

Studienarbeit STUD-367

**Anwendung der
dynamischen Spannungsanalyse
auf die Bildfehlerkompensation
bei optischen Systemen**

von
Johannes Störkle

Betreuer: Prof. Dr.-Ing. Prof. E.h. P. Eberhard
Dipl.-Ing. N. Wengert
Dipl.-Ing. C. Tobias

Universität Stuttgart
Institut für Technische und Numerische Mechanik
Prof. Dr.-Ing. Prof. E.h. P. Eberhard

November 2011

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Aufbau der Arbeit	2
2	Theoretische Grundlagen	3
2.1	Mehrkörperdynamik	3
2.1.1	Elastische Mehrkörpersysteme	3
2.1.2	Modellreduktion auf Basis modaler Reduktion	6
2.1.3	Dynamische Spannungsberechnung modellreduzierter Körper	8
2.1.4	MatMorembs	10
2.1.5	Neweul- M^2	11
2.2	Optik	11
2.2.1	Raytracing	12
2.2.2	Aberrationen	12
2.2.3	OM-Sim	14
3	Prozessketten und Implementierungen	16
3.1	Prozesskette einer klassischen EMKS-Simulation	16
3.2	Prozesskette mit Spannungsanalyse	19
3.2.1	Stand vor der Implementierung	19
3.2.2	Implementierung der dynamischen Spannungsanalyse in <i>Neweul-M^2</i>	21

3.2.3	Beispiel für die dynamischen Spannungsanalyse in Neweul-M ²	25
4	Mechanische Simulation	28
4.1	Modelldefinition des elastischen Linsensystems	29
4.2	Erzeugung der elastischen Platte	29
4.2.1	Platten-Modellierung in Ansys	29
4.2.2	Modellreduktion und Spannungsmodenberechnung	30
4.2.3	Validierung der dynamischen Spannungsanalyse	31
4.3	Simulation des elastischen Linsensystems	33
5	Optische Simulation	36
5.1	Modellierung in OM-Sim	36
5.2	Bildfehler-Kompensation	37
5.2.1	Mechanisches Modell des optisches Systems	38
5.2.2	Regelungstechnisches Modell	39
5.2.3	Entwurf eines statischen Reglers	39
5.3	Simulation mit Bildfehler-Kompensation	41
5.3.1	Lineare Störung	44
5.3.2	Harmonische Störung	44
5.3.3	Zufällige Störung	47
6	Zusammenfassung und Ausblick	49
	Anhang	50
A.1	Dynamische Spannungsanalyse eines Krylov-reduzierten Modells .	50
A.2	Inhalt der CD-ROM	54
	Abbildungsverzeichnis	55

Kapitel 1

Einleitung

1.1 Motivation

Für die Herstellung moderner Halbleiterbauelemente werden im Bereich der Fotolithographie hochpräzise Objektive eingesetzt, welche kleinste Strukturen auf eine lichtempfindliche Schicht projizieren. In der Regel handelt es sich dabei um ein mit Fotolack beschichtetes Substrat (Waver), welches durch Belichten, Entwickeln und Ätzen zu einem fortschrittlichen Mikrochip verarbeitet wird.

Für die zuverlässige Erzeugung einer scharfen Abbildung spielt die Genauigkeit des optischen Systems eine entscheidende Rolle. Während des Belichtungsprozesses sollten deshalb Störungen aufgrund dynamischer Anregungen detektiert und ausgeglichen werden. Im Falle eines Hochleistungsobjektivs kann diese Anregung durch Bodenvibrationen, Kühlergeräusche oder von dem sogenannten Waferstepper selbst hervorgerufen werden. Neben Ansätzen zur Versteifung oder Dämpfung der Lagerung gibt es auch die Möglichkeit einen entstandenen Bildfehler durch die aktive Verformung einer Linse zu kompensieren [WengertEberhard11]. Dabei wird ein gegensätzlicher Bildfehler erzeugt, der durch die Änderung des Spannungsniveaus in der deformierbaren Linse hervorgerufen wird.

In dieser Arbeit steht speziell der Spannungseinfluss durch dynamische Deformationen auf den Bildfehler im Vordergrund. Dieser wird simulativ erfasst und die Spannungen im Linsenmaterial können dann gezielt eingesetzt werden, um andere Bildfehler zu kompensieren.

In der hier verfolgten Methode wird zunächst durch eine mechanische Simulation eines vereinfachten Objektivs die Linsen-Bewegungen und Spannungen analysiert und visualisiert. Das Linsensystem ist als elastisches Mehrkörpersystem (EMKS) modelliert. Als zweiter Schritt wird durch eine kleine statische Linsenverschiebung in einer optischen Simulation des Modells ein Bildfehler rekon-

struiert, welcher durch ein deformierbares Element ausgeglichen wird.

1.2 Aufbau der Arbeit

Die Arbeit ist wie folgt gegliedert:

- *Kapitel 2* erläutert einige Grundlagen zur Mehrkörperdynamik und Optik. Hierbei steht die mathematische Beschreibung von elastischen Mehrkörpersystemen sowie die dynamische Spannungsberechnung modellreduzierter Körper im Vordergrund. Es werden außerdem die am Institut für Technische und Numerische Mechanik der Universität Stuttgart (ITM) entwickelten *Matlab*-Programme *MatMorembs*, *Neweul-M²* und *OM-Sim* vorgestellt.
- *Kapitel 3* geht auf die neuen Implementierungen der automatisierten Spannungsmodenberechnung in *MatMorembs* und der Spannungs-Animation in *Neweul-M²* ein. Anhand eines Anwendungsbeispiels wird die Durchführung einer Spannungsanalyse verdeutlicht.
- *Kapitel 4* analysiert ein elastisches Objektiv-Modell mit der Anwendung der dynamischen Spannungsanalyse. Dieses mechanische System wird in *Neweul-M²* aufgebaut.
- *Kapitel 5* zeigt die optische Simulation eines vereinfachten statischen Starrkörper-Objektiv-Modells in *OM-Sim* auf. Dabei werden optische Fehler kompensiert, die durch Linsenverschiebungen zustande kamen.
- *Kapitel 6* gibt nach einer kurzen Zusammenfassung ein Ausblick auf mögliche Erweiterungen.

Kapitel 2

Theoretische Grundlagen

In diesem Kapitel werden einige theoretische Grundlagen erläutert, welche die Basis für die folgende Arbeit bilden. Nur die wichtigsten Gleichungen und Erklärungen werden dargestellt. Detaillierte Herleitungen und zugehörige Erläuterungen sind der zitierten Literatur zu entnehmen.

2.1 Mehrkörperdynamik

In der klassischen Mehrkörperdynamik sind starre Körper durch masselose Bindungs- und Kopplungselemente miteinander und mit ihrer Umgebung verbunden und bilden so ein mechanisches System, das mit der Methode der Mehrkörpersysteme (MKS) beschrieben werden kann, vgl. [SchwertassekWallrapp99]. Aufgrund von zunehmenden Leichtbauanforderungen und höheren Arbeitsgeschwindigkeiten können Körper nicht immer als starr angenommen werden. Das Vernachlässigen von Deformationen führt insbesondere bei hochdynamischen Vorgängen zu unrealistischen Simulationsergebnissen. Deshalb ist es notwendig, zumindest einzelne Körper mit signifikanten Verformungen in einem System elastisch zu beschreiben.

2.1.1 Elastische Mehrkörpersysteme

Um neben der nichtlinearen Starrkörperbewegung auch die Verformungen einzelner Körper zu beschreiben, eignet sich die Modellierung als elastisches Mehrkörpersystem (EMKS). Sind die elastischen Verformungen im Vergleich zur Starrkörperbewegung sehr klein, empfiehlt sich für die kinematische Beschreibung die Methode des bewegten Referenzsystems entsprechend [SchwertassekWallrapp99]. Hierbei wird die Bewegung eines elastischen Körpers in eine nichtlineare Starrkörperbewegung überführt.

perbewegung des Referenzkoordinatensystems K_R bezüglich des Inertialsystems K_I und in eine elastische Verformung \mathbf{u}_P bezüglich des Referenzkoordinatensystems K_R aufgeteilt, vgl. Abbildung 2.1,

$$\mathbf{r}_P = \mathbf{r}_{IR} + \mathbf{R} + \mathbf{u}_P. \quad (2.1)$$

Während \mathbf{r}_{IR} die Lage von K_R und \mathbf{R} die Lage des Punktes P im unverformten Zustand beschreibt, stellt \mathbf{u}_P die elastische Verformung dar. Der Punkt P liegt im Ursprung des lokalen Koordinatensystems K_P und dessen Bewegung \mathbf{r}_P kann mit folglich mit Gl. (2.1) beschrieben werden. Die Definition des Referenzsystems kann nach [SchwertassekWallrapp99] als Sehnen-, Tangenten- oder Buckenssystem erfolgen. Bei letzterem wird die elastische Deformation im Mittel minimiert.

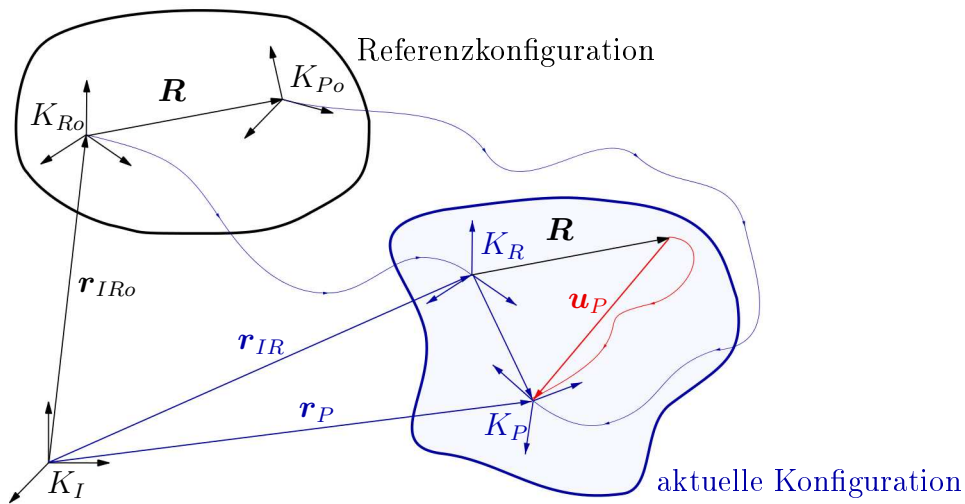


Abbildung 2.1: Kinematik eines elastischen Körpers.

Eine weit verbreitete Methode zur Modellierung der elastischen Deformationen stellt die lineare Finite Elemente Methode (FEM) dar. Wie in [Willner04] und [Bathe96] beschrieben, ist die FEM ein diskretes Verfahren zur numerischen Analyse komplexer Feldprobleme, die mittels partiellen Differentialgleichungen formuliert sind. Die Beschreibung der elastischen Verformungen erfolgt dabei auf der Basis von Ansatzfunktionen, die jeweils aus einer Linearkombination von sogenannten Formfunktionen elementweise definiert sind. Fügt man diese zu einer globalen Ansatzfunktion Φ zusammen, wird mit dem Vektor \mathbf{q} der Knotenverschiebungen und -verdrehungen die elastische Verformung \mathbf{u}_P vgl. Gl. (2.2) formuliert,

$$\mathbf{u}_P = \Phi \cdot \mathbf{q}. \quad (2.2)$$

Die linearisierten strukturmechanischen Bewegungsgleichungen ergeben sich auf Basis des d'Alembertschen Prinzips nach [Shabana98] zu

$$\mathbf{M}_e \cdot \ddot{\mathbf{q}} + \mathbf{D}_e \cdot \dot{\mathbf{q}} + \mathbf{K}_e \cdot \mathbf{q} = \mathbf{h}_e. \quad (2.3)$$

Darin beschreibt \mathbf{M}_e die Massenmatrix, \mathbf{D}_e die Dämpfungsmatrix und \mathbf{K}_e die Steifigkeitsmatrix. Der Vektor \mathbf{q} stellt unverändert die Knotenkoordinaten und \mathbf{h}_e die äußeren Kräfte bzw. Momente des FE-Modells dar.

Betrachtet man das FE-Modell abstrakt als dynamisches System mit mehreren Ein- und Ausgängen, so lässt sich die strukturmechanische Bewegungsgleichung auch als lineares Differentialgleichungssystem zweiter Ordnung im Zeitbereich, vgl. [Bathe96] formulieren

$$\begin{aligned} \mathbf{M}_e \cdot \ddot{\mathbf{q}}(t) + \mathbf{D}_e \cdot \dot{\mathbf{q}}(t) + \mathbf{K}_e \cdot \mathbf{q}(t) &= \mathbf{B}_e \cdot \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}_e \cdot \mathbf{q}(t). \end{aligned} \quad (2.4)$$

Die Eingangsmatrix \mathbf{B}_e ist eine Boolesche Matrix, welche den zeitabhängigen Kraftvektor $\mathbf{u}(t)$ auf die Freiheitsgrade (FHG) verteilt. Weiterhin wird eine Ausgangsmatrix \mathbf{C}_e zur Berechnung der relevanten Ausgänge $\mathbf{y}(t)$ gewählt.

Mit obigen Gleichungen und kinematischen Beziehungen kann durch die Anwendung des Jourdainischen Prinzips nach [SchwertassekWallrapp99] die Bewegungsgleichung eines elastischen Einzelkörpers in einem EMKS hergeleitet werden

$$\underbrace{\left[\begin{array}{cc|c} m\mathbf{I} & m\tilde{\mathbf{c}}^T(\mathbf{q}) & \mathbf{C}_t^T \\ m\tilde{\mathbf{c}}(\mathbf{q}) & \mathbf{J}(\mathbf{q}) & \mathbf{C}_r^T \\ \hline \mathbf{C}_t & \mathbf{C}_r & \mathbf{M}_e \end{array} \right]}_{\mathbf{M}} \cdot \underbrace{\begin{bmatrix} \mathbf{a} \\ \boldsymbol{\alpha} \\ \ddot{\mathbf{q}} \end{bmatrix}}_{\mathbf{z}_{III}} = \underbrace{\begin{bmatrix} \mathbf{h}_t^a(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{h}_r^a(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{h}_e^a(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix}}_{\mathbf{h}^a} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -\mathbf{K}_e \cdot \mathbf{q} - \mathbf{D}_e \cdot \dot{\mathbf{q}} \end{bmatrix}}_{\mathbf{h}^e}. \quad (2.5)$$

In der Darstellung nach Gl. (2.5) enthält die linke obere 2×2 Block-Matrix der Massenmatrix \mathbf{M} die Starrkörperbewegungen des elastischen Körpers. Darin sind m die Masse, $\mathbf{J}(\mathbf{q})$ der Trägheitstensor und $\tilde{\mathbf{c}}$ die Lage des Schwerpunktes. Die Koppelmatrizen \mathbf{C}_t und \mathbf{C}_r beschreiben den Zusammenhang der Starrkörpertranslation bzw. der Starrkörperrotation und der elastischen Verformung. Die Beschleunigung kann mit Hilfe der in \mathbf{z}_{III} zusammengefassten translatorischen Beschleunigung \mathbf{a} , der Winkelbeschleunigung $\boldsymbol{\alpha}$ und der zweiten zeitlichen Ableitungen der elastischen Koordinaten $\ddot{\mathbf{q}}$ beschrieben werden.

Die Matrizen \mathbf{M}_e , \mathbf{K}_e und \mathbf{D}_e stellen entsprechend Gleichung (2.3) das elastische Systemverhalten dar. Sie sind keine Funktionen der elastischen Koordinaten und können daher in einem Preprocessing-Schritt mit Hilfe der FEM berechnet werden. Im Vektor \mathbf{h}^a sind die Zentrifugal- und Coriolis-Kräfte, die Volumen- und

Oberflächenkräfte und aller am Körper angreifenden diskreten Kräfte / Momente enthalten. Die Gleichungen zur Berechnung der einzelnen Vektoren und Matrizen sind in [SchwertassekWallrapp99] explizit aufgeführt. Die inneren Kräfte des elastischen Körpers sind im Vektor \mathbf{h}^e dargestellt.

2.1.2 Modellreduktion auf Basis modaler Reduktion

Da bei der Modellbildung flexibler Körper in FEM-Programmen Modelle mit einer hohen Anzahl an FHG und damit einer hohen Komplexität entstehen, ist vor dem Integrieren des elastischen Körpers in das EMKS eine Modellreduktion sinnvoll. Durch verschiedene Reduktionsverfahren kann die Anzahl der Freiheitsgrade verringert werden. Das Hauptziel ist, das Verhalten des Originalsystems bestmöglich zu approximieren.

Ein allgemeines Verfahren der Modellreduktion ist die Projektion des Systems in einen Unterraum mit reduzierter Dimension. Dieses Verfahren beruht darauf, dass ein lineares, zeitinvariantes System, vgl. Gl. (2.4), der Dimension n durch ein System der Dimension $m < n$ approximiert wird. Dafür wird der Vektor \mathbf{q} in einen Vektor von reduzierten FHG $\bar{\mathbf{q}}$ des Unterraums \mathcal{V} transformiert, vgl. Gl. (2.6). Dies erfolgt mit Hilfe der Projektionsmatrix $\mathbf{V} \in \mathbb{R}^{n \times m}$, welche den Unterraum aufspannt,

$$\mathbf{q}(t) \approx \mathbf{V} \bar{\mathbf{q}}(t). \quad (2.6)$$

Das Einsetzen der Gl. (2.6) in (2.4) ergibt das überbestimmte Gleichungssystem

$$\begin{aligned} \mathbf{M}_e \cdot \mathbf{V} \cdot \ddot{\bar{\mathbf{q}}}(t) + \mathbf{D}_e \cdot \mathbf{V} \cdot \dot{\bar{\mathbf{q}}}(t) + \mathbf{K}_e \cdot \mathbf{V} \cdot \bar{\mathbf{q}}(t) &= \mathbf{B}_e \cdot \mathbf{u}(t) + \mathbf{r}(t) \\ \bar{\mathbf{y}}(t) &= \mathbf{C}_e \cdot \mathbf{V} \cdot \bar{\mathbf{q}}(t). \end{aligned} \quad (2.7)$$

Da durch die Reduktion das Systemverhalten nur noch näherungsweise beschrieben wird, verbleibt ein Residuum $\mathbf{r}(t) \neq \mathbf{0}$. Mit der Definition eines zweiten Unterraum \mathcal{W} bzw. der Projektionsmatrix $\mathbf{W} \in \mathbb{R}^{m \times n}$ unter der Orthogonalitätsbedingung $\mathbf{W}^T \mathbf{r}(t) = \mathbf{0}$, kann das Residuum jedoch eliminiert werden, siehe [Lehner07]. Durch Linksmultiplikation der Gl. (2.7) mit \mathbf{W}^T ergibt sich das reduzierte System explizit zu

$$\begin{aligned} \mathbf{W}^T \cdot \mathbf{M}_e \cdot \mathbf{V} \cdot \ddot{\bar{\mathbf{q}}} + \mathbf{W}^T \cdot \mathbf{D}_e \cdot \mathbf{V} \cdot \dot{\bar{\mathbf{q}}} + \mathbf{W}^T \cdot \mathbf{K}_e \cdot \mathbf{V} \cdot \bar{\mathbf{q}} &= \mathbf{W}^T \cdot \mathbf{B}_e \cdot \mathbf{u} \\ \bar{\mathbf{y}} &= \mathbf{C}_e \cdot \mathbf{V} \cdot \bar{\mathbf{q}}. \end{aligned} \quad (2.8)$$

Zur Bildung der Projektionsmatrizen \mathbf{V} und \mathbf{W} gibt es verschiedene Modellreduktionsverfahren, davon sind drei beispielhaft aufgeführt, nämlich

- Reduktion auf Basis von Eigenmoden und

- Reduktion auf Basis von Krylov-Unterräumen und Moment-Matching
- Reduktion auf Basis von Balancierung und balancierter Approximation.

In dieser Arbeit wird eine modale Reduktion verwendet, und es gilt $\mathbf{W} = \mathbf{V}$, man bezeichnet diese Projektion als orthogonal, vgl. [FehrEberhard09]. Die Grundidee der modalen Reduktion besteht darin, dass m Eigenvektoren in einer Modalmatrix $\bar{\Phi} = [\phi_1 \ \dots \ \phi_m]$ zusammengefasst werden und die Projektion durch $\mathbf{V} := \bar{\Phi}$ und $\mathbf{W} = \mathbf{V}$ in Form von Gl. (2.6) durchgeführt wird.

Dabei sind die niederfrequenten ersten Eigenmoden für die Abbildung des Systemverhaltens oft relevanter als hochfrequenten, vgl. [Lehner07]. Der große Vorteil der modalen Reduktion stellt die einfache Bestimmung der Projektionsmatrix als die Modalmatrix dar, die sich aus einem Eigenwertproblem berechnen lässt. Eine Prinzipdarstellung der modalen Reduktion auf Basis der ersten zwei Eigenmoden am Beispiel einer ebenen Platte ist in Abbildung 2.2 illustriert.

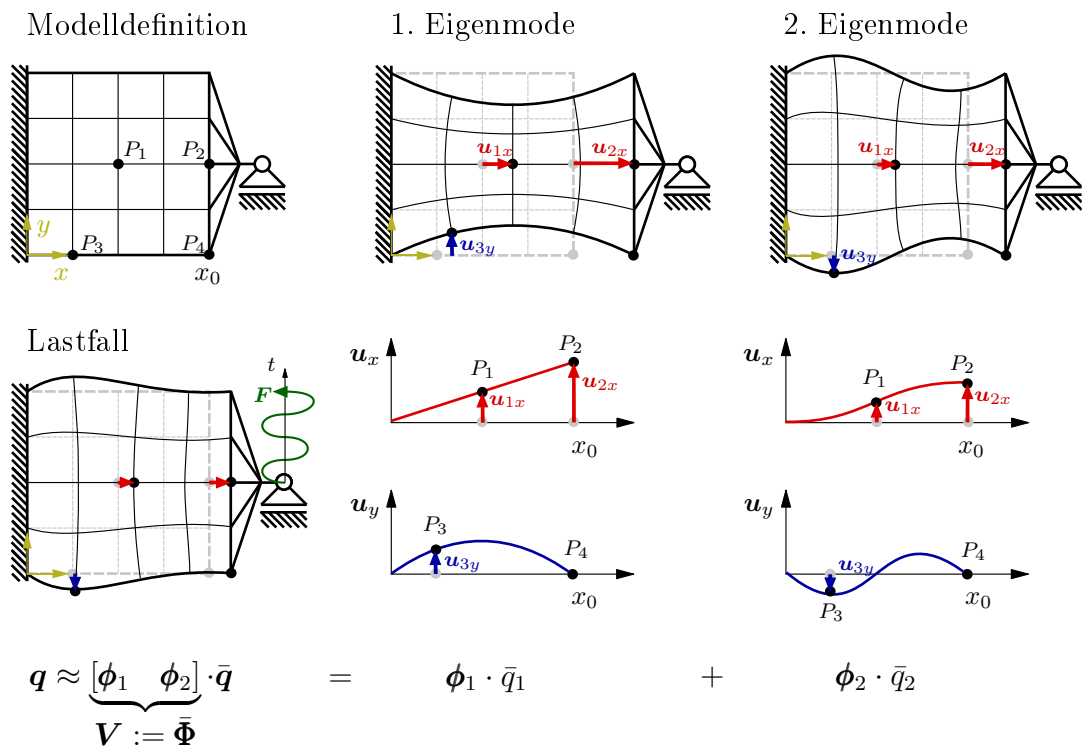


Abbildung 2.2: Prinzipdarstellung der modalen Reduktion auf Basis der ersten zwei Eigenmoden an dem Beispiel einer ebenen Platte.

Baut man das reduzierte System nach Gleichung (2.8) mit der Bedingung $\mathbf{W} = \mathbf{V}$ in die Gl. (2.5) ein, so ergibt sich die Bewegungsgleichung eines EMKS auf der

Basis von modaler Reduktion

$$\begin{bmatrix} m\mathbf{I} & m\tilde{\mathbf{c}}^T & \mathbf{C}_t^T \cdot \mathbf{V} \\ m\tilde{\mathbf{c}} & \mathbf{J} & \mathbf{C}_r^T \cdot \mathbf{V} \\ \mathbf{V}^T \cdot \mathbf{C}_t & \mathbf{V}^T \cdot \mathbf{C}_r & \mathbf{V}^T \cdot \mathbf{M}_e \cdot \mathbf{V} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{a}_i \\ \boldsymbol{\alpha}_i \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_t^a \\ \mathbf{h}_r^a \\ \mathbf{V}^T \cdot \mathbf{h}_e^a \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -\mathbf{V}^T \cdot \mathbf{K}_e \cdot \mathbf{V} \cdot \bar{\mathbf{q}} - \mathbf{V}^T \cdot \mathbf{D}_e \cdot \mathbf{V} \cdot \dot{\mathbf{q}} \end{bmatrix}. \quad (2.9)$$

2.1.3 Dynamische Spannungsrechnung modellreduzierter Körper

Sind bei der Analyse eines EMKS neben den Verschiebungen auch die Spannungen in einem Bauteil von hoher Bedeutung, so ist es notwendig, diese auf Basis der kontinuumsmechanischen Grundgleichungen vgl. [SchwertassekWallrapp99] zu ermitteln. Insbesondere bei der Verwendung eines modellreduzierten Körpers kann die Spannung durch eine dynamische Spannungsanalyse unter Verwendung von sogenannten Spannungsmoden, vgl. [TobiasEberhard11], berechnet werden. Im Folgenden sind die wichtigsten Zusammenhänge erläutert.

Wie schon in Gl. (2.2) angedeutet, können in einem FE-Modell die Verschiebungen $\mathbf{u} = [u_x \ u_y \ u_z]^T$ eines materiellen Punktes \mathbf{R} explizit durch den Ansatz

$$\mathbf{u}(\mathbf{R}, t) = \Phi(\mathbf{R}) \cdot \mathbf{q}(t) \quad (2.10)$$

beschrieben werden. Die Matrix $\Phi(\mathbf{R})$ enthält die globalen Ansatzfunktionen der FEM und $\mathbf{q}(t)$ ist der zeitabhängige Vektor der Knotenpunktverschiebungen. Aus der Annahme eines linear-elastischen Materialverhaltens folgt, dass die Dehnungen $[\epsilon_{xx} \ \epsilon_{yy} \ \epsilon_{zz}]^T$ und Schubverzerrungen $[\gamma_{xy} \ \gamma_{xz} \ \gamma_{yz}]^T$ nur linear von den Verschiebungen abhängen. Der Dehnungsvektor kann somit durch die partiellen Ortsableitungen der Knotenverschiebungen, mit Hilfe der Differentialoperatorenmatrix \mathbf{D} als

$$\boldsymbol{\epsilon}(\mathbf{R}, t) = \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{bmatrix} = \underbrace{\begin{bmatrix} \partial/\partial r_x & 0 & 0 \\ 0 & \partial/\partial r_y & 0 \\ 0 & 0 & \partial/\partial r_z \\ \partial/\partial r_y & \partial/\partial r_x & 0 \\ \partial/\partial r_z & 0 & \partial/\partial r_x \\ 0 & \partial/\partial r_z & \partial/\partial r_y \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}}_{\mathbf{u}(\mathbf{R}, t)}. \quad (2.11)$$

ausgedrückt werden. Der Operator \mathbf{D} wird dabei auf \mathbf{u} angewendet, was durch die geschweiften Klammern angedeutet wird. Der lineare Zusammenhang zwischen

den Spannungen und Dehnungen wird durch die Materialmatrix

$$\mathbf{C} = \frac{E}{1 + \nu} \begin{bmatrix} \frac{1-\nu}{1-2\nu} & \frac{\nu}{1-2\nu} & \frac{\nu}{1-2\nu} & 0 & 0 & 0 \\ & \frac{1-\nu}{1-2\nu} & \frac{\nu}{1-2\nu} & 0 & 0 & 0 \\ & & \frac{1-\nu}{1-2\nu} & 0 & 0 & 0 \\ & & & \frac{1}{2} & 0 & 0 \\ & \text{sym.} & & & \frac{1}{2} & 0 \\ & & & & & \frac{1}{2} \end{bmatrix} \quad (2.12)$$

basierend auf den konstitutiven Gleichungen beschrieben. Die Variable E stellt das Elastizitätsmodul und ν die Querkontraktionszahl dar. Unter der Verwendung von Gl. (2.10) und Gl. (2.11) lassen sich die Spannungen an dem Punkt \mathbf{R} folglich mit

$$\boldsymbol{\sigma}(\mathbf{R}, t) = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yz} \end{bmatrix} = \mathbf{C} \cdot \boldsymbol{\epsilon}(\mathbf{R}, t) \quad (2.13)$$

$$= \mathbf{C} \cdot \mathbf{D}\{\boldsymbol{\Phi}(\mathbf{R}) \cdot \mathbf{q}(t)\} = \underbrace{\mathbf{C} \cdot \mathbf{D}\{\boldsymbol{\Phi}(\mathbf{R})\}}_{\bar{\boldsymbol{\Phi}}_{\sigma}(\mathbf{R})} \cdot \mathbf{q}(t) \quad (2.14)$$

berechnen. Für die Umformulierung der Differential-Operation in Gl. (2.14) sind weitere Zwischenschritte notwendig, auf die hier nicht näher eingegangen wird. Konform zu [TobiasEberhard11] lassen sich die Materialmatrix und der Differentialoperator angewandt auf die Ansatzfunktionen zu einer Matrix $\bar{\boldsymbol{\Phi}}_{\sigma}(\mathbf{R})$ zusammenfassen. Die Spalten werden dabei als Spannungsmoden bezeichnet.

Die entsprechende Spannungsmode $\phi_{\sigma_i}(\mathbf{R})$ erhält man aus Gl. (2.13), indem der i -te FHG in \mathbf{q} zu Eins gesetzt wird, während die anderen gleich Null gesetzt werden. Mit dieser Vorgehensweise lassen sich die entsprechenden Spalten von $\bar{\boldsymbol{\Phi}}_{\sigma}(\mathbf{R})$ nacheinander berechnen, siehe Abbildung 2.3 links oben. Analog zu den Spannungsmoden des vollständigen Modells werden in [TobiasEberhard11] die sogenannten reduzierten Spannungsmoden $\bar{\boldsymbol{\Phi}}_{\sigma}(\mathbf{R})$ mit Hilfe der lokalen Projektionsmatrix $\mathbf{V}(\mathbf{R})$ aus Gl. (2.6) definiert,

$$\boldsymbol{\sigma}(\mathbf{R}, t) = \bar{\boldsymbol{\Phi}}_{\sigma}(\mathbf{R}) \cdot \mathbf{q} \approx \underbrace{\bar{\boldsymbol{\Phi}}_{\sigma}(\mathbf{R}) \cdot \mathbf{V}}_{\bar{\boldsymbol{\Phi}}_{\sigma}(\mathbf{R})} \cdot \bar{\mathbf{q}}(t). \quad (2.15)$$

In [TobiasEberhard11] wird vorgeschlagen, die Spalten der globalen Projektionsmatrix als Zwangsbedingungen auf die Knoten-FHG \mathbf{q} zu interpretieren. Somit lässt sich die globale reduzierte Spannungsmodenmatrix $\bar{\boldsymbol{\Phi}}_{\sigma}$ mittels eines

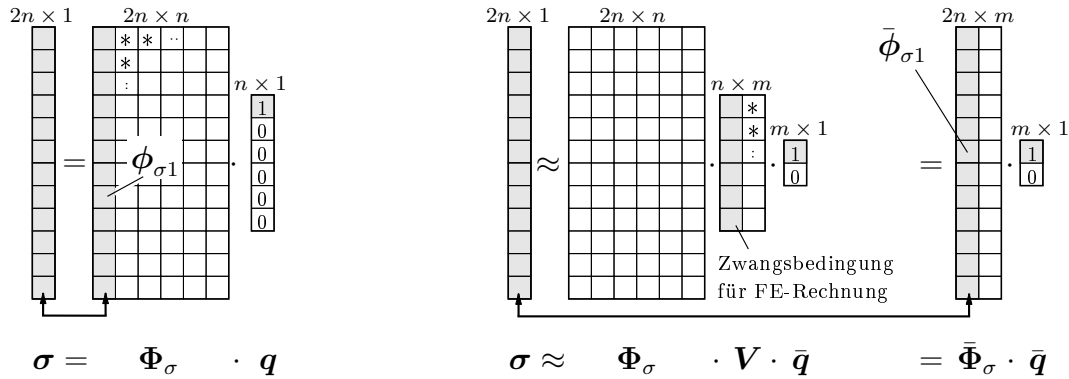


Abbildung 2.3: Links: Schematische Matrizen-Berechnung einer ersten Spannungsmode des unreduzierten Modells. Rechts: Berechnung einer ersten reduzierten Spannungsmode mit Hilfe eines FE-Programms.

FE-Solvers spaltenweise berechnen. Eine schematische Darstellung ist in Abbildung 2.3 dargestellt.

Im Folgenden werden die für die Berechnung der Mehrkörperdynamik verwendeten Softwarepakete *Morembs* und *Neweul-M²* vorgestellt.

2.1.4 MatMorembs

Morembs ist ein am ITM entwickelter Präprozessor für die Modellreduktion elastischer Körper. Im Softwarepaket sind einige Konverter für verschiedene FE-Programme und eine Vielzahl von Reduktionsmethoden implementiert. Die Software liegt sowohl als eine *Matlab*-basierte Version (*MatMorembs*) als auch in *C++* (*Morembs++*) vor. Für die Modellreduktion kleiner bis mittelgroßer Modelle hat sich *MatMorembs* bewährt. Große Modelle hingegen werden durch die Verwendung von leistungsfähigeren Numerikbibliotheken und eine geeignetere Speicherauslastung mit *Morembs++* reduziert. Zunächst muss ein Modell in einem Standard-FE-Programm wie *Ansys*, *Abaqus* oder *Permas* aufgebaut und dynamisch analysiert werden. Anschließend kann der Reduktion-Prozess beginnen, der sich in drei modular aufgebaute Abschnitte unterteilen lässt.

Im ersten Schritt werden Daten aus dem FE-Programm ausgelesen und die Systemmatrizen des vollen FE-Modells aufgebaut. Die anschließend durchgeführte Modellreduktion liefert die elastischen Ansatzfunktionen, die auf Basis unterschiedlicher Verfahren generiert werden können. Im letzten Schritt werden die Daten des reduzierten Modells für die Verwendung in einem Standard-MKS-

Programm mittels sogenannter *SID*-Struktur, vgl. [SchwertassekWallrapp99] aufgearbeitet. Beim Verwenden der Kombination *MatMorembs* - *Neweul-M²* können die Daten auch direkt in *Matlab*-Speicherstrukturen übergeben werden. Die Prozesskette ist in Abbildung 2.4 dargestellt, welche sich auf der Homepage des ITM's befindet (www.itm.uni-stuttgart.de).

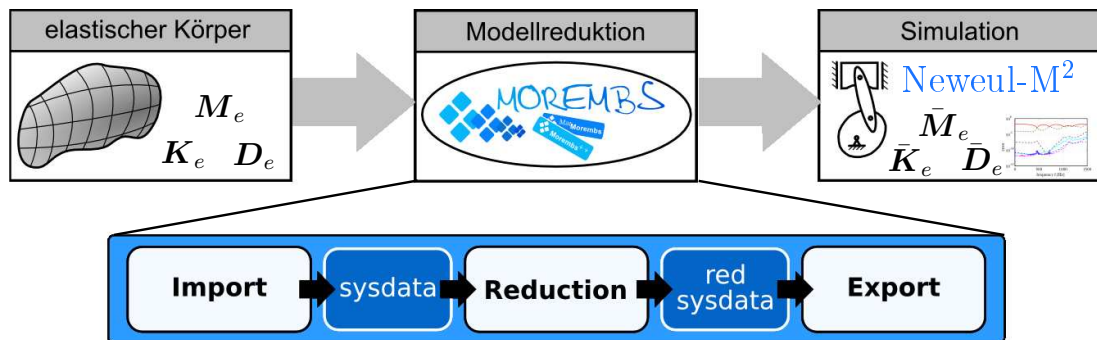


Abbildung 2.4: Prozesskette zur Modellreduktion mittels Morembs

2.1.5 Neweul-M²

Neweul-M² ist ein Software Paket für die dynamische Analyse mechanischer Systeme mit Hilfe der Methode der elastischen Mehrkörpersysteme. Es umfasst das Aufstellen der symbolischen Bewegungsgleichungen basierend auf dem Newton-Euler-Formalismus, sowie die Simulation und Analyse des dynamischen Verhaltens. Das Tool wurde für die *Matlab*-Umgebung programmiert und verwendet die *Symbolic Math Toolbox* für die symbolischen Berechnungen. Dies bringt zwei entscheidende Vorteile: Zum Einen kann mit *Maple* oder *MuPad* ein kommerzieller symbolischer Manipulator verwendet werden und andererseits stehen dem Nutzer die effizienten numerischen Methoden von *Matlab* zur Verfügung. Für das Animieren und Darstellen von Körpern, Koppelementen und deren Bewegungen, stehen neben den Standard-*Matlab*-Befehlen auch zusätzliche *Neweul-M²* - Funktionen zur Verfügung.

2.2 Optik

In dieser Arbeit wird nicht nur die Dynamik des optischen Systems, sondern auch der Einfluss einer axialen Linsenbewegung auf die optischen Größen untersucht. Diese Kombination lässt sich als dynamisch-optisch-gekoppeltes System bezeichnen. Im Folgenden werden die wichtigsten optischen Begriffe für die

se Arbeit vorgestellt und veranschaulicht. Eine umfassende Beschreibung wird in [PedrottiEtAl02, EberhardWengertTouihri10] ausgeführt.

2.2.1 Raytracing

Die vektorielle Verfolgung von Lichtstrahlen stellt das Standardverfahren zur Simulation optischer Systeme dar und wird in der Optik als *Raytracing* bezeichnet. Basierend auf den Gesetzen der geometrischen Optik lassen sich die Lichtausbreitung in Form gerade verlaufender Strahlen berechnen und simulieren. Ausgehend von einer Lichtquelle durchlaufen Strahlen sequentiell das optische System und werden durch Ausbreitung, Brechung und Reflexion an Linsen beeinflusst. Im Bildpunkt O soll das Licht auf den Schirm treffen. Aus Symmetriegründen werden in dieser Arbeit nur die Strahlen in der y - z -Ebene betrachtet, siehe Abbildung 2.5.

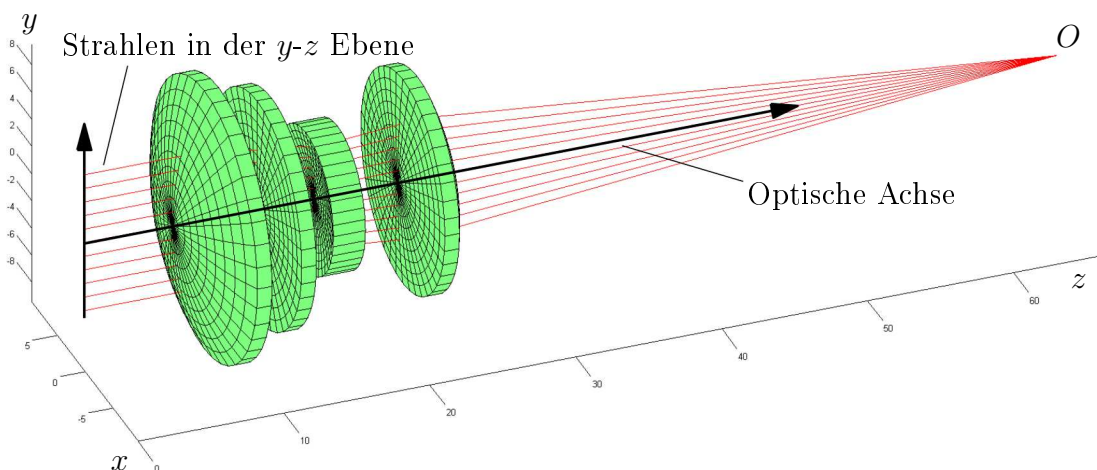


Abbildung 2.5: Darstellung der berechneten Strahlen beim *Raytracing* mit *OM-Sim*

2.2.2 Aberrationen

In der Optik werden Abbildungsfehler auch *Aberrationen* genannt. Im Folgenden wird die *Wellenfrontaberration* und dessen Beschreibung mittels *Zernike-Polynomen* kurz erläutert.

Die *Wellenfrontaberration* charakterisiert optische Weglängenabweichungen aller Strahlen, bezogen auf die optische Weglänge des Hauptstrahls. In einem Medium

mit dem Refraktionsindex n ist die optische Weglänge eines Strahles n -mal länger als die geometrische Weglänge. Zeichnet man ursprungsgleiche Strahlen, die ein optisches System durchlaufen, so auf, dass jeder Strahl die gleiche optische Weglänge durchläuft, dann bildet sich im Idealfall eine sphärische Wellenfrontfläche W_{ideal} , dessen Krümmungskreismittelpunkt im Bildpunkt O liegt. Ist diese Fläche in irgendeiner Form deformiert, führt dies zu unscharfen und fehlerhaften Abbildungen, siehe Abbildung 2.6.

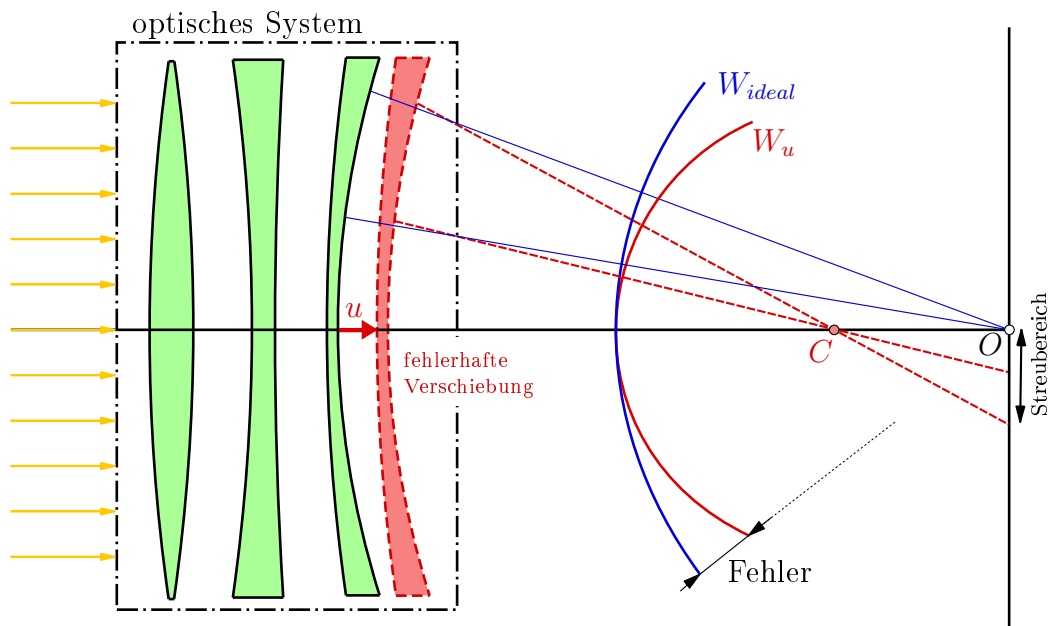


Abbildung 2.6: Veranschaulichung eines Defokus als Wellenfrontfehler (4. Zernike-Polynom), resultierend aus einer Linsenverschiebung

Die Form einer deformierten Wellenfrontfläche kann durch die Betrachtung der optischen Weglängen modelliert und durch geeignete Funktionen approximiert werden. Für gewöhnlich eignet sich hierfür die Verwendung von *Zernike-Polynomen*. Dabei werden komplexe Wellenaberrationen in eine Gruppe von polynomischen Basisfunktionen zerlegt. Jeder dieser Zernike-Polynome repräsentiert eine spezielle Wellenfrontaberration, sodass durch deren Superposition allgemeine optische Abberationen beschrieben werden können. Eine ausführliche Beschreibung findet sich in [WengertEberhard11].

Ein Zernike-Polynom besteht aus einem radialen Term $R_n^m(r)$ und einem von dem Umlaufwinkel ϕ abhängigen Term. Die Polynome werden in der Regel mit dem

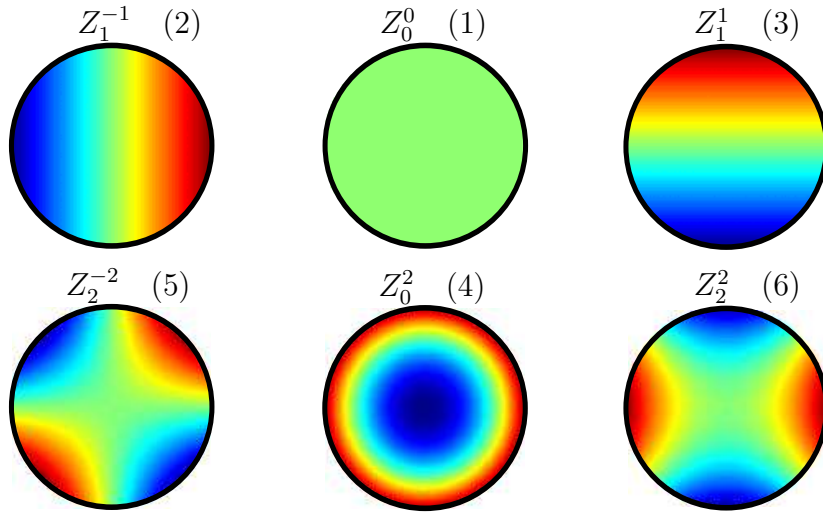


Abbildung 2.7: Darstellung der ersten sechs Zernike-Polynome

Doppelindizierungsschema

$$\begin{aligned}
 Z_n^m(r, \phi) &= R_n^m(r) \cos(m\phi) \\
 Z_n^{-m}(r, \phi) &= R_n^m(r) \sin(m\phi) \\
 Z_n^0(r, \phi) &= R_n^0(r)
 \end{aligned} \tag{2.16}$$

repräsentiert, mit

$$R_n^m(r) = \sum_{k=0}^{(n-m)/2} \frac{(-1)^k (n-k)!}{k! ((n+m)/2 - k)! ((n-m)/2 - k)!} r^{n-2k}. \tag{2.17}$$

Die Indizes m und n stehen für die radiale Ordnungen. Sie sind positive ganze Zahlen, die beide gerade oder ungerade sind und es gilt $n \geq m$. In Abbildung 2.7 sind die ersten sechs Zernike-Polynome graphisch dargestellt.

2.2.3 OM-Sim

Die Berechnung der Bildfehler erfolgt im Rahmen dieser Arbeit mit der am ITM entwickelten Optik-Software *OM-Sim* in der *Matlab*-Umgebung. Die Linsen eines optischen Systems werden dabei mit Hilfe von einfachen Parametern wie Krümmungsradius, Dicke, Medium und Durchmesser definiert. Somit ist die Reihenfolge, in der das Licht die Oberflächen passiert, fest vorgeschrieben und die Berechnung der Lichtstrahlenverläufe erfolgt entsprechend den Gesetzen der geometrischen Optik. Während der Rechnung wird das optische System entsprechend Abbildung 2.5 in einer 3D-Grafik animiert. Es wird der Abbildungsfehler

mittels Zernike-Polynomen dargestellt und die berechnete Bildschärfe kann anhand eines Beispiel-Bildes verdeutlicht werden.

Kapitel 3

Prozessketten und Implementierungen

Das folgende Kapitel beschreibt zunächst die Prozesskette einer klassischen EMKS-Simulation mit *Ansys*, *MatMorembs* und *Neweul-M²*. Anschließend werden die Implementierungen zur dynamischen Spannungsanalyse eines modellreduzierten Körpers in *MatMorembs* und *Neweul-M²* vorgestellt. Da sämtliche Teilfunktionen hierzu bereits entwickelt und implementiert wurden, ist die Hauptaufgabe ein geschicktes Zusammenfassen und Erweitern der Programmteile zu überschaubaren und anwendungsfreundlichen Funktionen.

Das Ziel dieser Erweiterung ist die Bereitstellung der Spannungsmoden für die dynamische Spannungsberechnung an gezielt definierten Knotenpunkten, genannt *Retained Nodes* (RN), sowie die Animation der Spannungsverteilung auf der Oberfläche eines reduzierten Körpermodells in *Neweul-M²*.

3.1 Prozesskette einer klassischen EMKS-Simulation

Im Rahmen eines klassischen Reduktionsprozesses mit *MatMorembs* bleiben die Spannungen im elastischen Körper unbeachtet. Unabhängig vom Reduktionsverfahren werden lediglich die Verschiebungen an *Retained Nodes* mittels elastischen Koordinaten und zugehörigen Ansatzfunktionen berücksichtigt. Als Ergebnis der Reduktionprozesskette erhält man die Struktur (*struct*) eines reduzierten elastischen Körpers, auch genannt *ElasticBody*; in *SID*-Form oder als *Matlab-Struct*-Variable. Der *ElasticBody* kann nun entweder in ein kommerzielles MKS-Programm wie z. B. *Simpack* oder in *Neweul-M²* eingebunden werden.

In Abbildung 3.1 ist eine klassische Prozesskette illustriert. In der rechten Spalte sind die wichtigsten *Ansys*, *MatMorembs* und *Neweul-M²* Befehle aufgeführt und links ist die Gruppierung von selbst definierten Funktionen dargestellt. Folgende Schritte sind dabei zu beachten:

- Das Modell des elastischen Körpers wird in *Ansys*-APDL-Scriptsprache definiert und enthält alle geometrischen und materiellen Information.
- Für die Berechnung der Dateien *struct.full* und *eigen.rst*, die für die Modellreduktion benötigt werden, wird außerdem eine *inp*-Datei vorbereitet. Diese Art der Modelldefinition ist die Voraussetzung für eine spätere Berechnung der Spannungsmoden.
- Die Modellreduktion in *MatMorembs* beginnt mit der Evaluierung der bereits erwähnten Dateien *struct.full* und *eigen.rst*, mittels *Ansys* Batch-Befehl.
- Im nächsten Schritt werden die errechneten Matrizen aus den FE-Ergebnis-Dateien ausgelesen und in *Matlab*-Strukturen gespeichert.
- Die eigentliche Reduktion findet auf der Basis eines ausgewählten Reduktionsverfahren statt. Im aufgeführten Beispiel wird eine modale Reduktion angewandt.
- Im letzten *MatMorembs* Schritt wird der oben beschriebene *ElasticBody* erzeugt. Die zusätzliche Funktion *elbodyAddMesh* fügt der Struktur des elastischen Körpers optional die Ansatzfunktionen aller Oberflächenknoten ($\bar{\Phi}_{Mesh}$) hinzu, sodass bei einer Simulation in *Neweul-M²* auch die Dynamik der Oberfläche animiert werden kann. Wenn der Körper in einer *SID*-Datei benötigt wird, kann dies mit aufgeführten Funktionen erzeugt werden; für *Neweul-M²* ist die *mat*-Datei ausreichend.
- In *Neweul-M²* wird zunächst das EMKS definiert. Der elastische Körper wird idealerweise direkt als *Matlab-Struct* eingebunden.
- Das Aufstellen, Berechnen und ggf. Linearisieren der Bewegungsgleichungen führt auf ein System, das in symbolischer Form analytisch hergeleitet wird.
- Im letzten Schritt wird die Zeitintegration durchgeführt, welche die Dynamik des Systems in Abhängigkeit der System-FHG berechnet. Zur Visualisierung wird das System in einer 3D-Abbildung dargestellt und über die Zeit animiert. Falls der elastische Körper die Ansatzfunktionen der Oberflächenknoten enthält, stellt sich dessen Deformation anschaulich dar.

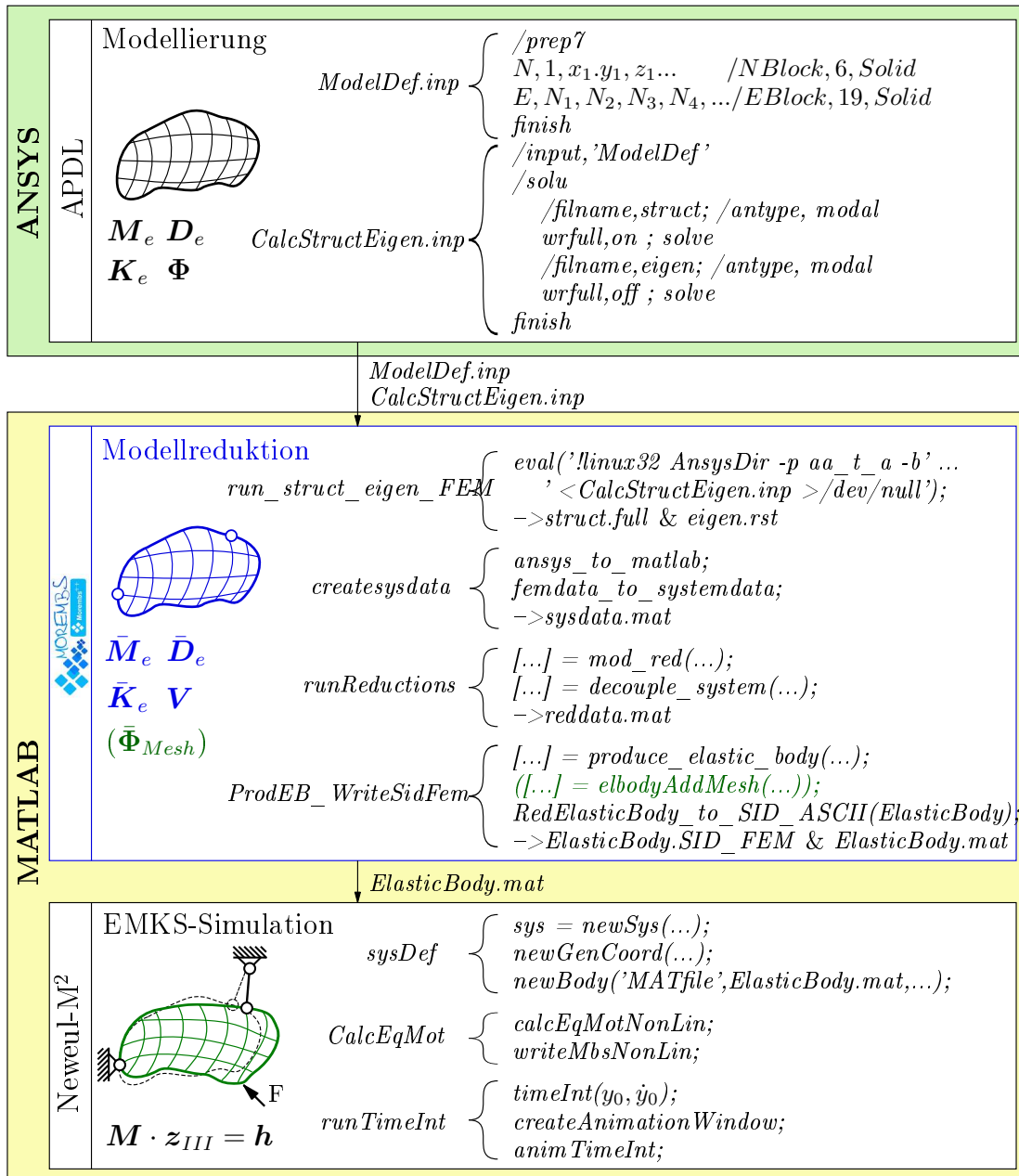


Abbildung 3.1: Klassischer Modell-Reduktions-Prozess mit anschließender EMKS Simulation in *Neueul-M²*

3.2 Prozesskette mit Spannungsanalyse

Die dynamische Spannungsanalyse eines modellreduzierten Körpers kann nach Kapitel 2.1.3 unter der Verwendung von Spannungsmoden erfolgen. Hierbei werden nach einer gewöhnlichen Modellreduktion die Knotenverschiebungen entsprechend den reduzierten Ansatzfunktionen schrittweise als Zwangsbedingungen (*LOADCASES*) auf die Knoten-FHG des Gesamtmodells aufgeprägt. Mittels der FEM lassen sich dadurch die reduzierten Spannungsmoden schrittweise rechnerisch ermitteln. In einem letzten Schritt müssen noch alle FE-Teilergebnisse ausgelesen und zu der reduzierten Spannungsmodenmatrix $\bar{\Phi}_\sigma$ zusammengefasst werden. Aus dieser Matrix können nun die Spannungsmoden für die *Retained Nodes* in *SID*-Form und für die Spannungsanimation der Oberflächen extrahiert werden. Im Folgenden wird ausgehend von einem *Ansys*-Modell die Implementierung dieser automatisierten Prozesskette beschrieben.

3.2.1 Stand vor der Implementierung

Neben der Prozesskette einer klassischen EMKS-Simulation nach Kapitel 3.1 gibt es auch Problemstellungen, die die Spannungen im reduzierten Bauteil berücksichtigen müssen. In der Vergangenheit wurden Spannungsuntersuchungen vor allem im Zusammenhang mit Struktur-Optimierungen benötigt, deshalb sind bereits Schnittstellen und Funktionen zu verschiedenen FE-Programmen in *MatMorembs* implementiert worden. Für die Berechnung der Spannungsmoden in *Ansys* sind die folgenden Funktionen vorhanden:

- *ansysstremo_preparecalcstressmodes(...)*
- *ansysstremo_run_FEM(...)*
- *ansysstremo_readstressmodes(...)*
- *SimpStremoWriteSidSigma(...)*
- *SimpStremoWriteNodesTable(...)* .

In Abbildung 3.2 ist eine mögliche Prozesskette für die Spannungsanalyse mittels den Programmen *Ansys*, *MatMorembs* und *NeweulM²* oder *Simpack* dargestellt. Die Unterschiede zu Kapitel 3.1 sind im Folgenden aufgeführt:

- Während des Modellreduktionsprozesses werden nach dem Erstellen des *ElasticBodys* die Spannungsmoden mittels *Ansys* berechnet.

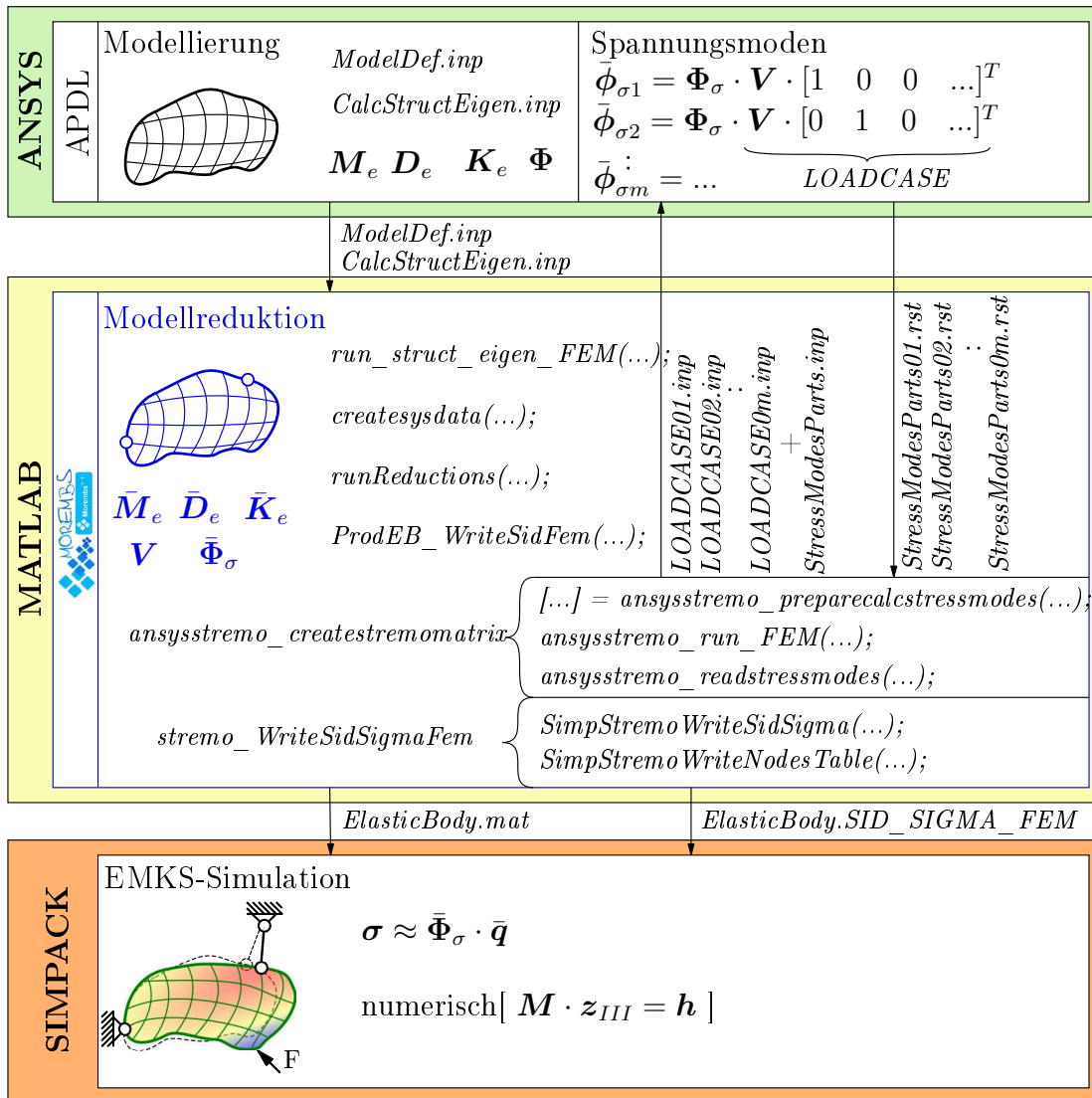


Abbildung 3.2: Modell-Reduktions-Prozess mit Berechnung der Spannungsmoden in *SID-Form*. Anschließend EMKS-Simulation in *Simpact* mit Spannungsanalyse, vgl. Abbildung 3.1 und Kapitel 2.1.3

- Im ersten Schritt werden sogenannte *LOADCASE**.inp*-Dateien vorbereitet. Diese enthalten Zwangsbedingungen in Form von Knotenverschiebungen, die aus den Spalten der Reduktionsmatrix \mathbf{V} resultieren, siehe Abbildung 3.2, rechts oben.
 - In einer weiteren *inp*-Datei werden die *Ansys*-Befehle zum Anwenden der Zwangsbedingungen und Berechnen der einzelnen Spannungsmoden definiert. Diese Skripte werden per *Batch*-Befehl von *Matlab* aus gestartet (analog zu *run_struct_eigen_FEM*).
 - Anschließend werden die FE-Ergebnisse ausgelesen und zu einer reduzierten Spannungsmoden-Matrix $\bar{\Phi}_\sigma$ zusammengefasst.
- Durch das Erzeugen einer *SID*-Struktur ist eine Schnittstelle zur Verwendung der Spannungsmoden in verschiedenen Programmen (z.B. *Simpack*) realisiert.

In einer EMKS-Simulation mit *Neweul-M²* können die Spannungsmoden nicht ohne weiteres für die Spannungsanalyse genutzt werden, da lediglich die *ElasticBody*-Struktur importiert wird. Deshalb sind neue Implementierungen nötig, die im Folgenden dargestellt werden.

3.2.2 Implementierung der dynamischen Spannungsanalyse in *Neweul-M²*

Für eine Spannungsanalyse der *Retained Nodes* und eine Möglichkeit zur Spannungsanimation der Oberfläche des *ElasticBodys* in *Neweul-M²* sind zunächst neue Implementierungen notwendig. Folgende Funktionen wurden modifiziert bzw. neu erstellt:

- *MatMorembs*
 - neu: *ansysstremo_createstremomatrix(...)*
 - neu: *elbodyAddStremo(...)*
 - modifiziert: *elbodyAddMesh(...)*
- *Neweul-M²*
 - modifiziert: *drawMesh(...)*
 - modifiziert: *updateSurfaces(...)*.

Die Prozesskette ist in Abbildung 3.3 dargestellt:

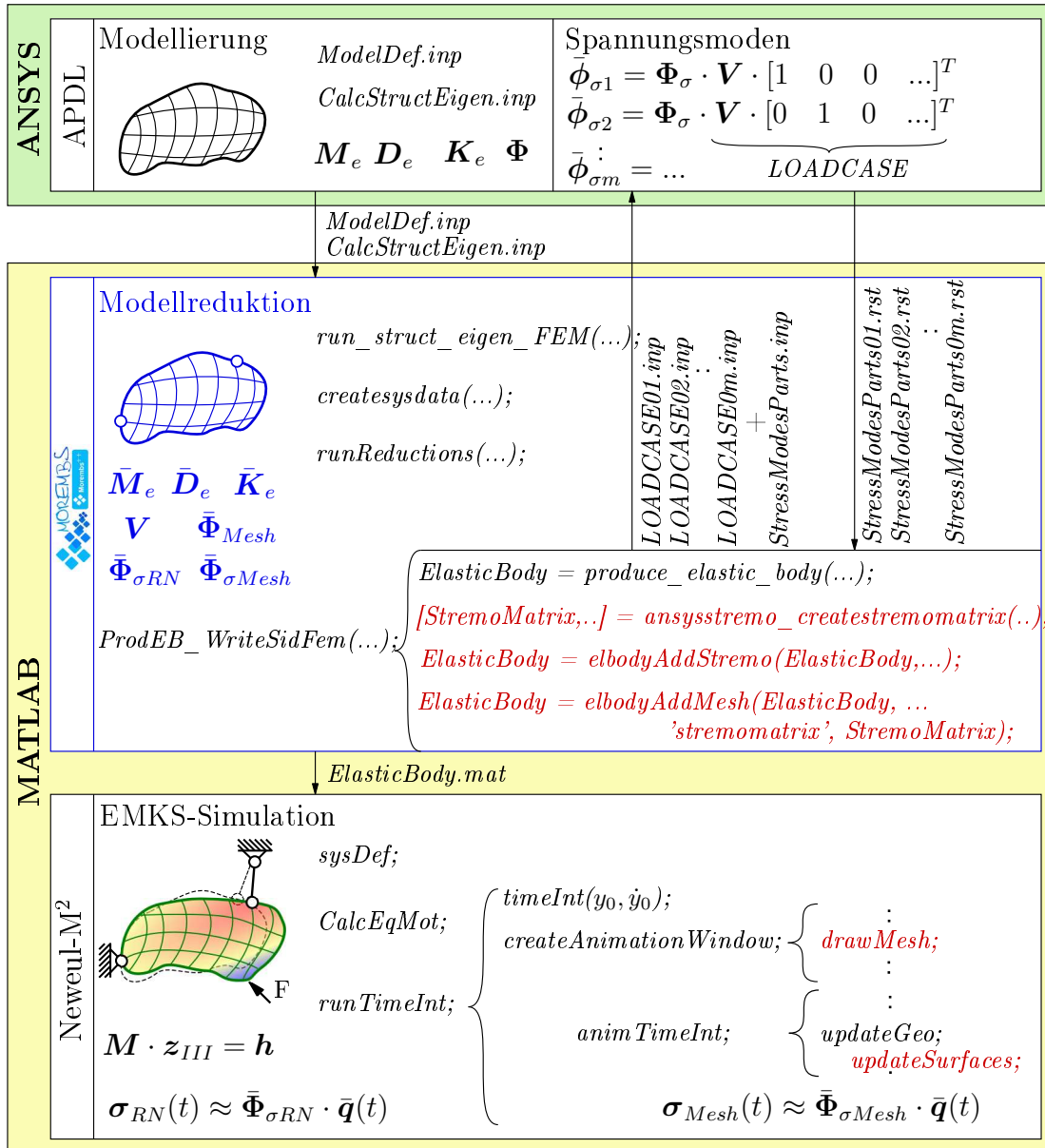
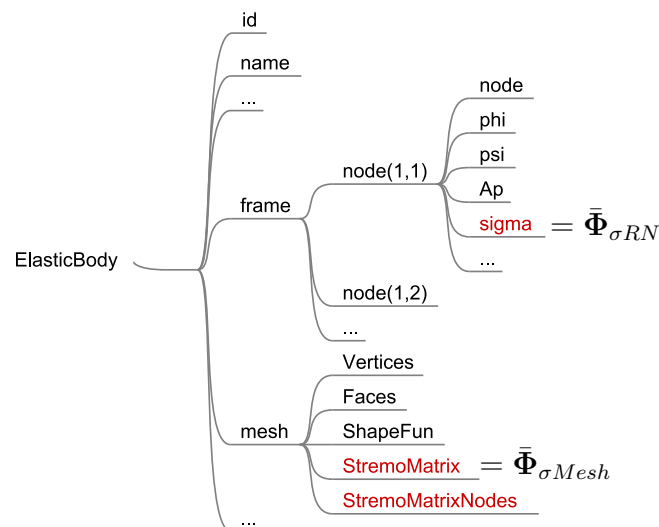


Abbildung 3.3: Modell-Reduktions-Prozess mit Spannungsmodenberechnung. Anschließend EMKS Simulation in *Neueul-M²* mit Animation der dynamischen Oberflächenspannungen. Die neu implementierten Funktionen sind in rot dargestellt.

- Unmittelbar nach dem Erzeugen des elastischen Körpers mit der Funktion $ElasticBody=produce_elastic_body(...)$, wird die reduzierte Spannungsmoden-Matrix $\bar{\Phi}_\sigma$, auch *StremoMatrix* genannt, berechnet. Dies erfolgt mit der neuen Funktion $[StremoMatrix,...]=ansysstremo_createstremomatrix(...)$, womit die *StremoMatrix* zurückgegeben wird.
- Um die Spannungsmoden in weiterführenden Berechnungen nutzen zu können, müssen diese in den *ElasticBody* eingebunden werden. In [SchwertassekWallrapp99] wird vorgeschlagen, lediglich die Spannungsmoden der *Retained Nodes* $\bar{\Phi}_{\sigma RN}$ in SID-Form zu übernehmen. Dies wird durch die neu implementierte Funktion $ElasticBody=elbodyAddStremo(ElasticBody,...)$ realisiert.
- Als Voraussetzung für die Animation der Oberflächenspannungen müssen die entsprechenden Spannungsmoden $\bar{\Phi}_{\sigma Mesh}$ auch im *ElasticBody* gespeichert werden. Deshalb wird durch den Aufruf der modifizierten Funktion $ElasticBody=elbodyAddMesh(ElasticBody, 'stremomatrix', StremoMatrix,...)$ die Speicherstruktur entsprechend Abbildung 3.4 erweitert.

Abbildung 3.4: Darstellung der *ElasticBody*-Struktur von *MatMoremb*s

- Für die EMKS-Simulation in *Neweul-M²* wird nun die Struktur des elastischen Körpers entsprechend Abbildung 3.5 in das System eingebunden. Das Aufstellen der Bewegungsgleichung und die numerische Zeitintegration erfolgt unverändert.

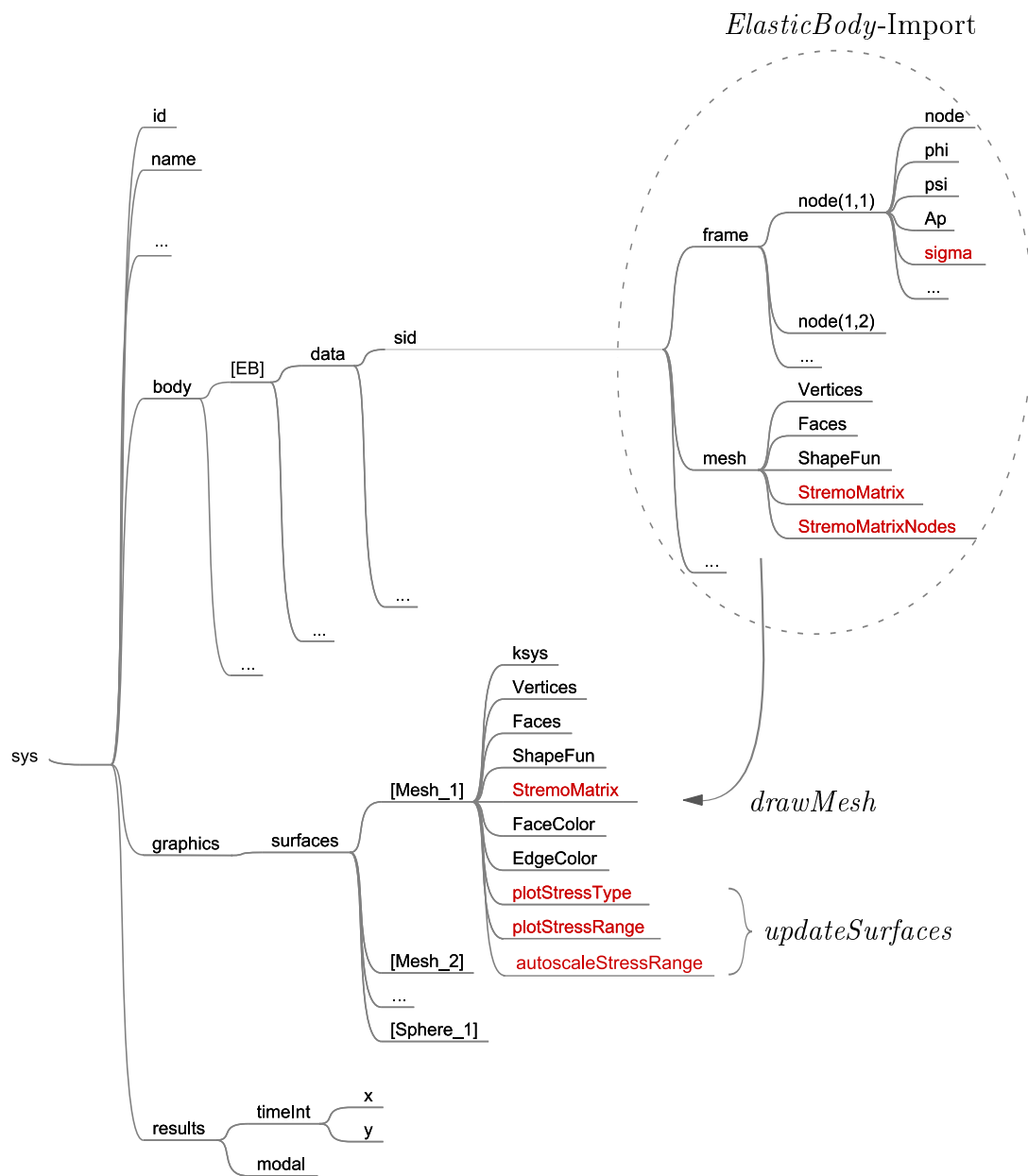


Abbildung 3.5: Darstellung der *sys*-Struktur von Neweul-M²

- Beim Erstellen des Animationsfensters werden zunächst die grafischen Informationen des elastischen Körpers mit der Funktion *drawMesh* separat abgelegt. Um hierbei die Spannungsmoden für die Oberflächenknoten zu übernehmen, ist eine Modifizierung der Funktion notwendig. Alle grafischen Objekte werden erzeugt und im Animationsfenster aufgezeigt. Die Generierung des Netzflächen erfolgt hierbei mit dem internen *Matlab*-Befehl *patch*.
- Während der Animation der Simulationsergebnisse wird in jedem Animationsschritt die Funktion *updateGeo* aufgerufen. Hierbei aktualisieren sich alle grafischen Elemente und dessen Position und Größe berechnen sich neu. Mit der Unterfunktion *updateSurface* wird die Oberflächenform eines elastischen Körpers neu berechnet und dargestellt. Für die Animation der Oberflächen-Spannung ist in dieser Funktion nun auch das Neuberechnen der Spannungen zu dem jeweiligen Animationszeitschritt t_{anim} implementiert mit $\sigma_{Mesh}(t_{anim}) \approx \bar{\Phi}_{\sigma_{Mesh}} \cdot \bar{q}(t_{anim})$.

Mit Hilfe des *patch*-Befehls können den Flächen-Eckpunkten jeweils Farben zugeordnet werden. Somit lässt sich die Spannung an den Netzknoten farblich mit *patch(..., 'FaceVertexCData', $\sigma_{Mesh}(t_{anim})$)* darstellen und durch eine lineare Interpolation global veranschaulichen. Der Struktur werden automatisch neue Parameter hinzugefügt, die vom Benutzer editiert werden können, vgl. Abbildung 3.5. Neben den sechs Spannungskomponenten σ_x , σ_y , σ_z , σ_{xy} , σ_{yz} und σ_{xz} lässt sich auch die Spannung σ_{vMises} animieren.

- Der transiente Spannungsverlauf an den *Retained Nodes* kann nun aus dem Produkt der lokalen Spannungsmoden-Matrizen und der Lösung der elastischen Koordinaten mit $\sigma_{RN}(t) \approx \bar{\Phi}_{\sigma_{RN}} \cdot \bar{q}(t)$ berechnet werden.

3.2.3 Beispiel für die dynamischen Spannungsanalyse in Neweul-M²

Eine beispielhafte Spannungsanalyse in *Neweul-M²* ist zur Verdeutlichung in Algorithmus 3.1 aufgeführt. Nach der Zeitintegration wird eine Animation gestartet, welche die Oberflächenspannungen farblich visualisiert. Anschließend wird das Plotten des transienten Spannungsverlaufs an einem *Retained Node* demonstriert. Im letzten Teil wird die Möglichkeit aufgezeigt, das System an einem bestimmten Lösungsschritt darzustellen, vgl. Abbildung 3.6.

Listing 3.1: Spannungsanalyse mit Neweul-M²

```

1 %% Example for a stress analysis with neweulm2
2 addpathNeweulm2;% Initialize neweulm2-Functions
3

```

```

4 % Modelling-Process:
5 sysDef;           % Define System with subroutine
6 CalcEqMot;       % Calculate the system equations with subroutine
7
8 % ----- START: runTimeInt - subroutine -----
9 %...           % Set initial conditions and define Integration-settings
10 timeInt (y_0,Dy_0);
11
12 % Specify a adjusted Animation-Window
13 createAnimationWindow;
14 % Set a black EdgeColor to the elastic body mesh
15 sys.graphics.surfaces.Mesh_1.EdgeColor=[0 0 0];
16 % Close & recreate AnimWindow to update Properties
17 close(gcf);
18 createAnimationWindow;
19
20 % Set plotStressType
21 sys.graphics.surfaces.Mesh_1.plotStressType.sig_x      = 0
22 sys.graphics.surfaces.Mesh_1.plotStressType.sig_y      = 0
23 sys.graphics.surfaces.Mesh_1.plotStressType.sig_z      = 1
24 sys.graphics.surfaces.Mesh_1.plotStressType.sig_xy     = 0
25 sys.graphics.surfaces.Mesh_1.plotStressType.sig_yz     = 0
26 sys.graphics.surfaces.Mesh_1.plotStressType.sig_xz     = 0
27 sys.graphics.surfaces.Mesh_1.plotStressType.sig_vMises = 0;
28
29 % Set user defined scale of plotStressRange
30 sys.graphics.surfaces.Mesh_1.autoscaleStressRange = 0;
31 sys.graphics.surfaces.Mesh_1.plotStressRange = 1.7e10*[-1, 1];
32
33 % start Animation
34 animTimeInt;
35
36 % Plot transient stress of a Retained Node (RN) }
37 RNs = sys.body.EB.data.sid.frame.node;
38 StremoRN = RNs(idxRN).sigma; % Obtain local stressmodes}
39 StressRN_all = StremoRN * sys.results.timeInt.y(1:dofEB,:);
40 idxStress = 3; % 1=Sig\_x, 2=Sig\_y...
41 Sig_idxStress = StressRN_all(idxStress,:); %Plot Sig_idxStress}
42
43 figure; % Create figure
44 time = sys.results.timeInt.x;
45 plot (time,Sig_idxStress); % Plot specified stress
46
47 % Draw animation at defined SolutionStep
48 SolutionStep = 22;
49 updateGeo (0,sys.results.timeInt.y(:,SolutionStep));

```

Nachdem die Prozessketten und Neuimplementierungen erklärt wurden, wird im folgenden Kapitel die mechanische Simulation des Linsensystems vorgestellt.

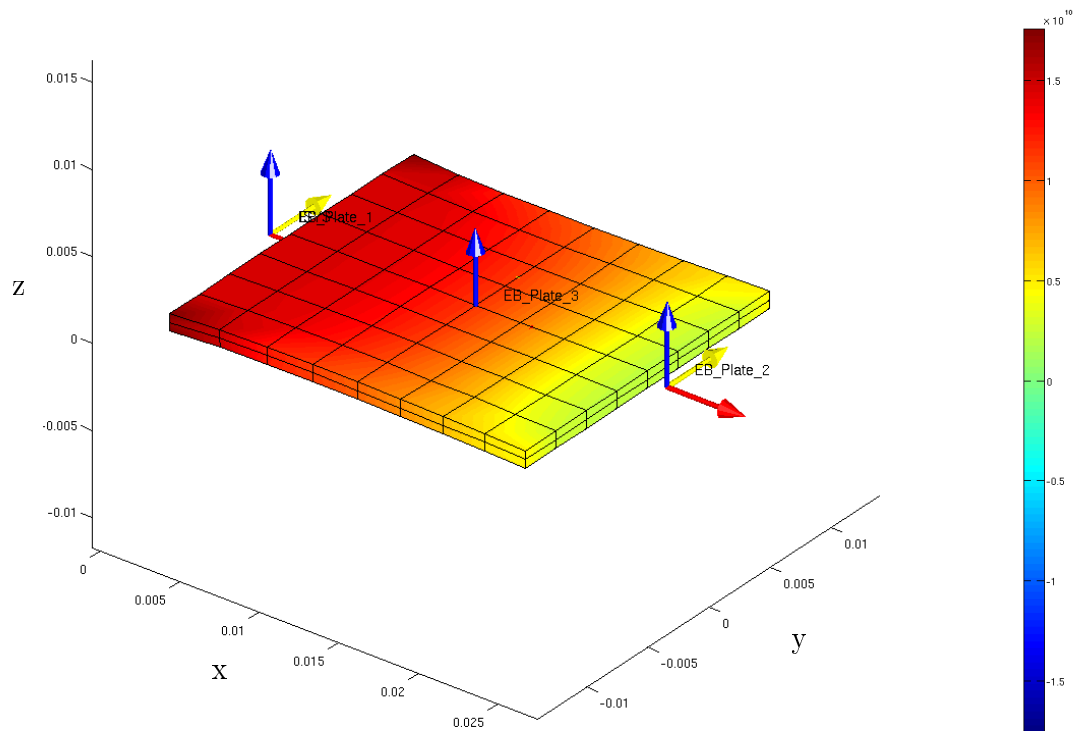


Abbildung 3.6: Darstellung der Spannung in x-Richtung eines zugbelasteten elastischen Körpers, animiert in *Neweul-M²*

Kapitel 4

Mechanische Simulation

In diesem Kapitel wird die Modellierung und Simulation des Linsensystem als EMKS aufgezeigt, vgl. Abbildung 4.1. Zuvor wird die Vorbereitung einer Platte als reduzierter elastischer Körper beschrieben.

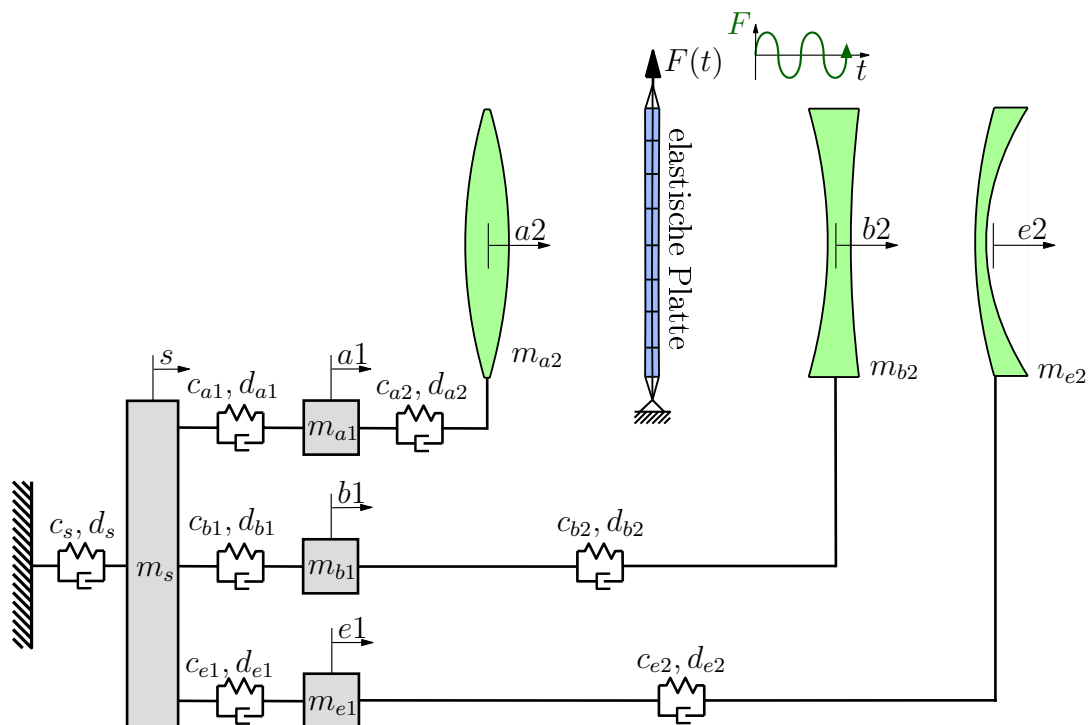


Abbildung 4.1: Darstellung des elastischen Linsensystems, modelliert als EMKS

4.1 Modelldefinition des elastischen Linsensystems

Das mechanische System besteht zum Einen aus gekoppelten Starrkörperlinsen, die sich in Richtung der optische Achse verschieben lassen und zum Anderen aus einer entkoppelten elastischen Platte. Der elastische Körper kann mit einer Kraft gemäß Abbildung 4.1 beansprucht werden. Eine 3D-Darstellung des Linsensystems ist in Abbildung 2.5 veranschaulicht. Das Ziel ist die dynamische Spannungsanalyse der elastischen Platte und die Untersuchung der Linsen-Starrkörperbewegungen. Zuvor sind einige Schritte notwendig, die aufgezeigt werden.

4.2 Erzeugung der elastischen Platte

Um die Platte als elastischen Körper in *Neweul-M²* implementieren zu können wird sie zuerst modelliert. Anschließend wird das Modell reduziert und als Einzelkörper in *Neweul-M²* analysiert.

4.2.1 Platten-Modellierung in Ansys

Die elastische Platte wird im FE-Programm *Ansys* modelliert. Das Ziel ist die Vorbereitung der *Ansys inp*-Datei zum Erzeugen des vernetzten Modells und die Bereitstellung einer *inp*-Datei zum Berechnen der Dateien *struct.full* und *eigen.rst*. Diese sind notwendig für die anschließende Modellreduktion.

Zunächst wird ein quaderförmiges Plattenvolumen erstellt und mit gleich großen Elementen vernetzt. Da die Platte in *Neweul-M²* als Tangentensystem eingebunden wird, ist bei der Erstellung der Kontur auf den Ursprung des Koordinatensystems zu achten. Er wird so verschoben, dass er genau an einer Schnittstelle liegt. Die für die Schnittstellen relevanten Knotenpunkte werden als *Master-Nodes M1 & M2* und als *Slave-Nodes S1, S2, ...* bezeichnet, siehe Abbildung 4.2.

Die Modellierung der Schnittstellen erfolgt mit Hilfe einer geschriebenen Unterfunktion *SUB_prod_interface.inp*, die außerdem einen selbst definierten *Ansys*-Befehl *FUNC_BEA.mac* aufruft. Dieser Befehl erzeugt Balken zwischen den *Master*- und *Slave*-Knotenpunkten und bildet die Grundlage einer elastischen Schnittstelle. Dies hat den Vorteil, dass der Körper zwar an einem Punkt angebunden wird, die Kräfte aber an einer großflächigen Kontur eingeleitet werden. Somit ist die Modell-Datei *Plate_model.inp* vollständig definiert.

Die für die Modellreduktion notwendigen Dateien *struct.full* und *eigen.rst* werden

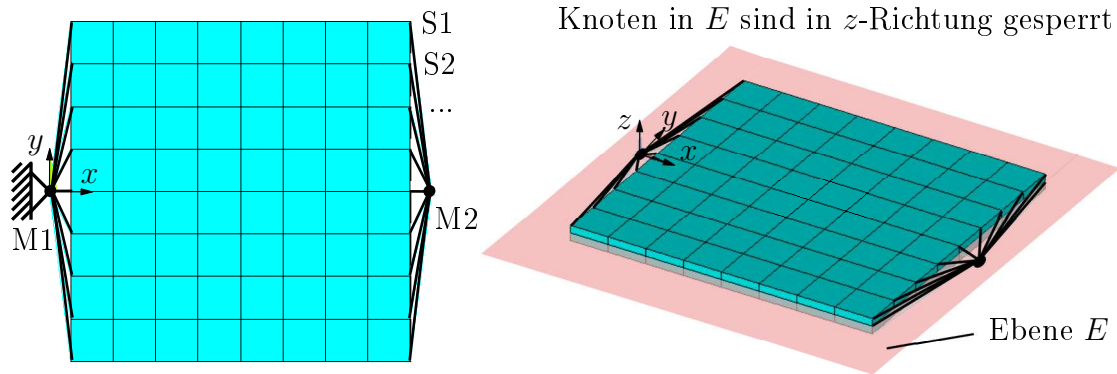


Abbildung 4.2: Darstellung des Ansys-Modells

mit Hilfe der Datei *MAIN_Plate_Calc_StructEigen.inp* nach einem vorgegebenen Muster berechnet, vgl. Abbildung 3.1. Um nur die Platten-Eigenmoden in der Ebene E zu berücksichtigen und eine Kraftanregung am Knoten $M2$ vorzubereiten, sind in dieser Datei außerdem die Randbedingungen wie folgt festgelegt, vgl. Abbildung 4.2:

- Die Verschiebungen in z -Richtung aller in der Ebene E liegenden Knotenpunkte sind gesperrt.
- Der Punkt $M1$ wird in allen Raumrichtungen fixiert.

Durch das Ausführen von *MAIN_Plate_Calc_StructEigen.inp* wird das komplette Modell erzeugt und die Modellinformationen in *struct.full* gespeichert. Die Lösung aus der Modalanalyse ist in *eigen.rst* enthalten.

4.2.2 Modellreduktion und Spannungsmodenberechnung

Die Modellreduktion der Platte erfolgt mit *MatMorembs*. In Abbildung 3.3 ist die Programmstruktur bereits schematisch aufgezeigt. Der Körper wird modal mit den ersten vier Eigenmoden reduziert. Für die Reduktionsmatrix folgt deshalb $\mathbf{V} = [\phi_1 \ \phi_2 \ \phi_3 \ \phi_4]$. Da der Körper nicht frei modelliert wurde, sind die eingeschränkten Starrkörperbewegungen im Reduktionsprozess zu berücksichtigen. Die Dämpfung im Inneren der Platte wird vernachlässigt.

Mit Hilfe von *Ansys* werden die reduzierten Spannungsmoden knotenweise berechnet und zu der reduzierten Spannungsmodenmatrix $\bar{\Phi}_\sigma = [\bar{\phi}_{\sigma 1} \ \bar{\phi}_{\sigma 2} \ \bar{\phi}_{\sigma 3} \ \bar{\phi}_{\sigma 4}]$ zusammengefasst. Die Neuimplementierung ermöglicht das Hinzufügen der lokalen Spannungsmoden für die *Retained Nodes* $\bar{\Phi}_{\sigma RN}$ zum elastischen Körper. Außerdem werden für die Animation der Oberflächenspannung

die lokalen Spannungsmoden der Oberflächenknoten $\bar{\Phi}_{\sigma Mesh}$ der Struktur hinzugefügt. Nun ist die *ElasticBody*-Struktur vollständig erzeugt.

4.2.3 Validierung der dynamischen Spannungsanalyse

Bevor das reduzierte Plattenmodell in das Linsensystem eingebaut wird, erfolgt eine Validierung der Spannungsergebnisse anhand einer großen harmonischen Anregung mit der ersten Platten-Eigenfrequenz. Hierfür wird der reduzierte Körper in *Neweul-M²* importiert, angeregt und transient analysiert. Das unreduzierte Modell wird in *Ansys* identisch angeregt und transient berechnet. In Abbildung 4.3 ist die Animation der Oberflächenspannung beider Programme illustriert. Die

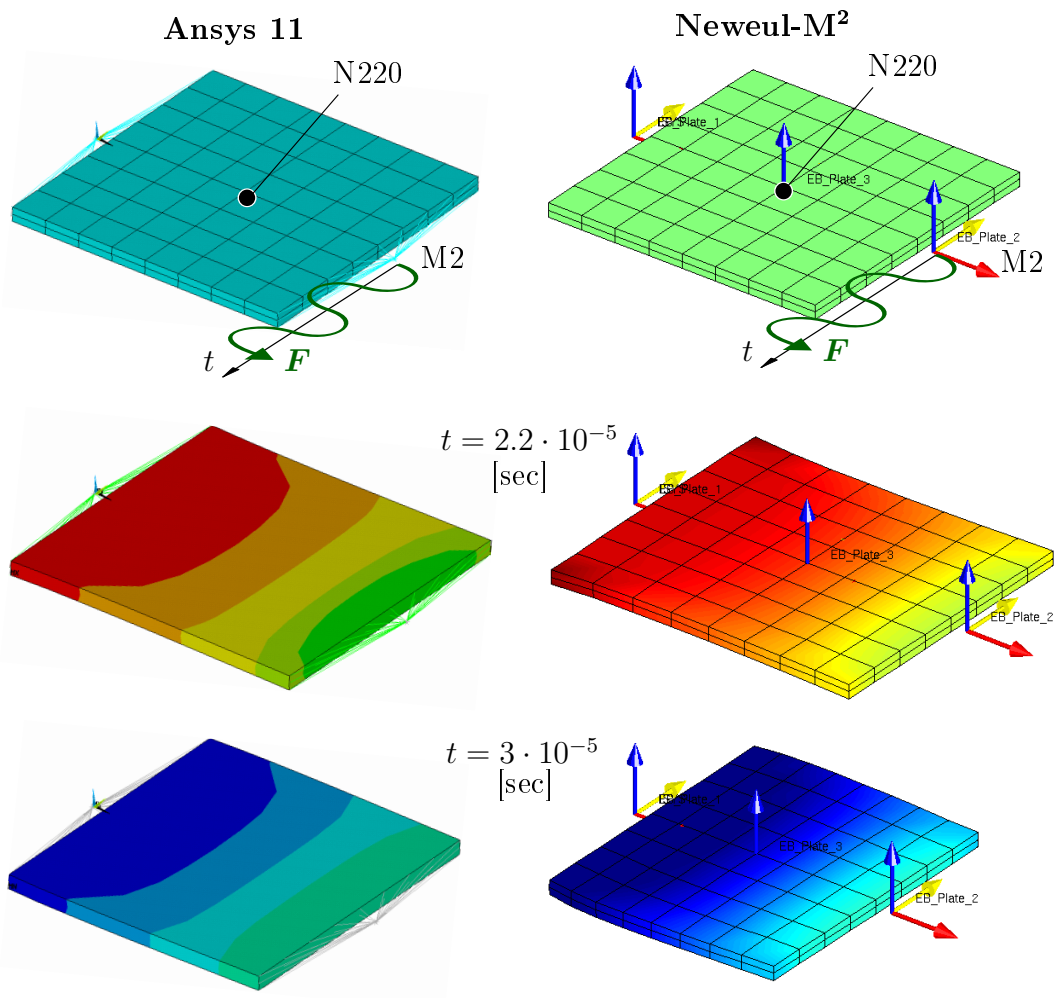


Abbildung 4.3: Validierung der Oberflächenspannungsanimation σ_x

Skalierung der Verschiebungen und Farben unterscheidet sich dabei unwesentlich. Für den numerischen Spannungsvergleich wird der Knotenpunkt N220 in der Mitte der Platte ausgewertet und verglichen. Die Simulationsergebnisse sind in Abbildung 4.4 dargestellt.

Es ist zu erkennen, dass die Knotenspannung von *Neweul-M²* mit den in *Ansys* berechneten Spannung kaum abweicht. Außerdem zeigt sich, dass die Spannung bei harmonischer Anregung steigt, da die Dämpfung vernachlässigt wird und das dynamische System somit instabil ist.

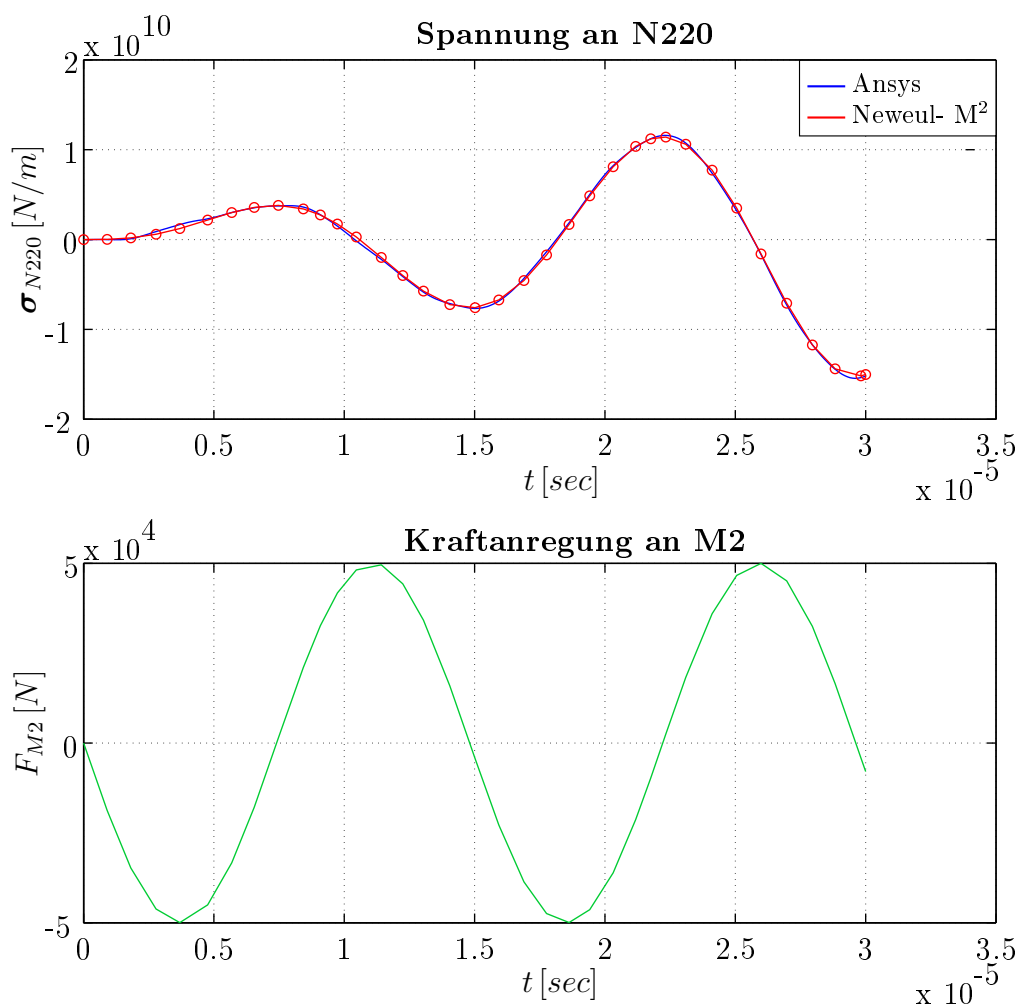


Abbildung 4.4: Validierung der Spannung am Knoten N220

4.3 Simulation des elastischen Linsensystems

Die Modellierung und Simulation des elastischen Mehrkörpersystems, dargestellt in Abbildung 4.1 erfolgt direkt in Neweul-M². Die verallgemeinerten Koordinaten des Gesamtsystems setzen sich aus den Starrkörperbegeugungen der Linsen und den elastischen Koordinaten der Platte wie folgt zusammen,

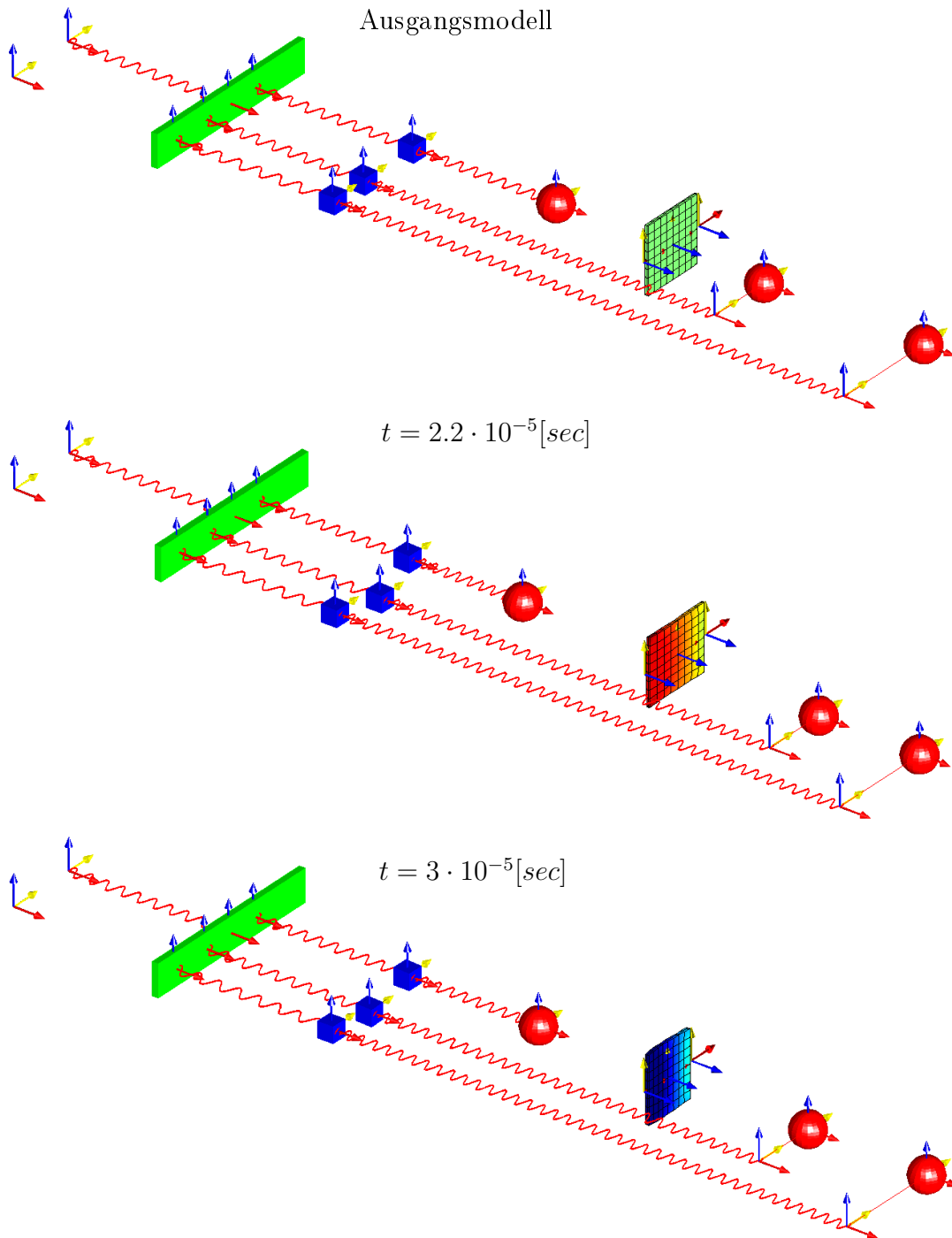
$$\bar{\mathbf{y}} = \left[\underbrace{s \quad a1 \quad a2 \quad b1 \quad b2 \quad e1 \quad e2}_{\bar{\mathbf{y}}_{starr}} \quad \underbrace{q01 \quad q02 \quad q03 \quad q04}_{\bar{\mathbf{q}}_{elastisch}} \right]^T. \quad (4.1)$$

Dabei ist die elastische Platte entkoppelt vom restlichen Teil des Linsensystem. Mit Hilfe von Neweul-M²-Funktionen werden die starren Linsen als Kugeln visualisiert. Die Federdämpfung wird vernachlässigt und die Federsteifigkeiten sind so angepasst, dass sie im Bereich der Plattensteifigkeit liegen, um bei einer Zeitintegration die Bewegungen aller verallgemeinerten Koordinaten zu animieren. In Abbildung 4.5 und 4.6 ist das Ergebnis einer beispielhaften Zeitintegration veranschaulicht, das aus den Anfangsbedingungen

$$\bar{\mathbf{y}}_0 = \left[\underbrace{0 \quad 0 \quad 0.005 \quad 0 \quad -0.01 \quad 0 \quad 0}_{\bar{\mathbf{y}}_{0starr}} \quad \underbrace{0 \quad 0 \quad 0 \quad 0}_{\bar{\mathbf{q}}_{0elastisch}} \right]^T \quad (4.2)$$

$$\dot{\bar{\mathbf{y}}}_0 = \left[\underbrace{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0}_{\dot{\bar{\mathbf{y}}}_{0starr}} \quad \underbrace{0 \quad 0 \quad 0 \quad 0}_{\dot{\bar{\mathbf{q}}}_{0elastisch}} \right]^T. \quad (4.3)$$

startet und die Dynamik bei einer harmonischen Plattenanregung mit der ersten Platten-Eigenfrequenz berechnet.

Abbildung 4.5: Animation des EMKS-Linsensystems in Neweul-M²

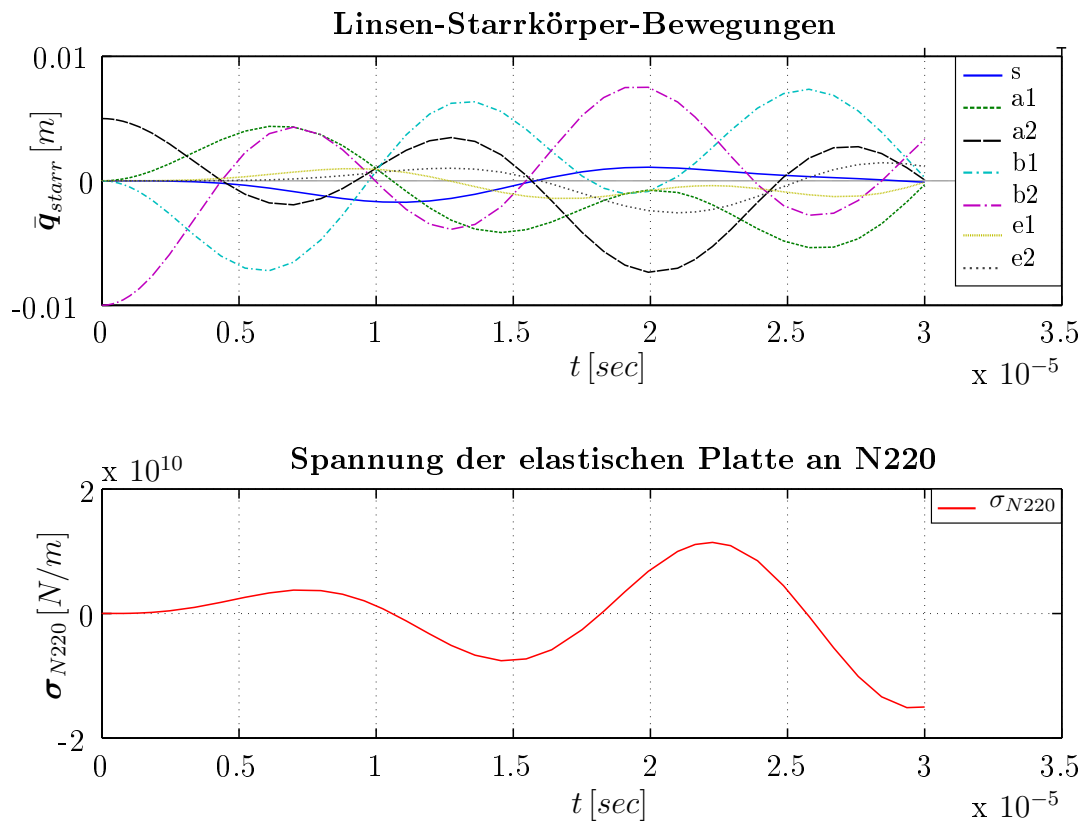


Abbildung 4.6: Grafik der Ergebnisse des EMKS-Linsensystems

Kapitel 5

Optische Simulation

In diesem Kapitel wird die Modellierung des Linsen-Objektivs als optisches System erläutert. Dies wird unabhängig vom mechanischen *Neweul-M²*-Modell, vgl. Kapitel 4, in *OM-Sim* durchgeführt. Anschließend wird eine automatische Wellenfrontfehlerkompensation entworfen, die bei der Verschiebung einer Linse getestet wurde. Für die Charakterisierung des Bildfehlers wird das vierte Zernike-Polynom verwendet, siehe Abbildung 2.6 und Abbildung 2.7. Im letzten Unterkapitel werden die Simulationsergebnisse der Bildfehler-Kompensation anhand beispielhafter Störungen aufgezeigt.

5.1 Modellierung in OM-Sim

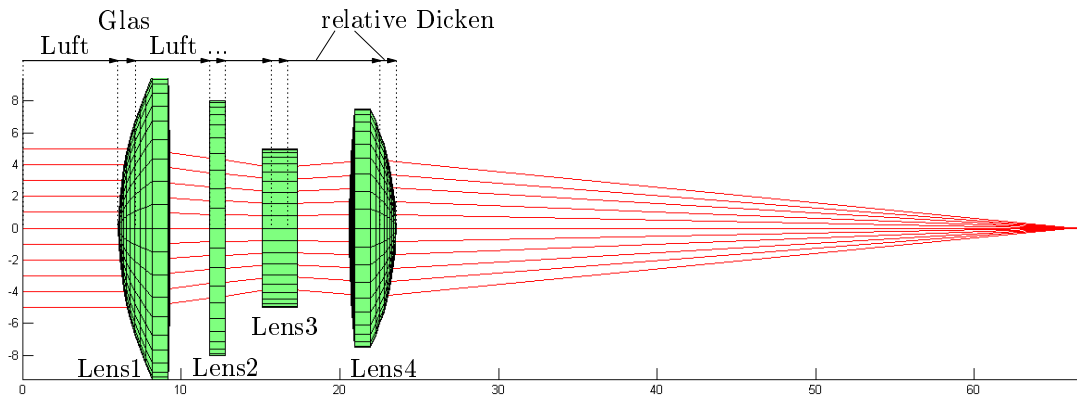
Die Modellierung des Linsensystems für die optische Simulation erfolgt in *OM-Sim* unter der *Matlab*-Umgebung. Im Unterschied zum mechanischen Modell sind die geometrische Definition und die optischen Parameter, wie z.B. der Brechungsindex der Linsen, entscheidend. Anhand dieser Informationen wird mittels *Raytracing* die *Aberration* berechnet. Durch die Definition des Radius, der Dicke und der Höhe werden die Parameter für die Erzeugung von sphärischen Flächen festgelegt, die die Grundlage der Linsengeometrie bilden.

Beginnend von links nach rechts werden die Oberflächen des optischen Modells relativ zu der jeweils links liegenden Fläche erzeugt. Der Glas-Parameter legt dabei fest, welche Materialeigenschaften vorliegen, bzw. ob es sich um Luft handelt, vgl. Abbildung 5.1. Während des *Raytracing*-Prozesses wird mit Hilfe einer Materialdatenbank auf die optischen Eigenschaften zugegriffen.

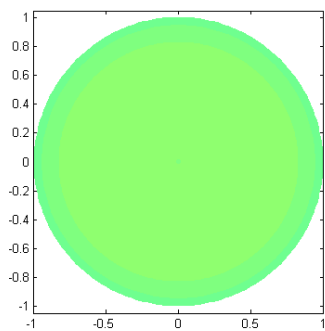
Als Beispiel wurde ein Linsen-*Triplet* ausgewählt, welches mit einer planen Glasplatte *Lens2* ergänzt wurde. Diese simuliert den elastischen Körper, der zur optischen Fehlerkompensation eingesetzt wird. Die Linsenabstände sind in der An-

fangskonfiguration so justiert, dass mit der unbelasteten Platte *Lens2* die Bildfehler näherungsweise verschwinden.

Raytracing-Visualisierung



Aberrations-Visualisierung



Bildschärfe-Visualisierung

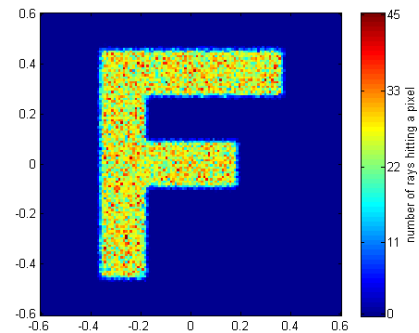


Abbildung 5.1: Darstellung der Anfangskonfiguration des Linsensystems, modelliert in der Optik-Software OM-Sim

5.2 Bildfehler-Kompensation

Für den Entwurf der Kompensation eines Wellenfrontfehlers wird das Linsenmodell abstrakt als dynamisches System betrachtet und mit regelungstechnischen Methoden beschrieben. Das Ziel ist ein Regelungsprozess, der das optische System unter Einwirkung einer einfachen Linsenverschiebung kompensiert.

5.2.1 Mechanisches Modell des optisches Systems

In Kapitel 5.1 ist das optische System in statischer Form beschrieben, es finden keine Bewegungen oder Kräfteinwirkungen statt. Dieser Zustand wird Anfangskonfiguration genannt. Für den Entwurf eines Bildfehler-Regelungsprozesses wird das System mechanisch erweitert, vgl. Abbildung 5.2.

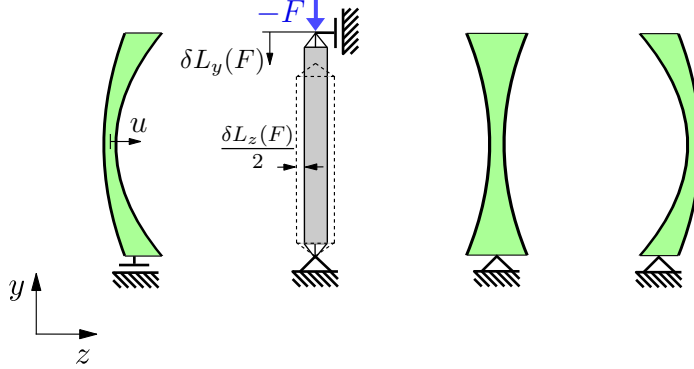


Abbildung 5.2: Darstellung des optischen Modells als dynamisches System

Die dynamische Kraft F wirkt auf die elastische Platte in y -Richtung und der Körper *Lens1* ist in z -Richtung frei gelagert. Relativ zur Anfangskonfiguration wird die Position der Linse mit der Verschiebung u festgelegt. Zur Vereinfachung wird die Platte als ideal elastisch angenommen und die Kraft verteilt sich homogen. Deshalb verhält sich die Plattenspannung proportional zur Kraft und wird in der gesamten Geometrie als konstant angesehen. Nach den elementaren Gleichungen der Festigkeitslehre ergeben sich die kraftabhängigen Änderungen der geometrischen Plattengrößen in y -Richtung ΔL_y und in z -Richtung ΔL_z zu

$$\Delta L_y(F) = L_{y0} \epsilon_y = L_{y0} \frac{F}{EA_{xz}} \quad (5.1)$$

$$\epsilon_z = -\epsilon_y \mu \quad (5.2)$$

$$\Delta L_z(F) = L_{z0} \epsilon_z = -L_{y0} \frac{F}{EA_{xz}} \mu \quad (5.3)$$

$$\sigma = \frac{F}{A_{xz}}. \quad (5.4)$$

Die Größen L_{y0} und L_{z0} stellen die Ausgangsmaße der unbelasteten Platte dar und ϵ_y bzw. ϵ_z die elastischen Dehnungen in den entsprechenden Richtungen. Dabei steht F für die Plattenkraft, A_{xz} für die Platten-Querschnittsfläche in der $x-z$ -Ebene und E für das Elastizitätsmodul des Plattenwerkstoffs. Das statische Optik-Modell aus Kapitel 5.1 wird dynamisch um die Plattenverformung und Linsenverschiebung erweitert.

5.2.2 Regelungstechnisches Modell

Durch die abstrakte Betrachtung des Linsenobjektivs als dynamisches System lässt sich die Aufgabe regelungstechnisch formulieren. Mit der Stellgröße F für die Plattenkraft, der Störgröße u für die fehlerhafte Verschiebung und der Regelgröße Z_4 , die einen Defokus durch das vierte Zernike-Polynom charakterisiert, sind die Systemgrößen vollständig definiert, vgl. Abbildung 5.3.

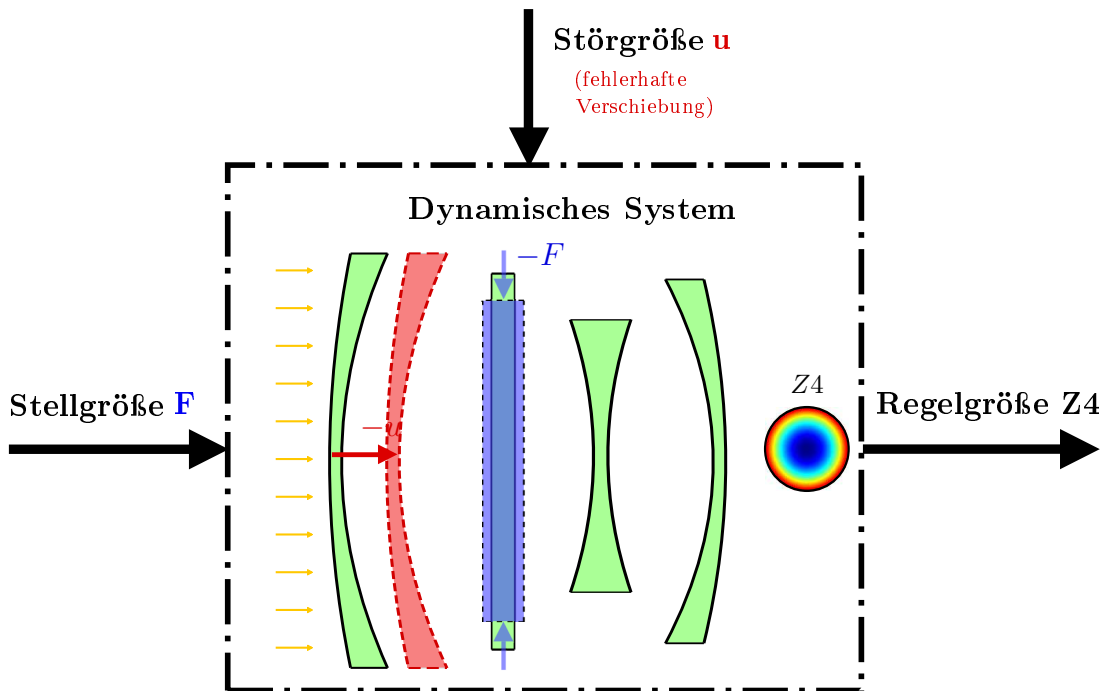
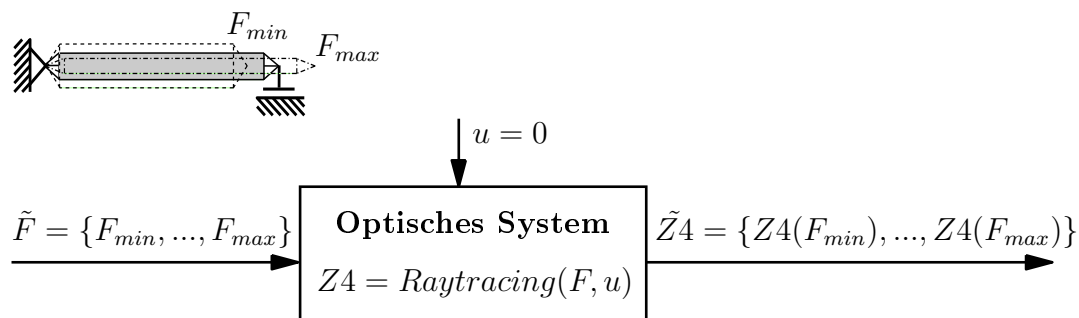


Abbildung 5.3: Darstellung des optischen Modells als dynamisches System

5.2.3 Entwurf eines statischen Reglers

Bei einem klassischen Reglerentwurf im Zeitbereich ist das Verhalten des dynamischen Systems bekannt und durch eine algebraische Differentialgleichung charakterisiert. Im hier vorliegenden Fall wird das optische System nur statisch zu bestimmten Zeitpunkten betrachtet und ggf. kompensiert. Außerdem handelt es sich bei der Regelgröße Z_4 um einen numerischen Wert, der durch das *Raytracing*-Verfahren berechnet wird. Daher können die systemtheoretischen Grundlagen eines Reglerentwurfs nicht angewandt werden.

Durch den Entwurf eines geeigneten Regelprozesses, illustriert in Abbildung 5.4, wird die nichtlineare Kompensation wie folgt ausgeführt:

Schritt 1: Diskrete Systemanalyse**Schritt 2:** Kontinuierliche Systembeschreibung / Invertierung

$$f(F) = \text{polyfit}(\tilde{F}, \tilde{Z4})$$

$$Z4 = f(F) \Rightarrow F = f^{-1}(Z4)$$

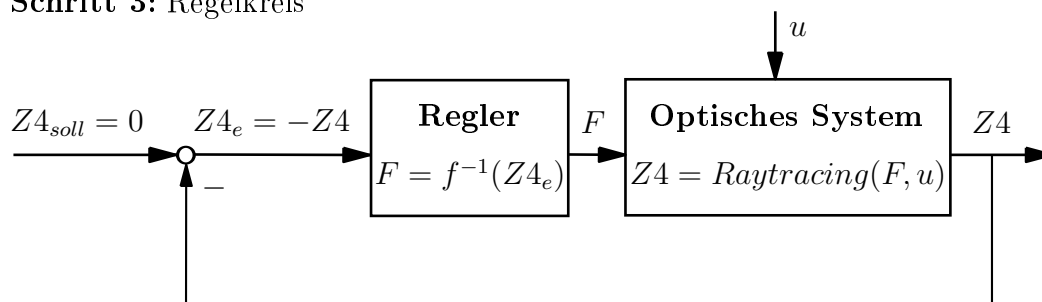
Schritt 3: Regelkreis

Abbildung 5.4: Darstellung des Regelprozesses

- Im ersten Schritt wird das optische System mit der Menge an Stellgrößen \tilde{F} und ohne Betrachtung der Störgröße u diskret analysiert. Dabei wird der Kräfteinfluss auf die Regelgröße $Z4$ untersucht. Durch die maximale Zug- und Druckkraft F_{max} und F_{min} wird die Stellgröße begrenzt. Die Diskretisierung wurde empirisch so fein gewählt, dass sich die Ergebnisse im nächsten Schritt nur marginal ändern. Es ergibt sich die Menge $\tilde{Z}4$, die den Zusammenhang zwischen der Kraft und dem dadurch resultierenden Bildfehler punktweise darstellt.
- Im zweiten Schritt werden die diskret berechneten Werte aus dem ersten Schritt durch eine Polynomfunktion $f(F)$ approximiert. Sie stellt somit das Übertragungsverhalten der Stellgröße auf die Regelgröße kontinuierlich dar. Da jeder Kraft nur ein Bildfehler zugeordnet ist und theoretisch jede Kraft erzeugt werden kann, ist die Funktion bijektiv und lässt sich invertieren. Mit Hilfe der invertierten Funktion lässt sich die Kraft berechnen, die für das Erzeugen eines Bildfehlers beaufschlagt werden muss.
- Schließlich kann im letzten Schritt der Regelkreis entworfen werden. Mit der Störgröße u wird zunächst der Bildfehler $Z4$ ermittelt und zurückgeführt. Um den Bildfehler vollständig zu kompensieren, wird die Sollgröße $Z4_{soll} = 0$ gesetzt. Die Regelabweichung ergibt sich somit zu $Z4_e = -Z4$. Mit Hilfe der Reglerfunktion f^{-1} kann die notwendige Stellgröße zur Kompensation berechnet und anschließend auf das optische System angewandt werden. Als Ergebnis wird der verschiebungsergebnisierte Bildfehler durch die Überlagerung des kraftresultierten Bildfehlers ausgeglichen.

Im Folgenden wird die entworfene Bildfehler-Kompensation anhand von simulierten Beispielstörungen mit *OM-Sim* getestet.

5.3 Simulation mit Bildfehler-Kompensation

Da das optische System in dieser Arbeit unabhängig vom komplexen mechanischen *Neweul-M²* Modell ist, wird die Regler-Simulation vollständig mit *OM-Sim* durchgeführt. Der Prozessablauf ist in Algorithmus 1 dargestellt. Die Feldgrößen und die reglerrelevanten Funktionsaufrufe sind fett formatiert. Diese Logik ist in einem *Matlab*-Skript umgesetzt, mit dem die Simulation durchgeführt wird.

Der Simulationsprozess ist so programmiert, dass die Verschiebungen und Spannungen parallel zur Rechnung animiert werden. Zur Verdeutlichung der Bildfehlerkorrektur werden zwei Fenster angezeigt. Im ersten Fenster wird das optische System *ohne* und im zweiten Fenster *mit* Kompensationsregler animiert. In der folgenden Abbildung 5.5 wird ein Zeitpunkt der Animation dargestellt. Die vierte

Algorithmus 1 Wellenfrontfehler-Kompensationsprozess

- 1: **Input:** Linsen-Geometrien, Material-Daten, $\mathbf{u}(\mathbf{t})$ and Randbedingungen
 - 2: **Output:** Animation, $\mathbf{Z4}_{uc}(\mathbf{t})$, $\mathbf{F}(\mathbf{t})$, $\mathbf{Z4}_c(\mathbf{t})$
 - 3: $optsys = DefineOptsys\{optsys\}$ /* Modellierung */
 - 4: $\sigma_{min} = \frac{F_{min}}{A_{xz}}$, $\sigma_{max} = \frac{F_{max}}{A_{xz}}$
 - 5: $\tilde{\mathbf{F}} = F_{max} : \frac{F_{max}-F_{min}}{n} : F_{min}$
 - 6: $\tilde{\mathbf{Z4}} = \mathbf{Raytracing}\{\tilde{\mathbf{F}}, 0\}$ /* Berechnung der kraftbeeinflusste Aberration */
 - 7: $f(F) = \mathbf{polyfit}\{\tilde{\mathbf{F}}, \tilde{\mathbf{Z4}}\}$
 - 8: $Z4 = f(F) \Rightarrow F = f^{-1}(Z4)$
 - 9: **for** $t = 0 : t_{end}$ **do**
 - 10: $Z4_{uc}(t) = \mathbf{Raytracing}\{0, u(t)\}$ /* unkompensierte Aberration */
 - 11: $animWind[1]\{0, u(t), Z4_{uc}(t)\}$ /* Animation des unregulierten Systems */
 - 12: $F(t) = f^{-1}(-Z4_{uc}(t))$
 - 13: $Z4_c(t) = \mathbf{Raytracing}\{F(t), u(t)\}$ /* kompensierte Aberration */
 - 14: $animWind[2]\{F(t), u(t), Z4_c(t)\}$ /* Animation des regulierten Systems */
 - 15: **end for**
 - 16: $plotResultSummary\{\mathbf{F}(\mathbf{t}), \mathbf{u}(\mathbf{t}), \mathbf{Z4}_{uc}(\mathbf{t}), \mathbf{Z4}_c(\mathbf{t})\}$
-

$t = 0.27$ [sec] $u = -0.3$ [mm]

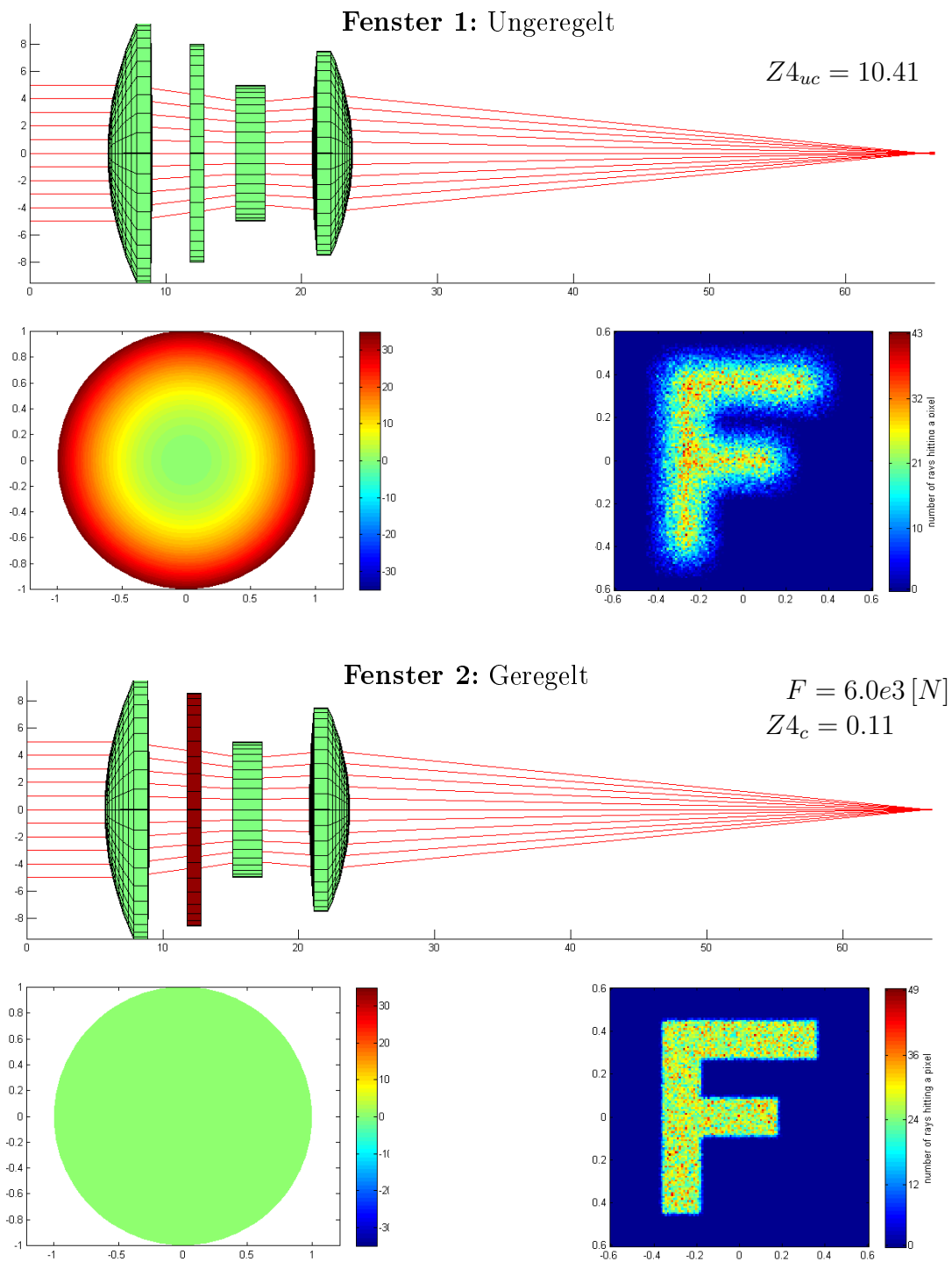


Abbildung 5.5: Darstellung der *OM-Sim*-Oberfläche: oben das ungeregelte Modell, unten das geregelte Modell

Linse wird in beiden Fenstern bewegt. Im unregelmäßigen System in Fenster 1 ist in der *Aberrations-Visualisierung* und *Bildschärfe-Visualisierung* deutlich erkennbar, dass ein Bildfehler vorliegt. Mit dem Gewichts-Faktor $Z_{4_{uc}}$ für das vierte Zernike-Polynom kann dieser in einer Zahl ausgedrückt werden. Das zweite Fenster zeigt den gleichen Zeitpunkt im geregelten Zustand. Mit Hilfe einer Farbänderung der Plattenlinse wird deren Spannung verdeutlicht. Die Kraft F , die zum Ausgleich nötig ist, und der kompensierte Rest-Fehler Z_{4_c} sind rechts angegeben.

Im Folgenden wird anhand von simulierten Beispiel-Störungen gezeigt, dass der Kompensationsprozess für die statische Verschiebung des Körpers *Lens1* sehr gut funktioniert.

5.3.1 Lineare Störung

Im ersten Beispiel wird von einer linearen Verschiebung des Körpers *Lens1* ausgegangen. Die zeitliche Positionsänderung wird beschrieben mit der Rampenfunktion

$$u_r(t) = \frac{u_{max}}{t_{end}} t. \quad (5.5)$$

Der Parameter u_{max} beschreibt die maximale Verschiebung zum Endzeitpunkt t_{end} . Neben diesen Anfangsbedingungen werden auch die Kraftbegrenzungswerte und die Anzahl der Iterationsschritte vom Benutzer vorgegeben. Zur Verdeutlichung der Animation werden die Linsenverschiebungen in überhöhter Form vorgegeben und der spannungsoptische Koeffizient der Glasplatte um den Faktor 500 erhöht. In Abbildung 5.6 sind die Ergebnisse dargestellt.

5.3.2 Harmonische Störung

Dieses Beispiel geht von einer harmonischen Verschiebung des Körpers *Lens1* aus. Die zeitliche Positionsänderung u_{harm} wird beschrieben mit der harmonischen Funktion

$$u_{harm}(t) = u_{max} \sin(\omega t + \Delta\phi). \quad (5.6)$$

Der Parameter Δu_{max} beschreibt die Amplitude, ω die Anregungskreisfrequenz und $\Delta\phi$ die Phasenverschiebung. Neben diesen Anfangsbedingungen werden wieder die Kraftbegrenzungswerte und die Anzahl der Iterationsschritte vom Benutzer vorgegeben. Alle weiteren Vorgaben entsprechen dem Beispiel aus Kapitel 5.3.1. In Abbildung 5.7 sind die Ergebnisse dargestellt.

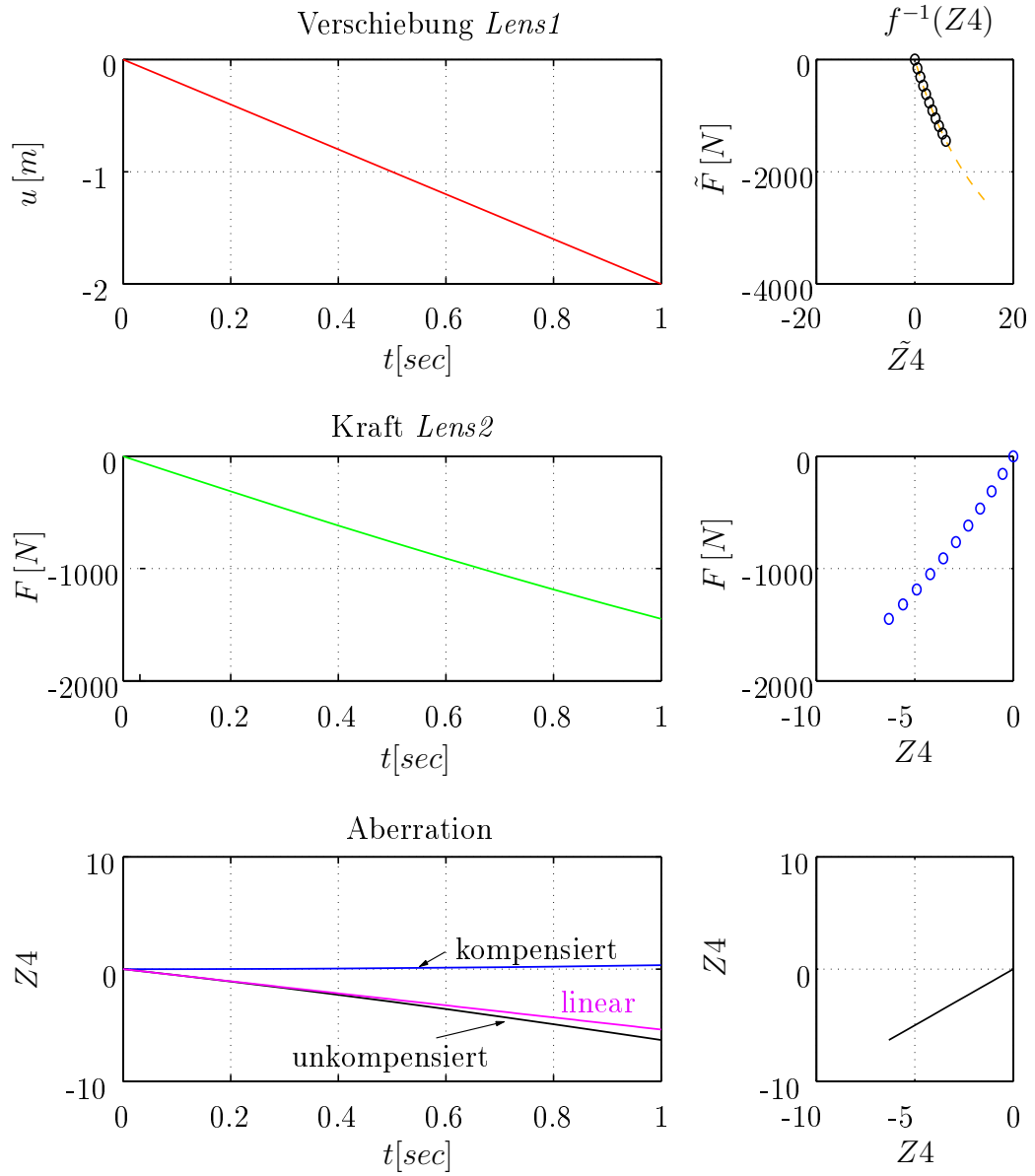


Abbildung 5.6: Darstellung der Regelung einer linearen Störung

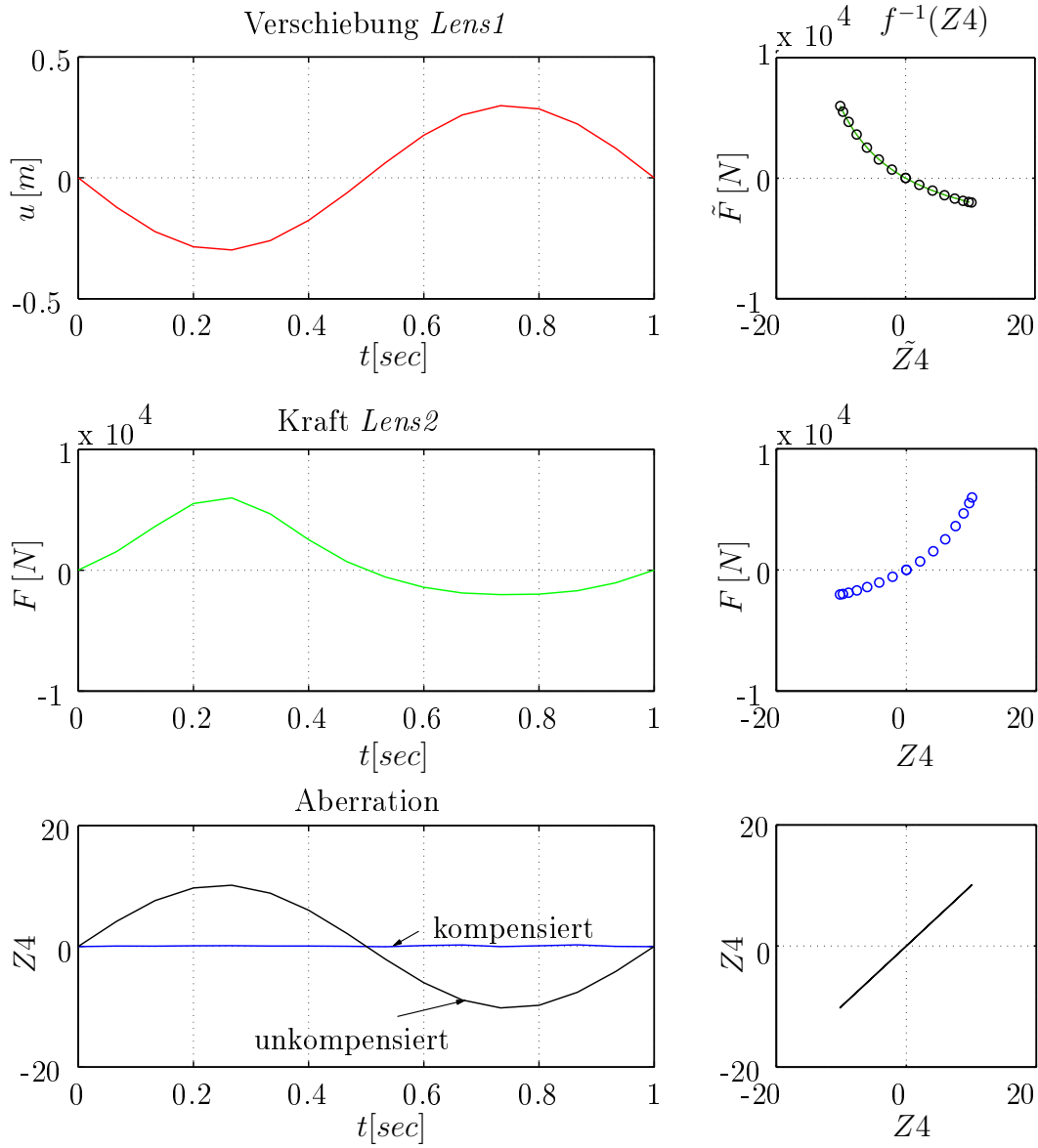


Abbildung 5.7: Darstellung der Regelung einer harmonischen Störung

5.3.3 Zufällige Störung

Im letzten Beispiel wird von einer zufälligen Verschiebung des Körpers *Lens1* ausgegangen. Die Positionsänderung u_{rand} wird in jedem Zeitschritt neu im folgenden Bereich generiert:

$$\frac{u_{max}}{2} < |u_{rand}| < u_{max}. \quad (5.7)$$

Durch die Variable Δu_{max} wird der Bereich der generierten Zufallswerte eingegrenzt. Neben diesen Anfangsbedingungen werden die Kraftbegrenzungswerte und die Anzahl der Iterationsschritte vom Benutzer vorgegeben. Alle weiteren Vorgaben entsprechen dem Beispiel aus Kapitel 5.3.1. In Abbildung 5.8 sind die Ergebnisse dargestellt.

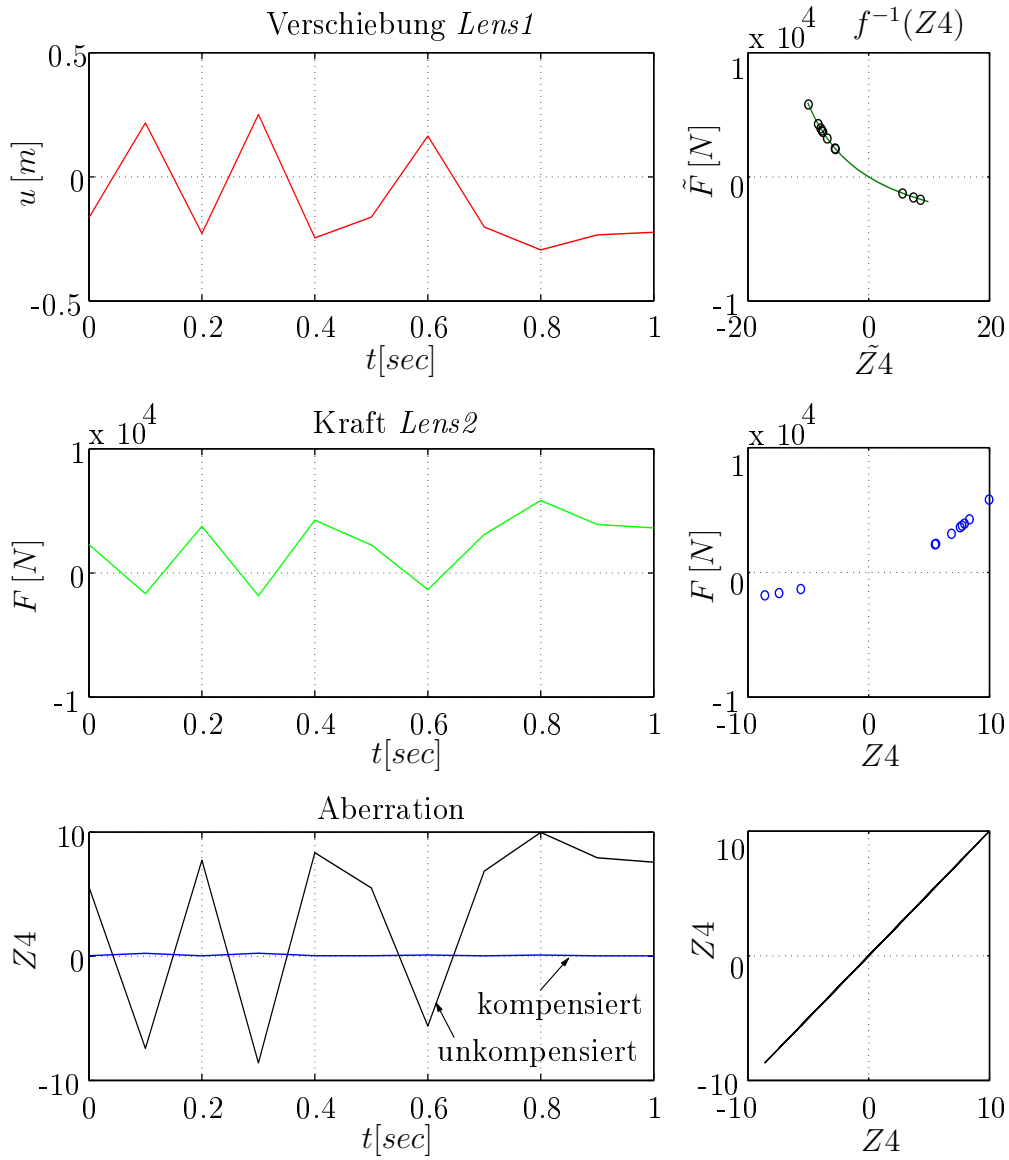


Abbildung 5.8: Darstellung der Regelung einer zufälligen Störung

Kapitel 6

Zusammenfassung und Ausblick

In dieser Arbeit wird die Methode der dynamischen Spannungsanalyse in *MatMormbs* und *Neweul-M²* implementiert. Damit können Spannungen in reduzierten elastischen Körpern mit *Neweul-M²* untersucht werden. Zum Einen können Spannungen an vorgegeben Knotenpunkten berechnet und zum Anderen können die dynamischen Oberflächenspannungen in einer Animation visualisiert werden.

Anhand des Beispiels einer modal reduzierten elastischen Glasplatte wird die Implementierung simulativ getestet und angewendet. Die Ergebnisse werden mit dem kommerziellen FE-Programm *Ansys* verifiziert.

Anschließend wird das mechanische Modell eines starren Linsensystems in *Neweul-M²* aufgebaut und die elastische Platte darin implementiert. Dieses Modell soll die Grundlage für zukünftige Arbeiten zur Regelung optischer Systeme bilden. Es liefert erste Aufschlüsse über die Dynamik der Linsenbewegung in Richtung der optischen Achse und ermöglicht gleichzeitig die Spannungsanalyse der Platte bei Kraftanregung. Eine Kopplung der elastischen Platte mit den Linsenstarrkörpern stellt eine mögliche Erweiterung des mechanischen Modells dar.

Unabhängig davon wird das optische Modell des Linsensystems in *OM-Sim* erstellt. Durch das Verschieben einer Linse wird ein Bildfehler erzwungen, der mittels einer kraft-beaufschlagten Glasplatte durch einen Kompensationsprozess ausgeglichen wird. Der Regelungsprozess wird für verschiedene Störungen getestet.

In einer weiter führenden Arbeit könnte das mechanische Linsen-EMKS mit dem optischen System gekoppelt werden. Dafür muss eine Schnittstelle der Geometrie-Größen des elastischen Körpers und der Starrkörper FHG geschaffen werden. Der Kompensationsprozess für das optische System muss außerdem für eine dynamische Regelung angepasst und erweitert werden.

Anhang

A.1 Dynamische Spannungsanalyse eines Krylov-reduzierten Modells

Die implementierte dynamische Spannungsanalyse wird im Folgenden anhand eines weiteren Modells getestet. Dieses Beispiel wurde auch in [TobiasEberhard11] verwendet und ist in Abbildung A.1 dargestellt. Die Verifizierung der dynamischen Analyse wird auf der Basis von Simulations-Ergebnissen durchgeführt, die mit dem FE-Programm *Permas* entsprechend [TobiasEberhard11] berechnet wurden.

Die Modelldateien zur Erzeugung des freien elastischen Körpers in *Ansys* liegen bereits aus [Schmitt11] vor. Dieses Modell soll zeigen, dass auch andere Reduktionsverfahren als die bereits getestete modale Reduktion genutzt werden können. Hier wird das Modell eines Trägers auf Basis von Krylov-Unterräumen erster Ordnung reduziert, siehe [Milakovic10]. Im elastischen Körper wird eine natürliche Dämpfung von 2% angenommen. Anschließend wird der reduzierte Körper in ein EMKS-Modell eingebaut und in *Neweul-M²* simuliert.

Das System wird mit den Anfangsbedingungen $\beta_1 = -63.7^\circ$ und $\beta_e = -63.7^\circ$ zeitintegriert. Die an dem *Retained Node* N857 ausgewerteten Spannungen sind als Graphen in Abbildung A.2 dargestellt. In dieser Abbildung sind zusätzlich die *Permas*-Ergebnisse aus [TobiasEberhard11] zum Vergleich dargestellt. Es ist zu erkennen, dass die *Neweul-M²*-Ergebnisse für die Normalspannungen nur geringfügig von den *Permas*-Ergebnissen abweichen. Die minimale Abweichung der Schubspannungsverläufe von den Referenzergebnissen wird hier nicht weiter untersucht. In Abbildung A.3 ist die Animation der Oberflächenspannung zu verschiedenen Zeitpunkten visualisiert.

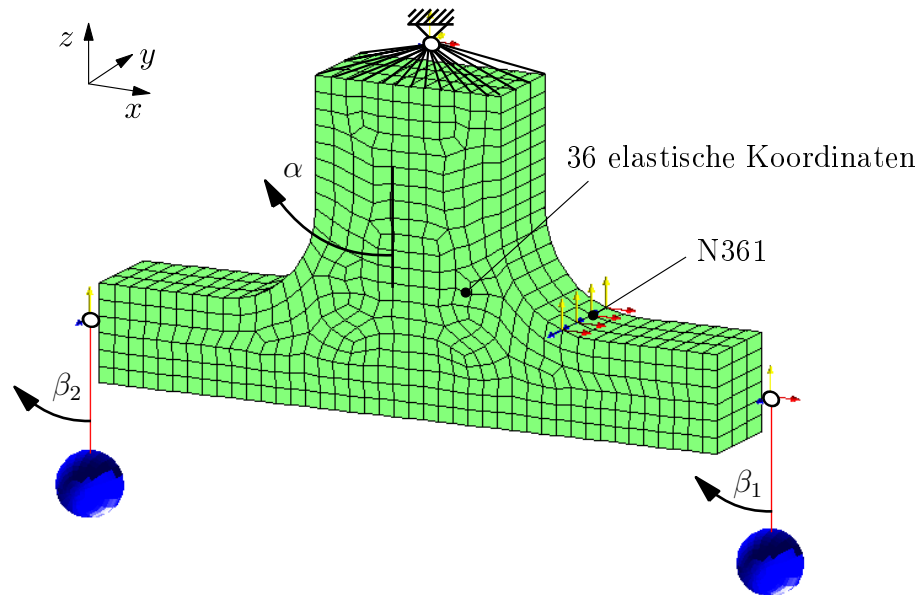


Abbildung A.1: Darstellung des Krylov-reduzierten EMKS-Modells

Simulationszeiten

Die Modellierung des EMKS und das Aufstellen der nichtlinearen Bewegungsgleichungen mit dem reduzierten Körper mit 36 FHG dauerte in *Neweul-M²* ca. 60 Minuten. Die Zeitintegration bis zu der Zeit $t = 0.3$ Sekunden dauerte ca. 40 Minuten. Ersetzt man den elastischen Körper mit einem Krylov-zweiter-Ordnung reduzierten Modell, das sich mit 72 FHG ergibt, so stößt man bei Zeiten von mehr als 8 Stunden schon an die Grenzen von *MatMorembs*. Das anschließende Aufstellen der Bewegungsgleichungen in *Neweul-M²* wurde nach 24 Stunden abgebrochen.

Im Vergleich dazu dauerte das Aufstellen des modal-reduzierten Linsenmodells aus Kapitel 4 weniger als eine Minute und die Zeitintegration im untersuchten Intervall weniger als 10 Sekunden.

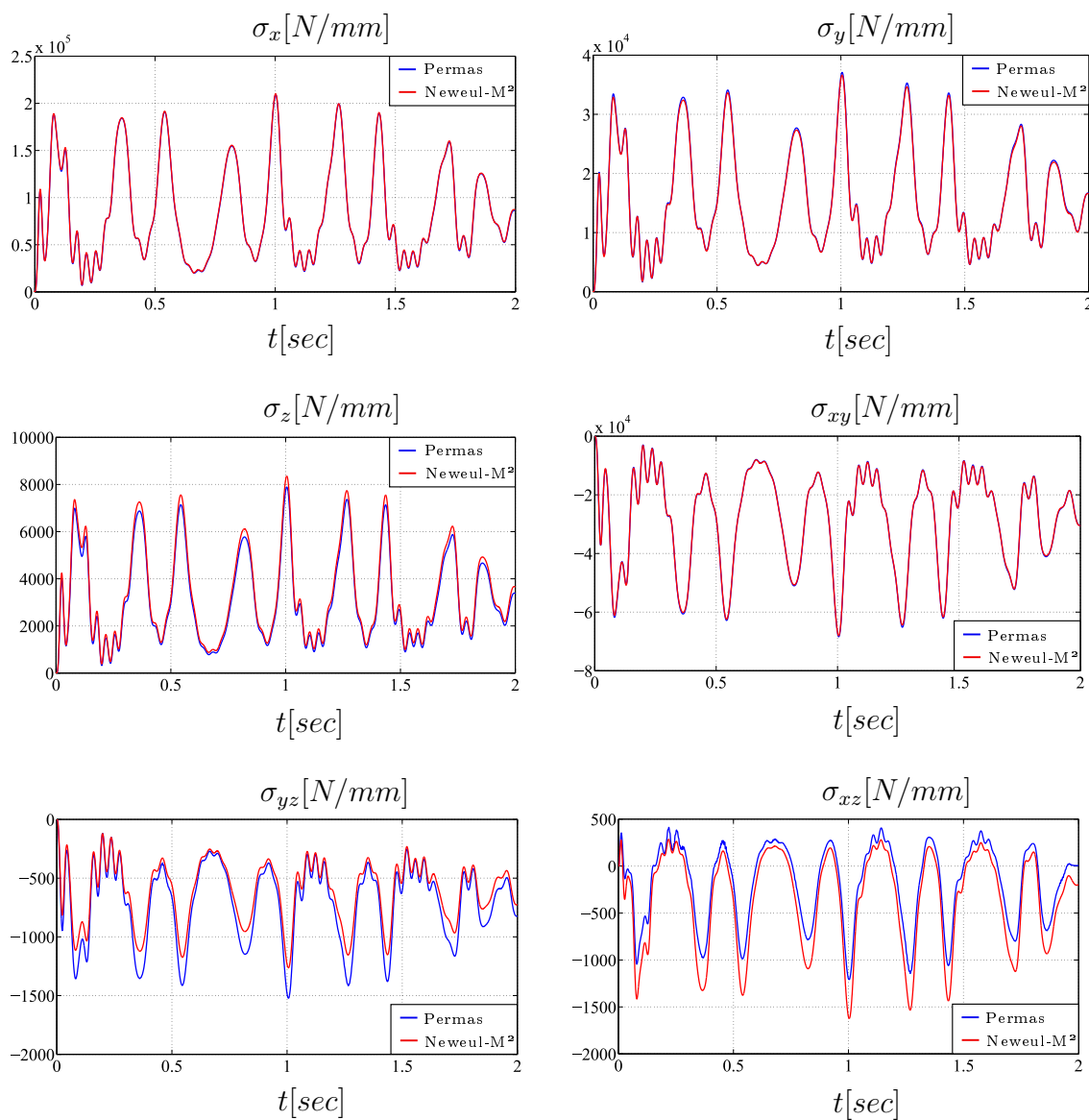


Abbildung A.2: Vergleich der Spannungsergebnisse von *Permas* und *Neweul-M²* am Knoten 857

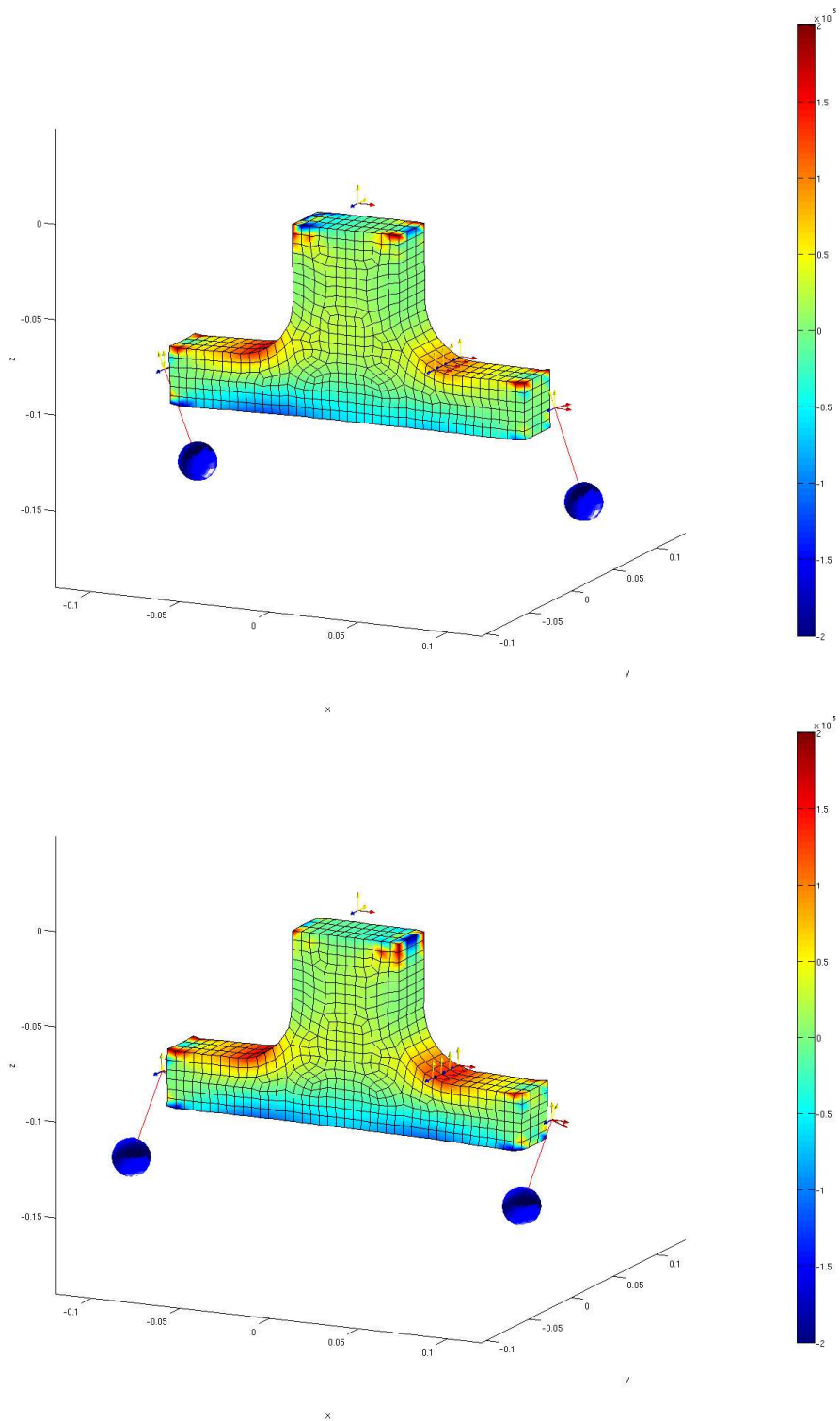


Abbildung A.3: Spannungsanimation der Oberfläche des krylovreduzierten EMKS-Modells in Neweul-M²

A.2 Inhalt der CD-ROM

Die beigelegte CD-ROM enthält in der obersten Dateistruktur die Einträge:

- **stud_367.pdf**: das PDF-File zur Studienarbeit STUD-367.
- **STUD_367/**: ein Verzeichnis mit den TEX-Dateien des in LaTeX verfassten Berichtes zur Studienarbeit STUD-367 sowie alle dazugehörigen Grafiken als *.eps und *.svg Dateien.
- **DATA/**: ein Verzeichnis mit den für diese Arbeit relevanten Daten, Hilfsprogrammen, Skripts und Simulationsumgebungen.

Zusätzliche Informationen stehen in den readme.txt-Dateien der jeweiligen Verzeichnisse zur Verfügung.

Abbildungsverzeichnis

2.1	Kinematik eines elastischen Körpers.	4
2.2	Prinzipdarstellung der modalen Reduktion auf Basis der ersten zwei Eigenmoden an dem Beispiel einer ebenen Platte.	7
2.3	Links: Schematische Matrizen-Berechnung einer ersten Spannungsmode des unreduzierten Modells. Rechts: Berechnung einer ersten reduzierten Spannungsmode mit Hilfe eines FE-Programms.	10
2.4	Prozesskette zur Modellreduktion mittels Morembs	11
2.5	Darstellung der berechneten Strahlen beim <i>Raytracing</i> mit <i>OM-Sim</i>	12
2.6	Veranschaulichung eines Defokus als Wellenfrontfehler (4. Zernike-Polynom), resultierend aus einer Linsenverschiebung	13
2.7	Darstellung der ersten sechs Zernike-Polynome	14
3.1	Klassischer Modell-Reduktions-Prozess mit anschließender EMKS Simulation in <i>Neueul-M²</i>	18
3.2	Modell-Reduktions-Prozess mit Berechnung der Spannungsmoden in <i>SID-Form</i> . Anschließend EMKS-Simulation in <i>Simpack</i> mit Spannungsanalyse, vgl. Abbildung 3.1 und Kapitel 2.1.3	20
3.3	Modell-Reduktions-Prozess mit Spannungsmodenberechnung. Anschließend EMKS Simulation in <i>Neueul-M²</i> mit Animation der dynamischen Oberflächenspannungen. Die neu implementierten Funktionen sind in rot dargestellt.	22
3.4	Darstellung der <i>ElasticBody</i> -Struktur von <i>MatMorembs</i>	23
3.5	Darstellung der <i>sys</i> -Struktur von <i>Neueul-M²</i>	24

3.6	Darstellung der Spannung in x-Richtung eines zugbelasteten elastischen Körpers, animiert in <i>Neweul-M²</i>	27
4.1	Darstellung des elastischen Linsensystems, modelliert als EMKS	28
4.2	Darstellung des Ansys-Modells	30
4.3	Validierung der Oberflächenspannungsanimation σ_x	31
4.4	Validierung der Spannung am Knoten N220	32
4.5	Animation des EMKS-Linsensystems in <i>Neweul-M²</i>	34
4.6	Grafik der Ergebnisse des EMKS-Linsensystems	35
5.1	Darstellung der Anfangskonfiguration des Linsensystems, modelliert in der Optik-Software <i>OM-Sim</i>	37
5.2	Darstellung des optischen Modells als dynamisches System	38
5.3	Darstellung des optischen Modells als dynamisches System	39
5.4	Darstellung des Regelprozesses	40
5.5	Darstellung der <i>OM-Sim</i> -Oberfläche: oben das unregelte Modell, unten das geregelte Modell	43
5.6	Darstellung der Regelung einer linearen Störung	45
5.7	Darstellung der Regelung einer harmonischen Störung	46
5.8	Darstellung der Regelung einer zufälligen Störung	48
A.1	Darstellung des Krylov-reduzierten EMKS-Modells	51
A.2	Vergleich der Spannungsergebnisse von <i>Permas</i> und <i>Neweul-M²</i> am Knoten 857	52
A.3	Spannungsanimation der Oberfläche des krylovreduzierten EMKS-Modells in <i>Neweul-M²</i>	53

Literaturverzeichnis

- [Bathe96] Bathe, K.J.: Finite Element Procedures. Upper Saddle River: Prentice-Hall, 1996.
- [EberhardWengertTouihri10] Eberhard, P.; Wengert, N.; Touihri, A.: Multibody Simulation of Optical Lens Systems to Analyze Image Aberrations. In Proceedings of the 5th Asian Conference on Multibody Dynamics, 23–26 August 2010, Kyoto, Japan. 2010.
- [FehrEberhard09] Fehr, J.; Eberhard, P.: Improving the Simulation Process in Flexible Multibody Dynamics by Enhanced Model Order Reduction Techniques. In Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics, 29 June – 2 July 2009, Warsaw, Poland. 2009.
- [Lehner07] Lehner, M.: Modellreduktion in elastischen Mehrkörpersystemen (in German). Dissertation, Schriften aus dem Institut für Technische und Numerische Mechanik der Universität Stuttgart, Band 10. Aachen: Shaker Verlag, 2007.
- [Milakovic10] Milakovic, S.: Dynamische Spannungsberechnung mit Krylov-Unterraummethoden. Studienarbeit STUD-347, Institut für Technische und Numerische Mechanik, Universität Stuttgart, 2010.
- [PedrottiEtAl02] Pedrotti, F.; Pedrotti, L.; Bausch, W.; Schmitt, H.: Optik für Ingenieure. Springer Verlag, Berlin, 2002.
- [Schmitt11] Schmitt, A.: Untersuchung von Ansätzen zur ganzheitlichen Strukturoptimierung elastischer Mehrkörpersysteme mit Spannungsnebenbedingungen. Studienarbeit STUD-359, Institut für Technische und Numerische Mechanik, Universität Stuttgart, 2011.
- [SchwertassekWallrapp99] Schwertassek, R.; Wallrapp, O.: Dynamik flexibler Mehrkörpersysteme (in German). Braunschweig: Vieweg, 1999.

- [Shabana98] Shabana, A.A.: Dynamics of Multibody Systems. Cambridge: Cambridge University Press, 1998.
- [TobiasEberhard11] Tobias, C.; Eberhard, P.: Stress Recovery with Krylov-Subspaces in Reduced Elastic Multibody Systems. Multibody System Dynamics, Bd. 25, Nr. 4, S. 377–393, 2011.
- [WengertEberhard11] Wengert, N.; Eberhard, P.: Archive of Mechanical Engineering, 2011 (Zur Veröffentlichung angenommen). 2011.
- [Willner04] Willner, K.: Die Methode der finiten Elemente. Universität Stuttgart, Institut für Angewandte und Experimentelle Mechanik, 2004.