



ENTWURF EINER FEHLERÜBERWACHTEN MODELLREDUKTION BASIEREND AUF KRYLOV-UNTERRAUMVERFAHREN UND ANWENDUNG AUF EIN STRUKTURMECHANISCHES MODELL

David Bernstein
Matrikelnummer 3473601

DIPLOMARBEIT

zur Erlangung des akademischen Grades

DIPLOMINGENIEUR (DIPL.-ING.)

Betreuer
Dipl.-Ing. Claudius Lein

Betreuender Hochschullehrer
Prof. Dr.-Ing. Michael Beitelschmidt

Eingereicht am 29.04.2014, überarbeitete Version vom 27.06.2014

KURZFASSUNG

Die FEM-MKS-Kopplung erfordert Modellordnungsreduktions-Verfahren, die mit kleiner reduzierter Systemdimension das Übertragungsverhalten mechanischer Strukturen abbilden. Rationale Krylov-Unterraum-Verfahren, basierend auf dem Arnoldi-Algorithmus, ermöglichen solche Abbildungen in frei wählbaren, breiten Frequenzbereichen. Ziel ist der Entwurf einer fehlerüberwachten Modelreduktion auf Basis von Krylov-Unterraumverfahren und Anwendung auf ein strukturelles mechanisches Modell.

Auf Grundlage der Software MORPACK wird eine Arnoldi-Funktion erster Ordnung um interpolativen Startvektor, Eliminierung der Starrkörperbewegung und Reorthogonalisierung erweitert. Diese Operationen beinhaltend, wird ein rationales, interpolatives SOAR-Verfahren entwickelt. Ein rationales Block-SOAR-Verfahren erweist sich im Vergleich als unterlegen. Es wird interpolative Gleichgewichtung verwendet. Das Arnoldi-Verfahren zeichnet sich durch geringeren Berechnungsaufwand aus. Das rationale, interpolative SOAR liefert kleinere reduzierte Systemdimensionen für gleichen abgebildeten Frequenzbereich. Die Funktionen werden auf Rahmen-, Getriebegehäuse- und Treibsatzwellen-Modelle angewendet.

Zur Fehlerbewertung wird eigenfrequenzbasiert ein \mathcal{H}_2 -Integrationsbereich festgelegt und der übertragungsfunktionsbasierte, relative \mathcal{H}_2 -Fehler berechnet.

Es werden zur Lösung linearer Gleichungssysteme mit MATLAB entsprechende Löser-Funktionen, auf Permutation und Faktorisierung basierend, implementiert.

ABSTRACT

FEM-MKS-coupling requires model order reduction methods to simulate the frequency response of mechanical structures using a smaller reduced representation of the original system. Most of the rational Krylov-subspace methods are based on Arnoldi-algorithms. They allow to represent the frequency response in freely selectable, wide frequency ranges. Subject of this thesis is the implementation of an error-controlled model order reduction based on Krylov-subspace methods and the application to a mechanical model.

Based on the MORPACK software, a first-order-Arnoldi function is extended by an interpolative start vector, the elimination of rigid body motion and a reorthogonalization. Containing these functions, a rational, interpolative Second Order Arnoldi (SOAR) method is designed that works well compared to a rational Block-SOAR-method. Interpolative equal weighting is used. The first-order-Arnoldi method requires less computational effort compared to the rational, interpolative SOAR that is able to compute a smaller reduction size for same frequency range of interest. The methods are applied to the models of a frame, a gear case and a drive shaft.

Error-control is realized by eigenfrequency-based \mathcal{H}_2 -integration-limit and relative \mathcal{H}_2 -error based on the frequency response function.

For solving linear systems of equations in MATLAB, solver functions based on permutation and factorization are implemented.

INHALTSVERZEICHNIS

Kurzfassung	I
Abstract	I
Symbolverzeichnis	VII
Abkürzungsverzeichnis	VII
Vorwort	1
1. Einleitung	1
1.1. Motivation	1
1.2. Einordnung	2
1.3. Aufbau der Arbeit	3
2. Theorie	4
2.1. Simulationsmethoden	4
2.1.1. Finite Elemente Methode	4
2.1.2. Mehrkörpersimulation	5
2.1.3. Kopplung der Simulationsmethoden	5
2.2. Zustandsraumdarstellung und Reduktion	7
2.3. Krylov Unterraum Methoden	8
2.4. Arnoldi-Algorithmen erster Ordnung	10
2.5. Arnoldi-Algorithmen zweiter Ordnung	14
2.6. Korrelationskriterien	17
2.6.1. Eigenfrequenzbezogene Kriterien	18
2.6.2. Eigenvektorbezogene Kriterien	18
2.6.3. Übertragungsfunktionsbezogene Kriterien	19
2.6.4. Fehlerbewertung	22
2.6.5. Anwendung auf Systeme sehr großer Dimension	22
3. Numerik linearer Gleichungssysteme	23
3.1. Grundlagen	23
3.2. Singularität der Koeffizientenmatrix	25
3.2.1. Randbedingungen des Systems	25
3.2.2. Verwendung einer generellen Diagonalperturbation	25
3.3. Iterative Lösungsverfahren	26

3.4.	Faktorisierungsverfahren	27
3.4.1.	Cholesky-Faktorisierung	28
3.4.2.	LU-Faktorisierung	33
3.4.3.	Fillin-Reduktion durch Permutation	34
3.4.4.	Fazit	38
3.5.	Direkte Lösungsverfahren	39
3.6.	Verwendung externer Gleichungssystem-Löser	40
3.7.	Zusammenfassung	41
4.	Implementierung	42
4.1.	Aufbau von MORPACK	42
4.2.	Anforderungen an Reduktions-Funktionen	43
4.3.	Eigenschaften und Optionen der KSM-Funktionen	45
4.3.1.	Arnoldi-Funktion erster Ordnung	45
4.3.2.	Rationale SOAR-Funktionen	45
4.4.	Korrelationskriterien	47
4.4.1.	Eigenfrequenzbezogen	47
4.4.2.	Eigenvektorbezogen	48
4.4.3.	Übertragungsfunktionsbezogen	48
4.5.	Lösungsfunktionen linearer Gleichungssysteme	50
4.5.1.	Anforderungen und Aufbau	50
4.5.2.	Verwendung der Gleichungssystem-Löser	50
4.5.3.	Hinweise zur Implementierung von Gleichungssystem-Lösern	53
5.	Anwendung	56
5.1.	Versuchsmodelle	56
5.1.1.	Testmodelle kleiner Dimension	56
5.1.2.	Getriebegehäuse	58
5.1.3.	Treibsatzwelle	59
5.2.	Validierung der Reduktionsmethoden an kleinem Modell	60
5.2.1.	Modifizierte Arnoldi-Funktion erster Ordnung	60
5.2.2.	Rationale SOAR-Funktionen	64
5.2.3.	Zusammenfassung	71
5.3.	Anwendung der KSM auf große Modelle	71
5.3.1.	Getriebegehäuse	72
5.3.2.	Treibsatzwelle	78
5.4.	Auswertung	84
6.	Zusammenfassung und Ausblick	86
6.1.	Zusammenfassung	86
6.2.	Ausblick	87
	Abbildungsverzeichnis	88

Tabellenverzeichnis	89
Diagrammverzeichnis	90
Literaturverzeichnis	93
A. Anhang	A-1
A.1. Modelle zum Test des Lösens linearer Gleichungssysteme	A-1
A.2. Vergleich verschiedener Normierungen beim Erstellen der Übertragungsfunktionsmatrizen	A-2
A.3. Weitere Diagramme und Tabellen zur Validierung der Reduktionsmethoden an kleinem Modell	A-5
A.3.1. Diagramme zur Reorthogonalisierung	A-5
A.3.2. Diagramme zum Netzvergleich	A-5
A.3.3. Diagramme und Tabellen zur Wahl des optimalen interpolativen Startvektors	A-9
A.3.4. Tabellen zum Vergleich interpolativer und Block-SOAR	A-11
A.3.5. Diagramme zur Wahl einzelner Entwicklungsstellen	A-13
A.3.6. Diagramme zur Reduktionsgüte in Abhängigkeit der reduzierten Systemdimension	A-17
A.3.7. Diagramme zur Kombination von Entwicklungsstellen	A-18
A.4. Weitere Diagramme zur Validierung der gewonnenen Erkenntnisse an Modellen großer Dimension	A-19

SYMBOLVERZEICHNIS

allgemeine Variablen

Matrizen

M	Massenmatrix des vollen Systems
D	Dämpfungsmatrix des vollen Systems
K	Steifigkeitsmatrix des vollen Systems
B	Eingangsmatrix des vollen Systems
C	Ausgangsmatrix des vollen Systems
$H(s)$	Übertragungsfunktionsmatrix des vollen Systems
Φ	Modalmatrix des vollen Systems
Λ	Spektralmatrix der Eigenwerte des vollen Systems
\bar{M}	Massenmatrix des reduzierten Systems
\bar{D}	Dämpfungsmatrix des reduzierten Systems
\bar{K}	Steifigkeitsmatrix des reduzierten Systems
\bar{B}	Eingangsmatrix des reduzierten Systems
\bar{C}	Ausgangsmatrix des reduzierten Systems
$\bar{H}(s)$	Übertragungsfunktionsmatrix des reduzierten Systems
$\bar{\Phi}$	Modalmatrix des reduzierten Systems
B_f	Eingangsmatrix der Darstellungsform erster Ordnung
C_f	Ausgangsmatrix der Darstellungsform erster Ordnung
E, A	Systemmatrizen der Darstellungsform erster Ordnung
G	initiale Basis, meist die Starrkörpermoden des vollen Systems

Q	Gewichtungsmatrix
V	Projektionsmatrix
Vektoren	
$\phi, \bar{\phi}$	Eigenvektor des vollen bzw. reduzierten Systems
ψ	expandierter Eigenvektor des reduzierten Systems
$\bar{\mathbf{q}}$	Vektor der Koordinaten des reduzierten Systems
b	Eingangsvektor oder Startvektor
c	Ausgangsvektor
f	Vektor der äußeren Lasten
h	Vektor der gyroskopischen Terme
j	Vektor der Ordnung der Reihenentwicklung für verschiedene Entwicklungsstellen
q	Vektor der Koordinaten des vollen Systems
s_{ES}	Vektor der Entwicklungsstellen
t_k	Wichtungsvektor, auch Satz tangentialer Richtungen genannt
u	Vektor der äußeren Verschiebungen
Skalare	
$\epsilon_{\mathcal{H}_2}^{rel}(s)$	relativer \mathcal{H}_2 -Fehler
$\epsilon_{H_F}^{rel}(s)$	relativer Frobenius-Fehler
$\omega, \bar{\omega}$	Eigenkreisfrequenz des vollen bzw. reduzierten Systems
D	LEHRsches Dämpfungsmaß
<i>def_{tol}</i>	Deflationstoleranz für Deflation der Orthogonalisierung
f	Frequenz oder Eigenfrequenz

m	Anzahl der zur Berechnung der Übertragungsfunktion des vollen Systems verwendeten Eigenmoden
N	Systemdimension des vollen Systems
n	Systemdimension des reduzierten Systems
r	Anzahl der Eingänge
s	komplexer Frequenzparameter
s_{ES}, f_{ES}	Entwicklungsstelle

Variablen für lineare Gleichungssysteme

α	Diagonalperturbation
A	quadratische Koeffizientenmatrix
B	Matrix der rechten Seite
b	Vektor der rechten Seite
F	Fehlermatrix
L	linke oder untere Dreiecksmatrix
U	rechte oder obere Dreiecksmatrix
X	Lösungsmatrix
x	Lösungsvektor

Operatoren

\dot{x}	Erste zeitliche Ableitung von x
\ddot{x}	Zweite zeitliche Ableitung von x
$\ \dots\ _2$	Euklidische Norm
$\ \dots\ _F$	Frobeniusnorm
$\ \dots\ _{\mathcal{H}_2}$	\mathcal{H}_2 -Norm, entspricht in dieser Arbeit definierter \tilde{H}_2 -Norm

ABKÜRZUNGSVERZEICHNIS

BT	Balanced Truncation
EF	Eigenfrequenz
EMKS	Elastische Mehrkörpersimulation
FEM	Finite Elemente Methode
FE	Finites Element
LGS	lineares Gleichungssystem
GUI	Grafischen Benutzeroberfläche
KSM	Krylov-Unterraum-Methoden oder Krylov subspace methods
MAC	Modal Assurance Criterion
MIMO	Multi-Input-Multi-Output
MIRIAM	MIMO Iterative Rational Interpolation Algorithm
MKS	Mehrkörpersimulation
modMAC	modifiziertes MAC
MOR	Modellordnungsreduktion
MORPACK	Modellordnungsreduktions-Paket
MPC	Multi Point Constraints
NNE	Nicht-Null-Elemente
NRFD	Normalized Relative Frequency Difference
pcg	preconditioned conjugate gradients
SISO	Single-Input-Single-Output
SOAR	Second Order Arnoldi

VORWORT

Die vorliegende Arbeit bildet den Abschluss meines Diplomstudiums, welches viele positive Überraschungen mit sich brachte und einen prägenden Einfluss auf mein Leben hatte. Es würde den Rahmen eines Vorwortes sprengen, all denen zu danken, die durch ihre Freundschaft und Hilfsbereitschaft das Studium für mich möglich gemacht haben. So seien hier nur diejenigen genannt, auf deren Mitwirken ich beim Erstellen dieser Arbeit nicht hätte verzichten können.

Ich danke Professor Beitelschmidt vor allem für das gute Management der Professur und die Prägung einer freundlichen Atmosphäre der Hilfsbereitschaft unter den Mitarbeitern, auf die auch wir Studenten immer zählen konnten. Durch die gemeinsame Arbeit mit anderen Studenten im Computer-Pool der Professur und den verbundenen Austausch bin ich überhaupt erst in der Lage gewesen, einen für die Arbeit notwendigen Wissensstand zu erlangen.

Ich danke meinem Betreuer Dipl.-Ing. Claudius Lein für seine große Geduld und Energie, sich in immer neue Problemstellungen hineinzudenken und trotz übervollem Terminkalender wöchentliche Beratungen zu ermöglichen. Eine bessere Diplombetreuung ist für mich kaum vorstellbar.

Ich danke meinen Mitdiplomanden, Jan Paulke für die langen Diskussionen auf Grundlage seiner tiefen, theoretischen Einarbeitung in viele Aspekte der Modellordnungsreduktion, die mein Theoriewissen maßgeblich geprägt haben, Andre Müller für die Modellbereitstellung und seine FE-Anwendungskenntnisse, die meine Einarbeitungszeit verkürzt haben, Florian Immel und Michael Lenz für viele hilfreiche bis existentielle Ratschläge.

Vor allem danke ich jedoch meiner Frau Lisa für ihre geduldige, bedingungslose Liebe, meiner ganzen Familie für ihren starken Rückhalt und Jesus Christus für die innere Sicherheit, immer neue Ideen und Rettung aus vielen scheinbaren Sackgassen.

1. EINLEITUNG

1.1. MOTIVATION

Über viele Jahrzehnte hinweg wurde im Maschinenbau die Festigkeit der gefertigten Bauteile mithilfe einfacher Grundlagenformeln sichergestellt. Effektiver Materialeinsatz spielte dabei eine untergeordnete Rolle und unbeabsichtigtes Versagen wurde häufig erst durch Tests am fertigen Bauteil erkannt. Mit der Möglichkeit computergestützter Simulation beliebig komplexer Bauteile erhöhten sich auch die Anforderungen an den Materialeinsatz und die Vorhersagbarkeit des Bauteil- und Maschinenverhaltens [LB11, S.127f]. Dabei entwickelten sich unabhängig voneinander zwei grundlegend verschiedene Berechnungsmethoden: die Finite Elemente Methode (FEM), mit der u.a. kleine Verformungen elastischer Körper unter Verwendung großer Systeme simuliert werden können, und die Mehrkörpersimulation (MKS)¹, mit der große Relativbewegungen vieler starrer Körper mithilfe kleiner Systeme abgebildet werden können [RS10, S.1]. Die Elastische Mehrkörpersimulation (EMKS) wurde entwickelt, um den Ansatz der MKS auch mit Körpern verwenden zu können, deren Deformierbarkeit nicht vernachlässigbar ist. Das Eigenschwingverhalten einzelner Körper spielt dann für das dynamische Verhalten der gesamten Struktur eine Rolle. Verfahren der Modellordnungsreduktion (MOR) wurden entwickelt, um die mithilfe FEM abgebildeten elastischen Körper auf eine kleine, von EMKS verwendbare Systemdimension zu reduzieren. Da herkömmliche Reduktionsverfahren die Anforderungen an reduzierte Systemdimension oder abzubildende Frequenzbereiche häufig nicht erfüllen, ist die Anwendung leistungstärkerer Verfahren auf die zugrundeliegenden mechanischen Systeme zweiter Ordnung notwendig.

Krylov-Unterraum-Methoden oder Krylov subspace methods (KSM), Methoden der Numerik, die für die Lösung verschiedenster numerischer Probleme genutzt werden können, gehören zu diesen Verfahren. Sie werden meist für Systeme erster Ordnung verwendet. Die Verfahrensvarianten der rationalen KSM erlauben nicht nur die Abbildung des Systemverhaltens im Bereich niedriger Eigenfrequenzen, sondern die freie Wahl eines abzubildenden Frequenzbereiches. Dies ermöglicht u.a. die Simulation spezieller akustischer Phänomene bei Reduktion auf kleine Systemdimensionen und spielt damit besonders im Leichtbau eine wichtige Rolle, wo effektiver Materialeinsatz oft unerwünschte Schwingungsphänomene nach sich zieht [LB11, S.127f].

¹In der Mechanik wird MKS überwiegend als Abkürzung für MehrKörperSysteme verwendet, hiervon weicht die vorliegende Arbeit ab.

Die vorliegende Arbeit behandelt aus der Gruppe der Krylov-Unterraum-Verfahren ein Arnoldi-Verfahren erster Ordnung und rationale SOAR-Verfahren, deren Implementierung und Anwendung auf verschiedene mechanische Systeme. Weiter wird ein geeignetes Fehlerbewertungskriterium entwickelt.

1.2. EINORDNUNG

Die Arbeit baut, KSM betreffend, hauptsächlich auf die in [Leh07] und [Feh11] vorgestellten Verfahren auf. Es handelt sich um Verfahren zweiter Ordnung, die aus denen erster Ordnung verschiedener Autoren, wie [Arn51] und [Gri97], entwickelt worden sind. Der grundlegende Forschungsstand der Professur für Dynamik und Mechanismentechnik der TU Dresden zu MOR geht auf [Kou09] zurück, wodurch eine einfache KSM-Funktion erster Ordnung zur Verfügung steht. Es wird das Modellordnungsreduktions-Paket (MORPACK) der Professur als Implementierungsumgebung genutzt.

Die Arbeit beschäftigt sich mit

- der zur Verfügung stehenden KSM-Funktion erster Ordnung,
- einem rationalen Block-SOAR-Verfahren und
- einem rationalen, interpolativen SOAR-Verfahren.

Die beiden letztgenannten Verfahren bieten die Möglichkeit der Verwendung mehrerer von Null verschiedener Frequenz-Entwicklungsstellen f_{ES} und die Einbeziehung der Dämpfung. Dämpfung als Teil des Reduktionsprozesses wird in dieser Arbeit nicht betrachtet.

Dies bildet die Grundlage der in [Feh11] vorgestellten iterativ rationalen SOAR-Verfahren, deren Behandlung den Umfang dieser Arbeit übersteigt.

Ein Teil der Arbeit nimmt die Behandlung der Basis-Operation der KSM, das numerische Lösen linearer Gleichungssysteme (LGS), ein. Hier werden die entsprechenden, in der Software MATLAB integrierten Funktionen und deren Kombinierbarkeit vorgestellt und darüber hinausgehende Möglichkeiten aufgezeigt.

Als Fehlerbewertungskriterium werden hauptsächlich das eigenfrequenzbezogene Normalized Relative Frequency Difference (NRFD)-Kriterium und die übertragungsfunktionsbezogenen Kriterien auf Grundlage der Frobenius-Norm $\|\mathbf{H}(s)\|_F$ und \mathcal{H}_2 -Norm $\|\mathbf{H}(s)\|_{\mathcal{H}_2}$ der Übertragungsfunktionsmatrix genutzt. Eine a priori Vorhersage der Reduktionsgüte ist dabei grundsätzlich nicht möglich. Fehlerkriterien, wie in [Feh11] genannt, die ohne Lösen des Eigenwertproblems z.B. durch Vergleich verschiedener Reduktionen das Reduktionsergebnis bewerten, werden nicht betrachtet.

1.3. AUFBAU DER ARBEIT

Kapitel 2 beschäftigt sich zunächst mit den theoretischen Grundlagen des Übertragungsverhaltens mechanischer Systeme, dem Arnoldi-Verfahren erster Ordnung, den rationalen SOAR-Verfahren und Fehlerbewertungskriterien, auch Korrelationskriterien genannt.

Es folgt in Kapitel 3 die mit Beispielen untermauerte theoretische Betrachtung linearer lineares Gleichungssystem (LGS) und deren Lösung innerhalb der Software MATLAB. Die Grenzen der Verfahren werden aufgezeigt und über MATLAB hinausgehende Möglichkeiten benannt.

Die Besonderheiten der implementierten KSM-Funktionen, Funktionen zur Lösung von LGS und Korrelationskriterien werden in Kapitel 4 erläutert, wobei sich einige Optionen der KSM-Funktionen bei den Tests in Kapitel 5 als nicht empfehlenswert erweisen.

In Kapitel 5 werden nach Vorstellung verschiedener Testmodelle die KSM-Funktionen validiert, verglichen und eine optimale Optionenwahl herausgearbeitet. Die an Modellen kleiner Dimension gemachten Erkenntnisse werden an Modellen großer Dimension validiert.

2. THEORIE

Es werden die Simulationsmethoden FEM und MKS und deren Kopplung erläutert. Es wird im Besonderen auf die Grundlagen und Algorithmen erster und zweiter Ordnung der KSM eingegangen. Korrelationskriterien zur Fehlerbewertung von Reduktionsergebnissen werden vorgestellt.

2.1. SIMULATIONSMETHODEN

Die hauptsächlich zur Simulation mechanischer Systeme verwendeten Methoden sind die Finite Elemente Methode (FEM) und die Mehrkörpersimulation (MKS). Beide sind aufgrund ihrer Eigenschaften für verschiedene Arten von Simulationen geeignet [LB12, S.547f]. Die grundlegende Herangehensweise beider Methoden wird im Folgenden kurz erläutert.

2.1.1. FINITE ELEMENTE METHODE

Die FEM ist eines der am häufigsten verwendeten Verfahren in der Strukturmechanik. Nach [LB12, S.548] sind zwar auch zunehmend stark nichtlineare Simulationen im Zeitbereich möglich, häufig gelten jedoch die Annahmen für lineare FEM

- der kleinen Verzerrungen und damit linearisierten Verschiebungs-Verzerrungs-Beziehung,
- des linear-elastischen Materialverhaltens und
- der Verwendung linearer Elemente und Kontaktbeschreibungen.

Bei der FEM findet eine Zerlegung der Struktur in diskrete Finite Elemente statt. Dazu wird ein Ritz-Ansatz für die Formfunktionen der Finiten Elemente und eine schwache Formulierung der Bilanzgleichungen verwendet. Das entstehende diskrete System hat die Struktur

$$\mathbf{M} \cdot \ddot{\mathbf{q}} + \mathbf{D} \cdot \dot{\mathbf{q}} + \mathbf{K} \cdot \mathbf{q} = \mathbf{f} \quad (2.1)$$

mit den quadratischen Systemmatrizen $[\mathbf{M}, \mathbf{D}, \mathbf{K}] \in \mathbb{R}^{N \times N}$ Massenmatrix, Dämpfungsmatrix und Steifigkeitsmatrix, dem Vektor der elastischen Koordinaten $\mathbf{q} \in \mathbb{R}^N$ sowie dem Vektor der äußeren Lasten $\mathbf{f} \in \mathbb{R}^N$. Dabei ist die direkte Bestimmung der Dämpfungsmatrix nicht möglich, welche aus diesem Grund in den meisten Fällen abgeschätzt wird.

Die Systemmatrizen sind generell nur dünn besetzt und haben eine Bandstruktur. Die Anzahl der Knotenfreiheitsgrade N kann abhängig von der Komplexität des Bauteils und der gewünschten Genauigkeit des Ergebnisses sehr groß sein, also $N = 10^3 \dots 10^7$. Dies führt zu potentiell langen Rechenzeiten bei großem Arbeitsspeicherbedarf. [LB12, S.548]

Für die hauptsächlichen Anwendungsgebiete der FEM sind die Behandlung von Festigkeitsproblemen und die Modalanalyse für Simulationen im Frequenzbereich. Für beide ist jeweils nur eine geringe Anzahl von Lösungsschritten notwendig. Die wesentlichen Merkmale der FEM sind in Tabelle 2.1 zusammengefasst.

2.1.2. MEHRKÖRPERSIMULATION

Im Gegensatz zur FEM werden bei der Mehrkörpersimulation (MKS) selten einzelne Bauteile betrachtet, sondern Bauteilgruppen, wobei die einzelnen Bauteile als starre Einzelkörper angenommen werden, die mit Gelenken kinematisch miteinander verbunden sind. Dieses Ersatzsystem ist äußeren Lasten ausgesetzt. Nach [RS10, S.2] ist in aktuellen Simulationsprogrammen zwar auch die Verwendung von elastischen Körpern möglich, dies führt aber zu stark erhöhten Rechenzeiten und ist lange der FEM vorbehalten gewesen. Die Einbindungsmöglichkeiten von Elastizitäten werden unter 2.1.3 erläutert.

Es werden bei der MKS große Auslenkungen betrachtet und Deformationen im Vergleich dazu als vernachlässigbar angenommen. Der Freiheitsgrad des entstehenden Ersatzsystems ist mit der Größenordnung $N = 10 \dots 1000$ vergleichsweise klein. In vielen Fällen lässt sich ein Mehrkörpermodell in der Form des Differentialgleichungssystems zweiter Ordnung

$$\mathbf{M}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (2.2)$$

darstellen, wobei $\mathbf{M} \in \mathbb{R}^{P \times P}$ für die quadratische Massenmatrix, $\mathbf{q} \in \mathbb{R}^P$ für den Vektor der generalisierten Koordinaten und $\mathbf{h} \in \mathbb{R}^P$ und $\mathbf{f} \in \mathbb{R}^P$ für die Vektoren der gyroskopischen Terme und der äußeren Lasten verwendet werden. Das Differentialgleichungssystem ist aufgrund der Berücksichtigung der Rotationen und großen Auslenkungen nichtlinear.

Trotz des kleinen Systemfreiheitsgrades entsteht erheblicher Aufwand bei der typischerweise durchgeführten numerischen Integration im Zeitbereich. Hier ist abhängig von der Simulationszeit und Schrittweite häufig eine Vielzahl von Rechenschritten notwendig. Die Erhöhung des Freiheitsgrades hat so auch einen wesentlichen Einfluss auf den Berechnungsaufwand. [LB12, S.549f]

2.1.3. KOPPLUNG DER SIMULATIONSMETHODEN

In Tabelle 2.1 sind die wesentlichen Unterschiede von Finite Elemente Methode (FEM) und Mehrkörpersimulation (MKS) zusammengefasst [LQK13, S.47].

Tabelle 2.1.: Unterscheidungsmerkmale Finite Elemente Methode (FEM) und Mehrkörpersimulation (MKS)

	Finite Elemente Methode	Mehrkörpersimulation
Art der Diskretisierung	Einteilung eines Kontinuums in Elemente	Einteilung einer Baugruppe in starre Massen und Gelenke
Systemfreiheitsgrad N	$10^3 \dots 10^7$	$< 10^3$
Verformungen	kleine Deformationen, lin.-elast. Material	große Auslenkungen, gyroskopische Anteile
Art des Differentialgleichungssystems	linear	nichtlinear
Systemlösbarkeit	direkt lösbar	nur iterativ lösbar
Anzahl der Lösungsschritte	wenige Lastfälle	viele Zeitschritte
Rechenzeit	groß für einen Lastschritt	klein für einen Zeitschritt
Anwendungsgebiet	kontinuierliche Systeme mit kleinen Deformationen	Mehrkörpersysteme aus starren Massen mit großen Auslenkungen
Analysemöglichkeiten	Festigkeitsanalyse, Modalanalyse	Massenkräfte bei Bewegungen

Werden in einer MKS elastische Deformationen berücksichtigt, so spricht man von einer Elastische Mehrkörpersimulation (EMKS). Nach [RS10, S.2] besteht in einigen Fällen die Möglichkeit, elastische Körper durch masselose Federn zwischen starren Ersatzmassen abzubilden.

Ist die Masse eines elastischen Körpers nicht vernachlässigbar, besteht die Möglichkeit, mithilfe einer Modellordnungsreduktion (MOR) den Freiheitsgrad eines durch FEM erzeugten Modells auf eine von EMKS verwendbare Dimension zu reduzieren. Gleichung 2.2 kann angewendet auf einen Einzelkörper für eine EMKS nach [LB12, S.549] als

$$\begin{bmatrix} \mathbf{M}_r & \mathbf{M}_{er} \\ \mathbf{M}_{re} & \mathbf{M}_e \end{bmatrix} \cdot \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{D}_e \cdot \dot{\mathbf{q}}_e + \mathbf{K}_e \cdot \mathbf{q}_e \end{bmatrix} + \begin{bmatrix} \mathbf{h}_r \\ \mathbf{h}_e \end{bmatrix} = \begin{bmatrix} \mathbf{f}_r \\ \mathbf{f}_e \end{bmatrix} \quad (2.3)$$

dargestellt werden. Dabei entsprechen die Matrix \mathbf{M}_r und die Vektoren $\ddot{\mathbf{q}}_r$, \mathbf{h}_r und \mathbf{f}_r den Größen aus Gleichung 2.2. Diese Differentialgleichung wird also lediglich um Gleichung 2.1 für ein reduziertes System und den Koppelterm \mathbf{M}_{er} erweitert. Dabei sind die (reduzierten) Systemmatrizen der elastischen Struktur \mathbf{M}_e , \mathbf{D}_e und \mathbf{K}_e im Gegensatz zu 2.1.1 in der Regel voll besetzt.

Ein elastisches Mehrkörpersystem beinhaltet starre und elastische Körper. Für die elastischen Strukturen gilt die Bedingung der linearen FEM der kleinen Deformationen weiter. Die Gesamtbewegung eines elastischen Körpers setzt sich so aus einer nichtlinearen, großen Starrkörperbewegung und einer linearisierten, lokale Deformation zusammen. [LB12, S.549]

Um den Aufwand für die EMKS bei guten Berechnungsergebnissen so gering wie möglich zu halten, besteht die wesentliche Anforderung an eine MOR, das Systemverhalten der FE-Struktur der Dimension N mit kleinstmöglicher reduzierter Systemdimension n bestmöglich abzubilden.

2.2. ZUSTANDSRAUMDARSTELLUNG UND REDUKTION

Auf mittels FEM erstellte, mechanische Systeme werden Reduktionsmethoden angewendet, um deren Systemgröße zu reduzieren, sodass sie in einer EMKS verwendet werden können. Die Bewegungsgleichung eines Systems zweiter Ordnung kann dabei nach [Leh07, S.101] folgendermaßen dargestellt werden:

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

Dabei bezeichnen $[\mathbf{M}, \mathbf{D}, \mathbf{K}] \in \mathbb{R}^{N \times N}$ die Systemmatrizen Massenmatrix, Dämpfungsmatrix und Steifigkeitsmatrix und die Matrizen $\mathbf{B} \in \mathbb{R}^{N \times e}$ und $\mathbf{C} \in \mathbb{R}^{a \times N}$ die Eingangs- bzw. Ausgangsmatrix. Für Verwendung gleicher Ein- und Ausgänge gilt $e = a = r$.

Für die Reduktion mechanischer Systeme sind Projektionsmatrizen von zentraler Bedeutung. In dieser Arbeit wird die Darstellung der Projektionsmatrix als $\mathbf{V} \in \mathbb{R}^{N \times n}$ verwendet. Nach [Leh07] werden die Projektionsmatrizen für schiefe Projektion häufig als \mathbf{V} und \mathbf{W} dargestellt. In dieser Arbeit wird generell von orthogonaler Projektion ausgegangen, womit nach [Leh07, S.134] der Zusammenhang

$$\mathbf{W} = \mathbf{V} \quad (2.5)$$

und für die reduzierten Matrizen $[\bar{\mathbf{M}}, \bar{\mathbf{D}}, \bar{\mathbf{K}}] \in \mathbb{R}^{n \times n}$, $\bar{\mathbf{B}} \in \mathbb{R}^{n \times r}$ und $\bar{\mathbf{C}} \in \mathbb{R}^{r \times n}$ die Zusammenhänge

$$\begin{aligned} \bar{\mathbf{M}} &= \mathbf{V}^T \cdot \mathbf{M} \cdot \mathbf{V}; & \bar{\mathbf{D}} &= \mathbf{V}^T \cdot \mathbf{D} \cdot \mathbf{V}; & \bar{\mathbf{K}} &= \mathbf{V}^T \cdot \mathbf{K} \cdot \mathbf{V}; \\ \bar{\mathbf{B}} &= \mathbf{V}^T \cdot \mathbf{B}; & \bar{\mathbf{C}} &= \mathbf{C} \cdot \mathbf{V} \end{aligned} \quad (2.6)$$

gelten. Der Zusammenhang zwischen den Koordinaten des Originalsystems $\mathbf{q} \in \mathbb{R}^N$ und denen des reduzierten Systems $\bar{\mathbf{q}} \in \mathbb{R}^n$ kann als

$$\mathbf{q} = \mathbf{V} \cdot \bar{\mathbf{q}} \quad (2.7)$$

dargestellt werden.

In [LB12, S.552] wird eine nach [Kou09, S.21-22] weiterentwickelte Kategorisierung der Reduktionsmethoden vorgeschlagen:

- Anschauungsraumbezogene Verfahren (reduzierte Koordinaten $\bar{\mathbf{q}}$ im Anschauungsraum interpretierbar)
- Verfahren im modalen Raum (reduzierte Koordinaten $\bar{\mathbf{q}}$ im modalen Raum interpretierbar)

- Verfahren im hybriden Raum (ein Teil der reduzierten Koordinaten $\bar{\mathbf{q}}$ im Anschauungsraum und anderer Teil im modalen Raum oder allgemeinen Vektorraum interpretierbar)
- Verfahren im allgemeinen Vektorraum (reduzierte Koordinaten $\bar{\mathbf{q}}$ in nicht näher spezifiziertem Vektorraum interpretierbar)

Die in dieser Arbeit betrachteten KSM gehören zur Gruppe der Verfahren im allgemeinen Vektorraum und sind zuerst von [Kry31] für Systeme erster Ordnung entwickelt worden.

Das System erster Ordnung entsteht durch Einführung des Koordinatenvektors $\mathbf{x}(t) = [\mathbf{q}(t) \quad \dot{\mathbf{q}}(t)]^T$, wodurch sich Gleichung (2.4) als das Gleichungssystem

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix}}_{\mathbf{E}} \cdot \underbrace{\begin{bmatrix} \dot{\mathbf{q}}(t) \\ \ddot{\mathbf{q}}(t) \end{bmatrix}}_{\dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K} & -\mathbf{D} \end{bmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{B} \end{bmatrix}}_{\mathbf{B}_f} \cdot \mathbf{u}(t) \quad (2.8)$$

$$\mathbf{y}(t) = \underbrace{\begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix}}_{\mathbf{C}_f} \cdot \underbrace{\begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{bmatrix}}_{\mathbf{x}(t)}$$

mit $[\mathbf{E}, \mathbf{A}] \in \mathbb{R}^{2N \times 2N}$, $\mathbf{B}_f \in \mathbb{R}^{2N \times r}$, $\mathbf{C}_f \in \mathbb{R}^{r \times 2N}$ und $[\dot{\mathbf{x}}, \mathbf{x}] \in \mathbb{R}^{2N}$ darstellen lässt.

2.3. KRYLOV UNTERRAUM METHODEN

Nach [Kou09, S.40] beruhen KSM auf der Approximation der Übertragungsfunktion eines Systems durch Reihenentwicklung. Die Übertragungsfunktionsmatrix lautet für Systeme zweiter Ordnung

$$\mathbf{H}(s) = \mathbf{C} \cdot (s^2\mathbf{M} + s\mathbf{D} + \mathbf{K})^{-1} \cdot \mathbf{B} \quad (2.9)$$

und für Systeme erster Ordnung

$$\mathbf{H}(s) = \mathbf{C}_f \cdot (s\mathbf{E} - \mathbf{A})^{-1} \cdot \mathbf{B}_f \quad (2.10)$$

Die von [Kry31] entwickelten KSM sind von verschiedenen Autoren weiterentwickelt worden. Dabei werden in der Literatur, z.B. bei [Gri97] und [GVD05], verschiedene Verfahrenstypen generell für Systeme erster Ordnung entwickelt. KSM können auf viele Systemarten, z.B. in der Elektro- und Fluidtechnik, aber auch zum iterativen Lösen linearer Gleichungssysteme und des Eigenwertproblems genutzt werden. Mechanische Systeme zweiter Ordnung sind also nur eine Einsatzmöglichkeit der KSM unter vielen. Möglichkeiten der Verfahrensanwendung auf mechanische Systeme zweiter Ordnung sind u.a. in [CTL09] und [Leh07] angegeben, wobei die vorliegende Arbeit großteils auf den Algorithmen nach [Leh07] aufbaut. Die dort dokumentierten Verfahren sind, wie unter 2.4 gezeigt, allgemein gehalten und implizieren so verschiedene Verfahrenstypen der KSM.

Tabelle 2.2.: Klassifizierungskriterien für Krylov-Unterraummethoden

Kriterium	Möglichkeiten	Beschreibung
Berechnungsverfahren	Arnoldi	einfach implementierbar, numerisch sehr stabil
	Lanczos	komplexer, numerisch instabiler als Arnoldi
Entwicklungsstellen s_{ES} der Reihenentwicklung	original Krylov	eine Entwicklungsstelle $s_{ES} = 0$
	rational Krylov	eine oder mehrere Entwicklungsstellen $s_{ES} \neq 0$
	iterativ rational Krylov	iterative Wahl der Entwicklungsstellen $s_{ES} \neq 0$
Ein- und Ausgangszahl	Single-Input-Single-Output (SISO)-System	ein Eingang bzw. Ausgang
	Multi-Input-Multi-Output (MIMO)-System	mehrere Eingänge bzw. Ausgänge
Umgang mit MIMO-Systemen	Block-Krylov	ein Startvektor je Ein- bzw. Ausgang, jeder Berechnungsschritt im Block mit einer Krylov-mode je Ein- bzw. Ausgang
	Interpolation	Interpolation der Ein- und Ausgänge auf Startvektor, in jedem Berechnungsschritt nur eine Krylov-Mode
	Nichtbeachtung	Startvektor unabhängig von Ein- bzw. Ausgängen, in jedem Berechnungsschritt nur eine Krylov-Mode
Projektionsart	orthogonal	Eingänge gleich Ausgänge $\mathbf{B} = \mathbf{C}^T$, nur eine Projektionsmatrix
	schief	Eingänge ungleich Ausgänge $\mathbf{B} \neq \mathbf{C}^T$, zwei Projektionsmatrix

Grundsätzlich lassen sich die Verfahrenstypen der KSM auf Grundlage der Gleichungen 2.9 und 2.10 nach verschiedenen Kriterien klassifizieren. Die folgende Klassifizierung hat nicht Vollständigkeit, sondern die Erläuterung von Grundprinzipien zum Ziel. Als Berechnungsverfahren zum Aufstellen von Krylov-Unterräumen eignen sich laut [Kou09, S.5] das Arnoldi-Verfahren nach [Arn51] und das Lanczos-Verfahren nach [Lan50]. Das Arnoldi-Verfahren wird als numerisch stabiler charakterisiert. Wird die Reihenentwicklung nicht um die Entwicklungsstelle $s_{ES} = 0$, sondern um eine oder mehrere Entwicklungsstellen $s_{ES_k} \neq 0$ durchgeführt, so spricht man von rationalen Krylov-Verfahren (nach [Gri97]). Werden diese Entwicklungsstellen der Reihenentwicklung s_{ES_k} iterativ gewählt, wird von iterativ rationalen Krylov-Verfahren gesprochen. In [Feh11] werden Möglichkeiten dieser iterativen Wahl genannt. Weiter lassen sich Verfahren nach Art der Ein- und Ausgangsmatrix \mathbf{B} und \mathbf{C} einteilen. Gibt es jeweils nur einen Ein- und Ausgang, so spricht man von Single-Input-Single-Output (SISO)-Systemen. Die Ein- und Ausgangsvektoren \mathbf{b} und \mathbf{c} haben dann nur einen von Null verschiedenen Eintrag. Andernfalls spricht man von Multi-Input-Multi-Output (MIMO)-Systemen. Mechanische Systeme zweiter Ordnung sind in der Regel MIMO-Systeme. Werden die Ein- bzw. Ausgangsmatrizen als Startmatrizen zur blockweisen Erstellung von Krylov-Moden verwendet, so wird die Bezeichnung Block-Verfahren verwendet. Es ist für MIMO-Systeme aber auch möglich, Ein- und Ausgangsvektoren \mathbf{b} und \mathbf{c} als Startvektoren aus den Matrizen \mathbf{B} und \mathbf{C} zu interpolieren. Ein solches interpolatives Vorgehen wird in [BGKVW07] für den MIMO Iterative Rational

Interpolation Algorithm (MIRIAM) vorgeschlagen. In [Kou09] wird für das MIMO-System ebenfalls nur ein Startvektor gewählt, dieser ist jedoch unabhängig von den Ein- und Ausgängen des Systems. Weiter kann in orthogonale und schiefe Projektion unterschieden werden. Für orthogonale Projektion besteht Gleichheit der Ein- und Ausgangsmatrix $\mathbf{B} = \mathbf{C}^T$, wodurch nur eine Projektionsmatrix \mathbf{V} berechnet werden muss. Dagegen folgen aus $\mathbf{B} \neq \mathbf{C}^T$ zwei Projektionsmatrizen \mathbf{V} und \mathbf{W} für die schiefe Projektion. Die verschiedenen Klassifizierungskriterien sind grundsätzlich unabhängig voneinander und in Tabelle 2.2 zusammengefasst.

In der vorliegenden Arbeit werden aufgrund ihrer numerischen Stabilität Arnoldi-Algorithmen verwendet. Es wird die nichtiterative Entwicklungsstellenwahl untersucht. Wie erwähnt, werden grundsätzlich MIMO-Systeme betrachtet, wobei die verschiedenen Arten des Umgangs mit MIMO-Systemen untersucht werden. Es wird lediglich orthogonale Projektion betrachtet, da in dieser Arbeit alle Systemeingänge auch -ausgänge sind. Nach [Feh11, S.55] ist dieses Vorgehen für mechanische Systeme gängige Praxis. Iterativ rationale Verfahren werden nicht umgesetzt.

2.4. ARNOLDI-ALGORITHMEN ERSTER ORDNUNG

Arnoldi-Algorithmen stellen nach [Leh07, S.124] eine einfach implementierbare und numerisch stabile Art und Weise dar, orthogonale Basen von Krylov-Unterräumen zu erzeugen. Rationale Verfahren werden z.B. in [Gri97] für Systeme erster Ordnung entwickelt und können für Systeme zweiter Ordnung abgewandelt werden.

Dieser Abschnitt beschäftigt sich zuerst mit einem allgemeinen, dann mit einem speziellen Arnoldi-Algorithmus erster Ordnung. Es wird zuerst ein rationaler Block-Arnoldi-Algorithmus vorgestellt. Daraufhin wird der für diese Arbeit in implementierter Form zur Verfügung stehende, einfache Arnoldi-Algorithmus eingeordnet. Bei der Vorstellung der Möglichkeiten rationaler Block-Arnoldi-Verfahren wird nicht auf iterative Verfahren eingegangen. Die Alternativen zur Verwendung von Block-Verfahren werden hauptsächlich unter 2.5 dargestellt.

Ein rationales Block-Arnoldi-Verfahren ist in Algorithmus 2.1 unter Verwendung der Gram-Schmidt-Orthogonalisierung aus Algorithmus 2.2 angegeben. Das Verfahren ist an [Leh07, S.125-127] angelehnt. Der Algorithmus aus [Leh07] ist dabei vereinfacht worden, um Grundprinzipien besser darzustellen. So entsteht hier u.a. lediglich eine Projektionsmatrix \mathbf{V} , da schiefe Projektion mit zwei Projektionsmatrizen ausgeschlossen wird. Es gilt für die Ein- und Ausgangsmatrizen $\mathbf{B}_f = \mathbf{C}_f^T$, wodurch die Eingangsmatrix \mathbf{B}_f als Eingangsgröße ausreicht.

Bei den Eingabe-Matrizen handelt es sich um die Matrizen aus Gleichung 2.8. Die Deflationstoleranz $deftol$ und Gewichtungsmatrix \mathbf{Q} werden für die Orthogonalisierung benötigt und im Zuge deren Behandlung erläutert. Der Algorithmus ist so angelegt, dass für jeden

Algorithmus 2.1 : rational Block-Arnoldi-Verfahren

Eingabe : Systemmatrizen \mathbf{E} , \mathbf{A} , \mathbf{B}_f , Vektor der Entwicklungsstellen \mathbf{s}_{ES} , Vektor der Reihenentwicklungsordnungen \mathbf{j} ; für Orthogonalisierung: Deflationstoleranz $deftol$, Gewichtungsmatrix \mathbf{Q}

Ausgabe : Basis \mathbf{V}

```

function [V] = RatBlockArnoldi(E, A, Bf, sES, j, deftol, Q)
1 for k = 1 to length(sES) do                                     % für alle Entwicklungspunkte
2     A_inv = (A - sES_k * E)^-1
3     for l = 1 to j_k do                                       % über Ordnung der Reihenentwicklung
4         if l = 1 then
5             R = A_inv * Bf
6         else
7             R = A_inv * E * R
8         end
9         R_bar = Orthogonalisierung(R, V, deftol, Q)
10        V = [V R_bar];                                       % Neue Basis zur alten Basis hinzufügen
11    end
12 end

```

Entwicklungspunkt s_{ES_k} die Ordnung der Reihenentwicklung j_k variiert werden kann. Mit Worten ausgedrückt, kann der Ablauf des Algorithmus so dargestellt werden:

Für jeden Entwicklungspunkt und für jede Ordnung der Reihenentwicklung am jeweiligen Entwicklungspunkt

- werden so viele Vektoren erzeugt, wie Ein-/Ausgänge verwendet werden,
- werden die Vektoren orthogonalisiert
- und der Basis aus zuvor erzeugten Vektoren zugefügt.

Die Spaltenzahl der Projektionsmatrix \mathbf{V} , die bei Nichtauftreten von Deflation erreicht wird, ergibt sich bei diesem Block-Verfahren zu

$$\text{size}(\mathbf{V}, 2) = \underbrace{\text{size}(\mathbf{B}_f, 2)}_{\text{Anzahl der Ein-/Ausgänge}} \cdot \underbrace{\sum_k j_k}_{\text{Summe der Ordnungen der Reihenentwicklung der } k \text{ Entwicklungsstellen}} \quad \text{mit} \quad \text{length}(\mathbf{j}) = \underbrace{\text{length}(\mathbf{s}_{ES})}_{\text{Anzahl der Entwicklungsstellen}}. \quad (2.11)$$

Der Orthogonalisierungs-Algorithmus 2.2 wird auch Gram-Schmidt-Orthogonalisierung genannt. Er hat zur Aufgabe, jeden einzelnen Vektor der Matrix \mathbf{R} gegenüber der Basis \mathbf{V} (Zeilen 2-4) und gegenüber jedem schon orthogonalisierten Vektor der Matrix \mathbf{R} (Zeilen 5-7) zu orthogonalisieren und zu normieren. Dabei ist es möglich, die einzelnen Zeilen der Basis über die Gewichtungsmatrix \mathbf{Q} zu wichten. Bei der Gram-Schmidt-Orthogonalisierung wird das gewichtete Skalarprodukt h aus zwei Vektoren gebildet (Zeile 3 und 6) und h mal dem unveränderten Vektor vom zu orthogonalisierenden Vektor abgezogen (Zeile 4 und 7). Folglich ist das gewichtete Skalarprodukt aus dem neuen Vektor und dem unveränderten Vektor gleich Null, die beiden Vektoren sind orthogonal. Das grundsätzliche Prinzip der

Algorithmus 2.2 : Gram-Schmidt-Verfahren

Eingabe : Matrix \mathbf{R} , Basis \mathbf{V} , Deflationstoleranz *deftol*, Gewichtungsmatrix \mathbf{Q}
Ausgabe : Basis $\bar{\mathbf{R}}$

```

function [ $\bar{\mathbf{R}}$ ] = Orthogonalisierung( $\mathbf{R}$ ,  $\mathbf{V}$ , deftol,  $\mathbf{Q}$ )
1 for  $i = 1$  to columns( $\mathbf{R}$ ) do
2   for  $k = 1$  to columns( $\mathbf{V}$ ) do
3      $h = \mathbf{v}_k^T \cdot \mathbf{Q} \cdot \mathbf{r}_i$ ;           % Energieskalarprodukt
4      $\mathbf{r}_i = \mathbf{r}_i - h \cdot \mathbf{v}_k$ 
5   end
6   for  $k = 1$  to columns( $\bar{\mathbf{R}}$ ) do
7      $h = \bar{\mathbf{r}}_k^T \cdot \mathbf{Q} \cdot \mathbf{r}_i$ ;   % Energieskalarprodukt
8      $\mathbf{r}_i = \mathbf{r}_i - h \cdot \bar{\mathbf{r}}_k$ 
9   end
10   $h = \sqrt{\mathbf{r}_i^T \cdot \mathbf{Q} \cdot \mathbf{r}_i}$ ;      % Energienorm
11  if  $h > \textit{deftol}$  then
12     $\mathbf{r}_i = \frac{1}{h} \cdot \mathbf{r}_i$ 
13     $\bar{\mathbf{R}} = [\bar{\mathbf{R}} \ \mathbf{r}_i]$ ;           % Vektor hinzufügen
14  else
15    Deflation
16  end
end

```

Orthogonalisierung wird in Abbildung 2.1 gezeigt. Dabei wird das einfache Skalarprodukt zweier Vektoren verwendet, was einer Gleichsetzung der Gewichtungsmatrix \mathbf{Q} mit der Einheitsmatrix entspricht.

Ist ein zu orthogonalisierender Vektor \mathbf{r}_i von der Basis \mathbf{V} und $\bar{\mathbf{R}}$ linear abhängig, so ergibt sich nach der Orthogonalisierung der Betrag des Vektors (in Zeile 8) zu $h = 0$. Ist der normierte Vektor schon vor der Orthogonalisierung senkrecht zu allen Basisvektoren gewesen, so ergibt sich nach dieser keine Betragsänderung von \mathbf{r}_i . Damit ist der Betrag h , der in Zeile 8 berechnet wird, auch ein Maß für die Neuheit der Richtungsinformation eines Vektors. Mit der Deflationstoleranz ist es somit möglich, ein Maß an Neuheit festzulegen, das ein berechneter Vektor beinhalten muss, um Teil der Projektionsmatrix zu werden.

Der Orthogonalisierungsalgorithmus in [Leh07, S.127] enthält zusätzlich die Möglichkeit der Reorthogonalisierung und der Verwendung einer initialen Basis. Diese Möglichkeiten werden für Systeme zweiter Ordnung in Algorithmus 2.5 berücksichtigt.

Der im MORPACK enthaltene KSM-Algorithmus wird vom Erstentwickler der Software in [Kou09, S.43] beschrieben und geht auf [Bai05, S.5-6] zurück. Die dort verwendete Methode wird in Algorithmus 2.3 beschrieben. Folgende Vereinfachungen sind zu treffen, um aus dem rationalen Block-Arnoldi-Verfahren in den Algorithmen 2.1 und 2.2 diesen vereinfachten Arnoldi-Algorithmus zu erhalten:

- Reihenentwicklung um eine Entwicklungsstelle $s_{ES} = 0$
- Ersetzen der Eingangsmatrix \mathbf{B}_f durch nicht näher definierten Startvektor \mathbf{b} (Vernachlässigung der MIMO-Systemeigenschaft)

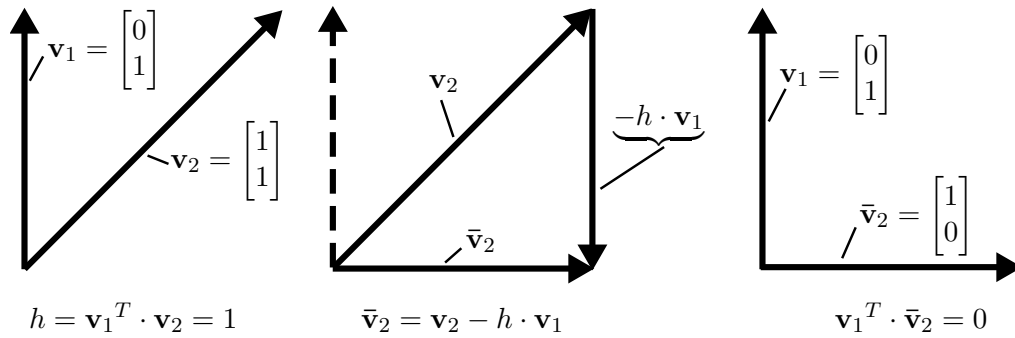


Abbildung 2.1.: Orthogonalisierung anhand zweier Vektoren im zweidimensionalen Raum

- Festlegen der Deflationstoleranz auf $deftol = 0$
- Ersetzen der Gewichtungsmatrix \mathbf{Q} durch die Einheitsmatrix, wodurch das Energiekalarprodukt zu einem einfachen Skalarprodukt wird

Im implementierten Algorithmus wird das Produkt $\mathbf{A}^{-1}\mathbf{E} = \mathbf{K}^{-1}\mathbf{M}$ gesetzt, was einem Quadranten des Produktes $\mathbf{A}^{-1}\mathbf{E}$ entspricht. Damit ergibt sich folgende weitere Einschränkung:

- Keine Berücksichtigung der Dämpfung

Die Spaltenanzahl der Projektionsmatrix \mathbf{V} ist damit gleich der Ordnung der Reihenentwicklung

$$\text{columns}(\mathbf{V}) = j \quad . \quad (2.12)$$

Algorithmus 2.3 : Arnoldi-Verfahren

Eingabe : Systemmatrizen \mathbf{E} , \mathbf{A} , Startvektor \mathbf{b} , Ordnung der Reihenentwicklung j

Ausgabe : Basis \mathbf{V}

```

function [V] = Arnoldi(E, A, b, j)
1 v1 = -A^-1 * b
2 v1 = v1 / ||v1||2 ; % Normierung
3 for k = 1 to j do % über Ordnung der Reihenentwicklung
4     r = A^-1 * E * vk
5     for l = 1 to j do % Orthogonalisierung
6         h = vl^T * r
7         r = r - h * vl
8     end
9     h = ||r||2 ; % Normierung
10    if h = 0 then
11        stop
12    else
13        vk+1 = r/h
14    end
end
end
    
```

2.5. ARNOLDI-ALGORITHMEN ZWEITER ORDNUNG

Arnoldi-Verfahren zweiter Ordnung verwenden die Systemmatrizen der Gleichung 2.4 und sind Umformulierungen der Arnoldi-Verfahren erster Ordnung. Sie werden auch Second Order Arnoldi (SOAR)-Verfahren genannt. Ein rationales Block-SOAR-Verfahren ist inklusive Orthogonalisierungs-Verfahren in den Algorithmen 2.4 und 2.5 angegeben. Das Verfahren ist an [Leh07, S.144-146] angelehnt und für orthogonale Projektion abgewandelt.

Algorithmus 2.4 : Rationales Block-SOAR-Verfahren

Eingabe : Systemmatrizen \mathbf{M} , \mathbf{D} , \mathbf{K} , \mathbf{B} , Vektor der Entwicklungsstellen \mathbf{s}_{ES} , Vektor \mathbf{j} , Deflationstoleranz $deftol$, Gewichtungsmatrix \mathbf{Q} , initiale Basis \mathbf{G}

Ausgabe : Basis \mathbf{V}

```

function [V] = RatBlockSOAR(M,D,K,B,sES,j,deftol,Q,G)
1 for k = 1 to length(sES) do                                     % für alle Entwicklungspunkte
2      $\tilde{\mathbf{K}}_{inv} = (s_{ES_k}^2 \mathbf{M} + s_{ES_k} \mathbf{D} + \mathbf{K})^{-1}$ 
3      $\tilde{\mathbf{D}} = 2 \cdot s_{ES_k} \mathbf{M} + \mathbf{D}$ 
4     for l = 1 to j_k do                                       % über Ordnung der Reihenentwicklung
5         if l = 1 then
6              $\tilde{\mathbf{V}} = -\tilde{\mathbf{K}}_{inv} \cdot \mathbf{B}$ 
7              $\tilde{\mathbf{U}} = 0$ 
8         else if l = 2 then
9              $\tilde{\mathbf{U}} = \tilde{\mathbf{V}}$ 
10             $\tilde{\mathbf{V}} = -\tilde{\mathbf{K}}_{inv} \cdot \tilde{\mathbf{D}} \cdot \tilde{\mathbf{V}}$ 
11        else
12             $\tilde{\mathbf{V}}_{alt} = \tilde{\mathbf{V}}$ 
13             $\tilde{\mathbf{V}} = -\tilde{\mathbf{K}}_{inv} \cdot \tilde{\mathbf{D}} \cdot \tilde{\mathbf{V}} - \tilde{\mathbf{K}}_{inv} \cdot \mathbf{M} \cdot \tilde{\mathbf{U}}$ 
14             $\tilde{\mathbf{U}} = \tilde{\mathbf{V}}_{alt}$ 
15        end
16         $[\tilde{\mathbf{V}} \ \tilde{\mathbf{U}}] = \text{OrthogonalisierungSOAR}(\tilde{\mathbf{V}}, \tilde{\mathbf{U}}, \hat{\mathbf{V}}, \hat{\mathbf{U}}, \mathbf{V}, deftol, \mathbf{Q}, \mathbf{G})$ 
17         $\hat{\mathbf{V}} = [\hat{\mathbf{V}} \ \tilde{\mathbf{V}}]$ 
18         $\hat{\mathbf{U}} = [\hat{\mathbf{U}} \ \tilde{\mathbf{U}}]$ 
19         $\tilde{\mathbf{V}} = \hat{\mathbf{V}}$ 
20         $\tilde{\mathbf{U}} = \hat{\mathbf{U}}$ 
21    end
22     $\mathbf{V} = [\mathbf{V} \ \hat{\mathbf{V}}]$ 
end

```

Der wesentliche Unterschied zwischen der Beschreibung zweiter Ordnung (z.B. Algorithmus 2.4) und der Beschreibung erster Ordnung (z.B. Algorithmus 2.1) besteht nach [Leh07, S.145] darin, dass im Gegensatz zum Verfahren erster Ordnung nicht die komplette Basis $\tilde{\mathbf{R}} = [\hat{\mathbf{V}}^T \ \hat{\mathbf{U}}^T]^T$ die Orthogonalitätsbedingung erfüllt, sondern lediglich die obere Hälfte $\hat{\mathbf{V}}$ bzw. die entstehende Projektionsmatrix \mathbf{V} . Dies wird aus der Berechnungsvorschrift für \mathbf{U} aus den Zeilen 11 und 14 des Algorithmus 2.5 deutlich.

Ein Block-SOAR-Verfahren kann aus dem rationalen Block-SOAR-Verfahren abgeleitet werden, indem nur ein Entwicklungspunkt $s_{ES} = 0$ verwendet wird. Wird zudem die Eingangsmatrix \mathbf{B} durch einen Vektor ersetzt und die Dämpfung vernachlässigt, so erhält

man einen einfachen Arnoldi-Algorithmus. Es kann gezeigt werden, dass dieser dem unter 2.4 erwähnten und in MORPACK implementierten Algorithmus entspricht.

Nach [Leh07, S.125] hängt die numerische Stabilität der Arnoldi-Verfahren maßgeblich von der Wahl des Orthogonalisierungsverfahrens ab. In [Saa03, S.11] wird das Modifizierte Gram-Schmidt-Verfahren vorgestellt, das die Grundlage des Algorithmus 2.5 (an [Leh07] angelehnt) darstellt. Hier wird mit dem ganzzahligen Parameter $NumReOrth$ Reorthogonalisierung ermöglicht, um Rundungsfehler zu minimieren. Dafür soll im Allgemeinen ein Reorthogonalisierungsschritt, also $NumReOrth = 2$, ausreichen.

Im Rahmen dieser Arbeit wird für das Energieskalarprodukt und die Energienorm als Gewichtungsmatrix die Massenmatrix $\mathbf{Q} = \mathbf{M}$ verwendet.

In [Leh07, S.135] wird die Notwendigkeit für ungefesselte Systeme bewiesen, eine Basis zu erzeugen, die orthogonal zu den Starrkörpermoden des Systems ist. Dies ist möglich, indem die Starrkörpermoden, im Normalfall¹ $\Phi_{rigid} \in \mathbb{R}^{N \times 6}$, als initiale Basis $\mathbf{G} = \Phi_{rigid}$ im Orthogonalisierungs-Prozess berücksichtigt werden, ohne sie der Projektionsmatrix \mathbf{V} beizufügen. Wie in Abschnitt 2.1.3 erläutert, können für das reduzierte System kleine Deformationen vorausgesetzt werden. So ist es unter kleinem Rechenaufwand möglich, nicht nur translatorische, sondern auch rotatorische Starrkörpermoden aus den Knotenkoordinaten des vollen Modells zu erzeugen.

Analog Algorithmus 2.2 ist hier die Deflationstoleranz $deftol$ von Bedeutung. Wie unter 2.4 erläutert, darf die Energienorm eines orthogonalisierten Vektors die Deflationstoleranz nicht unterschreiten. In dem Fall wäre der Vektor in zu hohem Maße von der schon berechneten Basis abhängig. Bei SOAR darf im Gegensatz zu Arnoldi erster Ordnung nicht nur der Deflationsparameter h für Deflation betrachtet werden. Für ein ungedämpftes System mit der Entwicklungsstelle $s_{ES} = 0$ entsteht in Algorithmus 2.4 in Zeile 10 die Nullmatrix $\tilde{\mathbf{V}} = \mathbf{0}$. Damit die Reihenentwicklung in diesem Fall nicht abgebrochen wird, wird zusätzlich in Algorithmus 2.5 Zeile 22 die Norm des Vektors $\tilde{\mathbf{u}}_i$ überprüft.

Im Gegensatz zu [Leh07] werden die Vektoren hier vor dem Orthogonalisierungsprozess normiert, um eine bessere Vergleichbarkeit des Deflationsparameters zu gewährleisten. Weiter werden die Entwicklungsstellen $s_{ES_k} = i \cdot f_{ES_k}$ und damit $s_{ES_k}^2 = -f_{ES_k}^2$ verwendet. In Algorithmus 2.4 auf Zeile 3 liefern Versuche mit $s_{ES_k} = i \cdot f_{ES_k}$ keine korrelierenden Reduktionsergebnisse. Setzt man dort von der Definition abweichend $s_{ES_k} = f_{ES_k}$, erhält man die unter Abschnitt 5 erläuterten Versuchsergebnisse. Der Einfluss dieser Modifikation ist zu untersuchen.

Wie unter 2.1 erläutert, besteht ein wesentliches Ziel darin, eine kleinstmögliche Systemdimension des reduzierten Systems n bei guter Systemapproximation im entsprechenden Frequenzbereich zu erzielen. Für rationale Block-SOAR gilt wie für Verfahren erster Ordnung für die Systemdimension des reduzierten Systems Gleichung 2.11. Die Dimension des reduzierten Systems (entspricht der Spaltenzahl der Projektionsmatrix \mathbf{V}) hängt demnach

¹Dies gilt für sechs auftretende Starrkörpermoden.

Algorithmus 2.5 : Modifiziertes Gram-Schmidt-Verfahren für SOAR

Eingabe : Matrizen $\tilde{\mathbf{V}}, \tilde{\mathbf{U}}$, Basen $\hat{\mathbf{V}}, \hat{\mathbf{U}}, \mathbf{V}$, Deflationstoleranz *deftol*, Gewichtungsmatrix \mathbf{Q} ,
 initiale Basis \mathbf{G}

Ausgabe : Basen $\bar{\mathbf{V}}, \bar{\mathbf{U}}$

```

function [ $\bar{\mathbf{V}}$   $\bar{\mathbf{U}}$ ] = OrthogonalisierungSOAR( $\tilde{\mathbf{V}}, \tilde{\mathbf{U}}, \hat{\mathbf{V}}, \hat{\mathbf{U}}, \mathbf{V}, deftol, \mathbf{Q}, \mathbf{G}$ )
1 for  $i = 1$  to columns( $\tilde{\mathbf{V}}$ ) do
2    $\tilde{\mathbf{v}}_i = \tilde{\mathbf{v}}_i / \sqrt{\tilde{\mathbf{v}}_i^T \cdot \mathbf{Q} \cdot \tilde{\mathbf{v}}_i}$ 
3    $\tilde{\mathbf{u}}_i = \tilde{\mathbf{u}}_i / \sqrt{\tilde{\mathbf{u}}_i^T \cdot \mathbf{Q} \cdot \tilde{\mathbf{u}}_i}$ 
4   for  $r = 1$  to NumReorth do
5     for  $k = 1$  to columns( $\mathbf{G}$ ) do
6        $h = \mathbf{g}_k^T \cdot \mathbf{Q} \cdot \tilde{\mathbf{v}}_i$  ; % Energieskalarprodukt
7        $\tilde{\mathbf{v}}_i = \tilde{\mathbf{v}}_i - h \cdot \mathbf{g}_k$ 
8     end
9     for  $k = 1$  to columns( $\mathbf{V}$ ) do
10       $h = \mathbf{v}_k^T \cdot \mathbf{Q} \cdot \tilde{\mathbf{v}}_i$  ; % Energieskalarprodukt
11       $\tilde{\mathbf{v}}_i = \tilde{\mathbf{v}}_i - h \cdot \mathbf{v}_k$ 
12    end
13    for  $k = 1$  to columns( $\hat{\mathbf{V}}$ ) do
14       $h = \hat{\mathbf{v}}_k^T \cdot \mathbf{Q} \cdot \tilde{\mathbf{v}}_i$  ; % Energieskalarprodukt
15       $\tilde{\mathbf{v}}_i = \tilde{\mathbf{v}}_i - h \cdot \hat{\mathbf{v}}_k$ 
16       $\tilde{\mathbf{u}}_i = \tilde{\mathbf{u}}_i - h \cdot \hat{\mathbf{u}}_k$ 
17    end
18    for  $k = 1$  to columns( $\bar{\mathbf{V}}$ ) do
19       $h = \bar{\mathbf{v}}_k^T \cdot \mathbf{Q} \cdot \tilde{\mathbf{v}}_i$  ; % Energieskalarprodukt
20       $\tilde{\mathbf{v}}_i = \tilde{\mathbf{v}}_i - h \cdot \bar{\mathbf{v}}_k$ 
21       $\tilde{\mathbf{u}}_i = \tilde{\mathbf{u}}_i - h \cdot \bar{\mathbf{u}}_k$ 
22    end
23  end
24   $h = \sqrt{\tilde{\mathbf{v}}_i^T \cdot \mathbf{Q} \cdot \tilde{\mathbf{v}}_i}$  ; % Energienorm
25   $h_u = \sqrt{\tilde{\mathbf{u}}_i^T \cdot \mathbf{Q} \cdot \tilde{\mathbf{u}}_i}$  ; % Energienorm
26  if  $h > deftol$  then
27     $\tilde{\mathbf{v}}_i = 1/h \cdot \tilde{\mathbf{v}}_i$ 
28     $\tilde{\mathbf{u}}_i = 1/h \cdot \tilde{\mathbf{u}}_i$ 
29     $\bar{\mathbf{V}} = [\bar{\mathbf{V}} \tilde{\mathbf{v}}_i]$ 
30     $\bar{\mathbf{U}} = [\bar{\mathbf{U}} \tilde{\mathbf{u}}_i]$  ; % Vektor hinzufügen
31  else if  $h_u > deftol$  then
32     $\tilde{\mathbf{v}}_i = 0$ 
33     $\tilde{\mathbf{u}}_i = \tilde{\mathbf{u}}_i$ 
34     $\bar{\mathbf{V}} = [\bar{\mathbf{V}} \tilde{\mathbf{v}}_i]$ 
35     $\bar{\mathbf{U}} = [\bar{\mathbf{U}} \tilde{\mathbf{u}}_i]$ 
36  else
37    Deflation
38  end
39 end

```

maßgeblich von der Anzahl der Eingänge (entspricht der Spaltenzahl der Eingangsmatrix \mathbf{B}) ab. Die Verwendung eines Vektors \mathbf{b} als Startvektor von Arnoldi-Verfahren führt bei gleicher Ordnung der Reihenentwicklung und Entwicklungsstellenanzahl zur kleinsten reduzierten Systemdimension. Aus der Eingangsmatrix \mathbf{B} eines MIMO-Systems kann nach [BGKVW07, S.1210f] ein Startvektor für Arnoldi-Verfahren interpoliert werden. Ein solcher

Vektor \mathbf{b}_k , der anstelle der Eingangsmatrix \mathbf{B} als Eingangsgröße eines SOAR-Verfahrens verwendet werden kann, ist mithilfe

$$\mathbf{b}_k = \mathbf{B} \cdot \mathbf{t}_k \quad \text{mit } (\cdot)_k \text{ zugehörig zu Entwicklungspunkten } s_{ES_k} \quad (2.13)$$

zu berechnen. Der Vektor $\mathbf{t}_k \in \mathbb{R}^r$ wird in [Ges09, S.25] als Satz tangentialer Richtungen bezeichnet. Er kann aber auch als Wichtungsvektor der einzelnen Spalten der Eingangsmatrix \mathbf{B} im Startvektor \mathbf{b}_k verstanden werden. Es ist generell möglich, den Vektor \mathbf{t}_k bei rationalen Verfahren für jede Entwicklungsstelle s_{ES_k} neu zu definieren. In [BGKVV07] werden vier Strategien vorgeschlagen, den Vektor \mathbf{t}_k zu definieren. Neben der zufälligen Wahl der Einträge des Vektors \mathbf{t}_k , wird die Wahl mithilfe vorgesetzter Balanced Truncation (BT)-MOR und alternativ als Eigenwerte des Systems vorgeschlagen. Die beiden letztgenannten schneiden für die dort gezeigten Systeme mit den besten Ergebnissen ab. Eine MOR zur Bestimmung von \mathbf{t}_k wird als zu aufwändig angesehen. Die Eigenwerte mechanischer Systeme hängen grundsätzlich nicht mit der (frei wählbaren) Eingangsreihenfolge zusammen. Folglich kann im Zuge dieser Arbeit nur der genannte Ansatz der zufälligen Interpolation zu Vergleichszwecken genutzt werden. Physikalisch sinnvoll erscheint die Verwendung eines Einser-Vektors $\mathbf{t} = \mathbf{1}$ zur Gleichwichtung der Eingänge. Dies kann als harmonische Einheitslast auf alle Systemeingänge interpretiert werden.

2.6. KORRELATIONSKRITERIEN

Um das Ergebnis einer MOR bewerten zu können, also, inwieweit das Systemverhalten des reduzierten Systems dem des vollen Systems entspricht, ist es notwendig, eine Modellkorrelation durchzuführen. Dabei kann nach [Imm14] der größte Teil der Korrelationskriterien in

- eigenfrequenzbezogene Kriterien,
- eigenvektorbezogene Kriterien,
- übertragungsfunktionsbezogene Kriterien und
- matrixbezogene Kriterien

unterteilt werden. Die meisten in jener Arbeit thematisierten Kriterien stellen lediglich kleine Variationen von grundlegenden Herangehensweisen dar, die hier kurz erläutert werden. Matrixbezogene Kriterien werden nicht verwendet.

Für die ersten beiden Kriterien spielt die Darstellbarkeit eines ungedämpften Systems nach Gleichung 2.4 als Eigenwertproblem unter Verwendung der Spektralmatrix der Eigenwerte $\mathbf{\Lambda} \in \mathbb{R}^{m \times m}$ und der Modalmatrix $\mathbf{\Phi} \in \mathbb{R}^{N \times m}$ als

$$\mathbf{0} = (\mathbf{K} - \mathbf{\Lambda} \cdot \mathbf{M}) \cdot \mathbf{\Phi} \quad (2.14)$$

eine Rolle. So sind in den genannten Matrizen alle Systeminformationen² enthalten. Das Übertragungsverhalten der Anbindungspunkte (oder Ein- und Ausgänge) mit der umgebenden Struktur wird von der Übertragungsfunktion nach Gleichung 2.9 dargestellt, die für die dritte Kriteriengruppe von Bedeutung ist.

Grundsätzlich ist das dynamische Verhalten des reduzierten Systems nur für einen gewissen Frequenzbereich von Bedeutung.

2.6.1. EIGENFREQUENZBEZOGENE KRITERIEN

Eine einfache Möglichkeit, die Reduktionsgüte abzuschätzen, bietet der Vergleich der Eigenfrequenzen des vollen Systems mit denen des reduzierten Systems. Die Eigenfrequenzen f_i lassen sich für ungedämpfte Systeme aus den Eigenwerten $f_i = \frac{1}{2\pi}\sqrt{\Lambda_{ii}}$ berechnen. Es sei das NRFD-Kriterium nach [Gra98] genannt, das die relative Abweichung der einzelnen Eigenfrequenzen bestimmt, wobei eine gute Übereinstimmung bei kleinen Werten $\text{NRFD}_i < 3\%$ vorliegt. In der vorliegenden Arbeit wird angelehnt an [Feh11] das schärfere Kriterium $\text{NRFD}_i < 0.1\%$ verwendet. Es wird die Berechnungsvorschrift

$$\text{NRFD}_i = \frac{|\bar{f}_i - f_i|}{f_i} \quad (2.15)$$

verwendet. Für den Fall, dass das reduzierte im Gegensatz zum vollen System keine Starrkörpermoden aufweist oder einzelne Eigenfrequenzen nicht abgebildet werden, ist zu berücksichtigen, dass die Werte des reduzierten Systems \bar{f}_i mit denen des vollen f_j mit $i \neq j$ verglichen werden müssen. [Imm14, S.19]

2.6.2. EIGENVEKTORBEZOGENE KRITERIEN

Eine Möglichkeit, die Eigenvektoren $\phi_i \in \mathbb{R}^N$ in den Modalmatrizen $\Phi \in \mathbb{R}^{N \times m}$ zu vergleichen, ist das sogenannte Modal Assurance Criterion (MAC). Wie in Abbildung 2.1 gezeigt, wird dabei die Orthogonalität der Eigenvektoren überprüft. Eine Berechnungsvorschrift des von [Kou09] als modifiziertes MAC (modMAC) bezeichneten Verfahrens lautet

$$\text{modMAC}_{ij} = \frac{(\phi_i^T \cdot \mathbf{M} \cdot \bar{\psi}_j)^2}{(\phi_i^T \cdot \mathbf{M} \cdot \phi_i) \cdot (\bar{\psi}_j^T \cdot \mathbf{M} \cdot \bar{\psi}_j)} \quad (2.16)$$

Sind die Modalmatrix Φ des vollen Systems und die expandierte Modalmatrix des reduzierten Systems³ $\bar{\Psi} \in \mathbb{R}^{N \times n}$ massennormiert, so ergibt sich die Berechnungsvorschrift zu

$$\text{modMAC}_{ij} = (\phi_i^T \cdot \mathbf{M} \cdot \bar{\psi}_j)^2 \quad (2.17)$$

²Dies gilt für das Lösen des vollständigen Eigenwertproblems mit $m = N$. Für $m < N$ wird das Systemverhalten lediglich approximiert.

³Die Expansion wird in Gleichung 2.18 gezeigt. Die expandierte Modalmatrix des reduzierten Systems hat n Moden, da für das reduzierte System allgemein die Berechnung des vollständigen Eigenwertproblems vorliegt.

Bei völliger Übereinstimmung der Modalmatrizen ergibt sich die Hauptdiagonale der modMAC-Matrix zu Eins. Sind die Eigenvektoren zueinander massenorthogonal, werden alle Nebendiagonalelemente zu Null. [Imm14, S.22]

Die Eigenvektoren des reduzierten Systems liegen für KSM nicht nur in einer anderen Dimension, sondern auch bezogen auf andere Koordinaten vor. Dieser Unterschied ist mithilfe der Projektionsmatrix \mathbf{V} als Expansion der Modalmatrix des reduzierten Systems $\bar{\Phi} \in \mathbb{R}^{n \times n}$ mit

$$\bar{\Psi} = \mathbf{V} \cdot \bar{\Phi} \quad (2.18)$$

behebbar. [Imm14, S.46ff]

Wendet man das modMAC auf zwei identische Modalmatrizen an, so ist an der Größe der Nebendiagonaleinträge die Orthogonalität der Eigenvektoren erkennbar. Dieses Verfahren wird auch als AutoModMAC bezeichnet und kann zum Test der Orthogonalität der Krylov-Moden verwendet werden. [Imm14, S.23]

2.6.3. ÜBERTRAGUNGSFUNKTIONSBEZOGENE KRITERIEN

Die Beziehungen zwischen den Ein- und Ausgängen eines Systems sind in der Übertragungsfunktion enthalten. Die Übertragungsfunktionsmatrix des vollen Systems zweiter Ordnung ist in Gleichung 2.9 angegeben. Unter Zuhilfenahme von Gleichung 2.6 kann die Übertragungsfunktionsmatrix des reduzierten Systems als

$$\bar{\mathbf{H}}(s) = \bar{\mathbf{C}} \cdot (s^2 \bar{\mathbf{M}} + s \bar{\mathbf{D}} + \bar{\mathbf{K}})^{-1} \cdot \bar{\mathbf{B}} \quad (2.19)$$

dargestellt werden. Die dargestellte Invertierung muss für jeden Wert von s neu ausgeführt werden, um das Übertragungsverhalten über einen Frequenzbereich abbilden zu können. Für das reduzierte System ist diese Rechenoperation aufgrund der kleinen Systemdimension problemlos durchführbar. Jedoch ist darauf zu achten, dass bei ungedämpften Systemen die Inversion für die Eigenkreisfrequenzen $s = j \cdot \omega_k$ nicht ausführbar ist.

Diese Rechenoperation (die Inversion aus Gleichung 2.9) ist für große Systeme aufgrund der maximalen Systemdimension $N \rightarrow 10^7$ nicht praktikabel. Ist für das System das Eigenwertproblem gelöst worden, so besteht nach [Imm14, S.37] die Möglichkeit, mithilfe der Eigenmoden ϕ_k und Eigenkreisfrequenzen $\omega_k = 2\pi \cdot f_k$ die Übertragungsfunktion anzunähern

$$\mathbf{H}(s) \approx \sum_{k=1}^m \frac{\mathbf{C} \cdot \phi_k \cdot \phi_k^T \cdot \mathbf{B}}{s^2 + 2 \cdot D \cdot \omega_k \cdot s + \omega_k^2} \quad (2.20)$$

Dabei stellt D das LEHRsche Dämpfungsmaß dar. Die Approximationsgüte hängt von der Anzahl m der einbezogenen Eigenmoden ϕ_k und Eigenkreisfrequenzen ω_k ab. Mithilfe der Dämpfung $D > 0$ kann die Übertragungsfunktion auch an der entsprechenden Eigenkreisfrequenz $s = i \cdot \omega_k$ ausgewertet werden. Es wird für das LEHRsche Dämpfungsmaß $D = 0.001$

gewählt. Um die Übertragungsfunktion des reduzierten Systems aus Gleichung 2.19 auf gleiche Weise zu dämpfen, wird diese ebenfalls modal dargestellt:

$$\bar{\mathbf{H}}(s) \approx \sum_{k=1}^n \frac{\bar{\mathbf{C}} \cdot \bar{\boldsymbol{\phi}}_k \cdot \bar{\boldsymbol{\phi}}_k^T \cdot \bar{\mathbf{B}}}{s^2 + 2 \cdot D \cdot \bar{\omega}_k \cdot s + \bar{\omega}_k^2} \quad . \quad (2.21)$$

Für übereinstimmende Eigenkreisfrequenzen $\omega_k = \bar{\omega}_k$ sind gleich normierte Eigenmoden zu wählen. Für gleiche Massennorm ist es sinnvoll, die expandierten Eigenmoden des reduzierten Systems aus Gleichung 2.18 gleich den Moden des vollen Systems zu normieren:

$$\bar{\mathbf{C}} \cdot \bar{\boldsymbol{\phi}}_k = \frac{\mathbf{C} \cdot \bar{\boldsymbol{\psi}}_k}{\bar{\boldsymbol{\psi}}_k^T \cdot \mathbf{M} \cdot \bar{\boldsymbol{\psi}}_k} \quad \text{und} \quad \mathbf{C} \cdot \boldsymbol{\phi}_k = \frac{\mathbf{C} \cdot \boldsymbol{\phi}_k}{\boldsymbol{\phi}_k^T \cdot \mathbf{M} \cdot \boldsymbol{\phi}_k} \quad . \quad (2.22)$$

Die Massennormierung auf Dimension des vollen Systems führt zu stark variierende Beträgen der Eingangsanteile der Eigenvektoren. Die Übertragungsfunktionsmatrix in Gleichung 2.20 setzt sich aus betragsmäßig kleinen und großen Summanden zusammen. Führen Eigenmoden außerhalb des betrachteten Frequenzbereiches zu großen Summanden im betrachteten Frequenzbereich, wird die Übertragungsfunktionsmatrix ‚gehoben‘. Dieser Effekt kann dazu führen, dass trotz übereinstimmender Eigenfrequenzen und Eigenvektoren zwischen vollem und reduziertem System in einem Frequenzbereich die Übertragungsfunktionsmatrizen eine große Differenz aufweisen. Der Effekt kann durch eine andere Normierungsmethode abgeschwächt werden. Hierzu wird die $\|\dots\|_2$ -Norm der Eingangsanteile der Eigenmoden zur Übereinstimmung gebracht

$$\|\mathbf{C} \cdot \boldsymbol{\phi}_k\|_2 = \|\bar{\mathbf{C}} \cdot \bar{\boldsymbol{\phi}}_k\|_2 \quad . \quad (2.23)$$

Im Gegensatz zum vollen Modell kann für das reduzierte Modell aufgrund der kleineren Anzahl n über alle n Eigenmoden summiert werden.

Mithilfe von Gleichung 2.19 und der Näherung aus Gleichung 2.20 ist es möglich, die Übertragungsfunktionen des reduzierten und vollen Systems für viele diskrete Frequenzstützstellen zu berechnen. Jedoch ist es nicht praktikabel, jeden Eintrag der Frequenzgangmatrix \mathbf{H} für den interessierenden Frequenzbereich auszuwerten. Es seien zwei Normen genannt, die es erlauben, das Reduktionsergebnis effektiv zu bewerten:

- FROBENIUS-Norm: Normierung der frequenzabhängigen Übertragungsfunktionsmatrix auf einen frequenzabhängigen Wert
- \mathcal{H}_2 -Norm: Integral der FROBENIUS-Norm der Übertragungsfunktionsmatrizen über die Erregerfrequenz

FROBENIUS-NORM

Nach [Tel12, S.22] kann die FROBENIUS-Norm einer Frequenzgangmatrix $\mathbf{H}(s)$ submultiplikativ als

$$\|\mathbf{H}(s)\|_F = \sqrt{\text{spur}(\mathbf{H}(s) \cdot \mathbf{H}^*(s))} \quad (2.24)$$

dargestellt werden, wobei $[\cdot]^*$ die Adjunkte einer Matrix bezeichnet. Die frequenzabhängige, absolute Abweichung zwischen reduziertem und vollem System wird über

$$\epsilon_{H_F}^{abs}(s) = \|\mathbf{H}(s) - \bar{\mathbf{H}}(s)\|_F \quad (2.25)$$

und die relative Abweichung über

$$\epsilon_{H_F}^{rel}(s) = \frac{\|\mathbf{H}(s) - \bar{\mathbf{H}}(s)\|_F}{\|\mathbf{H}(s)\|_F} \quad (2.26)$$

berechnet. Diese Art der Fehlerbewertung wird auch in [Feh11, S.37] vorgeschlagen.

H2-NORM

Ein Fehlerskalar lässt sich über die \mathcal{H}_2 -Norm der Frequenzgangmatrizendifferenz berechnen. Die \mathcal{H}_2 -Norm einer Übertragungsfunktionsmatrix ist nach [Feh11, S.36] über

$$\|\mathbf{H}\|_{\mathcal{H}_2} = \left(\int_{-\infty}^{\infty} \|\mathbf{H}(j \cdot 2\pi \cdot f)\|_F^2 df \right)^{\frac{1}{2}} \quad (2.27)$$

möglich. Weiter wird die Modifikation als $\tilde{\mathcal{H}}_2$ -Norm vorgeschlagen

$$\|\mathbf{H}\|_{\tilde{\mathcal{H}}_2} = \left(\frac{1}{2 \cdot \pi} \cdot \int_{f_{min}}^{f_{max}} \|\mathbf{H}(j \cdot 2\pi \cdot f)\|_F^2 df \right)^{\frac{1}{2}}, \quad (2.28)$$

lediglich über einen Frequenzbereich zu integrieren. Die $\tilde{\mathcal{H}}_2$ -Norm wird auf die Differenz der Übertragungsfunktionen von vollem und reduziertem Modell angewendet

$$\epsilon_{\tilde{\mathcal{H}}_2}^{abs} = \|\bar{\mathbf{H}}(s) - \mathbf{H}(s)\|_{\tilde{\mathcal{H}}_2} \quad \text{oder} \quad \epsilon_{\tilde{\mathcal{H}}_2}^{rel} = \frac{\|\bar{\mathbf{H}}(s) - \mathbf{H}(s)\|_{\tilde{\mathcal{H}}_2}}{\|\mathbf{H}(s)\|_{\tilde{\mathcal{H}}_2}} \quad (2.29)$$

Damit ist mit einem einzelnen Wert der Vergleich verschiedener Reduktionsergebnisse in einem beliebigen Frequenzbereich möglich. Nachfolgend wird die $\tilde{\mathcal{H}}_2$ -Norm als \mathcal{H}_2 -Norm bezeichnet.

2.6.4. FEHLERBEWERTUNG

Vor allem bei der automatischen Bewertung des Reduktionsergebnisses bei iterativer Wahl der Entwicklungsstellen ist es wichtig, auf numerisch effektive Art und Weise die Abweichung des reduzierten Systems vom vollen System zu berechnen. Hierfür können grundsätzlich die genannten Korrelationskriterien verwendet werden.

Besonders eignen sich Fehlerangaben basierend auf der \mathcal{H}_2 -Norm aus Gleichung 2.29, da diese die Informationen

- der Eigenfrequenzen f_k ,
- der Eigenvektoren ϕ_k und
- der Ein- und Ausgänge (in den Matrizen \mathbf{C} und \mathbf{B})
- bezogen auf den interessierenden Frequenzbereich
- in einem einzelnen, relativen oder absoluten Wert

zusammenfassen. Somit kann das Ergebnis z.B. bei iterativ rationalen KSM nach jedem Iterationsschritt automatisch als Verbesserung oder Verschlechterung im Vergleich zum vorherigen Ergebnis bewertet werden. Diese Art der Fehlerbewertung ist ebenfalls ein gutes Bewertungsinstrument für eine geeignete Parameterwahl der nichtiterativen, rationalen Arnoldi-Verfahren und wird als solches verwendet.

Um den Frequenzbereich guter Übereinstimmung abschätzen zu können, wird zusätzlich das NFRD-Kriterium mit $NFRD < 0.1\%$ verwendet. Der Integrationsbereich des \mathcal{H}_2 -Fehlers wird auf einen Bereich guter Übereinstimmung festgelegt, da ein minimaler Frequenzbereichs-Anteil großer \mathcal{H}_2 -Abweichung den \mathcal{H}_2 -Fehler des Gesamtbereiches dominiert und somit keine Aussage über die Korrelationsgüte des Restbereiches guter Übereinstimmung getroffen werden kann.

2.6.5. ANWENDUNG AUF SYSTEME SEHR GROSSER DIMENSION

Die genannten Korrelationskriterien basieren auf einer Eigenanalyse des vollen Modells. Für Systeme sehr großer Dimension ($N > 3 \cdot 10^6$) steht aus numerischen Gründen nicht immer die Lösung des Eigenwertproblems zur Verfügung. Für den genannten Fall wird in [Feh11, S.83] vorgeschlagen, einen maximalen Fehler der Reduktion durch Näherung des Übertragungsverhaltens des vollen Modells abzuschätzen. Auf diese Art der Fehlerschätzung wird im Zuge dieser Arbeit nicht eingegangen. Eine Möglichkeit der Fehlervorhersage besteht bei KSM grundsätzlich nicht.

3. NUMERIK LINEARER GLEICHUNGSSYSTEME

Den rechentechnisch aufwändigsten Schritt der behandelten KSM stellt das Lösen linearer Gleichungssysteme (LGS) dar. Nach einer kurzen Einordnung der auftretenden LGS, wobei unterbestimmte LGS eine wesentliche Rolle spielen, wird auf iterative LGS-Löser eingegangen. Dafür notwendige Faktorisierungsverfahren werden erläutert und auf das direkte Lösen von LGS angewendet.

Für das Lösen von LGS bei KSM spielen Steifigkeitsmatrizen $\mathbf{K} \in \mathbb{R}^{N \times N}$ mechanischer System eine entscheidende Rolle. Es werden Balkenmodelle mit variierendem Querschnitt und Freiheitsgrad verwendet. Die Modelle sind im Anhang unter A.1 angegeben.

3.1. GRUNDLAGEN

Das Lösen linearer Gleichungssysteme (LGS) der Form

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \quad (3.1)$$

mit der quadratischen Koeffizienten-Matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ ist eine der Grundoperationen in der Numerik und spielt bei den meisten MOR-Verfahren eine grundlegende Rolle. Dabei können anstelle der Vektoren $\mathbf{x} \in \mathbb{R}^N$ und $\mathbf{b} \in \mathbb{R}^N$ auch die Matrizen $\mathbf{X} \in \mathbb{R}^{N \times \dots}$ und $\mathbf{B} \in \mathbb{R}^{N \times \dots}$ verwendet werden. Diese sind in den angewendeten Fällen durch eine kleine Spaltenzahl gegenüber der Zeilenzahl gekennzeichnet. Bei den in dieser Arbeit verwendeten mechanischen Systemen handelt es sich um dünnbesetzte Matrizen \mathbf{A} mit vollbesetzter Hauptdiagonale und Bandstruktur. Es werden sowohl kleine Systeme ab einer Dimension $N = 10^4$, als auch große Systeme mit einer Dimension bis zu $N = 10^7$ verwendet. Lösen eines LGS bedeutet, bei bekanntem \mathbf{A} und \mathbf{b} , den Vektor \mathbf{x} zu berechnen. Das kann wie folgt als Inversion von \mathbf{A} ausgedrückt werden:

$$\mathbf{x} = \mathbf{A}^{-1} \cdot \mathbf{b} \quad (3.2)$$

Im Gegensatz zu anderen Reduktionsverfahren, wie z.B. der statischen Reduktion, können bei den KSM unterbestimmte LGS auftreten. Ein unterbestimmtes LGS liegt vor, wenn nicht alle Zeilen der Koeffizientenmatrix \mathbf{A} voneinander linear unabhängig sind, die Matrix \mathbf{A} also keinen vollen Rang aufweist. Dies tritt z.B. auf, wenn die Koeffizienten-Matrix \mathbf{A}

Tabelle 3.1.: Vergleich Direkter und Iterativer Lösungsverfahren linearer Gleichungssysteme

	Direkte Verfahren	Iterative Verfahren
Einmaliger Aufwand (z.B. durch Vorkonditionierung)	groß	klein
Aufwand pro Vektor \mathbf{b}	klein	groß
Aufwand für kleine Systeme ($N \approx 10^{1\dots5}$)	klein	größer
Aufwand für große Systeme ($N \approx 10^{4\dots7}$)	groß	kleiner
Verfahrensgrenzen	Rundungsfehler Arbeitsspeicherauslastung (Abbruch)	Divergenz bzw. langsame Konvergenz
Umgang mit Singularität von \mathbf{A}	falsche oder schlechte Ergebnisse	Divergenz

der Steifigkeitsmatrix \mathbf{K} aus Gleichung 2.4 entspricht. Der Fall tritt beispielsweise beim einfachen Arnoldi-Algorithmus aus 2.4 auf. Die Gleichung 2.4 entsprechende statische Gleichung lautet

$$\mathbf{K}\mathbf{q} = \mathbf{B}\mathbf{u} \quad . \quad (3.3)$$

Das LGS ist statisch unterbestimmt, wenn bei dem zugrundeliegenden System Starrkörperbewegung möglich ist. Für diesen Fall ist keine eindeutige Berechnung des Koordinatenvektors $\mathbf{q} \in \mathbb{R}^N$ aus dem Lastvektor $\mathbf{u} \in \mathbb{R}^r$ möglich. Die Invertierung in

$$\mathbf{q} = \mathbf{K}^{-1} \cdot \mathbf{B}\mathbf{u} \quad (3.4)$$

ist nicht möglich, die Koeffizienten-Matrix \mathbf{K} bzw. \mathbf{A} ist singular und positiv semidefinit. Die Behandlung solcher unterbestimmter LGS stellt in der Numerik eine besondere Herausforderung dar.

Da die Inversion einer Matrix eine numerisch ineffektive Operation ist, wird \mathbf{x} ohne die explizite Berechnung dieser Inversen bestimmt. Die Verfahren zum Lösen von LGS können in zwei Verfahrensgruppen eingeteilt werden, in direkte und iterative Lösungsverfahren. Dabei unterscheiden sich die beiden Verfahrensgruppen in der Anwendung hauptsächlich nach den in Tabelle 3.1 angegebenen Kriterien, die an [Dan14] angelehnt sind.

Tabelle 3.1 zeigt, dass direkte Lösungsverfahren bei kleiner und iterative Verfahren bei großer Systemgröße eher zum gewünschten Ziel führen. Jedoch ist die Instabilität iterativer Verfahren bei Singularität oder singularitätsnahen Zuständen ein Grund, auch bei großen Systemen direkte Lösungsverfahren anzuwenden. Sollten viele Vektoren \mathbf{b} vorliegen, so besteht die Möglichkeit, dass der hohe einmalige Aufwand bei direkten Verfahren durch den relativ hohen Aufwand der iterativen Verfahren je Vektor \mathbf{b} ausgeglichen wird.

Im Folgenden wird als Koeffizienten-Matrix \mathbf{A} die Steifigkeitsmatrix \mathbf{K} aus Gleichung 2.4 für ungefesselte Systeme verwendet. Die Matrix ist also singular und es besteht die Herausforderung an das Lösungsverfahren, die Lösbarkeit des LGS zu ermöglichen und dabei die charakteristischen Eigenschaften des zugrundeliegenden mechanischen Systems,

wie Eigenfrequenzen und Eigenmoden, in bestmöglicher Weise beizubehalten. Für rationale KSM mit Entwicklungsstellen $f_{ES} > 0$ sind die Koeffizienten--Matrizen \mathbf{A} negativ definit. Die resultierenden Gleichungssysteme sind damit lösbar.

3.2. SINGULARITÄT DER KOEFFIZIENTENMATRIX

Bei der Unterbestimmtheit der verwendeten mechanischen Systeme handelt es sich speziell um eine Semi-Definitheit der Koeffizientenmatrix \mathbf{A} , da abhängig von der Anzahl der Freiheitsgrade der einzelnen FE-Knoten bis zu sechs der Eigenfrequenzen gleich Null sein können. Diese sind der Starrkörperbewegung des ungefesselten Systems zugeordnet. Wie unter 3.1 gezeigt, ist die Invertierung aus Gleichung 3.4 damit nicht möglich. Es werden Ansätze vorgestellt, um das unterbestimmte System dennoch zu lösen.

3.2.1. RANDBEDINGUNGEN DES SYSTEMS

Eine einfache Möglichkeit, die Lösbarkeit des System auf mechanisch nachvollziehbare Art und Weise sicherzustellen, besteht in der Fixierung von Freiheitsgraden [Bet97, S.108]. Es wird angenommen, dass die Bewegung einzelner Freiheitsgrade nicht möglich ist. Da es sich hierbei nur um Anbindungs-Freiheitsgrade mit der umgebenden Struktur handeln kann, deren Ein- und Ausgangsverhalten mithilfe der MOR simuliert werden soll, scheidet diese Möglichkeit aus. Durch falsche Auswahl dieser mindestens sechs Freiheitsgrade¹ bestünde darüber hinaus die Gefahr, nicht alle sechs Starrkörperbewegungen des Systems zu verhindern. Dies steht einer Automatisierbarkeit entgegen.

Eine weitere Möglichkeit besteht darin, alle sechs Freiheitsgrade eines Strukturknotens zu fesseln. Dies hat den Vorteil der Automatisierbarkeit, da sichergestellt ist, dass keine Starrkörperbewegung möglich ist. Jedoch stimmen nur noch die Eigenmoden des ungefesselten Systems mit denen des gefesselten Systems überein, bei denen die fixierten Freiheitsgrade keine Bewegung ausführen, es sich also um einen Schwingungsknoten handelt. Das Systemverhalten wird dadurch erheblich verändert. Diese Methode ist also lediglich für den akademischen Zweck geeignet, den Einfluss des nachfolgend genannten Verfahrens auf die Reduktionsgüte zu quantifizieren.

3.2.2. VERWENDUNG EINER GENERELLEN DIAGONALPERTURBATION

In [Kou09, S.115] wird vorgeschlagen, das System nach Gleichung 3.1 zu verstimmen. Es wird stattdessen

$$(\mathbf{A} + \alpha \cdot \mathbf{A}_{diag}) \cdot \mathbf{x} = \mathbf{b} \quad (3.5)$$

¹Die kleinste Anzahl zu sperrender Freiheitsgrade und die Anzahl der Starrkörpermoden des Systems sind gleich

Tabelle 3.2.: Iterative Methoden zur Lösung linearer Gleichungssysteme in MATLAB

Funktion	Art der Matrix	Methodenname
pcg	hermitesch, positiv definit	Preconditioned conjugate gradient method
minres	hermitesch	Minimum residual method
symmlq	hermitesch	Symmetric LQ method
lsqr	allgemein	Conjugate gradients on normal equations
qmr	allgemein	Quasi-minimal residual method
gmres	allgemein	Generalized minimum residual method
cgs	allgemein	Conjugate gradient squared method
bicg	allgemein	BiConjugate gradient method
bicgstab	allgemein	BiConjugate gradient stabilized method

gelöst. Dabei handelt es sich bei $\mathbf{A}_{diag} \in \mathbb{R}^{N \times N}$ um die Matrix der Diagonalelemente von \mathbf{A} , also

$$\mathbf{A}_{diag} = \begin{bmatrix} a_{ii} \end{bmatrix}, \quad i = 1, 2, \dots, N \quad . \quad (3.6)$$

Der Parameter α ist dabei erfahrungsgemäß in der Größenordnung 10^{-9} zu wählen und kann nach [Kou09] über die Größenordnungsverteilung der Einträge \mathbf{A} und die gewünschte Approximationsgüte gegenüber dem Ausgangsmodell berechnet werden.

Physikalisch betrachtet wird jeder Freiheitsgrad durch eine skalare Feder sehr kleiner Federkonstante mit einem festen Bezugssystem verbunden. Die Federsteifigkeit dieser Feder entspricht α mal der Steifigkeitssumme am jeweiligen Freiheitsgrad.

Diese Methode wird in der vorliegenden Version von MORPACK und im Rahmen dieser Arbeit verwendet.

Der Ansatz, Freiheitsgrade mit sehr schwachen Federn zu verbinden, kann auch auf einen Teil der Freiheitsgrade angewendet werden. So können nur die Eingänge, also die Freiheitsgrade, die mit der umliegenden Struktur verbunden sind, mit schwachen Federn mit einem festen Bezugssystem verbunden werden. Diese Herangehensweise kann ähnlich der allgemeinen Diagonalperturbation ausgeführt werden, jedoch ist der Parameter α zu erhöhen.

3.3. ITERATIVE LÖSUNGSVERFAHREN

Wie in Tabelle 3.1 aufgezeigt, kommen für Systeme mit großem Freiheitsgrad eher iterative Verfahren in Frage, da diese für solche Systeme weniger Systemressourcen benötigen.

Nach [HH05, S.138] stehen in MATLAB die Funktionen nach Tabelle 3.2 zum iterativen Lösen von LGS zur Verfügung.

Eine Einordnungshilfe zur Verwendung der Verfahren für verschiedene Arten der Koeffizienten-Matrix \mathbf{A} liefert [Get09, S.20]. Er empfiehlt für symmetrische und positiv definite Koeffizienten-Matrizen das pcg-Verfahren und für symmetrische, nicht positiv definite Matrizen je nach Kondition der Matrix \mathbf{A} das lsqr-Verfahren bzw. das minres-Verfahren.

Jedoch sind nach [Mei08, S.112 ff.] für die hier genannten Lösungsverfahren generell reguläre Koeffizienten-Matrizen Bedingung.

In der vorliegenden Version von MORPACK wird das preconditioned conjugate gradients (pcg)-Verfahren verwendet, wobei wie unter 3.2.2 beschrieben, die Koeffizienten-Matrix mithilfe einer allgemeinen Diagonalperturbation α in den positiv definiten Raum gehoben wird. Es lässt sich zeigen, dass das Konvergenzverhalten aller Verfahren in ähnlicher Weise von α abhängt und somit eine Reduzierung der Diagonalperturbation mithilfe einer Verfahrensänderung nicht möglich ist. Das pcg-Verfahren zeigt für mechanische Systeme das beste Konvergenzverhalten und liefert somit eine Lösung unter kleinem Zeitaufwand. Um jedoch ein sinnvolles Ergebnis in angemessener Zeit zu erzielen, ist es notwendig, die Koeffizienten-Matrix \mathbf{A} in geeigneter Weise zu faktorisieren. Faktorisierung ist eine Art der Vorkonditionierung.

3.4. FAKTORISIERUNGSVERFAHREN

Es ist möglich, die quadratische Matrix \mathbf{A} in Faktoren zu zerlegen. Solch eine Faktorisierung kann je nach Genauigkeit in verschiedener Art und Weise für das Lösen von LGS verwendet werden. So ist es nach [HH05, S.137] notwendig, einem iterativen Lösungsverfahren eine Vorkonditionierung voranzusetzen. Dabei sind die grundsätzlichen Eigenschaften des betrachteten Systems bei der Wahl eines guten Vorkonditionier-Verfahrens von Bedeutung. Als genutzte Vorkonditionierer werden die unvollständige Cholesky- und unvollständige LU-Faktorisierung genannt.

Diese beiden Faktorisierungen, vollständig und unvollständig ausgeführt, gehören zu den meistverwendeten Zerlegungsverfahren bei der Lösung von LGS und werden deshalb näher erläutert.

Faktorisierungsverfahren werden in einen theoretischen Hintergrund eingeordnet und die Eigenschaften der zur Verfügung stehenden MATLAB-Funktionen bei der Anwendung auf Koeffizientenmatrizen \mathbf{A} verschiedener Dimension getestet. Der Einfluss von Permutation, also dem Vertauschen von Zeilen und Spalten, wird untersucht.

Für die grundlegende Herangehensweise der LU- und Cholesky-Faktorisierung und entsprechende Algorithmen sei auf [Mei08, S.36ff] verwiesen. Für die LU-Zerlegung werden in [Saa03] eine Vielzahl von Modifikationen, unter anderem verschiedene unvollständige Verfahren und parallelisierbare Algorithmen, angegeben. Auf diese wird hier nicht weiter eingegangen.

Eine einfache Form der Zerlegung einer Matrix stellt nach [Mei08, S.36ff] die Gauß-Elimination dar. Hier wird Gleichung 3.1 sukzessive in die Form

$$\mathbf{L} \cdot \mathbf{U} \cdot \mathbf{x} = \mathbf{b} \quad (3.7)$$

mit der unteren Dreiecksmatrix $\mathbf{L} \in \mathbb{R}^{N \times N}$ und der oberen Dreiecksmatrix $\mathbf{U} \in \mathbb{R}^{N \times N}$ transformiert. Das LGS wird dann über Vor- und Rücksubstitution gelöst. Ein entsprechender Algorithmus ist in [Mei08, S.42] angegeben.

Die Zerlegung der Koeffizientenmatrix \mathbf{A} in die zwei Faktoren \mathbf{L} und \mathbf{U}

$$\mathbf{A} = \mathbf{L} \cdot \mathbf{U} \quad (3.8)$$

wird dabei als LU-Faktorisierung² bezeichnet. Die Gauß-Elimination selbst ist für Systeme großer Dimension numerisch uneffizient. Es werden viele Einträge in den Matrizen \mathbf{L} und \mathbf{U} berechnet, deren Beträge die Maschinengenauigkeit unterschreiten. Diese haben somit keinen Einfluss auf die Lösung, binden jedoch Rechenkapazität. Faktorisierungsverfahren verwenden aus diesem Grund Pivottisierung, das bedeutet Wichtung und Permutation, von Zeilen und Spalten der Matrix, um zwischen für die Berechnung relevanten und irrelevanten Einträgen unterscheiden zu können.

Für symmetrische, positiv definite Matrizen kann der Aufwand für die Faktorisierung mithilfe der Cholesky-Zerlegung, deren Eigenschaften unter 3.4.1 erläutert werden, verringert werden. [Mei08]

3.4.1. CHOLESKY-FAKTORISIERUNG

Bei der Cholesky-Faktorisierung wird die Matrix \mathbf{A} in zwei Faktoren zerlegt

$$\mathbf{A} = \mathbf{L} \cdot \mathbf{L}^T \quad , \quad (3.9)$$

also in die untere Dreiecksmatrix $\mathbf{L} \in \mathbb{R}^{N \times N}$ und ihre Transponierte.

Die Ausführbarkeit der Cholesky-Zerlegung ist ein Kriterium für die Positiv-Definitheit einer Matrix. Diese Eigenschaft und die Symmetrie der Koeffizienten-Matrix \mathbf{A} werden von dieser Zerlegung laut [Mei08, S.211] gezielt ausgenutzt. Weiter beschreibt er im Vergleich zur vollständigen Faktorisierung die unvollständige Cholesky-Faktorisierung als Erweiterung dieser Zerlegung um die Fehlermatrix $\mathbf{F} \in \mathbb{R}^{N \times N}$

$$\mathbf{A} = \mathbf{L} \cdot \mathbf{L}^T + \mathbf{F} \quad . \quad (3.10)$$

Dabei existieren laut Definition in der unteren Dreiecksmatrix \mathbf{L} lediglich an den Stellen von Null verschiedene Einträge, an denen auch in der Matrix \mathbf{A} Einträge existieren. Diese Variante wird als ‚nofill‘ bezeichnet, da bei der Zerlegung keine Null-Einträge mit Zahlenwerten ‚gefüllt‘ werden. Dies ist von Bedeutung, da bei den verwendeten sparse-Matrizen Null-Einträge nicht gespeichert werden. Bei ‚fillin‘ steigt damit die Größe der Matrix an und im Arbeitsspeicher ist das Aufweiten der Matrixspalten notwendig. Bei der

²Im deutschsprachigen Raum ist auch die Bezeichnung LR-Zerlegung üblich, welche statt L für ‚lower‘ und U für ‚upper‘ die Abkürzung L für ‚linke‘ und R für ‚rechte‘ Dreiecksmatrix verwendet. In dieser Arbeit wird der MATLAB-lu-Funktion entsprechend die Bezeichnung ‚LU‘ verwendet.

‚nofill‘-Variante ist die Fehlermatrix \mathbf{F} häufig in der Größenordnung des ersten Summanden aus Gleichung 3.10, es handelt sich damit um eine stark fehlerbehaftete Faktorisierung.

ANWENDUNG DER UNVOLLSTÄNDIGEN CHOLESKY-FAKTORISIERUNG

Die in MATLAB implementierte Funktion für die unvollständige Cholesky-Zerlegung `ichol` bietet die Möglichkeit, den ‚fillin‘ zu begrenzen. Es wird ein spaltenabhängiger Normwert eingeführt und Elemente, die diesen unterschreiten, werden aus der Matrix \mathbf{L} entfernt. Der Benutzer hat mit dem Droptoleranz-Parameter *droptol* Einflussmöglichkeit auf diesen Prozess. Je näher an Null die Droptoleranz gewählt wird, desto genauer die Faktorisierung und desto größer, die Dimension der Dreiecksmatrix \mathbf{L} . Diagramm 3.1 zeigt die Abhängigkeit der Zerlegungs-Faktoren vom Wert der Droptoleranz.

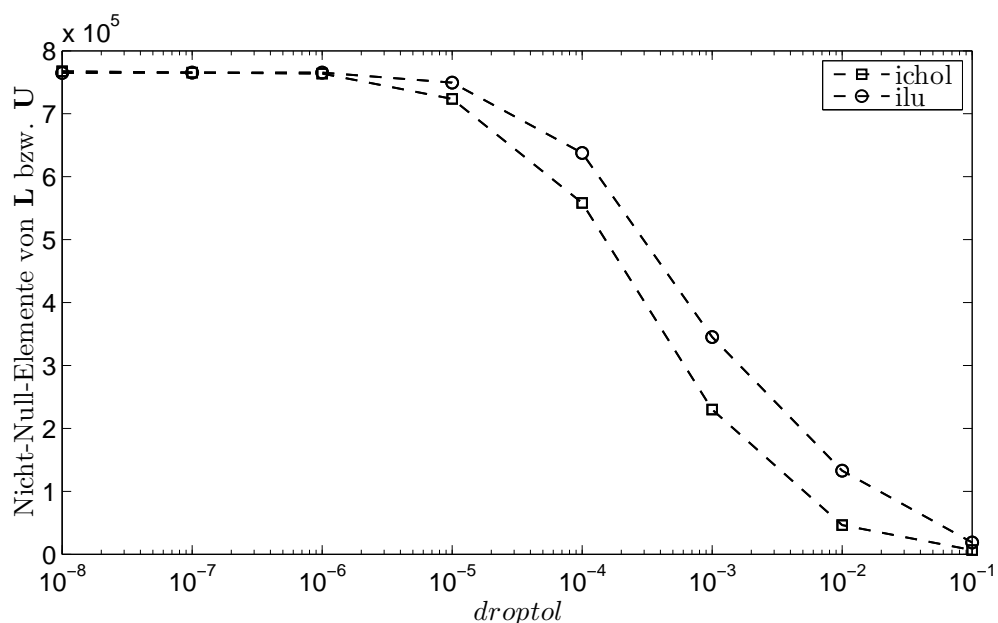


Diagramm 3.1: Anzahl der NNE der Matrix \mathbf{L} für variierende Droptoleranz bei der unvollständigen Cholesky-Faktorisierung und der Matrizen \mathbf{L} und \mathbf{U} bei der unvollständigen LU-Faktorisierung

Anhand des `pcg`-Verfahrens wird der Einfluss der Droptoleranz auf das Konvergenzverhalten gezeigt. Es wird die Steifigkeitsmatrix eines nach 3.2.1 gefesselten Systems verwendet, es gibt also keinen Einfluss der Singularität. Die Dimension der Matrix \mathbf{A} beträgt $N \approx 3500$. Die notwendigen Iterationsschritte zum Erreichen einer Fehlertoleranz von 10^{-6} beim Lösen von Gleichung 3.1 für einen zufällig gewählten Vektor \mathbf{b} sind für verschiedene Werte der Droptoleranz in Diagramm 3.2 angegeben.

Es wird gezeigt, dass eine niedrig gewählte Droptoleranz zu einem sehr guten Konvergenzverhalten des `pcg`-Verfahrens führt. Ist es das Ziel, den Speicherbedarf der Dreiecksmatrix \mathbf{L} zu minimieren, so hat sich für die verwendeten Testsysteme eine Droptoleranz von $\text{droptol} = 10^{-5}$ als Richtwert für ein ausreichend stabiles Konvergenzverhalten erwiesen.

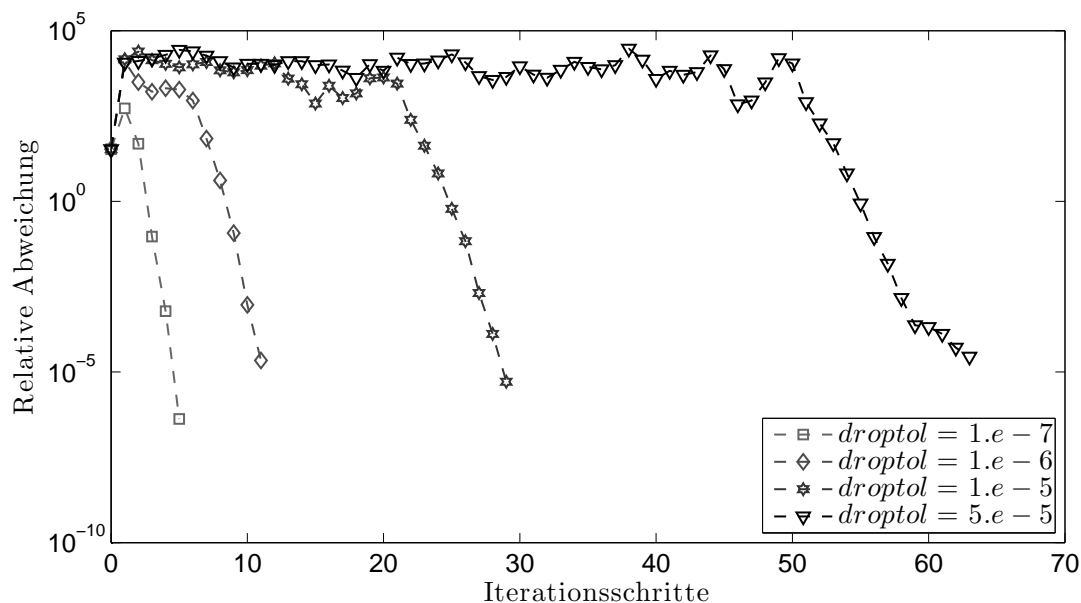


Diagramm 3.2: Iterationsschritte des pcg-Verfahrens bei variierender Droptoleranz der unvollständigen Cholesky-Faktorisierung

Wählt man eine Droptoleranz $droptol < 10^{-6}$, erhöht sich der Speicherbedarf nur unwesentlich, das Konvergenzverhalten des pcg-Verfahrens verbessert sich dagegen deutlich.

Ab einer Droptoleranz $droptol > 10^{-6}$ kommt es trotz positiv definiten Koeffizientenmatrix \mathbf{A} zum Abbruch der `ichol`-Funktion mit der Begründung fehlender Positiv-Definitheit. Das Entfernen betragsmäßig kleiner Werte aus der Dreiecksmatrix \mathbf{L} hat also Einfluss auf die scheinbare Definitheit der Koeffizientenmatrix \mathbf{A} . Das Problem kann mithilfe einer allgemeinen Diagonalperturbation nach 3.2.2 gelöst werden. Der `ichol`-Funktion kann dabei als Option ein solcher Wert α übergeben werden. Dieser Wert beginnt erfahrungsgemäß ab $\alpha \geq droptol \dots 50 \cdot droptol$, den Funktionsabbruch zu verhindern.

ANWENDUNG DER VOLLSTÄNDIGEN CHOLESKY-ZERLEGUNG

Setzt man die Droptoleranz der `ichol`-Funktion (MATLAB) auf $droptol = 0$, bedeutet dies die Ausführung der vollständigen Cholesky-Zerlegung nach Gleichung 3.9. In MATLAB können die vollständigen Zerlegungen (Cholesky und LU) auf zwei Arten berechnet werden. Zum einen können die Funktionen `chol` und `lu`, zum anderen die Funktionen `ichol` und `ilu` mit einer Droptoleranz von Null verwendet werden. Die erstgenannte Möglichkeit erweist sich als effizienter, da diese Funktionen parallelisiert auf mehreren Prozessoren arbeiten können. Dies führt zu einer deutlichen Rechenzeitverkürzung.

Die mit der vollständigen Cholesky-Zerlegung berechnete Dreiecksmatrix kann für direkte Lösungsverfahren nach Abschnitt 3.5 verwendet werden. Wie aus Diagramm 3.1 ersichtlich, steigt der Speicherbedarf für den vollständig zerlegten Faktor \mathbf{L} im Vergleich zu einem unvollständig zerlegten, der für das pcg-Verfahren verwendet werden kann ($droptol \approx 10^{-5}$), nicht wesentlich an. Müssen viele Vektoren \mathbf{b} gelöst werden, kann es also günstiger sein,

den Rechenaufwand für die Faktorisierung zu erhöhen und damit den Rechenaufwand für jeden Vektor \mathbf{b} deutlich zu verkleinern. Es sei erwähnt, dass für sparse-Matrizen der Speicherbedarf linear von der Anzahl der NNE abhängt. Dabei gilt der Zusammenhang³

$$\begin{aligned} \text{Speicher}(\mathbf{L}) &\approx 16 \cdot \text{NNE}(\mathbf{L}) \text{ Byte} \quad \text{oder} \\ \text{Speicher}(\mathbf{L}) &\approx 1.5 \cdot 10^{-8} \cdot \text{NNE}(\mathbf{L}) \text{ GigaByte} \end{aligned} \quad (3.11)$$

Um Abschätzen zu können, ob die Berechnung der vollständigen Cholesky-Faktorisierung auch für große Modelle durchführbar ist, ist der ‚fillin‘ in Abhängigkeit der Systemgröße zu quantifizieren. Die Ergebnisse zum ‚fillin‘ der vollständigen Cholesky-Zerlegung für verschiedene Modellgrößen sind in den Diagrammen 3.3 und 3.4 dargestellt. Es wird dabei die ‚fillin‘-reduzierende *symamd*-Permutation aus 3.4.3 verwendet. Es fließen also die Erkenntnisse des Abschnitts 3.4.3 in dieses Ergebnis ein und führen zu keiner weiteren Verbesserung. Zur Validierung der Ergebnisse werden zusätzlich zu den Steifigkeitsmatrizen \mathbf{K} der unter A.1 aufgeführten die Modelle aus 5.1 verwendet. Die dort vorgestellte Treibsatzwelle wird gesondert betrachtet.

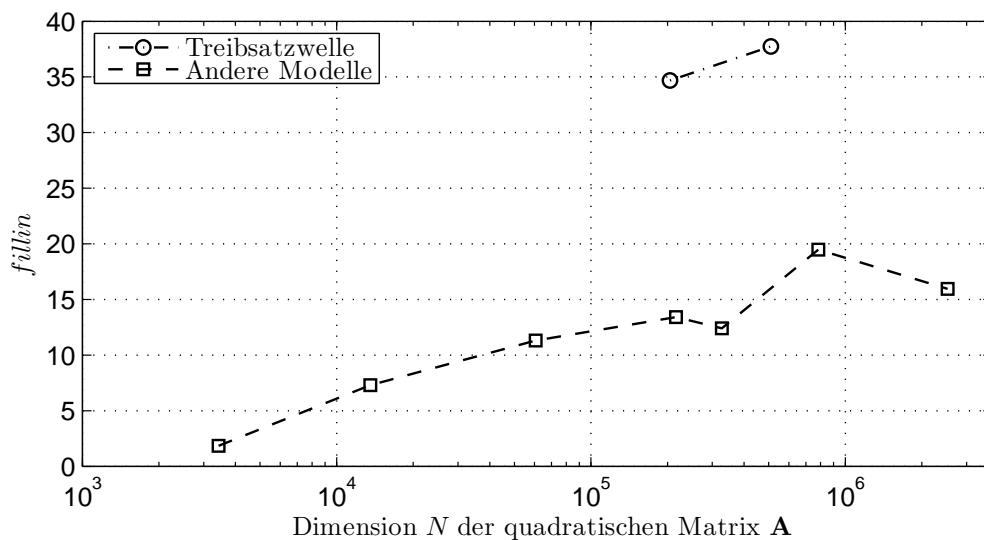


Diagramm 3.3: ‚fillin‘ der vollständigen Cholesky-Zerlegung für verschiedene Dimensionen der Koeffizienten-Matrix \mathbf{A} , die dem Freiheitsgrad der Systeme entsprechen

Der ‚fillin‘ in Diagramm 3.3 berechnet sich aus der Anzahl der NNE der Dreiecksmatrix \mathbf{L} und denen der unteren Dreiecksmatrix von \mathbf{A} zu

$$\text{fillin} = \text{NNE}(\mathbf{L}) / \text{NNE}(\mathbf{A}_{\triangleright}) - 1 \quad (3.12)$$

und gibt damit an, um welchen Faktor im Vergleich zur ursprünglichen Besetzung in \mathbf{A} Elemente in \mathbf{L} hinzugefügt werden.

³Der Speicherplatzbedarf für einen Eintrag setzt sich aus jeweils 4Byte für den Integer-Eintrag der Spalten- und Zeilennummer und 8Byte für den eigentlichen Eintrag im double-Format zusammen, der einmalige Speicherplatzbedarf einer Matrix ist für viele NNE demgegenüber vernachlässigbar.

Diagramm 3.3 zeigt einen vielfachen ‚fillin‘ der Treibsatzwelle-Modelle im Gegensatz zu den anderen Modellen. Dies ist auf deren Struktur als mit Volumenelementen voll vernetzte Struktur in Stabform zurückzuführen, worauf unter 3.4.3 eingegangen wird. Derartige Strukturen sind demnach gesondert zu betrachten.

Abgesehen davon ist ersichtlich, dass sich von der Dimension von \mathbf{A} und damit dem Freiheitsgrad des Systems nicht eindeutig auf den ‚fillin‘ und damit die NNE von \mathbf{L} schließen lässt. Die Zerlegungen der Matrizen mit einer Dimension bis zu $8 \cdot 10^5$ sind in Diagramm 3.4 auf eine weitere Art und Weise dargestellt.

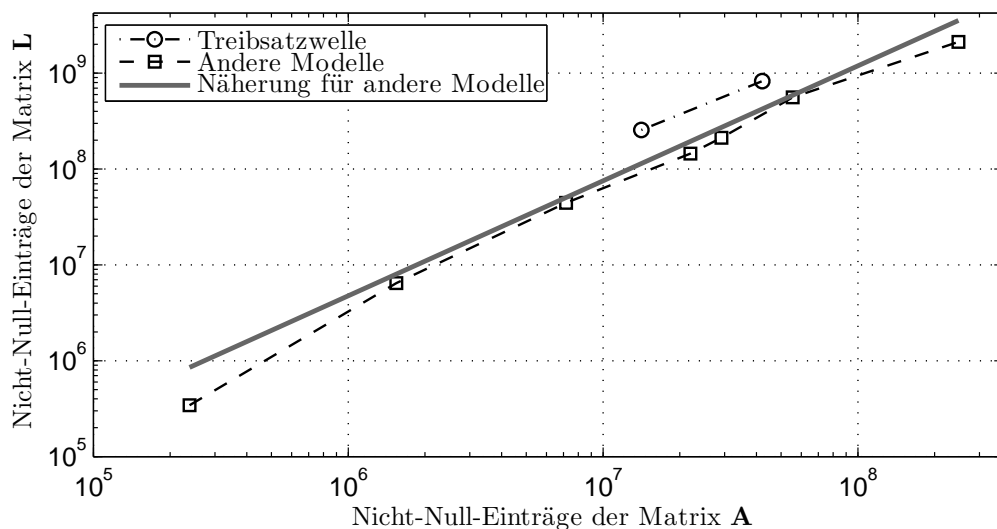


Diagramm 3.4: Anzahl der Nicht-Null-Elemente der Dreiecksmatrix \mathbf{L} gegenüber denen der Koeffizienten-Matrix \mathbf{A}

Diagramm 3.4 zeigt, dass die Anzahl der NNE der Koeffizientenmatrix \mathbf{A} einen besseren Anhaltspunkt für die Größe von \mathbf{L} liefert. Es ist die Anzahl der NNE der Matrix \mathbf{L} , die sich über Gleichung 3.11 in den Arbeitsspeicherbedarf umrechnen lässt.

In Diagramm 3.4 ist eine Gerade für die doppeltlogarithmische Skalierung dargestellt. Die zugrundeliegende Funktionsgleichung lautet

$$\text{NNE}(\mathbf{L}) = 0.3 \cdot (\text{NNE}(\mathbf{A}))^{1.2} \quad (3.13)$$

und kann in Kombination mit Gleichung 3.11 als Vorhersage des maximalen Arbeitsspeicherbedarfs der vollständigen Cholesky-Faktorisierung mit den MATLAB-Funktionen `chol` oder `ichol` unter Verwendung der `symamd`-Permutation aus Abschnitt 3.4.3 verwendet werden. Die Definition dieser Abschätzung bringt mit sich, dass für kleine Systeme nur schlechte Prognosen abgegeben werden können. Die Vorhersage ist nur für Matrizen \mathbf{L} , deren Speicherplatzbedarf den Arbeitsspeicher übersteigen kann, von Interesse. Die Einschränkung ist demnach nicht von Bedeutung. Die in MATLAB verwendete `ichol`-Funktion hat lediglich den Arbeitsspeicherbedarf der entstehenden Dreiecksmatrix \mathbf{L} . Für diese Funktion ist das

genannte Prognoseverfahren das entscheidende Kriterium für die Ausführbarkeit der Zerlegung. Im Vergleich dazu hat die `chol`-Funktion aufgrund der parallelisierten Arbeitsweise einen leicht erhöhten Arbeitsspeicherbedarf. Für die Anwendung auf weitere Modelle ist die Näherung durch zusätzliche Beispielsysteme zu bestätigen und auf größere Modelle zu erweitern. Weiter sind die Eigenschaften von Systemen, die ähnlich der Treibsatzwelle einen erhöhten ‚fillin‘ erzeugen, weiter zu charakterisieren.

Der exponentielle Zusammenhang zwischen den NNE der Koeffizientenmatrix \mathbf{A} und ihren Zerlegungen findet sich auch in den Beispielen in [Li13a, S.4] wieder.

3.4.2. LU-FAKTORISIERUNG

Die vollständige LU-Faktorisierung aus Gleichung 3.8 kann analog Gleichung 3.10 auch als unvollständige Zerlegung der Koeffizientenmatrix \mathbf{A}

$$\mathbf{A} = \mathbf{L} \cdot \mathbf{U} + \mathbf{F} \quad (3.14)$$

mit der Fehlermatrix \mathbf{F} dargestellt werden. Auch die unvollständige LU-Zerlegung steht analog der unvollständigen Cholesky-Zerlegung in MATLAB als ‚nofill‘- oder ‚fillin‘- Variante mit Droptoleranz zur Verfügung.

Der für den Anwender entscheidende Unterschied zwischen LU- und Cholesky-Zerlegung ist nach [HH05, S.128], dass bei der LU-Zerlegung zwei Matrizen erstellt werden und positiv-semidefinite, also singuläre Matrizen zerlegt werden können. Bei den Matrizen handelt es sich um eine untere Dreiecksmatrix \mathbf{L} und eine obere Dreiecksmatrix \mathbf{U} , die die gleiche Anzahl von Null verschiedenen Einträge haben. Es kann gezeigt werden, dass die in MATLAB verwendbare `ilu`- im Vergleich zur `ichol`-Funktion sowohl mehr Rechenzeit, als auch mehr Arbeitsspeicher benötigt.

Die Anzahl der Nicht-Null-Elemente (NNE) der Faktoren \mathbf{L} und \mathbf{U} für verschiedene Droptoleranz-Werte der MATLAB-`ilu`-Funktion im Vergleich zu `ichol` ist in Diagramm 3.1 angegeben. Da die unvollständige LU-Zerlegung im Vergleich zur unvollständigen Cholesky-Zerlegung nicht auf die Positiv-Definitheit der Matrix \mathbf{A} angewiesen ist, ist für die Zerlegung keine Diagonalperturbation notwendig. Es sei darauf hingewiesen, dass unabhängig von der Ausführbarkeit der LU-Faktorisierung ein singuläres LGS nicht lösbar ist. Da iterative Lösungsverfahren generell auf reguläre Koeffizientenmatrizen angewiesen sind, stellt die unvollständige LU-Zerlegung mit Droptoleranz als Vorkonditionierer für iterative Verfahren zum Lösen singulärer LGS keine sinnvolle Alternative zur unvollständigen Cholesky-Zerlegung dar⁴.

Wie in Diagramm 3.1 gezeigt, stimmen die Anzahl der NNE der in MATLAB berechneten Matrizen \mathbf{L}_{chol} , \mathbf{L}_{lu} und \mathbf{U}_{lu} für vollständige Zerlegungen überein. Die Besetzungsstruktur

⁴Dies gilt für nichtrationale KSM. Bei rationalen KSM sind die Koeffizientenmatrizen negativ definit, womit iterative Lösungsverfahren möglich sind. Im Zuge dieser Arbeit werden diese jedoch nicht untersucht.

von \mathbf{L}_{chol} entspricht dabei der Besetzungsstruktur von \mathbf{L}_{ilu} und $\mathbf{U}_{\text{ilu}}^T$. Daraus folgt, dass für die LU-Zerlegung im Vergleich zur Cholesky-Zerlegung der doppelte Speicherplatz benötigt wird. Da der größte Zeitanteil bei diesen Zerlegungsverfahren für das Einsortieren der ‚fillin‘-Elemente benötigt wird, wirkt sich dies stark auf die Berechnungsdauer der Zerlegung aus. In Diagramm 3.5 wird für verschiedene Modellgrößen gezeigt, dass die `ilu`- gegenüber der `ichol`-Funktion in MATLAB mehr als die doppelte Zeit benötigt.

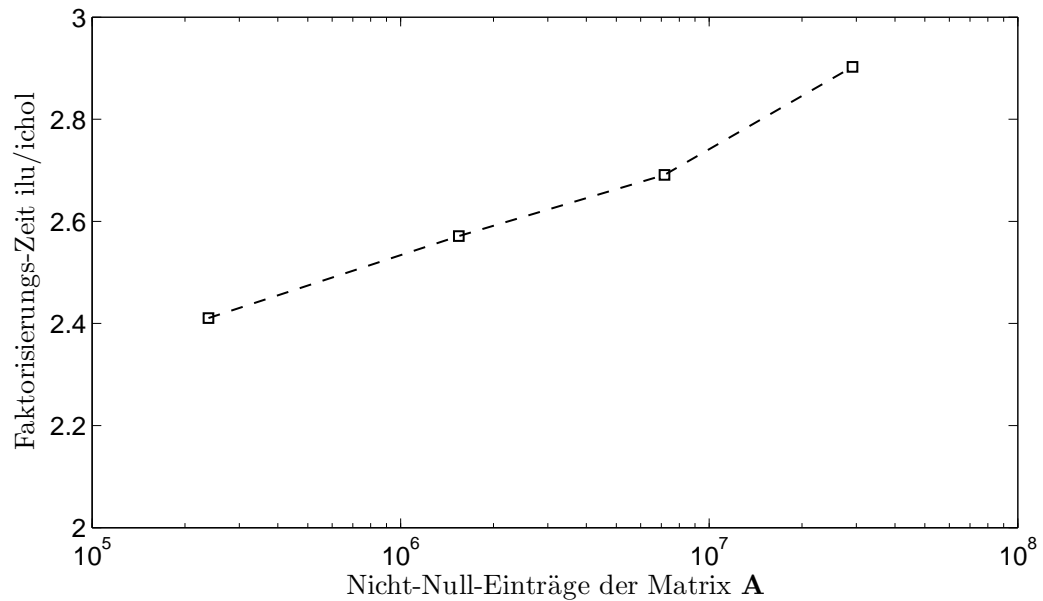


Diagramm 3.5: Vielfaches der Zeit für vollständig ausgeführte `ilu`- gegenüber `ichol`-Funktion in MATLAB

Speziell für die Lösung der LGS für rationale Krylov-Verfahren spielt die LU-Faktorisierung eine Rolle. [FE13] verweist darauf, dass der invertierte Teil aus Gleichung 2.9 für spezielle s nicht-definit ist und empfiehlt daher die Verwendung der LU-Faktorisierung als konservatives Faktorisierungs-Verfahren.

3.4.3. FILLIN-REDUKTION DURCH PERMUTATION

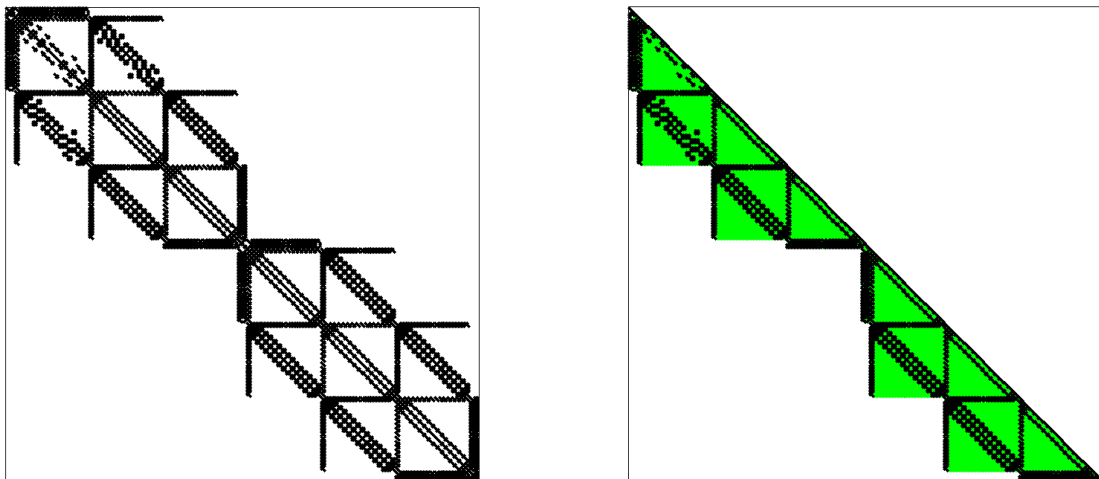
Nach [HH05, S.231f] wird durch die unvollständigen LU- und Cholesky-Funktionen `ichol` und `ilu` in MATLAB für sparse-Matrizen generell keine Permutation durchgeführt. Die Funktionen für vollständige Zerlegung `chol` und `lu` führen optional Permutationen aus. Dabei bedeutet Permutation das Vertauschen von Zeilen und Spalten der Matrix, um den ‚fillin‘ zu reduzieren. [HH05] empfiehlt die Verwendung der in MATLAB implementierten Permutationsfunktionen `symrcm`, `symamd`, `colamd` und `colperm`, wobei auf eine Testrechnung mit kleinstem ‚fillin‘ der Cholesky-Zerlegung mit `symamd`-Permutation verwiesen wird. Die genauen Bezeichnungen und Kurzbeschreibungen dieser Methoden sind in Tabelle 3.3 angegeben.

[Li13b, S.9] verweist auf das ‚fillpath‘-Theorem nach [GL80, S.95], wonach in einer Zeile des

Tabelle 3.3.: Permutationsfunktionen in Matlab

Funktion	Methodenname	Kurzbeschreibung
symrcm	Sparse reverse Cuthill-McKee ordering	Bandbreitenminimierung für quadratische Matrizen
symamd	Symmetric approximate minimum degree permutation	,fillin'-Minimierung für symmetrische, positiv definite Matrizen
colamd	Column approximate minimum degree permutation	,fillin'-Minimierung v.a. für unsymmetrische, auch rechteckige Matrizen, nur Spaltenvertauschung
colperm	Sparse column permutation based on nonzero count	mitunter gute Permutation für symmetrische und unsymmetrische Matrizen (dann nur Spaltenvertauschung)

Cholesky-Faktors \mathbf{L} nur Elemente besetzt sein können, die innerhalb der besetzten Elemente der symmetrischen Matrix \mathbf{A} liegen. Die Matrix \mathbf{L} stellt eine untere Dreiecksmatrix dar und die Hauptdiagonale von \mathbf{A} ist für mechanische Systeme voll besetzt. Demzufolge ist ,fillin' in der Matrix \mathbf{A} zwischen dem ersten Element jeder Zeile und der Hauptdiagonale möglich. Dies wird in Abbildung 3.1 anhand einer von ANSYS automatisch permutierten und exportierten Steifigkeitsmatrix gezeigt.


Abbildung 3.1.: Besetzungsstruktur der Koeffizientenmatrix $\mathbf{A} (\cong \mathbf{K})$ und ihrer cho1-Zerlegung \mathbf{L} mit Permutation nach ANSYS und grünem ,fillin'; $\text{NNE}(L) = 19559$

Eine Möglichkeit, den ,fillin' zu reduzieren, besteht darin, die Bandbreite der Matrix zu minimieren. Gute Ergebnisse sind dabei für lange, schmale Geometrien zu erwarten⁵. Die in MATLAB verfügbare `symrcm`-Funktion beinhaltet den Cuthill-McKee-Algorithmus nach [CM69]. Dieser Algorithmus erzeugt auf numerisch sehr effektive Art und Weise eine solche Bandbreitenminimierung. Wie in Abbildung 3.2 dargestellt, ergibt sich das hier verwendete Beispiel keine wesentliche Verringerung des ,fillin'. Da in der Modellordnungsreduktion nicht von langen, schmalen Geometrien ausgegangen werden kann, ist die Bandbreitenminimierung demnach nicht das optimale Werkzeug zur Minimierung des

⁵Für solche Geometrien existieren von einem Knoten ausgehend nur in eine geometrische Richtung Verbindungen zu vielen anderen Knoten. Dies ermöglicht eine sehr schmalbandige Permutation.

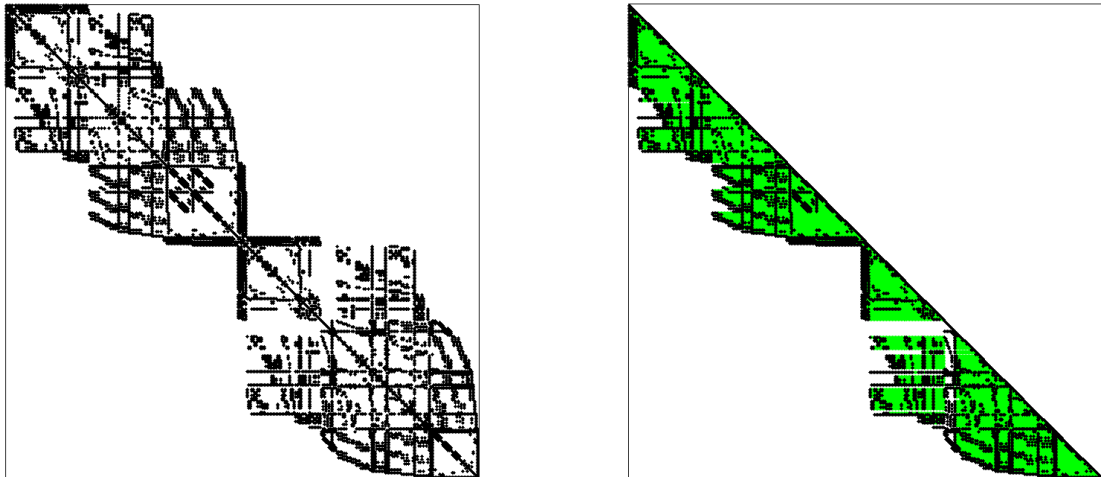


Abbildung 3.2.: Besetzungsstruktur der Koeffizientenmatrix $\mathbf{A} (\hat{=}\mathbf{K})$ und ihrer chol -Zerlegung \mathbf{L} mit Cuthill-McKee-Permutation und grünem ‚fillin‘; $\text{NNE}(\mathbf{L}) = 17239$

‚fillin‘.

An einem einfachen Beispiel, angelehnt an [Li13b], wird eine optimale Permutationsstrategie erläutert.

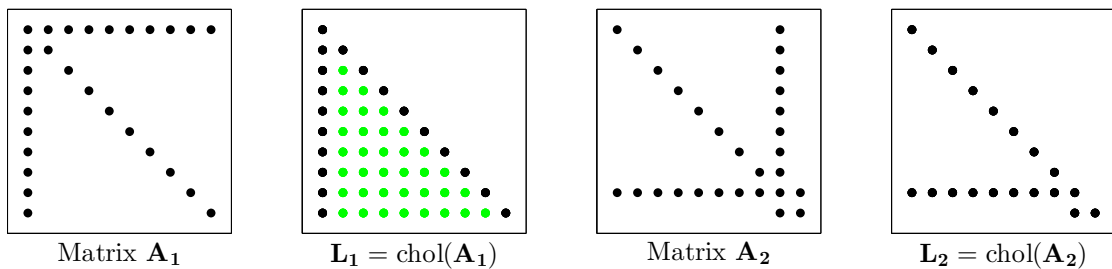


Abbildung 3.3.: Besetzungsstruktur der Koeffizientenmatrix \mathbf{A} und ihrer chol -Zerlegung \mathbf{L} für maximalen und minimalen ‚fillin‘ (grün)

Wie in Abbildung 3.3 gezeigt, kann eine Matrix mit ungünstiger Permutation (Matrix \mathbf{A}_1) zu sehr großem ‚fillin‘ führen. Unabhängig von der Bandbreite der Matrix kann eine Permutation gefunden werden (Matrix \mathbf{A}_2), die den ‚fillin‘ minimiert, im gezeigten Fall beseitigt.

Permutationen nach diesem Muster werden für große Matrizen von sogenannte minimum-degree-Permutationen erzeugt, zu denen die `symamd`-Funktion in MATLAB gehört. Die in den Abbildungen 3.1 und 3.2 gezeigte Matrix, nach dieser Funktion permutiert, ist in Abbildung 3.4 dargestellt.

Hier zeigt sich, dass das ‚fillpath‘ -Theorem zwar zutrifft, jedoch nicht alle Positionen innerhalb des damit abgegrenzten Bereichs tatsächlich mit Elementen gefüllt sein müssen. Eine Vorhersage der besetzten Positionen ist mit diesem Theorem nicht möglich, lediglich eine Abgrenzung, in welchen Bereichen ‚fillin‘ -Einträge möglich sind.

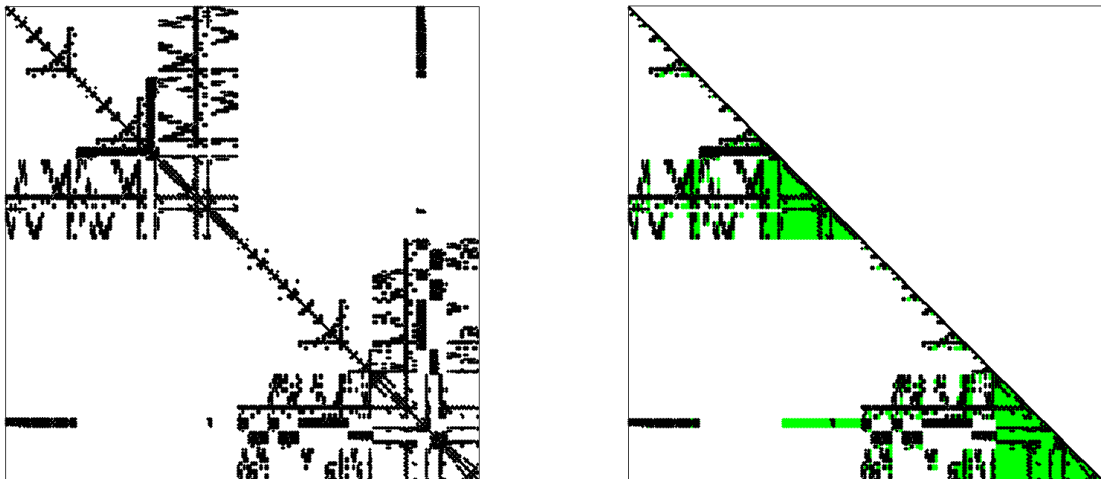


Abbildung 3.4.: Besetzungsstruktur der Koeffizientenmatrix \mathbf{A} ($\hat{=}\mathbf{K}$) und ihrer chol-Zerlegung \mathbf{L} mit symamd-Permutation und grünem ‚fillin‘; $\text{NNE}(\mathbf{L}) = 10793$

In Testrechnungen mit den genannten Matrizen verschiedener Größe liefert die `symamd`-Permutation allgemein die besseren Ergebnisse. Das Einsparpotential an Rechenzeit und Arbeitsspeicher (linear abhängig von Anzahl der NNE) für verschiedene Systemgrößen ist in den Diagrammen 3.6 und 3.7 angegeben.

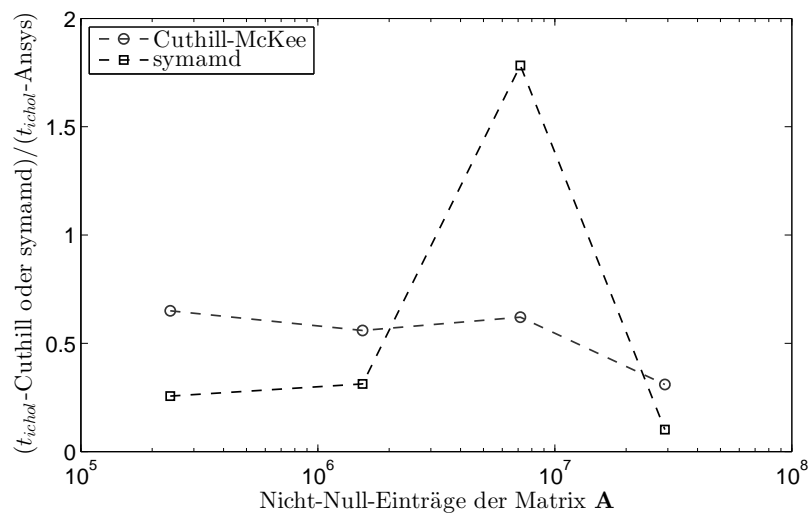


Diagramm 3.6: Berechnungszeit der vollständig ausgeführten `ichol`-Funktion für MATLAB-Permutationen im Verhältnis zu denen mit Standard-ANSYS-Permutation

Die allgemeingültige Annahme, dass die `symamd`-Permutation die besten Ergebnisse liefert, kann nicht getroffen werden. Wie das dritte betrachtete Modell in den genannten Diagrammen zeigt, kann diese Permutation sogar zu deutlich schlechteren Ergebnissen als die Cuthill-McKee-Permutation führen. Beim genannten Modell handelt es sich um einen voll volumenvernetzten Balken. Die Speicherplatzvorhersage aus Gleichung 3.13 gilt im verwendeten Bereich für das Modell unabhängig davon, obwohl es sich um ein Modell handelt, das strukturmechanisch der Treibsatzwelle aus Abschnitt 5.1 ähnlich ist. Für die genannte Welle liefert die Cuthill-McKee-Permutation nur für eine der beiden untersuchten

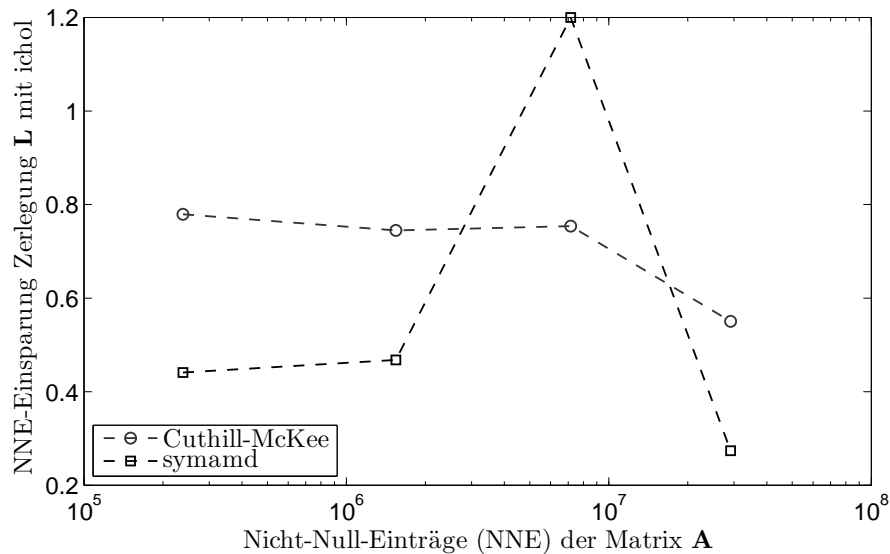


Diagramm 3.7: Anzahl der NNE des chol-Faktors \mathbf{L} für MATLAB-Permutationen im Verhältnis zu der mit Standard-ANSYS-Permutation

Vernetzungsdichten ein besseres Ergebnis als die `symamd`-Permutation. Es ist nach Stand dieser Arbeit nicht vorhersagbar, in welchen Fällen die Anwendung dieser Permutation sinnvoll ist.

Zusammenfassend kann also gesagt werden, dass durch Permutation das Lösen von LGS entscheidend beschleunigt werden kann.

3.4.4. FAZIT

Es wird gezeigt, dass Cholesky- und LU-Faktorisierungen entscheidend zum Lösen von LGS beitragen. Die Empfehlung aus der Literatur, die unvollständigen ‚nofill‘-Faktorisierungen als Vorkonditionierer für iterative Verfahren zu nutzen, ist nicht bestätigt worden, da die iterativen Verfahren erst für nahezu vollständige Zerlegungen konvergieren. Es wird festgestellt, dass die vollständigen und damit auch die nahezu vollständigen Zerlegungen für Matrizen großer Dimension sehr speicherplatzintensive Ergebnisse bei großer Rechendauer liefern. Die in MATLAB implementierte Cholesky-Zerlegung, die definitionsgemäß nur für positiv definite Matrizen ausführbar ist, ist der LU-Zerlegung (MATLAB) aufgrund ihrer Effizienz vorzuziehen. Bei großen Matrizen ist es für die vollständigen Zerlegungen sinnvoll, die MATLAB-Funktionen für `chol` und `lu` zu nutzen. Die Berechnungszeit und der Arbeitsspeicherbedarf kann durch Permutation wesentlich reduziert werden. Hier erzielt man in MATLAB für die verwendeten Systeme mit der `symamd`-Funktion die besten Ergebnisse. Systemmatrizen voll volumenvernetzter Balken und Wellen sind zum Teil durch stark erhöhten Zerlegungsaufwand gekennzeichnet. Vollständige Faktorisierungen können für direkte Lösungsverfahren verwendet werden.

3.5. DIREKTE LÖSUNGSVERFAHREN

Der Hauptvorteil iterativer Lösungsverfahren besteht darin, dass große Systeme mit weniger Arbeitsspeicherbedarf gelöst werden können. Es ist gezeigt worden, dass dieser Vorteil für die verwendeten Systeme im Wesentlichen nicht genutzt werden kann, da die iterativen Verfahren für gutes Konvergenzverhalten reguläre, beinahe komplett zerlegte Koeffizientenmatrizen \mathbf{A} benötigen. Da bei KSM nahezu singuläre Matrizen \mathbf{A} auftreten und der Mehraufwand für vollständige Zerlegungen klein ist, ist es zweckmäßig, trotz der großen Systemdimension direkte Lösungsverfahren zu verwenden.

[FE13] schlägt vor, speziell für KSM direkte Lösungsmethoden zu nutzen, da der große Aufwand für die Faktorisierung durch die Zeiteinsparung bei der Berechnung der einzelnen Moden gegenüber iterativen Verfahren aufgewogen wird. Weiter empfiehlt er die Verwendung parallel arbeitender Zerlegungsalgorithmen, worauf unter 3.6 eingegangen wird und die Nutzung großer Rechnerserver für große Modelle.

Nach [FP10, S.99ff] und [Bör13, S.9] besteht bei direkten Lösungsverfahren durch Fehlerfortpflanzung⁶ von Rundungsfehlern die Gefahr nichtquantifizierbarer Abweichungen. Dies tritt vor allem bei großen LGS auf. Es wird vorgeschlagen, die Fehler durch sogenannte Nachiteration zu minimieren. Die meisten Vorteile der iterativen Verfahren beim Lösen von LGS sind bei der Nachiteration nicht gegeben.

Im Folgenden wird auf die Möglichkeiten der Anwendung direkter Verfahren in MATLAB eingegangen. Auf Nachiteration wird dabei nicht näher eingegangen.

Die in MATLAB hauptsächlich genutzte Funktion zum Lösen von LGS ist der `backslash`- oder `\`-Operator. Nach [FE13] wird dabei auf das in C implementierte UMFPAK-Paket zum Lösen von LGS zurückgegriffen. Nach [HH05, S.123] zieht die Funktion Vorteile aus speziellen Koeffizienten-Matrizen \mathbf{A} . So werden quadratische Matrizen allgemein mit einer LU-Faktorisierung zerlegt. Positiv-definite Matrizen werden mithilfe der Cholesky-Zerlegung faktorisiert und obere sowie untere Dreiecksmatrizen durch Substitution sehr effektiv berechnet, was für die Anwendung im Rahmen dieser Arbeit von Bedeutung ist. Als problematisch wird genannt, dass für nahezu singuläre Matrizen kein Abbruch der Berechnung erfolgt, sondern lediglich eine Warnmeldung ausgegeben wird. Da Rundungsfehler bei der Berechnung von LGS mit nahezu singulären Matrizen sehr groß werden, ist die Güte des Ergebnisses nicht erkennbar.

Bei den KSM wird an einem Entwicklungspunkt für jede Ordnung der Reihenentwicklung bei gleichbleibender Koeffizientenmatrix \mathbf{A} ein Vektor oder mehrere Vektoren \mathbf{x} berechnet. Die direkte Anwendung des `backslash`-Operators in der Form

$$\mathbf{x} = \mathbf{A} \backslash \mathbf{b} \tag{3.15}$$

⁶Direkte Lösungsverfahren sind rekursiv aufgebaut, in die Berechnung eines Eintrages des Vektors \mathbf{x} fließen die zuvor berechneten Einträge ein.

kommt nicht infrage, da die Faktorisierung einer Matrix \mathbf{A} mehrfach ausgeführt würde, was numerisch ineffizient ist. Wird die Faktorisierung dagegen in einem Extraschritt ausgeführt und der `backslash`-Operator in der Form

$$\mathbf{x} = \mathbf{L}_{\text{chol}}^T \backslash \mathbf{L}_{\text{chol}} \backslash \mathbf{b} \quad \text{oder} \quad \mathbf{x} = \mathbf{U}_{\text{lu}} \backslash \mathbf{L}_{\text{lu}} \backslash \mathbf{b} \quad (3.16)$$

ausgeführt, so können die Vorteile direkten Lösens von LGS gut ausgenutzt werden. Der `\`-Operator führt dann eine sehr effektive Substitution durch. Diese Möglichkeit wird im Rahmen der MOR-Implementierung verwendet. Diese wird unter 4.5 erläutert.

3.6. VERWENDUNG EXTERNER GLEICHUNGSSYSTEM-LÖSER

Wie unter 3.5 erwähnt, ist es empfehlenswert, auf alternative LGS-Löser, die nicht in MATLAB verfügbar sind, zurückzugreifen. Im Rahmen dieser Arbeit werden solche Löser nicht verwendet. Es empfiehlt sich für die Anwendung auf Systeme mit einem Freiheitsgrad $N > 10^6$ auf diese Löser zurückzugreifen⁷. Hier sollen Möglichkeiten aufgezeigt werden, solche Systeme in hinreichender Zeit zu lösen.

[FE13] stellt die in MATLAB verwendete UMFPACK-Bibliothek der frei erhältlichen MUMPS-Bibliothek gegenüber. Die wesentlichen Vorteile dieser Bibliothek gegenüber MATLAB bestehen in einer ‚out-of-core‘ -Option. Diese erlaubt es, den Arbeitsspeicher durch effektive Auslagerung auf die Festplatte zu entlasten und ermöglicht somit die Berechnung sehr großer Modelle. UMFPACK ist in der Programmiersprache Fortran verfügbar. Die Einbindung dieser Bibliothek in MATLAB ist auf verschiedene Arten möglich.

In [Ste12, S.309ff] wird beschrieben, wie mithilfe von ‚mex-Files‘ C-Funktionen in MATLAB eingebunden werden können, wobei auch eine Parameterübergabe-Routine existiert. UMFPACK ist dazu erst in C zu übersetzen.

Eine weitere Möglichkeit besteht darin, UMFPACK über ein ausführbares Programm zu verwenden, welches von MATLAB aus gestartet wird. Dazu werden die Eingabedaten von MATLAB und die Ausgabedaten von genanntem Programm an einen definierten Ort auf der Festplatte gespeichert, von wo aus sie jeweils geladen werden.

Ein Problem bei beiden Möglichkeiten besteht, wenn das Reduktions-Verfahren nicht komplett ausgelagert werden soll. Für diesen Fall muss die berechnete Zerlegung der Koeffizientenmatrix \mathbf{A} zwischengespeichert werden. Eine Rückgabe der Faktoren an MATLAB ist nicht sinnvoll, da dies für große Modelle den Arbeitsspeicher überlastet. Zusätzlich muss gewährleistet werden, dass für eine neue Matrix \mathbf{A} erneut eine Faktorisierung durchgeführt wird und nicht auf die gespeicherten Faktoren zurückgegriffen wird.

⁷Matlab ist (im Gegensatz zu anderen Berechnungsprogrammen), nicht in der Lage, Daten effektiv auf die Festplatte auszulagern. Die maximale Größe faktorisierbarer Matrizen ist demnach abhängig von der Größe des Arbeitsspeichers.

Weiter ist von Bedeutung, dass für rationale KSM die Besetzungs-Struktur der Koeffizientenmatrizen \mathbf{A} und damit auch die Besetzungsstruktur ihrer Zerlegungen erhalten bleiben. Es ist zu untersuchen, ob aus dem Vorliegen der Besetzungsstruktur der Zerlegungen ein Rechenzeitgewinn erzielt werden kann.

3.7. ZUSAMMENFASSUNG

Das Lösen von LGS als eine Grundaufgabe der Numerik wird dargestellt, wobei zwischen direkten und iterativen Lösungsverfahren unterschieden wird. Um die bei KSM auftretenden singulären Koeffizientenmatrizen unter näherungsweise Beibehaltung der Systemeigenschaften lösen zu können, wird eine Diagonalperturbation verwendet. Das iterative Lösen der Gleichungssysteme ist aufgrund des Zusammenhangs der Verfahrensdivergenz mit Singularitätsnähe nicht empfehlenswert, jedoch potentiell zur Nachiteration direkter Lösungen verwendbar. Faktorisierungsverfahren werden vorgestellt, wobei die in MATLAB verfügbaren, vollständigen und unvollständigen LU- und Cholesky-Zerlegungen untersucht werden. Dabei ist für positiv definite Koeffizientenmatrizen die Cholesky-Zerlegung aufgrund geringem Speicherbedarf und kürzerer Rechenzeit vorzuziehen. Bei rationalen SOAR-Verfahren ist aufgrund der Negativ-Definitheit der Koeffizientenmatrix die Cholesky-Zerlegung nicht möglich, weshalb die LU-Zerlegung zu verwenden ist. Speziell für die MATLAB-Funktionen `chol` und `lu` ist zur Verkürzung von Rechendauer und Speicherbedarf Permutation sinnvoll. Es liefert die `symamd`-Permutation und für einige volumenvernetzte, stabförmige Strukturen die Cuthill-McKee-Permutation gute Ergebnisse. Vollständige Zerlegungen sind der Hauptrechenschritt direkter Lösungsverfahren, jedoch durch die Kapazität des Arbeitsspeichers begrenzt. Der Speicherbedarf steigt mit der Systemdimension exponentiell an. Für Systeme sehr großer Dimension ist es aus diesem Grund sinnvoll, LGS mithilfe externer Bibliotheken zu lösen, die effektiv mit auf die Festplatte ausgelagerten Faktorisierungsergebnissen arbeiten können. Die implementierten LGS-Lösungsfunktionen werden unter 4.5 erläutert.

4. IMPLEMENTIERUNG

In diesem Kapitel wird die Anwendung der MOR innerhalb des Softwarepakets MORPACK erläutert. Dazu wird der grundsätzliche Aufbau der Software erklärt und speziell auf die Umgebung der MOR eingegangen. Die einzelnen implementierten Krylov-Verfahren werden mit ihren vom Benutzer definierbaren Optionen und deren Einfluss auf die MOR erläutert. Die implementierten Algorithmen werden in die unter Abschnitt 2 genannten Verfahrensgruppen der Krylov-MOR eingeordnet. Es wird auf Besonderheiten der numerischen Umsetzung auf die MOR eingegangen. Es werden die Fehlerbewertungskriterien erläutert. Weiter wird die Anwendung der Lösungsalgorithmen linearer Gleichungssysteme auf der Ebene der Programmierung erläutert.

4.1. AUFBAU VON MORPACK

Das Modellordnungsreduktions-Paket (MORPACK) wird an der Professur für Dynamik und Mechanismentechnik der Technischen Universität Dresden entwickelt. In diesem Abschnitt wird kurz der grundlegende Aufbau der Software erläutert. Dabei wird gesondert auf den Aufbau des MOR-Moduls eingegangen, um darauf aufbauend die Anwendung der implementierten Krylov-Reduktionsmethoden zu erläutern. In Abbildung 4.1 wird der modulare Grundaufbau von MORPACK dargestellt. Dabei werden die Module, die für diese Arbeit von Bedeutung sind, hervorgehoben.

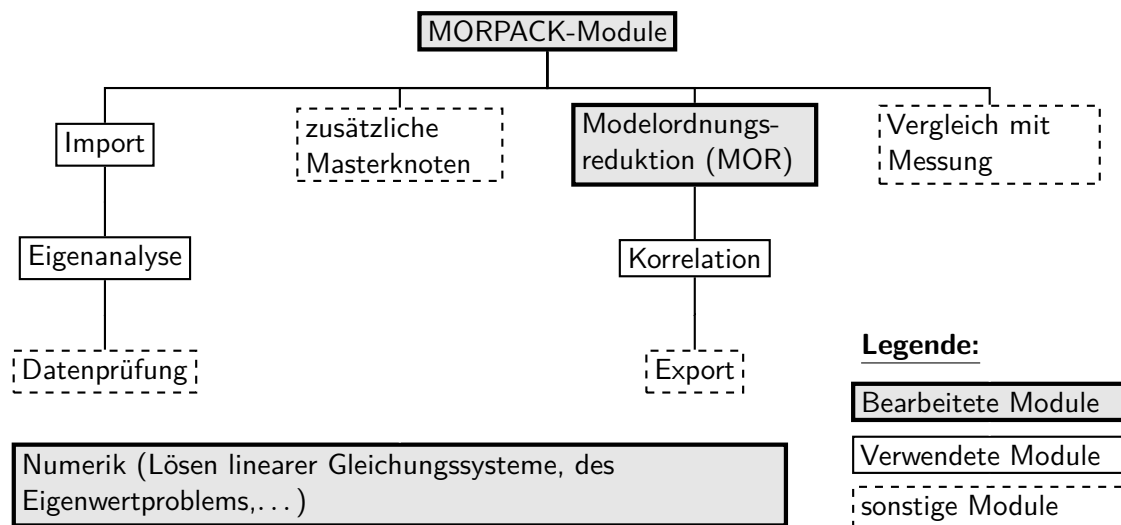


Abbildung 4.1.: Aufbau von MORPACK in Modulen und deren Bedeutung für die Implementierung der KSM

Die KSM sind Teil des Modellordnungsreduktions-Moduls. Die verwendeten Modelle werden über den `Import` aus der FE-Software importiert. Um die Güte des Reduktionsergebnisses beurteilen zu können, werden die Korrelationskriterien aus Abschnitt 2.6 verwendet, die dem `Korrelations`-Modul zuzuordnen sind. Alle Korrelationskriterien greifen auf das Ergebnis der `Eigenanalyse` zurück.

4.2. ANFORDERUNGEN AN REDUKTIONSFUNKTIONEN

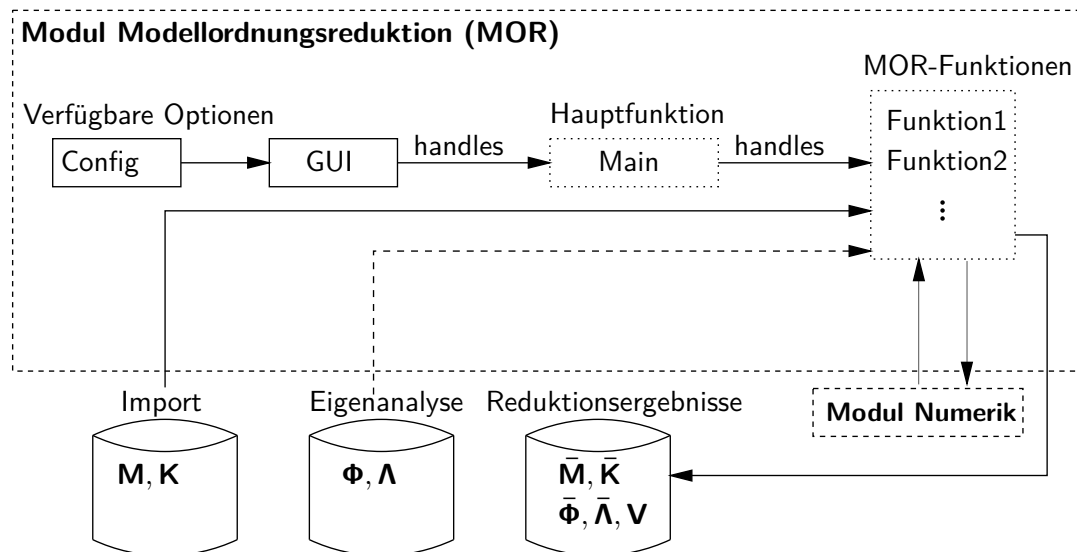


Abbildung 4.2.: Grundgerüst des MOR-Moduls mit Reduktionsfunktionen

Das Grundgerüst, in das Reduktionsfunktionen eingebunden werden, ist in Abbildung 4.2 dargestellt. Die `config`-Datei enthält alle verfügbaren Optionen der Reduktionsfunktionen. Die vom Benutzer getätigten Eingaben werden in der Strukturvariable `handles` an die Hauptfunktion weitergegeben, welche die gewählte MOR-Funktion aufruft. Die MOR-Funktion greift u.a. auf die vollen Systemmatrizen aus dem `Import`-Ordner zu und verwendet Numerikfunktionen, z.B. zum Lösen linearer Gleichungssysteme. Einige MOR-Funktionen verwenden Ergebnisse der Eigenanalyse. Die Reduktionsergebnisse werden gespeichert. Die Pfade aller Ordner, aus denen Daten geladen oder in die Daten gespeichert werden, sind in der Strukturvariable `handles` gespeichert.

Die `config`-Datei mit den verfügbaren Optionen ist ein `cell-Array` und enthält folgende Informationen jeder Reduktionsmethode:

- Name der Reduktionsmethode
- Bezeichnung der zugehörigen MATLAB-Funktion
- Kurzbeschreibung der Reduktionsmethode

- Optionen der Reduktionsmethode in einem `cell-Array` folgenden Aufbaus für jede Option:
 - Optionsname
 - Name der Variable (`OptionsName`), die die gewählte Einstellung beinhaltet
 - Kurzbeschreibung der Option
 - Optionstyp und ggf. Wahlmöglichkeiten:
 - * An-Aus-Schalter
 - * Dropdown-Liste
 - * editierbarer Zahlenwert und Einschränkungen¹
 - Standardwert der Option
 - Abhängigkeit des Erscheinens der Option von der Wahl anderer Optionen²

Durch die Angabe des Funktionsnamens in der `config`-Datei kann die MOR-Funktion von der Hauptfunktion ausgeführt werden. Die Funktion wird dabei mit

```
cFlag = fReduktionsfunktion(handles)
```

aufgerufen, wobei die Eingabe-Strukturvariable `handles` unter anderem die vom Benutzer auf der Grafischen Benutzeroberfläche (GUI) getätigten Optionen nach dem Muster

```
handles.OptionsMor.OptionsName
```

enthält. Reduktions-Funktionen geben lediglich die Information über den Erfolg der Reduktion in `cFlag` direkt an die Hauptfunktion zurück und müssen diesen Richtlinien genügen:

- Verwendung einer benutzerdefinierten Funktion zum Lösen linearer Gleichungssysteme nach Abschnitt 4.5
- Optionale Verwendung bzw. Reduktion einer Dämpfungsmatrix \mathbf{D}
- Information über den Reduktions-Fortschritt

Weiter ist auf einen Arbeitsspeicherbedarf reduzierenden und effektiven Programmablauf zu achten.

¹Zahlenwerte können auf Datentypen, wie ganzzahlige Werte, eingeschränkt werden. Weiter ist ein möglicher Wertebereich definierbar.

²Das Erscheinen kann über eine ODER-Verknüpfung von mehreren anderen Optionen abhängen.

4.3. EIGENSCHAFTEN UND OPTIONEN DER KSM-FUNKTIONEN

MORPACK enthält eine auf Algorithmus 2.3 aufbauende KSM-Funktion. Diese Funktion wird im Zuge dieser Arbeit erweitert. Es sind zwei weitere Reduktionsmethoden implementiert, eine rationale Block-SOAR- und eine rationale, interpolative SOAR-Funktion. Die Eigenschaften und Optionen dieser Funktionen werden in diesem Abschnitt erläutert.

4.3.1. ARNOLDI-FUNKTION ERSTER ORDNUNG

Die implementierte Arnoldi-Funktion ist um die Funktionalitäten:

- Eliminierung der Starrkörperbewegung im reduzierten System durch Orthogonalität der Spalten der Projektionsmatrix \mathbf{V} zu den Starrkörpermoden,
- wahlweise Interpolation des Startvektors \mathbf{b} aus der Eingangsmatrix \mathbf{B} ,
- Verbesserung der Orthogonalität der Krylov-Moden in der Projektionsmatrix \mathbf{V} durch einfache Reorthogonalisierung und
- wahlweise Verwendung einer generellen Diagonalperturbation nach Abschnitt 3.2.2

erweitert worden. Die vom Benutzer wählbaren Optionen sind in Tabelle 4.1 angegeben. Eine Diagonalperturbation ist bei freien Systemen zu wählen, um das entstehende

Tabelle 4.1.: Optionen der Arnoldi-Funktion erster Ordnung

Option	Wahlmöglichkeit	Beschreibung	auto
Wahl des Startvektors	nach Kou09	Startvektor aus Steifigkeitsmatrix herausgegriffen (nach [Kou09])	
	interpolativ	Gleichwichtung der Spalten der Eingangsmatrix \mathbf{B} auf Startvektor	x
Art der Wahl der Diagonalperturbation	keine	System wird nicht diagonal perturbiert	
	automatisch	Start ohne, bei Abbruch schrittweise Erhöhung der Diagonalperturbation bei Verfahrensabbruch	x
	manuell	manuelle Eingabe des Wertes	
Wert der Diagonalperturbation (nur bei manueller Wahl)	$0 \leq \text{Wert} \leq 1$	numerische Verstimmung des Systems zum Erreichen der Lösbarkeit	10^{-9}

Gleichungssystem trotz Unterbestimmtheit, wie unter 3.2.2 aufgeführt, zu lösen.

4.3.2. RATIONALE SOAR-FUNKTIONEN

Es sind zwei Funktionen, eine rationale Block-SOAR- und eine rationale, interpolative SOAR-Funktion nach Abschnitt 2.5 implementiert worden. Der wesentliche Unterschied der Funktionen besteht darin, dass beim Block-SOAR-Verfahren für jeden Lösungsschritt eine Krylov-Mode für jeden Eingang generiert wird. Die reduzierte Systemdimension ergibt sich aus Gleichung 2.11 und ist damit ein Vielfaches der Eingangsanzahl. Beim interpolativen

SOAR wird ein Startvektor aus der Eingangsmatrix \mathbf{B} interpoliert und damit nur eine Krylov-Mode je Lösungsschritt erzeugt. Der weitere Funktionsaufbau ist identisch. Auf eine Zusammenlegung der Funktionen ist aufgrund des großen Dimensionsunterschiedes der blockweise und interpolativ reduzierten Systeme bei gleicher Optionenwahl (nach Tabelle 4.2) verzichtet worden.

Tabelle 4.2.: Optionen der rationalen SOAR-Funktionen

Option	Wahlmöglichkeit	Beschreibung	auto
Anzahl der Entwicklungsstellen	≥ 1	Anzahl der Entwicklungsstellen der Reihenentwicklung	1
Art der Wahl der Entwicklungsstellen	linear verteilt	Entwicklungsstellen werden linear auf abzubildenden Frequenzbereich verteilt, Start- und Endwert inklusive	x
	Eigenfreq. verteilt	linear auf Nummer der Eigenfrequenzen innerhalb des abzubildenden Frequenzbandes verteilt	
Startfrequenz des Frequenzbereiches	$\geq 0^{1/s}$	kleinste Frequenz des Frequenzbereiches, bei Wert 0 immer Entwicklungsstelle	0
Endfrequenz des Frequenzbereiches	$\geq 0^{1/s}$	größte Frequenz des Frequenzbereiches	0
Ordnung der Reihenentwicklung	≥ 1	Ordnung der Reihenentwicklung der Frequenzstützstellen	1
Deflationstoleranz	$0 \leq \text{Wert} \leq 1$	Grenze linearer Abhängigkeit der Krylov-Moden	$10 \cdot \text{eps}^{(3)}$
zusätzliche Option der rationalen, interpolativen SOAR-Funktion			
Art der Interpolation der Eingänge	Gleichwichtung	Gleiche Wichtung der Eingänge auf Startvektor \mathbf{b}	x
	zufällig	zufällige Wichtung der Eingänge auf Startvektor \mathbf{b}	
	Import	manuell erzeugter und übergebener Wichtungsvektor	

Die rationalen SOAR-Funktionen haben die Funktionalitäten:

- Wahlmöglichkeit einer oder mehrerer Frequenzstützstellen der Reihenentwicklung in einem definierbaren Frequenzbereich⁴,
- Wählbarer Deflationstoleranz *deftol*,
- Eliminierung der Starrkörperbewegung im reduzierten System durch Orthogonalität der Spalten der Projektionsmatrix \mathbf{V} zu den Starrkörpermoden und
- Verbesserung der Orthogonalität der Krylov-Moden in der Projektionsmatrix \mathbf{V} durch einfache Reorthogonalisierung.

Die rationale, interpolative SOAR-Funktion hat zusätzlich die Funktion:

- wählbare Interpolationsart der Eingangsmatrix \mathbf{B} auf einen Startvektor \mathbf{b} .

³Der Wert *eps* steht für die Maschinengenauigkeit.

⁴Die Frequenzstützstellen können im Frequenzbereich linear verteilt oder nahe an Eigenfrequenzen gewählt werden.

Die vom Benutzer definierbaren Optionen der beiden Funktionen sind in Tabelle 4.2 angegeben. Die optimale Wahl der Frequenzstützstellen ist der Kern der rationalen SOAR-Verfahren und Ziel der iterativen rationalen Verfahren.

Es ist zu erwarten, dass einige implementierte Optionsmöglichkeiten nicht zu sinnvollen Ergebnissen führen, weshalb Versionen der jeweiligen Funktionen beigefügt werden, die auf den Erkenntnissen aus 5 begründet nur zielführende Optionen enthalten.

4.4. KORRELATIONSKRITERIEN

Es werden verschiedene, zum Teil im Zuge dieser Arbeit implementierte oder modifizierte Korrelationskriterien und Darstellungsformen verwendet und hier erläutert.

4.4.1. EIGENFREQUENZBEZOGEN

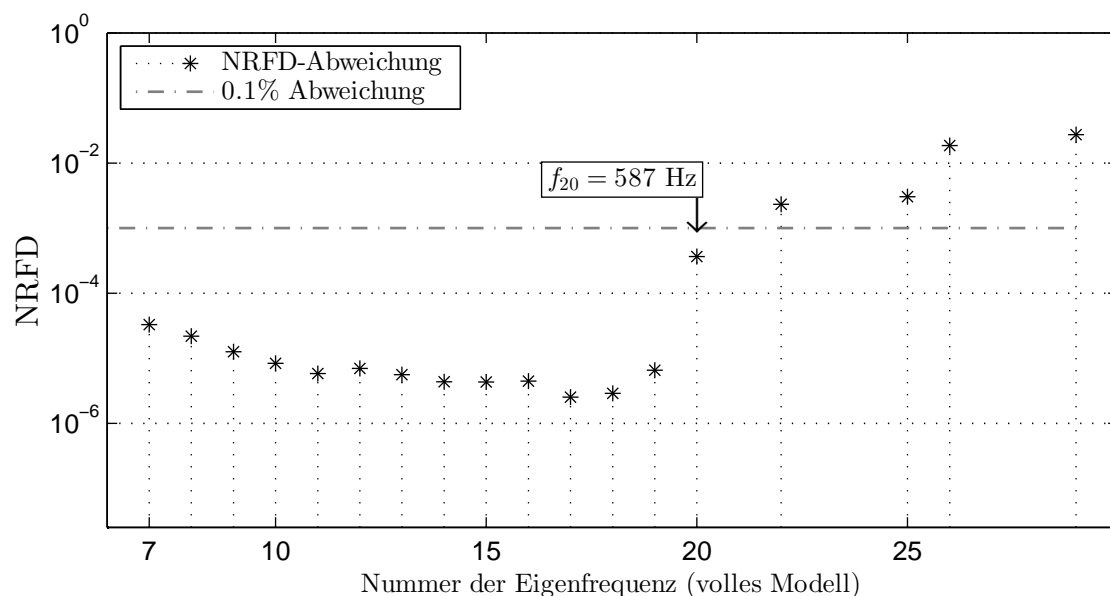


Diagramm 4.1: NRFD-Korrelationskriterium dargestellt in einem Diagramm mit Fehlerschranke und letzter Eigenfrequenz unterhalb der Fehlerschranke

Es wird das eigenfrequenzbezogene Normalized Relative Frequency Difference (NRFD)-Kriterium aus Abschnitt 2.6.1 verwendet. Diagramm 4.1 zeigt die entsprechende Darstellungsform. Es werden die relativen Abweichungen der Eigenfrequenzen des reduzierten Modells im Vergleich zum vollen Modell logarithmisch dargestellt. Dabei entspricht der Fehler $NRFD = 10^0 = 1$ einer Abweichung von 100%. Es wird eine Fehlerschranke, hier $NRFD = 10^{-3} \hat{=} 0.1\%$ dargestellt. Es wird die letzte der zusammenhängenden abgebildeten Eigenfrequenzen des vollen Modells unterhalb dieser Schranke angezeigt. Auf der x-Achse sind die Nummern der Eigenfrequenzen des vollen Modells dargestellt. Beginnen diese, wie in Diagramm 4.1, mit Nummer 7, so beginnt die Darstellung mit dem NRFD

der ersten Eigenfrequenz des reduzierten Modells mit der siebten des vollen Modells. Das entspricht in diesem Fall der ersten Nichtstarrkörpereigenfrequenz.

4.4.2. EIGENVEKTORBEZOGEN

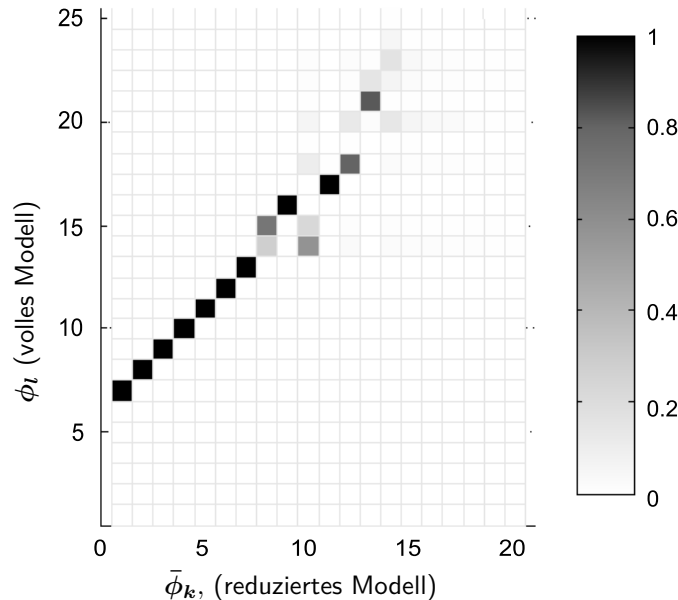


Diagramm 4.2: Korrelationskriterium modMAC der Eigenvektoren eines reduzierten Systems mit denen des vollen Systems

In wenigen Fällen wird ein eigenvektorbezogenes Korrelationskriterium aus Abschnitt 2.6.2 verwendet. Ein entsprechendes modMAC ist in Diagramm 4.2 angegeben. Es werden die Eigenvektoren des reduzierten mit denen des vollen Systems verglichen. Werte nahe Eins zeigen eine hohe Übereinstimmung. Im Diagramm entspricht der erste Eigenvektor des reduzierten Systems dem siebten Eigenvektor des vollen Systems und es werden sieben aufeinanderfolgende Eigenvektoren sehr gut abgebildet.

4.4.3. ÜBERTRAGUNGSFUNKTIONSBEZOGEN

Hauptsächlich werden übertragungsfunktionsbezogene Kriterien aus Abschnitt 2.6.3 verwendet, da diese die Informationen der Eigenfrequenzen und Eigenmoden für die Systemein- und -ausgänge in übersichtlich darstellbarer Form beinhalten. Diagramm 4.3 zeigt die verwendeten Darstellungsformen übertragungsfunktionsbezogener Korrelationskriterien. Das obere Teildiagramm zeigt die Frobeniusnorm der Übertragungsfunktionsmatrizen eines vollen und reduzierten Systems $\|\mathbf{H}(s)\|_F$ und $\|\bar{\mathbf{H}}(s)\|_F$ nach Gleichung 2.24. Die Frobeniusnorm der relativen Abweichung des reduzierten Systems zum vollen System ϵ_{HF}^{rel} nach Gleichung 2.26 ist im unteren Teildiagramm dargestellt. Ein Fehler von $\epsilon_{HF}^{rel} = 10^0 = 1$ entspricht 100% Abweichung. Die Fehlergrenze $\epsilon_{HF} = 5 \cdot 10^{-2} = 0.05 \hat{=} 5\%$ wird zur

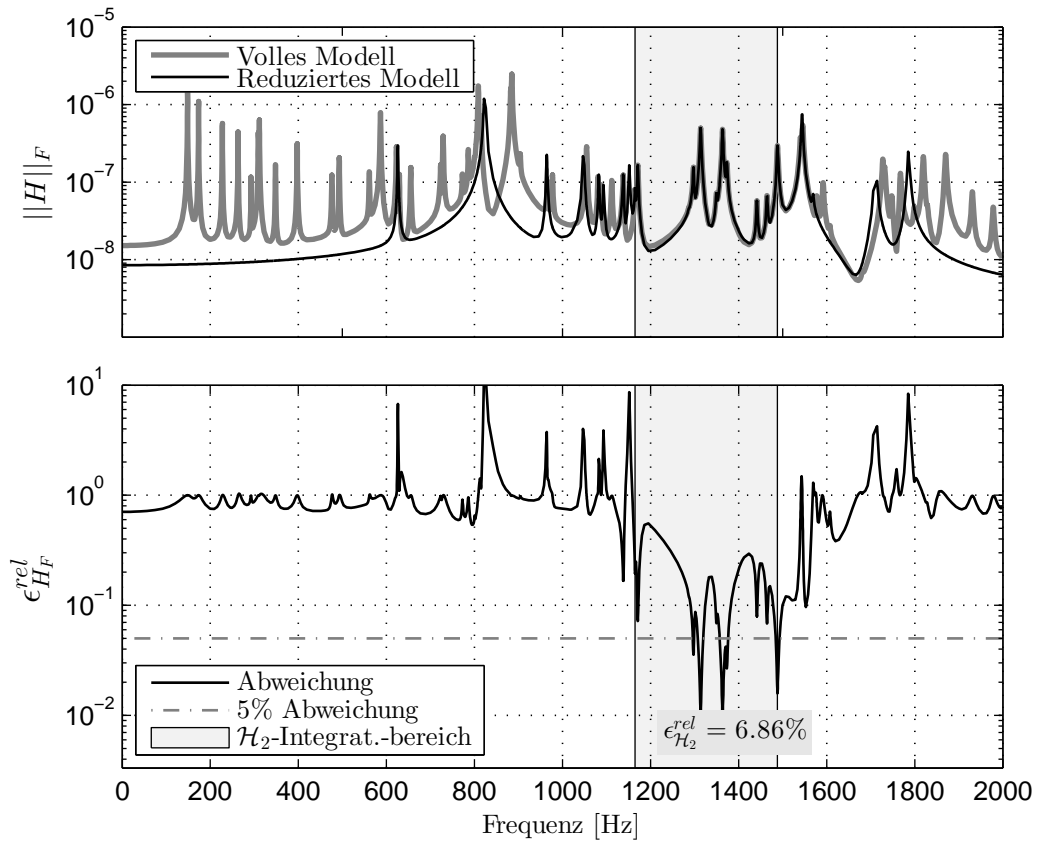


Diagramm 4.3: Frequenzgangbezogene Korrelationskriterien (frequenzabhängige Frobeniusnorm der Übertragungsfunktionsmatrizen und deren relativer Abweichung sowie \mathcal{H}_2 -Norm) für übereinstimmenden Systemverhalten in mittlerem Frequenzbereich (\mathcal{H}_2 -Integrationsbereich aus Übereinstimmung der Eigenfrequenzen)

besseren Orientierung angegeben. Es wird weiter der relative Fehler der \mathcal{H}_2 -Norm⁵ $\epsilon_{\mathcal{H}_2}^{rel}$ in % nach Gleichung 2.29 und der verwendete Integrationsbereich in Grau angegeben. Der \mathcal{H}_2 -Integrationsbereich ergibt sich aus der Übereinstimmung der Eigenfrequenzen.

Im Anhang unter A.2 wird der Einfluss der Massennormierung nach Gleichung 2.22 im Vergleich zu Normierung des Eingangsanteils der Eigenmoden nach Gleichung 2.23 diskutiert.

⁵Wie erwähnt, gilt die Definition $\epsilon_{\mathcal{H}_2}^{rel} = \epsilon_{\tilde{\mathcal{H}}_2}^{rel}$, da lediglich die $\tilde{\mathcal{H}}_2$ -Norm mit Integration über einen Frequenzbereich verwendet wird.

4.5. LÖSUNGSFUNKTIONEN LINEARER GLEICHUNGSSYSTEME

Es wird der im Zuge dieser Arbeit entwickelte Aufbau der LGS-Löser-Funktionen in MORPACK erläutert. Die implementierten Funktionen, deren Optionen und Entscheidungshilfen zum Einsatz werden aufgezeigt und Hinweise zur Implementierung weiterer Funktionen gegeben.

4.5.1. ANFORDERUNGEN UND AUFBAU

Im modular aufgebauten MORPACK bestehen verschiedene Anforderungen an LGS-Löser-Funktionen. Sie ergeben sich aus der Anwendung auf alle implementierten MOR und die Bearbeitung durch mehrere Autoren. Die Anforderungen können folgendermaßen zusammengefasst werden:

- Verwendbarkeit für alle Arten auftretender linearer LGS
 - Anwendung auf Vektoren \mathbf{b} und Matrizen \mathbf{B} als rechte Seite
 - Verwendung streng positiv-definiten und positiv-semidefiniten Koeffizienten-Matrizen \mathbf{A}
 - Verwendung der gleichen Koeffizienten-Matrix \mathbf{A} in verschiedenen LGS innerhalb einer MOR-Funktion, also Wiederverwendung von z.B. Faktorisierungen
- Beeinflussbarkeit des LGS-Lösers durch Optionen, die auf einer Grafischen Benutzeroberfläche (GUI) getätigt werden
- Beliebige Erweiterbarkeit durch weitere LGS-Löser ohne genaue Kenntnis und Bearbeitung der bereits vorhandenen Löser-Funktionen

Um diese Anforderungen zu erfüllen, wird ein Grundgerüst für Funktionen zum Lösen von LGS entwickelt, das diese Anforderungen erfüllt. Das Grundgerüst ist in Algorithmus 4.1 angegeben.

Unter anderem werden Namen, Funktionsnamen und Optionen in einer Konfigurationsdatei gespeichert, die beliebig erweitert werden kann und die alle Informationen über die LGS-Löser enthält. Diese Konfigurations-Datei wird von der GUI verwendet, auf der der Benutzer einen LGS-Löser auswählt und Optionen definiert.

4.5.2. VERWENDUNG DER GLEICHUNGSSYSTEM-LÖSER

Wie erläutert, ist grundsätzlich jede Gleichungslöser-Funktion in zwei Teile unterteilt, in die Vorkonditionierung und das Lösen selbst. Welcher der beiden Teile ausgeführt wird, hängt von der Anzahl der Eingabeparameter ab. Welche Parameterwahl welche Operation nach sich zieht, ist in Tabelle 4.3 zusammengefasst.

Algorithmus 4.1 : Gerüst für Funktionen zum Lösen linearer Gleichungssysteme in Matlab
Eingabe : Koeffizientenmatrix **A**, Optionen beinhaltende Strukturvariable *handles*, Rechte Seite **B**,
 Cell-Array mit Vorkonditionier-Ergebnis *Preconditioner*
Ausgabe : Ergebnis **X**, *Preconditioner*, Cell-Array für Fehlermeldungen *Flag*

```

function [X, Preconditioner, Flag] = SolverLse(A, handles, B, Preconditioner)
1 AnzEingang = (Anzahl der Eingangsvariablen)
2 OptionenPrekond = handles.optionsLse...
3 OptionenLoeser = handles.optionsLse...
4 if AnzEingang = 2 OR AnzEingang = 3 then           % wenn Vorkonditionierung erforderlich
5     [Preconditioner  Flag] = Faktorisierung(A, OptionenPrekond)
6     if Flag ≠ 0 then                               % wenn Vorkonditionierung fehlgeschlagen
7         | return
8     end
9 end
10 if AnzEingang = 3 OR AnzEingang = 4 then        % wenn Lösen des LGS erwünscht
11     [X  Flag] = Loeser(A, B, Preconditioner, OptionenLoeser)
12     if Flag ≠ 0 then                               % wenn Löser fehlgeschlagen
13         | return
14     end
15 end
end

```

Tabelle 4.3.: Anwendung der Lösungs-Funktionen für LGS in MORPACK nach Algorithmus 4.1

Eingabeparameter	Ausgabeparameter	ausgeführte Operation
A , <i>handles</i>	X = 0, <i>Preconditioner</i> , <i>Flag</i>	Nur Vorkonditionierung (in den meisten Fällen Faktorisierung)
A , <i>handles</i> , B	X , <i>Preconditioner</i> , <i>Flag</i>	Vorkonditionierung und Lösen
A , <i>handles</i> , B , <i>Preconditioner</i>	X , <i>Preconditioner</i> , <i>Flag</i>	Nur Lösen

Es ist also möglich, die LGS-Löser auf folgende Art und Weise zu verwenden:

- Vollständiges, einmaliges Lösen eines LGS
 - Drei Eingabeparameter, Unterdrückung der Ausgabe von *Preconditioner*
 - Verwendung analog zu MATLAB `backslash`-Operator
- Vollständiges Lösen eines LGS mit Beschleunigung späterer Berechnungen
 - Erstes LGS: Drei Eingabeparameter, alle Ausgabeparameter
 - Weitere LGS: Vier Eingabeparameter, nur bei gleicher Koeffizienten-Matrix **A** möglich, Unterdrückung der Ausgabe von *Preconditioner*
 - Vorteil: Einmalige Faktorisierung bei schneller Lösung vieler LGS
- Vorkonditionierung und Lösen vieler LGS getrennt
 - Vorkonditionierung: Zwei Eingabeparameter, Unterdrückung der Ausgabe von **X**

- LGS: Vier Eingabeparameter, nur bei gleicher Koeffizienten-Matrix **A** möglich, Unterdrückung der Ausgabe von *Preconditioner*
- Vorteil: Einmalige Faktorisierung, schnelle Lösung vieler LGS auch in Schleifen

Bei der Anwendung ist darauf zu achten, dass eine Ausgabe *Preconditioner* nur als Eingabe für ein LGS genutzt werden darf, wenn es sich um die identische Koeffizientenmatrix **A** handelt. Weiter ist nach der Ausführung der LGS-Löser-Funktionen der erste Eintrag des cell-Arrays *Flag* zu überprüfen. Beträgt dieser Wert Null, so ist das LGS erfolgreich gelöst worden, bei Wert Eins ist ein Fehler aufgetreten und die Berechnung darf nicht fortgeführt werden.

Welche LGS-Löser zur Verfügung stehen, ist in MORPACK in einer Konfigurations-Datei hinterlegt. Diese enthält Namen, Funktionsnamen, Beschreibungen und Optionen aller möglichen Löser-Funktionen. Der Benutzer kann dabei auf einer GUI einen Löser und dessen Optionen auswählen. An die Funktion, in der ein LGS auftritt, wird der Name des gewünschten Löser als string-Variable `sSolverLse` und die Optionenauswahl unter der Variablenbezeichnung `handles.optionsLSE` (Strukturvariable) übergeben. Um die gewünschte Funktion auszuführen, ist der string `sSolverLse` in eine Funktion zu wandeln und wie beschrieben auszuführen.

Die Optionen der implementierten LGS-Funktionen sind in Tabelle 4.4 angegeben. Es sind die LGS-Funktionen

1. Direkter Solver

- Lösen über MATLAB `backslash`-Operator

2. LU + Direkter Solver

- vollständige LU-Faktorisierung mit MATLAB `lu` der Koeffizientenmatrix **A** und direktes, substitutives Lösen über MATLAB `backslash`-Operator

3. Chol + Direkter Solver

- vollständige Cholesky-Faktorisierung mit MATLAB `chol` der Koeffizientenmatrix **A** und direktes, substitutives Lösen über MATLAB `backslash`-Operator

4. (I)Chol + iterativer pcg-Solver

- unvollständige Cholesky-Faktorisierung über MATLAB `ichol` und iteratives Lösen mit `pcg`-Solver
 - Optionenwahl gemäß Vorschlag der Tabelle 4.4
- vollständige Cholesky-Faktorisierung über MATLAB `chol`, direktes Lösen und Nachiteration zur Minimierung von Rundungsfehlern mit `pcg`-Solver
 - für vollständige Faktorisierung `droptol = 0` und `diagpert = 0`
 - kleine maximale Iterationsschrittzahl < 10

implementiert.

Tabelle 4.4.: Übersicht über die Optionen der einzelnen LGS-Löser-Funktionen

Option	Direkter Solver (1)	LU + Direkter Solver (2)	Chol + Direkter Solver (3)	(I)Chol + Iterativer pcg-Solver (4)
Permutationsart		×	×	×
Droptoleranz				×
Diagonalperturbation				×
Solver-Fehlertoleranz				×
Maximale Anzahl Iterationsschritte				×

Option	Wahlmöglichkeit	Beschreibung	auto
Permutationsart	auto	Nur bei Solver (2) und (3), integrierte Permutation der MATLAB chol- oder lu-Funktion	(2,3)
	symamd	Kurze Berechnungszeit bei kleinem Arbeitsspeicherbedarf für die meisten Modelle	(4)
	Cuthill-McKee	Zum Teil kürzere Berechnungszeit bei volumenvernetzten, stabförmigen Modellen	
Droptoleranz	$0 \leq droptol \leq 1$	Minimum einzelner Einträge der Faktoren bei Faktorisierung	(2,3): 0 (4): 10^{-5}
Diagonalperturbation	$0 \leq diagpert \leq 1$	Störung der Koeffizientenmatrix zur Verbesserung des Konvergenzverhaltens bei iterativem Lösen	(3): 0 (4): 10^{-4}
Solver-Fehlertoleranz	$0 \leq tol \leq 1$	Erlaubter Fehler bei iterativem Lösen für Iterationsabbruch	10^{-6}
Maximale Anzahl Iterationsschritte	> 0	Maximale Iterationsschrittzahl bei iterativem Lösen oder Nachiterieren	100

Alle LGS-Löser-Funktionen mit integrierter Faktorisierung verwenden Permutation zur Minimierung des Speicherbedarfes und der Faktorisierungszeit. Wie in Abschnitt 3.4.3 erläutert, liefert die MATLAB symamd-Permutation für die meisten mechanischen Systeme die besten Ergebnisse, die in die MATLAB-Funktionen chol und lu integrierte Permutation ähnelt dieser. Weiter wird erläutert, dass sich die Faktorisierung von Systemmatrizen volumenvernetzter, stabförmiger Modelle als rechentechnisch sehr aufwändig erweist. Zum Teil liefert hier die Cuthill-McKee-Permutation (MATLAB-symrcm) die besten Ergebnisse.

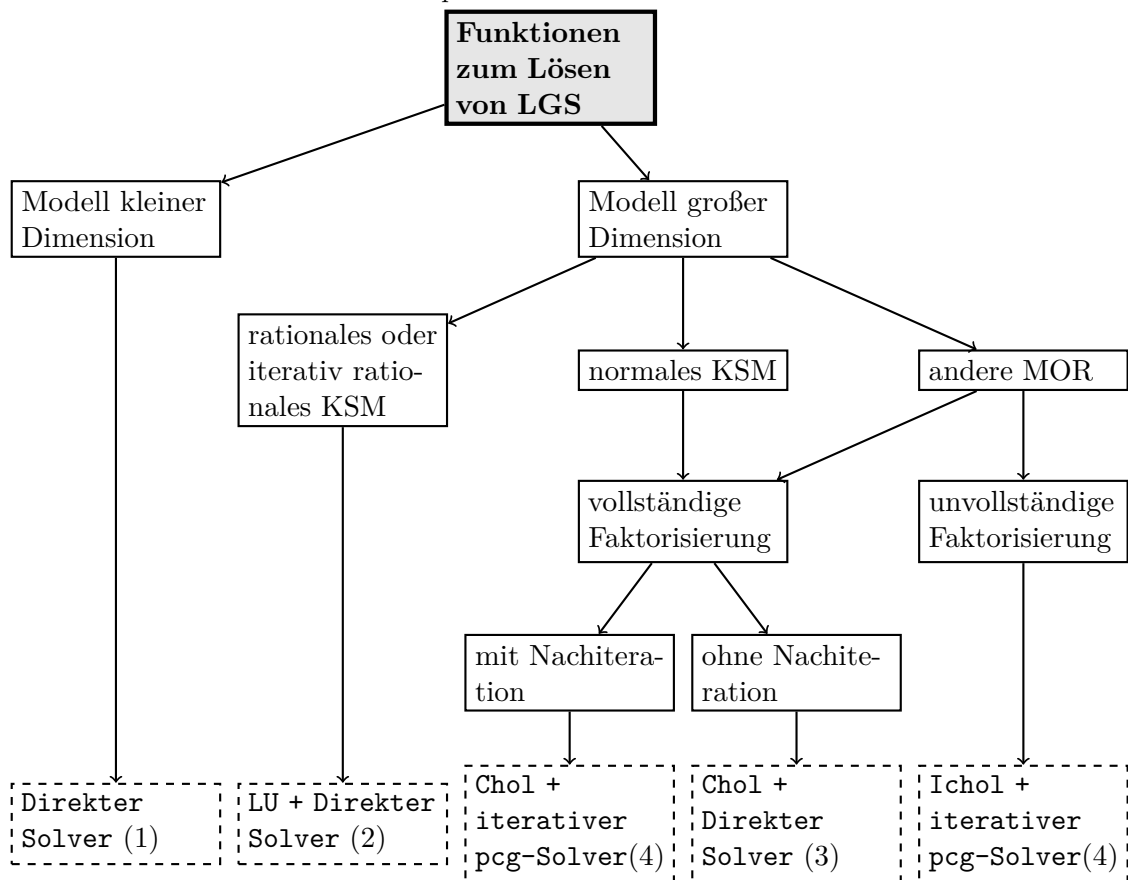
Abbildung 4.3 zeigt Entscheidungskriterien zur Wahl eines geeigneten LGS-Verfahrens.

4.5.3. HINWEISE ZUR IMPLEMENTIERUNG VON GLEICHUNGSSYSTEM-LÖSERN

Um neue LGS-Löser zu implementieren, ist es sinnvoll, sich den Interpretationsspielraum der Funktionsstruktur aus Algorithmus 4.1 zu verdeutlichen. Dabei wird in **Teil 1**, die Vorkonditionierung und **Teil 2**, das Lösen des LGS unterschieden.

Generell ist auf eine sinnvolle Wahl der Einflussmöglichkeiten durch den Benutzer zu achten. Dabei sollten nur Optionen zur Wahl gestellt werden, die die Ergebnisgüte für verschiedene

Abbildung 4.3.: Entscheidungskriterien zur Wahl von Lösungsverfahren für LGS, Nummerierung der Verfahren entsprechend Tabelle 4.4



Arten von LGS beeinflussen. Eine Option ist für den Benutzer nicht sinnvoll, wenn eine mögliche Auswahl (fast) nur Vorteile und die andere (fast) nur Nachteile bietet.

In **Teil 1** werden alle Lösungsschritte ausgeführt, die für ein LGS nach 3.1 mit mehreren Vektoren \mathbf{b} oder einer Matrix \mathbf{B} , lediglich ein mal ausgeführt werden. Dabei darf die Matrix \mathbf{B} nicht verwendet werden. In diesem Schritt können demnach Faktorisierung (nach 3.4) und Permutation (nach 3.3) vorgenommen werden. Alle Informationen und Daten, die für **Teil 2** notwendig sind und nicht in der Strukturvariable *handles* oder den Eingangs-Parametern enthalten sind, müssen im cell-Array *Preconditioner* gespeichert werden. Wird dies nicht konsequent verfolgt, so ist zwar die Ausführung mit drei Eingangsparametern möglich, jedoch nicht die Splittung von Vorkonditionierung und Lösen. Die Art der Besetzung der Zellen von *Preconditioner* ist nicht begrenzt. Da ein in **Teil 1** erstellter *Preconditioner* nur im dazugehörigen **Teil 2** verwendet wird, ist nur auf die Kompatibilität dieser Teile zueinander zu achten. \mathbf{X} ist mit dem Wert Null zu belegen.

In **Teil 2**, also dem Lösen des LGS, wird die Matrix \mathbf{B} verwendet. Außer den Eingangs-Parametern der Funktion inklusive der Variable *Preconditioner* darf dieser Teil keine Informationen benötigen. Für einen Vektor \mathbf{b} , muss ein Vektor \mathbf{x} und für eine Matrix \mathbf{B} eine dimensionsgleiche Matrix \mathbf{X} erzeugt werden.

Generell ist darauf zu achten, dass mögliche Programmabbrüche verhindert werden und entweder automatisch korrigiert oder als Wert eins an erster Stelle des cell-Arrays *Flag* ausgegeben werden. Die Position 2 von *Flag* ist für eine Fehlertext und die Position 3 für Funktionsname und Zeile im Funktionstext, in der der Fehler aufgetreten ist, vorgesehen.

Nach der Implementierung einer neuen Funktion ist die Konfigurations-Datei zu aktualisieren, dabei ist auf mögliche Einschränkungen der Optionen-Anzahl zu achten.

5. ANWENDUNG

Es sind drei KSM-Funktionen implementiert, jeweils eine Funktion aufbauend auf dem modifizierten Arnoldi-Algorithmus erster Ordnung, dem rationalen Block-SOAR- und rationalen, interpolativen SOAR-Algorithmus. Die Reduktionsfunktionen sind über Optionen steuerbar. Weiter sind zur Fehlerüberwachung Korrelationskriterien implementiert, wobei überwiegend frequenzgangbasierte Kriterien verwendet werden.

Nun werden die Reduktionsfunktionen an mechanischen Modellen getestet, deren Struktur zunächst beschrieben wird. Die einzelnen Optionen der Reduktionsfunktionen werden erst an zwei Modellen kleiner Dimension getestet, um Rückschlüsse auf eine sinnvolle Optionenwahl zu schließen und den Einfluss des Finite-Element-Typs zu quantifizieren. Anschließend werden die gewonnenen Erkenntnisse auf Modelle großer Dimension angewendet.

5.1. VERSUCHSMODELLE

Die verwendeten Modelle sind in Tabelle 5.1 zusammengefasst.

Tabelle 5.1.: Strukturmechanische Modelle zum Test der Reduktionsverfahren

Modellname	Elementtyp	Freiheitsgrad	Eingänge	Besonderheiten
Rahmen-Hex	Hexaeder	19029	36	kleine Dimension für MOR-Optionentest, drei Symmetrieebenen
Rahmen-Tet	Tetraeder	29358	36	kleine Dimension für MOR-Optionentest, drei Symmetrieebenen
Getriebe-Grob	Tetraeder	310338	54	
Getriebe-Fein	Tetraeder	749625	54	
Welle-Grob	Hexaeder	204492	42	Rotationssymmetrie
Welle-Fein	Tetraeder	510069	24	Rotationssymmetrie

5.1.1. TESTMODELLE KLEINER DIMENSION

Um unter vertretbarem Aufwand Tests jeglicher Optionen der Reduktionsmethoden durchführen zu können, werden zwei mechanische Systeme vergleichsweise kleiner Dimension verwendet. Es wird dafür die Geometrie eines Balkentragwerks in Form eines Rahmens verwendet. Die Struktur wird für ein System mit Hexaedern, für das andere mit Tetraedern vernetzt. Die vernetzten Strukturen sind in den Abbildungen 5.1 und 5.2 dargestellt.

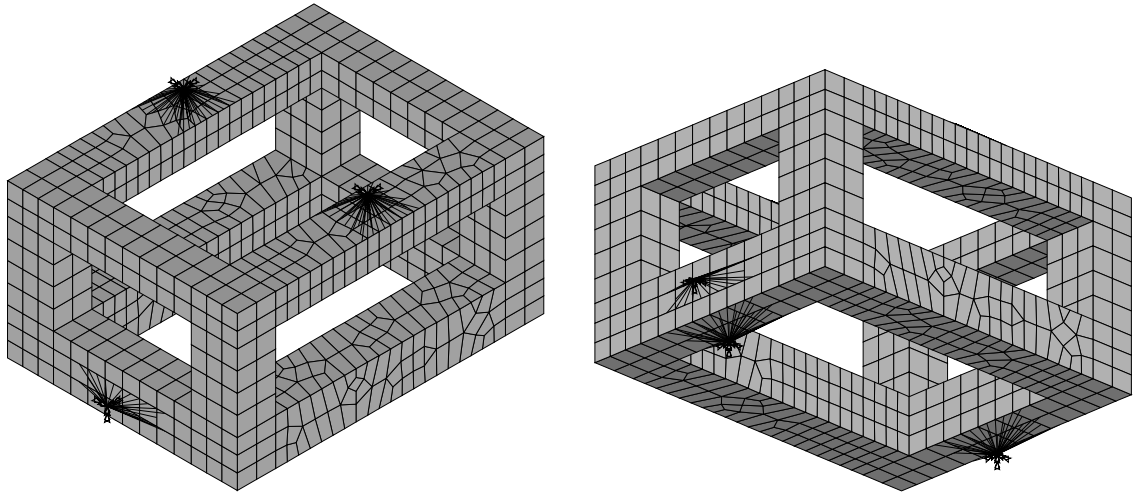


Abbildung 5.1.: Vernetztes Modell Rahmen-Hex (Hexaedervernetzung) mit Anbindungsknoten und deren Verbindung über MPC mit der Geometrie

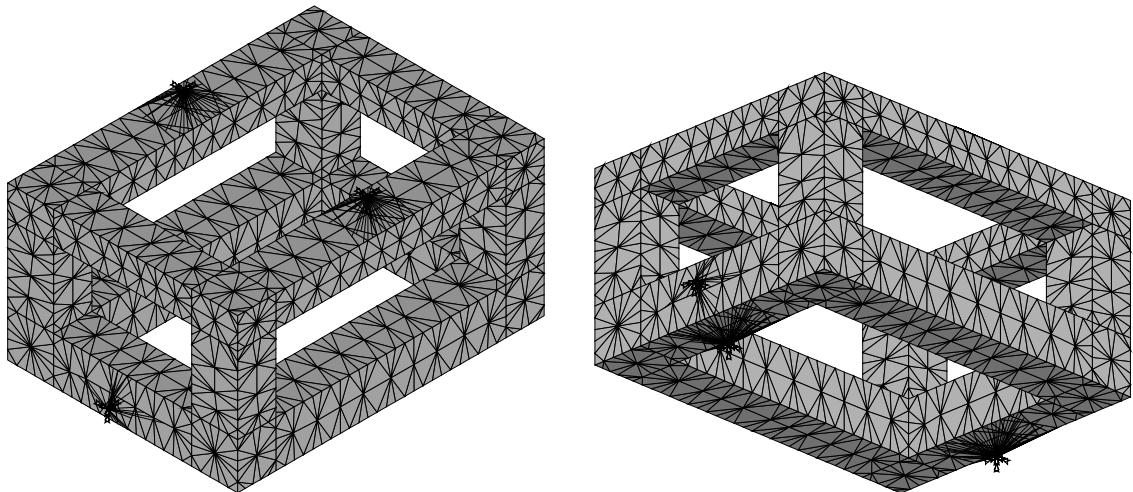


Abbildung 5.2.: Vernetztes Modell Rahmen-Tet (Tetraedervernetzung) mit Anbindungsknoten und deren Verbindung über MPC mit der Geometrie

Bestmögliche Berechnungsergebnisse sind für derartige Geometrien mithilfe der Vernetzung mit Balkenelementen zu erreichen. Da Aussagen über das Verhalten vorrangig tetraeder- vernetzter Modelle großer Dimension getroffen werden sollen, wird mit den genannten Volumenelemente diskretisiert.

Die Geometrie hat drei Symmetrieebenen und ist über Multi Point Constraints (MPC), in den Abbildungen durch spinnennetzförmige Verbindungen dargestellt, mit sechs Anbin- dungsknoten, deren Freiheitsgrade die Systemein- und -ausgänge darstellen, verbunden. In Anlehnung an einen Wagenkasten werden je zwei Anbindungsknoten für Fahrwerke, hori- zontale Kopplung und Dachaggregate positioniert. Durch die MPC sind die verbundenen Knoten starr gekoppelt. Die unterschiedliche Vernetzung hat so durch die unterschiedlich versteiften Flächen einen Einfluss auf das Systemverhalten.

Wesentliche Parameter der Geometrie und resultierenden mechanischen Systeme sind in

Tabelle 5.2 angegeben.

Tabelle 5.2.: Geometrie- und Netzdaten der Rahmen-Modelle

	Rahmen-Hex	Rahmen-Tet
Länge x Breite x Höhe	2.0m x 1.5m x 1.0m	
Elementtyp	Hexaeder	Tetraeder
Elementanzahl	1754	5615
Knotenanzahl	7083	10664
Freiheitsgrad	19029	29358
Ein- und Ausgänge	36	
f_7 (Erste Nichtstarrkörper-EF)	147Hz	148Hz
f_{50}	1435Hz	1464Hz
f_{100}	2858Hz	2959Hz

5.1.2. GETRIEBEGEHÄUSE

Es wird ein Getriebegehäuse, mit Tetraederelementen verschiedener Kantenlänge vernetzt, verwendet. Das grob vernetzte Modell ist in Abbildung 5.3 dargestellt.

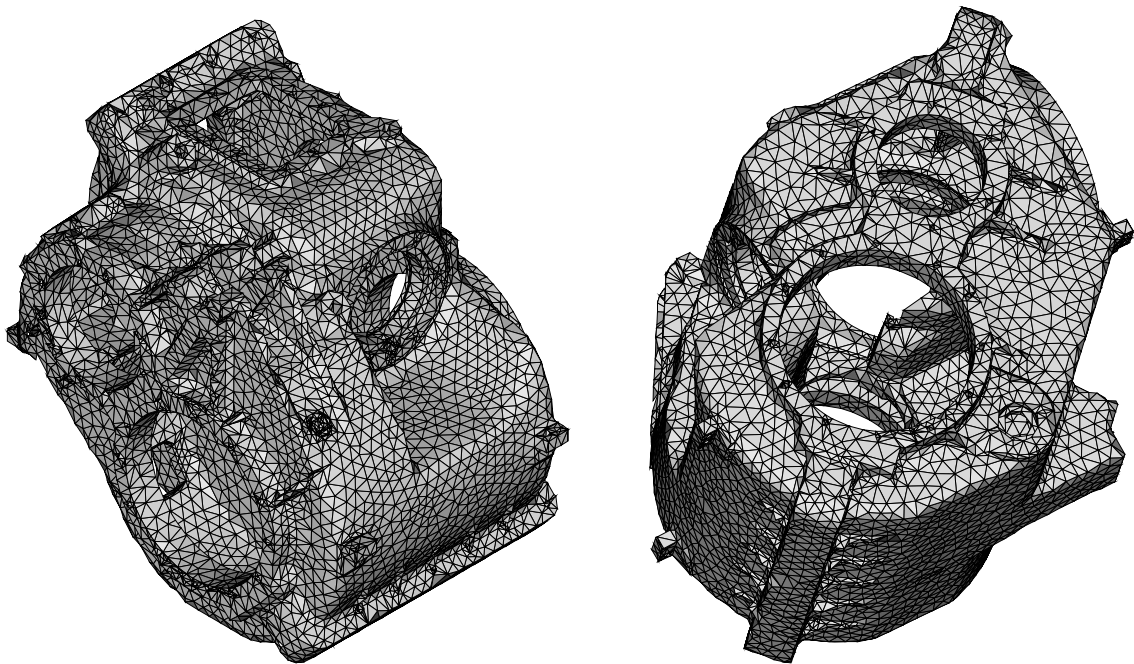


Abbildung 5.3.: Vernetztes Modell des Getriebegehäuses Getriebe-Grob

Die Anbindungsknoten mit der umgebenden Struktur befinden sich im Mittelpunkt der dargestellten Bohrungen und sind mit den jeweiligen Innenflächen der Bohrungen über elastische Verbindungselemente verbunden. Wesentliche Systemparameter sind in Tabelle 5.3 angegeben.

Tabelle 5.3.: Geometrie- und Netzdaten der Getriebegehäuse-Modelle

	Getriebe-Grob	Getriebe-Fein
Elementtyp	Tetraeder	
Elementanzahl	53058	141365
Knotenanzahl	100600	244208
Freiheitsgrad	310338	749625
Ein- und Ausgänge	54	
f_7 (Erste Nichtstarrkörper-EF)	586Hz	589Hz
f_{50}	1654Hz	1638Hz
f_{100}	2514Hz	2494Hz

5.1.3. TREIBSATZWELLE

Es wird das Modell einer Treibsatzwelle, mit Tetraedern und Hexaedern vernetzt, verwendet. Die beiden Vernetzungsarten und die Anbindungsknoten sind in Abbildung 5.4 dargestellt.

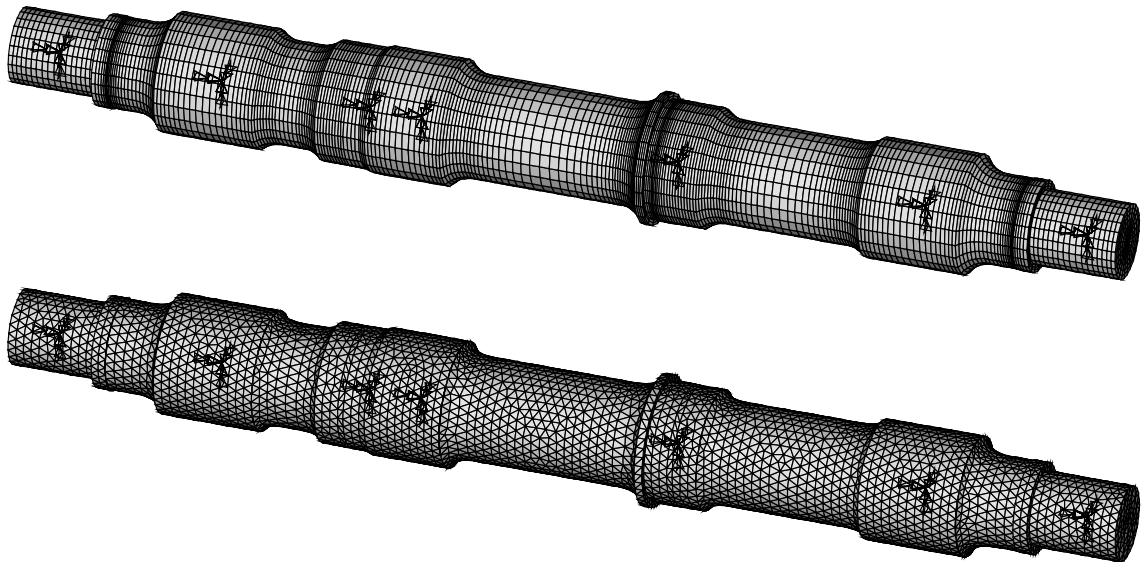


Abbildung 5.4.: Vernetztes Modell der Treibsatzwellen Welle-Grob und Welle-Fein

Die Anbindungsknoten sind im jeweiligen Querschnitt mit 4 weiteren Knoten verbunden. Wesentliche Systemparameter sind in Tabelle 5.4 angegeben.

Tabelle 5.4.: Geometrie- und Netzdaten der Treibsatzwellen-Modelle

	Welle-Grob	Welle-Fein
Elementtyp	Hexaeder	Tetraeder
Elementanzahl	67639	119809
Knotenanzahl	68185	171325
Freiheitsgrad	204492	510069
Ein- und Ausgänge	42	24
f_7 (Erste Nichtstarrkörper-EF)	218Hz	218Hz
f_{50}	7777Hz	7739Hz
f_{100}	11796Hz	11733Hz

5.2. VALIDIERUNG DER REDUKTIONSMETHODEN AN KLEINEM MODELL

Die implementierten Reduktions-Funktionen werden anhand der Modelle kleiner Dimension aus Abschnitt 5.1.1 validiert. Anhand der modifizierten Arnoldi-Funktion erster Ordnung wird

- die Eliminierung der Starrkörperbewegung,
- die einfache Reorthogonalisierung und
- der Einfluss des Finite-Element-Typs des vollen Modells

untersucht.

Anhand der rationalen SOAR-Funktionen wird

- die optimale Wahl des Startvektors,
- das interpolative im Vergleich zum Block-SOAR-Verfahren,
- die optimale Wahl einzelner Entwicklungsstellen $f_{ES} > 0$,
- die Reduktionsgüte in Abhängigkeit der reduzierten Systemdimension n und
- die Kombination verschiedener Entwicklungsstellen

untersucht.

5.2.1. MODIFIZIERTE ARNOLDI-FUNKTION ERSTER ORDNUNG

Es wird die Eliminierung der Starrkörperbewegung und die einfache Reorthogonalisierung validiert. Es wird der Einfluss des Elementtyps beim Vernetzen des Rahmen-Modells untersucht.

STARRKÖRPERBEWEGUNG

Wie unter 4.3.1 erläutert, werden die Krylov-Moden zu den Starrkörpermoden des vollen Systems orthogonalisiert. Daraus folgt die Eliminierung der Starrkörperbewegung im reduzierten System. Diagramm 5.1 zeigt das modMAC eines mit und ohne Orthogonalisierung zu den Starrkörpermoden reduzierten Systems.

Beim System unter Beibehaltung der Starrkörperbewegung gleichen sich die jeweils siebten Eigenmoden des vollen und reduzierten Systems. Das sind die ersten Nichtstarrkörpermoden. Beim System mit Eliminierung der Starrkörperbewegung entspricht die erste Eigenmode des reduzierten Systems der siebten Eigenmode des vollen Systems, die genannte Eliminierung funktioniert demnach.

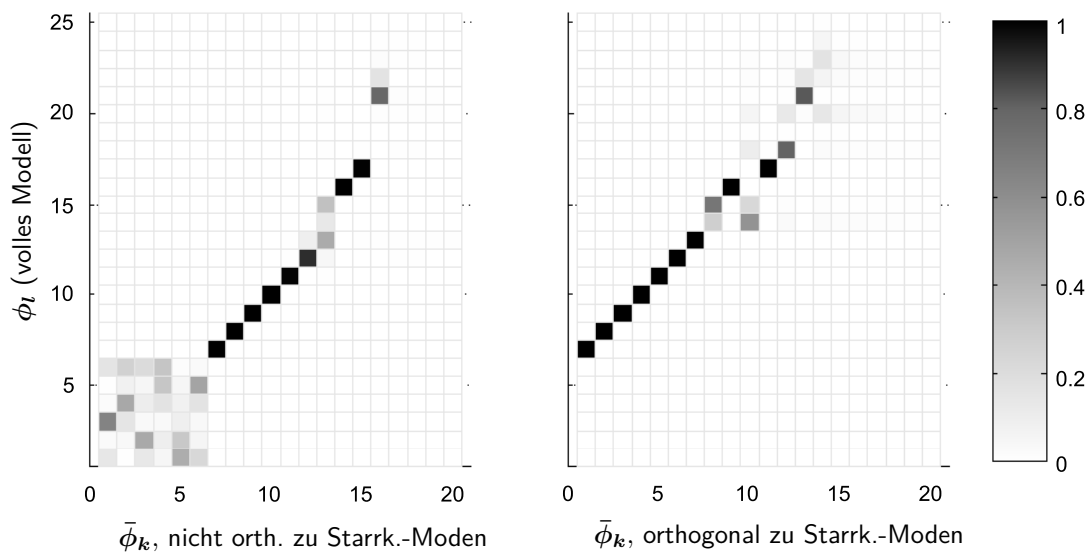


Diagramm 5.1: Korrelationskriterium modMAC eines reduzierten Systems mit und ohne Orthogonalisierung zu den Starrkörpermoden

REORTHOGONALISIERUNG

Wie unter 4.3.1 erläutert, ist die einfache Reorthogonalisierung aus Algorithmus 2.5 auch in die Funktion erster Ordnung integriert, um Rundungsfehler in der Orthogonalisierung zu reduzieren. Diagramm 5.2 zeigt die mittlere lineare Abhängigkeit der Krylov-Moden von den zuvor berechneten Krylov-Moden entsprechend Bild 3 in Abbildung 2.1.

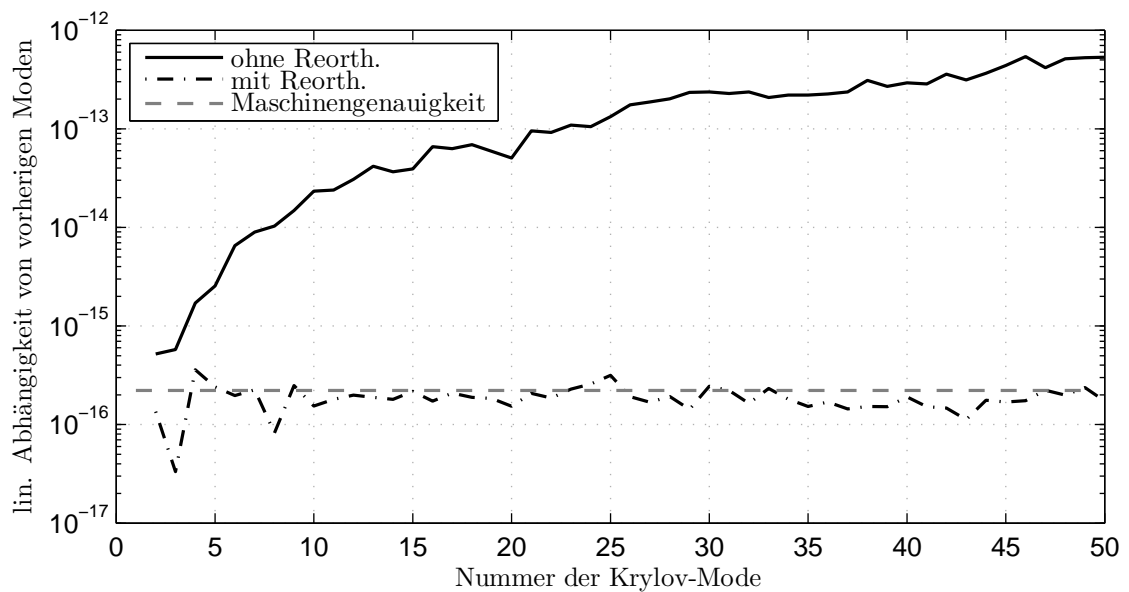


Diagramm 5.2: Mittlere lineare Abhängigkeit der Krylov-Moden von den vorherigen Krylov-Moden (Mittelwert von h in Algorithmus 2.5) bei Reduktion auf Dimension 50 (Rahmen-Tet)

Werte nahe Null bedeuten eine niedrige lineare Abhängigkeit und damit einen hohen

Grad an Orthogonalität. Es wird die Maschinengenauigkeit als Referenzgröße angegeben, die bei numerischen Berechnungen nicht wesentlich unterschritten werden kann. Ohne Reorthogonalisierung steigt die mittlere lineare Abhängigkeit mit zunehmender Krylov-Moden-Nummer. Mit Reorthogonalisierung sinkt die mittlere lineare Abhängigkeit unter die Maschinengenauigkeit. Dies gilt auch für größere, reduzierte Systemdimensionen, wie im Anhang unter A.3.1 gezeigt. Eine einfache Reorthogonalisierung ist demnach ausreichend.

VERGLEICH DES FINITE-ELEMENT-TYPS

Das Rahmen-Modell ist mit Hexaedern und Tetraedern vernetzt, um den Einfluss des Elementtyps zu quantifizieren. Wie erwähnt, wirkt sich die Anbindung an Anbindungsflächen verschiedener Größe auf das Systemverhalten aus. Dadurch sind die Eigenfrequenzen der tetraedervernetzten Struktur gegenüber der hexaedervernetzten leicht erhöht. Der Eigenfrequenzvergleich der auf Dimension 30 reduzierten Modelle **Rahmen-Tet** und **Rahmen-Hex** mit dem jeweiligen Vollmodell wird in Diagramm 5.3 dargestellt, der zugehörige Frobenius- und \mathcal{H}_2 -Fehler in Diagramm 5.4. Die Ergebnisse für weitere reduzierte Systemdimensionen sind in Tabelle 5.5 zusammengefasst, die entsprechenden Diagramme sind im Anhang unter A.3.2 angegeben.

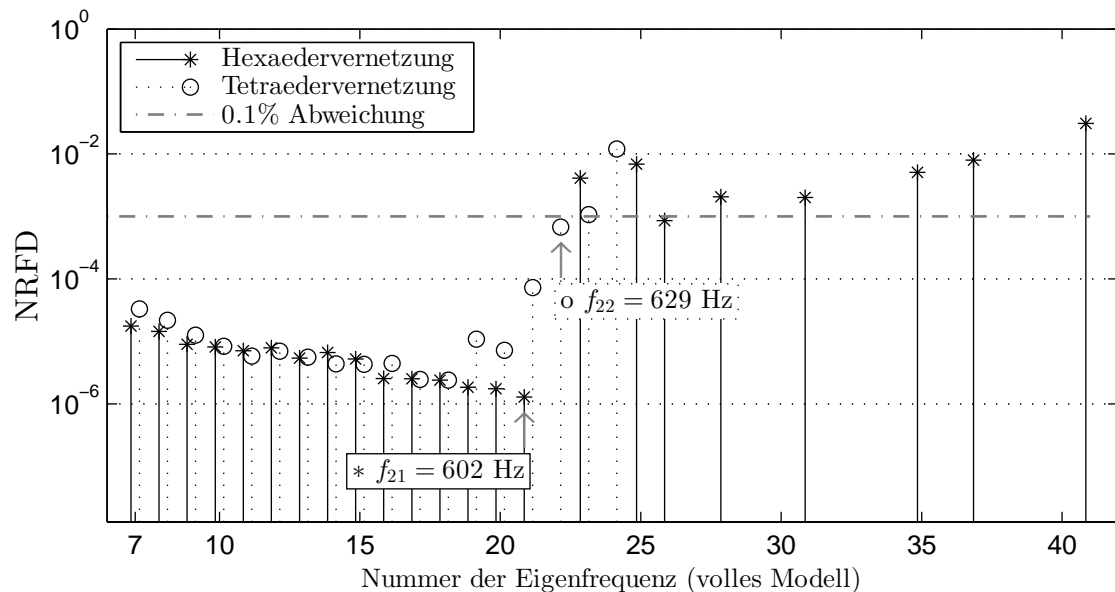


Diagramm 5.3: NRFD der Reduktion des hexaeder- und tetraedervernetzten Modells mit 30 Krylov-Moden

Diagramm 5.3 zeigt für **Rahmen-Hex** einen markanten Sprung zwischen den Abweichungen unterhalb und oberhalb der Fehlerschranke. Für **Rahmen-Tet** steigt die Abweichung für die größten vier Eigenfrequenzen unterhalb der Fehlerschranke qualitativ an. Tabelle 5.5 ist entnehmbar, dass die Grenzfrequenz¹ bei **Rahmen-Tet** generell oberhalb der von **Rahmen-Hex** liegt.

¹Mit Grenzfrequenz ist die größte Eigenfrequenz der zusammenhängenden Folge von Eigenfrequenzen mit $NRFD < 0.1\%$ gemeint.

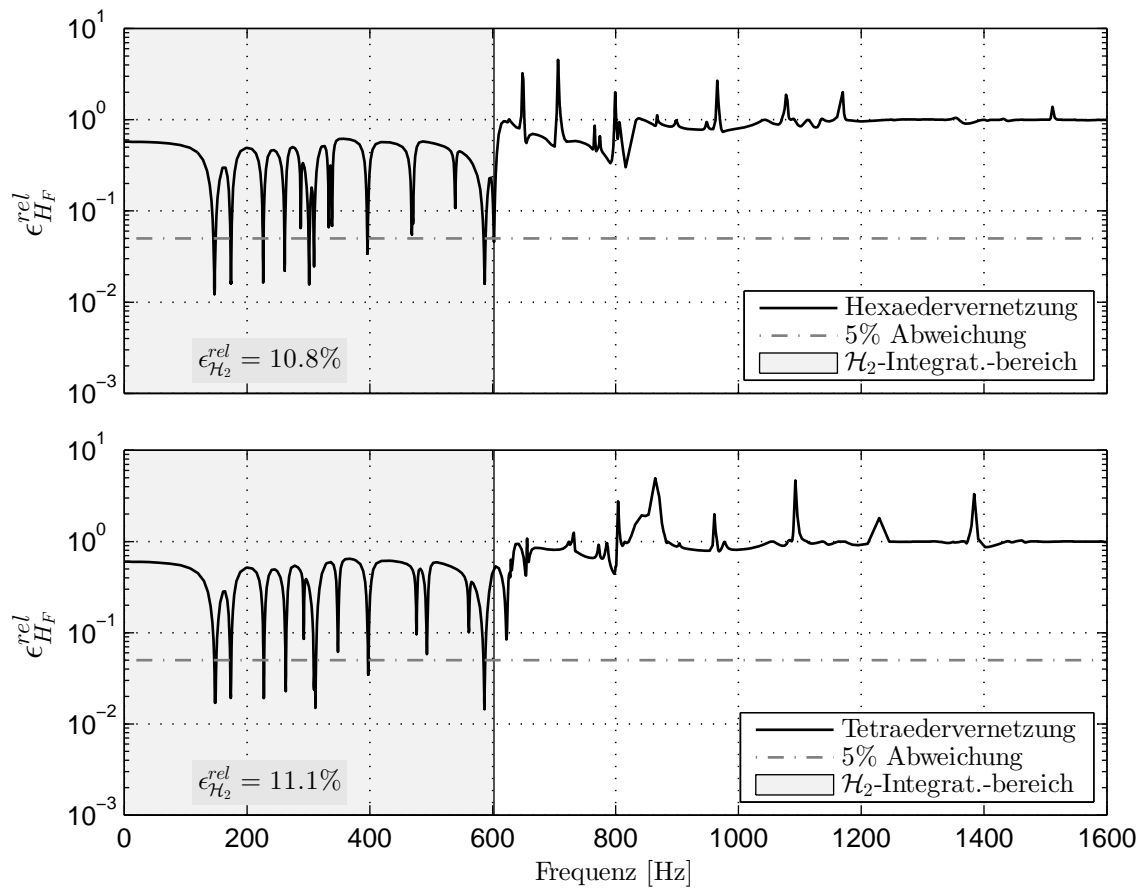


Diagramm 5.4: Abweichung des mithilfe von 30 Krylov-Moden reduzierten, hexaeder- und tetraedervernetzten Rahmenmodells (mod. Arnoldi-Verfahren erster Ordnung)

Tabelle 5.5.: Größte Eigenfrequenz unterhalb der 0.1%-Schranke, \mathcal{H}_2 -Integrationsbereich und relativer \mathcal{H}_2 -Fehler für Tetraeder- und Hexaedervernetzung für verschiedene reduzierte Systemdimensionen

Red. Dim.	Grenzfreq. Hexaeder	Grenzfreq. Tetraeder	\mathcal{H}_2 -Integr.-Bereich	\mathcal{H}_2 -Fehler Hexaeder $\epsilon_{\mathcal{H}_2}^{rel}$	\mathcal{H}_2 -Fehler Tetraeder $\epsilon_{\mathcal{H}_2}^{rel}$
20	$f_{11} = 287\text{Hz}$	$f_{13} = 311\text{Hz}$	$[0 \dots 287\text{Hz}]$	10.9%	9.99%
30	$f_{21} = 602\text{Hz}$	$f_{22} = 629\text{Hz}$	$[0 \dots 602\text{Hz}]$	10.8%	11.1%
40	$f_{26} = 767\text{Hz}$	$f_{26} = 773\text{Hz}$	$[0 \dots 767\text{Hz}]$	11.1%	14.3%
50	$f_{28} = 798\text{Hz}$	$f_{35} = 977\text{Hz}$	$[0 \dots 798\text{Hz}]$	4.13%	6.18%

In Diagramm 5.4 sind die entsprechenden Frobenius-Fehler ϵ_F^{rel} angegeben. Der \mathcal{H}_2 -Integrationsbereich wird durch den jeweils kleineren Übereinstimmungsbereich der Eigenfrequenzen festgelegt. Für diesen kleineren Integrationsbereich liefert **Rahmen-Hex** überwiegend kleinere \mathcal{H}_2 -Fehler (siehe Tabelle 5.5).

Damit ist das Modell mit Tetraedervernetzung in Bezug auf die Eigenfrequenzübereinstimmung und das Modell mit Hexaedervernetzung bezüglich des \mathcal{H}_2 -Fehlers im übereinstimmenden Frequenzbereich besser reduzierbar.

5.2.2. RATIONALE SOAR-FUNKTIONEN

Es wird die optimale Wahl eines interpolativen Startvektors untersucht und das rationale, interpolative SOAR-Verfahren mit Gleichwichtung mit dem Block-SOAR-Verfahren verglichen. Die Wahl einzelner Entwicklungsstellen $f_{ES} > 0$ und die Abhängigkeit der Reduktionsgüte von der reduzierten Systemdimension n wird untersucht. Es wird die Anzahl zu kombinierender Entwicklungsstellen zur Abbildung eines Frequenzbereiches ermittelt.

WAHL DES INTERPOLATIVEN STARTVEKTORS

Wie unter 4.3.2 angegeben, sind in der interpolativen, rationalen SOAR-Funktion mehrere Möglichkeiten vorgesehen, einen Startvektor der Reihenentwicklung zu interpolieren. Dabei wird die Gleichwichtung, bei der der Wichtungsvektor \mathbf{t}_k aus Gleichung 2.13 einem Einservektor entspricht, gewählt. Das berechnete Ergebnis wird mit zufälliger Interpolation und Startvektoren aus einzelnen Eingängen verglichen. In Tabelle 5.6 sind Parameterwahl und Ergebnisse für **Rahmen-Tet** bei reduzierter Dimension $n = 25$ zusammengefasst. Die Versuchswiederholung für größere reduzierte Dimension und das Modell **Rahmen-Hex** ist im Anhang unter A.3.3 aufgeführt.

Tabelle 5.6.: Parameterwahl und Ergebnisse des Testes mit zufälligen Startvektoren, einzelnen Eingängen und gleichgewichteter Interpolation am tetraedervernetzten Rahmenmodell **Rahmen-Tet**

Parameterwahl	
\mathcal{H}_2 -Integrationsbereich	[0 ... 580]Hz
Anzahl Frequenzstützstellen	1
Frequenzstützstelle	0Hz
Reduzierte Systemdimension	25
Lehrsche Dämpfung (für Korrelation)	$D = 0.001$
Versuche mit zufälliger Interpolation	1000
Versuche mit einzeltem Startvektor	36

Ergebnisse	
Größte Eigenfrequenz $NRFD < 0.1\%$ (Gleichwichtung)	$f_{20} = 587\text{Hz}$
\mathcal{H}_2 -Fehlerbereich mit zufälliger Interpolation	[9.01% ... 19.8%]
Mittelwert \mathcal{H}_2 -Abweichung mit zufälliger Interpolation	11.2%
Mittelwert \mathcal{H}_2 -Abweichung mit einzelnen Startvektoren	15.4%
Gleichgewichtete Interpolation	<u>10.8%</u>
Altes MORPACK-KSM (Spalte von K als Startvektor)	15.3%

Verglichen wird der \mathcal{H}_2 -Fehler für verschiedene Interpolationen. Um Vergleichbarkeit zu gewährleisten, ist ein fester \mathcal{H}_2 -Integrationsbereich zu wählen. Dieser orientiert sich an den übereinstimmenden Eigenfrequenzen für Gleichwichtung. Es werden die 36 Eingänge einzeln und 1000 zufällige Interpolationen betrachtet. Diagramm 5.5 zeigt die klassierte Verteilung der \mathcal{H}_2 -Fehler. Große Abweichungen werden nicht dargestellt und sind lediglich aus dem in Tabelle 5.6 angegebenen Mittelwert erkennbar.

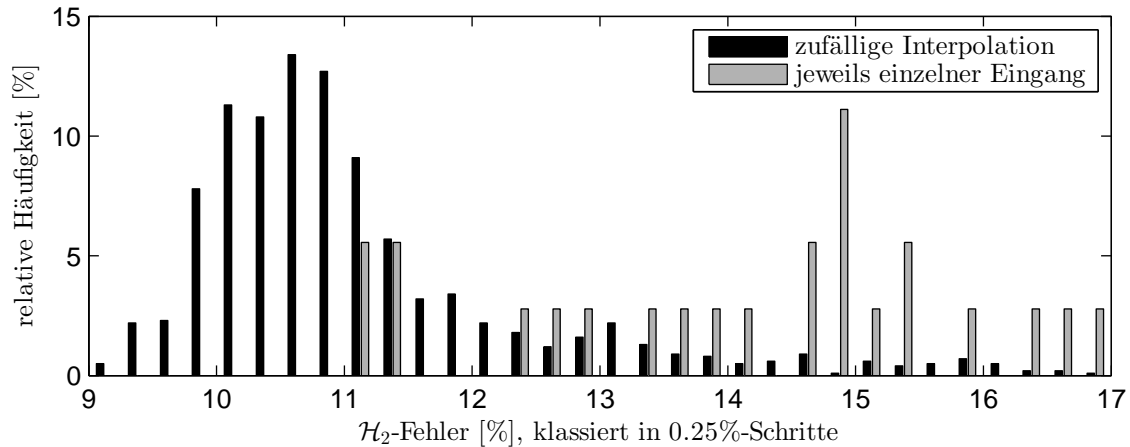


Diagramm 5.5: klassierte \mathcal{H}_2 -Abweichung bei 1000 zufällig interpolierten Startvektoren im Vergleich zu 36 Startvektoren aus einzelnen Eingängen (Rahmen-Tet)

Die \mathcal{H}_2 -Fehler bei zufälliger Interpolation sind näherungsweise normalverteilt. Der Mittelwert der Fehler ist wesentlich kleiner als der Mittelwert der Fehler mit Startvektoren einzelner Eingänge, welcher nicht normalverteilt ist. Der \mathcal{H}_2 -Fehler bei Gleichwichtung ist größer als das Maximum der Normalverteilung zufälliger Interpolation, kleiner als deren Mittelwert und wesentlich kleiner als der Mittelwert für einzelne Eingänge und das Ergebnis des vorliegenden MORPACK-KSM (ohne physikalisch interpretierbaren Startvektor).

Von den weiteren Versuchen im Anhang unter A.3.3 werden die gemachten Aussagen über Gleichwichtung bis auf die Unterschreitung des Mittelwertes der zufälligen Interpolation bestätigt.

Gleichwichtung als gewählte Interpolationsform liefert ein Reduktionsergebnis mittlerer Güte. Es besteht Entwicklungsbedarf bezüglich der Charakteristik optimaler Interpolation, wodurch die Möglichkeit einer deutlichen Senkung des \mathcal{H}_2 -Fehlers besteht.

VERGLEICH INTERPOLATIVER UND BLOCK-SOAR

Für den Vergleich des rationalen, interpolativen mit dem Block-SOAR-Verfahren werden verschiedene Kombinationen aus Eingängen für das Block-Verfahren mit interpolativer Gleichwichtung verglichen. Dabei wird nicht die Abweichung bei gleicher Ordnung der Reihenentwicklung für beide Verfahren bestimmt, sondern die Abweichung bei gleicher reduzierter Systemdimension und die reduzierte Dimension für ähnliche Abweichung. Die ausführlichen Ergebnisse sind im Anhang unter A.3.4 angegeben. Eine zusammenfassende Auswahl wird in Tabelle 5.7 angegeben. Wie unter 2.5 erläutert, ist die reduzierte Systemdimension beim Block-Verfahren immer ein Vielfaches der Eingangsanzahl.

Es werden die Eingangskombinationen

- alle Eingänge,
- Kombinationen aus den Eingängen mit bestem Einzelergebnis,

Tabelle 5.7.: Reduktionsergebnisse für verschiedene Eingangskombinationen bei Block-SOAR-Verfahren

reduz. Dimension	Eingänge	Letzte Eig.-Freq. mit $NRFD < 0.1\%$	\mathcal{H}_2 -Integr. Bereich	\mathcal{H}_2 -Fehler $\epsilon_{\mathcal{H}_2}^{rel}$	Red. Dim., interpolativ, ähnlicher $\epsilon_{\mathcal{H}_2}^{rel}$
Alle Eingänge					
72	1 bis 36	$f_{16} = 397\text{Hz}$	$[0 \dots 397\text{Hz}]$	9.33%	22
Alle Eingänge/Freiheitsgrade eines Anbindungsknotens (Masterknotens)					
24	13 bis 18	$f_{13} = 311\text{Hz}$	$[0 \dots 311\text{Hz}]$	11.0%	17
Translat. oder rotat. Eingänge/Freiheitsgrade eines Anbindungsknotens					
24 (*)	13 bis 15	$f_{14} = 348\text{Hz}$	$[0 \dots 397\text{Hz}]$	14.9%	17
24	16 bis 18	$f_{13} = 311\text{Hz}$	$[0 \dots 397\text{Hz}]$	13.2%	17
24 (*)	25 bis 27	$f_{18} = 493\text{Hz}$	$[0 \dots 397\text{Hz}]$	9.82%	22
24	28 bis 30	$f_{16} = 397\text{Hz}$	$[0 \dots 397\text{Hz}]$	10.5%	22
Jeweils zwei der drei Eingänge mit best./schlecht. gemeins. Ergebnis (*)					
24	25 26	$f_{19} = 561\text{Hz}$	$[0 \dots 476\text{Hz}]$	10.3%	22
24	25 27	$f_{17} = 476\text{Hz}$	$[0 \dots 476\text{Hz}]$	10.3%	22
24	26 27	$f_{16} = 397\text{Hz}$	$[0 \dots 476\text{Hz}]$	14.7%	20
24	13 14	$f_{18} = 493\text{Hz}$	$[0 \dots 476\text{Hz}]$	12.0%	22
24	13 15	$f_{18} = 493\text{Hz}$	$[0 \dots 476\text{Hz}]$	11.9%	22
24	14 15	$f_{16} = 397\text{Hz}$	$[0 \dots 476\text{Hz}]$	12.9%	21

- alle Eingänge eines Knotens und
- translatorische oder rotatorische Eingänge eines Knoten

untersucht. Es wird keine Kombination aus mehreren Eingängen gefunden, die die Reduktionsgüte des interpolativen Verfahrens mit Gleichwichtung erreicht. Von guten Einzelergebnissen ist nicht auf geeignete Eingangskombinationen schließbar, eher durch Herausgreifen einer Untermenge aus einer Kombination mit gutem Ergebnis. Letztgenanntes gilt lediglich tendenziell.

Grundsätzlich ist der interpolative Ansatz unter Verwendung von Gleichwichtung gegenüber dem Block-Ansatz zu bevorzugen, da mit gleicher reduzierter Systemdimension generell bessere Ergebnisse erzielt werden.

WAHL EINZELNER ENTWICKLUNGSSTELLEN

Um gute Kombinationen verschiedener Entwicklungsstellen finden zu können, werden zunächst einzelne Entwicklungsstellen $f_{ES} > 0$ betrachtet. Diagramm 5.6 zeigt bei einer reduzierten Dimension von $n = 25$ den jeweils resultierenden \mathcal{H}_2 -Fehler und den aus der Übereinstimmung der Eigenfrequenzen bestimmten \mathcal{H}_2 -Integrationsbereich. Eine Darstellung der Ergebnisse als Diagrammvideo der Frobeniusnorm der Übertragungsfunktionsmatrizen ist auf der beigelegten Daten-CD enthalten.

Der Abstand der getesteten Entwicklungsstellen ist mit der Nähe zu Eigenfrequenzen abnehmend gewählt. Diagramm 5.6 zeigt höhere \mathcal{H}_2 -Fehler und kleinere Integrationsbereiche bei

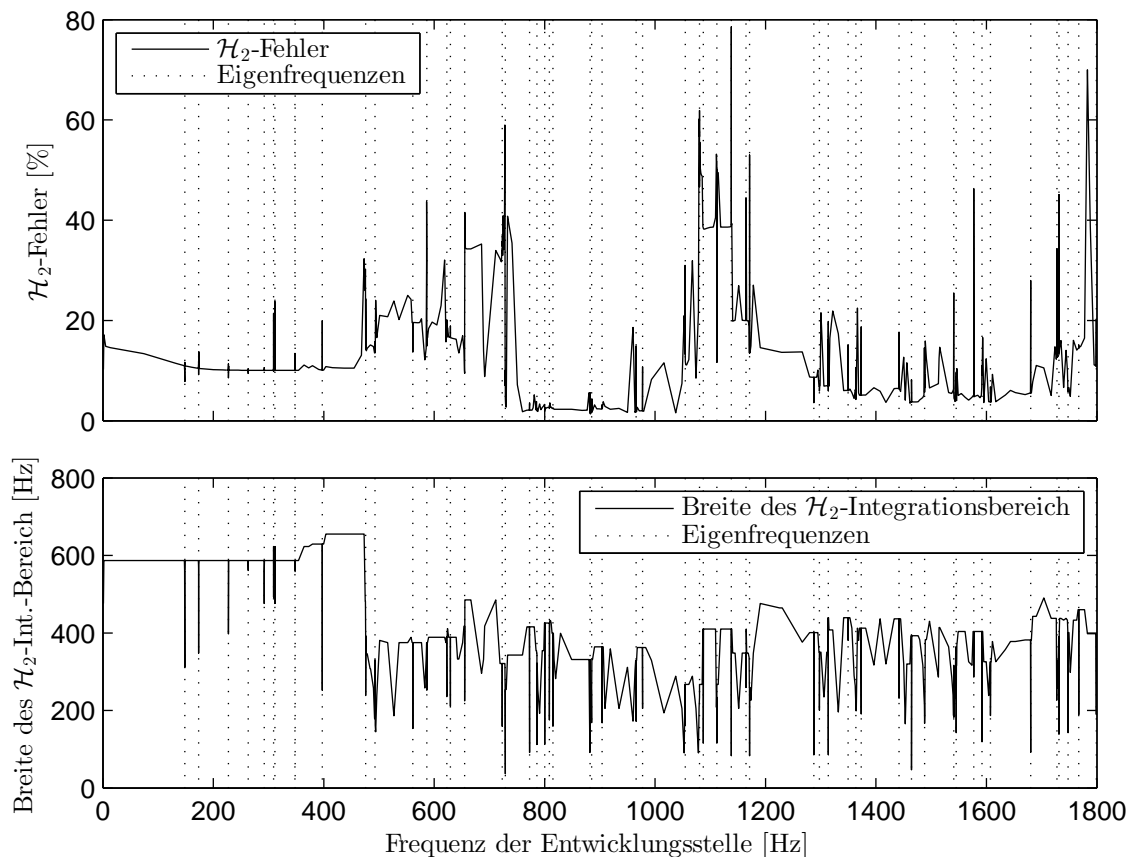


Diagramm 5.6: \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für verschiedene Entwicklungsstellen bei reduzierter Dimension von $n = 25$ des Rahmenmodells (Rahmen-Tet)

Entwicklungsstellen auf den Eigenfrequenzen². Die Vergrößerung des Integrationsbereiches für Entwicklungsstellen kleiner Frequenz gegenüber denen hoher Frequenzen entspricht der ersten Eigenfrequenz, die entsprechenden Entwicklungsstellen sind damit nicht besser zu bewerten. Wie in Abschnitt 4.4.3 beschrieben, befinden sich im Frequenzbereich $f = [800 \dots 1000]$ Hz Eigenmoden großer Amplitude, die einen großen Einfluss auf den angrenzenden Frequenzbereich haben. Werden diese Eigenmoden vom reduzierten System abgebildet, hat dies einen kleinen \mathcal{H}_2 -Fehler zu Folge. Im angrenzenden Bereich³ ist der \mathcal{H}_2 -Fehler groß, da hier diese nicht abgebildeten Eigenmoden einen großen Einfluss auf die Übertragungsfunktion haben. In größerem Abstand⁴ ist der Einfluss dieser dominanten Eigenmoden gering, was kleine \mathcal{H}_2 -Fehler zur Folge hat.

Im Anhang unter A.3.5 sind einige ausgewählte Diagramme mit den Übertragungsfunktionen für verschiedene Entwicklungspunkte dargestellt.

Generell ist die Wahl von Entwicklungsstellen auf Eigenfrequenzen ungeeignet ist. Von den Entwicklungsstellen als Nichteigenfrequenzen führt für einen Frequenzbereich zwischen

²Die höheren Fehler bei kleinerem Integrationsbereich beziehen sich auf die Ergebnisse unter Verwendung der jeweils benachbarten Entwicklungsstellen.

³Es sind die Frequenzbänder der Entwicklungsstellen $[550 \dots 800]$ Hz und $[1000 \dots 1200]$ Hz gemeint.

⁴Die Frequenzbänder der Entwicklungsstellen $[0 \dots 550]$ Hz und $[1000 \dots 1800]$ Hz sind gemeint.

zwei Eigenfrequenzen der überwiegende Teil möglicher Entwicklungsstellen zu ähnlichen Reduktionsergebnissen.

REDUKTIONSGÜTE IN ABHÄNGIGKEIT DER REDUZIERTEN SYSTEMDIMENSION FÜR EINZELNE ENTWICKLUNGSSTELLEN

Es wird untersucht, wie die Breite des \mathcal{H}_2 -Integrationsbereiches und der \mathcal{H}_2 -Fehler bei einzelnen Entwicklungsstellen von der reduzierten Systemdimension abhängt. Diagramm 5.7 zeigt die genannten Parameter bei Entwicklung um $f_{ES} = 0$ für das Modell Rahmen-Tet. Der Versuch dieser Art sind als Diagrammvideo der Frobeniusnorm der Übertragungsfunktionsmatrizen auf der beigelegten Daten-CD enthalten.

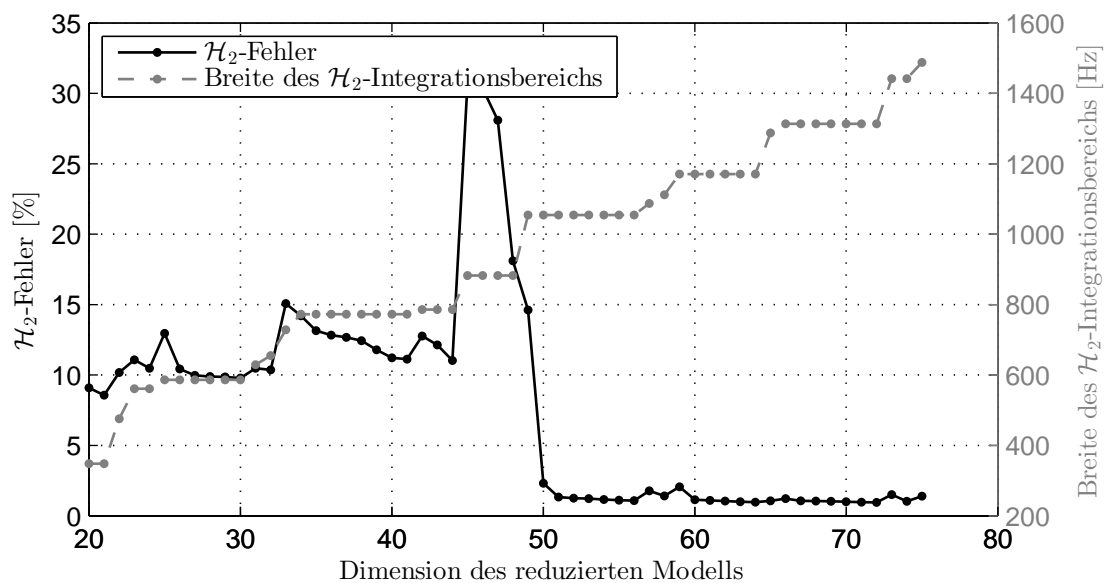


Diagramm 5.7: \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für variierende Dimension des reduzierten Systems n (Rahmen-Tet)

Die Breite des \mathcal{H}_2 -Integrationsbereiches steigt nicht kontinuierlich, sondern sprunghaft an. Es werden mit Reduktionsdimensionserhöhung die einzelnen Eigenfrequenzen nicht in geordneter Reihenfolge unter die Abweichungsgrenze $NRFD < 0.1\%$ verschoben, sondern zum Teil schon höhere Eigenfrequenzen abgebildet oder mehrere Eigenfrequenzen mit einer Dimensionserhöhung unter die Grenze bewegt. Das Maximum des \mathcal{H}_2 -Fehlers ist auf die Nähe der Integrationsbereichsgrenze zu den erwähnten dominanten Moden⁵ aufgrund der Massennormierung der Eigenmoden vor der Frequenzgangberechnung zurückzuführen. Das Ergebnis für Längennormierung des Eingang-Anteils der Eigenmoden ist im Anhang in Diagramm A.20 dargestellt. In diesem Fall sinkt der \mathcal{H}_2 -Fehler qualitativ mit zunehmender Systemdimension. Auf die Breite des \mathcal{H}_2 -Integrationsbereiches hat die Normierung keinen Einfluss.

⁵Die dominanten Moden werden im vorhergehenden Abschnitt und unter 4.4.3 erläutert.

Diagramm 5.8 zeigt die genannten Parameter für die Entwicklung um die Entwicklungsstelle $f_{ES} = 979\text{Hz}$, also einen Punkt nahe den dominanten Eigenmoden.

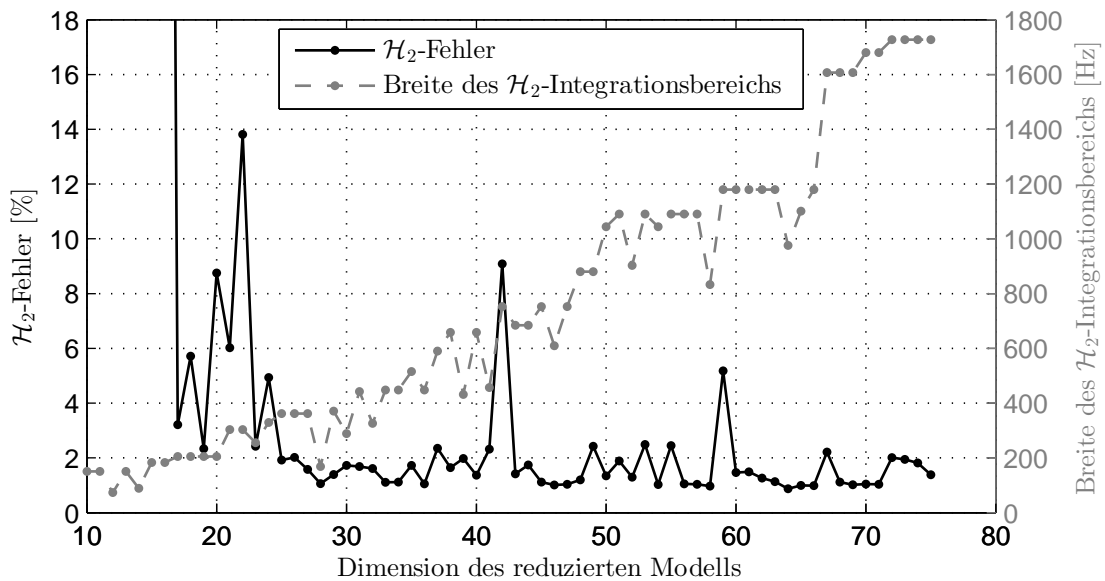


Diagramm 5.8: \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für aufsteigende, reduzierte Systemdimension bei Entwicklungsstelle $f_{ES} = 979\text{Hz}$ (Rahmen-Tet)

Der \mathcal{H}_2 -Integrationsbereich wächst im Gegensatz zur Entwicklung um $f_{ES} = 0$ nicht monoton, sondern lediglich qualitativ. Kleine \mathcal{H}_2 -Integrationsbereiche treten auf, wenn einzelne Eigenfrequenzen des reduzierten Systems im abgebildeten Frequenzband auftreten, die im vollen System nicht vorkommen. Bei reduzierter Dimension $n = 67$ wird die erste Nichtstarrkörper-Eigenfrequenz des vollen Modells ausreichend genau abgebildet und die Startfrequenz des \mathcal{H}_2 -Integrationsbereiches springt auf $f = 0$. Der \mathcal{H}_2 -Integrationsbereich ist dann größer als der mit Reihenentwicklung um $f_{ES} = 0$. Im Gegensatz zur Reduktion mit Entwicklungsstelle $f_{ES} = 0$ treten für einige Reduktionsdimensionen n auch nach der Abbildung der dominanten Moden⁶ einzelne große \mathcal{H}_2 -Fehler auf.

Im Anhang unter A.3.6 in Diagramm A.21 wird der Versuch für die Entwicklungsstelle $f_{ES} = 1230\text{Hz}$ wiederholt.

Es kann gesagt werden, dass die Reihenentwicklung um eine Entwicklungsstelle $f_{ES} > 0$ einen größeren \mathcal{H}_2 -Integrationsbereich, aber auch einen stärker variierenden \mathcal{H}_2 -Fehler zur Folge hat.

KOMBINATION VON ENTWICKLUNGSSTELLEN

Wie beschrieben, sind Entwicklungsstellen außerhalb von Eigenfrequenzen zu wählen. Weitere Aussagen über die optimale Wahl einer Entwicklungsstelle zwischen zwei Eigenfrequenzen können nicht getroffen werden. Um geringe \mathcal{H}_2 -Fehler zu erzielen, ist bei massengewichteten

⁶Die dominanten Moden werden ab einer reduzierten Dimension von $n = 25$ abgebildet.

Übertragungsfunktionen die Abbildung der Übertragungsfunktion im Bereich dominanter Moden von Bedeutung.

Es werden Kombinationen aus variierender Anzahl von Entwicklungsstellen, die in einem abzubildenden Frequenzbereich äquidistant verteilt sind, getestet. Die k Entwicklungsstellen f_{ES_i} werden äquidistant mit Δf im Frequenzbereich $[0 \dots 2000]$ Hz verteilt, wobei f_1 für die erste Eigenfrequenz steht

$$\underbrace{f_{start} \dots f_1}_{=0\text{Hz}} \underbrace{\dots}_{\frac{\Delta f}{2}} \underbrace{f_{ES_1}}_{\Delta f} \underbrace{\dots}_{\Delta f} \underbrace{f_{ES_2}}_{\Delta f} \underbrace{\dots}_{\Delta f} \dots \underbrace{\dots}_{\Delta f} \underbrace{f_{ES_i}}_{\Delta f} \underbrace{\dots}_{\frac{\Delta f}{3}} \underbrace{f_{end}}_{=2000\text{Hz}} \quad (5.1)$$

Die Ordnung der Reihenentwicklung ist für alle Entwicklungsstellen gleich. Damit ist die reduzierte Systemdimension n ein Vielfaches der Anzahl der Entwicklungsstellen k . Es wird die Dimension des reduzierten Systems zur Abbildung des vollständigen Frequenzbereiches nach NRFD-Kriterium und der \mathcal{H}_2 -Fehler als Bewertungskriterium gewählt. Das Ergebnis für Anzahl der Entwicklungsstellen $k = 1 \dots 20$ ist in Diagramm 5.9 dargestellt.

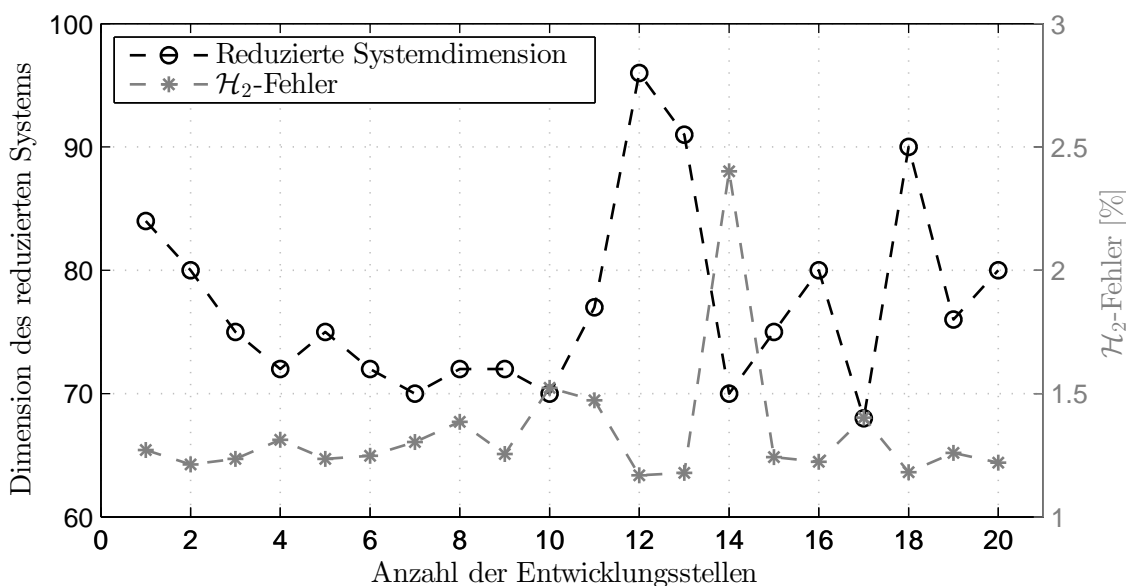


Diagramm 5.9: Notwendige reduzierte Systemdimension zur Abbildung des Frequenzbereiches $f = [0 \dots 2000]$ Hz für verschiedene Anzahl äquidistant verteilter Entwicklungsstellen und zugehörige \mathcal{H}_2 -Fehler

Für jede der k Entwicklungsstellen ist eine LU-Zerlegung nach Abschnitt 3.4.2 zu berechnen. Da schon eine einmalige Zerlegung für Systeme großer Dimension⁷ den größten Zeitanteil der Berechnung einnimmt, ist eine kleine Anzahl von Entwicklungsstellen anzustreben.

⁷Die größten Berechnungszeitanteile bei KSM mit einer Entwicklungsstelle sind die Zeit für eine Faktorisierung und die mit zunehmender Krylov-Modenzahl steigende Orthogonalisierungszeit. Die Systemdimension, ab der die Faktorisierungszeit die Orthogonalisierungszeit übersteigt, hängt demnach von der reduzierten Systemdimension ab und bewegt sich überwiegend im Bereich $N = [10^5 \dots 10^6]$.

Wie aus Diagramm 5.9 ersichtlich, ist der \mathcal{H}_2 -Fehler für alle Versuche vergleichsweise klein und somit für die Wahl der Entwicklungsstellenanzahl nicht ausschlaggebend. Mit reduzierten Systemdimensionen im Bereich von $n = [70 \dots 75]$ liefern Entwicklungsstellenanzahlen von $k = [3 \dots 10]$ die besten Ergebnisse, wobei $n = 70$ bei $k = 7$ und $k = 10$ erreicht wird. Bei größeren k variiert n stark⁸.

Der Versuch am Modell **Rahmen-Hex** ist im Anhang unter A.3.7 angegeben und liefert kleine reduzierte Systemdimensionen im Bereich $n = [72 \dots 78]$ für Entwicklungsstellenanzahlen von $k = [3 \dots 9]$, mit Minimum bei $k = 6, 8, 9$, jedoch maximalem \mathcal{H}_2 -Fehler⁹ bei $k = 6$. Die Ordnung der Reihenentwicklung j ergibt sich zu

$$j = \frac{n}{k} \quad . \quad (5.2)$$

5.2.3. ZUSAMMENFASSUNG

Auf Grundlage der Versuche an den Modellen kleiner Dimension

- ist die Eliminierung der Starrkörperbewegung und
- die Funktionalität der einfachen Reorthogonalisierung validiert,
- sind gute Ergebnisse mit Gleichwichtung der Eingänge bei interpolativem SOAR erzielt,
- ist der rationale, interpolative SOAR gegenüber dem Block-SOAR als effektiver eingestuft und
- sind deutlich bessere Ergebnisse mit mehreren Entwicklungsstellen gegenüber einer Entwicklungsstelle erzielt

worden. Die Anzahl der äquidistant verteilten Entwicklungsstellen f_{ES_k} ist für das rationale, interpolative Verfahren im Bereich $k = [3 \dots 9]$ zu wählen. Die Vernetzungsart hat Einfluss auf die Reduktion, das hexaedervernetzte Modell neigt zu schlechteren Reduktionsergebnissen. Die Reduktionsgüte nimmt mit steigender reduzierter Systemdimension zu, bei Entwicklung um die Entwicklungsstelle $f_{ES} = 0$ kontinuierlich, für $f_{ES} > 0$ lediglich im Mittel.

5.3. ANWENDUNG DER KSM AUF GROSSE MODELLE

Es ist das rationale, interpolative SOAR-Verfahren mit Gleichwichtung als Favorit aus der Untersuchung des Rahmenmodells hervorgegangen. Dabei ist eine Anzahl äquidistant verteilter Frequenz-Entwicklungsstellen im Bereich $k = [3 \dots 9]$ zu wählen. Alternativ liefert eine Entwicklungsstelle $f_{ES} = 0$ stabilere, jedoch schlechtere Ergebnisse.

⁸Der Grund für das starke Variieren ist folgender: Die reduzierte Systemdimension n ist ein Vielfaches von k . Die Dimension n wächst bei nur knappem Verfehlen des NRFD-Kriteriums um k an.

⁹Der \mathcal{H}_2 -Fehler ist an dieser Stelle mit $\epsilon_{\mathcal{H}_2}^{rel} = 1.58\%$ ca doppelt so groß wie für die anderen k .

Beide Funktionen werden anhand von Modellen großer Dimension verglichen. Es werden die Modelle `Getriebe-Grob`, `Getriebe-Fein`, `Welle-Grob` und `Welle-Fein` verwendet.

Als Bewertungskriterium gilt die notwendige reduzierte Systemdimension zur Abbildung eines Frequenzbereiches und der \mathcal{H}_2 -Fehler bei gleicher reduzierter Dimension in einem festen Frequenzbereich.

5.3.1. GETRIEBEGEHÄUSE

Das Modell Getriebegehäuse, mit verschiedener Elementgröße vernetzt, wird in Abschnitt 5.1.2 vorgestellt. Es wird das kleinere System `Getriebe-Grob` und das größere System `Getriebe-Fein` untersucht.

Es werden die Reduktionsergebnisse für

- eine Entwicklungsstelle $f_{ES} = 0$ mit denen für
- mehrere äquidistant verteilte Entwicklungsstellen

verglichen. Es wird jeweils die rationale, interpolative SOAR-Funktion mit Gleichwichtung verwendet. Der Versuch mit $f_{ES} = 0$ ist auch mithilfe der modifizierten Arnoldi-Funktion erster Ordnung durchführbar.

Die Übertragungsfunktionsmatrix wird für die Getriebegehäuse-Modelle über die $\|\dots\|_2$ -Norm der Eingangsanteile der Eigenmoden nach Gleichung 2.23 berechnet, da viele dominante Moden außerhalb des betrachteten Frequenzbereiches das Übertragungsverhalten sehr stark beeinflussen.

REDUKTIONSGÜTE IN ABHÄNGIGKEIT DER REDUZierten SYSTEMDIMENSION FÜR EINZELNE ENTWICKLUNGSSTELLEN

Wie in Abschnitt 5.2.2 durchgeführt, wird die Abhängigkeit der Reduktionsgüte von der reduzierten Systemdimension überprüft. Die Ergebnisse des Versuchs für die beiden Getriebegehäuse-Modelle sind in Diagramm 5.10 angegeben, des entsprechenden Versuchs für das Rahmenmodell in Diagramm A.20. Es wird dabei die Entwicklungsstelle $f_{ES} = 0$ verwendet. Der Versuch für das Modell `Getriebe-Fein` mit maximaler reduzierter Systemdimension $n = 150$ ist im Anhang in Diagramm A.23 dargestellt. Die Versuche sind durch Diagrammvideos der Frobeniusnorm der Übertragungsfunktionsmatrizen auf der Daten-CD anschaulich visualisiert.

Das Diagramm 5.10 zeigt sowohl für den auf Eigenfrequenzübereinstimmung basierenden \mathcal{H}_2 -Integrationsbereich, als auch für den \mathcal{H}_2 -Fehler qualitative Übereinstimmung beider Vernetzungsarten. Die Breite des \mathcal{H}_2 -Integrationsbereiches steigt mit zunehmender reduzierter Systemdimension monoton in Sprüngen an. Dies entspricht dem Verhalten des Rahmenmodells. Der \mathcal{H}_2 -Fehler nimmt qualitativ ab, ist aber mit Werten im Bereich $\epsilon_{\mathcal{H}_2}^{rel} = [9 \dots 20]\%$ relativ groß.

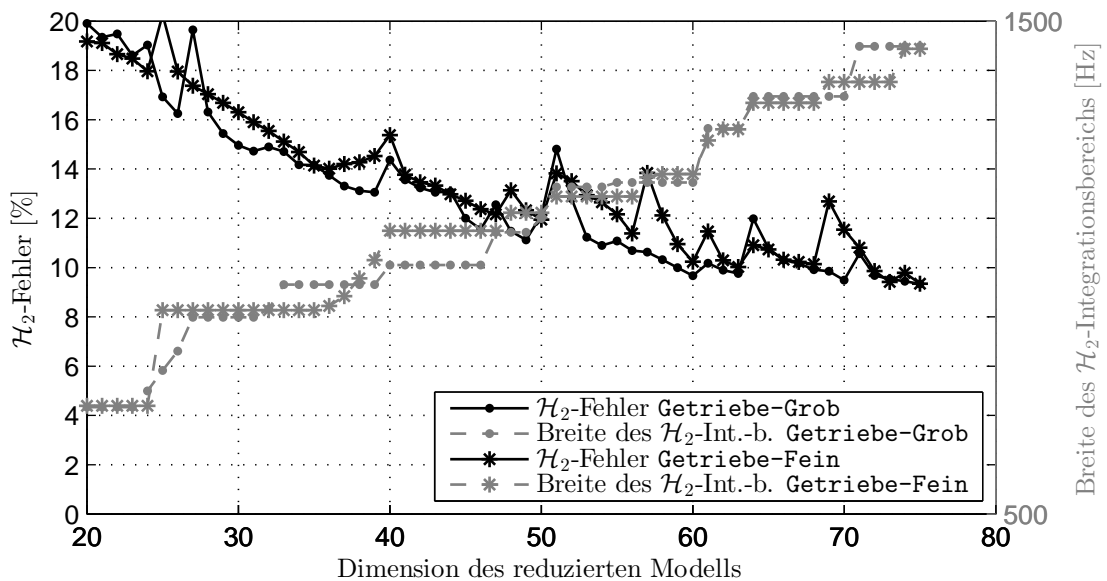


Diagramm 5.10: \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für variierende Dimension des reduzierten Systems n für die Modelle *Getriebe-Grob* und *Getriebe-Fein* mit einer Entwicklungsstelle $f_{ES} = 0$

Für das Getriebegehäuse-Modell hat folglich die Netzfeinheit einen geringen Einfluss auf die Reduktionsgüte. Diese Aussage gilt im betrachteten Bereich $N = [3 \cdot 10^5 \dots 8 \cdot 10^5]$.

Die Abhängigkeit der Reduktionsgüte von der reduzierten Systemdimension für die Entwicklungsstelle $f_{ES} = 1300\text{Hz}$ ist in Diagramm 5.11 angegeben. Die entsprechenden Versuche für das Modell *Rahmen-Tet* sind in den Diagrammen 5.8 und A.21 dargestellt und werden in Abschnitt 5.2.1 behandelt. Die genannten Diagramme für *Getriebe-Grob* und *Rahmen-Tet* ähneln sich qualitativ.

Diagramm 5.11 zeigt, dass für das Modell *Getriebe-Grob* die Breite des \mathcal{H}_2 -Integrationsbereiches nicht monoton, sondern nur qualitativ steigt. Ab einer reduzierten Systemdimension $n = 68$ wird die erste Eigenfrequenz hinreichend genau abgebildet und der \mathcal{H}_2 -Integrationsbereich beginnt mit $f = 0$. Der \mathcal{H}_2 -Integrationsbereich ist dann für gleiche reduzierte Systemdimension breiter als mit Entwicklungsstelle $f_{ES} = 0$. Der \mathcal{H}_2 -Fehler variiert stark, sinkt aber im Mittel. Mit Abbilden der ersten Eigenfrequenz sinkt der \mathcal{H}_2 -Fehler streng monoton.

Die Reduktion mit Entwicklungsstelle $f_{ES} = 0$ liefert wie bei Modell *Rahmen-Tet* mit wachsender reduzierter Systemdimension kontinuierlich bessere Reduktionsergebnisse. Dies trifft nur bedingt auf die Reduktion mit einer Entwicklungsstelle $f_{ES} > 0$ zu, mithilfe welcher dagegen ein größerer Frequenzbereich abgebildet werden kann.

Reduktion mit der Entwicklungsstelle $f_{ES} = 0$ führt zu stabileren, die Reduktion mit $f_{ES} > 0$ zu potentiell besseren Reduktionsergebnissen.

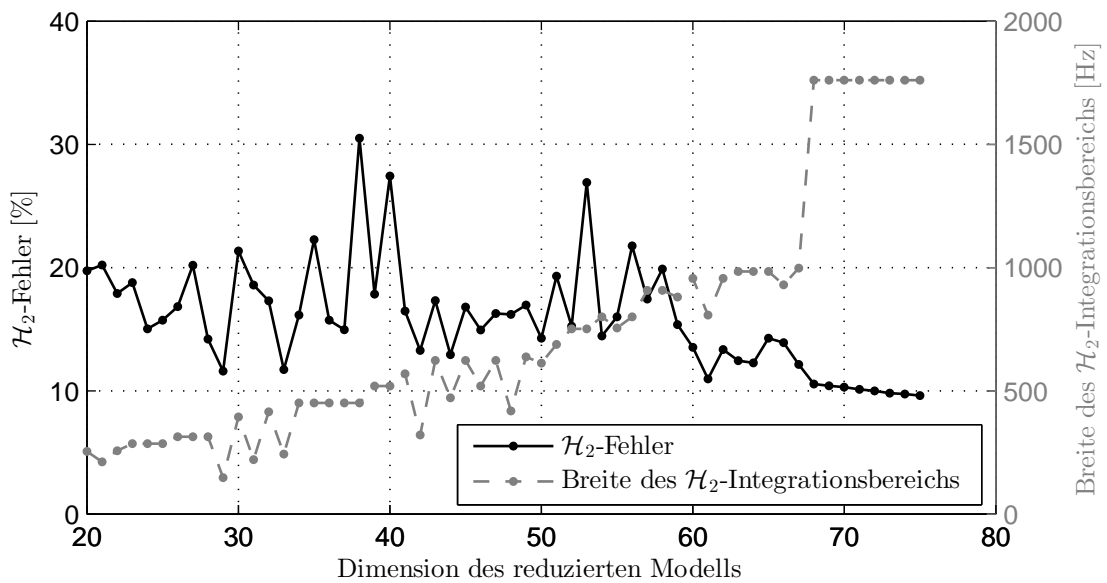


Diagramm 5.11: \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für variierende Dimension des reduzierten Systems n für das Modell **Getriebe-Grob** und Entwicklungsstelle $f_{ES} = 1300\text{Hz}$

KOMBINATION VON ENTWICKLUNGSTELLEN

Es wird die Kombination von Entwicklungsstellen zur Abbildung des Frequenzbereiches $f = [0 \dots 2000]\text{Hz}$ anhand des Modells **Getriebe-Grob** untersucht und auf das Modell **Getriebe-Fein** angewendet. Die entsprechenden Versuche für die Modelle **Rahmen-Tet** und **Rahmen-Hex** mit bis zu 20 Entwicklungsstellen sind in den Diagrammen 5.9 und A.22 dargestellt und werden in Abschnitt 5.2.2 behandelt. Es wird eine optimale Entwicklungsstellenanzahl $k = [3 \dots 9]$ vorgeschlagen. Für das Modell **Getriebe-Grob** werden bis zu $k = 13$ äquidistant verteilte Frequenz-Entwicklungsstellen verwendet.

Diagramm 5.12 zeigt die notwendige reduzierte Systemdimension zur Abbildung des Frequenzbereiches $f = [0 \dots 2000]\text{Hz}$ anhand des Modells **Getriebe-Grob** und den entsprechenden \mathcal{H}_2 -Fehler.

Es werden, wie auch bei den Versuchen in Abschnitt 5.2.2 kleine reduzierte Systemdimensionen $n = [70 \dots 78]$ mit Entwicklungsstellenanzahlen $k = [3 \dots 9]$ erreicht. Das Minimum mit $n = 70$ und das Maximum des \mathcal{H}_2 -Fehlers wird mit $k = 7$ Entwicklungsstellen erreicht. Der \mathcal{H}_2 -Fehler liegt im Bereich $\epsilon_{\mathcal{H}_2}^{rel} = [10 \dots 11.5]\%$.

Es wird das Reduktionsergebnis für vier Entwicklungsstellen untersucht. Der Frobenius- und \mathcal{H}_2 -Fehler für das Modell **Getriebe-Grob** mit reduzierter Systemdimension $n = 76$ ist in Diagramm 5.13 dargestellt.

Aufgrund des zunehmenden Einflusses der Moden außerhalb des abgebildeten Frequenzbereiches steigt der Frobenius-Fehler mit steigender Frequenz.

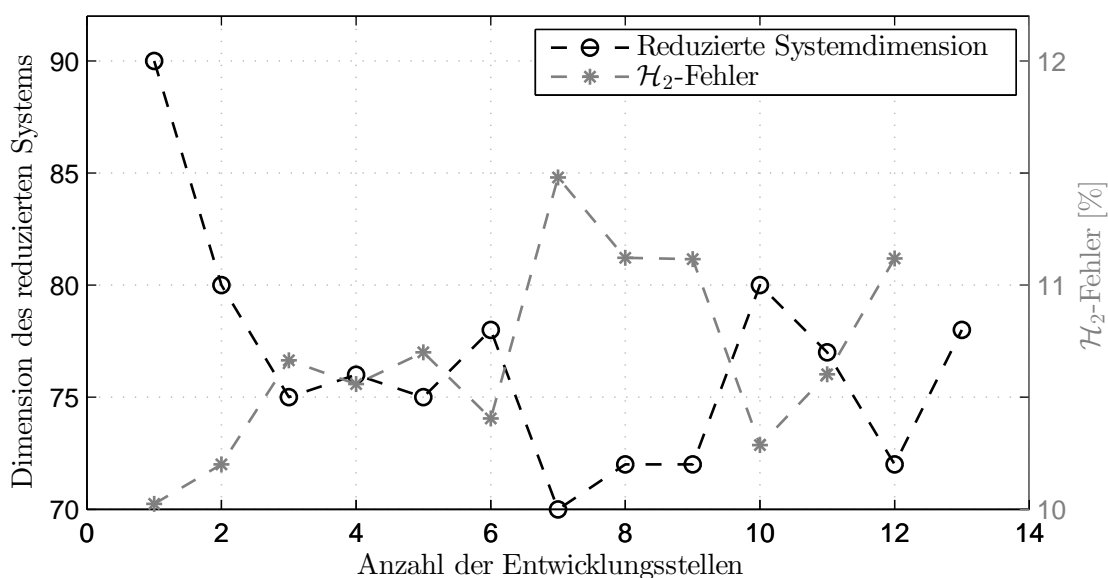


Diagramm 5.12: Notwendige reduzierte Systemdimension zur Abbildung des Frequenzbereiches $f = [0 \dots 2000]$ Hz für verschiedene Anzahl linear verteilter Entwicklungsstellen und zugehörige \mathcal{H}_2 -Fehler (Getriebe-Grob)

Es wird untersucht, wie sich eine kleinere reduzierte Systemdimension auf das Reduktionsergebnis auswirkt. Diagramm 5.14 zeigt das eigenfrequenzbasierte NRFD-Kriterium mit $k = 4$ Entwicklungsstellen und reduzierter Systemdimension $n = 72$. Der zugehörige Frobenius-Fehler ist im Anhang in Diagramm A.24 dargestellt.

Die 56. Eigenfrequenz des vollen Systems wird nicht abgebildet. Damit ist nicht der Rand des abzubildenden Frequenzbereiches fehlerhaft. Der Abstand der äußeren Entwicklungsstellen f_{ES_1} und f_{ES_k} zum Rand des abzubildenden Frequenzbereiches nach Gleichung 5.1 ist demnach klein genug.

Es wird das Modell **Getriebe-Fein** mit $k = 4$ Entwicklungsstellen und reduzierter Dimension $n = 76$ reduziert. Im Anhang zeigt Diagramm A.25 die entsprechenden NRFD-Abweichungen. Es wird der Frequenzbereich $[0 \dots 2030]$ Hz mit $NRFD < 0.1\%$ abgebildet. Diagramm 5.15 zeigt den resultierenden Frobenius- und \mathcal{H}_2 -Fehler.

Der \mathcal{H}_2 -Fehler ist mit $\epsilon_{\mathcal{H}_2}^{rel} = 11.8\%$ größer als der Fehler bei Reduktion mit gleichen Parametern für das Modell **Getriebe-Grob** mit $\epsilon_{\mathcal{H}_2}^{rel} = 10.5\%$. Um den Frequenzbereich $[0 \dots 2000]$ Hz mit einer Entwicklungsstelle $f_{ES} = 0$ abzubilden, ist eine reduzierte Systemdimension $n = 118$ notwendig, wie aus Diagramm A.23 im Anhang hervorgeht. Jedoch wird mit $\epsilon_{\mathcal{H}_2}^{rel} = 9.25\%$ ein kleinerer \mathcal{H}_2 -Fehler erreicht.

Die Erkenntnisse aus den Versuchen an den Rahmen-Modellen sind auf die beiden Getriebegehäuse -Modelle anwendbar. Im Vergleich zum Verfahren mit Entwicklungsstelle $f_{ES} = 0$ kann mit vier äquidistant verteilten Entwicklungsstellen die reduzierte Systemdimension um 35% reduziert werden. Als Fehlerkriterium gilt dabei die Abbildung der Eigenfrequenzen des vorgegebenen Frequenzbereiches mit $NRFD < 0.1\%$.

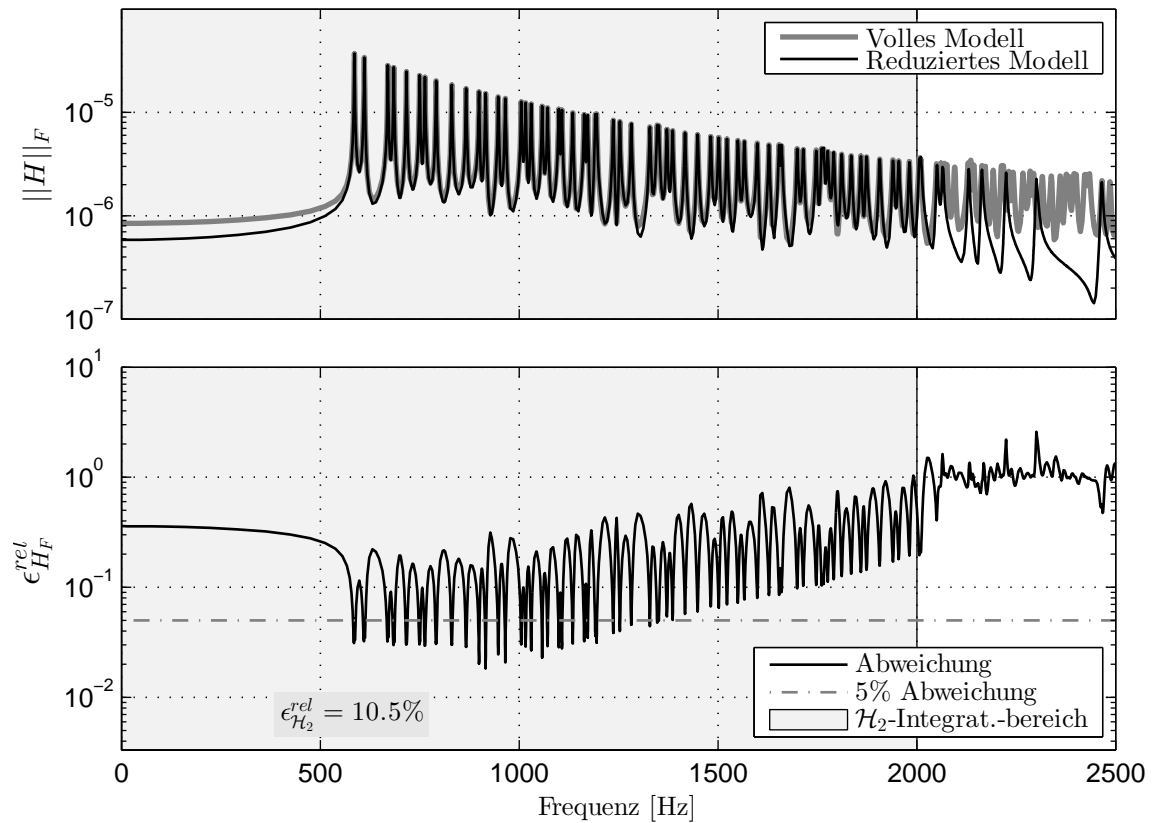


Diagramm 5.13: Frobenius-Norm der Übertragungsfunktionsmatrizen und relative Abweichung sowie \mathcal{H}_2 -Fehler für reduzierte Dimension $n = 76$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 2000]$ Hz (Getriebe-Grob)

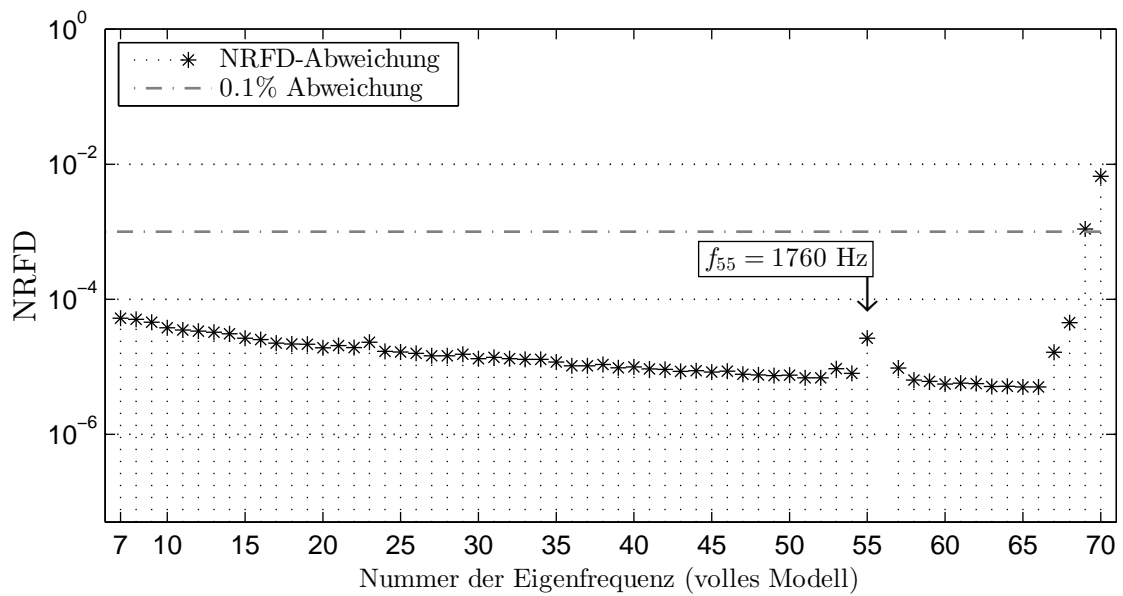


Diagramm 5.14: NREFD der Eigenfrequenzen für reduzierte Dimension $n = 72$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 2000]$ Hz (Getriebe-Grob)

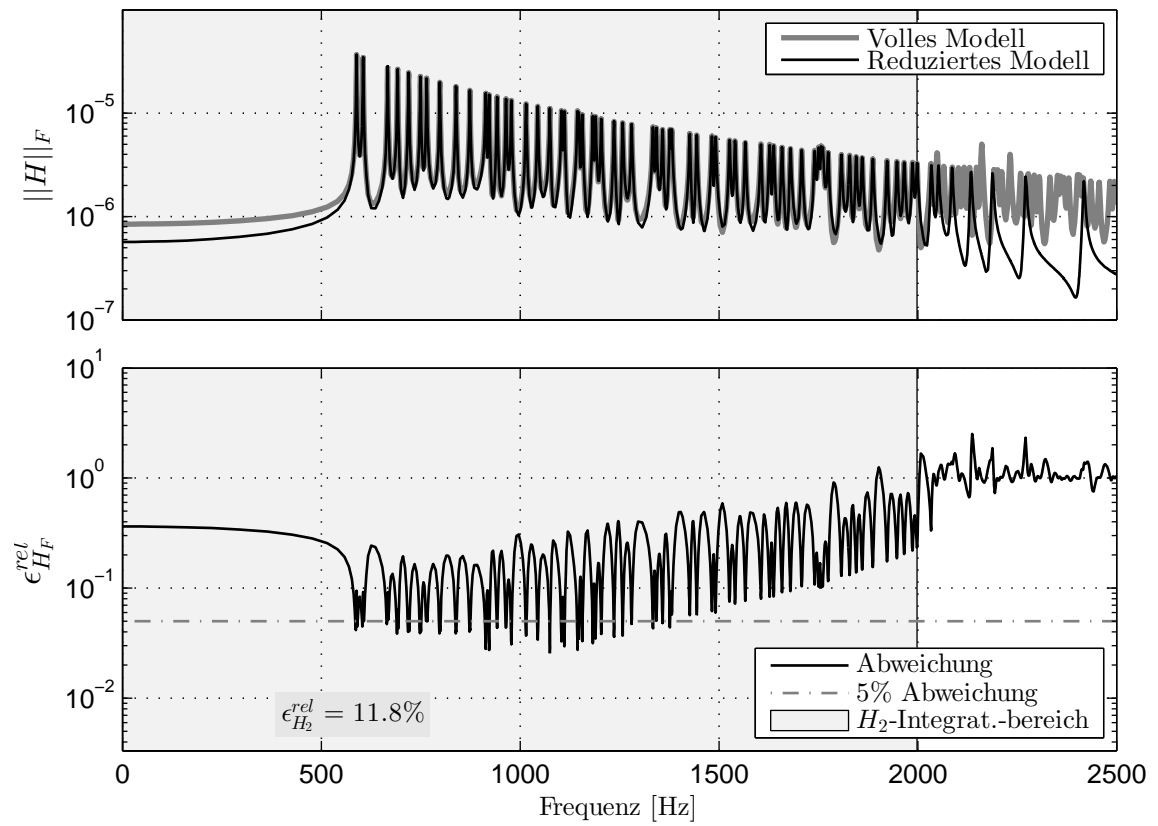


Diagramm 5.15: Frobenius-Norm der Übertragungsfunktionsmatrizen und relative Abweichung sowie \mathcal{H}_2 -Fehler für reduzierte Dimension $n = 76$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 2000]$ Hz (Getriebe-Fein)

5.3.2. TREIBSATZWELLE

Das Modell Treibsatzwelle, grob mit Hexaedern und fein mit Tetraedern vernetzt, wird in Abschnitt 5.1.3 vorgestellt. Es wird das kleinere System Welle-Grob und das größere System Welle-Fein untersucht.

Es werden die Reduktionsergebnisse für

- eine Entwicklungsstelle $f_{ES} = 0$ mit denen für
- mehrere äquidistant verteilte Entwicklungsstellen

verglichen. Es wird jeweils die rationale, interpolative SOAR-Funktion mit Gleichwichtung verwendet. Der Versuch mit $f_{ES} = 0$ ist auch mithilfe der modifizierten Arnoldi-Funktion erster Ordnung durchführbar.

Die Übertragungsfunktionsmatrix wird für die Treibsatzwellenmodelle über Massennormierung der Eigenmoden nach Gleichung 2.22 berechnet.

REDUKTIONSGÜTE IN ABHÄNGIGKEIT DER REDUZIERTEN SYSTEMDIMENSION FÜR EINZELNE ENTWICKLUNGSSTELLEN

Wie in Abschnitt 5.3.1 wird die Abhängigkeit der Reduktionsgüte von der reduzierten Systemdimension überprüft. Die Versuchsergebnisse für die beiden Wellen-Modelle sind in Diagramm 5.16 dargestellt. Der entsprechende Versuch mit den Getriebegehäuse-Modellen ist in Diagramm 5.10 angegeben. Es wird dabei die Entwicklungsstelle $f_{ES} = 0$ verwendet. Auf der Daten-CD sind Diagrammvideos der Frobeniusnorm der Übertragungsfunktionsmatrizen angegeben.

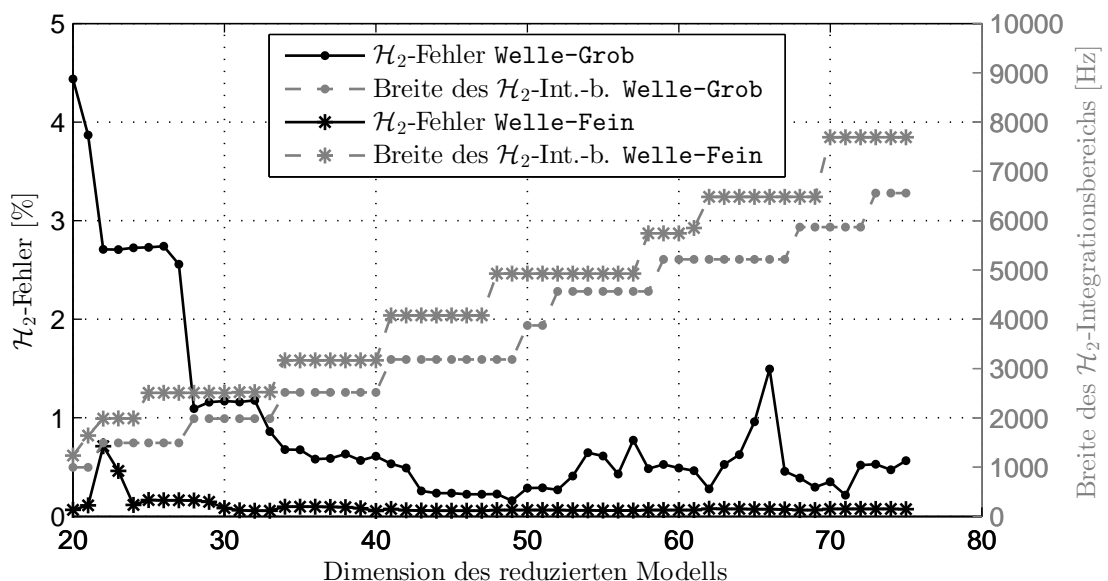


Diagramm 5.16: \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für variierende Dimension des reduzierten Systems n für die Modelle Welle-Grob und Welle-Fein

Diagramm 5.16 zeigt für die Modelle *Welle-Grob* und *Welle-Fein* monoton wachsende \mathcal{H}_2 -Integrationsbereiche mit größerem Integrationsbereich bei Modell *Welle-Fein*. Der \mathcal{H}_2 -Fehler ist für *Welle-Fein* kleiner als für *Welle-Grob*, ist in Diagramm 5.17 gesondert dargestellt und variiert im Gegensatz zu *Welle-Grob* kaum.

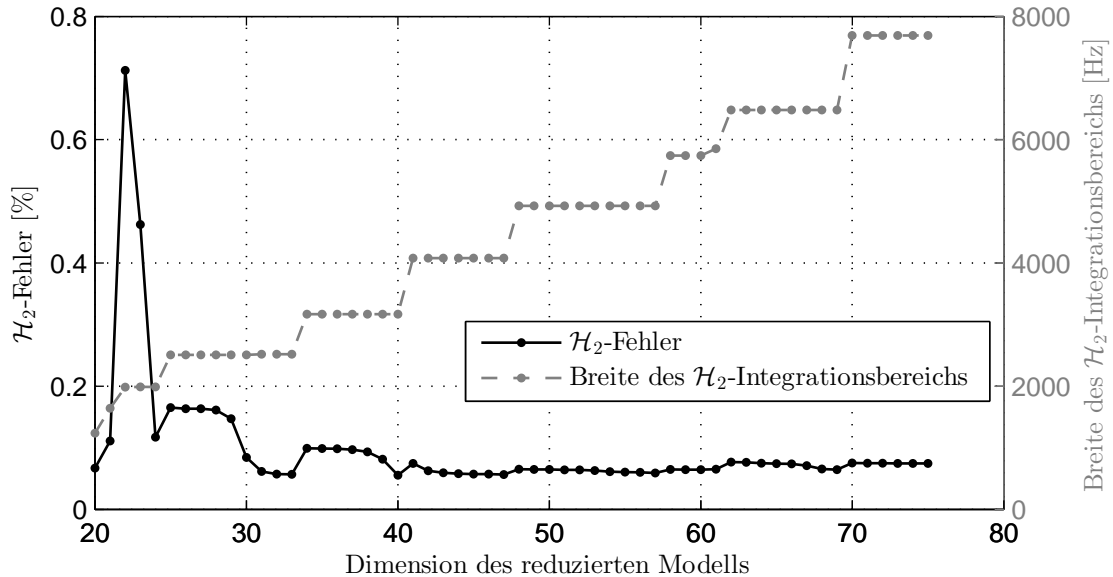


Diagramm 5.17: \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für variierende Dimension des reduzierten Systems n für das Modell *Welle-Fein*

Die Reduktion der Treibsatzwelle mit Entwicklungsstelle $f_{ES} = 0$ liefert ähnliche Ergebnisse zur Reduktion des Getriebegehäuses in Abschnitt 5.3.1. Aussagen über einzelne Entwicklungsstellen $f_{ES} > 0$ erfordern weitere Untersuchungen mit überarbeiteter Detektion zusammenhängend abgebildeter Eigenfrequenzen und damit bessere Bestimmung des \mathcal{H}_2 -Integrationsbereiches bei doppelten Eigenfrequenzen rotationssymmetrischer Geometrien.

KOMBINATION VON ENTWICKLUNGSTELLEN

Wie die Getriebegehäuse-Modelle werden die Treibsatzwellen-Modelle mithilfe von vier äquidistant verteilten Entwicklungsstellen reduziert. Es wird der abzubildende Frequenzbereich $f = [0 \dots 6500]$ Hz gewählt. Dieser Frequenzbereich wird mit Reihenentwicklung um Entwicklungsstelle $f_{ES} = 0$ von Modell *Welle-Grob* mit reduzierter Systemdimension $n = 75$ erreicht. Da die automatische Detektion des abgebildeten Frequenzbereiches auf Grundlage der Eigenfrequenzübereinstimmung bei rotationssymmetrischer Struktur eine Überarbeitung bedarf, wird konservativ die reduzierte Systemdimension $n = 76$ gewählt. Für die Getriebegehäuse-Modelle ist gezeigt worden, dass eine kleinere reduzierte Systemdimension möglich ist. So wird überprüft, ob die Reduktion der Treibsatzwellen-Modelle mit Kombination von Entwicklungsstellen grundsätzlich möglich ist. Auf die Berechnung der Übertragungsfunktionsmatrizen haben die doppelten Eigenfrequenzen der rotationssymmetrischen Struktur keinen Einfluss.

Die Frobenius-Norm der Übertragungsfunktionsmatrizen, der Frobenius- und der \mathcal{H}_2 -Fehler des Modells **Welle-Grob** ist in Diagramm 5.19 angegeben. Die zugehörigen NTFD-Abweichungen der Eigenfrequenzen sind in Diagramm 5.18 abgebildet. Der \mathcal{H}_2 -Integrationsbereich wird nicht anhand der Übereinstimmung der Eigenfrequenzen gewählt, sondern auf den Frequenzbereich $f = [0 \dots 6500]$ Hz festgelegt. Die Ergebnisse werden mit denen aus Diagramm 5.16 verglichen.

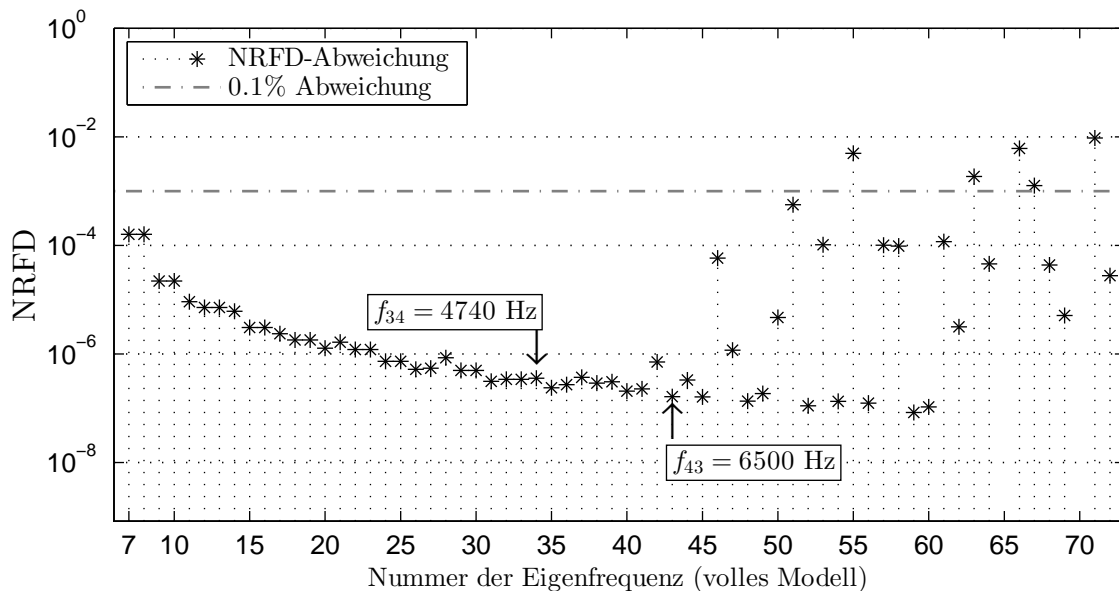


Diagramm 5.18: NTFD der Eigenfrequenzen für reduzierte Dimension $n = 76$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 6500]$ Hz (**Welle-Grob**)

Diagramm 5.18 zeigt im betrachteten Frequenzbereich sehr gute Übereinstimmung der Eigenfrequenzen. Die angegebenen Eigenfrequenzwerte zeigen nicht den Bereich zusammenhängender Eigenfrequenzen, sondern sind zur Abschätzung des betrachteten Frequenzbereiches angegeben. Nach der Übereinstimmung der Eigenfrequenzen zu urteilen, ist die Kombination mehrerer Entwicklungsstellen auf das Modell **Treibsatzwelle-Grob** möglich.

Entgegen der Übereinstimmung der Eigenfrequenzen zeigt der Übertragungsfunktionsvergleich in Diagramm 5.19 nur Übereinstimmung mittlerer Güte im betrachteten Frequenzbereich. Der \mathcal{H}_2 -Fehler beträgt mit $\epsilon_{\mathcal{H}_2}^{rel} = 11.1\%$ gegenüber dem \mathcal{H}_2 -Fehler bei Reihenentwicklung um $f_{ES} = 0$ das Zwanzigfache.

Um das Verhalten des \mathcal{H}_2 -Fehlers weiter zu untersuchen, wird die reduzierte Systemdimension auf $n = 100$ erhöht. Es werden wieder vier Entwicklungsstellen im Bereich $f = [0 \dots 6500]$ Hz verwendet. Der Versuch wird an beiden Treibsatzwellen-Modellen durchgeführt. Die Frobenius-Norm der Übertragungsfunktionsmatrizen, der Frobenius- und der \mathcal{H}_2 -Fehler des Modells **Welle-Grob** ist in Diagramm 5.20 und des Modells **Welle-Fein** in Diagramm 5.21 dargestellt. Die zugehörigen NTFD-Abweichungen der Eigenfrequenzen sind im Anhang in den Diagrammen A.27 und A.28 abgebildet. Die Ergebnisse werden mit denen aus Diagramm 5.16 verglichen.

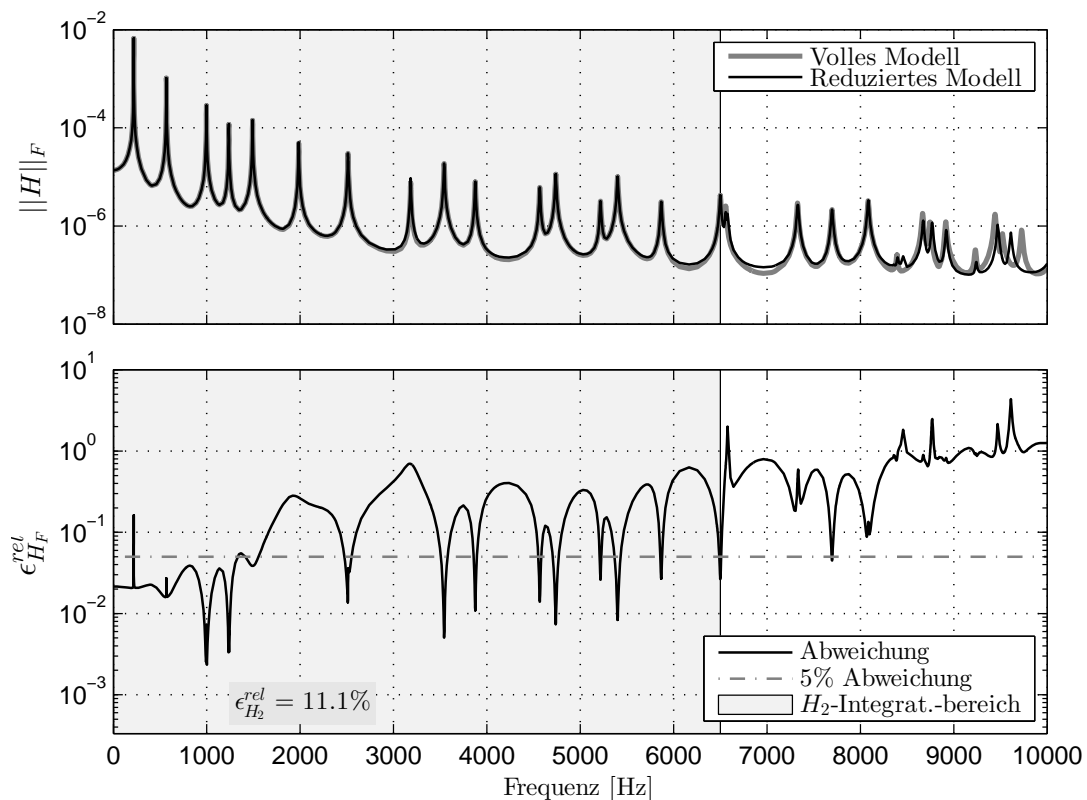


Diagramm 5.19: Frobenius-Norm der Übertragungsfunktionsmatrizen und relative Abweichung sowie \mathcal{H}_2 -Fehler für reduzierte Dimension $n = 76$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 6500]$ Hz (Welle-Grob)

Wie für die reduzierte Systemdimension $n = 76$ zeigt der Eigenfrequenzvergleich Übereinstimmung bei beiden Modellen im betrachteten Frequenzbereich. Nach Diagramm 5.20 ist der \mathcal{H}_2 -Fehler des Modells **Welle-Grob** mit $n = 100$ im Vergleich zu Diagramm 5.19 mit $n = 76$ nur wenig gesunken.

Auch der \mathcal{H}_2 -Fehler des Modells **Welle-Fein** in Diagramm 5.21 ist im betrachteten Frequenzbereich mit $\epsilon_{\mathcal{H}_2}^{rel} = 38.2\%$ sehr groß. Der Wert mit einer Entwicklungsstelle $f_{ES} = 0$ und reduzierter Systemdimension $n = 69$ beträgt $\epsilon_{\mathcal{H}_2}^{rel} = 0.064\%$. Der \mathcal{H}_2 -Fehler ist trotz größerer reduzierter Systemdimension um ein Vielfaches größer als der mit einzelner Entwicklungsstelle $f_{ES} = 0$.

Es besteht für beide Treibsatzwellen-Modelle eine Diskrepanz zwischen den Fehlerbewertungskriterien NRFD-Abweichung und \mathcal{H}_2 -Fehler für die Kombination von Entwicklungsstellen. Dies kann durch

- nicht ausreichende Abbildung des Übertragungsverhaltens des vollen Systems bei der Reduktion mit Entwicklungsstellen $f_{ES} > 0$ oder
- Fehler bei der Berechnung der Übertragungsfunktionsmatrizen aus Eigenmoden doppelter Eigenfrequenzen

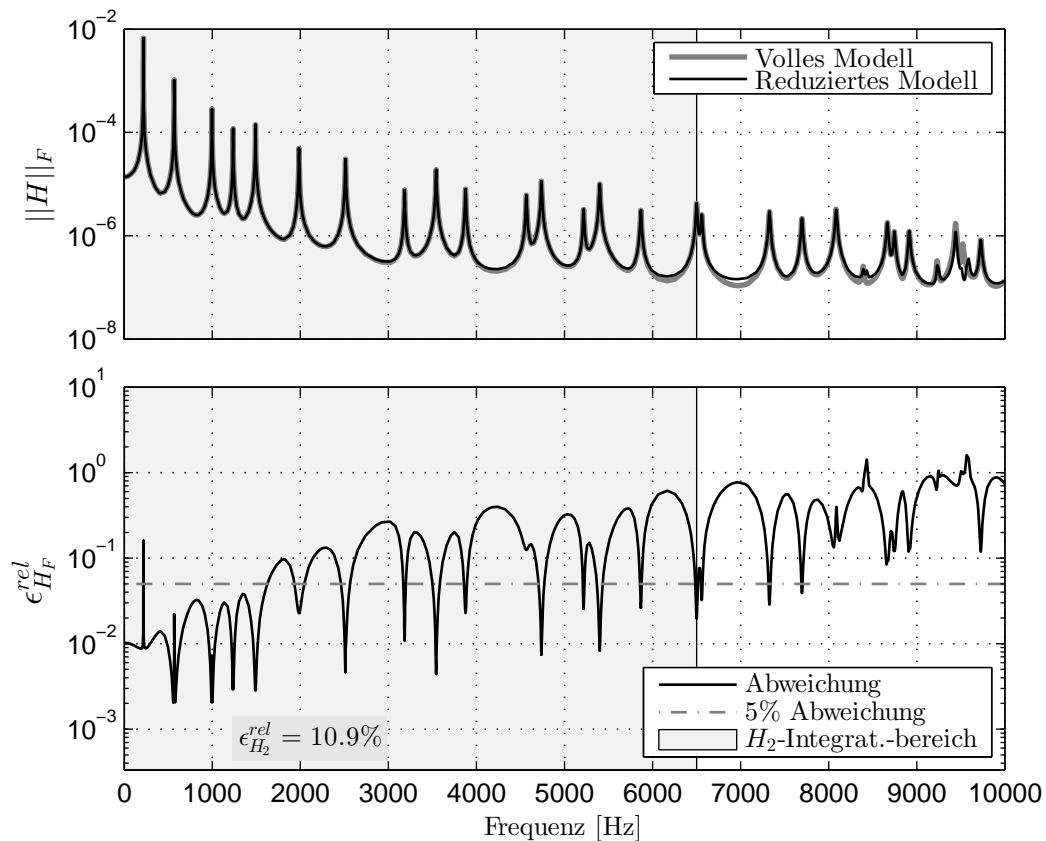


Diagramm 5.20: Frobenius-Norm der Übertragungsfunktionsmatrizen und relative Abweichung sowie \mathcal{H}_2 -Fehler für reduzierte Dimension $n = 100$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 6500]$ Hz (**Welle-Grob**)

bedingt sein. Überhöhte \mathcal{H}_2 -Fehler treten bei einer Entwicklungsstelle $f_{ES} = 0$ nicht auf. Dies spricht gegen die fehlerhafte Berechnung der Übertragungsfunktionsmatrizen. Die Reduktion rotationssymmetrischer Strukturen bedarf weiterer Untersuchungen.

Die Reduktion der Treibsatzwellen-Modelle mit einer Entwicklungsstelle $f_{ES} = 0$ ist problemlos möglich. Für beide Treibsatzwellen-Modelle führt die Kombination von Entwicklungsstellen zu guter Übereinstimmung der Eigenfrequenzen und großer Abweichung des Übertragungsverhaltens. Die Eingrenzung dieser Diskrepanz auf Fehler bei der Reduktion oder des Fehlerbewertungskriteriums sind weitere Untersuchungen notwendig. Die kleinstmögliche Reduktionsdimension bei Kombination von Entwicklungsstellen kann ebenfalls Gegenstand weiterer Untersuchungen sein.

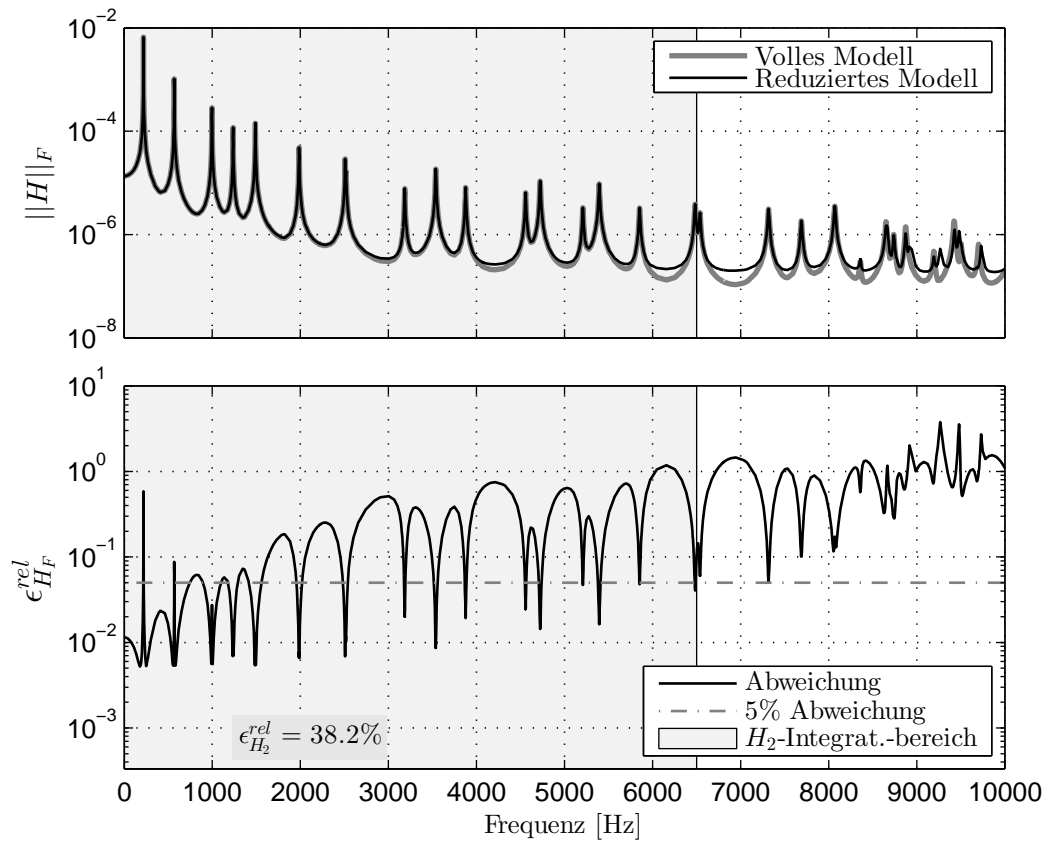


Diagramm 5.21: Frobenius-Norm der Übertragungsfunktionsmatrizen und relative Abweichung sowie \mathcal{H}_2 -Fehler für reduzierte Dimension $n = 100$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 6500]$ Hz (Welle-Fein)

5.4. AUSWERTUNG

Es werden die implementierten Funktionen und Korrelationskriterien auf jeweils zwei Modelle einer Rahmengeometrie, eines Getriebegehäuses und einer Treibsatzwelle angewendet.

Die Funktionalitäten

- Eliminierung der Starrkörperbewegung und
- einfache Reorthogonalisierung

werden validiert.

Es wird der Einfluss der Vernetzung auf das Reduktionsergebnis untersucht. Es wird festgestellt,

- dass das tetraedervernetzte Rahmen-Modell bessere Ergebnisse als das hexaedervernetzte Rahmen-Modell liefert,
- dass die Netzfeinheit der beiden Getriebegehäuse-Modelle keinen Einfluss auf die Reduktionsgüte hat und
- dass die feiner, tetraedervernetzte Treibsatz-Welle bessere Ergebnisse bei einer Entwicklungsstelle $f_{ES} = 0$ liefert.

Anhand der Rahmen-Modelle wird das rationale, interpolative SOAR-Verfahren und das rationale Block-SOAR-Verfahren untersucht. Es wird Gleichwichtung als Ansatz des interpolativen Startvektors gewählt und gezeigt, dass damit Ergebnisse mittlerer Güte erzielt werden. Werden gleiche reduzierte Systemdimensionen n verglichen, sind mithilfe des rationalen, interpolativen SOAR-Verfahrens mit Gleichwichtung gegenüber dem rationalen Block-SOAR-Verfahrens bessere Reduktionsergebnisse erzielbar. Entwicklungsstellen $f_{ES} > 0$ sind außerhalb von Eigenfrequenzen $f_{ES} \neq f_i$ zu wählen.

Mithilfe der Kombination von $k = [3 \dots 9]$ äquidistant verteilten Entwicklungsstellen $f_{ES_i} > 0$ gleicher Reihenentwicklungsordnung sind bessere Reduktionsergebnisse erzielbar, als mit der Reihenentwicklung um eine Entwicklungsstelle $f_{ES} = 0$. Die Kombination von Entwicklungsstellen führt zu

- guter Definierbarkeit eines abzubildenden Frequenzbereiches,
- Verringerung der reduzierten Systemdimension um 35% (Modell Getriebegehäuse) bei gleichem abgebildeten Frequenzbereich,
- steigendem Berechnungsaufwand mit zunehmender Anzahl von Entwicklungsstellen durch zu wiederholende LU-Faktorisierung nach Abschnitt 3.4.2 und
- größerem Arbeitsspeicherbedarf durch Verwendung der LU- anstelle der Cholesky-Faktorisierung nach Abschnitt 3.4.1 (bei Entwicklungsstelle $f_{ES} = 0$).

Im Gegensatz zur rationalen Kombination von Entwicklungsstellen liefert das modifizierte Arnoldi-Verfahren mit $f_{ES} = 0$ gute Reduktionsergebnisse für alle verwendeten Modelle, wobei die Systemeigenschaften der rotationssymmetrischen Treibsatzwellen-Modelle genauer zu untersuchen sind.

Es werden für Reduktion mittels KSM die beiden Funktionen

- rationale, interpolative SOAR-Funktion mit $k = [3 \dots 9]$ äquidistant verteilten Entwicklungsstellen und
- modifizierte Arnoldi-Funktion erster Ordnung (entspricht Entwicklungsstelle $f_{ES} = 0$)

empfohlen. Die Vorteile beider Funktionen sind in Tabelle 5.8 zusammengefasst.

Tabelle 5.8.: wesentliche Vorteile der empfohlenen KSM-Funktionen

rationale, interpolative SOAR- Funktion mit $k = [3 \dots 9]$ Entwicklungsstellen	modifizierte Arnoldi-Funktion erster Ordnung
<ul style="list-style-type: none"> • definierbarer abzubildender Frequenzbereich • kleine reduzierte Systemdimension 	<ul style="list-style-type: none"> • kleine Berechnungszeit • kleiner Arbeitsspeicherbedarf • lückenlos abgebildeter Frequenzbereich

Da für iterativ rationale SOAR-Verfahren der Berechnungsaufwand für jeden Iterationsschritt gleich groß ist, ist aus dem Gesichtspunkt der Berechnungszeit die Implementierung von iterativ rationalen Verfahren nicht sinnvoll. Solche Verfahren suchen iterativ nach einer optimalen Wahl von Entwicklungsstellen. Zur entsprechenden Weiterentwicklung ist der Einfluss der Verteilung von Entwicklungsstellen im Frequenzbereich genauer zu untersuchen.

Alternativ besteht die Möglichkeit der Weiterentwicklung des rationalen, interpolativen SOAR-Verfahrens mit äquidistant verteilten Entwicklungsstellen auf Grundlage von Konvergenz. Dazu wird das System auf eine große, reduzierte Dimension reduziert, die für den abzubildenden Frequenzbereich mit Sicherheit ein gutes Reduktionsergebnis erzielt. Ausgehend von diesem Ergebnis können mit geringem Aufwand kleinere Reduktionsdimensionen erzeugt werden. Konvergiert das Übertragungsverhalten im abzubildenden Frequenzbereich gegen die Ausgangslösung, ist diese als Referenzlösung validiert. Ausgehend von der Referenzlösung kann ein System kleinerer reduzierter Dimension mit vorgegebener Reduktionsgüte erzeugt werden. Die Fehlerbewertung dieser Konvergenz-Reduktion ist damit ohne die aufwändige Berechnung der Eigenlösung des vollen Modells möglich.

Die Validierung der Reduktionsfunktionen an weiteren Modellen, insbesondere die weiterführende Untersuchung an rotationssymmetrischen Geometrien ist notwendig. Die Charakteristik optimaler Interpolation ist zu untersuchen.

6. ZUSAMMENFASSUNG UND AUSBLICK

6.1. ZUSAMMENFASSUNG

Es wird fehlerüberwachte Modellordnungsreduktion auf Grundlage von Krylov-Unterraumverfahren (KSM) entwickelt und auf strukturmechanische Modelle angewendet. Dabei liegt ein Schwerpunkt auf dem Lösen linearer Gleichungssysteme (LGS) mithilfe der Software MATLAB. Die entwickelten Funktionen werden in das MATLAB-basierte MORPACK der Professur für Dynamik und Mechanismentechnik der TU-Dresden integriert.

Eine vorliegende KSM-Funktion auf Grundlage des Arnoldi-Algorithmus erster Ordnung wird um die Wahl eines geeigneten Startvektors und einfache Reorthogonalisierung erweitert. Es wird ein rationales, interpolatives und ein rationales Block-SOAR-Verfahren entworfen und implementiert. Im Gegensatz zum modifizierten Arnoldi-Verfahren mit Reihenentwicklung um $f_{ES} = 0$ ist bei den rationalen Verfahren die Wahl einer beliebigen Anzahl von Entwicklungsstellen möglich. Die implementierten Funktionen werden validiert. Vergleichende Reduktionen an kleinen Versuchsmodellen ergeben für die interpolative Funktion mit Gleichwichtung der Eingangsmatrix auf einen Startvektor Ergebnisse durchschnittlicher Güte. Bei gleicher reduzierter Systemdimension ist mithilfe der interpolativen Funktion ein besseres Reduktionsergebnis erzielbar, als mit der Block-Funktion. Mit Kombination von $k = [3 \dots 9]$ im Frequenzbereich äquidistant verteilten Entwicklungsstellen ist die reduzierte Systemdimension bei gleichem abgebildeten Frequenzbereich deutlich kleiner als mit dem modifizierten Arnoldi-Verfahren erster Ordnung. Das Ergebnis wird am Beispiel eines Getriebegehäuses validiert. Für eine rotationssymmetrische Treibsatzwelle liefert das modifizierte Arnoldi-Verfahren bessere Reduktionsergebnisse. Es wird die Verwendung des rationalen, interpolativen SOAR-Verfahrens für kleine reduzierte Systemdimensionen und des modifizierten Arnoldi-Verfahrens für kleinen Berechnungsaufwand empfohlen.

Die Fehlerbewertung wird durch das eigenfrequenzbezogene NFRD-Kriterium zur Bestimmung des übereinstimmenden Frequenzbereiches zwischen vollem und reduzierten System und den modifizierten \mathcal{H}_2 -Fehler zu Bewertung des Übertragungsverhaltens in diesem Bereich vorgenommen. Auf den \mathcal{H}_2 -Fehler aus den Übertragungsfunktionsmatrizen, auf voller Systemdimension normiert, können dabei für die Systemeingänge dominante Eigenmoden außerhalb des betrachteten Frequenzbereiches einen großen Einfluss haben. Es wird alternativ die Normierung des Einganganteils der Eigenmoden vorgeschlagen.

Für das Lösen linearer LGS werden direkte vor iterativen Verfahren aufgrund größerer Leistungsfähigkeit und Stabilität bei Anwendung auf KSM bevorzugt. Für positiv definite Systeme (modifizierter Arnoldi) wird die Cholesky-Zerlegung (`chol`-Funktion) und für andere Systeme die aufwändigere LU-Zerlegung (`lu`-Funktion), jeweils mit Lösen durch direkte Substitution, vorgeschlagen. Große Ressourceneinsparung ist durch geeignete Permutation, bevorzugt die `symamd`-Funktion, möglich. Grundsätzlich ist die Zerlegung von Systemmatrizen volumenvernetzter, stabförmiger Modelle mit hohem Aufwand verbunden und unter Umständen durch alternative Permutation verbesserbar.

Es stehen mit der modifizierten Arnoldi-Funktion erster Ordnung und der rationalen, interpolativen SOAR-Funktion mit $k = [3 \dots 9]$ äquidistant verteilten Entwicklungsstellen zwei leistungsstarke KSM-Verfahren für verschiedene Anwendungsfälle zur Verfügung. Die bestgeeigneten linearen LGS-Löser der Software MATLAB als Hauptoperation der KSM sind als Funktionen in MORPACK implementiert. Es steht ein Verfahren zur Bestimmung des abgebildeten Frequenzbereiches mit dem \mathcal{H}_2 -Fehler als skalaren Fehlerwert dieses Bereiches zur Verfügung.

6.2. AUSBLICK

Es sind weitere Untersuchungen zur Validierung der implementierten KSM-Funktionen notwendig. Speziell ist die Reduktion und Fehlerbewertung für rotationssymmetrische Modelle zu validieren und gegebenenfalls weiterzuentwickeln. Zur Lösung von LGS ist zu untersuchen, für welche Modelle die Cuthill-McKee-Permutation gute Faktorisierungsergebnisse erzielt.

An der rationalen, interpolativen SOAR-Funktion ist

- die Einbeziehung der Dämpfung
- die optimale Wahl eines interpolativen Startvektors und
- die optimale Wahl und Kombination von Entwicklungsstellen

weiter zu untersuchen.

Die Weiterentwicklung der rationalen, interpolativen SOAR-Funktion ist als

1. (rechenintensive) iterativ rationale, interpolative Funktion zur iterativen Bestimmung optimaler Entwicklungsstellen nach [Feh11] und als
2. Konvergenz-Funktion auf Grundlage schrittweiser Verkleinerung des Reduktionsergebnisses nach Abschnitt 5.4

möglich.

Weiter ist die Einbeziehung des komplexen Anteils bei den rationalen SOAR-Verfahren ($s_{ES_k} = i \cdot f_{ES}$ in Algorithmus 2.4, Zeile 3) und die Möglichkeit externer Bibliotheken zum Lösen von LGS zu untersuchen.

ABBILDUNGSVERZEICHNIS

2.1.	Orthogonalisierung anhand zweier Vektoren im zweidimensionalen Raum . .	13
3.1.	Besetzungsstruktur der Koeffizientenmatrix \mathbf{A} ($\hat{=}\mathbf{K}$) und ihrer <code>chol</code> -Zerlegung \mathbf{L} mit Permutation nach ANSYS und grünem ‚fillin‘ ; $\text{NNE}(L) = 19559$	35
3.2.	Besetzungsstruktur der Koeffizientenmatrix \mathbf{A} ($\hat{=}\mathbf{K}$) und ihrer <code>chol</code> -Zerlegung \mathbf{L} mit Cuthill-McKee-Permutation und grünem ‚fillin‘ ; $\text{NNE}(L) = 17239$	36
3.3.	Besetzungsstruktur der Koeffizientenmatrix \mathbf{A} und ihrer <code>chol</code> -Zerlegung \mathbf{L} für maximalen und minimalen ‚fillin‘ (grün)	36
3.4.	Besetzungsstruktur der Koeffizientenmatrix \mathbf{A} ($\hat{=}\mathbf{K}$) und ihrer <code>chol</code> -Zerlegung \mathbf{L} mit <code>symamd</code> -Permutation und grünem ‚fillin‘ ; $\text{NNE}(L) = 10793$	37
4.1.	Aufbau von MORPACK in Modulen und deren Bedeutung für die Implementierung der KSM	42
4.2.	Grundgerüst des MOR-Moduls mit Reduktionsfunktionen	43
4.3.	Entscheidungskriterien zur Wahl von Lösungsverfahren für LGS, Nummerierung der Verfahren entsprechend Tabelle 4.4	54
5.1.	Vernetztes Modell Rahmen-Hex (Hexaedervernetzung) mit Anbindungsknoten und deren Verbindung über MPC mit der Geometrie	57
5.2.	Vernetztes Modell Rahmen-Tet (Tetraedervernetzung) mit Anbindungsknoten und deren Verbindung über MPC mit der Geometrie	57
5.3.	Vernetztes Modell des Getriebegehäuses Getriebe-Grob	58
5.4.	Vernetztes Modell der Treibsatzwellen Welle-Grob und Welle-Fein	59

TABELLENVERZEICHNIS

2.1. Unterscheidungsmerkmale Finite Elemente Methode (FEM) und Mehrkörpersimulation (MKS)	6
2.2. Klassifizierungskriterien für Krylov-Unterraummethoden	9
3.1. Vergleich Direkter und Iterativer Lösungsverfahren linearer Gleichungssysteme	24
3.2. Iterative Methoden zur Lösung linearer Gleichungssysteme in MATLAB . . .	26
3.3. Permutationsfunktionen in Matlab	35
4.1. Optionen der Arnoldi-Funktion erster Ordnung	45
4.2. Optionen der rationalen SOAR-Funktionen	46
4.3. Anwendung der Lösungs-Funktionen für LGS in MORPACK nach Algorithmus 4.1	51
4.4. Übersicht über die Optionen der einzelnen LGS-Löser-Funktionen	53
5.1. Strukturmechanische Modelle zum Test der Reduktionsverfahren	56
5.2. Geometrie- und Netzdaten der Rahmen-Modelle	58
5.3. Geometrie- und Netzdaten der Getriebegehäuse-Modelle	59
5.4. Geometrie- und Netzdaten der Treibsatzwellen-Modelle	59
5.5. Größte Eigenfrequenz unterhalb der 0.1%-Schranke, \mathcal{H}_2 -Integrationsbereich und relativer \mathcal{H}_2 -Fehler für Tetraeder- und Hexaedervernetzung für verschiedene reduzierte Systemdimensionen	63
5.6. Parameterwahl und Ergebnisse des Testes mit zufälligen Startvektoren, einzelnen Eingängen und gleichgewichteter Interpolation am tetraedervernetzten Rahmenmodell Rahmen-Tet	64
5.7. Reduktionsergebnisse für verschiedene Eingangskombinationen bei Block-SOAR-Verfahren	66
5.8. wesentliche Vorteile der empfohlenen KSM-Funktionen	85

DIAGRAMMVERZEICHNIS

3.1.	Anzahl der NNE der Matrix \mathbf{L} für variierende Droptoleranz bei der unvollständigen Cholesky-Faktorisierung und der Matrizen \mathbf{L} und \mathbf{U} bei der unvollständigen LU-Faktorisierung	29
3.2.	Iterationsschritte des pcg-Verfahrens bei variierender Droptoleranz der unvollständigen Cholesky-Faktorisierung	30
3.3.	„fillin“ der vollständigen Cholesky-Zerlegung für verschiedene Dimensionen der Koeffizienten-Matrix \mathbf{A} , die dem Freiheitsgrad der Systeme entsprechen	31
3.4.	Anzahl der Nicht-Null-Elemente der Dreiecksmatrix \mathbf{L} gegenüber denen der Koeffizienten-Matrix \mathbf{A}	32
3.5.	Vielfaches der Zeit für vollständig ausgeführte <code>ilu</code> - gegenüber <code>ichol</code> -Funktion in MATLAB	34
3.6.	Berechnungszeit der vollständig ausgeführten <code>ichol</code> -Funktion für MATLAB-Permutationen im Verhältnis zu denen mit Standard-ANSYS-Permutation	37
3.7.	Anzahl der NNE des <code>chol</code> -Faktors \mathbf{L} für MATLAB-Permutationen im Verhältnis zu der mit Standard-ANSYS-Permutation	38
4.1.	NRFD-Korrelationskriterium dargestellt in einem Diagramm mit Fehlerschranke und letzter Eigenfrequenz unterhalb der Fehlerschranke	47
4.2.	Korrelationskriterium modMAC der Eigenvektoren eines reduzierten Systems mit denen des vollen Systems	48
4.3.	Frequenzgangbezogene Korrelationskriterien (frequenzabhängige Frobeniusnorm der Übertragungsfunktionsmatrizen und deren relativer Abweichung sowie \mathcal{H}_2 -Norm) für übereinstimmenden Systemverhalten in mittlerem Frequenzbereich (\mathcal{H}_2 -Integrationsbereich aus Übereinstimmung der Eigenfrequenzen)	49
5.1.	Korrelationskriterium modMAC eines reduzierten Systems mit und ohne Orthogonalisierung zu den Starrkörpermoden	61
5.2.	Mittlere lineare Abhängigkeit der Krylov-Moden von den vorherigen Krylov-Moden (Mittelwert von h in Algorithmus 2.5) bei Reduktion auf Dimension 50 (Rahmen-Tet)	61
5.3.	NRFD der Reduktion des hexaeder- und tetraedervernetzten Modells mit 30 Krylov-Moden	62
5.4.	Abweichung des mithilfe von 30 Krylov-Moden reduzierten, hexaeder- und tetraedervernetzten Rahmenmodells (mod. Arnoldi-Verfahren erster Ordnung)	63

5.5. klassierte \mathcal{H}_2 -Abweichung bei 1000 zufällig interpolierten Startvektoren im Vergleich zu 36 Startvektoren aus einzelnen Eingängen (Rahmen-Tet)	65
5.6. \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für verschiedene Entwicklungsstellen bei reduzierter Dimension von $n = 25$ des Rahmenmodells (Rahmen-Tet)	67
5.7. \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für variierende Dimension des reduzierten Systems n (Rahmen-Tet)	68
5.8. \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für aufsteigende, reduzierte Systemdimension bei Entwicklungsstelle $f_{ES} = 979\text{Hz}$ (Rahmen-Tet)	69
5.9. Notwendige reduzierte Systemdimension zur Abbildung des Frequenzbereiches $f = [0 \dots 2000]\text{Hz}$ für verschiedene Anzahl äquidistant verteilter Entwicklungsstellen und zugehörige \mathcal{H}_2 -Fehler	70
5.10. \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für variierende Dimension des reduzierten Systems n für die Modelle Getriebe-Grob und Getriebe-Fein mit einer Entwicklungsstelle $f_{ES} = 0$	73
5.11. \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für variierende Dimension des reduzierten Systems n für das Modell Getriebe-Grob und Entwicklungsstelle $f_{ES} = 1300\text{Hz}$	74
5.12. Notwendige reduzierte Systemdimension zur Abbildung des Frequenzbereiches $f = [0 \dots 2000]\text{Hz}$ für verschiedene Anzahl linear verteilter Entwicklungsstellen und zugehörige \mathcal{H}_2 -Fehler (Getriebe-Grob)	75
5.13. Frobenius-Norm der Übertragungsfunktionsmatrizen und relative Abweichung sowie \mathcal{H}_2 -Fehler für reduzierte Dimension $n = 76$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 2000]\text{Hz}$ (Getriebe-Grob)	76
5.14. NRFD der Eigenfrequenzen für reduzierte Dimension $n = 72$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 2000]\text{Hz}$ (Getriebe-Grob)	76
5.15. Frobenius-Norm der Übertragungsfunktionsmatrizen und relative Abweichung sowie \mathcal{H}_2 -Fehler für reduzierte Dimension $n = 76$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 2000]\text{Hz}$ (Getriebe-Fein)	77
5.16. \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für variierende Dimension des reduzierten Systems n für die Modelle Welle-Grob und Welle-Fein . .	78
5.17. \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für variierende Dimension des reduzierten Systems n für das Modell Welle-Fein	79
5.18. NRFD der Eigenfrequenzen für reduzierte Dimension $n = 76$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 6500]\text{Hz}$ (Welle-Grob) . .	80
5.19. Frobenius-Norm der Übertragungsfunktionsmatrizen und relative Abweichung sowie \mathcal{H}_2 -Fehler für reduzierte Dimension $n = 76$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 6500]\text{Hz}$ (Welle-Grob)	81
5.20. Frobenius-Norm der Übertragungsfunktionsmatrizen und relative Abweichung sowie \mathcal{H}_2 -Fehler für reduzierte Dimension $n = 100$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 6500]\text{Hz}$ (Welle-Grob)	82

5.21. Frobenius-Norm der Übertragungsfunktionsmatrizen und relative Abweichung sowie \mathcal{H}_2 -Fehler für reduzierte Dimension $n = 100$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 6500]$ Hz (Welle-Fein)	83
--	----

LITERATURVERZEICHNIS

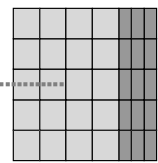
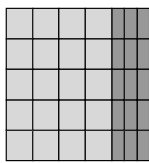
- [Arn51] W. E. Arnoldi. The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem. *Quarterly of Applied Mathematics*, (9):17–29, 1951.
- [Bai05] Bai, Z., Meerbergen K., Su, Y. Arnoldi methods for structure-preserving dimension reduction of second-order dynamical systems, 2005. Department of Computer Science and Department of Mathematics, University of California.
- [Bet97] Josef Betten. *Finite Elemente für Ingenieure*. Springer-Lehrbuch. Springer, Berlin [u.a.], 1997.
- [BGKVV07] A. Bunse-Gerstner, D. Kubalińska, G. Vossen, and D. Wilczek. h2-norm optimal model reduction for large scale discrete dynamical MIMO systems. *Journal of Computational and Applied Mathematics*, 233(5):1202–1216, 2007.
- [Bör13] Steffen Börm. Iterative Lösungsverfahren für große lineare Gleichungssysteme, 2013. URL: <http://www.informatik.uni-kiel.de/sb/data/GLGS.pdf> (Stand: 22.02.2014).
- [CM69] E. Cuthill and J. McKee. *Reducing the Bandwidth of Sparse Symmetric Matrices*. Defense Technical Information Center, Ft. Belvoir, 1969.
- [CTL09] C. C. Chu, H. C. Tsai, and M. H. Lai. Structure Preserving Model-Order Reductions of MIMO Second-Order Systems Using Arnoldi Methods. In *Mathematical and Computer Modelling*, volume 51, pages 956–973. 2009.
- [Dan14] Dankert, Prof. Dr.-Ing. habil. Jürgen. Lineare Gleichungssysteme, Matrixinversion, 2014. URL: <http://www.rzbt.haw-hamburg.de/dankert/WWWergVert/html/lineare-gleichungssysteme-mat.html> (Stand: 14.02.2014).
- [FE13] Michael Fischer and Peter Eberhard. Linear model reduction of large scale industrial models in elastic multibody dynamics. Springer Science+Business Media Dordrecht 2013, 01 2013.
- [Feh11] Jörg Fehr. *Automated and error controlled model reduction in elastic multibody systems*, volume 21 of *Schriften aus dem Institut für Technische und Numerische Mechanik der Universität Stuttgart*. Shaker, Aachen, 2011.

- [FP10] Hermann Friedrich and Frank Pietschmann. *Numerische Methoden: Ein Lehr- und Übungsbuch*. De Gruyter Studium. De Gruyter, Berlin, 2010.
- [Ges09] Christine Geschwinder. Implementierung und Test H2-optimaler Reduktionsverfahren für flexible Körper. Institut für Technische und Numerische Mechanik, Universität Stuttgart, 2009.
- [Get09] Pascal Getreuer. Writing fast Matlab Code, 2009. URL: <http://www.getreuer.info/matopt.pdf> (Stand: 13.02.2014).
- [GL80] Alan George and Joseph W-H Liu. *Computer solution of large sparse positive definite systems*. Prentice-Hall series in computational mathematics. Prentice-Hall, Englewood Cliffs and N.J, 1980.
- [Gra98] Henning Grafe. *Model updating of large structural dynamics models using measured response functions*. Department of Mechanical Engineering, Dynamics Section, London, South Kensington, 1998.
- [Gri97] Eric James Grimme. *Krylov projection methods for model reduction*. University of Illinois at Urbana Champaign, 1997.
- [GVD05] K. Gallivan, A. Vandendorpe, and P. V. Dooren. *Model reduction of MIMO systems via tangential interpolation*, volume 26 of *SIAM Journal on Matrix Analysis and Applications*. 2 edition, 2005.
- [HH05] Desmond J. Higham and Nicholas J. Higham. *MATLAB guide*. SIAM, Philadelphia and Pa, 2. ed edition, 2005.
- [Imm14] Florian Immel. Implementierung von Modellkorrelationsverfahren und Vergleich anhand gemessener und reduzierter Datensätze mit Anwendung auf strukturmechanische Modelle. Institut für Festkörpermechanik, Technische Universität Dresden, 2014.
- [Kou09] Panagiotis Koutsovasilis. *Model order reduction in structural mechanics: Coupling the rigid and elastic multi body dynamics*, volume Nr. 712 of *Fortschritt-Berichte VDI. Reihe 12, Verkehrstechnik/Fahrzeugtechnik*. VDI Verlag, Düsseldorf, 2009.
- [Kry31] A. N. Krylov, editor. *On the numerical solution of the equation by which in technical questions frequencies of small oscillations of material systems are determined*. 1931.
- [Lan50] C. Lanczos. An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators. *Journal of research of the National Bureau of Standards*, (45):255–282, 1950.
- [LB11] Claudius Lein and Michael Beitelschmidt. Untersuchung rollkontaktbedingter Triebstrangschwingungen einer vierachsigen Lokomotive. *Schwingungen in Antrieben, VDI*, (7):127–140, 2011.

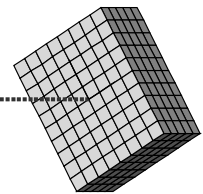
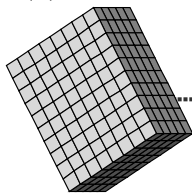
- [LB12] Claudius Lein and Michael Beitelschmidt. MORPACK-Schnittstelle zum Import von FE-Strukturen nach SIMPACK. *at - Automatisierungstechnik*, 60(9):547–560, 2012.
- [Leh07] Michael Lehner. *Modellreduktion in elastischen Mehrkörpersystemen*, volume Bd. 2007,10 of *Schriften aus dem Institut für Technische und Numerische Mechanik der Universität Stuttgart*. Shaker, Aachen, 2007.
- [Li13a] Xiaoye Sherry Li. Parallel Sparse Factorization, Triangular Solution, 2013. URL: <http://crd-legacy.lbl.gov/xiaoye/G2S3/Lecture-Notes/Direct-Parallel-5.pptx> (Stand: 13.02.2013).
- [Li13b] Xiaoye Sherry Li. Sparse Direct Method: Combinatorics, 2013. URL: <http://crd-legacy.lbl.gov/xiaoye/G2S3/Lecture-Notes/Direct-Combinatorial-3.pptx> (Stand: 13.02.2013).
- [LQK13] Claudius Lein, Volker Quarz, and Panagiotis Koutsovasilis. Einbindung elastischer Strukturen in Mehrkörpersysteme. *Technische Universität Dresden, Institut für Festkörpermechanik, Professur für Dynamik und Mechanismen-technik*, 2013.
- [Mei08] Andreas Meister. *Numerik linearer Gleichungssysteme: Eine Einführung in moderne Verfahren ; mit MATLAB-Implementierungen von C. Vömel*. Friedr. Vieweg & Sohn Verlag, Wiesbaden, 3., überarbeitete aufl edition, 2008.
- [RS10] Georg Rill and Thomas Schaeffer. *Grundlagen und Methodik der Mehrkörpersimulation*. Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden, Wiesbaden, 2010.
- [Saa03] Yousef Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia and Pa, 2nd ed edition, 2003.
- [Ste12] Ulrich Stein. *Programmieren mit MATLAB: Programmiersprache, grafische Benutzeroberfläche, Anwendungen*. Fachbuchverl. Leipzig, München, 4., überarb. und erw. aufl edition, 2012.
- [Tel12] Christian Telke. Erstellung und Implementierung reduzierbarer Verbindungselemente für die Schnittstellenmodellierung bei elastischen Mehrkörpersystemen. Institut für Festkörpermechanik, Technische Universität Dresden, 2012.

A. ANHANG

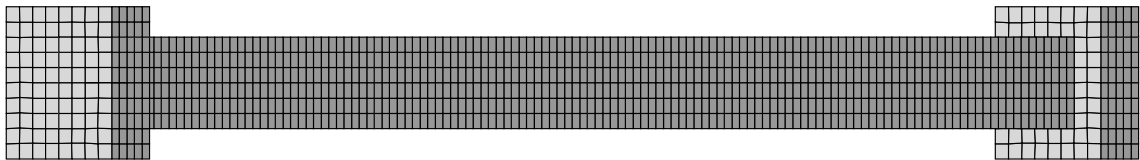
A.1. MODELLE ZUM TEST DES LÖSENS LINEARER GLEICHUNGSSYSTEME



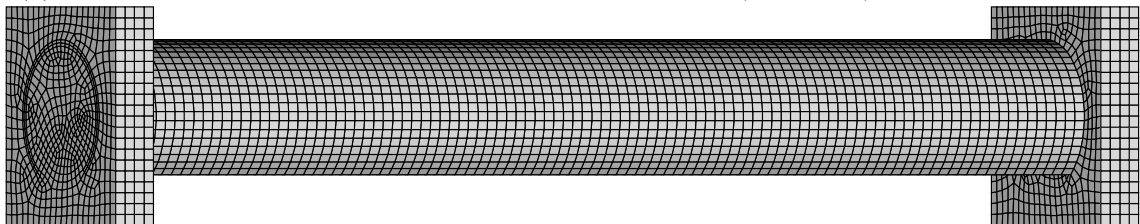
(a) Balken, mit zwei Solid-Elementblöcken und Balkenelementen vernetzt, Freiheitsgrad: 3432



(b) Balken, mit zwei Solid-Elementblöcken und Balkenelementen vernetzt, Freiheitsgrad: 13554



(c) Balken mit quadratischem Querschnitt, hexaedervernetzt (nur Solids), Freiheitsgrad: 60615



(d) Balken mit Kreisringquerschnitt, hexaedervernetzt (nur Solids), Freiheitsgrad: 215847

Abbildung A.1.: Vernetzte Balkengeometrien für mechanische Systeme zum Test von LGS

In Abschnitt 3 wird die Numerik linearer Gleichungssysteme betrachtet. Bei den dort verwendeten Modellen handelt es sich um die in Abbildung A.1 dargestellten. Die Systemmatrizen der ersten beiden Modelle (a und b), hauptsächlich mit Balkenelementen vernetzt, weisen im Bereich der Balkenelemente eine sehr kleine Bandbreite auf. Das dritte Modell (c) ist voll vernetzt, die Systemmatrizen weisen dementsprechend eine wesentlich größere Bandbreite als die ersten beiden Modelle auf. Beim vierten Modell (d) mit Kreisringquerschnitt wird eine große Fläche eingeschlossen, jedoch ist der Ring in Radiusrichtung nur dünn mit einem Element vernetzt.

A.2. VERGLEICH VERSCHIEDENER NORMIERUNGEN BEIM ERSTELLEN DER ÜBERTRAGUNGSMATRIZEN

Es wird erläutert, welchen Einfluss die verschiedenen Normierungstypen der Gleichungen 2.22 und 2.23 in Abschnitt 2.6.3 auf die Darstellung der Frobeniusnorm der Übertragungsfunktionsmatrizen und den \mathcal{H}_2 -Fehler haben. Bei der Massenwichtung in Dimension des vollen Systems (hierzu ist die Expansion der reduzierten Eigenvektoren notwendig) spielen dominante Moden außerhalb des Integrationsbereiches für den \mathcal{H}_2 -Fehler eine Rolle. Im gewählten Beispiel befinden sich diese Moden im Bereich $f = [800 \dots 1000]$ Hz. Diagramm A.1 zeigt ein reduziertes System ohne gute Näherung der dominanten Moden.

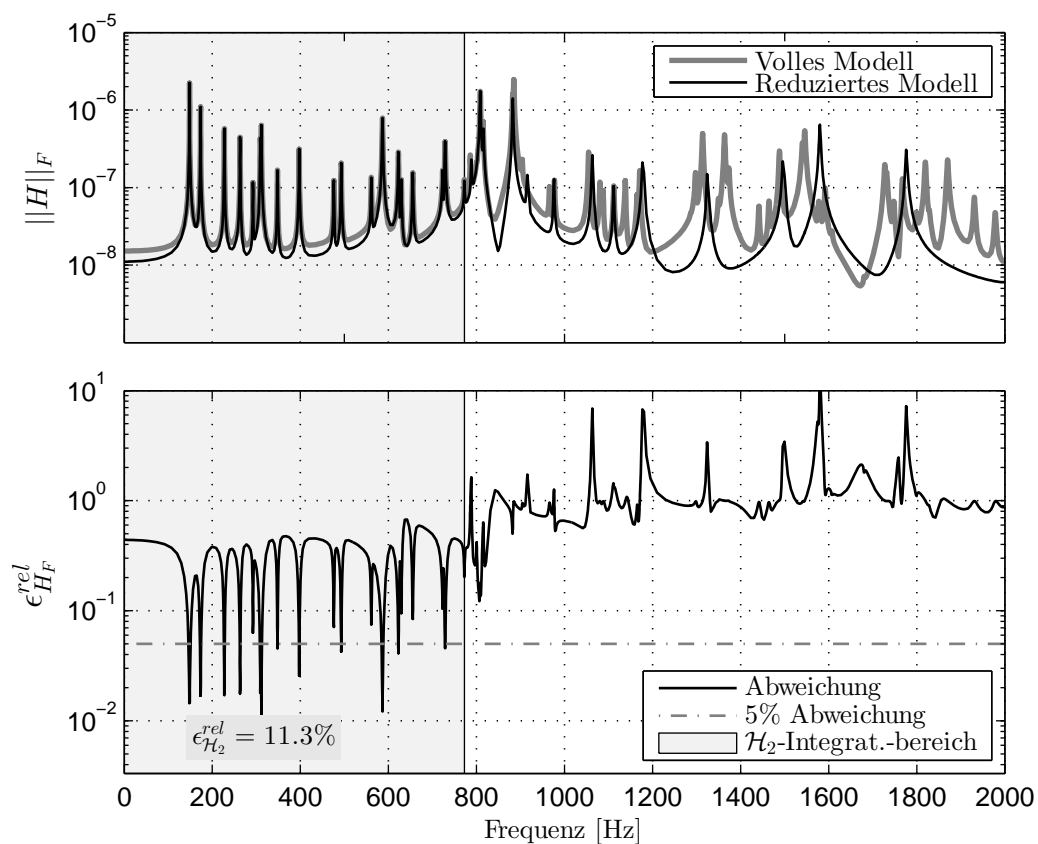


Diagramm A.1: Frequenzgangkriterium bei Übertragungsfunktionsmatrix aus massennormierten Eigenvektoren ohne Abbildung der dominanten Moden

Der \mathcal{H}_2 -Integrationsbereich schließt die dominanten Moden nicht ein. Der relative \mathcal{H}_2 -Fehler ist mit $\epsilon_{\mathcal{H}_2}^{rel} = 11.3\%$ nicht klein.

Diagramm A.2 zeigt ein reduziertes System, welches bis genau zur Eigenmode größter Dominanz ein gutes NTFD-Ergebnis zeigt.

Die dominanten Moden werden jedoch nicht ausreichend genau abgebildet, sodass der \mathcal{H}_2 -Fehler mit $\epsilon_{\mathcal{H}_2}^{rel} = 30.8\%$ groß ist.

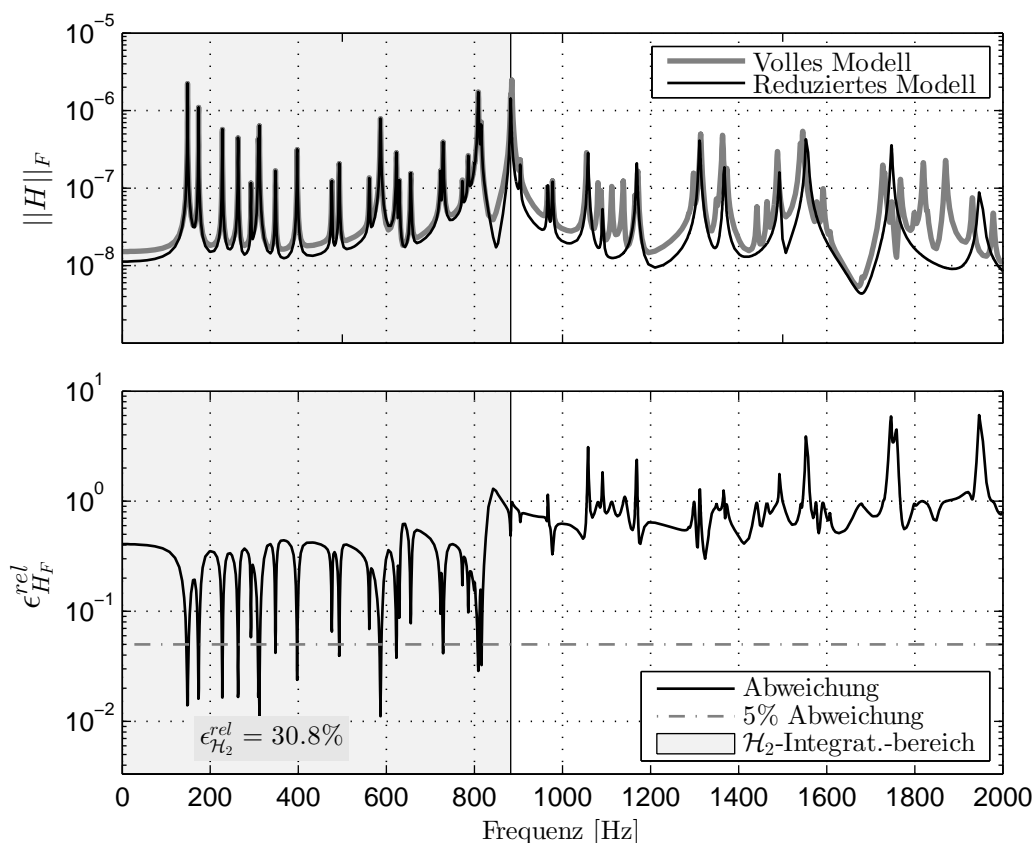


Diagramm A.2: Frequenzgangkriterium bei Übertragungsfunktionsmatrix aus massennormierten Eigenvektoren bei schlechter Abbildung der dominanten Moden

In Diagramm A.3 ist das Übertragungsverhalten der dominanten Moden sehr gut abgebildet, die \mathcal{H}_2 -Integrationsgrenze liegt im Frequenzbereich weit oberhalb der dominanten Moden.

Der \mathcal{H}_2 -Fehler ist mit $\epsilon_{\mathcal{H}_2}^{rel} = 2.43\%$ sehr klein. Die Mitnahme der dominanten Moden ist demnach das wesentliche Kriterium für kleine \mathcal{H}_2 -Fehler.

Soll dieser Effekt nicht auftreten, besteht die Möglichkeit, die Eigenvektoren des vollen und reduzierten Systems nicht massenzuwichten, sondern auf die $\|\dots\|_2$ -Norm des Eingangsanteils der Eigenvektoren nach Gleichung 2.23. Das entsprechende Ergebnis ist in Diagramm A.4 angegeben.

Der \mathcal{H}_2 -Fehler bleibt nun unabhängig von der reduzierten Systemdimension im Bereich um $\epsilon_{\mathcal{H}_2}^{rel} = 2.5\%$. Die Amplituden der Eigenmoden sind bedingt durch die einheitliche Dämpfung mit zunehmender Eigenfrequenz abnehmend, es gibt damit keine dominanten Moden. Der relative Frobenius-Fehler $\epsilon_{H_F}^{rel}$ ist mit zunehmender Frequenz ansteigend, da hier der Einfluss nicht abgebildeter Moden mit deren Nähe zunimmt.

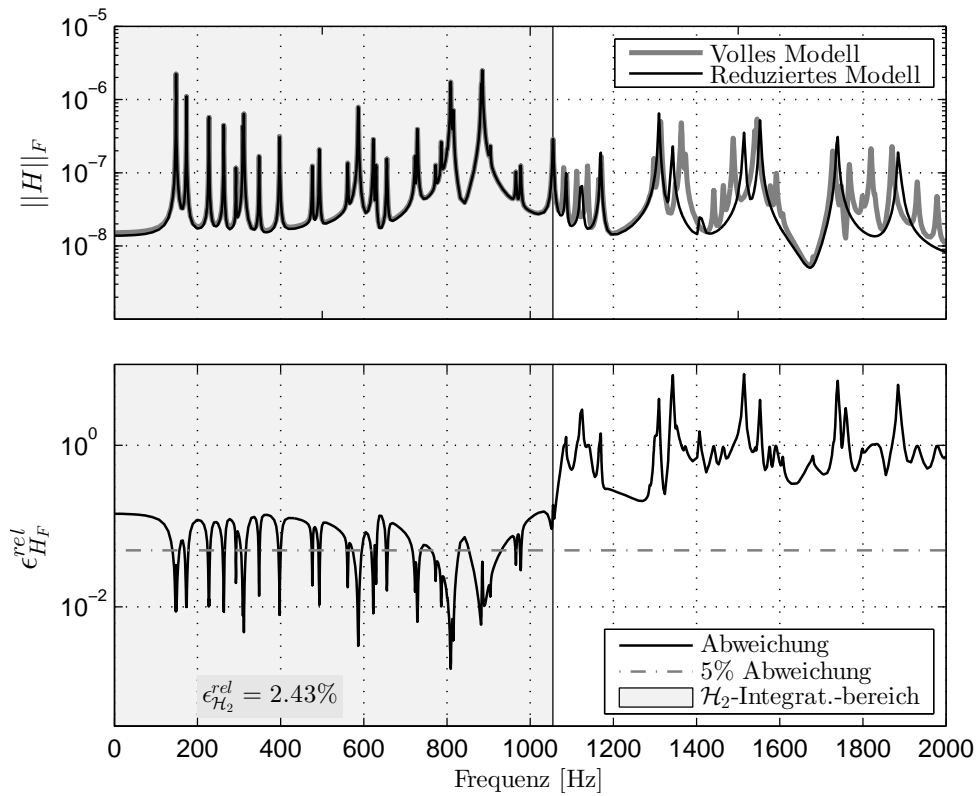


Diagramm A.3: Frequenzgangkriterium bei Übertragungsfunktionsmatrix aus massenormierten Eigenvektoren bei guter Abbildung der dominanten Moden

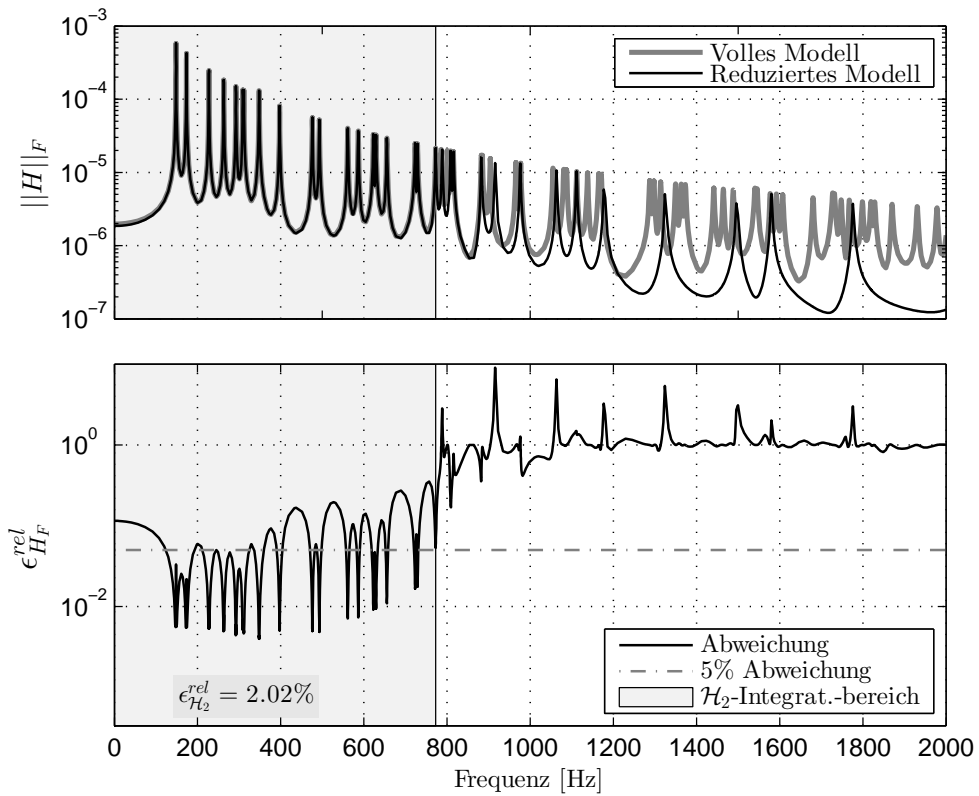


Diagramm A.4: Frequenzgangkriterium bei Übertragungsfunktionsmatrix normierten Eingangsanteilen der Eigenvektoren

A.3. WEITERE DIAGRAMME UND TABELLEN ZUR VALIDIERUNG DER REDUKTIONSMETHODEN AN KLEINEM MODELL

Es werden weitere Diagramme und Tabellen abgebildet, die die Aussagen über die KSM in Abschnitt 5.2.1 und 5.2.2 unterstreichen. Es handelt sich um Tests an den Modellen Rahmen-Tet und Rahmen-Hex.

A.3.1. DIAGRAMME ZUR REORTHOGONALISIERUNG

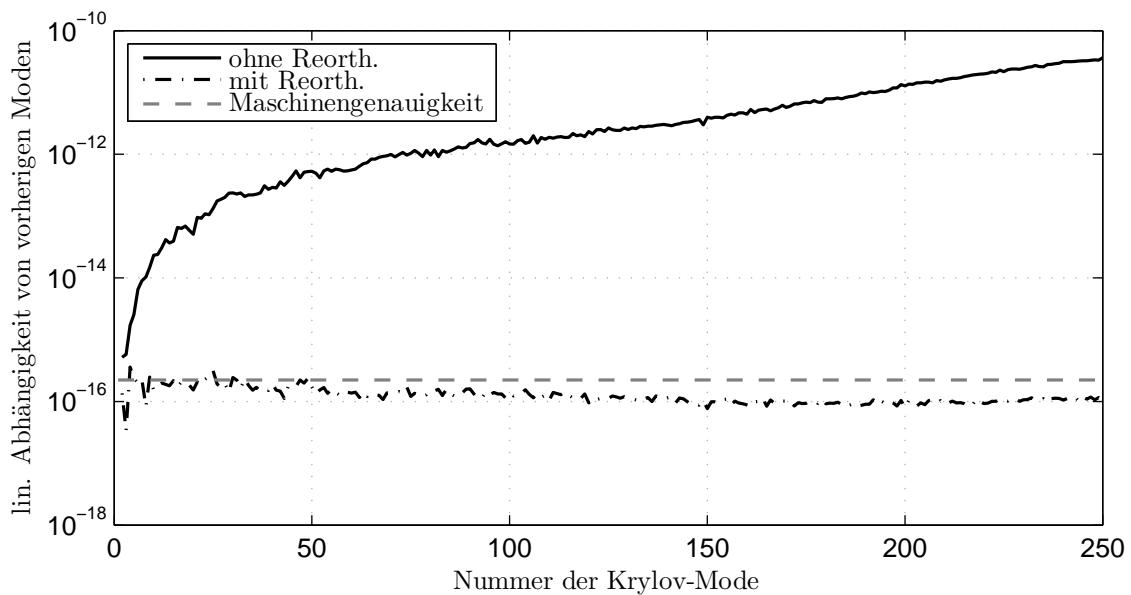


Diagramm A.5: Mittlere lineare Abhängigkeit der Krylov-Moden von den vorherigen Krylov-Moden bei 250 Krylov-Moden (Tetraedervernetzung)

A.3.2. DIAGRAMME ZUM NETZVERGLEICH

Im Hauptteil unter 5.2.1 wird der Vernetzungstyp des Rahmenmodells verglichen. Neben der reduzierten Systemdimension von $n = 30$ dort, ist der Versuch für die reduzierten Dimensionen $n = 20$, $n = 40$ und $n = 50$ durchgeführt worden. Die Ergebnisse werden in diesem Abschnitt präsentiert. Es wird die Darstellungsform des eigenfrequenzbasierten NRFD-Kriteriums und des übertragungsfunktionsbasierten Frobenius-Fehlers und \mathcal{H}_2 -Fehlers gewählt.

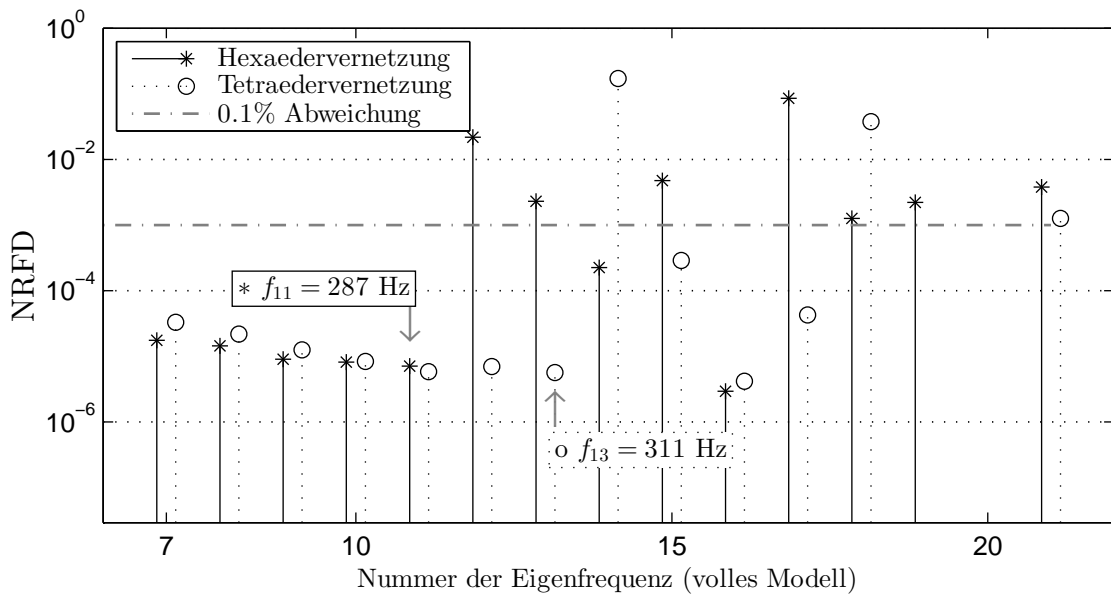


Diagramm A.6: NREFD der Reduktion des hexaeder- und tetraedervernetzten Modells mit 20 Krylov-Moden

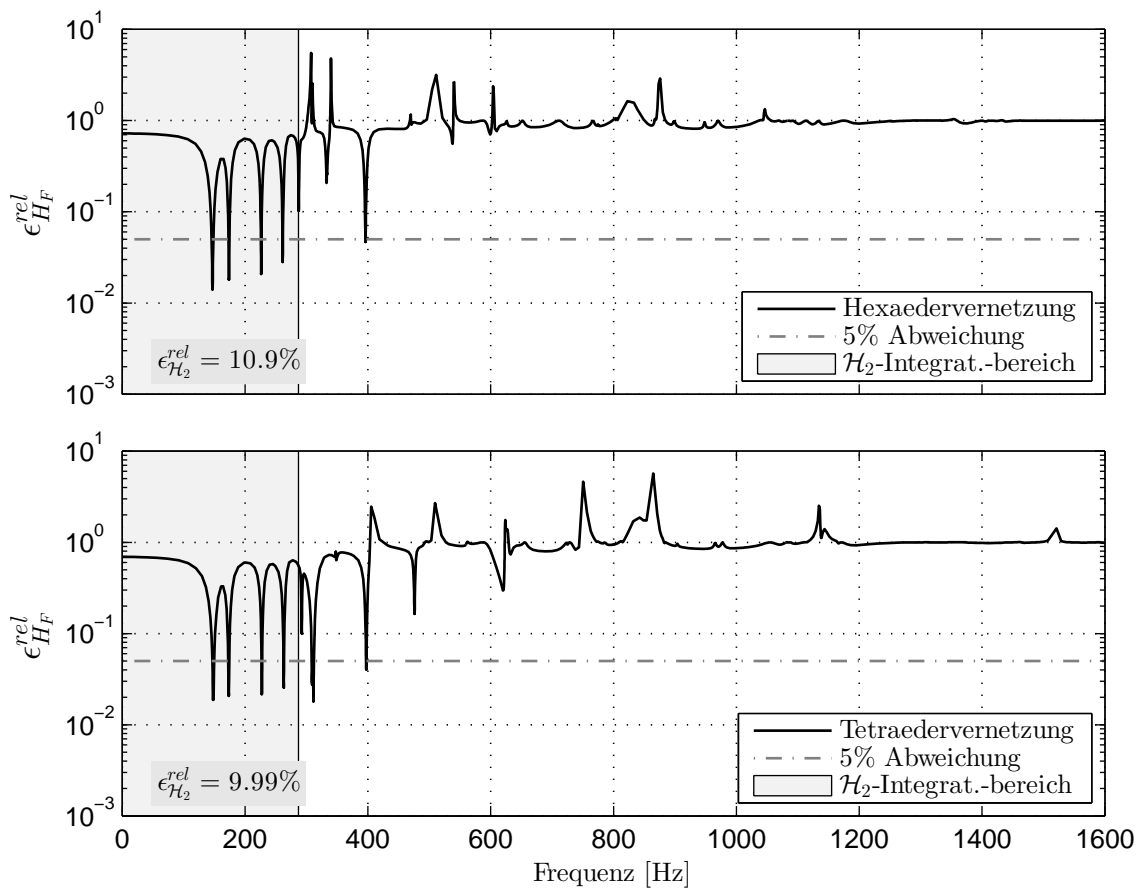


Diagramm A.7: Abweichung des mithilfe von 20 Krylov-Moden reduzierten, hexaeder- und tetraedervernetzten Rahmenmodells (Arnoldi-Verfahren erster Ordnung)

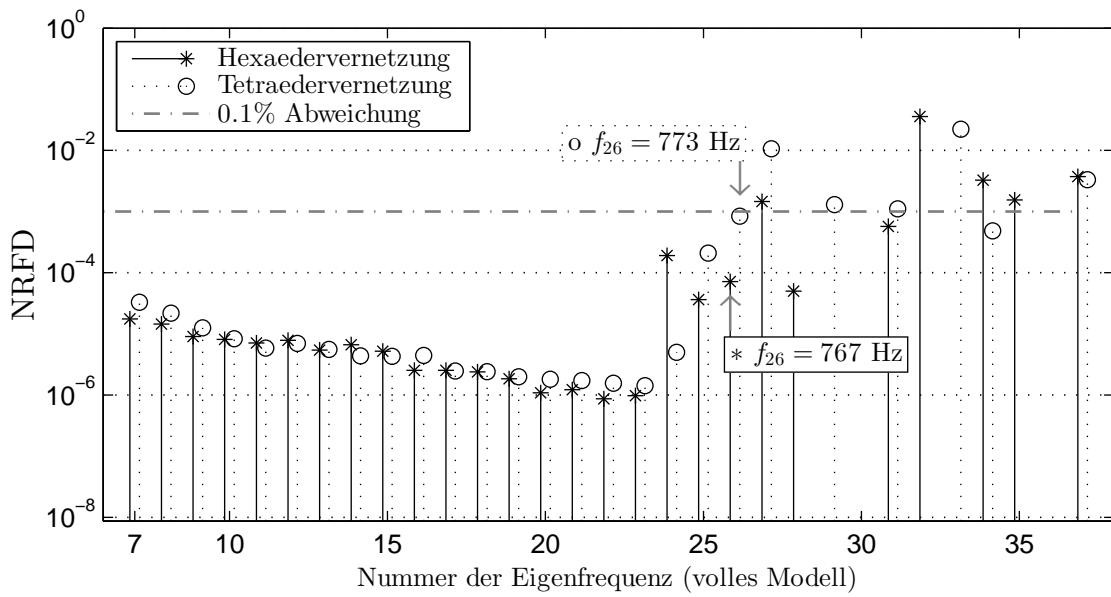


Diagramm A.8: NRRFD der Reduktion des hexaedervernetzten und tetraedervernetzten Modells mit 40 Krylov-Moden

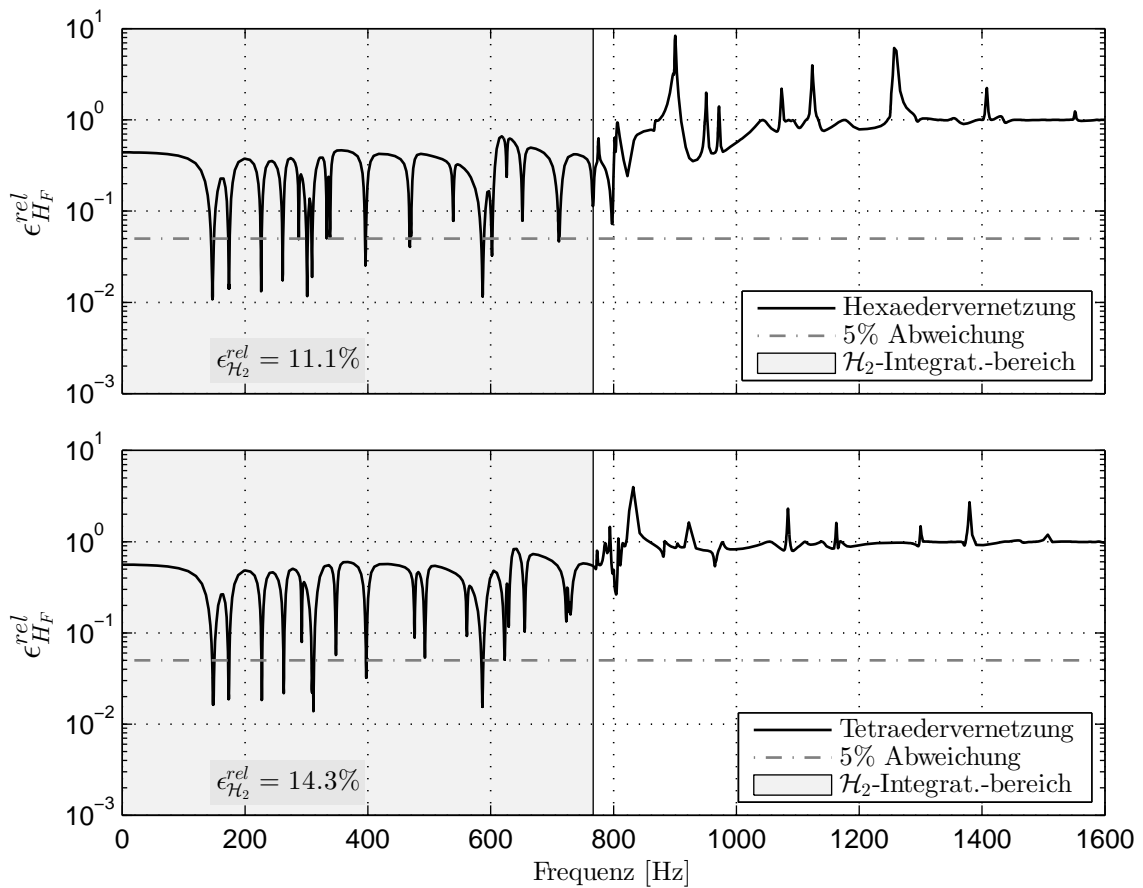


Diagramm A.9: Abweichung des mithilfe von 40 Krylov-Moden reduzierten, hexaedervernetzten und tetraedervernetzten Rahmenmodells (Arnoldi-Verfahren erster Ordnung)

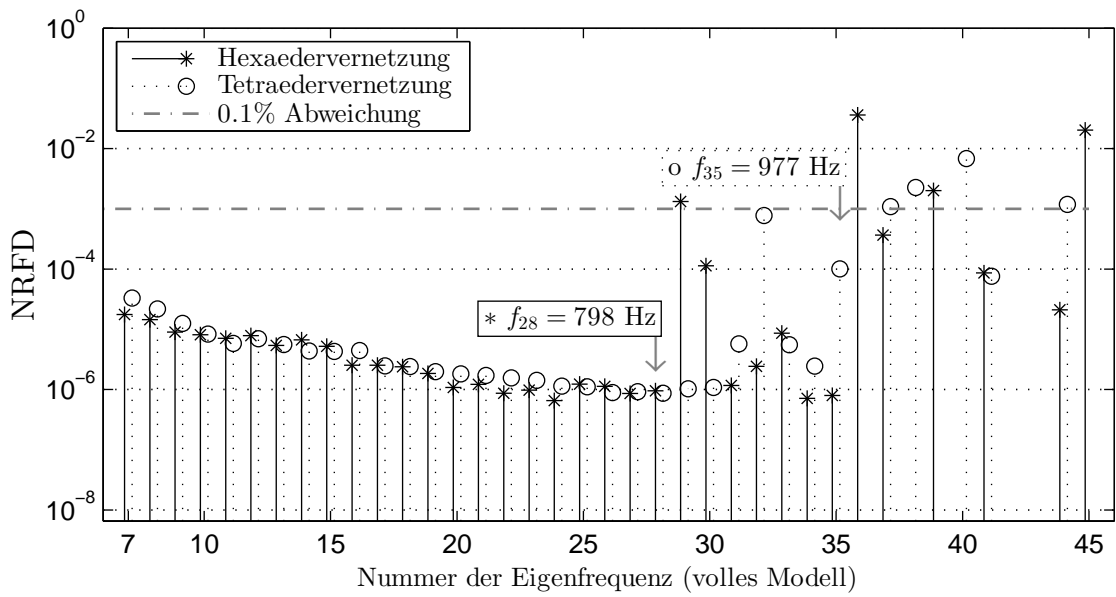


Diagramm A.10: NRFD der Reduktion des hexaeder- und tetraedervernetzten Modells mit 50 Krylov-Moden

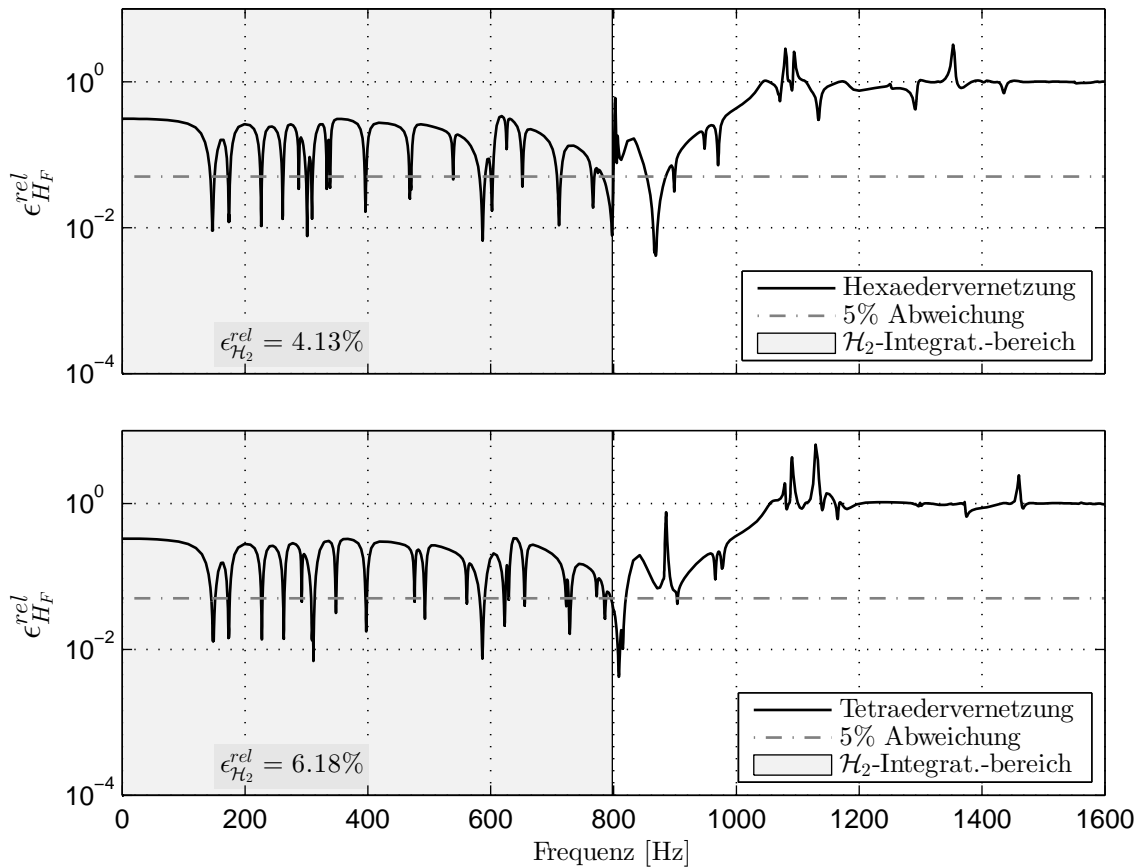


Diagramm A.11: Abweichung des mithilfe von 50 Krylov-Moden reduzierten, hexaeder- und tetraedervernetzten Rahmenmodells (Arnoldi-Verfahren erster Ordnung)

A.3.3. DIAGRAMME UND TABELLEN ZUR WAHL DES OPTIMALEN INTERPOLATIVEN STARTVEKTORS

Es wird unter 5.2.2 die optimale Wahl eines interpolativen Startvektors untersucht. Neben der Untersuchung am Modell **Rahmen-Tet** mit reduzierter Dimension $n = 25$ ist der Versuch an Modell **Rahmen-Hex** mit reduzierter Systemdimension $n = 25$ und an Modell **Rahmen-Tet** mit reduzierter Systemdimension $n = 40$ durchgeführt worden. Die Ergebnisse werden in diesem Abschnitt dargestellt.

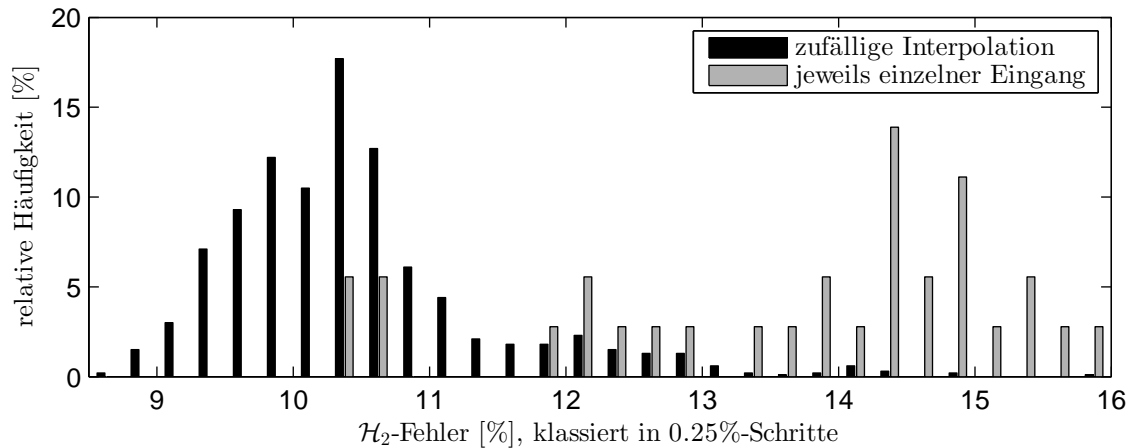


Diagramm A.12: klassierte \mathcal{H}_2 -Abweichung bei 1000 zufällig interpolierten Startvektoren im Vergleich zu 36 Startvektoren aus einzelnen Eingängen (**Rahmen-Hex**), reduziert auf Dimension 25

Tabelle A.1.: Parameterwahl und Ergebnisse des Testes mit zufälligen Startvektoren, einzelnen Eingängen und gleichgewichteter Interpolation an **Rahmen-Hex**, auf Dimension 25 reduziert

Parameterwahl	
\mathcal{H}_2 -Integrationsbereich	[0 . . . 530Hz]
Anzahl Frequenzstützstellen	1
Frequenzstützstelle	0Hz
Reduzierte Systemdimension	25
LEHRsche Dämpfung (für Korrelation)	$D = 0.001$
Versuche mit zufälliger Interpolation	1000
Versuche mit einzeltem Startvektor	36

Ergebnisse	
Größte Eigenfrequenz $NRFD < 0.1\%$ (Gleichwichtung)	$f_{19} = 539\text{Hz}$
\mathcal{H}_2 -Fehlerbereich mit zufälliger Interpolation	[8.72% . . . 21.96%]
Mittelwert \mathcal{H}_2 -Abweichung mit zufälliger Interpolation	10.5%
Mittelwert \mathcal{H}_2 -Abweichung mit einzelnen Startvektoren	14.3%
Gleichgewichtete Interpolation	<u>10.6%</u>
Altes MORPACK-KSM (Spalte von K als Startvektor)	11.9%

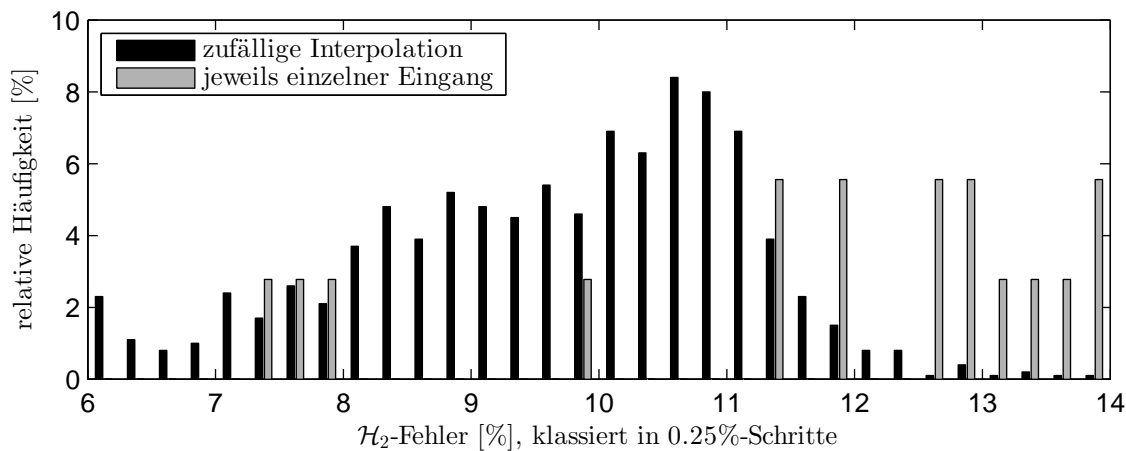


Diagramm A.13: klassierte \mathcal{H}_2 -Abweichung bei 1000 zufällig interpolierten Startvektoren im Vergleich zu 36 Startvektoren aus einzelnen Eingängen (**Rahmen-Tet**), reduziert auf Dimension 40

Tabelle A.2.: Parameterwahl und Ergebnisse des Testes mit zufälligen Startvektoren, einzelnen Eingängen und gleichgewichteter Interpolation an **Rahmen-Tet**, auf Dimension 40 reduziert

Parameterwahl	
\mathcal{H}_2 -Integrationsbereich	[0 ... 770Hz]
Anzahl Frequenzstützstellen	1
Frequenzstützstelle	0Hz
Reduzierte Systemdimension	40
LEHRsche Dämpfung (für Korrelation)	$D = 0.001$
Versuche mit zufälliger Interpolation	1000
Versuche mit einzelner Startvektor	36

Ergebnisse	
Größte Eigenfrequenz $NRFD < 0.1\%$ (Gleichwichtung)	$f_{26} = 773\text{Hz}$
\mathcal{H}_2 -Fehlerbereich mit zufälliger Interpolation	[4.49% ... 15.0%]
Mittelwert \mathcal{H}_2 -Abweichung mit zufälliger Interpolation	9.58%
Mittelwert \mathcal{H}_2 -Abweichung mit einzelnen Startvektoren	13.8%
Gleichgewichtete Interpolation	<u>11.2%</u>
Altes MORPACK-KSM (Spalte von K als Startvektor)	14.4%

A.3.4. TABELLEN ZUM VERGLEICH INTERPOLATIVER UND BLOCK-SOAR

In Abschnitt 5.2.2 wird das interpolative SOAR- mit dem Block-SOAR-Verfahren verglichen. Die vollständigen Versuchsergebnisse sind in den Tabellen dieses Abschnitts dargestellt. Tabelle A.4 zeigt Referenzergebnisse für das interpolative Verfahren. In Tabelle A.5 sind die Ergebnisse für verschiedene Herangehensweisen beim Block-Verfahren angegeben.

Tabelle A.3.: Parameterwahl und Ergebnisse des Vergleiches des rationalen, interpolativen mit dem Block-SOAR

Parameterwahl	
Anzahl Frequenzstützstellen	1
Frequenzstützstelle	0Hz
Lehrsche Dämpfung (für Korrelation)	$D = 0.001$
Inpolationsart für interpolatives Verfahren	Gleichwichtung

Tabelle A.4.: Reduktionsergebnisse mit interpolativem SOAR-Verfahren und Gleichwichtung

reduzierte Dimension	Ordnung der Reihenentwicklung	Letzte Eig.-Freq. mit $NRFD < 0.1\%$	\mathcal{H}_2 -Integrations-Bereich	\mathcal{H}_2 -Fehler
16	32	$f_{14} = 348\text{Hz}$	[0 ... 311Hz] [0 ... 397Hz] [0 ... 476Hz]	13.7% 22.6% 23.7%
17	34	$f_{14} = 348\text{Hz}$	[0 ... 311Hz] [0 ... 397Hz] [0 ... 476Hz]	9.14% 12.3% 14.4%
18	36	$f_{14} = 348\text{Hz}$	[0 ... 311Hz] [0 ... 397Hz] [0 ... 476Hz]	8.59% 11.7% 13.5%
19	38	$f_{14} = 348\text{Hz}$	[0 ... 311Hz] [0 ... 397Hz] [0 ... 476Hz]	8.41% 11.5% 17.1%
20	40	$f_{14} = 348\text{Hz}$	[0 ... 311Hz] [0 ... 397Hz] [0 ... 476Hz]	8.39% 12.0% 13.8%
21	42	$f_{14} = 348\text{Hz}$	[0 ... 311Hz] [0 ... 397Hz] [0 ... 476Hz]	8.33% 11.1% 12.6%
22	44	$f_{17} = 476\text{Hz}$	[0 ... 311Hz] [0 ... 397Hz] [0 ... 476Hz]	8.13% 9.04% 10.3%
23	46	$f_{19} = 561\text{Hz}$	[0 ... 311Hz] [0 ... 397Hz] [0 ... 476Hz]	7.77% 8.60% 9.59%
24	48	$f_{19} = 561\text{Hz}$	[0 ... 311Hz] [0 ... 397Hz] [0 ... 476Hz]	7.68% 8.48% 9.44%

Tabelle A.5.: Reduktionsergebnisse für verschiedene Eingänge im Block-SOAR-Verfahren

reduz. Dimen- sion	Eingänge	Ordnung der Reihen- entwicklung	Letzte Eig- Freq. mit <i>NRFD</i> < 0.1%	\mathcal{H}_2 -Integr. Bereich	\mathcal{H}_2 - Fehler
Alle Eingänge					
72	1 bis 36	3	$f_{16} = 397\text{Hz}$	[0...397Hz]	9.33%
Sechs Eingänge mit bestem Einzelergebnis					
24	30/31/25/36/12/24	8	$f_8 = 173\text{Hz}$	[0...311Hz]	45.1%
Sechs Eingänge, jeweils bester Eingang einer Koordinatenrichtung					
24	30/31/26/22/5/15	8	$f_{13} = 311\text{Hz}$	[0...311Hz]	13.8%
Alle Eingänge/Freiheitsgrade eines Anbindungsknoten (Masterknoten)					
24	1 bis 6	8	$f_{13} = 311\text{Hz}$	[0...311Hz]	13.9%
24	7 bis 12	8	$f_{13} = 311\text{Hz}$	[0...311Hz]	11.9%
24	13 bis 18	8	$f_{13} = 311\text{Hz}$	[0...311Hz]	11.0%
24	19 bis 24	8	$f_{13} = 311\text{Hz}$	[0...311Hz]	11.7%
24	25 bis 30	8	$f_{13} = 311\text{Hz}$	[0...311Hz]	13.1%
24	31 bis 36	8	$f_{12} = 309\text{Hz}$	[0...311Hz]	13.7%
Translat. oder rotat. Eingänge/Freiheitsgrade eines Anbindungsknoten					
24	1 bis 3	16	$f_{16} = 397\text{Hz}$	[0...397Hz]	11.8%
24	4 bis 6	16	$f_{17} = 476\text{Hz}$	[0...397Hz]	11.8%
24	7 bis 9	16	$f_{16} = 397\text{Hz}$	[0...397Hz]	11.6%
24	10 bis 12	16	$f_{17} = 476\text{Hz}$	[0...397Hz]	11.7%
24 (*)	13 bis 15	16	$f_{14} = 348\text{Hz}$	[0...397Hz]	14.9%
24 (*)	16 bis 18	16	$f_{13} = 311\text{Hz}$	[0...397Hz]	13.2%
24	19 bis 21	16	$f_{16} = 397\text{Hz}$	[0...397Hz]	12.4%
24	22 bis 24	16	$f_{17} = 476\text{Hz}$	[0...397Hz]	11.7%
24 (*)	25 bis 27	16	$f_{18} = 493\text{Hz}$	[0...397Hz]	9.82%
24 (*)	28 bis 30	16	$f_{16} = 397\text{Hz}$	[0...397Hz]	10.5%
24	31 bis 33	16	$f_{15} = 348\text{Hz}$	[0...397Hz]	11.9%
24	34 bis 36	16	$f_{14} = 348\text{Hz}$	[0...397Hz]	11.6%
Jeweils zwei der drei Eingänge mit bestem/schlechtestem gemeins. Ergebnis (*)					
24	25 26	24	$f_{19} = 561\text{Hz}$	[0...476Hz]	10.3%
24	25 27	24	$f_{17} = 476\text{Hz}$	[0...476Hz]	10.3%
24	26 27	24	$f_{16} = 397\text{Hz}$	[0...476Hz]	14.7%
24	28 29	24	$f_{18} = 493\text{Hz}$	[0...476Hz]	14.7%
24	28 30	24	$f_{17} = 476\text{Hz}$	[0...476Hz]	9.70%
24	29 30	24	$f_{16} = 397\text{Hz}$	[0...476Hz]	11.0%
24	16 17	24	$f_{19} = 561\text{Hz}$	[0...476Hz]	12.6%
24	16 18	24	$f_{16} = 397\text{Hz}$	[0...476Hz]	13.1%
24	17 18	24	$f_{17} = 476\text{Hz}$	[0...476Hz]	13.0%
24	13 14	24	$f_{18} = 493\text{Hz}$	[0...476Hz]	12.0%
24	13 15	24	$f_{18} = 493\text{Hz}$	[0...476Hz]	11.9%
24	14 15	24	$f_{16} = 397\text{Hz}$	[0...476Hz]	12.9%

A.3.5. DIAGRAMME ZUR WAHL EINZELNER ENTWICKLUNGSSTELLEN

In Abschnitt 5.2.1 wird der Einfluss der Wahl einzelner Entwicklungsstellen auf die Übereinstimmung der Eigenfrequenzen und den \mathcal{H}_2 -Fehler im resultierenden Integrationsbereich untersucht. Neben den Tabellen des genannten Abschnitts und Video-Plots auf der Daten-CD sind in diesem Abschnitt einzelne Darstellungen des Frobenius- und \mathcal{H}_2 -Fehlers angegeben. Es spielen dabei die dominanten Moden der Übertragungsfunktion im Bereich $f = [800 \dots 1000]$ Hz eine wesentliche Rolle.

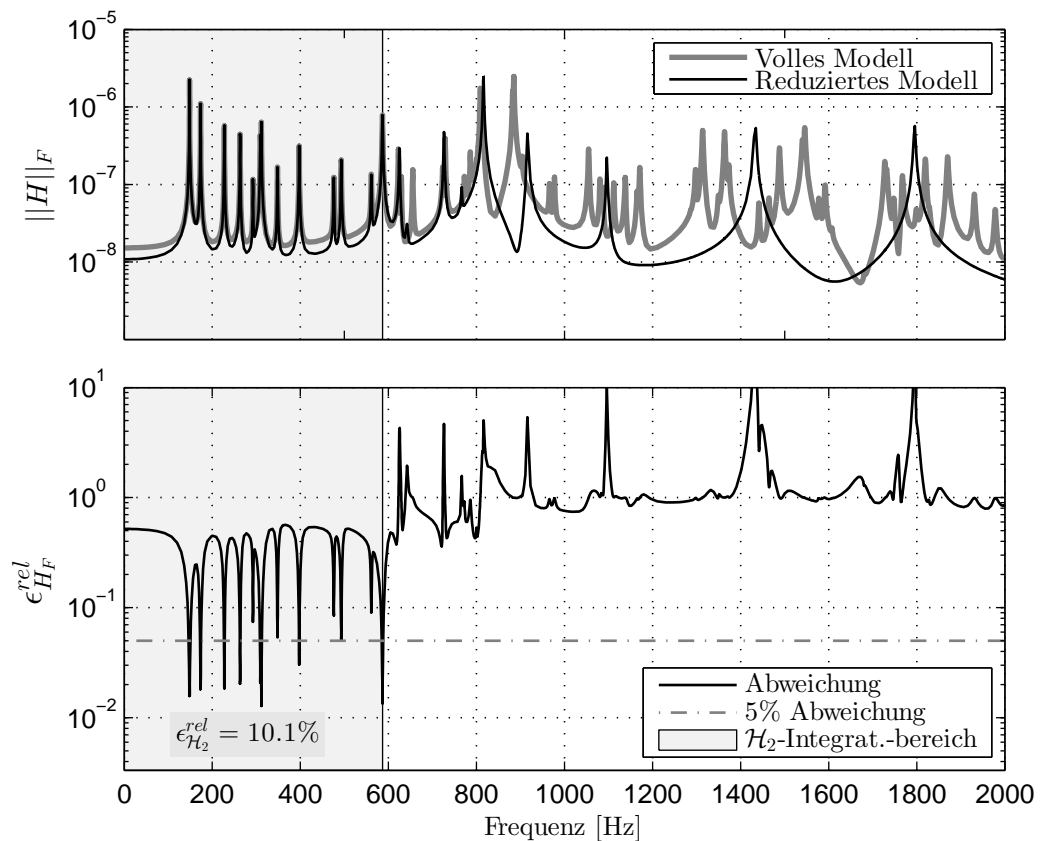


Diagramm A.14: Frobenius-Norm der Frequenzgänge des vollen und reduzierten Systems und relative Abweichung für eine Entwicklungsstelle bei $f_{ES} = 300$ Hz (Rahmen-Tet, reduzierte Dimension $n = 25$); kleiner \mathcal{H}_2 -Fehler aufgrund Entfernung zu dominanten Moden

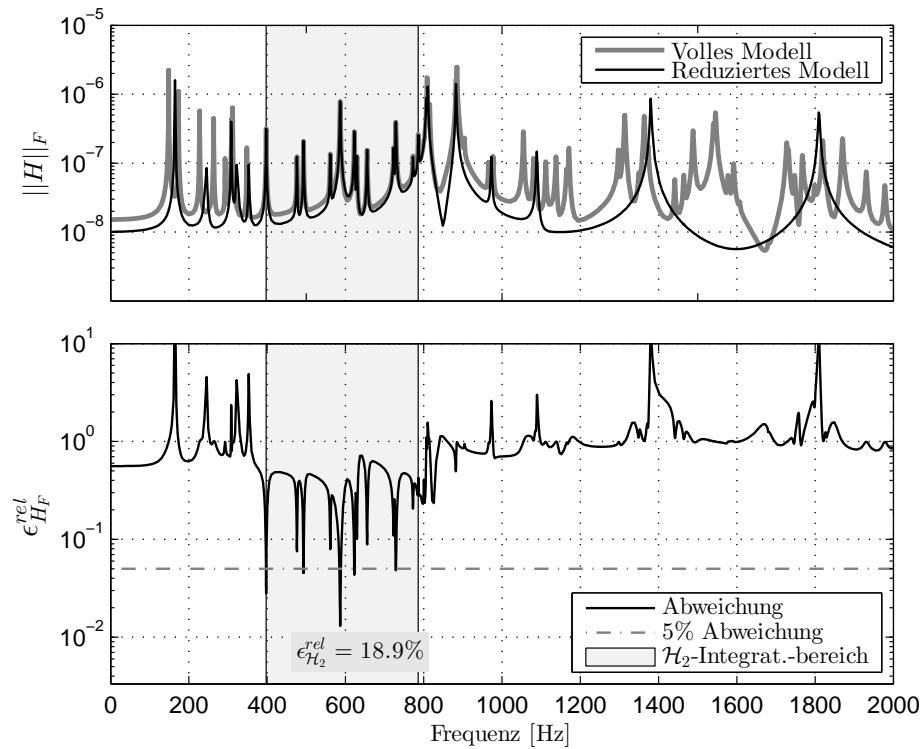


Diagramm A.15: Frobenius-Norm der Frequenzgänge des vollen und reduzierten Systems und relative Abweichung für eine Entwicklungsstelle bei $f_{ES} = 610\text{Hz}$ (Rahmen-Tet, reduzierte Dimension $n = 25$); großer \mathcal{H}_2 -Fehler aufgrund Nähe zu dominanten Moden

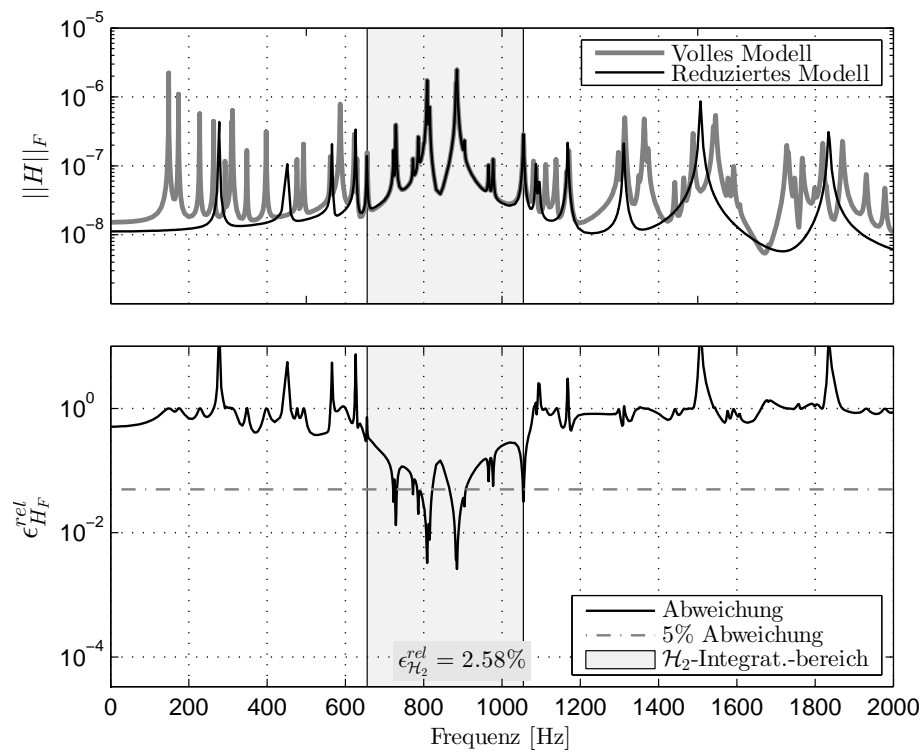


Diagramm A.16: Frobenius-Norm der Frequenzgänge des vollen und reduzierten Systems und relative Abweichung für eine Entwicklungsstelle bei $f_{ES} = 870\text{Hz}$ (Rahmen-Tet, reduzierte Dimension $n = 25$); sehr kleiner \mathcal{H}_2 -Fehler aufgrund Mitnahme der dominanten Moden

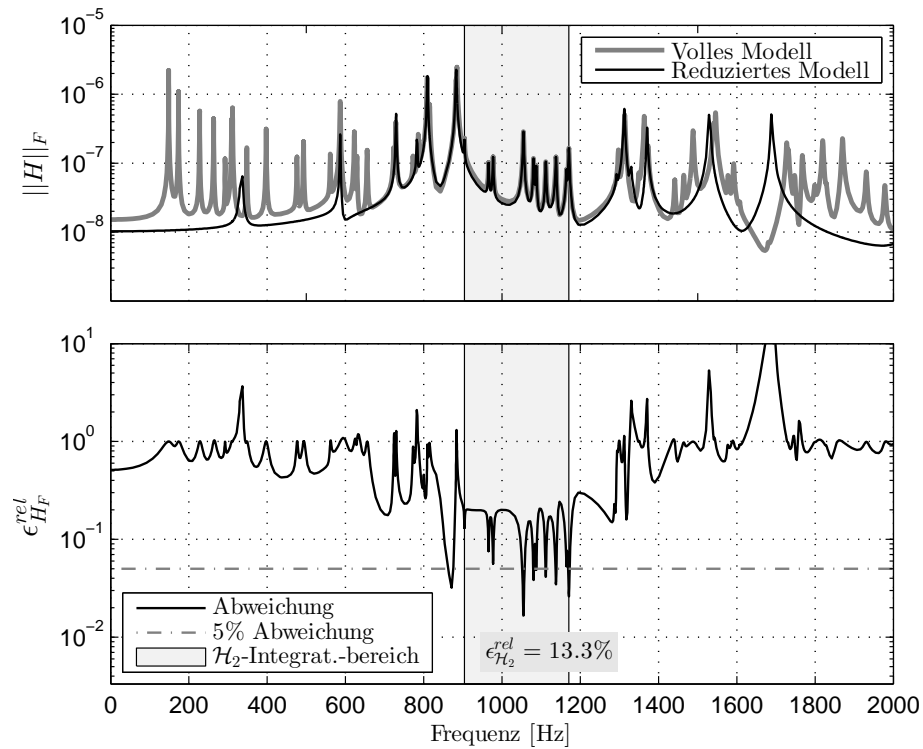


Diagramm A.17: Frobenius-Norm der Frequenzgänge des vollen und reduzierten Systems und relative Abweichung für eine Entwicklungsstelle bei $f_{ES} = 1065\text{Hz}$ (Rahmen-Tet, reduzierte Dimension $n = 25$); großer \mathcal{H}_2 -Fehler aufgrund Nähe zu dominanten Moden

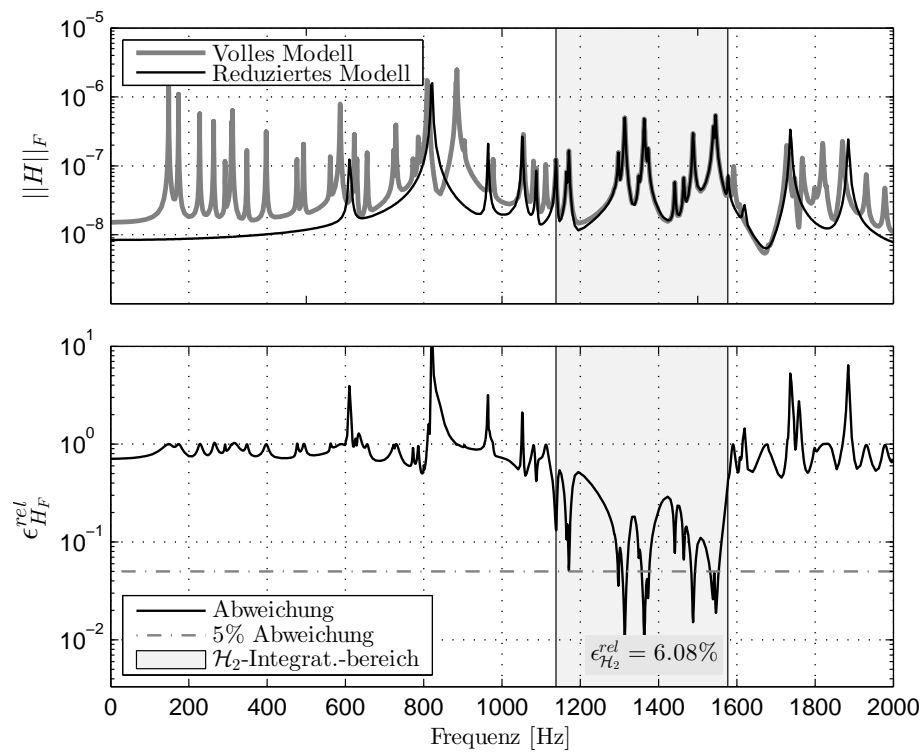


Diagramm A.18: Frobenius-Norm der Frequenzgänge des vollen und reduzierten Systems und relative Abweichung für eine Entwicklungsstelle bei $f_{ES} = 1340\text{Hz}$ (Rahmen-Tet, reduzierte Dimension $n = 25$); kleiner \mathcal{H}_2 -Fehler aufgrund Entfernung zu dominanten Moden

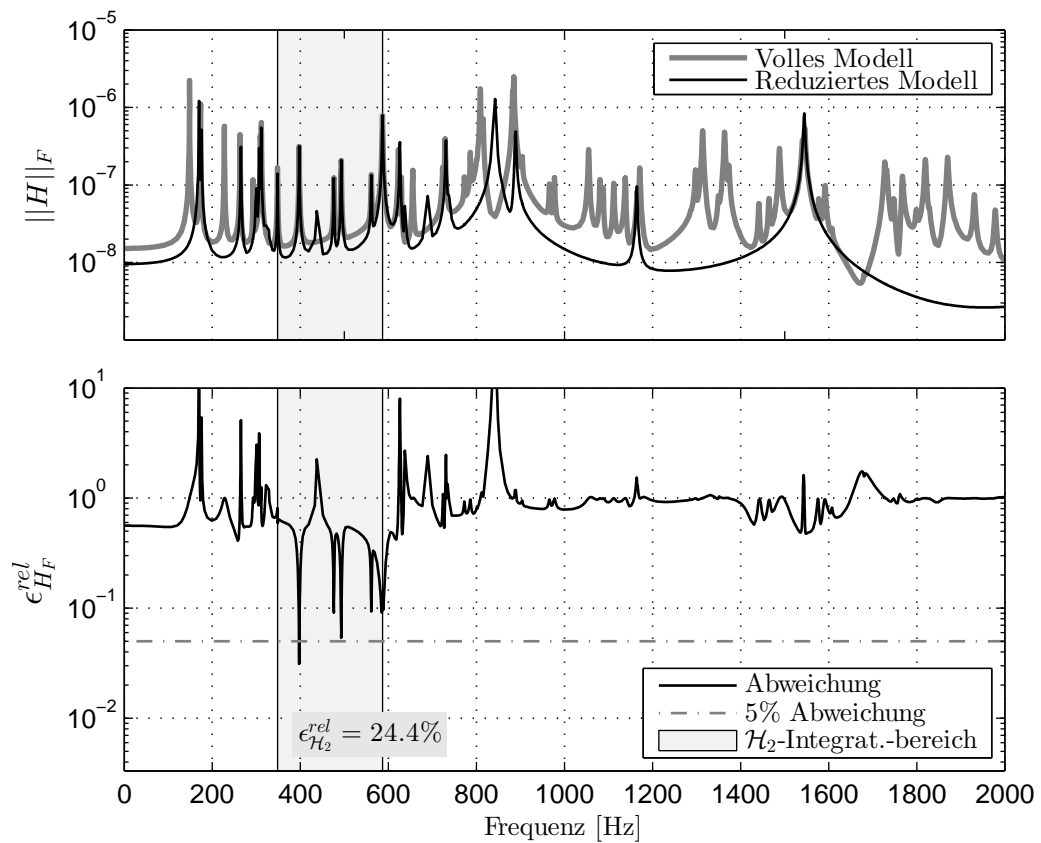


Diagramm A.19: Frobenius-Norm der Frequenzgänge des vollen und reduzierten Systems und relative Abweichung (Rahmen-Tet, reduzierte Dimension $n = 25$); großer \mathcal{H}_2 -Fehler und schmaler \mathcal{H}_2 -Integrationsbereich durch einzelne Entwicklungsstelle auf Eigenfrequenz

A.3.6. DIAGRAMME ZUR REDUKTIONSGÜTE IN ABHÄNGIGKEIT DER REDUZIERTEN SYSTEMDIMENSION

In Abschnitt 5.2.2 wird für einzelne Entwicklungsstellen die Abhängigkeit der Reduktionsgüte von der Ordnung der Reihenentwicklung untersucht. In diesem Abschnitt wird der Einfluss der Wichtung beim Berechnen der Übertragungsfunktionsmatrix auf den \mathcal{H}_2 -Fehler und die Entwicklung um eine weitere Entwicklungsstelle $f_{ES} > 0$ dargestellt.

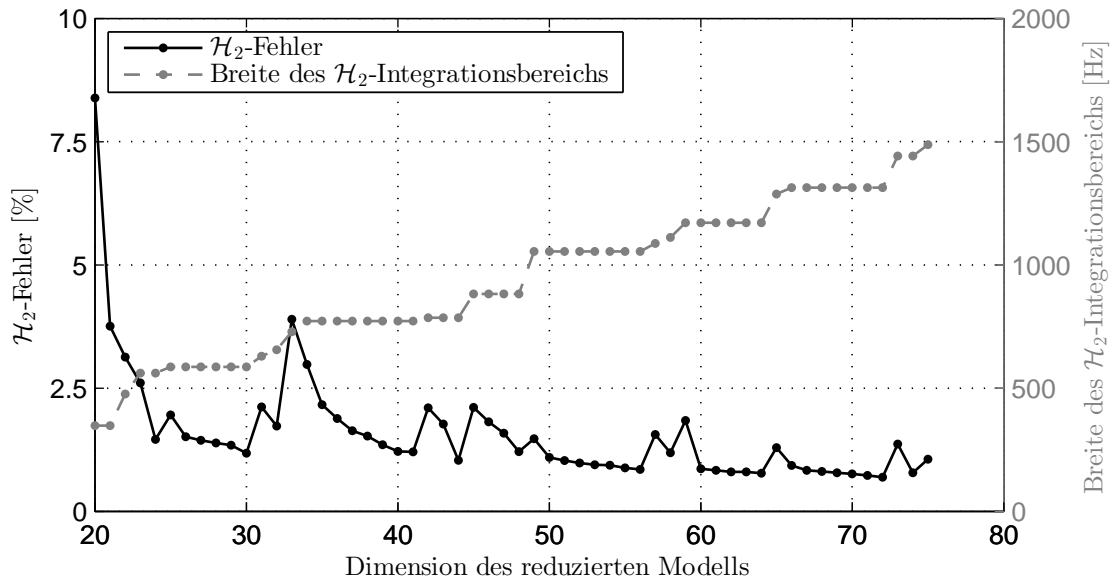


Diagramm A.20: \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für variierende Dimension des reduzierten Systems n (Rahmen-Tet) ohne Berücksichtigung der modalen Masse

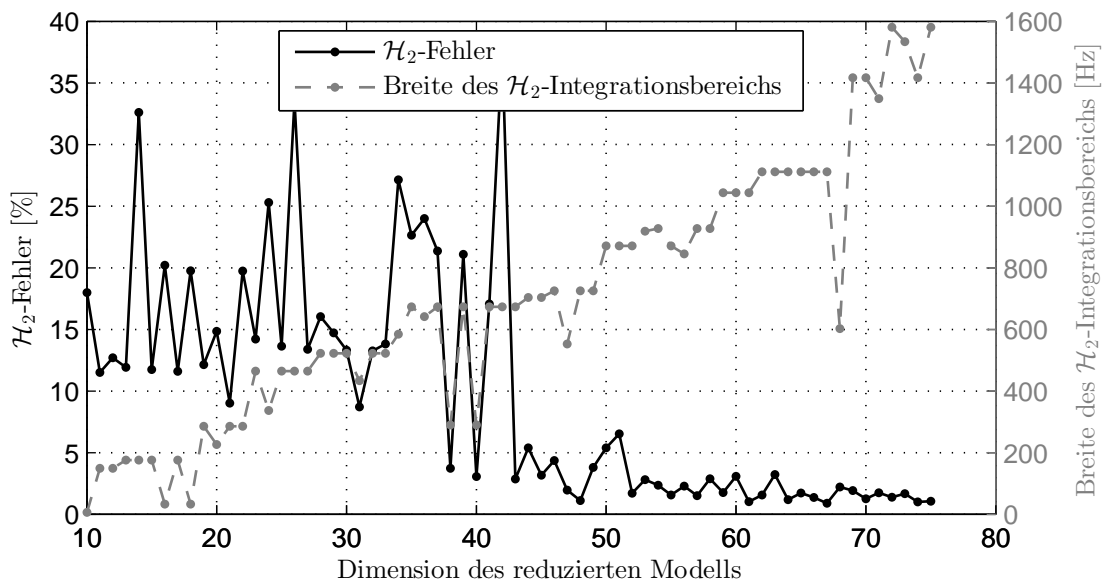


Diagramm A.21: \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für aufsteigende, reduzierte Systemdimension bei Entwicklungsstelle $f_{ES} = 1230\text{Hz}$ (Rahmen-Tet)

A.3.7. DIAGRAMME ZUR KOMBINATION VON ENTWICKLUNGSSTELLEN

In Abschnitt 5.2.2 wird anhand des Modells Rahmen-Tet minimale reduzierte Systemdimension für verschiedene Anzahl äquidistant verteilter Entwicklungsstellen zur Abbildung des Frequenzbereiches $f = [0 \dots 2000]$ Hz untersucht. Dieser Versuch wird in diesem Abschnitt für das Modell Rahmen-Hex wiederholt.

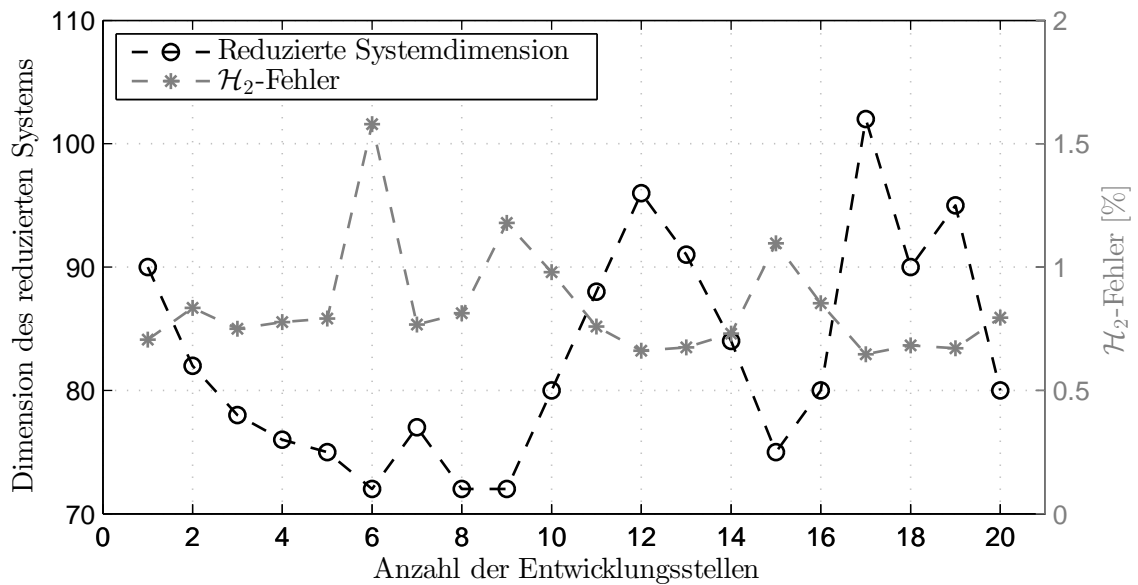


Diagramm A.22: Notwendige reduzierte Systemdimension zur Abbildung des Frequenzbereiches $f = [0 \dots 2000]$ Hz für verschiedene Anzahl linear verteilter Entwicklungsstellen und zugehörige \mathcal{H}_2 -Fehler (Rahmen-Hex)

A.4. WEITERE DIAGRAMME ZUR VALIDIERUNG DER GEWONNENEN ERKENNTNISSE AN MODELEN GROSSER DIMENSION

In Abschnitt 5.3.1 werden die Ergebnisse, die an den Rahmen-Modellen erzielt worden sind, auf die Getriebegehäuse- und Treibsatzwellen-Modelle angewendet. In diesem Abschnitt werden weitere Diagramme angegeben, die die gemachten Aussagen unterstreichen. Für die Getriebegehäuse- Modelle wird die Abhängigkeit der Reduktionsgüte von der reduzierten Dimension und der Einfluss der Reihenentwicklungsminderung auf das Reduktionsergebnis untersucht. Es ist das eigenfrequenzbezogene NTFD-Kriterium für die beiden Treibsatzwellen- Modelle mit Kombination von vier Entwicklungsstellen angegeben.

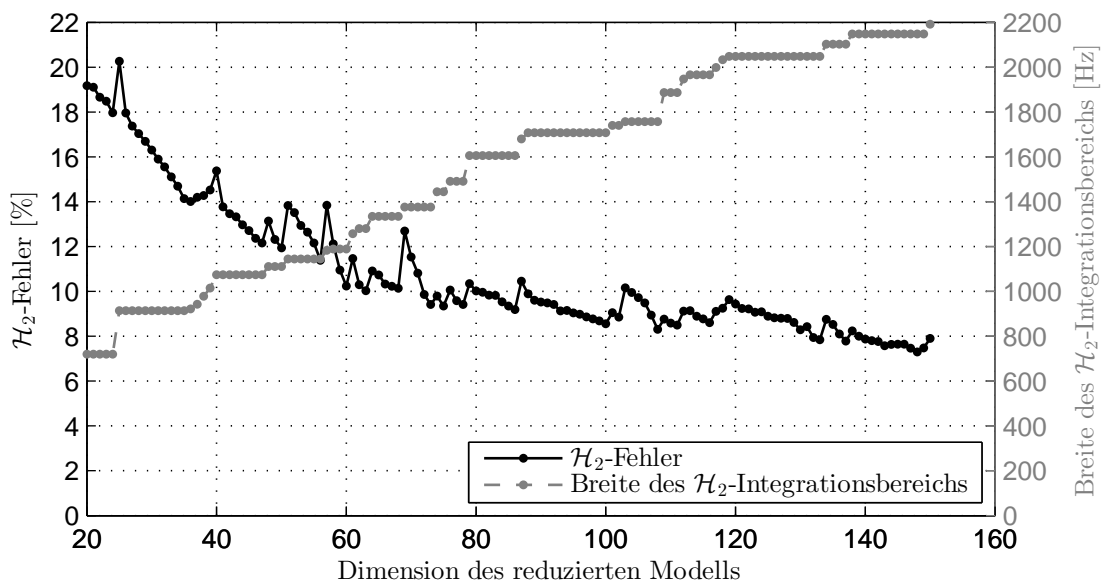


Diagramm A.23: \mathcal{H}_2 -Fehler und Breite des \mathcal{H}_2 -Integrationsbereiches für variierende Dimension des reduzierten Systems n für das Modell **Getriebe-Fein** und Entwicklungsstelle $f_{ES} = 0\text{Hz}$. Frequenzbereich $f = [0 \dots 2000]\text{Hz}$ ab reduzierter Dimension $N = 118$ mit $\epsilon_{\mathcal{H}_2}^{rel} = 9,25\%$ abgebildet.

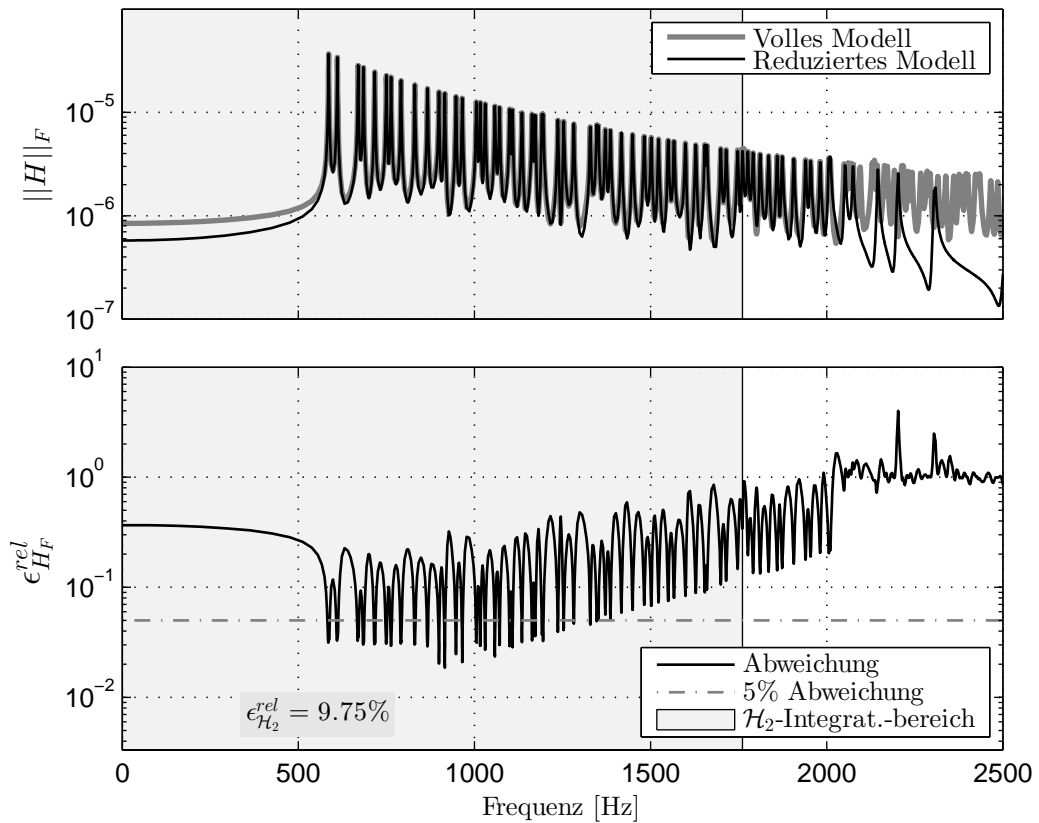


Diagramm A.24: Frobenius-Norm der Übertragungsfunktionsmatrizen und relative Abweichung sowie \mathcal{H}_2 -Fehler für reduzierte Dimension $n = 72$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 2000]$ Hz (Getriebe-Grob)

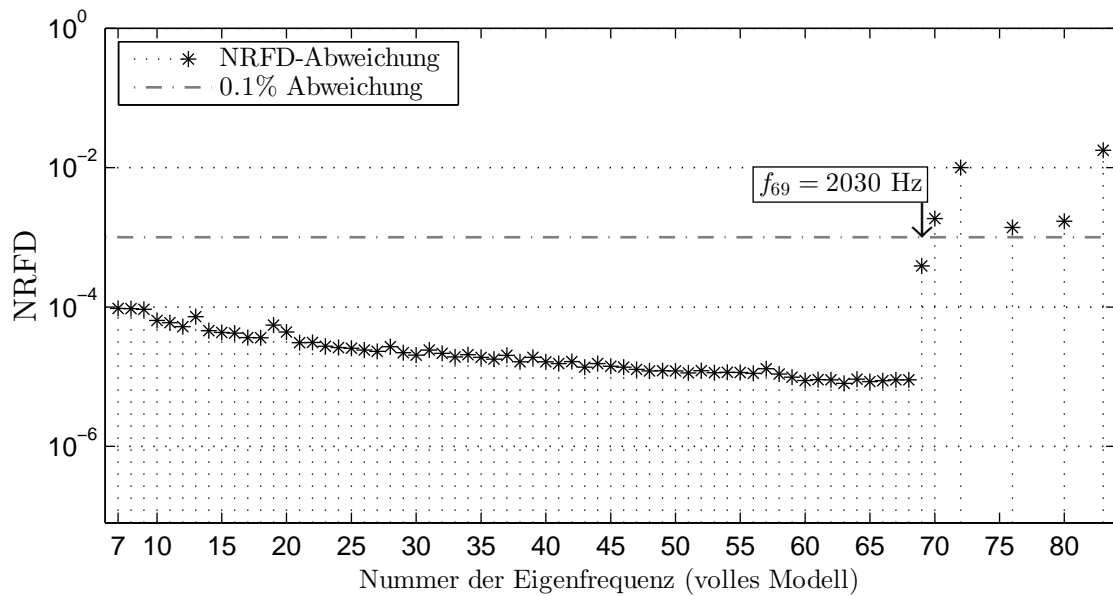


Diagramm A.25: NRFD der Eigenfrequenzen für reduzierte Dimension $n = 76$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 2000]$ Hz (Getriebe-Fein)

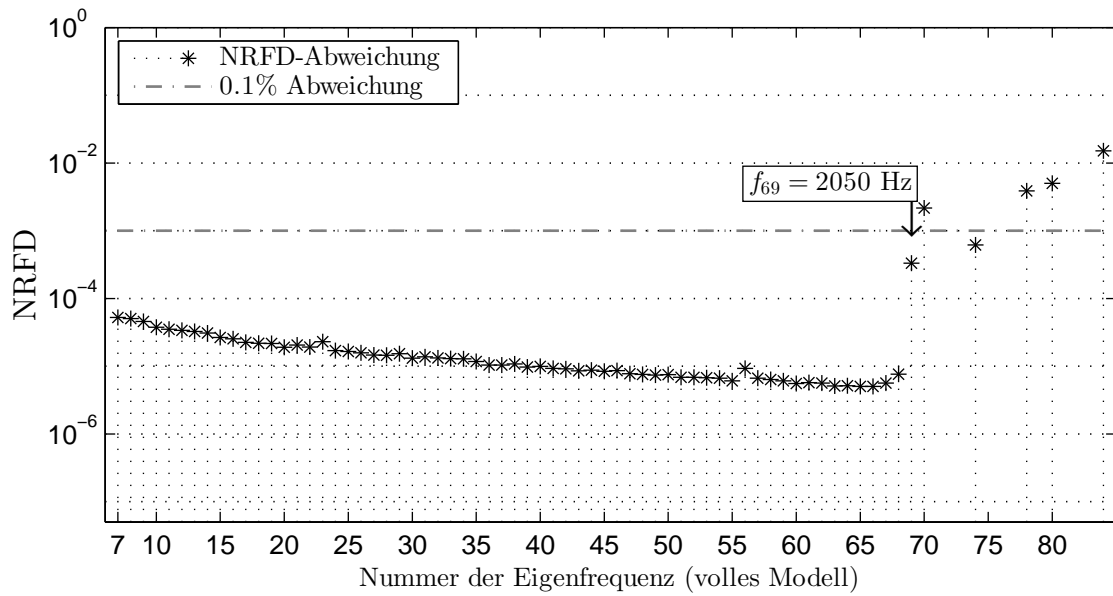


Diagramm A.26: NTFD der Eigenfrequenzen für reduzierte Dimension $n = 76$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 2000]$ Hz (Getriebe-Grob)

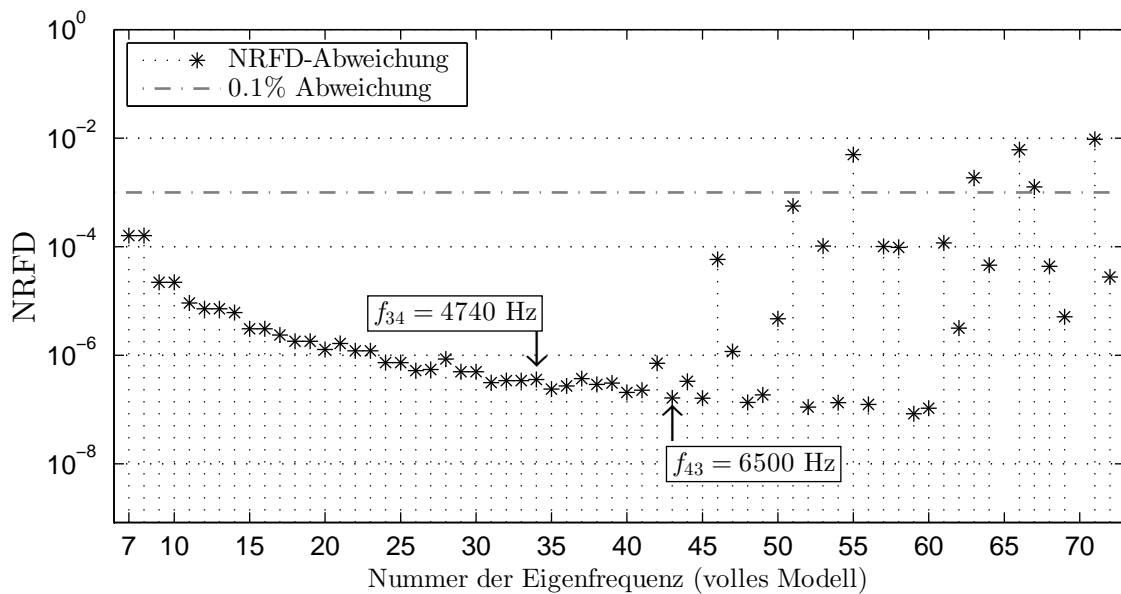


Diagramm A.27: NTFD der Eigenfrequenzen für reduzierte Dimension $n = 100$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 6500]$ Hz (Welle-Grob)

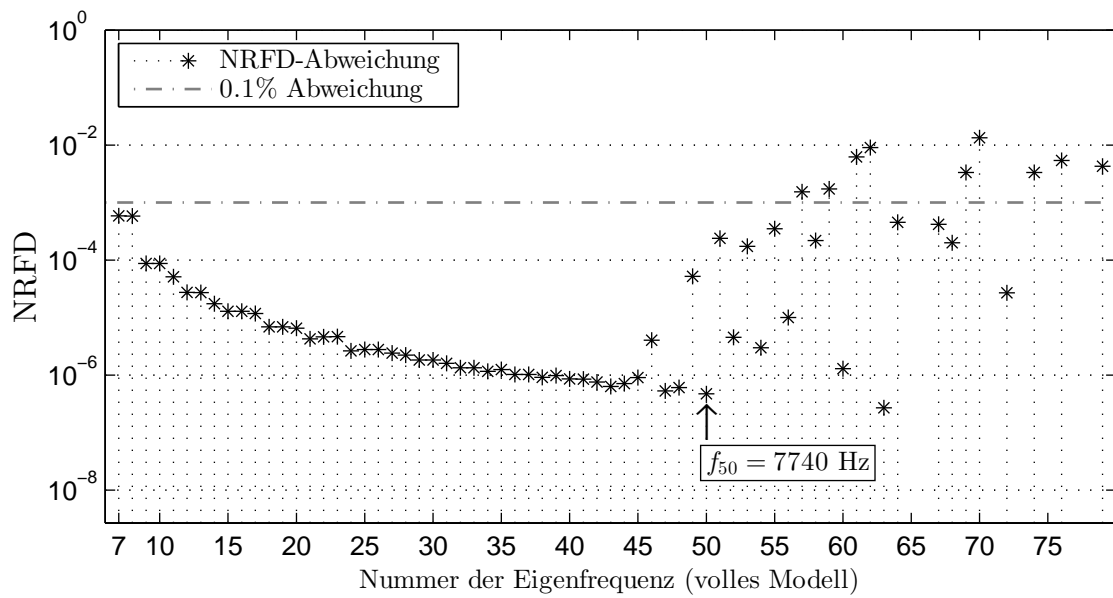


Diagramm A.28: NRFD der Eigenfrequenzen für reduzierte Dimension $n = 100$ und vier Entwicklungsstellen im Frequenzbereich $f = [0 \dots 6500] \text{ Hz}$ (Welle-Fein)