



TECHNISCHE UNIVERSITÄT  
BERGAKADEMIE FREIBERG

Die Ressourcenuniversität. Seit 1765.

Fakultät für Maschinenbau, Verfahrens- und Energietechnik  
Institut für Mechanik und Fluidodynamik  
Technische Mechanik - Dynamik



Studiengang: **Master Fahrzeugbau**

THEMA DER PROJEKTARBEIT

## **Modalanalyse mit Hilfe der Finite-Elemente-Methode**

THEMA IN ENGLISCHER SPRACHE

## **Modalanalysis with the Finite-Element-Method**


bearbeitet von:

**Sun, Qian**

Übergabetermin des Themas:  
Abgabetermin der Projektarbeit:

**09.11.2018  
08.05.2019**

Prüfer:



**Prof. Dr. A. Ams**



---

## Erklärung

Hiermit versichere ich, die vorliegende Arbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Freiberg, den \_\_\_\_ . \_\_\_\_ . 20 \_\_\_\_

---

(Vorname Name)





Ausschreibung einer Projektarbeit vom 6. November 2018

# Modalanalyse mit Hilfe der Finite-Elemente-Methode

**Aufgabenstellung:** Im Rahmen der ausgeschriebenen Arbeit ist ein Programm zur numerischen Modalanalyse mittels der Finite-Elemente-Methode zu entwickeln.

**Teilaufgaben:**

1. Einarbeitung in das Thema Modalanalyse, Finite-Elemente-Methode
2. Programmierung der Modalanalyse mit FEM für ausgewählte Beispiele (wahlweise mit MATLAB® oder MATHEMATICA®)
3. Vergleich der Ergebnisse mit der kommerziellen FEM-Software ANSYS®
4. Schriftliche Dokumentation der Arbeit

**Kontakt:** Prof. Dr.-Ing. A. Ams (WEI 129, Tel. 39-3109), R. Schmidt (WEI 214, Tel. 39-4141,  
Rico.Schmidt@imfd.tu-freiberg.de)



## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Theoretische Grundlagen</b>	<b>3</b>
2.1	Grundlagen der Finite-Elemente-Methode . . . . .	3
2.2	Prinzip von Hamilton . . . . .	4
<b>3</b>	<b>Finite-Elmenete und Matrizen</b>	<b>7</b>
3.1	Stabelemente . . . . .	7
3.1.1	Linearer Ansatz . . . . .	8
3.1.2	Quadratischer Ansatz . . . . .	9
3.2	Balkenelemente . . . . .	10
3.2.1	Kubischer Ansatz . . . . .	11
3.3	Plattenelemente . . . . .	12
3.3.1	Konforme Plattenelemente . . . . .	13
<b>4</b>	<b>Ergebnisse</b>	<b>15</b>
4.1	Längsschwingungen eines Stabs . . . . .	15
4.2	Biegeschwingung eines Balkens . . . . .	15
4.3	Biegeschwingungen einer Platte . . . . .	16
<b>5</b>	<b>Zusammenfassung</b>	<b>19</b>
	<b>Quellenverzeichnis</b>	<b>21</b>
	<b>Anhang A: Programm Stabelemente</b>	<b>A-1</b>
	<b>Anhang B: Programm Balkenelemente</b>	<b>B-1</b>
	<b>Anhang C: Programm Plattenelemente</b>	<b>C-1</b>





## Symbol- und Abkürzungsverzeichnis

### Lateinische Notation

$A$	$\text{m}^2$	Querschnittfläche
$b$	$\text{m}$	Breit des Querschnitts
$B$	$\text{m}$	Breit der Platte
$E$	$\text{N}/\text{mm}^2$	E-Modul
$E_{kin}$	$\text{J}$	kinetische Energie
$E_{pot}$	$\text{J}$	potentielle Energie
$f$	$\text{Hz}$	Eigenfrequenz
$h$	$\text{m}$	Höhe des Querschnitts
$H$	$\text{m}$	Dicke der Platte
$i, k$	—	Zählvariablen
$I$	$\text{m}^4$	Flächenträgheitsmoment
$l$	$\text{m}$	Länge
$l_e$	$\text{m}$	Länge eines Elements

### Griechische Notation

$\varepsilon$	—	Dehnung
$\lambda$	—	Eigenwert
$\nu$	—	Poissionzahl
$\rho$	$\text{kg}/\text{m}^3$	Dichte

### Vektoren und Matrizen

$\mathbf{D}$	Dämpfungsmatrix
$\mathbf{K}$	Steifigkeitsmatrix
$\mathbf{M}$	Massenmatrix
$\mathbf{K}_e$	Steifigkeitsmatrix eines Elements
$\mathbf{M}_e$	Massenmatrix eines Elements
$\vec{N}$	Vektor der Formfunktion
$u(x, t)$	Längsverschiebung
$w(x, t)$	Durchbiegung

**Indices**

$(\quad)_{,t}$	Erste Ableitung nach der Zeit
$(\quad)_{,tt}$	Zweite Ableitung nach der Zeit
$(\quad)_{,x} ; (\quad)_{,y} ; (\quad)_{,z}$	Erste Ableitung in Richtung x,y,z
$(\quad)_{,xx} ; (\quad)_{,yy} ; (\quad)_{,zz}$	Zweite Ableitung in Richtung x,y,z

## 1 Einleitung

Die Modalanalyse stellt die Untersuchung der dynamischen Eigenschaften von Systemen im Frequenzbereich dar. Beispiele hierfür sind das Messen der Vibrationen einer Fahrzeugkarosserie, wenn diese an einem Rüttler angebracht ist, oder das Rauschmuster in einem Raum, wenn dieser von einem Lautsprecher angeregt wird. Im konstruktiven Ingenieurbau verwendet die Modalanalyse die Gesamtmasse und Steifigkeit einer Struktur, um die Eigenfrequenzen zu ermitteln.

Die Eigenfrequenzen sind im konstruktiven Ingenieurbau sehr wichtig, da es unerlässlich ist, dass diese nicht mit den erwarteten Erregungsfrequenzen übereinstimmen (Resonanz). Falls es trotzdem zur Resonanz kommt, kann das Objekt strukturellen Schaden erleiden. Beispielsweise wird bei Bauwerken wie Brücken versucht, die Eigenfrequenzen von den Frequenzen der auf der Brücke gehenden Personen fernzuhalten. Andere natürliche Anregungen können auch vorhanden sein und die Eigenmoden einer Brücke erregen (z.B.: Anregung durch Wind).

Aus methodologischer Sicht ist die Modalanalyse in numerische und experimentelle Modalanalyse zu unterteilen. Wenn die Modalparameter beispielsweise durch die Finite-Elemente-Methode erhalten werden, wird sie als numerische Modalanalyse bezeichnet. Bei der experimentellen Modalanalyse werden Anregung und Antwort über Sensoren gemessen. Anschließend werden die modalen Parameter durch Parameteridentifikation analysiert.

In dieser Arbeit wird die numerische Modalanalyse mit der Finite-Elemente-Methode betrachtet. Die Finite-Elemente-Methode ist ein allgemeines, bei unterschiedlichen physikalischen Aufgabenstellungen angewendetes numerisches Verfahren. Typische Anwendungsbereiche sind Strukturanalyse, Wärmeübertragung, Fluidodynamik, Massentransport und elektromagnetisches Potential. Der Berechnungsbereich ist in viele Teile mit einfacher Form unterteilt z.B. Quadrate oder Tetraeder. Sie sind die „finiten Elemente“. Ihr physikalisches Verhalten lässt sich aufgrund ihrer einfachen Geometrie mit bekannten Ansatzfunktionen gut berechnen. Das physikalische Verhalten des gesamten Körpers wird dadurch wiedergegeben, wie diese Elemente auf die Kräfte, Belastungen und Randbedingungen reagieren und wie sich Belastungen und Reaktionen beim Übergang von einem Element zum Nachbarn durch bestimmte problembezogene Steifigkeitsbedingungen ausbreiten, die die Ansatzfunktion erfüllen müssen.

Darüber hinaus enthalten die Ansatzfunktionen die Parameter, die normalerweise eine physikalische Bedeutung haben, z. B. das Verschieben eines bestimmten Punkts in der Komponente zu einem bestimmten Zeitpunkt. Die Suche nach der Bewegungsfunktion wird somit auf die Suche nach den Werten der Parameter der Funktionen zurückgeführt. Durch die Verwendung von immer mehr Parametern (z. B. mehr und mehr kleinere Elemente) oder Ansatzfunktionen mit immer höherem Wert kann die Genauigkeit der der Näherungslösung verbessert werden.

Um die Bewegungsgleichungen eines Körpers zu gewinnen, wird das Prinzip von HAMILTON benutzt, das auch als das Prinzip der stationären Wirkung bezeichnet wird. Anschließend wird die FE Simulation mit MATLAB<sup>®</sup> beschrieben. Außerdem werden die FE-Software ANSYS<sup>®</sup> sowie analytische Lösungen nach [SP95] zum Vergleichen der Ergebnisse benutzt.



## 2 Theoretische Grundlagen

### 2.1 Grundlagen der Finite-Elemente-Methode

Die Finite-Elemente-Methode (FEM) ist eine numerische Methode zur Lösung von Problemen der Ingenieurwissenschaften und der mathematischen Physik. FEM ist eine leistungsstarke, häufig verwendete Berechnungsmethode, die zu einem System algebraische Gleichungen führt. Die Methode nähert sich der unbekannten Funktion über das Grundgebiet an. Um das Problem zu lösen, wird ein großes System in kleinere, einfachere Teile unterteilt, die als finite Elemente bezeichnet werden. Die einfachen Gleichungen, die diese finiten Elemente modellieren, werden dann zu einem größeren Gleichungssystem zusammengefügt, das das gesamte Problem beschreibt. Die FEM verwendet Methoden aus der Variationsrechnung, um eine Lösung anzunähern, indem der zugehörige Fehler minimiert wird.

Die Unterteilung des ganzen Gebiets in einfachere Teile hat mehrere Vorteile:

- Darstellung von komplexen Geometrien,
- Einbeziehung unterschiedlicher Materialeigenschaften,
- Einfache Darstellung der Gesamtlösung,
- Erfassung von lokalen Auswirkungen.

Ein typischer Vorgang des Verfahrens umfasst:

1. Das Unterteilen (*Diskretisieren*) des ganzen Grundgebiets des Problems in eine Sammlung von einfachen Teilgebieten, die sogenannten finiten Elemente, wobei jedes Teilgebiet durch eine Reihe von Elementgleichungen für das ursprüngliche Problem repräsentiert wird.
2. Systematische Rekombination aller Elementgleichungen zu einem globalen Gleichungssystem für die endgültige Berechnung. Das globale Gleichungssystem hat eine bekannte Lösungsmethode und kann aus den Anfangswerten des ursprünglichen Problems berechnet werden, um eine numerische Lösung zu erhalten.

Im ersten Schritt handelt es sich bei den Elementgleichungen um einfache Gleichungen, die sich lokal den zu untersuchenden ursprünglichen komplexen Gleichungen annähern, wobei die ursprünglichen Gleichungen oft partielle Differentialgleichungen sind.

Die Finite-Elemente-Methode diskretisiert das Grundgebiet in einen Gesamtbau endlicher Finite-Elemente. Die Ecken der Finite-Elemente werden Knoten genannt. Diese Knoten bilden die diskrete Untergruppe für die numerische Methode. Auf den Elementen werden Annäherungsfunktionen eingeführt, die die begrenzte anstehenden und unbekannten Knotengrößen als Parameter enthalten. Die resultierenden Elementintegrale werden mit numerischer Quadratur berechnet. Die Elementgleichungen werden durch Kontinuitätsanforderungen an den Elementgrenzen zusammengesetzt. Auf diese Weise werden Randwertprobleme für lineare partielle Differentialgleichungen in ein lineares Gleichungssystem mit symmetrischen Systemmatrizen übertragen. Für nichtlineare Differentialgleichungen ist der Algorithmus analog mit dem Unterschied, dass nichtlineare Abhängigkeiten unter Verwendung geeigneter Verfahren (z.B. NEWTON-Verfahren) iterativ linearisiert werden und das lineare Gleichungssystem in jedem inkrementellen Schritt eingerichtet wird.

Für bestimmte Aufgaben ist die Unterteilung in Elemente durch das Problem bereits weitgehend vorbestimmt, beispielsweise bei räumlichen Fachwerken, bei denen die einzelnen Stäbe die Konstruktionselemente bilden. Bei zweidimensionalen Problemen wird das Grundgebiet z. B. in Dreiecke oder Vierecke

unterteilt. Räumliche Probleme werden behandelt, indem das dreidimensionale Gebiet in Tetraederelemente, Quader oder andere an das Problem angepasste Elemente, möglicherweise auch gekrümmt, unterteilt wird.

Im zweiten Schritt wird ein globales Gleichungssystem aus den Elementgleichungen durch eine Transformation der Koordinaten von den lokalen Knoten der Teilgebiete zu den globalen Knoten des Gesamtgebiets generiert. Diese räumliche Transformation enthält geeignete Abbildungsmatrizen, die in Bezug auf das Koordinatensystem angewendet werden.

In jedem der Elemente wird für die gesuchte Funktion, bzw. allgemeiner für die das Problem beschreibenden Funktionen, ein problemgerechter Ansatz gewählt. Im Besonderen eignen sich dazu ganze rationale Funktionen in den unabhängigen Raumkoordinaten. Für eindimensionale Elemente (Stäbe, Balken) kommen Polynome ersten, zweiten, dritten und gelegentlich sogar höheren Grades in Frage. Bei zweidimensionalen Problemen finden lineare, quadratische oder höhergradige Polynome Verwendung [Sch91].

Strukturmechanische FEM-Systeme werden durch lineare Gleichungssysteme 2. Ordnung

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{D}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{P}(t) \quad (2.1)$$

dargestellt.  $\mathbf{M}$ ,  $\mathbf{D}$  und  $\mathbf{K}$  sind die Massen-, Dämpfungs- und Steifigkeitsmatrix des Systems und  $\mathbf{P}(t)$  ist der Vektor der externen Kräfte, die auf das System wirken. Der Vektor der Freiheitsgrade ist mit  $\mathbf{u}(t)$  gegeben. Um die verschiedenen Matrizen zu berechnen, werden unterschiedliche physikalische oder energetische Prinzipien verwendet, wie z. B. in dieser Arbeit das Prinzip von HAMILTON.

## 2.2 Prinzip von Hamilton

In der Physik ist das Prinzip von HAMILTON der Ausdruck des 1833 vom irischen Physiker WILLIAM HAMILTON veröffentlichten Prinzips. Das Prinzip von HAMILTON besagt, dass die Dynamik eines physikalischen Systems mittels Variationsrechnung einer einzigen Funktion, der LAGRANGE-Funktion welche alle physikalischen Informationen über das System und die darauf einwirkenden Kräfte enthält, bestimmt wird.

Die LAGRANGE-Funktion, auch LAGRANGE-Größe genannt, enthält alle physikalischen Konnotationen dieses physischen Systems. Für ein Partikelsystem mit einem generalisierten Potential und Zwangsbedingungen lautet die LAGRANGE-Funktion

$$L = E_{kin} - E_{pot}, \quad (2.2)$$

wobei  $E_{kin}$  die gesamte kinetische Energie und  $E_{pot}$  die potentielle Energie des Systems sind.

Die kinetische Energie eines Objekts ist die Energie, die es aufgrund seiner Bewegung besitzt. Diese Energie wird als die Arbeit definiert, die erforderlich ist, um einen Körper einer bestimmten Masse von der Ruhe auf seine vorgegebene Geschwindigkeit zu beschleunigen. Nachdem der Körper diese Energie während seiner Beschleunigung gewonnen hat, erhält er diese kinetische Energie aufrecht, wenn sich seine Geschwindigkeit nicht ändert. Die gleiche Arbeit wird vom Körper geleistet, wenn er von seiner aktuellen Geschwindigkeit in einen Ruhezustand verlangsamt wird.

In der klassischen Mechanik hängt die kinetische Energie eines Punktoobjekts oder eines nicht-rotierenden starren Körpers von der Masse des Körpers und seiner Geschwindigkeit ab.

Übliche Arten von potentieller Energie umfassen die potentielle Schwerkraftenergie eines Objekts, die von seiner Masse und seinem Abstand vom Massenschwerpunkt eines anderen Objekts abhängt, der elastischen potentiellen Energie einer ausgedehnten Feder und der elektrischen potentiellen Energie.

Das Prinzip von HAMILTON ist auch ein wichtiges Variationsprinzip in der Elastodynamik. Im Gegensatz zu einem System, das aus starren Körper besteht, haben verformbare Körper unendlich viele Freiheitsgrade und besetzen ununterbrochene Raumregionen. Der Zustand des Systems wird daher durch kontinuierliche Funktionen von Raum und Zeit beschrieben. Das erweiterte Prinzip von HAMILTON für solche Körper wird durch

$$\delta \int_{t_1}^{t_2} (E_{kin} - E_{pot}) dt + \int_{t_1}^{t_2} \delta W dt = 0 \quad (2.3)$$

gegeben. Nacheinander sind also die kinetische Energie  $E_{kin}$  und die potenzielle Energie  $E_{pot}$  des Systems sowie die virtuelle Arbeit  $\delta W$  aller angreifenden potentiallosen Kräfte, zugeschnitten auf das zu betrachtende ein- oder zweiparametrische Strukturmodell, zu bestimmen. [Wau14]





### 3 Finite-Elmenete und Matrizen

Jedes Element vom Grundgebiet hat ein eigenes lokales Koordinatensystem. Die lokalen Koordinaten eines Elements sind von -1 bis +1 definiert. Die globale Länge eines Elements ist  $l_e$ .

Zunächst soll die Transformation von der lokalen Variable  $\xi$  mit dem Intervall  $[-1, 1]$  auf die globale Koordinate  $x$  mit dem Intervall  $[0, l_e]$  mit

$$\left. \begin{aligned} \xi(x) &= a_0 + a_1 x \\ \xi(x=0) &\stackrel{!}{=} -1 \\ \xi(x=l_e) &\stackrel{!}{=} 1 \end{aligned} \right\} \Rightarrow \xi = -1 + \frac{2}{l_e}x \Rightarrow \frac{\partial \xi}{\partial x} = \frac{2}{l_e} \quad (3.1)$$

beschrieben werden.

#### 3.1 Stabelemente

Für ein ungedämpftes Stabelement lautet die potentielle Energie in Abhängigkeit der Längsverschiebung  $u(x, t)$

$$E_{pot} = \frac{1}{2} \int_0^{l_e} EA u_{,x}^2 dx, \quad (3.2)$$

die kinetische Energie

$$E_{kin} = \frac{1}{2} \int_0^{l_e} \rho A u_{,t}^2 dx \quad (3.3)$$

und die virtuelle Arbeit der potentiallosen Kräfte

$$\delta W = 0. \quad (3.4)$$

Mit dem Prinzip von HAMILTON (Gleichung 2.3) ergibt sich

$$\delta \int_{t_0}^{t_1} \int_0^{l_e} \frac{1}{2} \rho A u_{,t}^2 - \frac{1}{2} EA u_{,x}^2 dx dt = 0 \quad (3.5)$$

und nach der Variation sowie partiellen Integration

$$\int_{t_0}^{t_1} \int_0^{l_e} \rho A u_{,tt} \delta u + EA u_{,x} \delta u_{,x} dx dt = 0. \quad (3.6)$$

Dabei handelt es sich um die schwache Formulierung der Bewegungsdifferentialgleichung des Dehnstabes. Für die Ortsdiskretisierung dieser Gleichung werden folgende Ansätze

$$\begin{aligned} u(x, t) &= \vec{N}(x) \vec{u}(t); & \delta u(x, t) &= \vec{N}(x) \delta \vec{u}(t) \\ u_{,x}(x, t) &= \vec{N}_{,x}(x) \vec{u}_{,t}(t); & \delta u_{,x}(x, t) &= \vec{N}_{,x}(x) \delta \vec{u}(t) \\ u_{,t}(x, t) &= \vec{N}(x) \vec{u}_{,t}(t) \\ u_{,tt}(x, t) &= \vec{N}(x) \vec{u}_{,tt}(t) \end{aligned} \quad (3.7)$$

verwendet. Die Vektoren für die Formfunktionen und die Elmenetfreiheitsgrade sind mit  $\vec{N}(x)$  und  $\vec{u}(t)$  gegeben. Zusammen mit Gleichung (3.6) ergibt sich

$$\int_{t_0}^{t_1} \int_0^{l_e} \left[ \rho A \vec{N}^T(x) \vec{N}(x) \vec{u}_{,tt}(t) + EA \vec{N}_{,x}^T(x) \vec{N}_{,x}(x) \vec{u}(t) \right] dx \delta \vec{u}(t) dt = 0. \quad (3.8)$$

Die Massenmatrix eines Elements lässt sich mit

$$\mathbf{M}_e = \int_0^{l_e} \rho A \vec{N}^T(x) \vec{N}(x) dx \quad (3.9)$$

und die Steifigkeitsmatrix mit

$$\mathbf{K}_e = \int_0^{l_e} EA \vec{N}_{,x}^T(x) \vec{N}_{,x}(x) dx \quad (3.10)$$

angeben. Nach dem Zusammenbau der Gesamtmassenmatrix  $\mathbf{M}$  und Gesamtsteifigkeitsmatrix  $\mathbf{K}$  aus den Elementmatrizen ergibt sich das FE-Gleichungssystem für den ungedämpften Stab zu

$$\mathbf{M} \cdot \ddot{\vec{q}}(t) + \mathbf{K} \cdot \vec{q}(t) = 0 \quad (3.11)$$

Der Vektor  $\vec{q}(t)$  beinhaltet dabei sämtliche Knotenfreiheitsgrade. Durch die Definition der Zusatzgrößen

$$\left. \begin{array}{l} \vec{z}_1 = \vec{q} \\ \vec{z}_2 = \dot{\vec{q}} \end{array} \right\} \Rightarrow \begin{array}{l} \dot{\vec{z}}_1 = \vec{z}_2 \\ \dot{\vec{z}}_2 = \ddot{\vec{q}} = -\mathbf{M}^{-1} \cdot \mathbf{K} \cdot \vec{q} = -\mathbf{M}^{-1} \cdot \mathbf{K} \cdot \vec{z}_1 \end{array} \quad (3.12)$$

kann Gleichung (3.11) im Zusatzraum

$$\begin{bmatrix} \dot{\vec{z}}_1 \\ \dot{\vec{z}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{E} \\ -\mathbf{M}^{-1} \cdot \mathbf{K} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \vec{z}_1 \\ \vec{z}_2 \end{bmatrix} \quad (3.13)$$

angegeben werden. Die Eigenfrequenzen des Stabes können durch die Eigenwerte der Systemmatrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{E} \\ -\mathbf{M}^{-1} \cdot \mathbf{K} & \mathbf{0} \end{bmatrix} \quad (3.14)$$

bestimmt werden.

### 3.1.1 Linearer Ansatz

Für die Längsverschiebung  $u(\xi)$  eines Elements soll ein linearer Ansatz der Form

$$u(\xi) = a_0 + a_1 \xi \quad (3.15)$$

verwendet werden. Zur Bestimmung der Koeffizienten  $a_i$  wird

$$u(-1) \stackrel{!}{=} u_1 ; u(1) \stackrel{!}{=} u_2 \quad (3.16)$$

gefordert. Die Knotenverschiebungen  $u_i(t)$  sind dabei Zeitfunktionen. Damit ergibt sich Gleichung (3.15) zu

$$u(\xi, t) = \frac{u_1 + u_2}{2} + \frac{1}{2}(-u_1 + u_2)\xi \quad (3.17)$$

bzw. in Vektor-Matrix Schreibweise

$$u(\xi, t) = \vec{N}(\xi) \vec{u}(t) = \begin{bmatrix} \frac{1}{2} - \frac{\xi}{2} & \frac{1}{2} + \frac{\xi}{2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (3.18)$$

Die Ableitung der Funktion  $u(\xi, t)$  nach der globalen Variable  $x$  erfolgt mit

$$u_{,x}(\xi, t) = \frac{\partial}{\partial x} [\vec{N}(\xi) \vec{u}(t)] = \frac{\partial}{\partial x} \frac{\partial \xi}{\partial \xi} [\vec{N}(\xi) \vec{u}(t)] = \frac{\partial \vec{N}(\xi)}{\partial \xi} \cdot \frac{\partial \xi}{\partial x} \cdot \vec{u}(t). \quad (3.19)$$

Mit dem Zusammenhang zwischen globaler und lokaler Koordinate (3.1) ergibt sich

$$\vec{N}_{,x}(\xi) = \frac{\partial \vec{N}(\xi)}{\partial \xi} \cdot \frac{\partial \xi}{\partial x} = \begin{bmatrix} -\frac{1}{2} & +\frac{1}{2} \end{bmatrix} \cdot \frac{2}{l_e}. \quad (3.20)$$

Die Elementmassenmatrix des  $i$ -ten Elements (Größe 2x2) ist mit

$$\mathbf{M}_{ei} = \begin{bmatrix} M_{ei,11} & M_{ei,12} \\ M_{ei,21} & M_{ei,22} \end{bmatrix} \quad (3.21)$$

gegeben. Damit kann die Gesamtmassenmatrix nach

$$\mathbf{M} = \begin{bmatrix} M_{e1,11} & M_{e1,12} & 0 & 0 & \cdots \\ M_{e1,21} & M_{e1,22} + M_{e2,11} & M_{e2,12} & 0 & \cdots \\ 0 & M_{e2,21} & M_{e2,22} + M_{e3,11} & M_{e3,12} & \cdots \\ 0 & 0 & M_{e3,21} & M_{e3,22} + M_{e4,11} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (3.22)$$

zusammengebaut werden. Nach gleicher Regel wird auch die Gesamtsteifigkeitsmatrix  $\mathbf{K}$  berechnet.

### 3.1.2 Quadratischer Ansatz

Ausgangspunkt ist der quadratische Ansatz

$$u(\xi) = a_0 + a_1 \xi + a_2 \xi^2 \quad (3.23)$$

mit den Forderungen

$$u(-1) \stackrel{!}{=} u_1 ; u(0) \stackrel{!}{=} u_2 ; u(+1) \stackrel{!}{=} u_3. \quad (3.24)$$

Nach Auswertung der Forderungen ergibt sich Gleichung (3.23) in Abhängigkeit der Knotenverschiebungen  $u_i(t)$  zu

$$u(\xi, t) = \vec{N}(\xi) \vec{u}(t) = \begin{bmatrix} \frac{\xi^2}{2} - \frac{\xi}{2} & 1 - \xi^2 & \frac{\xi}{2} + \frac{\xi^2}{2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \quad (3.25)$$

Die Ableitung des Vektors der Formfunktionen lautet

$$\vec{N}_{,x}(x) = \frac{\partial \vec{N}(\xi)}{\partial x} = \frac{\partial \vec{N}(\xi)}{\partial \xi} \cdot \frac{\partial \xi}{\partial x} = \begin{bmatrix} \xi - \frac{1}{2} & -2\xi & \frac{1}{2} + \xi \end{bmatrix} \cdot \frac{2}{l_e}. \quad (3.26)$$

Die Elementmassenmatrix des  $i$ -ten Elements (Größe 3x3) ist mit

$$\mathbf{M}_{ei} = \begin{bmatrix} M_{ei,11} & M_{ei,12} & M_{ei,13} \\ M_{ei,21} & M_{ei,22} & M_{ei,23} \\ M_{ei,31} & M_{ei,33} & M_{ei,33} \end{bmatrix} \quad (3.27)$$

gegeben. Damit kann die Gesamtmassenmatrix nach

$$\mathbf{M} = \begin{bmatrix} M_{e1,11} & M_{e1,12} & M_{e1,13} & \cdots \\ M_{e1,21} & M_{e1,22} & M_{e1,23} & \cdots \\ M_{e1,31} & M_{e1,32} & M_{e1,33} + M_{e2,11} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (3.28)$$

zusammengebaut werden. Nach gleicher Regel wird auch die Gesamtsteifigkeitsmatrix  $\mathbf{K}$  berechnet.

### 3.2 Balkenelemente

Für ein ungedämpftes Balkenelement lautet die kinetische Energie in Abhängigkeit der Durchbiegung  $w(x, t)$

$$E_{kin} = \frac{\rho A}{2} \int_0^{l_e} \dot{w}_{,t}^2 dx, \quad (3.29)$$

die potenzielle Energie

$$E_{pot} = \frac{1}{2} \int_0^{l_e} EI w_{,xx}^2 dx \quad (3.30)$$

und die virtuelle Arbeit der potentiallosen Kräfte

$$\delta W = 0. \quad (3.31)$$

Mit dem Prinzip von HAMILTON (Gleichung 2.3) ergibt sich

$$\delta \frac{1}{2} \int_{t_0}^{t_1} \int_0^{l_e} \rho A \dot{w}_{,t}^2 - EI w_{,xx}^2 dx dt = 0 \quad (3.32)$$

und nach der Variation sowie partiellen Integration

$$\int_{t_0}^{t_1} \int_0^{l_e} \rho A \dot{w}_{,tt} \delta w + EI w_{,xx} \delta w_{,xx} dx dt = 0. \quad (3.33)$$

Die schwache Formulierung der Bewegungsdifferentialgleichung eines Biegebalkens wird mit den Ansätzen

$$\begin{aligned} w(x, t) &= \vec{N}(x) \vec{w}(t); & \delta w(x, t) &= \vec{N}(x) \delta \vec{w}(t) \\ w_{,x}(x, t) &= \vec{N}_{,x}(x) \vec{w}(t); & \delta w_{,x}(x, t) &= \vec{N}_{,x}(x) \delta \vec{w}(t) \\ w_{,xx}(x, t) &= \vec{N}_{,xx}(x) \vec{w}(t); & \delta w_{,xx}(x, t) &= \vec{N}_{,xx}(x) \delta \vec{w}(t) \\ w_{,t}(x, t) &= \vec{N}(x) \dot{\vec{w}}(t) \\ w_{,tt}(x, t) &= \vec{N}(x) \ddot{\vec{w}}(t) \end{aligned} \quad (3.34)$$

ortsdiskretisiert. Zusammen mit Gleichung (3.33) ergibt sich

$$\int_{t_0}^{t_1} \int_0^{l_e} \left[ \rho A \vec{N}^T(x) \vec{N}(x) \ddot{\vec{w}}(t) + EI \vec{N}_{,xx}^T(x) \vec{N}_{,xx}(x) \vec{w}(t) \right] dx \delta \vec{w}(t) dt = 0. \quad (3.35)$$

Die Massenmatrix  $\mathbf{M}_e$  eines Elements ist mit

$$\mathbf{M}_e = \int_0^{l_e} \rho A \vec{N}^T(x) \vec{N}(x) dx \quad (3.36)$$

und die Steifigkeitsmatrix  $\mathbf{K}_e$  mit

$$\mathbf{K}_e = \int_0^{l_e} EI \vec{N}_{,xx}^T(x) \vec{N}_{,xx}(x) dx \quad (3.37)$$

gegeben. Der Übergang zum Zustandsraum und die Angabe der Systemmatrix ist analog wie für den Dehnstab in Abschnitt 3.1 beschrieben (Gleichungen 3.11 und 3.14). Die Eigenwerte und Eigenvektoren der Systemmatrix liefern wieder die Eigenfrequenzen und Eigenschwingformen des Balkens.

### 3.2.1 Kubischer Ansatz

Mit einem kubischen Ansatz für die Funktion  $w(\xi)$  lässt sich ein auch für die Balkenbiegung konformes Element gewinnen, wenn neben den Funktionswerten auch noch die ersten Ableitung in den Endpunkten als Knotenvariable eingeführt werden. Es wird die Ansatzfunktion

$$w(\xi) = a_0 + a_1\xi + a_2\xi^2 + a_3\xi^3 \quad (3.38)$$

verwendet und zur Bestimmung der Koeffizienten  $a_i$  wird

$$w(-1) \stackrel{!}{=} w_1 ; w_{,\xi}(-1) \stackrel{!}{=} \phi_1 ; w(+1) \stackrel{!}{=} w_2 ; w_{,\xi}(+1) \stackrel{!}{=} \phi_2 \quad (3.39)$$

gefordert. Damit ergibt sich Gleichung (3.38) zu

$$w(\xi, t) = \vec{N}(\xi) \vec{w}(t) = \begin{bmatrix} -\frac{1}{16} + \frac{\xi}{16} + \frac{9\xi^2}{16} - \frac{9\xi^3}{16} \\ \frac{9}{16} - \frac{27\xi}{16} - \frac{9\xi^2}{16} + \frac{27\xi^3}{16} \\ \frac{9}{16} + \frac{27\xi}{16} - \frac{9\xi^2}{16} - \frac{27\xi^3}{16} \\ -\frac{1}{16} - \frac{\xi}{16} + \frac{9\xi^2}{16} + \frac{9\xi^3}{16} \end{bmatrix}^T \cdot \begin{bmatrix} w_1 \\ \phi_1 \\ w_2 \\ \phi_2 \end{bmatrix}. \quad (3.40)$$

Für die Balkenelemente wird die zweite Ableitung von  $\vec{N}(\xi)$  nach  $x$  benötigt, die durch

$$\vec{N}_{,xx}(\xi) = \frac{\partial^2 \vec{N}(\xi)}{\partial \xi^2} \cdot \left[ \frac{\partial \xi}{\partial x} \right]^2 = \begin{bmatrix} \frac{1}{16} + \frac{9\xi}{8} - \frac{27\xi^2}{16} \\ -\frac{27}{16} - \frac{9\xi}{8} + \frac{81\xi^2}{16} \\ \frac{27}{16} - \frac{9\xi}{8} - \frac{81\xi^2}{16} \\ -\frac{1}{16} + \frac{9\xi}{8} + \frac{27\xi^2}{16} \end{bmatrix}^T \cdot \left( \frac{2}{l_e} \right)^2 \quad (3.41)$$

gegeben ist. Mit den Gleichungen (3.36) und (3.37) können die Massenmatrix  $\mathbf{M}_e$  und Steifigkeitsmatrix  $\mathbf{K}_e$  eines Elements berechnet werden und die Gesamtmassenmatrix  $\mathbf{M}$  sowie Gesamtsteifigkeitsmatrix  $\mathbf{K}$  zusammengebaut werden. Die Massenmatrix des  $i$ -ten Elements (Größe 4x4) lautet

$$\mathbf{M}_{ei} = \begin{bmatrix} M_{ei,11} & M_{ei,12} & M_{ei,13} & M_{ei,14} \\ M_{ei,21} & M_{ei,22} & M_{ei,23} & M_{ei,24} \\ M_{ei,31} & M_{ei,32} & M_{ei,33} & M_{ei,34} \\ M_{ei,41} & M_{ei,42} & M_{ei,43} & M_{ei,44} \end{bmatrix} \quad (3.42)$$

und die Gesamtmassenmatrix  $\mathbf{M}$  ergibt sich zu

$$\mathbf{M} = \begin{bmatrix} M_{e1,11} & M_{e1,12} & M_{e1,13} & M_{e1,14} & \cdots \\ M_{e1,21} & M_{e1,22} & M_{e1,23} & M_{e1,24} & \cdots \\ M_{e1,31} & M_{e1,32} & M_{e1,33} + M_{e2,11} & M_{e1,34} + M_{e2,12} & \cdots \\ M_{e1,41} & M_{e1,42} & M_{e1,43} + M_{e2,21} & M_{e1,44} + M_{e2,22} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \quad (3.43)$$

Nach gleicher Regel wird auch die Gesamtsteifigkeitsmatrix  $\mathbf{K}$  berechnet.

### 3.3 Plattenelemente

Es wird eine Platte mit Rechteckform betrachtet, deren Dicke  $H$  im Verhältnis zu den beiden anderen Abmessungen klein ist. Die Mittelebene soll mit der  $(x, y)$ -Ebene zusammenfallen. Mit den verbleibenden Spannungen  $t_{xx}, t_{yy}, t_{xy} \neq 0$  kann die potentielle Energie mit

$$E_{pot} = \frac{1}{2} \int_V (t_{xx} \epsilon_{xx} + t_{yy} \epsilon_{yy} + 2t_{xy} \epsilon_{xy}) dV. \quad (3.44)$$

angegeben werden. Mit dem HOOKESCHEN Gesetz lauten die Spannungen  $t_{ij}$

$$t_{xx} = \frac{E}{1-\nu^2} (\epsilon_{xx} + \nu \epsilon_{yy}), \quad t_{yy} = \frac{E}{1-\nu^2} (\epsilon_{yy} + \nu \epsilon_{xx}), \quad t_{xy} = \frac{E}{1+\nu} \epsilon_{xy}, \quad (3.45)$$

womit sich

$$E_{pot} = \frac{E}{2(1-\nu^2)} \int_A \int_{-H/2}^{+H/2} (\epsilon_{xx}^2 + 2\nu \epsilon_{xx} \epsilon_{yy} + \epsilon_{yy}^2) dZ dA + \frac{E}{(1+\nu)} \int_A \int_{-H/2}^{+H/2} \epsilon_{xy}^2 dZ dA \quad (3.46)$$

ergibt. Der Verschiebungsvektor  $\vec{u}$  für ein Plattenelement kann nach [Wau14] mit

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & -Z \frac{\partial}{\partial x} \\ 0 & 1 & -Z \frac{\partial}{\partial y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \quad (3.47)$$

angegeben werden. Es treten drei voneinander unabhängige Verschiebungen  $w_x$ ,  $w_y$  und  $w_z$  auf und die Biegewinkel lassen sich über  $w_{z,x}$  und  $w_{z,y}$  ausdrücken. Die in Gleichung (3.46) benötigten Verzerrungen  $\epsilon_{ij}$  ergeben sich aus den Verzerrungs-Verschiebungs-Relationen

$$\epsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i}) \quad (3.48)$$

und nach dem Ersetzen der Verschiebungsfunktionen

$$\begin{aligned} \epsilon_{xx} &= w_{x,x} - Z w_{z,xx}, & \epsilon_{yy} &= w_{y,y} - Z w_{z,yy} \\ \epsilon_{xy} &= \frac{1}{2} (w_{x,y} + w_{y,x}) - Z w_{z,xy}. \end{aligned} \quad (3.49)$$

Es werden nur die Verschiebung  $w_z$  betrachtet, weil es an dieser Stelle um die Durchbiegung einer Platte geht. Deshalb sind die Verschiebungen  $w_x$  und  $w_y$  zu null gefordert. Mit dem differenziellen Volumen  $dV = dZ dA$  und der Plattendicke  $h$ , die im Allgemeinen von den Parametern  $x, y$  abhängt, ergibt sich nach der Integration über  $z$  das Endergebnis für die potenzielle Energie

$$E_{pot} = \frac{E}{2 \cdot 12(1-\nu^2)} \int_A H^3 [w_{z,xx}^2 + w_{z,yy}^2 + 2\nu w_{z,xx} w_{z,yy} + 2(1-\nu) w_{z,xy}^2] dA. \quad (3.50)$$

Die kinetische Energie ist ähnlich zu dem Balkenelement, wobei hier über die Fläche integriert wird

$$E_{kin} = \frac{\rho}{2} \int_A H w_{z,t}^2 dA. \quad (3.51)$$

Wird wieder  $\delta W = 0$  angenommen und der Ansatz

$$w_z = \vec{N}(x, y) \vec{w}(t) \quad (3.52)$$

gewählt, liefert das Prinzip von HAMILTON (2.3) folgende Gleichung

$$0 = \int_{t_0}^{t_1} \int_A \left\{ \rho H \vec{N}^T(x, y) \vec{N}(x, y) \vec{w}_{,tt}(t) + \frac{EH^3}{12(1-\nu^2)} \left[ \vec{N}_{,xx}^T(x, y) \vec{N}_{,xx}(x, y) + \nu (\vec{N}_{,xx}^T(x, y) \vec{N}_{,yy}(x, y) + \vec{N}_{,yy}^T(x, y) \vec{N}_{,xx}(x, y)) + \vec{N}_{,yy}^T(x, y) \vec{N}_{,yy}(x, y) + 2(1-\nu) \vec{N}_{,xy}^T(x, y) \vec{N}_{,xy}(x, y) \right] \right\} dA \delta \vec{w}(t) dt \quad (3.53)$$

Damit ist die Massenmatrix eines Elements

$$\mathbf{M}_e = \int_{A_e} \rho H \vec{N}^T(x, y) \vec{N}(x, y) dA_e \quad (3.54)$$

und die Steifigkeitsmatrix eines Elements ist

$$\mathbf{K}_e = \int_{A_e} \frac{EH^3}{12(1-\nu^2)} \left[ \vec{N}_{,xx}^T(x, y) \vec{N}_{,xx}(x, y) + \nu (\vec{N}_{,xx}^T(x, y) \vec{N}_{,yy}(x, y) + \vec{N}_{,yy}^T(x, y) \vec{N}_{,xx}(x, y)) + \vec{N}_{,yy}^T(x, y) \vec{N}_{,yy}(x, y) + 2(1-\nu) \vec{N}_{,xy}^T(x, y) \vec{N}_{,xy}(x, y) \right] dA_e. \quad (3.55)$$

### 3.3.1 Konforme Plattenelemente

Zuerst wird ein konformes Rechteckelement diskutiert, da seine Beschreibung und die Verifikation der Stetigkeit der Normalableitung relativ einfach sind. Im Einheitsquadrat  $Q_e$  gilt für die Durchbiegung  $w(\xi, \eta)$  der bikubischer Ansatz

$$w(\xi, \eta) = a_0 + a_1 \xi + a_2 \eta + a_3 \xi^2 + a_4 \xi \eta + a_5 \eta^2 + a_6 \xi^3 + a_7 \xi^2 \eta + a_8 \xi \eta^2 + a_9 \eta^3 + a_{10} \xi^3 \eta + a_{11} \xi^2 \eta^2 + a_{12} \xi \eta^3 + a_{13} \xi^3 \eta^2 + a_{14} \xi^2 \eta^3 + a_{15} \xi^3 \eta^3. \quad (3.56)$$

Mit den Knotenvariablen

$$w; w_{,\xi}; w_{,\eta}; w_{,\xi\eta} \quad (3.57)$$

ergeben sich in den vier Eckpunkten des Elements die Forderungen

$$\begin{aligned} w(-1, -1) &= w_1 & w(+1, -1) &= w_2 & w(-1, +1) &= w_3 & w(+1, +1) &= w_4 \\ w_{,\xi}(-1, -1) &= w_{1,\xi} & w_{,\xi}(+1, -1) &= w_{2,\xi} & w_{,\xi}(-1, +1) &= w_{3,\xi} & w_{,\xi}(+1, +1) &= w_{4,\xi} \\ w_{,\eta}(-1, -1) &= w_{1,\eta} & w_{,\eta}(+1, -1) &= w_{2,\eta} & w_{,\eta}(-1, +1) &= w_{3,\eta} & w_{,\eta}(+1, +1) &= w_{4,\eta} \\ w_{,\xi\eta}(-1, -1) &= w_{1,\xi\eta} & w_{,\xi\eta}(+1, -1) &= w_{2,\xi\eta} & w_{,\xi\eta}(-1, +1) &= w_{3,\xi\eta} & w_{,\xi\eta}(+1, +1) &= w_{4,\xi\eta}. \end{aligned} \quad (3.58)$$

Die Formfunktionen  $\vec{N}(\xi, \eta)$  ergeben sich in diesem Fall zwangsläufig als Produkte von Formfunktionen  $\vec{N}(\xi)$ ,  $\vec{N}(\eta)$  des eindimensionalen Falls, sodass

$$\begin{aligned} N_1(\xi, \eta) &= \frac{1}{16}(-1+\xi)^2(2+\xi)(-1+\eta)^2(2+\eta) \\ N_2(\xi, \eta) &= \frac{1}{16}(-1+\xi)^2(1+\xi)(-1+\eta)^2(2+\eta) \\ N_3(\xi, \eta) &= \frac{1}{16}(-1+\xi)^2(2+\xi)(-1+\eta)^2(1+\eta) \\ N_4(\xi, \eta) &= \frac{1}{16}(-1+\xi)^2(1+\xi)(-1+\eta)^2(1+\eta) \\ &\text{etc.} \end{aligned} \quad (3.59)$$

gilt. Die Ableitungen der Formfunktionen in  $x,y$ -Richtung sind identisch zu den Ableitungen des eindimensionalen Falls. Anschließend können die Matrizen für Masse und Steifigkeit nach der Gleichung (3.54) und (3.55) berechnet werden. Im Gegensatz zum eindimensionalen Fall ist der Zusammenbau der Gesamtmatrizen komplizierter. Notwendig ist der Zusammenhang zwischen lokalen und globalen Knotennummern.

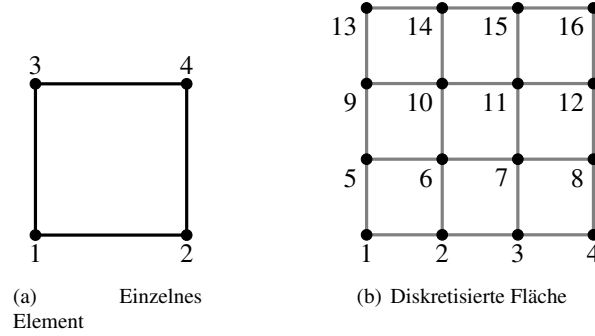


Abbildung 3.1: Diskretisierung mit Hilfe von Rechteckelementen.

Die globalen Knotennummern vom ersten Element in der diskretisierten Fläche sind nicht  $[1, 2, 3, 4]$ , sondern  $[1, 2, 5, 6]$ . In der folgenden Tabelle sind die globalen Knotennummern aller Elemente aus Abbildung 3.1 angegeben.

Element	1	2	3	4	5	6	7	8	9
Knotennummer	[1,2,5,6]	[2,3,6,7]	[3,4,7,8]	[5,6,9,10]	[6,7,10,11]	[7,8,11,12]	[9,10,13,14]	[10,11,14,15]	[11,12,15,16]

Tabelle 3.1: Globale Knotennummern aller Elemente aus Abbildung 3.1.

Die Zeilen und Spalten der Gesamtmassenmatrix und Gesamtsteifigkeitsmatrix werden entsprechend der Knotennummern gefüllt. Mit der Elementmassenmatrix des  $i$ -ten Elements (Größe  $4 \times 4$ )

$$\mathbf{M}_{e1} = \begin{bmatrix} \mathbf{M}_{ei,11} & \mathbf{M}_{ei,12} & \mathbf{M}_{ei,13} & \mathbf{M}_{ei,14} \\ \mathbf{M}_{ei,21} & \mathbf{M}_{ei,22} & \mathbf{M}_{ei,23} & \mathbf{M}_{ei,24} \\ \mathbf{M}_{ei,31} & \mathbf{M}_{ei,32} & \mathbf{M}_{ei,33} & \mathbf{M}_{ei,34} \\ \mathbf{M}_{ei,41} & \mathbf{M}_{ei,42} & \mathbf{M}_{ei,43} & \mathbf{M}_{ei,44} \end{bmatrix} \quad (3.60)$$

kann die Gesamtmassenmatrix wie folgt

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{e1,11} & \mathbf{M}_{e1,12} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{e1,13} & \cdots \\ \mathbf{M}_{e1,21} & \mathbf{M}_{e1,22} + \mathbf{M}_{e2,11} & \mathbf{M}_{e2,12} & \mathbf{0} & \mathbf{M}_{e1,23} & \cdots \\ \mathbf{0} & \mathbf{M}_{e2,21} & \mathbf{M}_{e2,22} + \mathbf{M}_{e3,11} & \mathbf{M}_{e3,12} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{e3,21} & \mathbf{M}_{e3,22} & \mathbf{0} & \cdots \\ \mathbf{M}_{e1,31} & \mathbf{M}_{e1,32} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{e1,33} + \mathbf{M}_{e4,11} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (3.61)$$

zusammengebaut werden. Nach der gleichen Regel wird auch die Gesamtsteifigkeitsmatrix zusammengebaut. Der Übergang in den Zustandsraum und die Berechnung von Eigenwerten und Eigenvektoren ist analog zum Stab- und Balkenelement.



## 4 Ergebnisse

### 4.1 Längsschwingungen eines Stabs

Es wird ein einseitig eingespannter Stab betrachtet. Die Länge des Stabs ist 10 m, die Querschnittsfläche ist  $0,0001 \text{ m}^2$ , die Dichte ist  $7850 \text{ kg/m}^3$  und der E-Modul ist  $2,1 \times 10^{11} \text{ N/m}^2$ .

Die Eigenfrequenzen und Moden werden wie in Kapitel 3 beschrieben mit MATLAB® berechnet. Zum Vergleich werden die analytischen Ergebnisse nach [SP95] mit angegeben. Die analytischen Eigenwerte  $\lambda_k$  des Randwertproblems eines einseitig eingespannten Stabs sind

$$\lambda_k = \frac{2k-1}{2l} \pi. \quad (4.1)$$

Der Parameter  $k$  gibt die Ordnung des jeweiligen Modes an. Die Eigenfrequenzen sind mit

$$f_k = \frac{\lambda_k \sqrt{\frac{E}{\rho}}}{2\pi}; (k = 1, 2, \dots) \quad (4.2)$$

definiert. In der folgenden Tabelle sind die Ergebnisse berechnet mit MATLAB® und die analytische Lösung angegeben. Bei den Ergebnissen mit MATLAB® wurden lineare und quadratische Ansatzfunktionen verwendet und der Stab in 9 oder 15 Elemente zerlegt.

Eigenfrequenz [Hz] \ Mode		1	2	3	4	5	6	7	8	9
Linearer Ansatz	9-Elemente	129,4690	392,3596	667,1834	961,8870	1283,2110	1632,6871	1996,5486	2327,9315	2537,4539
	15-Elemente	129,3639	389,5117	653,9326	925,5110	1207,1451	1501,6589	1811,5811	2138,6850	2483,1315
Quadratischer Ansatz	9-Elemente	129,3050	387,9544	647,0233	907,7234	1172,4705	1445,1451	1731,3816	2039,0736	2379,3269
	15-Elemente	129,3048	387,9197	646,5906	905,4860	1164,9579	1425,5942	1688,2579	1954,1176	2224,6784
Analytisches Ergebnis		129,3048	387,9145	646,5242	905,1339	1163,7436	1422,3533	1680,9630	1939,5728	2198,1825

Tabelle 4.1: Eigenfrequenzen der Längsschwingung eines einseitig eingespannten Stabs.

Im Vergleich zur analytischen Lösung ist ersichtlich, dass der quadratische Ansatz, bei gleicher Anzahl an Elementen, besser ist als der lineare Ansatz. Je mehr Elemente verwendet werden, desto besser ist die Genauigkeit.

### 4.2 Biegeschwingung eines Balkens

Es wird ein einseitig eingespannter Balken betrachtet. Die Länge des Balkens beträgt  $l = 10 \text{ m}$ , die Dichte  $\rho = 7850 \text{ kg/m}^3$ , der E-Modul  $E = 2,1 \times 10^{11} \text{ N/m}^2$  und die rechteckige Querschnittsfläche hat die Breite  $b = 0,005 \text{ m}$  sowie die Höhe  $h = 0,02 \text{ m}$ .

Es werden wieder die FE-Ergebnisse (MATLAB®) mit der analytischen Lösung nach [SP95] verglichen. Die analytischen Eigenwert  $\lambda_k$  des einseitig eingespannten Balkens sind

$$\begin{aligned} \lambda_1 &= 1,87810; & \lambda_2 &= 4,69409; \\ \lambda_3 &= 7,85476; & \lambda_4 &= 10,99554; \\ \lambda_k &= \frac{(2k-1)\pi}{2}; (k = 5, 6, \dots). \end{aligned} \quad (4.3)$$

Der Parameter  $k$  gibt die Ordnung des jeweiligen Modes an. Die Eigenfrequenzen sind mit

$$f_k = \frac{\lambda_k^2 \sqrt{\frac{EI}{\rho A l^4}}}{2\pi} \quad (4.4)$$

definiert. Die Parameter  $A$  und  $I$  geben die Querschnittsfläche und das Flächenträgheitsmoment des Balkens an. In der folgenden Tabelle sind die Ergebnisse berechnet mit MATLAB<sup>®</sup> und die analytische Lösung angegeben. Bei den Ergebnissen mit MATLAB<sup>®</sup> wurden kubische Ansatzfunktion verwendet und der Balken in 9 oder 15 Elemente zerlegt.

Mode		1	2	3	4	5	6	7	8	9
Eigenfrequenz [Hz]										
Kubischer Ansatz	9-Elemente	0,167103	1,047271	2,933378	5,754348	9,5349445	14,305110	20,113484	26,998875	34,636055
	15-Elemente	0,167103	1,047225	2,932393	5,747150	9,5036211	14,205800	19,862454	26,487922	34,103683
Analytisches Ergebnis		0,167102	1,047218	2,932244	5,746024	9,4985889	14,189250	19,818043	26,384969	33,890027

Tabelle 4.2: Eigenfrequenzen der Biegeschwingungen eines einseitig eingespannten Balkens.

Auch für den Balken ist erkennbar, dass mehr Elemente eine höhere Genauigkeit erzielen. Bild 4.1 zeigt die verschiedenen Eigenschwingformen des einseitig eingespannten Balkens, die anhand der FE-Ergebnisse dargestellt werden.

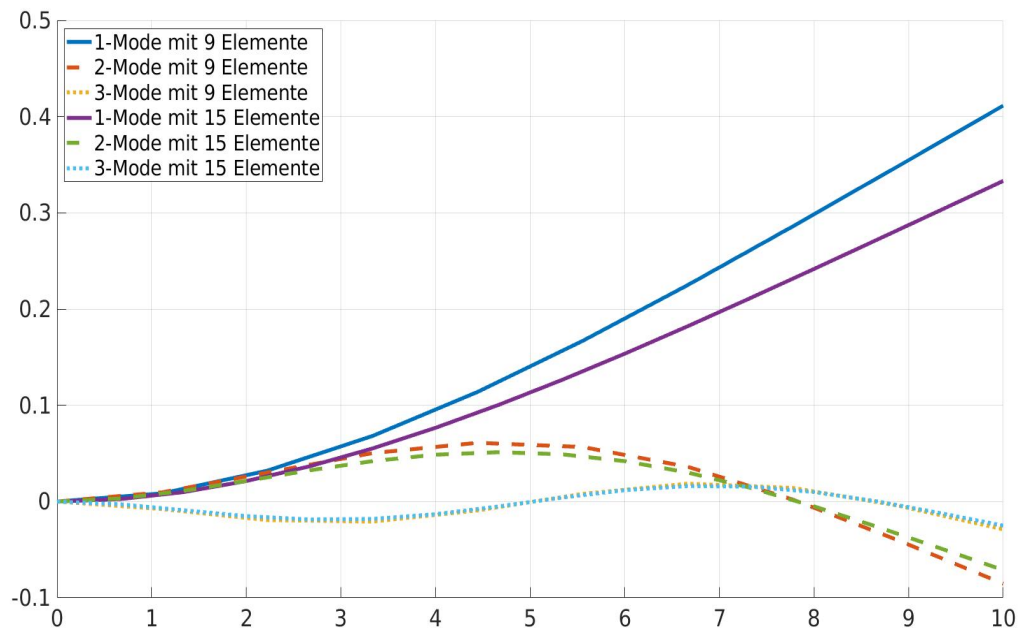


Abbildung 4.1: Eigenschwingformen des Balkens mit unterschiedlicher Elementanzahl.

### 4.3 Biegeschwingungen einer Platte

Es wird eine einseitig eingespannte Platte betrachtet. Die Länge der Platte ist  $L = 2\text{ m}$ , die Breite  $B = 2\text{ m}$ , die Dicke  $H = 0,01\text{ m}$ , die Dichte  $\rho = 7850\text{ kg/m}^3$ , der E-Modul  $E = 2,1 \times 10^{11}\text{ N/m}^2$  und die Poissonzahl  $\nu = 0,3$ .

Es werden die FE-Ergebnisse (MATLAB<sup>®</sup>) mit denen aus ANSYS<sup>®</sup> verglichen. In der folgenden Tabelle sind die verschiedenen Ergebnissen aufgelistet. Bei der Simulation mit MATLAB<sup>®</sup> wird die Platte durch

konforme Elemente diskretisiert mit je 9 oder 15 Elementen pro Seite. Bei der Simulation mit ANSYS® wird jede Seite der Platte in 1000 Teile zerlegt.

Eigenfrequenz [Hz] \ Mode	1	2	3	4	5	6	7	8	9
9 Elemente je Seite	2,162105	5,299083	13,259113	16,940469	19,284920	33,759497	38,167191	39,961917	44,225406
15 Elemente je Seite	2,161776	5,298090	13,256507	16,938819	19,280190	33,748742	38,150961	39,947782	44,201679
Ergebnis ANSYS®	2,161505	5,293666	13,254572	16,932654	19,265101	33,713487	38,157460	39,942079	44,185442

Tabelle 4.3: Eigenfrequenzen der Biegeschwingungen einer einseitig eingespannten Platte.

Es ist ersichtlich, dass die Genauigkeit des Eigenfrequenzen, im Vergleich zu ANSYS®, von der Anzahl der Elemente abhängig ist. D.h. je mehr Elemente verwendet werden, desto besser ist das Ergebnis. Obwohl mehr Elemente die Genauigkeit der Ergebnisse steigern, ist die Simulation mit mehr Rechenzeit verbunden. Deshalb muss die Anzahl der Elemente richtig bestimmt werden, um eine schnellen Rechnung mit ausreichender Genauigkeit zu erreichen.

Die Bilder 4.2 und 4.3 zeigen die verschiedenen Eigenschwingformen der einseitig eingespannten Platte, die durch die Simulation mit MATLAB® berechnet und dargestellt wurden.

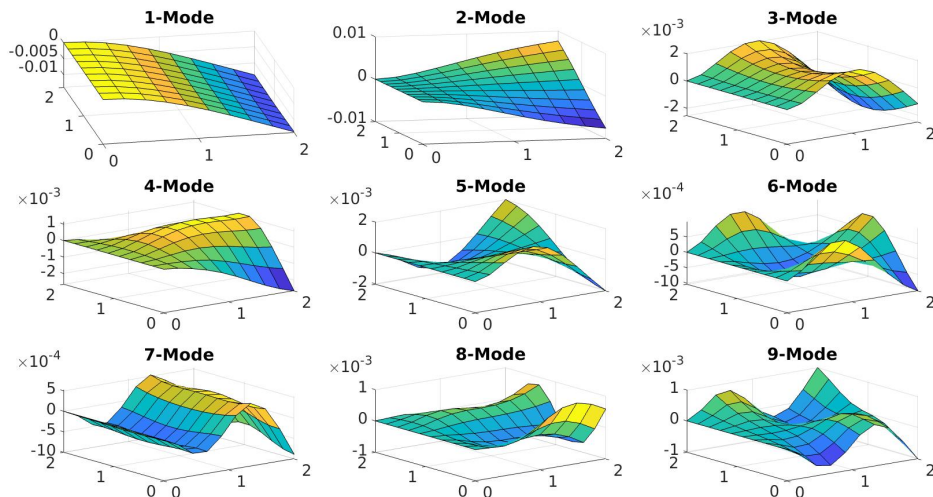


Abbildung 4.2: Eigenschwingformen einer einseitig eingespannten Platte mit 9 Elementen je Seite.

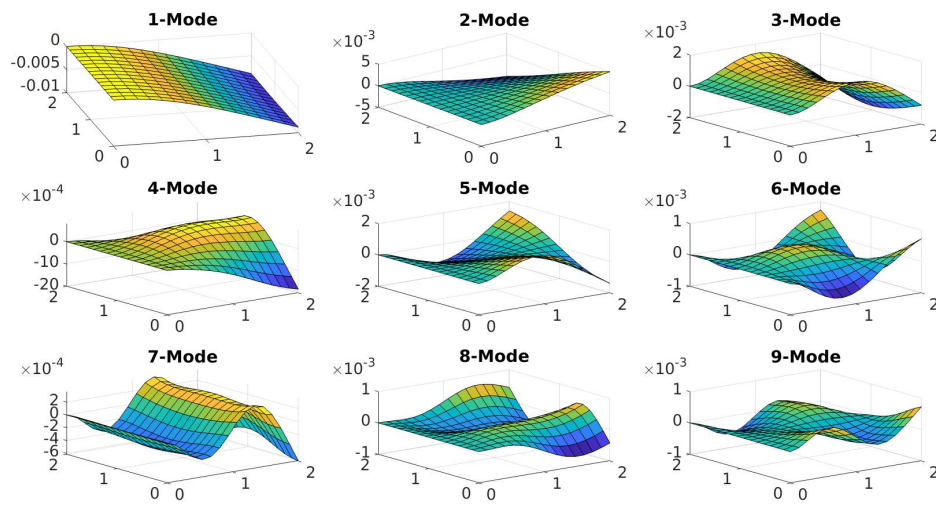


Abbildung 4.3: Eigenschwingformen einer einseitig eingespannten Platte mit 15 Elementen je Seite.

## 5 Zusammenfassung

Das Ziel der Modalanalyse besteht darin, eine Basis für die Identifizierung der modalen Parameter des Systems bereitzustellen. Die Analyse der Schwingungseigenschaften des strukturellen Systems dient zur Diagnose und Vorhersage von Vibrationsfehlern sowie der Optimierung der dynamischen Strukturmerkmale. Daher untersucht die Modalanalyse von Anfang an die Eigenschaften der Struktur. Das Verständnis der Eigenfrequenzen und der Eigenschwingformen hilft beim Entwurf eines Systems, das den Anforderungen von Lärm- und Vibrationsanwendungen entspricht.

In dieser Arbeit wurde die numerische Modalanalyse mit der Finite-Elemente-Methode betrachtet. Die FE Simulationen wurde mit MATLAB<sup>®</sup> durchgeführt und mit der kommerziellen Software ANSYS<sup>®</sup> sowie analytischen Lösungen nach [SP95] verglichen.

Durch die theoretischen Grundlagen wurden die numerischen Verfahren deutlich erklärt und das Fundament für die Simulationen gelegt. In den Simulationen wurden drei verschiedene Körper (Stab, Balken, Platte) betrachtet, die häufig im konstruktiven Ingenieurbau angewendet werden. Anhand der Ergebnisse lässt sich der Zusammenhang angeben, dass die Genauigkeit der Näherungslösung durch die Verwendung von immer mehr Parametern (z. B. mehr und mehr kleinere Elemente) oder Ansatzfunktionen mit immer höherem Wert verbessert werden kann.

Darüber hinaus bedeuten immer mehr Parameter oder Ansatzfunktionen mit immer höherem Wert, dass die Simulationen mit mehr Rechenzeit verbunden sind. Deshalb muss die Anzahl der Elemente oder die Ansatzfunktionen problemorientiert berücksichtigt werden, um schnelle Rechnungen mit ausreichender Genauigkeit zu gewährleisten.

Die Fehler der FE-Lösungen können auf unterschiedliche Gründe zurückgeführt werden. Vor allem sind die Anzahl der Elemente und die Ansatzfunktionen dafür verantwortlich. Außerdem gibt es numerische Fehler wie beispielsweise durch die Integralrechnung. Verschiedene Quadratur-Verfahren haben unterschiedliche numerische Genauigkeiten. In dieser Arbeit wurde nur die GAUSS-Quadratur benutzt, da es sich um ein einfaches und schnelles Verfahren handelt.

Durch die Erklärung der theoretischen Grundlagen und die Simulation von verschiedenen Körpern wurde die Modalanalyse für den eindimensionalen und zweidimensionalen Fall betrachtet. Da viele Probleme ein dreidimensionale Modell erfordern, kann als nächster Schritt dieser Fall betrachtet werden.



## Quellenverzeichnis

- [Sch91] Hans-Rudolf Schwarz. Methode der finiten Elemente: eine Einführung unter besonderer Berücksichtigung der Rechenpraxis, volume 47. Teubner-Verlag, 1991.
- [SP95] Wolfgang Stephan and Rudolf Postl. Schwingungen elastischer Kontinua. Springer, 1995.
- [Wau14] Jörg Wauer. Kontinuumsschwingungen. Springer, 2014.





## Anhang A: Programm Stabelemente

### Hauptprogramm linearer Ansatz

```

clear
clc
close
#####
% main Program
#####
% parameter
E=2.1e11;           % N/m^2
A=0.0001;           % m^2
l=10;               % m
rho=7850;            % Dichte in [kg/m^3]
mu=rho*A;           % Massenbelegung in [kg/m]
Nel=15;              % number of elements
Nno=Nel+1;           % number of nodes
le=l/Nel;            % length of an element

% define empty matrice
Kt=zeros(Nno);      % empty global stiffness-matrix
M=zeros(Nno);        % empty global mass-matrix

% call element routinesmbclient
[Kte,Me] = Elementroutine_linear(A,E,rho,le);

% loop over every element
for j=1:Nel
    M(j : j+1, j : j+1) = M(j : j+1, j : j+1) + Me;
    Kt(j : j+1, j : j+1) = Kt(j : j+1, j : j+1) + Kte;
end

% implementation of essential boundary conditions
Kt(1,:) = [ ];
Kt(:,1) = [ ];
M(1,:) = [ ];
M(:,1) = [ ];

% define system-matrix
null=zeros(size(M));
Eins = eye(size(M));
SysMat=[null,Eins;-inv(M)*Kt,null];

% compute Eigenvalues
[V,D]=eig(SysMat);
temp_d = diag(D);
[nd, sortindex] = sort(temp_d);
temp_v = V(:,sortindex);

```

```

temp_f = imag(nd)/(2*pi);

% analysis method
lamda = zeros(Nel, 1);
omega = zeros(Nel, 1);
f = zeros(Nel, 1);
for k = 1: Nel
    lamda(k) = (2 * k - 1) * pi / (2 * l);
    omega(k) = lamda(k)*sqrt(E/rho);
    f(k) = omega(k)/(2*pi);
end

```

### Elementrechnung linearer Ansatz

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Elementroutine
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Kte,Me] = Elementroutine_linear(A,E,rho,le)
% Elementroutine: compute Kte, Me
% define empty Kte
Kte=[0,0;0,0];
% define empty Me
Me=[0,0;0,0];

% define sampling points for Gauss-quadrature
xiVec=[-sqrt(1/3),sqrt(1/3)];
% weights for sampling points of Gauss-quadrature
wVec=[1,1];

for i=1:length(xiVec)
    xi=xiVec(i);
    w=wVec(i);

    % define N, B vector
    N=[0.5-xi/2  0.5+xi/2];
    Nx=[-0.5  0.5]*(2/le);

    % compute Kte and Me for sampling point of Gauss-integration
    Me=Me + rho * A * (N' * N) * w;
    Kte=Kte + E * A * (Nx' * Nx) * w;
end

end

```

### Hauptprogramm quadratischer Ansatz

```

clear
close
clc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% main Program
#####
% parameter
E=2.1e11;           % N/m^2
A=0.0001;           % m^2
l=10;               % m
rho=7850;           % Dichte in [kg/m^3]
mu=rho*A;           % Massenbelegung in [kg/m]
Nel=15;             % number of elements
Nno=Nel*2+1;        % number of nodes
le=l/Nel;           % length of an element

% define empty matrice
Kt=zeros(Nno);      % empty global stiffness-matrix
M=zeros(Nno);       % empty global mass-matrix

% call element routinesmbclient
[Kte,Me] = Elementroutine_quadra(A,E,mu,le);
% loop over every element
for j=1:2 : Nno-1
    M(j : j+2, j : j+2) = M(j : j+2, j : j+2) + Me;
    Kt(j : j+2, j : j+2) = Kt(j : j+2, j : j+2) + Kte;
end

% implementation of essential boundary conditions
Kt(1,:) = [ ];
Kt(:,1) = [ ];
M(1,:) = [ ];
M(:,1) = [ ];

% define system-matrix
null=zeros(size(M));
Eins = eye(size(M));
SysMat=[null,Eins;-inv(M)*Kt,null];

% compute Eigenvalues
[V,D]=eig(SysMat);
temp_d = diag(D);
[nd, sortindex] = sort(temp_d);
temp_v = V(:,sortindex);
temp_f = imag(nd)/(2*pi);

% analysis method
lamda = zeros(18, 1);
omega = zeros(18, 1);
f = zeros(18, 1);
for k = 1: 18
    lamda(k) = (2 * k - 1) * pi / (2 * l);
    omega(k) = lamda(k)*sqrt(E/rho);

```

```

    f(k) = omega(k)/(2*pi);
end

```

### Elementrechnung quadratischer Ansatz

```

#####
% Elementroutine
#####
function [Kte,Me] = Elementroutine_quadra(A,E,mu,le)
% Elementroutine: compute Kte, Me
% define empty Kte
Kte=zeros(3);
% define empty Me
Me=zeros(3);

% define sampling points for Gauss-quadrature
xiVec=[-sqrt(1/3),sqrt(1/3)];
% weights for sampling points of Gauss-quadrature
wVec =[1,1];

for i=1:length(xiVec)
    xi=xiVec(i);
    w=wVec(i);

    % define N, B vector
    N=[ xi^2/2-xi/2  1-xi^2  xi/2+xi^2/2];
    Nx=[ xi-0.5  -2*xi  0.5+xi ]*(2/le);

    % compute Kte and Me for sampling point of Gauss-integration
    Me=Me + mu * (N' * N) * w;
    Kte=Kte + E * A * (Nx' * Nx) * w;
end

end

```

## Anhang B: Programm Balkenelemente

### Hauptprogramm kubischer Ansatz

```

clear
clc
close
#####
% main Program
#####
% parameter
E=2.1e11;           % N/m^2
A=0.0001;           % m^2
l=10;               % m
rho=7850;            % Dichte in [kg/m^3]
mu=rho*A;           % Massenbelegung in [kg/m]
Nel=15;             % number of elements
Nno=Nel+1;          % number of nodes
le=l/Nel;           % length of an element
B=0.005;            % Breit von Balken
H=0.02;             % Hoehe von Balken
I=(B*(H^3))/12;     % Flaechentraegheitsmoment

% define empty matrice
Kt=zeros(Nno*2);    % empty global stiffness-matrix
M=zeros(Nno*2);     % empty global mass-matrix

% call element routinesmbclient
[Kte,Me] = Elementroutine_Balken(A,E,rho,le,I);

% loop over every element
for j=1: 2 : length(M)-2
    M(j : j+3, j : j+3) = M(j : j+3, j : j+3) + Me;
    Kt(j : j+3, j : j+3) = Kt(j : j+3, j : j+3) + Kte;
end

% implementation of essential boundary conditions
Kt(1,:) = [ ];
Kt(:,1) = [ ];
M(1,:) = [ ];
M(:,1) = [ ];

Kt(1,:) = [ ];
Kt(:,1) = [ ];
M(1,:) = [ ];
M(:,1) = [ ];

% define system-matrix
null=zeros(size(M));

```

```

Eins = eye(size(M));
SysMat=[null,Eins;-inv(M)*Kt,null];

% compute Eigenvalues
[V,D]=eig(SysMat);

temp_d = diag(D);
[nd, sortindex] = sort(temp_d);
temp_v = V(:,sortindex);
temp_f = imag(nd)/(2*pi);

for i=1:Nel*2
    temp_f(i,:)=[];
end
% vector with node coordinates
lVec = zeros(Nno, 1);
for k = 1: Nno
    lVec(k)=l/Nel*(k-1);
end
% delete lower part von eigenvectors
EigMat=temp_v(1:length(M),:);
EigMat=imag(EigMat);
% delete unnecessary rows
for k = 1: length(M)/2
    EigMat(1+k,:)=[];
end
% delete unnecessary columns
for k = 1: length(M)
    EigMat(:,k)=[];
end
% fill up EigMat for first with zero displacement
EigMat=[zeros(1,length(M));EigMat];

figure
hold on
grid on
plot(lVec,-EigMat(:,1),'LineWidth',4);
plot(lVec,EigMat(:,2),'—','LineWidth',4);
plot(lVec,EigMat(:,3),':', 'LineWidth',4);
set(gca,'FontSize',24);
legend('1-Mode', '2-Mode', '3-Mode','Location','northwest');

% analysis method
lamda = zeros(Nel, 1);
omega = zeros(Nel, 1);
f = zeros(Nel, 1);
for k = 1: Nel
    switch k
        case 1

```

```

        lamda(k) = 1.87510 ;
    case 2
        lamda(k) = 4.69409 ;
    case 3
        lamda(k) = 7.85476 ;
    case 4
        lamda(k) = 10.99554;
    end
    if k >= 5
        lamda(k) = ((2*k-1)*pi)/2 ;
    end
    omega (k) = ((lamda(k))^2)*sqrt((E*I)/(rho*A*(l^4)));
    f(k)=omega(k)/(2*pi);
end

```

#### Elementrechnung kubischer Ansatz

```

#####
% Elementroutine
#####
function [Kte,Me] = Elementroutine_Balken(A,E,rho,le,I)
% Elementroutine: compute Kte, Me
% define empty Kte
Kte=zeros(4);
% define empty Me
Me=zeros(4);
% define sampling points for Gauss-quadrature
xiVec=[-sqrt(3/5),0,sqrt(3/5)];
% weights for sampling points of Gauss-quadrature
wVec=[5/9,8/9,5/9];
for i=1:length(xiVec)
    xi=xiVec(i);
    w =wVec(i);
    % zwei Freiheitsgrade
    N = [1/2-(3*xi)/4+(xi^3)/4    1/4-xi/4-(xi^2)/4+(xi^3)/4
         1/2+(3*xi)/4-(xi^3)/4    -1/4-xi/4+(xi^2)/4+(xi^3)/4];
    Nxx = [(3*xi)/2    -1/2+(3*xi)/2    -(3*xi)/2    1/2+(3*xi)
            /2]*((2/le)^2);
    % compute Kte and Me for sampling point of Gauss-integration
    Me=Me + rho * A * (N' * N) * w;
    Kte=Kte + E * I * (Nxx' * Nxx) * w;
end

end

```





## Anhang C: Programm Plattenelemente

### Hauptprogramm konforme Plattenelemente

```

clear
clc
close
#####
% main Program
#####
% parameter
E=2.1e11;          % N/m^2
A=0.0001;          % m^2
L=2;               % m length of plate
B=2;               % m wide of plate
rho=7850;           % Dichte in [kg/m^3]
mu=rho*A;          % Massenbelegung in [kg/m]
Nelx=15;            % number of elements per rand
Nely=15;
Nnox=Nelx+1;        % number of nodes per rand
Nnoy=Nely+1;
Nel_all=Nelx * Nely; % total elements
Nno_all=Nnox * Nnoy; % total nodes
lex=L/Nelx;         % length of an element
ley=B/Nely;
H=0.01;              % m Dicke von Platte
v=0.30;              % Poissionzahl

% define empty matrice
Kt=zeros(Nno_all*4); % empty global stiffnes-matrix
M=zeros(Nno_all*4);  % empty global mass-matrix

% call element routinesmbclient
[Kte,Me] = Elementroutine_Platten(H,E,rho,lex,ley,v);
% loop over every element
r=0;
a=zeros(4,1);
b=a;
c=[1 5 9 13];
d=[4 8 12 16];
for j=1 : Nely
    for k=1:Nelx
        a(1)=((j-1)*Nnox+k-1)*4+1;
        b(1)=a(1)+3;
        a(2)=((j-1)*Nnox+k)*4+1;
        b(2)=a(2)+3;
        a(3)=((j*Nnox)+k-1)*4+1;
        b(3)=a(3)+3;
        a(4)=((j*Nnox)+k)*4+1;
    
```

```

b(4)=a(4)+3;
for m=1:4
    for n=1:4
        M(a(m):b(m), a(n):b(n)) = M(a(m):b(m), a(n):b(n)) + Me(c(
            m):d(m), c(n):d(n));
        Kt(a(m):b(m), a(n):b(n)) = Kt(a(m):b(m), a(n):b(n)) + Kte
            (c(m):d(m), c(n):d(n));
    end
end
end
end

% implementation of essential boundary conditions
M(1:Nnox*4,:) = [];
M(:,1:Nnox*4) = [];

Kt(1:Nnox*4,:) = [];
Kt(:,1:Nnox*4) = [];

% define system-matrix
null=zeros(size(M));
Eins = eye(size(M));
SysMat=[null, Eins; -inv(M)*Kt, null];

% compute Eigenvalues
[V,D]=eig(SysMat);

temp_d = diag(D);
[nd, sortindex] = sort(temp_d);
temp_v = V(:, sortindex);
temp_f = imag(nd)/(2*pi);

%% Mode Plot
% vector with node coordinates
xVec = zeros(Nnox, 1);
yVec = zeros(Nnoy, 1);
for k = 1: Nnox
    xVec(k)=L/Nelx*(k-1);
end

for g = 1: Nnoy
    yVec(g)=B/Nely*(g-1);
end

% delete lower part von eigenvectors
EigMat=temp_v(1:length(M),:);
EigMat=imag(EigMat);
% delete unnecessary rows
for k = 1: length(M)/4

```

```

    EigMat(1+k:1+k+2,:)=[];
end
% delete unnecessary columns
for k = 1: length(M)
    EigMat(:,k)=[];
end
% fill up EigMat
p = zeros(Nnox,Nnoy-1,9);
for i = 1:9
    for j = 1:Nnoy-1
        for k = 1:Nnox
            p(k,j,i) = EigMat(k+(j-1)*Nnox,i);
        end
    end
end

for i = 1 : 9
    p1(:, :, i) = [zeros(Nnox,1),p(:, :, i)];
end

figure
hold on
for k = 1:9
    subplot(3,3,k);
    surf(yVec,xVec,p1(:, :, k));
    set(gca, 'FontSize', 20);
    title([num2str(k), '-Mode']);
end

```

### Elementrechnung konforme Plattenelemente

```

#####
% Elementroutine
#####
function [Kte,Me] = Elementroutine_Platten(H,E,rho,lex,ley,v)
% Elementroutine: compute Kte, Me
% define empty Kte
Kte=zeros(16);

% define empty Me
Me=zeros(16);

% define sampling points for Gauss-quadrature
xiVec=[-sqrt(3/7+2/7*sqrt(6/5)), -sqrt(3/7-2/7*sqrt(6/5)), sqrt(3/7-2/7*sqrt(6/5)), sqrt(3/7+2/7*sqrt(6/5))];
etaVec=[-sqrt(3/7+2/7*sqrt(6/5)), -sqrt(3/7-2/7*sqrt(6/5)), sqrt(3/7-2/7*sqrt(6/5)), sqrt(3/7+2/7*sqrt(6/5))];
% weights for sampling points of Gauss-quadrature
wVec=[(18-sqrt(30))/36, (18+sqrt(30))/36, (18+sqrt(30))/36, (18-sqrt(30))/36];

```

```

uVec=wVec;

for i=1:length(xiVec)
    x = xiVec(i);
    w = wVec(i);
    for j = 1:length(etaVec)
        y = etaVec(j);
        u = uVec(j);
        % define N, Nxx, Nyy, Nxy vector
N=[ (1/16)*(-1 + x)^2*(2 + x)*(-1 + y)^2*(2 + y) ...
    (1/16)*(-1 + x)^2*(1 + x)*(-1 + y)^2*(2 + y) ...
    (1/16)*(-1 + x)^2*(2 + x)*(-1 + y)^2*(1 + y) ...
    (1/16)*(-1 + x)^2*(1 + x)*(-1 + y)^2*(1 + y) ...
    (-(1/16))*(-2 + x)*(1 + x)^2*(-1 + y)^2*(2 + y) ...
    (1/16)*(-1 + x)*(1 + x)^2*(-1 + y)^2*(2 + y) ...
    (-(1/16))*(-2 + x)*(1 + x)^2*(-1 + y)^2*(1 + y) ...
    (1/16)*(-1 + x)*(1 + x)^2*(-1 + y)^2*(1 + y) ...
    (-(1/16))*(-1 + x)^2*(2 + x)*(-2 + y)*(1 + y)^2 ...
    (-(1/16))*(-1 + x)^2*(1 + x)*(-2 + y)*(1 + y)^2 ...
    (1/16)*(-1 + x)^2*(2 + x)*(-1 + y)*(1 + y)^2 ...
    (1/16)*(-1 + x)^2*(1 + x)*(-1 + y)*(1 + y)^2 ...
    (1/16)*(-2 + x)*(1 + x)^2*(-2 + y)*(1 + y)^2 ...
    (-(1/16))*(-1 + x)*(1 + x)^2*(-2 + y)*(1 + y)^2 ...
    (-(1/16))*(-2 + x)*(1 + x)^2*(-1 + y)*(1 + y)^2 ...
    (1/16)*(-1 + x)*(1 + x)^2*(-1 + y)*(1 + y)^2 ];

Nxx=[ (3*x*(-1 + y)^2*(2 + y))/(2*lex^2) ...
    ((-1 + 3*x)*(-1 + y)^2*(2 + y))/(2*lex^2) ...
    (3*x*(-1 + y)^2*(1 + y))/(2*lex^2) ...
    ((-1 + 3*x)*(-1 + y)^2*(1 + y))/(2*lex^2) ...
    -((3*x*(-1 + y)^2*(2 + y))/(2*lex^2)) ...
    ((1 + 3*x)*(-1 + y)^2*(2 + y))/(2*lex^2) ...
    -((3*x*(-1 + y)^2*(1 + y))/(2*lex^2)) ...
    ((1 + 3*x)*(-1 + y)^2*(1 + y))/(2*lex^2) ...
    -((3*x*(-2 + y)*(1 + y)^2)/(2*lex^2)) ...
    -(((1 + 3*x)*(-2 + y)*(1 + y)^2)/(2*lex^2)) ...
    (3*x*(-1 + y)*(1 + y)^2)/(2*lex^2) ...
    ((-1 + 3*x)*(-1 + y)*(1 + y)^2)/(2*lex^2) ...
    (3*x*(-2 + y)*(1 + y)^2)/(2*lex^2) ...
    -(((1 + 3*x)*(-2 + y)*(1 + y)^2)/(2*lex^2)) ...
    -((3*x*(-1 + y)*(1 + y)^2)/(2*lex^2)) ...
    ((1 + 3*x)*(-1 + y)*(1 + y)^2)/(2*lex^2) ];

Nyy=[ (3*(-1 + x)^2*(2 + x)*y)/(2*ley^2) ...
    (3*(-1 + x)^2*(1 + x)*y)/(2*ley^2) ...
    ((-1 + x)^2*(2 + x)*(-1 + 3*y))/(2*ley^2) ...
    ((-1 + x)^2*(1 + x)*(-1 + 3*y))/(2*ley^2) ...
    -((3*(-2 + x)*(1 + x)^2*y)/(2*ley^2)) ...
    (3*(-1 + x)*(1 + x)^2*y)/(2*ley^2) ...

```

```

-((( -2 + x)*(1 + x)^2*(-1 + 3*y))/(2*ley^2)) ...
((-1 + x)*(1 + x)^2*(-1 + 3*y))/(2*ley^2) ...
-((3*(-1 + x)^2*(2 + x)*y)/(2*ley^2)) ...
-((3*(-1 + x)^2*(1 + x)*y)/(2*ley^2)) ...
((-1 + x)^2*(2 + x)*(1 + 3*y))/(2*ley^2) ...
((-1 + x)^2*(1 + x)*(1 + 3*y))/(2*ley^2) ...
(3*(-2 + x)*(1 + x)^2*y)/(2*ley^2) ...
-((3*(-1 + x)*(1 + x)^2*y)/(2*ley^2)) ...
-((( -2 + x)*(1 + x)^2*(1 + 3*y))/(2*ley^2)) ...
((-1 + x)*(1 + x)^2*(1 + 3*y))/(2*ley^2) ];

Nxy=[ (9*(-1 + x^2)*(-1 + y^2))/(4*lex*ley) ...
(3*(-1 + x)*(1 + 3*x)*(-1 + y^2))/(4*lex*ley) ...
(3*(-1 + x^2)*(-1 + y)*(1 + 3*y))/(4*lex*ley) ...
((-1 + x)*(1 + 3*x)*(-1 + y)*(1 + 3*y))/(4*lex*ley) ...
-((9*(-1 + x^2)*(-1 + y^2))/(4*lex*ley)) ...
(3*(1 + x)*(-1 + 3*x)*(-1 + y^2))/(4*lex*ley) ...
-((3*(-1 + x^2)*(-1 + y)*(1 + 3*y))/(4*lex*ley)) ...
((1 + x)*(-1 + 3*x)*(-1 + y)*(1 + 3*y))/(4*lex*ley) ...
-((9*(-1 + x^2)*(-1 + y^2))/(4*lex*ley)) ...
-((3*(-1 + x)*(1 + 3*x)*(-1 + y^2))/(4*lex*ley)) ...
(3*(-1 + x^2)*(1 + y)*(-1 + 3*y))/(4*lex*ley) ...
((-1 + x)*(1 + 3*x)*(1 + y)*(-1 + 3*y))/(4*lex*ley) ...
(9*(-1 + x^2)*(-1 + y^2))/(4*lex*ley) ...
-((3*(1 + x)*(-1 + 3*x)*(-1 + y^2))/(4*lex*ley)) ...
-((3*(-1 + x^2)*(1 + y)*(-1 + 3*y))/(4*lex*ley)) ...
((1 + x)*(-1 + 3*x)*(1 + y)*(-1 + 3*y))/(4*lex*ley) ];

% compute Kte and Me for sampling point of Gauss-integration
Me=Me + rho * H * (N' * N) * w * u;
Kte=Kte + ((E*H^3)/(12*(1 - v^2))) * ((Nxx' * Nxx)+v*((Nxx'*
Nyy)+(Nyy'*Nxx))+(Nyy'*Nyy)+2*(1 - v)*(Nxy'*Nxy)) * w * u;
end
end

end

```