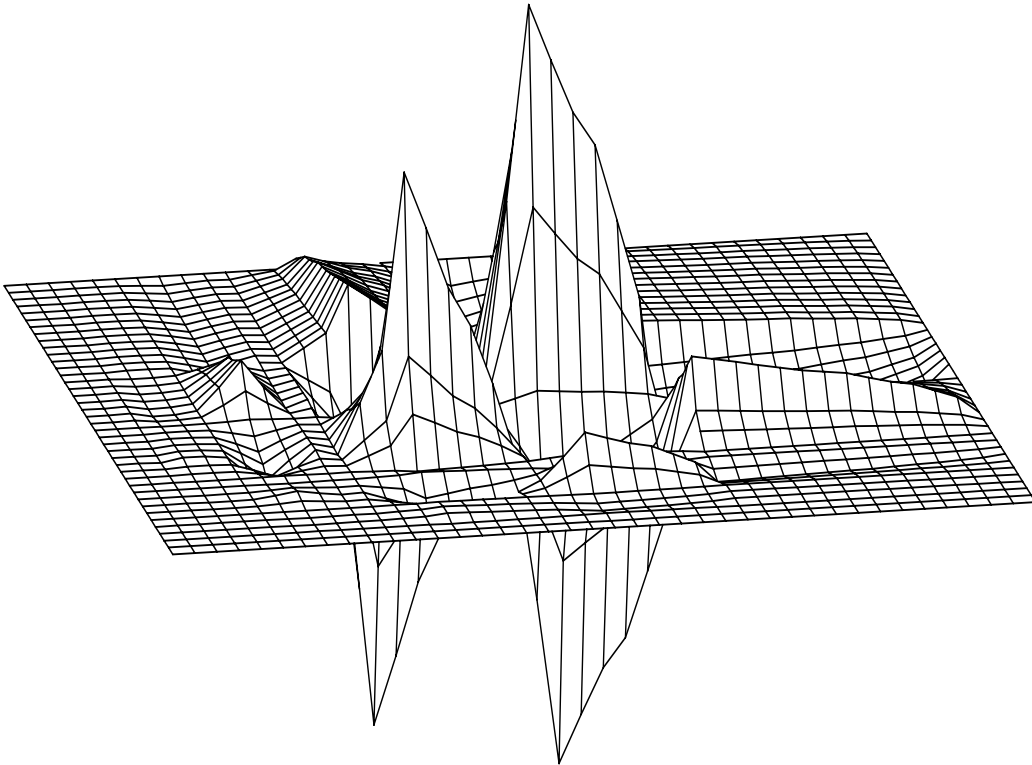


**Die FDTD im Waveletbereich**  
—  
**Verfahren zur Kompression  
linearer Operatoren mit Wavelets**



Marc Walter



Die FDTD im Waveletbereich  
—  
Verfahren zur Kompression  
linearer Operatoren mit Wavelets

Von der Fakultät für Ingenieurwissenschaften der  
Universität Duisburg-Essen

zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigte Dissertation

von

Dipl.-Ing. Marc Walter

aus

Neuss

Referent : Prof. Dr.-Ing. Ingo Wolff  
Korreferent: Prof. Dr.-Ing. Adalbert Beyer  
Tag der mündlichen Prüfung: 10. März 2004



# Inhaltsverzeichnis

Symbol- und Abkürzungsverzeichnis	vii
<b>1 Einleitung</b>	<b>1</b>
<b>2 Das FDTD-Verfahren</b>	<b>9</b>
2.1 Die Diskretisierung der Maxwell'schen Gleichungen . . . . .	10
2.2 Numerische Dispersion und Stabilität . . . . .	14
2.3 Randbedingungen . . . . .	16
2.3.1 Ideal leitende Grenzschichten . . . . .	16
2.3.2 Absorbierende Randbedingungen . . . . .	17
<b>3 Wavelets</b>	<b>21</b>
3.1 Die Fouriertransformation . . . . .	21
3.2 Grundlagen zur Wavelet-Theorie . . . . .	23
3.2.1 Die kontinuierliche Wavelet-Transformation . . . . .	23
3.2.2 Die Multi-Skalen-Analyse . . . . .	25
3.2.3 Skalierungsfunktionen und Wavelets . . . . .	28
3.2.4 Die schnelle Wavelet-Transformation . . . . .	31
3.2.5 Diskrete periodische Wavelet-Transformation . . . . .	32

<b>4</b>	<b>Mehrdimensionale Basen und Skalen</b>	<b>39</b>
4.1	Homogene Tensorprodukt-Basen . . . . .	39
4.2	Separable MSA-Basen . . . . .	41
4.3	Allgemeine MSA-Basen . . . . .	45
<b>5</b>	<b>Wavelet-Transformation von Operatoren</b>	<b>47</b>
5.1	Grundlagen . . . . .	47
5.1.1	Diskrete Darstellung von Operatoren . . . . .	48
5.1.2	Transformation von Matrix-Operatoren . . . . .	49
5.1.3	Eindimensionale WT von Operatoren . . . . .	49
5.1.4	Notationen . . . . .	51
5.2	Matrix-Darstellung von Operatoren . . . . .	55
5.2.1	Klassifizierung von WT-Operatoren . . . . .	56
5.2.2	Operatortypen in der WT . . . . .	58
5.2.3	Algorithmen zur Matrix-basierten WT von Operatoren . . . . .	62
5.2.3.1	$U_{\mathbf{0}}^{\mu}$ -Operatoren im Waveletbereich . . . . .	62
5.2.3.2	$U_{\mathbf{1}_{\mu}}^{\mu}$ -Operatoren . . . . .	64
5.2.3.3	Weitere Operatoren . . . . .	69
5.2.3.4	Vorgehensweise bei MSA-Basen . . . . .	71
5.3	Direkte Verkettung von Operatoren . . . . .	73
5.3.1	Elementare Abbildungen . . . . .	74
5.3.2	Aufbau einer $O^3$ -Liste . . . . .	78
5.3.2.1	Vergleich von $O^2$ -Listen . . . . .	78
5.3.2.2	Sortierung von $O^3$ -Listen . . . . .	79
5.3.3	Durchführung einer Verkettung . . . . .	82
5.3.4	Operatoren und Verkettungen bei Vektorfeldern . . . . .	83

---

<b>6 Operatoren im Simulationsprogramm</b>	<b>85</b>
6.1 Speicherarten und Zugriffszeiten . . . . .	85
6.2 Berechnungen mit Operatormatrizen . . . . .	88
6.3 Listen-basierte Berechnungen . . . . .	88
<b>7 Die FDTD im Waveletbereich</b>	<b>93</b>
7.1 Vorbetrachtungen . . . . .	93
7.2 Matrixoperatoren der FDTD in der WT . . . . .	97
7.2.1 Differenziationen . . . . .	97
7.2.2 Randbedingungen . . . . .	98
7.2.3 Materialparameter . . . . .	100
7.2.4 Feldanregung und -ausgabe . . . . .	101
7.2.5 Adaptive Berechnungen . . . . .	102
7.2.5.1 Räumliche Adaptivität . . . . .	102
7.2.5.2 Zeitliche Adaptivität . . . . .	104
7.3 Listen-basierte WT der FDTD . . . . .	105
7.3.1 Operatoren der klassischen FDTD . . . . .	106
7.3.2 Automatische Bereichsanpassung . . . . .	109
7.3.3 Absorbierende Randbedingungen . . . . .	111
7.3.4 Optimierte Feldanregung und -ausgabe . . . . .	113
7.3.5 Teilbereichs-WT . . . . .	116
7.3.6 Das Detailgitter-Verfahren . . . . .	120
7.3.6.1 Motivation . . . . .	120
7.3.6.2 Operatoren und Algorithmus . . . . .	121
7.3.6.3 Stabilitätsbetrachtungen . . . . .	123
7.3.7 Lokale Nichtlinearitäten . . . . .	140

<b>8</b>	<b>Simulationsergebnisse</b>	<b>143</b>
8.1	Simulationszeiten bei der Matrix-basierten FDTD . . . . .	144
8.2	Simulation mit der Listen-basierten FDTD . . . . .	148
8.2.1	Hohlleiter-Diskontinuität . . . . .	148
8.2.1.1	Einflüsse des Detailgitter-Verfahrens . . . . .	151
8.2.1.2	Operatorimplementierungen und Simulationszeiten . . . . .	156
8.2.1.3	Skalen und Wavelet-Ordnungen . . . . .	159
8.2.1.4	Abschließender Vergleich . . . . .	162
8.2.2	Mikrostreifenleitungsfilter . . . . .	164
8.2.3	Analysen . . . . .	168
8.2.3.1	Allgemeine Betrachtungen . . . . .	168
8.2.3.2	Vektor- und Faktorfolgen von $O^3$ -Listen . . . . .	170
8.2.3.3	Mehrstufige Transformationen . . . . .	176
8.2.3.4	Mehrdimensionale Felder . . . . .	178
8.2.3.5	Weitere Ergebnisse . . . . .	182
<b>9</b>	<b>Ausblicke und Weiterentwicklungen</b>	<b>185</b>
9.1	MRTD – Multiresolution time-domain . . . . .	186
9.1.1	MRTD mit Battle-Lemarié-Funktionen . . . . .	187
9.1.2	Daubechies-Funktionen . . . . .	195
9.1.3	Haar-Funktionen . . . . .	196
9.1.4	CDF-Funktionen . . . . .	202
9.1.5	Fazit . . . . .	204
9.2	Erweiterungsmöglichkeiten . . . . .	205
9.2.1	Entwicklungen in der Wavelet-Theorie . . . . .	206
9.2.2	Konzeptionelle Modifikationen . . . . .	210

---

<b>10 Zusammenfassung</b>	<b>215</b>
<b>A Ergänzende mathematische Sätze und Beweise</b>	<b>219</b>
A.1 Spiegelung von Wavelets und Koeffizienten . . . . .	219
A.2 Indexverschiebung von Koeffizienten . . . . .	221
<b>B Software-Implementierungen</b>	<b>223</b>
B.1 Sortierverfahren . . . . .	223
B.2 Faktorvergleich mit Toleranzschwelle . . . . .	224
<b>Literaturverzeichnis</b>	<b>227</b>



# Symbol- und Abkürzungsverzeichnis

## Schriftstile

$a$	Skalar; Funktion
$a$	Komponente
$A$	Skalar; Menge; diskrete Abbildung zu $\mathcal{A}$
$\mathbf{a}$	Vektor
$\alpha$	Matrix, Tensor
$\mathbf{A}$	Vektor; Matrix; Raum
$\Theta$	Raum
$\mathcal{A}$	Operator

## Notationen

$0, \mathbf{0}$	Nullelement (z. B. Zahl Null, Nullvektor, Null-Operator)
$\hat{a}$	Fouriertransformierte von $a$
$\tilde{a}$	Wavelet-Transformierte von $a$
$\bar{a}$	Abschluss von $a$
$\mathbf{A}^T$	transponierte Matrix von $\mathbf{A}$
$\mathcal{A}^T, \mathbf{X}^T$	dualer Operator bzw. dualer Raum zu $\mathcal{A}$ bzw. $\mathbf{X}$
$\mathcal{A}^\dagger$	adjungierter Operator zu $\mathcal{A}$
$\mathbf{A}^{-1}$	inverse Matrix von $\mathbf{A}$
$\mathbf{A}^*$	Matrix zu $\mathcal{A}^*$
$\mathcal{A}^{-1}, \mathcal{A}^*$	Umkehroperation zu $\mathcal{A}$
$\bar{\mathcal{A}}$	komplementäre Operation zu $\mathcal{A}$
$\Delta a$	Diskretisierungsschrittweite für $a$
$\Delta \mathbf{a}$	Verschiebungsvektor

---

$\Delta \mathbf{A}$	Differenzraum
$\mathbf{a}_{\text{tan}}$	tangentialer Anteil von $\mathbf{a}$
$\ a\ _{\mathbf{L}_2}$	$\mathbf{L}_2$ -Norm von $a$
$(a_k)_{k \in \mathbb{Z}}$	Folge aller $a_k$
$\lceil a \rceil$	obere Gauß-Klammer, Aufrundung von $a$
$\lfloor a \rfloor$	untere Gauß-Klammer, Abrundung von $a$
$\langle a, b \rangle_{\mathbf{L}_2}$	$\mathbf{L}_2$ -Skalarprodukt von $a$ und $b$
$a \bmod b$	Modulo-Operation, ganzzahliger nicht-negativer Rest der Division $\frac{a}{b}$
$\mathbf{AB}$	Produkt der Matrizen $\mathbf{A}$ und $\mathbf{B}$ ; kartesisches Produkt der Räume $\mathbf{A}$ und $\mathbf{B}$ (auch $\mathbf{A} \times \mathbf{B}$ )
$\sum_{\mathbf{k}}$	$\sum_{k_1} \cdots \sum_{k_n}$
$\leftarrow$	Komponente ersetzen (s. Abschnitt 5.1.4)
$\downarrow$	Komponente entfernen (s. Abschnitt 5.1.4)
$\circ, \bigcirc$	Verkettung / Verknüpfung
$\oplus$	direkte Summe
$\emptyset$	leere Menge
$\partial$	partiell Differential
$\wedge$	und
$\forall$	für alle
$\exists$	es existiert ein
$D(\mathcal{A})$	Definitionsbereich von $\mathcal{A}$
diag	Diagonalmatrix
div	Divergenz
id	Identität
ld	$\log_2$
rot	Rotation
span	lineare Hülle
sup	Supremum
supp	Träger
$\mathbb{R}_+$	Menge der positiven reellen Zahlen
$\mathbf{L}_1(\mathbb{R})$	Raum der absolut integrierbaren Funktionen
$\mathbf{L}_2(\mathbb{R})$	Raum der quadratisch integrierbaren Funktionen
$\mathbf{l}_2(\mathbb{Z})$	Raum der quadratisch summierbaren Funktionen
$\mathbf{L}(\mathbf{X}, \mathbf{Y})$	Menge der Abbildungen $\mathbf{X} \mapsto \mathbf{Y}$ (s. Abschnitt 7.3.6.3)

## Lateinische Formelzeichen

$a$	Dilatationsparameter; Index; Breite; reeller Wert
$\mathbf{a}, a_{\mathbf{k}}, a_{\nu}^{\mathbf{v}}$	Entwicklungskoeffizienten
$A$	positive Konstante; Anzahl Koeffizienten
$A_{\mathcal{T}}$	maximale Norm von $\mathcal{T}^n$ über alle $n$
$\mathbf{A}$	Dilatationsmatrix
$\mathcal{A}$	linearer Operator
$b$	Translationsparameter; Index; Höhe; reeller Wert
$\mathbf{b}, b_{\mathbf{k}}$	Entwicklungskoeffizienten
$B$	positive Konstante
$\mathbf{B}$	magnetische Flussdichte
$\mathcal{B}$	linearer Operator
$c$	Lichtgeschwindigkeit; Koeffizient; Index
$c_k^m, \mathbf{c}^m$	Entwicklungskoeffizienten für $\mathbf{V}_m$
$c_{\psi}$	Wavelet-Konstante
$C$	Menge von Entwicklungskoeffizienten
$\mathbf{C}, \mathcal{C}$	linearer Operator; zusammengefasster FDTD-Operator
$\mathbf{D}$	elektrische Flussdichte
$d$	Anzahl der Komponenten eines Vektorfeldes; Dicke (Anzahl von Yee-Zellen) eines Detailbereiches
$d_k^m, \mathbf{d}^m$	Entwicklungskoeffizienten für $\mathbf{W}_m$
$\mathcal{D}$	diskretisierter Rotationsoperator
$\mathbf{e}_i$	Einheitsvektor in Richtung $i$
$E, \mathbf{E}$	elektrische Feldstärke
$\mathcal{E}$	linearer Operator; Dualitätsabbildung
$f, \mathbf{f}$	Funktion
$F$	Menge von Entwicklungskoeffizienten
$F, \mathbf{F}$	Feldkomponente, -vektor; zusammengefasstes Feld
$\mathcal{F}$	Fouriertransformation
$g, \mathbf{g}$	Funktion
$G$	Menge von Entwicklungskoeffizienten
$g_k$	Waveletkoeffizient
$h$	Schrittweite
$h_k$	Skalierungskoeffizient
$H, \mathbf{H}$	magnetische Feldstärke
$i$	Index; Richtung; Positionsindex (meist für x-Richtung); Stromstärke
$I_t$	Anzahl der $O^1$ -Elemente in einer $O^2$ -Liste

$I_t^2$	Anzahl der $O^2$ -Listen in einer $O^3$ -Liste
$\mathbf{I}, \mathbf{I}^a$	Einheitsmatrix (der Größe $a \times a$ )
$j$	Index; Positionsindex (y-Richtung)
$j, \mathbf{j}$	Skala
$j$	imaginäre Einheit
$\mathbf{J}$	elektrische Stromdichte
$k$	Index, Positionsindex (meist für z-Richtung)
$\mathbf{k}$	mehrdimensionaler Index; Koordinatenvektor im $\mathbb{Z}^n$
$K_\nu$	obere Grenze für $k_\nu$
$\mathbf{K}$	Raum von $\mathbf{k}$ -Vektoren
$l$	Index; Translation
$l_\nu$	Ausdehnung des Feldraumes in Richtung $\nu$
$\mathbf{l}_\nu$	reduzierter $\mathbf{k}$ -Vektor (s. Abschnitt 5.1.4)
$\mathbf{L}_\nu^{\mathbf{k}}$	Raum von $\mathbf{l}_\nu$ -Vektoren
$L$	Anzahl (Elemente, Stützstellen in einer Skala); Matrixgröße; Dimensionszahl
$\bar{L}$	Anzahl bereits transformierter Koeffizienten
$\mathcal{L}$	Projektionsoperator
$m$	Grad; Skala; Matrixelement; Referenznummer; Index
$m, \mathbf{m}$	Verschiebungswert
$m_{\cdot, \cdot}$	diskretes Element (aus Matrix, Tensor) zu $\mathcal{M}$
$M$	Anzahl Transformationsschritte
$\mathbf{M}, \mathcal{M}$	lineare Operation (Matrix, Operator); Maskierungsoperator
$\mathcal{M}_G, \mathcal{M}_G^{\bar{\mathbf{F}}}$	Maskierungsoperator beim Detailgitter-Verfahren
$n$	Index; Zeitindex; $2^n$ =Anzahl von Stützstellen; Dimensionszahl
$N$	Ordnung von Daubechies-Wavelets; Anzahl Zeitschritte
$o_{\cdot, \cdot}$	diskretes Element (aus Matrix, Tensor) zu $\mathcal{O}$
$\mathcal{O}$	Skalierungsoperator
$O^1$	Gruppierung aus Faktor und Verschiebungsvektor (äquivalente Darstellung eines Tensorelementes)
$O^2$	Liste aus $O^1$ -Elementen (Berechnung eines Bildpunktes)
$O^3$	Liste aus $O^2$ -Listen (äquivalente Darstellung eines Operators)
$O$	Ordnung (Landau-Symbol)
$\mathbf{P}, \mathcal{P}, P$	Transformation zu höheren Skalen ( $\mathbf{V}$ - und $\mathbf{W}$ -Räume)
$p_{\cdot, \cdot}, p_{\cdot, \cdot}^*$	diskretes Element (aus Matrix, Tensor) zu $\mathcal{P}, \mathcal{P}^*$

$Q$	Wavelet-Transformation, Abbildung auf $\mathbf{W}$ -Raum
$Q^*$	inverse Wavelet-Transformation
$\mathcal{R}$	Abbildung auf $\mathbf{V}$ -Raum
$r$	Kompressionsrate
$r_l$	Rechteckfunktion
$R_{\mathcal{T}}$	Resolventenmenge von $\mathcal{T}$
$r_{\mathcal{T}}, r_{\mathcal{C}_1}$	Spektralradius von $\mathcal{T}, \mathcal{C}_1$
$s$	Abtastwert; Streuparameter
$s_{\cdot, \cdot}$	diskretes Element (aus Matrix, Tensor) zu $\mathcal{S}$
$\mathcal{S}$	Verschiebungsoperator; Vertauschungsoperator
$S_{\mathcal{T}}$	Spektrum von $\mathcal{T}$
$S_1, S_2$	Simulationszeiten
$t$	Zeit; Parameter; maximal mögliche Transformationstiefe
$t_{\cdot}, t_{\cdot, \cdot}$	diskretes Element (aus Matrix, Tensor) zu $\mathcal{T}$
$\mathbf{T}, \mathcal{T}, T$	lineare Operation (Matrix, Operator, diskrete Abbildung)
$u$	Spannung
$U^0$	Menge der Skalierungsoperatoren (s. Abschnitt 5.1.4)
$U^\nu$	Menge der $\nu$ -parallelen Operatoren (s. Abschnitt 5.1.4)
$\mathbf{v}_\nu$	Eigenvektor
$\mathbf{v}^j$	Koeffizientenvektor nach $j$ Transformationsstufen
$\mathbf{V}_j$	Raum der Skalierungsfunktionen auf Skala $j$
$V$	Multiplikator für Zeitschrittweite
$\mathbf{W}_j$	Raum der Wavelets auf Skala $j$
$x, \mathbf{x}$	Koordinatenwert; Richtung, Komponente
$\mathbf{x}$	Koordinatenvektor im $\mathbb{R}^n$
$\mathbf{X}$	linearer Raum
$y, \mathbf{y}$	Koordinatenwert; Richtung, Komponente
$\mathbf{Y}$	linearer Raum
$z, \mathbf{z}$	Koordinatenwert; Richtung, Komponente

### Griechische Formelzeichen

$\alpha$	Zwischenwert
$\boldsymbol{\alpha}$	Operatormatrix
$\beta$	Phasenkonstante
$\boldsymbol{\beta}$	Operatormatrix
$\boldsymbol{\gamma}$	Operatormatrix
$\delta$	Toleranzschwelle

$\delta_{,}$	Kronecker-Symbol
$\delta$	Operatormatrix
$\varepsilon$	kleine Größe ( $0 < \varepsilon \ll 1$ ), Toleranzschwelle
$\varepsilon, \boldsymbol{\varepsilon}$	Permittivität
$\eta$	Index; Richtung
$\theta_{\mathbf{k}}$	skalare Funktion auf dem $\mathbb{R}^n$
$\Theta$	Funktionsraum der $\theta_{\mathbf{k}}$
$\kappa, \boldsymbol{\kappa}$	elektrische Leitfähigkeit
$\lambda_{\mathbf{B}}(\mathcal{A})$	Bildraum von $\mathcal{A}$ (auf Koordinaten bezogen)
$\lambda_{\mathbf{U}}(\mathcal{A})$	Urbildraum von $\mathcal{A}$ (auf Koordinaten bezogen)
$\lambda(\mathcal{B}, \mathcal{A})$	$\lambda_{\mathbf{U}}(\mathcal{B}) \cap \lambda_{\mathbf{B}}(\mathcal{A})$
$\lambda_{\mathbf{G}}$	Bildraum von $\mathcal{M}_{\mathbf{G}}$
$\lambda_{\mathbf{V}_j}$	Koordinatenraum der Koeffizienten von $\mathbf{V}_j$
$\lambda_{\nu}$	Eigenwert
$\mu$	Skala; Richtung; Index
$\mu, \boldsymbol{\mu}$	Permeabilität
$\nu$	Index; Richtung
$\xi_k^i$	eindimensionale Funktion aus $\Xi^i$
$\Xi^i$	Funktionsraum der $\xi_k^i$
$\rho$	elektrische Raumladungsdichte
$\sigma, \boldsymbol{\sigma}$	magnetische Leitfähigkeit
$\tau$	Verschiebungswert
$\varphi$	Skalierungsfunktion
$\psi$	Wavelet
$\omega$	Kreisfrequenz; Parameter

## Abkürzungen

ABC	Absorbing Boundary Condition
ALU	Arithmetical Logical Unit
A-UPML	Anisotropic Uniaxial Perfectly Matched Layer
CDF	Cohen-Daubechies-Feauveau
DPWT	Diskrete periodische WT
FD	Finite Differenz
FDM	Finite-Differenzen-Methode
FDTD	Finite Difference Time Domain
FEM	Finite-Elemente-Methode
FFT	Fast Fourier Transform

---

FPU	Floating Point Unit
FT	Fouriertransformation
FWT	Fast Wavelet Transform (schnelle WT)
IWT	Inverse Wavelet-Transformation
MRTD	Multiresolution time-domain
MSA	Multi-Skalen-Analyse
ONB	Orthonormalbasis
PML	Perfectly Matched Layer
SATS	Scale Adaptive Time Steps
TLM	Transmission-Line-Matrix-Methode
WT	Wavelet-Transformation



# Kapitel 1

## Einleitung

Die computergestützte Simulation elektromagnetischer Wellen findet in vielen industriellen Bereichen Anwendung. Beispielhaft seien hier die Entwicklung von Antennen, die Vorhersage der Ausbreitung von Funkwellen und das Verhalten elektronischer Bauelemente und Schaltungen genannt. Oftmals sind derartige Simulationen jedoch sehr zeitaufwendig. Es sollen hier neue Möglichkeiten vorgestellt werden, wie eines der am häufigsten eingesetzten Verfahren, die *Finite Difference Time Domain* Methode (FDTD), mit der Hilfe sog. *Wavelets* beschleunigt werden kann. Die Wavelet-Theorie hat in den vergangenen Jahren einen regelrechten Boom erlebt und u. a. zu der Entwicklung sehr effizienter Bildkompressionstechniken geführt.

Simulationsverfahren und Bildkompressionstechniken sind Themengebiete, die scheinbar nichts miteinander zu tun haben. Simulationen helfen in vielen Gebieten, komplizierte physikalische und andere Abläufe auf einem Computer nachzubilden. Moderne Bildkompressionstechniken dagegen erlauben es, die Datenmenge von Bildern und Filmen zu reduzieren, sodass sie weniger Platz auf einem Computer in Anspruch nehmen oder sich schneller übertragen lassen.

Die Motivation für die vorliegende Arbeit lässt sich nun auf eine kurze Gedankenkette reduzieren: Je mehr Daten ein Simulationsverfahren verarbeitet, desto länger dauert die Berechnung. Kompressionstechniken können Datenmengen reduzieren. Kann eine Simulation durch die Verwendung komprimierter Daten beschleunigt werden? Anders formuliert:

Kann ein Simulationsverfahren komprimiert werden?

Es lässt sich an dieser Stelle bereits vorwegnehmen, dass dies tatsächlich möglich ist. Die in den folgenden Kapiteln entwickelten Methoden lassen sich nicht nur auf die FDTD, sondern auch auf einige andere Simulationsverfahren anwenden. Dennoch darf der Aufwand, der zur Beschleunigung der FDTD aufgebracht werden muss, nicht unterschätzt werden. Bei jedem Simulationsverfahren ist stets abzuschätzen, ob die zu untersuchende Struktur damit überhaupt gelöst werden kann und ob sich der mit dem jeweiligen Verfahren verbundene Aufwand lohnt. Daher soll zunächst auf die Relevanz elektromagnetischer Simulationen und die typischen Probleme beim Einsatz der Berechnungsverfahren eingegangen werden. In diesem Zusammenhang wird dann auch die Verbindung zu Bildkompressionstechniken klarer.

Betrachten wir als erstes Beispiel elektronische Schaltungen. Häufig werden günstigere elektronische Bauteile an der Grenze ihres Einsatzbereiches betrieben, um den Einsatz von teureren Bauteilen zu vermeiden. Vielfach wird sogar versucht, einzelne Bauelemente durch den Einsatz neuer Schaltungskonzepte ganz wegzurationalisieren. Die Funktionalität solcher Schaltungen lässt sich durch die Herstellung von Probeaufbauten validieren. Erweisen sich diese als unbrauchbar, sind in einem iterativen Prozess Änderungen auszuprobieren.

Weitere Gründe, die Versuche manchmal unumgänglich machen, sind vielfältig: Elektromagnetische Wellen, deren Wellenlängen in der Größenordnung der Schaltungen und Strukturen liegen; Bauelemente, die aus Platzgründen so dicht bei einander liegen, dass sie miteinander koppeln; Leiteranordnungen, die unerwünscht Wellen abstrahlen; oder ganz einfach die bewusste Benutzung von Strukturen, von denen man das Verhalten lediglich abschätzen, keinesfalls jedoch mit einer „Faustformel“ genau berechnen kann – viele Antennen, Filter und Kopplungen fallen unter diese Kategorie.

Simulationen erweisen sich auf Grund der beständig steigenden Leistungsfähigkeit von Computern immer häufiger als probates Mittel, die Anzahl von Versuchsaufbauten zu reduzieren und so teilweise beträchtliche Kosten einzusparen. Dennoch genügt es nicht, sich allein auf steigende „Rechenpower“ zu verlassen – die Simulationsverfahren an sich müssen verbessert werden. Dies wird auch von Seiten der Industrie auf Konferenzen immer wieder betont [1].

Die Notwendigkeit zur Weiterentwicklung hat im Laufe der Zeit eine beträchtliche Anzahl unterschiedlichster Verfahren hervorgebracht. Dabei gehören Momentenmethoden, Finite-Elemente-Methoden (FEM), Finite-Differenzen-Methoden (FDM) und die Transmission-Line-Matrix-Methoden (TLM) wohl zu den am weitesten verbreiteten Verfahren, die derzeit bei der Simulation elektromagnetischer Felder und Wellen Verwendung finden.

Auf der Suche nach einem Weg, noch schnellere Methoden zu erhalten, ergeben sich im Wesentlichen drei Auswahlmöglichkeiten. Zunächst wäre der Einsatz neuer mathematischer Ansätze zu nennen, um damit aus den Maxwell'schen Gleichungen, die das Verhalten elektromagnetischer Felder beschreiben, einen neuen Algorithmus zu gewinnen. Als zweites wäre die Kombination bestehender Verfahren zu sog. *Hybriden Verfahren* zu nennen. Und schließlich besteht natürlich noch die Möglichkeit, bereits etablierte Verfahren zu erweitern und zu modifizieren.

Bei der hier vorliegenden Arbeit werden im Prinzip alle drei genannten Möglichkeiten genutzt. Dazu soll kurz auf eine Gemeinsamkeit der oben genannten FEM, FDM und TLM eingegangen werden. Bei allen drei Verfahren muss der Raum, in dem die Felder zu simulieren sind, in diskrete Elemente bzw. Punkte eingeteilt werden. Die Genauigkeit des simulierten Feldes im Vergleich zum physikalischen Feld lässt sich somit über die räumliche Auflösung, d. h. die Dichte der Punkte einstellen.

Reicht eine bisher erreichte Genauigkeit nicht aus, lässt sich dies oft durch Halbierung des Punktabstandes korrigieren – mit der Konsequenz, dass sich bei einer dreidimensionalen Struktur die Anzahl aller Punkte schlagartig verachtfacht. Um der einhergehenden Erhöhung der Rechenzeit entgegenzuwirken, wurden Mehrgitter- und Untergitterverfahren für bestehende Methoden entwickelt. Diese erweisen sich häufig als sinnvoll, wenn die Erhöhung der Auflösung in lediglich kleinen Teilbereichen des Simulationsgebietes ausreicht, damit die Genauigkeit auf das gewünschte Maß steigt. Hierzu werden nur in diesen Bereichen zusätzliche Punkte eingeführt, während der Rest des Raumes seine gröbere Strukturierung beibehält. Als anschauliche Analogie sei hier eine Bildkompressionstechnik genannt: Flächen (zusammenhängende Bildausschnitte) mit konstanter Farbe und Helligkeit enthalten keine variierenden Bildinformationen, sodass sie sich grob auflösen lassen. In der Nähe von Kanten (Helligkeits- und Farbsprüngen) dagegen ist eine hohe räumliche Auflösung vonnöten,

wenn die Kante nicht unscharf wiedergegeben werden soll. Oft erweisen sich die um Untergitter erweiterten Verfahren jedoch als instabil, d. h. sie liefern häufig keine verwertbaren Resultate.

Die Idee der Bildkompression lässt sich aber weiter ausbauen: Man stelle sich einfach das elektromagnetische Feld als Bild vor und benutze ein modernes Bildkompressionsverfahren, um eine deutlich geringere Datenmenge ohne nennenswerten Qualitätsverlust zu erhalten. Wenn dann das Simulationsverfahren direkt mit derart komprimierten Daten arbeiten kann, so könnte der Rechenaufwand insgesamt betrachtet ebenfalls deutlich sinken.

Dass sich elektromagnetische Felder bildhaft visualisieren lassen, wird an Hochschulen bereits in Grundlagenfächern demonstriert. Es stellen sich somit nur noch zwei Fragen: Welche Kompressionsverfahren können angewendet werden und wie lassen sich Kompressions- und Simulationsverfahren zu einem einzigen Verfahren miteinander verschmelzen? Bei der Beantwortung der ersten Frage treten schnell die eingangs genannten Wavelets auf den Plan. Diese mathematischen Funktionen ermöglichen die Entwicklung von Algorithmen, die bei gleicher Qualität deutlich höhere Kompressionsraten erlauben als andere verbreitete Verfahren wie beispielsweise JPEG. Es ist derzeit absehbar, dass sich Wavelet-basierte Bildkompressionsverfahren als Standardverfahren bei der Übertragung und Speicherung von Bildern etablieren werden.

Auf die Frage nach der „Verschmelzung“ von Kompression und Simulation zu einer neuartigen Simulation wird in dieser Arbeit als Antwort häufig die *Verkettung von Operatoren* genannt werden. Dieser mathematisch wohldefinierte Begriff führt erfahrungsgemäß leicht zu Irritationen, wenn es um Überlegungen zur konkreten Implementierung in einem Computerprogramm bzw. Algorithmus geht. Ohne an dieser Stelle näher darauf eingehen zu wollen, sollen dennoch die Voraussetzungen für die hier untersuchte Vorgehensweise und die sich daraus ergebenden Vorteile genannt werden.

Zur Untersuchung der Idee auf ihre Umsetzbarkeit wurde als Basisverfahren die FDTD gewählt. Sie wurde bereits 1966 entwickelt [2], ist sehr weit verbreitet, einfach zu programmieren und gehört zu den Methoden mit dem größten Einsatzbereich bei elektromagnetischen Problemstellungen. Die FDTD wurde ursprünglich zur Simulation von Gebieten mit linearen Materialeigenschaften entwickelt. Obwohl später Erweiterungen zur

---

Behandlung nichtlinearer Medien publiziert wurden, ist man bei der Verbindung von FDTD und Wavelets im Wesentlichen auf lineare Eigenschaften angewiesen.<sup>1</sup> Weiterhin ist die FDTD zwar ein iteratives Verfahren, dennoch sind die Gleichungen zur Berechnung des Feldes für den jeweils nächsten Zeitschritt *explizit*. Somit entfällt im Gegensatz zu *impliziten* Verfahren das Lösen von Gleichungssystemen.

Damit wurden auch schon die meisten Voraussetzungen genannt: Das Simulationsverfahren muss sich als linearer Operator auf einem diskreten Raum darstellen lassen. Diese Voraussetzungen treffen auf sehr viele Simulationsverfahren zu, sodass der hier vorgestellte Ansatz einen weiten Einsatzbereich finden dürfte.

Die zur Kompression erforderliche *Wavelet-Transformation* (WT) erfüllt ebenso diese Voraussetzungen.<sup>2</sup> Sind die Operatoren der FDTD und der WT einmal im Computer implementiert, berechnet dieser den neuen Operator automatisch durch die bereits erwähnte Verkettung. Auch wenn an dieser Stelle noch einige Details unerwähnt bleiben müssen, wird ein fundamentaler Unterschied zur konventionellen Entwicklung neuer Simulationsverfahren offensichtlich: Die „händische“ Herleitung von Operatoren entfällt! Man ist auch sehr viel flexibler bei der Erweiterung des ursprünglichen Simulationsverfahrens durch beispielsweise verschiedene *absorbierende Randbedingungen* (engl. Absorbing Boundary Condition, ABC; vgl. Abschnitt 2.3.2). Es genügt i. A. nicht, FDTD und ABC getrennt voneinander zu transformieren und dann miteinander zu verknüpfen. Vielmehr muss die *Kombination* aus FDTD und ABC einer WT unterzogen werden.<sup>3</sup> Bei manueller Herleitung ist dies mit einem relativ hohen Aufwand verbunden, der sich bei jeder Erweiterung der FDTD wiederholt. Dieser Aufwand bleibt einem mit der Darstellung der ABC als Operator jedoch erspart!

Allgemeiner formuliert müssen die klassische FDTD und ihre Erweiterungen lediglich als Operatoren dargestellt werden, um „automatisch“

---

<sup>1</sup>Es lassen sich nichtlineare Eigenschaften höchstens dann sinnvoll einbeziehen, wenn diese lediglich einen unbedeutenden Anteil am zu simulierenden Volumen haben. Siehe hierzu Abschnitt 7.3.7.

<sup>2</sup>Genau genommen handelt es sich um eine *diskrete* WT (DWT). Die vereinfachende Bezeichnung WT ist jedoch in der Literatur ebenfalls üblich.

<sup>3</sup>Diese Aussagen sind etwas vereinfachend. Genaueres hierzu wird in Abschnitt 7.3.5 erörtert.

von den mit Wavelets verbundenen Kompressionseigenschaften profitieren zu können. Dies gilt entsprechend dem oben Gesagten sogar für sämtliche Simulationsverfahren, welche die genannten Voraussetzungen erfüllen. Darüber hinaus ist man bei der Wahl der verschiedenen Wavelets und der Art der Wavelet-Transformation vollkommen frei! Aus Gründen des Umfangs kann die vorliegende Arbeit jedoch nur einen Teil der Möglichkeiten untersuchen.

Zur Einführung in die angesprochenen Themen werden zunächst Grundlagen zur FDTD und Wavelets durchgesprochen. Die Darstellung und Verkettung von Operatoren wird in Kapitel 5 behandelt. Hier werden zwei grundsätzlich verschiedene Verfahren vorgestellt. Das erste Verfahren basiert auf einer Methode zur Transformation der FDTD in den Waveletbereich, welche erstmalig in [3] veröffentlicht wurde. In Abschnitt 5.2 wird dieser Ansatz verallgemeinert und einer detaillierten mathematischen Formulierung und Analyse unterzogen. Die recht umfangreichen Berechnungen sowie die daraus resultierenden Algorithmen offenbaren einige deutliche Nachteile dieses Verfahrens. Es wurde daher ein neuer Ansatz entwickelt, welcher in Abschnitt 5.3 vorgestellt wird. Mit diesem Ansatz lässt sich auch die Verkettung von Operatoren anschaulicher darstellen.

Im weiteren Verlauf der Arbeit werden verschiedene Möglichkeiten zur Reduktion der Simulationszeiten, d. h. zur Kompression der Operatoren behandelt. So wird u. a. in Abschnitt 7.3.6 eine Art *analytisches Untergitterverfahren* mit statischem Gitter entwickelt. Hier wird die Verwandtschaft von Untergitterverfahren und Bildkompressionsverfahren besonders deutlich. Analog zur Bildkompression sollen auch bei der Datenkompression mit Hilfe der Wavelets nur an ausgewählten Stellen die elektromagnetischen Felder bis ins Detail bekannt sein, während im übrigen Raum eine näherungsweise Darstellung genügt. Dadurch wird über die Wavelets auf analytische Weise eine Art Untergitter erzeugt, wobei im Gegensatz zu klassischen Untergitterverfahren jedoch keine scharfe Grenze zwischen dem „groben“ und dem „feinen“ Gitter existiert. Darüber hinaus kann gezeigt werden, dass das so gewonnene Verfahren die Stabilität der FDTD beibehält.

Die Ergebnisse aus einigen Simulationen werden in Kapitel 8 besprochen. Aus der Analyse wird erkennbar sein, wie sich weitere Leistungssteigerungen erzielen lassen. Kapitel 9 beschäftigt sich ebenfalls mit Verbesserungsmöglichkeiten. Zum einen lassen sich diese aus einem genaueren

Vergleich mit einem anderen Wavelet-basierten Simulationsverfahren, der sog. *Multiresolution time-domain* (MRTD) ableiten. Zum anderen werden bemerkenswerte Entwicklungen im Bereich der Wavelet-Theorie kurz vorgestellt, die ein hohes Innovationspotenzial erkennen lassen.

Die in dieser Arbeit entwickelten Techniken sind in der allgemein zugänglichen Literatur nach bestem Kenntnisstand unbekannt. Diese Techniken bilden ein recht mächtiges Werkzeug, welches die Untersuchung aller möglichen Kombinationen aus Simulationsverfahren, Wavelets und Transformationen erlaubt. Die Neuheit des Verfahrens und die Vielzahl der Kombinationen und Erweiterungsmöglichkeiten lassen hier lediglich die Untersuchung eines kleinen Ausschnitts zu. Neben der Validierung der Ideen schafft die vorliegende Arbeit damit ebenfalls die Grundlage für weitere Untersuchungen, welche hoch effiziente FDTD-Simulatoren für insbesondere räumlich ausgedehnte, d. h. *elektrisch große* Strukturen als Ziel haben können.



# Kapitel 2

## Das FDTD-Verfahren

Die Grundlage der Theorie elektromagnetischer Felder bilden die Maxwell'schen Gleichungen, die im Wesentlichen partielle Differentialgleichungen<sup>1</sup> in Raum und Zeit sind. Für eine computergerechte Simulation dieser Differentialgleichungen müssen Näherungen gefunden werden, die eine Beschränkung auf endlich viele Koeffizienten erlauben.

Yee löste diese Aufgabe 1966 dadurch, dass er die kontinuierlichen Differenziale durch zentrale finite Differenzen ersetzte [2]. Der resultierende Algorithmus ist in seiner ursprünglichen Form ausgesprochen einfach in einem Computerprogramm zu implementieren und verdankt nicht zuletzt dieser Eigenschaft seine große Verbreitung.

Mit diesem Verfahren ließen sich bereits viele abgeschirmte, d. h. vollständig mit Metall umgebene Strukturen simulieren. Der Wunsch nach der Simulation offener Strukturen leitete die Entwicklung sog. *absorbierender Randbedingungen* ein. Es existieren viele Erweiterungen und Verbesserungen der klassischen FDTD, so z. B. die Simulation nichtlinearer Medien, die Reduzierung des numerischen Aufwandes, die Erhöhung der numerischen Genauigkeit oder auch die Kombination mit anderen Simulationsverfahren. Einige dieser Erweiterungen sind beispielsweise in [4] nachzulesen.

Der Fokus wird in den folgenden Abschnitten auf dem „klassischen“ FDTD-Algorithmus liegen, ergänzt um absorbierende Randbedingungen.

---

<sup>1</sup>Genau genommen ist auch eine Integraldarstellung möglich.

Die dort dargestellten Gleichungen bilden die Grundlage für die in dieser Arbeit durchgeführten Simulationen. Sie veranschaulichen ebenfalls die bereits in der Einleitung erwähnten Eigenschaften, die für die noch darzustellende Wavelet-Transformation notwendig sind.

## 2.1 Die Diskretisierung der Maxwell'schen Gleichungen

Zur Herleitung einer Näherung der Maxwell'schen Gleichungen beginnen wir mit ihrer Darstellung in Differenzialform:

$$\operatorname{rot} \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} , \quad (2.1)$$

$$\operatorname{rot} \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} , \quad (2.2)$$

$$\operatorname{div} \mathbf{D} = \varrho , \quad (2.3)$$

$$\operatorname{div} \mathbf{B} = 0 . \quad (2.4)$$

Von diesen vier Gleichungen sind in den meisten Fällen die ersten beiden ausreichend, um das Feld vollständig zu beschreiben. Betrachtet man dann noch Materialien mit lediglich linearen und zeitinvarianten Eigenschaften, lassen sich die Beziehungen

$$\mathbf{D} = \varepsilon \mathbf{E} , \quad (2.5)$$

$$\mathbf{J} = \kappa \mathbf{E} , \quad (2.6)$$

$$\mathbf{B} = \mu \mathbf{H} \quad (2.7)$$

angeben. Damit vereinfachen sich die Gleichungen (2.1) und (2.2) zu

$$\operatorname{rot} \mathbf{H} = \kappa \mathbf{E} + \varepsilon \frac{\partial \mathbf{E}}{\partial t} , \quad (2.8)$$

$$\operatorname{rot} \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t} . \quad (2.9)$$

Für die weiteren Betrachtungen genügt es, sich auf eine skalare Gleichung zu beschränken. Dazu sind noch die Feldwerte und der Rotationsoperator

für ein kartesisches Koordinatensystem aufzuschreiben:

$$\operatorname{rot} \mathbf{F} = \begin{pmatrix} \frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z} \\ \frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x} \\ \frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \end{pmatrix}, \quad \mathbf{F} \in \{\mathbf{E}, \mathbf{H}\}. \quad (2.10)$$

Außerdem müssen die Materialtensoren  $\boldsymbol{\varepsilon}$ ,  $\boldsymbol{\mu}$  und  $\boldsymbol{\kappa}$  Hauptdiagonal-Matrizen sein:

$$\boldsymbol{\varepsilon} = \operatorname{diag}(\varepsilon_x, \varepsilon_y, \varepsilon_z), \quad \boldsymbol{\mu} = \operatorname{diag}(\mu_x, \mu_y, \mu_z), \quad \boldsymbol{\kappa} = \operatorname{diag}(\kappa_x, \kappa_y, \kappa_z). \quad (2.11)$$

Betrachten wir als Beispiel die x-Komponente der Gleichung (2.9), so stellt sich diese wie folgt dar:

$$\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\mu_x \frac{\partial H_x}{\partial t}. \quad (2.12)$$

Mit der Approximation der Differenziale durch Differenzenquotienten,

$$\left. \frac{df}{d\xi} \right|_{ih} \approx \frac{f((i + \frac{1}{2})h) - f((i - \frac{1}{2})h)}{h}, \quad i \in \mathbb{Z}, \quad (2.13)$$

wird schließlich die Diskretisierung eingeführt,<sup>2</sup> bei der die  $i$ -te Position den Koordinatenwert  $ih$  hat. Da Feldwerte nur noch an solchen Positionen betrachtet werden sollen, kann eine vereinfachende Schreibweise bzw. Abbildung benutzt werden, die sich auf die Indizierung der Positionen beschränkt:

$$\mathbf{E}^{n+\frac{1}{2}}(i, j, k) \stackrel{\text{def}}{=} \mathbf{E} \left( i\Delta x, \quad j\Delta y, \quad k\Delta z, \quad (n + \frac{1}{2})\Delta t \right), \quad (2.14)$$

$$\mathbf{H}^n(i, j, k) \stackrel{\text{def}}{=} \mathbf{H} \left( (i + \frac{1}{2})\Delta x, \quad (j + \frac{1}{2})\Delta y, \quad (k + \frac{1}{2})\Delta z, \quad n\Delta t \right), \quad (2.15)$$

$$i, j, k, n \in \mathbb{Z}.$$

Die Diskretisierungsschrittweiten  $\Delta x$ ,  $\Delta y$  und  $\Delta z$  geben hierbei die örtliche Auflösung an, während  $\Delta t$  die zeitliche Auflösung beschreibt. Abbildung 2.1 veranschaulicht die Zuordnung von Indizes zu Positionen im Raum. Aus Gleichung (2.12) wird mit diesem Schema

<sup>2</sup>Die Schrittweite  $h$  muss natürlich „genügend klein“ sein. Hierauf wird in Abschnitt 2.2 näher eingegangen.

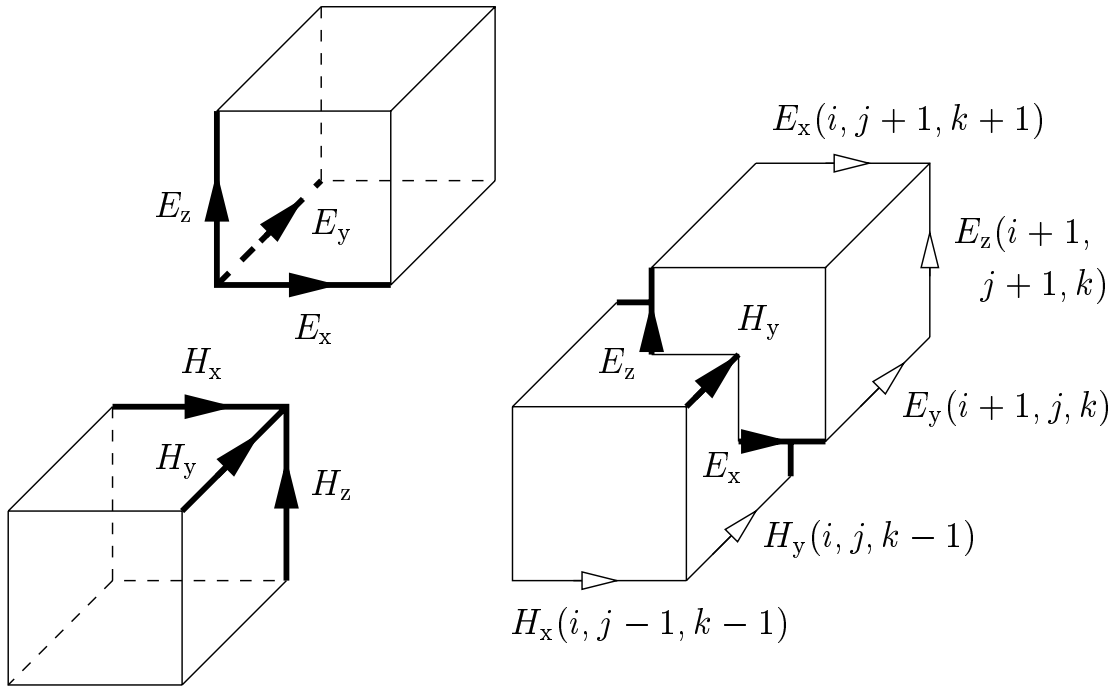


Abbildung 2.1: FDTD-Gitter. Links: Elementarzellen des  $\mathbf{E}$ - und  $\mathbf{H}$ -Feldes, Position  $(i, j, k)$ . Rechts: Yee-Zelle (beide Teilgitter zusammen) mit einigen Komponenten benachbarter Zellen.

$$\begin{aligned} & \frac{E_z^{n+\frac{1}{2}}(i, j+1, k) - E_z^{n+\frac{1}{2}}(i, j, k)}{\Delta y} - \frac{E_y^{n+\frac{1}{2}}(i, j, k+1) - E_y^{n+\frac{1}{2}}(i, j, k)}{\Delta z} \\ & = -\frac{\mu_x(i, j, k)}{\Delta t} \left( H_x^{n+1}(i, j, k) - H_x^n(i, j, k) \right) . \quad (2.16) \end{aligned}$$

Durch Umstellen von Gleichung (2.16) nach  $H_x^{n+1}(i, j, k)$  ist somit die Möglichkeit gegeben, aus den letzten beiden Zeitindizes  $n$  und  $n + \frac{1}{2}$  Feldwerte für den nächsten Zeitschritt  $n + 1$  zu berechnen:

$$\begin{aligned} H_x^{n+1}(i, j, k) & = H_x^n(i, j, k) \\ & - \frac{\Delta t}{\mu_x(i, j, k)} \left( \frac{E_z^{n+\frac{1}{2}}(i, j+1, k) - E_z^{n+\frac{1}{2}}(i, j, k)}{\Delta y} \right. \\ & \quad \left. - \frac{E_y^{n+\frac{1}{2}}(i, j, k+1) - E_y^{n+\frac{1}{2}}(i, j, k)}{\Delta z} \right) . \quad (2.17) \end{aligned}$$

Mit der x-Komponente von Gleichung (2.8) ist ähnlich zu verfahren. Das Problem ist jedoch, dass auf der rechten Seite der Gleichung die elek-

trische Feldstärke  $\mathbf{E}$  bzw. ihre Komponente  $E_x$  einmal nicht-abgeleitet und einmal nach der Zeit abgeleitet vorkommt. Nach der Diskretisierung ergibt das die Ausdrücke

$$E_x^n(i, j, k) \quad \text{und} \quad \frac{E_x^{n+\frac{1}{2}}(i, j, k) - E_x^{n-\frac{1}{2}}(i, j, k)}{\Delta t}. \quad (2.18)$$

Um nun analog wie mit  $H_x$  verfahren zu können, wird  $E_x^n$  durch den Mittelwert aus dem vorhergehenden und dem nachfolgenden (Halb-) Zeitschritt ersetzt:

$$E_x^n \stackrel{\text{def}}{=} \frac{E_x^{n-\frac{1}{2}} + E_x^{n+\frac{1}{2}}}{2}. \quad (2.19)$$

Dadurch kann nun  $E_x^{n+\frac{1}{2}}$  in Abhängigkeit von  $E_x^{n-\frac{1}{2}}$  und  $H_{y,z}^n$  ausgedrückt werden:

$$\begin{aligned} E_x^{n+\frac{1}{2}}(i, j, k) &= \frac{2\varepsilon_x(i, j, k) - \kappa_x(i, j, k)\Delta t}{2\varepsilon_x(i, j, k) + \kappa_x(i, j, k)\Delta t} E_x^{n-\frac{1}{2}}(i, j, k) \\ &+ \frac{2\Delta t}{2\varepsilon_x(i, j, k) + \kappa_x(i, j, k)\Delta t} \left( \frac{H_z^n(i, j, k) - H_z^n(i, j-1, k)}{\Delta y} \right. \\ &\quad \left. - \frac{H_y^n(i, j, k) - H_y^n(i, j, k-1)}{\Delta z} \right). \quad (2.20) \end{aligned}$$

In analoger Weise sind Ausdrücke für  $E_{y,z}^{n+\frac{1}{2}}$  und  $H_{y,z}^{n+1}$  herzuleiten. Gibt man sich nun eine Feldverteilung  $\mathbf{H}^0$ ,  $\mathbf{E}^{\frac{1}{2}}$  als Anfangswerte vor, lassen sich nacheinander die Feldwerte  $\mathbf{H}^1$ ,  $\mathbf{E}^{\frac{3}{2}}$ ,  $\mathbf{H}^2$ ,  $\mathbf{E}^{\frac{5}{2}}$  usw. berechnen.

Die explizite Ortsabhängigkeit der Materialparameter  $\varepsilon_x$ ,  $\mu_x$  und  $\kappa_x$  wird bei der Wavelet-Transformation der FDTD-Gleichungen noch eine große Rolle spielen. Genau genommen sind sogar die räumlichen Diskretisierungsschrittweiten  $\Delta x$ ,  $\Delta y$  und  $\Delta z$  vom Ort und damit von  $(i, j, k)$  abhängig, da in der Praxis mit sog. *nicht-äquidistanten Gittern* die zu simulierenden Strukturen im allgemeinen genauer nachgebildet werden können als mit einer über den ganzen Raum gleichbleibenden Gitterweite.

## 2.2 Numerische Dispersion und Stabilität

Die Zeitschrittweite  $\Delta t$  in den Gleichungen (2.17) und (2.20) scheint beliebig wählbar. Eine genauere mathematische Analyse ergibt jedoch, dass bei einem zu groß gewählten Wert für  $\Delta t$  die Feldwerte in ihrer Amplitude mit zunehmender Zeit exponentiell wachsen können – das Verfahren wird instabil. Dispersion ist eine weitere Eigenschaft des Algorithmus von Yee, die bei der Wahl der räumlichen Auflösung zu beachten ist. Es soll hier kurz auf diese Analyse eingegangen werden, deren Details beispielsweise in [4] nachzulesen sind.

Betrachtet wird ein homogener, isotroper Raum mit der Permittivität  $\varepsilon$  und der Permeabilität  $\mu$ . Der Raum wird mit konstanten Werten für  $\Delta x$ ,  $\Delta y$  und  $\Delta z$  diskretisiert. In diesem Raum soll sich eine ungedämpfte Welle mit den Phasenkonstanten  $\beta_x$ ,  $\beta_y$  und  $\beta_z$  ausbreiten. Die Kreisfrequenz der Welle wird mit  $\omega$  bezeichnet. Wird diese Welle mathematisch entsprechend formuliert in die FDTD-Gleichungen eingesetzt, so kann gezeigt werden, dass folgende Gleichung erfüllt sein muss:

$$\frac{\sin^2(\frac{1}{2}\omega\Delta t)}{(c\Delta t)^2} = \frac{\sin^2(\frac{1}{2}\beta_x\Delta x)}{(\Delta x)^2} + \frac{\sin^2(\frac{1}{2}\beta_y\Delta y)}{(\Delta y)^2} + \frac{\sin^2(\frac{1}{2}\beta_z\Delta z)}{(\Delta z)^2}. \quad (2.21)$$

Hierbei ist  $c = (\varepsilon\mu)^{-\frac{1}{2}}$  die Lichtgeschwindigkeit in dem betrachteten Medium. Aus dem Vergleich mit den physikalischen Verhältnissen,

$$\left(\frac{\omega}{c}\right)^2 = \beta_x^2 + \beta_y^2 + \beta_z^2, \quad (2.22)$$

lässt sich zeigen, dass sich die Welle in dem diskretisierten Raum je nach Ausbreitungsrichtung verschieden schnell fortbewegt. Der Grad dieser Richtungsabhängigkeit ist darüber hinaus von dem Verhältnis von Wellenlänge zu örtlicher Auflösung abhängig: Die Dispersion ist umso ausgeprägter je gröber der Raum diskretisiert wurde. In der Praxis wird zur Einschränkung dieses Effektes als Obergrenze für die Gitterweiten oft das 0,1-fache der minimal zu simulierenden Wellenlänge genommen.

Es bleibt noch nach einer oberen Schranke für  $\Delta t$  zu suchen. Dazu soll Gleichung (2.21) zunächst nach  $\omega$  umgestellt werden:

$$\omega = \frac{2}{\Delta t} \arcsin(\alpha), \quad (2.23)$$

mit

$$\alpha = c\Delta t \sqrt{\frac{\sin^2(\frac{1}{2}\beta_x\Delta x)}{\Delta x^2} + \frac{\sin^2(\frac{1}{2}\beta_y\Delta y)}{\Delta y^2} + \frac{\sin^2(\frac{1}{2}\beta_z\Delta z)}{\Delta z^2}}. \quad (2.24)$$

Damit die Welle eine konstante Amplitude hat, muss  $\omega$  reellwertig sein, was wegen des arcsin in Gleichung (2.23) zu der Ungleichung  $0 \leq |\alpha| \leq 1$  und damit zu

$$\Delta t \leq \frac{1}{c \sqrt{\frac{\sin^2(\frac{1}{2}\beta_x\Delta x)}{\Delta x^2} + \frac{\sin^2(\frac{1}{2}\beta_y\Delta y)}{\Delta y^2} + \frac{\sin^2(\frac{1}{2}\beta_z\Delta z)}{\Delta z^2}}} \quad (2.25)$$

führt. Um nun die Stabilität des FDTD-Verfahrens für Wellen mit beliebiger Ausbreitungsrichtung zu gewährleisten, kann wegen  $\sin^2(\cdot) \leq 1$  eine neue obere Schranke für  $\Delta t$  eingeführt werden:

$$\Delta t \leq \Delta t_{\text{FDTD}} \stackrel{\text{def}}{=} \frac{1}{c \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}}}. \quad (2.26)$$

Aus der Definition für  $\Delta t_{\text{FDTD}}$  wird ersichtlich, dass eine Erhöhung der räumlichen Auflösung eine Verringerung der maximalen Zeitschrittweite erfordert. Auf diese Weise erhöht sich der gesamte Rechenaufwand einer Simulation nicht nur durch die entsprechend erhöhte Anzahl an Feldkoeffizienten, sondern zusätzlich noch durch die notwendigerweise gestiegene Anzahl an Zeitschritten.

Aus verschiedenen Gründen ist  $\Delta t_{\text{FDTD}}$  oftmals noch weiter abzusenken. So ist bei der Simulation eines inhomogenen Mediums für jeden Punkt die maximal zulässige Zeitschrittweite zu berechnen, um dann am Ende aus der Menge dieser Werte den kleinsten davon als obere Schranke für den gesamten Simulationsraum zu verwenden. Hierbei sind auch Anisotropien und ortsabhängige Diskretisierungsweiten zu berücksichtigen. Lokal „ungünstige“ Parameter wirken sich damit global aus.

Darüber hinaus erfordern auch viele Erweiterungen der klassischen FDTD, dass ein kleineres  $\Delta t$  gewählt wird als durch  $\Delta t_{\text{FDTD}}$  erlaubt. Dadurch führen solche „potenziell instabilen“ Verfahren wenigstens für beispielsweise einige tausend Zeitschritte zu verwertbaren Ergebnissen.

Viele der bereits erwähnten absorbierenden Randbedingungen gehören zu den Verfahren, bei denen es zu instabilen Simulationsergebnissen kommen kann. Dennoch sind diese Erweiterungen zu einem unverzichtbaren Bestandteil des FDTD-Verfahrens geworden, sodass lieber eine erhöhte Simulationszeit in Kauf genommen wird, als gänzlich auf solche Neuerungen zu verzichten.

## 2.3 Randbedingungen

Die bisherige Beschreibung der FDTD geht von einem unendlich ausgedehnten Raum aus. Bei geschlossenen Strukturen, wie z. B. Hohlraumresonatoren, genügt jedoch die Beschränkung auf den mit metallischen Wänden begrenzten Hohlraum. Offene Strukturen lassen sich mitunter in genügend großen, aber dennoch abgeschlossenen Räumen simulieren. Eine Verbesserung dieser Vorgehensweise stellen die bereits erwähnten absorbierenden Randbedingungen dar. Oftmals sind auch halboffene Strukturen zu simulieren, bei denen beide Randbedingungen zum Einsatz kommen. Ihre Implementation im FDTD-Verfahren soll in den folgenden Abschnitten dargestellt werden.

### 2.3.1 Ideal leitende Grenzschichten

In der klassischen Feldtheorie führt die Annahme elektrisch ideal leitender Metalle zu der Grenzschichtbedingung  $\mathbf{E}_{\text{tan}} = 0$ , d. h. zur Grenzschicht tangential elektrische Feldkomponenten verschwinden. Derartige Grenzschichten werden auch *elektrische Wände* genannt. Zu ihrer Implementierung sind in der FDTD und anderen Verfahren zwei Methoden gebräuchlich. Die Auswahl hängt davon ab, wo sich die Grenzschicht im Vergleich zu dem Diskretisierungsschema nach Gleichung (2.14) befindet.

Bei der ersten Methode wird angenommen, dass die Grenzschicht mit Flächen der Elementarzellen des elektrischen Feldes zusammenfällt (siehe Abbildung 2.1). Dadurch befinden sich die zugehörigen elektrischen Feldkomponenten genau in der ideal leitenden Grenzschicht. In den FDTD-Gleichungen für  $\mathbf{H}^{n+1}$  sind dann diese elektrischen Feldkomponenten einfach durch den Wert Null zu ersetzen. Ferner wird die Berechnung dieser elektrischen Feldkomponenten unterdrückt.

Bei der zweiten Methode befindet sich die Grenzschicht symmetrisch zwischen den Flächen der Elementarzellen des elektrischen Feldes und damit symmetrisch zwischen den zugehörigen Feldkomponenten. Für diese Feldkomponenten kann das Spiegelungsprinzip angewendet werden. Liegt beispielsweise bei  $z = \frac{1}{2}\Delta z$  eine in x- und y-Richtung unendlich ausgedehnte, ideal leitende Grenzschicht, so ist

$$E_\nu(i, j, 0) = -E_\nu(i, j, 1) \quad \forall i, j \in \mathbb{Z}, \quad \nu \in \{x, y\}, \quad (2.27)$$

in die betroffenen Gleichungen für  $\mathbf{H}^{n+1}$  einzusetzen. Diese Methode kommt vor allem bei solchen Verfahren zum Einsatz, die zur Approximation der Differenziale mehr als nur zwei Stützstellen benutzen.

Ganz analog lassen sich magnetische Grenzschichten definieren, in denen  $\mathbf{H}_{\text{tan}} = 0$  gilt. Diese *magnetischen Wände* kommen vor allem dann zum Einsatz, wenn a priori eine entsprechende Symmetrie der Felder bekannt ist bzw. forciert wird. Dadurch lässt sich der zu simulierende Raum and damit die benötigte Rechenzeit halbieren.

Bei beiden Methoden müssen kurvige und schräg zu den Koordinatenachsen verlaufende Grenzschichten durch stufenweise Approximation dem Diskretisierungsgitter angepasst werden. Dies ist einer der Gründe, warum oftmals auf eine hohe räumliche Auflösung nicht verzichtet werden kann.

### 2.3.2 Absorbierende Randbedingungen

Während bei abgeschirmten Strukturen der zu simulierende Raum in natürlicher Weise durch die äußere Berandung gegeben ist, muss bei offenen Strukturen eine künstliche Beschränkung auf den „wesentlichen“ Feldbereich stattfinden. Der außerhalb dieses Feldbereichs liegende Raum ist so gut wie möglich durch absorbierende Randbedingungen (engl. *Absorbing Boundary Condition*, ABC) nachzubilden. Zwei Anforderungen sind dabei besonders zu beachten: Die an die Struktur gebundenen Feldverteilungen dürfen nicht gestört werden und die ungebundenen (abstrahlenden) Feldanteile müssen von der ABC absorbiert werden.

Die erste Anforderung bedingt bei allen ABCs die Einhaltung eines gewissen Mindestabstands von den Gebieten mit relativ hohen Anteilen an gebundener Feldenergie. Dieser Abstand ist von der Art der ABC eher unabhängig und bewirkt, dass der zu simulierende Raum nicht durch ABCs

beliebig klein gemacht werden kann. Bei der zweiten Anforderung bestehen starke Unterschiede zwischen den ABCs, sowohl im Hinblick auf den unter optimalen Bedingungen erreichbaren Reflexionsfaktor der ABC als auch auf die Winkelabhängigkeit des Reflexionsfaktors. Eine gute Übersicht über verschiedene Verfahren und ihre Eigenschaften ist in [4] zu finden.

Von allen bisher für die FDTD entwickelten absorbierende Randbedingungen hat sich in den letzten Jahren vor allem die von Bérenger entwickelte *Perfectly Matched Layer* (PML) hervorgetan. Im Vergleich zu bis dato bereits etablierten ABCs erlaubte sie sehr häufig dramatische Verbesserungen der Genauigkeit der Simulationsergebnisse. Dies ist vor allem auf ihre sehr hohen Absorptionskoeffizienten für einfallende Wellen aus beliebigen Winkeln zurückzuführen.

Dennoch hat auch die PML Nachteile: Sie kann instabil sein [5, 6], weist unter bestimmten Bedingungen ungewöhnlich hohe Reflexionsfaktoren auf [7] und erfordert eine unphysikalische Erweiterung und Zerlegung der Maxwell'schen Gleichungen und ihrer diskretisierten Formen. Verschiedene Variationen und ähnliche Konzepte wurden zur Überwindung dieser Nachteile entwickelt [8, 9, 10].

Aus verschiedenen Gründen wird hier dennoch die PML in ihrer klassischen Form verwendet. So dient sie nicht nur der Erzeugung von Referenzergebnissen. Es soll damit in den noch folgenden Kapiteln auch demonstriert werden, dass sogar solche Erweiterungen der klassischen FDTD bei der Wavelet-Transformation verwendet werden können, die zwar teilweise zur FDTD ähnliche aber dennoch deutlich unterschiedliche Gleichungen benutzen.

Zunächst wird ein Medium mit isotropen dielektrischen Eigenschaften vorausgesetzt, sodass sich die Materialtensoren  $\boldsymbol{\varepsilon}$  und  $\boldsymbol{\mu}$  auf die skalaren Größen  $\varepsilon$  und  $\mu$  reduzieren. Bérengers PML basiert auf der Modifikation der Maxwell'schen Gleichungen in der Form, dass senkrecht auf eine PML einfallende Wellenanteile absorbiert werden und parallel zu ihr verlaufende Wellenanteile ungestört ausbreitungsfähig bleiben:

$$\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\mu \frac{\partial H_x}{\partial t} \quad \begin{array}{l} \nearrow \\ \searrow \end{array} \quad \begin{array}{l} \frac{\partial(E_{zx} + E_{zy})}{\partial y} = -\mu \frac{\partial H_{xy}}{\partial t} - \sigma_y H_{xy} \\ -\frac{\partial(E_{yx} + E_{yz})}{\partial z} = -\mu \frac{\partial H_{xz}}{\partial t} - \sigma_z H_{xz} \end{array} \quad (2.28)$$

Diese Aufteilung von Gleichung (2.12) beinhaltet verschiedene neuartige Elemente. Zum einen werden die Feldkomponenten in jeweils zwei Anteile aufgespalten:

$$E_z = E_{zx} + E_{zy} , \quad E_y = E_{yx} + E_{yz} , \quad H_x = H_{xy} + H_{xz} . \quad (2.29)$$

Weiterhin gibt es mit den Elementen  $\sigma_y$  und  $\sigma_z$  eine Art anisotroper magnetischer Leitfähigkeit. Zusammen mit der Aufstellung der restlichen PML-Gleichungen erhält man dann die beiden Leitfähigkeitsmatrizen  $\boldsymbol{\kappa} = \text{diag}(\kappa_x, \kappa_y, \kappa_z)$  und  $\boldsymbol{\sigma} = \text{diag}(\sigma_x, \sigma_y, \sigma_z)$ . Es ist zu beachten, dass die elektrischen Leitfähigkeiten hier in anderer Form in Erscheinung treten als in den Maxwell'schen Gleichungen. Zum Vergleich: Die x-Komponente von Gleichung (2.8) enthält den Term  $\kappa_x E_x$ , während die beiden zugehörigen PML-Gleichungen die Terme  $\kappa_y E_{xy}$  und  $\kappa_z E_{xz}$  enthalten.

Nicht-verschwindende Leitfähigkeiten sollen dazu dienen, die senkrecht auf eine PML eintreffenden Wellen zu absorbieren (vergleichbar einem sog. *Sumpf* in der Hochfrequenztechnik). Dazu ist dem Feldraum an der entsprechenden Stelle eine PML mit genügender Tiefe anzugliedern. In Abbildung 2.2 ist ein solches Element mit der Bezeichnung „Fläche“ angegeben.

Zeigt der Normalenvektor der Grenzfläche beispielsweise in x-Richtung, so müssen zur Absorption von in x-Richtung laufenden Wellenanteilen  $\kappa_x$  und  $\sigma_x$  von Null verschieden sein. Damit in y- und z-Richtung laufende Wellenanteile in der PML-Fläche ungestört ausbreitungsfähig sind, müssen  $\kappa_{y,z}$  und  $\sigma_{y,z}$  Null sein.

In PML-Kanten sind die Leitfähigkeiten so zu wählen, dass Wellen nur noch entlang der Kantenlänge ausbreitungsfähig sind. In PML-Ecken werden schließlich Wellen aller Ausbreitungsrichtungen absorbiert. Damit Wellen nicht bereits an den Grenzflächen zwischen Feldraum und PMLs reflektiert werden, sind die Feldwellenimpedanzen anzupassen. Dazu erhält jede PML-Zelle zunächst einmal die dielektrischen Konstanten

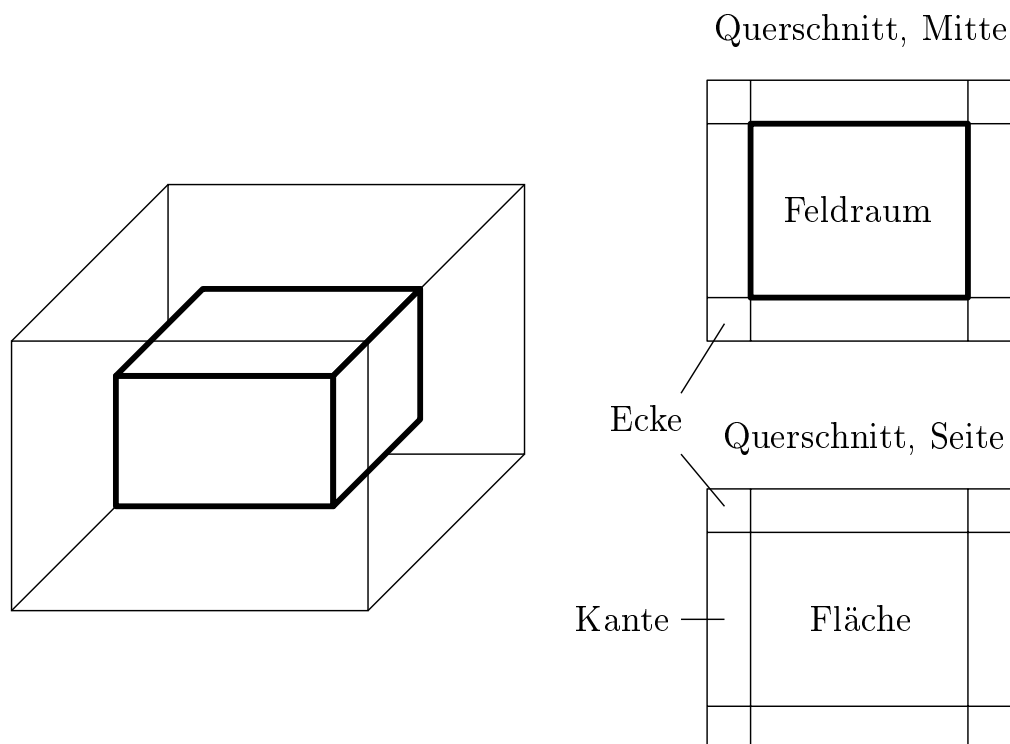


Abbildung 2.2: Allseitig mit PMLs umgebener Feldraum. Links dreidimensionale Ansicht, rechts Querschnittsansichten mit Angabe der PML-Elemente.

derjenigen Yee-Zelle, die ihr am nächsten ist. Danach sind die Leitfähigkeiten so zu wählen, dass stets

$$\frac{\kappa_\nu}{\varepsilon} = \frac{\sigma_\nu}{\mu} \quad \forall \nu \in \{x, y, z\} \quad (2.30)$$

gilt. Über die Tiefe der PML, die Abhängigkeit der Leitfähigkeiten von der Entfernung zum Feldraum und die Auswahl der maximalen Leitfähigkeitswerte kann der noch verbleibende Reflexionsfaktor bestimmt werden. Umgekehrt können auch mit der Vorgabe des Reflexionsfaktors und weiterer Werte z. B. die maximal zulässigen Leitfähigkeiten errechnet werden. Näheres hierzu kann in [4] nachgeschlagen werden.

# Kapitel 3

## Wavelets

Die Entwicklung vieler Funktionenklassen war das Resultat der Lösung mathematischer Formulierungen technischer Probleme. Die Funktionenklassen – unter denen sich so bekannte wie Polynomfunktionen, trigonometrische Funktionen und Besselfunktionen befinden – werden häufig angewendet, um Näherungslösungen in möglichst kompakter Form, d. h. mit möglichst wenigen Termen beschreiben zu können.

Bevor auf die Eigenschaften der hier vorzustellenden Funktionenklasse der Wavelets eingegangen wird, sollen am Beispiel der Sinus- und Kosinus-Funktionen einige Merkmale klassischer Entwicklungsfunktionen dargestellt werden. Beim Vergleich mit den Wavelets wird dann deutlich, warum sich letztgenannte in den letzten Jahren vor allem im Bereich der Bildkompression zu einem äußerst nützlichen Werkzeug entwickelt haben.

### 3.1 Die Fouriertransformation

Ein Signal  $f_0(t)$  soll auf seine Frequenzanteile untersucht werden. Das Signal soll stückweise stetig und absolut integrierbar sein.<sup>1</sup> Dann berechnet sich die Fouriertransformierte von  $f_0(t)$  zu

$$\hat{f}_0(\omega) = \mathcal{F} f_0(\omega) \stackrel{\text{def}}{=} \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f_0(t) e^{-j\omega t} dt . \quad (3.1)$$

---

<sup>1</sup>Im Allgemeinen muss es der *Dirichlet-Jordan-Bedingung* genügen, vgl. [11].

Mit der inversen Fouriertransformation

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}_0(\omega) e^{j\omega t} d\omega \quad (3.2)$$

gilt dann

$$f(t) = \frac{1}{2} \lim_{\varepsilon \rightarrow 0} (f_0(t - \varepsilon) + f_0(t + \varepsilon)) . \quad (3.3)$$

Gleichung (3.3) drückt aus, dass  $f_0(t)$  durch  $f(t)$  im angegebenen Intervall bis auf die Unstetigkeitsstellen exakt wiedergegeben werden kann. Die Gleichung (3.1) zeigt, dass zur Berechnung der Frequenzanteile stets das vollständige Intervall herangezogen werden muss. Die Entwicklungsfunktion  $e^{-j\omega t}$  kann als Gewichtungsfunktion betrachtet werden. Wegen ihrer Eigenschaft  $|e^{-j\omega t}| = 1$  gilt offensichtlich, dass die Fouriertransformation keine direkten Rückschlüsse auf lokal begrenzte Eigenschaften von  $f_0(t)$  zulässt – alle Werte von  $f_0(t)$  gehen gleichermaßen in alle Frequenzanteile ein. Umgekehrt gehen auch alle Frequenzanteile gleichermaßen in die Rücktransformierte ein.

Ein kleines Beispiel soll das verdeutlichen. Es soll

$$f_0(t) = \begin{cases} 1 & , t_1 \leq t \leq t_2 \\ 0 & , \text{sonst} \end{cases} , \quad t_1 < t_2 \quad (3.4)$$

gelten. Dann ergibt sich das zugehörige Frequenzspektrum zu

$$\hat{f}_0(\omega) = \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{-j\omega} [e^{-j\omega t}]_{t_1}^{t_2} = \frac{j}{\sqrt{2\pi}\omega} (e^{-j\omega t_2} - e^{-j\omega t_1}) . \quad (3.5)$$

Der Ausdruck  $e^{-j\omega t_2} - e^{-j\omega t_1}$  stellt eine gleichbleibende Oszillation im Frequenzbereich dar. Eine anschauliche Beziehung zwischen einem derartigen Verlauf und der Position des Impulses zu finden, ist kaum möglich. Weiterhin ist zu sehen, dass im Wesentlichen nur die Proportionalität  $\frac{1}{\omega}$  in Gleichung (3.5) eine Beschränkung des Frequenzspektrums erlaubt, wenn eine näherungsweise Darstellung genügt. Dennoch ist eine derartige Abnahme der Amplituden mit zunehmender Frequenz eher als langsam zu bezeichnen und im Hinblick auf Kompressionszwecke daher nur bedingt nutzbringend. Alles in allem wird deutlich, dass es der Fouriertransformation an der Fähigkeit mangelt, örtlich begrenzte Signale auch durch

ein begrenztes Frequenzspektrum darzustellen. Die diskreten Formen der Fouriertransformation unterliegen diesem Mangel in ähnlicher Weise.

Die Entwicklung von Signalen in Fourierintegrale ist vor allem bei glatten Funktionen sinnvoll, da dort das Amplitudenspektrum eher ein schnelles Abklingverhalten besitzt. Viele Signale oder auch Bilder besitzen in großen Bereichen glatte Verläufe, sodass gute Kompressionsraten mit Verfahren erzielbar sind, die auf Entwicklungen in trigonometrischen Funktionen basieren. Dennoch sind oft Kompressionsverfahren wünschenswert, die toleranter sind gegenüber weniger glatten Funktionen wie z. B. dem gerade behandelten Rechteckimpuls.

## 3.2 Grundlagen zur Wavelet-Theorie

### 3.2.1 Die kontinuierliche Wavelet-Transformation

Die im Folgenden dargestellten Grundlagen wurden vorwiegend [12] und [13] entnommen, wenn auch teilweise eine andere Symbolik verwendet wurde. Betrachten wir den Raum  $L_2(\mathbb{R})$  der auf dem reellen Raum quadratisch integrierbaren Funktionen. Eine Funktion  $\psi \in L_2(\mathbb{R})$  ist genau dann ein Wavelet, wenn sie die *Zulässigkeitsbedingung*

$$0 < c_\psi = 2\pi \int_{\mathbb{R}} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty \quad (3.6)$$

erfüllt.<sup>2</sup> Setzt man zusätzlich  $\psi \in L_1(\mathbb{R})$  voraus, ergeben sich bereits einige Eigenschaften von Wavelets:

1. Wavelets besitzen eine Art Lokalisierung. Das bedeutet, dass ein endliches Intervall angegeben werden kann, innerhalb dessen die Fläche von  $|\psi|$  größer ist als außerhalb. Dies folgt aus der Eigenschaft der Fouriertransformation  $\mathcal{F}\psi(\omega) \rightarrow 0$  für  $\omega \rightarrow \pm\infty$ , siehe z. B. [11].

---

<sup>2</sup>Es existieren auch allgemeinere Definitionen von Wavelets, siehe z. B. [12], Lemma 1.1.4.

2. Wavelets haben einen verschwindenden Mittelwert. Dies folgt aus der Stetigkeit der Fouriertransformierten von  $\psi$  (siehe [11]) und der Zulässigkeitsbedingung (3.6).

Mit dem Wavelet  $\psi$  soll eine Funktion  $f \in L_2(\mathbb{R})$  nun transformiert werden. Die Wavelet-Transformation ist dann definiert zu

$$\mathcal{Q}_\psi f(a, b) \stackrel{\text{def}}{=} \frac{1}{\sqrt{|a|c_\psi}} \int_{\mathbb{R}} f(t) \psi\left(\frac{t-b}{a}\right) dt, \quad (3.7)$$

mit

$$a \in \mathbb{R} \setminus \{0\}, \quad b \in \mathbb{R}. \quad (3.8)$$

Entsprechend der Definition kann man auch  $a$  als Dilatationsparameter und  $b$  als Translationsparameter bezeichnen. Zusammen mit den o. g. Eigenschaften von Wavelets können bereits jetzt einige Vergleiche zwischen Wavelettransformation (WT) und Fouriertransformation (FT) gezogen werden:

1. Die Lokalisierungseigenschaft von Wavelets erlaubt es, Funktionen „abzutasten“. Mit dem Translationsparameter  $b$  ist das Untersuchen von  $f$  auf Anteile von  $\psi$  über den Ort auflösbar. In der FT ist dagegen keine ortsauflösende Untersuchung möglich, vgl. Abschnitt 3.1.
2. Der Dilatationsparameter  $a$  gestattet es, die ortsauflösende Untersuchung auf verschiedenen *Skalen* durchzuführen. Diese Skalen entsprechen den Frequenzen in der FT, jedoch kommt in der WT zusätzlich die Lokalisierung zum Tragen.

Zu der WT existiert auch eine Inversionsformel. Durch die zwei Parameter  $(a, b)$  stellt sich die Rücktransformation allerdings etwas komplizierter dar. Sei  $g = \mathcal{Q}_\psi f$ , dann ergibt der durch

$$\mathcal{Q}_\psi^* g(t) \stackrel{\text{def}}{=} \frac{1}{\sqrt{c_\psi}} \int_{\mathbb{R}} \int_{\mathbb{R}} \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) g(a, b) \frac{da db}{a^2} \quad (3.9)$$

definierte adjungierte Operator wieder  $f = \mathcal{Q}_\psi^* g$ .

Die WT weist wie weiter oben beschrieben den Vorzug der Lokalisierungseigenschaft gegenüber der FT auf. Das Doppelintegral in Gleichung (3.9) stellt jedoch gegenüber dem einfachen Integral bei der inversen Fouriertransformation einen Nachteil dar. So ist es nicht verwunderlich, dass die Wavelet-Theorie erst mit der Entdeckung der im folgenden Abschnitt beschriebenen diskreten WT eine größere Verbreitung erfuhr.

### 3.2.2 Die Multi-Skalen-Analyse

Die Vereinfachung von Gleichung (3.9) lässt sich erreichen, indem an die Wavelets weitere Voraussetzungen gestellt werden. Aus diesen Voraussetzungen wurde das Konzept der *Wavelet-Frames* entwickelt, welches die Benutzung von Doppelsummen statt Doppelintegralen erlaubt [12]. Der wohl am häufigsten benutzte Frame führt dabei auf die weiter unten angegebene Definition der Multi-Skalen-Analyse (MSA).

Die MSA erlaubt im Gegensatz zur kontinuierlichen WT die Zerlegung eines Signals in diskrete *Räume*, wobei jeder Raum eine bestimmte Auflösungs- bzw. Detailstufe oder auch Skala repräsentiert (Dilationsparameter  $a$ ). Außerdem sind die zugehörigen Basisfunktionen eines Raums diskrete Translate (Translationsparameter  $b$ ) einer einzigen Funktion. Während in der FT bzw. bei Fourierreihen [11] die Frequenzanteile eines Signals unabhängig voneinander ermittelt werden können, bauen die Räume in der MSA aufeinander auf: Jeder Raum einer bestimmten Auflösungsstufe enthält sämtliche Räume der „gröberen“ Auflösungsstufen. Dadurch wird eine andere Art der Zerlegung eines Signals in höher- und niederfrequente Anteile möglich als es bei der FT der Fall ist.

Eine MSA des  $L_2(\mathbb{R})$  besteht aus einer aufsteigenden Folge abgeschlossener Unterräume  $\mathbf{V}_m$  mit folgenden Eigenschaften:

$$\emptyset \subset \dots \subset \mathbf{V}_2 \subset \mathbf{V}_1 \subset \mathbf{V}_0 \subset \mathbf{V}_{-1} \subset \mathbf{V}_{-2} \subset \dots \subset L_2(\mathbb{R}) , \quad (3.10)$$

$$\overline{\bigcup_{m \in \mathbb{Z}} \mathbf{V}_m} = L_2(\mathbb{R}) , \quad (3.11)$$

$$\bigcap_{m \in \mathbb{Z}} \mathbf{V}_m = \emptyset , \quad (3.12)$$

$$f(\cdot) \in \mathbf{V}_m \iff f(2^m \cdot) \in \mathbf{V}_0 . \quad (3.13)$$

Weiterhin muss es eine Funktion  $\varphi \in \mathbf{L}_2(\mathbb{R})$  geben, die mit ihren ganzzahligen Translaten eine Riesz-Basis von  $\mathbf{V}_0$  erzeugt. Dies führt auf die Bedingungen

$$\mathbf{V}_0 = \overline{\text{span}\{\varphi(\cdot - k) \mid k \in \mathbb{Z}\}} \quad (3.14)$$

und

$$\begin{aligned} \exists A, B \in \mathbb{R}_+ \quad : \quad A \sum_{k \in \mathbb{Z}} c_k^2 \leq \left\| \sum_{k \in \mathbb{Z}} c_k \varphi(\cdot - k) \right\|_{\mathbf{L}_2}^2 \leq B \sum_{k \in \mathbb{Z}} c_k^2 \\ \forall (c_k)_{k \in \mathbb{Z}} \in \mathbf{l}_2(\mathbb{Z}) . \end{aligned} \quad (3.15)$$

Die Funktion  $\varphi$  wird *Skalierungsfunktion* genannt. Setzt man sie in Gleichung (3.13) ein, so zeigt sich, wie durch die Auswahl eines Raums die Wahl für eine bestimmte Auflösungsstufe getroffen werden kann.

Mit Gleichung (3.10) lassen sich aus den (Skalierungs-)Räumen  $\mathbf{V}_m$  nun Waveleträume  $\mathbf{W}_m$  definieren:

$$\mathbf{V}_m = \mathbf{V}_{m+1} \oplus \mathbf{W}_{m+1} . \quad (3.16)$$

Damit sind die  $\mathbf{W}_m$  orthogonal zum jeweiligen  $\mathbf{V}_m$ . Die in Dilatation und Translation diskreten Projektionen auf  $\mathbf{V}$ - und  $\mathbf{W}$ -Räume sollen mit  $\mathcal{R}$  und  $\mathcal{Q}$  bezeichnet werden, sodass gilt

$$\mathcal{R}_m f \in \mathbf{V}_m , \quad (3.17)$$

$$\mathcal{Q}_m f \in \mathbf{W}_m , \quad (3.18)$$

$$\mathcal{R}_m f = \mathcal{R}_{m+1} f + \mathcal{Q}_{m+1} f . \quad (3.19)$$

Auf Gleichung (3.10) angewendet ist folgende Darstellung möglich:

$$\begin{array}{ccccccccccc} \mathbf{L}_2(\mathbb{R}) & \dots & \mathbf{V}_{-1} & \xrightarrow{\mathcal{R}_0} & \mathbf{V}_0 & \xrightarrow{\mathcal{R}_1} & \mathbf{V}_1 & \xrightarrow{\mathcal{R}_2} & \mathbf{V}_2 & \dots & \emptyset \\ & & & \searrow \mathcal{Q}_0 & & \searrow \mathcal{Q}_1 & & \searrow \mathcal{Q}_2 & & & \\ & & \dots & & \mathbf{W}_0 & & \mathbf{W}_1 & & \mathbf{W}_2 & \dots & \emptyset \end{array} \quad (3.20)$$

Daher gilt offensichtlich

$$\mathbf{V}_m = \bigoplus_{n \geq m+1} \mathbf{W}_n , \quad (3.21)$$

$$\mathbf{L}_2(\mathbb{R}) = \overline{\bigoplus_{m \in \mathbb{Z}} \mathbf{W}_m} . \quad (3.22)$$

Analog zu den Gleichungen (3.13) und (3.14) gelten auch

$$f(\cdot) \in \mathbf{W}_m \iff f(2^m \cdot) \in \mathbf{W}_0, \quad (3.23)$$

$$\mathbf{W}_0 = \overline{\text{span}\{\psi(\cdot - k) \mid k \in \mathbb{Z}\}}. \quad (3.24)$$

Während in Abschnitt 3.2.1 Wavelets über die Zulässigkeitsbedingung definiert werden konnten, ergeben sie sich hier aus den Unterschieden zwischen den Skalierungsräumen. Obwohl die Gültigkeit von Gleichung (3.24) einleuchtet, ist das Ergebnis keineswegs trivial und erfordert detaillierte Betrachtungen, die hier aber nicht wiedergegeben werden sollen. Stattdessen soll nur die Ähnlichkeit zur kontinuierlichen WT aufgezeigt werden: Ein Signal lässt sich durch eine entsprechende Linearkombination translaterter und dilatierter Wavelets darstellen. Die zugehörigen Waveleträume sind dabei orthogonal und somit koexistent, während im Unterschied dazu bei einer Darstellung durch Skalierungsfunktionen immer nur ein einzelner Raum Verwendung finden kann.<sup>3</sup> Trotz dieses wesentlichen Unterschieds bestehen weitere Ähnlichkeiten zwischen Skalierungsfunktionen und Wavelets.

Die Skalierungsfunktion  $\varphi$  erfüllt die *Skalierungsgleichung*<sup>4</sup>:

$$\exists (h_k)_{k \in \mathbb{Z}} \quad : \quad \varphi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \varphi(2x - k). \quad (3.25)$$

Für Wavelets gilt dann

$$\psi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} g_k \varphi(2x - k). \quad (3.26)$$

Zwischen den Skalierungskoeffizienten  $h_k$  und den Waveletkoeffizienten  $g_k$  gilt der Zusammenhang

$$g_k = (-1)^k h_{1-k}. \quad (3.27)$$

---

<sup>3</sup>Genau genommen ist dies kein angebarter Raum, da im allgemeinen der Grenzwert  $\lim_{m \rightarrow -\infty} \mathbf{V}_m$  gebildet werden muss. Gilt allerdings  $f \in \mathbf{V}_m$  für ein bestimmtes  $m$ , so ist die Aussage korrekt.

<sup>4</sup>Gleichung (3.25) wird in den meisten Fällen zur Bestimmung der Skalierungskoeffizienten  $h_k$  bei gegebener Skalierungsfunktion  $\varphi$  benutzt. Unter bestimmten Voraussetzungen ist auch der umgekehrte Weg durch Vorgabe der  $h_k$  möglich, vgl. [12].

Hier wird deutlich, wie aus einer Skalierungsfunktion ein Wavelet definiert werden kann. Es ist jedoch auch eine Definition mit einer geradzahlgigen Indexverschiebung zulässig:

$$g_k = (-1)^k h_{2l+1-k} , \quad l \in \mathbb{Z} . \quad (3.28)$$

Die Indexverschiebung bewirkt lediglich, dass ein anderes Translat aus  $\mathbf{W}_0$  ausgewählt wird, vgl. hierzu Anhang A.2.<sup>5</sup>

Wie später noch zu sehen ist, lassen sich sehr viele Berechnungen in der MSA unter Umgehung der kontinuierlichen Funktionen  $\varphi$  und  $\psi$  ausschließlich mit den Koeffizienten  $h_k$  und  $g_k$  ausdrücken. Bei diesen Berechnungen finden folgende Abkürzungen Verwendung:

$$\varphi_{m,k}(x) = 2^{-\frac{m}{2}} \varphi(2^{-m}x - k) , \quad (3.29)$$

$$\psi_{m,k}(x) = 2^{-\frac{m}{2}} \psi(2^{-m}x - k) , \quad (3.30)$$

$$m, k \in \mathbb{Z} . \quad (3.31)$$

Für ein festes  $m$  bilden die  $\varphi_{m,k}$  eine Orthonormalbasis (ONB) von  $\mathbf{V}_m$  und die  $\psi_{m,k}$  eine von  $\mathbf{W}_m$ . Die in den Gleichungen (3.17) und (3.18) definierten Projektionen lassen sich damit folgendermaßen darstellen:

$$\mathcal{R}_m f = \sum_{k \in \mathbb{Z}} c_k^m \varphi_{m,k} , \quad c_k^m = \langle f, \varphi_{m,k} \rangle_{\mathbf{L}_2} , \quad (3.32)$$

$$\mathcal{Q}_m f = \sum_{k \in \mathbb{Z}} d_k^m \psi_{m,k} , \quad d_k^m = \langle f, \psi_{m,k} \rangle_{\mathbf{L}_2} . \quad (3.33)$$

### 3.2.3 Skalierungsfunktionen und Wavelets

In den meisten Fällen werden Skalierungsfunktionen über ihre Fouriertransformierten definiert. Eine der wenigen Ausnahmen bildet ein Rechteckimpuls, die Haar-Skalierungsfunktion:

$$\varphi(x) = \begin{cases} 1 & , \quad 0 \leq x < 1 \\ 0 & , \quad \text{sonst} \end{cases} . \quad (3.34)$$

---

<sup>5</sup>Es lassen sich zu einer MSA auch andere Wavelets definieren, vgl. Korollar 2.2.13 und Bemerkung 2.2.14 in [12].

In die Skalierungsgleichung (3.25) eingesetzt ergibt sich unmittelbar

$$h_k = \begin{cases} \frac{1}{\sqrt{2}} & , k \in \{0, 1\} \\ 0 & , \text{sonst} \end{cases} . \quad (3.35)$$

Aus der Differenz  $\mathcal{R}_m f - \mathcal{R}_{m+1} f = \mathcal{Q}_{m+1} f$  folgt das zugehörige Wavelet:

$$\psi(x) = \begin{cases} 1 & , 0 \leq x < \frac{1}{2} \\ -1 & , \frac{1}{2} \leq x < 1 \\ 0 & , \text{sonst} \end{cases} , \quad \text{mit } g_k = \begin{cases} \frac{1}{\sqrt{2}} & , k = 0 \\ -\frac{1}{\sqrt{2}} & , k = 1 \\ 0 & , \text{sonst} \end{cases} . \quad (3.36)$$

Obwohl Rechteckfunktionen nur bedingt geeignet sind, glatte Signalverläufe zu approximieren, stellen sie durch ihre Einfachheit ein wichtiges Hilfsmittel dar. So lässt sich die FDTD ebenfalls über das Galerkin-Verfahren aus den Maxwell'schen Gleichungen herleiten, indem Rechteckfunktionen als Entwicklungsfunktionen benutzt werden. Ein weiterer Vorteil von Haar-Wavelets gegenüber anderen Wavelets stellt die geringe Anzahl der von Null verschiedenen Waveletkoeffizienten dar. Es wird später noch zu sehen sein, wie sehr der numerische Aufwand von der Anzahl der Koeffizienten abhängt.

Die bereits erwähnte Definition von Skalierungsfunktionen über ihre Fouriertransformierten lässt erahnen, dass es sich meistens um Funktionen mit unendlichem Träger handelt. Neben den Haar-Funktionen haben verschiedene von Ingrid Daubechies entwickelte Klassen von Skalierungsfunktionen nebst zugehörigen Wavelets einen kompakten Träger [13]. Die wohl bekannteste dieser Klassen<sup>6</sup> soll hier kurz wiedergegeben werden.

Auf die Herleitung der Daubechies-Wavelets wird hier verzichtet. Stattdessen sollen ihre Eigenschaften und einige Koeffizienten wiedergegeben werden. Haben die Daubechies-Skalierungsfunktion und das zugehörige Wavelet die Ordnung  $N \in \mathbb{N}$ , so gilt:

---

<sup>6</sup>In [13] wird diese Klasse mit den Eigenschaften „extremale Phase“ und „maximale Anzahl verschwindender Momente, kompatibel mit der Trägerbreite“ angegeben.

- Die Anzahl der Skalierungskoeffizienten ist  $2N$ .
- Die Differenzierbarkeitsordnung der Wavelets steigt mit wachsendem  $N$ , d. h. die Funktionen werden glatter.
- Die Wavelets haben  $N$  verschwindende Momente:

$$\int x^m \psi(x) dx = 0, \quad m \in \{0, 1, \dots, N-1\}. \quad (3.37)$$

- Im Grundraum  $\mathbf{V}_0$  liegen Polynome vom Grad  $m$  mit  $0 \leq m \leq N-1$ . Die folgende Gleichung gilt punktweise:

$$\exists (a_k^m)_{k \in \mathbb{Z}} : \sum_{k \in \mathbb{Z}} a_k^m \varphi(x-k) = x^m, \quad m \in \{0, 1, \dots, N-1\}. \quad (3.38)$$

- Die Trägerbreite der Funktionen ist  $2N-1$ . Genauer gesagt gilt

$$\text{supp}(\varphi) = [0, 2N-1], \quad \text{supp}(\psi) = [1-N, N]. \quad (3.39)$$

Für  $N=1$  und  $N=2$  lassen sich die von Null verschiedenen Skalierungskoeffizienten noch analytisch angeben:

$$N=1 : h_0 = h_1 = \frac{1}{\sqrt{2}}, \quad (3.40)$$

$$N=2 : h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \quad h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}. \quad (3.41)$$

Das Daubechies-Wavelet erster Ordnung ist somit identisch zu dem Haar-Wavelet. Ab der dritten Ordnung lassen sich die Koeffizienten nur noch numerisch bestimmen. Tabellierte Werte sind z. B. in [13] zu finden.<sup>7</sup>

Obwohl die Trägerbreite  $2N-1$  beträgt, scheinen die Funktionen mit steigender Ordnung einen deutlich kleineren Träger zu haben, wie in Abbildung 3.1 zu sehen ist.

---

<sup>7</sup>Je nach Literaturquelle liegen Koeffizienten schon mal in umgekehrter Reihenfolge vor. In Anhang A.1 wird dieser Punkt näher beleuchtet.

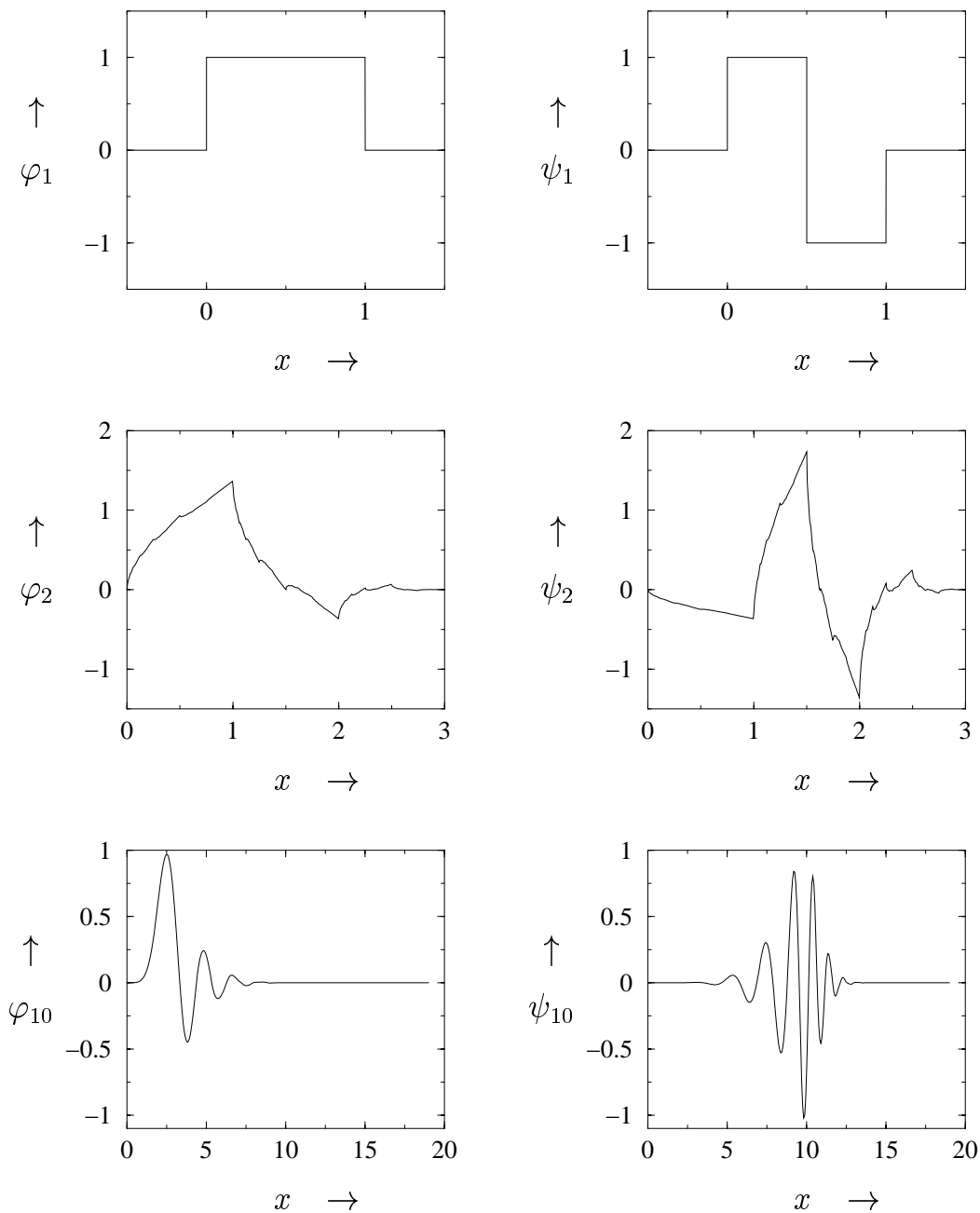


Abbildung 3.1: Daubechies-Funktionen der Ordnung  $N = 1, 2$  und  $10$  (Skalierungsfunktionen  $\varphi_N$ ; Translat eines Wavelets  $\psi_N$  mit gleichem Träger). Der Träger bei  $N = 1$  ist  $[0, 1[$ . Für  $N = 2, 10$  sind die Funktionen nur innerhalb ihres Trägers dargestellt.

### 3.2.4 Die schnelle Wavelet-Transformation

Die Multi-Skalen-Analyse erlaubt die Entwicklung einer effizienten diskreten Variante der kontinuierlichen Wavelet-Transformation. Die resul-

tierende schnelle Wavelet-Transformation (engl. *Fast Wavelet Transform*, FWT) ist dabei bezüglich der Ordnung ihrer Komplexität bzw. Berechnungsaufwandes sogar deutlich besser als die so genannte schnelle Fouriertransformation [12].

Die FWT erfordert die Approximation einer Funktion  $f$  im Grundraum  $\mathbf{V}_0$ , wodurch  $f$  im Allgemeinen diskretisiert wird. Dann wird  $f$  nach Gleichung (3.32) durch die Entwicklungskoeffizienten  $c_k^0$  repräsentiert:

$$f(x) = \sum_{k \in \mathbb{Z}} c_k^0 \varphi(x - k) . \quad (3.42)$$

Mit der FWT ist es möglich, in einfacher Weise von den Entwicklungskoeffizienten des Raumes  $\mathbf{V}_m$  zu den Koeffizienten der Räume  $\mathbf{V}_{m+1}$  und  $\mathbf{W}_{m+1}$  zu gelangen. Unter Verwendung der Gleichungen (3.32) und (3.33) gilt:

$$c_k^m = \sum_{l \in \mathbb{Z}} h_{l-2k} c_l^{m-1} , \quad (3.43)$$

$$d_k^m = \sum_{l \in \mathbb{Z}} g_{l-2k} c_l^{m-1} . \quad (3.44)$$

Nach insgesamt  $M$  Schritten wird  $f$  dann durch die Koeffizienten  $d_k^m$ ,  $m \in \{1, \dots, M\}$ , sowie  $c_k^M$  dargestellt, mit  $k \in \mathbb{Z}$ .

Die inverse FWT wird bei  $\mathbf{V}_M$ ,  $\mathbf{W}_M$  begonnen, um sukzessiv die Entwicklungskoeffizienten der Skalierungsräume bis  $\mathbf{V}_0$  zu berechnen:<sup>8</sup>

$$c_k^m = \sum_{l \in \mathbb{Z}} h_{k-2l} c_l^{m+1} + \sum_{l \in \mathbb{Z}} g_{k-2l} d_l^{m+1} . \quad (3.45)$$

### 3.2.5 Diskrete periodische Wavelet-Transformation

Die gerade beschriebene FWT ist mathematisch sehr einfach aufgebaut. In [14, 15] wird die Anwendung der FWT auf Filter untersucht, die mit Signalen endlicher Länge umgehen sollen, d. h. mit einer endlichen Anzahl

<sup>8</sup>Gleichung (3.45) entspricht Gleichung (2.3.4) in [12]. Allerdings befinden sich in letztgenannter sowie der darüber liegenden Gleichung Indexfehler, wie leicht nachgeprüft werden kann. Die Gleichungen nach Lemma 2.3.1 aus [12] sind jedoch korrekt.

von Entwicklungskoeffizienten im  $\mathbf{V}_0$ . Dabei treten einige algorithmische Schwierigkeiten bzw. „Unschönheiten“ auf, die hier jedoch nicht näher betrachtet werden sollen. Einen Ausweg hieraus bietet die periodisierte Form der FWT, die diskrete periodische Wavelet-Transformation (DPWT), wie sie auch z. B. in [16, 17] nachzulesen ist.

Beim Vergleich der Literaturstellen wird man feststellen, dass die Definitionen der DPWT uneinheitlich und oft sogar unvollständig sind. Daher soll hier versucht werden, eine möglichst vollständige und eindeutige Definition der DPWT aufzustellen. Dazu gelten folgende Voraussetzungen:

- Die Skalierungsfunktion hat einen kompakten Träger. Für die Skalierungskoeffizienten  $h_k$  lässt sich  $k$  eingrenzen auf das Intervall  $[0, 2N - 1]$ ,  $N \in \mathbb{N}$ . Außerhalb dieses Intervalls ist  $h_k$  identisch Null.
- Für das Wavelet  $\psi$  gilt  $g_k = (-1)^k h_{2N-1-k}$ ,  $k \in \{0, \dots, 2N - 1\}$ .
- Die Projektion einer Funktion  $f$  auf  $\mathbf{V}_0$  führt auf die Entwicklungskoeffizienten  $c_k^0$ .  $(c_k^0)_{k \in \mathbb{Z}}$  soll  $2^n$ -periodisch in  $k$  sein (ggf. durch periodische Fortsetzung), wobei  $n \in \mathbb{N}$  ist.
- Transformationen werden nur bis zum  $\mathbf{V}_t$ ,  $\mathbf{W}_t$  mit  $t \leq n$ ,  $t \in \mathbb{N}$  durchgeführt.

Die Transformation von  $\mathbf{V}_0$  nach  $\mathbf{V}_t$ ,  $\mathbf{W}_1, \dots, \mathbf{W}_t$  (mit den Zwischenergebnissen  $\mathbf{V}_1, \dots, \mathbf{V}_{t-1}$ ) erfolgt in  $t$  Schritten und analog wie bei der FWT. Es gilt für die Abbildungen  $\mathbf{V}_{m-1} \mapsto \mathbf{V}_m$  und  $\mathbf{V}_{m-1} \mapsto \mathbf{W}_m$

$$c_k^m = \sum_{l=0}^{2N-1} h_l c_{2k+l}^{m-1}, \quad d_k^m = \sum_{l=0}^{2N-1} g_l c_{2k+l}^{m-1}, \quad (3.46)$$

bzw.

$$c_k^m = \sum_{l=0}^{2N-1} h_l c_{(2k+l) \bmod 2^{n-(m-1)}}^{m-1}, \quad d_k^m = \sum_{l=0}^{2N-1} g_l c_{(2k+l) \bmod 2^{n-(m-1)}}^{m-1}, \quad (3.47)$$

mit

$$0 \leq k \leq 2^{n-m} - 1 \quad (3.48)$$

und

$$c_k^m, d_k^m : 2^{n-m}\text{-periodisch in } k. \quad (3.49)$$

Gleichung (3.46) ergibt sich durch eine Indexverschiebung in den Gleichungen (3.43) und (3.44) und anschließender Berücksichtigung der endlichen Anzahl von Skalierungskoeffizienten. Gleichung (3.47) verdeutlicht die Beschränkung auf eine einzelne Periode der Koeffizienten.

Die Rücktransformation von  $\mathbf{V}_t, \mathbf{W}_1, \dots, \mathbf{W}_t$  nach  $\mathbf{V}_0$  (wieder mit den Zwischenergebnissen  $\mathbf{V}_1, \dots, \mathbf{V}_{t-1}$ ) wird ebenfalls rekursiv durchgeführt:

$$c_k^m = \sum_{l=\lceil \frac{k-(2N-1)}{2} \rceil}^{\lfloor \frac{k}{2} \rfloor} (h_{k-2l} c_{l \bmod 2^{n-(m+1)}}^{m+1} + g_{k-2l} d_{l \bmod 2^{n-(m+1)}}^{m+1}) . \quad (3.50)$$

Diese Gleichung entspricht Gleichung (3.45), wobei die Summengrenzen an die endliche Anzahl von Skalierungs- und Waveletkoeffizienten angepasst wurden. Die Modulo-Operationen in den Indizes der Entwicklungskoeffizienten sind notwendig, um sich wieder auf eine Periode beschränken zu können.

Die meisten Wavelets haben keinen kompakten, sondern einen unendlichen Träger. Um sie dennoch in einer DPWT verwenden zu können, werden sie meistens periodisiert. Alternativ werden sie oft auch im Rahmen der jeweils erforderlichen Genauigkeit durch Einführung eines Schwellenwertes begrenzt.

Die Verwendung von Zweierpotenzen als Intervalllängen erfordert in vielen praktischen Fällen eine künstliche Erweiterung von Signalen bzw. Funktionen auf diese Längen – im ungünstigsten Fall sogar auf fast das Doppelte der eigentlichen Datenmenge. Dieser Nachteil lässt sich nur auf Kosten algorithmischer Einfachheit umgehen [14]. Der Vorteil der Verwendung von Zweierpotenzen wird besonders deutlich, wenn man sich die Anzahl  $A$  der Koeffizienten einer Periode anschaut: Wegen  $A(c_k^m) = A(d_k^m) = 2^{n-m}$  gilt

$$A(c_k^m) = A(c_k^{m+1}) + A(d_k^{m+1}) . \quad (3.51)$$

Diese Konstanz beim Übergang von einer Skala auf die nächste erlaubt eine besonders einfache Speicherverwaltung bei der Umsetzung in ein Computerprogramm.

Mit der Einführung einer weiteren Einschränkung lassen sich die Transformationen auch in Matrixform darstellen. Dazu wird die Abkürzung

$$L \stackrel{\text{def}}{=} 2^{n-m} \tag{3.52}$$

eingeführt. Im Folgenden soll dann stets

$$L \geq 2N \tag{3.53}$$

gelten. Die Entwicklungskoeffizienten werden in Vektoren zusammengefasst:

$$\mathbf{c}^m = (c_0^m, \dots, c_{L-1}^m)^T, \quad \mathbf{d}^m = (d_0^m, \dots, d_{L-1}^m)^T. \tag{3.54}$$

Dann soll die quadratische  $L \times L$ -Matrix  $\mathbf{P}_{m,m+1}^{L \times L}$  mit

$$\begin{pmatrix} \mathbf{c}^{m+1} \\ \mathbf{d}^{m+1} \end{pmatrix} = \mathbf{P}_{m,m+1}^{L \times L} \cdot \mathbf{c}^m \tag{3.55}$$

eine DPWT von Skala  $m$  nach  $m + 1$  bewirken. Der Aufbau von  $\mathbf{P}_{m,m+1}^{L \times L}$  kann Gleichung (3.47) entnommen werden:

$$\mathbf{P}_{m,m+1}^{L \times L} = \begin{pmatrix} h_0 & h_1 & h_2 & \dots & h_{2N-1} & 0 & 0 & \dots & 0 \\ 0 & 0 & h_0 & \dots & \dots & h_{2N-1} & 0 & \dots & 0 \\ \vdots & & & & \ddots & & & & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & h_{2N-1} \\ h_{2N-2} & h_{2N-1} & 0 & \dots & \dots & \dots & \dots & \dots & h_{2N-3} \\ \vdots & & & & \ddots & & & & \vdots \\ h_2 & \dots & \dots & \dots & \dots & \dots & 0 & h_0 & h_1 \\ g_0 & g_1 & g_2 & \dots & g_{2N-1} & 0 & 0 & \dots & 0 \\ 0 & 0 & g_0 & \dots & \dots & g_{2N-1} & 0 & \dots & 0 \\ \vdots & & & & \ddots & & & & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & g_{2N-1} \\ g_{2N-2} & g_{2N-1} & 0 & \dots & \dots & \dots & \dots & \dots & g_{2N-3} \\ \vdots & & & & \ddots & & & & \vdots \\ g_2 & \dots & \dots & \dots & \dots & \dots & 0 & g_0 & g_1 \end{pmatrix} \tag{3.56}$$

Die Rücktransformation von Skala  $m + 1$  nach Skala  $m$  ist mit der Matrix  $\mathbf{P}_{m+1,m}^{*L \times L}$  möglich:

$$\mathbf{c}^m = \mathbf{P}_{m+1,m}^{*L \times L} \cdot \begin{pmatrix} \mathbf{c}^{m+1} \\ \mathbf{d}^{m+1} \end{pmatrix}. \quad (3.57)$$

Für  $\mathbf{P}_{m+1,m}^{*L \times L}$  gilt nach [17]

$$\mathbf{P}_{m+1,m}^{*L \times L} = (\mathbf{P}_{m,m+1}^{L \times L})^{-1} = (\mathbf{P}_{m,m+1}^{L \times L})^T, \quad (3.58)$$

d. h.  $\mathbf{P}_{m,m+1}^{L \times L}$  ist orthogonal.

Der Aufbau von  $\mathbf{P}_{m,m+1}^{L \times L}$  nach Gleichung (3.56) liefert auch gleich die Erklärung für die Einschränkung  $L \geq 2N$  nach Gleichung (3.53): Die Anzahl der Skalierungskoeffizienten ( $2N$ ) darf nicht die Periodenlänge der Koeffizienten bzw. die Kantenlänge der Matrix ( $L$ ) überschreiten. Dadurch folgt allerdings auch, dass i. A. weniger Transformationsschritte durchgeführt werden dürfen als mit den allgemeineren Summenformeln (3.46)-(3.50): Es ist wieder  $t$  die maximal mögliche Transformationstiefe. Dann wird mit  $t = m + 1$  Gleichung (3.53) ausgewertet:

$$2^{n-(t-1)} \geq 2N \quad (3.59)$$

$$\Leftrightarrow 2^t \leq \frac{2^{n+1}}{2N} = \frac{2^n}{N} \quad (3.60)$$

$$\Leftrightarrow t \leq n - \text{ld } N \quad (3.61)$$

$$\Rightarrow t \leq n - \lceil \text{ld } N \rceil. \quad (3.62)$$

Selbstverständlich muss auch noch  $n \geq \lceil \text{ld } N \rceil$  gelten. Da abgesehen vom Haar-Wavelet immer  $N > 1$  gilt, folgt bei der Verwendung der gezeigten Matrixform  $t < n$  im Gegensatz zu  $t \leq n$  bei den Summenformeln.

Aus den Matrizen für einzelne Transformationsstufen lässt sich jetzt eine Matrix  $\mathbf{P}_{0,j}$  für eine vollständige Abbildung  $\mathbf{V}_0 \mapsto \{\mathbf{W}_1, \dots, \mathbf{W}_j, \mathbf{V}_j\}$ ,  $j \leq t$ , angeben. Dazu wird die Erweiterung von  $\mathbf{P}_{m,m+1}^{L \times L}$  um die Identitätsabbildung für die jeweiligen Räume  $\mathbf{W}_1, \dots, \mathbf{W}_m$  benötigt:

$$\tilde{\mathbf{P}}_{m,m+1} = \mathbf{P}_{m,m+1}^{L \times L} \oplus \mathbf{I}^{\bar{L}}, \quad \bar{L} = 2^n - L. \quad (3.63)$$

Dann erfüllt

$$\mathbf{P}_{0,j} = \prod_{m=j-1}^0 \tilde{\mathbf{P}}_{m,m+1} = \prod_{i=1}^j \tilde{\mathbf{P}}_{j-i,j-i+1} \quad (3.64)$$

die Abbildung

$$\mathbf{v}^j = \mathbf{P}_{0,j} \cdot \mathbf{c}^0 \quad \text{mit} \quad \mathbf{v}^j = \begin{pmatrix} \mathbf{c}^j \\ \mathbf{d}^j \\ \mathbf{d}^{j-1} \\ \vdots \\ \mathbf{d}^1 \end{pmatrix}. \quad (3.65)$$

Mit Hilfe der Rechenregel  $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$  für Matrizen folgt für die inverse Abbildung

$$\mathbf{P}_{j,0}^* = (\mathbf{P}_{0,j})^{-1} = \prod_{m=0}^{j-1} (\tilde{\mathbf{P}}_{m,m+1})^T. \quad (3.66)$$

Wegen  $\mathbf{B}^T \mathbf{A}^T = (\mathbf{AB})^T$  folgt ferner  $\mathbf{P}_{j,0}^* = (\mathbf{P}_{0,j})^T$  und damit die Orthogonalität der Gesamttransformation:

$$\mathbf{P}_{j,0}^* = (\mathbf{P}_{0,j})^{-1} = (\mathbf{P}_{0,j})^T. \quad (3.67)$$

Zur Vereinfachung wird in den folgenden Kapiteln zwischen „reinen“ WT-Matrizen ( $\mathbf{P}_{a,b}^{L \times L}$ ) und erweiterten Matrizen ( $\tilde{\mathbf{P}}_{a,b}$ ) nur bei gegebener Notwendigkeit unterschieden werden. Meistens wird daher das Symbol  $\mathbf{P}_{a,b}$  Verwendung finden, das je nach Kontext einem der genannten Fälle entspricht.



# Kapitel 4

## Mehrdimensionale Basen und Skalen

Bisher wurden immer nur skalare Funktionen bzw. Signale im eindimensionalen Raum betrachtet, d. h. Abbildungen der Form  $f : \mathbb{R} \mapsto \mathbb{R}$ . In diesem Kapitel werden verschiedene mehrdimensionale Basen diskutiert. Dies bildet die Grundlage für die Darstellung von Operatoren in den unterschiedlichen Basen. Produkte von Räumen sind dabei immer im kartesischen Sinn zu verstehen, d. h.  $\mathbf{VW}$  ist gleichbedeutend mit  $\mathbf{V} \times \mathbf{W}$ .

### 4.1 Homogene Tensorprodukt-Basen

In Abschnitt 3.2.3 wurde erwähnt, dass sich die FDTD aus einem Galerkin-Verfahren herleiten lässt. Dabei werden eindimensionale Rechteckfunktionen

$$r_l(x) = \begin{cases} 1 & , \quad -\frac{1}{2} \leq x - l < \frac{1}{2} \\ 0 & , \quad \text{sonst} \end{cases} \quad (4.1)$$

benutzt. Ein kontinuierliches elektrisches Feld kann nach Gleichung (2.14) dann auch folgendermaßen approximiert werden:

$$\mathbf{E}(x, y, z, (n + \frac{1}{2})\Delta t) \approx \sum_{i,j,k} \mathbf{E}^{n+\frac{1}{2}}(i, j, k) r_i(x) r_j(y) r_k(z) . \quad (4.2)$$

Die Rechteckfunktionen bilden damit über ein Tensorprodukt eine Basis im  $\mathbb{R}^3$ .

Das Konstruktionsprinzip für diese Art mehrdimensionaler Basen lässt sich allgemeiner formulieren: Für jede Koordinate  $i$  spannen die Funktionen  $\xi_k^i : \mathbb{R} \mapsto \mathbb{R}$  einen eindimensionalen Funktionenraum  $\Xi^i$  auf, d. h.  $\Xi^i = \text{span}\{\xi_k^i(\cdot) \mid k \in \mathbb{Z}\}$ . Das Tensorprodukt

$$\theta_{\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^n \xi_{k_i}^i(x_i), \quad \theta_{\mathbf{k}} : \mathbb{R}^n \mapsto \mathbb{R}, \quad (4.3)$$

mit

$$\mathbf{x} = (x_1, \dots, x_n), \quad \mathbf{k} = (k_1, \dots, k_n), \quad (4.4)$$

ergibt dann eine homogene Basis für den Funktionenraum

$$\Theta = \text{span}\{\theta_{\mathbf{k}}(\cdot) \mid \mathbf{k} \in \mathbb{Z}^n\}. \quad (4.5)$$

Anschaulicher gesprochen wird der Raum  $\Theta$  aus sämtlichen Permutationen des Produktes  $\xi_{k_1}^1 \xi_{k_2}^2 \dots \xi_{k_n}^n$ ,  $\mathbf{k} \in \mathbb{Z}^n$ , gebildet. Für eine Funktion  $f \in \Theta$  soll es dann eine eindeutige Darstellung

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^n} a_{\mathbf{k}} \theta_{\mathbf{k}}(\mathbf{x}) \quad (4.6)$$

geben, wobei die  $a_{\mathbf{k}}$  Entwicklungskoeffizienten sind.

In den meisten praktischen Fällen sind nur endliche Volumina relevant. Gilt somit für die Koordinate  $i$  die Beschränkung auf das Intervall  $[x_{i,1}, x_{i,2}]$ , soll entsprechend  $k_{i,1} \leq k_i \leq k_{i,2}$  gelten. Die Entwicklungskoeffizienten sind dadurch in einem  $n$ -dimensionalen Quader angeordnet, unabhängig vom zu Grunde liegenden Koordinatensystem. Im Falle der (klassischen) FDTD gibt es sogar eine sehr direkte Zuordnung zwischen dem Indexvektor und der Position im Raum.

Eine fundamentale Eigenschaft des homogenen Tensorproduktes ist, dass keine Abhängigkeiten zwischen den Koordinaten bestehen. Transformationen entlang einer Koordinatenachse können daher unabhängig von den anderen Koordinatenachsen durchgeführt werden.

Es soll jetzt  $\Theta = \mathbf{V}_0^n$  gelten, d. h. es wird das Tensorprodukt der Skalierungsfunktionen des Grundraumes benutzt ( $\xi_{k_i}^i(\cdot) = \varphi(\cdot - k_i)$ ). Mittels einer  $n$ -fachen eindimensionalen WT kann eine Funktion  $f \in \mathbf{V}_0^n$  durch eindimensionale Wavelets und Skalierungsfunktionen höherer Skalen dargestellt werden.

Die Grundlage hierfür bildet die Tatsache, dass  $\mathbf{V}_0 = \mathbf{W}_1 \oplus \dots \oplus \mathbf{W}_j \oplus \mathbf{V}_j$  gilt. Dadurch ist jedes  $\xi_{k_i}^i$  ein Wavelet aus einer der Skalen 1 bis  $j$  oder eine Skalierungsfunktion der Skala  $j$ . Die Unabhängigkeit der Koordinaten voneinander erlaubt die Anwendung der eindimensionalen DPWT nach Gleichung (3.65). Das damit verbundene Ordnungsschema sieht im Falle zweier Dimensionen und dreier Transformationsstufen wie in Abbildung 4.1 gezeigt aus.

Die schematische Darstellung zeigt gleichzeitig mit der erwähnten Permutation von Basisfunktionen auch, dass unterschiedliche Skalen miteinander vermischt werden. Damit verlässt man nach [12] das Gebiet der MSA, die auch in mehreren Dimensionen auf der Definition einer Skalierungsgleichung basiert (vgl. nächsten Abschnitt). Dennoch wird diese Art der Transformation wegen ihrer Einfachheit häufig benutzt, um auch bei mehrdimensionalen Signalen mit Entwicklungsfunktionen verschiedener Auflösungsstufen Datenkompression zu ermöglichen. Eine weitere Eigenschaft ist, dass für die einzelnen Richtungen verschiedene Transformationstiefen verwendet werden können, weshalb man hier auch von einer mehrdimensionalen Skala sprechen kann.

## 4.2 Separable MSA-Basen

Die Definition der mehrdimensionalen MSA ist der eindimensionalen sehr ähnlich. Im Wesentlichen werden der skalare Dilatationsfaktor durch eine Matrix und die skalare Translation durch einen Vektor ersetzt. Die Skalierungseigenschaft ergibt sich aus

$$f(\cdot) \in \mathbf{V}_m \iff f(\mathbf{A}\cdot) \in \mathbf{V}_{m-1} . \quad (4.7)$$

Während im vorherigen Abschnitt noch  $\varphi : \mathbb{R} \mapsto \mathbb{R}$  und  $\varphi \in \mathbf{V}_0 \subset \mathbf{L}_2(\mathbb{R})$  galt, treten hier die Räume und Funktionen direkt für  $n$  Dimensionen

$\longrightarrow x, k_x$

	$\mathbf{V}_3 \mathbf{V}_3$	$\mathbf{W}_3 \mathbf{V}_3$	$\mathbf{W}_2 \mathbf{V}_3$	$\mathbf{W}_1 \mathbf{V}_3$
	$\mathbf{V}_3 \mathbf{W}_3$	$\mathbf{W}_3 \mathbf{W}_3$	$\mathbf{W}_2 \mathbf{W}_3$	$\mathbf{W}_1 \mathbf{W}_3$
	$\mathbf{V}_3 \mathbf{W}_2$	$\mathbf{W}_3 \mathbf{W}_2$	$\mathbf{W}_2 \mathbf{W}_2$	$\mathbf{W}_1 \mathbf{W}_2$
$\downarrow$ $y, k_y$	$\mathbf{V}_3 \mathbf{W}_1$	$\mathbf{W}_3 \mathbf{W}_1$	$\mathbf{W}_2 \mathbf{W}_1$	$\mathbf{W}_1 \mathbf{W}_1$

Abbildung 4.1: Ordnungsschema nach einer zweidimensionalen DPWT für homogene Tensorprodukt-Basen. Bei mehr als einer Transformationsstufe (hier: drei) vermischen sich die Skalen.

auf, d. h.  $\varphi : \mathbb{R}^n \mapsto \mathbb{R}$ ,  $\varphi \in \mathbf{V}_0 \subset \mathbf{L}_2(\mathbb{R}^n)$ . Der Grundraum  $\mathbf{V}_0$  wird folgendermaßen durch die Skalierungsfunktion aufgespannt:

$$\mathbf{V}_0 = \overline{\text{span}\{\varphi(\cdot - \mathbf{k}) \mid \mathbf{k} \in \mathbb{Z}^n\}} . \quad (4.8)$$

An die Dilatationsmatrix  $\mathbf{A}$  werden die Anforderungen gestellt, dass ihre Eigenwerte alle größer als Eins sind und dass sie nur ganzzahlige Elemente enthält. Es kann dann gezeigt werden, dass genau  $|\det \mathbf{A}| - 1$  Wavelets zu der MSA existieren, die das orthogonale Komplement zu  $\mathbf{V}_0$  in  $\mathbf{V}_{-1}$  erzeugen.

Die separablen Skalierungsfunktionen, um die es hier gehen soll, werden wie im vorhergehenden Abschnitt über ein Tensorprodukt eindimensionaler Skalierungsfunktionen definiert, die bezüglich der Koordinatenachsen voneinander verschieden sein dürfen ( $\varphi^i \in \mathbf{V}_0^{(i)}$ ,  $i = 1, \dots, n$ ). Die zu der jeweiligen eindimensionalen MSA gehörenden Wavelets werden mit  $\psi^i \in \mathbf{W}_0^{(i)}$  bezeichnet. Die erzeugende Skalierungsfunktion ist dann

$$\varphi(\mathbf{x}) = \prod_{i=1}^n \varphi^i(x_i) . \quad (4.9)$$

Zusammen mit der Dilatationsmatrix

$$\mathbf{A} = \text{diag}(\underbrace{2, \dots, 2}_n) \quad (4.10)$$

lassen sich die zugehörigen erzeugenden separablen Wavelets sehr einfach angeben: Es sind alle Permutationen des Produktes  $\xi^1 \dots \xi^n$  mit  $\xi^i \in \{\varphi^i, \psi^i\}$  durchzugehen, wobei jedoch immer wenigstens ein Wavelet benutzt werden muss. Im zweidimensionalen Fall mit gleicher eindimensionaler Skalierungsfunktion für alle Richtungen lauten die erzeugende Skalierungsfunktion

$$\varphi(x_1)\varphi(x_2) \quad (4.11)$$

und die zugehörigen Wavelets

$$\psi(x_1)\psi(x_2) , \quad \psi(x_1)\varphi(x_2) , \quad \varphi(x_1)\psi(x_2) . \quad (4.12)$$

Dieses Ergebnis lässt sich auch anhand einer dreistufigen DPWT für zwei Dimensionen in Abbildung 4.2 wiederfinden. Im Vergleich mit Abbildung 4.1 wird deutlich, dass mit jedem Transformationsschritt nur der jeweils verbliebene zweidimensionale Skalierungsraum transformiert wird. Es kann hier zwar ebenfalls die eindimensionale DPWT Anwendung finden, doch die Vorgehensweise ist eine andere: Für homogene Tensorproduktbasen kann für jede Richtung direkt eine mehrstufige eindimensionale DPWT durchgeführt werden. Bei separablen MSA-Basen dagegen muss eine einstufige DPWT in allen Richtungen innerhalb des aktuellen mehrdimensionalen Skalierungsraumes durchgeführt werden, bevor in der nächsten Skala eine DPWT begonnen werden kann. Außerdem ist die Transformationstiefe vom Prinzip her in allen Richtungen gleich groß.

Soll ein mehrdimensionales Signal lediglich um eine Stufe transformiert werden, d. h.  $\mathbf{V}_0 \mapsto \mathbf{V}_1 \oplus \mathbf{W}_1$ , so ist die WT für separable MSA-Basen nach dem oben Gesagten identisch zu der für homogene Tensorprodukt-Basen. Dies gilt zumindest für die hier vorgestellte Art separabler Wavelets. Weitere separable MSA-Basen sollen hier nicht näher betrachtet werden.

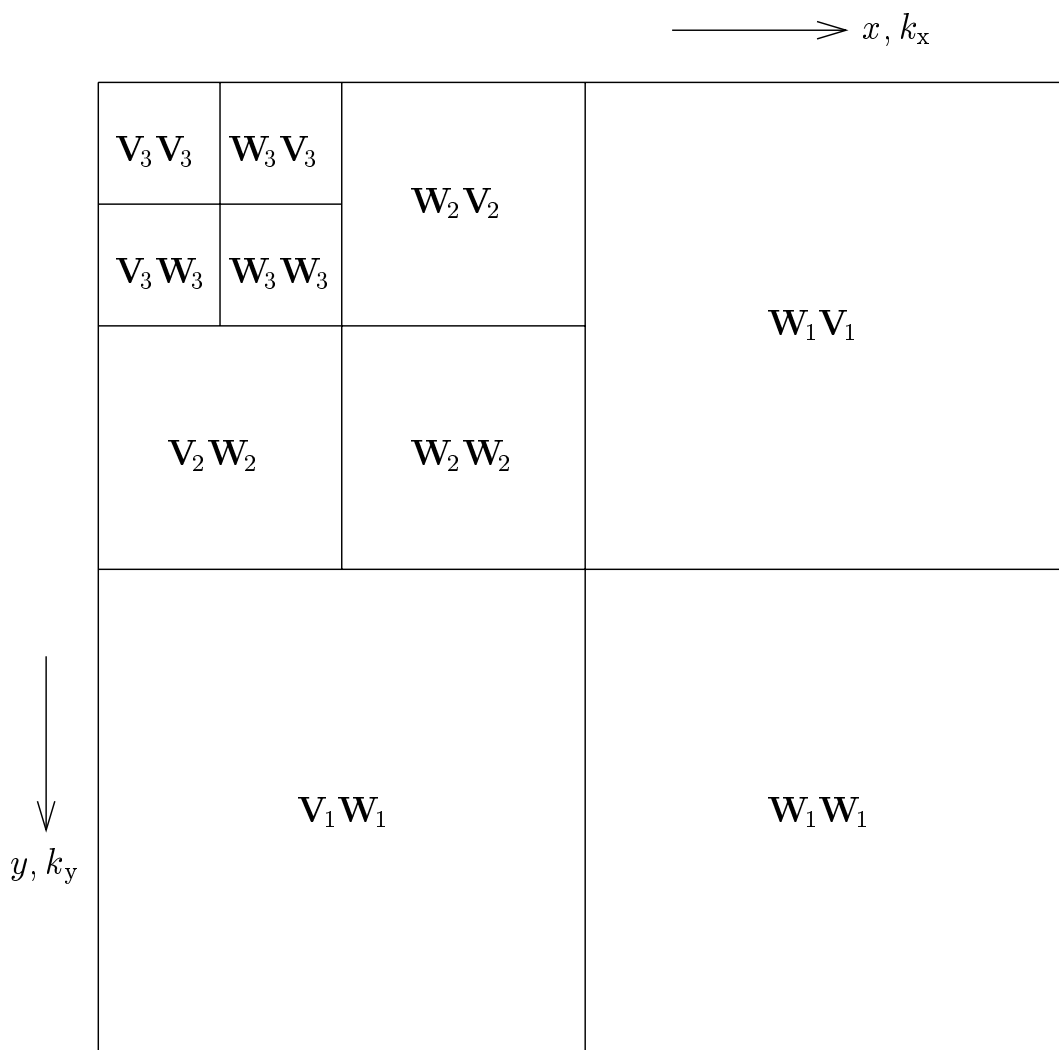


Abbildung 4.2: Ordnungsschema nach einer zweidimensionalen dreistufigen DPWT für separable MSA-Basen zu Gleichung (4.10). Drei Teilräume  $\mathbf{V}_i \mathbf{W}_i$ ,  $\mathbf{W}_i \mathbf{W}_i$ ,  $\mathbf{W}_i \mathbf{V}_i$  ergeben zusammen den vollständigen zweidimensionalen Waveletraum der Skala  $i$ .

## 4.3 Allgemeine MSA-Basen

Die Entwicklung von Wavelets mit bestimmten gewünschten Eigenschaften ist im mehrdimensionalen Fall sehr schwierig. Schnell kommt es bei nicht-separablen MSA-Basen zu nicht-differenzierbaren Funktionen. Im Allgemeinen lässt sich auch die mehrdimensionale DPWT nicht mehr auf eindimensionale Transformationen zurückführen. Verschiedene Gründe sprechen jedoch dafür, die Entwicklung nicht-separabler MSA-Basen im Auge zu behalten: Die vorgestellten separablen Wavelets weisen „Vorzugsrichtungen“ und damit anisotrope Eigenschaften auf. Datenkompression lässt sich in der Regel aber mit isotropen Funktionen besser durchführen. Dies steht auch im Zusammenhang mit der Tatsache, dass die Anzahl der beteiligten Waveleträume  $|\det \mathbf{A}| - 1$  und somit meistens größer als eins ist. Nicht-separable MSA-Basen erlauben durch entsprechende Wahl der Dilatationsmatrix mit weniger Funktionsräumen auszukommen.

Obwohl die Konstruktion dieser Basen mit einigen Schwierigkeiten verbunden ist, lässt sich ganz analog zum eindimensionalen Fall eine schnelle Wavelet-Transformation angeben. Auf weitere Einzelheiten zu solchen Basen soll hier aber nicht weiter eingegangen werden. In [12] können ein Einstieg in die Thematik und einige Beispiele für den zweidimensionalen Fall nachgelesen werden.



# Kapitel 5

## Wavelet-Transformation von Operatoren

Im letzten Kapitel wurde gezeigt, wie mehrdimensionale Funktionen in verschiedene Auflösungsstufen zerlegt werden können. Mit einfachen Schwellwertverfahren lassen sich diese Signaldarstellungen dann oft stark komprimieren. In diesem Kapitel soll nun untersucht werden, wie Operatoren transformiert werden können, sodass sie direkt auf transformierte Signale angewendet werden können. Die Ergebnisse hieraus bilden dann die Grundlage für die Transformation der FDTD in den Waveletbereich.

Abschnitt 5.1 stellt einige grundlegende Gleichungen und Beispiele vor. Ferner werden neuartige Notationen entwickelt, welche vor allem in Abschnitt 5.2 benötigt werden. Dort wird die WT von Operatoren mit Hilfe von Matrizen präsentiert und im Detail untersucht. In Abschnitt 5.3 wird im Gegensatz dazu ein vollkommen anderer Ansatz entwickelt, der von Matrizen keinerlei Gebrauch macht und eine sehr schematische Vorgehensweise bei Problemstellungen mit beliebig vielen Dimensionen erlaubt.

### 5.1 Grundlagen

Einige wesentliche Gleichungen zur WT von Feldern und Operatoren werden in den Abschnitten 5.1.1 bis 5.1.3 aufgestellt. Um mehrdimensionale Problemstellungen besser untersuchen zu können, wurden im Verlauf

dieser Forschungsarbeit spezielle Notationen entwickelt, die in Abschnitt 5.1.4 erläutert werden. Diese Notationen bilden die Grundlage für eine detaillierte Analyse der Matrix-basierten WT von Operatoren, welche in Abschnitt 5.2 vorgestellt wird.

### 5.1.1 Diskrete Darstellung von Operatoren

Es sei ein Funktionensystem gegeben, das einen Raum aufspannt:

$$\theta_{\mathbf{k}} : \mathbb{R}^n \mapsto \mathbb{R} , \quad (5.1)$$

$$\Theta = \text{span}\{\theta_{\mathbf{k}}(\cdot) \mid \mathbf{k} \in \mathbb{Z}^n\} . \quad (5.2)$$

Eine Funktion  $f \in \Theta$  lässt sich dann darstellen durch

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^n} a_{\mathbf{k}} \theta_{\mathbf{k}}(\mathbf{x}) , \quad \mathbf{x} \in \mathbb{R}^n . \quad (5.3)$$

Die Entwicklungskoeffizienten bilden eine Menge  $C$ :

$$a_{\mathbf{k}} : \mathbb{Z}^n \mapsto \mathbb{R} , \quad (5.4)$$

$$C = \{a_{\mathbf{k}} \mid \mathbf{k} \in \mathbb{Z}^n\} . \quad (5.5)$$

Jetzt sei mit  $\mathcal{T} : \Theta \mapsto \Theta'$  ein linearer Operator gegeben, sodass gilt

$$g(\mathbf{x}) = \mathcal{T}f(\mathbf{x}) = \mathcal{T}\left(\sum_{\mathbf{k}} a_{\mathbf{k}} \theta_{\mathbf{k}}(\mathbf{x})\right) = \sum_{\mathbf{k}} a_{\mathbf{k}} \mathcal{T} \theta_{\mathbf{k}}(\mathbf{x}) \quad (5.6)$$

sowie

$$g(\mathbf{x}) = \sum_{\mathbf{k}} b_{\mathbf{k}} \theta'_{\mathbf{k}}(\mathbf{x}) \quad \text{mit} \quad \theta'_{\mathbf{k}}(\mathbf{x}) \in \Theta' , \quad b_{\mathbf{k}} \in C' . \quad (5.7)$$

Obwohl i. A. verschiedene Funktionenräume für  $f$  und  $g$  vorliegen, lassen sich die weiteren Betrachtungen wegen der diskreten Entwicklung der Funktionen auf die Transformation der Entwicklungskoeffizienten beschränken. Dadurch kann von der kontinuierlichen Abbildung  $\mathcal{T} : \Theta \mapsto \Theta'$  auf die diskrete Abbildung  $T : C \mapsto C'$  übergegangen werden.  $T$  wird dann dargestellt durch eine  $2n$ -dimensionale Relation, die durch

$$b_{\mathbf{k}} = \sum_{\mathbf{k}'} t_{\mathbf{k},\mathbf{k}'} a_{\mathbf{k}'} \quad (5.8)$$

beschrieben werden kann. Die direkte Implementierung von Gleichung (5.8) ist mitunter problematisch. Nehmen wir der Einfachheit halber an, dass in jeder Richtung  $L$  Stützstellen für die Funktion im Bild- und Urbildbereich vorliegen. Dann gehören zu dem Operator  $\mathcal{T}$  im ungünstigsten Fall  $(L^n)^2 = L^{2n}$  Tensorelemente  $t_{\mathbf{k},\mathbf{k}'}$ . Die Grenzen eines Computerspeichers sind damit schnell überschritten. In der Praxis erlauben jedoch viele Operatoren eine effizientere Darstellung, sodass man oftmals mit wenigen Matrizen der Kantenlänge  $L$  ( $L^2$  Elemente) auskommt, wie noch zu sehen ist.

### 5.1.2 Transformation von Matrix-Operatoren

Im Fall  $n = 1$  lässt sich Gleichung (5.8) als Matrix-Gleichung  $\mathbf{b} = \mathbf{T} \cdot \mathbf{a}$  schreiben. Für die zu  $\mathbf{a}$  gehörende Funktion  $f$  gelte  $f \in \mathbf{V}_0$ . Dann lautet die Transformation von

$$g = \mathcal{T}f \quad (5.9)$$

in den Waveletbereich mittels einer DPWT

$$\mathcal{P}_{0,j}g = (\mathcal{P}_{0,j} \mathcal{T} \mathcal{P}_{j,0}^*)(\mathcal{P}_{0,j}f) . \quad (5.10)$$

Die mit  $\mathcal{P}_{0,j}g$  und  $\mathcal{P}_{0,j}f$  verbundene Transformation der Koeffizientenvektoren  $\mathbf{a}$  und  $\mathbf{b}$  ist nach Gleichung (3.65) durch Matrix-Vektor-Multiplikationen auszuführen. Analog bedeutet das für den transformierten Operator  $\mathcal{P}_{0,j} \mathcal{T} \mathcal{P}_{j,0}^*$  die Auswertung der Matrixmultiplikation

$$\tilde{\mathbf{T}} = \mathbf{P}_{0,j} \mathbf{T} \mathbf{P}_{j,0}^* = \mathbf{P}_{0,j} \mathbf{T} (\mathbf{P}_{0,j})^T . \quad (5.11)$$

Fasst man die Matrix  $\mathbf{T}$  als nebeneinander aufgereihte Spaltenvektoren bzw. als untereinander aufgereihte Zeilenvektoren auf, so ist zu erkennen, dass die Transformation von  $\mathbf{T}$  in den Waveletbereich durch eine spalten- und zeilenweise DPWT erfolgt.

### 5.1.3 Eindimensionale WT von Operatoren

Die Erweiterung von Gleichung (5.11) auf Abbildungen mit einer Darstellung wie in Gleichung (5.8) ist einfach, allerdings auch etwas unübersichtlich. Es wird die eindimensionale WT in Richtung  $\nu$  von Gleichung

(5.9) betrachtet, wobei  $n \geq 1$  sein darf. Die zu  $\mathcal{P}$  und  $\mathcal{P}^*$  gehörenden Tensorelemente seien  $p_{\mathbf{k},\mathbf{k}'}$  und  $p_{\mathbf{k},\mathbf{k}'}^*$ , die transformierte Gleichung laute

$$\tilde{g} = \tilde{\mathcal{T}} \tilde{f} . \quad (5.12)$$

Dann lässt sich zu

$$\tilde{g} = \mathcal{P}_{0,j_\nu}^\nu g = \mathcal{P}_{0,j_\nu}^\nu \mathcal{T} f \quad (5.13)$$

der folgende Zusammenhang der Entwicklungskoeffizienten angeben:

$$b_{\mathbf{k}^1}^2 = \sum_{\mathbf{k}^2} p_{\mathbf{k}^1,\mathbf{k}^2}^{0,j_\nu} a_{\mathbf{k}^2}^2 = \sum_{\mathbf{k}^2} p_{\mathbf{k}^1,\mathbf{k}^2}^{0,j_\nu} \sum_{\mathbf{k}^3} t_{\mathbf{k}^2,\mathbf{k}^3} a_{\mathbf{k}^3}^1 , \quad (5.14)$$

d. h.  $b_{\mathbf{k}^1}^2$  repräsentiert  $\tilde{g}$  und  $a_{\mathbf{k}^3}^1$  entsprechend  $f$ . Weiter gilt

$$\tilde{g} = \underbrace{\mathcal{P}_{0,j_\nu}^\nu \mathcal{T} \mathcal{P}_{j_\nu,0}^{*\nu}}_{\tilde{\mathcal{T}}} \tilde{f} \quad (5.15)$$

bzw.

$$b_{\mathbf{k}^1}^2 = \sum_{\mathbf{k}^2} p_{\mathbf{k}^1,\mathbf{k}^2}^{0,j_\nu} \sum_{\mathbf{k}^3} t_{\mathbf{k}^2,\mathbf{k}^3} \sum_{\mathbf{k}^4} p_{\mathbf{k}^3,\mathbf{k}^4}^{*j_\nu,0} b_{\mathbf{k}^4}^1 \quad (5.16)$$

mit

$$\tilde{f} = \mathcal{P}_{0,j_\nu}^\nu f \quad \text{bzw.} \quad b_{\mathbf{k}^4}^1 = \sum_{\mathbf{k}^5} p_{\mathbf{k}^4,\mathbf{k}^5}^{0,j_\nu} a_{\mathbf{k}^5}^1 . \quad (5.17)$$

Aus Gleichung (5.16) ist zu erkennen, wie Operator  $\tilde{\mathcal{T}}$  als Tensor zu berechnen ist:

$$\tilde{t}_{\mathbf{k}^1,\mathbf{k}^4} = \sum_{\mathbf{k}^2} p_{\mathbf{k}^1,\mathbf{k}^2}^{0,j_\nu} \sum_{\mathbf{k}^3} t_{\mathbf{k}^2,\mathbf{k}^3} p_{\mathbf{k}^3,\mathbf{k}^4}^{*j_\nu,0} . \quad (5.18)$$

Genau genommen ist die Einschränkung der WT auf nur eine Dimension für die Herleitung von Gleichung (5.18) nicht notwendig. In den weiteren Abschnitten trägt diese Vereinfachung jedoch zur Übersichtlichkeit bei.

### 5.1.4 Notationen

Für die folgenden Betrachtungen sind einige Notationen hilfreich, die den Umgang mit Vektoren und die Klassifizierung von Operatoren betreffen.

**Definition 1.** Es sei  $\mathbf{k} \in \mathbb{Z}^n$ ,  $\mathbf{k} = \sum_{i=1}^n k_i \mathbf{e}_i = (k_1, \dots, k_n)^T$ . Dann bewirkt das Symbol  $\leftarrow$  den Ersatz einer oder mehrerer Komponenten:

$$\mathbf{k} \leftarrow c \mathbf{e}_\nu \stackrel{\text{def}}{=} \sum_{\substack{i=1 \\ i \neq \nu}}^n k_i \mathbf{e}_i + c \mathbf{e}_\nu = (k_1, \dots, k_{\nu-1}, c, k_{\nu+1}, \dots, k_n)^T, \quad (5.19)$$

$$\mathbf{k} \leftarrow (c \mathbf{e}_\nu, d \mathbf{e}_\mu) \stackrel{\text{def}}{=} (\mathbf{k} \leftarrow c \mathbf{e}_\nu) \leftarrow d \mathbf{e}_\mu, \quad (5.20)$$

$$1 \leq \mu, \nu \leq n, \quad c, d \in \mathbb{Z}. \quad (5.21)$$

**Definition 2.** Es sei  $\mathbf{k} \in \mathbb{Z}^n$ ,  $\mathbf{k} = \sum_{i=1}^n k_i \mathbf{e}_i = (k_1, \dots, k_n)^T$ . Dann bewirkt das Symbol  $\downarrow$  die Herausnahme einer Komponente.

$$\mathbf{k} \downarrow \mathbf{e}_\nu \stackrel{\text{def}}{=} \sum_{i=1}^{\nu-1} k_i \mathbf{e}_i + \sum_{i=\nu+1}^n k_i \mathbf{e}_{i-1} = (k_1, \dots, k_{\nu-1}, k_{\nu+1}, \dots, k_n)^T, \quad (5.22)$$

$$1 \leq \nu \leq n.$$

Sehr häufig wird auch von der Abkürzung

$$\mathbf{l}_\nu = \mathbf{k} \downarrow \mathbf{e}_\nu \quad (5.23)$$

Gebrauch gemacht werden. Diese Notation wird vor allem dann nützlich sein, wenn Operatoren für den  $n$ -dimensionalen Raum in  $n - 1$  eindimensionale Teiloperatoren aufgeteilt werden können.  $\mathbf{l}_\nu$  bezeichnet dann gewissermaßen eine  $(n - 1)$ -dimensionale Querschnittsposition im Raum.

Es sei angemerkt, dass die oben angegebenen Symbole in der hier verwendeten Form nicht zu den Standard-Symbolen der Mathematik gehören. Sie erlauben jedoch eine anschaulichere Darstellung der folgenden Rechnungen, als es mit Tensoren, Faktorräumen und linearen Mannigfaltigkeiten möglich wäre.

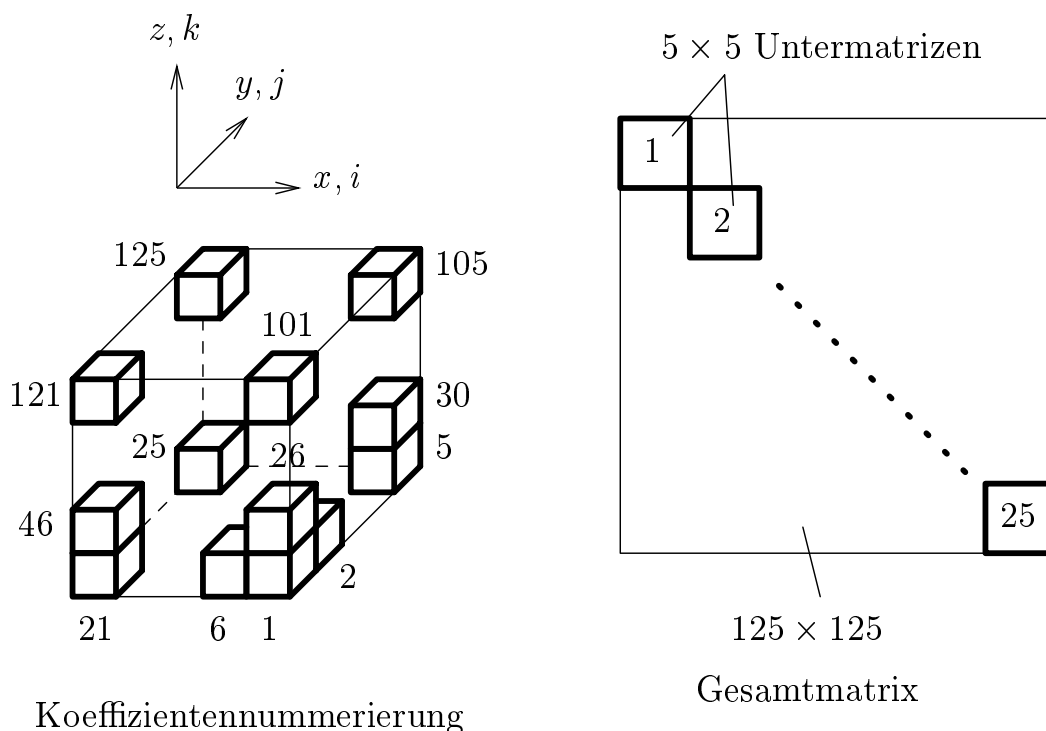


Abbildung 5.1: Nummerierung von Entwicklungskoeffizienten im dreidimensionalen Raum mit 5 Koeffizienten je Richtung. In diesem Beispiel bewirkt die Nummerierung parallel zur  $y$ -Achse, dass ein Operator durch 25 Untermatrizen der Größe  $5 \times 5$  darstellbar ist.

Zur Verdeutlichung der nun folgenden Klassifizierung von Operatoren soll zunächst ein Beispiel diskutiert werden, welches in Abbildung 5.1 veranschaulicht ist. Es geht hierbei um ein dreidimensionales Feldgebiet der Kantenlänge 5, d. h. das Feld wird durch  $5^3 = 125$  Entwicklungskoeffizienten dargestellt. Eine lineare Abbildung von diesem Feld auf eines mit den gleichen Abmessungen lässt sich darstellen durch eine  $125 \times 125$ -Matrix. Die Entwicklungskoeffizienten werden dazu zu einem Vektor zusammengefasst, der dann mit der Matrix multipliziert wird. Dabei ist der genaue Aufbau der Matrix von der Reihenfolge der Entwicklungskoeffizienten in dem Bild- und dem Urbild-Vektor abhängig.

Bei manchen Operatoren lässt sich die Matrix durch eine geschickte Nummerierung der Koeffizienten in eine spezielle Jordan-Normalform bringen. Genauer gesagt wird hier eine zeilenweise Nummerierung der Feldkoeffizienten eingeführt, wie sie in Abbildung 5.1 links zu sehen ist. Jede Zeile liegt in diesem Beispiel parallel zur  $y$ -Achse und enthält fünf Feld-

koeffizienten. Es sollen hier solche Operatoren interessieren, die dann aus  $5 \times 5$ -Untermatrizen aufgebaut werden können, welche auf der Hauptdiagonalen der Gesamtmatrix wie in Abbildung 5.1 angeordnet sind.

Durch die Orientierung der Zeilen entlang der  $y$ -Achse wird die  $x$ - $z$ -Ebene zur Querschnittsfläche. So gesehen lässt sich die  $125 \times 125$ -Gesamtmatrix durch eine  $5 \times 5$ -Untermatrix darstellen, deren Koeffizienten von der Querschnittsposition abhängig sind. Jede  $y$ -Zeile des Koeffizientenblocks ist dann entsprechend ihrer Querschnittsposition mit der zugehörigen Untermatrix zu multiplizieren.

Eine Nummerierung der Feldkoeffizienten entlang der  $x$ - oder  $z$ -Achse würde in dem gezeigten Beispiel bei den meisten Operatoren dazu führen, dass die Gesamtmatrix nicht mehr durch  $5 \times 5$ -Untermatrizen auf der Hauptdiagonalen aufgebaut werden kann. *Skalierungsoperatoren* bilden eine Ausnahme:

**Definition 3.** Die Menge aller Skalierungsoperatoren wird mit  $U^0$  bezeichnet. Es gilt:

$$U^0 = \{ \mathcal{T} \mid t_{\mathbf{k}, \mathbf{k}'} = 0 \text{ für } \mathbf{k} \neq \mathbf{k}' \} . \quad (5.24)$$

In Abbildung 5.1 würde sich ein Skalierungsoperator dadurch zeigen, dass alle Untermatrizen und damit auch die Hauptmatrix reine Diagonalmatrizen wären. Skalierungsoperatoren dienen dazu, jeden Entwicklungskoeffizienten mit einem (i. A. ortsabhängigen) Faktor zu gewichten. Gleichung (5.8) kann in diesem Fall daher auch zu  $b_{\mathbf{k}} = t(\mathbf{k})a_{\mathbf{k}}$  umgeschrieben werden. Der Vorfaktor in Gleichung (2.17) ist ein Beispiel für ein Element eines Skalierungsoperators in der FDTD, wobei dann

$$t(\mathbf{k} = (i, j, k)^T) = -\frac{\Delta t}{\mu_x(i, j, k)} \quad (5.25)$$

ist.

Bei Skalierungsoperatoren ist also jeder Bildkoeffizient stets nur von einem Urbildkoeffizienten mit der gleichen Position abhängig. Es sollen jetzt wieder durch Untermatrizen wie in Abbildung 5.1 darstellbare Operatoren betrachtet werden. Statt der Abbildung zwischen zwei positionsgleichen Punkten geht es dann um eine Abbildung zwischen zwei Zeilen mit gleicher Querschnittsposition. Die Zeilen liegen dabei parallel zu einer Achse  $\nu \in \{1, \dots, n\}$ . Dieser Zusammenhang soll mit der folgenden Definition formalisiert werden:

**Definition 4.** Die Menge aller  $\nu$ -parallelen Operatoren wird mit  $U^\nu$  bezeichnet. Es gilt:

$$U^\nu = \{\mathcal{T} \mid t_{\mathbf{k}, \mathbf{k}'} = 0 \text{ für } \mathbf{l}_\nu \neq \mathbf{l}'_\nu\} , \quad (5.26)$$

wobei  $\mathbf{l}_\nu = \mathbf{k} \downarrow \mathbf{e}_\nu$  und  $\mathbf{l}'_\nu = \mathbf{k}' \downarrow \mathbf{e}_\nu$  Querschnittspositionen des Bild- und des Urbildfeldes sind.

Operatoren vom Typ  $U^\nu$  sind insbesondere für Abschnitt 5.2 von fundamentaler Bedeutung. Da diese durch ihre Untermatrizen vollständig charakterisiert werden, wird im Folgenden häufig der Begriff *Operatormatrix* verwendet werden. Damit ist dann meistens eine von der Querschnittsposition abhängige Matrix gemeint, welche die gleiche Größe wie die Untermatrizen hat und diese in sich vereinigt.

Die Menge  $U^\nu$  lässt sich in zwei weitere Mengen aufteilen:

**Definition 5.** Die Menge aller  $\mathbf{l}_\nu$ -unabhängigen  $\nu$ -parallelen Operatoren wird mit  $U^\nu_{\mathbf{0}}$  bezeichnet. Es gilt:

$$U^\nu_{\mathbf{0}} = \{\mathcal{T} \in U^\nu \mid t_{\mathbf{k}, \mathbf{k} \leftarrow \mathbf{k}' \mathbf{e}_\nu} \neq f(\mathbf{l}_\nu)\} . \quad (5.27)$$

**Definition 6.** Die Menge aller  $\mathbf{l}_\nu$ -abhängigen  $\nu$ -parallelen Operatoren wird mit  $U^\nu_{\mathbf{l}_\nu}$  bezeichnet. Es gilt:

$$U^\nu_{\mathbf{l}_\nu} = U^\nu \setminus U^\nu_{\mathbf{0}} . \quad (5.28)$$

$U^\nu_{\mathbf{0}}$ -Operatoren verfügen über eine querschnittsunabhängige Operatormatrix, d. h. die Untermatrizen wie in Abbildung 5.1 sind alle identisch. Bei  $U^\nu_{\mathbf{l}_\nu}$ -Operatoren dagegen sind wenigstens zwei der Untermatrizen verschieden voneinander.

Für die beiden Operatortypen lassen sich ebenfalls Beispiele in Gleichung (2.17) finden. Dazu wird die finite Differenziation in  $y$ -Richtung betrachtet. Als Operator formuliert wäre dieser vom Typ  $U^y$ . Bei einer äquidistanten Diskretisierung  $\Delta y$  gilt dann sogar die Zugehörigkeit zu  $U^y_{\mathbf{0}}$ . Ist jedoch  $\Delta y$  vom Ort abhängig (genauer: von  $i$  und/oder  $k$ ), stammt der Operator aus  $U^y_{\mathbf{l}_y}$ , wobei  $\mathbf{l}_y = (i, k)^T$  ist.<sup>1</sup>

<sup>1</sup>Im klassischen FDTD-Verfahren kann  $\Delta y$  nicht von  $i$  oder  $k$  abhängen, lediglich von  $j$ . Das zugehörige Diskretisierungsgitter wird auch als *graded mesh*, d. h. abgestuftes Gitter bezeichnet. Wird die FDTD jedoch um einen Untergitter-Algorithmus erweitert, ist eine  $i$ - und  $k$ -Abhängigkeit von  $\Delta y$  möglich.

Aus der Definition der Skalierungsoperatoren lässt sich außerdem schließen, dass

$$U^0 \subset U^\nu \quad \forall \nu \quad (5.29)$$

gilt. Im Einzelfall ist daher  $U^0$  aus der Betrachtung von Transformationseigenschaften auszunehmen.

In diesem Abschnitt wurden vornehmlich Operatortypen beschrieben und in den Definitionen formalisiert. Der Zusammenhang mit (Unter-)Matrizen und Vektoren stand dabei zwar immer im Vordergrund, wurde jedoch nur anschaulich wiedergegeben. Im folgenden Abschnitt 5.2 werden u. a. am konkreten Beispiel der Wavelet-Transformation die Formalismen für die Operatormatrizen und Vektoren eingeführt.

## 5.2 Matrix-Darstellung von Operatoren

Für den eindimensionalen Raum wurde in Abschnitt 5.1.2 bereits gezeigt, wie sich die Operatoren durch Matrizen darstellen und durch Matrixmultiplikationen transformieren lassen. Diese Art der Transformation erscheint besonders einfach. Es soll hier deshalb untersucht werden, unter welchen Bedingungen diese Methodik beibehalten werden kann. Hierzu ist es nötig Operatoren zu klassifizieren, wie es schon in Abschnitt 5.1.4 für einige typische FDTD-Operatoren geschehen ist. Ebenso müssen die Operatoren der Wavelet-Transformation klassifiziert werden.

Die grundlegende Idee zur Transformation der FDTD in den Waveletbereich mit Hilfe von Matrizen wurde erstmalig in [3] präsentiert. Einige Details dieses Ansatzes bleiben dort jedoch im Unklaren und bedürfen einer ausführlicheren Untersuchung. Die folgenden Abschnitte präsentieren daher eine rigorose Analyse der Matrix-basierten WT von Operatoren. Die Betrachtung allgemeiner linearer Operatoren in beliebig vielen Dimensionen stellt ferner eine weitreichende Verallgemeinerung gegenüber der FDTD dar. Dies erlaubt es, die Anwendbarkeit dieser Form der WT für weitere Simulationsverfahren besser abzuschätzen. Darüber hinaus gibt diese Untersuchung einen Einblick in die generellen Möglichkeiten und Grenzen bei der Matrix-basierten Transformation von Operatoren mit Wavelets und anderen Funktionen (z. B. die trigonometrischen Funktionen der FFT). Speziell für die FDTD fanden einige Erkenntnisse aus den

folgenden Abschnitten Eingang in [18], wo der Algorithmus zur WT der FDTD etwas detaillierter beschrieben wird.

Die Analyse gliedert sich wie folgt: Zunächst werden die Operatoren der WT für die verschiedenen Basen nach Kapitel 4 klassifiziert. Im Anschluss daran werden  $U^\nu$ -Operatoren in allgemeiner Form auf ihre Eignung zu einer Matrix-basierten Transformation in den Waveletbereich untersucht (Abschnitt 5.2.2). Als Resultat dieser Untersuchung werden Algorithmen vorgestellt, mit denen sich eine Transformation für die verschiedenen Operortypen durchführen lässt (Abschnitt 5.2.3). Im Vorgriff auf diesen Abschnitt soll bereits hier erwähnt werden, dass der Umgang mit Matrix-basierten Transformationen auch für scheinbar einfache Problemstellungen relativ kompliziert werden kann. Für den Fall der FDTD wird dies u. a. am Beispiel der Materialparameter in Abschnitt 7.2.3 deutlich werden. Ein unbestreitbarer Vorteil dieser Methode ist jedoch ihre einfache Implementierbarkeit in einem Simulationsprogramm, da sich alle Operationen auf gewöhnliche Multiplikationen von (Operator-)Matrizen zurückführen lassen. Letztlich ist für den jeweiligen Einsatzzweck zu entscheiden, welcher Nutzen aus dem Verfahren der Matrix-basierten Transformation gezogen werden kann.

### 5.2.1 Klassifizierung von WT-Operatoren

In Abschnitt 4.1 wurde gesagt, dass bei der Verwendung homogener Tensorprodukt-Basen eine  $n$ -fache eindimensionale DPWT benutzt werden kann:  $\mathcal{P}_{\mathbf{0},\mathbf{j}} = \mathcal{P}_{0,j_1}^1 \circ \dots \circ \mathcal{P}_{0,j_n}^n$ . Für jede eindimensionale DPWT  $\mathcal{P}_{0,j_\nu}^\nu$  existiert die zugehörige Matrix  $\mathbf{P}_{0,j_\nu}^\nu$ , da  $\mathcal{P}_{0,j_\nu}^\nu$  vom Typ  $U_0^\nu$  ist.

Das zu transformierende Signal wird durch den  $n$ -dimensionalen Quader der Koeffizienten  $a_{\mathbf{k}}$  repräsentiert. Nimmt man eine zur  $\mathbf{e}_\nu$ -Richtung parallele Reihe aus dem Quader, lässt sich diese als Vektor  $\mathbf{a}$  darstellen. Der transformierte Vektor lautet dann  $\mathbf{b} = \mathbf{P}_{0,j_\nu}^\nu \mathbf{a}$ . Diese Prozedur ist über den gesamten Querschnitt durchzuführen.

Formalisiert lässt sich das so darstellen: Für jede Position  $\mathbf{k} \in \mathbf{K} \subset \mathbb{Z}^n$  gelte  $1 \leq k_\nu \leq K_\nu$ ,  $K_\nu \in \mathbb{N}$ ,  $1 \leq \nu \leq n$ . Ferner lässt sich eine Isomorphie zwischen allen Koeffizienten und Vektoren bilden: Mit

$$\mathbf{a}_{\mathbf{l}_\nu} = (a_{\mathbf{k} \leftarrow -1\mathbf{e}_\nu}, \dots, a_{\mathbf{k} \leftarrow -K_\nu\mathbf{e}_\nu})^T, \quad \mathbf{l}_\nu = \mathbf{k} \downarrow \mathbf{e}_\nu, \quad (5.30)$$

gilt die Äquivalenz

$$\{a_{\mathbf{k}} \mid \mathbf{k} \in \mathbf{K}\} \Leftrightarrow \{\mathbf{a}_{\mathbf{l}_\nu} \mid \mathbf{l}_\nu \in \mathbf{L}_\nu^{\mathbf{k}}\}, \quad (5.31)$$

wobei

$$\mathbf{L}_\nu^{\mathbf{k}} = \{\mathbf{l}_\nu \mid \mathbf{l}_\nu = \mathbf{k} \downarrow \mathbf{e}_\nu, \mathbf{k} \in \mathbf{K}\} \subset \mathbb{Z}^{n-1} \quad (5.32)$$

ist. Repräsentieren die  $a_{\mathbf{k}}$  bzw.  $\mathbf{a}_{\mathbf{l}_\nu}$  die Funktion  $f$  und die  $b_{\mathbf{k}}$  bzw.  $\mathbf{b}_{\mathbf{l}_\nu}$  die in einer Richtung Wavelet-Transformierte  $g = \mathcal{P}_{0,j_\nu}^\nu f$ , so gilt

$$\mathbf{b}_{\mathbf{l}_\nu} = \mathbf{P}_{0,j_\nu}^\nu \mathbf{a}_{\mathbf{l}_\nu}, \quad (5.33)$$

was in diesem Fall äquivalent zu

$$b_{\mathbf{k}} = \sum_{\mathbf{k}'} p_{\mathbf{k},\mathbf{k}'}^{0,j_\nu}(\nu) a_{\mathbf{k}'} \quad (5.34)$$

ist, wobei  $p_{\mathbf{k},\mathbf{k}'}^{0,j_\nu}(\nu)$  die zu  $\mathcal{P}_{0,j_\nu}^\nu$  gehörende  $2n$ -dimensionale Relation ist. Für die vollständige Transformation  $\mathcal{P}_{0,\mathbf{j}}$  müssen also  $n$  i. A. voneinander verschiedene Matrizen  $\mathbf{P}_{0,j_\nu}^\nu$  angegeben werden.

Bei separablen MSA-Basen nach Abschnitt 4.2 sind in der Regel mehr Matrizen vonnöten. Der Grund hierfür liegt in der Vorgehensweise: Für jede Skala sind erst in allen Richtungen einstufige Wavelet-Transformationen durchzuführen, bevor mit der nächsten Skala begonnen werden kann. Insgesamt sind dadurch  $n \cdot j$  DPWT-Matrizen erforderlich.

Es wird die einstufige eindimensionale Transformation  $\mathcal{P}_{\mu,\mu+1}^\nu$  betrachtet. Urbild und Bild sind von gleicher Größe wie der  $n$ -dimensionale Raum  $\mathbf{V}_\mu$ . Auf dieses Transformationsgebiet bezogen ist  $\mathcal{P}_{\mu,\mu+1}^\nu$  vom Typ  $U_{\mathbf{0}}^\nu$ , wenn man das Ordnungsschema nach Abbildung 4.2 benutzt.<sup>2</sup> Bezieht man jedoch auch den aus  $\mathbf{W}_1, \dots, \mathbf{W}_\mu$  bestehenden Restraum mit ein, ist  $\mathcal{P}_{\mu,\mu+1}^\nu$  vom Typ  $U_{\mathbf{l}_\nu}^\nu$ :

Bezeichnet  $\mathbf{l}_\nu$  eine Reihe mit Koeffizienten, die im Gebiet von  $\mathbf{V}_\nu$  liegen, so besteht die Transformationsmatrix  $\mathbf{P}_{\mu,\mu+1}^\nu$  für diese Reihe entsprechend Gleichung (3.63) aus der direkten Summe der eigentlichen DPWT-Matrix für den  $\mathbf{V}_\nu$ -Bereich sowie einer Einheitsmatrix für den Restbereich. Wird jedoch auch eine Reihe betrachtet, die vollständig außerhalb

<sup>2</sup>Die in der Abbildung 4.2 gezeigten Bezeichnungen  $\mathbf{V}$  und  $\mathbf{W}$  sind eindimensionale Funktionenräume, während in diesem Abschnitt  $\mathbf{V}$  und  $\mathbf{W}$  je nach Kontext auch  $n$ -dimensionale Funktionenräume sein können.

des  $V_\nu$ -Gebietes liegt, dann ist die für dieses  $I_\nu$  zuständige Matrix  $\mathbf{P}_{\mu,\mu+1}^\nu$  identisch mit einer Einheitsmatrix.

Zu nicht-separablen MSA-Basen lässt sich nach Abschnitt 4.3 zwar eine schnelle Wavelet-Transformation angeben, doch eine matrizenartige Formulierung dieses Algorithmus dürfte im Allgemeinen schwierig sein. In Abschnitt 5.2.3.4 wird auf die Probleme separabler und nicht-separabler MSA-Basen nochmals näher eingegangen.

## 5.2.2 Operortypen in der WT

Im Folgenden wird untersucht, unter welchen Umständen Operatoren vom Typ  $U^\nu$  einer ein- oder mehrdimensionalen WT durch Matrixmultiplikationen unterzogen werden können. Der transformierte Operator berechnet sich jeweils zu  $\tilde{\mathcal{T}} = \mathcal{P} \mathcal{T} \mathcal{P}^*$ .

Es wird sich in diesem Abschnitt zeigen, dass die Voraussetzungen für eine Matrix-basierte WT sehr restriktiv sind. Im hierauf folgenden Abschnitt 5.2.3 werden dann Algorithmen vorgestellt, mit deren Hilfe die Beschränkungen umgangen werden können.

**Satz 1.** *Es sei  $\mathcal{T} \in U^\nu$  sowie  $\mathcal{P} = \mathcal{P}_{0,j_\nu}^\nu \in U^\nu$ . Dann gilt  $\tilde{\mathcal{T}} \in U_{\mathbf{0}}^\nu$  nur dann, wenn  $\mathcal{T}$  und  $\mathcal{P}$  vom Typ  $U_{\mathbf{0}}^\nu$  sind. Anderenfalls ist  $\tilde{\mathcal{T}} \in U_{\mathbf{1}_\nu}^\nu$ . In jedem Fall lässt sich  $\tilde{\mathcal{T}}$  durch Multiplikation der  $\mathcal{T}$ -,  $\mathcal{P}$ - und  $\mathcal{P}^*$ -Matrizen berechnen.*

*Beweis.* Wegen der Zugehörigkeit der Operatoren zu  $U^\nu$  gilt für die Tensoren

$$p_{\mathbf{k},\mathbf{k}'}^{0,j_\nu} = p_{\mathbf{k},\mathbf{k}'}^{*j_\nu,0} = t_{\mathbf{k},\mathbf{k}'} = 0 \quad \text{für} \quad \mathbf{1}_\nu \neq \mathbf{1}'_\nu. \quad (5.35)$$

Ausgehend von  $\mathbf{k}^1$  kann damit in Gleichung (5.16)  $\mathbf{k}^i = \mathbf{k}^1 \leftarrow k_\nu^i \mathbf{e}_\nu$  gesetzt werden:

$$b_{\mathbf{k}^1}^2 = \sum_{k_\nu^2} p_{k_\nu^1, k_\nu^2}^{0,j_\nu}(\mathbf{1}_\nu^1) \sum_{k_\nu^3} t_{k_\nu^2, k_\nu^3}(\mathbf{1}_\nu^1) \sum_{k_\nu^4} p_{k_\nu^3, k_\nu^4}^{*j_\nu,0}(\mathbf{1}_\nu^1) b_{\mathbf{k}^1 \leftarrow k_\nu^4 \mathbf{e}_\nu}^1, \quad (5.36)$$

wobei  $t_{k_\nu^2, k_\nu^3}(\mathbf{1}_\nu^1) = t_{\mathbf{k}^1 \leftarrow k_\nu^2 \mathbf{e}_\nu, \mathbf{k}^1 \leftarrow k_\nu^3 \mathbf{e}_\nu}$  als Element einer  $I_\nu^1$ -abhängigen Matrix gesehen werden kann. Analoges gilt für die WT-Tensoren. Dadurch ist

$$b_{\mathbf{k}^1}^2 = \sum_{k_\nu^4} \tilde{t}_{k_\nu^1, k_\nu^4}(\mathbf{1}_\nu^1) b_{\mathbf{k}^1 \leftarrow k_\nu^4 \mathbf{e}_\nu}^1 \quad (5.37)$$

bzw. mit der äquivalenten Darstellung nach Gleichung (5.31)

$$\mathbf{b}_{\mathbf{l}_\nu^1}^2 = \tilde{\mathbf{T}}(\mathbf{l}_\nu^1) \mathbf{b}_{\mathbf{l}_\nu^1}^1 \quad (5.38)$$

mit

$$\tilde{t}_{k_\nu^1, k_\nu^4}(\mathbf{l}_\nu^1) = \sum_{k_\nu^2} p_{k_\nu^1, k_\nu^2}^{0, j_\nu}(\mathbf{l}_\nu^1) \sum_{k_\nu^3} t_{k_\nu^2, k_\nu^3}(\mathbf{l}_\nu^1) p_{k_\nu^3, k_\nu^4}^{*, j_\nu, 0}(\mathbf{l}_\nu^1) \quad (5.39)$$

bzw.

$$\tilde{\mathbf{T}}(\mathbf{l}_\nu^1) = \mathbf{P}_{0, j_\nu}^\nu(\mathbf{l}_\nu^1) \mathbf{T}(\mathbf{l}_\nu^1) \mathbf{P}_{j_\nu, 0}^{*, \nu}(\mathbf{l}_\nu^1) . \quad (5.40)$$

Aus Gleichung (5.40) folgt, dass  $\tilde{\mathcal{T}} \in U^\nu$  ist und nur dann  $\tilde{\mathcal{T}} \in U_0^\nu$  ist, wenn  $\mathcal{T}$  und  $\mathcal{P}$  vom Typ  $U_0^\nu$  sind.  $\square$

**Satz 2.** *Es sei  $\mathcal{T} \in U_{\mathbf{l}_\mu^1}^\mu \setminus U^0$  sowie  $\mathcal{P} = \mathcal{P}_{0, j_\nu}^\nu \in U_0^\nu$  und  $\mu \neq \nu$ . Dann ist im Allgemeinen  $\tilde{\mathcal{T}} \notin U^\mu$  und  $\tilde{\mathcal{T}} \notin U^\nu$ .*

*Beweis.* Es wird wieder mit Gleichung (5.16) begonnen. Es gilt jetzt  $t_{\mathbf{k}^2, \mathbf{k}^3} = 0$  für  $\mathbf{l}_\mu^2 \neq \mathbf{l}_\mu^3$ . Es wird daher  $\mathbf{k}^2 = \mathbf{k}^3 \leftarrow k_\mu^2 \mathbf{e}_\mu$  gesetzt und die Summe über  $\mathbf{k}^3$  nach vorne gezogen:

$$b_{\mathbf{k}^1}^2 = \sum_{k_\mu^3} \sum_{\mathbf{l}_\mu^3} \sum_{k_\mu^2} p_{\mathbf{k}^1, \mathbf{k}^3 \leftarrow k_\mu^2 \mathbf{e}_\mu}^{0, j_\nu} t_{k_\mu^2, k_\mu^3}(\mathbf{l}_\mu^3) \sum_{\mathbf{k}^4} p_{\mathbf{k}^3, \mathbf{k}^4}^{*, j_\nu, 0} b_{\mathbf{k}^4}^1 . \quad (5.41)$$

In  $p_{\mathbf{k}^1, \mathbf{k}^2}^{0, j_\nu}$  kann wegen  $\mathcal{P} = \mathcal{P}_{0, j_\nu}^\nu \in U_0^\nu$  jetzt  $\mathbf{l}_\nu^2 = \mathbf{l}_\nu^1$  gesetzt werden. Da auch  $\mu \neq \nu$  gilt, folgt  $k_\mu^2 = k_\mu^1$ , d. h. die Summe über  $k_\mu^2$  kann entfallen. Außerdem werden die Freiheiten von  $\mathbf{k}^3$  beschränkt. Es muss

$$\mathbf{l}_\nu^2 = (\mathbf{k}^3 \leftarrow k_\mu^1 \mathbf{e}_\mu) \downarrow \mathbf{e}_\nu \stackrel{!}{=} \mathbf{l}_\nu^1 \quad (5.42)$$

gelten, woraus  $k_i^3 = k_i^1$ ,  $i \notin \{\mu, \nu\}$  folgt, d. h.

$$\mathbf{k}^3 = \mathbf{k}^1 \leftarrow (k_\mu^3 \mathbf{e}_\mu, k_\nu^3 \mathbf{e}_\nu) . \quad (5.43)$$

Damit reduziert sich die Summe über  $\mathbf{l}_\mu^3$  auf  $k_\nu^3$ . Ferner ist jetzt

$$\mathbf{l}_\mu^3 = (\mathbf{k}^1 \leftarrow k_\nu^3 \mathbf{e}_\nu) \downarrow \mathbf{e}_\mu . \quad (5.44)$$

Daraus folgt

$$b_{\mathbf{k}^1}^2 = \sum_{k_\mu^3} \sum_{k_\nu^3} p_{k_\nu^1, k_\nu^3}^{0, j_\nu} t_{k_\mu^1, k_\mu^3}(\mathbf{l}_\mu^3) \sum_{\mathbf{k}^4} p_{\mathbf{k}^3, \mathbf{k}^4}^{*j_\nu, 0} b_{\mathbf{k}^4}^1. \quad (5.45)$$

Schließlich kann in  $p_{\mathbf{k}^3, \mathbf{k}^4}^{*j_\nu, 0}$  noch die Einschränkung  $\mathbf{l}_\nu^4 = \mathbf{l}_\nu^3$  angewendet werden:

$$\mathbf{k}^4 \downarrow \mathbf{e}_\nu \stackrel{!}{=} \mathbf{k}^3 \downarrow \mathbf{e}_\nu = (\mathbf{k}^1 \leftarrow k_\mu^3 \mathbf{e}_\mu) \downarrow \mathbf{e}_\nu \quad (5.46)$$

$$\implies \mathbf{k}^4 = \mathbf{k}^1 \leftarrow (k_\mu^3 \mathbf{e}_\mu, k_\nu^4 \mathbf{e}_\nu). \quad (5.47)$$

Damit ergibt sich zusammen mit der  $\mathbf{l}_\nu^3$ -Unabhängigkeit von  $p_{\mathbf{k}^3, \mathbf{k}^4}^{*j_\nu, 0}$

$$b_{\mathbf{k}^1}^2 = \sum_{k_\mu^3} \sum_{k_\nu^3} p_{k_\nu^1, k_\nu^3}^{0, j_\nu} t_{k_\mu^1, k_\mu^3}(\mathbf{l}_\mu^3) \sum_{k_\nu^4} p_{k_\nu^3, k_\nu^4}^{*j_\nu, 0} b_{\mathbf{k}^4}^1, \quad (5.48)$$

$$\text{mit } \mathbf{l}_\mu^3 = (\mathbf{k}^1 \leftarrow k_\nu^3 \mathbf{e}_\nu) \downarrow \mathbf{e}_\mu, \quad \mathbf{k}^4 = \mathbf{k}^1 \leftarrow (k_\mu^3 \mathbf{e}_\mu, k_\nu^4 \mathbf{e}_\nu).$$

Wegen der  $k_\nu^3$ -Abhängigkeit von  $\mathbf{l}_\mu^3$  lässt sich der Ausdruck i. A. nicht weiter vereinfachen. In Satz 1 konnte jedem einzelnen  $b_{\mathbf{k}^1}^2$  eine eindimensionale Reihe von  $b_{\mathbf{k}^4}^1$ -Koeffizienten zugeordnet werden. In Gleichung (5.48) jedoch zeigt die Abhängigkeit von  $b_{\mathbf{k}^4}^1$  von den Summationsindizes  $k_\mu^3$  und  $k_\nu^4$  eine zweidimensionale Zuordnung von Koeffizienten. Eine einfache Matrixdarstellung scheidet damit als Möglichkeit aus.  $\square$

Gleichung (5.48) bleibt allenfalls für zweidimensionale Operatoren (z. B. bei einer zweidimensionalen FDTD) interessant, da nur über drei skalare Laufindizes summiert werden muss, im Vergleich zu sechs Indizes bei Gleichung (5.16).

Obwohl mit  $\mathcal{P} \in U_{\mathbf{0}}^\nu$  eine Vereinfachung im Vergleich zu  $\mathcal{P} \in U^\nu$  in Satz 1 benutzt wurde, ist der Versuch der Herleitung einer einfachen Matrixdarstellung gescheitert. Man kann sich leicht denken, dass die auftretenden Schwierigkeiten bei einer Voraussetzung wie  $\mathcal{P} \in U_{\mathbf{l}_\nu}^\nu$  noch größer würden.

**Satz 3.** *Es sei  $\mathcal{T} \in U_{\mathbf{0}}^\mu$  sowie  $\mathcal{P} = \mathcal{P}_{0, j_\nu}^\nu \in U_{\mathbf{0}}^\nu$  und  $\mu \neq \nu$ . Dann ist  $\tilde{\mathcal{T}} = \mathcal{T}$ .*

*Beweis.* Es kann von Gleichung (5.48) ausgegangen werden. Wegen  $\mathcal{T} \in U_{\mathbf{0}}^\mu$  ist  $t_{k_\mu^1, k_\mu^3}$  von  $\mathbf{l}_\mu^3$  und damit von  $k_\nu^3$  unabhängig. Daher ist folgende

Umordnung zulässig:

$$b_{\mathbf{k}^1}^2 = \sum_{k_\mu^3} t_{k_\mu^1, k_\mu^3} \sum_{k_\nu^4} \underbrace{\sum_{k_\nu^3} p_{k_\nu^1, k_\nu^3}^{0, j_\nu} p_{k_\nu^3, k_\nu^4}^{* j_\nu, 0}}_{m_{k_\nu^1, k_\nu^4}} b_{\mathbf{k}^4}^1 . \quad (5.49)$$

$m_{k_\nu^1, k_\nu^4}$  entspricht einem Matrixelement aus der Verknüpfung  $\mathcal{P}\mathcal{P}^* = \text{id}$ , d. h. es gilt

$$m_{k_\nu^1, k_\nu^4} = \delta_{k_\nu^1, k_\nu^4} , \quad (5.50)$$

wobei  $\delta$  das Kronecker-Symbol ist. Dadurch kann in Gleichung (5.49)  $k_\nu^4 = k_\nu^1$  gesetzt werden:

$$b_{\mathbf{k}^1}^2 = \sum_{k_\mu^3} t_{k_\mu^1, k_\mu^3} b_{\mathbf{k}^1 \leftarrow k_\mu^3 \mathbf{e}_\mu}^1 . \quad (5.51)$$

□

Der folgende Satz betrachtet die in Satz 2 ausgenommenen Skalierungsoperatoren.

**Satz 4.** *Es sei  $\mathcal{T} \in U^0 \cap U_{\mathbf{1}_\mu}^\mu$  sowie  $\mathcal{P} = \mathcal{P}_{0, j_\nu}^\nu \in U_{\mathbf{0}}^\nu$  und  $\mu \neq \nu$ . Dann ist  $\tilde{\mathcal{T}} \in U^\nu$ .*

*Beweis.* Wieder kann Gleichung (5.48) benutzt werden. Die Voraussetzung  $\mathcal{T} \in U^0$  erlaubt die Beschränkung auf  $k_\mu^3 = k_\mu^1$ :

$$b_{\mathbf{k}^1}^2 = \sum_{k_\nu^3} p_{k_\nu^1, k_\nu^3}^{0, j_\nu} t_{k_\mu^1, k_\mu^1}(\mathbf{1}_\mu^3) \sum_{k_\nu^4} p_{k_\nu^3, k_\nu^4}^{* j_\nu, 0} b_{\mathbf{k}^4}^1 , \quad (5.52)$$

$$\text{mit } \mathbf{1}_\mu^3 = (\mathbf{k}^1 \leftarrow k_\nu^3 \mathbf{e}_\nu) \downarrow \mathbf{e}_\mu , \quad \mathbf{k}^4 = \mathbf{k}^1 \leftarrow k_\nu^4 \mathbf{e}_\nu . \quad (5.53)$$

Dann ergibt eine Umsortierung und genaue Analyse

$$b_{\mathbf{k}^1}^2 = \sum_{k_\nu^4} \tilde{t}_{k_\nu^1, k_\nu^4}(\mathbf{1}_\nu^1) b_{\mathbf{k}^1 \leftarrow k_\nu^4 \mathbf{e}_\nu}^1 , \quad (5.54)$$

$$\tilde{t}_{k_\nu^1, k_\nu^4}(\mathbf{1}_\nu^1) = \sum_{k_\nu^3} p_{k_\nu^1, k_\nu^3}^{0, j_\nu} t_{k_\mu^1, k_\mu^1}(\mathbf{1}_\mu^3) p_{k_\nu^3, k_\nu^4}^{* j_\nu, 0} . \quad (5.55)$$

Dass  $\tilde{t}_{k_\nu^1, k_\nu^4}$  von  $\mathbf{l}_\nu^1$  statt einem  $\mathbf{l}_\mu$  abhängt, mag etwas überraschen, ergibt sich aber eindeutig aus der eindimensionalen Relation zwischen  $b_{\mathbf{k}^1}^2$  und  $b_{\mathbf{k}^4}^1$ , die wegen  $\mathbf{k}^4 = \mathbf{k}^1 \leftarrow k_\nu^4 \mathbf{e}_\nu$  einem Operator vom Typ  $U^\nu$  entspricht, nicht jedoch einem vom Typ  $U^\mu$ . Des Weiteren kann nun die rechte Seite von Gleichung (5.55) herangezogen werden.

Unter Berücksichtigung der Abhängigkeit von  $\mathbf{l}_\mu^3$  treten nach der Summation über  $k_\nu^3$  außer  $k_\nu^4$  auch sämtliche  $\mathbf{k}^1$ -Koeffizienten auf:  $k_\nu^1$  im Index von  $p$ ,  $k_\mu^1$  im Index von  $t$  und alle anderen  $\mathbf{k}^1$ -Koeffizienten in  $\mathbf{l}_\mu^3$ . Werden  $k_\nu^1$  und  $k_\nu^4$  entsprechend der eindimensionalen Relation zwischen  $b_{\mathbf{k}^1}^2$  und  $b_{\mathbf{k}^4}^1$  als Matrixindizes für  $\tilde{t}$  verwendet, bleiben noch die  $\mathbf{l}_\nu^1$ -Koeffizienten als Abhängigkeit für  $\tilde{t}$  übrig.  $\square$

Nach Satz 4 liegen für die Transformation in eine zweite Richtung Verhältnisse vor, wie sie in Satz 2 diskutiert wurden. Lediglich weitere Einschränkungen würden demnach Matrix-basierte Transformationen erlauben.

### 5.2.3 Algorithmen zur Matrix-basierten WT von Operatoren

Die vorangegangenen Untersuchungen haben gezeigt, dass lediglich dann Operatoren in mehreren Richtungen mittels Matrixmultiplikationen in den Waveletbereich transformiert werden können, wenn Voraussetzungen vorliegen wie sie in den Sätzen 1 und 3 angegeben wurden. Durch schrittweise Verallgemeinerung wird gezeigt, wie sich Operatoren derart modifizieren lassen, dass sie diesen Voraussetzungen genügen. Es soll wiederum jeweils  $\tilde{\mathcal{T}} = \mathcal{P} \mathcal{T} \mathcal{P}^*$  berechnet werden.

#### 5.2.3.1 $U_0^\mu$ -Operatoren im Waveletbereich

Es gelten folgende Voraussetzungen:

- $\mathcal{T} \in U_0^\mu$  für ein konstantes  $\mu$  mit  $1 \leq \mu \leq n$
- $\mathcal{P} = \mathcal{P}_{0,j_1}^1 \cdots \mathcal{P}_{0,j_n}^n$  mit  $\mathcal{P}_{0,j_\nu}^\nu \in U_0^\nu \forall \nu$

Dann gilt entsprechend der Sätze 1 und 3

$$\tilde{\mathcal{T}} = \mathcal{P} \mathcal{T} \mathcal{P}^* = \mathcal{P}_{0,j_\mu}^\mu \mathcal{T} \mathcal{P}_{j_\mu,0}^{*\mu} \in U_{\mathbf{0}}^\mu, \quad (5.56)$$

d. h.  $\mathcal{T}$  muss lediglich in Richtung  $\mu$  transformiert werden. In Matrixform geschrieben heißt das

$$\tilde{\mathbf{T}} = \mathbf{P}_{0,j_\mu}^\mu \mathbf{T} \mathbf{P}_{j_\mu,0}^{*\mu}. \quad (5.57)$$

Im Unterschied zu Gleichung (5.40) entfällt also eine Querschnittsabhängigkeit  $l_\mu$ . Abbildung 5.2 veranschaulicht, wie eine kontinuierliche Abbildung  $\mathcal{T} : f \mapsto g$ ,  $\mathcal{T} \in U_{\mathbf{0}}^y$  in diskreter Form durchgeführt wird (vgl. auch Abschnitt 5.2.1). Wegen der  $U_{\mathbf{0}}^\mu$ -Zugehörigkeit gilt dieses Prinzip auch für den in Gleichung (5.56) berechneten Wavelet-transformierten Operator  $\tilde{\mathcal{T}}$ .

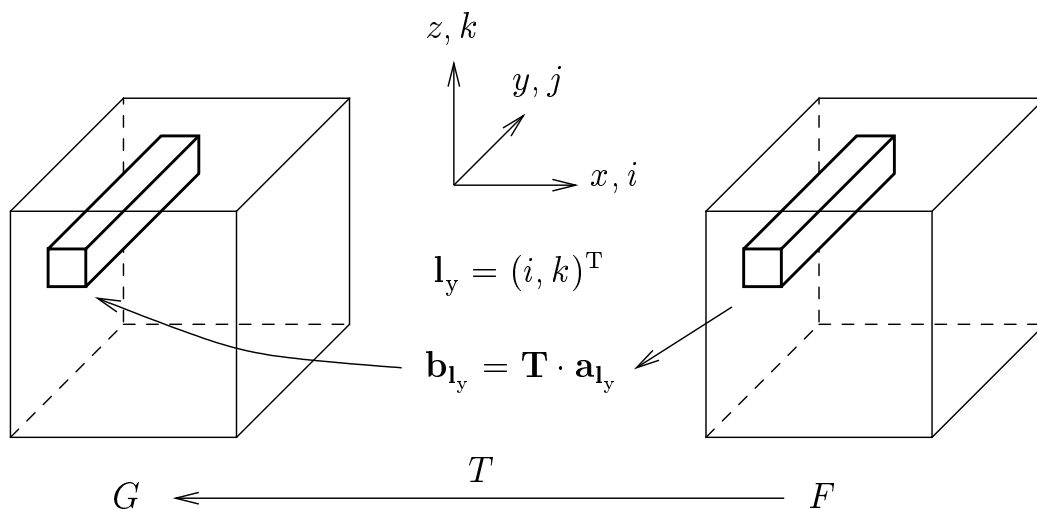


Abbildung 5.2: Abbildung von Entwicklungskoeffizienten ( $T : F \mapsto G$ ) bei einem  $U_{\mathbf{0}}^y$ -Operator in drei Dimensionen.  $\mathbf{T}$  ist die (konstante) Operatormatrix zur diskreten Abbildung  $T$ . Koeffizientenreihen bilden Vektoren  $(\mathbf{a}, \mathbf{b})$ .

### 5.2.3.2 $U_{\mathbf{l}_\mu}^\mu$ -Operatoren

Als erste Verallgemeinerung wird eine Querschnittsabhängigkeit der Operatormatrizen zugelassen. Es gelten die Voraussetzungen

- $\mathcal{T} \in U_{\mathbf{l}_\mu}^\mu$  für ein konstantes  $\mu$  mit  $1 \leq \mu \leq n$ ,
- $\mathcal{P} = \mathcal{P}_{0,j_1}^1 \cdots \mathcal{P}_{0,j_n}^n$  mit  $\mathcal{P}_{0,j_\nu}^\nu \in U_0^\nu \forall \nu$ .

Die Vorgehensweise ist folgende:

1. Bildung von  $U_0^\mu$ -Operatoren, die für wenigstens ein  $\mathbf{l}_\mu$  mit  $\mathcal{T}$  übereinstimmen.
2. Bildung von Maskierungsoperatoren. (Diese sind i. A. vom Typ  $U_{\mathbf{l}_\nu}^\nu$ .)
3. Zerlegung der Maskierungsoperatoren in  $U_0^\nu$ -Operatoren ( $\nu \neq \mu$ ).

Am Ende liegen dann noch nur noch  $U_0^\eta$ -Operatoren vor, die gemäß Abschnitt 5.2.3.1 transformiert werden können. Die konkreten Arbeitsschritte sind:

1. Aufteilung der Menge  $\mathbf{L}_\mu^{\mathbf{k}}$  der zu  $\mathcal{T}$  bzw.  $T$  gehörenden  $\mathbf{l}_\mu$ -Vektoren, sodass Folgendes gilt:

$$\bigcup_a \mathbf{L}_{\mu,a}^{\mathbf{k}} = \mathbf{L}_\mu^{\mathbf{k}}, \quad (5.58)$$

$$\mathbf{L}_{\mu,a}^{\mathbf{k}} \cap \mathbf{L}_{\mu,b}^{\mathbf{k}} = \emptyset \quad \forall \quad a, b \quad \text{mit} \quad a \neq b, \quad (5.59)$$

$$\mathbf{T}(\mathbf{l}_{\mu,a}^b) = \mathbf{T}(\mathbf{l}_{\mu,a}^c) \quad \forall \quad a, b, c; \quad \mathbf{l}_{\mu,a}^\eta \in \mathbf{L}_{\mu,a}^{\mathbf{k}}. \quad (5.60)$$

Damit werden Bereiche mit gleichen Eigenschaften in  $\mathcal{T}$  identifiziert. Die Anzahl der Bereiche sollte so gering wie möglich ausfallen. Im ungünstigsten Fall jedoch enthält jedes  $\mathbf{L}_{\mu,a}^{\mathbf{k}}$  genau ein Element.

Abbildung 5.3 zeigt ein Beispiel für einen dreidimensionalen Operator vom Typ  $U_{\mathbf{l}_y}^y$ . Zu sehen ist  $\mathcal{T}$  in der xz-Ebene (entsprechend  $\mathbf{l}_y = (i, k)^T$ ). Für jede Querschnittsposition  $(i, k)$  ist die zugehörige Matrix bezeichnet.  $\mathcal{M}_1 \mathcal{T}_1$  stellt das erste Ergebnis bei der Suche nach gleichen Transformationseigenschaften dar (Matrix  $\alpha$ ).

$\alpha$	$\beta$	$\alpha$	$\alpha$	=	$\alpha$		$\alpha$	$\alpha$	+		$\beta$			
$\gamma$	$\beta$	$\delta$	$\delta$							$\gamma$	$\beta$	$\delta$	$\delta$	
$\alpha$	$\beta$	$\alpha$	$\delta$		$\alpha$		$\alpha$				$\beta$		$\delta$	
$\gamma$	$\beta$	$\delta$	$\delta$							$\gamma$	$\beta$	$\delta$	$\delta$	

$$\mathcal{T} = \mathcal{M}_1 \mathcal{T}_1 + \mathcal{T} - \mathcal{M}_1 \mathcal{T}_1$$
  

$\alpha$		$\alpha$	$\alpha$	=	<b>I</b>		<b>I</b>	<b>I</b>	\circ	$\alpha$	$\alpha$	$\alpha$	$\alpha$
										$\alpha$	$\alpha$	$\alpha$	$\alpha$
$\alpha$		$\alpha$			<b>I</b>	<b>I</b>				$\alpha$	$\alpha$	$\alpha$	$\alpha$
										$\alpha$	$\alpha$	$\alpha$	$\alpha$

$$\mathcal{M}_1 \mathcal{T}_1 = \mathcal{M}_1 \circ \mathcal{T}_1$$

Abbildung 5.3: Zerlegung eines  $U_{1_y}^y$ -Operators ( $\mathcal{T}$ ) in einen  $U_0^y$ -Anteil ( $\mathcal{T}_1$ ), einen Maskierungsoperator ( $\mathcal{M}_1$ ) sowie einen analog zerlegbaren Restoperator.  $\alpha, \beta, \gamma, \delta$  sind die zu  $\mathcal{T}$  geh\u00f6renden Matrizen. **I** ist die Einheitsmatrix. Leere K\u00e4stchen stehen f\u00fcr leere Matrizen (alle Elemente identisch Null).

2. Bildung von  $U_0^\mu$ -Operatoren aus  $\mathcal{T}$ . F\u00fcr jedes  $\mathcal{T}_a \in U_0^\mu$  soll

$$\mathbf{T}_a = \mathbf{T}(\mathbf{l}_\mu), \quad \mathbf{l}_\mu \in \mathbf{L}_{\mu,a}^{\mathbf{k}} \tag{5.61}$$

gelten, d. h. die Operatormatrix  $\mathbf{T}_a$  zu  $\mathcal{T}_a$  ist wie gefordert konstant. In Abbildung 5.3 gilt dies f\u00fcr  $\mathcal{T}_1$ .

3. Bildung von Maskierungsoperatoren  $\mathcal{M}_a \in U^0 (\cap U_{1_\mu}^\mu)$  \u00fcber die Tensorelemente  $m_{\mathbf{k},\mathbf{k}'}^a$ :

$$m_{\mathbf{k},\mathbf{k}'}^a \stackrel{!}{=} \begin{cases} 1 & , \mathbf{k} = \mathbf{k}' \wedge \mathbf{k} \downarrow \mathbf{e}_\mu \in \mathbf{L}_{\mu,a}^{\mathbf{k}} \\ 0 & , \text{sonst} \end{cases} . \tag{5.62}$$

Dadurch gilt

$$\text{id} = \sum_a \mathcal{M}_a, \quad (5.63)$$

$$\mathcal{T} = \sum_a \mathcal{M}_a \mathcal{T}_a. \quad (5.64)$$

Das Beispiel in Abbildung 5.3 zeigt  $\mathcal{M}_1$  als Maskierungsoperator für  $\mathcal{T}_1$ .

4. Zerlegung der  $\mathcal{M}_a$ . Im Allgemeinen gilt  $\mathcal{M}_a \notin U_0^\nu \forall \nu$ . Es muss jetzt für jedes  $\mathcal{M}_a$  eine Zerlegung in weitere Maskierungsoperatoren gefunden werden, sodass diese vom Typ  $U_0^\nu \cap U^0$  sind und über Verkettung und Summierung wieder  $\mathcal{M}_a$  ergeben.

Formal betrachtet ist die Zerlegung in zwei Stufen durchzuführen, jedoch hängt jede der Stufen von der jeweils anderen ab. Die wesentliche Idee ist, dass in der zweiten Stufe die Maskierung von (gezielt gruppierten bzw. „gesammelten“) Bildpunkten dadurch realisiert wird, dass diese Bildpunkte als Schnittpunkte orthogonaler Flächen dargestellt werden. Hierbei ist mit einem Punkt ein 0-dimensionales Element gemeint (ein einzelnes  $\mathbf{k} \in \mathbb{Z}^n$ ), während eine Fläche  $n - 1$  Dimensionen haben soll und damit auch einen vollständigen Querschnitt des  $n$ -dimensionalen Raumes bildet. Innerhalb einer Fläche ist der zugehörige (Teil-)Maskierungsoperator somit homogen. Ist  $\mathbf{e}_\nu$  der Normalenvektor der Fläche, so ist der Operator damit vom Typ  $U_0^\nu$ . Daran ändert sich auch nichts, wenn der Operator weitere parallele Flächen repräsentiert.

Betrachten wir jetzt die beiden Stufen im Detail. Zunächst wird  $\mathbf{L}_{\mu,a}^{\mathbf{k}}$  weiter zerlegt in disjunkte Räume  $\mathbf{L}_{\mu,a,b}^{\mathbf{k}}$ , sodass analog zu Punkt 1 gilt  $\bigcup_b \mathbf{L}_{\mu,a,b}^{\mathbf{k}} = \mathbf{L}_{\mu,a}^{\mathbf{k}}$  und  $\mathbf{L}_{\mu,a,b}^{\mathbf{k}} \cap \mathbf{L}_{\mu,a,c}^{\mathbf{k}} = \emptyset \forall b, c$  mit  $b \neq c$ . Mit  $\mathbf{L}_{\mu,a,b}^{\mathbf{k}}$  kann dann  $\mathcal{M}_{a,b}$  gebildet werden (vgl. Gleichung (5.62)), was zu

$$\sum_b \mathcal{M}_{a,b} = \mathcal{M}_a \quad (5.65)$$

führt. Diese Aufteilung muss derart gewählt werden, dass eine weitere Zerlegung in  $n - 1$  Maskierungsoperatoren  $\mathcal{M}_{a,b}^\nu \in U_0^\nu \cap U^0$  auf

die folgende Weise möglich ist:<sup>3</sup>

$$\mathcal{M}_{a,b} = \bigcirc_{\substack{\nu=1 \\ \nu \neq \mu}}^n \mathcal{M}_{a,b}^\nu, \quad \mathcal{M}_{a,b}^\nu \in U_{\mathbf{0}}^\nu \cap U^0, \quad m_{\mathbf{k},\mathbf{k}'}^{a,b,\nu} \in \{0,1\}. \quad (5.66)$$

In der Regel sind solche Aufteilungen nicht eindeutig. Abbildung 5.4 zeigt eine mögliche Aufteilung für den Maskierungsoperator  $\mathcal{M}_1$  aus dem Beispiel in Abbildung 5.3.

Nach erfolgter Aufteilung ( $\mathcal{M}_{1,1}$  und  $\mathcal{M}_{1,2}$  in Abbildung 5.4) sind die Maskierungsoperatoren  $\mathcal{M}_{a,b}^\nu$  zu berechnen. Dazu ist ein Punkt  $\mathbf{k}_{a,b}^1$  auszuwählen, für den  $m_{\mathbf{k}_{a,b}^1, \mathbf{k}_{a,b}^1}^{a,b} = 1$  gilt. Dann existiert für jede Richtung  $\nu$  eine Matrix  $\mathbf{M}_{a,b}(\mathbf{k}_{a,b}^1 \downarrow \mathbf{e}_\nu)$ , indem  $\mathcal{M}_{a,b}$  als zu  $U^\nu$  gehörig betrachtet wird (vgl. Gleichung (5.29)). Damit kann dann  $\mathcal{M}_{a,b}^\nu$  über seine (konstante) Matrix  $\mathbf{M}_{a,b}^\nu$  definiert werden:

$$\mathbf{M}_{a,b}^\nu \stackrel{!}{=} \mathbf{M}_{a,b}(\mathbf{k}_{a,b}^1 \downarrow \mathbf{e}_\nu). \quad (5.67)$$

$\mathcal{M}_{a,b}^\nu$  ist dadurch vollständig bestimmt, d. h. es müssen nicht noch andere Punkte außer  $\mathbf{k}_{a,b}^1$  betrachtet werden. In Abbildung 5.4 gilt dies für die Operatoren  $\mathcal{M}_{1,1}^x$ ,  $\mathcal{M}_{1,1}^z$ ,  $\mathcal{M}_{1,2}^x$  und  $\mathcal{M}_{1,2}^z$ . Im Einzelfall kann sich für einen Operator auch schon mal  $\mathcal{M}_{a,b}^\nu \equiv \text{id}$  ergeben. Der Operator und seine zugehörige Matrix (in diesem Fall eine Einheitsmatrix) können dann aus sämtlichen Berechnungen herausgenommen werden.

Mit dem Beispiel aus Abbildung 5.3 ergibt sich für  $\mathcal{M}_1 \mathcal{T}_1$  damit folgende Darstellung:

$$\mathcal{M}_1 \mathcal{T}_1 = (\mathcal{M}_{1,1}^x \mathcal{M}_{1,1}^z + \mathcal{M}_{1,2}^x \mathcal{M}_{1,2}^z) \mathcal{T}_1. \quad (5.68)$$

Man beachte, dass für die vollständige Darstellung von  $\mathcal{T}$  ein ähnlicher Aufwand jeweils noch für die  $\beta$ -,  $\gamma$ - und  $\delta$ -Anteile durchzuführen ist.

---

<sup>3</sup>Da  $\mathcal{M}_{a,b}^\nu \in U^0 \forall \nu$  gilt (d. h. es liegen ausschließlich Skalierungsoperatoren vor), ist die Verkettung in Gleichung (5.66) kommutativ.

$$\begin{array}{|c|c|c|c|} \hline \mathbf{I} & & \mathbf{I} & \mathbf{I} \\ \hline & & & \\ \hline \mathbf{I} & & \mathbf{I} & \\ \hline & & & \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline \mathbf{I} & & \mathbf{I} & \\ \hline & & & \\ \hline \mathbf{I} & & \mathbf{I} & \\ \hline & & & \\ \hline \end{array} + \begin{array}{|c|c|c|c|} \hline & & & \mathbf{I} \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}$$

$$\mathcal{M}_1 = \mathcal{M}_{1,1} + \mathcal{M}_{1,2}$$

$$\begin{array}{|c|c|c|c|} \hline \mathbf{I} & & \mathbf{I} & \\ \hline & & & \\ \hline \mathbf{I} & & \mathbf{I} & \\ \hline & & & \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline \mathbf{I} & & \mathbf{I} & \\ \hline \mathbf{I} & & \mathbf{I} & \\ \hline \mathbf{I} & & \mathbf{I} & \\ \hline \mathbf{I} & & \mathbf{I} & \\ \hline \end{array} \circ \begin{array}{|c|c|c|c|} \hline \mathbf{I} & \mathbf{I} & \mathbf{I} & \mathbf{I} \\ \hline & & & \\ \hline \mathbf{I} & \mathbf{I} & \mathbf{I} & \mathbf{I} \\ \hline & & & \\ \hline \end{array}$$

$$\mathcal{M}_{1,1} = \mathcal{M}_{1,1}^x \circ \mathcal{M}_{1,1}^z$$

$$\begin{array}{|c|c|c|c|} \hline & & & \mathbf{I} \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline & & & \mathbf{I} \\ \hline & & & \mathbf{I} \\ \hline & & & \mathbf{I} \\ \hline & & & \mathbf{I} \\ \hline \end{array} \circ \begin{array}{|c|c|c|c|} \hline \mathbf{I} & \mathbf{I} & \mathbf{I} & \mathbf{I} \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}$$

$$\mathcal{M}_{1,2} = \mathcal{M}_{1,2}^x \circ \mathcal{M}_{1,2}^z$$

Abbildung 5.4: Zerlegung des  $U_{1y}^y$ -Maskierungsoperators ( $\mathcal{M}_1$ ) aus Abbildung 5.3 in  $U_0^x$ - und  $U_0^z$ -Anteile.

Die Linearität sämtlicher Operatoren erlaubt im allgemeinen Fall schließlich die Darstellung

$$\tilde{\mathcal{T}} = \sum_a (\mathcal{P} \mathcal{M}_a \mathcal{P}^*) (\mathcal{P} \mathcal{T}_a \mathcal{P}^*) \quad (5.69)$$

$$= \sum_a \left( \sum_b \bigcirc_{\substack{\nu=1 \\ \nu \neq \mu_a}}^n \tilde{\mathcal{M}}_{a,b}^\nu \right) \tilde{\mathcal{T}}_a \quad (5.70)$$

mit

$$\tilde{\mathcal{M}}_{a,b}^\nu = \mathcal{P}_{0,j_\nu}^\nu \mathcal{M}_{a,b}^\nu \mathcal{P}_{j_\nu,0}^{*\nu}, \quad (5.71)$$

$$\tilde{\mathcal{T}}_a = \mathcal{P}_{0,j_{\mu_a}}^{\mu_a} \mathcal{T}_a \mathcal{P}_{j_{\mu_a},0}^{*\mu_a}. \quad (5.72)$$

Die Anwendung des in diesem Abschnitt beschriebenen Algorithmus in einer FDTD-Simulation zur Darstellung der Materialverteilung als Operator sowie die WT dieses Operators wurde erstmalig in [3] erwähnt, allerdings nur in sehr kurzer Form.<sup>4</sup> Simulationsergebnisse hierzu wurden wohl bereits in [19] präsentiert. In [18] wurden dann erstmalig Details zu dem Algorithmus für die FDTD veröffentlicht.

### 5.2.3.3 Weitere Operatoren

Abschließend werden lineare Operatoren betrachtet, für die  $\mathcal{T} \notin U^\nu \forall \nu$  gilt. Diese können mit Hilfe von Maskierungen und Translationsoperatoren in die bereits behandelten Operatoren vom Typ  $U^\nu$  zerlegt werden. Der Rotationsoperator der FDTD (s. Gleichung (2.17)) ist beispielweise ein solcher Operator. Für den Fall allgemeiner MSA-Basen fallen auch die Operatoren zur Transformation in den Waveletbereich in diese Kategorie. Dies gilt ebenso bei homogenen Tensorprodukt-Basen, wenn die eindimensionalen WT-Operatoren zu einem Gesamtoperator zusammengefasst werden.

Es sind verschiedene Arten von Zerlegungen in  $U^\nu$ -Operatoren denkbar. Eine der einfachsten soll hier kurz vorgestellt werden. Dazu wird im Folgenden von einem Vektorraum  $\mathbf{K} \subset \mathbb{Z}^n$  ausgegangen, für den das gleiche

---

<sup>4</sup>Die folgenden Anmerkungen mögen bei der Durchsicht von [3] hilfreich sein: Die Herleitung und Darstellung des Algorithmus in [3] beschränkt sich im Wesentlichen auf zwei Gleichungen, (3.112) und (3.113). Gleichung (3.112) beschreibt hierbei die Zerlegung eines Materialoperators. Allerdings kommt die Zerlegung an dieser Stelle zu früh, da der Operator noch invertiert werden muss (Materialparameter treten als Operator formuliert im Nenner auf, vgl. Gleichung (5.25) in diesem Buch). Für jede Feldkomponente  $\nu$  wird im Operator ein Tripel eindimensionaler Funktionen  $\epsilon_{\nu i}(x)$ ,  $\epsilon_{\nu i}(y)$ ,  $\epsilon_{\nu i}(z)$  benötigt. Zu beachten ist, dass jeweils zwei dieser Funktionen Maskierungselemente sind (also nur die Werte 0 oder 1 annehmen können) und die verbleibende des Tripels den Materialparameter trägt. Noch etwas komplizierter wird es, wenn leitfähige Materialien zu simulieren sind: Permittivitäts- und Leitfähigkeitsoperatoren können dann nicht mehr getrennt voneinander zerlegt werden, vgl. Gleichung (2.20) und Abschnitt 7.2 in dieser Arbeit.

gelten soll wie in Abschnitt 5.2.1 ( $\mathbf{k} \in \mathbf{K} \Leftrightarrow 1 \leq k_\nu \leq K_\nu \forall \nu$ ). Es wird nun  $\mathcal{T}$  aufgeteilt in Operatoren mit gleichem Verschiebungsvektor:<sup>5</sup>

$$\mathcal{T} = \sum_{\mathbf{m} \in \mathbf{K}} \mathcal{T}_{\mathbf{m}} , \quad (5.73)$$

$$\mathcal{T}_{\mathbf{m}} : t_{\mathbf{k}, \mathbf{k}'}^{\mathbf{m}} = \begin{cases} t_{\mathbf{k}, \mathbf{k}'} & , \mathbf{k}' = (\mathbf{k} + \mathbf{m}) \bmod \mathbf{K} \\ 0 & , \text{sonst} \end{cases} . \quad (5.74)$$

Die weitere Vorgehensweise besteht darin, die mehrdimensionalen Verschiebungen  $\mathbf{m}$  durch quasi-eindimensionale zu ersetzen.

**Definition 7.** Es seien  $a_{\mathbf{k}}$  skalare Koeffizienten über dem Vektorraum  $\mathbf{K}$ . Die eindimensionale Verschiebung der  $a_{\mathbf{k}}$  um  $m$  Positionen in Richtung  $\nu$  ist definiert durch den Operator  $\mathcal{S}_m^\nu$ :

$$\mathcal{S}_m^\nu : a_{\mathbf{k}} \mapsto a_{(\mathbf{k} + m\mathbf{e}_\nu) \bmod \mathbf{K}} , \quad m \in \mathbb{Z} . \quad (5.75)$$

Die zu  $\mathcal{S}_m^\nu$  gehörenden Tensorelemente  $s_{\mathbf{k}, \mathbf{k}'}^{m, \nu}$  ergeben sich damit zu

$$s_{\mathbf{k}, \mathbf{k}'}^{m, \nu} = \begin{cases} 1 & , l'_\nu = l_\nu \wedge k'_\nu = (k_\nu + m) \bmod K_\nu \\ 0 & , \text{sonst} \end{cases} . \quad (5.76)$$

Daher ist offensichtlich  $\mathcal{S}_m^\nu \in U_{\mathbf{0}}^\nu$ .

**Definition 8.** Es gelte Definition 7. Ferner sei  $\mathbf{m} \in \mathbf{K}$ . Die mehrdimensionale Verschiebung der  $a_{\mathbf{k}}$  ist definiert durch den Operator  $\mathcal{S}_{\mathbf{m}}$ :<sup>6</sup>

$$\mathcal{S}_{\mathbf{m}} \stackrel{\text{def}}{=} \bigcirc_{\nu=1}^n \mathcal{S}_m^\nu . \quad (5.77)$$

Offensichtlich gilt dann

$$\mathcal{S}_{\mathbf{m}} : a_{\mathbf{k}} \mapsto a_{(\mathbf{k} + \mathbf{m}) \bmod \mathbf{K}} . \quad (5.78)$$

<sup>5</sup>Eine Verschiebung um  $(0, \dots, 0)^T$  tritt wegen der modulo-Operation in Gleichung (5.74) im Falle  $\mathbf{m} = (K_1, \dots, K_n)^T$  auf.

<sup>6</sup>Die eindimensionalen Verschiebungsoperatoren sind bezüglich der Verkettung in Gleichung (5.77) kommutativ. Dies ergibt sich aus der Struktur von  $\mathbf{K}$  und der zyklischen Verschiebung der Vektoren (Vektoraddition modulo  $\mathbf{K}$ ).

Die Eigenschaft der Indexverschiebung von  $\mathcal{T}_{\mathbf{m}}$  kann also bereits mit  $\mathcal{S}_{\mathbf{m}}$  nachgebildet werden. Lediglich die skalierende bzw. gewichtende Eigenschaft von  $\mathcal{T}_{\mathbf{m}}$  ist noch als Operator zu formulieren. Dazu wird zuerst ein invertierender Vektor  $\mathbf{m}^*$  benötigt:

$$\mathbf{m}^* \stackrel{\text{def}}{=} (K_1 - m_1, \dots, K_n - m_n)^T. \quad (5.79)$$

Damit ist dann  $\mathcal{S}_{\mathbf{m}}\mathcal{S}_{\mathbf{m}^*} = \text{id}$ , d. h.  $(\mathcal{S}_{\mathbf{m}})^{-1} = \mathcal{S}_{\mathbf{m}^*}$ . Offenbar folgt dadurch  $\mathcal{S}_{\mathbf{m}^*}\mathcal{T}_{\mathbf{m}} \in U^0$ . Daher kann nun ein Operator  $\mathcal{O}_{\mathbf{m}}$  definiert werden:

$$\mathcal{O}_{\mathbf{m}} : o_{\mathbf{k},\mathbf{k}'}^{\mathbf{m}} = t_{(\mathbf{k}+\mathbf{m}^*) \bmod \mathbf{K},\mathbf{k}'}^{\mathbf{m}}. \quad (5.80)$$

Aus der Umstellung von  $\mathcal{O}_{\mathbf{m}} = \mathcal{S}_{\mathbf{m}^*}\mathcal{T}_{\mathbf{m}} \in U^0$  nach  $\mathcal{T}_{\mathbf{m}} = \mathcal{S}_{\mathbf{m}}\mathcal{O}_{\mathbf{m}}$  folgt

$$\mathcal{T} = \sum_{\mathbf{m} \in \mathbf{K}} \mathcal{S}_{m_1}^1 \cdots \mathcal{S}_{m_n}^n \mathcal{O}_{\mathbf{m}}. \quad (5.81)$$

Jeder der Teiloperatoren gehört zu einem  $U^\nu$  und lässt sich demnach entsprechend der vorangegangenen Abschnitte in den Waveletbereich transformieren.

Es sind sicherlich etwas effizientere Zerlegungen denkbar. Jedoch ist es fraglich, ob sich durch solche verbesserten Algorithmen die Anzahl der Zerlegungen deutlich reduzieren lässt. Die Problematik ist wohl eher so einzuschätzen, dass der Aufwand für die in diesem Abschnitt behandelten Operatoren in den meisten Fällen hoch bleiben dürfte.

#### 5.2.3.4 Vorgehensweise bei MSA-Basen

Zu Beginn von Abschnitt 5.2 wurde darauf hingewiesen, dass der Fokus bei der WT von Operatoren auf homogenen Tensorprodukt-Basen liegt. Diese stellen nach zwei Transformationsstufen und mehr als einer Dimension zwar keine MSA-Basen dar, erlauben aber wegen der  $U_0^\nu$ -Zugehörigkeit der zu Grunde liegenden eindimensionalen WT-Operatoren relativ einfache Matrix-basierte Transformationen von Operatoren.

Eine genauere Betrachtung von Abschnitt 5.2.2 zeigt, dass die Sätze zur Matrix-basierten Verkettung von Operatoren nicht speziell auf Wavelet-Transformationen zugeschnitten sind. Vielmehr geht es um die Darstellung der Verkettung beliebiger linearer Operatoren. Mit den Algorithmen

aus Abschnitt 5.2.3 lassen sich demnach auch die mehrdimensionalen WT-Operatoren der MSA-Basen einbringen. Hierbei sind allerdings zwei Fälle zu unterscheiden.

Bei separablen MSA-Basen nach Abschnitt 4.2 ergibt sich die Querschnittsabhängigkeit der WT-Operatoren bereits aus Abbildung 4.2. In Abschnitt 5.2.1 wurde jedoch schon erwähnt, wie stattdessen die  $U_0^v$ -WT-Operatoren homogener Tensorprodukt-Basen angewendet werden können. Dadurch ergibt sich, dass zusätzliche Maskierungsoperatoren definiert werden müssen, um bereits vollständig transformierte Bereiche auszumaskieren. Als anschauliches Beispiel soll eine weitere Transformationsstufe in Abbildung 4.2 nachvollzogen werden.

Es bilden in der Abbildung jeweils drei rechte untere Blöcke zusammen einen zweidimensionalen  $\mathbf{W}$ -Raum:

$$\mathbf{W}_i(n=2) = (\mathbf{V}_i \mathbf{W}_i \cup \mathbf{W}_i \mathbf{W}_i \cup \mathbf{W}_i \mathbf{V}_i)(n=1) . \quad (5.82)$$

Auf der linken Seite ist dabei der zweidimensionale Raum angegeben ( $n=2$ ), der sich aus eindimensionalen Räumen ( $n=1$ ) der rechten Seite zusammensetzt. Für eine weitere Skala darf nun lediglich der Raum  $\mathbf{V}_3(n=2) = (\mathbf{V}_3 \mathbf{V}_3)(n=1)$  transformiert werden. Dazu können die eindimensionalen  $U_0^v$ -WT-Operatoren genommen werden, wenn der Bildbereich durch einen Operator  $\mathcal{M}_{\mathbf{V}_3}$  auf den  $\mathbf{V}_3$  beschränkt wird, d. h. der restliche Raum wird ausmaskiert. Wesentlich ist allerdings, dass die WT-Operatoren immer nur um eine Skala weiter transformieren können. Demnach wären dies dann die Operatoren  $\mathcal{P}_{3,4}^x$  und  $\mathcal{P}_{3,4}^y$ , die zusammen

$$\mathcal{P}_{3,4}^{xy} = \mathcal{P}_{(3,3)^T, (4,4)^T} = \mathcal{P}_{3,4}^x \circ \mathcal{P}_{3,4}^y \quad (5.83)$$

bilden.

Der Anteil für die nächst höhere Skala 4 ist damit beschrieben. Zur Beibehaltung der in den Räumen  $\mathbf{W}_1$  bis  $\mathbf{W}_3$  erzielten Ergebnisse (der „restliche Raum“) muss ebenfalls ein Maskierungsoperator benutzt werden, der zu  $\mathcal{M}_{\mathbf{V}_3}$  komplementär ist.

Betrachten wir noch einmal Abbildung 4.2, ohne eine weitere Stufe 4. Die gesamte Wavelet-Transformation hierzu lautet nach dem bisher Gesagten

$$\mathcal{P}_{0,3} = \bar{\mathcal{M}}_{\mathbf{V}_1} \mathcal{P}_{0,1}^{xy} + \bar{\mathcal{M}}_{\mathbf{V}_2} \mathcal{P}_{1,2}^{xy} \mathcal{M}_{\mathbf{V}_1} \mathcal{P}_{0,1}^{xy} + \mathcal{M}_{\mathbf{V}_3} \mathcal{P}_{2,3}^{xy} \mathcal{M}_{\mathbf{V}_2} \mathcal{P}_{1,2}^{xy} \mathcal{M}_{\mathbf{V}_1} \mathcal{P}_{0,1}^{xy} \quad (5.84)$$

$$= ((\mathcal{M}_{\mathbf{V}_3} \mathcal{P}_{2,3}^{xy} \mathcal{M}_{\mathbf{V}_2} + \bar{\mathcal{M}}_{\mathbf{V}_2}) \mathcal{P}_{1,2}^{xy} \mathcal{M}_{\mathbf{V}_1} + \bar{\mathcal{M}}_{\mathbf{V}_1}) \mathcal{P}_{0,1}^{xy} , \quad (5.85)$$

wobei  $\bar{\mathcal{M}}_{\mathbf{V}_i}$  der zu  $\mathcal{M}_{\mathbf{V}_i}$  komplementäre Maskierungsoperator ist, d. h.  $\mathcal{M}_{\mathbf{V}_i} + \bar{\mathcal{M}}_{\mathbf{V}_i} = \text{id}$ .

Der zweite Fall betrifft allgemeine MSA-Basen. Die zugehörigen Transformationsoperatoren sind zwar linear, doch die fehlende Separabilität der Basen dürfte in den meisten Fällen für Matrix-basierte Darstellungen viel zu aufwendig sein.

## 5.3 Direkte Verkettung von Operatoren

Die folgenden Abschnitte präsentieren eine völlig neuartige Methodik zur Darstellung und Transformation von Operatoren. Die Notwendigkeit zu diesem Schritt folgt aus den Erkenntnissen aus Abschnitt 5.2. Dort hat es sich gezeigt, dass eine Beschränkung rein auf Matrix-Basis sehr aufwendig werden kann. Der Aufwand steigt dabei umso mehr, je inhomogener die Abbildungseigenschaften eines Operators sind. Es stellt sich hierbei stets die Aufgabe, den Operator „geschickt“ zu zerlegen, sodass so wenige Teiloperatoren wie möglich entstehen. Hierfür gibt es jedoch kein Standardverfahren. Man muss also einen mehr oder weniger aufwendigen Optimierungsalgorithmus implementieren oder aber bei jedem neuen Operator die Zerlegung händisch vorgeben. Der zunächst einleuchtende Vorteil der Anwendbarkeit einfach zu realisierender Matrix-Vektor-Multiplikationen wird somit in vielen Fällen von dem nachteiligen Zerlegungsaufwand zunichte gemacht.

Die Matrix-Darstellung wurde ursprünglich gewählt, um den mit Gleichung (5.8) implizierten Aufwand zu umgehen. Auch eine Transformation von Operatoren nach Gleichung (5.16) wirft augenscheinlich ähnliche Probleme auf. In beiden Fällen sollte man aber nicht aus den Augen verlieren, dass Operatoren in sehr vielen praktischen Fällen von weitaus geringerer Komplexität sind als die in Abschnitt 5.1.1 angegebenen  $L^{2n}$  Tensorelemente. Eine Ausführung der Gleichungen (5.8) und (5.16) über sämtliche Indizes ist dann wenig praktikabel. Daher stellt sich die Frage, wie eine effiziente Implementierung der Tensorelemente und der Verkettung zweier Operatoren aussehen kann. Dabei bezieht sich die Implementierung auf die Entwicklung von Speicherstrukturen und Algorithmen für ein Computerprogramm.

Die Vorteile, die sich aus einer direkten Tensor-Darstellung ergeben, liegen auf der Hand: Allgemeine Operatoren nach Abschnitt 5.2.3.3 müssen nicht mehr in Translations- und Maskierungsoperatoren zerlegt werden. Operatoren vom Typ  $U_{\mathbf{l}}^{\nu}$  müssen nicht mehr in solche vom Typ  $U_{\mathbf{0}}^{\nu}$  zerlegt werden. Überhaupt ist die Frage nach möglichst optimalen Zerlegungen nicht mehr relevant. Stattdessen können beliebige Operatoren „einfach so“ benutzt und miteinander verkettet werden.<sup>7</sup> Damit steigen auch die Freiheiten bei der Auswahl von Transformationen deutlich.

Professionelle Programme zur symbolischen oder numerischen Mathematik<sup>8</sup> können für diese Aufgabenstellung in der Regel nicht benutzt werden. Für kleinere Probleme lassen sie sich zwar einsetzen, bei größeren Anordnungen sind jedoch spezielle Speicherstrukturen und Algorithmen vonnöten. Auch in einigen Mathematik-Newsgroups<sup>9</sup> fand sich kein Hinweis auf Lösungsansätze oder erhältliche Programme. In den folgenden Abschnitten werden daher neue Strukturen und Algorithmen entwickelt, mit denen schließlich auch Operatoren am Computer berechnet wurden.

Neben der effizienten Speicherung der Daten liegt das Hauptaugenmerk vor allem auf der schnellen Berechnung der verketteten Operatoren. Wie später noch zu sehen ist, können Wavelet-transformierte FDTD-Operatoren überraschend komplexe Abbildungseigenschaften aufweisen. Dies hat je nach Art der verwendeten Algorithmen zur Berechnung der Verkettung erhebliche Auswirkungen auf die Dauer der Berechnung. Aus diesem Grund wird verschiedentlich auf die Komplexität bzw. Ordnung von Algorithmen eingegangen werden.

### 5.3.1 Elementare Abbildungen

Bei der Frage nach der Darstellung eines vollständigen Operators im Computer sind die Aspekte Speicheraufwand, Rechenzeit, Flexibilität und Einfachheit zu berücksichtigen. Treten bei einem ausgewählten Arbeitsgebiet beispielsweise ausschließlich  $U_{\mathbf{0}}^{\nu}$ -Operatoren auf, so sind die in Abschnitt 5.2 diskutierten Matrix-basierten Algorithmen sicherlich flexibel genug

---

<sup>7</sup>Das gilt zumindest so lange, wie ihre Komplexität dies erlaubt.

<sup>8</sup>Beispielsweise *Mathematica*, *Maple*, *MATLAB* oder *MuPAD*.

<sup>9</sup>Gesucht wurde in de.sci.mathematik, sci.math und sci.math.num-analysis.

und genügen auch dem Anspruch nach einfacher Realisierbarkeit. Bezogen auf Speicheraufwand und Rechenzeit stellen reine Matrizen allerdings keine optimale Lösung dar, wenn diese dünn besetzt sind.

Die Beschäftigung mit Matrizen bietet dennoch einen Anhaltspunkt für die konkrete Ausgestaltung der Speicherstrukturen eines Operators. Viele Operatoren lassen sich in der Praxis mit zirkulären Matrizen darstellen. Es liegt nahe, sich die Abbildungseigenschaften solcher Operatoren für eine effiziente Implementierung zunutze zu machen. Schreiben wir zunächst Gleichung (5.8) in eine äquivalente Form um:

$$b_{\mathbf{k}} = \sum_{\Delta \mathbf{k}} t_{\mathbf{k}, \mathbf{k} + \Delta \mathbf{k}} a_{\mathbf{k} + \Delta \mathbf{k}} = \sum_{\Delta \mathbf{k}} t_{\mathbf{k}}(\Delta \mathbf{k}) a_{\mathbf{k} + \Delta \mathbf{k}} . \quad (5.86)$$

Bei einem  $U_0^\nu$ -Operator gilt dann

$$t_{\mathbf{k}}(\Delta \mathbf{k}) = t_{\mathbf{k} + c\mathbf{e}_\mu}(\Delta \mathbf{k}) \quad \forall \mathbf{k}, c, \mu \quad \text{mit} \quad \mu \neq \nu , \quad (5.87)$$

was in diesem Fall der  $\mathbf{l}_\nu$ -Unabhängigkeit der Operatormatrix entspricht.<sup>10</sup> Selbst wenn eine Operatormatrix von einer Querschnittsposition  $\mathbf{l}_\nu$  abhängt, so ist sie doch oft stückweise konstant. Daher ist es sinnvoll, alle  $\mathbf{k}'$  zu identifizieren, für die

$$b_{\mathbf{k}'} = \sum_{\Delta \mathbf{k}} t_{\mathbf{k}}(\Delta \mathbf{k}) a_{\mathbf{k}' + \Delta \mathbf{k}} \quad (5.88)$$

gilt, d. h. bei denen ebenfalls  $t_{\mathbf{k}}(\Delta \mathbf{k})$  statt  $t_{\mathbf{k}'}(\Delta \mathbf{k})$  benutzt werden kann. Das ist auch dann von Nutzen, wenn die Operatormatrix zwar mit  $\mathbf{l}_\nu$  stark variiert, dafür aber an vielen Stellen wenigstens stückweise zirkulär ist. In diesem Fall gilt nämlich häufig  $t_{\mathbf{k}}(\Delta \mathbf{k}) = t_{\mathbf{k} + c\mathbf{e}_\nu}(\Delta \mathbf{k})$ . Und auch bei allgemeinen Operatoren, die sich keinem  $U^\nu$  zuordnen lassen, herrschen oftmals ähnliche Abbildungseigenschaften zwischen verschiedenen Bildpunkten vor.

Betrachten wir einen Bildpunkt  $\mathbf{k}$ . Es werden nun alle Urbildpunkte  $\mathbf{k}'$  gesucht, für die  $t_{\mathbf{k}, \mathbf{k}'} \neq 0$  gilt. Daraus wird eine Liste mit Abbildungselementen gebildet. Die Größe dieser Liste sei  $I_t(\mathbf{k})$ . Jedes Listenelement  $O_i^1(\mathbf{k})$

<sup>10</sup>Genau genommen ist dann  $\Delta \mathbf{k} = \Delta k_\nu \mathbf{e}_\nu$ , doch auf diese Einschränkung kommt es hier nicht an.

besteht aus einem Faktor  $t_i$  und einem Verschiebungsvektor  $\Delta \mathbf{k}_i = \mathbf{k}'_i - \mathbf{k}$ , d. h.

$$O_i^1(\mathbf{k}) \stackrel{\text{def}}{=} (t_i, \Delta \mathbf{k}_i)(\mathbf{k}) \quad (5.89)$$

mit

$$t_i(\mathbf{k}) = t_{\mathbf{k}, \mathbf{k} + \Delta \mathbf{k}_i(\mathbf{k})} \neq 0, \quad (5.90)$$

$$i \in \{1, \dots, I_t(\mathbf{k})\}. \quad (5.91)$$

Dann gilt

$$b_{\mathbf{k}} = \begin{cases} \sum_{i=1}^{I_t(\mathbf{k})} t_i(\mathbf{k}) a_{\mathbf{k} + \Delta \mathbf{k}_i(\mathbf{k})} & , I_t(\mathbf{k}) > 0 \\ 0 & , I_t(\mathbf{k}) = 0 \end{cases}. \quad (5.92)$$

Die vollständige Abbildungsliste für einen Bildpunkt wird als  $O^2$ -Liste bezeichnet:

$$O^2(\mathbf{k}) \stackrel{\text{def}}{=} (O_1^1, \dots, O_{I_t}^1)(\mathbf{k}). \quad (5.93)$$

Damit stellt die Menge aller  $O^2(\mathbf{k})$  eine äquivalente Darstellung zur Menge aller  $t_{\mathbf{k}, \mathbf{k}'}$  dar. Der obere und mittlere Teil von Abbildung 5.5 stellen dies für einen einzelnen Bildpunkt dar.

Zwei Bildpunkte  $\mathbf{k}^1, \mathbf{k}^2$  haben dann die gleichen Abbildungseigenschaften, wenn  $O^2(\mathbf{k}^1) = O^2(\mathbf{k}^2)$  gilt. Durch Vergleich aller  $O^2(\mathbf{k})$  miteinander lässt sich so eine Liste

$$O^3 \stackrel{\text{def}}{=} (O_{I_1}^2, \dots, O_{I_t^2}^2) \quad (5.94)$$

aller benötigten  $O_{\mu}^2$  bilden. Die Größe dieser Liste ist  $I_t^2$ . Nach dem oben Gesagten sollte diese Liste möglichst keine Redundanzen haben, d. h. es sollte  $O_{\mu}^2 \neq O_{\nu}^2 \forall \mu, \nu$  mit  $\mu \neq \nu$  gelten. Dadurch kann jedem Bildpunkt  $\mathbf{k}$  ein Index  $\mu$  zugewiesen werden, sodass mit  $O_{\mu}^2 \in O^3$  die benötigten Informationen zur Berechnung von  $b_{\mathbf{k}}$  zur Verfügung stehen. Im unteren Teil von Abbildung 5.5 werden die Zusammenhänge graphisch veranschaulicht.

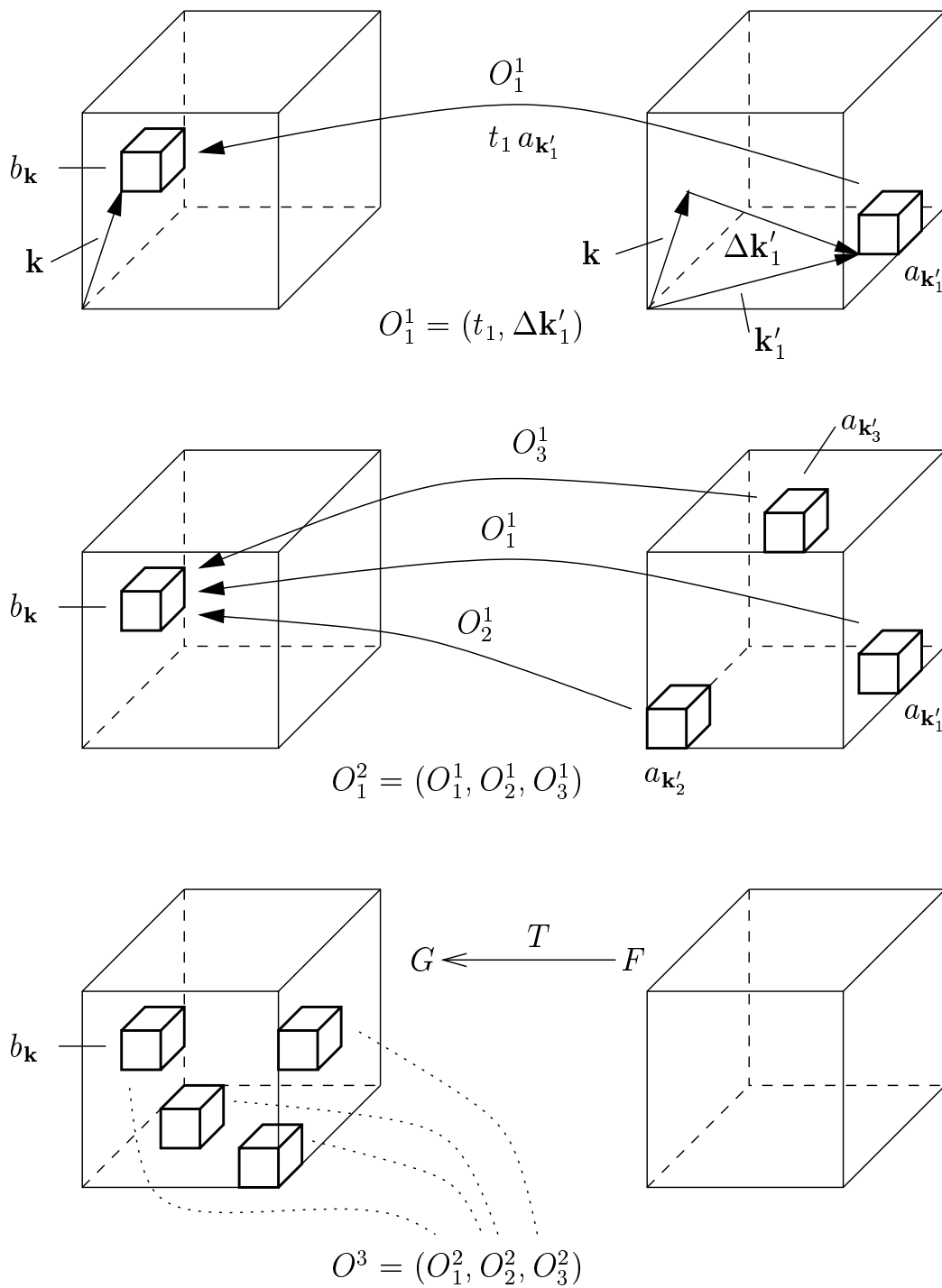


Abbildung 5.5: Listen-basierte Darstellung einer diskreten Abbildung  $T$  im dreidimensionalen Raum. Oben: Einzelnes Abbildungselement  $O_1^1$  als Ersatz für  $t_{\mathbf{k}, \mathbf{k}'_1}$ . Mitte: Vollständige Berechnung eines Bildelementes  $b_{\mathbf{k}}$  durch  $O_1^2$ . Unten: Referenzierung auf  $O_\mu^2$  für jedes Bildelement.

### 5.3.2 Aufbau einer $O^3$ -Liste

Bei der Implementierung eines Operators im Computer sind entsprechend den Angaben des vorangegangenen Abschnitts für alle Bildpunkte die zugehörigen  $O^2$ -Listen zu definieren. Damit die hieraus entstehende  $O^3$ -Liste redundanzfrei ist, muss die Frage nach einem effizienten Vergleich zweier  $O^2$ -Listen geklärt werden. Ein weiterer wesentlicher Aspekt ist, auf welche Art möglichst schnell in einer  $O^3$ -Liste eine  $O^2$ -Liste mit bestimmten Abbildungseigenschaften gefunden werden kann. Diese beiden Probleme gehören mit zu den häufigsten Operationen, die bei der Durchführung einer Verkettung auftreten können, vgl. Abschnitt 5.3.3. In den beiden folgenden Abschnitten wird ein Lösungsansatz hierzu vorgestellt.

#### 5.3.2.1 Vergleich von $O^2$ -Listen

Zwei Listen  $O_\mu^2, O_\nu^2$  können nur dann gleich sein, wenn sie gleich groß sind, d. h. wenn  $I_{t,\mu} = I_{t,\nu}$  gilt. Ist dies der Fall, sind alle  $O^1$ -Elemente zwischen beiden Listen miteinander zu vergleichen. Geht man von einer unsortierten Reihenfolge aller  $O^1$ -Elemente aus, so hat der Vergleich eine Komplexität von der Ordnung  $O((I_t)^2)$ .

Bei FDTD-Operatoren gilt in den meisten Fällen  $I_t = 1$  oder  $I_t = 2$ , sodass hier kein Bedarf an einer Reduktion der Komplexität besteht. Im Falle von WT-Operatoren ist die Größe von  $I_t$  von der verwendeten Skalierungsfunktion abhängig, vgl. Abschnitt 3.2.3. Damit besteht hier allenfalls bedingt ein Optimierungsbedarf. Transformiert man jedoch FDTD-Operatoren in den Waveletbereich, d. h. führt man die Verkettung  $\mathcal{P} \mathcal{T} \mathcal{P}^*$  durch, so kann sich eine ganz andere Situation ergeben. Es wird später noch zu sehen sein, dass im Gesamtoperator Einzelabbildungen mit  $I_t > 50$  auftreten können. Da der Vergleich zweier  $O^2$ -Listen bei großen Operatoren sehr häufig durchgeführt werden muss, ist die Optimierung der Vergleichsprozedur unabdingbar.

Als Lösungsansatz bietet sich die Sortierung der  $O^1$ -Elemente an. Da jeder Verschiebungsvektor  $\Delta \mathbf{k}^i$  innerhalb einer  $O^2$ -Liste höchstens einmal auftreten kann, soll hierfür ein Ordnungskriterium definiert werden:

**Definition 9.** Es seien  $\Delta \mathbf{k}^1, \Delta \mathbf{k}^2 \in \mathbb{Z}^n$ . Dann ist  $\Delta \mathbf{k}^1 < \Delta \mathbf{k}^2$ , wenn ein

$j \in \{1, \dots, n\}$  existiert, sodass gilt

$$\Delta k_j^1 < \Delta k_j^2 \quad \wedge \quad \Delta k_l^1 = \Delta k_l^2 \quad \forall l > j. \quad (5.95)$$

Analoges gilt für  $\Delta \mathbf{k}^1 > \Delta \mathbf{k}^2$ . Gilt weder  $\Delta \mathbf{k}^1 < \Delta \mathbf{k}^2$  noch  $\Delta \mathbf{k}^1 > \Delta \mathbf{k}^2$ , so ist  $\Delta \mathbf{k}^1 = \Delta \mathbf{k}^2$ .

Der Vergleich zwischen zwei sortierten Listen  $O_\mu^2$  und  $O_\nu^2$  besteht dann aus den Vergleichen zwischen  $O_{\mu,i}^1$  und  $O_{\nu,i}^1$ , mit  $i \in \{1, \dots, I_t\}$ . Da in jedem  $O^1$ -Element außer einem Vektor nur noch ein Faktor enthalten ist, hat der Vergleich von  $O_\mu^2$  und  $O_\nu^2$  lediglich eine Komplexität von der Ordnung  $O(I_t)$ . Sind  $O_\mu^2$  und  $O_\nu^2$  jedoch nicht sortiert, ist noch der Aufwand für eine Sortierung zu berücksichtigen. Hierzu existieren Verfahren mit einer Ordnung von  $O(I_t \log I_t)$ . In Abschnitt B.1 wird darauf näher eingegangen.

Die Umsetzung des Vergleichs der Verschiebungsvektoren zweier  $O^1$ -Elemente stellt im Hinblick auf die Genauigkeit von Prozessoren kein Problem dar, da hier nur ganze Zahlen miteinander verglichen werden. Bei der Berechnung von Faktoren dagegen können prinzipbedingt numerische Ungenauigkeiten auftreten. Die Untersuchung zweier Faktoren auf Gleichheit sollte daher mit einer gewissen Toleranz durchgeführt werden. Abschnitt B.2 beschäftigt sich mit diesem Thema.

### 5.3.2.2 Sortierung von $O^3$ -Listen

Beim Aufbau einer  $O^3$ -Liste werden nach und nach  $O^2$ -Listen hinzugefügt. Um die Redundanzfreiheit der  $O^3$ -Liste zu gewährleisten, ist mit jeder neuen  $O^2$ -Liste zu prüfen, ob diese nicht bereits in der  $O^3$ -Liste vorhanden ist. Der Aufwand dieser Prozedur ist im Allgemeinen von der Ordnung  $O((I_t^2)^2)$  (der Aufwand für den Vergleich zweier  $O^2$ -Listen wurde hier zu Eins gesetzt). Die Transformation eines FDTD-Operators in den Waveletbereich kann eine  $O^3$ -Liste mit mehreren hundert oder tausend  $O^2$ -Listen ergeben. Daher sollte jede  $O^3$ -Liste möglichst so aufgebaut werden, dass sie sich in Bereiche mit ähnlichen Abbildungseigenschaften aufteilen lässt. Auf diese Weise kann der Aufwand bei der Überprüfung neuer  $O^2$ -Listen reduziert werden. Folgender Ansatz wurde für die vorliegende Arbeit realisiert und ist in Abbildung 5.6 visualisiert:

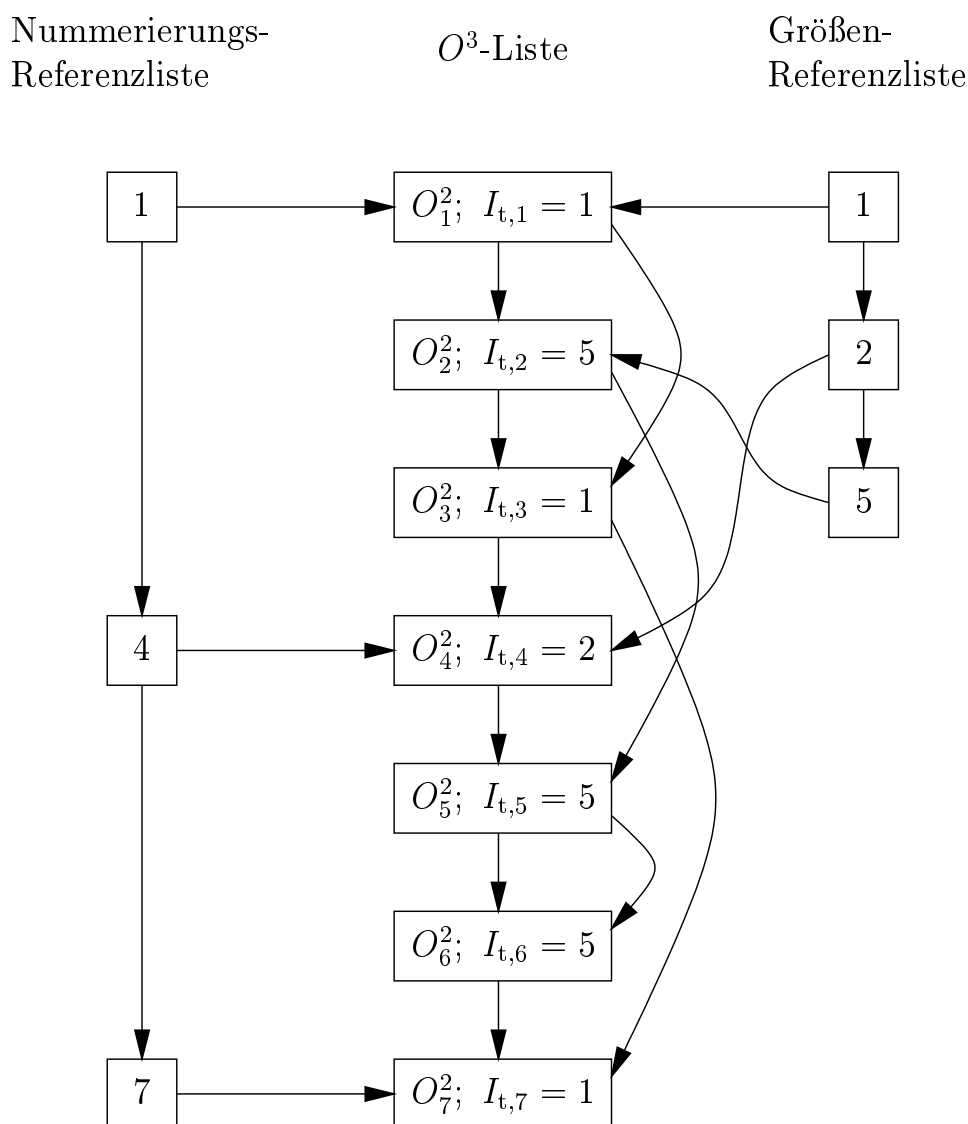


Abbildung 5.6: Aufbau einer  $O^3$ -Liste und Referenzlisten. Der besseren Übersicht wegen zeigt die Nummerierungs-Referenzliste im Bild – abweichend von den Angaben im Text – auf jede dritte  $O^2$ -Liste.

- $O^3$ -Listen sind als sog. *verkettete Listen* aufgebaut. Wegen der teilweise sehr unterschiedlichen Größen der enthaltenen  $O^2$ -Listen ist eine solche Implementierung einfacher und flexibler als eine Feld- bzw. Vektor-basierte Strukturierung.
- Die  $O^2$ -Listen in einer  $O^3$ -Liste sind aufsteigend nummeriert. Diese Nummern werden zur Referenzierung benötigt, vgl. Abbildung 5.5.

- Neu aufzunehmende  $O^2$ -Listen werden an das Ende der  $O^3$ -Liste angehängen. Alternativ wäre eine Einordnung in den Bereich der  $O^3$ -Liste denkbar, in dem sich  $O^2$ -Listen mit ähnlichen Abbildungseigenschaften befinden. Das würde jedoch eine sehr zeitaufwendige Ummummerierung erfordern.
- Eine Nummerierungs-Referenzliste zeigt auf jede achte  $O^2$ -Liste. Dadurch gelangt man schneller zu  $O^2$ -Listen mit höheren Nummern. Ein direkter Zugriff auf eine  $O^2$ -Liste mit einer bestimmten Nummer wäre nur bei einer Feld- bzw. Vektor-basierten Strukturierung der  $O^3$ -Liste möglich (s. o.).
- Eine Größen-Referenzliste besteht aus einer streng monoton steigenden Folge aller in der  $O^3$ -Liste vorkommenden Größen der  $O^2$ -Listen (die Menge aller  $I_{t,\nu}$ ; z. B. (1, 2, 5)). Für jede Größe ist in der Referenzliste die erste  $O^2$ -Liste mit dieser Größe eingetragen.
- Jede  $O^2$ -Liste enthält einen Verweis auf die nächste  $O^2$ -Liste gleicher Größe.

Im Vergleich zu einer einfachen verketteten Liste ist diese Implementierung sehr aufwendig. Insbesondere die kontinuierliche Aufrechterhaltung der Synchronizität der drei aufeinander verweisenden Listen<sup>11</sup> stellt eine recht hohe Anforderung dar. Dennoch erwies sich die Umsetzung als vorteilhaft: Die Einführung der Referenzlisten hat in einigen Versuchen die Zeit für den Aufbau einer  $O^3$ -Liste drastisch reduziert. Der Grad der Beschleunigung hängt allerdings stark von den Inhalten der  $O^3$ -Liste ab, d. h. von den Eigenschaften des Operators, sodass keine konkreten Werte angegeben werden können.

Fasst man alle  $O^2$ -Listen gleicher Größe zu einer eigenständigen Liste  $O^{3'}$  zusammen, so ist leicht einzusehen, dass der Vergleich einer neuen  $O^2$ -Liste mit allen  $O^2$ -Listen aus  $O^{3'}$  von der Ordnung  $O((I_t^{2'})^2)$  ist. Die hier vorgeschlagene Methode mit den Referenzlisten ist demnach nur bedingt von Nutzen. Sie ist dann ausreichend, wenn nur wenige  $O^2$ -Listen vorhanden sind. Mit zunehmender Inhomogenität der Abbildungseigenschaften eines Operators empfehlen sich jedoch effizientere Verfahren.

---

<sup>11</sup>In Abbildung 5.6 sind alle Verweise unidirektional eingezeichnet. Im entwickelten Computerprogramm jedoch wurden viele Verweise bidirektional ausgeführt. Dies beschleunigt teilweise den Wechsel von einer Liste auf eine andere.

### 5.3.3 Durchführung einer Verkettung

Als Beispiel für die Verkettung zweier Operatoren soll ein Teil von Gleichung (5.18) betrachtet werden. Genauer gesagt geht es um die Berechnung des Zwischenoperators

$$\mathcal{T}' = \mathcal{T} \mathcal{P}_{j\nu,0}^{*\nu} . \quad (5.96)$$

Die zugehörigen Tensorelemente berechnen sich dann zu

$$t'_{\mathbf{k}^2, \mathbf{k}^4} = \sum_{\mathbf{k}^3} t_{\mathbf{k}^2, \mathbf{k}^3} p_{\mathbf{k}^3, \mathbf{k}^4}^{*j\nu, 0} . \quad (5.97)$$

Ein Zwischenergebnis für das Feld ist demnach durch

$$a_{\mathbf{k}^2}^2 = \sum_{\mathbf{k}^4} t'_{\mathbf{k}^2, \mathbf{k}^4} b_{\mathbf{k}^4}^1 \quad (5.98)$$

bestimmt. Die Verkettung läuft dann in der zugehörigen Listen-Darstellung für jeden Bildpunkt  $\mathbf{k}^2$  folgendermaßen ab:

1. Ermittlung von  $O^{2, \mathcal{T}}(\mathbf{k}^2)$ .
2. Iteration über alle  $O_i^{1, \mathcal{T}}(\mathbf{k}^2)$ ,  $i \in \{1, \dots, I_t^{\mathcal{T}}(\mathbf{k}^2)\}$ .
  - (a) Berechnung von  $\mathbf{k}^3 = \mathbf{k}^2 + \Delta \mathbf{k}_i^{\mathcal{T}}$
  - (b) Ermittlung von  $O^{2, \mathcal{P}_{j\nu,0}^{*\nu}}(\mathbf{k}^3)$ .
  - (c) Iteration über alle  $O_j^{1, \mathcal{P}_{j\nu,0}^{*\nu}}(\mathbf{k}^3)$ ,  $j \in \{1, \dots, I_t^{\mathcal{P}_{j\nu,0}^{*\nu}}(\mathbf{k}^3)\}$ .
    - i. Berechnung von  $\Delta \mathbf{k}_{i,j}^{\mathcal{T}'} = \Delta \mathbf{k}_i^{\mathcal{T}} + \Delta \mathbf{k}_j^{\mathcal{P}_{j\nu,0}^{*\nu}}$ .
    - ii. Berechnung von  $t_{i,j}^{\mathcal{T}'} = t_i^{\mathcal{T}} \cdot t_j^{\mathcal{P}_{j\nu,0}^{*\nu}}$ .
    - iii. Hinzunahme von  $O^{1, \mathcal{T}'} = (t_{i,j}^{\mathcal{T}'}, \Delta \mathbf{k}_{i,j}^{\mathcal{T}'})$  zu  $O^{2, \mathcal{T}'}(\mathbf{k}^2)$ .
3. Sortierung von  $O^{2, \mathcal{T}'}(\mathbf{k}^2)$ . Hierbei können noch durch die Verkettung entstandene  $O^1$ -Elemente mit gleichem Verschiebungsvektor vorhanden sein.
4. Zusammenfassung aller  $O_l^{1, \mathcal{T}'}(\mathbf{k}^2)$  mit gleichem Verschiebungsvektor (nur ein  $O^1$ -Element mit diesem Vektor und der Summe der Faktoren bleibt übrig).

5. Entfernung aller  $O_l^{1, \mathcal{T}'}(\mathbf{k}^2)$  bei denen der Faktor den Wert Null hat. Das ist vor allem deshalb wichtig, damit funktional gleiche  $O^2$ -Listen auch als gleich erkannt werden können.
6. Suche nach  $O^{2, \mathcal{T}'}(\mathbf{k}^2)$  in  $O^{3, \mathcal{T}'}$ ; als neue Liste hinzunehmen, wenn die Suche erfolglos war. Das Ergebnis ist die Nummer  $m$  der  $O^2$ -Liste in  $O^{3, \mathcal{T}'}$ .
7. Eintragung der Referenznummer  $m$  für den Bildpunkt  $\mathbf{k}^2$ .

Das vorgestellte Verfahren ist sehr schematisch und bietet noch einige Optimierungsmöglichkeiten. Das gilt vor allem für den Fall, dass die miteinander zu verkettenden Operatoren vorwiegend homogene Abbildungseigenschaften haben.

### 5.3.4 Operatoren und Verkettungen bei Vektorfeldern

Bisher wurde immer nur von der Abbildung zwischen skalaren Feldern im mehrdimensionalen Raum gesprochen. Bei der FDTD beispielsweise werden aber Vektorfelder aufeinander abgebildet. Wegen der Linearität der FDTD-Operatoren können diese auf einzelne skalare Operatoren aufgeteilt werden, sodass skalare Abbildungen zwischen verschiedenen Feldkomponenten realisiert werden. Es sei  $d$  die Anzahl der Feldkomponenten. Dann wird

$$\mathbf{g} = \mathcal{T} \mathbf{f} \quad (5.99)$$

mit

$$\mathbf{g}, \mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^d, \quad \mathbf{g} = (g_1, \dots, g_d)^T, \quad \mathbf{f} = (f_1, \dots, f_d)^T, \quad (5.100)$$

dargestellt als

$$g_\mu = \sum_{\nu=1}^d \mathcal{T}_\nu^\mu f_\nu. \quad (5.101)$$

Entsprechend können die diskreten Abbildungen für die Felder der Entwicklungskoeffizienten angegeben werden:

$$T_\nu^\mu : F_\nu \mapsto G_\mu . \quad (5.102)$$

Bei einer skalaren Abbildung bezeichnet  $m(\mathbf{k})$  die Referenznummer der  $O^2$ -Liste, welche die Berechnungsparameter für das Bildelement  $b_{\mathbf{k}}$  enthält ( $O_{m(\mathbf{k})}^2$ ). Bei Vektorfeldern ist für jedes skalare Urbildfeld eine Referenznummer anzugeben. Die Berechnung der skalaren Komponente  $b_{\mathbf{k}}^\mu$  ist dann bestimmt durch einen Referenzvektor  $(m_1^\mu, \dots, m_d^\mu)^\top(\mathbf{k})$ . Abbildung 5.7 veranschaulicht die Verhältnisse.

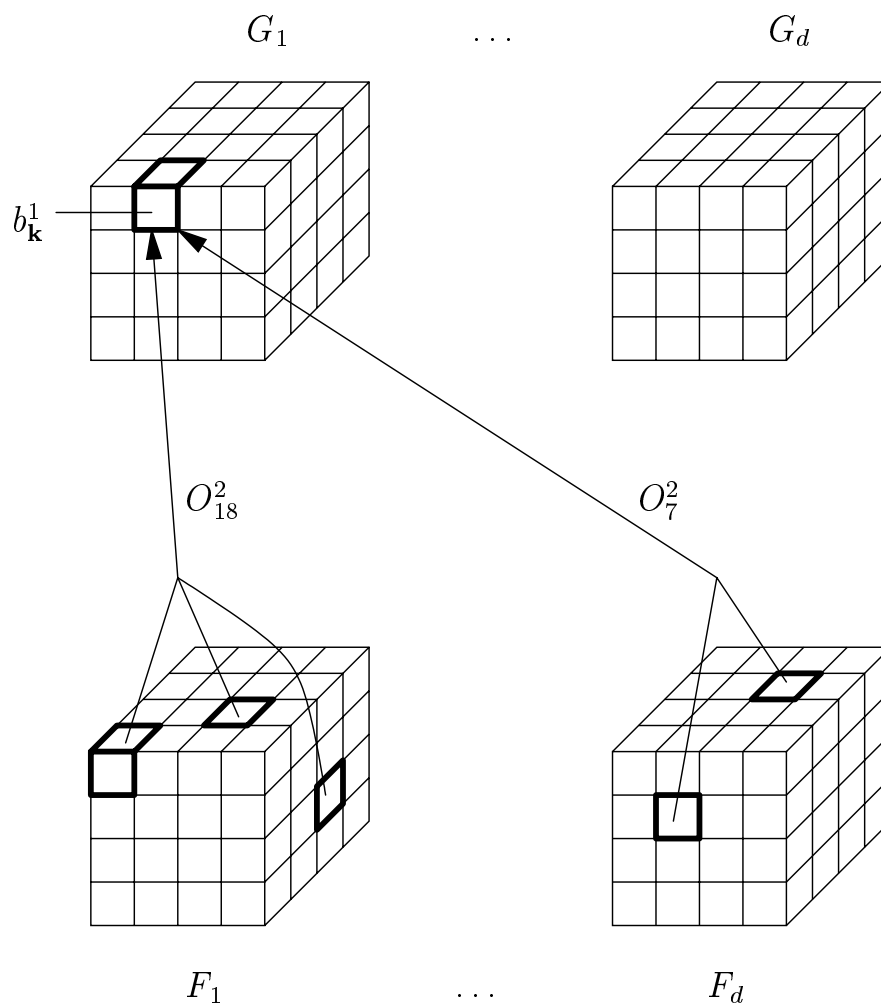


Abbildung 5.7: Beispiel einer diskretisierten Vektorfeld-Abbildung auf Listen-Basis. Für den ausgewählten Ortsvektor  $\mathbf{k}$  ist die Berechnung des  $F_1$ -Anteils für  $b_{\mathbf{k}}^1$  durch  $O_{18}^2$  angegeben. Der  $F_d$ -Anteil wird mit  $O_7^2$  berechnet.

# Kapitel 6

## Operatoren im Simulationsprogramm

In den vorangegangenen Kapiteln wurden die Eigenschaften von Operatoren beschrieben, verschiedene Möglichkeiten ihrer Darstellung aufgezeigt sowie Algorithmen zur Transformation bzw. Verkettung von Operatoren entwickelt. Mit diesen Informationen ist es möglich, Computerprogramme zu realisieren, die beispielsweise eine FDTD-Simulation im Waveletbereich ausführen. So entstand auch im Laufe dieser Arbeit die erste Version des Listen-basierten Simulationsprogramms. Erste Simulationen ergaben die Notwendigkeit, die Berechnungen mit den Operatoren zu optimieren. Die hier vorgestellten Grundlagen bildeten die Basis für die durchgeführten Verbesserungen. Weitere Optimierungsmöglichkeiten werden anhand von Analysen in Abschnitt 8.2.3 entwickelt.

### 6.1 Speicherarten und Zugriffszeiten

In modernen Rechnern sind Zugriffe von Recheneinheiten (ALU<sup>1</sup>, FPU) auf Speicherbereiche mehrstufig ausgeführt. Abbildung 6.1 zeigt dies schematisch, wobei auf die Darstellung von Registern und nichtflüchtigen Speichern verzichtet wurde.

---

<sup>1</sup>*Arithmetical Logical Unit.*

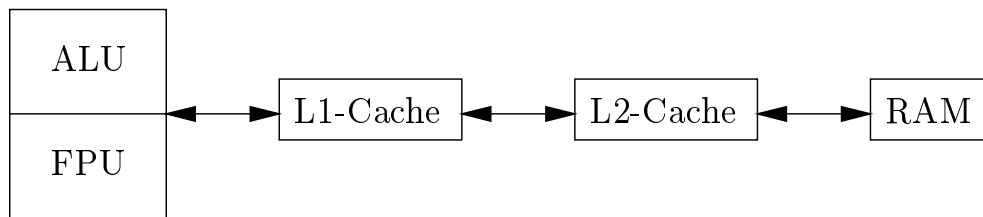


Abbildung 6.1: Datentransfer zwischen Recheneinheiten (ALU, FPU) und Speichereinheiten. Zur rechten Seite hin steigen Zugriffszeiten und Größe der Speichereinheiten an.

L1- und L2-Caches dienen der Beschleunigung von Speicherzugriffen. Je größer Speicherbausteine sind, desto langsamer laufen Zugriffe auf die Speicherinhalte ab. Darüber hinaus läuft die Anbindung Prozessor-externer RAM-Bausteine meist über einen vergleichsweise langsamen Datenbus. Daher können mit RAM-Bausteinen zwar sehr große Mengen an Daten gespeichert werden, allerdings erkauft man sich damit auch höhere Zugriffszeiten.

Die Größe von L1-Caches liegt typischerweise im ein- oder zweistelligen Kilobyte-Bereich. Da sie ein monolithischer Bestandteil von CPUs sind, benötigen Signale zwischen Speicher- und Recheneinheiten sehr viel weniger Zeit als im Falle externer RAM-Bausteine. Weiterhin hat der CPU-interne Datenbus eine weitaus höhere Übertragungsrate.

L2-Caches haben bei PCs meistens eine Größe im dreistelligen Kilobyte-Bereich. Sie sind entweder als Hybrid-Schaltung im Prozessorgehäuse oder aber außerhalb des Prozessorgehäuses angebracht. In jedem Fall liegen sie allerdings deutlich näher an den Prozessoreinheiten als RAM-Bausteine. Auch sind ihre Zugriffszeiten deutlich kürzer.

Der gesamte Speicherbedarf für eine FDTD-Simulation kann bei genügender Größe des diskretisierten Raumes im zwei- oder dreistelligen Megabyte-Bereich liegen. In diesem Fall sind dann nicht die arithmetischen Operationen, sondern Speicherzugriffe der begrenzende Faktor bezüglich der Rechengeschwindigkeit. Zur Verdeutlichung soll Gleichung (2.20) dienen. Werden alle (zeitlich konstanten) Vorfaktoren vor Beginn der Simulation berechnet, kann die Berechnung von  $E_x^{n+\frac{1}{2}}(i, j, k)$  auf drei Multiplikationen und vier Additionen reduziert werden. Außerdem müssen acht Lesezugriffe (drei Faktoren und fünf Feldwerte bzw. Ent-

wicklungskoeffizienten) und ein Schreibzugriff auf den Speicher ausgeführt werden.

Je nach Implementierung des FDTD-Algorithmus können sich einige der zu lesenden Feldwerte noch im L1-Cache befinden. Dennoch sollte möglichst auch eine Optimierung in der Form angestrebt werden, dass der zu simulierende Feldraum in Bereiche mit örtlich konstanten Faktoren aufgeteilt wird, sodass das Lesen der Faktoren immer nur zu Beginn eines jeden Bereiches erfolgen muss.

Es wurde im vorangegangenen Kapitel bereits erwähnt, dass die Abbildungseigenschaften Wavelet-transformierter FDTD-Operatoren deutlich komplexer ausfallen können als im untransformierten Fall. So ist zwar die Anzahl insgesamt zu berechnender Entwicklungskoeffizienten unverändert (sie kann durch Kompressionsverfahren allenfalls sinken, siehe z. B. Abschnitte 7.2.5.1 und 7.3.6), doch die Anzahl arithmetischer Operationen pro Entwicklungskoeffizient steigt. Auch sind die Abbildungseigenschaften insgesamt inhomogener, d. h. dass unter anderem die Bereiche kleiner ausfallen, innerhalb derer Faktoren konstant sind.

Die Komplexität der Operatoren ist auch im Hinblick auf die Verteilung der Urbildkoeffizienten im Raum zu sehen. Eine konventionelle FDTD lässt sich immer so implementieren, dass wenigstens zwei der Urbildkoeffizienten bezüglich ihrer Speicheradressen direkt hintereinander liegen. Dadurch können sie mit einem sog. *Burst*<sup>2</sup> schneller aus dem RAM geladen werden als durch zwei einzelne Ladevorgänge. Bei räumlich sehr verteilten Urbildkoeffizienten dagegen, wie es bei Wavelet-transformierten FDTD-Operatoren der Fall sein kann, sinken i. d. R. die Möglichkeiten zur Ausnutzung von Bursts.

Alles in allem sollten daher Operatoren (nachdem sie einmal definiert bzw. berechnet wurden) in der Weise in einem Simulationsprogramm ausgeführt werden, dass L1- und L2-Caches möglichst effizient ausgenutzt und Zugriffe auf langsames RAM reduziert werden. Zusätzlich sollte versucht werden, Speicherzugriffe mit Bursts zu ermöglichen.

---

<sup>2</sup>Einen Einstieg in diese Technik bieten z. B. [20] und [21].

## 6.2 Berechnungen mit Operatormatrizen

In [3] wurden erstmalig FDTD-Operatoren für einfache Simulationsanordnungen einer WT unterzogen. Dieser Abschnitt geht vor allem auf Operatormatrizen ein, wie sie in [3] untersucht wurden.

In der genannten Arbeit wurde festgestellt, dass die Wavelet-transformierten Material- und Ableitungsmatrizen dünn besetzt sind. Das gilt vor allem für den Fall, dass Matrixelemente mit einem Betrag unterhalb eines vertretbaren Schwellenwertes zu Null gesetzt werden. Entsprechend können bei Matrix-Vektor-Multiplikationen Techniken eingesetzt werden, welche die Matrixbereiche mit verschwindenden Koeffizienten überspringen. Eine solche Technik wurde auch in dem Programm angewendet, welches der Autor von [3] für seine Untersuchungen geschrieben hat.

Eine weitere Möglichkeit zur Beschleunigung von Berechnungen mit Matrizen besteht darin, diese zu *blocken*, d. h. in kleinere Blockmatrizen aufzuteilen. Dadurch befinden sich Matrixelemente häufiger im L1-Cache. Details zum Verfahren geblockter Matrizen können [22] entnommen werden.

Der Nutzen dieses Verfahrens für die hier vorliegenden dünn besetzten Matrizen wurde jedoch nicht näher untersucht. Es ist ebenso unklar, ob es sinnvoll ist, die Matrizen auf Folgen gleicher Faktoren zu untersuchen. Die grundlegende Idee zu diesen Folgen ist in Abbildung 6.2 zu sehen. Die Folgen könnten einmal gespeichert werden und müssten nur noch referenziert werden, wie es in ähnlicher Weise in Abschnitt 5.3.2 für Listenbasierte Abbildungen erwähnt wurde. Zumindest könnten die Matrizen dadurch noch kompakter gespeichert werden und wären demnach auch schneller aus dem RAM zu laden.

## 6.3 Listen-basierte Berechnungen

Die im vorigen Abschnitt erwähnten Inhomogenitäten der Abbildungseigenschaften treten bei  $O^2$ -Listen erfahrungsgemäß noch stärker zu Tage. Um nun Ladeoperationen aus dem RAM soweit wie möglich zu reduzieren, ist die Frage zu klären, welche Art von Daten eine höhere „Wiederverwendbarkeit“ hat: Sind es die Urbildkoeffizienten, die für mehrere

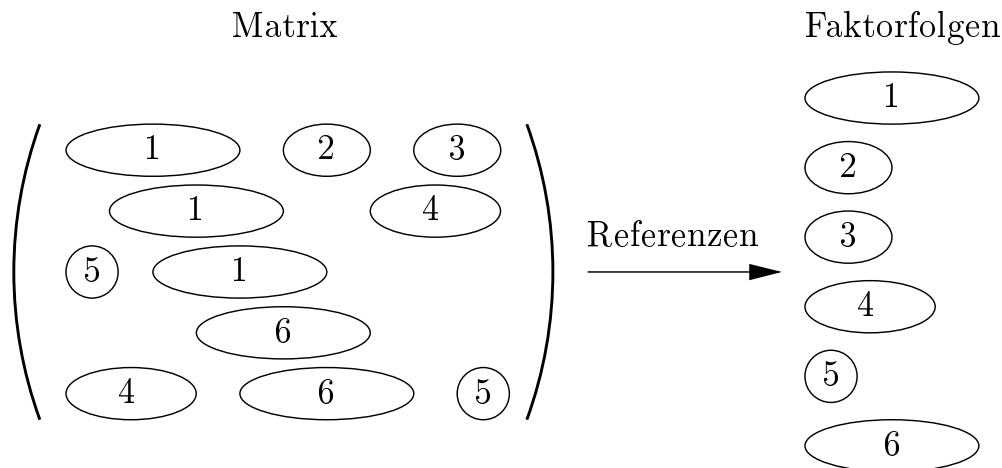


Abbildung 6.2: Matrix mit (Zeilen-)Folgen von endlichen Faktoren (gekennzeichnet durch Ellipsen). Gleiche Folgen können identifiziert und referenziert (nummeriert) werden. Nicht nummerierte Bereiche kennzeichnen Matrixelemente mit dem Wert Null.

Bildkoeffizienten einen Anteil liefern? Oder kommt es häufiger vor, dass gleiche Abbildungseigenschaften, d. h. gleiche  $O^2$ -Listen, für verschiedene Bildkoeffizienten verwendet werden können?

Betrachten wir der Einfachheit halber eine FDTD zur Simulation eines homogenen Mediums. Das Diskretisierungsgitter soll äquidistant sein. In diesem Fall sind die Vorfaktoren vor den jeweiligen Urbildkoeffizienten über den gesamten Raum (bis auf das Vorzeichen) konstant. Selbst bei praxisnäheren Simulationsstrukturen ist die Anzahl der auftretenden Faktoren gering. Auch wenn sich diese Situation nach einer WT der FDTD nicht mehr so günstig darstellt, können trotzdem sich wiederholende Abbildungseigenschaften erwartet werden.

Einige Versuche erbrachten, dass teilweise mehrere Bildkoeffizienten von den gleichen Urbildkoeffizienten abhängen können. Noch häufiger kam es vor, dass verschiedene Bildpunkte trotz unterschiedlicher Verschiebungsvektoren in den  $O^2$ -Listen und trotz unterschiedlicher (absoluter) Positionen von Urbildkoeffizienten mit gleichen Faktoren zu berechnen waren. Daher wurde ein Verfahren entwickelt, welches verschiedene Arten mehrfach verwendbarer Daten berücksichtigt. Zuvor sollen noch drei Begriffe eingeführt werden:

**$O^2$ -Strecke:**

Eine zusammenhängende (eindimensionale) Reihe von Bildpunkten, welche die gleiche  $O^2$ -Liste referenzieren. Sie ist gekennzeichnet durch die  $O^2$ -Liste, die Anfangskoordinate der Reihe, die Länge und die Richtung (parallel zu einer der Achsen) der Reihe. Entlang dieser Strecke ist die Abbildungseigenschaft also konstant.

**Urbild-Koordinatenfolge:**

Eine Folge aus Koordinaten von Urbildkoeffizienten. Diese *absoluten* Koordinaten geben an, aus welchen Urbildanteilen sich ein Bildkoeffizient zusammensetzen soll. Zum Vergleich: In  $O^2$ -Listen sind *relative* Koordinaten angegeben.

 **$O^2$ -Faktorfolge:**

Alle Faktoren einer  $O^2$ -Liste werden zu einer Folge zusammengefasst.

Die Abbildungsdaten eines Operators werden nun folgendermaßen gruppiert:

1. Sammeln von  $O^2$ -Strecken mit einer Länge von mindestens Zwei.
2. Zusammenfassen der  $O^2$ -Strecken nach gleichen  $O^2$ -Listen und gleichen Richtungen.
3. Sammeln von Urbild-Koordinatenfolgen in den Bildbereichen, wo keine  $O^2$ -Strecken detektiert wurden.
4. Erstellen einer Liste mit  $O^2$ -Faktorfolgen. Berücksichtigt werden dabei nur diejenigen  $O^2$ -Listen, die von einem der Bildpunkte nach Punkt 3 benötigt werden.
5. Zusammenfassen aller Bildpunkte, welche die gleiche Urbild-Koordinatenfolge haben. Für jeden der Bildpunkte wird die zugehörige  $O^2$ -Faktorfolge referenziert.

Die Berechnung gliedert sich in zwei Teile, die unabhängig voneinander ausgeführt werden können:

- Berechnung mit  $O^2$ -Strecken:
  1.  $O^2$ -Liste der Strecke aus dem RAM laden.

2. Anfangskoordinate (erster Bildpunkt) der  $O^2$ -Strecke holen.
  3. Berechnen des selektierten Bildkoeffizienten entsprechend der  $O^2$ -Liste.
  4. Auswählen der nächsten Bildkoordinate der Strecke. Weiter nach Punkt 3.
- Berechnung mit Urbilddaten:
    1. Urbild-Koordinatenfolge aus dem RAM laden.
    2. Bildkoordinate und zugehörige Referenz auf  $O^2$ -Faktorfolge laden.
    3. Berechnen des selektierten Bildkoeffizienten aus Urbildkoeffizienten und Faktorfolge.
    4. Nächsten Bildkoeffizienten mit der gleichen Urbild-Koordinatenfolge selektieren. Weiter nach Punkt 2.

Mit diesen Angaben können Operatoren zur Berechnung eines Bildsignals aus einem Urbildsignal benutzt werden. Das Programm, welches für die vorliegende Arbeit entwickelt wurde, geht allerdings noch einen Schritt weiter: Die in diesem Abschnitt genannten Strukturierungen der Operatordaten werden benutzt, um daraus ein Operator-spezifisches Programm zu generieren. Konkret bedeutet das, dass aus den ermittelten Daten und Strukturen ein Programm in der Sprache *C* erzeugt wird, welches danach kompiliert wird. Der Vorteil bei dieser Vorgehensweise ist, dass der Compiler weitere Optimierungen speziell für diesen Operator auf einer Prozessor-spezifischen Ebene durchführen kann. Das eigentliche Simulationsverfahren wird von dieser Art der Geschwindigkeitsoptimierung jedoch nicht beeinflusst.



# Kapitel 7

## Die FDTD im Waveletbereich

### 7.1 Vorbetrachtungen

Bisher wurde bei Wavelet-Transformationen stets davon ausgegangen, dass eine Funktion  $f \in \mathbf{V}_0$  vorliegt, damit diese dann in die Räume der höheren Skalen projiziert werden kann. Analog führt eine Rücktransformation wieder auf eine Funktion im Grundraum  $\mathbf{V}_0$ . Nach Abschnitt 3.2.2 und Kapitel 4 ist  $\mathbf{V}_0$  immer aus Skalierungsfunktionen zusammengesetzt. Streng genommen muss also erst einmal eine Funktion  $f$  auf diesen Raum projiziert werden, damit eine WT überhaupt stattfinden kann. Es werden in der Literatur nur wenige Fälle behandelt, die ohne eine solche Projektion auskommen. Mit einer erweiterten Betrachtungsweise lassen sich die bestehenden Grenzen ausdehnen. Dazu sind die folgenden Aspekte zu berücksichtigen:

1. Entwicklung in  $\mathbf{V}_0$ :  
Der Trivialfall: Alle Felder und Operatoren sind Element des  $\mathbf{V}_0$  bzw. wurden in  $\mathbf{V}_0$  entwickelt.
2. Bijektivität der diskreten WT:  
Die WT-Operatoren  $\mathcal{P}$  und  $\mathcal{P}^*$  sind nach dem bisher Gesagten bijektiv zueinander, d. h. es gilt  $\mathcal{P}\mathcal{P}^* = \text{id}$ . Genau das gleiche gilt auch für die diskreten Abbildungen  $P$  und  $P^*$ . Sind Berechnungen (Simulationen) im Computer durchzuführen, liegen sämtliche Operatoren

und Felder nur noch in diskreter Form vor. Werden diese mit  $P$  und  $P^*$  transformiert, besteht eine Isomorphie zwischen den transformierten und den untransformierten Größen. Die Transformation in den Waveletbereich bewirkt also nur eine andere, aber dennoch äquivalente Darstellung der Entwicklungskoeffizienten. Dass mit der WT bzw. den zugehörigen Skalierungs- und Waveletkoeffizienten auch Basisfunktionen verknüpft sind, ist in diesem Zusammenhang nicht von Belang.

Zur Verdeutlichung sollen die beiden folgenden Gleichungen betrachtet werden:

$$g = \mathcal{T}f \xrightarrow{\text{diskretisieren}} G = TF \xrightarrow{\text{WT } (P, P^*)} \tilde{G} = \tilde{T}\tilde{F}, \quad (7.1)$$

$$g = \mathcal{T}f \xrightarrow{\text{WT } (P, P^*)} \tilde{g} = \tilde{T}\tilde{f} \xrightarrow{\text{diskretisieren}} \tilde{G} = \tilde{T}\tilde{F}. \quad (7.2)$$

In beiden Fällen werden Simulationen mit den transformierten Größen durchgeführt. Gleichung (7.1) entspricht dabei der o. g. Vorgehensweise. Bei einer Rücktransformation von  $\tilde{G}$  nach  $G$  sind also wieder die Entwicklungsfunktionen zu nehmen, die zum gleichen Raum gehören wie  $g$ . In Gleichung (7.2) dagegen entspricht die WT gleichzeitig der Projektion auf Skalierungs- und Waveleträume. Nach der Rücktransformation von  $\tilde{G}$  nach  $G$  liegt demnach eine Entwicklung in  $\mathbf{V}_0$  vor. Dadurch tritt ein Wechsel des Funktionsraumes ein, sofern der Operator und die Funktionen nicht bereits in  $\mathbf{V}_0$  entwickelt waren (vgl. Punkt 1).

### 3. Die WT als Kompressionsverfahren:

Bei den bisherigen mathematischen Betrachtungen ist das eigentliche Ziel der WT von Operatoren und Signalen, nämlich die Reduktion des Berechnungsaufwands durch Kompression der Daten, etwas in den Hintergrund geraten. Letztlich führt die Kompression zu einer Approximation, welche die Realität einfach so gut wie möglich bzw. nötig beschreiben soll. Unter diesem Aspekt gesehen geht es um eine Verringerung der Anzahl von Entwicklungskoeffizienten. Unabhängig von den Basisfunktionen, die zu diesen Koeffizienten gehören, muss das  $n$ -dimensionale Wertemuster der Koeffizienten eine Verteilung aufweisen, mit der eine Kompression mittels einer WT auch erreichbar ist. Es ist denkbar, dass es besser sein kann, ein Verfahren nicht in  $\mathbf{V}_0$  zu entwickeln, weil das Verfahren mit anderen

Funktionenräumen (siehe Punkt 2) ein günstigeres Verhältnis von numerischem Aufwand und Komprimierbarkeit der Entwicklungskoeffizienten aufweist.

#### 4. Interpolierende Funktionen:

Sehr oft liegen Funktionen nicht analytisch vor, sondern in Form von *Abtastwerten*, d. h. als eine Folge von Koeffizienten  $(s_k)_{k \in \mathbb{Z}}$ . In [23] wird das Problem behandelt, die Abtastwerte mit Hilfe der Skalierungsfunktionen aus dem Grundraum  $\mathbf{V}_0$  zu interpolieren.

Nach Gleichung (3.32) ist eine Funktion  $f \in \mathbf{V}_0$  durch eine Entwicklung

$$f(x) = \sum_{k \in \mathbb{Z}} c_k^0 \varphi_{0,k}(x) \quad (7.3)$$

darstellbar. Somit lautet die Frage, welcher Zusammenhang zwischen  $(s_k)_{k \in \mathbb{Z}}$  und  $(c_k^0)_{k \in \mathbb{Z}}$  besteht. Dazu wird zunächst die folgende Definition benötigt:

**Definition 10.** Eine Skalierungsfunktion  $\varphi(x)$  ist *interpolierend*, wenn gilt

$$\varphi(k) = \delta_{0,k} , \quad k \in \mathbb{Z} . \quad (7.4)$$

In diesem Fall besteht die einfache Beziehung

$$c_k^0 = s_k . \quad (7.5)$$

Eine leichte Verallgemeinerung lässt sich mit der *verschobenen Interpolation* erreichen, wo ein Verschiebungswert  $\tau \in [0, 1[$  existieren muss, sodass  $\varphi(\tau + k) = \delta_{0,k}$  ist.

In dieser Arbeit wurden nur Daubechies-Skalierungsfunktionen benutzt, für die keines der beiden Kriterien exakt zutrifft. Es ist in [23] jedoch nachzulesen, dass diese Funktionen das Kriterium der verschobenen Interpolation *beinahe* erfüllen. Exemplarisch gilt für die  $D_2$ -Skalierungsfunktion ( $\varphi_N$  mit  $N = 2$ , vgl. Abbildung 3.1)

$$\begin{aligned} \tau &= \frac{1 + \sqrt{3}}{4} \approx 0,6830127 , \\ \varphi(\tau) &\approx 1,0002086 , \\ \varphi(\tau + 1) &\approx -4,1718154 \cdot 10^{-4} , \\ \varphi(\tau + 2) &\approx 2,0859077 \cdot 10^{-4} . \end{aligned} \quad (7.6)$$

Werden Skalierungskoeffizienten in umgekehrter Reihenfolge zu Gleichung (3.41) benutzt, d. h. im Falle der  $D_2$ -Skalierungsfunktion  $h'_k = h_{3-k}$ , ändert sich  $\tau$  entsprechend Abschnitt A.1 zu  $\tau' = 3 - \tau \approx 2,3169873$ .

Die FDTD ist in dieser Arbeit der einzige konkrete Anwendungsfall für die Transformation eines Simulationsverfahrens in den Waveletbereich. Hierfür treffen die Punkte 2 bis 4 zu. Insbesondere die fast-interpolierende Eigenschaft der verwendeten Daubechies-Skalierungsfunktionen sowie die stückweise Stetigkeit elektromagnetischer Felder garantieren eine direkte Benutzbarkeit der Feldkoeffizienten, ohne auf einen anderen Funktionenraum wechseln zu müssen. In diesem Fall spielt es auch keine Rolle, dass für das diskretisierte elektrische Feld eigentlich andere Basisfunktionen gelten als für das magnetische Feld (es existiert ein Versatz um eine halbe Diskretisierungsweite).

Die Verwendung unterschiedlicher Funktionenräume erfordert dennoch im Allgemeinen eine besondere Sorgsamkeit. In [3] und [24] beispielsweise werden mit Hilfe der Momentenmethode die Entwicklungskoeffizienten der Felder und die Ableitungsoperatoren der Maxwell'schen Gleichungen unter anderem auch im Grundraum  $V_0$  der  $D_2$ -Skalierungsfunktionen entwickelt. Die Ableitungsoperatoren für absorbierende Randbedingungen jedoch werden grundsätzlich über zentrale finite Differenzen approximiert, wobei die Feldkoeffizienten in diesem Gebiet Rechteckfunktionen als Basisfunktionen haben. Hier stellt sich die Frage, wie der eigentliche Feldraum und der ABC-Raum miteinander zu verknüpfen sind. Dabei kommt einem die (approximierte) Eigenschaft der verschobenen Interpolation der Daubechies-Skalierungsfunktionen zugute, die eine Art *Lokalisierung* erlaubt, d. h. eine einfache Beziehung zwischen einem diskreten Ortsvektor  $(i, j, k)^T$  und einer Position  $(x, y, z)^T$  im  $\mathbb{R}^3$ . Hierbei ist auch die Reihenfolge der Skalierungskoeffizienten unbedingt zu beachten, siehe Punkt 4.

Finite-Differenzen-Verfahren gemischter Ordnung stellen im Gegensatz dazu kein Problem dar. Wird die Ordnung eines FD-Verfahrens angegeben, so bezieht sich das auf die Approximation des Differenzialoperators. Je höher die Ordnung, desto mehr Entwicklungskoeffizienten berücksichtigt der FD-Operator. Es ist jedoch i. A. schwierig, bei der Simulation eines Feldgebietes *ausschließlich* einen FD-Operator vierter oder höherer

Ordnung zu verwenden. So ist es nicht unüblich, in einem FD-Verfahren vierter Ordnung an „problematischen Stellen“ auf einen einfacheren FD-Operator zweiter Ordnung auszuweichen. Das entspricht allerdings nicht einem Wechsel des Funktionensystems, da beide FD-Operatoren auf gleichermaßen diskretisierte Feldwerte zugreifen.

Als Fazit bleibt zu sagen, dass es sich – wie in Abschnitt 5.1.1 bereits gesagt – empfiehlt, ein Verfahren zu diskretisieren und sich danach auf die Entwicklungskoeffizienten zu beschränken. Von da an spielt es keine Rolle mehr, ob das Verfahren in  $\mathbf{V}_0$  entwickelt wurde oder nicht. Die Diskretisierung sollte jedoch entsprechend Punkt 3 so durchgeführt werden, dass eine effiziente Kompression der Daten möglich ist.

## 7.2 Matrixoperatoren der FDTD in der WT

In [3] werden Operatormatrizen für eindimensionale Differenzierungen, Randbedingungen und Materialverteilungen behandelt. Erweiterungen zu absorbierenden Randbedingungen sind in [24] und [25] zu finden. Zu den in diesen und weiteren Literaturquellen vorgestellten Verfahren und Operatoren sollen im Folgenden einige Besonderheiten herausgestellt werden. Ferner wird mit Hilfe der Erkenntnisse aus Abschnitt 5.2 ausgearbeitet, welche algorithmischen Konsequenzen sich aus der Matrix-Darstellung ergeben.

### 7.2.1 Differenzierungen

In der klassischen FDTD wird der Rotationsoperator aufgeteilt in partielle Differenzierungen in x-, y- und z-Richtung. Jeder der Operatoren  $\partial_\nu$ ,  $\nu \in \{x, y, z\}$ , ist bezogen auf den zu simulierenden Feldraum nach der Diskretisierung vom Typ  $U_0'$ . Die in Abschnitt 3.2.5 vorgestellte DPWT erfordert in jeder zu transformierenden Richtung eine Anzahl von Entwicklungskoeffizienten, die genau einer Zweierpotenz entspricht. Daher ist i. d. R. der Simulationsraum durch das Anhängen von zusätzlichem Raum soweit zu vergrößern, dass das genannte Kriterium für eine DPWT erfüllt wird, siehe Abbildung 7.1. Nach Abschnitt 5.2 müssen sämtliche

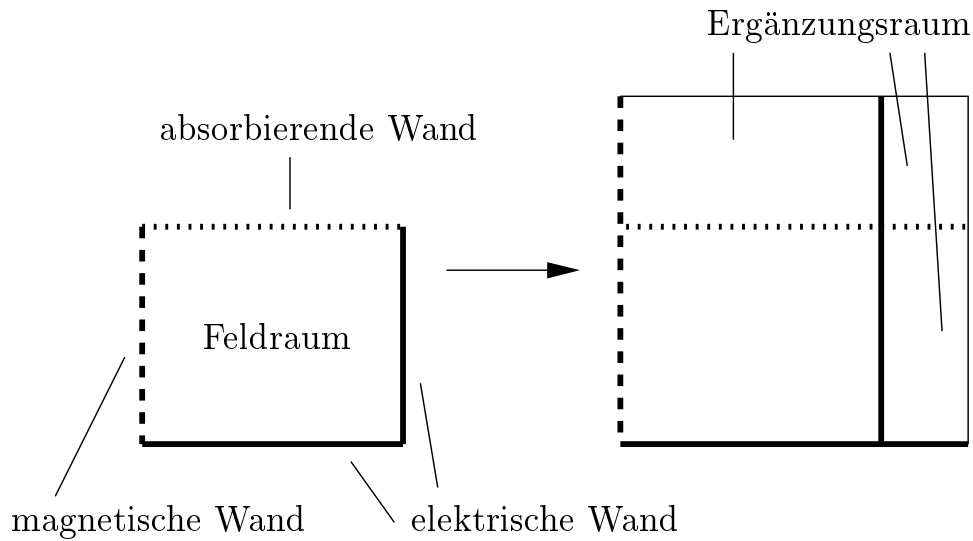


Abbildung 7.1: Erweiterung eines diskretisierten zweidimensionalen Simulationsgebietes für eine DPWT (Zweierpotenz-Längen in jeder Richtung). Äußere Randbedingungen und Differenziationsoperatoren werden homogen fortgesetzt.

Operatoren in Teiloperatoren vom Typ  $U_0^\nu$  zerlegt bzw. auf solche erweitert werden. Da das ebenso für die  $\partial_\nu$ -Operatoren gilt, steigt demzufolge durch den zusätzlichen Raum auch der Rechenaufwand.

## 7.2.2 Randbedingungen

Äußere Berandungen eines Simulationsgebietes sind meist homogen entlang einer Fläche, d. h. es liegen beispielsweise elektrische und magnetische Wände nicht direkt nebeneinander. Diese können daher bei der im vorherigen Abschnitt erwähnten Erweiterung des Raumes ebenfalls homogen fortgesetzt werden und sind somit vom Typ  $U_0^\nu$ .

In [3] werden äußere elektrische und magnetische Wände dadurch realisiert, dass die Differenziationsoperatoren angepasst werden. Nehmen wir als Beispiel eine bei  $z = 0$  bzw. im diskretisierten Raum  $k = 0$  liegende, zur  $xy$ -Ebene parallele elektrische Wand an. Zur Berechnung von  $H_x$  ist unter anderem  $E_y$  nach  $z$  zu differenzieren bzw. eine Differenz entlang  $k$  zu bilden (siehe Gleichung (2.17)). Der diskretisierte  $\partial_z$ -Operator ist nun so zu modifizieren, dass die zu  $k = 0$  gehörende Spalte seiner Operatormatrix nur Elemente mit dem Wert Null enthält. Damit wird das

Verschwinden der tangentialen elektrischen Feldkomponente bei  $k = 0$  in der Matrix hinterlegt. Als Schlussfolgerung ergibt sich, warum diese Vorgehensweise allenfalls für Außenwände praktikierbar ist: Nur diese sind wie die Differenziationsoperatoren vom Typ  $U'_0$ , d. h. homogen in der zugehörigen Querschnittsfläche.

Innerhalb des Simulationsgebietes liegende ideal leitende Wände werden dadurch nachgebildet, dass an den entsprechenden Stellen die Permittivität auf einen sehr hohen Wert gesetzt wird. Betrachten wir wieder eine zur  $xy$ -Ebene parallele Wand, so muss der Permittivitätstensor bei den Raumkoordinaten der Wand  $\epsilon = \text{diag}(a, a, \epsilon_z)$ ,  $a \rightarrow \infty$  lauten. Die Erklärung hierfür ist in Gleichung (2.20) zu finden: Durch die sehr hohe Permittivität wird der diskrete Rotationsoperator quasi mit Null multipliziert. Dadurch bleibt der Wert der  $E_x$ -Komponente auf Null, wenn dieser zu Beginn der Berechnungen ebenfalls Null war. Eine Modifikation der Berechnung des magnetischen Feldes wie im Fall der Außenwände erübrigt sich somit, da die zur Wand tangentiale elektrische Feldstärke auf dem Wert Null gehalten wird. Dennoch fällt für  $E_x$  Rechenaufwand an, da der Rotationsoperator in Gleichung (2.20) nur künstlich unwirksam gemacht wird.

Abbildung 7.1 zeigt, dass absorbierende Randbedingungen ebenfalls zu den äußeren Berandungen gehören. Während in [3] die einfache ABC von Mur implementiert wurde, konnte in [24] eine Matrix-basierte Formulierung der PML für eine zweidimensionale FDTD realisiert werden. Ein Problem bei einer solchen Umsetzung besteht allerdings darin, dass die Feldkomponenten im Gebiet der ABC in jeweils zwei Anteile separiert werden müssen, siehe Gleichung (2.28). Damit das ganze Simulationsgebiet einer WT unterzogen werden kann, wurden durch Anwendung von Gleichung (2.29) im kompletten Simulationsraum die Feldkomponenten separiert. Damit verdoppelt sich im Bereich des eigentlichen Feldraumes der numerische Aufwand.

Eine neuere Variante der PML mit der Abkürzung A-UPML (Anisotropic Uniaxial Perfectly Matched Layer) wurde in [25] als Matrixoperator formuliert. Dabei wurden erstmalig sowohl im Feldraum als auch in der ABC die Ableitungsoperatoren im Grundraum  $\mathbf{V}_0$  entwickelt. Die A-UPML erfordert genau wie die PML zusätzliche Feldkomponenten in ihrem Bereich (diskretisierte  $\mathbf{B}$ - und  $\mathbf{D}$ -Felder). Auch in diesem Fall wurden die neuen Feldkomponenten über den kompletten Simulationsraum ausgedehnt. Da-

durch erhöht sich ebenfalls der Rechenaufwand, allerdings weniger als im oben genannten Fall mit der PML. Bei beiden Varianten verdoppelt sich aber gleichermaßen der Speicherbedarf.

### 7.2.3 Materialparameter

Materialkenngrößen und Diskretisierungszeitschritt sind Bestandteil der Skalierungsoperatoren<sup>1</sup> der FDTD-Gleichungen. In den beiden Gleichungen (2.17) und (2.20) sind die Tensorelemente dreier Skalierungsoperatoren für eine Koordinate zu finden:

$$a_1 = -\frac{\Delta t}{\mu_x(i, j, k)}, \quad (7.7)$$

$$a_2 = \frac{2\varepsilon_x(i, j, k) - \kappa_x(i, j, k)\Delta t}{2\varepsilon_x(i, j, k) + \kappa_x(i, j, k)\Delta t}, \quad (7.8)$$

$$a_3 = \frac{2\Delta t}{2\varepsilon_x(i, j, k) + \kappa_x(i, j, k)\Delta t}. \quad (7.9)$$

In [3] werden die zugehörigen Operatoren *Materialoperatoren* genannt. Nach dem bisher Gesagten sind sie ebenfalls in  $U_0'$ -Operatoren zu zerlegen. Eine manuelle Zerlegung birgt oftmals auch bei scheinbar einfachen Konfigurationen ein hohes Fehlerpotenzial. Wegen der Verkopplung von Permittivitäten und Leitfähigkeiten in den Faktoren  $a_2$  und  $a_3$  genügt es nämlich nicht, getrennt nach Bereichen mit konstantem  $\varepsilon$  und konstantem  $\kappa$  Ausschau zu halten<sup>2</sup> – es sind immer beide Materialeigenschaften *gemeinsam* zu betrachten. Hierbei ist auch die Darstellung ideal leitfähiger Wände durch sehr hohe (anisotrope) Permittivitäten zu berücksichtigen.

Vor allem elektrische Wände können aus diesem Grund eine Zerlegung äußerst kompliziert werden lassen. Dieser Fall tritt hauptsächlich dann ein, wenn unebene (gewölbte) Wände, oder solche, die schräg zu den Koordinatenachsen liegen, im Simulationsgebiet auftreten. Dann nämlich müssen diese bei genügend feiner Diskretisierung durch sehr viele kleine Wandstücke approximiert werden, welche parallel zu den Achsen liegen. Entsprechend inhomogen fällt dann die Permittivitätsverteilung aus.

<sup>1</sup>Siehe Definition 3 auf Seite 53.

<sup>2</sup>Genau genommen ist bei Skalierungsoperatoren eine Ortsabhängigkeit der Tensorelemente in höchstens einer Richtung zulässig.

## 7.2.4 Feldanregung und -ausgabe

Mit Beginn einer Simulation ist dem Feldraum als Startvorgabe bzw. Anfangsbedingung eine Feldverteilung aufzuprägen, was auch als *Anregung* bezeichnet wird. Bei einem Wavelet-transformierten Verfahren bedeutet das, dass als Startwert die Wavelet-transformierte Feldverteilung zu benutzen ist. Nach dem bisher Gesagten ist klar, dass hierbei das Feld des kompletten Simulationsraumes einer WT zu unterziehen ist, auch wenn die Anregung eigentlich nur in einem kleinen Teilbereich stattfinden soll.

Noch häufiger als eine Anregung zum Startzeitpunkt  $t = 0$  kommt es vor, dass das Anregungsfeld zeitabhängig ist und die Anregung daher über mehrere Zeitschritte verläuft. Dann ist bei jedem dieser Zeitschritte die anregende Feldverteilung einer WT zu unterziehen. Abschnitt 5.2.1 und Kapitel 4 geben Auskunft über den Aufwand, der dann jedesmal anfällt. Im Falle homogener Tensorprodukt-Basen ist für jede Richtung eine Matrix  $\mathbf{P}_{0,j_\nu}^\nu$  erforderlich. Numerisch effizienter ist jedoch die stufenweise (rekursive) Transformation nach Gleichung (3.55) bzw. (3.63). Die Vorgehensweise ist dann sehr ähnlich zu der WT bei separablen MSA-Basen.<sup>3</sup> Nehmen wir als Beispiel zwei Transformationsstufen für alle drei Richtungen, dann ergibt das insgesamt sechs WT-Operatoren, die jeweils auf den zugehörigen Feldbereich anzuwenden sind. Der Aufwand ist schließlich noch mit der Anzahl der anzuregenden Feldkomponenten zu multiplizieren.

Anregungen finden meistens nur in einem relativ kurzen Zeitabschnitt statt, sodass der beschriebene Aufwand nicht so sehr ins Gewicht fällt. Das gilt allerdings nicht für *Ausgaben*, auf die eine analoge Vorgehensweise anzuwenden ist. Die Auswertung von Simulationsergebnissen erfordert häufig eine Frequenzanalyse einzelner Feldbereiche. Soll dies mit einer FFT (*Fast Fourier Transform*) geschehen, müssen für eine genügende Frequenzschärfe mehrere tausend Zeitpunkte berücksichtigt werden. Dazu ist nach jedem Simulationszeitschritt das komplette Feld einer inversen WT zu unterziehen, um dann gezielt auf die Feldwerte an den relevanten

---

<sup>3</sup>Ein Unterschied besteht darin, dass nur bei homogenen Tensorprodukt-Basen für alle Richtungen eine separate Transformationstiefe gewählt werden kann. Bei separablen MSA-Basen dagegen reduziert sich der zu transformierende Urbildbereich nach jeder Skala stärker als bei homogenen Tensorprodukt-Basen.

Stellen zugreifen zu können.<sup>4</sup>

Feldwerte an einzelnen Punkten im Raum erfordern also sowohl bei der Anregung als auch bei der Ausgabe die Transformation des kompletten Feldes. Wavelets erlauben zwar eine gewisse Art von Lokalisierung, doch diese ist nicht ohne Weiteres auf Matrix-basierte mehrdimensionale Wavelet-Transformationen übertragbar.

## 7.2.5 Adaptive Berechnungen

Es wurden in [3] und [26] verschiedene Verfahren vorgeschlagen, wie mit Hilfe der WT Simulationszeiten reduziert werden können. Die dort gezeigten Methoden kombinieren die Wavelet-transformierte FDTD mit einer räumlich oder zeitlich adaptiven Rechenvorschrift.

### 7.2.5.1 Räumliche Adaptivität

Wavelet-transformierte Signale lassen sich dadurch komprimieren, dass diejenigen Entwicklungskoeffizienten des Wavelettraumes vernachlässigt werden, deren Beträge sich unterhalb eines Schwellenwertes befinden. Um sich nun die komprimierenden Eigenschaften der WT in einer Zeitbereichssimulation zunutze zu machen, ist es erforderlich, bereits vor der Berechnung des Feldes für den nächsten Zeitschritt zu wissen, welche Koeffizienten vernachlässigbar sein werden und welche nicht. Dazu wurde in [3] ein Verfahren entwickelt, welches kurz beschrieben werden soll.

Das Verfahren der räumlichen Adaptivität beginnt damit, dass ein zu einem bestimmten Zeitpunkt existierendes Feld komprimiert wird. Die Stellen der verbliebenen Koeffizienten werden gespeichert. Für die nächsten  $N$  Zeitschritte werden dann ausschließlich diese und gewisse „benachbarte“ Koeffizienten neu berechnet. Danach ist die Verteilung der vernachlässigbaren Koeffizienten zu aktualisieren.

Der Grundgedanke bei diesem Verfahren ist, dass sich elektromagnetische Wellen nur mit einer berechenbaren maximalen Geschwindigkeit ausbreiten können. Damit kann sich ein nicht-vernachlässigbarer Feldwert innerhalb der  $N$  Zeitschritte nur in einem begrenzten Gebiet ausbreiten. Das

---

<sup>4</sup>Unter Umständen genügt auch die Ausgabe nach beispielsweise jedem vierten Zeitschritt.

gilt ebenfalls im Waveletbereich, wobei das Gebiet aber auf verschiedene Skalen verteilt ist.

Die beschriebene Kompression wird genau genommen nicht im gesamten Koeffizientenraum vorgenommen: Der Skalierungsraum der höchsten Skala,  $V_j$ , wird grundsätzlich vollständig berechnet. Dies wird in [3] als eine Art *Grunddiskretisierung* beschrieben.

Verschiedene Aspekte dieses Verfahrens senken die Beschleunigung der Simulationszeit auf einen Wert, der unterhalb dessen liegt, was man anhand der Anzahl der nicht-vernachlässigbaren Koeffizienten erwartet. So sind schließlich nicht nur diese, sondern auch die erwähnten benachbarten Koeffizienten zu berechnen. Etwas Rechenzeit ist natürlich auch für die Aktualisierung der Verteilung der nicht-vernachlässigbaren Koeffizienten aufzuwenden.

Weiterhin kann Zeit für die Berechnung des Schwellenwertes anfallen. In [3] wird hierzu eine Art Energieinhalt des Feldes berechnet und das Ergebnis mit einem kleinen Wert von z. B. 0,01 multipliziert. Dieser (einheitenlose) Wert dient dann als Maß für die Vernachlässigung von Koeffizienten. In [26] wird dagegen erwähnt, dass statt eines zeitabhängigen Schwellenwertes auch ein fester Wert vorgegeben werden kann. Es stellt sich aber die Frage, welcher Wert für eine anstehende Simulation sinnvoll ist. Die Verwendung und Berechnung von Schwellenwerten lässt sich sicherlich noch weiter optimieren, insbesondere wenn die Eigenschaften elektromagnetischer Wellen bzw. die des FDTD-Verfahrens berücksichtigt werden, z. B. Energieerhaltung.

Die Verwendung der Informationen aus der Komprimierung wird nach diesem Verfahren eigentlich nur zur Hälfte genutzt: Die Selektion findet lediglich bei den zu berechnenden Bildkoeffizienten statt, nicht jedoch bei den Urbildkoeffizienten, die für die Berechnung herangezogen werden müssen. Es ist jedoch zweifelhaft, ob sich diese Effizienz bei einem auf Matrix-Vektor-Multiplikationen basierenden Verfahren steigern lässt.<sup>5</sup> Erst wenn ein Ansatz hierzu existiert, lässt sich auch der in Abschnitt 7.2.4 erwähnte Aufwand für Feldausgaben reduzieren.

---

<sup>5</sup>Zur Klärung dieser Frage wären im Wesentlichen zwei Punkte zu berücksichtigen: Die Besetzung der zu verwendenden Matrizen (dünn / voll besetzt) sowie der Zeitaufwand für die Überprüfung, ob ein Urbildkoeffizient benutzt werden soll.

Abschließend soll noch die Frage der Stabilität der räumlichen Adaptivität erörtert werden. Ohne die Vernachlässigung von Koeffizienten existieren mit den WT-Operatoren bijektive Abbildungen zwischen der untransformierten und der transformierten FDTD. Das bedeutet, dass die Stabilität der FDTD nach der WT erhalten bleibt. Die Vernachlässigung von Koeffizienten entspricht jedoch einer Veränderung der WT-Operatoren und hebt damit deren Bijektivität auf. Demnach kann eine Änderung des Stabilitätskriteriums aus Gleichung (2.26) erwartet werden. Eine Herleitung eines neuen Kriteriums wurde in [3] nicht unternommen, jedoch wurde empirisch festgestellt, dass die Einhaltung des Stabilitätskriteriums der FDTD genügt, um stabile Simulationen mit dem Verfahren der räumlichen Adaptivität durchzuführen. Abschnitt 7.3.6.3.3 wird sich dieses Themas nochmals annehmen.

### 7.2.5.2 Zeitliche Adaptivität

Als ein möglicher Ansatz zur Verbesserung der Effizienz bei der räumlichen Adaptivität wurde in [26] die Idee der *Scale Adaptive Time Steps* (SATS) entwickelt. In das Stabilitätskriterium der FDTD gehen die Ausbreitungsgeschwindigkeit elektromagnetischer Wellen und die räumliche Diskretisierung ein, siehe Gleichung (2.26). Je gröber die Auflösung ist, desto größer ist die maximal zulässige Zeitschrittweite. Dieser Zusammenhang wird in [26] auf die mit einer WT verbundenen Zerlegung eines Signals in verschiedene Auflösungsstufen bzw. Skalen ausgedehnt.

Abbildung 4.1 auf Seite 42 zeigt für homogene Tensorprodukt-Basen, welche Funktionenräume nach einer dreistufigen WT im zweidimensionalen Raum auftreten. Innerhalb eines solchen Funktionenraumes haben die Basisfunktionen gleiche Auflösungseigenschaften. In [26] ist nun eine Rechenvorschrift angegeben, die beispielsweise dem Raum  $\mathbf{W}_3\mathbf{W}_2$  einen Faktor  $V_{3,2}$  zuweist. Dieser Faktor gibt an, der wievielfache Zeitschritt in diesem Raum erlaubt ist. Damit sind Koeffizienten von  $\mathbf{W}_3\mathbf{W}_2$  nur alle  $V_{3,2}$  Zeitschritte zu berechnen.

Die Räume der niedrigen Skalen sind wegen ihrer hohen Auflösungsstufe am häufigsten zu berechnen. Für  $\mathbf{W}_1\mathbf{W}_1$  z. B. ist  $V_{1,1} = 1$ , d. h. Koeffizienten dieses Raumes sind in jedem Zeitschritt zu berechnen. Diese Räume enthalten zwar auch die meisten Koeffizienten, sodass über alle Räume betrachtet nur verhältnismäßig wenig Koeffizienten von einer

gröberen zeitlichen Auflösung profitieren können. Im Zusammenspiel mit der räumlichen Adaptivität ändern sich die Verhältnisse allerdings, da sich bei einer genügend hohen Kompression die nicht-vernachlässigbaren Koeffizienten vorwiegend in den Räumen der hohen Skalen konzentrieren. Damit sollten im Endeffekt die meisten Koeffizienten weniger oft berechnet werden müssen als bei ausschließlicher Anwendung der räumlichen Adaptivität.

Erste Ergebnisse aus Simulationsversuchen mit SATS sind in [27] zu finden. Als Simulationsprogramm wurde das auf Matrix-Vektor-Multiplikationen basierende Programm des Autors von [3] herangezogen und angepasst. Die Genauigkeit der Streuparameteranalyse aus der Simulation einer Meanderleitung soll damit eine gute Übereinstimmung zu Messergebnissen erbracht haben. Laut Aussage der Autoren war die Implementierung des Algorithmus zum Zeitpunkt der Veröffentlichung nicht besonders effizient. Daher wurden wegen des damals frühen Stadiums des Programms auch keine Simulationszeiten zur Demonstration der Beschleunigung veröffentlicht. Ein weiterer offener Punkt ist die Betrachtung von SATS unter dem Aspekt der Stabilität. Dennoch ist der Ansatz vielversprechend, sodass man auf die Ergebnisse weiterer Untersuchungen sicherlich gespannt sein darf.

### 7.3 Listen-basierte WT der FDTD

Als Alternative zur Darstellung von Operatoren mit Matrizen wurde in Abschnitt 5.3 die Formulierung mit Listen entwickelt. Es ist nun zu klären, wie sich dies am praktikabelsten für FDTD-Simulationen einsetzen lässt. Dazu werden in den folgenden Abschnitten verschiedene FDTD-Operatoren betrachtet und ihre Listen-Elemente angegeben. Ferner wird ein Verfahren zur gegenseitigen Anpassung Wavelet-komprimierter FDTD-Operatoren hergeleitet und es werden neuartige optimierte Methoden zur Anregung und Ausgabe von Feldern im Wavelet-Bereich vorgestellt. Ebenso wird eine Vorgehensweise entwickelt, mit der sich durch die WT eine Art analytisches Untergitterverfahren für die FDTD gewinnen lässt, welches die Basis für die angestrebten Reduktionen von Simulationszeiten ist. Abschließend wird demonstriert, wie sich sogar FDTD-Operatoren mit nichtlinearen Eigenschaften einbeziehen lassen.

Zum Einstieg soll gezeigt werden, wie ein Übergang auf eine Darstellung mit Listen vonstatten gehen kann, wenn FDTD-Operatoren bereits in Matrixform vorliegen. Doch zuvor soll noch die Notation der Operatoren festgelegt werden.

Die Berechnung eines Feldes für den nächsten Zeitschritt wurde in Gleichung (2.20) anhand der x-Komponente des elektrischen Feldes demonstriert. Für alle Feldkomponenten zusammen kann dies als

$$\mathbf{E}^{n+\frac{1}{2}} = \mathcal{O}_{\mathbf{E}}^{\mathbf{E}} \mathbf{E}^{n-\frac{1}{2}} + \mathcal{O}_{\mathbf{E}}^{\mathbf{H}} \mathcal{D}^{\mathbf{H}} \mathbf{H}^n \quad (7.10)$$

geschrieben werden.  $\mathcal{O}_{\mathbf{E}}^{\mathbf{E}}$  und  $\mathcal{O}_{\mathbf{E}}^{\mathbf{H}}$  sind Skalierungsoperatoren und enthalten die Angaben über die Materialverteilung und die Zeitschrittweite.  $\mathcal{D}^{\mathbf{H}}$  ist der diskretisierte Rotationsoperator für das magnetische Feld.

Die Wavelet-transformierten Operatoren  $\tilde{\mathcal{O}}_{\mathbf{E}}^{\mathbf{E}}$ ,  $\tilde{\mathcal{O}}_{\mathbf{E}}^{\mathbf{H}}$  und  $\tilde{\mathcal{D}}^{\mathbf{H}}$  sind nach bekanntem Schema durch Verkettung von  $\mathcal{O}_{\mathbf{E}}^{\mathbf{E}}$ ,  $\mathcal{O}_{\mathbf{E}}^{\mathbf{H}}$  und  $\mathcal{D}^{\mathbf{H}}$  mit  $\mathcal{P}$  und  $\mathcal{P}^*$  zu berechnen. Ist für einen Operator  $\mathcal{T}$  bereits eine Matrix-Darstellung bekannt, so kann diese leicht in eine Listen-Darstellung überführt werden. Dazu sei  $\mathcal{T} \in U^\nu$  und für die Matrixelemente gelte entsprechend die Darstellung  $t_{a,b}(\mathbf{l}_\nu)$  mit  $\mathbf{l}_\nu = \mathbf{k} \downarrow \mathbf{e}_\nu$ ,  $a = k_\nu$ . Nach Abschnitt 5.3.1 bestimmt sich das zugehörige Listenelement zu

$$O^1(\mathbf{k}) = (t_{k_\nu,b}(\mathbf{l}_\nu), (b - k_\nu)\mathbf{e}_\nu) . \quad (7.11)$$

Dadurch können sämtliche Operatoren, für die eine Matrixformulierung bekannt ist, unmittelbar in eine Listen-basierte Darstellung überführt werden.

Explizit aufgeführt wurde bislang die Matrix-Darstellung der eindimensionalen WT, siehe Abschnitt 3.2.5. Die relativ einfachen Erweiterungen zu einer mehrdimensionalen Transformation in Kapitel 4 beruhen ebenfalls auf der Verwendung von Matrizen. Statt des Umwegs über Matrizen können die Gleichungen (3.47) und (3.50) auch direkt zur Bildung von  $O^2$ -Listen benutzt werden.

### 7.3.1 Operatoren der klassischen FDTD

Die Skalierungsoperatoren  $\mathcal{O}_{\mathbf{E}}^{\mathbf{E}}$  und  $\mathcal{O}_{\mathbf{E}}^{\mathbf{H}}$  lassen sich ausgesprochen einfach realisieren. Für den  $E_x$ -Anteil aus  $\mathcal{O}_{\mathbf{E}}^{\mathbf{E}}$  beispielsweise gilt

$$O^1(i, j, k) = (a_2, \mathbf{0}) , \quad (7.12)$$

siehe Gleichungen (2.20) und (7.8). Zu jeder Bildkoordinate  $(i, j, k)^T$  kann der Skalierungskoeffizient also direkt berechnet und eingetragen werden, ohne erst nach Bereichen mit gleichen Materialeigenschaften suchen zu müssen, wie es in Abschnitt 7.2.3 der Fall ist.

Für die Implementierung des Rotationsoperators wird dieser in Differenzierungen entlang der Koordinatenachsen aufgeteilt, d.h.  $\mathcal{D}^{\mathbf{H}} = \mathcal{D}_x^{\mathbf{H}} + \mathcal{D}_y^{\mathbf{H}} + \mathcal{D}_z^{\mathbf{H}}$ . Dann besteht der  $E_x$ -Anteil aus  $\mathcal{D}_z^{\mathbf{H}}$  entsprechend Gleichung (2.20) aus den beiden Listenelementen

$$O_1^1(i, j, k) = \left( -\frac{1}{\Delta z}, \mathbf{0} \right), \quad (7.13)$$

$$O_2^1(i, j, k) = \left( \frac{1}{\Delta z}, -\mathbf{e}_z \right). \quad (7.14)$$

Als nächstes sind elektrische und magnetische Wände zu berücksichtigen. Dazu werden die zugehörigen Maskierungsoperatoren  $\mathcal{M}^{\mathbf{E}}$  und  $\mathcal{M}^{\mathbf{H}}$  definiert. Als Beispiel sollen alle elektrischen Wände betrachtet werden, die parallel zur  $xy$ - oder  $xz$ -Ebene liegen und somit die tangential Komponente  $E_x$  zu Null setzen. Die Listenelemente bestimmen sich zu

$$O^1(i, j, k) = \begin{cases} (0, \mathbf{0}) & , (i, j, k)^T \text{ liegt in einer } xy\text{- oder } xz\text{-Wand} \\ (1, \mathbf{0}) & , \text{sonst} \end{cases}. \quad (7.15)$$

Der obere Zweig in Gleichung (7.15) ist genau genommen überflüssig, da  $O^1$ -Elemente mit einem Faktor von Null nicht zu berücksichtigen sind, vgl. Abschnitt 5.3.1. In Gleichung (7.15) soll damit lediglich veranschaulicht werden, dass für eine Koordinate „fehlende“  $O^1$ -Elemente mit solchen identisch sind, deren Faktoren den Wert Null haben.

Diese Art der Implementierung elektrischer Wände entspricht dem in Abschnitt 2.3.1 zuerst genannten Fall, bei dem sich tangential Feldkomponenten genau in der Wand befinden. Der zweite Fall, die Positionierung elektrischer Wände symmetrisch zwischen tangentialen Feldkomponenten, erfordert eine andere Herangehensweise. Es ist hierbei das Spiegelungsprinzip anzuwenden, wie es in Gleichung (2.27) zum Ausdruck kommt. Im Folgenden wird dieser Fall jedoch nicht weiter betrachtet.

Nachdem die Listenelemente für alle Wände und Feldkomponenten ermittelt wurden, sind die Operatoren aus Gleichung (7.10) durch Verkettung

zu maskieren:

$$\mathbf{E}^{n+\frac{1}{2}} = \mathcal{O}_{\mathbf{E}}^{\mathbf{E}'} \mathbf{E}^{n-\frac{1}{2}} + \mathcal{O}_{\mathbf{E}}^{\mathbf{H}'} \mathcal{D}^{\mathbf{H}'} \mathbf{H}^n \quad (7.16)$$

mit

$$\mathcal{O}_{\mathbf{E}}^{\mathbf{E}'} = \mathcal{O}_{\mathbf{E}}^{\mathbf{E}} \mathcal{M}^{\mathbf{E}} (= \mathcal{M}^{\mathbf{E}} \mathcal{O}_{\mathbf{E}}^{\mathbf{E}} = \mathcal{M}^{\mathbf{E}} \mathcal{O}_{\mathbf{E}}^{\mathbf{E}} \mathcal{M}^{\mathbf{E}}), \quad (7.17)$$

$$\mathcal{O}_{\mathbf{E}}^{\mathbf{H}'} = \mathcal{O}_{\mathbf{E}}^{\mathbf{H}} \mathcal{M}^{\mathbf{E}} (= \mathcal{M}^{\mathbf{E}} \mathcal{O}_{\mathbf{E}}^{\mathbf{H}} = \mathcal{M}^{\mathbf{E}} \mathcal{O}_{\mathbf{E}}^{\mathbf{H}} \mathcal{M}^{\mathbf{E}}), \quad (7.18)$$

$$\mathcal{D}^{\mathbf{H}'} = \mathcal{D}^{\mathbf{H}} \mathcal{M}^{\mathbf{H}}. \quad (7.19)$$

Analog ist mit Bereichen endlicher Dicke zu verfahren, für die eine ideale Leitfähigkeit angenommen wird und die durch das Ausmaskieren von Feldwerten realisiert werden.

Die Erweiterung des Simulationsgebietes entsprechend Abbildung 7.1 auf Seite 98 ist als vorbereitende Maßnahme für die WT selbstverständlich auch bei der Listen-basierten Darstellung der FDTD durchzuführen. Im Gegensatz zu der in Abschnitt 7.2 behandelten Formulierung mit Matrizen sind hier jedoch keine Erweiterungen der Differenziations-, Skalierungs- und Randoperatoren notwendig. Sämtliche Koordinaten des Ergänzungsraumes des  $\mathbf{E}$ - und des  $\mathbf{H}$ -Feldes können stattdessen ausmaskiert werden, wie es ähnlich auch bei elektrischen und magnetischen Wänden der Fall ist. Das gilt natürlich nicht für die WT-Operatoren  $\mathcal{P}$  und  $\mathcal{P}^*$ . In die transformierten Operatoren (z. B.  $\tilde{\mathcal{O}}_{\mathbf{E}}^{\mathbf{E}}$ ) gehen diese ausmaskierten Bereiche jedoch automatisch ein, sodass auch keine Notwendigkeit zur Modifikation von  $\mathcal{P}$  und  $\mathcal{P}^*$  besteht.

Abschließend soll noch erwähnt werden, dass durch die Möglichkeit der direkten Verkettung von Operatoren in der Listen-basierten Darstellung Abbildungen wie  $\mathcal{O}_{\mathbf{E}}^{\mathbf{H}'} \mathcal{D}^{\mathbf{H}'}$  zu einem einzelnen Operator zusammengefasst werden können. Das reduziert i. A. die Berechnung von Zwischenergebnissen und damit auch den Speicherbedarf sowie die Anzahl von Speicherzugriffen. Dennoch werden aus Gründen der Übersichtlichkeit und Allgemeinheit in den weiteren Betrachtungen Operatoren meistens wie bisher separat aufgeführt werden. In vielen Fällen ist das Ausführen separater Operatoren numerisch sogar effizienter als bei einem einzelnen Gesamtoperator. Das trifft beispielsweise auf die mehrstufige WT zu. Dennoch ist jeder Fall im Einzelnen auf die Eignung zur Vereinigung von Operatoren zu untersuchen.

### 7.3.2 Automatische Bereichsanpassung

Der vorangegangene Abschnitt zeigte, wie Abbildungseigenschaften einzelner Operatoren durch elektrische oder magnetische Wände reduziert werden. Die Reduktion eines einzelnen Operators hat im Allgemeinen auch Auswirkungen auf andere Operatoren, insbesondere wenn diese direkt miteinander verkettet sind. Je nach Umfang der Reduktion können auch diese Operatoren davon durch eine Verringerung der numerischen Operationen profitieren. Konkreter formuliert bedeutet das die Bereichsanpassung von Operatoren. Damit ist eine Reduktion der Operatoren auf die tatsächlich nur benutzten Bild- und Urbildkoeffizienten gemeint. Zur Verdeutlichung benötigen wir zunächst die Gleichung zur Berechnung des magnetischen Feldes:

$$\mathbf{H}^{n+1} = \mathcal{O}_{\mathbf{H}}^{\mathbf{H}'} \mathbf{H}^n + \mathcal{O}_{\mathbf{H}}^{\mathbf{E}'} \mathcal{D}^{\mathbf{E}'} \mathbf{E}^{n+\frac{1}{2}}, \quad (7.20)$$

mit

$$\mathcal{O}_{\mathbf{H}}^{\mathbf{H}'} = \mathcal{O}_{\mathbf{H}}^{\mathbf{H}} \mathcal{M}^{\mathbf{H}} (= \mathcal{M}^{\mathbf{H}} \mathcal{O}_{\mathbf{H}}^{\mathbf{H}} = \mathcal{M}^{\mathbf{H}} \mathcal{O}_{\mathbf{H}}^{\mathbf{H}} \mathcal{M}^{\mathbf{H}}), \quad (7.21)$$

$$\mathcal{O}_{\mathbf{H}}^{\mathbf{E}'} = \mathcal{O}_{\mathbf{H}}^{\mathbf{E}} \mathcal{M}^{\mathbf{H}} (= \mathcal{M}^{\mathbf{H}} \mathcal{O}_{\mathbf{H}}^{\mathbf{E}} = \mathcal{M}^{\mathbf{H}} \mathcal{O}_{\mathbf{H}}^{\mathbf{E}} \mathcal{M}^{\mathbf{H}}), \quad (7.22)$$

$$\mathcal{D}^{\mathbf{E}'} = \mathcal{D}^{\mathbf{E}} \mathcal{M}^{\mathbf{E}}. \quad (7.23)$$

Als Beispiel wird eine ideal leitfähige Kugel betrachtet, deren Streuverhalten simuliert werden soll. Durch die ideale Leitfähigkeit ist das Innere der Kugel feldfrei.<sup>6</sup> Der Maskierungsoperator  $\mathcal{M}^{\mathbf{E}}$  führt demgemäß in Gleichung (7.16) zu reduzierten Operatoren, sodass die elektrischen Feldkomponenten des Kugelinneren und der Kugeloberfläche nicht berechnet werden. Mit dem bisherigen Schema wird allerdings keine Reduktion des magnetischen Feldes erreicht. Dies kann jedoch mit der folgenden Methode erreicht werden, bei der es um eine iterative Anpassung von Bild- und Urbildkoeffizienten geht.

Zuerst sollen zwei Operatoren  $\mathcal{T}_1$  und  $\mathcal{T}_2$  betrachtet werden.  $\lambda_{\mathbf{B}}(\mathcal{T}_1)$  ergibt den Bildraum von  $\mathcal{T}_1$ . Hiermit sind alle Bildkoordinaten gemeint, für die  $\mathcal{T}_1 f$  einen nicht-verschwindenden Wert liefern kann. Analog dazu ergibt  $\lambda_{\mathbf{U}}(\mathcal{T}_2)$  den Urbildraum, d. h. die Menge aller Koordinaten, an denen der Wert von  $f$  für die Abbildung  $\mathcal{T}_2 f$  eine Rolle spielt. Sind zwei Funktionen

<sup>6</sup>Stationäre magnetische Felder werden hier außer Acht gelassen.

$f_1, f_2$  auf  $\lambda_U(\mathcal{T}_2)$  identisch und ansonsten unterschiedlich, ist dennoch  $\mathcal{T}_2 f_1 = \mathcal{T}_2 f_2$ .

Die Schnittmenge der beiden Bereiche ist

$$\lambda(\mathcal{T}_2, \mathcal{T}_1) = \lambda_U(\mathcal{T}_2) \cap \lambda_B(\mathcal{T}_1) . \quad (7.24)$$

Zu  $\lambda(\mathcal{T}_2, \mathcal{T}_1)$  lässt sich jetzt ein Maskierungsoperator  $\mathcal{M}_\lambda = \mathcal{M}_\lambda(\mathcal{T}_2, \mathcal{T}_1)$  bestimmen, sodass gilt

$$\mathcal{T}_2 \mathcal{T}_1 = (\mathcal{T}_2 \mathcal{M}_\lambda)(\mathcal{M}_\lambda \mathcal{T}_1) = \mathcal{T}'_2 \mathcal{T}'_1 . \quad (7.25)$$

Die eigentliche Wellenausbreitung wird durch die Ausdrücke  $\mathcal{O}_{\mathbf{E}}^{\mathbf{H}'} \mathcal{D}^{\mathbf{H}'}$  und  $\mathcal{O}_{\mathbf{H}}^{\mathbf{E}'} \mathcal{D}^{\mathbf{E}'}$  in den Gleichungen (7.16) und (7.20) bewirkt.  $\mathcal{O}_{\mathbf{E}}^{\mathbf{E}'}$  und  $\mathcal{O}_{\mathbf{H}}^{\mathbf{H}'}$  beeinflussen jeweils nur das eigene Feld und dabei sogar nur jeden Raumpunkt für sich. Setzen wir

$$\mathcal{T}_1 = \mathcal{O}_{\mathbf{E}}^{\mathbf{H}'} \mathcal{D}^{\mathbf{H}'}, \quad \mathcal{T}_2 = \mathcal{O}_{\mathbf{H}}^{\mathbf{E}'} \mathcal{D}^{\mathbf{E}'} \quad (7.26)$$

und vernachlässigen wir zunächst  $\mathcal{O}_{\mathbf{E}}^{\mathbf{E}'}$  und  $\mathcal{O}_{\mathbf{H}}^{\mathbf{H}'}$ , so führt das Einsetzen der Gleichungen (7.16) und (7.20) ineinander über mehrere Zeitschritte betrachtet zu einer Operatorverkettung

$$\mathcal{T}_2 \mathcal{T}_1 \mathcal{T}_2 \mathcal{T}_1 \mathcal{T}_2 \mathcal{T}_1 \dots \quad (7.27)$$

Mit

$$\mathcal{M}_\lambda^{1,l} = \mathcal{M}_\lambda(\mathcal{T}_2^l, \mathcal{T}_1^l), \quad \mathcal{M}_\lambda^{2,l} = \mathcal{M}_\lambda(\mathcal{T}_1^l, \mathcal{T}_2^l), \quad l \in \mathbb{N}, \quad (7.28)$$

und

$$\mathcal{T}_1^{2l+1} = \mathcal{M}_\lambda^{1,2l} \mathcal{T}_1^{2l}, \quad \mathcal{T}_2^{2l+1} = \mathcal{T}_2^{2l} \mathcal{M}_\lambda^{1,2l}, \quad (7.29)$$

$$\mathcal{T}_1^{2l+2} = \mathcal{T}_1^{2l+1} \mathcal{M}_\lambda^{2,2l+1}, \quad \mathcal{T}_2^{2l+2} = \mathcal{M}_\lambda^{2,2l+1} \mathcal{T}_2^{2l+1}, \quad (7.30)$$

können die Operatoren nun so lange aneinander angepasst werden, bis  $\mathcal{T}_1^{l+2} = \mathcal{T}_1^l$  und  $\mathcal{T}_2^{l+2} = \mathcal{T}_2^l$  gilt.<sup>7</sup> Die Maskierungsoperatoren des letzten Anpassungsschrittes sollen mit  $\mathcal{M}_\lambda^{1,\infty}$  und  $\mathcal{M}_\lambda^{2,\infty}$  bezeichnet werden. Mit

<sup>7</sup>Es genügt nicht,  $\mathcal{T}_{1,2}^{l+1} = \mathcal{T}_{1,2}^l$  zu fordern. So mögen  $\mathcal{T}_1$  und  $\mathcal{T}_2$  zu Beginn in der Verkettung  $\mathcal{T}_1 \mathcal{T}_2$  bereits einander angepasst sein, was jedoch nicht unbedingt für  $\mathcal{T}_2 \mathcal{T}_1$  gelten muss.

ihnen können schließlich auch die Skalierungsoperatoren  $\mathcal{O}_{\mathbf{E}}^{\mathbf{E}'}$  und  $\mathcal{O}_{\mathbf{H}}^{\mathbf{H}'}$  reduziert werden:

$$\mathcal{O}_{\mathbf{E}}^{\mathbf{E}',l} = \mathcal{O}_{\mathbf{E}'}^{\mathbf{E}'} \mathcal{M}_{\lambda}^{1,\infty}, \quad \mathcal{O}_{\mathbf{H}}^{\mathbf{H}',l} = \mathcal{O}_{\mathbf{H}'}^{\mathbf{H}'} \mathcal{M}_{\lambda}^{2,\infty}. \quad (7.31)$$

Ginge es bei der Bereichsanpassung von Operatoren nur um elektrisch oder magnetisch ideal leitende Bereiche, ließen sich vermutlich auch einfachere Vorgehensweisen anwenden. Die Hauptanwendung des hier entwickelten Verfahrens liegt jedoch in der Generierung sog. *Detailgitter*, welche in Abschnitt 7.3.6 vorgestellt werden.

### 7.3.3 Absorbierende Randbedingungen

Liegen Operatoren absorbierender Randbedingungen bereits in einer Matrixformulierung vor, lassen sich diese wie andere Matrixoperatoren auch nach dem bereits gezeigten Schema mit Listen darstellen. Bei den in Abschnitt 7.2.2 diskutierten Implementierungen von PML- und A-UPML-Operatoren wurde u. a. erwähnt, dass eine Verdopplung von Feldkomponenten über den gesamten Simulationsraum vonnöten ist. Gibt man jedoch die  $U_0'$ -Zugehörigkeit und damit eine einfache Matrixformulierung der (Teil-)Operatoren auf, lässt sich eine effizientere Implementierung angeben.

Die Feldkoeffizienten der PML werden in die zwei Gruppen  $\{F_{xy}, F_{yz}, F_{zx}\}$  und  $\{F_{yx}, F_{zy}, F_{xz}\}$  mit  $F \in \{E, H\}$  aufgeteilt. Die Feldkoeffizienten der ersten Gruppe werden nach der Diskretisierung des Raumes wie bisher direkt um den eigentlichen Feldraum angeordnet, siehe Abbildung 2.2 auf Seite 20. Die Komponenten der zweiten Gruppe werden in eigenen Bereichen an die Bereiche der ersten Gruppe angelagert. Diese Anordnung führt gewissermaßen zu einer zweiten Schale um den Feldraum, wobei die zweite Schale nicht vollständig gefüllt ist. In Abbildung 7.2 ist diese räumliche Struktur visualisiert.

Erreicht wird auf diese Weise, dass über den gesamten Simulationsraum nur jeweils drei Komponenten für das elektrische und das magnetische Feld vorhanden sind. Obwohl der Raum der zweiten Schale nicht vollständig ausgeschöpft wird, ist die Raumnutzung trotzdem deutlich besser als bei der Speicherung von jeweils sechs Komponenten, wie es

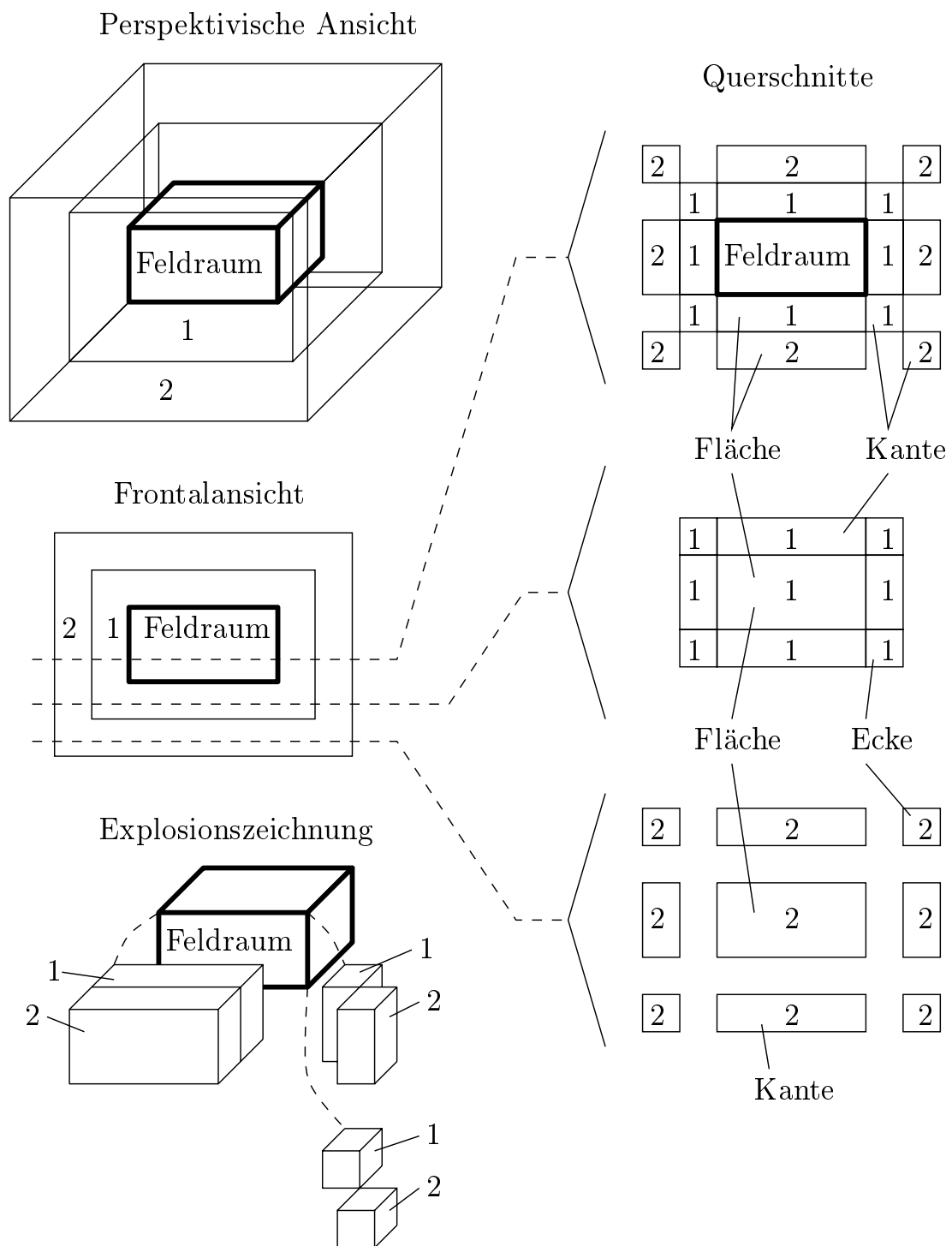


Abbildung 7.2: Anordnung der PML-Feldkomponenten gemäß ihrer Gruppenzugehörigkeit (1, 2). In der Explosionszeichnung ist die Aufteilung von drei PML-Bereichen perspektivisch veranschaulicht.

in Abschnitt 7.2.2 beschrieben wurde. Auf die A-UPML lässt sich die effizientere Anordnung ganz analog anwenden.

Gleichung (2.28) auf Seite 19 gibt beispielhaft an, dass zwischen den beiden Gruppen von Feldkomponenten Abhängigkeiten bestehen. Übertragen auf die neue Anordnung bedeutet das, dass z. B. zwischen den Feldkomponenten der PML-Ecken räumlich diagonale Abbildungen existieren. Das verdeutlicht die eingangs genannte Aufgabe der  $U_0'$ -Zugehörigkeit von PML-Operatoren. Eine Realisierung mit Matrix-basierten Operatoren würde daher recht umständliche Zerlegungen erfordern, wie sie in Abschnitt 5.2.3.3 beschrieben wurden.

### 7.3.4 Optimierte Feldanregung und -ausgabe

Die Matrix-basierte Simulation der Wavelet-transformierten FDTD erfordert nach Abschnitt 7.2.4 bei jedem Zeitschritt die WT des vollständigen Simulationsraumes. In den meisten Fällen benötigt man zur Analyse einer Simulation jedoch nicht die Kenntnis des vollständigen Raumes, sondern nur einzelner kleiner Bereiche. Mit der Definition von Anregungs- und Ausgabeoperatoren und ihrer direkten Verkettung mit  $\mathcal{P}$  bzw.  $\mathcal{P}^*$  lassen sich effiziente, örtlich begrenzte Anregungen bzw. Ausgaben für den Waveletbereich realisieren.

Die Untersuchung des Resonanzverhaltens eines geschlossenen Resonators wird oft durch eine kurzzeitige punktförmige Anregung eingeleitet. Bei der Simulation des Streuverhaltens von Objekten lässt man i. d. R. eine ebene Welle vom Rand des Simulationsgebietes starten und auf das Objekt zulaufen. Streuparameter von Schaltungen werden ermittelt, indem in die Zuleitungen durch Feldanregung ein Signal eingekoppelt wird. In den letzten beiden Fällen findet die Anregung immer in einer einzelnen Ebene statt, im ersten Fall sogar nur an einem Punkt. Passend zu den anzuregenden Koordinaten kann ein Maskierungsoperator  $\mathcal{M}$  angegeben werden, der in allen drei genannten Fällen den weitaus größten Teil des Raumes ausblendet. Zur Anregung des Waveletbereichs ist dann

$$\tilde{\mathcal{M}} = \mathcal{P} \mathcal{M} \quad (7.32)$$

zu verwenden.  $\mathbf{E}_0$  sei eine gegebene elektrische Feldverteilung, mit der die

Anregung stattfinden soll. Dann wird

$$\tilde{\mathbf{E}}_0 = \tilde{\mathcal{M}}\mathbf{E}_0 \quad (7.33)$$

benutzt, um das Anregungsfeld für den Waveletbereich zu berechnen. Auch beim transformierten Feld  $\tilde{\mathbf{E}}_0$  werden nur wenige Stellen von Null verschiedene Werte annehmen, wenn das bereits für das Feld im Ursprungsraum galt. Der Anregungsoperator  $\tilde{\mathcal{M}}$  wird durch eine entsprechend geringe Anzahl nicht-verschwindender Tensorelemente repräsentiert und ist daher numerisch deutlich effizienter als die WT des vollständigen Raumes.

Analog zur Anregung sind bei Ausgaben die Feldwerte ebenfalls oft nur an einzelnen Punkten oder Flächen von Interesse. Bezeichnen wir den zugehörigen Maskierungsoperator wieder mit  $\mathcal{M}$ , so liefert

$$\mathbf{E}_0 = \tilde{\mathcal{M}}\tilde{\mathbf{E}}_0 \quad (7.34)$$

mit

$$\tilde{\mathcal{M}} = \mathcal{M}\mathcal{P}^* \quad (7.35)$$

das gesuchte Ausgabefeld  $\mathbf{E}_0$  im Grundraum.

Damit sind die Möglichkeiten bei der Verkettung von Anregungs- bzw. Ausgabeoperatoren und WT-Operatoren aber noch lange nicht ausgeschöpft. Es bietet sich oft an, auch die räumliche Analyse des Feldes mit Operatoren direkt im Waveletbereich zu bewerkstelligen. Zwei einfache Beispiele hierzu sind in Abbildung 7.3 angegeben.

In Abbildung 7.3a wird der Querschnitt einer Mikrostreifenleitung gezeigt. Eingezeichnet sind ein offener und ein geschlossener Integrationsweg zur Bestimmung der Spannung  $u$  bzw. des eingeschlossenen Stromes  $i$  aus dem elektrischen bzw. dem magnetischen Feld. Nach der Diskretisierung der Struktur bestehen die Integrationsoperatoren  $\mathcal{T}_u$  und  $\mathcal{T}_i$  aus einer Summation entsprechender Feldkoeffizienten entlang des Integrationsweges.<sup>8</sup> Dann ergeben

$$u = \tilde{\mathcal{T}}_u \tilde{\mathbf{E}} \quad \text{mit} \quad \tilde{\mathcal{T}}_u = \mathcal{T}_u \mathcal{P}^* \quad (7.36)$$

---

<sup>8</sup>Der Einfachheit halber sollen geometrieabhängige Gewichtungsfaktoren hier unberücksichtigt bleiben.

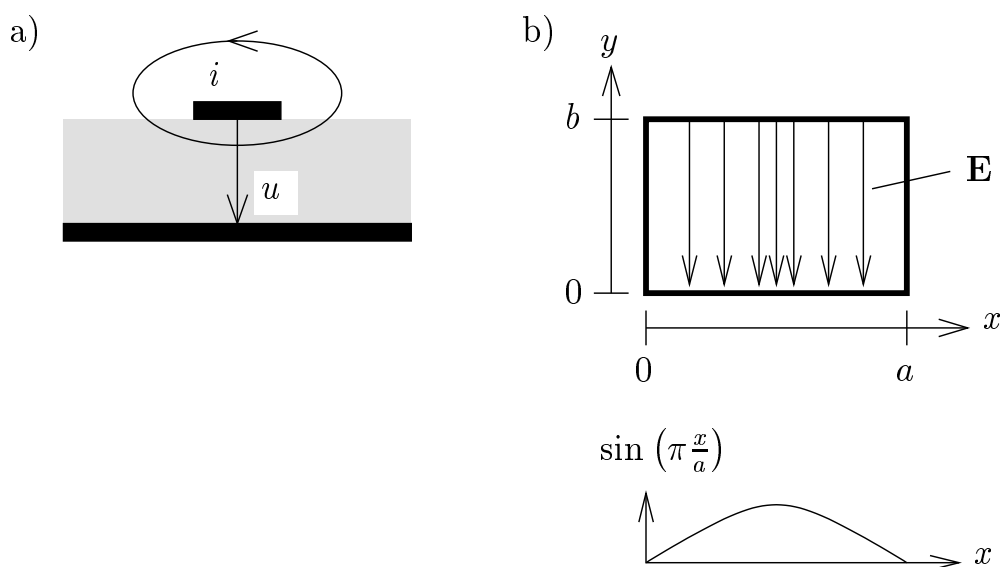


Abbildung 7.3: Beispiele für die räumliche Auswertung von Feldverläufen aus Simulationen. Bild a: Spannungs- und Stromstärkenbestimmung bei einer Mikrostreifenleitung. Bild b, oben: Transversale Verteilung des elektrischen Feldes beim  $TE_{10}$ -Wellentyp. Unten: Ortsabhängigkeit von  $E_y$ .

und

$$i = \tilde{\mathcal{T}}_i \tilde{\mathbf{H}} \quad \text{mit} \quad \tilde{\mathcal{T}}_i = \mathcal{T}_i \mathcal{P}^* \quad (7.37)$$

direkt aus dem Waveletbereich die gesuchte Spannung bzw. Stromstärke. Häufig sind auch aus Hohlleitern austretende Signale auf ihre Zusammensetzung aus Moden zu untersuchen. Gegebenenfalls interessiert sogar nur ein einzelner Wellentyp. Abbildung 7.3b skizziert die elektrische Feldverteilung des  $H_{10}$ -Wellentyps eines Rechteckhohlleiters. Soll der zeitliche Verlauf dieser TE-Welle untersucht werden, so können durch Auswertung von

$$E_y^{10}(z_0, t) = \int_0^a \int_0^b E_y(x, y, z_0, t) \sin\left(\pi \frac{x}{a}\right) dy dx \quad (7.38)$$

alle anderen Wellentypen quasi ausgeblendet werden.<sup>9</sup> Die Entwicklung von Gleichung (7.38) in das Funktionensystem des Simulationsverfahrens

<sup>9</sup>Es handelt sich bei Gleichung (7.38) um eine Art Galerkin-Verfahren zur Entwicklung des Feldes in die Strukturfunktionen der Moden des Rechteckhohlleiters. In diesem Fall geht es um den  $H_{10}$ -Wellentyp, dessen Strukturfunktion die Ortsabhängigkeit  $\sin\left(\pi \frac{x}{a}\right)$  hat [28].

führt dann auf die Darstellung

$$E_y^{10} = \mathcal{T}_{H_{10}} \mathbf{E} \quad \text{mit} \quad \mathcal{T}_{H_{10}} = \mathcal{T}_{\Sigma}^y \mathcal{O}_{\text{sin}} . \quad (7.39)$$

$\mathcal{O}_{\text{sin}}$  gewichtet jeden Feldkoeffizienten entsprechend der Ortsabhängigkeit der Strukturfunktion.  $\mathcal{T}_{\Sigma}^y$  summiert dann bei  $z = z_0$  alle y-Komponenten dieses Zwischenschrittes zu einem skalaren Wert. Durch die Berechnung von

$$\tilde{\mathcal{T}}_{H_{10}} = \mathcal{T}_{H_{10}} \mathcal{P}^* \quad (7.40)$$

erhält man so einen einzelnen Operator, der den zeitlichen Verlauf des  $H_{10}$ -Wellentyps bei  $z = z_0$  direkt aus dem Waveletbereich herausfiltert.

Die hier vorgestellten Beispiele stellen nur einen sehr kleinen Auszug aus allen Möglichkeiten zur Auswertung von Feldverläufen dar. Eine Voraussetzung zur Verlagerung einer Auswertung in den Waveletbereich ist ihre Linearität. Die Berechnung des Leistungstransportes mittels des Poynting-Vektors oder die Berechnung des Energieinhaltes sind nichtlineare Operationen, die nur mit anderen Methoden im Waveletbereich durchgeführt werden können.<sup>10</sup> Weiterhin ist in jedem Fall zu beachten, dass der auszuwertende (Urbild-)Bereich keinen zu großen Raum einnimmt, da sich ansonsten eine direkte Verkettung, d. h. die Berechnung eines einzelnen Auswertungsoperators, nicht mehr lohnt.

### 7.3.5 Teilbereichs-WT

Unter bestimmten Umständen ist es vorteilhaft, lediglich einen Teil des Simulationsraumes einer WT zu unterziehen statt den vollständigen Raum. Das ist vor allem dann der Fall, wenn ein Operator durch die Transformation in den Waveletbereich aus numerischer Sicht ineffizient wird. Ein Beispiel soll das verdeutlichen. Gilt für einen Operator  $\mathcal{T} = \text{id}$ , so ist  $\tilde{\mathcal{T}} = \mathcal{P} \mathcal{T} \mathcal{P}^* = \text{id}$ . Die Identität bedeutet, dass außer den Tensorelementen  $\tilde{t}_{\mathbf{k},\mathbf{k}}$  alle anderen identisch Null sind, d. h. der Operator  $\tilde{\mathcal{T}}$  ist im Vergleich zu „praxisnäheren“ Operatoren numerisch ziemlich effizient. Sobald ein Tensorelement von  $\mathcal{T}$  von diesem „Idealfall“ abweicht, ist  $\tilde{\mathcal{T}}$

<sup>10</sup>Hierzu ist die Darstellung der Multiplikation von Signalen im Waveletbereich vonnöten. In [29] wird ein Algorithmus dafür vorgestellt.

i. d. R. kein Skalierungsoperator mehr, da jetzt auch für einige Koordinaten  $\mathbf{k}' \neq \mathbf{k}$  gilt, dass  $\tilde{t}_{\mathbf{k},\mathbf{k}'} \neq 0$  ist. Mit steigender Inhomogenität der Abbildungseigenschaften von  $\mathcal{T}$  muss mit einer noch stärker steigenden Inhomogenität der Abbildungseigenschaften von  $\tilde{\mathcal{T}}$  gerechnet werden.

Differenziationsoperatoren sind normalerweise weitgehend homogen. Ideal leitende Wände reduzieren die Homogenität nur verhältnismäßig wenig. Abgestufte Gitter, d. h. weitläufige Schwankungen in der räumlichen Diskretisierungsschrittweite (*graded mesh*, vgl. Fußnote 1 auf Seite 54), üben hier einen wesentlich stärkeren Einfluss aus. Das Ausblenden dieser Bereiche ist kaum praktikabel, da sie für gewöhnlich mitten im Simulationsgebiet (um metallische Kanten und Ecken herum) eingesetzt werden. Sie sollten daher auf ein notwendiges Minimum reduziert werden.

Die zu simulierenden Strukturen weisen in den wohl meisten Fällen nur sehr wenige unterschiedliche Materialparameter auf, d. h. die Gebiete sind stückweise homogen, was einer WT der zugehörigen Skalierungsoperatoren zugute kommt. Beim Abschluss dieser Gebiete mit absorbierenden Randbedingungen sieht das aber u. U. ganz anders aus. Dabei stellen einfache ABCs wie die von Mur kaum ein Problem dar, da diese aus sehr wenigen numerischen Operationen bestehen, vgl. [3]. PMLs mit Schichtdicken von vier und mehr Zellen erfordern da schon deutlich mehr Operationen. Was aber vielleicht noch stärker wiegt ist die Ortsabhängigkeit der Materialparameter, die zur Erzielung optimaler Reflexionsfaktoren so gewählt wird, dass die Absorption mit zunehmender Tiefe in der PML steigt. Während diese Ortsabhängigkeit bei PML-Flächen nur eindimensional ist, ist sie bei PML-Kanten bereits zweidimensional und bei Ecken dreidimensional (vgl. Abschnitt 2.3.2). Von Bedeutung ist auch die Verdopplung des PML-Koeffizientenraumes durch die Aufspaltung der PML-Anteile in zwei Gruppen, wie sie in Abschnitt 7.3.3 beschrieben wurde.

Die Vorgehensweise für eine Teilbereichs-WT soll anhand eines zweidimensionalen Beispiels in Abbildung 7.4 besprochen werden. Zum Vergleich soll das gleiche Simulationsgebiet betrachtet werden wie in Abbildung 7.1. In dieses Gebiet ist links oben in Abbildung 7.4 der zu transformierende Teilraum eingezeichnet. Analog wie im Fall von Abbildung 7.1 ist auch der Teilraum durch einen Ergänzungsraum so weit zu vergrößern, wie es eine WT erfordert. Charakteristische Größen sind hier die zusätzliche Breite  $a$  und die zusätzliche Höhe  $b$ .

Im ersten Schritt wird der ursprüngliche Simulationsraum einfach um die

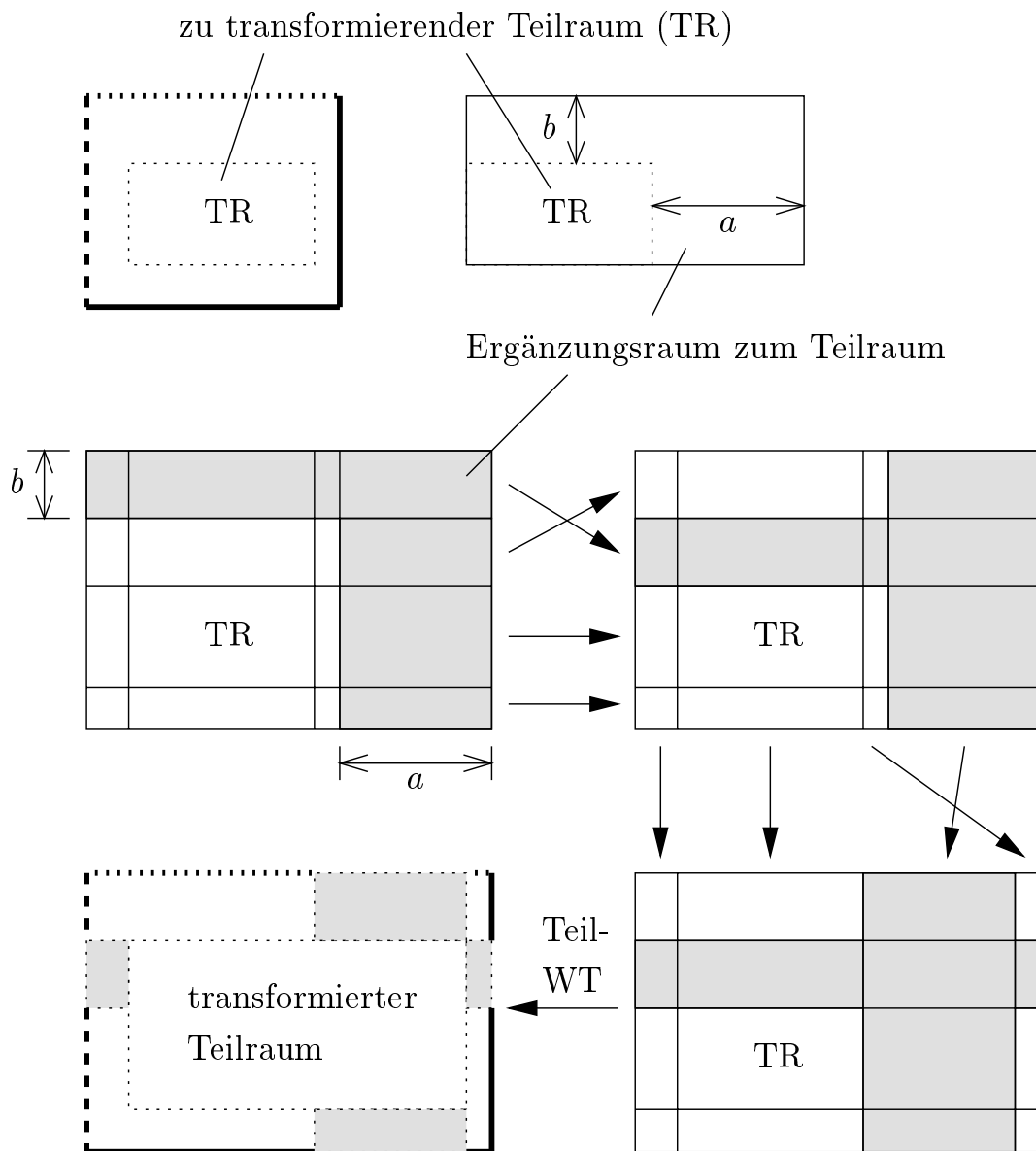


Abbildung 7.4: Transformation eines Teilraumes (TR) des Simulationsraumes aus Abbildung 7.1. Erforderlich Schritte sind: 1) Vergrößerung des Simulationsraumes um notwendigen Ergänzungsraum (graue Bereiche). 2) Vertauschung von Raumbereichen zur Erweiterung des Teilraumes. 3) Teilbereichs-WT im erweiterten Teilraum.

Maße  $a$  und  $b$  erweitert. Der so entstehende, zunächst noch funktionslose Raum ist in Abbildung 7.4 grau gefüllt. Durch die Vertauschung von Raumbereichen ist jetzt der Ergänzungsraum an den Teilraum anzulagern. Erreicht wird das in Abbildung 7.4 durch eine zeilen- und eine spalten-

tenweise Vertauschung. Anschließend kann der auf diese Weise erweiterte Teilraum einer WT unterzogen werden.

Die konkrete Umsetzung erfordert einige Formalismen. Zunächst ist für jede Richtung  $\nu$  die Definition eines Vertauschungsoperators  $\mathcal{S}_\nu$  erforderlich. Dieser verschiebt (Urbild-)Koeffizienten in Richtung  $\nu$  vom Bereich des Ergänzungsraumes direkt an den Teilraum. Gleichzeitig werden Koeffizienten dieses „Zielbereichs“ an die Stelle des Ergänzungsraumes verschoben. Alle anderen Koeffizienten bleiben unverändert (Identitätsabbildung). Zu dem Operator  $\mathcal{S}_\nu$  existiert daher ein eindeutiger inverser Operator  $\mathcal{S}_\nu^*$ , sodass

$$\mathcal{S}_\nu \mathcal{S}_\nu^* = \text{id} \quad (7.41)$$

gilt. Alle Vertauschungsoperatoren zusammen ergeben den Operator

$$\mathcal{S} = \bigcirc_{\nu} \mathcal{S}_\nu , \quad (7.42)$$

zu dem ebenfalls der inverse Operator  $\mathcal{S}^*$  existiert. Aus einer Abbildung

$$g = \mathcal{T} f \quad (7.43)$$

wird dadurch

$$g' = \mathcal{T}' f' \quad (7.44)$$

mit

$$g' = \mathcal{S} g , \quad (7.45)$$

$$f' = \mathcal{S} f , \quad (7.46)$$

$$\mathcal{T}' = \mathcal{S} \mathcal{T} \mathcal{S}^* . \quad (7.47)$$

Die WT ergibt dann

$$\tilde{g}' = \tilde{\mathcal{T}}' \tilde{f}' \quad (7.48)$$

mit

$$\tilde{g}' = \mathcal{P}' g' , \quad (7.49)$$

$$\tilde{f}' = \mathcal{P}' f' , \quad (7.50)$$

$$\tilde{\mathcal{T}}' = \mathcal{P}' \mathcal{T} \mathcal{P}^{*'} , \quad (7.51)$$

wobei  $\mathcal{P}'$  und  $\mathcal{P}^{*}$  im Gegensatz zu  $\mathcal{P}$  und  $\mathcal{P}^*$  Transformationen lediglich im Bereich des erweiterten Teilraumes durchführen und die Koeffizienten des übrigen Raumes unverändert lassen.

Die Vertauschung von Raumbereichen ist mit sämtlichen FDTD-Operatoren durchzuführen, so wie auch die Erweiterung des Simulationsraumes in Abbildung 7.1 alle FDTD-Operatoren betraf. Anregungs- und Ausgabeoperatoren sind hier mit eingeschlossen. Abschließend sei noch gesagt, dass die unterschiedlichen Vorgehensweisen der Abbildungen 7.1 und 7.4 im Allgemeinen auf verschieden große Koeffizientenräume führen.

## 7.3.6 Das Detailgitter-Verfahren

### 7.3.6.1 Motivation

Abschnitt 7.2.5.1 stellte im Rahmen der Simulation mit Matrix-basierten Operatoren ein räumlich adaptives Verfahren vor. Dieses Verfahren ist sehr allgemein gehalten und nicht an die Verwendung von Matrizen gebunden. Bei der Simulation mit Listen-basierten Operatoren lässt es sich daher ebenso anwenden, wobei die in Abschnitt 7.2.5.1 bereits genannten Beschränkungen der Effizienz auch hier gelten. Zur kurzen Wiederholung sei gesagt, dass ein Grund für die reduzierte Effizienz in dem Aufwand für die Aktualisierung der Verteilung der nicht-vernachlässigbaren Feldkoeffizienten zu finden ist. Der zweite wesentliche Grund ist die Tatsache, dass bei den Urbildern die Informationen über vernachlässigbare Koeffizienten nicht ausgewertet werden, d. h. es findet an dieser Stelle keine Kompression statt.

Für die FDTD wurden verschiedene Untergitter-Verfahren entwickelt, welche die Erhöhung der räumlichen Auflösung in auszuwählenden Bereichen erlauben. Einige dieser Verfahren werden in [4] vorgestellt. Ihnen allen ist gemein, dass eine geringere Zeitschrittweite verwendet werden muss, als es nach der klassischen FDTD möglich ist. Ohne diese Herabsetzung der Zeitschrittweite ist eine stabile Simulation nicht möglich. Eine weitere Gemeinsamkeit ist, dass das Maß der Herabsetzung immer nur empirisch ermittelt wurde und eine analytische Stabilitätsanalyse bisher nicht möglich war.

Im folgenden Abschnitt wird ein Detailgitter-Verfahren entwickelt, welches auch als *analytisches Untergitter-Verfahren* bezeichnet werden könnte. Einige Details hierzu wurden erstmalig in [30] vorgestellt. Es stellt eine Kombination der bisher genannten Methoden dar: Aus den Angaben zu Bereichen mit gewünschter höherer Auflösung wird auf analytischem Weg eine Ausmaskierung von Koeffizienten im Waveletbereich vorgenommen. Diese Art der Kompression ist anders als bei der räumlichen Adaptivität von statischer Natur. Der Vorteil ist jedoch, dass die Kompression auch grundsätzlich die Urbildkoeffizienten betrifft. Ferner fällt kein Aufwand für die Aktualisierung der Verteilung der nicht-vernachlässigbaren Koeffizienten an. Weiterhin kann ein Stabilitätskriterium vorgestellt werden.

### 7.3.6.2 Operatoren und Algorithmus

Das Verfahren beginnt mit der Auswahl der Bereiche, in denen die räumliche Auflösung erhalten bleiben soll (Detailbereiche). Dazu kann ein Maskierungsoperator  $\mathcal{M}_1$  angegeben werden, für dessen Tensorelemente nach der Diskretisierung gilt

$$m_{\mathbf{k},\mathbf{k}'}^1 = \begin{cases} 1 & , \mathbf{k} = \mathbf{k}' \wedge \mathbf{k} \text{ gehört zum Detailbereich} \\ 0 & , \text{sonst} \end{cases} . \quad (7.52)$$

Abbildung 7.5 zeigt dies am Beispiel eines koplanaren Wellenleiters. Eine linksseitige WT<sup>11</sup> ergibt

$$\mathcal{M}_2 = \mathcal{P} \mathcal{M}_1 . \quad (7.53)$$

Von  $\mathcal{M}_2$  interessiert nur der Bildraumbereich, in den Abbildungen stattfinden können, d. h.  $\lambda_{\mathbf{B}}(\mathcal{M}_2)$ . Hieraus wird aber noch kein Maskierungsoperator gebildet, da erst noch eine Ergänzung des Bildraumes stattfinden hat.

Die WT wird bis zur Skala  $\mathbf{j}$  durchgeführt. Der Raumbereich von  $\mathbf{V}_{\mathbf{j}}$  werde mit  $\lambda_{\mathbf{V}_{\mathbf{j}}}$  bezeichnet. In Abschnitt 7.2.5.1 wurde beschrieben, dass dieser Raumbereich beim Verfahren der räumlichen Adaptivität grundsätzlich

<sup>11</sup>In [30] ist eine beidseitige WT angegeben, d. h.  $\mathcal{M}_2 = \mathcal{P} \mathcal{M}_1 \mathcal{P}^*$ . Da in den weiteren Betrachtungen jedoch nur Bildräume interessieren, ist die linksseitige WT nach Gleichung (7.53) für das Verfahren vollkommen äquivalent.

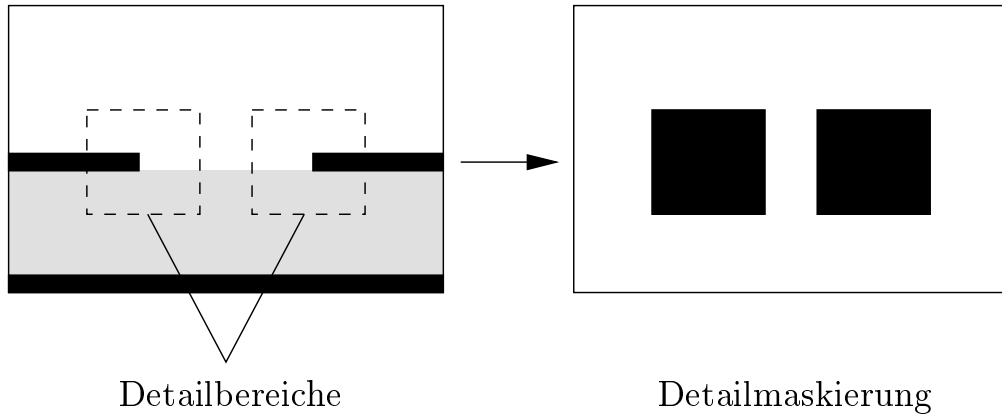


Abbildung 7.5: Detailmaskierung am Beispiel eines koplanaren Schlitzes. Nur die Feldkrümmungen an den Spitzen des Schlitzes sollen räumlich hoch aufgelöst werden (links). Die zugehörige Maske blendet im Grundraum alle anderen Bereich aus (weiße Fläche im rechten Bild).

bei jedem Zeitschritt vollständig berechnet wird. Verglichen wurde diese Regel mit einer Art Grunddiskretisierung, was auch der Anschauung entspricht. Hier dürfte es auch eine Rolle spielen, dass der Mittelwert bzw. das Moment nullter Ordnung von Skalierungsfunktionen Eins beträgt, wogegen Wavelets einen verschwindenden Mittelwert haben, vgl. [12].

Passend zu der Vereinigung beider Waveletbereiche,

$$\lambda_G = \lambda_B(\mathcal{M}_2) \cup \lambda_{\mathbf{v}_j} , \quad (7.54)$$

lässt sich nun ein Maskierungsoperator  $\mathcal{M}_G$  angeben, der das Detailgitter und die Grunddiskretisierung im Waveletbereich beschreibt. Nachdem die Gleichungen (7.16) und (7.20) einer Bereichsanpassung<sup>12</sup> unterzogen wurden (siehe Abschnitt 7.3.2), können sie in den Waveletbereich transformiert und dann auf die durch  $\lambda_G$  bestimmten Bereiche reduziert werden:

$$\tilde{\mathbf{E}}^{n+\frac{1}{2}} = \mathcal{M}_G \mathcal{P} \mathcal{O}_{\mathbf{E}}^{\mathbf{E}'} \mathcal{P}^* \mathcal{M}_G \tilde{\mathbf{E}}^{n-\frac{1}{2}} + \mathcal{M}_G \mathcal{P} \mathcal{O}_{\mathbf{E}}^{\mathbf{H}'} \mathcal{D}^{\mathbf{H}'} \mathcal{P}^* \mathcal{M}_G \tilde{\mathbf{H}}^n , \quad (7.55)$$

$$\tilde{\mathbf{H}}^{n+1} = \mathcal{M}_G \mathcal{P} \mathcal{O}_{\mathbf{H}}^{\mathbf{H}'} \mathcal{P}^* \mathcal{M}_G \tilde{\mathbf{H}}^n + \mathcal{M}_G \mathcal{P} \mathcal{O}_{\mathbf{H}}^{\mathbf{E}'} \mathcal{D}^{\mathbf{E}'} \mathcal{P}^* \mathcal{M}_G \tilde{\mathbf{E}}^{n+\frac{1}{2}} . \quad (7.56)$$

Für das elektrische und das magnetische Feld werden in dem gezeigten Ansatz die gleichen Maskierungsoperatoren verwendet, obwohl die diskretisierten Felder einen räumlichen Versatz aufweisen. Dieser Versatz

<sup>12</sup>Die Bereichsanpassung ist vor allem für Stabilitätsbetrachtungen in Abschnitt 7.3.6.3.1 relevant.

beträgt allerdings nur eine halbe Yee-Zelle. Auch wurde kein Unterschied zwischen den Komponenten eines Feldes gemacht, trotz der erfahrungsgemäß oft unterschiedlichen Anforderungen an die räumliche Auflösung aufgrund verschieden starker Feldgradienten. Bei genauer Betrachtung des Verfahrens wird ersichtlich, dass scheinbar keine zwingende Notwendigkeit zur Verwendung gleicher Maskierungsoperatoren besteht. Letztlich ist der hier gezeigte Ansatz als ein Spezialfall allgemeinerer Maskierungsmöglichkeiten zu verstehen.

### 7.3.6.3 Stabilitätsbetrachtungen

Für die klassische FDTD steht ein einfaches Kriterium zur Verfügung, welches stabile Simulationen garantiert. Dazu ist lediglich für die Zeitschrittweite die Obergrenze nach Gleichung (2.26) einzuhalten. Einige sehr nützliche Erweiterungen der FDTD neigen bei dieser Obergrenze allerdings zu instabilem Verhalten. In Abschnitt 2.2 wurde jedoch bereits erklärt, dass durch die Wahl einer noch kleineren Zeitschrittweite oft stabile Simulationen erreicht werden können.

Die vielfältigen Erweiterungsmöglichkeiten der FDTD erfordern eine recht allgemeine Herangehensweise bei der Analyse des Stabilitätsverhaltens des Detailgitter-Verfahrens. Eigenwertanalysen lassen sich nur begrenzt vornehmen, daher soll die Untersuchung auch mit Hilfe von *Normen* durchgeführt werden. Das ist möglich, da wir es ausschließlich mit linearen Räumen und Operatoren zu tun haben. Eine gute Darstellung der Theorie der Räume und Normen bietet [31].

Zunächst soll eine übersichtlichere Schreibweise für die FDTD-Gleichungen eingeführt werden:

$$\mathbf{E}^{n+\frac{1}{2}} = \mathcal{A}_1 \mathbf{E}^{n-\frac{1}{2}} + \mathcal{A}_2 \mathbf{H}^n, \quad (7.57)$$

$$\mathbf{H}^{n+1} = \mathcal{A}_3 \mathbf{H}^n + \mathcal{A}_4 \mathbf{E}^{n+\frac{1}{2}}. \quad (7.58)$$

Mit der Vereinigung der Feldvektoren zu

$$\mathbf{F}^{n+1} = \begin{pmatrix} \mathbf{E}^{n+\frac{1}{2}} \\ \mathbf{H}^{n+1} \end{pmatrix}, \quad \mathbf{F}^{n+\frac{1}{2}} = \begin{pmatrix} \mathbf{E}^{n+\frac{1}{2}} \\ \mathbf{H}^n \end{pmatrix}, \quad n \in \mathbb{N}, \quad (7.59)$$

lassen sich die Gleichungen (7.57) und (7.58) zusammenfassen zu

$$\mathbf{F}^{n+1} = \mathcal{B}_1 \mathcal{B}_2 \mathbf{F}^n \quad \text{bzw.} \quad \mathbf{F}^{n+\frac{1}{2}} = \mathcal{B}_2 \mathcal{B}_1 \mathbf{F}^{n-\frac{1}{2}} \quad (7.60)$$

mit

$$\mathcal{B}_1 = \begin{pmatrix} \text{id} & 0 \\ \mathcal{A}_4 & \mathcal{A}_3 \end{pmatrix}, \quad \mathcal{B}_2 = \begin{pmatrix} \mathcal{A}_1 & \mathcal{A}_2 \\ 0 & \text{id} \end{pmatrix}, \quad (7.61)$$

wobei 0 der Nulloperator ist (keine Abbildung). Als weitere Abkürzungen sollen

$$\mathcal{C}_1 = \mathcal{B}_1 \mathcal{B}_2 \quad \text{und} \quad \mathcal{C}_2 = \mathcal{B}_2 \mathcal{B}_1 \quad (7.62)$$

dienen. Die WT erfolgt nach bekanntem Schema, d. h.

$$\tilde{\mathbf{F}}^{n+1} = \mathcal{P}\mathbf{F}^{n+1} = \begin{pmatrix} \mathcal{P}\mathbf{E}^{n+\frac{1}{2}} \\ \mathcal{P}\mathbf{H}^{n+1} \end{pmatrix}, \quad \tilde{\mathcal{C}}_1 = \mathcal{P}(\mathcal{B}_1\mathcal{B}_2)\mathcal{P}^* = \tilde{\mathcal{B}}_1 \tilde{\mathcal{B}}_2 \quad \text{usw.} \quad (7.63)$$

Es sollen nun einige Beispiele betrachtet werden. Dabei werden sowohl Normen als auch Eigenwerte von Operatoren herangezogen, um ein plausibles Stabilitätskriterium aufstellen zu können.

### 7.3.6.3.1 Stabilitätsanalyse mit Normen

Die Simulation starte mit  $\mathbf{F}^0$ . Dann gilt nach  $n$  Zeitschritten

$$\mathbf{F}^n = \mathcal{C}_1^n \mathbf{F}^0, \quad (7.64)$$

wobei  $\mathcal{C}_1^n$  die  $n$ -fache Verkettung von  $\mathcal{C}_1$  mit sich selbst ist. Bei einer instabilen Simulation würde der Betrag der Feldkoeffizienten für  $n \rightarrow \infty$  ins Unendliche steigen. Die Stetigkeit der Norm  $\|\mathbf{F}^n\|$  führt dann dazu, dass auch  $\|\mathbf{F}^n\| \rightarrow \infty$  gilt.

Für zwei Operatoren  $\mathcal{A}$ ,  $\mathcal{B}$  und ein Element  $f$  gilt bei normierten Räumen

$$\|\mathcal{A}f\| \leq \|\mathcal{A}\| \|f\|, \quad (7.65)$$

$$\|\mathcal{A}\mathcal{B}\| \leq \|\mathcal{A}\| \|\mathcal{B}\|. \quad (7.66)$$

Daraus folgt dann

$$\|\mathbf{F}^n\| = \|\mathcal{C}_1^n \mathbf{F}^0\| \leq \|\mathcal{C}_1^n\| \|\mathbf{F}^0\| \quad (7.67)$$

sowie

$$\|\mathcal{C}_1^n\| \leq \|\mathcal{C}_1\|^n. \quad (7.68)$$

Daher gilt für instabile Simulationen  $\|\mathcal{C}_1\| > 1$ . Die Ursache einer Instabilität kann anschaulich auch als Verstärkungsprozess innerhalb des Verfahrens gedeutet werden.

Stabile Simulationen ohne elektrische oder magnetische Verluste (und ohne Abstrahlung) sind dadurch gekennzeichnet, dass die Feldenergie erhalten bleibt, was u. a.  $0 < \|\mathbf{F}^n\| \forall n$  und damit  $\|\mathcal{C}_1\| \geq 1$  bedeutet. Aus der ebenfalls gültigen Bedingung  $\|\mathbf{F}^n\| < \infty \forall n$  lässt sich mit Hilfe des Satzes von Banach-Steinhaus<sup>13</sup> eine weitere Einschränkung gewinnen. Dazu wird zunächst das Symbol  $\mathbf{L}(\mathbf{X}, \mathbf{Y})$  definiert:

**Definition 11.** Es seien  $\mathbf{X}$  und  $\mathbf{Y}$  normierte Räume. Dann bezeichnet  $\mathbf{L}(\mathbf{X}, \mathbf{Y})$  die Menge aller linearen beschränkten Operatoren, die  $\mathbf{X}$  in  $\mathbf{Y}$  abbilden.

**Satz 5.** (Banach-Steinhaus).  $\mathbf{X}$  sei ein Banachraum,  $\mathbf{Y}$  ein normierter Raum und  $(\mathcal{T}_\nu)_{\nu \in \mathbb{N}}$  eine Folge von Operatoren mit  $\mathcal{T}_\nu \in \mathbf{L}(\mathbf{X}, \mathbf{Y})$ . Für jeden Operator  $\mathcal{T}_\nu$  gelte

$$\sup_{\nu} \{\|\mathcal{T}_\nu f\|\} < \infty \quad (7.69)$$

für jedes feste  $f \in \mathbf{X}$ . Dann gibt es eine Zahl  $A > 0$ , sodass gilt

$$\|\mathcal{T}_\nu\| \leq A \quad \forall \nu \in \mathbb{N}. \quad (7.70)$$

Setzen wir  $\mathcal{T}_n = \mathcal{C}_1^n$ , so folgt dass sich über alle Zeitschritte betrachtet eine gemeinsame obere (endliche) Schranke der Norm der Operatoren  $\mathcal{C}_1^n$  angeben lässt, die mit  $A_{\mathcal{C}_1}$  bezeichnet werden soll. Insbesondere ist dann auch  $\|\mathcal{C}_1\| \leq A_{\mathcal{C}_1}$ .

In einer stabilen Simulation mit Verlusten bzw. Abstrahlung verschwindet das Feld mit zunehmender Zeit, d. h.  $\|\mathbf{F}^n\| \rightarrow 0$  für  $n \rightarrow \infty$ . Damit kann man zwar auf  $\lim_{n \rightarrow \infty} \|\mathcal{C}_1^n\| = 0 (< 1)$  schließen, aber für  $\mathcal{C}_1$  lässt sich nur  $0 < \|\mathcal{C}_1\| < A_{\mathcal{C}_1}$  angeben, wobei sich  $A_{\mathcal{C}_1}$  wieder aus Satz 5 ergibt.

Aus Gleichung (7.68) folgt, dass im Falle von  $\|\mathcal{C}_1\| \leq 1$  ein Operator für eine stabile Simulation vorliegt. Umgekehrt kann aber nicht geschlossen

<sup>13</sup>Für unsere Zwecke genügt die einfachere Version des Satzes aus [32]. Die allgemeinere Darstellung findet sich in [33].

werden, dass jeder Operator mit  $\|\mathcal{C}_1\| > 1$  instabile Simulationen hervorruft. Bei der noch folgenden Betrachtung der Eigenwerte von FDTD-Operatoren wird jedoch deutlich werden, dass eine Unterscheidung dieser beiden Fälle durchaus Sinn macht. Daher soll in der weiteren Diskussion die Bedingung  $\|\mathcal{C}_1\| \leq 1$  als *starkes Stabilitätskriterium* betrachtet werden, wogegen der Satz von Banach-Steinhaus eher ein *schwaches Stabilitätskriterium* darstellt.

Die Zeitschrittweite kann stetig von z. B.  $\Delta t = 2\Delta t_{\text{FDTD}}$  bis  $\Delta t \approx 0$  herabgesetzt werden. Da alle Operatoren ebenfalls stetig sind und diese Eigenschaft auch auf Normen zutrifft, sollte  $\|\mathcal{C}_1\|$  beim Übergang von instabilen zu stabilen Simulationsoperatoren in gleichfalls stetiger Weise eine Stabilitätsgrenze, die hier mit  $\|\mathcal{C}_1\| = 1$  angesetzt wird, erreichen oder sogar unterschreiten. Es ist somit nahe liegend, dass auch  $\|\mathcal{C}_1\| < 1$  oder zumindest  $\|\mathcal{C}_1\| = 1$  gilt bei stabilen Simulationen mit Verlusten.<sup>14</sup> Besonders deutlich werden diese Verhältnisse, wenn sich die Norm aus der Energieerhaltung ableiten lässt.<sup>15</sup> Instabile Simulationen bedeuten eine Energieverstärkung und damit einen Wert der Operatornorm oberhalb von Eins.

In gleicher Weise lassen sich diese Argumente für  $\mathcal{C}_2$  und die Wavelet-transformierten Operatoren  $\tilde{\mathcal{C}}_1$  und  $\tilde{\mathcal{C}}_2$  vortragen. Insbesondere gilt

$$\tilde{\mathcal{C}}_1^n = (\mathcal{P} \mathcal{C}_1 \mathcal{P}^*)^n = \mathcal{P} \mathcal{C}_1^n \mathcal{P}^* \quad (7.71)$$

und damit

$$\|\tilde{\mathcal{C}}_1^n\| \leq \|\mathcal{P}\| \|\mathcal{C}_1^n\| \|\mathcal{P}^*\| \leq \|\mathcal{P}\| A_{\mathcal{C}_1} \|\mathcal{P}^*\| . \quad (7.72)$$

Des Weiteren ist  $\|\mathcal{P}\| = \|\mathcal{P}^*\| = 1$ , wenn für den physikalischen Raum (Grundraum) und den Waveletbereich die gleiche Normdefinition verwendet werden.<sup>16</sup> Hierzu wird der Banachraum um das Skalarprodukt  $\langle f, g \rangle$

<sup>14</sup>In einem elektrisch und magnetisch vollständig verlustbehafteten Raum wäre dann  $\|\mathcal{C}_1\| < 1$ . Ist allerdings wenigstens ein Teil des Raumes verlustfrei, dürfte aufgrund der allgemeinen Definition von Operatornormen, die mittels eines Suprenums eine Abschätzung zu höheren Werten erbringt, wiederum  $\|\mathcal{C}_1\| = 1$  gelten.

<sup>15</sup>Die sog. *Quadratnorm* bzw. *euklidische Norm* beispielsweise lässt sich als ein Spezialfall einer Energienorm betrachten, vgl. [31].

<sup>16</sup>Für einen bijektiven Operator  $\mathcal{T}$  und seine Inverse  $\mathcal{T}^{-1}$  gilt i. A. weder  $\|\mathcal{T}\| = \|\mathcal{T}^{-1}\| = 1$  noch  $\|\mathcal{T}^{-1}\| = \|\mathcal{T}\|^{-1}$ . Solche günstigen Beziehungen treten nur unter bestimmten Voraussetzungen auf bzw. sind für jeden Operator(-Typ) gesondert nachzuweisen.

erweitert, damit die weiteren Betrachtungen mit Hilfe der Gesetze von Hilberträumen durchgeführt werden können.<sup>17</sup>

Sowohl  $\mathcal{P}$  als auch  $\mathcal{P}^*$  haben die Eigenschaft, Orthonormalbasen aufeinander abzubilden. Ist  $f$  bzw.  $\tilde{f}$  ein Element der ONB des Grundraumes bzw. des Waveletbereichs, gilt daher  $\|f\| = \|\mathcal{P}f\| = 1$  bzw.  $\|\tilde{f}\| = \|\mathcal{P}^*\tilde{f}\| = 1$ . Zwei Elemente  $f, g$  sind orthogonal zueinander, wenn  $\langle f, g \rangle = 0$  gilt. Für die Norm soll dann  $\|f + g\| = \|f\| + \|g\|$  gelten.<sup>18</sup>

Jedes beliebige Element der Räume lässt sich durch eine Linearkombination aus der jeweiligen ONB bilden. Zusammen mit der Orthogonalität der Basenelemente und der Normerhaltung bei der WT bzw. ihrer Umkehrung folgt die Behauptung  $\|\mathcal{P}\| = \|\mathcal{P}^*\| = 1$ . Damit gilt nach Gleichung (7.72), dass  $\|\tilde{\mathcal{C}}_1^n\| \leq \|\mathcal{C}_1^n\|$  ist. Analog kann man

$$\mathcal{C}_1^n = (\mathcal{P}^* \tilde{\mathcal{C}}_1 \mathcal{P})^n = \mathcal{P}^* \tilde{\mathcal{C}}_1^n \mathcal{P} \quad (7.73)$$

angeben und damit auf  $\|\mathcal{C}_1^n\| \leq \|\tilde{\mathcal{C}}_1^n\|$  schließen. Beide Ungleichungen zusammen ergeben dann

$$\|\tilde{\mathcal{C}}_1^n\| = \|\mathcal{C}_1^n\| . \quad (7.74)$$

Als nächstes wird der Begriff des *Projektionsoperators* benötigt [31]:

**Definition 12.** Es sei  $\mathbf{X}$  ein Hilbertraum und  $\mathbf{X}'$  ein abgeschlossener Unterraum von  $\mathbf{X}$ . Dann lässt sich jedes  $f \in \mathbf{X}$  eindeutig in der Form  $f = f' + f''$  mit  $f' \in \mathbf{X}'$  und  $f'' \in \mathbf{X}''$  darstellen, wobei  $\mathbf{X}''$  das eindeutig bestimmte orthogonale Komplement von  $\mathbf{X}'$  ist. Die Abbildung  $\mathcal{L} : \mathbf{X} \mapsto \mathbf{X}'$  mit

$$\mathcal{L}f = f' , \quad f \in \mathbf{X} , \quad f' \in \mathbf{X}' \quad (7.75)$$

heißt *Projektionsoperator*.

Über die Eigenschaften von  $\mathcal{L}$  lässt sich Folgendes sagen:

**Satz 6.** Der durch Gleichung (7.75) erklärte Projektionsoperator  $\mathcal{L}$  ist eine lineare und beschränkte Abbildung von  $\mathbf{X}$  auf  $\mathbf{X}'$  mit  $\|\mathcal{L}\| = 1$ .

<sup>17</sup>Das Skalarprodukt  $\langle f, g \rangle_{\mathbf{L}_2}$ , wurde bereits in Abschnitt 3.2.2 benutzt. Aus Gründen der Übersichtlichkeit wird der Index  $\mathbf{L}_2$  weggelassen.

<sup>18</sup>Das ist beispielsweise dann gegeben, wenn die Norm durch  $\|f\| \stackrel{\text{def}}{=} \sqrt{\langle f, f \rangle}$  definiert wird [33].

Der Maskierungsoperator  $\mathcal{M}_G$  aus Abschnitt 7.3.6 ist ein Projektionsoperator: Durch  $\mathcal{M}_G$  werden Entwicklungskoeffizienten entweder beibehalten oder zu Null gesetzt. Ferner gehört jeder Entwicklungskoeffizient zu einem Element (einer Funktion) der ONB des Waveletbereichs. Mit

$$\mathcal{M}_G^{\tilde{\mathbf{F}}} = \begin{pmatrix} \mathcal{M}_G & 0 \\ 0 & \mathcal{M}_G \end{pmatrix} \quad (7.76)$$

wird der zugehörige Maskierungsoperator bzw. Projektionsoperator für  $\tilde{\mathbf{F}}^{n+1}$  und  $\tilde{\mathbf{F}}^{n+\frac{1}{2}}$  gebildet. Dadurch kann ein approximierter FDTD-Operator für den Waveletbereich formuliert werden:

$$\tilde{\mathcal{C}}_1' = \mathcal{M}_G^{\tilde{\mathbf{F}}} \tilde{\mathcal{C}}_1 = \mathcal{M}_G^{\tilde{\mathbf{F}}} \tilde{\mathcal{B}}_1 \tilde{\mathcal{B}}_2 . \quad (7.77)$$

Der Projektionsoperator gewährleistet, dass die Norm von  $\tilde{\mathcal{C}}_1$  durch die Maskierung nicht erhöht wird, d. h. es gilt

$$\|\tilde{\mathcal{C}}_1'\| = \|\mathcal{M}_G^{\tilde{\mathbf{F}}} \tilde{\mathcal{C}}_1\| \leq \|\tilde{\mathcal{C}}_1\| = \|\mathcal{C}_1\| . \quad (7.78)$$

Ist  $\|\mathcal{C}_1\| \leq 1$ , so folgt, dass auch eine Simulation mit  $\tilde{\mathcal{C}}_1'$  stabil sein wird. Das gleiche gilt auch für eine Approximation der Art  $\tilde{\mathcal{C}}_1'' = \tilde{\mathcal{C}}_1 \mathcal{M}_G^{\tilde{\mathbf{F}}}$  oder  $\tilde{\mathcal{C}}_1''' = \mathcal{M}_G^{\tilde{\mathbf{F}}} \tilde{\mathcal{C}}_1 \mathcal{M}_G^{\tilde{\mathbf{F}}}$ . Analoge Betrachtungen gelten für  $\mathcal{C}_2$ ,  $\tilde{\mathcal{C}}_2'$  usw.

Gleichung (7.77) kann als vereinfachte bzw. reduzierte Approximation aufgefasst werden. Für diese Art der Approximation ist nun geklärt, unter welchen Voraussetzungen sich kein negativer Einfluss auf die Stabilität einer Simulation ergibt. Es bleibt zu untersuchen, ob auch die Approximationen in den Gleichungen (7.55) und (7.56) die Stabilität wahren. Die Vermutung liegt nahe, denn definiert man einen neuen Operator

$$\tilde{\mathcal{C}}_3' = \tilde{\mathcal{B}}_2 \mathcal{M}_G^{\tilde{\mathbf{F}}} \tilde{\mathcal{B}}_1 , \quad (7.79)$$

so lässt sich die zeitliche Entwicklung des approximierten FDTD-Operators  $\tilde{\mathcal{C}}_1^m$  darstellen durch

$$\tilde{\mathcal{C}}_1^m = \tilde{\mathcal{B}}_2^{-1} \tilde{\mathcal{C}}_3^m \tilde{\mathcal{B}}_2 \quad (7.80)$$

(sofern  $\tilde{\mathcal{B}}_2^{-1}$  existiert; Näheres hierzu folgt auf Seite 130) oder

$$\tilde{\mathcal{C}}_1^m = \mathcal{M}_G^{\tilde{\mathbf{F}}} \tilde{\mathcal{B}}_1 \tilde{\mathcal{C}}_3^{m-1} \tilde{\mathcal{B}}_2 . \quad (7.81)$$

Kann die Stabilität von  $\tilde{\mathcal{C}}'_3$  gezeigt werden, so wäre eine stabile Simulation mit der Approximation nach den Gleichungen (7.55) und (7.56) äquivalent zu einer beschränkten Norm von  $(\mathcal{M}_G^{\mathbb{F}} \tilde{\mathcal{C}}'_3 \mathcal{M}_G^{\mathbb{F}})^n = (\mathcal{M}_G^{\mathbb{F}} \tilde{\mathcal{B}}_1 \mathcal{M}_G^{\mathbb{F}} \tilde{\mathcal{B}}_2 \mathcal{M}_G^{\mathbb{F}})^n$  für jeden Zeitschritt  $n$ .

Mit  $\tilde{\mathcal{C}}'_1$  sind nach dem oben Gesagten stabile Simulationen möglich, daher muss nach Satz 5 auch  $\tilde{\mathcal{C}}'_3$  in Gleichung (7.80) stets beschränkt sein, was dem schwachen Stabilitätskriterium entspricht. Im Gegensatz zu  $\tilde{\mathcal{C}}'_1$  kann für die Norm von  $\tilde{\mathcal{C}}'_3$  jedoch nicht unmittelbar geschlossen werden, dass sie ebenfalls kleiner oder gleich Eins ist. Zumindest aus Gleichung (7.79) ist ein solcher Zusammenhang nicht herleitbar.<sup>19</sup> Mit Hilfe der Theorien sog. *dualer Operatoren* und *adjungierter Operatoren* kann jedoch gezeigt werden, dass auch  $\tilde{\mathcal{C}}'_3$  das starke Stabilitätskriterium erfüllt. Aus Gründen des Umfangs sollen hier nur einige wenige Aussagen dieser Theorien wiedergegeben werden. Weiterführende Grundlagen zu dem Thema sind in [33] zu finden.

Es sei ein Hilbertraum  $\mathbf{X}$  gegeben und  $\mathcal{T} : \mathbf{X} \mapsto \mathbf{X}$  ein linearer Operator. Zu  $\mathbf{X}$  existiert dann ein *dualer Raum*  $\mathbf{X}^T$  und zu  $\mathcal{T}$  ein eindeutiger dualer Operator  $\mathcal{T}^T : \mathbf{X}^T \mapsto \mathbf{X}^T$ . Ist  $\mathcal{T}$  linear und stetig, dann ist auch  $\mathcal{T}^T$  linear und stetig und es gilt  $\|\mathcal{T}\| = \|\mathcal{T}^T\|$ .

Das zu einem Hilbertraum gehörende Skalarprodukt erlaubt auch die eindeutige Angabe eines zu  $\mathcal{T}$  adjungierten Operators  $\mathcal{T}^\dagger$ . Die Eigenschaften Linearität und Stetigkeit übertragen sich von  $\mathcal{T}$  auch auf  $\mathcal{T}^\dagger$ . Ferner gilt dann  $\|\mathcal{T}\| = \|\mathcal{T}^\dagger\|$ .

Für einen linearen Operator  $\mathcal{T} : D(\mathcal{T}) \mapsto \mathbf{X}$ ,  $D(\mathcal{T}) \subseteq \mathbf{X}$ , auf einem Hilbertraum  $\mathbf{X}$  über  $\mathbb{C}$  (oder  $\mathbb{R}$ ) und dichtem Definitionsbereich  $D(\mathcal{T})$  gilt

$$\mathcal{T}^\dagger = \mathcal{E}^{-1} \mathcal{T}^T \mathcal{E} , \quad (7.82)$$

wobei  $\mathcal{E} : \mathbf{X} \mapsto \mathbf{X}^T$  die Dualitätsabbildung bezeichnet.<sup>20</sup>

<sup>19</sup>Genauer gesagt wären die Voraussetzungen an  $\mathcal{B}_1, \mathcal{B}_2$  vermutlich zu einschränkend als dass man diese wenigstens anschaulich begründen bzw. rechtfertigen könnte. Vgl. auch hierzu Fußnote 16 auf Seite 126.

<sup>20</sup>Beim Nachlesen in [33] ist zu beachten, dass dort  $\mathbf{X}^\dagger \equiv \mathbf{X}^T$  gilt.  $\mathbf{X}^\dagger$  bezeichnet in [33] nicht den Definitionsbereich von  $\mathcal{T}^\dagger$ , denn dieser ist ja nach dem oben Gesagten  $\mathbf{X}$ , also der gleiche wie von  $\mathcal{T}$ .

Identifiziert man nun  $\tilde{\mathcal{C}}'_1 = \mathcal{T}^\dagger$ ,  $\tilde{\mathcal{B}}_2 = \mathcal{E}$  und  $\tilde{\mathcal{C}}'_3 = \mathcal{T}^\top$ , so lassen sich die Gleichungen (7.80) und (7.82) für  $n = 1$  ineinander überführen. Das ist allerdings nur dann zulässig, wenn der Teiloperator  $\tilde{\mathcal{B}}_2$  bzw.  $\mathcal{B}_2$  invertierbar ist. Das bedeutet, dass man auch eine Simulation „rückwärts“ rechnen kann, d. h. von dem aktuellen Zeitpunkt ausgehend müssen die Feldwerte der vorangegangenen Zeitschritte berechenbar sein.

Setzen wir zunächst die Existenz der Inversen von  $\mathcal{B}_2$  voraus, so sind  $\tilde{\mathcal{C}}'_1$  und  $\tilde{\mathcal{C}}'_3$  der duale und der adjungierte Operator zu  $\mathcal{T}$ . Demnach gilt dann

$$\|\tilde{\mathcal{C}}'_1\| = \|\tilde{\mathcal{C}}'_3\| = \|\mathcal{T}\| . \quad (7.83)$$

Dabei ist es unerheblich, wie die Abbildungseigenschaften von  $\mathcal{T}$  im Detail beschaffen sind. Relevant ist an dieser Stelle lediglich, dass die genannten Voraussetzungen die Existenz dieses Operators sicherstellen.

Bevor nun Gleichung (7.83) für weitere Normanalysen herangezogen wird, soll die Invertierbarkeit von  $\mathcal{B}_2$  im Detail betrachtet werden. Ein Blick in die FDTD-Gleichungen (2.17) und (2.20) zeigt, dass es ohne Schwierigkeiten möglich ist, die Gleichungen nach  $H_x^n$  und  $E_x^{n-\frac{1}{2}}$  umzustellen. Unter diesem Gesichtspunkt betrachtet sind sowohl  $\mathcal{B}_1$  als auch  $\mathcal{B}_2$  bijektiv<sup>21</sup>, ebenso  $\tilde{\mathcal{B}}_1$  und  $\tilde{\mathcal{B}}_2$ .

Ein etwas anderes Bild ergibt sich aber, wenn man elektrische und magnetische Wände oder sogar feldfreie Volumina betrachtet. Nach den Ausführungen von Abschnitt 7.3.1 werden die Eigenschaften dieser Strukturen durch eine Ausmaskierung entsprechender räumlicher Bereiche bzw. Feldkoeffizienten nachgebildet. Diese Operation ist irreversibel, da sie einer Multiplikation von (Urbild-)Koeffizienten mit dem Faktor Null entspricht. So gesehen wären  $\mathcal{B}_1$  und  $\mathcal{B}_2$  nicht bijektiv. Allerdings kann man sich mit einer Reduktion des (Hilbert-)Raumes behelfen: Tangentiale elektrische Feldkomponenten auf elektrischen Wänden beispielsweise sind immer Null, d. h. sowohl vor der Anwendung eines Operators  $\mathcal{B}_2$  als auch danach. Durch die Bereichsanpassung in Abschnitt 7.3.2 gibt es keinerlei Verbindung zwischen den Feldkoeffizienten elektrischer Wände und anderen Feldkoeffizienten. Daher kann man diese Koeffizienten, deren Wert

<sup>21</sup>Diese Eigenschaft der FDTD wird manchmal dazu benutzt, um aus einem empfangenen Signal auf die (ungefähre) Form des abstrahlenden bzw. reflektierenden Objektes schließen zu können.

stets Null ist, aus dem Simulationsraum herausnehmen, was einer Reduktion des Hilbertraumes  $\mathbf{X}$  entspricht.

Die durch die Bereichsanpassung reduzierten Hilberträume werden mit  $\mathbf{X}'_1$  und  $\mathbf{X}'_2$  bezeichnet, wobei gilt

$$\mathcal{B}_1 : \mathbf{X}'_1 \mapsto \mathbf{X}'_2, \quad \mathcal{B}_2 : \mathbf{X}'_2 \mapsto \mathbf{X}'_1, \quad \mathbf{X}'_1, \mathbf{X}'_2 \subseteq \mathbf{X}. \quad (7.84)$$

Auf diese Räume bezogen sind  $\mathcal{B}_1$  und  $\mathcal{B}_2$  wieder bijektiv, sofern auch die Feldanregung bereits diesen Räumen entspricht, d. h.  $\mathbf{F}^0 \in \mathbf{X}'_2$  (siehe Gleichung (7.64)).<sup>22</sup> Die Bereichsanpassung kann durch die Maskierungsoperatoren  $\mathcal{M}_\lambda^{1,\infty}$ ,  $\mathcal{M}_\lambda^{2,\infty}$  ausgedrückt werden, welche bereits in Abschnitt 7.3.2 eingeführt wurden. Auf den zusammengefassten Feldvektor  $\mathbf{F}$  erweitert bedeutet das<sup>23</sup>

$$\mathcal{B}_1 = \mathcal{M}_\lambda^{\mathbf{F}} \mathcal{B}_1 \mathcal{M}_\lambda^{\mathbf{F}}, \quad \mathcal{B}_2 = \mathcal{M}_\lambda^{\mathbf{F}} \mathcal{B}_2 \mathcal{M}_\lambda^{\mathbf{F}} \quad (7.85)$$

mit

$$\mathcal{M}_\lambda^{\mathbf{F}} = \begin{pmatrix} \mathcal{M}_\lambda^{1,\infty} & 0 \\ 0 & \mathcal{M}_\lambda^{2,\infty} \end{pmatrix}. \quad (7.86)$$

Offensichtlich gilt daher sogar

$$\mathbf{X}'_1 = \mathbf{X}'_2 = \mathbf{X}' = \mathcal{M}_\lambda^{\mathbf{F}} \mathbf{X}. \quad (7.87)$$

Durch die WT ergibt sich der Raum  $\tilde{\mathbf{X}}' = \mathcal{P} \mathcal{M}_\lambda^{\mathbf{F}} \mathbf{X}$ , der eine Teilmenge von  $\tilde{\mathbf{X}} = \mathcal{P} \mathbf{X}$  darstellt.<sup>24</sup> Während die Maskierung im Urbildraum  $\mathbf{X}$  einfach Feldkoeffizienten zu Null setzt, stellt sich die Natur des reduzierten Raumes im Waveletbereich anders dar. In Bezug auf die Entwicklung von Funktionen in Basen des Waveletbereiches kann es zwar ebenfalls vorkommen, dass bestimmte Entwicklungskoeffizienten identisch Null sind. Im Wesentlichen äußert sich die Reduktion von  $\tilde{\mathbf{X}}$  auf  $\tilde{\mathbf{X}}'$  jedoch darin, dass die Entwicklungskoeffizienten nicht mehr unabhängig voneinander

<sup>22</sup>Es macht beispielsweise auch wenig Sinn, zur Anregung einer Simulation einen von Null verschiedenen Wert für eine tangentielle elektrische Feldkomponente in einer elektrischen Wand zu verwenden.

<sup>23</sup>Die Identitäten in  $\mathcal{B}_1$  und  $\mathcal{B}_2$  in Gleichung (7.61) gelten wegen der Bereichsanpassung ebenfalls nur in  $\mathbf{X}'_1$ ,  $\mathbf{X}'_2$ .

<sup>24</sup>Nur im Falle von  $\mathcal{M}_\lambda^{\mathbf{F}} = \text{id}$  ist  $\tilde{\mathbf{X}}' = \mathcal{P} \mathbf{X}$ .

sind, d. h. es sind nicht mehr alle Wertekombinationen der Koeffizienten erlaubt.<sup>25</sup>

Die Detailgitter-Maskierung mit  $\mathcal{M}_G^{\tilde{\mathbf{F}}}$  maskiert dagegen einfach einzelne Entwicklungskoeffizienten in  $\tilde{\mathbf{X}}$  aus, ganz so, wie es  $\mathcal{M}_\lambda^{\mathbf{F}}$  im Grundraum  $\mathbf{X}$  macht. Etwas umgangssprachlich formuliert „weiß“  $\mathcal{M}_G^{\tilde{\mathbf{F}}}$  an dieser Stelle nichts davon, dass bestimmte Wertekombinationen in  $\tilde{\mathbf{X}}'$  nicht vorkommen. Dadurch kann es passieren, dass das Ergebnis von  $\mathcal{M}_G^{\tilde{\mathbf{F}}}\tilde{\mathbf{F}}$  nicht mehr zu  $\tilde{\mathbf{X}}'$  gehört, d. h. es gilt im Allgemeinen  $\mathcal{M}_G^{\tilde{\mathbf{F}}}\tilde{\mathbf{X}}' \neq \tilde{\mathbf{X}}'$  oder noch genauer

$$\Delta\tilde{\mathbf{X}}'_G \stackrel{\text{def}}{=} (\mathcal{M}_G^{\tilde{\mathbf{F}}}\tilde{\mathbf{X}}') \setminus \tilde{\mathbf{X}}' \neq \emptyset. \quad (7.88)$$

Dieser Aspekt ist deshalb relevant, weil die mit Gleichung (7.82) verbundenen Normbeziehungen in Gleichung (7.83) natürlich nur für die Räume gelten, in denen die Operatoren definiert sind.  $\Delta\tilde{\mathbf{X}}'_G$  gehört nicht zum Definitionsbereich von  $\tilde{\mathcal{B}}_2$ , jedoch der gemeinsame Raum

$$\tilde{\mathbf{X}}'' \stackrel{\text{def}}{=} (\mathcal{M}_G^{\tilde{\mathbf{F}}}\tilde{\mathbf{X}}') \cap \tilde{\mathbf{X}}'. \quad (7.89)$$

Erinnern wir uns noch einmal an den Aufbau von  $\mathcal{B}_2$ : Sowohl der Bild- als auch der Urbildbereich werden automatisch maskiert, d. h. bestimmte „unpassende“ Feldverteilungen werden von weiteren Berechnungen ausgeschlossen. Diese Eigenschaft überträgt sich in entsprechender Weise auch auf  $\tilde{\mathcal{B}}_2$ . Zusammen mit der Detailgitter-Maskierung bedeutet das, dass die Simulation im Waveletbereich im (Unter-)Raum  $\tilde{\mathbf{X}}''$  stattfindet, welcher eine Teilmenge von  $\tilde{\mathbf{X}}'$  ist. Statt des Detailgitter-Operators  $\mathcal{M}_G^{\tilde{\mathbf{F}}}$  kann deshalb in den Gleichungen (7.77) bis (7.81) auch der Operator

$$\mathcal{M}_G^{\tilde{\mathbf{F}}'} = (\mathcal{P} \mathcal{M}_\lambda^{\mathbf{F}} \mathcal{P}^*) \mathcal{M}_G^{\tilde{\mathbf{F}}} (\mathcal{P} \mathcal{M}_\lambda^{\mathbf{F}} \mathcal{P}^*) \quad (7.90)$$

eingesetzt werden. Auf diese Weise wird ersichtlich, dass das Urbild von  $\tilde{\mathcal{B}}_2$  in Gleichung (7.79) (d. h. das Bild von  $\mathcal{M}_G^{\tilde{\mathbf{F}}}\tilde{\mathcal{B}}_1$  bzw.  $\mathcal{M}_G^{\tilde{\mathbf{F}}'}\tilde{\mathcal{B}}_1$ ) stets Element von  $\tilde{\mathbf{X}}''$  ist. Das Bild von  $\tilde{\mathcal{B}}_2$  ist dann Element von  $\tilde{\mathbf{X}}'$ , was der Definitionsbereich von  $\tilde{\mathcal{B}}_2^{-1}$  ist.

Unter der Beschränkung auf die reduzierten Räume bleibt  $\tilde{\mathcal{B}}_2$  also umkehrbar, sodass diese Voraussetzung von Gleichung (7.82) erfüllt ist. Eine

---

<sup>25</sup>Dennoch ist  $\tilde{\mathbf{X}}'$  linear und separabel, d. h. es lässt sich eine Basis in diesem Raum angeben. Diese Basisfunktionen setzen sich im Allgemeinen aus mehreren Funktionen der ONB des ursprünglichen Waveletbereiches zusammen.

weitere Bedingung lautete, dass der Definitionsbereich des Operators  $\mathcal{T}$  dicht sein muss. Auch das wird durch die reduzierten Räume gewährleistet. Somit kann Gleichung (7.83) als gültig angesehen werden.

Die Normanalysen sind damit fast beendet. Aus Gleichung (7.83) ist ersichtlich, dass  $\tilde{\mathcal{C}}'_3$  das starke Stabilitätskriterium erfüllt:

$$\|\tilde{\mathcal{C}}'_3\| = \|\tilde{\mathcal{C}}'_1\| \leq 1 . \quad (7.91)$$

Analog wie für  $\tilde{\mathcal{C}}_1$  lässt sich damit zeigen, dass dann auch die Normen von  $\mathcal{M}_G^{\tilde{\mathcal{F}}}\tilde{\mathcal{C}}'_3$ ,  $\tilde{\mathcal{C}}'_3\mathcal{M}_G^{\tilde{\mathcal{F}}}$  und  $\mathcal{M}_G^{\tilde{\mathcal{F}}}\tilde{\mathcal{C}}'_3\mathcal{M}_G^{\tilde{\mathcal{F}}}$  (und ebenso mit  $\mathcal{M}_G^{\tilde{\mathcal{F}}}$ ) den Wert Eins nicht überschreiten. Hiermit folgt endlich die gesuchte Ungleichung

$$\|\mathcal{M}_G^{\tilde{\mathcal{F}}}\tilde{\mathcal{B}}_1\mathcal{M}_G^{\tilde{\mathcal{F}}}\tilde{\mathcal{B}}_2\mathcal{M}_G^{\tilde{\mathcal{F}}}\| \leq 1 , \quad (7.92)$$

welche stabile Simulationen mit der Approximation nach den Gleichungen (7.55) und (7.56) gewährleistet, sofern die in diesem Abschnitt genannten Voraussetzungen erfüllt sind. In gleicher Weise gelten die gezeigten Ableitungen und Schlussfolgerungen auch für den Fall, dass von  $\mathcal{C}_2$  statt von  $\mathcal{C}_1$  ausgegangen wird.

### 7.3.6.3.2 Eigenwerte und Eigenvektoren

Für weitere Betrachtungen sollen jetzt Eigenwerte herangezogen werden. Dazu werden einige Grundlagen benötigt, die vor allem [33] entnommen wurden und hier etwas vereinfacht wiedergegeben werden sollen.

Es sei  $\mathbf{X}$  ein nichtleerer komplexer Banachraum und  $\mathcal{T} : \mathbf{X} \mapsto \mathbf{X}$  ein linearer und abgeschlossener Operator mit dichtem Definitionsbereich. Zu  $\mathcal{T}$  lässt sich dann eine *Resolventenmenge*  $R_{\mathcal{T}}$  und ein *Spektrum*  $S_{\mathcal{T}}$  angeben, die zueinander komplementär sind und zusammen den Raum der komplexen Zahlen bilden ( $R_{\mathcal{T}} \cap S_{\mathcal{T}} = \emptyset$ ,  $R_{\mathcal{T}} \cup S_{\mathcal{T}} = \mathbb{C}$ ). Das Spektrum  $S_{\mathcal{T}}$  besteht unter anderem aus den komplexen Werten  $\lambda$ , für die eine Lösung  $f \neq 0$  der Gleichung

$$\mathcal{T}f = \lambda f , \quad f \in \mathbf{X} , \quad \lambda \in \mathbb{C} \quad (7.93)$$

existiert.<sup>26</sup> Die komplexen Werte  $\lambda$  werden *Eigenwerte* genannt, was die Verwandtschaft zu den Eigenwerten analoger Matrixgleichungen andeutet, auf die wir noch zu sprechen kommen. Zuvor soll jedoch eine wichtige

<sup>26</sup>Im Allgemeinen besteht  $S_{\mathcal{T}}$  nicht nur aus dem *Punktspektrum*, welches aus den komplexen Werten  $\lambda$  gebildet wird, sondern auch noch aus einem *Stetigkeitsspektrum* und einem *Restspektrum*, siehe [32].

Eigenschaft des Spektrums betrachtet werden. Der Spektralradius von  $S_{\mathcal{T}}$  ist definiert zu

$$r_{\mathcal{T}} \stackrel{\text{def}}{=} \sup_{\lambda \in S_{\mathcal{T}}} \{|\lambda|\} . \quad (7.94)$$

Es gilt nun der folgende Satz:

**Satz 7.** *Bei einem linearen stetigen Operator  $\mathcal{T} : \mathbf{X} \mapsto \mathbf{X}$  ist  $S_{\mathcal{T}}$  nicht leer und kompakt. Für den Spektralradius  $r_{\mathcal{T}}$  gilt*

$$r_{\mathcal{T}} = \lim_{n \rightarrow \infty} \|\mathcal{T}^n\|^{\frac{1}{n}} \leq \|\mathcal{T}\| . \quad (7.95)$$

Die Ungleichung  $\lim_{n \rightarrow \infty} \|\mathcal{T}^n\|^{\frac{1}{n}} \leq \|\mathcal{T}\|$  ist nicht weiter überraschend, denn sie folgt bereits aus Gleichung (7.66). Interessanter ist der Zusammenhang mit dem Spektralradius, wie gleich zu sehen ist.

Es kann jetzt der Übergang zu den Simulationsoperatoren eingeleitet werden, denn es gilt, dass das Spektrum eines linearen Operators  $\mathcal{T} : \mathbf{X} \mapsto \mathbf{X}$  mit  $\dim \mathbf{X} < \infty$  aus genau allen Eigenwerten besteht. Numerische Simulationen können nur endlich viele Elemente beinhalten, daher ist  $\dim \mathbf{X} < \infty$  erfüllt. Ferner können die linearen Operationen durch Matrizen ausgedrückt werden, deren (abzählbare) Eigenwerte mit denen des Operators identisch sind. Mit diesen Matrizen sind nicht die Operator- bzw. Untermatrizen wie aus Abbildung 5.1 auf Seite 52 gemeint, sondern die Gesamtmatrizen, welche die Abbildung sämtlicher Koeffizienten eines  $L$ -dimensionalen Raumes in einem Schritt bewirken.

Bereits bei den ersten publizierten Untersuchungen des Verhaltens der FDTD mit Hilfe einer Eigenwertanalyse ergab sich als notwendige Bedingung für die Stabilität, dass für die Eigenwerte  $\lambda_{\nu}$  des Operators  $\mathcal{C}_1$

$$|\lambda_{\nu}| \leq 1 \quad \forall \nu \quad (7.96)$$

gelten muss, was nach Gleichung (7.94) zur Ungleichung  $r_{\mathcal{C}_1} \leq 1$  führt.<sup>27</sup> Laut Satz 7 kann für einen stabilen Simulationsoperator im Prinzip auch

---

<sup>27</sup>Die erwähnten Untersuchungen wurden für den Fall eines unendlich ausgedehnten, homogenen und äquidistant diskretisierten Raumes durchgeführt. In der Praxis überträgt man dann das so gewonnene Stabilitätskriterium – die maximale Zeitschrittweite – auf die „ungünstigsten“ Stellen in dem zu simulierenden (meist inhomogenen) Raum.

dann  $\|\mathcal{C}_1\| > 1$  sein, wenn  $r_{\mathcal{C}_1} = 1$  ist. Gilt jedoch  $\|\mathcal{C}_1\| \leq 1$ , ist automatisch auch die Bedingung  $r_{\mathcal{C}_1} \leq 1$  erfüllt.

Für die weiteren Betrachtungen gehen wir jetzt zur Darstellung aller Operatoren durch ihre (Gesamt-)Matrizen über. Zunächst soll die Bedeutung der Eigenvektoren geklärt werden. Als Beispiel dient die einfache Dreiecksmatrix

$$\mathbf{T} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}. \quad (7.97)$$

Wie sich leicht nachprüfen lässt, hat  $\mathbf{T}$  den doppelten Eigenwert  $\lambda_1 = \lambda_2 = 1$  und lediglich einen Eigenvektor  $\mathbf{v}_1 = (1, 0)^T$ . Die Quadratnorm lautet bei zweidimensionalen Vektoren

$$\|\mathbf{a}\| = \|(a_1, a_2)^T\| = \sqrt{|a_1|^2 + |a_2|^2}. \quad (7.98)$$

Damit ist dann auch  $\|\mathbf{v}_1\| = \|\mathbf{T}\mathbf{v}_1\| = 1$ . Der orthogonale Vektor  $\mathbf{b} = (0, 1)^T$  hat ebenfalls die Norm Eins, aber es ist  $\|\mathbf{T}\mathbf{b}\| = \sqrt{2} > 1$ . Folglich ist  $\|\mathbf{T}\| > 1$ , also größer als die Eigenwerte und der resultierende Spektralradius  $r_{\mathcal{T}} = 1$  hoffen lassen.<sup>28</sup>

Das geschilderte Problem hängt anscheinend damit zusammen, dass kein vollständiges System von Eigenvektoren vorlag, d. h. es gab weniger Eigenvektoren als Eigenwerte,<sup>29</sup> wobei letztere entsprechend ihrer Vielfachheit zu zählen sind. Im Allgemeinen ist die Analyse des FDTD-Operators einer zu simulierenden Struktur auf mehrfache Eigenwerte aufgrund der algorithmischen Komplexität bekannter Verfahren sowie numerischer Ungenauigkeiten nicht praktikabel. Eine Notwendigkeit zu solch einer Analyse stellt sich häufig auch nicht, denn in [34] wird erwähnt, dass Matrizen mit doppelten Eigenwerten in der Praxis kaum auftreten. Demnach scheint es für die Matrizen praxisnaher Probleme meistens ein vollständiges System von Eigenvektoren zu geben. Selbst wenn Eigenwerte doppelt oder noch häufiger sein sollten, ist das System der Eigenvektoren also nicht notwendigerweise unvollständig. Vor allem die bei der

<sup>28</sup>Die Norm von  $\mathbf{T}$  ist sogar größer als  $\sqrt{2}$ . Eine genaue Berechnung ergibt  $\|\mathbf{T}\| \approx 1,618034$ . Bei der Verwendung anderer Normdefinitionen sind davon abweichende Werte der Norm von  $\mathbf{T}$  möglich.

<sup>29</sup>Man sagt auch, dass die algebraischen Vielfachheiten der Eigenwerte verschieden sind von ihren geometrischen Vielfachheiten.

klassischen FDTD jahrzehntelange weltweite Erfahrung mit stabilen Simulationen legt nahe, dass man i. d. R. wohl von einem der beiden Fälle ausgehen kann:

1. Das System der Eigenvektoren ist vollständig.
2. Das System der Eigenvektoren ist unvollständig, aber es gilt trotzdem für alle Vektoren  $\mathbf{a}$ , dass  $\|\mathbf{C}_1\mathbf{a}\| \leq \|\mathbf{a}\|$  ist, was  $\|\mathbf{C}_1\| \leq 1$  bedeutet.

$\mathbf{C}_1$  ist hierbei die Matrix des Operators  $\mathcal{C}_1$ . Punkt 2 wurde bereits in Abschnitt 7.3.6.3.1 behandelt, sodass wir uns hier auf Punkt 1 konzentrieren.

Zusätzlich zu der Vollständigkeit soll gelten, dass alle Eigenvektoren  $\mathbf{v}_\nu$  paarweise orthonormal zueinander sind.<sup>30</sup> Als Norm wird die Quadratnorm gewählt, die auch über das Skalarprodukt angegeben werden kann:

$$\|\mathbf{a}\| = \sqrt{\langle \mathbf{a}, \mathbf{a} \rangle} . \quad (7.99)$$

Jeder Vektor  $\mathbf{a}$  lässt sich als Linearkombination der Eigenvektoren schreiben:

$$\mathbf{a} = \sum_{\nu} a_{\nu}^{\mathbf{v}} \mathbf{v}_{\nu} . \quad (7.100)$$

Durch die Orthonormalität gilt dann

$$\|\mathbf{a}\| = \sum_{\nu} |a_{\nu}^{\mathbf{v}}| . \quad (7.101)$$

Für die Abbildung mit  $\mathbf{C}_1$  ist

$$\mathbf{b} = \mathbf{C}_1\mathbf{a} = \mathbf{C}_1 \sum_{\nu} a_{\nu}^{\mathbf{v}} \mathbf{v}_{\nu} = \sum_{\nu} \lambda_{\nu} a_{\nu}^{\mathbf{v}} \mathbf{v}_{\nu} \quad (7.102)$$

und daher

$$\|\mathbf{b}\| = \sum_{\nu} |\lambda_{\nu} a_{\nu}^{\mathbf{v}}| \leq \max_{\nu} \{|\lambda_{\nu}|\} \sum_{\nu} |a_{\nu}^{\mathbf{v}}| = \max_{\nu} \{|\lambda_{\nu}|\} \|\mathbf{a}\| . \quad (7.103)$$

---

<sup>30</sup>Unter der Voraussetzung der Vollständigkeit ist die Angabe eines orthonormalen Systems immer möglich.

Ein Eigenwert mit maximalem Betrag sei bei  $\nu_0$  zu finden, d. h.  $|\lambda_{\nu_0}| = \max_{\nu} \{|\lambda_{\nu}|\} = r_{\mathcal{C}_1}$ . Setzen wir  $\mathbf{a} = a_{\nu_0} \mathbf{v}_{\nu_0}$ , so wird die obere Grenze in Gleichung (7.103) erreicht, d. h. es ist  $\mathbf{b} = \lambda_{\nu_0} \mathbf{a}$  und  $\|\mathbf{b}\| = r_{\mathcal{C}_1} \|\mathbf{a}\|$ . Hier zeigt sich, dass man das Stabilitätsverhalten von Simulationen im Falle vollständiger Sätze von Eigenvektoren am Spektralradius, d. h. am maximalen Betrag aller Eigenwerte, festmachen kann: Ist  $r_{\mathcal{C}_1} > 1$ , können die Beträge der Feldkoeffizienten über alle Schranken wachsen. Nur wenn  $r_{\mathcal{C}_1} \leq 1$  ist, bleiben die Feldkoeffizienten beschränkt.

Weiterhin kann aus dem Ungleichheitszeichen in Gleichung (7.95) ein Gleichheitszeichen gemacht werden: Aus der Definition der Norm,

$$\|\mathcal{T}\| \stackrel{\text{def}}{=} \sup_{\substack{f \in \mathcal{D}(\mathcal{T}) \\ f \neq 0}} \left\{ \frac{\|\mathcal{T}f\|}{\|f\|} \right\} = \sup_{\substack{f \in \mathcal{D}(\mathcal{T}) \\ \|f\|=1}} \{\|\mathcal{T}f\|\}, \quad (7.104)$$

und dem Zusammenhang mit den Matrizen und Eigenvektoren folgt

$$\|\mathcal{C}_1\| = \max_{\|\mathbf{v}_{\nu}\|=1} \{\|\mathbf{C}_1 \mathbf{v}_{\nu}\|\} = |\lambda_{\nu_0}| = r_{\mathcal{C}_1}. \quad (7.105)$$

Bisher wurde immer gesagt, dass  $\|\mathcal{C}_1\| > 1$  sein kann, selbst wenn  $r_{\mathcal{C}_1} \leq 1$  ist. Diese Aussage ist i. A. auch weiterhin gültig, wenn kein vollständiges System von Eigenvektoren vorliegt. Ist das System aber vollständig, sind Operatornorm und Spektralradius identisch! Hier zeigt sich eigentlich die deutlichste Rechtfertigung für die Annahme des starken Stabilitätskriteriums in Abschnitt 7.3.6.3.1: Ist der maximale Betrag aller Eigenwerte höchstens gleich Eins, so ist wegen  $\|\mathcal{C}_1\| = r_{\mathcal{C}_1}$  auch  $\|\mathcal{C}_1\| \leq 1$ .

Gleichung (7.105) erlaubt sogar eine noch weitergehende Schlussfolgerung. Ist  $\{\lambda_{\nu}\}$  die Menge aller Eigenwerte von  $\mathbf{T}$  und bildet  $\{\mathbf{v}_{\nu}\}$  ein vollständiges System von Eigenvektoren zu  $\mathbf{T}$ , so gilt, dass die  $i$ -te Potenz der Matrix,  $\mathbf{T}^i$ , die Eigenwerte  $\{(\lambda_{\nu})^i\}$  hat [35]. Ferner bestimmen sich die Eigenvektoren zu  $\{\mathbf{v}_{\nu}^i = \mathbf{T}^{i-1} \mathbf{v}_{\nu}\}$ . Angewendet auf die FDTD-Operatoren gilt mit Gleichung (7.105)

$$\|\mathcal{C}_1^n\| = \max_{\|\mathbf{v}_{\nu}\|=1} \{\|\mathbf{C}_1^n \mathbf{v}_{\nu}\|\} = |\lambda_{\nu_0}|^n = \|\mathcal{C}_1\|^n. \quad (7.106)$$

Die Herleitung der in Abschnitt 7.3.6.3.1 präsentierten Stabilitätskriterien wurde oftmals dadurch erschwert, dass es zu Abschätzungen wie  $\|\mathcal{C}_1^n\| \leq \|\mathcal{C}_1\|^n$  (Gleichung (7.68)) keine weiteren bzw. schärferen Alternativen gab. Insbesondere die Verfügbarkeit einer kleineren oberen Schranke

als  $\|\mathcal{C}_1\|^n$  wäre hilfreich gewesen. Mit dem hier diskutierten Fall zeigt sich anhand von Gleichung (7.106) jedoch, dass diese Abschätzung durchaus ihre Berechtigung hat.

Im Waveletbereich liegen die gleichen Verhältnisse wie für  $\mathcal{C}_1$  und Gleichung (7.105) vor, wie kurz gezeigt werden soll. Die Matrix des Wavelet-transformierten Operators  $\tilde{\mathcal{C}}_1$  ist durch die Matrixmultiplikation

$$\tilde{\mathcal{C}}_1 = \mathbf{P}\mathbf{C}_1\mathbf{P}^* \quad (7.107)$$

zu bestimmen. Wegen  $(\mathbf{P})^{-1} = \mathbf{P}^*$  bedeutet die Zuordnung in Gleichung (7.107), dass  $\tilde{\mathcal{C}}_1$  und  $\mathbf{C}_1$  über die reguläre Matrix  $\mathbf{P}$  *ähnlich* zueinander sind [11]. Damit haben  $\tilde{\mathcal{C}}_1$  und  $\mathbf{C}_1$  die gleichen Eigenwerte, was folglich auch für den Operator  $\tilde{\mathcal{C}}_1$  gilt. Die Eigenvektoren sind i. A. allerdings unterschiedlich, dennoch bleibt die Vollständigkeit erhalten: Aus

$$\mathbf{C}_1\mathbf{v}_\nu = \lambda_\nu\mathbf{v}_\nu = \mathbf{P}^*\tilde{\mathcal{C}}_1\mathbf{P}\mathbf{v}_\nu \quad (7.108)$$

folgt durch durch linksseitige Multiplikation mit  $\mathbf{P}$

$$\lambda_\nu\mathbf{P}\mathbf{v}_\nu = \tilde{\mathcal{C}}_1\mathbf{P}\mathbf{v}_\nu . \quad (7.109)$$

Daher sind alle Vektoren  $\mathbf{w}_\nu$  mit  $\mathbf{w}_\nu = \mathbf{P}\mathbf{v}_\nu$  Eigenvektoren von  $\tilde{\mathcal{C}}_1$ . Außerdem ist die Anzahl der Eigenvektoren von  $\tilde{\mathcal{C}}_1$  die gleiche wie bei  $\mathbf{C}_1$ .

### 7.3.6.3.3 Abschließende Betrachtungen

Am Ende von Abschnitt 7.3.6.2 wurde erwähnt, dass das vorgestellte Detailgitter-Verfahren etwas allgemeiner implementiert werden kann, indem man verschiedene Detailgitter für das elektrische und das magnetische Feld zulässt.  $\mathcal{M}_G$  würde dann ersetzt werden durch  $\mathcal{M}_G^{\tilde{\mathbf{E}}}$  für das elektrische und  $\mathcal{M}_G^{\tilde{\mathbf{H}}}$  für das magnetische Feld. Für den zusammengesetzten Feldvektor  $\tilde{\mathbf{F}}$  ergäbe sich die Detailgitter-Maskierung aus Gleichung (7.76) dann zu

$$\mathcal{M}_G^{\tilde{\mathbf{F}}} = \begin{pmatrix} \mathcal{M}_G^{\tilde{\mathbf{E}}} & 0 \\ 0 & \mathcal{M}_G^{\tilde{\mathbf{H}}} \end{pmatrix} . \quad (7.110)$$

Die Überprüfung der Stabilitätsanalysen aus Abschnitt 7.3.6.3.1 auf diese Änderung hin zeigt, dass auch in diesem Fall sämtliche Stabilitätsaussagen ihre Gültigkeit bewahren.

Im Rahmen der Matrix-basierten FDTD im Waveletbereich hat Abschnitt 7.2.5.1 das Verfahren der räumlichen Adaptivität behandelt. Hierfür wurde in [3] lediglich empirisch festgestellt, dass dieses Verfahren scheinbar dem gleichen Stabilitätskriterium unterliegt wie eine klassische FDTD. Die Erkenntnisse aus Abschnitt 7.3.6.3 können zumindest teilweise auf das Verfahren der räumlichen Adaptivität übertragen werden.

Ein Unterschied zwischen dem Verfahren der räumlichen Adaptivität und dem Detailgitter-Verfahren besteht darin, dass Ersteres keinen Nutzen aus einer Kompression des Urbildes zieht. Das bedeutet, dass zwar vernachlässigbare Entwicklungskoeffizienten nicht berechnet werden, allerdings werden bei dieser Berechnung sowohl die vernachlässigbaren als auch die nicht-vernachlässigbaren Koeffizienten herangezogen. Dementsprechend können die Ansätze mit den Maskierungsoperatoren des Detailgitter-Verfahrens nicht unmittelbar übernommen werden. Ein weiterer Unterschied besteht darin, dass die räumliche Adaptivität zeitabhängig ist, da die Festlegung, welche Koeffizienten nicht zu vernachlässigen sind, von Zeit zu Zeit aktualisiert wird. Ferner wird lediglich beim Detailgitter-Verfahren von der Bereichsanpassung Gebrauch gemacht. Das ist auch in dem Zusammenhang zu sehen, dass elektrische und magnetische Wände in beiden Verfahren unterschiedlich implementiert werden.

Dennoch sind sich die Verfahren recht ähnlich. Die Vernachlässigung von Koeffizienten im Bildraum lässt sich darstellen als

$$\tilde{\mathbf{F}}^{n+1} = \begin{pmatrix} \text{id} & 0 \\ 0 & \mathcal{M}_{\tilde{\mathbf{G}}}^{\tilde{\mathbf{H}}}(n) \end{pmatrix} \tilde{\mathcal{C}}_1 \begin{pmatrix} \mathcal{M}_{\tilde{\mathbf{G}}}^{\tilde{\mathbf{E}}}(n) & 0 \\ 0 & \text{id} \end{pmatrix} \tilde{\mathcal{C}}_2 \tilde{\mathbf{F}}^n . \quad (7.111)$$

In [3] wird nicht erwähnt, ob die vernachlässigbaren Koeffizienten explizit zu Null gesetzt werden, oder auf ihrem (betragsmäßig kleinen) Wert belassen werden. Wahrscheinlich ist von Ersterem auszugehen. Gleichung (7.111) gibt dann das Verfahren der räumlichen Adaptivität nur näherungsweise wieder.

Die Ausmaskierung von Entwicklungskoeffizienten wurde beim Detailgitter-Verfahren relativ allgemein gehalten. Sowohl die einfache als auch die zwei- bzw. dreifache Maskierung (Gleichungen (7.77) und (7.92)) konnten unter den angesetzten Voraussetzungen als stabil ausgewiesen werden. Ebenso sind verschiedene Maskierungsoperatoren für das elek-

trische und das magnetische Feld zulässig. Trotz der bestehenden Unterschiede beider Verfahren erscheint es als wahrscheinlich, dass die in Abschnitt 7.3.6.3.1 gezeigten Methoden in entsprechend angepasster Form auch für die räumliche Adaptivität angewendet werden können. Da jedoch vernachlässigbare Koeffizienten – wie oben bereits erwähnt – vermutlich nicht zu Null gesetzt werden, ist ein Stabilitätsbeweis deutlich schwieriger und soll daher hier auch nicht unternommen werden. Die Verwandtschaft beider Verfahren sowie Erfahrungen mit Simulationen erheben allerdings die Vermutung, dass auch mit der räumlichen Adaptivität stets stabile Simulationen durchgeführt werden können.

Zum Schluss soll ein Blick auf Erweiterungen der klassischen FDTD im Zusammenhang mit dem Detailgitter-Verfahren geworfen werden. Viele dieser Erweiterungen weisen die gleichen Voraussetzungen auf, die auch beim Stabilitätsbeweis eine Rolle spielten (beispielsweise Linearität, Bijektivität). Trotz des nicht unerheblichen Eingriffs durch die Ausmaskierung von Koeffizienten bleiben die Simulationsverfahren stabil. Diese Art der Approximation scheint sich demnach relativ „harmlos“ auf Simulationsverfahren auszuwirken. Das bedeutet, dass die Detailgitter-Maskierung vermutlich oft auch dann keinen negativen Einfluss auf die Stabilität eines Simulationsverfahrens hat, wenn dieses nicht die Voraussetzungen aus Abschnitt 7.3.6.3.1 erfüllt.

Ein Beweis dieser Vermutung kann zwar nicht angeführt werden, es kann aber zumindest auf Erfahrungen im Umgang mit PMLs verwiesen werden. Diese neigen manchmal zu instabilem Verhalten, wie in Abschnitt 2.3.2 bereits erwähnt wurde. Im Rahmen dieser Arbeit sind jedoch nie Instabilitäten bei der Anwendung des Detailgitter-Verfahrens aufgetreten, obwohl PMLs als ABC verwendet wurden. Wie sehr sich diese Erfahrung auch auf andere Erweiterungen der FDTD anwenden lässt, müssen weitergehende Untersuchungen erbringen. So bleibt an dieser Stelle lediglich zu sagen, dass die Chancen gut stehen, dass die Detailgitter-Maskierung die Stabilität eines Verfahrens generell nicht oder nur wenig beeinflusst.

### 7.3.7 Lokale Nichtlinearitäten

Eine wesentliche Voraussetzung für die Darstellung der FDTD im Waveletbereich war bisher die Linearität aller Operatoren. Die Simulation

nichtlinearer Medien beispielsweise erfordert oft die Erweiterung der FDTD um nichtlineare Algorithmen [4]. Es sollen hier zwei Möglichkeiten durchgesprochen werden, wie nichtlineare Eigenschaften einbezogen werden können.

Die erste Möglichkeit besteht darin, eine Teilbereichs-WT nach Abschnitt 7.3.5 anzuwenden und dabei nur in dem Bereich die WT durchzuführen, der ausschließlich lineare Eigenschaften aufweist. In dem untransformierten Bereich können dann die nichtlinearen Algorithmen des ursprünglichen Verfahrens eingesetzt werden. Diese Möglichkeit lässt sich allerdings nur dann sinnvoll einsetzen, wenn sich die Nichtlinearitäten nicht mitten im Simulationsgebiet befinden, sodass ein zusammenhängender großer Raumbereich für die Teilbereichs-WT zur Verfügung steht.

Eine zweite Möglichkeit ergibt sich für den Fall, dass lediglich einzelne Stellen bzw. relativ kleine Raumbereiche nichtlineare Eigenschaften aufweisen. Diese Situation liegt beispielsweise dann vor, wenn diskrete Bauelemente mit nichtlinearen Kennlinien in einem ansonsten linearen Medium simuliert werden sollen. Hierbei kommt die in Abschnitt 7.3.4 diskutierte optimierte Feldanregung und -ausgabe zum Einsatz. Die Berechnung der nichtlinearen Stellen wird so durch lokale Transformationen im untransformierten Grundraum ermöglicht. Die Idee soll nun kurz formalisiert werden.

Der nichtlineare Operator  $\mathcal{T}$  wird aufgeteilt in einen rein linearen Anteil  $\mathcal{T}'$  und in den nichtlinearen Anteil  $\mathcal{T}''$ , sodass gilt  $\mathcal{T} = \mathcal{T}' + \mathcal{T}''$ .  $\mathcal{T}'$  kann wie gewohnt durch Tensorelemente ausgedrückt werden. Zu dem Bildbereich  $\lambda_B(\mathcal{T}')$  kann ein Maskierungsoperator  $\mathcal{M}'$  angegeben werden, wodurch  $\mathcal{T}' = \mathcal{M}'\mathcal{T}$  gilt. Ferner wird der komplementäre Bildbereich durch  $\mathcal{M}'_1 = \bar{\mathcal{M}}'$  gebildet, d. h. es ist  $\mathcal{M}' + \mathcal{M}'_1 = \text{id}$ . Schließlich wird noch ein zum Urbildbereich  $\lambda_U(\mathcal{T}'')$  passender Maskierungsoperator  $\mathcal{M}''_2$  benötigt. Damit gilt

$$\mathcal{T} = \mathcal{T}' + \mathcal{T}'' = \mathcal{T}' + \mathcal{M}''_1 \mathcal{T}'' \mathcal{M}''_2 . \quad (7.112)$$

Der Wavelet-transformierte Operator ergibt sich dann zu

$$\tilde{\mathcal{T}} = \mathcal{P} \mathcal{T}' \mathcal{P}^* + \mathcal{P} \mathcal{M}''_1 \mathcal{T}'' \mathcal{M}''_2 \mathcal{P}^* \quad (7.113)$$

$$= \tilde{\mathcal{T}}' + \tilde{\mathcal{M}}''_1 \mathcal{T}'' \tilde{\mathcal{M}}''_2 , \quad (7.114)$$

d. h. es ist

$$\tilde{\mathcal{T}}' = \mathcal{P} \mathcal{T}' \mathcal{P}^* , \quad \tilde{\mathcal{M}}_1'' = \mathcal{P} \mathcal{M}_1'' , \quad \tilde{\mathcal{M}}_2'' = \mathcal{M}_2'' \mathcal{P}^* . \quad (7.115)$$

Der Wavelet-transformierte nichtlineare Operator wird somit nach Gleichung (7.114) zerlegt in die linearen Anteile nach Gleichung (7.115) sowie den nichtlinearen Anteil  $\tilde{\mathcal{T}}''$ . Während  $\tilde{\mathcal{T}}'$  der üblichen WT eines linearen Operators entspricht, stellen  $\tilde{\mathcal{M}}_1''$  und  $\tilde{\mathcal{M}}_2''$  Operatoren zur optimierten Feldanregung und -ausgabe dar, wie sie im bereits erwähnten Abschnitt 7.3.4 Verwendung finden. Auf diese Weise wird deutlich, wie örtlich begrenzte Nichtlinearitäten effizient in Simulationsverfahren für den Waveletbereich integriert werden können, sofern die Tensorelemente von  $\tilde{\mathcal{M}}_1''$  und  $\tilde{\mathcal{M}}_2''$  dünn besetzt sind.

Es sei an dieser Stelle bemerkt, dass die hier genannten Verfahren mit einer Matrix-basierten Formulierung der Wavelet-transformierten FDTD nur sehr umständlich möglich wären. Die Erklärung hierfür ist wiederum in der aufwendigen Zerlegung der Operatoren zu suchen, um die nichtlinearen Bildstellen von den linearen trennen zu können.

Abschließend soll nicht unerwähnt bleiben, dass abgesehen von den beiden genannten Möglichkeit natürlich versucht werden kann, die Nichtlinearitäten direkt im Waveletbereich auszudrücken. Techniken wie die in [29] vorgestellte können hierbei hilfreich sein. Es soll hierauf aber nicht näher eingegangen werden, stattdessen soll in dieser Arbeit der Fokus weiterhin auf der Transformation linearer Operatoren bleiben. Immerhin können auf diese Weise Operatoren für den Waveletbereich nutzbar gemacht werden, ohne sie „händisch“ in Waveletbasen entwickeln zu müssen. Sogar einzelne nichtlineare Elemente sind dabei handhabbar.

# Kapitel 8

## Simulationsergebnisse

Die FDTD ist als altbewährtes Hilfsmittel mit all ihren Vor- und Nachteilen hinlänglich bekannt. Insbesondere zum Thema numerische Genauigkeit wurden zahlreiche Untersuchungen publiziert. Werden FDTD-Operatoren mit den bijektiven WT-Operatoren  $\mathcal{P}$  und  $\mathcal{P}^*$  in den Waveletbereich transformiert, erbt das so erzeugte Simulationsverfahren alle Eigenschaften der FDTD. Erst die Kompression, d. h. die Vernachlässigung von Entwicklungskoeffizienten durch ein Schwellenwertverfahren oder das Detailgitter-Verfahren, bringt eine Änderung der numerischen Genauigkeit mit sich.

Es kann erwartet werden, dass die Genauigkeit der Wavelet-komprimierten FDTD von der untransformierten FDTD nur wenig abweicht, wenn die Kompression dem Simulationsproblem gut angepasst ist. Insofern ist der Aspekt der Genauigkeit zunächst einmal nur von sekundärer Bedeutung. Als deutlich wichtiger erscheint dagegen die Frage, ob sich der durch die WT der FDTD entstehende zusätzliche Aufwand überhaupt lohnt. Als Beispiel soll die Matrix-basierte WT der FDTD dienen: Allein der Hinweis auf die unvermeidbare Zerlegung von Operatoren (siehe Abschnitt 5.2.3) zeigt auf, dass hier mit einem erheblichen numerischen Mehraufwand zu rechnen ist. Die Kompression muss den Mehraufwand überkompensieren, damit im Endeffekt eine Zeitersparnis das neue Simulationsverfahren rechtfertigt. Dabei dürfen jedoch keine zu hohen Kompressionsraten verwendet werden, da dann durch die Abnahme der Genauigkeit ein Vergleich mit der untransformierten FDTD nicht mehr möglich wäre.

## 8.1 Simulationszeiten bei der Matrix-basierten FDTD

Das Verfahren der räumlichen Adaptivität (Abschnitt 7.2.5.1) bewirkt eine Kompression in der Art, dass in bestimmten Zeitabständen die Feldkoeffizienten darauf untersucht werden, ob ihre Beträge unterhalb eines bestimmten (sinnvollen) Schwellenwertes liegen. Im Grundraum sind in der Praxis kaum Koeffizienten zu finden, die mit diesem Kriterium als vernachlässigbar gekennzeichnet werden könnten. Erst im Waveletbereich ändert sich diese Situation.

Die WT zerlegt ein Signal in verschiedene Auflösungsstufen. Grob formuliert wird das Signal in Mittelwerte und verschieden große „Kantenstücke“ aufgeteilt. Dadurch findet bei genügend glatten Signalen eine Art Konzentrationsprozess statt, d. h. verhältnismäßig wenige Entwicklungskoeffizienten genügen, um das Signal hinreichend genau darzustellen. Diese Konzentration steigt meistens mit zunehmender Transformationstiefe, zumindest bis zu einem gewissen Grad. Das bedeutet, dass die Anzahl von Entwicklungskoeffizienten, deren Betrag unterhalb des Schwellenwertes liegt, mit der Transformationstiefe steigt. Letztlich bedeutet das, dass eine Variation der Transformationstiefe bei konstantem Schwellenwert im Wesentlichen nur die Kompressionsrate beeinflusst. Die Genauigkeit dagegen ändert sich allenfalls geringfügig.

In [3] und [19] wurde die Genauigkeit des Verfahrens der räumlichen Adaptivität bereits demonstriert. Daher soll hier ein besonderes Augenmerk auf die Kompressionsrate und die Rechenzeiten gelegt werden. Dazu wurde das Simulationsprogramm eines der Autoren der o. g. Artikel (Markus Werthen) verwendet, mit dem auch die dort publizierten Ergebnisse erzielt wurden. Die nun folgenden Beispiele wurden in ähnlicher Form bereits in [18] veröffentlicht.

Zunächst wurde die Ausbreitung einer transversalen elektromagnetischen Welle in einer homogenen Bandleitung simuliert, vgl. Abbildung 8.1. Durch die seitliche Berandung mit magnetischen Wänden bleibt die x-Polarisation der anregenden elektrischen Feldstärke während der Ausbreitung entlang der z-Achse erhalten. Tabelle 8.1 zeigt für verschiedene Fälle auf, welche Abhängigkeit die Simulationszeit von der Transformationstiefe hat. Dabei entspricht die erste Zeile mit  $j_x = j_y = j_z = 0$  der Simulation

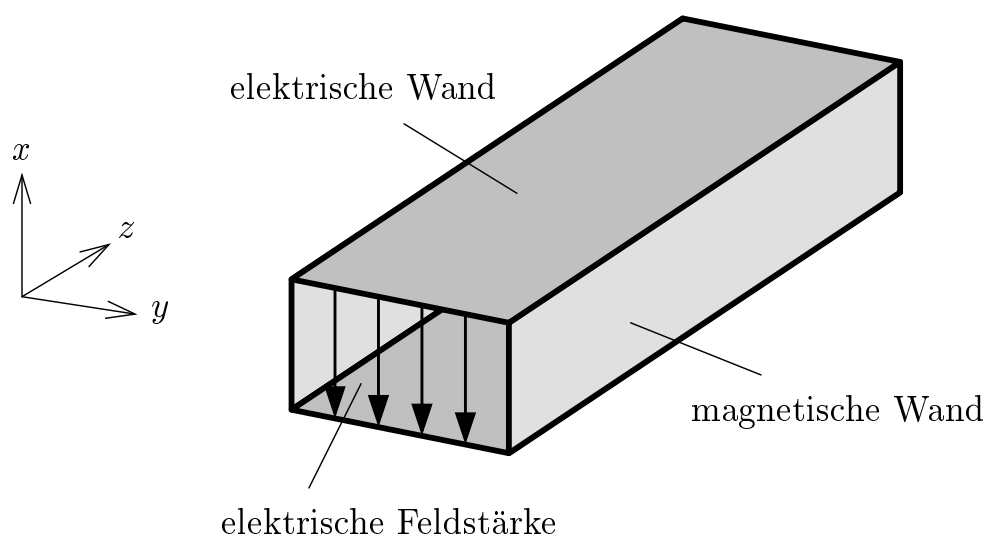


Abbildung 8.1: Anregung einer transversalen elektromagnetischen Welle in einer idealen Bandleitung. Eine in x-Richtung polarisierte Welle breitet sich entlang der z-Achse wie im unendlich ausgedehnten homogenen Raum aus. Diese Eigenschaft ist von den Abmessungen der Bandleitung unabhängig.

einer untransformierten FDTD. In diesem Fall wird kein Entwicklungskoeffizient vernachlässigt: Nach Abschnitt 7.2.5.1 wird im Skalierungsraum mit der höchsten Skala (das ist hier der gesamte Raum) grundsätzlich jeder Koeffizient berechnet. In den übrigen Fällen wird für den Schwellwertfaktor entsprechend der Empfehlung in [3] der Wert  $5 \cdot 10^{-4}$  angesetzt.

Der Übersichtlichkeit halber ist die Tabelle in einzelne Blöcke aufgeteilt, innerhalb derer die Transformationstiefen steigen. Es ist festzustellen, dass die Simulationszeiten deutlich von den Transformationstiefen abhängen. In den meisten Fällen ist entsprechend dem oben Gesagten die Abnahme der Simulationszeit mit steigender Transformationstiefe zu beobachten. Die Simulationszeit konnte in diesen Versuchen um höchstens 28,7% gesenkt werden.

Zum Vergleich wurde als noch einfachere Struktur ein Hertz'scher Dipol im Vakuum simuliert. Das Feldvolumen wurde durch eine Mur'sche ABC erster Ordnung begrenzt. Tabelle 8.2 zeigt die Simulationszeiten. Es fällt auf, dass die maximale Einsparung an Simulationszeit im Vergleich zu Tabelle 8.1 geringer, aber dennoch ähnlich ist. In keinem Fall liegt die Reduktion unterhalb von fünf Prozent. Insofern scheint die Abhängig-

$j_x$	$j_y$	$j_z$	Zeit/s	Zeit/%
0	0	0	80,3	100,0
1	0	0	71,9	89,5
2	0	0	68,8	85,7
3	0	0	64,9	80,8
4	0	0	64,8	80,7
0	1	0	69,6	86,7
0	2	0	63,9	79,6
0	3	0	58,5	72,3
0	0	1	77,8	96,9
0	0	2	62,0	77,2
0	0	3	57,3	71,3
1	1	1	79,8	99,4
2	2	2	67,5	84,1
3	3	3	67,8	84,4
0	1	1	79,6	99,1
0	2	2	60,8	75,7
0	3	3	60,9	75,8

Tabelle 8.1: Simulationszeiten für die Ausbreitung einer ebenen Welle in der Bandleitung nach Abbildung 8.1. Die Simulation wurde mit homogenen Tensorprodukt-Basen durchgeführt.  $j_\nu$  ist die Transformationstiefe in Richtung  $\nu$ .

keit von der Transformationstiefe in diesem Beispiel etwas weniger stark ausgeprägt zu sein.

Um die Effizienz des Simulationsprogramms zu untersuchen, wurde ein reines FDTD-Programm geschrieben, d. h. der FDTD-Algorithmus wurde direkt implementiert anstatt über Matrix-Vektor-Multiplikationen. Die Simulationszeit wurde mit dem Fall  $j_x = j_y = j_z = 0$  aus Tabelle 8.2 verglichen. Es stellte sich dabei heraus, dass das reine FDTD-Programm etwa um den Faktor zehn schneller war.

Obwohl das Matrix-basierte Simulationsprogramm sicher noch einigen Optimierungsspielraum lässt, stellt dieser Geschwindigkeitsunterschied eine ziemliche Hürde dar. Ein Grund für die doch überraschend hohe Rechenzeit mag darin liegen, dass keine lokale Rücktransformation möglich

$j_x$	$j_y$	$j_z$	Zeit/s	Zeit/%
0	0	0	17043	100
0	0	1	14913	88
0	0	3	13643	80
0	1	0	14544	85
0	3	0	12978	76
1	0	0	14471	85
3	0	0	12988	76
1	1	1	14377	84
2	2	2	14801	87

Tabelle 8.2: Berechnungszeiten für die Simulation eines Hertz'schen Dipols im Vakuum bei verschiedenen Transformationstiefen.

ist, welche bei der Ausgabe von Feldwerten von Vorteil wäre, vgl. Abschnitt 7.2.4. Die Geschwindigkeit des Matrix-basierten Verfahrens ist auch unter dem Aspekt zu sehen, dass bei komplizierteren Problemen eine aufwendige Zerlegung der Operatoren in  $U_0^\nu$ -Operatoren notwendig ist, was den numerischen Aufwand weiter in die Höhe treibt. Ferner ist dieses Simulationsverfahren durch die erwähnte Zerlegungsproblematik recht unflexibel. Es konnte auch trotz der Transformation in den Waveletbereich und der Vernachlässigung von Koeffizienten keine genügende Beschleunigung erzielt werden. Aufgrund dieser Beobachtungen wurden auch keine weiteren Simulationen mit diesem Verfahren durchgeführt.

Als Fazit bleibt festzuhalten, dass das Matrix-basierte Verfahren zwar einfach zu implementieren ist, für praktische Probleme aber anscheinend keine ausreichende Effizienz aufweist. Daher eignet es sich vorwiegend für grundlegende Untersuchungen, wie beispielsweise den Aufbau von Operatoren nach einer WT, erzielbare Kompressionsraten oder auch die Erweiterung Wavelet-basierter Simulationsverfahren. So konnte das Simulationsprogramm erfolgreich um die in Abschnitt 7.2.5.2 beschriebene zeitliche Adaptivität erweitert werden.

## 8.2 Simulation mit der Listen-basierten FDTD

### 8.2.1 Hohlleiter-Diskontinuität

Abbildung 8.2 zeigt einen Rechteck-Hohlleiter, in dem ein dielektrischer Pfosten platziert ist. Der Wellenleiter wurde zunächst mit einer konventionellen FDTD für verschiedene Diskretisierungsweiten  $h$  simuliert. Hierbei hatte der Wellenleiter in allen Fällen eine Gesamtlänge von 180 mm, wobei die Enden mit einer sechs Zellen dicken PML-Schicht abgeschlossen wurden. Analysiert werden sollte der Grundtyp ( $TE_{10}$ ) des Rechteck-Hohlleiters. Da dieser keine Abhängigkeit von  $y$  hat und dies auch für die Materialverteilung im Inneren des Hohlleiters gilt, konnte der Feldraum in  $y$ -Richtung durch eine einzelne Schicht aus Yee-Zellen diskretisiert werden. Die Anzahl der Zeitschritte wurde in Abhängigkeit von der maximal erlaubten Zeitschrittweite jeweils so gewählt, dass die vollständige Simulation einer physikalischen Zeit von 17,5 ns entsprach.

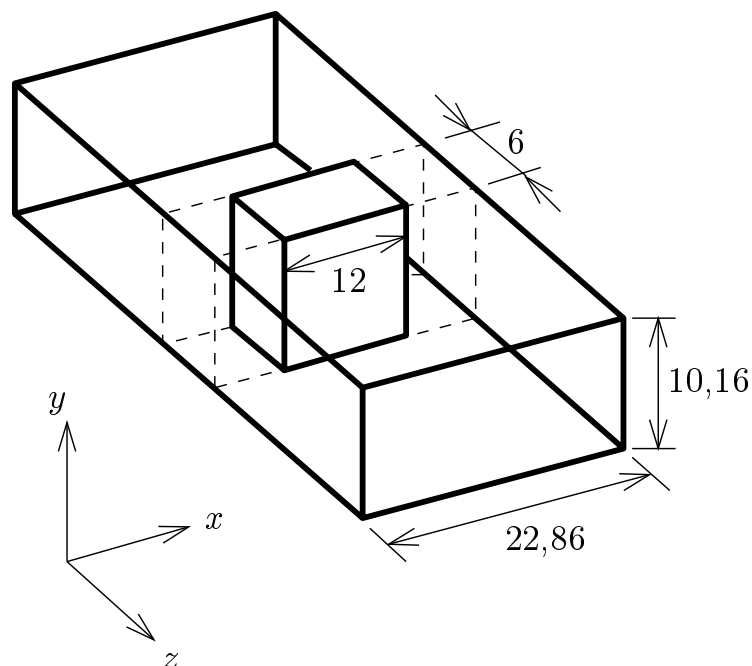


Abbildung 8.2: In  $z$ -Richtung orientierter Rechteck-Hohlleiter mit dielektrischem Pfosten (Permittivitätszahl  $\epsilon_r = 8,2$ ). Alle Angaben in mm.

Der Vergleich zwischen Simulations- und Messergebnissen in Abbildung 8.3 zeigt, dass eine Verringerung der räumlichen Diskretisierungsweite  $h$  unter einen Wert von 0,6 mm keine nennenswerte Verbesserung in der Genauigkeit erbringt. Die auch noch bei 0,3 mm Diskretisierungsweite bestehenden Abweichungen von den Messergebnissen mögen zumindest teilweise in prinzipiellen numerischen Beschränkungen des FDTD-Verfahrens begründet sein. Da jedoch auch mit einer anderen numerischen Methode in [36] gleichartige Abweichungen zu finden sind, dürften die Hauptursachen hierfür Mess- und Herstellungsungenauigkeiten sein.

Aus Tabelle 8.3 gehen die Simulationszeiten hervor, die für die verschiedenen Diskretisierungsweiten aufzuwenden waren. Zusammen mit der aus

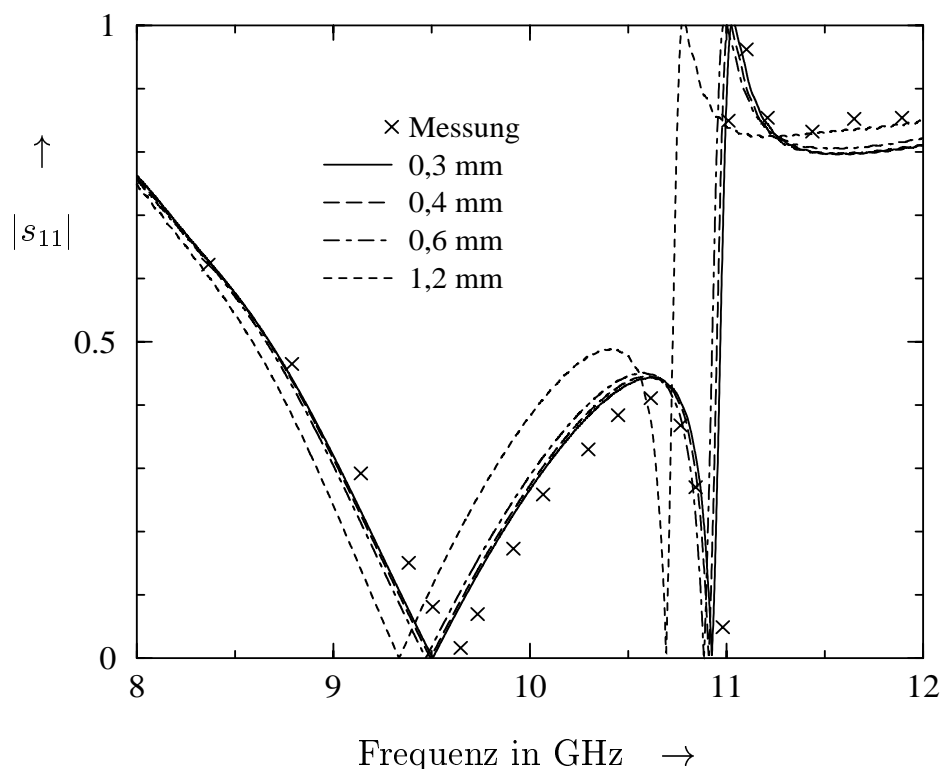


Abbildung 8.3: Simulationsergebnisse (Linien) und Messergebnisse (Symbole) des Betrags des Streuparameters  $s_{11}$  für den Wellenleiter nach Abbildung 8.2. Die angegebenen Diskretisierungsweiten  $h$  von 0,3 mm bis 1,2 mm gelten für den Bereich des Dielektrikums. Für den übrigen Bereich wurden geringfügig höhere Werte angesetzt. Messergebnisse wurden [36] entnommen, wobei dort für die Permittivitätszahl die Angabe  $8,2-j0,006$  zu finden ist.

$h/\text{mm}$	Zeit/s
0,3	111,0
0,4	46,7
0,6	11,2
1,2	2,3

Tabelle 8.3: Gemessene Zeiten für die Simulationen nach Abbildung 8.3 in Abhängigkeit der Diskretisierungsweite  $h$ .

Abbildung 8.3 ersichtlichen Genauigkeit scheint für  $h = 0,6$  mm ein akzeptabler Kompromiss vorzuliegen, sodass dies als Referenz für die weiteren Untersuchungen dienen soll.

Vor der Durchführung der Simulation im Waveletbereich sind zunächst einige Fragen zu klären und Parameter festzulegen:

- In welche Richtungen soll transformiert werden?
- Bis zu welcher Skala soll die WT jeweils durchgeführt werden?
- Welche Wavelets sollen verwendet werden?
- Welchen Einfluss hat es, wenn elektrische und magnetische Wände nicht im Detailbereich liegen (vgl. Abschnitt 7.3.6)?
- PMLs haben eine sehr inhomogene Materialverteilung. Welchen numerischen Einfluss hat es, wenn sie nicht Wavelet-transformiert werden (vgl. Abschnitt 7.3.5, Teilbereichs-WT)? Wenn sie im Bereich der WT liegen, müssen sie dann in einem Detailbereich liegen?
- An der Grenzschicht Luft-Dielektrikum können Feldkomponenten einen „Knick“ haben (sie sind stetig, aber nicht mehr differenzierbar). Im gesamten Dielektrikum sind Wellenlängen gegenüber der Luft verkürzt, d. h. vor allem hier ist eine hohe räumliche Auflösung erforderlich. Genügt es, den Bereich der Grenzschicht als Detailbereich zu definieren oder sollte dieser das gesamte Dielektrikum umfassen?
- Sollen die Rotationsoperatoren als Gesamtoperator transformiert werden, oder ist es besser, die drei kartesischen Ableitungen getrennt zu behandeln?

- Ist es von Vorteil, Feldanregungen und -ausgaben mit angepassten Operatoren durchzuführen (vgl. Abschnitt 7.3.4)?

Angesichts der Fülle an Möglichkeiten dürfte klar sein, dass hier nicht alle Kombinationen untersucht werden können. Stattdessen sollen die Fragestellungen anhand einiger ausgewählter Beispiele untersucht werden.

### 8.2.1.1 Einflüsse des Detailgitter-Verfahrens

Zuerst soll demonstriert werden, welche Notwendigkeit besteht, Detailbereiche um elektrische Wände herum anzulegen. Dazu soll der Wellenleiter in Abbildung 8.2 in  $x$ -Richtung transformiert werden. Für die Gebiete der beiden PMLs und des dielektrischen Pfostens (sowie zusätzlich ein bzw. zwei Zellen des sie umgebenden Raumes) wird jeweils ein vollständiger Detailbereich definiert, sodass aus diesen Bereichen keine zusätzlichen numerischen Ungenauigkeiten zu erwarten sind. Für die beiden senkrecht zur  $x$ -Richtung orientierten elektrischen Wände können Detailbereiche beliebiger Breite definiert werden, wie in Abbildung 8.4 gezeigt wird.

Das Ergebnis einiger Simulationen ist in Abbildung 8.5 zu sehen. Wird kein Detailbereich um die elektrischen Wände gelegt ( $d = 0$ ), können deutliche Abweichungen des Streuparameters auftreten. Lediglich im ein-

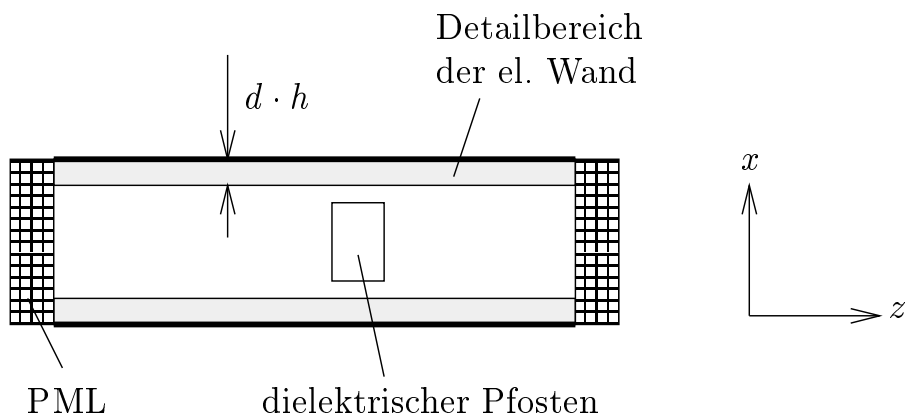


Abbildung 8.4: Platzierung von Detailbereichen an äußeren elektrischen Wänden des Wellenleiters aus Abbildung 8.2 (Darstellung nicht maßstabsgerecht).  $d$  ist die Breite des Bereiches in Zellen. Die Bereiche beginnen mit den elektrischen Wänden ( $d = 1$ ) und wachsen mit zunehmendem  $d$  in den Feldbereich.

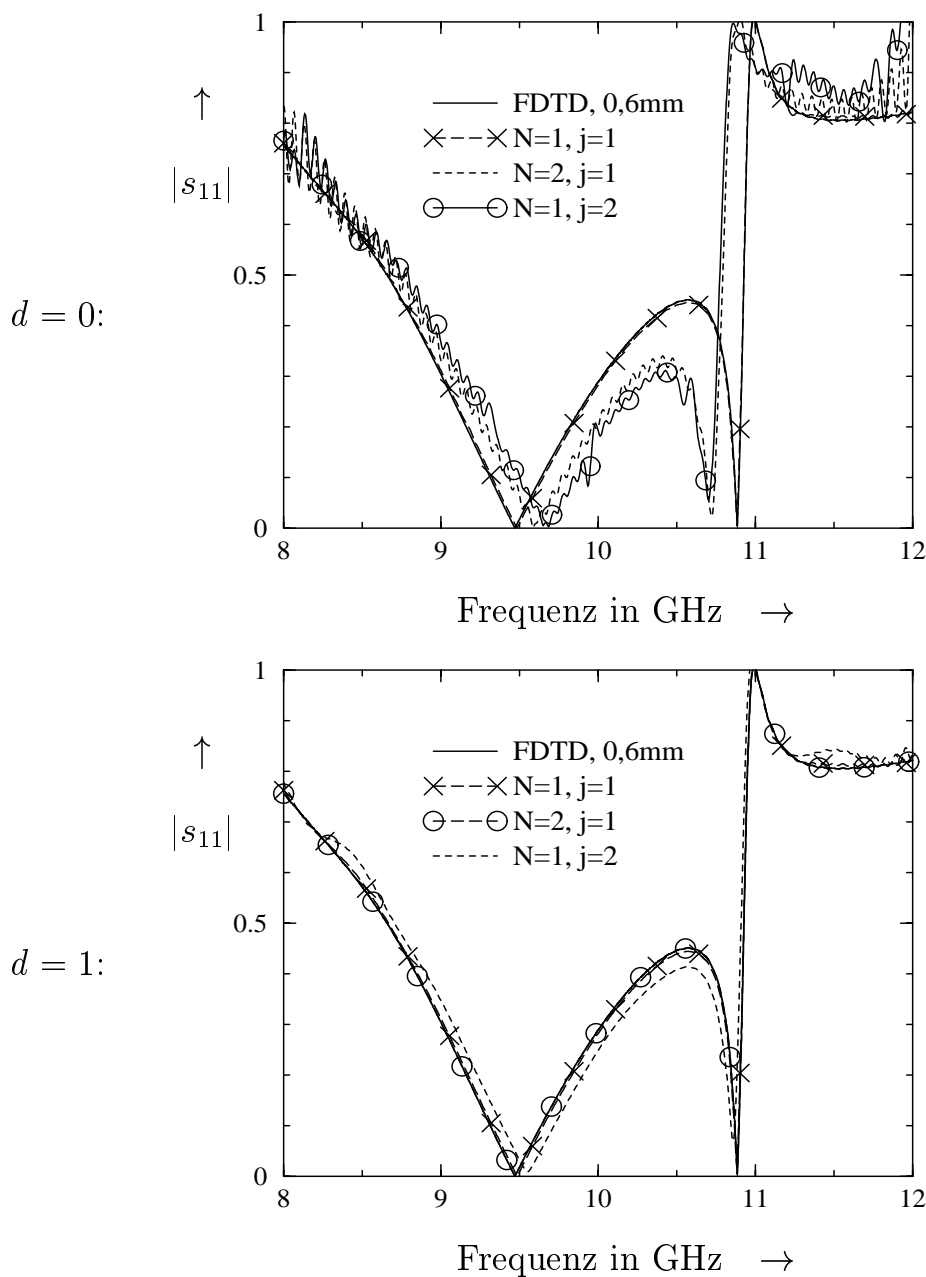


Abbildung 8.5: Einfluss der Dicke  $d$  der Detailbereiche aus Abbildung 8.4. „FDTD, 0,6 mm“ ist der Referenzgraph aus Abbildung 8.3. Für die weiteren Graphen gilt: WT des Wellenleiters in x-Richtung um  $j$  Stufen; Daubechies-Funktionen der Ordnung  $N$  (s. Abschnitt 3.2.3); weitere Detailbereiche um beide PMLs und den dielektrischen Pfosten (in Abbildung 8.4 nicht eingezeichnet).  $d = 0$  bedeutet, dass für die elektrischen Wände kein Detailbereich definiert wird.

fachsten Fall (Haar-Funktionen, nur eine Transformationsstufe) sind die Abweichungen sehr gering. Das gilt zumindest in dem gezeigten Frequenzbereich und für dieses Beispiel. Die Simulationsgenauigkeit verbessert sich bereits dann, wenn nur für die Wände selber ( $d = 1$ ) Detailbereiche angegeben werden. Weitere Versuche mit  $d = 2$  ergaben Graphen, die mit denen für  $d = 1$  deckungsgleich waren und daher hier nicht wiedergegeben sind. Die Deckungsgleichheit lässt darauf schließen, dass die bei der zweistufigen WT ( $j = 2$ ) beobachtbaren Abweichungen nicht speziell auf eine mangelnde Auflösung der elektrischen Wände zurückzuführen sind, sondern dass hier generelle Kompressionsartefakte in Erscheinung treten.

Als nächsten Punkt sollen jetzt PMLs betrachtet werden. Es wurde bereits mehrfach angedeutet, dass die Transformation von PMLs in den Waveletbereich eine deutliche Erhöhung des numerischen Aufwands bedeuten kann. Bevor dieser Aspekt genauer untersucht wird, soll überprüft werden, ob der Aufwand durch eine komprimierte Darstellung der PMLs reduziert werden kann, ohne die Genauigkeit zu sehr abzusenken.

Abbildung 8.6 zeigt einige Simulationsergebnisse. In allen Fällen wurden für den dielektrischen Pfosten, die elektrischen Wände und eine zwei Zellen breite Schicht am Wellenleitereingang Detailbereiche definiert. In der letztgenannten Schicht wurden Feldanregungen und -ausgaben durchgeführt. Für die einstufige WT wurden die Haar-Funktionen benutzt. Im Gegensatz zu den Simulationen aus Abbildung 8.5 wurde diesmal in  $z$ -Richtung transformiert, da die Materialparameter der PMLs in diesem Beispiel lediglich von  $z$  abhängen.

Die größten Abweichungen treten dann auf, wenn die PMLs nicht in einem Detailbereich liegen (Graphen „Wellenleiter“ und „Diel. Pfosten“). Sobald jedoch für die PMLs ein Detailbereich definiert wird und die PMLs dadurch nach der WT unkomprimiert bleiben, verringern sich die Abweichungen. Weiterhin wurden Simulationen mit einer zweistufigen WT durchgeführt. In diesem Fall wurden die Simulationen instabil, wenn für die PMLs kein Detailbereich angegeben wurde.

In PMLs klingen die Felder mit zunehmender Tiefe sehr schnell ab. Solche Verläufe lassen sich nur durch Funktionen mit räumlich hoher Auflösung gut beschreiben. Das dürfte der Grund dafür sein, warum es bei komprimierten PMLs zu den beobachtbaren Abweichungen kommt. Die hervorgerufenen Ungenauigkeiten mögen ggf. noch akzeptabel sein, doch die

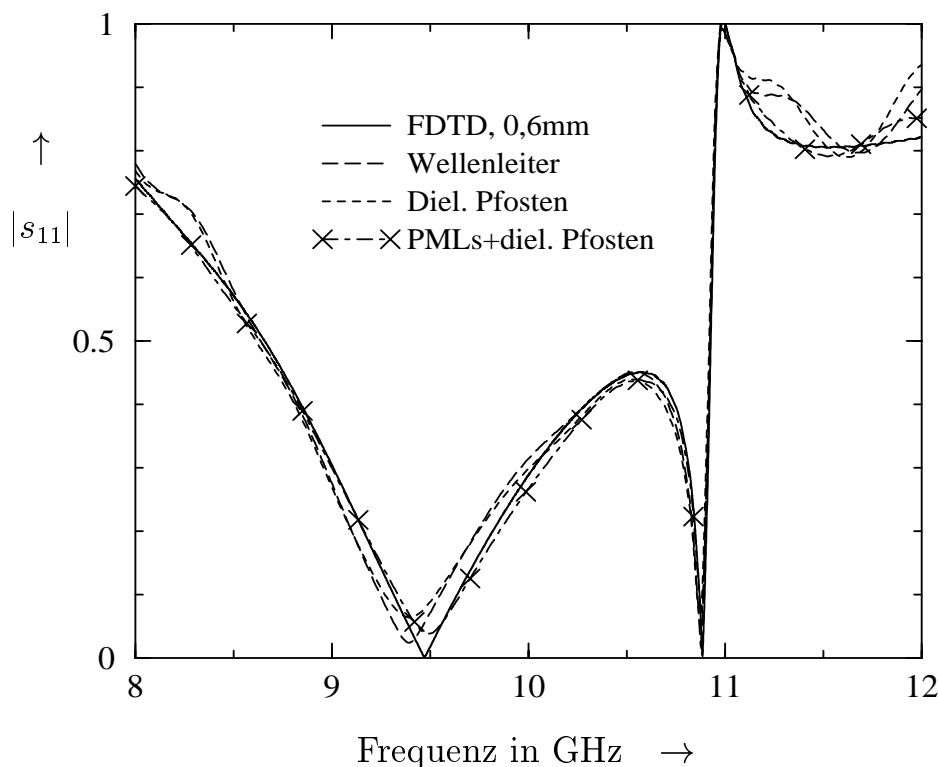


Abbildung 8.6: Detailbereiche und PMLs: Die Simulationsergebnisse für den Wellenleiter aus Abbildung 8.2 zeigen, dass für PMLs bei einer WT Detailbereiche angegeben werden müssen. „FDTD, 0,6 mm“ : Referenzgraph aus Abbildung 8.3. „Wellenleiter“ : Bis auf die PMLs liegt alles in einem Detailbereich. „Diel. Pfosten“ : Nur der dielektrische Pfosten liegt in einem Detailbereich. „PMLs+diel. Pfosten“ : Beide PMLs und der dielektrische Pfosten liegen in Detailbereichen.

Gefahr des Entstehens von Instabilitäten zeigt, dass PMLs entweder nicht transformiert werden sollten oder aber zumindest nach der WT unkomprimiert bleiben sollten.

Gehen wir nun die Frage an, wie dielektrische Volumina bzw. ihre Grenzschichten zu behandeln sind. Zwei Vorgehensweisen sollen hier miteinander verglichen werden. Im ersten Fall wird die Grenzfläche des dielektrischen Pfostens mit einem zwei Zellen breiten Detailbereich umgeben. Im zweiten Fall wird dieser Detailbereich auf das gesamte Volumen des Pfostens erweitert. Abbildung 8.7 gibt die Simulationsergebnisse beider Fälle wieder.

Bei allen Graphen gilt, dass für die elektrischen Wände und PMLs De-

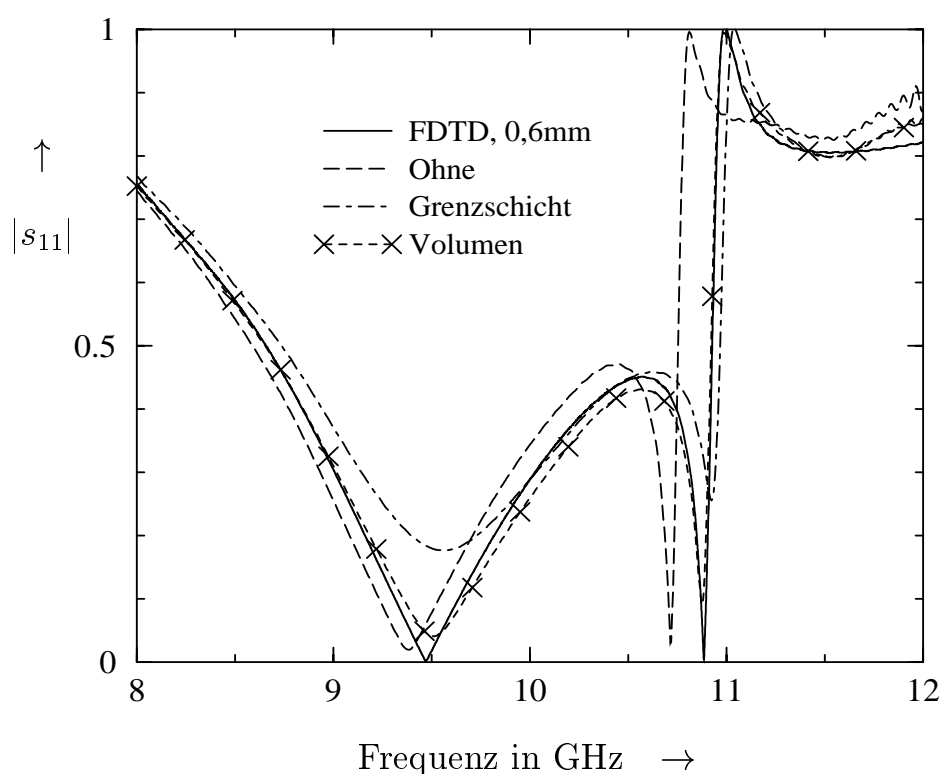


Abbildung 8.7: Detailbereiche bei dielektrischen Grenzschichten am Beispiel des dielektrischen Pfostens aus Abbildung 8.2. „FDTD, 0,6 mm“ : Referenzgraph aus Abbildung 8.3. „Ohne“ : Für den Pfosten ist kein Detailbereich angegeben. „Grenzschicht“ : Die Grenzschicht Luft-Pfosten wird mit einem Detailbereich versehen. „Volumen“ : Für den gesamten Pfosten wird ein Detailbereich angegeben.

tailbereiche definiert waren. Die WT wurde in x- und z-Richtung mit je einer Stufe ( $\mathbf{j} = (j_x, j_y, j_z)^T = (1, 0, 1)^T$ ) und mit Haar-Funktionen durchgeführt. Ohne einen Detailbereich beim dielektrischen Pfosten treten erkennbar Frequenzverschiebungen auf, die denen aus Abbildung 8.3 für die Diskretisierungsweite von 1,2 mm sehr ähnlich sind. Das war zu erwarten, da die einstufige WT in x- und z-Richtung einer Halbierung der räumlichen Auflösung bzw. Verdopplung der Diskretisierungsweite von 0,6 mm auf 1,2 mm entspricht - nur in den Gebieten mit Detailbereichen bleibt die ursprüngliche räumliche Auflösung erhalten.

Wird zumindest die Grenzfläche unkomprimiert transformiert, so verschwinden die Frequenzabweichungen, aber der Nulldurchgang bei etwa 9,5 GHz wird nur sehr unzureichend approximiert. Ist jedoch der Pfo-

sten vollständig durch einen Detailbereich abgedeckt, verbessert sich die Genauigkeit im Bereich dieses Nulldurchgangs deutlich.

### 8.2.1.2 Operatorimplementierungen und Simulationszeiten

Die Abschnitte 7.3.4 und 7.3.5 stellten zwei Möglichkeiten dar, die WT von Operatoren für bestimmte Bereiche zu optimieren bzw. auf diese Bereiche zu begrenzen. In Abschnitt 7.3.4 werden dazu optimierte Anregungs- und Ausgabeoperatoren vorgestellt, die Felder lediglich in bestimmten Bereichen vom Grundraum in den Waveletbereich oder umgekehrt transformieren. Das ist im Gegensatz zu der vollständigen Transformation eines Feldes zu sehen. Abschnitt 7.3.5 behandelt die Idee, dass Operatoren nur für ein zuvor spezifiziertes Volumen in den Waveletbereich transformiert werden. Im übrigen Bereich arbeiten die Operatoren dann nach wie vor im Grundraum. Dadurch können solche Bereiche von einer WT ausgenommen werden, die sich nur mit hohem numerischem Aufwand im Waveletbereich darstellen lassen. Der Einfluss dieser Verfahren auf Simulationszeiten soll hier dargestellt werden.

Beginnen wir mit der Teilbereichs-WT. Von dem gesamten in Abbildung 8.4 gezeigten Wellenleiter soll nur der normale Feldraum einer WT unterzogen werden, d. h. die PMLs verbleiben im Grundraum. Elektrische Wände, die dielektrische Pfofen sowie eine zwei Zellen breite Schicht am Anfang und am Ende des Feldraumes werden für das Detailgitter-Verfahren vorgesehen. Im Vergleich dazu steht die Transformation des gesamten Wellenleiters, bei dem für die PMLs ebenfalls ein Detailbereich angegeben wird.

In Tabelle 8.4 sind für verschiedene Fälle die Simulationszeiten aufgeführt. Die Graphen für  $|s_{11}|$  sind innerhalb einer Zeile, d. h. bei der Verwendung gleicher Transformationsparameter, nicht voneinander zu unterscheiden. Der genaue Verlauf der Graphen wird erst in Abschnitt 8.2.1.3 von Bedeutung sein und ist daher hier noch nicht wiedergegeben.

Es ist in der Tabelle zu erkennen, dass die Anwendung der Teilbereichs-WT bereits im einfachsten Fall eine – wenn auch geringe – Reduktion der Simulationszeit mit sich bringt. Für den Fall  $N = j = 2$  beträgt die Abnahme sogar über 43%. Die Simulationszeit ist in diesem Fall zwar zu hoch (die Ursache hierfür wird in Abschnitt 8.2.1.3 näher beleuchtet),

$N$	$j$	Detailbereich	Teilbereichs-WT
1	1	10,13 s	9,50 s
1	2	16,70 s	13,82 s
2	1	23,06 s	18,32 s
2	2	240,50 s	135,38 s

Tabelle 8.4: Die Simulation mit transformierten PMLs („Detailbereich“) benötigt mehr Zeit als bei der Simulation mit einer Teilbereichs-WT, welche die PMLs im Grundraum belässt.  $N$  ist die Wavelet-Ordnung,  $\mathbf{j} = (j, 0, j)^T$  die Transformationstiefe.

dennoch verdeutlicht das Ergebnis eindrucksvoll, dass sich der numerische Aufwand stark inhomogener (Teil-)Operatoren durch eine WT drastisch erhöhen kann.

Der Einspareffekt fällt für  $N = j = 1$  sehr gering aus. Das liegt vor allem an dem Verhältnis zwischen Feldraum und PML-Raum. Der Feldraum wird entlang seiner Längsachse mit 300 Yee-Zellen diskretisiert, wogegen es die beiden PMLs zusammen auf lediglich 12 Zellen bringen (24 Zellen, berücksichtigt man die räumliche Aufteilung in Abbildung 7.2 auf Seite 112). In vielen praktischen Fällen jedoch haben PMLs einen höheren Anteil am gesamten Simulationsraum, sodass eine Einsparung durch eine Teilbereichs-WT deutlicher ausfallen dürfte.

Ebenfalls zu beachten ist, dass der Wellenleiter lediglich durch Flächen-PMLs abgeschlossen wird, welche nur in einer Richtung Inhomogenitäten aufweisen. Sind jedoch offene Strukturen zu simulieren, müssen diese auch mit Kanten- und Ecken-PMLs abgeschlossen werden, welche in zwei bzw. drei Richtungen inhomogen sind und hinsichtlich einer WT entsprechend problematischer sind.

Es bleibt jetzt noch der Einfluss optimierter Feldanregungen und -ausgaben zu untersuchen. Der Wellenleiter wurde in allen bisher durchgeführten Simulationen stets mit dem transversalen Feld des  $H_{10}$ -Typs am Eingang angeregt. Von der gesamten Feldverteilung wurde ebenso nur der Querschnittsbereich am Wellenleitereingang ausgegeben und dann zur Berechnung des Eingangsreflexionsfaktors weiter analysiert.

Die Simulationszeiten für die Teilbereichs-WT in Tabelle 8.4 wurden durch Optimierung der Anregungen und Ausgaben entsprechend Ab-

$N$	$j$	vollständige (I)WT	lokale (I)WT
1	1	125,86 s	9,50 s
1	2	268,48 s	13,82 s
2	1	163,08 s	18,32 s
2	2	433,13 s	135,38 s

Tabelle 8.5: Simulationszeiten bei vollständiger Transformation des Feldes (Spalte „vollständige (I)WT“) für Anregungen (WT) und Ausgaben (IWT) im Vergleich zur Anwendung optimierter Anregungs- und Ausgabeoperatoren (Spalte „lokale (I)WT“; vgl. auch Tabelle 8.4).  $N$  ist die Wavelet-Ordnung,  $\mathbf{j} = (j, 0, j)^T$  die Transformationstiefe.

schnitt 7.3.4 erzielt. Diese Zeiten sind zum direkten Vergleich auch in Tabelle 8.5 wiedergegeben (Spalte „lokale (I)WT“). Unter der Spalte „vollständige (I)WT“ zeigt sich in dieser Tabelle, wie die Simulationszeiten zunehmen, wenn das Feld für jede Anregung und Ausgabe vollständig transformiert werden muss.

Das offensichtliche Einsparpotenzial kommt dadurch zustande, dass der auszugebende bzw. anzuregende Bereich ein sehr kleines Volumen im Vergleich zum gesamten Simulationsraum einnimmt. In manchen Fällen ist die Ausgabe eines deutlich größeren oder sogar des vollständigen Volumens vonnöten. In der Praxis ist das aber nur für einige wenige Zeitpunkte erforderlich, beispielsweise um die Feldverteilung im Simulationsraum ab und zu visualisieren zu können. In diesen Fällen ist dann die vollständige WT bzw. IWT des Feldes durchzuführen. Eine permanente Ausgabe weniger Feldkomponenten lässt sich dagegen mit optimierten Ausgabeoperatoren sehr viel effizienter durchführen.

Eine weitere Option besteht manchmal darin, Anregungs- und Ausgabebereiche durch eine Teilbereichs-WT von der Transformation auszunehmen. Das wäre auch in diesem Fall möglich gewesen. Im Allgemeinen geht das jedoch nur dann, wenn sich die Bereiche am Rand des Simulationsgebietes befinden. Sind beispielsweise Spannungen an verschiedenen Stellen innerhalb einer Schaltung aufzuzeichnen, lässt sich diese Methode nicht mehr anwenden, wogegen die optimierte Ausgabe nach wie vor zur Verfügung steht.

### 8.2.1.3 Skalen und Wavelet-Ordnungen

In Abschnitt 8.1 wurde für die Matrix-basierte FDTD erwähnt, dass Kompressionsraten mit der Transformationstiefe zunehmen (zumindest in gewissen Grenzen). Die Adaptivität des Verfahrens erlaubt die Anwendung beliebiger Transformationstiefen, da Entwicklungskoeffizienten aus den Räumen der niedrigeren Skalen (höhere räumliche Auflösungen) in Abhängigkeit von der Regularität des Feldes automatisch berücksichtigt werden, um die Genauigkeit der Simulation sicherzustellen.

Das Detailgitter-Verfahren für die Listen-basierte FDTD dagegen, erzeugt eine Art analytisches Untergitterverfahren, wie bereits in Abschnitt 7.3.6.1 erwähnt wurde. Dieses Gitter ist von statischer Natur, was einen Geschwindigkeitsvorteil mit sich bringt, da der Aufwand für die Adaptivität entfällt. Auf der anderen Seite kann dadurch nicht mehr für den gesamten Simulationsraum die gleiche räumliche Auflösung erreicht werden. Diese Einschränkung ist eine inhärente Eigenschaft statischer Untergitterverfahren. Mit zunehmender Transformationstiefe werden die außerhalb eines Detailbereiches liegenden Gebiete gröber aufgelöst. Dadurch steigt die Kompression des Feldes, gleichzeitig nimmt aber die Ungenauigkeit durch die stärkere Approximation zu. Da keine adaptive Hinzunahme von Koeffizienten räumlich besser auflösender Funktionen, d. h. solche aus niedrigeren Skalen erfolgt, darf die Transformationstiefe nicht beliebig hoch angesetzt werden.

Aus den Untersuchungen zu PMLs ist hervorgegangen, dass diese sich nur ungünstig in den Waveletbereich transformieren lassen. Allgemein erhöht eine WT zunächst den numerischen Aufwand eines Operators. Diese Erhöhung wirkt der Beschleunigung durch die Kompression des Feldes entgegen.

Ein ähnlicher gegensätzlicher Effekt wie bei der Transformationstiefe ergibt sich für die Auswahl der Entwicklungsfunktionen, d. h. für die Auswahl der Ordnung der Daubechies-Funktionen. Mit zunehmender Ordnung werden die Funktionen glatter, andererseits steigen aber auch ihre Trägerbreite und die Anzahl der Skalierungs- und Waveletkoeffizienten (vgl. Abschnitt 3.2.3). Durch die höhere Trägerbreite werden Operatoren im Waveletbereich i. A. inhomogener, was sich aufgrund der Beschränkungen der gängigen Speicherkonzepte negativ auf die Ausführungsgeschwindigkeit auswirkt, siehe Abschnitt 6.1. Ferner reduziert sich mit steigenden

Trägerbreiten der Kompressionseffekt durch das Detailgitter-Verfahren, weil mehr Entwicklungsfunktionen mit ihrem Träger den Detailbereich überschneiden.

Fasst man die genannten Aspekte zusammen, ergibt sich die Erklärung für die in Abschnitt 8.2.1.2 beobachteten Zunahmen von Simulationszeiten für steigende Ordnungen und Transformationstiefen. Ohne aber an dieser Stelle näher auf Simulationszeiten eingehen zu wollen, sollen jetzt numerische Einflüsse auf Operatoren aufgezeigt werden.

In Abbildung 8.8 sind die Graphen aus der Simulation zu Tabelle 8.4 von Abschnitt 8.2.1.2 dargestellt. Ausgehend von dem einfachsten Fall mit  $N = j = 1$ , d. h. Haar-Funktionen und nur eine Transformationsstufe in  $x$ - und  $z$ -Richtung, zeigt sich eine Verbesserung der Genauigkeit durch Erhöhung der Wavelet-Ordnung auf  $N = 2$ . Die Vertiefung der Transformation von  $j = 1$  auf  $j = 2$  für  $N = 1$  senkt die Genauigkeit deutlich, was jedoch durch die Verwendung von  $D_2$ -Wavelets ( $N = 2$ ) vermieden werden kann.

Tabelle 8.6 gibt einen Einblick in den Umfang der numerischen Operationen der miteinander verketteten Rotations- und Materialoperatoren. Ebenfalls angegeben ist das Verhältnis des Aufwands der benutzten (transformierten und komprimierten) Operatoren zu dem der unkomprimierten Operatoren (transformiert, aber keine Detailbereiche).

Es fällt u. a. auf, dass bei dem beschriebenen Übergang von  $N = 1$  auf  $N = 2$  für  $j = 1$  die Komplexität der Operatoren deutlich steigt. Hier bestätigt sich die Zunahme der Entwicklungskoeffizienten (Zuweisungen)

$N$	$j$	Zuweisungen	Additionen	Multiplikationen
–	0	35239	58334	93573
1	1	12371 (35%)	31185 (20%)	43556 (23%)
1	2	9140 (25%)	60568 (24%)	69708 (24%)
2	1	16090 (42%)	161276 (39%)	177366 (39%)
2	2	17755 (42%)	939939 (67%)	957694 (66%)

Tabelle 8.6: Numerische Aufwände für die Simulationen nach Abbildung 8.8. In Klammern ist angegeben, wieviel Prozent das im Vergleich zu dem Fall ohne Kompression mit dem Detailgitter-Verfahren sind.  $N$  ist die Wavelet-Ordnung,  $\mathbf{j} = (j, 0, j)^T$  die Transformationstiefe.

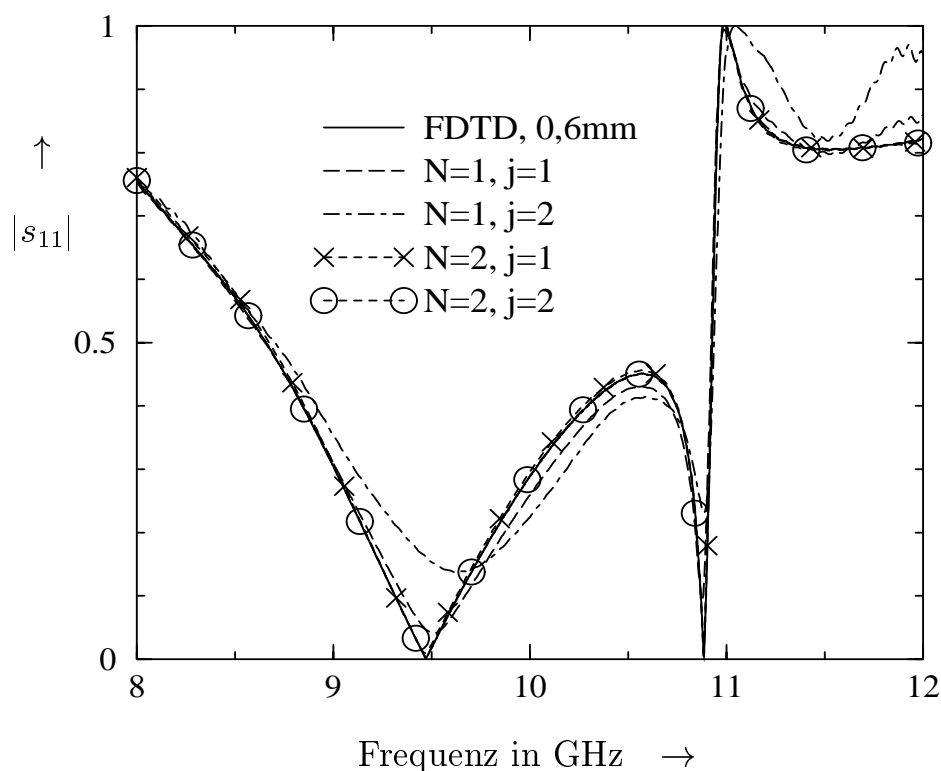


Abbildung 8.8: Graphen zu den Simulationen von Tabelle 8.4. „FDTD, 0,6 mm“ : Referenzgraph aus Abbildung 8.3.  $N$  ist die Wavelet-Ordnung,  $\mathbf{j} = (j, 0, j)^T$  die Transformationstiefe.

durch die höhere Trägerbreite. Noch stärker ist der Anstieg bei den Additionen und Multiplikationen. Anhand der Zahlen würde man wohl eine Zunahme der Rechenzeit auf mehr als das Vierfache erwarten, laut Tabelle 8.4 verdoppelt sie sich aber nur knapp. Das deutet darauf hin, dass ein hoher Prozentsatz der nun zahlreicheren Operationen mit Abbildungsdaten oder Feldwerten durchgeführt wird, die sich aus vorangegangenen Rechenschritten noch in einem schnellen Speicher befinden, vgl. Abschnitt 6.1.

In den ersten beiden Zeilen ( $N = 1$ ) zeichnet sich ab, wie die Erhöhung der Transformationstiefe Felder komprimiert, indem weniger Koeffizienten zu berechnen sind. Dennoch steigt auch die Anzahl der Additionen und Multiplikationen. Das bedeutet, dass die Abhängigkeiten der Entwicklungskoeffizienten untereinander zunehmen. Es ist zu vermuten, dass für jede zu simulierende Struktur eine obere Grenze für die Transformationstiefe existiert, oberhalb derer solche Abhängigkeiten zunehmen.

Ganz besonders deutlich wird dies beim Übergang von  $j = 1$  auf  $j = 2$  für  $N = 2$ . Hier steigen nicht nur die Anzahlen an Additionen und Multiplikationen, sondern auch die Zuweisungen nehmen wieder zu.

#### 8.2.1.4 Abschließender Vergleich

Bisher wurde der Wellenleiter jeweils mit einigen wenigen Parametern simuliert, um die generellen Einflüsse dieser Parameter und der angewendeten Verfahren untersuchen zu können. Tabelle 8.7 stellt die Ergebnisse einer vollständigeren Variation über die Wavelet-Ordnung und die Transformationstiefen vor.

Von den verwertbaren Ergebnissen sind lediglich drei Kombinationen aus  $N$ ,  $j_x$  und  $j_z$  ein wenig schneller als die reine FDTD-Simulation. Für  $N = 1$  ergeben sich zu hohe Ungenauigkeiten, sobald die WT in wenigstens einer Richtung zweistufig erfolgt. Mit  $N = 2$  erhält man dagegen durchweg Ergebnisse hoher Genauigkeit. Verwendet man jedoch Daubechies-Wavelets dritter Ordnung, können instabile Simulationen auftreten. Nicht in der Tabelle dargestellt sind Versuche mit einer dreistufigen WT in wenigstens einer Richtung. In diesem Fall sind ausschließlich instabile Simulationen zustande gekommen.

Auf Erklärungen für die geringen Möglichkeiten zur Beschleunigung der Simulation dieses Wellenleiters sowie Verbesserungsmöglichkeiten werden wir in Abschnitt 8.2.3 zu sprechen kommen. Was die Instabilitäten hervorruft kann nur vermutet werden. Da die PMLs durch die Teilbereichs-WT außerhalb des Wavelet-Bereiches liegen, können sie allenfalls indirekt an der Entstehung der Instabilitäten beteiligt sein (vgl. auch Abschnitt 8.2.1.1).

Die Grenze zwischen Feldraum und PML ist mit einem zwei Zellen breiten Detailbereich versehen worden. Dieser soll sicherstellen, dass die Felder dort möglichst mit der gleichen Genauigkeit berechnet werden wie bei einer untransformierten Simulation. Mit zunehmender Wavelet-Ordnung und/oder Transformationstiefe werden die Träger der Funktionen in den einzelnen Skalen zunehmend breiter und übersteigen teilweise deutlich die Breite des Detailbereiches. Dies war auch in einigen früheren Simulationen der Fall. Vermutlich gibt es jedoch eine Grenze, oberhalb derer

$N$	$j_x$	$j_z$	Zuweis.	Addit.	Multipl.	Zeit/s	Genauigkeit
–	0	0	35239	58334	93573	11,2	
1	0	1	19076	33351	52427	10,4	+
1	0	2	11088	21604	32692	8,3	–
1	1	0	20293	37861	58154	11,5	+
1	1	1	12371	31185	43556	9,1	+
1	1	2	8550	30284	38834	9,9	–
1	2	0	14335	38532	52867	10,1	–
1	2	1	10764	49530	60294	9,5	–
1	2	2	9140	60568	69708	13,6	–
2	0	1	19376	59882	79258	12,4	+
2	0	2	11801	54817	66618	10,8	+
2	1	0	23029	92077	115106	14,7	+
2	1	1	16090	161276	177366	18,2	+
2	1	2	13021	252425	265446	29,6	+
2	2	0	20665	202225	222890	21,7	+
2	2	1	18094	532802	550896	51,1	+
2	2	2	17239	910693	927932	121,0	+
3	0	1	19676	89809	109485	13,2	+
3	0	2	12364	96426	108790	13,9	xx
3	1	0	25765	175941	201706	20,7	+
3	1	1	19702	518702	538404	52,2	+
3	1	2	17242	929282	946524	212,0	xx
3	2	0	26005	463278	489283	42,8	+
3	2	1	24709	1861289	1885998	236,7	+
3	2	2	24631	3357686	3382317	1005,0	+

Tabelle 8.7: Weitere Ergebnisse zur Simulation des Wellenleiters. Verfahren: Analog zu Abbildung 8.8 (Detailbereiche für elektrische Wände und dielektrischen Pfosten; Teilbereichs-WT mit dem normalen Feldraum, d. h. ohne PML; Tensorprodukt-Basen). Erste Zeile der Tabelle : untransformierte FDTD-Simulation. Genauigkeit : „+“=geringfügige Abweichung von der untransformierten FDTD-Simulation (z. B. Graph „ $N = 1, j = 1$ “ in Abbildung 8.8); „–“=stärkere Abweichung; „xx“=instabile Simulation.  $N$  ist die Wavelet-Ordnung,  $j_\nu$  ist die Transformationstiefe in Richtung  $\nu$ .

Effekte im Zusammenhang mit Detailbereichen auftreten, die zu instabilen Wellenausbreitungen führen. Ob es dabei um die Detailbereiche an der Grenze PML–Feldraum oder um einen der anderen Detailbereiche geht, kann nicht ohne Weiteres festgestellt werden.

## 8.2.2 Mikrostreifenleitungsfiler

Der Rechteckhohlleiter mit der dielektrischen Diskontinuität aus Abschnitt 8.2.1 konnte mit zweidimensionalen Simulationen analysiert werden. Abbildung 8.9 zeigt einen Tiefpass in Mikrostreifenleitungstechnik, welcher dreidimensionale Simulationen erfordert. Dieser Filter wurde auch bereits in [4] und [37] mit verschiedenen Methoden simuliert.

Als Feldraum wurde ein quaderförmiges Volumen mit den Abmessungen  $l_x \approx 34,04$  mm,  $l_y \approx 57,57$  mm,  $l_z \approx 4,23$  mm gewählt. Die räumlichen Diskretisierungsschrittweiten betragen  $\Delta x \in [0,402, 0,404]$  mm,  $\Delta y = 0,42\bar{3}$  mm,  $\Delta z = 0,264\bar{6}$  mm. Der Feldraum wurde an 5 Seiten mit sechs

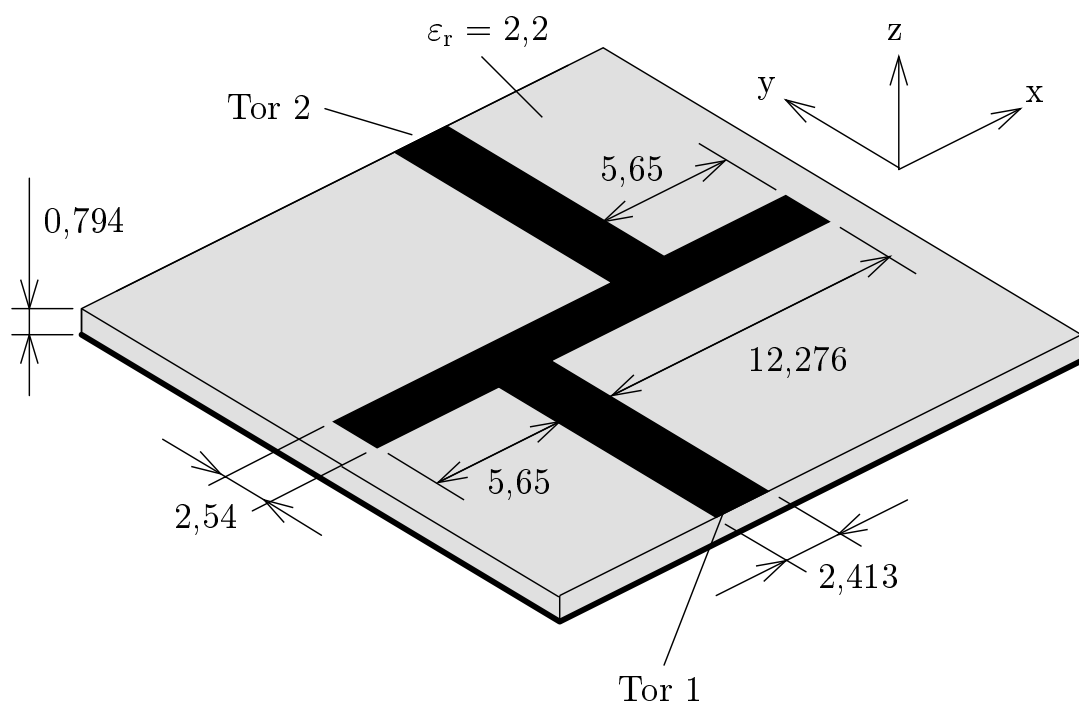


Abbildung 8.9: Mikrostreifenleitungsfiler auf RT/Duroid Substrat mit der Permittivitätszahl  $\varepsilon_r = 2,2$ . Alle Maßangaben sind in Millimeter.

Zellen tiefen PMLs abgeschlossen. Simuliert wurde ein Zeitraum von etwa 1,764 ns.

Die WT der FDTD-Operatoren wurde wieder mit einer Teilbereichs-WT auf den Feldraum beschränkt, d. h. die PMLs blieben untransformiert. Um den numerischen Einfluss der WT besser untersuchen zu können, wurde die Rechenzeit für die PMLs ermittelt und aus den Simulationszeiten herausgerechnet.

Der Leiter auf dem Substrat wurde nach allen Seiten hin mit einem drei Zellen dicken Detailbereich umgeben (in  $z$ -Richtung ist der Detailbereich damit überall insgesamt sechs Zellen dick). Ferner wurden die Grundmetallisierung und die Grenze zwischen dem Raum der Teilbereichs-WT und dem übrigen Feldraum mit einem Detailbereich maskiert, der jeweils nur eine Zelle dick ist.

In Tabelle 8.8 werden die Ergebnisse für verschiedene Simulationsparameter aufgeführt. Die Spalte „ $S_1/s$ “ gibt die Simulationszeiten für den Fall wieder, dass jeder Rotationsoperator als Gesamtoperator transformiert wird (wie es in bisherigen Simulationen immer der Fall war).

Im Vergleich dazu enthält die Spalte „ $S_2/s$ “ die Rechenzeiten für die Simulationen, bei denen die Rotationsoperatoren durch einzelne partielle Ableitungsoperatoren dargestellt wurden. In Abschnitt 8.2.1 wurde bereits die Frage gestellt, ob die Transformation separater kartesischer Operatoren günstigere Eigenschaften aufweist. Durch diese Untersuchung kann ein weiterer Vergleich zur Matrix-basierten WT der FDTD gezogen werden, die ja ebenfalls die Ableitungen in den drei Raumrichtungen getrennt voneinander vornimmt. Dennoch sollte im Auge behalten werden, dass die partiellen Ableitungsoperatoren in der Listen-basierten WT der FDTD nicht in  $U_0^\nu$ -Operatoren zerlegt werden müssen.

Der Tabelle ist zu entnehmen, dass die Separation der Rotationsoperatoren in den meisten Fällen die Simulationszeit erhöht. Nur bei den zweistufigen Transformationen ergeben sich (teilweise sehr deutliche) Geschwindigkeitsvorteile. Vermutlich zahlt sich in diesem Bereich die einfachere Struktur der Ableitungsoperatoren aus, sodass sich der numerische Aufwand von Inhomogenitäten bei der WT nicht ganz so schnell erhöht. Bei den einstufigen Transformationen dagegen macht sich dieser Vorteil noch nicht bemerkbar. Hier dominieren vorwiegend die zusätzlichen Speicherzugriffe: Für jeden Teiloperator muss das Urbild-Feld erneut ausgewertet

$N$	$j_x$	$j_y$	$j_z$	Basis	$S_1/s$	$S_2/s$	Genauigkeit
–	0	0	0	–	225,9	–	
1	1	0	0	hom.	188,3	249,2	+
1	0	1	0	hom.	180,0	227,9	+
1	0	0	1	hom.	211,3	277,8	+
1	1	1	0	hom.	175,8	213,5	+
2	1	1	0	hom.	357,8	359,5	+
1	1	1	1	hom./sep.	262,6	280,9	+
2	1	1	1	hom./sep.	884,8	– <sup>(*)</sup>	+
1	2	2	0	hom.	353,9	271,5	–
2	2	2	0	hom.	877,8	733,4	+
1	2	2	2	hom.	721,6	482,5	+/-
1	2	2	2	sep.	563,8	451,7	+

Tabelle 8.8: Simulationszeiten (abzüglich der Rechenzeit für die PMLs) für den Filter aus Abbildung 8.9. Die oberste Zeile gehört zur Simulation mit der untransformierten FDTD.  $N$  ist die Wavelet-Ordnung,  $j_\nu$  ist die Transformationstiefe in Richtung  $\nu$ . „hom.“ : Homogene Tensorprodukt-Basis; „sep.“ : Separable MSA-Basis; siehe Kapitel 4.  $S_1$  : Simulationszeit bei nicht-separierten Rotationsoperatoren.  $S_2$  : Simulationszeit bei separierten Rotationsoperatoren. Genauigkeit : „+“ = geringfügige Abweichung von der untransformierten FDTD-Simulation; „–“ = stärkere Abweichung (Graphen der Abbildungen 8.10 und 8.11). <sup>(\*)</sup> : Konnte wegen Speicherbegrenzung nicht durchgeführt werden.

bzw. aus dem RAM in schnellere Speicherbereiche geladen werden. Dadurch erhöhen sich die Simulationszeiten soweit, dass sich die Aufteilung des Rotationsoperators als nachteilig erweist.

In den weiteren Betrachtungen konzentrieren wir uns daher wieder auf die Spalte „ $S_1/s$ “. Hier zeigt sich in lediglich den vier einfachsten Fällen, d. h. WT mit Haar-Funktionen und nur eine Transformationsstufe in ein oder zwei Richtungen, dass die Simulationszeit der FDTD unterboten werden kann.

Die Transformation um zwei Stufen in x- und y-Richtung führt ebenfalls zu einer Erhöhung der Rechenzeit. Dass hierbei das Feld stärker komprimiert werden konnte, lässt sich an der Abnahme der Genauigkeit

erkennen. Die Verwendung der glatteren  $D_2$ -Funktionen führt dagegen wieder zu sehr genauen Ergebnissen, allerdings auf Kosten einer noch weiter erhöhten Rechenzeit. Die Graphen beider Fälle sind in Abbildung 8.10 zu sehen.

Die Kompression eines Feldes bzw. Feldraumes bedeutet scheinbar nicht auch gleichzeitig die Kompression einer darauf operierenden Abbildung, wie beispielsweise dem Rotationsoperator. Das gilt auch für den Fall der zweistufigen Transformation in alle drei Raumrichtungen. Ein interessanter Effekt ist hier aber zu beobachten: Wird eine separable MSA-Basis verwendet, ergeben sich eine kürzere Rechenzeit und genauere Resultate als bei der Benutzung einer homogenen Tensorprodukt-Basis. Auf eine mögliche Erklärung hierfür wird in Abschnitt 8.2.3.5 eingegangen. Der Genauigkeitsunterschied kann in Abbildung 8.11 begutachtet werden.

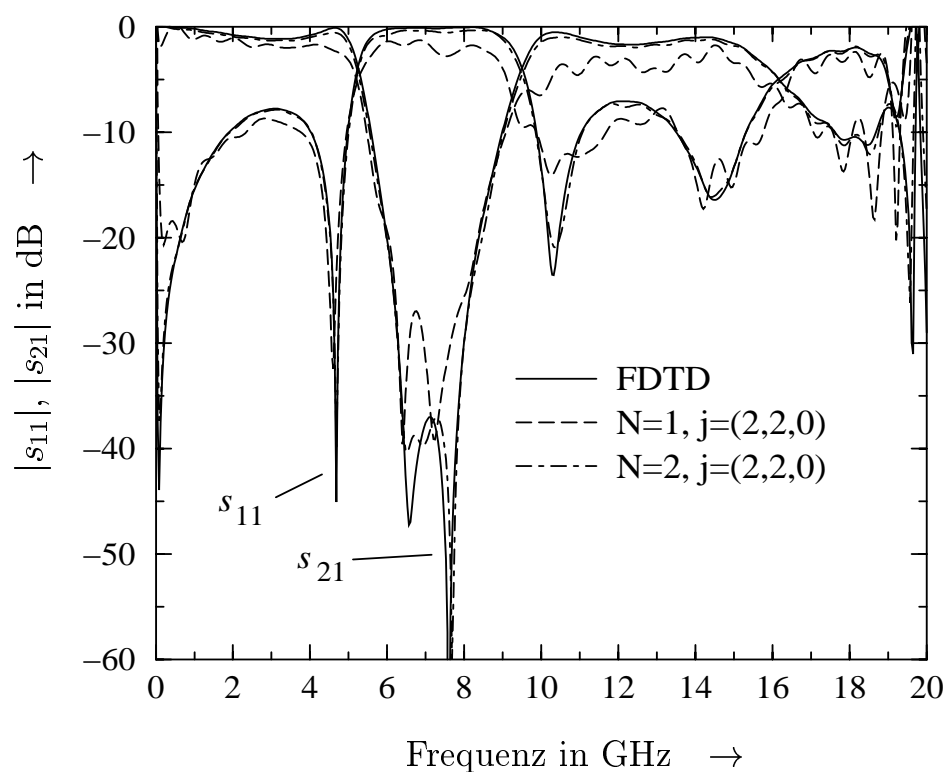


Abbildung 8.10: Streuparameter aus Simulationen des Filters nach Abbildung 8.9. Zweistufige WT in x- und y-Richtung ( $\mathbf{j} = (2, 2, 0)^T$ ).  $N$  ist die Wavelet-Ordnung. WT mit homogenen Tensorprodukt-Basen.

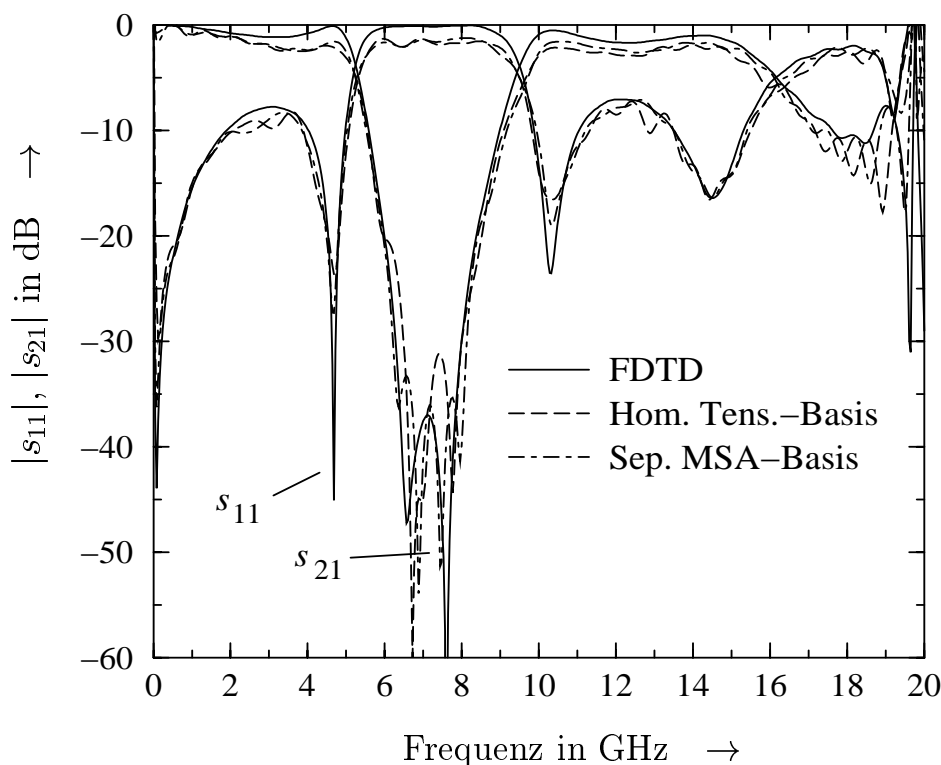


Abbildung 8.11: Streuparameter aus Simulationen des Filters nach Abbildung 8.9. Zweistufige WT in allen Raumrichtungen ( $\mathbf{j} = (2, 2, 2)^T$ ) mit Haar-Funktionen. Die WT wurde einmal mit einer homogenen Tensorprodukt-Basis und einmal mit einer separablen MSA-Basis durchgeführt.

## 8.2.3 Analysen

### 8.2.3.1 Allgemeine Betrachtungen

Die beiden simulierten Beispiele haben gezeigt, dass Reduktionen der Rechenzeiten gegenüber einer reinen FDTD möglich sind. Allerdings fallen die zeitlichen Einsparungen deutlich geringer aus, als man es von Bildkompressionen mit Wavelets erwarten würde. Zwei mögliche Erklärungen für dieses Phänomen bieten sich an:

1. Felder werden zwar komprimiert, doch der numerische Aufwand von Operatoren steigt mit der Transformation in den Waveletbereich überproportional an.

2. Die gewählten Datenstrukturen sind für Wavelet-transformierte Operatoren ungeeignet. Dadurch sinkt der Anteil optimierter Speicherzugriffe drastisch, was zu erhöhten Wartezyklen in der CPU führt, vgl. Abschnitt 6.1.

Aus Tabelle 8.7 auf Seite 163 lassen sich bereits einige Anhaltspunkte für die Bewertung der beiden Möglichkeiten ableiten. Für  $N = 1$  bleiben die numerischen Aufwände in der gleichen Größenordnung. Eine überproportionale Zunahme der Rechenoperationen ist nicht zu beobachten. Das trifft teilweise auch auf  $N = 2$  und  $N = 3$  zu. Die Aufwände steigen dort vor allem dann schnell an, wenn in zwei Richtungen transformiert wird oder die WT in einer Richtung zweistufig durchgeführt wird. Allerdings liegt die Rechenzeit bereits vor dem Erreichen dieser Parameterbereiche jenseits der Simulationszeit der FDTD. Der überproportionale Anstieg von Rechenoperationen lässt sich somit zwar beobachten, kann aber zumindest im Fall des Rechteckhohlleiters mit dem dielektrischen Pfosten nicht als Erklärung für die geringen Einsparungen bei den Rechenzeiten dienen.

Der zweite Erklärungsversuch wird durch Tabelle 8.7 eher bestätigt: Bei  $N = 1$  liegen die Anzahlen sämtlicher Operationen bis auf eine Ausnahme unterhalb derer der FDTD. Auch bei  $N = 2$  ist das in einem Fall erfüllt, in einem weiteren ist lediglich die Anzahl an Additionen ein wenig höher. Aufschlussreich ist ferner die dritte Zeile der Tabelle (Parameter  $N = 1$ ,  $j_x = 0$ ,  $j_z = 2$ ). Die Genauigkeit ist in diesem Fall zwar gering, dennoch eignet sich dieses Beispiel sehr gut zur Verdeutlichung einer Diskrepanz: Die Angaben bei Zuweisungen, Additionen und Multiplikationen liegen zwischen 31% und 35% im Vergleich zur FDTD. Dennoch ist das Verhältnis der Rechenzeiten mit etwa 74% mehr als doppelt so hoch.

Bevor wir uns weiter mit den Simulationsergebnissen befassen, soll kurz auf einen Zusammenhang zwischen Raumdimensionen und Kompressionsraten eingegangen werden. Die durchschnittliche Kompressionsrate einer Werteverteilung (z. B. gemessenes Signal, Bild, elektromagnetisches Feld) steigt im Normalfall mit der Dimension des Trägers. So lässt sich beispielsweise ein zweidimensionales Bild i. d. R. stärker mit Wavelets komprimieren als eine (eindimensionale) Messkurve. Analog sollten Daten eines dreidimensionalen Raumes besser zu komprimieren sein als zweidimensionale Bilder. Das hängt natürlich stets von der Art der Daten ab und setzt ferner die Durchführung der WT in alle Richtungen voraus.

Um den Grund für dieses Phänomen besser erklären zu können, betrachten wir der Einfachheit halber stückweise konstante Werteverteilungen, wobei es nur wenige dieser homogenen Bereiche geben soll. Die homogenen Bereiche lassen sich dann mit verhältnismäßig wenigen Skalierungsfunktionen aus dem Raum der höchsten Skala darstellen. Die Anzahl an Skalierungsfunktionen ist deshalb gering, weil die Funktionen in diesem Raum einen größeren Träger besitzen als die Funktionen im  $V_0$ . Lediglich in der Umgebung der Unstetigkeiten werden Wavelets der niedrigeren Skalen benötigt, um die erforderliche höhere räumliche Auflösung erreichen zu können. Die Kompressionsrate hängt damit vorwiegend von den homogenen Bereichen ab. Kann man für eine Richtung innerhalb eines Bereiches angeben, dass sich die Anzahl der Entwicklungsfunktionen um den Faktor  $r$  ( $0 < r < 1$ ) reduziert, so ergibt das bei  $n$  Dimensionen eine Kompressionsrate von  $r^n$ . Wegen  $0 < r < 1$  gilt dann  $r^n < r^{n-1}$ , d. h. in höherdimensionalen Räumen ist eine bessere Kompressionsrate zu erwarten.

Die Gültigkeit dieser Aussage ist streng genommen von diversen Voraussetzungen abhängig, auf die hier aber nicht näher eingegangen werden soll. Nach dem bisher Gesagten wäre jedenfalls zu erwarten gewesen, dass die dreidimensionale Simulation des Mikrostreifenleitungsfilters bezüglich des Rechenzeitverhältnisses besser abschneidet als die zweidimensionale Simulation des Rechteckhohlleiters. Das Ausbleiben dieser Beschleunigung ist ein weiterer Hinweis auf grundsätzliche Probleme bei der Implementierung der transformierten Operatoren. Dieser Hinweis soll durch den Vergleich einiger Eigenschaften untransformierter und transformierter FDTD-Operatoren im folgenden Abschnitt näher beleuchtet werden.

### 8.2.3.2 Vektor- und Faktorfolgen von $O^3$ -Listen

Untransformierte FDTD-Operatoren haben eine einfache, sehr übersichtliche Struktur, die sich wegen der überwiegend homogenen Bereiche auch kompakt darstellen lässt. Innerhalb eines homogenen Bereiches gibt es nur sehr wenige unterschiedliche Faktoren, mit denen Feldkoeffizienten multipliziert werden. Nachdem die erste Yee-Zelle berechnet wurde, müssen die Faktoren nicht mehr aus dem RAM in den L1-Cache bzw. prozessorinterne Register kopiert werden, da sie für die restlichen Yee-Zellen des Bereiches identisch sind. In [4] wird demonstriert, welche Beschleunigung eine

Simulation durch die Einsparung solcher RAM-Zugriffe erfahren kann.

FDTD- und Wavelet-Operatoren werden in dieser Arbeit unter anderem durch Listen dargestellt, um Verkettungen berechnen zu können und so Simulationsoperatoren für den Waveletbereich zu erhalten. Ein paar grundlegende Begriffe der Listen-basierten Darstellung sollen kurz wiederholt werden. Bei skalaren Feldern ist die Abbildung auf einen einzelnen Bildpunkt durch genau eine  $O^2$ -Liste bestimmt, welche aus Faktor- und Vektorfolgen besteht (vgl. Abschnitt 5.3.1). Im Falle vektorieller Felder kann mehr als nur eine  $O^2$ -Liste zur Berechnung eines Bildpunktes erforderlich sein. Aus den  $O^2$ -Listen aller Bildpunkte zusammen wird eine  $O^3$ -Liste gebildet, in der jede  $O^2$ -Liste nur einmal enthalten ist.

Durch die Herausnahme von PMLs aus den Transformationsbereichen konnten Simulationen bereits beschleunigt werden. Es drängt sich jedoch die Vermutung auf, dass die Anzahl von  $O^2$ -Listen dennoch so hoch ist, dass diese immer wieder aus dem RAM geladen werden müssen. Eine weitere Reduktion der Rechenzeit wäre in diesem Fall nur dann zu erreichen, wenn die Abbildungsinformationen aus den  $O^2$ -Listen kompakter abgespeichert werden können, damit sich die Anzahl an RAM-Zugriffen verringert. Dazu sollen nun die  $O^2$ -Listen aus der Simulation des Mikrostreifenleitungsfilters für zwei Fälle auf Redundanzen untersucht werden.

Wird eine  $O^3$ -Liste in zwei separate Listen aus Faktor- und Vektorfolgen aufgeteilt, können diese getrennt auf mehrfach vorkommende Elemente untersucht werden. Tabelle 8.9 zeigt das Ergebnis dieser Untersuchung. So ist beispielsweise zu erkennen, dass die 25264  $O^2$ -Listen des  $\tilde{\mathcal{A}}_2$ -Operators der Wavelet-transformierten FDTD lediglich 557 unterschiedliche Faktorfolgen enthalten.

Weiterhin wurde das Auftreten negativ-doppelter Folgen untersucht: Ist  $(c_k)_{k \in \mathbb{Z}}$  eine Folge von Faktoren, so ist  $(-c_k)_{k \in \mathbb{Z}}$  das negative Gegenstück dazu. Treten beide in einer  $O^3$ -Liste auf, werden die  $(-c_k)_{k \in \mathbb{Z}}$  als negativ-doppelte Faktorfolgen gezählt. Analoges gilt für Vektorfolgen. Es fällt auf, dass in keinem Fall negativ-doppelte Vektorfolgen zu beobachten sind. Bei den Faktorfolgen treten negative Duplikate lediglich für die Operatoren  $\mathcal{A}_2$  und  $\tilde{\mathcal{A}}_2$  auf. Eine Begründung für diese Asymmetrien kann jedoch nicht angegeben werden.

Während sich der Speicherbedarf für Faktoren und damit auch die zugehörigen Zugriffszeiten offensichtlich drastisch reduzieren lassen, fällt das

	FDTD		WT-FDTD	
	$\mathcal{A}_2$	$\mathcal{A}_4$	$\tilde{\mathcal{A}}_2$	$\tilde{\mathcal{A}}_4$
$O^2$ -Listen	1242	707	25264	22972
Vektorfolgen	275	261	6267	11307
ND-Vektorfolgen	0	0	0	0
Faktorfolgen	315	125	557	254
ND-Faktorfolgen	117	0	219	0

Tabelle 8.9: Die Aufteilung von  $O^2$ -Listen in zwei separate Listen mit Vektor- und Faktorfolgen gibt Auskunft über Redundanzen in  $O^2$ -Listen. Angegeben sind die Anzahlen verschiedener Folgen zu den  $O^2$ -Listen eines Operators. WT-FDTD : Siehe Tabelle 8.8 auf Seite 166,  $N = 1$ ,  $\mathbf{j} = (1, 1, 0)^T$ .  $\mathcal{A}_{2,4}$ ,  $\tilde{\mathcal{A}}_{2,4}$  : Verkettete Skalierungs- und Rotationsoperatoren, siehe Gleichungen (7.57) und (7.58) auf Seite 123. ND-Folgen : negativ-doppelte Folgen (siehe Text).

Ausmaß an Einsparungen bei den Vektorfolgen i. A. deutlich geringer aus. Statt nach Duplikaten ist also nach anderen Arten von Redundanzen Ausschau zu halten. Dazu sollen diese sortiert werden.

Tabelle 8.10a gibt einen Einblick in das Ergebnis der Sortierung für den Operator  $\tilde{\mathcal{A}}_2$  aus Tabelle 8.9. Jede Zeile enthält die Vektorfolge einer  $O^2$ -Liste. Schreibt man alle Vektorfolgen untereinander auf, so entsteht eine Tabelle aus ganzen (skalaren) Zahlen, die zeilenweise sortiert werden kann. Von der linken zur rechten Spalte hin nimmt die Priorität beim Sortieren ab.

Die in Tabelle 8.10a wiedergegebenen Auszüge aus der sortierten Tabelle zeigen sehr regelmäßige Muster. Es konnte festgestellt werden, dass diese Muster in über 98% der Tabelle zu finden sind. Hierzu gehören auch Vektorfolgen, die aus weniger als sechs Vektoren bestehen.

Betrachtet man ausschließlich die Beispiele in Tabelle 8.10a, so könnte man die Redundanzen verringern, indem die konstante Vektor-Teilfolge  $((0, 0, 0), (0, 0, 6))$  abgespalten und gesondert behandelt wird (formal entspricht dies der Aufteilung des Operators  $\tilde{\mathcal{A}}_2$  in zwei Teiloperatoren). Die Abspaltung würde den Speicherverbrauch allerdings lediglich um ein Drittel reduzieren.

a)	(-23, 0, -1),	(41, 0, -1),	(-23, 64, -1),	(41, 64, -1),	(0, 0, 0),	(0, 0, 6)
	(-23, -1, -1),	(41, -1, -1),	(-23, 63, -1),	(41, 63, -1),	(0, 0, 0),	(0, 0, 6)
	(-23, -2, -1),	(41, -2, -1),	(-23, 62, -1),	(41, 62, -1),	(0, 0, 0),	(0, 0, 6)
	(-23, -3, -1),	(41, -3, -1),	(-23, 61, -1),	(41, 61, -1),	(0, 0, 0),	(0, 0, 6)
		⋮				
	(-23, -63, -1),	(41, -63, -1),	(-23, 1, -1),	(41, 1, -1),	(0, 0, 0),	(0, 0, 6)
	(-23, -64, -1),	(41, -64, -1),	(-23, 0, -1),	(41, 0, -1),	(0, 0, 0),	(0, 0, 6)
	(-24, 0, -1),	(40, 0, -1),	(-24, 64, -1),	(40, 64, -1),	(0, 0, 0),	(0, 0, 6)
	(-24, -1, -1),	(40, -1, -1),	(-24, 63, -1),	(40, 63, -1),	(0, 0, 0),	(0, 0, 6)
	(-24, -2, -1),	(40, -2, -1),	(-24, 62, -1),	(40, 62, -1),	(0, 0, 0),	(0, 0, 6)
	(-24, -3, -1),	(40, -3, -1),	(-24, 61, -1),	(40, 61, -1),	(0, 0, 0),	(0, 0, 6)
		⋮				
	(-24, -63, -1),	(40, -63, -1),	(-24, 1, -1),	(40, 1, -1),	(0, 0, 0),	(0, 0, 6)
	(-24, -64, -1),	(40, -64, -1),	(-24, 0, -1),	(40, 0, -1),	(0, 0, 0),	(0, 0, 6)
b)	(-23, 0, -1),	(41, 0, -1),	(-23, 64, -1),	(41, 64, -1),	(0, 0, 0),	(0, 0, 6)
	( 0, -1, 0),	( 0, -1, 0),	( 0, -1, 0),	( 0, -1, 0),	(0, 0, 0),	(0, 0, 0)
	(-24, 0, -1),	(40, 0, -1),	(-24, 64, -1),	(40, 64, -1),	(0, 0, 0),	(0, 0, 6)
	( 0, -1, 0),	( 0, -1, 0),	( 0, -1, 0),	( 0, -1, 0),	(0, 0, 0),	(0, 0, 0)

Tabelle 8.10: a) Zwei sortierte Auszüge zum Operator  $\tilde{\mathcal{A}}_2$  aus Tabelle 8.9: Jede Zeile enthält die relativen Vektoren einer  $O^2$ -Liste. Zu weiteren Mustern siehe Angaben im Text. b) Reduktion der Vektorfolgen durch Delta-Vektorfolgen: Jede Vektorfolge aus a) lässt sich als die Summe von einer Haupt-Vektorfolge (erste bzw. dritte Zeile in b) ) und dem ganzzahlig Vielfachen einer Delta-Vektorfolge (zweite bzw. vierte Zeile in b) ) darstellen.

Eine effizientere und vermutlich auch allgemeinere Möglichkeit bietet die Einführung von Haupt- und Delta-Vektorfolgen. Zu den beiden Beispielen aus Teil a) der Tabelle 8.10 sind diese Vektorfolgen in Teil b) angegeben. Aufeinander folgende Vektorfolgen aus a) unterscheiden sich um die Delta-Vektorfolge aus b). Jeder Vektor aus a) lässt sich demnach durch den Hauptvektor aus b) und einem ganzzahligen Vielfachen des zugehörigen Delta-Vektors angeben. Dadurch lässt sich die Tabelle sehr stark komprimieren. Die beiden Delta-Vektorfolgen in Tabelle 8.10b sind sogar identisch. Eine weitere Analyse der vollständigen Tabelle hat ergeben, dass es insgesamt nur wenige unterschiedliche Delta-Vektorfolgen gibt, sodass hier ein weiteres Einsparpotenzial vorliegt.

Algorithmisch betrachtet entsteht durch die Einführung der Haupt- und Delta-Vektorfolgen ein Mehraufwand, da die benötigten Vektorfolgen nicht mehr einfach aus dem RAM zu laden sind, sondern durch die Addition zweier Vektorfolgen berechnet werden müssen. Bei der Bewertung dieses Mehraufwands ist aber zu berücksichtigen, dass die Addition nicht über jede Vektorkomponente zu erfolgen hat: Da nur ein begrenzter Raum simuliert wird, ist die Anzahl der Stützstellen des Feldes endlich. Die Koordinaten der Stützstellen können dann linear adressiert werden, d. h. durch einen skalaren Index ersetzt werden (vgl. Abbildung 5.1 auf Seite 52). Die Addition zweier Vektoren reduziert sich damit auf die Addition zweier Zahlen. Diese Operation kann in einer CPU üblicherweise schneller ausgeführt werden, als einen Wert aus dem RAM zu laden. Bei der Implementierung von Haupt- und Delta-Vektorfolgen ist nur darauf zu achten, dass jedes Vektorfolgenpaar möglichst nur einmal geladen wird, d. h. sortierte Tabellenteile wie die in Tabelle 8.10a sollten am Stück abgearbeitet werden.

Es stellt sich die Frage, warum überhaupt so viele unterschiedliche Vektorfolgen zustande kommen. Die Hauptursache hierfür dürfte im Falle des Einsatzes einer Teilbereichs-WT der Grenzbereich zwischen dem transformierten und dem untransformierten Feldraum sein. Abbildung 8.12 veranschaulicht anhand eines einfachen Beispiels, wieso entlang des Grenzbereiches keine zwei Vektorfolgen gleich sind.

Hier zeigt sich ein gewisses Dilemma: Die Teilbereichs-WT wurde eingeführt, um den numerischen Aufwand zu umgehen, den Wavelet-transformierte PMLs mit sich bringen. Andererseits kann dies zu einer deutlichen Zunahme von Vektorfolgen entlang der Grenzflächen führen.

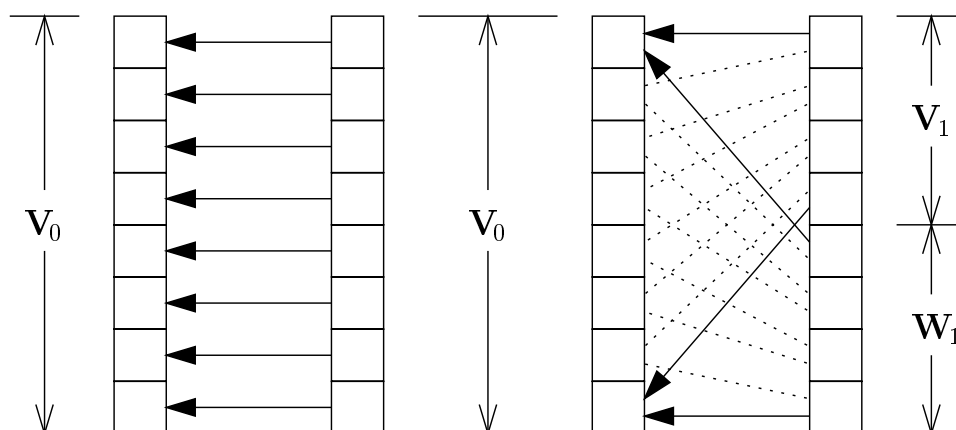


Abbildung 8.12: Beispiel für Relativvektoren und Teilbereichs-WT. Links: Homogene Abbildung zwischen zwei benachbarten Reihen mit jeweils 8 Koeffizienten (die Reihen sind nur der besseren Darstellung wegen räumlich getrennt aufgezeichnet). Beide Reihen befinden sich im Grundraum. Rechts: Wird nur ein Bereich transformiert, treten sehr unterschiedliche Relativvektoren auf. Aus Gründen der Übersicht sind nur für zwei Bildpunkte die Vektoren durchgängig gezeichnet.

Dennoch hat die Teilbereichs-WT Simulationszeiten reduzieren können (siehe Tabelle 8.4 auf Seite 157). Damit dürften die vorgeschlagenen Verbesserungen der Teilbereichs-WT einen weiteren Vorteil verschaffen. Als Beleg hierfür soll nochmals das Beispiel in Abbildung 8.12 dienen: Für die acht Einzelabbildungen werden fünf Folgen aus je zwei Relativvektoren benötigt:  $((0), (4))$ ,  $((-1), (3))$ ,  $((-2), (2))$ ,  $((-3), (1))$  und  $((-4), (0))$ . Dies lässt sich reduzieren auf die Hauptvektorfolge  $((0), (4))$  und die Delta-Vektorfolge  $((-1), (-1))$ .

Die untransformierte FDTD kann von den bislang beschriebenen Verbesserungen ebenfalls profitieren. Allerdings würden sich sehr wahrscheinlich nur geringfügige Beschleunigungen einstellen. Das hat im Wesentlichen zwei Gründe: Der erste Grund ist, dass die Anzahl an  $O^2$ -Listen nicht besonders hoch ist. Demnach können im Vergleich zum transformierten Fall wesentlich größere Bereiche mit den gleichen Operatordaten berechnet werden. Der zweite Grund ist, dass die  $O^2$ -Listen i. d. R. aus deutlich kürzeren Folgen bestehen als bei der transformierten FDTD. Zusammen mit der geringeren Anzahl an  $O^2$ -Listen ergibt sich so ein insgesamt weitaus kleinerer Speicherbedarf, d. h. Caches werden besser ausgenutzt.

### 8.2.3.3 Mehrstufige Transformationen

Je nach Wahl der Transformationsparameter entstehen  $O^2$ -Listen mit sehr unterschiedlichen Größen. Während die  $O^2$ -Listen bei FDTD-Operatoren aus maximal vier<sup>1</sup>  $O^1$ -Elementen bestehen, enthalten die  $O^2$ -Listen des Operators  $\tilde{\mathcal{A}}_2$  aus Tabelle 8.9 bis zu sechs  $O^1$ -Elemente. Mit der Transformationstiefe  $\mathbf{j} = (2, 2, 2)^T$  ergaben sich bei der Verwendung von Haar-Funktionen für die homogene Tensorprodukt-Basis  $O^2$ -Listen, die aus bis zu 70  $O^1$ -Elementen bestanden. In manchen Simulationen wurden sogar noch höhere Werte entdeckt.

In der Regel steigt der numerische Aufwand eines Operators, wenn er einer WT unterzogen wird. Daher ist es einleuchtend, dass mehrstufige Transformationen zu komplexeren Abbildungen führen als einstufige. Einsparungen in der Simulationszeit ergeben sich erst durch die Unterdrückung von Berechnungen.

Abbildung 8.13 zeigt ein typisches Schema für einen eindimensionalen Operator, der um zwei Stufen transformiert wurde. Auch der Matrix-Operator für die eindimensionale Differenziation weist nach der WT ein solches Koeffizientenmuster auf, wie in [3] sehr schön zu sehen ist.<sup>2</sup> Im untransformierten Zustand ist die Differenzationsmatrix eine Bandmatrix mit einem sehr schmalen Band aus nur zwei Koeffizienten in jeder Zeile. In Abbildung 8.13 dagegen ist allein die Anzahl der sog. *Finger* in jeder Zeile mindestens drei. In dem genannten Beispiel aus [3] ist darüber hinaus zu sehen, dass sich die Breite jedes Fingers über mehrere Koeffizienten erstreckt. Hierzu muss aber noch gesagt werden, dass für die WT  $D_2$ -Funktionen verwendet wurden, welche ja einen größeren Träger besitzen als die Haar-Funktionen.

Weiterhin kann Abbildung 8.13 entnommen werden, dass die Vektorfolgen zweier benachbarter Feldkoeffizienten (Koeffizienten aus  $\tilde{\mathbf{b}}$ ) bei solchen Matrixstrukturen verschieden voneinander sind. Dazu sollen zwei Bildkoeffizienten  $\tilde{b}_\mu$  und  $\tilde{b}_\nu$  betrachtet werden, wobei  $\mu < \nu$  gelten soll. Zum Vergleich der Abbildungsvorschriften beider Koeffizienten ist die Zeile  $\nu$  aus  $\tilde{\mathbf{T}}$  um  $\nu - \mu$  Stellen nach links zu verschieben. Enthalten die beiden Zeilen  $\mu$  und  $\nu$  danach an den gleichen Stellen von Null verschiedene

<sup>1</sup>Im normalen Feldraum bzw. ohne PMLs sind es sogar nur zwei.

<sup>2</sup>Bild 3.10 auf Seite 71 in [3].



genau drei Fingern geschnitten werden (was für den größten Teil von  $\tilde{\mathbf{T}}$  gilt). Nehmen wir der Einfachheit halber an, dass die Finger lediglich ein Element breit sind. Dann können die Vektorfolgen dieser Bereiche dargestellt werden aus der Delta-Vektorfolge  $((0), (0), (1))$  sowie einer Haupt-Vektorfolge. Das gilt analog für den Bildraum  $\mathbf{W}_2$ . Für den Bildraum  $\mathbf{W}_1$  lautet die Delta-Vektorfolge  $((-1), (-1), (0))$ , da hier das Gefälle der beiden linken Finger doppelt so hoch ist wie das des dritten Fingers. Damit kann die im vorigen Abschnitt entwickelte Methode auch hier angewendet werden, um Speicherzugriffe einzusparen. Das gilt auch dann, wenn sich die Finger in der Breite über mehr als nur ein Element erstrecken.

#### 8.2.3.4 Mehrdimensionale Felder

Die graphischen Beispiele der beiden vorangegangenen Abschnitte veranschaulichten die Vorteile der Anwendung von Haupt- und Delta-Vektorfolgen für den Fall eindimensionaler Signale bzw. Felder. Es stellt sich die Frage, inwiefern das auf mehrdimensionale Felder und ihre Operatoren übertragbar ist. Die Methode wurde zwar anhand der Daten aus einer mehrdimensionalen Simulation entwickelt (siehe Tabelle 8.10), doch ist noch zu klären, ob ein effizienter Einsatz der Methode auch bei der Verwendung anderer Transformationsparameter erwartet werden kann. Insbesondere die Kombination verschiedener Basen mit ein- und mehrstufigen Wavelet-Transformationen ist in Augenschein zu nehmen.

Skalierungsoperatoren sind bzgl. ihrer Transformation in den Waveletbereich i. d. R. als unkritisch einzustufen, wenn sie nur wenige Inhomogenitäten aufweisen. Ein vollkommen homogener Skalierungsoperator entspricht der Multiplikation des Feldes mit einer Konstanten  $a$ . Schreibt man die Gesamtmatrix des Operators auf, ergibt sich eine Diagonalmatrix  $\text{diag}(a)$ . Diese Eigenschaft ändert sich nicht durch eine WT, d. h. der Operator ist weiterhin durch eine konstante Diagonalmatrix darstellbar. Werden dem Operator im Grundraum nur wenige Inhomogenitäten hinzugefügt, ist die Gesamtmatrix des transformierten Operators nach wie vor überwiegend eine Diagonalmatrix.<sup>3</sup> Diese Eigenschaft dürfte von den Transformationsparametern und der Dimension des Raumes weitgehend unabhängig sein. Daher sollen Skalierungsoperatoren hier nicht weiter betrachtet werden.

<sup>3</sup>Ein Beispiel hierzu ist in [3] auf Seite 73 in Abbildung 3.11 zu finden.

Bevor andere mehrdimensionale Operortypen besprochen werden, soll kurz eine anschauliche Erklärung für die Matrix des eindimensionalen Operators nach Abbildung 8.13 gegeben werden. Im untransformierten Raum soll die Operatormatrix keine reine Diagonalmatrix, sondern eine Bandmatrix mit einem sehr schmalen Band um die Hauptdiagonale sein. Ist das Band drei Elemente breit, gilt  $b_\mu = f(a_{\mu-1}, a_\mu, a_{\mu+1})$  für alle  $\mu$  mit  $2 \leq \mu \leq L - 1$ , wobei  $L$  die Anzahl der Vektorkomponenten ist. Das bedeutet, dass Urbildkoeffizienten wieder auf ungefähr die gleichen Stellen im Bildvektor abgebildet werden. Operatoren mit dieser Eigenschaft sollen als *ortstreu* bezeichnet werden. Die zentralen finiten Differenzen der FDTD sind demnach ortstreu Operatoren für den mehrdimensionalen Raum.

Aus der Wavelet-transformierten Operatormatrix nach Abbildung 8.13 ist ebenfalls zu erkennen, dass der Operator im Grundraum ortstreu ist: Für jeden Entwicklungskoeffizienten eines der Teilräume  $\mathbf{V}_2$ ,  $\mathbf{W}_2$  oder  $\mathbf{W}_1$  lässt sich durch die Lokalisierungseigenschaft der Skalierungsfunktionen und Wavelets angeben, welcher räumliche Bereich bzw. Träger im Grundraum dadurch abgedeckt ist. Diese Zuordnung gilt sowohl für die Bild- als auch für die Urbildkoeffizienten. Dadurch kann einfach durch Anschauung eine grobe Skizze angefertigt werden, die Auskunft darüber gibt, von welchen Urbildkoeffizienten ein Bildkoeffizient abhängt: Es muss lediglich geprüft werden, ob sich der Träger der Entwicklungsfunktion eines Urbildkoeffizienten mit dem Träger der Entwicklungsfunktion des Bildkoeffizienten überschneidet. Oder etwas kürzer formuliert: Die Koeffizienten müssen wenigstens teilweise zum gleichen Raumbereich gehören.<sup>4</sup>

Ein Beispiel für zweidimensionale Felder ist in Abbildung 8.14 für den Fall einer homogenen Tensorprodukt-Basis und einer separablen MSA-Basis wiedergegeben. Vom Bildraum wird der Teilraum  $\mathbf{V}_j$  mit  $\mathbf{j} = (2, 2, 2)^T$  betrachtet. Aus diesem Teilraum wird eine Reihe von Bildpunkten ausgewählt. Berechnet man die Bildpunkte dieser Reihe nacheinander, so ist zu erkennen, dass die Vektorfolgen immer unterschiedlich sein müssen. Es ist jedoch ebenfalls zu erkennen, dass für diese Reihe eine Delta-

---

<sup>4</sup>Im Grunde genommen stellen Skalierungsoperatoren einen Spezialfall ortstreuer Operatoren dar. Transformiert man z. B. einen geringfügig inhomogenen Skalierungsoperator um zwei Stufen, so treten – zusätzlich zu den Hauptdiagonalelementen – von Null verschiedene Koeffizienten in dem Bereich der in Abbildung 8.13 dargestellten Finger auf. Das ist auch in dem in Fußnote 3 genannten Beispiel zu sehen.

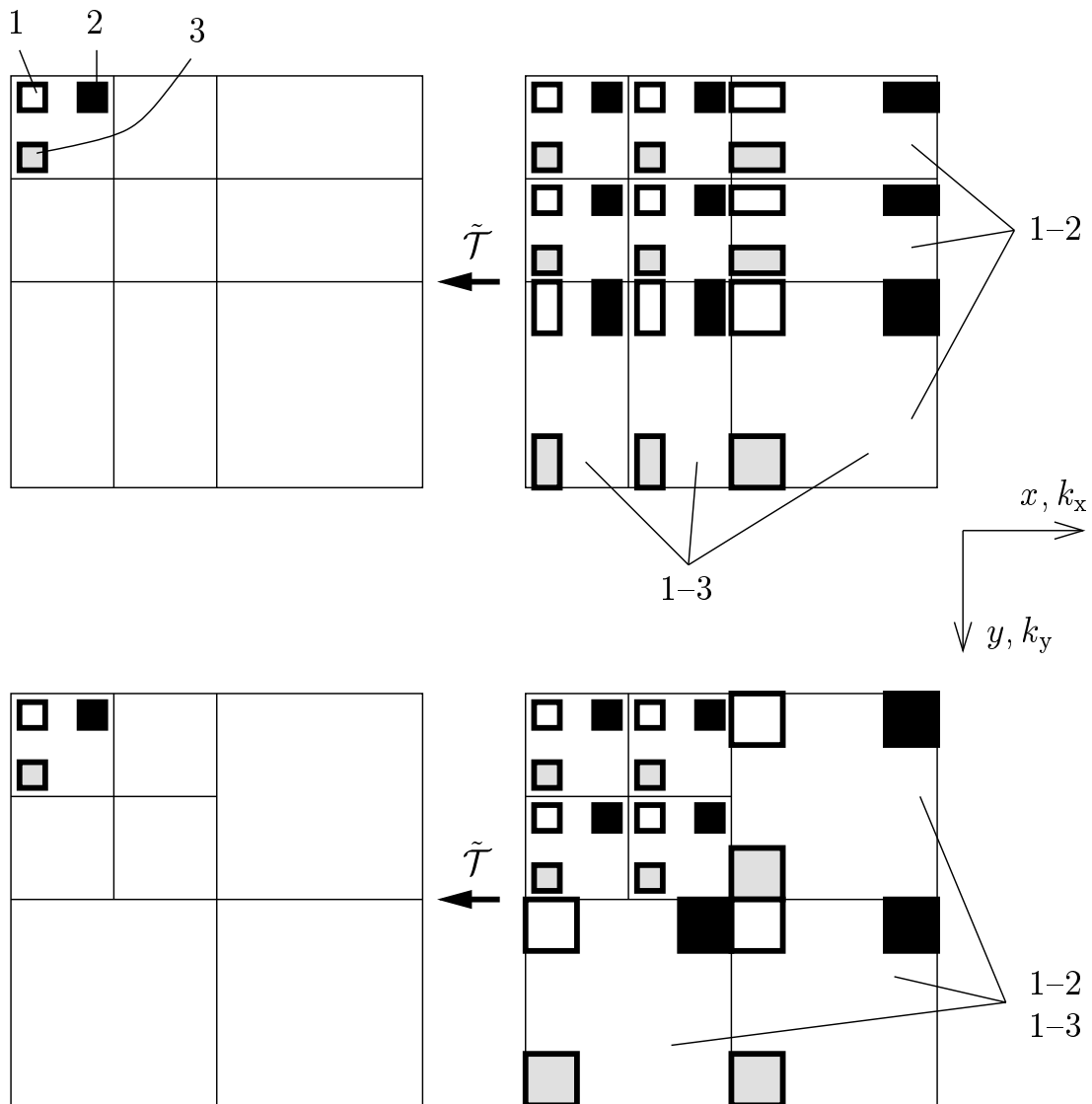


Abbildung 8.14: Abbildung auf drei Bildpunkte (1, 2, 3; linke Felder) mit einem zweistufig Wavelet-transformierten ortstreuem Operator  $\tilde{T}$  im zweidimensionalen Raum. Durch die Füllfarbe ist gekennzeichnet, welcher Bildkoeffizient von welchen Urbildkoeffizienten (rechte Felder) abhängt. Oben: homogene Tensorprodukt-Basis; unten: separable MSA-Basis; Raumaufteilung siehe Kapitel 4. Die mit „1-2“ gekennzeichneten Bereiche erfordern nicht-verschwindende Delta-Vektoren, wenn die zwischen den Bildpunkten 1 und 2 liegenden Bildpunkte nacheinander berechnet werden. Analog „1-3“ für Bildpunkte zwischen 1 und 3.

Vektorfolge angegeben werden kann.

Am einfachsten ist das nachzuvollziehen, indem man sich die mit „1“ und „2“ markierten Bildpunkte und die Bereiche der zugehörigen Urbildkoeffizienten für den Fall der homogenen Tensorprodukt-Basis ansieht. Zunächst sehen wir uns die Urbildbereiche im linken oberen Teilraum ( $\mathbf{V}_2 \times \mathbf{V}_2$ ) an. Die Distanz zwischen den beiden Bereichen ist genauso groß die wie Distanz der beiden Bildpunkte. Die gleiche Aussage gilt für die übrigen fünf Teilräume der beiden linken Spalten ( $\mathbf{V}_2 \times \mathbf{W}_2$ ,  $\mathbf{V}_2 \times \mathbf{W}_1$ ,  $\mathbf{W}_2 \times \mathbf{V}_2$ ,  $\mathbf{W}_2 \times \mathbf{W}_2$ ,  $\mathbf{W}_2 \times \mathbf{W}_1$ ). In diesen Räumen werden daher keine Delta-Vektoren benötigt (bzw. sie sind mit Nullvektoren identisch). Die Abstände in den drei Teilräumen der rechten Spalten ( $\mathbf{W}_1 \times \mathbf{V}_2$ ,  $\mathbf{W}_1 \times \mathbf{W}_2$ ,  $\mathbf{W}_1 \times \mathbf{W}_1$ ) sind dagegen vergrößert. Hier sind Delta-Vektoren vonnöten, die einen zusätzlichen Schritt nach rechts bewirken.

Je nach Auswahl der Richtung (horizontale Reihen wie z. B. zwischen den Bildpunkten „1“ und „2“, vertikale Reihen wie z. B. zwischen „1“ und „3“) und Art der Basis ergeben sich unterschiedliche Delta-Vektorfolgen. In allen Fällen scheint es aber möglich und sinnvoll zu sein, die Methode der Haupt- und Delta-Vektorfolgen einzuführen. Eine detailliertere Untersuchung zur Verifikation dieser plausiblen Vermutung wäre sicher nutzbringend, soll jedoch nicht mehr Gegenstand dieser Arbeit sein.

Statt dessen soll die Diskussion zu mehrdimensionalen Feldern mit einem Hinweis abgeschlossen werden: Wird der vollständige Raum<sup>5</sup> lediglich um eine Stufe transformiert, erübrigt sich die Notwendigkeit von Delta-Vektorfolgen. Zur Verdeutlichung genügt es, sich die vier kleinen quadratischen Teilräume der Bild- und Urbildfelder aus Abbildung 8.14 anzuschauen ( $\mathbf{V}_2 \times \mathbf{V}_2$ ,  $\mathbf{V}_2 \times \mathbf{W}_2$ ,  $\mathbf{W}_2 \times \mathbf{V}_2$  und  $\mathbf{W}_2 \times \mathbf{W}_2$ ) und die restlichen Teilräume wegzudenken. Bei einer einstufigen WT lauten die Teilräume dann  $\mathbf{V}_1 \times \mathbf{V}_1$ ,  $\mathbf{V}_1 \times \mathbf{W}_1$ ,  $\mathbf{W}_1 \times \mathbf{V}_1$  und  $\mathbf{W}_1 \times \mathbf{W}_1$ . Die Teilräume, die zuvor Delta-Vektoren erforderten, sind jetzt ausgeblendet. Die Distanzen zwischen den in der Abbildung markierten Bereichen sind in jedem Teilraum dieselben. Delta-Vektoren erübrigen sich somit.

---

<sup>5</sup>Eine Teilraum-WT wird in dieser Betrachtung explizit ausgeschlossen!

### 8.2.3.5 Weitere Ergebnisse

Bei der dreidimensionalen Simulation des Mikrostreifenleitungsfilters in Abschnitt 8.2.2 wurde entdeckt, dass mit einer separablen MSA-Basis genauere Ergebnisse in kürzerer Zeit berechnet werden konnten als mit einer homogenen Tensorprodukt-Basis. Mit Hilfe der Darstellungen in Abbildung 8.14 kann eine mögliche Erklärung hierfür gegeben werden.

Der Unterschied zwischen den beiden Basen ist, dass bei der homogenen Tensorprodukt-Basis eine Vermischung der Skalen auftritt, wie in Abschnitt 4.1 bereits erwähnt wurde. Ausgehend vom Raum  $\mathbf{V}_1 \times \mathbf{W}_1$  bei der separablen MSA-Basis kann man auch sagen, dass aus diesem Raum durch eine weitere WT in x-Richtung die Teilräume  $\mathbf{V}_2 \times \mathbf{W}_1$  und  $\mathbf{W}_2 \times \mathbf{W}_1$  der homogenen Tensorprodukt-Basis erzeugt werden. Analoges gilt für  $\mathbf{W}_1 \times \mathbf{V}_1$ , der durch die WT in y-Richtung in  $\mathbf{W}_1 \times \mathbf{V}_2$  und  $\mathbf{W}_1 \times \mathbf{W}_2$  aufgeteilt wird.

Man kann also sagen, dass Operatoren bei ihrer WT bzgl. einer homogenen Tensorprodukt-Basis mehr Transformationsschritte zu durchlaufen haben als im Falle der separablen MSA-Basis. Es wurde des öfteren betont, dass Operatoren umso komplexer werden, je mehr sie transformiert werden. Insofern wäre es nicht verwunderlich, wenn diese zusätzlichen Transformationen auch zu der beobachteten Erhöhung der Rechenzeit beitragen.

Im Hinblick auf das Detailgitter-Verfahren gilt aber, dass Operatoren mit zunehmender Transformationstiefe stärker komprimiert werden können. Es wurde jedoch auch gesagt, dass es wahrscheinlich immer eine obere Grenze gibt, oberhalb derer der durch eine weitere WT-Stufe steigende numerische Aufwand nicht mehr durch die Kompression mit dem Detailgitter-Verfahren kompensiert werden kann. Daher sind vermutlich die zusätzlichen Möglichkeiten der Kompression, die sich bei der Vermischung der Skalen ergeben, kein wirklicher Gewinn, da die numerische Komplexität des Operators zu sehr ansteigt.

Die zusätzliche Kompression kann an den Simulationsergebnissen in Abbildung 8.11 auf Seite 168 abgelesen werden: Mit der homogenen Tensorprodukt-Basis wurden etwas ungenauere Ergebnisse erzielt als mit der separablen MSA-Basis. Der Unterschied ist zwar nicht sonderlich groß, aber doch deutlich erkennbar. Es ist zu vermuten, dass eine WT zu noch

höheren Skalen hin die Unterschiede zwischen den beiden Basistypen vergrößern würde.

Abschließend soll zu dem Vergleich der beiden Basistypen nochmals betont werden, dass die Simulationen noch ohne die beschriebenen Verbesserungen der Listen-Darstellung durchgeführt wurden. Weitere Vergleiche auf der Basis eines dahingehend optimierten Simulationsprogrammes könnten daher die Unterschiede in den Rechenzeiten verringern.

Als nächstes sollen noch einige Anmerkungen zu der Listen-Darstellung von Operatoren getroffen werden. Die auf Seite 172 in Tabelle 8.9 gezeigten hohen Anzahlen an Listen führen dazu, dass auch die Verkettung von Operatoren übermäßig viel Zeit in Anspruch nimmt. Da diese Programmteile jedoch nicht Gegenstand der Untersuchung in dieser Arbeit sind, wurden bisher auch keine konkreten Angaben hierzu gemacht. Bei näherer Betrachtung dürfte jedoch ersichtlich sein, dass sich die in Abschnitt 5.3.2.2 erwähnte quadratische Ordnung beim Aufbau von  $O^3$ -Listen deutlich bemerkbar macht. Die beschriebenen Methoden zur Kompression der Listen lassen demnach erwarten, dass sich auch die Verkettung von Operatoren drastisch beschleunigt.

In der Literatur wurde kein Hinweis auf effiziente Speicherkonzepte für transformierte mehrdimensionale Operatoren gefunden. So konnte erst durch die Analyse einiger Simulationen entdeckt werden, in welchem Ausmaß die Abbildungsdaten eines Operators bei seiner Transformation in den Waveletbereich zunehmen. Die aufgedeckten Probleme zeigen, dass die Simulation weiterer Strukturen mit der ursprünglichen Listen-Darstellung nicht sinnvoll ist. Vielmehr sollten die vorgeschlagenen Verbesserungen (Separation der Faktorfolgen von den Vektorfolgen; Einführung von Haupt- und Delta-Vektorfolgen) erprobt werden. Die notwendigen Änderungen am bestehenden Simulationsprogramm wären jedoch so umfangreich, dass dies nicht mehr Bestandteil dieser Arbeit sein kann und daher im Fokus weiterer Untersuchungen liegen sollte.

Es sollte bedacht werden, dass mit Wavelet-transformierten Operatoren – trotz der beschriebenen Datenzunahme – Abnahmen bei den Rechenzeiten zu beobachten waren. Unter diesem Aspekt erscheinen die Verbesserungen, welche zu einer weitaus effizienteren Operator-Darstellung führen, als vielversprechende Möglichkeit, Simulationszeiten deutlich reduzieren zu können.



# Kapitel 9

## Ausblicke und Weiterentwicklungen

Nachdem die Matrix- und Listen-basierte WT der FDTD untersucht worden ist, sollen noch einige Ausblicke auf weitere Simulationstechniken gegeben werden. Hierbei ist vor allem die sog. *Multiresolution time-domain* (MRTD) zu erwähnen. Von allen Wavelet-basierten Simulationsverfahren für elektromagnetische Felder weist die MRTD die wohl größten Ähnlichkeiten zu den in dieser Arbeit untersuchten Verfahren auf. Einige Aspekte der MRTD sollen daher im folgenden Abschnitt besprochen werden. Die sich daraus ergebenden Erkenntnisse werden mitunter von hohem Nutzen bei der Auslotung der Erweiterungsmöglichkeiten der Listen-basierten WT der FDTD sein.

Weiterhin wurden in der Literatur *implizite* Verfahren veröffentlicht, welche das Lösen von Gleichungssystemen erfordern. Hierzu gehört auch die *Method of Lines*, die in [38] und [39] zur Anwendung von Daubechies-Funktionen weiterentwickelt wurde. In [39] wird u. a. auf ein implizites Verfahren eingegangen, welches Wavelets zur Entwicklung Green'scher Funktionen einsetzt. Ferner werden dort Veröffentlichungen zu weiteren Verfahren genannt, die in den letzten Jahren auf Wavelets angepasst wurden.

Abschnitt 9.2 wird schließlich einige Entwicklungen im Bereich der Wavelets vorstellen, die im Rahmen der WT der FDTD von Nutzen sein könnten. Vor allem mit diesem Abschnitt soll deutlich werden, wie jung

das Forschungsgebiet der Simulation elektromagnetischer Felder mit Wavelets angesichts der bislang ungenutzten Möglichkeiten ist.

## 9.1 MRTD – Multiresolution time-domain

Ursprünglich wurde die MRTD auf der Basis von Battle-Lemarié-Skalierungsfunktionen und -Wavelets entwickelt, die von kubischen Splines abgeleitet wurden. Später wurden auch andere Funktionenklassen eingesetzt. Allen Varianten ist gemein, dass die Maxwell'schen Gleichungen unter Anwendung eines Galerkin-Verfahrens zunächst nur mit den Skalierungsfunktionen diskretisiert werden. Auf die gleiche Weise kann auch die FDTD hergeleitet werden, indem Haar-Skalierungsfunktionen benutzt werden.

Das auf Skalierungsfunktionen basierende Simulationsverfahren wird als S-MRTD bezeichnet [40]. Die S-MRTD führt Berechnungen nur auf einer Auflösungsstufe durch. Die Bezeichnung als „Multi-Resolution“-Verfahren ist dennoch gerechtfertigt, da die Hinzunahme von Wavelets zumindest möglich ist. Diese Weiterentwicklung wird in [41] ausführlich dargestellt und als W-MRTD-Schema bezeichnet.

An dieser Stelle zeigt sich bereits ein wesentlicher Unterschied zwischen der MRTD und der in dieser Arbeit beschriebenen WT der FDTD: Die MRTD wird von einer groben Auflösungsstufe her entwickelt. Zur Erhöhung der Auflösung müssen Elemente aus wenigstens einem Waveletraum hinzugenommen werden. Dazu sind die Operatoren mittels des Galerkin-Verfahrens für alle benötigten Funktionenräume zu diskretisieren. Bei der WT der FDTD dagegen werden lediglich die FDTD-Operatoren für die höchste benötigte Auflösung aufgestellt. Die Operator-Darstellungen im Waveletbereich ergeben sich automatisch durch die Verkettung mit den Transformationsoperatoren.

Hieraus folgt auch schon der nächste Unterschied: In der MRTD wird die gleiche Funktionenklasse für die Diskretisierung der Operatoren und die Aufteilung in verschiedene Auflösungsstufen verwendet. Mit den Techniken der vorliegenden Arbeit dagegen wird am Beispiel der FDTD demonstriert, wie ein bereits diskretisiertes Verfahren mit Hilfe einer WT zu einem Verfahren mit mehreren Auflösungsstufen erweitert werden kann.

Das bedeutet, dass Diskretisierungs- und Transformationsfunktionen unabhängig voneinander gewählt werden können.

Das Ziel dieses Abschnitts soll es sein, die MRTD und ihre Entwicklung in groben Zügen zu beschreiben, ohne auf die mathematischen Details näher einzugehen. Dadurch sollen weitere beachtenswerte Unterschiede zur WT der FDTD vermittelt werden. Darüber hinaus ist dieser Abschnitt als eine Art „kommentiertes Inhaltsverzeichnis“ zu verstehen, das die Einarbeitung in die MRTD erleichtern und einen vertiefenden Einblick verschaffen soll. Es soll jedoch ausdrücklich betont werden, dass dies das Studium der genannten Veröffentlichungen nicht ersetzen kann.

### 9.1.1 MRTD mit Battle-Lemarié-Funktionen

Die beiden bereits genannten Veröffentlichungen [40] und [41] gehören zu den ersten Beiträgen, welche die Entwicklung der MRTD einleiteten. Hierbei wurden erstmalig Battle-Lemarié-Funktionen zur Diskretisierung der Maxwell'schen Gleichungen eingesetzt.

Die Battle-Lemarié-Funktionen sind im Gegensatz zu Daubechies-Funktionen geringer Ordnung differenzierbar. Allerdings ist ihr Träger unendlich, was bei einem exakten Galerkin-Verfahren unendlich viele Entwicklungskoeffizienten zur Folge hätte. Da sie jedoch ein exponentielles Abklingverhalten besitzen, genügt i. d. R. ein kurzes Intervall, um hinreichend genaue Berechnungen durchführen zu können.

Durch die Entwicklung in Battle-Lemarié-Funktionen werden je nach gewünschter Genauigkeit zwischen 18 und 30 Stützstellen zur Berechnung der eindimensionalen Ableitung an einer Stelle benötigt. Der numerische Aufwand scheint damit zunächst weitaus höher zu sein als bei der FDTD, die mit zwei Stützstellen auskommt. Andererseits präsentieren die Autoren in [40], dass die S-MRTD deutlich bessere Dispersionseigenschaften aufweist als die FDTD und somit eine sehr viel gröbere Diskretisierung des Raumes erlaubt.

Zu diesem Argument wird in [42] kritisch angemerkt, dass der numerische Aufwand der MRTD mit dem einer FDTD 18ter Ordnung verglichen werden kann, die Dispersion dagegen lediglich mit einer FDTD 14ter Ordnung. Ferner wird angemerkt, dass sich FDTD-Verfahren höherer Ordnung aufgrund des numerischen Aufwands in der Praxis als unvorteilhaft

erwiesen haben. Eine teilweise Bestätigung dieser Aussagen findet sich in [43]. Es wird jedoch auch deutlich, dass eine MRTD gegenüber einer FDTD höherer Ordnung es zu leisten vermag, Simulationen mit lokal höherer Auflösung durchzuführen.

Die Möglichkeiten der MRTD, gegenüber der FDTD genauere Ergebnisse liefern zu können, werden nach [41] durch das Auftreten unphysikalischer Lösungen (engl. *spurious modes*) etwas eingeschränkt. In [44] wird berichtet, dass die in [41] beobachteten Effekte jedoch nicht auf unphysikalische Lösungen, sondern auf Simulationsungenauigkeiten zurückzuführen sind. Hierunter ist auch die Verwendung einer zu geringen Anzahl von Stützstellen zu verstehen.

Bereits in [41] wurden Stabilitätsanalysen der S- und W-MRTD für den Fall gleicher Diskretisierungsweiten in allen drei Raumrichtungen durchgeführt. Hierbei wurden 18 Stützstellen für die Differenzialoperatoren angesetzt. Die Ergebnisse zeigen, dass die maximale Zeitschrittweite für die MRTD etwas geringer ausfällt als für die FDTD:

$$\text{FDTD} \quad : \quad \Delta t \leq \Delta t_{\text{FDTD}} \approx 0,57735 \frac{\Delta l}{c}, \quad (9.1)$$

$$\text{S-MRTD} \quad : \quad \Delta t \leq \Delta t_{\text{S-MRTD}} \approx 0,368112 \frac{\Delta l}{c}, \quad (9.2)$$

$$\text{W-MRTD} \quad : \quad \Delta t \leq \Delta t_{\text{W-MRTD}} \approx 0,253064 \frac{\Delta l}{c}. \quad (9.3)$$

Für die zweidimensionale MRTD werden detaillierte Herleitungen und Analysen des Dispersionsverhaltens in [43] durchgeführt. Insbesondere die Abhängigkeit von der Anzahl der Stützstellen wird demonstriert. Auch der Fall, dass nur in einer Raumrichtung Wavelets hinzugenommen werden, wird untersucht. Insgesamt betrachtet fallen einige unstetige Eigenschaften der MRTD im Dispersionsverhalten auf, für die jedoch keine Erklärungen gegeben werden können. Ferner wird in [43] die Aussage aus [41] bestätigt, dass die MRTD eigentlich nur dann nennenswert weniger dispersiv als die FDTD ist, wenn die gewählte Zeitschrittweite deutlich unterhalb des erlaubten Maximums liegt. So wird häufig  $\Delta t = \frac{1}{5} \Delta t_{\text{MRTD}}$  angesetzt.

Eine Analyse des Dispersionsverhaltens der FDTD, S-MRTD und verschiedener weiterer Verfahren wird in [45] präsentiert. Sowohl S-MRTD

als auch FDTD werden hier mit 10 Zellen pro Wellenlänge diskretisiert. Bei jedem Verfahren wird eine Zeitschrittweite gewählt, die jeweils eine minimale Dispersion ermöglicht. Als Resultat wird festgestellt, dass die MRTD im Vergleich zur FDTD nur bei hohen Frequenzen weniger dispersiv ist. Interessanterweise weist sie sogar eine Art Bandpass-Verhalten auf, d. h. ihre Dispersion ist in einem relativ schmalen Frequenzbereich besonders niedrig.

Zwei Jahre nach der Veröffentlichung von [45] wurde in [44] ein alternatives Diskretisierungsschema der MRTD beschrieben. Vernachlässigt man keine Entwicklungskoeffizienten, so sollte die W-MRTD identische Ergebnisse liefern wie eine S-MRTD, die mit doppelt so hoher Auflösung diskretisiert wurde. Auch die maximal zulässige Zeitschrittweite sollte sich dann halbieren. Tatsächlich ist jedoch

$$\Delta t_{W-MRTD} = 0,687465 \Delta t_{S-MRTD} . \quad (9.4)$$

Das deutet darauf hin, dass sich die Auflösung etwas weniger als um den Faktor 2 erhöht. Diese Entdeckung in [44] wird mit dem Diskretisierungsschema erklärt, welches seit [40] angewendet wird: Bei der FDTD sind die Stützstellen des elektrischen und des magnetischen Feldes symmetrisch zueinander angeordnet, wodurch zentrale finite Differenzen zur Approximation der Ableitungen eingesetzt werden können. Dieses Schema, das als *MRTD-Schema I* bezeichnet werden soll, wurde auch für die Stützstellen bei der S-MRTD angesetzt. Nimmt man nun Wavelets hinzu, wird die Auflösung jedes der Felder für sich betrachtet verdoppelt. Damit verdoppelt sich zwar auch die Anzahl der Stützstellen, doch die Stützstellen des elektrischen und des magnetischen Feldes sind jetzt nicht mehr symmetrisch zueinander angeordnet. Stattdessen liegen sie nun an den gleichen Stellen.<sup>1</sup> Diese sehr starke Asymmetrie führt zu einem Stabilitätskriterium, bei dem die maximal erlaubte Zeitschrittweite weniger als halbiert wird.

Dass die Zeitschrittweite größer ausfällt als erwartet, ist in diesem Fall aber nicht unbedingt von Vorteil. In [44] wird weiter gezeigt, dass auch die Dispersion bei der Hinzunahme von Wavelets weniger abnimmt, als zu erwarten gewesen wäre. Von diesen Erkenntnissen ausgehend präsentieren

---

<sup>1</sup>Genauer gesagt sind die beiden Felder nun um genau eine Diskretisierungsweite gegeneinander versetzt, statt um eine halbe wie bei der S-MRTD oder FDTD.

die Autoren ein verändertes Diskretisierungsschema, welches von einer asymmetrischen Anordnung bei der S-MRTD ausgeht. Beim Übergang auf eine W-MRTD ergeben Skalierungsfunktionen und Wavelets zusammen dann doppelt so hoch aufgelöste Felder, die symmetrisch zueinander platziert sind. Erst mit diesem neuen Schema (*MRTD-Schema II*) kann eine W-MRTD so betrachtet werden, als wäre eine S-MRTD mit doppelt so hoher Auflösung aufgestellt und dann um eine Stufe in den Waveletraum transformiert worden.

Materialoperatoren stellen nach Durchsicht der Literatur wohl die größte Hürde bei der Implementierung einer MRTD dar. In den meisten Veröffentlichungen wird zunächst vorausgesetzt, dass sich die skalare ortsabhängige Permittivität als

$$\varepsilon(\mathbf{r}) = \varepsilon_x(x)\varepsilon_y(y)\varepsilon_z(z) \quad (9.5)$$

schreiben lässt, d. h. als das Tensorprodukt dreier eindimensionaler Funktionen. Diese recht starke Einschränkung lässt in [46] nur die Simulation solcher Strukturen zu, deren Materialverteilung sich exakt auf diese Weise darstellen lässt. Die Integrale einiger möglicher Fälle werden explizit angegeben und numerisch ausgewertet. Eine etwas schematischere Vorgehensweise wird in [41] präsentiert, die u. a. das Lösen einer Matrixgleichung erfordert. In [47] wird beschrieben, welchen Umfang eine exakte Darstellung einer allgemeineren Materialverteilung, die sich nicht durch Gleichung (9.5) darstellen lässt, im Waveletbereich haben kann. Als Alternative werden die Möglichkeiten von Gleichung (9.5) etwas erweitert, indem der Raum derart in Teilräume aufgeteilt wird, dass die Gleichung in jedem dieser Teilräume anwendbar ist.

Das Lösen einer Matrixgleichung ist auch in [48] erforderlich. Die Berechnung wird durch die Angabe einer Koeffiziententabelle zur Umgehung der numerischen Integration etwas erleichtert. Weiterhin wird beschrieben, wie Untermatrizen den numerischen Aufwand zum Lösen der Matrixgleichung reduzieren können.

Die Schwierigkeiten bei der Berechnung der Materialoperatoren scheinen der wesentliche Grund dafür zu sein, warum in keiner der durchgesehenen Literaturstellen außer den Funktionen des  $\mathbf{V}_0$  und  $\mathbf{W}_0$  höher auflösende Funktionen ( $\mathbf{W}_\nu, \nu < 0$ ) in Simulationsbeispielen eingesetzt wurden. Zumindest werden in [48] erstmalig alle notwendigen Gleichungen hierfür

hergeleitet. Dennoch beschränken sich die Beispiele auf die einstufige W-MRTD. Ein weiterer Grund für diese Beschränkung könnte sein, dass nicht nur die Koeffizienten der Ableitungsoperatoren für jede Wavelet-Auflösung separat berechnet werden müssen, sondern auch die Koeffizienten, welche die Verkopplung zwischen den unterschiedlichen Räumen beschreiben. Hierzu müssen diverse Integrale numerisch ausgewertet werden. Daher sind in [48] auch nur die Koeffizienten für die einstufige W-MRTD angegeben.

Ferner ist anzumerken, dass in [48] das o. g. verbesserte Diskretisierungsschema erwähnt wird, jedoch bewusst das alte MRTD-Schema I eingesetzt wird. Ein Übergang zum MRTD-Schema II erfordert vermutlich die Modifikation der Integrale, die zur Berechnung der o. g. Koeffizienten erforderlich sind.

Ein weiterer Unterschied zwischen der (transformierten und untransformierten) FDTD und der MRTD besteht in der Behandlung von Randwerten. Zur Darstellung elektrisch oder magnetisch ideal leitfähiger Wände genügt bei der S-MRTD die Anwendung des Spiegelungsprinzips [41]. Die Hinzunahme von Wavelets erfordert zusätzlich die Verdopplung der zugehörigen Entwicklungskoeffizienten für die Wavelets, deren Träger im Bereich der Wand liegen [48].

Schließlich soll noch kurz auf abgestufte Gitter (*graded meshes*) eingegangen werden, die in der FDTD eine räumliche Variation der Diskretisierungsweite erlauben, um die Geometrien physikalischer Strukturen besser nachbilden zu können. Während dies in der FDTD sehr häufig eingesetzt wird, kommt in der S-MRTD nur eine homogene Diskretisierung zum Einsatz. Eine Erweiterung auf abgestufte Gitter dürfte recht schwierig sein. Daher sind räumliche Variationen nur über die lokale Erhöhung der Auflösung mittels Wavelets durchführbar. Da nach dem oben Gesagten bislang nur Wavelets des  $\mathbf{W}_0$  eingesetzt werden, bleiben die Möglichkeiten der räumlichen Anpassung gegenüber der FDTD recht beschränkt.

Die Vorteile der MRTD gegenüber der FDTD sind bei einer genaueren Analyse der in der Literatur präsentierten Simulationsbeispiele nicht ganz klar auszumachen. In [40] beispielsweise wird ein luftgefüllter quaderförmiger Hohlraum simuliert, dessen Wände ideal leitfähig sind. Für die FDTD wird eine räumliche Diskretisierung von  $\Delta l = 0,1$  m und eine Zeitschrittweite von  $\Delta t = 100$  ps gewählt. Nach Gleichung (2.26) ist bei

der angegebenen räumlichen Auflösung jedoch eine maximale Zeitschrittweite von etwa 190 ps zulässig. Im zeitlichen Vergleich erscheint die FDTD hierdurch benachteiligt.

Ähnliches ist in [46] zu finden. Es wird dort u. a. ein quaderförmiger Hohlraum mit ideal leitfähigen Wänden simuliert. In einer der Innenkanten ist ein dielektrischer Stab mit der Permittivitätszahl  $\varepsilon_r = 6$  platziert.<sup>2</sup> In jeder Richtung wird der Raum für die FDTD mit 40 Elementen diskretisiert, wobei  $\Delta l = 1,5$  mm ist. Damit ergibt sich nach Gleichung (2.26)  $\Delta t_{\text{FDTD}} \approx 2,8$  ps, angesetzt wird in [46] aber  $\Delta t = 1$  ps. Dagegen werden die MRTD-Simulationen von vornherein mit einer Zeitschrittweite von 3 ps ausgeführt, wobei die räumlichen Diskretisierungen zwischen 6 mm und 10 mm betragen.

Bei gleichem  $\Delta l$  liegen die maximal zulässigen Zeitschrittweiten nach den Gleichungen (9.1)-(9.3) für FDTD und MRTD in der gleichen Größenordnung. Das lässt vermuten, dass auch die FDTD in dem gerade genannten Beispiel gröber diskretisiert werden könnte. Als Richtwert wurde in Abschnitt 2.2 für die FDTD gesagt, dass  $\Delta l$  nicht größer werden sollte als das 0,1-fache der kürzesten zu simulierenden Wellenlänge. In dem betrachteten Beispiel werden nur Frequenzen unterhalb von 2 GHz simuliert. Bei einer ebenen Welle in dem dielektrischen Stab entspricht dies einer Wellenlänge von etwa 61 mm (in Luft etwa 150 mm). Nach dem o. g. Kriterium wäre für die FDTD ein Wert von  $\Delta l \approx 6$  mm zulässig gewesen, wobei die maximal zulässige Zeitschrittweite dann ebenfalls größer ausfällt.

In diesem Beispiel ist die FDTD gleich mehrfach benachteiligt: Es werden aufgrund der feinen Auflösung mehr Elemente berechnet als erforderlich sind und das mit einer unnötig niedrigen Zeitschrittweite. Hierbei ist auch zu berücksichtigen, dass die numerischen Eigenschaften der FDTD dann am besten sind, wenn sie an der Stabilitätsgrenze betrieben wird, d. h. wenn  $\Delta t = \Delta t_{\text{FDTD}}$  gilt [49]. Dabei steht vor allem die Dispersion im Vordergrund, die mit zunehmender Zeitschrittweite abnimmt [50, 51].

Sicherlich darf eine MRTD wegen der Verwendung von Funktionen höherer Regularität immer etwas gröber diskretisiert werden als eine konventionelle FDTD. Aus dem gleichen Grund wurden auch FDTD-Varianten höherer Ordnung entwickelt, wie beispielsweise die o. g. Verfahren 14ter

---

<sup>2</sup>Siehe Abbildung 7 auf Seite 63 in [46].

und 18ter Ordnung. In der MRTD ist es nach [48] zulässig, dass die Materialeigenschaften innerhalb eines diskretisierten Raumelementes inhomogen sein dürfen. Die klassische FDTD geht dagegen von konstanten Werten in einer Yee-Zelle aus. Allerdings sind auch in der FDTD recht einfache Methoden gebräuchlich, die Inhomogenitäten im Bereich einer Yee-Zelle zulassen [4]. Daher muss die Diskretisierung der FDTD nicht notwendigerweise an den kleinsten dielektrischen Strukturen ausgerichtet werden.

Insofern sind auch die Aussagen zu der zweidimensionalen Simulation eines Bandleitungsfilters in [48] nicht ganz einfach zu bewerten. Es wird festgestellt, dass die FDTD-Simulation etwa 23 mal so viel Speicher benötigt wie die W-MRTD-Simulation. Zum einen kommt dies durch die feinere Diskretisierung in der Ausbreitungsrichtung zustande (1600 Elemente gegenüber 200 bei der MRTD). Nimmt man wieder die höchste zu simulierende Frequenz als Entscheidungskriterium für die Wahl der Diskretisierungsweite, wären weitaus weniger Elemente erforderlich.

Der zweite Grund für den höheren Speicheraufwand liegt in der Diskretisierung der Höhe begründet, die mit acht Elementen bei der FDTD viermal so hoch ist wie bei der MRTD. Allerdings gibt es bei diesem Beispiel weder bei den Feldern noch bei dem Wellenleiter Variationen in der Höhe, sodass es genügt hätte, für die Höhe lediglich ein Diskretisierungselement anzusetzen. Die Bandleitung wäre damit durch eine eindimensionale Simulation zu analysieren gewesen.

Zu dem Bandleitungsfilter werden in [48] zwar keine Simulationszeiten für einen Vergleich von FDTD und MRTD gegeben, dafür wird aber die Adaptivität der W-MRTD demonstriert, indem für einen Zeitschritt die räumliche Verteilung der nicht vernachlässigbaren Entwicklungskoeffizienten für die Wavelets aufgezeichnet wird. Es wird – wie in vielen der durchgesehenen Literaturstellen – darauf hingewiesen, dass die Adaptivität die Einsparung von Speicherplatz ermöglicht, ohne auf numerische Genauigkeit verzichten zu müssen. Es ließen sich aber in keiner dieser Literaturstellen konkrete Angaben bzw. Algorithmen dazu finden, wie die Adaptivität speichertechnisch umgesetzt wird. Während bei einem statischen Gitter Entwicklungskoeffizienten an konstanten, deterministischen Speicherplätzen lokalisiert sind, kommt bei einem auch speichertechnisch umgesetzten adaptiven Gitter mit den zeitlichen Veränderungen eine hohe Dynamik ins Spiel, die kaum noch Raum lässt für konstante

Speicherplätze. Für jeden Bild- und jeden Urbild-Koeffizienten ist jeweils zu ermitteln, wo sich dieser im Adressraum befindet. Zum einen erfordern solche Auswertungen Zeit. Auch das Reservieren und Freigeben von Speicher kostet Zeit. Zum anderen benötigt man für die Hinterlegung der notwendigen Informationen Speicherplatz, was die angestrebte Reduktion des Speicheraufwands schmälert.

Zumindest wird in [48] eine Speicherstruktur angedeutet, die eine teilweise dynamische Verwaltung erlauben könnte. Grob formuliert lassen sich alle Entwicklungskoeffizienten aller Waveleträume dem jeweils „räumlich nächsten“ Koeffizienten des Skalierungsraumes zuweisen. Alle Koeffizienten der Waveleträume, die zu einem Koeffizienten eines Skalierungsraumes gehören, können dann speichertechnisch als zusammengehörig betrachtet werden.

Mit diesem Konzept reduzieren sich die Möglichkeiten der Kompression deutlich. Dafür werden jedoch auch die oben beschriebenen Probleme der dynamischen Speicherverwaltung abgemildert. Dennoch ist der Aufwand für die verbliebene Dynamik nicht unerheblich. Selbst in [48] wird nur erwähnt, dass mit dem beschriebenen Konzept eine dynamische Speicherverwaltung *möglich* ist, doch ob es in den durchgeführten Simulationen auch eingesetzt wurde, bleibt offen.

Angesichts der aufgezeigten Probleme ist es fraglich, ob die räumliche Adaptivität bei einer W-MRTD tatsächlich jemals zu Einsparungen im Speicherbedarf geführt hat. Die damit verbundene algorithmische Komplexität und der zusätzliche Zeitaufwand bei der Simulation lassen eine vollständig statische Anordnung der Entwicklungskoeffizienten der Waveleträume als wahrscheinlicher erscheinen. Unterstützt wird diese Vermutung dadurch, dass in [48] nicht der tatsächliche Speicherverbrauch der einzelnen Simulationen angegeben wird, sondern lediglich die Gitterauflösungen der Skalierungsräume und die Anzahl nicht-vernachlässigbarer Koeffizienten der Waveleträume.

Auch bei der Simulation sog. *elektrisch großer Probleme*, welche eine hohe Anzahl von Stützstellen in jeder Raumrichtung erfordern, stellt Speichermangel heutzutage i. d. R. kein nennenswertes Problem dar. Dies gilt oft sogar für den Einsatz handelsüblicher PCs. Bei der Entscheidung zwischen speicheroptimalen und zeitoptimalen Berechnungsverfahren fällt die Wahl daher meistens zugunsten geringerer Rechenzeiten aus, was in diesem Fall

dem statischen Speicherkonzept entspricht. Sind jedoch Strukturen zu simulieren, welche vorhandene Speicherbegrenzungen bei dem statischen Konzept überschreiten, bietet die dynamische Speicherverwaltung eine nutzbringende Alternative.

### 9.1.2 Daubechies-Funktionen

In [52] wird die Idee aus [19] aufgegriffen, die Maxwell'schen Gleichungen mit Hilfe von Daubechies-Funktionen zu diskretisieren, wobei jedoch nur  $D_2$ -Skalierungsfunktionen eingesetzt werden. Die Diskretisierung der Ableitungsoperatoren ist die gleiche wie bei der S-MRTD mit Battle-Lemarié-Skalierungsfunktionen. Die Materialoperatoren werden jedoch vollkommen anders behandelt: Mit Hilfe der Eigenschaft der verschobenen Interpolation (s. Punkt 4 auf Seite 95), die von Daubechies-Funktionen näherungsweise erfüllt wird, wird argumentiert, dass weder die Auswertung von Integralen, noch das Lösen von Matrixgleichungen erforderlich ist. Stattdessen werden die Berechnungen analog zu der FDTD durchgeführt: An jeder Position  $(i, j, k)$  des Raumes werden Permittivitätszahl und Feldkoeffizient einfach miteinander multipliziert. Im Unterschied zur FDTD gehören die Koeffizienten aber nicht zu Haar-, sondern zu  $D_2$ -Skalierungsfunktionen. Obwohl Materialverteilungen mit Daubechies-Funktionen anders dargestellt werden als mit Battle-Lemarié-Funktionen, soll die Klassifizierung als S-MRTD-Schema beibehalten werden.

Diese Vorgehensweise erfordert deutlich weniger Aufwand als die im vorangegangenen Abschnitt genannten Methoden. Sie entspricht jedoch keineswegs einer Approximation mittels eines Galerkin-Verfahrens. Sind Materialverteilung und Feldwerte jeweils im Raum  $\mathbf{V}_0$  approximiert, so ergibt ihre Multiplikation miteinander Funktionen, die mit dem Raum  $\mathbf{V}_0$  i. d. R. nicht mehr identisch sind und daher wieder eine Projektion auf  $\mathbf{V}_0$  erfordern. Dies kann beispielsweise durch die in [29] hergeleitete Technik bewerkstelligt werden. Dadurch wird gewährleistet, dass die Approximation so genau wie möglich ausfällt. Die Methode aus [52] entspricht dagegen einer weniger genauen Approximation.

Die Erweiterung auf mehrere Auflösungsstufen wird im Anhang von [52] besprochen. Für die Differenzialoperatoren werden explizite Gleichungen angegeben. Zu der Behandlung inhomogener Materialverteilungen wird

lediglich erwähnt, dass Integrale berechnet werden müssen. Ferner geht aus den Gleichungen hervor, dass das MRTD-Schema I eingesetzt wird (vgl. vorigen Abschnitt).

In [52] und [53] werden nur sehr einfache kubische Resonatoren simuliert und mit der FDTD verglichen. Dies ist auch in [54] der Fall, wobei jedoch auch  $D_3$ - und  $D_4$ -Skalierungsfunktionen eingesetzt werden und eine Dispersionsanalyse durchgeführt wird. Als Ergebnis wird festgehalten, dass mit  $D_4$ -Funktionen die geringste Dispersion erreicht werden kann. Die Simulationszeiten von FDTD und S-MRTD sind vergleichbar, doch kann mit der S-MRTD gröber diskretisiert und damit der Speicherbedarf reduziert werden.

Praxisnähere Strukturen wurden erstmalig in [55] und [56] simuliert. Auch in diesem Fall erweist sich die S-MRTD nicht als schneller im Vergleich zur FDTD, benötigt dafür aber weniger Speicher. Zur Demonstration der Genauigkeit werden vorwiegend Signale im Zeitbereich miteinander verglichen, was erfahrungsgemäß nicht unproblematisch ist. Der einzige Vergleich eines Streuparameters in [56] ergibt für die dort diskutierte zweidimensionale Struktur sogar, dass FDTD und S-MRTD gleich fein diskretisiert werden müssen.

Eine interessante Aussage ist in diesem Zusammenhang [55] zu entnehmen: Es wird bei der Simulation eines einfachen kubischen Resonators mit inhomogenem Dielektrikum festgestellt, dass die Daubechies-basierte S-MRTD weniger genaue Ergebnisse als die FDTD liefert, wenn die Abmessungen der Materialinhomogenität kleiner sind als die betrachtete Wellenlänge. Werden Strukturen simuliert, die größer als eine gewählte Wellenlänge sind, relativiert sich diese Aussage wieder. Eine Erklärung für dieses Phänomen wird nicht gegeben. Es steht jedoch zu vermuten, dass die eingangs beschriebene ungewöhnliche Approximation von Materialoperatoren hierfür verantwortlich zeichnet.

### 9.1.3 Haar-Funktionen

Die MRTD wurde erstmalig in [57] auf die Verwendung von Haar-Funktionen für eindimensionale Probleme angepasst. Während eine S-MRTD in diesem Fall dem gewöhnlichen FDTD-Verfahren nach Yee entspricht, bewirkt die Hinzunahme von Wavelets aus dem Raum  $\mathbf{W}_0$  die

Erhöhung der Auflösung. Da jedoch das MRTD-Schema I angewendet wird, kann entsprechend den Aussagen von Abschnitt 9.1.1 nur bedingt von einer Verdopplung der Auflösung gesprochen werden. Dennoch zeigen die Dispersionsdiagramme eine Verbesserung der Dispersionseigenschaften, wie es von einer FDTD zu erwarten ist, wenn sie feiner diskretisiert wird. Es werden jedoch keine Aussagen zu der Stabilität und dem verwendeten Zeitschritt gemacht. Dabei ist für die FDTD bekannt, dass sie im eindimensionalen homogenen Raum und bei Verwendung der maximal möglichen Zeitschrittweite keinerlei Dispersion aufweist [4].

Die Erweiterung auf sehr einfache zweidimensionale Probleme wird in [58] präsentiert. Es werden u. a. Simulationsergebnisse einer FDTD mit  $80 \times 80$  und mit  $40 \times 40$  Elementen, sowie einer MRTD mit  $40 \times 40$  Elementen miteinander verglichen. Die Autoren stellen die Vergleichbarkeit der  $80 \times 80$ -FDTD mit der  $40 \times 40$ -MRTD heraus, doch Abbildung 4 in [58] zeigt sehr deutlich, dass die  $40 \times 40$ -MRTD in großen Bereichen eher mit der  $40 \times 40$ -FDTD übereinstimmt. Als wesentlicher Grund hierfür ist wohl die Verwendung des MRTD-Schemas I auszumachen.

Die räumliche Adaptivität wird durch eine gezielte Vernachlässigung von Entwicklungskoeffizienten erreicht. [59] stellt einen Ansatz vor, bei dem sowohl absolute als auch relative Schwellwertkriterien zum Einsatz kommen. Demonstriert wird das Verfahren an Leitungsgleichungen, die mit Haar-Funktionen der Räume  $\mathbf{V}_0$  und  $\mathbf{W}_0$  diskretisiert wurden. Die diskretisierten Leitungsgleichungen sind mit einer eindimensionalen MRTD vergleichbar.

Abschnitt 7.2.5.2 stellte die in [26] veröffentlichte Möglichkeit vor, je nach Auflösungsstufe mit verschiedenen Zeitschrittweiten simulieren zu können. In [60] wird für eine eindimensionale Haar-basierte MRTD ebenfalls ein solches Verfahren präsentiert. Diese Methode erfordert es allerdings, die Entwicklungskoeffizienten der letzten drei Zeitschritte zwischenzuspeichern, um mit einer rückwärtigen Interpolation zweiter Ordnung Feldwerte von Zwischenzeitschritten berechnen zu können. Damit ließen sich nach Aussage der Autoren stabile Simulationen durchführen, wogegen lineare Interpolationen zu instabilen Simulationen führten.

Weiterhin stellt der Artikel Gleichungen für beliebige Auflösungsstufen nach dem MRTD-Schema I vor, bleibt jedoch auf Strukturen mit homogenem Dielektrikum beschränkt. Die Gleichungen enthalten u. a. eine künstliche Aufteilung, durch die Berechnungen effizienter durchgeführt werden

sollen. Ferner wird die Umsetzung elektrischer und magnetischer Wände behandelt. Wie schon bei der MRTD mit Battle-Lemarié-Funktionen, ist auch hier teilweise die Duplizierung von Entwicklungsfunktionen im Bereich der Wände sowie die Anwendung des Spiegelungsprinzips erforderlich.

Die Simulation eines dreidimensionalen Beispiels mit den (entsprechend erweiterten) vorgenannten Techniken ist in [61] zu finden. Die simulierte Flip-Chip-Struktur enthält Bereiche mit unterschiedlichen Dielektrika. Im Artikel wird beschrieben, dass in Bereichen mit dielektrischen Grenzflächen das Lösen einer Matrixgleichung erforderlich ist. Konkrete Gleichungen werden nicht angegeben. Die Simulationsergebnisse der MRTD werden mit Ergebnissen aus einer Finite-Elemente-Methode und Messungen verglichen.

Eine weitere dreidimensionale Struktur wird in [62] für einen Vergleich von FDTD und MRTD herangezogen. Demnach liefern beide Methoden gleichwertige Ergebnisse, wobei die MRTD deutlich weniger Speicher benötigt. Der Speichervorteil ergibt sich laut Artikel durch die adaptive Bestimmung der zu berechnenden bzw. zu vernachlässigenden Koeffizienten. Algorithmen für einen effizienten Umgang mit dynamisch benutzten Entwicklungskoeffizienten wurden in der Literatur zur Haar-basierten MRTD nicht gefunden. Falls eine solche Technik eingesetzt wurde, dann vermutlich die gleiche, die in [48] publiziert und in Abschnitt 9.1.1 beschrieben wurde. Allerdings wird in [62] – ähnlich wie in [48] – nicht der tatsächlich benötigte Speicherbedarf aufgeführt, sondern es wird nur die Anzahl der Entwicklungskoeffizienten des Skalierungsraumes der MRTD (zu den Waveleträumen werden keine Anzahlen wiedergegeben) mit der Koeffizientenanzahl der FDTD verglichen.

Weiterhin wird herausgestellt, dass die MRTD nur ein Viertel der Rechenzeit der FDTD benötigt. Alle Leitungen haben die gleiche Breite und werden daher gleichmäßig diskretisiert. Mit einer Auflösung von zehn Yee-Zellen über die Leiterbreite wird die FDTD jedoch recht fein diskretisiert. Zum Vergleich: Die Leiter des in Abschnitt 8.2.2 simulierten Mikrostreifenleitungsfilters hatten im Querschnitt eine Breite von lediglich sechs Zellen. Die Simulation koplanarer Spulen erfolgte in [3] sogar mit nur vier Zellen breiten Leitern.

Da die räumliche Auflösung einen sehr starken Einfluss auf die Rechen-dauer hat, wird im Rahmen der erforderlichen Genauigkeit in der Praxis

stets versucht, eine Struktur so grob wie möglich zu diskretisieren. Die Verwendung einer höheren Genauigkeit steigert nicht nur die Simulationsdauer, sondern sie macht auch einen direkten Vergleich mit Wavelet-basierten Verfahren schwieriger. Der Grund dafür ist, dass Feldvariationen zwischen benachbarten Zellen abnehmen, wenn die Auflösung erhöht wird. Je feiner ein kontinuierliches Signal diskretisiert wird, desto geringer werden die Unterschiede zwischen benachbarten Stützstellen. Das bedeutet gleichzeitig, dass sich das Signal besser komprimieren lässt. Höher aufgelöste Strukturen sind daher bei Wavelet-basierten Verfahren in dem Sinne vorteilhaft, dass sie eine stärkere Kompression und eine entsprechend reduzierte Rechenzeit erlauben. Bei einer größeren Auflösung ist zwar eine höhere Zeitschrittweite erlaubt, doch die Möglichkeiten zur Kompression, d. h. zur Vernachlässigung von Entwicklungskoeffizienten sinkt. Bei gleicher Diskretisierungsweite erweist sich die MRTD gegenüber der FDTD daher umso vorteilhafter, je höher die räumliche Auflösung gewählt wird.

Eine gänzlich andere Vorgehensweise zur Herleitung einer MRTD mit Haar-Funktionen wird in [63] präsentiert. Eine Yee-Zelle wird mit Haar-Skalierungsfunktionen und -Wavelets, deren Träger mit den Abmessungen der Yee-Zelle übereinstimmen, in acht Unterzellen aufgeteilt. Die Autoren benennen die zugehörige Verachtfachung von *Freiheitsgraden* als Grund dafür, dass die MRTD generell doppelt so grob aufgelöst werden kann wie eine FDTD. Eigentlich geschieht jedoch nichts anderes, als dass durch die Hinzunahme von Wavelets aus dem Raum  $\mathbf{W}_0$  die Auflösung verdoppelt wird. Schaut man sich die Gleichungen genauer an, so ist zu erkennen, dass die Verdopplung der Auflösung nach dem MRTD-Schema I erfolgt.

Bisher war für einstufige W-MRTD-Schemata immer die Herleitung eines eigenen Stabilitätskriteriums erforderlich. Die Autoren von [63] wenden demgegenüber für die W-MRTD das gleiche Kriterium an wie für die FDTD, welches identisch zu dem der S-MRTD ist. Es wird demnach nicht berücksichtigt, dass eine Verdopplung der räumlichen Auflösung erfolgt. Die Gültigkeit dieses Ansatzes wird nicht hergeleitet, sondern anhand von Simulationen demonstriert. Die dabei angegebenen Zeitschrittweiten für die Simulationen liegen nach Aussage der Autoren sehr nahe an dem Stabilitätskriterium. Diese Aussage lässt sich allerdings nicht mit den Angaben aus dem Artikel in Einklang bringen. Für die FDTD-Simulation einer koplanaren Spule beispielsweise wird  $\Delta t = 0,21535$  ps angesetzt, was das 0,98-fache von  $\Delta t_{\text{FDTD}}$  sein soll. Nach Gleichung (2.26) ergibt sich

jedoch  $\Delta t_{\text{FDTD}}$  zu etwa 0,68 ps.

Eine ähnliche Abweichung ist bei den Zeitschrittweiten für die (gröber diskretisierte) MRTD zu finden. Dadurch können lediglich eingeschränkte Aussagen zur Stabilität des Verfahrens getroffen werden. Erfordert die Erhöhung der Auflösung in allen bisher vorgestellten MRTD-Varianten die Reduktion der Zeitschrittweite zur Wahrung der Stabilität, so sollte das eigentlich auch bei der Variante aus [63] zu erwarten sein. Verifiziert werden können Stabilitätsgrenzen jedoch nur durch Simulationen knapp unterhalb der maximal erlaubten Zeitschrittweiten.

Während die Wavelets zu einer Verdopplung der Auflösung führen, müssen Materialverteilungen in [63] mit der gröberen Auflösung des Skalierungsraumes diskretisiert werden. Das bedeutet, dass die Materialparameter in den o. g. acht Unterzellen identisch sein müssen.

Inhomogenitäten in der Materialverteilung werden auf zwei verschiedene Arten behandelt. In dem Fall mit einer einfachen Approximation wird von einem anisotropen Verhalten dielektrischer Grenzflächen und dem Auftreten von Instabilitäten berichtet. Der zweite Fall basiert auf einer genaueren Behandlung der Grenzflächen und erfordert an diesen Stellen das Lösen einer Matrixgleichung. Damit konnten dann keine Instabilitäten mehr beobachtet werden.

Die Behandlung elektrisch ideal leitfähiger Wände erfolgt in [63] zunächst durch die Approximation von Ableitungen durch Vorwärts- und Rückwärtsdifferenzen. Es wird ferner beschrieben, wie statt dessen auch Lagrange-Interpolationen eingesetzt werden können. Letztere erhöhen die Genauigkeit der Simulationsergebnisse.

Im Vergleich zu den FDTD-Simulationen benötigt die MRTD nach [63] stets nur etwa die Hälfte der Rechenzeit. Die Ergebnisse beider Verfahren stimmen gut miteinander überein und weichen nur bei höheren Frequenzen teilweise voneinander ab. Diese Variante der MRTD erlaubt ferner die Anwendung abgestufter Gitter, was für die zuvor untersuchten MRTD-Varianten nicht bestätigt werden kann.

Die beiden vorgestellten Methoden zur Diskretisierung der Maxwell'schen Gleichungen mit Haar-Wavelets führen auf sehr ähnliche Gleichungen. Unterschiede sind im Wesentlichen nur bei der Behandlung von Materialeigenschaften auszumachen. Betrachtet man den einfachen Fall eines

unendlich ausgedehnten homogenen verlustlosen Raumes, so zeigen die Gleichungen bei beiden Ansätzen eine interessante Gemeinsamkeit, wenn außer den Skalierungsfunktionen nur noch Wavelets des Raumes  $\mathbf{W}_0$  hinzugenommen werden, wie es häufig praktiziert wird. In diesem Fall sind die Entwicklungskoeffizienten des Skalierungsraumes und die des Waveletraumes vollkommen unabhängig voneinander. Das bedeutet, dass zur Berechnung des  $\mathbf{V}_0$ -Anteils des elektrischen Feldes nur dieser Anteil selbst sowie  $\mathbf{V}_0$ -Anteile des magnetischen Feldes herangezogen werden. Analoges gilt für die  $\mathbf{W}_0$ -Anteile sowie die Berechnung des magnetischen Feldes.

In [59] und [63] wird dies als Entkopplung der Räume bezeichnet. Lediglich an den Stellen, wo Materialparameter variieren, sowie an elektrischen und magnetischen Wänden koppeln die beiden Räume miteinander. Die Eigenschaft der Entkopplung wird in [64] kritisch untersucht. Demnach erhöhen Wavelets zwar die räumliche Auflösung der Felder, tragen jedoch nicht zu einer Verbesserung der Dispersion bei. Ferner wird die Entkopplung als Grund dafür ausgemacht, dass für die S-MRTD die gleiche Zeitschrittweite wie für die W-MRTD in [63] angesetzt werden kann. Beide Aussagen stehen damit im Widerspruch zu Stabilitäts- und Dispersionsaussagen, die z. B. in [57, 44] zu finden sind.

Eine allgemeinere Betrachtung führt in [64] zu dem Ergebnis, dass der jeweils letzte Waveletraum, d. h. der mit der höchsten räumlichen Auflösung, keinen Beitrag zur Verbesserung der Genauigkeit beisteuert. Auch die Verkopplung der Räume durch elektrische Wände führt in der Praxis zu keiner Verbesserung der Situation.

An dieser Stelle soll nochmals betont werden, dass für die Haar-basierte MRTD bisher nur das MRTD-Schema I zur Anwendung kam. Die Einführung des MRTD-Schemas II wird erstmalig in der bereits an anderer Stelle genannten Veröffentlichung [44] für Battle-Lemarié- und Haar-Funktionen demonstriert. Die Aussagen aus [64] und [44] zum Dispersionsverhalten stimmen zumindest soweit überein, dass das MRTD-Schema I eine geringere Abnahme der Dispersion bewirkt, als die räumliche Auflösung der Felder erwarten lässt.

In Abschnitt 8.2 wurde demonstriert, dass Inhomogenitäten die numerische Komplexität Wavelet-transformierter Operatoren stark ansteigen lassen können. Das gilt insbesondere bei der Verwendung von PMLs als absorbierende Randbedingungen. Um diese Probleme umgehen zu

können, wurde die Teilraum-WT eingeführt. In [65] wird das Problem großer Inhomogenitäten ebenfalls aufgegriffen und für die Haar-basierte MRTD eine andere Lösung präsentiert. Auch hier wird zwischen einem untransformierten und einem transformierten Feldraum unterschieden. Ersterer ist der gewöhnliche FDTD-Raum, was auf die Verwendung von Haar-Funktionen zurückzuführen ist. Der transformierte Bereich entspricht dann dem bekannten MRTD-Raum. Verbunden werden die beiden Bereiche durch eine WT sowie eine IWT, die im Bereich der Grenzfläche ausgeführt wird.

### 9.1.4 CDF-Funktionen

Die bisher beschriebenen Funktionenklassen erfüllen sehr strenge Orthogonalitätsbeziehungen. Diese Strenge schränkt die Möglichkeiten zur Konstruktion von Wavelets mit bestimmten gewünschten Eigenschaften stark ein. Die Battle-Lemarié-Funktionen beispielsweise sind symmetrisch, besitzen jedoch einen unendlich ausgedehnten Träger. Daubechies-Funktionen dagegen haben einen kompakten Träger, sind jedoch nicht symmetrisch. Die Einschränkung durch die Orthogonalitätsbeziehungen verhindert die Entwicklung symmetrischer Wavelets mit kompaktem Träger [12].

Geht man von der Forderung nach Orthogonalität über zur sog. *Biorthogonalität*, lassen sich symmetrische Wavelets mit kompaktem Träger herleiten. In [13] wird eine Funktionenklasse hierzu vorgestellt, die in der Literatur unter der Abkürzung *CDF* (Cohen-Daubechies-Feauveau) bekannt ist. Biorthogonale Funktionenklassen verwenden für die WT andere Funktionen und damit auch andere Koeffizienten, als für die IWT. Daher wird die Ordnung bei CDF-Funktionen – anders als bei Daubechies-Funktionen – nicht durch einen Parameter, sondern durch zwei Parameter bestimmt. Entsprechend werden die möglichen Kombinationen mit  $CDF(2,2)$ ,  $CDF(2,4)$ ,  $CDF(2,6)$ ,  $CDF(3,3)$  usw. bezeichnet.

Mit den von der MRTD bekannten Entwicklungsmethoden werden in [66] die Maxwell'schen Gleichungen mit Hilfe von CDF-Funktionen der Räume  $\mathbf{V}_0$  und  $\mathbf{W}_0$  diskretisiert. Insbesondere aus der ausführlicheren Veröffentlichung [67] geht hervor, dass hierbei das MRTD-Schema I angewendet wurde. In beiden Artikeln fällt es auf, dass die maximal erlaubte Zeitschrittweite beim Übergang von der S-MRTD auf die einstufige W-MRTD

mit CDF-Funktionen weniger stark abnimmt, als dies bei der Verwendung von Battle-Lemarié-Funktionen der Fall ist. Dabei bezieht sich der Vergleich noch auf eine Battle-Lemarié-MRTD nach MRTD-Schema I. Hierzu ist anzumerken, dass das MRTD-Schema II von einer anderen Autorengruppe veröffentlicht wurde, und zwar zeitgleich mit [66] und nach der Einreichung von [67].

Der kompakte Träger der CDF-Funktionen erlaubt es, genau wie bei der Anwendung von Daubechies-Funktionen mit einer genau begrenzten Zahl an Stützstellen zur Approximation der Ableitungsoperatoren auszukommen. Je nach gewählter Ordnung der CDF-Funktionen ergeben sich bis zu einem Drittel weniger Stützstellen, als sie üblicherweise bei Battle-Lemarié-Funktionen vonnöten sind.

In [66] wird ferner herausgestellt, dass die CDF-MRTD eine gröbere Diskretisierung von Simulationsstrukturen im Vergleich zur FDTD erlaubt. Demonstriert wird dies an einem einfachen Beispiel. Allerdings scheint die Auflösung bei der FDTD mit 80 Stützstellen für eine mittlere Wellenlänge recht hoch gewählt zu sein.

Die Daubechies-basierte MRTD macht sich die näherungsweise erfüllte Bedingung der verschobenen Interpolation zunutze, um inhomogene Materialverteilungen einfacher diskretisieren zu können. CDF-Skalierungsfunktionen erfüllen diese Bedingung nach [66] sogar exakt, weshalb die Autoren die gleiche Methode zur Darstellung von Materialverteilungen ansetzen. Zu dieser Vorgehensweise gelten jedoch auch hier die bereits in Abschnitt 9.1.2 angemerkten Bedenken, dass dadurch nicht mehr ein durchgehend nach dem Galerkin-Verfahren hergeleitetes Verfahren vorliegt, sondern eine schwächere Approximation insbesondere im Bereich dielektrischer Grenzflächen zur Anwendung kommt. Die am Ende von Abschnitt 9.1.2 beschriebenen numerischen Effekte lassen sich daher aller Wahrscheinlichkeit nach auch bei der CDF-basierten MRTD finden.

Stabilitätsbetrachtungen und Dispersionsverhalten werden ausführlich in [67] behandelt. Mit Dispersionsdiagrammen wird der bereits beschriebene Effekt aufgezeigt, dass die Battle-Lemarié-MRTD (nach MRTD-Schema I) ein schlechteres Verhalten aufweisen kann als die FDTD. Mit CDF-Funktionen dagegen fallen Dispersionseffekte stets schwächer aus als bei der FDTD. Das gilt allerdings nur unter der Voraussetzung, dass die Zeitschrittweite bei der CDF-MRTD deutlich unterhalb des erlaubten Maximums liegt.

Die Autoren vergleichen CDF-Funktionen der eingangs genannten verschiedenen Ordnungen unter dem Gesichtspunkt der Genauigkeit und der numerischen Effizienz miteinander. Sie kommen zu dem Schluss, dass CDF(2,2) den besten Kompromiss darstellt. Eine andere Veröffentlichung, welche die Kompressionseigenschaften von CDF-Funktionen für bestimmte Signale untersucht, kommt zu dem Ergebnis, dass CDF(3,1) am effizientesten ist [68]. Durch die Vielzahl der Möglichkeiten lässt sich wohl erst nach und nach feststellen, welche Parameterkombination für praxisnahe Probleme am praktikabelsten ist.

### 9.1.5 Fazit

Abgesehen von der Haar-MRTD erlauben alle MRTD-Varianten die Reduktion der numerischen Dispersion, was vor allem für die Simulation elektrisch großer Strukturen von Bedeutung sein kann. Die meisten zitierten Veröffentlichungen benutzen diese Eigenschaft jedoch, um eine gröbere Diskretisierung erreichen zu können, als dies mit der klassischen FDTD möglich wäre. Die vergleichenden Simulationen lassen allerdings kein klares Bild erkennen, ob dies tatsächlich zu deutlich geringeren Simulationszeiten gegenüber der FDTD führt.

Die Behandlung von Materialinhomogenitäten und elektrischen sowie magnetischen Wänden scheint in allen Varianten die größten Probleme zu bereiten. Das gilt vor allem dann, wenn zusätzlich zu den Skalierungsfunktionen Wavelets eingesetzt werden, um die Auflösung lokal zu erhöhen. Die exakte Behandlung der Verkopplungen zwischen den Skalen erfordert das Lösen von Matrixgleichungen. Selbst für homogene Gebiete koppeln die Skalen in den Ableitungsoperatoren miteinander. Die Diskretisierung der Operatoren erfordert dabei die numerische Auswertung von Integralen. Lediglich Haar-basierte MRTD-Verfahren bilden in diesen Punkten eine Ausnahme bzw. sind einfacher zu handhaben.

Eine abschließende Bewertung der MRTD kann an dieser Stelle nicht gegeben werden. Das ist auch unter dem Aspekt zu sehen, dass das MRTD-Schema II noch nicht auf alle Funktionenklassen übertragen wurde und bisher zu wenig Vergleichsmöglichkeiten zum MRTD-Schema I bestehen. Die in den vergangenen Abschnitten aufgezeigten Stärken und Schwächen können jedoch dazu dienen, Möglichkeiten für die Weiterentwicklung der

in dieser Arbeit vorgestellten Simulationstechniken auszuarbeiten und den praktischen Nutzen besser einzuschätzen.

Hier wäre vor allem der Einsatz von CDF-Funktionen zur Transformation in den Waveletbereich zu erwähnen. Diese stellen eine interessante Alternative zu den Daubechies-Funktionen dar. Auch ihre dispersionsmindernden Eigenschaften bei der Diskretisierung der Ableitungsoperatoren lassen sich bei Bedarf einbringen. Von Vorteil ist hierbei, dass die in der vorliegenden Arbeit behandelten Techniken zur Transformation eines Simulationsverfahrens es lediglich erfordern, dass es linear ist.

Die in der MRTD übliche Vorgehensweise, vom Skalierungsraum ausgehend die Auflösung durch die Hinzunahme von Wavelets zu erhöhen, erweist sich nach dem bisher Gesagten als problematisch. Der in dieser Arbeit beschrittene Weg scheint demgegenüber sehr viel schematischer zu sein, da die Operatoren des gewünschten Simulationsverfahrens lediglich für die höchste räumliche Auflösung entwickelt werden müssen. Die Darstellung der Operatoren in den einzelnen Skalen, die Verkopplung der Skalen und die Behandlung von Materialinhomogenitäten werden bei der Transformation in den Waveletbereich automatisch berücksichtigt. Insofern könnte die Technik der WT von Operatoren auch für die MRTD von Nutzen sein.

Abschließend soll noch erwähnt werden, dass sämtliche durchgesehenen Veröffentlichungen zur W-MRTD ausschließlich homogene Tensorprodukt-Basen berücksichtigen. Der Einsatz separabler MSA-Basen wird nicht diskutiert. Da sich letztere in dieser Arbeit als vorteilhaft erwiesen, könnte dies durchaus auch für die MRTD zutreffen. Zumindest die Haar-basierte MRTD dürfte sich mit überschaubarem Aufwand entsprechend modifizieren lassen. Bei den anderen Funktionenklassen ist es fraglich, ob separable MSA-Basen jemals eingesetzt werden, da bisher niemand über die einstufige W-MRTD hinausgegangen ist.

## 9.2 Erweiterungsmöglichkeiten

In den folgenden Abschnitten soll aufgezeigt werden, um welche Möglichkeiten die in dieser Arbeit entwickelten Simulationstechniken erweitert werden können. Zuerst sollen einige bedeutende Entwicklungen im Bereich der Wavelet-Theorie präsentiert werden. Anschließend werden Ideen

vorgestellt, die sich im Laufe der Arbeit aus den gesammelten Erfahrungen entwickelt haben.

### 9.2.1 Entwicklungen in der Wavelet-Theorie

In Ergänzung zu den bisher vorgestellten Wavelets und Algorithmen sollen im Folgenden einige weitere Möglichkeiten aufgeführt werden, die in der Literatur zu finden sind. Aus Gründen des Umfangs können die wesentlichen Ideen jedoch lediglich kurz skizziert werden. Tiefer gehende Informationen können den jeweils zitierten Publikationen entnommen werden.

Einen Überblick über verschiedene Arten von Wavelets geben [12, 69, 70]. So werden in [69] neben den bereits erwähnten orthogonalen und biorthogonalen Wavelets auch sog. *semiorthogonale* Wavelets vorgestellt, die in ihren Eigenschaften eine Art Kompromiss zwischen den beiden erstgenannten Funktionstypen darstellen. In Abschnitt 3.2.3 wurde erwähnt, dass Ingrid Daubechies unterschiedliche Wavelets mit kompaktem Träger entwickelt hat. Einige davon werden in [69] kurz klassifiziert. Ferner werden Wavelets besprochen, die speziell der Darstellung von Signalen auf einem abgeschlossenen Intervall dienen. Auf die Relevanz der letztgenannten Wavelets soll etwas näher eingegangen werden.

Von dem physikalischen Raum lässt sich stets nur ein endlicher Bereich simulieren. Insofern sollten auf Intervalle beschränkte Wavelets besonders dazu geeignet sein, Feldverläufe im Simulationsbereich darzustellen. Insbesondere die Behandlung der Randwerte an den Intervallgrenzen sollte sich damit transparenter gestalten lassen. Das trifft allerdings nicht auf solche Wavelets zu, die durch einen Periodisierungsprozess künstlich auf ein endliches Intervall angepasst wurden. Die diskrete periodische WT (DPWT), welche in dieser Arbeit zur Anwendung kam, impliziert einen solchen Periodisierungsprozess. Am besten lässt sich dies anhand von Gleichung (3.56) auf Seite 35 ansehen: Die letzte Zeile der Transformationsmatrix zeigt, wie die Waveletkoeffizienten den Anfang des Intervalls mit seinem Ende gewissermaßen verbinden. Dadurch können Funktionswerte am Intervallende die Werte am Intervallanfang beeinflussen, wenn in diesen Bereichen durch die Vernachlässigung von Entwicklungskoeffizienten im Waveletbereich eine zu grobe Approximation stattfindet. Daher

wurde die Approximation bei den Simulationen in dieser Arbeit auch durch die Einrichtung von Detailbereichen an den Intervallgrenzen unterbunden.

Die Benutzung von Wavelets, welche nicht zu einer Verkopplung der Intervallgrenzen führen, hat unter diesem Aspekt einige Vorteile. Allerdings wird dies mit dem Nachteil erkauft, dass die WT etwas komplizierter wird. Dennoch gibt es für diese Funktionen schnelle Transformationsalgorithmen. [71] geht ausführlicher auf diese Themen ein. Eine Erweiterung der Daubechies-Funktionen zweiter Ordnung um sog. *Randfunktionen* zur Trennung der Intervallgrenzen wird u. a. in [12] vorgestellt.

Eine Erweiterung der klassischen WT der MSA führt zu sog. *Wavelet packets*. Dazu wird die Idee der Aufspaltung eines Skalierungsraumes  $\mathbf{V}_\nu$  in die Räume  $\mathbf{V}_{\nu+1}$  und  $\mathbf{W}_{\nu+1}$  ebenfalls auf die Waveleträume angewandt.  $\mathbf{W}_\nu$  wird auf diese Weise abgebildet auf  $\mathbf{W}_{\nu,1}$  und  $\mathbf{W}_{\nu,2}$ . Auch diese Räume können weiter aufgeteilt werden. Damit erhält man eine zum Skalierungsraum symmetrische Vorgehensweise. Der größte Vorteil hierbei ist, dass man sehr flexibel bei der Wahl ist, ob ein Skalierungs- oder Waveletraum weiter transformiert werden soll oder nicht. Durch diese Freiheit lassen sich Signale noch effizienter komprimieren. Für eine anschauliche Darstellung sei an dieser Stelle wieder auf [69] verwiesen.

Durch die Einführung der Biorthogonalität bei Wavelets konnten die Möglichkeiten zur Konstruktion von Wavelets mit gewünschten Eigenschaften gegenüber den orthogonalen Wavelets erweitert werden. Dennoch ist es nicht möglich, Wavelets zu entwickeln, die gleichzeitig die Bedingungen nach Isotropie, Orthogonalität und kompaktem Träger erfüllen. In [70] wird erwähnt, dass sog. *Multiwavelets* zu diesem Problem eine Lösung bieten können. Die zusätzlichen Freiheitsgrade kommen daher, dass statt lediglich einer Skalierungsfunktion und einem Wavelet mehrere Skalierungsfunktionen und mehrere Wavelets zur Konstruktion einer Basis des  $\mathbf{L}_2(\mathbb{R})$  herangezogen werden. Diese Erweiterung ist ausführlicher in [72] nachzulesen. Eine weitere interessante Variante sind *M-Band Multiwavelets* [73]. Solche Funktionensysteme erlauben die Erzeugung von Skalierungsfunktionen mit disjunkten Trägern.

Eine gänzlich andere Art der Konstruktion von Wavelets ist mit dem sog. *Lifting Scheme* möglich [74]. Die Erzeugung bzw. Berechnung von Wavelets mit gewünschten Eigenschaften erfordert üblicherweise die Anwendung der Fouriertransformation und Analysen im Ortsfrequenzbereich.

Mit dem Lifting Scheme dagegen werden Wavelets vorwiegend im Ortsbereich konstruiert. Neben der besseren Anschauung erlaubt das Verfahren die Beschleunigung der WT und kann diese sogar in dem gleichen Speicherbereich durchführen, der von dem zu transformierenden Signal eingenommen wird (sog. *in-place calculation*). Vor allem aber lassen sich sehr einfach Wavelets für begrenzte Gebiete und sehr unregelmäßige (irreguläre) Diskretisierungsgitter erzeugen.

Das Lifting Scheme erlaubt zwar eine Signalanalyse auf verschiedenen Skalen bzw. Auflösungsstufen, doch diese Analyse beruht im Gegensatz zu der MSA nicht mehr auf der Translation und Dilatation von Funktionen. Genauer gesagt unterliegt das Lifting Scheme nicht mehr den Einschränkungen, die mit festen Translations- und Dilatationsregeln einhergehen. Zusammen mit weiteren Vorzügen, die Wavelets aus dem Lifting Scheme mit sich bringen, hat dies zu der Bezeichnung *Wavelets der zweiten Generation* (second generation wavelets) geführt.

Während [74] das Prinzip des Lifting Schemes nur recht kurz beschreibt, geht [75] auf einige Grundlagen ein und demonstriert auch die Konstruktion einiger Wavelets. Eine weitere Formalisierung und Verallgemeinerung ist in [76] zu finden. Hier wird demonstriert, mit welchen Methoden biorthogonale Wavelets mit gewünschten Eigenschaften konstruiert werden können. Dabei wird von den Funktionen einer bestehenden MSA ausgegangen, die dann mit Hilfe einer einfachen Transformation verändert werden. Auf diese Weise können in mehreren Schritten Wavelets mit einer beliebig hohen Anzahl verschwindender Momente entwickelt werden. Ferner wird in dem Artikel kurz beschrieben, wie sich Wavelets auf dem Intervall, gewichtete Wavelets, adaptive Wavelets und nichtseparable mehrdimensionale Wavelets mit dem Lifting Scheme generieren lassen. Auch Wavelet packets und  $M$ -band Wavelets lassen sich durch das Lifting Scheme erzeugen. Es existiert ferner Kursmaterial zum Lifting Scheme, welches demonstriert, wie sich verschiedene Auswahlen an interpolierenden Eigenschaften zur Konstruktion von Wavelets nutzen lassen [77].<sup>3</sup>

Die bisher beschriebenen Entwicklungen in der Wavelet-Theorie beziehen sich fast ausschließlich auf Wavelets mit verschiedenen Eigenschaften und ihrem Potenzial zur Kompression von Signalen. Mit der Kompression von Operatoren setzen sich die Beiträge [78, 79] auseinander.

---

<sup>3</sup>Die genannten Artikel sind auch im Internet verfügbar:  
<http://cm.bell-labs.com/who/wim/papers/index.html> .

Zwei verschiedene Formen der Darstellung im Waveletbereich werden für Calderon-Zygmund-Operatoren und Pseudo-Differenzial-Operatoren vorgestellt und untersucht. Betrachtet werden zunächst nur Operatoren für eindimensionale Signale der Länge  $N$ , was auf Operatormatrizen der Größe  $N \times N$  führt. In der sog. *Standard-Form* haben transformierte Operatoren eine numerische Komplexität der Ordnung  $O(N \log(N))$ . Auch die in dieser Arbeit verwendete WT von Operatoren entspricht der Standard-Form. Die in den genannten Artikeln erstmalig vorgestellte *Nicht-Standard-Form* erlaubt dagegen eine Reduktion der Ordnung auf lediglich  $O(N)$ .

Erreicht wird die Einsparung durch eine eigentlich redundante Datenerhaltung: Während normalerweise zur Darstellung eines Signals im Waveletbereich die Koeffizienten aus den  $j$  Waveleträumen  $\mathbf{W}_1, \dots, \mathbf{W}_j$  und dem Skalierungsraum  $\mathbf{V}_j$  genügen, erfordert die Nicht-Standard-Form außer den  $j$  Waveleträumen auch  $j$  Skalierungsräume  $\mathbf{V}_1, \dots, \mathbf{V}_j$ . Das verdoppelt in etwa den Speicherbedarf bei eindimensionalen Signalen, wenn  $j \gg 1$  ist. Im Falle von  $j = 1$ , d. h. einer einstufigen WT, sind die beiden Formen identisch.

Laut den Angaben in [78] lässt sich das Verfahren in analoger Weise für  $n$ -dimensionale Probleme anwenden, jedoch werden keine weiteren Angaben hierzu gemacht. Dennoch kann davon ausgegangen werden, dass die Speicherzunahme im eindimensionalen Fall nicht notwendigerweise eine Erhöhung des Speicherbedarfs auf das  $2^n$ -fache im  $n$ -dimensionalen Fall bedeutet. Hierbei ist zu berücksichtigen, dass das Verfahren nicht einfach auf jede Dimension getrennt angewendet werden darf, so wie es beispielsweise bei einer WT mit homogenen Tensorprodukt-Basen der Fall ist. Es ist vielmehr das Konzept der Wavelet- und Skalierungsräume einer MSA für den  $n$ -dimensionalen Raum im Auge zu behalten [12]. Da homogene Tensorprodukt-Basen nicht die Definition einer MSA erlauben, sind separable oder allgemeine MSA-Basen zu benutzen. Geht man nun von dem Konzept der Räume einer höherdimensionalen MSA aus, so würde sich der Speicherbedarf bei  $n$  Dimensionen ebenfalls lediglich verdoppeln. Da in den meisten Fällen die Rechenzeit und nicht der Speicher der begrenzende Faktor für eine Simulation ist, könnte die Nicht-Standard-Form insbesondere bei elektrisch großen Strukturen und mehrstufigen Transformationen eine interessante Möglichkeit zur Reduktion des numerischen Aufwands bieten.

## 9.2.2 Konzeptionelle Modifikationen

Anhand einiger Charakteristika der in dieser Arbeit vorgestellten Listenbasierten WT der FDTD soll im Folgenden aufgezeigt werden, welche strukturellen bzw. algorithmischen Veränderungen flexiblere oder sogar schnellere Simulationen erlauben. Die folgenden Eigenschaften und Regeln sollen betrachtet und anschließend näher erläutert werden:

1. Nur ein Detailgitter für alle Felder und Komponenten:  
Aus den angegebenen Detailbereichen lässt sich ein Detailgitter für den Waveletbereich berechnen. Dieses wird ohne weitere Unterscheidung auf alle Komponenten des elektrischen und des magnetischen Feldes angewandt.
2. Detailbereiche nur für die höchstmögliche Auflösung:  
Die Definition der Detailbereiche führt dazu, dass diese Stellen trotz Operatorkompression mit der gleichen Genauigkeit und damit Auflösung berechnet werden wie im untransformierten Fall. In einer Umgebung dieser Bereiche bewirkt der Algorithmus implizit, dass *Übergangsbereiche* entstehen, in denen die Berechnung mit einer „mittleren“ Genauigkeit abläuft. Übergangsbereiche können jedoch nicht manuell vorgegeben werden.
3. Das Detailgitter ist statisch:  
Aus den Detailbereichen wird das Detailgitter zu Beginn einer Simulation berechnet und danach nicht mehr verändert. Eine zeitliche Variation in der räumlichen Auflösung findet nicht statt.
4. Zusammenfassung aller Teilräume:  
Ein Bildpunkt im Waveletbereich hängt i. A. von Urbildpunkten aus verschiedenen Waveleträumen und dem Skalierungsraum ab. Die Listen-Darstellung unterscheidet jedoch nicht zwischen Teilräumen, sondern bewirkt die Berechnung des Bildpunktes in einer einzelnen Summe.
5. Sämtliche Operatoren werden über Listen dargestellt:
  - Die Transformation eines Operators geschieht dadurch, dass der Operator selber sowie die WT und die IWT mit Listen

dargestellt werden. Anschließend werden die Operatoren miteinander verkettet.

- Bei einer Teilbereichs-WT werden auch die Abbildungen in den nicht zu transformierenden Bereichen (z. B. PMLs) durch Listen dargestellt.

6. Faktor- und Relativvektor-Folgen bilden Einheiten:

Jeder Faktor einer  $O^2$ -Liste ist jeweils mit einem Relativvektor direkt gepaart.

Die Bedeutung von Punkt 6 und eine mögliche Umgehung der damit verbundenen Probleme wurde bereits in Abschnitt 8.2.3 behandelt (Trennung von Vektor- und Faktorfolgen; Einführung von Delta-Vektorfolgen). Auch Punkt 1 wurde in Abschnitt 7.3.6.3.3 kurz angesprochen. Es ist offensichtlich, dass die Möglichkeit zur separaten Definition von Detailbereichen für die einzelnen Feldkomponenten nicht nur mehr Flexibilität mit sich bringt, sondern auch den i. A. unterschiedlich starken Gradienten der Feldkomponenten mehr Rechnung trägt. Dadurch sollte eine gezieltere Kompression der Operatoren erreichbar sein, die ja bislang an der Definition von solchen Bereichen ausgerichtet ist, wo starke räumliche Feldgradienten erwartet werden.

Zu Punkt 2 ist zu sagen, dass sich das Detailgitter-Verfahren nur daran orientiert, dass in ausgewählten Bereichen keine Kompression erfolgen soll. Das bedeutet, dass Berechnungen im Rahmen der gewählten Grunddiskretisierung mit der höchstmöglichen Genauigkeit durchzuführen sind. Bedingt durch die Trägerbreite der Wavelets der verschiedenen Auflösungsstufen bewirkt das Verfahren, dass die Kompression mit zunehmendem Abstand von den Detailbereichen ansteigt. Es ist mit diesem Verfahren nicht möglich, den Übergangsbereich anderweitig zu steuern. Dabei wäre es durchaus vorteilhaft, die Abnahme der Genauigkeit in gewissen Grenzen vorgeben zu können. Es ist als sehr wahrscheinlich zu erachten, dass sich das Verfahren in dieser Richtung erweitern lässt. Dadurch ließen sich dann auch gezielt Bereiche definieren, die grundsätzlich mit einer „mittleren“ Genauigkeit simuliert werden sollen.

Die in Punkt 3 angesprochene statische Steuerung der räumlichen Auflösung erlaubt eine effizientere Kompression der Operatoren, als es mit der in Abschnitt 7.2.5.1 beschriebenen adaptiven Auflösung möglich wäre. Die

Gründe hierfür wurden in Abschnitt 7.3.6 im Einzelnen aufgeführt. Dies wird jedoch mit dem Nachteil erkaufte, dass die Kompression vor Beginn einer Simulation durch eine sorgsame Wahl von Detailbereichen, Wavelets und Transformationstiefen festzulegen ist. Eine ungünstige Wahl beeinträchtigt die Genauigkeit und lässt sich nicht während der Simulation korrigieren. Das kann bei der adaptiven Auflösung nicht passieren, sofern nicht ein zu hoher Schwellenwert für die Vernachlässigung von Feldwerten angesetzt wird.

Beide Verfahren lassen sich wie folgt kombinieren:  $\tilde{\mathcal{T}}$  sei ein Wavelet-transformierter und unkomprimierter Operator. Die Kompression mit dem Detailgitter-Verfahren ergibt dann den Operator  $\tilde{\mathcal{T}}'$ . Die hierdurch entfallenden Abbildungselemente lassen sich in einem weiteren Operator  $\tilde{\mathcal{T}}''$  darstellen:  $\tilde{\mathcal{T}}'' = \tilde{\mathcal{T}} - \tilde{\mathcal{T}}'$ . Der Operator  $\tilde{\mathcal{T}}'$  führt Berechnungen aus, die i. d. R. nie vernachlässigt werden dürfen und daher immer durchgeführt werden müssen. Es ist zu erwarten, dass  $\tilde{\mathcal{T}}''$  demgegenüber einen sehr viel geringeren Anteil beisteuert. Daher dürfte sich die Anwendung des räumlich adaptiven Verfahrens auf  $\tilde{\mathcal{T}}''$  mit einer hohen Effizienz bewerkstelligen lassen.

Bei der Kombination des Detailgitter-Verfahrens und der adaptiven Auflösung stellt sich nicht die Frage, ob dieses umsetzbar ist, sondern mit welchen Methoden die Kombination erfolgen muss, um ein numerisch effizientes Gesamtverfahren zu erlangen. Erfahrungsgemäß sind derartige Problemstellungen zumindest nicht trivial zu lösen. So kann sicher erst nach vertiefenden Untersuchungen eine Aussage zur Tauglichkeit dieses Ansatzes für praktische Probleme getroffen werden.

Ein weiterer Aspekt zum Thema der effizienten Darstellung von Operatoren wird mit Punkt 4 angesprochen. Abschnitt 8.2.3 untersuchte das Problem, warum bei der Transformation der Simulationsoperatoren unerwartet viele unterschiedliche  $O^2$ -Listen auftraten. In den Ausführungen zu Punkt 6 wurden einige Vorschläge zur Lösung des Problems kurz wiederholt. Eine weitere Verbesserung ließe sich vermutlich dadurch erreichen, dass man die Abbildungen zwischen den Teilräumen des Waveletbereiches nicht mehr mit einem einzelnen Operator bewerkstelligt, sondern stattdessen für jede Kombination aus Urbild-Teilraum und Bild-Teilraum einen separaten Teil-Operator definiert. Die Summe aller Teil-Operatoren entspricht dann wieder dem ursprünglichen Operator.

Die Aufteilung hätte auch im Hinblick auf eine Implementierung des in Abschnitt 7.2.5.2 beschriebenen Verfahrens der zeitlichen Adaptivität Vorteile. Dieses Verfahren ermöglicht in einigen Teilräumen die Anwendung einer größeren Zeitschrittweite, wodurch sich ein weiteres Ersparnis an Rechenzeit erzielen lässt. Mit der Listen-basierten Darstellung eines Operators lässt sich dies nicht umsetzen, sofern ein Bildpunkt mit lediglich einer Summe berechnet wird, in die sämtliche Teilräume des Urbildes eingehen. Die vorgeschlagene Aufteilung in entsprechende Teil-Operatoren schafft hier Abhilfe.

Während sich Punkt 4 auf die Verbesserung der Darstellung Wavelet-transformierter Operatoren bezieht, zielt Punkt 5 auf die Behandlung von Operatoren oder Teil-Operatoren ab, die nicht transformiert werden. In diesen Fällen kann eine andere Implementierung als die über Listen deutlich effizienter sein. Es ist häufig so, dass sich zu einem festen gegebenen Algorithmus (z. B. der WT) programmiertechnische Optimierungen finden lassen, die eine besonders schnelle Abarbeitung erlauben.

Beispielsweise muss ein Operator  $\mathcal{T}$  nicht notwendigerweise dadurch in den Waveletbereich transformiert werden, dass WT- und IWT-Operatoren in Form von Listen aufgebaut werden, um dann mit  $\mathcal{T}$  verkettet zu werden. Stattdessen ist es auch möglich, die Transformation in einem eigens erstellten Programmteil durchzuführen. Auf diese Weise liegen lediglich  $\mathcal{T}$  und  $\tilde{\mathcal{T}}$  in Listen-Darstellung vor.

Die WT von Operatoren dient hier lediglich als einfaches Beispiel. Da Operatoren nur zu Beginn einer Simulation, d. h. lediglich einmal, transformiert werden, hat eine Optimierung an dieser Stelle sicher nur einen sehr begrenzten Nutzen. Anders sieht es dagegen mit Transformationen aus, die immer oder zumindest über mehrere Zeitschritte hinweg durchzuführen sind. Das offensichtlichste Beispiel hierfür ist die Anregung und Ausgabe von Feldverteilungen, insbesondere wenn große Raumbereiche betroffen sind, sodass sich die Methoden nach Abschnitt 7.3.4 nicht mehr anwenden lassen. Optimierte Programmteile zur WT oder IWT der Felder können dann durchaus die Rechenzeit reduzieren.

Auch für Erweiterungen der FDTD wie PMLs können Implementierungen angegeben werden, die eine kürzere Rechenzeit erlauben, als es mit der schematischen Listen-Darstellung möglich wäre. Einer der wesentlichen Gründe hierfür ist in der starken Inhomogenität der Materialverteilung

zu finden, die in der Listen-Darstellung deutlich mehr Speicherzugriffe und damit mehr Rechenzeit erfordert, als es in homogenen Bereichen der Fall ist. Da PMLs oft nicht unerhebliche Anteile des gesamten zu simulierenden Raumes einnehmen, kann eine optimierte Implementierung durchaus von nicht zu vernachlässigendem Vorteil sein. Die in Abschnitt 7.3.5 entwickelte Teilbereichs-WT verliert dadurch keineswegs an Relevanz, da sich manuelle Implementierungen von Algorithmen nicht in jedem Fall lohnen.

Die Verbindung Listen-basierter Operatoren und manuell implementierter Operatoren bzw. Algorithmen kann nach den o. g. Beispielen Vorteile bringen. Der höheren Flexibilität steht allerdings auch ein sicherlich erhöhter Programmieraufwand gegenüber – sowohl was die Programmierung der Operatoren angeht, als auch in Bezug auf die Verbindung selber. Die während dieser Forschungsarbeit gewonnenen Erfahrungen lassen es jedoch als sehr wahrscheinlich erachten, dass vor allem die Einbeziehung manuell implementierter Operatoren überwiegend vorteilhaft ist.

# Kapitel 10

## Zusammenfassung

In der vorliegenden Arbeit wird am Beispiel der FDTD demonstriert, wie bestehende Simulationsverfahren mit Hilfe von Wavelets (insbesondere Haar- und Daubechies-Funktionen) komprimiert und beschleunigt werden können. Diese Möglichkeit besteht für Simulationsverfahren mit linearen Eigenschaften. Es lassen sich jedoch auch einzelne Nichtlinearitäten einbeziehen.

Es werden zwei verschiedene Methoden vorgestellt, mit denen Simulationsverfahren in den Waveletbereich transformiert werden können. Bei der Matrix-Methode werden alle Operatoren des Simulationsverfahrens sowie die Transformationsoperatoren der WT mit (Unter-)Matrizen formuliert. Die Matrizen sind dann in entsprechender Weise miteinander zu multiplizieren und zu addieren. Diese Operationen lassen sich mit wenig Aufwand in einem Computerprogramm umsetzen. Die Untersuchung zeigt jedoch, dass bereits die Darstellung der Operatoren in Matrixform für realistische Probleme äußerst aufwendig werden kann, da die Matrizen bestimmte Bedingungen erfüllen müssen.

In einigen Versuchen erwies sich dieser Ansatz dementsprechend als langsamer im Vergleich zu einer reinen FDTD-Simulation. Vergleicht man allerdings die Matrix-basierte Methode nur mit sich selbst, indem man verschiedene Transformationstiefen für die WT auswählt und die Ordnung der Daubechies-Funktionen variiert, so lassen sich Reduktionen in den Simulationszeiten ausmachen. Diese Beobachtung war die Grundlage für die Entwicklung der zweiten Methode, bei der Operatoren mit Listen

aus Faktoren und Relativvektoren dargestellt werden.

Mit Hilfe der Listen lassen sich Operatoren i. d. R. sehr viel einfacher und effizienter darstellen als mit Matrizen. Auf diese Weise konnten mit der Wavelet-transformierten FDTD in einigen Fällen kürzere Simulationszeiten erzielt werden als mit der untransformierten FDTD. Die beobachteten Beschleunigungen blieben jedoch hinter den Erwartungen zurück. Allerdings konnte in der nachfolgenden Untersuchung aufgezeigt werden, wo die Ursachen für die unzureichende Abnahme der Rechenzeiten liegen.

Die Gründe lassen sich im Wesentlichen darauf zurückführen, dass die genaue Struktur von FDTD-Operatoren im Waveletbereich in der Literatur bislang nicht behandelt wurde. Erst mit der durchgeführten Untersuchung wurden einige essenzielle Eigenschaften der Urbildvektoren aufgedeckt. Als Ergebnis konnten Verbesserungsvorschläge zu der Listen-Darstellung aufgestellt werden, die eine sehr deutliche Erhöhung der Effizienz und eine entsprechende Abnahme bei den Simulationszeiten erwarten lassen.

Mit der Listen-basierten WT der FDTD ist es möglich, den numerischen Aufwand bei örtlich begrenzten Anregungen und Ausgaben von Feldwerten drastisch zu reduzieren. Es können sogar mit geringem Aufwand effiziente Auswertungsoperatoren erzeugt werden, die z. B. das Potenzial eines Leiters direkt im Waveletbereich bestimmen. Weiterhin erlaubt das Detailgitter-Verfahren eine sehr schematische Vorgehensweise bei der Kompression der FDTD. Das resultierende Verfahren kann wegen der unterschiedlichen Auflösungsstufen des Waveletbereichs als eine Art Untergitter-Verfahren interpretiert werden.

Im Vergleich zu vielen anderen Untergitter-Verfahren können für das Detailgitter-Verfahren Stabilitätsaussagen getroffen werden. Dabei zeigt sich, dass die Stabilität der untransformierten FDTD nach der WT erhalten bleibt, obwohl Entwicklungskoeffizienten vernachlässigt werden. Eine derartige Stabilitätsaussage ist in der Literatur zur MRTD, die ja ebenfalls oft in mehreren Auflösungsbereichen arbeitet, nicht zu finden. Dort werden Stabilitätskriterien lediglich für den Fall hergeleitet, dass kein Entwicklungskoeffizient vernachlässigt wird.

Für die Matrix-basierte WT der FDTD sowie für die MRTD werden verschiedene räumlich adaptive Verfahren eingesetzt, mit denen die Vernachlässigung von Entwicklungskoeffizienten und damit die räumliche Auflösung gesteuert wird. Das Detailgitter-Verfahren ist demgegenüber

---

von statischer Natur. Während letzteres von der Kompression eines Feldes besseren Gebrauch macht, ist das adaptive Verfahren flexibler und meist unkritischer in der Anwendung. Die adaptive Kompression lässt sich natürlich auch in der Listen-basierten WT der FDTD anwenden. Von größerem Interesse dürfte jedoch die in Abschnitt 9.2.2 angesprochene Möglichkeit sein, das statische und das adaptive Verfahren miteinander zu kombinieren.

Die Vergleiche mit der MRTD sind sehr hilfreich für die Bewertung und Weiterentwicklung der Listen-basierten WT der FDTD. Beispielsweise finden sich hierdurch überwiegend Bestätigungen für die Entscheidung zu Haar- und Daubechies-Funktionen anstatt Battle-Lemarié-Funktionen, da letztere einen recht hohen numerischen Aufwand erfordern und die künstliche Begrenzung ihres Trägers Ungenauigkeiten hervorruft. Als sehr interessante Alternative zu diesen Funktionen erweisen sich die CDF-Funktionen, die in jüngerer Zeit Eingang in die MRTD gefunden haben.

Der Umgang mit Materialinhomogenitäten, Variationen in der Gitterweite und elektrischen Wänden stellt sich durch den Listen-Ansatz als problemlos dar. Während in der MRTD oft Matrixgleichungen zu lösen sind, Integrale numerisch ausgewertet werden müssen, ungenauere Approximationen als die eines Galerkin-Verfahrens zum Einsatz kommen und Spiegelungsprinzipien angewendet werden, sind in der Listen-basierten WT der FDTD lediglich alle benötigten Operatoren so zu definieren, wie sie in der FDTD angewendet werden. Die Verkettung mit den Wavelet-Operatoren führt dann ganz automatisch zu einem Simulationsverfahren im Waveletbereich.

Diese Vorgehensweise stellt wohl den größten Vorteil dar, den die Listen zu bieten haben: Ein „beliebiges“ Simulationsverfahren ist lediglich im Grundraum zu definieren. Seine Darstellung im Waveletbereich muss nicht hergeleitet werden, indem auf jeden benötigten Operator in allen Auflösungsstufen für das jeweils interessierende Wavelet ein Galerkin-Verfahren angewendet wird. Stattdessen sind die Operatoren des Grundraumes lediglich mit den Wavelet-Operatoren zu verketteten.

Der Vorteil dieser einfachen Herangehensweise gewinnt weiter an Bedeutung, wenn man die zahlreichen Erweiterungen ins Auge fasst, die sich in der FDTD etabliert haben. Hier sei beispielhaft die Möglichkeit zur genaueren Diskretisierung schräg liegender oder krummliniger Wände mit

lokal nicht-orthogonalen Zellen genannt. Manche dieser Erweiterungen machen ggf. eine Modifikation der WT- und IWT-Operatoren erforderlich (z. B. um mit Stützstellen auf nicht-regulären Gitterplätzen besser umgehen zu können), doch es entfällt die Notwendigkeit zur manuellen Entwicklung dieser Erweiterungen in den Waveletbereich.

Die mit der Listen-Darstellung gestiegene Flexibilität ist keineswegs auf die Simulationsoperatoren beschränkt. Auch in Bezug auf Wavelet-Funktionen und -Transformationen ergeben sich größere Freiheiten. Durch einen Austausch der Skalierungs- und Waveletkoeffizienten lassen sich beliebige Funktionen aus dem Bereich der MSA auf ihre Kompressionseigenschaften untersuchen. Sogar Wavelet packets und Multiwavelets können eingesetzt werden. Am bedeutendsten dürfte jedoch die Möglichkeit zur Anwendung des Lifting schemes sein, welches die Konstruktion problem-spezifischer Wavelets erlaubt. Weiterhin wäre das Beylkin-Verfahren zu nennen, das insbesondere für elektrisch große Strukturen eine kompaktere Darstellung von Operatoren im Waveletbereich erlaubt.

Eine rigorose Untersuchung aller Möglichkeiten und Kombinationen lässt sich mit der konventionellen Vorgehensweise, d. h. der konsequenten Anwendung des Galerkin-Verfahrens, kaum bewerkstelligen. Die Methode der Darstellung mit Listen wurde mit dem Ziel entwickelt, ein Werkzeug zur Durchführung solcher Untersuchungen zu erschaffen. Beispielsweise konnte die FDTD auf diese Weise erstmalig mit separablen Tensorprodukt-Basen in den Waveletbereich transformiert werden.

Wenngleich die Effizienz des Verfahrens noch zu wünschen übrig lässt, konnten dennoch sehr vielversprechende Wege zur Verbesserung der Listen-Darstellung aufgezeigt werden. Die Vielzahl der Ausbaumöglichkeiten in konzeptioneller und mathematischer Hinsicht im Detail zu untersuchen, würde den Umfang der vorliegenden Arbeit jedoch bei weitem sprengen. So stellt die vorliegende Arbeit mit ihren grundlegenden Untersuchungen zur Matrix- und Listen-basierten WT der FDTD vor allem eine Basis für weiterführende Untersuchungen dar. Die Vielseitigkeit der Wavelettheorie mit den Simulationstechniken der FDTD zu kombinieren, dürfte damit auch in der näheren Zukunft von hohem Interesse sein und angesichts der bislang ungenutzten Möglichkeiten zu bemerkenswerten Resultaten führen.

# Anhang A

## Ergänzende mathematische Sätze und Beweise

### A.1 Spiegelung von Wavelets und Koeffizienten

Vergleicht man den Funktionsgraphen des Daubechies-Wavelets zweiter Ordnung aus [12] (Abbildung 2.9) mit dem aus [13] (Abbildung 6.3), so fällt auf, dass sie innerhalb ihres Trägers spiegelverkehrt zueinander sind. Auch die Skalierungskoeffizienten werden in umgekehrter Reihenfolge aufgeführt (Tabelle 2.3 bzw. 6.1). In der durchgesehenen Literatur fand sich kein Hinweis auf die Relevanz der Reihenfolge der Koeffizienten, sodass der folgende Satz hergeleitet wurde:

**Satz 8.** *Es sei  $\{h_k\}_{k \in \mathbb{Z}}$  die Folge von Skalierungskoeffizienten einer MSA und  $\varphi$  die zugehörige Skalierungsfunktion. Für ein beliebiges  $M \in \mathbb{Z}$  ist dann  $\{\tilde{h}_k\}_{k \in \mathbb{Z}}$  mit  $\tilde{h}_k = h_{M-k}$  die Folge von Skalierungskoeffizienten, die zu der Skalierungsfunktion  $\tilde{\varphi}$  mit  $\tilde{\varphi}(\cdot) = \varphi(M - \cdot)$  gehört.*

*Beweis.* Es ist zu zeigen, dass  $\{\tilde{h}_k\}_{k \in \mathbb{Z}}$  und  $\tilde{\varphi}$  ebenfalls eine Skalierungsgleichung erfüllen. Beginnen wir dazu mit der Skalierungsgleichung für

$\{h_k\}_{k \in \mathbb{Z}}$  und  $\varphi$ :

$$\varphi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \varphi(2x - k) \quad \left| \quad h_k = \tilde{h}_{M-k} \quad (\text{A.1})\right.$$

$$= \sqrt{2} \sum_{k \in \mathbb{Z}} \tilde{h}_{M-k} \varphi(2x - k) \quad \left| \quad k \rightarrow M - k \quad (\text{A.2})\right.$$

$$= \sqrt{2} \sum_{k \in \mathbb{Z}} \tilde{h}_k \varphi(2x - M + k) . \quad (\text{A.3})$$

Jetzt ersetzen wir  $\varphi(\cdot) = \tilde{\varphi}(M - \cdot)$  :

$$\tilde{\varphi}(M - x) = \sqrt{2} \sum_{k \in \mathbb{Z}} \tilde{h}_k \tilde{\varphi}(M - (2x - M + k)) \quad (\text{A.4})$$

$$= \sqrt{2} \sum_{k \in \mathbb{Z}} \tilde{h}_k \tilde{\varphi}(2(M - x) - k) \quad \left| \quad y = M - x \quad (\text{A.5})\right.$$

$$\tilde{\varphi}(y) = \sqrt{2} \sum_{k \in \mathbb{Z}} \tilde{h}_k \tilde{\varphi}(2y - k) . \quad (\text{A.6})$$

Gleichung (A.6) ist die gesuchte Skalierungsgleichung für die „gespiegelten“ Größen  $\tilde{h}_k$  und  $\tilde{\varphi}$ .  $\square$

Es ist nahe liegend, dass auch die Wavelets ihre Form nicht ändern, sondern nur eine Spiegelung stattfindet.

**Satz 9.** *Es gelten die Angaben von Satz 8. Dann gilt für die zu  $\{\tilde{h}_k\}_{k \in \mathbb{Z}}$  gehörenden Waveletkoeffizienten  $\tilde{g}_k = (-1)^M g_{2-M-k}$  und für das Wavelet  $\tilde{\psi}(\cdot) = (-1)^M \psi(1 - \cdot)$ .*

*Beweis.* Zunächst werden mit Gleichung (3.27) die neuen Waveletkoeffizienten hergeleitet.

$$g_k = (-1)^k h_{1-k} \quad \left| \quad h_l = \tilde{h}_{M-l} \quad (\text{A.7})\right.$$

$$= (-1)^k \tilde{h}_{M-1+k} \quad \left| \quad k \rightarrow 2 - M - k \quad (\text{A.8})\right.$$

$$g_{2-M-k} = (-1)^{2-M-k} \tilde{h}_{1-k} = (-1)^M (-1)^k \tilde{h}_{1-k} \quad (\text{A.9})$$

$$= (-1)^M \tilde{g}_k . \quad (\text{A.10})$$

Dieses Ergebnis kann direkt in Gleichung (3.26) eingesetzt werden:

$$\psi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} (-1)^M \tilde{g}_{2-M-k} \varphi(2x - k) \quad \Bigg| \quad k \rightarrow 2 - M - k \quad (\text{A.11})$$

$$(-1)^M \psi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} \tilde{g}_k \varphi(2x - 2 + M + k) \quad \Bigg| \quad \varphi(\cdot) = \tilde{\varphi}(M - \cdot) \quad (\text{A.12})$$

$$= \sqrt{2} \sum_{k \in \mathbb{Z}} \tilde{g}_k \tilde{\varphi}(M - (2x - 2 + M + k)) \quad (\text{A.13})$$

$$= \sqrt{2} \sum_{k \in \mathbb{Z}} \tilde{g}_k \tilde{\varphi}(2(1 - x) - k) \quad \Bigg| \quad y = 1 - x \quad (\text{A.14})$$

$$(-1)^M \psi(1 - y) = \sqrt{2} \sum_{k \in \mathbb{Z}} \tilde{g}_k \tilde{\varphi}(2y - k) = \tilde{\psi}(y) . \quad (\text{A.15})$$

□

Wird bei Daubechies-Wavelets der Ordnung  $N$  für  $M$  die Trägerbreite  $2N - 1$  gewählt, ergibt sich eine Spiegelung der Skalierungsfunktion und des Wavelets innerhalb des Trägers. Durch Einsetzen der Trägergrenzen aus Gleichung (3.39) kann dies leicht nachgeprüft werden. Für Skalierungs- und Waveletkoeffizienten gilt diese Aussage in ganz analoger Weise.

**Fazit:** Die Wahl einer auf- oder absteigenden Reihenfolge von Koeffizienten hat – von Spiegelungen abgesehen – keinen Einfluss auf die Form von Skalierungsfunktion und Wavelet.

## A.2 Indexverschiebung von Koeffizienten

Der folgende Satz gewährt eine kleine Freiheit bei der Anpassung der WT an Problemstellungen aus der Praxis:

**Satz 10.** *Die Verschiebung der Skalierungskoeffizienten  $\{h_k\}_{k \in \mathbb{Z}}$  um  $l$  Positionen ( $l \in \mathbb{N}$ ) nach links (rechts) führt auf ein um  $l$  nach links (rechts) verschobenes Translat der Skalierungsfunktion  $\varphi$ . Analoges gilt bei Wavelets für  $\{g_k\}_{k \in \mathbb{Z}}$  und  $\psi$ .*

*Beweis.* Der Beweis erfolgt ähnlich wie der in Abschnitt A.1. Gegeben ist  $\{h_k\}_{k \in \mathbb{Z}}$  mit  $\varphi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \varphi(2x - k)$ . Nun wird  $\tilde{h}_k = h_{k+l}$  bzw.  $h_k = \tilde{h}_{k-l}$ ,  $l \in \mathbb{Z}$ , in die Skalierungsgleichung eingesetzt:

$$\varphi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} \tilde{h}_{k-l} \varphi(2x - k) = \sqrt{2} \sum_{k \in \mathbb{Z}} \tilde{h}_k \varphi(2x - k - l) \quad (\text{A.16})$$

$$= \sqrt{2} \sum_{k \in \mathbb{Z}} \tilde{h}_k \varphi(2(x-l) - k + l) \quad \left| \quad y = x - l \quad (\text{A.17})$$

$$\varphi(y+l) = \sqrt{2} \sum_{k \in \mathbb{Z}} \tilde{h}_k \varphi(2y - k + l) \quad \left| \quad \tilde{\varphi}(\cdot) = \varphi(\cdot + l) \quad (\text{A.18})$$

$$\tilde{\varphi}(y) = \sqrt{2} \sum_{k \in \mathbb{Z}} \tilde{h}_k \tilde{\varphi}(2y - k) . \quad (\text{A.19})$$

Wird in dem Beweis  $h$  durch  $g$  und  $\varphi$  durch  $\psi$  ersetzt, so folgt unmittelbar der Beweis für  $\{g_k\}_{k \in \mathbb{Z}}$  und  $\psi$ .  $\square$

# Anhang B

## Software-Implementierungen

### B.1 Sortierverfahren

In Abschnitt 5.3.2.1 wurde das Problem angesprochen, dass  $O^2$ -Listen aus Gründen der Effizienz sortiert werden müssen. Einige der bekannten Sortierverfahren werden in [80] und [81] vorgestellt und ihre Eigenschaften diskutiert. Eines der am häufigsten benutzten Verfahren ist wohl *Quicksort*. In vielen Fällen liefert Quicksort eine sortierte Liste in deutlich kürzerer Zeit als alle anderen bekannten Sortierverfahren. Im ungünstigsten Fall jedoch ist Quicksort bei  $L$  Elementen von der Ordnung  $O(L^2)$ .

Als Alternative zu Quicksort wurde für die vorliegende Arbeit schließlich *Mergesort* gewählt. Dieses Verfahren ist *worst case optimal*, d. h. es hat auch unter den ungünstigsten Umständen lediglich eine Ordnung von  $O(L \log L)$ .

Zentraler Bestandteil von Mergesort ist das Zusammenfassen zweier sortierter Listen zu einer sortierten Gesamtliste. In der Standard-Implementierung wird damit begonnen, die  $L$  Elemente als  $L$  Listen mit je einem Element zu betrachten. Danach wird mit der beschriebenen Zusammenfassung von je zwei Listen zu einer Liste begonnen. Aus  $L$  Listen werden dadurch  $\frac{L}{2}$  Listen. In der nächsten Stufe bzw. Iteration werden dann wiederum Listen zusammengefasst, sodass nur noch  $\frac{L}{4}$  Listen übrig bleiben. Das Ganze wird sooft wiederholt, bis schließlich nur noch eine vollständig sortierte Liste vorhanden ist.

Abweichend von der Standard-Implementierung wurde ein Verfahren zur Ausnutzung normaler und inverser Vorsortierungen entwickelt. Dazu werden im ersten Schritt benachbarte Elemente miteinander verglichen, um möglichst große Listen mit steigender oder fallender Sortierreihenfolge zu detektieren. Dadurch reduziert sich die Anzahl der durchzuführenden Iterationen grundsätzlich um wenigstens Eins. Im günstigsten Fall ist keine Iteration mehr notwendig. Abbildung B.1 veranschaulicht das Prinzip.

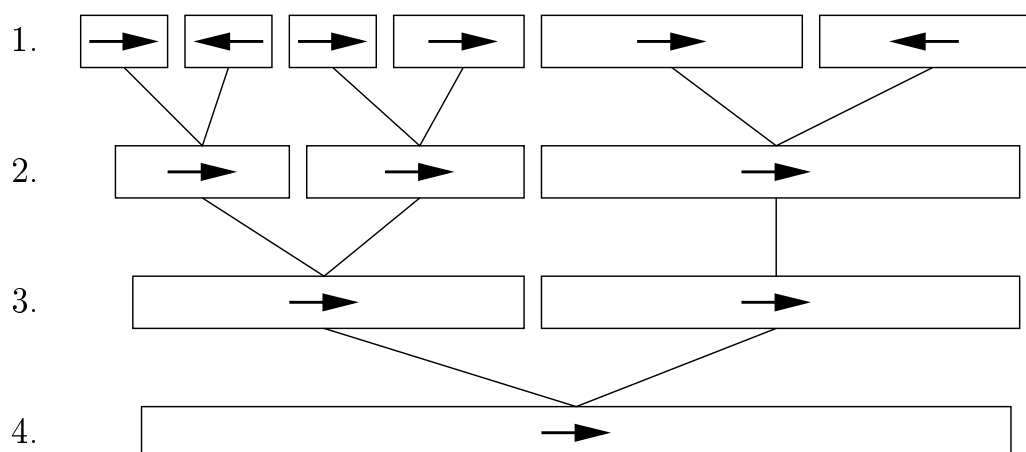


Abbildung B.1: Prinzipbild für ein abgewandeltes Mergesort-Verfahren. Im ersten Schritt wird die zu sortierende Liste auf Teillisten mit normaler ( $\rightarrow$ ) und inverser ( $\leftarrow$ ) Sortierreihenfolge untersucht. Anschließend folgt das schrittweise Zusammenfassen von Teillisten nach dem üblichen Mergesort-Verfahren, wobei jedoch bei der ersten Iteration (Übergang von Schritt 1 auf Schritt 2) inverse Sortierreihenfolgen berücksichtigt werden.

## B.2 Faktorvergleich mit Toleranzschwelle

Im Gegensatz zu den wohl meisten anderen numerischen Simulationsverfahren wird der Vergleich von zwei Faktorwerten beim Verfahren nach Abschnitt 5.3 sehr häufig durchgeführt. Daher soll hier auf bestimmte Probleme eingegangen werden, die mit einem solchen Vergleich verbunden sind.

Aus mathematischer Sicht ist es relativ leicht zwei Werte  $a, b \in \mathbb{R}$  auf

ungefähre Gleichheit zu prüfen. Die folgende Lösung erscheint plausibel:

$$a \approx b \quad \text{wenn} \quad \left| \frac{a}{b} - 1 \right| \leq \varepsilon . \quad (\text{B.1})$$

$\varepsilon$  ist hier eine vorgebbare Toleranzschwelle. Gleichung (B.1) ist aber nur dann anwendbar, wenn  $b \neq 0$  ist. Man könnte also definieren

$$a \approx b \quad \text{wenn} \quad \begin{cases} |a| \leq \delta & , \quad b = 0 \\ \left| \frac{a}{b} - 1 \right| \leq \varepsilon & , \quad b \neq 0 \end{cases} , \quad (\text{B.2})$$

wobei  $\delta$  eine Toleranzschwelle sein soll, welche die ungefähre Gleichheit zu Null angibt. Nach Gleichung (B.2) ist also zunächst die Gleichheit von  $b$  zu Null zu testen. Dabei stellt sich die Frage, ob  $b$  auch ungefähr Null sein darf, d. h.  $|b| < \delta$ . Doch selbst wenn  $b > \delta$ , aber dennoch sehr klein ist, kann die Berechnung von  $\frac{a}{b}$  auf einem Prozessor bzw. einer FPU (*Floating Point Unit*) Probleme bereiten. Ist nämlich der Betrag von  $a$  genügend groß, kann die Division zu einem Überlauf führen und eine Ausnahmebehandlung auslösen (*Overflow exception*). Sind die Größenverhältnisse genau umgekehrt, kann es zu einem Unterlauf kommen.

Derartige Situationen lassen sich teilweise durch Renormierung von Operatoren vermeiden. Das numerisch bedingte Auftreten von Faktorwerten sehr nahe bei Null wird jedoch durch Verkettungen wie  $\mathcal{P}\mathcal{T}\mathcal{P}^*$  begünstigt und kann oft nicht verhindert werden.<sup>1</sup>

Das folgende Verfahren wurde zur Umgehung der genannten Schwierigkeiten entwickelt. Dabei sind die fünf Punkte priorisiert, d. h. es ist der erste Punkt zu wählen, bei dem die Voraussetzung erfüllt ist. Zusätzlich zu den Toleranzschwellen  $\varepsilon$  und  $\delta$  tritt eine weitere Größe  $A$  auf, die größer als Eins und nicht zu nahe bei Eins sein soll:

$$A > 1 \quad \wedge \quad A - 1 \not\ll 1 . \quad (\text{B.3})$$

Mit dieser Größe kann über eine Multiplikation statt über eine Division festgestellt werden, ob  $a$  und  $b$  deutlich verschieden voneinander sind

---

<sup>1</sup>Die Hauptursache liegt darin begründet, dass nicht nur  $\mathcal{P}$ , sondern auch der inverse Operator  $\mathcal{P}^*$  unmittelbar angewendet werden. Je nach den Abbildungseigenschaften eines Operators  $\mathcal{T}$  führt die Verkettung bei analytischer Berechnung zu exakten Kompensationen in einigen Bildbereichen. Die numerische Berechnung allerdings kann zu ungenauen Kompensationen führen und so Faktorwerte nahe bei Null liefern.

(Punkte 3 und 4). Damit soll ein möglicher Überlauf oder Unterlauf bei der Division  $\frac{a}{b}$  verhindert werden.

1. Voraussetzung:  $a = 0$ .

Dann ist

$$a \approx b \quad \text{wenn} \quad |b| \leq \delta, \quad (\text{B.4})$$

$$a \not\approx b \quad \text{wenn} \quad |b| > \delta. \quad (\text{B.5})$$

2. Voraussetzung:  $b = 0$ .

Dann ist

$$a \approx b \quad \text{wenn} \quad |a| \leq \delta, \quad (\text{B.6})$$

$$a \not\approx b \quad \text{wenn} \quad |a| > \delta. \quad (\text{B.7})$$

3. Voraussetzung:  $|a| > A \cdot |b|$ .

Dann ist  $a \not\approx b$ .

4. Voraussetzung:  $|b| > A \cdot |a|$ .

Dann ist  $a \not\approx b$ .

5. Keine Voraussetzung.

Es ist

$$a \approx b \quad \text{wenn} \quad \left| \frac{a}{b} - 1 \right| \leq \varepsilon, \quad (\text{B.8})$$

$$a \not\approx b \quad \text{sonst.} \quad (\text{B.9})$$

# Literaturverzeichnis

- [1] Industrial applications of electromagnetic (EM) solvers. Workshop zur Konferenz *IEEE MTT-S International Microwave Symposium 2001*, Phoenix, Arizona. Session WFA
- [2] YEE, K. S.: Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. In: *IEEE Antennas and Propagation* **14** (1966), Nr. 5, S. 302–307
- [3] WERTHEN, M.: *Analyse von dreidimensionalen Komponenten der Mikro- und Millimeterwellentechnik mit Hilfe numerischer Zeitbereichsverfahren*. Hochfrequenztechnik, Aachen: Shaker, 1998
- [4] TAFLOVE, A.: *Computational Electromagnetics: The finite-difference time-domain method*. Boston: Artech House, 1995
- [5] MITTRA, R.; PEKEL, Ü.: A new look at the perfectly matched layer (PML) for the reflectionless absorption of electromagnetic waves. In: *IEEE Microwave and Guided Wave Letters* **5** (1995), Nr. 3, S. 84–86
- [6] NAISHADHAM, K.: Stability considerations in the application of PML absorbing boundary condition to FDTD simulation of microwave circuits. In: *Microwave Symposium Digest*, S. 581–584, IEEE MTT-S International, 1996
- [7] BÉRENGER, J.-P.: Evanescent waves in PML's: Origin of the numerical reflection in wave-structure interaction problems. In: *IEEE Antennas and Propagation* **47** (1999), Nr. 10, S. 1497–1503
- [8] GEDNEY, S. D.: An anisotropic perfectly matched layer-absorbing medium for the truncation of FDTD lattices. In: *IEEE Antennas and Propagation* **44** (1996), Nr. 12, S. 1630–1639

- 
- [9] ZHAO, L.; CANGELLARIS, A. C.: GT-PML: Generalized theory of perfectly matched layers and its application to the reflectionless truncation of finite-difference time-domain grids. In: *IEEE Transactions on Microwave Theory and Techniques* **44** (1996), Nr. 12, S. 2555–2563
- [10] YU, W.; WERNER, D. H.; MITTRA, R.: An artificially-synthesized absorbing medium for the truncation of FDTD lattices. In: *IEEE Microwave and Guided Wave Letters* **10** (2000), Nr. 4, S. 128–130
- [11] BRONSTEIN, I. N.; SEMENDJAJEW, K. A.: *Taschenbuch der Mathematik*, Bd. 1. Thun: Harri Deutsch, 24. Aufl., 1989
- [12] LOUIS, A. K.; MAASS, P.; RIEDER, A.: *Wavelets*. Stuttgart: Teubner, 1994
- [13] DAUBECHIES, I.: *Ten lectures on wavelets*. Philadelphia, Pennsylvania: SIAM, 1992
- [14] TASWELL, C.; MCGILL, K. C.: Wavelet transform algorithms for finite-duration discrete-time signals. In: *ACM Transactions on Mathematical Software* **20** (1994), Nr. 3, S. 398–412
- [15] RESTREPO, J. M.; LEAF, G. K.: Inner product computations using periodized Daubechies wavelets. In: *Journal for Numerical Methods in Engineering* **40** (1997), S. 3557–3578
- [16] GETZ, N. H.: A fast discrete periodic wavelet transform. Memorandum UCB/ERL M92-138, Electronics Research Laboratory Berkeley, University of California, 1992
- [17] JAMESON, L.: On the Daubechies-based wavelet differentiation matrix. Tech. Rep. 93-95, Inst. for Comp. Appl. in Sci. and Eng., NASA Langley Res. Cent., Hampton, Virginia, 1993
- [18] WALTER, M.; WOLFF, I.: An algorithm for realizing Yee's FDTD-method in the wavelet domain. In: *Microwave Symposium Digest*, IEEE MTT-S International, Anaheim, California, Juni 1999. Session WEF1-15

- 
- [19] WERTHEN, M.; WOLFF, I.: A novel wavelet based time domain simulation approach. In: *IEEE Microwave and Guided Wave Letters* **6** (1996), Nr. 12, S. 438–440
- [20] STILLER, A.: DRAMatische Modularitäten. In: *c't* 1995, Nr. 4, S. 334–342
- [21] WINDECK, C.: Rasender Daten-Bus. In: *c't* 2000, Nr. 3, S. 232–241
- [22] ÜBERHUBER, C.: *Computer-Numerik*, Bd. 1. Berlin: Springer, 1995
- [23] SWELDENS, W.; PIESSENS, R.: Wavelet sampling techniques. In: *1993 Proceedings of the Statistical Computing Section*, S. 20–29, American Statistical Association, Aug. 1993
- [24] REICHELT, A.: Absorbierende Randbedingungen für eine Wavelet-basierte Momentenmethode im Zeitbereich. Gerhard-Mercator-Universität – Gesamthochschule Duisburg, Diplomarbeit, Dez. 1998. FB9/ATE
- [25] OTTO, S.: The application of higher-order Daubechies-wavelets in methods of moments. Gerhard-Mercator-Universität Duisburg, Studienarbeit, Febr. 2002. FB9/ATE
- [26] RENNINGS, A.: The application of Daubechies-wavelets in time domain field analysis techniques. Gerhard-Mercator-Universität – Gesamthochschule Duisburg, Diplomarbeit, Febr. 2000. FB9/ATE
- [27] RENNINGS, A.; TONI, V.; WALDOW, P.; QIAN, Y.; ITOH, T.; WOLFF, I.: A novel adaptivity for EM time domain methods: Scale Adaptive Time Steps (SATS). In: *Microwave Symposium Digest*, Bd. 2, S. 757–760, IEEE MTT-S International, Phoenix, Arizona, Mai 2001. Session WE1F-5
- [28] SIMONYI, K.: *Theoretische Elektrotechnik*. Berlin: VEB Deutsche Verlag der Wissenschaften, 9. Aufl., 1989
- [29] BEYLKIN, G.: On the fast algorithm for multiplication of functions in the wavelet bases. In: *Int. Conf. Wavelets and Applications*, (Toulouse), Juni 1992

- [30] WALTER, M.; WALDOW, P.; WOLFF, I.: A wavelet-based FDTD-multigrid-method. In: *Microwave Symposium Digest*, Bd. 3, S. 2023–2026, IEEE MTT-S International, Phoenix, Arizona, Mai 2001. Session THIF-7
- [31] BURG, K.; HAF, H.; WILLE, F.: *Höhere Mathematik für Ingenieure*, Bd. 5. Stuttgart: Teubner, 2. Aufl., 1993
- [32] BRONSTEIN, I. N.; SEMENDJAJEW, K. A.: *Taschenbuch der Mathematik*, Bd. 2. Thun: Harri Deutsch, 6. Aufl., 1991. Ergänzende Kapitel
- [33] GROSCHE, G.; ZEIDLER, E.; ZIEGLER, D.; ZIEGLER, V. (HRSG.): *Teubner-Taschenbuch der Mathematik*, Bd. 2. Stuttgart: Teubner, 7. Aufl., 1995
- [34] ZURMÜHL, R.; FALK, S.: *Matrizen und ihre Anwendungen*, Bd. 1. Aachen: Springer, 6. Aufl., 1992
- [35] BURG, K.; HAF, H.; WILLE, F.: *Höhere Mathematik für Ingenieure*, Bd. 2. Stuttgart: Teubner, 3. Aufl., 1992
- [36] TAO, J.-W.; BAUDRAND, H.: Multimodal variational analysis of uniaxial waveguide discontinuities. In: *IEEE Transactions on Microwave Theory and Techniques* **39** (1991), Nr. 3, S. 506–516
- [37] POLYCARPOU, A. C.; BALANIS, C. A.: An optimized anisotropic PML for the analysis of microwave circuits. In: *IEEE Microwave and Guided Wave Letters* **8** (1998), Nr. 1, S. 30–32
- [38] PERTZ, O.; BEYER, A.: Applying wavelets to electromagnetic field simulation: The method of lines. In: *Microwave Symposium Digest*, (Boston, USA), S. 129–132, IEEE MTT-S International, Juni 2000
- [39] PERTZ, O.: *Anwendung von waveletbasierten Verfahren bei der Simulation von Millimeter- und Mikrometerwellenschaltungen mit der Methode der Geraden*. Gerhard-Mercator-Universität Duisburg, Dissertation, Juni 2001
- [40] KRUMPHOLZ, M.; KATEHI, L. P. B.: New prospects for time domain analysis. In: *IEEE Microwave and Guided Wave Letters* **5** (1995), Nr. 11, S. 382–384

- 
- [41] KRUMPHOLZ, M.; KATEHI, L. P. B.: MRTD: New time-domain schemes based on multiresolution analysis. In: *IEEE Transactions on Microwave Theory and Techniques* **44** (1996), Nr. 4, S. 555–571
- [42] TAM, W. Y.: Comments on „New Prospects for Time Domain Analysis“. In: *IEEE Microwave and Guided Wave Letters* **6** (1996), Nr. 11, S. 422
- [43] TENTZERIS, E. M.; ROBERSTON, R. L.; HARVEY, J.; KATEHI, L. P. B.: Stability and dispersion analysis of Battle-Lemarie-based MRTD schemes. In: *IEEE Transactions on Microwave Theory and Techniques* **47** (1999), Nr. 7, S. 1004–1013
- [44] SARRIS, C. D.; KATEHI, L. P. B.: Fundamental gridding related dispersion effects in multiresolution time domain. In: *Microwave Symposium Digest*, Bd. 2, S. 749–752, IEEE MTT-S International, Phoenix, Arizona, Mai 2001. Session WE1F-3
- [45] SHLAGER, K. L.; SCHNEIDER, J. B.: Relative accuracy of several low-dispersion finite-difference time-domain schemes. In: *IEEE AP-S International Symposium and URSI Radio Science Meeting*, Bd. 1, (Orlando, FL), S. 168–171, Juli 1999
- [46] ROBERTSON, R.; TENTZERIS, E.; KRUMPHOLZ, M.; KATEHI, L. P. B.: Modelling of dielectric cavity structures using multiresolution time-domain analysis. In: *Int. J. Numer. Model.* **11** (1998), Nr. 1, S. 55–68
- [47] ROBERTSON, R.; TENTZERIS, E.; KATEHI, L. P. B.: MRTD applied to complex geometry air-dielectric interfaces in 3D microwave structures. In: *Microwave Symposium Digest*, Bd. 2, (Anaheim, CA, USA), S. 1463–1466, IEEE MTT-S International, Juni 1999
- [48] TENTZERIS, E. M.; CANGELLARIS, A.; KATEHI, L. P. B.; HARVEY, J.: Multiresolution time-domain (MRTD) adaptive schemes using arbitrary resolutions of wavelets. In: *IEEE Transactions on Microwave Theory and Techniques* **50** (2002), Nr. 2, S. 501–516
- [49] NGUYEN, B. T.: On comparisons of time-domain scattering schemes. Vorabdruck, Febr. 1998. Institute for Mathematics and its

- Applications, University of Minnesota, Minneapolis, USA. URL: <http://www.ima.umn.edu/preprints/FEB98/FEB98.html>
- [50] HASTINGS, F. D.; SCHNEIDER, J. B.; BROCHAT, S. L.: A finite-difference time-domain solution to scattering from a rough pressure-release surface. In: *Journal of the Acoustical Society of America* **102** (1997), Nr. 6, S. 3394–3400
- [51] SCHNEIDER, J. B.; KRUHLAK, R. J.: Dispersion of homogeneous and inhomogeneous waves in the Yee finite-difference time-domain grid. In: *IEEE Transactions on Microwave Theory and Techniques* **49** (2001), Nr. 2, S. 280–287
- [52] CHEONG, Y. W.: Wavelet-Galerkin scheme of inhomogeneous electromagnetic problems using the time-dependent Maxwell's equations. Südkorea, Kwangwoon University, MSc thesis, 1999
- [53] CHEONG, Y. W.; LEE, Y. M.; RA, K. H.; KANG, J. G.; SHIN, C. C.: Wavelet-Galerkin scheme of time-dependent inhomogeneous electromagnetic problems. In: *IEEE Microwave and Guided Wave Letters* **9** (1999), Nr. 8, S. 297–299
- [54] FUJII, M.; HOEFER, W. J. R.: Dispersion of time domain wavelet Galerkin method based on Daubechies' compactly supported scaling functions with three and four vanishing moments. In: *IEEE Microwave and Guided Wave Letters* **10** (2000), Nr. 4, S. 125–127
- [55] FUJII, M.; HOEFER, W. J. R.: A Daubechies'-wavelet-based time domain electromagnetic field modeling technique. In: *International Symposium on Antennas and Propagation*, (Fukuoka, Japan), S. 437–440, Aug. 2000
- [56] FUJII, M.; HOEFER, W. J. R.: Time-domain wavelet Galerkin modeling of two-dimensional electrically large dielectric waveguides. In: *IEEE Transactions on Microwave Theory and Techniques* **49** (2001), Nr. 5, S. 886–892
- [57] GOVERDHANAM, K.; TENTZERIS, E.; KRUMPHOLZ, M.; KATEHI, L. P. B.: An FDTD multigrid based on multiresolution analysis. In: *Proceedings of the 1996 IEEE AP-S Conference*, (Baltimore), S. 352–355, Juli 1996

- [58] GOVERDHANAM, K.; KATEHI, L. P. B.; CANGELLARIS, A.: Applications of multiresolution based FDTD multigrid. In: *Microwave Symposium Digest*, S. 333–336, IEEE MTT-S International, Denver, Colorado, Juni 1997. Session TH1E-1
- [59] TENTZERIS, E. M.; HARVEY, J.; KATEHI, L. P. B.: Time adaptive time-domain techniques for the design of microwave circuits. In: *IEEE Microwave and Guided Wave Letters* **7** (1999), Nr. 3, S. 96–98
- [60] TENTZERIS, E. M.: Computational optimization of MRTD Haar-based adaptive schemes used for the design of RF packaging structures. In: *Applied Computational Electromagnetics, Conference proc.*, (Monterey, CA), S. 548–555, März 2000
- [61] MCGARVEY, B.; STACIULESCU, D.; TENTZERIS, E. M.; LASKAR, J.: Adaptive modeling of complex packaging geometries using Haar-based MRTD algorithms. In: *Proceedings of 2000 European Microwave Conference*, Bd. 1, S. 413–416, Paris, Frankreich, Okt. 2000
- [62] BUSHYAGER, N.; OBATOYINBO, A.; TENTZERIS, M.; LASKAR, J.: Multilayer package modeling using the multiresolution time-domain technique. In: *Proceedings of the 2001 IEEE AP-S Symposium*, (Boston, MA), S. 806–809, Juli 2001
- [63] FUJII, M.; HOEFER, W. J. R.: A three-dimensional Haar-wavelet-based multiresolution analysis similar to the FDTD method – Derivation and application. In: *IEEE Transactions on Microwave Theory and Techniques* **46** (1998), Nr. 12, S. 2463–2475
- [64] GRIVET-TALOCIA, S.: On the accuracy of Haar-based multiresolution time-domain schemes. In: *IEEE Microwave and Guided Wave Letters* **10** (2000), Nr. 10, S. 397–399
- [65] SARRIS, C. D.; KATEHI, L. P. B.: Development and application of an efficient FDTD / Haar MRTD numerical interface. In: *Microwave Symposium Digest*, Bd. 2, S. 753–756, IEEE MTT-S International, Phoenix, Arizona, Mai 2001. Session WE1F-4
- [66] DOGARU, T.; CARIN, L.: Efficient time-domain electromagnetic analysis using CDF biorthogonal wavelet expansion. In: *Microwave*

- Symposium Digest*, Bd. 3, S. 2019–2022, IEEE MTT-S International, Phoenix, Arizona, Mai 2001. Session THIF-6
- [67] DOGARU, T.; CARIN, L.: Multiresolution time-domain using CDF biorthogonal wavelets. In: *IEEE Transactions on Microwave Theory and Techniques* **49** (2001), Nr. 5, S. 902–912
- [68] GILES, M. B.: Wavelet compression for unsteady CFD data. Report, Oxford University Computing Laboratory, 1997
- [69] JAWERTH, B.; SWELDENS, W.: An overview of wavelet based multiresolution analyses. In: *SIAM Review* **36** (1994), Nr. 3, S. 377–412
- [70] SWELDENS, W.: Wavelets: What next?. In: *Proc. of the IEEE* **84** (1996), Nr. 4, S. 680–685
- [71] COHEN, A.; DAUBECHIES, I.; VIAL, P.: Wavelets on the interval and fast wavelet transforms. In: *Appl. Comput. Harmon. Anal.* **1** (1993), Nr. 1, S. 54–81
- [72] PLONKA, G.; STRELA, V.: From wavelets to multiwavelets. In: DAEHLEN, M.; LYCHE, T.; SCHUMAKER, L. (HRSG.): *Math. Methods for Curves and Surf. II*, Lillehammer: Vanderbilt University Press, 1998
- [73] FERNANDES, F. C. A.; BURRUS, C. S.: M-band multiwavelet systems. In: *ICASSP proceedings*, 1999
- [74] SWELDENS, W.: Wavelets and the lifting scheme: A 5 minute tour. In: *Z. Angew. Math. Mech.* **76** (1996), S. 41–44
- [75] SWELDENS, W.: The lifting scheme: A custom-design construction of biorthogonal wavelets. In: *Appl. Comput. Harmon. Anal.* **3** (1996), Nr. 2, S. 186–200
- [76] SWELDENS, W.: The lifting scheme: A construction of second generation wavelets. In: *SIAM J. Math. Anal.* **29** (1997), Nr. 2, S. 511–546
- [77] SWELDENS, W.; SCHRÖDER, P.: Building your own wavelets at home. In: *Wavelets in Computer Graphics*, Course Notes, S. 15–87, ACM SIGGRAPH, 1996

- 
- [78] BEYLKIN, G.; COIFMAN, R.; ROKHLIN, V.: Fast wavelet transforms and numerical algorithms I. In: *Comm. Pure Appl. Math.* **44** (1991), S. 141–183
- [79] BEYLKIN, G.: Wavelets and fast numerical algorithms. In: DAUBECHIES, I. (HRSG.): *Different Perspectives on Wavelets*, Bd. 47: *Proceedings of Symposia in Applied Mathematics*, S. 89–117, American Math. Soc., Providence, RI, 1993
- [80] PORTH, D.: Sortierverfahren. In: *c't* 1992, Nr. 3, S. 263–268
- [81] OESTEREICH, B.: Sortieren. In: *c't* 1988, Nr. 11, S. 273–276



# Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter im Fachgebiet Allgemeine und Theoretische Elektrotechnik an der Gerhard-Mercator-Universität Duisburg.

Das Forschungsthema wurde von Herrn Prof. Dr.-Ing. Ingo Wolff angeregt. Für seine Betreuung der Arbeit und die Gelegenheit zur Forschung auf diesem interessanten Gebiet möchte ich mich sehr bedanken.

Herr Prof. Dr.-Ing. Adalbert Beyer förderte bereits während meines Studiums mein Interesse im Bereich der numerischen Feldsimulation, wofür ich ihm meinen herzlichen Dank ausspreche.

Bei Dr.-Ing. Markus Werthen bedanke ich mich für die Überlassung seines Programmes, welches mir erste Simulationen mit der Matrix-basierten WT der FDTD ermöglichte.

Mein ganz besonderer Dank gilt Dr.-Ing. Oliver Pertz sowie Dipl.-Ing. André Rennings für die zahlreichen anregenden Diskussionen und die Korrektur des Manuskripts.

Weiterhin möchte ich mich bei den Mitarbeitern des Fachgebiets Allgemeine und Theoretische Elektrotechnik und bei meinen Kollegen der Siemens AG für die gute Zusammenarbeit und Förderung meiner Arbeit bedanken.

Vor allem bei meinen Eltern und meinen Freunden möchte ich mich für ihre Geduld, Verständnis und Ermunterungen bedanken, die mir eine wesentliche Stütze bei dieser Arbeit waren.