

# **Software für Berechnungen im Bauingenieurwesen nach der Methode der Finiten Element**

## **FE Berechnungen**

## Ziele und Motivation

### 1. Ausbildung von Studierenden des Bauingenieurwesens

Ein wesentliches Ziel der Ausbildung von Bauingenieuren für den Entwurf und die Ausführung von Bauwerken besteht darin, ein Verständnis des Tragverhaltens von Strukturen im Bauwesen unter externen Einwirkungen zu entwickeln und den Einfluss von Modifikationen und Anpassungen im Tragwerksentwurf abschätzen zu lernen. Zu diesen Zwecken muss eine Idealisierung (ein **Modell**) entwickelt werden, welches die wesentlichen Entwurfsparameter einzelner Bauteile oder des gesamten Bauwerks erfasst und abbildet. Dies sind heute in der Regel digitale Bauteil- oder Bauwerksmodelle, welche mit unterschiedlichen Softwaresystemen erstellt und bearbeitet werden können.

Diese Modelle dienen als Grundlage unterschiedlicher Berechnungsmethoden und sind häufig spezifisch für unterschiedliche Berechnungsverfahren und sogar für unterschiedliche Softwaresysteme. In der Konsequenz gibt es in den meisten Bauprojekten eine Vielzahl unterschiedlicher Modelle für unterschiedliche Zwecke. Dies ist auch heute noch im Wesentlichen der Stand der Technik.

Die relativ neue **BIM**-Technologie steht für „**B**auwerks**I**nformations**M**odell“ oder „**B**uilding**I**nformation**M**odel“ und hat als ein wesentliches Ziel, die Vereinheitlichung und Standardisierung solcher Modelle in ein einheitliches Gesamtmodell oder zumindest in eine Menge kompatibler Teilmodelle.

Komplexe Projekte im Bauwesen werden auch nach heutigem Stand der Technik fast immer entwickelt und ausgeführt auf Basis einer großen Menge Technischer Dokumente - in Großprojekten können dies Tausende von Dokumenten sein. Diese werden zwar meist in digitalen Formaten erstellt, sind aber in aller Regel weitgehend unabhängig voneinander. Man kann auch heute noch mit großer Sicherheit davon ausgehen, dass **kein Großprojekt im Bauwesen ausgeführt wird auf Basis einer konsistenten Menge von Technischen Dokumenten**, d.h. Widersprüche und Unstimmigkeiten sind unvermeidlich. Letztlich müssen dann die resultierenden Probleme oft vor Ort auf der Baustelle durch Erfahrung und Improvisation der verantwortlichen Bauingenieure gelöst werden. Oftmals resultieren diese jedoch in vielfältige **Mängel am Bau**.

Technische Dokumente im Bauwesen werden z.B. von Konstrukteuren und Bauzeichnern erstellt, die hierzu ein Vorstellungsvermögen des neu zu erstellenden Bauwerks entwickeln müssen. Hierzu gehören aber auch sogenannte Statikberechnungen und -berichte, für die aufwändige Tragwerksberechnungen erforderlich sind. Diese werden i.d.R. von Bauingenieuren (Statikern, Dynamikern) ausgeführt, die in ihrem Studium die entsprechenden Berechnungsverfahren gelernt haben.

An dieser Stelle sind die zwei Welten Modellbildung und Berechnung voneinander abhängig. Es muss sichergestellt werden, dass Änderungen im Modell in ihren Konsequenzen auf nachfolgende Berechnungen abgeschätzt werden, ggf. wird es notwendig sein, die Berechnung neu durchzuführen. Hierzu ist es jedoch erforderlich, dass entsprechende Werkzeuge verfügbar sind, die eine Neuberechnung möglichst mit überschaubarem Aufwand unmittelbar, interaktiv unterstützen.

In der Regel lernen Studierende des Bauingenieurwesens die gängigen Berechnungsverfahren für Tragwerksberechnungen, Wärmeberechnungen oder Festigkeitsberechnungen anhand einfacher Beispiele „von Hand“, anhand allgemeiner, genereller Softwarewerkzeuge wie Tabellenkalkulation, Mathematik Bibliotheken, etc. oder anhand einer Vielzahl unterschiedlicher Spezialsoftware. Diese haben inhärent einen überschaubaren Leistungsumfang, bieten aber relativ einfache und nachvollziehbare Funktionsabläufe. Auf dieser Basis kann den Studierenden ein Verständnis der Voraussetzungen und der Abläufe von Lösungsalgorithmen im Bauwesen vermittelt werden.

Da Berechnungsverfahren im Bauwesen in der Regel aus vielen, zeitaufwändigen Rechenschritten bestehen, hat dies schon den Bauingenieur Konrad Zuse motiviert, diese Vorgänge durch mechanisch ausführbare Lösungsalgorithmen zu unterstützen. Konrad Zuse erfand hierfür die Computertechnik und gilt als der Erfinder des Computers.

In der Praxis werden die Absolventen später konfrontiert mit höchst komplexen und umfangreichen Programmsystemen, die einen allgemeingültigen und anspruchsvollen Leistungsumfang bieten, mit denen komplexe Probleme für den professionellen Tragwerksentwurf gelöst werden können. Diese eignen sich aber in aller Regel nur sehr bedingt für die Ausbildung künftiger Bauingenieure, da sie keinerlei Einblicke in Funktionsumfang und Ablauf der internen Lösungsalgorithmen bieten. Solche Systeme werden auch oft als „Black Box Systeme“ bezeichnet. Eingabedaten werden aufbereitet, ein Berechnungslauf wird gestartet und Ergebnisse werden ausgegeben. Auf dieser Basis ist es kaum möglich, den künftigen Bauingenieuren ein Verständnis der Voraussetzungen und der Funktionsweise von Lösungsalgorithmen im Bauwesen zu vermitteln.

Das Entwickeln eines Verständnisses des Tragverhaltens und des Einflusses von Modifikationen im Tragwerksentwurf, erfordert jedoch Werkzeuge, die es einem Nutzer erlauben, Einblicke zu nehmen in den Funktionsumfang und Lösungsabläufe dieser Werkzeuge. Ebenso sollten diese Werkzeuge interaktive Einflussnahme auf Modellparameter und Darstellung der Ergebnisse unterstützen. So sollte es z.B. möglich sein, interaktiv Einwirkungen und Entwurfparameter zu verändern und möglichst umgehend Rückmeldungen zu erhalten über daraus resultierende Konsequenzen. Entsprechende Werkzeuge, die für die Ausbildung im Bauingenieurwesen genutzt werden können müssen bestimmte Voraussetzungen erfüllen, um die vorgenannten Ziele erfüllen zu können. Die Werkzeuge müssen

- in ihren inneren Abläufen uneingeschränkt einsehbar sein (**transparent**), sie müssen
- ohne Restriktionen in lesbarer Form zur Verfügung stehen (**offen**) und sie müssen
- in ihren Abläufen und Voraussetzungen relativ einfach, Schritt für Schritt, einschließlich relevanter Zwischenergebnisse beurteilt werden können (**nachvollziehbar**).

Eine interaktive Einflussnahme auf Modellparameter und Darstellung der Ergebnisse erfordert zwingend eine interaktive grafische Benutzeroberfläche (User Interface **UI**) in einer Berechnungssoftware, das es dem Programmierer ermöglicht, eine Vielzahl von nutzergesteuerten Ereignissen zu definieren und zu nutzen, und die es dem Nutzer ermöglichen, Ereignisse für eine Einflussnahme zu initiieren und entsprechend darauf zu reagieren.

Dies erfordert eine

- objektorientierte, ereignisgesteuerte Benutzeroberfläche (UI).

Heute ist völlig unstrittig, dass sowohl in der Lehre wie in der professionellen Nutzung Softwaresysteme genutzt werden für die Lösung von Berechnungen im Bauwerksentwurf. Klassische Verfahren, die „von Hand“ ausführbar waren, spielen praktisch kaum mehr eine Rolle. Lediglich zu Zwecken von Plausibilitätskontrollen und Überschlagsabschätzungen haben diese noch ihre Berechtigung.

## 2. Softwaresysteme für Tragwerksberechnungen

Die meisten Lehrpläne für Bauwerksberechnungen im Fach Bauingenieurwesen erfordern von den Studierenden das Lernen und die Ausführung von Berechnungen auf Basis *vorgegebener* Tragwerksmodellierungen. Ein einmal festgelegter Entwurf wird entweder „von Hand“ oder mittels einer Softwarelösung berechnet. Dies geschieht z.B. durch Idealisierungen wie Biegebalken oder Rahmentragwerk mit Auflagerbedingungen und Lasteinwirkungen.

Ursprünglich war die Aufstellung eines Modells für eine Bauwerksberechnung aufwändig und zeitintensiv. Auch die Ausführung einer Berechnung konnte nicht an einem „Personal Computer“ ausgeführt werden, sondern in einem bestenfalls firmeninternen Rechenzentrum. Die Interpretation der Berechnungsergebnisse erfolgte auf Basis langer alfanumerischer Listen, die aufwändig interpretiert und ausgewertet werden mussten. Dies hatte oft zur Folge, dass bestehende Entwürfe nur sehr zögerlich in ihren Grundannahmen geändert wurden und eher an die Anforderungen angepasst wurden. Dies wurde manchmal auch als „gesund rechnen“ eines bestehenden Entwurfs bezeichnet.

Die Leistungsfähigkeit der Computer und die Konzeption der Rechenprogramme erlaubten es ursprünglich nicht, die Berechnung eines bestimmten einzelnen Tragwerksentwurfs so schnell durchführen zu können, dass Entscheidungen über Entwurfsalternativen berücksichtigt werden konnten. Die Berechnung eines Tragwerks besteht aus den Schritten:

- Idealisierung und Modellierung eines Tragwerkentwurfs,
- Aufstellung eines Rechenmodells,
- Durchführung der Berechnung und
- Auswertung der Ergebnisse.

In diesem Ablauf sollte es unterstützt werden, Alternativen im Entwurf zu untersuchen und Konsequenzen von Entwurfsalternativen zu analysieren.

Um dieses Dilemma zu beheben, wäre ein Werkzeug erforderlich, das es den Nutzern erlaubt:

- eine Tragwerksberechnung unmittelbar auf vorhandene digitale Modelldaten aufzubauen oder einen Tragwerksentwurf mit wenig Aufwand für eine Neuberechnung zu formulieren,

- den Entwurf unmittelbar zu berechnen und
- die für den Entwurf wesentlichen Rechenergebnisse interaktiv darzustellen.

Mögliche Entwurfsvarianten - wie im Tragwerksentwurf z.B. zusätzliche oder entfernte Tragwerkselemente, geänderte Lagerbedingungen, Änderungen von Materialkennwerten oder Querschnittänderungen von Balkenelementen - müssen interaktiv geändert werden können, umgehend neu berechnet werden können und interaktiv, visuell mit den Ergebnissen vorhergehender Entwürfe verglichen und beurteilt werden können.

Mit einem solchen Werkzeug läge das Hauptaugenmerk der Studierenden nicht mehr auf der mechanischen Umsetzung *eines* vorgegebenen Entwurfs, sondern auf dem „Durchspielen“ unterschiedlicher Entwurfsvarianten und auf der Bewertung und Beurteilung unterschiedlicher Entwurfsvarianten.

Nach dem derzeitigen Stand der Technik stehen nahezu alle verfügbare Programmsysteme für Berechnungen im Bauingenieurwesen nur als ausführbare Programmsysteme zur Verfügung, die nur einen sehr beschränkten Einblick in ihre Leistungsfähigkeit und ihren Funktionsumfang erlauben. Oftmals sind solche Programmsysteme nur als sogenannte „Black-Box“ nutzbar, d.h. Einblicke in Funktionsweise und Einschränkungen sind kaum bis gar nicht möglich.

Die Brücke zwischen den einfachen und nachvollziehbaren Lösungsverfahren, die es erlauben, in der Lehre vermittelt zu werden, und den komplexen Berechnungsprogrammen, die in der Praxis verwendet werden, ist derzeit nur schwierig zu schließen.

Entweder werden „Lehrbeispiele“ als nachvollziehbare Programme genutzt, die Studierenden Einblicke und Verständnis von Funktionsweise und -umfang ermöglichen oder komplexe Berechnungsaufgaben werden anhand professioneller Software gelöst, deren Strukturen, Lösungsverfahren und Funktionsumfang dem Anwender verborgen sind, ja meist sogar lizenzrechtlich geschützt sind.

Diese Lücke soll die vorliegende Computersoftware für allgemeine Berechnungen nach dem Verfahren der Finiten Elemente schließen. Es ist der Anspruch dieser Software

- ein allgemeingültiges Lösungsverfahren für komplexe Berechnungsaufgaben im Bauingenieurwesen zur Verfügung zu stellen, das
- Studierenden einen vollen, uneingeschränkten Einblick erlaubt in die Funktionsweise und Lösungsabläufe dieser Verfahren und das
- konzeptionell die gängigen Problemklassen darstellt wie Lösung Systeme 1. Ordnung (Wärme), Systeme 2. Ordnung (Festigkeit), 2-dimensional, 3-dimensional, stationär bzw. statisch und instationär bzw. dynamisch.

### 3. Transparenz, Offenheit und Nachvollziehbarkeit

Die oben definierten Ziele erfordern, dass alle Lösungsabläufe für Berechnungen im Ingenieurwesen absolut **transparent** sind, d.h. eine Verwendung von Programmbestandteilen, die lediglich als maschinenlesbare, ausführbare Programme zur Verfügung stehen, ist

nicht akzeptabel. Solche Programmbestandteile erlauben keinerlei Einblicke in ihre internen Abläufe und Lösungsstrategien. Für Ingenieure ergibt sich hier eine natürliche Grenze zwischen Systemsoftware, die völlig unabhängig von spezifischen Anwendungen genutzt werden kann, und Programmbestandteilen, die spezifisch sind für die Lösung anwendungsspezifischer Probleme wie z.B. im Bauingenieurwesen. Allerdings haben sich die Grenzen zwischen diesen zwei Welten in den letzten Jahren gravierend verändert und verschoben. Mittlerweile stehen in moderner Systemsoftware eine Vielzahl an allgemeinen Funktionalitäten zur Verfügung, die von aktueller Anwendungssoftware selbstverständlich erwartet werden. Hierzu gehören insbesondere Funktionalitäten für den Dateizugriff, die Datenverwaltung, Netzwerkkommunikation, Grafik und Benutzeroberflächen. Als Teil der Systemsoftware sind diese dann nicht mehr unbedingt transparenter Teil der Anwendungssoftware. Sie werden manchmal auch als *Middleware* bezeichnet.

Transparenz wird häufig beeinträchtigt oder gar verhindert durch rechtliche Restriktionen wie Urheber-, Nutzungs- und Verwertungsrechte. Transparenz erfordert, dass jegliche Softwarelösung in diesem Zusammenhang frei sein muss von rechtlichen oder geschäftlichen Einschränkungen. Sie muss also absolut **offen** frei verfügbar sein.

Transparenz und Offenheit sind Voraussetzungen für eine vollständige **Nachvollziehbarkeit** aller relevanten Lösungsschritte. D.h. ein Nutzer solcher Programme muss in der Lage sein, jeden einzelnen Schritt eines spezifischen Lösungsverfahrens bei Bedarf Schritt-für-Schritt nachvollziehen zu können.

Zu diesen Zwecken wird ein - nicht unumstrittener - Weg gewählt für eine Implementierung von Lösungsalgorithmen im Bauingenieurwesen. In der Regel gilt in der akademischen Ausbildung die Forderung nach der Nutzung frei verfügbarer, nicht eigentumsrechtlich geschützter Software, sogenannter „Open Source“ Software. Diese Forderung galt ursprünglich auch für die genutzte Betriebssystemsoftware. Daher wurden an Universitäten für die Lehre und Forschung meist Betriebssysteme wie UNIX oder LINUX genutzt.

In Lehre und Forschung wurde die Programmierung akademischer Softwarelösungen ebenfalls mit Hilfe frei verfügbarer Entwicklungsumgebungen und -sprachen wie C und Java durchgeführt.

Diese Diskrepanz führte zu häufigen Klagen in der Praxis, dass die Ausbildung „praxisfern“ sei, da in der Berufspraxis der Absolventen im Bauingenieurwesen andere Systeme genutzt werden, die allerdings in der Regel „nicht-offen“ und urheberrechtlich geschützt sind.

Die Diskrepanz wird oftmals noch verschärft, da sich die Entwicklung kommerzieller Systeme meist - teilweise wesentlich - dynamischer vollzieht als die Entwicklung frei verfügbarer Softwaresysteme. Im Bereich der *Betriebssystemsoftware* hat sich in den letzten Jahren das System Windows der Fa. Microsoft durchgesetzt und wird inzwischen auch vermehrt im akademischen Bereich genutzt. Im Bereich der *Entwicklung* von Softwarelösungen (Programmierung) im Bauingenieurwesen wird nach wie vor stark auf offene Software wie Java mit entsprechenden Entwicklungsumgebungen (IDE Integrated Development Environment) wie Eclipse gesetzt. Seit die Sprache Java jedoch von der Fa. Oracle übernommen wurde und - trotz gegenteiliger Versprechungen der Fa. Oracle - verstärkt eigentumsrechtlich eingeschränkt wird, werden zunehmend andere, professionelle Entwicklungsum-

gebungen als Alternative genutzt obwohl sie proprietär sind, d.h. eigentumsrechtlich geschützt.

Für die Zwecke dieser Softwareentwicklung wurde die relativ junge Programmiersprache C# der Fa. Microsoft gewählt, da diese als moderne Entwicklung (erstmal veröffentlicht im Jahre 2001) der Fa. Microsoft verfügbar ist, welche

- die Konzepte der **Objektorientierung** konsequent umsetzt, welche
- mit **.NET Framework** eine Middleware zur Verfügung stellt für Dateizugriff, Datenverwaltung (**Language Integrated Query (LINQ)**), Netzwerkkommunikation und Grafik, welche
- über eine moderne Nutzeroberfläche (User Interface UI) verfügt, die als **Windows Presentation Format (WPF)** voll in die Windows Umgebung integriert ist mit einer weitgehenden Trennung von Entwurf (**Extensible Application Markup Language (XAML)**) und Funktionalität des UI und welche
- mit **Visual Studio** durch eine moderne, professionelle Entwicklungsumgebung (IDE) unterstützt wird, die für die Nutzung im akademischen Bereich frei verfügbar ist.

Die Softwareentwicklung soll allen Studierenden frei zur Verfügung stehen inklusive aller zugehöriger Komponenten im Quellcode. Hiermit sollen Nachvollziehbarkeit und Erweiterbarkeit aller Modellierungen und Lösungsschritte gewährleistet werden.

Auf die Nutzung jedweder Bibliotheken von Drittanbietern wurde verzichtet, um die Transparenz und Nachvollziehbarkeit aller Modellierungen und Lösungsschritte sicher zu stellen.

#### 4. Konzeptstudie für einen Integrativen Tragwerksentwurf

In der gegenwärtigen Praxis der Bauingenieure sind praxisrelevante Tragwerksentwürfe in der Regel hochkomplex und hoch umfangreich. Die Entwicklung sogenannter Bauwerksinformationsmodelle (**BIM**) hat zum ständigen Anstieg der Komplexität solcher Modelle beigetragen.

Noch in den 1980er Jahren waren die erforderlichen Einzelschritte für die Berechnung eines Tragwerkentwurfs dermaßen aufwändig, dass zwischen digitalem Entwurf eines Rechenmodells, Entwurfsberechnung und Ergebnisauswertung teilweise Tage bis Wochen lagen.

Im Jahre 2012 haben die Autoren *F. Gerold, K. Beucke, F. Seible* einen Artikel im „Journal of Computing in Civil Engineering“ veröffentlicht unter dem Titel „Integrative Structural Design“. In diesem Artikel wird die Notwendigkeit eines „Integrative Design“ im Gegensatz zum „Integrated Design“ formuliert. „Integrated“ wird hier verstanden als die integrierte Berücksichtigung zusammenhängender Aspekte eines Entwurfs in einem automatisierten Prozess. „Integrativ“ hingegen wird verstanden als die interaktive Tätigkeit eines Entwerfenden im Hinblick auf die Entwicklung eines optimalen Entwurfs gemäß seinen Intentionen. Beurteilungen im Hinblick auf Entwurfskriterien verbleiben bei dem entwerfenden In-

genieur. Dieser bekommt lediglich ein anderes Werkzeug, das es ihm ermöglicht, solche Entwurfsentscheidungen auf Basis komplexer Berechnungen interaktiv zu treffen.

Dass dies heute möglich ist, zeigt der Artikel anhand von Ergebnissen für Erdbebenberechnungen eines komplexen Rahmentragwerks. Eine dynamische Berechnung auf Basis realer Beschleunigungsdaten mit 30.000 Einzelschritten erfolgt im Sekundenbereich und die maßgeblichen Entwurfsdaten können direkt visuell dargestellt und beurteilt werden. Änderungen in den Modelldaten wie Verstärkungen von Tragelementen und Hinzufügen zusätzlicher Tragelemente können in ihren Auswirkungen interaktiv ermittelt und beurteilt werden. Gerade im Fall von Erdbebenberechnungen ist ein solches Vorgehen von großer Bedeutung, da bisher solche Berechnungen sehr zeitaufwändig sind und vor allem in ihren Konsequenzen nur sehr schwierig intuitiv erfassbar sind. Verstärkung von Tragelementen können in solchen Lastfällen nicht nur positive, sondern manchmal sogar negative Folgen mit sich bringen.

Der Weg zu einem „Integrativen Design“ in diesem beschriebenen Sinne ist sicher noch lang, aber es besteht die begründete Hoffnung, dass die vorliegende Software auch als Konzeptstudie für eine solche Unterstützung dienen kann.

Grundsätzlich sind die Programmfunktionalitäten unterteilt in:

- Modelldaten einlesen, editieren und sichern
- Modelldaten anzeigen und visualisieren
- Modell berechnen
- Ergebnisse der Berechnungen tabellarisch anzeigen und grafisch visualisieren

Modellvarianten können in separaten Fenstern parallel angezeigt und visualisiert werden. Die Ergebnisse unterschiedlicher Modellvarianten können ebenso parallel dargestellt und beurteilt werden.

Die Definition von Modelldaten erfolgt klassischerweise in Textdateien. Einfache Beispiele sind für die unterschiedlichen Anwendungen in input-Verzeichnissen verfügbar. Ein einfacher Texteditor erlaubt die Darstellung und das Editieren vorhandener Eingabedateien.

Nach dem Einlesen einer Eingabedatei können die Modelldaten in tabellarischer Ausgabe angezeigt werden oder grafisch visualisiert werden. In der tabellarischen Anzeige der Daten können Eingabedaten editiert oder neu definiert werden. Das bedeutet dann eine interaktive Veränderung der aktuellen Modelldaten und diese aktualisierten Modelldaten können dann in einer neuen Eingabedatei gesichert werden.

Modellberechnungen auf Basis der aktuellen Modelldaten können entweder explizit veranlasst werden oder werden implizit neu durchgeführt, wenn Ergebnisse neu dargestellt werden. Dies gilt nicht für instationäre/dynamische Berechnungen. Diese müssen immer explizit vom Nutzer veranlasst werden.

Die grafische Darstellung von Ergebnisdaten kann interaktiv untersucht werden. Hierzu veranlasst ein „Click“ auf einzelne Modelldarstellungen wie Knoten, Elemente, Lasten, Randbedingungen, etc. die Darstellung eines Pop-Up-Fensters mit der Textdarstellung der



wesentlichen Informationen, die mit diesem „Click“ verbunden sein können. Dies schließt auch die textliche Darstellung der Berechnungsergebnisse wie z.B. Schnittgrößen für Balkenelemente und Verformungen von Modellknotenpunkten. Ein solches Pop-Up-Fenster kann mit einem Rechts-Click wieder geschlossen werden.

## Benutzerhandbuch

### Definition der Daten eines neuen Modells:

Grundlage einer jeden Berechnung ist die Definition eines **Modells** mit entsprechenden **Modelldaten**, welche das **Modellverhalten** bestimmen.

Die Modelldaten bestehen aus **Knoten**, welche die **Modellgeometrie** bestimmen, aus **Elementen**, welche die **Modelltopologie** bestimmen, aus äußeren **Einwirkungen** (Lasten) auf das Modell und aus **Randbedingungen**, die z.B. Temperatur, Festhaltungen und Lager bestimmen.

Die Definition sämtlicher Daten für ein konkretes, vollständiges Modell erfolgt in der Regel in einer separaten Textdatei, da dies häufig effizienter und einfacher ist als in interaktiven Nutzerdialogen.

Als Anwendungsgebiete wurden exemplarisch die Tragwerksberechnung, Wärmeberechnung und Elastizitätsberechnung ausgewählt. Die Tragwerksberechnung wird häufig als ein gesonderter Spezialfall von Bauwerksberechnungen behandelt. Sie ist aber für Bauingenieure von zentraler Bedeutung und soll in diesem Zusammenhang auf der gleichen allgemeinen Basis vorgestellt werden wie andere physikalische Berechnungen. Sie wird hier als 1D oder 2D Idealisierung mit 2 oder 3 Freiheitsgraden behandelt. Die physikalisch einfachste Berechnung ist die Wärmeberechnung, die auf der gleichen allgemeinen Basis behandelt wird. Hier werden Implementierungen in 1D und 2D vorgestellt mit 1 Knotenfreiheitsgrad. Tragwerksberechnungen sind in der Regel ein spezieller Fall allgemeiner Festigkeitsberechnungen, die aus der Elastizitätstheorie abgeleitet werden und sie werden daher in der Praxis häufig zusammen behandelt. Der allgemeinere Fall der Elastizitätsberechnungen wird in der Ausbildung aber häufig erst später behandelt, da sowohl Theorie und Interpretation der Ergebnisse anspruchsvoll sind. Ergebnis einer Elastizitätsberechnung sind Dehnungen und Spannungen, die in der praktischen Nutzung häufig nicht direkt nutzbar sind, da hier in der Regel Elementschnittkräfte genutzt werden.

Physikalische Berechnungen werden in der Regel untersucht für Lasteinwirkungen, die zeitlich unveränderlich sind. Dies wird in der Tragwerksberechnung als „statisch“ und in der Wärmeberechnung als „stationär“ bezeichnet. Sollen zudem aber auch zeitlich veränderlich wirkende Lasteinwirkungen wie Wind, Erdbeben oder Wärmesteigerungen wie Anfahrkurven für einen Kamin berücksichtigt werden, so werden Verfahren für instationäre Wärmeberechnungen oder dynamische Tragwerksberechnungen benötigt.

Einfache Beispiele stehen in einem **Unterverzeichnis „input“** für die unterschiedlichen Anwendungsgebiete zur Verfügung. Dies sind

<b>input</b>	<b>Tragwerksberechnung, Wärmeberechnung und Elastizitätsberechnung</b>
--------------	--

Alle Eingabedateien sind gekennzeichnet durch die Dateiendung „inp“.

„Leere“ Eingabedateien mit auskommentierten Schablonen als Beispiele für die jeweils unterstützten Arten von Eingabezeilen stehen in Dateien „Vorlage.inp“ zur Verfügung.

### **Veranschaulichung und Verändern/Ergänzen der Modelldefinitionen:**

Gemäß der Grundintention einer interaktiven Variation unterschiedlicher Modelldaten für eine **interaktive Modellexploration**, stehen unterschiedliche Methoden und Werkzeuge für das Editieren, Löschen und Ergänzen von Modell bestimmenden Daten zur Verfügung.

Sämtliche Daten, die für ein bestimmtes Modell definiert sind, können in alphanumerischer, tabellarischer Form ausgegeben und angezeigt werden (**tabellarische Modelldarstellung**). Jede Zeile kann in einer Ausgabetabelle ausgewählt und markiert werden. Angewählte Tabellenzeilen können dann z.B. komplett gelöscht werden, womit zugleich auch die zugehörigen Informationen im Modell gelöscht werden. Es ist aber auch möglich einzelne Zellen einer angewählten Tabellenzeile anzuwählen und deren Inhalt zu editieren. So können z.B. Knotenkoordinaten editiert werden, um die Modellgeometrie zu verändern, oder es können ganze Knoten oder Elemente aus dem Modell gelöscht werden, um direkt anschließend das geänderte Modell neu berechnen zu können und dessen geändertes Verhalten beurteilen zu können.

Zusätzliche, neue Modelldaten (Knoten, Elemente, Material, Einwirkungen und Randbedingungen) können durch einen Doppelklick in einer entsprechenden Tabelle initiiert werden. Es öffnet sich dann ein entsprechender Nutzerdialog zur Festlegung der zugehörigen, erforderlichen Modelldaten. Das Modell wird dann entsprechend erweitert.

Häufig ist es anschaulicher Modelldaten nicht in einer tabellarischen Auflistung zu inspizieren und zu verändern, sondern in einer zusätzlichen **visuellen Modelldarstellung**. Hierzu kann eine visuelle Modelldarstellung unabhängig oder auch parallel zur tabellarischen Darstellung in einem separaten Fenster geöffnet werden. In diesen Darstellungen werden Modelldaten durch Grafik- und Textelemente veranschaulicht. Die zugehörigen Modelldaten der meisten Grafik- und Textelemente können durch einen Linksklick ausgewählt werden, woraufhin sich dann an der Position des Klicks ein sogenanntes „Popup-Fenster“ öffnet mit den zugehörigen Modellinformationen zu den angewählten Grafik- oder Textelementen. Mit einem Rechtsklick wird das Fenster wieder geschlossen.

Grundsätzlich sind programminterne Modelländerungen nur zur Programmlaufzeit gültig und werden nicht automatisch bei Beendigung des Programms dauerhaft gespeichert. Ein aktueller Modellzustand kann jedoch durch den Nutzer bewusst durch die Auswahl des Menüpunkts „Tragwerksdaten sichern“ (bzw. Wärmedaten oder Elastizitätsdaten) in einer neuen Eingabedatei mit festzulegendem Namen gespeichert werden.

### **Beurteilung der Berechnungsergebnisse:**

Eine zweite Grundintention dieser Software ist die **interaktive Ergebnisexploration**. Zu diesem Zweck werden die wesentlichen Ergebnisse einer Berechnung wieder in **tabellarischer Form** dargestellt und können unabhängig davon oder parallel dazu in visueller Form veranschaulicht werden.

Die primären Ergebnisse einer Berechnung sind die primalen Variablen (z.B. Knotentemperaturen oder Knotenverformungen) und die dualen Variablen (Reaktionen an vordefinierten Randbedingungen, z.B. Lager). Für die Beurteilung des Modellverhaltens sind jedoch in der Regel weitere, abgeleitete Ergebnisse von wesentlicher Bedeutung. Dies sind z.B. der Wärmefluss oder die Elementschnittkräfte. Diese werden berechnet und können tabellarisch oder visuell dargestellt werden.

Die **visuelle Darstellung** der Ergebnisse einer Berechnung beinhaltet zum einen die grafische Darstellung der primalen Ergebnisse (Modellverformungen bzw. -temperaturen), der dualen Ergebnisse (Wärmefluss bzw. Lagerreaktionen) und zum anderen eine grafische Darstellung abgeleiteter Ergebnisse wie z.B. Schnittkraftverläufe oder Wärmefluss.

Die tabellarische Darstellung der Ergebnisse erlaubt eine schnelle Gesamtübersicht über die relative Größe der Ergebnisse. Die visuelle Darstellung hilft die Lokalisierung und Verteilung der Ergebnisse zu veranschaulichen. Sie wird unterstützt durch Popup-Fenster, die nach Linksklick auf Grafik- und Textelemente aufklappen und Detailinformationen zu Ergebnissen an diesen Elementen anzeigen. Mit einem Rechtsklick wird das Fenster wieder geschlossen.

### Programm Download über Git:

<https://github.com/KarlBeucke/FE-Berechnungen-App>

mit einer ausführbaren Applikation und einer entsprechenden Setup-Datei für eine Installation unter MS-WINDOWS. Der Installationsort sollte der Standardordner *Dokumente/* sein mit einem Verzeichnis *Dokumente/FE Berechnungen*. In diesem sind ebenfalls eine Reihe einfacher Eingabedateien gespeichert, die für jedes Anwendungsgebiet einfache Lehrbeispiele zur Verfügung stellen. Schließlich ist dort noch diese Dokumentation gespeichert.

<https://github.com/KarlBeucke/FE-Berechnungen>

mit einem kompletten Projekt für MS-Visual Studio, mit allen Programmdateien im Quellcode einschließlich einer FE Bibliothek für allgemeine Standardfunktionalitäten wie Matrizenalgebra, Gleichungslöser, Zeitschrittlöser, allgemeine Modellinformationen wie z.B. Geometrieknoten und abstrakten Modellinformationen in einer Klassenhierarchie.

Dieses Projekt kann gestartet werden und im Debug-Modus mit „*break points*“ und „*watch points*“ im Ablauf Schritt-für-Schritt nachvollzogen werden.

Eine Solution-Datei (*FE Berechnungen.sln*) kann direkt in MS-Visual Studio gestartet werden.

## Programmstart:

Im Startfenster des Programms befindet sich lediglich die Auswahl für

- Tragwerksberechnungen
- Wärmeströmungsberechnungen
- Elastizitätsberechnungen

Das allgemeine weitere Vorgehen erfolgt in den drei Anwendungen generell in den Schritten

- Modelldaten einlesen, editieren und sichern
- Modelldaten anzeigen und visualisieren
- Berechnung ausführen
- Berechnungsergebnisse anzeigen und visualisieren

Zur Verfügung stehen einerseits **statische bzw. stationäre Berechnungsverfahren** und ggf je nach Anwendungsgebiet auch **dynamische bzw. instationäre Zeitschrittverfahren** mit Berechnungen einer festzulegenden Anzahl von Eigenlösungen der eingelesenen physikalischen Systeme, d.h. die **Eigenwerte und Eigenvektoren**.

Hierfür sind Funktionalitäten folgende verfügbar:

- zeitabhängige Modelldaten anzeigen  
-----
- Eigenlösungen berechnen
- Eigenlösungen Resultate anzeigen
- Eigenlösungen Resultate visualisieren  
-----
- Zeitverlaufsrechnung durchführen  
-----
- zeitabhängige Ergebnisse anzeigen
- zeitabhängige Modellzustände visualisieren
- Kontenzeitverläufe visualisieren

Das **Einlesen der Modelldaten** erfolgt in der Regel aus einer Textdatei. Diese kann editiert werden und als neue Modelldatei unter einem nutzerdefinierten Namen gesichert werden.

**Modelldaten** können aber auch **interaktiv über die Nutzeroberfläche editiert und ergänzt** werden. Hierzu werden die aktuellen Modelldaten in Tabellenform angezeigt. In der Anzeige können Tabellenzeilen gelöscht werden, Tabellenelemente editiert werden und neue Elemente durch Doppelklick auf die jeweilige Tabelle interaktiv erzeugt werden. We-

sentlich ist dabei, dass die Modelldaten im Hintergrund konsistent angepasst werden.

Die **Berechnungsergebnisse** können in separaten Fenstern entweder **alphanumerisch in Tabellenform** angezeigt werden oder grafisch **visualisiert** werden.

Für zeitabhängige Berechnungen kann der **Zustand des Gesamtmodells an einem bestimmten Zeitschritt visuell dargestellt** werden.

Zeitabhängige Berechnungsergebnisse werden als **Zeitverlauf über die gesamte Berechnungszeit an einem bestimmten Modellknoten visuell** dargestellt.

### **Beschreibung der Modelldaten**

Ein Berechnungsmodell besteht generell aus **Knoten**, welche die Geometrie eines Modells definieren. Die Zusammenhänge eines Modells werden bestimmt durch **Elemente**, welche die Topologie eines Modells und das physikalische Verhalten definieren. Vorgegebene Randwerte werden bestimmt durch **Randbedingungen**, welche je nach Typ des Modells z.B. Auflagerbedingungen oder Wärmewerte sein können. **Einwirkungen auf das Modell** werden häufig als **Lasten** bezeichnet, welche z.B. Kräfte oder Wärmequellen sein können.

Auf Basis dieser Modellinformationen wird ein lineares Gleichungssystem erstellt und gelöst. Im Ergebnis werden die resultierenden **primalen unbekannten Knotenwerte** (Verformungen, Temperatur) für die Knotenfreiheitsgrade und **dualen unbekannten Reaktionen** (Kräfte, Wärmefluss) an den vorgegebenen Randbedingungen berechnet.

Der **Zustand eines Berechnungsmodells** (Schnittgrößen, Wärmefluss) wird auf der Basis der resultierenden Knotenwerte und Reaktionen für jedes einzelne Element berechnet.

**Zeitabhängige Berechnungen** (dynamisch bzw. instationär) werden mit Hilfe von Zeitschrittverfahren durchgeführt. Diese werden durch zeitveränderliche Einwirkungen (Erdbeben und Wind bzw. Wärmeeinstrahlung) bestimmt und werden auch als „Berechnungen im Zeitbereich - time domain“ bezeichnet. Alternativ stehen Berechnungen im „Frequenzbereich - frequency domain“ zur Verfügung. Hierfür wird eine bestimmte Anzahl von Eigenlösungen (Eigenwerte und Eigenvektoren) ermittelt. Im Ergebnis werden die Modellzustände an einem bestimmten Zeitschritt ermittelt oder der Zeitverlauf von Knotenwerten über den gesamten Berechnungszeitraum.

### **Modelldaten eingeben und einlesen**

Standardmäßig werden die Modelldaten aus einer Datei in einem input-Ordner der jeweiligen Anwendung eingelesen. Der Speicherort der Eingabedatei kann aber auch vom Anwender spezifisch ausgewählt werden. Standardmäßig ist der Dateityp „.inp“ für die Eingabedatei vorgesehen.

Zur Eingabe der Modelldaten kann ein beliebiger Texteditor benutzt werden. Ein sehr einfacher Editor ist integriert in die Software, um Eingabedateien innerhalb der Softwareumgebung editieren zu können.

Eine neue Zeile mit einem Identifikator (**Schlüsselwort**) gefolgt von beliebig vielen Zeilen

kennzeichnet jeweils einen Bereich von Eingabewerten, die zeilenweise festgelegt werden. Unterschiedliche Eingabebereiche kennzeichnen thematisch zusammenhängende Eingaben. Das Ende eines Eingabebereichs ist durch eine Leerzeile festgelegt.

Einzelne, unterschiedliche Eingabewerte jeweils werden durch einen Tabulator getrennt.

**Ergänzungen der Modelldaten in der tabellarischen Darstellung** der aktuellen Modelldaten werden durch spezifische Eingabedialoge für alle Modellelemente unterstützt, welche durch Doppelklick auf die jeweilige Tabelle (DataGrid) aktiviert werden.

### **Anwendungsübergreifende Eingabewerte**

**Alle** Modelldaten (Knoten, Elemente, Randbedingungen, Einwirkungen) und auch das Modell selber sind durch einen **eindeutigen, dauerhaft gespeicherten, laufzeitunabhängigen Identifikator** charakterisiert, der nicht durch die Anwendung verändert wird. Der **Zugriff auf** Modelldaten und **Bezüge zwischen** Modelldaten (z.B. Element auf Knoten) erfolgen **ausschließlich** über die Identifikatoren. Der Zusammenhang zwischen Identifikatoren und Speicherort (Referenz) von Modelldaten wird erst unmittelbar vor der Nutzung der Modelldaten hergestellt. Dies erlaubt interaktive Änderungen der **Inhalte** von Modelldaten (z.B. Koordinaten von Knoten) und deren konsistente Nutzung im weiteren Programmablauf.

#### **Liste verfügbarer Schlüsselworte:**

##### Schlüsselwort ***ModellName***

kennzeichnet die Festlegung eines spezifischen Namens für die Modelldaten in der Folgezeile.

##### Der Identifikator ***Raumdimension***

kennzeichnet die modellübergreifende Festlegung einer zwei- oder dreidimensionalen Modellierung in der Folgezeile und getrennt durch einen Tabulator die modellübergreifende Anzahl der Knotenfreiheitsgrade.

##### Schlüsselwort ***Knoten***

erlaubt in der Folgezeile entweder die Eingabe einer spezifischen Anzahl von Knotenfreiheitsgraden für Folgeknoten oder die Angabe eines eindeutigen Knotenidentifikators und die Knotenkoordinaten in Abhängigkeit der gewählten Raumdimension.

Folglich können Folgezeilen entweder

- einen Wert für die Anzahl der Knotenfreiheitsgrade oder
- zwei, drei oder vier Werte für eindeutige Knotenidentifikatoren, x- und y- und z-Koordinate des Knotens enthalten.

In der einfachsten Form sieht folglich der allgemeine Inhalt einer Eingabedatei für Modell- und Knotenwerte wie folgt aus:

ModellName  
Stockwerkrahmen

Raumdimension		bzw. mit Knotengelenk		Raumdimension
2	3			2
				2

Knoten

k1	0	0
k2	0	4
2		
k3	3	4
3		
k4	3	0

Der Modellname ist definiert als „Stockwerkrahmen“. Die Raumdimension ist definiert mit 2D und 3 Knotenfreiheitsgraden für eine x- und z-Verformung und eine Knotenverdrehung. Zwei Knoten sind definiert mit den eindeutigen Identifikatoren *k0* und *k1* und den x,y-Koordinaten (0,0) bzw. (6,0).

### Generierung von Modellknoten

Regelmäßig verteilte Knoten können auch generiert werden. Hierfür stehen folgende Identifikatoren zur Verfügung:

Eine **Knotengruppe** ist definiert durch einen Anfangsidentifikator definiert in der 1. Zeile, gefolgt von einem 6-stelligen Zähler und einer Folge von Koordinaten. Knoten n000000 mit (0, 3) und n000001 mit (0,2) werden generiert durch:

Knotengruppe

n	
0	3
0	2

Eine **äquidistantes Knotennetz** ist definiert durch eine Zeile mit jeweils 3 Werten für Startkoordinate, Inkrement und Anzahl Wiederholungen in x,y und z. Die eindeutigen Identifikatoren werden generiert durch Inkrementierung von jeweils 2 Stellen in x, y und z.

**Äquidistantes Knotennetz**

**1D**    *N*    0.    2.    12  
erzeugt 3 Knoten in 1D in Abständen von x=2:  
*N00 (0), N01 (2), N02 (4)*

**2D**    *A*    0.    1.    3    1.    1.  
erzeugt ein Netz von 3\*3 Knoten in 2D mit Startkoordinaten (0;1) und Abständen von 1. in x, y:  
*A0000 (0;1) A0100 (1;1) A0200 (2;1)*  
*A0001 (0;2) A0101 (1;2) A0201 (2;2)*  
*A0002 (0;3) A0102 (1;3) A0202 (2;3)*

**3D**    *B*    0.    1.    3    1.    1.    3    2.    1.    3  
erzeugt ein Netz von 3\*3\*3 Knoten in 3D mit Startkoordinaten (0;1;2) und Abständen von 1. in x, y, z:  
*B000000 (0;1;2) B010000 (1;1;2) B020000 (2;1;2)*    erzeugt eine x,y Ebene mit z=2  
*B000100 (0;2;2) B010100 (1;2;2) B020100 (2;2;2)*  
*B000200 (0;3;2) B010200 (1;3;2) B020200 (2;3;2)*  
  
*B000001 (0;1;3) B010001 (1;1;3) B020001 (2;1;3)*    erzeugt eine x,y Ebene mit z=3  
*B000101 (0;2;3) B010101 (1;2;3) B020101 (2;2;3)*  
*B000201 (0;3;3) B010201 (1;3;3) B020201 (2;3;3)*

*B000002 (0;1;4) B010002 (1;1;4) B020002 (2;1;4) erzeugt eine x,y Ebene mit z=4*  
*B000102 (0;2;4) B010102 (1;2;4) B020102 (2;2;4)*  
*B000202 (0;3;4) B010202 (1;3;4) B020202 (2;3;4)*

## Eine **Variables Knotennetz**

ist definiert durch eine beliebige Folge von 2 Zeilen mit jeweils 1 Zeile mit beliebig vielen Knotenabständen gefolgt von einer weiteren Zeile mit einem Textidentifikator für alle Knoten und den Startkoordinaten, die entweder 1, 2 oder 3 Werte enthalten je nach der Dimensionalität (1D, 2D, 3D) des Netzes. Mehrere Netze können generiert werden.

### *Variables Knotennetz*

**1D**    0.        1.        3.        6.        erzeugt 4 Knoten in 1D mit der Startkoordinate (0) und  
           X        0.                                Koordinaten 0, 1, 3, 6  
           X00 (0) X01 (1) X02 (3) and X03 (6)

**2D**    0.        1.        3.        6.        erzeugt 4\*4 Knoten in 2D mit Startkoordinaten (0;0) und  
           Y        0.        0.                                Koordinaten 0, 1, 3 und 6:  
           Y0000 (0;0) Y0100 (1;0) Y0200 (3;0) and Y0300 (6;0),  
           Y0001 (0;1) Y0101 (1;1) Y0201 (3;1) and Y0301 (6;1),  
           Y0002 (0;3) Y0102 (1;3) Y0202 (3;3) and Y0302 (6;3),  
           Y0003 (0;6) Y0103 (1;6) Y0203 (3;6) and Y0303 (6;6)

**3D**    0.        1.        3.        6.        erzeugt 4\*4\*4 Knoten in 3D mit Startkoordinaten (0;0;0) und  
           Z        0.        0.        0.                                Koordinaten 0, 1, 3 und 6:  
           Z000000 (0;0;0) Z010000 (1;0;0) Z020000 (3;0;0) and Z030000 (6;0;0),  
           Z000100 (0;1;0) Z010100 (1;1;0) Z020100 (3;1;0) and Z030100 (6;1;0),  
           Z000200 (0;3;0) Z010200 (1;3;0) Z020200 (3;3;0) and Z030200 (6;3;0),  
           Z000300 (0;6;0) Z010300 (1;6;0) Z020300 (3;6;0) and Z030300 (6;6;0)  
           Z000001 (0;0;1) Z010001 (1;0;1) Z020001 (3;0;1) and Z030001 (6;0;1),  
           Z000101 (0;1;1) Z010101 (1;1;1) Z020101 (3;1;1) and Z030101 (6;1;1),  
           Z000201 (0;3;1) Z010201 (1;3;1) Z020201 (3;3;1) and Z030201 (6;3;1),  
           Z000301 (0;6;1) Z010301 (1;6;1) Z020301 (3;6;1) and Z030301 (6;6;1)  
           Z000002 (0;0;3) Z010002 (1;0;3) Z020002 (3;0;3) and Z030002 (6;0;3),  
           Z000102 (0;1;3) Z010102 (1;1;3) Z020102 (3;1;3) and Z030102 (6;1;3),  
           Z000202 (0;3;3) Z010202 (1;3;3) Z020202 (3;3;3) and Z030202 (6;3;3),  
           Z000302 (0;6;3) Z010302 (1;6;3) Z020302 (3;6;3) and Z030302 (6;6;3)  
           Z000003 (0;0;6) Z010003 (1;0;6) Z020003 (3;0;6) and Z030003 (6;0;6),  
           Z000103 (0;1;6) Z010103 (1;1;6) Z020103 (3;1;6) and Z030103 (6;1;6),  
           Z000203 (0;3;6) Z010203 (1;3;6) Z020203 (3;3;6) and Z030203 (6;3;6),  
           Z000303 (0;6;6) Z010303 (1;6;6) Z020303 (3;6;6) and Z030303 (6;6;6)



## Eingabewerte Tragwerksberechnung

Die Implementierung von Elementen für die Tragwerksberechnung ist im Wesentlichen beschränkt auf 2-dimensionale Fachwerk- und Biegebalkenelemente mit und ohne Gelenken, d.h. die Raumdimension ist 2 und die Anzahl der Knotenfreiheitsgrade ist 2 oder 3.

Gelenke in der Tragstruktur werden definiert, indem 2 Knotenfreiheitsgrade vor der Knotendefinition festgelegt werden oder indem Elemente mit Gelenkdefinitionen an ihren Enden ausgewählt werden. Verfügbare **Schlüsselworte** beinhalten:

- **Biegebalken**

erfordert in der Folgezeile die Eingabe eines eindeutigen Elementidentifikators, gefolgt von 2 Knotenidentifikatoren, 1 Identifikator für den zugeordneten Querschnitts- und einem weiteren für den zugeordneten Materialidentifikator.

Folglich können Identifikator und Folgezeilen in der einfachsten Form wie folgt aussehen:

*Biegebalken*

*Col00 F0 G0 c0 m0*

- **Querschnitt**

erfordert in der einfachsten Form in Folgezeilen die Eingabe eines eindeutigen Identifikators für den Querschnittstyp, einen Wert für die zugeordnete Querschnittsfläche und einem optionalen Wert für das zugeordnete Trägheitsmoment.

*Querschnitt*

*c0 0,18 0,0054*

- **Material**

erfordert in Folgezeilen die Eingabe eines eindeutigen Identifikators für den Materialtyp, einen Wert für den zugeordnete Elastizitätsmodul und optional 1 - 5 weiteren Werten für die zugeordnete Querdehnungszahl, die spezifische Masse und Federsteifigkeiten in x,y und r, die für Federelemente benötigt werden.

*Material*

*m0 2,1e7 0,22 0,175*

- **Knotenlast**

erfordert in Folgezeilen die Eingabe eines eindeutigen Identifikators für die Knotenlast, einen Wert für den zugeordnete Knoten, 1 bis 3 weiteren Werte für die Lastwerte in x-, y- und z-Richtung.

*Knotenlast*

*Knotenlast1 U3-3 1000 0 0*

- **Lager**

erfordert in Folgezeilen die Eingabe eines eindeutigen Identifikators für die Lagerbedingung, für den zugeordnete Knoten, eine Zeichenfolge aus xyr für die Festhaltungen in x,y,r und optional 3 weitere Werte für vorgegebene Lagerverformungen in x, y und r.

Lager  
Lager0            F0       xy

Über diese grundlegenden Modellelemente hinaus gibt es **weitere Optionen** für die Festlegung von Elementen und Lasten.

- 
- **Fachwerk**

mit Elementidentifikator, Anfangs- und Endknoten und Materialidentifikator.

Fachwerk  
e00    n0       n1       c1       iso

- **BiegebalkenGelenk**

mit Elementidentifikator, Identifikatoren für 1. und 2. Elementknoten, Querschnittsidentifikator, Materialidentifikator und Festlegung, ob das Gelenk am 1. oder 2. Knoten ist.

BiegebalkenGelenk  
e2       k1       k2       c0       m0       2  
e3       k2       k3       c0       m0       1

- **Federelement**

mit Elementidentifikator, Identifikator für den zugeordneten Elementknoten und Materialidentifikator.

Federelement  
Drehfeder       k1       m1

Der Materialidentifikator für ein Federelement legt einen Identifikator für den Materialtyp (hier: *Drehfeder*) fest und die Federsteifigkeiten in x-, y-Richtung und Verdrehung.

Material  
m1    Drehfeder       0       0       100

- **Punktlast**

mit Identifikator für Punktlast, Elementidentifikator des Elementes, dem die Punktlast zugeordnet ist, Lastwerte in x- und y-Richtung und Angriffsort der Punktlast prozentual zur Elementlänge (hier 50% der Elementlänge).

Punktlast  
P1    Bm10   0       -500    0,5

- **Linienlast**

mit Identifikator für Linienlast, Elementidentifikator des Elementes, dem die Linienlast zugeordnet ist und Lastwerte in x- und y-Richtung am Elementanfang und -ende.

Linienlast  
Linienlast1    Bm10   0       -200    0       -200

## Eingabewerte für dynamische Tragwerksberechnung

folgende weitere Eingabedaten können definiert werden:

- **Eigenlösungen**

mit Identifikator der Eigenlösung und Anzahl der zu berechnenden Eigenlösungen.

*Eigenlösungen*

*2DOFEigen*     2

- **Zeitintegration**

mit Identifikator der Zeitintegration, Maximalzeit der Berechnung, Länge der Zeitschritte, Identifikation der Integrationsmethode (1: Newmark, 2: Wilson Theta und 3: Alfa) und Parameter der jeweiligen Methode (Newmark: *beta* und *gamma*, Wilson: *theta*, Taylor: *alfa*).

*Zeitintegration*

*SechsDOFBodenanregung*     125     0,4     1     0,25     0,5

- **Zeitabhängige Knotenlast**

mit Identifikator der zeitabhängigen Knotenlast, Identifikator des zugeordneten Knotens (oder: *boden* bei Bodenanzregung) und Angabe des Knotenfreiheitsgrades.

*zL0*     *k2*     0

alternativ kann auch eine zeitabhängige Belastung durch Bodenanzregung definiert werden. Hierzu wird als Knotenidentifikator „boden“ definiert und der zugehörige Knotenfreiheitsgrad festgelegt.

*zL1*     *boden*     0

In der Folgezeile wird die zeitabhängige Belastung festgelegt entweder durch

- Einlesen aus Datei (aktivieren des Dateiauswahldialog)

*Datei*

- Eingabe als stückweise lineare Folge von Wertepaaren für <Zeit; Lastwert>

*0;0 0,8;1 1,6;0 3,2;-1 4,8;0 5,6;1 6,4;0*

- Eingabe von Amplitude, Frequenz und Phasenverschiebung für harmonische Belastung

*1 0,03 0*

- **Anfangsbedingungen**

mit Identifikator des zugeordneten Knotens gefolgt von 2 Werten für Anfangsverformung und -geschwindigkeit je Knotenfreiheitsgrad (hier: 1 Freiheitsgrad)

*Anfangsbedingungen*

*k0 1 0*

*k1 0,9 0*

- **Dämpfung**

mit Angabe eines einheitlichen modalen Dämpfungsmaßes für alle Eigenzustände oder unterschiedlicher Dämpfungsmaße für unterschiedliche Eigenzustände

*Dämpfung*

0,02

## Eingabewerte Wärmeberechnung

Die Implementierung von Elementen für Wärmeberechnungen ist beschränkt Elementen mit 2, 3 oder 4 Knoten in 2D und einem Element mit 8 Knoten in 3D, die Anzahl der Knotenfreiheitsgrade ist 1.

- **Elemente2D2Knoten**

erfordert in der Folgezeile die Eingabe eines eindeutigen Elementidentifikators, gefolgt von 2 Knotenidentifikatoren und einem weiteren für den zugeordneten Materialidentifikator.

*Elemente2D2Knoten*

*e0 k00 k01 iso*

- **Elemente2D3Knoten**

erfordert in der Folgezeile die Eingabe eines eindeutigen Elementidentifikators, gefolgt von 3 Knotenidentifikatoren und einem weiteren für den zugeordneten Materialidentifikator.

*Elemente2D3Knoten*

*e0 k00 k01 k02 iso*

- **Elemente2D4Knoten**

erfordert in der Folgezeile die Eingabe eines eindeutigen Elementidentifikators, gefolgt von 4 Knotenidentifikatoren und einem weiteren für den zugeordneten Materialidentifikator.

*Elemente2D3Knoten*

*e0 k00 k01 k02 k03 iso*

- **Elemente3D8Knoten**

erfordert in der Folgezeile die Eingabe eines eindeutigen Elementidentifikators, gefolgt von 8 Knotenidentifikatoren und einem weiteren für den zugeordneten Materialidentifikator.

*Elemente3D8Knoten*

*e0 k00 k01 k02 k03 k04 k05 k06 k07 iso*

- **Material**

erfordert in Folgezeilen die Eingabe eines eindeutigen Identifikators für den Materialtyp, einen Wert für die Wärmeleitfähigkeit und optional einem weiteren Wert für die Materialdichte \* Wärmeleitfähigkeit. Im Fall einer 3D-Berechnung können 3 Werte für die Wärmeleitfähigkeit in x-, y- und z-Richtung angegeben werden.

*Material*

*iso 5 1*

- **KnotenLasten**

erfordert in Folgezeilen die Eingabe eines eindeutigen Identifikators für die Knotenlast, eines weiteren für den zugeordneten Knoten und einen Wert für den Wärmeeintrag.

*Knotenlasten*

*KL0 k5 5*

- **LinienLasten**

erfordert in Folgezeilen die Eingabe eines eindeutigen Identifikators für die Linienlast, zwei weiteren für die zugeordneten Start- und Endknoten und zwei Werten für den linear verteilten Wärmeeintrag am Start- und Endknoten.

*Linienlasten*

LL0    k05    k06    5       10

- **ElementLast3**

erfordert in Folgezeilen die Eingabe eines eindeutigen Identifikators für die Elementlast, einem weiteren für das zugeordneten 3-Knoten-Element und 3 Werten für den linear verteilten Wärmeeintrag an den 3 Elementknoten.

*ElementLast3*

EL0    e2       30       30       30

- **ElementLast4**

erfordert in Folgezeilen die Eingabe eines eindeutigen Identifikators für die Elementlast, einem weiteren für das zugeordneten 4-Knoten-Element und 4 Werten für den linear verteilten Wärmeeintrag an den 4 Elementknoten.

*ElementLast4*

EL1    e5       30       30       30       30

- **Randbedingungen**

erfordert in Folgezeilen die Eingabe eines eindeutigen Identifikators für die Randbedingung, eines weiteren für den zugeordneten Knoten und einem Wert für die vordefinierte Randtemperatur.

*Randbedingungen*

r00    k00    10

## Eingabewerte für instationäre Wärmeberechnungen

werden über folgende weitere Eingabedaten definiert:

- **Eigenlösungen**

mit Identifikator der Eigenlösung und Anzahl der zu berechnenden Eigenlösungen.

*Eigenlösungen*  
*RechenbeispielEigen* 2

- **Zeitintegration**

mit Identifikator der Zeitintegration, Maximalzeit der Berechnung, Länge der Zeitschritte und Parameter *alfa* für das Zeitschrittverfahren 1. Ordnung.

*Zeitintegration*  
*KonstantLinksLinearUnten* 30 0,5 0,8

- **Anfangstemperaturen**

mit Knotenidentifikator (inkl. *alle*) und Temperaturwert oder „stationäre Lösung“

*Anfangstemperaturen*  
*stationäre Lösung*  
*alle* 30

- **Zeitabhängige Randbedingungen**

mit Randbedingungidentifikator und

⇒ mit Knotenidentifikator und *datei* für Temperaturverlauf aus Datei oder

⇒ mit Knotenidentifikator und einem Wert für konstante Temperatur oder

⇒ mit Amplitude, Frequenz und Phasenverschiebung oder

⇒ mit einer Folge von Wertepaaren <Zeit; Temperatur> für einen stückweise linearen Temperaturverlauf

z.B.

*Zeitabhängige Randbedingungen*  
*ZRt0* *k06* *datei*

*Zeitabhängige Randbedingungen*  
*ZRt0* *k06* 0;0 15;20 20;30

- **Zeitabhängige Knotenlast**

mit Knotenlastidentifikator, Knotenidentifikator und Temperaturverlauf mit gleichen Variationen wie unter zeitabhängige Randtemperaturen

- **Zeitabhängige Elementlast**

mit Elementlastidentifikator, Elementidentifikator und Temperaturverlauf mit gleichen Variationen wie unter zeitabhängige Randtemperaturen

## Eingabewerte Elastizitätsberechnung

Die Implementierung von Elementen für Elastizitätsberechnungen ist exemplarisch beschränkt auf 2DElemente mit 3 Knoten und 3DElemente mit 8 Knoten, die Anzahl der Knotenfreiheitsgrade ist 3.

- **Element2D3**

mit Elementidentifikator, 3 Knotenidentifikatoren, 1 Identifikator für den Elementquerschnitt und 1 Identifikator für den Materialtyp.

```
Element2D3
ELower00    n00    n10    n11    thick    planeStress
```

- **Element3D8**

mit Elementidentifikator, 8 Knotenidentifikatoren und 1 Identifikator für den Materialtyp. Die Knotenidentifikatoren sehen jeweils 100 Knoten je Richtung vor, damit diese einfacher generiert werden können - beginnend mit 00-99 in x-Richtung, 00-99 in y-Richtung und 00-99 in z-Richtung.

```
Element3D8
E000000    k000000    k000100    k010100    k010000    k000001    k000101    k010101    k010001    planeStress
```

### 3D8ElementNetz

mit Initial für die Generierung von Elementen mit 8 Knoten in 3D, mit Knoteninitial für generiertes Knotennetz, Anzahl der Elemente in alle 3 Richtungen und Materialidentifikator.

Die Generierung basiert i.d.R. auf einem ebenfalls generiertem Knotennetz (s.o.):

```
Variables Knotennetz
0      1      3      7      15
K      0      0      0
```

hier: Initial für Elementidentifikatoren, Initial für Knotenidentifikatoren, je 4 Elemente in x,y,z und Material iso

```
3D8ElementNetz
E      K      4      iso
```

- **Querschnitt**

mit Identifikator für den Elementquerschnitt und 1 Wert für die Elementdicke.

```
Querschnitt
thick    1
```

- **Material**

mit Identifikator für das Elementmaterial, 1 Wert für den Elastizitätsmodul und 1 Wert für die Querdehnungszahl.

```
Material
planeStress    3e7    0
```



- **Knotenlasten**

mit Identifikator für die Knotenlast, Identifikator für den zugeordneten Knoten und 2 oder 3 Werten für den Lastwert in x,y und z.

*Knotenlasten*

P1 k62 0 -10000

- **Linienlasten**

mit Identifikator für die Linienlast, Identifikator für den 1. zugeordneten Knoten und 2 Lastwerte in x,y, und Identifikator für den 2. zugeordneten Knoten und 2 Lastwerte in x,y.

*Linienlasten*

L1 k62 0 -10000 k42 0 -10000

- **Randbedingungen und RandbedingungFläche**

an einem Knoten mit Identifikator für die Randbedingung, 1 Knotenidentifikator, Kennzeichner für Festhaltungen (z.B. xy) und vordefinierte Verformungswerte in x- und y-Richtung.

*Randbedingungen*

Rk00 k00 xy 0 0

Randbedingungen werden im 3dimensionalen Raum in der Regel für ganze Randflächen definiert und werden als solche ebenfalls generiert.

Die Randbedingungen an den Symmetrieflächen für die Festhaltung in Richtung x an der Fläche x=0 und in Richtung y an der Fläche y=0 werden wie folgt definiert. Hierbei steht das Initial F für die generierten Identifikatoren der Festhaltungen, die Initialen X0 und Z0 für die generierten Identifikatoren der Symmetrieflächen, das Initial K für die generierten Knotenidentifikatoren, der Wert 5 für die Anzahl der generierten Knoten in eine Richtung und der Kennzeichner x bzw. y für die Richtung der Festhaltung.

*RandbedingungFläche*

F X0 K 5 x

F Y0 K 5 y

An den Außenrändern des elastischen Halbraums gelten die Boussinesq-Lösungen. Hierbei wird in der ersten Zeile die Anzahl der Knoten in eine Richtung und ihre wachsenden Knotenkoordinaten festgelegt. In den Folgezeilen bezeichnet B eine Boussinesq-Lösung, der folgende Kennzeichner die Maximalfläche, K das Initial für die generierten Knoten und schließlich der Kennzeichner für die Richtung der Festhaltung.

*RandbedingungBoussinesq*

0 1 3 7 15

B XMax K x

B YMax K y

B ZMax K z

## Beispieldaten, ausführbares Programm, Quellen und Dokumentation

Im Verzeichnis „FE Berechnungen-App“ befindet sich die Applikation „FE Berechnungen.application“ mit der zugehörigen „setup.exe“. Ferner befindet sich dort ein Unterverzeichnis „input“ mit Beispieldaten für jedes Anwendungsgebiet und es gibt noch ein Unterverzeichnis „Dokumentation“ mit diesem und weiteren Dokumenten.

Im Verzeichnis „FE Berechnungen Quellen“ befinden sich alle Programmquellen und ein Projekt („FE Berechnungen.sln“) als „solution“ in VisualStudio.

### Beispieldaten Tragwerksberechnungen

Im Verzeichnis „input/Tragwerksberechnung“ sind einige, einfache Beispiele für

- Fachwerk
- Gebäude
- Gelenkrahmen
- Rahmen
- Rahmen mit Feder
- Schornstein und
- Zweifeldrahmen

Im Verzeichnis „input/Tragwerksberechnung/**Dynamik**“ sind einige, einfache Beispiele für

- Fachwerk
- Schornstein mit Bodenanregung (Beschleunigungsdaten El Centro Erdbeben aus Datei)
- Schornstein mit Bodenanregung (Beschleunigungsdaten als Zeit-Beschleunigungs-Wertepaare aus Datei)
- einfaches System mit 6 Freiheitsgraden (Beschleunigungsdaten als Zeit-Beschleunigungs-Wertepaare)
- einfaches System mit 6 Freiheitsgraden mit Anfangsverformung

Im Verzeichnis „input/Tragwerksberechnung/Dynamik/**Anregungsdateien**“ sind Beschleunigungsdaten für

- BM68elc (Messdaten El Centro Erdbeben)
- LineareIntervalle (Zeit-Beschleunigungs-Wertepaare aus Datei mit linearer Interpolation)

## Beispieldaten Wärmeberechnungen

Im Verzeichnis „input/Wärmeberechnung“ sind einige, einfache Beispiele für

- Rechenbeispiel
- kamin.2D (Industrieschornstein)
- Wandecke

Im Verzeichnis „input/Wärmeberechnung/**instationär**“ sind einige, einfache Beispiele für

- einfaches System mit drei Freiheitsgraden unter Anfangstemperaturen an 2 Knoten
- einfaches System mit drei Freiheitsgraden unter sinusförmiger Knotenlast
- einfaches System mit drei Freiheitsgraden unter sinusförmiger Einwirkung am Rand
- Rechenbeispiel mit stationärer Lösung als Anfangsbedingung, zeitabhängigen Randtemperaturen und zeitabhängiger Elementlast
- kamin.2D (Industrieschornstein mit Anfahrkurve für Wärmebelastung)
- VierDOFInitial (3 Stabelemente mit Anfangsbedingungen an einem Knoten)
- Wandecke mit stationärer Lösung als Anfangsbedingung und zeitabhängigen Randtemperaturen

Im Verzeichnis „input/Wärmeberechnung/instationär/**Anregungsdateien**“ sind Temperaturdaten für

- KaminAnlauf.0.50inSec (Anfahrkurve für Industrieschornstein)
- KaminKaltstart (Industrieschornstein mit Kaltstart)

## **Beispieldaten Elastizitätsberechnung**

Im Verzeichnis „input/Elastizitätsberechnung“ sind einige, einfache Beispiele für

- Halbraum (Elastizitätsberechnung in 3D für elastischen Halbraum mit Boussinesq-Bedingungen an den Rändern)
- Kragarm (Elastizitätsberechnung in 2D)