ansicht Fundament Achse 10



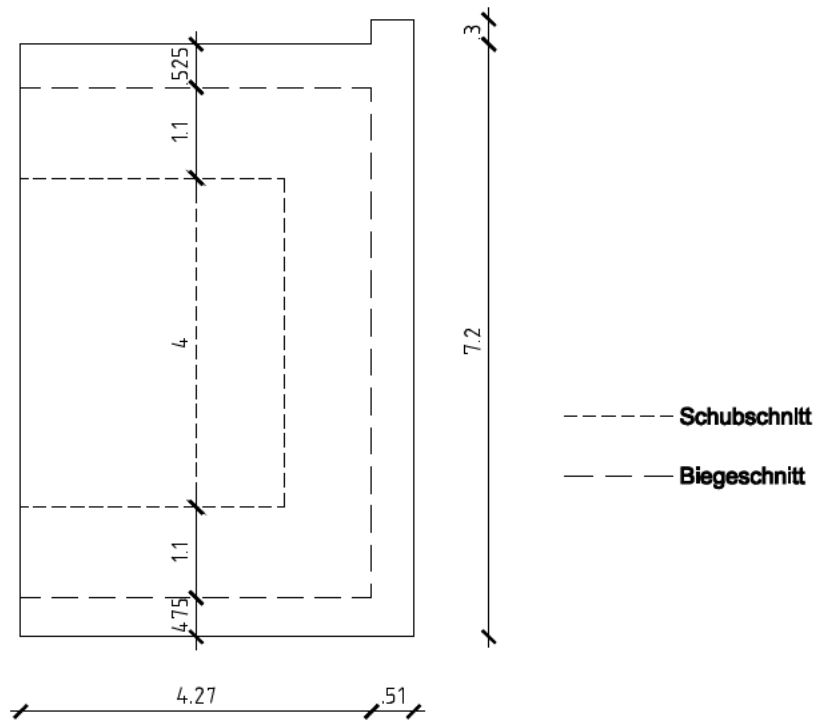
B 6: Ansicht Fundament Achse 20



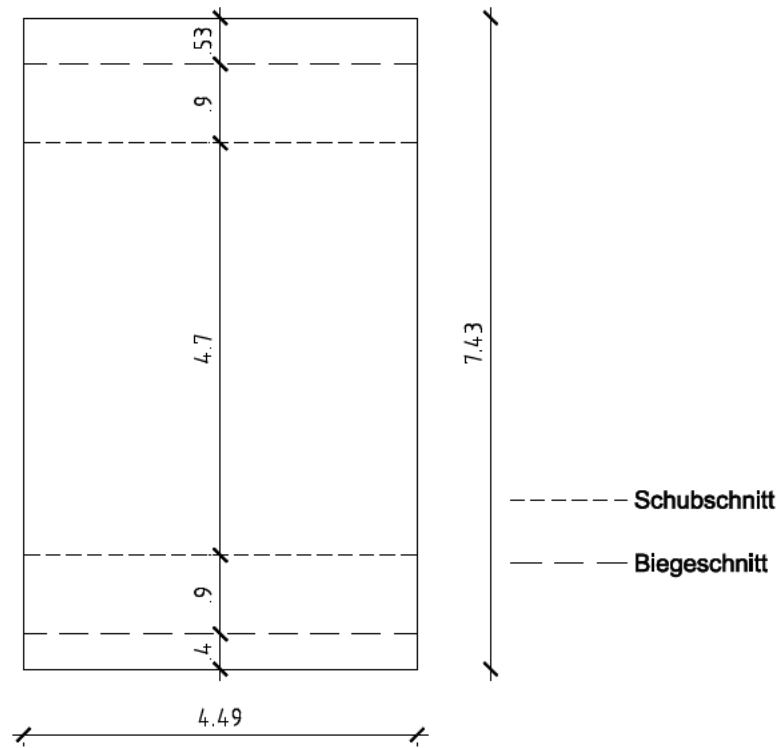
B 7: Ansicht Fundament Achse 30



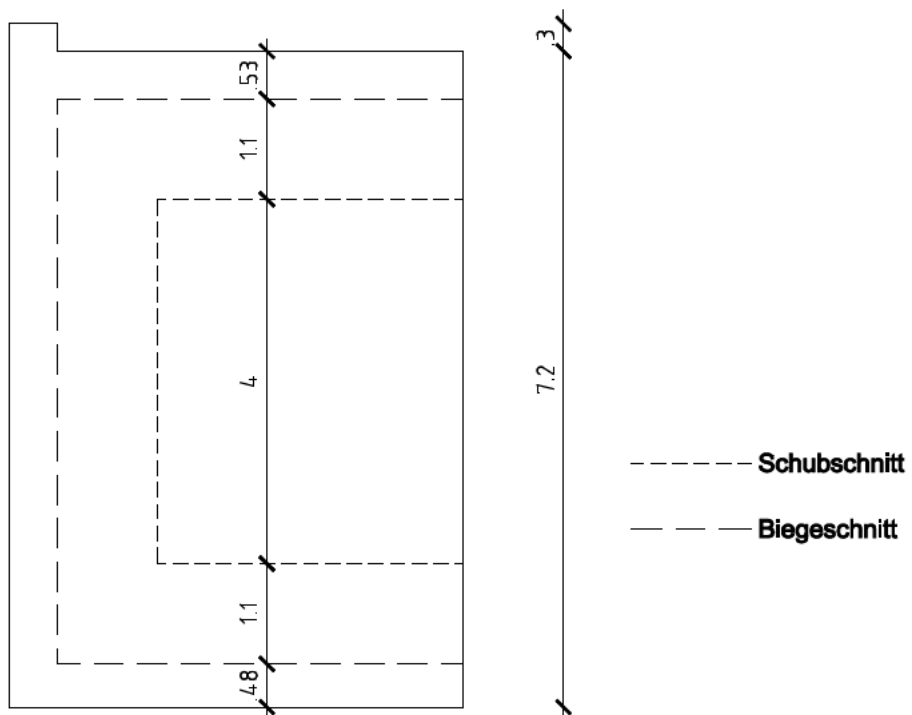
B 8: Ansicht Kammerwand Achse 10



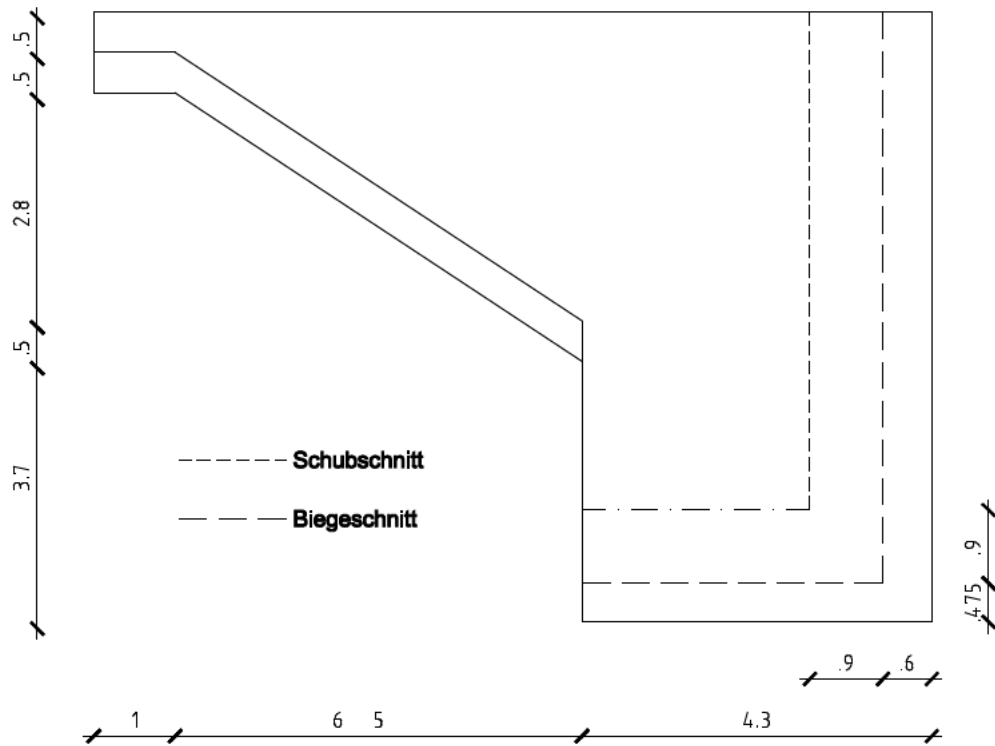
B 9: Ansicht Mittelpfeiler Achse 20



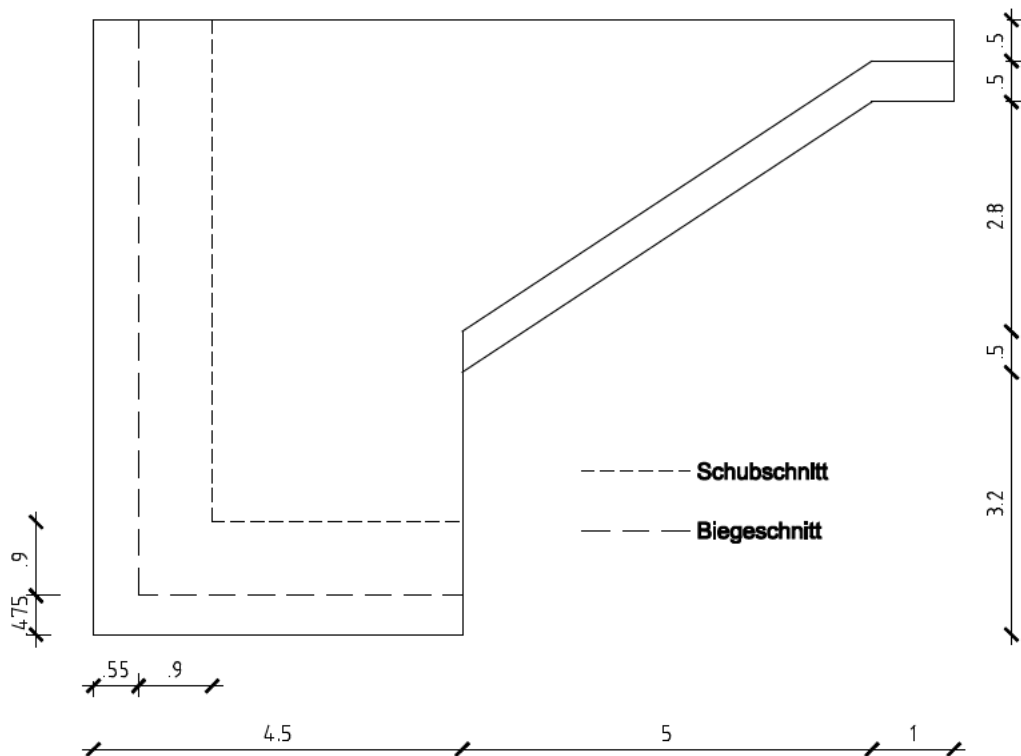
B 10: Ansicht Kammerwand Achse 30



B 11: Ansicht Widerlager Achse 10

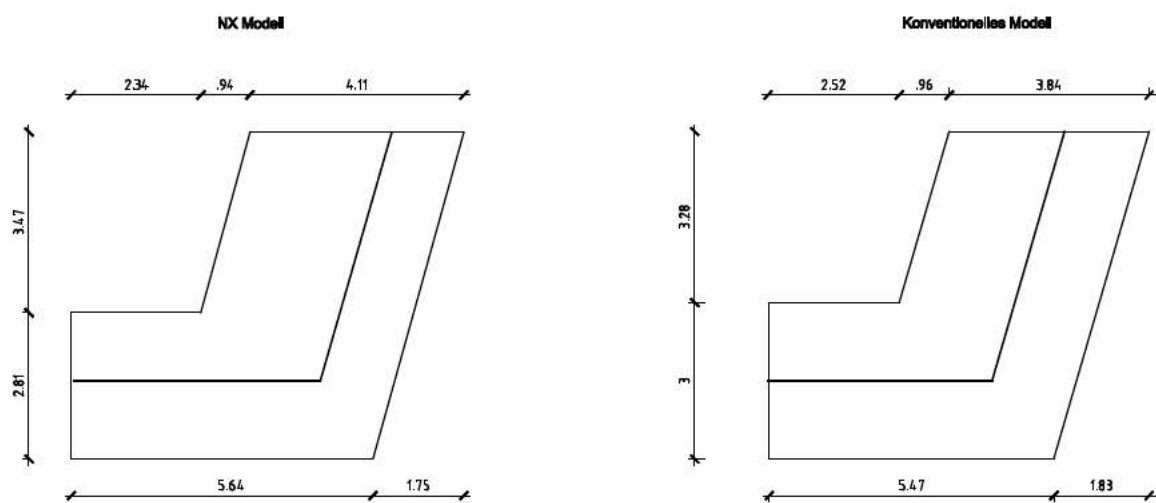


B 12: Ansicht Widerlager Achse 30

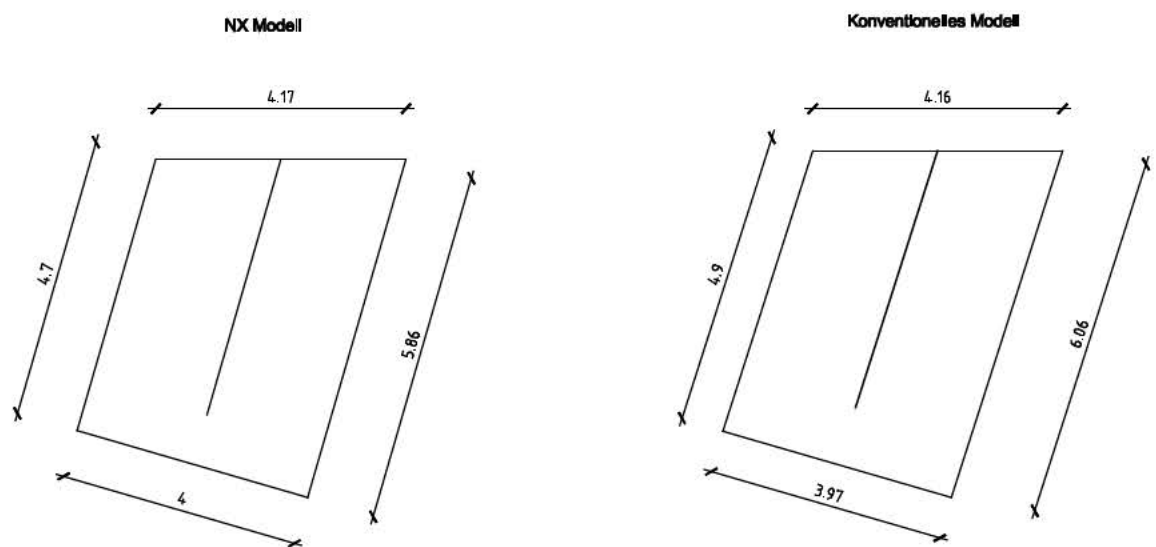


Geometrische Abmessungen im Vergleich

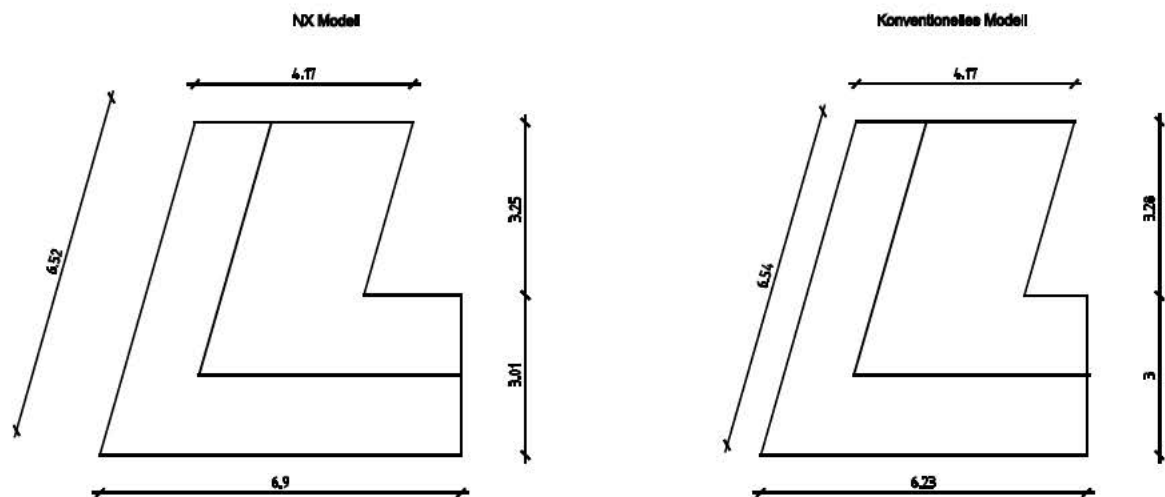
B 13: Abmessungen Fundament Achse 10



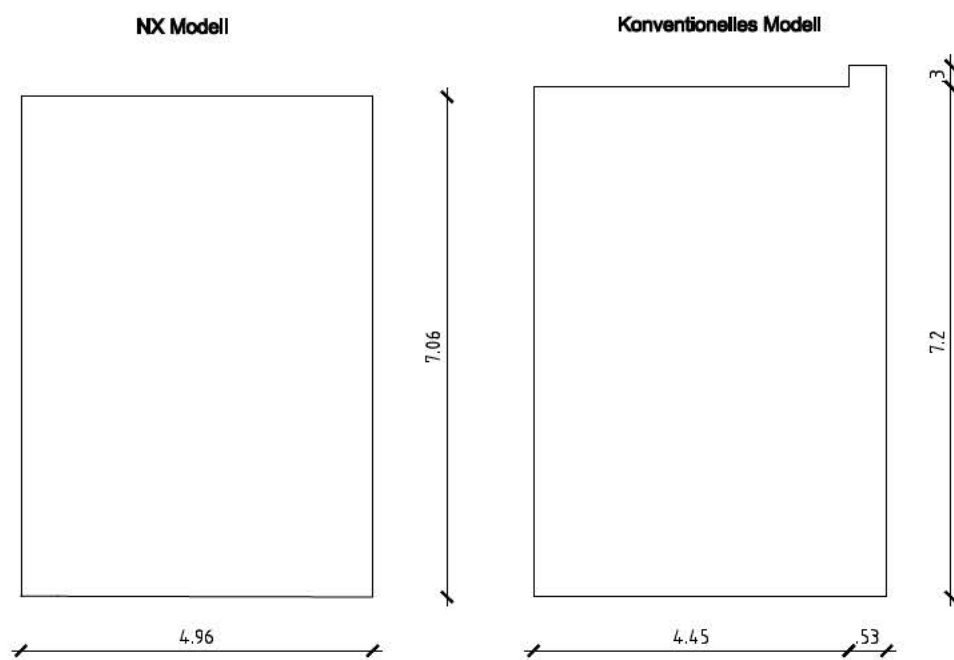
B 14: Abmessungen Fundament Achse 20



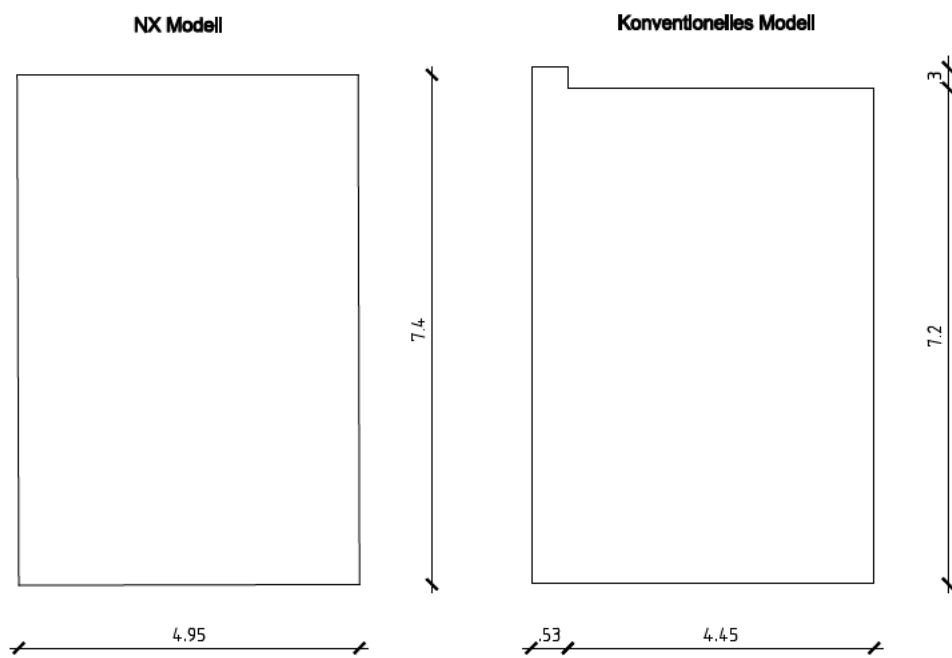
B 15: Abmessungen Fundament Achse 30



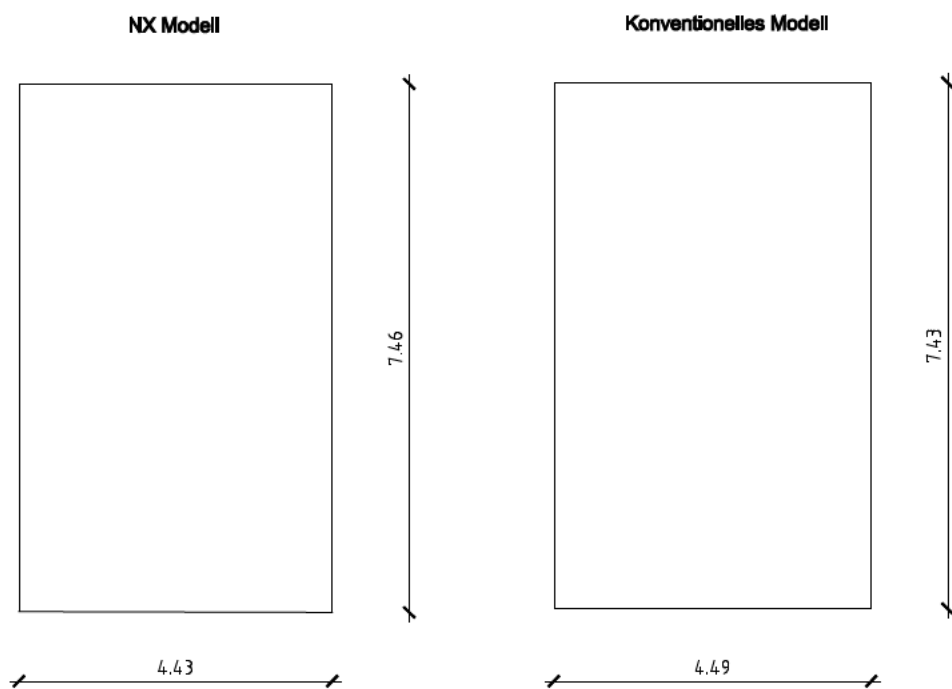
B 16: Abmessungen Kammerwand Achse 10



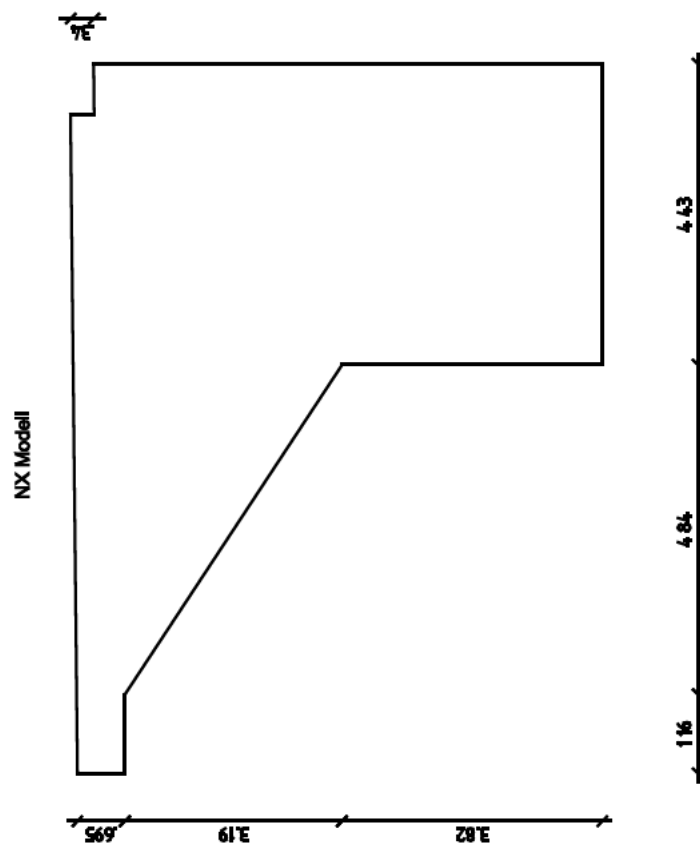
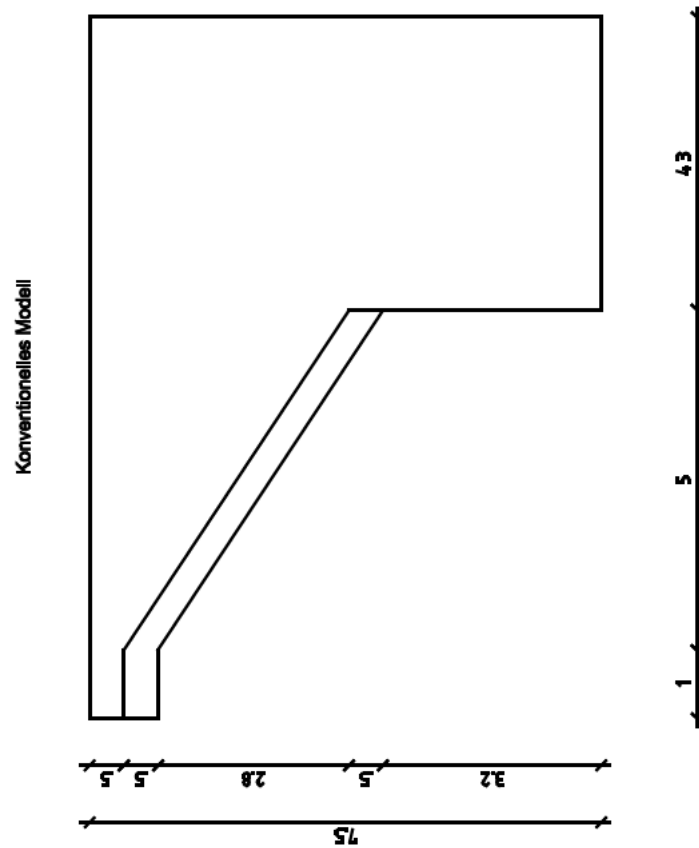
B 17: Abmessungen Kammerwand Achse 30



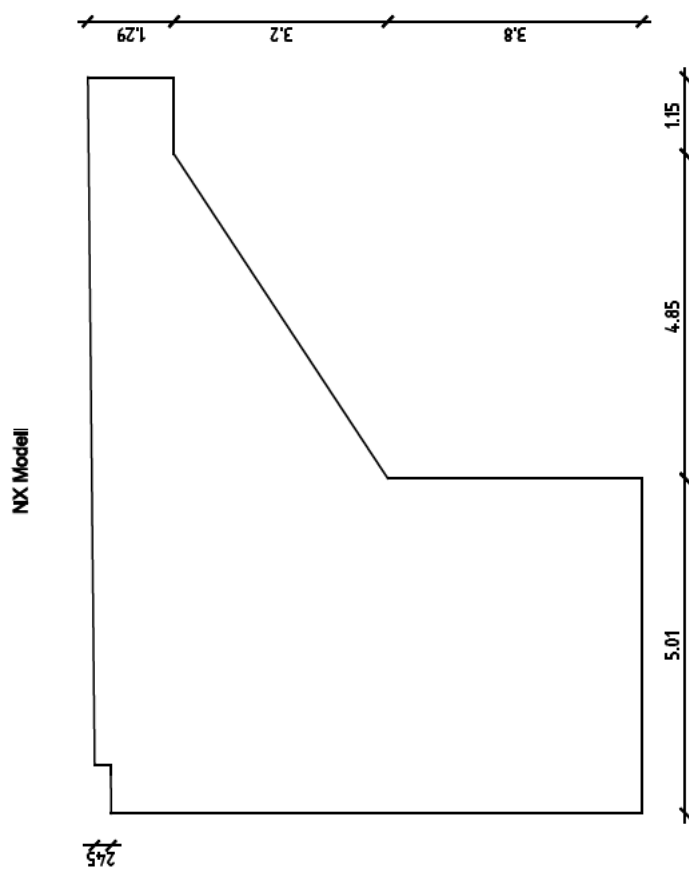
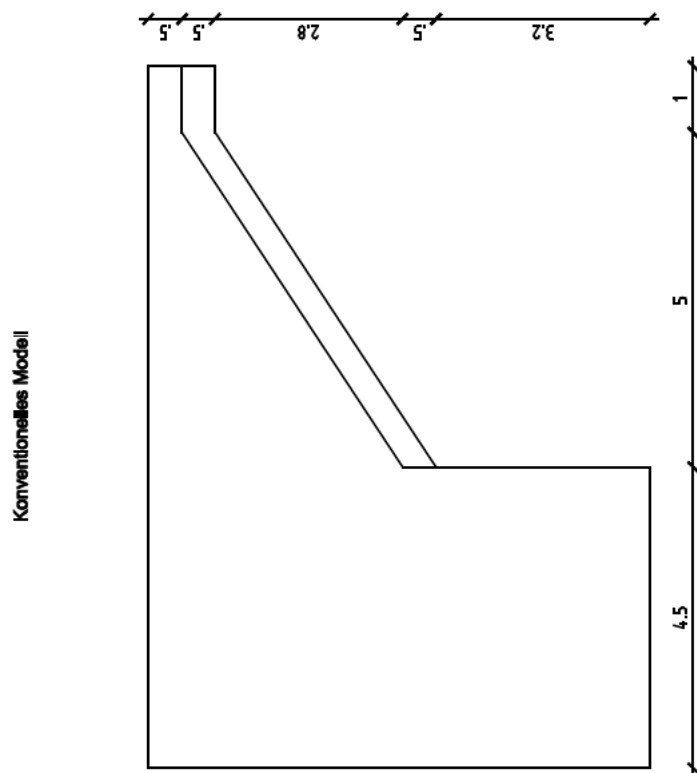
B 18: Abmessungen Mittelpfeiler Achse 20



B 19: Abmessungen Widerlager Achse 10

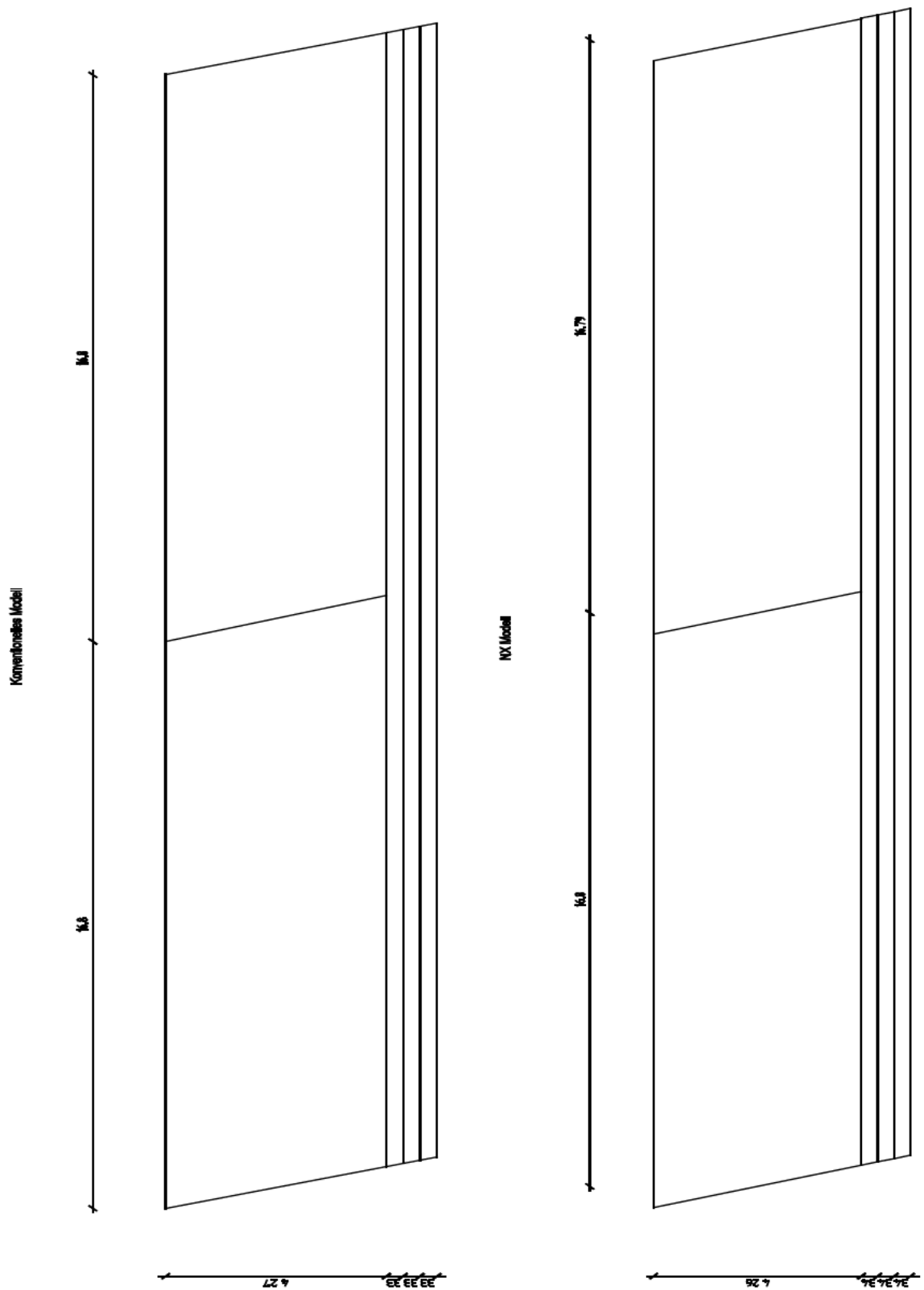


B 20: Abmessungen Widerlager Achse 30



245

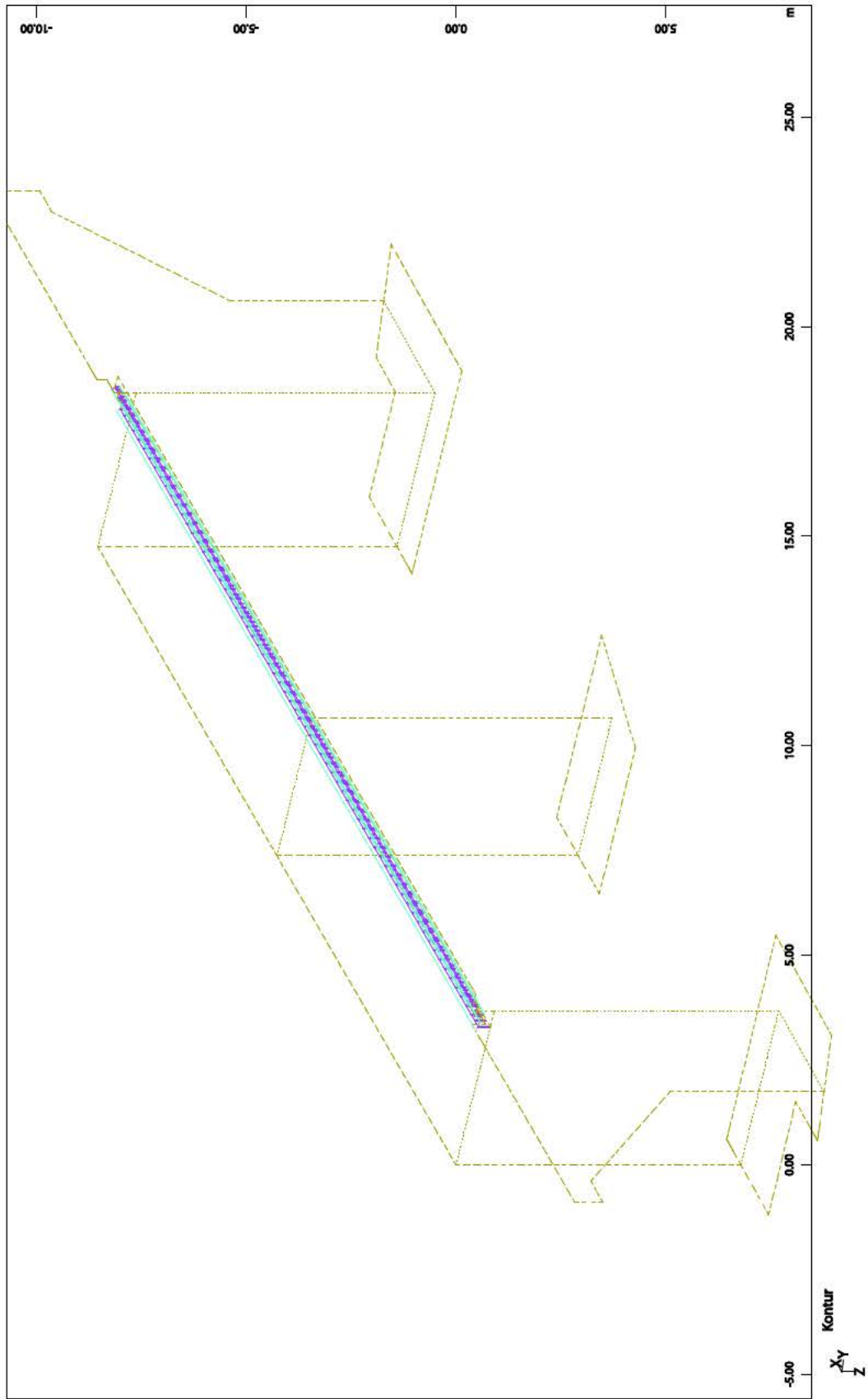
B 21: Abmessungen Fahrbahnplatte



Anhang C

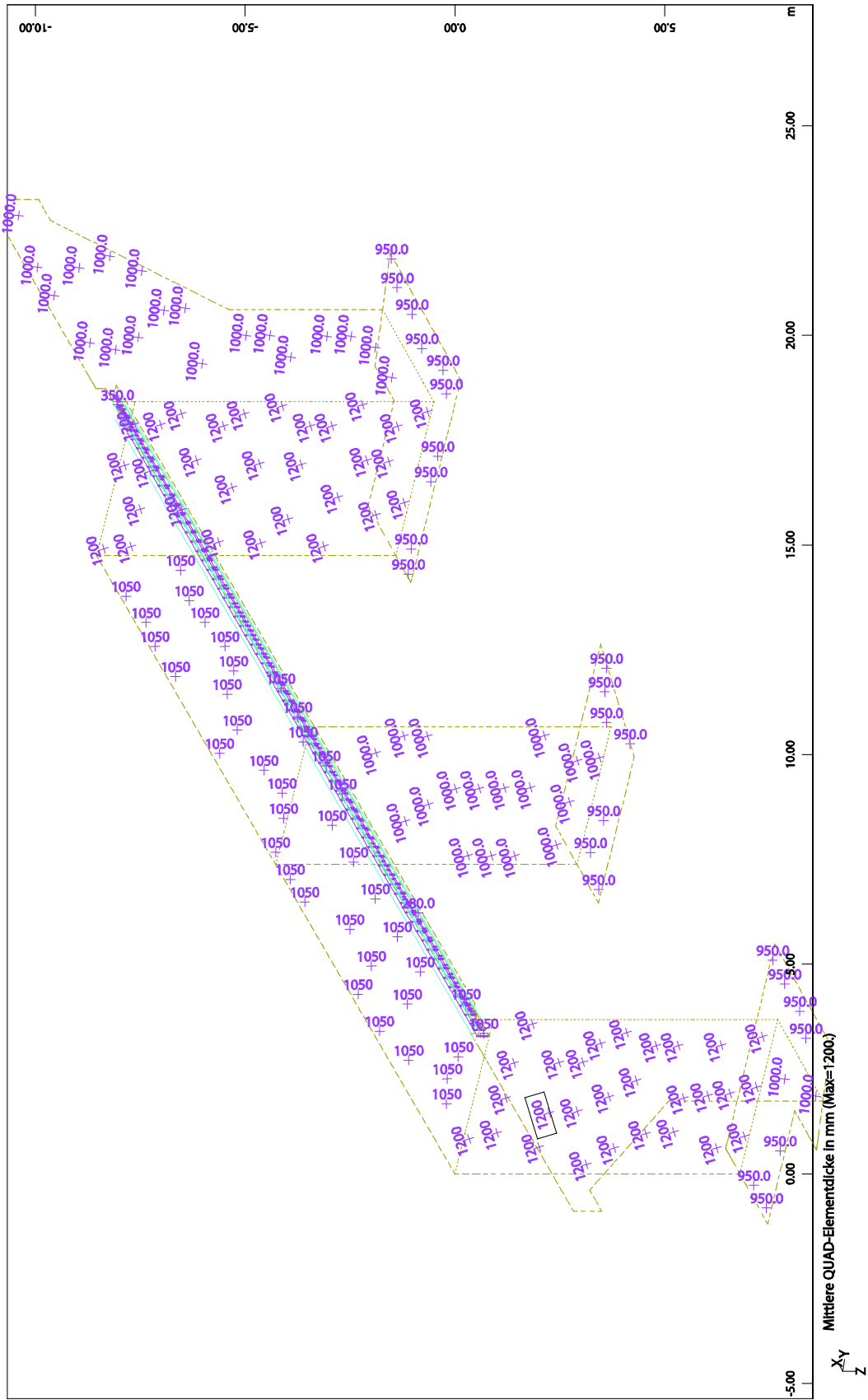
Berechnungsergebnisse SOFiSTiK

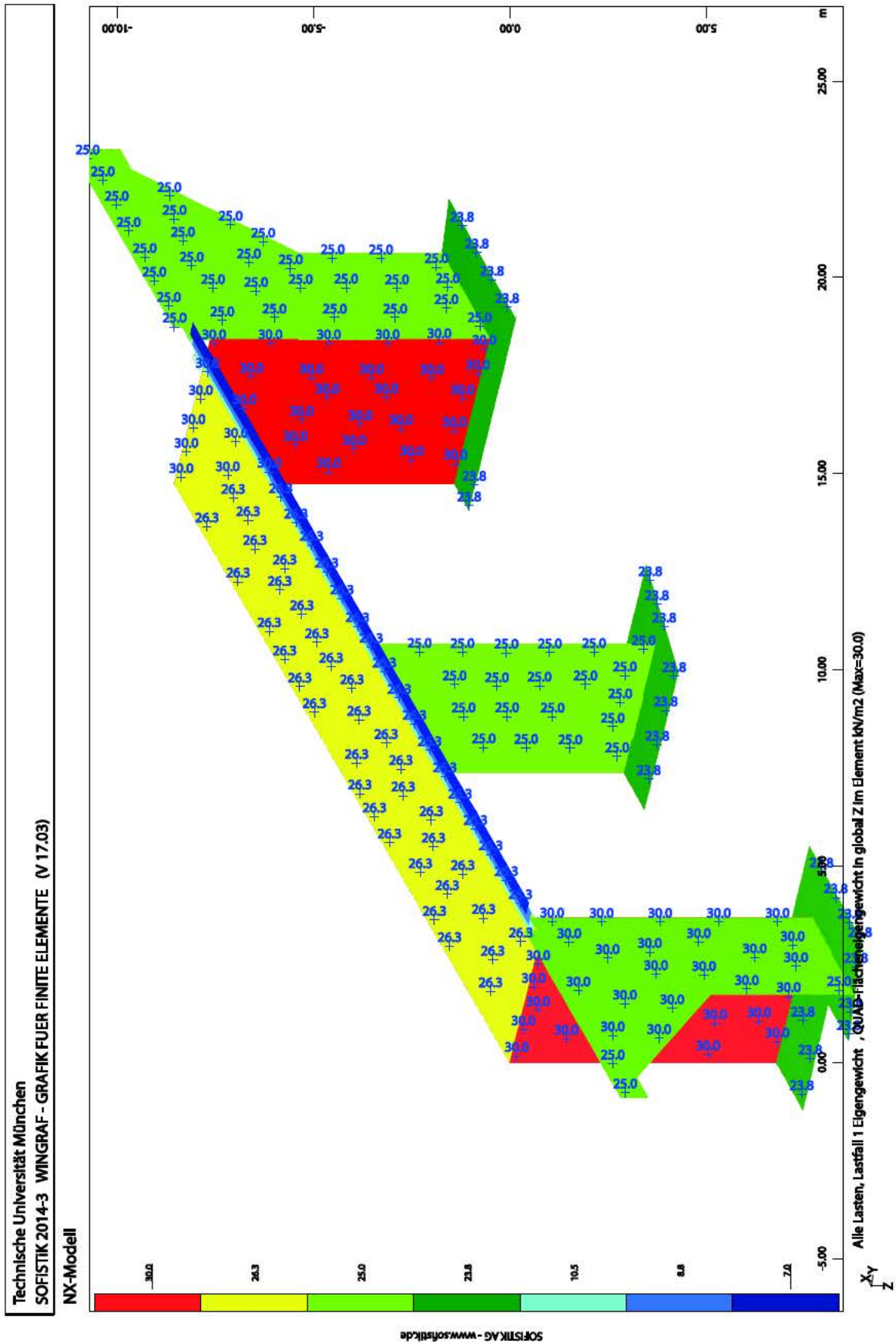
Technische Universität München
SOFISTIK 2014-3 WINGRAF - GRAFIK FUER FINITE ELEMENTE (V 17.03)
NX-Modell



Technische Universität München
SOFISTIK 2014-3 WINGRAF - GRAFIK FUER FINITE ELEMENTE (V 17.03)

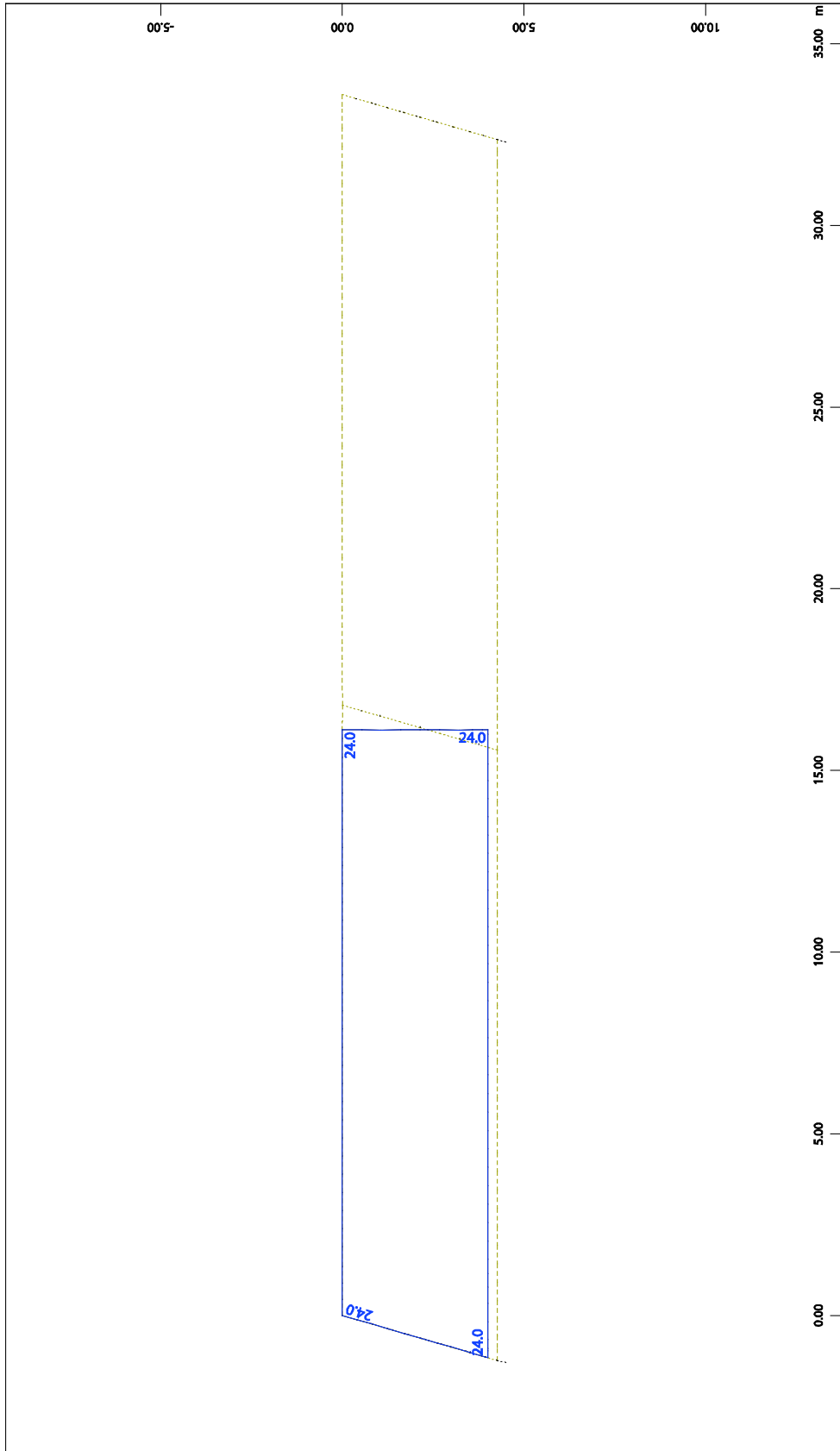
NX-Modell



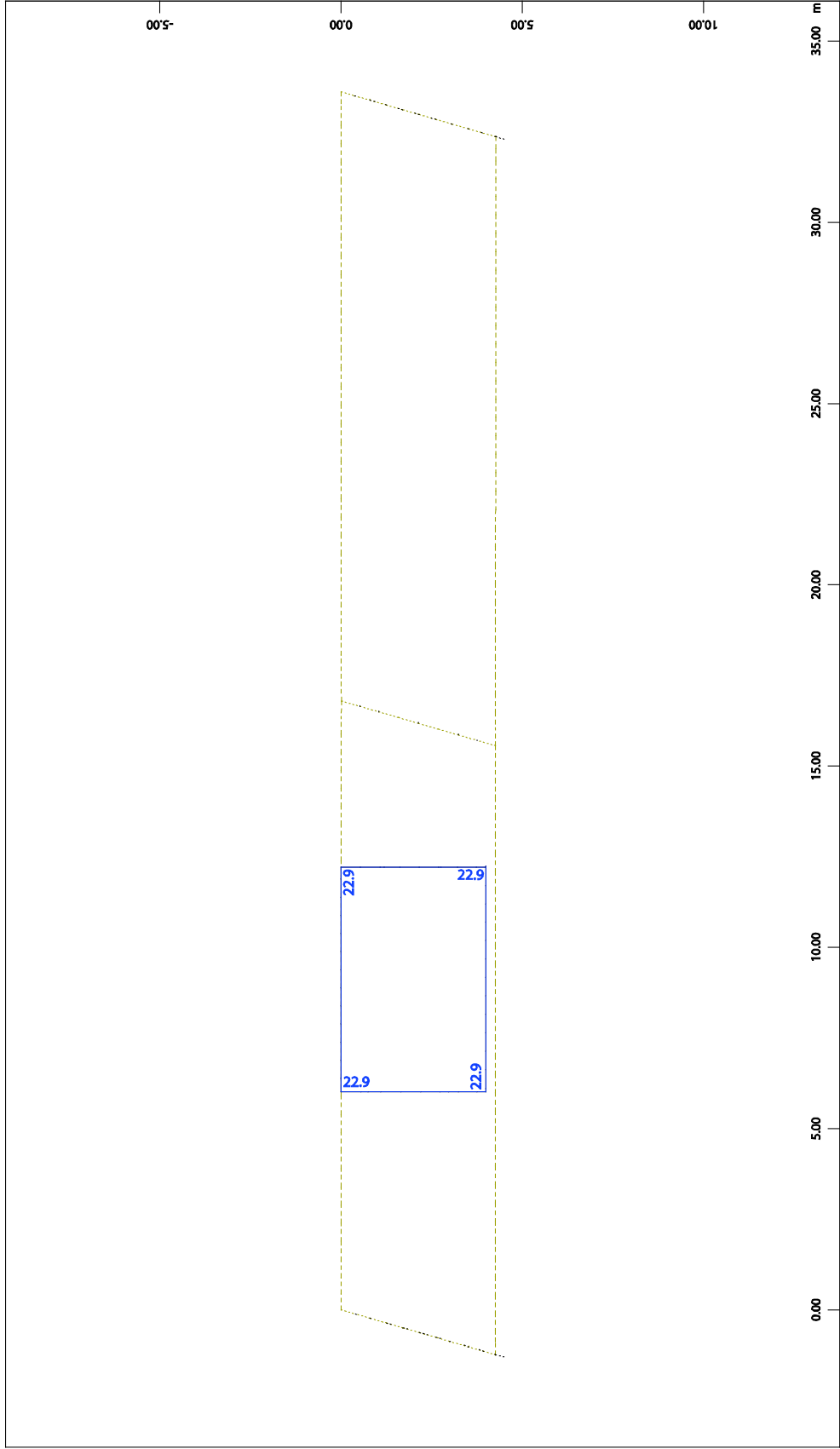


Technische Universität München
SOFISTIK 2014-3 WINGRAF - GRAFIK FUER FINITE ELEMENTE (V17.03)

NX-Modell



Technische Universität München
SOFISTIK 2014-3 WINGRAF - GRAFIK FUER FINITE ELEMENTE (V 17.03)
NX-Modell

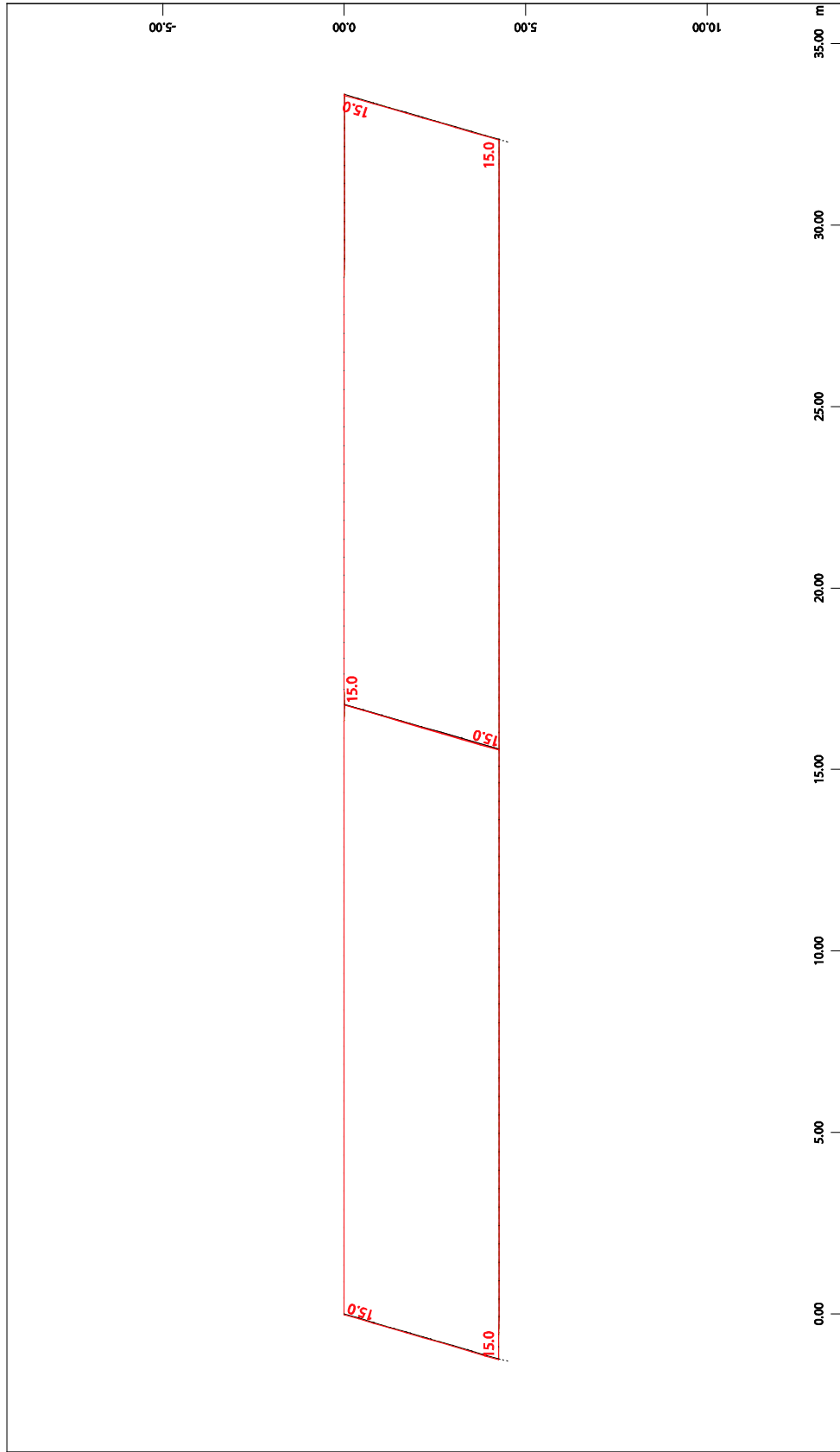


(Max=22.9)

Systemausschnitt Gruppe 1
Alle Lasten, Lastfall 23 Verkehrsast LM 71, (1 cm im Raum = Unit) Freie Flächenlast (Kraft) in global Z (Unit=-20.0 kN/m²)

Z-X
Y

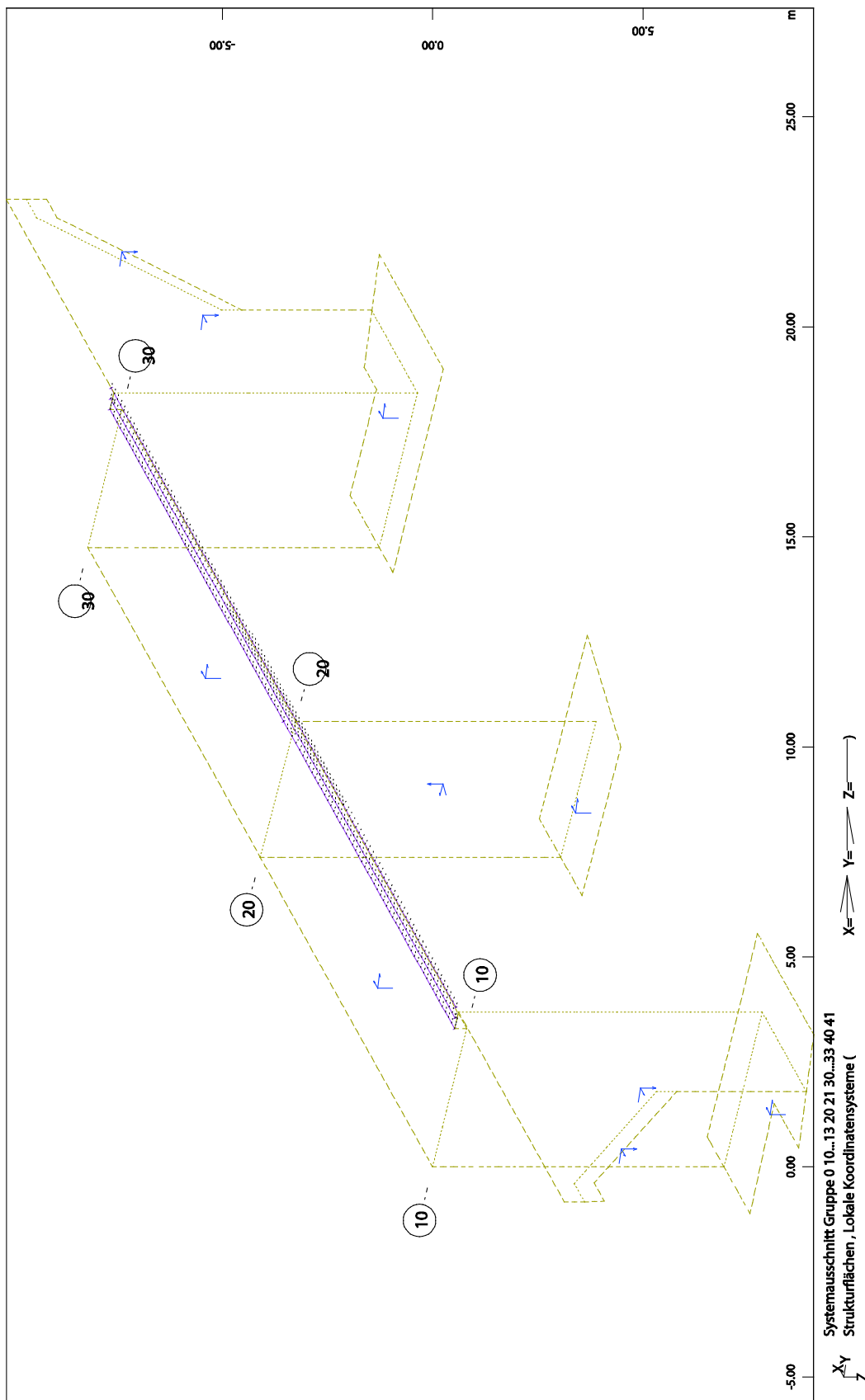
Technische Universität München
SOFISTIK 2014-3 WINGRAF - GRAFIK FUER FINITE ELEMENTE (V17.03)
NX-Modell



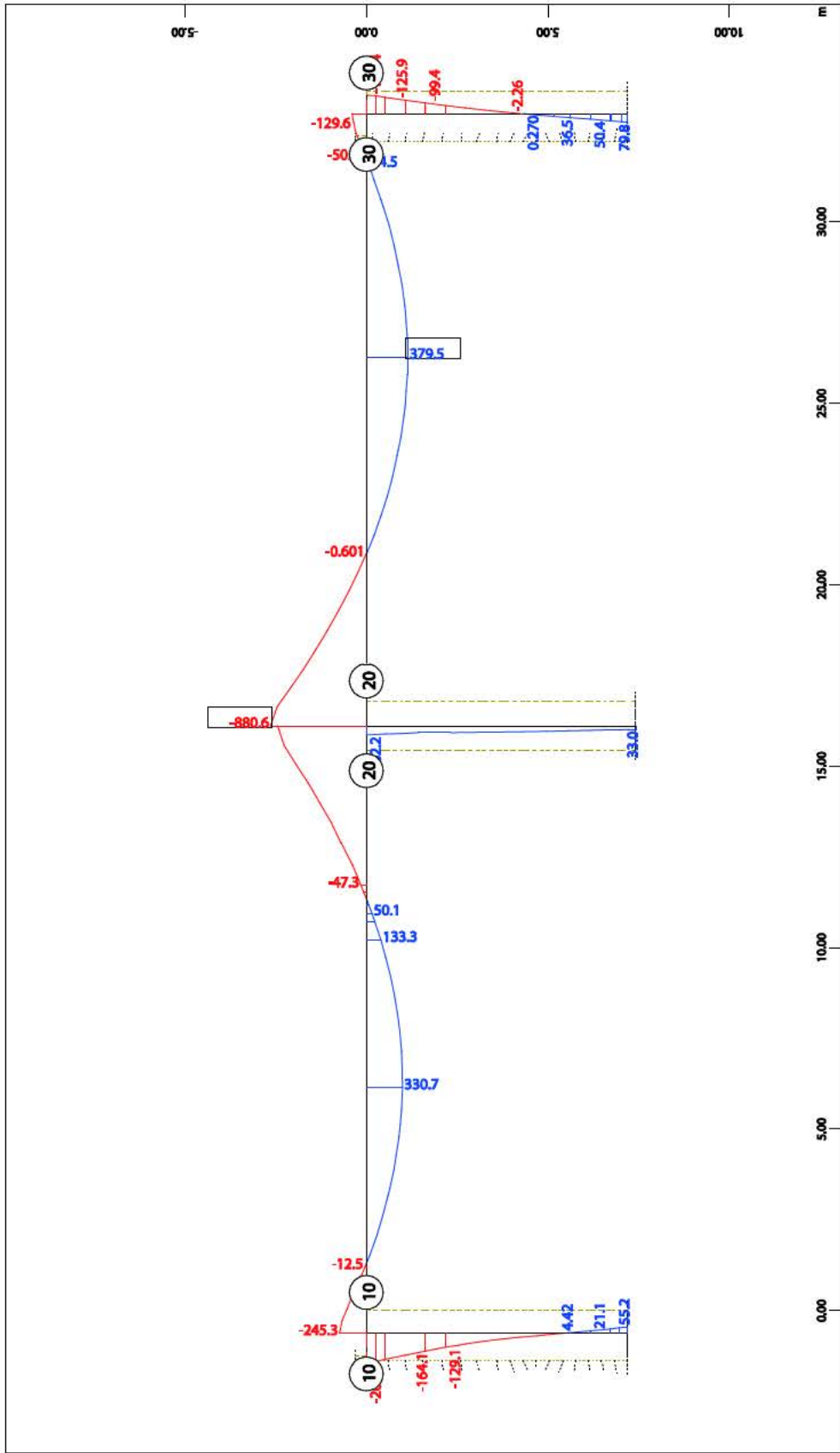
SOFISTIK AG - www.sofistik.de

Systemauschnitt Gruppe 1
Alle Lasten, Lastfall 33 Temperatur , (1 cm im Raum = Unit) Flächenelementlast (Temperaturzuwachs) (Unit=10.0 °C)
(Max=15.0)
(Min=-15.0)

Technische Universität München
SOFISTIK 2014-3 WINGRAF - GRAFIK FUER FINITE ELEMENTE (V17.03)
Konventionelles-Modell



Technische Universität München
 SOFIS TIK 2014-3 WINGRAF - GRAFIK FUER FINITE ELEMENTE (V17.03)
 Konventionelles-Modell

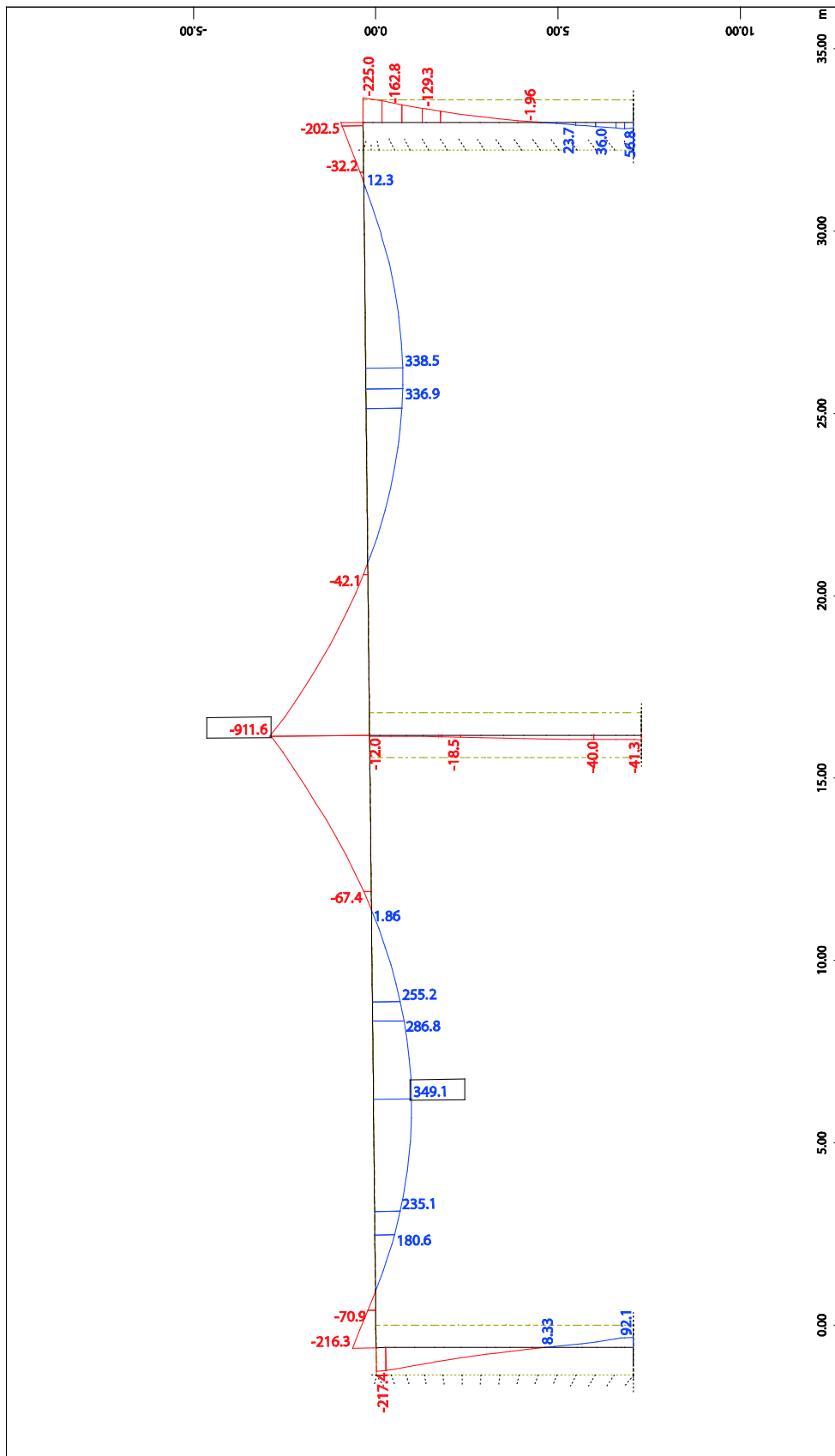


SOFISTIK AG - www.sofistik.de

Systemausschnitt Flächenelemente Gruppe 11 21 31 40 41
 Y-X Biegemoment in Schnittrichtung aus der Elementmitte, Lastfall 1 Eigengewicht, 1 cm im Raum = 500.0 kNm/m (Min=-880.6) (Max=379.5)
 Z

Technische Universität München
SOFISTIK 2014-3 WINGRAF - GRAFIK FUER FINITE ELEMENTE (V 17.03)

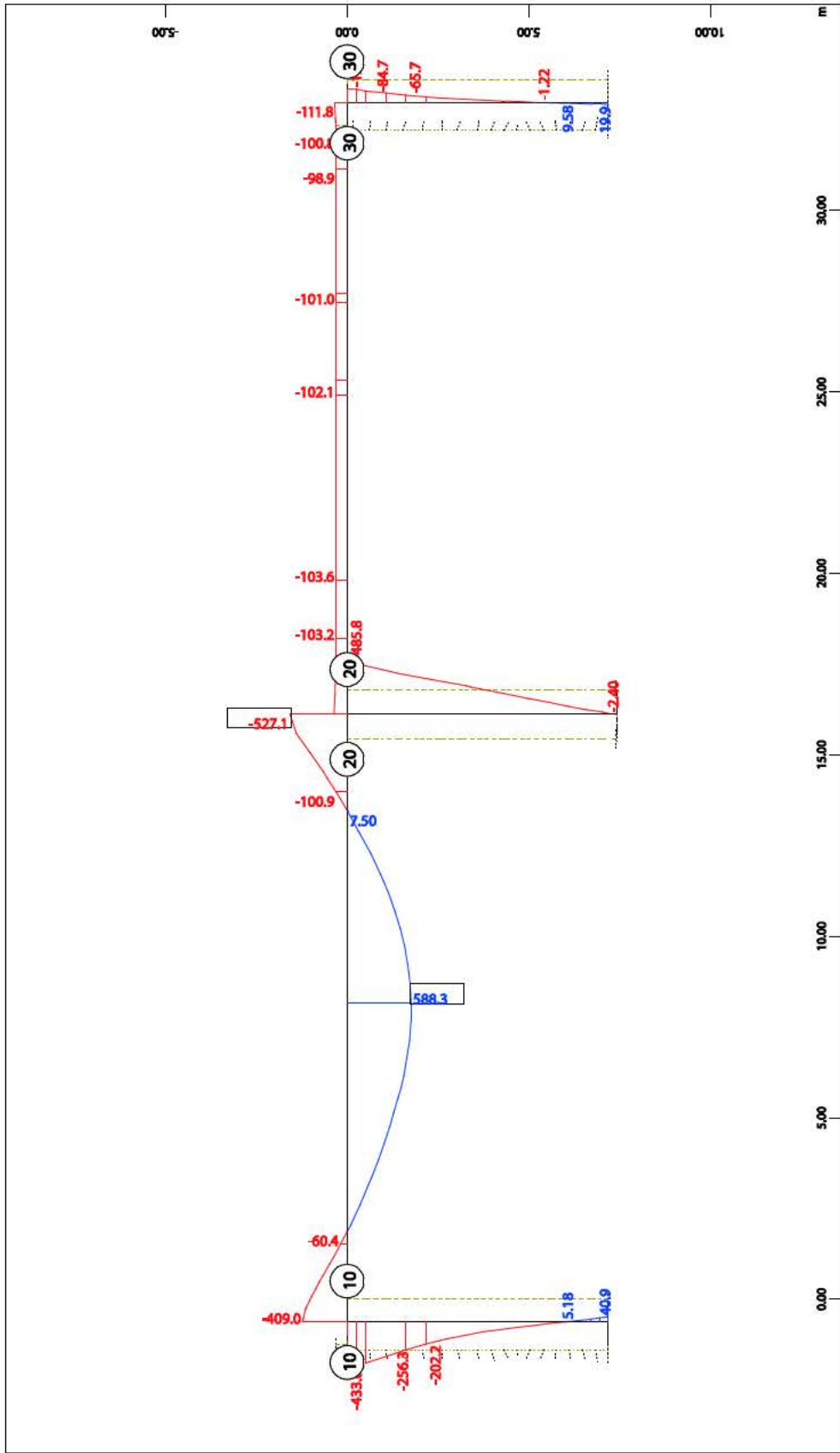
NX-Modell



SOFISTIK AG - www.sofistik.de

Systemausschnitt Flächenelemente Gruppe 1 4 6 10
Biegemoment in Schnittrichtung im Knoten, Lastfall 1 Eigengewicht, 1 cm im Raum = 500.0 kNm/m (Min=-911.6) (Max=349.1)

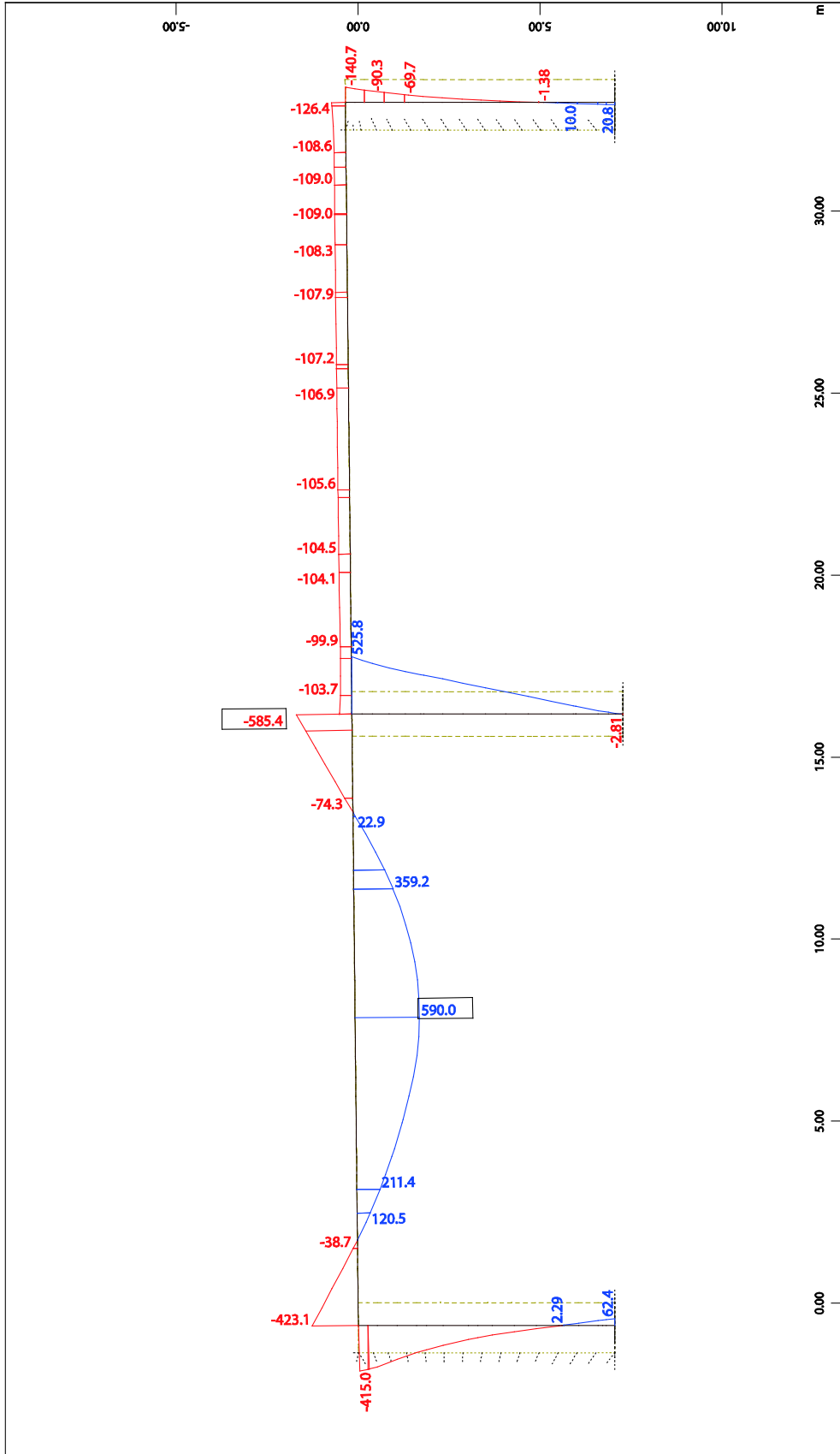
Technische Universität München
 SO FISTIK 2014-3 WINGRAF - GRAFIK FUER FINITE ELEMENTE (V17.03)
 Konventionelles-Modell



SOFSTIK AG - www.sofstik.de

Systemausschnitt Flächenelemente Gruppe 11 21 31 40 41
 Biegemoment in Schnittrichtung aus der Elementmitte, Lastfall 30 LF 11 + LF 23 , 1 cm im Raum = 500.0 kNm/m (Min=-527.1) (Max=588.3)

Technische Universität München
SOFISTIK 2014-3 WINGRAF - GRAFIK FUER FINITE ELEMENTE (V 17.03)
NX-Modell

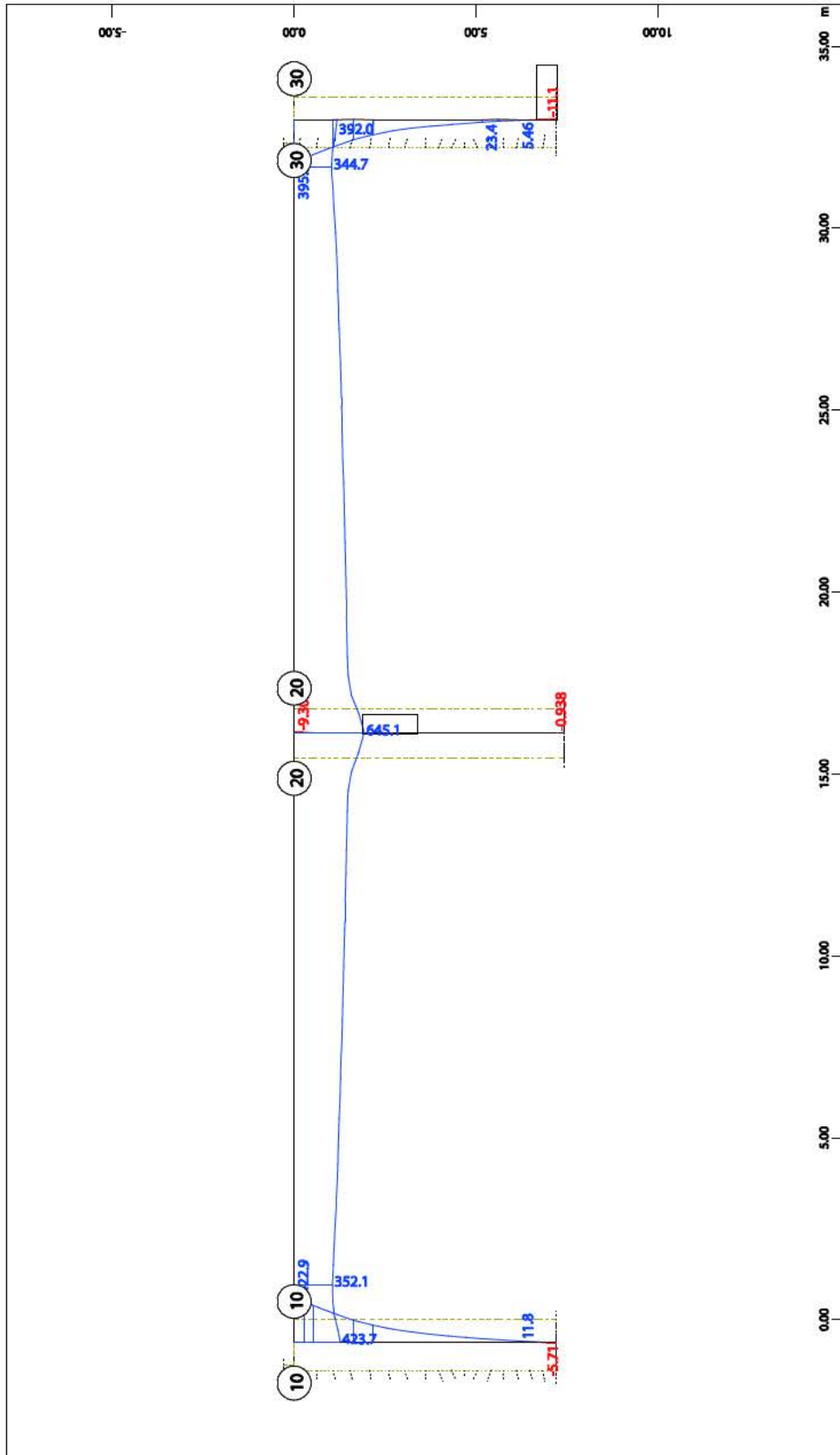


SOFISTIK AG - www.sofistik.de

Systemausschnitt Flächenelemente Gruppe 1 4 6 10
Biegemoment in Schnittrichtung im Knoten, Lastfall 30 LF 11 + LF 23 , 1 cm im Raum = 500.0 kNm/m (Min=-615.4) (Max=590.0)

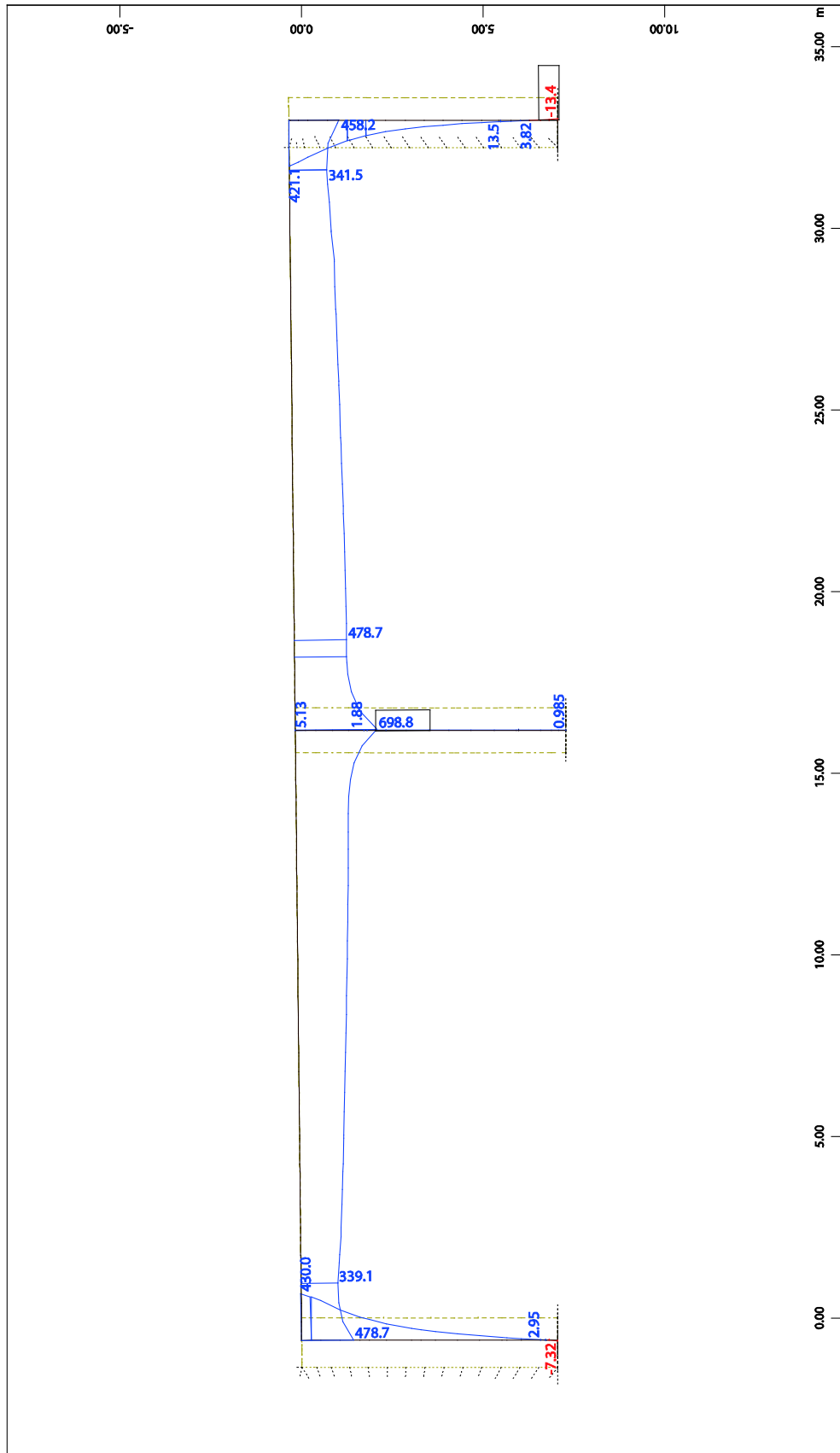
Y-X
Z

Technische Universität München
 SO FISTIK 2014-3 WINGRAF - GRAFIK FUER FINITE ELEMENTE (V 17.03)
 Konventionelles-Modell



Technische Universität München
 SOFISTIK 2014-3 WINGRAF - GRAFIK FUER FINITE ELEMENTE (V17.03)

NX-Modell



Systemausschnitt Flächenelemente Gruppe 1 4 6 10
 Biegemoment in Schnittrichtung im Knoten, Lastfall 33 Temperatur, 1 cm im Raum = 500.0 kNm/m (Min=-13.4) (Max=750.8)

Anhang D

Compact Disc

Auf der beigefügten CD befindet sich folgender Inhalt:

- Der schriftliche Teil der Arbeit als Worddokument und als PDF
- Das Bauwerkinformationsmodell des Projekts *Eisenbahnüberführung Ventschow* (Dateiformate .prt und .jt)
- Die verwendete JT-Konfigurationseinstellungen
- Die Schnittstellenprogramme *NX2Sofistik_Vol* und *NX2Sofistik_Area*
- Die UML-Klassendiagramme als PDF
- Die verwendeten Grafiken

Abbildungsverzeichnis

Abbildung 1: BIM-Vergleich mit konventionellen Planungsverfahren nach (Tonke, 1999) ..	7
Abbildung 2: Anzahl benötigter Direktschnittstellen (Friedewald, et al., 2011).....	9
Abbildung 3: Anzahl benötigter Neutralschnittstellen (Friedewald, et al., 2011).....	10
Abbildung 4: IfcDoor (Tür) beschrieben mit dem EXPRESS-Schema (buildingSMART, 2014).....	11
Abbildung 5: Klassifizierung der BIM Anwendung (buildingSMART, 2014)	12
Abbildung 6: Normzellenschema (Bungartz, et al., 1996).....	15
Abbildung 7: zweidimensionaler Quadtree (Encarnaçao, et al., 1990).....	15
Abbildung 8: CSG als Binärbaum nach (Frank, 2000).....	16
Abbildung 9: Mehrdeutigkeit beim Drahtmodell (Bungartz, et al., 1996)	17
Abbildung 10: Topologische und geometrische Hierarchie nach (Sorger, 2012)	17
Abbildung 11: Oberflächendarstellung nach (Encarnaçao, et al., 1990).....	18
Abbildung 12: Facettenmodell eines Quaders	19
Abbildung 13: Extrusion der Mittelfläche (Sorger, 2012)	19
Abbildung 14: von NX generierte Mittelfläche (links) und gekoppelte Mittelfläche (rechts)	24
Abbildung 15: Baugruppenstruktur in Siemens NX nach (Weinholzer, 2014)	27
Abbildung 16: Bauwerkstruktur einer Brücke nach (Wang, 2012)	28
Abbildung 17: Ressourcenliste mit geöffnetem Teile-Navigator, in Anlehnung an (Hochmuth, 2012).....	29
Abbildung 18: Rahmenbrücke in Siemens NX	30
Abbildung 19: Auswahl der relevanten Komponenten für FEM-Export.....	33
Abbildung 20: oberer und unterer Teilkörper des Fundaments (Wang, 2012).....	35
Abbildung 21: erzeugtes Flächenmodell in Siemens NX	36
Abbildung 22: halber Brückenquerschnitt (blau) mit Mittelfläche (gelb).....	36
Abbildung 23: nicht verbundene Mittelflächen (links) und verbundene Mittelflächen (rechts)	37
Abbildung 24: SOFiSTiK Structural Desktop (SOFiSTiK, 2014).....	39
Abbildung 25: AQUA-Modul im TEDDY Editor	40
Abbildung 26: Systemgenerierung unter SOFIMSHC	41

Abbildung 27: Variablenzuweisung	42
Abbildung 28: Kopplung der Teilflächen	43
Abbildung 29: Lastmodell 71 für Eisenbahnbrücken (DIN e.V, 2013)	44
Abbildung 30: Flächenlast $Q'_{v,k}$ bzw. $q'_{v,k}$	44
Abbildung 31: vereinfachter Ansatz des Temperaturlastfalls.....	45
Abbildung 32: Ausschnitt des UML-Diagramms JT-API nach (Weinholzer, 2014).....	48
Abbildung 33: Ausschnitt des UML-Diagramms die Klassen Vertex, Face und BuildingPart nach (Weinholzer, 2014).....	49
Abbildung 34: Ausschnitt der Klasse LoadJT: Generieren der Dreiecksflächen	49
Abbildung 35: Ausschnitt der Klasse WriteDatFile: Zuordnung der Materialien	51
Abbildung 36: Ausschnitt der Klasse WriteDatFile: Jedes Bauteil wird in einem SOFIMSHC Modul generiert	51
Abbildung 37: Entfernen doppelter Strukturlinien (rot).....	53
Abbildung 38: Ausschnitt des UML-Diagramms die Klassen BuildingPartBuffer, OutputBuffer und WriteDatFile	53
Abbildung 39: Verarbeitung der Dreiecksflächen.....	54
Abbildung 40: Widerlager Achse 10 mit Fundament als Volumenmodell in SOFiSTiK.....	57
Abbildung 41: LM 71 Laststellung in Feldmitte (LF30).....	59
Abbildung 42: Längsschnitt A-A durch Flächentragwerk.....	60
Abbildung 43: gemittelte Fläche bei gevouteten Bauteilen.....	62

Tabellenverzeichnis

Tabelle 1: Lastkombinationen und Lastfälle	46
Tabelle 2: Biegemoment m_{-xx} [kNm], Eigengewicht (LF1)	60
Tabelle 3: Biegemoment m_{-xx} [kNm], LM 71 (LF30)	60
Tabelle 4: Biegemoment m_{-xx} [kNm], Temperatur (LF33)	60

Abkürzungsverzeichnis

2D	zweidimensional
3D	dreidimensional
API	Application Programming Interface
BIM	Building Information Modeling
B-REP	Boundary Representation
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CSG	Constructive Solid Geometry
DAT	Data File
DFG	Deutsche Forschungsgemeinschaft
DGM	Digitales Geländemodell
DIN	Deutsches Institut für Normung e. V.
DWG	Drawing
DXF	Drawing Interchange File Format
FEM	Finite Elemente Methode
IAI	International Alliance for Interoperability
IDE	Integrated Development Environment
IFC	Industry Foundation Classes
IGES	Initial Graphics Exchange Specification
ISO	Internationale Organisation für Normung
IT	Informationstechnik
JT	Jupiter Tessellation
LF	Lastfall
LM	Lastmodell
LOD	Level of Detail
NPA	Anfangsknoten

NPE	Endknoten
PDF	Portable Document Format
RGB	Rot, Grün, Blau
SAR	Structure Activity Relationships
SLN	Strukturlinie
SSD	Sofistik Structural Desktop
SSP	Schwerpunktprogramm
STEP	Standard for the exchange of product model data
UML	Unified Modeling Language
VRML	Virtual Reality Modeling Language
WCS	Work Coordinate System

Literaturverzeichnis

Born Günter Referenzhandbuch Dateiformate - Grafik, Text, Datenbanken, Tabellenkalkulation [Buch]. - Bonn : Addison-Wesley, 1996.

buildingSMART buildingSMART [Online] // buildingSMART. - 2014. - 27. September 2015. - <http://www.buildingsmart.de/>.

Bungartz Hans-Joachim, Griebel Michael und Zenger Christoph Einführung in die Computergraphik [Buch]. - Braunschweig/Wiesbaden : Vieweg & Sohn Verlagsgesellschaft, 1996.

DIN e.V Handbuch Eurocode 1 - Einwirkungen Band 3: Brückenlasten [Buch]. - Berlin : Beuth, 2013.

Eastmann Chuck [et al.] BIM Handbook [Buch]. - New Jersey : John Wiley & Sons, 2011.

Encarnação José und Barth Wilhelm Geometrische Verfahren der graphischen Datenverarbeitung [Buch]. - Berlin : Springer, 1990.

Frank Anton Christian Organisationsprinzipien zur Integration von geometrischer Modellierung, numerischer Simulation und Visualisierung [Buch]. - München : Utz Wiss, 2000.

Friedewald Axel [et al.] Benchmark neutraler Formate für den prozessübergreifenden Datenaustausch im Schiffbau [Bericht]. - [s.l.] : Fraunhofer IGD, 2011.

Günthner Willibald und Borrmann André Digitale Baustelle- innovativer Planen, effizienter Ausführen [Buch]. - Heidelberg : Springer-Verlag, 2011.

Hartmann Dietrich Objektorientierte Modellierung in Planung und Konstruktion [Buch]. - Weinheim : Wiley-VCH, 2000.

Hochmuth Markus 3D Planung im Ingenieurbau- Handout zu NX [Bericht] : firmeninternes Dokument. - München : OBERMEYER Planen + Beraten GmbH, 2012.

Jernigan Finith Big BIM, little bim [Buch]. - Salisbury : 4Site Press, 2008.

- Ji Yang, Obergrießer Mathias und Borrmann André** Prototypische Entwicklung IFC-Bridge-basierter Anwendung [Bericht]. - [s.l.] : Technische Universität München, Hochschule Regensburg, 2010.
- Klette Guido, Nulsch Michael und Vajna Sándor** NX7.5 - kurz und bündig [Buch]. - Wiesbaden : Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH Wiesbaden, 2011.
- Kuloyants Valery** Entwicklung eines IFC-basierenden Datenaustauschstandards für den Unterbau von Brückenbauwerken [Buch]. - München : Technische Universität München, 2014.
- Liebich Thomas, Schweer Carl-Stephan und Wernik Siegfried** Die Auswirkungen von Building Information Modeling (BIM) auf die Leistungsbilder und Vergütungsstruktur für Architekten und Ingenieure sowie auf die Vertragsgestaltung [Bericht]. - [s.l.] : Bundesinstitut für Bau-, Stadt- und Raumforschung (BBSR) im Bundesamt für Bauwesen und Raumentwicklung (BBR), 2011.
- Niggel Andreas Konrad** Tragwerksanalyse am volumenorientierten Gesamtmodell -Ein Ansatz zur Verbesserung der computergestützten Zusammenarbeit im konstruktiven Ingenieurbau [Buch]. - München : Universitätsbibliothek der TU München, 2008.
- Raida Johannes** johannes-raida.de [Online]. - 2015. - 17. Juli 2015. - <http://www.johannes-raida.de/jnetcad.htm>.
- Romberg Richard** Gebäudemodell-basierte Strukturanalyse im Bauwesen [Buch]. - München : Universitätsbibliothek der TU München, 2007.
- Rüppel Uwe** Vernetzt-kooperative Planungsprozesse im Konstruktiven Ingenieurbau [Buch]. - Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2007.
- Sendler Ulrich** 3D wird Standard - JT von ISO akzeptiert [Artikel] // CAD CAM. - München : Carl Hanser Verlag, 2009. - 11-12.
- SOFiSTiK** SOFiSTiK Basisfunktionalitäten [Buch]. - Oberschleissheim : SOFiSTiK AG, 2014.
- SOFiSTiK** SOFiSTiK Handbuch - SOFiMSHC Geometrische Modellierung [Buch]. - Oberschleissheim : SOFiSTiK AG, 2014.
- Sorger Christian** Generierung von Netzen für Finite Elemente hoher Ordnung in zwei und drei Raumdimensionen [Buch]. - München : Technische Universität München, 2012.

Tonke R. AI - Internationale Allianz für Interoperabilitä [Bericht]. - [s.l.] : VDI-Verlag, 1999.

Wang Minyi 3D-Planung von Brückenbauwerken mit Siemens NX 7.5 [Buch]. - München : Technische Universität München, 2012.

Weinholzer Michael Analyse und Implementierung von Datenaustauschformaten zwischen CAD- und AVA-Systemen im Zuge einer BIM-basierten Projektrealisierung im Ingenieurbau [Buch]. - München : Technische Universität München, 2014.

Werkle Horst Finite Elemente in der Baustatik [Buch]. - Wiesbaden : Vieweg & Sohn Verlag, 2008.

Willibald Günthner und Borrmann André Digitale Baustelle - innovativer Planen, effizienter Ausführen [Buch]. - Berlin Heidelberg : Springer-Verlag, 2011.

Zilch Konrad [et al.] Grundlagen des Bauingenieurwesens [Buch]. - Berlin Heidelberg : Springer-Verlag, 2013.