technische universität
dortmund

Fakultät für Elektrotechnik und
Informationstechnik
Lehrstuhl für Hochfrequenztechnik
Prof. Dr.-Ing. Peter M. Krummrich

Dissertation

Effiziente numerische Zeitbereichsverfahren für die elektromagnetische Analyse von Komponenten der Photonik

Hendrik Kleene

Februar 2021

Lehrstuhl für Hochfrequenztechnik

Genehmigte Dissertation zur Erlangung des akademischen Grades Doktor der Ingenieurwissenschaften (Dr.-Ing.) der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität Dortmund

Ort und Datum der Einreichung:	Dortmund, 03.12.2019
Ort und Datum der mündlichen Prüfung:	Dortmund, 19.08.2020
Hauptreferent:	apl. Prof. Dr.-Ing. Dirk Schulz
Korreferent:	PD Dr.-Ing. Stefan Helfert

Kurzzusammenfassung

Zeitbereichssimulationen sind eine häufig verwendete Methode zur Analyse von elektromagnetischen Ausbreitungsphänomenen. Dies ist unter anderem darin begründet, dass sie eine sehr allgemeine Beschreibung der Wechselwirkung der elektromagnetischen Felder und der sie umgebenden Materie erlauben. Außerdem können auch nichtlineare Effekte direkt berücksichtigt werden. Das macht diese Klasse von Algorithmen zu einem mächtigen Werkzeug bei der Analyse und Optimierung von Komponenten in vielen technischen Bereichen wie zum Beispiel der Photonik.

Die Methoden haben allerdings auch Nachteile. Die zu berechnenden Systeme resultieren in vielen praktischen Fällen in Modellen mit einer großen Anzahl von Unbekannten. Dadurch werden die Modelle schnell sehr rechenaufwändig. Zur Berechnung dieser großen Systeme ist die Verwendung von parallelen Rechnerarchitekturen unverzichtbar. Um eine gute Parallelisierung zu ermöglichen, ist die Verwendung von expliziten Algorithmen erstrebenswert. Im Gegensatz zu impliziten Algorithmen müssen bei den expliziten keine großen linearen Gleichungssysteme gelöst werden. Dies vereinfacht eine Implementierung auf parallelen und verteilten Rechnerarchitekturen. Klassische explizite Methoden weisen allerdings niedrige Lösungsordnungen auf und sind an die Wahl von kleinen Zeitschritten gebunden.

In dieser Arbeit werden alternative Algorithmen für Zeitbereichssimulationen von elektromagnetischen Ausbreitungsphänomenen auf Basis von Polynomapproximationen untersucht. Diese erlauben die Verwendung von großen Zeitschritten bei einer expliziten Formulierung der Algorithmen. Die vorgestellten Methoden ermöglichen die Berücksichtigung von diversen Materialmodellen. Es können sowohl lineare als auch nichtlineare Modelle verwendet werden. Beispiele hierfür sind dispersive Materialien oder verstärkende Medien mit Sättigungsverhalten. Außerdem werden Methoden entwickelt, mit denen elektromagnetische Felder effizient in das Simulationsgebiet eingekoppelt werden können. Gezeigt wird, dass die verwendeten Approximationen eine sehr genaue Beschreibung der Zeitpropagation erlauben. Diese Eigenschaften werden durch eine Berücksichtigung der örtlichen Diskretisierung, der Materialmodelle und der untersuchten elektromagnetischen Felder bei der Entwicklung der Approximation von der Zeitpropagation erreicht. Darüber hinaus wird eine Methode vorgestellt, welche die Charakteristik der untersuchten elektromagnetischen Felder sogar bei der Definition des Operators verwendet, welcher die Propagation beschreibt.

Dies wird bei einer weiterhin expliziten Formulierung erreicht. Es lassen sich auch nichtlineare Probleme ohne implizite Ansätze betrachten. Auf diese Weise kann eine einfache Parallelisierung selbst auf verteilten Rechnerarchitekturen gewährleistet werden.

Danksagung

Mein besonderer Dank gilt Herrn apl. Prof. Dr.-Ing. Dirk Schulz für die Betreuung dieser Arbeit und seine Unterstützung während des gesamten Projektes. Durch seine unermüdliche Diskussionsbereitschaft und seine zahllosen Anregungen hat er entscheidend zur Entstehung der Arbeit beigetragen. Er hatte stets ein offenes Ohr für meine Anliegen.

Außerdem möchte ich mich bei Herrn PD Dr.-Ing. Stefan Helfert dafür bedanken, dass er sich bereit erklärt hat, das Zweitgutachten dieser Arbeit zu übernehmen.

Herrn Prof. Dr.-Ing. Peter Krummrich möchte ich Dank sagen, dass ich diese Arbeit an seinem Lehrstuhl durchführen durfte.

Allen Kollegen, mit denen ich während meiner Zeit am Lehrstuhl für Hochfrequenztechnik zusammengearbeitet habe, haben durch ihre Bereitschaft zu einem fachlichen Austausch für eine angenehme Arbeitsatmosphäre gesorgt und waren dadurch eine wertvolle Hilfe.

Besonders möchte ich meiner Familie und meinen Freunden Danke sagen. Ihre Unterstützung und ihr Verständnis waren in Phasen, in denen der Schreibprozess nicht vorangehen wollte, ausgesprochen wertvoll.

Zuletzt gilt der Deutschen Forschungsgemeinschaft (DFG) mein Dank für die finanzielle Unterstützung des Projektes. Die vorliegende Arbeit ist im Rahmen des DFG-Projektes SCHU 1016/6-1 entstanden.

Inhaltsverzeichnis

1 Einleitung

Seit der Einführung der Maxwell-Gleichungen durch James Clerk Maxwell sind mittlerweile über 150 Jahre vergangen. Sie bilden die Grundlage für die Beschreibung einer Vielzahl von Anwendungen aus der modernen Kommunikationstechnik, der Radartechnik und der Sensorik, um nur einige Beispiele zu nennen. Aufgrund der zahlreichen Anwendungen wird seit ihrer Einführung an Lösungsverfahren für die Maxwell-Gleichungen gearbeitet. Für einfache Geometrien gibt es analytische Lösungsverfahren für die Maxwell-Gleichungen. Im Bereich der Photonik und der Terahertz-Technik sind analytische Lösungen häufig nicht möglich. Insbesondere bei der Beschreibung der Interaktion von elektromagnetischen Wellen mit der umgebenden Materie treten, aufgrund der Beschreibungen von komplexen Geometrien und Materialien, zahlreiche Herausforderungen auf. Dies führt dazu, dass für die Auslegung und die Untersuchung von neuartigen Bauteilen, Systemen und Effekten auf numerische Lösungsverfahren zurückgegriffen werden muss.

In der Vergangenheit ist zu diesem Zweck eine Vielzahl von verschiedenen numerischen Lösungsverfahren entwickelt worden. In der Photonik sind die klassischen Frequenzbereichsverfahren [1] und die sogenannte Beam Propagation Method (BPM) [2, 3] zu nennen. Bei den klassischen Frequenzbereichsverfahren werden die Maxwell-Gleichungen für jeweils eine Frequenz gelöst, während bei den BPM-Methoden die Propagation elektromagnetischer Felder entlang einer Ausbreitungsrichtung berechnet wird. Letztere zählen ebenfalls zu den Frequenzbereichsverfahren und sind insbesondere bei langen Wellenleiterstrukturen attraktiv. Oft muss allerdings auf Zeitbereichsverfahren zurückgegriffen werden. Dies begründet sich in der inhärenten Berücksichtigung von Reflexionen sowie der Möglichkeit, Systeme in einem breiten Frequenzbereich mit einer einzigen Simulation zu charakterisieren. Liegen in dem betrachteten System Materialien mit nichtlinearen Eigenschaften vor, so ist die Verwendung von Zeitbereichsverfahren in vielen Fällen alternativlos. Dies ist darin begründet, dass sie eine direkte Integration von nichtlinearen Materialmodellen erlauben. Hierfür müssen keine Annahmen getroffen werden, welche die Anwendung des Modells auf spezielle Fälle beschränken [4].

Die wohl am weitesten verbreitete Methode zur Lösung der Maxwell-Gleichungen ist die Finite-Difference Time-Domain (FDTD)-Methode, die 1966 von Yee vorgeschlagen worden ist [5]. Diese Methode basiert auf einer Zentrale-Differenzen-Approximation für die Zeit- und Ortsdifferenziale [4]. Aufgrund ihrer systematischen Formulierung und der guten Erweiterbarkeit wird die FDTD-Methode in einer Vielzahl von Anwendungsgebieten eingesetzt. Außerdem erlaubt die explizite Formulierung eine einfache Parallelisierung der Algorithmen. Ein zentraler Nachteil dieser Methode und weiterer Zeitbereichsverfahren ist die Begrenzung der Zeitschrittweite durch das Courant-Friedrich-Lewy (CFL)-Stabilitätskriterium. Dieses Kriterium koppelt die maximal verwendbare Zeitschrittweite an die verwendete örtliche Diskretisierung. Aufgrund dessen müssen für feine Diskretisierungen sehr kleine Zeitschritte gewählt werden [4]. In der Photonik liegen häufig komplexe Strukturen vor, welche eine feine Diskretisierung erfordern. Kleine Zeitschrittweiten haben längere Simulationszeiten zur Folge. Des Weiteren ist die Kopplung der Zeitschrittweite

an die örtliche Diskretisierung ungünstig, falls multiphysikalische Systeme mit Effekten verschiedener Zeitskalen betrachtet werden. Vor diesem Hintergrund ist eine flexible Wahl der Zeitschrittweite wünschenswert, die eine Anpassung an die untersuchten elektromagnetischen Felder sowie die Eigenschaften der Modelle im untersuchten System erlaubt.

Für einige Anwendungen kann darüber hinaus die niedrige Lösungsordnung der Methode ein Nachteil darstellen, da so die numerisch berechnete Wellenausbreitung von der im realen System abweicht. Die Fehler machen sich bei zunehmender Strukturgröße durch eine fehlerhafte Beschreibung der Ausbreitungsgeschwindigkeiten, der sogenannten numerischen Dispersion, bemerkbar. Insbesondere im Zusammenspiel mit nichtlinearen Effekten bei großen Systemen kann dies zu zusätzlichen Fehlern führen [6, 7].

Zur Minimierung dieser Effekte wird in der Literatur eine Vielzahl von verschieden Ansätzen untersucht. Hierzu werden im Kontext der FDTD-Methode die alternating-direction implicit (ADI)-Methode [8], locally one-dimensional (LOD)-Methoden [9, 10] und Crank-Nicholson-Ansätze [11] in der Literatur untersucht.

Während diese Ansätze größere Zeitschrittweiten erlauben, resultieren sie allerdings in impliziten Algorithmen. Die implizite Formulierung hat zur Folge, dass in jedem Zeitschritt lineare Gleichungssysteme gelöst werden müssen. Für Systeme mit einer moderaten Anzahl von Unbekannten kann eine geringere Rechenzeit als bei der klassischen FDTD-Methode erzielt werden. Große Systeme mit vielen Unbekannten erfordern eine Parallelisierung auf verteilten Rechnersystemen. Bei der Verwendung von impliziten Verfahren für Systeme mit vielen Unbekannten ist allerdings die Parallelisierung der Algorithmen für verteilte Rechnersysteme erschwert. Während bei expliziten Verfahren immer nur die direkt benachbarten Feldwerte benötigt werden, sind bei der impliziten Berechnungsvorschrift die Elemente in einem linearen Gleichungssystem miteinander gekoppelt, welches gelöst werden muss. Dies führt zu einem erheblichen Kommunikationsaufwand zwischen den Rechenknoten. Außerdem weisen diese Verfahren für Zeitschrittweiten größer als das CFL-Limit mitunter zusätzliche numerische Dispersionsfehler auf [12, 13].

Daher sollen im Rahmen dieser Arbeit alternative Ansätze zur Realisierung der Zeitpropagation untersucht werden. Diese Ansätze sollen eine flexible Einbindung von Materialmodellen erlauben und nicht auf bestimmte Approximationen oder Vereinfachungen bezüglich der Materialeigenschaften angewiesen sein. Im Hinblick auf die Implementierung der Algorithmen auf parallelen Rechnerarchitekturen sollen nur explizite Algorithmen betrachtet werden. Analog zu dem klassischen FDTD-Ansatz soll eine unproblematische Implementierung auf parallelen Rechnersystemen, wie modernen Central Processing Units (CPUs), Graphics Processing Units (GPUs) oder Field Programmable Gate Arrays (FPGAs) sowie verteilten Rechnersystemen, ermöglicht werden. Zu diesem Zweck sollen Zeitpropagationsalgorithmen auf Basis von Polynomapproximationen entwickelt werden. Mit diesen Approximationen sollen Algorithmen formuliert werden, die Zeitschrittweiten erlauben, welche das CFL-Limit überschreiten. Hierdurch soll die verwendete Zeitschrittweite von der örtlichen Diskretisierung entkoppelt werden. Im Hinblick auf die Untersuchung großer Systeme ist außerdem eine möglichst genaue Beschreibung der Zeitpropagation von entscheidender Bedeutung.

Um diese Ziele zu erreichen, sollen die Eigenschaften der Systemmatrizen der betrachteten Modelle untersucht werden. Die Eigenschaften des Eigenwertspektrums der Systemmatrizen enthalten hierbei Informationen über das betrachtete System. Diese Eigenschaften sollen für eine effizientere Approximation der Zeitpropagation verwendet werden. Darüber hinaus sollen die

Eigenschaften der untersuchten Feldverteilungen berücksichtigt werden. In vielen technischen Systemen wird die Ausbreitung von elektromagnetischen Feldern untersucht, welche um eine Trägerfrequenz bandbegrenzt sind. Dieses Wissen soll in die Approximation des Zeitpropagationsschemas einfließen.

Die entwickelten Algorithmen sollen mit den bekannten Vergleichsalgorithmen wie der klassischen FDTD-Methode verglichen werden. Für die Vergleiche sollen von der Implementierung unabhängige Bewertungsmaßstäbe für den Rechenaufwand verwendet werden. Bei der Bewertung der Algorithmen soll neben dem Rechenaufwand auch die Genauigkeit der Algorithmen in die Bewertung der Effizienz mit aufgenommen werden. Die Betrachtung soll auf die Realisierung der Zeitpropagation beschränkt werden. Die Algorithmen werden so konstruiert, dass verschiedene Ortsdiskretisierungsverfahren verwendet werden können, sofern sie eine explizite Formulierung erlauben. Hierbei sind zum Beispiel das Yee-Gitter [5], pseudospektrale Ansätze [14, 15] und Finite-Elemente-Methoden auf Basis des diskontinuierlichen Galerkin-Ansatzes [16, 17] zu nennen.

In Kapitel 2 werden die theoretischen Grundlagen der untersuchten Materialmodelle zusammengefasst und in Kapitel 3 werden die benötigten numerischen Methoden beschrieben. Im Anschluss wird in Kapitel 4 ein erster Ansatz auf Basis von Polynomapproximationen untersucht und charakterisiert. Mit den gewonnenen Erkenntnissen wird in Kapitel 5 eine Faberpolynom-Methode zur Zeitpropagation vorgestellt und untersucht. In Kapitel 6 wird die Faberpolynom-Methode um die Möglichkeit, Quellterme einzubinden, erweitert, zum Beispiel in Form von Stromdichten. Anschließend wird der Algorithmus in Kapitel 7 für nichtlineare Probleme verallgemeinert. In Kapitel 8 werden die gewonnenen Erkenntnisse auf Algorithmen der Klasse der Time Domain Beam Propagation (TDBPM)-Algorithmen angewendet. Durch eine Operatorapproximation mit den Faberpolynomen wird eine explizite Formulierung ermöglicht. Im nachfolgenden Kapitel werden einige Anwendungsbeispiele mit den erarbeiteten Algorithmen simuliert. Abschließend werden die Ergebnisse in Kapitel 10 zusammengefasst und ein Ausblick auf mögliche zukünftige Untersuchungen gegeben.

2 Theoretische Grundlagen der Modellierung

Zunächst sollen einige theoretische Grundlagen für die mathematische Beschreibung der untersuchten Modelle vorgestellt werden. Hierzu wird zuerst auf die nötige elektromagnetische Feldtheorie eingegangen. Im Anschluss werden einige Materialmodelle betrachtet, welche in dieser Arbeit verwendet werden. Darüber hinaus wird eine Formulierung vorgestellt, mit der die Materialmodelle für die später untersuchten Algorithmen effizient eingebunden werden können. Danach wird auf Randbedingungen zur Begrenzung des Simulationsgebietes eingegangen. Im Anschluss soll die Einbindung von elektromagnetischen Feldern in die Zeitbereichssimulation mithilfe geeigneter Stromdichten in den Blick genommen werden. Dies soll am Beispiel der Einkopplung von Eigenmoden an einem dielektrischen Wellenleiter noch veranschaulicht werden.

2.1 Klassischer Elektromagnetismus

Bevor die numerischen Lösungsverfahren beschrieben werden, sollen im Folgenden einige Grundlagen der klassischen Feldtheorie präsentiert werden. Diese Zusammenfassung beschränkt sich auf die Zusammenhänge, welche wichtig für die betrachteten numerischen Verfahren oder die Notation sind. Für ausführliche Betrachtungen sei auf Standardwerke wie [18] verwiesen.

2.1.1 Maxwell-Gleichungen

Die von James Clerk Maxwell eingeführten Maxwell-Gleichungen beschreiben den Zusammenhang zwischen elektrischen und magnetischen Feldern und die Interaktion der elektromagnetischen Felder mit dem umgebenen Medium. Die Maxwell-Gleichungen sind gegeben durch:

$$\nabla \cdot \vec{D} = \rho \tag{2.1a}$$

$$\nabla \cdot \vec{B} = 0 \tag{2.1b}$$

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} - \vec{K} \tag{2.1c}$$

$$\nabla \times \vec{H} = \frac{\partial \vec{D}}{\partial t} + \vec{J}. \tag{2.1d}$$

Das elektrische Feld ist durch $\vec{E}$ gegeben, während das magnetische Feld durch $\vec{H}$ gegeben ist. Die Größen $\vec{D}$ und $\vec{B}$ stehen für die elektrische beziehungsweise magnetische Flussdichte. Die elektrische Stromdichte ist mit $\vec{J}$ gegeben, während mit $\vec{K}$ eine formal eingeführte magnetische Stromdichte gegeben ist. Diese wird später zur Einkopplung von Wellen in die Simulationsmodelle benötigt. Mit ρ ist die Ladungsdichte gegeben. Für ein System ohne freie elektrische Ladung, also $\rho = 0$, ist die Beschreibung der zeitlichen Propagation von elektromagnetischen Wellen

durch die letzten beiden Gleichungen in (2.1) gegeben [4]. Der Zusammenhang zwischen der elektrischen Feldstärke $\vec{E}$ und der elektrischen Flussdichte $\vec{D}$ sowie der Zusammenhang zwischen $\vec{H}$ und $\vec{B}$ kann allgemein durch die Einführung von einer elektrischen Polarisation $\vec{P}$ und einer magnetischen Polarisation $\vec{M}$ hergestellt werden:

$$\vec{D} = \epsilon_0 \vec{E} + \vec{P} \tag{2.2a}$$

$$\vec{B} = \mu_0 \vec{H} + \vec{M}. \tag{2.2b}$$

Für den linearen, isotropen und nicht dispersiven Fall kann (2.2) auch als $\vec{D} = \epsilon \vec{E}$ und $\vec{B} = \mu \vec{H}$ dargestellt werden. Hier gilt $\epsilon = \epsilon_0 \epsilon_r$, wobei ϵ_0 die elektrische Permittivität im Vakuum ist und ϵ_r die relative Permittivität ist. Die Permeabilität im Vakuum ist mit μ_0 gegeben und die relative Permeabilität μ_r ist analog mit $\mu = \mu_0 \mu_r$ definiert. Damit ergibt sich für ein System ohne elektrische oder magnetische Stromdichten das Induktions- und das Durchflutungsgesetz in (2.1)

$$\vec{\nabla} \times \vec{E} = -\mu \frac{\partial \vec{H}}{\partial t} \Leftrightarrow \begin{cases} \frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\mu \frac{\partial H_x}{\partial t} \\ \frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -\mu \frac{\partial H_y}{\partial t} \\ \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\mu \frac{\partial H_z}{\partial t} \end{cases} \tag{2.3}$$

und

$$\vec{\nabla} \times \vec{H} = \epsilon \frac{\partial \vec{E}}{\partial t} \Leftrightarrow \begin{cases} \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} = \epsilon \frac{\partial E_x}{\partial t} \\ \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = \epsilon \frac{\partial E_y}{\partial t} \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = \epsilon \frac{\partial E_z}{\partial t}. \end{cases} \tag{2.4}$$

Zweidimensionaler und eindimensionaler Fall

Im Folgenden soll auf einige Spezialfälle eingegangen werden, die sich für ein- und zweidimensionale Systeme ergeben. Indem in (2.3) und (2.4) alle Größen als invariant in Richtung der z-Koordinate angenommen werden, wodurch $\frac{\partial}{\partial z} = 0$ gilt, kann ein vereinfachter Ausdruck für das zweidimensionale System abgeleitet werden:

$$\vec{\nabla} \times \vec{E} = -\mu \frac{\partial \vec{H}}{\partial t} \Leftrightarrow \begin{cases} -\frac{\partial E_z}{\partial y} = -\mu \frac{\partial H_x}{\partial t} \\ -\frac{\partial E_z}{\partial x} = -\mu \frac{\partial H_y}{\partial t} \\ \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\mu \frac{\partial H_z}{\partial t} \end{cases} \tag{2.5}$$

$$\vec{\nabla} \times \vec{H} = \epsilon \frac{\partial \vec{E}}{\partial t} \Leftrightarrow \begin{cases} \frac{\partial H_z}{\partial y} = \epsilon \frac{\partial E_x}{\partial t} \\ -\frac{\partial H_z}{\partial x} = \epsilon \frac{\partial E_y}{\partial t} \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = \epsilon \frac{\partial E_z}{\partial t}. \end{cases} \tag{2.6}$$

Das System von partiellen Differenzialgleichungen lässt sich in zwei voneinander unabhängige Systeme zerlegen, welche sich auch unabhängig voneinander berechnen lassen. Für die Notation dieser Moden wird in dieser Arbeit die Notation aus [4, 19] verwendet. In der Wellenleitertheorie ist wiederum eine umgekehrte Definition in Verwendung [20]. Das erste ist die sogenannte transversal elektrische (TE) Mode [4], da nur die elektrischen Feldkomponenten E_x und E_y in den Ausbreitungsrichtungen x, y vorliegen:

$$\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\mu \frac{\partial H_z}{\partial t} \tag{2.7}$$

$$\frac{\partial H_z}{\partial y} = \epsilon \frac{\partial E_x}{\partial t} \tag{2.8}$$

$$-\frac{\partial H_z}{\partial x} = \epsilon \frac{\partial E_y}{\partial t}. \tag{2.9}$$

Bei dem zweiten Teil handelt es sich um die transversal magnetische (TM) Mode, bei der nur die magnetischen Feldkomponenten H_x und H_y in Ausbreitungsrichtung vorliegen:

$$\frac{\partial E_z}{\partial y} = -\mu \frac{\partial H_x}{\partial t} \tag{2.10}$$

$$-\frac{\partial E_z}{\partial x} = -\mu \frac{\partial H_y}{\partial t} \tag{2.11}$$

$$\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = \epsilon \frac{\partial E_z}{\partial t}. \tag{2.12}$$

Werden in (2.3) und (2.4) alle Größen als invariant in x und y Richtung angenommen, lässt sich das eindimensionale System ableiten. Wie im zweidimensionalen Fall liegen wieder zwei unabhängige Systeme vor:

$$\frac{\partial E_x}{\partial z} = -\mu \frac{\partial H_y}{\partial t} \tag{2.13}$$

$$-\frac{\partial H_y}{\partial z} = \epsilon \frac{\partial E_x}{\partial t} \tag{2.14}$$

$$-\frac{\partial E_y}{\partial z} = -\mu \frac{\partial H_x}{\partial t} \tag{2.15}$$

$$\frac{\partial H_x}{\partial z} = \epsilon \frac{\partial E_y}{\partial t}. \tag{2.16}$$

2.1.2 Operatordarstellung

Um eine kompakte Notierung zu erlauben, wird in dieser Arbeit eine Operator-Notation zur Darstellung der Maxwellgleichungen verwendet. Neben der kompakteren Darstellung der Gleichungen mit zusätzlichen Materialmodellen, welche das betrachtete System beschreiben, vereinfacht diese Darstellung die Anwendung der Operatorentwicklungs-Methoden, welche in dieser Arbeit betrachtet werden. Hierzu werden die Maxwell-Gleichungen in die folgende Form gebracht:

$$\frac{\partial}{\partial t}\vec{\psi}(\vec{r},t) = \begin{bmatrix} 0 & \frac{1}{\epsilon(\vec{r})}\nabla\times \\ -\frac{1}{\mu(\vec{r})}\nabla\times & 0 \end{bmatrix} \vec{\psi}(\vec{r},t). \tag{2.17}$$

Der Vektor $\vec{\psi}(\vec{r},t)$ enthält alle Feldkomponenten und ist gegeben durch:

$$\vec{\psi}(\vec{r},t) = \begin{bmatrix} \vec{E}(\vec{r},t) & \vec{H}(\vec{r},t) \end{bmatrix}^{\mathrm{T}}. \tag{2.18}$$

Werden zusätzliche Variablen und Gleichungen, wie zum Beispiel durch komplexere Materialmodelle mitberücksichtigt, so werden sie dem Vektor $\vec{\psi}(\vec{r},t)$ angefügt und der Matrixoperator in (2.17) wird um die entsprechenden Operatoren erweitert. Diese Schreibweise bildet die Grundlage der im Rahmen dieser Arbeit entwickelten Algorithmen.

2.2 Materialmodelle

Die bisher eingeführten Gleichungen erlauben bereits die Analyse von einer Vielzahl von Systemen. Allerdings ist die Darstellung auf lineare und nicht dispersive Systeme begrenzt. Die Materialparameter dürfen daher zum Beispiel keine Frequenzabhängigkeit haben. Im Frequenzbereich optischer Strahlung haben viele Materialien allerdings dispersive Eigenschaften. Außerdem spielt bei hohen Frequenzen auch in metallischen Materialien die begrenzte Elektronenbeweglichkeit eine zunehmende Rolle [21]. Das Metall kann daher nicht mehr als perfekt leitend angenommen werden. Daher sollen in diesem Abschnitt einige wichtige Materialmodelle vorgestellt werden, welche später in die numerischen Modelle mit aufgenommen werden.

2.2.1 Drude-Modell

Die Wechselwirkung zwischen elektromagnetischen Feldern und Metallen können in einem weiten Frequenzbereich mit dem sogenannten Drude-Modell beschrieben werden. Das Modell ist 1900 von Paul Drude eingeführt worden [22]. Dieses Modell findet beispielsweise bei der Beschreibung von plasmonischen Effekten Anwendung [23, KS1]. Bei diesem Modell wird das Metall als Kristall ortsfester Ionen angenommen, durch die sich die Elektronen frei bewegen können. Nach [23] ist die Bewegungsgleichung der Ladungsträger mit

$$m\frac{d^2}{dt^2}\vec{x} + m\gamma_D\frac{d}{dt}\vec{x} = -e\vec{E}. \tag{2.19}$$

gegeben. Hierbei gibt m die Masse der oszillierenden Elektronen und e gibt die Elementarladung an. Bei γ_D handelt es sich um die Kollisionsfrequenz. Die Größe $\vec{x}$ gibt die Auslenkung des Elektrons an. Nun wird (2.19) in den Fourier-Bereich transformiert. Im Anschluss wird die Gleichung nach $\vec{x}(\omega)$ umgeformt und die makroskopische Polarisation $\vec{P}_D = ne\vec{x}$ wird mit

$$\vec{P}_D = -\frac{ne^2}{m(\omega^2 + j\gamma_D\omega)}\vec{E} \tag{2.20}$$

bestimmt. Diese wird über den Polarisationsterm in (2.2) berücksichtigt, sodass sich

$$\vec{D} = \epsilon_0(1 - \frac{\omega_D^2}{(\omega^2 + j\gamma_D\omega)})\vec{E} \tag{2.21}$$

ergibt, wobei mit $\omega_D^2 = \frac{ne^2}{\epsilon_0 m}$ die Plasmafrequenz des Elektronengases gegeben ist. Mithilfe von (2.21) kann nun auch ein Ausdruck für die Permittivität des Metalls gefunden werden:

$$\epsilon(\omega) = \epsilon_0\epsilon_\infty - \frac{\epsilon_0\omega_D^2}{(\omega^2 + j\gamma_D\omega)}. \tag{2.22}$$

Hierbei wird in der Regel noch ein Parameter ϵ_∞ verwendet, welcher die Permittivität für $\omega >> \omega_D$ angibt. Die Parameter ω_D, γ_D und ϵ_∞ werden in der Regel an experimentell ermittelte Werte für $\epsilon(\omega)$ angepasst [23]. Diese Parameter können für einige Edelmetalle beispielsweise in [21] gefunden werden.

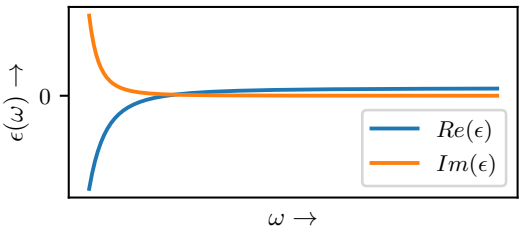


Abbildung 2.1: Die Abbildung zeigt schematisch den Verlauf der Permittivität ϵ aufgeteilt nach Real- und Imaginärteil über der Frequenz ω für ein Drude-Modell.

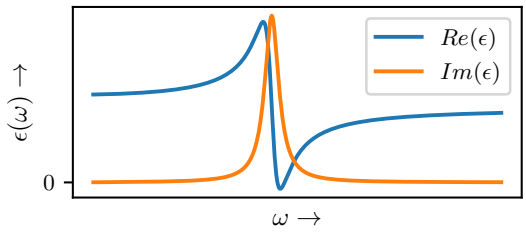


Abbildung 2.2: In der Abbildung ist der schematische Verlauf des ϵ über die Frequenz ω bei einem Lorentz-Modell.

2.2.2 Lorentz-Modell

Mit dem Drude-Modell ist ein erstes Modell für die Interaktion der freien Elektronen in einem Metall mit einem externen elektromagnetischen Feld gegeben. Die Modellierung der Interaktion mit den gebundenen Elektronen eines Festkörpers ist durch das Lorentz-Modell möglich, was eine Beschreibung des Frequenzgangs der Permittivität in dem betrachteten Material erlaubt. Die Interaktion der Elektronen mit dem externen elektromagnetischen Feld wird mithilfe eines gedämpften harmonischen Oszillators beschrieben [24]. Die Bewegungsgleichung eines Elektrons, welches vom externen elektromagnetischen Feld angeregt wird, ist nach dem Lorentz-Modell durch

$$m\frac{\delta^2}{\delta t^2}\vec{x} + m\gamma_L\frac{\delta}{\delta t}\vec{x} + m\omega_L^2\vec{x} = -e\vec{E} \tag{2.23}$$

gegeben [24]. Hierbei gibt $\vec{x}$ die Auslenkung der Elektronen an, während m die Masse beschreibt. Der Parameter γ_L gibt die Dämpfung des Oszillators an. ω_L ist die Resonanzfrequenz des Oszillators. Wird (2.23) über die Polarisation in (2.2) miteinbezogen, kann folgender Zusammenhang für die Frequenzabhängigkeit der Permittivität des Materials abgeleitet werden [24]:

$$\epsilon(\omega) = \epsilon_0\epsilon_\infty + \frac{\epsilon_0\Delta\epsilon\omega_L^2}{\omega_L^2 + j\gamma_L - \omega^2}. \tag{2.24}$$

Der Parameter ϵ_∞ gibt hierbei wieder die relative Permittivität für hohe Frequenzen an, während über den Parameter $\Delta\epsilon = \epsilon_s - \epsilon_\infty$ die Permittivität im statischen Fall berücksichtigt wird [25]. Der Real- und der Imaginärteil der Permittivität von (2.24) ist in Abbildung 2.2 dargestellt. In der Regel haben Festkörper mehrere dieser Resonanzen, sodass für eine präzisere Beschreibung eine Überlagerung von mehreren Lorentz-Oszillatoren verwendet wird. Diese werden entsprechend ihrer Ausprägung mit den Faktoren a_i gewichtet. Die Permittivität, gegeben durch ein Lorentz-Modell mit N Oszillatoren, kann mit

$$\epsilon(\omega) = \epsilon_0\epsilon_\infty + \sum_{i=1}^{N}\frac{a_i\omega_{L,i}^2}{\omega_{L,i}^2 - \omega^2 + j\gamma_{L,i}\omega} \tag{2.25}$$

dargestellt werden. Wie beim zuvor beschriebenen Drude-Modell werden die Parameter des Lorentz-Modells in der Regel experimentell ermittelt und an das gewählte Modell angepasst, um die besten Ergebnisse zu erzielen.

2.2.3 Nichtlineare Effekte

Bei den in den vorangegangenen Abschnitten betrachteten Materialmodellen handelt es sich um lineare Modelle. In diesem Abschnitt sollen nichtlineare Materialien in den Blick genommen werden. Eine Vielzahl von wichtigen technischen Anwendungen erfordern darüber hinaus die Berücksichtigung von nichtlinearen Effekten, um ihr Verhalten korrekt zu erfassen. Beispiele hierfür sind unter anderem Modulatoren, welche zum Beispiel mithilfe des elektrooptischen Effektes realisiert werden können. Auch die Beschreibung von Verstärkern mit Sättigungsverhalten führt auf nichtlineare Zusammenhänge. Außerdem können nichtlineare Effekte bei langen Propagationswegen, wie zum Beispiel in Glasfasern, in Kombination mit hohen Leistungen signifikanten Einfluss auf die Funktion der betrachteten Komponenten haben. Daher kann auch hier die Berücksichtigung dieser Effekte wichtig werden, um das Verhalten der Komponenten zu erfassen.

Für die Modellierung von nichtlinearen Effekten eignen sich unter anderem besonders die in dieser Arbeit betrachteten Zeitbereichsverfahren. Bei diesen können die nichtlinearen Effekte in der Regel direkt angegeben werden, da die ortsabhängigen Feldgrößen direkt im Zeitbereich vorliegen [4]. Außerdem ermöglicht die Betrachtung in Kombination mit den vollständigen Maxwell-Gleichungen die Studie von Effekten, welche durch die Wechselwirkung der Felder mit der umliegenden Geometrie der Strukturen und den nichtlinearen Effekten zustande kommen.

Im Folgendem sollen daher die physikalischen Grundlagen der in dieser Arbeit betrachteten Effekte kurz umrissen werden. Zunächst soll auf die Beschreibung der nichtlinearen Suszeptibilität eingegangen werden. Im Anschluss ein Zwei-Niveau-Modell in den Blick genommen. Bei diesem wird ein Mikrosystem mit zwei Energieniveaus und dessen Kopplung mit dem externen elektrischen Feld modelliert. Dies erlaubt zum Beispiel die Beschreibung von Verstärkern oder absorbierendem Verhalten mit Sättigungseffekten. Für eine ausführliche Diskussion sei auf die Literatur verwiesen [26].

Nichtlineare Suszeptibilität

Zunächst soll von einem Material ohne dispersive Eigenschaften ausgegangen werden. Im linearen und nicht dispersiven Fall hängt die Polarisation $\vec{P}$ dann von der linearen Suszeptibilität ab:

$$\vec{P} = \epsilon \vec{E} = \epsilon_0 \chi^{(1)} \vec{E}. \tag{2.26}$$

Für hohe Feldstärken kann die Polarisation $\vec{P}$ nicht mehr als linear abhängig von der Feldstärke approximiert werden. Stattdessen kann die Polarisation $\vec{P}$ als Funktion betrachtet werden, welche nichtlinear von der Feldstärke $\vec{E}$ abhängt. In der Regel wird diese als Taylorreihen-Entwicklung dargestellt [26]:

$$P = \epsilon_0(\chi^{(1)}\vec{E} + \chi^{(2)}\vec{E}^2 + \chi^{(3)}E^3 + \chi^{(4)}E^4 + ...). \tag{2.27}$$

Diese Koeffizienten haben bei gängigen Materialien Werte, welche klein gegenüber der linearen Suszeptibilität $\chi^{(1)}$ sind. Für Materialien mit einem starken elektrooptischen Effekt zweiter Ordnung $\chi^{(2)}$ und dritte Ordnung $\chi^{(3)}$ sind dies zum Beispiel Werte im Bereich [26]

$$\chi^{(2)} = 1{,}94\,\mathrm{pm/V} \quad \text{und} \quad \chi^{(3)} = 3{,}78\,\mathrm{pm^2/V^2}.$$

Die Beschreibung (2.27) nimmt eine sofortige Reaktion des Mediums auf das Feld an. Das Medium muss daher aufgrund der Kramers-Kronig-Relation dämpfungsfreie und nicht dispersive Materialeigenschaften aufweisen [26]. In der Regel weisen die nichtlinearen Suszeptibilitäten $\chi^{(i)}$ zusätzlich eine Richtungsabhängigkeit auf, sodass sie durch Tensoren beschrieben werden müssen [26].

Zwei-Niveau-Modell

Eine präzisere Beschreibung der Interaktion des elektrischen Feldes mit der umliegenden Materie kann mithilfe eines Zwei- oder Mehrniveau-Modell erfolgen [26, 27]. Die Modellierung des strahlenden Überganges und die Kopplung an das Elektromagnetische Feld erfolgt mithilfe eines Lorentz-Oszillators

$$\frac{\partial^2 \vec{P}_m}{\partial t^2} + \Gamma_m \frac{\partial \vec{P}_m}{\partial t} + \left(\omega_m^2 + \frac{\Gamma_m^2}{4}\right)\vec{P} = \sigma_m \Delta N \vec{E}, \tag{2.28}$$

wobei $\Delta N = N_2 - N_1$ ist [26, 28, 29]. Die Größe N_1 gibt hierbei die Besetzungsdichte in dem Niveau eins an, während N_2 die des Niveaus zwei angibt. Weiterhin entspricht ω_m der Übergangsfrequenz, während Γ_m die Halbwertsbreite (FWHM) des Übergangs angibt. Mit σ_m ist die Stärke der Kopplung an das externe elektrische Feld gegeben. Für die Besetzungsdichten gilt mit dem strahlenden Übergang (2.28) [26–29]:

$$\begin{aligned}
\frac{\partial \vec{N}_2}{\partial t} &= \frac{1}{\hbar\omega_m}\vec{E}\left(\frac{\partial \vec{P}_m}{\partial t} + \frac{\Gamma}{2}\vec{P}_m\right) + \gamma_{12}N_1 - \gamma_{21}N_2 \\
\frac{\partial \vec{N}_1}{\partial t} &= -\frac{1}{\hbar\omega_m}\vec{E}\left(\frac{\partial \vec{P}_m}{\partial t} + \frac{\Gamma}{2}\vec{P}_m\right) - \gamma_{12}N_1 + \gamma_{21}N_2.
\end{aligned} \tag{2.29}$$

Die Größen γ_{12} und γ_{21} geben die Übergangsraten zwischen den Niveaus an. Hierbei handelt es sich um die Formulierung, welche direkt aus der Dichtematrix-Formulierung abgeleitet werden kann [26]. In vielen Fällen werden der Term $\frac{\Gamma^2}{4}\vec{P}$ in (2.28) und die $\frac{\Gamma}{2}\vec{P}$ Terme in (2.29) vernachlässigt [26]. Diese Darstellung ermöglicht die Beschreibung von Absorption mit Sättigung [30–32]. Außerdem kann das Modell als einfaches Verstärkermodell mit Sättigung herangezogen werden. Um die dynamischen Eigenschaften von komplexeren Mikrosystemen zu erfassen, kann die beschriebene Formulierung um mehr Niveaus ergänzt werden [27, 28, 30]. Hierbei wird jeder zusätzliche strahlende Übergang durch eine weitere Oszillator-Gleichung (2.28) beschrieben.

2.3 Einbindung der Materialmodelle

Im Folgenden soll auf die Einbindung der zuvor beschriebenen Materialmodelle in die untersuchten Zeitbereichssimulationen eingegangen werden. Hierzu wird in dieser Arbeit ein Auxiliary-Differential-Equation (ADE)-Ansatz herangezogen. Dieser stellt eine systematische Möglichkeit dar, diverse lineare Materialmodelle in das Simulationsmodell mit aufzunehmen [33].

Die Frequenzabhängigkeit der Permittivität, welche zum Beispiel durch Drude- oder Lorentz-Modelle eingeführt wird, kann bei der Einbindung in die Zeitbereichssimulation eine Reihe

von Problemen zur Folge haben. Eine direkte Einbindung, welche zum Beispiel für die FDTD Methode eingesetzt wird, führt auf Faltungsterme zur Beschreibung des dispersiven Verhaltens der Materialien [4]. Dieser klassische Ansatz kann allerdings zu einer Reihe von Problemen führen. Zum Einem kann die Fehlerordnung bei ungeeigneter Wahl des Zeitpropagationschemas reduziert werden [4]. Außerdem kann die numerische Berechnung des Faltungsintegrals Probleme im Hinblick auf die Genauigkeit bereiten [4]. Darüber hinaus sind die ADEs auch eine Möglichkeit, die später beschriebenen Perfectly-Matched-Layer (PML) zu implementieren. Daher sollen die Betrachtungen in dieser Arbeit auf die oben genannte ADE-Methode beschränkt werden. Diese sollen am Beispiel des Lorentz- und des Drude-Modells erläutert werden. Andere Modelle können mit ähnlichen Ansätzen in die Simulation eingebunden werden [4]. Die Polarisation in dem oben beschriebene Lorentz-Modell wird von der folgenden Gleichung beschrieben [34]:

$$\frac{\partial^2}{\partial t^2}\vec{P}_L + \gamma_L \frac{\partial}{\partial t}\vec{P}_L + \omega_L^2 \vec{P}_L = \epsilon_0 \Delta\epsilon \omega_L^2 \vec{E}. \tag{2.30}$$

Für die elektrische Flussdichte $\vec{D}$ gilt

$$\vec{D} = \epsilon_0 \epsilon_\infty \vec{E} + \vec{P}_L. \tag{2.31}$$

Nun soll (2.30) auf ein System von Gleichungen erster Ordnung zurückgeführt werden. Hierzu wird mit

$$\vec{J}_L = \frac{\partial}{\partial t}\vec{P}_L \tag{2.32}$$

ein Polarisationsstrom definiert. Wird (2.32) in (2.30) eingesetzt, so lässt sich folgende Gleichung herleiten:

$$\frac{\partial}{\partial t}\vec{J}_L = -\gamma_L \vec{J}_L - \omega_L^2 \vec{P}_L + \epsilon_0 \Delta\epsilon \omega_L^2 \vec{E}. \tag{2.33}$$

Nun soll die durch das Lorentz-Medium hervorgerufene Polarisation in die Maxwell-Gleichungen eingearbeitet werden. Mit (2.31) und (2.32) gilt:

$$\frac{\partial}{\partial t}\vec{D} = \frac{\partial}{\partial t}(\epsilon_0 \epsilon_\infty \vec{E} + \vec{P}_L) = \epsilon_0 \epsilon_\infty \frac{\partial}{\partial t}\vec{E} + \vec{J}_L. \tag{2.34}$$

Wird dies in die Maxwell-Gleichungen (2.1) eingesetzt, kann das System in die oben beschriebene Operatordarstellung umgeschrieben werden:

$$\frac{\partial}{\partial t}\vec{\psi} = \begin{bmatrix} 0 & \frac{1}{\epsilon_0\epsilon_\infty}\nabla\times & 0 & -\frac{1}{\epsilon_0\epsilon_\infty} \\ -\frac{1}{\mu}\nabla\times & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \epsilon_0\Delta\epsilon\omega_0^2 & 0 & -\omega_0^2 & -\gamma_L \end{bmatrix} \vec{\psi}. \tag{2.35}$$

Hierbei ist der Vektor $\vec{\psi}$ mit $\vec{\psi} = [\vec{E}, \vec{H}, \vec{P}_L, \vec{J}_L]^T$ gegeben. Damit ist der Ansatz kompatibel zu den im Laufe dieser Arbeit vorgestellten Operatorentwicklungs-Methoden. Sollen wie in (2.25) mehrere Oszillatoren berücksichtigt werden, so wird das Verfahren jeweils für die einzelnen Oszillatoren angewandt.

Alternativ kann das oben vorgestellte System auch für den klassischen FDTD-Algorithmus genutzt werden [4, 35]. Hierbei muss insbesondere auf die Abtastung in der Zeit der einzelnen Variablen geachtet werden. Im Ort sind $\vec{P}_L$ und $\vec{J}_L$ mit dem elektrischen Feld $\vec{E}$ ko-allokiert.

Das Drude-Modell kann auf ähnliche Weise eingeführt werden. Die hervorgerufene Polarisation wird durch

$$\frac{\partial^2}{\partial t^2}\vec{P}_D + \gamma_D \frac{\partial}{\partial t}\vec{P}_D = \epsilon_0 \omega_D^2 \vec{E} \tag{2.36}$$

beschrieben. Durch das Einführen des Polarisationsstroms (2.32) kann die Gleichung wieder auf ein System von Differenzialgleichungen erster Ordnung zurückgeführt werden:

$$\frac{\partial}{\partial t}\vec{J}_D = -\gamma_D \vec{J}_D + \epsilon_0 \omega_D^2 \vec{E}. \tag{2.37}$$

Wird der Polarisationsstrom $\vec{J}_D$ analog zu dem oben beschriebenen Ansatz in die Maxwell-Gleichungen aufgenommen, so lässt sich das Modell wie folgt in der Operatordarstellung herleiten [33]:

$$\frac{\partial}{\partial t}\vec{\psi} = \begin{bmatrix} 0 & \frac{1}{\epsilon_0 \epsilon_\infty}\nabla\times & -\frac{1}{\epsilon_0 \epsilon_\infty} \\ -\frac{1}{\mu}\nabla\times & 0 & 0 \\ \epsilon_0 \omega_D^2 & 0 & -\gamma_D \end{bmatrix} \vec{\psi}. \tag{2.38}$$

Hierbei gilt $\vec{\psi} = [\vec{E}, \vec{H}, \vec{J}_D]^T$. Im Vergleich zum Lorentz-Modell ist es beim Drude-Modell nicht nötig, die Polarisation $\vec{P}$ explizit zu berechnen, da sie für die (2.37) nicht benötigt wird.

Weitere dispersive Medien wie zum Beispiel Debye-Modelle können mit dem gleichen Ansatz behandelt werden. Hierbei werden analog zu dem Drude- beziehungsweise Lorentz-Modell zusätzliche Variablen eingeführt, um die Materialgleichungen auf ein System von Differenzialgleichungen erster Ordnung zurückzuführen. Auch der lineare Teil des Lorentz-Oszillators bei dem nichtlineaeren Zwei-Level-Modell in (2.28) kann mit diesem Ansatz berücksichtigt werden. Der nichtlineare Teil muss allerdings bei der Konstruktion des Zeitpropagationschemas speziell berücksichtigt werden, was in Kapitel 7 in den Blick genommen wird.

Ein Nachteil der ADE-Methode ist allerdings, dass durch das Einführen der zusätzlichen Variablen der Speicherbedarf des Algorithmus erhöht wird. In einem Lorentz-Medium muss zusätzlich zu jeder elektrischen Feldkomponente ein zusätzlicher Wert für die Polarisation $\vec{P}_{L,i}$ und ein Wert für den Polarisationsstrom $\vec{J}_{L,i}$ gespeichert werden. Diese Werte müssen für jeden Lorentz-Oszillator i einzeln gespeichert werden. Bei dem Drude-Modell muss zusätzlich der Polarisationsstrom $\vec{J}_D$ gespeichert werden. Der Aufwand lässt sich reduzieren, wenn sich das Lorentz- beziehungsweise das Drude-Material nicht über den gesamten Simulationsbereich erstreckt. In diesem Fall müssen die Variablen $\vec{J}$ beziehungsweise $\vec{P}$ nur für den Bereich mit dem Material gespeichert werden, was den Speicherbedarf in vielen Fällen signifikant reduzieren kann [33].

2.4 PML – Perfectly Matched Layer

In diesem Abschnitt soll die Implementierung der für die Simulation von offenen Systemen wichtigen PML beschrieben werden. Diese ermöglicht mithilfe einer absorbierenden Schicht, Wellen an den Rändern des Simulationsbereiches zu absorbieren.

Auch bei der Simulation eines offenen Systems muss die Diskretisierung des Systems auf einen finiten Bereich beschränkt werden. An den Rändern des Simulationsbereiches werden in der Regel die Feldwerte außerhalb des Simulationsbereiches als Null angenommen. Diese entspricht entweder

perfekt elektrisch leitend (PEC) oder perfekt magnetisch leitend (PMC)-Randbedingungen [4]. Dies hat zur Folge, dass Wellen, die auf die Ränder des Simulationsbereiches treffen, reflektiert werden. Dieser Umstand ist für die Simulation von offenen Systemen, wie zum Beispiel bei der Untersuchung der Streuung einer Welle an einem Objekt, höchst unerwünscht. Um zu verhindern, dass die am Streuobjekt reflektierten Wellen an den Rändern wieder auf das Objekt gestreut werden, müsste das Simulationsgebiet sehr groß gewählt werden. Dies würde zu sehr langen Simulationszeiten und hohem Speicherbedarf führen. Aus diesem Grund beschäftigt sich eine beträchtliche Anzahl von Arbeiten mit Ansätzen, dieses zu vermeiden. Eine wichtige Klasse sind hierbei die sogenannten Absorbing Boundary Conditions (ABC) [36, 37]. Diese basieren auf einer Lösung der Wellengleichung. Mit dieser Lösung werden Wellen, welche auf den Simulationsrand zulaufen, absorbiert. Vorteilhaft an diesem Ansatz ist, dass dieser lokal definiert möglich ist und dadurch die Feldwerte nur am Rand des Simulationsgebietes angepasst werden müssen [4]. Bei zwei- und dreidimensionalen Problemen gewinnt außerdem die Richtungsabhängigkeit der Absorption der Wellen an den Simulationsrändern an Bedeutung. Insbesondere ABC niedriger Ordnung zeigen hierbei Schwächen [4].

Ein anderer Ansatz zur Absorption von Wellen an den Rändern des Simulationsgebietes wird bei den PML verfolgt. Die PML sind 1994 zuerst von Berenger beschrieben worden [38]. Die Idee hierbei ist, ein absorbierendes Medium an den Rändern des Simulationsgebietes einzusetzen, welches Wellen, die in Richtung der Ränder propagieren, möglichst frequenz- und richtungsunabhängig auf einen vernachlässigbar niedrigen Wert dämpft. Diese absorbierende Schicht sollte idealerweise möglichst dünn sein, um den zusätzlichen Aufwand an Rechenzeit und Speicherbedarf gering zu halten. Um Reflexionen an der absorbierenden selber Schicht zu vermeiden, müssen diese an das benachbarte Medium angepasst sein. Die Formulierung von Berenger basiert auf einer Aufteilung aller Feldkomponenten in der PML in zwei orthogonale Komponenten (Split-Field) [4, 38].

Mittlerweile sind eine Reihe weiterer Formulierungen entwickelt worden, welche die Eigenschaften der PML im Vergleich zu der ersten Formulierung von Berenger erheblich verbessern. Diese ermöglichen eine Formulierung ohne den Split-Field Ansatz. Hierbei sind insbesondere die uniaxiale-PML (UPML) [39, 40] und die Complex Frequency Shifted (CFS)-PML zu nennen [41]. Die CFS-PML weist insbesondere günstige Eigenschaften im Hinblick auf die Absorption von evaneszenten Wellen auf [41]. Aufgrund des umfangreichen theoretischen Hintergrunds und der Vielzahl der verschiedenen Formulierungen soll die Betrachtung in dieser Arbeit auf die CFS-PML reduziert werden. Hierbei wird gezeigt, wie diese mit der oben beschriebenen ADE-Methode implementiert werden kann.

2.4.1 Formulierung

Neben den absorbierenden Eigenschaften muss das PML-Medium so formuliert werden, dass der Übergang zwischen der PML und dem umgebenden Medium reflexionsfrei ist, wie es in Abbildung 2.3 schematisch dargestellt ist [4, 42]. Dies kann zum Beispiel mit einem künstlichen anisotropen Medium oder mithilfe einer komplexen Koordinatenstreckung realisiert werden. Hierbei gibt es, wie oben diskutiert, verschiedene in Teilen äquivalente Beschreibungen. Hier wird zur Realisierung der PML die komplexe Koordinatenstreckung verwendet [4, 41, 42]. Hierzu wird mit den Koeffizienten s_x, s_y und s_z eine komplexe Koordinatenstreckung eingeführt. Mit dieser werden die partiellen Ableitungen in den Maxwell-Gleichungen wie folgt transformiert [4]:

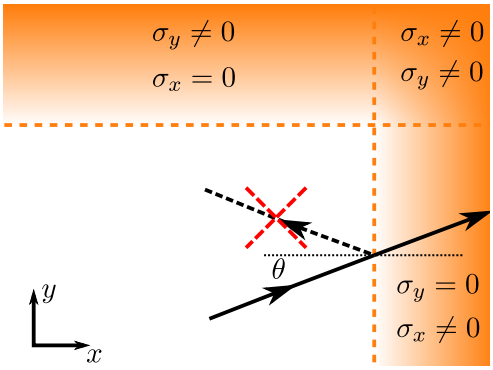


Abbildung 2.3: In der Abbildung wird eine Welle an einer PML-Grenzfläche schematisch dargestellt.

$$\frac{\partial}{\partial \tilde{x}} = \frac{1}{s_x}\frac{\partial}{\partial x}\,,\ \frac{\partial}{\partial \tilde{y}} = \frac{1}{s_y}\frac{\partial}{\partial y}\,,\ \frac{\partial}{\partial \tilde{z}} = \frac{1}{s_z}\frac{\partial}{\partial z}. \tag{2.39}$$

Die Parameter s_x, s_y und s_z werden gemäß

$$s_v = \kappa_v + \frac{\sigma_v}{\alpha_v + j\omega\epsilon_0} \tag{2.40}$$

für alle Koordinaten mit $v = x, y, z$ definiert [41]. Diese wird nun in die Maxwell-Gleichungen im Frequenzbereich eingesetzt:

$$\nabla_s \times \vec{E(\omega)} = -j\omega\vec{B}(\omega) \tag{2.41a}$$

$$\nabla_s \times \vec{H(\omega)} = j\omega\vec{D}(\omega). \tag{2.41b}$$

Hierbei gilt

$$\nabla_s = \left(\frac{1}{s_x}\frac{\partial}{\partial x}, \frac{1}{s_y}\frac{\partial}{\partial y}, \frac{1}{s_z}\frac{\partial}{\partial z}\right)^T. \tag{2.42}$$

Wird (2.41) anschließend direkt in den Zeitbereich zurücktransformiert, so ergibt sich eine Formulierung, welche die Auswertung von Faltungstermen erfordert. Für die FDTD-Methode lässt sich auf diese Weise eine Berechnungsvorschrift ableiten [41]. Allerdings ist dieser Ansatz für die untersuchten Operatorentwicklungs-Methoden ungeeignet. Eine andere Möglichkeit ist, auch an dieser Stelle auf ADEs zurückzugreifen. Indem [41]

$$\frac{1}{s_y} = \frac{1}{\kappa_y} - \frac{\sigma_y}{\kappa_y}\frac{1}{(\kappa_y(j\omega\epsilon_0 + \alpha_y) + \sigma_y)} \tag{2.43}$$

eingeführt wird, kann (2.41) umgeschrieben werden. Hierzu wird (2.43) in die partiellen Ableitungen in (2.41) eingesetzt. Für $\frac{1}{s_y}\frac{\partial}{\partial y}E_z$ gilt beispielsweise

$$\frac{1}{s_y}\frac{\partial}{\partial y}E_z = \frac{1}{\kappa_y}\frac{\partial}{\partial y}E_z + Q^E_{y,z}, \tag{2.44}$$

wobei die zusätzliche Variable $Q^E_{y,z}$ mit

$$Q^E_{y,z} = -\frac{\sigma_y}{\kappa_y}\frac{1}{(\kappa_y(j\omega\epsilon_0 + \alpha_y) + \sigma_y)}\frac{\partial}{\partial y}E_z \tag{2.45}$$

gegeben ist. Für $Q^E_{y,z}$ ergibt sich folgende zusätzliche Differentialgleichung [41]:

$$\frac{\partial}{\partial t}Q^E_{y,z} = -\frac{\kappa_y\alpha_y + \sigma_y}{\kappa_y\epsilon_0}Q^E_{y,z} - \frac{\sigma_y}{\kappa_y^2\epsilon_0}\frac{\partial}{\partial y}E_z. \tag{2.46}$$

Damit gilt exemplarisch:

$$-\mu_x x\frac{\partial}{\partial t}H_x = \frac{1}{\kappa_y}\frac{\partial}{\partial y}E_z - \frac{1}{\kappa_z}\frac{\partial}{\partial z}E_y + Q^E_{y,z} - Q^E_{z,y}. \tag{2.47}$$

Für die anderen Feldkomponenten ergeben sich die Gleichungen analog.

Wie in (2.47) erkennbar, erfordert die Einbindung der CFS-PML mithilfe von ADEs die Einführung von mehreren zusätzlichen Variablen. Im allgemeinen dreidimensionalen Fall beläuft sich diese Anzahl auf zwei Variablen pro Feldkomponente, also auf insgesamt zwölf zusätzliche Variablen Q und Gleichungen der Form (2.47). Für den zweidimensionalen Fall sind es vier zusätzliche Variablen und Gleichungen, während es im eindimensionalen Fall zwei sind. Diese zusätzlichen Terme erhöhen den Speicherbedarf, da die neu eingeführten Variablen zusätzlich zu den anderen Feldkomponenten gespeichert werden müssen. Außerdem ist der Rechenaufwand für die zusätzlichen Terme (2.47) nicht unerheblich. Allerdings muss dies nur für die Bereiche erfolgen, an denen sich die PML befindet. Die obige Formulierung erlaubt die Realisierung von sehr leistungsfähigen PML, sodass sich die Dicke dieser absorbierenden Schicht für den FDTD-Algorithmus in vielen Fällen auf nur zehn Gitterpunkte eines Yee-Gitter reduzieren lässt [41]. Außerdem ermöglicht die CFS-PML neben propagierenden Wellen auch die Dämpfung von evaneszenten Feldern. Dies ermöglicht es, in einigen Fällen die Ränder des Simulationsbereiches deutlich näher an das simulierte Bauteil zu platzieren, da kein großer Abstand für die evaneszenten Felder eingehalten werden muss [41].

2.4.2 Wahl der PML-Parameter

Wellen, welche auf die PML treffen, werden beim Durchlaufen dieser gedämpft. Für eine Dämpfungsfunktion $\sigma_x(x)$ ergibt sich für das obige Beispiel, was in Abbildung 2.3 dargestellt ist, eine totale Rest-Reflexion von

$$R(\theta) = e^{-2\sqrt{\mu/\epsilon}\cos\theta\int_0^d \sigma_x(x)dx}, \tag{2.48}$$

wobei d die Länge der PML ist. Hierbei wird die Länge d zweimal durchlaufen, da die Anteile der Wellen nach einmaligem Durchlaufen an den metallischen Randbedingungen des Simulationsgebietes reflektiert werden.

Der Übergang in das oben beschriebene PML-Medium ist nur für den analytischen Fall tatsächlich reflexionsfrei. Wenn diese zum Beispiel in einer FDTD-Simulation eingesetzt werden soll, müssen die Gleichungen sowohl örtlich als auch in der Zeit diskretisiert werden. In der diskretisierten Form geht diese Eigenschaft durch die Abweichungen von dem analytischen Zusammenhang verloren. Es können insbesondere erhebliche Reflexionen auftreten, wenn die Dämpfung $\sigma_x(x)$ sehr groß gewählt wird. Dabei sind große Dämpfungswerte gemäß (2.48) wünschenswert, da sich damit niedrigere Reflexionen mit der gleichen Länge d realisieren lassen. Um Reflexionen an der Grenzfläche zu vermeiden, ist von Berenger vorgeschlagen worden, die Dämpfung σ_x in der PML von einem geringen Wert ausgehend schrittweise zu erhöhen [38].

Praktisch haben sich für die Wahl des Funktionsverlaufes von $\sigma(x)$ polynomiale Profile bewährt [4]. Diese kann, wenn $x = 0$ der Rand der PML ist, mit

$$\sigma_x(x) = (x/d)^m \sigma_{x,\max} \tag{2.49}$$

angegeben werden. Für die Wahl des Parameters m haben sich Werte im Bereich $3 \leq m \leq 4$ als optimal für viele FDTD-Simulationen erwiesen [4]. Die maximale Dämpfung $\sigma_{x,\max}$ kann in Abhängigkeit von den übrigen Parametern wie folgt angegeben werden [4]:

$$\sigma_{x,\max} = -\frac{(m+1)\ln(R(0)}{2\sqrt{\mu/\epsilon}d}. \tag{2.50}$$

Hierbei handelt es sich bei $R(0)$ um die Reflexion an der PML, welche eingestellt werden soll. Sehr niedrige Werte führen hier zu sehr großen $\sigma_{x,\max}$, wodurch wiederum vermehrt Reflexionen am Übergang zur PML im diskretisierten System auftreten. Die Funktion für den Parameter $\kappa_x(x)$ wird zu

$$\kappa_x(x) = 1 + (\kappa_{x,\max} - 1)(x/d)^m \tag{2.51}$$

gewählt. Für $\alpha_x(x)$ wird gemäß [4]

$$\alpha_x(x) = \left(\frac{d-x}{d}\right)^{m_\alpha} \alpha_{x,\max} \tag{2.52}$$

gewählt.

2.5 Einkopplung von Wellen

In dem folgenden Abschnitt soll eine einheitliche Methode für die Einkopplung von Feldern in das Simulationsgebiet beschrieben werden.

Mit den bisher betrachteten Modellen können die Felder nur über initiale Feldverteilung $\vec{E}(\vec{r}, t = 0)$ beziehungsweise $\vec{H}(\vec{r}, t = 0)$ definiert werden. Soll zum Beispiel die Streuung eines elektromagnetischen Impulses an einem Streuobjekt untersucht werden, müsste, neben dem Streuobjekt selber, noch Platz für die örtliche Diskretisierung der initialen Feldverteilung von dem Impuls im Simulationsgebiet vorgesehen werden. Dies kann, besonders bei in Frequenzbereich schmalbandigen Impulsen, die Größe des zu diskretisierenden Bereiches stark erhöhen. Für eindimensionale Problemstellungen ist dies oft noch möglich. Werden aber zwei- oder dreidimensionale Probleme betrachtet, dann führt dies zu einem signifikanten Anstieg der Anforderungen im Hinblick auf den Speicherbedarf und die Rechenzeit.

Daher ist es wünschenswert, die Felder während der Simulation einzukoppeln. Für die FDTD-Methode sind zu diesem Zweck eine Reihe von speziellen Verfahren entwickelt worden. Für die Untersuchung von Streuung ist in diesem Zusammenhang unter anderem der sogenannte Total field scattered field (TFSF)-Ansatz wichtig [4, 42]. Eine weitere wichtige Anwendung ist die Einkopplung von Wellenleitermoden an beispielsweise dielektrischen Wellenleitern.

Viele dieser Ansätze sind allerdings speziell auf die FDTD-Methode angepasst und nicht auf die in dieser Arbeit untersuchten Algorithmen übertragbar. In [42] wird eine allgemeinere Beschreibung auf Basis des Äquivalenzprinzips vorgestellt. Bei dieser werden die einzukoppelnden Felder

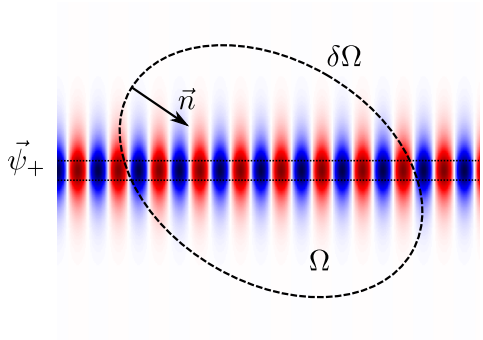


Abbildung 2.4: Die Abbildung zeigt die Feldverteilung $\vec{\psi}_+$ in dem Gebiet Ω mit der Grenze $\partial\Omega$.

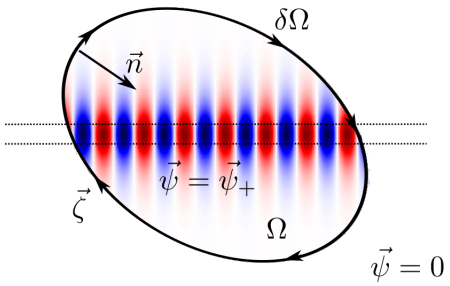


Abbildung 2.5: Hier ist das Feld mit $\vec{\psi} = 0$ außerhalb von Ω gezeigt, während innerhalb Ω $\vec{\psi} = \vec{\psi}_+$, sowie die Oberflächenströme ζ auf $\partial\Omega$.

mithilfe von Stromdichten $\vec{J}$ beziehungsweise magnetischen Stromdichten $\vec{K}$ dargestellt. Diese Darstellung erlaubt eine mit den Operatorentwicklungs-Methoden konsistente Darstellung der Einkopplung von Feldern in das Simulationsgebiet. Dieser Ansatz soll daher in den folgenden Abschnitten kurz beleuchtet werden.

2.5.1 Äquivalenzprinzip

Die Maxwell-Gleichungen können in Gegenwart von einer allgemein angesetzten Stromdichte $\vec{J}$ und einem magnetischen Strom $\vec{K}$ wie folgt umgeschrieben werden [42]:

$$\begin{bmatrix} 0 & \nabla\times \\ -\nabla\times & 0 \end{bmatrix} = \frac{\partial}{\partial t}\left(\begin{bmatrix} \epsilon_0\vec{E} \\ \mu_0\vec{H} \end{bmatrix} + \begin{bmatrix} 0 & \epsilon_0(\epsilon_r - 1) \\ \mu_0(\mu_r - 1) & 0 \end{bmatrix} * \begin{bmatrix} \vec{E} \\ \vec{H} \end{bmatrix}\right) + \begin{bmatrix} \vec{J} \\ \vec{K} \end{bmatrix}. \tag{2.53}$$

Hierbei gilt neben $\vec{\psi} = \begin{bmatrix} \vec{E} & \vec{H} \end{bmatrix}^T$ für die Stromdichten:

$$\vec{\zeta} = \begin{bmatrix} \vec{J} \\ \vec{K} \end{bmatrix}. \tag{2.54}$$

Nun soll angenommen werden, dass eine gewünschte Feldverteilung $\vec{\psi}^+$ vorliegt. Bei dieser kann es sich zum Beispiel wie im oben beschriebenen Fall um einen Impuls handeln, dessen Interaktion mit einem Streuobjekt untersucht werden soll. Eine andere Möglichkeit ist zum Beispiel eine Wellenleitermode. Dies ist in Abbildung 2.4 dargestellt. $\vec{\psi}_+$ ist die Lösung für das System ohne Streuobjekt. Die einzige Anforderung an diese Feldverteilung $\vec{\psi}_+$ ist, dass sie eine Lösung der Maxwell-Gleichungen im umgebenden Medium, welches noch als unendlich ausgedehnt angesehen werden kann, sein muss [42]. Nun wird eine Domäne Ω definiert, in deren Innerem sich das Streuobjekt befinden soll. Das Gebiet Ω ist durch die Oberfläche $\partial\Omega$ begrenzt. Dies ist in Abbildung 2.4 schematisch dargestellt. Nun soll in einem zweiten Schritt das Feld

außerhalb von Ω als Null angenommen werden:

$$\vec{\psi} = \begin{cases} \vec{\psi}_+ & \text{innerhalb } \Omega \\ 0 & \text{außerhalb } \Omega. \end{cases} \tag{2.55}$$

Nun wird (2.55) in die Gleichung (2.53) eingesetzt. Um diese lösen zu können und um die Bedingung (2.55) zu erfüllen, müssen die Ströme $\vec{\zeta}$ an der Oberfläche $\partial\Omega$ von dem Gebiet Ω eingeführt werden. Die Lösung führt auf den folgenden Ausdruck für den Oberflächenstrom [42]:

$$\vec{\zeta} = \delta(\partial\Omega) \begin{bmatrix} 0 & \vec{n}\times \\ -\vec{n}\times & 0 \end{bmatrix} \vec{\psi}_+ = \delta(\partial\Omega) \begin{bmatrix} 0 & \vec{n} \times \vec{H}_+ \\ -\vec{n} \times \vec{E}_+ & 0 \end{bmatrix} = \delta(\partial\Omega) \begin{bmatrix} \vec{n} \times \vec{H}_+ \\ -\vec{n} \times \vec{E}_+ \end{bmatrix}. \tag{2.56}$$

Bei $\delta(\partial\Omega)$ handelt es sich um eine Dirac-Delta-Funktion. Diese kommt durch die Diskontinuität an der Grenzfläche zustande [42]. Bei $\vec{n}$ handelt es sich um den Normalenvektor an der Oberfläche $\partial\Omega$, welcher auf eins normiert ist und in das Innere von Ω zeigt. Dies ist in Abbildung 2.5 schematisch dargestellt.

Bei den Größen $\vec{H}_+$ und $\vec{E}_+$ in (2.56) handelt es sich um die Feldkomponenten der vorgegebenen Verteilung $\vec{\psi}_+$. Genauer werden die Tangentialkomponenten $\vec{n} \times \vec{H}_+$ und $-\vec{n} \times \vec{E}_+$ an der Oberfläche $\partial\Omega$ benötigt. Der Wert $\delta(\partial\Omega)$ hängt im örtlich diskretisierten System von der Dicke der Grenze $\partial\Omega$ in der Richtung von $\vec{n}$ ab. Diese ist im örtlich diskretisierten System nicht infinitesimal klein, sondern hängt von der Schrittweite der Ortsdiskretisierung ab [42].

Nun kann das zu untersuchende Streuobjekt oder Ähnliches in das Gebiet Ω eingeführt werden. Außerhalb des Gebietes Ω liegt mit den Oberflächenströmen gemäß (2.56) nun nur noch das an dem Objekt gestreute Feld vor. Nun kann das noch unendlich ausgedehnte System diskretisiert werden. Um Reflexionen vom Rand des Rechengebietes zu vermeiden, kann zum Beispiel eine PML am Rand genutzt werden.

Der oben beschriebene Ansatz bildet damit die Grundlage für die TFSF-Ansätze bei FDTD-Berechnungen, für deren Formulierung auf die Literatur verwiesen sei [4, 42, 43]. Die obige Formulierung des einfallenden Feldes an der Grenze von Ω mithilfe von Oberflächenströmen erlaubt außerdem eine Übertragung auf die später untersuchten Operatorentwicklungs-Methoden.

2.5.2 Beispiel: Wellenleitermode

Die Verwendung der Methode soll anhand der Einkopplung einer Wellenleitermode illustriert werden. Hierbei soll es sich um ein dreidimensionales System handeln, wobei in dem ein Wellenleiter vorliegt, welcher parallel zur z-Achse verläuft. Entlang der x-y-Ebene an $z = 0$ soll nun eine Eigenmode eingekoppelt werden. Hierzu werden zunächst die Modenfelder des Wellenleiters benötigt. Diese können zum Beispiel mit einem numerischen Modenlöser bestimmt werden. Damit ist das Feld im Wellenleiter mit

$$\begin{bmatrix} E_{x,m} \\ E_{y,m} \\ E_{z,m} \end{bmatrix} = \mathcal{R} \left\{ \begin{bmatrix} \tilde{E}_{x,m} \\ \tilde{E}_{y,m} \\ \tilde{E}_{z,m} \end{bmatrix} \exp(j(\omega t - \beta z)) \right\} \tag{2.57}$$

und

$$\begin{bmatrix} H_{x,m} \\ H_{y,m} \\ H_{z,m} \end{bmatrix} = \mathcal{R} \left\{ \begin{bmatrix} \tilde{H}_{x,m} \\ \tilde{H}_{y,m} \\ \tilde{H}_{z,m} \end{bmatrix} \exp(j(\omega t - \beta z)) \right\} \tag{2.58}$$

gegeben, wobei mit jeweils $\tilde{E}_{x,m} \equiv \tilde{E}_{x,m}(x,y)$ die Verteilungen der einzelnen Feldkomponenten der Mode gegeben sind. Da hier nur eine Grenzfläche betrachtet wird, gilt folglich für ψ:

$$\vec{\psi} = \begin{cases} \vec{\psi}_+ = \begin{bmatrix} \vec{E}_m & \vec{H}_m \end{bmatrix}^T & z \geq 0 \\ 0 & z < 0. \end{cases} \tag{2.59}$$

Da die Wellenleitermode normal zur x-y-Ebene eingekoppelt werden soll, gilt für den Normalenvektor $\vec{n} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. Damit sind die Oberflächenströme in (2.56) für dieses Beispiel mit

$$\vec{\zeta} = \delta(\partial\Omega) = \delta(\partial\Omega) \begin{bmatrix} \vec{n} \times \vec{H}_+ \\ -\vec{n} \times \vec{E}_+ \end{bmatrix} = \delta(\partial\Omega) \begin{bmatrix} -H_{y,m} \\ H_{x,m} \\ 0 \\ E_{y,m} \\ -E_{x,m} \\ 0 \end{bmatrix} \tag{2.60}$$

gegeben. Auch wenn hier von einem normalen Einfall ausgegangen wird, können mit der diskutierten Methode auch schräg einfallende Wellenleiter implementiert werden. Der Vorteil der allgemeinen Darstellung (2.56) ist außerdem, dass beliebige Moden angeregt werden können. Während bei Wellenleitern mit nur einer ausbreitungsfähigen Mode in einigen Fällen eine Punktquelle oder ein Linienstrom ausreichen kann, um die Mode in dem Wellenleiter hinreichend gut anzuregen, ist dies bei mehrmodigen Wellenleitern nicht mehr der Fall [42]. (2.56) ermöglicht in diesem Fall auch die gezielte Anregung von höheren Moden. Die Bestimmung der Modenfelder kann entweder analytisch erfolgen oder, wie in vielen Fällen nötig, numerisch.

2.5.3 Diskussion

Neben Streuuntersuchungen kann der Ansatz auch genutzt werden, um, wie beispielhaft gezeigt, Wellenleitermoden an einer Oberfläche $\partial\Omega$, wie in Abbildung 2.4 dargestellt, anzuregen. Einschränkungen ergeben sich bei der Diskretisierung. Die numerische Dispersion hat zur Folge, dass die vorgegebenen analytischen Lösungen für das Feld ψ_+ nicht mit den numerischen Lösungen übereinstimmen. Bei dem in Abbildung 2.5 schematisch dargestellten Wellenleiter liegen dann insbesondere Fehler an der Ausgangsseite vor. Dies kann bei Streuuntersuchungen mit der TFSF-Methode für Fehler sorgen, da Teile des einfallenden Feldes in das gestreute Feld entweichen können [42]. Im Kontext der FDTD-Methode werden hierzu eine Reihe von Lösungsansätzen in der Literatur diskutiert [4, 42–45].

Bei der Untersuchung von Wellenleitern kann dies ebenfalls Probleme bereiten. Hier können ebenfalls geringe Abweichungen der Propagationskonstanten in der Simulation von der mit $\vec{\psi}_+$ vorgegebenen Mode zu Abweichungen führen. Diese Abweichungen haben zusätzliche Streuung in die Zone außerhalb von Ω zur Folge. Die Fehler nehmen mit feiner werdender Diskretisierung allerdings ab [42]. Auch hierzu werden in der Literatur verschiedene Lösungen diskutiert. Eine Möglichkeit ist, die Simulation zunächst nur mit dem Wellenleiter durchzuführen, um die numerische Wellenleitermode zu bestimmen. Der Nachteil dieses Vorgehens ist die zusätzlich nötige Simulation [4, 42]. Ein weiterer Ansatz ist die genauere Approximation der Propagation während der Simulation. Hierzu eignen sich Simulationsverfahren, welche in dieser Arbeit untersucht werden. Aufwendiger wird dies, wenn zusätzlich noch Dispersion durch die Materialien

vorliegt oder, wenn bei Betrachtung eines Wellenleiters bei breitbandigen Signalen, zusätzlich die Wellenleiterdispersion noch eine signifikante Rolle spielt [42].

3 Numerische Methoden und Implementierung

Im vorangegangenen Kapitel werden die theoretischen Grundlagen der verwendeten Modelle beleuchtet. Nun soll auf deren numerische Implementierung eingegangen werden. Dabei soll der Fokus auf der Ortsdiskretisierung liegen, während sich die restliche Arbeit mit der Approximation der Zeitpropagation beschäftigt. Dennoch bildet eine effiziente Implementierung der Ortsdiskretisierung die Grundlage für die betrachteten Zeitpropagationsverfahren.

Zunächst werden die zur Ortsdiskretisierung verwendeten Verfahren vorgestellt. Im Anschluss wird die klassische FDTD-Methode in den Blick genommen werden, welche als Vergleichsalgorithmus herangezogen wird. Zum Schluss werden einige Aspekte der Implementierung für die hier untersuchten Algorithmen beleuchtet.

3.1 Ortsdiskretisierung

Bei der Ortsdiskretisierung handelt es sich um die numerische Darstellung der Feldgrößen im Simulationsgebiet. Da sich diese Arbeit vorwiegend mit der Beschreibung der Zeitpropagation beschäftigt, soll hier nur kurz auf die verwendeten Ortsdiskretisierungverfahren eingegangen werden. Bei diesen handelt es sich einerseits um die Finite Differenzen (FD)-Diskretisierung nach Yee [5] und andererseits um pseudospektrale Methoden zur Ortsdiskretisierung [15].

3.1.1 Finite Differenzen mit dem Yee-Gitter

Das Yee-Gitter, welches zuerst im Jahre 1966 [5] von Yee beschrieben worden ist, ist ein Verfahren zur Ortsdiskretisierung der Maxwell-Gleichungen. Als Ortsdiskretisierungverfahren für den FDTD-Algorithmus ist das Yee-Gitter weit verbreitet. Allerdings findet es auch bei der numerischen Berechnung von Wellenleitermoden oder Frequenzbereichsalgorithmen Anwendung [46]. Im Folgenden soll der Ansatz kurz skizziert und auf einige wichtige Eigenschaften eingegangen werden, welche den Algorithmus für viele Anwendungen so attraktiv machen. Außerdem sollen im Anschluss die Schwächen des Ansatzes betrachtet werden. Für eine tiefergehende Diskussion sei auf die Literatur verwiesen [4, 5, 47, 48].

Eindimensionaler Fall

Um das Verfahren zu skizzieren, soll zunächst auf den eindimensionalen Fall eingegangen werden, welcher durch die Gleichungen (2.13) und (2.14) beschrieben wird. Die z-Ableitungen werden

bei dem Verfahren mit zentralen finiten Differenzen approximiert. Im Folgenden wird für die diskretisierten Feldkomponenten die Definition

$$F|_{i,j,k} = F(i\Delta x, j\Delta y, k\Delta z) \tag{3.1}$$

verwendet [5]. Es ergeben sich die folgenden örtlich diskretisierten Gleichungen:

$$\begin{aligned} \frac{\partial}{\partial t} H_y|_{k+\frac{1}{2}} &= -\frac{1}{\mu_{yy}|_{k+\frac{1}{2}}} \frac{E_x|_{k+1} - E_x|_k}{\Delta z} \\ \frac{\partial}{\partial t} E_x|_k &= -\frac{1}{\epsilon_{xx}|_k} \frac{H_y|_{k+\frac{1}{2}} - H_y|_{k-\frac{1}{2}}}{\Delta z}. \end{aligned} \tag{3.2}$$

Hierbei ist zu beachten, dass die Feldkomponenten einer Diskretisierungszelle nicht für den gleichen Ort definiert sind. Die E_x und die H_y Komponente ist um einen halben Diskretisierungsschritt $\Delta z/2$ versetzt.

Zweidimensionaler Fall

Die Ortsdiskretisierung des zweidimensionalen Falles nimmt für die TM-Mode die Form

$$\begin{aligned} \frac{\partial}{\partial t} H_x|_{i,j+\frac{1}{2}} &= -\frac{1}{\mu_{xx}|_{i,j+\frac{1}{2}}} \frac{E_z|_{i,j+1} - E_z|_{i,j}}{\Delta y} \\ \frac{\partial}{\partial t} H_y|_{i+\frac{1}{2},j} &= \frac{1}{\mu_{yy}|_{i+\frac{1}{2},j}} \frac{E_z|_{i+1,j} - E_z|_{i,j}}{\Delta x} \\ \frac{\partial}{\partial t} E_z|_{i,j} &= \frac{1}{\epsilon_{zz}|_{i,j}} \left(\frac{H_y|_{i+\frac{1}{2},j} - H_y|_{i-\frac{1}{2},j}}{\Delta x} - \frac{H_x|_{i,j+\frac{1}{2}} - H_x|_{i,j-\frac{1}{2}}}{\Delta y} \right) \end{aligned} \tag{3.3}$$

an und ist in Abbildung 3.1 dargestellt. Die Feldkomponente E_z ist hierbei im Zentrum der Yee-Zelle positioniert. Für die TE-Mode lassen sich mit

$$\begin{aligned} \frac{\partial}{\partial t} H_z|_{i,j} &= -\frac{1}{\mu_{zz}|_{i,j}} \left(\frac{E_y|_{i+\frac{1}{2},j} - E_y|_{i-\frac{1}{2},j}}{\Delta x} - \frac{E_x|_{i,j+\frac{1}{2}} - E_x|_{i,j-\frac{1}{2}}}{\Delta y} \right) \\ \frac{\partial}{\partial t} E_x|_{i,j-\frac{1}{2}} &= \frac{1}{\epsilon_{xx}|_{i,j-\frac{1}{2}}} \frac{H_z|_{i,j} - H_z|_{i,j-1}}{\Delta y} \\ \frac{\partial}{\partial t} E_y|_{i-\frac{1}{2},j} &= -\frac{1}{\epsilon_{yy}|_{i-\frac{1}{2},j}} \frac{H_z|_{i,j} - H_z|_{i-1,j}}{\Delta x} \end{aligned} \tag{3.4}$$

die ortsdiskretisierten Gleichungen angeben. Hierbei befindet sich nun die H_z Komponente im Zentrum der Gitterzelle. Der Aufbau der Gitterzellen für die beiden Polarisationen ist in Abbildung 3.1 dargestellt.

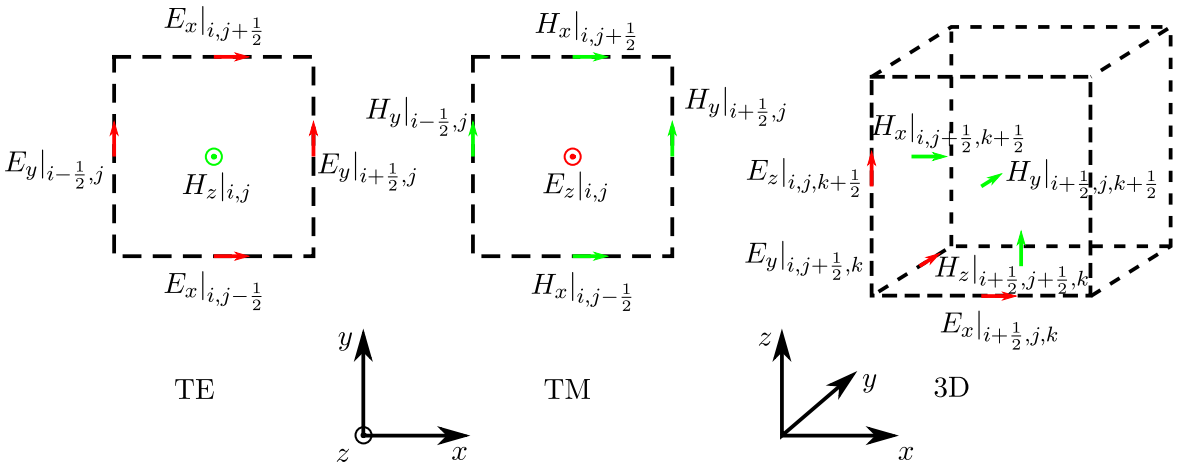


Abbildung 3.1: Die Abbildung zeigt die Yee-Gitterzellen für die zweidimensionale TM- und die TE-Mode sowie die dreidimensionale Yee-Zelle.

Dreidimensionaler Fall

Die Ortsdiskretisierung für den dreidimensionalen Fall hat die Form

$$
\begin{aligned}
\frac{\partial}{\partial t}H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}} &= -\frac{1}{\mu_{xx}|_{i,j+\frac{1}{2},k+\frac{1}{2}}}\left(\frac{E_z|_{i,j+1,k+\frac{1}{2}}-E_z|_{i,j,k+\frac{1}{2}}}{\Delta y}-\frac{E_y|_{i,j+\frac{1}{2},k+1}-E_y|_{i,j+\frac{1}{2},k}}{\Delta z}\right)\\
\frac{\partial}{\partial t}H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}} &= -\frac{1}{\mu_{yy}|_{i+\frac{1}{2},j,k+\frac{1}{2}}}\left(\frac{E_x|_{i+\frac{1}{2},j,k+1}-E_x|_{i+\frac{1}{2},j,k}}{\Delta z}-\frac{E_z|_{i+1,j,k+\frac{1}{2}}-E_z|_{i,j,k+\frac{1}{2}}}{\Delta x}\right)\\
\frac{\partial}{\partial t}H_z|_{i+\frac{1}{2},j+\frac{1}{2},k} &= -\frac{1}{\mu_{zz}|_{i+\frac{1}{2},j+\frac{1}{2},k}}\left(\frac{E_y|_{i+1,j+\frac{1}{2},k}-E_y|_{i,j+\frac{1}{2},k}}{\Delta x}-\frac{E_x|_{i+\frac{1}{2},j+1,k}-E_x|_{i+\frac{1}{2},j,k}}{\Delta y}\right)\\
\frac{\partial}{\partial t}E_x|_{i+\frac{1}{2},j,k} &= \frac{1}{\epsilon_{xx}|_{i+\frac{1}{2},j,k}}\left(\frac{H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}-H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}}{\Delta y}-\frac{H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}-H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\Delta z}\right)\\
\frac{\partial}{\partial t}E_y|_{i,j+\frac{1}{2},k} &= \frac{1}{\epsilon_{yy}|_{i,j+\frac{1}{2},k}}\left(\frac{H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}-H_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}}{\Delta z}-\frac{H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}-H_z|_{i-\frac{1}{2},j+\frac{1}{2},k}}{\Delta x}\right)\\
\frac{\partial}{\partial t}E_z|_{i,j,k+\frac{1}{2}} &= \frac{1}{\epsilon_{zz}|_{i,j,k+\frac{1}{2}}}\left(\frac{H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}-H_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}}{\Delta x}-\frac{H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}-H_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}}{\Delta y}\right).
\end{aligned}
\tag{3.5}
$$

Die Gitterzelle ist in Abbildung 3.1 dargestellt. Der örtliche Versatz zwischen den Diskretisierungspunkten der einzelnen Feldkomponenten hat die folgenden Konsequenzen. Eine der wichtigsten Eigenschaften ist die inhärente Divergenz-Freiheit des Diskretisierungsverfahrens. Die Divergenz-Freiheit in (2.1) also $\nabla \cdot (\epsilon\vec{E}) = 0$ und $\nabla \cdot (\mu\vec{H}) = 0$ ist in Abwesenheit einer Raumladung $\vec{\rho}$ durch das Yee-Gitter erfüllt [4]. Das hat zur Folge, dass die Divergenz-Freiheit nur zum Beginn der Simulation bei der Initialisierung der Feldwerte berücksichtigt werden muss, da das Yee-Gitter diese erhält [48]. Eine weitere Konsequenz des versetzten Gitters ist,

dass die ϵ und μ mit ihren Feldkomponenten ko-allokiert sind. In (3.2) führt dies dazu, dass ϵ_{xx} und μ_{yy} auch um $\Delta z/2$ versetzt definiert sind. Im zwei- und dreidimensionalen Fall, bei dem noch weitere Feldkomponenten vorhanden sind, ist für jede Feldkomponente das ϵ und μ separat zu definieren. In dieser Arbeit soll das durch die Indizes ϵ_{xx}, beziehungsweise ϵ_{yy} und ϵ_{zz} berücksichtigt werden. Der Versatz muss neben der Diskretisierung der Materialien auch bei der Einkopplung von Wellen in das Simulationsgebiet beachtet werden. Der Versatz führt zu Phasenunterschieden, welche mit in Betracht gezogen werden müssen. Ein weiterer Vorteil ist die korrekte Beschreibung der Stetigkeitsbedingungen für die Feldgrößen an Sprungstellen von Materialien [48]. Die Beziehungen für die Stetigkeit an Sprungstellen von ϵ und μ werden automatisch erfüllt, sofern das Gitter an die Geometrie der Sprungstelle angepasst ist [4]. Allerdings rufen die Approximationsfehler bei der Diskretisierung auch die sogenannte numerische Dispersion hervor [4]. Diese führt dazu, dass sich die Wellen auf dem numerischen Gitter langsamer ausbreiten als in der Realität. Dieser Effekt ist eine zentrale Einschränkung in vielen praktischen Anwendungsfällen, weshalb dieser im nächsten Abschnitt genauer beleuchtet werden soll.

Numerische Dispersion

Die Dispersionsrelation in einem isotropen Medium mit der Permittivität $\epsilon = \epsilon_0\epsilon_r$ und der Permeabilität $\mu = \mu_0\mu_r$ ist mit

$$\left(\frac{\omega}{c}\right)^2 = k_x^2 + k_y^2 + k_z^2 \tag{3.6}$$

gegeben, wobei $c = 1/\sqrt{\epsilon\mu} = c_0/\sqrt{\epsilon_r\mu_r}$ gilt. Durch die Approximationsfehler, welche bei der FD-Diskretisierung auf dem Yee-Gitter auftreten, liegt eine andere Dispersionsrelation vor [4]:

$$\left(\frac{\omega}{\tilde{c}}\right)^2 = \left(\frac{2}{\Delta x}\sin\left(\frac{k_x\Delta x}{2}\right)\right)^2 + \left(\frac{2}{\Delta y}\sin\left(\frac{k_y\Delta y}{2}\right)\right)^2 + \left(\frac{2}{\Delta z}\sin\left(\frac{k_z\Delta z}{2}\right)\right)^2. \tag{3.7}$$

Hier wird c mit $\tilde{c}$ ersetzt, da die Ausbreitungsgeschwindigkeit auf dem numerischen Gitter von c abweicht. Diese zusätzliche unphysikalische Abweichung wird numerische Dispersion genannt [4]. Problematisch an dieser Abweichung ist, dass die Dispersionsrelation (3.7) durch die rechteckige Struktur des Gitters anisotrop ist. Das bedeutet, dass die numerische Ausbreitungsgeschwindigkeit $\tilde{c}$ richtungsabhängig ist. Die gesamte Abweichung lässt sich durch eine feinere Diskretisierung reduzieren, allerdings verbleibt die Anisotropie der Dispersionsrelation [4].

3.1.2 Pseudospektraler Ansatz

Im letzten Abschnitt wird die Ortsdiskretisierung der Maxwell-Gleichungen mithilfe des Yee-Gitters beleuchtet. Eine der Schwächen dieses Ansatzes ist die im Anschluss analysierte numerische Dispersion. Diese kann durch eine feinere Auflösung reduziert werden. Wenn aber im Vergleich zum betrachteten Wellenlängenbereich sehr große Strukturen untersucht werden, führt dies auf eine sehr große Anzahl von Diskretisierungspunkten. Eine Alternative ist die Verwendung von pseudospektralen Methoden [4]. Bei diesen Pseudospectral Time-Domain (PSTD)-Ansätzen werden die Ortsableitungen mithilfe von trigonometrischen Funktionen oder mit Tschebyscheff-Polynomen approximiert. Im Kontext der Maxwell-Gleichungen sind diese unter anderem von

Liu untersucht worden [15, 49]. Diese ermöglichen bei ausreichend glatten Lösungen eine spektrale Konvergenz. Dies bedeutet, dass der Fehler bei der Approximation der Ortsableitungen exponentiell mit der Anzahl der Diskretisierungspunkte geringer wird [4, 14]. Praktisch lässt sich der anisotrope numerische Dispersionfehler, welcher bei dem FD-Yee-Gitter ein Problem darstellt, bei homogenen Medien für $N_\lambda = \lambda/\Delta >= 2$ auf null reduzieren. $N_\lambda >= 2$ ist hierbei die Anzahl der Abtastpunkte pro Wellenlänge [4]. Liegen aber mehrere verschiedene Domänen vor, so müssen diese über geeignete Übergangsbedingungen miteinander gekoppelt werden [4, 42].

Darüber hinaus macht der geringere numerische Dispersionsfehler das Verfahren für die Simulation von nichtlinearen Effekten interessant [50, 51]. Aufgrund der Konvergenzeigenschaften und der potenziellen Anwendungsmöglichkeiten bei der Untersuchung von nichtlinearen Effekten sowie großen Systemen mit niedrigen Brechzahlkontrasten ist dieses Diskretisierungsverfahrens eine interessante Alternative zum Yee-Gitter im Kontext der untersuchten Methoden. Daher soll der Ansatz im Folgendem kurz erläutert werden.

Approximation der Ortsableitungen

Für die Approximation der Ortsableitungen wird bei den pseudospektralen Verfahren genutzt, dass für die Ableitung einer Funktion $f(x)$

$$\frac{\partial}{\partial u} f(u) = \mathcal{F}^{-1}(j\omega_u \mathcal{F}(f(u))) \tag{3.8}$$

gilt, wobei $\mathcal{F}$ die Fouriertransformation und $\mathcal{F}^{-1}$ die Rücktransformation ist. Für das diskrete System wird zu diesem Zweck die Diskrete Fourier Transformation (DFT) beziehungsweise die inverse DFT verwendet. Hierzu soll ein Rechengebiet der Länge L_z angenommen werden, welches äquidistant mit N_z Punkten abgetastet ist. Dies entspricht einer Schrittweite von $\Delta z = L_z/N_z$. Für das diskrete System kann die Ortsableitung der Feldkomponenten analog zu (3.8) bestimmt werden. Diese ist mit

$$\frac{\partial}{\partial z} E_x = \frac{2\pi}{N_z \Delta z} \mathcal{F}_D^{-1}(j n_z \mathcal{F}_D(E_x)_{n_z}) \tag{3.9}$$

gegeben [4, 42]. Bei n_z handelt es sich jeweils um die Indizes der Fourierreihen-Entwicklung der Feldkomponente, die bei der DFT zugrunde liegt:

$$E_x(x) = \frac{1}{N_z} \sum_{n_z=-N_z/2}^{N_z/2-1} \tilde{E}_{x,n_z} \exp(j2\pi n_z (z - z_0)/N_z \Delta_z). \tag{3.10}$$

Bei $\tilde{E}_{x,n_z}$ handelt es sich um den n_z-ten Entwicklungskoeffizienten der Fourierreihe. Dabei ist zu beachten, dass die Funktionen hierbei als periodisch angenommen werden. Die schnelle Fourier Transformation (FFT) erlaubt eine effiziente Berechnung der DFT $\mathcal{F}_D$ beziehungsweise der inversen DFT $\mathcal{F}_D^{-1}$. Der Rechenaufwand für eine Transformation beträgt hierbei $\mathcal{O}(N \log_2 N)$. Im Vergleich mit dem FD-Yee-Gitter ist zu beachten, dass die Feldkomponenten nicht mehr örtlich versetzt, sondern alle für den gleichen Punkt definiert sind. Im allgemeinen dreidimensionalen Fall lassen sich die Feldkomponenten daher mit der Definition (3.1) angeben:

$$\begin{aligned} \vec{E}(i,j,k) &= \vec{E}|_{(i+\frac{1}{2})\Delta x,(j+\frac{1}{2})\Delta y,(k+\frac{1}{2})\Delta z} \\ \vec{H}(i,j,k) &= \vec{H}|_{(i+\frac{1}{2})\Delta x,(j+\frac{1}{2})\Delta y,(k+\frac{1}{2})\Delta z}. \end{aligned} \tag{3.11}$$

Für zwei- und dreidimensionale Probleme werden die Ortsableitungen in der folgenden Form realisiert:

$$\frac{\partial}{\partial z}E_x(:, j_0, k_0) = \frac{2\pi}{N_x \Delta x}\mathcal{F}^{-1}(jn_x\mathcal{F}(E_x(:, j_0, k_0))). \tag{3.12}$$

Hierbei bedeutet : in (3.12), dass alle Koordinatenpunkte in x-Richtung, daher über alle Indizes i, mit Fourier transformiert werden. Zu beachten ist, dass in (3.12) keine klassische zweidimensionale oder dreidimensionale DFT beziehungsweise FFT durchgeführt wird. Die Transformation wird nur entlang der x-Richtung ausgeführt [4]. Dadurch, dass die tangentialen Feldkomponenten in den Maxwell-Gleichungen immer stetig sind, ist es auch möglich (3.12), bei Materialien mit Sprüngen in der Permittivität beziehungsweise der Permeabilität zu verwenden. Allerdings kann bei metallischen Grenzflächen oder bei Übergängen mit einem sehr hohen Brechzahlkontrast eine Reduzierung der Konvergenzordnung auftreten, sodass eine dichtere Diskretisierung nötig wird [4].

Die Verwendung der Fourierreihe und die periodische Natur der trigonometrischen Funktion in (3.10) resultiert in dem sogenannten "Wrap-Around-Effekt"[4]. Dies bedeutet, dass Wellen, welche während der Zeitpropagation auf die Ränder zulaufen, an der anderen Seite des Rechengebietes wieder auftreten. Sofern das modellierte System selber nicht periodisch ist, ist dieser Effekt unerwünscht. Als mögliche Lösung wird in der Literatur die Verwendung von PML zur Absorption der Wellen an den Rändern vorgeschlagen [15]. Alternativ können neben den trigonometrischen Funktionen auch Tschebyscheff-Polynome verwendet werden. Die Transformation kann bei diesen effizient mithilfe der diskreten Kosinus Transformation erfolgen [4].

Zusammenfassung

Das oben beschriebene Verfahren eignet sich am besten, wenn sich die Medien im Rechengebiet nur kontinuierlich ändern [4, 14]. Durch die spektrale Konvergenz der Approximation kann bei homogenen Materialien eine grobe Diskretisierung des Rechengebietes von bis zu $N_\lambda >= 2$ ausreichen. Dies erlaubt die Betrachtung von sehr großen Strukturen. Außerdem liegt keine Anisotropie bei den numerischen Ausbreitungsgeschwindigkeiten vor, wie bei dem Yee-Gitter in Abschnitt 3.1.1 der Fall ist [4]. Wenn sich im Rechengebiet Diskontinuitäten wie zum Beispiel Sprünge in der Brechzahlverteilung befinden, ist der Ansatz zwar anwendbar, es kommt aber zu einer Reduktion der Konvergenzordnung [4, 14]. Daher werden in der Praxis mehr Diskretisierungspunkte pro Wellenlänge benötigt. Bei sehr großen Brechzahlsprüngen oder sogar metallischen Übergängen kann dies unpraktikabel werden. In diesem Fall bieten sich die Aufteilung des Rechengebietes und die Verwendung von lokalen Fourier-Basisfunktionen in den einzelnen Gebieten an. Das Rechengebiet wird hierbei entlang der oben beschriebenen Diskontinuitäten in einzelne Bereiche eingeteilt. Die Verbindung zwischen den Domänen wird mit speziellen Übergangsbedingungen realisiert [4, 42].

3.2 FDTD – Finite Differenzen im Zeitbereich

Der im Jahr 1966 von Yee in [5] beschriebene FDTD-Algorithmus wird auch heute noch in vielen Bereichen für die Zeitbereichslösung der Maxwell-Gleichungen eingesetzt. Der Algorithmus verwendet das bereits beschriebene Yee-Gitter zur Ortsdiskretisierung der Maxwell-Gleichungen

und verwendet ein Leap-Frog-Schema für die Zeitableitung, welches ebenfalls auf finiten Differenzen beruht [4]. Die bestehende Popularität dieses Ansatzes lässt sich unter anderem auf seine einfache Struktur und dadurch gute Erweiterbarkeit zurückführen. Diese erlaubt zum einem eine einfache Implementierung und die direkte Einbindung sowohl von linearen als auch nichtlinearen Materialmodellen [4]. Zum anderen eignet sich die Struktur des Algorithmus gut für die Parallelisierung [4]. Der FDTD-Algorithmus soll als Standardverfahren für den Vergleich mit den in dieser Arbeit vorgestellten alternativen Verfahren verwendet werden. Im Folgenden soll kurz auf den Algorithmus eingegangen werden. Eine umfangreiche Diskussion ist beispielsweise in [4] zu finden.

3.2.1 Zeitpropagation

Mit der Betrachtung des Yee-Gitters in Abschnitt 3.1.1 ist die örtliche Diskretisierung des FDTD Algorithmus bereits gegeben. Für die Approximation der Zeitableitung werden ebenfalls zentrale Differenzen herangezogen. Wie bei der Ortsdiskretisierung mit dem Yee-Gitter, werden für die Approximation der Zeitableitungen bei dem FDTD-Algorithmus das elektrische Feld $\vec{E}$ und das magnetische Feld $\vec{H}$ auch zeitlich um Δt versetzt definiert [4]. Der Algorithmus nutzt das folgende Schema zur Zeitpropagation [5]:

$$\begin{aligned} \vec{H}|^{t+\Delta t/2} &= \vec{H}|^{t-\Delta t/2} - \frac{\Delta t}{\mu}\vec{\nabla} \times \vec{E}|^{t} \\ \vec{E}|^{t+\Delta t} &= \vec{E}|^{t} + \frac{\Delta t}{\epsilon}\vec{\nabla} \times \vec{H}|^{t+\Delta t/2}. \end{aligned} \tag{3.13}$$

Die Ortsdifferenzen, welche in Abschnitt 3.1 beschrieben werden, sind hier für eine übersichtliche Notation nicht dargestellt. Der Versatz in der Zeit führt zu Phasendifferenzen zwischen den Feldwerten $\vec{E}$ und $\vec{H}$, welche berücksichtigt werden müssen. Diese spielen insbesondere bei der Definition von initialen Feldverteilungen und bei der Einkopplung von Wellen in die Simulation eine Rolle. Darüber hinaus müssen bei der Einbindung von zusätzlichen Materialmodellen, welche in Abschnitt 2.2 beschrieben sind, sowohl der Versatz im Ort als auch der Versatz in der Zeit der einzelnen Feldkomponenten beachtet werden. Außerdem muss dieser bei der Interpretation der berechneten Ergebnisse berücksichtigt werden. In (3.13) ist zu erkennen, dass die Differenzen-Gleichungen in (3.13) immer nur von bekannten Feldwerten abhängen. Der Algorithmus ist also explizit. Allerdings ist der Algorithmus nicht unbegrenzt stabil, was bei einer zu großen Wahl der Zeitschrittweite Δt zu divergierenden Ergebnissen führt. Die Stabilität soll daher im nächsten Abschnitt analysiert werden.

3.2.2 Stabilität und Numerische Dipersion

Die Stabilität und die numerische Dispersion des FDTD-Algorithmus kann untersucht werden, indem alle Felder nach ebenen Wellen entwickelt werden [4]. Dieser Ansatz wird in die diskretisierten Gleichungen (3.13) eingesetzt. Für den dreidimensionalen Fall und unter Annahme eines homogenen Mediums ergibt sich der folgende Zusammenhang für die numerische Dispersionsrelation [4]:

$$\left(\frac{\sin(\tilde{\omega}\Delta t/2)}{c\Delta t}\right)^2 = \left(\frac{\sin(\tilde{k}_x\Delta x/2)}{\Delta x}\right)^2 + \left(\frac{\sin(\tilde{k}_y\Delta y/2)}{\Delta y}\right)^2 + \left(\frac{\sin(\tilde{k}_z\Delta z/2)}{\Delta z}\right)^2. \tag{3.14}$$

Hierbei ist zu beachten, dass es sich bei $\tilde{\omega}$ um die numerische Kreisfrequenz handelt, welche von der physikalischen abweichen kann. Ebenso stellen $\tilde{k}_x$, $\tilde{k}_y$ und $\tilde{k}_z$ die numerische Wellenzahl dar [4]. Die Lichtgeschwindigkeit c in dem betrachteten homogenen Medium ist mit $c = 1/\sqrt{\epsilon\mu}$ gegeben. Im Vergleich zu der Dispersionsrelation des reinen Yee-Gitters in (3.7) liegt in (3.14) zusätzlich noch eine Abhängigkeit bezüglich des Zeitschritts vor. Durch die Zeitdiskretisierung werden zusätzliche Fehler bei der Ausbreitungsgeschwindigkeit der Wellen auf dem numerischen Gitter verursacht. Im Gegensatz zu der numerischen Dispersion, welche durch das Gitter zustande kommt, ist der Fehler, welcher durch die Zeitdiskretisierung vorliegt, isotrop, also nicht von der Ausbreitungsrichtung abhängig [4, 52].

Um die Stabilität des Verfahrens zu untersuchen, wird (3.14) nach der Frequenz ω umgestellt [4]:

$$\tilde{\omega} = \frac{2}{\Delta t} \arcsin\left(\Delta t c \sqrt{\frac{1}{\Delta x^2}\sin^2\left(\frac{\tilde{k}_x \Delta x}{2}\right) + \frac{1}{\Delta y^2}\sin^2\left(\frac{\tilde{k}_y \Delta y}{2}\right) + \frac{1}{\Delta z^2}\sin^2\left(\frac{\tilde{k}_z \Delta z}{2}\right)}\right). \quad (3.15)$$

Damit das Verfahren stabil ist, muss $\tilde{\omega}$ rein reell sein. Ist dies für einige Frequenzen nicht der Fall, so kommt es bei diesen Frequenzen zu unphysikalisch gedämpften oder auch ansteigenden Lösungen. Dieses führt zu instabilen Simulationen. Damit die Bedingung $Im(\tilde{\omega}) = 0$ erfüllt ist, muss das gesamte Argument der arcsin-Funktion in (3.15) kleiner als eins sein. Die obere Grenze für die $\sin^2$-Funktionen ist bei allen möglichen reellen Werten für $\tilde{k}$ eins [4]. Damit hängt die Bedingung für das Argument der arcsin-Funktion für gegebene Materialeigenschaften und Diskretisierung nur von Δt ab. Die Bedingung ist erfüllt, wenn

$$\Delta t \leq \Delta t_{\mathrm{CFL}} = \frac{1}{c\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}}} \quad (3.16)$$

gilt [4]. Bei (3.16) handelt es sich um die eingangs bereits erwähnte CFL-Bedingung. Ist dies nicht gegeben, kann die Frequenz $\tilde{\omega}$ auf dem Gitter einen Imaginärteil $Im(\tilde{\omega}) \neq 0$ aufweisen, was zu exponentiell steigenden oder fallenden Lösungen führt. Die Bedingung (3.16) koppelt die maximale stabile Zeitschrittweite Δt_{CFL} an die Schrittweite der Ortsdiskretisierung des Simulationsgebietes [4]. Dies hat zur Folge, dass für kleine Schrittweiten in der Ortsdiskretisierung sehr kleine Zeitschrittweiten gewählt werden müssen, um die Stabilität der Simulation zu gewährleisten. Für den zweidimensionalen Fall kann die CFL-Bedingung mit

$$\Delta t \leq \Delta t_{\mathrm{CFL}} = \frac{1}{c\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}}} \quad (3.17)$$

angegeben werden. Im eindimensionalen Fall ist sie durch

$$\Delta t \leq \Delta t_{\mathrm{CFL}} = \frac{1}{c\sqrt{\frac{1}{\Delta z^2}}} \quad (3.18)$$

gegeben.

3.3 Implementierungsaspekte

In diesem Abschnitt soll auf die Implementierung der oben beschriebenen Materialmodelle und Verfahren zur örtlichen Diskretisierung für die hier untersuchten Algorithmen zur Zeitpropagation

eingegangen werden. Diese Algorithmen basieren formal auf der Matrixform (2.17) der Maxwell-Gleichungen. Daher soll zunächst auf die Diskretisierung der Maxwell-Gleichungen in (2.17) eingegangen werden sowie auf einige der grundlegenden Eigenschaften der dabei bestimmten Systemmatrix. Im Anschluss sollen einige Aspekte für die parallele Implementierung auf CPUs und GPUs in den Blick genommen werden.

3.3.1 Diskretisierung der Systemmatrix

Zuerst soll auf die Diskretisierung mit einem FD-Gitter nach Yee eingegangen werden. Aus der Feldverteilung $\vec{\psi}(t, \vec{r})$ in (2.17) wird durch die Diskretisierung ein großer vollbesetzter Vektor $\vec{\Psi}(t) \in \mathbb{R}^N$, welcher die diskreten Feldkomponenten an allen Abtastpunkten enthält. Die ADEs, welche bei der Beschreibung von Materialmodellen oder für die Einbindung von PMLs Anwendung finden, werden ebenfalls auf diese Weise diskretisiert. Hierbei ist bei der Verwendung des Yee-Gitters der örtliche Versatz der Feldkomponenten zu beachten. Ein Versatz in der Zeit, wie bei dem FDTD-Algorithmus [4], liegt nicht vor. Aus dem Matrixoperator in (2.17) wird hierbei also eine große dünnbesetzte Matrix $\mathcal{H} \in \mathbb{R}^{N \times N}$. Das örtlich diskretisierte System ist mit

$$\frac{\partial}{\partial t}\vec{\Psi}(t) = \mathcal{H}\vec{\Psi}(t) \tag{3.19}$$

gegeben. Am Rand des Systems müssen hierbei Randbedingungen für die Zellen an den Grenzen des Simulationsgebietes gewählt werden. Hier werden in der Regel mit $E_x = 0$ PEC-Randbedingungen oder mit $H_y = 0$ PMC-Randbedingungen gewählt [4]. Für periodische Systeme können auch periodische Randbedingungen verwendet werden [4]. Die Diskretisierung soll an einem eindimensionalen System (2.13) mit $N_z = 5$ Punkten illustriert werden. Die diskretisierten Versionen der Ortsableitungen (2.13) sind mit

$$D_z^E = \frac{1}{\Delta z}\begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad D_z^H = -\left(D_z^E\right)^T = \frac{1}{\Delta z}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{3.20}$$

bestimmt. Damit ist die Systemmatrix $\mathcal{H}$ mit

$$\mathcal{H} = \begin{bmatrix} 0 & -[\frac{1}{\epsilon_{xx}}]D_z^H \\ -[\frac{1}{\mu_{yy}}]D_z^E & 0 \end{bmatrix}, \tag{3.21}$$

gegeben, wobei es sich bei $[\frac{1}{\epsilon_{xx}}]$ beziehungsweise $[\frac{1}{\mu_{yy}}]$ um die diskretisierten Werte für die Permittivität und Permeabilität handelt. Diese sind mit den diskretisierten Werten von E_x beziehungsweise H_y ko-allokiert.

Für die dämpfungsfreien Maxwell-Gleichungen (2.17) kann durch eine Transformation die Schiefsymmetrie der Systemmatrix $\mathcal{H}$ explizit gezeigt werden [4, 53]. Eine reelle schiefsymmetrische Matrix hat rein imaginäre Eigenwerte, welche symmetrisch zur reellen Achse verteilt sind [54]. Die Systemmatrix $\mathcal{H}$ ist bei praktischen Problemen sehr groß. Bei der FD-Diskretisierung nach Yee liegen bei einem zweidimensionalen System mit 1000 Diskretisierungspunkten entlang jeder Achse schon insgesamt $N_x N_y = 1 \times 10^6$ Abtastpunkte für jede Feldkomponente vor, sodass der Vektor

$\vec{\Psi}(t)$ $N = 3 \times 10^6$ Elemente aufweist. Die Matrix $\mathcal{H}$ hat damit eine Größe von $6 \times 10^6 \times 6 \times 10^6$, was in insgesamt 9×10^{12} Matrixeinträgen resultiert. Bei einem dreidimensionalen System mit 100 Diskretisierungspunkten entlang jeder Achse liegen mit $N = 6N_xN_yN_z = 6 \times 10^6$ insgesamt $3{,}6 \times 10^{13}$ Matrixeinträge vor. Damit ist es auch auf leistungsfähigen Rechnern in der Regel nicht möglich, die komplette Systemmatrix für praxisrelevante Systeme überhaupt zu speichern. Die Verwendung von Materialmodellen verschärft dieses Problem noch weiter. Allerdings ist die Systemmatrix $\mathcal{H}$ von Diskretisierungsverfahren, wie der FD-Diskretisierung nach Yee [5], Verfahren wie den Finite Integrationstechnik (FIT)-Verfahren [55] oder dem diskontinuierlich Galerkin (DG)-Verfahren [17] in der Regel dünn besetzt.

Die Anwendung des pseudospektralen Diskretisierungsverfahren aus Abschnitt 3.1.2 erfolgt analog. Ein wichtiger Unterschied hierbei ist, dass die Diskretisierung der örtlichen Differenziale formal auf dicht besetzte Matrizen führen würde. Dies ist in der Verwendung der globalen Basisfunktionen begründet. Daher wird die Systemmatrix in der Praxis nicht explizit berechnet, da nach den obigen Betrachtungen eine tatsächliche Speicherung für große Systeme unpraktikabel ist. Die Ortsableitungen in der Systemmatrix (3.21) werden stattdessen mit der Vorschrift (3.9) mithilfe der FFT bestimmt. Dieses Vorgehen wird im Folgenden erneut aufgegriffen.

3.3.2 Effiziente Implementierung

Die formale örtliche Diskretisierung nach Yee führt, wie oben beschrieben, auf große dünnbesetzte Systemmatrizen $\mathcal{H}$. Daher kann für diese auf die Verwendung von speziellen Formaten für dünnbesetzte Matrizen zurückgegriffen werden [56]. Bei diesen ist es insbesondere möglich, Matrix-Vektor-Multiplikationen effizient durchzuführen. Allerdings wird bei der Verwendung eines solchen Standardformates die Struktur des örtlichen Diskretisierungsverfahrens nicht mehr ausgenutzt. Ein besserer Ansatz ist es, die Matrix-Vektor-Multiplikation $\mathcal{H}\Psi(t)$ direkt zu implementieren. Dieses Vorgehen erlaubt eine effizientere, auf die örtliche Diskretisierung angepasste, Implementierung für parallele Architekturen wie Mehrkern-CPUs und GPUs. Hierbei kann die Matrix-Vektor-Multiplikation gezielt optimiert werden. Dazu wird für viele der im Rahmen dieser Arbeit beschriebenen Modelle die Matrix-Vektor-Multiplikation mit OpenCL™ [57] implementiert, während der Rest der Programmierung in der Skriptsprache Python erfolgt.

Außerdem kann festgestellt werden, dass eine Matrix-Vektor-Multiplikation $\mathcal{H}\vec{\Psi}(t)$ im Hinblick auf den Rechenaufwand einem Zeitschritt mit der FDTD-Methode einspricht [4, 53]. Diese Parallele ist von Bedeutung, da sie erlaubt, die Matrix-Vektor-Multiplikation $\mathcal{H}\vec{\Psi}(t)$ im Laufe der Arbeit als Maß für den Rechenaufwand der untersuchten Algorithmen heranzuziehen. Die Anzahl der benötigten Matrix-Vektor-Multiplikation $\mathcal{H}\vec{\Psi}(t)$ ist hierbei ein von der Implementierung unabhängiges Maß für den benötigten Rechenaufwand. Für den pseudospektralen Ansatz werden, wie oben bereits beschrieben, die Ortsableitungen der Felder in der Matrix-Vektor-Multiplikation $\mathcal{H}\vec{\Psi}(t)$ sehr effizient mit der FFT bestimmt. Für die parallele Implementierung großer Systeme auf verteilten Rechnersystemen ergeben sich allerdings bei dem in Abschnitt 3.1.2 betrachteten Fourier-Ansatz Probleme durch die Verwendung der globalen Basisfunktionen. Die globalen Basisfunktionen resultieren in einem großen Datenaustausch, welcher bei der Implementierung auf verteilten Rechnersystemen zu Problemen führt [42]. Ein Lösungsansatz dafür ist der Rückgriff auf die oben bereits genannte Aufteilung des Rechengebietes und die Verwendung von lokalen Fourier-Basisfunktionen. Diese erlauben eine Aufteilung des Systems und eine parallele Berechnung der einzelnen Gebiete [4, 42, 58].

4 Unitärer Algorithmus zur Zeitpropagation

Im Folgenden wird ein erster Ansatz evaluiert, um mit Polynomentwicklungen ein alternatives Verfahren zur Zeitpropagation zu konstruieren. Das Ziel soll es hierbei sein, mithilfe einer Operatorentwicklung auf Basis von Tschebyscheff-Polynomen ein Zeitpropagationsschema zu konstruieren, welches Zeitschritte erlaubt, die größer sind als der Zeitschritt Δt_{CFL} in (3.16). Der entwickelte trigonometrische Operator basiert hierbei auf der formalen Anwendung eines Zweischrittverfahrens auf das örtlich diskretisierte System. Hierbei soll ein explizites Zeitpropagationsschema bestimmt werden.

Ein Algorithmus zur Lösung der Maxwell-Gleichungen auf Basis von Approximationen mit Tschebyscheff-Polynomen wird in [53] vorgestellt. Bei diesem Ansatz wird allerdings ein Matrixexponential mithilfe der Tschebyscheff-Polynome entwickelt. In [59] wird ein Zweischrittverfahren mithilfe einer Taylorpolynom-Approximation niedriger Ordnung zur Lösung der Schrödingergleichung verwendet. In der Mathematik ist die Klasse der Propagations-Verfahren mit trigonometrischen Operatoren zuerst von Gautschi [60] untersucht worden. Weitere Ausführungen sind in [61] zu finden. Zu Differenzialgleichungen zweiter Ordnung sind darüber hinaus Untersuchungen in [62] zu finden.

Das untersuchte Zeitpropagationsschema soll die Lösung der Maxwell-Gleichungen unter Berücksichtigung aller Feldkomponenten erlauben. Hierzu wird ein spezieller Operator mithilfe einer Transformation der Matrix bestimmt. Eine Besonderheit des vorgestellten Ansatzes gegenüber dem Tschebyscheff-Polynom-Ansatz in [53] ist, dass durch die Formulierung des Algorithmus gewährleistet wird, dass der Wachstumsfaktor immer betragsmäßig eins ist. Außerdem erfolgt eine direkte Berücksichtigung der Eigenschaften des Eigenwertspektrums der Systemmatrix $\mathcal{H}$.

Nach der Formulierung des Ansatzes wird seine Stabilität untersucht. Es wird eine Methode vorgestellt, mit der Stabilität sichergestellt werden kann. Anschließend wird dies mithilfe eines Beispieles illustriert. Abschließend wird der Fehler des Algorithmus betrachtet. Teile der vorgestellten Ansätze sind in [KS2, KS3] veröffentlicht. Die Ergebnisse werden im Folgenden dargestellt und erweitert. Voruntersuchungen, auf die nicht weiter eingegangen werden soll, erfolgten mithilfe der skalaren Wellengleichung. Diese sind in [KS4, KS5] veröffentlicht beziehungsweise liegen in [63] vor.

4.1 Theorie

Der Ausgangspunkt für die Formulierung des Algorithmus sind die örtlich diskretisierten Maxwell-Gleichungen in der Operatordarstellung in (3.19). Hierbei wird ein lineares, isotropes, dämpfungsfreies Medium angenommen. Die örtliche Diskretisierung wird mit einem FD-Yee-Gitter [5], wie in Abschnitt 3.3 diskutiert, durchgeführt. Für die Diskretisierung kommen noch andere

Verfahren wie die FIT-Methode [55] oder pseudospektrale Ansätze [15] infrage. Die formale Lösung von (3.19) ist mit

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t \mathcal{H}) \vec{\Psi}(t_n) \tag{4.1}$$

gegeben. Bei Δt handelt es sich um den Zeitschritt des Algorithmus, während es sich bei t_n um den aktuellen Zeitpunkt handelt. Das Verfahren zur Zeitpropagation wird wie folgt umformuliert:

$$\vec{\Psi}(t_n + \Delta t) = \mathcal{P}(\mathcal{H}) \vec{\Psi}(t_n) + \vec{\Psi}(t_n - \Delta t). \tag{4.2}$$

Hierbei handelt es sich bei $\vec{\Psi}(t)$ um die Feldverteilung zum Zeitpunkt $t = t_n$. Bei $\vec{\Psi}(t_n + \Delta t)$ und $\vec{\Psi}(t_n - \Delta t)$ handelt es sich wiederum um die Feldverteilungen zum Zeitpunkt $t = t_n + \Delta t$ beziehungsweise $t = t_n - \Delta t$. Der Operator $\mathcal{P}(\mathcal{H})$ ist eine Funktion der Systemmatrix $\mathcal{H}$. Durch Einsetzen von (4.1) in (4.2) kann diese mit

$$\mathcal{P}(\mathcal{H}) = 2 \sinh(\Delta t \mathcal{H}) \tag{4.3}$$

angegeben werden. Für eine ausführliche Herleitung von (4.3) sei darüber hinaus auf Anhang B.1 verwiesen. Um den Algorithmus zu realisieren, muss nun die $\mathcal{P}(\mathcal{H}) = 2 \sinh(\Delta t \mathcal{H})$ bestimmt werden. Aufgrund der Größe der diskretisierten Matrix $\mathcal{H}$, welche in Abschnitt 3.3 genauer beleuchtet wird, kommen viele klassische Verfahren zur Berechnung von Matrixfunktionen für den betrachteten Anwendungsfall nicht infrage. Dazu zählen beispielsweise Verfahren auf Basis von Padé-Approximationen der Funktion $\mathcal{P}(\mathcal{H})$ [64]. Prinzipiell ist es bei einer zeitlich konstanten Systemmatrix $\mathcal{H}$ möglich, die Matrixfunktion $\mathcal{P}(\mathcal{H})$ vor der Zeitpropagation zu berechnen. Dieses Vorgehen hätte den Vorteil, dass die Approximation nur einmal bestimmt werden muss. Dies nicht möglich, da die Matrixfunktion wie $\mathcal{P}(\mathcal{H})$ im Allgemeinen nicht dünnbesetzt ist, selbst wenn $\mathcal{H}$ dünnbesetzt ist [64]. Um die in Abschnitt 3.3 vorgestellten Eigenschaften zu nutzen, sollte der verwendete Algorithmus nur Matrix-Vektor-Multiplikationen mit $\mathcal{H}$ verwenden. Daher soll $\mathcal{P}(\mathcal{H})$ mithilfe einer Polynomapproximation berechnet werden.

Bei der Polynomapproximation soll genutzt werden, dass die Funktion $\mathcal{P}(\mathcal{H})$ nur für die Eigenwerte σ_k von $\mathcal{H}$ definiert ist [64]. Für die Polynomapproximation hat dies zwei wichtige Konsequenzen. Die erste wichtige Konsequenz ist, dass die Matrixfunktion $\mathcal{P}(\mathcal{H})$ wie eine skalare Funktion in Abhängigkeit der Eigenwerte approximiert werden kann. Die zweite Konsequenz ist, dass der Approximationsbereich auf das Eigenwertspektrum $\sigma(\mathcal{H})$ der Systemmatrix beschränkt werden kann. Dabei ist es unerheblich, welche Werte eine Approximation für Eigenwerte außerhalb dieses Bereiches annimmt [64].

Erste Eigenschaften des Eigenwertspektrums von $\mathcal{H}$ werden in Abschnitt 3.3 diskutiert. Die Eigenwerte von $\mathcal{H}$ liegen in einem Intervall $\sigma_k \in [-j\sigma_{\max}, j\sigma_{\max}]$. Hierbei gibt $\sigma_{\max}$ den betragsmäßig größten Eigenwert von $\mathcal{H}$ an. Dieser kann beispielsweise mit dem Gerschgorin-Theorem bestimmt werden [54]. Der Wert wird hierbei maßgeblich von der örtlichen Diskretisierung und der Lichtgeschwindigkeit c, welche wiederum von $\mu(\vec{r})$ und $\epsilon(\vec{r})$ abhängt, bestimmt. Das Intervall mit den diskreten Eigenwerten σ_k wird im Folgenden als kontinuierlich angenommen: $\sigma_k \to \sigma$. Um trotz der imaginären Eigenwerte eine Approximation auf einem reellen Intervall durchführen zu können, wird die Transformation $\sigma_{\mathrm{B}} = -j\sigma/\sigma_{\max}$ angewendet, welche auch in [53] beschrieben wird. Damit lässt sich (4.3) zu

$$\mathcal{P}(\sigma_{\mathrm{B}}) = 2j \sin(\Delta t \sigma_{\mathrm{B}} \sigma_{\max}) \tag{4.4}$$

umschreiben. Das transformierte Intervall ist definiert für $\sigma_B \in [-1, 1]$. Die Matrixfunktion (4.4) ist für verschiedene Zeitschrittweiten in Abbildung 4.1 dargestellt und wird nun mithilfe einer

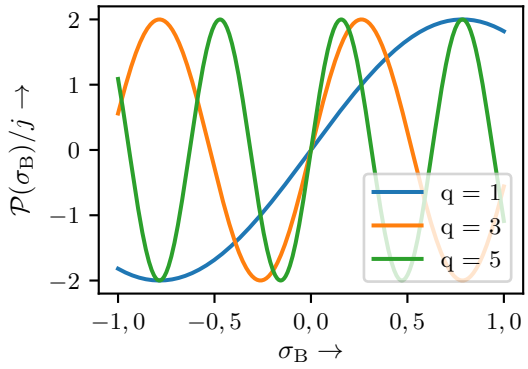


Abbildung 4.1: Die Abbildung zeigt den Verlauf der transformierten Operatorfunktion $\mathcal{P}(\sigma_\mathrm{B})$ für verschiedene normierte Zeitschrittweiten mit $\Delta t = q\Delta t_\mathrm{CFL}$.

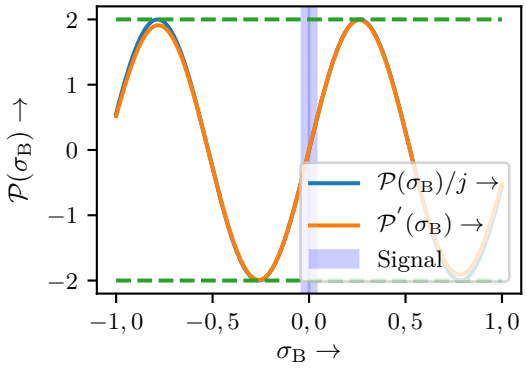


Abbildung 4.2: Die Anwendung der Fensterfunktion sowie der für die Darstellung des Signals verwendete Eigenwertbereich werden schematisch dargestellt. Die grüne Linie stellt den Stabilitätsbereich dar.

Polynomapproximation der Form approximiert:

$$\mathcal{P}(\sigma_\mathrm{B}) \approx \hat{\mathcal{P}}(\sigma_\mathrm{B}) = j \sum_{n=0}^{N_\mathrm{Pol}} c_n P_n(\sigma_\mathrm{B}). \tag{4.5}$$

Bei $\hat{\mathcal{P}}(\sigma_\mathrm{B})$ handelt es sich um die Polynom-Approximation von $\mathcal{P}(\sigma_\mathrm{B})$. Hierzu werden Tschebyscheff-Polynome verwendet. Diese bieten sich in diesem Zusammenhang an, da sie eine genaue Approximation auf einem abgeschlossenen Intervall erlauben [65]. In diesem Fall muss $\mathcal{P}(\sigma_\mathrm{B})$ auf $\sigma_B \in [-1, 1]$ approximiert werden. Außerdem müssen für die Approximation nur ungerade Polynome $P_n(\sigma_\mathrm{B})$ verwendet werden, da die Funktion $\mathcal{P}(\sigma_\mathrm{B})$ ungerade ist. Gerade Polynome liefern hierbei keinen Betrag. Die Genauigkeit der Approximation wird durch die verwendete Polynomordnung N_Pol in (4.5) bestimmt. Die Entwicklungskoeffizienten c_n in (4.5) müssen nach der Approximation noch zurücktransformiert werden. Hierzu wird in (4.5) $\sigma_\mathrm{B} = -j\sigma/\sigma\mathrm{max}$ gesetzt. Damit ist die Approximation der Funktion $\mathcal{P}(\mathcal{H})$ bestimmt.

Die Berechnung der Approximation kann mithilfe der Rekursionsbeziehung der Tschebyscheff-Polynome erfolgen [53, 65]. Dies hat den Vorteil, dass die Rekursionsbeziehung nur Matrix-Vektor-Multiplikationen mit $\mathcal{H}$ benötigt, sodass der Algorithmus vollständig explizit ist. Effektiv bedeutet dies, dass, statt mit $\mathcal{P}(\mathcal{H})$ die Matrixfunktion zu berechnen, stattdessen $\mathcal{P}(\mathcal{H})\vec{\Psi}(t)$ berechnet wird. Auf diese Weise können die oben beschriebenen Probleme vermieden werden. Während die Polynomentwicklung des Operators formal in Abhängigkeit von den Eigenwerten der Systemmatrix durchgeführt wird, ist eine Bestimmung von Eigenwerten zu keinem Zeitpunkt nötig. Da das Zeitpropagationsschema (4.2) die Feldverteilungen zu zwei Zeitpunkten, $t = t_n$ und $t = t_n + \Delta t$, verwendet, müssen Startwerte für die Feldverteilungen zu beiden Zeitpunkten gefunden werden. Das wird, sofern möglich, mit analytischen Lösungen für die Startverteilungen realisiert. Alternativ werden die Startwerte mithilfe des FDTD-Algorithmus bestimmt.

4.2 Stabilitätanalyse

Nun soll die Stabilität des Zeitpropagationsschemas untersucht werden. Dies erfolgt mit der Stabilitätsanalyse nach Von-Neumann. Anstelle von ebenen Wellen wird das Zeitpropagationsschema in (4.2) hier in Abhängigkeit von den Eigenwerten von $\mathcal{H}$ betrachtet. Die diskreten Feldverteilungen $\Psi(t)$ werden auch formal nach den Eigenfunktionen entwickelt und im Anschluss mit $\vec{v}(\sigma, t)$ bezeichnet. Die Funktionen $\vec{v}(\sigma, t)$ für die verschiedenen Zeitpunkte $t_n - \Delta t$, t_n und $t_n + \Delta t$ in (4.2) werden nun mithilfe eines Wachstumsfaktors $g(\sigma)$ dargestellt: $\vec{v}(\sigma, t_n + \Delta t) = g(\sigma)\vec{v}(\sigma, t_n)$ und in (4.2) eingesetzt. Damit kann ein Ausdruck für den Wachstumsfaktor $g(\sigma)$ bestimmt werden:

$$g(\sigma) = \hat{\mathcal{P}}(\sigma)/2 \pm \sqrt{\hat{\mathcal{P}}^2(\sigma)/4 + 1}. \tag{4.6}$$

Für eine genaue Herleitung sei auf Anhang B.2 verwiesen. Um die Stabilität des Zeitpropagationsschemas zu gewährleisten, muss $|g(\sigma)| \leq 1$ auf dem ganzen Intervall $\sigma_{\mathrm{B}} \in [-1, 1]$ gegeben sein. An dieser Stelle soll die Bedingung noch weiter verschärft werden, indem mit $|g(\sigma)| = 1$ Unitarität gefordert wird. Nun wird der Absolutbetrag $|g(\sigma)|$ in (4.6) auf dem Intervall $\sigma_k \in [-j\sigma_{\max}, j\sigma_{\max}]$ mit den Eigenwerten der Systemmatrix $\mathcal{H}$ betrachtet. Hierbei lässt sich zeigen, dass $|g(\sigma)| = 1$ auf dem ganzen Intervall gegeben ist, wenn

$$|\hat{\mathcal{P}}(\sigma)| \leq 2 \tag{4.7}$$

durch die Polynomapproximation erfüllt ist. Ist dies der Fall, ist der Algorithmus unitär. Daher geht von Schritt zu Schritt keine Energie im System verloren. Die Besonderheit hierbei ist, dass die Unitarität durch die Struktur des Algorithmus gewährleistet wird und sie unabhängig von der Approximationsordnung gegeben ist, sofern die Bedingung (4.7) erfüllt ist. Ein weiterer Punkt ist, dass keine direkte Bedingung für den Zeitschritt Δt vorliegt. Daher ist es auch möglich, Zeitschritte $\Delta t > \Delta t_{\mathrm{CFL}}$ zu verwenden. Bei der praktischen Realisierung der Approximation können Fehler, hervorgerufen durch das Abbrechen der Polynomapproximation in (4.5), dazu führen, dass die Bedingung (4.7) verletzt wird. Hierbei spielen insbesondere die Werte eine Rolle, bei denen $\mathcal{P}(\sigma_{\mathrm{B}})$ die Stabilitätsgrenze (4.7) tangiert, wie es in Abbildung 4.2 zu erkennen ist. Dies kann zu Fehlern in der Approximation und sogar zu der Instabilität des Zeitpropagationsschemas führen. Um dies zu verhindern, wird die Approximation modifiziert. Der Ansatz ist, statt die Funktion $\mathcal{P}(\sigma_{\mathrm{B}})$ direkt zu approximieren, eine modifizierte Funktion $\mathcal{P}'(\sigma_{\mathrm{B}})$ zu verwenden. Diese Funktion wird durch die Anwendung einer Fensterfunktion mit

$$\mathcal{P}'(\sigma_{\mathrm{B}}) = \rho(\sigma_{\mathrm{B}})\mathcal{P}(\sigma_{\mathrm{B}}) \tag{4.8}$$

bestimmt. Hierbei ist $\rho(\sigma_{\mathrm{B}})$ die Fensterfunktion. Ihr Effekt ist in Abbildung 4.2 dargestellt. Die Funktion $\rho(\sigma_{\mathrm{B}})$ wird so gewählt, dass die Bedingung (4.7) auch für die kritischen Eigenwerte erfüllt ist. Bei diesen handelt es sich um die Werte σ_{B}), an denen $\mathcal{P}(\sigma_{\mathrm{B}})$ die Stabilitätsgrenze (4.7) tangiert. Hier werden beispielsweise Raised-Cosine-Funktionen verwendet. Indem $\rho(\sigma_{\mathrm{B}}) = 1$ für die Eigenwerte gewählt wird, welche für die Simulation interessant sind, wird sichergestellt, dass die Fensterfunktion keine zusätzlichen Fehler bei diesen zur Folge hat. Für andere Eigenwerte muss dies nicht gegeben sein. Hierbei wird genutzt, dass die Frequenzen f der untersuchten Felder mit $f \mathrel{\widehat{=}} \pm\sigma/(2\pi)$ mit den Eigenwerten der Systemmatrix korrespondieren. Da durch die feine Diskretisierung $2\pi f << \sigma_{\max}$ gilt, liegen die für die betrachteten Felder entscheidenden Eigenwerte, bezogen auf das gesamte Eigenwertspektrum $\sigma(\mathcal{H})$, nahe dem Nullpunkt. Damit

lassen sich die Eigenwerte für das Signalspektrum, wie in Abbildung 4.2 dargestellt, im Eigenwertspektrum $\sigma(\mathcal{H})$ der Systemmatrix $\mathcal{H}$ lokalisieren. Der Vorteil von diesem Verfahren ist, dass auch bei niedrigen Approximationsordnungen die Stabilität des Verfahrens gewährleistet werden kann.

4.3 Fehlerbetrachtung

Im Folgenden sollen zunächst die Eigenschaften des unitären Zeitpropagationsschemas durch einen Vergleich mit einem Algorithmus, welcher diese unitären Eigenschaften nicht hat, illustriert werden. Der Vergleichsalgorithmus wird durch eine Approximation des Exponentialoperators (4.1) konstruiert [53]. Im Anschluss wird der Fehler des untersuchten Algorithmus in den Blick genommen. Für den Vergleich wird die Propagation eines gaußförmigen Impulses in einem eindimensionalen System betrachtet. Hierbei wird für beide Algorithmen eine Schrittweite von $\Delta z = 10\,\mathrm{nm}$ für die örtliche Diskretisierung nach dem FD-Yee-Gitter verwendet. Mit $\Delta t = 10\Delta t_{\mathrm{CFL}} = 0{,}334\,\mathrm{fs}$ wird eine zeitliche Schrittweite eingesetzt, welche zehnmal größer ist als es die CFL-Bedingung für den FDTD-Algorithmus erlaubt. Sowohl der hier vorgestellte Algorithmus als auch der Vergleichsalgorithmus, basierend auf dem Exponentialoperator [53], greifen zu Tschebyscheff-Polynomen für die Entwicklung der Operatoren. Die Entwicklung wird für beide Fälle bis zu einem Grad von $N_{\mathrm{Pol}} = 25$ durchgeführt. Diese Entwicklungsordnung ist bei dem verwendeten Zeitschritt verhältnismäßig niedrig. Der Fehler bei der Polynomentwicklung ist also noch nicht auf einen vernachlässigbar kleinen Wert abgefallen. Die Konsequenzen hiervon werden in den folgenden Betrachtungen deutlich.

Für den untersuchten Algorithmus wird der Fensterfunktionsansatz genutzt, um die Stabilität des Zeitpropagationsschemas zu gewährleisten. Es wird eine Raised-Cosine-Funktion eingesetzt, welche um das Zentrum $\sigma_{\mathrm{B}} = 0$ des Eigenwertspektrums zentriert wird. Die Funktion wird so parametrisiert, dass für $\sigma_B \in [-0,026\,0,026]$ $\rho(\sigma_{\mathrm{B}}) = 1$ erfüllt ist. Bei $|\sigma_B| \geq 3,930$ gilt für die Fensterfunktion $\rho(\sigma\mathrm{B}) = 0$. Zuerst werden die Wachstumsfaktoren $g(\sigma_{\mathrm{B}})$ der Algorithmen in den Blick genommen. Diese werden in Abhängigkeit von den normalisierten Eigenwerten σ_{B} untersucht. Diese sind für beide Algorithmen in Abbildung 4.3 dargestellt. Bei dieser fällt auf, dass für den Betrag des Wachstumsfaktors des untersuchten Ansatzes im gesamten Bereich $|g(\sigma_{\mathrm{B}})| = 1$ gegeben ist. Der Betrag des Wachstumsfaktors $|g(\sigma_{\mathrm{B}})|$ des Vergleichsalgorithmus schwankt wiederum mit einer maximalen Abweichung von $0,01$ um den Wert $|g(\sigma_{\mathrm{B}})| = 1$. Die Folgen hiervon können bei der Impuls-Propagation beobachtet werden. Hierzu wird in dem System, was in Abbildung 4.4 dargestellt ist, ein gaußförmiger Impuls mit einer FWHM-Bandbreite $B = 140\,\mathrm{THz}$ und einer Amplitude von $E_0 = 1\,\mathrm{V/m}$ initialisiert. Der Impuls wird so initialisiert, dass er in positiver z-Richtung propagiert. Die Simulation wird für 250 Zeitschritte Δt durchgeführt. Die Ergebnisse sind in Abbildung 4.4 dargestellt.

Bei der Betrachtung fällt auf, dass die Feldverteilung des hier betrachteten Algorithmus den Erwartungen entspricht. Die Amplituden des an dem Brechzahlsprung transmittierten Anteils und des reflektierten Anteils entsprechen mit $0{,}5\,\mathrm{V/m}$ dem Wert, der gemäß der Fresnel-Formeln zu erwarten ist. Die finale Feldverteilung des Vergleichsalgorithmus zeigt ein anderes Verhalten. Bei dieser ist die Amplitude des Impulses auf einen Wert von $0{,}033\,\mathrm{V/m}$ abgesunken. Diese Betrachtung verdeutlicht die unitären Eigenschaften des hier vorgestellten Algorithmus. Selbst bei einer unzureichenden Polynomentwicklung bleibt dieser stabil und erhält die Energie im

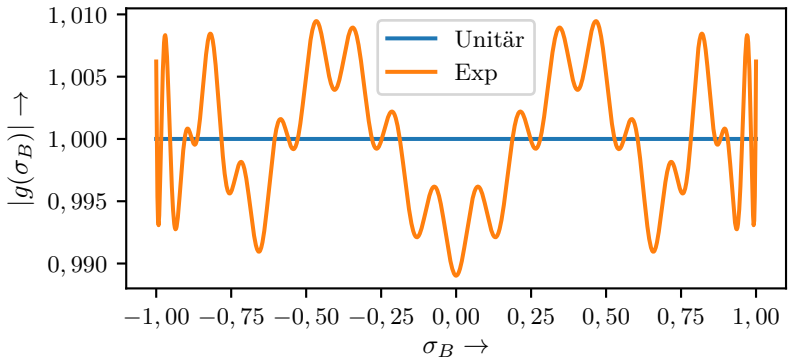


Abbildung 4.3: Die Abbildung zeigt die Absolutbeträge der Wachstumsfaktoren $g(\sigma_\mathrm{B})$ in Abhängigkeit der normierten Eigenwerte σ_B der untersuchten Algorithmen. Die blaue Linie zeigt den untersuchten Ansatz, während die gestichelte grüne Linie den Algorithmus mit dem Exponentialoperator darstellt.

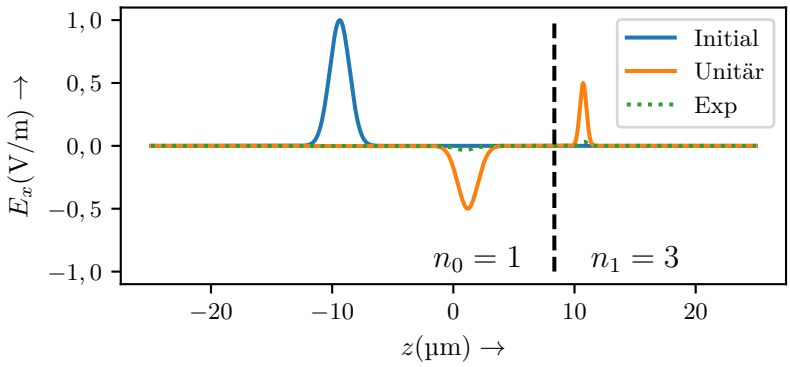


Abbildung 4.4: Die Abbildung zeigt die E_x-Komponenten der initialen Feldverteilung mit der gestrichelten blauen Linie. Im Simulationsbereich liegt ein Brechzahlsprung vor, dessen Position markiert ist. Das Ergebnis des vorgestellten Ansatzes ist mit der grünen Linie dargestellt, während der des Vergleichsalgorithmus mit dem Exponentialoperator mit der gepunkteten roten Linie dargestellt wird.

System. Die Bedingung hierfür ist, dass (4.7) erfüllt ist. Bei dem Vergleichsalgorithmus sorgt die nicht vollständig konvergierte Approximation für einen Wachstumsfaktor, welcher signifikant von eins abweicht. Das hat in diesem Beispiel zur Folge, dass die Impulse stark gedämpft werden. Im Allgemeinen führt dies allerdings zu instabilen Simulationen. Um dies zu vermeiden, muss bei dem Vergleichsalgorithmus daher eine vollständige Konvergenz wie in [53] sichergestellt werden.

In der vorangegangenen Betrachtung wird gezeigt, wie die Formulierung des vorgestellten Ansatzes gewährleistet, dass der Betrag des Wachstumsfaktors immer eins beträgt. Auch wenn damit die Stabilität gesichert ist, treten Fehler in Form von Phasenfehlern auf, wenn die Entwicklungsordnung für die gewählte Schrittweite nicht hoch genug ist. Um dies zu untersuchen, wird der Phasenfehler des Wachstumsfaktors in Abhängigkeit von der Zeitschrittweite für eine Entwicklungsordnung von $N_\mathrm{Pol} = 50$ untersucht. Der Wachstumsfaktor der Approximation

ist mit $g(\sigma) = \hat{\mathcal{P}}(\sigma)/2 \pm \sqrt{\hat{\mathcal{P}}^2(\sigma)/4 + 1}$ gegeben. Die analytische Referenz ist mit $g_{\text{ref}}(\sigma) = 2\sinh(\sigma)/2 \pm \sqrt{\sinh^2(\sigma) + 1}$ gegeben. Nun wird der mittlere Fehler der Phase β von $g(\sigma)$ in dem zuvor gewählten Eigenwertbereich bestimmt, welcher mit dem Signalspektrum korrespondiert. Die mittlere Abweichung der Phase wird durch Integration im Eigenwertspektrum bestimmt: $\Delta\beta = \int_{\sigma_{\min}}^{\sigma_{\max}} |\beta(g(\sigma) - \beta(g_{\text{ref}}(\sigma))| \mathrm{d}\sigma$. Die Parametrisierung des vorherigen Beispiels wird übernommen. Die Ergebnisse sind in Abbildung 4.5 dargestellt. Der Fehler ist für kleine Zeitschrittweiten

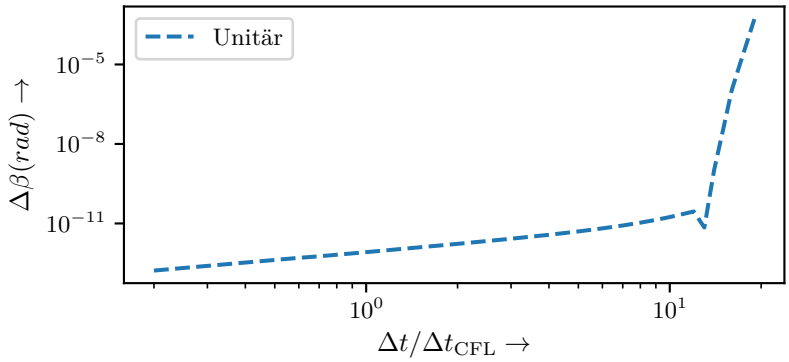


Abbildung 4.5: Die Abbildung zeigt die mittlere Abweichung der Phase des Wachstumsfaktors zu dem analytischen Wert in Abhängigkeit von Δt. Hier erfolgt die Betrachtung erfolgt für das Signalspektrum. Die Zeitschrittweite ist mit der CFL-Zeitschrittweite Δt_{CFL} normiert.

Δt gering und steigt langsam mit größer werdenden Zeitschrittweiten an. Ab einem gewissen Punkt, welcher hier bei $\Delta t/\Delta t_{\text{CFL}} \approx 15$ liegt, nimmt er rapide zu. Die genaue Position von diesem Punkt hängt hierbei von der verwendeten Polynomordnung ab. Höhere Zeitschrittweiten erfordern hier höhere Polynomordnungen.

4.4 Diskussion

Mit diesem ersten Ansatz wird ein unitärer Algorithmus zur Lösung der Maxwell-Gleichungen im Zeitbereich mit allen Feldkomponenten vorgestellt. Die Polynomapproximation mit den Tschebyscheff-Polynomen erlaubt es, Zeitschritte zu wählen, welche die CFL-Zeitschrittweite Δt_{CFL} für das verwendete Gitter übersteigen. Damit ist es möglich, die in Abschnitt 3.2.2 beschriebene Abhängigkeit des Zeitschrittes Δt in (3.16) von der örtlichen Diskretisierung zu überwinden. Die Besonderheit hierbei ist, dass die Unitarität nicht von der Approximation abhängt, sondern durch die Formulierung des Algorithmus sichergestellt wird. Darüber hinaus kann der Algorithmus vollständig explizit berechnet werden, was eine einfache Parallelisierung ermöglicht. Von besonderer Bedeutung ist hierbei das Eigenwertspektrum der Systemmatrix $\mathcal{H}$. Durch die Betrachtung des Operators in Abhängigkeit von diesem ist es möglich, den Approximationsbereich einzugrenzen und Bereiche zu identifizieren, welche das Signalspektrum beschreiben. Dies wird bei der Wahl der Fensterfunktion genutzt. Dadurch kann auch bei niedriger Approximationsordnung die Stabilität des Zeitpropagationsschemas gewährleistet werden, was in dem Beispiel 4.4 illustriert ist.

Die Wahl der Fensterfunktion und die Approximation mit den Polynomen bieten noch weitere interessante Möglichkeiten zur Verbesserung des Algorithmus. Dabei hat sich insbesondere die Verwendung von Gewichtsfunktionen als vielversprechender Ansatz erwiesen, um die Approximation besser zu steuern. Außerdem spielt die Wahl der Startwerte eine Rolle. Bei ihrer Wahl sollte die numerische Dispersion des Gitters, wie in 3.1.1 beschrieben, in Betracht gezogen werden. Durch die numerische Dispersion kann es selbst bei der Verwendung von analytischen Lösungen für die Startwerte zu Abweichungen kommen, da sie nicht mit denen im diskretisierten System übereinstimmen.

Die unitäre Formulierung hat jedoch einen natürlichen Nachteil. Ohne Weiteres ist es nicht möglich, dämpfende Materialmodelle in die Systemmatrix $\mathcal{H}$ aufzunehmen. Dies liegt hierbei nicht in der Approximation mit den Tschebyscheff-Polynomen begründet, sondern in der Formulierung des Propagationsschemas. Bei der beschriebenen Formulierung können nur Systemmatrizen verwendet werden, deren Eigenwerte auf der imaginären Achse liegen. Hierbei kann es sich zum Beispiel um Drude- oder Lorentz-Modelle mit Dämpfung, wie sie in Abschnitt 2.2 beschrieben werden, oder um PMLs wie in 2.4 handeln. Es ist zwar möglich, diese mithilfe von Operator-Splitting-Ansätzen [66, 67] zu berücksichtigen, allerdings wird der Algorithmus dadurch deutlich komplexer und es werden durch das Splitting zusätzliche Fehler eingeführt. Daher sollen in den folgenden Kapiteln alternative Ansätze auf Basis von Polynomapproximationen untersucht werden, welche eine direkte Einbindung in die Systemmatrix erlauben.

5 Faberpolynome zur Zeitpropagation

In dem folgenden Kapitel soll ein Algorithmus zur numerischen Lösung der Maxwell-Gleichungen auf Basis von Faberpolynom-Approximationen untersucht werden. Unter anderem soll es das Ziel sein, die Probleme des zuvor beschriebenen Ansatzes zu lösen. Insbesondere die Einbindung von dämpfenden Materialmodellen ist hier von Bedeutung. Außerdem sollen für die neue Methode Materialmodelle aller Art, wie beispielsweise dispersive Materialien, flexibel eingesetzt werden können. Hierzu wird mit den Faberpolynomen eine weitere Klasse von polynomialen Integratoren untersucht. Die Faberpolynome sind 1903 von Georg Faber eingeführt worden [68].

Die Faberpolynome sind für diesen Zweck interessant, weil sie eine Approximation von Funktionen in der komplexen Ebene erlauben, welche für einen bestimmten Approximationsbereich stabil ist [69, 70]. Sie ermöglichen darüber hinaus die Berücksichtigung von Form und Ort des Eigenwertspektrums der Systemmatrix in der komplexen Ebene. Im Gegensatz zu dem auf Tschebyscheff-Polynomen basierenden Verfahren in [53] können so auch Systeme betrachtet werden, in welchen dämpfende Materialien vorhanden sind oder zum Beispiel absorbierende Randbedingungen wie PMLs angewendet werden. Außerdem können die Faberpolynome mithilfe einer Rekursionsbeziehung berechnet werden [71]. Wie später gezeigt, ermöglicht diese Eigenschaft eine effiziente Berechnung, welche auch die Implementierung auf parallelen Architekturen wie Mehrkern CPUs oder sogar GPUs erlaubt. Diese Merkmale machen die Faberpolynome zu einem interessanten Lösungsansatz für die numerische Lösung der Maxwell-Gleichungen im Zeitbereich. In den folgenden Abschnitten sollen diese daher genauer beleuchtet werden.

Zuerst wird auf die bestehende Literatur zur Anwendung von Faberpolynomen eingegangen. Außerdem wird der Ansatz in den Kontext mit alternativen Methoden gesetzt. Im Anschluss wird die Verwendung für die Lösung der Maxwell-Gleichungen beleuchtet. Hierbei wird auf die auftretenden Probleme eingegangen und entsprechende Lösungsansätze werden vorgestellt. In diesem Zusammenhang wird insbesondere die Abschätzung des Eigenwertspektrums der Systemmatrix analysiert, da diese von großer Bedeutung für die Approximation mithilfe der Faberpolynome ist. Im Anschluss wird die Effizienz der resultierenden Verfahren im Hinblick auf die Genauigkeit der Entwicklung und den Rechenaufwand analysiert.

5.1 Vorüberlegungen und Einordnung

Zuerst soll das Zeitpropagationsschema formuliert werden. Der Startpunkt für die Überlegungen sind die Maxwell-Gleichungen im Zeitbereich. Die örtliche Diskretisierung erfolgt hierbei beispielsweise mit dem in Abschnitt 3.3 beschriebenen FD-Yee-Gitter. Zunächst soll ein System ohne externe Ströme betrachtet werden. Die formale Lösung der örtlich diskretisierten Maxwell-Gleichungen (3.19) ist mit

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t \mathcal{H}) \vec{\Psi}(t_n) \tag{5.1}$$

gegeben. Auf Basis dieser Formulierung soll der Zeitpropagations-Algorithmus realisiert werden, indem das Matrixexponential in (5.1) geeignet approximiert wird. Im Gegensatz zu der in Kapitel 4 vorgestellten Methode liegt hier keine direkte Stabilitätsbedingung vor. Um die Stabilität zu garantieren und die in den Abbildungen 4.4 und 4.3 illustrierten divergierenden Lösungen zu verhindern, ist hier eine präzise Approximation des Matrixexponentials in (5.1) erforderlich [59]. In [72] werden diverse Methoden für die Berechnung des Matrixexponentials vorgestellt. Wie im vorangegangenen Kapitel ist das Ergebnis der Berechnung für ein Matrixexponential wie $\exp(\Delta t\mathcal{H})$ im Allgemeinen dicht besetzt [64]. Damit ist eine einmalige Vorberechnung des gesamten Matrixexponentials $\exp(\Delta t\mathcal{H})$ aufgrund der Größe von $\mathcal{H}$ nicht praktikabel. Dennoch kann dieser Ansatz in Spezialfällen sehr effizient sein. Dies ist insbesondere dann der Fall, wenn es sich bei $\mathcal{H}$ um ein spezielles Modell handelt, bei dem $\exp(\Delta t\mathcal{H})$ sehr effizient bestimmt werden kann. Dies gilt beispielsweise für homogene Rechengebiete [4]. Hier soll allerdings von einem allgemeineren System ausgegangen werden. Wie im Kapitel 4 sollen daher Methoden in den Blick genommen werden, welche, statt zuerst $\exp(\Delta t\mathcal{H})$ zu approximieren und anschließend mit der resultierenden vollbesetzten Matrix (5.1) auszuwerten, $\exp(\Delta t\mathcal{H})\vec{\Psi}(t)$ direkt bestimmen. Hierbei sollen nur Matrix-Vektor-Multiplikationen mit $\mathcal{H}$ verwendet werden.

In der Literatur wird hierzu eine Reihe von Lösungen diskutiert. Vertreter dieser Methoden ist beispielsweise die oben schon beschriebene Approximation mit Tschebyscheff-Polynomen, welche in [53] für die Lösung der Maxwell-Gleichungen mit einem Operator der Form (5.1) verwendet werden. Dieser Ansatz ist nur für symmetrische beziehungsweise schiefsymmetrische Matrizen $\mathcal{H}$ geeignet [53]. In [73] wird eine Integration von absorbierenden Randbedingungen mit Splitting-Ansätzen diskutiert. Eine direkte Integration in die Systemmatrix $\mathcal{H}$ ist allerdings nicht möglich.

Ein weiterer Ansatz sind die Krylov-Unterraum-Methoden, welche in [33, 74–76] für die Lösung der Maxwell-Gleichungen untersucht werden. Für allgemeine Matrizen kann hierbei der Arnoldi-Prozess verwendet werden. Für hermitesche Matrizen kann der Lanczos-Prozess eingesetzt werden. Dieser hat den Vorteil, dass er mit einer kurzen Rekursionsbeziehung zu berechnen ist und den Speicheraufwand gegenüber dem Arnoldi-Prozess reduziert [64]. Eine weitere Klasse von Methoden basiert auf Leja-Punkten [77–79].

Hier sollen Faberpolynome verwendet werden, um das Matrixexponential in (5.1) zu entwickeln. Die Anwendung von Faberpolynomen wird in der Literatur ausführlich behandelt. Hierbei werden diese zur Approximation von Funktionen in der komplexen Ebene [69, 70, 80–82], aber auch zur Approximation von Matrixfunktionen verwendet [83–86]. Die Anwendung zur Lösung von partiellen Differenzialgleichungen wird in [71, 87–89] diskutiert. Die Verwendung von Faberpolynomen zur Lösung der Maxwell-Gleichungen im Zeitbereich ist bisher nur in [90, 91] untersucht worden. Da sie eine Approximation in der komplexen Ebene erlauben, können auch dämpfende Materialien sowie absorbierende Randbedingungen ohne Splitting-Ansätze direkt in der Systemmatrix $\mathcal{H}$ berücksichtigt werden. Darüber hinaus erlaubt die Polynom-Klasse eine rekursive Berechnung mit festen Speicheranforderungen, die von dem verwendeten Zeitschritt unabhängig sind. Dies ist ein Vorteil gegenüber den Krylov-Unterraum-Methoden, bei denen im allgemeinen Fall, wenn der Arnoldi-Algorithmus verwendet wird, die Speicheranforderungen mit der Größe des Zeitschritts ansteigen [35]. Außerdem erlauben die Faberpolynome eine flexible Wahl des Konvergenzbereiches, was eine Optimierung an das verwendete Materialmodell erlaubt. Der Nachteil hierbei ist, dass, im Gegensatz zu den Krylov-Unterraum-Methoden, im Vorfeld zumindest die Grenzen des Eigenspektrums der Systemmatrix $\mathcal{H}$ bekannt sein müssen. Im

Folgenden werden daher effiziente Methoden zur Abschätzung des Spektrums vorgestellt. In den bestehenden Untersuchungen zur Lösung der Maxwell-Gleichungen [90, 91] zeigt sich die Faberpolynom-Methode als vielversprechender Ansatz. Besonders in [87, 90, 92] erweist sie sich darüber hinaus als kompetitiv mit den Krylov-Unterraum-Methoden in Hinblick auf die Rechenzeit und die Speicheranforderungen. Allerdings wird die Untersuchung in [90, 91] auf lineare, quellfreie Probleme beschränkt. In den folgenden Kapiteln werden daher noch effiziente Möglichkeiten zur Einbindung von Quelltermen wie Stromdichte entwickelt und die Methode für die Berücksichtigung von Nichtlinearitäten erweitert. Daher soll die Approximation von Matrixfunktionen mit Faberpolynomen genauer beleuchtet werden. Teile der Untersuchungen in den folgenden Abschnitten sind in [KS6, KS7] veröffentlicht und werden im Folgenden mit Ergänzungen dargestellt.

5.2 Approximation mit Faberpolynomen

Im aktuellen Fall soll die Faberpolynom-Entwicklung zur Approximation des Matrixexponentials (5.1) verwendet werden. Im Rahmen dieser Arbeit werden allerdings noch Approximationen für weitere Matrixfunktionen benötigt. Daher soll die Faberpolynom-Entwicklung allgemein erläutert werden. Wird die Faberpolynom-Entwicklung auf (5.1) angewendet, so ergibt sich formal die folgende Approximation für das Zeitpropagation-Schema:

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t \mathcal{H})\vec{\Psi}(t_n) = f(\mathcal{H})\vec{\Psi}(t_n) = \left(\sum_{m=0}^{\infty} c_m F_m(\mathcal{H}) \right) \vec{\Psi}(t_n). \tag{5.2}$$

Hierbei handelt es sich bei $F_m(\mathcal{H})$ um das Faberpolynom m-ter Ordnung. Bei c_m handelt es sich um den zum m-ten Polynom zugehörigen Entwicklungskoeffizienten. Die Definition der Faberpolynome und die Entwicklung nach diesen findet im Eigenwertraum der Systemmatrix $\mathcal{H}$ statt. Wie im letzten Kapitel bereits beschrieben, müssen nur die Eigenwerte der Matrix $\mathcal{H}$ für die Approximation berücksichtigt werden [64]. Daher sollen die folgenden Überlegungen in der komplexen z-Ebene stattfinden. In dieser z-Ebene sollen auch die Eigenwerte von $\mathcal{H}$ definiert sein. Aus dem Grund wird im Folgenden $\mathcal{H} \to z$ definiert. Für die Matrixfunktion $f(\mathcal{H})$ gilt demnach:

$$f(\mathcal{H}) \to f(z) = \exp(\Delta t z). \tag{5.3}$$

Praktisch lässt sich die Approximation von $f(\mathcal{H})$ beziehungsweise $f(z)$ als eine Approximation einer komplexwertigen Funktion ansehen:

$$f(z) = \sum_{m=0}^{\infty} c_m F_m(z). \tag{5.4}$$

Die Faberpolynome $F_m(z)$ sind hierbei für einen kompakten Bereich K in der komplexen z-Ebene definiert, sodass das Komplement von K einfach zusammenhängend ist [84, 91]. Mithilfe des Riemannschen Abbildungssatzes lässt sich zeigen, dass es in diesem Fall eine konforme Abbildung $\psi(w)$ gibt, welche das Komplement eines geschlossenen Kreises mit dem Radius ρ und dem Mittelpunkt im Ursprung in einer komplexen w-Ebene auf das Komplement von K abbildet [93]. Die Abbildung zwischen den komplexen Ebenen ist in Abbildung 5.1 illustriert. Die Größe ρ ist hierbei die sogenannte logarithmische Kapazität von K und ist über den Radius

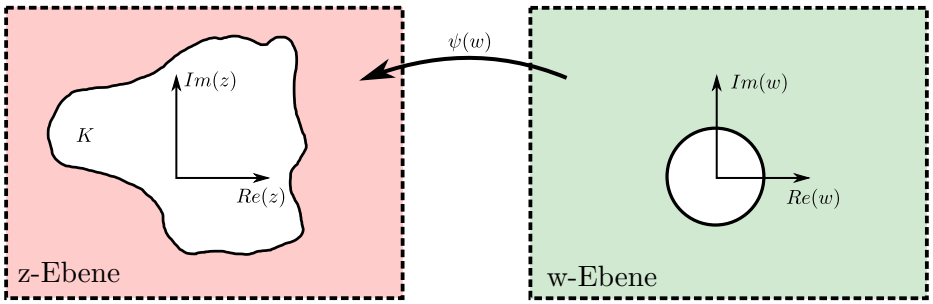


Abbildung 5.1: Die Abbildung zeigt schematisch das Konvergenzgebiet K in der komplexen z-Ebene sowie den Kreis in der komplexen w-Ebene. Es wird außerdem die Abbildung des Komplements des Kreises in der w-Ebene auf das Komplement von K in der z-Ebene mit der konformen Abbildung $\psi(w)$ dargestellt.

des oben genannten Kreises definiert [70, 83, 88, 94]. Die konforme Abbildung $\psi(w)$ hat eine Laurent-Entwicklung in ∞, welche die Form

$$\psi(w) = w + \gamma_0 + \gamma_1 w^{-1} + \gamma_2 w^{-2} + ... = w + \sum_{k\geq 0}^{\infty} \gamma_k w^{-k} \tag{5.5}$$

hat. Hierbei wird für $\psi(w)$ die Normierung $\lim_{|w|\to\infty} \psi(w)/w \to 1$ verwendet [88, 90]. Handelt es sich bei K um einen Bereich, der sich mit Polygonen darstellen lässt, so kann die Laurent-Entwicklung der konformen Abbildung $\psi(w)$ mithilfe der Schwarz-Christoffel-Transformation bestimmt werden [71, 90]. Hierzu kann beispielsweise [95] verwendet werden. Für speziellere Formen von K existieren weitere analytische Entwicklungen, auf die später genauer eingegangen wird. Mit der Laurent-Entwicklung für $\psi(w)$ lassen sich die Faberpolynome formal mit

$$\frac{\psi'(w)}{\psi(w) - z} = \sum_{m\geq 0}^{\infty} \frac{F_m(z)}{w^{m+1}} \rho^{-m} \tag{5.6}$$

angeben, wobei $z \in K$ gilt [70, 94]. Hierbei gibt $\psi'(w)$ die Ableitung von $\psi(w)$ nach w an. Während sich die Faberpolynome formal mit (5.6) bestimmen lassen, sind diese auch über eine Rekursionsbeziehung definiert [88, 90]:

$$F_{m+1}(z) = zF_m(z) - \sum_{k=0}^{m} \gamma_k F_{m-k}(z) - m\gamma_m. \tag{5.7}$$

Für die Startbedingungen der Rekursionsbeziehung gilt $F_0 = 1$ und für m gilt $m \geq 0$ [88]. Mit dieser lässt sich die Approximation berechnen, ohne im Vorfeld die Faberpolynome explizit bestimmen zu müssen. Die Entwicklungskoeffizienten c_m in (5.2) lassen sich mit

$$c_m = \frac{1}{2\pi j} \int_{|w|=R} \frac{f(\psi(w))}{w^{m+1}} \mathrm{d}w, \; R \geq \rho \tag{5.8}$$

berechnen. Wie in (5.8) zu erkennen, wird entlang eines Kreises C_R mit $|w| = R$, der in der w-Ebene definiert ist, integriert. Hierbei muss R mit $R > \rho$ ausreichend klein sein, damit die

Funktion f auf dem Gebiet I_R analytisch fortgesetzt werden kann [83, 91]. I_R ist ein Gebiet in der z-Ebene, welches durch die Jordan-Kurve Γ_R begrenzt wird. Die Jordan-Kurve Γ_R ist die Abbildung von dem Kreis C_R mit $\psi(w)$. Wenn $\psi(w)$ auf der Grenzfläche des Kreises auch kontinuierlich fortgesetzt werden kann und wenn K durch eine geschlossene Jordan-Kurve C begrenzt wird, kann auch $R = \rho$ verwendet werden [70, 71, 90]. Daher soll im Folgenden das Gebiet K immer so gewählt werden, dass es von einer solchen geschlossenen Jordan-Kurve C begrenzt wird. Eine Jordan-Kurve ist eine stetige Kurve ohne Überschneidungen. Beispiele für solche Kurve C sind Kreise, Ellipsen oder Polygone.

Die Voraussetzung für die Konvergenz der Faberpolynom-Entwicklung ist, dass die Funktion $f(z)$ in dem Gebiet K analytisch ist. Wenn $f(z)$ in der gesamten z-Ebene analytisch ist, dann konvergiert die Faberpolynom-Approximation superlinear [84]. Beide Bedingungen sind für Matrixexponential in (5.1) erfüllt.

Damit die Faberpolynom-Entwicklung für $f(\mathcal{H})$ in (5.2) stabil ist, müssen alle Eigenwerte von der Systemmatrix $\mathcal{H}$ in dem Gebiet K liegen, wie in Abbildung 5.2 exemplarisch dargestellt. Hierbei

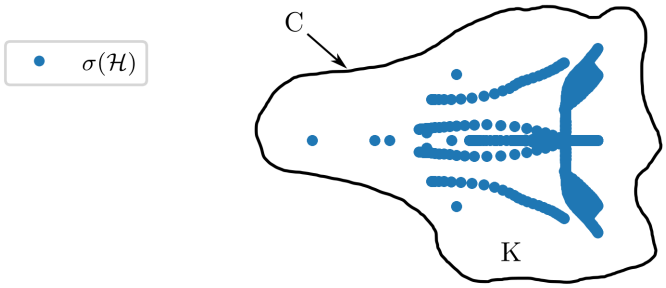


Abbildung 5.2: Die Abbildung zeigt exemplarisch das Eigenwertspektrum einer Systemmatrix $\mathcal{H}$ in der komplexen z-Ebene. Die einzelnen Eigenwerte sind mit blauen Punkten dargestellt. C stellt die Grenze der Ebene K dar.

sollte K möglichst genau mit dem Eigenwertspektrum $\sigma(\mathcal{H})$ übereinstimmen. Je enger $\sigma(\mathcal{H})$ von dem Gebiet umschlossen wird, desto schneller konvergiert die Faberpolynom-Approximation [84].

Das Vorgehen bei der Approximation mit Faberpolynomen ist im vorangegangenen Abschnitt allgemein beschrieben. Die einzelnen Schritte, welche in der Praxis benötigt werden, um die Approximation durchzuführen, sind im Anhang B.3.1 zusammengefasst. Im Anhang C.1 wird das Vorgehen bei der Approximation zusätzlich an einem Beispiel illustriert.

5.3 Implementierungsaspekte

Mit den Grundlagen aus dem vorangegangenen Abschnitt lässt sich ein Algorithmus zur Zeitpropagation mit der Faberpolynom-Approximation konstruieren. Der schematische Ablauf des Algorithmus ist in Abbildung 5.3 dargestellt. Grundsätzlich kann die benötigte Faberpolynom-Approximation in zwei Schritte unterteilt werden: Im ersten Schritt muss das Gebiet K so gewählt werden, dass alle Eigenwerte von der Systemmatrix $\mathcal{H}$ darin liegen. Mit diesem Gebiet

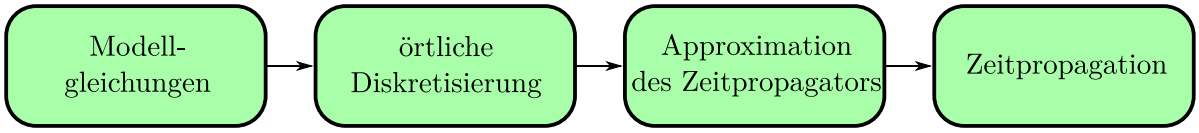


Abbildung 5.3: Die Abbildung zeigt den schematischen Ablauf des Algorithmus.

wird die konforme Abbildung $\psi(w)$ bestimmt. Liegt keine spezielle finite Laurent-Entwicklung für $\psi(w)$ vor, muss die Laurent-Entwicklung (5.5) bei einem Wert $k = N_L$ abgebrochen werden [71, 84]. Im zweiten Schritt werden die Koeffizienten c_m der Faberpolynom-Entwicklung bestimmt. Hierzu muss (5.8) für die Koeffizienten gelöst werden. In der Praxis muss hier die Entwicklung an einem maximalen Polynomgrad $m = N_{\text{pol}}$ abgebrochen werden. Im Anschluss kann die Approximation genutzt werden, um (5.2) zu berechnen. Für die praktische Umsetzung lassen sich allerdings einige Probleme identifizieren. Auf diese wird im folgenden Abschnitt eingegangen.

5.3.1 Wahl des Konvergenzbereiches

Wie im vorangegangenen Abschnitt beschrieben, muss zuerst ein Bereich K in der komplexen Ebene gewählt werden. Dieser muss das Eigenwertspektrum der Systemmatrix $\mathcal{H}$ enthalten. Wenn die Grenze von K durch einen Polygonzug beschrieben werden kann, kann die Schwarz-Christoffel-Transformation genutzt werden, um eine Laurent-Entwicklung der Form (5.5) für $\psi(w)$ zu bestimmen [95]. Der Vorteil dieses Vorgehens ist, dass die Form von K sehr frei gewählt werden kann. Der Nachteil ist, dass mit steigender Anzahl an Termen bei der Laurent-Entwicklung in (5.5) auch die Anzahl der Terme der Rekursionsbeziehung (5.7) für die Berechnung ansteigt. Die Anzahl der Matrix-Vektor-Multiplikationen steigt zwar nicht mit, allerdings steigt die Anzahl der Vektoren, welche während der Berechnung im Speicher vorliegen müssen. Diese Vektoren haben jeweils die Größe des diskretisierten Feldes $\vec{\Psi}(t_n)$. Daher ist es wünschenswert, die Anzahl der Terme in (5.5) gering zu halten, indem die Entwicklung bei einem niedrigen Wert $k = N_L$ abgebrochen wird. Für einige Formen für K gibt es konforme Abbildungen mit einer geringen Anzahl an Koeffizienten. Hierbei handelt es sich beispielsweise bei K um

- einen Kreis [69],
- eine Ellipse [88, 90, 91]
- oder ein abgerundetes Quadrat [88].

Im Fall des Kreises ist die Abbildung mit

$$\psi(w) = w + \gamma_0 \tag{5.9}$$

gegeben [69]. Für die Ellipse ergibt sich mit

$$\psi(w) = w + \gamma_0 + \gamma_1/w \tag{5.10}$$

eine kompakte Laurent-Reihe, die nach γ_1 abbricht [90, 91]. Das abgerundete Quadrat hat die folgende Entwicklung [88]:

$$\psi(w) = w + \gamma_0 - 1/(2w)^3. \tag{5.11}$$

Wie in 3.3 beschrieben, hat die untersuchte Systemmatrix für den dämpfungsfreien Fall ein rein imaginäres Eigenwertspektrum $\sigma(\mathcal{H})$. Wenn auch dämpfende Materialien betrachtet werden, erweitert sich das Spektrum in die negative, reelle Halbebene. Allerdings resultiert die Dämpfung der meisten Materialmodelle in reellen Eigenwerten, welche klein gegenüber dem Imaginärteil sind. Daher ist ein Kreis eine ungeeignete Wahl für das Konvergenzgebiet. Aus diesen Gründen wird auch das abgerundete Quadrat nicht weiter betrachtet. In vielen Fällen ist die Ellipse eine gute Näherung für die Form des Eigenwertspektrums $\sigma(\mathcal{H})$. Der Algorithmus soll daher für elliptische Gebiete K genauer betrachtet werden. Diese Gebiete werden auch in den Algorithmen verwendet, welche von [90] und [91] vorgestellt werden. Mit der elliptischen Form kann die Rekursionsbeziehung (5.7) wie folgt umgeschrieben werden [69, 88]:

$$F_{m+1}(z) = (z - \gamma_0)F_m(z) - \gamma_1 F_{m-1}(z) \quad ; \quad m \geq 2. \tag{5.12}$$

Die Startbedingungen für die Rekursion sind mit

$$\begin{aligned} F_0(z) &= 1 \\ F_1(z) &= z - \gamma_0 \\ F_2(z) &= (z - \gamma_0)F_1(z) - 2\gamma_1 \end{aligned} \tag{5.13}$$

gegeben [69, 88]. Für eine ausführliche Herleitung sei auf Anhang B.3.2 verwiesen. Für $\gamma_0 = 0$ und für $\gamma_1 = 1/4$ entsprechen die Faberpolynome hierbei den normalisierten Tschebyscheff-Polynomen [88]. Die Ellipse in der komplexen z-Ebene kann mit

$$(x - x_0)^2/a^2 + (y - y_0)^2/b^2 = 1 \tag{5.14}$$

beschrieben werden. Hierbei gilt $z = x + jy$. Der Mittelpunkt der Ellipse ist mit $z_0 = x_0 + jy_0$ gegeben. Die Größen a und b sind die Halbachsenparameter der Ellipse. Mit der konformen Abbildung (5.10) lassen sich diese mit

$$a = \rho + \gamma_1/\rho \quad \text{und} \quad b = \rho - \gamma_1/\rho \tag{5.15}$$

angeben [90, 91]. Bei ρ handelt es um die logarithmische Kapazität des Gebiets K, welches hier eine Ellipse ist. (5.15) lässt sich zu dem Ausdruck

$$\rho = (a + b)/2 \tag{5.16}$$

zusammenfassen Wie in [83, 88, 90, 91] beschrieben, muss, zur Gewährleistung der Stabilität auch für hohe Entwicklungsordnungen N_{pol}, das Eigenwertspektrum $\sigma(\mathcal{H})$ skaliert werden. Dieses soll so skaliert werden, dass das Gebiet K eine logarithmische Kapazität $\rho = 1$ aufweist [83]. Hierzu wird ein Skalierungsfaktor λ_s eingeführt, mit dem die Systemmatrix $\mathcal{H}$ und damit auch ihr Spektrum skaliert werden [77, 90, 91]. Für den Skalierungsfaktor λ_s gilt

$$\mathcal{H}_s = \mathcal{H}/\lambda_s, \tag{5.17}$$

sowie

$$\Delta t_s = \Delta t \lambda_s. \tag{5.18}$$

Dadurch wird erreicht, dass das Spektrum von einer Ellipse mit einer logarithmischen Kapazität von $\rho = 1$ umschlossen werden kann. Mit (5.17) und (5.18) kann die Approximation (5.2) wie folgt umgeschrieben werden:

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t_s \mathcal{H}_s)\vec{\Psi}(t_n) = f(\mathcal{H}_s)\vec{\Psi}(t_n) = \left(\sum_{m=0}^{\infty} c_m F_m(\mathcal{H}_s)\right)\vec{\Psi}(t_n). \tag{5.19}$$

Nun muss die Ellipse bestimmt werden, um die konforme Abbildung (5.5) und damit schließlich die Approximation berechnen zu können. Hierzu sind einige Annahmen bezüglich des Spektrums der Systemmatrix $\mathcal{H}$ nötig. Eine Möglichkeit ist es, die Form des Eigenwertspektrums als rechteckig anzunehmen wie in [90, 91]. Dieses Vorgehen ist insofern vorteilhaft, als sich die oberen und unteren Grenzen der Real- und Imaginärteile des Eigenwertspektrums durch geeignete Methoden gut abschätzen lassen. Eine Möglichkeit hierzu wird im Rahmen dieser Arbeit noch vorgestellt.

Das elliptische Konvergenzgebiet muss dann um das Rechteck, welches das Eigenwertspektrum der Systemmatrix enthält, platziert werden. Die Grenzen des Realteils des Rechtecks seien hier mit $[R_{\min}, R_{\max}]$ gegeben. Die Grenzen des Imaginärteils sollen mit $[I_{\min}, I_{\max}]$ gegeben sein. In diesem Fall ist der Mittelpunkt des Rechtecks mit $z_0 = x_0 + jy_0$ bestimmt, wobei $x_0 = R_{\min} + (R_{\max} - R_{\min})/2$ und $y_0 = I_{\min} + (I_{\max} - I_{\min})/2$. Zusätzlich liegt noch der Skalierungsfaktor λ_s als freier Parameter vor. Dieser soll, wie oben beschrieben, so gewählt werden, dass $\rho = 1$ gilt. Wird (5.18) in Betracht gezogen, so wird klar, dass, um den größten Zeitschritt zu erreichen, λ_s möglichst minimal sein muss [88, 90]. Nun wird das skalierte Rechteck betrachtet, was mit $[R_{\min,s}, R_{\max,s}]$ und $[I_{\min,s}, I_{\max,s}]$ gegeben ist. $R_{\min,s}$ lässt sich gemäß (5.17) mit $R_{\min,s} = R_{\min}/\lambda_s$ aus $R_{\min}$ bestimmen. Die Bestimmung der anderen Größen erfolgt analog. Der Mittelpunkt des skalierten Rechtecks und damit auch γ_0 ist mit

$$\gamma_0 = z_0/\lambda_s = (x_0 + jy_0)/\lambda_s \tag{5.20}$$

gegeben. Für die Bestimmung der restlichen Größen werden zunächst einige Hilfsgrößen eingeführt:

$$c = |R_{\max} - R_{\min}|/2 \tag{5.21}$$

$$l = |I_{\max} - I_{\min}|/2 \tag{5.22}$$

beziehungsweise

$$c_s = |R_{\max,s} - R_{\min,s}|/2 \tag{5.23}$$

$$l_s = |I_{\max,s} - I_{\min,s}|/2. \tag{5.24}$$

Die Notation ist an der Notation in [91] orientiert. Die Ellipse sowie der Skalierungsprozess sind in Abbildung 5.4 schematisch dargestellt. Das Ziel ist es nun, eine Ellipse minimaler Größe zu finden, welche das skalierte Rechteck umschließt. In diesem Fall liegen die Ecken des Rechtecks auf dem Rand der Ellipse, sodass durch Einsetzen in (5.14)

$$\frac{b^2}{a^2} = \frac{b^2 - l_s^2}{c_s^2} \tag{5.25}$$

bestimmt werden kann. Unter diesen Voraussetzungen kann die Ellipse mit der minimalen Fläche bestimmt werden. Gemäß [91] ist die Ellipse mit minimaler Fläche für

$$b = \sqrt{l_s^2 + (c_s l_s^2)^{2/3}} \tag{5.26}$$

gegeben, wobei $\rho = 1$ gilt. Da $\rho = 1$ gilt, ist mit (5.16) auch a bestimmt:

$$a = 2 - b. \tag{5.27}$$

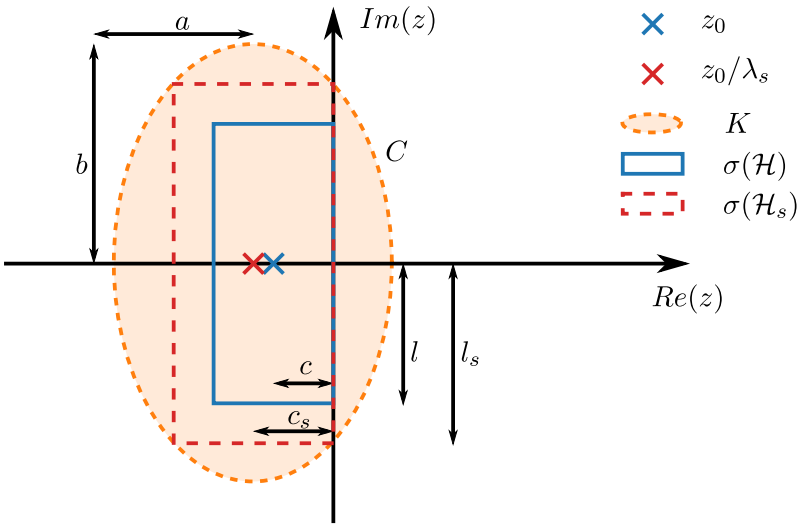


Abbildung 5.4: Das elliptische Konvergenzgebiet K sowie die Skalierung des Eigenwertspektrums $\sigma(\mathcal{H})$ in der komplexen z-Ebene sind mit allen eingeführten Größen illustriert.

Der Skalierungsfaktor λ_s kann mit

$$\lambda_s = \frac{(l^{2/3} + c^{2/3})^{2/3}}{2} \tag{5.28}$$

berechnet werden [91]. Eine weitere Besonderheit des elliptischen Konvergenzgebietes ist, dass sich bei diesem eine analytische Formel für die Bestimmung der Faberpolynom-Entwicklungskoeffizienten c_m finden lässt [88, 90, 91]. Zunächst werden die Operatorfunktion, hier das Matrixexponential, sowie die konforme Abbildung (5.10), in die Berechnungsvorschrift für die Entwicklungskoeffizienten (5.8) eingesetzt. Damit kann (5.8) wie folgt umgeschrieben werden:

$$c_m = \frac{1}{2\pi j} \int\limits_{|w|=R} \frac{\exp(\Delta t_s (w + \gamma_0 + \gamma_1 / w))}{w^{m+1}} \mathrm{d}w. \tag{5.29}$$

Unter Verwendung der erzeugenden Funktion der Bessel-Funktionen erster Ordnung [96], gegeben mit

$$e^{z(t-1/t)/2} = \sum_{n=0}^{\infty} J_n(z) t^n \tag{5.30}$$

kann

$$c_m = \frac{1}{2\pi j} \exp(\Delta t_s \gamma_0) \sum_{n=0}^{\infty} \left(\frac{1}{j\sqrt{\gamma_1}} \right)^n J_n\left(j 2 \Delta t_s \sqrt{\gamma_1}\right) \int\limits_{|w|=R} \frac{1}{w^{m+1-n}} \mathrm{d}w \tag{5.31}$$

bestimmt werden. Hierbei gilt $t = -jw/\sqrt{\gamma}_1$ und $z = j2\Delta t_s \sqrt{\gamma}_1$. Das Integral (5.31) kann mit der Cauchy-Integral-Formel gelöst werden [93]. Dies führt auf eine analytische Berechnungsvorschrift für die Entwicklungskoeffizienten c_m der Faberpolynome [90, 91]:

$$c_m = \exp(\Delta t_s \gamma_0) \left(\frac{1}{j\sqrt{\gamma_1}} \right)^m J_m\left(j 2 \Delta t_s \sqrt{\gamma_1}\right). \tag{5.32}$$

Eine ausführlichere Herleitung ist im Anhang B.3.2 zu finden. Mit diesem Ausdruck können die Koeffizienten für $\rho = 1$ und elliptische Konvergenzbereiche bestimmt werden. Die Bessel-Funktion J_m in (5.32) führt dazu, dass der Absolutbetrag der Entwicklungskoeffizienten mit steigendem m, also mit steigender Polynomordnung, exponentiell abfällt [90, 91]. Hier wird die Entwicklung, wie in [90] vorgeschlagen, bei einem Absolutbetrag von $|c_m| < 10^{-15}$ abgebrochen. Ein weiterer Punkt, welcher hier noch nicht mit berücksichtigt wird, ist die Position der Ellipse. In [90] wird gezeigt, dass sich diese bei der vorliegenden Konfiguration nicht zu weit in der rechten Halbebene befinden sollte, da dies zu einem schlechteren Konvergenzverhalten führen kann [88, 90]. Eine Möglichkeit, um dem Problem entgegenzuwirken, ist die Erhöhung des Parameters l, wie in [90] vorgeschlagen.

Mit diesem Schema lässt sich die Faberpolynom-Entwicklung für ein elliptisches Konvergenzgebiet realisieren. Für die Bestimmung werden allerdings Informationen über das Eigenwertspektrum der Systemmatrix benötigt. Eine direkte Berechnung ist in der Regel nicht möglich und würde, sofern möglich, die Verwendung der Faberpolynome oder sonstiger Abschätzungen unnötig machen, da in diesem Fall die Lösung des Problems bekannt wäre [72]. Für die oben beschriebene Vorgehensweise wird nur eine obere und untere Abschätzung der Real- und Imaginärteile des Eigenwertspektrums benötigt. Hierbei ist es erstrebenswert, eine möglichst genaue Abschätzung zu erlangen, da eine verkleinerte Fläche zu einem geringeren λ_s führt, was über $\Delta t = \Delta t_s / \lambda_s$ direkt größere Zeitschrittweite bei der gleichen Polynomordnung erlaubt. Die Genauigkeit der Approximation des Spektrums wirkt sich deshalb direkt auf die Effizienz des Verfahrens aus. Daher wird im folgenden Abschnitt eine Methode entwickelt, mit der das Spektrum für verschiedenste lineare Materialgleichungen für die Maxwell-Gleichungen im Zeitbereich abgeschätzt werden kann [KS6].

5.3.2 Spektrale Untersuchung und Abschätzung

Im vorangegangenen Abschnitt wird die Faberpolynom-Entwicklung mit einem elliptischen Konvergenzgebiet betrachtet. Um dieses zu bestimmen, wird ein rechteckiges Gebiet benötigt, welches das Eigenwertspektrum der Systemmatrix $\mathcal{H}$ begrenzt. Je präziser das Gebiet das Spektrum begrenzt, desto größere Zeitschritte können mit derselben Polynomordnung erreicht werden [84]. Daher wird im Folgendem die Systemmatrix auf ihre spektralen Eigenschaften untersucht. Ziel soll es hierbei zunächst sein, obere und untere Grenzen für die Real- und Imaginärteile des Spektrums zu bestimmen. Zunächst sollen einige allgemeine Eigenschaften zusammengefasst werden. Als Ortdiskretisierung für die Gleichung (5.1) wird, wie in Abschnitt 3.3 beschrieben, ein FD-Ansatz nach Yee [5] verwendet. Wenn ein dämpfungs- und verstärkungsfreies Medium vorhanden ist, dann liegen alle Eigenwerte auf der imaginären Achse und sind symmetrisch zu null. Wird ein Medium mit Dämpfung modelliert, ergeben sich in der diskretisierten Systemmatrix Eigenwerte in der linken Halbebene. Die Eigenwerte haben daher einen negativen Realteil. Je größer die Dämpfung ist, desto weiter reicht das Spektrum in die linke Halbebene. Im Fall von verstärkenden Medien reicht das Spektrum zusätzlich in die rechte Halbebene. Um eine Abschätzung für die Grenzen zu erlangen, gibt es verschiedene Ansätze. Eine Möglichkeit, welche in [87] untersucht wird, ist die Verwendung eines Algorithmus zur numerischen Bestimmung von Eigenwerten für dünnbesetzte Matrizen wie ARPACK [97]. Hierbei werden nicht alle Eigenwerte bestimmt, sondern nur einzelne. Diese iterativen Löser basieren auf Krylov-Unterraum-Abbildungen und lassen sich mit Matrix-Vektor-Multiplikationen mit

$\mathcal{H}$ berechnen. Mit diesem Ansatz lassen sich die betragsmäßig größten Eigenwerte besonders effizient bestimmen, da die iterative Berechnung schon mit einer geringen Anzahl von Matrix-Vektor-Multiplikationen konvergiert [87]. Werden andere Werte benötigt, kann die Konvergenz deutlich langsamer ausfallen. Dies ist insbesondere der Fall, wenn die Imaginär- und Realteile sehr unterschiedliche Größenordnungen annehmen können. Abhängig von der Systemgröße kann dies eine erhebliche Zeit in Anspruch nehmen. Allerdings müssen die Berechnungen nur einmal im Vorfeld durchgeführt werden. Je nach Simulationszeit T tritt dies in den Hintergrund [87].

In [90, 91] werden Ansätze basierend auf dem numerischen Wertebereich (engl. "Field of Values") und des Rayleigh-Koeffizienten verwendet [98]. Der numerische Wertebereich $W(\mathcal{H})$ einer Matrix $\mathcal{H}$ ist eine weitere Art, die Matrix zu charakterisieren. Entscheidend ist hierbei, dass der numerische Wertebereich auch eine obere Grenze für die Eigenwerte $\sigma(\mathcal{H})$ der Matrix liefert [98, 99]:

$$\sigma(\mathcal{H}) \subseteq W(\mathcal{H}). \tag{5.33}$$

Die Matrix wird in einen hermiteschen und einen schiefhermiteschen Teil aufgeteilt. Anschließend werden mithilfe des numerischen Wertebereichs Grenzen für die beiden Teile bestimmt [91, 99]. Es lässt sich zeigen, dass diese Teilgrenzen auch für den numerischen Wertebereich der gesamten Matrix gelten [99]. Über (5.33) ist damit auch eine Grenze für das Eigenwertspektrum bestimmt. Allerdings liegen die Grenzen für das Spektrum $\sigma(\mathcal{H})$ und den numerischen Wertebereich $W(\mathcal{H})$ nur für normale Matrizen aufeinander. Im allgemeinen Fall ist der numerische Wertebereich nur eine obere Grenze für das Eigenwertspektrum [99]. In [91] führt die Methode auf eine zu konservative Abschätzung der unteren Grenze des Realteils des Spektrums.

Daher soll hier ein alternativer Ansatz, der in [KS6] vorgestellt ist, entwickelt werden. Der Ansatz benötigt mehr Annahmen im Vorfeld, da auch das Simulationsmodell und die Modellgleichungen in Betracht gezogen werden und nicht nur mit der finalen diskretisierten Systemmatrix gearbeitet wird. Hierzu werden die einzelnen Modellgleichungen durch die Entwicklung nach ebenen Wellen in den k-Raum transformiert. Da hier die äußeren Grenzen des Eigenwertspektrums bestimmt werden sollen, sind bei der Untersuchung nur die Maximalwerte interessant. Die Methode erlaubt in einigen Fällen sogar die Bestimmung von analytischen Grenzen, sofern die Anzahl der Modellgleichungen nicht zu groß ist. Für andere Fälle wird ein numerischer Lösungsansatz vorgestellt.

Ein ähnlicher Ansatz wird in [100, 101] verfolgt, um die Stabilität von PMLs zu untersuchen. Außerdem wird nur eine geringe Anzahl von Gleichungen betrachtet, sodass eine analytische Lösung noch möglich ist. Darüber hinaus werden ähnliche Ansätze auch bei Stabilitätsuntersuchungen von FDTD-Methoden verwendet [4, 42]. Dieser Ansatz soll im Folgenden beleuchtet werden. Zuerst wird das Modell nach ebenen Wellen in den einzelnen Materialbereichen entwickelt. Hierbei wird angenommen, dass die Materialbereiche stückweise konstant sind. Auf diese Weise soll der Einfluss der einzelnen Bereiche auf die Grenzen des Eigenwertspektrums untersucht werden. Bei den einzelnen Materialbereichen kann es sich zum Beispiel um Drude- oder Lorentz-Modelle handeln oder um Bereiche, welche nur durch ein frequenzunabhängiges ϵ oder μ beschrieben werden. Auch die PML-Regionen werden einzeln untersucht. Das Vorgehen soll am Beispiel eines eindimensionalen Systems mit einem Drude-Modell näher erläutert werden. Das Drude-Modell wird mit der in Abschnitt 2.2 vorstellten ADE-Methode eingebunden. Das

System kann in diesem Fall mit

$$\frac{\partial}{\partial t}\vec{\psi} = \begin{bmatrix} 0 & -\frac{1}{\epsilon_\infty(z)}\frac{\partial}{\partial z} & -1/\epsilon_\infty(z) \\ -\frac{1}{\mu(z)}\frac{\partial}{\partial z} & 0 & 0 \\ \epsilon_0\omega_D^2(z) & 0 & -\gamma_D(z) \end{bmatrix} \vec{\psi} \tag{5.34}$$

angegeben werden, während $\vec{\psi}$ mit

$$\vec{\psi} = \begin{bmatrix} E_x(t,z) & H_y(t,z) & J_x(t,z) \end{bmatrix}^{\mathrm{T}} \tag{5.35}$$

gegeben ist. Im Anschluss wird das System nach ebenen Wellen entwickelt. Hierbei müssten alle Größen transformiert werden, welche ortsabhängig sind. Das führt dazu, dass bei direkter Anwendung auf das komplette Modell, abhängig von der räumlichen Struktur des Simulationsgebietes, sehr komplexe Ausdrücke gefunden werden müssen, da auch die Strukturen transformiert werden müssten. Daher wird, wie oben beschrieben, angenommen, dass die Materialbereiche einzeln untersucht werden und als stückweise stetig angenommen werden. Das erlaubt es, die Materialparameter als Konstanten anzunehmen, was die Transformation erleichtert. Denn nun müssen nur noch die Feldgrößen und die Ortsdifferenziale betrachtet werden. Für die übrigen Größen werden nur die Extremwerte betrachtet, da allein die Grenzen des Spektrums untersucht werden sollen. Für ϵ und μ werden die kleinsten vorkommenden Werte verwendet, da in diesen Bereichen die höchste Phasengeschwindigkeit vorliegt. Diese bestimmt die Ausdehnung des Eigenwertspektrums entlang der imaginären Achse. Für γ_D und ω_D werden die größten vorliegenden Werte verwendet. Wenn der Bereich mit dem Drude-Modell nicht den gesamten Simulationsbereich einnimmt, muss der andere Bereich noch separat betrachtet werden. Für den Bereich mit dem Drude-Modell kann die folgende transformierte Matrix bestimmt werden:

$$\mathcal{H}_k = \begin{bmatrix} 0 & -\frac{1}{\epsilon_\infty} j k_z & -1/\epsilon_\infty \\ -\frac{1}{\mu} j k_z & 0 & 0 \\ \epsilon_0\omega_D^2 & 0 & -\gamma_D \end{bmatrix}. \tag{5.36}$$

Bei k_z handelt es sich hierbei um die z Komponente des numerischen Wellenvektors. Nun soll mithilfe des transformierten Systems der Einfluss auf das Eigenspektrum der gesamten Matrix abgeschätzt werden. Zu diesem Zweck werden die Eigenwerte der Matrix $\mathcal{H}_k$ untersucht. Hierzu werden hier zwei Ansätze betrachtet. Wenn das System, wie in (5.36), mit nur einer geringen Anzahl von Gleichungen beschrieben wird, kann in vielen Fällen ein analytischer Ausdruck für die Eigenwerte von $\mathcal{H}_k$ gefunden werden. Gelingt dies, muss im Anschluss noch die örtliche Diskretisierung des Simulationsmodells berücksichtigt werden. Bei der hier verwendeten FD-Diskretisierung nach Yee [5] können die Wellenvektoren nicht beliebige Werte annehmen. Durch die finite Schrittweite der Ortsdiskretisierung gibt es eine maximale Wellenzahl, welche von dem Gitter dargestellt werden kann. Der maximale Wert für den Betrag des Wellenvektors $|k|$ ist mit $|k_{\max}| = 2/\Delta$ gegeben. Die Größe Δ ist die örtliche Diskretisierungsweite. Die Komponenten k_x, k_y und k_z des Wellenvektors in $\mathcal{H}_k$ können daher nur Werte $k \in [-k_{\max}, k_{\max}]$ annehmen [KS6]. Auf diesem Intervall lassen sich nun die Minima und Maxima für den Realteil und Imaginärteil des Spektrums bestimmen. Auf diese Weise können mithilfe des analytischen Ausdrucks die Grenzen des Spektrums berechnet werden. Für einige Modelle sind darüber hinaus analytische Ausdrücke für die Grenzen des Spektrums zu finden. Ein Beispiel hierfür sind die dämpfungsfreien Maxwell-Gleichungen. Hier wird die Systemmatrix zunächst mit dem Ansatz in [53] normalisiert. Im Anschluss ist die resultierende Matrix schiefsymmetrisch. Das System

hat also nur imaginäre Eigenwerte. Nach Bestimmung der Ausdrücke für die Eigenwerte werden die Maximalwerte $k_{\max}$ in diese eingesetzt. Damit lässt sich für den eindimensionalen Fall

$$l = \frac{2}{\sqrt{\mu\epsilon}\Delta z} \tag{5.37}$$

als obere Grenze für den Imaginärteil bestimmen. Im zweidimensionalen Fall ist die Grenze mit

$$l = \frac{2}{\sqrt{\mu\epsilon}}\sqrt{\left(\frac{1}{\Delta x}\right)^2 + \left(\frac{1}{\Delta y}\right)^2} \tag{5.38}$$

gegeben und für den dreidimensionalen Fall mit

$$l = \frac{2}{\sqrt{\mu\epsilon}}\sqrt{\left(\frac{1}{\Delta x}\right)^2 + \left(\frac{1}{\Delta y}\right)^2 + \left(\frac{1}{\Delta z}\right)^2}. \tag{5.39}$$

Die Größe der Matrix $\mathcal{H}_k$ kann abhängig von den verwendeten Materialmodellen allerdings schnell ansteigen. Besonders im zwei- und dreidimensionalen Fall hat die Matrix $\mathcal{H}_k$ oft viele Terme, was die Bestimmung eines analytischen Ausdrucks erschwert. Wird zum Beispiel die in Abschnitt 2.4 beschriebene CFS-PML verwendet und mit ADEs eingebunden, so liegen bei eindimensionalen Systemen vier Gleichungen vor, sodass $\mathcal{H}_k$ auch eine Größe von vier hat. Im dreidimensionalen Fall liegen hier 12 zusätzliche ADEs vor, sodass das System insgesamt von 18 Gleichungen modelliert wird. Während es für Matrizen mit einer Größe von vier oder kleiner noch direkte Methoden gibt, um die Eigenwerte analytisch zu bestimmen, ist dies ab einer Größe von fünf nicht mehr möglich. Eine Möglichkeit für solche Systeme ist der Einsatz von symbolischen Lösungsalgorithmen, um einen analytischen Zusammenhang zu bestimmen. Für sehr komplexe Modelle kann auch ihr Einsatz nicht immer zu einem Ergebnis führen. Deshalb wird hier als zweiter Lösungsansatz eine numerische Methode vorgestellt, welche selbst für diese Systeme eine Abschätzung liefern kann.

Der Ausgangspunkt sind jeweils die Systemmatrizen $\mathcal{H}_k$ der einzelnen Bereiche im k-Raum. Werden einzelne Werte für k_z eingesetzt, so ist die Matrix $\mathcal{H}_k$ vollständig bestimmt. In diesem Fall lassen sich die Eigenwerte der Matrix mit einem direkten Algorithmus bestimmen. Auch für sehr komplexe Modelle ist die Anzahl der Gleichungen nicht so groß, dass sie von einem numerischen Eigenwertlöser nicht mehr zu lösen wären. Nun soll wieder genutzt werden, dass die örtliche Diskretisierung die Werte für die Komponenten k_x, k_y und k_z des Wellenvektors einschränkt. Nun werden die Komponenten des Wellenvektors k_x, k_y und k_z in $\mathcal{H}_k$ gleichförmig in $k \in [-k_{\max}, k_{\max}]$ diskretisiert. Hierzu hat sich eine Anzahl von 50 Abtastpunkten als ausreichend erwiesen [KS6]. Wichtig ist es insbesondere, die Ränder des Intervalls $k \in [-k_{\max}, k_{\max}]$ und null in die Berechnung einzuschließen. Diese Werte korrespondieren in der Regel mit Extremwerten an den Rändern des Eigenwertspektrums. Für die einzelnen Werte k werden die Eigenwerte der Matrix $\mathcal{H}_k$ numerisch bestimmt. Durch die geringe Größe von $\mathcal{H}_k$ ist der hierzu nötige Rechenaufwand gering. Die so erlangten maximalen Eigenwerte werden zur Abschätzung der Ränder des Spektrums von $\mathcal{H}$ verwendet.

5.4 Untersuchung der Effizienz

In den vorangegangenen Abschnitten wird ein Algorithmus zur numerischen Lösung der Maxwell-Gleichungen vorgestellt. Die bisherigen Überlegungen erlauben eine Lösung bei Systemen, welche

sich in der Form (5.1) darstellen lassen. Mit diesem Ansatz lassen sich alle linearen Materialmodelle erfassen. Für die Approximation mit den Faberpolynomen werden allerdings Informationen über das Eigenwertspektrum der Systemmatrizen benötigt. Mit den zuvor vorgestellten Methoden steht auch hierfür ein Ansatz bereit, welcher für beliebige lineare Materialmodelle mit einer FD-Yee-Diskretisierung angewendet werden kann. Nun verbleibt die Frage nach der Effizienz des Algorithmus im Hinblick auf den Rechenaufwand und die erreichte Genauigkeit der Approximation. Daher soll der Faberpolynom-Algorithmus im Folgenden mit der klassischen FDTD-Methode verglichen werden. Das ist in der Literatur bisher noch nicht untersucht. Hierbei soll primär die Approximation der Zeitpropagation in den Blick genommen werden. Aus diesem Grund wird für beide Algorithmen bei allen Vergleichen immer dasselbe örtliche Gitter verwendet.

Die Referenzlösung wird mit einer Faberpolynom-Approximation von sehr hoher Ordnung bestimmt. Als Maß für den Rechenaufwand wird die Anzahl der Matrix-Vektor-Multiplikationen herangezogen. Sofern für die örtliche Diskretisierung das gleiche Verfahren verwendet wird, so entspricht ein Zeitschritt der FDTD-Methode einer Matrix-Vektor-Multiplikation [53]. Insbesondere für zwei- und dreidimensionale Systeme wird die Rechenzeit für die verwendete kurze Rekursionsbeziehung (5.12) der Faberpolynome primär von den Matrix-Vektor-Multiplikationen in Anspruch genommen. Es sollen zwei Fälle betrachtet werden: Zuerst soll die Effizienz für ein dämpfungsfreies System untersucht werden. Im Anschluss wird ein System mit Verlusten betrachtet. Hierbei ist es von Interesse, ob die Faberpolynom-Approximation auch hier genaue Ergebnisse liefern kann.

5.4.1 Dämpfungsfreies System

Für die Untersuchung wird ein zweidimensionales Testsystem gewählt, bei dem die TM-Mode in der x-y Ebene untersucht wird. Der Simulationsbereich hat die Abmessungen $L_x = L_y = 5\,\mu\text{m}$. Für die Ortsdiskretisierung mit der FD-Yee-Methode werden die Diskretisierungsweiten $\Delta = \Delta x = \Delta y = 10\,\text{nm}$ verwendet. Der Koordinatenmittelpunkt $\vec{r} = (x, y)^T = (0, 0)^T$ befindet sich in der Mitte des Rechengebietes. Das Rechengebiet wird in zwei Hälften aufgeteilt. Auf der linken Seite des Rechengebietes für $x < 0$ liegt eine relative Permittivität von $\epsilon_{r,1} = 1$ vor. In der Mitte des Rechengebietes verläuft eine Sprungstelle in ϵ parallel zur y Achse. Auf der linken Seite dieser Sprungstelle mit $x > 0$ gilt $\epsilon_{r,2} = 2$. Die Simulation wird mit einer Feldverteilung $\vec{\Psi}(t = 0)$ initialisiert. Bei dieser handelt es sich um einen gaußförmigen Impuls, welcher in der linken Halbebene platziert wird. Die Feldverteilung wird so initialisiert, dass der Impuls in positiver x-Richtung in Richtung der rechten Halbebene propagiert. Die FWHM-Bandbreite des Impulses wird zu $B = 300\,\text{THz}$ gewählt. Die Simulationszeit beträgt $T = 0{,}047\,\text{ps}$. Für die Auswertung wird der Wert der E_z-Komponente an der Stelle $\vec{r}_p = (x_p, y_p)^T = (1{,}25\,\mu\text{m}, 0\,\mu\text{m})^T$ in Abhängigkeit von der Zeit gemessen. Diese initialen Parameter werden sowohl für die FDTD-Simulationen als auch für die Faberpolynom-Methode verwendet. Bei der FDTD-Methode ist bei der Initialisierung der Feldgrößen neben dem örtlichen Versatz zwischen den Feldgrößen E_z, H_x und H_y durch das Yee-Gitter auch der zeitliche Versatz berücksichtigt worden, welcher bei der FDTD-Methode zwischen den diskretisierten Größen vorliegt.

Für die Faberpolynom-Methode wird das elliptische Konvergenzgebiet verwendet. Mit der Methode zur Abschätzung des Eigenwertspektrums ergibt sich $l = 8{,}479 \times 10^{16}\,\text{s}^{-1}$ und $c = 0$. Die Simulation wird mit den oben genannten Parametern für verschiedene Zeitschrittweiten Δt

durchgeführt. Um die Einordnung zu erleichtern, wird die Zeitschrittweite auf Δt_{CFL} normiert:

$$\Delta t = q \Delta t_{\mathrm{CFL}}. \tag{5.40}$$

Für die Propagation werden abhängig von der verwendeten Zeitschrittweite $N_T = T/\Delta t$ Zeitschritte benötigt, um die Simulation durchzuführen. Der Rechenaufwand hierfür wird hier über die benötigte Anzahl von Matrix-Vektor-Multiplikationen N_{MatVec} charakterisiert. Wie oben beschrieben, hängt die Anzahl pro Zeitschritt bei der Faberpolynom-Methode von der Polynomordnung N_{pol} ab. Damit gilt also:

$$N_{\mathrm{MatVec,Faber}} = T/\Delta t N_{\mathrm{pol}}. \tag{5.41}$$

Um den Fehler in Abhängigkeit des Rechenaufwandes zu charakterisieren, wird die Simulation für jeden verwendeten Zeitschritt Δt mit verschiedenen Entwicklungsordnungen N_{pol} und damit gemäß (5.41) einer verschiedenen Anzahl von Matrix-Vektor-Multiplikationen durchgeführt. Auch der FDTD-Algorithmus wird mit verschiedenen Zeitschrittweiten Δt berechnet. Wie oben beschrieben, entspricht der Rechenaufwand für einen Zeitschritt mit dem FDTD-Algorithmus einer Matrix-Vektor-Multiplikation mit der Matrix $\mathcal{H}$. Die Anzahl der Matrix-Vektor-Multiplikationen N_{MatVec} entspricht daher der Anzahl der Zeitschritte, welche benötigt werden, um die Simulation der Länge T durchzuführen:

$$N_{\mathrm{MatVec,FDTD}} = N_T = T/\Delta t. \tag{5.42}$$

Die Simulation mit der FDTD-Methode wird mit verschiedenen Zeitschrittweiten Δt durchgeführt, wodurch gemäß (5.42) der Rechenaufwand in Form der Matrix-Vektor-Multiplikationen variiert. Aufgrund der CFL-Bedingung muss hierbei $\Delta t < \Delta t_{\mathrm{CFL}}$ erfüllt sein, um eine stabile Propagation zu gewährleisten.

Die Ergebnisse der Simulationen sind in Abbildung 5.5 zusammengefasst. Die Abbildung zeigt den relativen Fehler der Zeitverläufe der E_z-Komponente, welcher in $\vec{r}_p$ aufgenommen worden ist. In der Untersuchung sollen Simulationen mit Zeitschrittweiten Δt von sehr unterschiedlichen Größenordnungen verglichen werden. Um dies zu ermöglichen, wird der Fehler im Spektrum der gemessenen Zeitverläufe untersucht. Hierzu wird die DFT $\mathcal{F}_D(E_z)$ der Zeitverläufe bestimmt. Der Fehler wird in einem Spektralbereich von $f = 0\,\mathrm{THz}$ bis $f = f_{\mathrm{thres}} = 1000\,\mathrm{THz}$ berechnet, welcher das gesamte Signalspektrum enthält. Da die Simulationszeit T konstant ist und immer der gleiche Spektralbereich gewählt wird, wird bei jeder Zeitschrittweite Δt im diskreten Spektrum immer dieselbe Anzahl N_k von diskreten Spektralkomponenten für die Fehlerberechnung verwendet. Der relative Fehler wird daher jeweils im Spektrum $\mathcal{F}_D(E_z)$ der Zeitverläufe mit $\epsilon_{rel} = 1/N_k \sum_{n=1}^{N_k} |\mathcal{F}_D(E_{z,ref}(\vec{r} = \vec{r}_p, f = f_n)) - \mathcal{F}_D(E_z(\vec{r} = \vec{r}_p, f = f_n))|/|\mathcal{F}_D(E_{z,ref}(\vec{r} = \vec{r}_p, f = f_n))|$ bestimmt. Der Fehler der Faberpolynom-Methode ist für alle verwendeten Zeitschrittweiten für geringe Werte N_{MatVec} hoch. Dies ist darin begründet, dass die Approximation in diesen Fällen noch nicht konvergiert ist, da die Entwicklungsordnung N_{pol} noch nicht ausreichend hoch ist. In vielen Fällen ist in diesem Bereich keine stabile Propagation möglich. Die Simulationen werden instabil und divergieren. Wird N_{MatVec} und damit auch die Entwicklungsordnung N_{pol} erhöht, so konvergiert die Methode. Dies hat zur Folge, dass der Fehler sehr schnell abfällt. Dieser Abfall des relativen Fehlers setzt sich fort, bis der Fehler einen finalen Wert erreicht hat. Hier liegen diese finalen Werte bei näherungsweise $\approx 1 \cdot 10^{-13}$ bis $\approx 9 \cdot 10^{-13}$. Sobald der Wert erreicht ist, führt eine Erhöhung der Approximationsordnung zu keiner weiteren Steigerung der

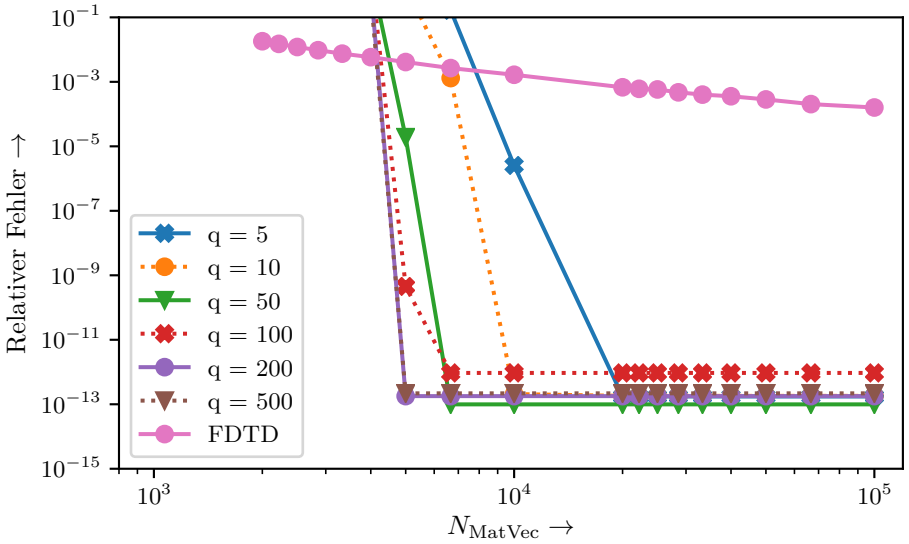


Abbildung 5.5: Der relative Fehler ist für die verschiedenen Algorithmen in Abhängigkeit von dem Rechenaufwand dargestellt, welcher benötigt wird, um die jeweilige Simulation durchzuführen. Der Rechenaufwand wird mit der Anzahl der benötigten Matrix-Vektor-Multiplikationen N_{MatVec} charakterisiert. Die Faberpolynom-Methode wird für sechs verschiedene Schrittweiten durchgeführt, bei welchen jeweils die Entwicklungsordnung N_{pol} variiert wird.

Genauigkeit. Auffällig ist außerdem, dass höhere Zeitschrittweiten zu einer schnelleren Konvergenz führen, da für große Zeitschrittweiten der Fehler früher einen Sättigungswert erreicht. Bei der FDTD-Methode ist ein anderer Verlauf zu beobachten. Die untere Grenze für den Rechenaufwand wird durch $N_{\mathrm{MatVec}} = N_T = T/\Delta t_{\mathrm{CFL}}$ begrenzt. Für größere Anzahlen der Zeitschritte N_T ist eine stabile Propagation möglich. Wird N_T und damit auch N_{MatVec} weiter erhöht, so sinkt gemäß (5.42) die Zeitschrittweite Δt. Ein geringeres Δt hat zur Folge, dass die Zeitpropagation genauer approximiert wird. Dies spiegelt sich auch in dem Verlauf in Abbildung 5.5 wider. Mit steigendem N_{MatVec} sinkt der Fehler der FDTD-Methode kontinuierlich. Werden die Methoden miteinander verglichen, so fällt auf, dass der FDTD-Algorithmus für niedrige Genauigkeiten effizienter ist. Mit steigendem Rechenaufwand N_{MatVec} wird zwar der Fehler von allen Algorithmen reduziert, allerdings sinkt der Fehler der Faberpolynom-Methode mit Einsetzen der Konvergenz so schnell, dass dieser die Kurve des FDTD-Algorithmus schneidet. Nach dem Schnittpunkt ist die Faberpolynom-Methode effizienter. Die schnelle Konvergenz führt außerdem dazu, dass die Faberpolynom-Methode Genauigkeiten erreicht, welche mit der FDTD-Methode nur mit sehr großem Rechenaufwand und praktisch in vielen Fällen gar nicht zu erreichen sind.

Die Schnittpunkte liegen in dem Bereich vor, in denen die Genauigkeitsanforderungen einen FDTD-Zeitschritt von $\Delta t = 0,5\Delta t_{\mathrm{CFL}}$ bis $\Delta t = 0,2\Delta t_{\mathrm{CFL}}$ erfordert. Darüber hinaus lässt sich feststellen, dass selbst Zeitschrittweiten ohne Probleme mit der Faberpolynom-Methode möglich sind, welche mit $\Delta t = 500\Delta t_{\mathrm{CFL}}$ mehrere hundertmal größer sind als es die CFL-Bedingung erlaubt.

5.4.2 Drude-Modell

Im Folgenden sollen die Algorithmen für ein System mit Dämpfung untersucht werden. Während die Berechnung des obigen Beispiels auch mit dem klassischen Tschebyscheff-Polynom-Ansatz von [53] möglich wäre, ist die folgende Untersuchung mit diesem nicht ohne weiteres möglich. Für das neue Beispiel wird zunächst das Testsystem variiert. Die rechte Halbebene für $x > 0$ ist mit einem Medium gefüllt, welches mithilfe eines Drude-Modells modelliert wird. Hierbei wird wieder die ADE-Formulierung verwendet, welche in Kapitel 2.2 beschrieben wird. Diese ADE-Formulierung wird sowohl für die FDTD-Methode als auch für die Faberpolynom-Methode verwendet, sodass sowohl das Materialmodell als auch die Ortsdiskretisierung übereinstimmen. Die folgenden Parameter werden verwendet: $\gamma_D = 3{,}23 \times 10^{13}\,\mathrm{s}^{-1}$ und $\omega_D = 1{,}39 \times 10^{15}\,\mathrm{s}^{-1}$. Durch den Parameter γ_D liegt Dämpfung in dem System vor. Die Systemmatrix $\mathcal{H}$ ist nicht länger rein schiefsymmetrisch und hat nicht nur rein imaginäre Eigenwerte. Mir den oben beschriebenen Methoden lassen sich die folgenden Parameter für das Spektrum bestimmen: $c = 16{,}15 \times 10^{12}\,\mathrm{s}^{-1}$ und $l = 8{,}479 \times 10^{16}\,\mathrm{s}^{-1}$. Die Ergebnisse der Simulationen sind in Abbildung 5.6 dargestellt. Die Verläufe der Kurven für das dämpfungsbehaftete System stimmen gut mit

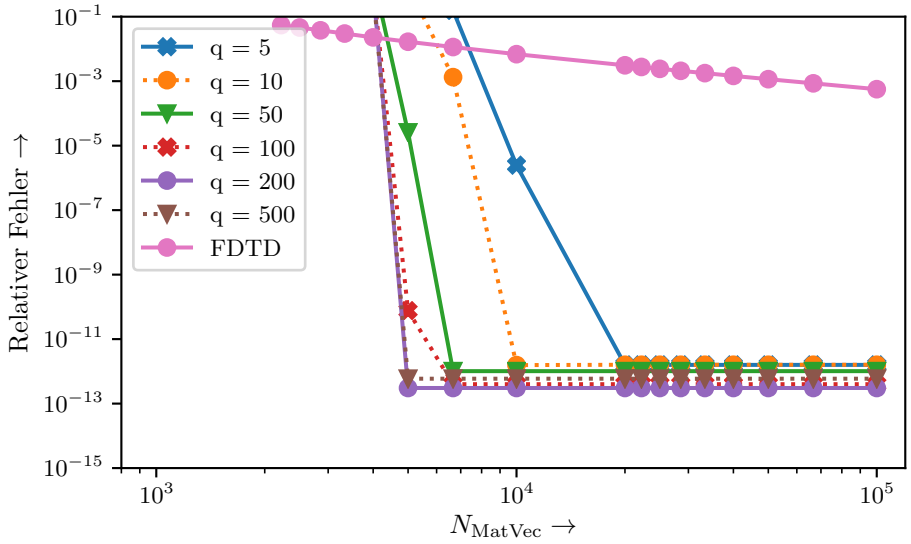


Abbildung 5.6: Es wird ein System simuliert, bei dem ein Halbraum mit einem nach dem Drude-Modell modellierten Material gefüllt ist. Der relative Fehler der verschiedenen Algorithmen wird in Abhängigkeit von dem Rechenaufwand gemessen und in Abhängigkeit von der Anzahl der nötigen Matrix-Vektor-Multiplikationen N_{MatVec} dargestellt. Die Faberpolynom-Methode wird mit verschiedenen Zeitschrittweiten durchgeführt.

denen des dämpfungsfreien Systems in 5.5 überein. Der Fehler der FDTD-Methode sinkt wieder kontinuierlich mit sinkender Zeitschrittweite, während der Fehler für die Faberpolynom-Methode mit Einsetzen der Konvergenz rapide auf Sättigungswerte im Bereich $\approx 3 \cdot 10^{-13}$ bis $\approx 15 \cdot 10^{-13}$ abfällt. Wie in dem zuvor betrachteten Beispiel setzt die Konvergenz der Faberpolynom-Methode bei den höheren Schrittweiten eher ein, als bei den niedrigeren Schrittweiten Δt. Auch hier sind, bezogen auf die maximale Schrittweite, nach der CFL-Bedingung Δt_{CFL} wieder extrem große

Schrittweiten mit der Faberpolynom-Methode möglich.

5.5 Diskussion

In den letzten Abschnitten wird eine Faberpolynom-Methode zur Lösung der Maxwell-Gleichungen vorgestellt und untersucht. Die Methode basiert auf einer Approximation des Exponential-Operators in (5.1). Die Approximation erfolgt hierbei in der komplexen Ebene in Abhängigkeit von den Eigenwerten der Systemmatrix. Diese Approximation ermöglicht es, beliebige lineare Materialmodelle in die Simulation aufzunehmen. Hierbei können auch dämpfende Materialmodelle direkt in die Systemmatrix $\mathcal{H}$ integriert werden. Die vorgestellte Methode zur Approximation der Grenzen des Eigenwertspektrums ermöglicht die Bestimmung der für die Faberpolynom-Approximation nötigen Parameter. Hierbei müssen an keiner Stelle Eigenwerte der in der Regel sehr großen Systemmatrix $\mathcal{H}$ direkt bestimmt werden.

In Abschnitt 5.4 wird die Faberpolynom-Methode im Hinblick auf ihre Effizienz mit dem klassischen FDTD-Algorithmus verglichen. Hierbei ist die benötigte Anzahl von Matrix-Vektor-Multiplikationen N_{MatVec} das Maß für den Rechenaufwand. Da für beide Methoden die gleiche örtliche Diskretisierung verwendet wird, erlaubt diese Betrachtung eine allgemeine und von der Implementierung unabhängige Aussage. Wie in Abschnitt 5.4 untersucht, ermöglicht die Faberpolynom-Methode eine sehr genaue Approximation der Zeitpropagation. Hervorzuheben ist, dass dies sowohl für das dämpfungsfreie als auch für das dämpfungsbehaftete System der Fall ist. Die Konvergenz setzt bei der Faberpolynom-Methode mit steigender Approximationsordnung abrupt ein und führt dann zu einem raschen Sinken des Fehlers. Werden Approximationen mit einer zu geringen Approximationsordnung N_{pol} eingesetzt, so ergeben sich große Fehlerwerte und mitunter instabile Simulationen. In diesem Bereich ist die FDTD-Methode effizienter. Insgesamt zeigt sich, dass die Faberpolynom-Methode bei hohen Genauigkeitsanforderungen eine um Größenordnungen effizientere Berechnung ermöglicht. Sind die Anforderungen an die Genauigkeit geringer, so ist die klassische FDTD-Methode effizienter.

Hervorzuheben ist, dass hier die Approximation der Zeitpropagation berücksichtigt wird. Der gesamte Fehler setzt sich, wie in Abschnitt 3.1 und 3.2 beschrieben, aus dem Fehler durch die Ortsdiskretisierung und die Zeitdiskretisierung zusammen. Um den Fehler durch die Ortdiskretisierung weiter zu senken, kann entweder die Diskretisierungsweite reduziert oder ein Verfahren höherer Ordnung verwendet werden. Hierbei bieten sich insbesondere pseudospektrale Verfahren [4, 14, 42], wie auch in 3.1 beschrieben, oder diskontinuierliche Galerkin-Verfahren an [17].

Ansätze zur weiteren Verbesserung der Approximation können an dieser Stelle durch eine weiter optimierte spektrale Approximation erreicht werden. Bei der hier vorgestellten Methode werden die Grenzwerte des Spektrums bestimmt. Mehr Informationen über die Form des Spektrums können bei der Faberpolynom-Methode durch einen genau angepassten Konvergenzbereich effektiv genutzt werden. Die nötigen konformen Abbildungen lassen sich mit der Schwarz-Christoffel-Transformation bestimmen.

Bisher werden nur lineare quellfreie Systeme betrachtet. Im nächsten Kapitel sollen Methoden entwickelt werden, die es erlauben, mit dem Algorithmus auch Quellterme wie Stromdichten zu berücksichtigen.

6 Einbindung von Quelltermen

Mit den Überlegungen aus dem vorangegangenen Abschnitt lässt sich ein Algorithmus für die numerische Lösung der Maxwell-Gleichungen im Zeitbereich mithilfe von Faberpolynomen realisieren. Dies bedeutet, dass alle Feldverteilungen zu Beginn der Zeitbereichs-Simulation im Simulationsgebiet initialisiert werden müssen. Mit der Methode können Probleme untersucht werden, bei denen die Entwicklung einer bekannten Feldverteilung über die Zeit berechnet werden soll. Allerdings lassen sich mit der Formulierung des letzten Kapitels noch keine Quellterme in die Simulation einbinden. Das schränkt die Anwendbarkeit der Methode deutlich ein. Die Quellterme sind nötig, um die Einkopplung von Feldern während der Simulation zu ermöglichen. Mit dem Begriff Quellterme sind hierbei die Stromdichten in (2.1) zusammengefasst. Mit der Methode in 2.5 lässt sich mithilfe der Stromdichten die Einkopplung von Ebenen Wellen, die Einkopplung von Eigenmoden an Wellenleitern oder auch TFSF-Formulierungen realisieren.

Die Einbindung von Quelltermen ist auch in [90] und [91] noch nicht untersucht worden. Daher sollen Methoden vorgestellt werden, welche dies effizient ermöglichen. Die Formulierung soll möglichst konsistent mit dem bisherigen Zeitpropagationsalgorithmus sein und keine zusätzlichen Fehler einführen, wie es beispielsweise bei Splitting-Ansätzen der Fall ist [102]. Insgesamt ist eine möglichst präzise Approximation erstrebenswert, um die hohe Genauigkeit des linearen Teils zu erhalten. Wie im Folgenden erläutert, kann die Einbindung von Quelltermen allerdings zu einem erheblichen zusätzlichen Rechenaufwand führen.

Daher werden Methoden entwickelt, welche die Beschaffenheit der Quellterme im Hinblick auf Orts- und Zeitabhängigkeit berücksichtigen. Zunächst wird hierzu die Formulierung der Quellterme für die Einbindung in die Simulation beschrieben. Im Anschluss werden verschiedene effiziente Strategien zur Realisierung diskutiert.

6.1 Direkte Evaluation der Quellterme

Zuerst wird die bisherige lineare, quellfreie Formulierung um die Quellfunktionen ergänzt. Im Anschluss wird das angepasste Zeitpropagationsschema formuliert. Darauf aufbauend werden verschiedene alternative Methoden diskutiert, mit denen die Quellterme approximiert werden können. Auf dieser Basis wird im Anschluss eine erste Approximation des Zeitpropagationsschemas mit Quelltermen auf Basis von Faberpolynomen entwickelt. Dieses soll an einem Testsystem numerisch evaluiert und im Anschluss bewertet werden.

Teile der im Anschluss beschriebenen Untersuchung sind in [KS8] veröffentlicht. Die Ergebnisse werden im Folgenden dargestellt und ergänzt.

6.1.1 Formulierung

Die Quellterme sollen hier allgemein mit dem in (2.54) bereits eingeführten $\zeta(\vec{r},t) = \begin{bmatrix}\vec{J} & \vec{K}\end{bmatrix}^T$ dargestellt werden. Diese Funktion $\zeta(\vec{r},t)$ kann im Allgemeinen sowohl orts- als auch zeitabhängig sein. Zunächst wird das System örtlich diskretisiert. Die diskretisierten Gleichungen lassen sich wie folgt darstellen:

$$\frac{\partial\vec{\Psi}(t)}{\partial t} = \mathcal{H}\vec{\Psi}(t) + \vec{\xi}(t). \tag{6.1}$$

Der Vektor $\vec{\xi} \in \mathbb{R}^N$ beinhaltet die örtlich, aber noch nicht zeitlich diskretisierten Stromdichten und ist mit

$$\vec{\xi}(t) = \begin{bmatrix}-\vec{J}(t)/\epsilon \\ -\vec{K}(t)/\mu\end{bmatrix} \tag{6.2}$$

gegeben. Nun soll wie im vorangegangenen Abschnitt die formale Lösung bestimmt werden. Mit den allgemein angesetzten Quellterm ergibt sich diese zu [71, 103]:

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t\mathcal{H})\vec{\Psi}(t_n) + \int_0^{\Delta t} \exp((\Delta t - \tau)\mathcal{H})\vec{\xi}(t_n + \tau)d\tau. \tag{6.3}$$

Hierbei fällt auf, dass die Gleichung für den linearen quellfreien Teil noch in derselben Form vorliegt wie zuvor ohne den Quellterm in (5.1). Neben dem Matrixexponential, welches den linearen Teil der Gleichung beschreibt, liegt nun zusätzlich ein Integralterm vor. Physikalisch beschreibt dieser Term nicht nur den Verlauf der Stromdichten, sondern auch die Interaktion mit dem umgebenden Medium für die Dauer des Zeitschritts Δt. Sofern $\vec{\xi}(t)$ nicht zeitunabhängig ist, muss der Integralterm in jedem Zeitschritt bestimmt werden.

Einordnung

In der Literatur werden hierzu verschiedene Ansätze diskutiert. Diese sollen an dieser Stelle kurz beschrieben werden. Besonders die Genauigkeit und der nötige Rechenaufwand sollen in den Blick genommen werden. Außerdem soll die Eignung für die verwendete Faberpolynom-Methode berücksichtigt werden.

Eine Möglichkeit ist die Anwendung von klassischen Quadraturregeln zur Berechnung des Integrals in (6.3). In [71] wird dieser Ansatz allgemein im Kontext der Faberpolynome untersucht. Hierbei werden keine weiteren Annahmen bezüglich der Funktionen in dem Integral getroffen. Dadurch ist dieser Ansatz flexibel in der Anwendung, da keine Einschränkungen im Hinblick auf die Zeitabhängigkeit oder die örtliche Verteilung der Quellfunktionen im Rechengebiet vorliegen. Die Anwendung der Quadraturregeln führt auf zusätzliche Matrixfunktionen. Diese müssen, wie das Matrixexponential für den quellfreien Teil von (6.3), auf geeignete Weise berechnet werden. Die Anzahl dieser zusätzlichen Funktionen ist durch die Approximationsordnung der Quadratur bestimmt. Je höher die Approximationsordnung ist, desto mehr zusätzliche Terme müssen bestimmt werden. Die Anforderung an die Approximationsordnung hängt in diesem Zusammenhang von den Anforderungen an die Genauigkeit und von der Zeitabhängigkeit der Funktion $\zeta(\vec{r},t)$ ab. Für Funktionen $\zeta(\vec{r},t)$, welche stark oszillieren, sind daher in der Regel viele zusätzliche Funktionen nötig. Da diese Matrixfunktionen während der Zeitpropagation berechnet werden müssen, führt dies zu einem stark erhöhten Rechenaufwand. Die Verwendung

einer Entwicklung von einer zu geringen Ordnung führt zu einer Reduzierung der Genauigkeit im Vergleich zu der genauen Approximation des quellfreien Teils.

Weitaus effizientere Algorithmen können konstruiert werden, indem $\vec{\xi}$ zunächst geeignet approximiert wird, die Approximation in das Integral eingesetzt und im Anschluss aufgeteilt und berechnet wird [103, 104]. Im Umfeld der Exponential-Integratoren existieren verschiedene Ansätze [104, 105]. Dieses Vorgehen ist effizienter als die allgemeine Anwendung von Quadraturregeln, da hierbei das Matrixexponential in (6.3) berücksichtigt wird. Die auftretenden Matrixfunktionen werden hierbei in der Regel mit Krylov-Unterraum basierten Ansätzen berechnet [75, 76, 104]. Allerdings werden die folgenden Untersuchungen zeigen, dass sich viele der Ergebnisse auch auf den untersuchten Ansatz auf Basis von Faberpolynomen übertragen lassen.

Eine weitere Möglichkeit ist die in [33] vorgestellte Methode. Bei dieser werden die Ströme mithilfe von geeigneten ADEs entwickelt. Diese werden dann wie bei den Drude- und Lorentz-Modellen in den vorangegangenen Abschnitten in die Systemmatrix $\mathcal{H}$ aufgenommen. Hierdurch wird das Integral in (6.3) komplett vermieden. Die Quellterme werden bei der Auswertung des Matrixexponentials mitberechnet. Diese Methode ist besonders für harmonische Zeitabhängigkeiten geeignet, da in diesem Fall nur zwei zusätzliche Gleichungen nötig sind, um diese exakt zu approximieren [33]. Da in diesem Fall keine Approximation nötig ist, liegt hier kein Verlust an Genauigkeit durch die Approximation der Quellterme vor [33]. Für komplexere Zeitabhängigkeiten müssen allerdings immer mehr ADEs verwendet werden.

Eine vierte Möglichkeit kann auf Basis der Ergebnisse in [103] entwickelt werden. Diese erlaubt, wie der ADE-Ansatz, die vollständige Vermeidung von zusätzlichen Matrixfunktionen. Außerdem erlaubt er im Gegensatz zu dem ADE-Ansatz auch die Berücksichtigung nichtlinearer Effekte. Daher soll auf diesen Ansatz in Abschnitt 6.3 und Kapitel 7 gesondert eingegangen werden.

Mit diesen Vorüberlegungen soll nun ein erster Ansatz zur Einbindung von Quelltermen vorgestellt werden. Hierbei soll der zweite Ansatz auf Basis der Exponential-Integratoren verwendet werden, da dieser eine sehr allgemeine von Einbindung von Quelltermen erlaubt. Die Methoden sollen hierbei mit dem in dieser Arbeit untersuchten Ansatz auf Basis von Faberpolynomen untersucht werden. Außerdem werden verschiedene Methoden zur effizienten Berechnung diskutiert.

Entwicklung mit Faberpolynomen

Eine erste Näherung für das Integral in (6.3) kann unter der Annahme bestimmt werden, dass die Funktion $\vec{\xi}(t)$ über das Intervall $[t_n, t_n + \Delta t]$ von einem Zeitschritt Δt konstant ist. Mit dieser Annahme kann (6.3) wie folgt umgeschrieben werden:

$$\int_0^{\Delta t} \exp((\Delta t - \tau)\mathcal{H})\vec{\xi}(t_n + \Delta t/2)d\tau = \Delta t\varphi(\Delta t\mathcal{H})\vec{\xi}(t_n + \Delta t/2). \tag{6.4}$$

Die Funktion $\varphi(\Delta t\mathcal{H})$ ist mit

$$\varphi(\Delta t\mathcal{H}) = \frac{I - \exp(\Delta t\mathcal{H})}{\Delta t\mathcal{H}} \tag{6.5}$$

gegeben [71]. Damit lässt sich das komplette Zeitpropagationsschema (6.3) wie folgt umschreiben:

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t\mathcal{H})\vec{\Psi}(t_n) + \Delta t\varphi(\Delta t\mathcal{H})\vec{\xi}(t_n + \Delta t/2). \tag{6.6}$$

Dieses Verfahren wird Exponential-Mittelpunkt-Verfahren genannt [104]. Bei Abtastung der $\vec{\zeta}$ in $t = t_n$ ergibt sich das sogenannte Exponential-Euler-Verfahren. Da es sich bei (6.5) auch um eine Matrixfunktion handelt und sich im allgemeinen Fall $\zeta(t_n + \Delta t/2)$ mit jedem Zeitschritt ändern kann, muss diese, wie das Matrixexponential, in jedem Zeitschritt evaluiert werden. Diese Berechnung wird hier mit Faberpolynomen realisiert. Im Gegensatz zu dem Matrixexponential sind für die Funktion $\varphi(\Delta t\mathcal{H})$ keine analytischen Zusammenhänge für die Entwicklungskoeffizienten der Faberpolynom-Entwicklung bekannt. Stattdessen werden die Koeffizienten (5.8) hier numerisch bestimmt. Insgesamt müssen mit der Formulierung (6.6) also zwei Matrixfunktionen in jedem Zeitschritt bestimmt werden.

Erfüllt die Funktion $\vec{\xi}(t)$ die Annahme eines konstanten Funktionswerts für ein Zeitintervall oder variiert innerhalb eines Zeitschrittes Δ nur wenig, so kann (6.6) eine hinreichend genaue Approximation darstellen. Für große Zeitschrittweiten und falls $\vec{\xi}(t)$ zeitlich stark oszilliert, führt (6.4) zu zusätzlichen Fehlern. Obwohl die einzelnen Matrixfunktionen mithilfe der Faberpolynom-Entwicklung, wie im letzten Abschnitt gezeigt, sehr präzise berechnet werden, führen die zusätzlichen Fehler bei der Beschreibung der Quellfunktionen zu einem größeren Gesamtfehler.

Daher sind, insbesondere wenn große Zeitschritte für die Simulation verwendet werden sollen, genauere Approximationen für das Integral in (6.3) von Interesse. Im Folgenden soll daher mit den Methoden der Exponential-Integratoren ein Ansatz entwickelt werden, der höhere Approximationsordnungen als die Methode (6.6) erlaubt [103, 104]. Wie eingangs beschrieben, soll hierbei zunächst die Quellfunktion $\vec{\xi}(t)$ geeignet approximiert und im Anschluss das Integral analytisch bestimmt werden. Zu diesem Zweck werden hier exemplarisch, wie in [103], Taylorpolynome verwendet. Hierzu wird $\vec{\xi}(t)$ in $t = t_n$ mithilfe von Taylorpolynomen entwickelt. Diese Entwicklung wird für $\vec{\xi}(t)$ in (6.3) eingesetzt. Damit lässt sich (6.3) wie folgt umschreiben [103]:

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t\mathcal{H})\vec{\Psi}(t_n) + \sum_{k=1}^{\infty} \varphi_k(\Delta t\mathcal{H})(\Delta t)^k \vec{u}_k. \tag{6.7}$$

Bei $\vec{u}_k$ handelt es sich um die Entwicklungskoeffizienten der Taylorpolynom-Entwicklung von der Quellfunktion $\vec{\xi}(t)$. Hierbei ist zu beachten, dass es sich bei $\vec{\xi}(t)$ um die örtlich diskretisierten Quellfunktionen handelt. Wenn $\vec{\xi}(t)$ nicht örtlich konstant ist, so enthalten auch die Entwicklungskoeffizienten $\vec{u}_k$ für die verschiedenen örtlichen Diskretisierungspunkte verschiedene Werte. Hierbei gilt $\vec{u}_k = \vec{j}^{(k-1)}(t_n)$ für die Koeffizienten der Taylorpolynom-Entwicklung. Die Funktion φ_k ist durch eine Rekursionsbeziehung definiert [103, 104]:

$$\varphi_k(\Delta t\mathcal{H}) = \frac{\varphi_{k-1}(\Delta t\mathcal{H}) - I/k!}{\Delta t\mathcal{H}}. \tag{6.8}$$

Die Startbedingung der Rekursionsbeziehung (6.8) ist mit $\varphi_0(\Delta t\mathcal{H}) = \exp(\Delta t\mathcal{H})$ gegeben. Um einen Propagationsalgorithmus mit (6.7) zu konstruieren, wird die Taylorpolynom-Entwicklung bei der Ordnung N_p abgebrochen. Hierbei kommt für jeden Term der Taylorpolynom-Entwicklung eine zusätzliche Matrixfunktion φ_k hinzu, welche während der Propagation ausgewertet werden muss. Gleichzeitig steigt die Genauigkeit der Approximation von dem Integral in (6.3). Der obige Ansatz lässt sich unter den gegebenen Voraussetzungen noch weiter verbessern. Der Verlauf von $\vec{\xi}(t)$ muss über den Zeitschritt bekannt sein und nicht nur an den Abtastpunkten vorliegen. In diesem Fall kann die Approximation mit der Taylorpolynom-Entwicklung verbessert werden, indem $\vec{\xi}(t)$ in $t = t_n + \Delta t/2$ entwickelt wird. In diesem Fall liegt allerdings keine geschlossene Formulierung, wie in (6.8), für die Bestimmung der Koeffizienten der φ-Funktionen vor. Diese

müssen über einen Koeffizientenvergleich bestimmt werden. Anstelle der hier verwendeten Taylorpolynome können auch andere Approximationen für $\vec{\xi}(t)$ verwendet werden. Während an dieser Stelle exemplarisch eine Taylorpolynom-Entwicklung verwendet wird, können auch andere Approximationen, wie zum Beispiel Lagrange-Polynome, eingesetzt werden. Auch diese alternativen Approximationen für $\vec{\xi}(t)$ führen auf Zusammenhänge mit Linearkombinationen aus den φ-Funktionen wie in (6.7) [104]. Das Vorgehen ist hierbei analog.

Die auftretenden Matrixfunktionen werden hier wieder mithilfe von Faberpolynomen entwickelt. Mit den vorgestellten Ansätzen lässt sich die Quellfunktion mit hoher Genauigkeit entwickeln. Allerdings liegt abhängig von der Approximationsordnung ein erheblicher Mehraufwand gegenüber dem quellfreien Algorithmus vor. Bei den bisherigen Überlegungen wird ein allgemeines $\vec{\xi}(t)$ angesetzt. Praktisch weisen diese Quellfunktionen, zum Beispiel bei der Einkopplung von Wellen in das Simulationsgebiet, insbesondere örtlich Eigenschaften auf, welche im folgenden Abschnitt für die Optimierung des Verfahrens verwendet werden können.

6.1.2 Effiziente Implementierung der Quellterme

An dieser Stelle sollen einige Ansätze zur effizienten Implementierung diskutiert werden, welche in [KS8] vorgestellt sind. Die erste ergibt sich aus der örtlichen Verteilung von praktisch auftretenden Quelltermen. Diese sind in vielen Fällen nur für einen lokal begrenzten Bereich definiert und werden daher von wenigen örtlichen Diskretisierungspunkten beschrieben. Praktische Beispiele sind die in 2.5 beschriebenen Methoden zur Einkopplung von Wellen in das Simulationsgebiet. Andere Beispiele sind Punktquellen oder Ähnliches. In diesen Fällen kann die Systemmatrix $\mathcal{H}$ in (6.3) durch eine reduzierte Systemmatrix $\mathcal{H}'$ ersetzt werden. Diese enthält nur wenige örtliche Punkte um die Quelle. Praktisch hängt der Bereich von dem verwendeten Zeitschritt und der maximalen Ausbreitungsgeschwindigkeit der Wellen ab. Der Rest der Entwicklung wird, wie oben beschrieben, durchgeführt. Die Ersparnis an Rechenzeit ergibt sich dadurch, dass die Bestimmung der Matrix-Vektor-Multiplikationen mit der reduzierten Matrix deutlich weniger aufwendig ist, da im Gegensatz zu $\mathcal{H}$ nur ein Bruchteil des Gitters involviert ist. Dieser Ansatz lässt sich für viele gängige Diskretisierungsverfahren, wie FD- oder FIT-Verfahren verwenden. Diskontinuierliche Galerkin-Verfahren sind prinzipiell auch möglich. Die Verwendung für Verfahren, die eine globale Basis für die Darstellung der Felder nutzen, wie die klassische Formulierung von pseudospektralen Methoden in 3.1.2, ist allerdings nicht ohne weiteres möglich. Eine weitere Optimierung wird durch die Faberpolynome ermöglicht. Hierbei ist wieder die Voraussetzung, dass die Quellfunktion die oben beschriebene örtlich begrenzte Natur aufweist. Die Faberpolynome ermöglichen eine Optimierung des Konvergenzbereiches an das Eigenwertspektrum der Systemmatrix. Dies wird genutzt, dass $\mathcal{H}'$ ein anderes Eigenwertspektrum als die gesamte Matrix $\mathcal{H}$ aufweisen kann. Bei der Entwicklung der Matrixfunktionen für die Approximation der Quellterme lässt sich dies nutzten [KS8].

Hervorzuheben ist, dass beide Optimierungen ohne die Verwendung von Approximationen realisiert werden. Das bedeutet, dass sich diese in keiner Weise auf die Genauigkeit der Approximation auswirken. Außerdem sind sie auch für andere Entwicklungen als für die hier verwendete Taylorpolynom-Entwicklung verwendbar. Im folgenden Abschnitt sollen die Ansätze zur Einbindung der Quellterme für die Faberpolynom-Methode in Hinblick auf die Genauigkeit evaluiert werden.

6.1.3 Numerische Evaluation

Mit den bisherigen Überlegungen lassen sich beliebige Quellterme in den Algorithmus integrieren. Um die Genauigkeit zu vergrößern, kann die Polynomordnung erhöht werden. Für eine festgelegte Entwicklungsordnung wird die Genauigkeit der Approximation insbesondere von dem Zeitschritt Δt der Simulation und von dem zeitlichen Verhalten der Funktion $\vec{\zeta}(\vec{r}, t)$ beziehungsweise $\vec{\xi}(t)$ abhängen. Hierbei ist zu erwarten, dass ein stark oszillierendes Verhalten zu einem größeren Fehler führt. Um diese Zusammenhänge zu untersuchen, soll im Folgenden ein Testsystem numerisch evaluiert werden. Hierbei handelt es sich um einen Wellenleiter in einem photonischen Kristall. Die TM-Mode der zweidimensionalen Struktur wird analysiert. Das System ist in Abbildung 6.1 dargestellt. Das Rechengebiet ist mit einem FD-Yee-Gitter diskretisiert und es

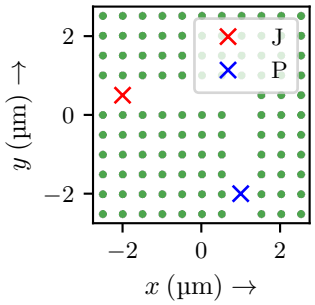
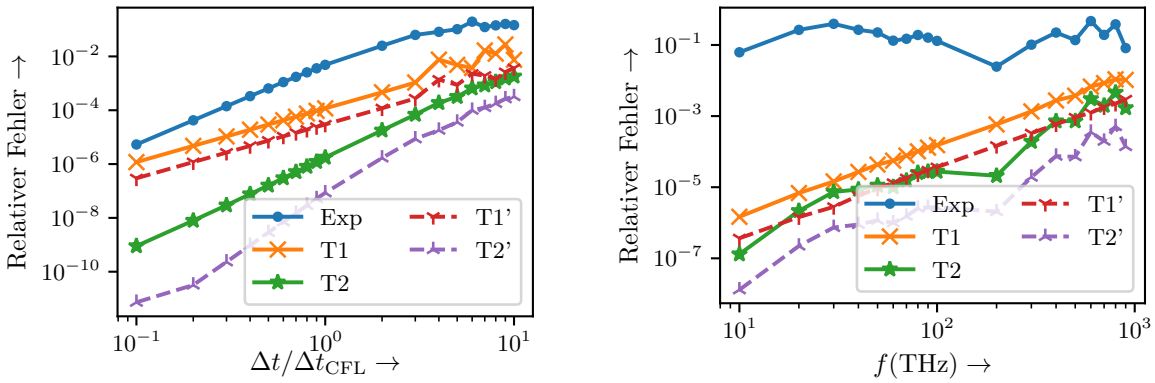


Abbildung 6.1: Das verwendete Testsystem wird in Abbildung 6.1 dargestellt. Die grünen Flächen weisen eine Permittivität von $\epsilon = 12\epsilon_0$, die weißen Bereiche eine Permittivität von $\epsilon = \epsilon_0$ auf. Der Radius der Kreise ist $r = 0{,}1\,\mu\text{m}$. Die Periodizität des photonischen Kristalls ist mit $a = 0{,}5\,\mu\text{m}$ gegeben. Das rote Kreuz markiert die Position der Punktquelle $\vec{J}$, während die Position des Messpunktes durch das blaue Kreuz angezeigt wird.

wird eine örtliche Schrittweite $\Delta = \Delta x = \Delta y = 10\,\text{nm}$ verwendet. An einer Punktquelle mit $\zeta(\vec{r}, t) = [J_0(\vec{r}) \sin(\omega_0 t)\ 0]^T$ wird ein Strom mit einer harmonischen Zeitabhängigkeit in die Simulation eingeprägt. Die Position des Stroms ist in Abbildung 6.1 gegeben. Die Frequenz ist zunächst mit $f_0 = 180\,\text{THz}$ gegeben. Nun wird das Testsystem mit den oben beschriebenen Methoden zur Einbindung der Quellterme berechnet. Hierbei wird die Zeitschrittweite zwischen $\Delta t = 0.1\Delta t_{CFL}$ und $\Delta t = 10\Delta t_{CFL}$ variiert. In diesem Zusammenhang ist hervorzuheben, dass die Faberpolynom-Methode wie bei dem quellfreien System Zeitschritte erlaubt, die das CFL-Limit Δt_{CFL} übersteigen. Die auftretenden Matrixfunktionen für die Approximation der Quellterme werden mithilfe von Faberpolynomen approximiert. Die Entwicklungskoeffizienten werden hierzu numerisch bestimmt. In Abbildung 6.2a sind die Ergebnisse dargestellt. Die betrachtete photonische Kristallstruktur hat eine Bandlücke bei der Frequenz f_0 der Quelle $\zeta(\vec{r}, t)$. Innerhalb der Kristallstruktur können sich die angeregten Wellen daher nicht ausbreiten, sodass sie innerhalb des Liniendefektes geführt werden und sich entlang von diesem ausbreiten. Die Felder, welche sich am Ende des so definierten Wellenleiters ergeben, sollen zur Fehlerberechnung herangezogen werden. Bei der Bewertung der Approximation der Quellfunktionen ist insbesondere die genaue Approximation des zeitlichen Verlaufes von Interesse. Daher wird der zeitliche Verlauf $E_z(\vec{r} = \vec{r}_P, t)$ der E_z-Komponente an dem in Abbildung 6.1 dargestellten Punkt am Ende



(a) Die Abbildung 6.2a zeigt den Fehler für verschiedene Zeitschritte Δt.

(b) In der Abbildung 6.2b ist der Fehler für verschiedene Frequenzen dargestellt.

Abbildung 6.2: Die Abbildungen 6.2a und 6.2b zeigen den Verlauf des relativen Fehlers für die verschiedenen Simulationen. Die blaue Kurve gibt die Ergebnisse für den Exponential-Euler-Algorithmus in (6.6) an. T1 beziehungsweise T2, dargestellt in Orange und Grün, geben den Verlauf des Fehlers für die Taylorpolynom basierte Methode in (6.7) für die erste und zweite Polynomordnung an. T1' und T2' geben den Fehler für die verbesserte Taylorpolynom-Methode an.

des Wellenleiters aufgenommen. Die Referenz für die Fehlerberechnung wird mithilfe eines ADE-Ansatzes bestimmt. Dieser erlaubt für zeitharmonische Quellterme deren Berücksichtigung ohne zusätzliche Fehler durch Approximationen [33]. Es wird der relative Fehler des berechneten Feldwertes für alle N Zeitschritte bestimmt und der mittlere relative Fehler betrachtet: $\epsilon_{rel} = 1/N \sum_{n=1}^{N} |E_{z,ref}(\vec{r} = \vec{r}_P, t = t_n) - E_z(\vec{r} = \vec{r}_P, t = t_n)|/|E_{z,ref}(\vec{r} = \vec{r}_P, t = t_n)|$. Beim Vergleich der Fehlerkurven fällt auf, dass alle Algorithmen erwartungsgemäß bei großen Zeitschritten einen größeren Fehler aufweisen. Der Ansatz (6.6) zeigt den größten Fehler. Der Taylorpolynom-Ansatz (6.7) mit Polynomen erster und zweiter Ordnung erlaubt eine deutliche Steigerung der Genauigkeit. Noch weiter kann diese mit dem verbesserten Taylorpolynom-Ansatz erhöht werden.

Nun wird der Zeitschritt auf $\Delta t = 2\Delta t_{CFL}$ festgelegt, während die Frequenz f_0 des Stroms zwischen $f_0 = 10\,\mathrm{THz}$ und $f_0 = 1000\,\mathrm{THz}$ variiert wird. Das Ergebnis ist in Abbildung 6.2b grafisch dargestellt. Bei der Untersuchung zeigt sich ein dem vorangegangenen Versuch ähnlicher Verlauf. Für hohe Frequenzen steigt der Fehler. Wieder zeigen die Algorithmen mit der erhöhten Entwicklungsordnung den geringsten Fehler.

6.1.4 Diskussion

In dem vorangegangenen Abschnitt wird der Faberpolynom-Algorithmus erweitert, sodass auch Quellterme berücksichtigt werden können. Ein erster Ansatz wird mithilfe einer Taylorpolynom-Entwicklung der Quellterme erreicht. Mit dessen Ordnung kann die Genauigkeit der Quellapproximation gesteuert werden. Hierbei hängen die Anforderungen an die Approximation primär von dem verwendeten Zeitschritt und dem Zeitverhalten der Quellfunktion ab. Die Untersuchungen

zeigen, dass auch bei hohen Zeitschritten beziehungsweise stark oszillierenden Quellfunktionen eine präzise Approximation der Quelle durch höhere Polynomordnungen möglich ist. Die hohe Approximationsgüte des linearen quellfreien Teils von (6.3) kann also bei der Approximation der Quellterme erhalten werden. Allerdings erhöht sich mit der Approximation auch die Anzahl der zusätzlichen Matrixfunktionen. Mit den vorgestellten Ansätzen lässt sich unter Verwendung der reduzierten Matrix der Berechnungsaufwand zwar erheblich reduzieren, jedoch sind die möglichen Einsparungen von dem untersuchten System und der örtlichen Struktur der Quellterme abhängig. Ein Ansatz zur weiteren Verbesserung der Quellapproximation ist die Verwendung einer anderen Approximation für die Quellfunktion anstelle der Taylorpolynom-Entwicklung. Hier bieten sich beispielsweise Tschebyscheff-Polynome oder Lagrange-Polynome an. Dennoch führen hohe Genauigkeitsanforderungen im Allgemeinen zu einer großen Zahl von zusätzlichen Matrixfunktionen. In den folgenden Kapiteln werden daher noch weitere Methoden untersucht, mit denen die Effizienz weiter zu steigern ist. Hierzu werden die Eigenschaften der Quellfunktionen im Hinblick auf ihre Orts- und Zeitabhängigkeit genutzt, um die Approximation effizienter zu gestalten.

6.2 Komplexe-Einhüllenden-Methode

Im vorangegangenen Abschnitt wird ein Algorithmus zur Lösung der Maxwell-Gleichungen mit Quelltermen auf Basis von Faberpolynomen vorgestellt. In diesem Zusammenhang konnte gezeigt werden, dass diese einen effizienten Zeitpropagationsalgorithmus ermöglichen.Bei der Entwicklung der Quellterme werden allerdings keine weiteren Annahmen bezüglich der Zeitabhängigkeit getroffen. In vielen Anwendungen aus der Photonik oder der Terahertz-Technik liegen hochfrequente Trägerschwingungen vor. Dabei handelt es sich zum Beispiel in der Photonik in der Regel um das Ausgangssignal eines Lasers. Dieser verfügt zwar über eine gewisse Linienbreite beziehungsweise wird mit einem Signal moduliert, allerdings ist die Bandbreite dieser Signale in der Regel deutlich kleiner im Vergleich zu der Frequenz der Trägerschwingung.

Für solche Problemstellungen ist es vorteilhaft, die Frequenz der Trägerschwingung bei der Konstruktion des Zeitpropagationsalgorithmus zu berücksichtigen. Ein intuitiver Ansatz hierfür ist, die Feldgrößen mithilfe einer komplexen Einhüllenden zu entwickeln. Methoden, welche diesen Ansatz verfolgen, werden in der Literatur im Zusammenhang mit dem FDTD-Algorithmus untersucht. Hierbei werden sowohl explizite [106–109], als auch implizite [110–112] Varianten beleuchtet.

Die Besonderheit bei diesen Ansätzen ist, dass es sich immer noch um Zeitbereichsalgorithmen handelt und nicht um Frequenzbereichssimulationen. Im Folgenden soll ein Algorithmus untersucht werden, bei dem mithilfe eines Komplexe-Einhüllenden-Ansatzes ein Zeitbereichsalgorithmus auf Basis der Faberpolynome realisiert wird. In diesem Zusammenhang soll insbesondere die Möglichkeit der Anpassung an das Eigenwertspektrum der Systemmatrix genutzt werden, welches die Faberpolynom-Entwicklung bietet, um eine effiziente Approximation zu gewährleisten.

Im Anschluss wird dargelegt, wie diese Formulierung zur effizienteren Einbindung von Quelltermen verwendet werden kann. Die Untersuchungen im vorangegangenen Abschnitt zeigen, dass deren Einbindung bei hohen Genauigkeitsanforderungen zu einer großen Anzahl von zusätzlichen Matrixfunktionen führen kann, welche während der Propagation ausgewertet werden müssen. In diesem Zusammenhang soll untersucht werden, für welche Fälle die Verwendung

der Komplexe-Einhüllenden-Methode eine effizientere Realisierung ermöglicht. Die Ergebnisse dieses Abschnitts liegen in Teilen in [KS9, KS10] veröffentlicht vor und werden im Folgendem mit Ergänzungen zur Darstellung gebracht.

6.2.1 Theorie – Maxwell-Gleichungen mit einer komplexen Einhüllenden

Zunächst soll die Komplexe-Einhüllenden-Methode für die Verwendung mit den quellfreien Maxwell-Gleichungen beschrieben werden. Hierzu wird in Gleichung (2.17) auf alle Feldgrößen ein komplexer Einhüllenden-Ansatz angewendet [107, KS9]:

$$\vec{\psi}(\vec{r},t) = \mathcal{R}\{\hat{\vec{\psi}}(\vec{r},t)e^{j\omega_0 t}\}. \tag{6.9}$$

Bei ω_0 handelt es sich um die Kreisfrequenz der Trägerschwingung des betrachteten Systems. Mit diesem Ansatz lässt sich (2.17) wie folgt umschreiben:

$$\frac{\partial\hat{\vec{\psi}}(\vec{r},t)}{\partial t} = \begin{bmatrix} -j\omega_0 & \frac{1}{\epsilon(\vec{r})}\nabla\times \\ -\frac{1}{\mu(\vec{r})}\nabla\times & -j\omega_0 \end{bmatrix} \hat{\vec{\psi}}(\vec{r},t), \tag{6.10}$$

wobei durch die Anwendung des komplexen Einhüllenden-Ansatzes ein neuer Matrixoperator in (6.10) vorliegt. Bei $\hat{\vec{\psi}}(\vec{r},t)$ handelt es sich um die mit (6.9) definierten Feldgrößen, welche hier mit

$$\hat{\vec{\psi}}(\vec{r},t) = \begin{bmatrix} \hat{\vec{E}}(\vec{r},t) \\ \hat{\vec{H}}(\vec{r},t) \end{bmatrix} \tag{6.11}$$

gegeben sind. Liegen neben dem elektrischen und dem magnetischen Feld noch weitere Variablen in dem Vektor $\vec{\psi}(\vec{r},t)$ vor, ist mit diesen genau wie mit dem elektrischen und dem magnetischen Feldgrößen zu verfahren. Dies ist beispielsweise der Fall, wenn Materialgleichungen mit dem in 2.3 beschriebenen ADE-Verfahren eingebunden werden sollen. Außerdem führt auch die in dieser Arbeit verwendete Variante der PML auf solche ADE, wie in 2.4 beschrieben wird. Im Anschluss wird (6.10) örtlich diskretisiert. Hierzu bieten sich neben dem FD-Yee-Gitter auch pseudospektrale Methoden oder die DG-Methode an. Hier soll allerdings zunächst das FD-Yee-Gitter verwendet werden, dessen Implementierung in Abschnitt 3.1 beschrieben ist. Das örtlich diskretisierte System ist mit

$$\frac{\partial}{\partial t}\hat{\vec{\Psi}}(t) = \hat{\mathcal{H}}\hat{\vec{\Psi}}(t) \tag{6.12}$$

gegeben. Die Vektor $\hat{\vec{\Psi}}(t)$ ist der örtlich diskretisierte Feldvektor $\hat{\vec{\psi}}(\vec{r},t)$ und $\hat{\mathcal{H}}$ ist die örtlich diskretisierte Systemmatrix. Wie in (6.10) zu erkennen, ist die Schwingung der Trägerfrequenz mit ω_0 nun fest in die Systemmatrix $\hat{\mathcal{H}}$ eingebettet. Das Schwingungsverhalten der Trägerschwingung ω_0 wird also mit in das Modell aufgenommen. Hierbei ist anzumerken, dass bei der Komplexe-Einhüllenden-Methode keine Approximation angewendet wird. Bisher sind keinerlei Näherungen bezüglich der Frequenzabhängigkeit der untersuchten Felder vorgenommen worden. Die formale Lösung des örtlich diskretisierten Systems (6.12) ist mit

$$\hat{\vec{\Psi}}(t_n + \Delta t) = \exp(\Delta t\hat{\mathcal{H}})\hat{\vec{\Psi}}(t_n) \tag{6.13}$$

gegeben. Um mit diesem Ansatz nun einen Zeitpropagationalgorithmus zu konstruieren, soll das Matrixexponential in (6.13) mithilfe von Faberpolynomen entwickelt werden. Dies und die nötigen Vorüberlegungen bezüglich des Eigenwertspektrums der neuen Systemmatrix $\hat{\mathcal{H}}$ werden im nächsten Abschnitt skizziert.

6.2.2 Entwicklung mit Faberpolynomen und Abschätzung des Eigenwertspektrums

Im Folgenden soll die Faberpolynom-Entwicklung verwendet werden, um das Matrixexponential in (6.13) zu entwickeln. Wie im vorangegangenen Abschnitt beschrieben, ist die Approximation in der komplexen Ebene definiert. Damit ein stabiler Algorithmus konstruiert werden kann, müssen alle Eigenwerte der Systemmatrix $\hat{\mathcal{H}}$ innerhalb eines zuvor definierten Konvergenzbereiches liegen. Hierzu müssen zuerst Informationen über das Eigenwertspektrum von $\hat{\mathcal{H}}$ bekannt sein. Im zweiten Schritt muss die Approximation mit den Faberpolynomen angepasst werden, um diese zu berücksichtigen.

Hierzu ist zu untersuchen, welchen Einfluss die Inklusion der Trägerfrequenz ω_0 in die Systemmatrix hat. Bei Betrachtung der Systemmatrix der komplexen Einhüllenden $\hat{\mathcal{H}}$ in (6.12) im Vergleich zu der ursprünglichen $\mathcal{H}$ in (2.17) fällt auf, dass diese sich nur durch die Einträge $-j\omega_0$ auf der Hauptdiagonalen der Matrix unterscheiden. Da der Komplexe-Einhüllenden-Ansatz (6.9) auf alle Feldkomponenten in (2.17) angewendet wird, liegt dieser Eintrag für alle Hauptdiagonalelemente der Matrix vor. Ausgehend von dieser Eigenschaft, lässt sich beispielsweise mithilfe dem Gerschgorin-Theorem [54, 113] zeigen, dass die Inklusion der Trägerschwingung in die Systemmatrix die Form ihres Eigenwertspektrums nicht beeinflusst. Stattdessen hat die Inklusion eine Verschiebung des gesamten Eigenwertspektrums von (2.17) entlang der imaginären Achse zur Folge. Dieser Prozess ist in Abbildung 6.3 schematisch dargestellt. Dies hat zur Folge,

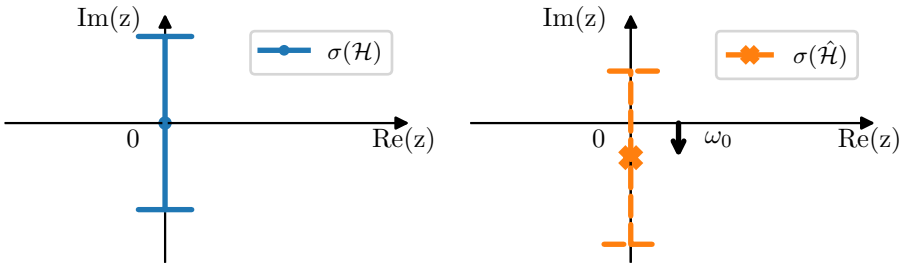


Abbildung 6.3: In Abbildung 6.3 ist auf der linken Seite die Verteilung der Eigenwerte der Systemmatrix $\mathcal{H}$ in der komplexen Ebene schematisch dargestellt. Auf der rechten Seite ist eine Darstellung der Eigenwerte der Systemmatrix $\hat{\mathcal{H}}$ zu finden, welche durch Anwendung der komplexen Einhüllenden aus $\mathcal{H}$ abgeleitet werden kann.

dass zunächst die Systemmatrix $\mathcal{H}$ ohne Einhüllende mit den Methoden des vorangegangenen Abschnitts hinsichtlich der Grenzen ihres Eigenwertspektrums untersucht werden kann. Im Anschluss kann die Verschiebung durch die Trägerschwingung berücksichtigt werden, um die Grenzen von $\hat{\mathcal{H}}$ zu bestimmen.

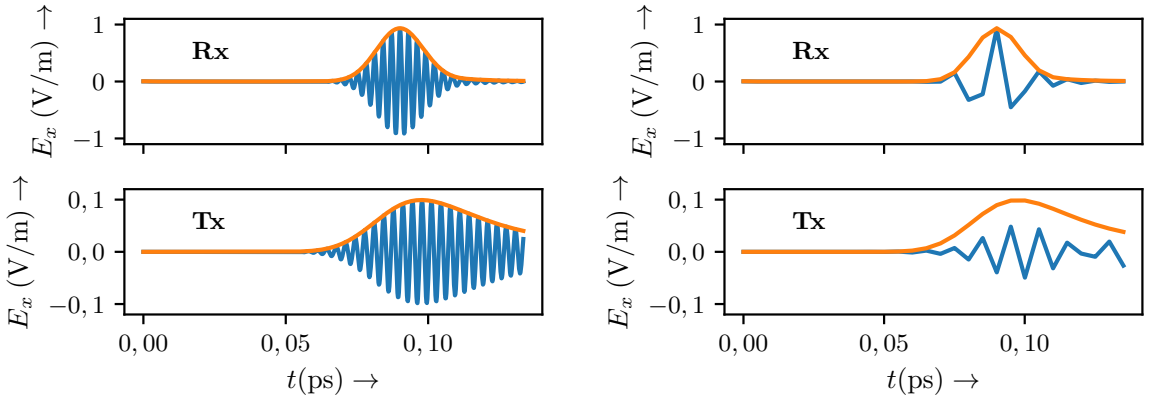
Im zweiten Schritt soll nun die Faberpolynom-Approximation auf die neue Systemmatrix $\hat{\mathcal{H}}$ angepasst werden. Die zu approximierende Matrixfunktion ist wie im vorangegangenen Abschnitt ein Matrixexponential. Hierbei soll wieder ein elliptischer Konvergenzbereich angesetzt werden. Daher sind wieder die Koeffizienten der konformen Abbildung in (5.10) zu finden. Außerdem ist der Parameter λ_s, mit dem die Matrix skaliert wird, zu bestimmen. Während die Parameter γ_1 und λ_s maßgeblich von der Form und der Ausdehnung der Ellipse beeinflusst werden und so analog zu Kapitel 5 berechnet werden können, wird der Parameter γ_0 von dem Mittelpunkt der Ellipse bestimmt.

Daher kann die Verschiebung entlang der imaginären Achse allein durch die Anpassung von γ_0 berücksichtigt werden, was mit $\gamma_0 = x_0 + jy_0$ gegeben ist. Genauer gilt

$$y_0 = -\omega_0. \tag{6.14}$$

Dies hat außerdem zur Folge, dass die Verwendung der komplexen Einhüllenden nicht zur einer Erhöhung des Approximationsaufwandes im Hinblick auf die nötige Polynomordnung führt. Allerdings sind die Koeffizienten c_m der Faberpolynome durch die Verwendung des Komplexe-Einhüllenden-Methode komplexwertig. Mit diesen Vorüberlegungen ist es nun möglich, das Matrixexponential in (6.13) mit Faberpolynomen zu approximieren und einen Zeitpropagationalgorithmus zu realisieren. Mithilfe von (6.9) können die berechneten Feldverteilungen jederzeit zurücktransformiert werden.

Die bisherigen Überlegungen sollen mit einem Beispiel illustriert werden. In dem Beispiel wird die Interaktion eines modulierten Impulses mit einem Bragg-Gitter untersucht. Hierzu wird ein eindimensionales System mit den Abmessungen $z \in [0, L_z]$ mit $L_z = 75\,\mu\text{m}$ und einer FD-Yee-Diskretisierung mit einer Schrittweite $\Delta z = 5\,\text{nm}$ betrachtet. Das Bragg-Gitter hat eine Gitter-Periode von $\Lambda = 400\,\text{nm}$ und beginnt bei $z = 35{,}5\,\mu\text{m}$ und hat eine Länge von $L_{\text{Bragg}} = 4\,\mu\text{m}$. Innerhalb einer Periode variiert die Brechzahl zwischen $n_1 = 3$ und $n_0 = 1$, während für den umgebenden Bereich $n_0 = 1$ gilt. Der gaußförmige Impuls hat eine FWHM-Bandbreite von $B = 50\,\text{THz}$ und hat eine Trägerschwingung mit einer Vakuumwellenlänge von $\lambda_0 = 900\,\text{nm}$. Der Impuls wird bei $z_0 = 11{,}25\,\mu\text{m}$ platziert und initialisiert, sodass er in Richtung des Gitters propagiert. Die Simulationszeit beträgt $T = 0{,}133\,\text{ps}$. Die transmittierten Signale werden an $z_{Tx} = 44{,}25\,\mu\text{m}$ gemessen sowie die reflektierten an $z_{Rx} = 18{,}75\,\mu\text{m}$. Die Simulation wird mit der Faberpolynom-Methode aus Kapitel 5 sowie mit der Komplexe-Einhüllenden-Methode mit Faberpolynomen bestimmt. Die Ergebnisse sind in den Abbildungen in 6.4 dargestellt. Zuerst wird in Abbildung 6.4a ein Zeitschritt mit $\Delta t = 10\Delta t_{\text{CFL}}$ verwendet. Da die Wellenlänge der Trägerschwingung des Impulses von der Mittenwellenlänge des Gitters abweicht, ist eine teilweise Reflexion beziehungsweise Transmission zu erwarten. Der Komplexe-Einhüllenden-Ansatz wird hier mit der Trägerfrequenz des Impulses berechnet. Die so berechnete Einhüllende stimmt mit den Ergebnissen der konventionellen Faber Methode überein. Die Trägerschwingung ist hier mit 18 Abtastpunkten pro Periode zeitlich aufgelöst. Hierbei kann mit (6.9) jederzeit das ursprüngliche Feld aus dem Ergebnis der Komplexe-Einhüllenden-Methode wiederhergestellt werden. In Abbildung 6.4b wird das Ergebnis derselben Simulation mit einem deutlich höheren Zeitschritt $\Delta t = 300\Delta t_{\text{CFL}}$ betrachtet. In diesem Fall zeigen sich deutliche Unterschiede. Die Zeitschrittweite ist so hoch, dass bei der Faberpolynom-Methode die Trägerschwingung mit $0,6$ Abtastpunkten pro Periode nicht mehr ausreichend aufgelöst werden kann. Bei der Komplexe-Einhüllenden-Methode hingegen ist die Form der Impulse noch zu erkennen, da hier nur die Einhüllenden des gaußförmigen Impulses abgetastet werden muss.



(a) Zeitschritt: $\Delta t = 10\Delta t_{\mathrm{CFL}}$.

(b) Zeitschritt: $\Delta t = 300\Delta t_{\mathrm{CFL}}$.

Abbildung 6.4: Die Abbildungen zeigen jeweils oben die reflektierten Signale und unten die transmittierten. Die Ergebnisse der konventionellen Faberpolynom-Methode sind mit den blauen Linien dargestellt, während für die Komplexe-Einhüllenden-Methode die orangen Linien verwendet werden. Bei der Komplexe-Einhüllenden-Methode wird der Betrag der Einhüllenden dargestellt.

Hierbei ist anzumerken, dass beide Algorithmen mit der hohen Genauigkeit, welche in Abschnitt 5.4 gezeigt wird, arbeiten. Die hohen Zeitschrittweiten sorgen für eine Unterabtastung der gemessenen Zeitverläufe. Die Komplexe-Einhüllenden-Methode stellt eine Möglichkeit dar, deutlich höhere Zeitschrittweiten zu verwenden, da hier nur die Einhüllende zeitlich abgetastet werden muss. Die hochfrequente Trägerschwingung ist in der Systemmatrix $\hat{\mathcal{H}}$ enthalten. Allerdings hat die Methode den Nachteil, dass sie die Verwendung von komplexen Zahlen erfordert, während die konventionelle Faberpolynom-Methode aus Kapitel 5 mit reellen Zahlen berechnet werden kann. In den folgenden Abschnitten wird der Komplexe-Einhüllenden-Ansatz auf Basis der bisherigen Ergebnisse noch erweitert werden.

6.2.3 Entwicklung der Quellterme

Nun soll auf die Einbindung von Quellfunktionen wie in (6.1) eingegangen werden. Das örtlich diskretisierte System für die Maxwell-Gleichungen mit der Komplexe-Einhüllenden-Methode und einer zunächst allgemein angesetzten Quellfunktion $\vec{\zeta}(\vec{r}, t)$ lässt sich wie folgt angeben:

$$\frac{\partial \hat{\vec{\Psi}}(t)}{\partial t} = \hat{\mathcal{H}}\hat{\vec{\Psi}}(t) + \hat{\vec{\xi}}(t). \tag{6.15}$$

Hierbei ist $\hat{\vec{\xi}}(t)$ die diskretisierte Quellfunktion nach Anwendung des Komplexe-Einhüllenden-Ansatzes und ist, abgesehen von der Einhüllenden, analog zu $\vec{\xi}(t)$ in (6.2) definiert. Die formale Lösung der Gleichung kann analog zu (6.3) mit

$$\hat{\vec{\Psi}}(t_n + \Delta t) = \exp(\Delta t\hat{\mathcal{H}})\hat{\vec{\Psi}}(t_n) + \int_0^{\Delta t} \exp((\Delta t - \tau)\hat{\mathcal{H}})\hat{\vec{\xi}}(t_n + \tau)d\tau \tag{6.16}$$

angegeben werden. Der zusätzliche Term für die Quellfunktionen hat die gleiche Struktur wie der Term ohne die komplexe Einhüllende in (6.3). Dies erlaubt es, die gleichen Methoden für die Entwicklung des Integralterms in (6.16) zu verwenden, welche schon in Abschnitt 6.1 vorgestellt werden. Eine erste Approximation für den Quellterm in (6.16) ergibt sich daher analog zu (6.4) mit

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t \hat{\mathcal{H}})\vec{\Psi}(t_n) + \Delta t \varphi(\Delta t \hat{\mathcal{H}})\vec{\xi}(t_n + \Delta t/2). \tag{6.17}$$

Hierbei ist 6.17 das Exponential-Mittelpunkt-Verfahren für die Komplexe-Einhüllenden-Methode. Der erste Vorteil der Komplexe-Einhüllenden-Methode ergibt sich bei Betrachtung der Zeitabhängigkeit des Quellterms $\vec{\zeta}(\vec{r}, t)$. Hat die Quellfunktion eine hochfrequente Trägerschwingung, um welche die Signale bandbegrenzt sind, so müsste eine hohe Approximationsordnung für die Quellterme gewählt werden. Dies ist darin begründet, dass für die Trägerschwingung eine große Anzahl von Approximationsfunktionen nötig ist, um die Schwingung auf dem Zeitschritt $t \in [t_n, t_n + \Delta t]$ zu approximieren. Dieser Effekt wird für größere Zeitschritte Δt immer ausgeprägter. Bei der Komplexe-Einhüllenden-Methode liegt in (6.16) allerdings nur noch die komplexe Einhüllende $\vec{\tilde{\xi}}(t)$ der Quellfunktion $\vec{\zeta}(\vec{r}, t)$ vor. Dadurch muss die hochfrequente Trägerschwingung nicht mit approximiert werden. Die Beschreibung dieser Schwingung ist hierbei in der Systemmatrix $\hat{\mathcal{H}}$ enthalten. Daher muss nur noch die komplexe Einhüllende $\vec{\tilde{\xi}}(t)$ approximiert werden, welche bei bandbegrenzten Signalen deutlich schwächer oszilliert als die Trägerschwingung ω_0. Dies erlaubt es, eine deutlich geringere Approximationsordnung für die Quellterme zu verwenden, was zu einer geringeren Anzahl von zusätzlichen Matrixfunktionen führt. In vielen Fällen ist sogar die Approximation (6.17), welche eine konstante einhüllende Funktion auf $t \in [t_n, t_n + \Delta t]$ annimmt, hinreichend.

Diese Eigenschaft wird in Abschnitt 6.2.4 numerisch evaluiert. Im Folgenden werden zunächst weitere Möglichkeiten zur Effizienzsteigerung durch den Komplexen-Einhüllenden-Ansatz in den Blick genommen.

Berücksichtigung der Eigenschaften der Quellfunktion

Neben der Zeitabhängigkeit ist auch die Ortsabhängigkeit der Quellfunktion $\vec{\tilde{\zeta}}(\vec{r}, t)$ von zentraler Bedeutung. In Abschnitt 6.1.2 wird bereits beschrieben, wie die örtliche Lokalisierung der Quellterme genutzt werden kann, um die Approximation mithilfe einer reduzierten Matrix effizienter zu gestalten. Die Verwendung der komplexen Einhüllenden erlaubt hierbei noch weitere Verbesserungen. In einigen Sonderfällen lässt sich die Ortsabhängigkeit der Quellfunktion vollständig von der Zeitabhängigkeit trennen:

$$\vec{\zeta}(\vec{r}, t) = \vec{\zeta_0}(\vec{r}) f(t). \tag{6.18}$$

Beispiele für solche Fälle sind Punktquellen oder die Einkopplung von ebenen Wellen mit einem Einfallswinkel, welcher normal zu der Ebene der Einkopplung liegt. Dadurch lässt sich für das diskrete System die folgende Vereinfachung durchführen:

$$\Delta\vec{\xi_0} = \int_0^{\Delta t} \exp((\Delta t - \tau)\mathcal{H})\vec{\xi_0} d\tau = \Delta t \frac{I - \exp(\Delta t \mathcal{H})}{\Delta t \mathcal{H}} \vec{\xi_0}. \tag{6.19}$$

Die Berechnung der Interaktion mit der statischen örtlichen Verteilung $\vec{\zeta_0}(\vec{r})$ beziehungsweise ihres diskretisierten Äquivalents $\vec{\xi_0}$ muss hierbei nur einmal zu Beginn der Simulation bestimmt

werden. Dadurch kann der Aufwand zur Einbindung des Quellterms signifikant gesenkt werden. Für die Einbindung muss keine zusätzliche Matrixfunktion evaluiert, sondern lediglich der Vektor $\Delta\vec{\xi}_0$ multipliziert mit der Funktion $f(t)$ zu dem quellfreien Teil addiert werden. Das Zeitpropagationsschema ist mit

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t\mathcal{H})\vec{\Psi}(t_n) + \Delta\vec{\zeta}_0(\vec{r})f(t) \tag{6.20}$$

gegeben. An dieser Stelle sei angemerkt, dass diese Optimierung auch mit der Faberpolynom-Methode ohne komplexe Einhüllende möglich ist. Ein weiterer Sonderfall liegt vor, wenn die Quellfunktion $\vec{\zeta}(\vec{r}, t)$ sich in der Form

$$\vec{\zeta}(\vec{r}, t) = \vec{\zeta}_0(\vec{r})\sin(\omega_0 t + \beta(\vec{r}))f(t) \tag{6.21}$$

darstellen lässt. In der Praxis ist dies beispielsweise für die Einkopplung von ebenen Wellen mit beliebigen Einfallswinkeln sowie bei der Verwendung des TFSF-Ansatzes gegeben. Außerdem trifft dies bei der Einkopplung von Eigenmoden in einen Wellenleiter zu. In diesem Fall ist es nicht mehr möglich, die Vereinfachung (6.21) mit der Faberpolynom-Methode ohne die komplexe Einhüllende zu verwenden. Durch die Komplexe-Einhüllenden-Methode hingegen lässt sich (6.21) zu

$$\vec{\tilde{\zeta}}(\vec{r}, t) = \vec{\zeta}_0(\vec{r})\exp(j(\beta(\vec{r}) - \pi/2))f(t) \tag{6.22}$$

umschreiben. Hierbei wird die ortsabhängige Phasenlage $\beta(\vec{r})$ in Bezug auf die Trägerschwingung in den ortsabhängigen Vektor integriert. Dadurch kann (6.22) wiederum mit (6.20) berücksichtigt werden. Dies ermöglicht die Anwendung auf eine große Anzahl von praxisrelevanten Fällen. Die Voraussetzung dazu ist, dass es sich bei $\vec{\zeta}(\vec{r}, t)$ um eine Funktion handelt, welche bandbegrenzt um eine Trägerschwingung ist. Dies ist jedoch keine formale Voraussetzung. Eine Anwendung ist auch in anderen Fällen möglich. Allerdings ist hierbei nicht zu erwarten, dass geringere Fehler erzielt werden können. Während bei den oben gezeigten Optimierungen zunächst mit (6.19) und (6.20) eine Approximation niedriger Ordnung verwendet wird, so sind auch höhere Approximationsordnungen mit diesen Verfahren möglich. Wird beispielsweise der Taylorpolynom-Ansatz in Abschnitt 6.1 verwendet, so führt jede zusätzliche Ordnung nur zu einer weiteren Addition mit einem gewichteten Vektor. Ohne diesen Ansatz muss jeweils eine weitere Matrixfunktion ausgewertet werden.

Optimierung für die Approximation von Quelltermen

Bei Implementierung des Zeitpropagationsschemas mit der komplexen Einhüllenden in (6.16) ist die Verwendung der komplexen Zahlen in Hinblick auf den Rechenaufwand problematisch. Die Faberpolynom-Methode ohne komplexe Einhüllende auf Basis von (6.3) lässt sich ohne die Verwendung von komplexen Zahlen realisieren. Durch die Verwendung von komplexer Algebra verdoppelt sich nicht nur der Speicherbedarf, da nun immer Real- und Imaginärteil gespeichert werden müssen. Zusätzlich ist der Rechenaufwand für alle arithmetischen Operationen höher [114]. Daher ist es wünschenswert, den Einsatz von komplexer Arithmetik zu minimieren. Wie zuvor beschrieben, ist der Einsatz der komplexen Einhüllenden besonders für Einbindung von hoch oszillierenden Quelltermen interessant. Daher soll ein Ansatz gefunden werden, mit dem die Einhüllende nur für die Quellterme eingesetzt wird. Durch Verwenden von (6.9) lässt sich (6.16) wie folgt darstellen:

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t\mathcal{H})\vec{\Psi}(\vec{r}, t_n) + \mathcal{R}\left\{e^{j\omega_0(t_n+\Delta t)}\int_0^{\Delta t}\exp((\Delta t - \tau)\hat{\mathcal{H}})\vec{\hat{\xi}}(t_n + \tau)d\tau\right\}. \tag{6.23}$$

Wie (6.23) zu erkennen, kann mit dieser Anpassung der lineare quellfreie Teil wieder vollständig ohne komplexe Arithmetik realisiert werden. Sind für die Quellterme, welche in (6.23) die Verwendung von komplexen Zahlen erfordern, die Optimierungen aus dem letzten Abschnitt anwendbar, so müssen die Matrixfunktionen für die Entwicklung im Vorfeld nur einmal bestimmt werden. Während der Propagation wird nur die Addition von gewichteten Vektoren benötigt, sodass keinerlei Matrixfunktion für die Evaluation der Quellterme bestimmt werden muss. Dies reduziert den Einsatz von komplexer Arithmetik erheblich.

6.2.4 Numerische Evaluation

In den vorangegangenen Abschnitten wird ein Ansatz entwickelt, bei dem die Beschreibung einer Trägerschwingung direkt in die Systemmatrix aufgenommen wird. Im Folgendem soll dieser Ansatz numerisch evaluiert werden. Hierbei wird besonders auf die Genauigkeit bei der Einbindung von Quelltermen eingegangen. Die Komplexe-Einhüllenden-Methode wird mit der Methode aus Abschnitt 6.1 verglichen.

Bewertung zur Einkopplung von Eigenmoden

Das Potenzial der Komplexe-Einhülleden-Methode soll zunächst mithilfe eines Testsystems untersucht werden. Bei dem System handelt sich um einen dielektrischen Wellenleiter, bei dem präzise eine Eigenmode angeregt werden soll. Um die Anregung umzusetzen, wird die in Abschnitt 2.5 beschriebene Formulierung auf Basis des Äquivalenzprinzips verwendet. Die Einbindung der auftretenden Quellterme wird einmal mit der konventionellen Faberpolynom-Methode und einmal mit der Methode mit der komplexen Einhüllenden berechnet. Die Methoden sollen auf ihre Genauigkeit verglichen werden. Bei der Untersuchung kann davon ausgegangen werden, dass mögliche Fehler primär bei der Einbindung der Quellterme auftreten, da der lineare Teil, wie in Abschnitt 5.4 untersucht, mit einer sehr hohen Genauigkeit bestimmt wird.

Bei dem Wellenleiter handelt es sich um einen Filmwellenleiter. Daher wird das System auf ein zweidimensionales reduziert. Das System wird in der x-y-Ebene betrachtet und die TE-Mode wird untersucht. Der Kern des Wellenleiters hat eine Breite von $w = 1\,\mu\text{m}$. Der Brechungsindex des Kerns ist mit $n_{\text{Kern}} = 3{,}673$ gegeben. Der Mantel hat einen Brechungsindex von $n_{\text{Mantel}} = 1{,}444$. Die einkoppelte Eigenmode soll sich in positiver x-Richtung ausbreiten. In Abbildung 6.5 ist das Testsystem zu sehen. Die Größe des Rechengebietes ist mit $L_y = 6\,\mu\text{m}$ und $L_x = 10\,\mu\text{m}$ gegeben. Das System wird mithilfe eines FD-Yee-Gitters örtlich diskretisiert. Die Diskretisierungsweiten sind hierbei mit $\Delta x = \Delta y = 20\,\text{nm}$ gegeben. An den Rändern des Rechengebietes wird eine CFS-PML gemäß [41] eingesetzt, um Reflexionen zu vermeiden. Die eingekoppelte Eigenmode wird für $f_0 = 193{,}4\,\text{THz}$ bestimmt. Die Moden werden mit einem numerischen Modenlöser bestimmt, welcher das gleiche örtliche Gitter verwendet. Der Modenlöser ist vollvektoriell und basiert auf der Generalized Transmission Line Equations (GTL)-Formulierung [115]. Hierbei wird die Grundmode des Wellenleiters eingekoppelt. Mit der Methode aus [42], welche in Abschnitt 6.1 näher beschrieben ist, werden aus der Modenlösung entsprechende Ströme an der Grenzfläche bestimmt. Die Eigenmode wird mit einer Einhüllenden mit einem gaußförmigen Impuls eingekoppelt. Zuerst soll eine Approximation mit einer niedrigen Ordnung in den Blick genommen werden. Hierzu wird der Ansatz (6.17) für die Komplexe-Einhüllenden-Methode beziehungsweise

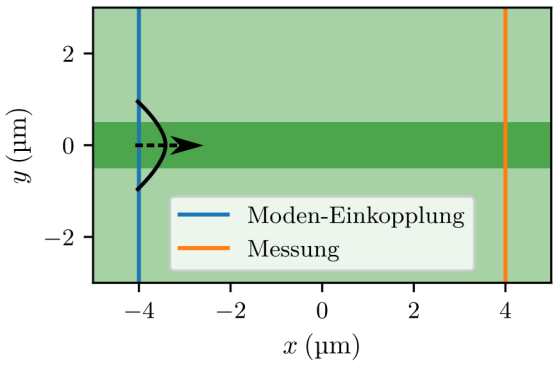


Abbildung 6.5: Die Abbildung zeigt das untersuchte Testsystem. Der dunkle grüne Bereich stellt den Kern des Wellenleiters dar, während der hellere Bereich den Mantel darstellt. Die Ebene mit $x_i = -4\,\mu\text{m}$, an der die Eigenmode eingekoppelt werden soll, ist mit der blauen Linie gegeben. Die orange-farbene Linie die Ebene mit $x_p = 4\,\mu\text{m}$ darstellt, an der die Ergebnisse gemessen werden sollen.

(6.6) für die konventionelle Faberpolynom-Methode verwendet. Für die komplexe Einhüllende wird die Trägerfrequenz f_0 verwendet. Die Simulation wird für eine Simulationszeit $T = 0{,}5\,\text{ps}$ ausgeführt. Die Ergebnisse werden an der Ebene mit $x = x_p$ aufgenommen. Zunächst wird die Simulation für einen Zeitschritt $\Delta t = \Delta t_{\text{CFL}}$ durchgeführt. Die resultierenden Zeitverläufe sind in Abbildung 6.6 dargestellt. Da die Trägerschwingung f_0 bei dem Zeitschritt mit $54,81$ Abtastwerten pro Schwingungsperiode $1/f_0$ ausreichend aufgelöst ist, resultieren beide Verfahren in dem erwarteten Ergebnis. Da eine Eigenmode eingekoppelt wird und der Wellenleiter sich in Ausbreitungsrichtung nicht ändert, ist unter Vernachlässigung von der Wellenleiterdispersion zu erwarten, dass sich die Form der Impulse bei der Propagation nicht ändert. Nun wird der Zeitschritt auf $\Delta t = 30\Delta t_{\text{CFL}}$ erhöht. Die Ergebnisse dieser zweiten Simulation sind in Abbildung 6.7 dargestellt. Hier ist die Trägerschwingung nur noch mit $3,536$ Abtastpunkten pro Schwingungsperiode abgetastet. Im Vergleich zu den Ergebnissen in Abbildung 6.6 fällt direkt die Verzerrung des Impulses über die Simulationszeit auf, welche bei der konventionellen Faberpolynom-Methode auftreten. Bei dem Feldbild für die Komplexe-Einhüllenden-Methode ist keine Änderung zu erkennen. Auffallend ist, dass die Form des Modenfeldes entlang der y-Achse in allen Fällen erhalten bleibt. Dies liegt darin begründet, dass zwar der zeitliche Verlauf bei einer zu geringen Approximationsordnung beziehungsweise einer zu hohen Zeitschrittweite fehlerhaft wiedergegeben wird, die Propagation der Felder selber aber immer sehr präzise erfasst wird. Die Propagation der Felder wird durch den linearen Teil in (6.3) beziehungsweise (6.16) beschrieben, welcher in beiden Fällen mit einer sehr hohen Genauigkeit evaluiert wird. Die in den Abbildungen 6.6 und 6.7 anschaulich erkennbaren Ergebnisse können noch weiter präzisiert werden. Hierzu werden die Feldverteilungen für $x = x_p$ entlang der y-Achse für jeden Zeitpunkt t nach dem Modenfeld entwickelt. Daraus resultiert jeweils ein Zeitverlauf eines Anregungskoeffizienten $a(t)$ bezüglich der Grundmode des Wellenleiters. Die Zeitverläufe sind in Abbildung 6.8 grafisch dargestellt. Die obigen Beobachtungen werden hier noch deutlicher. Während die Verläufe für die niedrigen Zeitschritte noch gut übereinstimmen, kommt es für große Zeitschritte Δt zu Fehlern bei der konventionellen Faberpolynom-Methode. In Abbildung 6.9 ist der relative Fehler des Anregungskoeffizienten für verschiedene Zeitschrittweiten aufgetragen. Da hier

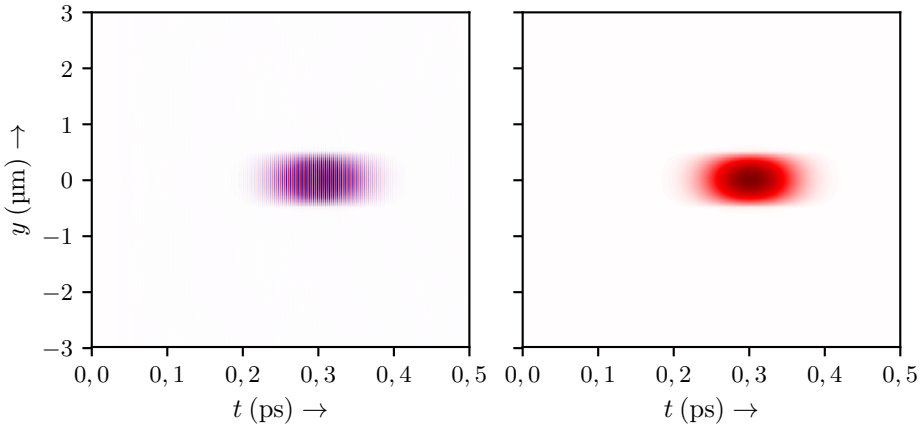


Abbildung 6.6: Die Abbildungen zeigen die H_z-Komponente in der y-Ebene für $x = x_p$ über die Simulationszeit. Auf der linken Seite wird die Faberpolynom-Methode dargestellt, während auf der rechten Seite die Komplexe-Einhüllende-Methode gezeigt wird. Für die letztere ist der Absolutbetrag der Einhüllenden abgebildet. Der verwendete Zeitschritt ist $\Delta t = \Delta t_{\mathrm{CFL}}$.

die Anregung einer Wellenleitermode betrachtet wird, soll hier auch der Anregungskoeffizient der anzuregenden Wellenleitermode als Fehlermaß herangezogen werden. Die Referenzlösung ist hierbei eine Simulation mit einer sehr geringen Zeitschrittweite, wobei der Fehler auf die maximale Amplitude von a_{ref} normiert wird. Der Fehler des Anregungskoeffizienten an $x = x_p$ wird als mit $\epsilon_{rel} = 1/N \sum_{n=1}^{N} |a_{\mathrm{ref}}(t = t_n) - a(t = t_n)|/max(|a_{\mathrm{ref}}|)$ bestimmt. Außerdem ist zu beobachten, dass der Fehler für beide Methoden mit steigender Zeitschrittweite Δt größer wird. Hierbei ist der Fehler der Komplexe-Einhüllenden-Methode nicht nur am Anfang geringer, sondern auch in der Steigung über den Zeitschritt Δt. Die Ergebnisse stimmen daher gut mit denen der vorherigen Beobachtungen überein. Bei beiden Varianten treten für große Zeitschritte Schwankungen in dem Fehlerwert auf.

Alles in allem decken sich die numerischen Ergebnisse mit der Theorie aus Abschnitt 6.2.3, nämlich dass mithilfe der Komplexe-Einhüllenden-Methode bei sonst gleichen Approximationsgrad die Genauigkeit erhöht werden kann. Dies wird besonders bei hohen Zeitschrittweiten deutlich. Hier profitiert die Komplexe-Einhüllenden-Methode besonders davon, dass die im Vergleich zur Bandbreite B große Trägerfrequenz f_0 der Quellterme nicht mit approximiert werden muss. Ist die Bandbreite B noch kleiner im Vergleich zu f_0, so ist dieser Effekt noch ausgeprägter. Die bisherigen Untersuchungen verwenden eine niedrige Approximationsordnung. Insbesondere im Hinblick auf die Optimierungen in Abschnitt 6.2.3 und auf die Verwendung von höheren Zeitschritten sind höhere Ordnungen interessant. Diese werden im nächsten Abschnitt betrachtet.

Approximation mit höheren Ordnungen

Im Folgenden sollen die konventionelle Faberpolynom-Methode und die Komplexe-Einhüllenden-Methode bei der Verwendung von höheren Ordnungen für die Entwicklung der Quellterme im Hinblick auf die Genauigkeit untersucht werden.

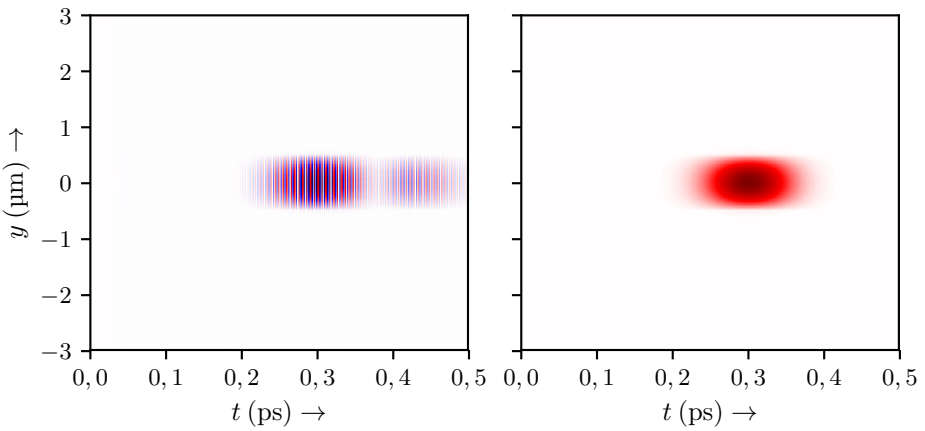


Abbildung 6.7: Die Abbildung zeigt die Ergebnisse der obigen Simulation mit einem Zeitschritt von $\Delta t = 30\Delta t_{\mathrm{CFL}}$.

Für die Untersuchung mit höheren Ordnungen wird ein weiteres Beispiel herangezogen. Hierbei wird ein zweidimensionales Gebiet in der x-y betrachtet, welches die Größe $L_x = L_y = 15\,\mu\mathrm{m}$ hat. Die TE-Mode wird untersucht. In dem Gebiet gilt $\epsilon_r = 1$ und $\mu_r = 1$. Das Gebiet ist mit einem FD-Yee-Gitter mit $\Delta x = \Delta y = 50\,\mathrm{nm}$ diskretisiert. Die Anregung einer Punktquelle an der Position $\vec{r}_i = (x_0, y_0)^T = (0{,}5\,\mu\mathrm{m}, 0{,}5\,\mu\mathrm{m})^T$ soll betrachtet werden. Hierbei befindet sich der Nullpunkt des Koordinatensystems in der Mitte des Simulationsgebietes. Diese wird mit $K_z = \sin(\omega_0/4t)\sin(\omega_0 t)$ beschrieben. Alle anderen Feldkomponenten von $\vec{\zeta}$ sind null. Hierbei gilt $f_0 = 193{,}4\,\mathrm{THz}$. Die Simulationszeit beträgt $T = 0{,}2\,\mathrm{ps}$.

Für die höheren Ordnungen wird der Taylorpolynom-Ansatz (6.7) für beide Methoden verwendet. Die Simulation wird mit beiden Ansätzen für verschiedene Zeitschrittweiten durchgeführt und für verschiedene Approximationsordnungen der Quellterme durchgeführt. Die Ergebnisse der Simulation mit einer Approximation der Quellterme bis zur dritten Ordnung sowie einem Zeitschritt von $\Delta t = 0.1\Delta t_{\mathrm{CFL}}$ dienen hier als Referenzlösung. Es wird der relative mittlere Fehler der H_z-Feldkomponente an dem Punkt $\vec{r}_P = (x_0, y_0)^T = (-5\,\mu\mathrm{m}, 5\,\mu\mathrm{m})^T$ mit $\epsilon_{rel} = 1/N \sum_{n=1}^{N} |H_{z,ref}(\vec{r} = \vec{r}_P, t = t_n) - H_z(\vec{r} = \vec{r}_P, t = t_n)|/|H_{z,ref}(\vec{r} = \vec{r}_P, t = t_n)|$ bestimmt. In Abbildung 6.10 sind die Ergebnisse der Simulationen zusammengefasst. In der Abbildung 6.10 kann beobachtet werden, dass der Fehler für höhere Entwicklungsordnungen reduziert werden kann. Außerdem ist der Fehler, wie erwartet, für kleine Zeitschrittweiten Δt geringer als bei großen. Wie in der vorangegangenen Untersuchung ist der Fehler bei der Komplexe-Einhüllenden-Methode bei gleicher Entwicklungsordnung in allen Fällen kleiner als bei der konventionellen Methode. Wieder kann für höhere Zeitschrittweiten eine Veränderung des Fehlerverlaufes festgestellt werden.

6.2.5 Diskussion

In den vorangegangenen Abschnitten wird eine Komplexe-Einhüllenden-Methode entwickelt. Bei dieser wird ein hochfrequenter Träger in die Systemmatrix des Modells aufgenommen, sodass eine komplexe Einhüllende anstelle des gesamten Feldes propagiert wird. Dies kann

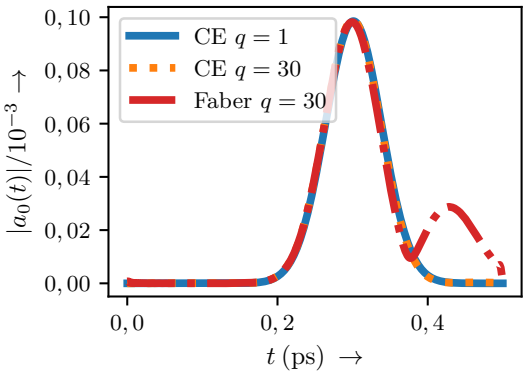


Abbildung 6.8: Der Anregungskoeffizient $a(t)$ bezüglich der Grundmode ist über die Zeit t für die verschiedenen numerischen Lösungen dargestellt. Der Zeitschritt ist mit $q = \Delta t/\Delta t_{\mathrm{CFL}}$ gegeben. Die Komplexe-Einhüllenden-Methode wird mit CE notiert. Das Modenfeld wird in $x = x_p$ gemessen.

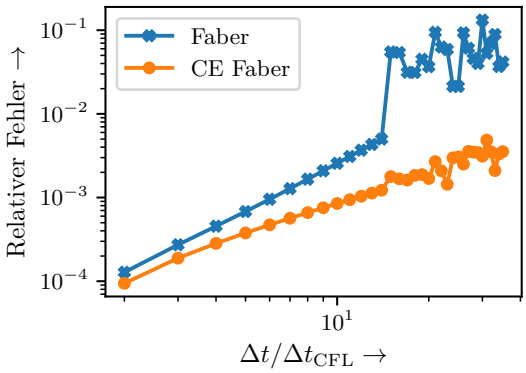


Abbildung 6.9: Der relative Fehler des Anregungskoeffizienten der Grundmode ist für die beiden Algorithmen für verschiedene Zeitschrittweiten Δt normiert mit Δt_{CFL} aufgetragen.

beispielsweise, wie in Abbildung 6.4 illustriert, hilfreich bei der Auswertung der Ergebnisse sein, da ein Überabtasten von Ausgängen nicht nötig ist. Für die Einbindung von Quellterme erweist sich der Komplexe-Einhüllenden-Ansatz aber als deutlich effizienter. Die Untersuchungen in 6.2.4 zeigen, dass die Verwendung der komplexen Einhüllenden signifikante Gewinne der Genauigkeit bei der Einbindung von Quellterme erlaubt. Das Einsatzgebiet sind hierbei Quellterme, welche bandbegrenzt um einen hoch oszillierenden Träger f_0 sind. In dem Zusammenhang werden einige praktische Anwendungsgebiete aufgezeigt, welche in dieses Einsatzgebiet fallen. Der Vorteil der Komplexe-Einhülleden-Methode liegt darin, dass die Schwingung des Trägers durch die Systemmatrix $\hat{\mathcal{H}}$ beschrieben wird, sodass nur noch die Einhüllende des Quellterms approximiert werden muss. Außerdem können für viele dieser Bereiche noch Optimierungen gefunden werden, welche die Einbindung der Quellterme auf Vektoradditionen statt zusätzlichen Matrixfunktionen reduzieren. Diese Eigenschaft reduziert den Rechenaufwand für die Einbindung erheblich. Der Einsatz von komplexer Arithmetik bei der Implementierung kann durch die Verwendung von (6.23) minimiert werden. Hervorzuheben ist außerdem, dass alle hier diskutierten Optimierungen keinerlei Approximationen verwenden, welche die Genauigkeit der Ergebnisse beeinflussen.

In den Fehlerkurven in den Abbildungen 6.9 und 6.10 ist zu erkennen, dass es bei hohen Zeitschrittweiten bei einigen der Verläufe zu einer Veränderung der Fehlerkurven kommt. Die Veränderung lässt sich durch numerische Fehler bei der Berechnung der Entwicklungskoeffizienten c_m der Faberpolynom-Approximation für die Quellterme begründen. Bei diesen müssen die in (6.7) auftretenden φ-Funktionen approximiert werden. Dafür liegt allerdings keine analytische Entwicklungsvorschrift wie für das Matrixexponential vor. Deshalb muss das Integral (5.8) mit einer numerischen Lösungsmethode bestimmt werden. Mit steigender Zeitschrittweite Δt wird

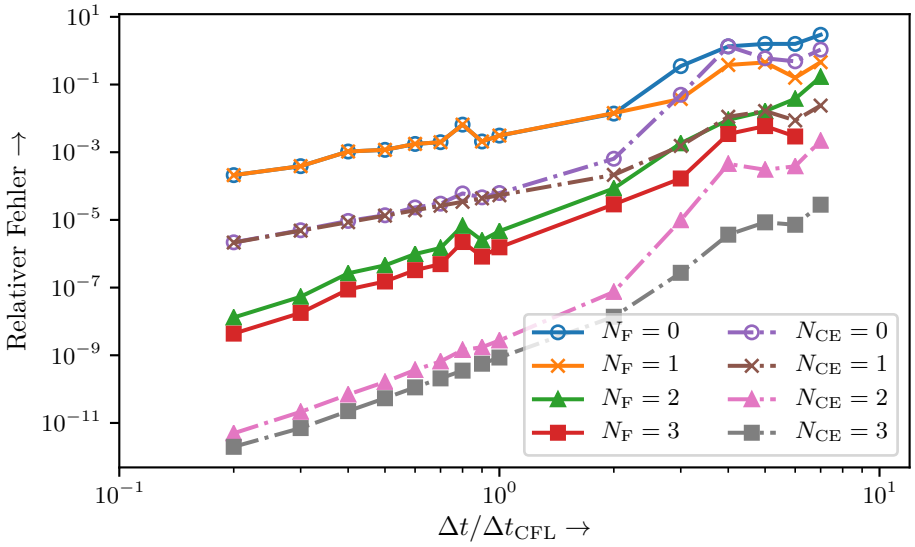


Abbildung 6.10: Die Abbildung zeigt den relativen Fehler des Verlaufes der H_z-Komponente im Punkt $\vec{r}_P$ über die verwendete Zeitschrittweite Δt. Bei der Untersuchung werden verschiedene Entwicklungsordnungen N_{F} für die konventionelle Faberpolynom-Methode sowie verschiedene Ordnungen N_{CE} für die Komplexe-Einhüllenden-Methode verwendet. Hierbei entsprechen $N_{\mathrm{F}} = 0$ und $N_{\mathrm{CE}} = 0$ jeweils dem Ansatz (6.17) beziehungsweise (6.4).

dies immer anspruchsvoller. Ein Lösungsansatz ist ein spezialisierter numerischer Löser oder, im bestem Fall, die Bestimmung einer analytischen Berechnungsvorschrift. In dem nächsten Abschnitt wird eine Methode vorgestellt, welche die Evaluation von φ-Funktionen komplett umgeht.

6.3 Entwicklung in die Systemmatrix

In diesem Abschnitt soll eine weitere Methode zur effizienten Einbindung von Quelltermen in den Blick genommen werden. Bei dem ersten Ansatz zu deren Einbindung in (6.7) treten je nach Approximationsgrad viele zusätzliche Matrixfunktionen auf. Im Folgenden wird ein Ansatz untersucht, welcher dies vollständig vermeidet. Statt (6.7) wird ein Matrixexponential mit $\exp(\Delta t\tilde{\mathcal{H}})$ mit einer etwas größeren Matrix $\tilde{\mathcal{H}}$ entwickelt. Dadurch müssen neben dem Matrixexponential keine zusätzlichen Matrixfunktionen mehr berechnet werden.

Der Ansatz basiert auf den Herleitungen in [103] und ist im Kontext von Faberpolynomen zur Lösung der Maxwell-Gleichungen mit Quelltermen noch nicht eingesetzt worden. Darüber hinaus ist der Einsatz nicht nur auf lineare Simulationen beschränkt. Eine Verwendung für nichtlineare Probleme ist auch möglich. Das soll in Kapitel 7 gesondert betrachtet werden. Aus diesen Gründen soll dieser Ansatz hier untersucht werden. Hierzu wird zunächst die Formulierung präsentiert. Im Anschluss wird der Ansatz numerisch evaluiert.

6.3.1 Formulierung

Der Ausgangspunkt ist eine Entwicklung des Problems (6.3) mit dem Ansatz aus Abschnitt 6.1: Die Entwicklung des örtlich diskretisierten Quellterms $\vec{\xi}(t)$ und das anschließende Einsetzen in (6.3) führt im Allgemeinen auf Ausdrücke, welche als Linearkombinationen von φ-Funktionen dargestellt werden können:

$$\vec{\Psi}(t_n + \Delta t) \approx \exp(\Delta t \mathcal{H})\vec{\Psi}(t_n) + \sum_{k=1}^{p} \varphi_k(\Delta t \mathcal{H})(\Delta t)^k \vec{w}_k. \tag{6.24}$$

Hier wird der Quellterm in (6.3) bis zu einem Grad p entwickelt. Eine Entwicklung dieser Form wird beispielsweise in (6.7) durch eine Taylorpolynom-Entwicklung von $\vec{\xi}(t)$ bestimmt. Statt nun die Matrixfunktionen φ_k zu approximieren, wird gemäß [103] eine erweiterte Matrix

$$\tilde{\mathcal{H}} = \begin{bmatrix} \mathcal{H} & \mathcal{W} \\ 0 & \mathcal{J} \end{bmatrix} \tag{6.25}$$

definiert. Für die ursprüngliche Systemmatrix gilt $\mathcal{H} \in \mathbb{C}^{N \times N}$, während $\tilde{\mathcal{H}} \in \mathbb{C}^{(N+p)\times(N+p)}$ für die Matrix $\tilde{\mathcal{H}}$ gilt. In diesem Zusammenhang ist $N >> p$, da N wie in Abschnitt 3.3 beschrieben, extrem große Werte annimmt, während p in der Praxis Werte von 1 bis 20 hat. Damit ist $\tilde{\mathcal{H}}$ nur unwesentlich größer als die ursprüngliche Systemmatrix $\mathcal{H}$. Die Matrix $\mathcal{J}$ hat die Größe $\mathcal{J} \in \mathbb{C}^{p \times p}$ und ist mit

$$\mathcal{J} = \begin{bmatrix} 0 & I_{p-1} \\ 0 & 0 \end{bmatrix} \tag{6.26}$$

gegeben. Hierbei ist I_{p-1} die Einheitsmatrix mit der Größe $(p-1) \times (p-1)$. Die Matrix $\mathcal{W} \in \mathbb{C}^{N \times p}$ ist mit

$$\mathcal{W} = [\vec{w}_1, \vec{w}_2, ..., \vec{w}_p] \tag{6.27}$$

gegeben. Für die Entwicklung des Quellterms $\vec{\xi}(t)$ nach Taylorpolynomen wie beispielsweise in (6.7) kann durch Vergleich von (6.7) mit (6.24) festgestellt werden, dass in diesem Fall $\vec{w}_k = \vec{u}_k$ gilt. Für andere Entwicklungen von $\vec{\xi}(t)$ ergeben sich andere Zusammenhänge für die Koeffizienten $\vec{w}_k$ in (6.27). Diese können analog durch einen Koeffizientenvergleich mit (6.24) bestimmt werden. Gemäß dem Theorem aus [103] gilt

$$\vec{\Psi}(t_n + \Delta t) = \begin{bmatrix} I_N & 0 \end{bmatrix} \exp(\Delta t \tilde{\mathcal{H}}) \begin{bmatrix} \vec{\Psi}(t_n) \\ \vec{e}_p \end{bmatrix}, \tag{6.28}$$

wobei $\vec{e}_p$ mit $\vec{e}_p = \begin{bmatrix} 0 & ... & 0 & 1 \end{bmatrix}^T$ gegeben ist. Mit diesem Konzept muss nur das Matrixexponential $\exp(\Delta t \tilde{\mathcal{H}})$ in jedem Zeitschritt evaluiert werden. Die Quellterme werden hierbei also anschaulich mit in die Systemmatrix aufgenommen. Damit eignet sich der Algorithmus insbesondere für die Verwendung mit Faberpolynomen, da keine zusätzlichen φ-Funktionen evaluiert werden müssen, sondern nur das Matrixexponential. Dies kann, wie in 5.4 demonstriert, effizient bestimmt werden. Im Anschluss wird auf die Implementierung des Ansatzes eingegangen sowie eine numerische Evaluation durchgeführt.

6.3.2 Implementierung und numerische Untersuchung

In diesem Abschnitt soll zunächst auf die Implementierung des Ansatzes mit Faberpolynom-Entwicklungen eingegangen werden. Im Anschluss wird das Verfahren numerisch evaluiert.

Implementierung

Um den neuen Algorithmus zu implementieren, muss das Matrixexponential in (6.28) mit Faberpolynomen approximiert werden. Hierzu werden Informationen bezüglich des Eigenwertspektrums $\sigma(\tilde{\mathcal{H}})$ der erweiterten Systemmatrix $\tilde{\mathcal{H}}$ benötigt. Nun stellt sich die Frage, inwiefern die Erweiterung das Eigenwertspektrum im Vergleich zu der ursprünglichen Matrix $\mathcal{H}$ verändert. Gemäß [116] lässt sich zeigen, dass

$$\sigma(\Delta t\tilde{\mathcal{H}}) = \sigma(\Delta t\mathcal{H}) \bigcup \{0\} \tag{6.29}$$

gilt. Also ändert sich das Eigenwertspektrum für die hier betrachten Matrizen $\mathcal{H}$ nicht. Damit sind die bisherigen Abschätzungen für das Eigenwertspektrum von $\mathcal{H}$ ausreichend und können für die Faberpolynom-Approximation von (6.28) verwendet werden. Ein weiterer Punkt ist die für die Faberpolynom-Approximation verwendete Skalierung, welche in Abschnitt 5.3 beschrieben wird. Diese wird im Anschluss an die Erweiterung der Systemmatrix $\mathcal{H}$ durchgeführt. Daher wird die erweiterte Systemmatrix $\tilde{\mathcal{H}}$ skaliert:

$$\begin{aligned} \vec{\Psi}(t_n + \Delta t) &= \begin{bmatrix} I_N & 0 \end{bmatrix} \exp(\Delta t\tilde{\mathcal{H}}) \begin{bmatrix} \vec{\Psi}(t_n) \\ \vec{e}_p \end{bmatrix} \\ &= \begin{bmatrix} I_N & 0 \end{bmatrix} \exp\left(\Delta t_s \begin{bmatrix} \mathcal{H}/\lambda_s & \mathcal{W}/\lambda_s \\ 0 & \mathcal{J}/\lambda_s \end{bmatrix} \right) \begin{bmatrix} \vec{\Psi}(t_n) \\ \vec{e}_p \end{bmatrix}. \end{aligned} \tag{6.30}$$

Mit diesen Vorüberlegungen lässt sich die Faberpolynom-Entwicklung für (6.30) durchführen. Da die Grenzen des Eigenwertspektrums sich im Vergleich zu $\mathcal{H}$ nicht verändern, ist auch die Entwicklungsordnung identisch. Allerdings wird durch die Erweiterung der Systemmatrix die Matrix-Vektor-Multiplikation $\tilde{\mathcal{H}}\vec{\Psi}$ aufwendiger im Vergleich zu $\mathcal{H}\vec{\Psi}$. Die Matrix $\tilde{\mathcal{H}}$ ist aufgrund $N >> p$ nur unwesentlich größer als $\mathcal{H}$. Die Systemmatrix $\tilde{\mathcal{H}}$ ist analog zu $\mathcal{H}$ dünnbesetzt. Eine vollbesetzte Matrix $\mathcal{W}$ würde dennoch zu einer signifikanten Erhöhung des Rechenaufwandes führen. An dieser Stelle bietet es sich daher an, die Konzepte aus den vorherigen Abschnitten zu verwenden und die Lokalisierung der Quellfunktionen miteinzubeziehen. Aufgrund dieser sind die Vektoren $\vec{w}_k$ in der Regel ebenfalls dünnbesetzt, sodass auch $\mathcal{W}$ dünnbesetzt ist.

Numerische Untersuchung und Diskussion

Um die Berechnungsmethode numerisch zu evaluieren, wird ein eindimensionales System betrachtet. Das Gebiet $z \in [0, L_z]$ mit $L_z = 50\,\mu\text{m}$ wird mit einer Schrittweite von $\Delta z = 10\,\text{nm}$ diskretisiert. Für das rechte Viertel des Simulationsgebiets gilt $\epsilon = 3\epsilon_0$. Für das restliche Gebiet gilt $\mu = \mu_0$ und $\epsilon = \epsilon_0$. Eine Punktquelle $J_x = \sin(\omega_0 t)$ wird in der Mitte des Simulationsgebiets platziert. Die Frequenz ist mit $f_0 = 200\,\text{THz}$ gegeben. Die Simulationszeit beträgt $T = 0{,}1\,\text{ps}$. Die Ergebnisse werden an $z_p = 35\,\mu\text{m}$ gemessen. Die Quellterme werden mit dem Taylorpolynom-Ansatz (6.7) mit verschiedenen Ordnungen p entwickelt. Die Ergebnisse sind in Abbildung 6.11 dargestellt. Die Referenzsimulation wird mithilfe des ADE-Ansatzes bestimmt [33]. Zur Fehlerberechnung wird analog zu den obigen Untersuchungen der Zeitverlauf der E_x-Komponente an der Stelle z_p herangezogen. Es wird jeweils der mittlere relative Fehler $\epsilon_{rel} = 1/N \sum_{n=1}^{N} |E_{x,ref}(z = z_p, t = t_n) - E_x(z = z_p, t = t_n)|/|E_{x,ref}(z = z_p, t = t_n)|$ bestimmt. Wie zu erwarten, steigt Abweichung mit steigender Zeitschrittweite. Durch eine Erhöhung der

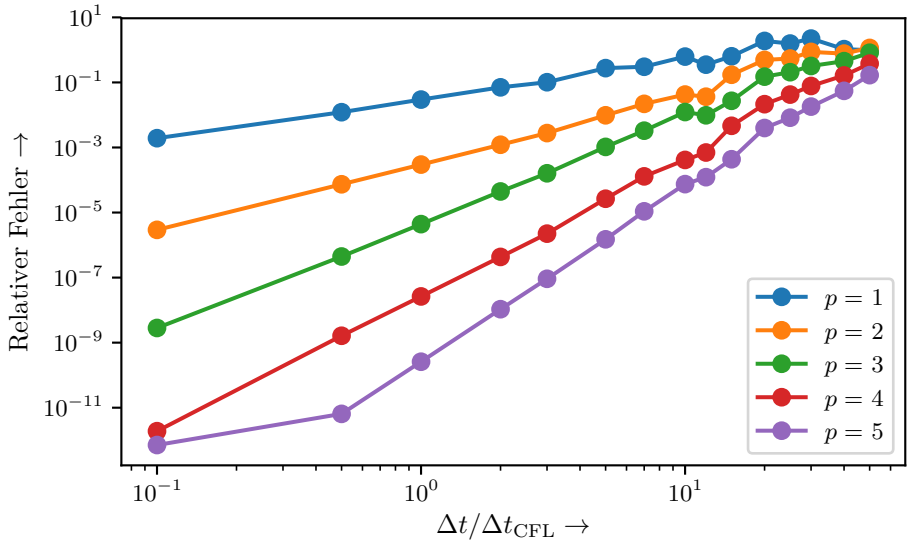


Abbildung 6.11: Die Abbildung zeigt den relativen Fehler des Ansatzes für eine Quelle mit harmonischer Zeitabhängigkeit. Der Fehler ist über die normierte Zeitschrittweite $\Delta t/\Delta t_{\mathrm{CFL}}$ für verschiedene Entwicklungsordnungen p aufgetragen.

Ordnung p wird der Fehler reduziert. Auffällig ist, dass durch die Vermeidung der Entwicklung der φ-Funktionen der Fehler auch für hohe Zeitschrittweiten niedrig bleibt und nicht übermäßig ansteigt, wie in Abschnitt 6.2.4 bei einigen Fällen beobachtet. Es werden zusätzliche Fehler bei der numerischen Entwicklung der φ-Funktionen vermieden, welche besonders bei hohen Zeitschrittweiten Δt aufgetreten sind.

6.4 Diskussion

In dem vorangegangenen Kapitel werden Ansätze zur Einbindung von Quelltermen in die Faberpolynom-Methode aus Kapitel 5 vorgestellt und untersucht. Die Zielsetzung liegt hierbei darin, die hohe Genauigkeit des linearen quellfreien Teils in (6.3) zu erhalten. Es zeigt sich, dass insbesondere bei hohen Zeitschrittweiten Δt und hoch oszillierenden Quelltermen eine hohe Entwicklungsordnung nötig ist. Dies führt zu zusätzlichen Matrixfunktionen, bei welchen es sich um Linearkombinationen aus φ-Funktionen handelt. Hierbei treten für den betrachteten Ansatz zwei Probleme auf: Das erste ist der größere Rechenaufwand für die Bestimmung der zusätzlichen Matrixfunktionen. Der Rechenaufwand für diese zusätzlichen Funktionen übertrifft den Aufwand des quellfreien Teils schon bei niedrigen Approximationsordnungen. Das zweite Problem zeigt sich bei der Entwicklung der φ-Funktionen mit den Faberpolynomen. Werden die Faberpolynom-Koeffizienten für diese Funktionen numerisch bestimmt, können zusätzliche Fehler auftreten, was insbesondere hohe Zeitschrittweiten betrifft. Um diese Probleme zu lösen, werden verschiedene Lösungsansätze vorgeschlagen.

Der Ansatz in Abschnitt 6.1.2 zielt auf die Reduzierung der Kosten für die Evaluierung der zusätzlichen Matrixfunktionen. Hierzu wird die Lokalisierung der Quellterme für die Reduzierung

der Systemmatrix verwendet. Die spektralen Eigenschaften der reduzierten Matrix lassen sich wiederum bei der Entwicklung berücksichtigen. Das Potenzial zur Rechenzeitreduzierung hängt maßgeblich von der Beschaffenheit des Quellterms und des Rechengebietes ab.

In Abschnitt 6.2 wird die Idee, die Eigenschaften der Quellfunktionen zu verwenden, noch erweitert. Neben der örtlichen Verteilung wird nun auch die Zeitabhängigkeit berücksichtigt. Zu diesem Zweck wird eine hochfrequente Trägerschwingung in die Systemmatrix mit aufgenommen. Diese Schwingung muss dadurch nicht mehr approximiert werden, was die Entwicklungsordnung, um eine bestimmte Genauigkeit zu erreichen, stark reduziert. Außerdem erlaubt die Komplexe-Einhüllenden-Methode in vielen praktischen Fällen die vollständige Vermeidung von zusätzlichen Matrixfunktionen während der Propagation. Diese müssen nur einmal im Voraus berechnet werden, während bei der Propagation nur einfache gewichtete Vektoradditionen durchgeführt werden müssen. Dies ist auch bei höheren Approximationsordnungen möglich.

Der dritte Ansatz basiert auf einem Theorem aus [103] und erlaubt die komplette Vermeidung der φ-Funktionen. Stattdessen wird das Matrixexponential mit einer etwas größeren Systemmatrix berechnet. Auch hier bietet es sich an, die Lokalisierung der Quellfunktionen bei der Implementierung zu berücksichtigen. Prinzipiell kann der Ansatz aber auf beliebige Quellfunktionen angewendet werden. Damit ist dieser Ansatz insbesondere für Quellfunktionen interessant, bei denen keine der oben beschriebenen Eigenschaften genutzt werden können. Gegenüber den ADE-Ansätzen unterscheidet sich dieser Ansatz durch seine allgemeinere Formulierung, da er direkt auf der Formulierung mit den φ-Funktionen in (6.24) aufbaut.

Insgesamt ermöglichen die hier vorgestellten Methoden eine präzise Einbindung von Quelltermen. Die Methoden erlauben die Erhaltung der hohen Genauigkeit des linearen Teils. Darüber hinaus bewirkt die Einbindung der Quellterme mit den beschriebenen Methoden, dass keine signifikante Erhöhung der Rechenzeit vorliegt. Dieses Ergebnis wird durch die konsequente Nutzung der Eigenschaften von den Quellfunktionen erreicht. Dabei erweist es sich als sinnvoll, die Quellfunktionen mit spezialisierten Algorithmen zu bestimmen und die Matrixfunktionen in (6.24) nicht direkt zu entwickeln. Damit ist nun die Betrachtung der Maxwell-Gleichungen mit Quelltermen mit der Faberpolynom-Methode möglich. Ihre Anwendbarkeit wird hierdurch auf ein breites Spektrum von technischen Problemen erweitert. Die verbleibende Einschränkung ist, dass bisher nur lineare Materialmodelle betrachtet werden können. Daher soll im folgenden Kapitel die Einbindung von nichtlinearen Effekten in den Blick genommen werden.

7 Nichtlineare Effekte

Nachdem in Kapitel 5 lineare, quellfreie Systeme betrachtet werden und in Kapitel 6 auf die Einbindung von Stromtermen eingegangen wird, sollen in diesem Kapitel nichtlineare Effekte mit Faberpolynom basierten Algorithmen zur Zeitpropagation untersucht werden.

Zu den klassischen Beispielen für nichtlineare Effekte gehören der Pockels- oder der Kerr-Effekt, welche in Abschnitt 2.2.3 skizziert werden. Sollen außerdem verstärkende Medien betrachtet werden, müssen für eine physikalische Modellierung Sättigungseffekte berücksichtigt werden. Hier kann als erste Näherung das Zwei-Niveau-Modell, welches in Abschnitt 2.2.3 beschrieben wird, verwendet werden. Darüber hinaus erlaubt das Zwei-Niveau-Modell auch eine Modellierung von Absorbern mit Sättigungsverhalten. Des Weiteren lassen sich durch die vektorielle Beschreibung der Felder ohne weitere Annahmen Phänomene erfassen, welche sich mit der sonst oft verwendeten nichtlinearen Schrödingergleichung nicht korrekt darstellen lassen [117].

Im Kontext der konventionellen FDTD-Methode gibt es verschiedene Ansätze zur Einbindung von Nichtlinearitäten. Viele dieser Ansätze führen auf implizite Ausdrücke, welche mit einem Newton-Verfahren gelöst werden [117–119]. Deren Berechnung ist für große dreidimensionale Probleme sehr aufwendig [120, 121]. Für einige Fälle ist es zwar möglich, explizite Ansätze zu finden, im Allgemeinen muss aber auf das Newton-Verfahren zurückgegriffen werden. Um dies zu umgehen, sind in der Literatur einige Ansätze untersucht worden. Hier sind insbesondere die in [120, 121] untersuchten Algorithmen interessant, da sie eine explizite Formulierung erlauben. Diese wird durch eine physikalisch motivierte Kopplung der Nichtlinearität mit einem Lorentz-Oszillator erreicht.

Die Einbindung von nichtlinearen Effekten in elektrodynamischen Problemstellungen mit der Faberpolynom-Methode ist in der Literatur bisher noch nicht thematisiert worden. In [122] wird die Verwendung von Exponential-Integratoren auf Basis von Krylov-Unterraum-Methoden für die Lösung von nichtlinearen elektrodynamischen Problemen vorgestellt. In den folgenden Abschnitten wird gezeigt, dass es mit dem bisherigen Formalismus und der effizienten Approximation des linearen Teils des Systems möglich ist, die Ergebnisse aus der Mathematik im Kontext der Exponential-Integratoren zu nutzen [105, 123–126]. So können auf Basis der Vorüberlegungen in den vorangegangenen Abschnitten Algorithmen mit hohen Ordnungen bestimmt werden. Außerdem erlauben die untersuchten Formalismen weiterhin eine explizite Berechnung. So ist eine direkte parallele Implementierung möglich.

Für die Einbindung von Nichtlinearitäten in den Faberpolynom basierten Algorithmus müssen die nichtlinearen Terme zunächst geeignet formuliert werden. Hierbei gibt es verschiedene Ansätze, auf welche zuerst eingegangen wird. Im Anschluss wird die Implementierung diskutiert. In Abschnitt darauf werden die verschiedenen Algorithmen anhand eines numerischen Beispiels miteinander verglichen. Teile der hier vorstellten Untersuchungen sind in [KS11] zur Veröffentlichung eingereicht.

7.1 Modellierung von Nichtlinearitäten

In diesem Abschnitt soll auf die Beschreibung der Nichtlinearitäten eingegangen werden. In der Literatur gibt es hierzu eine Vielzahl von Ansätzen. An dieser Stelle soll die Betrachtung auf eine Auswahl begrenzt werden, welche eine effiziente Implementierung mit der Faberpolynom-Approximation erlaubt. Die Methoden unterscheiden sich zum einen in der Formulierung des zu lösenden Problems und zum anderen in der Art, in der die Nichtlinearität für den Zeitschritt linearisiert wird. Im allgemeinen Fall haben die betrachteten nichtlinearen Probleme die Form

$$\frac{\partial \vec{\Psi}(t)}{\partial t} = F(\vec{\Psi}(t)). \tag{7.1}$$

Hierbei handelt es sich bei (7.1) um das bereits örtlich, aber noch nicht zeitlich diskretisierte System. Um mithilfe von Exponential-Integratoren berechnet werden zu können, muss das System in einer semilinearen Schreibweise der Form

$$\frac{\partial \vec{\Psi}(t)}{\partial t} = \mathcal{H}\vec{\Psi}(t) + \mathcal{N}(\vec{\Psi}(t)) \tag{7.2}$$

vorliegen [104, 122]. Hierzu ist eine Linearisierung von (7.1) nötig. Im allgemeinen Fall können zwei Methoden verwendet werden. Der erste Ansatz basiert auf einer festen Approximation der Jakobi-Matrix $\mathcal{H}$ des kompletten linearen Teils [104, 105, 122, 125]. Die Jakobi-Matrix $\mathcal{H}$ ändert sich also nicht mit dem Zeitschritt und der lineare Teil ist vollständig von dem nichtlinearen Restglied $\mathcal{N}(\vec{\Psi}(t))$ getrennt. Die Jakobi-Matrix, welche den linearen Teil des Systems beschreibt, kann entweder geeignet approximiert oder, wenn möglich, analytisch bestimmt werden. Eine mögliche Approximation ist:

$$\mathcal{H} \approx \frac{\partial F}{\partial \vec{\Psi}}(\vec{\Psi}(t_0)). \tag{7.3}$$

Eine Approximation wie (7.3) ist möglich, wenn sich das System nicht weit aus dem anfänglichen Zustand herausbewegt. Andernfalls kommt es zu zusätzlichen Fehlern, da für $\vec{\Psi}(t_0)! = \vec{\Psi}(t)$ die Approximation (7.3) zunehmend von der analytischen Lösung abweicht. In vielen Fällen lässt sich eine Darstellung (7.2) auch analytisch finden. Diese ist immer dann sinnvoll, wenn das nichtlineare Restglied gegenüber dem linearen Teil klein ist. Hier soll nur die analytische Variante verwendet werden.

Die zweite Möglichkeit ist, es diese Linearisierung zu jedem Zeitschritt dynamisch zu bestimmen. Hierzu muss die Jakobi-Matrix in jedem Zeitschritt bestimmt werden:

$$\mathcal{H} \approx J_n = DF(\vec{\Psi}(t_n)). \tag{7.4}$$

Diese wird mit dem nichtlinearem Restglied, welches mit

$$R(\vec{\Psi}(t)) = F(\vec{\Psi}(t)) - J_n\vec{\Psi}(t) \tag{7.5}$$

gegeben ist, zur Berechnung des nächsten Zeitschritts verwendet [104, 125].

Die Verwendung einer festen approximierten Jakobi-Matrix wie in (7.3) ist nur dann sinnvoll, wenn die später auftretenden Matrixexponentiale effizient vorberechnet und immer wieder verwendet werden können. In dem hier betrachteten Anwendungsfall ist dies, wie in den vorangegangen Kapiteln 4 und 5 schon festgestellt, praktisch nicht möglich. Außerdem wird bei

der Approximation mit Faberpolynomen das Produkt $\exp(\Delta t\mathcal{H})\vec{\Psi}(t)$ bestimmt und für jede Berechnung neu approximiert. Daher ist eine Approximation (7.3) aufgrund der zusätzlichen Fehler durch die fixe Approximation hier nicht sinnvoll. Aus diesem Grund werden im Folgenden der Ansatz mit einer variablen Jakobi-Matrix (7.4) und nichtlinearem Restglied oder, wenn möglich, analytische Jakobi-Matrizen verwendet. Wenn das System in der Form (7.2) vorliegt, kann wie bei der Betrachtung der Quellterme mit

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t\mathcal{H})\vec{\Psi}(t_n) + \int_0^{\Delta t} \exp((\Delta t - \tau)\mathcal{H})\mathcal{N}(\vec{\Psi}(t_n + \tau))d\tau \tag{7.6}$$

die formale Lösung des Systems angegeben werden [104]. Hier wird eine feste, zum Beispiel analytisch bestimmte, Jakobi-Matrix $\mathcal{H}$ verwendet werden. Durch die Linearisierung $\mathcal{N}(\vec{\Psi}(t))$ in $t = t_n$ kann analog zu den Betrachtungen bei den Quelltermen in Kapitel 6 ein erstes Lösungsverfahren bestimmt werden [104]:

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t\mathcal{H})\vec{\Psi}(t_n) + \Delta t\varphi(\Delta t\mathcal{H})\mathcal{N}(\vec{\Psi}(t_n)). \tag{7.7}$$

Dieses Verfahren wird Exponential-Euler-Verfahren genannt. Generell fällt mit (7.6) und (7.7) die Ähnlichkeit zu der Beschreibung von Quelltermen in Kapitel 6 auf. In beiden Fällen führen die Approximationen auf Ausdrücke mit gewichteten Summen von φ-Funktionen. Die Einbindung unterscheidet sich allerdings in einigen entscheidenden Punkten. Der wichtigste Unterschied ist, dass bei den Nichtlinearitäten die Funktion $\mathcal{N}(\vec{\Psi}(t))$ im Integralterm der formalen Lösung (7.6) nur zum Zeitpunkt $t = t_n$ bekannt ist. Bei den Quelltermen ist der Verlauf der Funktion $\xi\zeta(t)$ über den gesamten Zeitschritt $t \in [t_n, t_n + \Delta t]$ im Vorfeld bekannt. Das führt dazu, dass bei den Nichtlinearitäten zur Bestimmung von höheren Ordnungen mehrstufige Verfahren nötig sind [104]. Bei den Quelltermen, hingegen, können durch eine geeignete Approximation von $\zeta(t)$ und das anschließende Einsetzen in den das Integral (6.3) höhere Approximationsordnungen erzielt werden.

Der zweite Punkt ist die effiziente Implementierung. In Kapitel 6 wird gezeigt, dass für die Einbindung von Quelltermen ihre Eigenschaften ausgenutzt werden können, um den Berechnungsaufwand für die zusätzlichen Matrixfunktionen in Form von φ-Funktionen massiv zu reduzieren. Daher lohnt es sich, die Quellfunktionen mit spezialisierten Algorithmen zu bestimmen und nicht wie Nichtlinearitäten zu behandeln. Bei den Nichtlinearitäten ist dies im Allgemeinen nicht möglich, sodass deren Einbindung zu einem deutlich höheren Rechenaufwand führt.

7.2 Algorithmen zur Einbindung mit Faberpolynomen

Im Folgenden soll auf einige für die Faberpolynom-Methode geeignete Algorithmen eingegangen werden. Hier wird die Auswahl auf die Klasse der Lawson-Integratoren und die Exponential-Rosenbrock-Verfahren beschränkt. Die Ansätze und ihre Eigenschaften werden im Folgenden näher erläutert.

Lawson-Verfahren

Diese Klasse von Methoden ist zuerst von Lawson untersucht worden [127]. Der Ausgangspunkt für die Lawson-Integratoren ist ein System der Form (7.2). Die Klasse der Lawson-Integratoren

basiert auf der Einführung der folgenden Transformation [105, 122]:

$$v(t) = \exp(-t\mathcal{H})\vec{\Psi}(t). \tag{7.8}$$

Auf diese Weise wird die Abhängigkeit von der Jakobi-Matrix zunächst entfernt [104, 105]:

$$\frac{\partial v(t)}{\partial t} = \mathcal{F}(v(t), t) = \exp(-t\mathcal{H})\mathcal{N}(\exp(+t\mathcal{H})v(t)). \tag{7.9}$$

Auf diese Gleichung kann dann ein geeignetes Lösungsverfahren wie zum Beispiel das explizite Euler-Verfahren angewendet werden:

$$v(t_n + \Delta t) = v(t_n) + \Delta t\mathcal{F}(v(t_n), t_n). \tag{7.10}$$

Die Rücktransformation führt auf den entsprechenden Lawson-Integrator [104, 105, 122]:

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t\mathcal{H})\vec{\Psi}(t_n) + \Delta t \exp(\Delta t\mathcal{H})\mathcal{N}(\vec{\Psi}(t_n)). \tag{7.11}$$

Bei (7.11) handelt es um das Lawson-Euler-Verfahren. Höhere Lösungsordnungen können durch die Verwendung von Lösungsverfahren höherer Ordnung auf (7.9) bestimmt werden. Durch die Anwendung eines Runge-Kutta-Verfahrens vierter Ordnung kann beispielsweise der folgende Algorithmus bestimmt werden [122]:

$$\begin{aligned}
Y_1 =& \vec{\Psi}(t_n) \\
Y_2 =& \Delta t/2 \exp(\Delta t\mathcal{H})\mathcal{N}(Y_1, t_n) + \exp(\Delta t/2\mathcal{H})Y_1 \\
Y_3 =& \Delta t/2\mathcal{N}(Y_2, t_{n+1/2}) + \exp(\Delta t/2\mathcal{H})Y_1 \\
Y_4 =& \Delta t \exp(\Delta t/2\mathcal{H})\mathcal{N}(Y_3, t_{n+1/2}) + \exp(\Delta t\mathcal{H})Y_1 \\
\vec{\Psi}(t_n + \Delta t) =& \Delta t/6 \Big(\exp(\Delta t\mathcal{H})\mathcal{N}(Y_1, t_n) + 2\exp(\Delta t/2\mathcal{H})\mathcal{N}(Y_2, t_{n+1/2}) \\
&+2\exp(\Delta t/2\mathcal{H})\mathcal{N}(Y_3, t_{n+1/2}) + \mathcal{N}(Y_4, t_{n+1})\Big) + \exp(\Delta t\mathcal{H})Y_1.
\end{aligned} \tag{7.12}$$

Die Lawson-Integratoren werden hier mit einer festen Jakobi-Matrix realisiert. Hierbei wird eine analytische Aufteilung des Systems in den linearen und den nichtlinearen Teil verwendet.

Exponential-Rosenbrock

Diese Methode verwendet die Definition der Jakobi-Matrix mit der lokalen Linearisierung (7.4) und dem nichtlinearem Restglied (7.5) in jedem Schritt [104, 125]:

$$\frac{\partial \vec{\Psi}(t)}{\partial t} = J_n\vec{\Psi}(t) + R_n. \tag{7.13}$$

Die Besonderheit der Exponential-Rosenbrock-Verfahren ist, dass diese die kontinuierliche Linearisierung explizit nutzen [104, 125]. So lässt sich ausgehend von (7.13) mit

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta tJ_n)\vec{\Psi}(t_n) + \Delta t\varphi(\Delta tJ_n)R(\vec{\Psi}(t_n)) \tag{7.14}$$

das Exponential-Rosenbrock-Euler-Verfahren angegeben [104]. Integratoren höherer Ordnung können bestimmt werden, indem Runge-Kutta-Methoden auf (7.13) angewendet werden [104]:

$$\begin{aligned} U_{n,i} &= \exp(c_i \Delta t J_n)\vec{\Psi}(t_n) + \Delta t \sum_{j=1}^{i-1} a_{i,j}(\Delta t J_n) R_n(U_{n,j}) \\ \vec{\Psi}(t_n + \Delta t) &= \exp(\Delta t J_n)\vec{\Psi}(t_n) + \Delta t \sum_{i=1}^{s} b_i(\Delta t J_n) R_n(U_{n,i}). \end{aligned} \tag{7.15}$$

Für die Bestimmung der Koeffizienten für bestimmte Integratoren sei auf die Literatur verwiesen [104].

7.3 Implementierung

In diesem Abschnitt soll auf die Realisierung der oben vorgestellten Ansätze in Kombination mit Faberpolynomen eingegangen werden. Mit Ausnahme der Lawson-Ansätze müssen bei den betrachten Algorithmen gewichtete Summen aus φ-Funktionen bestimmt werden. Daher bietet sich der Ansatz aus Abschnitt 6.3 an. Mit diesem Ansatz muss nur eine Matrixfunktion pro Stufe bestimmt werden. Aufgrund von der Bedingung (6.29) muss bei der Approximation mit Faberpolynomen nur das Eigenwertspektrum der Jakobi-Matrix $\mathcal{H}$ berücksichtigt werden. Allerdings kann es insbesondere bei der Einbindung von Nichtlinearitäten dazu kommen, dass die Norm $||\mathcal{W}||$ der Matrix $\mathcal{W}$ sehr große Werte annimmt. Dies kann zu numerischen Fehlern und daraus folgenden Instabilitäten führen [103, 116]. Um dies zu vermeiden, wird hier die in [116] vorgeschlagene Normierung verwendet. Hierbei sei die gewichtete Summe aus φ-Funktionen in der Form

$$\vec{\Psi}(t_n + \Delta t) \approx \exp(\Delta t \mathcal{H})\vec{\Psi}(t_n) + \Delta t \sum_{k=1}^{p} \varphi_k(\Delta t \mathcal{H})\vec{\hat{w}}_k \tag{7.16}$$

gegeben. Mit der Skalierung der Faberpolynom-Approximation ist die Berechnungsvorschrift dann mit

$$\vec{\Psi}(t_n + \Delta t) = \begin{bmatrix} I_N & 0 \end{bmatrix} \exp\left(\Delta t_s \begin{bmatrix} \mathcal{H}/\lambda_s & \mathcal{W}/\lambda_s \\ 0 & \mathcal{J}/\Delta t/\lambda_s \end{bmatrix} \right) \begin{bmatrix} \vec{\Psi}(t_n) \\ \vec{e}_p \end{bmatrix} \tag{7.17}$$

gegeben, wobei $\mathcal{W} = [\vec{\hat{w}}_1, \vec{\hat{w}}_2, ..., \vec{\hat{w}}_p]$. Im Anhang C.2 ist die Bestimmung der Koeffizienten der $\mathcal{W}$-Matrix beispielhaft für das Rosenbrock-Euler-Verfahren sowie für das **exprb32**-Verfahren aus [104], welches ein Verfahren dritter Ordnung ist, gegeben.

7.4 Evaluation der Ansätze

In dem vorangegangenen Abschnitt werden verschiedene Algorithmen auf Basis von Faberpolynomen vorgestellt, welche es erlauben, nichtlineare Effekte in die Zeitbereichssimulation aufzunehmen. Wie bei der Berücksichtigung von Quelltermen kann die Einbindung in zusätzlichen Matrixfunktionen resultieren, welche während der Zeitpropagation ausgewertet werden müssen.

Die Verwendung von Verfahren höherer Ordnung ermöglichen allerdings eine höhere Genauigkeit der Ergebnisse und bergen das Potenzial, höhere Zeitschritte anwenden zu können. Im Folgendem sollen daher die Verfahren auf ihre Effizienz im Hinblick auf die Genauigkeit der Ergebnisse und den Rechenaufwand untersucht werden. Als Maß für den Rechenaufwand soll wie in den Untersuchungen der letzten Kapitel die Anzahl der Matrix-Vektor-Multiplikationen dienen. Zur Evaluation soll ein Material verwendet werden, welches eine Pockels-Nichtlinearität aufweist [26] . Hier wird das Modell, das in [121] vorgeschlagen wird, verwendet. Das Modell in [121] lässt sich wie folgt in der Operator-Schreibweise angeben und in den linearen und nichtlinearen Teil aufteilen:

$$\frac{\partial}{\partial t}\vec{\psi}(t) = \begin{bmatrix} 0 & \frac{1}{\epsilon_0\epsilon_\infty}\nabla\times & 0 & \frac{-1}{\epsilon_0\epsilon_\infty} \\ -\frac{1}{\mu}\nabla\times & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \epsilon_0\omega_L^2\chi^{(1)} & 0 & -\omega_L^2 & -\gamma_L \end{bmatrix}\vec{\psi}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega_L^2\epsilon_0\chi^{(2)}\vec{E}(t)^2 \end{bmatrix}. \tag{7.18}$$

Dabei gilt $\vec{\psi}(t) = [\vec{E}(t), \vec{H}(t), \vec{P}(t), \vec{J}(t)]^T$. Bei (7.18) handelt es sich um das System vor der örtlichen Diskretisierung. Soll die Formulierung mit der dynamisch bestimmten Jakobi-Matrix (7.4) werden, so ist diese mit

$$J_n = \frac{\partial F}{\partial \vec{\psi}}(\psi(t_n)) = \begin{bmatrix} 0 & \frac{1}{\epsilon_0\epsilon_\infty}\nabla\times & 0 & \frac{-1}{\epsilon_0\epsilon_\infty} \\ -\frac{1}{\mu}\nabla\times & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \epsilon_0\omega_L^2(\chi^{(1)} + 2\chi^{(2)}\vec{E}(t_n)) & 0 & -\omega_L^2 & -\gamma_L \end{bmatrix} \tag{7.19}$$

gegeben. Das nichtlineare Restglied ist mit

$$R(\psi(t)) = F(\psi(t)) - J_n\vec{\psi}(t) = \begin{bmatrix} 0 & 0 & 0 & \epsilon_0\omega_L^2(\chi^{(2)}\vec{E}(t)^2 - 2\chi^{(2)}\vec{E}(t_n))\vec{E}(t) \end{bmatrix}^T \tag{7.20}$$

zu bestimmen.

Als Beispiel wird ein eindimensionales System mit der Länge $L_z = 85\,\mu\text{m}$ gewählt, welches gleichmäßig mit einer Schrittweite von $\Delta z = 10\,\text{nm}$ abgetastet wird. Zur Ortsdiskretisierung wird ein pseudospektraler Ansatz angewendet, um die numerische Dispersion durch die Ortsdiskretisierung zu minimieren. Andernfalls kann es bei einer ausgeprägten numerischen Dispersion zu einer Interaktion mit den nichtlinearen Effekten kommen, welche die Ergebnisse verfälschen.

Die rechte Hälfte des Simulationsgebietes ist mit einem Material gefüllt, welches durch das Pockels-Modell in (7.18) beschrieben wird. Bei diesem gilt $\chi^{(1)} = 2,427\epsilon_0$, $\chi^{(2)} = 30\,\text{pV/m}$ sowie $\omega_L = 1{,}549 \times 10^{16}$ und $\gamma_L = 0$. In dem restlichen Bereich liegt eine Permittivität von $\epsilon = 5,226\epsilon_0$ vor. Die Propagation eines gaußförmigen Impulses wird betrachtet. Der Impuls hat eine Bandbreite von $B = 30\,\text{THz}$ und ist mit einer Trägerfrequenz von $f_0 = 200\,\text{THz}$ moduliert. Der Impuls hat eine Amplitude von $E_0 = 1000\,\text{GV/m}$. Bei der Untersuchung wird eine derart hohe Amplitude gewählt, um innerhalb kurzer Simulationszeiten einen möglichst starken nichtlinearen Effekt zu erzielen. Der Impuls wird in der linken Hälfte des Rechengebiets platziert und so initialisiert, dass er in Richtung des nichtlinearen Materials propagiert. Die Simulationszeit beträgt $T = 0{,}45\,\text{ps}$.

Das Exponential-Euler-Verfahren (7.7), das Lawson-Euler-Verfahren (7.11), das Lawson-Verfahren vierter Ordnung (7.12), das Rosenbrock-Euler-Verfahren (7.14) sowie das Rosenbrock-Verfahren **exprb32** dritter Ordnung aus [104] sollen verglichen werden. Bei der Berechnung

der Lawson-Verfahren treten nur Matrixexponentiale auf, welche mithilfe von Faberpolynomen bestimmt werden. Die Linearkombinationen aus φ-Funktionen, welche bei den übrigen Methoden auftreten, werden mit dem Algorithmus (7.17) bestimmt. Alle Faberpolynom-Approximationen werden durchgeführt, bis $|c_m| < 10^{-15}$ für die Koeffizienten gilt. Der Vergleichsalgorithmus ist ein klassischer FD-Ansatz für die Zeitabhängigkeit. Dessen Verwendung ist aufgrund des in [121] vorgestellten Materialmodells möglich. Daher handelt es sich bei dem Vergleichsansatz auch um einen expliziten Algorithmus.

Die Algorithmen werden mit verschiedenen Zeitschrittweiten Δt durchgeführt. Die Ergebnisse sind in Abbildung 7.1 dargestellt. Als Referenz wird hier eine Simulation auf Basis des Lawson-Verfahrens mit sehr kleiner Zeitschrittweite verwendet. In den vorangegangenen Untersuchungen bezüglich der Faberpolynom-Methode für lineare Medien ohne Quellen sowie mit Quellen werden für die Fehlerberechnung Zeitverläufe einer Feldkomponente an einem Ortspunkt herangezogen. Eine fehlerhafte Beischreibung der Quellterme tritt hierbei konzentriert bei deren Einkoppelpunkt auf und setzt sich bei der Propagation durch das Medium fort, sodass die oben beschriebene Methodik ein zuverlässiges Maß für den Fehler darstellt. In dieser Untersuchung stehen die nichtlinearen Effekte im Vordergrund. Diese wirken örtlich verteilt in dem definierten nichtlinearen Medium. Dies hat zur Folge, dass auch eine möglicherweise fehlerhafte Beschreibung der nichtlinearen Effekte durch die Algorithmen örtlich verteilt auftritt. Um dies bei der Auswertung in dieser Untersuchung zu erfassen, wird die finale Feldverteilung zum Zeitpunkt $t = T$ der E_x-Komponente über das Simulationsgebiet betrachtet. Analog zu [122] wird die euklidische Norm zur Fehlerberechnung herangezogen. Daher wird Fehler der finalen Feldverteilung der E_x-Komponente mit $\epsilon_{rel} = \sqrt{\sum_{k=1}^{N_z}(E_{x,ref}(z = z_k, t = T) - E_x(z = z_k, t = T))^2}/\sqrt{\sum_{k=1}^{N_z} E_{x,ref}^2(z = z_k, t = T)}$ bestimmt. Bei dem Vergleich der Fehlerkurven fällt zuerst auf, dass alle Faberpolynom basierten

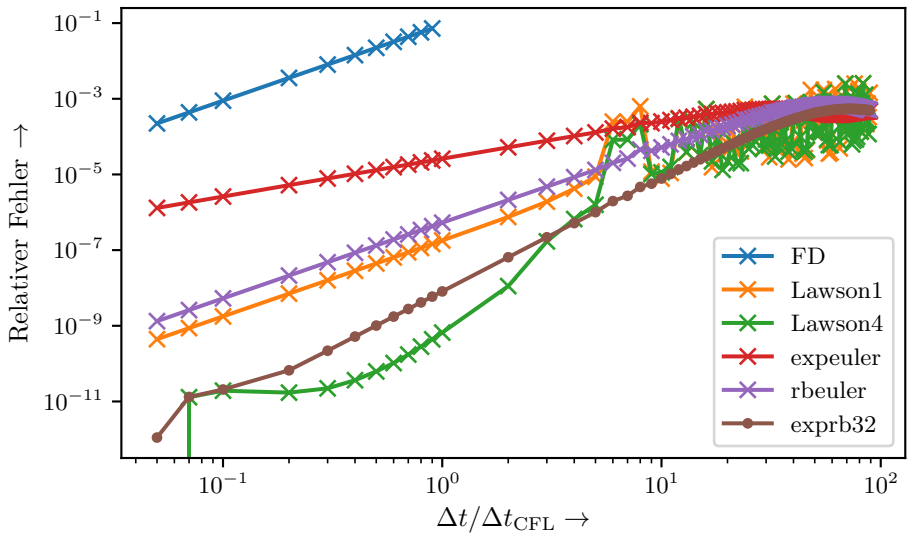


Abbildung 7.1: Die Abbildung zeigt den Verlauf des relativen Fehlers der untersuchten Algorithmen für das nichtlineare System für verschiedene Zeitschrittweiten. Für die Nichtlinearität gilt $\chi^{(2)} = 30\,\mathrm{pV/m}$.

Verfahren deutlich kleinere Fehler aufweisen als der Vergleichsalgorithmus auf Basis des FD-

Ansatzes. Dies ist dadurch zu erklären, dass die Nichtlinearität des Materials in nichtlinearen Termen resultiert, welche nur einen geringen Effekt gegenüber dem linearen Teil des Systems aufweisen. Der lineare Teil wiederum, wird durch die Faberpolynom-Methode, wie in Abschnitt 5.4 gezeigt, sehr genau approximiert. Die Kurve des FD-Algorithmus bricht außerdem bei einer gewissen Zeitschrittweite ab. Dies ist darin begründet, dass ab dieser Zeitschrittweite die Simulation instabil wird. Die Faberpolynom-Verfahren zeigen dieses Verhalten in dem vorliegendem Beispiel nicht. Allerdings ist am Verlauf des Fehlers zu erkennen, dass bei zunehmender Zeitschrittweite der Fehler immer weiter steigt, was durch die zunehmend unzureichende Approximation der Nichtlinearität zu erklären ist. Bei Betrachtung der Faberpolynom-Verfahren fällt auf, dass das Rosenbrock-Euler-Verfahren bei gleicher Zeitschrittweite deutlich genauer als das herkömmliche Exponential-Euler-Verfahren ist. Diese Verfahren werden durch die gleiche Berechnungsvorschrift bestimmt und unterscheiden sich nur in der Realisierung der Beschreibung der Nichtlinearität. Darüber hinaus ist zu erkennen, dass hier die kontinuierliche Beschreibung, welche von den Rosenbrock-Verfahren verwendet wird, bei sonst gleicher Ordnung zu genaueren Ergebnissen führt. Das Lawson-Euler-Verfahren erzielt einen noch geringeren Fehler.

Um auch den Rechenaufwand zu berücksichtigen, wird im Folgenden der Fehler in Abhängigkeit von der nötigen Anzahl der Matrix-Vektor-Multiplikationen mit den verwendeten Jakobi-Matrizen untersucht. Die Ergebnisse sind in Abbildung 7.2 dargestellt. Außerdem fällt auf,

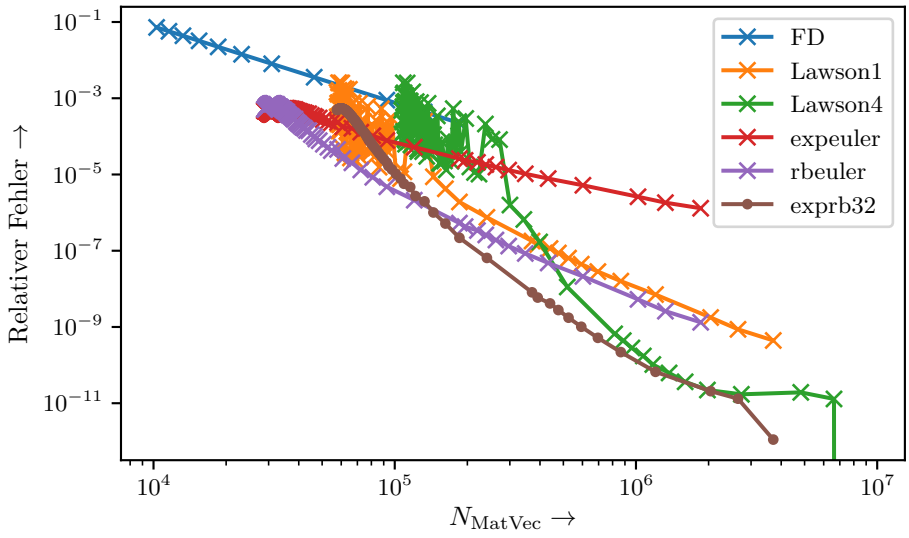


Abbildung 7.2: Die Abbildung zeigt den Fehler der Verfahren in Abhängigkeit von der Gesamtzahl der Matrix-Vektor-Multiplikationen mit der Jakobi-Matrix bei $\chi^{(2)} = 30\,\mathrm{pV/m}$.

dass für niedrige Genauigkeiten der FD-Algorithmus geringe Anzahlen von Matrix-Vektor-Multiplikationen erfordert. Dieser Bereich wird von den Faberpolynom basierten Algorithmen teilweise nicht einmal erreicht. Für höhere Genauigkeiten zeigen sich diese wiederum als deutlich effizienter. Insbesondere das Rosenbrock-Euler-Verfahren ist für Fehler bis zu $1{,}3 \times 10^{-6}$ am effizientesten. Für höhere Genauigkeiten ist das **exprb32**-Verfahren am effizientesten unter den hier untersuchten Verfahren. Das Lawson-Euler-Verfahren zeigt in Abbildung 7.1 zwar genauere

Ergebnisse als das Rosenbrock-Euler-Verfahren, ist aber, wie in Abbildung 7.2 zu erkennen, ineffizienter. Dies ist durch den geringeren Berechnungsaufwand des Rosenbrock-Euler-Verfahren zu begründen. Die hier vorgeschlagene Berechnung mit dem Ansatz (7.17) erlaubt eine Berechnung mit nur einem Matrixexponential pro Zeitschritt. Das Lawson-Euler-Verfahren profitiert nicht von dieser Implementierung und benötigt zwei Matrixexponentiale pro Zeitschritt.

Die Beiträge des nichtlinearen Terms sind in dem untersuchten System gering. Daher soll das System mit hundertmal höheren nichtlinearen Koeffizienten $\chi_2 = 3\,\mathrm{nV/m}$ erneut berechnet werden. Dieser Wert ist unphysikalisch hoch, soll hier aber verwendet werden, um das Verhalten der Algorithmen zu untersuchen. Alle anderen Parameter bleiben unverändert. Die Ergebnisse sind in Abbildung 7.4 und 7.3 dargestellt. Die Referenzlösung wird mit dem **rb32**-Verfahren bei kleiner Zeitschrittweite bestimmt. In der Abbildung 7.3 ist zu erkennen, dass die stärkere

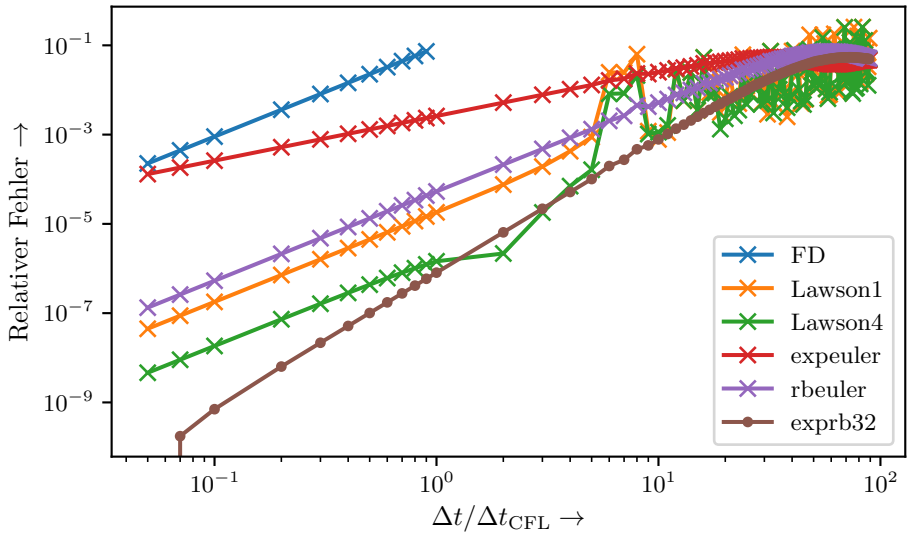


Abbildung 7.3: Die Abbildung zeigt den Verlauf des relativen Fehlers über die Zeitschrittweite für $\chi_2 = 3\,\mathrm{nV/m}$.

Nichtlinearität einen Einfluss auf den Verlauf der Fehlerkurven der untersuchten Algorithmen hat. Insgesamt fällt der Fehler von allen Algorithmen größer aus. Der FD-Referenzalgorithmus ist hiervon nicht betroffen und zeigt einen unveränderten Fehlerverlauf. Allerdings liefert der Algorithmus für alle untersuchten Zeitschrittweiten weiterhin ungenauere Ergebnisse. Bei der Betrachtung von Abbildung 7.4 fällt auf, dass für hohe Genauigkeiten die Faberpolynom-Verfahren weiterhin die effizientere Wahl sind. Außerdem zeigen die Rosenbrock-Verfahren im Vergleich zu den anderen Verfahren eine bessere Effizienz bei der höheren Nichtlinearität.

7.5 Bewertung

In diesem Kapitel werden verschiedene Ansätze untersucht, den Faberpolynom-Algorithmus für die Inklusion von nichtlinearen Effekten zu erweitern. Hierzu werden zunächst verschiedene Strategien zur Formulierung der Nichtlinearität vorgestellt. Im Anschluss wird eine effiziente

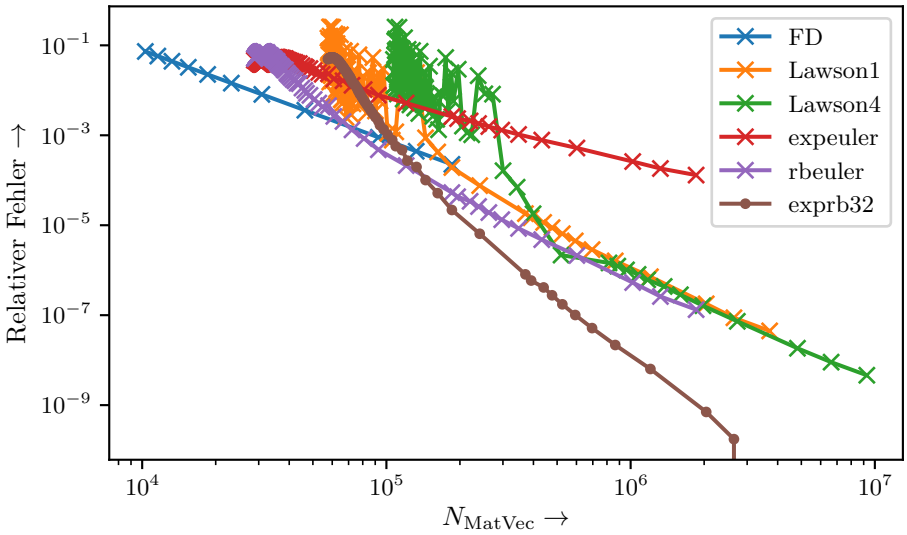


Abbildung 7.4: Die Abbildung zeigt den Fehler der Verfahren in Abhängigkeit von der Anzahl der Matrix-Vektor-Multiplikation für $\chi_2 = 3\,\mathrm{nV/m}$.

Möglichkeit zur Berechnung der auftretenden Termen mit φ-Funktionen vorgeschlagen. Alle untersuchten Algorithmen sind explizit und erlauben eine effiziente parallele Implementierung. Im Anschluss werden die Algorithmen mit einem Beispiel numerisch evaluiert. Hierbei zeigt sich wie bei den linearen Simulationen in 5.4, dass die Faberpolynom basierten Verfahren insbesondere für hohe Genauigkeiten eine um Größenordnungen höhere Effizienz als vergleichbare klassische Verfahren aufweisen. Bei der Formulierung der Nichtlinearität zeigt sich, dass eine Beschreibung über eine dynamisch bestimmte Jakobi-Matrix (7.4) mit einem nichtlinearen Restglied (7.5) zu deutlich genaueren Ergebnissen führt. Hierfür ist die Faberpolynom-Methode insbesondere geeignet, da ihre Berechnung nur Matrix-Vektor-Multiplikationen mit der Jakobi-Matrix erfordert. So ist eine wiederholte, aufwendige Besetzung der Jakobi-Matrix unnötig. Durch die Verwendung von Verfahren höherer Ordnungen kann die Genauigkeit noch erheblich gesteigert werden. Insbesondere, wenn hohe Genauigkeiten gefordert werden, sind diese Verfahren deutlich effizienter als solche niedriger Ordnung. Allerdings erfordern diese die Berechnung von Termen mit gewichteten Summen aus φ-Funktionen. Diese werden effizient mit dem Algorithmus (7.17) bestimmt. So ist nur die Berechnung von einer Matrixfunktion pro Stufe des Integrators nötig. In diesem Zusammenhang ist es fraglich, ob ohne diesen Ansatz eine derart effiziente Berechnung möglich wäre. Im Anschluss wird ein System mit einer deutlich stärkeren Nichtlinearität betrachtet. Hier verschlechtert sich die Genauigkeit der Faberpolynom-Algorithmen. Jedoch sind sie weiterhin genauer als der FD-Vergleichsalgorithmus. Hier erweisen sich insbesondere die Rosenbrock-Verfahren als effizient.

Ein möglicher Ansatzpunkt für eine Erweiterung ist die Verwendung einer variablen Zeitschrittweite Δt. Außerdem ist gerade bei Verfahren höherer Ordnung der Rückgriff auf größere Zeitschrittweiten interessant. In diesem Zusammenhang sollte weiter untersucht werden, für welche Zeitschrittweiten die Verfahren am effizientesten sind. Für das vorliegende Beispiel wird auch bei den dynamisch bestimmten Jakobi-Matrizen bei den Rosenbrock-Verfahren für jeden

Zeitschritt die gleiche Faberpolynom-Approximation verwendet. Dies ist möglich, da hier kein signifikanter Einfluss auf die Grenzen des Eigenwertspektrums der Jakobi-Matrix vorliegt. Für andere Systeme sind allerdings kompliziertere Abhängigkeiten zu erwarten. Hier kann durch eine dynamische Anpassung an die vorliegende Jakobi-Matrix die Approximation weiter optimiert werden.

8 Time-Domain-Beam-Propagation

In den vorangegangenen Abschnitten werden verschiedene Algorithmen zur Zeitbereichssimulation der Maxwell-Gleichungen mithilfe von Polynom-Approximation basierenden Algorithmen untersucht. Hierbei haben sich besonders die Faberpolynome als leistungsfähig und flexibel erwiesen. Was alle bisherigen Algorithmen allerdings gemeinsam haben, ist die Eigenschaft, dass keine Approximationen bezüglich der Systemmatrix $\mathcal{H}$ getroffen werden. Auch bei dem Algorithmus mit der komplexen Einhüllenden wird zwar eine Trägerschwingung von potenziell hoher Frequenz in die Systemmatrix aufgenommen, allerdings wird bezüglich der anderen Frequenzen keine Approximation getroffen. Das bedeutet, dass in diesem Fall immer noch das gesamte Eigenwertspektrum der Systemmatrix $\mathcal{H}$ für die Approximation berücksichtigt werden muss. Der Ansatz ermöglicht eine effiziente Einbindung von Quelltermen, welche um eine Trägerschwingung bandbegrenzt sind. Eine Effizienzsteigerung für das quellfreie System ist allerdings nicht zu erwarten.

In der Regel erfordert die Geometrie von Problemen der Photonik eine deutlich feinere Diskretisierung, als es die Frequenzen der eigentlich vorliegenden Signale erfordern würden. Dies spiegelt sich dann auch im Eigenwertspektrum der Systemmatrix $\mathcal{H}$ wider. Deren Eigenwertspektrum wird durch die feine Diskretisierung verbreitert. Wie in den vorangegangenen Kapiteln schon untersucht, lassen sich die Frequenzen der untersuchten elektromagnetischen Felder den Eigenwerten der Systemmatrix $\mathcal{H}$ zuordnen. Durch die feinere Diskretisierung werden daher sehr hohe Frequenzkomponenten mitberechnet, welche keinen Beitrag zu dem technisch interessanten Frequenzbereich liefern.

In den folgenden Abschnitten soll eine Klasse von Ansätzen untersucht werden, welche eine Anpassung der Systemmatrix ermöglichen. Bei diesen handelt es sich um die Klasse der TDBPM-Algorithmen. Sie werden in der Literatur für die Analyse von Bauelementen der Photonik eingesetzt [128–131]. Diese Klasse von Methoden basiert auf einer Umformulierung der Maxwell-Gleichungen in die vektorielle Wellengleichung. Im Anschluss wird ein komplexer Einhüllenden-Ansatz verwendet. Von dieser Formulierung ausgehend, können weitere Approximationen bezüglich der Frequenzabhängigkeit der Felder getroffen werden [132]. Darüber hinaus erlaubt die Methode eine effiziente Realisierung von Algorithmen zur Berechnung der Propagation von kurzen Impulsen über lange Distanzen. Dies wird durch ein dem Impuls folgendes Fenster realisiert [133–136].

In der Literatur werden diese Ansätze in der Regel mit impliziten Lösungsmethoden behandelt [128–130, 132, 137]. Wie in den vorherigen Kapiteln beschrieben, ermöglichen diese zwar in vielen Fällen die Anwendung höherer Zeitschritte und damit oft auch eine Steigerung der Effizienz der Algorithmen, allerdings treten Probleme bei der Parallelisierung auf, wenn lineare Gleichungssysteme mit sehr vielen Unbekannten gelöst werden müssen. Das Problem tritt besonders dann in den Vordergrund, wenn große dreidimensionale Probleme betrachtet werden. In diesen Fällen werden die Berechnungen auf mehrere Rechnerknoten verteilt. Implizite Algorithmen erfordern

dann zusätzlichen Kommunikationsaufwand, welcher die Parallelisierung erschwert.

Daher soll in dem folgenden Abschnitt ein neuer TDBPM-Algorithmus auf der Basis von Faberpolynomen entwickelt werden. Hierbei soll die Faberpolynom-Entwicklung genutzt werden, um einen expliziten Algorithmus zu formulieren. Neben dem Rechenaufwand soll besonders die Genauigkeit der Approximation der Zeitpropagation untersucht werden. Diese Betrachtungen ermöglichen im Anschluss eine Aussage über die Effizienz der untersuchten Verfahren. Insbesondere die Effizienz im Vergleich zu der FDTD-Methode ist interessant. Die Untersuchungen in [132] deuten darauf hin, dass viele der in der Literatur beschriebenen TDBPM-Algorithmen ineffizienter sind als die FDTD-Methode.

Im Folgendem soll kurz die Theorie dieser Ansätze beleuchtet werden. Die Ansätze lassen sich in drei Klassen einteilen: Narrow-Band (NB), Wide-Band (WB) und Full-Band (FB). Diese werden beschrieben und hinsichtlich ihrer Eignung charakterisiert. Teile der hier vorstellten Untersuchungen sind in [KS12] und [KS13] veröffentlicht. Die Ergebnisse werden im Folgenden mit Ergänzungen dargestellt.

8.1 Formulierung der Methode

Hier soll zunächst auf die Formulierung der TDBPM-Methoden eingegangen werden. In der Literatur werden viele dieser Methoden auf Basis der skalaren Wellengleichung entwickelt [130, 132]. In dieser Arbeit soll ein vollvektorieller Ansatz entwickelt werden.

Zuerst wird, ausgehend von den Maxwell-Gleichungen, das magnetische Feld $\vec{H}(\vec{r},t)$ durch Umformen eliminiert. Hierdurch ergibt sich die vektorielle Wellengleichung [138]:

$$-\epsilon(\vec{r})\frac{\partial^2}{\partial t^2}\vec{E}(\vec{r},t) = \nabla \times \left(\frac{1}{\mu(\vec{r})}\nabla \times \vec{E}(\vec{r},t)\right). \tag{8.1}$$

Die Betrachtung erfolgt zunächst ohne Stromterme und für lineare Materialien ohne dispersive Eigenschaften oder Dämpfung. Nun wird ein komplexer Einhüllenden-Ansatz auf alle Feldgrößen angewendet:

$$\vec{E}(\vec{r},t) = \mathcal{R}\{\vec{\hat{E}}(\vec{r},t)e^{j\omega_0 t}\}. \tag{8.2}$$

Dieser wird in (8.1) eingesetzt. Damit lässt sich die Darstellung ableiten, welche die Grundlage der hier entwickelten TDBPM-Methoden bildet:

$$-\frac{\partial^2}{\partial t^2}\vec{\hat{E}}(\vec{r},t) - j2\omega_0\frac{\partial}{\partial t}\vec{\hat{E}}(\vec{r},t) + \omega_0^2\vec{\hat{E}}(\vec{r},t) = \frac{1}{\epsilon(\vec{r})}\nabla \times \left(\frac{1}{\mu(\vec{r})}\nabla \times \vec{\hat{E}}(\vec{r},t)\right). \tag{8.3}$$

Ausgehend von dieser Darstellung werden die NB- und WB-Ansätze entwickelt, welche in den folgenden Abschnitten untersucht werden. Auf die in der Literatur beschriebenen FB-Methoden [132] soll hier nicht weiter eingegangen werden, da diese direkt auf (8.3) basieren und keine weitere Approximation ansetzen. Damit stellen diese nur eine andere Form der bereits untersuchten komplexe Einhüllenden-Methode dar.

8.1.1 Einbindung von Stromtermen

Um in den oben beschriebenen Ansatz Stromterme aufzunehmen, muss dieser erweitert werden. Hierzu werden allgemein die elektrische Stromdichte $\vec{J}(\vec{r},t)$ und formal eine magnetische Stromdichte $\vec{K}(\vec{r},t)$ in die Maxwell-Gleichungen eingesetzt. Unter diesen Voraussetzungen ergibt sich die folgende Gleichung:

$$-\epsilon(\vec{r})\frac{\partial^2}{\partial t^2}\vec{E}(\vec{r},t) = \nabla\times(\frac{1}{\mu(\vec{r})}\nabla\times\vec{E}(\vec{r},t)) + \frac{\partial}{\partial t}\vec{J}(\vec{r},t) + \nabla\times\frac{1}{\mu(\vec{r})}\vec{K}(\vec{r},t). \tag{8.4}$$

Nun wird wie bei dem quellfreiem System der komplexe Einhüllenden-Ansatz (8.2) angewendet. Damit lässt sich

$$\begin{aligned} -\frac{\partial^2}{\partial t^2}\vec{\hat{E}}(\vec{r},t) - j2\omega_0\frac{\partial}{\partial t}\vec{\hat{E}}(\vec{r},t) + \omega_0^2\vec{\hat{E}}(\vec{r},t) = \\ \frac{1}{\epsilon(\vec{r})}\nabla\times\left(\frac{1}{\mu(\vec{r})}\nabla\times\vec{\hat{E}}(\vec{r},t)\right) + \frac{j\omega_0}{\epsilon(\vec{r})}\vec{\hat{J}}(\vec{r},t) + \frac{1}{\epsilon(\vec{r})}\frac{\partial}{\partial t}\vec{\hat{J}}(\vec{r},t) \\ +\frac{1}{\epsilon(\vec{r})}\nabla\times\frac{1}{\mu(\vec{r})}\vec{\hat{K}}(\vec{r},t) \end{aligned} \tag{8.5}$$

bestimmen. Zu beachten ist, dass der Einhüllenden-Ansatz (8.2) auf alle Feldgrößen angewendet werden muss, was die Ströme $\vec{J}(\vec{r},t)$ und $\vec{K}(\vec{r},t)$ mit einschließt.

8.1.2 Systemmatrix der vektoriellen Wellengleichung

In den vorangegangenen Abschnitten werden die Maxwell-Gleichungen in ihrer ursprünglichen Form als System erster Ordnung verwendet. Auf ihre Diskretisierung wird in Abschnitt 3.3 eingegangen. In 5.3.2 werden hierzu die Eigenschaften des Eigenwertspektrums der Systemmatrix beleuchtet. Diese hängen sowohl mit der Struktur der Gleichungen, der örtlichen Diskretisierung als auch der Materialparameter innerhalb des Simulationsgebietes zusammen. Im Folgendem sollen wieder Polynom basierte Entwicklungsmethoden für die Zeitpropagationsoperatoren verwendet werden. Im Speziellen sollen die Faberpolynome genutzt werden. Um eine effiziente Approximation der Operatoren mit diesen zu erreichen, sind Informationen über das Eigenwertspektrum der betrachteten Systemmatrizen nötig.

Zunächst (8.4) wird örtlich diskretisiert. Hierzu soll zunächst eine FD-Diskretisierung nach Yee verwendet werden. Das örtlich diskretisierte System ist mit

$$-\frac{\partial^2}{\partial t^2}\vec{\phi}(t) = \mathcal{M}\vec{\phi}(t) + \vec{\vartheta}(t) \tag{8.6}$$

gegeben. Das örtlich diskretisierte System von (8.5) kann analog mit

$$-\frac{\partial^2}{\partial t^2}\vec{\hat{\phi}}(t) - j2\omega_0\frac{\partial}{\partial t}\vec{\hat{\phi}}(t) + \omega_0^2\vec{\hat{\phi}}(t) = \mathcal{M}\vec{\hat{\phi}}(t) + \vec{\hat{\vartheta}}(t) \tag{8.7}$$

bestimmt werden. Dabei enthalten die Vektoren $\vec{\phi}$ beziehungsweise $\vec{\hat{\phi}}$ die diskretisierte Feldverteilung. In $\vec{\vartheta}(t)$ und $\vec{\hat{\vartheta}}(t)$ werden die diskretisierten Äquivalente der Stromdichten in (8.6) und (8.5) zusammengefasst.

Tabelle 8.1: Die Anzahl der nötigen Vektor-Vektor-Multiplikationen der Größen N_z, N_xN_y beziehungsweise $N_xN_yN_z$ für den ein-, zwei und dreidimensionalen Fall ist für die Systemmatrix der vektoriellen Wellengleichung und für die Systemmatrix der Maxwell-Gleichungen gegeben.

	1D	2D	3D
Maxwell: $\mathcal{H}$	4	8	24
Vektorielle Wellengleichung $\mathcal{M}$	3	5	39

Anders als bei den zuvor betrachten Systemmatrizen ist das Eigenwertspektrum nun nicht symmetrisch um die reelle Achse verteilt. Das Eigenwerte σ_k der Systemmatrix $\mathcal{M}$ für ein dämpfungsfreien System liegen auf der reellen Achse auf dem Intervall $\sigma_k \in [0, \sigma_{\max}]$. Die obere Grenze $\sigma_{\max}$ kann hierbei beispielsweise mit dem Gerschgorin-Theorem bestimmt werden [54, 113]. Bei ϵ und μ sind hierbei jeweils die niedrigsten Werte im Rechengebiet zu verwenden. Für den eindimensionalen Fall ergibt sich

$$\sigma_{\max,\,1\mathrm{D}} = \frac{4}{\epsilon\mu\Delta^2}, \tag{8.8}$$

für die zweidimensionale TM-Mode ergibt sich

$$\sigma_{\max,\,2\mathrm{DTM}} = \frac{8}{\epsilon\mu\Delta^2} \tag{8.9}$$

und für den dreidimensionalen Fall kann $\sigma_{\max,\,3\mathrm{D}}$ mit

$$\sigma_{\max,\,3\mathrm{D}} = \frac{16}{\epsilon\mu\Delta^2} \tag{8.10}$$

bestimmt werden. Die Komplexität der Matrix-Vektor-Multiplikation unterscheidet sich aufgrund der anderen Struktur der Systemmatrix $\mathcal{M}$ im Vergleich zu der Systemmatrix $\mathcal{H}$. Um die Ergebnisse in den folgenden Abschnitten hinsichtlich der Effizienz der Algorithmen besser einordnen zu können, wird in Tabelle 8.1 der jeweilige Rechenaufwand für eine FD-Yee-Diskretisierung gemäß [138] angegeben.

8.2 Schmalband Approximation

Zuerst soll ein Algorithmus der sogenannten NB-Verfahren entwickelt werden. Diese Klasse von TDBPM-Verfahren wird ausgehend von (8.7) entwickelt. Die Annahme ist hierbei, dass die zweite Ableitung in (8.7) vernachlässigt werden kann [132]. Damit ist der Algorithmus nur für Systeme mit sehr bandbegrenzten Feldern geeignet. Die Besonderheit ist, dass der neu vorgestellte Algorithmus mit einer Operatorentwicklung auf Basis der bereits untersuchten Faberpolynome realisiert wird. Hierdurch soll eine explizite Berechnungsvorschrift ermöglicht werden. Die Ergebnisse sind in Teilen in [KS9] vorgestellt und werden im Folgenden ergänzt ausgeführt.

8.2.1 Formulierung und Untersuchung

Mit der oben genannten Vernachlässigung der zweiten Ableitung in (8.7) kann

$$\frac{\partial}{\partial t}\vec{\tilde{\phi}}(t) = j\frac{\mathcal{M} - \omega_0^2 I_N}{2\omega_0}\vec{\tilde{\phi}}(t) - \vec{\tilde{\vartheta}}(t) \tag{8.11}$$

bestimmt werden. Bei I_N handelt es sich um eine Einheitsmatrix der Größe N, wobei N die gesamte Anzahl der diskretisierten Feldgrößen ist. Um die Formulierung kompakter zu gestalten, wird ein weiterer Matrixoperator eingeführt:

$$\hat{\mathcal{M}} = j\frac{\mathcal{M} - \omega_0^2 I_N}{2\omega_0}. \tag{8.12}$$

Die Stromterme sind in der Variablen $\vec{\hat{\vartheta}}(t)$ zusammengefasst. Da der Verlauf der Ströme in der Regel bekannt ist, müssen ihre Zeitableitungen nicht wie die elektrischen und magnetischen Felder in der Zeitpropagation berechnet, sondern können im Vorfeld bestimmt werden. Damit kann die formale Lösung mit

$$\vec{\hat{\phi}}(t_n + \Delta t) = \exp(\Delta t\hat{\mathcal{M}})\vec{\hat{\phi}}(t_n) + \int_0^{\Delta t} \exp((\Delta t - \tau)\hat{\mathcal{M}})(-1)\vec{\hat{\vartheta}}(t_n + \tau)d\tau \tag{8.13}$$

angegeben werden. Die Struktur der formalen Lösung (8.13) entspricht hierbei der formalen Lösung (6.3) aus Kapitel 6. Sie unterscheiden sich aber in der Systemmatrix. Hier wird mit der Systemmatrix $\hat{\mathcal{M}}$ eine Matrix verwendet, welche den Einhüllenden-Ansatz sowie die oben beschriebene NB-Näherung enthält. Diese liegt in Kapitel 6 nicht vor. Dennoch erlaubt die gleiche Struktur von (8.13) die Verwendung der gleichen Methoden zur Lösung. Zunächst wird ein quellfreies System mit (8.13) betrachtet. Dieses wird durch das Matrixexponential beschrieben. Die Matrixfunktion kann nun mit den zuvor untersuchten Methoden entwickelt werden. Hierbei stellt sich die Frage, ob es möglich ist, einen effizienten Algorithmus mit diesem Ansatz zu konstruieren. Der Operator entspricht dem der vorherigen Kapitel. Die Ansätze unterscheiden sich nur durch die Systemmatrix $\hat{\mathcal{M}}$ in (8.12) im Vergleich zu $\mathcal{H}$ in (6.1). Daher sollen sich die folgenden Untersuchungen auf die Systemmatrix und ihre Eigenschaften beschränken. Diese werden mit den Eigenschaften der Systemmatrix $\mathcal{H}$ ohne komplexe Einhüllende verglichen.

Die Untersuchung soll exemplarisch für die TM-Mode eines zweidimensionalen Systems durchgeführt werden. Die Zusammenhänge für die anderen Fälle lassen sich analog ableiten. Darüber hinaus wird eine Ortsdiskretisierung mit dem Yee-Gitter angesetzt. Andere Ansätze, wie die in Abschnitt 3.1.2 beschriebenen pseudospektralen Methoden oder die diskontinuierlichen Galerkin-Methoden, sind auch möglich.

Bei einer gleichförmigen Diskretisierung mit $\Delta = \Delta x = \Delta y$ in beide Ortsrichtungen x und y ergibt sich das Eigenwertspektrum der Matrix $\mathcal{M}$ der vektoriellen Wellengleichung zu (8.9). Das Eigenwertspektrum $\sigma(\mathcal{M})$ in der komplexen Ebene ist in Abbildung 8.1a dargestellt.

Durch die Anwendung des NB-Ansatzes ergibt sich die Systemmatrix $\hat{\mathcal{M}}$ in (8.12). Dies hat zwei wichtige Auswirkungen auf das Eigenwertspektrum der Systemmatrix $\hat{\mathcal{M}}$: Zunächst wird $\mathcal{M}$ um den Faktor $2\omega_0$ gestaucht. Im Anschluss wird diese noch um den Wert $\omega_0/2$ verschoben und mit der imaginären Einheit j multipliziert. Die Eigenwerte liegen nun in dem Bereich

$$\sigma \in [-j\omega_0/2, j(\sigma_{\max}/(2\omega_0) - \omega_0/2)]. \tag{8.14}$$

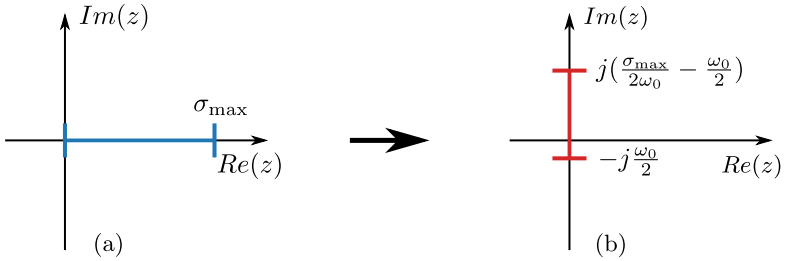


Abbildung 8.1: In 8.1a ist die Verteilung der Eigenwerte der Systemmatrix der vektoriellen Wellengleichung $\mathcal{M}$ dargestellt. In 8.1b ist das Spektrum für die Systemmatrix $\hat{\mathcal{M}}$ der NB-TDBPM-Methode dargestellt.

In Abbildung 8.1b ist das Spektrum von $\sigma(\hat{\mathcal{M}})$ schematisch dargestellt. Nun soll der Operator in (8.13) mithilfe von Faberpolynomen entwickelt werden. Damit ist es möglich, eine Approximation für das Matrixexponential in (8.13) zu bestimmen. Die resultierende Approximation kann unter Verwendung des elliptischen Konvergenzbereiches mit der Rekursionsbeziehung (5.12) bestimmt werden. So ist eine explizite Berechnung mit großen Zeitschrittweiten Δt möglich.

Die Verschiebung und die Multiplikation mit dem Faktor j müssen hierbei nur in der Faberpolynom-Entwicklung berücksichtigt werden. Anders verhält es sich mit der Skalierung um den Faktor $2\omega_0$. Die nötige Polynomordnung bei der Entwicklung mit Faberpolynomen, um eine bestimmte Genauigkeit zu erreichen, ist, wie oben erläutert, im betrachteten Fall von der Breite des Eigenwertspektrums abhängig [84, 90, 91]. Dies ist der Fall, da in dem dämpfungsfreien System alle Eigenwerte auf der imaginären beziehungsweise reellen Achse liegen. Die Breite D des Spektrums sei hier mit dem maximalen Abstand $D = |\sigma_k - \sigma_i|_{\max}$ von zwei Eigenwerten σ_k, σ_i definiert. Die Skalierung um den Faktor $2\omega_0$ verkleinert die Breite $D_{\text{NB-TDBPM}}$ des Eigenwertspektrums $\sigma(\hat{\mathcal{M}})$, welche mit

$$D_{\text{NB-TDBPM}} = \sigma_{\max}/(2\omega_0) \tag{8.15}$$

gegeben ist, mit steigender Frequenz ω_0 zunehmend. Dies hat zur Folge, dass niedrigere Entwicklungsordnungen für die Faberpolynom-Entwicklung möglich sind. Nun stellt sich die Frage, ob dieser Effekt genutzt werden kann, um effizientere Zeitpropagations-Algorithmen zu konstruieren.

Wird die Breite $D_{\text{FB-TDBPM}} = \sigma_{\max}$ mit (8.9) des Eigenwertspektrums $\sigma(\mathcal{M})$ des ursprünglichen Operators $\mathcal{M}$ untersucht, so fällt auf, dass dessen Größe mit feiner werdender Diskretisierung immer weiter zunehmen wird. Daher liegen zwei gegenläufige Effekte vor. In diesem Zusammenhang muss also untersucht werden, ob es einen technisch nutzbaren Optimalwert gibt.

Zur Klärung soll das Eigenwertspektrum $\sigma(\hat{\mathcal{M}})$ der NB-TDBPM-Methode mit dem Eigenwertspektrum $\sigma(\mathcal{H})$ des ursprünglichen Systems ohne Einhüllende in (3.19) unter Verwendung der gleichen Diskretisierung verglichen werden. Die gesamte Breite des Eigenwertspektrums $\sigma(\mathcal{H})$ der Systemmatrix ist unter Verwendung von (5.38) mit

$$D_{\text{ref}} = 2\sqrt{2}/(\Delta\sqrt{\epsilon\mu}) - (-4\sqrt{2}/(\Delta\sqrt{\epsilon\mu})) = 4\sqrt{2}/(\Delta\sqrt{\epsilon\mu}) \tag{8.16}$$

zu bestimmen. Nun soll das Verhältnis $R = D_{\text{ref}}/\sigma_{\text{NB-TDBPM}}$ in Abhängigkeit von der örtlichen Diskretisierungsdichte λ_0/Δ untersucht werden. Die Verwendung von λ_0/Δ ermöglicht eine

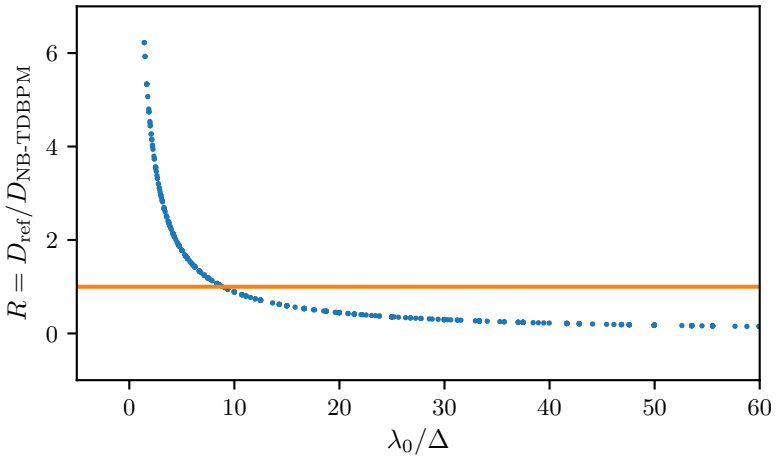


Abbildung 8.2: Das Verhältnis R der Breite D_{ref} des Eigenwertspektrums des Vergleichsansatzes aus Kapitel 5 zu der Breite $D_{\text{NB-TDBPM}}$ des NB-TDBPM-Algorithmus ist in Abhängigkeit von λ_0/Δ mit den blauen Punkten dargestellt. Die orangefarbende Linie markiert das Verhältnis $R = 1$. Oberhalb dieses Wertes sind Gewinne im Hinblick auf die Effizienz des NB-TDBPM-Algorithmus zu erwarten.

Untersuchung unabhängig von dem konkreten Wert der Frequenz f_0. Wenn das Verhältnis R größer als eins ist, so ist das Spektrum des untersuchten NB-TDBPM-Ansatzes kleiner im Vergleich zu dem Referenzansatz. Folglich sind in diesem Fall Verbesserungen im Hinblick auf die Effizienz des Verfahrens zu erwarten, da eine niedrigere Polynomordnung nötig ist. Die Ergebnisse der Untersuchung sind in Abbildung 8.2 grafisch dargestellt. In der Abbildung ist zu erkennen, dass für feine Auflösungen keine Gewinne im Hinblick auf die Rechenzeit zu erwarten sind. Mit sinkender Auflösung λ_0/Δ steigt das Verhältnis und überschreitet den Wert $R = 1$ bei $\lambda_0/\Delta \leq 9$. Unterhalb von diesem Wert ist es möglich, durch die geringere Polynomordnungen der Faberpolynome den Rechenaufwand für die Approximation zu reduzieren.

8.2.2 Diskussion

In dem vorangegangenem Abschnitt wird ein NB-TDBPM-Algorithmus auf Basis einer Polynomapproaximation mit Faberpolynomen untersucht. Hierbei wird ausgenutzt, dass das Eigenwertspektrum des Operators (8.12) mit der verwendeten Trägerfrequenz ω_0 gestaucht ist. Wie oben beschrieben, hängt der Approximationsaufwand der Faberpolynome maßgeblich von der Größe des Eigenwertspektrums ab. Diesem wirkt das im Vergleich größere Eigenwertspektrum des Operators der Wellengleichung entgegen. Es wird untersucht, ob sich auf diese Weise eine Effizienzsteigerung erreichen lässt. Die Untersuchung ist für den zweidimensionalen Fall mit der TM-Mode und einem FD-Yee-Gitter als örtliche Diskretisierung durchgeführt worden.

Die Ergebnisse der Untersuchung, welche in Abbildung 8.2 dargestellt sind, zeigen, dass eine Effizienzsteigerung für grobe Auflösungen zu erwarten ist. Für feinere Auflösungen benötigt die ursprüngliche Faberpolynom-Methode ohne NB-TDBPM-Approximation weniger Terme für die Entwicklung und ist daher für diesen Bereich vorzuziehen. Die Grenze hierfür liegt bei

$\lambda_0/\Delta \leq 9$. Die in dieser Untersuchung verwendete FD-Diskretisierung auf Basis des Yee-Gitters, zeigt bei dieser Auflösung bereits signifikante Fehler durch numerische Dispersion [4]. Daher ist eine Verwendung des NB-TDBPM-Algorithmus mit Faberpolynomen hier nicht sinnvoll im Hinblick auf die Effizienz des Algorithmus.

Die Verwendung von pseudospektralen Methoden für die Ortsdiskretisierung ist ein möglicher Ansatz, um diesem Problem entgegenzuwirken. Allerdings ist die Abweichung des NB-TDBPM-Operators in (8.13) von der exakten Lösung der vektoriellen Wellengleichung nur in der Trägerfrequenz ω_0 null. Für alle anderen Bereiche führt die Verwendung des Operators zu zusätzlichen Fehlern [132]. Der Versuch, die Genauigkeit der Approximation zu erhöhen, führt auf die bereits erwähnten WB-TDBPM-Methoden. Diese Methoden weisen einen geringeren Fehler in der Nähe der Trägerfrequenz auf. Außerdem liegen mehr Freiheitsgrade bei der Wahl der Approximation vor. Daher soll diese Klasse von Algorithmen im nächsten Abschnitt daraufhin untersucht werden, ob sie sich zur Verwendung mit Faberpolynom-Approximationen eignen.

8.3 Breitband-Approximation

In diesem Abschnitt soll eine WB-TDBPM-Methode untersucht werden. Zuerst wird wieder zunächst der Propagationsoperator der Methode bestimmt. Im Gegensatz zu früheren Ansätzen soll das Zeitpropagationsschema hier mithilfe einer Operatorapproximation mit Faberpolynomen entwickelt werden. Dieser Ansatz profitiert zum einen von der hohen Genauigkeit der Faberpolynom-Entwicklung und zum anderen von der expliziten Natur der daraus resultierenden Approximation. Allerdings stellt sich aufgrund der Ergebnisse des letzten Abschnitts zu dem Faberpolynom basierten NB-TDBPM-Algorithmus die Frage, ob es auch bei dem WB-TDBPM-Ansatz eine einschränkende Abhängigkeit von der Auflösung gibt und wie diese beschaffen sein könnte.

8.3.1 Theorie

Hierzu sollen zunächst einige theoretische Vorüberlegungen getroffen werden. Der Ausgangspunkt ist, wie bei der NB-Variante, die vektorielle Wellengleichung (8.3) mit einem Einhüllenden-Ansatz. Die quellfreien Maxwell-Gleichungen sollen betrachtet werden. Diese werden zunächst örtlich diskretisiert. Die Ortsdiskretisierung erfolgt mit einem FD-Yee-Gitter [5]. Um den WB-Algorithmus zu bestimmen, wird (8.7) gemäß [130] umformuliert:

$$(-\mathcal{M} + \omega_0^2 I_N)\vec{\tilde{\phi}}(t) = \frac{\partial^2}{\partial t^2}\vec{\tilde{\phi}}(t) + j2\omega_0 \frac{\partial}{\partial t}\vec{\tilde{\phi}}(t). \tag{8.17}$$

Von (8.17) ausgehend wird der Ausdruck weiter umgeformt. Hierzu wird

$$\hat{\mathcal{X}} = -\mathcal{M}/\omega_0^2 + I_N \tag{8.18}$$

definiert. Der Akzent bei $\hat{\mathcal{X}}$ soll verdeutlichen, dass hier wieder die Trägerschwingung ω_0 in die Systemmatrix aufgenommen ist. Damit kann (8.17) zu

$$\frac{\partial}{\partial t}\vec{\tilde{\phi}}(t) = -j\omega_0 \left(I_N - \sqrt{I_N - \hat{\mathcal{X}}}\right)\vec{\tilde{\phi}}(t) \tag{8.19}$$

umformuliert werden [130]. Für eine ausführlichere Herleitung von (8.19) sei auf Anhang B.4 verwiesen. Diese Gleichung ist der Startpunkt für die betrachteten WB-TDBPM-Ansätze. Es wird eine Padé-Approximation der Ordnung (1,1) auf (8.19) angewendet [130]:

$$\frac{\partial}{\partial t}\vec{\hat{\phi}}(t) \approx -j\omega_0 \frac{\frac{\hat{\mathcal{X}}}{2}}{I_N - \frac{\hat{\mathcal{X}}}{4}} \vec{\hat{\phi}}(t). \tag{8.20}$$

Während klassische TDBPM-Algorithmen (8.20) mit einem impliziten Schema lösen [130, 132], wird hier ein alternativer Ansatz auf Basis einer Operatorentwicklung verfolgt, um einen expliziten Algorithmus zu realisieren. Hierzu wird die formale Lösung von (8.20) bestimmt:

$$\vec{\hat{\phi}}(t_n + \Delta t)) = \exp\left(-j\Delta t\omega_0 \frac{\frac{\hat{\mathcal{X}}}{2}}{I_N - \frac{\hat{\mathcal{X}}}{4}}\right) \vec{\hat{\phi}}(t_n)). \tag{8.21}$$

Diese Matrixfunktion soll nun mit Faberpolynomen in Abhängigkeit von $\hat{\mathcal{X}}$ approximiert werden. Da die Faberpolynome mit der Rekursionsgleichung (5.7) berechnet werden können, ermöglicht dieser Ansatz die Realisierung eines expliziten Algorithmus für die Propagation. Im Folgendem werden die hierfür benötigten Vorbetrachtungen durchgeführt. Zunächst müssen hierzu Informationen über das Eigenwertspektrum der Matrix $\hat{\mathcal{X}}$ vorliegen. Der Startpunkt ist wieder das Eigenwertspektrum der Systemmatrix $\mathcal{M}$ der vektoriellen Wellengleichung, welches in Abschnitt 8.1.2 untersucht wird. Mit (8.18) lässt sich das Eigenwertspektrum $\sigma(\hat{\mathcal{X}})$ wie mit

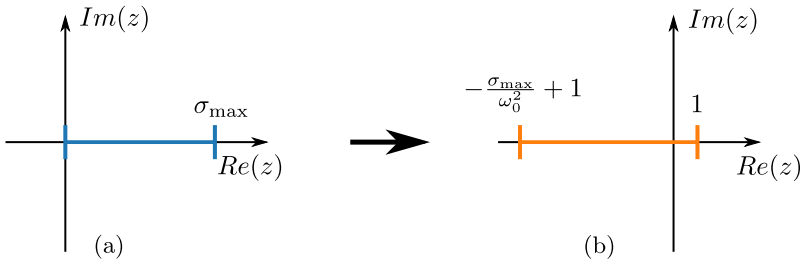


Abbildung 8.3: In Abbildung 8.3a ist die Verteilung der Eigenwerte der Systemmatrix $\mathcal{M}$ der vektoriellen Wellengleichung dargestellt. In Abbildung 8.3b ist das Eigenwertspektrum der Systemmatrix $\hat{\mathcal{X}}$ für die WB-TDBPM-Methode dargestellt.

$$\sigma_k \in \left[-\frac{\sigma_{\max}}{\omega_0^2} + 1, 1\right] \tag{8.22}$$

angegeben. Dieses ist in Abbildung 8.3 skizziert. Zum einen ist $\sigma(\mathcal{M})$ im Vergleich zu $\sigma(\hat{\mathcal{X}})$ entlang der reellen Achse verschoben. Zum anderen ist das Spektrum mit dem Faktor ω_0^2 skaliert. Dies wird auch bei dem in Abschnitt 8.2 untersuchten NB-TDBPM-Algorithmus beobachtet. Auch hier liegt dadurch das Potenzial für eine Reduzierung der Approximationsordnung vor, da das Eigenwertspektrum in seiner Breite verringert wird.

Im Vergleich zu dem in Abschnitt 8.2 untersuchten NB-TDBPM-Algorithmus liegt mit (8.21) darüber hinaus auch kein Matrixexponential mehr vor. Daher soll (8.21) im Folgenden in den Blick genommen werden. Zunächst wird der Einfluss der Padé-Approximation untersucht. Hierzu

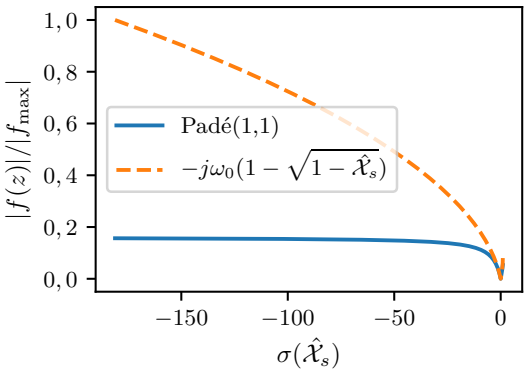


Abbildung 8.4: Die Abbildung zeigt die Absolutwerte der Operatoren (8.19) und (8.20) in Abhängigkeit des Eigenwertspektrums der Systemmatrix $\hat{\mathcal{X}}$. Hierbei ist (8.19) mit der orangefarbenen gestrichelten Linie, während die Padé-Approximation (8.20) mit der blauen Linie dargestellt wird. Die Kurven sind mit dem maximalen Funktionswert $|f_{\max}| = \sqrt{\sigma_{\max}} - \omega_0$ normiert.

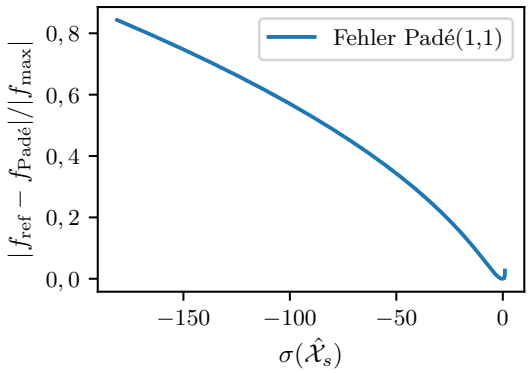


Abbildung 8.5: Der Absolutwert der Abweichung von (8.20) wird im Vergleich zu (8.19) in Abhängigkeit von den Eigenwerten von $\hat{\mathcal{X}}$ dargestellt und mit dem maximalen Absolutwert $|f_{\max}|$ der Funktion normiert.

wird (8.19) mit der Padé-Approximation (8.20) in Abbildung 8.4 verglichen. Darin ist direkt zu erkennen, dass die Anwendung der Padé-Approximation zu einer Abweichung von dem ursprünglichen Operator (8.19) führt. Weiterhin fällt auf, dass die Abweichung besonders an den Rändern des Eigenwertspektrums ansteigt. In null liegt kein Fehler vor, während die Abweichung in der näheren Umgebung von null gering ist. Diese Eigenwerte entsprechen der verwendeten Trägerfrequenz ω_0. Daher wird, obwohl die Padé-Approximation zusätzliche Fehler zur Folge hat, nur ein geringer Fehler in der Umgebung der Trägerfrequenz erzeugt. Für die Trägerfrequenz selbst liegt darüber hinaus kein Fehler vor.

Die Verwendung der Padé-Approximation hat einen weiteren Einfluss, welcher in 8.4 zu erkennen ist. Sie hat zur Folge, dass die Werte in dem Matrixexponential in (8.21) effektiv begrenzt werden. Dies hat zusätzlich das Potenzial, die Approximationsordnung für die Faberpolynom-Entwicklung von (8.21) zu senken. Daher soll diese Eigenschaft hier illustriert werden.

Zu diesem Zweck wird der Operator in dem Matrixexponential in (8.21) betrachtet, welcher hier als

$$\mathcal{F} = -j\omega_0 \frac{\hat{\mathcal{X}}/2}{I_N - \hat{\mathcal{X}}/4} \tag{8.23}$$

definiert wird. Der Operator $\mathcal{F}$ wird verwendet, um die Methode mit der klassischen Formulierung auf Basis des Systems erster Ordnung in (3.19) zu vergleichen. Hierbei soll das Eigenwertspektrum der Operatoren verglichen werden. Für beide Fälle wird ein FD-Yee-Gitter mit $\Delta = \Delta x = \Delta y$ zur örtlichen Diskretisierung verwendet. Der zweidimensionale Fall mit der TM-Mode wird betrachtet. In Abbildung 8.6 wird die komplette Breite der Eigenwertspektren in Abhängigkeit von der Schrittweite Δ der örtlichen Diskretisierung verglichen. Für den TDBPM-Ansatz wird $f_0 = 200\,\mathrm{THz}$ verwendet. Die Breite des Eigenwertspektrums D_{ref} von der Systemmatrix $\mathcal{H}$ ist in (8.16) gegeben. Die Breite $D_{\text{WB-TDBPM}}$ des Eigenwertspektrums von $\mathcal{F}$ ist mit (8.23) und (8.22) zu bestimmen. Hierbei wird $\sigma_{\max}$ abhängig von der Dimensionalität des Problems mit den Berechnungsvorschriften aus Abschnitt 8.1.2 bestimmt. In der Abbildung 8.6 kann

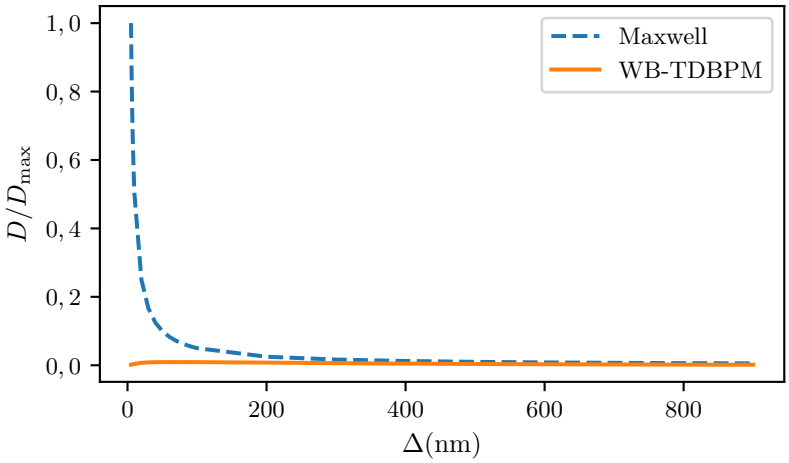


Abbildung 8.6: Die Abbildung zeigt die Breite D_{ref} des Eigenwertspektrums des Vergleichssystems in (3.19) und die Breite $D_{\text{WB-TDBPM}}$ des Operators $\mathcal{F}$ in dem Exponential des WB-TDBPM-Ansatzes für verschiedene örtliche Schrittweiten $\Delta = \Delta_x = \Delta_y$.

beobachtet werden, dass das Spektrum des klassischen Ansatzes, wie erwartet, für kleine Schrittweiten Δ große Werte annimmt. Ausgehend von den Berechnungsvorschriften für die vektorielle Wellengleichung in 8.1.2, könnte ein ähnlicher Verlauf für den $\mathcal{F}$ erwartet werden. Bei Betrachtung von 8.6 fällt allerdings auf, dass hier die Breite des Spektrums einen anderen Verlauf nimmt. Die Padé-Approximation limitiert die Werte von $\mathcal{F}$, sodass der Wert von $\mathcal{F}$ für feine Auflösungen $\Delta \to 0$ nicht gegen Unendlich konvergiert.

Zu beachten ist, dass das Eigenwertspektrum des TDBPM-Ansatzes nicht nur von der örtlichen Diskretisierung, sondern auch von der Wahl der Trägerfrequenz ω_0 abhängt, während der Referenzansatz von dieser unabhängig ist. Daher soll die Betrachtung hier noch verallgemeinert werden. Dazu wird die Breite des Spektrums in Abhängigkeit von der Auflösung der Vakuumwellenlänge λ_0/Δ untersucht. Hierzu wird das Verhältnis

$$R = \frac{D_{\text{ref}}}{D_{\text{WB-TDBPM}}} \tag{8.24}$$

aus der Breite des Eigenwertspektrums des konventionellen Ansatzes und der Breite des TDBPM-Ansatzes definiert. Der Verlauf von R über λ_0/Δ ist Abbildung 8.7 dargestellt. Gilt für das

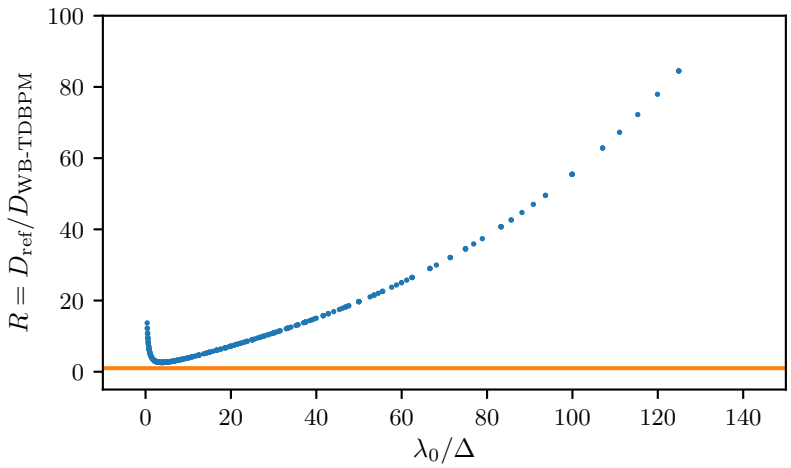


Abbildung 8.7: Das Verhältnis R der Breite D_{ref} des Eigenwertspektrums der Faberpolynom-Methode aus Kapitel 5 zu der Breite des Operators $\mathcal{F}$ in dem Exponential des WB-TDBPM-Ansatzes wird dargestellt. R ist für verschiedene Auflösungen λ_0/Δ mit blauen Punkten dargestellt. Die orangefarbene Linie markiert das Verhältnis $R = 1$.

Verhältnis (8.24) $R = 1$, so befinden sich die Eigenwerte beider Systemmatrizen auf einem gleich großen Intervall. Gilt $R > 1$, so ist das Intervall bei dem TDBPM-Ansatz kleiner. Aus Abbildung 8.7 ist erkennbar, dass zu keinem Zeitpunkt $R < 1$ gilt. Daher ist das Intervall mit den Eigenwerten von $\mathcal{F}$ für jeden hier betrachteten Wert für λ_0/Δ kleiner als das Referenzsystem. Dies lässt sich dadurch erklären, dass die Padé-Approximation die Eigenwerte von $\mathcal{F}$ effektiv limitiert.

Mit diesen Vorbetrachtungen ist es nun möglich, die Faberpolynom-Approximation für (8.21) durchzuführen. Hierzu soll hier wie in den vorangegangenen Kapiteln das elliptische Konvergenz-

gebiet verwendet werden. Um Rundungsfehler zu vermeiden, sollte, wie in Kapitel 5 dargelegt, die logarithmische Kapazität ρ der Ellipse eins sein [90, 91]. Hierzu wird hier wieder ein Skalierungsfaktor λ_s eingeführt, welcher bereits in Abschnitt 5.3 für die Faberpolynom-Approximation verwendet wird. Für diesen gilt wieder $\Delta t_s = \lambda_s \Delta t$ und $\hat{\mathcal{X}}_s = \hat{\mathcal{X}}/\lambda_s$. Damit lässt sich (8.21) zu

$$\vec{\hat{\phi}}(t_n + \Delta t)) = \exp\left(-j\Delta t\omega_0 \frac{\frac{\Delta t_s \hat{\mathcal{X}}_s}{2}}{\Delta t - \frac{\Delta t_s \hat{\mathcal{X}}_s}{4}}\right) \vec{\hat{\phi}}(t_n)) \tag{8.25}$$

umformulieren. Unter diesen Voraussetzungen können die Parameter der Ellipse mit den in Kapitel 5 beschriebenen Methoden bestimmt werden.

Im Anschluss müssen die Koeffizienten c_m der Faberpolynom-Approximation bestimmt werden. Hierfür steht keine analytische Berechnungsformel, wie für die Matrixexponentiale [90, 91], bereit. Daher muss das Integral in (5.8) zur Bestimmung von c_m numerisch gelöst werden. Zu beachten ist, dass die Funktion in (8.25) eine Singularität hat. Die Singularität liegt allerdings nicht in dem Intervall mit den Eigenwerten von $\hat{\mathcal{X}}$. Wenn ein Konvergenzgebiet verwendet wird, welches das Eigenwertspektrum von $\hat{\mathcal{X}}$ eng umschließt, so ist die Funktion in dem Konvergenzbereich weiterhin analytisch. Daher sind die Voraussetzungen für eine Faberpolynom-Approximation gegeben [84]. Wenn größere Konvergenzbereiche verwendet werden, muss dies jedoch beachtet werden.

Für die Berechnung der Approximation der Propagation kann die kurze Rekursionsbeziehung in (5.12) herangezogen werden, da das elliptische Konvergenzgebiet verwendet wird. Hierfür werden wieder nur Matrix-Vektor-Multiplikationen benötigt, sodass sich der Algorithmus explizit formulieren lässt.

Nun verbleibt die Frage, ob die Skalierung des Eigenwertspektrums mit der Trägerfrequenz und die effektive Limitierung des Operators in dem Matrixexponential in (8.21) durch die Padé-Approximation in eine Reduzierung der Approximationsordnung ermöglicht. Dies soll im folgendem Abschnitt numerisch untersucht werden.

8.3.2 Numerische Evaluation

In diesem Abschnitt soll die Effizienz des vorgestellten WB-TDBPM-Algorithmus auf Basis einer Faberpolynom-Entwicklung in den Blick genommen werden. Die WB-Approximation führt bei dieser zu zusätzlichen Fehlern, während die Faberpolynom-Methode aus Kapitel 5 keine Approximationen verwendet. Daher stellt sich die Frage, ob gegenüber der Faberpolynom-Methode eine Reduzierung der Rechenzeit ermöglicht werden kann. Darüber hinaus soll der Algorithmus in Hinblick auf die Genauigkeit mit dem NB-Algorithmus aus dem letzten Abschnitt verglichen werden. Des Weiteren ist zu untersuchen, wie sich die Genauigkeit und der Rechenaufwand im Vergleich mit der FDTD-Methode verhalten.

In den Voruntersuchungen hat sich gezeigt, dass die Beschaffenheit des Eigenwertspektrums nicht nur von den Materialien und der Ortsdiskretisierung abhängt, sondern auch von der Trägerfrequenz des Einhüllenden-Ansatzes. Darüber hinaus ist zu erwarten, dass die Genauigkeit aufgrund der Padé-Approximation von der Bandbreite der Signale um die Trägerfrequenz abhängt. Dies soll bei den folgenden Untersuchungen genauer beleuchtet werden. Als Referenzansatz dient wieder eine Simulation mit der Faberpolynom-Methode, welche mit einer hohen Entwicklungsordnung berechnet wird.

Vergleich mit der Faberpolynom-Methode

Zuerst soll der vorgeschlagene Algorithmus mit der Faberpolynom-Methode verglichen werden. Die letztere verwendet keinerlei Approximationen neben der eigentlichen Faberpolynom-Approximation. In Abschnitt 5.4 wird außerdem gezeigt, dass die Zeitpropagation mit einer sehr hohen Genauigkeit beschrieben wird. Daher erlaubt diese zweifelsfrei die genauere Approximation. Hier stellt sich vielmehr die Frage, ob der WB-TDBPM-Algorithmus eine effizientere Berechnung erlaubt. Für beide Algorithmen wird dieselbe örtliche Diskretisierung mit einem FD-Yee-Gitter verwendet. Für beide Algorithmen wird ein elliptisches Konvergenzgebiet verwendet. Aus dem Grund können beide Faberpolynom-Approximationen mithilfe der Rekursionsbeziehung (5.12) berechnet werden. Im Folgenden soll die Entwicklungsordnung, welche nötig ist, um einen Zeitschritt Δt zu realisieren, betrachtet werden. Aus dieser lässt sich die Anzahl der Matrix-Vektor-Multiplikationen bestimmen, welche erforderlich sind, um die Simulation mit der Simulationszeit T durchzuführen. Daher soll die Anzahl Matrix-Vektor-Multiplikationen wieder als Maß für die Rechenzeit in der Untersuchung verwendet werden. Um die Untersuchung von der Dimensionalität des Problems möglichst unabhängig durchzuführen, wird die unterschiedliche Komplexität der Matrix-Vektor-Multiplikationen in Tabelle 8.1 vernachlässigt.

Hierzu wird im Folgenden ein Testsystem betrachtet. Bei diesem handelt es sich um ein zweidimensionales Problem, dessen TM-Mode untersucht wird. Die Größe des Simulationsgebietes ist unerheblich, da die Entwicklungsordnung von der Größe unabhängig ist. Für die Schrittweite der örtlichen Diskretisierung gilt $\Delta = \Delta x = \Delta y$. Die Simulationszeit T ist mit $T = 100\,\mathrm{ps}$ gegeben. Die Trägerfrequenz für den WB-TDBPM-Algorithmus wird zunächst auf $f_0 = 200\,\mathrm{THz}$ gesetzt, während der Zeitschritt der Δt variiert wird. Die Faberpolynom-Entwicklung wird für beide Algorithmen abgebrochen, wenn für den Absolutbetrag der Entwicklungskoeffizienten $|c_m| < 10^{-15}$ gilt.

Die Ergebnisse der Untersuchung sind in Abbildung 8.8 zusammengefasst. In der Abbildung ist zu beobachten, dass beide Algorithmen eine große Anzahl von Matrix-Vektor-Multiplikationen benötigen, wenn sehr kleine Zeitschrittweiten Δt vorliegen. Beide Kurven fallen mit steigenden Zeitschritt ab und verwenden daher weniger Matrix-Vektor-Multiplikationen N_{MatVec} für dieselbe Simulationszeit T und sind damit effizienter für größere Zeitschritte. Dieses Verhalten kann auch bei der Untersuchung der Faberpolynom-Methode in Abschnitt 5.4 beobachtet werden. Während zunächst die Faberpolynom-Methode effizienter ist, schneiden sich die beiden Verläufe, sodass der vorgestellte WB-TDBPM-Algorithmus ab diesem Punkt effizienter ist. Der Schnittpunkt liegt bei $\Delta t/\Delta t_{\mathrm{CFL}} \approx 45$. Beide Algorithmen verwenden hierbei Zeitschrittweiten, welche den maximalen Zeitschritt Δt_{CFL} der FDTD-Methode weit überschreiten.

Im zweiten Schritt soll die Abhängigkeit von der örtlichen Auflösung der Trägerschwingung λ_0/Δ untersucht werden. Hierzu werden die Matrix-Vektor-Multiplikationen N_{MatVec}, welche für die Simulation benötigt werden, für verschiedene λ_0/Δ aufgetragen. Abbildung 8.9 zeigt die Ergebnisse dieser Parametervariation. Erneut weisen beide Kurven einen ähnlichen Verlauf auf. Für grobe Auflösungen und daher kleine Werte λ_0/Δ ist der WB-TDBPM-Algorithmus effizienter. Die Kurven nähern sich mit feiner werdender Auflösung immer weiter an. Bei $\lambda_0/\Delta \approx 50$ schneiden sich die Kurven. Ab diesem Wert ist die klassische Faberpolynom-Methode effizienter. Hierbei fällt die Parallele zu dem Verhalten des NB-TDBPM-Algorithmus aus Abschnitt 8.2 auf. Allerdings liegt der Schnittpunkt nun bei deutlich feineren Auflösungen im Vergleich zu dem Wert des NB-Ansatzes von $\lambda_0/\Delta \approx 9$.

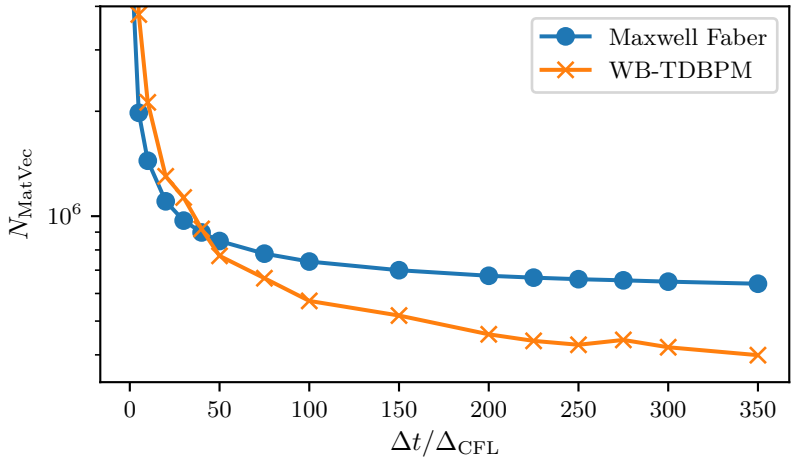


Abbildung 8.8: Die Abbildung zeigt die Anzahl der Matrix-Vektor-Multiplikationen N_{MatVec}, welche die klassische Faberpolynom-Methode und der vorgestellte WB-TDBPM-Algorithmus in Abhängigkeit von der Zeitschrittweite Δt benötigen. Die Zeitschrittweite Δt ist auf die CFL-Zeitschrittweite Δt_{CFL} des Gitters normiert.

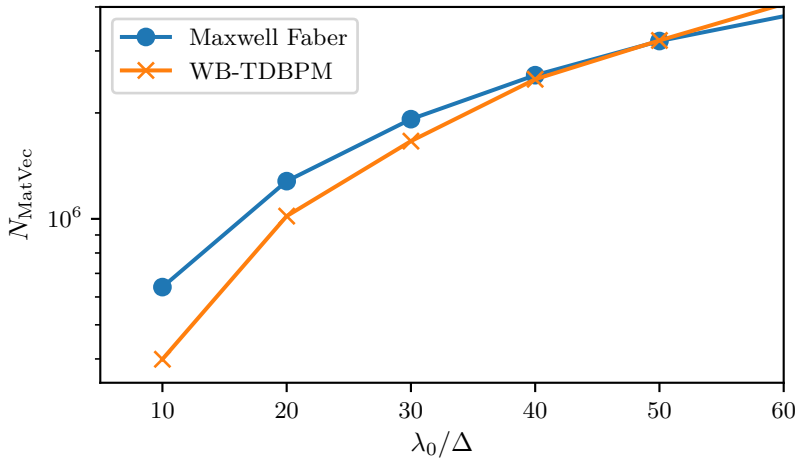


Abbildung 8.9: Die Abbildung zeigt die Anzahl der nötigen Matrix-Vektor-Multiplikationen N_{MatVec}, in Abhängigkeit von der örtlichen Auflösung λ_0/Δ der Vakuumwellenlänge der Trägerschwingung f_0.

Insgesamt lässt sich in den Untersuchungen feststellen, deren Ergebnisse in den Abbildungen 8.9 und 8.8 vorliegen, dass der WB-TDBPM-Algorithmus eine Approximation mit einem geringeren Rechenaufwand erlaubt. Im Gegensatz zu dem NB-Ansatz ist dies auch in einem technisch sinnvollen Bereich möglich.

Vergleich mit dem NB-TDBPM Ansatz

Im Folgendem soll der Einfluss der verwendeten Padé-Approximation ermittelt werden. Die Genauigkeit wird in Abhängigkeit von der Bandbreite um das Trägersignal untersucht. Unter diesem Aspekt soll der WB-Algorithmus mit dem NB-TDBPM-Algorithmus verglichen werden. Hierbei ist zu erwarten, dass der WB-Algorithmus genauere Ergebnisse ermöglicht.

Bei dem verwendeten Testsystem wird die Propagation eines Impulses in einem eindimensionalen System betrachtet. Die Propagation eines gaußförmigen Impulses entlang der z-Achse wird untersucht. Für das Simulationsgebiet gilt $z \in [0, L_z]$ mit $L_z = 10\,\mathrm{mm}$. Für $z > 3L_z/4$ liegen eine Permittivität von $\epsilon = 4\epsilon_0$ und Permeabilität $\mu = \mu_0$ vor, während für alle anderen Bereiche für Permittivität und Permeabilität $\epsilon = \epsilon_0$ beziehungsweise $\mu = \mu_0$ gelten. Der Impuls ist mit der Trägerfrequenz $f_0 = 200\,\mathrm{THz}$ moduliert. Für die örtliche Auflösung wird eine Schrittweite verwendet, welche $\lambda_0/\Delta = 30$ entspricht. Die Simulationszeit beträgt $T = 5\,\mathrm{ps}$ und es wird der Zeitschritt $\Delta t = 200\Delta t_{\mathrm{CFL}}$ gewählt.

Um den Fehler durch die Padé-Approximation bei dem WB-Algorithmus beziehungsweise den Fehler bei dem NB-Algorithmus zu untersuchen, wird die Bandbreite B der Einhüllenden des gaußförmigen Impulses variiert. Der Impuls wird für $t = 0$ im Simulationsgebiet initialisiert. Es soll der Fehler der beiden Methoden soll bestimmt werden, wobei die Faberpolynom-Methode als Referenz dient. Hierbei wird die finale Feldverteilung für $t = T$ der E_x-Komponente betrachtet. Alternativ könnte allerdings auch der zeitliche Verlauf einer Feldkomponente entlang eines Messpunktes für die Fehlerberechnung herangezogen werden. Hierbei werden die Ergebnisse der TDBPM-Simulationen, welche als komplexe Einhüllende vorliegen, zunächst wieder in reelle Größen zurücktransformiert. Es wird mit $\epsilon_{rel} = \sqrt{\sum_{k=1}^{N_z}(E_{x,ref}(z = z_k, t = T) - E_x(z = z_k, t = T))^2}/\sqrt{\sum_{k=1}^{N_z} E_{x,ref}^2(z = z_k, t = T)}$ die euklidische Norm herangezogen. Die Ergebnisse sind in Abbildung 8.10 dargestellt. Der Verlauf

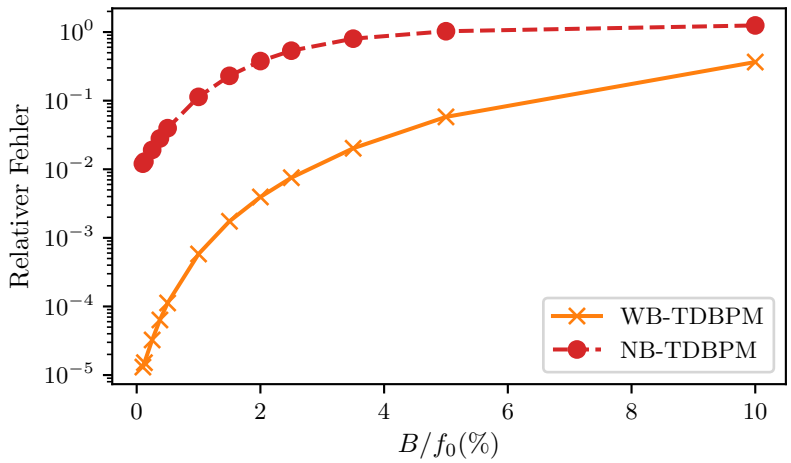


Abbildung 8.10: In der Abbildung ist der relative Fehler der NB-TDBPM- und der WB-TDBPM-Methode in Abhängigkeit von der normierten Bandbreite B/f_0 dargestellt.

des Fehlers nimmt für die beiden Ansätze einen ähnlichen Verlauf. Erwartungsgemäß ist der Fehler bei geringen Bandbreiten B kleiner. Mit steigender Bandbreite B steigt auch der Fehler.

Im Vergleich zu dem NB-Algorithmus liegt bei dem WB-Algorithmus ein deutlich niedrigerer Fehler vor. Die Padè-Approximation führt also zu einer genaueren Approximation.

Vergleich mit der FDTD-Methode

Nun soll der WB-Algorithmus mit der klassischen FDTD-Methode verglichen werden. In Abschnitt 5.4 wird gezeigt, dass die Faberpolynom-Methode für hohe Genauigkeitsanforderungen deutlich effizienter ist. Der hier vorgestellte WB-Algorithmus auf Basis von Faberpolynomen verwendet Approximationen, welche einen geringeren Rechenaufwand für die Evaluation ermöglicht. Hierbei stellt sich nun die Frage, wie sich die Genauigkeit des WB-Algorithmus im Vergleich mit der FDTD-Methode verhält. Für den Vergleich wird das Testsystem des letzten Abschnitts verwendet. Erneut wird die Bandbreite B des Impulses variiert. Für beide Algorithmen wird dasselbe örtliche Gitter verwendet. Um einen Vergleich im Hinblick auf die Effizienz zu erlauben, wird für die FDTD-Simulation der Zeitschritt Δt so gewählt, dass die Anzahl der Zeitschritte $N_T = T/\Delta t$ der Anzahl der Matrix-Vektor-Multiplikationen entspricht, welche für die Berechnung des WB-Algorithmus benötigt wird. Diese Anzahl bestimmt die Berechnungszeit der Faberpolynom-Approximation maßgeblich. So kann von einem näherungsweise gleichen Rechenaufwand ausgegangen werden. Die Ergebnisse der Untersuchung sind in Abbildung 8.11 dargestellt. Wie in der vorangegangenen Untersuchung in Abbildung 8.10 wird der Fehler der finalen Feldverteilung der E_x-Komponente betrachtet. Die Berechnung erfolgt analog. In der

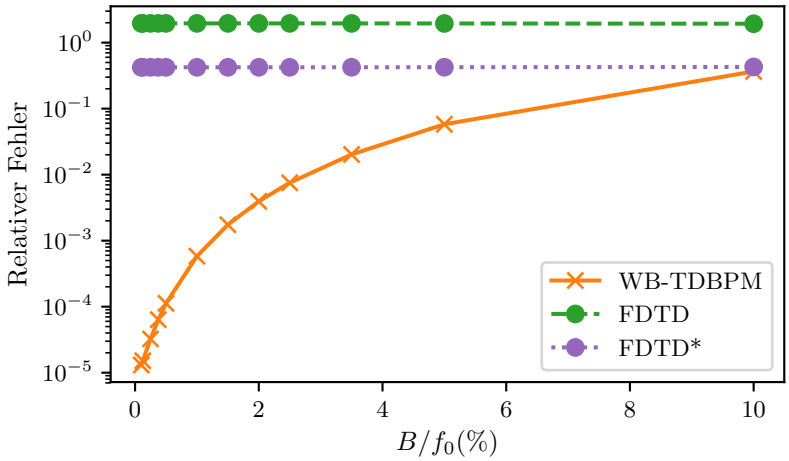


Abbildung 8.11: Die Abbildung zeigt den relativen Fehler des WB-TDBPM-Algorithmus und der FDTD-Methode in Abhängigkeit von der normierten Bandbreite B/f_0 der betrachteten Signale. Bei der FDTD-Methode entspricht die Anzahl der Zeitschritte Δt der Anzahl der Matrix-Vektor-Multiplikationen des WB-TDBPM-Algorithmus. Die Kurve FDTD* zeigt den Fehler der zweiten Simulationsreihe, bei der die Zeitschritte der FDTD-Simulation so gewählt werden, dass die Rechenzeit der gemessenen Rechenzeit des WB-TDBPM-Algorithmus entspricht.

Abbildung kann beobachtet werden, dass der FDTD-Algorithmus keine signifikante Abhängigkeit von der Bandbreite B zeigt. Das ist darin begründet, dass die Bandbreite B in diesem Beispiel

klein im Verhältnis zu der Bandbreite ist, welche von dem FDTD-Algorithmus dargestellt werden kann. Im Vergleich zu dem WB-Algorithmus ist zu erkennen, dass der WB-Algorithmus abhängig von der verwendeten Bandbreite eine höhere Genauigkeit zeigt. Daher erlaubt der WB-Algorithmus bei dem gleichen Rechenaufwand eine genauere Approximation.

Bisher wird hier die Anzahl der Matrix-Vektor-Multiplikationen als Maß für den Rechenaufwand verwendet. Abhängig von der Implementierung der Algorithmen und auch durch die unterschiedliche Struktur der Matrizen $\mathcal{H}$ und $\hat{\mathcal{X}}$, kommt es zu Abweichungen. Daher soll zusätzlich die gemessene Rechenzeit einer nicht-parallelen Implementierung in den Blick genommen werden. Hier hat der Faberpolynom-Ansatz für die Referenz im Mittel 723,2 s benötigt. Die FDTD-Methode benötigt im Mittel 104,7 s, die NB-TDBPM-Methode in Abbildung 8.10 1085 s und die untersuchte WB-TDBPM-Methode 264,8 s. Um diese Diskrepanz auszugleichen, werden weitere FDTD-Simulationen betrachtet. Bei diesen wird die Zeitschrittweite Δt so weit verringert, dass die Rechenzeit mit derjenigen der WB-TDBPM-Methode übereinstimmt. Die resultierende Kurve ist ebenfalls in Abbildung 8.11 dargestellt. Diese zeigt im Vergleich mit der ersten FDTD-Simulationsreihe einen geringeren Fehler. Allerdings ist der Fehler der WB-TDBPM-Methode, insbesondere für geringe Bandbreiten B, weiterhin kleiner als der Fehler der FDTD-Methode.

8.3.3 Bewertung

In den Untersuchungen konnte gezeigt werden, dass der WB-Algorithmus zum einen eine niedrigere Entwicklungsordnung als die Faberpolynom-Methode benötigt und zum anderen auch eine genauere Approximation der Zeitpropagation im Vergleich zu dem FDTD-Algorithmus erlaubt. In der Vorbetrachtung in Abschnitt 8.3.1 werden die Eigenwertspektren der vorliegenden Systemmatrizen untersucht, um Grenzen für den neuen Ansatz abschätzen. Da bei dem WB-Algorithmus aufgrund des anderen Operators kein direkter Vergleich mit der klassischen Faberpolynom-Methode mit dem Matrixexponential möglich ist, wird der Operator in dem Matrixexponential in (8.21) untersucht, um eine Einschätzung von dessen Eigenschaften zu erlangen. Hierbei stellt sich heraus, dass die Padé-Approximation dafür sorgt, dass der Funktionswert im Exponenten in (8.21) effektiv begrenzt wird. Außerdem liegt wieder eine Skalierungseigenschaft vor, durch welche das Eigenwertspektrum des Operators mit der Trägerfrequenz skaliert wird. Allerdings handelt es sich bei (8.21) um eine komplexere Funktion im Vergleich zu einem einfachen Matrixexponential, was die Approximation aufwendiger macht. Beim Vergleich der WB-Methode mit der konventionellen Faberpolynom-Methode in 8.3.2, lässt sich feststellen, dass die Effizienz der WB-Methode sowohl von der verwendeten Zeitschrittweite Δt als auch von der örtlichen Auflösung der Trägerfrequenz abhängig ist. Die Diskrepanz zu der Vorbetrachtung lässt sich unter anderem durch die Singularität der Operatorfunktion in (8.21) erklären. Aufgrund der Singularität ist die mit den Faberpolynomen zu approximierende Funktion nicht mehr analytisch auf der gesamten komplexen Ebene. Die Funktion in (8.21) ist allerdings analytisch in dem Eigenwertspektrum von $\hat{\mathcal{X}}$. So ist es möglich, einen Konvergenzbereich zu wählen, auf dem die Funktion analytisch ist. Dadurch konvergiert die Faberpolynom-Entwicklung. Allerdings liegt keine superlineare Konvergenz vor wie bei der Exponentialfunktion, welche analytisch auf der gesamten komplexen z-Ebene ist [84]. Außerdem liegt für die Funktion in (8.21) keine analytische Berechnungsvorschrift für die Bestimmung der Entwicklungskoeffizienten vor. Eine ungenaue numerische Approximation der Koeffizienten führt zu einem zusätzlichen Fehler. Bei den numerischen Untersuchungen zeigt sich, dass die Berechnung der Koeffizienten, insbesondere für sehr

feine Auflösungen λ_0/Δ und große Zeitschritte Δt, anfällig für Fehler bei der Approximation ist.

8.4 Diskussion und Ausblick

In den vorangegangenen Abschnitten werden ein NB- und ein WB-TDBPM-Algorithmus untersucht. Durch die Operatorentwicklung auf Basis von Faberpolynomen ist es möglich, eine explizite Berechnungsvorschrift für den Algorithmus zu erhalten. Die explizite Formulierung erlaubt eine direkte parallele Implementierung des Algorithmus. Der NB-Ansatz auf Basis von Faberpolynomen zeigt bei der Evaluation der Eigenwertspektren, dass bei diesem eine starke Abhängigkeit von der Auflösung der gewählten Trägerfrequenz vorliegt. Für die zweidimensionale TM-Mode führt dies dazu, dass der NB-Algorithmus nur für Auflösungen von $\lambda_0/\Delta < 9$ effizienter sein kann als der Faberpolynom-Algorithmus ohne TDBPM-Formulierung. Dadurch ist dieser für die meisten praktischen Probleme ungeeignet. Der WB-Ansatz erweist sich insofern als vielversprechender, da bei ihm auch eine solche Grenze vorliegt, diese aber bei deutlich feineren Auflösungen auftritt. Für die untersuchte zweidimensionale TM-Mode liegt diese Grenze bei $\lambda_0/\Delta \approx 50$.

Ein Erfolg versprechender Ansatz für weitere Untersuchungen ist die Verwendung von Padé-Approximationen höherer Ordnung. Durch diese kann die Genauigkeit verbessert werden. Außerdem ergeben sich hierdurch weitere Freiheitsgrade zur Steuerung der Approximation.

Die Einbindung von dämpfenden Medien und PMLs ist mit dem oben beschriebenen Ansatz ohne weiteres möglich. Allerdings kann die Einbindung dazu führen, dass eine Ellipse mit einer Fläche eingesetzt werden muss, welche größer als das Eigenwertspektrum ist. Dies kann darin resultieren, dass die Singularität näher bei der Ellipse oder sogar in ihr liegt. Das verschlechtert die Konvergenz oder verhindert sie vollständig. Hier bieten sich andere Formen für den Konvergenzbereich an, wie sie in Abschnitt 5.3 vorgestellt werden.

9 Anwendungen

In den vorangegangenen Kapiteln werden verschiedene numerische Lösungsverfahren für die Maxwell-Gleichungen auf Basis von Faberpolynomen theoretisch und numerisch untersucht. Die vorstellten Verfahren ermöglichen die Simulation von einer Vielzahl verschiedener technischer Komponenten und Systeme. Insbesondere erlaubt die flexible Formulierung die Implementierung diverser Materialmodelle für die Simulationsmodelle. So können sowohl dispersive Materialien als auch nichtlineare Modelle betrachtet werden.

In diesem Abschnitt werden einige Anwendungsmöglichkeiten für die zuvor untersuchten Algorithmen beleuchtet. Zunächst soll ein dreidimensionales System gekoppelter Wellenleiter betrachtet werden. Im Anschluss werden die entwickelten Algorithmen genutzt, um ein Lasermodell zu konstruieren.

9.1 Gekoppelte Wellenleiter

Zuerst soll ein dreidimensionales System mit zwei gekoppelten Wellenleitern betrachtet werden. Die Faberpolynom-Methode wird hier verwendet, um die Kopplung zwischen den Wellenleitern zu untersuchen. Dieses System wird hier herangezogen, da es sich durch die Größe des Simulationsgebietes und die Anwendung der CFS-PML auszeichnet. Die Ergebnisse der Faberpolynom-Methode werden mit denen der FDTD-Methode verglichen. Teile der folgenden Untersuchungen sind in [KS6] publiziert und werden ergänzt dargestellt.

Bei den untersuchten Wellenleitern handelt es sich um Rechteckwellenleiter, welche parallel zueinander verlaufen. Ein Querschnitt in der x-y-Ebene ist in Abbildung 9.1 gegeben. Die Propagation erfolgt in z-Richtung. Die Kerne der Wellenleiter haben eine Brechzahl von $n_K = 3,673$, während der die Kerne umgebende Mantel eine Brechzahl von $n_M = 1,444$ aufweist. Die Breite der Wellenleiter ist mit $w = 500\,\mathrm{nm}$ gegeben, während die Höhe $h = 220\,\mathrm{nm}$ beträgt. Der Abstand zwischen den Wellenleitern ist $p = 200\,\mathrm{nm}$. Die Abmessungen der Wellenleiter sind an denen in [139] orientiert. Das Rechengebiet hat die Abmessungen $L_x = 2{,}62\,\mu\mathrm{m}$, $L_y = 2{,}22\,\mu\mathrm{m}$ und $L_z = 48\,\mu\mathrm{m}$. Das Rechengebiet wird mit dem FD-Verfahren nach Yee [5] diskretisiert. Hierbei werden die Schrittweiten $\Delta x = \Delta y = \Delta z = 20\,\mathrm{nm}$ für die örtliche Diskretisierung verwendet. Außerdem wird in z-Richtung das Rechengebiet mithilfe einer CFS-PML [41] begrenzt. Die Formulierung erfolgt mit der in Abschnitt 2.4 beschriebenen ADE-Methode. Für die PML werden die Parameter $\alpha_{\max} = 0$, $\kappa_{\max} = 1$ und $\sigma_{\max} = 530\,883$ verwendet. Die PML weist eine Weite von zehn Gitterpunkten auf. Die in Abbildung 9.1 gezeigte Mode wird im linken Wellenleiter des Systems angeregt. Die hierzu benötigten Ströme werden mit der in [33] beschriebenen ADE-Methode eingebunden. Die Modenfelder werden mithilfe eines numerischen Modenlösers auf Basis der GTL-Methode bestimmt [115]. Für diesen wird auch das zuvor beschriebene örtliche

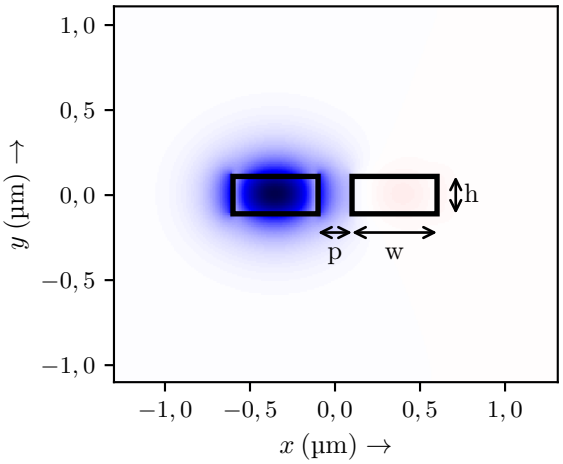


Abbildung 9.1: Die Abbildung zeigt das System mit den beiden Wellenleitern in der x-y-Ebene. Der Rand der Kerne der Wellenleiter wird durch die schwarzen Rechtecke angegeben. Die Verteilung der E_x-Komponente der eingekoppelten Mode wird gezeigt, welche aus der Überlagerung der ersten symmetrischen und der ersten asymmetrischen Systemmode bestimmt wird.

Diskretisierungs-Schema verwendet. Die Moden werden für eine Frequenz von $f_0 = 180\,\mathrm{THz}$ bestimmt.

Für die Faberpolynom-Approximation werden die Grenzen des Spektrums mit den in Abschnitt 5.3.2 beschriebenen Methoden bestimmt. Mithilfe der Methode lassen sich die Werte $c = 27{,}84 \times 10^{15}\,/\mathrm{s}$ für den Realteil und $l = 66{,}28 \times 10^{15}\,/\mathrm{s}$ für den Imaginärteil bestimmen. Für die Faberpolynom-Methode wird eine Zeitschrittweite von $\Delta t = 30\Delta t_{\mathrm{CFL}}$ verwendet. Die Approximation wird ausgeführt, bis für den Betrag der Entwicklungskoeffizienten $|c_m| < 10^{-15}$ gilt. Hierzu ist hier eine Polynomordnung von $N_P = 139$ nötig. Die Simulationszeit beträgt $T = 0{,}809\,\mathrm{ps}$.

In Abbildung 9.2 sind die Ergebnisse der Simulation dargestellt. Mit den Strömen wird in dem linken Wellenleiter eine geführte Eigenmode angeregt, die zunächst in diesem propagiert. Durch den geringen Abstand p der beiden Wellenleiter kommt es zur Kopplung zwischen ihnen. Nach einer Koppellänge L_K ist die Mode komplett von dem linken Wellenleiter in den rechten Wellenleiter gekoppelt. Mithilfe der Ausbreitungskonstanten, welche bei der Bestimmung der Modenfelder mit berechnet werden, wird die Referenzlösung bestimmt. Die Koppellänge ergibt sich zu $L_{K,ref} = 44{,}74\,\mathrm{µm}$. Die Kopplungslänge der Faberpolynom-Methode ist mit $L_{K,F} = 42{,}32\,\mathrm{µm}$ gegeben, während für den FDTD-Algorithmus eine Länge von $L_{K,\mathrm{FDTD}} = 41{,}72\,\mathrm{µm}$ bestimmt wird. Die Faberpolynom-Methode erlaubt daher eine präzisere Berechnung als die konventionelle FDTD-Methode auf demselben Gitter. Dies ist durch die genaue Approximation des Zeitpropagationsschemas zu erklären, welche in Abschnitt 5.4 untersucht wird. Der verbleibende Unterschied zu der Referenzlösung lässt sich durch die Fehler der örtlichen Diskretisierung erklären. Hier spielt insbesondere der Fehler in z-Richtung eine entscheidende Rolle.

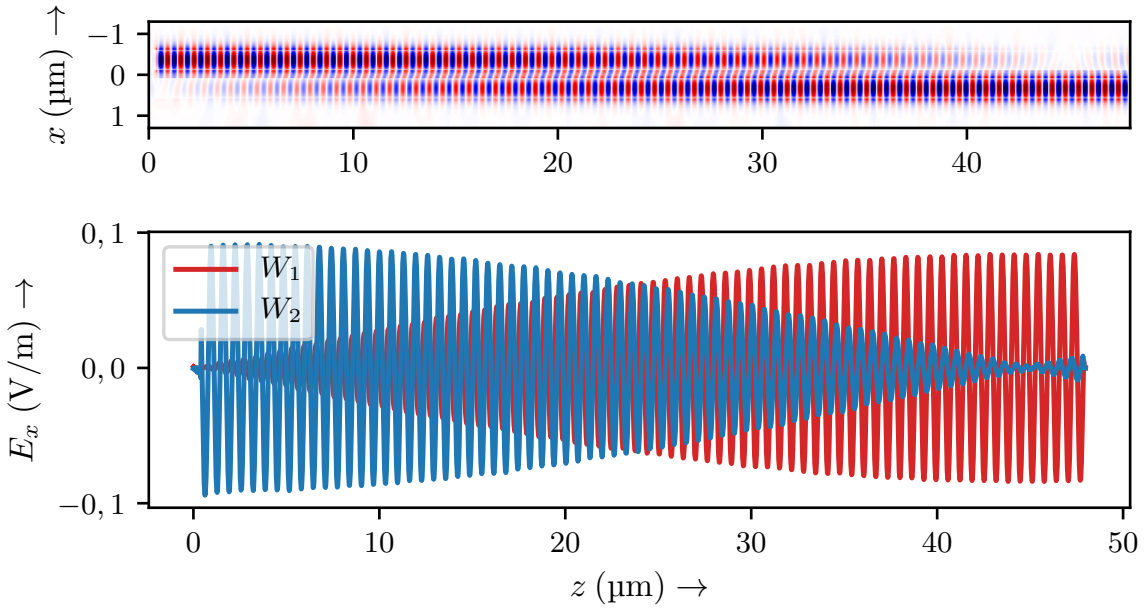


Abbildung 9.2: In Abbildung 9.2a ist für $t = T$ die finale Verteilung der E_x-Komponente in der x-z-Ebene dargestellt. Abbildung 9.2b zeigt die Werte der E_x-Komponenten in der Mitte der Wellenleiter, wobei W_1 den rechten Wellenleiter angibt und W_2 den linken Wellenleiter zeigt.

9.2 Nichtlineares System: Lasermodell

In diesem Abschnitt soll der in Kapitel 7 beschriebene Algorithmus für die Berücksichtigung von nichtlinearen Effekten verwendet werden, um ein Lasermodell zu entwickeln. Mit diesem soll die Möglichkeit des Faberpolynom-Algorithmus, komplexe nichtlineare dynamische Systeme zu betrachten, gezeigt werden. Hierzu wird das in Abschnitt 2.2.3 beschriebene Zwei-Niveau-Modell verwendet.

Die betrachtete Teststruktur orientiert sich an den in [29, 140, 141] verwendeten Strukturen. Die Teststruktur ist in Abbildung 9.3 schematisch dargestellt. Hierbei liegt ein strahlender Übergang zwischen den beiden Niveaus 1 und 2 vor. Dieser wird durch das in Abschnitt 2.2.3 gegebene Oszillatormodell modelliert. Dieses Modell hat eine Übergangsfrequenz $\omega_L = 7{,}535 \times 10^{16}\,\mathrm{Hz}$ und eine FWHM-Bandbreite von $\Gamma_L = 1{,}507 \times 10^{16}\,\mathrm{Hz}$. Die Größe σ_L beschreibt die Kopplung mit dem elektrischen Feld und ist hier mit $\sigma_L = 5{,}716 \times 10^{-7}\,\mathrm{C}^2/\mathrm{kg}$ gegeben [26, 27]. Wie in Abbildung 9.3 schematisch dargestellt, haben die beiden Niveaus die Besetzungsdichten N_1 und N_2. Die gesamte Teilchendichte $N_0 = N_1 + N_2$ wird zu $N_0 = 1 \times 10^{29}\,/\mathrm{m}^3$ gewählt. Für $t = 0$ werden $N_1 = N_2 = N_0/2$ gewählt. Der Gleichgewichtszustand, der sich ohne äußeres Feld einstellt, liegt bei der hier verwendeten Konfiguration bei $N_{1,e} = 0{,}49985 N_0$ und $N_{2,e} = 0{,}50015 N_0$. Die Größe γ_{21} ist die Übergangsrate von dem Niveau 2 zu Niveau 1. Die γ_{12} ist die Übergangsrate von Niveau 1 zu Niveau 2, welche in diesem vereinfachten Lasermodell die Pumpstrahlung

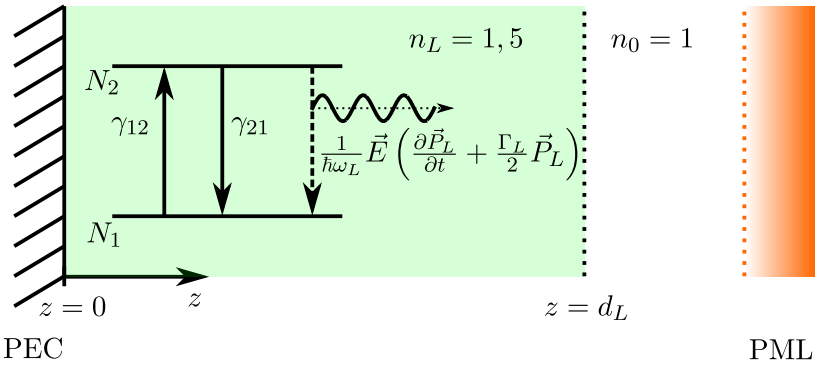


Abbildung 9.3: In der Abbildung ist die untersuchte Laserstruktur skizziert. Der grün eingefärbte Bereich stellt das aktive Medium dar, welches durch das Zwei-Niveau-Modell beschrieben wird. Der orange gefärbte Bereich zeigt das PML-Gebiet. Außerdem ist das Zwei-Niveau-Modell schematisch dargestellt.

modelliert. Für die beiden Größen sind mit $\gamma_{21} = 1{,}4989 \times 10^{12}\,/\mathrm{s}$ und $\gamma_{12} = 1{,}5289 \times 10^{12}\,/\mathrm{s}$ gegeben.

Der Resonator wird durch den PEC-Spiegel auf der linken Seite mit $z = 0$ und den Brechzahlsprung bei $z = d_L$ realisiert. Die Länge d_L des Resonators wird so gewählt, dass der Resonator mit $d_L = 10\lambda_L$ eine Länge von zehn Wellenlängen $\lambda_L = 2\pi c_0/\omega_L/n_L$ der Übergangsfrequenz ω_L hat.

Die rechte Seite des Simulationsgebietes ist mit einer CFS-PML mit $N_{\mathrm{PML}} = 50$ Abtastpunkten abgeschlossen. Das Simulationsgebiet ist mit einem FD-Yee-Gitter mit $\Delta z = 0{,}25\,\mathrm{nm}$ örtlich diskretisiert. Die Länge des Simulationsgebietes beträgt $L_z = 0{,}25\,\mu\mathrm{m}$. Die Zeitpropagation wird mit einem Lawson-Euler-Verfahren durchgeführt. Hierzu wird das System analytisch in einen linearen und einen nichtlinearen Teil aufgeteilt. Bei der Zeitpropagation wird eine Zeitschrittweite von $\Delta t = 5\Delta t_{\mathrm{CFL}}$ verwendet.

Der Zeitverlauf des Ausgangssignals des Lasers wird in dem Bereich mit $n_0 = 1$ an dem Brechzahlsprung gemessen. Hierbei wird die E_x-Komponente des Feldes gemessen. Die Simulation wird mit einem gaußförmigen Impuls in dem aktiven Bereich initialisiert. Die Simulation wird ausgeführt, bis sich ein stabiler Zustand eingestellt hat. Hierzu wird eine Simulationszeit $T = 2\,\mathrm{ps}$ verwendet. Die Ergebnisse sind in den Abbildungen 9.4, 9.5 und 9.6 dargestellt. In Abbildung 9.4 ist zu erkennen, dass drei Frequenzen in dem Laser anschwingen. Eine der drei Frequenzen entspricht dabei der Übergangsfrequenz f_L. Die Abbildung 9.5 zeigt die Besetzungsdichte in der Struktur. In dem Verlauf der Besetzungsdichte ist ein örtliches Lochbrennen an den Maxima der elektrischen Feldstärke wie in [140] zu erkennen. Dieses ist an dem PEC-Spiegel am ausgeprägtesten. In dem Einschwingvorgang in Abbildung 9.6 zeigen sich verschiedene transiente Vorgänge, bevor sich der stabile Zustand mit den drei Frequenzen in Abbildung 9.4 einstellt.

An dem vorliegendem Beispiel ist zu erkennen, dass die Faberpolynom-Methode auch die Modellierung von komplexen nichtlinearen Materialien erlaubt. Die Zeitbereichsformulierung ermöglicht hierbei auch die Untersuchung von dynamischen Effekten wie Einschwingvorgängen. Die Faberpolynom-Methode weist hierbei weiterhin in allen Fällen eine explizite Formulierung

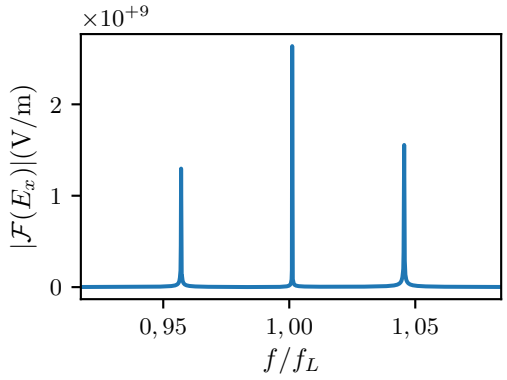


Abbildung 9.4: Die Abbildung zeigt die Fouriertransformierte des gemessenen Ausgangssignals, aufgetragen über die Frequenz f. Die Frequenz ist mit f/f_L ist auf die Übergangsfrequenz $f_L = \omega_L/(2\pi)$ normiert.

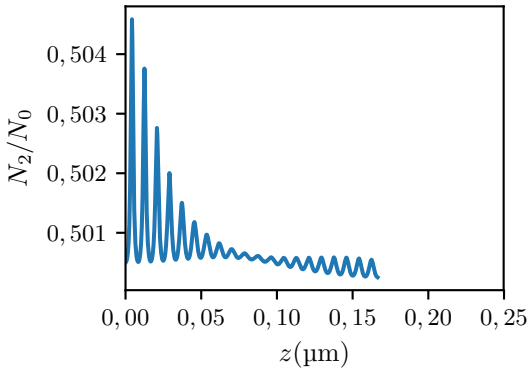


Abbildung 9.5: In der Abbildung ist die normierte Verteilung der Besetzungsdichten N_2/N_0 in Abhängigkeit von z gegeben.

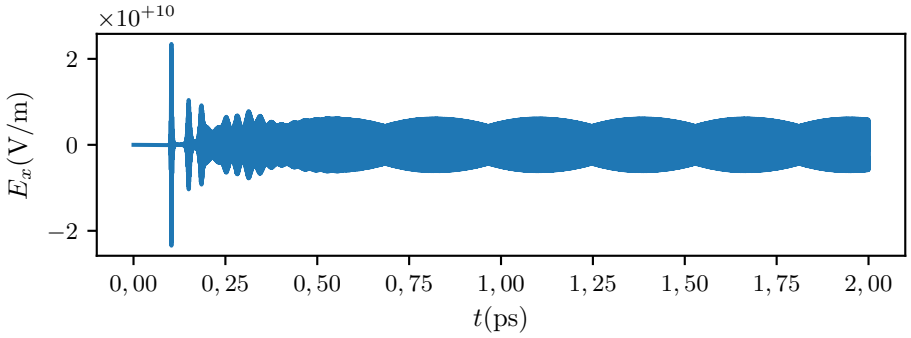


Abbildung 9.6: Die Abbildung zeigt den Einschwingvorgang in der betrachteten Laserstruktur. Der Verlauf der E_x-Komponente am Ausgang der Laserstruktur wird dargestellt.

der Algorithmen auf und gewährleistet so eine einfache Parallelisierung. Außerdem sind bei dem vorliegendem Modell noch weitere Optimierungen möglich. Wird der lineare Teil in den Blick genommen, so fällt auf, dass die Ratengleichungen völlig von den restlichen linearen Termen entkoppelt sind. Die Kopplung erfolgt nur über den nichtlinearen Term, welcher den strahlenden Übergang beschreibt. So kann die Systemmatrix in zwei kommutierende Matrizen $\mathcal{H} = \mathcal{A} + \mathcal{B}$ aufgeteilt werden. Das erlaubt die Berechnung der beiden Einzellösungen des linearen Teils ohne Splitting-Fehler. Der Term mit den Ratengleichungen besteht aus örtlich untereinander nicht gekoppelten Differenzialgleichungen. Diese können sehr effizient berechnet werden, was weitere Einsparungen in der Rechenzeit erlaubt. Das Modell kann noch weiter verbessert werden, indem mehr Niveaus in die Beschreibung aufgenommen werden. Dies ist mit den vorgestellten Modellgleichungen ohne weiteres möglich.

10 Zusammenfassung und Fazit

Im Rahmen dieser Arbeit werden Algorithmen zur numerischen Lösung der Maxwell-Gleichungen für Problemstellungen der Photonik und Terahertz-Technik untersucht.

In den Kapiteln 2 und 3 werden zunächst systematische Ansätze zur Einbindung von Randbedingungen, zur Einkopplung von Feldern, zur Modellierung von Materialeigenschaften und zur örtlichen Diskretisierung erarbeitet. Insbesondere die ADE-Methode erweist sich in diesem Zusammenhang aufgrund ihrer generellen Beschreibung als effizientes Instrument zur Einbindung von Materialmodellen. Anschließend wird in Kapitel 4 ein erster Ansatz zur Zeitpropagation auf Basis einer Polynomentwicklung untersucht. Dieser Ansatz ermöglicht die Verwendung von Zeitschrittweiten Δt, die größer als das CFL-Limit sind. Diese größeren Zeitschrittweiten werden durch die Berücksichtigung des Eigenwertspektrums der Systemmatrix und der spektralen Eigenschaften der betrachteten Feldgrößen erreicht. Die Besonderheit ist hierbei das Stabilitätskriterium, was eine stabile Propagation unabhängig von der Entwicklungsordnung der Polynomapproximation erlaubt. Die Entwicklungsordnung steuert ausschließlich den Phasenfehler der propagierenden Wellen. Allerdings erlaubt die Formulierung des Algorithmus keine direkte Integration von dämpfenden Materialmodellen in die Systemmatrix.

Daher wird in Kapitel 5 ein weiterer Algorithmus auf Basis einer Operatorapproximation mit Faberpolynomen erarbeitet. Die verwendete Formulierung des Zeitpropagationsschemas erlaubt dabei eine systematische Einbindung von linearen Materialmodellen. Die Approximation mit den Faberpolynomen benötigt allerdings Informationen bezüglich des Eigenwertspektrums der Systemmatrix. Um Informationen zu erlangen, wird ein Algorithmus vorgestellt, welcher eine Approximation der Grenzen des Eigenwertspektrums auf Basis einer Entwicklung nach ebenen Wellen ermöglicht. Mit diesem Algorithmus können für beliebige lineare Materialmodelle die Grenzen des Eigenwertspektrums approximiert werden. Mit diesem Algorithmus kann die Faberpolynom-Approximation ohne die explizite Bestimmung von Eigenwerten bezüglich der Systemmatrix berechnet werden. Die Approximation wird für ein elliptisches Konvergenzgebiet durchgeführt, welches flexibel an das Eigenwertspektrum der Systemmatrix angepasst werden kann. Die Zeitschrittweite kann hierbei deutlich größer gewählt werden, als es das CFL-Kriterium für den FDTD-Algorithmus erlaubt. Hierbei können Zeitschrittweiten gewählt werden, welche das CFL-Kriterium um mehr als den Faktor 10^2 überschreiten. Auf diese Weise ist die Wahl der Zeitschrittweite effektiv von der örtlichen Diskretisierung entkoppelt. Bei dem Vergleich der Effizienz des Zeitpropagationsalgorithmus für lineare Probleme mit der klassischen FDTD-Methode stellt sich heraus, dass die Faberpolynom-Methode für hohe Genauigkeitsanforderungen eine um Größenordnungen effizientere Approximation erlaubt. Für niedrige Genauigkeitsanforderungen ist die FDTD-Methode effizienter. Die explizite Formulierung der Faberpolynom-Methode erlaubt hierbei ebenfalls eine unproblematische Parallelisierung der Algorithmen. Die im Rahmen dieser Arbeit betrachteten Algorithmen werden hierbei sowohl auf Mehrkern-Prozessoren als auch auf GPUs implementiert.

In Kapitel 6 wird der Algorithmus um die Einbindung von Quelltermen erweitert. Deren Einbindung ist in der Literatur für die Faberpolynom-Methode bisher noch nicht betrachtet worden. Die Herausforderung ist hierbei, dass eine Einbindung mit einer hohen Genauigkeit in einer Vielzahl von zusätzlichen Matrixfunktionen resultiert. In jedem Zeitschritt müssen diese Matrixfunktionen zusätzlich zu dem linearen Teil ausgewertet werden. Bei einer direkten Einbindung der Matrixfunktionen kann der Rechenaufwand für deren Evaluation den Rechenaufwand für den linearen Teil übersteigen. Um dies zu vermeiden, werden mehrere Ansätze vorgestellt und untersucht. Bei den Betrachtungen stellt sich insbesondere der Komplexe-Einhüllenden-Ansatz als leistungsfähig für bandbegrenzte Quellterme heraus. Bei diesem Ansatz wird die Trägerfrequenz des Systems in die Systemmatrix mit aufgenommen. Auf diese Weise muss lediglich die Einhüllende bei der Approximation der Quellterme berücksichtigt werden. Diese Berücksichtigung führt bei gleicher Entwicklungsordnung zu einer deutlichen Steigerung der Genauigkeit. Wird zusätzlich die Ortsverteilung der Quellterme bei dem Komplexe-Einhüllenden-Ansatz berücksichtigt, kann die Berechnung von zusätzlichen Matrixfunktionen vermieden werden. Die Berechnung kann, auch für hohe Ordnungen, auf einfache Vektor-Additionen reduziert werden. Für Fälle, in denen sich ein solcher Einhüllenden-Ansatz nicht anbietet, wird noch ein weiterer Ansatz entwickelt. Bei diesem Ansatz wird die Approximation der Quellterme direkt in die Systemmatrix entwickelt. Diese Integration vermeidet die Evaluation zusätzlicher Matrixfunktionen auf Kosten einer minimal größeren Systemmatrix vollständig. Im Gegensatz zu früheren Ansätzen auf Basis von ADEs zeichnet sich diese Methode durch ihre allgemeine Formulierung aus.

Diese allgemeine Formulierung erlaubt auch die Anwendung für nichtlineare Probleme, worauf in Kapitel 7 eingegangen wird. Der im Rahmen dieser Arbeit entwickelte Algorithmus ist der erste, auf der Faberpolynom-Methode basierende Algorithmus zur numerischen Lösung der Maxwell-Gleichungen mit nichtlinearen Materialien. Zu diesem Zweck werden zunächst verschiedene Algorithmen zur Beschreibung der Nichtlinearität beleuchtet. Die Entwicklung des nichtlinearen Modells in die Systemmatrix erlaubt hierbei eine effiziente Realisierung der Algorithmen mithilfe von Faberpolynomen. Im Anschluss werden verschiedene Algorithmen miteinander verglichen. Bei dem Vergleich erweisen sich insbesondere die Exponential-Rosenbrock-Methoden für starke nichtlineare Effekte als vielversprechend. Im Gegensatz zu vielen Algorithmen, die im Kontext der klassischen FDTD-Methode für die Einbindung von Nichtlinearitäten verwendet werden, sind die hier betrachteten Methoden alle vollständig explizit. Analog zu den Ergebnissen für die linearen Systeme zeigen sich die Faberpolynom-Methoden auch für nichtlineare Systeme im Vergleich zu dem Referenzansatz effizienter, wenn hohe Genauigkeitsanforderungen vorliegen. Bei gleichem Rechenaufwand konnten um mehrere Größenordnungen geringere relative Fehler im Vergleich zum Referenzansatz erreicht werden. In Kapitel 8 werden die Erkenntnisse der letzten Kapitel genutzt, um einen Operatorapproximation basierten TDBPM-Algorithmus zu konstruieren. Die TDBPM-Algorithmen erlauben zusätzliche Approximationen bezüglich der betrachteten Felder. In diesem Zusammenhang wird zum einen gezeigt, dass gegenüber der Faberpolynom-Methode der Approximationsaufwand gesenkt werden kann. Zum anderen kann gezeigt werden, dass der Algorithmus weiterhin effizienter als der FDTD-Algorithmus ist. Im Anschluss werden die Algorithmen auf einige praktische Beispiele angewandt.

Insgesamt hat sich die Faberpolynom-Methode als interessante Alternative zu klassischen Algorithmen für die numerische Lösung der Maxwell-Gleichungen erwiesen. Insbesondere die Flexibilität bei der Approximation birgt noch weiteres Potenzial. Die Faberpolynom-Methode erlaubt eine flexible Wahl des Konvergenzgebietes. Zu diesem Zweck müssen weitere Informationen

zur Gestalt des Eigenwertspektrums der Systemmatrizen erlangt werden. Diese Informationen können direkt für eine effizientere Approximation verwendet werden. Diese Anpassung ist besonders für nichtlineare Problemstellungen interessant, bei denen sich die zu entwickelnde Matrixfunktionen in jedem Zeitschritt ändern. So kann nicht nur die Zeitschrittweite wie bei vielen klassischen numerischen Lösungsverfahren adaptiv angepasst werden, sondern auch die Form des Konvergenzbereiches in der komplexen Ebene. Die Faberpolynom-Methode ermöglicht die Zeitschrittweite flexibel zu wählen, was in Kombination mit Parallel-In-Time-Methoden [142, 143] oder Local Time-Stepping (LTS)-Methoden [144] interessante Anwendungsmöglichkeiten bietet. Bei den LTS-Methoden kann das Simulationsgebiet beispielsweise in verschiedene Bereiche mit feiner und grober örtlicher Diskretisierung eingeteilt werden. Für diese Bereiche können mit der Faberpolynom-Methode verschiedene Approximationen bestimmt werden, um beide Bereiche mit der gleichen Zeitschrittweite zu berechnen.

Bei den TDBPM-Algorithmen bieten insbesondere die WB-Algorithmen noch Potenzial für weitere Optimierungen. Hier können höherwertige Approximationen bei der Definition des WB-TDBPM-Algorithmus verwendet werden, um weitere Freiheitsgrade bei der Approximation zu erlangen.

Während der Schwerpunkt hier auf der Realisierung der Zeitpropagation liegt, sollte die Faberpolynom-Methode mit weiteren örtlichen Diskretisierungen untersucht werden. Hier bieten sich insbesondere pseudospektrale Ansätze mit einer Unterteilung in des Rechengebietes [42, 58, 145] sowie diskontinuierliche Galerkin-Methoden an [17, 91, 146].

Im Rahmen dieser Arbeit werden die Matrixfunktionen mithilfe von Faberpolynom-Methode approximiert. Viele der hierzu vorgestellten Methoden können auch für andere Berechnungsansätze auf Basis von Krylov-Unterraum-Verfahren oder Leja-Punkt-Methoden angewendet werden. Insbesondere der Komplexe-Einhüllenden-Ansatz für die Quellterme und der TDBPM-Ansatz auf Basis von Operatorapproximationen haben auch bei diesen Berechnungsansätzen das Potenzial für große Effizienzsteigerungen.

Literaturverzeichnis

[1] J. Jin, *The Finite Element Method in Electromagnetics*, 3rd. Wiley-IEEE Press, 2014.

[2] G. R. Hadley, „Wide-angle beam propagation using Padé approximant operators“, *Opt. Lett.*, Jg. 17, Nr. 20, S. 1426, Okt. 1992.

[3] K. Q. Le, „Complex Padé approximant operators for wide-angle beam propagation“, *Opt. Commun.*, Jg. 282, Nr. 7, S. 1252–1254, Apr. 2009.

[4] A. Taflove und S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Third. Artech House, 2005.

[5] Kane Yee, „Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media“, *IEEE Trans. Antennas Propag.*, Jg. 14, Nr. 3, S. 302–307, 1966.

[6] A. Bourgeade und E. Freysz, „Computational modeling of second-harmonic generation by solution of full-wave vector Maxwell equations“, *J. Opt. Soc. Am. B*, Jg. 17, Nr. 2, S. 226, Feb. 2000.

[7] T.-W. Lee und S. C. Hagness, „Pseudospectral time-domain methods for modeling optical wave propagation in second-order nonlinear materials“, *J. Opt. Soc. Am. B*, Jg. 21, Nr. 2, S. 330, Feb. 2004.

[8] T. Namiki, „A new FDTD algorithm based on alternating-direction implicit method“, *IEEE Trans. Microw. Theory Tech.*, Jg. 47, Nr. 10, S. 2003–2007, 1999.

[9] J. Shibayama, M. Muraki, J. Yamauchi und H. Nakano, „Efficient implicit FDTD algorithm based on locally one-dimensional scheme“, *Electron. Lett.*, Jg. 41, Nr. 19, S. 1046, 2005.

[10] I. Ahmed, E. K. Chua, E. P. Li und Z. Chen, „Development of the three-dimensional unconditionally stable LOD-FDTD method“, *IEEE Trans. Antennas Propag.*, Jg. 56, Nr. 11, S. 3596–3600, 2008.

[11] G. Sun und C. Trueman, „Approximate Crank–Nicolson Schemes for the 2-D Finite-Difference Time-Domain Method for TEz Waves“, *IEEE Trans. Antennas Propag.*, Jg. 52, Nr. 11, S. 2963–2972, Nov. 2004.

[12] S. Garcia, Tae-Woo Lee und S. Hagness, „On the accuracy of the ADI-FDTD method“, *IEEE Antennas Wirel. Propag. Lett.*, Jg. 1, Nr. 2, S. 31–34, 2002.

[13] S. Garcia, R. Rubio, A. Bretones und R. Martin, „On the dispersion relation of ADI-FDTD“, *IEEE Microw. Wirel. Components Lett.*, Jg. 16, Nr. 6, S. 354–356, Juni 2006.

[14] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, Second Edi. Courier, 2001, S. 668.

[15] Q. H. Liu, „The pseudospectral time-domain (PSTD) method: a new algorithm for solutions of Maxwell's equations“, in *IEEE Antennas and Propagation Society International Symposium 1997. Digest*, Bd. 1, Juli 1997, 122–125 vol.1.

[16] S. D. Gedney, C. Luo, B. Guernsey, J. A. Roden, R. Crawford und J. a. Miller, „The Discontinuous Galerkin Finite Element Time Domain Method (DGFETD): A High Order, Globally-Explicit Method for Parallel Computation“, in *2007 IEEE Int. Symp. Electromagn. Compat.*, IEEE, Juli 2007, S. 1–3.

[17] K. Busch, M. König und J. Niegemann, „Discontinuous Galerkin methods in nanophotonics“, *Laser Photon. Rev.*, Jg. 5, Nr. 6, S. 773–809, 2011.

[18] J. D. Jackson, *Classical electrodynamics*, 3rd ed. New York: John Wiley & Sons., 1999.

[19] J. Joannopoulos, S. Johnson, J. Winn und R. Meade, *Photonic Crystals: Molding the Flow of Light*, Second. Princeton University Press, 2011.

[20] K. Okamoto, *Fundamentals of Optical Waveguides*. Elsevier, 2006.

[21] P. B. Johnson und R. W. Christy, „Optical Constants of the Noble Metals“, *Phys. Rev. B*, Jg. 6, Nr. 12, S. 4370–4379, Dez. 1972.

[22] P. Drude, „Zur Elektronentheorie der Metalle“, *Ann. Phys.*, Jg. 306, Nr. 3, S. 566–613, 1900.

[23] S. A. Maier, *Plasmonics: Fundamentals and Applications*. Boston, MA: Springer US, 2007.

[24] H. Ibach und H. Lüth, *Festkörperphysik*, 7. Aufl., Ser. Springer-Lehrbuch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, Bd. 91, S. 399–404.

[25] J. L. Young und R. O. Nelson, „A summary and systematic analysis of FDTD algorithms for linearly dispersive media“, *IEEE Antennas Propag. Mag.*, Jg. 43, Nr. 1, S. 61–77, 2001.

[26] R. W. Boyd, *Nonlinear Optics, Third Edition*, 3rd. Orlando, FL, USA: Academic Press, Inc., 2008.

[27] P. Bermel, E. Lidorikis, Y. Fink und J. D. Joannopoulos, „Active materials embedded in photonic crystals and coupled to electromagnetic radiation“, *Phys. Rev. B*, Jg. 73, Nr. 16, S. 165 125, 2006.

[28] S.-L. Chua, Y. Chong, A. D. Stone, M. Soljacic und J. Bravo-Abad, „Low-threshold lasing action in photonic crystal slabs enabled by Fano resonances“, *Opt. Express*, Jg. 19, Nr. 2, S. 1539, Jan. 2011.

[29] MEEP. (2019). Saturable Gain and Absorption, Adresse: `https://meep.readthedocs.io/en/latest/Materials/#saturable-gain-and-absorption` (besucht am 31. 01. 2021).

[30] A. Nagra und R. York, „FDTD analysis of wave propagation in nonlinear absorbing and gain media“, *IEEE Trans. Antennas Propag.*, Jg. 46, Nr. 3, S. 334–340, März 1998.

[31] S. I. Azzam und A. V. Kildishev, „Time-domain dynamics of saturation of absorption using multilevel atomic systems“, *Opt. Mater. Express*, Jg. 8, Nr. 12, S. 3829, Dez. 2018.

[32] G. Slavcheva, J. Arnold und R. Ziolkowski, „FDTD Simulation of the Nonlinear Gain Dynamics in Active Optical Waveguides and Semiconductor Microcavities“, *IEEE J. Sel. Top. Quantum Electron.*, Jg. 10, Nr. 5, S. 1052–1062, Sep. 2004.

[33] K. Busch, J. Niegemann, M. Pototschnig und L. Tkeshelashvili, „A Krylov-subspace based solver for the linear and nonlinear Maxwell equations“, *Phys. status solidi*, Jg. 244, Nr. 10, S. 3479–3496, 2007.

[34] J. Zimmerling, L. Wei, P. Urbach und R. Remis, „A Lanczos model-order reduction technique to efficiently simulate electromagnetic wave propagation in dispersive media“, *J. Comput. Phys.*, Jg. 315, S. 348–362, Juni 2016.

[35] J. Niegemann, „Higher-Order Methods for Solving Maxwell's Equations in the Time-Domain“, Diss., 2012.

[36] G. Mur, „Absorbing Boundary Conditions for the Finite-Difference Approximation of the Time-Domain Electromagnetic-Field Equations“, *IEEE Trans. Electromagn. Compat.*, Jg. EMC-23, Nr. 4, S. 377–382, 1981.

[37] B. Engquist und A. Majda, „Absorbing Boundary Conditions for the Numerical Simulation of Waves“, *Math. Comput.*, Jg. 31, Nr. 139, S. 629, 1977.

[38] J.-P. Berenger, „A perfectly matched layer for the absorption of electromagnetic waves“, *J. Comput. Phys.*, Jg. 114, Nr. 2, S. 185–200, 1994.

[39] Z. Sacks, D. Kingsland, R. Lee und Jin-Fa Lee, „A perfectly matched anisotropic absorber for use as an absorbing boundary condition“, *IEEE Trans. Antennas Propag.*, Jg. 43, Nr. 12, Intergovernmental Panel on Climate Change, Hrsg., S. 1460–1463, 1995.

[40] S. Gedney, „An anisotropic perfectly matched layer-absorbing medium for the truncation of FDTD lattices“, *IEEE Trans. Antennas Propag.*, Jg. 44, Nr. 12, S. 1630–1639, 1996.

[41] S. D. Gedney und B. Zhao, „An Auxiliary Differential Equation Formulation for the Complex-Frequency Shifted PML“, *IEEE Trans. Antennas Propag.*, Jg. 58, Nr. 3, S. 838–847, 2010.

[42] A. Taflove, A. Oskooi und S. G. Johnson, *Advances in FDTD Computational Electrodynamics: Photonics and Nanotechnology*, First. Artech House, 2013.

[43] T. Tan und M. Potter, „FDTD Discrete Planewave (FDTD-DPW) Formulation for a Perfectly Matched Source in TFSF Simulations“, *IEEE Trans. Antennas Propag.*, Jg. 58, Nr. 8, S. 2641–2648, Aug. 2010.

[44] J. Schneider, „Plane Waves in FDTD Simulations and a Nearly Perfect Total-Field/Scattered-Field Boundary“, *IEEE Trans. Antennas Propag.*, Jg. 52, Nr. 12, S. 3280–3287, Dez. 2004.

[45] K. Abdijalilov und J. B. Schneider, „Analytic field propagation TFSF boundary for FDTD problems involving planar interfaces: Lossy material and evanescent fields“, *IEEE Antennas Wirel. Propag. Lett.*, Jg. 5, Nr. 1, S. 454–458, 2006.

[46] C.-p. Yu und H.-c. Chang, „Yee-mesh-based finite difference eigenmode solver with PML absorbing boundary conditions for optical waveguides and photonic crystal fibers“, *Opt. Express*, Jg. 12, Nr. 25, S. 6165, 2004.

[47] J. Schneider. (2010). Understanding the Finite-Difference Time-Domain Method, Adresse: `www.eecs.wsu.edu/~schneidj/ufdtd` (besucht am 31. 01. 2021).

[48] U. Inan und R. Marshall, *Numerical Electromagnetics: The FDTD Method*. Cambridge University Press, 2011.

[49] Q. H. Liu, „The PSTD algorithm: A time-domain method requiring only two cells per wavelength“, *Microw. Opt. Technol. Lett.*, Jg. 15, Nr. 3, S. 158–165, 1997.

[50] T.-W. Lee und S. C. Hagness, „Pseudospectral time-domain methods for modeling optical wave propagation in second-order nonlinear materials“, *J. Opt. Soc. Am. B*, Jg. 21, Nr. 2, S. 330, 2004.

[51] F. Devaux, E. Lantz und M. Chauvet, „3D pseudospectral time domain for modeling second-harmonic generation in periodically poled lithium niobate ridge-type waveguides“, *J. Opt. Soc. Am. B*, Jg. 33, Nr. 4, S. 703, 2016.

[52] Y. Liu, „Fourier Analysis of Numerical Algorithms for the Maxwell Equations“, *J. Comput. Phys.*, Jg. 124, Nr. 2, S. 396–416, März 1996.

[53] H. De Raedt, K. Michielsen, J. Kole und M. Figge, „Solving the Maxwell equations by the Chebyshev method: a one-step finite-difference time-domain algorithm“, *IEEE Trans. Antennas Propag.*, Jg. 51, Nr. 11, S. 3155–3160, 2003.

[54] R. A. H. Horn und C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 2012.

[55] T. Weiland, „A Discretization Method for the Solution of Maxwell's Equations for Six-Component Fields“, *AEU Int. J. Electron. C.*, Jg. 31, Nr. 3, S. 116–120, 1977.

[56] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*. Society for Industrial und Applied Mathematics, Jan. 2011, Bd. 66, S. 1–27.

[57] A. Klöckner, N. Pinto, Y. Lee, B. Catanzaro, P. Ivanov und A. Fasih, „PyCUDA and PyOpenCL: A Scripting-Based Approach to GPU Run-Time Code Generation“, *Parallel Computing*, Jg. 38, Nr. 3, S. 157–174, 2012.

[58] M. Ding und K. Chen, „Staggered-grid PSTD on local Fourier basis and its applications to surface tissue modeling“, *Opt. Express*, Jg. 18, Nr. 9, S. 9236, Apr. 2010.

[59] C. Leforestier, R. Bisseling, C. Cerjan, M. Feit, R. Friesner, A. Guldberg, A. Hammerich, G. Jolicard, W. Karrlein, H.-D. Meyer, N. Lipkin, O. Roncero und R. Kosloff, „A comparison of different propagation schemes for the time dependent Schrödinger equation“, *J. Comput. Phys.*, Jg. 94, Nr. 1, S. 59–80, Mai 1991.

[60] W. Gautschi, „Numerical integration of ordinary differential equations based on trigonometric polynomials“, *Numer. Math.*, Jg. 3, Nr. 1, S. 381–397, 1961.

[61] P. Deuflhard, „A study of extrapolation methods based on multistep schemes without parasitic solutions“, *Zeitschrift für Angew. Math. und Phys. ZAMP*, Jg. 30, Nr. 2, S. 177–189, März 1979.

[62] C. Carle, M. Hochbruck und A. Sturm, „On leapfrog-Chebyshev schemes“, Karlsruher Institut für Technologie (KIT), Techn. Ber. 19, 2019, 29 S.

[63] H. Kleene, „Optimierung von Algorithmen zur Simulation transienter elektrodynamischer Ausbreitungsphänomene“, Masterarbeit, TU Dortmund, 2016.

[64] N. J. Higham, *Functions of Matrices: Theory and Computation*. Philadelphia, PA, USA: Society for Industrial und Applied Mathematics, 2008.

[65] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3. Aufl. New York, NY, USA: Cambridge University Press, 2007.

[66] Z. X. Huang, X. L. Wu, W. E. I. Sha und B. Wu, „Optimized Operator-Splitting Methods in Numerical Integration of Maxwell's Equations“, *Int. J. Antennas Propag.*, Jg. 2012, Nr. January, S. 1–8, 2012.

[67] S. MacNamara und G. Strang, „Operator Splitting“, in, 2016, S. 95–114.

[68] G. Faber, „Über polynomische Entwickelungen“, *Mathematische Annalen*, Jg. 57, Nr. 3, S. 389–408, Sep. 1903.

[69] J. H. Curtiss, „Faber Polynomials and the Faber Series“, *Am. Math. Mon.*, Jg. 78, Nr. 6, S. 577, 1971.

[70] S. W. Ellacott, „Computation of Faber Series With Application to Numerical Polynomial Approximation in the Complex Plane“, *Math. Comput.*, Jg. 40, Nr. 162, S. 575, 1983.

[71] P. Novati, „Solving linear initial value problems by Faber polynomials“, *Numer. Linear Algebr. with Appl.*, Jg. 10, Nr. 3, S. 247–270, 2003.

[72] C. Moler und C. Van Loan, „Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later“, *SIAM Rev.*, Jg. 45, Nr. 1, S. 3–49, 2003.

[73] H. De Raedt und K. Michielsen, „Unconditionally stable perfectly matched layer boundary conditions“, *Phys. Status Solidi Basic Res.*, Jg. 244, Nr. 10, S. 3497–3505, 2007.

[74] A. G. Borisov und S. V. Shabanov, „Lanczos Pseudospectral Propagation Method for Initial-Value Problems in Electrodynamics of Passive Media“, S. 11, Okt. 2004.

[75] M. Botchev, „A short guide to exponential Krylov subspace time integration for Maxwell's equations“, 2012.

[76] M. A. Botchev, „Krylov subspace exponential time domain solution of Maxwell's equations in photonic crystal modeling“, *J. Comput. Appl. Math.*, Jg. 293, S. 20–34, 2016.

[77] L. Bergamaschi, M. Caliari, A. Martínez und M. Vianello, „Comparing Leja and Krylov Approximations of Large Scale Matrix Exponentials“, in *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, Bd. 3994 LNCS, 2006, S. 685–692.

[78] M. Caliari, P. Kandolf, A. Ostermann und S. Rainer, „The Leja Method Revisited: Backward Error Analysis for the Matrix Exponential“, *SIAM J. Sci. Comput.*, Jg. 38, Nr. 3, A1639–A1661, Jan. 2016.

[79] M. Merkel, I. Niyonzima und S. Schöps, „ParaExp Using Leapfrog as Integrator for High-Frequency Electromagnetic Simulations“, *Radio Sci.*, Jg. 52, Nr. 12, S. 1558–1569, 2017.

[80] K. O. Geddes und J. C. Mason, „Polynomial Approximation by Projections on the Unit Circle“, *SIAM J. Numer. Anal.*, Jg. 12, Nr. 1, S. 111–120, 1975.

[81] S. Ellacott, „A survey of Faber methods in numerical approximation“, *Comput. Math. with Appl.*, Jg. 12, Nr. 5-6, S. 1103–1107, 1986.

[82] G. Faber, „Über Tschebyscheffsche Polynome“, *Journal für die reine und angewandte Mathematik*, Jg. 150, S. 79–106, 1919.

[83] H. Tal-Ezer, „Polynomial approximation of functions of matrices and applications“, *J. Sci. Comput.*, Jg. 4, Nr. 1, S. 25–60, 1989.

[84] I. Moret und P. Novati, „The computation of functions of matrices by truncated Faber series“, *Numer. Funct. Anal. Optim.*, Jg. 22, Nr. 5-6, S. 697–719, 2001.

[85] I. Moret und P. Novati, „An interpolatory approximation of the matrix exponential based on Faber polynomials“, *J. Comput. Appl. Math.*, Jg. 131, Nr. 1-2, S. 361–380, 2001.

[86] B. Beckermann und L. Reichel, „Error Estimates and Evaluation of Matrix Functions via the Faber Transform“, *SIAM J. Numer. Anal.*, Jg. 47, Nr. 5, S. 3849–3883, 2009.

[87] L. Bergamaschi, M. Caliari und M. Vianello, „Efficient approximation of the exponential operator for discrete 2D advection-diffusion problems“, *Numer. Linear Algebr. with Appl.*, Jg. 10, Nr. 3, S. 271–289, 2003.

[88] W. Huisinga, L. Pesce, R. Kosloff und P. Saalfrank, „Faber and Newton polynomial integrators for open-system density matrix propagation“, *J. Chem. Phys.*, Jg. 110, Nr. 12, S. 5538–5547, 1999.

[89] Y. Huang, D. J. Kouri und D. K. Hoffman, „General, energy-separable Faber polynomial representation of operator functions: Theory and application in quantum scattering“, *J. Chem. Phys.*, Jg. 101, Nr. 12, S. 10 493–10 506, 1994.

[90] A. G. Borisov und S. V. Shabanov, „Wave packet propagation by the Faber polynomial approximation in electrodynamics of passive media“, *J. Comput. Phys.*, Jg. 216, Nr. 1, S. 391–402, 2006.

[91] H. Fahs, „Investigation on polynomial integrators for time-domain electromagnetics using a high-order discontinuous Galerkin method“, *Appl. Math. Model.*, Jg. 36, Nr. 11, S. 5466–5481, 2012.

[92] L. Bergamaschi und M. Vianello, „Efficient computation of the exponential operator for large, sparse, symmetric matrices“, *Numer. Linear Algebr. with Appl.*, Jg. 7, Nr. 1, S. 27–45, Jan. 2000.

[93] E. Freitag und R. Busam, *Complex Analysis*. Springer-Verlag Berlin Heidelberg, 2009.

[94] T. Kövari, „On the order of polynomial approximation for closed Jordan domains“, *J. Approx. Theory*, Jg. 5, Nr. 4, S. 362–373, 1972.

[95] T. Driscoll, „Algorithm 756; a MATLAB toolbox for Schwarz-Christoffel mapping“, *ACM Trans. Math. Softw.*, Jg. 22, S. 168–186, Juni 1996.

[96] M. Abramowitz, *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables*. Dover Publications, Incorporated, 1974.

[97] R. Lehoucq, D. Sorensen und C. Yang, *ARPACK Users' Guide*. Society for Industrial und Applied Mathematics, 1998.

[98] M. Fiedler, „Geometry of the numerical range of matrices“, *Linear Algebra Appl.*, Jg. 37, Nr. C, S. 81–96, 1981.

[99] P. J. Asarrakos und M. J. Tsatsomeros, „Numerical range: (in) a matrix nutshell“, *Preprint*, S. 13, 2002.

[100] E. Bécache und P. Joly, „On the analysis of Bérenger's Perfectly Matched Layers for Maxwell's equations“, *ESAIM Math. Model. Numer. Anal.*, Jg. 36, Nr. 1, S. 87–119, Jan. 2002.

[101] E. Becache, P. Petropoulos und S. Gedney, „On the Long-Time Behavior of Unsplit Perfectly Matched Layers“, *IEEE Trans. Antennas Propag.*, Jg. 52, Nr. 5, S. 1335–1342, 2004.

[102] M. A. Botchev, I. Faragó und R. Horváth, „Application of operator splitting to the Maxwell equations including a source term“, *Appl. Numer. Math.*, Jg. 59, Nr. 3-4, S. 522–541, 2009.

[103] A. H. Al-Mohy und N. J. Higham, „Computing the Action of the Matrix Exponential, with an Application to Exponential Integrators“, *SIAM J. Sci. Comput.*, Jg. 33, Nr. 2, S. 488–511, 2011.

[104] M. Hochbruck und A. Ostermann, „Exponential integrators“, *Acta Numer.*, Jg. 19, S. 209–286, Mai 2010.

[105] B. V. Minchev und W. M. Wright, „A review of exponential integrators for first order semi-linear problems“, *Preprint Numerics*, Jg. 2, S. 1–45, 2005.

[106] J. Pursel und P. Goggans, „A finite-difference time-domain method for solving electromagnetic problems with bandpass-limited sources“, *IEEE Trans. Antennas Propag.*, Jg. 47, Nr. 1, S. 9–15, 1999.

[107] F. Ma, „Slowly varying envelope simulation of optical waves in time domain with transparent and absorbing boundary conditions“, *J. Lightw. Technol.*, Jg. 15, Nr. 10, S. 1974–1985, 1997.

[108] Changning Ma und Zhizhang Chen, „Stability analysis of the CE-FDTD method“, *IEEE Microw. Wirel. Components Lett.*, Jg. 14, Nr. 5, S. 243–245, 2004.

[109] Changning Ma und Zhizhang Chen, „Stability and numerical dispersion analysis of CE-FDTD method“, *IEEE Trans. Antennas Propag.*, Jg. 53, Nr. 1, S. 332–338, 2005.

[110] Saehoon Ju, Kyung-Young Jung und Hyeongdong Kim, „Investigation on the characteristics of the envelope FDTD based on the alternating direction implicit scheme“, *IEEE Microw. Wirel. Components Lett.*, Jg. 13, Nr. 9, S. 414–416, Sep. 2003.

[111] C. Ma und Z. Chen, „The complex envelope (CE) ADI-FDTD method“, *IEEE MTT-S Int. Microw. Symp. Dig.*, Jg. 2005, Nr. C, S. 1119–1120, 2005.

[112] K.-Y. Jung, F. L. Teixeira, S. G. Garcia und R. Lee, „On Numerical Artifacts of the Complex Envelope ADI-FDTD Method“, *IEEE Trans. Antennas Propag.*, Jg. 57, Nr. 2, S. 491–498, Feb. 2009.

[113] A. Householder, *The Theory of Matrices in Numerical Analysis*, Ser. Dover Books on Mathematics. Dover Publications, 2013.

[114] H. Alt und J. van Leeuwen, „The complexity of basic complex operations“, *Computing*, Jg. 27, Nr. 3, S. 205–215, Sep. 1981.

[115] S. Helfert und R. Pregla, „A finite difference beam propagation algorithm based on generalized transmission line equations“, *Opt. Quantum Electron.*, Jg. 32, Nr. 6–8, S. 681–690, 2000.

[116] A. Koskela und A. Ostermann, „A Moment-Matching Arnoldi Iteration for Linear Combinations of φ Functions“, *SIAM J. Matrix Anal. Appl.*, Jg. 35, Nr. 4, S. 1344–1363, 2014.

[117] J. H. Greene und A. Taflove, „General vector auxiliary differential equation finite-difference time-domain method for nonlinear optics“, *Opt. Express*, Jg. 14, Nr. 18, S. 8305, 2006.

[118] R. Joseph und A. Taflove, „FDTD Maxwell's equations models for nonlinear electrodynamics and optics“, *IEEE Trans. Antennas Propag.*, Jg. 45, Nr. 3, S. 364–374, 1997.

[119] M. Fujii, M. Tahara, I. Sakagami, W. Freude und P. Russer, „High-Order FDTD and Auxiliary Differential Equation Formulation of Optical Pulse Propagation in 2-D Kerr and Raman Nonlinear Dispersive Media“, *IEEE J. Quantum Electron.*, Jg. 40, Nr. 2, S. 175–182, 2004.

[120] C. Varin, G. Bart, R. Emms und T. Brabec, „Saturable Lorentz model for fully explicit three-dimensional modeling of nonlinear optics“, *Opt. Express*, Jg. 23, Nr. 3, S. 2686, 2015.

[121] C. Varin, R. Emms, G. Bart, T. Fennel und T. Brabec, „Explicit formulation of second and third order optical nonlinearity in the FDTD framework“, *Comput. Phys. Commun.*, Jg. 222, S. 70–83, 2018.

[122] M. Pototschnig, J. Niegemann, L. Tkeshelashvili und K. Busch, „Time-Domain Simulations of the Nonlinear Maxwell Equations Using Operator-Exponential Methods“, *IEEE Trans. Antennas Propag.*, Jg. 57, Nr. 2, S. 475–483, 2009.

[123] M. Hochbruck, C. Lubich und H. Selhofer, „Exponential Integrators for Large Systems of Differential Equations“, *SIAM J. Sci. Comput.*, Jg. 19, Nr. 5, S. 1552–1574, 2003.

[124] M. Tokman, „Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods“, *J. Comput. Phys.*, Jg. 213, Nr. 2, S. 748–776, 2006.

[125] M. Hochbruck, A. Ostermann und J. Schweitzer, „Exponential Rosenbrock-Type Methods“, *SIAM J. Numer. Anal.*, Jg. 47, Nr. 1, S. 786–803, Jan. 2009.

[126] J. Loffeld und M. Tokman, „Comparative performance of exponential, implicit, and explicit integrators for stiff systems of ODEs“, *J. Comput. Appl. Math.*, Jg. 241, Nr. 1, S. 45–67, März 2013.

[127] J. D. Lawson, „Generalized Runge-Kutta Processes for Stable Systems with Large Lipschitz Constants“, *SIAM J. Numer. Anal.*, Jg. 4, Nr. 3, Intergovernmental Panel on Climate Change, Hrsg., S. 372–380, Sep. 1967.

[128] M. Koshiba, Y. Tsuji und M. Hikari, „Time-domain beam propagation method and its application to photonic crystal circuits“, *J. Lightw. Technol.*, Jg. 18, Nr. 1, S. 102–110, 2000.

[129] T. Fujisawa und M. Koshiba, „Time-domain beam propagation method for nonlinear optical propagation analysis and its application to photonic crystal circuits“, *J. Lightw. Technol.*, Jg. 22, Nr. 2, S. 684–691, 2004.

[130] K. Q. Le, T. Benson und P. Bienstman, „Application of modified Padé approximant operators to time-domain beam propagation methods“, *J. Opt. Soc. Am. B*, Jg. 26, Nr. 12, S. 2285, 2009.

[131] K. Le, H. Dantanarayana, E. Romanova, T. Benson und P. Bienstman, „Comparative assessment of time-domain models of nonlinear optical propagation“, in *2009 3rd Ict. Mediterr. Winter Conf.*, Bd. 2, IEEE, Dez. 2009, S. 1–4.

[132] J. Shibayama, M. Muraki, J. Yamauchi und H. Nakano, „Comparative study of several time-domain methods for optical waveguide analyses“, *J. Lightw. Technol.*, Jg. 23, Nr. 7, S. 2285–2293, 2005.

[133] H. M. Masoudi und M. S. Akond, „Time-Domain BPM Technique for Modeling Ultra Short Pulse Propagation in Dispersive Optical Structures: Analysis and Assessment“, *J. Lightw. Technol.*, Jg. 32, Nr. 10, S. 1936–1943, 2014.

[134] M. S. Akond, „Analysis of femtosecond pulsed beam propagation in dispersive directional coupler“, *Opt. Rev.*, Jg. 21, Nr. 3, S. 249–255, 2014.

[135] H. M. Masoudi, „A Novel Nonparaxial Time-Domain Beam-Propagation Method for Modeling Ultrashort Pulses in Optical Structures“, *J. Lightw. Technol.*, Jg. 25, Nr. 10, S. 3175–3184, 2007.

[136] H. M. Masoudi und M. S. Akond, „Stable Time-Domain Beam Propagation Method for Modeling Ultrashort Pulse Propagation in Dispersive Optical Structures“, *IEEE Photonics Technol. Lett.*, Jg. 24, Nr. 9, S. 769–771, 2012.

[137] J. Shibayama, A. Yamahira, T. Mugita, J. Yamauchi und H. Nakano, „A finite-difference time-domain beam-propagation method for TE- and TM-wave analyses“, *J. Lightw. Technol.*, Jg. 21, Nr. 7, S. 1709–1715, 2003.

[138] J. F. Lee und R. Mittra, „A Hybrid Yee Algorithm/Scalar-Wave Equation Approach“, *IEEE Transactions on Microwave Theory and Techniques*, Jg. 41, Nr. 9, S. 1593–1600, 1993.

[139] Z. Lu, H. Yun, Y. Wang, Z. Chen, F. Zhang, N. A. F. Jaeger und L. Chrostowski, „Broadband silicon photonic directional coupler using asymmetric-waveguide based phase control“, *Opt. Express*, Jg. 23, Nr. 3, S. 3795, Feb. 2015.

[140] A. Cerjan, Y. Chong, L. Ge und A. D. Stone, „Steady-state ab initio laser theory for N-level lasers“, *Opt. Express*, Jg. 20, Nr. 1, S. 474, Jan. 2012.

[141] A. Cerjan, Y. D. Chong und A. D. Stone, „Steady-state ab initio laser theory for complex gain media“, *Opt. Express*, Jg. 23, Nr. 5, S. 6455, März 2015.

[142] M. J. Gander und S. Güttel, „PARAEXP: A Parallel Integrator for Linear Initial-Value Problems“, *SIAM J. Sci. Comput.*, Jg. 35, Nr. 2, S. C123–C142, Jan. 2013.

[143] M. J. Gander und S. Vandewalle, „Analysis of the Parareal Time-Parallel Time-Integration Method“, *SIAM J. Sci. Comput.*, Jg. 29, Nr. 2, S. 556–578, Jan. 2007.

[144] M. J. Grote und T. Mitkova, „High-order explicit local time-stepping methods for damped wave equations“, *J. Comput. Appl. Math.*, Jg. 239, Nr. 1, S. 270–289, Feb. 2013.

[145] S.-F. Chiang, B.-Y. Lin, H.-C. Chang, C.-H. Teng, C.-Y. Wang und S.-Y. Chung, „A Multidomain Pseudospectral Mode Solver for Optical Waveguide Analysis“, *J. Lightw. Technol.*, Jg. 30, Nr. 13, S. 2077–2087, 2012.

[146] H. Fahs und S. Lanteri, „A high-order non-conforming discontinuous Galerkin method for time-domain electromagnetics“, *J. Comput. Appl. Math.*, Jg. 234, Nr. 4, S. 1088–1096, Juni 2010.

[147] G. L. Pedrola, *Beam Propagation Method for Design of Optical Waveguide Devices*. Chichester, UK: John Wiley & Sons, Ltd, 2015.

Schriftenverzeichnis

[KS1] C. Spenner, H. Kleene, P. Sarapukdee, K. Kallis und D. Schulz, „Analysis of SiO2- and MgF2-Based Surface Plasmon Resonance Sensors“, in *26th Optical Wave & Waveguide Theory and Numerical Modelling - OWTNM*, Apr. 2018, P–21.

[KS2] H. Kleene und D. Schulz, „Unitary Polynomial Propagator Solving Maxwell's Equations Allowing Arbitrarily Large Time Steps“, *IEEE Photonics Technol. Lett.*, Jg. 30, Nr. 2, S. 193–196, Jan. 2018.

[KS3] H. Kleene und D. Schulz, „Investigation of a Unitary Explicit Algorithm for Electromagnetic Time Domain Simulations“, in *25th Optical Wave & Waveguide Theory and Numerical Modelling - OWTNM*, Apr. 2017, OT1.3.

[KS4] H. Kleene und D. Schulz, „Assessment of Chebychev Expansions for the Approximation of the Time Domain Propagator“, in *Advanced Photonics 2016 (IPR, NOMA, Sensors, Networks, SPPCom, SOF)*, Washington, D.C.: OSA, Juli 2016, JTu4A.11.

[KS5] H. Kleene und D. Schulz, „An Assessment of Polynomial Approximations for the Time-Domain Beam Propagator“, in *24th Optical Wave & Waveguide Theory and Numerical Modelling - OWTNM*, Mai 2016, O–31.

[KS6] H. Kleene und D. Schulz, „Time Domain Solution of Maxwell's Equations using Faber Polynomials“, *IEEE Trans. Antennas Propag.*, Jg. 66, Nr. 11, S. 6202–6208, 2018.

[KS7] H. Kleene und D. Schulz, „An Assessment of Faber Polynomial Expansions for the Time Domain Solution of Maxwell ' s equations“, in *26th Optical Wave & Waveguide Theory and Numerical Modelling - OWTNM*, Apr. 2018, O–3.3.

[KS8] H. Kleene und D. Schulz, „On the Evaluation of Sources in Highly Accurate Time Domain Simulations on the Basis of Faber Polynomials“, in *2018 Prog. Electromagn. Res. Symp.*, IEEE, Aug. 2018, S. 352–356.

[KS9] H. Kleene und D. Schulz, „Concept of a Complex Envelope Faber Polynomial Approach for the Solution of Maxwell's Equations“, in *2018 IEEE MTT-S Int. Conf. Numer. Electromagn. Multiphysics Model. Optim.*, IEEE, Aug. 2018, S. 1–3.

[KS10] H. Kleene und D. Schulz, „Complex envelope Faber polynomial method for the solution of Maxwell's equations“, *Opt. Quantum Electron.*, Jg. 51, Nr. 12, S. 381, Dez. 2019.

[KS11] H. Kleene, T. D. P. Luong und D. Schulz, „Faber Polynomial based Approximations of nonlinear Integrators for Electrodynamics“, *Zur Veröffentlichung eingereicht bei: IEEE J. Multiscale Multiphys. Comput. Tech.*, 2020.

[KS12] H. Kleene und D. Schulz, „Assessment of a Time Domain Beam Propagation Algorithm Based on Faber Polynomial Expansions“, in *2019 IEEE MTT-S Int. Conf. Numer. Electromagn. Multiphysics Model. Optim.*, (Cambridge, MA), IEEE, Mai 2019, WEP.1–I.

[KS13] H. Kleene und D. Schulz, „Explicit Wideband Time-Domain Beam Propagation Algorithm Based on Faber Polynomials“, *IEEE J. Multiscale Multiphys. Comput. Tech.*, Jg. 4, S. 282–289, 2019.

A Abkürzungsverzeichnis

CPU Central Processing Unit

GPU Graphics Processing Unit

ADE Auxiliary-Differential-Equation

TE transversal elektrische

TM transversal magnetische

FDTD Finite-Difference Time-Domain

CFL Courant-Friedrich-Lewy

CFS Complex Frequency Shifted

ADI alternating-direction implicit

LOD locally one-dimensional

LTS Local Time-Stepping

FD Finite Differenzen

PEC perfekt elektrisch leitend

PMC perfekt magnetisch leitend

ABC Absorbing Boundary Conditions

PSTD Pseudospectral Time-Domain

DFT Diskrete Fourier Transformation

DG diskontinuierlich Galerkin

FPGA Field Programmable Gate Array

FIT Finite Integrationstechnik

BPM Beam Propagation Method

TDBPM Time Domain Beam Propagation

NB Narrow-Band

WB Wide-Band

FB Full-Band

PML Perfectly-Matched-Layer

UPML uniaxiale Perfectly-Matched-Layer

TFSF Total field scattered field

FFT schnelle Fourier Transformation

FWHM Halbwertsbreite

GTL Generalized Transmission Line Equations

B Mathematische Umformungen

B.1 Herleitung des unitären Zeitpropagationsschemas

Im Folgenden soll die mathematische Herleitung des unitären Zeitpropagationsschemas aus Kapitel 4 genauer beschrieben werden. Der Formalismus ist mit

$$\vec{\Psi}(t_n + \Delta t) = \mathcal{P}(\mathcal{H})\vec{\Psi}(t_n) + \vec{\Psi}(t_n - \Delta t) \tag{B.1}$$

gegeben. Bei $\mathcal{P}(\mathcal{H})$ handelt es sich um eine zunächst allgemein angesetzte Matrixfunktion. Diese soll für den vorliegenden Fall bestimmt werden. Dazu wird die formale Lösung

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t\mathcal{H})\vec{\Psi}(t_n) \tag{B.2}$$

des örtlich diskretisierten Systems (3.19) benötigt. Die formale Lösung (B.2) wird verwendet, um die Feldverteilungen $\vec{\Psi}(t_n + \Delta t)$ und $\vec{\Psi}(t_n - \Delta t)$ zu den Zeitpunkten $t = t_n + \Delta t$ beziehungsweise $t = t_n - \Delta t$ zu beschreiben. Diese werden im Anschluss in das Zeitpropagationsschema (B.1) eingesetzt:

$$\exp(\Delta t\mathcal{H})\vec{\Psi}(t_n) = \mathcal{P}(\mathcal{H})\vec{\Psi}(t_n) + \exp(-\Delta t\mathcal{H})\vec{\Psi}(t_n). \tag{B.3}$$

Nun wird (B.3) nach $\mathcal{P}(\mathcal{H})$ aufgelöst:

$$\mathcal{P}(\mathcal{H}) = \exp(\Delta t\mathcal{H}) - \exp(-\Delta t\mathcal{H}). \tag{B.4}$$

Anschließend wird die Definition der hyperbolischen Funktionen auf Basis von Exponentialfunktionen verwendet, welche mit

$$\sinh(z) = \frac{\exp(z) - \exp(-z)}{2} \tag{B.5}$$

gegeben ist. Mit (B.5) kann aus (B.4) der Operator

$$\mathcal{P}(\mathcal{H}) = 2\sinh(\Delta t\mathcal{H}) \tag{B.6}$$

bestimmt werden.

B.2 Herleitung des Wachstumsfaktors des unitären Zeitpropagationsschemas

Die Feldverteilung $\vec{\Psi}(t_n)$ wird formal nach den Eigenfunktionen von $\mathcal{H}$ entwickelt. Die nach Eigenfunktionen entwickelte Feldverteilung $\vec{\Psi}(t_n)$ wird mit $\vec{v}(\sigma, t_n)$ bezeichnet. Es wird der Wachstumsfaktors $g(\sigma)$ definiert:

$$\vec{v}(\sigma, t_n + \Delta t) = g(\sigma)\vec{v}(\sigma, t_n). \tag{B.7}$$

Mit diesem Zusammenhang lassen sich die nach Eigenfunktionen entwickelten Feldverteilungen $\vec{v}(\sigma, t_n)$ für die verschiedenen Zeitpunkte $t = t_n + \Delta t$ beziehungsweise $t = t_n - \Delta t$ formal beschreiben. Die Beschreibung (B.7) wird auf das Zeitpropagationsschema (4.2) angewendet und umgeformt:

$$\begin{aligned} g(\sigma)\vec{v}(\sigma, t_n) &= \hat{\mathcal{P}}(\sigma)\vec{v}(\sigma, t_n) + g^{-1}(\sigma)\vec{v}(\sigma, t_n) \\ &\Leftrightarrow 0 = g^2(\sigma)\vec{v}(\sigma, t_n) - g(\sigma)\hat{\mathcal{P}}(\sigma)\vec{v}(\sigma, t_n) - \vec{v}(\sigma, t_n) \\ &\Leftrightarrow 0 = g^2(\sigma) - g(\sigma)\hat{\mathcal{P}}(\sigma) - 1. \end{aligned} \tag{B.8}$$

Die quadratische Gleichung (B.8) wird nach $g(\sigma)$ gelöst:

$$g(\sigma) = \hat{\mathcal{P}}(\sigma)/2 \pm \sqrt{\hat{\mathcal{P}}^2(\sigma)/4 + 1}. \tag{B.9}$$

B.3 Faberpolynome

B.3.1 Vorgehen bei der Approximation

An dieser Stelle soll das Vorgehen bei der Bestimmung der Faberpolynom-Approximation einer Funktion $f(\mathcal{H})$ zusammengefasst werden. Für den allgemeinen Fall lassen sich für die Approximation von $f(\mathcal{H})$ folgende Schritte definieren:

- Ausgangspunkt: Funktion $f(\mathcal{H})$ mit Matrix $\mathcal{H}$.
- Die Approximation von der Funktion $f(\mathcal{H})$ wird als Approximation einer komplexwertigen Funktion $f(z)$ realisiert.
- Das Eigenwertspektrum von $\mathcal{H}$ definiert den minimalen Konvergenzbereich für eine Approximation von $f(\mathcal{H})$, da ein kleiner Konvergenzbereich den Approximationsaufwand minimiert.
- Daher: Abschätzung des Eigenwertspektrums von $\mathcal{H}$ erforderlich.
- Das Konvergenzgebiet K wird so gewählt, dass das Eigenwertspektrum von $\mathcal{H}$ eng umschlossen wird.
- Die Laurententwicklung $\psi(w)$ definiert die Faberpolynome.
- Daher wird die Laurententwicklung der konformen Abbildung $\psi(w)$ bestimmt: Im allgemeinen Fall kann beispielsweise die Schwarz-Christoffel-Transformation genutzt werden, um die Laurententwicklung der konformen Abbildung $\psi(w)$ zu berechnen.
- Die Entwicklungskoeffizienten c_m der Faberpolynom-Approximation von der Funktion $f(\mathcal{H})$ werden mithilfe von Formel (5.8) und der Laurententwicklung von $\psi(w)$ bestimmt.
- Die Faberpolynom-Approximation der Funktion $f(\mathcal{H})$ wird mit der Rekursionsbeziehung (5.7) berechnet.

B.3.2 Zusammenhänge für elliptische Konvergenzgebiete

Rekursionsbeziehung

Die Rekursionsbeziehung der Faberpolynome ist für $m \geq 1$ mit

$$F_{m+1}(z) = zF_m(z) - \sum_{k=0}^{m} \gamma_k F_{m-k}(z) - m\gamma_m \tag{B.10}$$

gegeben, wobei $F_0(z) = 1$ gilt [88, 90]. Bei γ_k beziehungsweise γ_m handelt es sich um den k-ten beziehungsweise den m-ten Koeffizienten der Laurent-Entwicklung der konformen Abbildung $\psi(w)$. Für ein elliptisches Konvergenzgebiet liegt mit

$$\psi(w) = w + \gamma_0 + \gamma_1/w \tag{B.11}$$

ein Spezialfall vor, bei der die Laurent-Entwicklung der konformen Abbildung bei einem niedrigen Grad abbricht [88, 90, 91]. Nun wird (B.11) in die Rekursionsbeziehung (B.10) eingesetzt. Mit der Startbedingung $F_0(z) = 1$ ergibt sich für $F_1(z)$:

$$F_1(z) = zF_0(z) - (\gamma_0 F_0(z)) - 0\gamma_0 \Leftrightarrow F_1(z) \quad = z - \gamma_0. \tag{B.12}$$

Durch weiteres Einsetzen von (B.12) in (B.10) kann $F_2(z)$ mit

$$\begin{aligned} & F_2(z) = zF_1(z) - (\gamma_0 F_1(z) + \gamma_1 F_0(z)) - 1\gamma_1 \\ \Leftrightarrow\ & F_2(z) = zF_1(z) - (\gamma_0 F_1(z) + 1\gamma_1) - 1\gamma_1 \\ \Leftrightarrow\ & F_2(z) = (z - \gamma_0)F_1(z) - 2\gamma_1 \end{aligned} \tag{B.13}$$

bestimmt werden. Um alle weiteren Terme $F_m(z)$ zu bestimmen, wird genutzt, dass in der Entwicklung (B.11) alle Terme γ_k für $k \geq 2$ Null sind. Es gilt also für diesen Spezialfall:

$$\begin{aligned} & F_{m+1}(z) = zF_m(z) - \sum_{k=0}^{m} \gamma_k F_{m-k}(z) - m0 \\ \Leftrightarrow\ & F_{m+1}(z) = zF_m(z) - (\gamma_0 F_m(z) + \gamma_1 F_{m-1}(z) + 0F_{m-2}(z) + \dots) \\ \Leftrightarrow\ & F_{m+1}(z) = (z - \gamma_0)F_m(z) - \gamma_1 F_{m-1}(z). \end{aligned} \tag{B.14}$$

Insgesamt ist die Rekursionsbeziehung für das elliptische Konvergenzgebiet mit

$$F_{m+1}(z) = (z - \gamma_0)F_m(z) - \gamma_1 F_{m-1}(z); m \geq 2 \tag{B.15}$$

gegeben, wobei die Startbedingungen für die Rekursion mit

$$\begin{aligned} F_0(z) &= 1 \\ F_1(z) &= z - \gamma_0 \\ F_2(z) &= (z - \gamma_0)F_1(z) - 2\gamma_1 \end{aligned} \tag{B.16}$$

anzugeben sind [69, 88].

Analytischer Zusammenhang für die Entwicklungskoeffizienten von Exponentialfunktionen

Die Entwicklungskoeffizienten der Faberpolynome für eine Funktion $f(z)$ und einer konformen Abbildung mit der Laurent-Entwicklung $\psi(w)$ sind mit

$$c_m = \frac{1}{2\pi j} \int\limits_{|w|=R} \frac{f(\psi(w))}{w^{m+1}} \mathrm{d}w,\ R \geq \rho \tag{B.17}$$

gegeben [88, 90]. Für Funktionen der Form $f(z) = \exp(\Delta t_s z)$ liegt bei elliptischen Konvergenzgebieten ein Spezialfall vor. Zunächst wird die Abbildung (B.11) und die Funktion $f(z)$ in (B.17) eingesetzt:

$$c_m = \frac{1}{2\pi j} \int\limits_{|w|=R} \frac{\exp(\Delta t_s (w + \gamma_0 + \gamma_1 / w))}{w^{m+1}} \mathrm{d}w. \tag{B.18}$$

Daraufhin wird (B.18) weiter umgeformt:

$$c_m = \frac{\exp(\Delta t_s \gamma_0)}{2\pi j} \int\limits_{|w|=R} \frac{\exp(\Delta t_s (w + \gamma_1 / w))}{w^{m+1}} \mathrm{d}w. \tag{B.19}$$

Im Anschluss wird (B.19) in die folgende äquivalente Form

$$c_m = \frac{1}{2\pi j} \exp(\Delta t_s \gamma_0) \int\limits_{|w|=R} \frac{\exp\left(j 2 \Delta t_s \sqrt{\gamma_1} \left(-jw/\sqrt{\gamma_1} - \frac{1}{-jw/\sqrt{\gamma_1}}\right)\right)}{w^{m+1}} \mathrm{d}w \tag{B.20}$$

umgeschrieben. Nun wird die erzeugende Funktion der Bessel-Funktionen erster Ordnung [96], gegeben mit

$$e^{z(t-1/t)/2} = \sum_{n=0}^{\infty} J_n(z) t^n, \tag{B.21}$$

verwendet. Hierbei handelt es sich bei $J_n(z)$ um die n-te Bessel-Funktion erster Ordnung. Es wird $t = -jw/\sqrt{\gamma_1}$ und $z = j 2 \Delta t_s \sqrt{\gamma_1}$ angesetzt. Durch Einsetzen des Zusammenhangs (B.21) lässt sich (B.20) zu

$$c_m = \exp(\Delta t_s \gamma_0) \sum_{n=0}^{\infty} \left(\frac{1}{j\sqrt{\gamma_1}}\right)^n J_n\left(j 2 \Delta t_s \sqrt{\gamma_1}\right) \frac{1}{2\pi j} \int\limits_{|w|=R} \frac{1}{w^{m+1-n}} \mathrm{d}w \tag{B.22}$$

umschreiben. Im Anschluss wird das Integral in (B.22) mithilfe der Cauchy-Integral-Formel [93]

$$f(z) = \frac{1}{2\pi j} \oint \frac{f(\xi)}{\xi - z} \mathrm{d}\xi \tag{B.23}$$

betrachtet. Dies ist möglich, da $w = |R|$ einen geschlossenen Kreis beschreibt. Hierzu werden zwei Fälle analysiert: Im Fall $n = m$ ergibt sich ein Spezialfall, für den gemäß der Cauchy-Integral-Formel

$$\frac{1}{2\pi j} \int\limits_{|w|=R} \frac{1}{w^1} \mathrm{d}w = \frac{1}{2\pi j} 2\pi j = 1 \tag{B.24}$$

gilt. Für alle Fälle $n \neq m$ gilt gemäß der Cauchy-Integral-Formel

$$\frac{1}{2\pi j} \int\limits_{|w|=R} \frac{w^{n-m}}{w} \mathrm{d}w = 0^{n-m} = 0. \tag{B.25}$$

Damit sind alle Terme der Summe in (B.22) für $n \neq m$ Null. Also lassen sich für diesen Fall die Entwicklungskoeffizienten der Faberpolynom-Approximation der Funktion $f(z) = \exp(\Delta t_s z)$ mit

$$c_m = \exp(\Delta t_s \gamma_0) \left(\frac{1}{j\sqrt{\gamma_1}} \right)^m J_m \left(j2\Delta t_s \sqrt{\gamma_1} \right). \tag{B.26}$$

angeben [90, 91].

Zusammenfassung

Insgesamt lässt sich die Approximation von Exponentialfunktionen für ein elliptisches Gebiet K wie folgt zusammenfassen:

- Ausgangspunkt: Funktion $f(\mathcal{H}) = \exp(\Delta t \mathcal{H})$ mit Matrix $\mathcal{H}$.
- Das Eigenwertspektrum von $\mathcal{H}$ definiert den minimalen Konvergenzbereich für eine Approximation von $f(\mathcal{H})$, da ein kleiner Konvergenzbereich den Approximationsaufwand minimiert.
- Daher ist eine Abschätzung des Eigenwertspektrums von $\mathcal{H}$ erforderlich.
- Das elliptische Konvergenzgebiet K wird an das Eigenwertspektrum von $\mathcal{H}$ angepasst.
- Die Laurententwicklung $\psi(w)$ definiert die Faberpolynome.
- Die Laurententwicklung der konformen Abbildung $\psi(w)$ liegt mit (5.10) analytisch vor. Die Parameter von (5.10) werden aus der Geometrie der Ellipse K berechnet.
- Für die Entwicklungskoeffizienten c_m der Faberpolynom-Approximation liegt mit Formel (5.32) ein analytischer Zusammenhang vor.
- Die Faberpolynom-Approximation der Funktion $f(\mathcal{H})$ kann mit der verkürzten Rekursionsbeziehung (5.12) berechnet werden.

B.4 Herleitung der Breitband-Approximation

Der Ausgangspunkt für die Herleitung ist die vektorielle Wellengleichung mit dem Einhüllenden-Ansatz (8.7). Dieser wird gemäß [130] umformuliert:

$$(-\mathcal{M} + \omega_0^2 I_N)\vec{\hat{\phi}}(t) = \frac{\partial^2}{\partial t^2}\vec{\hat{\phi}}(t) + j2\omega_0 \frac{\partial}{\partial t}\vec{\hat{\phi}}(t). \tag{B.27}$$

Von (8.17) ausgehend wird der Ausdruck weiter umgeformt. Hierzu wird

$$\hat{\mathcal{X}} = -\mathcal{M}/\omega_0^2 + I_N \tag{B.28}$$

definiert. Wird (B.28) in (B.27) eingesetzt, so ergibt sich

$$0 = \frac{\partial^2}{\partial t^2}\vec{\hat{\phi}}(t) + j2\omega_0\frac{\partial}{\partial t}\vec{\hat{\phi}}(t) - \hat{\mathcal{X}}\vec{\hat{\phi}}(t). \tag{B.29}$$

Indem (B.29) als quadratische Gleichung betrachtet wird und nach $\frac{\partial}{\partial t}$ aufgelöst wird, so kann die formale Lösung von (B.27) bestimmt werden [2, 130, 147]:

$$\frac{\partial}{\partial t}\vec{\hat{\phi}}(t) = -j\omega_0\left(I_N - \sqrt{I_N - \hat{\mathcal{X}}}\right)\vec{\hat{\phi}}(t). \tag{B.30}$$

C Beispielrechnungen

C.1 Bestimmung der Faberpolynom-Approximation

Das Vorgehen bei der Bestimmung einer Faberpolynom-Approximation einer komplexen Funktion $f(z)$ soll im Folgenden an einem Beispiel erläutert werden. Es soll die Approximation der Funktion

$$f(z) = \exp(qz) \tag{C.1}$$

in den Blick genommen werden. Der Koeffizient q soll im Anschluss variiert werden. Die Funktion $f(z)$ soll für ein elliptisches Gebiet K in der komplexen Ebene approximiert werden. Dieses Gebiet K ist in Abbildung C.1 dargestellt. Im nächsten Schritt muss eine konforme Abbildung

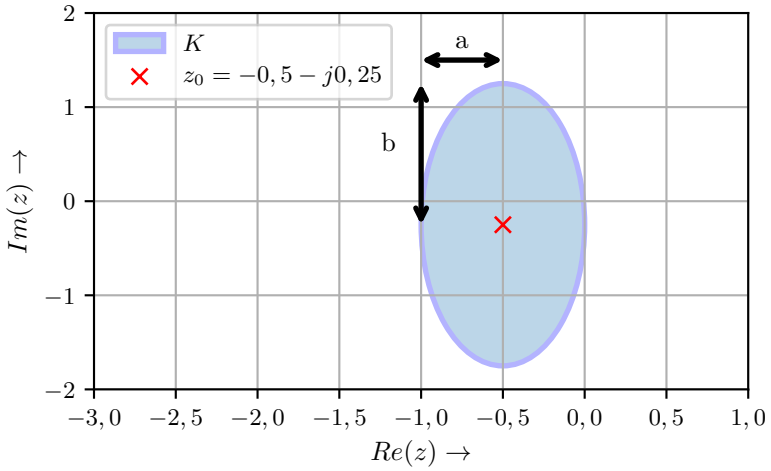


Abbildung C.1: Die Abbildung zeigt das Gebiet K in der komplexen z-Ebene, auf dem $f(z)$ approximiert werden soll. Das Gebiet K ist elliptisch. Die geometrischen Parameter dieser Ellipse sind in der Abbildung gegeben. Für die Ellipse gilt $(x - x_0)^2/a^2 + (y - y0)^2/b^2 = 1$. Hierbei gilt $a = 0,5$ und $b = 1,5$ sowie $z_0 = x_0 + jy_0$.

gefunden werden, welche das Komplement eines Kreises um den Nullpunkt der komplexen w-Ebene auf das Komplement von K abbildet. Zu dieser konformen Abbildung muss eine Laurententwicklung der Form

$$\psi(w) = w + \gamma_0 + \gamma_1 w^{-1} + \gamma_2 w^{-2} + ... = w + \sum_{k\geq 0}^{\infty} \gamma_k w^{-k} \tag{C.2}$$

gefunden werden. Deren Bestimmung ist für komplexe Fälle, wie oben beschrieben, mithilfe der Schwarz-Christoffel-Transformation möglich [90, 91, 95]. Die Entwicklung (C.2) muss an einer geeigneten Stelle abgebrochen werden. Dies kann allerdings zu Abweichungen von dem angestrebten Konvergenzbereich K führen [71]. Auf der anderen Seite ist es wünschenswert, eine möglichst kurze Entwicklung zu finden, da die Faberpolynome mit der Rekursionsbeziehung (5.7) bestimmt werden sollen. Zusätzliche Koeffizienten in (C.2) erhöhen den Berechnungsaufwand der Rekursionsbeziehung (5.7) und insbesondere den Speicheraufwand. Daher ist es vorteilhaft, wie in Kapitel 5 beschrieben, auf abbrechende Entwicklungen zurückzugreifen. Diese liegen für eine Reihe von Formen K in der Literatur vor. Dies ist auch für das elliptische Gebiet K in Abbildung C.1 der Fall:

$$\psi(w) = w + \gamma_0 + \gamma_1/w. \tag{C.3}$$

Die Koeffizienten γ_0 und γ_1 sind aus der Geometrie der Ellipse zu bestimmen. Der Parameter γ_0 hängt mit dem Mittelpunkt der Ellipse zusammen: $\gamma_0 = x_0 + jy_0$. In diesem Fall gilt, wie in Abbildung C.1 zu erkennen, $\gamma_0 = -0,5 - j0,25$. Der Parameter γ_1 hängt mit den Halbachsenparametern a und b der Ellipse zusammen. Diese sind mit $\rho = (a+b)/2$ verknüpft. Für γ_1 gilt $\gamma_1 = \rho^2 - \rho b$. Hierbei ist ρ die sogenannte logarithmische Kapazität der Ellipse K. Um die Stabilität der Entwicklung zu gewährleisten, sollte $\rho \leq 1$ gelten [83, 88, 90, 91]. Wie in Abbildung C.1 zu erkennen, ist $a = 0,5$ und $b = 1,5$. Damit ist $\rho = (a+b)/2 = 1$ gegeben. Für Fälle, in denen dies nicht der Fall ist, kann auf die Skalierungsstrategie aus Abschnitt 5.3 zurückgegriffen werden. Somit gilt $\gamma_1 = 1 - b = -0,5$. Damit ist die Laurententwicklung der konformen Abbildung $\psi(w)$ für das Gebiet K in Abbildung C.1 bestimmt. Diese Abbildung $\psi(w)$ definiert nun mit (5.6) beziehungsweise mit (5.7) die Faberpolynome. Dies kann verdeutlicht werden, indem mithilfe (5.7) einige Terme $F_m(z)$ bestimmt werden. Für die Ellipse mit der Abbildung (C.3) ergibt sich

$$\begin{aligned}
F_0(z) &= 1 \\
F_1(z) &= z - \gamma_0 \\
F_2(z) &= z^2 - (\gamma_1 + \gamma_0)z + \gamma_1\gamma_0 - 2\gamma_1 \\
F_3(z) &= z^3 + (-2\gamma_0 - \gamma_1)z^2 + (\gamma_0^2 + 2\gamma_0\gamma_1 - 3\gamma_1)z + (-\gamma_1\gamma_0^2 + 3\gamma_1\gamma_0).
\end{aligned} \tag{C.4}$$

Wird nun ein anderes Gebiet K mit einer anderen Abbildung $\psi(w)$ verwendet, ergeben sich unterschiedliche Polynomterme $F_m(z)$. Ist K beispielsweise ein Kreis, so kann die Abbildung $\psi(w) = w + \gamma_0$ verwendet werden. Mit dieser können die Faberpolynome als

$$\begin{aligned}
F_0(z) &= 1 \\
F_1(z) &= z - \gamma_0 \\
F_2(z) &= z^2 - 2\gamma_0 z + \gamma_0^2 \\
F_3(z) &= z^3 - 3\gamma_0 z^2 + 3\gamma_0^2 z - \gamma_0^3
\end{aligned} \tag{C.5}$$

berechnet werden.

Diese Abbildung $\psi(w)$ soll nun in den Blick genommen werden. Abbildung C.2 zeigt den Kreis um den Nullpunkt in der komplexen w-Ebene. Dieser Kreis C_R lässt sich mit $|w| = R$ oder $w = R\exp(j\theta)$ beschreiben, wobei $\theta \in [0, 2\pi]$. In der Abbildung C.3 wird das Bild von C_R unter $z = \psi(w = R\exp(j\theta))$ und $R = 1$ dargestellt. Mit diesen Voraussetzungen lassen sich die Koeffizienten c_m der Entwicklung bestimmen. Hierzu muss im allgemeinen Fall (5.8) bestimmt

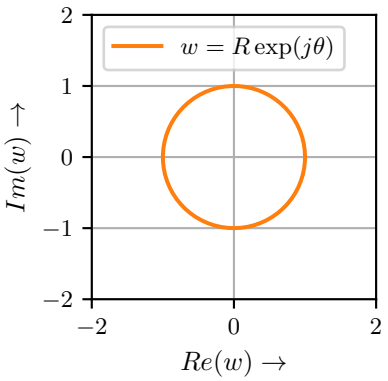


Abbildung C.2: Die Abbildung zeigt den Kreis C_R in der komplexen w-Ebene.

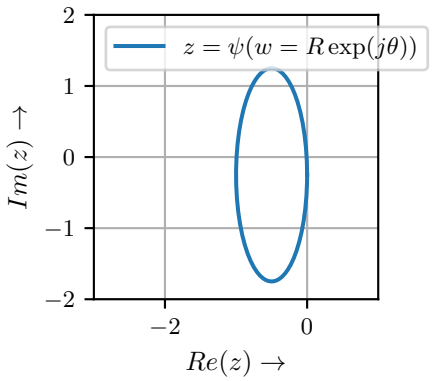


Abbildung C.3: Die Abbildung zeigt das Bild $z = \psi(w = w = R\exp(j\theta))$ des Kreises C_R in der komplexen z-Ebene für $R = 1$.

werden. Da es sich bei der Umrandung der Ellipse K um eine Jordan-Kurve handelt, kann in (5.8), wie in Abschnitt 5.2 beschrieben, $R = \rho = 1$ verwendet werden. Für einige Fälle kann bei der Berechnung auf eine analytische Bestimmungsformel für c_m zurückgegriffen werden. Dies ist wünschenswert, da ein zusätzlicher Rechenaufwand sowie mögliche zusätzliche numerische Fehler bei der Bestimmung der Koeffizienten vermieden werden. Die Koeffizienten c_m werden hier für verschiedene Werte q bestimmt. Der Absolutbetrag der Koeffizienten ist in Abbildung C.4 dargestellt.

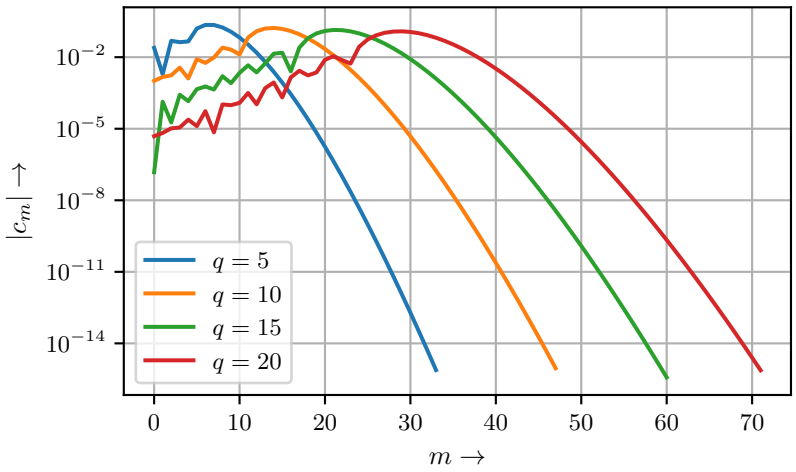
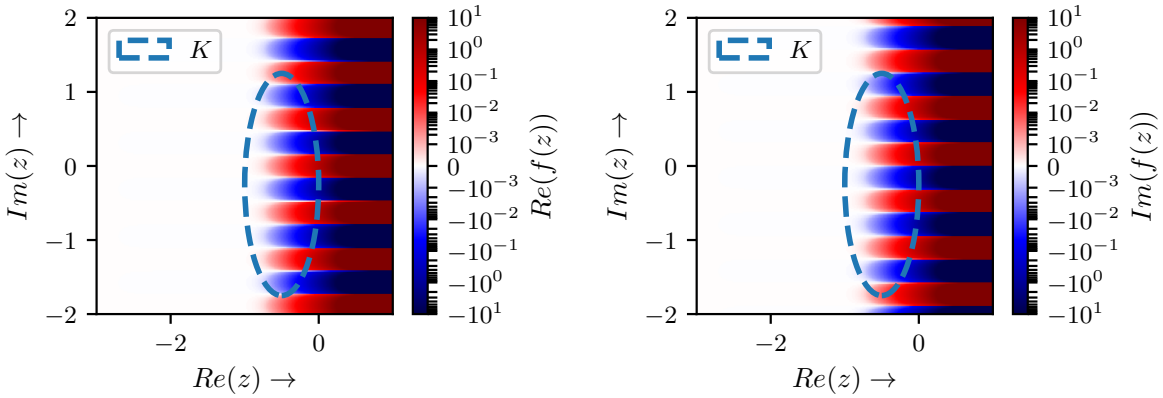


Abbildung C.4: Die Abbildung zeigt den Absolutbetrag $|c_m|$ für verschiedene q. Die Entwicklung wird ausgeführt, bis $|c_m| < 10^{-15}$ gilt.

Es ist zu beobachten, dass die Anzahl der Koeffizienten mit q ansteigt. Im Kontext der Untersuchungen im verbleibenden Teil dieser Arbeit entspricht ein größeres q einem größeren Zeitschritt

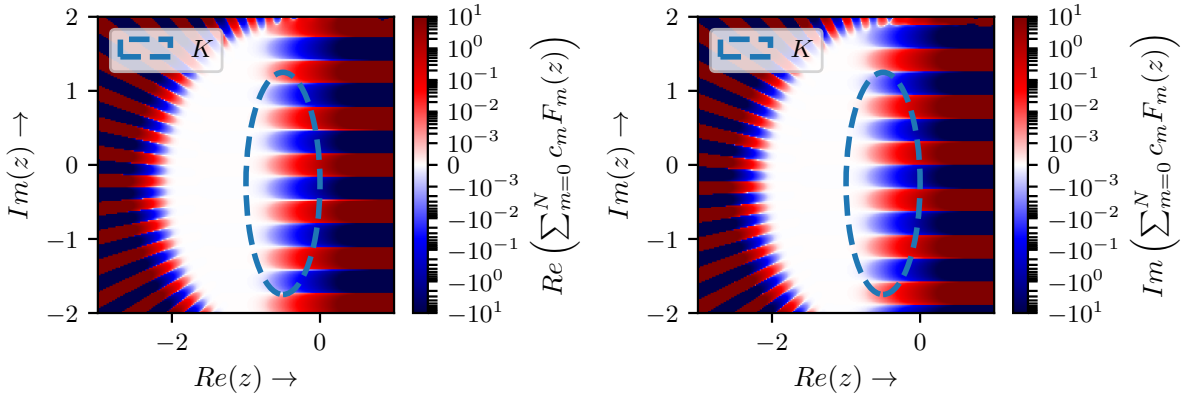
Δt. Die Approximation kann nun mit der Rekursionsbeziehung (5.12) bestimmt werden. Hier soll das Ergebnis in der komplexen z-Ebene evaluiert werden. In den Abbildungen C.5a und C.5b ist die Funktion $f(z)$ für $q = 10$ dargestellt. In den Abbildungen C.6a und C.6b ist die Faberpolynom-Approximation der Funktion $f(z)$ gegeben. Die Approximation $\sum_{m=0}^{N} c_m F_m(z)$



(a) Realteil der Funktion $f(z)$.

(b) Imaginärteil der Funktion $f(z)$.

Abbildung C.5: Die Abbildungen zeigen jeweils den Real- und Imaginärteil der Funktion $f(z)$ mit $q = 10$ mit einer logarithmischen Skalierung. Außerdem ist der Konvergenzbereich K der untersuchten Faberpolynom-Approximation in den Abbildungen markiert.



(a) Realteil der Approximation.

(b) Imaginärteil der Approximation.

Abbildung C.6: Die Abbildungen zeigen jeweils den Real- und Imaginärteil der Faberpolynom-Approximation $\sum_{m=0}^{N} c_m F_m(z)$ der Funktion $f(z)$ mit $q = 10$ mit einer logarithmischen Skalierung. Außerdem ist der Konvergenzbereich K in den Abbildungen markiert.

zeigt den erwarteten Verlauf in der komplexen Ebene. Innerhalb des Konvergenzgebietes K sind beim Vergleich der Abbildungen in C.5 und C.6 keine Abweichungen zu erkennen. Außerhalb von K, insbesondere im Bereich der negativen Halbebene, sind Abweichungen von der Funktion

$f(z)$ zu beobachten. Um die Genauigkeit der Approximation zu untersuchen, wird nun zusätzlich die Abweichung von $f(z)$ betrachtet. Diese ist in Abbildung C.7 dargestellt. Die Abweichung

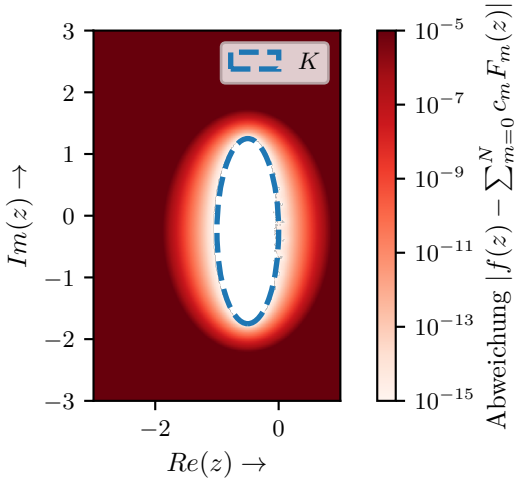


Abbildung C.7: Die Abbildung zeigt die Abweichung $|f(z)-\sum_{m=0}^{N} c_m F_m(z)|$ des Absolutbetrages der Approximation von $f(z)$ mit einer logarithmischen Skalierung sowie den Konvergenzbereich K.

erreicht innerhalb des Konvergenzgebietes K mit 10^{-15} erwartungsgemäß sehr niedrige Werte. Es fällt bei Berücksichtigung der logarithmischen Skalierung auf, dass die Abweichung außerhalb von K schnell ansteigt.

C.2 Bestimmung der Koeffizienten der W-Matrix für nichtlineare Probleme

Im Folgenden soll beispielhaft gezeigt werden, wie die Koeffizienten des Zeitpropagationsschemas für den in Kapitel 7 beschriebenen Algorithmus berechnet werden können. Dieser Algorithmus erlaubt eine Berechnung von Approximationen der Form

$$\vec{\Psi}(t_n + \Delta t) \approx \exp(\Delta t \mathcal{H})\vec{\Psi}(t_n) + \Delta t \sum_{k=1}^{p} \varphi_k(\Delta t \mathcal{H})\vec{w}_k \tag{C.6}$$

mithilfe der Berechnungsvorschrift

$$\vec{\Psi}(t_n + \Delta t) = \begin{bmatrix} I_N & 0 \end{bmatrix} \exp\left(\Delta t_s \begin{bmatrix} \mathcal{H}/\lambda_s & \mathcal{W}/\lambda_s \\ 0 & \mathcal{J}/\Delta t/\lambda_s \end{bmatrix} \right) \begin{bmatrix} \vec{\Psi}(t_n) \\ \vec{e}_p \end{bmatrix}. \tag{C.7}$$

Hierbei gilt $\mathcal{W} = [\vec{w}_1, \vec{w}_2, ..., \vec{w}_p]$.

C.2.1 Rosenbrock-Euler-Verfahren

Zunächst soll das Rosenbrock-Euler-Verfahren in den Blick genommen werden. Dieses ist, wie in Kapitel 7 beschrieben, mit

$$\vec{\Psi}(t_n + \Delta t) = \exp(\Delta t J_n)\vec{\Psi}(t_n) + \Delta t \varphi(\Delta t J_n) R_n(\vec{\Psi}(t_n)) \tag{C.8}$$

gegeben. Durch Vergleich mit (C.6) lassen sich die Koeffizienten direkt bestimmen:

$$\mathcal{W} = [\vec{w}_1 = R_n(\vec{\Psi}(t_n))]. \tag{C.9}$$

Damit lässt sich die Berechnungsvorschrift durch Einsetzten in (C.7) zu

$$\vec{\Psi}(t_n + \Delta t) = \begin{bmatrix} I_N & 0 \end{bmatrix} \exp\left(\Delta t_s \begin{bmatrix} J_n/\lambda_s & [R_n(\vec{\Psi}(t_n))]/\lambda_s \\ 0 & \mathcal{J}/\Delta t/\lambda_s \end{bmatrix} \right) \begin{bmatrix} \vec{\Psi}(t_n) \\ \vec{e}_1 \end{bmatrix} \tag{C.10}$$

bestimmen.

C.2.2 Rosenbrock-Verfahren: exprb32

Nun soll mit dem **exprb32**-Verfahren aus [104] ein komplexerer Fall betrachtet werden. Im Gegensatz zu dem Rosenbrock-Euler-Verfahren erfordert das **exprb32**-Verfahren die Berechnung von Zwischenschritten. Allgemein können diese mit

$$\begin{aligned} U_{n,i} &= \exp(c_i \Delta t J_n)\vec{\Psi}(t_n) + \Delta t \sum_{j=1}^{i-1} a_{i,j}(\Delta t J_n) R_n(U_{n,j}) \\ \vec{\Psi}(t_n + \Delta t) &= \exp(\Delta t J_n)\vec{\Psi}(t_n) + \Delta t \sum_{i=1}^{s} b_i(\Delta t J_n) R_n(U_{n,i}) \end{aligned} \tag{C.11}$$

notiert werden [104]. Die Koeffizienten der einzelnen Verfahren können kompakt mit dem sogenannten Butcher-Tableau notiert werden [104]:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1j} \\ c_2 & a_{21} & a_{22} & \dots & a_{2j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_i & a_{i1} & a_{i2} & \dots & a_{ij} \\ \hline & b_1 & b_2 & \dots & b_j \end{array} . \tag{C.12}$$

Das Butcher-Tableau des **exprb32**-Verfahren kann gemäß [104] mit

$$\begin{array}{c|cc} 0 & & \\ 1 & \varphi_1 & \\ \hline & \varphi_1 - 2\varphi_3 & 2\varphi_3 \end{array} \tag{C.13}$$

angeben werden. Mithilfe von (C.13) und (C.11) kann das Zeitpropagationsschema des Verfahrens mit

$$\begin{aligned}
U_{n,1} =& \vec{\Psi}(t_n) \\
U_{n,2} =& \exp(\Delta t J_n)\vec{\Psi}(t_n) + \Delta t \varphi_1(\Delta t J_n)(\Delta t J_n) R_n(U_{n,1}) \\
\vec{\Psi}(t_n + \Delta t) =& \exp(\Delta t J_n)\vec{\Psi}(t_n) + \Delta t((\varphi_1(\Delta t J_n) - 2\varphi_3(\Delta t J_n))R_n(U_{n,1}) \\
& + 2\varphi_3(\Delta t J_n) R_n(U_{n,2})) \\
\Leftrightarrow \vec{\Psi}(t_n + \Delta t) =& \exp(\Delta t J_n)\vec{\Psi}(t_n) + \Delta t(\varphi_1(\Delta t J_n) R_n(U_{n,1}) \\
& + \varphi_3(\Delta t J_n)(2R_n(U_{n,2}) - 2R_n(U_{n,1})))
\end{aligned} \tag{C.14}$$

bestimmt werden. Durch Vergleich mit (C.6) lassen sich die jeweiligen $\mathcal{W}$ der Berechnungsschritte zu $\mathcal{W} = [R_n(U_{n,1}]$ und $\mathcal{W} = [R_n(U_{n,1}), 2R_n(U_{n,2}) - 2R_n(U_{n,1})]$ ermitteln. Damit ist die gesamte Berechnungsvorschrift mit

$$\begin{aligned}
U_{n,1} &= \vec{\Psi}(t_n) \\
U_{n,2} &= \begin{bmatrix} I_N & 0 \end{bmatrix} \exp\left(\Delta t_s \begin{bmatrix} J_n/\lambda_s & [R_n(U_{n,1})]/\lambda_s \\ 0 & \mathcal{J}/\Delta t/\lambda_s \end{bmatrix} \right) \begin{bmatrix} \vec{\Psi}(t_n) \\ \vec{e}_1 \end{bmatrix} \\
\vec{\Psi}(t_n + \Delta t) &= \begin{bmatrix} I_N & 0 \end{bmatrix} \exp\left(\Delta t_s \begin{bmatrix} J_n/\lambda_s & [R_n(U_{n,1}), 2R_n(U_{n,2}) - 2R_n(U_{n,1})]/\lambda_s \\ 0 & \mathcal{J}/\Delta t/\lambda_s \end{bmatrix} \right) \begin{bmatrix} \vec{\Psi}(t_n) \\ \vec{e}_3 \end{bmatrix}
\end{aligned} \tag{C.15}$$

gegeben.