



HBC.

Automatisierte Modellierung mechanisch verbundener CLT-Wände

– Entwicklung eines algorithmischen Workflows am Beispiel der
Woodbloc Massivholzwand von Penzkofer

Eine wissenschaftliche Arbeit zur Erlangung des Grades eines Bachelor of Engineering

Automatisierte Modellierung mechanisch verbundener CLT-Wände

– Entwicklung eines algorithmischen Workflows am Beispiel der Woodbloc Massivholzwand von Penzkofer.

Vorgelegt bei:

Prof. Dr.-Ing Patrik Aondio

Vorgelegt von:

Lukas Götz

Mat. Nr.: 548711

Holzbau-Projektmanagement / Bauingenieurwesen

Sommersemester 2025

Hochschule Biberach

Eingereicht am:

25.07.2025

Vorwort und Danksagung

Mit der vorliegenden Arbeit schließe ich meine akademische Ausbildung an der Hochschule Biberach ab. Im Rahmen des Biberacher Modells hatte ich über mehr als fünf Jahre hinweg die Möglichkeit, mich intensiv mit dem Holzbau auseinanderzusetzen. Das Zusammenspiel aus einer handwerklichen Ausbildung zum Zimmerer und dem Studium an der Hochschule prägte nicht nur meinen beruflichen Werdegang, sondern auch den Wunsch, die Abschlussarbeit im Bereich des modernen Holzbaus zu verorten.

So entstand das Thema dieser Arbeit, bei dem ich mich mit der Entwicklung eines umfangreichen Automatisierungsprozesses beschäftigt habe. Die Bearbeitung lehrte mich viel über die Potenziale der Digitalisierung im Holzbau und die Herausforderungen interdisziplinärer Prozesse.

Mein besonderer Dank gilt Prof. Dr.-Ing. Patrik Aondio für die engagierte Betreuung, die fachliche Unterstützung und die Begleitung während des gesamten Bearbeitungszeitraums.

Ebenso danke ich Maximilian Ernst von der Penzkofer Bau GmbH für die praxisnahe Begleitung und die zahlreichen informativen Gespräche, die mein Verständnis für die Woodbloc Massivholzwand entscheidend vertieft haben.

Mein nächster Dank gebührt meinem Ausbildungsbetrieb, der Zimmerei Kaspar Holzbau GmbH, welche mich auf meinem Weg stets unterstützte und mir zeigte, was Holzbau alles kann.

Abschließend möchte ich mich bei Familie und Freunden bedanken, welche mich bis zum heutigen Tag auf meinem Weg stets unterstützt und ermutigt haben.

Vielen Dank!

Inhaltsverzeichnis

Vorwort und Danksagung	ii
Inhaltsverzeichnis	iii
Eidesstattliche Erklärung	vi
Abkürzungsverzeichnis.....	vii
Symbolverzeichnis.....	viii
1. Einleitung.....	1
1.1. Problemstellung und Relevanz.....	1
1.2. Zielsetzung und Forschungsthema	3
1.3. Methodisches Vorgehen	4
2. Theoretische Grundlagen	5
2.1. Grundlagen des Holzbaus und des CLTs	5
2.1.1. Historische Entwicklung des Holzbaus	5
2.1.2. Definition und Grundprinzip von Brettsperrholz	6
2.1.3. Anwendungsbereiche von CLT im modernen Holzbau.....	7
2.1.4. Verbindungstechnologin von Brettsperrholz	8
2.1.5. Relevanz des Nagelbildes bei mechanischen Verbindungen.....	10
2.2. Lastabtrag von CLT-Platten	10
2.2.1. Tragverhalten von CLT und Verbundmodelle.....	11
2.2.2. Bedeutung für die Modellierung	12
2.3. Normen und Richtlinien	13
2.3.1. Europäisches Bauproduktrecht	14
2.3.2. Nationales Bauproduktrecht	15
2.3.3. Rechtliche Einordnung der Woodbloc Massivholzwand	15
2.4. Digitaler Informationsfluss im Holzbau	16
2.4.1. Hoher Planungsaufwand im Holzbau.....	16
2.4.2. Datenstruktur	17
2.4.3. Die Rolle von direkten und indirekten Schnittstellen	18

2.4.4.	Der Workflow: Excel – Python – RFEM6	19
2.4.5.	Software.....	20
2.5.	Die wichtigsten Erkenntnisse aus Kapitel 2	21
3.	Entwicklung des Automatisierungsprozesses.....	23
3.1.	Analyse des ETA 24/0946 der Woodbloc Massivholzwand	23
3.1.1.	Aufbau der Woodbloc Massivholzwand	23
3.1.2.	Nagelbild nach ETA 24/0946	24
3.1.3.	Abstände der VBM	27
3.1.4.	Zusammenfassung der Anforderungen an das analytische Modell	28
3.2.	Verbindungsmittel und Schubverhalten der Woodbloc Massivholzwand	31
3.2.1.	LignoLoc Holznägel (ETA 23/0041).....	32
3.2.2.	Würth Assy 4 Schrauben (ETA 11/0190).....	35
3.3.	Entwicklung der Eingabeoberfläche	36
3.3.1.	Ziel der Eingabeoberfläche	36
3.3.2.	Die Eingabeoberfläche	40
3.4.	Format der Ergebnisausgabe	40
3.5.	Codeentwicklung.....	42
3.5.1.	Übersicht Berechnungsmodule	43
3.5.2.	Der Hauptcode und seine Funktionsweise	44
3.5.3.	Ergebnisexport.....	47
3.6.	Manuelle Eingaben	48
4.	Anwendung des erarbeiteten Prozesses	51
4.1.	Anwendung allgemein	51
4.2.	Eingabebeispiel mit Realmodell	52
4.3.	Verformungsvergleich Realmodell vs. Idealmodell	55
4.4.	Zeitaufwand.....	56
4.5.	Vergleich zu manuellem Modellieren.....	56
4.5.1.	Fehleranfälligkeit	56
4.5.2.	Zeitaufwand	57

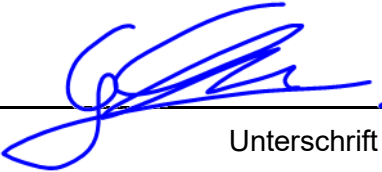
4.5.3. Direkter Vergleich	57
5. Ergebnisdiskussion und Zielerreichung	58
5.1. Bewertung der analytischen Korrektheit der Modelle	58
5.2. Bewertung der Zielerreichung	58
6. Zusammenfassung und Ausblick	60
6.1. Grenzen der aktuellen Umsetzung	60
6.2. Herausforderungen in der Entwicklung	61
6.3. Potentiale für Weiterentwicklungen	61
Abbildungsverzeichnis	64
Tabellenverzeichnis	66
Literaturverzeichnis	67
Anhang I. Anweisungen Ersteinrichtung	I
Anhang II. Ergebnisse aus dem Eingabebeispiel	VI
Anhang III. Verformungsergebnisse Real- & Idealmodell	XIII
Anhang IV. CODE REALMODELL	XIV
Anhang V. CODE IDEALMODELL	XXXVI
Anhang VI. CODE PYTHON	LV

Eidesstattliche Erklärung

Hiermit erkläre ich, Lukas Götz (Mat. Nr. 548711), dass ich die vorliegende Arbeit eigenständig und ohne fremde Hilfe angefertigt habe. Textpassagen, die wörtlich oder dem Sinn nach auf Publikationen oder Äußerungen anderer Autoren beruhen, sind als solche kenntlich gemacht. Es wurden keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Sprachmodelle wurden nicht anders als im KI-Hinweis¹ beschrieben eingesetzt. Die Arbeit wurde bisher keiner anderen Hochschule und keinem anderen Studiengang vorgelegt und auch nicht veröffentlicht.

Biberach an der Riß, 25.07.2025

Ort, Datum



Unterschrift

¹ KI-Hinweis

Zur Optimierung einzelner Programmierabschnitte im Rahmen des Automatisierungsprozesses wurden KI-gestützte Modelle verwendet. Die Anwendung erfolgte ausschließlich zur Effizienzsteigerung und entspricht den mit dem Betreuer abgestimmten Rahmenbedingungen.

Abkürzungsverzeichnis

aBG	allgemeine Bauartengenehmigung
abZ	allgemeine bauaufsichtliche Zulassungen
BIM	Building Information Modelling
BSPH	Brettsperholz
BauPVO	Bauproduktenverordnung
CLT	Cross-Laminated-Timber
EC5	Eurocode 5
ETA	Europäischen technischen Bewertungen
hEN	harmonisierte Europäische Produktnorm
IFC	Industry Foundation Classes
KLED	Klasse der Lasteinwirkungsdauer
LBO	Landesbauordnung
MBO	Musterbauordnung
NKL	Nutzungsklasse
SOAP	Simple Object Access Protocol
VBA	Visual Basic for Applications
VBM	Verbindungsmittel
ZiE	Zustimmung im Einzelfall

Symbolverzeichnis

A	Querschnittsfläche
d	Durchmesser
$F_{v,Rd}$	Tragfähigkeit des Verbindungsmittels rechtwinklig zur Achsrichtung
$f_{h,1,d}$	Bemessungswert der Lochleibungsfestigkeit des Bauteils 1
$f_{h,1,k}$	Charakteristischer Wert der Lochleibungsfestigkeit des Bauteils 1
$f_{h,2,d}$	Bemessungswert der Lochleibungsfestigkeit des Bauteils 2
$f_{h,2,k}$	Charakteristischer Wert der Lochleibungsfestigkeit des Bauteils 2
G_{fiktiv}	Fiktives Schubmodul zur Berücksichtigung der Nachgiebigkeit
K_{ser}	Steifigkeit der Verbindung
$K_{ser,axial}$	Steifigkeit entlang der Achse der Verbindung
$K_{ser,lateral}$	Schubsteifigkeit quer zur Achse der Verbindung
l	Länge des Verbindungsmittels pro Scherfuge
$M_{u,d}$	Bemessungswert des Biegebruchmoments des Nagels
$M_{u,k}$	Charakteristisches Biegebruchmoment des Nagels
t_1	Eindringtiefe des Nagels in Bauteil 1
t_2	Eindringtiefe des Nagels in Bauteil 2
$t_{1,req}$	Erforderliche Dicke des Bauteils 1
$t_{2,req}$	Erforderliche Dicke des Bauteils 2
α	Winkel zur Faserrichtung
β	Verhältnis der Lochleibungsfestigkeiten der verbundenen Bauteile
ρ_k	Charakteristische Rohdichte
ρ_{mean}	Mittlere Dichte
γ_M	Teilsicherheitsbeiwert für das Material

1. Einleitung

1.1. Problemstellung und Relevanz

Nach nunmehr über 30 Jahren Forschung und Entwicklung ist Brettsperrholz (BSPH, engl. CLT) heute vom Holzbau nicht mehr wegzudenken. Zwar liegt der Ursprung des Begriffs und der Idee hinter Brettsperrholz schon weit länger zurück, jedoch kann Prof. Gerhard Schickhofers Dissertation *Statistisches und dynamisches Verhalten von Brettsperrholz* 1994 an der Technischen Universität Graz (TU Graz)² als Meilenstein und Grundlage für die moderne Anwendung von Brettsperrholz im heutigen Holzbau gesehen werden. Seine Arbeit trug wesentlich dazu bei, dass vermehrt Hersteller eigene bauaufsichtliche Zulassungen (abZ) bzw. Europäische Technische Bewertungen (ETA) einreichten. Gleichzeitig nahm auch die Adaption dieser neuen Bauprodukte im Holzbau stetig zu und ersetzte in Teilen klassische Konstruktionen wie den Holzrahmenbau oder den Holzskelettbau. Gebäude wie Ein- und Mehrfamilienhäuser konnten nun vielfältiger gestaltet werden da man nicht mehr in den für die im Holzrahmenbau klassischen Raster denken musste, sondern nun ähnlich wie bei den klassischen Massivbauweisen in Flächen denken konnte. Neben dem Bau von Ein- und Mehrfamilienhäusern sind Vollholzprodukte vermehrt auch im Einsatz beim Mehrgeschossbau, Gewerbebau, öffentlichen Gebäuden und Sonderbauten wie Brücken oder Hallen. Aktuelle Markttrends zufolge nimmt auch weiterhin der Anteil an BSP im europäischen Holzbau zu³. Hierbei schreibt man diesem Holzprodukt viele positive Eigenschaften zu, wie zum Beispiel⁴ :

- Witterungsunabhängiger und passgenauer Produktion
- Nahezu unbegrenzte Designmöglichkeiten
- Vorteile gegenüber herkömmlichen Bau Produkten im Bereich Wärme-, Feuchte-, Brand- und Schallschutz
- Gesünderes Raumklima
- CO₂ Speicher
- Energiearme Herstellung

Die auf dem Markt weit verbreiteten CLT-Produkte basieren auf mehrlagige Elemente mit min. 3 Lagen aus Holzbrettern der Sortierklasse S7 oder höher. Je nach Hersteller liegt hier bei den Produkten ein symmetrischer oder asymmetrischer Aufbau vor, jedoch ist immer gegeben, dass die Lagen kreuzweise im Wechsel verlegt werden und die Neigungen dabei im Bereich

² ProHolz Austria (2015, S. 15).

³ ProHolz Austria (2008, S. 3).

⁴ Brettsperrholz.org ((o.A.)).

von 0° bis $\pm 90^\circ$ variieren können. Die Lagen werden in der Regel durch flächige Verklebungen mit direkter hydraulischer oder vakuumierter Verpressung miteinander verbunden.

Dieses Verfahren bietet Vorteile wie eine

- gleichmäßige Kraftübertragung,
- hohe Verbundwirkung,
- Formstabilität / geringe Nachgiebigkeit.

Jedoch sind damit in Verbindung stehend auch kritische Aspekte einhergehend. So ist der flächige Einsatz von Klebstoff aus nachhaltiger Sicht schlecht zu bewerten, die Diffusionsfähigkeit kann eingeschränkt werden und auch die Langlebigkeit von den verwendeten Klebstoffen variiert stark⁵. Das Phänomen des nachlassenden Leimes ist dabei ein ernstzunehmendes Problem, welches i.d.R. durch eine unzureichende Verleimung oder auf Feuchtigkeitseinflüsse zurückzuführen ist. Weiter gilt in Deutschland für Hersteller eine organisatorische Hürde aufgrund der Pflicht zur Vorlegung der Leimgenehmigung gemäß DIN 1052:2008. Auch das Wohlbefinden der Endnutzer kann durch das großflächige Vorhandensein von Leim in Aufenthaltsräumen als unangenehm empfunden werden, was insbesondere für Allergiker problematisch sein kann.

Dadurch rückt mittlerweile der Blick wieder vermehrt auf alternative Lösungen, darunter zählt die Verwendung von mechanischen Verbindungen wie beispielsweise Schrauben, Nägeln oder Dübeln. Zwar sind solche Produkte nicht neu auf dem Markt, jedoch gestaltet sich die Umsetzung noch immer aufwendiger als bei flächig verleimten CLT-Vollholzwänden. Dies hängt unter anderem damit zusammen, dass solche Produkte keine breitflächige Adaptation in Planungsprozessen finden. Das Modellieren einer verleimten CLT-Vollholzwand innerhalb einer dafür üblichen Software kann häufig über mehrschichtige Volumenobjekte vorgenommen werden. Eine solche Eingabe umfasst Konfigurationsparameter wie Dicke der einzelnen Lagen, Faserausrichtungen sowie Querschnitts- und Materialeigenschaften. Mechanisch Verbundene Elemente erfordern bei der Modellierung hingegen eine detailliertere Ausführung unter korrekter Berücksichtigung sämtlicher Aufbauten und detailliertem Einbezug der Verbindungsmittel aufgrund ihrer erhöhten Nachgiebigkeit.

Verglichen mit dem Modellieren verleimter CLT-Elemente sind mechanisch verbundene Elemente damit sehr zeit – und detailintensiv. Auch die Fehleranfälligkeit steigt mit erhöhtem Detaillierungsgrad einer Modellierung rapide an. Das Ziel der Modellierungsaufgabe definiert hierbei den Umfang. So spielt der exakte Aufbau und die Art der Verbindungen innerhalb einer CLT- Wand bei physischen Modellen, wie sie in Architekturmodellen üblich sind, nur eine untergeordnete Rolle. Hier geht es in erster Linie um die korrekte visuelle Darstellung des

⁵ Schickhofer (2010).

Modells. Die für die Tragwerksplanung zugrunde liegenden analytische Modelle hingegen zielen auf ein möglichst realitätsnahes Verhalten der Modelle ab, um aussagekräftige Berechnungen von Verformungen und Schnittgrößen ermöglichen zu können.

1.2. Zielsetzung und Forschungsthema

Die in 1.1 beschriebene Problematik führt bei der Erstellung des analytischen Modells einer mehrschichtigen, mechanisch verbundenen Vollholzwand zu einem höheren Zeitaufwand und birgt zudem aufgrund des hohen Detaillierungsgrades auch ein erhöhtes Fehlerpotenzial. Ziel dieser Arbeit soll sein, den beschriebenen Mehraufwand zu reduzieren, sowie das Fehlerpotential einzudämmen. Durch das Automatisieren von bestimmten Arbeitsschritten bei der Modellierung des analytischen Modells soll die Zielsetzung erreicht werden.

Konkret ist vorgesehen, ein frei skalierbares Template einer solchen Massivholzwand zu erstellen, welches automatisch und korrekt ausgeführt die mechanischen Verbindungen, sowie die korrekten Konfigurationen wichtiger analytischer Eigenschaften beinhaltet. Weiter wird dabei ein konkretes Produkt im Mittelpunkt dieser Arbeit stehen. So soll dieses Template den Modellierungsprozess des analytischen Modells der Woodbloc Massivholzwand von Penzkofer optimieren. Grundlage hierfür bietet das ETA-24/0946 vom 22.11.2024. Bei der Woodbloc Massivholzwand handelt es sich nach dem ETA-24/0946 um eine Massivholzwand mit sieben unterschiedlich ausgerichteten Brettlagen, welche mechanisch durch Schrauben und den Holznägeln LignoLoc (ETA 23/0041) verbunden sind. Es soll möglich sein, aufbauend auf ein fertig vorbereitetes Template, die gewünschte Geometrie einzugeben und ohne weiteren Aufwand ein verwendbares analytisches Modell der gewünschten Massivholzwand zu erhalten. Ausklinkungen, sowie Tür- und Fenster Öffnungen sollen dabei frei hinzugefügt werden können, ohne dass nachträgliche Anpassungen nötig werden. Ein besonderer Fokus wird hierbei auf die Thematik rund um die mechanischen Verbindungen innerhalb der Massivholzwand gelegt. Hierfür soll mithilfe eines Algorithmus automatisch ein Nagelbild entsprechend den Vorgaben aus dem ETA erstellt werden. Als Vergleichsbasis und als Indikator zur Bewertung der Zielerreichung soll das manuelle Modellieren einer Vergleichswand hinzugezogen werden, hierbei wird der benötigte Zeitaufwand zur Modellierung dieser manuell erstellten Massivholzwand gemessen. Ziel ist es einen Prozess zu entwickeln, der den Zeitaufwand um mindestens 30% beschleunigt sowie auch das Fehlerpotential reduziert. Die Zielsetzungen begleiten den Prozess dieser Bachelorarbeit, die sich folgendem Forschungsthema widmet:

Automatisierte Modellierung mechanisch verbundener CLT-Wände – Entwicklung eines algorithmischen Workflows am Beispiel der Woodbloc Massivholzwand von Penzkofer.

1.3. Methodisches Vorgehen

Zur Lösung der beschriebenen Problemstellung soll ein Prozess entwickelt werden, welcher mithilfe von Automatisierungen in gewissen Bereichen Teile der Modellierungsarbeit abnimmt. Hierfür wird zur Erörterung des aktuellen Sachstandes mit einer Literaturrecherche begonnen und im Umfang dessen werden auch Software und deren Möglichkeiten zur Einbindung untersucht.

Das Entwickeln des automatisierten Prozesses wird im Mittelpunkt stehen und soll linear aufbauend erfolgen. So wird mit minimalen Anforderungen begonnen, um im ersten Schritt ein funktionierendes Grundmodell zu erhalten. Darauf aufbauend sollen weitere Funktionen und Möglichkeiten mit eingebaut werden. Dieses Vorgehen ermöglicht einen koordinierten Entwicklungsprozess und verhindert unstrukturiertes Vorgehen. So wird in verschiedenen Entwicklungsstadien gearbeitet:

- | | |
|----------|---|
| 1 Stufe: | <u>Basisfunktionalität - Nagelbild</u> |
| | <ul style="list-style-type: none">• Entwicklung der Excel-Eingabeoberfläche• Basis Code mit Grundstruktur in VBA• Export Prozess über Python-RFEM API-Schnittstelle |
| 2 Stufe | <u>Erweiterungen - Nagelbild</u> |
| | <ul style="list-style-type: none">• Implementierung aller ETA-Vorgaben• Entwicklung der Ergebnisausgabe• Dynamische Öffnungsberücksichtigung |
| 3 Stufe | <u>Brettlagen</u> |
| | <ul style="list-style-type: none">• Wiederholung Stufe 1 + 2, aber diesmal für die Brettlagen |
| 4 Stufe | <u>Vervollständigung</u> |
| | <ul style="list-style-type: none">• Stäbe• Lager• Sonstige Randbedingungen |

Nach der vollständigen Entwicklung des Prozesses bis in Stufe vier wird eine Debugging Phase folgen. Hierbei ist es Ziel möglichst viele Varianten durchrechnen zu lassen und Fehler ausfindig zu machen. Gefundene Fehler sollen nach Möglichkeit korrigiert werden oder zumindest Lösungsansätze vorgeschlagen werden, falls der Umfang zu groß oder aus anderen Gründen nicht realisierbar wäre. Nach Abschluss der Korrekturen wird der Entwicklungsprozess beendet sein und es werden die Dokumentationen der Ergebnisse, sowie das Verschriftlichen der Arbeit folgen.

2. Theoretische Grundlagen

2.1. Grundlagen des Holzbaus und des CLTs

2.1.1. Historische Entwicklung des Holzbaus

Als eine der ältesten Bauweisen gehört der Holzbau zu den Urgesteinen des Bauwesens und kann bereits auf eine lange Historie zurückblicken. Historiker gehen davon aus, dass Holz bereits vor 40.000 Jahren, noch zu Zeiten des Jäger- und Sammlertums, für erste Schutzbauten vom Menschen verwendet wurden.⁶ Mit dem sesshaft werden des Menschen vor ca. 10.000 Jahren begann der Mensch verstärkt Holzbauweisen zu entwickeln, welche über primitive Schutzhütten hinausgehen. Am häufigsten wurden Bauwerke entweder in Blockbauweise oder in Gegenden mit geringerem Holzvorkommen auch in Stabbauweise errichtet. Die bei der Stabbauweise eher geringe Lebenszeit der damaligen Bauten von nur etwa 20 bis 30 Jahren⁷, welche eine Folge der Einspannung des Holzes im Boden ist, wurde mit der Zeit durch neue Entwicklungen und Bauweisen stetig verbessert. Durch die hohe Verfügbarkeit des Baumaterials Holz konnte sich der Holzbau auch als dominierende Bauweise über sehr lange Zeiten halten. Dabei entwickelten sich immer langlebigere Bauweisen über den Pfahlbau, dem Pfostenbau bis hin zum Fachwerkbau. Viele Holzbauweisen hatten zwar in bestimmten Epochen ihre Blütezeit, dennoch hielten sich einige dieser Bauweisen über mehrere Jahrhunderte bis ins späte Mittelalter. Das große Vorhandensein des Holzes in Verbindung mit der verglichen einfachen Verarbeitung des Materials ließen anderen Bauweisen lange kaum eine Chance. Selbst im Mittelalter dominierte der Holzbau noch etwa 90% des kompletten Baus in unseren Regionen. Man erkannte, dass durch das Verbinden von Horizontalen, vertikalen und diagonalen Hölzern (im Fachwerkbau wird hierbei von Schwelle / Rähm, Pfosten und Streben gesprochen)⁸ Wandscheiben erhalten konnte, welche die Möglichkeiten zu bauen revolutionierten. Häuser konnten nun mit bis zu 6 Geschossen errichtet werden.

Bis ins 1900 Jahrhundert hinein wurde der Holzbau immer intensiver weiterentwickelt, erst mit dem Aufkommen des Stahlbaus und später auch des Betons kamen ernstzunehmende Konkurrenten auf den Markt. Ab dieser Zeit reduzierte sich der Anteil des Holzbaus in weniger als einem halben Jahrhundert auf etwa 30% des gesamten Marktes. Durch seine bisherige Beschränkung auf das Verwenden von Voll- und Schnittholz, welches lediglich bearbeitet wurde, unterlag der Holzbau dem sich rasant entwickelten Stahlbau. Es war nötig, den Holzbau durch erste industrielle Entwicklungen wieder wirtschaftlich konkurrenzfähig zu machen und komplexere Tragwerke zu ermöglichen. Die Entwicklung der „Hetzer-Träger“ von Weimar Otto

⁶ Lißner und Rug (2018, S. 5-7).

⁷ Stefan Krötsch (o.A.).

⁸ Anm. regional kann es hier zu starken Abweichungen in der Bezeichnung kommen

Hetzer im Jahre 1906 mit dem Patent *Tragwerke aus verleimten Brettlamellen* gaben den Startschuss für ein Umdenken im Holzbau: Erstmals wurde im großen Stil der Baustoff Holz genutzt, um daraus ein Bauprodukt, dem aus mehreren Lamellen unter Druck verleimten Träger, mit wesentlich besseren Eigenschaften als konventionelle Holzstäbe zu erhalten. Große Spannweiten für Lager oder Hallen konnten hierdurch wieder in Holzbauweise konstruiert werden. Dabei war diese Bauweise häufig günstiger als das Bauen mit Stahl oder Beton und fand so wieder einen kleinen Aufschwung. Es folgten Entwicklungen von kombinierten Holzprodukten im Holz-Beton-Verbund⁹, dem Holzrahmenbau oder auch wirtschaftliche und leistungssteigernde Veredelungen von Holz zu neuen Holzprodukten, welche dem Holzbau über die Jahre wieder etwas mehr Aufmerksamkeit in der Baubranche bescherten. Bei den Veredelungen des Holzes wurden Produkte aus dem Rundholz selbst, aber auch aus zusammengesetzten Holzpartikeln oder auch Furnieren auf den Markt gebracht. Mit Prof. Gerhard Schickhofers Dissertation zum Brettsperrholz 1994 wurde der Grundstein für salonfähiges Brettsperrholz gelegt. Damit konnte im Holzbau erstmals auch in Platten gedacht werden. In Verbindung mit der zur gleichen Zeit immer stärker wachsenden Nachfrage nach ökologischen Lösungen erlebt der Holzbau bis heute einen regelrechten Aufschwung.

2.1.2. Definition und Grundprinzip von Brettsperrholz

Brettsperrholz ist ein veredeltes mehrschichtiges Holzprodukt, welches aus mindestens drei in unterschiedlichen Faserausrichtungen verlegten Brettlagen besteht und dessen Schichten mit einer flächigen Verklebung oder auch einer mechanischen Verbindung miteinander verbunden sind. Die Faserausrichtungen sind dabei meist im Wechsel zwischen 90° und 0° verlegt, jedoch hängt dies immer vom Hersteller und dessen konkretem Produkt ab. So sind oft auch CLT mit diagonalen $\pm 60^\circ$ Faserausrichtungen oder auch Schichtaufbauten mit faserparallelen benachbarten Lagen erhältlich. In der Regel handelt es sich bei den Brettern meist um Nadelholzbretter¹⁰ mindestens der Sortierklasse S7 mit Brettdicken zwischen 17mm und 45mm. Häufig aber werden Bretter der Sortierklasse S10 verwendet, um mehr Anwendungsbereiche abdecken zu können. Dabei haben sich auf dem Markt als typische Brettstärken 20mm, 30mm, 40mm oder auch 45mm etabliert. Bei CLT-Elementen mit Brettern von nur einer Sortierklasse spricht man von homogenen CLT-Elementen, ansonsten von inhomogenen CLT-Elementen. Häufig wird die Möglichkeit inhomogener CLT-Elemente genutzt, um die Kosten zu sparen. Dabei werden i.d.R. für die äußeren Lagen höherwertigere Brettlamellen verwendet und die weniger belasteten Zwischenschichten erhalten minderwertigere Brettlamellen. Der Aufbau der Lagen ist in der Regel symmetrisch, sodass handelsüblich Produkte mit 3-, 5-, 7- oder in Ausnahmefällen auch 9- und mehr Lagen

⁹ Stefan Krötsch (o.A.).

¹⁰ EN 16351, S. 24.

erhältlich sind.¹¹ Dies in Kombination mit den üblichen Brettdicken ermöglicht CLT-Elemente mit Dicken von 60mm bis zu 300mm. Auch hier hängen die tatsächlich verfügbaren Varianten wieder vom konkreten Hersteller ab. Mit entsprechender Zulassung können Hersteller auch Elemente mit 500mm oder mehr produzieren.¹² Die Herstellung von Brettsperrholz beginnt mit der Ernte des Holzes und folgt im Allgemeinen folgendem Ablauf:

- 1) Holzernte: Meist hochwertiges Nadelholz wie Fichte oder Tanne, seltener aber auch Kiefer, Douglasie oder Lärche¹³
- 2) Sägen und Trocknen: Das Holz wird in Bretter oder Bohlen gesägt und in Trockenkammern auf eine Holzfeuchte von $12 \pm 2\%$ getrocknet¹⁴
- 3) Festigkeitssortierung: Das getrocknete Holz wird visuell, oder wie heute eher üblich maschinell kontrolliert. Das Holz wird jeweils der Sortierklasse zugeteilt, dessen Sortierkriterien es einhält
- 4) Keilzinkverbindung: Die Hölzer werden getrennt und individuell wieder mit einer Keilverzinkung zu Brettlamellen verleimt, um zum einen unerwünschte Holzfehler zu entfernen und zum anderen die gewünschten Lamellenlängen zu erhalten (Endloslamellen)
- 5) Hobeln: Nun werden die Brettlamellen auf die endgültige Dicke gehobelt
- 6) Verbinden der Brettlagen: Die fertigen Brettlamellen werden lagenweise je nach Herstellerprodukt verlegt und miteinander verbunden
- 7) Zuschnitt und Abbund: Nun können die fertigen Elemente auf die gewünschten Maße zugeschnitten werden

Die Brettsperrholz Produkte müssen in Deutschland mit dem Ü-Kennzeichen zur Erfüllung der nationalen Anforderungen oder mit dem CE-Kennzeichen gemäß der europäischen Zulassung vom Hersteller versehen werden.

2.1.3. Anwendungsbereiche von CLT im modernen Holzbau

Die hohe Veredelung des Holzes hin zum Brettsperrholz wird mittlerweile aufgrund seiner Eigenschaften in der Baubranche sehr wertgeschätzt. Gerade Architekten schätzen die hohe Flexibilität von CLT-Elementen, da diese keine rasterabhängige Planung erfordern, wie sie im Gegensatz zu den meisten anderen Holzbauweisen erforderlich wäre. Die Elementmaße sind sehr flexibel, abbindbar und ermöglichen so vielfältige architektonische Gestaltungsmöglichkeiten. Dies betrifft zum Beispiel das Setzen von Tür- und Fensteröffnungen: CLT-Elemente erlauben eine weitaus höhere Flexibilität als es beispielsweise beim weit verbreiteten Holzrahmenbau üblich ist, da die Planung beim Holzrahmenbau unter anderem

¹¹ Pfeifer Group – Passion for Timber (2025b).

¹² Brettsperrholz.org ((o.A.), S. 4-8).

¹³Anm.: Laubhölzer sind für tragende CLT-Elemente zurzeit nicht zugelassen und können daher nur in nichttragenden Lagen verwendet werden (z.B. aus ästhetischen Gründen)

¹⁴ Vgl. EN 16351, S. 82.

an das Raster der Ständer gebunden ist. Weiter sind CLT-Elemente auch bei Planern sehr beliebt aufgrund der Charakteristik, sowie eindeutiger und hoher Materialkennwerte der Platte. Dies führt zu einer hohen Belastbarkeit bei einem gleichzeitig geringen Eigengewicht. Das spielt besonders in der Tragwerksplanung eine Rolle. Dem Brettsperrholz wird auch eine hohe Formstabilität zugeschrieben, da durch den mehrschichtigen und kreuzweisen Aufbau der Brettlamellen unterschiedliche Schwund- und Quelländerungen sich gegenseitig aufheben und das Produkt so nicht weniger stark orthotrop reagiert, als es für herkömmliches Schnittholz üblich ist. Unter den bauausführenden Unternehmen und Herstellern ist besonders die leichte Verarbeitung der Brettsperrholzelemente beliebt, da es eine Witterungsunabhängige Vorfertigung mit kurzen Montagezeiten ermöglicht. Werden die CLT-Elemente mit anderen Massivbauweisen, wie bspw. dem Stahlbetonbau, verglichen, so stechen auch die guten Wärmedämmeigenschaften des Holzes sowie die genaue Kalkulierbarkeit beim Brandschutz positiv hervor. Die im Allgemeinen gute Ökobilanz des Holzes durch das Einspeichern von CO₂, sowie geringem CO₂ Ausstoß in der Herstellung spielt in der aktuellen Zeit ebenfalls eine wichtige Rolle und ist unter den Massivbauweisen fast einzigartig. Abschließend kann gesagt werden, dass sich Brettsperrholz unter den Bauherren aufgrund seiner angenehmen Ästhetik, sowie dem angenehmen Raumklima einer großen Beliebtheit erfreut.¹⁵

Aufgrund dieser und weiterer Faktoren findet Brettsperrholz bei der Herstellung tragender und raumteilender Bauteile Anwendung. So hat sich das Produkt bereits in Wohnbauten, öffentlichen Gebäuden, wie Kindergärten und Schulen, Gewerbebauten, Ausstellungs-/ Versammlungs-/ Sporthallen, sowie auch Türmen, Brücken oder Sonderbauten etabliert.¹⁶ In Zukunft wird durch das Nachverdichten bereits bebauten Wohnraums unter den Massivbauweisen besonders das Brettsperrholz davon profitieren, da es ein vergleichbar geringes Eigengewicht hat und so im Bestand leichter aufgestockt und nachverdichtet werden kann.

2.1.4. Verbindungstechnologin von Brettsperrholz

Brettsperrholz-Produkte sind als Massivholzbauprodukt stark veredelt. Dabei umfasst die Veredelung mehrere Schritte bis zum fertigen Massivholzprodukt. Als Teil des zuvor bereits beschriebenen Herstellungsprozesses ist das Keilverzinken der einzelnen Brettlamellen integraler Bestandteil und somit eine erste wichtige Verbindungstechnologie. Bei einer Keilverzinkung handelt es sich um eine selbstzentrierende Endverbindung. Diese wird gebildet, indem eine Reihe von gleichartigen, symmetrischen, spitz zulaufenden Keilzinken mit einer Keilzinkfräse maschinell in die Endstücke von Brettlamellen bzw. Vollhölzern eingefräst und dann miteinander verklebt werden.¹⁷ Zum einen werden hierdurch Endloslamellen mit der

¹⁵ Pfeifer Group – Passion for Timber (2025a).

¹⁶ Brettsperrholz.org ((o.A.), S. 16-34).

¹⁷ Simon Aicher (S. 62).

gewünschten Länge geschaffen und zum anderen können Festigkeitsmindernde Holzfehler ausgelassen werden. Neben den Brettlammen werden auch die einzelnen Lagen des Brettspertholzes flächig miteinander verklebt, sofern das konkrete Produkt eines Herstellers nicht auf eine mechanische Verbindung der einzelnen Brettlagen setzt. Weiter kann auch eine Schmalseitenverklebung der einzelnen Brettlamellen durchgeführt werden, jedoch ist auch dies wieder Herstellerabhängig und nicht verpflichtend vorgeschrieben. Aus der Abbildung 1 lassen sich alle genannten Verklebungen entnehmen.

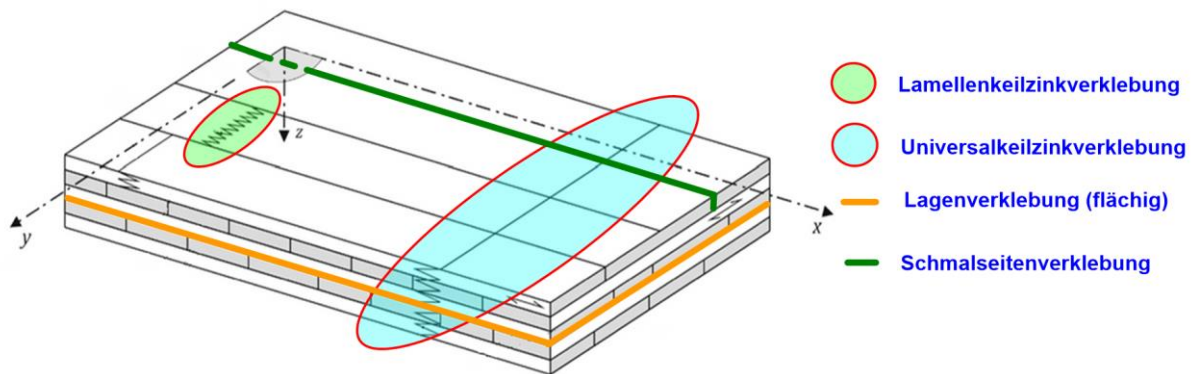


Abbildung 1 Übersicht möglicher Verklebungen eines CLT-Elements (EN 16351, S. 13 Bild 2 (geändert))

Bei der Auswahl der verwendeten Klebstoffe gibt die EN 16351 zum einen Grenzwerte der Formaldehydabgabe von CLT-Produkten vor (Formaldehydabgabeklasse E1 oder E2)¹⁸, zum anderen wird deklariert, welche Klebstofffamilien grundsätzlich zulässig sind. Dies sind¹⁹

- Phenoplast- und Aminoplast-Klebstoffe (z.B. MF, MUF, PRF)
- Feuchtigkeitsvernetzende Ein-Komponenten-Klebstoffe auf Polyurethanbasis (PUR-Klebstoffe)
- Emulsion-Polymer-Isocyanat-Klebstoffe (EPI-Klebstoffe).

In der EN 16351 werden weitere Aspekte im Zusammenhang mit den Verklebungsprozessen festgelegt, welche aber für die weitere Ausarbeitung dieser Arbeit von geringer Relevanz sind und daher nicht weiter ausgeführt werden. Dazu zählen die Art des Klebstoffauftrags, des nötigen Pressdrucks, der maximalen Klebefugendicke, den Aushärtungszeiten und der Feuchtegehalte von Lamellen und Lagen. Auch die Prüfverfahren dieser genannten Punkte werden darin festgelegt.²⁰

Auffällig und wichtig für den weiteren Verlauf dieser Arbeit ist die Tatsache, dass sich die EN 16351 ausschließlich auf Flächenverklebungen fokussiert und nicht auf die Möglichkeiten der mechanischen Verbindungen der einzelnen Lagen eingeht. Dennoch ist es möglich, anstelle der flächigen Verklebungen der einzelnen Lamellenlagen mit mechanischen Verbindungen zu

¹⁸ EN 16351, S. 32.

¹⁹ EN 16351, S. 33.

²⁰ EN 16351, S. 82–87.

arbeiten. Dabei werden in der Regel stiftförmige Verbindungsmittel wie etwa Holzschrauben, Stahl- / Aluminium- / Holznägel oder Dübel verwendet. Da dies nicht in der EN 16351 geregelt ist, bedarf es bei mechanisch verbundenen CLT-Elementen weiterer normativer Betrachtungen. An dieser Stelle auf das Kapitel 2.3 zu verweisen.

2.1.5. Relevanz des Nagelbildes bei mechanischen Verbindungen

Da nun aus der EN 16351 hervorgeht, dass ausschließlich geklebte Verbindungen zwischen den flächigen Lagen beschrieben und spezifiziert sind, bedarf es einer gesonderten Erwähnung zu den mechanischen Verbindungen. Im Allgemeinen wird bei flächigen Verklebungen von Klebefugen und bei den mechanischen Verbindungen von Scherfugen gesprochen. Aufgabe beider Verbindungsarten ist das Übertragen von Kräften zwischen den Lagen. Sollte nun abweichend vom allgemeinen Vorgehen anstelle der Klebefugen mit mechanisch verbundenen Scherfugen gearbeitet werden, so wie es bei der Woodbloc Massivholzwand von Penzkofer der Fall ist, so ist dies meist mit einem Mehraufwand in verschiedenen Bereichen verbunden. Während bei Klebefugen Parameter wie die Art des Klebstoffs oder des Pressdrucks ausschlaggebende Faktoren für die Festigkeit der Fugen waren, spielen nun bei den Scherfugen mechanischer Verbindungen Parameter wie die Art des VBM, Anzahl und Abstände der VBM, Dimensionen der VBM, Material und auch die Steifigkeiten der Scherfugen eine ausschlaggebende Rolle. Weiterhin gilt es auch zu berücksichtigen, dass solche mechanischen Verbindungen deutlich nachgiebiger sind als es bei den Klebefugen der Fall ist. Die Unterschiede zwischen Klebefugen und den Scherfugen mechanisch verbundener Lagen erfordern eine angepasste Betrachtungsweise von mechanisch verbundenen CLT-Produkten, insbesondere bei Aufgaben, die sich mit Verformungen und Schnittgrößen befassen.

Da in dieser Arbeit der Modellierungsprozess einer mechanisch verbundenen CLT-Massivholzwand im Fokus steht, werden im Folgenden die dafür nötigen theoretischen Hintergründe im Zusammenhang mit dem Nagelbild genauer betrachtet.

2.2. Lastabtrag von CLT-Platten

Die Bemessung von CLT-Elementen erfolgt primär nach den Vorgaben des Eurocode 5 (DIN EN 1995 / NA) in Zusammenhang mit den produktspezifischen Anforderungen des jeweiligen ETA / der jeweiligen abZ des Herstellers der jeweiligen Massivholzwand.²¹ Es sollen im Folgenden nun die nötigen Grundlagen bezüglich der Lastabtragungen und des Verformungsverhaltens für den praktischen Teil dieser Arbeit geschaffen werden. Ein besonderer Fokus wird hierbei auf die Unterscheidung zwischen einem geklebten Verbund

²¹ Fachverband Holzindustrie Österreich (2024, S. 11).

und einem mechanischen Verbund, wie in Kapitel 2.1 beschrieben, liegen, da dies für den praktischen Teil dieser Arbeit von erheblicher Arbeit ist.

2.2.1. Tragverhalten von CLT und Verbundmodelle

CLT-Massivholzwände werden als Flächenelemente betrachtet und weisen aufgrund ihrer kreuzweisen Lagenanordnung eine zweiachsige Tragwirkung auf, sodass sie Lasten in zwei orthogonale Richtungen abtragen können.²² Je nachdem, wie die Lasten später auf das Element wirken kann zwischen einer Plattenwirkung und einer Scheibenwirkung unterschieden werden:

- 1) Plattenwirkung: Bei senkrecht zur Elementebene wirkenden Lasten wird von einer Plattenbeanspruchung gesprochen. Hierdurch werden Biegebeanspruchungen erzeugt, die in erster Linie über die parallel zur Spannrichtung verlaufenden Brettlagen abgetragen werden.²³ Diese Art der Lasteinwirkung tritt in der Regel bei Deckenelementen auf. Platten werden meist linienförmig gelagert, in diesen Fällen wird auch nur ein einachsiger Lastabtrag in Spannrichtung betrachtet. Jedoch bieten CLT-Elemente aufgrund ihrer Fähigkeit zum zweidimensionalen Lastabtrag auch die Möglichkeit von Punktlagerungen.²⁴
- 2) Scheibenwirkung: Bei parallel zur Elementebene wirkenden Lasten wird von einer Scheibenbeanspruchung gesprochen. Hierbei werden Normal- und Querkraftbeanspruchungen erzeugt, welche überwiegend über die vertikal angeordneten Brettlagen abgetragen werden (vgl. Abbildung 2). Diese Art des Lastabtrags tritt z.B. bei Wänden unter Einfluss vertikaler Lasten auf.²⁵ Erfahren die Wände jedoch noch zusätzlich zu den vertikalen Lasten auch horizontale Lasten durch Querkräfte, so kommt es zusätzlich zu einer Biegebeanspruchung und somit auch wieder zu einer Plattenwirkung. Gelagert werden Wände in der Regel mit einem Linienlager, jedoch erlauben CLT-Elemente auch Punktlager, wodurch die Wand als Wandartiger Träger angesetzt werden kann.

Für die Modellierung der Woodbloc Massivholzwand ist insbesondere die Scheibenwirkung von Relevanz. Dabei wird die Tragwirkung der einzelnen Brettlagen durch den Verbund zwischen ihnen gewährleistet.

Grundsätzlich werden zwei Haupttypen des Verbunds unterschieden:

- 1) Starrer Verbund: Bei den klassischen, vollflächig verklebten Brettsperrholz-Elementen wird ein starrer Verbund nach der starren Verbundtheorie²⁶ zwischen

²² Informationsdienst Holz (2010, S. 9).

²³ Mestek Peter (2010, S. 4).

²⁴ Mestek Peter (2010, S. 10).

²⁵ Mestek Peter (2010, S. 11).

²⁶ Mestek Peter (2010, S. 6).

den Lagen angenommen. Hierbei übertragen die Klebefugen die Schubkräfte vollständig zwischen den einzelnen Brettlagen. Dies vereinfacht die statische Bemessung, da das Element als homogene Platte oder Scheibe betrachtet werden kann.

- 2) Nachgiebiger Verbund: Im Gegensatz zum starren Verbund weisen mechanische Verbindungen, wie sie durch das Nagelbild bei der Woodbloc Massivholzwand vorkommt, einen nachgiebigen Verbund auf. Dies bedeutet, dass die VBM (hier Schrauben und Nägel) unter Lasteinwirkung einen gewissen Schlupf zulassen. Der Schlupf beschreibt eine relative Verschiebung zwischen zwei Bauteilschichten. Wie steif bzw. nachgiebig eine mechanische Verbindung zweier Bauteile (Scherfuge zwischen zwei Brettlagen) ist, kann durch die Federsteifigkeit (auch Verschiebungsmodul genannt) der Verbindung, dem K_{ser} -Wert, angegeben werden. Wichtig ist hierbei zwischen dem lateralen Verschiebungsmodul $K_{ser,lateral}$ und dem axialen Verschiebungsmodul $K_{ser,axial}$ zu unterscheiden. Das laterale Verschiebungsmodul berücksichtigt die Quer- bzw. Schubsteifigkeit (also gegen Kräfte senkrecht zur Achse der Verbindung) zwischen zwei verbundenen Bauteilen. Das axiale Verschiebungsmodul berücksichtigt hingegen die Steifigkeit entlang der Achse der Verbindung, also gegen Zug / Druck.

Verglichen mit dem Verbund von verklebten Brettsperrholz-Elementen haben mechanisch Verbundene Brettsperrholz-Elemente eine geringere Gesamtsteifigkeit und somit potenziell auch höhere Verformungen.

2.2.2. Bedeutung für die Modellierung

Aufgrund des nachgiebigen Verbunds können mechanisch verbundene CLT-Elemente nicht wie flächig verklebte CLT-Elemente mit vereinfachten Platten- / Scheibenmodellen (häufig unter „Mehrschichtige Flächenelemente“ zu finden) betrachtet werden. Eine statische Analyse einer mechanisch Verbundenen Brettsperrholzwand erfordert daher eine exakte Betrachtung aller Brettlagen unter korrekter Berücksichtigung der einzelnen Brettlagen, sowie der eingesetzten VBM. Dabei spielt auch die Art der aufkommenden Belastung eine wichtige Rolle. Maßgebende Belastungen der Woodbloc Massivholzwand liegen innerhalb der Plattenebene (vgl. Abbildung 2), diese ist somit auch als eine Wandscheibe bemessen wird. Die Kräfte sind hier quer zur Achse der mechanischen Verbindung, womit die Betrachtung des lateralen Verschiebungsmoduls maßgebend wird. Dieser Ansatz ist fundamental für die Entwicklung des Automatisierungsprozesses in dieser Arbeit, da es Ziel ist ein korrektes analytisches Modell zu generieren. Somit muss der spezifische $K_{ser,lateral}$ der Verbindungen innerhalb der Woodbloc Massivholzwand hinterlegt sein. Die genaue Herleitung der entsprechenden Werte folgt im praktischen Teil der Arbeit mit Kapitel 3.2 behandelt.

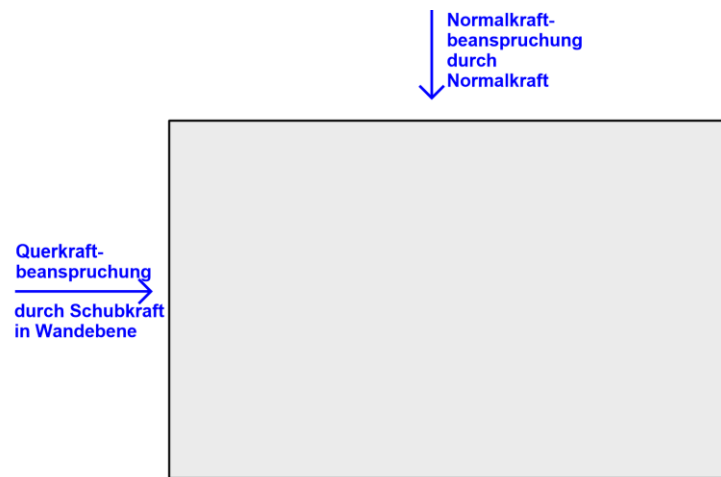


Abbildung 2 Scheibenwirkung durch Lasten in Wandebene (eigene Darstellung)

In der Tragwerksplanung bezeichnet das analytische Modell das Tragwerksmodell. Dieses bildet die Gebäudestruktur (oder Elementstruktur) bestehend aus den Lastabtragenden Elementen, wie beispielsweise Knoten, Linien, Flächen, ab und dient unter anderem der Berechnung von Schnittgrößen und Verformungen.²⁷

2.3. Normen und Richtlinien

Bauprodukte unterliegen einer komplexen Struktur, welche ausgehend von europäischen Verordnungen bis hin zu herstellerproduktspezifischen Angaben unterschiedlichen Verordnungen unterliegt. Für einen sicheren und auch regelkonformen Umgang mit einem Bauprodukt innerhalb der Bundesrepublik Deutschland ist es daher unumgänglich, diese Thematik aufzubereiten. Nach der Begriffsdefinition aus der Musterbauordnung (MBO) sind Bauprodukte²⁸

- 1) Produkte, Baustoffe, Bauteile und Anlagen [...], die hergestellt werden, um dauerhaft in bauliche Anlagen eingebaut zu sein,
- 2) aus Produkten [...] vorgefertigte Anlagen, die hergestellt werden, um mit dem Erdboden verbunden zu werden.

Nach dieser Definition sind auch Brettsperrholz-Elemente als Bauprodukt anzusehen und unterliegen somit der allgemein gültigen Rangfolge von Normen und Richtlinien. Diese wird im Folgenden genauer betrachtet. Zu unterscheiden ist hier in erster Linie zwischen europäischem Bauproduktrecht und dem nationalem Bauproduktrecht. Das Ziel eines Bauproduktrechts ist es, dass für auf den Markt gebrachte Bauprodukte einheitliche Regelungen gelten.²⁹

²⁷ Borrmann, König, Koch und Beetz (2021, S. 409).

²⁸ ARGEBAU (2002)§2 Abs. 10.

²⁹ Oliver Justus (S. 7).

2.3.1. Europäisches Bauproduktrecht

Auf europäischer Ebene werden die Rahmenbedingungen für den Einsatz von Bauprodukten geschaffen. Dies beinhaltet besonders das Einführen eines für alle Mitglieder gültigen harmonisierten Bewertungsverfahrens von Produkten sowie deren Eigenschaften und Spezifikationen. Dabei bleibt jedoch die Verantwortung über die Einhaltung dieser Rahmenbedingungen in der Zuständigkeit der einzelnen Mitglieder. Zentraler Bestandteil dieses europäischen Regelwerks ist die Bauproduktenverordnung (BauPVO), welche zuletzt 2024 neu veröffentlicht wurde und die allgemeinen Regelungen für das Inverkehrbringen von Bauprodukten auf den europäischen Markt regelt. Im Sinne der BauPVO gibt es harmonisierte europäische Produktnormen (hEN), dies sind von einer europäischen Normungsorganisation (CEN oder CENELEC) erstellte Normen, welche zudem im Amtsblatt der Europäischen Union (OJEU) veröffentlicht sind.³⁰ Diese hEN sind nach der BauPVO mit einer CE-Kennzeichnung und einer Leistungserklärung (Declaration of Performance (DoP)) zu versehen. Wichtig hierbei ist zu verstehen, dass die Veröffentlichung im Amtsblatt der Europäischen Union Voraussetzung einer hEN ist.

So ist beispielsweise die EN 14080 (Brettschichtholz) eine harmonisierte Europäische Produktnorm, die EN 16351 (Brettsperrholz), welche für diese Arbeit von zentraler Bedeutung ist, wurde bisher jedoch nicht im Amtsblatt der Europäischen Union gelistet und ist somit auch keine harmonisierte Produktnorm.³¹ Daher sind bei der Arbeit mit Brettsperrholz aktuell wichtige Punkte zu berücksichtigen:

- 1) CE-Kennzeichnung erfolgt nach ETA-Bewertungskriterien und nicht nach der BauPVO
- 2) Die EN 16351 entspricht zwar allg. technischem Standard, ist aber keine gesetzlich verbindliche Bewertungsgrundlage im Sinne der BauPVO
- 3) Für ein CLT-Produkt sind weitere alternative Bewertungen nötig

Daraus folgt für Brettsperrholz, dass jedes Produkt einer eigenen Bewertung bedarf. Dies muss vom Hersteller beantragt und finanziert werden. Dabei wird von einer Europäischen Technischen Bewertung (ETA) gesprochen, wenn das Produkt auf europäischer Ebene zugelassen werden soll. Wird jedoch lediglich eine nationale Zulassung angestrebt, so stehen dem Hersteller noch andere, auf nationaler Ebene wirkende Regelungen anstelle der europäischen ETA zur Verfügung, um sein Bauprodukt auf den nationalen Markt zu bringen.

Dieser signifikante Mehraufwand für Hersteller eines CLT-Produkts verdeutlicht nochmals die grundlegende Problematik, dass das Arbeiten mit Brettsperrholz aktuell noch mit einem Mehraufwand verbunden ist.

³⁰ EU-Bauproduktenverordnung online (2025).

³¹ Fachverband Holzindustrie Österreich (2024, S. 2).

2.3.2. Nationales Bauproduktrecht

Auf nationaler Ebene werden in Deutschland die allgemeinen Anforderungen an Bauprodukte durch die Landesbauordnungen (LBO) der einzelnen Länder geregelt. Damit die Regelungen zwischen den einzelnen Bundesländern nicht zu stark voneinander abweichen, wird von der ARGEBAU die sogenannte Musterbauordnung (MBO) festgelegt, an welcher sich die LBOs der Bundesländer orientieren sollen. Konkretisiert werden die allgemeinen Anforderungen der MBO durch die Muster-Verwaltungsvorschrift Technische Baubestimmungen (MVV TB)³².

Anhand der LBO in Verbindung mit der MBO und der MVV TB wird auch festgelegt, wann und wie in Deutschland Bauprodukte verwendet werden dürfen. Nach der MBO dürfen Bauprodukte mit einer CE-Kennzeichnung automatisch auch in Deutschland verwendet werden.³³ Fehlt nun diese CE-Kennzeichnung bei einem Bauprodukt (keine hEN und auch keine ETA vorhanden), so kann das Bauprodukt dennoch verwendet werden, indem der Hersteller beim Deutschen Institut für Bautechnik (DIBt) einen Verwendbarkeitsnachweis für sein Bauprodukt beantragt. Dies führt entweder zu einer allgemeinen bauaufsichtlichen Zulassung (abZ), einer allgemeinen Bauartengenehmigung (aBG) (seit 2017)³⁴ oder auch einer Zustimmung im Einzelfall für das Bauprodukt (ZiE). Bauprodukte mit einer dieser Zulassungsarten werden mit einem Übereinstimmungszeichen (Ü-Zeichen) versehen.

Die Thematik rund um die europäische und nationale Gesetzgebung bezüglich der Zulassung und Verwendung von Bauprodukten umfasst weitere Vertiefungen und Sonderfälle. Für das Verständnis zur Einordnung der Woodbloc Massivholzwand von Penzkofer genügen jedoch die hier erörterten Grundlagen.

2.3.3. Rechtliche Einordnung der Woodbloc Massivholzwand

Bei der Woodbloc Massivholzwand handelt es sich um ein Brettsperrholz-Bauprodukt. Die EN 16351 ist eine europäische Norm, welche Anforderungen zu Brettsperrholz formuliert. Sie kann als die am ehesten passenden Norm für die Woodbloc Massivholzwand angesehen werden. Es ist jedoch festzuhalten, dass die EN 16351 sich ausschließlich mit starr verbundenen Brettsperrholzelementen befasst und die Woodbloc Massivholzwand im Gegensatz dazu mit nachgiebigen Verbindungen arbeitet. Daher muss von wesentlichen Abweichungen des Bauprodukts von der technischen Baubestimmung, der EN 16351, ausgegangen werden. Es ist auch festzuhalten, dass die EN 16351 zurzeit (stand 24.06.2025) nicht im Amtsblatt der europäischen Union geführt ist und somit keine harmonisierte Europäische Norm (hEN) darstellt.

Wird angestrebt die Woodbloc Massivholzwand ausschließlich auf nationaler Ebene zu verwenden, so würde ein Verwendbarkeitsnachweis nach §17 MBO genügen. Dieser kann

³² DIBt - FAQ Service (o.A.).

³³ Vgl. §16b und §16c ARGEBAU (2002).

³⁴ DIBt - FAQ Service (o.A.).

durch eine abZ, aBG oder einer ZiE erfolgen. Ein Inverkehrbringen der Woodbloc Massivholzwand auf europäischer Ebene erfordert aufgrund der nicht-harmonisierung eine ETA.

Im Fall der Woodbloc Massivholzwand liegt eine ETA³⁵ vor, wodurch sie als Bauprodukt auf nationaler und europäischer Ebene unter Einhaltung der in der ETA 24/0946 beschriebenen Anforderungen verwendet werden darf.

2.4. Digitaler Informationsfluss im Holzbau

2.4.1. Hoher Planungsaufwand im Holzbau

Der Holzbau hebt sich durch seinen hohen Planungsaufwand und den vielen Details stark von den meisten anderen Bauweisen ab. So ist es im Holzbau schon in frühen Phasen eines Projektes von entscheidender Bedeutung wesentliche Entscheidungen und Festlegungen zu treffen. Fachplaner sollten bereits nach Fertigstellung des Entwurfs kooperativ zusammenarbeiten und wichtige Anforderungen aufeinander abstimmen. Legt der Architekt in der Entwurfsplanung zum Beispiel die Wandstärke der Außenwand fest, so sollten auch Fachplaner aus dem Bereich der Statik, dem Brandschutz, dem Schallschutz, der Bauphysik (Dämmung, Feuchteschutz, ...), der Technischen Gebäude Ausstattung (TGA) und Weitere direkt mit eingebunden werden und deren Anforderungen an Aufbauten und Mindest- oder Maximalstärken kundgeben. Auch unterscheidet sich der Holzbau zu anderen Bauweisen dadurch, dass bereits in der Planung das Bauen, also Zuschnitt, Vorfertigung und die Montage, mitberücksichtigt werden müssen. Nur so kann verhindert werden, dass es in späteren Phasen zu plötzlichen Änderungen kommt, welche Einfluss auf verschiedene Fachplaner hat und somit hohe Kosten verursachen kann.³⁶

Um trotz des hohen Planungsaufwandes und den meist umfangreichen Ausführungsdetails dennoch gegenüber anderen Bauformen wettbewerbsfähig bleiben zu können, entwickelte sich der Holzbau als innovativer Vorreiter unter den Bauweisen³⁷ und leistete zum Beispiel im Bereich der Vorfertigung große Innovationsschritte durch Ablaufoptimierungen und Automatisierungen, welche mittlerweile auch von anderen Bauweisen wie dem Stb.-Bau als Vorbild gesehen und nachgeahmt werden. Doch Prozessoptimierungen und Automatisierungen finden nicht nur bei der Ausführung statt, denn mit dem Aufkommen der Digitalisierung wurden diese fachübergreifenden Kompetenzen der Planungsphasen immer stärker miteinander verzahnt, sodass auch hier die Prozesse immer weiter miteinander verbunden und optimiert werden konnten. Nach den Worten von Johann Baptist Lindner, dem Vorstandsvorsitzenden der Dietrich's AG, schließt die Digitalisierung „sowohl Planungs-,

³⁵ ETA 24/0946.

³⁶ Gerd Prause (o.A.).

³⁷ Johann Baptist Lindner (2018).

Produktions-, Bestell-, Liefer- und Montageprozesse als auch betriebswirtschaftliche Abläufe ein und umfasst die gesamte Wertschöpfungskette“.³⁸

Im Bereich der Digitalisierung steht also der Umgang mit Daten im Mittelpunkt. Für eine nachhaltige Steigerung der Produktivität ist es daher unumgänglich, den Umgang mit Daten zu untersuchen und zu optimieren. Eine umfangreiche Datengenerierung, sowie die Datenaufbereitung, die Datenbereitstellung und schließlich auch die Verknüpfung von Daten spielen dabei eine zentrale Rolle bei der Entwicklung verbesserter und optimierter Prozessautoabläufe.³⁹ Um zu verstehen, wie es möglich ist den Modellierungsprozess der Woodbloc Massivholzwand von Penzkofer im Umfang der Planungsphase zu automatisieren, ist es notwendig den Informationsfluss von Daten innerhalb von digitalen Umgebungen zu verstehen.

2.4.2. Datenstruktur

Der strukturierte Umgang mit Informationen ist von zentraler Bedeutung. Im Holzbau betrifft dies insbesondere die konsistente Erfassung, Verarbeitung und Weitergabe von Geometrie- und Materialdaten. Vor allem bei der Planung und den darauf aufbauenden Berechnungen einer mechanisch verbundenen CLT-Vollholzwand wie der Woodbloc Massivholzwand besteht ein hoher Bedarf an präzisen und maschinenlesbaren Datenmodellen, da diese Art von Produkten noch wenig Vergleichsmodelle oder auch Modellvorlagen hat und somit eine verstärkte und intensivere Betrachtung erforderlich ist, um Fehler zu vermeiden. Beim Thema Daten ist von grundlegender Bedeutung die spätere Art der Verwendung, für die die Daten aufbereitet werden. Im Umfang dieser Arbeit steht das analytische Modell im Mittelpunkt. Typische Datenkategorien, die für den Modellierungsprozess eines analytischen Modells von Relevanz sind, lassen sich wie folgt einteilen:⁴⁰

- 1) Geometriedaten: Hierzu zählen Abmessungen und Positionen. Diese Daten werden meist mithilfe von kartesischen Koordinaten beschrieben.
- 2) Querschnittsdaten: Diese beschreiben die Dimensionierung von eingesetzten Strukturobjekten (Stäbe, Platten, Scheiben, Volumenkörper).
- 3) Materialdaten: Hierzu zählen beispielsweise Festigkeits- und Steifigkeitswerte des eingesetzten Materials.
- 4) Verbindungsdaten: Typ, Position, Dimension und mechanische Eigenschaften von Verbindungen. Diese können auch durch Gelenke repräsentiert werden.
- 5) Randbedingungen: Lagerungen, Lasten und Zwänge.

Welche Arten von Daten tatsächlich für ein analytisches Modell in Frage kommen, hängt immer vom individuellen Projekt und dem Ziel der Modellierungsaufgabe ab. Diese Daten müssen

³⁸ Johann Baptist Lindner (2018).

³⁹ Heinzmann (2022, S. 83).

⁴⁰ Dlubal Online-Handbuch (2018).

vollständig erfasst, miteinander verknüpft und in geeigneter Weise strukturiert sein. Die Digitalisierung schafft hier die Voraussetzung, um komplexe Modellierungsprozesse softwaregestützt umzusetzen.

2.4.3. Die Rolle von direkten und indirekten Schnittstellen

Ein entscheidender Faktor zur erfolgreichen Umsetzung automatisierter Prozesse ist die Fähigkeit, Daten zwischen verschiedenen Programmen verlustfrei auszutauschen. Hierbei spielen Schnittstellen eine zentrale Rolle. Bei den Schnittstellen lässt sich grundsätzlich zwischen den indirekten, textbasierten Schnittstellen und den direkten Schnittstellen unterscheiden. Bereits in den 1990er Jahren wurde in Deutschland die Notwendigkeit von Schnittstellen zur Übertragung von Modellen mit für die Statik relevanten Daten erkannt. Auf nationaler Ebene entwickelten sich für Tragwerksmodelle anfänglich textbasierte Schnittstellen wie:⁴¹

- 1) IFC Structural Analysis View
- 2) SDNF-Format
- 3) Produktschnittstelle Stahlbau
- 4) CIS/2

Diese Schnittstellen basieren auf dem Austausch von Dateien über Textdateien. Die Kommunikation der Schnittstellen erfolgt indirekt durch den Export der Dateien aus einer Software, gefolgt von einem Import in eine andere Software. Um die Kommunikation zu ermöglichen, müssen beide Software die gleiche „Sprache“ sprechen – die Regeln werden in den genannten Schnittstellenformaten 1) bis 4) festgelegt. Der IFC-Standard (Industry Foundation Classes), innerhalb dessen der IFC Structural Analysis View für Tragwerke zuständig ist, setzte sich mit der Zeit als international meist verwendeter Standard durch und spielt bis heute eine zentrale Rolle bei indirekten Schnittstellen.

Gegenüber diesen indirekten Schnittstellen entwickelten sich auch die direkten Schnittstellen, bei welchen die Softwareprogramme mittels sogenannter Application Programming Interfaces (APIs) direkt miteinander kommunizieren, sodass ein manueller Export gefolgt von einem manuellen Import entfallen. Bei APIs werden Daten mittels einer Client-Server-Kommunikation ausgetauscht. Dies ermöglicht das automatische Integrieren oder auch bereitstellen von Daten auf Grundlage von Regeln und Protokollen, welche vom Server – das ist die Software, welche auf Anfragen vom Client reagiert – festgelegt sind. Der Client ist hingegen der Teil, welcher die Anfragen stellt und eine Reaktion des Servers erwartet. Über die API wird diese Client-Server-Kommunikation ermöglicht.⁴²

⁴¹ Dlubal Online-Handbuch (2018).

⁴² Michael Goodwin (2024).

Der Vorteil von einer direkten API-Kommunikation gegenüber einer indirekten textbasierten Kommunikation liegt im höheren Automatisierungsgrad. API-Schnittstellen ermöglichen es, dass externe Programme wie Excel oder Python (diese haben dann die Funktion des Clients) auf Funktionen und Objekte innerhalb einer bestimmten Software (dem Server), wie beispielsweise RFEM6, zugreifen und mit diesen interagieren können. Für diese Arbeit ist daher besonders die Webservice API von RFEM6 von Bedeutung, da es zum einen dem Ziel der Automatisierung dienlich ist und es zum anderen externen Anwendungen wie Excel oder der Programmiersprache Python erlaubt auf RFEM6 direkt zuzugreifen und mit diesem zu interagieren. Hierdurch können über die Schnittstelle beispielsweise Objekte wie Knoten direkt vom Client (z.B. Python) in ein Modell eingefügt werden. Aufgrund der Einfachheit bei gleichzeitiger Leistungsstärke hat sich Dlubal dazu entschieden besonders Python für diese Schnittstelle zu unterstützen und daher bereits vorgefertigte Python Bibliotheken bereitgestellt,⁴³ welche eine einheitliche Server-Client-Kommunikation zwischen Python und RFEM6 sicherstellen. Diese Kommunikation findet direkt über eine SOAP-Verbindung (Simple Object Access Protocol⁴⁴) statt, einer XML basierten Kommunikation. Dabei besetzt RFEM6 einen bestimmten Port im Netzwerk und regiert auf Kommunikationen innerhalb dieser Ports. Standardmäßig nutzt RFEM6 die Ports 8081 bis 8091.

Sendet Python (Client) nun also im Port 8081 eine Anfrage und nutzt dabei die in den RFEM6 Python Bibliotheken festgelegten Begriffe und Strukturen, so empfängt RFEM6 (Server) diese Daten und weiß automatisch auch ob und wie zu reagieren ist. Dieses Verständnis ist von fundamentaler Bedeutung für das Ziel dieser Arbeit. Mithilfe dieser Schnittstelle ist es möglich, automatisiert auf RFEM6 zuzugreifen und die nötigen Objekte für das analytische Modell der Woodbloc Massivholzwand von Penzkofer automatisch erstellen zu lassen.

2.4.4. Der Workflow: Excel – Python – RFEM6

Um dieses Potential zur Automatisierung bestmöglich für die Zielerreichung dieser Arbeit ausschöpfen zu können, lässt sich folgender Workflow herausarbeiten:

- 1) Dateneingabe: Zu Beginn müssen die relevanten Daten der Woodbloc Massivholzwand eingegeben werden (Geometriedaten wie z.B. Wandabmessungen und Öffnungen). Hierfür bietet sich eine Eingabeoberfläche in Excel an.
- 2) Berechnungsphase: Auf Grundlage der Dateneingabe müssen alle weiteren für die korrekte Modellierung erforderlichen Daten berechnet werden. Hierfür bietet sich ein VBA-Algorithmus innerhalb der Entwicklerumgebung in Excel an. VBA steht für Visual Basic for Applications und ist eine in allen Office Anwendungen integrierte Programmiersprache.

⁴³ Dlubal Online-Handbuch (2022).

⁴⁴ Michael Goodwin (2024).

- 3) Datenübertragung: Die berechneten Daten müssen nun über die API-Schnittstelle von Dlubal nach RFEM6 übertragen werden. Hier bietet es sich an, dass zum einen der VBA-Algorithmus automatisch die berechneten Daten in einer separaten Excel Datei abspeichert und danach von Python diese Daten abgerufen und mithilfe der RFEM6 Python Bibliotheken über die API-Schnittstelle nach RFEM6 übertragen werden.
- 4) Modellierung: Die von RFEM6 empfangenen Daten müssen automatisch in die korrekten Objekte wie Knoten umgewandelt und in das Modell integriert und richtig strukturiert werden.

Durch diesen Ablauf kann mithilfe der von Dlubal bereitgestellten API-Schnittstelle der Modellierungsprozess erheblich automatisiert werden.

2.4.5. Software

Basierend auf diesem Workflow, welcher sich wiederum aus den Möglichkeiten der Automatisierungs- und Schnittstellen Thematik ergibt, wird für diese Arbeit mit folgender Software gearbeitet:

- 1) RFEM6 (Dlubal)

Bei RFEM6 handelt es sich um eine fortschrittliche Statik-Software, welche Berechnungen an allen Tragwerkstypen ermöglicht, vom Einfeldträger bis hin zu komplexen 3D-Schalentragwerken. Das Programm ist modular aufgebaut, dabei werden im Hauptprogramm Stabtragwerke sowie auch räumliche Tragwerke definiert. RFEM6 liefert dann Berechnungen von Verformungen, Schnittgrößen, Spannungen, Lagerkräften und Sohlspannungen. Durch weitere Add-Ons, welche dem Hauptprogramm hinzugefügt werden können, sind auch weitere Analysen und Bemessungen möglich. Dabei sind in RFEM6 alle weltweit gängigen Normen hinterlegt, welche je nach Projektanforderungen ausgewählt werden können. Durch unterschiedliche Assistenten unterstützt RFEM den Nutzer in den unterschiedlichsten Bereichen. Die Software richtet sich an komplexe Projektaufgaben und hat weltweit über 130.000 Nutzer.⁴⁵

- 2) Excel inkl. VBA (Microsoft Office 365)

Excel ist ein Tabellenkalkulationsprogramm und Teil der Microsoft Office 365 Anwendungen. Die Hauptanwendung von Excel umfasst demnach die Datenverarbeitung. Alle Office Anwendungen können Visual Basic for Applications (VBA) anwenden. Dabei handelt es sich um eine einfache und leistungsstarke Programmiersprache, welche bereits in den Umgebungen der Office Anwendungen integriert ist und Automatisierungen, Wiederholungen und Anwendungsübergreifende Interaktionen ermöglichen kann.⁴⁶

⁴⁵ Dlubal Überblick (2021).

⁴⁶ Microsoft Learn (o.A.).

3) Python 3.13

Python ist eine vielfältig einsetzbare Open Source Programmiersprache mit einer einfachen Benutzeroberfläche. Aufgrund seiner breiten Akzeptanz bei der Anwendung und Integration findet Python in den unterschiedlichsten Bereichen von Webservices, Datenanalysen, der Softwareentwicklung bis hin zur Forschung Anwendung.⁴⁷

2.5. Die wichtigsten Erkenntnisse aus Kapitel 2

Im Rahmen der theoretischen Grundlagen des 2. Kapitels wurden zentrale Aspekte für das Verständnis und die Umsetzung des Automatisierungsprozesses zur Modellierung mechanisch verbundener CLT-Elemente herausgearbeitet:

- 1) **Holzbau und Brettsperrholz (CLT)** haben sich durch technologische Entwicklungen, wie die Einführung von flächig verklebtem CLT, deutlich weiterentwickelt. Gleichzeitig wächst die Bedeutung mechanisch verbundener Alternativen, da sie ökologische und gesundheitliche Vorteile bieten. (vgl. Kapitel 2.1)
- 2) **Der Lastabtrag von CLT-Platten** unterscheidet sich je nach Verbindungstyp: Während flächig verklebte Elemente als starre Verbundsysteme behandelt werden können, erfordern mechanisch verbundene Systeme (wie die Woodbloc Massivholzwand von Penzkofer) eine differenzierte Betrachtung mit nachgiebigem Verbund, was sich direkt auf die Modellierungsanforderungen auswirkt. (vgl. Kapitel 2.2)
- 3) **Die mechanischen Verbindungen** wie Schrauben oder Holznägel ersetzen die Klebefugen bei verklebten CLT-Produkten. Das Nagelbild (Anordnung, Anzahl, Abstände und Eigenschaften der VBM) hat erheblichen Einfluss auf die Tragwirkung und muss daher exakt modelliert werden. (vgl. Kapitel 2.2)
- 4) **Die relevanten normativen Grundlagen** zeigen, dass die EN 16351 zwar eine wesentliche Referenz für CLT-Produkte darstellt, derzeit jedoch nicht harmonisiert ist. Mechanisch verbundene Produkte wie die Woodbloc Massivholzwand weichen zudem erheblich von den Anforderungen der EN 16351 ab. Eine Zulassung auf europäischer Ebene ist daher nur über eine ETA möglich – im Fall der Woodbloc Massivholzwand liegt diese bereits mit der ETA 24/0946 vor. (vgl. Kapitel 2.3)
- 5) **Digitalisierung und Automatisierung** im Holzbau gewinnen zunehmend an Bedeutung. Die strukturierte Datenverarbeitung, insbesondere für analytische Modelle, ermöglicht Automatisierungspotenziale entlang der gesamten Planungskette. (vgl. Kapitel 2.4)
- 6) Ein zentraler Aspekt bei der Automatisierung ist **die Nutzung von API-Schnittstellen** wie der Webservice API von Dlubal für RFEM6, welche die direkte Interaktion zwischen

⁴⁷ Python.org (o.A.).

Softwaretools wie Excel/VBA, Python und RFEM6 ermöglichen. Damit lassen sich Daten zur Geometrie, dem Material, VBM, Randbedingungen und Weiteres gezielt verarbeiten und modellieren. (vgl. Kapitel 2.4)

Diese Erkenntnisse bilden das theoretische Fundament für die nachfolgende Entwicklung eines automatisierten Modellierungsprozesses und machen deutlich, warum eine Automatisierung besonders bei mechanisch verbundenen CLT-Produkten notwendig und gleichzeitig technisch herausfordernd ist.

3. Entwicklung des Automatisierungsprozesses

Die funktionalen Anforderungen bezüglich der Konfigurationen und Eigenschaften, die sich später im analytischen Modell wieder finden müssen, sind aus dem ETA 24/0946, dem europäischen Bewertungsdokument des Bauprodukts Woodbloc Massivholzwand von Penzkofer, herauszuarbeiten. Weiter sind auch mit dem ETA 24/0946 in Verbindung stehende ETAs zu berücksichtigen, sofern sie in irgendeiner Form Bestandteil der Woodbloc Massivholzwand sind. Hierzu zählen das ETA 23/0041 (die europäische Bewertung für die in der Massivholzwand verwendeten Nägel) sowie das ETA 11/0190 für die verwendeten Schrauben.

3.1. Analyse des ETA 24/0946 der Woodbloc Massivholzwand

Die Woodbloc Massivholzwand wurde vom Österreichischen Institut für Bautechnik nach den Anforderungen des europäischen Bewertungsdokuments „Massive plattenförmige Holzbauelemente – Elemente aus mit Dübeln für tragende Bauteile in Bauwerken“⁴⁸ bewertet und für den Hersteller Penzkofer Bau GmbH ausgestellt. Bei der Woodbloc Massivholzwand handelt es sich um ein mechanisch verbundenes Vollholzelement aus sieben Lagen Nadelholzbrettern (europäische Fichte (*Picea abies*) oder gleichwertiges Nadelholz), welches mit Holznägeln mittig und Schrauben im Randbereich zu einem massiven Element verbunden wird und für tragende sowie nichttragende Zwecke in der Nutzungsklasse 1 und 2 vorgesehen ist.⁴⁹

3.1.1. Aufbau der Woodbloc Massivholzwand

Die einzelnen Brettlagen sind unterschiedlich ausgerichtet und haben je nach Ausrichtung die Abmessungen (breite x tiefe) 184,8mm x 22,8mm oder 160mm x 22,8mm, wobei eine Toleranz von ± 2 mm zulässig ist. Weiter sind die Bretter gehobelt und ohne Keilzinkverbindung, sowie auch ohne Stumpfstöße auszuführen. Damit die Wand Kräfte aufnehmen und weiterleiten kann, sind am oberen und am unteren Ende der Wand zusätzlich Laschenbretter angebracht. Ausrichtungen und Abmessungen sind der Tabelle 1 zu entnehmen.

Position	Abmessungen [mm]	Beschreibung
Lage 1 (90°)	b = 184,8 ^{±2} , t = 22,8 ^{±1}	C16 gemäß EN338 ⁵⁰
Lage 2 (-60°)	b = 160 ^{±2} , t = 22,8 ^{±1}	C16 gemäß EN338
Lage 3 (90°)	b = 184,8 ^{±2} , t = 22,8 ^{±1}	C16 gemäß EN338
Lage 4 (60°)	b = 160 ^{±2} , t = 22,8 ^{±1}	C16 gemäß EN338
Lage 5 (90°)	b = 184,8 ^{±2} , t = 22,8 ^{±1}	C16 gemäß EN338

⁴⁸ EAD 130002-00-0304.

⁴⁹ ETA 24/0946, S. 3.

⁵⁰ DIN EN 338:2016-07. (2016)

Lage 6 (0°)	$b = 160^{\pm 2}, t = 22,8^{\pm 1}$	C16 gemäß EN338
Lage 7 (90°)	$b = 184,8^{\pm 2}, t = 22,8^{\pm 1}$	C16 gemäß EN338
Holznägel	LignoLoc: $d = 4,7, l = 90$	gemäß ETA 23/0041
Schrauben	Würth Assy 4: $d=6, l = 150$	gemäß ETA 11/0190
Laschenbrett oben	$b = 120, t = 60$	C24 gemäß EN338
Laschenbrett unten	$b = 120, t = 60$	C24 gemäß EN338

Tabelle 1 Komponenten der Woodbloc Massivholzwand (eigene Darstellung nach ETA 24/0946, S. 3)

3.1.2. Nagelbild nach ETA 24/0946

Das Nagelbild definiert die Positionen der Verbindungsmittel (Nägel und Schrauben), mit denen die Brettlagen miteinander verbunden werden und stellt so sicher, dass die statischen Anforderungen und Tragfähigkeiten gewährleistet werden.

Das ETA gibt in dessen Kapitel 2.2 einen textlichen Hinweis und weiter mit den darauffolgenden Abbildungen die Anforderungen an das Nagelbild. Nach der Beschreibung ist „je Kreuzungspunkt (...) mindestens 1 Holznagel erforderlich“⁵¹, jedoch wird nicht weiter auf die genaue Definition eines Kreuzungspunktes im Zusammenhang mit dieser Zulassung eingegangen. Es bleibt textlich erstmal unklar, ob es sich hierbei um den Schnitt der Stabachsen von Brettern handelt oder ob ein Schneiden der Querschnittsflächen von Brettern ausschlaggebend ist. Weiter wird eine Abbildung mit dem generellen Verbindungsmuster gegeben, worauf die Anordnung von zwei Nagelungen, den Schrauben und der Bretter erkenntlich ist (Abbildung 3).

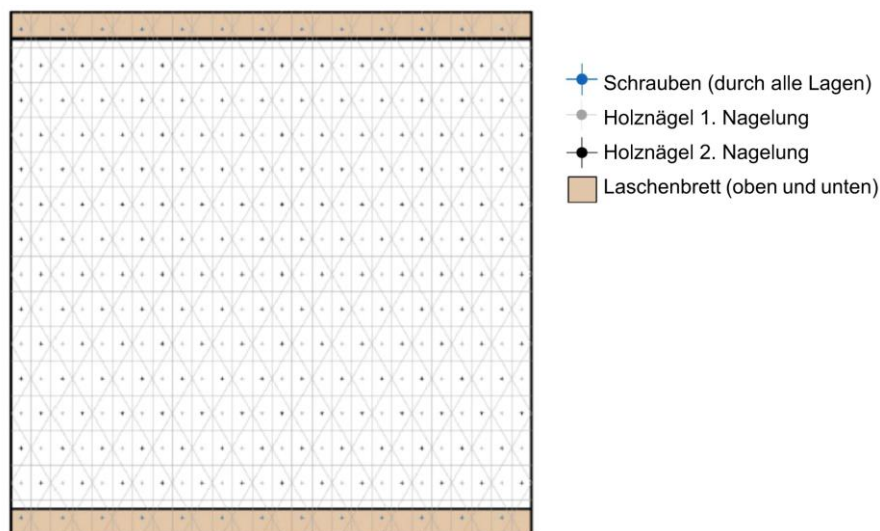


Abbildung 3 Abbildung aus dem ETA zum Nagelbild (ETA 24/0946 vgl. Seite 12 Abbildung)

Aus dieser Abbildung lässt sich wiederum schließen:

- 1) Die zwei Nagelungen sind mit einem Rasterversatz zueinander angeordnet (erkenntlich durch die versetzten Lagen der hell- und dunkelgrauen Markierungen). Dies wird durch die Abbildung 4 bestätigt.

⁵¹ ETA 24/0946, S. 4.

- 1 Nagelung: Durchdringt Lage 1 bis 4
- 2 Nagelung: Durchdringt Lage 4 bis 7

In der 4. Lage kommt es zur Überschneidung der ersten und der zweiten Nagelung. Zwar sind die Nägel im Raster versetzt und somit nicht direkt übereinander, aber dennoch werden durch diese Überschneidung in Lage 4 die Kräfte von der ersten Nagelung zur zweiten Nagelung übertragen.

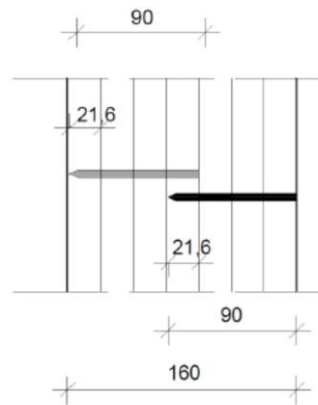


Abbildung 4 Nagelmuster für Holznägel (ETA 24/0946 vgl. Seite 12 Abbildung 3)

- Die Markierungen von VBM sind nicht an den Stellen, wo sich die Linien kreuzen (Kreuzungspunkte), sondern mittig davon.
- Es wird nicht näher beschrieben, ob es sich bei den Linien um die Stabachsen der Bretter, der unterschiedlichen Lagen oder um die äußeren Brettanten der Bretter handelt. Zwar könnte die farbliche Hervorhebung der Laschenbretter darauf schließen lassen, dass es sich um die Flächen der Bretter handelt (äußere Linien begrenzen farblich markierte Flächen und wären somit die Brettanten und keine Stabachsen), jedoch ist die Lage der Laschenbretter nicht mit dem Raster der 0° Brettlagen abgestimmt und lässt somit ein nachträgliches und von den restlichen Lagen unabhängiges Einfügen der Laschenbretter vermuten.

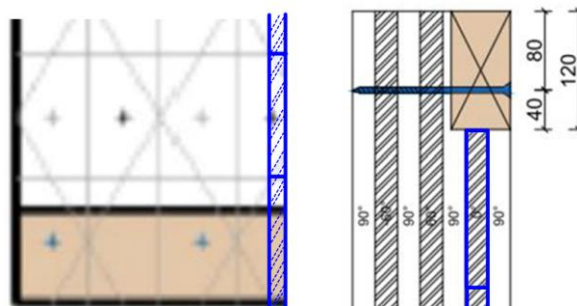


Abbildung 5 Gegenüberstellung der Details aus dem ETA. links aus Abbildung 2 (ETA 24/0946 vgl. Seite 12 Abbildung 2), rechts (ETA 24/0946 vgl. Seite 14 Abbildung 6) aus Abbildung 6. Blau markiert die jeweils vorhandene Lage der 0° Bretter

Durch die blaue Hervorhebung in Abbildung 5 ist ersichtlich, dass es hier Unstimmigkeiten in der linken Abbildung gibt. Auch unter Berücksichtigung der zweiten Möglichkeit, dass es sich bei den Linien um die Stabachsen der Bretter handeln könnte, geht das Raster nicht auf. Die rechte Darstellung hingegen wird korrekt sein.

- 4) Die 90° Linien sind, unabhängig davon, ob es sich hierbei um die repräsentativen Stabachsen oder die äußeren Brettanten handelt, in doppelter Menge vorhanden. Bei der Betrachtung des Details aus Abbildung 6 wird dies ersichtlich.

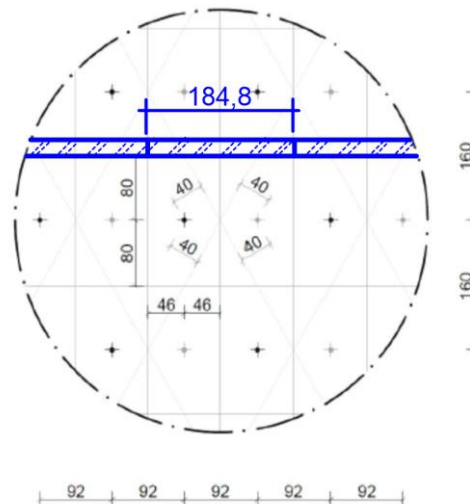


Abbildung 6 Detail Nagelmuster (ETA 24/0946vgl. Seite 12 Abbildung 3)

Hier wurde wieder Blau markiert, wie die 90° Bretter mit ihrer entsprechenden Abmessung nach Tabelle 1 liegen. Die aus dem Detail ersichtlichen 90° Linien (senkrecht) sind in einem Raster von 92mm vorhanden. Dies entspricht etwa der Hälfte einer Brettbreite (die Differenz von 0,8mm kann auf Rundungsfehler und der Toleranz von $\pm 2\text{mm}$ zurückgeführt werden). Auch diese Feststellung ist wieder unabhängig der Tatsache, ob es sich bei den 90° Linien um die repräsentativen Stabachsen der Bretter oder der äußeren Brettanten handelt.

Unter Berücksichtigung aller analysierter Punkte aus dem ETA, vor allem mit Einbezug der Textpassage „je Kreuzungspunkt (..) mindestens 1 Holznagel“⁵², liegt die größte Plausibilität in der Darstellung Abbildung 3 darin, dass die dargestellten Linien der Außenkante der Bretter entsprechen, die 90° liegenden Bretter halbiert wurden und die repräsentativen Stabachsen mittig der Bretter zu verlaufen haben. Dadurch wird die Bedingung erfüllt, dass die Nägel genau im Kreuzpunkt der Stabachsen der Bretter liegen, womit ein analytisch sinnvolles Modell geschaffen würde (Abbildung 7). Dabei fällt jedoch auf, dass sich an den Kreuzungspunkten nur die Stabachsen der 90° und der 0° Bretter kreuzen. Die Stabachsen der $\pm 60^\circ$ Bretter kreuzen sich an anderer Stelle.

⁵² ETA 24/0946, S. 4.

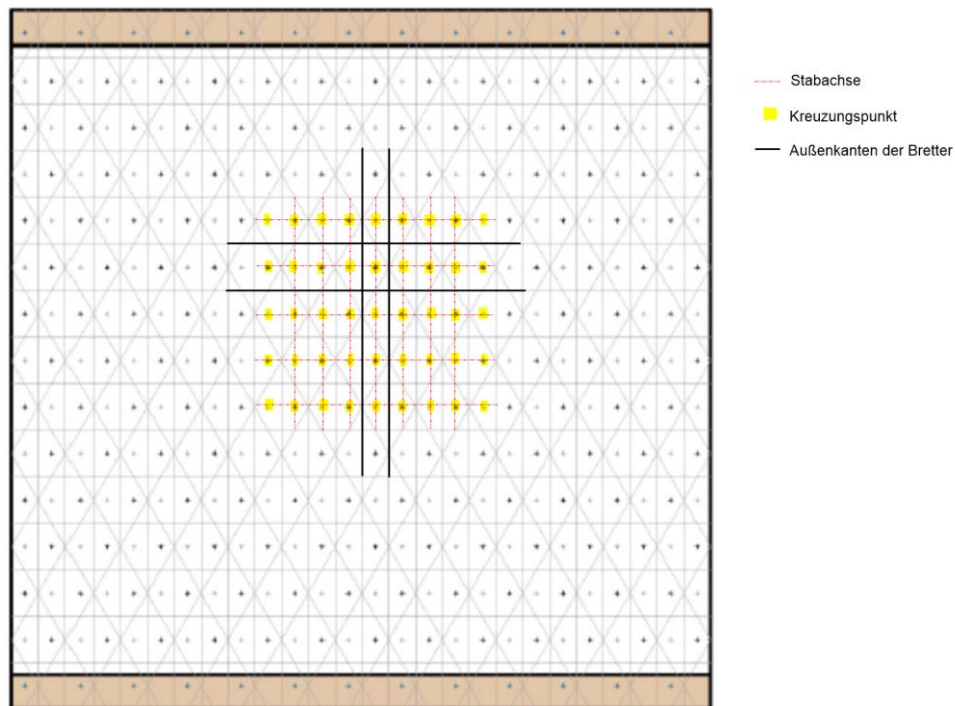


Abbildung 7 Darstellung der Stabachsen und Kreuzungspunkte (eigene Darstellung nach ETA 24/0946Seite 12 Abbildung 2)

3.1.3. Abstände der VBM

In weiteren Details aus dem ETA werden zudem die unterschiedlichen Abstände der Verbindungsmittel (VBM) zueinander und auch zu den Rändern des Elementes hin ersichtlich. Das Raster lässt sich in einen horizontalen und einen vertikalen Bereich einteilen. Dabei werden die vertikalen Abstände zwischen den VBM mit einem Abstand zueinander feldmittig von 160mm und am Rand, dies betrifft die erste und letzte Rasterreihe, mit einem Randabstand von 80mm gesetzt. In Abbildung 8 ist dies ersichtlich.

Die horizontalen Abstände hingegen sind innerhalb einer Nagelung feldmittig im Raster von 184,8mm⁵³ und an den Randbereichen mit einem Abstand von entweder 46mm oder 138mm angegeben. Der Randabstand wechselt immer zwischen 46mm und 146mm und hängt mit dem Rasterversatz der ersten und der zweiten Nagelung zusammen. In Abbildung 8 sind alle Abstände dargestellt.

⁵³ die Angaben aus dem ETA variieren zwischen 185mm und 184mm, dies kann aber auf Rundungen und der zulässigen Toleranz zurückgeführt werden. 184,8mm ist die Breite der 90° Bretter und somit auch ausschlaggebend für das Raster der VBM

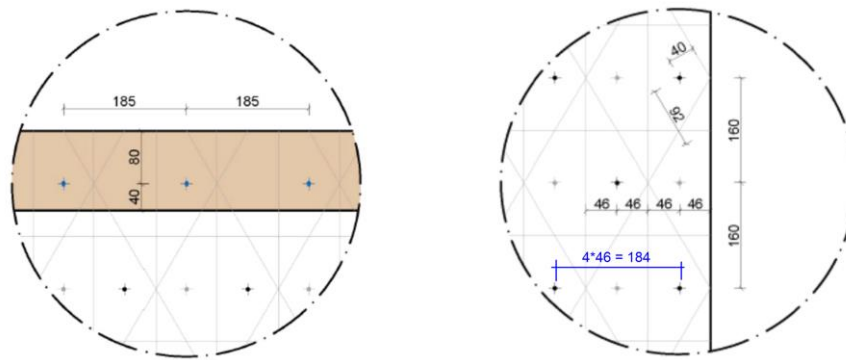


Abbildung 8 Detail zu den vertikalen und horizontalen Rasterabständen der VBM (ETA 24/0946 vgl. Seite 13 Abbildungen 4 und 5)

3.1.4. Zusammenfassung der Anforderungen an das analytische Modell

Ziel ist es, ein analytisches Modell zu erstellen, welches ein möglichst realistisches Abbild der realen Verhaltensweise der Woodbloc Massivholzwand liefert. Hierfür müssen die erfassten Ergebnisse der Analyse des ETA 24/0946 umgesetzt werden.

Im Umfang dieser Arbeit werde zwei Ansätze verfolgt:

1) Modellansatz 1: Realmodell

Dieser Modellansatz verfolgt ein Stabmodell mit einen möglichst korrekten Aufbau nach den Angaben aus dem ETA.

Brettaufbau:

- Raster der Stäbe gemäß Abbildung 7
- 7 Schichten mit den Ausrichtungen der Bretter nach Tabelle 1
- Querschnitte nach Tabelle 2

Lage	Querschnitt b x t [mm]	Bemerkung
90°	92,4 x 22,8	$\frac{184,8}{2} \text{ mm} = 92,4 \text{ mm}$
0°	160 x 22,8	
+/- 60°	160 x 22,8	
Laschen	160 x 60	Die Breite wurde von 120mm auf 160mm geändert, damit die Stabachsen auf gleicher Höhe wie die Schrauben liegen (80mm Randabstand)

Tabelle 2 Querschnitte Modellansatz 1

Nagelbild:

- 2 Nagelungen mit abwechselnden Rasterabstand
- Abstände nach Tabelle 3

Abstand..	Maß [mm]	Bemerkung
vertikaler Randbereiche	46,2	die Nagellagen wechseln sich nach jeder Reihe mit dem Randabstand ab sodass ein Rasterversatz entsteht
	138,6	
horizontaler Randbereiche	80	erste und letzte Reihe wird verschraubt (dort liegen auch die Laschenbretter) mit einem Randabstand von 80mm
vertikal im Feld	160	
horizontal im Feld	92,4	unter beachtung beider Nagelungen
	184,8	innerhalb einer Nagelung

Tabelle 3 Abstände im Nagelbild beim Modellansatz 1

2) Modellansatz 2: Idealmodell

Dieser Modellansatz verfolgt ein Stabmodell, das dahingehend idealisiert wird, dass die Kräfte im Modell besser zwischen den einzelnen Brettlagen übertragen werden. Das Raster der 90° und der 0° Bretter wird so verschoben, dass sich alle Brettlagen inklusive der $\pm 60^\circ$ Lagen an einem gemeinsamen Kreuzungspunkt schneiden. Die 90° Bretter müssen nicht mehr in der Breite auf die Hälfte (92,4mm) reduziert werden, sondern entsprechen auch im analytischen Modell ihrer tatsächlichen Breite von 184,8mm. Weiter wird das Nagelbild im analytischen Modell optimiert:

In der 4. Brettlage haben die erste und die zweite Nagelung ihre gemeinsame Ebene, über welche die Nagelungen in der Realität verbunden sind. Im analytischen Modell hingegen werden bei realistischer Nachmodellierung (Modellansatz 1) die Kräfte aufgrund des Rasterversatzes zwischen den beiden Nagelungen nicht übertragen (kein gemeinsamer Knoten zwischen erster und zweiter Nagelung). Um dies zu beheben, gibt es unterschiedliche Möglichkeiten wie zum Beispiel das Koppeln der Nägel untereinander oder das Ersetzen der zwei im Raster zueinander versetzten Nagelungen durch eine einzige durchgängige Nagelung. Im Modellansatz 2 werden die zwei getrennten Nagelungen durch eine durchgängige Nagelung ersetzt. Weiter wird das Raster der VBM dahingehend angepasst, dass die Verbindungsmittel mit allen Kreuzungspunkten der Brettlagen übereinstimmen.

Die Vereinfachungen des Modellansatz 2 können der Abbildung 9 entnommen werden.

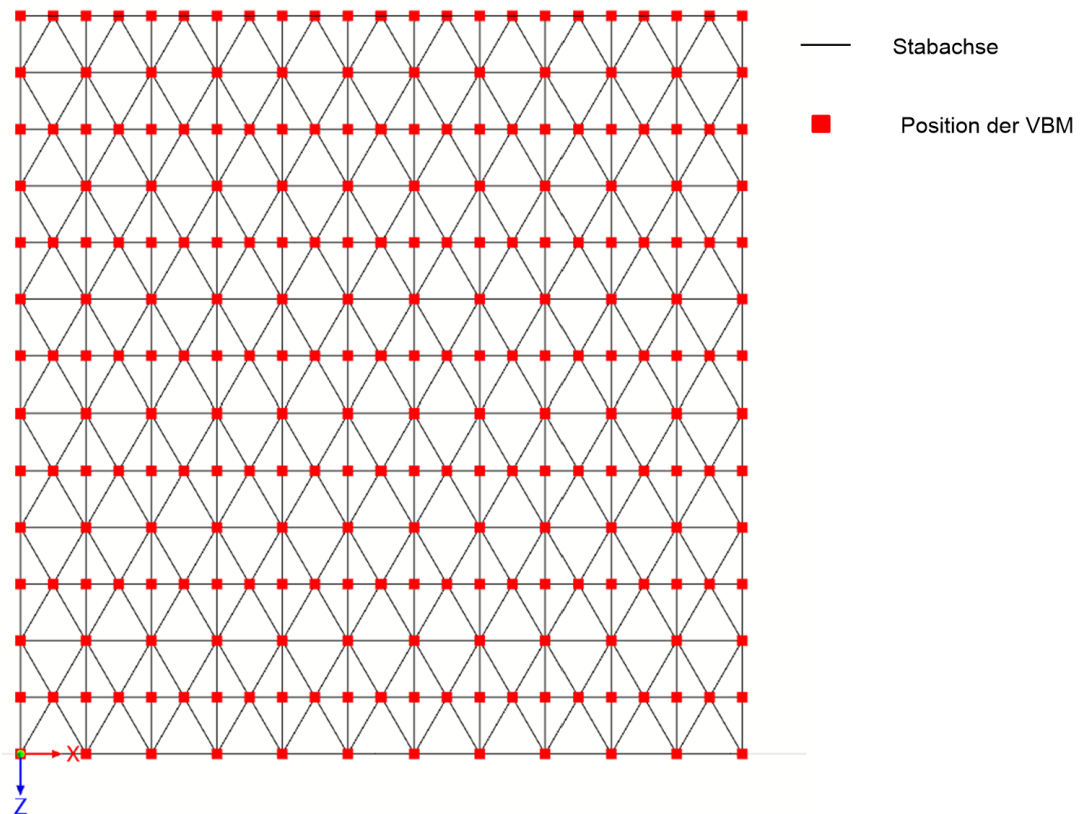


Abbildung 9 Raster der Brettlagen und VBM im Modellansatz 2

Brettaufbau:

- Angepasstes Raster mit gemeinsamen Kreuzungspunkten aller Lagen
- 7 Lagen mit den Ausrichtungen der Bretter nach Tabelle 1
- Querschnitte nach Tabelle 4

Lage	Querschnitt b x t [mm]	Bemerkung
90°	184,8 x 22,8	/
0°	160 x 22,8	/
+ - 60°	160 x 22,8	/
Laschen	120 x 60	/

Tabelle 4 Querschnitte Modellansatz 2

Nagelbild:

- Eine durch alle Lagen durchgängige Nagellage
- Abstände nach Tabelle 5

Abstand..	Maß [mm]	Bemerkung
vertikal	160	/
horizontal	92,4	In Reihen mit Kreuzungspunkten alle 92,5mm
	184,8	In Reihen mit Kreuzungspunkten alle 184,8mm

Tabelle 5 Abstände im Nagelbild beim Modellansatz 2

Für diese beiden Modellansätze sind nun die Anforderungen definiert, aufbauend hierauf werden zwei Modelle mit unterschiedlichen Algorithmen erstellt, welche den jeweils hier definierten Anforderungen unterliegen. Weiter ist es von genauso großer Relevanz, den genutzten Verbindungsmitteln zugrunde liegenden ETAs genauer zu betrachten. Dies ermöglicht eine korrekte Konfiguration und Einbeziehung der Nachgiebigkeit der analytischen Modelle.

3.2. Verbindungsmittel und Schubverhalten der Woodbloc Massivholwand

Für die Herstellung der Woodbloc Massivholzwand wird bei den Verbindungsmitteln auf die Produkte „LignoLoc stiftförmige Verbindungsmittel aus Holz“⁵⁴ und den „Würth selbstbohrende Schrauben“⁵⁵ zurückgegriffen. Wie bereits aus Kapitel 2.2 hervorgeht, spielt beim Erstellen des analytischen Modells zur Woodbloc Massivholzwand die Steifigkeit der Verbindungsmittel eine wichtige Rolle, da nur durch eine korrekte Berücksichtigung dessen die Nachgiebigkeit der Wand realistisch abgebildet wird. Daher wird bei der Betrachtung der verwendeten Verbindungsmittel zwischen den einzelnen Schichten ein besonderer Fokus auf die in den jeweiligen ETAs angegebene $K_{ser,lateral}$ Werte gelegt. Mithilfe dieses Wertes soll bei der Konfiguration des analytischen Modells jeweils das Schubmodul der Verbindungsmittel zwischen den Schichten modifiziert werden, sodass die Nachgiebigkeit berücksichtigt wird. Das fiktive Schubmodul kann mit

$$G_{fiktiv} = \frac{K_{ser} \cdot l}{A} [N/mm^2] \quad (3.1)$$

berechnet werden. Dabei ist

K_{ser}	Steifigkeit der Verbindung
l	Länge l des Verbindungsmittels pro Scherfuge
A	Querschnitt des Verbindungsmittels

⁵⁴ ETA 23/0041.

⁵⁵ ETA 11/0190.

3.2.1. LignoLoc Holznägel (ETA 23/0041)

Die LignoLoc Nägel der Firma Raimund Beck bestehen aus verdichtetem Buchenschichtholz nach EN 61061-3-1⁵⁶ mit einer charakteristischen Mindestrohdichte von 1100 kg/m³. Sie verfügen über einen glatten, zylindrischen Schaft und sind kopflos. Weiter sind die Nägel ohne Vorbohren senkrecht zur Faserrichtung in die tragende Holzkonstruktion einzubringen. Das ETA umfasst Holznägel mit Durchmessern zwischen 2,8mm bis 5,3mm und Längen von 34mm bis 130mm.⁵⁷ Für die Woodbloc Massivholzwand werden Nägel mit einem Durchmesser von 4,7mm und einer Länge von 90mm verwendet. Gemäß dem ETA 23/0041 sind die Nägel zur Herstellung von Holz-Holz- oder Platten-Holz-Verbindungen in tragenden Holzkonstruktionen mit Bauteilen aus Nadelholz vorgesehen. Damit erfüllen sie die Anforderungen um als Verbindungsmittel (VBM) in der Woodbloc Massivholzwand verwendet zu werden.

Nach dem ETA 23/0041 werden im Abschnitt „mechanische Festigkeit und Standsicherheit“ für einen LignoLoc Nagel mit einem Durchmesser $d = 4,7\text{mm}$ die Werte mit

- Biegebruchmoment $M_{u,k} = 2200\text{Nmm}$
- Steifigkeit der Verbindung $K_{ser} = \frac{F_{v,Rk}}{0,3\text{mm}}$ (3.2)
- Zugtragfähigkeit $f_{tens,k} = 1,4\text{kN}$

angeben.⁵⁸ Neben diesen Angaben finden sich weitere Angaben Im Abschnitt „mechanische Festigkeit und Stabilität“⁵⁹ wie

- Ausziehtragfähigkeit
- Tragfähigkeit rechtwinklig zur Achsrichtung

$$F_{v,Rd} = \sqrt{\frac{2 \cdot \beta}{1 + \beta}} * \sqrt{1,5 * M_{u,d} * f_{h,1,d} * d} * \min \left\{ \begin{matrix} 1 \\ t_1/t_{1,req} \\ t_2/t_{2,req} \end{matrix} \right. [N] \quad (3.3)$$

Dabei ist:

$f_{h,1,d}$	Bemessungswert der Lochleibungsfestigkeit
$M_{u,d}$	Bemessungswert des Biegebruchmoments des LignoLoc Nagels
β	Verhältnis der Lochleibungsfestigkeit 2 zur Lochleibungsfestigkeit 1
t_1	Eindringtiefe des Nagels in Bauteil 1
$t_{1,req}$	Erforderliche Dicke des Bauteils 1

⁵⁶ DIN EN 61061-3-1.

⁵⁷ ETA 23/0041, S. 3.

⁵⁸ ETA 23/0041, 4–9.

⁵⁹ ETA 23/0041, S. 5.

t_2 Eindringtiefe einschließlich Spitze in Bauteil 2

$t_{2,req}$ Erforderliche Dicke des Bauteils 2

Für den Umfang dieser Arbeit sind die Steifigkeit der Verbindung (Gleichung 3.2) und die Tragfähigkeit rechtwinklig zur Achsrichtung (Gleichung 3.3) maßgebend. Da Gleichung 3.3 die Tragfähigkeit rechtwinklig zur Achsrichtung berechnet und dann in Gleichung 3.2 weiterverwendet wird, entspricht das Ergebnis aus (3.2) dem lateralen Verschiebungsmodul.

In den folgenden Berechnungen muss an verschiedene Stellen mit dem Modifikationsbeiwert k_{mod} gerechnet werden. Hierzu ist anzumerken:

Der Modifikationsbeiwert k_{mod} muss entsprechend der Lasteinwirkungsdauer ($KLED$) und der Nutzungsklasse (NKL) des Bauteils gewählt werden. Aus dem ETA 24/0946 der Woodbloc Massivholzwand geht hervor, dass die Zulässigkeit nur für die NKL 1 und 2 freigegeben ist.⁶⁰ Da die $KLED$ jedoch je nach vorgesehenem Einsatz der Woodbloc Massivholzwand variieren kann, muss bei der Eingabeoberfläche des Automatisierungsprozesses dieser Arbeit die Möglichkeit zur Auswahl aller $KLED$ nach EC5⁶¹ gegeben sein. Im Umfang dieser beispielhaften Berechnung in diesem Abschnitt wird mit der Lasteinwirkungsdauer mittel weitergerechnet, sodass $k_{mod} = 0,8$ angesetzt werden kann. Damit können nun die erforderlichen Komponenten zur Lösung von (3.3) berechnet werden.

Zur Lösung der Gleichung (3.3) muss zuerst die Lochleibungssteifigkeit mit

$$f_{h,1,d} = \frac{f_{h,1,k} * k_{mod,1}}{\gamma_m} = \frac{11,25 \frac{N}{mm^2} * 0,8}{1,3} = 6,92 \text{ N/mm}^2 \quad (3.4)$$

ermittelt werden. Basierend auf dem charakteristischen Wert der Lochleibungssteifigkeit

$$f_{h,1,k} = \frac{0,082 * \rho_k * d^{-0,3}}{(1,35 + 0,015 * d) \sin^2 \alpha + \cos^2 \alpha} = \frac{0,082 * 310 \frac{kg}{m^3} * 4,7 mm^{-0,3}}{(1,35 + 0,015 * 4,7) \sin^2 90 + \cos^2 90} = 11,25 \text{ N/mm}^2 \quad (3.5)$$

Für Bauteil 2 können identische Werte angesetzt werden, sodass

$$f_{h,2,d} = f_{h,1,k} = \frac{f_{h,1,k} * k_{mod,1}}{\gamma_m} = \frac{11,25 \frac{N}{mm^2} * 0,8}{1,3} = 6,92 \text{ N/mm}^2 \quad (3.6)$$

Die Dichte ρ_k wurde für C16 (Bauteil 1 und 2) gemäß DIN EN 338 mit $310 \frac{kg}{m^3}$ gewählt. Weiter wurden der Durchmesser $d = 4,7 mm$ sowie $\alpha = 90^\circ$ aufgrund der Nagelung senkrecht zur Faser gewählt.

⁶⁰ ETA 24/0946, S. 4.

⁶¹ DIN EN 1995-1-1:2023, S. 73 Tabelle 5.3.

Mit den Lösungen aus (3.4) und (3.6) ergibt sich für das Verhältnis β

$$\beta = \frac{6,92 \text{ N/mm}^2}{6,92 \text{ N/mm}^2} = 1 \quad (3.7)$$

Der Bemessungswert des Biegebruchmoments des LignoLoc Nagels

$$M_{u,d} = \frac{M_{u,k} * k_{mod,M}}{\gamma_M} = \frac{2200 \text{ Nmm} * 0,5}{1,3} = 846,15 \text{ Nmm} \quad (3.8)$$

verwendet den Modifikationsbeiwert $k_{mod,M} = 0,5$, sowie das charakteristische Biegebruchmoment $M_{u,k} = 2200 \text{ Nmm}$ ⁶² und den Teilsicherheitsbeiwert $\gamma_M = 1,3$ für Holz.

Anhand des Details der Woodbloc Massivholzwand zu den Eindringtiefen der Nägel sind zwei Werte für die Eindringtiefen zu betrachten (vgl. Abbildung 4). Zum einen für die Situation, in der der Nagel komplett durch eine Brettlage verläuft (in diesem Fall beträgt die Eindringtiefe die volle Brettdicke von 22,8mm) und zum anderen die Situation an der Nagelspitze (in diesem Fall beträgt die Eindringtiefe 21,6mm). Um auf der sicheren Seite zu sein, wird der kleinere Wert von 21,6mm bei den Eindringtiefen t_1 und t_2 angesetzt.

Die erforderlichen Bauteildicken für Bauteil 1 und 2 lassen sich gleichermaßen berechnen mit

$$\begin{aligned} t_{1,req} &= t_{2,req} = \left(\sqrt{\frac{\beta}{1+\beta}} + 1 \right) * \sqrt{\frac{4 * M_{u,d}}{0,75 * f_{h,i,d} * d}} \\ &= \left(\sqrt{\frac{1}{1+1}} + 1 \right) * \sqrt{\frac{4 * 846,15 \text{ Nmm}}{0,75 * 6,92 \text{ N/mm}^2 * 4,7 \text{ mm}}} = 20,11 \text{ mm} \end{aligned} \quad (3.9)$$

Zusammenfassend mit den Werten aus den Gleichungen (3.4) bis (3.9) lässt sich

$$\begin{aligned} F_{v,Rd} &= \sqrt{\frac{2 * \beta}{1+\beta}} * \sqrt{1,5 * M_{u,d} * f_{h,1,d} * d} * \min \left\{ \begin{matrix} 1 \\ t_1/t_{1,req} \\ t_2/t_{2,req} \end{matrix} \right\} [N] \\ &= \sqrt{\frac{2 * 1}{1+1}} * \sqrt{1,5 * 846,15 \text{ Nmm} * 6,92 \text{ N/mm}^2 * 4,7 \text{ mm}} * \min \left\{ \begin{matrix} 1 \\ 21,6 \text{ mm} / 20,11 \text{ mm} \\ 21,6 \text{ mm} / 20,11 \text{ mm} \end{matrix} \right\} [N] \\ &= 203,23 [N] \end{aligned} \quad (3.10)$$

berechnen. Dies führt zu einer anzusetzenden lateralen Steifigkeit von

$$K_{ser} = \frac{F_{v,Rk}}{0,3 \text{ mm}} = \frac{203,23 \text{ N}}{0,3 \text{ mm}} = 677,37 \frac{\text{N}}{\text{mm}} \quad (3.11)$$

womit das fiktive Schubmodul aus Gleichung (3.1) zur Berücksichtigung der Nachgiebigkeit der Woodbloc Massivholzwand mit folgendem Wert angenommen werden kann

⁶² ETA 23/0041, S. 5 Tabelle 2 und Anhang B Tabelle B.1.

$$G_{fiktiv,nagel} = \frac{677,37 \frac{N}{mm} * 22,8 mm}{17,35 mm^2} \left[\frac{N}{mm^2} \right] = 890,15 \left[\frac{N}{mm^2} \right] \quad (3.11)$$

3.2.2. Würth Assy 4 Schrauben (ETA 11/0190)

Bei den Würth Assy Schrauben handelt es sich um selbstbohrende Holzschrauben aus Kohlenstoffstahl oder nichtrostendem Stahl mit Gewindedurchmessern von 3mm bis hin zu mehr als 14mm. Gemäß der technischen Beschreibung des Produkts liegt die Gesamtlänge der Schraube zwischen 13mm und 2000mm.⁶³ Bei der Woodbloc Massivholzwand kommt in den Randbereichen eine Teilgewindeschraube Würth Assy 4 mit den Abmessungen 6,0 x 150mm zum Einsatz.

Identisch wie bei den LignoLoc Nägeln soll auch für die Schrauben ein fiktives Schubmodul nach Gleichung (3.1) ermittelt werden, um die Schubsteifigkeit der Woodbloc Massivholzwand korrekt im analytischen Modell abbilden zu können. Das ETA 11/0190 gibt zwar für Beanspruchungen in Achsrichtung der Verbindungsmittel eine Gleichung für die axiale Steifigkeit mit

$$K_{ser,axial=25 * d * l_{ef} \left[\frac{N}{mm} \right]} \quad (3.12)$$

an⁶⁴, nicht aber für die laterale Steifigkeit $K_{ser,lateral}$. Diese muss nach dem Eurocode 5 gemäß der Tabelle 11.13⁶⁵ mit

$$K_{ser,lateral} = \frac{\rho_{mean}^{1,5} * d}{23} N/mm \quad (3.13)$$

berechnet werden. Nach dem ETA 11/0190 sind die Gewindeaußendurchmesser als wirksame Durchmesser der Schraube in Übereinstimmung mit dem EN 1995-1-1 zu verwenden.⁶⁶ Damit ergibt sich für die Schraube im Randbereich der Woodbloc Massivholzwand eine laterale Steifigkeit von

$$K_{ser,lateral} = \frac{\rho_{mean}^{1,5} * d}{23} \frac{N}{mm} = \frac{370^{1,5} * 6}{23} = 1856,63 N/mm \quad (3.14)$$

Dabei wird für ρ_{mean} die mittlere Dichte von Nadelholz C16 gemäß DIN EN 338⁶⁷ mit 370 kg/m³ verwendet. Hiermit kann nun identisch zum fiktiven Schubmodul des Nagels auch das fiktive Schubmodul der Würth Schrauben mit der Gleichung (3.15) berechnet werden.

$$G_{fiktiv,schraube} = \frac{K_{ser} * l}{A} \left[\frac{N}{mm^2} \right] = \frac{1856,63 \frac{N}{mm} * 22,8 mm}{28,27 mm^2} \left[\frac{N}{mm^2} \right] = 1497,39 \left[\frac{N}{mm^2} \right] \quad (3.15)$$

⁶³ ETA 11/0190, S. 3.

⁶⁴ ETA 11/0190, S. 12.

⁶⁵ DIN EN 1995-1-1:2023, S. 206 Tabelle 11.13 (5).

⁶⁶ ETA 11/0190, S. 10.

⁶⁷ DIN EN 338:2016-07, S. 7 Tabelle 1.

3.3. Entwicklung der Eingabeoberfläche

Die Eingabeoberfläche dient dem Nutzer, um die Randbedingungen so weit festzulegen, dass der hinterlegte Algorithmus im VBA-Code diese Eingaben abrufen, darauf aufbauend die gewünschte Wand berechnen und die Koordinaten für den Export nach RFEM6 vorbereiten kann. Als Umgebung für diesen Ablauf wird Excel verwendet, da dies sehr verbreitet, einfach zu handhaben und bereits mit einer Programmiersprache (VBA) ausgestattet ist.

3.3.1. Ziel der Eingabeoberfläche

Es soll so wenig wie möglich, aber so viel wie nötig von der Eingabeoberfläche an Informationen abgefragt werden, um die Berechnung der Wand starten zu können. Als Eingabeparameter sind zu konfigurieren:

1. Wandgeometrie:

Die Eingabe sollte mithilfe von kartesischen Koordinaten erfolgen, da auch in RFEM6 mit diesen gearbeitet wird und die Beschreibung der Geometrie so einheitlich wie möglich bleiben sollte. Das Koordinatensystem wird wie folgt festgelegt:

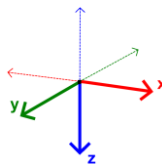


Abbildung 10 Festlegung Koordinatensystem (eigene Darstellung)

Da die Wanddicke durch den Wandaufbau bereits festgelegt ist, werden bei der Eingabe nur x-Koordinaten für die Länge und z-Koordinaten für die Höhe der Wand abgefragt. Um Problematiken mit der Vorzeichenkonvention zu vermeiden, wird es in der Eingabeoberfläche nur möglich sein, positive Werte einzugeben. Dies wird im Algorithmus berücksichtigt und entsprechend dem festgelegten Koordinatensystem angepasst werden. Aus Abbildung 11 kann der Eingabebereich für die Wandgeometrie entnommen werden.

	Von[m]	Bis[m]	Angepasst*
x	0	2	2,03
z	0	2	2,08

| : Alle Werte sind ohne Vorzeichen einzugeben
 * : Geometrie wird auf Raster d. Bretter angepasst

Abbildung 11 Eingabebereich der Wandgeometrie. Hier ist beispielhaft eine 2x2m Wand eingegeben.

2. Öffnungsgeometrie

Die Eingabe von Öffnungen soll nach dem gleichen Prinzip wie bei der Wandgeometrie möglich sein. Eine Öffnung wird durch die x-Koordinaten und die z-Koordinaten definiert. Dabei sollte die Anzahl an Öffnungen frei vom Anwender festgelegt werden können.

Öffnungen - Berücksichtigt nur Nagelbild			
	Von[m]	Bis[m]	
x	0,5	1,5	
z	0,5	1,5	
Weitere Öffnung hinzufügen			Letzte Öffnung löschen

Abbildung 12 Eingabebereich Öffnungsgeometrie. Hier ist beispielhaft eine 1x1m Öffnung eingegeben.

Durch das Betätigen des Buttons „Weitere Öffnung hinzufügen“ aus Abbildung 12 wird ein weiterer Eingabeblock für eine Öffnung rechts der bisherigen Eingabefelder für Öffnungen erzeugt. Mit dem Button „Letzte Öffnung löschen“ lässt sich das zusätzliche Eingabefeld wieder löschen.

3. Festlegung von Startnummern

Da in RFEM6 erzeugte Objekte wie Knoten, Linien und Stäbe eine fortlaufende Nummerierung erhalten, muss es dem Anwender ermöglicht werden, die Startnummer der Nummerierung festzulegen.

Hintergrund:

Wird keine Startnummer festgelegt, so beginnt RFEM6 bei der Nummerierung mit dem Wert 1 bei den importierten Objekten (Knoten und Linien). Falls im geöffneten Modell aber bereits Objekte mit dem identischen Nummerierungswert vorhanden sind, so überschreibt RFEM6 diese. Die bereits vorhandenen Objekte werden damit automatisch gelöscht. Durch das Festlegen auf einen höheren Wert, wie beispielsweise in Abbildung 13 mit einem Wert von 100, kann der Anwender dies umgehen.

Startnummern (max. 999 999)	
Knoten	100
Linien & Stäbe	100

Abbildung 13 Eingabebereich für die Startnummern bei der Knoten- / Liniennummerierung

4. Auswahlmöglichkeit zur vorgesehenen KLED der Wand

Wie in Kapitel 3.2 ersichtlich wurde, spielt die gewählte KLED eine Rolle bei der Festlegung des zu hinterlegenden Schubmoduls im Material der eingesetzten VBM (Schraube oder Nagel). Mithilfe eines Dropdown Menüs sollte der Anwender alle nach dem EC5⁶⁸ möglichen Klassen der Lasteinwirkungsdauer auswählen können und abhängig davon direkt die entsprechenden Werte von $G_{fiktiv,Nagel}$ und $K_{ser,lateral,nagel}$ ausgegeben bekommen. (Abbildung 14)

⁶⁸ DIN EN 1995-1-1:2023, S. 73 Tabelle 5.3.

The image shows a software interface. On the left, there is a dropdown menu labeled 'KLED der Wand' with the value 'mittel' selected. The dropdown list includes options: 'ständig', 'lang', 'mittel', 'kurz', 'k. / s. k.', and 'sehr kurz'. On the right, there is a table titled 'Konfigurationen'.

Konfigurationen	
KLED der Wand	mittel
$K_{ser, LignoLoc}$	677,37 N/mm
fiktives Schubmodul $G'_{LignoLoc}$	890,17 N/mm ²

Abbildung 14 Dropdown Menü für KLED-Festlegung mit direkter Ausgabe der entsprechenden Werte

Da die Werte bei den Würth Schrauben hingegen nicht von der KLED abhängig sind, sondern immer den Wert aus Gleichung (3.15) haben, ist eine Ausgabe dieses Wertes in der Oberfläche nicht von Nöten.

5. Auswahlmöglichkeit für Schrauben im Randbereich

Da es nach dem ETA 24/0946 dem Anwender überlassen bleibt, ob die Wand mit oder ohne Schrauben im vertikalen Randbereich konstruiert werden soll, muss die Eingabeoberfläche dem Anwender eine Auswahlmöglichkeit hierfür bereitstellen. Die Wand muss mit und ohne Schrauben im vertikalen Randbereich berechnet werden können.

6. Auswahlmöglichkeit der zu berechnenden Bereiche

Der hinterlegte Algorithmus muss in der Lage sein, die Koordinaten der Knoten aller Bretter, sowie aller Verbindungsmittel berechnen zu können. Da es unter Umständen in manchen Situationen vorteilhaft sein kann nicht beide Bereiche gleichzeitig, sondern getrennt zu berechnen und nach RFEM6 zu importieren, soll es möglich sein vor dem Start der Berechnung eine Auswahl über die zu berechnenden Bereiche treffen zu können. (Abbildung 15)

The image shows a dialog box titled 'Zu berechnen:'. It contains two checked checkboxes: 'Brettlagen' and 'Nagelbild'. To the right of each checkbox is the word 'WAHR'. At the bottom of the dialog box are two buttons: 'Berechnen' and 'Export nach RFEM'.

Abbildung 15 Auswahlbereich für Berechnungen

7. Hinterlegung ModelName

Um den Export der berechneten Ergebnisse in das gewünschte RFEM6 Modell hinein zu ermöglichen, ist es notwendig den Modellnamen mit anzugeben. Python (mithilfe dessen die Ergebnisse nach RFEM6 importiert werden) muss diesen Namen abrufen können, um Zugriff zum richtigen Modell in RFEM6 zu erhalten.

8. Pfadhinterlegungen

Da die Anwendung Programmübergreifend mit unterschiedlichen Schnittstellen arbeitet, müssen die Pfade hinterlegt sein. Diese Pfade können sich je nach Rechner, auf dem die Anwendung genutzt wird, unterscheiden. Daher sollte in der

Eingabeoberfläche die Möglichkeit gegeben sein, die aktuellen Pfade zu hinterlegen. Dazu gehören:

- 1) Der Pfad zum Python Code
- 2) Der Pfad zur Python.exe

9. Blattschutz

Um zu verhindern, dass der Anwender ungewollt die Eingabeoberfläche in der Excel manipuliert, sollte standardmäßig der Blattschutz aktiviert sein, sodass nur in den dafür vorgesehenen Zellen Änderungen getätigt werden können. Damit die Oberfläche aber dennoch flexibel angepasst werden kann, wird es eine Auswahlmöglichkeit zum Aktivieren und Deaktivieren des Blattschutzes geben, um gezielte Änderungen zuzulassen.

Da es wichtig ist, die Excel Oberfläche übersichtlich zu strukturieren, werden mehrere Arbeitsblätter innerhalb der Excel angelegt. Dazu zählen:

- Input

Dieses Arbeitsblatt wird das zentrale Element für den Anwender sein. Hier befinden sich alle Eingabemöglichkeiten und auch die Buttons zum Starten der Berechnung sowie zum Ausführen des Exports der Ergebnisse.

- VBM

In diesem Arbeitsblatt wird übersichtlich die Berechnung der lateralen Steifigkeit und des darauffolgenden fiktiven Schubmoduls stattfinden. Es wird eine Verknüpfung zum Dropdownmenü aus dem Input-Arbeitsblatt geben, von wo aus die Vorgabe zum KLED entnommen wird.

- Brettlagen

Hier findet sich eine Liste mit den berechneten Koordinaten für die Bretter. Diese Liste wird automatisiert nach der erfolgten Berechnung vom VBA-Algorithmus erstellt.

- Nagelbild

Hier findet sich eine Liste mit den berechneten Koordinaten für die VBM. Diese Liste wird automatisiert nach der erfolgten Berechnung vom VBA-Algorithmus erstellt.

- Stäbe

Hier findet sich eine Liste mit den Stabzuordnungen für alle Elemente. Es werden jeder Linie (welche immer zwischen zwei Knoten entsteht) ein zugehöriger Stab hinterlegt. Erst hierdurch weiß RFEM6 später, ob es sich bei einer Linie um ein Brett, einen Nagel oder eine Schraube mit den entsprechenden Querschnittseigenschaften handelt.

3.3.2. Die Eingabeoberfläche

Aufbauend auf den Anforderungen und Zielen der Eingabeoberfläche für den Anwender wurde die aus Abbildung 16 ersichtliche Oberfläche entwickelt. Diese Eingabeoberfläche ist der Grundstein und auch zentrales Element für den Automatisierungsprozess. Beide Modellansätze (Modellansatz 1 Realmodell und Modellansatz 2 Idealmodell) werden diese Eingabeoberfläche nutzen.

Woodbloc Massivholzwand			
Wandgeometrie			
	Von[m]	Bis[m]	Angepasst*
x	0	2	2,03
z	0	2	2,08
: Alle Werte sind ohne Vorzeichen einzugeben * : Geometrie wird auf Raster d. Bretter angepasst			
		Zu berechnen: ☒ Brettlagen WAHR ☒ Nagelbild WAHR Berechnen Export nach RFEM	
Öffnungen - Berücksichtigt nur Nagelbild			
	Von[m]	Bis[m]	
x			
z			
Weitere Öffnung hinzufügen		Letzte Öffnung löschen	
Startnummern (max. 999 999)			
1	Knoten	100	
2	Linien & Stäbe	100	
Konfigurationen			
3	KLED der Wand	mittel	
4	$K_{ser,LignoLoc}$	677,37 N/mm	
5	fiktives Schubmodul $G'_{LignoLoc}$	890,17 N/mm ²	
6	Schrauben im vertikalen Randbereich	☐ FALSCH	
7	Modellname im R-FEM 6	Modell2	
8	Pfad der Python Export Datei	C:\Users\goetz\OneDrive -	
Allgemeines			
9	Pfad zur Python.exe	C:\Users\goetz\AppData\L	
10	Blattschutz aktivieren	☒ WAHR	

Abbildung 16 Entwickelte Eingabeoberfläche

3.4. Format der Ergebnisausgabe

Die Entwicklung der Algorithmen dient dem Ziel, bestimmte Daten auf Grundlage der in der Eingabeoberfläche getätigten Eingaben automatisiert zu berechnen. Daher wird zu Beginn der Entwicklung des Codes definiert, welche Daten zu berechnen sind. Der entwickelte Code hat dann später das Ziel, die festgelegten Daten korrekt zu berechnen und in einer Excel Datei strukturiert abzuspeichern.

Koordinaten

Die Koordinaten sind das wichtigste Element der berechneten Daten. Über diese kann die Position jedes einzelnen Knoten exakt bestimmt werden. Dabei sind für jeden Knoten die x-, y-, und z-Koordinaten zu berechnen. Wird im Raster zum Beispiel eine Schraube gesetzt, so

muss der Algorithmus 7 Knoten berechnen, denn an jedem Schnittpunkt zwischen der Stabachse der Schraube und der Stabachse eines Bretts einer jeden Lage (insg. sind es 7 Lagen) soll ein Knoten gesetzt werden. Ein jeder dieser 7 Knoten erhält seine spezifischen x-, y- und z-Koordinaten. Dabei sind in diesem Fall die x- und z-Koordinaten aller 7 Knoten identisch (da es sich um ein VBM handelt und sich damit alle Knoten an der gleichen Stelle des Rasters befinden) und lediglich die y-Koordinate ändert ihre Position von Knoten zu Knoten um -0,0228. Dies ist der Abstand der Stabachsen der Brettlagen untereinander.

Knotennummer

In RFEM6 erhält jeder gesetzte Knoten seine eigene Knotennummer zur eindeutigen Identifikation des Knotens. Um Kollisionen zwischen bereits existenten Knoten und den neuen berechneten Knoten zu vermeiden, muss die Knotennummer in der Berechnung berücksichtigt werden und auf einen kollisionsfreien Wert festgelegt werden.

Liniennummer

In RFEM6 erhält jede gesetzte Linie ihre eigene Knotennummer zur eindeutigen Identifikation der Linie. Um Kollisionen zwischen bereits existenten Linien und den neuen Linien zu vermeiden, muss die Liniennummer in der Berechnung berücksichtigt werden und auf einen kollisionsfreien Wert festgelegt werden.

Zuordnungsbezeichnung

Linien werden zwischen zusammengehörenden Knoten gesetzt. Als zusammengehörende Knoten sind alle Knoten eines gleichen Objektes zu betrachten. Um dieses Setzen der Linien automatisiert durchführen zu können, müssen diese Knoten ein gemeinsames Kriterium erhalten, anhand dessen ein Algorithmus die Zuordnung vollziehen kann. Dieses Kriterium erfolgt durch die Zuordnungsbezeichnung. Das entwickelte System der Zuordnungsbezeichnung ist Tabelle 6 zu entnehmen.

Element	Bezeichnung	Variabel a	Variabel b	Beispiel
Schrauben	Schrauben "a" "b"	Position a = "unten": untere SR* a = "oben": obere SR a = "vertikal rechts": rechte SR a = "vertikal links": linke SR	Nummer der Schraube fortlaufend	Schraube unten 1 Erste Schraube der unteren Schraubenreihe
Nägel	L"a"N"b"	Nagellung a = 1: erste Nagelung a = 2: zweite Nagelung	Nummer des Nagels fortlaufend	L1N1 Erster Nagel der ersten Nagelung
Laschenbretter	Lasche "a" "b"	Position a = "unten": untere Lasche a = "oben": obere Lasche	Nummer des Brettes fortlaufend	LascheUnten1 Die untere Lasche
Bretter	L"a"B"b"	Brettlage a = 1: erste Brettlage a = 2: zweite Brettlage ... a = 7: siebte Brettlage	Nummer des Brettes fortlaufend	L1B1 Erstes Brett der ersten Brettlage (90° Lage)

*SR = "Schraubenreihe"

Tabelle 6 System der Zuordnungsbezeichnung

Querschnittsnummer

Es muss berechnet werden, wofür eine Linie repräsentativ ist und dann dieser Linie die Nummer des passenden, bereits vorkonfigurierten Querschnitts, zugeteilt werden.

3.5. Codeentwicklung

Der entwickelte Automatisierungsprozess basiert auf einer modularen Architektur, welche in Abbildung 17 als Flussdiagramm dargestellt ist. Der Prozess gliedert sich in 3 Hauptkomponenten:

- 1) Die Eingabeoberfläche in Excel
- 2) Die Berechnungsmodule in VBA
- 3) Die Exportprozesse über Python und Copy & Paste

Diese Trennung ermöglicht eine klare Aufgabenteilung und erleichtert spätere Erweiterungen. Die Unterscheidung der beiden Modellansätze (Realmodell und Idealmodell) findet sich ausschließlich im Bereich 2) Berechnungsmodule in VBA und wird im Folgenden an den entsprechenden Stellen kenntlich gemacht. Ihre Unterscheidung liegt in den Algorithmen für die Berechnung der Brettlagen sowie der Berechnung des Nagelbildes. Dabei ist jeweils der Algorithmus so konstruiert, dass er die entsprechenden Ziele des Modellansatzes nach Kapitel 3.1 umsetzt. Die vollständigen Codes sind dem Anhängen Anhang IV, Anhang V und Anhang VI zu entnehmen.

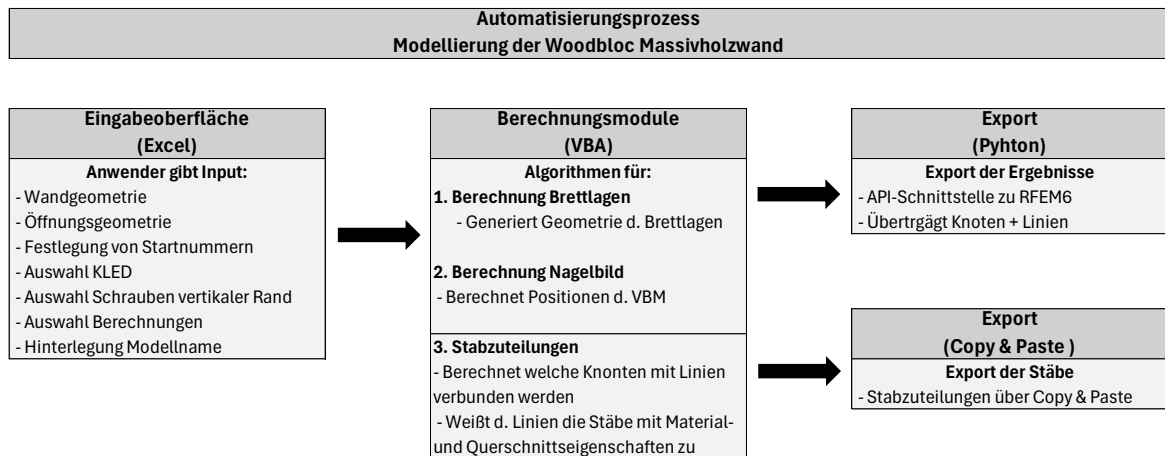


Abbildung 17 Flussdiagramm Automatisierungsprozess

3.5.1. Übersicht Berechnungsmodule

Die VBA-Projektstruktur gliedert sich in fünf Microsoft Excel Objekte sowie in weitere vier Module (Abbildung 18). Bei den Microsoft Excel Objekten handelt es sich um die arbeitsmappenspezifischen Elemente der fünf Arbeitsblätter VBM, Input, Nagelbild, Brettlagen und Stäbe, welche bereits in Kapitel 3.3 erörtert wurden. Innerhalb jeden Bereichs kann ein VBA-Code geschrieben werden, welcher mit dem Arbeitsblatt der Excel interagieren kann. Da das Arbeitsblatt Input als zentrale Eingabeoberfläche dient, wurde in diesem Element auch der vollständige Code zur Steuerung des Prozesses geschrieben. Alle anderen arbeitsmappenspezifischen Elemente (VBM, Nagelbild, Brettlagen und Stäbe) wurden von Excel automatisch erzeugt und sind leer. Daher wird im nächsten Abschnitt der Fokus vollständig auf dem VBA-Inhalt aus der Arbeitsmappe Input gelegt.

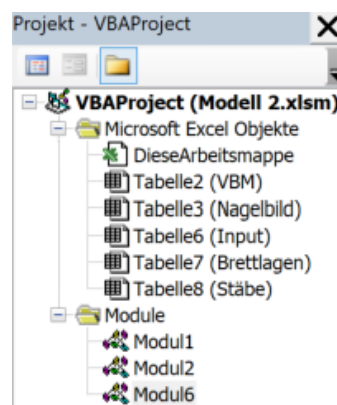


Abbildung 18 VBA-Projektstruktur

Bei den drei separaten Modulen (Modul 1, 2 und 6) handelt es sich um zusätzliche Objekte, die für eine bestimmte Funktion erstellt wurden und allgemein gültige Codes enthalten. Diese erfüllen folgende Aufgaben:

- Modul 1: Auf dieses Modul greift der Button „Export nach RFEM“ zurück. Der hinterlegte Code startet python.exe und öffnet dann den Python Code mit der Schnittstelle zu RFEM.
- Modul 2: Ermöglicht das Hinzufügen und Löschen von weiteren Eingabefeldern für weitere Öffnungen.
- Modul 6: Hier wird der Blattschutz geregelt.

3.5.2. Der Hauptcode und seine Funktionsweise

Im Wesentlichen wird die zentrale Steuerung der Berechnungen über das Modul *Sub Gesamtberechnung ()* gesteuert. Innerhalb dessen werden:

- Grundlagen definiert
- Startnummern aus Eingabefelder übernommen
- Gültigkeit der eingegebenen Werte überprüft
- Geprüft, was zu berechnen ist und entsprechend der Auswahl werden weitere Module aufgerufen:

If berechneNagelbild Then Call Nagelbild_Erstellung(startKnoten, startLinien)

- ➔ Prüft, ob das Nagelbild berechnet werden soll und ruft *Sub Nagelbild_Erstellung ()* auf, falls erfordert. Dabei werden direkt Startnummern mit übermittelt.

If berechneBrettlagen Then Call Wand_Erstellen(startKnoten, startLinien)

- ➔ Prüft ob Brettlagen berechnet werden sollen, und ruft *Sub Wand_Erstellen ()* auf, falls erfordert. Dabei werden direkt Startnummern mit übermittelt.

- Ruft nach den Berechnungen das Modul zum Erstellen der Exportdatei mit den Ergebnissen auf und übergibt Ergebnisse der Brettlagen sowie des Nagelbilds mit:

Call ErstelleExportdatei(berechneBrettlagen, berechneNagelbild)

- Ruft zum Schluss das Modul mit den Berechnungen zur Stabzuteilung auf mit:

Call StabZuteilung()

Diese vier Sub-Module führen die konkreten spezifischen Berechnungen durch und werden teilweise durch weitere Sub-Module unterstützt. Die folgenden Erläuterungen arbeiten mit dem Idealmodell (Modellansatz 2). Das Realmodell (Modellansatz 1) wird aufgrund der starken Ähnlichkeit nicht zusätzlich aufbereitet und kann dem Anhang Anhang IV entnommen werden.

1) *Sub Nagelbild_Erstellung()*

Dieses Sub-Modul arbeitet mit 10 Schritten:

1. Schritt Basics festlegen: Daten werden aus dem Input eingelesen und Variablen werden definiert.

- Es wird geprüft, ob das Nagelbild Schrauben im vertikalen Randbereich berücksichtigen soll.
2. Schritt Wandgeometrie: Die eingelesenen Wandgeometriedaten werden geprüft und auf darauf basierend werden die Maße der Wand berechnet.
3. Schritt Öffnungen: Es wird geprüft ob Öffnungen eingegeben wurden und falls ja werden diese in einem Array (Variable mit mehreren Einspeicherungen) hinterlegt.
4. Schritt Arbeitsblatt: Das Arbeitsblatt „Nagelbild“ wird für die Ausgabe vorbereitet (alte Werte löschen & Kopfzeile schreiben)
5. Schritt Konstanten: Konstanten werden mit den nötigen Rasterwerten festgelegt. Hier findet sich die Hauptunterscheidung zwischen Modellansatz 1 und 2!
6. Schritt Schrauben unten: Die unterste Reihe des Nagelbildes, die Schrauben am Laschenbrett, wird berechnet. Jeder berechnete Knotenpunkt wird in eine Zeile in das zuvor vorbereitete Arbeitsblatt „Nagelbild“ geschrieben.
7. Schritt LignoLoc Nägel: Es werden hier alle Knoten der LignoLoc Nägel berechnet. Anzahl und Position wird durch den Algorithmus bestimmt. Falls Schrauben im vertikalen Randbereich vorgesehen sind, werden die entsprechenden Positionen ausgelassen. Falls Öffnungen eingelesen wurden, so werden die Stellen, an denen eine Öffnung ist, übersprungen (kein VBM dort). Alle berechneten Knoten werden wieder in „Nagelbild“ geschrieben.
8. Schritt Schrauben oben: Ablauf identisch zu Schrauben unten.
9. Schritt Schrauben vertikal: Nun folgt die Berechnung der Schrauben im vertikalen Randbereich, falls erforderlich (ansonsten entfällt der Schritt).
10. Schritt Liniennummern: Es werden die Linien berechnet. Knoten mit identischer Zuordnungsbezeichnung hängen zusammen und bekommen daher eine gemeinsame Liniennummer.
- 2) [Sub Wand_Erstellen\(\)](#)
- Dieses Sub-Modul arbeitet mit 6 Schritten:
1. Schritt Basics festlegen: Daten werden aus dem Input eingelesen und Variablen werden definiert.

2. Schritt Wandgeometrie: Die eingelesenen Wandgeometriedaten werden geprüft und darauf basierend werden die Maße der Wand berechnet.
3. Schritt Arbeitsblatt: Das Arbeitsblatt „Brettlagen“ wird für die Ausgabe vorbereitet (alte Werte löschen & Kopfzeile schreiben)
4. Schritt Berechnungen: Die Koordinaten der Bretter aller Lagen werden berechnet. Der Algorithmus geht lagenweise vor.
5. Schritt Zusatzinformation: Zusätzliche Informationen, die für den weiteren Prozess wichtig sind, werden eingelesen und abgespeichert.
7. Schritt Liniennummern: Es werden die Linien berechnet. Knoten mit identischer Zuordnungsbezeichnung hängen zusammen und bekommen daher eine gemeinsame Liniennummer.

3) [Sub StabZuteilung\(\)](#)

Dieses Sub-Modul arbeitet mit 3 Schritten:

1. Schritt Arbeitsblatt: Das Arbeitsblatt „Stäbe“ wird für die Ausgabe vorbereitet (alte Werte löschen & Kopfzeile schreiben)
2. Schritt Liniennummern: Alle zuvor berechneten Liniennummern werden gesammelt und mit ihrer entsprechenden Zuordnung versehen. Mithilfe dieser Verknüpfung von Liniennummer und Zuordnung können die richtigen Querschnitte zu den Linien zugeordnet werden.
3. Schritt Querschnitte: Es wird geprüft, wofür eine jede Linie repräsentativ ist und abhängig davon wird der richtige Querschnitt zugewiesen. Dadurch können später in RFEM die richtigen Stäbe automatisch erzeugt werden. Die Ergebnisse werden in das Arbeitsblatt „Stäbe“ geschrieben.

4) [Sub ErstelleExportdatei\(\)](#)

Dieses Modul erstellt eine neue .xlsx Datei (Microsoft Excel Arbeitsblatt) mit dem Namen „Export_Gesamt“. In dieser Datei werden die Ergebnisse der Arbeitsblätter „Nagelbild“, „Brettlagen“ und „Stäbe“ eingeschrieben. Diese Datei wird später von Python ausgelesen und deren Daten werden für die Modellerstellung in RFEM genutzt.

Wie aus den hier aufgeführten Erörterungen zu den Hauptmodulen hervorgeht, findet sich lediglich beim [Sub Nagelbild_Erstellung\(\)](#) mit dem 3. Schritt eine Einbindung und Berücksichtigung der Öffnungen. Für ein korrektes analytischen Modell sollte beim Vorhandensein einer Öffnung innerhalb einer Wand neben dem Auslassen der Verbindungsmittel auch ein Auslassen der Bretter an der Stelle der Öffnung im Modell

unternommen werden. Nur so wird bei den mit dem Modell durchgeführten Berechnungen die Unterbrechung der Kraftübertragungen in Bereichen einer Öffnung korrekt berücksichtigt. Im Umfang dieser Ausarbeitung wurde im Berechnungsalgorithmus die Miteinbeziehung von Öffnungen lediglich zur Veranschaulichung im [Sub](#) Nagelbild_Erstellung() durchgeführt, aufgrund der hohen Komplexität und der zeitlichen Einschränkung durch fixe Abgabetermine der Arbeit wurde die Berücksichtigung der Öffnungen nicht im [Sub](#) Wand_Erstellen() implementiert.

Dadurch ist die Funktion zur Eingabe von Öffnungen zwar in der Eingabeoberfläche vorhanden, jedoch dient diese in der aktuellen Version nur zur Veranschaulichung der Möglichkeiten und kann ausdrücklich nicht als fertig einsetzbare Funktion des erarbeiteten Prozesses angesehen werden. Es kann jedoch als teil-automatisierte Funktion betrachtet werden, da ein automatisches Generieren des Nagelbildes in Kombination mit dem manuellen Modellieren der Brettlagen wiederum ein korrektes Abbild einer Wand mit einer integrierten Öffnung ermöglicht.

3.5.3. Ergebnisexport

Im Anschluss an die vollständige Berechnung aller Koordinaten der Knoten wird ein Exportmodul benötigt, dass die generierten Daten aus der Ergebnis Excel automatisiert in das RFEM6-Modell überträgt. Dies erfolgt über ein Python Skript, das direkt mit der RFEM Webservice API kommuniziert. Sobald der Anwender in der Eingabeoberfläche auf den Button „Export nach RFEM“ drückt, wird Python.exe geöffnet und das Skript „ExportRFEM.py“ wird ausgeführt. Der Ablauf dieses Skriptes gliedert sich in zwei Hauptphasen:

1. Einlesen der Daten:

Zunächst wird die Exportdatei „Export_Gesamt.xlsx“ lokalisiert und geöffnet. Startnummern für Linien und Knoten sowie die Modellbezeichnung werden ausgelesen und dienen als Ausgangslage für den anschließenden Importprozess.

2. Generierung der RFEM-Objekte:

Es wird eine Verbindung zum RFEM-Modell hergestellt. Alle Knoten werden mit ihren Koordinaten eingelesen und anhand Ihrer Bezeichnung aus der Spalte „Zuordnung“ gruppiert. Parallel werden Linienobjekte erstellt, welche jeweils die Knotenpunkte einer gemeinsamen Gruppe verbinden. Diese Linien werden mit einem Kommentar mit der entsprechenden Zuordnungsbezeichnung versehen.

Der Vollständige Python-Code ist dem Anhang Anhang VI zu entnehmen.

3.6. Manuelle Eingaben

Der durch das Python Skript durchgeführte Export umfasst die Koordinaten aller Knoten der Verbindungsmittel, sowie der Bretter. Mithilfe der für jeden Knoten individuellen Zuordnung, eine Bezeichnung im Kommentarbereich der Knoten, kann das Python Skript auch die Linien zwischen zusammengehörenden Knoten setzen. Damit werden im Umfang des Python Exports Knoten und Linien dem RFEM-Modell hinzugefügt.

Diese Objekte bilden zwar die Grundlage und gleichzeitig auch den Hauptteil des Modells, jedoch fehlen bis zum berechnungstauglichen analytischen Modell der Woodbloc Massivholzwand noch weitere Punkte wie:

- 1) Stäbe inklusiv ihrer materialspezifischen Querschnitte
- 2) Lager und Bettungen
- 3) Lasten
- 4) Individuelle sonstige Konfigurationen und Anpassungen

Trotz unterschiedlicher Ansätze konnte im Rahmen dieser Arbeit nicht erreicht werden, dass diese Punkte beim automatischen Exportprozess mithilfe des Python Skripts integriert werden. Dabei kamen unter anderem folgende Schwierigkeiten auf:

- 1) Automatisches Erstellen von Stabobjekten über die Webservice Schnittstelle wird von RFEM6 nach 12 Objekten blockiert. In Abbildung 19 ist ein Ausschnitt der Ausgabe des Terminals beim Ausführen des Python Skriptes zu sehen. Hier lassen sich unterschiedliche Dinge herauslesen. Grundlegend wurde das Python Skript so programmiert, dass für jedes erstellte Objekt eine Zeile mit den Daten zum erstellten Objekt im Terminal ausgegeben wird. Dies dient dem Anwender zur besseren Übersicht. Auch Fehler oder Meldungen werden hier wiedergegeben. Aus dem Ausschnitt in Abbildung 19 geht hervor:
 - a. Eine Linie mit der Liniennummer 5136 und der Zuordnungsbezeichnung L7B137 (= Brett Nummer 137 der 7. Brettlage, vgl. Kapitel 3.4) wurde erfolgreich im Modell erstellt.
 - b. Darauf folgen 12 Stäbe (Member) die erfolgreich erstellt wurden.
 - c. Eine Meldung: Demoeinschränkung – nur 12 Stäbe erlaubt. Weitere Stäbe werden übersprungen.

```

Linie 5136 erstellt für Kommentar 'L7B137' und verbindet Knoten [5272, 5273]
Member 5000 erstellt: Querschnitt Nr. 5, Kommentar 'LascheUnten1', basiert auf Linie 5000
Member 5001 erstellt: Querschnitt Nr. 5, Kommentar 'LascheOben2', basiert auf Linie 5001
Member 5002 erstellt: Querschnitt Nr. 4, Kommentar 'L1B3', basiert auf Linie 5002
Member 5003 erstellt: Querschnitt Nr. 4, Kommentar 'L1B4', basiert auf Linie 5003
Member 5004 erstellt: Querschnitt Nr. 4, Kommentar 'L1B5', basiert auf Linie 5004
Member 5005 erstellt: Querschnitt Nr. 4, Kommentar 'L1B6', basiert auf Linie 5005
Member 5006 erstellt: Querschnitt Nr. 4, Kommentar 'L1B7', basiert auf Linie 5006
Member 5007 erstellt: Querschnitt Nr. 4, Kommentar 'L1B8', basiert auf Linie 5007
Member 5008 erstellt: Querschnitt Nr. 4, Kommentar 'L1B9', basiert auf Linie 5008
Member 5009 erstellt: Querschnitt Nr. 4, Kommentar 'L1B10', basiert auf Linie 5009
Member 5010 erstellt: Querschnitt Nr. 4, Kommentar 'L1B11', basiert auf Linie 5010
Member 5011 erstellt: Querschnitt Nr. 4, Kommentar 'L1B12', basiert auf Linie 5011
Warnung: Demoeinschränkung erreicht - nur 12 Stäbe erlaubt. Weitere Stäbe werden übersprungen.
Fehler beim Abschreiben der Modifikation: b Server raised fault: Objekt: Stab Nr. 5000 (Attribut: Zugewiesen an Linie Nr.

```

Abbildung 19 Terminalausgabe: Ausschnitt mit Fehler beim Stabexport

- 2) Lasten und Lager / Bettungen können sehr individuell gestaltet sein und können besser innerhalb der RFEM6 Umgebung erstellt und konfiguriert werden

Nach Rücksprache mit dem Dlubal Service für RFEM stellte sich heraus, dass die Einschränkung auf 12 Stäbe allgemein im „alten“ API-Webservice existiert. Bei Umstellung auf den „neuen“ API II / gRPC Service von Dlubal wäre die Einschränkung auf 1000 Abfragen angehoben – für mehr Abfragen muss dann ein API-Abo⁶⁹ abgeschlossen werden. Da dieses Abo Modell für den Umfang dieser Arbeit nicht sinnvoll ist (hohe Kosten und je nach Abo Modell weiterhin eine Beschränkung bezüglich der Menge der Abfragen), wurde der gRPC-Service von Dlubal nicht in dieser Arbeit weiter thematisiert. Stattdessen wurde eine Lösung für die Arbeit mit dem bereits vorgestellten API-Webservice entwickelt.

An dieser Stelle kommen zusätzlich zum Exportprozess durch das Python Skript manuelle Eingriffe hinzu. Dabei sieht der Lösungsansatz das Verwenden einer vorkonfigurierten Modellvorlage für das RFEM-Modell vor. Es wurde ein leeres RFEM-Modell mit dem Namen „MASTER TEMPATE_Nur kopieren“ erstellt, in welchem die grundlegenden Konfigurationen bereits eingerichtet sind. Der Ablauf sieht vor, dass der Anwender bei der Erstellung eines neuen Modells dieses vorgefertigte Template als Vorlage verwendet. Damit hat er als Ausgangslage bereits folgende Punkte im Modell:

- 1) Stäbe mit den materialspezifischen Querschnitten der
 - a. Würth Assy 4 6x150mm Schrauben
 - b. LignoLoc 4,7x90mm Nägel
 - c. Bretter C16 184,8/22,8mm
 - d. Bretter C16 160/22,8mm
 - e. Laschenbretter C24 120/60mm
- 2) Lager und Bettungen mit vorgefertigten Standardwerten

Nun kann der Anwender die Stäbe durch ein einmaliges Copy & Paste aus der berechneten Excel-Liste kopieren und nach dem erfolgten Export der Ergebnisse durch das Python Skript in den Tabellenreiter „Stäbe“ innerhalb des RFEM-Modells wieder einfügen. Dieser Ablauf

⁶⁹ Dlubal API Abos (2025).

gewährleistet, dass durch ein einmaliges manuelles Eingreifen alle Stäbe des gesamten Modells korrekt erstellt werden.

Anmerkung: Die Materialkonfiguration der LignoLoc Nägel berücksichtigt das fiktive Schubmodul der lateralen Steifigkeit für KLED mittel. Sollte ein anderes gewählt werden, so kann der aus der Excel Eingabeoberfläche berechnete Wert in die Materialeigenschaften des LignoLoc Nagels innerhalb des RFEM-Modells übertragen werden. Dies ist manuell durchzuführen.

4. Anwendung des erarbeiteten Prozesses

Bevor mit der Anwendung des Automatisierungsprozesses begonnen werden kann, ist bei erstmaliger Nutzung eine Ersteinrichtung durchzuführen. Dies ist erforderlich, um alle notwendigen Programme richtig zu installieren und erforderliche Pfade richtig zu verknüpfen. Eine ausführliche Anweisung zur Ersteinrichtung ist dem Anhang I zu entnehmen. Danach kann mit der Anwendung begonnen werden.

4.1. Anwendung allgemein

Anwendung

Die Anwendung wird über die Excel (Realmodell oder Idealmodell) gesteuert. Im Arbeitsblatt „Input“ müssen zu Beginn die erforderlichen Eingaben zur Konfiguration der Wand getätigt werden. Nach Eingabe der gewünschten Konfiguration kann über den Button „Berechnen“ die Berechnung gestartet werden. Nachdem die Berechnungen vollständig abgeschlossen sind, werden die Ergebnisse zusätzlich in einer separaten .xlsx Datei mit dem Namen „Export_Gesamt“ am gleichen Ort, an dem die Excel Datei des Modells abgelegt wurde, abgespeichert. Dieser Prozess kann, je nach Größe der Wand und Rechenleistung des PCs, einige Sekunden bis hin zu Minuten in Anspruch nehmen.

Nach erfolgter Berechnung kann zum Export nach RFEM6 übergegangen werden. Anzumerken ist hierbei, dass das betreffende Modell in RFEM6 geöffnet und im Vordergrund aktiv sein muss, da der Exportvorgang nur unter diesen Bedingungen erfolgreich durchgeführt werden kann. Zudem darf sich kein offener Dialog in RFEM6 befinden, da dies die API-Kommunikation unterbricht.

Mit dem Betätigen des Buttons „Export nach RFEM“ erfolgt der Export. Dieser überträgt die berechneten Knoten, inklusive deren Informationen zur Nummerierung und Zuordnung, direkt aus der Excel Datei „Export_Gesamt“ mithilfe von Python an RFEM. Hierbei werden automatisch alle Knoten, sowie auch die Linien der Verbindungsmittel und Bretter generiert.

Manuelle Ergänzungen

Nach dem erfolgreichen Export befinden sich sämtliche Knoten, sowie die Linien, welche Knoten mit der gleichen Zuordnungsbezeichnung verbinden, in RFEM6. Jedoch ist das Modell zu diesem Zeitpunkt noch unvollständig und es bedarf weiterer Schritte bevor damit gearbeitet werden kann:

1) Einfügen der Stäbe

Im Excel Arbeitsblatt „Stäbe“ sind die Spalten A bis G vollständig zu kopieren (Strg. + C) und in RFEM6 im Tab „Stäbe“ einzufügen (Strg. + V). Mögliche Fehlermeldungen können ignoriert werden, da diese auftreten, sofern die Kopfzeile mitkopiert wurde.

2) Modellkontrolle

Eine visuelle und strukturelle Überprüfung des Modells auf überflüssige oder fehlerhafte Knoten und Stäbe ist erforderlich. Besonders in den Ecken können kurze, nicht angebundene (heißt kein Kontakt zu einem VBM) Stäbe auftreten. Überflüssige Elemente sind manuell aus dem Modell zu löschen.

3) Definition von Lagern und Lasten

Die Auflagerbedingungen sowie die Einwirkungen sind je nach Anwendungssituation individuell einzufügen.

4) Verbinden aller Knoten und Stäbe

Um die strukturelle Kontinuität und Kraftübertragung im analytischen Modell sicherzustellen, sind alle sich kreuzenden Stäbe über die Funktion *Extras – Modellkontrolle – Kreuzende oder nicht verbundene Stäbe* miteinander zu verbinden. Die Verbindung sollte dabei je nach Modellvariante erfolgen:

- Modellansatz 1 Realmodell: Die Verbindung erfolgt über *Standardknoten*
- Modellansatz 2 Idealmodell: Die Verbindung erfolgt über *Knoten auf Stab*

Durchführung der Berechnung

Nach Abschluss der Modellierung und der Modellkontrolle kann die statische Berechnung gestartet werden. RFEM6 ermittelt auf Grundlage des erstellten Modells die Schnittgrößen, Verformungen und Lagerreaktionen der individuellen Woodbloc Massivholzwand. Durch den bis hierher durchgeführten Automatisierungsprozess wird eine möglichst fehlerfreie Modellstruktur sichergestellt, was wiederum die Aussagekraft der Analyseergebnisse erhöht.

4.2. Eingabebeispiel mit Realmodell

Nun soll das Szenario eines konkreten Beispiels durchgespielt und veranschaulicht werden. Ziel wird es sein, eine 4m x 2,8m Wand berechnen zu lassen. Dabei wird die in Kapitel 3.3 entwickelte Eingabeoberfläche verwendet. Die Konfigurationseingaben werden wie aus Abbildung 27 im Anhang II ersichtlich. Die Berechnung erfolgt im Realmodell.

Berechnung

Nach einstellen der Konfiguration wird die Berechnung gestartet und nach einer Berechnungsdauer von 10,83 Sekunden (gemessen) sind alle Koordinaten erfolgreich berechnet. In Abbildung 28 bis Abbildung 30 aus Anhang II sind die berechneten Ergebnisse ersichtlich.

Automatischer Export

Im nächsten Schritt wird nun der Export der Daten gestartet. Wichtig ist hierbei, dass das vorgesehene RFEM6 Modell basierend aus der MASTER_TEMPLATE Vorlage bereits geöffnet und mit dem Modellnamen „Eingabebeispiel“ versehen ist. Im Anhang II ist das noch leere Modell „Eingabebeispiel“ ersichtlich. Durch das Verwenden des vorbereiteten Templates sind alle nötigen Grundeinstellungen bereits vorbereitet. So sind im Navigator (links im Bild) die 5 Querschnitte der Stäbe bereits ersichtlich (Querschnittsnummer 1 bis 5). Die in Abbildung 30 zu sehende Liste teilt dem Stab der Liniennummer 100 die Querschnittsnummer 1 zu, demnach wird es sich hierbei um eine Würth Assy 4 Schraube handeln. Der Stab mit der Linie Nr. 122 hingegen erhält die Querschnittsnummer 2 und ist somit repräsentativ für einen LignoLoc Nagel.

Es folgt nun der automatische Export der berechneten Daten über den Button „Export nach RFEM“. Nach gemessenen 46,11 Sekunden erscheint auf der Excel Eingabeoberfläche die Meldung „Python-Skript erfolgreich ausgeführt!“ und bestätigt somit den Abschluss des automatischen Exportvorgangs. Aus der Abbildung 32 ist bereits die Struktur der berechneten Wand ersichtlich. Bei genauer Betrachtung lassen sich die einzelnen Strukturen erkennen, wie bspw. die zueinander versetzten Nagel Reihen oder auch die $\pm 60^\circ$ Brettlagen.

Manueller Export

Nachdem nun Knoten und Linien erfolgreich dem Modell hinzugefügt wurden, folgt nun der manuelle Export der Stäbe. Dazu werden alle Spalten der Stäbe aus Abbildung 30 per Strg. + C kopiert und dem Modell im Tab Stäbe wieder per Strg. + V eingefügt. Aus Abbildung 33 ist, im unteren Bereich das Tab der Stäbe im Modell mit den nun bereits eingefügten Stabdaten zu sehen (vgl. zu Abbildung 32 unten, hier ist der Tab noch leer).

Modellkontrolle

Bei der visuellen Modellkontrolle werden besonders die Eckbereiche überprüft. Dabei ist ersichtlich, dass die äußerste Brettdiagonale (Abbildung 20, gelb markiert) von mindestens einem VBM geschnitten wird (blaue Kreise). In diesem Fall geht das berechnete Raster korrekt auf und es ist keine manuelle Korrektur erforderlich. Weiter fällt ein alleinstehender Knoten im rechten Eck auf. Dieser trägt die Bezeichnung „L2B76“ (Abbildung 21) und ist damit für das 76te Brett der 2 Brettlage berechnet worden. Der alleinstehende Knoten ist für das analytische Modell von keinem Nutzen und sollte manuell herausgelöscht werden.

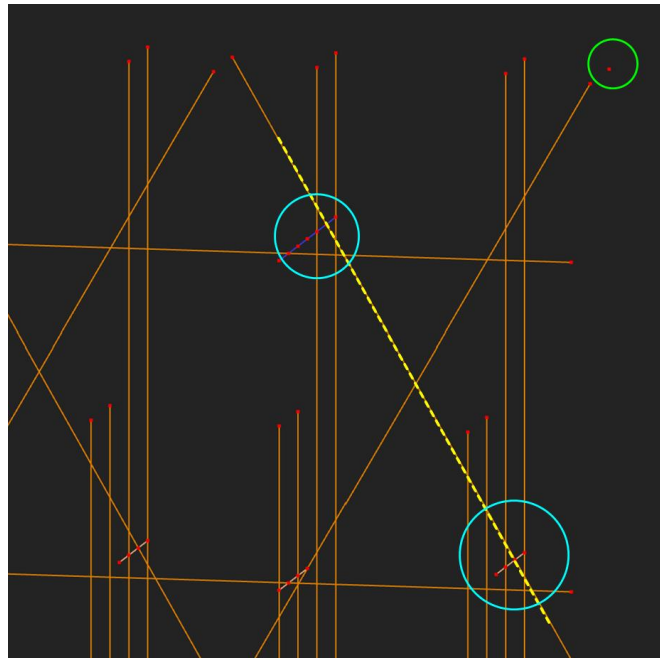


Abbildung 20 Visuelle Kontrolle nach Datenexport des Eingabebeispiels

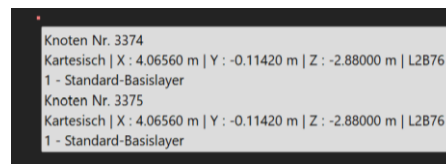


Abbildung 21 Visuelle Kontrolle: alleinstehender Knoten im Eingabebeispiel

Lagerdefinition

Im nächsten Schritt können nun die Lager festgelegt werden. Hierzu finden sich im Navigator (Abbildung 34) bereits die Einstellungen für

Knotenlager (Abbildung 35)

- 1) Lager Nr. 1: Voreingestellt für die unteren Knoten bei $z = 0$
- 2) Lager Nr. 2: Voreingestellt für die oberen Knoten $z = 2.88$
- 3) Lager Nr. 3: Voreingestellt für die Knoten der 90° Bretter, welche auf die Laschenbretter stoßen

Linienlager (Abbildung 36)

- 1) Lager Nr. 1: Voreingestellt für die Bettung des unteren Laschenbrettes
- 2) Lager Nr. 2: Voreingestellt für die Bettung des oberen Laschenbrettes

Aus Abbildung 37 im Anhang II ist das fertig mit Lagern versehene Modell ersichtlich. Die individuelle Konfiguration der Lager ist vom Anwender selbst anzupassen und wird aufgrund des zusätzlichen Umfangs nicht weiter in dieser Arbeit behandelt.

Verbinden sich kreuzender Stäbe

Als letzter auszuführender Schritt gilt es nun alle Schnittpunkte von Knoten und Stäben zu verbinden, sodass im Gesamtsystem die Kräfte korrekt übertragen werden (Abbildung 38).

An der Stelle ist die Modellierung des Modells fertig abgeschlossen. Bis zu diesem Punkt sind alle erledigten Aufgaben Teil des erstellen Prozesses dieser Arbeit. Die nächsten Schritte mit dem Modell hängen von den konkreten Zielen des Anwenders ab. I.d.R. werden eine Schnittgrößen- und Verformungsbetrachtung durchgeführt. Hierfür wären die spezifischen Lasten anzusetzen und die Berechnung mithilfe des RFEM Solver zu starten.

Idealmodell

Dieses Beispiel wurde anhand des Realmodells durchgeführt. Bei der Verwendung des Idealmodells sind alle Schritte identisch auszuführen, einzig das Verbinden sich kreuzender Stäbe erfolgt nun über „Knoten auf Stab“ anstelle von „Standardknoten“.

4.3. Verformungsvergleich Realmodell vs. Idealmodell

Ein direkter Vergleich der beiden Modellansätze, Realmodell und Idealmodell, liefert unterschiedliche Verformungsergebnisse. Zur Veranschaulichung wird im Folgenden eine 2x2m Wand mit beiden Modellansätzen generiert und anschließend mit identischen Lagerbedingungen und einer vertikalen Last von 50kN in z-Richtung konfiguriert. Die Wand soll ohne Schrauben im vertikalen Randbereich und ohne Öffnungen generiert werden.

Ein direkter Vergleich liefert die Ergebnisse aus Tabelle 7. Die dazugehörigen Verformungsbilder können dem Anhang III entnommen werden.

	Realmodell	Idealmodell
max u 	29,2 mm	14 mm
min u 	0 mm	0 mm
Verformungsrichtungen	x- & z-Richtung	x- & z-Richtung
Verformungsverlauf	Zunehmend von oben nach unten	Zunehmend von oben nach unten
Strukturverhalten	Diagonale "drücken seitlich heraus"	Diagonale "drücken seitlich heraus"

Tabelle 7 Vergleich des Verformungsbildes Realmodell VS Idealmodell

Auffällig ist hierbei, dass das Verformungsverhalten grundlegend einmal ähnlich ist, jedoch die maximal auftretende Verformung u beim Realmodell um etwa das Doppelte größer ist als beim Idealmodell. Dieses Ergebnis bestätigt die Annahmen der Modellansätze, dass das Realmodell eine erhöhte Nachgiebigkeit aufgrund der realistischen Nachmodellierung aufweist. Besonders die zwei getrennten Nagellungen sowie die nicht einheitlichen

Kreuzungspunkte spielen hierbei eine Rolle. Weiter wird ersichtlich, dass das Idealmodell durch analytisch sinnvolle Anpassungen diese erhöhte Nachgiebigkeit ausgleicht, ohne dabei das grundlegende Verformungsverhalten gegenüber dem Realmodell zu ändern.

4.4. Zeitaufwand

Um die Effizienz des entwickelten Prozesses beurteilen zu können, wurden die Zeiten für die Modellierung, wie in Kapitel 4.2 vorgenommen, gemessen. Dabei wurden 6 Durchgänge, drei mit dem Realmodell und weitere drei mit dem Idealmodell, durchgeführt.

Durchgang	Zeitaufwand in [s]				Summe
	Dateneingabe	Berechnung	Automatischer Export	Manuelle Eingaben	
1	19	10,83	46,11	163,39	239,33
2	18,4	12,53	44,55	159,5	234,98
3	19,1	10,49	41,44	150,44	221,47
4	16	9,43	32,3	176,43	234,16
5	18,5	8,97	31,58	178,55	237,6
6	18,1	9,15	31,59	162,44	221,28

Tabelle 8 Zeitaufwand bei Anwendung des entwickelten Prozesses

Unter Berücksichtigung von Messfehlern kann bei beiden Modellen ein etwa gleicher Zeitaufwand festgestellt werden. Dabei beträgt die durchschnittliche Zeit, um das Modell wie in Kapitel 4.2 beschrieben zu modellieren, etwa 234,57 Sekunden, bzw. ca. 4 min.

4.5. Vergleich zu manuellem Modellieren

Im Vergleich mit dem Modellieren des analytischen Modells ohne Nutzung des hier erarbeiteten Automatisierungsprozesses können zum einen die Fehleranfälligkeit des Ergebnisses und zum anderen der Aufwand bis zum Erreichen des fertigen analytischen Modells untersucht und verglichen werden.

4.5.1. Fehleranfälligkeit

Das vollständig manuelle Modellieren der Woodbloc Massivholzwand erfordert viele einzelne Schritte. Dabei sind alle Raster und Abstände manuell einzugeben, Knoten müssen manuell gesetzt werden und Stäbe werden immer neu konfiguriert und dann einzeln gesetzt. Daraus ergibt sich, dass alle Eingaben immer geprüft werden müssen:

1. Eingegebene Koordinaten aller Knoten müssen geprüft werden
2. In Zusammenhang mit den Knoten-Koordinaten auch Raster und Abstände
3. Konfiguration der Knoten
4. Konfiguration der Materialien und Querschnitte
5. Konfiguration der Lager und Bettungen

Besonders durch den großen Umfang an Knoten (viele VBM und viele Bretter) bei gleichzeitig geringen Abständen der Elemente untereinander, sowie der vielen unterschiedlichen Ausrichtungen (Bretter mit Ausrichtungen in 90°, 0°, -60° und +60°) wird das Ganze nochmals deutlich komplexer.

Im Gegensatz hierzu ist der Umfang an beeinflussbaren Variablen bei der Verwendung des hier entwickelten Automatisierungsprozesses deutlich geringer, da die Punkte 1 bis 3 nur von der Eingabe der Wandgeometrie abhängig sind. Daher ist in diesem Fall nur zu überprüfen, ob die in der Eingabeoberfläche angegebenen Abmessungen korrekt sind. Durch die im Algorithmus fest hinterlegten Abstände und Raster wird eine konsistente und korrekte Berechnung der Knoten gewährleistet. Um Änderungen an den Abständen oder dem Raster vorzunehmen, muss direkt auf den Algorithmus zugegriffen werden. Die Punkte 4 und 5 sind bei dem hier entwickelten Automatisierungsprozess zwar zu Teilen auch vorbereitet, jedoch bedarf es hier einer direkten Kontrolle der Werte.

Aufgrund dessen, dass der fehleranfällige Teil der Modellierung, das korrekte Setzen aller Knoten, durch den Modellierungsprozess entfällt, kann von einer erheblichen Minimierung des Fehlerpotentials ausgegangen werden.

4.5.2. Zeitaufwand

Der Umfang des zeitlichen Aufwands beim manuellen Erstellen des analytischen Modells ist stark anhängig vom Erfahrungsstand und der Routine der modellierenden Person. Im Umfang dieser Arbeit wurde das in Kapitel 4.2 erstellte Beispielmmodell zusätzlich noch manuell modelliert und die dafür benötigte Zeit gemessen. Dabei war zum Erreichen eines identischen Modells eine Zeit von knapp 2 Stunden notwendig. Da bei dieser Zeitmessung der Modellierer nur einen mittleren Erfahrungsstand sowie auch keine Routine speziell für das Modellieren der Woodbloc Massivholzwand aufweisen kann, darf angenommen werden, dass Erfahrene Anwender etwa 20% bis 30% schneller sind.

4.5.3. Direkter Vergleich

Zusammenfassend führt ein direkter Vergleich, zwischen der Verwendung des hier entwickelten Automatisierungsprozesses und dem manuellen Erstellen der Wand, zu dem Ergebnis aus Tabelle 9.

	Fehleranfälligkeit	Ca. Zeitaufwand
Manuell	Hoch	1h 30 min
Automatisiert	Gering	4 min

Tabelle 9 Vergleich manuellem zu automatisiertem Modellieren

Damit kann durch das Verwenden des automatisierten Prozesses der Zeitaufwand um 95,6% reduziert werden.

5. Ergebnisdiskussion und Zielerreichung

5.1. Bewertung der analytischen Korrektheit der Modelle

Es wurden im Umfang dieser Arbeit zwei verschiedene Modellansätze ausgearbeitet und für jeden dieser Ansätze konnten die Modellierungsprozesse weitestgehend automatisiert werden. Nun ist jedoch die Korrektheit eines analytischen Modells dafür ausschlaggebend, ob ein Modell tatsächlich auch verwendet werden kann. Die sich unter Lasteinwirkung ergebenden Schnittgrößen und Verformungen sollten so nah wie möglich an den tatsächlichen Werten sein, die in der Realität unter gleichen Umständen auftreten würden. Nur wenn dies gewährleistet und verifiziert ist, kann eins dieser hier entwickelten analytischen Modelle auch für Berechnungen und Nachweisführungen verwendet werden.

In Kapitel 4.3 wurden die Verformungsergebnisse beider Modellansätze gegenübergestellt. Dabei wurden die Auswirkungen der größeren Nachgiebigkeit des Realmodells gegenüber dem Idealmodell ersichtlich (um etwa dem Faktor 2). Diese Differenz spielt eine bedeutende Rolle, denn sollten die Ergebnisse des Realmodells näher an der Realität sein, so würde man bei der Verwendung des Idealmodells eine zu geringe Nachgiebigkeit der Wand ansetzen. Umgekehrt würde man bei der Verwendung des Realmodells zu große Nachgiebigkeiten ansetzen, wenn die Ergebnisse des Idealmodells genauer wären. Diese Auswirkungen haben erheblichen Einfluss auf die Tragfähigkeit, aber auch die Wirtschaftlichkeit eines berechneten Elementes und daher ist es zwingend notwendig, die Ergebnisse zu validieren.

Die Validierung der hier entwickelten Modelle muss von einer mit dem analytischen Modell der Woodbloc Massivholzwand vertrauten Person durchgeführt werden. Dabei sollte eine Verifizierung der Ergebnisse mithilfe eines Abgleichs mit bereits bestätigten Referenzmodellen durchgeführt werden. Da keine verifizierten Referenzmodelle vorliegen, wird die Wertung bezüglich der analytischen Korrektheit der Modelle in dieser vorliegenden Arbeit nicht untersucht werden können.

5.2. Bewertung der Zielerreichung

Ziel dieser Arbeit war es, einen Automatisierungsprozess zur Modellierung am Beispiel der Woodbloc Massivholzwand zu entwickeln, der sowohl den Modellierungsaufwand als auch das Fehlerpotential im Vergleich zur manuellen Modellierung signifikant reduziert. Konkret wurde angestrebt, durch ein variabel konfigurierbares Template in Kombination mit automatisierten Berechnungs- und Datenübertragungsprozessen ein Zeitersparnis von mindestens 30% gegenüber der herkömmlichen, manuellen Modellierung zu erzielen (Kapitel 1.2).

Um die Zielerreichung zu bewerten, wurde ein direkter Vergleich zwischen der manuell erstellten Modellierung und der automatisiert erzeugten Modellierung auf Grundlage des entwickelten Prozesses durchgeführt (Kapitel 4.5). Dabei ergaben sich folgende zentrale Beobachtungen:

- Die durchschnittliche Zeitersparnis beim Erstellen eines vollständigen analytischen Modells der Woodbloc Massivholzwand lag bei über 90% (95,6%), womit das formulierte Ziel einer Reduktion von mindestens 30% deutlich übertroffen wurde.
- Der Modellierungsprozess konnte dank der VBA-basierten Eingabeoberfläche sowie der Anbindung über die RFEM6 Webservice API deutlich standardisiert werden. Dadurch wurde der Prozess nicht nur schneller, sondern auch weniger fehleranfällig.
- Insbesondere die automatisierte Generierung des Nagelbildes stellte im Vergleich zur manuellen Variante einen erheblichen Fortschritt dar. Die korrekte Umsetzung des ETA-konformen Nagelrasters mit der Modellvariante 1 Realmodell konnte erfolgreich umgesetzt werden. Unter Berücksichtigung eines analytisch sinnvolleren Modells konnte auch ein Idealmodell entwickelt werden, welches den Anforderungen aus dem ETA 24/0946 dem Sinne nach entspricht.

Die ursprünglich formulierte Zielsetzung, den Modellierungsaufwand signifikant zu senken und gleichzeitig eine reproduzierbare und fehlerreduzierte Modellgenerierung zu ermöglichen, wurde demnach erreicht. Der entwickelte Automatisierungsprozess kann als robuste, skalierbare Lösung betrachtet werden. Jedoch wurden nicht alle gewünschten Konfigurationsmöglichkeiten vollständig dem Automatisierungsprozess integriert. So werden in der aktuellen Version Öffnungen nur unvollständig berücksichtigt und die Eingabe von Lagern erfolgt manuell innerhalb der Modellumgebung in RFEM6.

Die Ergebnisse verdeutlichen zudem, dass der gewählte methodische Aufbau mit schrittweiser Entwicklung (vgl. Methodisches Vorgehen Stufenmodell Kapitel 1.3) zielführend war und eine hohe Prozessstabilität bei gleichzeitig wachsendem Funktionsumfang ermöglichte.

6. Zusammenfassung und Ausblick

Der im Rahmen dieser Arbeit entwickelte Automatisierungsprozess konnte das zentrale Ziel, den Modellierungsaufwand und das Fehlerpotential signifikant zu reduzieren, nachweislich erreichen. Die erzielte Zeitersparnis von über 90% übertrifft die angestrebte Marke von 30% bei Weitem. Dennoch stößt der Prozess in seiner aktuellen Ausbaustufe an technologische und konzeptionelle Grenzen, Herausforderungen und den daraus resultierenden Potenzialen für zukünftige Weiterentwicklungen.

6.1. Grenzen der aktuellen Umsetzung

Die derzeitige Version des Automatisierungsprozesses unterliegt hauptsächlich drei Einschränkungen:

1) Geometrische Einschränkungen:

Der Algorithmus ist aktuell ausschließlich für die Erstellung von rechteckigen Wandgeometrien ausgelegt. Die Eingabe erfolgt über die Definition von Anfangs- und Endpunkten mithilfe von x- und z-Koordinaten. Die Modellierung von komplexeren, polygonalen oder freien Geometrien wie sie beispielsweise bei Giebelwänden vorkommt, wird nicht unterstützt.

2) Unvollständige Implementierung von Öffnungen:

Obwohl die Eingabeoberfläche die Definition von Öffnungen ermöglicht, werden diese im Berechnungsalgorithmus nur teilweise berücksichtigt. Der Code für das Nagelbild lässt die Verbindungsmittel im Bereich der Öffnung korrekt aus. Jedoch wurde die weitaus komplexere Anpassung der Brettlagen, also das rechnerische Auftrennen der Stäbe an den Öffnungsändern, im Rahmen dieser Arbeit nicht mehr umgesetzt. Ein mit Öffnungen generiertes Modell ist daher in der aktuellen Version analytisch nicht korrekt modelliert, ohne dass manuelle Nacharbeiten an den Brettlagen durchgeführt werden.

3) Teilautomatisierter Datenexport:

Der Export der Modelldaten nach RFEM6 ist nicht vollständig automatisiert. Während Knoten und Linien über das Python Skript und die Webservice API Schnittstelle erfolgreich erstellt werden, muss die Zuweisung der Stäbe manuell über Copy & Paste aus der automatisch erzeugten Excel Liste erfolgen. Der Grund hierfür ist eine Beschränkung der genutzten API-Version, die das automatische Erstellen von mehr als 12 Stabelementen verhindert.

6.2. Herausforderungen in der Entwicklung

Die genannten Grenzen resultieren aus spezifischen Herausforderungen, die während der Konzeption und Umsetzung des Prozesses bewältigt werden mussten:

1) Algorithmische Komplexität:

Eine zentrale Herausforderung lag darin, die teils komplexen und nicht immer eindeutigen Strukturen aus der ETA 24/0946 in einen gut strukturierten und fehlerfreien Algorithmus zu übersetzen. Insbesondere die korrekte Interpretation des Nagelbildes und der Brettlagen erforderte eine detaillierte Analyse und führte zur Entwicklung von zwei unterschiedlichen Modellansätzen (Real – und Idealmodell), um sowohl eine möglichst realitätsnahe als auch eine analytisch sinnvolle Abbildung zu ermöglichen.

2) Funktionalität der Schnittstellen:

Die größte technische Hürde stellte die limitierte Funktionalität der genutzten RFEM6 Webservice API Schnittstelle dar. Die Beschränkung auf 12 Stäbe verhinderte eine vollumfängliche Automatisierung des Datenexports. Zwar bietet Dlubal mit dem neueren gRPC-API Service eine leistungsfähigere Alternative, diese ist jedoch mit kostenpflichtigen Abonnements verbunden, deren Nutzung den Rahmen dieser Bachelorarbeit überschritten hätte.

3) Verifizierung der Modelle:

Wie in Kapitel 5.1 dargelegt, konnte eine abschließende Bewertung der analytischen Korrektheit der erzeugten Modelle nicht durchgeführt werden. Es fehlten verifizierte Referenzmodelle oder Versuchsergebnisse, um die berechneten Schnittgrößen und Verformungen validieren zu können.

6.3. Potentiale für Weiterentwicklungen

Trotz der genannten Grenzen und Herausforderungen bietet der entwickelte Prozess eine robuste Grundlage und zeigt Potentiale für zukünftige Erweiterungen. Die aktuellen Herausforderungen können insbesondere mit Blick auf die rasanten technologischen Fortschritte der aktuellen Zeit als lösbar betrachtet werden. Weiter ergeben sich interessante Potentiale für Weiterentwicklungen:

1) Überwindung der aktuellen Grenzen:

Zukünftige Arbeiten könnten sich darauf konzentrieren, die Algorithmen zur Geometrieberechnung zu erweitern, um auch Polygone Wandformen zu unterstützen. Ebenso sollte die Implementierung von Öffnungen vervollständigt werden, sodass auch die Brettlagen korrekt ausgespart werden. Eine Umstellung auf die neuere gRPC-Schnittstelle von Dlubal würde zudem eine vollständige Automatisierung des Exports

ermöglichen und den Prozess mit den aktuellen Entwicklungen bei Dlubal in Einklang bringen.

2) Erweiterung des Funktionsumfangs:

Der Prozess ließe sich um weitere Funktionen ergänzen. Denkbar wäre zum Beispiel eine automatisierte Generierung von Lagern in Verbindung mit einem Auswahlbereich von spezifischen Verbindungsmitteln und Systemen. Hierfür könnte überprüft werden, ob beispielsweise über API-Schnittstellen aktuelle Herstellerdaten von ausgewählten Produkten abgerufen und direkt in eine passende Lagerkonfiguration umgewandelt werden können. Die aktuelle Eingabeoberfläche könnte so um einen „Lagerkonfigurator“ erweitert werden. Als weiteres Beispiel könnte geprüft werden, wie das Umsetzen von variablen Lagenaufbauten der Bretter als weitere Funktion mitberücksichtigt werden könne.

3) Integration in übergeordnete BIM-Workflows:

Ein weiteres großes Potential liegt in der Einbettung des entwickelten Prozesses in umfassende BIM-Workflows. Anstelle der manuellen Dateneingabe in Excel könnten Wandgeometrien und Öffnungspositionen direkt aus einem Architekturmodell im IFC-Format ausgelesen werden. Das erzeugte analytische Modell könnte nach der Berechnung wiederum als angereichertes Datenmodell zurückgespielt werden. Dies würde den Planungsablauf weiter beschleunigen und die Datenkonsistenz über verschiedene Planungsphasen hinweg sicherstellen.

Zusammenfassend lässt sich sagen, dass der Entwickelte Prozess als Prototyp mit enormem Potential für weitere Entwicklungen betrachtet werden kann. Er hat das Potential in Automatisierungsprozessen bei der Modellierung eines analytischen Modells im Holzbau erörtert und eine Grundlage für weitere Entwicklungen gelegt.

Abbildungsverzeichnis

Abbildung 1 Übersicht möglicher Verklebungen eines CLT-Elements (EN 16351, S. 13 Bild 2 (geändert))	9
Abbildung 2 Scheibenwirkung durch Lasten in Wandebene (eigene Darstellung)	13
Abbildung 3 Abbildung aus dem ETA zum Nagelbild (ETA 24/0946 vgl. Seite 12 Abbildung)	24
Abbildung 4 Nagelmuster für Holznägel (ETA 24/0946 vgl. Seite 12 Abbildung 3)	25
Abbildung 5 Gegenüberstellung der Details aus dem ETA. links aus Abbildung 2 (ETA 24/0946 vgl. Seite 12 Abbildung 2), rechts (ETA 24/0946 vgl. Seite 14 Abbildung 6) aus Abbildung 6 . Blau markiert die jeweils vorhandene Lage der 0° Bretter	25
Abbildung 6 Detail Nagelmuster (ETA 24/0946 vgl. Seite 12 Abbildung 3)	26
Abbildung 7 Darstellung der Stabachsen und Kreuzungspunkte (eigene Darstellung nach ETA 24/0946 Seite 12 Abbildung 2)	27
Abbildung 8 Detail zu den vertikalen und horizontalen Rasterabständen der VBM (ETA 24/0946 vgl. Seite 13 Abbildungen 4 und 5)	28
Abbildung 9 Raster der Brettlagen und VBM im Modellansatz 2	30
Abbildung 10 Festlegung Koordinatensystem (eigene Darstellung)	36
Abbildung 11 Eingabebereich der Wandgeometrie. Hier ist beispielhaft eine 2x2m Wand eingegeben.	36
Abbildung 12 Eingabebereich Öffnungsgeometrie. Hier ist beispielhaft eine 1x1m Öffnung eingegeben.	37
Abbildung 13 Eingabebereich für die Startnummern bei der Knoten- / Liniennummerierung	37
Abbildung 14 Dropdown Menü für KLED-Festlegung mit direkter Ausgabe der entsprechenden Werte	38
Abbildung 15 Auswahlbereich für Berechnungen	38
Abbildung 16 Entwickelte Eingabeoberfläche	40
Abbildung 17 Flussdiagramm Automatisierungsprozess	43
Abbildung 18 VBA-Projektstruktur	43
Abbildung 19 Terminalausgabe: Ausschnitt mit Fehler beim Stabexport	49
Abbildung 20 Visuelle Kontrolle nach Datenexport des Eingabebeispiels	54
Abbildung 21 Visuelle Kontrolle: alleinstehender Knoten im Eingabebeispiel	54

Abbildung 22 Allgemeiner Workflow	I
Abbildung 23 RFEM6 Programmooptionen	I
Abbildung 24 RFEM6 Aktivierung Webservice	II
Abbildung 25 Installationssetup Python	II
Abbildung 26 Ersteinrichtung Pfade hinterlegen	IV
Abbildung 27 Eingabeoberfläche für das Eingabebeispiel	VI
Abbildung 28 Ausschnitt der Ergebnisse "Brettlagen" im Eingabebeispiel	VII
Abbildung 29 Ausschnitt der Ergebnisse "Nagelbild" im Eingabebeispiel	VII
Abbildung 30 Ausschnitt der Ergebnisse "Stäbe" im Eingabebeispiel	VIII
Abbildung 31 Leeres RFEM6 Modell geöffnet und bereit für Datenempfang aus Eingabebeispiel	VIII
Abbildung 32 Ergebnis nach automatischem Export des Eingabebeispiels: REALMODELL	IX
Abbildung 33 Ergebnis inkl. manuellem Export: REALMODELL	IX
Abbildung 34 Lager und Bettungen im Navigator	X
Abbildung 35 Knotenlager im Eingabebeispiel	X
Abbildung 36 Linienlager im Eingabebeispiel	XI
Abbildung 37 Modell des Eingabebeispiels inkl. Lager	XI
Abbildung 38 Eingabebeispiel verbinden sich kreuzender Stäbe	XII
Abbildung 39 Berechnungsergebnisse Realmodell mit 50kN Last, Blick in -y Richtung	XIII
Abbildung 40 Berechnungsergebnisse Idealmodell mit 50kN Last, Blick in -y Richtung	XIII

Tabellenverzeichnis

Tabelle 1 Komponenten der Woodbloc Massivholzwand (eigene Darstellung nach ETA 24/0946, S. 3)	24
Tabelle 2 Querschnitte Modellansatz 1	28
Tabelle 3 Abstände im Nagelbild beim Modellansatz 1	29
Tabelle 4 Querschnitte Modellansatz 2	30
Tabelle 5 Abstände im Nagelbild beim Modellansatz 2	31
Tabelle 6 System der Zuordnungsbezeichnung.....	42
Tabelle 7 Vergleich des Verformungsbildes Realmodell VS Idealmodell	55
Tabelle 8 Zeitaufwand bei Anwendung des entwickelten Prozesses	56
Tabelle 9 Vergleich manuellem zu automatisiertem Modellieren.....	57

Literaturverzeichnis

ARGEBAU. (2002). MUSTERBAUORDNUNG. MBO. Verfügbar unter:

<https://www.bauministerkonferenz.de/verzeichnis.aspx?id=1623&o=1623>

Borrmann, A., König, M., Koch, C. & Beetz, J. (Hrsg.). (2021). *Building Information Modeling. Technologische Grundlagen und industrielle Praxis* (Springer eBook Collection, 2., aktualisierte Auflage). Wiesbaden: Springer Fachmedien Wiesbaden.

<https://doi.org/10.1007/978-3-658-33361-4>

Brettsperrholz.org. ((o.A.)). *Brettsperrholz (BSP, X-Lam) - Vorteile der Brettsperrholzbauweise* - [brettsperrholz.org](https://www.brettsperrholz.org). Verfügbar unter:

https://www.brettsperrholz.org/brettsperrholz-bsp-x-lam/vorteile-der-brettsperrholzbauweise/mn_45181

DIN EN, 1995-1-1:2023 (10.2023). *Eurocode 5*.

ETA, 11/0190 (03.06.2013). *Würth selbstbohrende Schrauben*.

DIBt - FAQ Service. (o.A.). *Das deutsche Regelungssystem für Bauprodukte und Bauarten*.

Verfügbar unter: <https://www.dibt.de/de/service/faqs/das-deutsche-regelungssystem-fuer-bauprodukte-und-bauarten/>

DIN EN, 338:2016-07. *DIN EN 338:2016-07: Bauholz für tragende Zwecke*.

DIN EN, 61061-3-1 (09.1999). *Nicht imprägniertes Kunstharzpreßholz für elektrotechnische Zwecke - Teil 3*.

Dlubal API Abos. (2025). *Abos & Preise*. Verfügbar unter:

<https://www.dlubal.com/de/loesungen/dlubal-api/api-dokumentation/abos-und-preise>

Dlubal Online-Handbuch (Dlubal, Hrsg.). (2018, 6. März). *Schnittstellen und relevante Funktionen für BIM-orientiertes Arbeiten*. Verfügbar unter:

<https://www.dlubal.com/de/support-und-schulungen/support/knowledge-base/001509>

Dlubal Online-Handbuch. (2022). *Webservice und API für Statiksoftware*, Dlubal. Verfügbar unter: <https://www.dlubal.com/de/loesungen/online-dienste/webservice-und-api>

Dlubal Überblick. (2021). *RFEM 6 - FEM-Programm für Tragwerksplaner*. Verfügbar unter:

<https://www.dlubal.com/de/produkte/fem-statik-software-rfem/rfem/was-ist-rfem>

EN, 16351 (2013). *EN 16351: Timber structures - Cross laminated timber - Requirements*.

ETA, 23/0041 (27.03.2023). *LignoLoc stiftförmige Verbindungsmittel aus Holz*.

- EU-Bauproduktenverordnung online. (2025, 24. Juni). *Harmonisierte Normen*. Verfügbar unter: <https://www.eu-bauproduktenverordnung.de/de/inhalt/din-normen/harmonisiert>
- Fachverband Holzindustrie Österreich. (2024). Brettsperrholz-Merkblatt. Verfügbar unter: https://www.brettsperrholz.org/publish/binarydata/aktuelles/2024/stghb_bsp-holz-merkblatt_2024-11_ihb_print_241106.pdf
- Gerd Prause (o.A.). BIM im Holzbau | Holz | Planungsprozesse | Baunetz_Wissen. *BauNetz*. Verfügbar unter: <https://www.baunetzwissen.de/holz/fachwissen/planungsprozesse/bim-im-holzbau-6969865>
- Heinzmann, A. (2022). *Automatisierung und Digitalisierung Im Holzbau* (1st ed.). Wiesbaden: Springer Fachmedien Wiesbaden GmbH. Verfügbar unter: <https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=7127611>
- Informationsdienst Holz. (2010). Bauen mit Brettsperrholz. Tragende Massivholzelemente für Wand, Decke und Dach. *Holzbau Handbuch*, (Reihe 1 Teil 6 Folge 1). Verfügbar unter: https://www.brettsperrholz.org/publish/binarydata/pdfs/2016.12_neuauflagen/ifos/stghb_brettsperrholz_4-auflage_d_ihb_print_161031.pdf
- Johann Baptist Lindner. (2018). *Digitalisierung wird Holzbau verändern. Aber was bedeutet das für das Handwerk?*, Holzkurier. Verfügbar unter: <https://www.holzkurier.com/holzbau/2018/05/digitalisierung-wird-holzbau-veraendern.html>
- Lißner, K. & Rug, W. (2018). *Holzbausanierung beim Bauen im Bestand* (VDI-Buch, 2. Aufl. 2018). Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-662-50377-5>
- Mestek Peter. (2010). Berechnung und Bemessung von Brettsperrholz - ein Überblick. 4. *Europäischer Kongress für energieeffizientes Bauen mit Holz 2011*. Verfügbar unter: https://events.forum-holzbau.com/pdf/ebh11_mestek.pdf
- Michael Goodwin. (2024). *Was ist eine API (Application Programming Interface)?* | IBM, IBM. Verfügbar unter: <https://www.ibm.com/de-de/think/topics/api>
- Microsoft Learn (Microsoft, Hrsg.). (o.A.). *Erste Schritte mit VBA in Office*. Verfügbar unter: <https://learn.microsoft.com/de-de/office/vba/Library-Reference/Concepts/getting-started-with-vba-in-office>
- ETA, 24/0946 (22.11.2024). *Woodbloc Massivholzwand*. Österreich: OiB.
- Oliver Justus.. *Regelung-von-Bauprodukten-und-Bauarten*. *Gütegemeinschaft Holzbau-Ausbau-Dachbau e.V.* Verfügbar unter: https://www.ghad.de/fileadmin/sites/ghad/Informationen_rechte_Seite/Regelung-von-Bauprodukten-und-Bauarten.pdf

- Pfeifer Group – Passion for Timber. (2025, 19. Junia). *Das Produkt - Pfeifer Group – Passion for Timber*. Verfügbar unter:
<https://www.pfeifergroup.com/de/produkte/holzbau/brettschichtholz/das-produkt/>
- Pfeifer Group – Passion for Timber. (2025, 19. Junib). *Sortiment & Daten - Pfeifer Group – Passion for Timber*. Verfügbar unter:
<https://www.pfeifergroup.com/de/produkte/holzbau/brettschichtholz/sortiment-daten/>
- ProHolz Austria. (2008). Massiv über Kreuz. *zuschnitt*, 8(31). Verfügbar unter:
<https://www.proholz.at/fileadmin/flippingbooks/zuschnitt31/files/assets/common/downloads/publication.pdf>
- ProHolz Austria. (2015). 25 Jahre (pro) Holz. *zuschnitt*, 15(60). Verfügbar unter:
<https://www.proholz.at/fileadmin/flippingbooks/zuschnitt60/files/assets/common/downloads/publication.pdf>
- Python.org. (o.A.). *About Python*, PSF. Verfügbar unter: <https://www.python.org/about/>
- Schickhofer, G. (Hrsg.). (2010). *BSPhandbuch, Holz- Massivbauweise in Brettsperrholz. Nachweise auf Basis des neuen europäischen Normenkonzepts* (2., überarb. Auflage). Graz: Verlag d. Technischen Universität Graz.
- Simon Aicher. Geklebte Verbindungen in Holzbauprodukten und -tragwerken. In.
<https://doi.org/10.13140/2.1.4101.6005>
- EAD, 130002-00-0304. *Solid wood slab element - element of dowel jointed timber boards to be used as a structural element in buildings*.
- Stefan Krötsch (o.A.). Geschichte des Holzbaus | Holz | Einführung | Baunetz_Wissen. *BauNetz*. Verfügbar unter:
<https://www.baunetzwissen.de/holz/fachwissen/einfuehrung/geschichte-des-holzbaus-6640622>

Anhang I. Anweisungen Ersteinrichtung

Die Arbeit mit dem bereitgestellten Modell sieht folgenden Workflow vor:

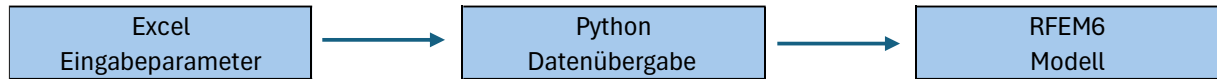


Abbildung 22 Allgemeiner Workflow

Bei erstmaliger Benutzung dieses Setups an einem neuen Rechner ist eine Ersteinrichtung notwendig, um

- 1) Die notwendige Software inkl. Bibliotheken zu installieren
- 2) Die korrekten Pfade in der Excel mit der Eingabeoberfläche zu hinterlegen

Da dieses Setup auf mehrere unterschiedlichen Dateien zurückgreift, empfiehlt es sich alle notwendigen Dateien in einen gemeinsamen Ordner anzulegen. Zur Veranschaulichung wird im Folgenden der Ordner mit dem Namen Ordner_Setup bezeichnet. Dieser beinhaltet:

- 1) Excel mit dem Modell (Modellansatz 1 oder Modellansatz 2)
- 2) ExportRFEM.py
- 3) Requirements.txt

Software

Der entwickelte Automatisierungsprozess sieht die Einbindung folgender Software vor:

- 1) MS Office Excel
- 2) Python 3.13
- 3) RFEM6

Die genannten Programme sind auf dem Rechner des Anwenders zu installieren.

Hinweis zu REFM6:

Es ist sicherzustellen, dass der Webservice in RFEM6 aktiviert ist. Dieser findet sich unter:

Optionen – Programmooptionen – API-Dienste & Webservice

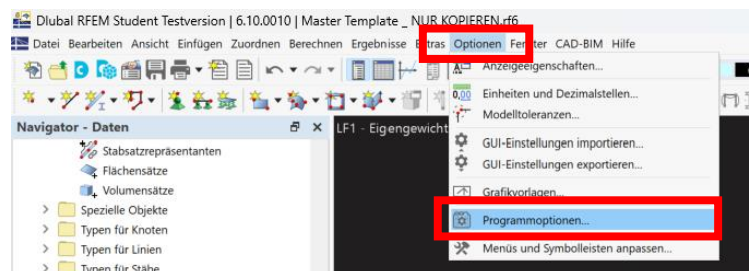


Abbildung 23 RFEM6 Programmooptionen

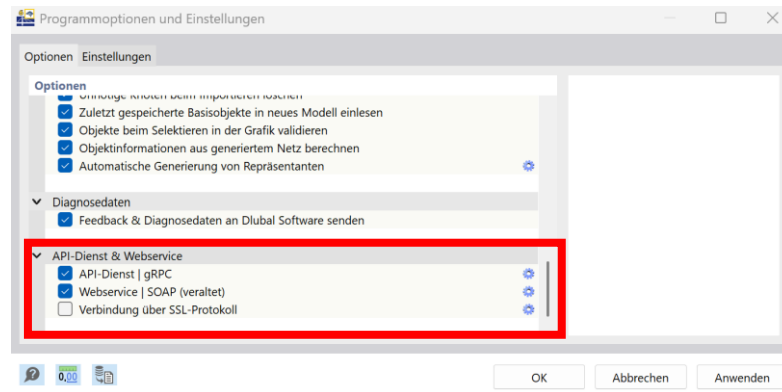


Abbildung 24 RFEM6 Aktivierung Webservice

Hinweis zu Python:

Entweder aus dem MS Store installieren (Python bei der Suche eingeben und neuste Version installieren) oder unter <https://www.python.org/downloads/> herunterladen und installieren. Es müssen Adminrechte aktiviert und python.exe zu PATH hinzugefügt werden:



Abbildung 25 Installationssetup Python

Nach erfolgreicher Installation sind die notwendigen Bibliotheken in Python zu installieren, welche die Kommunikation zwischen Python und RFEM6 ermöglichen. Zu den notwendigen Bibliotheken gehören:

certifi
charset-normalizer
colorama
et_xmlfile
exceptiongroup
idna
iniconfig
mock
numpy
openpyxl
packaging
pandas
pluggy

Pygments
pytest
python-dateutil
pytz
requests
RFEM
six
suds-py3
tomli
typing_extensions
tzdata
urllib3
xmltod

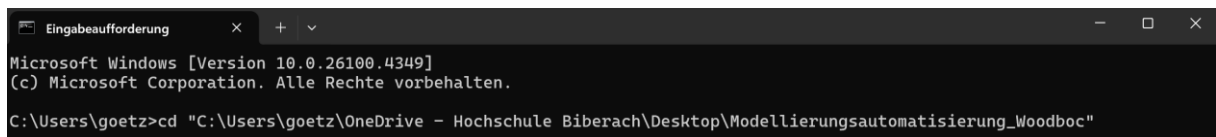
Um diese Bibliotheken installieren zu können kann entweder manuell vorgegangen werden, oder automatisiert mithilfe einer .txt Datei. Die manuelle Installation ist folgenderweise durchzuführen:

- 1) Eingabeaufforderung starten (unter Windows-Suche *CMD* eingeben und öffnen)
- 2) Im CMD eingeben: *pip install NAME_DER_BIBLIOTHEK*
- 3) ENTER

Da dies bei der Menge an erforderlichen Bibliotheken aufwendig ist, bietet es sich an dies automatisiert durchzuführen. Dazu wird eine .txt Datei mit den Namen aller Bibliotheken erstellt und in Ordner_Setup unter dem Namen requirements.txt abgespeichert.

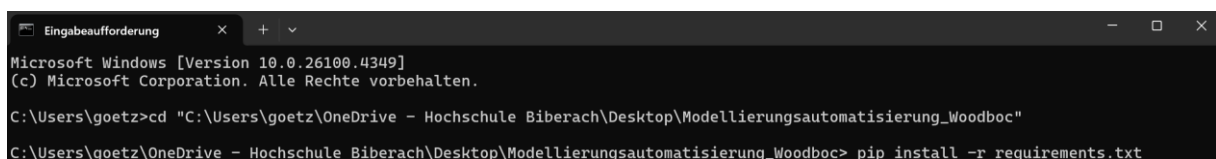
Die Bibliotheken können nun automatisiert wie folgt installiert werden:

- 1) Eingabeaufforderung starten (unter Windows-Suche *CMD* eingeben und öffnen)
- 2) Pfad zum Ordner mit der.txt Datei angeben:
 - a. Rechtsklick auf den Ordner, in welchem die Datei liegt und *Als Pfad kopieren* auswählen
 - b. Im CMD eingeben: *cd HIER JETZT DEN PFAD EINFÜGEN*
 - c. ENTER



```
Eingabeaufforderung
Microsoft Windows [Version 10.0.26100.4349]
(c) Microsoft Corporation. Alle Rechte vorbehalten.
C:\Users\goetz>cd "C:\Users\goetz\OneDrive - Hochschule Biberach\Desktop\Modellierungsautomatisierung_Woodboc"
```

- 3) Bibliotheken installieren:
 - a. Eingeben: *pip install -r requirements.txt*
 - b. ENTER



```
Eingabeaufforderung
Microsoft Windows [Version 10.0.26100.4349]
(c) Microsoft Corporation. Alle Rechte vorbehalten.
C:\Users\goetz>cd "C:\Users\goetz\OneDrive - Hochschule Biberach\Desktop\Modellierungsautomatisierung_Woodboc"
C:\Users\goetz\OneDrive - Hochschule Biberach\Desktop\Modellierungsautomatisierung_Woodboc> pip install -r requirements.txt
```

Das CMD kann geschlossen werden, sobald alle Bibliotheken installiert sind.

Pfade hinterlegen

Bei erstmaliger Nutzung müssen zu Beginn einmalig die richtigen Pfade in der Excel zur Modelleingabe hinterlegt werden. Wie auch alle anderen Eingaben und Konfigurationen geschieht dies im Arbeitsblatt *Input*, zu hinterlegen sind:

- 1) Pfad der Python Export Datei
- 2) Pfad zur Python.exe

Woodbloc Massivholzwand

Wandgeometrie

	Von[m]	Bis[m]	Angepasst*
x	0	2	2,03
z	0	2	2,08

| | : Alle Werte sind ohne Vorzeichen einzugeben
 * : Geometrie wird auf Raster d. Bretter angepasst

Zu berechnen:
☒ Brettlagen WAHR
☒ Nagelbild WAHR

Öffnungen - Berücksichtigt nur Nagelbild

	Von[m]	Bis[m]
x		
z		

Startnummern (max. 999 999)

1	Knoten	100
2	Linien & Stäbe	100

Konfigurationen

3	KLED der Wand	mittel
4	$K_{ser,LignoLoc}$	677,37 N/mm
5	fiktives Schubmodul $G'_{LignoLoc}$	890,17 N/mm ²
6	Schrauben im vertikalen Randbereich	☐ FALSCH
7	Modellname im R-FEM 6 Modell?	
8	Pfad der Python Export Datei	C:\Users\goetz\OneDrive -
9	Pfad zur Python.exe	C:\Users\goetz\AppData\L
10	Blattschutz aktivieren	☒ WAHR

Abbildung 26 Ersteinrichtung Pfade hinterlegen

1) Pfad der Python Excel Datei

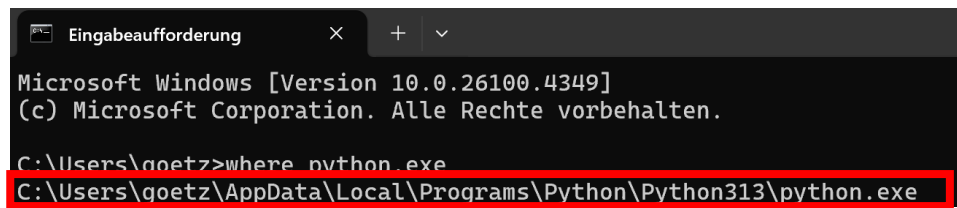
Hier ist der Pfad zum Python Skript *ExportRFEM.py* zu hinterlegen. Diese findet sich im Ordner. Rechtsklick auf die Datei *ExportRFEM* machen und *Als Pfad kopieren* auswählen. Der Pfad ist im dafür vorgesehenen Feld einzufügen (falls beim Kopieren Anführungsstriche mit kopiert werden, diese entfernen).

2) Pfad zur Python.exe

Nun ist der Pfad zum Installationsort der Python.exe zu hinterlegen. Falls dieser nicht bekannt ist:

- a) CMD öffnen
- b) Eingeben: `where python.exe`
- c) ENTER

d) Pfad kopieren



```
Eingabeaufforderung
Microsoft Windows [Version 10.0.26100.4349]
(c) Microsoft Corporation. Alle Rechte vorbehalten.
C:\Users\goetz>where python.exe
C:\Users\goetz\AppData\Local\Programs\Python\Python313\python.exe
```

Der kopierte Pfad ist nun einfach in der entsprechenden Zelle einzufügen.

Nun ist die Einrichtung abgeschlossen und es kann mit der eigentlichen Anwendung begonnen werden.

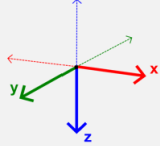
Anhang II. Ergebnisse aus dem Eingabebeispiel

Die Konfiguration der Eingabeoberfläche ist folgenderweise eingestellt:

Woodbloc Massivholzwand

Wandgeometrie

	Von[m]	Bis[m]	Angepasst*
x	0	4	4,07
z	0	2,8	2,88



| : Alle Werte sind ohne Vorzeichen einzugeben
* : Geometrie wird auf Raster d. Bretter angepasst

Zu berechnen:

☒ Brettlagen WAHR
 ☒ Nagelbild WAHR

Berechnen
Export nach RFEM

Öffnungen - Berücksichtigt nur Nagelbild

	Von[m]	Bis[m]
x		
z		

	Von[m]	Bis[m]
x		
z		

Weitere Öffnung hinzufügen
Letzte Öffnung löschen

Startnummern (max. 999 999)

1	Knoten	100
2	Linien & Stäbe	100

Konfigurationen

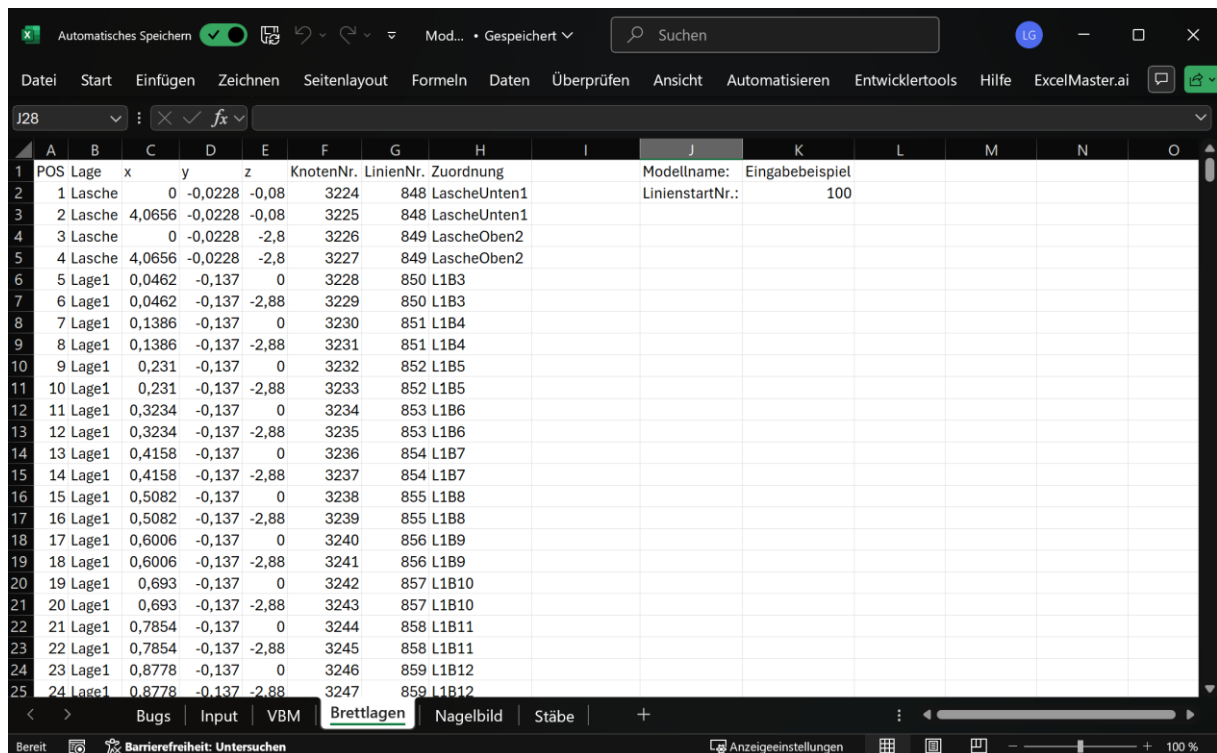
3	KLED der Wand	mittel
4	$K_{ser, LignoLoc}$	677,37 N/mm
5	fiktives Schubmodul $G'_{LignoLoc}$	890,17 N/mm ²
6	Schrauben im vertikalen Randbereich	☐ FALSCH
7	Modellname im R-FEM 6	Eingabebeispiel
8	Pfad der Python Export Datei	C:\Users\goetz\OneDrive - I

Allgemeines

9	Pfad zur Python.exe	C:\Users\goetz\AppData\Lc
10	Blattschutz aktivieren	☒ WAHR

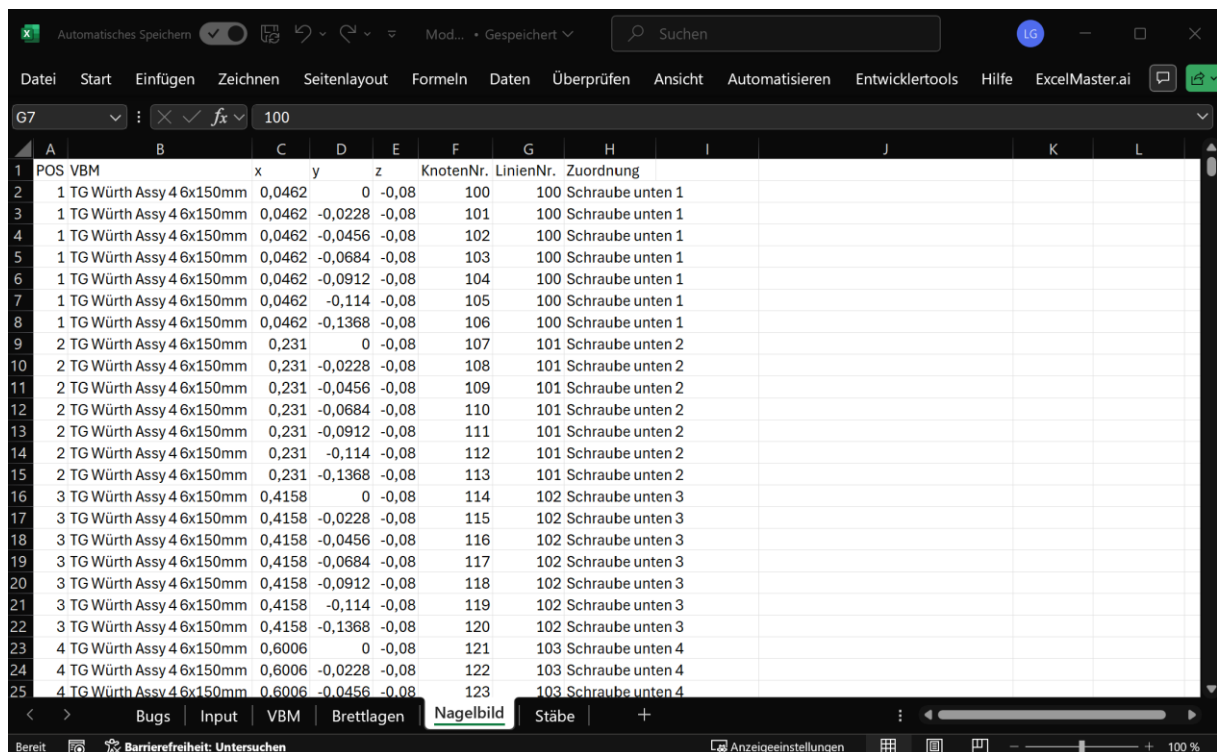
Abbildung 27 Eingabeoberfläche für das Eingabebeispiel

Nach der Berechnung wurden folgende Ergebnisse in den Arbeitsblättern „Brettlagen“, „Nagelbild“ und „Stäbe“ ausgegeben:



POS	Lage	x	y	z	KnotenNr.	LinienNr.	Zuordnung	Modellname	Eingabebeispiel
1	Lasche	0	-0,0228	-0,08	3224	848	LascheUnten1	LinienstartNr.:	100
2	Lasche	4,0656	-0,0228	-0,08	3225	848	LascheUnten1		
3	Lasche	0	-0,0228	-2,8	3226	849	LascheOben2		
4	Lasche	4,0656	-0,0228	-2,8	3227	849	LascheOben2		
5	Lage1	0,0462	-0,137	0	3228	850	L1B3		
6	Lage1	0,0462	-0,137	-2,88	3229	850	L1B3		
7	Lage1	0,1386	-0,137	0	3230	851	L1B4		
8	Lage1	0,1386	-0,137	-2,88	3231	851	L1B4		
9	Lage1	0,231	-0,137	0	3232	852	L1B5		
10	Lage1	0,231	-0,137	-2,88	3233	852	L1B5		
11	Lage1	0,3234	-0,137	0	3234	853	L1B6		
12	Lage1	0,3234	-0,137	-2,88	3235	853	L1B6		
13	Lage1	0,4158	-0,137	0	3236	854	L1B7		
14	Lage1	0,4158	-0,137	-2,88	3237	854	L1B7		
15	Lage1	0,5082	-0,137	0	3238	855	L1B8		
16	Lage1	0,5082	-0,137	-2,88	3239	855	L1B8		
17	Lage1	0,6006	-0,137	0	3240	856	L1B9		
18	Lage1	0,6006	-0,137	-2,88	3241	856	L1B9		
19	Lage1	0,693	-0,137	0	3242	857	L1B10		
20	Lage1	0,693	-0,137	-2,88	3243	857	L1B10		
21	Lage1	0,7854	-0,137	0	3244	858	L1B11		
22	Lage1	0,7854	-0,137	-2,88	3245	858	L1B11		
23	Lage1	0,8778	-0,137	0	3246	859	L1B12		
24	Lage1	0,8778	-0,137	-2,88	3247	859	L1B12		

Abbildung 28 Ausschnitt der Ergebnisse "Brettlagen" im Eingabebeispiel



POS	VBM	x	y	z	KnotenNr.	LinienNr.	Zuordnung	Modellname	Eingabebeispiel
1	1 TG Würth Assy 4 6x150mm	0,0462	0	-0,08	100	100	Schraube unten 1	LinienstartNr.:	100
2	1 TG Würth Assy 4 6x150mm	0,0462	-0,0228	-0,08	101	100	Schraube unten 1		
3	1 TG Würth Assy 4 6x150mm	0,0462	-0,0456	-0,08	102	100	Schraube unten 1		
4	1 TG Würth Assy 4 6x150mm	0,0462	-0,0684	-0,08	103	100	Schraube unten 1		
5	1 TG Würth Assy 4 6x150mm	0,0462	-0,0912	-0,08	104	100	Schraube unten 1		
6	1 TG Würth Assy 4 6x150mm	0,0462	-0,114	-0,08	105	100	Schraube unten 1		
7	1 TG Würth Assy 4 6x150mm	0,0462	-0,1368	-0,08	106	100	Schraube unten 1		
8	2 TG Würth Assy 4 6x150mm	0,231	0	-0,08	107	101	Schraube unten 2		
9	2 TG Würth Assy 4 6x150mm	0,231	-0,0228	-0,08	108	101	Schraube unten 2		
10	2 TG Würth Assy 4 6x150mm	0,231	-0,0456	-0,08	109	101	Schraube unten 2		
11	2 TG Würth Assy 4 6x150mm	0,231	-0,0684	-0,08	110	101	Schraube unten 2		
12	2 TG Würth Assy 4 6x150mm	0,231	-0,0912	-0,08	111	101	Schraube unten 2		
13	2 TG Würth Assy 4 6x150mm	0,231	-0,114	-0,08	112	101	Schraube unten 2		
14	2 TG Würth Assy 4 6x150mm	0,231	-0,1368	-0,08	113	101	Schraube unten 2		
15	3 TG Würth Assy 4 6x150mm	0,4158	0	-0,08	114	102	Schraube unten 3		
16	3 TG Würth Assy 4 6x150mm	0,4158	-0,0228	-0,08	115	102	Schraube unten 3		
17	3 TG Würth Assy 4 6x150mm	0,4158	-0,0456	-0,08	116	102	Schraube unten 3		
18	3 TG Würth Assy 4 6x150mm	0,4158	-0,0684	-0,08	117	102	Schraube unten 3		
19	3 TG Würth Assy 4 6x150mm	0,4158	-0,0912	-0,08	118	102	Schraube unten 3		
20	3 TG Würth Assy 4 6x150mm	0,4158	-0,114	-0,08	119	102	Schraube unten 3		
21	3 TG Würth Assy 4 6x150mm	0,4158	-0,1368	-0,08	120	102	Schraube unten 3		
22	4 TG Würth Assy 4 6x150mm	0,6006	0	-0,08	121	103	Schraube unten 4		
23	4 TG Würth Assy 4 6x150mm	0,6006	-0,0228	-0,08	122	103	Schraube unten 4		
24	4 TG Würth Assy 4 6x150mm	0,6006	-0,0456	-0,08	123	103	Schraube unten 4		

Abbildung 29 Ausschnitt der Ergebnisse "Nagelbild" im Eingabebeispiel

Linie Nr.	Stabtyp	Querschnittsverteilung	Drehung [°] / Knoten Nr.	Anfang i	Ende j	Innen k
1	100 Balkenstab	Gleichmäßig	0	1	1	1
2	101 Balkenstab	Gleichmäßig	0	1	1	1
3	102 Balkenstab	Gleichmäßig	0	1	1	1
4	103 Balkenstab	Gleichmäßig	0	1	1	1
5	104 Balkenstab	Gleichmäßig	0	1	1	1
6	105 Balkenstab	Gleichmäßig	0	1	1	1
7	106 Balkenstab	Gleichmäßig	0	1	1	1
8	107 Balkenstab	Gleichmäßig	0	1	1	1
9	108 Balkenstab	Gleichmäßig	0	1	1	1
10	109 Balkenstab	Gleichmäßig	0	1	1	1
11	110 Balkenstab	Gleichmäßig	0	1	1	1
12	111 Balkenstab	Gleichmäßig	0	1	1	1
13	112 Balkenstab	Gleichmäßig	0	1	1	1
14	113 Balkenstab	Gleichmäßig	0	1	1	1
15	114 Balkenstab	Gleichmäßig	0	1	1	1
16	115 Balkenstab	Gleichmäßig	0	1	1	1
17	116 Balkenstab	Gleichmäßig	0	1	1	1
18	117 Balkenstab	Gleichmäßig	0	1	1	1
19	118 Balkenstab	Gleichmäßig	0	1	1	1
20	119 Balkenstab	Gleichmäßig	0	1	1	1
21	120 Balkenstab	Gleichmäßig	0	1	1	1
22	121 Balkenstab	Gleichmäßig	0	1	1	1
23	122 Balkenstab	Gleichmäßig	0	2	2	2
24	123 Balkenstab	Gleichmäßig	0	2	2	2

Abbildung 30 Ausschnitt der Ergebnisse "Stäbe" im Eingabebeispiel

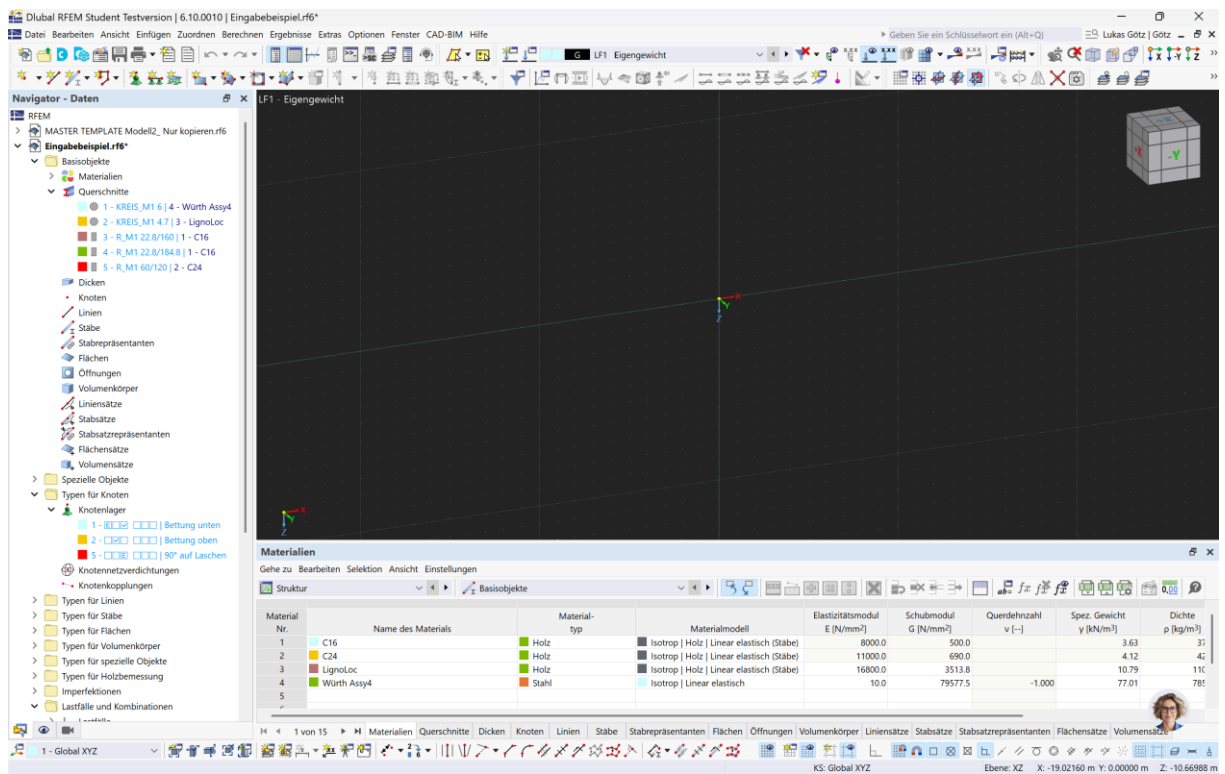


Abbildung 31 Leeres RFEM6 Modell geöffnet und bereit für Datenempfang aus Eingabebeispiel

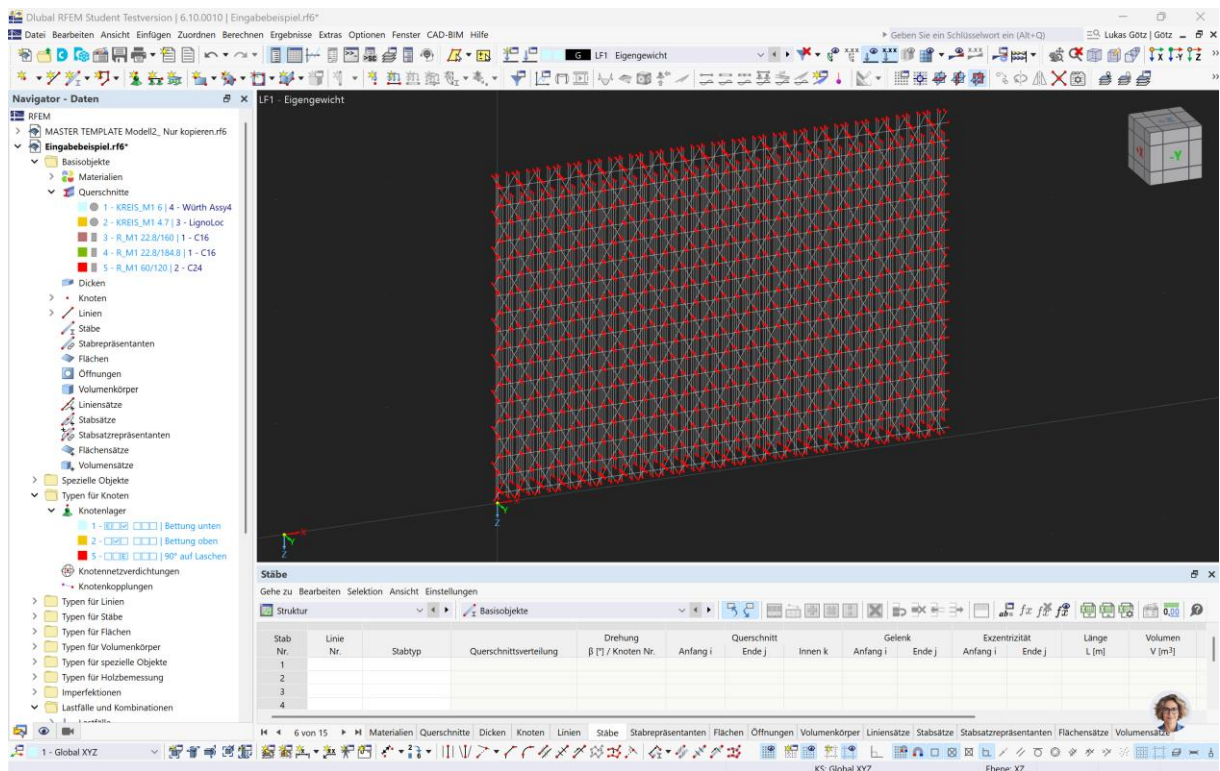


Abbildung 32 Ergebnis nach automatischem Export des Eingabebeispiels: REALMODELL

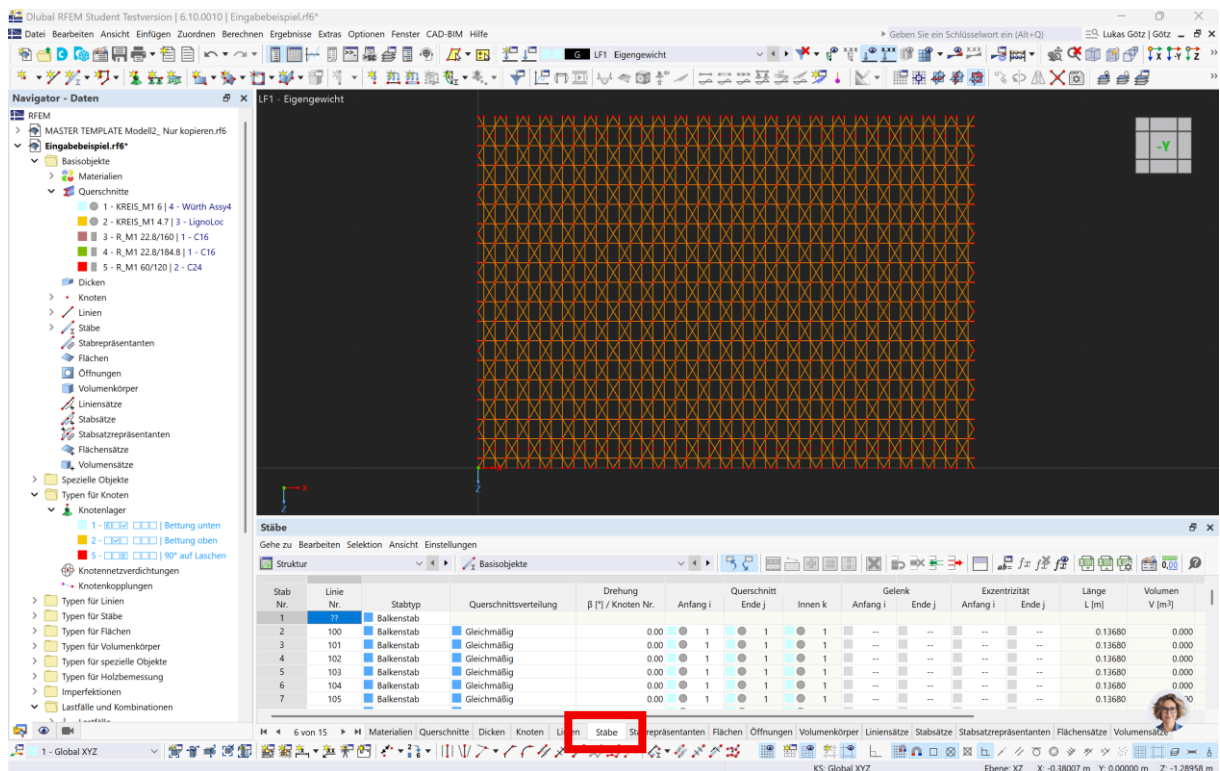


Abbildung 33 Ergebnis inkl. manuellem Export: REALMODELL

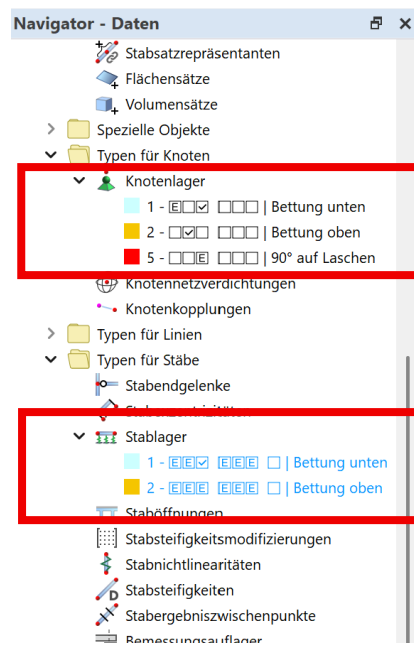


Abbildung 34 Lager und Bettungen im Navigator

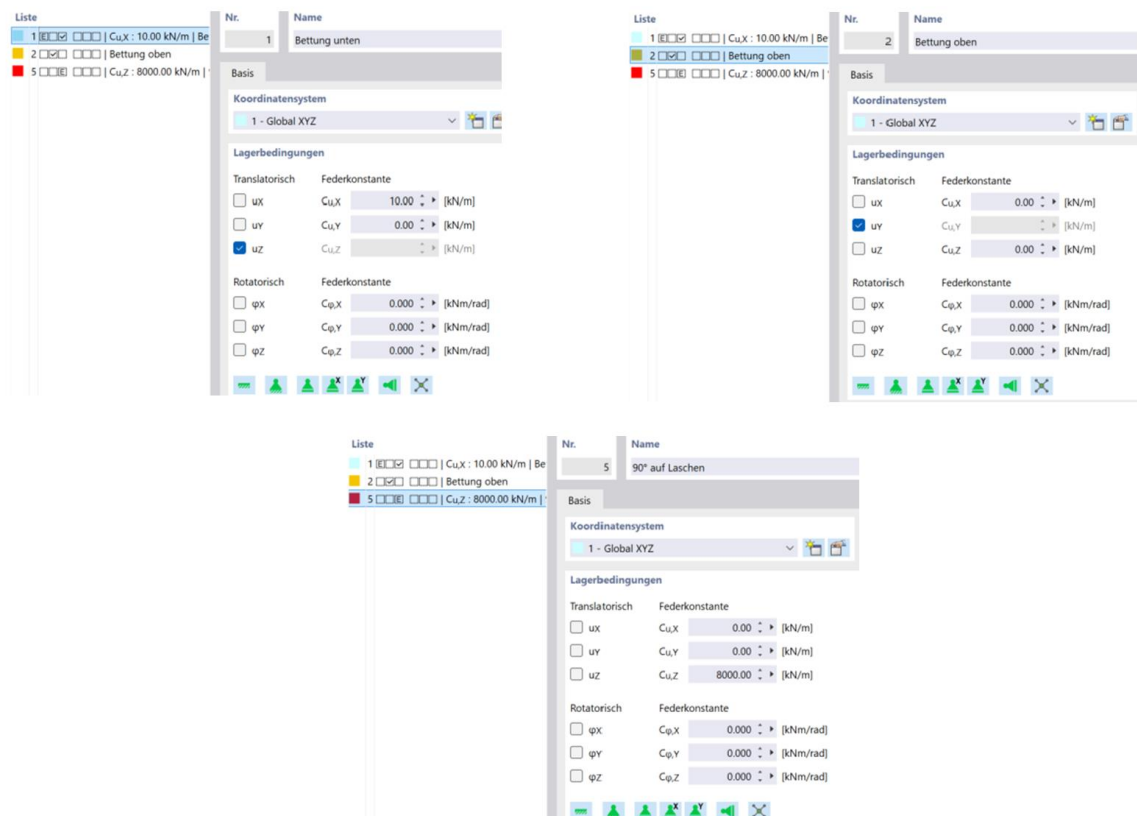


Abbildung 35 Knotenlager im Eingabebeispiel

Liste

Nr.	Name
1	Bettung unten
2	Bettung oben

Basis

Lagerbedingungen

Translatorisch

☐ u_x $C_{u,x}$ 10.000 [kN/m²]

☐ u_y $C_{u,y}$ 10.000 [kN/m²]

☒ u_z $C_{u,z}$ [kN/m²]

Schub

☐ s_x $C_{s,x}$ 10.000 [kN]

☐ s_y $C_{s,y}$ 10.000 [kN]

☐ s_z $C_{s,z}$ 15.000 [kN]

Rotatorisch

☐ ϕ_x $C_{\phi,x}$ 0.000 [kNm·rad⁻¹·m⁻¹]

Liste

Nr.	Name
1	Bettung oben
2	Bettung unten

Basis

Lagerbedingungen

Translatorisch

☐ u_x $C_{u,x}$ 10.000 [kN/m²]

☐ u_y $C_{u,y}$ 10.000 [kN/m²]

☐ u_z $C_{u,z}$ 8000.000 [kN/m²]

Schub

☐ s_x $C_{s,x}$ 10.000 [kN]

☐ s_y $C_{s,y}$ 10.000 [kN]

☐ s_z $C_{s,z}$ 15.000 [kN]

Rotatorisch

☐ ϕ_x $C_{\phi,x}$ 0.000 [kNm·rad⁻¹·m⁻¹]

Abbildung 36 Linienlager im Eingabebeispiel

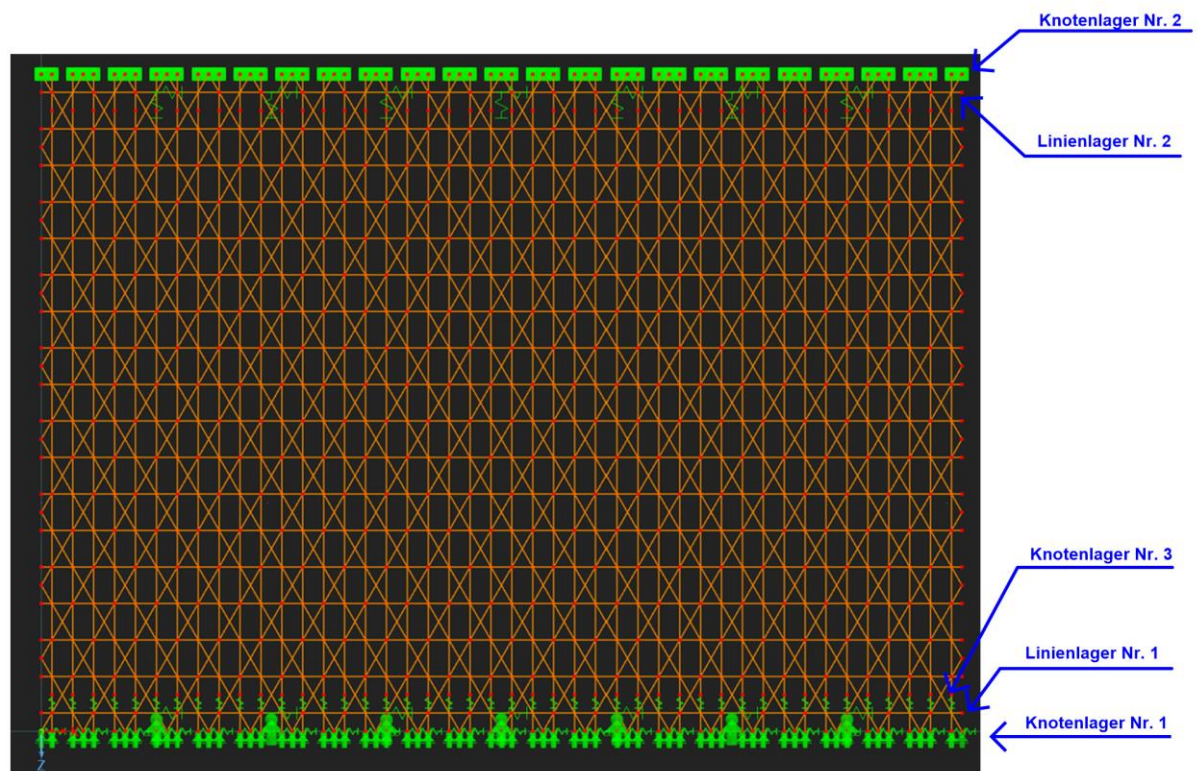


Abbildung 37 Modell des Eingabebeispiels inkl. Lager

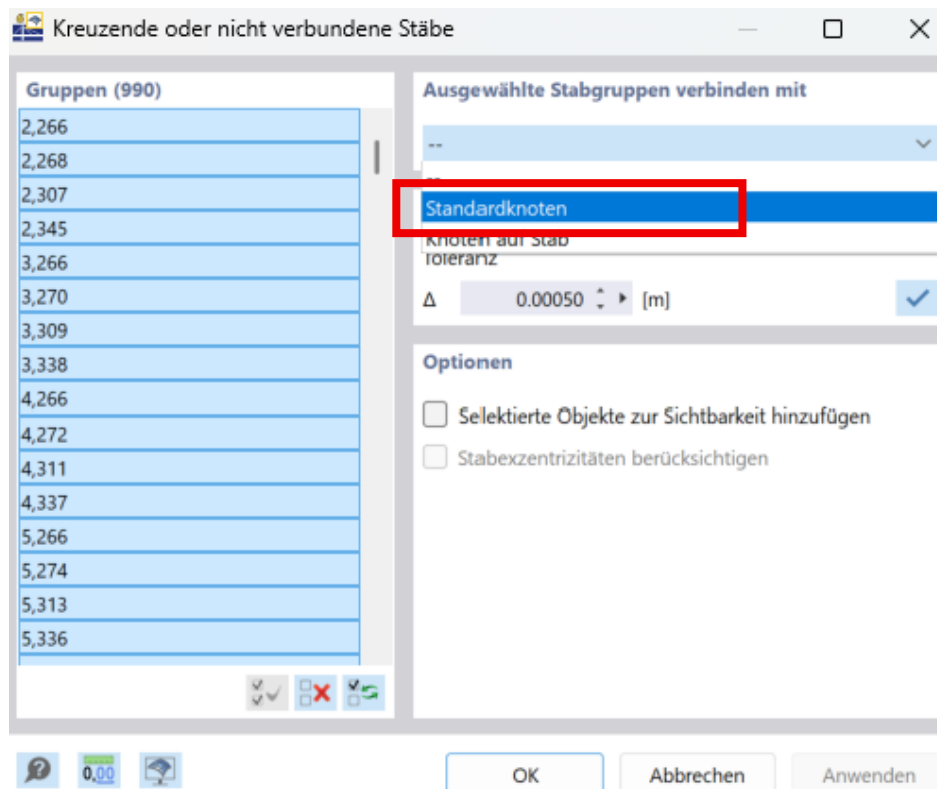
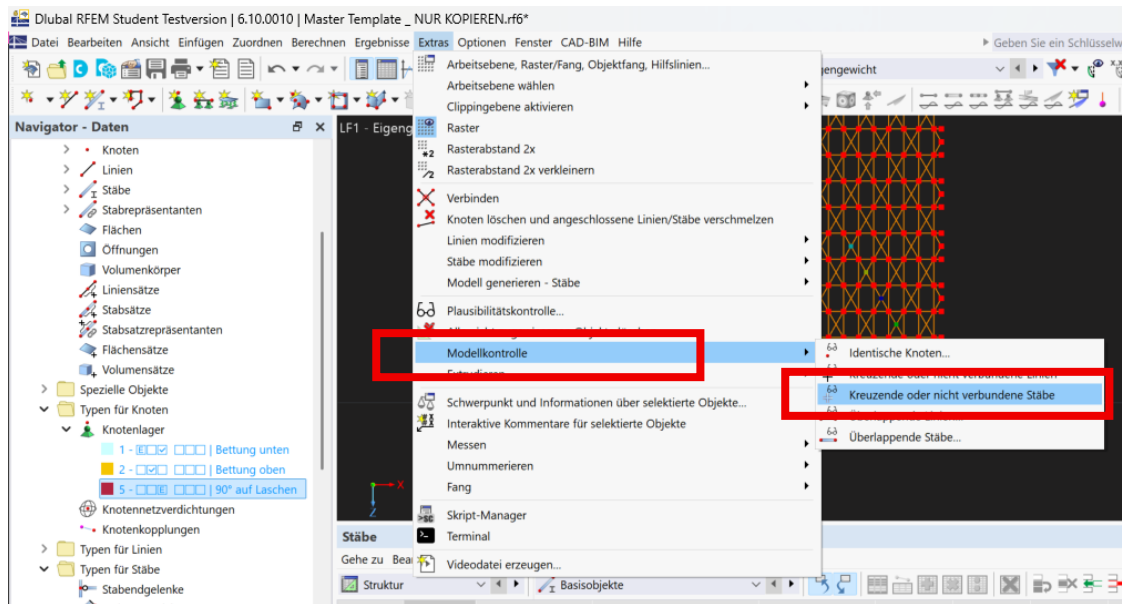


Abbildung 38 Eingabebeispiel verbinden sich kreuzender Stäbe

Anhang III. Verformungsergebnisse Real- & Idealmodell

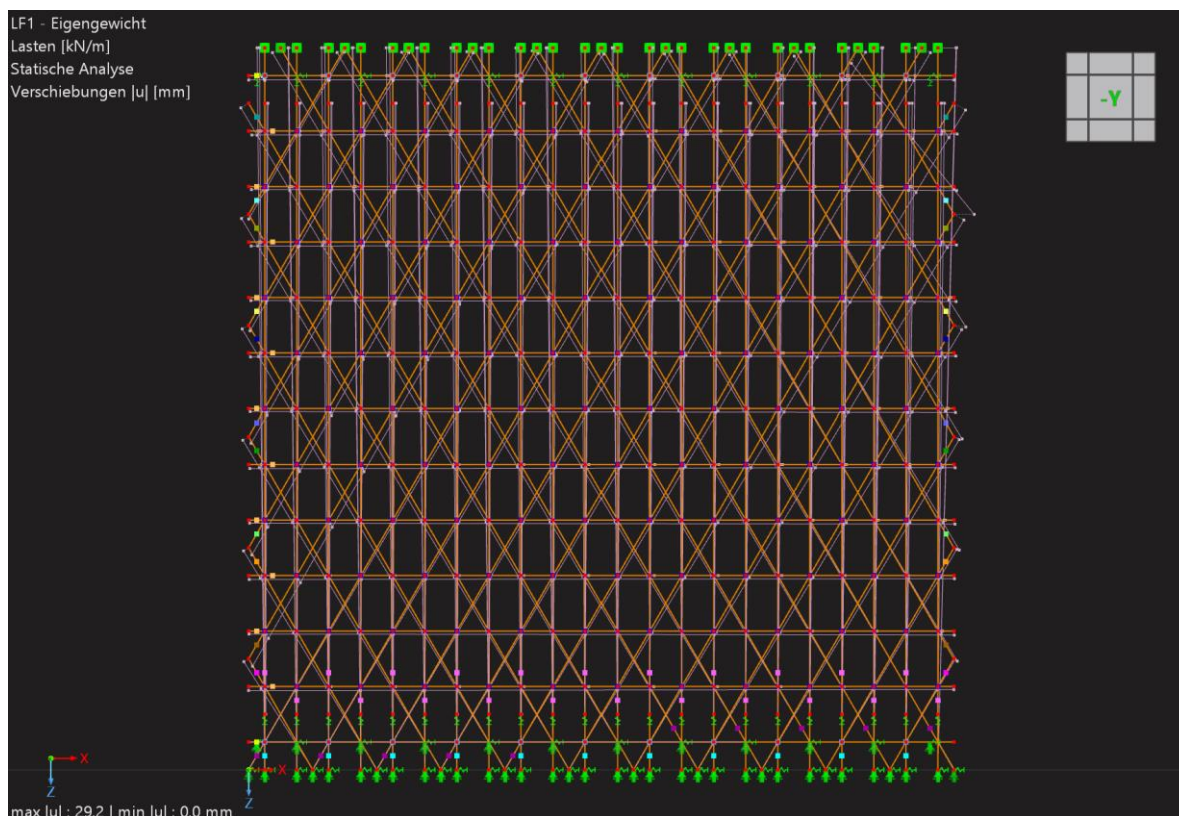


Abbildung 39 Berechnungsergebnisse Realmodell mit 50kN Last, Blick in -y Richtung

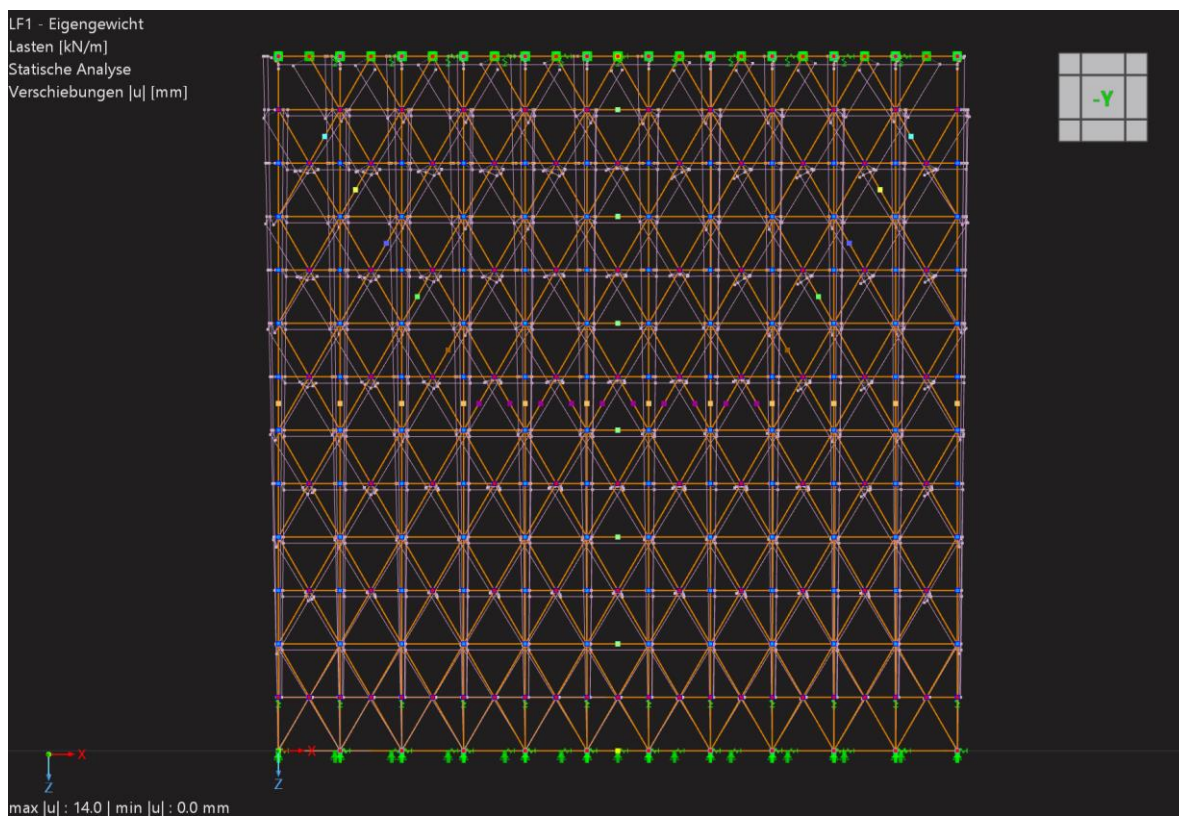


Abbildung 40 Berechnungsergebnisse Idealmodell mit 50kN Last, Blick in -y Richtung

Anhang IV. CODE REALMODELL

```

Sub Gesamtberechnung()          ' Hauptsub zur zentralen Steuerung

    Dim wsInput As Worksheet: Set wsInput = ThisWorkbook.Sheets("Input")

    Dim berechneBrettlagen As Boolean
    Dim berechneNagelbild As Boolean
    Dim startKnoten As Long
    Dim startLinien As Long

    berechneBrettlagen = CBool(wsInput.Range("Brettlagen").Value)
    berechneNagelbild = CBool(wsInput.Range("Nagelbild").Value)

    If Not berechneBrettlagen And Not berechneNagelbild Then
        MsgBox "Bitte mindestens eine Option unter 'Zu berechnen'
auswählen"
        Exit Sub
    End If

    ' zentrale Startnummern einlesen
    If Not IsNumeric(wsInput.Range("StartKnoten").Value) Or Not _
IsNumeric(wsInput.Range("StartLinien").Value) Then
        MsgBox "Ungültige Startwerte!", vbCritical
        Exit Sub
    End If
    startKnoten = CLng(wsInput.Range("StartKnoten").Value)
    startLinien = CLng(wsInput.Range("StartLinien").Value)

    ' Weitergabe als Referenz
    If berechneNagelbild Then Call Nagelbild_Erstellung(startKnoten,
startLinien)
    If berechneBrettlagen Then Call Wand_Erstellen(startKnoten,
startLinien)

    Call StabZuteilung
    Call ErstelleExportdatei(berechneBrettlagen, berechneNagelbild)

End Sub

```

```

Sub Nagelbild_Erstellung(ByRef knotennummerAktuell As Long, ByRef
liniennummerAktuell As Long)

    On Error GoTo ErrorHandler 'Regelt Umgang mit Laufzeitfehlern

    '1 Schritt: Basics festlegen

    Dim wsInput As Worksheet, wsOutput As Worksheet 'Vorhandensein d.
Arbeitsblätter kontrollieren
    Dim längeX1 As Double, längeX2 As Double 'Hier Wandlänge: von längeX1
bis längeX2
    Dim höheZ1 As Double, höheZ2 As Double 'Hier Wandhöhe: von höheZ1 bis
höheZ2
    Dim länge As Double, höhe As Double 'Hier kommen Wandabmessungen rein
    Dim länge2 As Double, höhe2 As Double 'Wandlängen nun Vielfache des
Rasters
    Dim zeile As Long
    Dim SchraubenVertikal As Boolean

```

```

Set wsInput = ThisWorkbook.Sheets("Input")
Set wsOutput = ThisWorkbook.Sheets("Nagelbild")
If wsInput Is Nothing Or wsOutput Is Nothing Then
    MsgBox "Fehler: Arbeitsblatt *Input* oder *Nagelblatt* nicht
gefunden!", vbCritical
    Exit Sub
End If

SchraubenVertikal = CBool(wsInput.Range("AbfrageVertikal").Value)

'2 Schritt: Wandgeometrie einlesen und auf Raster Runden

If Not WandGeometrieEinlesen(wsInput, längeX1, längeX2, höheZ1, höheZ2,
länge, länge2, höhe, höhe2, SchraubenVertikal) Then Exit Sub

'3.1 Öffnungen einlesen

Dim Öffnungen() As Variant
Dim Öffnungszähler As Long
Call ÖffnungenEinlesen(wsInput, Öffnungen, Öffnungszähler)

' 4 Schritt: Arbeitsblatt für das Nagelbild vorbereiten

wsOutput.Cells.Clear
wsOutput.Range("A1:H1").Value = Array("POS", "VBM", "x", "y", "z",
"KnotenNr.", "LinienNr.", "Zuordnung") 'Kopfzeile der Ausgabe
zeile = 2 ' Dateninhalt soll dann ab 2. Zeile beginnen

' 5 Schritt: Abstände aus ETA-24-0946 als Konstanten festlegen
'5.1 Horizontale Abstände
Const dR1 As Double = 0.0462 'Randabstand 1 beträgt 46,2mm
Const dR2 As Double = dR1 * 3
Const dF As Double = 0.1848 'Normaler Anstand im Feld

'5.2 Vertikale Abstände

Const dVR As Double = 0.08 'Vertikaler Randabstand oben und unten
jeweils 80mm
Const dV As Double = 0.16 'vertikaler Abstand innerhalb einer Nagellage
= 0.16

'6 Schritt: Beginnen mit unterer horizontaler Schraubenreihe (auf
Laschenbrett)
Dim K_SU As Double
Dim x As Double, y As Double, z As Double
Dim POS As Long
Dim i As Integer 'setzt bei jeden Lagenschnitt eines VBMs ein
Knotenpunkt

POS = 1
x = längeX1
z = höheZ1
K_SU = 1

Do While x <= längeX2 - dR2
    If x < längeX1 + dR1 Then 'Erste Schraube wird mit 46,2mm
Randabstand gesetzt
        x = längeX1 + dR1

```

```

        If Not KnotenInÖffnung(x, z, Öffnungen, Öffnungszähler)
Then
        For i = 0 To 6
            y = -i * 0.0228
            wsOutput.Cells(zeile, 1).Value = POS
            wsOutput.Cells(zeile, 2).Value = "TG Würth Assy 4
6x150mm"

            wsOutput.Cells(zeile, 3).Value = x
            wsOutput.Cells(zeile, 4).Value = y
            wsOutput.Cells(zeile, 5).Value = -(z + dVR) 'Untere
Schraubenreihe auf Laschenbrett
            wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
            wsOutput.Cells(zeile, 7).Value = ""
            wsOutput.Cells(zeile, 8).Value = "Schraube unten " &
K_SU

            knotennummerAktuell = knotennummerAktuell + 1
            zeile = zeile + 1
        Next i
        POS = POS + 1
        K_SU = K_SU + 1
    End If

Else
    x = x + dF
    If Not KnotenInÖffnung(x, z, Öffnungen, Öffnungszähler)
Then
        For i = 0 To 6
            y = -i * 0.0228
            wsOutput.Cells(zeile, 1).Value = POS
            wsOutput.Cells(zeile, 2).Value = "TG Würth Assy 4
6x150mm"

            wsOutput.Cells(zeile, 3).Value = x
            wsOutput.Cells(zeile, 4).Value = y
            wsOutput.Cells(zeile, 5).Value = -(z + dVR)
            wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
            wsOutput.Cells(zeile, 7).Value = ""
            wsOutput.Cells(zeile, 8).Value = "Schraube unten " &
K_SU

            knotennummerAktuell = knotennummerAktuell + 1
            zeile = zeile + 1
        Next i
        POS = POS + 1
        K_SU = K_SU + 1
    End If
End If
Loop

' 7 Schritt: 1. Nagellage (2. Nagelung laut ETA): Berechnung von x-,y- und
z-Koordinaten der VBM

    Dim K_L1 As Double 'Zählt Nummer für Zuordnung zu VBM
    Dim g As Integer    'Zählt ob Reihe gerade / ungerade (gerade: erster
Nagel bei 3x46,2mm; ungerade dann bei 1x46,2mm)

    z = höheZ1 + dVR 'kann hier beginnen da in 6. Schritt die Reihe bei 8cm
bereits gesetzt wurde
    K_L1 = 1

```

```

g = 1

Do While z < höheZ2 - dV - dVR - 0.0001
    z = z + dV
    x = längeX1 'für jede neue neue z Reihe muss x wieder
zurückgesetzt werden!

    Do While x < längeX2 - dR2

        If x < längeX1 + dR1 Then 'Erster Nagel wird mit 46,2mm
Randabstand gesetzt
            g = g + 1
            If g Mod 2 = 0 Then 'Durch Unterscheidung gerade und
ungerade Reihen kann jede zweite Reihe mit einem anderen Randabstand
beginnen
                x = längeX1 + dR2
            Else
                x = längeX1 + dR1
                If SchraubenVertikal Then GoTo Überspringen1
            End If

            If Not KnotenInÖffnung(x, z, Öffnungen, ÖffnungsZähler)
Then
                For i = 0 To 3
                    y = -i * 0.0228 'y-Koordinate für jeden Schnitt mit
einer Brettlage die ein Nagel macht
                    wsOutput.Cells(zeile, 1).Value = POS
                    wsOutput.Cells(zeile, 2).Value = "LignoLoc 4,7mm/90mm"
                    wsOutput.Cells(zeile, 3).Value = x
                    wsOutput.Cells(zeile, 4).Value = y
                    wsOutput.Cells(zeile, 5).Value = -z
                    wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
                    wsOutput.Cells(zeile, 7).Value = ""
                    wsOutput.Cells(zeile, 8).Value = "L1N" & K_L1

                    knotennummerAktuell = knotennummerAktuell + 1
                    zeile = zeile + 1
                Next i
                POS = POS + 1
                K_L1 = K_L1 + 1
            End If

        Else
            x = x + dF

            If SchraubenVertikal Then

                If g Mod 2 = 0 Then 'Durch Unterscheidung gerade und
ungerade Reihen kann jede zweite Reihe mit einem anderen Randabstand
beginnen
                    If x > längeX2 - dR1 Then
                        x = x + dF
                        GoTo Überspringen1
                    End If
                Else
                    If x > längeX2 - dR1 Then
                        x = x + dF
                        GoTo Überspringen1
                    End If
                End If
            End If
        End If
    End While
End While

```



```

        wsOutput.Cells(zeile, 2).Value = "LignoLoc 4,7mm/90mm"
        wsOutput.Cells(zeile, 3).Value = x
        wsOutput.Cells(zeile, 4).Value = y
        wsOutput.Cells(zeile, 5).Value = -z
        wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
        wsOutput.Cells(zeile, 7).Value = ""
        wsOutput.Cells(zeile, 8).Value = "L2N" & K_L2

        knotennummerAktuell = knotennummerAktuell + 1
        zeile = zeile + 1
    Next i
    POS = POS + 1
    K_L2 = K_L2 + 1
End If
Else
    x = x + dF

    If SchraubenVertikal Then

        If g Mod 2 = 0 Then 'Durch Unterscheidung gerade und
ungerade Reihen kann jede zweite Reihe mit einem anderen Randabstand
beginnen
            If x > längeX2 - dR1 Then
                x = x + dF
                GoTo Überspringen2
            End If
        Else
            If x > längeX2 - dR1 Then
                x = x + dF
                GoTo Überspringen2
            End If
        End If

    End If
    If Not KnotenInÖffnung(x, z, Öffnungen, ÖffnungsZähler)
Then
        For i = 3 To 6
            y = -i * 0.0228 'y-Koordinate für jeden Schnitt
Brettlage / Nagel
            wsOutput.Cells(zeile, 1).Value = POS
            wsOutput.Cells(zeile, 2).Value = "LignoLoc 4,7mm/90mm"
            wsOutput.Cells(zeile, 3).Value = x
            wsOutput.Cells(zeile, 4).Value = y
            wsOutput.Cells(zeile, 5).Value = -z
            wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
            wsOutput.Cells(zeile, 7).Value = ""
            wsOutput.Cells(zeile, 8).Value = "L2N" & K_L2

            knotennummerAktuell = knotennummerAktuell + 1
            zeile = zeile + 1
        Next i
        POS = POS + 1
        K_L2 = K_L2 + 1
    End If
End If
Überspringen2:
    Loop
Loop

g = 0 ' Reset der Variable

```

```

'9 Schritt: Schrauben ab letzter horizontaler Reihe, oberes Laschenbrett
Dim K_SO As Double

x = längeX1
K_SO = 1

Do While x <= längeX2 - dR2
    If x < längeX1 + dR1 Then 'Erste Schraube wird mit 46,2mm
Randabstand gesetzt
        x = längeX1 + dR1

        If Not KnotenInÖffnung(x, z, Öffnungen, ÖffnungsZähler)
Then
            For i = 0 To 6
                y = -i * 0.0228 'y-Koordinate für jeden Schnitt
Brettlage / Nagel
                wsOutput.Cells(zeile, 1).Value = POS
                wsOutput.Cells(zeile, 2).Value = "TG Würth Assy 4
6x150mm"

                wsOutput.Cells(zeile, 3).Value = x
                wsOutput.Cells(zeile, 4).Value = y
                wsOutput.Cells(zeile, 5).Value = -(z + dV) 'nimmt
letztes z und setzt mit Abstand 16cm obere Schraubenreihe
                wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
                wsOutput.Cells(zeile, 7).Value = ""
                wsOutput.Cells(zeile, 8).Value = "Schraube oben " &
K_SO

                knotennummerAktuell = knotennummerAktuell + 1
                zeile = zeile + 1
            Next i
            POS = POS + 1
            K_SO = K_SO + 1
        End If

    Else
        x = x + dF
        If Not KnotenInÖffnung(x, z, Öffnungen, ÖffnungsZähler)
Then
            For i = 0 To 6
                y = -i * 0.0228
                wsOutput.Cells(zeile, 1).Value = POS
                wsOutput.Cells(zeile, 2).Value = "TG Würth Assy 4
6x150mm"

                wsOutput.Cells(zeile, 3).Value = x
                wsOutput.Cells(zeile, 4).Value = y
                wsOutput.Cells(zeile, 5).Value = -(z + dV)
                wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
                wsOutput.Cells(zeile, 7).Value = ""
                wsOutput.Cells(zeile, 8).Value = "Schraube oben " &
K_SO

                knotennummerAktuell = knotennummerAktuell + 1
                zeile = zeile + 1
            Next i
            POS = POS + 1
            K_SO = K_SO + 1
        End If
    End If
End If

```

```

Loop

'10 Schrauben Vertikaler Randbereiche einfügen

If SchraubenVertikal Then

    Dim K_SVL As Double, K_SVR As Double

    K_SVL = 1
    K_SVR = 1

    '10.1 Linker Rand

    x = längeX1 + dR1
    z = höheZ1 + dVR + dV

    Do While z <= höheZ2 - dV
        If Not KnotenInÖffnung(x, z, Öffnungen, ÖffnungsZähler) Then
            For i = 0 To 6
                y = -i * 0.0228
                wsOutput.Cells(zeile, 1).Value = POS
                wsOutput.Cells(zeile, 2).Value = "TG Würth Assy 4 6x150mm"
                wsOutput.Cells(zeile, 3).Value = x
                wsOutput.Cells(zeile, 4).Value = y
                wsOutput.Cells(zeile, 5).Value = -z
                wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
                wsOutput.Cells(zeile, 7).Value = ""
                wsOutput.Cells(zeile, 8).Value = "Schraube vertikal links " &
K_SVL

                knotennummerAktuell = knotennummerAktuell + 1
                zeile = zeile + 1
            Next i
            POS = POS + 1
            K_SVL = K_SVL + 1
        End If
        z = z + dV
    Loop

    'rechter Rand

    x = längeX2 - dR1
    z = höheZ1 + dVR + dV

    Do While z <= höheZ2 - dV
        If Not KnotenInÖffnung(x, z, Öffnungen, ÖffnungsZähler) Then
            For i = 0 To 6
                y = -i * 0.0228
                wsOutput.Cells(zeile, 1).Value = POS
                wsOutput.Cells(zeile, 2).Value = "TG Würth Assy 4 6x150mm"
                wsOutput.Cells(zeile, 3).Value = x
                wsOutput.Cells(zeile, 4).Value = y
                wsOutput.Cells(zeile, 5).Value = -z
                wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
                wsOutput.Cells(zeile, 7).Value = ""
                wsOutput.Cells(zeile, 8).Value = "Schraube vertikal rechts " &
K_SVR

                knotennummerAktuell = knotennummerAktuell + 1
                zeile = zeile + 1
            Next i

```

```

        POS = POS + 1
        K_SVR = K_SVR + 1
    End If
    z = z + dV
Loop
End If

'11 Schritt Liniennummer fortlaufend

    Dim dictLinie As Object: Set dictLinie =
CreateObject("Scripting.Dictionary")
    Dim lastRow As Long: lastRow = wsOutput.Cells(wsOutput.Rows.Count,
1).End(xlUp).Row
    Dim rowIndex As Long

    For rowIndex = 2 To lastRow
        Dim zuordnung As String
        zuordnung = wsOutput.Cells(rowIndex, 8).Value ' H = Zuordnung

        If Not dictLinie.exists(zuordnung) Then
            dictLinie.Add zuordnung, liniennummerAktuell
            liniennummerAktuell = liniennummerAktuell + 1
        End If

        wsOutput.Cells(rowIndex, 7).Value = dictLinie(zuordnung) ' G = LinienNr
    Next rowIndex

Exit Sub
ErrorHandler:
    MsgBox "Ein Fehler ist aufgetreten: " & Err.Description, vbCritical
End Sub

' Hier werden nun die Informationen aus der Eingabeoberfläche zur
Wandgeometrie eingelesen
Function WandGeometrieEinlesen(wsInput As Worksheet, _
                                ByRef längeX1 As Double, ByRef längeX2 As
Double, _
                                ByRef höheZ1 As Double, ByRef höheZ2 As
Double, _
                                ByRef länge As Double, ByRef länge2 As
Double, _
                                ByRef höhe As Double, ByRef höhe2 As
Double, _
                                Optional SchraubenVertikal As Boolean =
False) As Boolean
    WandGeometrieEinlesen = False

    ' X-Richtung prüfen
    If UCase(wsInput.Range("A6").Value) = "X" Then
        If IsNumeric(wsInput.Range("x_1").Value) And
IsNumeric(wsInput.Range("x_2").Value) Then
            längeX1 = wsInput.Range("x_1").Value
            längeX2 = wsInput.Range("x_2").Value
            länge = längeX2 - längeX1
            länge2 = WorksheetFunction.Ceiling_Math(länge, 0.1848)
            längeX2 = längeX1 + länge2
            If SchraubenVertikal And länge2 < 0.923 Then
                MsgBox "FEHLER:" & vbCrLf & _
                    "Schrauben im vertikalen Randbereich nur ab
Wandlänge > 0,924m! " _

```

```

        & vbCrLf & "Aktuelle Wandlänge: " & länge2,
vbCritical
        Exit Function
    End If
Else
    MsgBox "Fehler: Ungültige Werte für x-Koordinaten in B6 oder
C6!", vbCritical
    Exit Function
End If
Else
    MsgBox "Fehler: Zelle A6 sollte x enthalten!", vbCritical
    Exit Function
End If

' Z-Richtung prüfen
If UCase(wsInput.Range("A7").Value) = "Z" Then
    If IsNumeric(wsInput.Range("z_1").Value) And
IsNumeric(wsInput.Range("z_2").Value) Then
        höheZ1 = wsInput.Range("z_1").Value
        höheZ2 = wsInput.Range("z_2").Value
        höhe = höheZ2 - höheZ1
        höhe2 = WorksheetFunction.Ceiling_Math(höhe, 0.16)
        höheZ2 = höheZ1 + höhe2
    Else
        MsgBox "Fehler: Ungültige Werte für z-Koordinaten in B7 oder
C7!", vbCritical
        Exit Function
    End If
Else
    MsgBox "Fehler: Zelle A7 muss z enthalten!", vbCritical
    Exit Function
End If

WandGeometrieEinlesen = True
End Function

```

```

' Hier wird geprüft, ob ein berechneter Knoten innerhalb einer Öffnung
liegt
Function KnotenInÖffnung(x As Double, z As Double, Öffnungen As Variant,
zähler As Long) As Boolean
    Dim j As Long
    For j = 0 To zähler - 1
        Dim öffnung As Variant
        öffnung = Öffnungen(j)
        If x >= öffnung(0) And x <= öffnung(1) And z >= öffnung(2) And z <=
öffnung(3) Then
            KnotenInÖffnung = True
            Exit Function
        End If
    Next j
    KnotenInÖffnung = False
End Function

```

```

Sub Wand_Erstellen(ByRef knotennummerAktuell As Long, ByRef
liniennummerAktuell As Long)
'1. Schritt Basics festlegen
On Error GoTo ErrorHandler

    Dim wsInput As Worksheet, wsOutput As Worksheet
    Dim längeX1 As Double, längeX2 As Double
    Dim höheZ1 As Double, höheZ2 As Double

```

```

Dim länge As Double, höhe As Double
Dim länge2 As Double, höhe2 As Double
Dim zeile As Long
Dim POS As Long

'Querschnittmaße aus ETA
Const dS = 0.0228 'Brettstärke = 22,8mm
Const d184k8 = 0.1848 'Bretter mit 184,8mm Breite
Const d160 = 0.16 'Bretter mit 160mm Breite

'Arbeitsblätter
On Error Resume Next
Set wsInput = ThisWorkbook.Sheets("Input")
Set wsOutput = ThisWorkbook.Sheets("Brettlagen")

'2. Schritt Wandgeometrie einlesen

'Länge
If UCase(wsInput.Range("A6").Value) = "X" Then
    If IsNumeric(wsInput.Range("x_1").Value) And
IsNumeric(wsInput.Range("x_2").Value) Then
        längeX1 = wsInput.Range("x_1").Value
        längeX2 = wsInput.Range("x_2").Value
        länge = längeX2 - längeX1
        länge2 = WorksheetFunction.Ceiling_Math(länge, d184k8)
        längeX2 = längeX1 + länge2

    Else
        MsgBox "Fehler: Ungültige Werte für x-koordinaten in B6 oder
C6!", vbCritical
        Exit Sub
    End If
End If

'Höhe
If UCase(wsInput.Range("A7").Value) = "Z" Then
    If IsNumeric(wsInput.Range("z_1").Value) And
IsNumeric(wsInput.Range("z_2").Value) Then
        höheZ1 = wsInput.Range("z_1").Value
        höheZ2 = wsInput.Range("z_2").Value
        höhe = höheZ2 - höheZ1
        höhe2 = WorksheetFunction.Ceiling_Math(höhe, d160) 'Runden auf
Raster
        höheZ2 = höheZ1 + höhe2 'neue angepasste Höhe
    Else
        MsgBox "Fehler: Ungültige Werte für z-Koordinaten in B7 oder
C7!", vbCritical
        Exit Sub
    End If
End If

'3 Schritt Arbeitsblatt "Brettlagen" für Ausgabe vorbereiten

wsOutput.Cells.Clear
wsOutput.Range("A1:H1").Value = Array("POS", "Lage", "x", "y", "z",
"KnotenNr.", _
"LinienNr.", "Zuordnung")

zeile = 2

```

```

    POS = 1

'4 Schritt Koordinaten der Knoten berechnen

    Dim lagenWinkel As Variant
    Dim lagenDicke As Variant
    Dim lagenNr As Variant
    Dim brettNr As Long

    Dim lage As Integer
    Dim winkel As Double
    Dim brettDicke As Double
    Dim k As Integer

    brettNr = 1

'4.1 Laschenbretter

    Dim dL As Double

    dL = 0.16 'Höhe der Laschenbretter = 160cm

    'Unten
    z = höheZ1 + (dL / 2)
    For k = 0 To 1
        wsOutput.Cells(zeile, 1).Value = POS
        wsOutput.Cells(zeile, 2).Value = "Lasche"
        If k = 0 Then
            wsOutput.Cells(zeile, 3).Value = längeX1
        Else
            wsOutput.Cells(zeile, 3).Value = längeX2
        End If
        wsOutput.Cells(zeile, 4).Value = -1 * dS 'Stabachse liegt
mittig der 5. Lage
        wsOutput.Cells(zeile, 5).Value = -z
        wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
        wsOutput.Cells(zeile, 7).Value = ""
        wsOutput.Cells(zeile, 8).Value = "LascheUnten" & brettNr

        knotennummerAktuell = knotennummerAktuell + 1
        zeile = zeile + 1
        POS = POS + 1
    Next k
    brettNr = brettNr + 1

    'Oben
    z = höheZ2 - (dL / 2)
    For k = 0 To 1
        wsOutput.Cells(zeile, 1).Value = POS
        wsOutput.Cells(zeile, 2).Value = "Lasche"
        If k = 0 Then
            wsOutput.Cells(zeile, 3).Value = längeX1
        Else
            wsOutput.Cells(zeile, 3).Value = längeX2
        End If
        wsOutput.Cells(zeile, 4).Value = -1 * dS 'Stabachse liegt
mittig der 5. Lage
        wsOutput.Cells(zeile, 5).Value = -z
        wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
        wsOutput.Cells(zeile, 7).Value = ""

```

```

        wsOutput.Cells(zeile, 8).Value = "LascheOben" & brettNr

        knotennummerAktuell = knotennummerAktuell + 1
        zeile = zeile + 1
        POS = POS + 1
    Next k
    brettNr = brettNr + 1

'4.2 Brettlagen

    Dim diagonalWinkel As Double
    diagonalWinkel = Atn(0.32 / 0.1848) * 180 / WorksheetFunction.Pi()
'genauer Winkel

    lagenWinkel = Array(90, -diagonalWinkel, 90, diagonalWinkel, 90, 0, 90)
'Ausrichtungen d. Lagen

    lagenDicke = Array(d184k8, d160, d184k8, d160, d184k8, d160, d184k8)
    lagenNr = Array(1, 2, 3, 4, 5, 6, 7)

    For lage = 0 To 3
        winkel = lagenWinkel(lage)
        brettDicke = lagenDicke(lage)
        y = lage * dS - 0.137

        If winkel = 90 Then 'Alle Lagen mit 90° (Lage 1&3)
            x = längeX1
            Do While x <= (längeX2 - 0.0001)
                For k = 0 To 1 'knoten: unten dann oben
                    wsOutput.Cells(zeile, 1).Value = POS
                    wsOutput.Cells(zeile, 2).Value = "Lage" &
lagenNr(lage)

                    wsOutput.Cells(zeile, 3).Value = x + brettDicke / 4
                    wsOutput.Cells(zeile, 4).Value = y
                    If k = 0 Then
                        wsOutput.Cells(zeile, 5).Value = -höheZ1
                    Else
                        wsOutput.Cells(zeile, 5).Value = -höheZ2
                    End If
                    wsOutput.Cells(zeile, 6).Value =
knotennummerAktuell

                    wsOutput.Cells(zeile, 7).Value = ""
                    wsOutput.Cells(zeile, 8).Value = "L" &
lagenNr(lage) & "B" & brettNr

                    knotennummerAktuell = knotennummerAktuell + 1
                    zeile = zeile + 1
                    POS = POS + 1
                Next k
                x = x + brettDicke / 2
                brettNr = brettNr + 1
            Loop

        ElseIf winkel = 0 Then 'Nun die Lage mit 0° Ausrichtung
            z = höheZ1
            Do While z < (höheZ2 - (dL + 0.0001))
                For k = 0 To 1
                    wsOutput.Cells(zeile, 1).Value = POS
                    wsOutput.Cells(zeile, 2).Value = "Lage" & lagenNr(lage)

```

```

        If k = 0 Then
            wsOutput.Cells(zeile, 3).Value = längeX1
        Else
            wsOutput.Cells(zeile, 3).Value = längeX2
        End If
        wsOutput.Cells(zeile, 4).Value = y
        wsOutput.Cells(zeile, 5).Value = -(z + brettDicke / 2)
        wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
        wsOutput.Cells(zeile, 7).Value = ""
        wsOutput.Cells(zeile, 8).Value = "L" & lagenNr(lage) &
"B" & brettNr

        knotennummerAktuell = knotennummerAktuell + 1
        zeile = zeile + 1
        POS = POS + 1
    Next k
    z = z + brettDicke
    brettNr = brettNr + 1
Loop

ElseIf winkel = -diagonalWinkel Or winkel = diagonalWinkel Then

    Dim phiRad As Double: phiRad = winkel * WorksheetFunction.Pi() / 180 '
Winkel in Radiant
    Dim dx As Double, dz As Double ' Richtungs- und Normalenvektor
    dx = Cos(phiRad): dz = Sin(phiRad)

    Dim nx As Double, nz As Double
    nx = -Sin(phiRad): nz = Cos(phiRad)

    Dim projLen As Double
    projLen = Abs(nx) * (längeX2 - längeX1) + Abs(nz) * (höheZ2 -
höheZ1)

    Dim brettZähler As Long
    brettZähler = WorksheetFunction.Ceiling_Math(projLen, brettDicke) /
brettDicke

    ' Durchgang 1: Startpunkte an der Unterkante (x von x1 bis x2)
    Dim i As Long, sx As Double, sy As Double
    For i = -brettZähler To brettZähler
        sx = längeX1 + i * brettDicke * nx 'Schnittpunkte der Diagonalen
mit den Wandkanten
        sy = höheZ1 + i * brettDicke * nz

        ' Schnittpunkte mit Wandkanten ermitteln
        Dim t As Double, xi As Double, zi As Double
        Dim cnt As Integer: cnt = 0
        Dim px(1) As Double, pz(1) As Double

        If dx <> 0 Then
            t = (längeX1 - sx) / dx
            zi = sy + t * dz
            If t <> 0 And zi >= höheZ1 And zi <= höheZ2 Then
                px(cnt) = längeX1
                pz(cnt) = zi
                cnt = cnt + 1
            End If
        End If
    End For

```

```

    If dx <> 0 Then
        t = (längeX2 - sx) / dx
        zi = sy + t * dz
        If t <> 0 And zi >= höheZ1 And zi <= höheZ2 Then
            ' Prüfe, ob der Schnittpunkt das rechte untere Eck ist und
            ignoriere ihn, wenn ja
            If Abs(zi - höheZ1) > 0.001 And Abs(t) > 0.001 Then '
Toleranz von 0.001 m
                px(cnt) = längeX2
                pz(cnt) = zi
                cnt = cnt + 1
            End If
        End If
    End If

    If dz <> 0 Then
        t = (höheZ1 - sy) / dz
        xi = sx + t * dx
        If t <> 0 And xi >= längeX1 And xi <= längeX2 Then
            ' Prüfe, ob der Schnittpunkt das rechte untere Eck ist und
            ignoriere ihn, wenn ja
            If Abs(xi - längeX2) > 0.001 And Abs(t) > 0.001 Then '
Toleranz von 0.001 m
                px(cnt) = xi
                pz(cnt) = höheZ1
                cnt = cnt + 1
            End If
        End If
    End If

    If dz <> 0 Then
        t = (höheZ2 - sy) / dz
        xi = sx + t * dx
        If t <> 0 And xi >= längeX1 And xi <= längeX2 Then
            px(cnt) = xi
            pz(cnt) = höheZ2
            cnt = cnt + 1
        End If
    End If

    ' Wenn genau 2 Schnittpunkte, ablegen
    If cnt = 2 Then
        Dim k2 As Integer
        For k2 = 0 To 1
            px(k2) = SnapToRaster(px(k2), 0.0462) 'Punktfang
            pz(k2) = SnapToRaster(pz(k2), 0.08)
            With wsOutput
                .Cells(zeile, 1).Value = POS
                .Cells(zeile, 2).Value = "Lage" & lagenNr(lage)
                .Cells(zeile, 3).Value = px(k2)
                .Cells(zeile, 4).Value = y
                .Cells(zeile, 5).Value = -pz(k2)
                .Cells(zeile, 6).Value = knotennummerAktuell
                .Cells(zeile, 7).Value = ""
                .Cells(zeile, 8).Value = "L" & lagenNr(lage) & "B" &
brettNr
            End With
            knotennummerAktuell = knotennummerAktuell + 1
            zeile = zeile + 1
        Next k2
    End If

```

```

        POS = POS + 1
        brettNr = brettNr + 1
    End If
Next i

' Manueller Sonderfall für linkes unteres Eck: Diagonale wird manuell
erstellt
If winkel = diagonalWinkel Then
    Dim manX(1) As Double, manZ(1) As Double
    manX(0) = längeX1: manZ(0) = höheZ1          ' Startpunkt unten links
    manX(1) = längeX1 + (höheZ2 - höheZ1) / Tan(phiRad)
    manZ(1) = höheZ2                            ' Endpunkt an oberer Kante

    For k2 = 0 To 1
        With wsOutput
            .Cells(zeile, 1).Value = POS
            .Cells(zeile, 2).Value = "Lage" & lagenNr(lage)
            .Cells(zeile, 3).Value = manX(k2)
            .Cells(zeile, 4).Value = y
            .Cells(zeile, 5).Value = -manZ(k2)
            .Cells(zeile, 6).Value = knotennummerAktuell
            .Cells(zeile, 7).Value = ""
            .Cells(zeile, 8).Value = "L" & lagenNr(lage) & "B" &
brettNr
        End With
        knotennummerAktuell = knotennummerAktuell + 1
        zeile = zeile + 1
    Next k2
    POS = POS + 1
    brettNr = brettNr + 1
End If

'manueller Sonderfall für rechtes unteres Eck (-60° Diagonale):
Diagonale manuell
If winkel = -diagonalWinkel Then
    Dim manX_neg(1) As Double, manZ_neg(1) As Double
    manX_neg(0) = längeX2: manZ_neg(0) = höheZ1          ' Startpunkt
unten rechts
    manX_neg(1) = längeX2 - (höheZ2 - höheZ1) / Tan(Abs(phiRad)) '
Endpunkt oben links
    manZ_neg(1) = höheZ2

    For k2 = 0 To 1
        With wsOutput
            .Cells(zeile, 1).Value = POS
            .Cells(zeile, 2).Value = "Lage" & lagenNr(lage)
            .Cells(zeile, 3).Value = manX_neg(k2)
            .Cells(zeile, 4).Value = y
            .Cells(zeile, 5).Value = -manZ_neg(k2)
            .Cells(zeile, 6).Value = knotennummerAktuell
            .Cells(zeile, 7).Value = ""
            .Cells(zeile, 8).Value = "L" & lagenNr(lage) & "B" &
brettNr
        End With
        knotennummerAktuell = knotennummerAktuell + 1
        zeile = zeile + 1
    Next k2
    POS = POS + 1
    brettNr = brettNr + 1
End If

```

```

End If
Next lage

For lage = 4 To 6
    winkel = lagenWinkel(lage)
    brettDicke = lagenDicke(lage)
    y = lage * dS - 0.137

    If winkel = 90 Then 'Alle Lagen mit 90° (Lage 5 & 7)
        x = längeX1
        Do While x <= (längeX2 - 0.0001)
            For k = 0 To 1 'knoten: unten dann oben
                wsOutput.Cells(zeile, 1).Value = POS
                wsOutput.Cells(zeile, 2).Value = "Lage" & lagenNr(lage)
                wsOutput.Cells(zeile, 3).Value = x + brettDicke / 4
                wsOutput.Cells(zeile, 4).Value = y
                If k = 0 Then
                    wsOutput.Cells(zeile, 5).Value = -(höheZ1 + dL)
                Else
                    wsOutput.Cells(zeile, 5).Value = -(höheZ2 - dL)
                End If
                wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
                wsOutput.Cells(zeile, 7).Value = ""
                wsOutput.Cells(zeile, 8).Value = "L" & lagenNr(lage) &
                "B" & brettNr

                knotennummerAktuell = knotennummerAktuell + 1
                zeile = zeile + 1
                POS = POS + 1
            Next k
            x = x + brettDicke / 2
            brettNr = brettNr + 1

        Loop

    Else 'kein elseif nötig da die lagen 5,6 & 7 entweder 90° oder 0°
    sind
        z = höheZ1 + dL
        Do While z < (höheZ2 - (dL + 0.0001))
            For k = 0 To 1
                wsOutput.Cells(zeile, 1).Value = POS
                wsOutput.Cells(zeile, 2).Value = "Lage" & lagenNr(lage)
                If k = 0 Then
                    wsOutput.Cells(zeile, 3).Value = längeX1
                Else
                    wsOutput.Cells(zeile, 3).Value = längeX2
                End If
                wsOutput.Cells(zeile, 4).Value = y
                wsOutput.Cells(zeile, 5).Value = -(z + brettDicke / 2)
                wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
                wsOutput.Cells(zeile, 7).Value = ""
                wsOutput.Cells(zeile, 8).Value = "L" & lagenNr(lage) & "B"
                & brettNr

                knotennummerAktuell = knotennummerAktuell + 1
                zeile = zeile + 1
                POS = POS + 1
            Next k
            z = z + brettDicke
            brettNr = brettNr + 1
        Loop
    End If
Next

```

```
        Loop
    End If
Next lage

'5. Schritt: Startwerte einselsen

'Einlesen der Konfigurationen
'Modellname
Dim MN As Variant
MN = wsInput.Range("Modellname").Value
wsOutput.Range("J1").Value = "Modellname: "
wsOutput.Range("K1").Value = MN

If IsEmpty(Range("Modellname").Value) Then
    MsgBox "Bitte Modellnamen eingeben"
    Exit Sub
End If

'Linien Startnummer
Dim LN As Long
LN = wsInput.Range("StartLinien").Value
wsOutput.Range("J2").Value = "LinienstartNr.:"
wsOutput.Range("K2").Value = LN

If IsEmpty(Range("StartLinien").Value) Then
    MsgBox "Bitte Startnummern eingeben"
    Exit Sub
End If

wsOutput.Columns.AutoFit

'6. Schritt Liniennummer fortlaufend

Dim dictLinie As Object: Set dictLinie =
CreateObject("Scripting.Dictionary")
Dim lastRow As Long: lastRow = wsOutput.Cells(wsOutput.Rows.Count,
1).End(xlUp).Row
Dim rowIndex As Long

For rowIndex = 2 To lastRow
    Dim zuordnung As String
    zuordnung = wsOutput.Cells(rowIndex, 8).Value ' H = Zuordnung

    If Not dictLinie.exists(zuordnung) Then
        dictLinie.Add zuordnung, liniennummerAktuell
        liniennummerAktuell = liniennummerAktuell + 1
    End If

    wsOutput.Cells(rowIndex, 7).Value = dictLinie(zuordnung) ' G = LinienNr
Next rowIndex

MsgBox "alle Brettlagen wurden erfolgreich berechnet"

Exit Sub

ErrorHandler:
MsgBox "Es ist ein Fehler aufgetreten!"
Exit Sub

End Sub
```

```

Sub ÖffnungenEinlesen(wsInput As Worksheet, ByRef Öffnungen() As Variant, _
    ByRef ÖffnungsZähler As Long)
    Dim letzteSpalte As Long
    Dim spalte As Long
    Dim xVon As Double, xBis As Double
    Dim zVon As Double, zBis As Double

    ÖffnungsZähler = 0
    letzteSpalte = wsInput.Cells(12,
wsInput.Columns.Count).End(xlToLeft).Column

    For spalte = 2 To letzteSpalte - 1
        If Trim(wsInput.Cells(12, spalte).Value) = "Von[m]" Then
            If IsNumeric(wsInput.Cells(13, spalte).Value) _
                And IsNumeric(wsInput.Cells(13, spalte + 1).Value) _
                And IsNumeric(wsInput.Cells(14, spalte).Value) _
                And IsNumeric(wsInput.Cells(14, spalte + 1).Value) Then

                xVon = wsInput.Cells(13, spalte).Value
                xBis = wsInput.Cells(13, spalte + 1).Value
                zVon = wsInput.Cells(14, spalte).Value
                zBis = wsInput.Cells(14, spalte + 1).Value

                ReDim Preserve Öffnungen(0 To ÖffnungsZähler)
                Öffnungen(ÖffnungsZähler) = Array(xVon, xBis, zVon, zBis)
                ÖffnungsZähler = ÖffnungsZähler + 1
            End If
        End If
    Next spalte
End Sub

```

' Funktion zum Snappen auf ein Raster um numerische Ungenauigkeiten zu beheben

```

Function SnapToRaster(ByVal val As Double, ByVal raster As Double) As Double
    SnapToRaster = WorksheetFunction.Round(val / raster, 0) * raster
End Function

```

```

Sub ErstelleExportdatei(berechneBrettlagen As Boolean, berechneNagelbild As Boolean)

```

```

    Dim wsInput As Worksheet
    Set wsInput = ThisWorkbook.Sheets("Input")

    Dim MN As Variant: MN = wsInput.Range("Modellname").Value
    Dim LN_Nagel As Variant: LN_Nagel = wsInput.Range("StartLinien").Value
    Dim LN_Brett As Variant: LN_Brett = wsInput.Range("StartLinien").Value
    If IsEmpty(MN) Then MsgBox "Modellname fehlt!": Exit Sub

```

```

    Dim wsB As Worksheet, wsN As Worksheet, wsS As Worksheet
    Set wsB = ThisWorkbook.Sheets("Brettlagen")
    Set wsN = ThisWorkbook.Sheets("Nagelbild")
    Set wsS = ThisWorkbook.Sheets("Stäbe")

```

```

    ' Wenn nicht berechnet, Dateninhalte löschen (aber Kopfzeile erhalten)
    If Not berechneNagelbild Then

```

```

        If wsN.Cells(Rows.Count, 1).End(xlUp).Row > 1 Then
            wsN.Range("A2", wsN.Cells(wsN.Rows.Count,
1).End(xlUp).EntireRow).ClearContents

```

```

        End If
    End If

    If Not berechneBrettlagen Then
        If wsB.Cells(Rows.Count, 1).End(xlUp).Row > 1 Then
            wsB.Range("A2", wsB.Cells(wsB.Rows.Count,
1).End(xlUp).EntireRow).ClearContents
        End If
    End If

    Dim bHasNagel As Boolean: bHasNagel = (wsN.Cells(wsN.Rows.Count,
1).End(xlUp).Row > 1)
    Dim bHasBrett As Boolean: bHasBrett = (wsB.Cells(wsB.Rows.Count,
1).End(xlUp).Row > 1)

    Dim exportWb As Workbook
    Set exportWb = Workbooks.Add

    ' === Nagelbild ===
    Dim wsNagelExport As Worksheet
    Set wsNagelExport = exportWb.Sheets(1)
    wsNagelExport.Name = "Nagelbild"

    wsNagelExport.Range("A1:H1").Value = Array("POS", "VBM", "x", "y", "z",
"KnotenNr", _
                                                "LinienNr.", "Zuordnung")

    wsNagelExport.Cells(1, 10).Value = "Modellname:"
    wsNagelExport.Cells(1, 11).Value = MN
    wsNagelExport.Cells(2, 10).Value = "LinienstartNr.:"
    wsNagelExport.Cells(2, 11).Value = LN_Nagel

    If bHasNagel Then
        Dim nLastRow As Long
        nLastRow = wsN.Cells(Rows.Count, 1).End(xlUp).Row
        wsN.Range("A2:H" & nLastRow).Copy
        Destination:=wsNagelExport.Range("A2")
    End If

    ' === Brettlagen ===
    Dim wsBrettExport As Worksheet
    Set wsBrettExport = exportWb.Sheets.Add(After:=wsNagelExport)
    wsBrettExport.Name = "Brettlagen"

    wsBrettExport.Range("A1:H1").Value = Array("POS", "Lage", "x", "y",
"z", "KnotenNr", _
                                                "LinienNr.", "Zuordnung")

    wsBrettExport.Cells(1, 10).Value = "Modellname:"
    wsBrettExport.Cells(1, 11).Value = MN
    wsBrettExport.Cells(2, 10).Value = "LinienstartNr.:"
    wsBrettExport.Cells(2, 11).Value = LN_Brett

    If bHasBrett Then
        Dim bLastRow As Long
        bLastRow = wsB.Cells(Rows.Count, 1).End(xlUp).Row
        wsB.Range("A2:H" & bLastRow).Copy
        Destination:=wsBrettExport.Range("A2")
    End If

    ' === Blatt "Stäbe" ===
    Dim wsStabExport As Worksheet

```

```

Set wsStabExport = exportWb.Sheets.Add(After:=wsBrettExport)
wsStabExport.Name = "Stäbe"
wsStabExport.Range("A1:G1").Value = Array("Linie Nr.", "Stabtyp",
"Querschnittsverteilung", _
                                "Drehung[°]/Knoten Nr", _
                                "Anfang i", "Ende j", "Innen k")

Dim sLastRow As Long
sLastRow = wsS.Cells(Rows.Count, 1).End(xlUp).Row
wsS.Range("A2:G" & sLastRow).Copy Destination:=wsStabExport.Range("A2")

' === Datei speichern ===
Dim zielPfad As String
zielPfad = ThisWorkbook.Path & "\Export_Gesamt.xlsx"

Application.DisplayAlerts = False
exportWb.SaveAs Filename:=zielPfad, FileFormat:=xlOpenXMLWorkbook
Application.DisplayAlerts = True
exportWb.Close SaveChanges:=False

MsgBox "Exportdatei 'Export_Gesamt.xlsx' wurde erstellt."
End Sub

```

```

Sub StabZuteilung()

'1 Schritt Blatt vorbereiten

Dim wsN As Worksheet, wsB As Worksheet, wsS As Worksheet
Set wsN = ThisWorkbook.Sheets("Nagelbild")
Set wsB = ThisWorkbook.Sheets("Brettlagen")
Set wsS = ThisWorkbook.Sheets("Stäbe")

wsS.Cells.ClearContents
wsS.Range("A1:G1").Value = Array("Linie Nr.", "Stabtyp",
"Querschnittsverteilung", _
                                "Drehung [°] / Knoten Nr.", "Anfang i",
"Ende j", "Innen k")
Dim dictZuordnung As Object
Set dictZuordnung = CreateObject("Scripting.Dictionary")

'2 Schritt alle Liniennummern und zugehörige Zuordnungen sammeln

Dim rowIndex As Long, lastRowN As Long, lastRowB As Long
Dim linie As Variant, zuordnung As String

lastRowN = wsN.Cells(wsN.Rows.Count, 1).End(xlUp).Row
For rowIndex = 2 To lastRowN
    linie = wsN.Cells(rowIndex, 7).Value
    zuordnung = wsN.Cells(rowIndex, 8).Value
    If Not dictZuordnung.exists(linie) Then dictZuordnung.Add linie,
zuordnung
Next rowIndex

lastRowB = wsB.Cells(wsB.Rows.Count, 1).End(xlUp).Row
For rowIndex = 2 To lastRowB
    linie = wsB.Cells(rowIndex, 7).Value
    zuordnung = wsB.Cells(rowIndex, 8).Value
    If Not dictZuordnung.exists(linie) Then dictZuordnung.Add linie,
zuordnung
Next rowIndex

```

```
'3 Schritt Querschnittsnummern zuweisen

Dim querschnitt As Integer
Dim outputRow As Long: outputRow = 2

For Each linie In dictZuordnung.Keys
    zuordnung = dictZuordnung(linie)

    Select Case True
        Case zuordnung Like "Lasche*"
            querschnitt = 5
        Case zuordnung Like "L1B*" Or zuordnung Like "L3B*" Or _
            zuordnung Like "L5B*" Or zuordnung Like "L7B*"
            querschnitt = 4
        Case zuordnung Like "L2B*" Or zuordnung Like "L4B*" Or
            zuordnung Like "L6B*"
            querschnitt = 3
        Case zuordnung Like "Schraube*"
            querschnitt = 1
        Case zuordnung Like "L1N*" Or zuordnung Like "L2N*"
            querschnitt = 2
        Case Else
            querschnitt = "" ' falls nicht erkennbar
    End Select

    wsS.Cells(outputRow, 1).Value = linie
    wsS.Cells(outputRow, 2).Value = "Balkenstab"
    wsS.Cells(outputRow, 3).Value = "Gleichmäßig"
    wsS.Cells(outputRow, 4).Value = "0.00"
    wsS.Cells(outputRow, 5).Value = querschnitt
    wsS.Cells(outputRow, 6).Value = querschnitt
    wsS.Cells(outputRow, 7).Value = querschnitt

    outputRow = outputRow + 1
Next linie

wsS.Columns.AutoFit

MsgBox "Stäbe wurden erfolgreich erstellt."

End Sub
```

```
Sub Kontrollkästchen7_Klicken()
    Dim ws As Worksheet

    Set ws = ThisWorkbook.Sheets("Input")

    If ws.Range("AbfrageVertikal").Value = True Then
        MsgBox ("Hinweis: Schrauben im vertikalen Randbereich nur ab
        Wandlänge > 0,924m")
    End If

End Sub
```

Anhang V. CODE IDEALMODELL

```

Sub Gesamtberechnung()      ' Hauptsub zur zentralen Steuerung

    Dim wsInput As Worksheet: Set wsInput = ThisWorkbook.Sheets("Input")

    Dim berechneBrettlagen As Boolean
    Dim berechneNagelbild As Boolean
    Dim startKnoten As Long
    Dim startLinien As Long

    berechneBrettlagen = CBool(wsInput.Range("Brettlagen").Value)
    berechneNagelbild = CBool(wsInput.Range("Nagelbild").Value)

    If Not berechneBrettlagen And Not berechneNagelbild Then
        MsgBox "Bitte mindestens eine Option unter 'Zu berechnen'
auswählen"
        Exit Sub
    End If

    ' zentrale Startnummern einlesen
    If Not IsNumeric(wsInput.Range("StartKnoten").Value) Or Not _
IsNumeric(wsInput.Range("StartLinien").Value) Then
        MsgBox "Ungültige Startwerte!", vbCritical
        Exit Sub
    End If
    startKnoten = CLng(wsInput.Range("StartKnoten").Value)
    startLinien = CLng(wsInput.Range("StartLinien").Value)

    ' Weitergabe als Referenz
    If berechneNagelbild Then Call Nagelbild_Erstellung(startKnoten,
startLinien)
    If berechneBrettlagen Then Call Wand_Erstellen(startKnoten,
startLinien)

    Call StabZuteilung
    Call ErstelleExportdatei(berechneBrettlagen, berechneNagelbild)

End Sub

```

```

Sub Nagelbild_Erstellung(ByRef knotennummerAktuell As Long, ByRef
liniennummerAktuell As Long)

    On Error GoTo ErrorHandler 'Regelt Umgang mit Laufzeitfehlern

    '1 Schritt: Basics festlegen

    Dim wsInput As Worksheet, wsOutput As Worksheet 'Vorhandensein d.
Arbeitsblätter kontrollieren
    Dim längeX1 As Double, längeX2 As Double 'Hier Wandlänge: von längeX1
bis längeX2
    Dim höheZ1 As Double, höheZ2 As Double 'Hier Wandhöhe: von höheZ1 bis
höhez2
    Dim länge As Double, höhe As Double 'Hier kommen Wandabmessungen rein
    Dim länge2 As Double, höhe2 As Double 'Wandlängen nun Vielfache des
Rasters
    Dim zeile As Long
    Dim SchraubenVertikal As Boolean

    Set wsInput = ThisWorkbook.Sheets("Input")

```

```

Set wsOutput = ThisWorkbook.Sheets("Nagelbild")
If wsInput Is Nothing Or wsOutput Is Nothing Then
    MsgBox "Fehler: Arbeitsblatt *Input* oder *Nagelblatt* nicht
gefunden!", vbCritical
    Exit Sub 'Da ohne die Arbeitsblätter nicht weiter gearbeitet
werden kann wird hier dann beendet
End If

SchraubenVertikal = CBool(wsInput.Range("AbfrageVertikal").Value)

'2 Schritt: Wandgeometrie einlesen und auf Raster Runden

If Not WandGeometrieEinlesen(wsInput, längeX1, längeX2, höheZ1, höheZ2,
länge, länge2, _
höhe, höhe2, SchraubenVertikal) Then Exit Sub

'3.1 Öffnungen einlesen

Dim Öffnungen() As Variant
Dim ÖffnungsZähler As Long
Call ÖffnungenEinlesen(wsInput, Öffnungen, ÖffnungsZähler)

' 4 Schritt: Arbeitsblatt für das Nagelbild vorbereiten

wsOutput.Cells.Clear
wsOutput.Range("A1:H1").Value = Array("POS", "VBM", "x", "y", "z",
"KnotenNr.", _
"LinienNr.", "Zuordnung") 'Kopfzeile der Ausgabe
zeile = 2 ' Dateninhalt soll dann ab 2. Zeile beginnen

' 5 Schritt: Abstände aus ETA-24-0946 als Konstanten festlegen

'5.1 Horizontale Abstände

Const dF As Double = 0.1848 'Abstand horizontal

'5.2 Vertikale Abstände

Const dV As Double = 0.16 'vertikaler Abstand innerhalb einer Nagellage
= 0.16

'6 Schritt: Beginnen mit unterer horizontaler Schraubenreihe (auf
Laschenbrett)
Dim K_SU As Double
Dim x As Double, y As Double, z As Double
Dim POS As Long
Dim i As Integer 'setzt bei jeden Lagenschnitt eines VBMs ein
Knotenpunkt

POS = 1
x = längeX1
z = höheZ1
K_SU = 1

'Erste Schraube links unten muss manuell gesetzt werden
For i = 0 To 6
    y = -i * 0.0228
    wsOutput.Cells(zeile, 1).Value = POS
    wsOutput.Cells(zeile, 2).Value = "TG Würth Assy 4 6x150mm"

```

```

        wsOutput.Cells(zeile, 3).Value = x
        wsOutput.Cells(zeile, 4).Value = y
        wsOutput.Cells(zeile, 5).Value = -z
        wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
        wsOutput.Cells(zeile, 7).Value = ""
        wsOutput.Cells(zeile, 8).Value = "Schraube unten " & K_SU

        knotennummerAktuell = knotennummerAktuell + 1
        zeile = zeile + 1
    Next i
    POS = POS + 1
    K_SU = K_SU + 1
    x = x + dF

    Do While x <= längeX2 + 0.001
        If Not KnotenInÖffnung(x, z, Öffnungen, ÖffnungsZähler) Then
            For i = 0 To 6
                y = -i * 0.0228
                wsOutput.Cells(zeile, 1).Value = POS
                wsOutput.Cells(zeile, 2).Value = "TG Würth Assy 4

6x150mm"

                wsOutput.Cells(zeile, 3).Value = x
                wsOutput.Cells(zeile, 4).Value = y
                wsOutput.Cells(zeile, 5).Value = -z
                wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
                wsOutput.Cells(zeile, 7).Value = ""
                wsOutput.Cells(zeile, 8).Value = "Schraube unten " &

K_SU

                knotennummerAktuell = knotennummerAktuell + 1
                zeile = zeile + 1
            Next i
            POS = POS + 1
            K_SU = K_SU + 1
        End If
        x = x + dF
    Loop

' 7 Schritt: Holznägel in jedem Kreuzungspunkt

Dim K_L1 As Double 'Zählt Nummer für Zuordnung zu VBM
Dim g As Integer 'Zählt ob Reihe gerade / ungerade (gerade: Raster dF2,
ungerade: Raster dF)
Dim ende As Double

z = höheZ1
K_L1 = 1
g = 1

If SchraubenVertikal Then
    ende = längeX2 - (dF / 2 - 0.001)
Else
    ende = längeX2 + 0.001
End If

Do While z <= höheZ2 - (dV + 0.001)
    z = z + dV

```

```

        x = längeX1 'für jede neue neue z Reihe muss x wieder
zurückgesetzt werden!

        Do While x < ende

            If SchraubenVertikal Then
                If x < längeX1 + dF / 2 Then
                    GoTo Überspringen1
                End If
            End If

            If Not KnotenInÖffnung(x, z, Öffnungen, Öffnungszähler)
Then
                For i = 0 To 6
                    y = -i * 0.0228 'y-Koordinate für jeden Schnitt
mit einer Brettlage die ein Nagel macht
                    wsOutput.Cells(zeile, 1).Value = POS
                    wsOutput.Cells(zeile, 2).Value = "LignoLoc
4,7mm/90mm"

                    wsOutput.Cells(zeile, 3).Value = x
                    wsOutput.Cells(zeile, 4).Value = y
                    wsOutput.Cells(zeile, 5).Value = -z
                    wsOutput.Cells(zeile, 6).Value =

knotennummerAktuell

                    wsOutput.Cells(zeile, 7).Value = ""
                    wsOutput.Cells(zeile, 8).Value = "L1N" & K_L1

                    knotennummerAktuell = knotennummerAktuell + 1
                    zeile = zeile + 1
                Next i 'mit jedem i wird einer der 7 Knoten eines jeden
Nagels gesetzt
                POS = POS + 1
                K_L1 = K_L1 + 1
            End If
        Überspringen1:
            If g Mod 2 = 0 Then
                x = x + dF
            Else
                x = x + dF / 2
            End If

        Loop
        g = g + 1
    Loop

    g = 0 ' Reset der Variable

'8 Schritt: Schrauben ab letzter horizontaler Reihe, oberes Laschenbrett
    Dim K_SO As Double

    x = längeX1
    K_SO = 1

    Do While x <= längeX2 + 0.0001
        If Not KnotenInÖffnung(x, z, Öffnungen, Öffnungszähler) Then
            For i = 0 To 6
                y = -i * 0.0228 'y-Koordinate für jeden Schnitt Brettlage /
Nagel

                wsOutput.Cells(zeile, 1).Value = POS
                wsOutput.Cells(zeile, 2).Value = "TG Würth Assy 4 6x150mm"
                wsOutput.Cells(zeile, 3).Value = x

```

```

        wsOutput.Cells(zeile, 4).Value = y
        wsOutput.Cells(zeile, 5).Value = -(z + dV) 'nimmt letztes z
und setzt mit Abstand 16cm obere Schraubenreihe
        wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
        wsOutput.Cells(zeile, 7).Value = ""
        wsOutput.Cells(zeile, 8).Value = "Schraube oben " & K_SO

        knotennummerAktuell = knotennummerAktuell + 1
        zeile = zeile + 1
    Next i
    POS = POS + 1
    K_SO = K_SO + 1
End If
x = x + dF
Loop

'9 Schrauben Vertikaler Randbereiche einfügen
If SchraubenVertikal Then

    Dim K_SVL As Double, K_SVR As Double

    K_SVL = 1
    K_SVR = 1

    '9.1 Linker Rand

    x = längeX1 + dR1
    z = höheZ1 + dVR + dV

    Do While z <= höheZ2 - dV
        If Not KnotenInÖffnung(x, z, Öffnungen, ÖffnungsZähler) Then
            For i = 0 To 6
                y = -i * 0.0228
                wsOutput.Cells(zeile, 1).Value = POS
                wsOutput.Cells(zeile, 2).Value = "TG Würth Assy 4 6x150mm"
                wsOutput.Cells(zeile, 3).Value = x
                wsOutput.Cells(zeile, 4).Value = y
                wsOutput.Cells(zeile, 5).Value = -z
                wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
                wsOutput.Cells(zeile, 7).Value = ""
                wsOutput.Cells(zeile, 8).Value = "Schraube vertikal links " &
K_SVL

                knotennummerAktuell = knotennummerAktuell + 1
                zeile = zeile + 1
            Next i
            POS = POS + 1
            K_SVL = K_SVL + 1
        End If
        z = z + dV
    Loop

    'rechter Rand

    x = längeX2 - dR1
    z = höheZ1 + dVR + dV

    Do While z <= höheZ2 - dV
        If Not KnotenInÖffnung(x, z, Öffnungen, ÖffnungsZähler) Then

```

```

        For i = 0 To 6
            y = -i * 0.0228
            wsOutput.Cells(zeile, 1).Value = POS
            wsOutput.Cells(zeile, 2).Value = "TG Würth Assy 4 6x150mm"
            wsOutput.Cells(zeile, 3).Value = x
            wsOutput.Cells(zeile, 4).Value = y
            wsOutput.Cells(zeile, 5).Value = -z
            wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
            wsOutput.Cells(zeile, 7).Value = ""
            wsOutput.Cells(zeile, 8).Value = "Schraube vertikal rechts " &
K_SVR

            knotennummerAktuell = knotennummerAktuell + 1
            zeile = zeile + 1
        Next i
        POS = POS + 1
        K_SVR = K_SVR + 1
    End If
    z = z + dV
Loop
End If

'10 Schritt Liniennummer fortlaufend

Dim dictLinie As Object: Set dictLinie =
CreateObject("Scripting.Dictionary")
Dim lastRow As Long: lastRow = wsOutput.Cells(wsOutput.Rows.Count,
1).End(xlUp).Row
Dim rowIndex As Long

For rowIndex = 2 To lastRow
    Dim zuordnung As String
    zuordnung = wsOutput.Cells(rowIndex, 8).Value ' H = Zuordnung

    If Not dictLinie.exists(zuordnung) Then
        dictLinie.Add zuordnung, liniennummerAktuell
        liniennummerAktuell = liniennummerAktuell + 1
    End If

    wsOutput.Cells(rowIndex, 7).Value = dictLinie(zuordnung) ' G = LinienNr
Next rowIndex

Exit Sub
ErrorHandler:
    MsgBox "Ein Fehler ist aufgetreten: " & Err.Description, vbCritical
End Sub

' Hier werden nun die Informationen aus der Eingabeoberfläche zur
Wandgeometrie eingelesen
Function WandGeometrieEinlesen(wsInput As Worksheet, _
                                ByRef längeX1 As Double, ByRef längeX2 As
Double, _
                                ByRef höheZ1 As Double, ByRef höheZ2 As
Double, _
                                ByRef länge As Double, ByRef länge2 As
Double, _
                                ByRef höhe As Double, ByRef höhe2 As
Double, _
                                Optional SchraubenVertikal As Boolean =
False) As Boolean
    WandGeometrieEinlesen = False

```

```

' X-Richtung prüfen
If UCase(wsInput.Range("A6").Value) = "X" Then
    If IsNumeric(wsInput.Range("x_1").Value) And
IsNumeric(wsInput.Range("x_2").Value) Then
        längeX1 = wsInput.Range("x_1").Value
        längeX2 = wsInput.Range("x_2").Value
        länge = längeX2 - längeX1
        länge2 = WorksheetFunction.Ceiling_Math(länge, 0.1848)
        längeX2 = längeX1 + länge2
        If SchraubenVertikal And länge2 < 0.923 Then
            MsgBox "FEHLER:" & vbCrLf & _
                "Schrauben im vertikalen Randbereich nur ab
Wandlänge > 0,924m! " & _
                & vbCrLf & "Aktuelle Wandlänge: " & länge2,
vbCritical
            Exit Function
        End If
    Else
        MsgBox "Fehler: Ungültige Werte für x-Koordinaten in B6 oder
C6!", vbCritical
        Exit Function
    End If
Else
    MsgBox "Fehler: Zelle A6 sollte x enthalten!", vbCritical
    Exit Function
End If

' Z-Richtung prüfen
If UCase(wsInput.Range("A7").Value) = "Z" Then
    If IsNumeric(wsInput.Range("z_1").Value) And
IsNumeric(wsInput.Range("z_2").Value) Then
        höheZ1 = wsInput.Range("z_1").Value
        höheZ2 = wsInput.Range("z_2").Value
        höhe = höheZ2 - höheZ1
        höhe2 = WorksheetFunction.Ceiling_Math(höhe, 0.16)
        höheZ2 = höheZ1 + höhe2
    Else
        MsgBox "Fehler: Ungültige Werte für z-Koordinaten in B7 oder
C7!", vbCritical
        Exit Function
    End If
Else
    MsgBox "Fehler: Zelle A7 muss z enthalten!", vbCritical
    Exit Function
End If

WandGeometrieEinlesen = True
End Function

' Hier wird geprüft ob ein berechneter Knoten innerhalb einer Öffnung liegt
Function KnotenInÖffnung(x As Double, z As Double, Öffnungen As Variant,
zähler As Long) As Boolean
    Dim j As Long
    For j = 0 To zähler - 1
        Dim öffnung As Variant
        öffnung = Öffnungen(j)
        If x >= öffnung(0) And x <= öffnung(1) And z >= öffnung(2) And z <=
öffnung(3) Then
            KnotenInÖffnung = True
            Exit Function
        End If
    Next j
End Function

```

```

        End If
    Next j
    KnotenInÖffnung = False
End Function

Sub Wand_Erstellen(ByRef knotennummerAktuell As Long, ByRef
liniennummerAktuell As Long)

'1. Schritt Basics festlegen
On Error GoTo ErrorHandler

    Dim wsInput As Worksheet, wsOutput As Worksheet
    Dim längeX1 As Double, längeX2 As Double
    Dim höheZ1 As Double, höheZ2 As Double
    Dim länge As Double, höhe As Double
    Dim länge2 As Double, höhe2 As Double
    Dim zeile As Long
    Dim POS As Long

    'Querschnittmaße aus ETA
    Const dS = 0.0228 'Brettstärke = 22,8mm
    Const d184k8 = 0.1848 'Bretter mit 184,8mm Breite
    Const d160 = 0.16 'Bretter mit 160mm Breite

    'Arbeitsblätter
    On Error Resume Next
    Set wsInput = ThisWorkbook.Sheets("Input")
    Set wsOutput = ThisWorkbook.Sheets("Brettlagen")

'2. Schritt Wandgeometrie einlesen

    'Länge
    If UCase(wsInput.Range("A6").Value) = "X" Then
        If IsNumeric(wsInput.Range("x_1").Value) And
IsNumeric(wsInput.Range("x_2").Value) Then
            längeX1 = wsInput.Range("x_1").Value
            längeX2 = wsInput.Range("x_2").Value
            länge = längeX2 - längeX1
            länge2 = WorksheetFunction.Ceiling_Math(länge, d184k8)
'Aufrunden der Wandlänge auf Raster von 184,8mm
            längeX2 = längeX1 + länge2 'Neue X2 Koordinate liegt nun im
Raster von 184,8mm

        Else
            MsgBox "Fehler: Ungültige Werte für x-koordinaten in B6 oder
C6!", vbCritical
            Exit Sub
        End If
    End If

    'Höhe
    If UCase(wsInput.Range("A7").Value) = "Z" Then
        If IsNumeric(wsInput.Range("z_1").Value) And
IsNumeric(wsInput.Range("z_2").Value) Then
            höheZ1 = wsInput.Range("z_1").Value
            höheZ2 = wsInput.Range("z_2").Value
            höhe = höheZ2 - höheZ1
            höhe2 = WorksheetFunction.Ceiling_Math(höhe, d160) 'Runden auf
Raster von 160mm
            höheZ2 = höheZ1 + höhe2 'Neue angepasste Höhe

```

```

        Else
            MsgBox "Fehler: Ungültige Werte für z-Koordinaten in B7 oder
C7!", vbCritical
            Exit Sub
        End If
    End If

'3. Schritt Arbeitsblatt "Brettlagen" für Ausgabe vorbereiten

    wsOutput.Cells.Clear
    wsOutput.Range("A1:H1").Value = Array("POS", "Lage", "x", "y", "z",
"KnotenNr.", "LinienNr.", "Zuordnung")

    zeile = 2
    POS = 1

'4. Schritt Koordinaten der Knoten berechnen

    Dim lagenWinkel As Variant
    Dim lagenDicke As Variant
    Dim lagenNr As Variant
    Dim brettNr As Long

    Dim lage As Integer
    Dim winkel As Double
    Dim brettDicke As Double
    Dim k As Integer

    brettNr = 1

'4.1 Laschenbretter

    Dim dL As Double

    dL = 0.16 'Höhe der Laschenbretter = 160cm

    'Unten
    z = höheZ1
    For k = 0 To 1
        wsOutput.Cells(zeile, 1).Value = POS
        wsOutput.Cells(zeile, 2).Value = "Lasche"
        If k = 0 Then
            wsOutput.Cells(zeile, 3).Value = längeX1
        Else
            wsOutput.Cells(zeile, 3).Value = längeX2
        End If
        wsOutput.Cells(zeile, 4).Value = -1 * dS 'Stabachse liegt
mittig der 5. Lage
        wsOutput.Cells(zeile, 5).Value = -z
        wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
        wsOutput.Cells(zeile, 7).Value = ""
        wsOutput.Cells(zeile, 8).Value = "LascheUnten" & brettNr

        knotennummerAktuell = knotennummerAktuell + 1
        zeile = zeile + 1
        POS = POS + 1
    Next k

```

```

        brettNr = brettNr + 1
    'Oben
    z = höheZ2
    For k = 0 To 1
        wsOutput.Cells(zeile, 1).Value = POS
        wsOutput.Cells(zeile, 2).Value = "Lasche"
        If k = 0 Then
            wsOutput.Cells(zeile, 3).Value = längeX1
        Else
            wsOutput.Cells(zeile, 3).Value = längeX2
        End If
        wsOutput.Cells(zeile, 4).Value = -1 * dS 'Stabachse liegt
mittig der 5. Lage
        wsOutput.Cells(zeile, 5).Value = -z
        wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
        wsOutput.Cells(zeile, 7).Value = ""
        wsOutput.Cells(zeile, 8).Value = "LascheOben" & brettNr

        knotennummerAktuell = knotennummerAktuell + 1
        zeile = zeile + 1
        POS = POS + 1
    Next k
    brettNr = brettNr + 1

'4.2 Brettlagen

    Dim diagonalWinkel As Double
    diagonalWinkel = Atn(0.32 / 0.1848) * 180 / WorksheetFunction.Pi()
'genauer Winkel

    lagenWinkel = Array(90, -diagonalWinkel, 90, diagonalWinkel, 90, 0, 90)
'Ausrichtungen d. Lagen

    lagenDicke = Array(d184k8, d160, d184k8, d160, d184k8, d160, d184k8)
    lagenNr = Array(1, 2, 3, 4, 5, 6, 7)

    For lage = 0 To 3
        winkel = lagenWinkel(lage)
        brettDicke = lagenDicke(lage)
        y = lage * dS - 0.137

        If winkel = 90 Then 'Alle Lagen mit 90° (Lage 1&3)
            x = längeX1
            Do While x <= längeX2 + 0.0001 'Letztes Brett genau auf X2
                For k = 0 To 1 'knoten: unten dann oben
                    wsOutput.Cells(zeile, 1).Value = POS
                    wsOutput.Cells(zeile, 2).Value = "Lage" &
lagenNr(lage)

                    wsOutput.Cells(zeile, 3).Value = x
                    wsOutput.Cells(zeile, 4).Value = y
                    If k = 0 Then
                        wsOutput.Cells(zeile, 5).Value = -höheZ1
                    Else
                        wsOutput.Cells(zeile, 5).Value = -höheZ2
                    End If
                    wsOutput.Cells(zeile, 6).Value =
knotennummerAktuell

                    wsOutput.Cells(zeile, 7).Value = ""
                    wsOutput.Cells(zeile, 8).Value = "L" &
lagenNr(lage) & "B" & brettNr
                Next k
            Loop
        End If
    Next lage

```

```

        knotennummerAktuell = knotennummerAktuell + 1
        zeile = zeile + 1
        POS = POS + 1
    Next k
    x = x + brettDicke
    brettNr = brettNr + 1
Loop

ElseIf winkel = -diagonalWinkel Or winkel = diagonalWinkel Then

    Dim phiRad As Double: phiRad = winkel * WorksheetFunction.Pi() / 180 '
Winkel in Radiant
    Dim dx As Double, dz As Double ' Richtungs- und Normalenvektor
    dx = Cos(phiRad): dz = Sin(phiRad)

    Dim nx As Double, nz As Double
    nx = -Sin(phiRad): nz = Cos(phiRad)

    Dim projLen As Double
    projLen = Abs(nx) * (längeX2 - längeX1) + Abs(nz) * (höheZ2 -
höheZ1)

    Dim brettZähler As Long
    brettZähler = WorksheetFunction.Ceiling_Math(projLen, brettDicke) /
brettDicke

    ' Durchgang 1: Startpunkte an der Unterkante (x von x1 bis x2)
    Dim i As Long, sx As Double, sy As Double
    For i = -brettZähler To brettZähler
        sx = längeX1 + i * brettDicke * nx 'Schnittpunkte der Diagonalen
mit den Wandkanten
        sy = höheZ1 + i * brettDicke * nz

        ' Schnittpunkte mit Wandkanten ermitteln
        Dim t As Double, xi As Double, zi As Double
        Dim cnt As Integer: cnt = 0
        Dim px(1) As Double, pz(1) As Double

        If dx <> 0 Then
            t = (längeX1 - sx) / dx
            zi = sy + t * dz
            If t <> 0 And zi >= höheZ1 And zi <= höheZ2 Then
                px(cnt) = längeX1
                pz(cnt) = zi
                cnt = cnt + 1
            End If
        End If

        If dx <> 0 Then
            t = (längeX2 - sx) / dx
            zi = sy + t * dz
            If t <> 0 And zi >= höheZ1 And zi <= höheZ2 Then
                ' Prüfe, ob der Schnittpunkt das rechte untere Eck ist und
ignoriere ihn, wenn ja
                If Abs(zi - höheZ1) > 0.001 And Abs(t) > 0.001 Then '
Toleranz von 0.001 m
                    px(cnt) = längeX2
                    pz(cnt) = zi
                End If
            End If
        End If
    Next i

```

```

        cnt = cnt + 1
    End If
End If
End If

If dz <> 0 Then
    t = (höheZ1 - sy) / dz
    xi = sx + t * dx
    If t <> 0 And xi >= längeX1 And xi <= längeX2 Then
        ' Prüfe, ob der Schnittpunkt das rechte untere Eck ist und
        ignoriere ihn, wenn ja
        If Abs(xi - längeX2) > 0.001 And Abs(t) > 0.001 Then '
        Toleranz von 0.001 m
            px(cnt) = xi
            pz(cnt) = höheZ1
            cnt = cnt + 1
        End If
    End If
End If

If dz <> 0 Then
    t = (höheZ2 - sy) / dz
    xi = sx + t * dx
    If t <> 0 And xi >= längeX1 And xi <= längeX2 Then
        px(cnt) = xi
        pz(cnt) = höheZ2
        cnt = cnt + 1
    End If
End If

' Wenn genau 2 Schnittpunkte, ablegen
If cnt = 2 Then
    Dim k2 As Integer
    For k2 = 0 To 1
        px(k2) = SnapToRaster(px(k2), 0.0462) 'Punktfang für
        exaktes Aufeinandertreffen
        pz(k2) = SnapToRaster(pz(k2), 0.08)
        With wsOutput
            .Cells(zeile, 1).Value = POS
            .Cells(zeile, 2).Value = "Lage" & lagenNr(lage)
            .Cells(zeile, 3).Value = px(k2)
            .Cells(zeile, 4).Value = y
            .Cells(zeile, 5).Value = -pz(k2)
            .Cells(zeile, 6).Value = knotennummerAktuell
            .Cells(zeile, 7).Value = ""
            .Cells(zeile, 8).Value = "L" & lagenNr(lage) & "B" &
brettNr
        End With
        knotennummerAktuell = knotennummerAktuell + 1
        zeile = zeile + 1
    Next k2
    POS = POS + 1
    brettNr = brettNr + 1
End If
Next i

' Manueller Sonderfall für linkes unteres Eck: Diagonale wird manuell
erstellt
If winkel = diagonalWinkel Then

```

```

Dim manX(1) As Double, manZ(1) As Double
manX(0) = längeX1: manZ(0) = höheZ1 ' Startpunkt unten links
manX(1) = längeX1 + (höheZ2 - höheZ1) / Tan(phiRad)
manZ(1) = höheZ2 ' Endpunkt an oberer Kante

For k2 = 0 To 1
    With wsOutput
        .Cells(zeile, 1).Value = POS
        .Cells(zeile, 2).Value = "Lage" & lagenNr(lage)
        .Cells(zeile, 3).Value = manX(k2)
        .Cells(zeile, 4).Value = y
        .Cells(zeile, 5).Value = -manZ(k2)
        .Cells(zeile, 6).Value = knotennummerAktuell
        .Cells(zeile, 7).Value = ""
        .Cells(zeile, 8).Value = "L" & lagenNr(lage) & "B" &
brettNr

    End With
    knotennummerAktuell = knotennummerAktuell + 1
    zeile = zeile + 1
Next k2
POS = POS + 1
brettNr = brettNr + 1
End If

'manueller Sonderfall für rechtes unteres Eck (-60° Diagonale):
Diagonale manuell
If winkel = -diagonalWinkel Then
    Dim manX_neg(1) As Double, manZ_neg(1) As Double
    manX_neg(0) = längeX2: manZ_neg(0) = höheZ1 ' Startpunkt
unten rechts
    manX_neg(1) = längeX2 - (höheZ2 - höheZ1) / Tan(Abs(phiRad)) '
Endpunkt oben links
    manZ_neg(1) = höheZ2

    For k2 = 0 To 1
        With wsOutput
            .Cells(zeile, 1).Value = POS
            .Cells(zeile, 2).Value = "Lage" & lagenNr(lage)
            .Cells(zeile, 3).Value = manX_neg(k2)
            .Cells(zeile, 4).Value = y
            .Cells(zeile, 5).Value = -manZ_neg(k2)
            .Cells(zeile, 6).Value = knotennummerAktuell
            .Cells(zeile, 7).Value = ""
            .Cells(zeile, 8).Value = "L" & lagenNr(lage) & "B" &
brettNr

        End With
        knotennummerAktuell = knotennummerAktuell + 1
        zeile = zeile + 1
    Next k2
    POS = POS + 1
    brettNr = brettNr + 1
End If

End If
Next lage

For lage = 4 To 6
    winkel = lagenWinkel(lage)
    brettDicke = lagenDicke(lage)

```

```

y = lage * dS - 0.137

If winkel = 90 Then 'Alle Lagen mit 90° (Lage 5 & 7)
    x = längeX1
    Do While x <= längeX2 + 0.0001
        For k = 0 To 1 'knoten: unten dann oben
            wsOutput.Cells(zeile, 1).Value = POS
            wsOutput.Cells(zeile, 2).Value = "Lage" & lagenNr(lage)
            wsOutput.Cells(zeile, 3).Value = x
            wsOutput.Cells(zeile, 4).Value = y
            If k = 0 Then
                wsOutput.Cells(zeile, 5).Value = -(höheZ1 + dL)
            Else
                wsOutput.Cells(zeile, 5).Value = -(höheZ2 - dL)
            End If
            wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
            wsOutput.Cells(zeile, 7).Value = ""
            wsOutput.Cells(zeile, 8).Value = "L" & lagenNr(lage) &
"B" & brettNr

            knotennummerAktuell = knotennummerAktuell + 1
            zeile = zeile + 1
            POS = POS + 1
        Next k
        x = x + brettDicke
        brettNr = brettNr + 1
    Loop

Else 'kein elseif nötig da die lagen 5,6 & 7 entweder 90° oder 0°
sind
    z = höheZ1 + brettDicke
    Do While z <= (höheZ2 - (dL - 0.0001))
        For k = 0 To 1
            wsOutput.Cells(zeile, 1).Value = POS
            wsOutput.Cells(zeile, 2).Value = "Lage" & lagenNr(lage)
            If k = 0 Then
                wsOutput.Cells(zeile, 3).Value = längeX1
            Else
                wsOutput.Cells(zeile, 3).Value = längeX2
            End If
            wsOutput.Cells(zeile, 4).Value = y
            wsOutput.Cells(zeile, 5).Value = -z
            wsOutput.Cells(zeile, 6).Value = knotennummerAktuell
            wsOutput.Cells(zeile, 7).Value = ""
            wsOutput.Cells(zeile, 8).Value = "L" & lagenNr(lage) & "B"
& brettNr

            knotennummerAktuell = knotennummerAktuell + 1
            zeile = zeile + 1
            POS = POS + 1
        Next k
        z = z + brettDicke
        brettNr = brettNr + 1
    Loop
End If
Next lage

'5. Schritt: Startwerte einlesen

'Einlesen der Konfigurationen
'Modellname

```

```

Dim MN As Variant
MN = wsInput.Range("Modellname").Value
wsOutput.Range("J1").Value = "Modellname: "
wsOutput.Range("K1").Value = MN

If IsEmpty(Range("Modellname").Value) Then
    MsgBox "Bitte Modellnamen eingeben"
    Exit Sub
End If

'Linien Startnummer
Dim LN As Long
LN = wsInput.Range("StartLinien").Value
wsOutput.Range("J2").Value = "LinienstartNr.:"
wsOutput.Range("K2").Value = LN

If IsEmpty(Range("StartLinien").Value) Then
    MsgBox "Bitte Startnummern eingeben"
    Exit Sub
End If

wsOutput.Columns.AutoFit

'6. Schritt Liniennummer fortlaufend

Dim dictLinie As Object: Set dictLinie =
CreateObject("Scripting.Dictionary")
Dim lastRow As Long: lastRow = wsOutput.Cells(wsOutput.Rows.Count,
1).End(xlUp).Row
Dim rowIndex As Long

For rowIndex = 2 To lastRow
    Dim zuordnung As String
    zuordnung = wsOutput.Cells(rowIndex, 8).Value ' H = Zuordnung

    If Not dictLinie.exists(zuordnung) Then
        dictLinie.Add zuordnung, liniennummerAktuell
        liniennummerAktuell = liniennummerAktuell + 1
    End If

    wsOutput.Cells(rowIndex, 7).Value = dictLinie(zuordnung) ' G = LinienNr
Next rowIndex

MsgBox "alle Brettlagen wurden erfolgreich berechnet"

Exit Sub

ErrorHandler:
MsgBox "Es ist ein Fehler aufgetreten!" & Err.Description, vbCritical
Exit Sub

End Sub

Sub ÖffnungenEinlesen(wsInput As Worksheet, ByRef Öffnungen() As Variant, _
ByRef ÖffnungsZähler As Long)
Dim letzteSpalte As Long
Dim spalte As Long
Dim xVon As Double, xBis As Double
Dim zVon As Double, zBis As Double

```

```

    Öffnungszähler = 0
    letzteSpalte = wsInput.Cells(12,
wsInput.Columns.Count).End(xlToLeft).Column

    For spalte = 2 To letzteSpalte - 1
        If Trim(wsInput.Cells(12, spalte).Value) = "Von[m]" Then
            If IsNumeric(wsInput.Cells(13, spalte).Value) _
            And IsNumeric(wsInput.Cells(13, spalte + 1).Value) _
            And IsNumeric(wsInput.Cells(14, spalte).Value) _
            And IsNumeric(wsInput.Cells(14, spalte + 1).Value) Then

                xVon = wsInput.Cells(13, spalte).Value
                xBis = wsInput.Cells(13, spalte + 1).Value
                zVon = wsInput.Cells(14, spalte).Value
                zBis = wsInput.Cells(14, spalte + 1).Value

                ReDim Preserve Öffnungen(0 To Öffnungszähler)
                Öffnungen(Öffnungszähler) = Array(xVon, xBis, zVon, zBis)
                Öffnungszähler = Öffnungszähler + 1
            End If
        End If
    Next spalte
End Sub

```

' Funktion zum Snappen auf ein Raster um numerische Ungenauigkeiten zu beheben

```

Function SnapToRaster(ByVal val As Double, ByVal raster As Double) As Double
    SnapToRaster = WorksheetFunction.Round(val / raster, 0) * raster
End Function

```

```

Sub ErstelleExportdatei(berechneBrettlagen As Boolean, berechneNagelbild As Boolean)

```

```

    Dim wsInput As Worksheet
    Set wsInput = ThisWorkbook.Sheets("Input")

    Dim MN As Variant: MN = wsInput.Range("Modellname").Value
    Dim LN_Nagel As Variant: LN_Nagel = wsInput.Range("StartLinien").Value
    Dim LN_Brett As Variant: LN_Brett = wsInput.Range("StartLinien").Value
    If IsEmpty(MN) Then MsgBox "Modellname fehlt!": Exit Sub

    Dim wsB As Worksheet, wsN As Worksheet, wsS As Worksheet
    Set wsB = ThisWorkbook.Sheets("Brettlagen")
    Set wsN = ThisWorkbook.Sheets("Nagelbild")
    Set wsS = ThisWorkbook.Sheets("Stäbe")

    ' Wenn nicht berechnet, Dateninhalte löschen (aber Kopfzeile erhalten)
    If Not berechneNagelbild Then
        If wsN.Cells(Rows.Count, 1).End(xlUp).Row > 1 Then
            wsN.Range("A2", wsN.Cells(wsN.Rows.Count,
1).End(xlUp).EntireRow).ClearContents
        End If
    End If

    If Not berechneBrettlagen Then
        If wsB.Cells(Rows.Count, 1).End(xlUp).Row > 1 Then
            wsB.Range("A2", wsB.Cells(wsB.Rows.Count,
1).End(xlUp).EntireRow).ClearContents
        End If
    End If

```

```

End If

Dim bHasNagel As Boolean: bHasNagel = (wsN.Cells(wsN.Rows.Count,
1).End(xlUp).Row > 1)
Dim bHasBrett As Boolean: bHasBrett = (wsB.Cells(wsB.Rows.Count,
1).End(xlUp).Row > 1)

Dim exportWb As Workbook
Set exportWb = Workbooks.Add

' === Nagelbild ===
Dim wsNagelExport As Worksheet
Set wsNagelExport = exportWb.Sheets(1)
wsNagelExport.Name = "Nagelbild"

wsNagelExport.Range("A1:H1").Value = Array("POS", "VBM", "x", "y", "z",
"KnotenNr", _
                                "LinienNr.", "Zuordnung")

wsNagelExport.Cells(1, 10).Value = "Modellname:"
wsNagelExport.Cells(1, 11).Value = MN
wsNagelExport.Cells(2, 10).Value = "LinienstartNr.:"
wsNagelExport.Cells(2, 11).Value = LN_Nagel

If bHasNagel Then
    Dim nLastRow As Long
    nLastRow = wsN.Cells(Rows.Count, 1).End(xlUp).Row
    wsN.Range("A2:H" & nLastRow).Copy
    Destination:=wsNagelExport.Range("A2")
End If

' === Brettlagen ===
Dim wsBrettExport As Worksheet
Set wsBrettExport = exportWb.Sheets.Add(After:=wsNagelExport)
wsBrettExport.Name = "Brettlagen"

wsBrettExport.Range("A1:H1").Value = Array("POS", "Lage", "x", "y",
"z", "KnotenNr", _
                                "LinienNr.", "Zuordnung")

wsBrettExport.Cells(1, 10).Value = "Modellname:"
wsBrettExport.Cells(1, 11).Value = MN
wsBrettExport.Cells(2, 10).Value = "LinienstartNr.:"
wsBrettExport.Cells(2, 11).Value = LN_Brett

If bHasBrett Then
    Dim bLastRow As Long
    bLastRow = wsB.Cells(Rows.Count, 1).End(xlUp).Row
    wsB.Range("A2:H" & bLastRow).Copy
    Destination:=wsBrettExport.Range("A2")
End If

' === Blatt "Stäbe" ===
Dim wsStabExport As Worksheet
Set wsStabExport = exportWb.Sheets.Add(After:=wsBrettExport)
wsStabExport.Name = "Stäbe"
wsStabExport.Range("A1:G1").Value = Array("Linie Nr.", "Stabtyp",
"Querschnittsverteilung", _
                                "Drehung[°]/Knoten Nr", _
                                "Anfang i", "Ende j", "Innen k")

Dim sLastRow As Long

```

```

sLastRow = wsS.Cells(Rows.Count, 1).End(xlUp).Row
wsS.Range("A2:G" & sLastRow).Copy Destination:=wsStabExport.Range("A2")

' === Datei speichern ===
Dim zielPfad As String
zielPfad = ThisWorkbook.Path & "\Export_Gesamt.xlsx"

Application.DisplayAlerts = False
exportWb.SaveAs Filename:=zielPfad, FileFormat:=xlOpenXMLWorkbook
Application.DisplayAlerts = True
exportWb.Close SaveChanges:=False

MsgBox "Exportdatei 'Export_Gesamt.xlsx' wurde erstellt."
End Sub

```

```

Sub StabZuteilung()

'1 Schritt Blatt vorbereiten

Dim wsN As Worksheet, wsB As Worksheet, wsS As Worksheet
Set wsN = ThisWorkbook.Sheets("Nagelbild")
Set wsB = ThisWorkbook.Sheets("Brettlagen")
Set wsS = ThisWorkbook.Sheets("Stäbe")

wsS.Cells.ClearContents
wsS.Range("A1:G1").Value = Array("Linie Nr.", "Stabtyp",
"Querschnittsverteilung", _
"Drehung [°] / Knoten Nr.", "Anfang i",
"Ende j", "Innen k")
Dim dictZuordnung As Object
Set dictZuordnung = CreateObject("Scripting.Dictionary")

'2 Schritt alle Liniennummern und zugehörige Zuordnungen sammeln

Dim rowIndex As Long, lastRowN As Long, lastRowB As Long
Dim linie As Variant, zuordnung As String

lastRowN = wsN.Cells(wsN.Rows.Count, 1).End(xlUp).Row
For rowIndex = 2 To lastRowN
    linie = wsN.Cells(rowIndex, 7).Value
    zuordnung = wsN.Cells(rowIndex, 8).Value
    If Not dictZuordnung.exists(linie) Then dictZuordnung.Add linie,
zuordnung
Next rowIndex

lastRowB = wsB.Cells(wsB.Rows.Count, 1).End(xlUp).Row
For rowIndex = 2 To lastRowB
    linie = wsB.Cells(rowIndex, 7).Value
    zuordnung = wsB.Cells(rowIndex, 8).Value
    If Not dictZuordnung.exists(linie) Then dictZuordnung.Add linie,
zuordnung
Next rowIndex

'3 Schritt Querschnittsnummern zuweisen

Dim querschnitt As Integer
Dim outputRow As Long: outputRow = 2

For Each linie In dictZuordnung.Keys
    zuordnung = dictZuordnung(linie)

```

```
Select Case True
    Case zuordnung Like "Lasche*"
        querschnitt = 5
    Case zuordnung Like "L1B*" Or zuordnung Like "L3B*" Or
zuordnung Like "L5B*" Or _
zuordnung Like "L7B*"
        querschnitt = 4
    Case zuordnung Like "L2B*" Or zuordnung Like "L4B*" Or
zuordnung Like "L6B*"
        querschnitt = 3
    Case zuordnung Like "Schraube*"
        querschnitt = 1
    Case zuordnung Like "L1N*" Or zuordnung Like "L2N*"
        querschnitt = 2
    Case Else
        querschnitt = "" ' falls nicht erkennbar
End Select

wsS.Cells(outputRow, 1).Value = linie
wsS.Cells(outputRow, 2).Value = "Balkenstab"
wsS.Cells(outputRow, 3).Value = "Gleichmäßig"
wsS.Cells(outputRow, 4).Value = "0.00"
wsS.Cells(outputRow, 5).Value = querschnitt
wsS.Cells(outputRow, 6).Value = querschnitt
wsS.Cells(outputRow, 7).Value = querschnitt

outputRow = outputRow + 1
Next linie

wsS.Columns.AutoFit

MsgBox "Stäbe wurden erfolgreich erstellt."

End Sub

Sub Kontrollkästchen7_Klicken()
Dim ws As Worksheet

Set ws = ThisWorkbook.Sheets("Input")

If ws.Range("AbfrageVertikal").Value = True Then
    MsgBox ("Hinweis: Schrauben im vertikalen Randbereich nur ab
Wandlänge > 0,924m")
End If

End Sub
```

Anhang VI. CODE PYTHON

```
import os
import glob
import pandas as pd
from openpyxl import load_workbook
from RFEM.initModel import Model
from RFEM.BasicObjects.node import Node
from RFEM.BasicObjects.line import Line

# -----
# Pfad zur neuesten Exportdatei (.xlsx)
# -----
script_dir = os.path.dirname(__file__)
xlsx_files = glob.glob(os.path.join(script_dir, "Export_Gesamt*.xlsx"))

if not xlsx_files:

    raise FileNotFoundError("Keine Datei gefunden mit Muster
'Export_Gesamt*.xlsx' im Skriptordner.")

excel_path = max(xlsx_files, key=os.path.getctime)

# -----
# Modellnamen und Startnummer aus Sheet "Nagelbild" auslesen
# -----
wb = load_workbook(excel_path, data_only=True)
ws = wb["Nagelbild"]
model_name = ws.cell(row=1, column=11).value # K1
start_no = int(ws.cell(row=2, column=11).value) # K2
wb.close()

print(f"Verwende Modellname: {model_name}")
print(f"Startnummer Linien: {start_no}")

# -----
# Verbindung zum RFEM-Modell
# -----
model = Model(False, model_name)
print("Erfolgreich mit dem Modell verbunden.")
model.clientModel.service.begin_modification()

# -----
# Relevante Sheets einlesen (Nagelbild & Brettlagen)
# -----
sheets_to_read = ["Nagelbild", "Brettlagen"]
all_nodes_df = pd.DataFrame()

for sheet in sheets_to_read:
    df = pd.read_excel(excel_path, sheet_name=sheet, engine="openpyxl")
    df.columns = df.columns.str.strip()
    all_nodes_df = pd.concat([all_nodes_df, df], ignore_index=True)

# -----
# Knoten erzeugen und nach Kommentar gruppieren (robust!)
# -----
gruppen = {}

for _, row in all_nodes_df.iterrows():
    try:
        # Knotennummer flexibel holen
```

```

        node_no = row.get("KnotenNr")
        if pd.isna(node_no):
            continue
        node_no = int(node_no)

        x = float(row["x"])
        y = float(row["y"])
        z = float(row["z"])

        # Kommentar aus passender Spalte holen
        kommentar = row.get("Zuordnung") or "KommentarFehlt"
        kommentar = str(kommentar) if not pd.isna(kommentar) else
"KommentarLeer"

        Node(node_no, x, y, z, comment=kommentar)
        print(f"Knoten {node_no} hinzugefügt mit Kommentar: '{kommentar}'")
        gruppen.setdefault(kommentar, []).append(node_no)

    except Exception as e:
        print(f"⚠ Fehler bei Zeile: {row.to_dict()} → {e}")

# -----
# Linien mit gleichem Kommentar erzeugen
# -----
line_no = start_no
# Linien aus den gegebenen Liniennummern erzeugen
liniengruppen = {}

for _, row in all_nodes_df.iterrows():
    try:
        linien_nr = row.get("LinienNr.")
        if pd.isna(linien_nr):
            continue
        linien_nr = int(linien_nr)

        node_no = row.get("KnotenNr")
        if pd.isna(node_no):
            continue
        node_no = int(node_no)

        kommentar = row.get("Zuordnung") or "KommentarFehlt"
        kommentar = str(kommentar) if not pd.isna(kommentar) else
"KommentarLeer"

        liniengruppen.setdefault(linien_nr, {"knoten": [], "kommentar":
kommentar})
        liniengruppen[linien_nr]["knoten"].append(node_no)

    except Exception as e:
        print(f"⚠ Fehler beim Einsortieren der Linie: {row.to_dict()} →
{e}")

# Linien erzeugen mit fixen Nummern
for linie_nr, eintrag in liniengruppen.items():
    knotens = sorted(eintrag["knoten"])
    kommentar = eintrag["kommentar"]

    if len(knotens) < 2:
        print(f"⚠ Linie {linie_nr} übersprungen - weniger als 2 Knoten.")
        continue

```

```
        Line(linie_nr, knotens, comment=komentar)
        print(f"Linie {linie_nr} erstellt für Kommentar '{komentar}' und
verbindet Knoten {knotens}")

# -----
# Abschließen
# -----
model.clientModel.service.finish_modification()
print("✅ Export Knoten und Linien abgeschlossen.")
```

