KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT COMPUTERWETENSCHAPPEN
AFDELING NUMERIEKE ANALYSE EN
TOEGEPASTE WISKUNDE
Celestijnenlaan 200A – B-3001 Heverlee

MULTIGRID METHODS FOR TIME-DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS

Promotor:
Prof. Dr. ir. S. Vandewalle

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Jan VAN LENT

7 februari 2006

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT COMPUTERWETENSCHAPPEN
AFDELING NUMERIEKE ANALYSE EN
TOEGEPASTE WISKUNDE
Celestijnenlaan 200A – B-3001 Heverlee

MULTIGRID METHODS FOR TIME-DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS

Jury:
Prof. Dr. ir. W. Sansen, voorzitter
Prof. Dr. ir. S. Vandewalle, promotor
Prof. Dr. ir. P. Dierckx
Prof. Dr. M. J. Gander
(Université de Genève)
Prof. Dr. ir. C. W. Oosterlee
(Technische Universiteit Delft)
Prof. Dr. ir. D. Roose
Prof. Dr. ir. M. Van Barel

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Jan VAN LENT

U.D.C. 681.3*G18, 519.63

7 februari 2006

© Katholieke Universiteit Leuven — Faculteit Ingenieurswetenschappen
Arenbergkasteel, B-3001 Heverlee, Belgium

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

D/2006/7515/12

ISBN 90-5682-678-6

Multigrid Methods for Time-Dependent Partial Differential Equations

Jan Van lent
Departement Computerwetenschappen, K.U.Leuven
Celestijnenlaan 200A, B-3001 Heverlee, België

Abstract

Time-dependent partial differential equations are solved numerically by discretizing both space and time. Since the resulting systems of equations can be very large, it is often necessary to use iterative methods that exploit the structure of these systems. For discretized parabolic problems multigrid methods are a particularly good choice. A typical model problem is the heat equation, discretized using finite differences or finite elements in space and a linear multistep method in time. We investigate here how multigrid techniques can be used for more general time-dependent problems. In particular we develop multigrid methods for anisotropic problems, high order time discretizations and problems with delay. Furthermore, we propose a new framework for the convergence analysis of multigrid methods for time-dependent partial differences equations.

Anisotropic partial differential equations have coefficients with a strong directional dependency. For such problems standard multigrid methods break down. By extending the techniques for stationary anisotropic problems, we develop efficient multigrid methods for time-dependent anisotropic problems. We consider methods based on line relaxation, semicoarsening and multiple semicoarsening. The same methods are also applied with good results to diffusion equations with coefficients that depend on position as well as direction.

Implicit Runge-Kutta methods, boundary value methods and general linear methods are powerful time discretization schemes providing high order

accuracy, good stability and many other desirable properties. For general time-dependent problems, however, the resulting systems of equations are harder to solve than the ones for linear multistep methods. We show that for discretized parabolic problems, very efficient multigrid methods can be developed. The stability of the time discretization schemes turns out to be very important for the convergence of the iterative methods. The same techniques are used to study iterative methods in combination with Chebyshev spectral collocation in time.

For standard time-dependent partial differential equations, the change of state at a certain time only depends on the current state of the system. For delay partial differential equations, the change of state also depends on the state of the system at times in the past. We study iterative methods for diffusion equations with one extra term with a fixed delay.

In all these cases the performance of the methods is assessed with a theoretical convergence analysis and numerical experiments. The theoretical analyses combine the theory of Volterra convolution operators and Laplace transforms for time-dependent problems and Fourier mode techniques for multigrid. We propose a new approach for the spectral analysis of iterative methods based on functional calculus. This theory unifies the Laplace analysis for time-dependent problems and the Fourier analysis of multigrid methods.

Preface

This thesis is a condensation of the research I have done over the past few years. Since it may not be obvious from the following pages, I would like to emphasize here how much I enjoyed doing this research. For me, it is great to have a job that involves spending hours in the library and even more hours reading. I have always enjoyed studying and doing research is essentially just that. However, doing research is not just a solitary endeavor and I would like to thank some of the people that were involved.

I acknowledge the Fund for Scientific Research - Flanders (F.W.O.-Vlaanderen) and the university (Katholieke Universiteit Leuven) for funding my research.

A very important thank you goes to Stefan Vandewalle for being an excellent advisor. Apart from being very knowledgeable about the subject of my research, he also helped me with many practical issues, such as getting papers published. He has introduced me to many interesting people, personally and by encouraging me to attend conferences, workshops, courses and seminars. Together with Dirk Roose, Stefan heads the Scientific Computing Group and they can both be commended for making it a pleasant working environment. Colleagues are of course also a very important part of this environment and my gratitude goes out to all of them and especially to Teo Peeters, Tim Boonen and Jan Maes for the many interesting discussions and to Patrick Van gucht, Gianni Codevico and Raf Vandebril for giving me the opportunity to let off some steam now and again during our lunchtime squash games. From the other staff of the department, I would like to thank in particular the secretaries and system administrators who were always ready to help.

I would like to thank Paul Dierckx and Dirk Roose for their proof-reading and Martin Gander, Kees Oosterlee and Marc Van Barel for agreeing to complete the jury of this thesis. Chapter 6 builds on research done by Jan Janssen and I am grateful for his work. I would like to thank Alfons Van Daele for his help with the theory of Chapter 7.

From the many people I have met abroad, I would specifically like to thank Francesca Mazzia and Felice Iavernaro for making my visit to Bari

such an enjoyable experience. It was always a pleasure to see Nicola Mastronardi again, first in Leuven when he was finishing his PhD and later in Bari and during his many visits to Leuven.

Life is not all about work, so I would also like to thank my friends in Leuven, Bornem, London, Grenada and elsewhere for all the good times.

For their continuing support, I would like to thank my parents, my sister and the rest of my family.

Finally, I would like to thank Tess for being a fantastic girlfriend. It is a joy to share my life with her.

Jan Van lent
Leuven, February 2006

Contents

List of Notations

The following list contains some of the notations used in the text, together with a brief explanation. Note that many common notations are omitted as well as some that are only used locally or infrequently. The numbers in brackets indicate the page where the notation is explained or used for the first time. Many notations are introduced in Chapter 2, others usually near the beginning of a chapter. The list starts with accents, binary operators, delimiters, subscripts, superscripts, symbols, numbers and then continues more or less alphabetical.

$\bar{x}$	coarse grid entity corresponding to x (27)
$\dot{u}$	time derivative of u (16)
$*$	convolution (21)
$\odot$	Hadamard product (61)
$\otimes$	Kronecker product (7)
$\setminus$	set difference (102)
$\|A\|$	norm of matrix A (6)
$\|T\|$	norm of operator T (14)
$\|x\|$	norm of vector x (6)
$\|x\|_X$	norm of x in Banach space X (14)
$\{x_n\}$	sequence of elements x_n (14)
u_n	derivative of u in the direction normal to the boundary (8)
u_t	derivative of u with respect to time (16)
u_x	derivative of u with respect to x (8)
u_{xx}	second derivative of u with respect to x (7)
$u^{(\nu)}$	quantity u at iteration step ν (11)
$\bullet$	value from stencil (11)
$\mathbf{1}_s$	vector containing s ones (84)

$a(w)$	characteristic polynomial of time discretization scheme (101)
α	angle of $A(\alpha)$-stability (126)
α_j	coefficients of time discretization scheme (83)
$b(w)$	characteristic polynomial of time discretization scheme (101)
β	anisotropy angle (37)
β_j	coefficients of time discretization scheme (83)
C	coarse grid correction matrix (28)
$C(X, Y)$	space of continuous functions on X with values in Y (15)
$\bar{\mathbb{C}}^+$	closed right half plane (22)
$\partial\Omega$	boundary of Ω (8)
Δt	time step (16)
Δx	distance between grid points in x-direction (8)
$\det A$	determinant of A (26)
$\mathcal{D}(T)$	domain of operator T (14)
ε	anisotropy coefficient (35)
e_s	unit vector of dimension s with 1 in last position (88)
$\mathcal{F}_\infty^{m\times m}(T)$	family of matrix-valued functions analytic in a neighborhood of $\sigma(T)$ and at infinity (155)
$\mathcal{F}^{m\times m}(T)$	family of matrix-valued functions analytic in a neighborhood of $\sigma(T)$ (152)
$\mathcal{F}_\infty(T)$	family of functions analytic in a neighborhood of $\sigma(T)$ and at infinity (155)
$\mathcal{F}(T)$	family of functions analytic in a neighborhood of $\sigma(T)$ (150)
ϕ	the empty set (22)
H	Laplace transform of convolution kernel h (23)
$\mathcal{H}$	Volterra convolution operator (22)
$\mathcal{H}_{\Delta t}$	discrete Volterra convolution operator (25)
h	measure of grid spacing (26)

I_n	identity matrix of dimension n (10)
K	iteration matrix (21)
$\mathcal{K}$	waveform relaxation operator (19)
$\mathcal{K}_{\Delta t}$	discrete waveform relaxation operator (20)
k	convolution kernel (21)
L^+, L^-	matrix splitting ($L = L^+ + L^-$) (11)
$L^\infty(X, Y)$	space of essentially bounded functions on X with values in Y (15)
$L^p(X, Y)$	space of p-integrable functions on X with values in Y (15)
$l^p(n, X)$	space of p-summable sequences of length n with values in X (15)
M	multigrid matrix (28)
$\mathcal{M}$	multigrid operator (29)
m	number of internal spatial grid points (9)
n_t	number of time steps (32)
ν	iteration step (11)
n_x	number of intervals in the x-direction (8)
Ω	spatial domain (7)
Ω	spatial grid (49)
Ω_t	time domain (37)
P	restriction operator (27)
Ψ	vector of Fourier harmonics (50)
ψ	exponential Fourier mode (49)
$q(\gamma)$	the complex number with modulus 1 and argument γ (49)
R	restriction operator (27)
$R(z)$	stability function (124)
R	convergence rate (13)
$R^{(\nu,\mu)}$	estimated convergence rate (13)
$\rho(A)$	spectral radius of matrix A (6)
$\rho(T)$	spectral radius of operator T (14)
$\rho^{(\nu,\mu)}$	estimated convergence factor (13)

S_α	sector in the complex plane (126)
S	open stability domain (101)
$\mathbb{S}$	unit circle (165)
Σ	subset of the complex plane, complement of open stability domain, spectrum of time derivative operator (20)
$\sigma(A)$	spectrum of matrix A (6)
$\sigma(T)$	spectrum of operator T (14)
$\sigma_\infty(T)$	extended spectrum of T (155)
$\operatorname{span} x$	vectorspace spanned by the elements of x (52)
t_F	endpoint of time interval $[0, t_F]$ (15)
t_F	time period (16)
τ	delay (135)
Θ	set of wavenumbers (49)
Θ'	set of wavenumbers for harmonics (50)
θ	wavenumber (49)
X	Banach space (14)

List of Acronyms

AMG	algebraic multigrid (28, 170)
BBVM	block boundary value method (82, 83, 88–95, 98–105, 109, 117, 122, 124, 128, 160)
BDF	backward difference formula (84, 109, 167)
BGAM	block generalized Adams method (110)
BGBDF	block generalized backward difference formula (110, 125, 131, 132)
BVM	boundary value method (81–83, 86, 88, 91–96, 98–102, 104–106, 109–111, 117, 122–124, 128, 129, 132, 148, 160, 167, 169, 172)
CSC	Chebyshev spectral collocation (122–124, 128, 129, 131, 133)
DCT	discrete cosine transform (120)
DDE	delay differential equation (4, 135–140, 172)
DPDE	delay partial differential equation (136, 137, 146, 172)
FFT	fast Fourier transform (120)
GAM	generalized Adams method (102, 110, 171)
GBDF	generalized backward difference formula (102, 109, 110, 123, 124, 129, 131, 132, 171)
GLM	general linear method (82, 83, 85, 86, 88, 89, 92–95, 99–102, 104–106, 109, 122, 148, 160, 162)
GS	Gauss-Seidel (31)

IRK implicit Runge-Kutta (19, 81–89, 92, 99–102, 117, 118, 122, 124, 126, 128, 148, 160, 167, 168)

LMM linear multistep method (20, 24, 83–87, 90, 92, 94, 97, 98, 100–102, 104–106, 109, 122, 148, 160, 170)

MG multigrid (31)

MGS multigrid as smoother (45, 53, 73, 113, 146)

ODE ordinary differential equation (1–3, 16–19, 21, 37, 40–42, 47, 66, 81–83, 86, 94–96, 98, 102, 113, 121, 122, 128, 133, 135, 136, 139, 148, 160, 167, 170–173)

PDE partial differential equation (1–5, 7, 11, 15–18, 34, 37, 39, 48, 66, 82, 83, 91, 117, 124, 128, 135, 137, 146, 167, 170–173)

RAD Radau IIA (110)

Nederlandse samenvatting

Multiroostermethodes voor tijdsafhankelijke partiële differentiaalvergelijkingen

Inhoudsopgave

1 Inleiding

1.1 Ruimte en tijd

In deze thesis bestuderen we tijdsafhankelijke partiële differentiaalvergelijkingen. Het standaard modelprobleem is de warmtevergelijking

$$u_t = u_{xx} + u_{yy} + f.$$

Dit soort problemen combineert karakteristieken van tijdsonafhankelijke elliptische problemen, enerzijds, en stijve gewone differentiaalvergelijkingen, anderzijds. We bestuderen hoe multiroostertechnieken voor elliptische problemen en geavanceerde tijdsdiscretisatieschema's voor stijve differentiaalvergelijkingen gecombineerd kunnen worden. We beschouwen zowel complicaties met betrekking tot de ruimtelijk veranderlijken, zoals variërende en richtingsafhankelijke coëfficiënten, als complicaties met betrekking tot de tijd, zoals vertraging.

1.2 Golfvormrelaxatie

We gebruiken golfvormrelaxatiemethodes als ruimer kader voor het bestuderen van de combinatie van gesofisticeerde technieken voor gewone differentiaalvergelijkingen en partiële differentiaalvergelijkingen. Golfvormrelaxatiemethodes kunnen beschouwd worden als de uitbreiding van iteratieve methodes voor stelsels van algebraïsche vergelijkingen naar stelsels van gewone differentiaalvergelijkingen. De combinatie van multiroostertechnieken en golfvormrelaxatie werd voor het eerst bestudeerd in [LO87] en [Van93].

1.3 Overzicht

Hoofdstuk 2 bevat een algemene beschrijving van de iteratieve methodes die we hier beschouwen. Ook worden de belangrijkste stellingen voor de convergentieanalyse van deze methodes geïntroduceerd. We beschouwen eerst iteratieve methodes voor stelsels van vergelijkingen, vervolgens methodes voor stelsels van differentiaalvergelijkingen en tenslotte multiroostermethodes, toepasbaar in beide gevallen. Dit hoofdstuk dient als samenvatting van de bekende resultaten uit de literatuur over (multirooster-)golfvormrelaxatie.

In hoofdstuk 3 beschouwen we multiroostermethodes voor anisotrope problemen. Gebaseerd op het elliptische geval ontwikkelen we methodes voor het parabolische geval.

In hoofdstuk 4 bekijken we golfvormrelaxatie voor problemen waarbij de tijd gediscretiseerd wordt met impliciete Runge-Kuttamethodes en randwaardemethodes.

Hoofdstuk 5 bouwt hierop verder en bestudeert Chebyshev spectrale collocatie als tijdsdiscretisatie.

In hoofdstuk 6 worden de iteratieve methodes en hun analyse uitgebreid naar partiële differentiaalvergelijkingen met vertraging.

Hoofdstuk 7 beschrijft een theorie voor de convergentieanalyse van iteratieve methodes voor tijdsafhankelijke problemen gebaseerd op functionele calculus. De theorie wordt ook toegepast op de tweeroosteranalyse van multiroostermethodes.

In hoofdstuk 8 geven we een overzicht van enkele implementaties die multiroostermethodes gebruiken voor tijdsafhankelijke problemen. We eindigen met enkele besluiten en ideeën voor verder onderzoek.

2 Iteratieve methodes voor tijdsafhankelijke problemen

In dit hoofdstuk introduceren we de iteratieve methodes die de basis zullen vormen voor de methodes in de rest van dit werk. We beschrijven eerst

iteratieve methodes voor stelsels van vergelijkingen. Als modelprobleem gebruiken we de Poissonvergelijking. Vervolgens bekijken we hoe gelijkaardige principes gebruikt kunnen worden voor iteratieve methodes voor stelsels van gewone differentiaalvergelijkingen. In dit geval nemen we de warmtevergelijking als modelprobleem. Voor beide gevallen geven we ook de basisprincipes en stellingen die gebruikt worden voor de theoretische analyse van de convergentie.

Voor de Poissonvergelijking en de warmtevergelijking zijn de klassieke iteratieve methodes, zoals de methode van Jacobi of de methode van Gauss-Seidel, erg traag. We beschrijven hoe multiroosterversnelling voor deze gevallen leidt tot efficiënte methodes.

We sluiten dit hoofdstuk af met enkele numerieke resultaten.

2.1 Iteratieve methodes voor vergelijkingen

Als modelprobleem beschouwen we de Poissonvergelijking

$$u_{xx} + u_{yy} = f.$$

Discretisatie met eindige differenties leidt tot een stelsels van de vorm

$$Lu = f,$$

waarbij L een grote, ijle matrix is. Door deze matrix te splitsen als $L = L^+ + L^-$ bekomen we de iteratie

$$L^+ u^{(\nu)} + L^- u^{(\nu-1)} = f.$$

Kiezen we voor L^+ de diagonaal van L, dan bestaat elke iteratiestap uit het oplossen van een reeks scalaire vergelijkingen, één voor elke onbekende. Dit is de klassieke methode van Jacobi. Voor deze methode is de volgorde waarin de scalaire vergelijkingen beschouwd worden niet van belang. Kiezen we voor L^+ het onderdriehoeksdeel van L, dan moeten we in elke iteratiestap nog steeds per onbekende een scalaire vergelijking oplossen. Dit is de klassieke methode Gauss-Seidel. In dit geval is de volgorde van de vergelijkingen echter wel van belang. Er zijn dus verschillende varianten mogelijk.

Als we de fout in iteratiestap ν schrijven als $e^{(\nu)} = u^{(\nu)} - u$, dan kunnen we uit het oorspronkelijke stelsel van vergelijkingen voor u en de iteratie voor $u^{(\nu)}$ de volgende iteratie voor de fout afleiden

$$L^+ e^{(\nu)} = -L^- e^{(\nu-1)}.$$

We kunnen dit ook schrijven als

$$e^{(\nu)} = -(L^+)^{-1} L^- e^{(\nu-1)} = K e^{(\nu-1)}.$$

We noemen $K = -(L^+)^{-1}L^-$ de iteratiematrix. De iteratie zal convergeren als $\rho(K) < 1$. De spectraalradius $\rho(K)$ wordt in dit geval ook wel de convergentiefactor genoemd. Hoe kleiner dit getal, hoe sneller de convergentie. In de numerieke experimenten vergelijken we de theoretische convergentiefactor met een schatting van de vorm

$$\rho^{(\nu)} = \frac{\|e^{(\nu)}\|}{\|e^{(\nu-1)}\|}.$$

Vaak zullen we de convergentiesnelheid

$$R = -\log_{10}(\rho)$$

gebruiken in plaats van de convergentiefactor ρ. De convergentiesnelheid geeft aan hoeveel juiste cijfers er gemiddeld per iteratie bijkomen.

2.2 Iteratieve methodes voor differentiaalvergelijkingen

Als modelprobleem beschouwen we de warmtevergelijking

$$u_t = u_{xx} + u_{yy} + f.$$

Discretisatie van de ruimtelijke veranderlijken met eindige differenties leidt tot een stelsel van gewone differentiaalvergelijkingen van de vorm

$$\dot{u} = Lu + f.$$

De matrix L is dezelfde matrix als bij de gediscretiseerde Poissonvergelijking. Er zijn verschillende manieren om dit stelsel van vergelijkingen op te lossen.

Passen we een schema zoals de impliciete Eulermethode toe, dan bekomen we een sequentie van stelsels van de vorm.

$$u_i = u_{i-1} + \Delta t L u_i + \Delta t f_i.$$

Dit stelsel voor de onbekende u_i kan nu op gelijkaardige manier opgelost worden als de gediscretiseerde Poissonvergelijking. Met een splitsing $L = L^+ + L^-$ wordt de methode: voor elke i, voor elke ν, bereken $u_i^{(\nu)}$ uit

$$u_i^{(\nu)} = u_{i-1}^{(\nu)} + \Delta t L^+ u_i^{(\nu)} + \Delta t L^- u_i^{(\nu-1)} + \Delta t f_i.$$

We kunnen de splitsing $L = L^+ + L^-$ ook rechtstreeks op het stelsel van differentiaalvergelijkingen toepassen. Zo krijgen we een continuë golfvormrelaxatiemethode: voor elke ν, bereken $u^{(\nu)}$ uit

$$\dot{u}^{(\nu)} = L^+ u^{(\nu)} + L^- u^{(\nu-1)} + f.$$

Deze iteratie is interessant om theoretisch te bestuderen, maar voor praktische berekeningen moeten de differentiaalvergelijkingen gediscretiseerd worden. Dit leidt dan tot discrete golfvormrelaxatiemethodes. Passen we de impliciete Eulermethode toe, dan bekomen we de methode: voor elke ν, voor elke i, bereken $u_i^{(\nu)}$ uit

$$u_i^{(\nu)} = u_{i-1}^{(\nu)} + \Delta t L^+ u_i^{(\nu)} + \Delta t L^- u_i^{(\nu-1)} + \Delta t f_i.$$

Merk op dat het enige verschil met de tijdstapmethode de volgorde van de lussen is.

Het verband tussen de fout $e^{(\nu)} = u^{(\nu-1)} - u$ voor opeenvolgende benaderingen kan geschreven worden als

$$e^{(\nu)} = \mathcal{K} e^{(\nu-1)}. \tag{1}$$

De convergentie kan opnieuw geanalyseerd worden aan de hand van de spectraalradius van de iteratieoperator $\mathcal{K}$. Voor de iteratieve methodes voor tijdsafhankelijke problemen die we hier beschouwen, geldt de formule

$$\rho(\mathcal{K}) = \max_{z \in \Sigma} \rho(K(z)).$$

De matrix $K(z)$ hangt af van de complexe parameter z en wordt gegeven door

$$K(z) = (zI - L^+)^{-1} L^-.$$

Dit is de iteratiematrix van de overeenkomstige iteratieve methode voor stelsels toegepast op

$$zu = Lu + f.$$

Afhankelijk van de manier waarop de tijdsdimensie behandeld wordt moet een verschillende verzameling $\Sigma \subset \mathbb{C}$ gebruikt worden. De volgende tabel vat enkele resultaten uit [MN87a, MN87b, LO87, Van93, JV96a, JV96b] samen.

tijd	domein	Σ
continu		
	eindig	$\{\infty\}$
	oneindig	$\bar{\mathbb{C}}^+$
	periodiek	$2\pi i\mathbb{Z}$
discreet		
	eindig	$\{1/\Delta t\}$
	oneindig	$\{(1-w)/\Delta t : \lvert w\rvert \leq 1\}$
	periodiek	$\{(1-w)/\Delta t : w = \exp(2\pi i j/n),\ 0 \leq j < n\}$

2.3 Multiroosterversnelling

Voor gediscretiseerde elliptische en parabolische vergelijkingen zijn de klassieke methodes van Jacobi en Gauss-Seidel erg traag. Deze methodes kunnen echter als basis dienen voor multiroostermethodes. Bij een multiroostermethode worden berekeningen op grovere roosters gebruikt om de iteraties op fijne roosters te versnellen. Stel dat we een benadering $\mathbf{x}^{(1)}$ hebben voor de oplossing $\mathbf{x}$ van het stelsel $\mathbf{Ax} = \mathbf{b}$. We kunnen de correctie $\mathbf{e} = \mathbf{x} - \mathbf{x}^{(1)}$ bepalen door het stelsel $\mathbf{Ae} = \mathbf{b} - \mathbf{Ax}^{(1)}$ op te lossen. Als $\mathbf{x}^{(1)}$ bekomen werd door een relaxatie (zoals de methode van Jacobi of Gauss-Seidel), dan kan de oplossing van dit stelsel goed benaderd worden op een grover rooster.

Algoritme 2.6.1 geeft schematisch weer hoe een tweeroostermethode voor een stelsel van de vorm $\mathbf{Ax} = \mathbf{b}$ eruitziet. Bij een tweeroostermethode wordt de vergelijking $\bar{\mathbf{A}}\bar{\mathbf{x}} = \bar{\mathbf{b}}$ op het grove rooster exact opgelost. We bekomen een multiroostermethode door ook deze vergelijking benaderend op te lossen door gebruik te maken van een nog grover rooster en zo verder tot we op een rooster komen met een stelsel dat klein genoeg is om eenvoudig exact opgelost te worden.

Voor de convergentieanalyse van multiroostermethodes wordt meestal enkel een tweeroosteriteratie beschouwd. De analyse verloopt op dezelfde manier als voor de iteraties op een enkel rooster. In de formule voor de convergentiefactor moet de matrix $K(z)$ vervangen worden door de matrix

$$M(z) = S(z)^{\nu_1} C(z) S(z)^{\nu_2}.$$

De matrices voor de relaxatie S en de grofroostercorrectie C zijn

$$\begin{aligned} S(z) &= (zI - L^+)^{-1} L^-, \\ C(z) &= I - P(zI - \bar{L})^{-1} R(zI - L). \end{aligned}$$

2.4 Numerieke experimenten

Tabel 2.1 geeft convergentiesnelheden voor een Gauss-Seidelmethode en een multiroostermethodes toegepast op de Poissonvergelijking gediscretiseerd op roosters van verschillende fijnheid. Het is duidelijk dat de Gauss-Seidelmethode enkel voor erg grove roosters werkt. De multiroostermethode daarentegen produceert bijna 1 juist cijfer per iteratie, onafhankelijk van de roosterfijnheid.

Tabel 2.2 toont aan dat dezelfde besluiten ook gelden voor de overeenkomstige methodes toegepast op de warmtevergelijking.

3 Anisotrope problemen

We bestuderen multiroostermethodes voor anisotrope tijdsafhankelijke partiële differentiaalvergelijkingen aan de hand van drie modelproblemen. De isotrope diffusievergelijking

$$u_t = u_{xx} + u_{yy} + f,$$

is de warmtevergelijking uit de vorige hoofdstukken. De anisotrope diffusievergelijking

$$u_t = \varepsilon u_{xx} + u_{yy} + f,$$

heeft een parameter ε die de mate van anisotropie aangeeft. De geroteerde anisotrope diffusievergelijking

$$u_t = (\varepsilon c^2 + s^2)u_{xx} + (c^2 + \varepsilon s^2)u_{yy} + 2(\varepsilon - 1)csu_{xy} + f,$$

met $c = \cos\beta$ en $s = \sin\beta$, heeft nog een extra parameter β die de richting van de anisotropie aangeeft. De geroteerde vergelijking kan afgeleid worden van de anisotrope diffusievergelijking door een rotatie van de coördinaatassen over een hoek β.

3.1 Multiroostermethodes

Een multiroostermethode met puntrelaxatie en standaardvergroving werkt goed voor de isotrope diffusievergelijking. Voor de anisotrope vergelijkingen met $\varepsilon \neq 1$ gaat de efficiëntie echter snel verloren.

Bij puntrelaxatie worden de onbekenden roosterpunt per roosterpunt gewijzigd. Om de nieuwe waarde in een roosterpunt te vinden moet slechts een scalaire vergelijking opgelost worden. Voorbeelden zijn Jacobirelaxatie en rood-zwart of vier-kleuren Gauss-Seidelrelaxatie (zie figuren 3.2).

Bij lijnrelaxatie worden de onbekenden geassocieerd met een hele lijn van roosterpunten tegelijkertijd gewijzigd. Voor elke lijn moet een stelsel van vergelijkingen opgelost worden. Deze stelsels zijn echter kleiner en eenvoudiger dan het oorspronkelijke stelsel. Voorbeelden zijn horizontale en verticale lijn Gauss-Seidelrelaxatie en horizontale en verticale zebra Gauss-Seidelrelaxatie (zie figuren 3.3). Een horizontale gevolgd door een verticale lijnrelaxatie noemen we alternerende lijnrelaxatie.

Bij standaard vergroving wordt de roosterafstand in beide richtingen verdubbeld om een grover rooster te bekomen. Figuur 3.4 toont een typische hiërarchie van roosters.

Bij semivergroving wordt de roosterafstand slechts voor een van de coordinaatrichtingen verdubbeld. De roosters op de eerste rij van figuur 3.6 komen overeen met semivergroving in de x-richting. De eerste kolom komt overeen met semivergroving in de y-richting.

Bij meervoudige semivergroving worden alle roosters uit figuur 3.6 gebruikt. We gebruiken hier de "multirooster als relaxatie" methode [Oos95, WO98]. Deze methode is gebaseerd op een multiroostercyclus met standaardvergroving (de diagonaal van figuur 3.6) waarbij de relaxatie voor elk van deze roosters bestaat uit een multiroostercyclus met semivergroving in de x-richting gevolgd door een multiroostercyclus met semivergroving in de y-richting.

3.2 Convergentieanalyse

Voor de convergentieanalyse beschouwen we een tweeroosteriteratie. Om de analyse nog verder te vereenvoudigen werken we met periodieke roosters. Dit laat een analyse toe op basis van Fouriermodes. De formule voor de spectraalradius wordt dan

$$\rho(\mathcal{M}) = \max_{z\in\Sigma} \max_{\theta\in\Theta'} \rho(M(z,\theta)).$$

De verzameling $\Sigma \subset \mathbb{C}$ is dezelfde als in het voorgaande hoofdstuk en hangt af van de manier waarop de tijdsdimensie behandeld wordt. De verzameling $\Theta' \subset [-\frac{\pi}{2}, \frac{\pi}{2}]^2$ bevat de golfgetallen van de Fouriermodes. De matrix $M(z,\theta)$ is een 4×4 matrix. De opbouw van deze matrix voor standaard vergroving en semivergroving wordt weergegeven in figuur 3.9.

3.3 Numerieke experimenten

Tabel 3.7 vat de resultaten voor de anisotrope diffusievergelijking samen (zie ook tabel 3.2). Puntrelaxatie met standaard vergroving werkt enkel voor $\varepsilon = 1$. Horizontale lijnrelaxatie met standaard vergroving werkt voor $\varepsilon > 1$ en verticale lijnrelaxatie werkt voor $\varepsilon < 1$. Puntrelaxatie met semivergroving werkt enkel voor beperkte bereiken van ε. Puntrelaxatie met meervoudige semivergroving werkt voor alle ε, net als alternerende lijnrelaxatie met standaard vergroving en horizontale lijnrelaxatie met semivergroving in de y-richting (en vice versa).

Analoge conclusies gelden voor de geroteerde anisotrope diffusievergelijking als de anisotropie in de richting van de coördinaatassen ligt (β een meervoud van 90°). Voor andere waarden van β kunnen alternerende lijnrelaxatie met standaard vergroving, puntrelaxatie met meervoudige semivergroving en lijnrelaxatie met semivergroving nog steeds gebruikt worden. De efficiëntie neemt echter af. De methode gebaseerd op meervoudige semivergroving lijkt de meest robuuste.

Numerieke experimenten tonen aan dat de voorgaande methodes ook bruikbaar zijn voor meer algemene diffusievergelijkingen van de vorm

$$u_t = (au_x)_x + (bu_y)_y + f,$$

waarbij de coëfficiënten $a(x, y)$ en $b(x, y)$ kunnen afhangen van x en y en erg verschillend kunnen zijn (zie tabel 3.11).

4 Hoge-orde tijdsdiscretisatieschema's

In de vorige hoofdstukken beschouwden we enkel eenvoudige tijdsdiscretisatieschema's zoals de impliciet Eulermethode. Meer gesofistikeerde schema's zoals impliciete Runge-Kuttamethodes en randwaardemethodes bieden veel voordelen, maar resulteren in grotere stelsels die in het algemeen moeilijker op te lossen zijn. Voor parabolische problemen kunnen we echter efficiënte multiroostermethodes ontwikkelen.

4.1 Tijdsdiscretisatieschema's

We beschrijven de beschouwde tijdsdiscretisatieschema's aan de hand van de scalaire gewone differentiaalvergelijking

$$\dot{y}(t) = f(t, y(t))$$

We geven voor elke methode de uitdrukking voor de benadering $y_i = y(t_i)$ op het tijdstip $t_i = i\Delta t$, met Δt de tijdstap.

Een k-staps lineaire multistapmethode (LMM) gebruikt benaderingen uit de vorige stappen in een formule van de vorm

$$\sum_{j=-k}^{0} \alpha_{k+j} y_{i+j} = \Delta t \sum_{j=-k}^{0} \beta_{k+j} f(t_{i+j}, y_{i+j}),$$

waarbij α_k en β_j de coëfficiënten van de methode zijn.

Een impliciete Runge-Kuttamethode (IRK) gebruikt een aantal tussenliggende waarden $\tilde{y}_i$ om een nieuwe benadering y_i te berekenen uit de voorgaande benadering y_{i-1}. De berekening van $\tilde{y}_i$ en y_i verloopt volgens de formules

$$\begin{aligned} \tilde{y}_i &= \mathbf{1}_s y_{i-1} + \Delta t A f(\tilde{t}_i, \tilde{y}_i), \\ y_i &= y_{i-1} + \Delta t b^T f(\tilde{t}_i, \tilde{y}_i), \end{aligned}$$

waarbij de matrix A en de vector b de coëfficiënten van de methode bevatten.

Algemene lineaire methode (GLM) kunnen beschouwd worden als een veralgemening van impliciete Runge-Kuttamethodes waarbij niet alleen $\tilde{y}_i$, maar ook y_i een vector met meerdere waarden is. Deze vectoren worden berekend uit

$$\begin{aligned} \tilde{y}_i &= C y_{i-1} + \Delta t A f(\tilde{t}_i, \tilde{y}_i), \\ y_i &= D y_{i-1} + \Delta t B f(\tilde{t}_i, \tilde{y}_i), \end{aligned}$$

waarbij de matrices A, B, C en D de coëficiënten van de methode bevatten. Zowel lineaire multistapmethodes als impliciete Runge-Kuttamethodes kunnen als algemene lineaire methodes geformuleerd worden.

Bij een k-staps randwaardemethodes (BVM) wordt voor elke tijdstip t_i een formule van de vorm

$$\sum_{j=-k_1}^{k_2} \alpha_{k_1+j} y_{i+j} = \Delta t \sum_{j=-k_1}^{k_2} \beta_{k_1+j} f(t_{i+j}, y_{i+j}).$$

Dit kan beschouwd worden als een veralgemening van de lineaire multistapmethodes. De waarden y_i worden bekomen door het oplossen van een stelsel met bandbreedte k dat niet noodzakkelijk onderdriehoeks is, zoals bij lineaire multistapmethodes wel het geval.

Blok-randwaardemethodes (BBVM) splitsen het tijdsinterval op in subintervallen en passen op elk subinterval een randwaardemethode toe. De vergelijkingen voor een subinterval kunnen geschreven worden als

$$Ay_i + A_0 y_{i-1} = \Delta t B f(t_i, y_i) + \Delta t B_0 f(t_{i-1}, y_{i-1}).$$

Elke blok-randwaardemethode kan geformuleerd worden als een algemene lineaire methode en vice versa.

4.2 Tijdsintegratie

Zoals in de voorgaande hoofdstukken beschouwen we een lineair stelsel van differentiaalvergelijkingen

$$\dot{u} = Lu + f.$$

dat mogelijk afkomstig is van een ruimtelijke discretisatie van een tijdsafhankelijke partiële differentiaalvergelijking. Tijdsdiscretisatie leidt tot een groot stelsel van vergelijkingen dat we via een splitsing $L = L^+ + L^-$ iteratief kunnen oplossen. Als we voor de tijdsdiscretisatie een algemene linaire methode gebruiken dan bekomen we de iteratie

$$\begin{aligned}
\tilde{u}_i^{(\nu)} = {} & (C \otimes I_m) u_{i-1}^{(\nu)} + \Delta t (A \otimes L^+) \tilde{u}_i^{(\nu)} \\
& + \Delta t (A \otimes L^-) \tilde{u}_i^{(\nu-1)} + \Delta t (A \otimes I_m) \tilde{f}_i, \\
u_i^{(\nu)} = {} & (D \otimes I_m) u_{i-1}^{(\nu)} + \Delta t (B \otimes L^+) \tilde{u}_i^{(\nu)} \\
& + \Delta t (B \otimes L^-) \tilde{u}_i^{(\nu-1)} + \Delta t (B \otimes I_m) \tilde{f}_i.
\end{aligned}$$

Als we een blok-randwaardemethode gebruiken dan bekomen we

$$\begin{aligned}
& (A \otimes I_m) u_i^{(\nu)} + (A_0 \otimes I_m) u_{i-1}^{(\nu)} = \\
& \quad \Delta t (B \otimes L^+) u_i^{(\nu)} + \Delta t (B \otimes L^-) u_i^{(\nu-1)} + \Delta t (B \otimes I_m) f_i + \\
& \quad \Delta t (B_0 \otimes L^+) u_{i-1}^{(\nu)} + \Delta t (B_0 \otimes L^-) u_{i-1}^{(\nu-1)} + \Delta t (B_0 \otimes I_m) f_{i-1}.
\end{aligned}$$

Merk op dat wanneer L^+ diagonaal of onderdriehoeks is, in elk roosterpunt een stelsel met een matrix van de vorm $I - lA$ of $A - lB$) moet opgelost worden. Dit kan efficiënt gebeuren vermits het om volle matrices of bandmatrices gaat. Voor gediscretiseerde parabolische vergelijkingen kunnen de bovenstaande iteratieve methodes als relaxatie in een multiroostermethode gebruikt worden.

4.3 Convergentieanalyse en stabiliteit

De convergentiefactor van multiroostermethodes voor tijdsafhankelijke problemen wordt gegeven door de formule

$$\rho = \sup_{z \in \frac{1}{\Delta t}\partial\Sigma} \rho(M(z)).$$

De functie $\rho(M(z))$ kan geschat worden met een Fourieranalyse van de tweeroosteriteratie voor de vergelijking

$$zu = Lu + f.$$

De onderstaande tabel beschrijft de verzamelingen Σ voor verschillende tijdsdiscretisatieschema's. De eerste rij verwijst naar tijdstapmethodes en discrete golfvormrelaxatie op eindige intervallen. De tweede rij verwijst naar discrete golfvormrelaxatie op oneindige intervallen ($|w| \geq 1$). Merk op dat de eerste rij overeenkomt met $|w| \to \infty$.

	GLM	BBVM
$[0, t_F]$	$\sigma(A^{-1})$	$\sigma(B^{-1}A)$
$[0, \infty]$	$\sigma\left((A + C(wI_r - D)^{-1}B)^{-1}\right)$	$\sigma\left((B_0 + wB)^{-1}(A_0 + wA)\right)$

Vermits $\Sigma_{\text{eindig}} \subset \Sigma_{\text{oneindig}}$ hebben we steeds $\rho_{\text{eindig}} \leq \rho_{\text{oneindig}}$. We tonen aan dat voor alle tijdsdiscretisatieschema's geldt

$$\Sigma_{\text{oneindig}} = \mathbb{C} \setminus S,$$

met S het open stabiliteitsgebied van het tijdsdiscretisatieschema. Er is dus een nauw verband tussen de convergentie van de iteratieve methodes en de stabiliteit van het tijdsdiscretisatieschema. Een tijdsdiscretisatieschema is A-stabiel als $S \subset \bar{\mathbb{C}}^-$. Dit is equivalent met $\Sigma \subset \bar{\mathbb{C}}^+$. De convergentiefactor voor continuë golfvormrelaxatie wordt gevonden met $\Sigma = \bar{\mathbb{C}}^+$. Deze convergentiefactor vormt dus steeds een bovengrens voor de convergentiefactor van de overeenkomstige discrete methodes als een tijdsdiscretisatieschema gebruikt wordt dat A-stabiel is.

4.4 Numerieke experimenten

Uit de theoretische analyse en numerieke experimenten blijkt dat multiroostermethodes uitstekend werken in combinatie met stabiele tijdsdiscretisaties. De convergentiesnelheid van de MGS methode toegepast op de vergelijking

$$\begin{aligned} u_t &= (au_x)_x + (bu_y)_y + f, \\ a(x,y,t) &= \exp(10(x-y)), \\ b(x,y,t) &= \exp(-10(x-y)), \end{aligned}$$

gediscretiseerd met verschillend tijdsdiscretisatieschema's, wordt gegeven in de tabel 4.5. Per iteratie neemt de nauwkeurigheid van de oplossing gemiddeld met meer dan een juist cijfer toe.

5 Spectrale tijdsdiscretisatieschema's

Uit het voorgaande hoofdstuk is gebleken dat multiroostermethodes erg efficiënt zijn voor diffusieproblemen, zelfs als geavanceerde tijdsdiscretisatieschema's gebruikt worden. Het is wel van groot belang dat het gebruikte schema goede stabiliteitseigenschappen heeft. De blok-randwaardemethodes uit het vorige hoofdstuk zijn enkel A-stabiel voor lage ordes van nauwkeurigheid. Stabiele methodes met een hoge orde van nauwkeurigheid kunnen bekomen worden door discretisaties op niet equidistante punten te beschouwen. We bestuderen een klasse van schema's gebaseerd op de Chebyshevpunten.

5.1 Chebyshev spectrale collocatie

Een interpolerende spectrale methode benadert de afgeleide van een functie v in een interval als volgt: Kies punten x_j in het beschouwde interval. Construeer de interpolerende veelterm p zodat $p(x_j) = v(x_j)$. Benader $v'(x_j)$, de afgeleide van de functie v in x_j met $w_j = p'(x_j)$, de afgeleide van de veelterm in x_j.

Het is duidelijk dat spectrale differentiatie een lineaire operatie is. We kunnen het verband tussen w en v dus schrijven als

$$w = Dv.$$

Voor de Chebyshevpunten

$$x_j = \cos\frac{j\pi}{n}, \qquad j = 0, \ldots, n.$$

is de differentiatiematrix D eenvoudig te construeren.

We kunnen nu deze spectrale differentiatie gebruiken om de differentiaalvergelijking

$$\dot{v}(t) = f(v(t))$$

te discretiseren. Nemen we als benadering voor de onbekende functie v de interpolerende veelterm door de onbekende waarden v_j in de punten x_j, dan bekomen we het stelsel

$$Dv = f(v).$$

Dit proces noemen we Chebyshev spectrale collocatie. We kunnen aantonen dat deze methode kan geïnterpreteerd worden als een impliciete Runge-Kuttamethode en als een blok-randwaardemethode.

5.2 Stabiliteit

Uit het vorige hoofdstuk weten we dat de convergentie van iteratieve methodes voor tijdsafhankelijke problemen nauw samenhangt met de stabiliteit van het gebruikte tijdsdiscretisatieschema. We onderzoeken daarom de stabiliteit van Chebyshev spectrale collocatie. Het blijkt dat enkel de methodes van orde 1 en 2 A-stabiel zijn. Alle methodes van hogere orde zijn echter $A(\alpha)$-stabiel met α dicht bij $90°$.

5.3 Numerieke experimenten

We kunnen de methode die afgeleid werd voor scalaire differentiaalvergelijkingen ook toepassen voor de tijdsdiscretisatie van stelsels van differentiaalvergelijkingen. Voor de vergelijking

$$\dot{u} = Lu + f$$

bekomen we

$$(D \otimes I_m) = (I_{n+1} \otimes L)u + f.$$

Voor een algemene matrix L is het niet eenvoudig dit stelsel van dimensie $(n+1)m$ op te lossen. Als de differentiaalvergelijking een gediscretiseerde diffusievergelijking is, kunnen we echter een multiroostermethode gebruiken. Typisch moet dan in elk roosterpunt een stelsel met een dichte matrix van dimensie $n+1$ opgelost worden. De afzonderlijke iteraties kunnen dus efficient geïmplementeerd worden. Vermits Chebyshev spectrale collocatie goede stabiliteit vertoont, kunnen we bovendien verwachten dat het totale aantal iteraties klein zal zijn.

Figuur 5.7 toont de convergentie van de norm van het verschil tussen de exacte continuë oplossing en de discrete benadering voor een standaard multiroostermethode toegepast op de warmtevergelijking

$$u_t = u_{xx} + u_{yy} + f.$$

Het aantal onbekenden is steeds hetzelfde. Het is duidelijk dat de methodes met een hoge orde van nauwkeurigheid met hetzelfde aantal onbekenden een veel kleinere fout bekomen. Bovendien presteert Chebyshev spectrale collocatie (CSC) voor orde 20 merkelijk beter dan de methodes gebaseerd op equidistante punten (GBDF, BGBDF).

De convergentiesnelheden in tabel 5.3 tonen aan dat de multiroostermethodes erg efficiënt zijn voor de methodes van lage orde. Voor orde 20 is de convergentie van de methodes gebaseerd op equidistante punten erg traag. De prestaties van de bijna A-stabiele Chebyshev spectrale collocatie blijven echter goed.

We besluiten dat tijdsdiscretisatieschema's gebaseerd op Chebyshev spectrale collocatie nauwkeurigheid van hoge orde en goede stabiliteit toelaten. Deze laatste eigenschap zorgt voor goede convergentie bij multiroostermethodes toegepast op gediscretiseerde diffusievergelijkingen.

6 Problemen met vertraging

We onderzoeken iteratieve methodes voor tijdsafhankelijke partiële differentiaalvergelijking met vertraging aan de hand van het modelprobleem

$$u_t(t,x,y) = u_{xx}(t,x,y) + u_{yy}(t,x,y) - u(t-\tau,x,y).$$

Discretisatie leidt tot een stelsel van differentiaalvergelijkingen met vertraging

$$\dot{u}(t) = Lu(t) + bu(t-\tau).$$

6.1 Golfvormrelaxatiemethodes

Gebaseerd op een splitsing $L = L^+ + L^-$ zijn twee iteraties mogelijk. Bij de Picarditeratie wordt voor de term met vertraging de oude benadering gebruikt. Het stelsel van differentiaalvergelijkingen met vertraging wordt omgezet naar een stelsel van gewone differentiaalvergelijkingen

$$\dot{u}^{(\nu)}(t) - L^+u^{(\nu)}(t) = L^-u^{(\nu-1)}(t) - u^{(\nu-1)}(t-\tau).$$

Bij de niet-Picarditeratie blijft de term met vertraging behouden. Het stelsel van differentiaalvergelijkingen met vertraging wordt omgezet naar een eenvoudiger stelsel van differentiaalvergelijkingen nog steeds met vertraging

$$\dot{u}^{(\nu)}(t) - L^+u^{(\nu)}(t) + u^{(\nu)}(t-\tau) = L^-u^{(\nu-1)}(t).$$

6.2 Convergentieanalyse

De convergentiefactor van de iteraties kan geschreven worden als

$$\rho(\mathcal{K}) = \sup_{z\in\bar{\mathbb{C}}^+} \rho(K(z)) = \sup_{\xi\in\mathbb{R}} \rho(K(i\xi)),$$

met voor het Picard geval

$$K(z) = (zI - L^+)^{-1}(-e^{-\tau z}I + L^-)$$

en voor het niet-Picard geval

$$K(z) = ((z + e^{-\tau z})I - L^+)^{-1}L^-.$$

De iteraties kunnen als relaxatie gebruikt worden in een multiroostermethode. De convergentieanalyse verloopt analoog aan het geval zonder vertraging.

6.3 Numerieke experimenten

Tabel 6.2 geeft convergentiefactoren voor een multiroostermethode toegepast op het modelprobleem met verschillende waarden voor de vertraging τ en de grootte van het ruimtelijke domein d. Als relaxatie gebruikt de multiroostermethode een Picard- of een niet-Picarditeratie. In beide gevallen is de multiroostermethode efficiënt.

Ook voor een vergelijking met variabele coëficiënten blijken de multiroostermethodes erg efficiënt.

7 Convergentieanalyse met functionele calculus

Om de convergentie van een iteratieve methode theoretisch te analyseren gaan we op zoek naar de spectraalradius van de bijhorende iteratieoperator. Voor een stelsel van vergelijkingen van de vorm

$$\dot{u} = Lu + f,$$

bekomen we met een splitsing $L = L^+ + L^-$ de iteratie

$$\dot{u}^{(\nu)} = L^+u^{(\nu)} + L^-u^{(\nu-1)} + f.$$

Afhankelijk van de manier waarop de tijdsdimensie behandeld wordt (tijdstap/golfvormrelaxatie, continu/discreet, eindig/oneindig/periodiek, verschillende tijdsdiscretisaties) krijgen we verschillende iteratieve methodes. De

formule voor de spectraalradius van de overeenkomstige iteratieoperatoren is steeds van de vorm

$$\rho(\mathcal{K}) = \sup_{z\in\Sigma} \rho((zI_m - L^+)^{-1}L^-).$$

De afleiding van deze formule is echter bij elk van de varianten op verschillende theorieën gebaseerd. We introduceren hier een aanpak gebaseerd op functionele calculus die de verschillende analyses verenigt.

7.1 Functionele calculus

Voor een gegeven operator T wijst een functionele calculus aan een functie f van een complexe parameter een operator $f(T)$ toe. De toewijzing is zodanig dat de rekenregels die gelden voor functies, ook gebruikt kunnen worden voor de overeenkomstige operatoren. De functionele calculus voor scalaire functies f is welbekend. We gebruiken hier een veralgemening voor matrixwaardige functies.

Voor een operator T en een matrixwaardige functie $F : \mathbb{C} \to \mathbb{C}^{m\times m}$ die analytisch is in een omgeving van $\sigma(T)$ definiëren we

$$F(T) = \frac{1}{2\pi i} \oint F(z) \otimes (zI_X - T)^{-1} dz.$$

De rekenregels die gelden voor aldus gedefinieerde operatoren zijn

$$\begin{aligned} \alpha F(T) + \beta G(T) &= (\alpha F + \beta G)(T) && \text{(lineaire combinatie)},\\ F(T)\cdot G(T) &= (F\cdot G)(T) && \text{(vermenigvuldiging)},\\ G(F(T)) &= (G\circ F)(T) && \text{(samenstelling van functies)}. \end{aligned}$$

Op basis van deze regels kunnen we bewijzen dat het spectrum van de operator $F(T)$ wordt gegeven door de formule

$$\sigma(F(T)) = \bigcup_{z\in\sigma(T)} \sigma(F(z)) =: \sigma(F(\sigma(T))).$$

Hieruit volgt dat de spectraalradius van $F(T)$ gegeven wordt door

$$\rho(F(T)) = \max_{z\in\sigma(T)} \rho(F(z)).$$

Deze formules zijn geldig wanneer T een matrix of een begrensde operator is. Als we $\sigma(T)$ vervangen door $\sigma_\infty(T) = \sigma(T)\cup\{\infty\}$, dan zijn de formules ook geldig voor gesloten operatoren.

7.2 Analyse van golfvormrelaxatiemethodes

De voorgaande theorie laat ons toe om de iteratieoperatoren van een heel gamma aan methodes op te bouwen, uitgaande van eenvoudige operatoren. Als we het spectrum van deze eenvoudige operatoren kennen, dan volgt onmiddellijk het spectrum van de iteratieoperator.

We beschouwen een algemeen stelsel

$$(I_m \otimes T)u = (L \otimes I_X)u + f$$

met de bijhorende iteratie

$$(I_m \otimes T)u^{(\nu)} = (L^+ \otimes I_X)u^{(\nu)} + (L^- \otimes I_X)u^{(\nu-1)} + f.$$

De iteratieoperator kan geschreven worden als

$$F(T) = (I_m \otimes T - L^+ \otimes I_X)^{-1}(L^- \otimes I_X),$$

waarbij de matrixwaardige functie F gegeven wordt door

$$F(z) = (zI - L^+)^{-1}L^-.$$

Deze functie is analytisch in $\mathbb{C} \setminus \sigma(L^+)$. Als we veronderstellen dat

$$\sigma(L^+) \cap \sigma(T) = \phi,$$

dan worden het spectrum en de spectraalradius van $F(T)$ beschreven door de formules uit de vorige paragraaf. De verschillende iteratieve methodes worden nu bekomen door een gepaste keuze van de operator T en de analyse is gereduceerd tot het bepalen van het spectrum van deze eenvoudigere operator.

Voor continuë golfvormrelaxatie kiezen we $T = \frac{d}{dt}$. Het spectrum van deze operator hangt af van het type van problemen. We vinden voor beginwaardeproblemen op eindige tijdsintervallen

$$\sigma_\infty(T) = \{\infty\}.$$

Op oneindige tijdsintervallen wordt dat

$$\sigma_\infty(T) = \bar{\mathbb{C}}^-.$$

Voor periodieke problemen bekomen we

$$\sigma_\infty(T) = 2\pi i\mathbb{Z}.$$

Bij discrete golfvormrelaxatie is T een discrete benadering van $\frac{d}{dt}$. Voor de impliciete Eulermethode hebben we bijvoorbeeld

$$(Tx)_i = \frac{x_i - x_{i-1}}{\Delta t}.$$

In het algemene geval kan T zelf geschreven worden als een functie van een eenvoudigere operator. Voor een algemene lineaire methode met als coëficiëntenmatrices A, B, C en D vinden we

$$T = G(S),$$

met S een verschuivingsoperator en

$$G(z) = \frac{1}{\Delta t}(A + zC(I_r - zD)^{-1}B)^{-1}.$$

In het geval van de impliciete Eulermethode wordt deze functie

$$G(w) = g(w) = \frac{1-w}{\Delta t}.$$

Voor beginwaardeproblemen op eindige tijdsintervallen vinden we

$$\sigma(S) = \{0\}.$$

Voor beginwaardeproblemen op oneindige tijdsintervallen wordt dat

$$\sigma(S) = \{w \in \mathbb{C} : |w| \leq 1\}.$$

Voor periodieke problemen bekomen we

$$\sigma(S) = \{w = \exp\left(\frac{2\pi ij}{n}\right), j = 0, \ldots, n-1\}.$$

7.3 Analyse van multiroostermethodes

Niet alleen de Laplaceanalyse van iteratieve methodes voor tijdsafhankelijke problemen, maar ook de Fourieranalyse van multiroostermethodes kan ingepast worden in deze theorie. We beschouwen hiertoe de tweeroosteriteratie voor een probleem op periodieke of oneindige roosters. De iteratieoperator voor een tweedimensionale vergelijking kan geschreven worden als

$$\mathcal{M} = M(P_x, P_y),$$

met P_x en P_y periodieke (of oneindige) verschuivingsoperatoren en

$$M : \mathbb{C}^2 \to \mathbb{C}^{4\times 4} : (w_x, w_y) \to M(w_x, w_y)$$

een matrixwaardige functie met twee complexe parameters w_x en w_y. In combinatie met de analyse voor tijdsafhankelijke problemen leidt dit uiteindelijk tot een formule van de vorm

$$\rho(\mathcal{M}) = \max_{|w_t|=|w_x|=|w_y|=1} \rho(M(w_t, w_x, w_y)),$$

waarbij $M(w_t, w_x, w_y)$ een kleine matrix is met 3 complexe parameters.

8 Slotbemerkingen

8.1 Implementatieaspecten

Voor de numerieke experimenten in hoofdstuk 3 gebruikte ik een code geschreven in de programmeertaal C++. Het merendeel van de methodes heb ik later opnieuw geïmplementeerd in Python. Het resulterende programma is beter leesbaar en makkelijker aan te passen. Meestal zijn programma's geschreven in Python minder effciënt dan C++ programma's, maar door het gebruik van gepaste bibliotheken was dit voor onze toepassing niet dramatisch.

De convergentieanalyse had ik oorspronkelijk geïmplementeerd in Matlab. De code was niet erg efficiënt en om ze efficiënt te maken zou een grondige herstructurering nodig geweest zijn. Ik koos daarom voor een nieuwe implementatie die de programmeertaal ZPL voor de rekenintensieve componenten combineert met Python.

Tijdens een bezoek aan de universiteit van Bari integreerde ik in samenwerking met Francesca Mazzia een multiroostermethode in GAMD, een pakket voor het oplossen van beginwaardeproblemen.

Multiroostermethodes kunnen ook gebruikt worden voor tijdsafhankelijke problemen gediscretiseerd met eindige elementen in plaats van eindige differenties. Er werden enkele experimenten gedaan voor problemen gediscretiseerd met eindige elementen in de ruimte en een impliciete Runge-Kuttamethode in de tijd. We gebruikten onder andere Femlab, een pakket voor het oplossen van partiële differentiaalvergelijkingen met de eindige elementenmethode en SAMG, een algebraïsche multiroostermethode voor stelsels van partiële differentiaalvergelijkingen.

8.2 Samenvatting en besluiten

Hoofdstuk 2 toonde aan de hand van twee modelproblemen hoe multiroostermethodes zowel voor tijdsonafhankelijke als voor tijdsafhankelijke problemen gebruikt kunnen worden. In de volgende hoofdstukken werd geïllustreerd dat zowel de methodes als hun analyse uitgebreid kunnen worden naar ruimere klasses van problemen.

In hoofdstuk 3 beschouwden we methodes voor anisotrope problemen.

De hoofdstukken 4 en 7 toonden aan dat er een nauw verband is tussen de convergentie van iteratieve methodes voor tijdsafhankelijke problemen en de stabiliteit van het gebruikte tijdsdiscretisatieschema.

We konden besluiten dat de combinatie van een gepaste multiroostermethode met een gepast tijdsdiscretisatieschema leidt tot een iteratieve methode die ongeveer even snel convergeert als de multiroostermethode voor het overeenkomstige tijdsonafhankelijke probleem. Het werk per iteratie

komt ongeveer overeen met het werk per iteratie voor het tijdsonafhankelijke geval vermenigvuldigd met de kost voor het oplossen van een scalaire differentiaalvergelijking.

De hoofdstukken 4 en 5 toonden aan dat wanneer multiroostermethodes gebruikt kunnen worden, het de moeite waard is om het gebruik van geavanceerde tijdsdiscretisatieschema's te overwegen.

In hoofdstuk 6 werd aangetoond dat golfvormrelaxatie ook gebruikt kan worden voor partiële differentiaalvergelijkingen met vertraging.

Hoofdstuk 7 introduceerde een elegante manier om de spectra van een groot aantal iteratieoperatoren te bepalen.

8.3 Suggesties voor verder onderzoek

Een huidige onderzoekspiste is het gebruik van algebraïsche multiroostermethodes voor het oplossen van grote eindige elementendiscretisaties van tijdsafhankelijke partiële differentiaalvergelijkingen. Dit soort methodes zijn interessant wanneer onregelmatige en adaptieve roosters nuttig zijn.

Bij sommige van de besproken methodes moet voor elk ruimtelijk roosterpunt een groot stelsel opgelost worden. We gebruikten hiervoor directe methodes voor volle of ijle matrices. Voor randwaardemethodes van hoge orde met veel tijdstappen zouden gepreconditioneerde iteratieve methodes gebruikt kunnen worden. Voor spectrale tijdsdiscretisaties zouden methodes gebaseerd op snelle Fourier- of consinustransformaties gebruikt kunnen worden.

Het zou interessant kunnen zijn om bestaande pakketten voor de tijdsintegratie van differentiaalvergelijkingen en differentiaalvergelijkingen met vertraging te gebruiken als componenten van een multirooster-golfvormrelaxatiemethode.

De theorie van hoofdstuk 7 kan uitgebreid worden naar differentiaalvergelijkingen met vertraging. Niet alleen spectra maar ook pseudospectra kunnen met functionele calculus bestudeerd worden. Vele formules kunnen eenvoudiger voorgesteld worden met veralgemeende eigenwaarden. In dit verband kan ook gekeken worden naar de toepassing van de theorie van samengestelde spectra. Die theorie en de verwante theorie van tensorproducten van operatoren kunnen gebruikt worden voor een uitbreiding van scalaire en matrixwaardige functies naar operatorwaardige functies.

Chapter 1

Introduction

1.1 Space and Time

The main class of problems studied in this thesis are time-dependent parabolic partial differential equations (PDE). The standard model problem for this class is the heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f.$$

Such problems combine characteristics of both time-independent elliptic PDEs and stiff ordinary differential equations (ODE).

To see the link with stationary elliptic PDEs, one can discretize time using, for example, the implicit Euler method. This results in a sequence of elliptic PDEs of the form

$$u_i - \Delta t \left(\frac{\partial^2 u_i}{\partial x^2} + \frac{\partial^2 u_i}{\partial y^2} \right) = u_{i-1} + \Delta t f_i.$$

It is well established that multigrid methods are among the most efficient methods to solve such elliptic equations.

To see the link with stiff ODEs, one can discretize space using, for example, finite differences. This results in a system of ODEs of the form

$$\dot{u} = Lu + f.$$

For discretized parabolic equations the system of ODEs is stiff and the matrix L is large, sparse and structured. Many sophisticated time discretization schemes are available to solve stiff systems of ODEs, but because the systems can be very large, it is necessary to exploit their special structure.

In this thesis we study the use of efficient multigrid techniques, traditionally used for elliptic equations, to exploit the structure arising from discretizing the spatial variables of a parabolic equation. For the time dimension, we use advanced time discretization schemes developed for stiff systems of ODEs. From the PDE point of view, the methods have to be extended so that schemes other than the implicit Euler method can be used. From the ODE point of view, we are looking for methods to efficiently solve the large systems of equations involved. Complications with respect to the spatial variables, such as varying and directionally dependent coefficients, as well as complication with respect to time, such as time delays, are considered.

1.2 Waveform Relaxation

The question of how to combine sophisticated techniques for ODEs and PDEs is studied in the general framework of waveform relaxation. Waveform relaxation methods are iterative methods obtained by extending iterative methods for systems of scalar equations to systems of differential equations.

Waveform relaxation was first studied as a practical solution method in the context of integrated circuit simulation in [LRSV82, WSVOR85, WSV87]. The convergence of waveform relaxation for linear systems of ODEs was studied by Miekkala and Nevanlinna in [MN87a]. The combination of waveform relaxation with multigrid techniques was first studied in [LO87] and [Van93]. Many other iterative methods have been extended to waveform relaxation methods (see, for example, [Jan97, LW03]).

1.3 Outline

Chapter 2 contains a general description of the iterative methods considered in this work. The main theorems used for the convergence analysis of these methods are also presented. First, iterative methods for systems of equations are considered, then methods for systems of ODEs, and finally multigrid methods, applicable in both cases. This chapter serves as a summary of well known results from the (multigrid) waveform relaxation literature.

In Chapter 3 the model problems discussed in Chapter 2 are extended with respect to their spatial interactions. More specifically, multigrid methods for anisotropic problems are considered. Based on experience from the elliptic case, we develop methods for the parabolic case. The convergence analysis combines the standard waveform relaxation convergence theory with a two-grid Fourier mode analysis of multigrid for anisotropic problems. This work was published as [VlV02].

In Chapter 4 we take a closer look at time discretization schemes for stiff systems of ODEs. We describe implicit Runge-Kutta and boundary value methods as well as iterative methods for ODEs using these methods. The standard waveform relaxation convergence analysis is extended to handle these time discretization schemes. This work was published as [VlV05b].

Chapter 5 builds on the previous chapter were it was shown that high order time discretization schemes can be used very effectively in conjunction with multigrid methods. Spectral methods are an important class of high order methods. We consider the use of Chebyshev spectral collocation as a time discretization scheme. This work was published in the proceedings paper [VlV05c].

In Chapter 6 the iterative methods and their analysis are extended to partial differential equations containing one term with a constant delay. This work was published as [VlJV05].

Chapter 7 describes a theory for the convergence analysis of iterative methods for ODEs based on functional calculus. The underlying theory is also applied to a two-grid convergence analysis for multigrid methods. This work is the subject of the technical report [VlV05a] and has been submitted for publication.

In Chapter 8 we present an overview of some implementations using multigrid methods in the context of time-dependent problems, give some conclusions and outline ideas for further research.

1.4 Main Messages

To conclude this introduction, we present some of the main messages of this thesis.

- Using the right methods and principles, iterative methods for time-independent PDEs can be transformed into methods for time-dependent PDEs (with similar convergence properties).
- The main principle is to group unknowns in the time dimension. Interpreted from the point of view of multigrid methods this means that we use block smoothers and do not coarsen in the time direction.
- Special multigrid methods for anisotropic elliptic equations can be extended to anisotropic parabolic equations.
- The convergence of iterative methods for time-dependent equations is closely related to the stability of the time discretization scheme. This was already known for linear multistep methods and implicit Runge-Kutta methods. The same principles can be extended to general linear methods and boundary value methods.

- Using high order time discretization schemes is straightforward when multigrid can be used.
- Systems of delay differential equations (DDE) derived from delay PDEs can also be handled by waveform relaxation.
- Not only the methods, but also the analysis can be extended from the time-independent to the time-dependent case. Traditionally this is done using the theory of Volterra convolution equations. It is possible to derive a framework based on functional calculus, that clarifies the connection with the time-independent case and highlights the importance of the stability of the time discretization scheme.
- Functional calculus simplifies the spectral analysis of iterative methods. It allows the iteration operators to be constructed from simpler operators whose spectrum is known.

Chapter 2

Iterative Methods for Time-Dependent Problems

2.1 Introduction

In this chapter we introduce the iterative methods that form the basis for the methods in the rest of this work. We first consider iterative methods for time-independent or stationary partial differential equation (PDE). This is useful since they form the inspiration for the methods for time-dependent PDEs. Furthermore, the convergence analysis of the iterative methods for time-dependent problems can be reduced to the analysis of related time-independent problems.

The exposition in this chapter is based on two model problems. For the time-independent case, our model problem is the Poisson equation. Discretization of this and similar elliptic PDEs leads to a large system of linear equations. These equations can be solved using iterative methods such as the Jacobi and Gauss-Seidel methods. For the time-dependent case, our model problem is the heat equation. Iterative methods for this parabolic equation can be developed based on the same principles as in the time-independent case. For both elliptic and parabolic equations, multigrid acceleration results in highly efficient methods.

Section 2.2 recalls the linear algebra needed for the time-independent case. Classical iterative methods for the time-independent case are considered in §2.3. Section 2.4 provides the functional analysis needed for the time-dependent case. Iterative methods for time-dependent equations are

considered in §2.5. Section 2.6 explains how the multigrid principle can be applied to obtain efficient iterative methods for time-independent as well as time-dependent equations. Some illustrative numerical results are given in §2.7.

2.2 Linear Algebra Preliminaries

This section introduces some linear algebra concepts and their notations. More details can be found in books on linear algebra [Hal58, Str80, HJ90, HJ94, Kel95] or functional analysis (see §2.4).

A norm of a vector $x \in \mathbb{C}^m$ is denoted by $\|x\|$. The corresponding matrix norm of a matrix $A \in \mathbb{C}^{m\times n}$ is given by

$$\|A\| = \sup_{\|x\|=1} \|Ax\|.$$

The vector $x \in \mathbb{C}^m$ is called an eigenvector with eigenvalue $\lambda \in \mathbb{C}$ of the square matrix $A \in \mathbb{C}^{m\times m}$ if

$$Ax = \lambda x.$$

The set of all eigenvalues of a matrix is called its spectrum. The spectrum of a matrix A is denoted by $\sigma(A)$. The spectral radius $\rho(A)$ of a matrix A is the largest eigenvalue in modulus, that is

$$\rho(A) = \max_{\lambda\in\sigma(A)} |\lambda|.$$

A Toeplitz band matrix is a matrix whose elements are given by

$$\begin{aligned} a_{i,j} &= a_{j-i}, \quad \text{for } -p \le j-i \le q, \\ &= 0, \qquad \text{otherwise.} \end{aligned}$$

We denote the Toeplitz band matrix

$$\begin{bmatrix} a_0 & a_1 & \cdots & a_q & 0 & \cdots \\ a_{-1} & \ddots & \ddots & \ddots & \ddots & \\ \vdots & \ddots & & & & \\ a_{-p} & \ddots & & & & \\ 0 & \ddots & & & & \\ \vdots & & & & & \end{bmatrix}$$

as

$$\begin{bmatrix} a_{-p} & \cdots & a_{-1} & \boxed{a_0} & a_1 & \cdots & a_q \end{bmatrix}.$$

The notation can also be used for Toeplitz operators (mapping semi-infinite sequences) and Laurent operators (mapping doubly-infinite sequences).

The Kronecker product $A \otimes B$ denotes the matrix defined by replacing every element a_{ij} of A by the matrix $a_{ij}B$, i.e.,

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}.$$

If $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{q \times r}$ then $A \otimes B \in \mathbb{R}^{mq \times nr}$. For more details on the Kronecker product see [Gra81], [MN88] and [HJ94, Ch.4].

2.3 Iterative Methods for Systems of Equations

First, we consider iterative methods for time-independent or stationary PDEs using the Poisson equation as model problem. A finite difference or finite element discretization of this equation leads to a large, sparse and highly structured system of linear equations. The classical iterative methods are easy to formulate and the iteration steps can be implemented very efficiently. Unfortunately, theoretical convergence analysis and numerical experiments show that the convergence of these iterative methods is slow. For systems of equation derived from elliptic equations this can be remedied by using multigrid acceleration as explained in §2.6.

2.3.1 Model Problem

The standard model problem for elliptic equations is the Poisson equation

$$u_{xx} + u_{yy} = f, \tag{2.1}$$

where u and f are functions defined on the unit square $\Omega = [0,1]^2$. The function u is the unknown solution and f is the source term. The subscripts denote the second derivatives with respect to x and y. The Poisson equation can, for example, be used to model the equilibrium temperature distribution in a heated plate with given boundary conditions. The source function f indicates where heat is added or removed.

To obtain a well-defined problem, boundary conditions have to be supplied. The standard boundary conditions are Dirichlet, Neumann, periodic or a combination of these. Dirichlet boundary conditions fix the value of the solution on the boundary and can be specified using a single function g

$$u(x,y) = g(x,y), \qquad (x,y) \in \partial\Omega,$$

or using a function for each section of the boundary. On the unit square this becomes

$$u(x,0) = g_1(x), \quad u(1,y) = g_2(y), \quad u(x,1) = g_3(x), \quad u(0,y) = g_4(y).$$

Neumann boundary conditions fix the normal derivative of the solution

$$u_n(x,y) = g(x,y), \qquad (x,y) \in \partial\Omega,$$

or, for the unit square,

$$u_y(x,0) = g_1(x), \quad u_x(1,y) = g_2(y), \quad u_y(x,1) = g_3(x), \quad u_x(0,y) = g_4(y).$$

A solution u on the unit square is periodic in both directions if

$$u(x,y) = u(x+1,y) = u(x,y+1).$$

The corresponding periodic boundary conditions for the Poisson equation are

$$\begin{aligned} u(x,0) &= u(x,1), \qquad & u_y(x,0) &= u_y(x,1), \\ u(0,y) &= u(1,y), \qquad & u_x(0,y) &= u_x(1,y). \end{aligned}$$

In what follows we always use Dirichlet boundary conditions, except in the convergence analysis where periodic boundaries are more convenient.

We discretize the Poisson equation with finite differences. Define the regular rectangular grid

$$(x_i, y_j) = (i\Delta x, j\Delta y), \quad i = 0, \ldots, n_x, \quad j = 0, \ldots, n_y,$$

where $\Delta x = n_x^{-1}$, $\Delta y = n_y^{-1}$ and n_x and n_y are the number of subintervals in each direction. The second order derivatives at the grid points are approximated by

$$\begin{aligned} u_{xx}(x_i, y_i) &\approx \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2}, \\ u_{yy}(x_i, y_i) &\approx \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\Delta y^2}, \end{aligned}$$

where $u_{i,j} \approx u(x_i, y_j)$. Using these approximations at the internal grid points results in the following system of equations

$$\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\Delta y^2} = f_{i,j},$$
$$i = 1, \ldots, n_x - 1, \quad j = 1, \ldots, n_y - 1,$$

	·	·	·	·	·	·	·	·	·
	·	42	43	44	45	46	47	48	·
	·	35	36	37	38	39	40	41	·
	·	28	29	30	31	32	33	34	·
	·	21	22	23	24	25	26	27	·
	·	14	15	16	17	18	19	20	·
	·	7	8	9	10	11	12	13	·
y/j	·	0	1	2	3	4	5	6	·
$\uparrow$	·	·	·	·	·	·	·	·	·
	$\rightarrow x/i$								

Figure 2.1: Lexicographical ordering for a rectangular grid ($n_x = n_y = 8$).

where $f_{i,j} = f(x_i, y_j)$. Together with the Dirichlet boundary conditions for $u_{0,j}$, $u_{n_x,j}$, $j = 1 \dots n_y - 1$ and $u_{i,0}$, u_{i,n_y}, $i = 1 \dots n_x - 1$ this results in a system of $m = (n_x - 1)(n_y - 1)$ unknowns and the same number of equations. This system of equations can be written as

$$Lu = f,$$

where the action of the linear operator L on the grid function u can be represented in compact form using the stencil notation

$$(Lu)_{i,j} = \begin{bmatrix} & \Delta y^{-2} & \\ \Delta x^{-2} & -2\Delta x^{-2} - 2\Delta y^{-2} & \Delta x^{-2} \\ & \Delta y^{-2} & \end{bmatrix} u_{i,j} \tag{2.2}$$

(see for example [Wes92, TOS01, WJ05]). This convenient way to describe the above equations will also be used to describe simple iterative solvers.

If the unknowns and source terms are ordered as in Figure 2.1 and placed in vectors u and $f \in \mathbb{R}^m$ the system can be written in matrix notation as

$$Lu = f. \tag{2.3}$$

The matrix $L \in \mathbb{R}^{m \times m}$ has a very regular structure. It is block tridiagonal with tridiagonal blocks on the main block diagonal and diagonal blocks on the block off-diagonals. Figure 2.2 shows the non-zero elements of L for $n_x = n_y = 10$.

The two-dimensional discretized Laplace operator can also be written as

$$L = L_x \otimes I_{n_y - 1} + I_{n_x - 1} \otimes L_y,$$

where $L_x = \Delta x^{-2} \begin{bmatrix} 1 & \boxed{-2} & 1 \end{bmatrix}$ and $L_y = \Delta y^{-2} \begin{bmatrix} 1 & \boxed{-2} & 1 \end{bmatrix}$ are the one-dimensional discretized Laplace operators given by tridiagonal Toeplitz

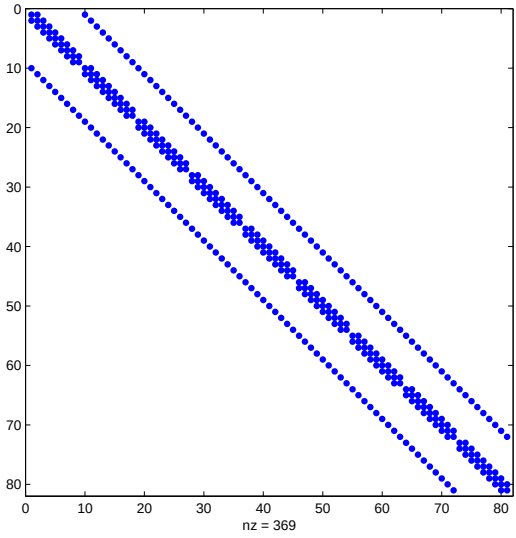


Figure 2.2: Structure of the discretized Laplace operator for $n_x = n_y = 10$.

matrices of dimensions $(n_x - 1) \times (n_x - 1)$ and $(n_y - 1) \times (n_y - 1)$. The matrices I_{n_x-1} and I_{n_y-1} are the corresponding identity matrices.

Other orderings than the lexicographical one can be useful as well. Figure 2.3 illustrates the so-called red-black ordering. The grid points are divided in two groups according to the parity of $i + j$. Assuming an even number of subintervals in each dimension ($n_x = 2\bar{n}_x$ and $n_y = 2\bar{n}_y$), the resulting matrix can be partitioned as

$$L = \begin{bmatrix} D_e & J^T \\ J & D_o \end{bmatrix}$$

where D_e and D_o are diagonal Toeplitz matrices of dimensions $m_e \times m_e$ and $m_o \times m_o$ with $m_e = (m + 1)/2$ and $m_o = m_e - 1$. The matrix J is a band matrix of dimensions $m_o \times m_e$ with 4 non-zero diagonals.

Many other types of discretization are possible such as methods using finite elements or finite volumes, continuous and discontinuous Galerkin methods, spectral methods. These methods can be written in the same matrix form as for finite differences. The convergence analysis further on, is given in a general form. If the matrices have appropriate properties the analysis can be carried over to other methods.

More general equations of the form

$$\mathcal{L}u = f,$$

·	·	·	·	·	·	·	·	·
·	21	46	22	47	23	48	24	·
·	42	18	43	19	44	20	45	·
·	14	39	15	40	16	41	17	·
·	35	11	36	12	37	13	38	·
·	7	32	8	33	9	34	10	·
·	28	4	29	5	30	6	31	·
·	0	25	1	26	2	27	3	·
·	·	·	·	·	·	·	·	·

Figure 2.3: Red-black ordering for a rectangular grid.

where $\mathcal{L}$ is a general linear elliptic operator can be handled in much the same way. Discretization will lead to linear systems with system matrices that have a structure similar to the discrete Laplace operator and possibly similar properties.

2.3.2 Classical Iterative Methods

It is well known that for systems of equations involving large sparse matrices, such as the ones derived from discretizing PDEs, direct methods are often inefficient. Many iterative methods, on the other hand, deal very well with such problems. Two classical iterative schemes are the Jacobi and the Gauss-Seidel methods. Both can be described using a splitting of the system matrix. Using $L = L^+ + L^-$ and (2.3) the iteration scheme becomes

$$L^+ u^{(\nu)} + L^- u^{(\nu-1)} = f, \tag{2.4}$$

where $u^{(\nu)}$ is the approximation after ν steps. The iteration is started from a given initial approximation $u^{(0)}$. The matrix L^+ is chosen such that this system can be easily solved. For the Jacobi method L^+ contains only the diagonal of L. For the Gauss-Seidel method L^+ is the lower triangular part of L. For each method a set of $m = (n_x - 1)(n_y - 1)$ scalar equations has to be solved repeatedly. Unlike for the Jacobi method, the ordering of the unknowns is relevant for the Gauss-Seidel method. Common orderings on two-dimensional grids are lexicographical (Figure 2.1) and red-black (Figure 2.3). For discretizations on regular grids, the Jacobi, lexicographical Gauss-Seidel and red-black Gauss-Seidel can be described compactly in stencil notation. Non-zero values in the stencils, taken from the original stencil (2.2), are indicated by "•".

Jacobi:

- for all (i,j), solve

$$\begin{bmatrix} \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} & \bullet & \\ \bullet & & \bullet \\ & \bullet & \end{bmatrix} u_{i,j}^{(\nu)} = f_{i,j}$$

Lexicographic Gauss-Seidel:

- for $j = 1, \ldots, n_y - 1$
  for $i = 1, \ldots, n_x - 1$, solve

$$\begin{bmatrix} \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} \bullet & \\ & \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} & \bullet \\ \bullet & \end{bmatrix} u_{i,j}^{(\nu)} = f_{i,j}$$

Red-Black Gauss-Seidel:

- for all (i,j) with $i+j$ even, solve

$$\begin{bmatrix} \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} & \bullet & \\ \bullet & & \bullet \\ & \bullet & \end{bmatrix} u_{i,j}^{(\nu)} = f_{i,j}$$

- for all (i,j) with $i+j$ odd, solve

$$\begin{bmatrix} \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} & \bullet & \\ \bullet & & \bullet \\ & \bullet & \end{bmatrix} u_{i,j}^{(\nu+1)} = f_{i,j}$$

A Jacobi or Gauss-Seidel iteration step can be implemented very efficiently in the case of sparse matrices. The implementation is even simpler when uniform grids are used. Unfortunately these simple splitting methods may converge very slowly. They do, however, form the basis of the very efficient multigrid method described in §2.6. In Chapter 3 methods appropriate for time-dependent parabolic equations with spatial anisotropy are described. For more variations and other iterative methods of this kind we refer to [TOS01].

2.3.3 Convergence Analysis

The convergence of splitting methods is analyzed using the matrix formulation of the iterations. Let $e^{(\nu)} = u^{(\nu)} - u$ be the error at iteration ν. If we subtract the equation $Lu = f$, or, equivalently, $L^+u + L^-u = f$, from the iteration $L^+u^{(\nu)} + L^-u^{(\nu-1)} = f$, we arrive at the error iteration

$$L^+e^{(\nu)} = -L^-e^{(\nu-1)}.$$

This can also be written as

$$e^{(\nu)} = -(L^+)^{-1}L^- e^{(\nu-1)} = Ke^{(\nu-1)},$$

where K is called the iteration matrix. It is well known that the error will converge to zero if $\rho(K) < 1$. The spectral radius $\rho(K)$ is also called the asymptotic convergence factor. In general we want ρ to be as small as possible in order to have a rapidly converging iterative method.

For the Jacobi and Gauss-Seidel methods applied to a finite difference discretization of the Poisson equation on a regular grid, the spectral radius of the iteration operator can be derived exactly. In Chapter 3 an analysis based on Fourier modes is described.

The spectral radius only tells us something about the asymptotic convergence of a method. The fact that a method will eventually converge does not always imply that it is a good method in practice. For the examples considered here, there is a close correspondence with the numerical experiments, however.

We prefer to report convergence rates instead of convergence factors. The convergence rate R is defined by

$$R = -\log_{10}(\rho) \tag{2.5}$$

and can be interpreted as the average number of additional correct digits obtained per iteration. The asymptotic convergence rate can be estimated using

$$\rho^{(\nu,\mu)} = \sqrt[\mu]{\frac{\|e^{(\nu)}\|}{\|e^{(\nu-\mu)}\|}}, \qquad R^{(\nu,\mu)} = -\log_{10}(\rho^{(\nu,\mu)}).., \tag{2.6}$$

For the numerical experiments we used, for example, the average of $R^{(\nu,1)}$ for $\nu = 10, \ldots, 20$. This corresponds to taking the geometric mean of $\rho^{(\nu,1)}$ or taking $\nu = 20$, $\mu = 10$ in (2.6). When the exact solution of the system of equations is not known the norm of the residual $f - Lu$ can be used instead of the norm of the error. This will usually give very similar results.

2.4 Functional Analysis Preliminaries

Basic notions from linear algebra were sufficient to analyze the convergence of the classical iterative methods for systems of equations considered in the previous section. The convergence analysis of iterative methods for time-dependent equations is more conveniently set in the more general abstract framework of functional analysis. Many concepts from functional analysis form a generalization of ideas from linear algebra to infinite dimensional linear spaces such as spaces of functions. This section

sets some notations and recalls the functional analysis concepts that are needed for the convergence analysis introduced in the following section. These concepts are also used for the convergence theory developed in Chapter 7. More details can be found in functional analysis textbooks such as [DS57, HP74, Rud73, Sch71, Tay58, Lax02].

Let x be an element of a complex Banach space X, then the norm of x is denoted by $\|x\|_X$ or $\|x\|$. The domain of a linear operator T is denoted by $\mathcal{D}(T)$. A linear operator T (with $\mathcal{D}(T) = X$) is *bounded* if

$$\exists M : \forall x \in X : \|Tx\| < M\|x\|.$$

The *norm* of a bounded linear operator is given by

$$\|T\| = \sup_{\|x\|=1} \|Tx\|.$$

A linear operator T is *closed* if the set of all points (x, Tx) (its graph) is closed. This is equivalent to

$$\Big(x_n \in \mathcal{D}(T) \wedge \lim_{n\to\infty} x_n = x \wedge \lim_{n\to\infty} Tx_n = y\Big) \Rightarrow x \in \mathcal{D}(T) \wedge y = Tx.$$

A closed linear operator T with $\mathcal{D}(T) = X$ is bounded. A linear operator T is *compact* if for every sequence $\{x_n\}$ with $\|x_n\| = 1$, the sequence $\{Tx_n\}$ has a convergent subsequence.

The *resolvent set* of a linear operator T in X is the set of complex numbers λ for which $(\lambda I_X - T)^{-1}$ is a bounded linear operator with a domain that is dense in X. The *spectrum* $\sigma(T)$ of T is the complement of the resolvent set in $\mathbb{C}$. The *spectral radius* of a linear operator is given by

$$\rho(T) = \sup_{\lambda\in\sigma(T)} |\lambda| = \lim_{n\to\infty} \|T^n\|^{1/n}.$$

The vector x is called an *eigenvector* with *eigenvalue* $\lambda \in \mathbb{C}$ of the operator T if

$$Tx = \lambda x.$$

The spectrum of a closed linear operator is a closed set. The spectrum of a bounded linear operator is a closed and bounded set. The spectrum of a compact linear operator is a countable set of eigenvalues with finite multiplicity.

In a general Banach space, an iteration of the form

$$x^{(\nu)} = Tx^{(\nu-1)} + b$$

converges, for any initial $x^{(0)}$, to the unique solution of $x - Tx = b$ if

$$\rho(T) < 1.$$

The spectral radius $\rho(T)$ can again be interpreted as an asymptotic convergence factor.

The Kronecker product notation $A \otimes B$ can easily be generalized when either A or B is an operator in an infinite dimensional Banach space. The case where both are operators in an infinite dimensional Banach space is not straightforward [BP66, Sch69, DS70, RS73, Ich78a, Ich78b], but we do not need this case in what follows.

The following Banach spaces of functions and sequences are used. The functions and sequences take values from a given Banach space X, typically $\mathbb{C}$ or C^m. The space $C([0,t_F],X)$ of continuous functions on an interval with norm

$$\|x\| = \max_{t\in[0,t_F]} \|x(t)\|$$

the space $L^p([0,t_F],X)$ of p-integrable functions on an interval with norm

$$\|x\|^p = \int_0^{t_F} \|x(t)\|^p dt, \quad 1 \le p < \infty,$$

the space $L^\infty([0,t_F],X)$ of essentially bounded functions on an interval with norm

$$\|x\| = \operatorname*{ess\,sup}_{t\in[0,t_F]} \|x(t)\|,$$

the space $l^p(n,X)$ of p-summable sequences of length n with norm

$$\|x\|^p = \sum_i \|x_i\|^p, \quad 1 \le p < \infty,$$
$$\|x\| = \sup_i \|x_i\|, \quad p = \infty.$$

The parameters t_F and n can be infinite for the L^p and l^p spaces.

2.5 Iterative Methods for Systems of Ordinary Differential Equations

In this section we introduce iterative methods for time-dependent equations using the heat equation as model problem. We give a brief description of time stepping and of continuous and discrete waveform relaxation. These methods are considered in more detail in Chapter 4.

2.5.1 Model Problem

The standard model problem for parabolic PDEs is the heat or isotropic diffusion equation

$$u_t = u_{xx} + u_{yy} + f, \tag{2.7}$$

where u and f are now not only functions of the spatial variables $(x, y) \in \Omega = [0, 1]^2$, but also of time $t \in [0, t_F]$. In addition to the boundary conditions, which have to be specified for all t, an initial condition $u(t, x, y) = u_0(x, y)$ or a periodicity condition $u(t, x, y) = u(t + t_F, x, y)$ has to be supplied to obtain a well-defined problem. In what follows we mainly consider initial value problems. If the period t_F is given and a good method is known for the initial value problem, the periodic case does not pose any special problems. When the boundary conditions and the source term f are independent of t, the solution will be independent of t if $u_t = 0$ and we recover the Poisson equation.

Using the same spatial discretization as for the stationary case leads to the system of ordinary differential equations (ODE)

$$\dot{u} = Lu + f, \tag{2.8}$$

where u and f are vector-valued functions of time and $\dot{u} = \frac{du}{dt}$. In general L can be a matrix-valued function of time, but for the convergence analysis it will always be assumed that L is a constant matrix. The transformation of a time-dependent PDE to a system of ODEs using a semi-discretization in space is also referred to as the method of lines (MOL).

When finite elements are used to discretize the spatial domain the resulting system of ODEs has the form

$$M\dot{u} = Lu + f,$$

where the mass matrix M is a large, sparse and structured matrix. The iterative methods and their analysis can be extended to this case (see [JV96a, JV96b] and §7.3.1).

2.5.2 Time Stepping

The system of equations (2.8) is a stiff system of ODEs, as is typical for discretized parabolic equations. Since the use of explicit methods leads to severe restrictions on the size of the time step, only implicit methods are considered. In this chapter and the next only the backward Euler method is used. Many more time discretization schemes are considered in Chapters 4 and 5. If we use the implicit Euler method to discretize (2.8) in time we obtain the following fully discrete system of equations

$$u_i = u_{i-1} + \Delta t L u_i + \Delta t f_i, \tag{2.9}$$

where u_i and f_i are $(n_x - 1) \times (n_y - 1)$ grid functions or equivalently vectors in $\mathbb{R}^m$. The approximations u_i are obtained by solving the above system for each time step i in order. The structure of the systems in each time step

is the same as for the corresponding discretized elliptic equation. We can therefore use exactly the same methods as for stationary problems. Note that for $\Delta t \to 0$ the matrix of the systems to be solved approaches the identity matrix. For $\Delta t \to \infty$ the system matrix approaches a multiple of L. We can therefore expect convergence similar to the stationary case for large Δt and better convergence for small Δt.

The method just described is equivalent to first discretizing the original PDE in time and then solving the resulting sequence of stationary PDEs as before. This is only true for linear systems with fixed grids. When nonlinearities have to be handled and adaptive grids are used, the ordering of discretization and solution becomes important.

Applying the same splitting $L = L^+ + L^-$ as in §2.3 results in the iteration

$$u_i^{(\nu)} = u_{i-1}^{(\nu)} + \Delta t L^+ u_i^{(\nu)} + \Delta t L^- u_i^{(\nu-1)} + \Delta t f_i. \tag{2.10}$$

Note that the loop over the time steps i forms the outer loop. For each time step i there is an inner loop for the iterates ν.

2.5.3 Continuous Waveform Relaxation

In the time stepping method the Jacobi and Gauss-Seidel iterative methods can be used to solve the systems (2.9). We can, however, also apply a splitting method directly to a system of ODEs. Plugging the same splitting $L = L^+ + L^-$ as in §2.3.2 into (2.8) results in the iteration

$$\dot{u}^{(\nu)} = L^+ u^{(\nu)} + L^- u^{(\nu-1)} + f, \tag{2.11}$$

where the iterates $u^{(\nu)}$ are now vectors containing functions of time instead of scalar values. This iterative method working on functions of time is called continuous waveform relaxation. The algorithms below illustrate the Jacobi, lexicographical Gauss-Seidel and red-black Gauss-Seidel methods for a discretization in space using a general five-point stencil on a regular rectangular grid. In each iteration a sequence of $m = (n_x - 1)(n_y - 1)$ scalar ODEs has to be solved.

Jacobi:

- for all (i, j), solve

$$\dot{u}_{i,j}^{(\nu+1)} = \begin{bmatrix} \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} & \bullet & \\ \bullet & & \bullet \\ & \bullet & \end{bmatrix} u_{i,j}^{(\nu)} + f_{i,j}$$

Lexicographic Gauss-Seidel:

- for $j = 1, \ldots, n_y - 1$
  for $i = 1, \ldots, n_x - 1$, solve

$$\dot{u}_{i,j}^{(\nu+1)} = \begin{bmatrix} \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} \bullet & \\ & \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} & \bullet \\ & \bullet \end{bmatrix} u_{i,j}^{(\nu)} + f_{i,j}$$

Red-Black Gauss-Seidel:

- for all (i, j) with $i + j$ even, solve

$$\dot{u}_{i,j}^{(\nu+1)} = \begin{bmatrix} \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} & \bullet & \\ \bullet & & \bullet \\ & \bullet & \end{bmatrix} u_{i,j}^{(\nu)} + f_{i,j}$$

- for all (i, j) with $i + j$ odd, solve

$$\dot{u}_{i,j}^{(\nu+1)} = \begin{bmatrix} \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} & \bullet & \\ \bullet & & \bullet \\ & \bullet & \end{bmatrix} u_{i,j}^{(\nu+1)} + f_{i,j}$$

2.5.4 Discrete Waveform Relaxation

To obtain fully discrete systems as in the time stepping case, the solutions to the scalar ODEs in continuous waveform relaxation can be approximated using an ODE integrator. Many time discretization schemes are considered in Chapters 4 and 5. For now we consider only the backward Euler method.

The time stepping method can be interpreted as a discretization in time resulting in a sequence of stationary PDEs which are then solved using known iterative methods. The waveform relaxation method discretizes in space and then applies an iterative method directly. Another way of obtaining these methods is to discretize the time-dependent PDE in space and time and to use different block iterative schemes (block Jacobi or Gauss-Seidel) to update the unknowns. For linear problems and fixed grids the system solved is the same, only the solution method is different.

Many other variations are possible. The time domain can be divided in subintervals, for example, which can then be handled sequentially. This block time stepping or time windowing techniques is discussed further in Chapter 4.

Using the implicit Euler method to discretize (2.11) results in

$$u_i^{(\nu)} = u_{i-1}^{(\nu)} + \Delta t L^+ u_i^{(\nu)} + \Delta t L^- u_i^{(\nu-1)} + \Delta t f_i. \tag{2.12}$$

Note that this is exactly the same expression as (2.10) obtained for time stepping using the implicit Euler method. In the discrete waveform re-

laxation case, however, the outer loop is over the iterations and for each iteration ν there is an inner loop over the time steps i.

2.5.5 Convergence Analysis

The convergence analysis of iterative methods used in a time stepping scheme is completely analogous to the analysis of the corresponding iterative methods for time-independent problems. We illustrate this here for the implicit Euler scheme applied to the discretized heat equation.

Waveform relaxation schemes were first introduced and analyzed in a qualitative way in the electrical engineering literature [LRSV82, WSV87, WSVOR85]. Here we follow the more quantitative convergence analysis introduced in [MN87a, MN87b]. This analysis was extended to multigrid waveform relaxation [LO87, Van93], implicit Runge-Kutta (IRK) time discretizations [Bur95, LO87], time periodic problems [Van93] and spatial discretizations using finite elements [JV96a, JV96b].

The convergence analysis of waveform relaxation methods can be based on the theory of Volterra integral equations. This essentially comes down to using a Laplace transformation with respect to time to reduce the time-dependent problem to a set of time-independent problems with a complex parameter. We give here the main results for continuous and discrete waveform relaxation, both for initial value problems on finite and infinite time intervals and for time-periodic problems. The main lemmas used to prove the results for initial value problems are stated as well. They are used in Chapter 4 to obtain convergence results for more general time discretization schemes. The results for both initial value and periodic problems are generalized in Chapter 7.

To study the convergence of iterative methods for ODEs, we analyze the asymptotic convergence factor of the iteration scheme. This convergence factor is given by the spectral radius of the iteration operator. The relation between the error $e^{(\nu)} = u^{(\nu)} - u$ of successive approximations can be written as

$$e^{(\nu+1)} = \mathcal{K} e^{(\nu)}. \tag{2.13}$$

For time stepping and discrete waveform relaxation on finite intervals the iteration operator $\mathcal{K}$ is a matrix with a structure that is essentially the same as in the time-independent case. For continuous and discrete waveform relaxation $\mathcal{K}$ is a linear Volterra convolution operator. In [MN87a, MN87b, LO87, Van93, JV96a, JV96b] it is shown that the spectral radius of the waveform relaxation operator is given by

$$\rho(\mathcal{K}) = \max_{z \in \Sigma} \rho(K(z)), \tag{2.14}$$

where $K(z)$ is the Laplace transform of the convolution kernel of the operator $\mathcal{K}$. The set Σ is a subset of the complex plane plus infinity. Its particular structure depends on the type of waveform relaxation. We recall some results from [MN87a, MN87b, LO87, Van93, JV96a, JV96b]. A more detailed discussion is given in Chapter 4. The time-independent case is recovered for $\Sigma = \{0\}$. For continuous waveform relaxation on a finite time interval, Σ consists of a single point, the point at infinity. For time stepping as well as discrete waveform relaxation on a finite time interval, using a linear multistep method (LMM) as the time discretization, Σ consists of the single point $\frac{\alpha_k}{\beta_k}\Delta t^{-1}$, with α_k and β_k parameters of the LMM. For infinite time intervals, Σ is the imaginary axis in the case of continuous time waveform relaxation and the boundary locus of the LMM scaled by Δt^{-1} for discrete time waveform relaxation. For time-periodic problems Σ consists of a discrete set of points on the imaginary axis or on the scaled boundary locus. One can learn a lot about the convergence of different types of waveform relaxation for a specific equation of the form (2.8) by visual inspection of the so-called spectral picture, a contour plot of $\rho(K(z))$. Note that for the Gauss-Seidel and Jacobi methods, $K(z)$ is the iteration operator for the corresponding method applied to

$$zu = Lu + f,$$

which is the Laplace transform with respect to time of (2.8). This means that it is straightforward to use a standard Fourier mode analysis to determine $\rho(K(z))$. An example is given in §2.6.2 for a multigrid method applied to the heat equation. More examples are given in the following chapters.

Time Stepping

The time stepping iteration (2.10) can be written in explicit form as

$$u_i^{(\nu)} = (I_m - \Delta t L^+)^{-1} \left(u_{i-1}^{(\nu)} + \Delta t L^- u_i^{(\nu-1)} + \Delta t f_i \right).$$

Define the iteration matrix as

$$\mathcal{K}_{\Delta t} = \left(\frac{1}{\Delta t} I_m - L^+ \right)^{-1} L^-.$$

The error $e^{(\nu)} = u^{(\nu)} - u$ satisfies the error iteration

$$e^{(\nu)} = \mathcal{K}_{\Delta t} e^{(\nu-1)}.$$

The asymptotic convergence of the iteration can be analyzed using the spectral radius of the iteration matrix $\mathcal{K}_{\Delta t}$

$$\rho\left(\mathcal{K}_{\Delta t}\right) = \rho\left(K\left(\frac{1}{\Delta t} \right) \right),$$

where the matrix $K(z)$ is given by

$$K(z) = (zI_m - L^+)^{-1}L^-. \tag{2.15}$$

This matrix $K(z)$, where z is in general a complex parameter, will return in the analysis of the other iterative methods considered here.

Continuous Waveform Relaxation

For linear initial value problems, a convergence analysis based on the theory of Volterra integral equations was introduced in the seminal paper [MN87a]. We follow here the exposition of [JV96a].

The solution of the system of linear ODEs

$$\dot{u} = Lu + f,$$

with initial condition $u(0) = u_0$, is given by

$$u(t) = e^{tL}u_0 + \int_0^t e^{(t-s)L}f(s)ds.$$

By applying this solution formula, the iteration (2.11) can be written in the explicit form

$$u^{(\nu)} = \mathcal{K}u^{(\nu-1)} + \varphi,$$

where the iteration operator $\mathcal{K}$ is given by the linear Volterra convolution operator

$$\mathcal{K}u(t) = (k * u)(t) = \int_0^t k(t-s)u(s)ds$$

with matrix-valued kernel

$$k(t) = e^{tL^+}L^-.$$

The error $e^{(\nu)} = u^{(\nu)} - u$ of successive approximations satisfies the error iteration

$$\dot{e}^{(\nu+1)} = L^+e^{(\nu+1)} + L^-e^{(\nu)},$$

with $e^{(\nu)}(0) = 0$. This can be written using the iteration operator as

$$e^{(\nu+1)} = \mathcal{K}e^{(\nu)}.$$

The spectral radius of $\mathcal{K}$ on finite intervals is given by the following theorem (p. 461 in [MN87a], Th. 3.1 in [JV96a]).

Theorem 2.5.1. *Consider $\mathcal{K}$ as an operator in $C([0,t_F],\mathbb{C}^m)$ or $L^p([0,t_f],\mathbb{C}^m)$ with $1 \le p \le \infty$. Then, $\mathcal{K}$ is a bounded operator and*

$$\rho(\mathcal{K}) = 0. \tag{2.16}$$

The spectral radius of $\mathcal{K}$ on infinite intervals is given by the following theorem (Th. 2.2 in [MN87a], Th. 3.4 in [JV96a]).

Theorem 2.5.2. *Consider $\mathcal{K}$ as an operator in $L^p([0,\infty],\mathbb{C}^m)$ with $1 \leq p \leq \infty$, and assume $\sigma(L^+) \cap \bar{\mathbb{C}}^+ = \phi$. Then, $\mathcal{K}$ is a bounded operator and*

$$\rho(\mathcal{K}) = \sup_{z \in \bar{\mathbb{C}}^+} \rho(K(z)). \tag{2.17}$$

For time-periodic problems the iteration is given by

$$\dot{u}^{(\nu)} = L^+ u^{(\nu)} + L^- u^{(\nu-1)} + f,$$

with boundary conditions $u(0) = u(t_F)$. This can again be written in the explicit form

$$u^{(\nu)} = \mathcal{K} u^{(\nu-1)} + \varphi,$$

where the iteration operator $\mathcal{K}$ is now a periodic convolution operator (see [VP93, Van93]). The spectral radius of the iteration operator is given be the following theorem (Th. 4.4.13 in [Van93]).

Theorem 2.5.3. *Consider $\mathcal{K}$ as an operator in $C([0,t_F],\mathbb{C}^m)$ or $L^p([0,t_F],\mathbb{C}^m)$ with $1 \leq p \leq \infty$, and assume $\sigma(L^+) \cap \Sigma = \phi$, where $\Sigma = \{2\pi i n/t_F, n \in \mathbb{Z}\}$. Then, $\mathcal{K}$ is a compact operator and*

$$\rho(\mathcal{K}) = \sup_{z \in \Sigma} \rho(K(z)). \tag{2.18}$$

From the inclusions

$$\Sigma_{\text{periodic}} \subset \partial\bar{\mathbb{C}}^+ \subset \bar{\mathbb{C}}^+ = \Sigma_{\text{infinite}}$$

and a maximum principle for $\rho(K(z))$ it follows that

$$\rho(\mathcal{K}_{\text{periodic}}) \leq \rho(\mathcal{K}_{\text{infinite}}).$$

For a large number of points the spectral radius for the periodic case converges to that of the infinite interval case. This is clearly not the case for the finite interval case.

For completeness we give the two main lemmas about the spectral radius of linear Volterra convolution operators that can be used to prove the above theorems for initial value problems on finite and infinite intervals [MN87a, Lub83, LO87, JV96a]. These lemmas are not used in what follows, but their discrete equivalents, given in §2.5.5, are used in Chapter 4. Furthermore, generalizations of these lemmas are proved in Chapter 7. Let $\mathcal{H}$ be a linear Volterra convolution operator with a matrix-valued kernel $h(t)$, i.e.,

$$\mathcal{H}x(t) = \int_0^t h(t-s)x(s)ds,$$

where $x(t) \in \mathbb{C}^s$ and $h(t) \in \mathbb{C}^{s\times s}$. The following lemma states a well known result about convolution operators acting on functions on a finite interval (see [MN87a, LO87, JV96a]).

Lemma 2.5.4. *Consider $\mathcal{H}$ as an operator in $C([0,t_F],\mathbb{C}^s)$ or $L^p([0,t_F],\mathbb{C}^s)$, with $1 \le p \le \infty$ and t_F finite. Then, $\mathcal{H}$ is bounded and*

$$\rho(\mathcal{H}) = 0. \tag{2.19}$$

The convergence factor of a convolution operator acting on functions on an infinite interval, can be related to the convergence factor of the Laplace transform of its kernel (see [MN87a, LO87, JV96a]).

Lemma 2.5.5. *Consider $\mathcal{H}$ as an operator in $L^p([0,\infty],\mathbb{C}^s)$ with $1 \le p \le \infty$, and assume $h \in L^1([0,\infty],\mathbb{C}^{s\times s})$. Then, $\mathcal{H}$ is bounded and*

$$\rho(\mathcal{H}) = \sup_{z\in\bar{\mathbb{C}}^+} \rho(H(z)) = \sup_{\xi\in\mathbb{R}} \rho(H(i\xi)), \tag{2.20}$$

where $H(z) = \int_0^\infty h(t)e^{-zt}dt$ denotes the Laplace transform of h.

This lemma can be proved using the Paley-Wiener theorem that gives a necessary and sufficient condition for the boundedness of the solution of a linear Volterra convolution equation [MN87a, Lub83, LO87, JV96a].

Discrete Waveform Relaxation

A framework for the analysis of discrete waveform relaxation was provided in [MN87b]. We follow here the exposition of [JV96b]. A detailed discussion for many different time discretization schemes is given in Chapter 4. We use here the model problem with an implicit Euler discretization in time as an illustration.

By subtracting from the iteration (2.12)

$$u_i^{(\nu)} = u_{i-1}^{(\nu)} + \Delta t L^+ u_i^{(\nu)} + \Delta t L^- u_i^{(\nu-1)} + \Delta t f_i,$$

the system of equations (2.9)

$$u_i = u_{i-1} + \Delta t L u_i + \Delta t f_i,$$

we obtain the error iteration for $e_i^{(\nu)} = u_i^{(\nu)} - u_i$

$$e_i^{(\nu)} = e_{i-1}^{(\nu)} + \Delta t L^+ e_i^{(\nu)} + \Delta t L^- e_i^{(\nu-1)}. \tag{2.21}$$

This equation defines an operator $\mathcal{K}_{\Delta t}$ that maps the sequence $e^{(\nu-1)}$ to the sequence $e^{(\nu)}$. Multiplying by w^{-i} and summing over i results in

$$e^{(\nu)}(w) = w^{-1}e^{(\nu)}(w) + \Delta t L^+ e^{(\nu)}(w) + \Delta t L^- e^{(\nu-1)}(w). \tag{2.22}$$

The $e^{(\nu)}(w)$ are the discrete Laplace transforms of the sequences $e^{(\nu)}$. This iteration can be written as

$$e^{(\nu)}(w) = K\left(\frac{1}{\Delta t}\frac{w-1}{w}\right) e^{(\nu-1)}(w),$$

where $K(z)$ as defined in (2.15). It can be shown that the iteration (2.21) converges if the iteration (2.22) converges for all w with $|w| \geq 1$. Furthermore, the spectral radius of $\mathcal{K}_{\Delta t}$ is given by

$$\rho(\mathcal{K}_{\Delta t}) = \max_{|w|=1} K\left(\frac{1}{\Delta t}\frac{w-1}{w}\right).$$

The curve in the complex plane described by $\frac{w-1}{w}$ for $|w| = 1$ is the boundary of the stability region of the implicit Euler method. The following results can be proved for linear multistep methods (see Chapters 4 and 7).

The convergence of discrete waveform relaxation on finite intervals can be analyzed using the following theorem (Th. 4.1 in [JV96b]).

Theorem 2.5.6. *Consider $\mathcal{K}$ as an operator in $l^p(n, \mathbb{C}^m)$ with $1 \leq p \leq \infty$, and assume $\sigma(L^+) \cap \Sigma = \phi$. Then, $\mathcal{K}$ is a bounded linear operator and*

$$\rho(\mathcal{K}) = \sup_{z \in \Sigma} \rho(K(z)), \tag{2.23}$$

where $\Sigma = \left\{ \frac{1}{\Delta t}\frac{\alpha_k}{\beta_k} \right\}$, with α_k and β_k parameters of the LMM (see (4.3)).

The convergence of discrete waveform relaxation on infinite intervals can be analyzed using the following theorem (Th. 3.1 in [MN87b], Th. 4.4 in [JV96b]).

Theorem 2.5.7. *Consider $\mathcal{K}$ as an operator in $l^p(\infty, \mathbb{C}^m)$ with $1 \leq p \leq \infty$, and assume $\sigma(L^+) \cap \Sigma = \phi$. Then, $\mathcal{K}$ is a bounded linear operator and*

$$\rho(\mathcal{K}) = \sup_{z \in \Sigma} \rho(K(z)), \tag{2.24}$$

where Σ is the complement of the interior of the stability region of the LMM scaled by $\frac{1}{\Delta t}$.

The convergence of discrete waveform relaxation for periodic problems can be analyzed using the following theorem (Th. 4.5.10 in [Van93]).

Theorem 2.5.8. *Consider $\mathcal{K}$ as an operator in $l^p(n, \mathbb{C}^m)$ with $1 \leq p \leq \infty$, and assume $\sigma(L^+) \cap \Sigma = \phi$. Then, $\mathcal{K}$ is a bounded linear operator and*

$$\rho(\mathcal{K}) = \sup_{z \in \Sigma} \rho(K(z)), \tag{2.25}$$

where Σ is a discrete set of points on the boundary of the stability region of the time discretization scheme scaled by $\frac{1}{\Delta t}$.

From the inclusions

$$\Sigma_{\text{finite}} \subset \Sigma_{\text{infinite}} \quad \text{and} \quad \Sigma_{\text{periodic}} \subset \partial\Sigma_{\text{infinite}}$$

and a maximum principle for $\rho(K(z))$ it follows that

$$\rho(\mathcal{K}_{\text{finite}}) \leq \rho(\mathcal{K}_{\text{infinite}}) \quad \text{and} \quad \rho(\mathcal{K}_{\text{periodic}}) \leq \rho(\mathcal{K}_{\text{infinite}}).$$

For a large number of points the spectral radius for periodic sequences converges to that for infinite sequences. This is, in general, *not* the case for the spectral radius for finite sequences. A detailed description of the sets Σ in each of the above theorems is given in Chapters 4 and 7.

As for the theorems concerning the convergence of continuous waveform relaxation, we provide the lemmas on which the previous theorems can be based. Let $\mathcal{H}_{\Delta t}$ be a discrete linear Volterra convolution operator with a matrix-valued kernel $h_{\Delta t}$, i.e.,

$$(\mathcal{H}_{\Delta t}x)_i = \sum_{j=0}^{i} h_{i-j}x_j,$$

where $x_i \in \mathbb{C}^s$ and $h_i \in \mathbb{C}^{s\times s}$. The convergence factor of a discrete convolution operator acting on finite sequences, can be related to the convergence factor of the constant term of the discrete Laplace transform of its kernel (see [MN87b, JV96b]).

Lemma 2.5.9. *Consider $\mathcal{H}_{\Delta t}$ as an operator in $l^p(n, \mathbb{C}^s)$, with $1 \leq p \leq \infty$ and n finite. Then, $\mathcal{H}_{\Delta t}$ is bounded and*

$$\rho(\mathcal{H}_{\Delta t}) = \rho(h_0) = \rho(H_{\Delta t}(\infty)), \tag{2.26}$$

where $H_{\Delta t}(w) = \sum_{i=0}^{n} h_i w^{-i}$ denotes the discrete Laplace transform of $h_{\Delta t}$.

For discrete convolution operators acting on infinite sequences we recall the following lemma (see [MN87b, LO87, JV96b]).

Lemma 2.5.10. *Consider $\mathcal{H}_{\Delta t}$ as an operator in $l^p(\infty, \mathbb{C}^s)$ with $1 \leq p \leq \infty$, and assume $h_{\Delta t} \in l^1(\infty, \mathbb{C}^{s\times s})$. Then, $\mathcal{H}_{\Delta t}$ is bounded and*

$$\rho(\mathcal{H}_{\Delta t}) = \sup_{|w|\geq 1} \rho(H_{\Delta t}(w)) = \sup_{|w|=1} \rho(H_{\Delta t}(w)), \tag{2.27}$$

where $H_{\Delta t}(w) = \sum_{i=0}^{\infty} h_i w^{-i}$ denotes the discrete Laplace transform of $h_{\Delta t}$.

These lemmas are based on a discrete equivalent of the Paley-Wiener theorem [MN87b, Lub83, LO87, JV96b]. To apply Lemma 2.5.10, the convolution kernels involved have to be in l^1. To this end we use a matrix version of Wiener's inversion theorem. For a proof we refer to [Lub83] or [MN87b].

Theorem 2.5.11 (Wiener's inversion theorem)**.** *Let $A_{\Delta t}$ be a matrix-valued sequence such that $A_{\Delta t} \in l^1(\infty, \mathbb{C}^{s\times s})$, and assume that*

$$\det \sum_{i=0}^{\infty} A_i w^{-i} \neq 0$$

for $|w| \geq 1$. Setting $\sum_{i=0}^{\infty} B_i w^{-i} = (\sum_{i=0}^{\infty} A_i w^{-i})^{-1}$, we have $B_{\Delta t} \in l^1(\infty, \mathbb{C}^{s\times s})$.

The formulae for the spectral radii of continuous and discrete waveform relaxation operators are clearly very similar. Chapter 7 suggests an approach that unifies the convergence analyses of the different waveform relaxation methods. The theory of Chapter 7 is based on functional calculus and provides information not only about the spectral radius, but about the whole spectrum of a more general class of operators. The theorems of this section will be derived as special cases.

2.6 Multigrid Acceleration

In the case of discretized elliptic equations, it is well known that simple iterations like the Jacobi and Gauss-Seidel method converge very slowly. The finer the discretization grid, the slower the convergence of these methods. The convergence factor is typically of the form $\rho = 1 - ch^2$, where h is a measure for the grid spacing and c is a positive constant. Since we often want to calculate on fine grids, these methods quickly become unworkable. Fortunately, they can be used as the basis for multigrid methods. The idea of a multigrid method is to use calculations on coarser grids to accelerate the convergence on a fine grid. Because the calculations on the coarser grids have to deal with less unknowns, the overall efficiency can be improved. A multigrid method works because of the interplay between smoothing and coarse grid correction. For discretized elliptic equations, the classical iterations, like Jacobi or Gauss-Seidel, are very good at removing the high frequency components of the error. After applying a few such iterations or smoothing steps, the remaining error can be approximated by solving a similar problem on a coarser grid. This coarse grid correction takes cares of the low frequency terms in the error. For many elliptic equations, multigrid methods with grid-independent convergence factors have been developed.

The same multigrid ideas can also be applied to parabolic equations. Here, we briefly describe a framework that fits both cases. More details are given in Chapter 3.

$\text{mg}(\mathbf{x}^{(0)}, \mathbf{A}, \mathbf{b}) \to \mathbf{x}^{(3)}$

- $\mathbf{x}^{(1)} \leftarrow \text{smooth}(\mathbf{x}^{(0)}, \mathbf{A}, \mathbf{b}, \mu_1)$
- $\bar{\mathbf{b}} \leftarrow \mathbf{R}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(1)})$
- $\bar{\mathbf{x}} \leftarrow \text{cgc}(\bar{\mathbf{A}}, \bar{\mathbf{b}})$
- $\mathbf{x}^{(2)} \leftarrow \mathbf{x}^{(1)} + \mathbf{P}\bar{\mathbf{x}}$
- $\mathbf{x}^{(3)} \leftarrow \text{smooth}(\mathbf{x}^{(2)}, \mathbf{A}, \mathbf{b}, \mu_2)$

$\text{cgc}(\bar{\mathbf{A}}, \bar{\mathbf{b}}) \to \bar{\mathbf{x}}$

- $\bar{\mathbf{x}} \leftarrow \bar{\mathbf{A}}^{-1}\bar{\mathbf{b}}$

$\text{smooth}(\mathbf{x}^{(0)}, \mathbf{A}, \mathbf{b}, \mu) \to \mathbf{x}^{(\mu)}$

- for $\nu = 1, \ldots, \mu$
  solve $\mathbf{A}^{+}\mathbf{x}^{(\nu)} = \mathbf{b} - \mathbf{A}^{-}\mathbf{x}^{(\nu-1)}$

Algorithm 2.6.1: Two-grid iteration for $\mathbf{A}\mathbf{x} = \mathbf{b}$, using the splitting $\mathbf{A} = \mathbf{A}^{+} + \mathbf{A}^{-}$.

$\text{cgc}(\bar{\mathbf{A}}, \bar{\mathbf{b}}) \to \bar{\mathbf{x}}$

- if $\bar{\mathbf{A}}$ is sufficiently small
  - $\bar{\mathbf{x}} \leftarrow \bar{\mathbf{A}}^{-1}\bar{\mathbf{b}}$
- else
  - $\bar{\mathbf{x}} \leftarrow 0$
  - γ times
    $\bar{\mathbf{x}} \leftarrow \text{mg}(\bar{\mathbf{x}}, \bar{\mathbf{A}}, \bar{\mathbf{b}})$

Algorithm 2.6.2: Coarse grid correction for multigrid iteration. V-cycle: $\gamma = 1$, W-cycle: $\gamma = 2$.

2.6.1 Standard Geometric Multigrid

Algorithm 2.6.1 gives a schematic overview of a two-grid iteration for a linear system of the form $\mathbf{A}\mathbf{x} = \mathbf{b}$. The quantities $\mathbf{x}$ and $\mathbf{b}$ are elements of a vector space approximating functions on a spatial domain. The operator $\mathbf{A}$ is typically a discrete approximation of an elliptic operator. The barred symbols indicate quantities and operators on a coarse grid. Restriction $\mathbf{R}$ and prolongation $\mathbf{P}$ are the intergrid operators, transferring quantities from the fine to the coarse level and vice versa. The whole iteration can be written as $\mathbf{x}^{(3)} = \mathbf{M}\mathbf{x}^{(0)} + \mathbf{q}$ or, equivalently, $\mathbf{e}^{(3)} = \mathbf{M}\mathbf{e}^{(0)}$ for the errors $\mathbf{e}^{(\cdot)} = \mathbf{x}^{(\cdot)} - \mathbf{x}$. It is straightforward to derive the following form for the

two-grid iteration operator

$$\begin{aligned}\mathbf{M} &= \mathbf{S}^{\mu_2}\mathbf{C}\mathbf{S}^{\mu_1},\\ \mathbf{C} &= \mathbf{I} - \mathbf{P}\bar{\mathbf{A}}^{-1}\mathbf{R}\mathbf{A},\\ \mathbf{S} &= -\mathbf{A}^{+^{-1}}\mathbf{A}^{-}.\end{aligned}$$

In the two-grid iteration the equation $\bar{\mathbf{A}}\bar{\mathbf{x}} = \bar{\mathbf{b}}$ on the coarse grid is solved exactly. We can, however, solve this equation using the same iterative algorithm. Algorithm 2.6.2 shows how the two-grid iteration can be extended to a multigrid iteration by applying this idea recursively. Discretizations on a hierarchy of increasingly coarser grids are used. The problem on the coarsest level can be solved exactly or approximately by applying a few smoothing steps.

For clarity we consider only linear problems. Non-linear problems can be solved by embedding the multigrid algorithm as a linear solver in a Newton iteration or by using a non-linear multigrid scheme [TOS01]. Here, we use the multigrid method as a solver, not as a preconditioner for a Krylov subspace method [WOW00]. Finally, we will consider only the standard geometric multigrid procedure for discretizations on regular, rectangular grids. For irregular grids and other multigrid techniques, such as the algebraic multigrid (AMG) method, we refer to [TOS01].

The standard geometric multigrid method for discretization (2.3) of the elliptic equation (2.1) fits the framework described here with

$$\begin{gathered}\mathbf{x} = u, \quad \mathbf{b} = f,\\ \mathbf{A} = L, \quad \mathbf{A}^{+} = L^{+}, \quad \mathbf{A}^{-} = L^{-},\end{gathered}$$

where $\mathbf{x}, \mathbf{b} \in \mathbb{R}^m$. For a given coarsest grid, the hierarchy of grids is built up by successively doubling the number of intervals in both the x- and the y-direction. The restriction and prolongation operators $\mathbf{R} = R$ and $\mathbf{P} = P$ are the standard full weighting restriction and bilinear interpolation operators.

2.6.2 Convergence Analysis

The convergence analysis of multigrid methods for time-dependent equations proceeds in much the same way as for the simple splitting methods. We give here some results for the two-grid case. For an analysis of the general multigrid case for time-dependent equations we refer to [JV96a, JV96b]. For an analysis of multigrid methods for time-independent equations that uses more than two grids, we refer to [WO01].

Each of the time-stepping or waveform relaxation methods described in §2.5.2, §2.5.3 and §2.5.4 can be plugged into the general multigrid framework. For continuous waveform relaxation $\mathbf{x}$ and $\mathbf{b}$ are grids of functions.

For time-stepping and discrete waveform relaxation $\mathbf{x}$ and $\mathbf{b}$ are grids of scalar values or discrete, finite or infinite sequences. By applying the continuous or discrete Laplace transform to the resulting equations, it can be seen that the Laplace transform of the corresponding multigrid iteration operator is given by

$$M(z) = S(z)^{\nu_1} C(z) S(z)^{\nu_2}.$$

The Laplace transform of the smoothing operator and coarse grid correction operator are given by

$$\begin{aligned} S(z) &= (zI - L^+)^{-1} L^-, \\ C(z) &= I - P(zI - \bar{L})^{-1} R(zI - L). \end{aligned}$$

The spectral radius of the multigrid iteration operator is related to the spectral radius of its Laplace transform in the same way as in the theorems of 2.5.5. That is, we have in all cases an expression of the form

$$\rho(\mathcal{M}) = \sup_{z \in \Sigma} \rho(M(z)).$$

The sets Σ are the same as before. For the two-grid analysis, in addition to the condition $\sigma(L^+) \cap \Sigma = \phi$, the conditions $\sigma(L) \cap \Sigma = \phi$ and $\sigma(\bar{L}) \cap \Sigma = \phi$ have to hold. Further conditions are necessary for the general multigrid case [JV96a, JV96b]. It is easily verified that these conditions hold for all the problems considered here. The necessity of the conditions on $\sigma(L^+)$ and Σ follows in a straightforward way from the theory from Chapter 7 which requires that $M(z)$ be a matrix-valued function analytic in a neighborhood of Σ.

The spectral radius of the iteration operator for a multigrid method applied to a discretized elliptic equation is typically approximated by considering the two-grid iteration. For equations with constant coefficients, discretized on regular grids, the spectral radius for the two-grid iteration can be efficiently calculated using a Fourier analysis. The same techniques can be applied to find the spectral radius $\rho(M(z))$. This two-grid Fourier analysis is described in Chapter 3. A similar procedure based on functional calculus, is given in Chapter 7.

Figure 2.4 shows contour lines of the convergence rate $R(M(z)) = -\log_{10}(\rho(M(z)))$. The spectral radius $\rho(M(z))$ is calculated using a Fourier mode analysis of a two-grid iteration with 1 pre- and 1 postsmoothing red-black Gauss-Seidel step, full weighting restriction and bilinear interpolation. The value at the origin gives an estimate for the convergence rate for the multigrid method applied to the Poisson equation. Minimization along the imaginary axis gives an estimate for the convergence rate for continuous

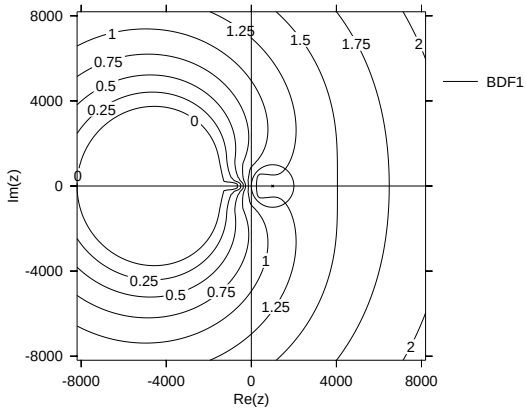


Figure 2.4: Contour lines of the convergence rate $R(M(z))$ based on a two-grid Fourier mode analysis. The boundary of the scaled stability domain for the implicit Euler method with $\Delta t = 1/1024$ is also shown (dashed line).

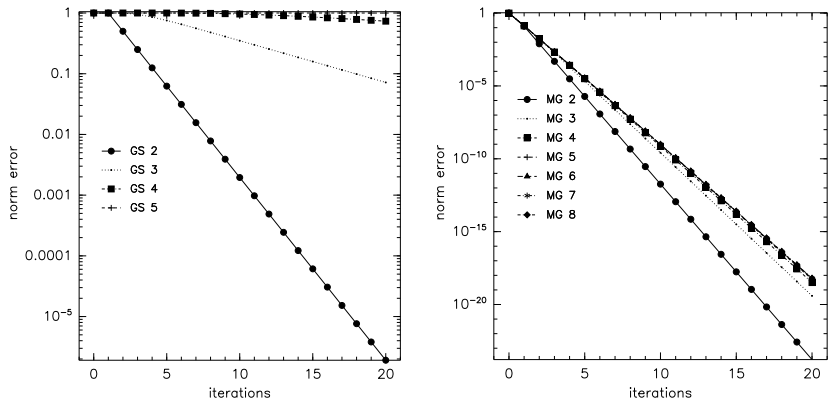
multigrid waveform relaxation applied to the heat equation. The figure also shows the boundary of the scaled stability domain of the implicit Euler method with $\Delta t = 1/1024$. Taking the minimum over this curve results in an estimate for the convergence rate for discrete multigrid waveform relaxation applied to the heat equation.

2.7 Numerical Results

To set the scene for the rest of this work, we report some numerical results for the Poisson and the heat equation. These results indicate the kind of efficiency to expect from multigrid methods. These results are a special case of the results in the next chapters.

2.7.1 Poisson Equation

We consider the Poisson equation (2.1) on the unit square, discretized using regular square grids. The boundary conditions and the source term are chosen so that the exact solution is $u = 0$. The initial approximation for the unknown internal points of the grid is set to 1. Figure 2.5 shows, for grids with $n_x = n_y = 2^{l+1}$, $l = 2, \ldots, 5$, the convergence of the maximum norm of the error for the red-black Gauss-Seidel method. It is clear that the convergence is quite slow, especially for the finer grids. Figure 2.5 also shows the convergence of of a multigrid V-cycle using the same red-black Gauss-



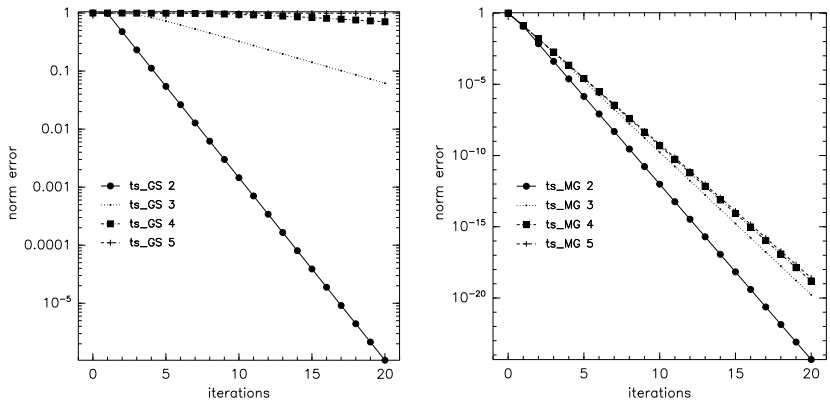
(a) Gauss-Seidel (b) Multigrid

Figure 2.5: Reduction of the error for a Gauss-Seidel and a multigrid method applied to the discretized Poisson equation for $n_x = n_y = 2^{l+1}$, $l = 2, 3, \ldots$.

$R(l)$	1	3	4	5	6	7	8
GS	0.30	0.07	0.01	0.00	0.00	0.00	0.00
MG	1.20	0.98	0.93	0.92	0.92	0.92	0.92

Table 2.1: Convergence rates of the Gauss-Seidel and multigrid methods for the Poisson equation on grids with $n_x = n_y = 2^{l+1}$.

Seidel method to perform 1 pre- and 1 postsmoothing step. The fine grids are the same as in the experiment using only Gauss-Seidel. For each fine grid, the hierarchy of grids contains all the coarser grids down to the coarsest grid with $n_x = n_y = 2$. Full weighting restriction and bilinear interpolation are used to transfer between the grids. The coarsest grid contains only one internal grid point, so that one smoothing step suffices to solve the system there. Table 2.1 shows that the convergence rates for the multigrid method are essentially independent of the grid spacing. The convergence rate gives an indication of the average number of extra digits gained per iteration. It is estimated as explained in §2.3.3. In this experiment the multigrid method increases the accuracy by almost one digit per iteration. It should be noted that the performance of the multigrid method deteriorates for stretched grids where $n_x \neq n_y$. Discretizations using stretched grids are related to the anisotropic problems considered in Chapter 3 and can be handled in the same way.



(a) Gauss-Seidel (b) Multigrid

Figure 2.6: Reduction of the error for a Gauss-Seidel and a multigrid method applied to the discretized heat equation for $n_x = n_y = 2^{l+1}$, $l = 2, \ldots, 5$, $n_t = 1$, $\Delta t = 1$.

2.7.2 Heat Equation

Consider the heat equation (2.7) discretized on the same spatial grids as in §2.7.1. We use the implicit Euler method for the discretization of time. The boundary conditions and the initial condition are chosen such that the exact solution is $u = 0$. All the unknowns are initially 1.

Time Stepping

Figure 2.6 compares the convergence of the maximum norm of the error for a Gauss-Seidel and a multigrid method applied to the system for one time step ($n_t = 1$) with $\Delta t = 1$. The same red-black Gauss-Seidel and multigrid methods as in §2.7.1 are used. The convergence rates for the time stepping case ('ts') are given in Table 2.2. The convergence rates for the multigrid method are again independent of the spacing of the spatial grid. For the spatial discretizations considered here, $\Delta t = 1$ is large and the behavior of the methods approaches the situation for the Poisson equation. For $\Delta t \to 0$ the system to be solved essentially becomes the identity matrix. In such cases, simple iterative methods like the Gauss-Seidel method can be effective. Table 2.3 illustrates this for $\Delta t = 1/1024$. This phenomenon is related to the fact that for $\Delta t \leq \Delta x^2$ an explicit time stepping method can be used.

$R(l)$	2	3	4	5
ts GS	0.31	0.07	0.01	0.00
ts MG	1.23	1.00	0.95	0.94
wr GS	0.30	0.07	0.01	0.00
wr MG	1.20	0.98	0.93	0.92

Table 2.2: Convergence rates of the Gauss-Seidel and multigrid methods for the heat equation on grids with $n_x = n_y = 2^{l+1}$, $n_t = 1024$, $\Delta t = 1$.

$R(l)$	2	3	4	5
ts GS	2.76	1.47	0.61	0.19
ts MG	4.77	2.69	1.56	1.18
wr GS	0.30	0.07	0.01	0.00
wr MG	1.63	0.98	0.93	0.92

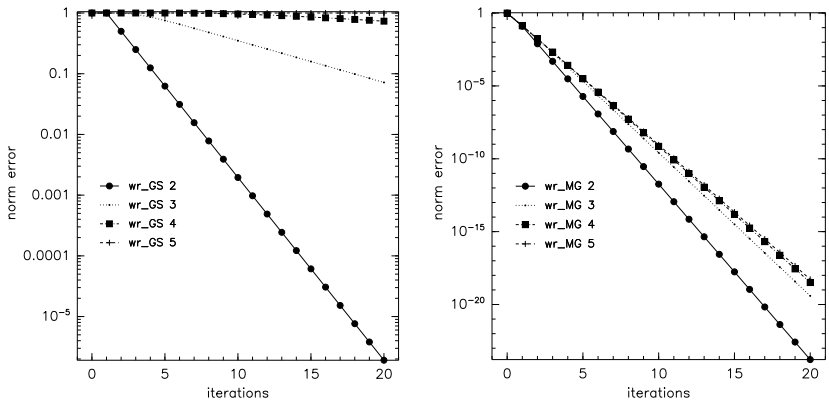
Table 2.3: Convergence rates of the Gauss-Seidel and multigrid methods for the heat equation on grids with $n_x = n_y = 2^{l+1}$, $n_t = 1024$, $\Delta t = 1/1024$.

Waveform Relaxation

We now take $n_t = 1024$, $\Delta t = 1$ and solve the resulting system using discrete waveform relaxation. Figure 2.7 compares the convergence of the maximum norm of the error for red-black Gauss-Seidel and multigrid waveform relaxation variants of the methods used in the previous examples. The convergence rates for waveform relaxation ('wr') are given in Table 2.2. The results for $\Delta t = 1/1024$ are given in Table 2.3. We see that the results for the large time step are analogous to the Poisson equation example. The results for the small step are almost the same, which shows that in the waveform relaxation case the multigrid acceleration is necessary even for small time steps.

2.8 Conclusions

In this chapter we introduced multigrid methods for time-dependent equations. The classical iterative schemes for time-independent problems were described using the Poisson equation as an example. It was then shown how these methods can be extended to time-dependent problems. A multigrid framework useful for both cases was described. We recalled the known results concerning the convergence analysis of all the presented methods. The numerical results of the last section illustrate the kind of efficiency we are aiming for.



(a) Gauss-Seidel (b) Multigrid

Figure 2.7: Reduction of the error for a Gauss-Seidel and a multigrid waveform relaxation applied to the discretized heat equation for $n_x = n_y = 2^{l+1}$, $l = 2, \ldots, 5$, $n_t = 1024$, $\Delta t = 1$.

We consider two-dimensional linear initial value problems on rectangular domains with Dirichlet boundary conditions. Uniform grids and finite differences are used for the spatial discretization. Issues such as other boundary conditions, irregular domains, unstructured grids and non-linearity can be dealt with, but they are not the subject of this work. The same is true for the use of different spatial discretization methods, as well as adaptive methods in both space and time.

The problems in this chapter all had constant coefficients. In Chapter 3 problems with coefficients that depend on space and time are considered. The implicit Euler method was used for the discretization of time. Chapters 4 and 5 study more sophisticated time discretization schemes. In Chapter 6 the class of problems under consideration is extended from PDEs to delay PDEs. In Chapter 7 the theorems mentioned here, are rederived and cast into a more general framework.

Chapter 3

Anisotropic Problems

In this chapter we consider anisotropic problems and extend multigrid methods developed for the stationary elliptic case to multigrid waveform relaxation methods for the time-dependent parabolic case. We study line-relaxation, semicoarsening and multiple semicoarsening multigrid methods. A two-grid Fourier-Laplace analysis is used to estimate the convergence of these methods for the anisotropic and rotated anisotropic diffusion equation. We treat both continuous time and discrete time algorithms. The results of the analysis are confirmed by numerical experiments.

3.1 Introduction

From the previous chapter we know that the classical model problem, that is, the isotropic, constant coefficient heat equation, can be solved efficiently by multigrid acceleration of the waveform relaxation method. In the stationary case, standard multigrid methods break down when applied to anisotropic problems, i.e., problems for which the rate of diffusion depends on the direction. Figure 3.1 shows the convergence of the maximum norm of the error for a standard multigrid method applied to the stationary diffusion equation

$$-\varepsilon\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f.$$

The equation is discretized using finite differences on regular grids. A hierarchy of 5 grids is used, where the coarsest grid has one internal point. Each iteration consists of a multigrid V-cycle with 1 pre- and 1 postsmoothing step of red-black Gauss-Seidel type, full weighting restriction and bilinear interpolation. The boundary conditions and the source term f are chosen such that $u = 0$. The initial approximation is set to 1. The convergence

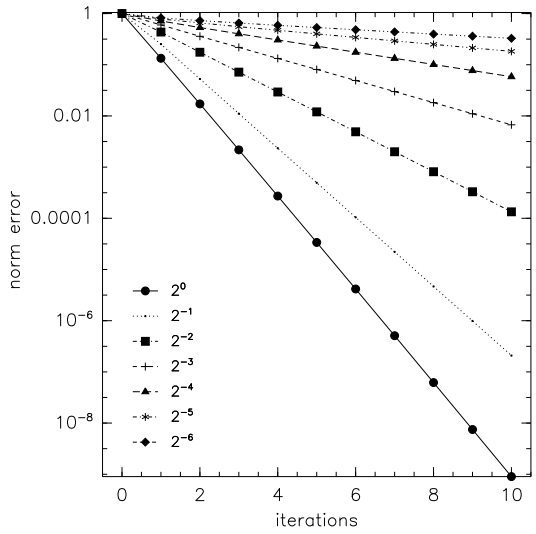


Figure 3.1: Convergence of standard multigrid for stationary anisotropic diffusion equation for various values of ε.

is very good for the isotropic case ($\varepsilon = 1$), but quickly deteriorates for anisotropic problems.

We show here that the same is true for standard multigrid waveform relaxation methods when applied to the corresponding parabolic problems. Several optimized methods have been proposed to handle stationary anisotropic problems (see [BHM00, TOS01]). In this chapter we study the extension of these methods to time-dependent parabolic problems. We use the time-dependent anisotropic and rotated anisotropic diffusion equation as test equations.

This chapter is organized as follows. We first describe the model problems and their finite difference discretization on regular rectangular grids in §3.2. Next we introduce several multigrid components in §3.3. Line relaxation, semicoarsening and multiple semicoarsening are proposed as methods for anisotropic problems. Section 3.4 explains how the convergence analysis proceeds. It is shown how the two-grid local mode Fourier analysis used for stationary multigrid, is extended to multigrid waveform relaxation. The results of the analysis are compared to numerical results in §3.5. This section also contains numerical results for a more general anisotropic equation. The methods are described as continuous waveform relaxation methods and the implicit Euler method is used as time discretization scheme. Time discretization is discussed in more detail in Chapter 4.

3.2 Model Problems

3.2.1 Continuous Equations

We consider time-dependent parabolic PDEs of the form

$$\frac{\partial u}{\partial t} = \mathcal{L}u + f, \tag{3.1}$$

with $\mathcal{L}$ an elliptic operator and with $u(t,x,y)$, $f(t,x,y) \in \mathbb{R}$ functions of time $t \in \Omega_t$ and spatial coordinates $(x,y) \in \Omega$. The time interval can be bounded, $\Omega_t = [0,T]$ or unbounded, $\Omega_t = [0,\infty)$. As in the previous chapter, we only consider initial value problems on the unit square $\Omega = [0,1]^2$ with Dirichlet boundaries.

The test equations used in this chapter are the isotropic diffusion equation, the anisotropic diffusion equation and the rotated anisotropic diffusion equation. These equation are standard test cases for the study of iterative methods [Wes92]. The isotropic diffusion equation or heat equation was already discussed in Chapter 2. It is of the form (3.1) with as operator the Laplacian

$$\mathcal{L} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}. \tag{3.2}$$

The anisotropic diffusion equation has a parameter ε indicating the strength of the diffusion in the x-direction. Its operator is given by

$$\mathcal{L} = \varepsilon\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}. \tag{3.3}$$

The isotropic diffusion equation is the special case $\varepsilon = 1$. The rotated anisotropic diffusion equation has two parameters ε and β, and is of the form (3.1) with

$$\mathcal{L} = (\varepsilon c^2 + s^2)\frac{\partial^2}{\partial x^2} + (c^2 + \varepsilon s^2)\frac{\partial^2}{\partial y^2} + 2(\varepsilon - 1)cs\frac{\partial^2}{\partial x \partial y}, \tag{3.4}$$

where $c = \cos\beta$ and $s = \sin\beta$. The angle β indicates the direction of the anisotropy and ε its strength. The rotated anisotropic diffusion equation can be obtained from the anisotropic diffusion equation by rotating the coordinate axes over an angle β.

Using the same finite differences approximation as in §2.3.1 results in a linear system of ODEs of the form

$$\dot{u} = Lu + f. \tag{3.5}$$

3.2.2 Discrete Equations in Stencil Notation

In stencil notation, the discretized differential operator for the isotropic diffusion equation becomes (see (2.2))

$$L = \begin{bmatrix} & \frac{1}{\Delta y^2} & \\ \frac{1}{\Delta x^2} & -\frac{2}{\Delta x^2} - \frac{2}{\Delta y^2} & \frac{1}{\Delta x^2} \\ & \frac{1}{\Delta y^2} & \end{bmatrix}. \tag{3.6}$$

The discretized operator for the anisotropic diffusion equation is

$$L = \begin{bmatrix} & \frac{1}{\Delta y^2} & \\ \frac{\varepsilon}{\Delta x^2} & -\frac{2\varepsilon}{\Delta x^2} - \frac{2}{\Delta y^2} & \frac{\varepsilon}{\Delta x^2} \\ & \frac{c^2}{\Delta y^2} & \end{bmatrix}. \tag{3.7}$$

The discretized differential operator for the rotated anisotropic diffusion equation becomes

$$L = \begin{bmatrix} -\frac{(\varepsilon-1)cs}{2\Delta x\Delta y} & \frac{c^2+\varepsilon s^2}{\Delta y^2} & \frac{(\varepsilon-1)cs}{2\Delta x\Delta y} \\ \frac{\varepsilon c^2+s^2}{\Delta x^2} & -2\frac{\varepsilon c^2+s^2}{\Delta x^2} - 2\frac{c^2+\varepsilon s^2}{\Delta y^2} & \frac{\varepsilon c^2+s^2}{\Delta x^2} \\ \frac{(\varepsilon-1)cs}{2\Delta x\Delta y} & \frac{c^2+\varepsilon s^2}{\Delta y^2} & -\frac{(\varepsilon-1)cs}{2\Delta x\Delta y} \end{bmatrix}. \tag{3.8}$$

In addition to the finite difference approximations of u_{xx} and u_{yy} from §2.3.1, second order finite differences spanning two mesh intervals are used to discretize the cross derivative term u_{xy} in (3.4) as

$$u_{xy}(x_i, y_i) \approx \frac{u_{i+1,j+1} - u_{i-1,j+1} - u_{i+1,j-1} + u_{i-1,j-1}}{4\Delta x\Delta y}.$$

3.2.3 Discrete Equations in Matrix Notation

When written as matrices the discretized operators of the isotropic and anisotropic diffusion equation have the same non-zero structure (see Figure 2.2). The 9-point stencil (3.8) for the discretized rotated anisotropic diffusion operator together with the lexicographical ordering of unknowns described in §2.3.1 still results in a block tridiagonal matrix with tridiagonal blocks. The off-diagonal blocks in the block tridiagonal structure are no longer diagonal, but tridiagonal. Using the Kronecker product notation the discretized operators can written in the following form. For the isotropic diffusion operator we have

$$L = L_x \otimes I_y + I_x \otimes L_y,$$

where $L_x = \Delta x^{-2} \begin{bmatrix} 1 & \boxed{-2} & 1 \end{bmatrix}$ and $L_y = \Delta y^{-2} \begin{bmatrix} 1 & \boxed{-2} & 1 \end{bmatrix}$ are the one-dimensional discretized Laplace operators given by tridiagonal Toeplitz

matrices (see §2.3.1). The discretized anisotropic diffusion operator becomes

$$L = \varepsilon L_x \otimes I_y + I_x \otimes L_y.$$

The discretized rotated anisotropic diffusion operator can be written as

$$L = (\varepsilon c^2 + s^2) L_x \otimes I_y + (c^2 + \varepsilon s^2) I_x \otimes L_y + 2(\varepsilon - 1) cs C_x \otimes C_y,$$

where the central differences are represented by the Toeplitz matrices

$$C_x = (2\Delta x)^{-1} \begin{bmatrix} -1 & \boxed{0} & 1 \end{bmatrix}$$

and

$$C_x = (2\Delta y)^{-1} \begin{bmatrix} -1 & \boxed{0} & 1 \end{bmatrix}.$$

3.3 Multigrid for Anisotropic Problems

The multigrid algorithm described in §2.6 is based on the interplay between smoothing and coarse grid correction. The coarse grid correction is determined by the hierarchy of grids and the restriction and prolongation operators used to transfer between grids at different levels. The appropriate choices for the smoother, the hierarchy of grids and the intergrid operators depend on the properties of the PDE under consideration. In this section we describe some standard multigrid components in the context of multigrid for time-dependent equations.

For the *isotropic* diffusion equation, point relaxation (e.g., Jacobi, Gauss-Seidel) can be used for smoothing. Standard coarsening can be used to construct the hierarchy of coarser grids. As will be shown by the analysis and numerical results, multigrid waveform relaxation with standard coarsening and a point relaxation method as smoother breaks down for *anisotropic* problems. To deal with anisotropic problems we can enhance the smoother and use, for example, line relaxation, which updates several grid points simultaneously. Another approach is to keep the point relaxation and to use semicoarsening to construct the hierarchy of grids. Semicoarsening methods coarsen the grids in the direction of strong diffusion, that is, the direction in which point relaxation smooths. Line relaxation and semicoarsening can be combined. Experience with the elliptic case suggests the use of the multigrid components given below. Their effectiveness for solving parabolic problems will be illustrated in further sections of this chapter. For more multigrid variants we refer to [TOS01, Wes92]. All the methods are described as working in two-dimensional grids of continuous functions. The extension to grids of sequences representing discretized approximations is straightforward.

3.3.1 Point Relaxation

We recall that continuous waveform relaxation for the system of ODEs

$$\dot{u} = Lu + f,$$

is given by the iteration

$$\dot{u}^{(\nu+1)} = L^+ u^{(\nu+1)} + L^- u^{(\nu)} + f. \tag{3.9}$$

The splitting $L = L^+ + L^-$ is selected in such a way that (3.9) is easier to solve than the original system, and the successive approximations $u^{(\nu)}$ converge to the solution of (3.5). The waveform relaxation variants of some well known iterative schemes are illustrated below for a general nine-point stencil discretization. Note that the red-black Gauss-Seidel method is only defined for five-point stencils. Non-zero values in the stencils (taken from the original stencil) are indicated by "•". For each method a set of $(n_x-1)(n_y-1)$ scalar ODEs of the form $\dot{y} = py + q$ has to be solved repeatedly. Discrete variants are obtained by using a time integration method (see Chapters 2 and 4).

Jacobi:

- for all (i,j), solve

$$\dot{u}_{i,j}^{(\nu+1)} = \begin{bmatrix} \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix} u_{i,j}^{(\nu)} + f_{i,j}$$

Lexicographic Gauss-Seidel:

- for $j = 1, \ldots, n_y - 1$
  for $i = 1, \ldots, n_x - 1$, solve

$$\dot{u}_{i,j}^{(\nu+1)} = \begin{bmatrix} \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} \bullet & & \\ \bullet & \bullet & \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} \bullet & \bullet & \bullet \\ & & \bullet \end{bmatrix} u_{i,j}^{(\nu)} + f_{i,j}$$

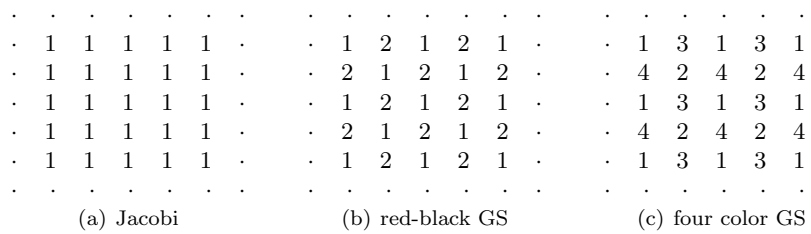
Four-Color Gauss-Seidel:

- for all (i,j) with i odd, j odd, solve

$$\dot{u}_{i,j}^{(\nu+1)} = \begin{bmatrix} \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix} u_{i,j}^{(\nu)} + f_{i,j}$$

- for all (i,j) with i even, j even, solve

$$\dot{u}_{i,j}^{(\nu+1)} = \begin{bmatrix} \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} \bullet & & \bullet \\ & & \\ \bullet & & \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} & \bullet & \\ \bullet & & \bullet \\ & \bullet & \end{bmatrix} u_{i,j}^{(\nu)} + f_{i,j}$$



(a) Jacobi (b) red-black GS (c) four color GS

Figure 3.2: Grid points that can be updated simultaneously for Jacobi, red-black and four color Gauss-Seidel relaxation.

- for all (i, j) with i even, j odd, solve

$$\dot{u}_{i,j}^{(\nu+1)} = \begin{bmatrix} \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} & \bullet & \\ \bullet & & \bullet \\ & \bullet & \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} \bullet & & \bullet \\ & & \\ \bullet & & \bullet \end{bmatrix} u_{i,j}^{(\nu)} + f_{i,j}$$

- for all (i, j) with i odd, j even, solve

$$\dot{u}_{i,j}^{(\nu+1)} = \begin{bmatrix} \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + f_{i,j}$$

In the Jacobi iteration all ODEs can be solved independently. For Gauss-Seidel methods the order in which the unknowns are updated is important. This results in many variants. For red-black and four-color Gauss-Seidel the order of the sub-steps is important, but in each sub-step the unknowns can be updated independently. Figures 3.2 illustrates which grid points can be updated simultaneously for Jacobi, Gauss-Seidel and four-color relaxation (see also Figures 2.1 and 2.3). Note that for the lexicographic Gauss-Seidel method applied to a nine-point stencil each grid points has to be considered sequentially. For a five-point stencil points on diagonals could be updated simultaneously. Four color Gauss-Seidel relaxation applied to a five-point stencil is equivalent to red-black Gauss-Seidel relaxation.

3.3.2 Line Relaxation

Instead of considering subsystems of only one equation, one can also take the set of ODEs on a line of the grid as the subsystems. The horizontal line Gauss-Seidel method corresponds to the following series of computations.

Horizontal Line Gauss-Seidel

- for $j = 1, \ldots, n_y - 1$, solve

$$\dot{u}_{i,j}^{(\nu+1)} = \begin{bmatrix} \bullet & \bullet & \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} & & \\ \bullet & \bullet & \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} \bullet & \bullet & \bullet \\ & & \end{bmatrix} u_{i,j}^{(\nu)} + f_{i,j}, \qquad 0 < i < n_x$$

The horizontal zebra Gauss-Seidel method corresponds to the following series of computations.

Horizontal Zebra Gauss-Seidel

- for all j odd, solve

$$\dot{u}_{i,j}^{(\nu+1)} = \begin{bmatrix} \bullet & \bullet & \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} \bullet & \bullet & \bullet \\ & & \\ \bullet & \bullet & \bullet \end{bmatrix} u_{i,j}^{(\nu)} + f_{i,j}, \quad 0 < i < n_x$$

- for all j even, solve

$$\dot{u}_{i,j}^{(\nu+1)} = \begin{bmatrix} \bullet & \bullet & \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + \begin{bmatrix} \bullet & \bullet & \bullet \\ & & \\ \bullet & \bullet & \bullet \end{bmatrix} u_{i,j}^{(\nu+1)} + f_{i,j}, \quad 0 < i < n_x$$

In these cases the equations describe a set of $n_y - 1$ systems of ODEs of the form $\dot{y} = Py + q$ with P tridiagonal. Similarly, one can define the vertical line and zebra Gauss-Seidel method. The so-called alternating zebra Gauss-Seidel method consists of one horizontal zebra step, followed by one vertical zebra step. Line relaxation methods will turn out to be useful to handle anisotropic problems. Figure 3.3 illustrates which points are updated simultaneously.

Horizontal line relaxation (e.g., horizontal zebra Gauss-Seidel) is expected to do well when there is strong coupling along the x-direction (e.g., $\varepsilon \gg 1$, $\beta = 0$). Conversely vertical line relaxation can be expected to work well for strong coupling along the y-direction (e.g., $\varepsilon \ll 1$, $\beta = 0$). Alternating line relaxation should work in both cases.

3.3.3 Standard Coarsening

For standard coarsening the grid spacing is doubled in both directions when going to a coarser grid, i.e., $(\overline{\Delta x}, \overline{\Delta y}) = (2\Delta x, 2\Delta y)$. Figure 3.4 shows a set of grids constructed from a fine grid by doubling the grid spacing in both directions.

·	·	·	·	·	·	·
·	5	5	5	5	5	·
·	4	4	4	4	4	·
·	3	3	3	3	3	·
·	2	2	2	2	2	·
·	1	1	1	1	1	·
·	·	·	·	·	·	·

(a) horizontal line

·	·	·	·	·	·	·
·	1	1	1	1	1	·
·	2	2	2	2	2	·
·	1	1	1	1	1	·
·	2	2	2	2	2	·
·	1	1	1	1	1	·
·	·	·	·	·	·	·

(b) horizontal zebra

Figure 3.3: Grid points that can be updated simultaneously for horizontal line and zebra relaxation.

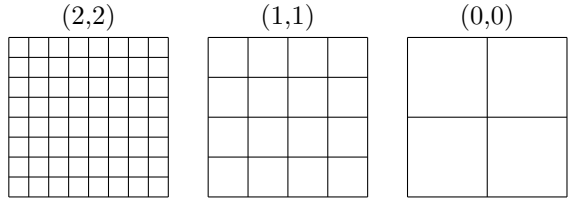
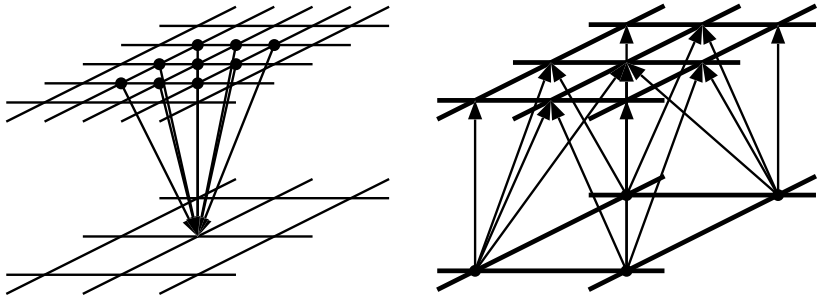


Figure 3.4: A standard coarsening grid hierarchy for multigrid methods solving a PDE, discretized on grid $(2, 2)$.



(a) Full weighting restriction (b) Bilinear interpolation

Figure 3.5: Schematic representation of the intergrid operators.

A grid point $(\bar{\imath}, \bar{\jmath})$ on the coarse grid corresponds to a point $(i, j) = (2\bar{\imath}, 2\bar{\jmath})$ on the fine grid. A standard coarsening strategy is often combined with the so-called full weighting restriction operator defined by

$$\begin{aligned}\overline{u}_{\bar{\imath},\bar{\jmath}} &= (Ru)_{\bar{\imath},\bar{\jmath}} \\ &= \frac{1}{4}u_{i,j} + \frac{1}{8}\big(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}\big) \\ &\qquad + \frac{1}{16}\big(u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j-1} + u_{i-1,j+1}\big).\end{aligned}$$

This can be written in stencil notation as

$$\overline{u}_{\bar{\imath},\bar{\jmath}} = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} u_{i,j}.$$

Figure 3.5 shows which values on the fine grid contribute to the restricted value on the coarse grid. Full weighting restriction is often combined with bilinear interpolation for the prolongation. The bilinear interpolation operator is given by

$$u_{i,j} = (P\overline{u})_{i,j} = \begin{cases} \overline{u}_{\bar{\imath},\bar{\jmath}} & \text{if } i \text{ and } j \text{ even} \\ (\overline{u}_{\bar{\imath}-1,\bar{\jmath}} + \overline{u}_{\bar{\imath}+1,\bar{\jmath}})/2 & \text{if } i \text{ odd and } j \text{ even} \\ (\overline{u}_{\bar{\imath},\bar{\jmath}-1} + \overline{u}_{\bar{\imath},\bar{\jmath}+1})/2 & \text{if } i \text{ even and } j \text{ odd} \\ (\overline{u}_{\bar{\imath}-1,\bar{\jmath}-1} + \overline{u}_{\bar{\imath}-1,\bar{\jmath}+1} + \overline{u}_{\bar{\imath}+1,\bar{\jmath}-1} + \overline{u}_{\bar{\imath}+1,\bar{\jmath}+1})/4 & \\ & \text{if } i \text{ and } j \text{ odd} \end{cases}$$

Figure 3.5 shows which values on the coarse grid are used to calculate the interpolated values on the fine grid. When the full weighting restriction operator and the bilinear interpolation operator are written as matrices R and P they are related by

$$P = cR^T, \tag{3.10}$$

for some constant c. It is very common to choose the restriction and prolongation operator such that this relation holds. All the combinations considered here are of this type.

3.3.4 Semicoarsening and Multiple Semicoarsening

Instead of using line relaxation, one can also adapt the coarsening strategy to the anisotropic features of the problem. Figure 3.6 shows a set of grids constructed from a fine grid by doubling the grid spacing only in x, only in y or in both directions. The grids used for standard coarsening are on the diagonal of Figure 3.6. When there is strong coupling in the x-direction, one can apply a *semicoarsening* strategy which means that the grid is coarsened only in one direction, i.e., $(\overline{\Delta x}, \overline{\Delta y}) = (2\Delta x, \Delta y)$. This corresponds to using the grids on the first row of Figure 3.6. A grid point $(\bar{\imath}, \bar{\jmath})$ on the coarse grid corresponds to a grid point $(i, j) = (2\bar{\imath}, \bar{\jmath})$ on the fine grid. Similarly, one can do semicoarsening in the y-direction, i.e., $(\overline{\Delta x}, \overline{\Delta y}) = (\Delta x, 2\Delta y)$. The full weighting restriction operators become

$$\begin{aligned}
\overline{u}_{\bar{\imath},\bar{\jmath}} &= (R_x u)_{\bar{\imath},\bar{\jmath}} = \frac{1}{2}u_{i,j} + \frac{1}{4}\big(u_{i+1,j} + u_{i-1,j}\big), \\
\overline{u}_{\bar{\imath},\bar{\jmath}} &= (R_y u)_{\bar{\imath},\bar{\jmath}} = \frac{1}{2}u_{i,j} + \frac{1}{4}\big(u_{i,j+1} + u_{i,j-1}\big).
\end{aligned}$$

In stencil notation

$$\overline{u}_{\bar{\imath},\bar{\jmath}} = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} u_{i,j}, \qquad \overline{u}_{\bar{\imath},\bar{\jmath}} = \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} u_{i,j}.$$

The standard prolongation for semicoarsening is linear interpolation

$$\begin{aligned}
u_{i,j} = (P_x\overline{u})_{i,j} &= \begin{cases} \overline{u}_{\bar{\imath},\bar{\jmath}} & \text{if } i \text{ even}, \\ (\overline{u}_{\bar{\imath}-1,\bar{\jmath}} + \overline{u}_{\bar{\imath}+1,\bar{\jmath}})/2 & \text{if } i \text{ odd}, \end{cases} \\
u_{i,j} = (P_y\overline{u})_{i,j} &= \begin{cases} \overline{u}_{\bar{\imath},\bar{\jmath}} & \text{if } j \text{ even}, \\ (\overline{u}_{\bar{\imath},\bar{\jmath}+1} + \overline{u}_{\bar{\imath},\bar{\jmath}-1})/2 & \text{if } j \text{ odd}. \end{cases}
\end{aligned}$$

Multiple semicoarsening methods [Mul89, OW95] use all the grids in Figure 3.6. We consider here the multigrid as smoother (MGS) (multigrid as smoother) method introduced in [Oos95] and further studied in [WO98].

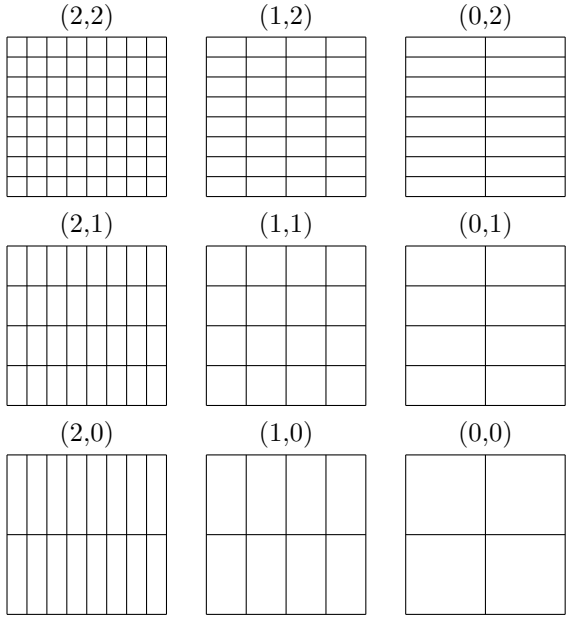


Figure 3.6: A hierarchy of grids for multigrid methods solving a PDE, discretized on grid $(2, 2)$. Standard coarsening uses the grids $(2, 2)$, $(1, 1)$, $(0, 0)$. Semicoarsening in the x-direction uses the grids $(2, 2)$, $(1, 2)$, $(0, 2)$.

One step of this method consists of a multigrid step with standard coarsening where the smoothing has been replaced by semicoarsening in the x-direction followed by semicoarsening in the y-direction. This method could also be called alternating semicoarsening. Full weighting restriction is used to transfer from a fine to a coarse grid. To transfer from a coarse to fine grid linear interpolation is used in the semicoarsening case, and bilinear interpolation is used in the standard coarsening case. This is the method used for the numerical experiments. A V-cycle is used for standard coarsening as well as semicoarsening. For Figure 3.6 the sequence of grids visited would be (2,2), (1,2), (0,2), (1,2), (2,2), (2,1), (2,0), (2,1), (2,2), (1,1), (0,1), (1,1), (1,0), (1,1), (0,0) and back. Figure 3.7 illustrates, on a grid hierarchy with 4 levels for each direction, a variant that combines a standard coarsening V-cycle with an x-semicoarsening V-cycle for presmoothing and a y-semicoarsening V-cycle for postsmoothing.

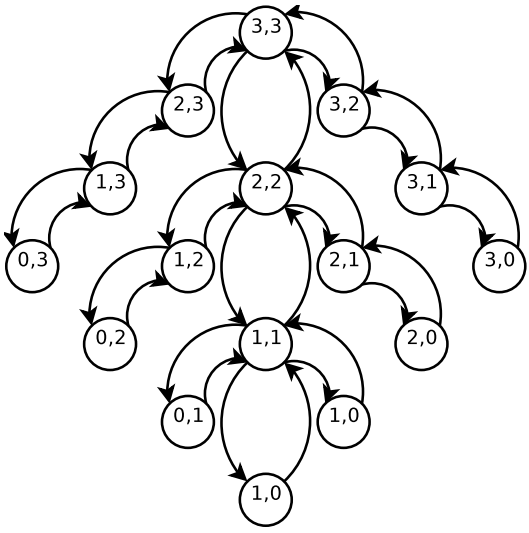


Figure 3.7: MGS iteration using V-cycles for standard coarsening, x-semicoarsening presmoothing and y-semicoarsening postsmoothing.

3.4 Convergence Analysis

As explained in the previous chapter the convergence of an iterative method of the form

$$u^{(\nu)} = \mathcal{M}u^{(\nu-1)} + g,$$

is determined by the spectral radius of the iteration operator $\mathcal{M}$. If $\mathcal{M}$ represents a multigrid method for the time-dependent equation

$$\dot{u} = Lu + f, \tag{3.11}$$

the spectral radius is calculated from

$$\rho(\mathcal{M}) = \max_{z \in \Sigma} \rho(M(z)).$$

The operator $M(z)$ is the iteration operator of the time-independent equation

$$zu = Lu + f. \tag{3.12}$$

The set of complex numbers Σ is determined by the way the time dimension is handled (see §2.5.5). The system of equations (3.12) can be interpreted as the Laplace transform of the system of ODEs (3.11), or equivalently as the discretization of the complex Helmholtz equation corresponding to the

time-dependent PDE that resulted in (3.11). The spectral radius $\rho(M(z))$ is the spectral radius of the multigrid method for (3.12) and can be analyzed using Fourier analysis [TOS01, Wes92, TZ95, WJ05]. In this section we describe the standard two-grid analysis. A two-grid analysis assumes one coarser grid on which the correction equation is solved exactly (see §2.6). The iteration operator can then be written as

$$M(z) = S(z)^{\nu_2} C(z) S(z)^{\nu_1}, \tag{3.13}$$

where

$$S(z) = (zI - L^+)^{-1} L^-$$

is the operator of the single-grid waveform relaxation method used as a smoother, and ν_1 and ν_2 are the number of pre- and postsmoothing steps. The operator $C(z)$ is the two-grid coarse grid correction operator. It can be written as

$$C(z) = I - P\bar{L}(z)^{-1} RL(z), \tag{3.14}$$

where $L(z) = zI - L$ is the diffusion operator on the fine grid, $\bar{L}(z) = z\bar{I} - \bar{L}$ is the differential operator on the coarse grid and R and P are the restriction and prolongation operators.

The effect of the operator $M(z)$ on the error can be analyzed using Fourier analysis. This is a classical way to analyze multigrid and other methods. To make the text more self contained, we present such an analysis in what follows. The Fourier analysis presented here also provides a reference point for a similar analysis based on functional calculus and shift operators (see §7.5). We follow the exposition of [Wes92]. Similar analyses can be found in [TOS01, HV95, VH95]. For analyses of many more methods and three-grid analyses we refer to [WJ05]. For the analysis we assume a general nine-point stencil. To keep the notation concise, the dependency on the complex parameter z is omitted at first. The parameter is easily reintroduced when the specific model problems are considered in §3.4.9.

3.4.1 Grids and Fourier Modes

To investigate the effect of the two-grid operator M on the error, we decompose the error into Fourier modes and study the effect of the iteration on these modes. For problems with Dirichlet boundary conditions a decomposition using sinusoidal modes is most natural. This leads to the so-called rigorous Fourier analysis. However, we use exponential Fourier modes rather than sines since they are easier to manipulate. This type of analysis is exact for problems with periodic boundary conditions. The results are only approximately valid for problems with Dirichlet conditions. A heuristic modification which disregards the constant Fourier modes, which do not

appear in the error because of the Dirichlet boundary conditions, can lead to a slightly better approximation. We do not use this heuristic here. In general, the analysis based on exponential Fourier modes gives an indication of the convergence to expect for other than periodic boundary conditions when they are treated properly (some extra smoothing near the boundary may be necessary for some problems for example). The so-called local Fourier mode analysis is a closely related variant where infinite grids are considered instead of periodic grids. The use of infinite grids clearly shows that no boundary conditions are taken into account. For more details on the different types of multigrid analysis see [TOS01].

We define a fine grid Ω and a coarse grid $\overline{\Omega}$ by

$$\begin{aligned}\Omega &= \{(x_i, y_j) : x_i = i\Delta x,\, y_j = j\Delta y,\, i = 0, \ldots, n_x,\, j = 0, \ldots, n_y\},\\ \overline{\Omega} &= \{(\overline{x}_{\overline{\imath}}, \overline{y}_{\overline{\jmath}}) : \overline{x}_{\overline{\imath}} = \overline{\imath}\overline{\Delta x},\, \overline{y}_{\overline{\jmath}} = \overline{\jmath}\overline{\Delta y},\, \overline{\imath} = 0, \ldots, \overline{n}_x,\, \overline{\jmath} = 0, \ldots, \overline{n}_y\},\end{aligned} \tag{3.15}$$

where $\Delta x = n_x^{-1}$ and $\Delta y = n_y^{-1}$ and $\overline{\Delta x} = \overline{n}_x^{-1}$ and $\overline{\Delta y} = \overline{n}_y^{-1}$. For simplicity, the numbers n_x, n_y, $\overline{n}_x$, and $\overline{n}_y$ are assumed to be even. For standard coarsening we have $(\overline{n}_x, \overline{n}_y) = (n_x/2, n_y/2)$, for x-semicoarsening $(\overline{n}_x, \overline{n}_y) = (n_x/2, n_y)$ and for y-semicoarsening $(\overline{n}_x, \overline{n}_y) = (n_x, n_y/2)$. The exponential Fourier mode with wavenumber θ is given by

$$\psi(\theta)_{i,j} = \exp\left(\sqrt{-1}(i\theta_x + j\theta_y)\right), \tag{3.16}$$

where i and j are the indices of the grid points of Ω and the set of all $\theta = (\theta_x, \theta_y)$ indicates the wavenumbers of the modes that can exist on the grid Ω. Because the letters i and j are used to denote grid indices the imaginary unit is written as $\sqrt{-1}$. To avoid this cumbersome notation we introduce the notation

$$q(\gamma) = \exp\left(\gamma\sqrt{-1}\right),$$

to denote the complex number at angle γ on the unit circle. Using this notation the exponential Fourier mode becomes

$$\psi(\theta)_{i,j} = q(i\theta_x + j\theta_y).$$

The exponential Fourier modes $\overline{\psi}(\overline{\theta})$, on the coarse grid, are defined analogously. The sets of wavenumbers are given by

$$\begin{aligned}\theta \in \Theta &= \{(\theta_x, \theta_y) : \theta_\alpha = 2\pi k_\alpha/n_\alpha\,,\ k_\alpha = -n_\alpha/2 + 1, \ldots, n_\alpha/2\,,\ \alpha = x, y\},\\ \overline{\theta} \in \overline{\Theta} &= \{(\overline{\theta}_x, \overline{\theta}_y) : \overline{\theta}_\alpha = 2\pi\overline{k}_\alpha/\overline{n}_\alpha\,,\ \overline{k}_\alpha = -\overline{n}_\alpha/2 + 1, \ldots, \overline{n}_\alpha/2\,,\ \alpha = x, y\}.\end{aligned} \tag{3.17}$$

In local Fourier mode analysis the wavenumbers are allowed to vary continuously (see [TOS01, WJ05]). The analysis presented here is based on periodic grid functions. For fine enough grids, the results are very similar.

Every periodic grid function $e^{(\nu-1)}$ defined on the grid Ω can be decomposed into Fourier modes as

$$e^{(\nu-1)} = \sum_{\theta \in \Theta} \tilde{e}^{(\nu-1)}(\theta)\psi(\theta). \tag{3.18}$$

3.4.2 Fourier Mode Harmonics

In what follows we show that certain spaces spanned by four related exponential modes are invariant under operator M. The Fourier modes (3.16) are eigenfunctions of the differential operator L and of the smoothing operator for the Jacobi method S_{Jac}, provided the problem has constant coefficients and periodic boundary conditions. Other multigrid operators map an exponential Fourier mode onto a linear combination of related Fourier modes. For each $\theta \in \Theta' = \Theta \cap [-\frac{\pi}{2}, \frac{\pi}{2})^2$, we define a vector

$$\Psi(\theta) = [\psi(\theta^1)\, \psi(\theta^2)\, \psi(\theta^3)\, \psi(\theta^4)]^T, \tag{3.19}$$

that groups four related Fourier modes (harmonics). The wavenumbers of the components are given by

$$\begin{aligned}
\theta^1 &= \theta \\
\theta^2 &= \theta - \begin{pmatrix} \text{sign}(\theta_x) \\ \text{sign}(\theta_y) \end{pmatrix} \pi, \\
\theta^3 &= \theta - \begin{pmatrix} 0 \\ \text{sign}(\theta_y) \end{pmatrix} \pi, \\
\theta^4 &= \theta - \begin{pmatrix} \text{sign}(\theta_x) \\ 0 \end{pmatrix} \pi.
\end{aligned} \tag{3.20}$$

The domains of the wavenumbers θ^1, θ^2, θ^3 and θ^4 are illustrated in Figure 3.8. Using the vectors $\Psi(\theta)$ from (3.19), any relation (3.18) can rewritten as

$$e^{(\nu-1)} = \sum_{\theta \in \Theta'} e^{(\nu-1)}(\theta)^T \Psi(\theta),$$

where $e^{(\nu-1)}(\theta)$ is a vector of four elements. The error after applying the iteration operator M is decomposed as

$$e^{(\nu)} = M e^{(\nu-1)} = \sum_{\theta \in \Theta'} e^{(\nu)}(\theta)^T \Psi(\theta).$$

We are now looking for the relation between $e^{(\nu)}(\theta)$ and $e^{(\nu-1)}(\theta)$.

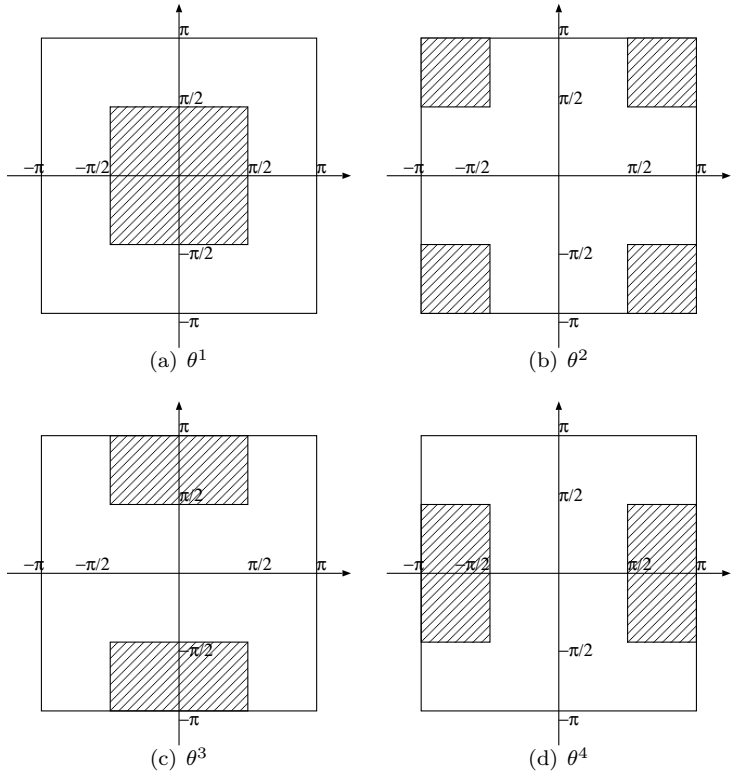


Figure 3.8: Wavenumbers of related exponential Fourier modes.

3.4.3 Two-Grid Operator

The operators involved in the two-grid iteration realize the following mappings, where standard coarsening is indicated by xy, semicoarsening is indicated by x or y:

$$L \quad : \quad \text{span}[\psi(\theta)] \rightarrow \text{span}[\psi(\theta)] \ ,$$

$$S \quad : \quad \text{span}\,\Psi(\theta) \rightarrow \text{span}\,\Psi(\theta) \ ,$$

$$R \quad : \quad \text{span}\,\Psi(\theta) \rightarrow \begin{cases} \text{span}[\psi(\theta^1)] & \text{(xy)} \\ \text{span}[\psi(\theta^1)\,\psi(\theta^3)] & \text{(x)} \\ \text{span}[\psi(\theta^1)\,\psi(\theta^4)] & \text{(y)} \end{cases} ,$$

$$\overline{L} \quad : \quad \begin{array}{lcll} \text{span}[\psi(\theta^1)] & \rightarrow & \text{span}[\psi(\theta^1)] & \text{(xy)} \\ \text{span}[\psi(\theta^1)\,\psi(\theta^3)] & \rightarrow & \text{span}[\psi(\theta^1)\,\psi(\theta^3)] & \text{(x)} \\ \text{span}[\psi(\theta^1)\,\psi(\theta^4)] & \rightarrow & \text{span}[\psi(\theta^1)\,\psi(\theta^4)] & \text{(y)} \end{array} ,$$

$$P \quad : \quad \left.\begin{array}{ll} \text{span}[\psi(\theta^1)] & \text{(xy)} \\ \text{span}[\psi(\theta^1)\,\psi(\theta^3)] & \text{(x)} \\ \text{span}[\psi(\theta^1)\,\psi(\theta^4)] & \text{(y)} \end{array}\right\} \rightarrow \text{span}\,\Psi(\theta) \ .$$

These mappings can be represented by matrices. The matrix corresponding to an operator is called the symbol of the operator. The rest of the analysis consists of the derivation of these symbols for the operators in the two-grid iteration. The two-grid operator maps a component of the error in the space spanned by $\Psi(\theta)$ into the same space. This map can be represented by a four by four matrix. We can therefore write

$$e^{(\nu)}(\theta) = M(\theta)e^{(\nu-1)}(\theta),$$

where $M(\theta) \in \mathbb{C}^{4\times 4}$ and $\theta \in \Theta'$. The symbol of the two-grid operator can be written as

$$M(\theta) = S(\theta)^{\nu_2}\left(I - P(\theta)\overline{L}(\overline{\theta})^{-1}R(\theta)L(\theta)\right)S(\theta)^{\nu_1},$$

where $S(\theta)$, $P(\theta)$, $\overline{L}(\overline{\theta})$, $R(\theta)$ and $L(\theta)$ are now symbols instead of operators and I is a four by four identity matrix. When the exponential Fourier modes are chosen as a basis for representing the errors, the operator M becomes block diagonal with 4×4 blocks on the diagonal (the symbols). Therefore, the spectral radius of the operator M corresponds to the maximum of the spectral radii of the symbols $M(z, \theta)$, and thus

$$\rho(\mathcal{M}) = \max_{z\in\Sigma}\max_{\theta\in\Theta'}\rho(M(z,\theta)). \tag{3.21}$$

For the MGS (or alternating semicoarsening) method, 3 coarser grids are used for the analysis: one corresponding to each of the semicoarsening steps and one for the standard coarsening step. The multigrid operator (and the corresponding symbol) can be written as follows

$$\begin{aligned} M &= (M_x M_y)^{\nu_2} C (M_x M_y)^{\nu_1}, & C &= I - P\bar{L}^{-1}RL, \\ M_x &= S^{\mu_2} C_x S^{\mu_1}, & C_x &= I - P_x \bar{L}_x^{-1} R_x L, \\ M_y &= S^{\mu_2} C_y S^{\mu_1}, & C_y &= I - P_y \bar{L}_y^{-1} R_y L, \end{aligned} \tag{3.22}$$

where M is the MGS operator (symbol), C is the standard coarsening operator (symbol), M_x and M_y are the semicoarsening multigrid operators (symbols) and C_x and C_y are the corresponding semicoarsening coarse grid correction operators (symbols).

We now derive for each operator the corresponding symbol.

3.4.4 Differential Operator

We assume that the spatial discretization is given by the general nine-point stencil

$$\begin{bmatrix} s_{-1,1} & s_{0,1} & s_{1,1} \\ s_{-1,0} & s_{0,0} & s_{1,0} \\ s_{-1,-1} & s_{0,-1} & s_{1,-1} \end{bmatrix}. \tag{3.23}$$

The resulting analysis is general enough for the model problems considered here. For an exponential Fourier mode $\psi(\theta)$ we can write the action of the differential operator L as

$$\begin{aligned} \left(L\psi(\theta)\right)_{i,j} =& s_{0,0}\,\psi(\theta)_{i,j} + \\ & s_{0,1}\,\psi(\theta)_{i,j+1} + s_{1,0}\,\psi(\theta)_{i+1,j} + \\ & s_{0,-1}\,\psi(\theta)_{i,j-1} + s_{-1,0}\,\psi(\theta)_{i-1,j} + \\ & s_{1,1}\,\psi(\theta)_{i+1,j+1} + s_{1,-1}\,\psi(\theta)_{i+1,j-1} + \\ & s_{-1,-1}\,\psi(\theta)_{i-1,j-1} + s_{-1,1}\,\psi(\theta)_{i-1,j+1} \end{aligned}$$

To transform this expression we use relations of the form

$$\psi(\theta)_{i\pm1,j} = q((i \pm 1)\theta_x + j\theta_y) = q(\pm\theta_y)\psi(\theta)_{i,j}.$$

In general we have

$$\psi(\theta)_{i+i',j+j'} = q(i'\theta_x + j'\theta_y)\psi(\theta)_{i,j}. \tag{3.24}$$

We introduce the shorthands

$$H(\theta) = s_{1,0}\, q(\theta_x) + s_{-1,0}\, q(-\theta_x), \tag{3.25}$$
$$V(\theta) = s_{0,1}\, q(\theta_y) + s_{0,-1}\, q(-\theta_y), \tag{3.26}$$
$$\begin{aligned} D(\theta) = \;& s_{1,1}\, q(\theta_x + \theta_y) + s_{1,-1}\, q(\theta_x - \theta_y) + \\ & s_{-1,-1}\, q(-\theta_x - \theta_y) + s_{-1,1}\, q(-\theta_x + \theta_y). \end{aligned} \tag{3.27}$$

The action of the differential operator L on an exponential Fourier mode $\psi(\theta)$ can be written as

$$\left(L\psi(\theta)\right)_{i,j} = \tilde{L}(\theta)\psi(\theta)_{i,j},$$

where

$$\tilde{L}(\theta) = s_{0,0} + H(\theta) + V(\theta) + D(\theta).$$

This means that applying the differential operator corresponds to multiplication by the number $\tilde{L}(\theta)$. The exponential Fourier modes are therefore eigenfunctions of the differential operator. It is clear from (3.24) that this is true for any operator that can be represented in stencil notation. Since the operator L maps each harmonic in $\Psi(\theta)$ onto a multiple of itself, the resulting symbol for the differential operator is the diagonal matrix

$$L(\theta) = \begin{bmatrix} \tilde{L}(\theta^1) & & & \\ & \tilde{L}(\theta^2) & & \\ & & \tilde{L}(\theta^3) & \\ & & & \tilde{L}(\theta^4) \end{bmatrix}.$$

3.4.5 Restriction Operator

Standard Coarsening

We derive the symbol for the full weighting restriction operator represented in stencil notation as

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

After restriction of the $\psi(\theta)$ a grid point $\bar{\imath}, \bar{\jmath}$ on the coarse grid has the value

$$\begin{aligned} \left(R\psi(\theta)\right)_{\bar{\imath},\bar{\jmath}} = & \frac{1}{4}\psi(\theta)_{i,j} + \\ & \frac{1}{8}\big(\psi(\theta)_{i,j+1} + \psi(\theta)_{i+1,j} + \psi(\theta)_{i,j-1} + \psi(\theta)_{i-1,j}\big) + \\ & \frac{1}{16}\big(\psi(\theta)_{i+1,j+1} + \psi(\theta)_{i+1,j-1} + \psi(\theta)_{i-1,j-1} + \psi(\theta)_{i-1,j+1}\big). \end{aligned}$$

Using (3.24) this can be written as

$$\begin{aligned}(R\psi(\theta))_{\bar{\imath},\bar{\jmath}} = \Big(&\frac{1}{4} + \frac{1}{8}\big(q(\theta_x) + q(-\theta_x) + q(\theta_y) + q(-\theta_y)\big) + \\ &\frac{1}{16}\big(q(\theta_x)q(\theta_y) + q(-\theta_x)q(\theta_y) + \\ &\quad q(\theta_x)q(-\theta_y) + q(-\theta_x)q(-\theta_y)\big)\Big)\psi(\theta)_{i,j} \\ = &\frac{1}{4}\left(1 + \cos(\theta_x) + \cos(\theta_y) + \cos(\theta_x)\cos(\theta_y)\right)\psi(\theta)_{i,j}.\end{aligned}$$

Since $(i,j) = (2\bar{\imath}, 2\bar{\jmath})$ and $(\overline{\theta}_x, \overline{\theta}_y) = (2\theta_x, 2\theta_y)$ (see (3.15) and (3.17)), we find that

$$\psi(\theta) = q(i\theta_x + j\theta_y) = q(\bar{\imath}\overline{\theta}_x + \bar{\jmath}\overline{\theta}_y) = \overline{\psi}(\overline{\theta}).$$

We can therefore write $R\psi(\theta)$ as a function of the Fourier mode $\overline{\psi}(\overline{\theta})$ on the coarse grid as

$$(R\psi(\theta))_{i,j} = \tilde{R}(\theta)\overline{\psi}(\overline{\theta})_{\bar{\imath},\bar{\jmath}}. \tag{3.28}$$

where

$$\begin{aligned}\tilde{R}(\theta) &= \frac{1}{4}\big(1 + \cos(\theta_x) + \cos(\theta_y) + \cos(\theta_x)\cos(\theta_y)\big) \\ &= \frac{1}{4}\big(1 + \cos(\theta_x)\big)\big(1 + \cos(\theta_y)\big).\end{aligned}$$

The restriction operator maps the Fourier mode $\psi(\theta)$ on the fine grid onto the corresponding mode $\overline{\psi}(\overline{\theta})$ on the coarse grid. For standard coarsening we have $(\overline{\theta}_x, \overline{\theta}_y) = (2\theta_x, 2\theta_y)$. Hence, the components $\overline{\theta}^1$, $\overline{\theta}^2$, $\overline{\theta}^3$ and $\overline{\theta}^4$ are equal up to a multiple of 2π (see (3.20)). In other words, the fine grid Fourier modes $\psi(\theta^1)$, $\psi(\theta^2)$, $\psi(\theta^3)$ and $\psi(\theta^4)$ are mapped onto a single coarse grid Fourier mode $\overline{\psi}(\overline{\theta}^1) = \overline{\psi}(\overline{\theta}^2) = \overline{\psi}(\overline{\theta}^3) = \overline{\psi}(\overline{\theta}^4)$. The symbol for standard coarsening full weighting restriction therefore becomes

$$R(\theta) = \begin{bmatrix} \tilde{R}(\theta^1) & \tilde{R}(\theta^2) & \tilde{R}(\theta^3) & \tilde{R}(\theta^4) \end{bmatrix}.$$

Semicoarsening

In the case of x- and y-semicoarsening the stencils for the full weighting restriction operators are

$$\frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}, \qquad \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}.$$

For x-coarsening restriction assigns to a grid point $(\bar{\imath}, \bar{\jmath})$ on the coarse grid the value

$$\begin{aligned}(R\psi(\theta))_{\bar{\imath},\bar{\jmath}} &= \frac{1}{4}\Big(2\psi(\theta)_{i,j} + \big(\psi(\theta)_{i+1,j} + \psi(\theta)_{i-1,j}\big)\Big) \\ &= \frac{1}{2}\big(1+\cos(\theta_x)\big)\psi(\theta)_{i,j}.\end{aligned}$$

Again we find that $\psi(\theta)_{i,j} = \overline{\psi}(\overline{\theta})_{\bar{\imath},\bar{\jmath}}$ so that we have

$$(R\psi(\theta))_{i,j} = \tilde{R}(\theta)\overline{\psi}(\overline{\theta})_{\bar{\imath},\bar{\jmath}},$$

where

$$\tilde{R}(\theta) = \frac{1}{2}\left(1+\cos(\theta_x)\right).$$

In the same way we find for y-coarsening

$$\tilde{R}(\theta) = \frac{1}{2}\left(1+\cos(\theta_y)\right).$$

For x-coarsening the equality (up to a multiple of 2π) only holds for the x-component of $\overline{\theta}^1$, $\overline{\theta}^2$, $\overline{\theta}^3$ and $\overline{\theta}^4$. The modes $\psi(\theta^1)$ and $\psi(\theta^4)$ are mapped onto the modes $\overline{\psi}(\overline{\theta}^1) = \overline{\psi}(\overline{\theta}^4)$. The modes $\psi(\theta^2)$ en $\psi(\theta^3)$ are mapped onto $\overline{\psi}(\overline{\theta}^2) = \overline{\psi}(\overline{\theta}^3)$. For y-coarsening the modes $\psi(\theta^1)$ and $\psi(\theta^3)$ are mapped onto $\overline{\psi}(\overline{\theta}^1) = \overline{\psi}(\overline{\theta}^3)$ and the modes $\psi(\theta^2)$ and $\psi(\theta^4)$ onto $\overline{\psi}(\overline{\theta}^2) = \overline{\psi}(\overline{\theta}^4)$. The symbols for full weighting restriction for x- and y-semicoarsening become

$$R(\theta) = \begin{bmatrix} \tilde{R}(\theta^1) & 0 & 0 & \tilde{R}(\theta^4) \\ 0 & \tilde{R}(\theta^2) & \tilde{R}(\theta^3) & 0 \end{bmatrix},$$

$$R(\theta) = \begin{bmatrix} \tilde{R}(\theta^1) & 0 & \tilde{R}(\theta^3) & 0) \\ 0 & \tilde{R}(\theta^2) & 0 & \tilde{R}(\theta^4) \end{bmatrix}.$$

3.4.6 Differential Operator on the Coarse Grid

Standard Coarsening

When using standard coarsening the restriction operator maps all modes onto $\overline{\psi}(\overline{\theta})$. The symbol of the differential operator on the coarse grid can be written as

$$\overline{L}(\overline{\theta}) = \begin{bmatrix} \tilde{L}(\overline{\theta}^1) \end{bmatrix}.$$

Semicoarsening

When using x-semicoarsening the restriction operator maps all modes onto $\overline{\psi}(\overline{\theta}^1)$ or $\overline{\psi}(\overline{\theta}^3)$. The symbol of the differential operator on the coarse grid can be written as

$$\overline{L}(\overline{\theta}) = \begin{bmatrix} \tilde{L}(\overline{\theta}^1) & \\ & \tilde{L}(\overline{\theta}^3) \end{bmatrix}.$$

For y-coarsening we find

$$\overline{L}(\overline{\theta}) = \begin{bmatrix} \tilde{L}(\overline{\theta}^1) & \\ & \tilde{L}(\overline{\theta}^4) \end{bmatrix}.$$

3.4.7 Prolongation Operator

Standard Coarsening

In the case of standard coarsening we use the bilinear interpolation operator to transfer from coarse to fine grids. If the grid indices i and j on the fine grid are both even, we transfer the value at the corresponding point on the coarse grid. If i is even and j is odd (and vice versa), then the point on the fine grid is between two points on the coarse grid. The value on the fine grid is the average of the value at the two points on the coarse grid. If both i and j are odd the value on the fine grid is the average of the four surrounding points on the coarse grid.

For a Fourier mode $\overline{\psi}(\overline{\theta})$ we find using (3.24) and $\overline{\psi}(\overline{\theta})_{\overline{i},\overline{j}} = \psi(\theta)_{i,j}$ that

$$(P\overline{\psi}(\overline{\theta}))_{i,j} = \begin{cases} \psi(\theta)_{i,j} & \text{if } i \text{ and } j \text{ even,} \\ \cos(\theta_x)\psi(\theta)_{i,j} & \text{if } i \text{ odd and } j \text{ even,} \\ \cos(\theta_y)\psi(\theta)_{i,j} & \text{if } i \text{ even and } j \text{ odd,} \\ \cos(\theta_x)\cos(\theta_y)\psi(\theta)_{i,j} & \text{if } i \text{ and } j \text{ odd.} \end{cases} \tag{3.29}$$

The grid function $\psi(\theta)$ or $\overline{\psi}(\overline{\theta})$ is clearly not an eigenfunction of the prolongation operator. But we can show that the prolongation operator maps the function $\overline{\psi}(\overline{\theta})$ onto a function in the space spanned by $\Psi(\theta)$. We formulate this property in a more general form. Every grid function satisfying

$$\phi_{i,j} = \begin{cases} \alpha_{00}\psi(\theta^1)_{i,j} & \text{if } i \text{ and } j \text{ even,} \\ \alpha_{10}\psi(\theta^1)_{i,j} & \text{if } i \text{ odd and } j \text{ even,} \\ \alpha_{01}\psi(\theta^1)_{i,j} & \text{if } i \text{ even and } j \text{ odd,} \\ \alpha_{11}\psi(\theta^1)_{i,j} & \text{if } i \text{ and } j \text{ odd.} \end{cases} \tag{3.30}$$

can be represented in the space spanned by $\Psi(\theta)$. This means that there are coefficients a^1, a^2, a^3 and a^4 such that

$$\phi = a^1\psi(\theta^1) + a^2\psi(\theta^2) + a^3\psi(\theta^3) + a^4\psi(\theta^4). \tag{3.31}$$

The proof of this property is based on the relation between the values of $\psi(\theta^1)$ and the other basis functions evaluated at the grid points. For $\psi(\theta^2)$ we find

$$\psi(\theta^2)_{i,j} = q(-\pi\,\mathrm{sign}(\theta_x)i)q(-\pi\,\mathrm{sign}(\theta_y)j)\psi(\theta^1)_{i,j} \tag{3.32}$$

or

$$\psi(\theta^2)_{i,j} = \begin{cases} \psi(\theta^1)_{i,j} & \text{if } i \text{ and } j \text{ both even or both odd} \\ -\psi(\theta^1)_{i,j} & \text{otherwise} \end{cases} \tag{3.33}$$

Similar expressions hold for $\psi(\theta^3)$ and $\psi(\theta^4)$. They are summarized in the following table.

	$\psi(\theta^1)$	$\psi(\theta^2)$	$\psi(\theta^3)$	$\psi(\theta^4)$
i even, j even	1	1	1	1
i odd, j odd	1	1	−1	−1
i even, j odd	1	−1	−1	1
i odd, j even	1	−1	1	−1

Together with (3.30) and (3.31) this results in the equations

$$\begin{aligned}
\alpha_{00}\psi(\theta^1)_{ij} &= a^1\psi(\theta^1)_{ij} + a^2\psi(\theta^1)_{ij} + a^3\psi(\theta^1)_{ij} + a^4\psi(\theta^1)_{ij}, \\
\alpha_{11}\psi(\theta^1)_{ij} &= a^1\psi(\theta^1)_{ij} + a^2\psi(\theta^1)_{ij} - a^3\psi(\theta^1)_{ij} - a^4\psi(\theta^1)_{ij}, \\
\alpha_{01}\psi(\theta^1)_{ij} &= a^1\psi(\theta^1)_{ij} - a^2\psi(\theta^1)_{ij} - a^3\psi(\theta^1)_{ij} + a^4\psi(\theta^1)_{ij}, \\
\alpha_{10}\psi(\theta^1)_{ij} &= a^1\psi(\theta^1)_{ij} - a^2\psi(\theta^1)_{ij} + a^3\psi(\theta^1)_{ij} - a^4\psi(\theta^1)_{ij}.
\end{aligned}$$

The relation between the α coefficients in (3.30) and the a coefficients in (3.31) can be written as

$$\begin{bmatrix} \alpha_{00} \\ \alpha_{11} \\ \alpha_{01} \\ \alpha_{10} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} a^1 \\ a^2 \\ a^3 \\ a^4 \end{bmatrix}.$$

The a coefficients can therefore be found as

$$\begin{bmatrix} a^1 \\ a^2 \\ a^3 \\ a^4 \end{bmatrix} = \mathbf{T}^{-1} \begin{bmatrix} \alpha_{00} \\ \alpha_{11} \\ \alpha_{01} \\ \alpha_{10} \end{bmatrix},$$

where

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}, \qquad \mathbf{T}^{-1} = \frac{1}{4}\mathbf{T}.$$

For bilinear interpolation we derive from (3.29) that

$$\begin{cases} \alpha_{00} = 1, \\ \alpha_{11} = \cos(\theta_x)\cos(\theta_y), \\ \alpha_{01} = \cos(\theta_y), \\ \alpha_{10} = \cos(\theta_x). \end{cases}$$

which leads to

$$P\overline{\psi}(\overline{\theta}) = P(\theta)\Psi(\theta)$$

with

$$\begin{aligned} P(\theta) &= \frac{1}{4}\big(1 + \cos(\theta_x) + \cos(\theta_y) + \cos(\theta_x)\cos(\theta_y)\big) \\ &= \frac{1}{4}\big(1 + \cos(\theta_x)\big)\big(1 + \cos(\theta_y)\big). \end{aligned}$$

Note that

$$P(\theta) = R(\theta)^T.$$

This is the equivalent of relation (3.10) that holds for the corresponding operators. The same relation holds for all the other restriction and prolongation combinations considered here.

Semicoarsening

For x-coarsening linear interpolation is used as the interpolation operator. Fine grid points with even i can take a value directly from the coarse grid. Points with odd i take the average of the two neighboring points on the coarse grid. Prolongation of a Fourier mode $\overline{\psi}(\overline{\theta})$ gives at the fine grid point (i, j)

$$(P\overline{\psi}(\overline{\theta}))_{i,j} = \begin{cases} \psi(\theta)_{i,j} & \text{if } i \text{ even}, \\ \cos(\theta_x)\psi(\theta)_{i,j} & \text{if } i \text{ odd}. \end{cases}$$

The coefficients in (3.31) are

$$\begin{cases} \alpha_{00} = \alpha_{01} = 1, \\ \alpha_{10} = \alpha_{11} = \cos(\theta_x). \end{cases}$$

Again we find

$$P\overline{\psi}(\overline{\theta}) = P(\theta)\Psi(\theta)$$

with $P(\theta) = R(\theta)^T$. This last relation also holds for y-coarsening.

3.4.8 Smoothing Operator

We derive the symbol for a smoothing operator that transforms $\psi(\theta^1)$, $\psi(\theta^2)$, $\psi(\theta^3)$ and $\psi(\theta^4)$ into functions of the form (3.30). Red-Black, four color and zebra line relaxation are all smoothers of this kind. If we apply these smoothers to $\psi(\theta^1)$ then we become grid functions of the form (3.30) with α coefficients given by the following expressions. The same notations as in the previous sections are used.

Red-Black

$$\begin{cases} \alpha_{11}(\theta^1) = \alpha_{00}(\theta^1) = -(H(\theta^1) + V(\theta^1))/s_{0,0}\,, \\ \alpha_{10}(\theta^1) = \alpha_{01}(\theta^1) = \alpha_{11}^2(\theta^1). \end{cases}$$

Four Color

$$\begin{cases} \alpha_{11}(\theta^1) = -(H(\theta^1) + V(\theta^1) + D(\theta^1))/s_{0,0}\,, \\ \alpha_{00}(\theta^1) = -(H(\theta^1) + V(\theta^1) + \alpha_{11}(\theta^1)D(\theta^1))/s_{0,0}\,, \\ \alpha_{01}(\theta^1) = -(\alpha_{11}(\theta^1)H(\theta^1) + \alpha_{00}(\theta^1)V(\theta^1) + D(\theta^1))/s_{0,0}\,, \\ \alpha_{10}(\theta^1) = -(\alpha_{00}(\theta^1)H(\theta^1) + \alpha_{11}(\theta^1)V(\theta^1) + \alpha_{01}(\theta^1)D(\theta^1))/s_{0,0}\,. \end{cases}$$

Horizontal Zebra

$$\begin{cases} \alpha_{11}(\theta^1) = \alpha_{01}(\theta^1) = -(V(\theta^1) + D(\theta^1))/(s_{0,0} + H(\theta^1)), \\ \alpha_{10}(\theta^1) = \alpha_{00}(\theta^1) = \alpha_{11}^2(\theta^1). \end{cases}$$

Vertical Zebra

$$\begin{cases} \alpha_{11}(\theta^1) = \alpha_{10}(\theta^1) = -(H(\theta^1) + D(\theta^1))/(s_{0,0} + V(\theta^1)), \\ \alpha_{01}(\theta^1) = \alpha_{00}(\theta^1) = \alpha_{11}^2(\theta^1). \end{cases}$$

The corresponding a coefficients are the first columns of the symbol $S(\theta)$, which is therefore equal to

$$\mathbf{T}^{-1} \begin{bmatrix} \alpha_{00}(\theta^1) \\ \alpha_{11}(\theta^1) \\ \alpha_{01}(\theta^1) \\ \alpha_{10}(\theta^1) \end{bmatrix}.$$

For the mode $\psi(\theta^2)$ we have to replace $\psi(\theta^1)$ in (3.30) by $\psi(\theta^2)$. Using the relation (3.33) between $\psi(\theta^1)$ and $\psi(\theta^2)$ we find that

$$\phi_{i,j} = \begin{cases} \alpha_{00}\psi(\theta^2)_{i,j} = \alpha_{00}\psi(\theta^1)_{i,j} & \text{if } i \text{ and } j \text{ even}, \\ \alpha_{10}\psi(\theta^2)_{i,j} = \alpha_{10}\psi(\theta^1)_{i,j} & \text{if } i \text{ odd and } j \text{ even}, \\ \alpha_{01}\psi(\theta^2)_{i,j} = -\alpha_{01}\psi(\theta^1)_{i,j} & \text{if } i \text{ even and } j \text{ odd}, \\ \alpha_{11}\psi(\theta^2)_{i,j} = -\alpha_{11}\psi(\theta^1)_{i,j} & \text{if } i \text{ and } j \text{ odd}. \end{cases}$$

so that we can use the following substitution

$$\begin{bmatrix} \alpha_{00} \\ \alpha_{11} \\ \alpha_{01} \\ \alpha_{10} \end{bmatrix} \rightarrow \begin{bmatrix} \alpha_{00} \\ \alpha_{11} \\ -\alpha_{01} \\ -\alpha_{10} \end{bmatrix}.$$

This corresponds to a component-wise multiplication by the second column of $\mathbf{T}$. Analogous results hold for the modes $\psi(\theta^3)$ and $\psi(\theta^4)$. This can be written in a compact form as

$$S(\theta) = \mathbf{T}^{-1}(\mathbf{T} \odot \mathbf{A}), \tag{3.34}$$

where $\mathbf{T} \odot \mathbf{A}$ denotes component-wise or Hadamard multiplication and

$$\mathbf{A} = \begin{bmatrix} \alpha_{00}(\theta^1) & \alpha_{00}(\theta^2) & \alpha_{00}(\theta^3) & \alpha_{00}(\theta^4) \\ \alpha_{11}(\theta^1) & \alpha_{11}(\theta^2) & \alpha_{11}(\theta^3) & \alpha_{11}(\theta^4) \\ \alpha_{01}(\theta^1) & \alpha_{01}(\theta^2) & \alpha_{01}(\theta^3) & \alpha_{01}(\theta^4) \\ \alpha_{10}(\theta^1) & \alpha_{10}(\theta^2) & \alpha_{10}(\theta^3) & \alpha_{10}(\theta^4) \end{bmatrix}. \tag{3.35}$$

For the calculation of $\mathbf{A}$ we can use the relations

$$\begin{array}{rcrcrcr} H(\theta^1) & = & -H(\theta^2) & = & H(\theta^3) & = & -H(\theta^4), \\ V(\theta^1) & = & -V(\theta^2) & = & -V(\theta^3) & = & V(\theta^4), \\ D(\theta^1) & = & D(\theta^2) & = & -D(\theta^3) & = & -D(\theta^4). \end{array}$$

3.4.9 Summary

Figure 3.9 illustrates the structure of the symbols for each operator making up the two-grid iteration operator. In this section we bring together the expressions for these symbols, give simplified expressions for some smoothers and give expressions for $s_{0,0}$, $H(\theta)$, $V(\theta)$ and $D(\theta)$ for the three model problems.

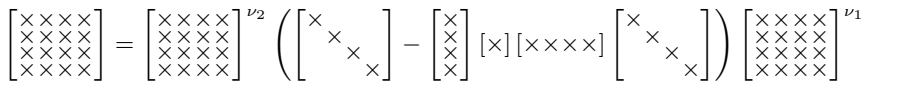
Coarse Grid Correction

The following three relations hold for all the coarse grid correction operators considered here (see also (3.20)).

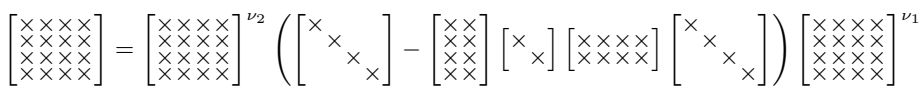
$$\tilde{L}(\theta) = s_{0,0} + H(\theta) + V(\theta) + D(\theta)$$

$$L(\theta) = \begin{bmatrix} \tilde{L}(\theta^1) & & & \\ & \tilde{L}(\theta^2) & & \\ & & \tilde{L}(\theta^3) & \\ & & & \tilde{L}(\theta^4) \end{bmatrix}$$

$$P(\theta) = R(\theta)^T$$



(a) Standard Coarsening



(b) Semicoarsening

Figure 3.9: Structure of the symbol of the two-grid operator.

Standard Coarsening

$$(\overline{\theta}_x^1, \overline{\theta}_y^1) = (2\theta_x^1, 2\theta_y^1)$$

$$\overline{L}(\overline{\theta}) = \left[\ \tilde{L}(\overline{\theta}^1)\ \right]$$

$$\tilde{R}(\theta) = \frac{1}{4}\big(1+\cos(\theta_x)\big)\big(1+\cos(\theta_y)\big)$$

$$R(\theta) = \left[\ \tilde{R}(\theta^1) \quad \tilde{R}(\theta^2) \quad \tilde{R}(\theta^3) \quad \tilde{R}(\theta^4)\ \right]$$

X-Coarsening

$$(\overline{\theta}_x^1, \overline{\theta}_y^1) = (2\theta_x^1, \theta_y^1), \qquad (\overline{\theta}_x^3, \overline{\theta}_y^3) = (2\theta_x^3, \theta_y^3)$$

$$\overline{L}(\overline{\theta}) = \begin{bmatrix} \tilde{L}(\overline{\theta}_1) & \\ & \tilde{L}(\overline{\theta}_3) \end{bmatrix}$$

$$\tilde{R}(\theta) = \frac{1}{2}\left(1+\cos(\theta_x)\right)$$

$$R(\theta) = \begin{bmatrix} \tilde{R}(\theta^1) & 0 & 0 & \tilde{R}(\theta^4) \\ 0 & \tilde{R}(\theta^2) & \tilde{R}(\theta^3) & 0 \end{bmatrix}$$

Y-Coarsening

$$(\overline{\theta}_x^1, \overline{\theta}_y^1) = (\theta_x^1, 2\theta_y^1), \qquad (\overline{\theta}_x^4, \overline{\theta}_y^4) = (\theta_x^4, 2\theta_y^4)$$

$$\overline{L}(\overline{\theta}) = \begin{bmatrix} \tilde{L}(\overline{\theta}_1) & \\ & \tilde{L}(\overline{\theta}_4) \end{bmatrix}$$

$$\tilde{R}(\theta) = \frac{1}{2}\left(1 + \cos(\theta_y)\right)$$

$$R(\theta) = \begin{bmatrix} \tilde{R}(\theta^1) & 0 & \tilde{R}(\theta^3) & 0) \\ 0 & \tilde{R}(\theta^2) & 0 & \tilde{R}(\theta^4) \end{bmatrix}$$

Smoothing

The general expressions obtained with (3.34) can be simplified for red-black Gauss-Seidel smoothing as well as for horizontal and vertical zebra Gauss-Seidel smoothing.

Red-Black If we define

$$\begin{aligned} \mu_1 &= \alpha_{00}(\theta^1) = -(H(\theta^1) + V(\theta^1))/s_{0,0}\,, \\ \mu_2 &= \alpha_{00}(\theta^3) = -(H(\theta^1) - V(\theta^1))/s_{0,0}\,, \end{aligned}$$

then the matrix A in (3.35) can be written as

$$A_{\mathrm{RB}}(\theta) = \begin{bmatrix} \mu_1 & -\mu_1 & \mu_2 & -\mu_2 \\ \mu_1 & -\mu_1 & \mu_2 & -\mu_2 \\ \mu_1^2 & \mu_1^2 & \mu_2^2 & \mu_2^2 \\ \mu_1^2 & \mu_1^2 & \mu_2^2 & \mu_2^2 \end{bmatrix}.$$

The symbol of the smoother becomes

$$S_{\mathrm{RB}}(\theta) = \frac{1}{2}\begin{bmatrix} \mu_1(1+\mu_1) & -\mu_1(1+\mu_1) & & \\ \mu_1(1-\mu_1) & -\mu_1(1-\mu_1) & & \\ & & \mu_2(1+\mu_2) & -\mu_2(1+\mu_2) \\ & & \mu_2(1-\mu_2) & -\mu_2(1-\mu_2) \end{bmatrix}.$$

Four Color For four color smoothing we use the general expression (3.34).

Horizontal Zebra If we define

$$\mu_1 = \alpha_{11}(\theta^1) = (-V(\theta^1) - D(\theta^1))/(s_{0,0} + H(\theta^1)),$$
$$\mu_2 = \alpha_{11}(\theta^2) = (V(\theta^1) - D(\theta^1))/(s_{0,0} - H(\theta^1)),$$

then the matrix A in (3.35) can be written as

$$A_{\mathrm{HZ}}(\theta) = \begin{bmatrix} \mu_1^2 & \mu_2^2 & \mu_1^2 & \mu_2^2 \\ \mu_1 & \mu_2 & -\mu_1 & -\mu_2 \\ \mu_1 & \mu_2 & -\mu_1 & -\mu_2 \\ \mu_1^2 & \mu_2^2 & \mu_1^2 & \mu_2^2 \end{bmatrix}.$$

The symbol of the smoother becomes

$$S_{\mathrm{HZ}}(\theta) = \frac{1}{2} \begin{bmatrix} \mu_1(1+\mu_1) & & \mu_1(1+\mu_1) & \\ & \mu_2(1+\mu_2) & & \mu_2(1+\mu_2) \\ -\mu_1(1-\mu_1) & & -\mu_1(1-\mu_1) & \\ & -\mu_2(1-\mu_2) & & -\mu_2(1-\mu_2) \end{bmatrix}.$$

Vertical Zebra If we define

$$\mu_1 = \alpha_{11}(\theta^1) = (-H(\theta^1) - D(\theta^1))/(s_{0,0} + V(\theta^1)),$$
$$\mu_2 = \alpha_{11}(\theta^2) = (H(\theta^1) - D(\theta^1))/(s_{0,0} - V(\theta^1)),$$

then the matrix A in (3.35) can be written as

$$A_{\mathrm{VZ}}(\theta) = \begin{bmatrix} \mu_1^2 & \mu_2^2 & \mu_2^2 & \mu_1^2 \\ \mu_1 & \mu_2 & -\mu_2 & -\mu_1 \\ \mu_1^2 & \mu_2^2 & \mu_2^2 & \mu_1^2 \\ \mu_1 & \mu_2 & -\mu_2 & -\mu_1 \end{bmatrix}.$$

The symbol of the smoother becomes

$$S_{\mathrm{VZ}}(\theta) = \frac{1}{2} \begin{bmatrix} \mu_1(1+\mu_1) & & & \mu_1(1+\mu_1) \\ & \mu_2(1+\mu_2) & \mu_2(1+\mu_2) & \\ & -\mu_2(1-\mu_2) & -\mu_2(1-\mu_2) & \\ -\mu_1(1-\mu_1) & & & -\mu_1(1-\mu_1) \end{bmatrix}.$$

Alternating Zebra Alternating zebra smoothing consists of one horizontal zebra smoothing step followed by one vertical zebra smoothing step. The symbol of the alternating zebra smoother can be found as

$$S(\theta)_{\mathrm{AZ}} = S(\theta)_{\mathrm{HZ}} S(\theta)_{\mathrm{VZ}}.$$

We follow the suggestion of Stüben and Trottenberg as cited in [Wes92] and use a variant of vertical zebra that updates the lines with even grid

index i first. This leads, for the classical multigrid cycle, to slightly better convergence factors and identical results for $\beta = 0$ and $\beta = \frac{\pi}{2}$ when applied to the rotated anisotropic diffusion equation. We give the symbol $S_{\mathrm{VZ}'}(\theta)$ for this variant of vertical zebra smoothing.

If we define

$$\mu_1 = \alpha_{00}(\theta^1) = (-H(\theta^1) - D(\theta^1))/(s_{0,0} + V(\theta^1)),$$
$$\mu_2 = \alpha_{00}(\theta^2) = (H(\theta^1) - D(\theta^1))/(s_{0,0} - V(\theta^1)),$$

then the matrix A in (3.35) can be written as

$$A_{\mathrm{VZ}'}(\theta) = \begin{bmatrix} \mu_1 & \mu_2 & -\mu_2 & -\mu_1 \\ \mu_1^2 & \mu_2^2 & \mu_2^2 & \mu_1^2 \\ \mu_1 & \mu_2 & -\mu_2 & -\mu_1 \\ \mu_1^2 & \mu_2^2 & \mu_2^2 & \mu_1^2 \end{bmatrix}.$$

The symbol of the smoother becomes

$$S_{\mathrm{VZ}'}(\theta) = \frac{1}{2} \begin{bmatrix} \mu_1(1+\mu_1) & & & -\mu_1(1+\mu_1) \\ & \mu_2(1+\mu_2) & -\mu_2(1+\mu_2) & \\ & \mu_2(1-\mu_2) & -\mu_2(1-\mu_2) & \\ \mu_1(1-\mu_1) & & & -\mu_1(1-\mu_1) \end{bmatrix}.$$

The symbol for the alternating zebra smoothing operator is found as

$$S_{\mathrm{AZ}}(\theta) = S_{\mathrm{HZ}}(\theta) S_{\mathrm{VZ}'}(\theta).$$

Model Problems

For time dependent problems we perform the two-grid analysis on a system of the form

$$(zI - L)u = f,$$

where L is one of the discrete operators discussed in §3.2. The complex parameter z can be considered as a Helmholtz coefficient and appears in the central stencil coefficient $s_{0,0}$. This shows again that for our model problems, M depends on z through L, $\bar{L}$ and S, but not through P and R. Using the stencils (3.6)-(3.8), we can specialize the stencil (3.23) and the shorthands (3.25)-(3.27).

Isotropic Diffusion Equation

$$\begin{aligned} s_{0,0} &= z + \frac{2}{\Delta x^2} + \frac{2}{\Delta x^2} \\ H(\theta) &= -\frac{2\cos(\theta_x)}{\Delta x^2} \\ V(\theta) &= -\frac{2\cos(\theta_y)}{\Delta y^2} \\ D(\theta) &= 0 \end{aligned}$$

Anisotropic Diffusion Equation

$$\begin{aligned} s_{0,0} &= z + \varepsilon\frac{2}{\Delta x^2} + \frac{2}{\Delta x^2} \\ H(\theta) &= -\varepsilon\frac{2\cos(\theta_x)}{\Delta x^2} \\ V(\theta) &= -\frac{2\cos(\theta_y)}{\Delta y^2} \\ D(\theta) &= 0 \end{aligned}$$

Rotated Anisotropic Diffusion Equation

$$\begin{aligned} s_{0,0} &= z + (\varepsilon c^2 + s^2)\frac{2}{\Delta x^2} + (\varepsilon s^2 + c^2)\frac{2}{\Delta y^2} \\ H(\theta) &= -(\varepsilon c^2 + s^2)\frac{2\cos(\theta_x)}{\Delta x^2} \\ V(\theta) &= -(\varepsilon s^2 + c^2)\frac{2\cos(\theta_y)}{\Delta y^2} \\ D(\theta) &= -(\varepsilon - 1)cs\frac{\cos(\theta_x + \theta_y) - \cos(\theta_x - \theta_y)}{\Delta x \Delta y} \end{aligned}$$

3.5 Numerical Results

For elliptic anisotropic PDEs robust methods have been developed. We try to establish here whether the conclusions for these methods carry over to the parabolic case. We study three classical model problems: the isotropic diffusion equation ($\varepsilon = 1$, $\beta = 0$), the anisotropic diffusion equation ($\varepsilon \in \mathbb{R}$, $\beta = 0$) and the rotated anisotropic diffusion equation ($\varepsilon \in \mathbb{R}$, $\beta \in [0, 2\pi)$). The equations are discretized on a grid with spacings $\Delta x = \Delta y = h = 2^{-5}$. The implicit Euler method (BDF1) with a constant time step $\Delta t = 10^{-3}$ ($n_t = 1000$, $T = 1$) is used as ODE solver. Other time discretizations are discussed in Chapter 4. The boundary and initial conditions and the source

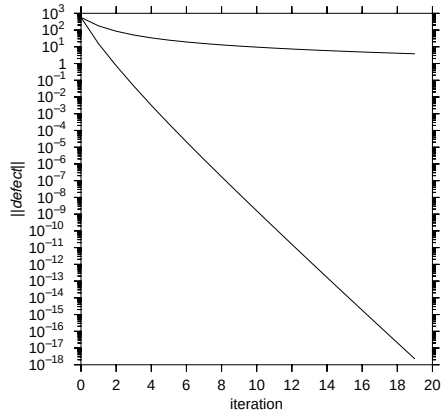


Figure 3.10: Norm of defect for single-grid waveform relaxation (upper curve, red-black Gauss-Seidel) and multigrid waveform relaxation (lower curve, red-black Gauss-Seidel smoother, full-weighting restriction, bilinear interpolation, V(1,1)-cycle). ($\varepsilon = 1$, $\beta = 0$, $h = 2^{-5}$, $\Delta t = 10^{-3}, T = 1$)

term f for (3.1) are set to 0. This corresponds to a solution $u = 0$. For the initial approximation $u^{(0)}$, a value chosen from a uniform distribution on $[-1, 1]$ is assigned to each grid point. The multigrid hierarchy has 5 levels so that the coarsest grid contains one internal point and can be solved by performing 1 smoothing step. In the numerical experiments the convergence factor is estimated by taking the quotient of the norms of the defects after the 20th and 19th iteration.

3.5.1 Isotropic Diffusion Equation

We first consider the isotropic diffusion equation, an important case both from theoretical and practical point of view. The diffusion equation is given by

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f. \tag{3.36}$$

This corresponds to (3.1) where the operator is the Laplacian (3.2).

Figure 3.10 illustrates the benefits of multigrid acceleration for this problem. The upper curve shows the norm of the defect for single-grid red-black Gauss-Seidel waveform relaxation. The lower curve shows the norm of the defect on the fine grid for multigrid waveform relaxation. We used a red-black Gauss-Seidel smoother and standard coarsening with a V-cycle using one pre- and one postsmoothing step.

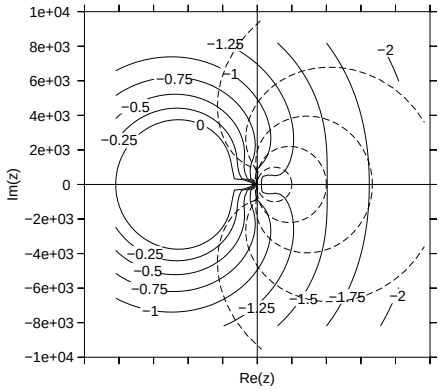


Figure 3.11: Spectral picture for two-grid waveform relaxation with red-black Gauss-Seidel as smoother and standard coarsening, applied to the discretized isotropic diffusion equation ($\varepsilon = 1$, $\beta = 0$, $h = 2^{-5}$). The solid lines represent contour lines of $\log_{10}(\rho(M(z)))$, the dashed lines represent the scaled boundary loci of the BDF1-5 methods ($\Delta t = 10^{-3}$).

Figure 3.11 illustrates the Fourier-Laplace analysis graphically. The dashed lines are the scaled boundary loci over which one has to maximize $\rho(M(z))$ to find $\rho(\mathcal{M})$ for discrete waveform relaxation on infinite time-intervals (see section 2.5.5). The contour lines of $\log_{10}\rho(M(z))$ are represented by solid lines. The convergence factor can be determined visually by finding the maximum of $\rho(M(z))$ over the appropriate set Σ. We can see for example that the convergence factor decreases as we move away from the origin. This illustrates the fact that $\rho = 0$ at infinity, corresponding to the continuous finite interval case. The convergence factor for the discrete finite interval case can be found at a point on the positive real axis.

For the analysis, the infinite interval case is chosen because in practice this gives an accurate approximation of the observed convergence factor [Van93, JV96a]. This is motivated by an argument based on pseudospectra in [LW97, JOW98]. The numerical results of the analysis are given in the tables discussed in the next two sections.

3.5.2 Anisotropic Diffusion Equation

The anisotropic diffusion equation is given by

$$\frac{\partial u}{\partial t} = \varepsilon \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f. \tag{3.37}$$

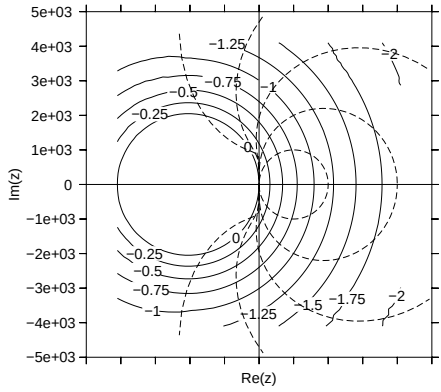


Figure 3.12: Spectral picture for two-grid waveform relaxation with red-black Gauss-Seidel as smoother and standard coarsening, applied to the discretized anisotropic diffusion equation. ($\varepsilon = 10^{-3}$, $\beta = 0$, $h = 2^{-5}$) The solid lines represent contour lines of $\log_{10}(\rho(M(z)))$, the dashed lines represent the scaled boundary loci of BDF1 to BDF5 ($\Delta t = 10^{-3}$).

It is of the form (3.1) with operator (3.3). One expects from experience with elliptic problems that point relaxation methods are inadequate for solving anisotropic problems. This is confirmed by Figure 3.12 which shows a spectral picture for an anisotropic problem. The line for $\rho = 1$ (almost) touches the origin. Therefore (3.21) will result in convergence factors very close to 1 for Σ sets corresponding to an infinite time interval. Figure 3.13 shows that when we use vertical zebra Gauss-Seidel as smoother we get a small convergence factor for $\varepsilon \ll 1$. The numerical values of the theoretical and observed asymptotic convergence factors can be found in a set of tables given below.

Table 3.1 illustrates the effect of ε. It shows the two-grid theoretical values, the two-grid observed values and the V- and W-cycle results. Large convergence factors are found for $\varepsilon \neq 1$ (red-black Gauss-Seidel smoothing and standard coarsening). In Tables 3.3 to 3.6 convergence factors for other combinations of smoothing and coarsening are given. The first column of each table indicates the smoothing and coarsening strategies used (abbreviations are explained in Table 3.2). Table 3.3 gives the theoretical convergence factor calculated by the two-grid Fourier-Laplace analysis. Table 3.4 gives convergence factors observed for two-grid waveform relaxation (the correction equation on the coarse grid is approximated by 100 smoothing steps). The observed convergence factors for multigrid waveform relaxation with V- and W-cycles are given in Tables 3.5 and 3.6. Table 3.1 compares these

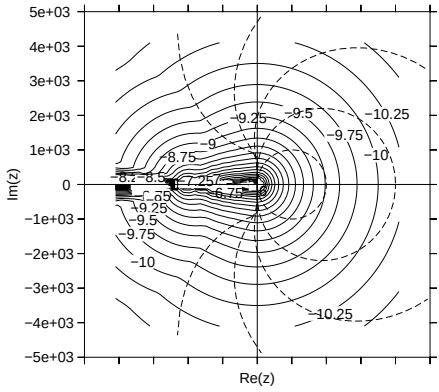


Figure 3.13: Spectral picture for two-grid waveform relaxation with vertical zebra Gauss-Seidel as smoother and standard coarsening, applied to the discretized anisotropic diffusion equation. ($\varepsilon = 10^{-3}$, $\beta = 0$, $h = 2^{-5}$) The solid lines represent contour lines of $\log_{10}(\rho(M(z)))$, the dashed lines represent the scaled boundary loci of BDF1 to BDF5 ($\Delta t = 10^{-3}$).

four cases for red-black Gauss-Seidel smoothing and standard coarsening. The observed two-grid convergence factors are very close to the theoretical two-grid convergence factor. The W-cycle is, in this case, completely equivalent to the two-grid cycle. In most cases the results for the V-cycle are comparable.

ε	10^{-4}	10^{-3}	10^{-2}	10^{-1}	2^{-1}	1	2	10^{1}	10^{2}	10^{3}	10^{4}
two-grid (th.)	0.994	0.990	0.956	0.680	0.197	0.075	0.197	0.683	0.961	0.996	1.000
V-cycle (num.)	0.936	0.931	0.890	0.648	0.210	0.108	0.209	0.649	0.898	0.937	0.937
W-cycle (num.)	0.936	0.931	0.890	0.646	0.188	0.072	0.188	0.647	0.898	0.937	0.937
two-grid (num.)	0.936	0.931	0.890	0.646	0.188	0.072	0.188	0.647	0.898	0.938	0.937

Table 3.1: Convergence factors for waveform relaxation with red-black Gauss-Seidel and standard coarsening, applied to the anisotropic diffusion equation. ($\beta = 0$, $h = 2^{-5}$, $\Delta t = 10^{-3}$, BDF1, th.=theoretical, num.=numerical)

F	red-black ($\beta = k\frac{\pi}{2}$) or four-color	S	standard coarsening
H	horizontal zebra	X	x-semicoarsening
V	vertical zebra	Y	y-semicoarsening
A	alternating zebra	A	alternating semicoarsening

Table 3.2: Abbreviations used to indicate the smoothing (first column) and coarsening (second column) strategies used. E.g., HY : horizontal zebra Gauss-Seidel with y-semicoarsening.

ε	10^{-4}	10^{-3}	10^{-2}	10^{-1}	2^{-1}	1	2	10^{1}	10^{2}	10^{3}	10^{4}
FS	0.994	0.990	0.956	0.680	0.197	0.075	0.197	0.683	0.961	0.996	1.000
HS	0.994	0.990	0.956	0.680	0.197	0.093	0.093	0.093	0.093	0.093	0.093
VS	$\sim 10^{-7}$	$\sim 10^{-4}$	0.014	0.145	0.114	0.093	0.197	0.683	0.961	0.996	1.000
AS	$\sim 10^{-7}$	$\sim 10^{-4}$	0.013	0.110	0.043	0.029	0.034	0.070	0.090	0.093	0.093
FX	0.994	0.990	0.956	0.680	0.197	0.072	0.061	0.053	0.013	0.001	$\sim 10^{-3}$
FY	0.093	0.093	0.093	0.090	0.080	0.072	0.197	0.683	0.961	0.996	1.000
FA	0.004	0.004	0.004	0.001	$\sim 10^{-4}$	$\sim 10^{-5}$	$\sim 10^{-4}$	$\sim 10^{-3}$	$\sim 10^{-3}$	$\sim 10^{-5}$	$\sim 10^{-7}$
HY	0.093	0.093	0.093	0.093	0.093	0.093	0.093	0.093	0.093	0.093	0.093
VX	$\sim 10^{-7}$	$\sim 10^{-4}$	0.014	0.145	0.114	0.093	0.072	0.053	0.013	0.001	$\sim 10^{-3}$

Table 3.3: *Theoretical* convergence factors for *two-grid* waveform relaxation with different combinations of smoothing and coarsening strategies, applied to the anisotropic diffusion equation. ($\beta = 0$, $h = 2^{-5}$, $\Delta t = 10^{-3}$, BDF1, infinite time interval)

ε	10^{-4}	10^{-3}	10^{-2}	10^{-1}	2^{-1}	1	2	10^{1}	10^{2}	10^{3}	10^{4}
FS	0.936	0.931	0.890	0.646	0.188	0.072	0.188	0.647	0.898	0.938	0.937
HS	0.936	0.931	0.889	0.646	0.188	0.087	0.086	0.079	0.048	0.002	$\sim 10^{-5}$
VS	$\sim 10^{-7}$	$\sim 10^{-4}$	0.040	0.120	0.106	0.086	0.188	0.647	0.898	0.938	0.937
AS	$\sim 10^{-7}$	$\sim 10^{-4}$	0.021	0.091	0.046	0.042	0.031	0.058	0.046	0.002	$\sim 10^{-5}$
FX	0.936	0.932	0.891	0.646	0.188	0.077	0.023	0.050	0.022	0.020	0.020
FY	0.089	0.089	0.088	0.085	0.076	0.023	0.250	0.647	0.900	0.938	0.937
FA	0.004	0.004	0.003	0.001	$\sim 10^{-4}$	$\sim 10^{-5}$	$\sim 10^{-4}$	$\sim 10^{-3}$	$\sim 10^{-4}$	$\sim 10^{-4}$	$\sim 10^{-3}$
HY	0.089	0.089	0.088	0.088	0.087	0.038	0.216	0.276	0.049	$\sim 10^{-3}$	$\sim 10^{-5}$
VX	$\sim 10^{-7}$	$\sim 10^{-4}$	0.037	0.117	0.065	0.027	0.026	0.050	0.023	0.020	0.020

Table 3.4: *Numerically observed* convergence factors for *two-grid* waveform relaxation with different combinations of smoothing and coarsening strategies, applied to the anisotropic diffusion equation. ($\beta = 0$, $h = 2^{-5}$, $\Delta t = 10^{-3}$, BDF1)

ε	10^{-4}	10^{-3}	10^{-2}	10^{-1}	2^{-1}	1	2	10^{1}	10^{2}	10^{3}	10^{4}
FS	0.936	0.931	0.890	0.648	0.210	0.108	0.209	0.649	0.898	0.937	0.937
HS	0.936	0.931	0.889	0.648	0.197	0.110	0.106	0.086	0.053	0.002	$\sim 10^{-5}$
VS	$\sim 10^{-7}$	$\sim 10^{-4}$	0.036	0.108	0.113	0.110	0.196	0.649	0.898	0.937	0.937
AS	$\sim 10^{-7}$	$\sim 10^{-4}$	0.020	0.085	0.053	0.052	0.040	0.057	0.050	0.002	$\sim 10^{-5}$
FX	0.935	0.930	0.895	0.868	0.863	0.852	0.821	0.659	0.237	0.036	0.004
FY	0.116	0.115	0.129	0.654	0.822	0.852	0.860	0.867	0.902	0.938	0.937
FA	0.011	0.011	0.009	0.004	0.035	0.034	0.030	0.011	0.003	$\sim 10^{-3}$	$\sim 10^{-4}$
HY	0.116	0.115	0.114	0.114	0.113	0.111	0.108	0.090	0.054	0.002	$\sim 10^{-5}$
VX	$\sim 10^{-7}$	$\sim 10^{-4}$	0.036	0.108	0.115	0.112	0.101	0.168	0.123	0.018	$\sim 10^{-3}$

Table 3.5: *Numerically observed* convergence factors for *multigrid* waveform relaxation with different combinations of smoothing and coarsening strategies, applied to the anisotropic diffusion equation. ($\beta = 0$, $h = 2^{-5}$, $\Delta t = 10^{-3}$, BDF1, *V-cycle*)

ε	10^{-4}	10^{-3}	10^{-2}	10^{-1}	2^{-1}	1	2	10^{1}	10^{2}	10^{3}	10^{4}
FS	0.936	0.931	0.890	0.646	0.188	0.072	0.188	0.647	0.898	0.937	0.937
HS	0.936	0.931	0.889	0.646	0.188	0.088	0.087	0.080	0.048	0.002	$\sim 10^{-5}$
VS	$\sim 10^{-7}$	$\sim 10^{-4}$	0.040	0.120	0.106	0.088	0.188	0.647	0.898	0.938	0.937
AS	$\sim 10^{-7}$	$\sim 10^{-4}$	0.021	0.091	0.046	0.042	0.030	0.058	0.045	0.002	$\sim 10^{-5}$
FX	0.936	0.931	0.890	0.704	0.571	0.480	0.369	0.050	0.010	$\sim 10^{-3}$	$\sim 10^{-4}$
FY	0.091	0.090	0.089	0.088	0.038	0.485	0.569	0.707	0.900	0.938	0.937
FA	0.005	0.004	0.004	0.001	$\sim 10^{-4}$	$\sim 10^{-5}$	$\sim 10^{-4}$	$\sim 10^{-3}$	$\sim 10^{-4}$	$\sim 10^{-6}$	$\sim 10^{-8}$
HY	0.091	0.090	0.089	0.089	0.089	0.089	0.088	0.081	0.048	0.002	$\sim 10^{-5}$
VX	$\sim 10^{-7}$	$\sim 10^{-4}$	0.040	0.121	0.106	0.088	0.069	0.049	0.010	$\sim 10^{-3}$	$\sim 10^{-5}$

Table 3.6: *Numerically observed* convergence factors for *multigrid* waveform relaxation with different combinations of smoothing and coarsening strategies, applied to the anisotropic diffusion equation. ($\beta = 0$, $h = 2^{-5}$, $\Delta t = 10^{-3}$, BDF1, *W-cycle*)

	FS	HS	VS	AS	FX	FY	FA	HY	VX
$\varepsilon \ll 1$	-	-	+	+	-	$\pm$	+	+	+
$\varepsilon \gg 1$	-	+	-	+	$\pm$	-	+	+	+

Table 3.7: Performance of multigrid waveform relaxation applied to the anisotropic diffusion equation. (+: fast, −: slow, ±: slow for all but a small range of values of ε)

For strong coupling in the x-direction ($\varepsilon \gg 1$) horizontal zebra Gauss-Seidel with standard coarsening (HS) shows good convergence. The convergence is very bad, however for strong coupling in the y-direction ($\varepsilon \ll 1$). The converse is true when we use vertical zebra Gauss-Seidel as smoother (VS). Combining these two methods into alternating zebra Gauss-Seidel (AS) results in good convergence for all values of ε. These results confirm the findings for the stationary multigrid case.

The results of the Fourier-Laplace analysis in Table 3.3 indicate that we could expect similar results for red-black Gauss-Seidel smoothing with x- and y-semicoarsening (FX, FY). The numerical results in Table 3.5, however, show poor convergence for moderate values of ε. The two-grid Fourier-Laplace analysis assumes that the correction equation on the coarse grid is solved exactly or nearly so. In the V- and W-cycle case, this assumption is not satisfied, because through grid stretching semicoarsening introduces an anisotropy that is strongly increasing with growing number of grid levels (see remark 5.1.3 on p. 134 of [TOS01]). We can therefore expect the multigrid method to fail on the coarser grids, which renders the two-grid approximation invalid. From the numerical results for V- and W-cycles it becomes clear that the semicoarsening methods only work for very specific values of ε. The alternating semicoarsening or MGS method with red-black Gauss-Seidel as smoother (FA), however, shows very good convergence for all values of ε. Combination of a smoother that works well for coupling in one direction with semicoarsening for the other direction results in robust solvers as well (rows HY and VX). These results confirm the results obtained for the stationary multigrid case. Table 3.7 summarizes our findings.

3.5.3 Rotated Anisotropic Diffusion Equation

The previous results show that good convergence can be obtained for problems with strong coupling in the direction of the coordinate axes. Tables 3.8, 3.9 and 3.10 show convergence factors for different methods as a function of the angle β, with fixed $\varepsilon = 10^{-3}$. Because of symmetry it is sufficient to consider $\beta \in [0, \frac{\pi}{2})$. We can conclude that all of the methods considered here have problems when the coupling is not aligned with the grid. Similar results are obtained for $\varepsilon \gg 1$. Again the result observed for the V-cycle is worse than expected from the two-grid Fourier-Laplace analysis. The W-cycle results are closer to the theoretical two-grid ones. The results for different smoothers and standard coarsening with a V-cycle are graphically illustrated in Figure 3.14. In this polar plot the convergence factor ρ is plotted in function of the direction β of the anisotropy. A circular curve with a small radius would indicate a robust method.

If the direction of coupling is unknown, it is best to use a method with good convergence for all values of ε in the non-rotated anisotropic case. It

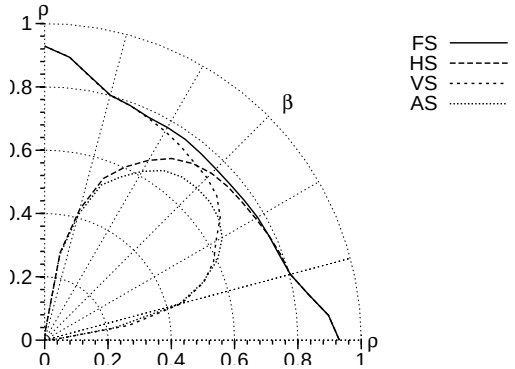


Figure 3.14: Polar plot (ρ, β) of *numerically observed* convergence factors for *multigrid* waveform relaxation with different smoothers and standard coarsening, applied to the rotated anisotropic diffusion equation. ($\varepsilon = 10^{-3}$, $h = 2^{-5}$, $\Delta t = 10^{-3}$, BDF1, *V-cycle*)

may be necessary, however, to consider other methods like incomplete LU factorization smoothing when strong non-alignment can occur. Algebraic multigrid methods can handle many types of anisotropy through their more flexible coarsening strategies.

3.5.4 Diffusion Equation with Varying Coefficients

We illustrate the results with the more general equation

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(a\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(b\frac{\partial u}{\partial y}\right) + cu + f \tag{3.38}$$

on the unit square $\Omega = [0,1]^2$ with varying coefficients $a(x,y) = e^{10(x-y)}$ and $b(x,y) = e^{-10(x-y)}$. This problem has strong coupling in both directions. The initial conditions, Dirichlet boundary conditions and source term f are chosen such that the exact solution is $u(t,x,y) = t + x + y$. We use again an equidistant rectangular grid

$$\begin{aligned} x_i &= i\Delta x, \qquad i = 0,\ldots,n_x, \qquad \Delta x = n_x^{-1}, \\ y_j &= j\Delta y, \qquad j = 0,\ldots,n_y, \qquad \Delta y = n_y^{-1}. \end{aligned}$$

Using central differences we can approximate (3.38) by

$$\dot{u}_{i,j} = c_{i,j}^{0,0}u_{i,j} + c_{i,j}^{-1,0}u_{i-1,j} + c_{i,j}^{1,0}u_{i+1,j} + c_{i,j}^{0,-1}u_{i,j-1} + c_{i,j}^{0,1}u_{i,j+1} + f_{i,j},$$

β	0°	15°	30°	45°	60°	75°	90°
FS	0.990	0.788	0.682	0.664	0.683	0.788	0.990
HS	0.990	0.788	0.667	0.611	0.541	0.335	$\sim 10^{-4}$
VS	$\sim 10^{-4}$	0.335	0.541	0.611	0.667	0.788	0.990
AS	$\sim 10^{-4}$	0.312	0.486	0.532	0.486	0.312	$\sim 10^{-4}$
FX	0.990	0.788	0.634	0.528	0.366	0.144	0.093
FY	0.093	0.144	0.364	0.530	0.637	0.788	0.990
FA	0.004	0.009	0.045	0.061	0.040	0.007	0.004
HY	0.093	0.162	0.354	0.479	0.493	0.330	$\sim 10^{-4}$
VX	$\sim 10^{-4}$	0.330	0.493	0.479	0.354	0.162	0.093

Table 3.8: *Theoretical* convergence factors for *two-grid* waveform relaxation with different combinations of smoothing and coarsening strategies applied to the rotated anisotropic diffusion equation. ($\varepsilon = 10^{-3}$, $h = 2^{-5}$, $\Delta t = 10^{-3}$, BDF1, infinite time interval)

where $i = 1, \ldots, n_x - 1$, $j = 1, \ldots, n_y - 1$, $u_{i,j} \approx u(x_i, y_j)$ and

$$c_{i,j}^{\pm 1,0} = \Delta x^{-2} a(x_i \pm \Delta x/2, y_j), \qquad c_{i,j}^{0,\pm 1} = \Delta y^{-2} b(x_i, y_j \pm \Delta y/2),$$
$$c_{i,j}^{0,0} = c(x_i, y_j) - c_{i,j}^{-1,0} - c_{i,j}^{1,0} - c_{i,j}^{0,-1} - c_{i,j}^{0,1}.$$

We can write this in a compact way using the stencil notation

$$(Lu)_{i,j} = \begin{bmatrix} & c^{0,1} & \\ c^{-1,0} & c^{0,0} & c^{1,0} \\ & c^{0,-1} & \end{bmatrix}_{i,j} u_{i,j}.$$

Packing the unknowns at the interior points of the grid into a vector of dimension $m = (n_x - 1)(n_y - 1)$ using, for example, the ordering

$$u_k = u_{i,j}, \qquad k = (i-1)(n_y - 1) + j,$$

the discrete system can be represented in the matrix form

$$\dot{u} = Lu + f, \tag{3.39}$$

with $u(t), f(t) \in \mathbb{R}^m$ and $L \in \mathbb{R}^{m \times m}$. (see Figure 2.1). The boundary conditions are incorporated in f. The matrix L is sparse and has the familiar block tridiagonal structure with tridiagonal subblocks. For varying coefficients, the matrix and its subblocks are no longer Toeplitz.

In Table 3.11 the convergence factors are given for multigrid waveform relaxation methods with different combination of smoothing and coarsening strategies. As expected only the robust methods (AS, FA, HY, VX) show

β	0°	15°	30°	45°	60°	75°	90°
FS	0.931	0.804	0.776	0.765	0.775	0.801	0.928
HS	0.931	0.804	0.769	0.745	0.656	0.428	$\sim 10^{-4}$
VS	$\sim 10^{-4}$	0.448	0.625	0.725	0.761	0.800	0.929
AS	$\sim 10^{-4}$	0.440	0.648	0.680	0.616	0.410	$\sim 10^{-4}$
FX	0.930	0.897	0.883	0.877	0.815	0.578	0.115
FY	0.115	0.625	0.816	0.874	0.889	0.892	0.928
FA	0.011	0.111	0.294	0.370	0.328	0.102	0.011
HY	0.115	0.472	0.658	0.685	0.625	0.415	$\sim 10^{-4}$
VX	$\sim 10^{-4}$	0.441	0.604	0.667	0.633	0.492	0.115

Table 3.9: *Numerically observed* convergence factors for *multigrid* waveform relaxation with different combinations of smoothing and coarsening strategies applied to the rotated anisotropic diffusion equation. ($\varepsilon = 10^{-3}$, $h = 2^{-5}$, $\Delta t = 10^{-3}$, BDF1, *V-cycle*)

β	0°	15°	30°	45°	60°	75°	90°
FS	0.931	0.766	0.697	0.688	0.696	0.767	0.929
HS	0.931	0.767	0.689	0.659	0.583	0.409	$\sim 10^{-4}$
VS	$\sim 10^{-4}$	0.410	0.571	0.648	0.686	0.766	0.929
AS	$\sim 10^{-4}$	0.422	0.582	0.606	0.555	0.398	$\sim 10^{-4}$
FX	0.931	0.853	0.799	0.732	0.585	0.232	0.090
FY	0.090	0.218	0.585	0.745	0.813	0.850	0.929
FA	0.004	0.020	0.119	0.162	0.100	0.017	0.004
HY	0.090	0.227	0.458	0.553	0.539	0.396	$\sim 10^{-4}$
VX	$\sim 10^{-4}$	0.399	0.533	0.542	0.466	0.226	0.090

Table 3.10: *Numerically observed* convergence factors for *multigrid* waveform relaxation with different combinations of smoothing and coarsening strategies applied to the rotated anisotropic diffusion equation ($\varepsilon = 10^{-3}$, $h = 2^{-5}$, $\Delta t = 10^{-3}$, BDF1, *W-cycle*).

FS	HS	VS	FX	FY
0.9207(>177)	0.9209(>178)	0.8926(>121)	0.8876(>121)	0.8867(>115)

AS	HY	VX	FA
0.0175(3)	0.1009(8)	0.1166(8)	0.0073(4)

Table 3.11: Numerically observed convergence factors for multigrid waveform relaxation applied to (3.38). Number of iteration needed for reduction of the norm of the defect by 10^{-8} between brackets. ($h = 2^{-5}$, $\Delta t = 10^{-3}$, BDF1)

good convergence. The same discretization as before is used ($\Delta x = \Delta y = 2^{-5}$, 5 levels, $\Delta t = 10^{-3}$, BDF1), together with V-cycles with one pre- and one postsmoothing step. The number in brackets indicates the number of iterations needed to get a reduction of the norm of the defect by a factor 10^{-8}. At this point precision up to discretization error was reached for all the converging methods. The convergence factor is estimated by the quotient of the norms of the defect in the last and second to last iteration. If more than 20 iterations were needed the convergence factor was estimated using the norm of the defect in the 19th and 20th iteration. This value is then used to estimate the expected number of iterations.

3.5.5 Correction for Computational Complexity

Obviously a multigrid iteration is more expensive than a single-grid iteration. Similarly line relaxation and semicoarsening are more expensive than point relaxation. To fairly compare the different methods we should introduce a convergence factor relative to the amount of work needed for one iteration. As a "work unit" one usually takes the amount of work needed to do one single-grid point relaxation on the finest grid.

The tridiagonal systems that have to be solved for line relaxation involve approximately twice the amount of work of point relaxation. The amount of work is also proportional to the number of smoothing steps ($\nu_1 + \nu_2$). For the different coarsening strategies we can estimate the amount of work by counting the number of points visited. For V-cycles this results in the factors given in table 3.12. We can now estimate, for example, that one multigrid waveform iteration with point relaxation and alternating semicoarsening with V(1,1)-cycles takes approximately $10\frac{2}{3}$ work units. For a method that has a convergence factor ρ, and takes α work units per iteration, the convergence factor per work unit is equal to $\tilde{\rho} = \sqrt[\alpha]{\rho}$. Figure 3.15 compares the corrected convergence factor for the standard coarsening multigrid waveform relaxation methods, applied to the anisotropic model problem ($\beta = 0$).

Point Relaxation	$\nu_1 + \nu_2$	Standard Coarsening	$\frac{4}{3}$
Line Relaxation	$2(\nu_1 + \nu_2)$	Semicoarsening	2
		Alternating Semicoarsening	$\frac{16}{3}$

Table 3.12: Correction factors for different smoothing and coarsening strategies (V-cycle)

The FS, VS and HS methods all have a parameter region where they work best and another region where they perform (very) poorly. The computationally expensive AS method works well over the entire parameter region. This behavior is typical also for the rotated problem and for other multigrid operator combinations.

3.6 Conclusions

We have shown that it is possible to extend the multigrid methods developed for stationary anisotropic problems to multigrid waveform relaxation methods for the corresponding time-dependent problems. The convergence rates are qualitatively similar for the stationary and the time-dependent methods. For problems where the anisotropy is aligned to the grid, alternating line relaxation with standard coarsening and point relaxation with alternating semicoarsening are appropriate methods. These methods are still useful for problems where the anisotropy is not aligned with the grid, but the performance is not optimal anymore.

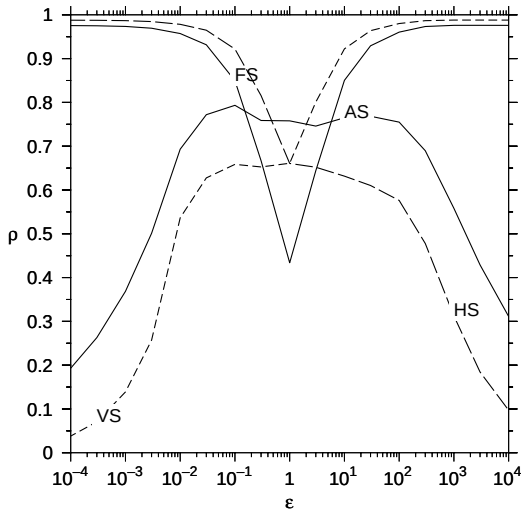


Figure 3.15: *Numerically observed* convergence factors "per work unit" for *multigrid* waveform relaxation with different smoothers and standard coarsening, applied to the anisotropic diffusion equation. ($\beta = 0$, $h = 2^{-5}$, $\Delta t = 10^{-3}$, BDF1, *V-cycle*)

Chapter 4

High Order Time Discretization Schemes

Advanced time discretization schemes for stiff systems of ordinary differential equations, such as implicit Runge-Kutta and boundary value methods, have many appealing properties. However, the resulting systems of equations can be quite large and expensive to solve. Many techniques, exploiting the structure of these systems, have been developed for general ordinary differential equations (ODE). For spatial discretizations of time-dependent partial differential equations these techniques are in general not sufficient and also the structure arising from spatial discretization has to be taken into consideration. We show here that for time-dependent parabolic problems, this can be done by multigrid methods, as in the stationary elliptic case. The key to this approach is the use of a smoother that updates several unknowns at a spatial grid point simultaneously. The overall cost is essentially proportional to the cost of integrating a scalar ODE for each grid point. Combinations of the multigrid principle with both time stepping and waveform relaxation techniques are described, together with a convergence analysis. Numerical results are presented for the isotropic heat equation and a general diffusion equation with varying coefficients.

4.1 Introduction

Large, stiff systems of ODEs are often solved using low order methods such as the implicit Euler scheme. This choice is usually dictated by implementation and computational complexity issues. Several more advanced time discretization schemes such as implicit Runge-Kutta (IRK) methods [HW96] and, more recently, boundary value methods (BVM), block boundary value

methods (BBVM) [BT98, IM98, IM99] and general linear methods (GLM) [But03] have been developed. These methods offer high orders of accuracy together with favorable stability properties, error estimators and other interesting features. Unfortunately, the linear algebra cost per time step is in general very large for these methods. When a general IRK with three stages or a BVM with three steps is applied, for example, the dimension of the linear systems is three times that of the implicit Euler case. The cost of solving these systems with standard methods is prohibitively large.

For general ODEs many specialized linear algebra methods have been developed and implemented in ODE software. These methods exploit the structure arising from carefully chosen time discretization formulae, that are, e.g., only diagonally implicit (DIRK-methods) or singly implicit (SIRK-methods) [Bur95, But03, HW96]. Since the systems arising from spatial discretization of a time-dependent partial differential equation (PDE) can be extremely large, it becomes important to use specialized methods that also exploit the structure arising from discretization in space. We will show here that for parabolic PDEs this can be done very efficiently by using multigrid methods. The methods exploiting structure arising from the time discretization formula can still be used as well, but with a good multigrid solver, this may no longer be absolutely necessary.

We consider the same type of PDEs as in Chapters 2 and 3, i.e., parabolic equations of the form

$$\frac{\partial u}{\partial t} = \mathcal{L}u + f, \tag{4.1}$$

where $\mathcal{L}$ is an elliptic operator. To allow theoretical analysis and to keep the notation concise, we assume that $\mathcal{L}$ is time independent. All algorithms presented here are, however, applicable to the more general case where $\mathcal{L}$ is time dependent. We also consider only discretization in time with a constant time step and finite difference discretizations in space on equidistant, rectangular grids with Dirichlet boundary conditions. The methods presented here can, however, be extended to three-dimensional problems and finite element discretizations on regular or irregular grids with more general boundary conditions. The geometric multigrid methods discussed in Chapters 2 and 3 are used.

Some of the results in this chapter, especially with respect to multigrid for IRK methods, have already been reported by other authors; see e.g., [Bur95, LO87]. Their results are considerably extended here and cast into the more general framework of time stepping and block time stepping with GLMs. The use of multigrid for BVM and BBVM discretizations is new. This approach enables efficient solution of the BVM and BBVM linear systems, even for very large-scale semidiscrete PDE problems.

In §4.2 we describe several high order time discretization schemes. The resulting discrete systems are solved using iterative methods. Methods

based on time stepping and waveform relaxation are presented in §4.3. Section 4.4 shows how the convergence of the iterative methods for time stepping can be analyzed. The main theoretical results are presented in §4.5 where waveform relaxation methods for block time stepping using GLMs and BBVMs are analyzed. Numerical results for two model problems, an isotropic and an anisotropic diffusion equation, are presented in §4.7. A number of different IRK and BVM schemes are considered. The corresponding theoretical multigrid convergence rates are evaluated by means of a two-grid Fourier mode analysis and verified by extensive numerical experiments.

4.2 Time Discretization Schemes

We consider methods for stiff initial value problems of the form

$$\dot{y}(t) = f(t, y(t)), \tag{4.2}$$

where $t \in \Omega_t$, $y(t) \in V$ and $y(0) = y_0 \in V$ is given. The time domain can be bounded ($\Omega_t = [0, T]$) or unbounded ($\Omega_t = [0, \infty)$). The space V can be, for example, $\mathbb{R}$ in the case of a scalar ODE or $\mathbb{R}^m$ for a system of ODEs. Time-dependent PDEs also fit this setting with V a space of functions on some spatial domain Ω.

In this section a brief description is given of linear multistep formulae, implicit Runge-Kutta methods, general linear methods and boundary value methods. For detailed information, we refer to the books [BT98, Bur95, But03, HNW93, HW96]. The stability domains of the methods are described in §4.5.1. The parameters of the schemes used in the numerical experiments, can be found in §4.2.6.

4.2.1 Linear Multistep Methods

Linear multistep methods (LMM) use solutions at several previous time steps (Figure 4.1) in a formula of the form

$$\sum_{j=-k}^{0} \alpha_{k+j} y_{i+j} = \Delta t \sum_{j=-k}^{0} \beta_{k+j} f(t_{i+j}, y_{i+j}), \tag{4.3}$$

where $y_i \approx y(t_i) \in V$, $t_i = i\Delta t$ and α_j, β_j, $j = 0, \ldots, k$ are the parameters of the method. There is an equation for every $i = k, \ldots, n$. Since only y_0 is given, the values $y_1, \ldots, y_{k-1}$ have to be obtained by other methods. The index j ranges from $-k$ to 0 to indicate the similarity with the boundary value methods formulated below and to emphasize that the set of

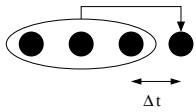


Figure 4.1: A linear multistep method uses values from several previous time steps to advance one time step.

equations (4.3) for $i = k, \ldots, n$ can be described by a banded lower triangular matrix. For stiff initial value problems, implicit LMMs ($\beta_k \neq 0$) will be used. A well-known class of implicit LMMs are the so-called backward difference formulas (BDF) [HW96, Lam73]. Note, however, that it is impossible to construct A-stable LMMs of order higher than 2 (for a definition of A-stability see §4.5.1). If $y(t) \in \mathbb{R}^m$, then in each step a system of m equations has to be solved. For the methods discussed below the systems can be much larger. Unlike the LMMs, the methods below can exhibit very good stability properties, even for high order discretizations.

4.2.2 Implicit Runge-Kutta Methods

Implicit Runge-Kutta (IRK) methods [Bur95, But03, HW96] use a number of intermediate quantities to advance one time step (Figure 4.2). These so-called stage values $\tilde{y}_i \in V^s$ are calculated from

$$\tilde{y}_i = \mathbf{1}_s y_{i-1} + \Delta t A f(\tilde{t}_i, \tilde{y}_i), \tag{4.4}$$

where $\tilde{t}_i = \mathbf{1}_s t_{i-1} + c\Delta t$, $\mathbf{1}_s = [1 \cdots 1]^T \in \mathbb{R}^s$ and f indicates componentwise application of f. The stage values $\tilde{y}_i$ are then used to calculate $y_i \in V$ using

$$y_i = y_{i-1} + \Delta t b^T f(\tilde{t}_i, \tilde{y}_i). \tag{4.5}$$

The parameters of the methods are contained in the matrix $A \in \mathbb{R}^{s\times s}$ and the vectors $b, c \in \mathbb{R}^s$ where s denotes the number of stage values. Some well-known methods are the Gauss, Radau and Lobatto methods [Bur95, But03, HW96].

If $V = \mathbb{R}^m$ all the unknowns for the stage values can be stacked into one large vector $\tilde{y}_i \in \mathbb{R}^{sm}$ and the system for the stage values and the update step can be written more explicitly using the Kronecker product notation for matrices (see §2.2). Let the first m elements of $\tilde{y}_i$ be the unknowns for the first stage and so on. The system (4.4) becomes

$$\tilde{y}_i = (\mathbf{1}_s \otimes I_m) y_{i-1} + \Delta t (A \otimes I_m) f(\tilde{t}_i, \tilde{y}_i), \tag{4.6}$$

and the update step becomes

$$y_i = (I_1 \otimes I_m) y_{i-1} + \Delta t (b^T \otimes I_m) f(\tilde{t}_i, \tilde{y}_i). \tag{4.7}$$

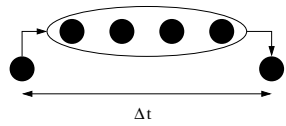


Figure 4.2: An implicit Runge-Kutta method uses intermediate values to advance one time step.

The function $f : \mathbb{R}^s \times \mathbb{R}^{sm} \to \mathbb{R}^{sm}$ returns the vectors $f([t_i]_j, [y_i]_j)$ for $j = 1, \ldots, s$ stacked into one large vector.

4.2.3 General Linear Methods

Both LMMs and IRK methods belong to the class of so-called general linear methods (GLM) [But03, HW96]. There are several ways to formulate this more general class of methods (see [But03]). We choose to use the equations

$$\tilde{y}_i = Cy_{i-1} + \Delta t A f(\tilde{t}_i, \tilde{y}_i), \tag{4.8}$$

$$y_i = Dy_{i-1} + \Delta t B f(\tilde{t}_i, \tilde{y}_i), \tag{4.9}$$

where $A \in \mathbb{R}^{s\times s}$, $B \in \mathbb{R}^{r\times s}$, $C \in \mathbb{R}^{s\times r}$ and $D \in \mathbb{R}^{r\times r}$. Both $\tilde{y}_i \in V^s$ and $y_i \in V^r$ are vectors. This formulation highlights the analogy with IRK methods. The stage values $\tilde{y}_i$ are typically approximations of $y(t)$ for some t in the current time step. The vectors y_i can contain, for example, approximations of $y(t)$, brought forward from previous time steps or scaled approximations of derivatives of $y(t)$. In order to have a stable method, D has to be power bounded. Further conditions are needed to ensure that the method is consistent. The matrices C and D are typically chosen such that D is of rank one and also $D\mathbf{1}_r = \mathbf{1}_r$ and $C\mathbf{1}_r = \mathbf{1}_s$.

For $V = \mathbb{R}^m$ the equations defining a GLM can again be given more explicitly using the Kronecker product notation. In the same way as for the IRK methods the updated values are stacked into a vector $y_i \in \mathbb{R}^{rm}$ and the stage values are stacked into a vector $\tilde{y}_i \in \mathbb{R}^{sm}$. The equations (4.8) and (4.9) become

$$\tilde{y}_i = (C \otimes I_m)y_{i-1} + \Delta t (A \otimes I_m) f(\tilde{t}_i, \tilde{y}_i), \tag{4.10}$$

$$y_i = (D \otimes I_m)y_{i-1} + \Delta t (B \otimes I_m) f(\tilde{t}_i, \tilde{y}_i). \tag{4.11}$$

The function $f : \mathbb{R}^s \times \mathbb{R}^{sm} \to \mathbb{R}^{sm}$ is defined in the same way as for the IRK methods.

Within the family of GLMs many special methods have been developed that try to combine the efficiency of LMMs and the good stability properties of IRK methods. The computational complexity of IRK methods and

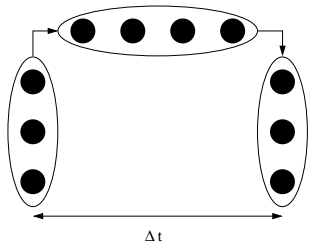


Figure 4.3: A general linear method method uses intermediate values to update several values.

GLMs depends mainly on the structure of A. In a Newton iteration for (4.8), the Jacobian is usually taken constant for all stages. An eigenvalue decomposition of A then reduces the system of order sm to s systems of order m. For general matrices A this will, however, involve using complex arithmetic. This can be avoided when A is lower triangular or diagonal. In some cases the cost can be further reduced if in addition the diagonal elements of A all have the same value. If A is diagonal, all stages can be solved in parallel. Increased efficiency can also be attained if A has only a single eigenvalue. For information and references about IRK and GLMs with special structures, see [Bur95, But03, HW96]. In the rest of the chapter, we will derive results for GLMs. The results for the subclass of IRK methods can be retrieved by taking $r = 1$, $B = b^T$, $C = \mathbf{1}_s$, and $D = 1$.

4.2.4 Boundary Value Methods

Another, more recent class of methods is the class of boundary value methods (BVM) [BT98, IM98, IM99] which can be interpreted as a generalization of the LMMs. The discretization is obtained by first assuming that the ODE is a boundary value problem and then imposing extra initial and final conditions on the values at the boundaries that are in fact unknown. A k-step BVM has for each $i = k_1, \ldots, n - k_2$ an equation of the form

$$\sum_{j=-k_1}^{k_2} \alpha_{k_1+j} y_{i+j} = \Delta t \sum_{j=-k_1}^{k_2} \beta_{k_1+j} f(t_{i+j}, y_{i+j}). \tag{4.12}$$

The main difference with the LMMs is that j ranges from $-k_1$ to k_2 ($k = k_1 + k_2$). The extra $k_1 - 1$ initial and k_2 final equations are of the form

$$\sum_{j=0}^{k} \alpha_j^{(i)} u_j = \Delta t \sum_{j=0}^{k} \beta_j^{(i)} f(t_j, u_j), \qquad i=1,\ldots,k_1-1, \tag{4.13}$$

$$\sum_{j=0}^{k} \alpha_j^{(i)} u_{n-k+j} = \Delta t \sum_{j=0}^{k} \beta_j^{(i)} f(t_{n-k+j}, u_{n-k+j}), \qquad i=n-k_2+1,\ldots,n, \tag{4.14}$$

where the coefficients $\alpha_j^{(i)}$ and $\beta_j^{(i)}$ are chosen such that the truncation errors for the initial and final conditions are of the same order as for the basic method (4.12). The matrices describing the systems involved are banded, as for LMMs, but no longer lower triangular. The n equations (4.12)-(4.14) can be written as

$$A_e y_e = \Delta t B_e f_e(t_e, y_e),$$

with $t_e \in \mathbb{R}^{n+1}$, $y_e \in V^{n+1}$ and A_e, $B_e \in \mathbb{R}^{n\times(n+1)}$. The function $f_e : \mathbb{R}^{n+1} \times V^{n+1} \to V^{n+1}$ applies f componentwise. The e subscript indicates that the known initial value is included. The matrix A_e has the following structure

$$A_e = \begin{bmatrix} \alpha_0^1 & \alpha_1^1 & \cdots & \alpha_k^1 & & & \\ \vdots & \vdots & \cdots & \vdots & & & \\ \alpha_0^{k_1-1} & \alpha_1^{k_1-1} & \cdots & \alpha_k^{k_1-1} & & & \\ & \alpha_0 & & \cdots & \alpha_k & & \\ & & \ddots & & \cdots & \ddots & \\ & & & \alpha_0 & & \cdots & \alpha_k \\ & & & \alpha_0^{n-k_2+1} & & \cdots & \alpha_k^{n-k_2+1} \\ & & & \vdots & & \cdots & \vdots \\ & & & \alpha_0^n & & \cdots & \alpha_k^n \end{bmatrix}. \tag{4.15}$$

Replacing α by β yields the structure for B_e. Using the partitions $A_e = [a_0|A]$ and $B_e = [b_0|B]$, where we split off the first columns, we can rewrite this as a system for the unknowns $y \in V^n$. We get

$$Ay = \Delta t B f(t, y) + g_0, \tag{4.16}$$

where $g_0 = -a_0 y_0 + \Delta t b_0 f(t_0, y_0)$ contains the initial condition and the function $f : \mathbb{R}^n \times V^n \to V^n$ applies f componentwise.

For $V = \mathbb{R}^m$ the unknown values can be stacked into a large vector $y \in \mathbb{R}^{nm}$ in the same way as the stage values for IRK methods. Using the Kronecker product notation we can write the system (4.16) as

$$(A \otimes I_m)y + (a_0 \otimes y_0) = \Delta t (B \otimes I_m) f(t, y) + \Delta t (b_0 \otimes f(t_0, y_0)). \tag{4.17}$$

4.2.5 Block Boundary Value Methods

We conclude this part on time discretization schemes with a method that can form an extra step in the process of transforming the continuous problem (4.2) to a discrete problem. Dividing the time interval into n subintervals leads to the sequence of subproblems

$$\begin{aligned} \dot{y}_i(t) &= f(t, y_i(t)), \quad y_i : [t_{i-1}, t_i] \to V, \quad {}_{i=1,\ldots,n}, \\ y_1(t_0) &= y_0, \quad y_i(t_{i-1}) = y_{i-1}(t_{i-1}), \quad {}_{i=2,\ldots,n}, \end{aligned}$$

where the final value of one subinterval is used as the initial condition on the next. This procedure is called time windowing or block time stepping.

Applying time windowing and a k-step BVM using s time steps on each of the subintervals yields a block boundary value method (BBVM) [BT98, IM98, IM99]:

$$Ay_i + A_0 y_{i-1} = \Delta t B f(t_i, y_i) + \Delta t B_0 f(t_{i-1}, y_{i-1}), \tag{4.18}$$

where $t_i \in \mathbb{R}^s$, $y_i \in V^s$, $A_0 = a_0 e_s^T$, $B_0 = b_0 e_s^T \in \mathbb{R}^{s\times s}$ and $e_s = [0 \cdots 0\, 1]^T \in \mathbb{R}^s$. This approach allows matrices A and $B \in \mathbb{R}^{s\times s}$ of much lower dimension. When $s \approx k$ these matrices can no longer be considered banded and the Toeplitz character is lost.

Remark 4.2.1. It is important to note that Δt has a different interpretation in GLMs and BBVMs. For BBVMs it is the distance between stage values; for GLMs it is the length of one time step, which contains s stage values. In the following remarks we consider BVMs on non-equidistant grids. Each equation can have its own time step. To keep the notation concise, we assume that the time steps are incorporated into the coefficient matrices for IRK methods, GLMs and BVMs.

Remark 4.2.2. It is straightforward to formulate IRK methods as BBVMs. Let y_i and $\tilde{y}_i$ be the values and stage values calculated by an IRK method defined by the matrix A, and the vector b. Using

$$\hat{y} = \begin{bmatrix} \tilde{y}_i \\ y_i \end{bmatrix}, \quad \text{and} \quad \hat{y}_0 = y_{i-1},$$

where we use a hat ($\hat{\cdot}$) to designate vectors and matrices related to the BVM, we get a BVM for $\hat{y}$ defined by the matrices

$$\hat{A} = \begin{bmatrix} I_s & 0 \\ 0 & 1 \end{bmatrix}, \quad \hat{a}_0 = \begin{bmatrix} -\mathbf{1}_s \\ -1 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} A & 0 \\ b^T & 0 \end{bmatrix}, \quad \hat{b}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

For IRK methods, such as the Radau IIA methods, where the last stage value of $\tilde{y}_i$ is the same as the new approximation y_i, we can use $\hat{y} = \tilde{y}_i$ and the last element of $\hat{a}_0$ and $\hat{b}_0$, and the last row and column of $\hat{A}$ and $\hat{B}$ can be dropped.

Remark 4.2.3. It is also possible to formulate BBVMs as IRK methods. Let $\hat{y}$ be the values calculated by a BBVM defined by the matrices $\hat{A}$ and $\hat{B}$, the vectors $\hat{a}_0$ and $\hat{b}_0$ and the initial value $\hat{y}_0$. If we assume that $\hat{A}^{-1}$ exists then it follows from the fact that the BBVM integrates the equation $y' = 0$ exactly that $\hat{A}^{-1}\hat{a}_0 = -\mathbf{1}_s$. Using

$$\tilde{y}_i = \begin{bmatrix} \hat{y}_0 \\ \hat{y} \end{bmatrix}, \quad \text{and} \quad y_i = \begin{bmatrix} 0 & e_s \end{bmatrix} \tilde{y}_i,$$

we get

$$A = \begin{bmatrix} 0 & 0 \\ \hat{A}^{-1}\hat{b}_0 & \hat{A}^{-1}\hat{B} \end{bmatrix}, \quad b^T = \begin{bmatrix} 0 & e_s \end{bmatrix} A.$$

Remark 4.2.4. Arbitrary GLMs can be formulated as so-called BBVMs with memory. The main difference with the IRK case is that the matrices $\hat{A}_0$ and $\hat{B}_0$ are general. If y_i and $\tilde{y}_i$ are the values and stage values calculated by a GLM defined by the coefficient matrices A, B, C and D, then using

$$\hat{y}_i = \begin{bmatrix} \tilde{y}_i \\ y_i \end{bmatrix},$$

we get a BBVM for $\hat{y}_i$ defined by the matrices

$$\hat{A} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}, \quad \hat{A}_0 = \begin{bmatrix} 0 & -C \\ 0 & -D \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} A & 0 \\ B & 0 \end{bmatrix}, \quad \hat{B}_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Remark 4.2.5. It is also possible to formulate BBVMs as GLMs. Using, for example,

$$\tilde{y}_i = y_i = \begin{bmatrix} \hat{y}_i \\ \hat{y}_{i-1} \end{bmatrix},$$

and assuming that $\hat{A}^{-1}$ exists, we get

$$A = B = \begin{bmatrix} \hat{A}^{-1}\hat{B} & \hat{A}^{-1}\hat{B}_0 \\ 0 & 0 \end{bmatrix}, \quad C = D = \begin{bmatrix} -\hat{A}^{-1}\hat{A}_0 & 0 \\ I & 0 \end{bmatrix}.$$

For more information on BBVMs and GLMs we refer to [BT98, But03, IM98, IM99].

4.2.6 Schemes used for the Numerical Experiments

We present here the coefficients of the methods used in the numerical experiments.

Implicit Runge-Kutta methods

We considered only the Radau IIA method with 3 stages.

RAD3

$$c = \begin{bmatrix} \frac{4-\sqrt{6}}{10} \\ \frac{4+\sqrt{6}}{10} \\ 1 \end{bmatrix}, \quad A = \begin{bmatrix} \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{bmatrix},$$
$$b^T = \begin{bmatrix} \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{bmatrix}$$

Boundary Value Methods

The coefficient matrices A_e and B_e for the LMMs and BBVMs we used are given below. The boldface rows correspond to the coefficients α_j and β_j of the basic method (4.12). The rows above correspond to the initial conditions (4.13) and the rows below to the final conditions (4.14). The coefficient matrices for $s > k$ are obtained by repeating the boldface rows as in (4.15).

BDF1

$$A_e = \left[\begin{array}{c|c} \mathbf{-1} & \mathbf{1} \end{array}\right], \quad B_e = \left[\begin{array}{c|c} \mathbf{0} & \mathbf{1} \end{array}\right]$$

GAM4

$$A_e = \left[\begin{array}{c|cccc} -1 & 1 & & & \\ \mathbf{-1} & \mathbf{1} & & & \\ & -1 & 1 & & \\ & & -1 & 1 & \end{array}\right],$$
$$B_e = \frac{1}{720}\left[\begin{array}{c|cccc} 251 & 646 & -264 & 106 & -19 \\ \mathbf{-19} & \mathbf{346} & \mathbf{456} & \mathbf{-74} & \mathbf{11} \\ 11 & -74 & 456 & 346 & -19 \\ -19 & 106 & -264 & 646 & 251 \end{array}\right]$$

GBDF5

$$A_e = \frac{1}{60}\left[\begin{array}{c|ccccc} -12 & -65 & 120 & -60 & 20 & -3 \\ 3 & -30 & -20 & 60 & -15 & 2 \\ \mathbf{-2} & \mathbf{15} & \mathbf{-60} & \mathbf{20} & \mathbf{30} & \mathbf{-3} \\ 3 & -20 & 60 & -120 & 65 & 12 \\ -12 & 75 & -200 & 300 & -300 & 137 \end{array}\right],$$

$$B_e = \left[\begin{array}{c|ccccc} 0 & 1 & & & & \\ 0 & & 1 & & & \\ \mathbf{0} & & & \mathbf{1} & & \\ 0 & & & & 1 & \\ 0 & & & & & 1 \end{array}\right]$$

BDF5

The last rows of the previous matrices contain the coefficients of the BDF5 method.

RAD3

The Radau IIA methods can be easily formulated as a BBVM. It is important to note that the points within a time step are not equidistant as in the previous BVMs and that the time step Δt has a different interpretation. See also §5.5.3.

$$A_e = \left[\begin{array}{c|ccc} -1 & 1 & & \\ -1 & & 1 & \\ -1 & & & 1 \end{array}\right], \quad B_e = \left[\begin{array}{c|ccc} 0 & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ 0 & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ 0 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array}\right]$$

4.3 Time Integration

In Chapter 2 we briefly explained how iterative methods and more specifically multigrid methods, can be used to solve systems of equations resulting from discretization of time-dependent PDEs. By changing the order in which the discretization and solution methods are applied, a whole variety of multigrid based solution methods for time-dependent problems can be derived. We now formulate the two families of methods we will focus on in the remainder of this chapter: time stepping and waveform relaxation. Block time stepping is mentioned as a method in between these two extremes.

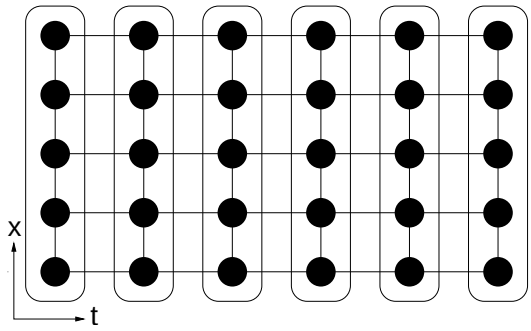


Figure 4.4: In a time stepping procedure all the values for one time step are updated simultaneously.

4.3.1 Time Stepping

If we apply any of the time discretization schemes discussed in §4.2 to a time-dependent partial differential equation, we get a so-called time stepping method (Figure 4.4). In the case of a parabolic initial value problem, the elliptic subproblems can be solved using a multigrid algorithm. One smoothing step consists of a loop over all spatial grid points, updating all unknowns for the considered time step simultaneously. For LMMs there is only one unknown per grid point and only scalar computations are required. An s-dimensional system has to be solved at each grid point in the IRK, GLM and BBVM cases. Since s is usually small, this can be done using a standard dense linear system solver. It is still possible to use eigenvalue decompositions or methods with a special structure, as described in §4.2, but this will generally not be necessary. The BVM case cannot be considered a time stepping method since the values at all time steps have to be obtained from a system of equations that is banded but not triangular or block triangular. This means that the solution components cannot be obtained in sequence; they must be computed simultaneously.

Applying to (4.1) first a time discretization from §4.2, then spatial discretization as in §2.3.1 or §3.2 and finally a splitting of the discretized elliptic operator as in §2.3.2 or §3.3, we get the following iterations. Note that all of these methods can be used as the smoother in a multigrid scheme.

For a LMM (4.3), the classical iteration (2.4), defined by the splitting

$L = L^+ + L^-$, results in

$$\begin{aligned}\sum_{j=-k}^{0} \alpha_{k+j} u_{i+j}^{(\nu)} =& \Delta t \sum_{j=-k}^{0} \beta_{k+j} L^+ u_{i+j}^{(\nu)} + \\ & \Delta t \sum_{j=-k}^{0} \beta_{k+j} L^- u_{i+j}^{(\nu-1)} + \Delta t \sum_{j=-k}^{0} \beta_{k+j} f_{i+j}.\end{aligned} \tag{4.19}$$

Note that additional methods have to be provided to approximate the initial values $u_i \in \mathbb{R}^m, i = 1, \ldots, k-1$ to turn this into a practical method.

When we apply a BVM (4.12), we get the following iteration

$$\begin{aligned}\sum_{j=-k_1}^{k_2} \alpha_{k_1+j} u_{i+j}^{(\nu)} =& \Delta t \sum_{j=-k_1}^{k_2} \beta_{k_1+j} L^+ u_{i+j}^{(\nu)} + \\ & \Delta t \sum_{j=-k_1}^{k_2} \beta_{k_1+j} L^- u_{i+j}^{(\nu-1)} + \Delta t \sum_{j=-k_1}^{k_2} \beta_{k_1+j} f_{i+j}.\end{aligned} \tag{4.20}$$

The initial and final conditions can be derived in the same way from (4.13) and (4.14). Using the matrices A and B the combined equations can be written in Kronecker product notation as

$$\begin{aligned}&(A \otimes I_m) u^{(\nu)} + (a_0 \otimes I_m) u_0^{(\nu)} = \\ &\quad \Delta t (B \otimes L^+) u^{(\nu)} + \Delta t (B \otimes L^-) u^{(\nu-1)} + \Delta t (B \otimes I_m) f + \\ &\quad \Delta t (b_0 \otimes (L^+ u_0^{(\nu)})) + \Delta t (b_0 \otimes (L^- u_0^{(\nu-1)})) + \Delta t (b_0 \otimes f_0),\end{aligned}$$

with $A,\ B \in \mathbb{R}^{n \times n}$, $a_0,\ b_0 \in \mathbb{R}^{n \times 1}$, $u^{(\nu)},\ f \in \mathbb{R}^{nm}$ and $u_0^{(\nu)},\ f_0 \in \mathbb{R}^m$.

For a GLM described by (4.8) and (4.9), the iteration becomes

$$\begin{aligned}\tilde{u}_i^{(\nu)} =\ & (C \otimes I_m) u_{i-1}^{(\nu)} + \Delta t (A \otimes L^+) \tilde{u}_i^{(\nu)} \\ & + \Delta t (A \otimes L^-) \tilde{u}_i^{(\nu-1)} + \Delta t (A \otimes I_m) \tilde{f}_i,\end{aligned} \tag{4.21}$$

$$\begin{aligned}u_i^{(\nu)} =\ & (D \otimes I_m) u_{i-1}^{(\nu)} + \Delta t (B \otimes L^+) \tilde{u}_i^{(\nu)} \\ & + \Delta t (B \otimes L^-) \tilde{u}_i^{(\nu-1)} + \Delta t (B \otimes I_m) \tilde{f}_i.\end{aligned} \tag{4.22}$$

Discretizing time using a BBVM (4.18) results in the iterative method

$$\begin{aligned}&(A \otimes I_m) u_i^{(\nu)} + (A_0 \otimes I_m) u_{i-1}^{(\nu)} = \\ &\quad \Delta t (B \otimes L^+) u_i^{(\nu)} + \Delta t (B \otimes L^-) u_i^{(\nu-1)} + \Delta t (B \otimes I_m) f_i + \\ &\quad \Delta t (B_0 \otimes L^+) u_{i-1}^{(\nu)} + \Delta t (B_0 \otimes L^-) u_{i-1}^{(\nu-1)} + \Delta t (B_0 \otimes I_m) f_{i-1}.\end{aligned} \tag{4.23}$$

The BVM discretization corresponds to taking the number of subintervals $n = 1$.

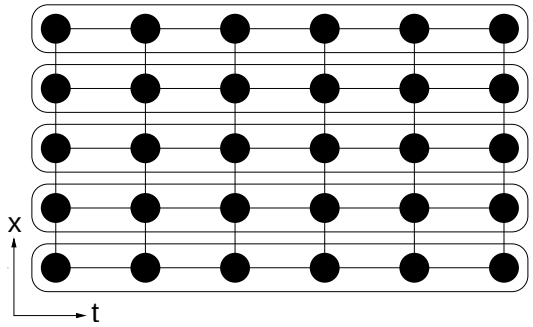


Figure 4.5: In a waveform relaxation procedure all the values for one spatial grid point are updated simultaneously.

4.3.2 Waveform Relaxation

One way of deriving waveform relaxation is to first transform (4.1) into the system of ODEs

$$\dot{u} = Lu + f \tag{4.24}$$

by discretizing only the spatial domain. This fits the framework outlined in §2.6 with $u, f : \Omega_t \to R^m$ and the operator $\mathbf{A} = \frac{d}{dt} - L$. The splitting $\mathbf{A}^+ = \frac{d}{dt} - L^+, \quad \mathbf{A}^- = -L^-$ gives us the iteration

$$\dot{u}^{(\nu)} = L^+ u^{(\nu)} + L^- u^{(\nu-1)} + f, \tag{4.25}$$

which can be used as a smoother in a multigrid algorithm. This algorithm can be defined on a time interval that is finite, $\Omega_t = [0, T]$ or infinite, $\Omega_t = [0, \infty)$. In this continuous waveform relaxation method, one step consists of the solution of m scalar ODEs, one for each grid point. Applying any of the time discretization schemes to solve these leads to a discrete waveform relaxation method (Figure 4.5). Waveform relaxation can also be considered an extreme form of block time stepping as discussed in the next section.

As mentioned in the previous section, an iterative solution procedure of waveform relaxation type is a natural choice for BVMs since all of the variables in one spatial grid point are coupled in a linear system that is not triangular. Similar waveform relaxation methods are possible for any other LMMs as well as for GLMs and BBVMs. For LMMs the subsystems are

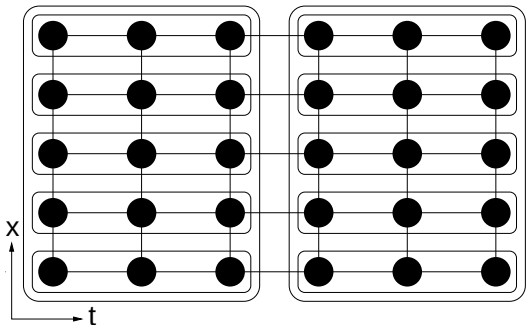


Figure 4.6: Block time stepping can be seen as intermediate between time stepping and waveform relaxation.

lower triangular and characterized by matrices A and B of the form

$$A = \begin{bmatrix} \alpha_k & & & & \\ \vdots & \ddots & & & \\ \alpha_0 & \dots & \alpha_k & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha_0 & \dots & \alpha_k \end{bmatrix}.$$

For BVMs they are banded of the form (4.15) and in the case of GLMs or BBVMs they are block lower triangular. For all these cases, the systems that arise when using a discrete waveform relaxation method can be easily solved by direct methods.

4.3.3 Block Time Stepping

An interesting modification of the time stepping schemes is to collect the equations associated with a consecutive set of time steps together into one big system of equations [HV95, SN88, Wom90]. Combining four BDF1 time steps, for example, results in a system of equations of the form

$$(A \otimes I_m)u = \Delta t(B \otimes L)u + \Delta t(B \otimes I_m)f + g_0,$$

where $u, f, g_0 \in \mathbb{R}^{4m}$, with m the number of equations in the original ODE, $B = I_4$ and

$$A = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & -1 & 1 & \\ & & -1 & 1 \end{bmatrix}.$$

Information about the previous time step is incorporated in g_0 as in (4.16). The splitting $L = L^+ + L^-$ leads to an iteration of the form

$$(A \otimes I_m)u^{(\nu)} = \Delta t(B \otimes L^+)u^{(\nu)} + \Delta t(B \otimes L^-)u^{(\nu-1)} + \Delta t(B \otimes I_m)f + g_0.$$

This can be seen as a collective update for all the unknowns on the set of time levels considered and can be executed by solving a series of small systems of equations, one 4×4 system at each grid point in this particular example. We call this type of method a block time stepping scheme (Figure 4.6). Taking this approach one step further and updating all time steps at a grid point simultaneously, results in a discrete waveform relaxation method as discussed in the previous section.

4.3.4 Remarks

The time stepping, block time stepping and waveform relaxation schemes for a linear problem on a given grid and discretization solve exactly the same equations. The corresponding multigrid methods differ only in the order in which the unknowns are updated. In the time stepping case the ODE solver forms the outer loop and a multigrid solver is used within each time step. In the waveform relaxation case, the multigrid solver forms the outer loop and the ODE solver is used within its smoother. In the block time stepping case the multigrid solver forms the outer loop. In each grid point there is a loop of over subintervals of the time domain and in each subinterval the ODE solver is used. Summarizing, we can say that a time stepping method loops over time steps, then multigrid iterations, then multigrid levels, then smoothing steps, and finally grid points. A waveform relaxation or block time stepping method loops over multigrid iterations, then multigrid levels, then smoothing steps, then grid points, and finally time steps. In all cases the cost of one multigrid iteration is equivalent to the cost of solving a scalar ODE at each grid point.

The approach discussed here has to be contrasted with other approaches for the solution of systems arising from the discretization of general ODEs. One approach to solving systems with matrices of the form $A \otimes I_m - \Delta t B \otimes L$, obtained by using BVMs, is to use circulant approximations for the matrices A and B to precondition a Krylov subspace iteration (see for example [Ber00, BN03, CNJ01, IT01]). This essentially comes down to a splitting of the matrices A and B, whereas we split the matrix L. The approaches are, of course, not mutually exclusive, but for the small subsystems that have to be solved at each grid point in our multigrid smoothers, direct methods are hard to beat. Preconditioning based on diagonal splittings of A and B is studied in [MN05]. Finally, we note that smoothers derived by directly applying the Jacobi or Gauss-Seidel methods to the system matrix give very unsatisfactory results.

4.4 Convergence Analysis for the Time Stepping Case

In this section we present a convergence analysis of the iterative methods for time stepping schemes discussed in the previous sections. We use the asymptotic convergence factor, i.e., the spectral radius of the iteration operator, as a measure for convergence. In the time stepping case the iterates are vectors and the iteration operator is a matrix. The convergence factors can therefore be derived using only linear algebra manipulations. This is an extension of the analysis for the implicit Euler method in §2.5.5. A different derivation of these results is also presented Chapter 7.

4.4.1 Single Grid Methods

For the LMM case, we obtain from (4.19) the error iteration

$$\alpha_k e_i^{(\nu)} = \Delta t \beta_k L^+ e_i^{(\nu)} + \Delta t \beta_k L^- e_i^{(\nu-1)}, \tag{4.26}$$

where the iteration errors are $e_i^{(\nu)} = u_i^{(\nu)} - u_i \in \mathbb{R}^m$. Let $\mathcal{K}_{\Delta t}$ be the corresponding iteration operator. The convergence factor is then given by

$$\begin{aligned}\rho\left(\mathcal{K}_{\Delta t}\right) &= \rho\left((\alpha_k I_m - \Delta t \beta_k L^+)^{-1} \Delta t \beta_k L^-\right) \\ &= \rho\left(\left(\frac{1}{\Delta t}\frac{\alpha_k}{\beta_k} I_m - L^+\right)^{-1} L^-\right),\end{aligned}$$

where we assumed that $\beta_k \neq 0$ and $\frac{1}{\Delta t}\frac{\alpha_k}{\beta_k} \notin \sigma(L^+)$. Introducing the operator

$$K(z) = (z I_m - L^+)^{-1} L^- \tag{4.27}$$

(see (2.15)) we can write this as

$$\rho\left(\mathcal{K}_{\Delta t}\right) = \rho\left(K\left(\frac{1}{\Delta t}\frac{\alpha_k}{\beta_k}\right)\right).$$

We can now state the following theorem.

Theorem 4.4.1. *Let $\mathcal{K}_{\Delta t}$ be the iteration operator of the iteration* (4.19) *used in the LMM time stepping method and let $\Sigma = \{\frac{\alpha_k}{\beta_k}\}$. If $\frac{1}{\Delta t}\Sigma \cap \sigma(L^+) = \phi$, then*

$$\rho(\mathcal{K}_{\Delta t}) = \rho\big(K(z)\big), \text{ with } z = \frac{1}{\Delta t}\frac{\alpha_k}{\beta_k}. \tag{4.28}$$

Note that for the Gauss-Seidel and Jacobi methods, the operator $K(z)$ is the iteration matrix for the equivalent splitting method applied to

$$zu = Lu + f, \tag{4.29}$$

which is the Laplace transform of the system of ODEs (4.24).

For block BVMs (with standard BVMs and LMMs as special cases), we can derive from (4.23) the error iteration

$$(A \otimes I_m)e_i^{(\nu)} = \Delta t(B \otimes L^+)e_i^{(\nu)} + \Delta t(B \otimes L^-)e_i^{(\nu-1)}, \tag{4.30}$$

where $e_i^{(\nu)} = u_i^{(\nu)} - u_i \in \mathbb{R}^{sm}$. Let $\mathcal{K}_{\Delta t}$ be the corresponding iteration operator. Assuming that B^{-1} exists and that $\frac{1}{\Delta t}\sigma(B^{-1}A) \cap \sigma(L^+) = \phi$, we get

$$\begin{aligned}\rho\left(\mathcal{K}_{\Delta t}\right) &= \rho\left((A \otimes I_m - \Delta t B \otimes L^+)^{-1}\Delta t(B \otimes L^-)\right) \\ &= \rho\left(\left(\frac{1}{\Delta t}B^{-1}A \otimes I_m - I_s \otimes L^+\right)^{-1}(I_s \otimes L^-)\right),\end{aligned}$$

which we can write as

$$\rho\left(\mathcal{K}_{\Delta t}\right) = \rho\left(K\left(\frac{1}{\Delta t}B^{-1}A\right)\right),$$

where formally define the matrix $K(T)$ as

$$K(T) = (T \otimes I_m - I_s \otimes L^+)^{-1}(I_s \otimes L^-).$$

From the calculation rules for Kronecker products it follows that if μ is an eigenvalue of $K(z)$ with eigenvector y and z is an eigenvalue of T with eigenvector x, then μ is an eigenvalue of $K(T)$ with eigenvector $x \otimes y$. Equivalently, using an eigenvalue decomposition $T = X\Lambda X^{-1}$ the matrix can be written as

$$K(T) = (X \otimes I_m)(\Lambda \otimes I_m - I_s \otimes L^+)^{-1}(I_s \otimes L^-)(X^{-1} \otimes I_m).$$

Both reasonings show that

$$\sigma(K(T)) = \bigcup_{z \in \sigma(T)} \sigma(K(z)).$$

This leads us to the following theorem.

Theorem 4.4.2. *Let $\mathcal{K}_{\Delta t}$ be the iteration operator of the iteration (4.23) used in the BBVM time stepping method and let $\Sigma = \sigma(B^{-1}A)$. If $\frac{1}{\Delta t}\Sigma \cap \sigma(L^+) = \phi$, then*

$$\rho(\mathcal{K}_{\Delta t}) = \max_{z \in \frac{1}{\Delta t}\Sigma} \rho(K(z)). \tag{4.31}$$

By choosing Σ an appropriate subset of $\mathbb{C}$, equation (4.31) will apply to all other time discretization methods considered here (max is replaced by sup when Σ is an infinite set).

In the IRK/GLM time stepping case, the iterative method is used to find the solutions of the equations (4.4) for the stage values. The values u_i are calculated directly from $\tilde{u}_i$ following the equations (4.4) and do not take part in the iteration. The iteration for the errors $\tilde{e}_i^{(\nu)} = \tilde{u}_i^{(\nu)} - \tilde{u}_i \in \mathbb{R}^{sm}$ becomes

$$\tilde{e}_i^{(\nu)} = \Delta t(A \otimes L^+)\tilde{e}_i^{(\nu)} + \Delta t(A \otimes L^-)\tilde{e}_i^{(\nu-1)}. \tag{4.32}$$

The convergence factor of the corresponding iteration operator $\mathcal{K}_{\Delta t}$ can be written as

$$\begin{aligned}\rho\left(\mathcal{K}_{\Delta t}\right) &= \rho\left((I_s \otimes I_m - \Delta t A \otimes L^+)^{-1}(\Delta t A \otimes L^-)\right) \\ &= \rho\left(K\left(\frac{1}{\Delta t}A^{-1}\right)\right),\end{aligned}$$

where we assume that A^{-1} exists and $\frac{1}{\Delta t}\sigma(A^{-1}) \cap \sigma(L^+) = \phi$. By the same reasoning as for the BVM case we find the following theorem.

Theorem 4.4.3. *Let $\mathcal{K}_{\Delta t}$ be the iteration operator of the iteration (4.21), used for the stage values of the GLM time stepping method and let $\Sigma = \sigma(A^{-1})$. If $\frac{1}{\Delta t}\Sigma \cap \sigma(L^+) = \phi$, then*

$$\rho(\mathcal{K}_{\Delta t}) = \max_{z \in \frac{1}{\Delta t}\Sigma} \rho(K(z)). \tag{4.33}$$

4.4.2 Multigrid Methods

All of the methods considered in the previous section can be used as a smoother inside a multigrid iteration. The convergence of the resulting methods can be analyzed by means of a two-grid local Fourier mode analysis. See §3.4 for a more detailed discussion. As an example, we show how the analysis for classical iterative methods can be extended to a two-grid time stepping method using a BBVM time discretization. The results for the other time stepping schemes are analogous.

First we recall the two-grid iteration operator (3.13) for solving (4.29)

$$M(z) = K(z)^{\nu_2}(I - P(zI_{\bar{m}} - \bar{L})^{-1}R(zI_m - L))K(z)^{\nu_1}. \tag{4.34}$$

This matrix is derived by applying the framework of §2.6 using $\mathbf{A} = zI_m - L$, $\mathbf{A}^+ = zI_m - L^+$, $\mathbf{A}^- = -L^-$, $\mathbf{R} = R$, $\mathbf{P} = P$, $\bar{\mathbf{A}} = zI_{\bar{m}} - \bar{L}$ and with $K(z)$ defined as before.

Doing the same for $\mathbf{A} = A \otimes I_m - \Delta t B \otimes L$, $\mathbf{A}^+ = A \otimes I_m - \Delta t B \otimes L^+$, $\mathbf{A}^- = -\Delta t B \otimes L^-$, $\mathbf{R} = I_s \otimes R$, $\mathbf{P} = I_s \otimes P$ and $\bar{\mathbf{A}} = A \otimes I_{\bar{m}} - \Delta t B \otimes \bar{L}$,

we derive that the iteration operator for the two-grid method using (4.23) as smoother can be written as

$$\mathcal{M}_{\Delta t} = M\left(\frac{1}{\Delta t}B^{-1}A\right).$$

The results from the previous section continue to hold for the two-grid iteration if we replace $K(z)$ by $M(z)$ and extend the condition on the set Σ to

$$\frac{1}{\Delta t}\Sigma \cap (\sigma(L^+) \cup \sigma(\bar{L})) = \phi.$$

The part involving $\sigma(L^+)$ guarantees that the smoothing symbol is well defined, the part involving $\sigma(\bar{L})$ guarantees that the coarse grid correction symbol is well defined (see (4.34)).

4.5 Convergence Analysis for the Discrete Waveform Relaxation Case

The theorems 2.5.6 and 2.5.7 show how the convergence of discrete waveform relaxation methods using LMMs can be analyzed [MN87b, JV96b]. An analysis of the IRK case was given in [LO87]. In this section we present an analysis for GLMs and BBVMs. As usual, we consider methods defined on infinite time intervals since this provides results that correspond better to numerically observed behavior. For further motivation of this approach we refer to [JV96a, MN87a, Van93]. Weighted norms or pseudospectra can also be used to obtain meaningful information about convergence. In [LW97] the convergence analysis on infinite time intervals is shown to be a limiting case of the pseudospectral analysis.

4.5.1 Stability Domains

Stability domains of time discretization schemes play an important part in the convergence analysis of discrete waveform relaxation methods. We recall the definitions of the stability domains of the methods considered here. For each method we also introduce a set $\Sigma \subset \mathbb{C}$ that we will later encounter in the convergence analysis. For the case of LMMs a close relation exists between Σ and the stability domain. We show that the same relation holds for GLMs (and therefore also for IRK methods), BVMs and BBVMs. As a bonus, this relation provides us with a convenient way to calculate points on a curve containing the boundary of the stability domain.

The linear stability of a time discretization scheme is studied using the Dahlquist test equation [HW96]

$$y' = \lambda y, \quad y(0) = y_0. \tag{4.35}$$

The stability domain consists of all $z = \Delta t\lambda \in \mathbb{C}$ for which the discrete solution is bounded. The interior of the stability domain, also called the open stability domain, consists of all z for which the discrete solution converges to 0. Only the definitions for open stability domains are given since they are simpler and sufficient for the rest of the discussion.

For a LMM (4.3) we define the polynomials

$$a(w) = \sum_{j=0}^{k} \alpha_j w^j \quad \text{and} \quad b(w) = \sum_{j=0}^{k} \beta_j w^j. \tag{4.36}$$

The common assumptions of irreducibility, consistency and zero stability are made [But03, HNW93]. Furthermore, we consider only implicit methods ($\beta_k \neq 0$). We can use the same definitions for BVMs [BT98]. Again, the methods are assumed to be irreducible, consistent and zero stable. BVMs are implicit by nature.

Definition 4.5.1. A polynomial is of type (k_1, k_2, k_3) if it has k_1 zeros inside the open unit disc, k_2 zeros on the unit circle and k_3 zeros outside the closed unit disc.

By applying each of the schemes in section 4.2 to the test equation (4.35), we obtain the following definitions.

Definition 4.5.2. The open stability domain S of a k-step LMM consists of all $z \in \mathbb{C}$ for which the polynomial $a(w) - zb(w)$ is of type $(k, 0, 0)$.

The following generalization of the stability concept to BVMs can be found in [AM95, BT98].

Definition 4.5.3. The open (k_1, k_2)-stability domain S of a BVM consists of all $z \in \mathbb{C}$ for which the polynomial $a(w) - zb(w)$ is of type $(k_1, 0, k_2)$.

Definition 4.5.4. The open stability domain S of an IRK method consists of all $z \in \mathbb{C}$ for which $|1 + b^T(z^{-1}I_s - A)^{-1}\mathbf{1}_s| < 1$ [But03, HW96].

Definition 4.5.5. The open stability domain S of a GLM consists of all $z \in \mathbb{C}$ for which the characteristic polynomial of the matrix $D+B(z^{-1}I_s-A)^{-1}C$ is of type $(s, 0, 0)$ [But03, HW96].

Definition 4.5.6. The open stability domain S of a BBVM consists of all $z \in \mathbb{C}$ for which the characteristic polynomial of the matrix $-(A - zB)^{-1}(A_0 - zB_0)$ is of type $(s, 0, 0)$ [BT98, IM99].

A LMM, GLM or BBVM is called A-stable if the left half-plane is contained in its stability domain. A BVM is called A_{k_1,k_2}-stable if the left half-plane is contained in its (k_1, k_2)-stability domain. These are desirable,

if not essential, properties when solving stiff systems of ODEs. An A-stable LMM cannot have order higher than 2. Many families of IRK methods exist that are A-stable for any order. Generalized Adams methods (GAM) and generalized backward difference formulas (GBDF) are A_{k_1,k_2}-stable BVMs. When used as BBVMs, GAMs are A-stable up to order 8 and GBDFs up to order 4. By adding a few points near the edges of the equidistant grid in a time step, A-stable methods of any order can be constructed.

In the convergence analysis of LMMs further on we make use of the set

$$\Sigma = \left\{ z \in \mathbb{C} : z = \frac{a}{b}(w), |w| \geq 1 \right\}. \tag{4.37}$$

In the BVM case we have

$$\Sigma = \{ z \in \mathbb{C} : a(w) - zb(w) \text{ is not of type } (k_1, 0, k_2) \}. \tag{4.38}$$

It is clear that in each case Σ is the complement in $\mathbb{C}$ of the open stability domain S, i.e.,

$$\Sigma = \mathbb{C} \setminus S. \tag{4.39}$$

We now introduce the sets Σ that we encounter in the convergence analysis of GLMs and BBVMs and show that (4.39) still holds.

Lemma 4.5.7. *Let S be the open stability domain of a GLM as defined by Definition 4.5.5 and let the set Σ be defined as*

$$\Sigma = \{ z \in \mathbb{C} : z^{-1} \in \sigma(A + C(wI_r - D)^{-1}B), |w| \geq 1 \}; \tag{4.40}$$

then relation (4.39) *is satisfied.*

We first prove the following lemma.

Lemma 4.5.8. *For all matrices $A \in \mathbb{C}^{s\times s}$, $B \in \mathbb{C}^{r\times s}$, $C \in \mathbb{C}^{s\times r}$, $D \in \mathbb{C}^{r\times r}$ and scalars $p, q \in \mathbb{C}$ the following statements are equivalent.*

1. $q \in \sigma(A + C(pI_r - D)^{-1}B)$,

2. $p \in \sigma(D + B(qI_s - A)^{-1}C)$,

assuming that the inverses exist ($p \notin \sigma(D)$, $q \notin \sigma(A)$).

Proof. Because of symmetry, it is enough to prove that the first statement implies the second. Using the definition of an eigenvalue, the first statement

results in the following sequence of implications:

$$\begin{aligned}
&\exists x \neq 0 : C(pI_r - D)^{-1}Bx + Ax = qx && \wedge \quad Bx \neq 0,\\
&\exists x \neq 0 : C(pI_r - D)^{-1}Bx = (qI_s - A)x && \wedge \quad Bx \neq 0,\\
&\exists x \neq 0 : (qI_s - A)^{-1}C(pI_r - D)^{-1}Bx = x && \wedge \quad Bx \neq 0,\\
&\exists x \neq 0 : B(qI_s - A)^{-1}C(pI_r - D)^{-1}Bx = Bx && \wedge \quad Bx \neq 0,\\
&\exists y \neq 0 : B(qI_s - A)^{-1}Cy = (pI_r - D)y && \wedge \quad Cy \neq 0,\\
&\exists y \neq 0 : B(qI_s - A)^{-1}Cy + Dy = py && \wedge \quad Cy \neq 0.
\end{aligned}$$

By the definition of an eigenvalue, this yields the second statement. □

Proof of Lemma 4.5.7. Using Lemma 4.5.8 we see that $z \in \Sigma$ if and only if there exists a w such that

$$w \in \sigma(D + B(z^{-1}I_s - A)^{-1}C) \quad \text{and} \quad |w| \geq 1$$

or, equivalently, $z \notin S$. □

Remark 4.5.9. Lemma 4.5.8 can also be proved by applying the identity

$$\begin{aligned}
\det\left(\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}\right) &= \det(M_{11})\det(M_{22} - M_{21}M_{11}^{-1}M_{12})\\
&= \det(M_{11} - M_{12}M_{22}^{-1}M_{21})\det(M_{22})
\end{aligned}$$

to the matrix

$$\begin{bmatrix} qI_s - A & -C \\ -B & pI_r - D \end{bmatrix}.$$

Lemma 4.5.10. *Let S be the open stability domain of a BBVM as defined by Definition 4.5.6 and let the set Σ be defined as*

$$\Sigma = \{z \in \mathbb{C} : z \in \sigma\big((B + w^{-1}B_0)^{-1}(A + w^{-1}A_0)\big); |w| \geq 1\}, \tag{4.41}$$

then relation (4.39) *is satisfied.*

We first prove the following lemma.

Lemma 4.5.11. *For all matrices $A, B, A_0, B_0 \in \mathbb{C}^{s\times s}$ and scalars $p, q \in \mathbb{C}$ the following statements are equivalent.*

1. $q \in \sigma((B_0 + pB)^{-1}(A_0 + pA))$,
2. $p \in \sigma(-(A - qB)^{-1}(A_0 - qB_0))$,

assuming that the inverses exist.

Proof. Because of symmetry, it is enough to prove that the first statement implies the second. Using the definition of an eigenvalue, the first statement results in the following sequence of implications:

$$\begin{aligned}
&\exists x \neq 0 : (B_0 + pB)^{-1}(A_0 + pA)x = qx,\\
&\exists x \neq 0 : (A_0 + pA)x = q(B_0 + pB)x,\\
&\exists x \neq 0 : (A_0 - qB_0)x = -p(A - qB)x,\\
&\exists x \neq 0 : -(A - qB)^{-1}(A_0 - qB_0)x = px.
\end{aligned}$$

By the definition of an eigenvalue, this yields the second statement. □

Proof of Lemma 4.5.10. Using Lemma 4.5.11 we see that $z \in \Sigma$ if and only if there exists a w such that

$$w \in \sigma\big(-(A - zB)^{-1}(A_0 - zB_0)\big) \quad \text{and} \quad |w| \geq 1$$

or, equivalently, $z \notin S$. □

Remark 4.5.12. Taking $w = e^{\epsilon + i\theta}$ with $\epsilon > 0$ in the definition of Σ provides a convenient way to calculate points outside the stability domain. If Σ is used in a formula of the form (4.31) and if all poles of $K(z)$ lie in S, then the maximum (supremum) can be taken over the curve defined by $\epsilon = 0$ instead of over the whole set Σ.

4.5.2 Single Grid Methods

In §2.5.5 the convergence analysis of discrete waveform relaxation using for implicit Euler discretizations was outlined. We now use Lemma 2.5.10 to derive convergence factors for the discrete waveform relaxation method on infinite time intervals using the LMM, BVM, GLM and BBVM time discretization schemes. Taking $|w| \to \infty$ shows that the results for finite time intervals are the same as for time stepping.

To apply Lemma 2.5.10 we have to prove that the convolution kernels involved are in l^1. To this end we use Theorem 2.5.11, a matrix version of Wiener's inversion theorem.

LMM

From (4.19) we find the error iteration

$$\sum_{j=-k}^{0} \alpha_{k+j} e_{i+j}^{(\nu)} = \Delta t \sum_{j=-k}^{0} \beta_{k+j} L^{+} e_{i+j}^{(\nu)} + \Delta t \sum_{j=-k}^{0} \beta_{k+j} L^{-} e_{i+j}^{(\nu-1)}.$$

By taking the discrete Laplace transform of these equations and assuming that $e_0 = \cdots = e_{k-1} = 0$, we get

$$\big(a(w)I_m - \Delta t b(w)L^+\big)e^{(\nu)}(w) = \Delta t b(w)L^- e^{(\nu-1)}(w),$$

where $e^{(\nu)}(w) = \sum_{i=k}^{\infty} e_i^{(\nu)} w^{-i}$ and a and b are the stability polynomials as defined by (4.36). The discrete Laplace transform of the operator $\mathcal{K}_{\Delta t}$, described by (4.19), is therefore

$$\begin{aligned} K_{\Delta t}(w) &= \big(a(w)I_m - \Delta t b(w)L^+\big)^{-1}\Delta t b(w)L^- \\ &= K\left(\frac{1}{\Delta t}\frac{a}{b}(w)\right). \end{aligned} \tag{4.42}$$

The following lemma is proved in [JV96b, MN87b]. We omit the proof since it is very similar to the ones given below for GLMs and BBVMs. The open stability domain S is the one defined by Definition 4.5.2.

Lemma 4.5.13. *If $\sigma(\Delta t L^+) \subset S$, then $\mathcal{K}_{\Delta t}$ is bounded in $l^p(\infty, \mathbb{C}^m)$.*

Combining Lemmas 2.5.10 and 4.5.13 with equations (4.42) and (4.37) results in the following theorem.

Theorem 4.5.14. *Consider $\mathcal{K}_{\Delta t}$ as an operator in $l^p(\infty, \mathbb{C}^m)$ with $1 \le p \le \infty$, and assume $\sigma(\Delta t L^+) \subset S$. Then,*

$$\rho(\mathcal{K}_{\Delta t}) = \sup_{z \in \frac{1}{\Delta t}\Sigma} \rho(K(z)) = \sup_{z \in \frac{1}{\Delta t}\partial\Sigma} \rho(K(z)). \tag{4.43}$$

We can take $|w| \to \infty$ and retrieve the results for time stepping and discrete waveform relaxation on finite intervals.

BVM

The convergence analysis for BVM waveform relaxation on infinite sequences can formally proceed as in the finite case (see (4.30)). As in the LMM case, we assume $e_0, \ldots, e_{k_1-1} = 0$. The error equation becomes

$$(A \otimes I_m)e^{(\nu)} = \Delta t(B \otimes L^+)e^{(\nu)} + \Delta t(B \otimes L^-)e^{(\nu-1)}, \tag{4.44}$$

where A and B are no longer matrices but infinite banded Toeplitz operators of the form

$$A = \begin{bmatrix} \alpha_{k_1} & \cdots & \alpha_k & & \\ \vdots & \ddots & & \ddots & \\ \alpha_0 & & & & \\ & \ddots & & & \end{bmatrix}, \qquad B = \begin{bmatrix} \beta_{k_1} & \cdots & \beta_k & & \\ \vdots & \ddots & & \ddots & \\ \beta_0 & & & & \\ & \ddots & & & \end{bmatrix}. \tag{4.45}$$

A relation of the form (4.31) can again be derived where the set Σ is now the spectrum of the operator $B^{-1}A$. It can be seen that this corresponds to Σ as defined by (4.38) (for more information we refer to [BT98]). The BVM case, of course, reduces to the LMM case for $k_2 = 0$.

IRK and GLM

From equations (4.21) and (4.22) we derive that the error iteration for the discrete waveform relaxation method using a GLM can be described by

$$\tilde{e}_i^{(\nu)} = (C \otimes I_m)e_{i-1}^{(\nu)} + \Delta t(A \otimes L^+)\tilde{e}_i^{(\nu)} + \Delta t(A \otimes L^-)\tilde{e}_i^{(\nu-1)}, \tag{4.46}$$

$$e_i^{(\nu)} = (D \otimes I_m)e_{i-1}^{(\nu)} + \Delta t(B \otimes L^+)\tilde{e}_i^{(\nu)} + \Delta t(B \otimes L^-)\tilde{e}_i^{(\nu-1)}. \tag{4.47}$$

Applying the discrete Laplace transform to these equations results in

$$P(w)\hat{e}^{(\nu)}(w) = Q(w)\hat{e}^{(\nu-1)}(w), \tag{4.48}$$

with

$$\hat{e}^{(\nu)}(w) = \begin{bmatrix} \tilde{e}^{(\nu)}(w) & e^{(\nu)}(w) \end{bmatrix}^T,$$

$$P(w) = \begin{bmatrix} I_s \otimes I_m - \Delta tA \otimes L^+ & -w^{-1}C \otimes I_m \\ -\Delta tB \otimes L^+ & I_r \otimes I_m - w^{-1}D \otimes I_m \end{bmatrix},$$

$$Q(w) = \begin{bmatrix} \Delta tA \otimes L^- & 0 \\ \Delta tB \otimes L^- & 0 \end{bmatrix}.$$

By eliminating $e^{(\nu)}(w)$ from (4.48) we get the following iteration for $\tilde{e}^{(\nu)}(w)$:

$$\begin{aligned} &\big[I_s \otimes I_m - (\Delta tA + w^{-1}C(I_r - w^{-1}D)^{-1}\Delta tB) \otimes L^+\big]\tilde{e}^{(\nu)}(w) \\ &\quad = \big[(\Delta tA + w^{-1}C(I_r - w^{-1}D)^{-1}\Delta tB) \otimes L^-\big]\tilde{e}^{(\nu-1)}(w). \end{aligned}$$

If we denote by $\mathcal{K}_{\Delta t}$ the operator mapping $\tilde{e}^{(\nu-1)}$ to $\tilde{e}^{(\nu)}$, then its discrete Laplace transform is

$$K_{\Delta t}(w) = K\left(\frac{1}{\Delta t}\big(A + w^{-1}C(I_r - w^{-1}D)^{-1}B\big)^{-1}\right). \tag{4.49}$$

The following lemma provides a sufficient condition for $\mathcal{K}_{\Delta t}$ to be bounded. The open stability domain S is the one defined by Definition 4.5.5.

Lemma 4.5.15. *If $\sigma(\Delta tL^+) \subset S$, then $\mathcal{K}_{\Delta t}$ is bounded in $l^p(\infty, \mathbb{C}^{sm})$.*

Proof. It is sufficient to prove that $P(w)^{-1}Q(w)$ is the discrete Laplace transform of an l_1-sequence. It is clear that both $P(w)$ and $Q(w)$ are discrete

Laplace transforms of l_1-sequences. By Wiener's theorem we have that $P(w)^{-1}$ is the Laplace transform of an l_1-sequence if

$$\det(P(w)) \neq 0 \quad \text{for } |w| \geq 1. \tag{4.50}$$

If this is true then $P(w)^{-1}Q(w)$ is the discrete Laplace transform of the convolution of two l_1-sequences and therefore the operator corresponding to (4.46) and (4.47) is bounded. Condition (4.50) can hold only if the condition

$$\det\left(\begin{bmatrix} I_s - zA & -w^{-1}C \\ -zB & I_r - w^{-1}D \end{bmatrix}\right) \neq 0 \tag{4.51}$$

holds for all $z \in \sigma(\Delta t L^+)$ and all $|w| \geq 1$. By considering the LU factorization of the matrix, we see that condition (4.51) is equivalent to the conditions

$$\det(I_s - zA) \neq 0, \tag{4.52}$$

$$\det(I_r - w^{-1}D - zB(I_s - zA)^{-1}w^{-1}C) \neq 0. \tag{4.53}$$

For z to be in S, the matrix $D + B(z^{-1}I_s - A)^{-1}C$ has to be well defined and therefore condition (4.52) is satisfied. Suppose equality holds for some $z \in \sigma(\Delta t L^+)$ in (4.53). This would mean that

$$w \in \sigma(D + zB(I_s - zA)^{-1}C)$$

with $|w| \geq 1$ and therefore $z \notin S$. This contradicts the assumption of the lemma. Hence, (4.53) is satisfied. □

Combining Lemmas 2.5.10 and 4.5.15 with equations (4.49) and (4.40) results in the following theorem.

Theorem 4.5.16. *Consider $\mathcal{K}_{\Delta t}$ as an operator in $l^p(\infty, \mathbb{C}^{sm})$ with $1 \leq p \leq \infty$, and assume $\sigma(\Delta t L^+) \subset S$. Then,*

$$\rho(\mathcal{K}_{\Delta t}) = \sup_{z \in \frac{1}{\Delta t}\Sigma} \rho(K(z)) = \sup_{z \in \frac{1}{\Delta t}\partial\Sigma} \rho(K(z)). \tag{4.54}$$

BBVM

Using (4.23) and $e_i^{(\nu)} = u_i^{(\nu)} - u_i \in \mathbb{C}^{sm}$, we find the error iteration

$$\begin{aligned}(A \otimes I_m)e_i^{(\nu)} + (A_0 \otimes I_m)e_{i-1}^{(\nu)} = \\ \Delta t(B \otimes L^+)e_i^{(\nu)} + \Delta t(B \otimes L^-)e_i^{(\nu-1)} + \\ \Delta t(B_0 \otimes L^+)e_{i-1}^{(\nu)} + \Delta t(B_0 \otimes L^-)e_{i-1}^{(\nu-1)}.\end{aligned}$$

Applying the discrete Laplace transform to these equations results in

$$\begin{aligned}\left[(A+w^{-1}A_0)\otimes I_m\right]e^{(\nu)}(w) = &\left[\Delta t(B+w^{-1}B_0)\otimes L^+\right]e^{(\nu)}(w)+\\ &\left[\Delta t(B+w^{-1}B_0)\otimes L^-\right]e^{(\nu-1)}(w).\end{aligned} \tag{4.55}$$

The discrete Laplace transform of the operator $\mathcal{K}_{\Delta t}$ described by (4.23) is therefore

$$K_{\Delta t}(w) = K\left(\frac{1}{\Delta t}(B+w^{-1}B_0)^{-1}(A+w^{-1}A_0)\right).$$

In order to apply Lemma 2.5.10, the operator $\mathcal{K}_{\Delta t}$ must be bounded. A sufficient condition is provided by the following lemma. The open stability domain S is the one defined by Definition 4.5.6.

Lemma 4.5.17. *If $\sigma(\Delta t L^+) \subset S$, then $\mathcal{K}_{\Delta t}$ is bounded in $l^p(\infty, \mathbb{C}^{sm})$.*

Proof. It is sufficient to prove that $K_{\Delta t}(w)$ is the discrete Laplace transform of an l_1-sequence. Following the same reasoning as for Lemma 4.5.15, we arrive at the condition

$$\det\left((A+w^{-1}A_0)\otimes I_m - \Delta t(B+w^{-1}B_0)\otimes L^+\right) \neq 0, \tag{4.56}$$

for $|w| \geq 1$. This condition can hold only if

$$\det\left((A+w^{-1}A_0) - z(B+w^{-1}B_0)\right) \neq 0 \tag{4.57}$$

for $z \in \sigma(\Delta t L^+)$. If we assume that equality holds, then using Lemma 4.5.11 we get

$$w \in \sigma(-(A-zB)^{-1}(A_0 - zB_0)).$$

Because $|w| \geq 1$ this would mean that $z \notin S$, which contradicts the assumption of the lemma and therefore (4.57) is true. □

Combining Lemmas 2.5.10 and 4.5.17 with equations (4.55) and (4.41) results in the following theorem.

Theorem 4.5.18. *Consider $\mathcal{K}_{\Delta t}$ as an operator in $l^p(\infty, \mathbb{C}^{sm})$ with $1 \leq p \leq \infty$, and assume $\sigma(\Delta t L^+) \subset S$. Then,*

$$\rho(\mathcal{K}_{\Delta t}) = \sup_{z \in \frac{1}{\Delta t}\Sigma} \rho(K(z)) = \sup_{z \in \frac{1}{\Delta t}\partial\Sigma} \rho(K(z)). \tag{4.58}$$

4.5.3 Multigrid Methods

The analysis of the multigrid methods proceeds in the same way as in §2.6.2 and §3.4, only the sets Σ are different. As in the time stepping case the results for the two-grid iteration can be found by substituting $M(z)$ for $K(z)$. The condition on the set Σ becomes $(\sigma(\Delta t L^+) \cup \sigma(\bar{L})) \subset S$.

LMM	GLM	BVM	BBVM
$\frac{\alpha_k}{\beta_k}$	$\sigma(A^{-1})$	$\sigma(B^{-1}A)$	$\sigma(B^{-1}A)$
$\frac{a(w)}{b(w)}$	$\sigma\left((A+C(wI_r-D)^{-1}B)^{-1}\right)$	$\frac{a(w)}{b(w)}$	$\sigma\left((B_0+wB)^{-1}(A_0+wA)\right)$

Table 4.1: The sets Σ to be maximized over in (4.31). First row: finite time intervals, second row: infinite time intervals ($|w| \geq 1$).

4.5.4 Relation to Continuous Waveform Relaxation

The asymptotic convergence factor for the continuous multigrid waveform relaxation method on infinite intervals using (4.25) as smoother, is given by

$$\rho = \sup_{z \in \bar{\mathbb{C}}^+} \rho(M(z)) = \sup_{z \in i\mathbb{R}} \rho(M(z)),$$

assuming that $(\sigma(L^+) \cup \sigma(\bar{L})) \subset \mathbb{C}^-$ [JV96a, LO87, Van93]. Therefore, when $\Sigma \subset \bar{\mathbb{C}}^+$, the convergence factor of discrete waveform relaxation is bounded by that of continuous waveform relaxation. This is the case for LMMs, GLMs and BBVMs that are A-stable and for A_{k_1,k_2}-stable BVMs.

4.6 Convergence Analysis Summary

For the multigrid methods for time-dependent problems described earlier, the asymptotic convergence factor can be found from

$$\rho = \sup_{z \in \frac{1}{\Delta t}\partial\Sigma} \rho(M(z)). \tag{4.59}$$

The appropriate sets Σ for different time discretization schemes are listed in Table 4.1. The first row refers to time stepping and discrete waveform relaxation on finite intervals; the second row refers to discrete waveform relaxation on infinite intervals. Note that except for the BVM case the first row can be derived by taking $|w| \to \infty$. The function $\rho(M(z))$ can be estimated by a local mode Fourier analysis of the two-grid iteration applied to the elliptic equation

$$zu = \mathcal{L}u + f.$$

4.7 Numerical Results

The time discretization schemes used for the numerical experiments are implicit Euler (=BDF1=GBDF1=RadauIIA1), the backward difference formula with 5 steps (BDF5), the Radau IIA implicit Runge-Kutta method

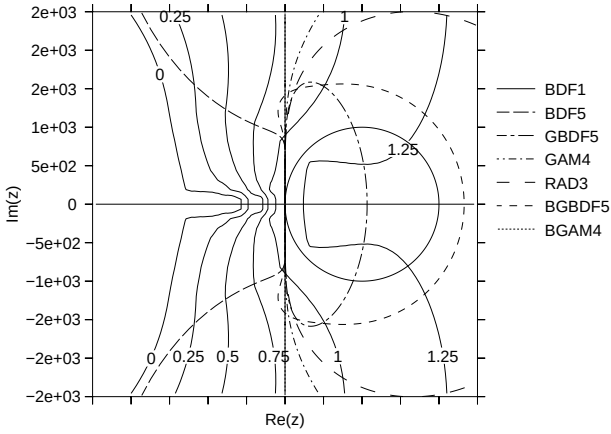


Figure 4.7: Contour lines of the convergence rate $R(z) = -\log_{10}\rho(M(z))$ and boundaries of the sets Σ for discrete waveform relaxation on infinite time domains. The boundary for the BGAM4 discretization coincides with the imaginary axis.

	BDF1	BDF5	GBDF5	GAM4	RAD3	BGBDF5	BGAM4
TS	1.29	1.32	-	-	1.16	0.92	1.08
WR	1.13	0.06	0.90	0.86	0.89	0.85	0.79

Table 4.2: Convergence rates obtained by a two-grid local Fourier analysis (heat equation).

with 3 stages (Radau IIA (RAD)3), the generalized Adams methods with 4 steps (GAM4) and the generalized backward difference formula with 5 steps (GBDF5). All but the implicit Euler method are fifth order methods. The GAM4 and GBDF5 boundary value methods were also used as block BVMs (block generalized Adams method (BGAM)4, block generalized backward difference formula (BGBDF)5). The coefficients for the discretization schemes were given in §4.2.6.

4.7.1 Isotropic Diffusion Equation

As a first example we consider the heat equation

$$u_t = u_{xx} + u_{yy} + f \tag{4.60}$$

on the unit square. The problem is discretized on a grid with $\Delta x = \Delta y = 2^{-5}$. The two-grid analysis was conducted for full weighting restriction, bi-

	BDF1	BDF5	GBDF5	GAM4	RAD3	BGBDF5	BGAM4
TS	1,1	1,5	-	-	1,3	1,5	1,4
WR	1020,1	1020,1	1020,1	1020,1	360,3	204,5	255,4

Table 4.3: Number of time steps and number of unknowns per time step for the time discretizations used in the numerical experiments.

linear interpolation and 1 pre- and 1 postsmoothing red-black Gauss-Seidel step. Time integration is over $\Omega_t = [0, 1]$; i.e., $\Delta t = 3 \cdot 10^{-3}$ for the RAD3 discretization and $\Delta t = 10^{-3}$ for all others. Figure 4.7 shows contour lines of the predicted convergence rate $R(z)$ together with the boundaries of the regions Σ of the discretization schemes used in waveform relaxation on infinite time domains. We expect convergence for the BDF5 discretization to be slow since the boundary of the corresponding set Σ nearly touches the line $R = 0$. If these lines were to touch or intersect, convergence would no longer be guaranteed. Note that all but the BDF5 and BGBDF5 method are A-stable. The BGBDF5 method is, however, very close to being A-stable and we expect very good convergence. The results are presented in Table 4.2. The first row contains the results for time stepping or discrete waveform relaxation on a finite domain. These results are obtained by maximizing $R(z)$ over Σ, which, in this case, is a discrete set of points. The BVMs (GAM4 and GBDF5) cannot be used in a time stepping sense. The second row of Table 4.2 contains the results for discrete waveform relaxation on infinite time domains. The results were obtained by maximizing over a number of points on the curves depicted in Figure 4.7. The points were found by taking $w = \exp(2\pi ik/n)$, $k = 0, \ldots, n-1$, $n = 200$ in the definitions of the sets Σ (see Remark 4.5.12). Since $\Sigma = \bar{\mathbb{C}}^+$ for the BGAM4 discretization, the last result in the table provides a good approximation to the convergence rate for continuous waveform relaxation. From these results we expect good convergence for all methods except waveform relaxation with BDF5 time discretization.

Decreasing Δt corresponds to a proportional increase in the size of the sets Σ in Figure 4.7. Similarly, when Δx and Δy are decreased simultaneously, the contour lines of $\rho(M(z))$ have approximately the same shape but on a larger scale (the dependency is quadratic in this case). This can be explained by observing that scaling L, L^+ and L^- in the formulae for $K(z)$ and $M(z)$ is equivalent to the inverse scaling of z. Decreasing Δx and Δy will, of course, also increase the size of all the matrices involved, but this can be neglected for fine discretizations (large matrices). Figure 4.7 can thus be used to estimate the convergence for other discretization resolutions by changing the relative size of the curves depicted.

For the numerical experiments the initial conditions, the Dirichlet bound-

	BDF1	BDF5	GBDF5	GAM4	RAD3	BGBDF5	BGAM4
TS	1.18	1.11	-	-	1.33	1.11	1.10
WR	0.92	0.03	0.87	0.87	0.87	0.87	0.88

Table 4.4: Convergence rates obtained from numerical experiments (heat equation).

ary conditions and the source term f are chosen such that the exact solution is $u = 0$. We use a standard W-cycle geometric multigrid algorithm with 5 levels, full weighting restriction, bilinear interpolation and 1 pre- and 1 postsmoothing red-black Gauss-Seidel step. The spatial domain is discretized using finite differences. The grid sizes are $\Delta x = \Delta y = 2^{-1}$ on the coarsest grid and $\Delta x = \Delta y = 2^{-5}$ on the finest grid. As in the analysis the time steps are $\Delta t = 3 \cdot 10^{-3}$ for RAD3 and $\Delta t = 10^{-3}$ otherwise. The number of time steps for each method was chosen such that the number of unknowns per spatial grid point was minimal in the time stepping case and the same for all methods in the waveform relaxation case. Table 4.3 shows the number of time steps and the number of unknowns per time step for each method. All unknowns were set to 1 for the initial approximation.

The results are presented in Table 4.4. It is clear that they correspond very well to the results obtained from the theoretical analysis. The theoretical convergence rates are not strictly lower bounds (upper bounds on the convergence factors), since they are asymptotic convergence rates obtained using a local Fourier mode two-grid approximation. Similar results with poor convergence for the BDF5 method and convergence rates close to 1 for the other methods were obtained for waveform relaxation on grids of dimensions $2^5 \times 2^5 \times 10^4$, $2^8 \times 2^8 \times 360$ and $2^9 \times 2^9 \times 60$. We can conclude that, for this problem, multigrid methods are very efficient even when using high order time discretization schemes.

4.7.2 Diffusion Equation with Varying Coefficients

The next example is a more general diffusion equation of the form

$$\begin{aligned} u_t &= (au_x)_x + (bu_y)_y + f, \\ a(x, y, t) &= \exp(10(x - y)), \\ b(x, y, t) &= \exp(-10(x - y)). \end{aligned}$$

This is an anisotropic problem since the diffusion coefficients in a certain point can be very different depending on the direction of diffusion. Standard multigrid methods do not handle this type of problem well and yield very slow convergence (e.g. $R = 0.1$). However, as we saw in Chapter 3, we can

	BDF1	BDF5	GBDF5	GAM4	RAD3	BGBDF5	BGAM4
TS	1.50	1.34	-	-	1.91	1.34	1.47
WR	1.22	0.69	1.22	1.22	1.22	1.22	1.22

Table 4.5: Convergence rates for the MGS iteration (diffusion equation with varying coefficients).

extend multigrid methods developed for time-independent anisotropic problems to time-dependent ones. Here, we used the MGS method, described in §3.3.4, which uses the same simple smoothers as before, together with an extended hierarchy of coarse grids.

The same discretizations, initial and boundary conditions and initial approximation were used as for the previous example. The multigrid algorithm uses full weighting restriction, bilinear and linear interpolation and 1 pre- and 1 postsmoothing step. The smoothing consists of a multigrid semicoarsening step in the x-direction followed by one in the y-direction, both using only 1 presmoothing red-black Gauss-Seidel step. A hierarchy of 5×5 grids is used. The results are presented in Table 4.5. The convergence history of the MGS waveform relaxation method is visualized in Figure 4.8 for different time discretization schemes. Except for the BDF5 method, the convergence histories are indistinguishable. We see once more that it is possible to obtain very efficient multigrid methods even when high order time discretization schemes are used.

4.8 Conclusions

We have shown by theoretical analysis and numerical experiments that it is possible to develop efficient multigrid methods for high order time discretizations of parabolic equations. The analysis shows that the stability of the time discretization scheme is important for the convergence of multigrid methods. Among schemes with good stability properties (such as A-stability), the convergence is comparable. Depending on the problem at hand, a suitable scheme can be chosen taking into account other properties such as implementation cost, efficiency and availability of error estimators. Since the cost of a multigrid iteration in a time stepping or block time stepping scheme is proportional to the cost of solving a sequence of scalar ODEs over a time step or over a block of time steps, it suffices to consider the properties of the time discretization scheme when applied to scalar equations. Furthermore, the convergence for parabolic problems is roughly equivalent to the convergence of the corresponding multigrid methods for elliptic equations. This means that both theoretical analysis and experience with elliptic

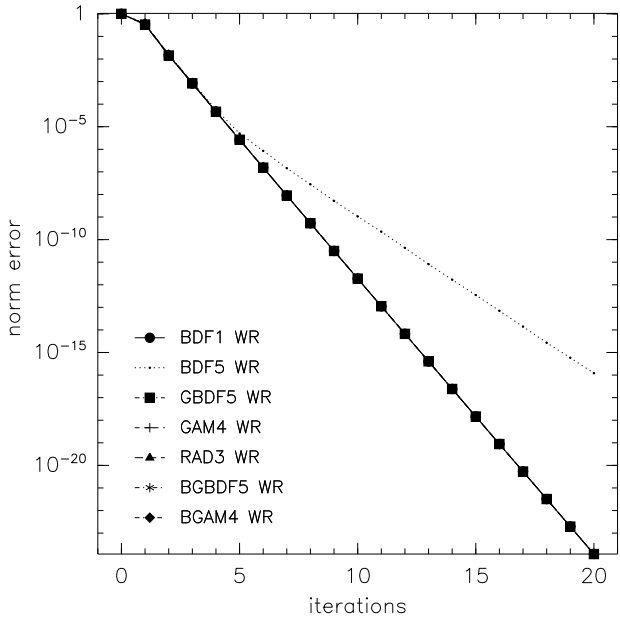


Figure 4.8: Convergence of MGS method (diffusion equation with varying coefficients).

equations can be used as guidelines to determine whether it is worthwhile to invest in the implementation of a multigrid method. Experience with the elliptic case also gives an indication of how the multigrid method will compare to other iterative methods such as GMRES or BiCGstab. It is also possible to use multigrid as a preconditioner for these methods [WOW00].

Chapter 5

Spectral Time Discretization Schemes

In this chapter we study the use of spectral methods for the time discretization of parabolic PDEs.

5.1 Introduction

In the previous chapter we have shown how high order time discretization schemes can be used to solve parabolic PDEs. The key is to use a multigrid method with a smoother that updates a block of unknowns simultaneously. If BVMs are used for the time discretization a band solver is used to solve the systems at each grid point. For BBVMs and IRK methods, a dense solver can be used. For equidistant time meshes BBVMs are A-stable only for low orders. Better stability properties can be obtained by choosing an appropriately clustered mesh for the BBVMs. It is well known that IRK methods exist that are A-stable for any order. These methods can be interpreted as polynomial collocation methods on non-equidistant meshes.

In this chapter we investigate spectral collocation methods. These methods also use clustered grids to obtain high accuracy and good stability properties. Furthermore, the matrices describing such methods are very easy to generate. As for BBVMs and IRK methods, full use can be made of highly optimized dense matrix solvers. For Chebyshev and Fourier spectral collocation it is even possible to use Fourier transforms to solve the systems. We concentrate here on Chebyshev spectral collocation.

5.2 Spectral Methods

There are two main types of spectral methods. *Non-interpolating* spectral methods represent an approximation to a function by the coefficients of an expansion in certain basis functions. *Interpolating spectral* methods use the values of the approximated function at specific mesh points. Non-interpolating and interpolating spectral methods are also called *spectral* and *pseudospectral* methods. For many types of problems and methods a mathematically equivalent formulation can be given using either perspective. Furthermore, many operations on spectral approximations, such as differentiation, integration and convolution, will be implemented by transforming to the most convenient representation, doing the operation and transforming back. A non-interpolating representation is said to belong to the frequency domain. An interpolating representation is said to belong to the physical domain. Differentiation of a periodic function given by its values at equidistant points, for example, can be performed by using a Fourier transform to go from the physical to the frequency domain, multiplying each of the resulting coefficients by a scalar and transforming back to the physical domain.

In this chapter we use the approach of [Tre00] based on interpolation. A spectral method based on collocation at the zeros of Chebyshev polynomials can be found in [Wri64]. Relationships between collocation methods and IRK methods are considered in [Wri70]. For other introductions to spectral and pseudospectral methods see, for example, [For96, GO77, Boy01]. A method using spectral methods in space and time for parabolic problems is described in [TE89].

Interpolating spectral methods approximate derivatives as follows: given a function v, choose points x_j and construct the interpolating polynomial p such that $p(x_j) = v(x_j)$. The derivative $v'(x_j)$ can now be approximated by $w_j = p'(x_j)$. Since we are using interpolation to approximate the derivatives, the choice of the interpolation points is very important. More points are necessary near the boundary to constrain the interpolating polynomial. Figure 5.1 illustrates the oscillations that may be present when using high order interpolation on an equidistant grid. The behavior of the interpolant on the Chebyshev mesh (5.1) is much better. For an explanation based on potential theory see [Tre00].

5.3 Chebyshev Spectral Differentiation

Common choices for the interpolation points are the abscissae for Gaussian quadrature formulae such as the Gauss-Legendre, Gauss-Legendre-Radau, Gauss-Legendre-Lobatto, Gauss-Chebyshev, Gauss-Chebyshev-Radau and

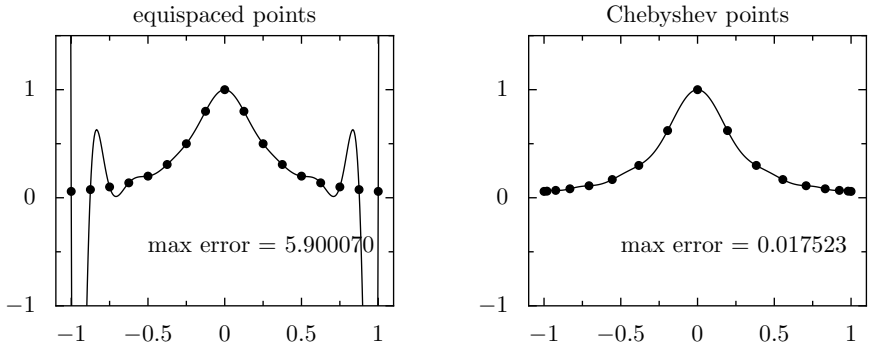


Figure 5.1: The function $(1+16x^2)^{-1}$ interpolated by polynomials of degree 16 on equidistant and Chebyshev meshes.

Gauss-Chebyshev-Lobatto points. The Gauss points do not include the boundary points of the interval. The Radau points include one boundary point and the Lobatto points include both boundary points.

We use here Chebyshev points since this leads to simple expressions as well as the possibility to use fast transform methods. More specifically we use the Gauss-Chebyshev-Lobatto points. The use of Legendre points provides a higher order approximation at the boundary points, but it is not obvious whether this is a significant advantage.

The Gauss-Chebyshev-Lobatto points are given by

$$x_j = \cos\frac{j\pi}{n}, \qquad j = 0, \ldots, n. \tag{5.1}$$

These points are also called the Chebyshev extreme points since they are the locations of the extrema of the Chebyshev polynomial of degree n in the interval $[-1, 1]$. Figure 5.2 is a graphical representation of (5.1).

Since spectral differentiation is a linear operation, the mapping from the vector $\bar{v}$, containing the values v_j, to the vector $\bar{w}$, containing the values w_j, can be represented by a matrix $\bar{D}$. In the case of spectral differentiation using the Chebyshev extreme points, the matrix has a very simple explicit form [Tre00]. For each $n \geq 1$ the diagonal entries of the Chebyshev differentiation matrix $\bar{D} \in \mathbb{R}^{(n+1)\times(n+1)}$ are given by

$$\bar{D}_{1,1} = \frac{2n^2+1}{6}, \quad \bar{D}_{n+1,n+1} = -\frac{2n^2+1}{6},$$
$$\bar{D}_{j+1,j+1} = \frac{-x_j}{2(1-x_j^2)}, \quad \text{for } j = 1, \ldots, n-1.$$

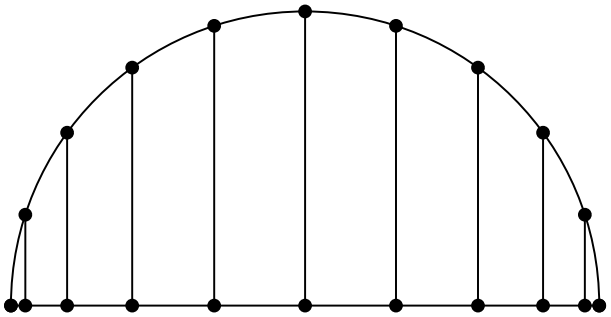


Figure 5.2: The Gauss-Chebyshev-Lobatto points.

The off-diagonal elements are given by

$$\bar{D}_{i+1,j+1} = \frac{1+\delta_i+\delta_{n-i}}{1+\delta_j+\delta_{n-j}} \frac{(-1)^{i+j}}{x_i - x_j}, \quad i \neq j, \quad i,j = 0,\ldots,n,$$

where $\delta_0 = 1$ and $\delta_i = 0$ for $i \neq 0$. The diagonal elements of $\bar{D}$ can also be calculated as

$$\bar{D}_{i,i} = -\sum_{i \neq j} \bar{D}_{i,j}, \tag{5.2}$$

which follows from the fact that the derivative of a constant function is zero and therefore $\bar{D}\mathbf{1}_{n+1} = 0$. The matrix derived using (5.2) has better stability properties in the presence of rounding errors [Tre00]. For differentiation matrices for other choices of interpolation points, see [Tre00, For96].

Spectral differentiation corresponds to multiplication by $\bar{D}$. Spectral integration corresponds to the inverse operation. It is clear that the result of integration is only unique up to a constant, since each constant vector is an eigenvector of $\bar{D}$ with eigenvalue 0. In the case of Chebyshev interpolation points, spectral differentiation and integration can be done in $O(n \log n)$ operations using fast Fourier transforms (FFT) or discrete cosine transforms (DCT) [Tre00, GO77, Boy01].

In the following section, we will need a Chebyshev differentiation matrix for functions defined on the interval $[0, \Delta t]$ instead of $[-1, 1]$. The corresponding Chebyshev points can be obtained by shifting and scaling the interpolation points x_j to

$$t_j = \frac{1 - x_j}{2} \Delta t.$$

This linear transformation results in a new differentiation matrix, given by

$$\tilde{D} = -\frac{2}{\Delta t} \bar{D}.$$

5.4 Chebyshev Spectral Collocation for ODEs

We can now use the spectral differentiation matrix to discretize ODEs. Consider the problem

$$v'(t) = f(v(t)), \qquad \text{with } v(0) = v_0,$$

on the interval $[0, \Delta t]$. The unknown function v can be approximated by a polynomial interpolating at the points t_j. The ODE is then approximated by the system of equations

$$\tilde{D}\tilde{v} = f(\tilde{v}),$$

The matrix-vector product can be partitioned as

$$\tilde{D}\tilde{v} = \left[\begin{array}{c|c} \cdot & \\ \hline d_0 & D \end{array}\right] \left[\begin{array}{c} v_0 \\ \hline v \end{array}\right]. \tag{5.3}$$

The vector v denotes the approximation to the vector of solution values

$$\left[\begin{array}{ccc} v(t_1) & \cdots & v(t_n) \end{array}\right]^T.$$

Since v_0 is given, the first row of (5.3) can be ignored and the system of equations to solve becomes

$$Dv = -d_0 v_0 + f(v). \tag{5.4}$$

In general, this system of equations is solved using a Newton iteration which requires a linearization of the function f. For the stability analysis and numerical experiments further on in this chapter, we assume, like in the previous chapters, that f is a linear function.

Fast transform techniques can be derived to solve systems with matrix D efficiently [GO77]. If an efficient procedure for spectral integration is already available, for example, applying it twice is sufficient to solve a linear system with matrix D. It is also possible to transform to the frequency domain, solve a tridiagonal system and transform back to the physical domain [GO77, ISW92]. It should be noted that it is not straightforward to determine in which cases fast transform methods will outperform methods based on dense matrix algorithms. The choice will depend on the size of the systems, the implementation and the hardware.

So far we have explained how to use spectral collocation to solve a single scalar ODE. Chebyshev spectral collocation can also be used to solve systems of ODEs, but except for very small n, i.e., low orders of accuracy, solving the linear systems using direct solvers becomes prohibitively expensive. Furthermore, the use of fast transforms is no longer straightforward. For systems of ODEs especially large systems, iterative methods have to be

considered. As we saw in the previous chapters, multigrid methods for discretized parabolic equations essentially only need a solver for scalar ODEs. Using high order methods, such as Chebyshev spectral collocation, therefore poses no problem, as will be explained and shown in §5.7.

5.5 Relation to Other Methods

We have already discussed LMMs, IRK methods, BVMs, BBVMs and GLMs in Chapter 4. In the previous section we introduced Chebyshev spectral collocation (CSC). Some other closely related methods are continuous and discontinuous Galerkin methods and differential quadrature methods. It is well known that the Gauss, Radau IIA and Lobatto IIIA IRK methods can be interpreted as polynomial collocation methods [HW96, But03]. A k-step BVM can be interpreted as spline collocation method with a spline of order k [MST]. For BBVMs the number of steps in an interval is often equal to k, which means that these methods can also be considered polynomial collocation methods. Some methods can be shown to be mathematically equivalent. We will highlight some specific relations here.

5.5.1 CSC as an IRK Method

Chebyshev spectral collocation can be applied to subintervals as is done for BVMs to obtain BBVMs. The resulting method is essentially an IRK method as well as a BBVM with a non-uniform mesh.

To see that block CSC is an IRK method we consider equation (5.4) to calculate s solution values in one time step

$$d_0 v_0 + Dv = f(v),$$

where d_0, $v \in \mathbb{R}^s$ and $D \in \mathbb{R}^{s\times s}$. Multiplying by D^{-1} results in

$$v = -D^{-1} d_0 v_0 + D^{-1} f(v).$$

Because $\tilde{D}\mathbf{1}_{s+1} = 0$, and thus $-D^{-1}d_0 = \mathbf{1}_s$, these equation are of the same form as the IRK system

$$v = v_0 + \Delta t A f(v).$$

As in the Radau IIA and Lobatto IIIA formulae, the last stage value is also the new approximation for the solution.

The classical Gauss, Radau and Lobatto methods are based on the Legendre family of points. It is clear that methods based on the Chebyshev family of points are another possibility. Methods based on Legendre points

have a higher order of approximation at end points, but this is not necessarily an important advantage, especially when dealing with stiff problems [BT98, Boy01, HW96, But03]. Fast transforms for Legendre spectral methods are much more complicated, hence, direct solvers for the linear systems arising from (5.4) are usually more efficient. Note that even for Chebyshev methods dense matrix methods are often more efficient than methods based on fast transforms. The choice of method depends very much on the size of the systems to be solved as well as on the quality of the implementation and the type of hardware used.

5.5.2 CSC as a BVM

In Chapter 4 we considered only BVM on uniform meshes. However, these methods can be formulated for any mesh [BT98]. For a BVM on an arbitrary mesh, every equation corresponds to different time step $\Delta t_i = t_i - t_{i-1}$. The equations can be written compactly as

$$a_0 v_0 + Av = Hb_0 f(v_0) + HBf(v), \tag{5.5}$$

where the different time steps are incorporated in the scaling matrix

$$H = \begin{bmatrix} \Delta t_1 & & \\ & \ddots & \\ & & \Delta t_s \end{bmatrix}.$$

In [BT98] it is shown that by introducing a few non-uniformly spaced points near the endpoints of the time interval, it is possible to construct A-stable BVMs of any order. By using a mostly uniform mesh the resulting matrices are mostly Toeplitz, which can be an advantage for computation. A natural next step, however, is to use completely non-uniform meshes. The n-step GBDF (corresponding to (5.5) with $b_0 = 0$ and $B = I$) on the Gauss-Chebyshev-Lobatto points is mathematically equivalent to CSC given by (5.4) through the identities

$$d_0 = H^{-1} a_0 \quad \text{and} \quad D = H^{-1} A.$$

It should be noted that the methods to calculate coefficients of BVMs, finite difference methods and spectral methods on general grids are very similar [BT98, For96, For98, Bri03].

5.5.3 IRK Methods as BVMs

The well known Gauss, Radau IIA and Lobatto IIIA methods can be interpreted as polynomial collocation at the Gauss-Legendre, Gauss-Legendre-Radau and Gauss-Legendre-Lobatto points. The GBDF class of BVMs can

also be interpreted as polynomial collocation methods. Two polynomial collocation methods using the same set of points, will be mathematically equivalent. The n-step GBDF on Gauss-Legendre-Radau points, for example, is mathematically equivalent to the n-stage Radau IIA method. The n-stage Gauss method can be retrieved by constructing the n-point GBDF on the Gauss-Legendre points. The resulting equations calculate the stage values of the Gauss IRK method. The value at the endpoint of the time step is not explicitly calculated by the GBDF method, but it can be obtained by polynomial extrapolation which is in this case equivalent to using the corresponding Gaussian quadrature formula.

BVMs can be considered a general and practical method to construct discretizations. They allow derivation, within the same framework, of spline collocation methods and finite difference methods on uniform and non-uniform meshes, as well as spectral collocation and implicit Runge-Kutta methods. Note that the use of BVMs is of course not restricted to discretization in time (see [BT98] for PDE examples).

5.6 Stability of Chebyshev Spectral Collocation

One can easily derive that the open stability domain S (see §4.5.1) for a CSC method is given by

$$S = \{z \in \mathbb{C} : |R(z)| < 1\} \tag{5.6}$$

with

$$R(z) = {e_n}^T (D - zI_n) d_0. \tag{5.7}$$

Similarly, one finds that the set Σ (see §4.5.1) is given by

$$\Sigma = \{z \in \sigma(D + w^{-1} {e_n}^T d_0),\ |w| \geq 1\}. \tag{5.8}$$

The vector $e_n \in \mathbb{R}^n$ is the last column of the identity matrix I_n. These expressions also follow from the formulation of a CSC method as an IRK method or as a BBVM. Setting $A = D$, $a_0 = d_0$, $B = \Delta t I$ and $b_0 = 0$ in (4.41), for example, leads to (5.8). The same result can be derived from (4.40). As for the time discretization methods discussed in Chapter 4, the sets S and Σ are each other's complement in the complex plane. Points on the boundary of Σ (and therefore S) can be found by setting $w = e^{i\theta}$ in (5.8). Figure 5.3 shows the boundaries of the stability domains for the Chebyshev spectral collocation methods up to order 10 by plotting the points corresponding to $\theta = j\pi/m$, $j = 0, \ldots, m-1$, $m = 256$. From this figure one could be tempted to conclude that these methods are A-stable for any order. A closer look at the imaginary axis, however, shows

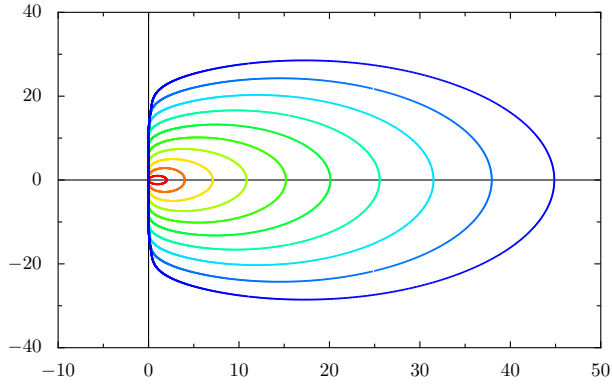


Figure 5.3: Stability domain boundaries for the Chebyshev spectral collocation methods of orders 1 to 10.

that this is not the case. Figure 5.4 illustrates that the methods of orders 3 to 10 are not A-stable. The methods of orders 1 and 2 are A-stable. The method of order one is the implicit Euler method. The method of order two is equivalent with the BGBDF-method with two equal steps. As described in [Wri70, HW96, But03], a method is A-stable if and only if its stability function $R(z)$ is analytic in the left half plane and $|R(iy)| \leq 1$ for $y \in \mathbb{R}$. From (5.7) it follows that the first condition always holds for Chebyshev spectral collocation methods since the eigenvalues of D are always contained in the right half plane [GO77, Boy01]. The rational function $R(z)$ can be written as

$$R(z) = \frac{P(z)}{Q(z)}$$

where $P(z)$ and $Q(z)$ are polynomials. The second condition for A-stability is equivalent to $E(y) \geq 0$ where

$$E(y) = Q(iy)Q(-iy) - P(iy)P(-iy)$$

is a polynomial in y^2 [Wri70, HW96, But03]. The polynomials $E(y)$ for the methods of orders 1, 2, 3 and 4 are y^2, y^4, $y^4(9y^2 - 32)$ and $y^6(y^2 - 16)$ which again shows that the methods of order 1 and 2 are A-stable. For collocation methods based on the zeros of standard orthogonal polynomials (Legendre, Chebyshev, Laguerre, Hermite), $E(y)$ can always be written as a polynomial with integer coefficients. Similarly, the boundaries of the stability domains of such methods correspond to the zero level curve of bivariate polynomials with integer coefficients. The boundary of the stability domain of the Chebyshev spectral collocation method of order 3, for example, is

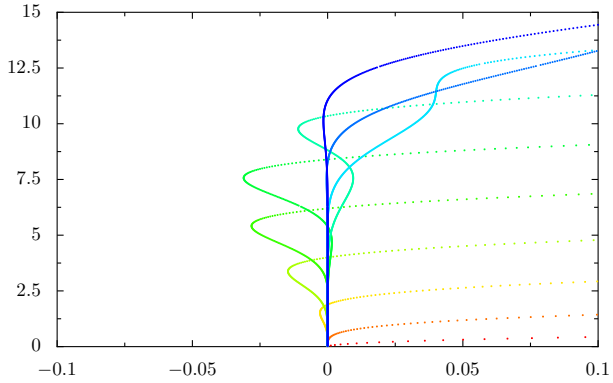


Figure 5.4: Detail of Figure 5.3 near the imaginary axis.

given by all $(x, y) \in \mathbb{R}^2$ such that

$$9\,y^6 + \left(27\,x^2 - 114\,x - 32\right)\,y^4 + \left(27\,x^4 - 228\,x^3 + 704\,x^2 - 896\,x\right)\,y^2 + 9\,x^6 - 114\,x^5 + 736\,x^4 - 3200\,x^3 + 6144\,x^2 - 18432\,x = 0.$$

It is clear from Figure 5.3 that, even though they are not A-stable, the Chebyshev spectral collocation methods have very good stability properties. A time discretization method is said to be $A(\alpha)$-stable [HW96, But03] if the stability domain S contains the sector

$$S_\alpha = \{z \in \mathbb{C} : |\arg(-z)| < \alpha, z \neq 0\}.$$

This weaker form of stability corresponds to A-stability for $\alpha = \frac{\pi}{2}$. Figure 5.5 shows the stability domain and the corresponding sector S_α for the BDF method of order 3. Table 5.1 shows some values of α for different methods and orders. The values are calculated by minimizing $|\arg(-z)|$ where the z-values are obtained by setting $w = e^{i\theta}$ in (5.8), with $\theta = j\pi/(m+1)$, $j = 1, \ldots, m$ and $m = 1024$. The standard Gauss, Radau and Lobatto IRK methods are not represented in the table since they are A-stable ($\alpha = 90°$) for any order.

5.7 Chebyshev Spectral Collocation for PDEs

5.7.1 Model Problem

As in the previous chapters we consider the linear parabolic equation

$$u_t = (au_x)_x + (bu_y)_y + cu + f,$$

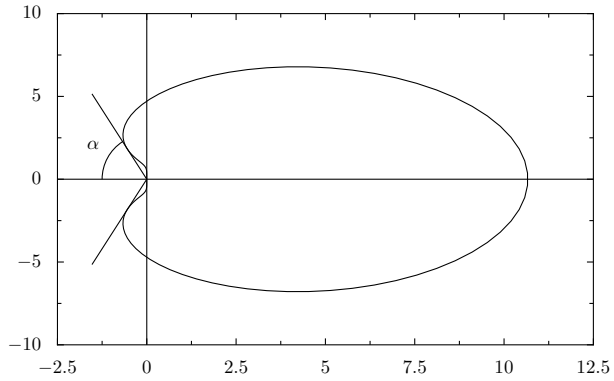


Figure 5.5: Boundary of the stability domain and corresponding sector S_α for the BDF method of order 3 ($\alpha = 73°$).

α	BGBDF	BGAM	CSC
1	90.0000	90.0000	90.0000
2	90.0000	90.0000	90.0000
3	89.3188	90.0000	89.8903
4	87.7322	90.0000	89.7473
5	85.6488	90.0000	89.6995
6	83.0153	90.0000	89.7638
7	79.6949	90.0000	89.9361
8	75.9502	90.0000	89.9995
9	72.5365	86.7188	89.9971
10	69.5617	82.3122	89.9912

Table 5.1: Values of α in degrees for several $A(\alpha)$-stable methods.

on the unit square, i.e., $(x, y) \in [0, 1]^2$. Using finite difference in space (method of lines), this PDE can be approximated by the stiff system of ODEs

$$\dot{u} = Lu + f,$$

where L is a large, sparse, structured matrix (see Chapters 2 and 3 for a detailed explanation). Time integration using Chebyshev spectral collocation results in the system of equations

$$(D \otimes I_m)u + (d_0 \otimes u_0) = (I_n \otimes L)u + f, \tag{5.9}$$

where u, $f \in \mathbb{R}^{nm}$ and $u_0 \in \mathbb{R}^m$. This system of nm equations can be solved using the same multigrid techniques as in Chapter 4. For a Jacobi or Gauss-Seidel smoother m systems with a matrix of the form $D - l_{kk} I_n$ have to be solved. Formulation (5.9) corresponds to the use of the CSC method as a spectral method. In this case, where one spectral discretization is used for the whole time interval, n would typically be quite large. The CSC method can also be used on subintervals. This leads to the formulation

$$(D \otimes I_m)u_i + (d_0 e_n \otimes I_m)u_{i-1} = (I_n \otimes L)u_i + f_i, \quad \text{for } i = 1, \cdots,$$

where $e_n = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T$. The first approach is more similar to a BVM discretization, the second approach is more similar to an IRK or BBVM discretization.

5.7.2 Theoretical Convergence Analysis

We first consider the isotropic diffusion equation

$$u_t = u_{xx} + u_{yy} + f. \tag{5.10}$$

This corresponds to setting $a = b = 1$, $c = 0$ in model problem (5.7.1). The convergence of multigrid time stepping or waveform relaxation methods with Chebyshev spectral collocation in time can be analyzed using the same Fourier-Laplace analysis as in the previous chapters. The following results can easily be obtained using the formulation of CSC as a BVM or IRK method. As before the asymptotic convergence factor is calculated by

$$\rho = \max_{z \in \Sigma} \rho(M(z)). \tag{5.11}$$

The quantities $\rho(M(z))$ can be computed using standard multigrid analysis techniques. For continuous waveform relaxation on infinite intervals $\Sigma = \bar{\mathbb{C}}^+$ or equivalently $z \in i\mathbb{R}$. The resulting asymptotic convergence factor is an upper bound for iterative methods based on A-stable time integration. For

CSC or block CSC on finite sequences setting $A = D$ and $\Delta t B = I$ in Theorem 4.4.2 or $\Delta t A = D^{-1}$ in Theorem 4.4.3 we find

$$\Sigma = \sigma(D) \subset \mathbb{C}^+.$$

For block CSC on infinite sequences setting $A = D$, $a_0 = d_0$, $B = \Delta t I$ and $b_0 = 0$ in (4.41) leads to (5.8)

$$\Sigma = \bigcup_{|w| \geq 1} \sigma(w^{-1} d_0 {e_n}^T + D).$$

The same result can be derived from (4.40). Because CSC methods are $A(\alpha)$-stable with $\alpha \approx \frac{\pi}{2}$, the asymptotic convergence factor for continuous waveform relaxation on infinite intervals gives a good approximative bound for methods using CSC in time.

As usual we report convergences rates

$$R = -\log_{10} \rho, \tag{5.12}$$

instead of convergence factors ρ. The spectral picture in Figure 5.6 shows contours of $R(z) = -log_{10}\rho(M(z))$ together with the boundaries of the stability domains for the 5-step GBDF, BGBDF and the CSC method on 1 and 5 intervals. The CSC1 method is the standard implicit Euler method. The other methods are all of order 5. Note that the BGBDF5 method is not A-stable since its curve is not entirely contained in the right half plane. This becomes more pronounced for higher order BVMs on equidistant points. Since $-\log_{10}$ is a monotonically decreasing function, it follows from (5.12) and (5.11) that the theoretical asymptotic convergence rate is given by

$$R = \min_{z \in \Sigma} R(z),$$

where Σ is the boundary of the stability region of the time discretization method. The asymptotic convergence rate corresponds to the average number of digits gained per iteration (asymptotically). Table 5.2 shows numerical values for the curves in Figure 5.6. The fact that the curve for the BGBDF5 method protrudes slightly into the left half plane is reflected by a lower convergence rate. It is possible to construct problems for which this methods becomes very inefficient. This is the case for example for problems characterized by eigenvalues close to the imaginary axis. In practice, however, the method will usually work well. For higher order GBDF methods on equidistant points, the lack of A-stability becomes more pronounced and the convergence rates are even lower.

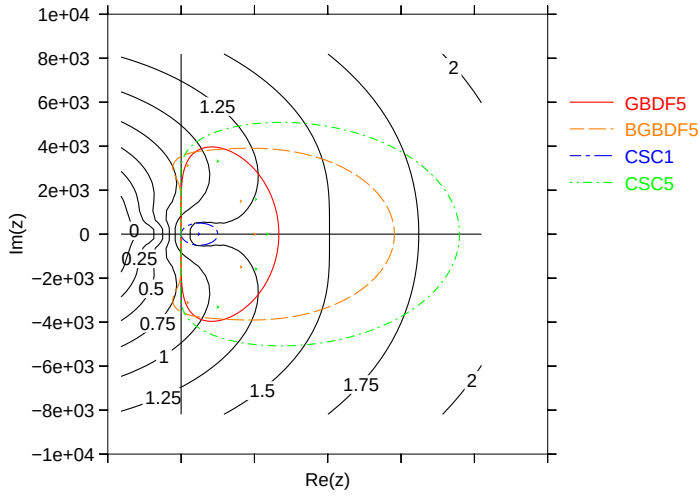


Figure 5.6: Spectral picture.

GBDF5	BGBDF5	CSC1	CSC5
0.80	0.74	1.13	0.79

Table 5.2: Theoretical convergences rates.

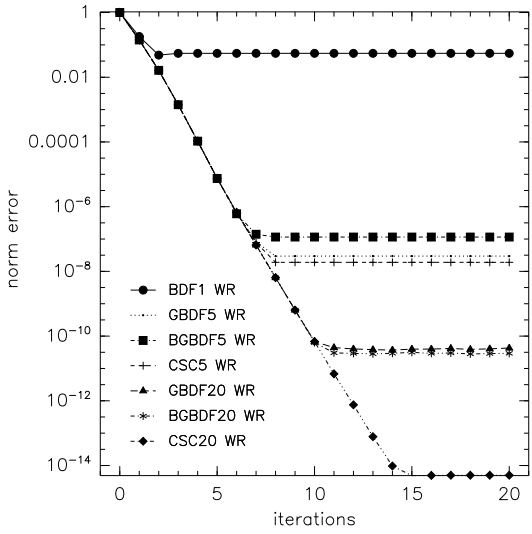


Figure 5.7: Error convergence.

5.7.3 Numerical Experiments

First, we investigate the discretization error of CSC and compare it to the discretization error of other methods. The isotropic diffusion equation (5.10) is discretized in space on a regular square grid with mesh sizes $\Delta x = \Delta y = \frac{1}{8}$. The Dirichlet boundary conditions and the right hand side are chose such that the exact solution is

$$u(x, y, t) = \sin \frac{2\pi t}{10}.$$

Because the solution does not depend on the spatial variables, the discretization error is only due to the time discretization. The time interval ($[0, 0.36]$) and the number of unknowns in the time dimension (360) are the same for all methods. This choice results in a time step $\Delta t = 10^{-3}$ for the implicit Euler method (BDF1). The systems of equations are solved using a standard multigrid V-cycle on 3 grids. Figure 5.7 shows the norm of the difference between the approximate and the exact solution. The discretization error is the smallest difference reached by a method and corresponds to the horizontal parts of the curves in Figure 5.7. As expected higher order methods are more precise. The methods of order 5 all give essentially the same discretization error. The fact that the GBDF and BGBDF methods of order 20 do not attain the same discretization error as the CSC method of order 20 is due to the poor conditioning of the systems resulting from the high or-

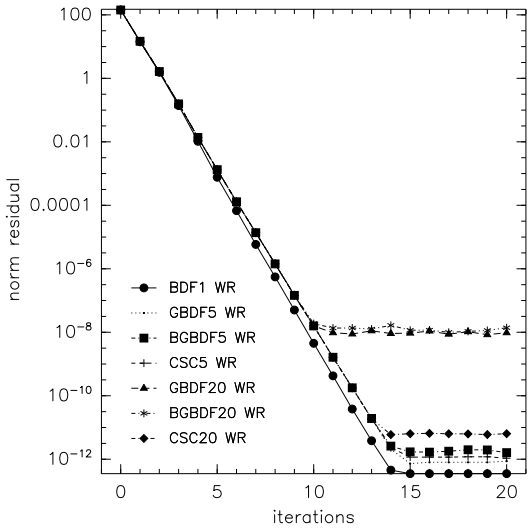


Figure 5.8: Residual convergence.

der time discretization on uniform grids. Figure 5.8 shows that the residual norm does not converge to machine precision for the GBDF and BGBDF methods. It should be noted that calculating the coefficients of high order BVMs using the standard Vandermonde solvers results in a serious loss of accuracy. This is not the cause of the problem as Figure 5.8 shows. Furthermore, essentially the same results were obtained when the coefficients were calculated accurately using the explicit expressions from [AT02]. The poor conditioning of the matrices for uniform meshes is essentially due to the size and the sign of the matrix entries.

Table 5.3 gives estimated convergence rates for a problem with exact solution $u(x, y, t) = 0$ discretized on a grid with mesh sizes $\Delta x = \Delta y = \frac{1}{32}$. The convergence factor for iteration ν is given by

$$\rho^{(\nu)} = \frac{\|u^{(\nu)}\|}{\|u^{(\nu-1)}\|}$$

and the corresponding convergence rate is

$$R^{(\nu)} = -\log_{10} \rho^{(\nu)}.$$

For each method 20 iterations are performed. The estimate in Table 5.3 is the average of the last 10 convergence rates (this corresponds to (2.6) with $\nu = 20$ and $\mu = 10$). It is clear that for low orders the methods

BDF1	GBDF5	BGBDF5	CSC5	GBDF20	BGBDF20	CSC20
0.97	0.87	0.87	0.87	0.11	0.10	0.88

Table 5.3: Experimental convergence rates.

presented here perform essentially the same. For high orders, however, the convergence of the methods using uniform time meshes is quite poor. The almost A-stable CSC that uses a clustered time mesh, on the other hand, maintains a good convergence for high orders.

5.8 Conclusions

From Chapter 4 we know that the convergence of iterative methods for systems of ODEs depends on the stability of the time discretization. If continuous waveform relaxation converges on $[0, \infty)$, then a discrete method using an A-stable ODE integrator will converge as well. Chebyshev spectral collocation methods are very close to A-stable for any order, resulting in good convergence. Such an implicit high order method is very expensive when applied directly to large system of ODEs, but in an iterative method such as Jacobi and Gauss-Seidel waveform relaxation only scalar ODEs have to be solved. For waveform relaxation it is therefore worth considering time discretization schemes that would be too expensive in other cases. Chebyshev spectral collocation is well suited as a time discretization for waveform relaxation if a high order of approximation is needed and possible, that is if the problem is smooth. In such cases a relatively small number of unknowns can provide a very high accuracy. Chebyshev spectral collocation involves systems with a dense matrix that has a simple explicit formula. In some cases it may be worthwhile to solve these systems using (vectorized) fast transform methods instead of dense matrix methods.

Chapter 6

Problems with Delay

The convergence of (multigrid) waveform relaxation methods is studied in a Fourier-Laplace framework for a specific class of delay differential equations. This leads to quantitative convergence estimates, which are illustrated and validated by means of numerical experiments with two delay partial differential model problems.

6.1 Introduction

In the previous chapters we considered iterative methods for systems of ordinary differential equations (ODE) of the form

$$\begin{aligned} \dot{v}(t) &= f(t, v(t)), \qquad t \in [0, t_F] \\ v(0) &= v_0. \end{aligned}$$

More specifically, we used multigrid methods for stiff systems derived by spatial discretization of parabolic partial differential equations (PDE). For ODEs the derivative of the solution depends only on the value of the evolving quantities at the current time. For delay differential equations (DDE) the derivative of the solution can depend on the whole history of the evolving quantity. The specific subclass we discuss here consists of DDEs with one constant delay

$$\begin{aligned} \dot{v}(t) &= f(t, v(t), v(t-\tau)), \qquad t \in [0, t_F], \\ v(t) &= v_0(t), \qquad\qquad\qquad\quad t \in [-\tau, 0]. \end{aligned}$$

Note that the initial condition specifies the function for a whole interval and no longer just at a single point. Such equations can be obtained from

discretizing certain delay partial differential equations (DPDE). A simple example is the heat equation with one constant delay

$$u_t = u_{xx} + u(t - \tau).$$

Other examples of DPDEs arise for example in population dynamics and in the study of non-linear materials with memory. An example from population dynamics is the Hutchinson equation with diffusion

$$u_t = au_{xx} + bu(1 - cu(t - \tau)),$$

which models the density of a migrating and reproducing population. A similar equation with distributed delay is given by

$$u_t = au_{xx} + b(1 - \int_{-\infty}^{t} Q(t - x)u(s)ds).$$

Such models appear in simulations where the age structure of a population is taken into account. Other integrodifferential equations such as

$$u_t = au_{xx} + (1 + bu - (1 + b)g * u),$$

where $*$ denotes convolution, are considered in [Bri90]. Many more examples of DPDEs can be found in the monographs by Wu [Wu96] and by Kolmanovskii and Myshkis [KM99].

Waveform relaxation was originally developed for solving large systems of ODEs. The method has, however, also been applied to DDEs [Bjø94, JKL97, ZKV99]. In particular, the authors of [ZKV99] introduced the so-called Picard waveform relaxation methods and derived error estimates which hold under certain Lipschitz conditions for the right-hand sides of the equations. Their results are illustrated by means of numerical experiments for semi-discrete DPDEs of parabolic type. Their theory provides conditions for convergence, but the results are qualitative and give no indications about the speed of convergence. In this chapter, quantitative convergence estimates of waveform relaxation methods for DDEs are derived by analyzing the methods in the classical Fourier-Laplace framework of Miekkala and Nevanlinna [MN87a]. The analysis is illustrated for the numerical method of lines approximation of a two-dimensional diffusion equation extended with a term that has a constant delay in time. A similar Fourier-Laplace framework is used to investigate the acceleration of the basic waveform methods by multigrid techniques.

This chapter is organized as follows. In §6.2 spatial discretization of a two-dimensional diffusion equation with a constant delay in time is considered. The Picard waveform relaxation method, introduced in [ZKV99] is analyzed in §6.3. Section 6.4 introduces and analyzes non-Picard waveform relaxation methods. Multigrid waveform relaxation for the model problem is described in §6.5. Results of numerical experiments are given in §6.6.

6.2 Model Problem

We consider the two-dimensional parabolic DPDE

$$\frac{\partial u(t,x,y)}{\partial t} = a\left(\frac{\partial^2 u(t,x,y)}{\partial x^2} + \frac{\partial^2 u(t,x,y)}{\partial y^2}\right) + bu(t-\tau,x,y), \qquad (6.1)$$

for $(x,y) \in \Omega = [0,d]^2$, $t \in [0,t_F]$. The constants a and τ are assumed to be positive. The initial and boundary conditions are chosen such that the exact solution becomes $u(t,x,y) = 0$. That is, $u(t,x,y) = 0$ for $(t,x,y) \in ([-\tau,0] \times \Omega) \cup ([0,t_F] \times \partial\Omega)$.

Applying to (6.1) the method of lines using the same finite difference discretization as in §2.3.1 and §3.2 leads to the following system of DDEs,

$$\begin{aligned}\dot{u}_{i,j}(t) = ah^{-2}\big(&-4u_{i,j}(t) + u_{i-1,j}(t) + u_{i+1,j}(t) + u_{i,j-1}(t) + u_{i,j+1}(t)\big)\\ &+ bu_{i,j}(t-\tau),\end{aligned}$$

for $i,j = 1,\ldots,n-1$ and with $\Delta x = \Delta y = h = d/n$ and $u_{i,j}(t) \approx u(ih,jh,t)$. In matrix notation this gives

$$\dot{u}(t) = Lu(t) + bu(t-\tau), \qquad (6.2)$$

for $t \in [0,t_F]$, with $u(t) = 0$ for $t \in [-\tau,0]$. Here, $L \in \mathbb{R}^{(n-1)^2 \times (n-1)^2}$ is the matrix with the familiar block-tridiagonal structure with tridiagonal blocks (see §2.3.1). The vector $u(t)$ contains approximations $u_{(n-1)(i-1)+j}(t) \approx u(t,ih,jh)$ to the solution of the DPDE.

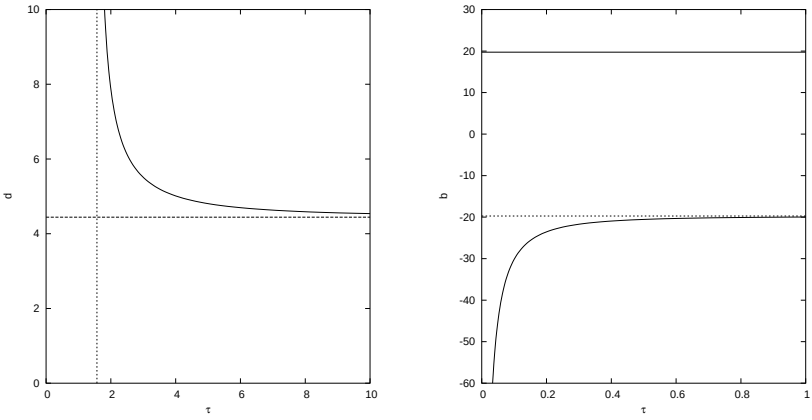
For the examples the parameters are chosen such that (6.1) has a stable zero solution. The following result is used [HV03, Corollary 3.3].

Theorem 6.2.1. The zero solution of (6.1) is asymptotically stable iff

- $a \geq 0$ and
- $-2\pi^2 a/d^2 < -b < \frac{\theta}{\tau \sin\theta}$

where θ is the root of $\theta\cos\theta = -2\tau a\pi^2/d^2$ such that $\pi/2 \leq \theta < \pi$.

Note that the stability of (6.1) is determined by the PDE coefficients a and b, the magnitude of the delay τ and the size of the spatial domain d. Figure 6.1(a) shows the stability region in the (τ,d)-plane for $a = 1$ and $b = -1$. Points below the curve result in stable solutions. Figure 6.1(b) shows the stability region in the (τ,b)-plane for $a = 1$ and $d = 1$. Points between the curves result in stable solutions. In [HV03] it is shown that the stability region of the semi-discrete problem (6.2) is a subset of the stability region of (6.1). Because the difference is negligible for small mesh sizes h, the details are omitted here. To simplify the notation the parameters a and b are set to 1 and -1.



(a) (τ, d)-plane ($a = 1$ and $b = -1$) (b) (τ, b)-plane ($a = 1$ and $d = 1$)

Figure 6.1: Stability regions for the model problem.

6.3 Picard Waveform Relaxation

The *Picard waveform relaxation* methods for DDEs that were introduced in [ZKV99] can be applied to (6.2). This gives, for $t \in [0, t_F]$,

$$\dot{u}^{(\nu)}(t) - L^+ u^{(\nu)}(t) = L^- u^{(\nu-1)}(t) - u^{(\nu-1)}(t-\tau) \tag{6.3}$$

where $u^{(\nu)}(t) = 0$ for $t \in [-\tau, 0]$ and $L = L^+ + L^-$. The term 'Picard' is used to stress that the delay term is taken from a previous iterate in all these schemes. The *direct/Picard* method is defined by $L^+ = L$, $L^- = 0$, while, the *Jacobi/Picard* and *Gauss-Seidel/Picard* waveform relaxation schemes correspond to taking for L^+ the diagonal or lower triangular part of L. See Chapter 2 for an explanation of the standard Jacobi and Gauss-Seidel waveform relaxation methods.

In [ZKV99], the convergence of the Picard waveform relaxation methods was analyzed for general DDEs by deriving error estimates which depend on Lipschitz conditions of the right-hand side. While such an analysis gives rise to qualitative results for a general class of problems, the current model problem also allows a quantitative analysis of the waveform methods in the Fourier-Laplace framework developed in [MN87a]. This framework is described in §2.5.5 and was used in the previous chapters.

To start the analysis of the Picard waveform relaxation methods, (6.2) is rewritten as

$$\dot{u}(t) - L^+ u(t) = L^- u(t) - u(t-\tau), \quad t \in [0, t_F].$$

Subtracting this equation from (6.3) gives the error equivalent of the latter,

$$\dot{e}^{(\nu)}(t) - L^+ e^{(\nu)}(t) = L^- e^{(\nu-1)}(t) - e^{(\nu-1)}(t-\tau), \quad t \in [0, t_F], \tag{6.4}$$

where $e^{(\nu)}(t) = u^{(\nu)}(t) - u(t)$. If the Laplace transform of $e^{(\nu)}(t)$ is denoted by $\widetilde{e}^{(\nu)}(z)$, Laplace transforming (6.4) results in

$$\widetilde{e}^{(\nu)}(z) = K(z)\widetilde{e}^{(\nu-1)}(z), \tag{6.5}$$

with the *Picard waveform relaxation symbol*

$$K(z) = (zI - L^+)^{-1}(-e^{-\tau z}I + L^-). \tag{6.6}$$

By an inverse Laplace-transform argument, the relation,

$$\begin{aligned} e^{(\nu)}(t) &= \mathcal{K}e^{(\nu-1)}(t) \\ &= \int_0^t k(t-s)e^{(\nu-1)}(s)ds, \end{aligned} \tag{6.7}$$

is derived, i.e., the *Picard waveform relaxation operator* $\mathcal{K}$ is a linear Volterra convolution operator with kernel $k(t)$. More precisely, since $\lim_{z\to\infty} K(z) = 0$, $k(t) \in L_1(0,\infty)$ if $K(z)$ is bounded and analytic in an open domain containing the closed right-half complex plane, i.e., if all eigenvalues of L^+ have negative real parts, see, e.g., [JSW82, Prop. 2.3].

The spectral radius $\rho(\mathcal{K})$ of this convolution operator $\mathcal{K}$ determines the asymptotic convergence of the corresponding Picard waveform relaxation method. It can be investigated in the spaces $L_p(0,\infty)$ of Lebesgue-measurable functions which are p-th power integrable (see Chapter 2).

Theorem 6.3.1. Consider equation (6.1), discretized in space using the numerical method of lines, and assume all eigenvalues of L^+ have negative real parts. Then, the spectral radius of the Picard waveform relaxation operator $\mathcal{K}$ considered as an operator in $L_p(0,\infty)$ with $1 \le p \le \infty$ is given by

$$\begin{aligned} \rho(\mathcal{K}) &= \sup_{z\in\overline{\mathbb{C}}^+} \rho(K(z)), \\ &= \sup_{\xi\in\mathbb{R}} \rho(K(i\xi)), \end{aligned}$$

with $K(z)$ as in (6.6).

6.4 Non-Picard Waveform Relaxation

Picard waveform relaxation methods solve a system of DDEs by solving a sequence of simple systems of ODEs. Standard (that is, non-Picard)

waveform relaxation methods use simple systems which are still DDEs. For (6.2) the iteration becomes

$$\dot{u}^{(\nu)}(t) - L^+ u^{(\nu)}(t) + u^{(\nu)}(t-\tau) = L^- u^{(\nu-1)}(t), \quad t \in [0, t_F]. \tag{6.8}$$

The *Jacobi* and *Gauss-Seidel* variants of which are defined in terms of their respective splittings given above. The non-Picard waveform relaxation methods (6.8) can be analyzed in a similar way as the Picard methods. Rewriting (6.2) as

$$\dot{u}(t) - L^+ u(t) + u(t-\tau) = L^- u(t), \quad t \in [0, t_F],$$

subtracting this from (6.8) and applying Laplace transforms, results in (6.5) with the *waveform relaxation symbol*

$$K(z) = ((z + e^{-\tau z})I - L^+)^{-1} L^-. \tag{6.9}$$

An inverse Laplace-transform argument then immediately implies that the *waveform relaxation operator* $\mathcal{K}$ is a linear Volterra convolution operator with kernel $k(t)$ (see (6.7)).

In order to be able to determine the spectral radius $\rho(\mathcal{K})$ in the spaces $L_p(0,\infty)$, one has to prove that $k(t) \in L_1(0,\infty)$. From [JSW82, Prop. 2.3], it follows that it suffices that $\lim_{z\to\infty} K(z) = 0$ and that $K(z)$ is bounded and analytic in an open domain containing the closed right-half complex plane. The first condition is easily checked. If we define

$$\Sigma = \{z + e^{-\tau z} : z \in \bar{\mathbb{C}}^+\},$$

then the second condition can be written as

$$\Sigma \cap \sigma(L^+) = \phi.$$

This condition always holds if $\mathrm{Re}(\lambda) < -1$ for all $\lambda \in \sigma(L^+)$. It also holds if the eigenvalues of L^+ are real and negative, as in our model problem, and $\tau < \frac{\pi}{2}$. If there are eigenvalues such that $-1 < \mathrm{Re}(\lambda) < 0$, then a $\tau > \frac{\pi}{2}$ can always be found such that $\lambda \in \Sigma$. In such cases convergence cannot be guaranteed. Figure 6.2 illustrates the shape of Σ for different values of τ.

Theorem 6.4.1. Consider equation (6.1), discretized in space using the numerical method of lines, and assume $\sigma(L^+) \cap \Sigma = \phi$. Then, the spectral radius of the waveform relaxation operator $\mathcal{K}$ considered as an operator in $L_p(0,\infty)$ with $1 \le p \le \infty$, is given by

$$\begin{aligned} \rho(\mathcal{K}) &= \sup_{z\in\bar{\mathbb{C}}^+} \rho(K(z)), \\ &= \sup_{\xi\in\mathbb{R}} \rho(K(i\xi)), \end{aligned}$$

with $K(z)$ as in (6.9).

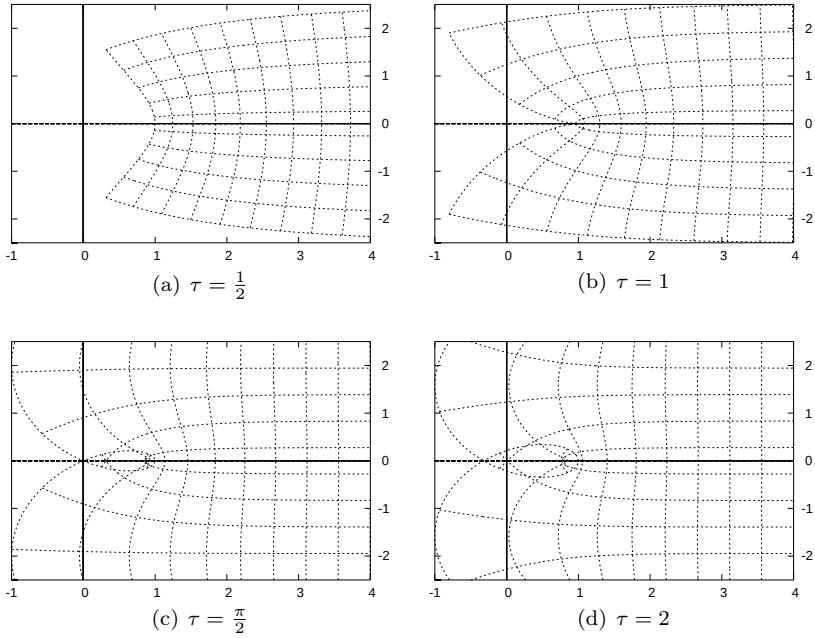


Figure 6.2: Image of the rectangle $[0, 4] \times [-2.5, 2.5] \subset \mathbb{C}$ transformed by $z \to z + e^{-\tau z}$ for different values of τ.

6.5 Multigrid Waveform Relaxation

The convergence rate of the Jacobi and Gauss-Seidel waveform relaxation methods, described in the previous sections, depends on the spatial mesh size. If they converge, the convergence will be very slow except on very coarse grids. This is to be expected since the same happens for the corresponding problems without delay. However, just as in the case without delay, efficient multigrid methods can be developed based on the simple waveform relaxation schemes. Below, a two-grid cycle for model problem (6.2) is stated. A Picard waveform relaxation method is used as a smoother. Two nested grids are used. The quantities on the coarse grid are denoted by barred symbols.

Presmoothing. Set $v^{(0)} = u^{(\nu-1)}$, and perform ν_1 Picard waveform relaxation steps: for $\nu = 1, 2, \ldots, \nu_1$, solve

$$\dot{v}^{(\nu)}(t) - L^+ v^{(\nu)}(t) = L^- v^{(\nu-1)}(t) - v^{(\nu-1)}(t-\tau), \tag{6.10}$$

with $v^{(\nu)}(t) = 0$, $t \in [-\tau, 0]$.

Coarse-grid correction. Compute the defect

$$d(t) = \dot{v}^{(\nu_1)}(t) - L v^{(\nu_1)}(t) + v^{(\nu_1)}(t-\tau).$$

Solve the coarse-grid equivalent of the defect equation,

$$\dot{\bar{u}}(t) - \bar{L}\bar{u}(t) - \bar{u}(t-\tau) = Rd(t), \tag{6.11}$$

with $\bar{u}(t) = 0$, $t \in [-\tau, 0]$ and R the *restriction operator* transferring fine-grid quantities to coarse-grid ones. Then, interpolate the correction $\bar{u}(t)$ to the fine grid, and correct the current approximation,

$$w^{(0)}(t) = v^{(\nu_1)}(t) - P\bar{u}(t),$$

with P the *prolongation operator* which projects coarse-grid quantities onto fine-grid ones.

Postsmoothing. Perform ν_2 iterations of type (6.10), starting with $w^{(0)}(t)$, and set $u^{(\nu)}(t) = w^{(\nu_2)}(t)$.

Since (6.11) is formally equal to (6.2) (except for the zero right-hand side in the latter problem), this two-grid cycle can be applied in a recursive way to obtain a multigrid cycle on more than two nested grids.

The theoretical analysis of the two-grid waveform relaxation method can be performed in analogy with the analysis of the (Picard) waveform

relaxation methods. Rewriting all the steps of the two-grid scheme in error notation and applying Laplace-transform techniques results in

$$\tilde{e}^{(\nu)}(z) = M(z)\tilde{e}^{(\nu-1)}(z),$$

with the *two-grid symbol*

$$\begin{aligned} M(z) &= S^{\nu_2}(z)C(z)S^{\nu_1}(z), \\ C(z) &= I - P\bar{L}(z)^{-1}RL(z), \\ \bar{L}(z) &= (z + e^{-\tau z})I - \bar{L}, \\ L(z) &= (z + e^{-\tau z})I - L, \\ S(z) &= (zI - L^+)^{-1}(-e^{-\tau z}I + L^-). \end{aligned} \tag{6.12}$$

An inverse Laplace-transform argument shows that the *two-grid waveform relaxation operator* $\mathcal{M}$ is of linear Volterra convolution type, i.e.,

$$e^{(\nu)} = \mathcal{M}e^{(\nu-1)} = \int_0^t m(t-s)e^{(\nu-1)}(s)ds.$$

To determine the spectral radius $\rho(\mathcal{M})$ in the spaces $L_p(0,\infty)$, it has to be shown that $m(t) \in L_1(0,\infty)$. From [JSW82, Prop. 2.3], it follows that to this end, it suffices that $\lim_{z\to\infty} M(z) = 0$ and that $M(z)$ is bounded and analytic in an open domain containing the closed right-half of the complex plane. The first condition follows immediately from the fact that $\lim_{z\to\infty} S(z) = 0$. On the other hand, $M(z)$ is bounded and analytic in an open domain containing the closed right-half of the complex plane if $S(z)$ and $C(z)$ are both bounded and analytic in such a region. This means that all eigenvalues of L^+ must have negative real parts and $\sigma(\bar{L}) \cap \Sigma = \phi$. The following theorem therefore holds.

Theorem 6.5.1. Consider equation (6.1), discretized in space using the numerical method of lines, and assume $\sigma(L^+) \cap \bar{\mathbb{C}}^+ = \phi$ and $\sigma(\bar{L}) \cap \Sigma = \phi$. The spectral radius of the two-grid waveform relaxation operator $\mathcal{M}$, considered as an operator in $L_p(0,\infty)$ with $1 \le p \le \infty$, is given by

$$\rho(\mathcal{M}) = \sup_{z\in\bar{\mathbb{C}}^+} \rho(M(z)), \tag{6.13}$$

$$= \sup_{\xi\in\mathbb{R}} \rho(M(i\xi)), \tag{6.14}$$

with $M(z)$ as in (6.12).

Figure 6.3 shows the spectral radius of the two-grid symbol $\rho(M(z))$ over the imaginary axis for (6.2), with $d = 10$, $a = 1$, $b = -1$, $\tau = 1$,

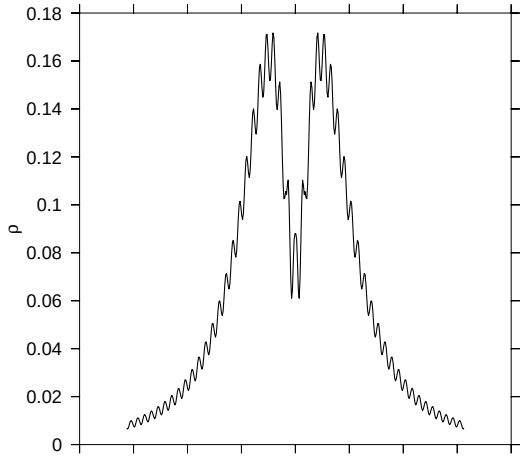


Figure 6.3: $\rho(M(i\xi))$ for $d = 10$, $a = 1$, $b = -1$, $\tau = 1$.

$M = 32$ and using Gauss-Seidel/Picard waveform relaxation as smoother. The scale of the horizontal axis is not specified since it can easily be shown that, except for the scale, the result is exactly the same for all parameters for which $a\tau d^{-2} = 10^{-2}$ and $b\tau = -1$. The delay manifests itself as a wiggle on top of the curve for the equation without delay ($b = 0$). Its amplitude and frequency depend on the choice of parameters.

Instead of a Picard waveform relaxation method, also the standard waveform relaxation method can be used as smoother, i.e., (6.10) can be replaced by

$$\dot{u}^{(\nu)}(t) - L^{+}u^{(\nu)}(t) + u^{(\nu)}(t-\tau) = L^{-}u^{(\nu-1)}(t).$$

Similar results as in Theorem 6.5.1 can be proved for the resulting multigrid waveform operator $\mathcal{M}$.

Table 6.1 shows the spectral radii computed by (6.14) for $a = 1$, $b = -1$, $M = 32$ and several values of d and τ.

6.6 Numerical Results

Some numerical results are presented for the multigrid waveform relaxation method applied to (6.1), discretized using the method of lines. Instead of solving (6.11) exactly on the coarse grid, the two-grid method is implemented in a recursive manner until a mesh with 1 internal grid point is obtained. The method uses a standard V-cycle with 1 pre- and 1 postsmooth-

Table 6.1: Values of $\rho(\mathcal{M})$ for $a = 1$, $b = -1$, $M = 32$

τ	d	Picard	non-Picard
1	1	0.1625	0.1624
1	2	0.1626	0.1620
1	5	0.1651	0.1630
2	1	0.1623	0.1625
2	2	0.1627	0.1625
2	5	0.1637	0.1650

Table 6.2: Averaged convergence factors for $a = 1$, $b = -1$, $M = 32$, $t_F = 50$

τ	d	Picard	non-Picard
1	1	0.1374	0.1161
1	2	0.1951	0.1477
1	5	0.2037	0.1797
2	1	0.1372	0.1168
2	2	0.2021	0.1124
2	5	0.1157	0.1089

ing step of Gauss-Seidel/Picard or Gauss-Seidel type, linear interpolation and full-weighting restriction. Obviously, the continuous-time method has to be discretized in time in an actual implementation. The effect of this time discretization on the convergence properties is not considered here. The backward differentiation formula of order two with time-step 0.1 is used for all the experiments. Initially all unknowns are set to one.

Table 6.2 reports averaged convergence factors for the parameters $a = 1$, $b = -1$, $M = 32$, on a time interval with length $t_F = 50$. The averages are obtained by taking the geometric average of quotient of the L_2-norm of consecutive errors for the last 20 iterations in a total of 40. These factors correspond roughly to the infinite-interval results of Table 6.1. For large values of d the convergence becomes slightly more erratic, but the methods are still efficient.

To show that it is possible to solve more general equations using the methods discussed here, a multigrid method was implemented to solve a general diffusion equation with varying coefficients and a term with a con-

stant delay.

$$\begin{aligned}\frac{\partial u}{\partial t} &= \frac{\partial}{\partial x}\left(a\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(b\frac{\partial u}{\partial y}\right)+cu+du(t-\tau)+f,\\ a(x,y,t) &= \exp(10(x-y)\sin(t)),\\ b(x,y,t) &= \exp(-10(x-y)\cos(\pi t)),\\ c(x,y,t) &= 2-\exp(-t),\\ d(x,y,t) &= 1+\exp(t).\end{aligned}$$

The function f is chosen such that the exact solution is $u(x,y,t) = x+y+t$. This is an anisotropic problem since the diffusion coefficients in a certain point can be very different depending on the direction of diffusion. Standard multigrid methods do not handle this type of problem well and yield very slow convergence. The multigrid as smoother (MGS) method, which uses the same simple smoothers as before, together with an extended hierarchy of coarse grids, was used here. For more information on this multigrid scheme we refer to §3.3.4. The multigrid algorithm uses full weighting restriction, bilinear and linear interpolation, 1 pre- and 1 postsmoothing step. The smoothing consists of a multigrid semicoarsening step in the x-direction followed by one in the y-direction both using only 1 presmoothing red-black Gauss-Seidel step. A hierarchy of 5×5 grids is used. A total of 10 iterations was performed. Average convergence factors, taken over the last 5 iterations, are 0.0557 for the Picard as well as the non-Picard smoother. The PDE is solved to within the spatial discretization error in 4 iterations (there is no discretization error in time).

6.7 Conclusions

In this chapter, a theoretical Fourier-Laplace framework was set up to derive quantitative convergence estimates of waveform relaxation methods for semi-discretized DPDEs.

Roughly speaking, a great resemblance with the convergence behavior of the related methods for PDEs without delay can be observed. That is, the multigrid waveform relaxation methods exhibit mesh-size independent convergence behavior.

Although a very simple model problem is used to explain and illustrate the methods and their analysis, the provided information should allow one to derive similar results for more complicated problems. Other issues which one might be interested in are, e.g., the influence of the particular time discretization method used on the algorithm's convergence speed, and the treatment of non-linearities and more general, variable or state-dependent delays.

Chapter 7

Convergence Analysis using Functional Calculus

The convergence analysis of waveform relaxation methods traditionally uses the theory of Volterra convolution equations. More specifically the convergence theory can be based on a theorem of Paley and Wiener that gives a condition for the solution of a linear Volterra convolution equation to be bounded. Extensions of this theorem to discrete convolution equations and vector-valued problems have been described in the literature. In this chapter we show that the same results can be derived by an alternative approach based on functional calculus. A functional calculus defines what is meant by a function of an operator. A spectral mapping theorem then relates the spectrum of the resulting operator to the spectrum of the original operator. The Dunford-Taylor functional calculus for scalar analytic functions is extended in this chapter to matrix-valued analytic functions. Using the corresponding spectral mapping theorem, it is straightforward to analyze the convergence of a large number of waveform relaxation algorithms. The theory is applied to the analysis of continuous waveform relaxation and discrete waveform relaxation based on general linear methods, both for initial value and time periodic systems of ordinary differential equations. A brief explanation is given of how the theory can be applied in the context of convergence analysis of multigrid methods.

7.1 Introduction

Consider the linear system

$$\dot{u} = Lu + f,$$

with $u(t)$, $f(t) \in \mathbb{R}^m$ and $L \in \mathbb{R}^{m \times m}$. As explained in Chapter 2, a matrix splitting $L = L^+ + L^-$ results in the waveform relaxation iteration

$$\dot{u}^{(\nu)} = L^+ u^{(\nu)} + L^- u^{(\nu-1)} + f.$$

For a given system and splitting, several waveform relaxation variants are possible depending on the way the time dimension is treated. From a theoretical point of view, we have shown that it is interesting to study continuous waveform relaxation, where the simpler ODEs are solved exactly, on either a finite or an infinite time interval. In a practical implementation the ODEs have to be discretized and solved numerically. Different time discretization schemes lead to different discrete waveform relaxation methods. An Implicit Euler discretization, for example, results in the following method

$$\frac{u_i^{(\nu)} - u_{i-1}^{(\nu)}}{\Delta t} = L^+ u_i^{(\nu)} + L^- u_i^{(\nu-1)} + f_i,$$

which was analyzed in Chapter 3. The more general case of discretization with a LMM, IRK method, BVM or GLM was studied in Chapter 4. The case of spectral discretization in time was considered in Chapter 5.

The convergence analysis of these methods was based on the theory of Volterra integral equations, as introduced in the seminal paper [MN87a]. There, it is shown that continuous waveform relaxation can be described by a linear Volterra convolution operator $\mathcal{K}$ whose spectral radius $\rho(\mathcal{K})$ determines the convergence of the method. Below we recall the main theorem, which characterizes the spectral radius. It can be proved using the Paley-Wiener theorem that gives a necessary and sufficient condition for the boundedness of the solution of a linear Volterra convolution equation [MN87a, LO87, JV96a].

Theorem 7.1.1 (Th. 2.2, [MN87a]). *Consider $\mathcal{K}$ as a linear operator in $L^p([0,\infty], \mathbb{C}^m)$ with $1 \leq p \leq \infty$, and assume $\sigma(L^+) \cap \bar{\mathbb{C}}^+ = \phi$. Then, $\mathcal{K}$ is a bounded operator and*

$$\rho(\mathcal{K}) = \sup_{z \in \bar{\mathbb{C}}^+} \rho((zI_m - L^+)^{-1} L^-). \tag{7.1}$$

A framework for the analysis of discrete waveform relaxation was provided in [MN87b]. The main theorem, which was also the basis of our analyses in the previous chapters, is restated below. Its proof is based on a discrete equivalent of the Paley-Wiener theorem [MN87b, LO87, JV96b].

Theorem 7.1.2 (Th. 3.1, [MN87b]). *Consider $\mathcal{K}$ as a linear operator in $l^p(\infty, \mathbb{C}^m)$ with $1 \leq p \leq \infty$, and assume $\sigma(L^+) \cap \Sigma = \phi$. Then, $\mathcal{K}$ is a bounded operator and*

$$\rho(\mathcal{K}) = \sup_{z \in \Sigma} \rho((zI_m - L^+)^{-1} L^-), \tag{7.2}$$

where Σ is the complement of the interior of the stability region of the time discretization scheme scaled by $\frac{1}{\Delta t}$.

The formulae for the spectral radii of continuous and discrete waveform relaxation operators are clearly very similar. This chapter suggests an approach, based on functional calculus, that unifies the convergence analyses of the different waveform relaxation methods. The approach can be summarized by the general formula

$$\rho(F(T)) = \sup_{z \in \sigma(T)} \rho(F(z)), \tag{7.3}$$

where T is any closed linear operator (with non-empty resolvent set) and F is a matrix-valued function analytic in a neighborhood of $\sigma(T)$, the spectrum of T (see Corollary 7.2.21). Taking $T = \frac{d}{dt}$ and $F(z) = (zI_m - L^+)^{-1}L^-$, for example, gives the formula for continuous waveform relaxation, as will be shown in §7.3.

The functional calculus for scalar functions is well known. That is, the definition of a scalar function of a matrix (or operator) is classic and well understood. It is shown here that the convergence of waveform relaxation methods can be analyzed by using a straightforward extension to matrix-valued functions. The approach based on operator theory can be interpreted as a generalization of the analysis of iterative methods for matrices to the analysis of the corresponding methods for matrices of operators (or equivalently matrix-valued operators). In §7.2 the operator calculus is first introduced for matrices, then for bounded linear operators and finally for closed linear operators. Section 7.3 shows how continuous and discrete waveform relaxation for initial value problems on finite and infinite time intervals and for time-periodic problems fit into the general framework. For the discrete case, time discretization based on general linear methods is considered. Linear multistep, implicit Runge-Kutta and block boundary value methods can be derived as special cases. Section 7.5 outlines how the theory can be used for the two-grid convergence analysis of multigrid methods. Section 7.6 provides some concluding remarks.

7.2 Spectral Mapping Theorems for Matrix-Valued Functions of Operators

The theory of this section is introduced first for matrices, then for bounded linear operators and finally for closed (possibly unbounded) linear operators. Each class of operators is a generalization of the next. In principle, it would suffice to consider only closed linear operators. However, for ease of understanding and clarity of exposition the step by step approach is

preferred. For each of the three cases, a definition is given for what is meant by $f(T)$, a function of an operator. A set of rules is provided for calculating with such functions. Using this so-called functional calculus, the spectrum of $f(T)$ is characterized.

The theory for scalar functions is well developed. Detailed descriptions of the Dunford-Taylor functional calculus for scalar functions can be found in functional analysis monographs such as [DS57, Tay58, HP74, Rud73, Vas82] and textbooks such as [Sch71, Lax02]. See [Haa05] for a recent state of the art. Here, extensions to matrix-valued functions are given.

7.2.1 Functional Calculus for Functions of Matrices

The functional calculus for functions of matrices is a special case of the general theory since matrices can be considered as representations of linear operators in finite dimensional vector spaces. Considering this case separately is nevertheless useful because the results can be obtained using only linear algebra (see §2.2). We start by recalling some standard results from the theory of scalar functions of matrices.

Scalar Functions of Matrices

There are many ways to define a function of a matrix. Most definitions turn out to be equivalent in practice. We use here a definition based on line integration in the complex plane. A similar formulation will be used as the starting point for defining functions of more general operators. For more details on functions of matrices we refer to [HJ94, Ch. 6], [GVL96, Ch. 11] and [Rin55].

Definition 7.2.1. The family of scalar functions $f : \mathbb{C} \to \mathbb{C}$, analytic in some neighborhood of $\sigma(T)$ is denoted by $\mathcal{F}(T)$.

Definition 7.2.2. Given a matrix $T \in \mathbb{C}^{n\times n}$ and a function $f \in \mathcal{F}(T)$, the matrix $f(T) \in \mathbb{C}^{n\times n}$ is defined by

$$f(T) = \frac{1}{2\pi i} \oint f(z)(zI_X - T)^{-1} dz. \tag{7.4}$$

All line integration is over a contour that is appropriately chosen in relation to the spectrum $\sigma(T)$ (for more details see [DS57, Tay58]). Note that (7.4) can be interpreted as an extension of the classical Cauchy formula for the value of an analytic function f at the point z_0

$$f(z_0) = \frac{1}{2\pi i} \oint \frac{f(z)}{z - z_0} dz.$$

A characterization of $f(T)$ derived from Definition 7.2.2 is given by the following theorem [GVL96, Th. 11.1.1], [HJ94, Def. 6.2.4].

Theorem 7.2.3. *Let $T = V\,\mathrm{diag}(J_1, \ldots, J_q)V^{-1}$ be the Jordan decomposition of $T \in \mathbb{C}^{n\times n}$ with Jordan blocks $J_i \in \mathbb{C}^{n_i \times n_i}$,*

$$J_i = \begin{bmatrix} \lambda_i & 1 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \ldots & 0 & \lambda_i \end{bmatrix}.$$

If $f \in \mathcal{F}(T)$, then

$$f(T) = V\,\mathrm{diag}(f(J_1), \ldots, f(J_q))V^{-1}$$

with

$$f(J_i) = \begin{bmatrix} f(\lambda_i) & f^{(1)}(\lambda_i) & \cdots & \frac{f^{(n_i-1)}(\lambda_i)}{(n_i-1)!} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & f^{(1)}(\lambda_i) \\ 0 & \ldots & 0 & f(\lambda_i) \end{bmatrix},$$

where $f^{(k)}$ denotes the k-th derivative of f.

The following rules constitute a so-called functional calculus for functions of matrices. They can be proved using Definition 7.2.2 or Theorem 7.2.3. These rules are also a special case of the rules for bounded linear operators.

Theorem 7.2.4. *If f, $g \in \mathcal{F}(T)$ and α, $\beta \in \mathbb{C}$, then*

- *$\alpha f + \beta g \in \mathcal{F}(T)$ and $\alpha f(T) + \beta g(T) = (\alpha f + \beta g)(T)$ (linearity),*
- *$f \cdot g \in \mathcal{F}(T)$ and $f(T) \cdot g(T) = (f \cdot g)(T)$ (multiplication),*
- *if f has a power series expansion $f(z) = \sum_{k=0}^{\infty} \alpha_k z^k$, valid in a neighborhood of $\sigma(T)$, then $f(T) = \sum_{k=0}^{\infty} \alpha_k T^k$ (power series).*

If $f \in \mathcal{F}(T)$, $g \in \mathcal{F}(f(T))$ and $h(z) = g(f(z))$, then

- *$h \in \mathcal{F}(T)$ and $h(T) = g(f(T))$ (function composition).*

The following spectral mapping theorem is a direct consequence of Theorem 7.2.3.

Theorem 7.2.5. *Let T be a complex square matrix. If $f \in \mathcal{F}(T)$, then*

$$\sigma(f(T)) = \{f(z),\, z \in \sigma(T)\} =: f(\sigma(T)).$$

Matrix-Valued Functions of Matrices

Definition 7.2.6. The family of all matrix-valued functions $F : \mathbb{C} \to \mathbb{C}^{m\times m}$, analytic in some neighborhood of $\sigma(T)$ is denoted by $\mathcal{F}^{m\times m}(T)$.

Note that a matrix-valued analytic function is equivalent to a matrix of analytic functions. Matrix-valued functions of matrices can be defined by applying Definition 7.2.2 componentwise.

Definition 7.2.7. Given a matrix $T \in \mathbb{C}^{n\times n}$ and a function $F \in \mathcal{F}^{m\times m}(T)$, the matrix $F(T) \in \mathbb{C}^{mn\times mn}$ is defined by

$$F(T) = \frac{1}{2\pi i} \oint F(z) \otimes (zI_X - T)^{-1} dz.$$

The expression $A \otimes B$ denotes the Kronecker or tensor product. The matrix $A \otimes B$ is obtained by replacing each element a_{ij} in A with the matrix $a_{ij}B$. If $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$ then $A \otimes B \in \mathbb{C}^{mn\times mn}$. For more details on the Kronecker product see [Gra81], [MN88] and [HJ94, Ch.4].

From the functional calculus for scalar functions, a functional calculus for matrix-valued functions can be derived. For example, if $F, G \in \mathcal{F}^{m\times m}(T)$, then $F \cdot G \in \mathcal{F}^{m\times m}(T)$ and $F(T) \cdot G(T) = (F \cdot G)(T)$.

By applying Theorem 7.2.3 componentwise, it can be found that

$$F(T) = (I_m \otimes V)\,\mathrm{diag}(F(J_1), \ldots, F(J_q))(I_m \otimes V^{-1}) \tag{7.5}$$

Since $\mathrm{diag}(F(J_1), \ldots, F(J_q))$ is block diagonal, the spectrum of $F(T)$ is given by the union of the spectra of the diagonal blocks $F(J_i)$. These blocks themselves contain triangular blocks with constant diagonals (or, equivalently, they are block triangular with a constant block diagonal if all the tensor products are reversed). This observation leads to the following generalization of the spectral mapping theorem.

Theorem 7.2.8. *Let T be a complex square matrix. If $F \in \mathcal{F}^{m\times m}(T)$, then*

$$\sigma(F(T)) = \bigcup_{z\in\sigma(T)} \sigma(F(z)) =: \sigma(F(\sigma(T))).$$

The following corollary follows immediately.

Corollary 7.2.9. *Let T be a complex square matrix. If $F \in \mathcal{F}^{m\times m}(T)$, then*

$$\rho(F(T)) = \max_{z\in\sigma(T)} \rho(F(z))$$

7.2.2 Functional Calculus for Functions of Bounded Linear Operators

In this and the following section the terminology and notation from §2.4 is used. We recall that the spectrum $\sigma(T)$ of a bounded linear operator contains all $\lambda \in \mathbb{C}$ for which $(\lambda I_X - T)$ does not have a bounded everywhere defined inverse. The spectrum of a bounded linear operator is a closed and bounded set.

Scalar Functions of Bounded Linear Operators

The definitions from §7.2.1 can still be used when T is a bounded linear operator. The formulation of the theorems is analogous as well. Proofs for the functional calculus and spectral mapping theorems for scalar functions of bounded linear operators can be found in [DS57, Tay58]. Only the spectral mapping theorems are stated here.

Theorem 7.2.10. *Let T be a bounded linear operator. If $f \in \mathcal{F}(T)$, then*

$$\sigma(f(T)) = f(\sigma(T)).$$

Matrix-Valued Functions of Bounded Linear Operators

It is straightforward to extend the functional calculus for scalar functions to matrix-valued functions. Using the resulting functional calculus, the following generalization of Theorem 7.2.8 to bounded linear operators can be given.

Theorem 7.2.11. *Let T be a bounded linear operator. If $F \in \mathcal{F}^{m\times m}(T)$, then*

$$\sigma(F(T)) = \sigma(F(\sigma(T))). \tag{7.6}$$

Proof. The proof is patterned after the proof of Theorem 7.2.10 [DS57, Th. 11, p. 569].

First, it is shown that $\sigma(F(T)) \supset \sigma(F(\sigma(T)))$. Let $\lambda \in \sigma(T)$ and $\mu \in \sigma(F(\lambda))$. Take v a normalized eigenvector of the matrix $F(\lambda)$ for the eigenvalue μ, i.e., $F(\lambda)v = \mu v$ and $v^*v = 1$. Define the matrix-valued function G in the domain of definition of F by

$$G(\xi) = (F(\lambda) - F(\xi))/(\lambda - \xi).$$

By the matrix-valued functional calculus it follows that

$$\begin{aligned} G(T)(I_m \otimes (\lambda I_X - T)) &= F(\lambda) \otimes I_X - F(T) \\ &= -(\mu I_m - F(\lambda)) \otimes I_X + \mu I_m \otimes I_X - F(T). \end{aligned}$$

Multiplying from the right by $v \otimes I_X$ and using $(\mu I_m - F(\lambda))v = 0$ results in

$$G(T)(v \otimes I_X)(\lambda I_X - T) = (\mu I_m \otimes I_X - F(T))(v \otimes I_X). \tag{7.7}$$

Assume $\mu \notin \sigma(F(T))$, then $\mu I_m \otimes I_X - F(T)$ has a bounded everywhere defined inverse, which we denote by A. Multiplying (7.7) from the left by $(v^* \otimes I_X)A$ leads to

$$(v^* \otimes I_X)AG(T)(v \otimes I_X)(\lambda I_X - T) = I_X.$$

Hence, $(v^* \otimes I_X)AG(T)(v \otimes I_X)$ would be a bounded everywhere defined inverse of $\lambda I_X - T$. This contradicts the assumption that $\lambda \in \sigma(T)$ and therefore $\mu \in \sigma(F(T))$.

Next, it is shown that $\sigma(F(T)) \subset \sigma(F(\sigma(T)))$. Let $\mu \in \sigma(F(T))$ and suppose that $\mu \notin \sigma(F(\sigma(T)))$. Then the function

$$H(\xi) = (F(\xi) - \mu I_m)^{-1}$$

is analytic in the same neighborhood of $\sigma(T)$ as F. By the matrix-valued functional calculus it follows that

$$H(T)(F(T) - \mu I_m \otimes I_X) = I_m \otimes I_X.$$

Hence $H(T)$ is a bounded everywhere defined inverse of $F(T) - \mu I_m \otimes I_X$ which contradicts the assumption that $\mu \in \sigma(F(T))$. □

The following analogue of Corollary 7.2.9 holds.

Corollary 7.2.12. *Let T be a bounded linear operator. If $F \in \mathcal{F}^{m\times m}(T)$, then*

$$\rho(F(T)) = \max_{z\in\sigma(T)} \rho(F(z))$$

7.2.3 Functional Calculus for Functions of Closed Linear Operators

Terminology and notation are again as in §2.4. We recall that a closed linear operator T with $\mathcal{D}(T) = X$ is bounded. The spectrum of a closed linear operator is closed, but not necessarily bounded. A definition of a function of an unbounded closed operator can be given by taking into account the behavior of the function at infinity. Operators whose spectrum is the whole complex plane have to be explicitly excluded.

Scalar Functions of Closed Linear Operators

Definition 7.2.13. The family of all scalar functions $f : \mathbb{C} \to \mathbb{C}$, analytic in some neighborhood of $\sigma(T)$ and at infinity is denoted by $\mathcal{F}_\infty(T)$.

The definition of a scalar function of a closed (possibly unbounded) linear operator can be based on the definition for bounded linear operators (see [DS57]). Let Φ be the homeomorphism of the Riemann sphere to itself defined by

$$\Phi(\lambda) = (\lambda - \alpha)^{-1}, \quad \Phi(\infty) = 0, \quad \Phi(\alpha) = \infty. \tag{7.8}$$

Definition 7.2.14. Given a closed linear operator T with a non-empty resolvent set, let α be an element of the resolvent set and $A = (T - \alpha I)^{-1}$ a bounded linear operator. For a function $f \in \mathcal{F}_\infty(T)$, the linear operator $f(T)$ is defined by $f(T) = \phi(A)$ where the function $\phi \in \mathcal{F}(A)$ is given by $\phi(z) = f(\Phi^{-1}(z))$.

The following theorem is derived in [DS57].

Theorem 7.2.15. *Given a closed linear operator T and a function $f \in \mathcal{F}_\infty(T)$, the linear operator $f(T)$, defined in Definition 7.2.14, is given by*

$$f(T) = f(\infty)I_X + \frac{1}{2\pi i}\oint f(z)(zI_X - T)^{-1}dz. \tag{7.9}$$

It is therefore independent of the choice of α from the resolvent set of T.

In [Tay58] the equality (7.9) is taken as the definition for $f(T)$.

The functional calculus is again analogous to the case of scalar functions of matrices and not repeated here.

Definition 7.2.16. The extended spectrum is defined as $\sigma_\infty(T) = \sigma(T) \cup \{\infty\}$.

The following spectral mapping theorem for scalar functions of closed linear operators follows from Theorem 7.2.10 by Definition 7.2.14 and the fact that Φ is a one-to-one mapping between $\sigma_\infty(T)$ and $\sigma(A)$ [DS57]. For a different proof see [Tay58].

Theorem 7.2.17. *Let T be a closed linear operator. If $f \in \mathcal{F}_\infty(T)$, then*

$$\sigma(f(T)) = f(\sigma_\infty(T)).$$

Matrix-Valued Functions of Closed Linear Operators

Definition 7.2.18. The family of all matrix-valued functions $F : \mathbb{C} \to \mathbb{C}^{m\times m}$, analytic in some neighborhood of $\sigma(T)$ and at infinity is denoted by $\mathcal{F}_\infty^{m\times m}(T)$.

The construction used to derive scalar functions of closed linear operators from the case of bounded linear operators can also be used for matrix-valued functions. Alternatively Theorem 7.2.15 can be applied componentwise.

Definition 7.2.19. Given a closed linear operator T and a function $F \in \mathcal{F}_\infty^{m\times m}(T)$, the linear operator $F(T)$ is defined by

$$F(T) = F(\infty) \otimes I_X + \frac{1}{2\pi i} \oint F(z) \otimes (zI_X - T)^{-1} dz.$$

A spectral mapping theorem for matrix-valued functions of closed linear operators can be derived from the one for bounded linear operators in the same way as for scalar functions.

Theorem 7.2.20. *Let T be a closed linear operator. If $F \in \mathcal{F}_\infty^{m\times m}(T)$, then*

$$\sigma(F(T)) = \sigma(F(\sigma_\infty(T))). \tag{7.10}$$

The following analogue of Corollaries 7.2.9 and 7.2.12 holds.

Corollary 7.2.21. *Let T be a closed linear operator. If $F \in \mathcal{F}_\infty^{m\times m}(T)$, then*

$$\rho(F(T)) = \max_{z\in\sigma_\infty(T)} \rho(F(z))$$

This formula is equivalent to (7.3).

7.3 Application of the Functional Calculus to Waveform Relaxation

In this section several waveform relaxation methods are described and their convergence is analyzed using the functional calculi of §7.2. The results derived with the theory based on Volterra equations (see Chapters 2 and 4 and [MN87a, LO87, VP93, JV96a, JV96b, VlV05b]) are reproduced in a much more general setting.

7.3.1 An Abstract Setting for the Convergence Analysis of Waveform Relaxation

The following general set of equations is considered

$$(I_m \otimes T)u = (L \otimes I_X)u + f, \tag{7.11}$$

with $u, f \in X^m$, X a Banach space, T an operator in X, $L \in \mathbb{C}^{m\times m}$ and I_X and I_m identity operators in X and $\mathbb{C}^m$. Typically, the operator T will be

the time derivative operator $\frac{d}{dt}$ or a discrete equivalent, although the results hold for any closed linear operator. A splitting of the form $L = L^+ + L^-$ results in the generalized waveform relaxation method

$$(I_m \otimes T)u^{(\nu)} = (L^+ \otimes I_X)u^{(\nu)} + (L^- \otimes I_X)u^{(\nu-1)} + f, \tag{7.12}$$

For the classical Jacobi (L^+: diagonal of L) or Gauss-Seidel (L^+: lower triangular part of L) splittings, one iteration requires the solution of m decoupled equations. By considering the corresponding iteration for the errors $e^{(\nu)} = u^{(\nu)} - u$

$$(I_m \otimes T)e^{(\nu)} = (L^+ \otimes I_X)e^{(\nu)} + (L^- \otimes I_X)e^{(\nu-1)}, \tag{7.13}$$

it is clear that the convergence of the waveform relaxation method can be studied using an analysis of the spectrum of the iteration operator

$$F(T) = (I_m \otimes T - L^+ \otimes I_X)^{-1}(L^- \otimes I_X). \tag{7.14}$$

The matrix-valued function F, defined by

$$F(z) = (zI - L^+)^{-1}L^- \tag{7.15}$$

is analytic in $\mathbb{C} \setminus \sigma(L^+)$ and at infinity. It is therefore always assumed that

$$\sigma(L^+) \cap \sigma(T) = \phi. \tag{7.16}$$

Equation (7.11) is considered for simplicity. The more general results from [JV96a, JV96b] for

$$(M \otimes T)u = (L \otimes I_X)u + f, \tag{7.17}$$

where I_m in (7.11) is replaced by a non-singular matrix M can be obtained by taking

$$F(z) = (M^+z - L^+)^{-1}(-M^-z + L^-) \quad \text{where} \quad M = M^+ + M^-$$

and assuming $\sigma(M^{+-1}L^+) \cap \sigma(T) = \phi$.

A whole range of waveform relaxation methods can now be analyzed by defining an appropriate 'time derivative' operator T. For each waveform relaxation method discussed in the following sections, the Banach space X, the operator T, its domain $\mathcal{D}(T)$ and spectrum $\sigma(T)$ or extended spectrum $\sigma_\infty(T)$ are specified. The spectrum and spectral radius of the waveform relaxation operator $F(T)$ can then be derived by using the theory of §7.2 as summarized by the formulae

$$\sigma(F(T)) = \sigma(F(\sigma_\infty(T))) \quad \text{and} \quad \rho(F(T)) = \max_{z \in \sigma_\infty(T)} \rho(F(z)). \tag{7.18}$$

In some cases T itself can be obtained as a function of an even simpler operator.

7.3.2 Continuous Waveform Relaxation

Continuous waveform relaxation is obtained by defining $Tx := \dot{x}$, in which case (7.11) is just the system of ordinary differential equations

$$\dot{u} = Lu + f,$$

for functions u and f defined on $[0, t_F]$. For initial value problems the extra condition $u(0) = u_0$ is added and t_F can be infinite. For periodic problems the extra condition is $u(0) = u(t_F)$.

The abstract iteration (7.12) takes the form

$$\dot{u}^{(\nu)} = L^+ u^{(\nu)} + L^- u^{(\nu-1)} + f,$$

together with $u^{(\nu)}(0) = u_0$ or $u^{(\nu)}(0) = u^{(\nu)}(t_F)$. The error iteration (7.13) becomes

$$\dot{e}^{(\nu)} = L^+ e^{(\nu)} + L^- e^{(\nu-1)},$$

together with $e^{(\nu)}(0) = 0$ or $e^{(\nu)}(0) = e^{(\nu)}(t_F)$.

Functions on Finite Time Intervals

Consider the Banach space $X = C[0, t_F]$ of continuous functions on the interval $[0, t_F]$ with norm $\|x\| = \max_{t\in[0,t_F]} \|x(t)\|$. Let T be the operator defined by $(Tx)(t) = \dot{x}(t)$ with domain

$$\mathcal{D}(T) = \{x : \dot{x} \in C[0, t_F], x(0) = 0\}.$$

T is a closed unbounded linear operator and its spectrum is empty ($\sigma(T) = \phi$). Its extended spectrum is therefore given by

$$\sigma_\infty(T) = \{\infty\},$$

see, for example, [DS57, p. 604], [Tay58, p. 297] and [Tay58, p. 291]. Together with (7.14)-(7.18) the classical result $\rho(F(T)) = 0$ is recovered [MN87a].

Functions on Infinite Time Intervals

Consider the Banach space $X = L^p([0, \infty], \mathbb{C})$, $1 \le p \le \infty$ and let T be the operator defined by $(Tx)(t) = \dot{x}(t)$ with domain

$$\begin{aligned}\mathcal{D}(T) = \{x : {} & x \text{ is absolutely continuous on } [0, a] \text{ for any } a > 0, \\ & \dot{x} \in X, x(0) = 0\}.\end{aligned}$$

T is a closed unbounded linear operator. The equation $\lambda x(t) - \dot{x}(t) = y(t)$, together with $x(0) = 0$, has a bounded solution

$$x(t) = -\int_0^t e^{\lambda(t-s)} y(s) ds$$

for every $y \in X$ whenever $\Re(\lambda) < 0$. Therefore the operator $(\lambda I_X - T)^{-1}$ is bounded and everywhere defined when $\Re(\lambda) < 0$ and, since the spectrum is closed, this gives

$$\sigma(T) = \bar{\mathbb{C}}^+, \qquad \sigma_\infty(T) = \bar{\mathbb{C}}^+ \cup \{\infty\}.$$

Together with (7.14)-(7.18) this leads to the classical result for the convergence of waveform relaxation on infinite time intervals given in Theorem 2.5.5.

Periodic Functions

Consider the Banach space $X = C[0, 1]$ and let T be the operator defined by $(Tx)(t) = \dot{x}(t)$ with domain

$$\mathcal{D}(T) = \{x : \dot{x} \in C[0, 1], x(0) = x(1)\}.$$

T is a closed unbounded linear operator. The equation $\lambda x(t) - \dot{x}(t) = y(t)$, together with $x(0) = x(1)$, has a bounded solution

$$x(t) = -\int_0^t e^{\lambda(t-s)} y(s) ds + \frac{e^\lambda}{e^\lambda - 1} \int_0^1 e^{\lambda(t-s)} y(s) ds$$

for every $y \in X$ whenever $\lambda \notin 2\pi i\mathbb{Z}$ and thus

$$\sigma(T) = 2\pi i\mathbb{Z}, \qquad \sigma_\infty(T) = 2\pi i\mathbb{Z} \cup \{\infty\},$$

from which the convergence results for time-periodic waveform relaxation in [VP93] follow. For this case see [DS57, p. 604]. Similar results can be obtained for the Banach spaces $X = L^p(-\pi, \pi)$, $1 \leq p \leq \infty$ on the unit circle [DS57, p. 605], [Tay58, p. 176].

7.3.3 Discrete Waveform Relaxation using General Linear Methods

In Chapters 2, 4 and 5 the convergence of discrete waveform relaxation using several time discretization schemes was analyzed. For discrete waveform relaxation, the operator T can be defined by discretizing the equation $\dot{x} = Tx$ using one of these schemes. In this section the theory is applied for

time discretization based on general linear methods (GLM) [But03, HW96]. The well known linear multistep methods (LMM) and implicit Runge-Kutta (IRK) methods [Bur95, But03, HW96] and also the block boundary value methods (BBVM) [BT98, IM98, IM99] belong to this class of methods. Boundary value methods (BVM) [BT98] are another generalization of linear multistep methods. The spectral properties of discrete waveform relaxation using boundary value methods can be derived in the same way as for GLMs by using the results in [BT98, IMT02, BS99].

There are several ways to formulate GLMs for an ODE of the form $\dot{x}(t) = f(t, x(t))$ (see [But03]). We choose to use the following equations (see (4.8) and (4.9))

$$\tilde{x}_i = Cx_{i-1} + \Delta t A f(\tilde{t}_i, \tilde{x}_i), \tag{7.19}$$

$$x_i = Dx_{i-1} + \Delta t B f(\tilde{t}_i, \tilde{x}_i), \tag{7.20}$$

with $A \in \mathbb{R}^{s\times s}$, $B \in \mathbb{R}^{r\times s}$, $C \in \mathbb{R}^{s\times r}$, $D \in \mathbb{R}^{r\times r}$, $\tilde{t}_i \in \mathbb{R}^s$, $\tilde{x}_i \in \mathbb{R}^s$, $x_i \in \mathbb{R}^r$ and $f : \mathbb{R}^s \times \mathbb{R}^s \to \mathbb{R}^s$. This formulation highlights the analogy with IRK methods for which $r = 1$. The s stage values $\tilde{x}_i$ are typically approximations of $x(t)$ for some t within the current time step. The r values x_i can contain, for example, approximations of $x(t)$, brought forward from previous time steps or scaled approximations of derivatives of $x(t)$. In order to have a stable method, the matrix D has to be power bounded. Further conditions are needed to ensure that the method is consistent. The matrices C and D are typically chosen such that all rows sum to one and D is of rank one.

Finite Sequences

The discrete equivalent of $\dot{x} = Tx$ is found by using (7.19)-(7.20) which results in

$$\tilde{x}_i = Cx_{i-1} + \Delta t A(T\tilde{x})_i, \tag{7.21}$$

$$x_i = Dx_{i-1} + \Delta t B(T\tilde{x})_i. \tag{7.22}$$

These equations define T as an operator in the space of sequences of stage values $\mathbb{C}^{ns}$. Using the backward shift operator $S \in \mathbb{C}^{n\times n}$, given by $(Sx)_1 = 0$, $(Sx)_i = x_{i-1}$, $1 < i \leq n$ or

$$S = \begin{bmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} \tag{7.23}$$

and taking into account the initial condition $x_0 = 0$ the set of equations (7.21) and (7.22) can also be written as

$$\tilde{x} = (C \otimes S)x + \Delta t(A \otimes I_X)T\tilde{x}, \tag{7.24}$$

$$x = (D \otimes S)x + \Delta t(B \otimes I_X)T\tilde{x}. \tag{7.25}$$

By eliminating x and using the matrix-valued functional calculus, it can be seen that $T = G(S)$ with

$$G(z) = \frac{1}{\Delta t}(A + zC(I_r - zD)^{-1}B)^{-1}. \tag{7.26}$$

Since the spectrum of S is $\{0\}$, it follows from the spectral mapping theorem for matrix-valued functions of matrices that

$$\sigma(T) = \sigma(G(\sigma(S))) = \sigma\left(\frac{1}{\Delta t}A^{-1}\right).$$

Together with (7.14)-(7.18) the results in §4.4 for the convergence analysis of discrete waveform relaxation on finite intervals are recovered.

Infinite Sequences

The same approach as for finite sequences allows T to be defined as an operator in the space of infinite sequences of stage values $l^p(\infty, \mathbb{C}^s)$. The backward shift operator in $l^p(\infty, \mathbb{C}^s)$ is defined by $(Sx)_1 = 0$, $(Sx)_i = x_{i-1}$, $i > 1$ or

$$S = \begin{bmatrix} 0 & & \\ 1 & \ddots & \\ & \ddots & \end{bmatrix}. \tag{7.27}$$

The spectrum of S is derived by considering the recurrence relation $\lambda x_i - x_{i-1} = y_i$ with $x_0 = 0$. Its solution is $x_i = \sum_{j=1}^{i} \lambda^{j-i-1} y_j$, $i \geq 1$. If $y \in l^p(\infty, \mathbb{C}^s)$, then $x \in l^p(\infty, \mathbb{C}^s)$ for $|\lambda| > 1$. Since the spectrum is closed this gives

$$\sigma(S) = \{\lambda \in \mathbb{C} : |\lambda| \leq 1\}.$$

(See also [Tay58, p. 266]). For more information about the spectral properties of Toeplitz operators such as S see [BS99, BT98]. Using the spectral mapping theorem for matrix-valued functions of bounded linear operators, one arrives at

$$\sigma(T) = \frac{1}{\Delta t} \bigcup_{|z| \leq 1} \sigma\left((A + zC(I_r - zD)^{-1}B)^{-1}\right).$$

It is shown in §4.5.1 that this is the complement of the interior of the stability domain of the GLM as defined in [But03, HW96], scaled by $\frac{1}{\Delta t}$. A similar result was originally derived for the special case of linear multistep methods in [MN87b]. In general, the interior of the stability region of a time discretization method corresponds to the resolvent set of the discrete time derivative operator in $l^p(\infty)$. Combining the expression for $\sigma(T)$ with (7.14)-(7.18) recovers the results in §4.5.2 for the convergence analysis of discrete waveform relaxation on infinite intervals.

Remark 7.3.1. Certain GLMs lead to a function G that has poles on the unit circle. The time derivative operator T for such methods is unbounded. Its spectrum can be derived directly or by using an extended functional calculus [Haa05]. Alternatively, the expression for $\rho(F(T))$ can be obtained as

$$\sup_{z \in \sigma(S)} \rho(F(G(z))),$$

i.e., by considering the function $F(G(z))$ which is again analytic on the closed unit disc.

Periodic Sequences

For time-periodic problems discretized by GLMs the time derivative operator T in C^{sn} can be formulated as a function of the circulant backward shift matrix

$$P = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 1 & \ddots & & 0 \\ & \ddots & \ddots & \vdots \\ & & 1 & 0 \end{bmatrix}, \tag{7.28}$$

with spectrum $\sigma(P) = \{e^{\frac{2\pi i j}{n}}, j = 0, \ldots, n-1\}$. Using the matrix-valued function (7.26) the operator T is given by $T = G(P)$. The corresponding spectrum is

$$\sigma(T) = \frac{1}{\Delta t} \bigcup_{\sigma(P)} \sigma\left((A + zC(I_r - zD)^{-1}B)^{-1}\right).$$

Together with (7.14)-(7.18) this results in an expression for the convergence rate of discrete waveform relaxation for time-periodic problems (for the case of linear multistep methods see Chapter 2 or [VP93]).

7.4 Tensor Products of Operators

For matrix-valued functions of the form

$$F(z) = (zI - L^+)^{-1}L^-, \tag{7.29}$$

a proof of the spectral mapping theorem can be given using the fact that

$$\sigma(A \otimes I + I \otimes B) = \sigma(A) + \sigma(B) := \{\lambda_A + \lambda_B : \lambda_A \in \sigma(A),\ \lambda_B \in \sigma(B)\}$$

for A and B bounded or closed linear operators. This is a special case of

$$\sigma(g(A \otimes I, I \otimes B)) = g(\sigma(A), \sigma(B))$$

where g is a polynomial (or a certain type of rational function with appropriate restrictions on $\sigma(A)$ and $\sigma(B)$) [BP66, Sch69, DS70, RS73, Ich78a, Ich78b].

Assume $\sigma(T) \cap \sigma(L^+) = \phi$, then the following equivalences hold

$$\begin{aligned}
&\mu \in \sigma((T \otimes I_m - I_X \otimes L^+)^{-1}(I_X \otimes L^-)) \\
&0 \in \sigma(\mu I - (T \otimes I_m - I_X \otimes L^+)^{-1}(I_X \otimes L^-)) \\
&0 \in \sigma(\mu(T \otimes I_m - I_X \otimes L^+) - (I_X \otimes L^-)) \\
&0 \in \sigma(\mu T \otimes I_m - I_X \otimes (\mu L^+ + L^-)) \\
&0 \in \sigma(\mu T) - \sigma(\mu L^+ + L^-) \\
&\exists z \in \sigma(T), w \in \sigma(\mu L^+ + L^-) : \mu z - w = 0 \\
&\exists z \in \sigma(T) : \mu z \in \sigma(\mu L^+ + L^-) \\
&\exists z \in \sigma(T) : 0 \in \sigma(\mu z I_m - \mu L^+ - L^-) \\
&\exists z \in \sigma(T) : 0 \in \sigma(\mu I_m - (z I_m - L^+)^{-1} L^-) \\
&\exists z \in \sigma(T) : \mu \in \sigma((z I_m - L^+)^{-1} L^-).
\end{aligned}$$

This can be summarized as

$$\sigma((T \otimes I_m - I_X \otimes L^+)^{-1}(I_X \otimes L^-)) = \{\mu \in \sigma((zI_m - L^+)^{-1}L^-),\ z \in \sigma(T)\}.$$

This method also extends to the more general case

$$F(z) = (zM^+ - L^+)^{-1}(-zM^- + L^-). \tag{7.30}$$

The proof presented here is only valid for rational functions of the form (7.29) or (7.30), as opposed to general analytic functions. However, the function F can be operator-valued instead of just matrix-valued. This suggests that it may be possible to extend the theory of §7.2 to operator-valued analytic functions.

7.5 Two-Grid Fourier Analysis

The functional calculus also provides an elegant way to perform the two-grid analysis presented in Chapter 3. We demonstrate this for the coarse grid correction part of a two-grid iteration applied to the one-dimensional Poisson equation. The analysis is based completely on simple shift matrices or operators and there is no need to consider Fourier modes or eigenfunctions.

As for the analysis in Chapter 3 it is assumed here that the problem is periodic. The number of fine grid points n_x is even and the number of points on the coarse grid is $\overline{n}_x = n_x/2$. We split the even and odd points and write the vector of unknown values as

$$u = \left[\begin{array}{cccc|cccc} u_0 & u_2 & \cdots & u_{n_x-2} & u_1 & u_3 & \cdots & u_{n_x-1} \end{array} \right]^T.$$

The matrix $P \in \mathbb{C}^{\overline{n}_x \times \overline{n}_x}$ is the operator that periodically shifts a sequence as follows (see also (7.28))

$$(u_0, u_1, \ldots, u_{\overline{n}_x-2}, u_{\overline{n}_x-1}) \to (u_{\overline{n}_x-1}, u_0, u_1, \ldots, u_{\overline{n}_x-2}).$$

Using this operator, the coarse grid correction operator for the one-dimensional Poisson equation (see 2.6 and Chapter 3) can be written as

$$K = \left[\begin{array}{c} I \\ \frac{1}{2}(I + P^{-1}) \end{array} \right] \overline{n}_x^{-2} \left[\begin{array}{c} P^{-1} - 2I + P \end{array} \right]^{-1} \left[\begin{array}{cc} \frac{1}{2}I & \frac{1}{4}(I+P) \end{array} \right] n_x^2 \left[\begin{array}{cc} -2I & I+P \\ I+P & -2I \end{array} \right].$$

Define the matrix-valued function $K : \mathbb{C} \to \mathbb{C}^{2\times 2}$ as

$$K(w) = \left[\begin{array}{c} 1 \\ \frac{1}{2}(1 + w^{-1}) \end{array} \right] \overline{n}_x^{-2} \left[\begin{array}{c} w^{-1} - 2 + w \end{array} \right]^{-1} \left[\begin{array}{cc} \frac{1}{2} & \frac{1}{4}(1+w) \end{array} \right] n_x^2 \left[\begin{array}{cc} -2 & 1+w \\ 1+w & -2 \end{array} \right].$$

From the functional calculus for matrices it follows that the coarse grid correction operator K is obtained by plugging the periodic shift operator P into the function $K(w)$. Other stencils, smoothers and the whole two-grid iteration can be handled in the same way. The spectrum and spectral radius of the two-grid iteration operator M can be found using the spectral mapping theorem as

$$\sigma(M) = \bigcup_{w \in \sigma(P)} \sigma(M(z)) \quad \text{and} \quad \rho(M) = \max_{w \in \sigma(P)} \rho(M(z)),$$

where $M : \mathbb{C} \to \mathbb{C}^{2\times 2}$ on the right hand side is a matrix-valued function of the form $M(z) = S(z)^{\nu_2} K(z) S(z)^{\nu_1}$.

The equivalent of a local Fourier mode analysis, where infinite grids are used, is obtained by plugging in the infinite shift operator S. This operator is defined in the space of doubly-infinite sequences $l^p(\mathbb{Z}, \mathbb{C})$. The spectrum of this Laurent operator S is the unit circle.

The two-dimensional case can easily be handled using tensor products. The grid functions are split into even-even, odd-odd, odd-even and even-odd. This results in matrix-valued functions of two complex parameters with exactly the same structure as the symbols in the Fourier analysis of Chapter 3. The whole analysis of two-grid method for time-dependent problems can now be formulated using a single four by four matrix-valued analytic function of three complex parameters. The iteration operator is obtained by plugging in the three appropriate shift operators. The spectrum is found as

$$\sigma(M) = \bigcup_{(w_t, w_x, w_y) \in \Sigma_t \times \Sigma_x \times \Sigma_y} \sigma(M(w_t, w_x, w_y)).$$

If we consider discrete waveform relaxation on infinite time intervals and infinite spatial grids (local Fourier analysis), then Σ_t is the closed unit disc and Σ_x and Σ_y are the unit circle. Because of a maximum principle the maximum of $\rho(M(w_t, w_x, w_y))$ is found on the boundary and therefore

$$\rho(\mathcal{M}) = \max_{(w_t, w_x, w_y) \in \mathbb{S}^3} \rho(M(w_t, w_x, w_y)),$$

where $\mathbb{S}$ is the unit circle $\{w \in \mathbb{C} : |w| = 1\}$.

7.6 Conclusions

A general framework for the analysis of waveform relaxation methods based on the Dunford-Taylor functional calculus was described. To do this, the theory of scalar functions of operators had to be extended to matrix-valued functions of operators. When applied to waveform relaxation for initial value problems, the results from the theory based on Volterra convolution operators are recovered. The theory presented here, however, is more general. For a given matrix splitting, a waveform relaxation method can be defined and analyzed for every closed linear operator. The theory also allowed an elegant formulation of the two-grid convergence analysis of multigrid methods.

Chapter 8

Concluding Remarks

In §8.1 I briefly discuss the codes I used for the experiments in this thesis as well as related implementations that were not mentioned in the rest of this thesis. Section 8.2 provides a summary of the previous chapters together with some conclusions. In §8.3 I suggest some topics that merit further investigation.

8.1 Implementation Aspects

The first two sections consider the codes used for the experiments in Chapters 2 to 6. In §8.1.1 the codes used for the numerical experiments are discussed. The codes used for the convergence analysis are discussed in §8.1.2. The last two sections consider implementations that use some of the ideas presented in this thesis. Section 8.1.3 discusses the implementation of a multigrid method for the package GAMD, a state-of-the-art ODE solver based on BVMs with features such as variable order and variable step size. Section 8.1.4 mentions research on combining IRK methods with finite element discretizations in space.

8.1.1 Multigrid Waveform Relaxation Code

The numerical experiments in Chapter 3 were done using a code written in C++, without any additional libraries. This code implements multigrid waveform relaxation with point and line relaxation, standard coarsening, semicoarsening and multiple semicoarsening for constant and space and time dependent 5- and 9-point stencils. Only BDF time discretization is available. Preliminary support for delay PDEs is also implemented in this code. Because of the compiled nature of the language and the fact that

it is relatively low-level, it was quite cumbersome to change the code and experiment with new ideas.

To make experimentation easier, I decided to make a new program using Python, a higher level language [vR]. It is in general less verbose than C++ and, in my opinion, more readable. Expressing the same algorithm in Python usually requires less typing than in C++. The C++ code for our specific application could be made more concise by using a library for working with multidimensional arrays such as Blitz++ [Vel98]. Some early experiments showed that the resulting code is very efficient, but also that development is much slower due to long compilation times.

Reimplementation is never an easy choice. There is a benefit to starting from scratch with the experience gained from a previous implementation. On the other hand, time spent reimplementing could have been spent on adding new features to the existing code. Furthermore, there might be a significant cost involved in mastering a new language. Using a different programming language in no way guarantees that the new implementation will be better. I found, however, that in this case it has been worth taking the risk of writing a new implementation.

High level languages such as Python often exchange efficiency of code for efficiency of coding. If necessary one can always rewrite a part of the program that is a bottleneck in a lower level language such as C or Fortran and integrate this in the original code. However, the resulting code would usually be less flexible, less portable and harder to maintain. Because good extensions, such as linear algebra routines, are available for the important subtasks of our problem, it was never deemed necessary to write low-level code.

Languages such as Python are very well suited for rapid prototyping. This is an important advantage when doing research. It should be easy to try out a new idea. Development proceeds according to an edit-test cycle instead of an edit-compile-test cycle. For a language with an explicit compilation step, separate configuration files that are read at run time allow experimenting with parameters without the need for recompilation. Often the configuration file is extended to a script, commands are added and in the end a whole new, ad-hoc scripting language could emerge. To prevent this, it is better to use a well designed scripting language to begin with. When using a language that does not require a separate compilation step, the program can be the configuration file. This makes the scope of possible experimentation much larger. The whole language can easily be used to conveniently set up experiments.

An additional advantage of Python is the availability of many libraries and specifically many packages for scientific computing. The Numeric package [ADH+01] was used to handle multidimensional arrays (e.g., four dimensions x, y, t and the stages for IRK). The Numeric extension provides

a convenient syntax to work with arrays as well as a wide range of useful functions. It is written in C for efficiency. Interfaces to BLAS and LAPACK provide support for dense linear algebra. The PySparse package [Geu02, Geu] was used to solve the sparse and banded systems resulting from discretization in time using BVMs. This package provides, amongst others, an interface to the sparse solver SuperLu [DEG+99], written in C. The multigrid algorithms, that is the iterations over grids and grid points, are written in Python. The overhead of this approach is negligible, if there is enough work in the time dimension. For a large number of time steps the efficiency of the sparse or dense solver is much more important than the efficiency of the program that calls it.

Another interesting feature of the code is that it avoids building a matrix for the fully discrete system whenever possible. All operators (smoother, residual, restriction, prolongation) work directly on grids. For constant stencils, only a few coefficients have to be stored. Such an approach is straightforward for regular grids and an important reason for considering their use. Because no adaptive methods are implemented, all grids can be allocated at the start of the computation.

Python is only one possible choice of high level language. Other possibilities are Perl or Ruby. Scheme, Common Lisp and OCaml are interesting options because they provide both the possibility of fast development using an interpreter and efficient code using a compiler. The fact that these languages are not 'main-stream' could be perceived as a disadvantage. A common choice for scientific computing is Matlab [Mat]. Matlab is a very rich language when it comes to numerical linear algebra, but it was not originally designed as a general computer language. Recent versions introduced data structures other than matrices of floating point numbers, but in my opinion still in a fairly ad-hoc way. Another disadvantage of Matlab is its cost, especially when compared to, for example, Python which is free and available on many platforms. On many systems Python is included in the initial installation.

8.1.2 Convergence Analysis Code

The convergence analysis for the experiments in Chapter 4 was implemented in Matlab. This implementation was straightforward, but quite slow. It may be possible to use Matlab for a more efficient implementation, but this would require serious restructuring of the code.

With the experience from the Matlab code, a much faster implementation was developed in the programming language ZPL [Sny99]. This is an array processing language that compiles to very efficient C code. ZPL also allows straightforward parallelization of algorithms that deal with regular grids, but this aspect of the system was not exploited. The ZPL code only

does the computation of $\rho(M(z))$ for a given set of z values. This is the most computationally intensive part. Calculation of the z values (imaginary axis, boundaries of stability domains), taking the maximum value and plotting is done by a Python program that calls the ZPL code.

8.1.3 Multigrid for the ODE Solver GAMD

During a visit to the University of Bari in Italy a multigrid method was integrated into the GAMD code. GAMD is a package for the solution of ODEs, developed by Francesca Mazzia and Felice Iavernaro. The code is based on the generalized Adams methods, a type of boundary value methods. Sophisticated methods for the control of both the step size and the order of the methods is incorporated in the code. To handle the large linear systems of the form $A \otimes I - B \otimes L$ (see Chapter 4), an iteration based on a splitting of the matrices A and B is used, so that only systems of the form $aI + bL$, with a and b scalars, have to be solved. This corresponds to the systems in the LMM case and the standard multigrid methods can be used. No block smoothing is necessary. The first results from experiments for parabolic model problems confirmed that the multigrid method allows much larger problems to be solved efficiently. A next step would be to compare this approach to the approach without the extra splitting for A and B, but with a block smoother.

8.1.4 Finite Elements in Space and IRK in Time

Waveform relaxation methods for finite element instead of finite difference discretizations were studied in [JV96a, JV96b]. Both the methods and their analysis can be extended to the finite element case.

Some experiments were done using Femlab, a package to solve PDEs using the finite element method. Applying the standard geometric multigrid method to the matrix obtained after discretization using finite elements in space and an implicit Runge-Kutta method in time, does not lead to satisfactory convergence. By providing a block smoother to the geometric multigrid of Femlab, it was confirmed that good convergence rates can be achieved. The preliminary Matlab implementation of the block smoother was, however, quite inefficient.

When a time-dependent parabolic PDE is discretized using finite elements on an general irregular grids, it is no longer straightforward to apply geometric multigrid methods. For elliptic equations, a possible solution is to use an algebraic multigrid (AMG) method. Such methods are called algebraic because they take as input the discretization matrix and no information about the geometry of the underlying mesh (such as the location of the grid points). Based on the matrix for the fine grid, a hierarchy of matrices for

coarser 'grids' and the corresponding transfer operators are automatically built. Advanced algebraic multigrid methods have been applied successfully to discretizations of time-independent problems with anisotropy, varying or jumping coefficients and irregular grids.

Just as for geometric multigrid a block smoother is necessary to achieve good convergence for time-dependent problems. The SAMG package [Stü01b, Stü01a] that implements algebraic multigrid methods for systems of PDEs, is well suited for these types of problems. This packages was used to solve a system of non-linear reaction-diffusion equations modeling the production and consumption of oxygen and carbon dioxide in fruit [VlSV+05]. For this application a Matlab interface to the SAMG code was developed.

8.2 Summary and Conclusions

It is well known that multigrid methods are an excellent choice for discretized elliptic as well as parabolic equations. Chapter 2 illustrated, using the Poisson and the heat equation as model problems, how multigrid methods for elliptic equations can be extended to multigrid methods for parabolic equations. In the following chapters the same principles were applied to derive methods for more general classes of problems and methods using different time discretizations.

In Chapter 3 methods were derived for problems with varying coefficients and coefficients that have a strong directional dependency. Two-grid Fourier mode analysis, a standard method to analyze multigrid methods for elliptic equations, was used to analyze the convergence of the methods.

Chapters 4 and 7 showed that the close link between the convergence of iterative methods for time-dependent equations and the stability of the time discretization, already established for linear multistep methods, also holds for the implicit Runge-Kutta and boundary value methods. If continuous waveform relaxation converges on the time interval $[0, \infty)$, then discrete waveform relaxation with any A-stable ODE integrator converges as well.

In conclusion, combining an appropriate multigrid method with an appropriate time discretization scheme results in an iterative method that converges approximately as fast as the multigrid method for the corresponding elliptic problem. The cost of one iteration is approximately that of one iteration for the elliptic case multiplied by the cost of integrating one scalar ODE.

Chapters 4 and 5 showed that if multigrid methods can be used it is worth considering time discretization schemes that would be very expensive in other cases. The Radau IIA implicit Runge-Kutta method and the GAM and GBDF boundary value methods were used in Chapter 4. Chebyshev spectral collocation, a spectral method, was studied in Chapter 5

In Chapter 6 differential equations with a delay term were considered. It was shown that multigrid waveform relaxation methods can be extended to handle DDEs derived from delay PDEs.

Chapter 7 presented an elegant way to determine the spectra of many iteration operators using functional calculus. The theory clearly illustrates the relation between spectra of waveform relaxation operators on infinite time domains and the stability domains of the time discretization scheme. It was briefly explained how the theory developed to analyze waveform relaxation, can also be used to analyze two-grid iterations such as the ones considered in Chapter 3.

8.3 Suggestions for Further Research

A current topic of research is the use of algebraic multigrid methods to solve large finite element discretizations of time-dependent PDEs. This is a promising direction when irregular and/or adaptive meshes are appropriate. Software such as SAMG, developed for systems of elliptic PDEs, could be used as a starting point. Preliminary results suggest that multigrid methods with well chosen block smoothers are prime candidates for solving systems of time-dependent PDEs.

Some of the methods discussed in this thesis lead to subsystems with a large number of unknowns in each spatial grid point. For very large systems it may become necessary to exploit the structure of these systems. For the banded matrices arising from BVMs, sparse solvers were used. In some cases it may be possible to use more specialized techniques such as iterative methods with Toeplitz or circulant preconditioners. For Chebyshev spectral collocation the use of fast Fourier or cosine transforms could be beneficial.

It would be interesting to incorporate ideas from existing ODE integrators into solvers for large scale problems. Codes such as GAMD [IM99] and RADAU5 [HW96] make use of error estimators, variable step size and variable order. Another approach would be to use an existing ODE integrator to solve the scalar subsystems that arise in multigrid waveform relaxation methods. Similarly, it would be very interesting to use advanced DDE integrators such as RADAR5 [GH01] as the scalar solver in multigrid methods for DPDEs. This would also allow the consideration of PDEs with multiple delays, distributed delays and delays that depend on time, position and/or the solution itself. The results from [HV03] for the stability of a model problem with both fixed and distributed delay could be useful for a convergence analysis. An important issue for general delay problems is that values of the solution in the past may be needed at times in between points of the time discretization mesh. The interpolation or continuous extension method used to approximate at such off-mesh points and its interaction with

the time discretization scheme will influence, for example, the convergence of iterative methods.

The theory developed in Chapter 7 covers most classical waveform relaxation results for linear systems of ODEs. It does not immediately cover waveform relaxation for linear systems of delay differential equations as discussed in Chapter 6. For this case, a functional calculus is needed for a larger class of functions that are not necessarily analytic at infinity. An extended functional calculus [Haa05] such as the Hille-Phillips calculus, based on the theory of semigroups of operators [HP74], would be appropriate. Another type of equations that could be studied are differential algebraic equations. Systems of differential algebraic equations can be written as (7.17) with M singular.

In this thesis the non-normality of the iterations operators for time-dependent problems was handled by considering infinite time intervals. Another option, that may provide different information, is the use of pseudospectra instead of spectra. A pseudospectral analysis of anisotropic PDEs and delay PDEs could be performed, for example. Not only spectra, but also pseudospectra of waveform relaxation operators [JOW98, LW97] could be studied using functions of operators. For a spectral mapping theorem for the pseudospectra of scalar functions of matrices and bounded linear operators see [Lui03]. It may be possible to extend this theory to matrix-valued functions.

Many formulae could be represented more naturally using matrix pencils and generalized eigenvalues (for example $\sigma(B^{-1}A) = \sigma(A, B)$). In this respect it would also be interesting to investigate how the theory of joint spectra of operators could be used [Tay70, CV78, Vas82]. The related spectral theory of tensor products of operators [BP66, Sch69, DS70, RS73, Ich78a, Ich78b] would also be needed to extend the spectral mapping theorems for scalar and matrix-valued functions to operator-valued functions.

Je n'ai fait celle-ci plus longue que parce que je n'ai pas eu le loisir de la faire plus courte. (Blaise Pascal)

Bibliography

[ADH+01] D. Ascher, P. F. Dubois, K. Hinsen, J. Hugunin, and T. Oliphant. Numerical python. `http://numeric.scipy.org`, September 2001. Lawrence Livermore National Laboratory. (168)

[AM95] P. Amodio and F. Mazzia. A boundary value approach to the numerical solution of initial value problems by multistep methods. *J. Differ. Equations Appl.*, 1(4):353–367, 1995. (101)

[AT02] L. Aceto and D. Trigiante. On the a-stable methods in the gbdf class. *Nonlinear Anal.: Real World Appl.*, 3(1):9–23, 2002. (132)

[Ber00] D. Bertaccini. A circulant preconditioner for the systems of LMF-based ODE codes. *SIAM J. Sci. Comput.*, 22(3):767–786, 2000. (96)

[BHM00] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2000. (36)

[Bjø94] M. Bjørhus. On dynamic iteration for delay differential equations. *BIT*, 43(3):325–336, 1994. (136)

[BN03] D. Bertaccini and M. K. Ng. Block $\{\omega\}$-circulant preconditioners for the systems of differential equations. *Calcolo*, 40(2):71–90, 2003. (96)

[Boy01] J. P. Boyd. *Chebyshev and Fourier spectral methods*. Dover Publications Inc., Mineola, NY, 2001. `http://www-personal.engin.umich.edu/~jpboyd`. (118, 120, 123, 125)

[BP66] A. Brown and C. Pearcy. Spectra of tensor products of operators. *Proc. Amer. Math. Soc.*, 17:162–166, 1966. (15, 163, 173)

[Bri90] N. F. Britton. Spatial structures and periodic travelling waves in an integro-differential reaction-diffusion population model. *SIAM J. Appl. Math.*, 50(6):1663–1688, 1990. (136)

[Bri03] D. Britz. Higher-order spatial discretisations in digital simulations. Algorithm for any multi-point first- or second derivative on an arbitrarily spaced grid. *Electrochem. Commun.*, 5(3):195–198, March 2003. (123)

[BS99] A. Böttcher and B. Silbermann. *Introduction to Large Truncated Toeplitz Matrices*. Springer-Verlag, New York, 1999. (160, 161)

[BT98] L. Brugnano and D. Trigiante. *Solving Differential Problems by Multistep Initial and Boundary Value Methods*. Gordon and Breach Science Publishers, Amsterdam, 1998. (82, 83, 86, 88, 89, 101, 106, 123, 124, 160, 161)

[Bur95] K. Burrage. *Parallel and Sequential Methods for Ordinary Differential Equations*. The Clarendon Press Oxford University Press, New York, 1995. (19, 82, 83, 84, 86, 160)

[But03] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons Ltd., Chichester, 2003. (82, 83, 84, 85, 86, 89, 101, 122, 123, 125, 126, 160, 162)

[CNJ01] R. H. Chan, M. K. Ng, and X.-Q. Jin. Strang-type preconditioners for systems of LMF-based ODE codes. *IMA J. Numer. Anal.*, 21(2):451–462, 2001. (96)

[CV78] Z. Ceauşescu and F.-H. Vasilescu. Tensor products and the joint spectrum in Hilbert spaces. *Proc. Amer. Math. Soc.*, 72(3):505–508, 1978. (173)

[DEG+99] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Analysis and Applications*, 20(3):720–755, 1999. `http://crd.lbl.gov/~xiaoye/SuperLU`. (169)

[DS57] N. Dunford and J. T. Schwartz. *Linear Operators. Part I.* Interscience Publishers, Inc., New York, 1957. (14, 150, 153, 155, 158, 159)

[DS70] A. T. Dash and M. Schechter. Tensor products and joint spectra. *Israel J. Math.*, 8:191–193, 1970. (15, 163, 173)

[For96] B. Fornberg. *A practical guide to pseudospectral methods*. Cambridge University Press, Cambridge, 1996. (118, 120, 123)

[For98] B. Fornberg. Calculation of weights in finite difference formulas. *SIAM Rev.*, 40(3):685–691, 1998. (123)

[Geu] R. Geus. Pysparse - Handling sparse matrices in Python. `http://people.web.psi.ch/geus/pyfemax/pysparse.html` and `http://pysparse.sourceforge.net`. (169)

[Geu02] R. Geus. *The Jacobi-Davidson algorithm for solving large sparse symmetric eigenvalue problems with application to the design of accelerator cavities*. PhD thesis, Swiss Federal Institute of Technology Zurich, December 2002. (169)

[GH01] N. Guglielmi and E. Hairer. Implementing Radau IIA methods for stiff delay differential equations. *Computing*, 67(1):1–12, 2001. (172)

[GO77] D. Gottlieb and S. A. Orszag. *Numerical analysis of spectral methods: theory and applications*. Society for Industrial and Applied Mathematics, Philadelphia, Pa., 1977. CBMS-NSF Regional Conference Series in Applied Mathematics, No. 26. (118, 120, 121, 125)

[Gra81] A. Graham. *Kronecker products and matrix calculus: with applications*. Ellis Horwood Ltd., Chichester, 1981. (7, 152)

[GVL96] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, third edition, 1996. (150)

[Haa05] M. Haase. *The Functional Calculus for Sectorial Operators*. Birkhäuser Verlag, Basel, 2005. `http://www.mathematik.uni-ulm.de/m5/haase/`. (150, 162, 173)

[Hal58] P. R. Halmos. *Finite-dimensional vector spaces*. D. Van Nostrand Co., Inc., Princeton-Toronto-New York-London, 1958. 2nd ed. (6)

[HJ90] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, 1990. (6)

[HJ94] R. A. Horn and C. R. Johnson. *Topics in matrix analysis.* Cambridge University Press, Cambridge, 1994. (6, 7, 150, 152)

[HNW93] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations. I.* Springer-Verlag, Berlin, 1993. Nonstiff problems. (83, 101)

[HP74] E. Hille and R. S. Phillips. *Functional Analysis and Semi-Groups.* American Mathematical Society, Providence, R. I., 1974. (14, 150, 173)

[HV95] G. Horton and S. Vandewalle. A space-time multigrid method for parabolic partial differential equations. *SIAM J. Sci. Comput.*, 16(4):848–864, 1995. (48, 95)

[HV03] C. Huang and S. Vandewalle. An analysis of delay-dependent stability for ordinary and partial differential equations with fixed and distributed delays. *SIAM J. Sci. Comput.*, 2003. (137, 172)

[HW96] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations. II.* Springer-Verlag, Berlin, 1996. (81, 82, 83, 84, 85, 86, 100, 101, 122, 123, 125, 126, 160, 162, 172)

[Ich78a] T. Ichinose. Spectral properties of tensor products of linear operators. I. *Trans. Amer. Math. Soc.*, 235:75–113, 1978. (15, 163, 173)

[Ich78b] T. Ichinose. Spectral properties of tensor products of linear operators. II. The approximate point spectrum and Kato essential spectrum. *Trans. Amer. Math. Soc.*, 237:223–254, 1978. (15, 163, 173)

[IM98] F. Iavernaro and F. Mazzia. Solving ordinary differential equations by generalized Adams methods: properties and implementation techniques. *Appl. Numer. Math.*, 28(2-4):107–126, 1998. (82, 86, 88, 89, 160)

[IM99] F. Iavernaro and F. Mazzia. Block-boundary value methods for the solution of ordinary differential equations. *SIAM J. Sci. Comput.*, 21(1):323–339, 1999. (82, 86, 88, 89, 101, 160, 172)

[IMT02] F. Iavernaro, F. Mazzia, and D. Trigiante. Eigenvalues and quasi-eigenvalues of banded Toeplitz matrices: some properties and applications. *Numer. Algorithms*, 31(1-4):157–170, 2002. (160)

[ISW92] G. Ierley, B. Spencer, and R. Worthing. Spectral methods in time for a class of parabolic partial differential equations. *J. Comput. Phys.*, 102(1):88–97, 1992. (121)

[IT01] F. Iavernaro and D. Trigiante. Preconditioning and conditioning of systems arising from boundary value methods. *Nonlinear Dyn. Syst. Theory*, 1(1):59–79, 2001. (96)

[Jan97] J. Janssen. *Acceleration of Waveform Relaxation Methods for Linear Ordinary and Partial Differential Equations.* Ph.D. thesis, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, December 1997. (2)

[JKL97] Z. Jackiewicz, M. Kwapisz, and E. Lo. Waveform relaxation methods for functional differential systems of neutral type. *J. Math. Anal. Appl.*, 207:255–285, 1997. (136)

[JOW98] Z. Jackiewicz, B. Owren, and B. Welfert. Pseudospectra of waveform relaxation operators. *Comput. Math. Appl.*, 36(8):67–85, 1998. (68, 173)

[JSW82] G. S. Jordan, O. J. Staffans, and R. L. Wheeler. Local analyticity in weighted L^1-spaces and applications to stability problems for Volterra equations. *Trans. Am. Math. Soc.*, 274(2):749–782, December 1982. (139, 140, 143)

[JV96a] J. Janssen and S. Vandewalle. Multigrid waveform relaxation of spatial finite element meshes: the continuous-time case. *SIAM J. Numer. Anal.*, 33(2):456–474, 1996. (xxii, 16, 19, 20, 21, 22, 23, 28, 29, 68, 100, 109, 148, 156, 157, 170)

[JV96b] J. Janssen and S. Vandewalle. Multigrid waveform relaxation on spatial finite element meshes: the discrete-time case. *SIAM J. Sci. Comput.*, 17(1):133–155, 1996. (xxii, 16, 19, 20, 23, 24, 25, 28, 29, 100, 105, 148, 156, 157, 170)

[Kel95] C. T. Kelley. *Iterative methods for linear and nonlinear equations.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995. (6)

[KM99] V. Kolmanovskii and A. Myshkis. *Introduction to the theory and applications of functional-differential equations.* Kluwer Academic Publishers, Dordrecht, 1999. (136)

[Lam73] J. D. Lambert. *Computational Methods in Ordinary Differential Equations.* John Wiley & Sons, London-New York-Sydney, 1973. (84)

[Lax02] P. D. Lax. *Functional Analysis*. Wiley-Interscience [John Wiley & Sons], New York, 2002. (14, 150)

[LO87] C. Lubich and A. Ostermann. Multigrid dynamic iteration for parabolic equations. *BIT*, 27(2):216–234, 1987. (xix, xxii, 2, 19, 20, 22, 23, 25, 82, 100, 109, 148, 156)

[LRSV82] E. Lelarasmee, A. E. Ruehli, and A. L. Sangiovanni-Vincentelli. The waveform relaxation method for time-domain analysis of large scale integrated circuits. *IEEE Trans. on CAD of IC Sys.*, 1:131–145, 1982. (2, 19)

[Lub83] C. Lubich. On the stability of linear multistep methods for Volterra convolution equations. *IMA J. Numer. Anal.*, 3(4):439–465, 1983. (22, 23, 25)

[Lui03] S.-H. Lui. A pseudospectral mapping theorem. *Math. Comp.*, 72(244):1841–1854, 2003. (173)

[LW97] A. Lumsdaine and D. Wu. Spectra and pseudospectra of waveform relaxation operators. *SIAM J. Sci. Comput.*, 18(1):286–304, 1997. (68, 100, 173)

[LW03] A. Lumsdaine and D. Wu. Krylov subspace acceleration of waveform relaxation. *SIAM J. Numer. Anal.*, 41(1):90–111, 2003. (2)

[Mat] T. Mathworks. Matlab. `http://www.mathworks.com`. (169)

[MN87a] U. Miekkala and O. Nevanlinna. Convergence of dynamic iteration methods for initial value problem. *SIAM J. Sci. Statist. Comput.*, 8(4):459–482, 1987. (xxii, 2, 19, 20, 21, 22, 23, 100, 136, 138, 148, 156, 158)

[MN87b] U. Miekkala and O. Nevanlinna. Sets of convergence and stability regions. *BIT*, 27(4):554–584, 1987. (xxii, 19, 20, 23, 24, 25, 100, 105, 148, 162)

[MN88] J. R. Magnus and H. Neudecker. *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons Ltd., Chichester, 1988. (7, 152)

[MN05] K.-A. Mardal and T. K. Nilssen. Reuse of standard preconditioners for higher-order time discretizations of parabolic PDEs. 2005. Submitted for publication. (96)

[MST] F. Mazzia, A. Sestini, and D. Trigiante. B-spline multistep method and their continuous extension. (122)

[Mul89] W. A. Mulder. A new multigrid approach to convection problems. *J. Comput. Phys.*, 83(2):303–323, 1989. (45)

[Oos95] C. W. Oosterlee. The convergence of parallel multiblock multigrid methods. *Appl. Numer. Math.*, 19(1-2):115–128, 1995. (xxv, 45)

[OW95] C. W. Oosterlee and P. Wesseling. On the robustness of a multiple semi-coarsened grid method. *Z. Angew. Math. Mech.*, 75(4):251–257, 1995. (45)

[Rin55] R. F. Rinehart. The equivalence of definitions of a matric function. *Amer. Math. Monthly*, 62:395–414, 1955. (150)

[RS73] M. Reed and B. Simon. Tensor products of closed operators on Banach spaces. *J. Functional Analysis*, 13:107–124, 1973. (15, 163, 173)

[Rud73] W. Rudin. *Functional Analysis*. McGraw-Hill Book Co., New York, 1973. (14, 150)

[Sch69] M. Schechter. On the spectra of operators on tensor products. *J. Functional Analysis*, 4:95–99, 1969. (15, 163, 173)

[Sch71] M. Schechter. *Principles of Functional Analysis*. Academic Press, New York, 1971. (14, 150)

[SN88] J. H. Saltz and V. K. Naik. Towards developing robust algorithms for solving partial differential equations on MIMD machines. *Parallel Comput.*, 6(1):19–44, 1988. (95)

[Sny99] L. Snyder. *Programming Guide to ZPL*. MIT Press, Cambridge, MA, 1999. `http://www.cs.washington.edu/research/zpl/zpl_guide.pdf`. (169)

[Str80] G. Strang. *Linear algebra and its applications*. Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1980. (6)

[Stü01a] K. Stüben. A review of algebraic multigrid. *J. Comput. Appl. Math.*, 128:281–309, 2001. (171)

[Stü01b] K. Stüben. *User's manual SAMG*, 2001. (171)

[Tay58] A. E. Taylor. *Introduction to Functional Analysis*. John Wiley & Sons Inc., New York, 1958. (14, 150, 153, 155, 158, 159, 161)

[Tay70] J. L. Taylor. A joint spectrum for several commuting operators. *J. Functional Analysis*, 6:172–191, 1970. (173)

[TE89] H. Tal-Ezer. Spectral methods in time for parabolic problems. *SIAM J. Numer. Anal.*, 26(1):1–11, 1989. (118)

[TOS01] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, San Diego, CA, 2001. (9, 12, 28, 36, 39, 48, 49, 73)

[Tre00] L. N. Trefethen. *Spectral methods in MATLAB*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. (118, 119, 120)

[TZ95] S. Ta'asan and H. Zhang. On the multigrid waveform relaxation method. *SIAM J. Sci. Comput.*, 16(5):1092–1104, 1995. (48)

[Van93] S. Vandewalle. *Parallel Multigrid Waveform Relaxation for Parabolic Problems*. B. G. Teubner, Stuttgart, 1993. (xix, xxii, 2, 19, 20, 22, 24, 68, 100, 109)

[Vas82] F.-H. Vasilescu. *Analytic functional calculus and spectral decompositions*. D. Reidel Publishing Co., Dordrecht, 1982. (150, 173)

[Vel98] T. L. Veldhuizen. Arrays in blitz++. In *Proceedings of the 2nd International Scientific Computing in Object-Oriented Parallel Environments (ISCOPE'98)*. Springer-Verlag, 1998. `http://www.oonumerics.org/blitz`. (168)

[VH95] S. Vandewalle and G. Horton. Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods. *Computing*, 54(4):317–330, 1995. (48)

[VlJV05] J. Van lent, J. Janssen, and S. Vandewalle. Multigrid waveform relaxation for delay partial differential equations. In *Time-delay systems 2004*, IFAC Proceedings Volumes, pages 203–208. Elsevier, Oxford, UK, 2005. (3)

[VlSV+05] J. Van lent, D. Smits, S. Vandewalle, N. Scheerlinck, and B. Nicolaï. Experiences with an algebraic multigrid method for a 3D biological respiration-diffusion model. *International Compumag Society (ICS) Newsletter*, 12(1):9–14, Mar. 2005. (171)

[VlV02] J. Van lent and S. Vandewalle. Multigrid waveform relaxation for anisotropic partial differential equations. *Numer. Algorithms*, 31(1-4):361–380, 2002. (2)

[VlV05a] J. Van lent and S. Vandewalle. A unified framework based on operational calculus for the convergence analysis of waveform relaxation methods. Report TW 421, Department of Computer Science, K.U.Leuven, Leuven, Belgium, Mar. 2005. (3)

[VlV05b] J. Van lent and S. Vandewalle. Multigrid methods for implicit Runge-Kutta and boundary value method discretizations of parabolic PDEs. *SIAM J. Sci. Comput.*, 27(1):67–92, August 2005. (3, 156)

[VlV05c] J. Van lent and S. Vandewalle. Waveform relaxation using spectral collocation in time. In *ICNAAM 2005 - International Conference on Numerical Analysis and Applied Mathematics 2005*, pages 850–853. Wiley-vch Verlag, 2005. (3)

[VP93] S. Vandewalle and R. Piessens. On dynamic iteration methods for solving time-periodic differential equations. *SIAM J. Numer. Anal.*, 30(1):286–303, 1993. (22, 156, 159, 162)

[vR] G. van Rossum. Python. `http://www.python.org`. (168)

[Wes92] P. Wesseling. *An Introduction to Multigrid Methods*. John Wiley & Sons Ltd., Chichester, 1992. (9, 37, 39, 48, 65)

[WJ05] R. Wienands and W. Joppich. *Practical Fourier Analysis for Multigrid Methods*. Chapman & Hall/CRC, 2005. (9, 48, 49)

[WO98] T. Washio and C. W. Oosterlee. Flexible multiple semicoarsening for three-dimensional singularly perturbed problems. *SIAM J. Sci. Comput.*, 19(5):1646–1666, 1998. (xxv, 45)

[WO01] R. Wienands and C. W. Oosterlee. On three-grid Fourier analysis for multigrid. *SIAM J. Sci. Comput.*, 23(2):651–671, 2001. (28)

[Wom90] D. E. Womble. A time-stepping algorithm for parallel computers. *SIAM J. Sci. Statist. Comput.*, 11(5):824–837, 1990. (95)

[WOW00] R. Wienands, C. W. Oosterlee, and T. Washio. Fourier analysis of GMRES(m) preconditioned by multigrid. *SIAM J. Sci. Comput.*, 22(2):582–603, 2000. (28, 115)

[Wri64] K. Wright. Chebyshev collocation methods for ordinary differential equations. *Comput. J.*, 6:358–365, 1964. (118)

[Wri70] K. Wright. Some relationships between implicit Runge-Kutta, collocation Lanczos τ methods, and their stability properties. *BIT*, 10:217–227, 1970. (118, 125)

[WSV87] J. White and A. Sangiovanni-Vincentelli. *Relaxation Techniques for the Simulation of VLSI Circuits*. Kluwer Academic Publishers, Boston, 1987. (2, 19)

[WSVOR85] J. White, A. Sangiovanni-Vincentelli, F. Odeh, and A. Ruehli. Waveform relaxation theory and practice. *Trans. Soc. for Comp. Simulation*, 2(1):95–133, 1985. (2, 19)

[Wu96] J. Wu. *Theory and applications of partial functional-differential equations*. Springer-Verlag, New York, 1996. (136)

[ZKV99] B. Zubik-Kowal and S. Vandewalle. Waveform relaxation for functional-differential equations. *SIAM J. Sci. Comput.*, 21(1):207–226, 1999. (136, 138)

Curriculum Vitae

Higher Education

2000–2006 Ph. D. in Engineering, Katholieke Universiteit Leuven, Leuven, Belgium
Thesis: Multigrid Methods for Time-Dependent Partial Differential Equations

1999–2000 M. Sc. in Machine Learning and Adaptive Computing, University of the West of England, Bristol, U.K.
Thesis: Modelling Two-Dimensional Sorting in Worker Ants

1994–1999 Burgerlijk Ingenieur in de Computerwetenschappen, optie toegepaste wiskunde (Engineer in Computer Science, option applied mathematics), Katholieke Universiteit Leuven, Leuven, Belgium
Thesis: Multigrid Methods for Solving Time-Dependent Partial Differential Equations on Space-Time Grids

Teaching

2001–2005 Teaching Assistant Numerical Linear Algebra.

Co-Supervision of Master Students

2001 Pieter De Ceuninck, Numerical simulation of the dynamics of space-plasmas

2002 Jeroen Christiaens, Multigrid methods for finite element discretisations of time-dependent partial differential equations

2003 Ingrid Joossens, Numerical simulation with neural networks

2004 Hendrik Speleers, Numerical simulation with Powell-Sabin splines

2005 Freek van de Griendt, A time-parallel time integration method

Publications

- A. Sendova-Franks, and J. Van lent, Random walk models of worker sorting in ant colonies, Journal of Theoretical Biology 217 (2), pp. 255–274, July, 2002.
- J. Van lent, and S. Vandewalle, Multigrid waveform relaxation for anisotropic partial differential equations, Numer. Algorithms 31 (1–4), pp. 361–380, December, 2002.
- J. Van lent, J. Janssen, and S. Vandewalle, Multigrid waveform relaxation for delay partial differential equations, Time-delay systems 2004 (Michiels, W. and Roose, D., eds.), IFAC Proceedings Volumes, pp. 203–208, 2005.
- J. Van lent, D. Smits, S. Vandewalle, N. Scheerlinck and B. Nicolaï, Experiences with an algebraic multigrid method for a 3D biological respiration-diffusion model, International Compumag Society (ICS) Newsletter 12 (1), pp. 9–14, March, 2005.
- J. Van lent, and S. Vandewalle, Multigrid methods for implicit Runge-Kutta and boundary value method discretizations of parabolic PDEs, SIAM J. Sci. Comput. 27 (1), pp. 67-92, August, 2005.
- J. Van lent, and S. Vandewalle, Waveform relaxation using spectral collocation in time, ICNAAM 2005 - International Conference on Numerical Analysis and Applied Mathematics 2005 (Simos, T.E. and Psihoyios, G. and Tsitouras, Ch., eds.), pp. 850–853, 2005
- J. Van lent, and S. Vandewalle, A functional calculus approach for the convergence analysis of waveform relaxation methods, 13 pp., 2005, submitted.

Conference Presentations

- Multigrid waveform relaxation for anisotropic partial differential equations, Auckland Numerical Ordinary Differential Equations, ANODE, Auckland, New Zealand, January 8–12, 2001
- Multigrid waveform relaxation for anisotropic partial differential equations, Copper Mountain Conference on Multigrid Methods, CMCMM, Copper Mountain, CO, US, April 1–6, 2001
- Multigrid Waveform Relaxation with Runge-Kutta and BVM Time-discretisation, International Conference on Scientific Computation and Differential Equations, SciCADE01, Vancouver, Canada, July 28–August 3, 2001
- Multigrid for high order time discretisation of parabolic equations, Tenth International Congress on Computational and Applied Mathematics, ICCAM2002, Leuven, Belgium, July 22–26, 2002
- Multigrid for high order time discretisation of parabolic equations, Seventh European Multigrid Conference, EMG02, Pforzheim-Hohenwart, Germany, October 7–10, 2002
- Multigrid methods for implicit Runge-Kutta and boundary value method discretizations of parabolic PDEs, International Workshop on the Technological Aspects of Mathematics - II, IWTAM2, Montecatini Terme (Pistoia), Italy, April 1–3, 2004
- Waveform relaxation with spectral collocation in time, Eleventh International Congress on Computational and Applied Mathematics, ICCAM2004, Leuven, Belgium, July 26–30, 2004
- Multigrid waveform relaxation for delay partial differential equations, Fifth IFAC International Workshop on Time Delay Systems, TDS04, Leuven, Belgium, September 8–10, 2004
- Waveform relaxation using spectral collocation in time, International Conference on Numerical Analysis and Applied Mathematics, ICNAAM 2005, Rhodes, Greece, September 16–20, 2005