Software for Solving Problems in   
Structural Engineering

with the Finite Element Analysis

FE Analysis

Prof. Dr.-Ing. Karl E. Beucke, Ettersburg

2021

**Goals and Motivation**

1. **Structural Engineering Education**

Most curricula in structural engineering currently will require students to learn a number of different, specialized analysis techniques for the analysis of the behavior of structures (strength, deformation, heat transfer, etc.). These can be carried out „by-hand“ or with some simple, general software tools (e.g. spreadsheets, math tools). Most of these have a very limited scope but will allow students to follow and understand each individual step in the solution process. Classical solution techniques that can be carried out „by-hand“ or with simple, general software tools are still relevant but only for general estimates of the behavior of structures or for checking of results produced by software systems.

Later, in professional problem solving however, engineers are confronted with highly complex and complicated software systems for the solution of such problems. These systems normally have a general scope with the aim of solving even the most demanding problems in structural engineering.

The objectives of educating students in structural engineering should be for students to develop an understanding of the behavior of structures under various external influences. This requires tools that will inform students about the consequences of external influences on structures and of changes to structural design. At best, this should be an immediate feedback. Instead, most tools today will tie up students with the mechanics of preparing input data - generating a digital model - and analyzing output results. Both are often tedious and time consuming processes. In consequence, digital models are often set up only once and never changed fundamentally in the process, i.e. the model is hardly ever adjusted but rather the design has to account for necessary requirements in strength.

Professional software systems are generally only available in „executable“ formats that are „black boxes“ for the users. It is highly difficult if not impossible to know and understand how the solution process functions, what implications are involved and what results of intermediate steps in the solution process are available. The user is left with „feeding the black box with data“ and has to live with what results are produced and what quality of results is produced.

This will specifically restrict - or even prevent - such software systems for use in education. Currently, this problem is often addressed by requiring engineering software to be available in human-readable source code. There were even initiatives that demanded that ONLY computer software was acceptable that was also available in a readable format (Open Source Initiative). However, with modern software systems being of enormous extent and complexity even this requirement will not solve the problem of transparency and demonstrability.

There exists a considerable gap between simple and demonstrable solution techniques that can be used in engineering education and the utilization of complex software systems in professional engineering problem solving. Either educational examples are being used that are transparent, can be followed openly by students and demonstrate the solution process or professional software systems are being utilized that are totally nontransparent in structure, solution algorithm and functionality and that are often even protected by license restrictions.

The software „FE Analysis“ was written totally from scratch in order to close the above mentioned gap. Written primarily for structural engineering education, the software does not depend on ANY third party functionality so that the COMPLETE source code is available on a totally transparent and open basis. This allows students to not just utilize the software but rather to dig into any detail of solution processes. By using a free Integrated Development Environment (IDE) like Visual Studio in debugging mode, students can have the functioning of each individual step in the solution of structural engineering problems demonstrated and can monitor intermediate results. Capabilities and limitations of solution algorithms can easily be inspected and taken into account.

With these goals in mind, it is the aim of the software to offer

* a general solution for solving complex problems in structural engineering that
* allows users a complete, unrestricted insight into functionality and functioning of any solution algorithms and approach and that
* supports solving fundamental classes of problems in stationary or transient 1st order analysis (e.g. heat transfer) and problems in static or dynamic 2nd order analysis (strength of materials) for 2-dimensional as well as 3-dimensional anlysis.

1. **Transparency, Openness and Demonstrability**

Zu diesem Zweck wird ein - nicht unumstrittener - Weg gewählt für Implementierung dieser Lösungen. 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 Entwicklungsumgebungen 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 (veröffentlicht im Jahre 2001) der Fa. Microsoft

* alle Konzepte der **Objektorientierung** konsequent umsetzt,
* über eine moderne Nutzeroberfläche verfügt, die als **Windows Presentation Format (WPF)** voll in die Windows Umgebung integriert ist und die
* mit **Visual Studio** durch eine moderne, professionelle Entwicklungsumgebung (IDE) unterstützt wird, welche 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.

1. **Schulung von Studierenden für den Entwurf von Tragwerken**

Studierende im Fach Bauingenieurwesen sind häufig voll gefordert mit dem Lernen und der Ausführung von Berechnungen *vorgegebener* Tragwerksmodellierungen. Ein einmal festgelegter Entwurf wird idealisiert und entweder „von Hand“ oder mittels einer Softwarelösung berechnet. Dies geschieht z.B. durch Idealisierungen wie Biegebalken oder Rahmentragwerk mit Auflagerbedingungen und Lasteinwirkungen. Da diese Berechnungen 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.

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.

Folglich war es in der Regel viel zu aufwändig, um Änderungen im Tragwerksentwurf oder Entwurfsvarianten zu berücksichtigen. Manchmal wurde es so formuliert, dass Bauingenieure lieber einen bestehenden Entwurf „heilig sprechen“ und „gesund rechnen“ anstatt aufwändig 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:

* einen Tragwerksentwurf mit wenig Aufwand zu formulieren,
* den Entwurf im Sekundenbereich zu berechnen und
* die für den Entwurf wesentlichen Rechenergebnisse unmittelbar 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* Entwurfs, sondern auf dem „Durchspielen“ unterschiedlicher Entwurfsvarianten und auf der Bewertung und Beurteilung unterschiedlicher Entwurfsvarianten.

Die vorliegende Software hat den Anspruch, ein solches Vorgehen konzeptionell zu unterstützen. Der Umfang der Lösungen müsste allerdings noch kontinuierlich erweitert werden.

1. **Konzeptstudie für einen Integrativen Tragwerksentwurf**

In der gegenwärtigen Praxis der Bauingenieure gilt die unter Punkt 2 beschriebene Vorgehensweise noch verschärft, da praxisrelevante Tragwerksentwürfe in der Regel hochkomplex und und hoch umfangreich sind. 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 Ingenieur. 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.

**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.

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

**input Elastizitätsberechnung  
 Tragwerksberechnung und  
 Wärmeberechnung**

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 modellbestimmenden Daten zur Verfügung.

Sämtliche Daten, die für ein bestimmtes Modell definiert sind, können in alfanumerischer, tabellarischer Form ausgegeben und angezeigt werden (**tabellarische Modelldarstellung**). Jede Zeile kann in einer Ausgabetabelle angewählt 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 unterhalb der bestehenden Tabellenzeilen 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 angewählt werden, woraufhin sich dann an der Position des Klicks ein sogenanntes „Popup-Fenster“ öffnet mit den zugehörigen Modellinformationen zu den angwählten 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. Lagerrektionen) 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>

**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, d.h alfanumerische Ausgabe in Tabellenform, und visualisieren
* Berechnung ausführen
* Berechnungsergebnisse anzeigen und visualisieren

ggf.

* zeitabhängige Modelldaten anzeigen
* Eigenlösungen berechnen, anzeigen und visualisieren
* Zeitverlaufsberechnung
* zeitabhängige Ergebnisse anzeigen
* zeitabhängige Modellzustände visualisieren
* Kontenzeitverläufe visualisieren

Zur Verfügung stehen einerseits **statische bzw. stationäre Berechnungsverfahren** und je nach Anwendungsgebiet auch **dynamische bzw. instationäre Zeitschrittverfahren**.

Zudem kann die Eigenlösung der eingelesenen physikalischen Systeme bestimmt werden, d.h. die **Eigenwerte und Eigenvektoren**.

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 i**nteraktiv 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. Wesentlich ist dabei, dass die Modelldaten im Hintergrund konsistent angepasst werden.

Die **Berechnungsergebnisse** können entweder **alfanumerisch in Tabellenform** angezeigt werden oder separat 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.

Identifikatoren kennzeichnen jeweils einen Bereich von Eingabewerten, die zeilenweise festgelegt werden. Unterschiedliche Eingabe**bereich**e kennzeichnen thematisch zusammenhängende Eingaben. Das Ende eines Eingabe**bereich**s ist durch eine Leerzeile festgelegt.

Einzelne, unterschiedliche Eingabe**werte** 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 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.

Der Identifikator ***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.

Der Identifikator ***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*

*k0 0 0  
k1 6 0*

Der Modellname ist definiert als „Stockwerkrahmen“. Die Raumdimension ist definiert mit *2*D 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.

In **1D** werden 12 Knoten N000000 bis N000011 von x=0. mit Inkrement 2. generiert durch:

*Aequidistantes Knotennetz*

*N 0. 2. 12*

In **2D** wird ein Netz von 3\*3 Knoten mit gleichem Knotenabstand definiert durch  
 3 Knoten A000000 bis A000002 von x=0. und y=1. mit Inkrement 1.,  
 3 Knoten A001000 bis A001002 von x=1. und y=1. mit Inkrement 1. und weitere  
 3 Knoten A002000 bis A002002 von x=2. und y=1. mit Inkrement 1.  
 generiert durch:

*Aequidistantes Knotennetz*

*A 0. 1. 3 1. 1. 3*

In **3D** wird ein Netz von 3\*3\*3 Knoten definiert durch jeweils 3 Knoten:  
 Z000000 bis Z000002 von x=0., y=1. und z=2. mit Inkrement 1.,  
 Z000100 bis Z000102 von x=0., y=2. und z=2. mit Inkrement 1.,  
 Z000200 bis Z000202 von x=0., y=3. und z=2. mit Inkrement 1.,  
 Z010000 bis Z010002 von x=1., y=1. und z=2. mit Inkrement 1.,  
 Z010100 bis Z010102 von x=1., y=2. und z=2. mit Inkrement 1.,  
 Z010200 bis Z010202 von x=1., y=3. und z=2. mit Inkrement 1.,  
 Z020000 bis Z020002 von x=2., y=1. und z=2. mit Inkrement 1.,  
 Z020100 bis Z020102 von x=2., y=2. und z=2. mit Inkrement 1.,  
 Z020200 bis Z020202 von x=2., y=3. und z=2. mit Inkrement 1.  
 generiert durch:

*Aequidistantes Knotennetz*

*Z 0. 1. 3 1. 1. 3 2. 1. 3*

Eine **Variables Knotennetz** ist definiert durch eine Zeile mit regelmäßigen Knotenabständen und einer weiteren Zeile mit einem Anfangsidentifikator und dem Koordinatenursprung

*Variables Knotennetz*

*0. 1. 3. 6.*

*X 0. 0.*

**Eingabewerte Tragwerksberechnung**

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

Gelenke in der Tragstruktur, d.h. 2 Knotenfreitsgrade, werden vor der Knotendefinition festgelegt.  
  
***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 zwei weiteren Werten für die zugeordnete Querdehnungszahl und die spezifische Masse.

*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, einen Wert für den zugeordnete Knoten und eine Zeichenfolge aus xyr für die Festhaltungen in x-, y-Richtung und die Rotation.

*Lager*

*Lager0 F0 xy*

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

weitere Identifikatoren für **Elemente** in der Tragwerksberechnung umfassen:

* ***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 EIc 2*

*e3 k2 k3 c0 EIc 1*

* ***Federelement***

mit Elementidentifikator, Identifikator für den zugeordneten Elementknoten und Materialidentifikator.

*Federelement*

*Drehfeder k1 m2*

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

*Material*

*m2 feder 0 0 100*

weitere Identifikatoren für **Lasten** in der Tragwerksberechnung umfassen:

* ***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*

**Dynamische Tragwerksberechnungen** werden über folgende weitere Eingabedaten definiert:

* ***Eigenlösungen***

mit Identifikator der Eigenlösung und Anzahl der zu berechnenden Eigenlösungen.

*Eigenlösungen*

*2DOFEigen 2*

* ***Zeitintegration***

mit Identifikator der Zeitintegration, Maximazeit 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* oder Wilson: *theta* oder *alfa*).

*Zeitintegration*

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

* ***Zeitabhängige Knotenlast***

mit Identifikator der zeitabhängigen Knotenlast, Identifikator des zugeordneten Knotens und Angabe des Knotenfreiheitsgrades.

*zL0 k2 0*

In der Folgzeile 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*

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

*zL1 boden 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 Knotens oder alle Knoten (hier: alle) und Angabe der Dämpfungsmaße je Freiheitsgrad

*Dämpfung*

*all 0,02 0*

**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.

*Elemente2D3Knoten*

*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.

*Elemente2D3Knoten*

*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 k5 k6 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*

**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 „*stationaere Loesung*“

*Anfangstemperaturen*

*stationäre Loesung*

* ***Zeitabhängige Randtemperaturen***

- mit Knotenidentifikator und Dateinamen für Temperaturverlauf oder  
 - mit Knotenidentifikator und einem Wert für konstante Temperatur oder  
 - mit Amplitude, Frequenz und Phasenverschiebung oder  
 - mit einer Reihe von Wertepaaren für <Zeit; Temperatur> für einen stückweise linearen  
 Temperaturverlauf

z.B.

*Zeitabhängige Randtemperaturen*

*ZRt0 k06 datei*

*Zeitabhängige Randtemperaturen*

*ZRt0 k06 0 0 15 20 20 30*

* ***Zeitabhängige Knotenlast***

mit Knotenidentifikator 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 Materialtyyp.

*Element2D3*

*ELower00 n00 n10 n11 thick planeStress*

* ***Element3D8***

mit Elementidentifikator, 8 Knotenidentifikatoren und 1 Identifikator für den Materialtyyp. 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*

* ***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, 1 Knotenidentifikator und 2 Lastwerten für die Lastwerte in x- und y-Richtung

*Knotenlasten*

*P k62 0 -1e5*

***Randbedingungen***

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*

In **3D** ist es sinnvoll, ein Knotennetz zu generieren und auf dessen Basis ein Elementnetz zu generieren. Beispielhaft soll dies an einem elastischen Halbraum gezeigt werden.

Für einen Quadrant eines elastischen Halbraums sieht das Beispiel wie folgt aus:

*Variables Knotennetz*

*0 1 3 7 15*

*K 0 0 0*

*3D8ElementNetz*

*E K 4 iso*

* ***Material***

mit Identifikator für das Elementmaterial, 1 Wert für den Elastizitätsmodul (E-Modul), 1 Wert für die Querdehnungszahl und 1 Wert für den Schubmodul (G-Modul).

*Material*

*iso 2,4 0,2 1*

* ***Knotenlasten***

mit Identifikator für die Knotenlast und je einen Wert für die Last in x-, y- und z-Richtung.

*Knotenlasten*

*XLOAD K000000 0,25 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 Bousinesq-Läsung, 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 Programm“ befindet sich die Applikation „FE Berechnungen.application“ mit der zugehörigen „setup.exe“. Ferner befindet sich dort ein Verzeichnis „input“ mit Beispieldaten für jedes Anwendungsgebiet. Dort befindet ich auch ein Verzeichnis mit allen Programmquellen und einem Projekt („FE Berechnungen.sln“) als „solution“ in VisualStudio. Schließlich gibt es noch ein Verzeichnis „Dokumentation“ mit diesem und weiteren Dokumenten.

**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
* sasol.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
* sasol.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

* \_SasolAnlauf.0.50inSec (Anfahrkurve für Industrieschornstein)
* \_SasolKaltstart (Industrieschornstein mit Kaltstart)

**Beispieldaten Elastizitätsberechnung**

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

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